

Semi-Supervised Federated Learning for LiDAR-based 3D Object Detection in Autonomous Driving

Ricardo Ramautar



TU Delft

Delft
University of
Technology

Semi-Supervised Federated Learning for LiDAR-based 3D Object Detection in Autonomous Driving

by

Ricardo Ramautar

to obtain the degree of Master of Science
at the TU Delft,
to be defended publicly on Thursday October 1, 2026 at 9:45 AM.

Student number: 6109217
Project duration: September 2025 – October 2026
Thesis committee: A.B. Ünal, PhD, TU Delft, responsible supervisor
Dr. H. Caesar, TU Delft, formal supervisor
Dr. L. Peternel, TU Delft, external examiner

Cover image generated by ChatGPT using a personal prompt

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Semi-Supervised Federated Learning for LiDAR-based 3D Object Detection in Autonomous Driving

Ricardo Ramautar

Ali Burak Ünal

Holger Caesar

Abstract

3D object detection is a fundamental perception task for autonomous driving. However, the generalization of 3D object detectors remains constrained by the limited geographical and environmental diversity of available annotated datasets. Although sensor data passively recorded by privately operated vehicles offer a promising source of diverse training data, the absence of object labels, together with the distributed and privacy-sensitive nature of such data, makes it difficult to use for training. Semi-supervised federated learning (SSFL) provides a potential solution by enabling clients to collaboratively train a shared global model on their unlabeled data without requiring raw data to be shared, while leveraging the limited labeled data available at the server to guide training. Nevertheless, SSFL remains unexplored for LiDAR-based 3D object detection. To address this gap, we implement labels-at-server SSFL for LiDAR-based 3D object detection by alternating client-side self-training with server-side supervised fine-tuning. We further investigate the impact of data heterogeneity arising naturally in autonomous-driving settings using controlled partitions of the nuScenes dataset that isolate location- and weather-based differences, alongside a more realistic log-based partition. The results show that the SSFL implementation successfully exploits unlabeled client data to improve global detection performance. Moreover, despite increased variation in pseudo-label quality and greater divergence between client updates under the non-IID partitions, global detection performance remains comparable to that of the corresponding IID configurations. Performance also remains stable under partial client participation.

1. Introduction

3D object detection is a fundamental perception task for autonomous driving, as it enables the identification and localization of surrounding traffic participants [4, 22]. However, training such models typically requires large amounts of labeled data, which are difficult to obtain as collecting and annotating 3D data is expensive and laborious [16, 33, 44]. As

a result, existing autonomous-driving datasets are often limited to specific locations and driving conditions [5, 9, 31], which may limit the ability of trained detectors to generalize across diverse driving domains [34, 37, 39].

This motivates research into alternative learning approaches for 3D object detection in autonomous driving. While semi-supervised learning (SSL) [25, 33, 40] and unsupervised learning [16, 36] can reduce the annotation burden by enabling models to learn from unlabeled data, the challenge of obtaining sufficiently diverse driving data remains. In contrast to dedicated data collection campaigns, privately operated vehicles already traverse a wide variety of locations and encounter diverse weather conditions and driving scenarios as part of regular use. Sensor data collected from privately operated vehicles therefore present a promising source of varied training data that may help improve the generalization of 3D object detectors [1, 6, 24].

However, the large volume and privacy-sensitive nature of vehicle sensor data make it difficult to gather such distributed data on a central server, particularly given strict data protection regulations such as GDPR [6, 24]. Although federated learning (FL) enables distributed clients to collaboratively train a global model without requiring raw data to be shared [23], the absence of annotations in client data poses a challenge [29]. Labels-at-server semi-supervised federated learning (SSFL) addresses this limitation by enabling clients to train on their unlabeled data while using limited labeled data at the server to guide training [8, 29]. SSFL therefore provides a natural framework for leveraging the distributed, unlabeled sensor data collected by privately operated vehicles for training 3D object detectors.

Although labels-at-server SSFL has been investigated for image classification [10, 15, 35, 42] and, more recently, 2D object detection [14, 30], its applicability to LiDAR-based 3D object detection remains unexplored. Consequently, it is unclear how labels-at-server SSFL behaves in this setting, particularly under heterogeneous client data arising from differences in driving environment. Data heterogeneity is a well-established challenge in FL, as differences between client data distributions can cause client drift, leading to slower convergence and degraded global model per-

formance [13, 35, 47]. The effects of such heterogeneity may be exacerbated in labels-at-server SSFL, as the global model used to generate pseudo-labels may generalize differently across clients, causing differences in pseudo-label error characteristics that can increase both divergence between client updates and confirmation bias [41, 45]. In autonomous-driving applications, such heterogeneity may arise naturally as vehicles operate in different locations and weather conditions [6, 37, 39]. While prior work on 2D object detection has shown that weather-based distribution shifts can adversely affect SSFL [14], it remains unclear whether location- and weather-based differences in LiDAR data have a similar effect on SSFL for 3D object detection.

This paper therefore investigates SSFL for LiDAR-based 3D object detection in a labels-at-server autonomous-driving setting, where the server has access to a limited labeled dataset and clients hold only unlabeled driving data. Specifically, we adopt the alternate training strategy introduced in SemiFL [8], which alternates client-side self-training on pseudo-labels generated by the global model with server-side supervised fine-tuning of the aggregated model. To investigate how data heterogeneity encountered in real-world driving affects SSFL performance, the resulting framework is evaluated under non-IID data partitions representing location- and weather-based heterogeneity, as well as a more realistic log-based partition combining differences in location, weather distribution, time of day, and client data volume. These client data partitions are constructed from the nuScenes dataset [5], and experiments are conducted using a LiDAR-only variant of the Cross Modal Transformer (CMT) [38].

Our contributions can be summarized as follows:

- We implement a labels-at-server SSFL framework for LiDAR-based 3D object detection, enabling a detector to learn from unlabeled client-side driving data while only a limited amount of labeled data is available at the server.
- We evaluate the impact of naturally occurring data heterogeneity on SSFL performance under controlled location- and weather-based partitions, as well as a more realistic log-based setting, in which we additionally assess robustness to partial client participation.
- We further investigate the client-level effects of the controlled non-IID partitions by analyzing inter-client update divergence and pseudo-label quality across clients.

2. Related Work

2.1. Semi-Supervised 3D Object Detection

Many semi-supervised 3D object detection (SS3DOD) methods adopt a teacher-student framework [19, 25, 33, 40, 44]. As illustrated in Figure 1, this setup comprises two network branches—a teacher and a student—that generally share the same architecture. Here, teacher predic-

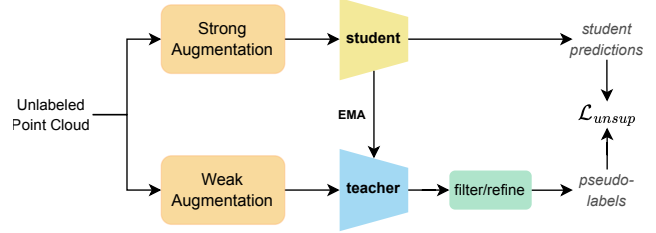


Figure 1. Generalized schematic of the self-training branch of the teacher-student framework commonly adopted in SS3DOD.

tions generated from a weakly augmented view of the input point cloud are used to supervise the student, which is trained on a more strongly augmented view of the same input. This asymmetric augmentation helps preserve pseudo-label quality while encouraging the student to learn predictions that are robust to input perturbations [28, 33]. Additionally, labeled samples are typically included in each batch to stabilize self-training and reduce confirmation bias [2, 19, 25, 33, 44]. The supervised and unsupervised losses are then combined to update the student model. Moreover, rather than remaining fixed, the teacher in SS3DOD methods is often updated as an exponential moving average (EMA) of the student following the Mean Teacher paradigm [32], allowing it to improve alongside the student in a more stable manner.

Nevertheless, the effectiveness of the teacher-student framework is strongly influenced by the quality of the pseudo-labels generated by the teacher model. Consequently, prior work on SS3DOD has proposed strategies to filter or refine teacher predictions used for student supervision. These include methods for filtering low-quality teacher predictions based on class confidence and estimated localization quality [33], cross-modal agreement [25], and adaptive thresholding [19], as well as pseudo-label refinement through spatial-temporal ensembling [40].

2.2. Labels-at-Server SSFL

Extending SS3DOD to FL is attractive for autonomous-driving applications, as it enables unlabeled sensor data collected by vehicles to be used to improve the detector without sharing raw data. However, centralized SSL methods typically train on batches containing both labeled and unlabeled samples, jointly optimizing supervised and unsupervised losses to reduce confirmation bias and improve pseudo-label quality [2, 19, 25, 33, 44]. The separation of labeled server data and unlabeled client data in the labels-at-server SSFL setting therefore makes it difficult to directly apply centralized SSL methods [8, 12, 15, 29]. Some methods address this separation by training the server model and client models in parallel before aggregating their parameters at the end of each communication round [42, 43]. Here, the server model is trained on the labeled data, while clients

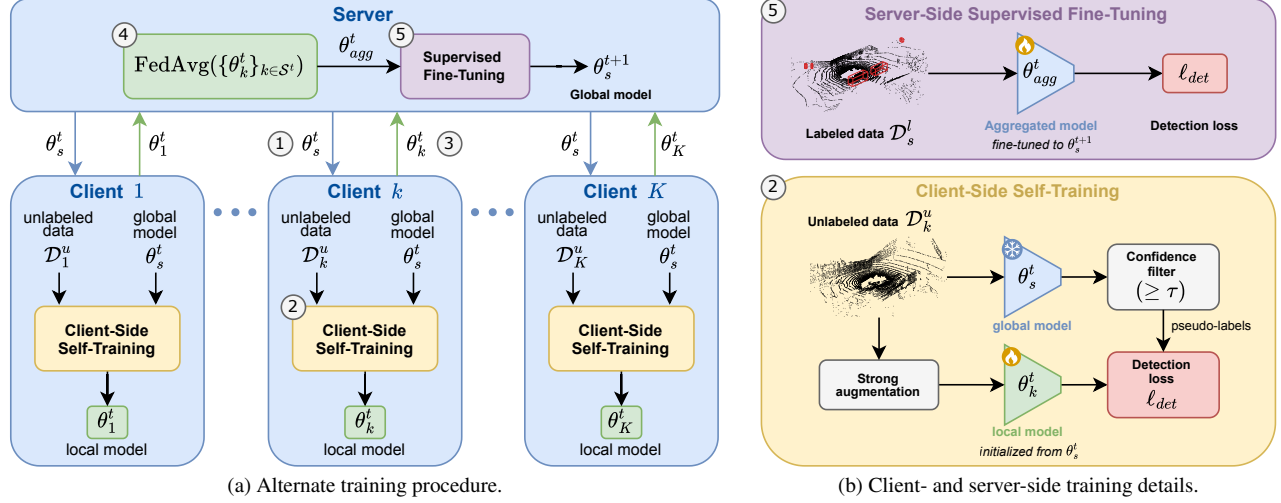


Figure 2. Overview of the implemented labels-at-server SSFL framework. In each communication round, (1) the server broadcasts the global model to the selected clients, which (2) perform self-training on their local unlabeled data and (3) return the resulting local models. (4) The server aggregates these models using FedAvg and (5) subsequently performs supervised fine-tuning on its labeled data to obtain the global model for the next round. Subfigure (a) illustrates this overall procedure, while subfigure (b) details the client-side self-training and server-side supervised fine-tuning procedures, with the snowflake and flame symbols denoting frozen and trainable parameters, respectively.

generate pseudo-labels using their evolving local models. Nevertheless, because the supervised server model is aggregated together with client models trained on imperfect pseudo-labels, the resulting global model is not directly optimized on labeled data and may be adversely affected by client updates [8]. Accordingly, labels-at-server SSFL methods commonly alternate client-side self-training with supervised fine-tuning of the aggregated model on labeled server data, thereby providing supervised correction before the next client-training phase [8, 14, 20, 21, 45].

Beyond this separation of labeled and unlabeled data, non-IID distributions across clients pose an additional challenge for labels-at-server SSFL [29, 35, 41, 45]. Part of this challenge is shared with supervised FL, where differences between local data distributions can cause client updates to drift in different directions, slowing global convergence and potentially degrading learning performance [13, 47]. Partial client participation can further compound these effects, as the global model is constantly skewed towards the data distribution of the participating clients in each round [11, 18]. In labels-at-server SSFL, however, the effects of heterogeneity extend beyond federated optimization because the global model also guides client-side self-training. In particular, under heterogeneous client distributions, the global model may generalize differently across clients, resulting in different pseudo-label error characteristics that can increase divergence between client updates. Furthermore, when the global model generalizes poorly to a particular client’s data, the resulting pseudo-labels may be less reliable, increasing confirmation bias [35, 41, 45]. Existing SSFL methods

address such heterogeneity through joint local and global pseudo-labeling [45], grouping- and frequency-based averaging [42, 43], and regularization to facilitate knowledge transfer between clients [35].

Although the effects of non-IID data in FL [3, 7, 13] and SSFL [35, 42, 43, 45] have been studied extensively in image-classification settings with label skew, the influence of statistical heterogeneity in vehicle sensor data on semi-supervised federated object detection has received limited attention, with existing work focusing on 2D detection [14, 30]. To the best of our knowledge, no prior work has investigated the impact of naturally occurring heterogeneity in point-cloud data collected by vehicles on SSFL. Consequently, it remains unclear how such heterogeneity affects labels-at-server SSFL performance for LiDAR-based 3D object detection.

3. Method

3.1. Problem Formulation

We consider the labels-at-server SSFL setting for LiDAR-based 3D object detection. The system comprises K clients indexed by $k \in \{1, \dots, K\}$ and a central server. The server holds a labeled dataset $\mathcal{D}_s^l = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_s}$ containing N_s samples, where $\mathbf{x}_i$ denotes an input point cloud and $\mathbf{y}_i = \{(\mathbf{b}_i^j, q_i^j)\}_{j=1}^{G_i}$ denotes its ground-truth annotations, comprising bounding box parameters $\mathbf{b}_i^j$ and semantic class labels q_i^j for G_i objects. By contrast, each client k holds a private unlabeled dataset $\mathcal{D}_k^u = \{\mathbf{x}_i\}_{i=1}^{N_k}$ of size N_k . The objective is therefore to learn weights θ_s for a

global detector f_{θ_s} by jointly exploiting the limited labeled data $\mathcal{D}_s^l$ available at the server and the distributed unlabeled datasets $\{\mathcal{D}_k^u\}_{k=1}^K$ across multiple communication rounds $t \in \{1, \dots, T\}$.

3.2. Alternate Training Strategy

To exploit both the limited labeled data available at the server and the private unlabeled data held by clients, we adopt the alternate training strategy introduced in SemiFL [8]. The overall training procedure, illustrated in Figure 2a, comprises five main steps repeated in each communication round. (1) At the start of round t , the central server randomly selects a subset $\mathcal{S}^t$ of m clients and broadcasts the current global model parameters θ_s^t to them. (2) Each participating client $k \in \mathcal{S}^t$ then performs self-training on its local unlabeled dataset $\mathcal{D}_k^u$. Following SemiFL, each client trains a local copy θ_k^t of the received global model parameters θ_s^t using pseudo-labels generated by the global model.

(3) After local self-training, the resulting client model parameters θ_k^t are returned to the server. (4) These model parameters are then aggregated using Federated Averaging (FedAvg) [23]:

$$\theta_{\text{agg}}^t = \sum_{k \in \mathcal{S}^t} \frac{N_k}{N} \theta_k^t, \quad (1)$$

where $N = \sum_{k \in \mathcal{S}^t} N_k$ denotes the total number of unlabeled samples held by the participating clients. (5) Finally, the server fine-tunes the aggregated model θ_{agg}^t on its labeled dataset $\mathcal{D}_s^l$ to obtain the updated global model parameters θ_s^{t+1} for the next communication round.

3.3. SSFL for 3D Object Detection

Client-side training follows the teacher-student paradigm commonly used in SS3DOD [33, 44]. As illustrated in Figure 2b, the received global model with parameters θ_s^t acts as the teacher, while the local model with parameters θ_k^t , initialized with θ_s^t , acts as the student. Specifically, the global model is used to generate predictions $\tilde{\mathbf{y}}_i$ for each unlabeled point cloud $\mathbf{x}_i \in \mathcal{D}_k^u$:

$$\tilde{\mathbf{y}}_i = f_{\theta_s^t}(\mathbf{x}_i) = \{(\mathbf{b}_i^j, q_i^j, s_i^j)\}_{j=1}^{M_i}, \quad (2)$$

where $\mathbf{b}_i^j$, q_i^j , and s_i^j denote the predicted bounding box parameters, semantic class label, and confidence score, respectively, for prediction $j \in \{1, \dots, M_i\}$. Predictions whose confidence scores are lower than a globally defined threshold τ are discarded to obtain the pseudo-label set:

$$\hat{\mathbf{y}}_i = \{(\mathbf{b}_i^j, q_i^j) \mid j \in \{1, \dots, M_i\} \wedge s_i^j \geq \tau\}. \quad (3)$$

During local training, strong augmentation is applied to the resulting pseudo-labeled samples using stochastic transformations $\mathcal{A}$, including flipping, translation, and rotation:

$$(\bar{\mathbf{x}}_i, \bar{\mathbf{y}}_i) = \mathcal{A}(\mathbf{x}_i, \hat{\mathbf{y}}_i). \quad (4)$$

Algorithm 1: Labels-at-Server SSFL

Input: K clients; m participating clients; batch size B ; E epochs; T communication rounds; confidence threshold τ ; client/server learning rates η_c and η_s ; initial model θ^0

Server Executes:

```

 $\theta_s^1 \leftarrow \text{PreTrain}(\theta^0, \mathcal{D}_s^l);$ 
foreach round  $t = 1, \dots, T$  do
   $\mathcal{S}^t \leftarrow$  (random set of  $m$  clients);
  foreach client  $k \in \mathcal{S}^t$  in parallel do
     $\theta_k^t \leftarrow \text{ClientSelfTrain}(\theta_s^t, k);$ 
   $\theta_{\text{agg}}^t \leftarrow \text{FedAvg}(\{(\theta_k^t, N_k)\}_{k \in \mathcal{S}^t});$ 
   $\theta_s^{t+1} \leftarrow \text{ServerFineTune}(\theta_{\text{agg}}^t, \mathcal{D}_s^l);$ 

```

ServerFineTune($\theta, \mathcal{D}_s^l$):

```

 $\mathcal{B} \leftarrow$  (split  $\mathcal{D}_s^l$  into batches of size  $B$ );
foreach epoch  $e = 1, \dots, E$  do
  for batch  $(\mathbf{x}, \mathbf{y}) \in \mathcal{B}$  do
     $\mathcal{L}_s^l \leftarrow \ell_{\text{det}}(f_{\theta}(\mathbf{x}), \mathbf{y});$ 
     $\theta \leftarrow \theta - \eta_s \nabla_{\theta} \mathcal{L}_s^l;$ 
  return  $\theta;$ 

```

ClientSelfTrain(θ, k): // Run on client k

```

initialize  $\hat{\mathcal{D}}_k^u \leftarrow \emptyset;$ 
initialize  $\theta_k \leftarrow \theta;$ 

// Generate pseudo-labels;
foreach point cloud  $\mathbf{x}_i \in \mathcal{D}_k^u$  do
   $\tilde{\mathbf{y}}_i \leftarrow f_{\theta}(\mathbf{x}_i) = \{(\mathbf{b}_i^j, q_i^j, s_i^j)\}_{j=1}^{M_i};$ 
   $\hat{\mathbf{y}}_i \leftarrow \{(\mathbf{b}_i^j, q_i^j) \mid j \in \{1, \dots, M_i\} \wedge s_i^j \geq \tau\};$ 
   $\hat{\mathcal{D}}_k^u \leftarrow \hat{\mathcal{D}}_k^u \cup \{(\mathbf{x}_i, \hat{\mathbf{y}}_i)\};$ 

// Self-training on pseudo-labels;
 $\mathcal{B} \leftarrow$  (split  $\hat{\mathcal{D}}_k^u$  into batches of size  $B$ );
foreach local epoch  $e = 1, \dots, E$  do
  for batch  $(\mathbf{x}, \hat{\mathbf{y}}) \in \mathcal{B}$  do
     $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leftarrow \mathcal{A}(\mathbf{x}, \hat{\mathbf{y}});$ 
     $\mathcal{L}_k^u \leftarrow \ell_{\text{det}}(f_{\theta_k}(\bar{\mathbf{x}}), \bar{\mathbf{y}});$ 
     $\theta_k \leftarrow \theta_k - \eta_c \nabla_{\theta_k} \mathcal{L}_k^u;$ 
  return  $\theta_k$  to server;

```

The local model parameters θ_k^t are then updated by minimizing the unsupervised objective over these samples:

$$\mathcal{L}_k^u(\theta) = \frac{1}{N_k} \sum_{i=1}^{N_k} \ell_{\text{det}}(f_{\theta}(\bar{\mathbf{x}}_i), \bar{\mathbf{y}}_i), \quad (5)$$

where ℓ_{det} denotes the standard CMT detection loss [38].

Following aggregation of the client model parameters, the server fine-tunes the aggregated model θ_{agg}^t on its la-

beled dataset $\mathcal{D}_s^l$ by minimizing the supervised objective

$$\mathcal{L}_s^l(\theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell_{det}(f_{\theta}(\mathbf{x}_i), \mathbf{y}_i), \quad (6)$$

to yield the updated global model parameters θ_s^{t+1} . Algorithm 1 summarizes the complete labels-at-server SSFL training procedure described above.

4. Experiments

4.1. Experimental Setup

Data Setup. To evaluate the implemented SSFL framework for LiDAR-based 3D object detection, we conduct experiments using the nuScenes dataset [5]. In addition to providing annotated LiDAR point clouds for 3D object detection, nuScenes contains driving data collected across different locations and weather conditions, making it suitable for constructing client distributions with naturally occurring environmental heterogeneity. Specifically, we construct two controlled training setups to investigate the impact of location- and weather-based heterogeneity on SSFL. For the location-based setup, we randomly sample 226 training scenes from Boston and 226 scenes from Singapore, all recorded during daytime under clear weather conditions to isolate differences arising from location. Similarly, for the weather-based setup, we randomly sample 125 training scenes recorded under clear weather conditions and 125 scenes recorded under rainy conditions. To isolate differences in weather, we only include daytime scenes recorded in Boston. For each setup, separate 2% and 5% labeled configurations are constructed by randomly selecting the corresponding fraction of scenes as labeled, with the remaining scenes treated as unlabeled training data. Corresponding validation sets are constructed from the nuScenes validation split using the same selection criteria as the training subsets.

For centralized SSL, the model is trained directly on these training subsets, whereas for SSFL, the training scenes are further divided among the server and clients. Specifically, for each controlled setup, the labeled portion is assigned to the server, while the unlabeled portion is distributed across $K = 2$ clients. For the location-based setup, the non-IID partition assigns data from Boston and Singapore to separate clients, whereas the IID partition distributes scenes from both locations evenly across the two clients. Likewise, in the weather-based setup, the non-IID partition assigns clear and rainy scenes to separate clients, while the IID partition distributes both weather conditions evenly across the two clients.

In addition to these two controlled setups, we construct a log-based data setup intended to more closely reflect how driving data would be distributed across vehicles in a real-world federated setting. Here, the nuScenes training set is

partitioned at the log level, with each log representing a continuous driving sequence. Labeled subsets containing 2% and 5% of the training data are constructed by manually selecting logs such that each resulting subset approximately preserves the overall distribution of location, weather, and time of day (i.e., daytime and nighttime). The remaining logs are treated as unlabeled training data. For SSFL, the non-IID setup randomly distributes the unlabeled training logs across $K = 16$ clients, with eight clients receiving only data from Boston and the remaining eight receiving only data from Singapore. This partition therefore captures heterogeneity in location, weather conditions, and time of day across clients, as well as variations in client dataset size resulting from differences in log sizes. By contrast, for the corresponding IID setup, we distribute the unlabeled training data across clients at the scene level, while ensuring that each client receives the same number of scenes for each location, weather condition, and time of day. We evaluate the log-based setup on the standard nuScenes validation set.

Training Configuration. We adopt the LiDAR-only configuration of CMT [38] as the detector in all experiments. For each data setup, separate models are pretrained on the corresponding 2% and 5% labeled subsets following CMT’s supervised training procedure. Specifically, we pretrain for 200 epochs in all configurations, except for the 5% log-based setup, which is pretrained for 100 epochs due to its larger labeled subset. The resulting model weights are used to initialize the SSL and SSFL experiments. For SSFL, we follow the framework described in Section 3, while for centralized SSL, we adopt the teacher-student framework described in Section 2.1, using the same augmentation and pseudo-label filtering strategies as in the SSFL setup.

Starting from the pretrained weights, the SSL and SSFL experiments are run for 20 epochs and communication rounds, respectively. In SSFL, participating clients train for one epoch on their unlabeled data, after which the aggregated model undergoes one epoch of server-side supervised fine-tuning. As in CMT [38], we use the AdamW optimizer throughout training. For both SSL and SSFL, the initial learning rate is set to 5×10^{-5} and reduced by a factor of 0.1 every five epochs and communication rounds, respectively. During server-side fine-tuning in SSFL, however, the learning rate is additionally scaled by a factor of 1.5. Both methods use a batch size of eight. In SSL, each batch comprises two labeled samples and six unlabeled samples, the unsupervised loss is weighted by a factor of four, and the teacher is updated with an EMA decay of 0.99995. In both methods, teacher predictions are filtered using a confidence threshold of $\tau = 0.4$. This threshold was selected empirically, as it performed best across the evaluated setups. The experiments in Table 1 are conducted under full client participation, whereas Table 3 reports results under varying

Table 1. Validation mAP across the three data setups with 2% and 5% of the training data labeled, for supervised training on the labeled subset, centralized SSL, and SSFL with IID and non-IID client partitions.

Data Setup	Weather-based		Location-based		Log-based	
Labeled Fraction	2%	5%	2%	5%	2%	5%
Supervised	17.45	34.28	28.71	41.71	30.91	42.65
Centralized SSL	20.47 ± 0.14	38.44 ± 0.13	31.80 ± 0.17	45.18 ± 0.10	34.84 ± 0.02	45.25 ± 0.21
SSFL (IID)	20.42 ± 0.08	38.19 ± 0.23	31.62 ± 0.29	44.76 ± 0.15	33.63 ± 0.17	44.24 ± 0.03
SSFL (Non-IID)	20.36 ± 0.09	38.38 ± 0.09	31.59 ± 0.15	44.72 ± 0.10	33.59 ± 0.22	44.47 ± 0.14

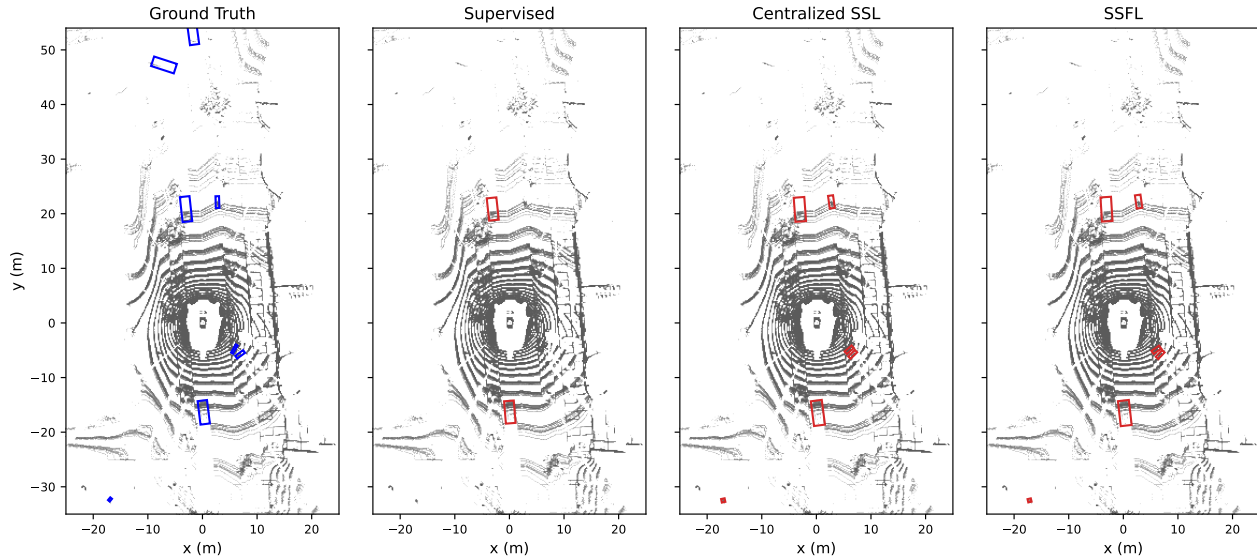


Figure 3. Qualitative comparison of 3D object detection results on a nuScenes validation sample in bird’s-eye view (BEV) following training on the log-based setup with 5% labeled data. The models are obtained through supervised training on the labeled subset, centralized SSL, and IID SSFL. Ground-truth boxes are shown in blue, and predicted boxes with a confidence score of at least 0.4 are shown in red. The BEV is cropped to exclude empty scene regions.

participation fractions. Detection performance is reported using the standard nuScenes mean average precision (mAP) [5]. For SSL and SSFL, validation mAP is averaged over three runs with different random seeds after each epoch and communication round, respectively, and the highest of these mean mAP values is reported with its corresponding standard deviation.

4.2. Comparison with Centralized SSL

Table 1 compares the validation performance of the supervised baseline trained only on the labeled subset, centralized SSL, and SSFL across the three data setups with 2% and 5% labeled data. In all experiments, SSL and SSFL improve upon the corresponding supervised baseline, showing that both approaches are able to exploit unlabeled data to improve detection performance. This performance improvement is also illustrated in Figure 3, which compares the detection results of these training strategies for a single validation sample. In this example, both centralized SSL

and IID SSFL recover several objects that are missed by the supervised baseline. While SSFL achieves performance close to centralized SSL in the weather- and location-based setups with both 2% and 5% labeled data, the results in Table 1 show a larger gap in the log-based setup, where centralized SSL outperforms IID SSFL by 1.21 mAP with 2% labeled data and 1.01 mAP with 5% labeled data.

4.3. Impact of Data Heterogeneity

To assess the impact of client data heterogeneity, we compare SSFL performance under the IID and non-IID client partitions in Table 1. Across all data setups, the IID and non-IID configurations achieve similar performance with both 2% and 5% labeled data, differing by at most 0.23 mAP. Moreover, the IID partitions do not show a consistent performance advantage over their non-IID counterparts. These results therefore indicate that the considered non-IID data partitions have little effect on global detection performance under full client participation.

Table 2. Relative increase in mean inter-client divergence under non-IID partitioning relative to IID partitioning across detector components in the location- and weather-based setups with 5% labeled data.

	Encoder	Backbone	Neck	Head
Location-based	15.8%	14.9%	21.7%	40.9%
Weather-based	9.6%	11.6%	19.1%	24.5%

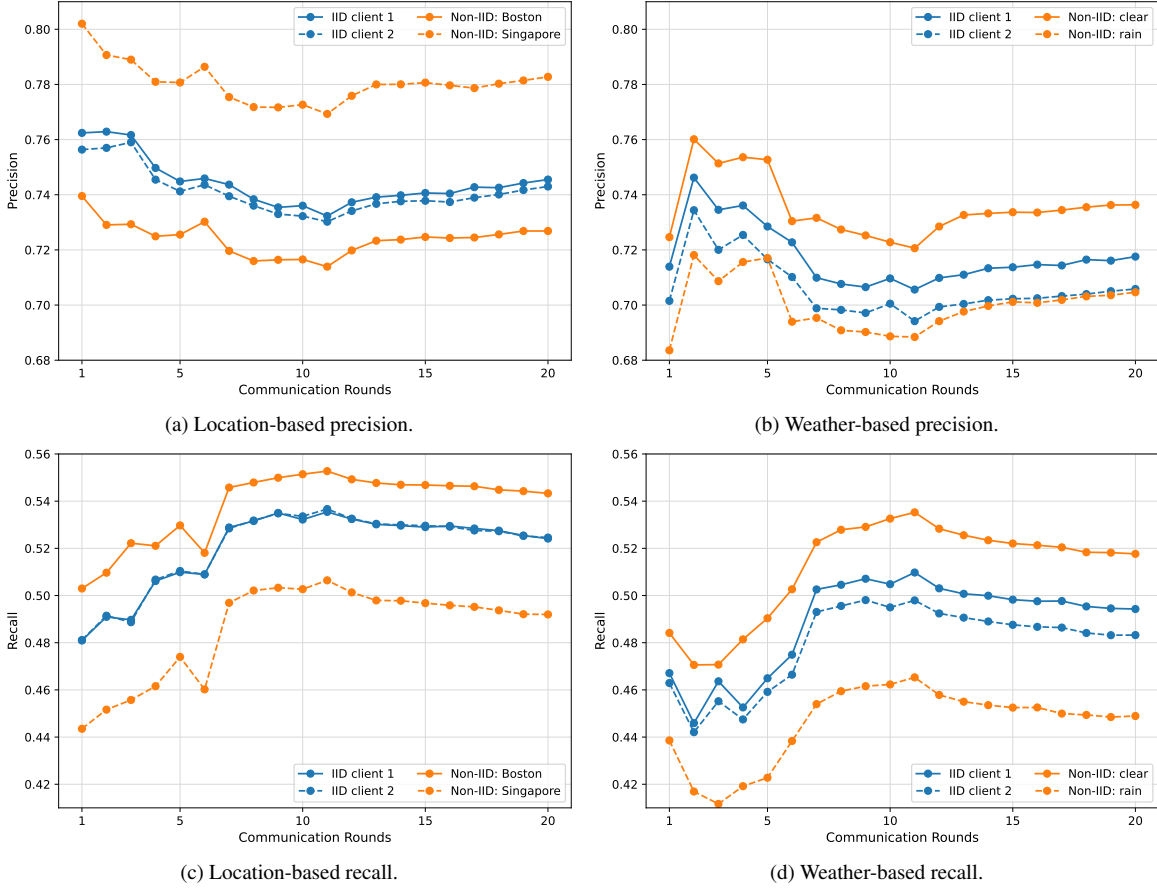


Figure 4. Pseudo-label precision and recall across communication rounds for the IID and non-IID location- and weather-based data setups with 5% labeled data.

This raises the question of whether the environmental differences used to construct the non-IID partitions actually affect client updates. To investigate this, we compare the divergence between client updates in the IID and non-IID configurations for each detector component in the controlled setups with 5% labeled data. Specifically, inter-client divergence is measured as

$$d^t = \frac{\|\Delta_1^t - \Delta_2^t\|_2}{(\|\Delta_1^t\|_2 + \|\Delta_2^t\|_2)/2}, \quad (7)$$

with $\Delta_k^t = \theta_k^t - \theta_s^t$ denoting the parameter update produced by client k in communication round t . Table 2 reports the relative increase in mean inter-client divergence under non-IID partitioning relative to IID partitioning, averaged across

communication rounds. As shown, the mean inter-client divergence is higher under non-IID partitioning for every detector component in both data setups, with relative increases ranging from 14.9% to 40.9% for the location-based setup and from 9.6% to 24.5% for the weather-based setup. Importantly, the non-IID configurations exhibit greater inter-client divergence in every communication round. This pattern is consistent with the behavior commonly observed under non-IID FL, where heterogeneous client distributions cause local updates to move in different directions [13, 17, 46, 47]. Together, these results show that the environmental differences captured by the constructed non-IID partitions do affect client updates, despite their limited effect on global detection performance.

4.4. Pseudo-Label Quality

We further evaluate the quality of the pseudo-labels generated by the global model on each client’s local data to investigate whether the global model generalizes differently across the non-IID client distributions. Figure 4 compares the precision and recall of client pseudo-labels under the IID and non-IID location- and weather-based data partitions for the 5% labeled-data setting. Here, true and false positives are defined using the nuScenes detection matching criterion [5] with a center-distance threshold of 0.5 m.

Although some variation is observed between the IID clients, their pseudo-label precision and recall remain relatively similar. In contrast, substantially larger differences emerge when the data are partitioned according to location or weather condition. For instance, in the weather-based setup, pseudo-labels generated on rainy data consistently exhibit lower precision and recall than those generated on clear-weather data (Figures 4b and 4d). For the location-based setup, pseudo-labels generated on Singapore data achieve higher precision, whereas those generated on Boston data achieve higher recall (Figures 4a and 4c). These results therefore show that the global model indeed generalizes differently across the constructed non-IID client distributions, leading to variation in pseudo-label quality across clients. Although prior SSFL work suggests that such variation in pseudo-label quality can adversely affect learning performance [41, 45], we do not observe such degradation of global detection performance in our experiments under full client participation.

4.5. Impact of Partial Client Participation

Table 3 evaluates the impact of varying the client participation fraction in the non-IID log-based setup with 2% and 5% labeled data. Varying the number of participating clients per round from 16 to 3 results in only small fluctuations in performance with no clear relationship between client participation and global model performance. Although the lowest performance is observed with the minimum participation level of 3 clients, the differences compared with full participation are marginal at 0.27 and 0.31 mAP for 2% and 5% labeled data, respectively. Overall, these results indicate that SSFL performance is relatively insensitive to partial client participation within the considered range and data setup.

5. Discussion

The experimental results demonstrate that labels-at-server SSFL can successfully exploit distributed unlabeled vehicle sensor data to improve LiDAR-based 3D object detection. Overall, centralized SSL achieves higher mean performance than SSFL, although the differences are small in the controlled data setups and more pronounced in the log-based setup. This trend is consistent with an advantage of

Table 3. Validation mAP for different client participation fractions in the non-IID log-based setup with 2% and 5% labeled data.

Participation Fraction	Validation mAP	
	2% labeled	5% labeled
16/16	33.59 ± 0.22	44.47 ± 0.14
13/16	33.70 ± 0.11	44.30 ± 0.12
10/16	33.81 ± 0.15	44.39 ± 0.04
6/16	33.66 ± 0.07	44.42 ± 0.12
3/16	33.32 ± 0.10	44.16 ± 0.14

centralized SSL, where labeled and unlabeled data can be jointly exploited during training, whereas SSFL separates client-side self-training on unlabeled data from server-side supervised fine-tuning on labeled data [8].

Interestingly, despite increased variation in pseudo-label quality across clients and greater divergence between client updates observed under the controlled non-IID partitions, global detection performance remains comparable to the corresponding IID configurations. Moreover, although partial client participation can exacerbate the effects of data heterogeneity [11, 18], we do not observe a consistent degradation in performance as client participation is reduced. One possible explanation for these findings is that the environmental heterogeneity considered in our data setups may not induce sufficiently large differences between client updates to degrade global performance. Prior work on FL and SSFL commonly simulates non-IID data through artificial label skew across clients [3, 13, 35, 42], which may induce stronger disagreement between local learning objectives than the heterogeneity considered here.

5.1. Limitations

These findings should, however, be interpreted in light of several limitations. First, all experiments are conducted on data from nuScenes, which has limited geographical and environmental diversity. The location heterogeneity captured in our location- and log-based setups is therefore restricted to differences between Boston and Singapore, which may limit the generalizability of the experimental results to other geographical regions. Similarly, the weather conditions represented in our experiments are limited to clear weather and mild rain, while other adverse conditions relevant to autonomous driving, such as snow, fog, and heavy rain, are not represented. Moreover, our work does not cover heterogeneity arising from differences in sensor configurations across clients, such as variations in LiDAR resolution, field of view, sensing range, and sensor placement. Further evaluation on data covering a broader range of locations and weather conditions, as well as greater variation in LiDAR configurations across clients, is therefore warranted to assess the generalizability of our findings.

Another limitation is that this work only evaluates the alternate training strategy introduced in SemiFL [8]. Other labels-at-server SSFL methods adopt different training mechanisms and may therefore behave differently under heterogeneous client data. Furthermore, all experiments are conducted using a single detector, namely the LiDAR-only configuration of CMT. Since prior work has shown that model architecture can influence robustness to heterogeneous data in FL [26, 27], alternative 3D object detection architectures may respond differently to the data heterogeneity considered in this work.

6. Conclusion

In this paper, we investigated labels-at-server semi-supervised federated learning for LiDAR-based 3D object detection in autonomous driving using the alternate training strategy introduced in SemiFL [8]. To investigate the impact of naturally occurring environmental heterogeneity in autonomous-driving settings, we evaluated the framework under controlled location- and weather-based data partitions alongside a more realistic log-based setup. The results show that SSFL can successfully exploit unlabeled client data to improve detection performance, demonstrating its potential for learning from distributed sensor data collected by privately operated vehicles. Overall, the implemented SSFL framework achieves performance close to centralized SSL in the controlled location- and weather-based setups, while a larger performance gap is observed in the log-based setup.

Furthermore, across the evaluated non-IID partitions, global detection performance remains comparable to the corresponding IID configurations. Performance in the log-based setup also varies only marginally across the evaluated client participation fractions. Together, these findings suggest that the considered forms of naturally occurring environmental heterogeneity have little effect on global detection performance. However, the scope of these findings is limited to the geographical and environmental diversity in nuScenes and the assumption of homogeneous LiDAR configurations across clients. Future work should therefore investigate whether these findings generalize to broader geographical and environmental conditions, as well as to heterogeneous LiDAR setups.

References

- [1] Oguzhan Akcin, Po-han Li, Shubhankar Agarwal, and Sandeep P Chinchali. Decentralized data collection for robotic fleet learning: A game-theoretic approach. In *Conference on Robot Learning*, pages 978–988. PMLR, 2023. 1
- [2] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *Proceedings of the International Joint Conference on Neural Networks*, 2020. 2
- [3] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *CoRR*, abs/1912.00818, 2019. 3, 8
- [4] Eduardo Arnold, Omar Y. Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3D object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20, 2019. 1
- [5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5, 6, 8
- [6] Vishnu Pandi Chellapandi, Liangqi Yuan, Christopher G. Brinton, Stanislaw H. Zak, and Ziran Wang. Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. *IEEE Transactions on Intelligent Vehicles*, 9, 2024. 1, 2
- [7] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *CoRR*, abs/2003.13461, 2020. 3
- [8] Enmao Diao, Jie Ding, and Vahid Tarokh. SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 4, 8, 9
- [9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012. 1
- [10] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *International Conference on Learning Representations*, 2021. 1
- [11] Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. FedVARP: Tackling the variance due to partial client participation in federated learning. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 906–916. PMLR, 2022. 3, 8
- [12] Yilun Jin, Yang Liu, Kai Chen, and Qiang Yang. Federated learning without full labels: A survey. *arXiv preprint arXiv:2303.14453*, 2023. 2
- [13] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2, 3, 7, 8
- [14] Taehyeon Kim, Eric Lin, Junu Lee, Christian Lau, and Vaikkunth Mugunthan. Navigating data heterogeneity in federated learning: A semi-supervised approach for object detection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3
- [15] Seungjoo Lee, Thanh-Long V Le, Jaemin Shin, and Sung-Ju Lee. (FL)²: Overcoming few labels in federated semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2

- [16] Ted Lentsch, Holger Caesar, and Darius M Gavrila. UNION: Unsupervised 3D object detection using object appearance-based pseudo-classes. *Advances in Neural Information Processing Systems*, 37:22028–22046, 2024. 1
- [17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 7
- [18] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*, 2020. 3, 8
- [19] Chuandong Liu, Chenqiang Gao, Fangcen Liu, Pengcheng Li, Deyu Meng, and Xinbo Gao. Hierarchical supervision and shuffle data augmentation for 3D semi-supervised object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2023. 2
- [20] Xinyi Liu, Linghui Zhu, Shu Tao Xia, Yong Jiang, and Xue Yang. GDST: Global distillation self-training for semi-supervised federated learning. In *Proceedings - IEEE Global Communications Conference, GLOBECOM*, 2021. 3
- [21] Shubham Malaviya, Manish Shukla, Pratik Korat, and Sachin Lodha. FedFAME: A data augmentation free framework based on model contrastive learning for federated semi-supervised learning. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1114–1121. Association for Computing Machinery, 2023. 3
- [22] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3D object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131, 2023. 1
- [23] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, 2017. 1, 4
- [24] Christian Neumeyer, Mario Bijelic, and Darius M. Gavrila. Learning to predict motion from raw 3D object detections. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1241–1247. Institute of Electrical and Electronics Engineers Inc., 2022. 1
- [25] Jinhyung Park, Chenfeng Xu, Yiyang Zhou, Masayoshi Tomizuka, and Wei Zhan. DetMatch: Two teachers are better than one for joint 2D and 3D semi-supervised object detection. In *Computer Vision – ECCV 2022*, 2022. 1, 2
- [26] Sara Pieri, Jose Restom, Samuel Horváth, and Hisham Cholakkal. Handling data heterogeneity via architectural design for federated visual recognition. In *Advances in Neural Information Processing Systems*, pages 4115–4136. Curran Associates, Inc., 2023. 9
- [27] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10061–10071, 2022. 9
- [28] 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. In *Advances in Neural Information Processing Systems*, 2020. 2
- [29] Zixing Song, Xiangli Yang, Yifei Zhang, Xinyu Fu, Zenglin Xu, and Irwin King. A systematic survey on federated semi-supervised learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8244–8252. International Joint Conferences on Artificial Intelligence Organization, 2024. Survey Track. 1, 2, 3
- [30] Xiangqing Su, Yan Huo, Xiaoxuan Wang, Jian Mao, Tao Jing, and Xin Fan. Enhancing object detection in IoV: A federated semi-supervised learning approach with data assessment. *IEEE Internet of Things Journal*, 12(19):40809–40821, 2025. 1, 3
- [31] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurélien 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 Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 1
- [32] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Neural Information Processing Systems*, 2017. 2
- [33] 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 Computer Society Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 4
- [34] 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 (CVPR)*, 2020. 1
- [35] Zhiguo Wang, Xintong Wang, Ruoyu Sun, and Tsung-Hui Chang. Federated semi-supervised learning with class distribution mismatch. *CoRR*, abs/2111.00010, 2021. 1, 2, 3, 8
- [36] Hai Wu, Shijia Zhao, Xun Huang, Chenglu Wen, Xin Li, and Cheng Wang. Commonsense prototype for outdoor unsupervised 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14968–14977, 2024. 1
- [37] Qiangeng Xu, Yin Zhou, Weiye Wang, Charles R. Qi, and Dragomir Anguelov. SPG: Unsupervised domain adaptation for 3D object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15446–15456, 2021. 1, 2
- [38] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal trans-

- former: Towards fast and robust 3D object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2023. [2](#), [4](#), [5](#)
- [39] 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. [1](#), [2](#)
- [40] Junbo Yin, Jin Fang, Dingfu Zhou, Liangjun Zhang, Cheng Zhong Xu, Jianbing Shen, and Wenguan Wang. Semi-supervised 3D object detection with proficient teachers. In *Computer Vision – ECCV 2022*, 2022. [1](#), [2](#)
- [41] Chao Zhang, Fangzhao Wu, Jingwei Yi, Derong Xu, Yang Yu, Jindong Wang, Yidong Wang, Tong Xu, Xing Xie, and Enhong Chen. Non-IID always bad? semi-supervised heterogeneous federated learning with local knowledge enhancement. In *International Conference on Information and Knowledge Management, Proceedings*, pages 3257–3267. Association for Computing Machinery, 2023. [2](#), [3](#), [8](#)
- [42] Zhe Zhang, Shiyao Ma, Jiangtian Nie, Yi Wu, Qiang Yan, Xiaoke Xu, and Dusit Niyato. Semi-supervised federated learning with non-IID data: Algorithm and system design. In *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 157–164, 2021. [1](#), [2](#), [3](#), [8](#)
- [43] Zhengming Zhang, Yaoqing Yang, Zhewei Yao, Yujun Yan, Joseph E. Gonzalez, Kannan Ramchandran, and Michael W. Mahoney. Improving semi-supervised federated learning by reducing the gradient diversity of models. *2021 IEEE International Conference on Big Data (Big Data)*, 2021. [2](#), [3](#)
- [44] Na Zhao, Tat Seng Chua, and Gim Hee Lee. SESS: Self-ensembling semi-supervised 3D object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. [1](#), [2](#), [4](#)
- [45] Qing Zhao, Jielei Chu, Zhaoyu Li, Wei Huang, Zhipeng Luo, and Tianrui Li. FedLGMatch: Federated semi-supervised learning via joint local and global pseudo labeling. *Knowledge-Based Systems*, 320:113642, 2025. [2](#), [3](#), [8](#)
- [46] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-IID data. *CoRR*, abs/1806.00582, 2018. [7](#)
- [47] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-IID data: A survey. *Neurocomputing*, 465:371–390, 2021. [2](#), [3](#), [7](#)