Journal of Systems Architecture 168 (2025) 103584

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

FedPillarNet: Unifying personalized and global features for federated 3D LiDAR object detection[☆]

Boyang Li, Siheng Ren, Shuai Zhao, Mingyue Cui, Kai Huang [*]

School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong 510275, China

ARTICLE INFO

Keywords:
Federated learning
Distributed learning
Lidar point clouds
Object detection
Non-iid
Cyber–physical systems

ABSTRACT

Object detection based on 3D Lidar point clouds through deep learning is widely applied in the cyber–physical system such as autonomous driving. Compared to the centrally training paradigm of conventional deep learning, federated learning (Fl) has inherent advantages in data diversity, privacy and practically due to the collaborative training with data stored in local devices. However, the non-independent and non-identically (non-iid) problems caused by the heterogeneous Lidar point clouds affect the training effectiveness of Fl, leading to the difficulty in convergence. Existing methodologies struggle with adapting to diverse data distributions across individual clients within the fixed Fl framework, in which the personalized features are not preserved, resulting in suboptimal model performance, as well as decreased generalization capability across the entire clients. To solve the problems, this paper proposed the FedPillarNet, a Fl framework concentrating on improving the training effectiveness through the Maximum Mean Discrepancy (Mmd) loss as well as the combination of personalized and global features. Extensive experiments on multiple types of heterogeneous scenarios demonstrate that the proposed FedPillarNet outperforms the existing methods in mitigating non-iid challenges in Lidar point clouds.

1. Introduction

The accurate identification and localization of objects are imperative for subsequent decision-making and global planning in autonomous driving. As a type of ranging sensor, Light Detection and Ranging (Lidar) perceives the surrounding environments in the form of a huge amount of reflected points, referred to as the point clouds that describe the scenario. On this basis, deep learning-based object detection utilizing 3D Lidar point clouds is extensively applied and studied in autonomous driving [1].

Conventional deep learning methods adopt centralized paradigm by gathering dataset in a central server to train different types of models [2,3]. Unfortunately, this type of framework has several drawbacks when training a detection model based on Lidar point clouds in autonomous driving. Firstly, the complexity and the variability of the traffic environment makes it challenging to create Lidar point clouds dataset containing abundant scenarios [4]. Secondly, dataset centralization raises concerns related to privacy and security in autonomous driving, as all the point clouds are uploaded on a single server [5]. In addition, such models are not suitable to be directly deployed on decentralized computing devices as it tends to overfit to the specific point clouds distributions during training [6], resulting in a decrease in model generalization performance. The aforementioned problems lead to challenges in the adaptability of trained models in real-world scenarios.

Diverging from the centrally training paradigm, federated learning (Fl) emerges as an innovative approach to tackle the above mentioned issues with data diversity, privacy concerns, and practicality in deep learning [7,8]. Fl provides collaborative training across distributed devices instead of centralizing data on a single server. Each device conducts local model training independently, and the resulting local parameters are aggregated to update the global model. This decentralized approach not only safeguards user privacy but also facilitates collective model improvement. Fl represents a paradigm shift in deep learning, offering a flexible and efficient solution to address issues related to large-scale datasets, privacy preservation, and real-world applicability [9–11].

Despite the series of advantages associated with Fl, it is not straightforward to be directly used for training based on Lidar point clouds

[☆] This work was supported in part by the Guangxi Key R & D Program under Grant No. GuikeAB24010324, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515011485.

[*] Corresponding author.

E-mail addresses: liby83@mail.sysu.edu.cn (B. Li), rensh6@mail2.sysu.edu.cn (S. Ren), zhaosh56@mail.sysu.edu.cn (S. Zhao), cuimy5@mail.sysu.edu.cn (M. Cui), huangk36@mail.sysu.edu.cn (K. Huang).

https://doi.org/10.1016/j.sysarc.2025.103584
Received 14 March 2025; Received in revised form 27 August 2025; Accepted 16 September 2025
Available online 30 September 2025

due to the non-independent and non identically (non-iid) problem [12]. Non-independence means that there is correlation between random variables while non-identical distribution means that the samples are not sampled from the same distribution. Specifically, point clouds generated by different LIDAR sensors exhibit significant variations in density. Additionally, the dynamic nature of autonomous driving scenarios results in varying label distributions across different clients, thereby increasing uncertainty in data distribution. Existing FL methods such as FedAvg [13] and FedProx [14] concentrate on designing the parameter aggregation to improve the training performance. However, these methods fail to effectively handle the non-iid of LIDAR point clouds as the relationship among local and global loss, as well as the impact of the individualized features are not expectedly considered, consequently, leading to a notable degradation in model accuracy and convergence. This imposes significant problems for the application of the FL architecture to autonomous driving.

In order to solve the aforementioned problems, this paper propose the FedPillarNet, a FL object detection framework used for heterogeneous LIDAR point clouds in autonomous driving. The core idea of the framework in this paper is to separate global and local features during the training process, handling each type of feature distinctly, thereby avoiding problems including the loss oscillation, non-convergence, and loss of personalized features caused by non-iid problem. The PillarFeatureNet [15] is used as part of our feature extractor, with its output sent into the backbone for training a federated global detector using the Maximum Mean Discrepancy (MMD) loss [16]. Meanwhile, an attention module is employed as a senior feature extractor to extract personalized features for computing the local client loss. These two types of loss are then aggregated for iteration. Compared with our previous work in [17], the key advantages of the proposed framework include: (i) improved trained model accuracy based on the combination of the personalized and global features, (ii) guaranteed convergence collaborative training process among the multiple clients with the mathematical discussion, (iii) robustness in the framework to diversed LIDAR point clouds. Extensive experiments demonstrate that our framework outperforms other FL frameworks in terms of convergence rate, accuracy, and utility. To sum up, the contributions of this paper are listed as follows:

- A FL object detection framework, FedPillarNet, is proposed for heterogeneous LIDAR point clouds in autonomous driving.
- A flexible structure as well as the robust loss function is designed for collaborative training while preserving the personalized point cloud features in the individual clients.
- The mathematical discussion and proof are provided regarding the convergence of the proposed framework.
- Extensive experiments are performed to demonstrate the utility and superiority of proposed framework.

The remainder of the paper is organized as follows. Section 2 reviews the related work. In Section 3 we present the constructed framework and the mathematical discussion. Section 4 presents and analyzes the experiment results. Finally, Section 5 makes a conclusion.

2. Related work

This section introduces the related researches including the learning-based object detection algorithms using LIDAR point clouds, non-iid problem caused by LIDAR point clouds, and the FL method based on non-iid data, respectively.

2.1. Object detection based on LIDAR point clouds

Currently, existing learning-based object detection algorithms using LIDAR point clouds are mainly divided into two main categories: region proposal-based and single-stage-based methods. In the region proposal-based algorithms, exemplified by studies such as [18,19], candidate regions of interest (ROI) are firstly generated and then classified to achieve object detection. Conversely, single-stage network algorithms [15] directly predict the positions and categories of objects from point clouds, eliminating the requirements for an additional region proposal generation step, thus offering faster detection speeds albeit potentially sacrificing part of the detection accuracy. However, despite the balance achieved between detection speed and accuracy by the existing researches, most algorithms are designed under the centralized training paradigm, which means that the model is trained and optimized from a huge amount of LIDAR point clouds with centralized computational resources. It is worth noting that while centralized training methods perform well in certain scenarios, challenges exist when handling large-scale dataset or complex scenarios. Distributed training has been emphasized in order to improve scalability and resolve the conflict between storage and training.

Although distributed training offers advantages such as utilizing multiple computing devices for collaborative processing to accelerate training and better accommodating the training requirements of massive, complex and diverse scenarios, there is limited exploration into the trained detection models under a distributed training paradigm based on LIDAR point clouds [20]. Therefore, how to apply distributed training techniques to object detection tasks based on LIDAR point clouds still requires further exploration [21].

2.2. Non-iid problem caused by LiDAR point clouds

The non-independent and non-identically distributed (Non-iid) problem stands as a fundamental challenge in federated learning-based methodologies. This predicament stems from the unique features of heterogeneous data on each client, resulting in the manifestation of client drift during the training process [22,23]. Client drift denotes the fluctuation in data distribution across clients over successive training iterations. Such fluctuations impede the maintenance of model consistency across disparate clients, thereby compromising the generalization capability and performance of the model.

For the non-iid problem of LIDAR point clouds in autonomous driving scenarios, the heterogeneous characteristics and discrepancies in label distributions become the main challenges. Firstly, diverse LIDAR sensors result in distinct features including the spatial distribution and density in the point clouds, attributable to differences in operational principles, sampling frequencies, and resolutions among the sensors. For instance, LIDAR sensors from different manufacturers may employ distinct scanning patterns or utilize varying wavelengths, resulting in variations in point cloud characteristics. Consequently, such disparities challenge the conventional assumption of independence and identical distribution in machine learning models, necessitating the development of adaptive methodologies capable of accommodating these variances.

Secondly, the inherent diversity of driving scenarios further exacerbates the challenges, giving rise to fluctuations in data label distributions. External factors such as varying traffic conditions and environmental changes contribute to this variability. In urban scenarios, for example, dense traffic patterns and complex road structures yield point clouds with unique label distributions, distinct from those encountered in highway environments characterized by sparse traffic and simpler road layouts. Moreover, environmental factors such as weather conditions also influence the distribution of labels within LIDAR point clouds. Thus, the non-iid problem stemming from LIDAR point clouds presents a multifaceted challenge for learning-based method in autonomous driving.

2.3. Federated learning based on non-iid data

Federated learning (FL) is a type of distributed learning paradigm where models are collaboratively trained with data preserved in local devices. Unfortunately, non-iid data severely affects both local and global convergence performance within Fl due to the unbalanced data distribution [24,25].

Various strategies have been proposed to tackle the problems. In [26], a federated framework called the FAug is proposed to perform data augmentation using shared Generative Adversarial Networks (GANs) to mitigate distribution imbalances between clients. In [27], a client selection algorithm is designed based on deep Q-learning to select the best subset of devices for collaboration in each communication round. FedProx [28] encourages all clients to collaboratively design the loss function to mitigate the effects of unbalanced data distribution. In [29], a federated learning framework based on hierarchical Bayesian modeling and variational inference is introduced to adjust the regularization effect of the global model by generating confidence values to trade-off the client's parameters in the aggregation phase. FedLAW [30], a federated learning strategy incorporating learnable weights is proposed to enhance generalization by leveraging a global weight contraction effect. In [31], the authors exploit a novel client selection mechanism to handle environment heterogeneity by eliminating diverged models. However, these approaches neglect the intrinsic personalized features unique to each client, resulting in suboptimal training performance.

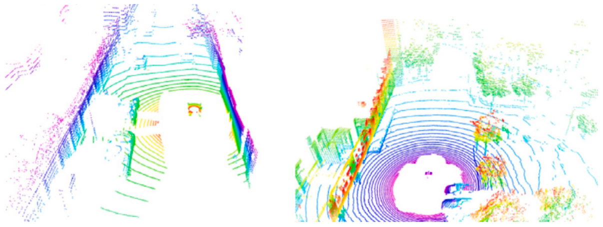
Another type of Fl solution is to divide the framework into personalized and globally shared modules. [32] designed a Fl framework containing both global and personalized embedding modules, merging the two types of information with Multi-Layer Perceptrons (MLPs) for robust features. [33,34] both adopt a personalized optimization approach based on the global models. They decouple the optimization process of personalized models from the learning process of global models. Specially, [34] constructs an HPM network to expedite fast optimization by aggregating historical client model information. [33] regulates the distance between the global model and the personalized model through loss term regularization, additionally incorporating personalized parameter bias based on the global model. On the other hand, [35,36] employ strategies centered on training a global model extractor alongside a personalized detection header. Meanwhile, [37] explores data segmentation to enhance global model performance by sharing key information. [38] proposes a method involving the separation of global and personalized information within a dataset using a conditional policy network, where personalized information is leveraged for tuning the client models upon completion of global training. [20] proposed the FedPCN structure for point clouds classification, dividing the framework into shared and personalized modules to improve generalizability and avoid convergence at false optima. Nonetheless, these methods adopt FedAvg [13] as the parameter aggregation algorithm, leading to unstable trained global models due to the heterogeneous representations of features. Thus, while Fl offers promising solutions for collaborative model training, addressing non-iid challenges necessitates leveraging personalized features and seeking stable parameter aggregation techniques to ensure optimal performance in heterogeneous scenarios.

Recent advances in privacy-preserving and personalized federated learning have also explored complementary directions. [39] proposes a robust Fl framework to defend against model poisoning attacks. [40] introduces a hybrid Fl strategy to preserve privacy in IoT environments. [41] develops a privacy-preserving and auditable split learning approach tailored for healthcare scenarios. [42,43] attempt to separate the global information and personalized information in the data, where the global information accelerates the convergence of the global model while the personalized information is used to fine-tune the local models. These directions offer promising extensions for improving security, flexibility, and accountability in Fl systems.

Based on the above discussions, non-iid caused by Lidar point clouds affects the performance of Fl. To solve the problems, this paper designs the robust global-personalized feature extraction structure as well as the specific loss function.

Table 1
Comparison of Lidar where the metrics are Hz, m, deg, and mean number of points per frame.

	FPS	Range	FOV	vRes	hRes	Mean PPF
VLS-128	5–20	300	40	0.11	0.2	220,668
Pandar64	10,20	200	40	0.167	0.2	108,556
HDL-64	5–20	120	26.9	0.33	0.16	124,296
OS1-64	10,20	120	33.2	0.53	0.35	57,749
Pandar40	10,20	200	40	0.33	0.2	67,687
VLP-32C	5–20	200	40	0.33	0.2	50,372
HDL-32C	5–20	100	41.33	1.33	0.16	60,056
RS-32	5,10,20	200	40	0.33	0.2	64,657
OS1-16	10,20	120	33.2	0.53	0.35	14,696
VLP-16	5–20	100	30	2.0	0.2	25,551



(a) HDL-32

(b) OS1-64

Fig. 1. Point clouds produced by different Lidar [47].

3. Motivation

The Non-iid problem caused by point clouds mainly include the diverse geometry features from different types of Lidar and the label distributions. This section analyzes the heterogeneous point clouds in density characteristics as well as label distribution scenarios as our motivation example.

In [44], the author compare the performance of several types of mainstream Lidar on the market, as shown in Table 1.

It can be observed that the significant differences in design metrics among different Lidar products directly result in substantial variations in both the quantity and density of point clouds. For example, the average number of points for VLS-128 is 220,668, while for OS1-16 it is only 14,696, indicating that the number of points for VLS-128 is approximately 15 times that of OS1-16. This vast difference vividly reflects the heterogeneity in design parameters such as detection range, FOV, and angular resolution among different Lidar, exerting a significant impact on the shape, density of point clouds.

Specifically, a larger detection range allows Lidar to capture targets from further distances, thereby generating more points. Additionally, a wider FOV enables Lidar to scan a broader area, covering more geographical information and producing richer point clouds. The angular resolution affects the accuracy and detail of point clouds, with higher angular resolutions capturing more fine-grained features and generating denser point clouds. Fig. 1 shows the point clouds generated by two different structures of Lidar. From the figures, it can be observed that variation in design metrics not only affects the quantity of point cloud data but also directly impacts its density, quality, and accuracy. The heterogeneous shape characteristics make it challenge in Fl.

For the label distributions, the traffic scenarios in public datasets such as KITTI [45] and Nuscene [46] are totally different because of the diverse collection conditions including cities, devices, roads, traffic rules, and weathers, leading to the unbalanced label distributions. Further, the problems of the real label distributions among distributed clients will be more complicated in Fl. To solve the non-iid problem caused by Lidar point clouds is the main motivation of this paper.

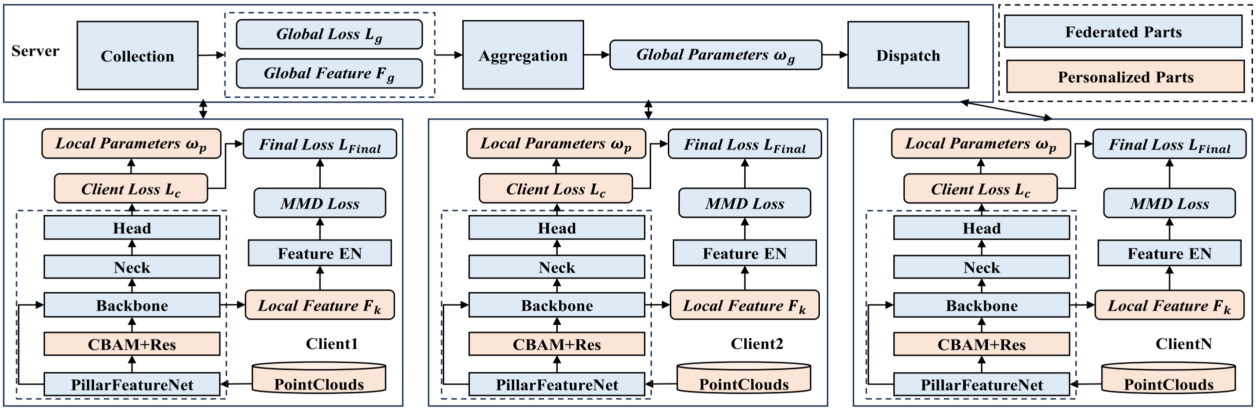


Fig. 2. The framework of the proposed FedPillarNet.

4. Approach

Fig. 2 displays the whole architecture of our proposed FedPillarNet. In the following subsections, we will first introduce the whole network architecture in Section 4.1, followed by global federated learning modeling in Section 4.2. In Section 4.3, we introduce the personalized feature extractor module, which optimizes the model for local data. Then the detailed implementation including the structure, and the training strategy in the server and the clients is provided in Section 4.4, followed by the discussion and mathematical proof regarding the convergence of the FedPillarNet in Section 4.5, privacy considerations in Section 4.6.

4.1. Network architecture

Comparing to the convolution operation on 2D images, the operation on 3D LIDAR point clouds requires much more computation resource due to the sparsity and irregular structure of LIDAR point clouds. To overcome this challenge, inspired by the work in PointPillar [15], our framework employs a similar strategy to handle the point clouds. Initially, 3D LIDAR point clouds are projected into 2D pseudo images. These pseudo images are subsequently forwarded to successive modules for further computation. This type of transformation not only facilitates more efficient processing but also streamlines subsequent operations for 2D images.

As illustrated in Fig. 2, the FedPillarNet comprises two types of modules. The blue boxes represent the global modules, while pink boxes denote the local personalized components. Within each client, the global modules encompass critical components including the PillarFeatureNet, the Backbone, the Neck, and the Head. These modules are responsible for transforming 3D points into pseudo images, extracting features, and detecting targets. The global modules in the server collect shared features from each clients, perform parameter aggregation, and distribute the aggregated parameters back to the clients for collaborative global model training in the next round. In contrast, the local personalized modules are deployed individually in each client to extract and analyze local personalized features. The combination of the MMD loss, which comprises both global aggregated and local personalized components, is utilized to update the parameters.

Given the architecture of the proposed FedPillarNet, which is segregated into server and client components, the parameters of the respective models are distinguished as the ω_g and the ω_p, respectively.

In the server model, ω_g is associated with a foundational network structure derived from PointPillar, which facilitates the collaborative training of a robust global model across multiple clients in a federated manner. To mitigate the non-iid effects, a type of robust loss function based on the concept of the Maximum Mean Discrepancy (MMD) is designed to measure the distance between local and aggregated global features distributions, which will be elaborated upon in the subsequent section for a comprehensive discussion.

On the other hand, ω_p within the local client model is configured with a personalized feature extractor specifically tailored to directly extract features from the local pseudo-images, which allows for the extraction of personalized features directly relevant to the clients data distribution. Consequently, the locally extracted loss is employed to update the local parameters, thereby enriching the trained model with unique personalized features based on the corresponding clients dataset distribution. This personalized approach ensures that the model could capture and leverage the client-specific characteristics, enhancing the overall effectiveness and adaptability in the local diverse scenarios.

In summary, our structure incorporates federated-trained global model and the locally trained personalized model. As a result, the model of each client exhibits unique characteristics, as it is a hybrid of the two components. This type of combination not only enhances the overall training process by leveraging both global and personalized insights, but also ensures that the model retains crucial personalized features for adapting to local scenarios effectively. This hybrid structure achieves a balance between the global robustness and local adaptability, thereby enhancing the performance in different types of scenarios among multiple clients.

4.2. Global federated learning modeling

As mentioned above, the global part within our structure adopts the FL paradigm to collaboratively train a federated model with parameters ω_g. Directly aggregating non-iid data from all clients into a single model leads to convergence difficulties and overfitting problem. To mitigate this issue, the concept of the Maximum Mean Discrepancy (MMD) is employed to design the loss function for quantifying the divergence distance among the multiple data distributions.

$$MMD = \left\| \frac{1}{|X|}\sum_{i=1}^{|X|}\phi\left(x_i\right) - \frac{1}{|Y|}\sum_{j=1}^{|Y|}\phi\left(y_j\right)\right\|_{\mathcal{H}} \tag{1}$$

MMD is a type of mathematical description for measuring the distance between the different data distributions of related variables. The basic principle of MMD lies in its foundation upon kernel functions and the statistical concept of maximum mean discrepancy. Kernel functions are employed to handle nonlinearities by mapping data into high-dimensional spaces to quantify the difference between the means of data distributions, thereby effectively capturing the nonlinear relationships. As displayed in Eq. (1). X and Y are the sets of variables that individually conform to a probability distribution. For each element $x_i \in X$ and $y_j \in Y$, the mapping kernel function ϕ is designed to map x_i and y_j into the Hilbert space $\mathcal{H}$. MMD is defined as the upper definitive bound on the expected difference between X and Y in $\mathcal{H}$.

Considering the non-iid challenges in FL in our scenario, conventional loss functions may inadequately guide the global model in addressing these distributional disparities. Introducing the concept of MMD as part of the loss function essentially entails incorporating considerations of distributional differences into the optimization process. By penalizing the differences between data distributions during optimization, the model is encouraged to better adapt to the diverse data characteristics of individual clients, thereby enhancing its generalization capacity. Leveraging MMD involves mapping data into high-dimensional spaces and computing the corresponding squared differences, enabling more accurate quantification of differences between data distributions and offering a more effective approach to tackling non-iid data challenges in federated learning. In the following steps, the federated system formulation and the designation of the loss function based on MMD will be described in detail.

4.2.1. Federated system formulation

In the federal learning, K clients collaboratively training a global model in FL across R rounds, where each client k in a single round r trains m epochs. N represents the total amount of the dataset, while N_k denotes the dataset size corresponding to client k. According to the principles of FL, the relationships among the locally extracted features F_k^t at epoch t and the global features F_g^r within each round r are displayed in Eq. (2).

$$F_g^r = \frac{1}{N \cdot m} \sum_{k=1}^{K} \sum_{t=1}^{m} N_k \cdot F_k^t \tag{2}$$

4.2.2. Design of loss function

The loss function $\mathcal{L}$ in our federated system comprises two components, the global loss L_g and the local loss L_{l_k} for each client k across all epochs within each round. The global loss function L_g is a consolidation of loss function across all training rounds, computed as the weighted average of the combined local loss function L_{l_k} contributed by each client, proportional to their respective data contributions. The expression for L_g is listed in Eq. (3).

$$L_g = \frac{1}{N} \sum_{k=1}^{K} N_k L_{l_k} \tag{3}$$

In each round r, the local combined loss function L_{l_k} of each client k comprises constituent parts, the local loss L_{c_k} and the MMD loss, as depicted in Eq. (4). L_{c_k} reflects the loss of individualized and global parts incurred by k during training on its local data within the given round. This loss component captures the deviation between the predictions made by the local model and the ground truth labels for the corresponding dataset. Incorporating the MMD loss into the local combined loss L_{l_k} provides a mechanism for encouraging model convergence towards a more globally representative solution based on the multiple non-iid data distributions. The MMD loss quantifies the discrepancy between the distributions of local features extracted by individual clients and the global features aggregated across all clients. By penalizing this distributional discrepancy, the MMD loss fosters alignment between local and global models, promoting convergence towards a model that generalizes well across diverse data sources.

$$L_{l_k} = \frac{1}{N_k} \sum_{t=1}^{m} \left(L_{c_k}^t + \frac{\mu}{2} MMD^2 \left(F_g^{r-1}, F_k^t \right) \right) \tag{4}$$

Specifically, the local features F_k^t and the aggregated global features F_g^{r-1} from last round are combined for MMD computation, fostering a holistic understanding that captures the nuanced client-specific details and the overarching global patterns. The core of the crafted designed loss function is beneficial to guide the local model to assimilate global features. By minimizing disparities between local personalized and global aggregated features representations using MMD as the loss metric, the combined loss function effectively addresses the challenges associated with non-iid problem caused by the point cloud densities and the diverse label distributions in autonomous driving scenarios in FL.

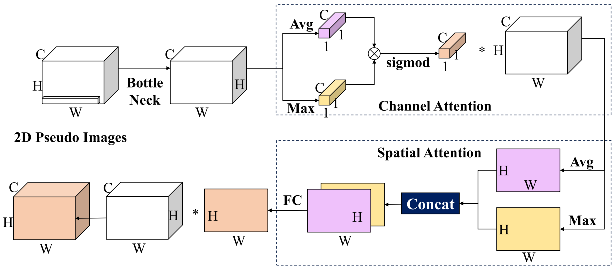


Fig. 3. The designation of personalized feature extractor.

The hyperparameter μ is used for fine-tuning the influence of the MMD loss relative to the personalized and the aggregated features within the overall loss function. Adjusting μ enables us to control the emphasis placed on encouraging alignment between local and global models, thereby striking an appropriate balance between preserving local data characteristics and promoting global model convergence. The performance of μ will be discussed in the experiment section.

In conclusion, based on the designed loss function, the optimization objective is to simultaneously minimize the global loss L_g and the local loss L_c, as depicted in Eq. (5).

$$V - min \quad \mathcal{L}(L_g, L_{c_1} = \sum_{t=1}^{m} L_{c_1}^t ... L_{c_k} = \sum_{t=1}^{m} L_{c_k}^t) \tag{5}$$

The joint optimization of L_g and L_c encapsulates the overarching objective of our FL framework, to collaboratively train a global model that performs well across diverse data sources while respecting the constraints and contributions of individual clients, which process promotes collaborative learning and information sharing, facilitating the development of a robust and effective model.

4.3. Personalized feature extractor

Based on the fact that there exists the personalized characteristics from point clouds on each local device with corresponding label distributions, an aggregated global model collaboratively trained by other clients cannot meet the requirements of the local scenario. In other words, the global model may not perform well in some clients when there is a big difference between the local and the global distribution.

Hence, to address this problem effectively, a personalized module is designed in the proposed network architecture, aimed at extracting individualized features. Due to the characteristics of the federated model, clients and servers have the same structure with same types of parameters. Therefore, the personalized module needs to have the property of plug-and-play without affecting the overall model structure.

In this context, leveraging the concept of 2D pseudo images generated through pillar projection through PillarFeatureNet, the object detection task has been transformed from 3D to 2D space. Due to the great success in capturing long-range dependencies and attending to relevant features in 2D image detection tasks and the flexible fetching properties in [48], a type of an attention-based module is used as the personalized module within our framework. On one hand, attention-based module facilitates the dynamic weighting of features based on their importance, enabling the network to focus on pertinent regions within the pseudo images generated from LIDAR point clouds. On the other hand, the flexibility of the attention mechanism allows the personalized module adapt to variations in local distributions across different devices, which is crucial for ensuring that the model can effectively extract personalized features that are representative of the unique characteristics in each client. More importantly, the attention module easily satisfies to extract features under the same scale of input and output parameters.

There are various options for attention modules. They are broadly categorized into Spatial Attention Module (SAM), Channel Attention

Module (CAM), and Convolutional Attention Module (CBAM), aiming to solve the problems of gradient vanishing and degradation in deep networks. A designation of the proposed personalized feature extractor is displayed in Fig. 3, where the input LIDAR pseudo images are extracted in the personalized modules for further feature maps computation and feature extraction. In this study, the personalized module will be investigated using comparison tests to select the combination of strategies with the best results in the experiment section.

4.4. Detailed implementation

4.4.1. The implementation of FedPillarNet

Considering the detailed implementation of the proposed FedPillarNet, the OpenPCDet [49] point cloud detection framework, a set of object detection library based on LIDAR point clouds including the basic PointPillar is chosen to design the $model-server$ and the $model-client$. Therefore, the PillarFeatureNet, the Backbone, and the Head in our structure are directly implemented based on the OpenPCDet. Other key modules including the personalized extractor, parameter collection, aggregation, and dispatch are incrementally implemented.

During each training round, several clients are randomly selected for local training in the specified epochs. Parameter aggregation is then performed to compute the current global parameters. In the client side, $model-client$ passes the public part of the parameters to the server side for aggregation. In the server side, $model-server$ dispatches the aggregated parameters back to the public part of $model-client$ for local collaboratively training.

Algorithm 1 Training process of FedPillarNet.

```
1: Initialize all the local and global parameters.
2: for each round $r$ in $R$ do
3:     Dispatch the $\omega_g^{r-1}$, $F_g^{r-1}$ to all clients.
4:     for each Epoch $t$ in $m$ do
5:         for each client $k$ in $K$ do
6:             $F_k^t \leftarrow \frac{1}{N_k} \sum_{e \in P_k} F_k^e$    ▷ Local features
7:             $L_{c_k}^t \leftarrow Forward(F_k^t)$    ▷ Local personalized loss
8:             $L_{l_k}^t \leftarrow \frac{1}{N_k} \sum \left( L_{ck}^t + \frac{\mu}{2} M\text{MD}^2 \left( F_g^{r-1}, F_k^t \right) \right)$    ▷ Local combined loss
9:             $\omega_{pk}^t \leftarrow \omega_{pk}^{t-1} + \eta \frac{\partial L_{c_k}^t}{\partial \omega_{pk}{}^{t-1}}$, $\omega_{gk}^t \leftarrow \omega_g^{r-1} + \eta \frac{\partial L_{l_k}^t}{\partial \omega_g{}^{r-1}}$    ▷ Update parameters
10:        end for
11:    end for
12:    $F_g^r \leftarrow \frac{1}{N} \sum_{k=1}^K N_k \cdot F_k$    ▷ Global Features
13:    $L_g \leftarrow \frac{1}{N} \sum_{k=1}^K N_k \cdot L_{l_k}$    ▷ Global loss
14:    $\omega_g^r \leftarrow \frac{1}{N} \sum_{k=1}^K \omega_{gk}^r$    ▷ Global parameters
15: end for
```

4.4.2. Training process

Alg. 1 describes the detailed training process of the FedPillarNet. The parameters in the framework are generally divided into ω_g and ω_p, referred to the parameters in the global federated and local personalized module, respectively. All the local and global parameters are initialized as zero in the first step. At the beginning of each round r, current global parameters and features are dispatched to all the clients for updating the corresponding loss (line 3). For each training epoch t of the client k, the local features F_k^t are extracted. Then the local personalized loss $L_{c_k}^t$ is computed through the forward operation (line 6–7). At the same time, the MMD loss aiming to measure the distance between the global and local feature distributions is also calculated. The combination of the two loss in the local client is used to update the local parameters ω_p (line 8–9).

In the following steps, current global features are weighted fused through the local features provided by all the clients (line 13). Besides, the global loss L_g is also fused by the combination loss from each client (line 13–14). By the end of this round, the global parameters are aggregated from multiple clients (line 15). In this way, the collaboratively trained global model could accelerate the training process while the personalized models preserve the local characteristics in the proposed FedPillarNet.

In addition, to extract the personalized features without affecting the model convergence, according to the pioneering work in [50], the personalized constraints are satisfied as follows.

$$\frac{\partial L\left(\hat{y}_k^i, y_k^i\right)}{\partial \omega_p^i} = \frac{\partial L\left(f\left(x_k^i; \omega_g, \omega_p^1, \dots, \omega_p^n\right), y_k^i\right)}{\partial \omega_p^j} = 0, \forall j \neq i \tag{6}$$

$\hat{y}_k^i$ and y_k^i are the predicted results and the corresponding ground truth values, respectively. $x_k^i \in P_k$, ω_p indicates the local personalized parameters. A requirement of the split personalized constraints is that the local personalized loss L_{c_k} of client k is independent of any other clients personalized parameters. The corollary of the constraint is that for the local training process on the client, the gradient of the local loss with respect to the personalized parameters of other clients are set as zero. In other means, it is assumed that the personalized part of the clients are affected by the global loss.

4.5. Convergence discussion

One of the advantage in the proposed framework is ensured converge with the global optimal solution when training non-iid LIDAR point clouds using FL. In this section, the discussions and mathematical proofs regarding the model convergence are provided. We present the convergence results for the scenario where all devices participate in training.

The definitions among the local and global loss function L_l, L_g are listed in Eqs. (3) and (4) in FL. For the discussion, We firstly make the following assumptions about the loss functions $L_{l_1}, \dots, L_{l_k}$, which are common in convergence proofs for FL and are applicable to various methods such as *L2-norm*, regularized linear regression, logistic regression, and softmax [51,52].

Assumption 1. $L_{l_1}, \dots, L_{l_k}$ are all L-smooth: For all v and w, $L_{l_i}(v) \leq L_{l_i}(w) + (v-w)^\top \nabla L_{l_i}(w) + \frac{L}{2}\|v-w\|^2$.

Assumption 2. $L_{l_1}, \dots, L_{l_k}$ are all M-strongly convex: For all v and w, $L_{l_i}(v) \geq L_{l_i}(w) + (v-w)^\top \nabla L_{l_i}(w) + \frac{\mu}{2}\|v-w\|^2$.

Assumption 3. Let ξ_{ti} be uniformly randomly sampled from the local device i. The variance of the random gradient is bounded: $\mathbb{E}\left[\|\nabla L_{l_i}(w_i^t; \xi_i^t) - \nabla L_{l_i}(w_i^t)\|^2\right] \leq \sigma_k^2$.

Assumption 4. Let ξ_{ti} be uniformly randomly sampled from the local device i. $\mathbb{E}\left[\|L_{l_i}(w_i^t; \xi_i^t) - L_{l_i}(w)\|\right] \leq \rho\|w_i^t - w\|$.

Assumption 5. The expected squared norm of the random gradient is uniformly bounded, for $i = 1, \dots, K$, $t = 1, \dots, T-1$, $\mathbb{E}\|\nabla L_{l_i}(w_i^t; \xi_i^t)\|^2 \leq G^2$.

For the equation of L_l in Eq. (4) and Assumption 4, we can expand the detailed terms as follows:

$$\begin{aligned} MMD^2(F_g, F_k) &= \left|\mathbb{P}_{global} - \mathbb{P}_{local}\right|^2 \\ &\leq \frac{\rho^2}{K} \sum_{i=1}^{K} \|\bar{\omega} - \omega^t\|^2 \end{aligned} \tag{7}$$

Where $\mathbb{P}_{global}$ and $\mathbb{P}_{local}$ in Eq. (7) represent the global and local feature means. w^t represents the optimal weights of client in round t.

Therefore, we can establish the following upper and lower bounds for each local loss function in Eq. (8).

$$L_{ck}^{t} \leq L_{l_k} \leq L_{ck}^{t} + \frac{\mu\rho^2}{2K} \sum_{i=1}^{K} \|\bar{\omega} - \omega^{t}\|^2 \tag{8}$$

As a result, if the upper and lower bounds of L_{l_k} converge to the same value, then L_{l_k} converges due to the squeeze theorem. The situations are same for the global loss function L_g. Our aim is to analyze the corresponding upper and lower bounds. After conducting a series of derivations on the loss function expression, a broader definition for mathematical forms resembling the right bound of the global loss function L_g is listed in Eqs. (9) and (10).

$$\min_{\omega_1,\ldots,\omega_n \in \mathbb{R}^d} L_g(\omega) := F(\omega) + \frac{\mu}{2} g(\omega) \tag{9}$$

where

$$F(\omega) := \frac{1}{K} \sum_{i=1}^{K} F_i(\omega_i), \quad g(w) := \frac{1}{K} \sum_{i=1}^{K} \|\omega_i - \bar{\omega}\|^2 \tag{10}$$

In Eqs. (9) and (10), $\mu \geq 0$ is a penalty parameter, $\omega_1, \ldots, \omega_n \in \mathbb{R}^d$ are local parameters, $x := (\omega_1, \omega_2, \ldots, \omega_n) \in \mathbb{R}^{nd}$ is the result of the global mixed model. $\bar{w} := \frac{1}{K}\sum_{i=1}^{K} \omega_i$ is the average of local models. We have already made the assumptions in Assumption 2. that F is strongly convex. Hence, Eq. (10) has a unique solution, which we denote as $x(\mu) := (x_1(\mu), \ldots, x_n(\mu)) \in \mathbb{R}^{nd}$. Furthermore, we define $\bar{x}(\mu) := \frac{1}{K}\sum_{i=1}^{K} x_i(\mu)$.

Overall, the more local gradient descent steps are performed, the closer the local model gets to a pure local model. The more average steps are performed, the closer the local models get to their averages. We assume that a global update is performed every E rounds. Let $p = \frac{1}{E}$, the relative quantities of local update steps and averaging steps are controlled by the parameter p, the expected number of local update steps is $\frac{1}{p}$, the expected number of consecutive aggregation steps is $\frac{1}{1-p}$.

In Assumptions 1–5, if $\alpha \leq \frac{1}{2\iota}$, we can get Eq. (11),

$$\mathbb{E}\left[\|x^k - x(\mu)\|^2\right] \leq \left(1 - \frac{\alpha\mu}{n}\right)^k \|x^0 - x(\mu)\|^2 + \frac{2n\alpha\sigma^2}{\mu} \tag{11}$$

where $\iota := \frac{1}{K}\max\left\{\frac{L}{1-p}, \frac{\mu\rho^2}{p}\right\}$ and $\sigma^2 := \frac{1}{K^2}\sum_{i=1}^{K}\left(\frac{1}{1-p}\|\nabla f_i(x_i(\mu))\|^2 + \frac{\mu^2\rho^4}{p}\|x_i(\mu) - x(\mu)\|^2\right)$. Parameters p, α can reach the neighborhood of the optimal solution $\left(O(\epsilon) + \frac{2n\alpha\sigma^2}{\mu}\right)$ at the fastest rate, satisfying:

$$\mathbb{E}\left[\|x^k - x(\mu)\|^2\right] \leq \epsilon\|x^0 - x(\mu)\|^2 + \frac{2n\alpha\sigma^2}{\mu} \tag{12}$$

Therefore, the upper bound in Eq. (8) is a convergent function, with the optimal model and the model at time t gradually converging. Meanwhile, the lower bound of the loss function is transformed into solving the optimal solution of FedAvg, whose convergence proof is listed in [52], suggesting that our loss function is convergent, with convergence efficiency lying between the two convergence functions.

4.6. Privacy considerations

Point cloud data contains privacy-sensitive information, as it encodes both semantic and geometric attributes of the environment. Our proposed FedPointNet mitigates such risks by adopting federated learning, where high-dimensional features and local loss values are transmitted to the server. The local loss is a scalar and does not reveal private information, while the high-dimensional features are extracted by the embedding layer and represent abstract data distributions. Considering that the inversion attacks may still reconstruct partial input information from such representations [53], secure aggregation techniques [54,55] can be employed to prevent the server from accessing individual client updates. Our approach is fully compatible with these methods, and the transmitted features can be protected by cryptographic techniques. The integration of such mechanisms into FedPillarNet will be further investigated in future work.

5. Experiment

In this section, extensive experiments are performed to verify the advantages of the proposed FedPillarNet. Firstly, the settings of dataset and the platform are introduced in detail. Secondly, experiments with existing methods are compared to verify the designed structure. Further, the effects of different personalized modules are studied to select the optimal personalization module. An ablation study is also performed to investigate the effect of the personalized module. Finally, experiments with key hyperparameters are performed to discuss the performance.

5.1. Dataset and platform

As the most popular datasets in autonomous driving, point clouds in the KITTI [45] and the nuScenes [46] collected in multiple scenarios are used in our experiments where cars, bus, pedestrians, and cyclists are the main target objects.

In our scenario, we set up 100 clients or 10 clients to verify the stability of the algorithm under different levels of client participation. For the non-iid problem caused by heterogeneous point clouds density, different levels of downsampling operations are performed on the original point clouds to simulate different laser beams of Lidar. For the non-iid problem caused by the heterogeneous label distributions, point clouds inside each client are selected from different types of traffic scenarios including downtown and highway with the corresponding data distributions. As a result, the point clouds in each client exhibit significant heterogeneity, thus being consistent with the actual autonomous driving. In addition, data augmentation including random flipping, random rotation, and random scaling are performed to help increase the diversity of the data, enabling the model to better adapt to more complex scenarios.

We use OpenPCDet [49] as the basic software platform. The hardware platform used in the experiment is the workstation with NVIDIA V100 GPU.

Following the public object detection evaluations, 2D bounding box accuracy (bbox), bird's-eye-view (bev) accuracy, 3D bounding box accuracy (3d), and average orientation similarity (aos) are used to measure the accuracy of detection results under the specific IoU (Intersection over Union).

5.2. Comparison with existing methods

We compare the detection performance with existing validated Fl framework. For the hyperparameter, the total training rounds is set as 30, where the each rounds contains 10 local training epochs. The batch size is set to 8, the Adam optimizer with an initial learning rate of 1×10^{-3}. The hyperparameter μ for Mmd loss is set as 0.1, λ for FedProx is set as 0.001, according to the related work. Results are listed in Table 2 where the first and second highest values among all methods are set to bold and underline, respectively.

Table 2 presents a comparison of results obtained from different methods across various metrics. Through a comparative analysis of experimental results, we clearly observe improvements achieved by our proposed method across various aspects, indicating substantial advantages over existing competing methods.

Enhancement in Accuracy and Robustness: Our method achieves higher accuracy across all metrics. For instance, compared to other methods, our method achieves notable increases in accuracy on *Client 2, 5, 8, 9* with all the metrics. This demonstrates our method's capability to more accurately capture the position and shape information of target objects, thereby ensuring reliable task accuracy. Furthermore, our method exhibits enhanced robustness. Despite variations in client characteristics and data distributions, our method maintains its superior performance consistently. Compared to other methods,

Table 2
Comparison with existing methods (mAP (%)).

Methods	Metric	Client1	Client2	Client3	Client4	Client5	Client6	Client7	Client8	Client9	Client10
PointPillar-Central [15]	bbox	78.65	65.83	85.86	76.73	69.64	**77.71**	**85.00**	50.42	67.11	**61.00**
	bev	90.30	84.83	**94.13**	88.26	83.72	**91.23**	91.99	67.80	83.04	**58.57**
	3d	88.86	80.56	**92.00**	85.86	80.08	87.56	89.45	61.57	80.68	**51.11**
	aos	78.17	65.28	85.66	76.09	69.02	**77.03**	**83.74**	50.02	66.60	**40.12**
FedAvg [26]	bbox	72.53	63.85	77.14	67.22	64.66	64.98	70.41	45.96	46.31	54.04
	bev	88.52	81.85	91.90	85.34	82.63	89.86	89.48	67.10	78.19	34.56
	3d	84.97	73.97	86.95	79.61	77.03	84.00	82.30	59.77	64.27	23.61
	aos	67.18	53.78	67.75	58.88	54.98	58.53	63.95	41.75	42.39	30.02
FedProx [28]	bbox	76.99	65.18	83.35	73.71	66.25	72.70	78.32	51.13	60.19	36.23
	bev	89.27	84.51	91.42	87.23	82.96	88.61	91.05	67.77	80.06	19.28
	3d	85.51	77.25	89.52	83.07	77.46	84.58	88.08	63.37	71.26	10.78
	aos	76.14	63.17	81.45	71.39	63.59	70.29	75.82	49.84	59.08	15.53
FedNova [56]	bbox	78.21	69.87	82.01	75.10	63.69	72.24	77.48	50.56	78.98	60.26
	bev	84.22	78.90	88.67	84.22	74.22	83.82	89.34	66.86	88.08	46.58
	3d	83.94	78.38	88.34	78.97	70.75	80.57	85.95	60.61	84.95	35.22
	aos	77.10	67.59	80.20	73.88	62.99	71.68	75.32	50.23	76.08	32.42
FedNorm [57]	bbox	66.74	59.62	70.49	61.63	42.48	61.55	64.94	45.35	71.51	48.24
	bev	82.56	75.62	92.29	83.63	67.17	81.54	89.58	57.48	88.13	26.51
	3d	78.81	71.96	84.44	74.77	57.89	74.12	80.94	51.31	80.69	14.60
	aos	63.87	55.03	65.77	57.34	41.69	59.11	60.10	42.01	67.44	22.68
FedMMD [17]	bbox	**80.74**	71.49	**87.14**	77.44	71.35	76.78	84.17	57.76	68.94	51.00
	bev	**91.65**	**87.47**	93.72	**88.93**	**85.42**	91.08	92.19	72.84	80.45	16.38
	3d	**89.22**	82.03	91.61	**86.27**	82.25	88.28	89.76	67.16	74.04	9.08
	aos	**79.60**	70.46	**86.38**	76.36	69.72	75.78	82.70	57.31	67.40	26.81
Ours	bbox	80.45	**75.94**	86.74	**79.26**	**74.77**	75.47	82.36	**67.97**	**80.74**	55.42
	bev	89.78	86.71	93.94	85.70	84.85	90.26	**92.20**	**79.23**	**89.31**	52.28
	3d	88.85	**82.73**	91.59	84.66	**82.38**	**88.32**	**91.00**	**74.88**	**87.70**	43.12
	aos	79.55	**75.10**	86.22	**78.54**	**73.49**	72.71	82.21	**65.62**	**78.77**	35.09

our method demonstrates less fluctuation in performance across different clients and metrics. For example, the performance disparity of our method across different clients is only around 0.2% to 1.3% for bbox, whereas FedAvg exhibits larger disparities, ranging from 5.5% to 10.3%. This improved robustness enhances the reliability and stability of our method in practical applications, providing significant support for generalization performance in real-world scenarios.

Further, the training curve of KITTI is displayed in Fig. 4(a) where the accuracy of bbox is chosen as the evaluation metric, which follows the KITTI benchmark.

It can be observed that our method converges equally rapidly with the central method. In terms of accuracy performance, our method achieves better than other methods. Although our method is unstable in the initial state due to the extraction of personalized features from different clients, it does not affect the server model's improvements, demonstrating that our model converges as proofed in the aforementioned section.

For Nuscene, the accuracy curves are displayed in Fig. 4(b). The data show that our method achieves faster convergence with high accuracy.

Personalized Adaptation Capability: The personalized adaptation mechanism in our method enables the model to better adapt to the data characteristics of specific clients. For instance, comparing Ours-Without Personalized and Ours-With Personalized, in some clients, Ours-With Personalized slightly outperforms Ours-Without Personalized, such as on *Client5* where bev metric increases from 91.08 to 92.20.

Considering the aforementioned aspects comprehensively, our method exhibits superiority in accuracy, robustness, and personalized adaptation capability. This comprehensive advantage enables our method to possess clear superiority in addressing the current task, providing a reliable and efficient solution for practical applications.

Faster Convergence Speed: As illustrated in Figs. 4(a) and 4(c), our method exhibits rapid convergence across various client participation scenarios. When the federated learning process involves 100 clients, the accuracy of the federated model is observed to decrease, which is primarily attributed to the reduced average participation rate of clients. Our proposed method continues to demonstrate superior performance, maintaining a higher level of accuracy and robustness compared to other approaches under similar conditions. Specifically, it still achieves a 5.8% accuracy improvement and a 48% faster convergence rate compared to other methods. These enhancements are particularly evident in scenarios where client participation is sporadic or inconsistent, which highlights the adaptability and resilience of our method.

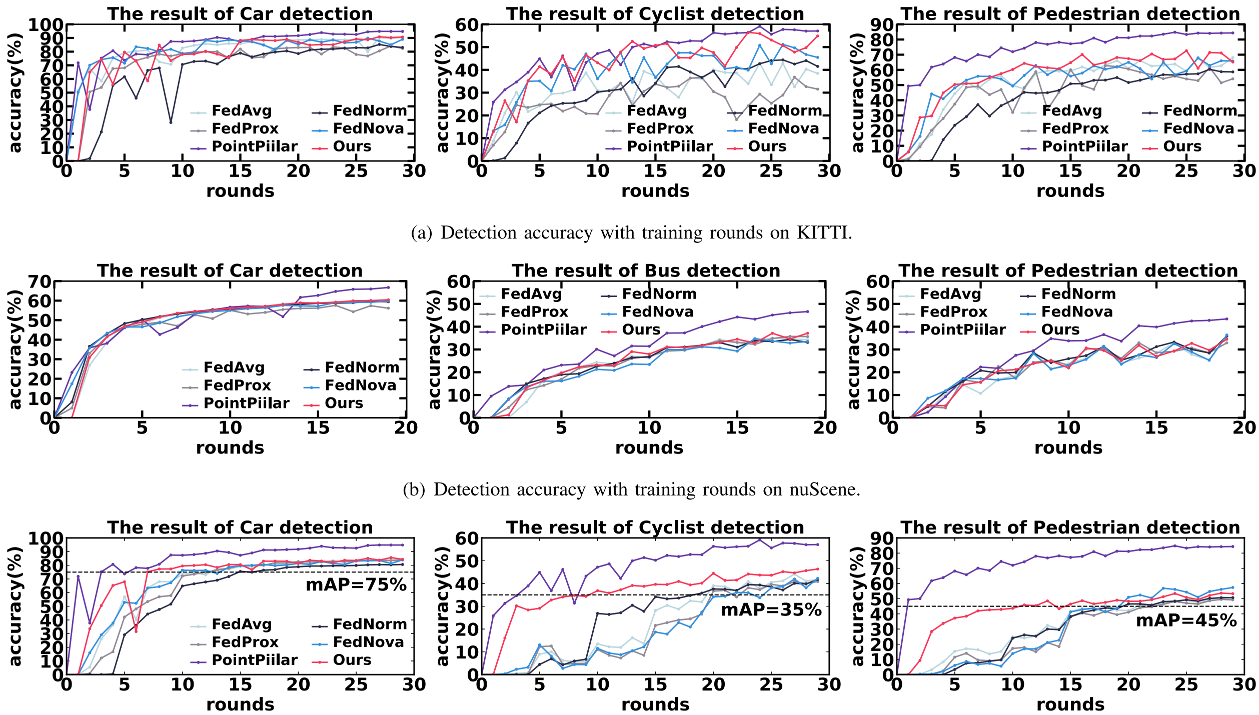
Table 3
Communication overhead during training.

Method	Average communication overhead
FedAvg [26]	418.06 MB
FedProx [28]	401.66 MB
FedNova [56]	426.25 MB
FedNorm [57]	418.06 MB
FedMMD [17]	401.66 MB
Ours	204.93 MB

In addition, the average communication overhead is also saved by reducing the aggregation rounds during training. As shown in Table 3, our method achieves most reductions in communication volume due to its feature-level update strategy and fast convergence. These results validate the practicality of our framework for resource-constrained scenarios.

Cross-domain Experiment: To verify the effectiveness of the model in different conditions, we designed two cross-domain experiments on KITTI [45] and SnowyKIITI [58], which is collected in snowy weather and has a strong domain shift. The results displayed in Table 4 indicate that our client-side feature extraction mechanism effectively identifies object features. Despite the interference caused by data shifts, our method still outperforms other methods. This proves the effectiveness and superiority of our model in handling cross domain issues.

In summary, compared with other existing federated methods, our method achieves higher accuracy on both KITTI and nuScene dataset, exhibiting high robustness in the target detection task based on 3D point clouds.



(a) Detection accuracy with training rounds on KITTI.

(b) Detection accuracy with training rounds on nuScene.

(c) Detection accuracy with training rounds on KITTI (100 clients).

Fig. 4. Detection accuracy with training rounds.

Table 4
Cross-domain comparison experiments of different methods (mAP (%)).

Setting	Methods	Car	Cyclist	Pedestrian
Train-KITTI Test-SnowyKITTI	FedAvg	88.72	61.26	53.03
	FedMMD	88.83	61.93	54.68
	Ours	**88.87**	**62.54**	**55.08**
Train-SnowyKITTI Test-KITTI	FedAvg	89.67	70.19	52.57
	FedMMD	89.58	72.65	**55.33**
	Ours	**89.78**	**74.05**	54.89

5.3. Effects of personalized module

To delve into the workings of the personalized module, a series of experiments are conducted utilizing FedPillarNet with and without the personalized feature extractor. The objective is to investigate the manner in which the personalized module enhances model performance. The outcomes of these experiments are displayed in Fig. 5.

Owing to variations in data distribution among clients, the personalized features across different clients exhibit heterogeneity, thereby posing challenges in evaluating the efficacy of the personalized module under uniform conditions. Consequently, we adopt a strategy of distinguishing between detection objects of different clients based on the distribution characteristics of the dataset.

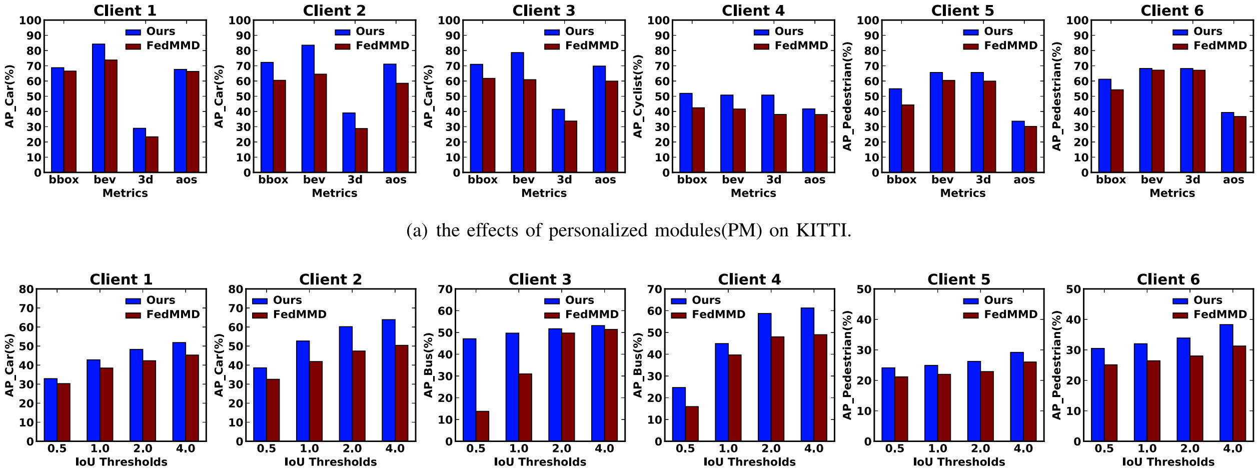
In typical federated scenarios, clients engage in the training process in a stochastic manner, experiencing discrepancies in the number of training rounds. Consequently, the ability to swiftly converge amidst shorter and unpredictable training instances holds paramount significance. To evaluate the robustness of the personalized module, experiments are conducted with clients joining the training process randomly.

For KITTI, *Client 1*, *Client 2*, and *Client 3* contains a huge amount of cars within their datasets. Consequently, the models from these clients were utilized to assess car detection. The primary object labels in *Client 4* pertain to cyclists, their model are employed to evaluate cyclist detection. In *Client 5* and *Client 6*, a majority of pedestrians are encompassed within the datasets. Therefore, it is utilized to evaluate pedestrian detection. For nuScene dataset, *Client 3* and *Client 4* contain a large number of buses. The experiments focus on evaluating the detection on bus. All the other hyperparameters remained consistent with those in Section 5.2.In Fig. 5, each client is represented by two clusters of detection results. The left cluster corresponds to results obtained using FedPillarNet with the personalized module, while the right cluster corresponds to results generated without the inclusion of the personalized module. Analysis of Fig. 5 reveals that the personalized feature extractor enhances the feature characterization capability of the client model.

Specifically, the majority of clients employing the personalized module exhibit significant performance enhancements. Notably, as displayed in Fig. 5(a), with the inclusion of the personalized module, *Client 3*'s model demonstrates a remarkable 35% increase in accuracy for detecting Car labels in the 3D metric, while *Client 4*'s model achieves a 33% accuracy improvement in detecting Cyclist labels. Furthermore, for the model of *Client 6*, there is a notable accuracy enhancement of more than 1.5% across all metrics pertaining to Pedestrian label detection. For nuScene, Fig. 5(b) exhibits an improvement in the model detection ability of the personalized module for different IoU thresholds. For the case of center distance threshold 0.5, a 10% accuracy improvement is achieved for all objects. And for the case of center distance threshold 4.0, the detection accuracy is improved by 17.4% on average.

The explanations for the aforementioned experimental phenomena are delineated as follows.

Firstly, the personalized modules play a pivotal role in enhancing feature extraction by the client model, particularly when datasets are limited in size. This configuration, featuring personalized feature extractors, confers a distinct advantage, primarily attributed to its capacity to explore multiple datasets for weight parameters within the model's solution space. Secondly, The utilization of attention mechanisms as personalized feature extractors offers significant advantages by enabling focused attention on pertinent areas within the 2D pseudo point cloud images. This targeted focus contributes to an improvement



(a) the effects of personalized modules(PM) on KITTI.

(b) the effects of personalized modules(PM) on nuScene.

Fig. 5. Effects of personalized module.

Table 5
Comparison of personalized modules through K-fold cross validation (mAP (%)).

Client	Object	Metrics	CBAM	CAM	SAM	NON
client1	Car	bbox	79.63	**80.34**	77.42	80.15
		bev	**90.26**	90.13	90.08	90.09
		3d	**88.18**	87.72	87.45	87.76
		aos	78.86	**79.54**	76.68	79.49
client2	Car	bbox	**88.53**	84.10	85.83	82.32
		bev	**94.62**	93.70	93.80	93.69
		3d	**92.64**	91.25	91.52	91.00
		aos	**87.71**	83.42	84.86	81.88
client3	Cyc.	bbox	28.33	30.27	**36.56**	29.53
		bev	27.51	29.77	**34.84**	27.14
		3d	27.51	29.77	**34.84**	27.00
		aos	25.57	26.54	**34.55**	26.49
client4	Cyc.	bbox	36.79	37.89	**40.50**	36.37
		bev	34.39	36.88	**39.55**	34.84
		3d	34.39	36.86	**39.55**	34.80
		aos	33.44	35.59	**38.02**	32.37
client5	Ped.	bbox	46.19	**47.33**	45.49	45.79
		bev	56.73	**58.09**	57.73	55.40
		3d	56.34	**57.53**	57.14	55.40
		aos	**32.26**	31.71	31.31	32.14
client6	Ped.	bbox	48.28	**58.19**	57.73	43.64
		bev	56.39	**69.61**	68.12	52.07
		3d	56.39	**69.56**	68.12	52.05
		aos	28.48	**38.44**	37.16	27.77

in model accuracy by effectively highlighting relevant features. Finally, the parameters of the personalized module remain unaffected by global model aggregation processes. This ensures the preservation of the module's ability to extract localized features, thereby enhancing the overall robustness and adaptability of the model.

To make conclusion, adding the personalized feature extractor can effectively improve the classification accuracy of heterogeneous data. Compared with conventional centralized training model, FedPillarNet has more robust generalization ability in the FL scenarios.

5.4. Ablation study

In order to further investigate the overall impact of different personalized modules on the model, ablution studies targeting personalized modules are conducted based on the above experiments. Different types of attention modules including SAM, CAM, CBAM are compared in this section. The detailed experimental results are listed in Table 5.

In this section, we delve into the various detection results caused by different personalized modules. Specifically, 6 clients are used to participate in each round of training, aiming to verify the effectiveness of the personalized modules. Notably, the most salient labeled features in each client are selected for assessment in the experiments. Furthermore, the targets within the dataset are intentionally chosen to scrutinize the model's generalization prowess. For each client, the local test and the training set have the similar data distributions, thus emulating real-world conditions where the model is globally trained and utilized locally.

K-fold cross-validation evaluates the stability of the performance of different modules on different subsets of data by minimizing the bias introduced by the dataset. We use K-fold cross validation to improve the assessment accuracy and measure the sensitivity of modules to the data changes.

As displayed in Table 5, the most notable values are highlighted, indicating the performance superiority of the method. Clients integrated with a personalized module demonstrate enhanced adaptive and generalization capabilities, consistently outperforming models that lack customization across a range of targets and varying levels of detection complexity. This highlights the effectiveness of the personalized module in improving overall detection performance.

Comparing with different personalized modules, SAM outperforms the other modules in cyclist detection, the most challenging task, up to nearly 10% improvement in specific metrics compared to the 2nd best. Meanwhile, SAM also performs well in all other categories. SAM stands out for its compactness and adeptness at swift convergence, owing to its nature as an insertion module. Moreover, functioning as a spatial attention module, SAM seamlessly aligns with the three-dimensional structure of point clouds, effectively directing attention towards pivotal regions within the pseudo-map, which makes SAM more adaptive with obvious spatial structure characteristics such as Cyclist.

CAM focuses on the interrelationships between different channels and emphasizes the importance of feature channels. For pedestrians, pose and appearance vary greatly across environments and viewpoints. CAM improves the performance by emphasizing channel features related to the pose and appearance. In contrast, CBAM has a more complex structure, which is benefit to capture both features in local area across channels. For cars, due to the complexity of the shapes, textures, and colors, CBAM focuses on both spatial details and channel variations to gain an advantage in the results of *Client 1* and *Client 2*.

There are two primary reasons why SAM surpasses other modules to such a remarkable degree. Firstly, its inherent simplicity and effortless adaptability significantly bolster its efficacy. Secondly, serving as a

Table 6
Comparisons of μ under different data heterogeneity with mAP (%).

	$\mu = 0.01$	$\mu = 0.1$	$\mu = 0.5$	$\mu = 1$	$\mu = 10$
Feature Skew	63.34	**66.87**	64.16	65.41	65.70
Quantity Skew	67.78	**69.10**	65.47	68.60	68.99
Label Skew	65.09	60.14	**67.36**	63.47	62.43
Mixed Skew	65.48	65.89	**66.36**	66.33	64.62

spatial attention module, SAM harmoniously meshes with the intricate three-dimensional architecture characteristic of the point clouds. This innate compatibility empowers SAM to adeptly pinpoint crucial regions within the compressed pseudo-map of the point clouds, thereby facilitating precise and targeted attention allocation precisely where it is most crucial.

In conclusion, the experiments indicates that the superior stability and adaptability of the SAM in meeting the demands of the personalized module. Additionally, selecting specific personalized modules for different categories of data will make the personalized modules work more effectively.

5.5. Hyperparameter experiments

In this section, the effects of hyperparameters are discussed. Considering the facts that factors including the parameter μ, the number of training epochs, training rounds, and the percentage of client participation C significantly influence the results, we conduct experiments as illustrated in Table 6 and Fig. 6 to explore their impacts.

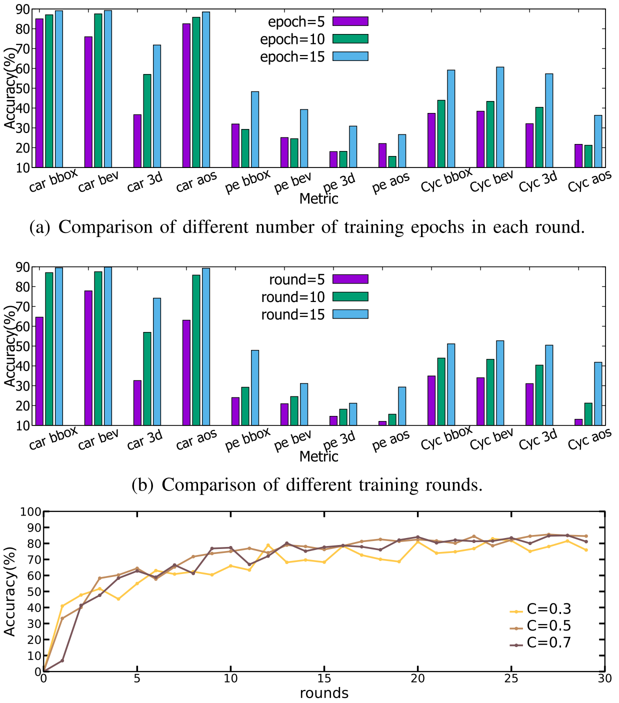
The hyperparameter μ primarily regulates the extent to which global features are aggregated by the MMD mechanism. A larger μ encourages stronger alignment across clients, thereby mitigating the effect of data heterogeneity. However, overly aggressive global feature aggregation may hinder the local model's ability to capture client-specific patterns, which can negatively affect convergence.

In the context of point clouds, heterogeneity can arise from three sources: feature skew, label skew, and quantity skew. Feature skew refers to the phenomenon where the same objects exhibit different features. To generate feature skew, we construct clients in different weather conditions and simulate point clouds generated by different LIDAR. Under label skew, the proportions of different objects in client datasets vary. We create label skew by sampling in different scenes. Quantity skew will cause differences in the convergence speed local of models. We construct quantity skew by partitioning the dataset according to the Dirichlet distribution.

In practice, these often appear simultaneously, forming mixed heterogeneity. To evaluate the impact of μ under realistic conditions, we conduct a comprehensive sensitivity analysis across all four types of heterogeneity. The final model performance under varying μ settings is reported in Table 6.

For feature skew and quantity skew scenarios, the best model performance is achieved when μ is set to 0.1. In contrast, for mixed skew and label skew scenarios, the optimal model performance is observed when μ is 0.5. As μ serves as a balancing factor between local and global models, values that are too high or too low can cause the model to favor one side, thereby reducing the accuracy of the global model. Given that feature skew and quantity skew have a relatively impact on the global model, employing a smaller μ value is sufficient to counteract the effects of data heterogeneity.

In summary, the parameter μ plays a crucial role in providing a balance between global features and local features within the clients' local model. When the distribution of the data is not well understood, it is advisable to choose a moderately sized value for μ. Such a choice helps the model maintain robustness when facing different types of data heterogeneity and avoids performance degradation due to improper parameter settings. A moderate μ value ensures that the model achieves the best balance when integrating local and global information, thereby improving the overall performance and generalization ability of the model.



(a) Comparison of different number of training epochs in each round.

(b) Comparison of different training rounds.

(c) Comparison of different levels of client participation

Fig. 6. Comparison with different hyperparameters.

Fig. 6(a) illustrates the outcomes obtained from conducting 10 rounds of training process while altering the epoch count for individual client training within each round. It is evident from the figure that increasing the number of epochs enhances the model's accuracy. However, in the case of pedestrian target metrics, training 10 times per round proves to be less effective than training 5 times per round. We hypothesize that this discrepancy may stem from the presence of noise within the dataset, leading to fluctuations and oscillations in the training process of the local model. By augmenting the number of client training epochs, the local model gains additional opportunities to adapt to such noise or outliers, thereby bolstering the overall performance of the global model.

Fig. 6(b) displays the results achieved by increasing the number of training rounds. Clearly, augmenting the number of rounds leads to a better fit of the model to global features and consequently improves the results. This observation indicates the importance of iterative refinement through multiple training rounds in enhancing the model's capability to capture and incorporate global features effectively. The incremental refinement across multiple rounds enables the model to obtain more comprehensive insights from the data, ultimately leading to improved performance across various metrics.

Fig. 6(c) shows the results obtained by increasing the number of client participation per round. The hyperparameter C in the experiment represents the percentage of client devices participating. As the number of iterations increases, the model of all experiments tends to converge. The increase in the percentage of client participation leads to an upward trend in the accuracy of the model. The model is able to learn from more data, which improves its ability to understand and predict data. Within a certain number of clients, increasing the number of clients can effectively improve the performance of the model. However, beyond this range, a larger percentage of client participation has a limited effect on model performance improvement, and additionally, there may be other factors including the communication costs, diversity of data distribution, which negatively affect model performance.

6. Conclusion

In this work, a FL based framework for heterogeneous LIDAR point cloud called the FedPillarNet is proposed to train the object detection model. To solve the non-iid problem caused by the point clouds from multiple vehicles, a personalized model is trained to meet the local features while a global model is collaboratively trained based on FL using the MMD as the part of the loss function for balancing the local and global loss. Extensive experiments demonstrate that the proposed FedPillarNet outperforms the related methods. In real-world federated settings, malicious clients may launch adversarial attacks or manipulate updates to degrade performance. Addressing such threats requires incorporating robust aggregation mechanisms or adversarial detection techniques will be explored in future work.

CRediT authorship contribution statement

Boyang Li: Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Siheng Ren:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Shuai Zhao:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization. **Mingyue Cui:** Writing – review & editing, Funding acquisition. **Kai Huang:** Writing – review & editing, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, Deep learning for 3d point clouds: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2020) 4338–4364.

[2] P. Wang, T. Ouyang, Q. Wu, Q. Huang, J. Gong, X. Chen, Hydra: Hybrid-model federated learning for human activity recognition on heterogeneous devices, J. Syst. Archit. 147 (2024) 103052, http://dx.doi.org/10.1016/J.SYSARC.2023.103052.

[3] J. Yin, J. Shen, X. Gao, D.J. Crandall, R. Yang, Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds, IEEE Trans. Pattern Anal. Mach. Intell. 45 (8) (2023) 9822–9835.

[4] J. Shi, Y. Wu, D. Zeng, J. Tao, J. Hu, Y. Shi, Self-supervised on-device federated learning from unlabeled streams, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 42 (12) (2023) 4871–4882.

[5] Y. Guo, F. Liu, T. Zhou, Z. Cai, N. Xiao, Privacy vs. efficiency: Achieving both through adaptive hierarchical federated learning, IEEE Trans. Parallel Distrib. Syst. 34 (4) (2023) 1331–1342.

[6] S. Zhang, J. Li, L. Shi, M. Ding, D.C. Nguyen, W. Tan, J. Weng, Z. Han, Federated learning in intelligent transportation systems: Recent applications and open problems, 2023, arXiv Preprint.

[7] S. Liu, H. Jin, Z. Tang, R. Zhai, K. Lu, J. Yu, C. Bai, Adapter-guided knowledge transfer for heterogeneous federated learning, J. Syst. Archit. 159 (2025) 103338, http://dx.doi.org/10.1016/J.SYSARC.2025.103338.

[8] R. Wang, L. Xiong, J. Geng, C. Xie, R. Li, An effective and verifiable secure aggregation scheme with privacy-preserving for federated learning, J. Syst. Archit. 161 (2025) 103364, http://dx.doi.org/10.1016/J.SYSARC.2025.103364.

[9] Y. Liu, Z. Jia, Z. Jiang, X. Lin, J. Liu, Q. Wu, W. Susilo, BFL-SA: blockchain-based federated learning via enhanced secure aggregation, J. Syst. Archit. 152 (2024) 103163, http://dx.doi.org/10.1016/J.SYSARC.2024.103163.

[10] Z. Wang, Z. Li, M. Fu, Y. Ye, P. Wang, Network traffic classification based on federated semi-supervised learning, J. Syst. Archit. 149 (2024) 103091, http://dx.doi.org/10.1016/J.SYSARC.2024.103091.

[11] M. Guan, H. Bao, Z. Li, H. Pan, C. Huang, H. Dai, SAMFL: secure aggregation mechanism for federated learning with Byzantine-robustness by functional encryption, J. Syst. Archit. 157 (2024) 103304, http://dx.doi.org/10.1016/J.SYSARC.2024.103304.

[12] S. Vahidian, M. Morafah, B. Lin, Personalized federated learning by structured and unstructured pruning under data heterogeneity, in: IEEE 41st International Conference on Distributed Computing Systems Workshops, 2021, pp. 27–34.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[14] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proc. Mach. Learn. Syst. 2 (2020) 429–450.

[15] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom, Pointpillars: Fast encoders for object detection from point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12697–12705.

[16] H. Yan, Z. Li, Q. Wang, P. Li, Y. Xu, W. Zuo, Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation, IEEE Trans. Multimed. 22 (9) (2020) 2420–2433.

[17] Y. Zhou, B. Li, K. Huang, An object detection method based on heterogeneous lidar point clouds using federated learning, in: International Conference on Autonomous Unmanned Systems, Springer, 2023, pp. 248–259.

[18] S. Shi, X. Wang, H. Li, Pointrcnn: 3d object proposal generation and detection from point cloud, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 770–779.

[19] C.R. Qi, X. Chen, O. Litany, L.J. Guibas, Imvotenet: Boosting 3d object detection in point clouds with image votes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4404–4413.

[20] A. Xue, X. Zhu, J. Wang, J. Xiao, Diversified point cloud classification using personalized federated learning, in: International Joint Conference on Neural Networks, IEEE, 2021, pp. 1–8.

[21] L. Chen, X. Fan, H. Jin, X. Sun, M. Cheng, C. Wang, FedRME: Federated road markings extraction from mobile LiDAR point clouds, in: IEEE 25th International Conference on Computer Supported Cooperative Work in Design, 2022, pp. 653–658.

[22] S.P. Karimireddy, S. Kale, M. Mohri, S.J. Reddi, S. Stich, A. Suresh, SCAFFOLD: Stochastic controlled averaging for federated learning, 2019.

[23] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, Found. Trends® Mach. Learn. 14 (1–2) (2021) 1–210.

[24] R. Chandrasekaran, K. Ergun, J. Lee, D. Nanjunda, J. Kang, T. Rosing, Fhdnn: Communication efficient and robust federated learning for aiot networks, in: Proceedings of the 59th ACM/IEEE Design Automation Conference, 2022, pp. 37–42.

[25] Z. Xu, F. Yu, J. Xiong, X. Chen, Helios: Heterogeneity-aware federated learning with dynamically balanced collaboration, in: 58th ACM/IEEE Design Automation Conference, 2021, pp. 997–1002.

[26] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, S.-L. Kim, Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data, 2018.

[27] H. Wang, Z. Kaplan, D. Niu, B. Li, Optimizing Federated Learning on Non-IID Data with Reinforcement Learning, IEEE, 2020.

[28] A.K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, 2018.

[29] J. Zhu, X. Ma, M.B. Blaschko, Confidence-aware personalized federated learning via variational expectation maximization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24542–24551.

[30] Z. Li, T. Lin, X. Shang, C. Wu, Revisiting weighted aggregation in federated learning with neural networks, in: International Conference on Machine Learning, PMLR, 2023, pp. 19767–19788.

[31] T. Zheng, A. Li, Z. Chen, H. Wang, J. Luo, AutoFed: Heterogeneity-aware federated multimodal learning for robust autonomous driving, in: Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, 2023, pp. 1–15.

[32] D. Bui, K. Malik, J. Goetz, H. Liu, S. Moon, A. Kumar, K. Shin, Federated user representation learning, 2019, arXiv.Org.

[33] C. T. Dinh, N. Tran, J. Nguyen, Personalized federated learning with moreau envelopes, Adv. Neural Inf. Process. Syst. 33 (2020) 21394–21405.

[34] X.-C. Li, D.-C. Zhan, Y. Shao, B. Li, S. Song, Fedphp: Federated personalization with inherited private models, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 587–602.

[35] M.G. Arivazhagan, V. Aggarwal, A.K. Singh, S. Choudhary, Federated learning with personalization layers, 2019, arXiv preprint arXiv:1912.00818.

[36] L. Collins, H. Hassani, A. Mokhtari, S. Shakkottai, Exploiting shared representations for personalized federated learning, in: International Conference on Machine Learning, PMLR, 2021, pp. 2089–2099.

[37] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, B. Han, FedFed: Feature distillation against data heterogeneity in federated learning, Adv. Neural Inf. Process. Syst. 36 (2024).

[38] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, H. Guan, Fedcp: Separating feature information for personalized federated learning via conditional policy, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3249–3261.

[39] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, R.M. Parizi, A robust privacy-preserving federated learning model against model poisoning attacks, IEEE Trans. Inf. Forensics Secur. (2024).

[40] A. Yazdinejad, A. Dehghantanha, G. Srivastava, H. Karimipour, R.M. Parizi, Hybrid privacy preserving federated learning against irregular users in next-generation internet of things, J. Syst. Archit. 148 (2024) 103088.

[41] A. Yazdinejad, A. Dehghantanha, G. Srivastava, AP2FL: Auditable privacy-preserving federated learning framework for electronics in healthcare, IEEE Trans. Consum. Electron. 70 (1) (2023) 2527–2535.

[42] Z. Wang, H. Xu, Y. Xu, Z. Jiang, J. Liu, S. Chen, FAST: enhancing federated learning through adaptive data sampling and local training, IEEE Trans. Parallel Distrib. Syst. 35 (2) (2024) 221–236.

[43] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, H. Guan, FedCP: Separating feature information for personalized federated learning via conditional policy, in: A.K. Singh, Y. Sun, L. Akoglu, D. Gunopulos, X. Yan, R. Kumar, F. Ozcan, J. Ye (Eds.), Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023, ACM, 2023, pp. 3249–3261.

[44] J. Lambert, A. Carballo, A.M. Cano, P. Narksri, D. Wong, E. Takeuchi, K. Takeda, Performance analysis of 10 models of 3D LiDARs for automated driving, IEEE Access 8 (2020) 131699–131722.

[45] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.

[46] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, Nuscenes: A multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11621–11631.

[47] L. Jie, Z. Jin, J. Wang, L. Zhang, X. Tan, A SLAM system with direct velocity estimation for mechanical and solid-state LiDARs, Remote. Sens. 14 (7) (2022) 1741.

[48] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, 2018.

[49] O.D. Team, OpenPCDet: An open-source toolbox for 3D object detection from point clouds, 2020.

[50] D. Bui, K. Malik, J. Goetz, H. Liu, S. Moon, A. Kumar, K.G. Shin, Federated user representation learning, 2019, arXiv preprint arXiv:1909.12535.

[51] H. Filip, R. Peter, Federated learning of a mixture of global and local models, 2020, arXiv preprint arXiv:2002.05516.

[52] X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of fedavg on non-iid data, 2019, arXiv preprint arXiv:1907.02189.

[53] J. Geiping, H. Bauermeister, H. Dröge, M. Moeller, Inverting gradients-how easy is it to break privacy in federated learning? Adv. Neural Inf. Process. Syst. 33 (2020) 16937–16947.

[54] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191.

[55] W.-N. Chen, C.A. Choquette-Choo, P. Kairouz, Communication efficient federated learning with secure aggregation and differential privacy, in: NeurIPS 2021 Workshop Privacy in Machine Learning, 2021.

[56] J. Wang, Q. Liu, H. Liang, G. Joshi, H.V. Poor, Tackling the objective inconsistency problem in heterogeneous federated optimization, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual, 2020.

[57] Z. Lian, W. Liu, J. Cao, Z. Zhu, X. Zhou, FedNorm: An efficient federated learning framework with dual heterogeneity coexistence on edge intelligence systems, in: IEEE 40th International Conference on Computer Design, ICCD 2022, Olympic Valley, CA, USA, October 23-26, 2022, IEEE, 2022, pp. 619–626.

[58] A. Seppänen, R. Ojala, K. Tammi, 4Denoisenet: Adverse weather denoising from adjacent point clouds, IEEE Robot. Autom. Lett. 8 (1) (2022) 456–463.

Boyang Li received the B.Sc. degree, the M.Sc. degree in software engineering and Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2016, 2018, and 2023 respectively. He is now a postdoc at Sun Yat-sen University, majoring in robots and autonomous systems.

Siheng Ren received the B.Sc. degree in Traffic Engineering from Sun Yat-sen University, ShenZhen, China, in 2024. He is currently pursuing his M.Sc. degree in computer science with a primary research focus on the application of federated learning to autonomous driving.

Shuai Zhao received the Ph.D. degree in computer science from the University of York, York, U.K., in 2018. He is an Associate Professor with Sun Yat-sen University, Guangzhou, China. His research interests include scheduling algorithms, multiprocessor resource sharing, schedulability analysis, and safety-critical programming languages.

Mingyue Cui received an M.Sc. degree in Software Engineering, and a Ph.D. degree in Computer Science from Sun Yat-sen University, Guangzhou, Guangdong, China, in 2017, and 2022, respectively.He was a Postdoctoral Fellow with the School of Computer Science and Engineering at Sun Yat-Sen University, China, from 2023 to 2025.He is currently working as an Associate Researcher with the School of Computer Science and Engineering at Sun Yat-Sen University, China. His research interests are in the area of 3D vision and robotics.

Kai Huang joined Sun Yat-sen University as a Professor in 2015. He was appointed as the director of the Institute of Artificial Intelligence and Unmanned Systems of School of Computer Science and Engineering in 2016. He was a senior researcher in the Computer Science Department, the Technical University of Munich, Germany from 2012 to 2015 and a research group leader in fortiss GmbH in Munich Germany in 2011. He obtained his Ph.D. degree in ETH Zurich, Switzerland in 2010. MSc from the University of Leiden, the Netherlands in 2005, and BSc from Fudan University, China in 1999. His research interests include techniques for the analysis, design, and optimization of embedded/CPS systems, particularly in the automotive, medical, and robot domains.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.