TeamFed: Teamwork Principles-Inspired Federated Learning for 3D Object Detection

Siheng Ren[1[0009-0001-4124-7048]], Boyang Li[1[0000-0003-1747-8011]], Shuai Liu[1[0000-0001-5002-4754]], Jiahui Liao[1[0009-0009-0261-9599]], Mingyue Cui[1[0000-0001-7094-2843]], and Kai Huang[1[0000-0003-0359-7810]]

School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
{rensh6,liush376,liaojh27}@mail2.sysu.edu.cn,
{liby83,cuimy5,huangk36}@mail.sysu.edu.cn

Abstract. 3D object detection based on LiDAR point clouds has been widely explored in autonomous driving. Federated Learning (FL), with its inherent advantages in distributed privacy protection and scalable multi-vehicle collaboration, presents a promising alternative to centralized training. However, its real-world deployment encounters the major challenges: (1) severe LiDAR data heterogeneity across vehicles hinders global model convergence; (2) inherent conflicts between global model consistency and local personalization. To address the challenges, we propose TeamFed, a FL framework for 3D object detection inspired by organizational principles in human teams. By incorporating a multi-leader collaboration paradigm and a personalized sampling strategy, TeamFed enables efficient global model training and effective local personalization tuning. Extensive experiments demonstrate that TeamFed achieves 15.8% and 22.1% performance improvements in global and personalized models respectively under heterogeneous autonomous driving scenarios, approaching or even surpassing the performance of centralized training.

Keywords: 3D Object Detection · Federated Learning · Autonomous Driving.

1 Introduction

Object detection using 3D point clouds has become a fundamental perception task in autonomous driving, enabled by LiDAR's ability to provide accurate geometric representations of the environment. However, the high transmission and storage costs associated with LiDAR data impose significant resource burdens during the training process [15]. With the continuous expansion of autonomous driving scenarios, recent advances in on-board computing, such as NVIDIA DRIVE Thor achieving 2000 TOPS, have made distributed training on vehicle-mounted platforms increasingly feasible.

Compared to centralized training, distributed training reduces the cost of data transmission while improving system robustness and fault tolerance. Within this paradigm, federated learning (FL) has emerged as a promising solution to preserve collaborative learning privacy [3]. By integrating FL into 3D point cloud detection tasks, it is possible to ensure data security while effectively utilizing the computing resources available on individual vehicles [30]. In addition, FL supports model personalization, enabling adaptation to various user and enterprise requirements in real-world autonomous driving scenarios [16].

Despite the extensive research since its initial introduction by Google in [11], applying FL to autonomous driving remains challenging due to data heterogeneity, resulting from variations in sensor types, scenario distributions, and environmental dynamics. This heterogeneity typically results in non-independent and identically distributed (non-IID) data across clients, in terms of features, sample sizes, and label distributions[20,25]. Such heterogeneity often causes model drift and degrades global model convergence after aggregation [5]. Existing solutions mitigate this problem by imposing constraints on local updates to align with the global objective [7,1], but these approaches tend to suppress useful local information, limiting their ability to support task-level personalization. Clustering-based methods [13,4] offer an alternative by grouping clients based on feature or data similarities, yet they are highly sensitive to clustering granularity and require careful hyperparameter tuning.

Another challenge in FL lies in the inherent tension between global model convergence and client-specific personalization [26]. Personalization aims to dynamically adapt models to local data distributions while maintaining discriminative feature representations during iterative global aggregation. In contrast, global optimization prioritizes cross-client generalization, requiring coordinated parameter updates to achieve Pareto efficiency throughout the system. To reconcile this conflict, some works adopt model decoupling architectures [16], embedding explicit personalization modules. Others use meta-learning and transfer learning techniques [2,10] for rapid adaptation to new scenarios. These methods are primarily suited for single-label classification tasks, limiting their applicability to multi-label scenarios in autonomous driving.

To tackle the challenges of data heterogeneity and the conflict between global convergence and local personalization, we propose TeamFed, an efficient collaborative FL framework. Unlike conventional FL methods that treat all clients uniformly, TeamFed conceptualizes the learning process as a role-based collaboration, where clients contribute unequally based on their capabilities and data relevance. Inspired by principles of human team organization, TeamFed introduces a multi-leader collaboration scheme and a personalized resource allocation strategy. At the global level, a policy network is used to select the leading clients and dynamically cluster similar clients to guide the aggregation process. At the local level, we leverage a multi-armed bandit (MAB) algorithm to adaptively sample training data, ensuring the effective utilization of both global knowledge and local characteristics. Theoretical analysis and empirical evaluations validate that TeamFed achieves faster convergence of the global model and efficient local personalization. To sum up, the key contributions are listed below.

- A multi-leader collaboration paradigm is proposed to mitigate the model drift caused by data heterogeneity.

– A personalized sampling mechanism is proposed to accelerate global model convergence while enhancing local model personalization.
– Extensive experiments are performed to evaluate the performance of TeamFed.

2 Related work

2.1 Addressing Data Heterogeneity in Federated Learning

Autonomous driving vehicles (ADVs) in diverse environments employ heterogeneous sensors, resulting in non-IID data by class imbalance, feature distribution shifts, and quantity skewness. These forms of heterogeneity often coexist in real-world scenarios. As illustrated in Fig. 1, significant variations in point cloud characteristics are observed across clients.

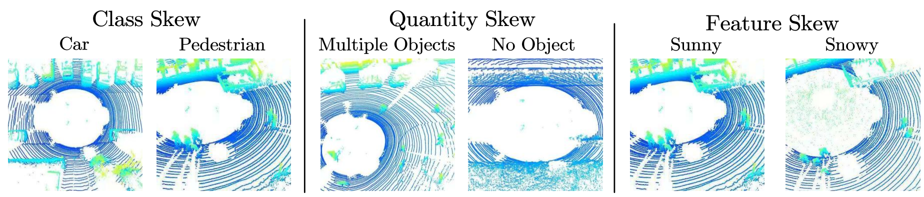

Fig. 1: Data heterogeneity in autonomous driving is mainly reflected in three dimensions: Class Skew, Quantity Skew, and Feature Skew. Details are in Sec.4.1.

Since Google firstly identified the performance degradation of FedAvg under data heterogeneity [11], numerous studies have focused on addressing this issue. Regularization-based methods such as FedProx [7] and FedDyn [1] constrain local model updates to remain close to the global model, offering computational efficiency and faster convergence. Unfortunately, their performance is limited in complex, highly heterogeneous scenarios. Clustering-based approaches, including FedSoft [13,4], group clients with similar data distributions to tackle heterogeneity. These methods struggle with dynamic adaptability and accurately measuring inter-client similarity. Client selection methods [29,5,28] aim to mitigate the heterogeneity by dynamically choosing participants for each training round, yet may raise fairness concerns. Other improvements like FedNova [19], FedBN [8], and FedNorm [9] have been proposed to improve convergence speed, reduce model bias, and improve data consistency, respectively.

Nevertheless, the multi-label nature of perception tasks and the pronounced skewness in 3D point clouds make autonomous driving a particularly challenging for federated learning.

2.2 Personalized Federated Learning

A single global model is insufficient to capture the diverse local characteristics of all vehicles, while fully independent local models fail to leverage shared global knowledge. In dynamic and heterogeneous autonomous driving scenarios, personalized federated learning (pFL) is essential for enhancing the adaptability and performance of local models.

The central goal of pFL is to effectively disentangle global information from personalized information, allowing local models to be fine-tuned for specific data distributions. Meta-learning approaches[2,10] enable fast adaptation but typically incur high computational overhead. Knowledge distillation[17] can facilitate efficient personalization, though they may suffer from performance degradation in complex environments. Clustering-based techniques[13,4] build multiple specialized global models for groups of clients with similar data distributions, offering limited personalization. Hybrid model architectures[27,23] introduce specialized modules to extract personalized features, yet often struggle with convergence in complex tasks. Data sampling strategies, such as FAST[21], utilize adaptive multi-armed bandits algorithms to dynamically identify personalized data, improving both efficiency and robustness.

In summary, while existing works provide valuable foundations, achieving efficient and scalable personalization in federated settings particularly for multi-label 3D perception tasks remains a key open challenge.

2.3 FL-Based Object Detection for ADVs

While FL provides an efficient solution for enabling collaborative model training in ADVs, the FL-based 3D detection remains underexplored and fragmented.

For example, AutoFed [30] addresses data heterogeneity through a multi-model framework that leverages autoencoder-based data imputation to enhance data complementarity. FedPylot [12] builds on YOLOv7 and integrates hybrid encryption schemes to improve training efficiency under privacy constraints; however, it lacks mechanisms for personalized model adaptation. In terms of personalization, Fed-PCN [22] introduces a personalized head feature extractor to enable local model fine-tuning, yet it has not been validated in autonomous driving scenarios. FedRAV [24] proposes a regional partitioning mechanism combined with a hypernetwork to train both vehicle-specific and region-specific models. Although it achieves a balance between global generalization and local adaptation, its reliance on vehicle location sharing raises privacy concerns.

Overall, existing research lacks an efficient FL framework for 3D object detection that balances global convergence and personalized model adaptation.

3 Method

3.1 The TeamFed Framework

FL involves two primary roles: a central server and multiple clients. In each communication round r, the server aggregates the model parameters uploaded by clients and broadcasts the updated global model. The clients, which refers to ADVs in our scenario, conduct local training for E epochs using their own data.

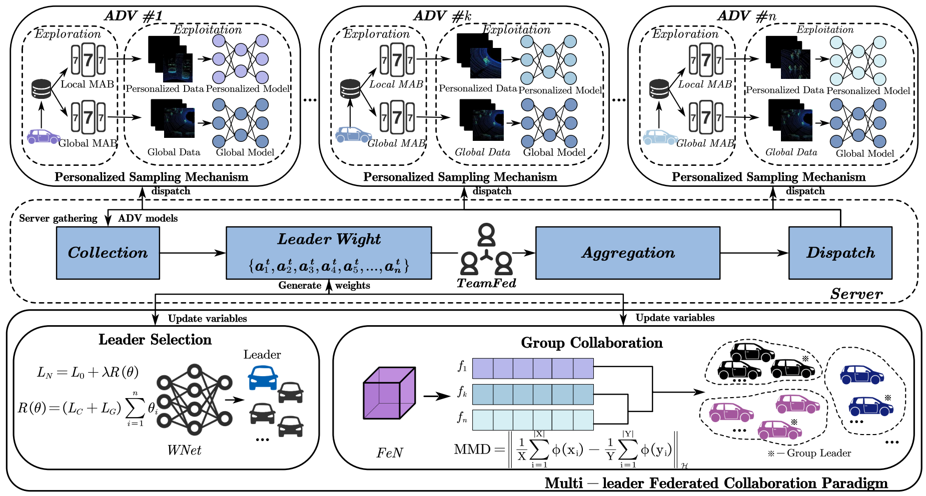

Fig. 2: The TeamFed framework. The *Multi-Leader Collaboration Paradigm* uses a policy network (WNet) to select leaders and employs the FeN module for feature extraction and clustering. The *Personalized Sampling Mechanism* uses the Multi-Armed Bandit (MAB) algorithm to separate global and personalized data for efficient global aggregation and local fine-tuning.

Fundamentally, FL is a collaborative learning paradigm in which clients jointly contribute to constructing a robust global model while simultaneously improving their local performance. As illustrated in Fig. 2, the proposed TeamFed framework consists of two key components: a *Multi-Leader Collaboration Paradigm* and a *Personalized Sampling Mechanism*. These components are designed to improve the collaborative process of federated learning through structured teamwork.

3.2 Multi-leader Collaboration Paradigm

In human society, effective team collaboration often relies on a division of labor, where capable individuals take on leadership roles to provide strategic direction and enhance task efficiency. Importantly, leadership is not static, it evolves with the stage and nature of the task. Inspired by this dynamic and adaptive leadership structure, we propose a FL leadership mechanism based on client weighting, enabling the system to dynamically identify and assign leadership roles during training.

In classical method FedAvg[11], the global model parameters ω_r at round r are calculated as a weighted average of clients' parameters $[\omega_r^1, \omega_r^2, \ldots, \omega_r^k]$, based on their data volumes $[n_1, n_2, \ldots, n_k]$:

$$\omega_r = \sum_{k=1}^{K} \frac{n_k}{N} \omega_r^k \tag{1}$$

where N is the total data volume and K is the number of clients. While data weighted aggregation alleviates the effects of data volume imbalance among clients, it remains insufficient for addressing other forms of heterogeneity, such as feature and label distribution shifts.

Theoretically, for round r, each client's update direction v_r^k is determined by its local data distribution, with the global optimal update direction denoted as v_r. The directional difference $D(v_r^k \| v_r)$ varies, where $D(\cdot\|\cdot)$ measures the directional discrepancy. Clients with v_k^r closer to v^r have smaller $D(v_k^r \| v^r)$, and these clients can be selected as leaders to guide the federated aggregation.

Assuming that the aggregation weights of the clients in round r are denoted by $\mathbf{A}\,[a_1^r, a_2^r, \ldots, a_k^r]$. We need to find the optimal $\mathbf{A}$ such that clients with smaller $D(v_k^r \| v^r)$ have larger a_k^r, thereby enabling these leading clients to facilitate federated aggregation. The optimization goal for a single round becomes:

$$\min_A \sum_{k=1}^{K} a_r^k D(v_r^k \| v_r) \tag{2}$$

subject to:

$$\sum_{k=1}^{K} a_r^k = 1 \quad \text{and} \quad a_r^k \geq 0 \quad \forall k$$

Since v_r is unknown, directly calculating the closest v_r^k and adjusting $\mathbf{A}$ is challenging[7]. To address this issue, we propose a leader selection module to adjust $\mathbf{A}$ based on the federated model's explicit performance, enabling weight adjustments even with incomplete information.

Leader Selection - Weight Adjustment We introduce a lightweight adjustment network, WNet, implemented by a multilayer perception (MLP), to dynamically adjust the weights of the clients based on the complexity of the model. The input and output of WNet are both the federated aggregation weights $\mathbf{A}$. The cross-entropy loss L_D between the latest weights and the historical weights is used to measure the change in the weights $\mathbf{A}$.

To jointly optimize the performance of the global and local model, we define the client performance loss L_C and the global performance loss L_G. Here, L_C quantifies individual client improvement by tracking the reduction in local training loss L_k, while L_G evaluates the enhancement of the global model using on its average accuracy on clients' local validation sets. Notably, the reliability of the validation metic mAP decreases as the evaluation interval Δr increases.

$$L_C = \frac{\sum \tanh(L_k)}{2K} \tag{3}$$

$$L_G = \frac{(1 - mAP)}{2\Delta r} \tag{4}$$

We introduce a regularization term $R(\theta)$ into the loss function of WNet, resulting in the final objective function L_w. Specifically, $R(\theta)$ is defined as the l_1 norm of the sum of the loss in client performance L_C and the loss in global performance L_G. The parameter θ represents the trainable weights of WNet and plays a key role in handling the non-differentiability of indicator variables during optimization.To ensure a balanced contribution of L_C and L_G in the regularization term $R(\theta)$, we apply a scaling factor to L_C using a tangent function.

$$L_w = L_D + \lambda R(\theta) \tag{5}$$

$$R(\theta) = (L_C + L_G) \sum \theta_i \tag{6}$$

The WNet, an auxiliary network initiated with FL model, is retrained before each round of federated aggregation and dynamically adjusts the weights $\mathbf{A}$ based on model performance. Clients with leading roles receive greater weights, aiding global model convergence in the optimal direction.

Group Collaboration - Dynamic Clustering Group collaboration is a widely adopted form of teamwork that improves efficiency and performance by dividing a large team into smaller, more agile subgroups. In the context of federated learning, this collaboration paradigm can be modeled as the clustering of clients. To address the limitations of existing clustering algorithms, such as poor dynamic adaptability and inaccurate assessment of local similarity, we propose a dynamic client clustering mechanism based on Maximum Mean Discrepancy (MMD).

Client-specific data discrepancies often lead to inconsistencies during federated aggregation. By grouping clients with similar data distributions, global model convergence can be significantly improved. To this end, we design a plug-and-play Feature Extractor Network (FeN) module that derives a low-dimensional representation vector f_k from intermediate outputs of the local base model. This representation enhances privacy protection and supports dynamic adjustments during training. The detailed architecture of FeN is provided in the appendix.

The feature vector f_k captures the statistical characteristics of the local data distributions, allowing the identification of clients with similar data traits. In each training round, f_k is uploaded to the server for further processing.

MMD quantifies the discrepancy between two distributions X and Y, based on their respective samples x_i and y_i. The MMD distance is calculated using Eq. 7, where $\phi(x)$ is a kernel function that maps the input into a Hilbert space $\mathcal{H}$. The server computes pairwise client distances using Eq. 7 and f_k to form a distance matrix $\mathbf{DM}$.

$$\mathrm{MMD}(X, Y) = \left\| \frac{1}{|X|} \sum_{i=1}^{|X|} \phi(x_i) - \frac{1}{|Y|} \sum_{j=1}^{|Y|} \phi(y_j) \right\|_{\mathcal{H}}^2 \tag{7}$$

With $\mathbf{DM}$, the server performs hierarchical clustering to dynamically group clients into G_r. This dynamic clustering evolves with changes in the feature vector matrix. For clients in the same cluster, initial weights a_g are assigned based on a

Gaussian distribution projection according to their distances. The client with the maximum a_g is selected as the group leader CL_g, while other clients' weights a_k within the group are adjusted using Eq. 8 and Eq. 9. We use the exponential function of the Gaussian distribution to adjust client weights, where σ is the standard deviation. Clients more similar to CL_g receive greater weights.

$$\mathrm{CL}_g = \arg\max_{i \in g} a_i \tag{8}$$

$$a'_k = a_k \times e^{-\frac{MMD(f_k, f_{\mathrm{CL}_g})}{2\sigma^2}} \tag{9}$$

3.3 Personalized Sampling Mechanism

Data-level personalized tuning is both broadly applicable and computationally efficient. To distinguish data samples contributing to global versus personalized model optimization, we formulate client-side data selection as a Multi-Armed Bandit (MAB) problem. The MAB framework enables optimal decision-making by balancing exploration and exploitation across multiple choices. In our formulation, each data segment within a client corresponds to an arm of the bandit. During the exploration phase, data are sampled and evaluated. During exploitation, high-reward samples are used for training. The reward signal is derived from training effectiveness, enabling dynamic differentiation of data samples.

Specifically, data samples associated with global features are aggregated to update the global model, while those aligned with personalized characteristics are retained for client-specific fine tuning. To facilitate this separation, we perform two independent exploration phases and design dedicated reward functions for each category. The global reward R_g in Eq. 10 is based on the improvement in global model performance, as measured by standard metrics such as mAP [14].

$$R_g = \sum_k \frac{n_k}{N} \Delta A p_k \tag{10}$$

For the personalized reward, we focus on the accuracy improvement of local model on the personalized validation set and the reduction in loss (L_k) after local training. We define this as R_p in Eq. 11. The variables in the reward function need to be normalized to balance the magnitudes.

$$R_{pk} = \Delta L_k + \Delta A p_k \tag{11}$$

Multi-Armed Bandit Exploration The MAB sampling process occurs before model training. During sampling, we record the selection frequency and average reward for each data point. For global sampling, the number of selections for sample i is denoted as $S_g[i]$, and its average reward is $\mu_g[i]$. For personalized sampling, the number of selections for sample i is $S_p[i]$, and its average reward is $\mu_p[i]$. The total number of samples is T_M. Each sampling iteration calculates the Upper Confidence Bound (UCB) using Eq. 12, selecting the top m_x samples with the highest UCB for training.

$$\mathrm{UCB}[i] = \mu[i] + \sqrt{\frac{2\ln T_M}{S[i]}} \tag{12}$$

As the multi-armed bandit undergoes multiple explorations, it can precisely obtain information, hence m_x gradually decreases with the aggregation round r. The hyperparameter C is used to mediate the rate of decrease.

$$m_x = m_k \cdot \frac{C}{\ln r + 1} \tag{13}$$

Multi-Armed Bandit Exploitation In the *Personalized sampling mechanism*, model training emulates the exploitation phase of MAB. Initially, it prioritizes the swift separation of personalized information. Later, it shifts to global exploitation to accelerate convergence. This is facilitated by initially utilizing personalized data D_p and progressively incorporating global data D_g. Once global convergence is achieved, the separated personalized data fine-tunes the model, creating a robust client-specific model without affecting global convergence.

For the number of training times E_p of the sampled data in each federated aggregation process r, based on an arithmetic sequence, we provide Eq. 14.

$$\frac{(E_b + E_p)R}{2} = E_i R \tag{14}$$

where E_i represents the initial value, E_b denotes the starting value, d is the adjustment step size, R is the total number of federated aggregation rounds.

The *Personalized sampling mechanism* achieves optimization through client model self-adjustment, enabling client self-management in the collaborative team.

3.4 Algorithm Process

The algorithmic process is detailed in Alg. 1. Lines 12-19 are for the *Multi-Leader Collaboration Paradigm*, and lines 21-28 are for the *Personalized sampling mechanism*, both executed according to the specific logic in lines 2-11 of Server. TeamFed is concise and efficient, saving time and computational resources. Extensive experiments demonstrate TeamFed's excellent performance in Sec.4.

Algorithm 1 TeamFed Algorithm

Input: K ADVs with local models $\{\omega_1, ..., \omega_K\}$, aggregation rounds R
Output: Global model ω_g,Personalized model Ω_p

Server executes:
Initialize θ, ω^g , $\Omega\{\omega_1, ..., \omega_K\}$,$A\{a_1, ..., a_K\}$,$G$, $\mathcal{F}\{f_1, ..., f_K\}$,**DM**
for round $r = 1$ to R **do**
    Broadcast ω^g to all ADVs
    $\mathcal{S} \leftarrow$ RandomSample($[n], m$)
    $\Omega, D_p, \mathcal{F} \leftarrow$ MultiArmControl($\mathcal{S}, D_k$)
    $G_r \leftarrow$ MakeGroup($\mathcal{F}$)
    $A_r \leftarrow$ LeaderReset(G_r, A_{r-1})
    $S_A \leftarrow \sum_{k\in\mathcal{S}} a_k^r$
    $\omega^g \leftarrow \sum_{k\in\mathcal{S}} \frac{a_k^r}{S_A}\omega_k^r$
$\Omega_p \leftarrow$ Personalized tuning on the D_p

function LeaderReset(G, A)
    $A \leftarrow$ Train WNet with Eq. 5,Eq. 6 and Group reset A with Eq. 8,Eq. 9
    return A

function MakeGroup($\mathcal{F}$)
    DM $\leftarrow$ Calculate distance with $\mathcal{F}$(Eq. 7)
    $G \leftarrow$ Hierarchical Clustering
    return G

Client executes:
function MultiArmControl($\mathcal{S}, D_k$)
    for each client $k \in \mathcal{S}$ **do**
        $D_{pk}, D_{gk} \leftarrow$ Exploration with UCB(Eq. 12)
        $\omega_k, f_k \leftarrow$ Exploitation with train on D_{gk}
        $\omega_k \leftarrow$ Exploitation with train on D_{pk}
        Updata personalized reward(Eq. 11) and global reward(Eq. 10)
    return $\Omega, D_p, \mathcal{F}$

4 Experiment

4.1 Experimental Setup

Datasets and Heterogeneity settings. We conducted experiments on KITTI and SnowyKITTI, widely used for evaluating computer vision algorithms in autonomous driving. We simulated four types of data heterogeneity: Class Skew, Feature Skew, Quantity Skew, and Hybrid Skew.

- Class Skew: The class distribution is imbalanced among ADVs.
- Feature Skew: Data for some vehicles is collected in snowy scenarios, and radar sampling rates vary across vehicles. Different features of the same object appear in vehicles' local datasets.

– Quantity Skew: The number of local data frames varies among different ADVs.
– Hybrid Skew: A Dirichlet distribution is used to assign different classes and quantities to ADVs. Additionally, snow noise and downsampling are applied to some ADV data.

Models. OpenPCDet [18] is widely used for 3D point cloud detection. For comparison, we implement methods like FedAvg on OpenPCDet. All ADVs use PointPillar [6] as the base model to reduce local computational burden and enable fast detection.

Baseline. Our TeamFed is compared with 6 categories of baseline methods: (1) vanilla FedAvg; (2) regularization methods FedProx and FedDyn; (3) clustering-based FedGC; (4) global adjustment methods FeNova and FedNorm; (5) client selection methods CSFedAvg and FedCCBS; (6) Personalized Method FedBN.

Implementation Details. We use 8 NVIDIA V100 GPUs for training. For global model performance comparison, we set up 100 clients with a sampling rate of 0.2 per round; for local model performance comparison, 10 clients with a sampling rate of 0.6 per round. Unless specified otherwise, we set the federated aggregation rounds to 30, with each ADV training locally for 3 epochs per round. We use the SGD optimizer with a batch size of 4. All methods are optimized using recommended parameters.

Table 1: The mAp (%) of BEV for FL methods under various data skews.

Algorithm	Class Skew			Feature Skew			Quantity Skew			Hybrid Skew		
	car	cyc.	ped.	car	cyc.	ped.	car	cyc.	ped.	car	cyc.	ped.
FedAvg[11]	83.90	67.97	56.56	84.61	58.97	53.21	85.02	63.17	57.98	85.17	59.07	51.86
FedProx[7]	79.72	59.58	52.00	79.23	45.85	49.24	79.95	59.63	54.65	79.66	52.22	44.09
FedNova [19]	83.95	67.64	57.79	83.27	55.41	55.95	84.64	64.07	59.58	86.66	61.30	53.08
FedNorm [9]	79.87	57.85	51.64	79.91	48.77	41.41	80.13	55.70	51.22	79.43	51.72	37.81
CsFedAvg [29]	82.20	55.77	47.28	84.20	54.40	49.94	85.02	60.84	59.01	84.60	58.48	48.78
FedDyn[1]	77.78	54.38	47.66	84.15	53.86	52.46	82.60	58.82	53.65	81.52	49.79	47.32
FedGC[5]	80.22	58.81	53.62	84.43	55.12	48.61	84.61	60.93	56.18	85.18	56.99	50.19
FedCCBS [28]	83.58	62.06	54.62	84.60	55.51	51.32	84.63	62.47	58.22	86.56	58.85	48.01
TeamFed	**85.40**	**70.28**	**58.93**	**88.38**	**68.07**	**63.45**	**86.05**	**68.32**	**65.18**	**88.98**	**68.98**	**64.46**
Centralized	89.02	74.15	65.30	88.74	71.72	<u>62.53</u>	89.02	74.15	65.30	<u>88.74</u>	71.72	<u>62.53</u>

4.2 Results

More Robust Global Model with TeamFed This experiment compares the performance of federated global models using different methods under varying data heterogeneity conditions. Performance is quantified using mAp of Bird's-Eye-View (BEV), and results are presented in Tab. 1.

The experimental results show that TeamFed consistently outperforms other algorithms under diverse data heterogeneity conditions. For instance, in the *Hybrid Skew* scenario, TeamFed achieves the best performance, with an average improvement of 13.6% over other algorithms. Across all categories, TeamFed demonstrates superior performance, primarily due to its effective client-side collaboration. Compared with other methods, TeamFed achieves an 15.8% improvement in global model detection accuracy, approaching or surpassing centrally trained models. Overall, TeamFed maximizes informative data utilization under identical training conditions, achieving remarkable performance.

Faster convergence with TeamFed TeamFed achieves faster convergence across diverse scenarios and categories, as illustrated in Fig. 3. Specifically, TeamFed requires 41% fewer communication rounds to reach the average standard compared to other methods. During the initial phase of federated aggregation, TeamFed collects data that better aligns with the global distribution through global exploration, achieving rapid convergence. Additionally, the multi-leader weight selection mechanism facilitates dynamic adjustment of the convergence step and direction, enabling TeamFed to outperform efficiently.

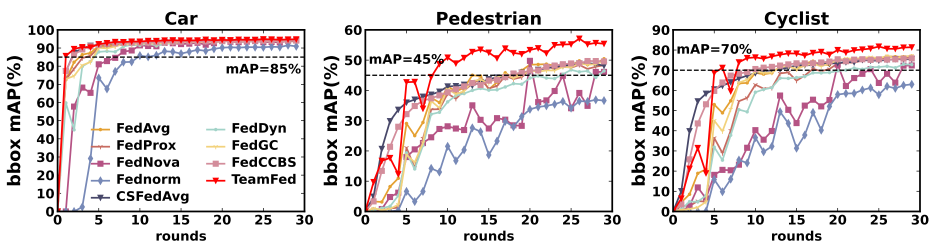

Fig. 3: Convergence comparison for FL methods on mAp (%) of BEV

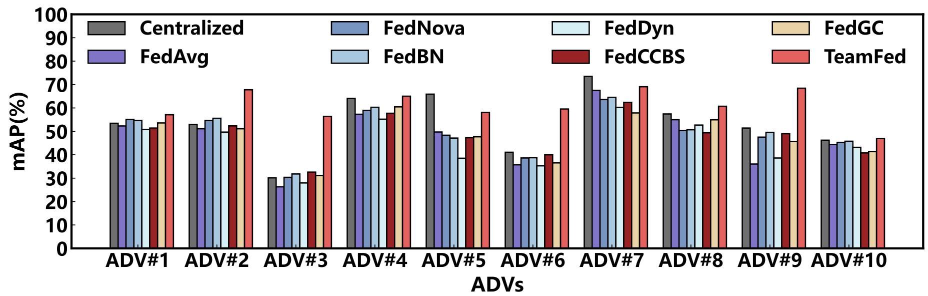

Fig. 4: Local model comparison for FL methods on mAp (%) of BEV

More Effective Local Model with TeamFed We conducted training under label skew conditions and constructed local test sets to evaluate the client models of different methods. The results are shown in Fig. 4. TeamFed demonstrated superior performance across multiple ADVs, with the highest improvement of 38% on ADV#9 compared to other FL methods. Notably, TeamFed's client models outperformed centralized training in 80% of ADVs, thereby demonstrating its remarkable ability to adapt to diverse data distributions. The Personalized Sampling Mechanism enhances client adaptability to diverse scenarios. Additionally, with $C = 0.8$ (Eq. 13), TeamFed reduces computational load by 29.5%, improving computational efficiency.

4.3 Ablation study

The *Personalized Sampling Mechanism* enhances model performance by enabling local models to better fit individual data distributions. We conducted ablation studies on local models and evaluated performance using autonomous driving metrics such as 2D bounding box (bbox), 3D bounding box (3D), bird's-eye-view (bev), and average orientation similarity (aos).

With the results shown in Fig. 5, Personalized Sampling significantly enhances the performance of local models, achieving the best performance on *Pedestrian*, with a average improvement of 26%. The personalized tuning of local models effectively meets the diverse scenarios in autonomous driving and aligns well with real-world requirements. More experiments are presented in the appendix.

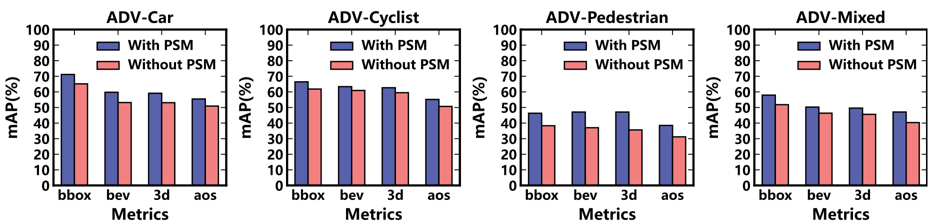

Fig. 5: Ablation study of Personalized Sampling using diverse Metrics on different data categories.

5 Conclusion

This work proposes TeamFed, a FL-based online collaborative training framework, designed to address data heterogeneity, privacy protection, and model personalization in 3D point cloud object detection for ADVs. Experiments conducted across diverse data heterogeneity scenarios demonstrate that TeamFed rapidly constructs a robust global model and achieves effective personalization through *Multi-leader Collaboration Paradigm* and *Personalized Sampling Mechanisms*. Future work will focus on validating TeamFed's performance in broader real-world scenarios and diverse ADVs.

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

References

1. Durmus, A.E., Yue, Z., Ramon, M., Matthew, M., Paul, W., Venkatesh, S.: Federated learning based on dynamic regularization. In: International conference on learning representations (2021)

2. Fallah, A., Mokhtari, A., Ozdaglar, A.E.: Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS (2020)
3. Ghodsi, Z., Javaheripi, M., Sheybani, N., Zhang, X., Huang, K., Koushanfar, F.: zprobe: Zero peek robustness checks for federated learning. In: IEEE/CVF International Conference on Computer Vision, ICCV, Paris. pp. 4837–4847 (2023)
4. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
5. Ji, X., Wu, H.T., Cui, T., Zhang, Y., Xu, L.: Fedgc: Federated learning on non-iid data via learning from good clients. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). pp. 181–194. Springer (2024)
6. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 12697–12705 (2019)
7. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)
8. Li, X., Jiang, M., Zhang, X., Kamp, M., Dou, Q.: Fedbn: Federated learning on non-iid features via local batch normalization. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria (2021)
9. Lian, Z., Liu, W., Cao, J., Zhu, Z., Zhou, X.: 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. pp. 619–626 (2022)
10. Liang, P.P., Liu, T., Ziyin, L., Allen, N.B., Auerbach, R.P., Brent, D., Salakhutdinov, R., Morency, L.P.: Think locally, act globally: Federated learning with local and global representations (2020)
11. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA. Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282 (2017)
12. Quéméneur, C., Cherkaoui, S.: Fedpylot: Navigating federated learning for real-time object detection in internet of vehicles. CoRR **abs/2406.03611** (2024)
13. Ruan, Y., Joe-Wong, C.: Fedsoft: Soft clustered federated learning with proximal local updating. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 8124–8131 (2022)
14. Schreier, T., Renz, K., Geiger, A., Chitta, K.: On offline evaluation of 3d object detection for autonomous driving. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023 - Workshops, Paris, France, October 2-6, 2023. pp. 4086–4091 (2023)
15. Shi, J., Wu, Y., Zeng, D., Tao, J., Hu, J., Shi, Y.: Self-supervised on-device federated learning from unlabeled streams. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **42**(12), 4871–4882 (2023)
16. Sun, G., Mendieta, M., Luo, J., Wu, S., Chen, C.: Fedperfix: Towards partial model personalization of vision transformers in federated learning. In: IEEE/CVF

International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 4965–4975 (2023)
17. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 8432–8440 (2022)
18. Team, O.D.: Openpcdet: An open-source toolbox for 3d object detection from point clouds (2020)
19. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
20. Wang, J., Yuan, Y., Luo, Z., Xie, K., Lin, D., Iqbal, U., Fidler, S., Khamis, S.: Learning human dynamics in autonomous driving scenarios. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 20739–20749 (2023)
21. Wang, Z., Xu, H., Xu, Y., Jiang, Z., Liu, J., Chen, S.: FAST: enhancing federated learning through adaptive data sampling and local training. IEEE Trans. Parallel Distributed Syst. **35**(2), 221–236 (2024)
22. Xue, A., Zhu, X., Wang, J., Xiao, J.: Diversified point cloud classification using personalized federated learning. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021. pp. 1–8 (2021)
23. Yang, X., Huang, W., Ye, M.: Fedas: Bridging inconsistency in personalized federated learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024. pp. 11986–11995 (2024)
24. Zhai, Y., Zhou, P., He, Y., Qu, F., Qin, Z., Jiao, X., Liu, G., Guo, S.: Fedrav: Hierarchically federated region-learning for traffic object classification of autonomous vehicles. CoRR **abs/2411.13979** (2024)
25. Zhang, G., Chen, J., Gao, G., Li, J., Liu, S., Hu, X.: Safdnet: A simple and effective network for fully sparse 3d object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024. pp. 14477–14486 (2024)
26. Zhang, J., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., Cao, J., Guan, H.: GPFL: simultaneously learning global and personalized feature information for personalized federated learning. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 5018–5028 (2023)
27. Zhang, J., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., Guan, H.: 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, KDD 2023, Long Beach, CA, USA. pp. 3249–3261 (2023)
28. Zhang, L., Lin, C., Bie, Z., Li, S., Bi, X., Zhao, K.: Client selection mechanism for federated learning based on class imbalance. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). pp. 266–278. Springer (2024)
29. Zhang, W., Wang, X., Zhou, P., Wu, W., Zhang, X.: Client selection for federated learning with non-iid data in mobile edge computing. IEEE Access **9**, 24462–24474 (2021)
30. Zheng, T., Li, A., Chen, Z., Wang, H., Luo, J.: Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving. In: Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, ACM MobiCom 2023, Madrid, Spain, October 2-6, 2023. pp. 15:1–15:15 (2023)