

Generalization vs. Personalization:

A Dynamic Layer Masking Solution for Data Heterogeneous Federated 3D Object Detection

Jolle Verhoog

Delft University of Technology



Generalization vs. Personalization:

A Dynamic Layer Masking Solution for Data
Heterogeneous Federated 3D Object Detection

by

Jolle Verhoog

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on June 10, 2026 at 10:00

Student Number:	5171598
Project Duration:	September 2025 – June 2026
Faculty:	Faculty of Mechanical Engineering, Delft
Track:	Robotics
Thesis Committee:	A. B. Ünal, PhD, TU Delft, responsible supervisor Dr. H. Caesar, TU Delft, chair, formal supervisor Dr. L. Ferranti, TU Delft, external examiner Dr. M. Popovic, TU Delft, external examiner

Cover: Generated using ChatGPT, personal prompt

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



Generalization vs. Personalization: A Dynamic Layer Masking Solution for Data Heterogeneous Federated 3D Object Detection

Jolle Verhoog*
TU Delft

Ali Burak Ünal TU
Delft

Holger Caesar
TU Delft

Abstract

Robust perception in intelligent vehicles demands 3D object detectors that remain dependable under domain shifts, such as changes in time of day, location, or weather. These factors result in heterogeneous data, where domain-specialized models may be advantageous. Naively re-training models for each scenario is computationally expensive, energy-intensive, and possibly infeasible due to data scarcity. Instead, federated learning offers an efficient alternative to costly retraining for multi-domain adaptation. This study explores how federated strategies for addressing data heterogeneity can improve the cross-domain robustness of 3D object detectors to temporal variations. Traditionally, these strategies require manually selecting which parts of the model are globally aggregated and which are locally personalized. To resolve this manual choice, a new Centered Kernel Alignment (CKA)-based strategy, FedCKA, is proposed. It dynamically handles the personalization-generalization trade-off by selectively sharing representation-consistent layers across clients. Evaluation on a unified multi-domain benchmark on nuScenes shows that FedCKA outperforms established federated baselines, including FedBN, FedRep, and FedSelect, in heterogeneous environments. The findings offer both a comparative benchmark and a promising direction for federated, location-, weather-, and illumination-robust 3D perception.

1. Introduction

3D Object Detection (3D-OD) is central to safe autonomous navigation because it provides the locations, orientations, and sizes of surrounding objects, making its robustness directly relevant to the reliability of Intelligent Vehicles (IV). However, domain shifts across vehicles, times of day, weather conditions, and locations can degrade model convergence and accuracy [7] (see Figure 1). Retraining entire detection models for each domain shift is expensive, unus-



Figure 1. Representative nuScenes [3] samples across Boston during the day and Singapore during the night (rows) and clear/rain weather conditions (columns), with nighttime rain in Singapore highlighted in the bottom-right image.

tainable [25], and infeasible when a domain lacks sufficient data. Moreover, for IV applications, the large volume and privacy-sensitive nature [18] of multi-modal sensor data can make centralized collection infeasible.

Federated Learning (FL) provides a practical alternative for multi-domain adaptation without data centralization [12, 14, 15]. In this paradigm, clients train local model copies across heterogeneous environments and exchange only model updates with a coordinating server. This enables domain-specific local adaptation while allowing data-scarce domains to benefit from shared global knowledge. By avoiding the centralization of raw sensor data, FL can mitigate potential bandwidth and privacy constraints, but it also introduces a central modeling question: which model components should remain globally shared, and which should adapt locally? This partitioning becomes especially challenging when domain shifts affect sensor backbones and sensor fusion modules differently [29].

This complexity helps explain why FL remains largely unexplored for multi-modal 3D-OD. Although the pioneering multi-modal FL detection works, AutoFed [34] and FedM2Former [33], report robustness to heterogeneous conditions, they do not split datasets into explicit location-, weather-, and illumination-based domains, nor are they compared within a unified benchmark setting. As a result, no prior review or benchmark systematically evaluates federated 3D-OD under these domain shifts, which this work addresses.

*MSc Robotics student, number 5171598. Thesis, 30 May 2026.

These gaps motivate two research questions. First, to what extent can federated personalization and generalization strategies mitigate the adverse effects of client-data heterogeneity in 3D-OD under temporal shifts, such as day-night and weather variations? Second, given that different model components may respond differently to such shifts, how can representation consistency be leveraged to dynamically determine which parameters to share across heterogeneous 3D-OD clients?

To answer these questions, this work studies federated 3D-OD, with particular attention to heterogeneous client data handling through personalization and aggregation methods. Although privacy, client reliability, communication limits, and sensor configuration variability remain important in FL-based 3D-OD, this work focuses on location-, weather-, and time-induced domain adaptation. Furthermore, this work assumes that client-level labeled data is available. These assumptions isolate the heterogeneity challenge from system-level constraints and lead to the following contributions:

1. Systematically shows that local personalization methods generally outperform global generalization methods when addressing data heterogeneity in federated 3D object detection.
2. Establishes a unified, problem-specific benchmark by reproducing and evaluating multiple representative federated methods for data heterogeneity in a common 3D object detection setting.
3. Introduces a dynamic layer masking method based on Centered Kernel Alignment (CKA), which selectively shares representation-consistent layers across clients and outperforms established baselines in heterogeneous environments.

The paper proceeds as follows. Section 2 reviews related work in multi-modal 3D-OD, FL, and data heterogeneity. Section 3 introduces the proposed CKA-based layer masking methodology. Section 4 details the experimental design, federated baselines, and the evaluation protocol. Section 5 presents the benchmark results and evaluates the proposed method against established baselines. Finally, Sections 6, 7, and 8 discuss the findings, limitations, and future directions before concluding the paper.

2. Related work

3D Object Detection underpins IV decision-making, but remains challenging under occlusion, truncation, distant targets, crowded scenes, adverse weather, and the accuracy-efficiency trade-off [31]. Because these challenges strongly interact with sensing geometry, 3D-OD methods are commonly categorized by their primary sensor input modality: camera-based, LiDAR-based, and multi-modal [27]. On benchmarks such as nuScenes, many 3D-OD methods combine LiDAR point clouds and images [4], integrating se-

mantic visual information with precise point-cloud depth. Robustness-focused multi-modal methods often rely on Bird’s-Eye-View (BEV) representations and cross-modal attention for fusion, but their performance remains limited across temporally different domains such as day, night, and adverse weather [2]. This limitation motivates learning strategies that can efficiently exploit data from multiple domains without requiring all sensor data to be centralized.

Federated Learning (FL) provides such a strategy by enabling multiple clients to train a global model while keeping their data decentralized. Unlike conventional distributed learning, FL does not assume client data to be independent and identically distributed (IID), centrally shared, or of equal size [18]. This heterogeneity introduces challenges for convergence [14], fairness [19], and cross-domain generalization [16].

The foundational FL algorithm, FedAvg [18], trains models independently on multiple clients using their local datasets. A central server then aggregates the client parameters through a weighted average and broadcasts the updated global model for the next round. While FedAvg can approach centralized performance under IID conditions, its performance degrades when clients have different data distributions [10]. This degradation makes data heterogeneity a central challenge in FL [14].

Data heterogeneity in FL fundamentally shapes what client models should share and how the server should aggregate their updates. Approaches to managing this heterogeneity span data preprocessing, training objectives, and aggregation schemes [21]. For all of these, the main distinction is the objective: aligning all clients toward one domain-invariant model, or producing multiple domain-adapted models through client personalization.

To learn a more domain-invariant global model, methods such as FedProx [14], FedDyn [1], and SCAFFOLD [12] reduce client drift during training. Other approaches use task arithmetic [11] to align client update gradients and estimate an invariant update direction [20, 30, 32]. However, one global model may not be optimal for all clients, and these methods require problem-specific manual parameter tuning.

To address this first limitation, personalized FL methods instead maintain multiple domain-adapted models that retain shared knowledge while adapting to client-specific domains. Common approaches keep batch normalization or other targeted layers client-specific, as in FedBN [15] and FedRep [5]. Because these methods require manually predefined global and domain-specific layer partitions, this static approach can be suboptimal when domain specificity varies across layers or clients.

Recent work addresses this rigidity by using model updates to infer which parameters to mask for personalization. FedSelect [26] assumes that the largest parameter updates are the most domain-significant and personalizes those pa-

parameters. FedMC [8] uses the Fisher information of model parameters to identify the most sensitive parameters for personalization. Although these methods avoid manual partitioning, they still rely on parameter-space importance and implicitly assume that same-index parameters across clients correspond to comparable functional components, which may not always hold [24].

To overcome the limitations of parameter-space comparisons and manual layer selection, this work selects personalized layers based on representational consistency across clients.

3. Proposed Method

3.1. Motivation

FedCKA is motivated by the idea that layers should be shared across clients only when their representations remain consistent after local adaptation. While existing federated methods rely on parameter-space distances [26] or gradient magnitudes [8], this work argues that representational consistency is a more reliable proxy for transferability and uses it to determine which layers to personalize. This strategy incorporates Centered Kernel Alignment (CKA) [13] to measure how differently two layers map the same set of incoming features to their outgoing features. This metric produces a normalized score bounded in $[0, 1]$, with 0 indicating strong dissimilarity and 1 indicating high similarity in the representation space transformation for each layer.

By comparing layers at identical depths across the global and local models, FedCKA estimates the degree to which each layer remains transferable after local adaptation. The most dissimilar layers for each model can then be personalized by masking them from the global model aggregation. This lets certain layers drift to domain-specific optima, while also sharing representationally similar layers to transfer global knowledge. Because only unmasked parameters are transmitted, FedCKA reduces the amount of shared parameter information by the masking percentage, improving bandwidth efficiency and potentially enhancing privacy.

3.2. FedCKA System Architecture

FedCKA follows a two-phase federated training protocol. The first phase uses standard FedAvg for the first quarter of the total communication rounds to learn an initial shared representation across clients. This warm-up phase prevents CKA from being computed on randomly initialized or weakly trained layers, where representation similarity scores would be uninformative. After this warm-up establishes an initial global model, training transitions to the CKA-guided personalization phase.

As illustrated in Figure 2, this phase entails coordinated exchanges between the clients and the central server. At the end of each round, each client has trained its local model for E epochs on its own data. The first step is to estimate local

divergence by computing the CKA score between the saved state of the previous global model, G_{t-1} , and its newly updated local model. This computation uses a random subset of the client’s local data passed through both models without requiring ground-truth labels. Based on the representational similarity of the layers, the client determines which layers to personalize and thus mask from global aggregation. Consequently, each client sends its unmasked layers to the central server. The server aggregates these shared parameters to obtain the new global model (G_t). Crucially, instead of broadcasting the entire global model downstream, the server acts as a personalized router, returning only the updated, unmasked layers specific to each client. Finally, each client updates its local model by injecting the unmasked parameters from the global model update. While payloads are reduced, this bandwidth efficiency comes at the cost of local computational overhead due to the double client-side forward pass. This process allows each client to personalize its most dissimilar layers while benefiting from the knowledge learned by other clients. Implementation details are provided in Appendix D.

3.3. Centered Kernel Alignment

Centered Kernel Alignment [13] is a similarity index between two layers, based on the Hilbert-Schmidt Independence Criterion (HSIC) [9], which can be used to compare the similarity between the representations induced by two layers. Different layer parameterizations may yield similar representations, making direct comparison in parameter space insufficient. CKA addresses this limitation by comparing layer activations rather than parameters. Similar use of CKA has been adopted in representation analysis of vision architectures; for example, Raghu et al. [23] use CKA to compare internal representations of transformers and convolutional neural networks across layers. In this work, the same principle is used in a federated setting, where CKA measures whether a locally trained client layer remains representationally aligned with the previous global model.

Given two matrices $X \in \mathbb{R}^{n \times p_1}$ and $Y \in \mathbb{R}^{n \times p_2}$ representing the output of two layers, the dot-product similarity can be computed through the $n \times n$ Gram matrices $K = XX^T$ and $L = YY^T$ as

$$\langle \text{vec}(XX^T), \text{vec}(YY^T) \rangle = \text{tr}(XX^TYY^T) = \|Y^T X\|_F^2.$$

Here, $\text{vec}(\cdot)$ denotes the column-wise vectorization and $\text{tr}(\cdot)$ represents the trace operator. Then the empirical HSIC is defined as the inner product between the centered Gram matrices,

$$\text{HSIC}(K, L) = \frac{\text{tr}(KHLH)}{(n-1)^2},$$

where $H = I_n - \frac{1}{n}J_n$ is the centering matrix, I_n is the $n \times n$ identity matrix, and J_n is the $n \times n$ all-ones matrix. The

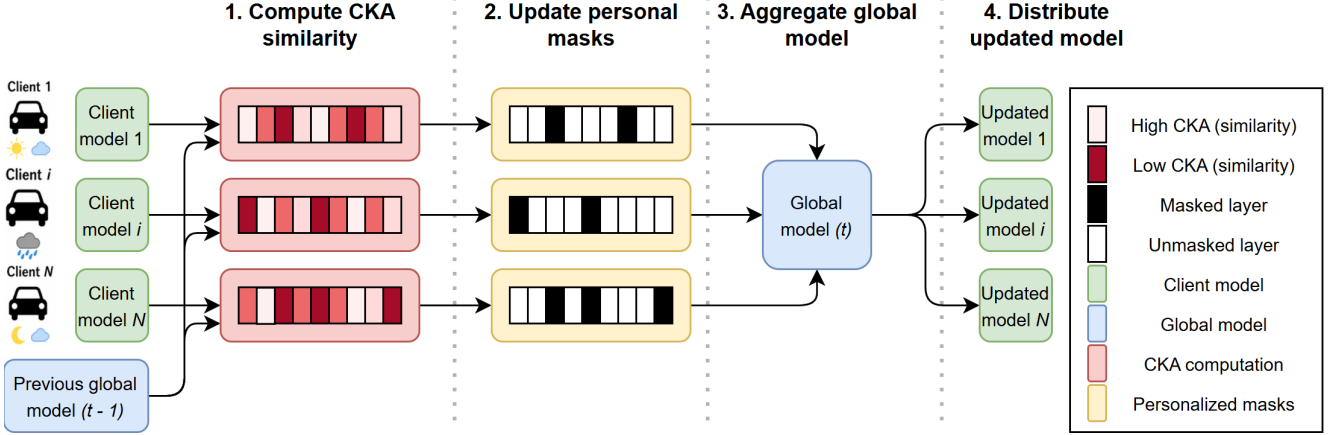


Figure 2. Proposed client-specific aggregation via CKA-based layer masking. At round t , each client compares its local model with the previous global model from round $t - 1$ using layer-wise CKA and converts the scores into a binary mask. Only unmasked layers are transmitted, aggregated by the server, and returned to the clients. Each client overwrites these layers with the global update, while masked layers remain local for the next round.

HSIC is then normalized, which yields the CKA similarity

$$\text{CKA}(X, Y) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K) \text{HSIC}(L, L)}}.$$

CKA therefore indicates whether the same samples produce a similar relational structure at a given layer. The metric is mean-centered and normalized, which makes it invariant to isotropic scaling and orthogonal transformations. Because CKA does not rely on one-to-one alignment between individual neurons or feature channels, the exact ordering of neurons does not affect the comparison. It therefore captures whether a layer similarly organizes the same input samples across representations, rather than whether individual activations match directly.

3.4. Personalization strategies

At each subsequent round, the personalization masks are constructed with the layer-wise CKA scores using one of two strategies: selecting a target percentage of parameters, analogous to FedSelect [26] and FedMC [8], or selecting all layers below a CKA similarity threshold. The latter allows client models to personalize a different amount of their local parameters based on layer dissimilarity.

To formally define the personalization strategies, let $s_l \in [0, 1]$ be the computed CKA similarity score for layer l , and $M_l \in \{0, 1\}$ be the binary mask where $M_l = 1$ denotes personalization. The locally updated weights $\hat{w}_l$ are computed from the client weights w_l and incoming global weights $\bar{w}_l$ as:

$$\hat{w}_l = M_l w_l + (1 - M_l) \bar{w}_l \quad (1)$$

This personalization mask is cumulative across communication rounds. Once a layer is selected for personalization ($M_l = 1$), it is permanently decoupled from global aggregation for subsequent rounds. Depending on the chosen strategy, the mask M_l is constructed in one of two ways:

Percentage-based masking: Let π be a permutation sorting the layers by ascending CKA score ($s_{\pi(1)} \leq \dots \leq s_{\pi(L)}$), p_l be the parameter count of layer l , and P be the total parameters in the network. Given a target personalization fraction $\rho \in (0, 1]$, we iteratively select the most dissimilar layers until their combined parameter count reaches the threshold ρP . Specifically, we select the first k layers in the sorted sequence such that:

$$\sum_{i=1}^{k-1} p_{\pi(i)} < \rho P \leq \sum_{i=1}^k p_{\pi(i)} \quad (2)$$

The subset of personalized layers is then defined as $V = \{\pi(1), \dots, \pi(k)\}$, and the mask is applied as:

$$M_l = \mathbb{I}(l \in V) \quad (3)$$

Since masking is performed at the layer granularity, the resulting personalized fraction can exceed ρ when the final selected layer, $\pi(k)$, crosses the target.

Threshold-based masking: Instead of enforcing a target proportion of the network to be personalized, this strategy evaluates the divergence of each layer independently. For a predefined absolute similarity threshold $\tau \in [0, 1]$, a layer is personalized only if its CKA score indicates sufficient divergence from the global model:

$$M_l = \mathbb{I}(s_l < \tau) \quad (4)$$

Unlike the percentage-based approach, which dictates a fixed personalization footprint for each client regardless of actual layer alignment, threshold-based masking dynamically adjusts the volume of personalized parameters. Consequently, the fraction of parameters withheld can vary across clients according to their layer-wise divergence, allowing the system to adapt.

4. Experiment Design

4.1. Preliminaries

The experimental framework uses a federated architecture in which each client is restricted to a distinct environmental domain. As illustrated in Figure 3, the domains encompass Singapore and Boston, featuring rainless and rainy scenes during both day and night. These five domains are labeled A-E for brevity.

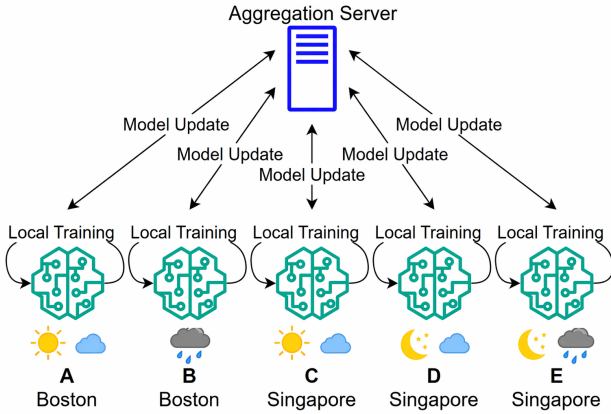


Figure 3. Overview of the proposed federated learning framework, in which five clients train models locally in their own domains (A-E) and share them with a global server. The server then sends the updated models back to the clients, who continue training on them.

4.2. Dataset

This benchmark uses the nuScenes dataset, one of the most impactful datasets for onboard perception [17], which also distinguishes between times of day and weather conditions. The 700 training and 150 validation scenes, each consisting of roughly 40 annotated samples in a 20-second time window, are split among the clients, as shown in Table 1. These clients therefore exhibit heterogeneity not only in location, time of day, and weather, but also in scene count and class composition. To illustrate, the Singapore training set contains, on average, three times as many annotated motorcycles, while Boston features more than twice as many cars. Although domain E consists of only three validation scenes and may therefore be too noisy for stable standalone conclusions, its small size and compounding domain shifts make it a useful stress test for knowledge transfer at the edges of the federated system. The full class imbalance analysis is provided in Appendix A.

4.3. Detector

To ensure the benchmark reflects data heterogeneity in federated settings rather than detector brittleness, this work uses the Cross-Modal Transformer (CMT) detection model. This architecture not only achieves strong robustness through attention-based multimodal fusion [28] but also outperforms other state-of-the-art methods on a syn-

Table 1. Federated domain definitions and corresponding train/validation scene counts. “Sing.” denotes Singapore.

Domain	A	B	C	D	E
Location	Boston	Boston	Sing.	Sing.	Sing.
Time	Day	Day	Day	Night	Night
Weather	Clear	Rain	Clear	Clear	Rain
Train scenes	265	125	226	71	13
Val scenes	53	24	58	12	3

thetic robustness benchmark dataset [2]. To isolate the impact of different federated strategies, the detector configuration and all hyperparameters are held constant across all experiments. The complete configuration details are provided in Appendix B.

4.4. Federated strategies

The benchmark compares strategies that either improve a single shared globally aligned model or preserve client-specific parameters for personalization. The globalization strategies include FedAvg [18], FedDyn [1], and PCGrad [30]. FedDyn aims to limit client drift by penalizing client models for deviating from the global model, while PCGrad aims to align client gradients toward a global consensus direction. For personalization strategies, two well-known layer-wise approaches are replicated: FedBN [15], which keeps the batch normalization parameters personalized, and FedRep [5], which shares the representation and personalizes the head. Furthermore, two parameter-personalization strategies are replicated: FedSelect [26], which identifies the biggest updating parameters as personal, and FedMC [8], which identifies the most influential parameters as personal. For all methods, the number of local epochs and federated aggregation rounds is the same, and for further details on implementations of these methods, see Appendix C.

4.5. Evaluation

To contextualize the federated benchmark, this work also evaluates detector performance on each domain separately and under centralized training. The own-domain-only performance represents the upper bound for that client if it had not joined the federated system, and the centralized performance is typically taken as the upper bound of the federated system’s performance. These reference points contextualize the federated results, which are reported in mean Average Precision (mAP), nuScenes Detection Score (NDS), and Car Average Precision (CAP). The NDS comprises the average of five times the mAP and the true-positive error terms for translation, scale, orientation, velocity, and attribute. Given that domains C, D, and E do not contain 1, 3, and 6 of the 10 classes, respectively, the mAP and NDS may be positively or negatively biased; hence, in addition to the standardized 3D-OD metrics, the CAP is also reported.

Table 2. Comparison of different FL methods across multiple domains. Non-federated runs are in italics. Performance is reported as mAP / NDS / CAP (Car AP) for each domain. The best-performing methods in both the Globalization and Personalization families are highlighted in bold. Full analyses, implementation details, and standard-deviation estimates are provided in Appendix C

Method	Domain A	Domain B	Domain C	Domain D	Domain E
<i>Centralized</i>	<i>0.61 / 0.66 / 0.84</i>	<i>0.61 / 0.68 / 0.85</i>	<i>0.72 / 0.67 / 0.83</i>	<i>0.60 / 0.64 / 0.85</i>	<i>0.58 / 0.58 / 0.84</i>
<i>Own Domain Only</i>	<i>0.58 / 0.64 / 0.81</i>	<i>0.47 / 0.51 / 0.75</i>	<i>0.59 / 0.62 / 0.76</i>	<i>0.14 / 0.24 / 0.47</i>	<i>0.01 / 0.07 / 0.04</i>
<i>Globalization</i>					
FedAvg [18]	0.28 / 0.37 / 0.54	0.27 / 0.36 / 0.51	0.34 / 0.38 / 0.53	0.28 / 0.35 / 0.56	0.20 / 0.33 / 0.51
FedDyn [1]	0.38 / 0.48 / 0.60	0.34 / 0.46 / 0.59	0.42 / 0.48 / 0.60	0.32 / 0.43 / 0.63	0.25 / 0.35 / 0.63
PCGrad [30, 32]	0.34 / 0.41 / 0.56	0.32 / 0.41 / 0.56	0.40 / 0.44 / 0.57	0.33 / 0.39 / 0.59	0.21 / 0.34 / 0.55
<i>Personalization</i>					
FedRep [5]	0.33 / 0.43 / 0.58	0.28 / 0.40 / 0.54	0.37 / 0.44 / 0.57	0.28 / 0.41 / 0.59	0.21 / 0.33 / 0.56
FedBN [15]	0.36 / 0.41 / 0.58	0.33 / 0.41 / 0.59	0.42 / 0.43 / 0.58	0.32 / 0.36 / 0.62	0.23 / 0.33 / 0.51
FedMC [8]	0.52 / 0.58 / 0.76	0.48 / 0.56 / 0.76	0.57 / 0.58 / 0.74	0.37 / 0.47 / 0.73	0.19 / 0.32 / 0.53
FedSelect [26]	0.52 / 0.59 / 0.77	0.46 / 0.51 / 0.72	0.55 / 0.56 / 0.73	0.32 / 0.40 / 0.71	0.27 / 0.34 / 0.53
FedCKA (<i>this paper</i>)	0.55 / 0.61 / 0.76	0.51 / 0.60 / 0.76	0.56 / 0.60 / 0.75	0.50 / 0.57 / 0.77	0.45 / 0.47 / 0.78

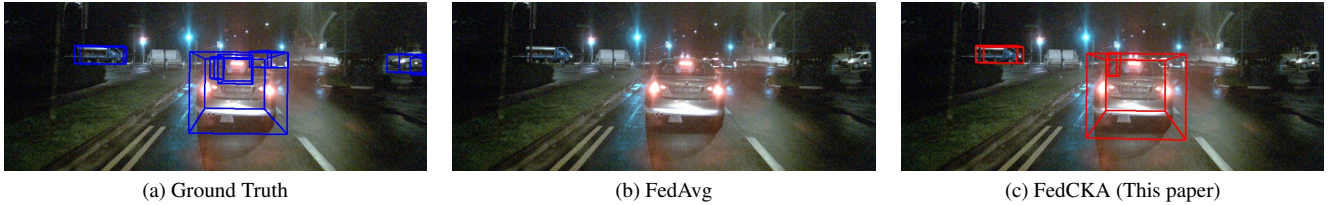


Figure 4. Qualitative comparison of 3D object detection performance under severe domain shift: Singapore at night with light rain. (a) Ground Truth bounding boxes. (b) Predictions from the baseline FedAvg model, illustrating missed detections. (c) Predictions from the proposed FedCKA framework demonstrate successful detection recovery and improved cross-domain robustness.

5. Experiments

To evaluate the performance of the proposed FedCKA method, this section compares it against federated heterogeneity methods and non-federated reference models. Federated baselines evaluate heterogeneous data handling, own-domain-only results measure isolated client performance, and centralized training provides an upper-bound reference. As shown in Table 2, the representational similarity-based approach outperforms all other federated baselines and comes closest to the centralized upper bound. Qualitative results for the most shifted domain, Singapore night rain, are also shown in Figure 4 and Figure 5.

5.1. Globalization versus Personalization

Traditional globalization methods do not produce a consensus model that rivals centralized performance. The static layer personalization strategies, FedBN and FedRep, which personalize the batch normalization (BN) layers or the head of each client, perform comparably to the globalization methods. However, even with this targeted adaptation, these models fall short. These BN and head layers may not have sufficient power to move the client models towards their local optima, as they have fewer personal parameters than the

parameter-level personalization strategies and do not influence the crucial intermediate cross-modal merging stages.

Among the established baselines, the parameter-level personalization strategies of FedSelect and FedMC achieve the highest performance. However, the proposed FedCKA method generally surpasses all evaluated methods across the three metrics (mAP, NDS, CAP), successfully navigating domain discrepancies through adaptive layer-wise personalization.

Ultimately, the difficulty of achieving a global consensus model across these clients highlights the severity of the domain gap. Because globalization forces different domains toward a single shared optimum, the resulting model remains suboptimal. Conversely, the success of the parameter-level baselines and FedCKA demonstrates that, by allowing a sufficiently large or important group of parameters to adapt to a domain, client models can move closer to their local optima.

5.2. Cross-Domain Generalization

Cross-domain generalization is a key test of robustness in a domain-heterogeneous setting, revealing the limitations of both isolated training and standard aggregation, as detailed in Tables 3a and 3b. Table 3a shows that although isolated

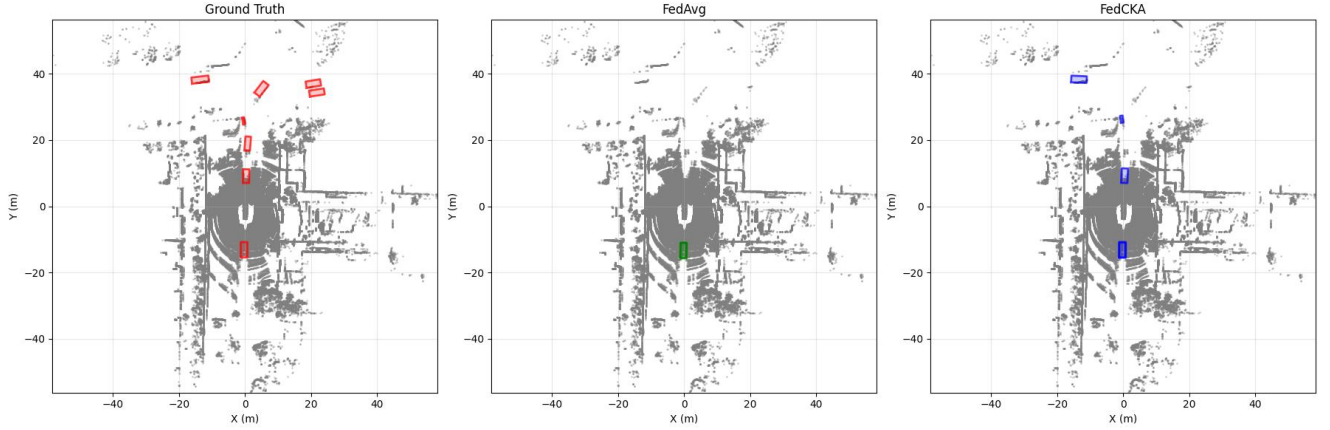


Figure 5. Qualitative comparison of 3D object detection performance under severe domain shift (Singapore night rain) in BEV view. On the left are the Ground Truth bounding boxes. In the middle are the predictions from the baseline FedAvg model, illustrating missed detections. On the right, the predictions from the proposed FedCKA framework demonstrate successfully recovered detections and improved cross-domain robustness.

Table 3. Cross-domain generalization matrices for client models (M) A–E. Each matrix reports NDS performance across validation domains (D) A–E. The diagonal shows performance when the training and validation data originate from the same domain, while the off-diagonal entries show cross-domain zero-shot performance.

(a) Own Domain Only					
	D-A	D-B	D-C	D-D	D-E
M-A	0.64	0.60	0.46	0.33	0.15
M-B	0.40	0.51	0.28	0.21	0.12
M-C	0.48	0.42	0.62	0.42	0.22
M-D	0.26	0.23	0.29	0.24	0.19
M-E	0.08	0.08	0.08	0.06	0.08

(b) FedCKA					
	D-A	D-B	D-C	D-D	D-E
M-A	0.62	0.60	0.59	0.52	0.47
M-B	0.58	0.59	0.56	0.49	0.45
M-C	0.56	0.55	0.59	0.53	0.48
M-D	0.45	0.44	0.53	0.50	0.53
M-E	0.47	0.47	0.52	0.45	0.45

training yields rich local models for domains A and C, their zero-shot transfer to some other domains remains poor. For instance, Model A drops to an NDS of 0.15 on domain E, indicating that strong own-domain performance does not necessarily translate into general cross-domain knowledge. This underscores that the benchmark captures meaningful domain heterogeneity rather than only differences in local dataset size.

Conversely, global aggregation with FedAvg, FedDyn, or PCGrad sacrifices local performance on the richer domains A, B, and C, while maintaining stable but limited cross-domain performance. Because globalization forces all clients to share a single global model, a cross-domain evaluation matrix for these methods yields identical rows for every client, indicating stable yet consistently subopti-

mal generalization across all domains.

In contrast, Table 3b shows that FedCKA combines stronger cross-domain transfer with client-specific adaptation by aggregating only layers with high representational similarity. FedCKA boosts not only cross-domain performance but also own-domain performance (shown on the diagonal), surpassing the baseline methods previously evaluated in Table 2. This indicates that FedCKA preserves domain-specific layers while still sharing layers that transfer useful knowledge across clients. For the other personalization models in Table 2, the cross-domain generalization matrices are provided in Appendix E.

5.3. Threshold Ablation

Both proposed personalization strategies, detailed in Section 3.4, depend on either a global masking ratio (ρ) or a similarity threshold (τ). At the upper bound ($\rho = 1$ or $\tau = 1$), the models are entirely personalized, reducing to the “Own Domain Only” baseline in Table 2; at the lower bound ($\rho = 0$ or $\tau = 0$), the system defaults to standard FedAvg aggregation. Table 4 summarizes the impact of varying these parameters, while the full ablation, mask stability analysis, and layer-type analysis are provided in Appendix F.

The resulting masks remain stable across repeated runs and consistently isolate the same structural components, suggesting that FedCKA reliably targets domain-sensitive layers rather than collapsing into arbitrary or fully local personalization.

Table 4. FedCKA ablations for Equations 3 & 4 reporting average NDS and masking ratio across clients.

Setting	NDS	Mask	Setting	NDS	Mask
$\rho = 0.01$	0.551	1.19%	$\tau = 0.05$	0.598	0.71%
$\rho = 0.10$	0.489	10.68%	$\tau = 0.25$	0.601	3.13%
$\rho = 0.40$	0.516	41.29%	$\tau = 0.50$	0.600	7.83%
			$\tau = 0.75$	0.596	4.42%

6. Discussion

The results on this problem-specific benchmark indicate that, for federated, data-heterogeneous 3D object detection, the global aggregation methods underperform relative to personalization methods. Although manually chosen static layer personalization can help, directly inferring personalization masks from the client models further boosts performance. Compared with magnitude- or gradient-based personalization, the results suggest that representation similarity is a stronger criterion for selecting personalized parameters, improving both out-of-domain and smaller-domain performance.

FedCKA reduces the need to hand-pick specific layer types or impose fixed personalization ratios, since the masking decision is instead based on measured representational similarity. Furthermore, FedCKA does not assume that same-index parameters serve the same function across clients, since it compares layer activations rather than parameter-space differences.

The learned masks provide further insight into which model components are most domain-sensitive. Among the learned masks, normalization layers are frequently personalized, suggesting sensitivity to domain-specific activation statistics and feature scaling under changes in weather, illumination, geography, and scene composition. The personalization of certain backbone, neck, and attention layers that encode higher-level and cross-modal features suggests that some deeper representations may be domain-sensitive. The attention mechanism may also adapt the weighting of the camera and LiDAR modalities. The task head remains largely shared, suggesting that the object-level prediction semantics are transferable, while the internal representations may adjust due to various shifts. This indicates that FedCKA does not create five independent local models but instead preserves a common detection-task structure.

FedCKA remains below centralized performance, as the centralized model has access to all data and can learn shared class-level patterns more effectively. Nonetheless, the relevant question this work investigates is therefore how much performance can be recovered when data must remain distributed due to privacy, bandwidth, or logistical constraints.

Beyond performance, FedCKA may also reduce communication volume by transmitting fewer personalized parameters, but this work does not provide a systematic bandwidth analysis. Therefore, the communication benefit remains a deployment implication rather than an empirically validated result.

7. Limitations

The primary limitation of this study is that the evaluation is restricted to a single detector, dataset, and domain split. The domain split is also highly imbalanced, meaning that the observed gains for smaller domains may depend on the specific scene distribution and validation composition. Differ-

ent detector tuning, datasets, domain splits, or CKA masking thresholds may yield different results.

A second limitation is that the framework assumes supervised local training, which requires reliable client-side labels or a labeling pipeline. This is a strong assumption, as newly collected IV data does not naturally include high-quality 3D bounding boxes. Although offline autolabelling on idle vehicles could provide pseudo-labels, this work does not evaluate how pseudo-label noise would affect federated training.

Another limitation is that although the CKA computation is relatively lightweight compared to model training, it still incurs additional computational costs for the client. It also requires flattening intermediate representations into two-dimensional matrices, which may simplify the original positional information. This flattening may discard part of the original spatial or sequential structure, even though the resulting representation was still useful for layer selection in this study.

8. Conclusion and Future Work

This work studied federated learning for data-heterogeneous 3D object detection under shifts in weather, time, and location, showing that partial personalization generally outperforms a single consensus model on this benchmark. The findings suggest that representation similarity is a useful criterion for balancing this global generalization and local adaptation. For real-world IV deployment, cross-domain robustness and local domain adaptation are both vital, and these findings may help clarify which model layers are best suited for each.

Future work should test whether the observed benefits hold across different detectors, datasets, domain definitions, and types of heterogeneity. Methodologically, FedCKA could be extended with adaptive thresholds, temporally smoothed masks, unsupervised domain clustering based on CKA scores, and further analysis of mask stability across domain batches and batch sizes.

A separate direction is to evaluate whether reduced parameter transmission enables more frequent communication, whether masking reduces global drift, and whether FedCKA remains effective under pseudo-labeled local training from offline autolabelling pipelines.

Finally, although FedCKA may reduce the amount of model information transmitted, it does not, by itself, provide formal privacy guarantees. Combining CKA-guided masking with secure aggregation or differential privacy is therefore an important direction for privacy-preserving deployment.

Overall, this work shows that representation similarity can provide a principled basis for deciding what should remain shared and what should become client-specific in federated 3D object detection.

Acknowledgements

I would like to express my sincere gratitude to my daily supervisor, Ali, for his invaluable guidance throughout this project. I am deeply grateful for our regular discussions, which not only helped me find a clear direction early on but also provided a space to explore fascinating technical topics and benefit from his insights. His constant availability, practical advice, and encouragement were essential in navigating the more challenging stages of this research.

I am also very grateful to my formal supervisor, Holger, for his pivotal role at both the beginning and the end of this journey. His guidance was crucial in helping me find my initial research direction, and his sharp feedback in the final stages greatly improved the clarity, presentation, and overall framing of this thesis.

Additionally, I would like to thank the Delft AI Cluster (DAIC) [6] for providing the computational resources. Access to this GPU infrastructure was essential for training and evaluating the federated 3D object detection models.

Finally, I want to thank my friends, family, and peers for their support, patience, and encouragement throughout this work.

References

- [1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. [2](#), [5](#), [6](#), [12](#)
- [2] Till Beemelmans, Quan Zhang, Christian Geller, and Lutz Eckstein. MultiCorrupt: A multi-modal robustness dataset and benchmark of lidar-camera fusion for 3D object detection. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 3255–3261, 2024. [2](#), [5](#)
- [3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#)
- [4] Jiaqi Cai and Yiquan Wu. Three-dimensional object detection for autonomous driving via deep learning: A review. *Engineering Applications of Artificial Intelligence*, 161:112238, 2025. [2](#)
- [5] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 2089–2099, 2021. [2](#), [5](#), [6](#), [12](#)
- [6] Delft AI Cluster (DAIC). RRID: SCR.025091, 2024. [9](#)
- [7] Yinpeng Dong, Caixin Kang, Jinlai Zhang, Zijian Zhu, Yikai Wang, Xiao Yang, Hang Su, Xingxing Wei, and Jun Zhu. Benchmarking robustness of 3D object detection to common corruptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1022–1032, 2023. [1](#)
- [8] Yingying Gao, Xin He, and Yan Chen. Personalized federated learning algorithm based on information content model customization. pages 834–838, 2025. [3](#), [4](#), [5](#), [6](#), [12](#)
- [9] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory: 16th International Conference (ALT 2005)*, pages 63–78. Springer, 2005. [3](#)
- [10] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 76–92. Springer, 2020. [2](#)
- [11] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. [2](#)
- [12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 5132–5143, 2020. [1](#), [2](#)
- [13] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3519–3529, 2019. [3](#)
- [14] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, pages 429–450, 2020. [1](#), [2](#)
- [15] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-iid features via local batch normalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. [1](#), [2](#), [5](#), [6](#), [12](#)
- [16] Ying Li, Xingwei Wang, Rongfei Zeng, Praveen Kumar Donta, Ilir Murturi, Min Huang, and Schahram Dustdar. Federated domain generalization: A survey. *Proceedings of the IEEE*, 113(4):370–410, 2025. [2](#)
- [17] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C. Knoll. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. *IEEE Transactions on Intelligent Vehicles*, 9(11):7138–7164, 2024. [5](#)
- [18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017. [1](#), [2](#), [5](#), [6](#), [12](#)
- [19] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *Proceedings of the 36th In-*

- ternational Conference on Machine Learning (ICML), pages 4615–4625, 2019. [2](#)
- [20] Trong Binh Nguyen, Duong Minh Nguyen, Jinsun Park, Viet Quoc Pham, and Won-Joo Hwang. Federated domain generalization with data-free on-server matching gradient. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. [2](#), [12](#)
- [21] Jiaming Pei, Wenxuan Liu, Jinhai Li, Lukun Wang, and Chao Liu. A review of federated learning methods in heterogeneous scenarios. *IEEE Transactions on Consumer Electronics*, 70(3):5983–5999, 2024. [2](#)
- [22] Krishna Pillutla, Kshitiz Malik, Abdelrahman Mohamed, Michael Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 17716–17758, 2022. [12](#)
- [23] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12116–12128, 2021. [3](#)
- [24] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 2020. [3](#)
- [25] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019. [1](#)
- [26] Rishub Tamirisa, Chulin Xie, Wenxuan Bao, Andy Zhou, Ron Arel, and Aviv Shamsian. FedSelect: Personalized federated learning with customized selection of parameters for fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29485–29494, 2024. [2](#), [3](#), [4](#), [5](#), [6](#), [12](#)
- [27] Hai Wang, Junhao Liu, Haoran Dong, and Zheng Shao. A survey of the multi-sensor fusion object detection task in autonomous driving. *Sensors*, 25(9):2794, 2025. [2](#)
- [28] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer: Towards fast and robust 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18268–18278, 2023. [5](#), [11](#)
- [29] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, pages 3320–3328, 2014. [1](#)
- [30] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5824–5836, 2020. [2](#), [5](#), [6](#), [12](#)
- [31] Peng Zhang, Xin Li, Xin Lin, and Liang He. A new literature review of 3D object detection on autonomous driving. *Journal of Artificial Intelligence Research*, 82:973–1015, 2025. [2](#)
- [32] Xinyu Zhang, Weiyu Sun, and Ying Chen. Tackling the non-iid issue in heterogeneous federated learning by gradient harmonization. *IEEE Signal Processing Letters*, 31:2595–2599, 2024. [2](#), [6](#), [12](#)
- [33] Liang Zhao, Xuan Li, Xin Jia, and Lulu Fu. Personalized FedM2former: An innovative approach towards federated multi-modal 3D object detection for autonomous driving. *Processes*, 13(2), 2025. [1](#)
- [34] Tianyue Zheng, Ang Li, Zhe Chen, Hongbo Wang, and Jun Luo. AutoFed: Heterogeneity-aware federated multimodal learning for robust autonomous driving. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2023. [1](#)

Appendix Overview

This appendix provides supporting material for the experimental setup and analysis. Appendix A details the domain imbalance. Appendix B lists the detector configuration. Appendix C describes the federated implementation. Appendix D explains the CKA computation. Appendix E reports additional cross-domain matrices. Appendix F analyzes masking sensitivity and selected layer types.

A. Dataset Imbalance Analysis

This appendix provides additional details on the domain imbalance in the federated multi-domain setup. Due to the large number of class-domain combinations, Figure 6 is best interpreted at high magnification.

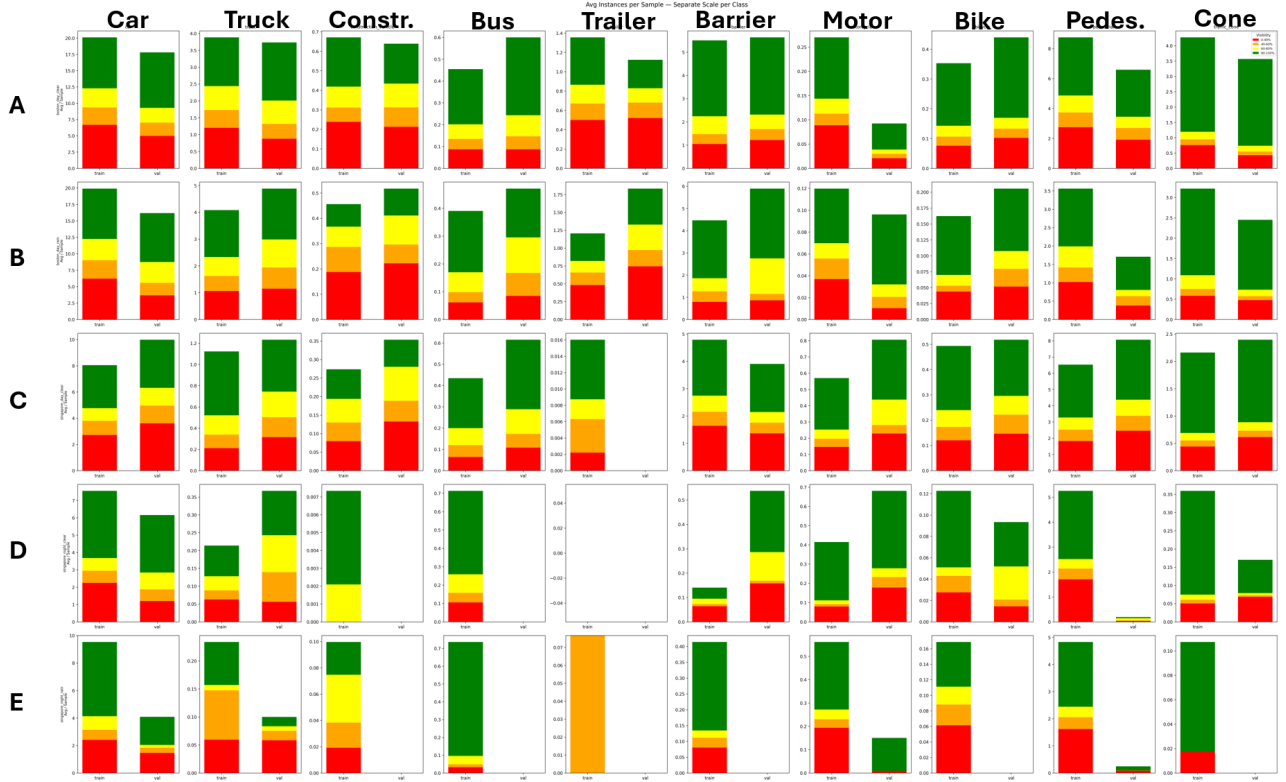


Figure 6. Dataset imbalance and visibility distribution per object class for the five federated domains A–E. Each row corresponds to one domain, and each column to one nuScenes object class. For every class-domain pair, the stacked bars show the average number of annotated instances per sample in the training and validation splits. In the leftmost column, it is clear that in Boston, domains A and B have twice as many cars on average as domains C, D, and E in Singapore. The colors indicate the nuScenes visibility level of the annotations: green denotes the highest visibility, yellow lower visibility, orange limited visibility, and red the hardest visibility cases. Empty bars indicate that no annotations of that class are present in the corresponding split. The figure shows that the federated domains differ not only in the number of available scenes but also in class frequency, train-validation composition, and object visibility distribution, all of which contribute to the heterogeneous evaluation setting. Constr.=Construction Vehicle, Pedes.=Pedestrian.

B. Configuration File

All experiments are based on the official [CMT codebase](#) [28] and follow its original environment and dataset setup. This includes the corresponding OpenMMLab installation, nuScenes preprocessing, and directory structure expected by the official implementation. Given this setup, the [configuration file](#) specifies the experimental settings used in this work. The configuration is compatible with `mmdetection3d v1.0.0rc5`. All experiments were run on two NVIDIA A40 GPUs with a batch size of 16 per GPU. Training required approximately three days per experiment. All other project files can be found at this [repository](#).

C. Federated Details

C.1. Shared Federated Training Protocol

To ensure a fair and consistent comparison across all evaluated methods, the benchmarked federated strategies share a standardized training schedule. The training process spans 40 communication rounds. During each round, each client performs exactly one local training epoch on its respective domain-specific dataset (Domains A through E). This setting is chosen to limit the client drift. All local training utilizes the configuration file provided in Appendix B. Furthermore, for methods that utilize weighted aggregation, client updates are weighted in proportion to the number of training scenes in their local datasets; e.g., Domain A contains 265 training scenes, while Domain E contains 13. All methods use the same detector architecture, optimizer, data preprocessing, and evaluation protocol, so that differences in performance can be attributed to the federated strategy rather than to changes in the underlying training setup.

C.2. Aggregation and Personalization Baselines

As outlined in Section 3, several established federated strategies are replicated. The evaluated baselines are divided into globalization methods, which aim to learn one shared model, and personalization methods, which retain selected client-specific parameters. The globalization methods include FedAvg [18], FedDyn [1], and PCGrad [30, 32], while the personalization methods include FedBN [15], FedRep [5], FedSelect [26], and FedMC [8]. These baselines utilize the common training schedule described above. Further details regarding the implementation of each are as follows.

FedAvg aggregates all shared model parameters through a scene-count-weighted average after each communication round. This method serves as the standard global aggregation baseline.

FedDyn uses the same local training schedule as FedAvg but adds a dynamic regularization term to reduce client drift. In this implementation, FedDyn uses a regularization coefficient of $\alpha = 0.01$.

PCGrad is implemented by projecting conflicting client update directions prior to aggregation. Batch-normalization parameters and task-head parameters are excluded from the projection because their inclusion led to unstable behavior in preliminary experiments.

For FedBN, all BN parameters are kept from the global aggregation. This means that each client retains its own batch-normalization statistics and affine BN parameters, while the remaining model parameters are aggregated globally.

For FedRep, the final shared convolution and the regression heads are excluded from global aggregation. These layers are kept client-specific, while the remaining representation layers are shared across clients.

For FedSelect and FedMC, a maximum masking threshold of 0.4 was used, which is at the midpoint of the suggested range in the paper [26], with rounds 11–30 each gradually personalizing by 0.02. This schedule prevents the masking budget from being applied abruptly and keeps the total amount of personalized parameters comparable across the two parameter-personalization baselines. FedSelect did not use LocalAlt but used LocalSim from the foundational work [22], as the LocalAlt method decreased validation performance in preliminary experiments and required more manual tuning to determine the balance between local and global training. The LocalSim method used a gamma scaling of 0.01 for the global parameters. FedMC follows the same masking schedule but selects personalization parameters based on its information-content criterion rather than on update similarity.

FedOMG [20] was tested during development but was excluded from the final benchmark. In this setup, the equal-client-size assumption was difficult to reconcile with the strongly imbalanced domain split. Without additional normalization, the method overemphasized the smallest-domain gradients from domains D and E; with normalization and clipping, the method became substantially less comparable to its original formulation. Therefore, it is not included in the final results table.

C.3. FedCKA Aggregation and Mask Construction

Unless stated otherwise, the FedCKA results in the main benchmark use threshold-based masking with $\tau = 0.25$, selected from the sensitivity analysis in Appendix F.1. For our evaluated FedCKA approach, the aggregation strategy is divided into two distinct phases. For the first 10 communication rounds, the server employs standard Federated Averaging (FedAvg) to establish a robust, generalized global baseline model. Starting from Round 11, the server transitions to the FedCKA framework.

The FedCKA masking mechanism leverages Centered Kernel Alignment (CKA) to regulate parameter sharing. At the end of a round, the algorithm computes the layer-wise CKA similarity between the previous global model and each client’s locally updated model. This CKA computation is performed on a local subset of the client data and does not require ground-truth labels because only intermediate activations are compared. Using the dynamic sparsity mode with a CKA threshold or by masking a set percentage of parameters, entire layers exhibiting the lowest similarity scores are flagged for personalization,

masked out, and retained locally. Masking is applied at the layer level rather than at the individual-parameter level. The mask is cumulative across communication rounds: once a layer is selected for personalization, it remains private for the rest of the training.

During transmission to the server, each client contributes only the parameters of its unmasked layers to the global aggregation. The server aggregates the available client parameters using the same scene-count weighting as in the other aggregation methods. If all clients personalize a specific layer, the server retains the previous round’s global weights for those parameters to preserve the global baseline. After aggregation, the server broadcasts the updated unmasked layers to each client. Each client then overwrites these unmasked layers with the received global parameters, while its masked layers remain local.

C.4. Repeated Runs and Uncertainty Estimates

Because full repeated training is computationally expensive, the reported results should be interpreted primarily as benchmark-level evidence rather than precise statistical estimates. This is especially relevant for the smallest validation domain, E, where only three scenes are available, and for threshold ablations whose average NDS values differ by less than one percentage point. Therefore, the main conclusion is not that a specific threshold is universally optimal, but that CKA-guided partial personalization consistently occupies a favorable region between full global aggregation and fully local training.

Selected FedCKA threshold settings were repeated to estimate the approximate scale of run-to-run variation. These repeated runs should not be interpreted as full confidence intervals because the computational cost of each full federated experiment (6 days of GPU time on an NVIDIA A40 card) prevented exhaustive repetition across all baselines and domains. Instead, they provide a practical indication of the noise level in the reported benchmark.

Table 5. Repeated-run uncertainty estimates for selected FedCKA threshold settings. Values report the mean average NDS and the observed variation across repeated runs.

Setting	Runs	Mean average NDS	Observed range
$\tau = 0.25$	2	0.605	0.009
$\tau = 0.50$	2	0.600	0.003
$\tau = 0.75$	2	0.596	0.003

The observed variation indicates that small differences between nearby threshold settings should not be over-interpreted. For this reason, the sensitivity analysis is used to support the broader conclusion that partial CKA-guided personalization is beneficial, rather than to claim that one threshold value is universally optimal.

C.5. Optimization and Metric Dynamics

Figures 7 and 8 provide additional insight into the optimization behavior of the evaluated federated strategies. Figure 7 shows the total training loss for the client models corresponding to Domains A and D. These domains are selected to represent two contrasting cases: Domain A is the largest and richest local domain, while Domain D is a smaller nighttime domain. Across both domains, globalization methods exhibit pronounced loss spikes at the start of several communication rounds. These spikes indicate that each aggregation step temporarily moves the client away from its locally adapted optimization state. The spikes are most pronounced in earlier rounds and gradually fade as training progresses, suggesting that the global and local optimization states become more aligned over time. However, the effect remains visible with static personalization methods such as FedBN and FedRep, indicating that keeping only batch-normalization or fixed-head layers private does not fully prevent aggregation-induced disruption.

In contrast, parameter- and layer-personalization methods such as FedMC, FedSelect, and FedCKA show smoother training trajectories after personalization begins. The same qualitative behavior is observed across the individual loss components, but it is most clearly amplified in the bounding-box regression loss shown in Figure 8. For Domain D, several personalized federated methods even reach lower training loss and bounding-box regression loss than the own-domain-only model. This should not be interpreted as direct evidence of better generalization, but it suggests that shared training can provide useful optimization benefits for smaller domains while personalization limits the disruptive effect of aggregation. Overall, the loss behavior indicates that aggregation-induced disruption is especially pronounced in the detector’s geometric localization component.

Figures 9 and 10 further decompose the validation performance into mAP and the five nuScenes true-positive error components: mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error

(mAOE), mean Average Velocity Error (mAVE), and mean Average Attribute Error (mAAE). For these metrics, higher values are better for mAP, while lower values are better for the five error terms. The largest differences among the strongest methods are evident in mAP, mATE, mAOE, and mAVE, indicating that the main performance gap lies in localization, orientation, and velocity estimation. In Domain A, FedCKA remains closest to the own-domain-only model on these components, which is relevant because Domain A contains the most training scenes and provides a strong local reference model. This suggests that FedCKA preserves much of the useful local geometric specialization of the strong local model while still benefiting from shared federated training.

The Domain D metric breakdown provides a complementary perspective. Unlike Domain A, Domain D is smaller and more shifted, so the own-domain-only model is not necessarily the strongest reference. In this case, the relevant behavior is whether a method can preserve local adaptation while still transferring useful information from other domains. The metric decomposition shows that partial-personalization methods better balance this trade-off than purely global aggregation, particularly for localization-related components. This supports the interpretation that retaining domain-sensitive parameters locally can reduce aggregation-induced disruption while still allowing shared parameters to benefit from federated training.

These curves and metric decompositions should be interpreted as supporting evidence rather than as primary evidence of final performance. Training loss is client-specific and does not replace validation metrics such as mAP, NDS, or CAP. Similarly, the metric breakdowns are intended to explain where the main performance differences arise, not to replace the aggregate benchmark comparison. Nevertheless, the reduced round-boundary loss spikes and the improved localization-related validation metrics are consistent with the main benchmark results: methods that retain domain-sensitive parameters locally are less disrupted by repeated aggregation and achieve stronger performance in heterogeneous domains.

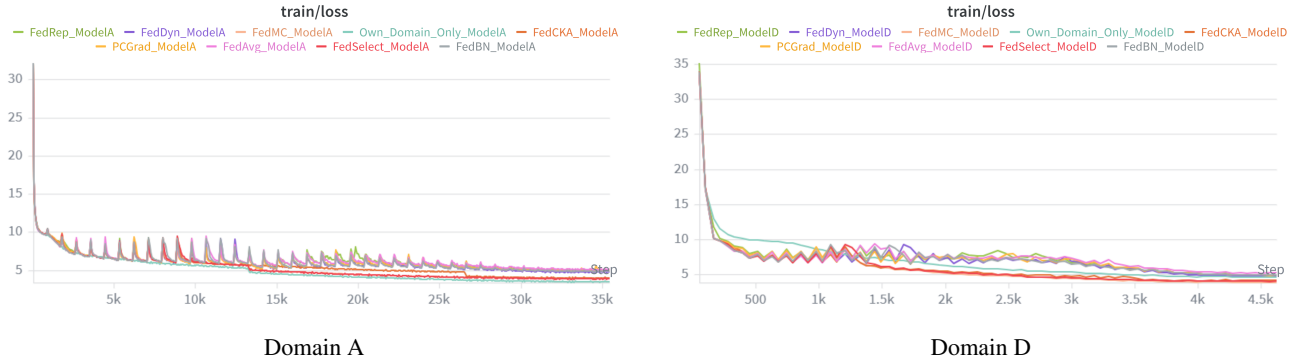


Figure 7. Total training loss curves for the client models corresponding to Domains A and D across the evaluated federated strategies. In both domains, globalization methods show visible loss spikes at the start of several communication rounds, indicating that aggregation perturbs the locally adapted client model. These spikes are most pronounced in earlier rounds and gradually fade as training progresses. Static personalization methods such as FedBN and FedRep reduce this effect only partially. Parameter- and layer-personalization methods such as FedMC, FedSelect, and FedCKA show smoother trajectories after personalization begins, suggesting that retaining domain-sensitive parameters locally reduces round-to-round optimization disruption. Since these curves show training loss for individual clients, they serve as qualitative evidence of optimization and should be interpreted alongside the validation results in Table 2.

C.6. Checkpointing and Resume Logic

Given the high computational demands and duration of the multi-domain training pipeline, the implementation incorporates an automated state-detection mechanism across all evaluated methods. The scripts detect the most recently successfully merged round or the individually completed client epoch, allowing training to resume seamlessly after job timeouts or cluster interruptions. This resume logic only affects job continuation after interruptions and does not alter the optimization procedure, aggregation rule, or evaluation protocol. To manage disk space efficiently, large model checkpoints (`.pth` files) older than two communication rounds are systematically purged. However, for strategies like FedCKA, the lightweight binary masks dictating the personalized client layers are permanently retained for historical visualization and analysis. Final reported results are taken from completed round-40 checkpoints.

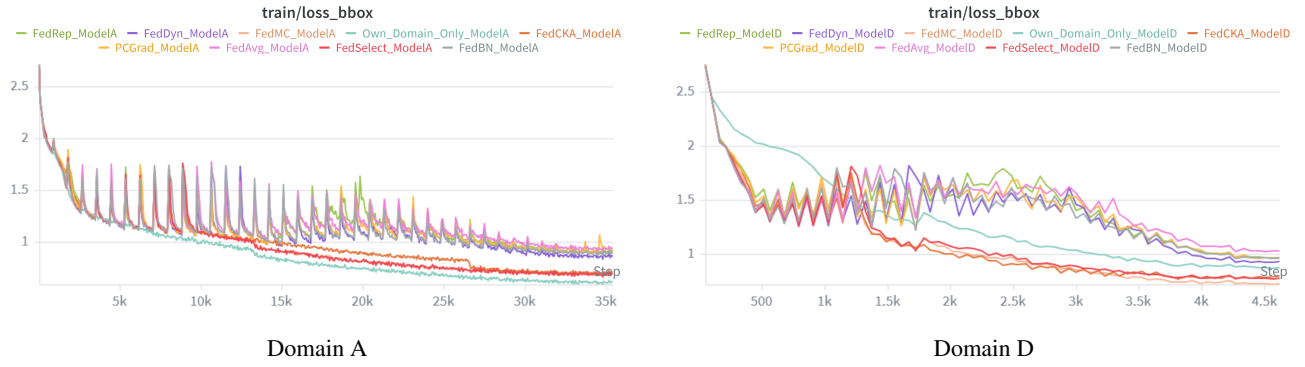


Figure 8. Bounding-box regression loss (loss_bbox) for the client models corresponding to Domains A and D across the evaluated federated strategies. The same round-boundary disruption observed in the total training loss is present across loss components, but is most clearly amplified in the bounding-box regression term. The spikes are strongest in earlier rounds and fade as training progresses, suggesting increasing alignment between local and aggregated model states. For Domain D, several personalized federated methods reach lower training loss than the own-domain-only model, indicating that shared training can provide optimization benefits for smaller domains. This should be interpreted as qualitative evidence of optimization rather than direct evidence of generalization performance.

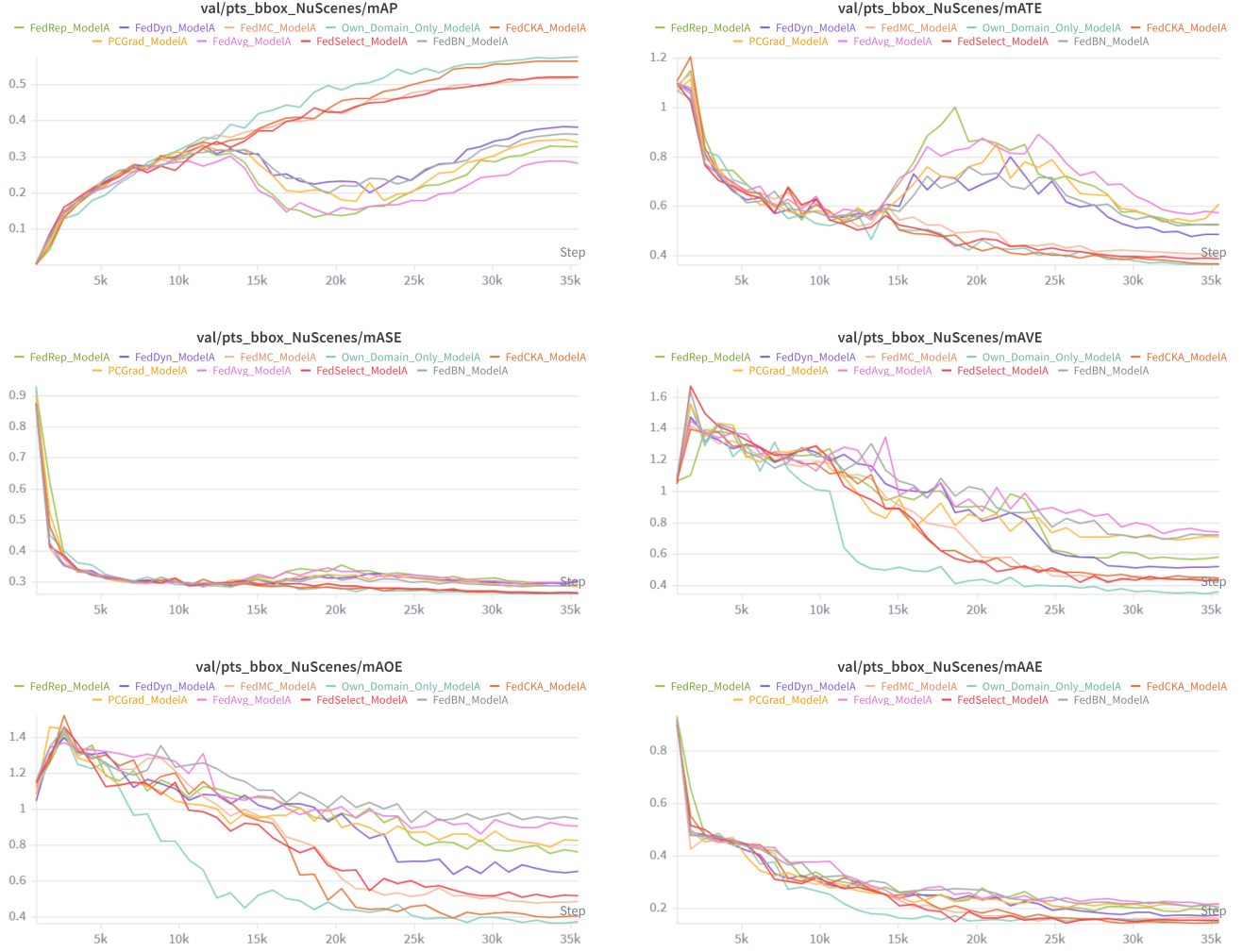


Figure 9. Validation metric breakdown for Domain A across the evaluated methods, including mAP and the five nuScenes true-positive error components. Higher values are better for mAP, while lower values are better for mATE, mASE, mAOE, mAVE, and mAAE. Domain A represents the largest and richest local domain, so closeness to the own-domain-only model indicates preservation of useful local specialization. The largest differences among the strongest methods are evident in mAP, mATE, mAOE, and mAVE, suggesting that the main gains are attributable not only to object detection quality but also to localization, orientation, and velocity estimation. FedCKA remains closest to the own-domain-only model across these components, indicating that it preserves much of the strong local-model behavior while still benefiting from shared federated training.

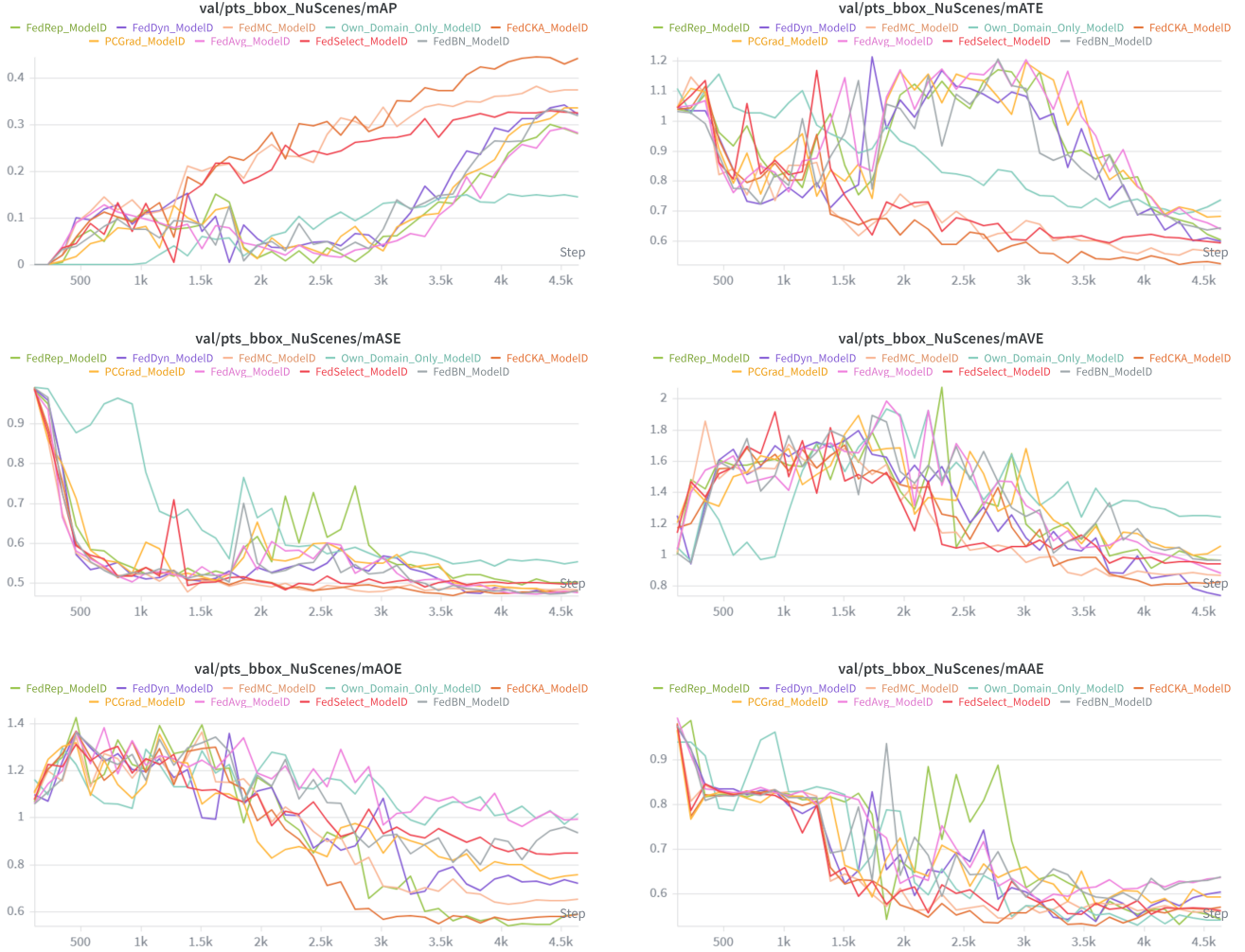


Figure 10. Validation metric breakdown for Domain D across the evaluated methods, including mAP and the five nuScenes true-positive error components. Higher values are better for mAP, while lower values are better for mATE, mASE, mAOE, mAVE, and mAAE. Domain D represents a smaller, more shifted nighttime domain, making it a useful contrast to the larger, richer Domain A. The metric decomposition shows how the evaluated methods differ not only in detection quality but also in localization, orientation, and velocity estimation. The stronger performance of partial-personalization methods on several of these components supports the interpretation that retaining domain-sensitive parameters locally helps balance local adaptation and shared knowledge transfer in more challenging domains.

D. CKA implementation details

To allow for the CKA computation, all intermediate representations must be normalized to a 2D matrix $X \in \mathbb{R}^{n \times d}$ where n is the number of observations and d is the feature dimension. The 4D convolutional activations ($B \times C \times H \times W$) are permuted and flattened to $(B \cdot H \cdot W) \times C$. This dynamically treats the spatial locations as independent observations per sample ($n = H \cdot W$) and the channels as features ($d = C$). CKA, therefore, measures whether the two models produce similar channel-wise feature geometry over the same spatial locations.

Similarly, 3D attention sequences ($B \times N \times D$) are reshaped to $(B \cdot N) \times D$. This preserves the embedding dimension while converting the token dimension into comparable observations, which then measures whether the same tokens are organized similarly in representation space across two models. This way, CKA can measure representation similarity across layers with different activation structures.

For linear CKA, the Frobenius norm of the centered sample Gram matrix can be computed equivalently through the centered feature covariance matrix, avoiding explicit construction of the $n \times n$ Gram matrix. By using this $d \times d$ covariance matrix instead, the similarity score can be calculated with significantly reduced complexity and memory overhead.

Furthermore, the original CKA implementation is not well suited to sparse layer structures, as converting them to dense matrices can cause out-of-memory errors. To address this, the proposed implementation bypasses dense tensor allocation by computing the covariance components only on the spatially intersecting active voxels between models and mathematically adjusts the algebraic centering using the total theoretical grid volume. This means that only voxel locations active in both compared representations contribute to the covariance estimate, while the centering term is corrected as if it were defined over the full sparse grid. This avoids treating missing sparse locations as explicitly stored dense zeros.

These three implementation details allow the CKA to be computed on the CPU for any number of samples in the evaluation batch on the benchmark used.

E. Additional Cross-Domain Results

This appendix contains additional cross-domain result matrices for the evaluated federated methods. Only methods for which client-specific final models were retained are shown. Globalization methods such as FedAvg, FedDyn, and PCGrad produce a single shared model after aggregation, so their cross-domain matrices contain identical rows and are less informative. Results are shown in Tables 6, 7, 8, and 9.

Table 6. Cross-domain generalization matrix for **FedRep**. Each entry reports fair NDS. The diagonal shows performance when the client model is evaluated on its own domain.

	D-A	D-B	D-C	D-D	D-E
M-A	0.40	0.40	0.40	0.33	0.38
M-B	0.39	0.41	0.39	0.33	0.36
M-C	0.38	0.37	0.42	0.36	0.38
M-D	0.37	0.36	0.40	0.39	0.40
M-E	0.34	0.33	0.36	0.32	0.38

Table 7. Cross-domain generalization matrix for **FedSelect**. Each entry reports fair NDS. The diagonal shows performance when the client model is evaluated on its own domain.

	D-A	D-B	D-C	D-D	D-E
M-A	0.59	0.54	0.51	0.44	0.38
M-B	0.50	0.57	0.44	0.42	0.37
M-C	0.43	0.37	0.58	0.52	0.49
M-D	0.32	0.28	0.41	0.46	0.38
M-E	0.29	0.27	0.36	0.38	0.38

Table 8. Cross-domain generalization matrix for **FedBN**. Each entry reports fair NDS. The diagonal shows performance when the client model is evaluated on its own domain.

	D-A	D-B	D-C	D-D	D-E
M-A	0.41	0.41	0.41	0.31	0.33
M-B	0.40	0.41	0.41	0.33	0.32
M-C	0.40	0.38	0.43	0.33	0.34
M-D	0.36	0.34	0.39	0.34	0.34
M-E	0.34	0.33	0.36	0.33	0.35

Table 9. Cross-domain generalization matrix for **FedMC**. Each entry reports fair NDS. The diagonal shows performance when the client model is evaluated on its own domain.

	D-A	D-B	D-C	D-D	D-E
M-A	0.58	0.54	0.51	0.45	0.35
M-B	0.54	0.56	0.47	0.42	0.36
M-C	0.46	0.42	0.57	0.50	0.47
M-D	0.36	0.35	0.43	0.44	0.46
M-E	0.27	0.25	0.29	0.26	0.26

F. FedCKA Masking Sensitivity and Layer Analysis

This appendix analyzes the masking behavior of FedCKA in more detail. First, it evaluates the sensitivity to percentage-based and threshold-based masking. It then examines which layers are selected for personalization, whether the dynamic masks remain stable across runs, and how the selected parameters are distributed across architectural layer types.

F.1. Sensitivity to Masking Strategy and Threshold

To determine the optimal threshold and the stability of the proposed approach, multiple different thresholds for both the percentage-based masking ρ and the threshold-based masking τ have been tested. See Table 10 for the full results.

The table shows that complete aggregation and complete personalization are both suboptimal. Standard FedAvg, corresponding to $\tau = 0.00$, achieves the lowest average NDS of 0.371, while own-domain training, corresponding to $\tau = 1.00$, reaches 0.577. The best FedCKA settings fall between the extremes, achieving an average NDS of 0.609 while masking only about 1% of the parameters. Therefore, the optimal strategy is not to personalize many layers, but to keep most of the network shared and only exclude the most representationally divergent layers from aggregation.

Threshold-based masking also performs better than percentage-based masking. The fixed masking ratios $\rho = 0.10$ and $\rho = 0.40$ reach average NDS scores of 0.536 and 0.575, respectively, but remain below the best threshold-based settings. This suggests that selecting the correct layers based on their CKA score is more important than enforcing a fixed masking budget. The optimal masking threshold is difficult to determine precisely because the results are subject to run-to-run variation. This is visible in the duplicate runs for $\tau = 0.25$, $\tau = 0.50$, and $\tau = 0.75$, where the average NDS varies by up to 0.009. Running more experiments could reduce this uncertainty, but each row requires approximately 140 GPU hours on an A40, making a

more exhaustive sensitivity analysis computationally infeasible for this study. Therefore, $\tau = 0.25$ is chosen for the main experiments, as it achieves the highest average NDS while allowing FedCKA to personalize beyond the first decoder block.

Table 10. FedCKA sensitivity analysis for percentage-based and threshold-based masking. Each entry reports the NDS / final masking percentage (in round 40). Each row represents the results of the experiment for one run. The best three runs for each domain are highlighted. The weighted averages are given in the average column. $\tau = 1.00$ is equal to running the domain models on their own domain only, and $\tau = 0.00$ is equal to running standard FedAvg.

Masking	Average	Domain A	Domain B	Domain C	Domain D	Domain E
<i>Percentage-based masking</i>						
$\rho = 0.01$	0.566 / 1.19%	0.613 / 1.11%	0.520 / 1.11%	0.578 / 1.11%	0.472 / 1.11%	0.381 / 1.51%
$\rho = 0.10$	0.536 / 10.68%	0.558 / 10.38%	0.536 / 11.40%	0.549 / 10.37%	0.423 / 10.32%	0.377 / 10.92%
$\rho = 0.40$	0.575 / 41.29%	0.589 / 42.22%	0.571 / 40.20%	0.591 / 40.88%	0.473 / 42.95%	0.360 / 40.22%
<i>Threshold-based masking</i>						
$\tau = 0.00$	0.371 / 0.00%	0.368 / 0.00%	0.360 / 0.00%	0.382 / 0.00%	0.346 / 0.00%	0.334 / 0.00%
$\tau = 0.05$	0.598 / 0.71%	0.605 / 0.71%	0.586 / 0.71%	0.597 / 0.71%	0.553 / 0.71%	0.538 / 0.71%
$\tau = 0.10$	0.594 / 0.71%	0.598 / 0.71%	0.597 / 0.71%	0.598 / 0.71%	0.539 / 0.71%	0.471 / 0.71%
$\tau = 0.15$	0.609 / 0.71%	0.612 / 0.71%	0.598 / 0.71%	0.612 / 0.71%	0.580 / 0.71%	0.468 / 0.71%
$\tau = 0.20$	0.602 / 0.79%	0.614 / 0.71%	0.603 / 0.71%	0.597 / 0.91%	0.547 / 0.91%	0.445 / 0.71%
$\tau = 0.25$	0.609 / 1.23%	0.622 / 0.71%	0.597 / 0.71%	0.612 / 0.71%	0.572 / 3.31%	0.466 / 0.71%
$\tau = 0.25$	0.600 / 5.04%	0.606 / 4.92%	0.590 / 6.32%	0.601 / 0.91%	0.572 / 4.31%	0.468 / 8.73%
$\tau = 0.50$	0.598 / 8.66%	0.620 / 7.99%	0.585 / 7.34%	0.598 / 9.40%	0.522 / 9.40%	0.421 / 9.19%
$\tau = 0.50$	0.601 / 7.00%	0.622 / 6.13%	0.584 / 5.73%	0.602 / 9.35%	0.548 / 6.19%	0.429 / 7.59%
$\tau = 0.75$	0.597 / 3.18%	0.611 / 3.17%	0.582 / 3.17%	0.602 / 3.18%	0.535 / 1.37%	0.415 / 5.03%
$\tau = 0.75$	0.594 / 5.65%	0.602 / 1.57%	0.592 / 7.95%	0.600 / 4.18%	0.509 / 9.35%	0.422 / 5.19%
$\tau = 1.00$	0.577 / 100.00%	0.637 / 100.00%	0.506 / 100.00%	0.621 / 100.00%	0.248 / 100.00%	0.074 / 100.00%

F.2. Interpretation of Selected Layers

Most layers remain quite similar, as they fall within a CKA score range of 0.75 to 1.0, with 1 indicating layers that are representationally very similar. Therefore, the layers selected by FedCKA are not arbitrary parameter groups, but the parts of the CMT architecture whose representations diverge most strongly across domains. The most divergent layers, in order of divergence, are the first decoder block, the range-view (RV) embeddings, and the second decoder block. By only personalizing the first decoder block’s attention and normalization layers, substantial NDS gains can be achieved. This is also the first part where the two modality-specific backbones are combined. Here, different attention matrices can adjust for rain, night, or location differences, while the deeper and shallower layers remain consistent across clients. This indicates that the domain shift is not uniformly distributed across the entire network but is concentrated in a small number of fusion layers.

For $\tau = 0.05, 0.10, 0.15$, the same three layers are personalized for all clients, resulting in a 0.71% personalization. These layers are located in the first transformer decoder block and correspond to the attention module and the first normalization layer:

```
layer=pts_bbox_head.transformer.decoder.layers.0.attentions.0 | cka=0.000267
layer=pts_bbox_head.transformer.decoder.layers.0.attentions.0.attn | cka=0.000267
layer=pts_bbox_head.transformer.decoder.layers.0.norms.0 | cka=0.002430
```

These are followed by the RV embeddings and the second attention block for all runs where the masks exceeded 0.71%. Around $\tau = 0.25$, these layers are also personalized, increasing the total percentage of masked parameters to above 0.71%. Consistently, the Range View embeddings are personalized second, accounting for approximately 0.2% of the total parameters, as can be seen for $\tau = 0.20$. This supports the interpretation that the first multi-modal decoder block is the most domain-sensitive part of the model, whereas the RV embeddings serve as a secondary source of domain-specific variation.

F.3. Stability of Dynamic Masking

This section assesses the empirical stability of the dynamic threshold-based masking rather than providing a formal proof of convergence. To empirically assess the stability of the dynamic threshold-based masking, we analyze the sensitivity of

the final masking topology (Table 10 and Appendix F.1). Because the personalization mask is cumulative—meaning a layer permanently decouples from global aggregation once masked—a primary concern is the risk of runaway personalization. If the system were unstable, it could cascade into permanently masking the majority of the network over successive rounds, negating the benefits of federated knowledge sharing. However, across a range of thresholds ($0.05 \leq \tau \leq 0.75$), the system does not exhibit this masking cascade; instead, the cumulative fraction of personalized parameters remains strictly bounded, predominantly plateauing at or below 10%. Furthermore, multiple independent runs at the same threshold (e.g., $\tau = 0.50$) demonstrate that minor variations in the final mask topology (7.00% vs. 8.66%) result in negligible variance in the final NDS (0.601 vs. 0.598).

Crucially, this stability extends beyond the overall volume of masked parameters to the specific structural patterns that emerge. As detailed in Appendix F.1, the dynamic masking is highly consistent in identifying which layers to permanently isolate. Across different clients and at lower thresholds ($\tau \leq 0.15$), the same three layers—specifically, the attention and normalization operations within the first transformer decoder block—consistently emerge as the most divergent. As the threshold increases, the system predictably targets the Range View (RV) embeddings next. This suggests that the system is not arbitrarily isolating layers due to noise, but repeatedly identifies the same cross-modal fusion layers as most sensitive to the evaluated domain shifts. This helps maintain a mostly shared global structure, rather than collapsing into fully isolated local models.

It should be noted that Table 10 only reports the final masking percentage after round 40. Therefore, the table does not show how the masks evolved during training. A layer that is masked at the end of training may have become dissimilar early in training and remained private throughout, or it may only have crossed the CKA threshold in a later round. This temporal behavior may influence the final performance, since early masking affects the optimization trajectory and the extent of shared training, whereas late masking primarily affects the final specialization. Analyzing the evolution of the mask over communication rounds would therefore be an interesting direction for future work.

F.4. Layer-Type Personalization Analysis

Table 11 details the per-block personalization percentages for the FedCKA configuration with $\tau = 0.5$ (Equation 4). Although $\tau = 0.25$ is used for the main experiments, the $\tau = 0.5$ setting is shown here because it yields a richer masking topology, making the distribution of personalized layer types easier to inspect. Rather than enforcing a fixed percentage of layers to be personalized, FedCKA achieves stronger performance than benchmarked models by personalizing only 7.5% of the parameters for the Boston models (A and B) and 9% for the Singapore models (C, D, E). The table should therefore be interpreted per architectural block: some layer types can show high masking percentages while the total number of personalized parameters remains small. This allows the system to surgically isolate highly domain-specific features, such as range-view (RV) embeddings and shared convolutions, while still sharing the generalized knowledge captured by the rest of the architecture. Because these domain-specific layers are retained locally, the transmission payload sent to the aggregation server is reduced by the corresponding percentage in each communication round, lowering the overall bandwidth requirement compared to standard FedAvg.

Table 11. Layer-wise selected update mask percentages per client model for FedCKA implementation with $\tau = 0.5$. Abbreviations: Img. BB = image backbone, Img. Neck = image neck, Pts. Enc. = point encoder, Pts. BB = point/LiDAR backbone, Pts. Neck = point/LiDAR neck, BEV = bird’s-eye-view embedding, RV = range-view embedding, Trans. = transformer, Sh. Conv = shared convolution, Heads = task heads.

Model	Img. BB	Img. Neck	Pts. Enc.	Pts. BB	Pts. Neck	BEV	RV	Trans.	Sh. Conv	Heads	Total
A	0.0	0.0	0.0	27.6	88.6	0.0	100.0	83.3	100.0	16.5	7.8
B	0.0	0.0	0.0	27.6	88.6	0.0	100.0	49.8	100.0	0.3	7.3
C	0.0	0.0	0.0	41.4	88.6	0.0	100.0	66.7	100.0	16.3	9.4
D	0.0	0.0	0.0	41.4	88.6	0.0	100.0	66.7	100.0	17.5	9.4
E	0.0	0.0	0.0	41.4	88.6	0.0	100.0	49.8	100.0	16.3	9.2