Semantic Segmentation in Federated Learning: clustering and crucial classes boosting

Margherita Soro, Nicola Candellieri, Lorenzo Sibille
Politecnico di Torino
s304667@studenti.polito.it, s314980@studenti.polito.it, s314927@studenti.polito.it

Abstract

Semantic Segmentation is crucial for autonomous self-driving vehicles. A large number of novel methods in this field have been proposed. However, underlying many of them, there's an unrealistic assumption: we can aggregate large-scale labelled data from all clients' devices to a central server. Federated Learning involves training statistical models over remote devices while keeping data localized, introducing novel challenges related to privacy concerns and statistical heterogeneity. Our approach involves synthetic datasets (IDDA and GTA5) and gradually moves towards a realistic federated scenario. We tackle domain adaptation and unlabelled target data challenges by applying the FDA technique for style extraction and performing unsupervised self-training. We experiment style-based clustering to create specialized submodels, improving performances. Additionally, we enhance recall and mIoUs for crucial classes in self-driving setting (pedestrians, streetlights) with different loss reductions. The code is available at `https://github.com/ZeTornio/MLDL_project`

1. Introduction

Semantic segmentation [6], a fundamental task in computer vision, aims to assign a class label to each pixel within an image, enabling fine-grained understanding of visual scenes. This operation finds extensive application in various real-world domains, including autonomous driving, medical image analysis, defect detection, and many others.

In the context of autonomous driving, semantic segmentation plays a crucial role in enabling vehicles to perceive and interpret their surroundings accurately. By segmenting the visual scene into different object categories, such as roads, pedestrians, and vehicles, autonomous vehicles can make informed decisions and navigate safely through complex traffic scenarios.

Although great strides have been made in the recent years, some challenges embedded in semantic segmentation framework persist. Two significant issues include the dependence on high-quality training datasets and the existence of substantial domain gaps between different datasets [6].

One approach to address these challenges involves leveraging customer vehicle data to augment the training dataset with images from diverse domains, encompassing various locations, times, and vehicles. However, privacy concerns often restrict the transfer of data from clients to the server.

To tackle this issue, an integration of semantic segmentation with Federated Learning [11, 15] is possible. Federated Learning is a decentralized paradigm for collaborative machine learning. It enables multiple entities to train a global model without sharing raw data, preserving privacy and addressing the limitations of centralized approaches. By training the model locally on each client device and aggregating the updated parameters, Federated Learning ensures data privacy while leveraging the collective knowledge from distributed devices.

However, Federated Learning introduces other challenges that need to be addressed. Firstly, clients collect data in a non-identically distributed manner, resulting in *statistical heterogeneity* that must be accounted for. Secondly, in realistic scenarios, the images stored on client devices are often unlabeled, necessitating the utilization of unsupervised learning techniques to extract meaningful information [22]. Lastly, the federated scenario involves multiple client devices with diverse hardware configurations, network conditions, and computational capabilities; this *systems heterogeneity* adds complexity to the federated learning process [11].

To address the challenges described, we considered and faced (4) different scenarios, gradually moving toward the most realistic one, covering, first separately and then jointly, the main challenges of semantic segmentation and federated learning. Among the many challenges faced, we focused on two main problems:

(i) Statistical heterogeneity of clients.

(ii) Performance imbalance among different classes.

Figure 1. Frequencies and mIoU among different classes in IDDA training set. mIoU values are computed in a centralized scenario.

To cope with (i) we used a clustering scheme as in [19], that allows the adoption of tailored submodels for customers who are similar to each other, where similarity is measured through low-level features in the Fourier domain. The presence of customized models allows to properly address different customer distributions.

The second problem (ii) arises from the fact that some classes are less frequent than others. Pedestrians, signs, and traffic lights are extremely important classes in an autonomous driving scenario, however, they are also among those with the lowest probability of being correctly classified, as shown in Figure 1. We addressed this problem through the adoption of weighted losses that force the model not to marginalize the least frequent classes. Our main contributions can be summarized as follows:

1. We employ a clustering scheme to create submodels tailored to clients with similar data distributions. This approach allows us to address the diversity present across their datasets effectively.

2. We adopt a specific weighted loss technique that enhances the performance for less frequent and more delicate classes, ultimately improving the overall mean Intersection over Union (mIoU).

2. Related works

Semantic Segmentation Deep neural networks have shown promising results in semantic segmentation, with ResNet [7] and MobileNet [9] being widely used as backbone architectures. ResNet's key contribution lies in integrating residual representation into the CNN network structure, while MobileNet balances accuracy and computational costs, making it suitable for real-time applications.

Among supervised methods, the Fully Convolutional Network (FCN) by Long et al. [14] stands out for introducing end-to-end training for pixel-wise prediction. Another notable approach is U-Net, proposed by Ronneberger et al. [18], which employs a symmetrical encoder-decoder architecture with dense skip connections to enhance feature propagation.

Dilated Convolutional Networks have attracted attention by using gaps in the filter kernel to achieve a larger effective receptive field without sacrificing spatial resolution. One of the most renowned DCNN models for semantic segmentation is the DeepLab series by L. Chen et al [3].

Federated Learning While numerous federated learning (FL) algorithms have been developed for classification tasks, fewer works have specifically focused on the more challenging task of semantic segmentation. The earliest FL approach, FedAvg [15] by HB. McMahan et al., optimizes the local model using individual data and aggregates weights through weighted averaging on the server. This simple approach is effective in homogeneous scenarios but falls short in achieving comparable performance when dealing with non-i.i.d. data.

To address the issue of statistical heterogeneity, FedProx [12] introduces a proximal regularization term that keeps the updated local parameters close to the global model, preventing gradient divergence. Scaffold [10], on the other hand, proposes a variance reduction strategy to correct for drifted local updates.

FedDrive [5], particularly relevant for our experiments, evaluates FL methods on an autonomous driving semantic segmentation dataset. While it focuses on domain heterogeneity, Miao et al. went a step further and recently proposed FedSeg [16], a federated learning approach for addressing class-heterogeneous data. FedSeg utilizes a modified cross-entropy loss to improve local optimization and it incorporates pixel-level contrastive learning to align local pixel embeddings with the global semantic space.

Domain Adaptation In semantic segmentation (SS), the manual process of labeling large datasets with pixel-level labels is both costly and time-consuming. An increasing number of approaches have therefore addressed this issue by training on synthetic data generated in virtual environments. Domain adaptation (DA) techniques have been introduced to handle the differences between the source and target domains. When target data lacks labels, it is referred to as Unsupervised DA (UDA).

The first DA methods attempted to close the gap by measuring domain divergence. W. Wong et al. [8] and numerous other researchers proposed innovative approaches based on adversarial learning, which involves training two competing models: a generator model and a discriminator model.

However, these methods can be complex and challenging to implement. Consequently, there have been introductions of non-trainable style translation algorithms to simplify the process. One notable algorithm is FDA [21], which utilizes Fast Fourier Transform (FFT) to compute the frequency components of each input image. It then substitues low-level frequencies of target images into source images, then reconstructed for training via inverse FFT.

In contemporary approaches, self-learning techniques are employed to generate pseudo-labels from unlabeled target data. Y. Li et al. [13] propose a new bidirectional learning framework that involves two separated modules: image-to-image translation model and segmentation adaptation model. Both models will be motivated to promote each other alternatively, forming a close loop learning cycle and causing the domain gap to be gradually reduced.

Clustering Client clustering is utilized to group clients based on specific criteria. This technique is commonly used to develop personalized models that perform well within a particular subdomain of interest. A framework named the Iterative Federated Clustering Algorithm, proposed by A. Ghosh et al. [4], is introduced. IFCA iteratively estimates the cluster assignments of users and optimizes model parameters for the user clusters through gradient descent.

While clustering can help reduce heterogeneity by identifying domains, it also restricts each cluster model from accessing data and supervision from other clusters. To address this limitation, D. Caldarola et al. [2] propose a method called Cluster-driven Graph Federated Learning. In FedCG, clustering is utilized to tackle statistical heterogeneity, while Graph Convolutional Networks (GCNs) facilitate knowledge sharing across the clusters.

Particularly relevant to the purpose of this paper is the work of D. Shenaj et al. [19], in which a unique federated clustered aggregation scheme based on the clients' visual styles [21] is used in combination with other techniques, such as self-training and knowledge distillation, to cope with Source Free Domain Adaptation

3. Method

This section provides a description of our main contributions: the clustering scheme (Section 3.1) and the weighted loss technique (Section 3.2).

3.1. Clustering Scheme

In the context of a federated setting, FedAvg [15] algorithm involves the central server generating a global model by simply taking the average of the models' parameters trained locally. Given $N' \leq N$ selected clients, let n_i be the number of images contained in client i and ${\omega^t}_i$ the model parameters of client i at time t. The aggregation step can be represented as follows:

$$ {\omega^{t+1}}_{global} = \sum_{i=1}^{N'} \frac{n_i}{n} \omega_i^{t+1}, \quad n = \sum_{i=1}^{N'} n_i \tag{1} $$

However, in real-world scenarios FedAvg's assumption of identical and independent data distribution across clients does not hold true. This implies that a single model may struggle to comprehensively handle these distinct distributions [20]. To overcome this limitation, we make use of the clustering scheme proposed in [19], where clients are clustered according to their style, and a personalized model is built for each cluster.

Firstly, the N clients are partitioned into a collection of non-empty clusters M, with each cluster $m \in M$ containing a group of similar clients. These clusters are computed based on the clients' visual styles s_i, which are represented by $l \times l$ matrices extracted from the center of the amplitude spectrum in the Fourier Domain, as in [21]. We create multiple clustering using k-means, for different values of k and many random initializations, selecting the one giving the best silhouette score.

Once the clusters are created, the goal is to build personalized models by modifying the FL model aggregation process. Instead of aggregating all model parameters equally, according to 1, some parameters are kept cluster specific. These are updated by aggregating only the parameters of the clients belonging to the same cluster. We denote the global parameters as ϕ^t and the cluster specific ones as θ_m^t, $\forall m \in M$, such that $\omega_m^t = \phi^t \cup \theta_m^t$ is the set of parameters defining the submodel related to cluster m. Hence, the parameters are computed as follows:

$$ \phi^{t+1} = \sum_{i=1}^{N'} \frac{n_i}{\sum_{j=1}^{N'} n_j} \phi_i^{t+1} \tag{2} $$

$$ \theta_m^{t+1} = \sum_{i \in m}^{N'} \frac{n_i}{\sum_{j \in m} n_j} \theta_i^{t+1} \tag{3} $$

In particular, the cluster-specific parameters coincide with the parameters of a set of layers L_θ of the neural network. Depending on how much you want to customize the models, you can expand or decrease the size of L_θ, and choose the best set of layers to keep cluster dependant, while considering also memory constraints. In our case, we take classifier's layers as L_θ. During the testing phase, given a client k, the algorithm is as follows:

(i) The style s_k of the client k is extracted.
(ii) The centroid μ_m at the minimum distance from s_k is determined.
(iii) The submodel with parameters ω_m^{t+1} is utilized to predict the labels of the images contained in client k.

	Transforms						Hyperparams			
	-	Random Crop	Normalize (Imagenet)	Random Rotation	Random HorizontalFlip	Color Jitter	$lr = 0.15$	$lr = 0.1$	–	$lr = 0.01$
mIoU [*TR*]	**0.6849**	0.6480	0.6763	0.6724	0.6806	0.6834	0.6562	**0.6637**	0.6625	0.6554
mIoU [*TS1*]	0.5793	0.6010	0.5773	**0.6037**	0.5911	0.5754	0.6028	0.6104	**0.6187**	0.6086
mIoU [*TS2*]	0.4817	**0.4961**	0.4771	0.4954	0.4880	0.4786	0.4624	0.4902	**0.5224**	0.4795

Table 1. Transforms results have been tested for 200 epochs, with $lr = 0.05$, $m = 0.9$, $wd = 0$, $bs = 4$. Regarding the hyperparameters, we report only the results obtained by varying the learning rate while keeping the other hyperparameters constant at the values specified for the transforms; these tests are performed for 400 epochs, using RandomCrop as common transform.

Although it is possible to cluster individual images instead of the client, allowing a finer level of granularity, this would have resulted in communication costs not in line with the goals and constraints of federated learning.

3.2. Weighted Loss Technique

First, let's consider a model prediction x, a $H \times W \times C$ matrix containing for each pixel a value associated the probability of each class, and the corresponding ground truth y, a $H \times W$ matrix containing for each pixel the associated class. We can define the cross entropy loss $\mathcal{L}$, which is a $H \times W$ matrix containing the cross entropy loss per each pixel, as

$$\mathcal{L}(x, y)_{i,j} = -\log \frac{e^{x_{i,j,y_{i,j}}}}{\sum_{k=1}^{C} e^{x_{i,j,k}}}$$

We can then perform different reduction methods, starting from the standard approach which is to perform the mean:

$$\ell_{Mean} = \frac{\sum_{i,j} \mathcal{L}_{i,j}}{H \times W}$$

However this approach gives unbalanced results in terms of mIoUs, due to the high unbalance of classes in the training set. We can try to resolve this problem introducing new simple reductions. We have to define the frequency (4) and the mean loss (5) for each class; we also define here the set of classes appearing in the image (6):

$$f_c = \frac{\sum_{i,j} \mathbb{1}\{y_{i,j} = c\}}{W \times H} \tag{4}$$

$$\ell_c = \frac{\sum_{i,j} \mathcal{L}_{i,j} \cdot \mathbb{1}\{y_{i,j} = c\}}{\sum_{i,j} \mathbb{1}\{y_{i,j} = c\}} \tag{5}$$

$$\mathcal{C} = \{c \in [1, C] : f_c > 0\} \tag{6}$$

The first option is to perform the average of the mean losses of the classes present in the image (7). Another possibility is to give an inverse proportionality with respect to the frequency of each class, in order to give more importance to rare classes (8). The last option is to assign to each class a fixed weight w_c, and perform a weighted average (9).

$$\ell_{MeanPerClass} = \frac{\sum_{c \in \mathcal{C}} \ell_c}{|\mathcal{C}|} \tag{7}$$

$$\ell_{InvFreq}(0 \leq p \leq 1) = \sum_{c \in \mathcal{C}} \ell_c \cdot f_c^{1-p} \tag{8}$$

$$\ell_{wMean}(\overline{w}) = \frac{\sum_{c=1}^{C} \ell_c \cdot w_c}{W \times H} \tag{9}$$

Note that

$$\ell_{InvFreq}(0) = \ell_{Mean} \qquad \ell_{InvFreq}(1) \propto \ell_{MeanPerClass}$$

4. Experiments

To perform our experiments, we use a subset of IDDA dataset [1], split into three different clients' sets:

- *TR*: 24 clients with 25 images each, used for training;
- *TS1*: 12 clients with 10 images each, having samples from the same domain of *TR*;
- *TS2*: 6 clients with 20 images each, having samples from a different domain wrt *TR*.

We will also use a subset of GTA5 dataset [17] to pretrain our model, made of 500 images, referred to as *GTA5*.

We perform our experiments using DeepLabV3 [3], with MobileNetV2 [9] as backbone.

Centralized scenario First, we consider a centralized server containing all the images of *TR*. In this scenario we test different augmentations techniques and hyperparameters as reported in Table 1, obtaining useful results to move towards a federated approach while establishing a centralized baseline. We use for the rest of the experiments $lr = 0.05$, $m = 0.9$, $wd = 0$, $bs = 4$, the best values found in this scenario.

Federated scenario We now test the performance in a federated scenario, using FedAvg as aggregation method, varying the number of clients randomly extracted per round $n_{clients}$, and the number of local epochs performed e. From Table 2 we observe that the best results are achieved with a small $n_{clients}$ and a relatively small e. While keeping

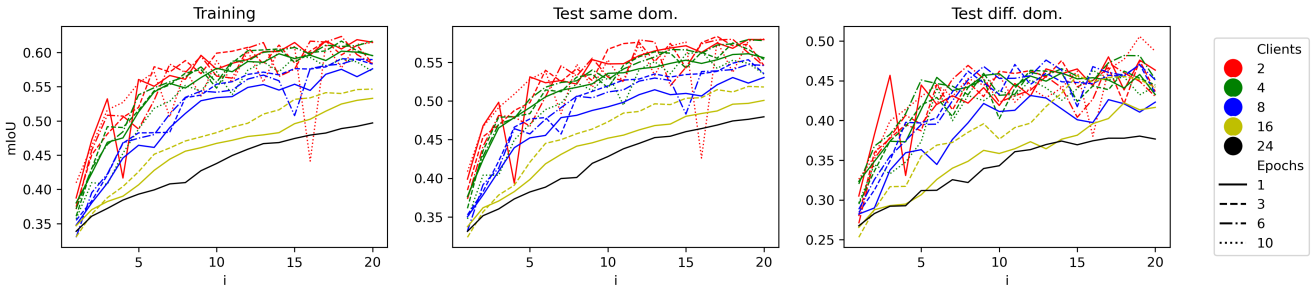

Figure 2. Results for FL; rounds have been normalized, using on x-axis $i = (n_{clients} \cdot e \cdot rounds)/240$.

Clients per round	Local Epochs	Rounds	mIoU [*TR*]	mIoU [*TS1*]	mIoU [*TS2*]
Server	200	1	0.6480	0.6010	0.4961
2	1	2400	0.6168	0.5781	0.4736
2	3	800	0.6212	0.5785	0.4724
2	6	400	0.6184	0.5794	0.4673
2	10	240	**0.6214**	**0.5786**	**0.4898**
2	15	160	0.6093	0.5761	0.4683
2	24	100	0.6035	0.5684	0.4628
4	1	1200	0.6002	0.5592	0.4588
4	3	400	0.6085	0.5686	0.4624
4	6	200	0.6124	0.5773	0.4785
4	10	120	0.5920	0.5591	0.4564
8	1	600	0.5732	0.5280	0.4286
8	3	200	0.5879	0.5469	0.4677
8	6	100	0.5876	0.5447	0.4713
16	1	300	0.5293	0.4979	0.4173
16	3	100	0.5445	0.5173	0.4553
24	1	200	0.4927	0.4767	0.3786

Table 2. We decided to keep $e \cdot rounds \cdot n_{clients}$ constant, to achieve a fair comparison. Since it is more unstable then a centralized approach, we report the average between the best three values obtained each run.

e small could be beneficial in a real scenario to not stress small devices involved, a low $n_{clients}$ higly reduces parallelism, thus a trade-off would be needed. Furthermore, we can see from Figure 2 that increasing $n_{clients}$ achieves more stable (although slightly lower) performances.

Unsupervised Federated scenario We add to our experiments the more realistic assumption of not having labeled data on clients' devices. First, we have to perform pre-training on a big labeled synthetic dataset, in our case *GTA5*. We test different augmentation techniques (Table 3), trying to increase domain adaptation. Among the standard ones, we also test FDA, substituting low frequencies with a random FDA style from *TR*. In our case we achieve the best results with a ColorJitter; however FDA provides good results, that may have been mitigated by the existing similarity between styles in *TR* and *GTA5*.

We can improve our results via self-training: proceeding as in federated learning, but using as labels the pseudo-labels generated by a teacher model, kept in parallel to the student model we are training. In particular, we use as pseudo-labels the 66% most confident predictions per class, together with all the predictions having a confidence > 0.9. Both models at the beginning are the result of pre-training on *GTA5*, then the teacher model is updated every T rounds, cloning the student model at that moment. In Table 5 and Table 4 we report our results: although we manage to improve the initial results with all our parameters, we can see in Figure 3 that on *TR* we have an immediate peak, but while with $e = 2$ it constantly decreases, with $e = 1$ we have an almost stable behaviour. In both cases we can see that the real improvement is on *TS1* and *TS2*, while in *TR* it is minimal.

Transforms	mIoU [*GTA5*]	mIoU [*TR*]	mIoU [*TS1*]	mIoU [*TS2*]
-	0.7548	0.3140	0.3186	0.2574
RandomCrop	0.6940	0.2742	0.2700	0.2151
Normalize(Imagenet)	0.7579	0.3232	0.3172	0.2606
RandomRotation	0.7334	0.3103	0.3162	0.2555
RandomHorizontalFlip	0.7506	0.3122	0.3093	0.2633
ColorJitter	0.7532	**0.3432**	**0.3432**	0.2730
Jitter+Crop	0.6786	0.3165	0.3105	0.2543
Jitter+Rotation	0.7267	0.3213	0.3269	**0.2899**
Jitter+Normalize(Imagenet)	0.7571	0.3382	0.3404	0.2852
FDA(1)	0.7574	0.3306	0.3310	**0.2742**
FDA(3)	0.7593	0.3296	0.3245	0.2606
FDA(5)	0.7581	**0.3312**	**0.3356**	0.2561
FDA(7)	0.7486	0.3204	0.3209	0.2601
FDA(25)	0.7565	0.3233	0.3241	0.2536
FDA(51)	0.7523	0.3091	0.3108	0.2327

Table 3. Results after 200 epochs for different transforms in pre-training our model on *GTA5*. For FDA, the window size is reported between brackets.

Clustering We perform clustering as explained in subsection 3.1. We keep only classifier's layers as cluster specific, testing different values as styles to perform clustering, both

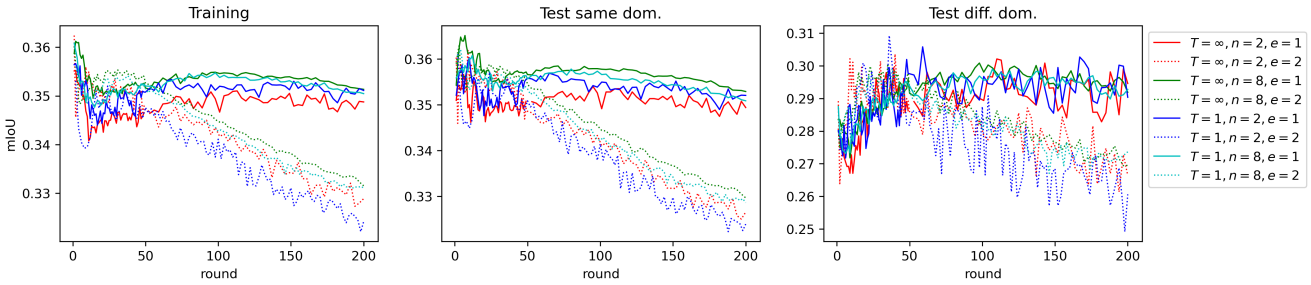

Figure 3. Results for self training, in 200 rounds, starting from Jitter pre-trained model.

Clients per round	Local Epochs	Teacher Update	mIoU [*TR*]	mIoU [*TS1*]	mIoU [*TS2*]
	baseline		0.3338	0.3241	0.2669
2	1	∞	0.3482	0.3502	0.2824
8	1	∞	0.3522	0.3541	0.2834
2	1	1	0.3500	0.3522	**0.2864**
8	1	1	**0.3535**	**0.3561**	0.2824

Table 4. Best results obtained in 120 rounds of self training, starting from a pretrained model on *GTA5* using FDA.

Clients per round	Local Epochs	Teacher Update	mIoU [*TR*]	mIoU [*TS1*]	mIoU [*TS2*]
	baseline		0.3504	0.3433	0.2866
2	1	∞	0.3552	0.3567	0.3021
2	2	∞	**0.3624**	0.3619	0.3036
8	1	∞	0.3612	**0.3652**	0.3009
8	2	∞	0.3602	0.3629	0.3006
2	1	1	0.3566	0.3603	0.3058
2	2	1	0.3531	0.3611	**0.3093**
8	1	1	0.3608	0.3608	0.2990
8	2	1	0.3561	0.3620	0.2991
2	1	4	0.3560	0.3587	0.3049
8	1	4	0.3602	0.3623	0.2977

Table 5. Best results obtained in 200 rounds of self training, starting from a pretrained model on *GTA5* using Jitter.

in supervised and unsupervised scenarios. As we can see in Figure 4, FDA is able to properly capture the styles, clustering together clients belonging to similar domains (represented as *Town_Weather_Car[_User]*).

In supervised learning, as shown in Table 6, we obtain better performances in *TR* and *TS1*, but worse generalization on *TS2*; we have also to consider that we are "distributing" rounds, thus we are training cluster-specific layers for less epochs in proportion. By performing clustering, we achieve a stable growing trend, whereas without clustering, we experience higher peaks but extremely oscillating behavior. In unsupervised scenario, it slightly improved the model pretrained with FDA, while, for Jitter, it achieved results in line with unclustered scenario.

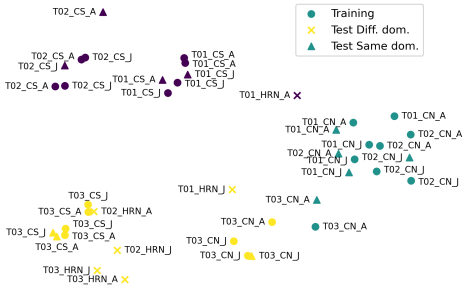

Figure 4. 2D PCA projection of the best clustering achieved on FDA(3).

Scenario		Cluster param	Clients per round	Local Epochs	Teacher Update	mIoU [*TR*]	mIoU [*TS1*]	mIoU [*TS2*]
Supervised		-	2	3	-	0.6172	0.5736	**0.4446**
		FDA(1)	2	3	-	**0.6331**	0.5768	0.3228
		FDA(3)	2	3	-	0.6307	**0.5891**	0.3337
		mean	2	3	-	0.6288	0.5757	0.3161
		-	8	3	-	0.6104	0.5714	0.4398
		FDA(1)	8	3	-	0.6230	0.5576	0.3390
		FDA(3)	8	3	-	0.6259	0.5721	0.3416
		mean	8	3	-	0.6236	0.5682	0.3455
Unsupervised	from FDA	FDA(1)	2	1	∞	0.3556	0.3570	0.2863
		FDA(1)	2	1	1	**0.3600**	**0.3623**	**0.2909**
		FDA(1)	8	1	∞	0.3555	0.3570	0.2767
		FDA(1)	8	1	1	0.3526	0.3538	0.2753
	from Jitter	FDA(1)	2	1	∞	**0.3658**	**0.3749**	**0.3046**
		FDA(1)	2	1	1	0.3638	0.3695	0.3045
		FDA(1)	8	1	∞	0.3591	0.3612	0.2967
		FDA(1)	8	1	1	0.3578	0.3637	0.2985

Table 6. For supervised setting we report the mean of the last 3 measures; 1600 rounds for 2 clients 3 epochs, 400 for 4 clients 3 epochs. For unsupervised, we report the best values achieved in 120 rounds; refer to baselines in 5 and 4. With FDA(1) and mean we obtained 5 clusters, while with FDA(3) they were 3.

Losses As described in subsection 3.2, we explore different reductions for the loss. We refer here with *wMean(a)* to the reduction $\ell_{wMean}(\overline{w})$ in which w_i is 1 for all classes except for pole, light, sign, person and rider for which w_i is a. Although the main goal was to increase recall, since it is better to overestimate vital classes in a real context, we achieved better results also in terms of mIoU, which con-

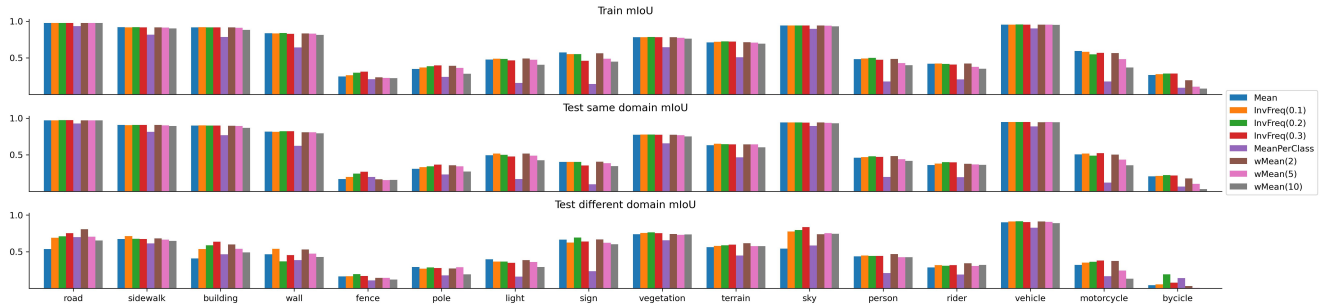

Figure 5. mIoU achieved with different reductions, in a centralized setting, after 200 epochs.

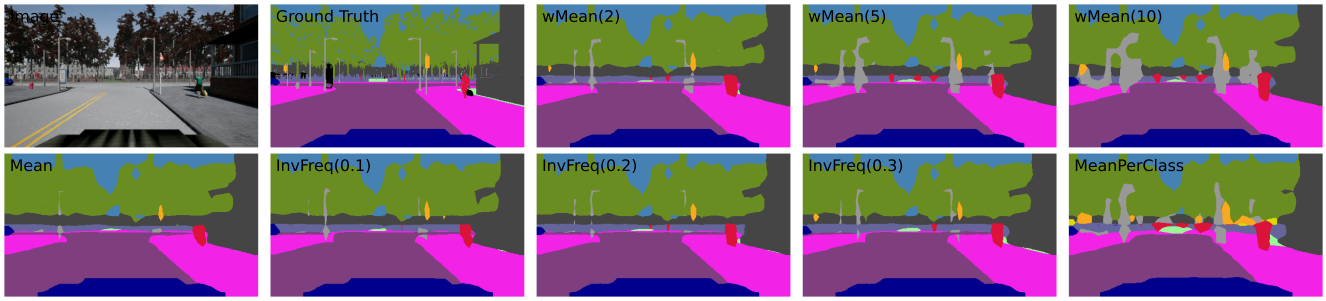

Figure 6. Example of predictions of models trained with different reductions.

Setting	Reduction	mIoU [*TR*]	mIoU [*TS1*]	mIoU [*TS2*]
Centralized	Mean	0.653	0.611	0.463
	InvFreq(0.1)	0.655	0.620	0.506
	InvFreq(0.2)	**0.658**	**0.624**	0.515
	InvFreq(0.3)	0.650	0.623	0.516
	MeanPerClass	0.471	0.456	0.379
	wMean(2)	0.648	0.616	**0.519**
	wMean(5)	0.621	0.598	0.484
	wMean(10)	0.592	0.569	0.453
Federated 2 clients 3 epochs	Mean	0.618	0.586	0.455
	InvFreq(0.1)	0.628	0.596	0.474
	InvFreq(0.2)	**0.629**	**0.602**	0.472
	InvFreq(0.3)	0.621	0.597	**0.474**
	wMean(2)	0.617	0.585	0.471
Jitter pre-train on *GTA5*	Mean	0.3353	0.3301	0.2816
	InvFreq(0.1)	**0.3536**	**0.3561**	**0.2991**
	InvFreq(0.2)	0.3530	0.3400	0.2764
	InvFreq(0.3)	0.3370	0.3345	0.2688

Table 7. Results after 200 epochs in Centralized setting, and 800 rounds in Federated Learning; using for both approaches RandomCrop.

siders how ”refined” are the predictions. In Figure 5 we can see the details for each class, and in Table 7 the average values: we see mIoU improvements in all sets for all scenarios, and in particular we achieve a better adaptation to different domains of *TS2*. In Figure 6 we can see how it tends to overestimate less frequent (or more weighted) classes, depending on the parameters; however we can appreciate how it manages to predict also distant pedestrians, lights and poles completely missing in the standard Mean reduction, even in not too much distorted predictions.

5. Conclusion

In this paper, we focus on a realistic federated scenario for Semantic Segmentation in autonomous driving. In this setting, target data on clients' devices are unlabeled and not homogenously distributed among them. We conduct experiments in various settings, to fine-tune hyperparameters and gain insights into the critical aspects of the federated scenario.

To overcome the challenges arising from statistical heterogeneity, we propose a client clustering scheme based on style extraction using the FDA technique. Additionally, to enhance the recall for less frequent classes such as pedestrians and riders, which hold significant importance in the context of self-driving vehicles, we employ a weighted loss technique. Notably, this approach not only improves IoU for the selected classes but also elevates the overall mIoU.

References

[1] Emanuele Alberti, Antonio Tavera, Carlo Masone, and Barbara Caputo. IDDA: a large-scale multi-domain dataset for autonomous driving. *CoRR*, abs/2004.08298, 2020. 4

[2] Debora Caldarola, Massimiliano Mancini, Fabio Galasso, Marco Ciccone, Emanuele Rodola, and Barbara Caputo. Cluster-driven graph federated learning over multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2749–2758, June 2021. 3

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016. 2, 4

[4] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568. Curran Associates, Inc., 2020. 3

[5] Lidia Fantauzzo, Eros Fanì, Debora Caldarola, Antonio Tavera, Fabio Cermelli, Marco Ciccone, and Barbara Caputo. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11504–11511, 2022. 2

[6] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[8] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[9] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. 2, 4

[10] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020. 2

[11] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 1

[12] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020. 2

[13] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. *CoRR*, abs/1904.10620, 2019. 3

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015. 2

[15] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016. 1, 2, 3

[16] Jiaxu Miao, Zongxin Yang, Leilei Fan, and Yi Yang. Fedseg: Class-heterogeneous federated learning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8042–8052, June 2023. 2

[17] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. *CoRR*, abs/1608.02192, 2016. 4

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[19] Donald Shenaj, Eros Fanì, Marco Toldo, Debora Caldarola, Antonio Tavera, Umberto Michieli, Marco Ciccone, Pietro Zanuttigh, and Barbara Caputo. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 444–454, 2023. 2, 3

[20] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *CoRR*, abs/2103.00710, 2021. 3

[21] Yanchao Yang and Stefano Soatto. FDA: fourier domain adaptation for semantic segmentation. *CoRR*, abs/2004.05498, 2020. 3

[22] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. A review of single-source deep unsupervised visual domain adaptation. *CoRR*, abs/2009.00155, 2020. 1