POLITECNICO DI TORINO
Repository ISTITUZIONALE

FedDrive: Generalizing Federated Learning to Semantic Segmentation in Autonomous Driving

Original
FedDrive: Generalizing Federated Learning to Semantic Segmentation in Autonomous Driving / Fantauzzo, L., Fani', E., Caldarola, D., Tavera, A., Cermelli, F., Ciccone, M., Caputo, B.. - (2022), pp. 11504-11511. (2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022) Kyoto (JPN) 23-27 October 2022) [10.1109/IROS47612.2022.9981098].

Availability:
This version is available at: 11583/2970566 since: 2022-08-09T08:47:20Z

Publisher:
IEEE Robotics and Automation Society

Published
DOI:10.1109/IROS47612.2022.9981098

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

10 September 2026

FedDrive: Generalizing Federated Learning to Semantic Segmentation in Autonomous Driving

Lidia Fantauzzo*,[1], Eros Fanì*,[1], Debora Caldarola[1], Antonio Tavera[1], Fabio Cermelli[1,2], Marco Ciccone[1], Barbara Caputo[1]

***Abstract*— Semantic Segmentation is essential to make self-driving vehicles autonomous, enabling them to understand their surroundings by assigning individual pixels to known categories. However, it operates on sensible data collected from the users' cars; thus, protecting the clients' privacy becomes a primary concern. For similar reasons, Federated Learning has been recently introduced as a new machine learning paradigm aiming to learn a global model while preserving privacy and leveraging data on millions of remote devices. Despite several efforts on this topic, no work has explicitly addressed the challenges of federated learning in semantic segmentation for driving so far. To fill this gap, we propose FedDrive, a new benchmark consisting of three settings and two datasets, incorporating the real-world challenges of statistical heterogeneity and domain generalization. We benchmark state-of-the-art algorithms from the federated learning literature through an in-depth analysis, combining them with style transfer methods to improve their generalization ability. We demonstrate that correctly handling normalization statistics is crucial to deal with the aforementioned challenges. Furthermore, style transfer improves performance when dealing with significant appearance shifts. Official website: https://feddrive.github.io.**

I. Introduction

Research in autonomous driving aims at improving our safety and driving experience. In order to derive and execute trustworthy actions autonomously, vehicles must be able to perceive and understand their surroundings [1], [2], [3], [4], [5]. To ensure the safety of the passengers, an autonomous vehicle needs to determine precisely whether there is a danger such as an obstruction or a pedestrian, and consequently decide whether to slow down or accelerate. To do so, vehicles rely on the semantic segmentation task [6], [7], [8], [9], whose goal is to provide a semantic prediction for every pixel of the image, giving a deep understanding of the surrounding environment. However, training robust semantic segmentation models requires having access to large scale datasets possibly representing all the conditions that can be encountered in the real world. One possible solution is to collect images from the customers' vehicles which, being already on the road, face different situations and cover multiple geographic locations, weather conditions, viewpoints, etc. While collecting the images from vehicles is an effortless process, sending them to a central server to train a model could potentially violate the users privacy. Indeed, other nodes of the communication infrastructure may access personal and privacy-protected information, thus violating the regulations in force [10], [11], [12].

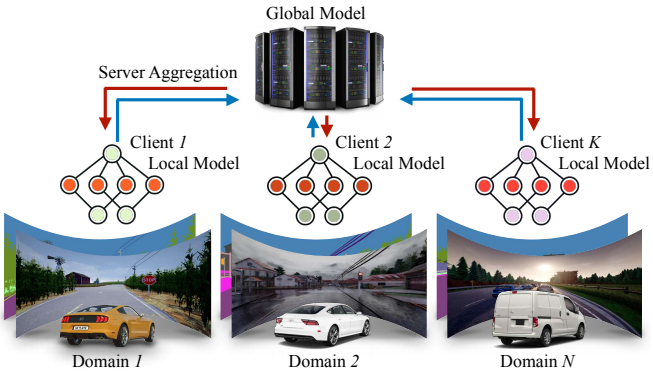

Fig. 1: Multiple K vehicles (*clients*), driving in different N domains, interact with a central *server* to learn a global model while keeping route data private. At each communication round, a subset of clients performs training on its non-iid data, sharing only the local update with the server, which subsequently sends back the newly aggregated model.

A clever solution to train a model using all the clients' data while protecting their privacy is introduced by federated learning (FL) [13]. FL algorithms learn a shared model leveraging a distributed training between several remote clients using their data in a privacy-compliant way. However, FL is a relatively new field and current methods mainly focus on simple vision tasks, such as image classification [13], [14], [15], [16]. Hence, the usage of FL for Semantic Segmentation in realistic urban environments opens the door to issues relating to the variety of witnessed scenes: images captured outdoors suffer from enormous variability in light, reflections, points of view, atmospheric conditions and types of settings, resulting in various *domains* (Fig. 1). It is unclear how the current FL methods may perform under such a challenging scenario.

To answer this open question, we introduce the first Semantic Segmentation benchmark in a Federated Learning scenario applied to autonomous driving, called *FedDrive*. While designing FedDrive, we focused on two important challenges that may occur when training a federated model in the real world: i) high statistical heterogeneity [17], [18], [19], and ii) domain generalization [15], [16]. Statistical heterogeneity is a relevant issue for FL methods in real-world scenarios [14], [19]. Each client (vehicle) usually observes part of a meta-distribution or different factors of variation

∗ Equal contribution.

[1] All the authors are supported by Politecnico di Torino, Turin, Italy. Corresponding authors: lidia.fantauzzo@studenti.polito.it, eros.fani@polito.it. Other authors: name.surname@polito.it.

[2] Fabio Cermelli is with Italian Institute of Technology, Genoa, Italy.

(geographical location, viewpoint, weather, illumination), which are generally different from other clients. When increasing the heterogeneity of the clients, FL algorithms suffer in terms of convergence speed, increasing training time and hampering the final model performance.

On the other hand, Domain Generalization (DG) requires a model to perform well also on unseen conditions, *i.e.* on domains different from the training ones. This is an essential ability for a model to operate in the real world, since not all the conditions can be anticipated at training time and a model should guarantee to safely operate on them.

To assess if current methods are able to deal with these real world challenges, we benchmark on FedDrive the main state-of-the-art federated learning algorithms FedAvg [13], SiloBN [15], FedBN [16], and different server optimizers [18], [20]. Moreover, in order to deal with the domain shift, we combine them with style-transfer methods [21], [22] to obtain a model capable of effectively generalizing on unseen domains.

To summarize, our main contributions are the following:

1) We introduce FedDrive, the first benchmark for FL in Semantic Segmentation applied to autonomous driving, focusing on the real world challenges of statistical heterogeneity and domain generalization.
2) We compare state-of-the-art federate learning methods and combine them with style transfer techniques to improve their generalization.
3) We show the importance of using the correct clients' statistics when dealing with different domains and that style transfer techniques improve the performance on unseen domains, proving to be a solid baseline for future research in federated semantic segmentation.

II. Related Work

A. Semantic Segmentation

Semantic segmentation is a crucial task for autonomous driving applications whose goal is to predict the class of every pixel in the image. State-of-the-art methods employ an encoder-decoder architecture based on Convolutional Neural Networks [6], [8] or Transformer modules [7], [23]. The vast majority of the segmentation methods assume to operate in a centralized setting, where the training is performed on a server and then deployed to multiple clients. To the best of our knowledge, no prior works investigated the real-world scenario of federated learning in semantic segmentation for autonomous driving. It has been mostly addressed in the field of medical images [24], [25], [26], [27], where these works only tried to solve the problem related to safeguarding the privacy of patients while using images collected from multiple sources, providing ad-hoc techniques. Only recently, FedProto [28] proposed a general method to federated learning for object segmentation. They introduced a novel approach that computes client deviation using margins of prototypical representations learned on distributed data and applies them to drive federated optimization via an attention mechanism.

B. Addressing Statistical Heterogeneity in FL

Recent years have seen a growing interest in Federated Learning and its applications to the real world, focusing many efforts on finding an effective solution in the face of the marked heterogeneity of clients' data [29], [19], [30].

The primary federated optimization approach FedAvg [13] exploits a weighted average of the clients' updates for learning the global model. Unfortunately, while effective on homogeneous scenarios, FedAvg usually loses in convergence performance and speed when facing non-i.i.d. data [29], [18]. Many works try to deal with the *client drift* introduced by the different data distributions by acting on the local optimization, so that the local models do not move in opposite directions with respect to the global one [31], [32], [14], [33]. Another line of research looks at the model aggregation introducing server-side momentum [18], and optimizers [20] to deal with FedAvg lack of adaptivity. While proving effective even in the most challenging scenarios, all those works mainly focus on the heterogeneity based on *labels* distribution skew. So far, there are only a few attempts at taking into account the non-i.i.d.ness induced by *domain features* shift, which is of interest for our work. Following this direction, FedRobust [34] addresses the affine distribution shift by learning affine transformations. Another consistent line of research tackles such issue by learning domain-specific batch normalization (BN) statistics [15], [35], [16]. In particular, SiloBN [15] keeps all BN statistics strictly local and leverages Adaptive Batch Normalization [36] to address new domains at test time, while FedBN [16] introduces local BN layers. Both these methods result in personalized local models. Differently, we aim at learning a *global* model capable of generalizing to different domains.

C. Federated Domain Generalization

Domain Generalization (DG) [37] seeks to extract a domain-agnostic model that can be applied to a previously unknown domain. Some strategies attempt to reduce domain shift between different source domains [38], [39], while others rely on meta-learning solutions [40], [41], [42]. Nonetheless, both solutions violate user privacy due to the necessity to access all data.

Methods that respect user privacy, such as [43], [44], and [45], rely on deep neural networks, training heuristics, or data augmentation, respectively. Despite this, they cannot deal with imbalances in data distribution across domains. Quande *et al.* [21] were the first to present a solution to the DG problem in Federated Learning. They propose a Continuous Frequency Space Interpolation (CFSI) to exchange data distribution information among clients while protecting their privacy. In addition, [46] proposes a novel gradient alignment loss based on the Maximum Mean Discrepancy (MMD). Following the footsteps of [21] and [46], we show and analyze how state-of-the-art image-to-image translation methods, such as the frequency exchange in [21] or the LAB color space transformation proposed in [22], can be used to address the problem of generalization to unknown domains in Federated Learning.

TABLE I: Summary of the settings in FedDrive. The uniform distribution is *iid* while the heterogeneous is *non-iid*.

Dataset	Setting	Distribution	# Clients	# Samples per user	Test clients
Cityscapes	-	uniform	146	10 - 40	unseen domains (new cities)
	-	heterogeneous	144	10 - 45	unseen domains (new cities)
IDDA	country	uniform	90	48	seen + unseen (country) domains
		heterogeneous	90	48	seen + unseen (country) domains
	rainy	uniform	69	48	seen + unseen (rainy) domains
		heterogeneous	69	48	seen + unseen (rainy) domains

III. FedDrive: A New Benchmark

In this Section, we first formalize the federated scenario and the addressed task, explaining a fundamental issue arising in real world environments, *i.e.* the data *statistical heterogeneity* and the *domain generalization*, and how it affects the autonomous driving setting. Then, we introduce FedDrive, the first benchmark for studying autonomous driving in federated scenarios.

A. Federated Learning Setup

The standard federated scenario is based on a client-server architecture. The server is the trusted party, while clients can be any computation-capable device, from personal smartphones to self-driving cars. Given a set of clients $\mathscr{K}$, with $|\mathscr{K}| = K$, a client $k \in \mathscr{K}$ has access to a *privacy-protected* local dataset $\mathscr{D}_k$ of dimensions n_k. In the proposed scenario, each instance of $\mathscr{D}_k$ is an image $x \in \mathscr{X}$ made of N pixels associated with its ground truth vector $y \in \mathscr{Y}^N$, denoting the set of N tuples with elements belonging to the class space $\mathscr{Y}$. Each pair is drawn from a local distribution $(x,y) \sim P_k$. In autonomous driving scenarios, data distributions could be determined by the city in which the user lives, his usual driving routes, or typical weather conditions. The goal is to learn a global shared model $F(w) : \mathscr{X} \rightarrow \mathbb{R}^{N \times |\mathscr{Y}|}$ with parameters w, mapping the image space $\mathscr{X}$ to a pixel-wise class probability vector, without breaking the privacy constraints. We aim to solve the task of assigning a semantic class $y_i \in \mathscr{Y}$ to each pixel x_i of the input image $x \in \mathscr{X}$. That is accomplished through a learning paradigm based on T communication rounds. During each round t, *i)* the server exchanges the current model parameters w^t with a subset $\mathscr{K}' \subseteq \mathscr{K}$ of the online devices, *ii)* the selected clients train $F(w^t)$ on their local data for E epochs and send back the update w_k^{t+1}, *iii)* all the updates are then aggregated on the server-side according to a chosen algorithm, consequently updating the global model.

B. Statistical Heterogeneity: A Real World Issue

When dealing with realistic scenarios, users' data distributions may drastically differ. Looking at the world of autonomous driving, we can easily imagine how the data acquired from a car in Northern Europe will contain snow with a higher probability than those obtained from another one located on the Southern Coast. Alternatively, they may have access to similar weather conditions but different landscapes, *e.g.* cityscapes instead of rural areas. These differences are brought together under the term *statistical heterogeneity* and represent one of the main challenges faced by the FL community today [18], [19], [30]. Indeed, the standard federated paradigm is usually effective in homogeneous scenarios but fails at generalizing when facing heterogeneous - and more realistic - ones. Formally, given two clients $(i, j) \in \mathscr{K} \times \mathscr{K}$ it is highly likely that $P_i \neq P_j \, \forall i \neq j$, where P_i is the probability distribution associated to the i-th client. As a consequence $F_i(w) \neq F_j(w) \neq F(w)$, *i.e.* the more the distributions are different, the greater the difference between locally learned models and the global one [32]. In general, let $(x,y) \sim P_i(x,y)$ be a sample drawn from the i-th client's local distribution. We usually face two kinds of heterogeneity: i) *label* distribution skew, *i.e.* the marginal distribution $P_i(y)$ may vary across clients i and j even if $P_i(y|x) = P_j(y|x)$; ii) *feature* distribution skew, *i.e.* the same label y can be associated with different features x across clients. Formally, $P_i(x)$ and the conditional distribution $P_i(x|y)$ vary even if $P_i(y|x) = P_j(y|x)$ and $P_i(y)$ is shared, $\forall i \neq j$. For example, the same concept of house can be associated with totally different structures in the world. In this work, we focus on the latter and refer to the distinct feature distributions as ***domains***. The introduction of SS in urban environments exacerbates these issues due to the presence of a larger number of classes within a single image, which results in more possible distributions. Moreover, the outdoor scenes typical of the autonomous driving setting introduce further shifts due to changes in light, reflections, point of view, atmospheric conditions, etc. The presence of different domains in the peripheral devices will be referred to as *domain imbalance*.

C. Federated Domain Generalization

In a realistic setting, not only clients have their own local data distribution, but new clients may appear at any time, introducing novel distributions. For example, in an autonomous driving scenario new clients may appear in a different location (such as in a different city or country) or may introduce a new vehicle with other characteristics. A robust model should be able to generalize to novel, unseen conditions to operate in the real world. For this reason, in our benchmark, we focus on model generalization and explicitly assess a model's ability to operate on unseen domains at test time. Formally, evaluation will be performed on a test client s having a distribution $P_s(x) \neq P_k(x) \, \forall k \in \mathscr{K}$.

D. FedDrive

To investigate the different challenges of Federated Learning in autonomous driving, we introduce a new benchmark

based on Cityscapes [47], and IDDA [48] datasets. We propose multiple federated versions for each of them, based on different levels of data heterogeneity between clients. Table I summarizes the multiple experimental settings emulating realistic scenarios.

Cityscapes is one of the most popular datasets for semantic segmentation and is a set of real photos taken in the streets of 50 different cities with good weather conditions. The dataset contains 2975 images for training and 500 for testing, providing annotations for 19 semantic classes. The first and more naïve choice for adapting the dataset to the federated scenario consists of *uniformly* splitting it among a fixed number of clients, *i.e.* each image is randomly drawn and assigned to one of the users. While this distribution allows for performance evaluation in a federated environment, it does not consider the real-world challenge of statistical heterogeneity. Therefore, we propose an additional split, denoted as *heterogeneous*, that takes into account the information about the city where the photo was taken, resulting in a non-i.i.d. domain distribution across clients. Specifically, we set up 8 clients for each of the 18 training cities for a total of 144 devices and divide the images of each city among its clients. The test client contains all the images belonging to cities never seen at training time for both the uniform and the heterogeneous settings.

IDDA is a synthetic dataset for semantic segmentation in the field of self-driving cars, providing annotations for 16 semantic classes with a broad variety of driving conditions, characterized by three axes: 7 towns (ranging from Urban to Rural environments), 5 viewpoints (simulating different vehicles), and 3 weather conditions (Noon, Sunset and Rainy scenarios) for a total of 105 domains. As done for Cityscapes, we provide both a uniform and a heterogeneous version of the federated dataset. In the *uniform* distribution, each client has access to 48 images drawn randomly from the whole dataset. Since such distribution is highly unrealistic, we only use it as a reference for the model's performance. The *heterogeneous* federated version of IDDA requires each client to see images belonging to a single domain. In addition, to test the generalization capabilities of the learned model, we use two test clients, one with images belonging to the already seen training domains (*seen-dom*) and one with never seen ones (*unseen-dom*).

IDDA is further distinguished into two possible settings in order to analyze both *semantic* and *appearance* shift. As for the former, the *unseen-dom* test client contains images of a country town, while in the second case, the photos are taken in rainy conditions. We denoted this two setting variations respectively as *country* and *rainy*.

IV. Methods

In this Section, we present the algorithms tested on our benchmark. Starting from the standard FedAvg [13] and its adaptive version [20], we address the domain shift between clients applying methods proper to the literature of heterogeneous Federated Learning [15], [16], Domain Adaptation and Generalization [37], which are here described.

A. FedAvg: An Overview

FedAvg [13] is the standard approach for handling federated baselines. At each round t, given a subset $\mathscr{K}'$ of clients with $|\mathscr{K}'| = K'$ and the current global model parameters w^t, the goal is to optimize

$$\arg\min_{w} \sum_{k=1}^{K'} p_k L_k(w) \tag{1}$$

where $L_k = \mathbb{E}_{(x,y)\sim\mathscr{D}_k}\left[\mathscr{L}\big(F(w;x),y\big)\right]$ is the local empirical risk, $\mathscr{L}$ is typically the cross-entropy loss and $p_k > 0$ is usually set to n_k/n, with $n = \sum_k n_k$ being the total number of training images. The global model w^{t+1} is then updated as $1/n \sum_{k\in\mathscr{K}'} n_k w_k^t$. The authors of [20] show that applying FedAvg is equivalent to use SGD optimizer on the server-side with learning rate $\eta = 1$. In particular, if w_k^t is the k-th client's local update, we set $\Delta_k^t := w_k^t - w^t$ and $\Delta^t := 1/K' \sum_{k\in\mathscr{K}'} \Delta_k^t$. Then the FedAvg update corresponds to applying SGD to the pseudo-gradient $-\Delta^t$. Such formulation allows for testing different server-side learning rates and optimizers with the goal of improving FedAvg convergence. In Sec. V-C, we propose an analysis using SGD with momentum [18], Adam [49] and Adagrad [50].

B. Batch Normalization for Domain Adaptation

While FedAvg is very effective in homogeneous scenarios, it loses in performance when facing non-i.i.d. data. Previous works investigated the critical role of Batch Normalization [51] layers in addressing non-i.i.d. data [52], [53], [54], [36], [55]. Batch Normalization (BN) was introduced by Ioffe & Szegedy [51] to speed up the training with faster learning rates and reduce the sensitivity to the network initialization and have proved essential ever since. A BN layer receives as an input the batch $x = (x_1, \ldots, x_B)$ of size B with $x_i \in \mathscr{X}\ \forall i \in [B]$ and normalizes it as

$$BN(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{2}$$

where μ and σ are the *BN statistics* that correspond to the running mean and variance of each channel, respectively; the scale γ and bias β are learnable parameters, and ε is a small scalar that prevents division by 0.

The main drawback of BN layers is the assumption that training and test data are drawn from the same distribution [55], which makes them unsuitable for our scenario. As a solution, Li *et al.* [16] propose FedBN, which leverages BN layers to alleviate the feature shift occurring during the server-side parameter aggregation. The local BN layers are updated locally but never shared and aggregated on the server-side, resulting in personalized parameters. The main issue arising with such an approach is the addressing of unseen domains or clients at test time. SiloBN [15] differentiates the roles played by the BN statistics $\{\mu, \sigma\}$ and the learnable parameters $\{\gamma, \beta\}$: the former capture *local* domain information, while the latter are transferable across domains. That is why the clients only share the parameters, keeping

Fig. 2: Examples of DG techniques used to transfer the domain from a source to a target image. CFSI produces artifacts if the amplitude interpolation tends toward the target; otherwise, the difference from the original image is not as noticeable. The LAB technique appears to work well in all conditions since the style of target is clearly transferred to the source.

the statistics local to address the heterogeneity. New clients and domains at test time are handled using AdaBN [36], recomputing the BN statistics on the new testing domain while freezing all the other parameters of the server model. Both FedBN and SiloBN will be used as baselines in Sec. V.

C. Generalization through Style Translation

SiloBN and FedBN address statistical heterogeneity by manipulating the BN statistics. However, in order to improve the generalization capabilities, some recently proposed techniques go beyond the model and directly modify the images [21], [22]. An example of these methods is reported in Fig. 2. The fundamental idea behind these methods is to share distribution information among clients to overcome the limits of decentralized datasets.

Continuous Frequency Space Interpolation (CFSI) [21]. Given an image x_i^k from the k-th client, the frequency space signal can be derived from the Fast Fourier Transform (FFT). The low-level information (*i.e.* color, brightness, etc.) is reflected in the amplitude spectrum of the FFT. To share the distribution information among local clients, a distribution bank $\mathscr{A} = \{\mathscr{A}^1, \ldots, \mathscr{A}^K\}$ is firstly created, where each $\mathscr{A}^k = \{\mathscr{A}_i^k\}_{i=1}^{n_k}$ contains all amplitude spectrum of images from the k-th client. A continuous interpolation of the frequencies is used to transfer multi-source distribution to each local client. [21] interpolates each client image to each domain distribution. Still, due to the significant computation required by the style transfer methods, this is not scalable or feasible in a realistic setting with several domains. As a result, we apply CFSI transformation to only half of the client data and sample only one target amplitude spectrum randomly from the shared distribution bank $\mathscr{A}$. The interpolation ratio λ is used to sample images uniformly in a range $[0,1]$.

LAB-based Image Translation (LAB) [22]. To share the distribution information across local clients, a shared bank $\mathscr{L} = \mathscr{L}^1, \ldots, \mathscr{L}^K$ is created, where each $\mathscr{L}^k = \{\mu_i^k, \sigma_i^k\}_{i=0}^{n_k}$ is the set of means and standard deviations for each image of client k, transformed in the LAB color space. As previously done, we translate only half of the client data. More specifically, an RGB image x_S^{RGB} of a client k is firstly converted to the LAB color space x_S^{LAB}. Then, the mean μ_S and the standard deviation are computed for each channel of x_S^{LAB}. A random pair of (μ_T, σ_T) is sampled from the distribution bank $\mathscr{L}$. The x_S^{LAB} image style is then translated as follow:

$$\hat{x}_S^{LAB} = \frac{x_S^{LAB} - \mu_S}{\sigma_S} * \sigma_T + \mu_T.$$

Following the alignment of the distribution, the translated LAB image $\hat{x}_S^{LAB}$ is converted back to the RGB color space as $\hat{x}_S^{RGB}$, for the subsequent training phase.

V. Experiments

A. Implementation Details

To fully exploit the limited capabilities of local devices, we choose a BiSeNet V2 [9] architecture, a lightweight network with two principal branches capturing spatial features and high-level semantic context. For simplicity, we use the common vanilla FedAvg aggregation with SGD as server optimizer for all the experiments, with learning rate 1. We also provide an ablation on the server optimizer, showing that adding momentum helps reaching very high performances in the uniform and heterogeneous settings and seen domain, while giving only few points of improvement in the domain shift case. We minimize the cross-entropy loss on every client with SGD as local client optimizer with an initial learning rate of 0.05 for Cityscapes and 0.1 for IDDA, momentum 0.9, weight decay 0.0005 for Cityscapes and no weight decay for IDDA, and batch size 16. We point out that when the clients do not have enough samples to use 16 as batch size, we virtually double their dataset by horizontally flipping the images. Following [9], we use a polynomial learning rate schedule on every client, *i.e.* the initial learning rate is multiplied by $(1 - \frac{\text{iter}}{\text{max_iter}})^{0.9}$. We apply OHEM (Online Hard-Negative Mining) [56], selecting 25% of the pixels having the highest loss for the optimization. We apply a standard data augmentation pipeline: we randomly scale the images

TABLE II: FedDrive: Cityscapes results.

Method	mIoU ± std (%)
FedAvg (uniform)	45.71 ± 0.37
FedAvg	43.85 ± 1.24
FedAvg + CFSI	41.50 ± 0.98
FedAvg + LAB	39.20 ± 1.37
SiloBN	**44.20 ± 1.43**
SiloBN + CFSI	40.48 ± 1.40
SiloBN + LAB	42.23 ± 1.23

in the range $(0.5, 1.5)$ for Cityscapes and $(0.5, 2)$ for IDDA, then extract a crop with size 512×1024 for Cityscapes and 512×928 for IDDA. Following a convergence analysis, we decide to use 5 clients per round and 2 local epochs as it is the optimal combination in terms of speed of convergence and overall runtime. In the lack of a previous reference, we set the total number of rounds for all the experiments at 1600, We decide to report the mean and the standard deviation of the mean Intersection over Union (mIoU) of the evaluations every 5 rounds over the last 100 rounds due to some instability of the test results.

B. Results

Cityscapes. Tab. II reports Cityscapes experiments. The first line shows the performance of FedAvg in the *uniform* scenario ($45.71 \pm 0.37\%$) as a reference for the performance in the absence of domain shift. Introducing an imbalance in the client's distribution, *i.e.* considering the *heterogeneous* scenario, all the corresponding experiments record a decrease in performance and an increase of the standard deviation. In Cityscapes, the improvements brought by SiloBN are limited in Cityscapes since the domain shift across cities is mostly related to semantic rather than style. Test images, in fact, belong to cities never seen during training but to the same country and are taken on similar weather conditions. For the same reason, DG methods do not contribute to performance improvement: SiloBN obtains 44.20% mIoU on average, while combining it with CFSI and LAB, it achieves, respectively, 40.48% and 42.23%. In particular, both CFSI and LAB act on the style of the image rather than on its content. A slight drop in performance is instead obtained since the network requires more time to converge with a stronger data augmentation given by the style transfer methods. Qualitative results in Fig. 3 confirm that SiloBN with LAB does not improve the performance w.r.t. FedAvg.

IDDA. In Tab. III, the experiments on the IDDA dataset are reported for the *heterogeneous* distribution of both *country* and *rainy* settings. Similarly to Cityscapes, the first row reports the performance of FedAvg in the uniform scenario as a reference. We can see that good performances are obtained when using the same domain test client (*seen*). However, on the *unseen* test client, the performances are lower, especially in the rainy setting, where the mIoU drops by 35 mIoU percentage points on average with respect to the *seen* test client. This underlines the significant domain shift introduced by the *unseen* test client. We remark that country and rainy settings provide different challenges, as can be seen from Fig. 3. The country domain has a similar style w.r.t. the training clients, since the images are taken in different weather conditions, but has other semantic characteristics, *e.g.* it rarely contains sidewalks near the road, as instead frequently happens on the training clients. On the opposite, the rainy domain has similar semantic characteristics but an essentially different appearance since there are no rainy images on training.

TABLE III: FedDrive: IDDA results in mIoU ± std (%).

Method	country		rainy	
	seen	unseen	seen	unseen
FedAvg (uniform)	63.57 ± 0.60	49.74 ± 0.79	62.72 ± 3.65	27.61 ± 2.80
FedAvg	42.43 ± 1.78	40.01 ± 1.26	38.18 ± 1.40	26.75 ± 2.32
FedAvg + CFSI	54.70 ± 1.12	45.70 ± 1.73	55.24 ± 1.65	31.05 ± 2.68
FedAvg + LAB	56.59 ± 0.90	45.68 ± 1.04	58.85 ± 0.89	26.82 ± 1.78
FedBN	54.39	-	56.45	-
SiloBN	58.82 ± 2.93	45.32 ± 0.90	62.48 ± 1.42	50.03 ± 0.79
SiloBN + CFSI	61.22 ± 3.88	49.17 ± 1.01	63.04 ± 0.31	50.54 ± 0.88
SiloBN + LAB	**64.32 ± 0.76**	**50.43 ± 0.63**	**65.85 ± 0.91**	**53.99 ± 0.79**

When introducing a domain imbalance, the performance of FedAvg has a significant drop, especially in the seen test client. On average, the performance on the seen test client drops by nearly 23 mIoU percentage points, achieving 42.43% on the country setting and 38.18% on the rainy one. This result emphasizes the domain imbalance across training clients and the high statistical heterogeneity of this setting. Differently, the performance drop on the unseen test client is limited, with a drop of about 2 mIoU percentage points on the country and a drop of about 12 mIoU percentage points on the rainy setting.

The introduction of techniques able to cope with statistical heterogeneity largely improves the performance on the seen test client. FedBN and SiloBN show an improvement, on average, of 14 mIoU percentage points on country and of 21 mIoU percentage points on rainy settings, strongly reducing the domain gap between training clients. Moreover, analyzing the results on the unseen test client, we see two different behaviors of SiloBN. On the country setting, it shows only a limited improvement with respect to FedAvg: $40.01 \pm 1.26\%$ FedAvg vs $45.32 \pm 0.90\%$ SiloBN. Differently, it largely improves the performance on the rainy setting: $26.75 \pm 2.32\%$ FedAvg vs $50.03 \pm 0.79\%$ SiloBN. These results confirm that SiloBN strongly reduces the domain shift related to the appearance. Still, it does not alleviate much the semantic domain shift present in the country domain.

Overall, we can see that applying CFSI and LAB style transfer methods improve the performance across domains and settings. In particular, on the seen test client, LAB provides a considerable performance boost when applied to both FedAvg and SiloBN: on the country setting, it improves FedAvg by about 22 mIoU percentage points and SiloBN by about 6 percentage points, while on the rainy setting, it gains about 27 mIoU percentage points on FedAvg and about 4 mIoU percentage points on SiloBN. We note that it largely improves FedAvg, while it is less effective on SiloBN since

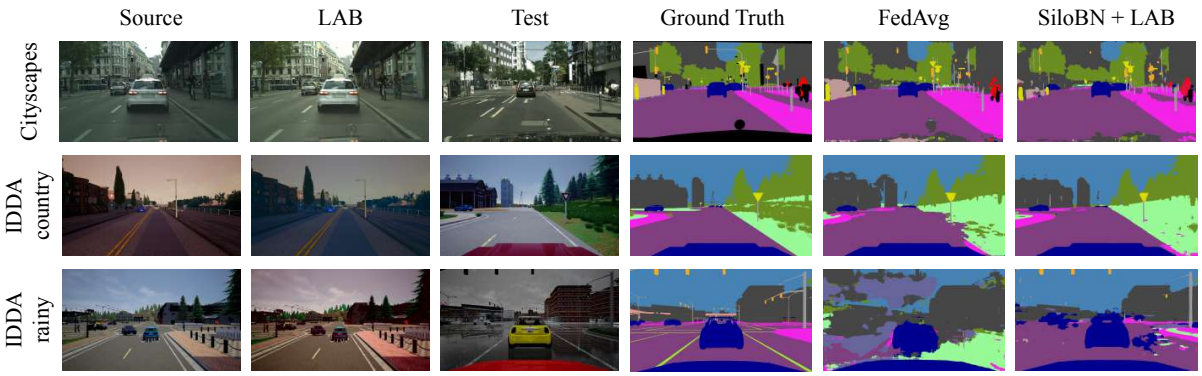

Fig. 3: Qualitative results in the *heterogeneous* scenarios for Cityscapes and for the two IDDA settings.

it has an effect similar to SiloBN: it transfers the style of the training domains across clients' images. However, we note that the style transfer is applied to the images rather than the BN statistics, as in SiloBN. When considering the performance on a different test domain (unseen), both LAB and CFSI improve the performance except in the rainy domain when applied to FedAvg, where LAB is less effective: FedAvg $26.75 \pm 2.32\%$ vs CFSI $31.05 \pm 2.68\%$ vs LAB $26.82 \pm 1.78\%$. Differently, we can see that LAB, when combined with SiloBN (Fig. 3), improves performance in both the country and rainy settings, by around 5 and 4 mIoU percentage points respectively compared to SiloBN, and by about 10 and 27 mIoU percentage points compared to FedAvg, respectively. Note how the combination of FedAvg + SiloBN + LAB is even much better than applying FedAvg only on the uniform split, expecially on the rainy unseen test domain, where it performs about 26 mIoU percentage points more than the uniform FedAvg only experiment.

Finally, observe how the standard deviation is generally higher for the unseen test domain than the seen test domain for both country and rainy tests, how it is higher for rainy than country experiments, underlining how the rainy partition is much more challenging than the country partition once again, and that CFSI and LAB contribute in reducing the variability of the results.

C. Ablation study on server optimizer

In Tab. IV we report the ablation study on the server optimizer on the IDDA settings. We compared four server optimizers: SGD with learning rate 1 and no momentum, FedAvgM [18], Adam [49] and AdaGrad [50]: for the last two we report the best result among the experiments with two learning rates: 0.1 and 1.0. According to the table, FedAvgM clearly outperforms the other server optimizers: in the uniform experiments it increases the average performance by around 8 mIoU percentage points for both the country and rainy settings for the seen test domain, while it increases the average performance by around 6 and 2 mIoU percentage points for the country and rainy partitions for the unseen test domain, respectively. For the heterogeneous distribution the increase in performance is lower but still worth, ranging from 2 to 5 mIoU percentage points according to the particular experiment and setting. Finally, the benefits of FedAvgM are less visible when using SiloBN, with slightly improvements and even a small decrease in performance of about 1.5 mIoU percentage points for the unseen test domain in the rainy setting with respect to the SGD server optimizer. In all the experiments, Adam and AdaGrad are not able to significantly outperform SGD and to get closer to FedAvgM results.

TABLE IV: Server optimizers comparison on IDDA.

	Method	Optimizer	country (mIoU%) seen	country (mIoU%) unseen	rainy (mIoU%) seen	rainy (mIoU%) unseen
uniform	FedAvg	SGD	63.57 ± 0.60	49.74 ± 0.79	62.72 ± 3.65	27.61 ± 2.80
		FedAvgM	**71.27 ± 0.85**	**55.47 ± 1.07**	**70.99 ± 0.71**	**29.83 ± 2.03**
		Adam	63.31 ± 0.37	50.21 ± 0.40	65.39 ± 0.52	31.72 ± 1.74
		AdaGrad	59.44 ± 0.94	46.09 ± 0.83	59.45 ± 1.30	27.80 ± 1.29
heterogeneous	FedAvg	SGD	42.43 ± 1.78	40.01 ± 1.26	38.18 ± 1.40	26.75 ± 2.32
		FedAvgM	**44.38 ± 1.98**	**42.42 ± 2.15**	**41.21 ± 1.98**	**31.91 ± 3.77**
		Adam	39.93 ± 2.44	38.15 ± 1.89	37.97 ± 2.04	28.47 ± 2.56
		AdaGrad	40.89 ± 2.05	38.03 ± 1.65	39.02 ± 1.71	27.08 ± 2.85
	SiloBN	**SGD**	58.82 ± 2.93	45.32 ± 0.90	62.48 ± 1.42	**50.03 ± 0.79**
		FedAvgM	**61.99 ± 1.51**	**46.20 ± 1.20**	**63.69 ± 1.25**	48.49 ± 1.04
		Adam	58.36 ± 1.26	42.31 ± 0.84	61.51 ± 0.90	47.22 ± 0.89
		AdaGrad	48.97 ± 1.34	41.69 ± 1.31	54.06 ± 1.29	45.80 ± 0.89

VI. Conclusions

We presented FedDrive, a benchmark for Semantic Segmentation in Federated Learning suited for the autonomous driving task. We proposed a realistic set of real-world scenarios to investigate the impact of heterogeneous domain distributions across clients, taking into account the constraints posed by stylistic and semantic shifts. To test our proposed benchmark and highlight the open challenges, we assess methods from both the Federated Learning and Domain Generalization literature. We show how the combination of SiloBN and DG methods, such as LAB and CFSI, yields only a slight improvement when dealing with the semantic shifts in Cityscapes. In comparison, when the gap in IDDA concerned both semantics and style, SiloBN had a positive impact, confirming its effectiveness only when the domain shift is related to appearance. Finally, we show that SiloBN and LAB together increases performance in all IDDA scenarios, alleviating the issues in the face of diverse types of domain shifts. We believe that FedDrive will foster research towards privacy-preserving autonomous driving.

VII. Acknowledgements

This work was supported by the Consorzio Interuniversitario Nazionale per l'Informatica (CINI).

References

[1] S. Jiang, Y. Wang, W. Lin, Y. Cao, L. Lin, J. Miao, and Q. Luo, "A high-accuracy framework for vehicle dynamic modeling in autonomous driving," in *IROS*, 2021.

[2] K. Tsiakas, I. Kostavelis, A. Gasteratos, and D. Tzovaras, "Autonomous vehicle navigation in semi-structured environments based on sparse waypoints and lidar road-tracking," in *IROS*, 2021.

[3] T. Ha, G. Lee, D. Kim, and S. Oh, "Road graphical neural networks for autonomous roundabout driving," in *IROS*, 2021.

[4] V. Lebakula, B. Tang, C. Goodin, and C. L. Bethel, "Shape estimation of negative obstacles for autonomous navigation," in *IROS*, 2021.

[5] D. Fontanel, F. Cermelli, M. Mancini, and B. Caputo, "Detecting anomalies in semantic segmentation with prototypes," in *(CVPRW)*, June 2021, pp. 113–121.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *T-PAMI*, vol. 40, 2017.

[7] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *NeurIPS*, vol. 34, 2021.

[8] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *IROS*, 2016.

[9] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet V2: bilateral network with guided aggregation for real-time semantic segmentation," *CoRR*, vol. abs/2004.02147, 2020.

[10] H. Zhang, J. Bosch, and H. H. Olsson, "End-to-end federated learning for autonomous driving vehicles," in *IJCNN*, 2021.

[11] A. Nguyen, T. Do, M. Tran, B. X. Nguyen, C. Duong, T. Phan, E. Tjiputra, and Q. D. Tran, "Deep federated learning for autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022.

[12] Y. Li, X. Tao, X. Zhang, J. Liu, and J. Xu, "Privacy-preserved federated learning for autonomous driving," *IEEE TITS*, 2021.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*. PMLR, 2017.

[14] H. Hsu, H. Qi, and M. Brown, "Federated visual classification with real-world data distribution," 2020.

[15] M. Andreux, J. O. du Terrail, C. Beguier, and E. W. Tramel, "Siloed federated learning for multi-centric histopathology datasets," in *Springer*, 2020.

[16] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fed{bn}: Federated learning on non-{iid} features via local batch normalization," in *ICLR*, 2021.

[17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *MLS*, 2020.

[18] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019.

[19] P. Kairouz *et al.*, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[20] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *ICLR*, 2021.

[21] Q. Liu, C. Chen, J. Qin, Q. Dou, and P.-A. Heng, "Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space," *CVPR*, 2021.

[22] J. He, X. Jia, S. Chen, and J. Liu, "Multi-source domain adaptation with collaborative learning for semantic segmentation," *CoRR*, 2021.

[23] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," 2021.

[24] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation," 2018.

[25] W. Li, F. Milletarì, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. J. Cardoso, and A. Feng, "Privacy-preserving federated brain tumour segmentation," in *Machine Learning in Medical Imaging*, 2019.

[26] L. Yi, J. Zhang, R. Zhang, J. Shi, G. Wang, and X. Liu, "Su-net: An efficient encoder-decoder model of federated learning for brain tumor segmentation," in *ICANN*, 2020.

[27] C. I. Bercea, B. Wiestler, D. Rueckert, and S. Albarqouni, "Feddis: Disentangled federated learning for unsupervised brain pathology segmentation," 2021.

[28] U. Michieli and M. Ozay, "Prototype guided federated learning of visual feature representations," 2021.

[29] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *ICLR*, 2019.

[30] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *CoRR*, vol. abs/1908.07873, 2019.

[31] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020.

[32] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *ICML*, 2020.

[33] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," *ICLR*, 2021.

[34] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust federated learning: The case of affine distribution shifts," in *NeurIPS*, vol. 33, 2020.

[35] D. Caldarola, M. Mancini, F. Galasso, M. Ciccone, E. Rodolà, and B. Caputo, "Cluster-driven graph federated learning over multiple domains," in *CVPR-W*, 2021.

[36] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," in *ICLR-W*, 2017.

[37] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," *NeurIPS*, 2011.

[38] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *ICCV*, 2015.

[39] Y.-C. Hsu, Z. Lv, and Z. Kira, "Learning to cluster in order to transfer across domains and tasks," *ArXiv*, vol. abs/1711.10125, 2018.

[40] Q. Liu, Q. Dou, and P.-A. Heng, "Shape-aware meta-learning for generalizing prostate mri segmentation to unseen domains," in *MICCAI*, 2020.

[41] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," in *NeurIPS*, 2018.

[42] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *ICCV*, 2017.

[43] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *ECCV*, 2012.

[44] Z. Huang, H. Wang, E. P. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *ECCV*, 2020.

[45] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," *CoRR*, vol. abs/1804.10745, 2018.

[46] C. X. Tian, H. Li, Y. Wang, and S. Wang, "Privacy-preserving constrained domain generalization for medical image classification," *CoRR*, vol. abs/2105.08511, 2021.

[47] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[48] E. Alberti, A. Tavera, C. Masone, and B. Caputo, "Idda: A large-scale multi-domain dataset for autonomous driving," *IEEE RAL*, vol. 5, 2020.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[50] A. Lydia and S. Francis, "Adagrad—an optimizer for stochastic gradient descent," *Int. J. Inf. Comput. Sci*, vol. 6, 2019.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[52] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, "Just dial: Domain alignment layers for unsupervised domain adaptation," in *ICIAP*, 2017.

[53] F. Maria Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulo, "Autodial: Automatic domain alignment layers," in *ICCV*, 2017.

[54] M. Mancini, S. R. Bulo, B. Caputo, and E. Ricci, "Adagraph: Unifying predictive and continuous domain adaptation through graphs," in *CVPR*, 2019.

[55] J. Bronskill, J. Gordon, J. Requeima, S. Nowozin, and R. Turner, "TaskNorm: Rethinking batch normalization for meta-learning," in *ICML*, 2020.

[56] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016.