Project 3: Federated Semantic Segmentation for self-driving cars

1st Gabriel Jenner de Faria Orsi
Politecnico di Torino
s294092@studenti.polito.it

2nd Victor Fiorin de Carvalho
Politecnico di Torino
s304547@studenti.polito.it

3rd Mohamad Shehadeh
Politecnico di Torino
s298222@studenti.polito.it

***Abstract*—Semantic Segmentation is a task able to provide the category information at pixel level, essential to make self-driving vehicles autonomous. However, most of the data are private sensitive and in large quantity. To this end, Federated Learning is a clever solution to train a model using all the client's data while protecting their privacy. It is unrealistic to assume that the client's data is labeled. Federated Source-free Domain Adaptation (FFreeDA) is introduced to represent a more realistic scenario, where the client's data is unlabeled. In this case the server access a source-labeled data, created by swapping the low frequency spectrum between source and target. Moreover, pseudo-labels are used as an unsupervised self-training technique to improve performance. The GitHub Repository can be found here: https://github.com/VictorCarvalho112/Federated-Semantic-Segmentation**

***Index Terms*—Semantic Segmentation, Federated Learning, self-driving cars, pseudo-labels, Domain Adaptation**

I. Introduction

Autonomous vehicles are expected to bring many social benefits, such as increased productivity, a reduction of accidents, and battery energy efficiency. The ability of driverless cars to operate effectively and safely has been a popular study topic in recent years, and many companies and research institutions are working to develop the first fully functional driverless car model. This is a very promising field with several possible benefits, including enhanced safety, lower prices, more comfortable travel, increased mobility, and a reduced environmental impact. The primary concern with self-driving automobiles is the detection and avoidance of obstacles. The autonomous vehicles must accurately understand and interpret the visual information from their surroundings. First, the car must know where the street is, and then must detect the sidewalk, the trees, and so on to correctly determine whether to brake, steer or accelerate. Self-driving cars need a Semantic Segmentation module. Semantic Segmentation is the task of assigning semantic labels to each pixel, i.e., it is the process of assigning each pixel of the received image into one of the predefined classes. These classes represent the segment labels of the image, e.g., roads, cars, signs, traffic lights, or pedestrians.

However, it operates on sensible data collected from the users' cars; thus, protecting the clients' privacy becomes a primary concern. To this end, Federated Learning is a possible solution. Federated Learning is a relatively new model that allows the training of a global model without sharing the data of the users, that in Federated Learning are the clients, but instead in Federated Learning only the model is shared among the clients, the data is always local. However, most of the existing works on FL unrealistically assume labeled data in the remote clients.

In this paper, a new task of Federated Source-Free domain adaptation model that does not require access to source datasets at all points of time is proposed. It is assumed that there is access to a classifier that has been trained using the source dataset. The model's applicability in real-world circumstances completely depends on the accessibility of the classifiers rather than the entire dataset. To learn the joint distribution, the pre-trained classifier is used by modeling it as an energy-based function.

II. Related Work

A. Semantic Segmentation

Semantic segmentation is a computer vision technique that involves classifying each pixel in an image into one of several pre-defined categories or classes. In recent years, the application of semantic segmentation to self-driving cars has become an area of intense research helping autonomous vehicles understand and interpret the environment around them, including the positions and movements of other vehicles, pedestrians, and obstacles. A recent suburban traffic scene labeling dataset was unveiled [17], featuring 3D semantic instance annotations and 2D annotations obtained via back-projection [4]. To achieve improved results, the Bilateral Segmentation Network (BiSeNet V2) [19], has been proposed, which uses a two-pathway architecture to separate spatial details and categorical semantics and balance speed and accuracy. Recently, FedProto [13] introduced a new method for federated learning of object segmentation, using a novel approach. This approach computes client deviations using margins of prototypical representations learned on distributed data and applies them to drive federated optimization via an attention mechanism.

B. Federated Learning

Federated learning is a machine learning technique that enables multiple devices to collectively learn a shared model, without sending raw data to a central server. This approach offers significant advantages for privacy and security [11], as

it prevents the transfer of sensitive data and harnesses the collective computing power of multiple devices to handle large-scale datasets [9]. This approach has already been applied to a wide range of computer vision tasks, including object detection, image classification, and semantic segmentation. With the potential to improve model performance by training on a larger and more diverse dataset, federated learning holds great promise for the future of machine learning. The most common method of model aggregation in FL studies is Federated Averaging [11] [7], a key component that combines local stochastic gradient descent (SGD) computations on each client device with model averaging performed by a central server. By effectively integrating the contributions of multiple devices, federated averaging plays a crucial role in enabling federated learning to achieve its full potential.

C. Addressing Statistical Heterogeneity in FL

One major challenge in Federated Learning is the heterogeneity of clients' data, which has received a lot of attention from researchers [10], [9]. FedAvg [12], the primary federated optimization approach, tries to address this challenge by taking a weighted average of clients' updates to learn the global model. However, FedAvg has limitations in terms of convergence performance and speed when dealing with non-i.i.d. data. Some studies have utilized local batch normalization (BN) to address the issue of gradient drift in federated learning under non-IID data. For example, FedBN [14] trains all BN parameters locally on each client, while SiloBN [1] updates only the batch mean and variance locally, with the scale and shift parameters being shared globally for aggregation by the server.

D. Federated Domain Generalization

Domain generalization [3] aims to learn a model from multiple source domains such that it can directly generalize to unseen target domains. Previous efforts in this area have focused on learning domain-invariant representations by minimizing domain discrepancy across multiple source domains [6]. For instance, Motiian et al. [14] use a contrastive loss to reduce the distance between samples from the same class but different domains. There are other methods for addressing domain generalization that manipulate deep neural network architectures [32], leverage self-supervision signals [2] employ training heuristics [8], or implement data augmentation techniques [16]. These methods do not require centralization of data. For instance, Carlucci et al. [2] use self-supervised learning to solve jigsaw puzzles, and Zhang et al. [20] perform extensive data augmentations on each source domain through a series of transformations. When applied in the federated learning setting, these methods can act as regularization for local training with individual source domain data. However, they do not fully exploit the rich data distributions across domains. Our method, on the other hand, aims to transfer distribution information across clients to fully utilize the multi-source distributions for FedDG. Through experimentation, we have shown superior performance compared to these typical methods in the federated learning setting.

E. BiSeNet V2

Bilateral Segmentation Network (BiSeNet V2) [19] strikes a balance between speed and accuracy for efficient and effective semantic segmentation. It features two branches: the Detail Branch which designed to capture low-level details, such as edges and textures, using wide channels and shallow layers which results in high-resolution feature representation, while the Semantic Branch uses narrow channels and deep layers to obtain high-level semantic context from the input image and this branch is responsible for incorporating global information about the image, such as object categories and relationships between objects. The output from the two branches is then combined to produce the final segmentation results.

Fig. 1 shows a scheme of the structure of the network. There are three main components: a two-pathway backbone, an aggregation layer, and a booster part. The two-pathway backbone is divided into two branches, the Detail Branch, and the Semantic Branch, each with multiple stages. The Detail Branch has wide channels and is designed to capture low-level details, while the Semantic Branch has narrow channels and is responsible for incorporating high-level semantic context. The combination of these two branches results in a system that can produce high-quality segmentation results while maintaining real-time performance. The bilateral aggregation layer is used in the aggregation layer, and the booster part includes auxiliary segmentation heads to improve performance without additional cost.

III. Methodology and Experiments

A. Databases

Two different databases were used in this work, Cityscapes and GTA 5:

- Cityscapes: is a large-scale database which focus on semantic understanding of urban street environment. Consists of video sequences recorded in streets from fifty different European cities during different day times and climate conditions. The labeling for the images were done using 30 classes including objects like vehicles, bicycles, pedestrians and background like building, road, and vegetation. The goal of the model is to identify only nineteen classes, to do so, the Cityscapes dataset is mapped accordingly in order to ignore the classes of not interest and so the model only recognizes objects of interests.
- GTA 5: is a dataset of images captured from a computer game. Creating large datasets with pixel-level labels, such as Cityscapes, is extremely costly due to the amount of human effort required, [15] shows that associations between image patches can be reconstructed from the communication between the game and the graphics hardware. This enables rapid propagation of semantic labels within and across images synthesized by the game, with no access to the source code or the content. Such dataset

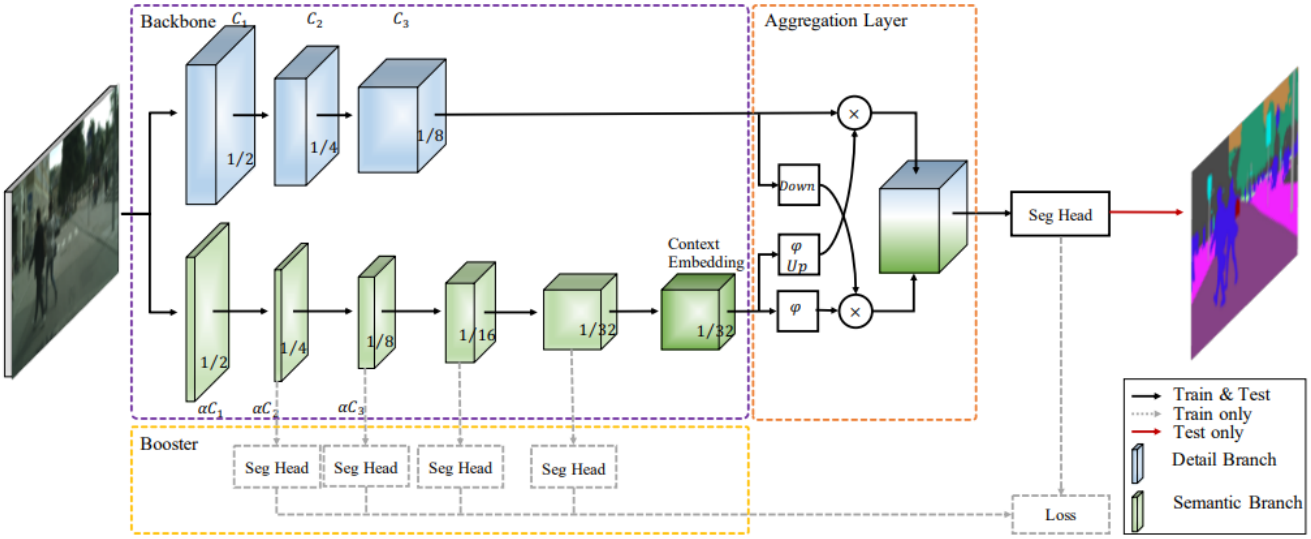

Fig. 1. Scheme of the Bilateral Segmentation Network.

can be used to train and test algorithms and models, with possible improvements when compared to training using only real world images.

B. Partition of Cityscapes dataset

Using 750 images from the Cityscapes dataset, two different splits are made, A and B, both of them containing training and test images. The test partition A is created by randomly picking two images from each city, with a total of 42 images, consequently the training partition A is composed by the remaining ones, being 708. The training and test partition B are created following the train and validation lists provided, with former having 500 images and the later with 250.

C. Centralized baseline

The centralized baseline is implemented using the BiSeNet V2. A preprocess on the images is done before feeding then to the network, a random horizontal flip, a random crop of size 512x1024 and a normalization of mean 0.5 and standard deviation 0.5, due to the fact that this "from the shelf" network was pretrained using the ImageNet Dataset, which applies the same normalization. The loss function used is the cross entropy loss and stochastic gradient descent (SGD) as the optimizer. The hyperparameters such as weight decay and momentum were taken from literature [19], batch size cannot be too large due to limited computational power so it is set to four, number of epochs are equal 30 and the step size is adjusted accordingly. Initial learning rate is set to 0.1 [11]. The network is trained with both training partitions A and B and then validated on the respective testing partition. Table I shows the average mIoU calculated over the entire test set together with the hyperparameters employed and figure 3 shows an example of the output of the network with the mean Intersection over Union close to 38%. The model trained using the split A performed better than the one trained with split B. Qualitative results are shown in figure 2.

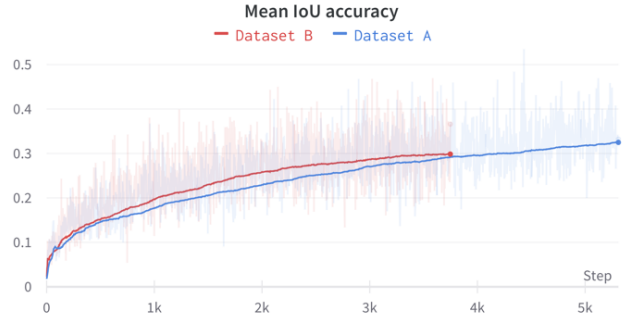

Fig. 2. —Mean IoU evolution during training with Cityscapes partitions A and B.

D. Federated Semantic Segmentation Experiments

FedDrive [5] proposed two different splits for the train partition. Heterogeneous split (I), in which each client is assigned images belonging to the same city but the number of images per client is not the same, thus representing the natural unbalance of federated learning method. And Uniform split (II), in which each client has the same number of images, collected from different cities. This promotes the similarity between data distribution on clients and avoid local bias. A similar but different approach towards the split is proposed for this work, in which the name of the split refers to the content of the images from one client.

I) Heterogeneous split: each client is randomly assigned no more than 20 images from different cities from the training set. Depending on the number of clients defined, the number of

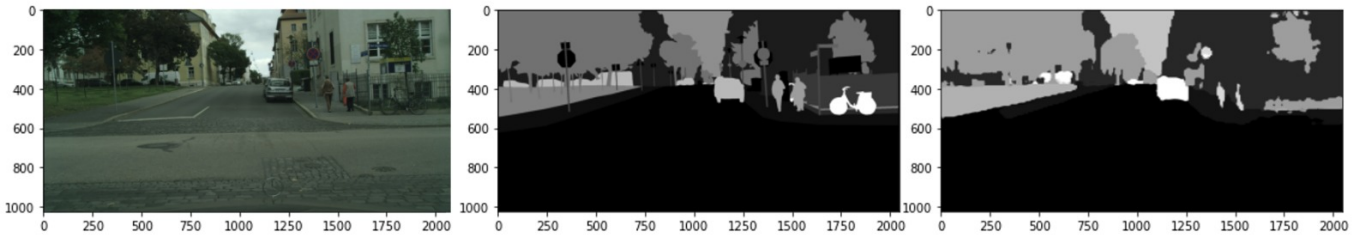

Fig. 3. Network output example with the centralized baseline result (mIoU = 38.31%).

TABLE I
RESULTS AND HYPERPARAMETERS OF CENTRALIZED BASELINE

Testing data	**Hyperparameters**							**Results**
	Learning rate	Moentum	Weight decay	Step Size	Gamma	Batch size	Epochs	mIoU
Cityscapes partition A	0.1	0.1	5.00E-04	10	0.1	4	30	0.324
Cityscapes partition B	0.1	0.1	5.00E-04	10	0.1	4	30	0.275

images per client will change, but keeping it the same for all clients.

II) Uniform split: each client is assigned no more than 20 images from the same city. The number of images per client will differ among them due to the fact that the number of images per city are not the same, and therefore some cities will have more clients than others, and some clients will have more images than others. Resulting in a non-i.i.d. domain distribution across clients.

Training sets A and B are divided both into uniform and heterogeneous splits following the concepts above. The test client has all the images not used at training time for both splits. Now with four different types of dataset distribution, the experiment can be started. It is first created a list of clients using one of the splits available, with hyperparameters from the previous step and some new hyperparameters such as epochs per client and number of rounds are chosen and available in table III. A round happens when the server uploads the main model on the selected clients, each client trains the model by the given number of epochs per client, all the selected clients send the updated weights to the server and then it aggregates all the weights using weighted average depending on the number of images used per client. It was selected 25 rounds per training, 4 epochs per clients and 5 clients per round. This training process is repeated with the remaining 3 splits. The validation of these trainings is done using the respective test set. Table II demonstrate that using the heterogeneous split, the model had a slightly better overall performance in terms of mIoU than the one training with the uniform split. This can be explained by the fact that model is able to generalize more when trained with mixed images than when trained with images from the same city. On the heterogeneous splits, the split A was again better than the split B.

TABLE II
RESULTS OF FEDERATED AVERAGING

Testing data	**Training data**	**Mean IoU**
Cityscapes Partition A	Uniform	0.133
	Heterogeneous	0.153
Cityscapes Partition B	Uniform	0.12
	Heterogeneous	0.133

TABLE III
HYPERPARAMETERS FOR FEDAVG

Hyperparameters	
Learning rate	0.1
Moentum	0.9
Weight decay	5.00E-04
Step Size	2
Gamma	0.1
Batch size	2
Epochs	4
Rounds	25

E. Federated Source-free Domain Adaptation – Pre-training phase

In the real world, self-driving cars do not have access to ground-truth labels. If they had ground-truth pseudo-labels, a semantic segmentation model would not be necessary. Moreover, manually labeling the images is a costly process, and it is not realistic to assume that the clients have access to the labels associated with the images they collect. To this end, it is used the synthetic dataset GTA 5. It is first mapped accordingly to the same classes as Cityscapes. The model is trained from scratch the same way done for the centralized baseline, using the same hyperparameters and transformations, but this time with the new dataset. The validation results are in the table IV in which the model was tested with both test partitions A and B as well as with its own dataset. Figure 5 shows the label

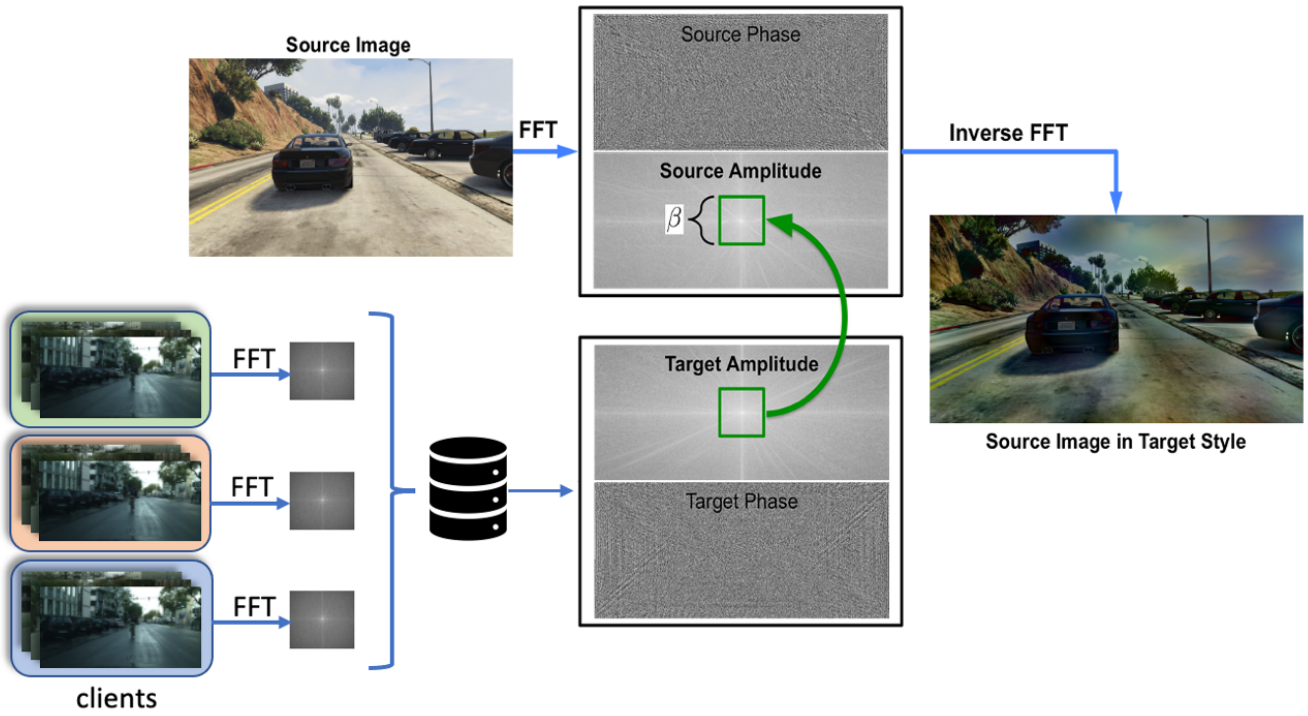

Fig. 4. Domain Adaptation using FFT.

predicted by the network when trained with the GTA5 dataset.

TABLE IV
RESULTS FROM GTA5 DATASET FOR TRAINING WITHOUT FDA

Testing data	Training data	Mean IoU
GTA5	GTA5	0.25
Cityscapes Partition A	GTA5	0.116
Cityscapes Partition B	GTA5	0.101

However, simply training the model on the source data does not yield satisfactory performance on the target data, due to the covariant shift [18], and it is reasonable to assume that the server does not have access to the data coming from the clients due to privacy issues. To this end, the idea of FDA is introduced to reduce the domain gap between the two datasets. This method consists in computing the fast Fourier transform (fft) of each input image, replace the low-level frequency of the target image into the source image, and with an inverse fast Fourier transform, it is possible to reconstruct the new image used from training. In this work, instead of applying this method directly from one image to another, it is first created a bank of styles, in which each style comes from the average of styles of a single client. Figure 4 illustrates how this dynamic works. Then, it is created a new dataset applying randomly, to every image, one of the styles from the bank of styles. Since there are four different splits, there are consequently four different bank of styles and thus four different new GTA 5 datasets. The window size to be used on the amplitude spectrum of the image can be selected through the parameter L, which multiplies the smallest aspect of the image (height or width), to get the size of the square. [18] tested different values of size window and showed that larger values generalized better if trained from scratch but induces more bias when combined with self-supervised training, therefore it is selected L = 0.01 as a default value for all the trainings. Results are displayed in table V.

TABLE V
RESULTS FROM GTA5 DATASET FOR TRAINING WITH FDA

Testing data	Training data (style)	Mean IoU
Cityscapes Partition A	GTA5 (A uniform)	0.185
	GTA5 (A heterogeneous)	0.18
Cityscapes Partition B	GTA5 (B uniform)	0.181
	GTA5 (B heterogeneous)	0.175

With the Fourier Domain Adaptation technique, the value of mIoU almost doubled when in comparison with the test using standard GTA5 images, showing the great benefit of using FDA to reduce the domain gap between source and target. Figure 6 shows the comparison of evolution of the mean IoU when training the models. It means that all three networks trained well regarding its own dataset, but the ones trained using the new images from the domain adaptation were able to generalize better across domain.

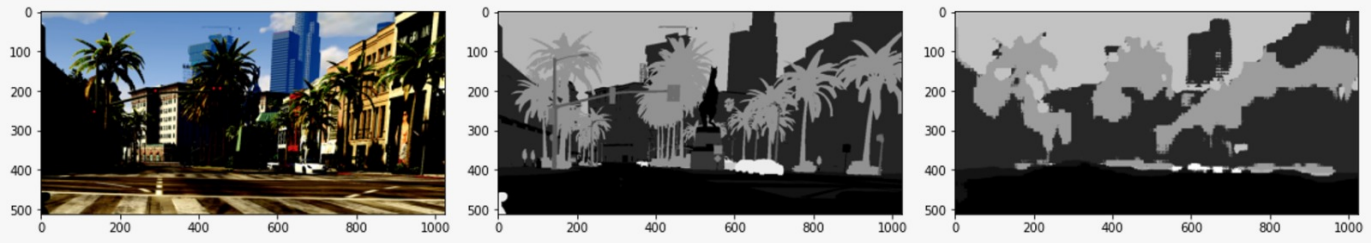

Fig. 5. Network output example when trained with the GTA5 dataset (mIoU = 25.01%).

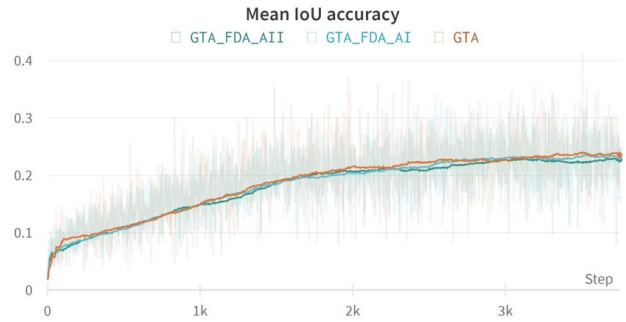

Fig. 6. Mean IoU evolution on GTA5 dataset training with and without FDA.

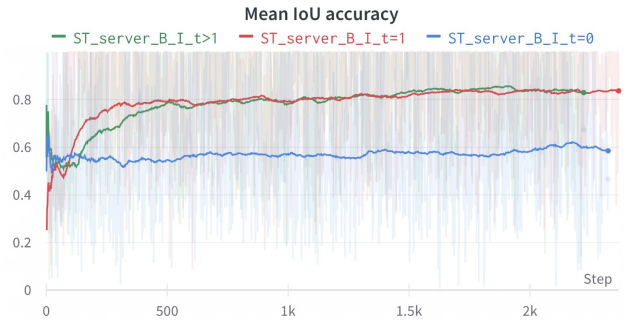

Fig. 7. Mean IoU evolution for different update rates of the teacher model (T value variation).

F. Federated self-training using pseudo-labels

In this final step, it is considered a more realistic scenario in which the client does not have access to the label of the images, so the teacher model is used. The teacher model is the best model[1] loaded from the previous step where the model was trained using standard GTA 5 images. It is responsible to generate a pseudo-label by running it with an image from Cityscapes and giving a prediction of labels to it. This prediction is filtered using a threshold and then fed to the student model as a ground-truth label. The update on the teacher model is done in three different approaches:

i) Teacher model never updated (T = 0)

ii) Teacher model = server model at the beginning of each round (T=1)

iii) Teacher model = server model at every T rounds (T>1)

The best way to update the teacher model is selected to be used for the remaining training splits. As a further step, the same training process is repeated using the best network achieved by the training with GTA 5 dataset with domain adaptation. Results in figure 7 demonstrate that the best way of improving the mIoU on the student model is to update the teacher model, and also that the network trained with the dataset after applying the domain adaptation outperforms the one trained only with the synthetic dataset.

[1]Since all the trainings are done in a relatively small number of epochs or rounds due to limited GPU power, it is reasonable to assume that the best model achieved on the training will be close to the one from the last update, since when the training is stopped, the loss is still going down and the mIoU is still going up, therefore the last update is used.

IV. Conclusion

From the results analyzed in section III it is possible to conclude the following for each task:

- Centralized Baseline: the network improved its mean IoU values throughout the training process, and both tests in A and B partitions are satisfactory, with mIoU values of 32.4 % for the first and 27.5 % for the second. The better result from the A partition is expected due to the greater size of its training partition, in comparison with B.
- Federated Averaging: when splitting the datasets into uniform and heterogeneous clients (splits) the results for the heterogeneous splits present higher mIoU values (15.3 % and 13.3 % for A and B partitions, against 13.3% and 12% for the uniform splits), due to the greater variation of the images available for training in each client. Thus the server model performed better when trained with an heterogeneous split. The results from partition A are higher for the same reason of the Centralized Baseline task.
- Fourier Domain Adaptation: the network is tested with the Cityscapes A and B test partitions, as in the other cases. It performed better when the styles of the splits were applied to the GTA5 dataset for the training phase, as expected. Thus, the implementation of FDA in the training data improved the performance of the network when training in the A and B test partitions.
- Self Training: for this task, the update of the teacher model for every round of the training phase increased the performance of the network, reducing its loss and

increasing its mIoU values throughout the training phase, suggesting that the pseudo labels are more accurate when the model is updated more often.

REFERENCES

[1] Mathieu Andreux et al. "Siloed federated learning for multi-centric histopathology datasets". In: *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2*. Springer. 2020, pp. 129–139.

[2] Fabio M Carlucci et al. "Domain generalization by solving jigsaw puzzles". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2229–2238.

[3] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. "Learning to balance specificity and invariance for in and out of domain generalization". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer. 2020, pp. 301–318.

[4] Marius Cordts et al. "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[5] Lidia Fantauzzo et al. "FedDrive: generalizing federated learning to semantic segmentation in autonomous driving". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 11504–11511.

[6] Muhammad Ghifary et al. "Domain generalization for object recognition with multi-task autoencoders". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2551–2559.

[7] Ying He et al. "Bift: A blockchain-based federated learning system for connected and autonomous vehicles". In: *IEEE Internet of Things Journal* 9.14 (2021), pp. 12311–12322.

[8] Zeyi Huang et al. "Self-challenging improves cross-domain generalization". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer. 2020, pp. 124–140.

[9] Tian Li et al. "Federated learning: Challenges, methods, and future directions". In: *IEEE signal processing magazine* 37.3 (2020), pp. 50–60.

[10] Xiaoxiao Li et al. "Fedbn: Federated learning on non-iid features via local batch normalization". In: *arXiv preprint arXiv:2102.07623* (2021).

[11] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[12] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[13] Umberto Michieli and Mete Ozay. "Prototype guided federated learning of visual feature representations". In: *arXiv preprint arXiv:2105.08982* (2021).

[14] Saeid Motiian et al. "Unified deep supervised domain adaptation and generalization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5715–5725.

[15] Stephan R Richter et al. "Playing for data: Ground truth from computer games". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer. 2016, pp. 102–118.

[16] Shiv Shankar et al. "Generalizing across domains via cross-gradient training". In: *arXiv preprint arXiv:1804.10745* (2018).

[17] Jun Xie et al. "Semantic instance annotation of street scenes by 3d to 2d label transfer". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 3688–3697.

[18] Yanchao Yang and Stefano Soatto. "Fda: Fourier domain adaptation for semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4085–4095.

[19] Changqian Yu et al. "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation". In: *International Journal of Computer Vision* 129 (2021), pp. 3051–3068.

[20] Ling Zhang et al. "Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation". In: *IEEE transactions on medical imaging* 39.7 (2020), pp. 2531–2540.