Federated Learning for Autonomous Driving:
Enhancing Style Transfer Techniques for FFreeDa

MLDL Project Track 2B, Summer ‘23

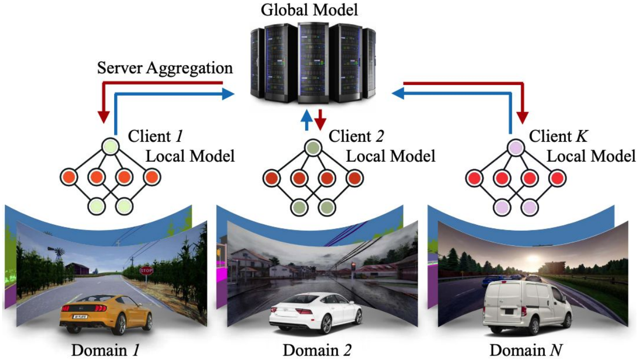
Luca Agnese, Fabio Rizzi, Flavio Spuri

OVERVIEW

SS

Semantic segmentation (SS) involves **assigning a class to each pixel** in an image. This task is critical for numerous applications, particularly **self-driving vehicles**, where it aids in precisely identifying key components like pedestrians, road signs, and cars.

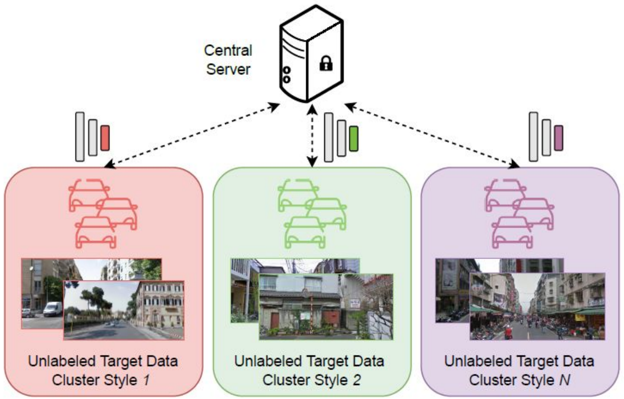
FL



Federated Learning (FL) is a learning paradigm in which the task is solved through a collaboration between several devices, called **clients**, coordinated by a central **server**. Each client works locally on its own data without the need of transmitting it. This allows the data to remain unseen by the server which solves the issue of respecting **users' privacy**.

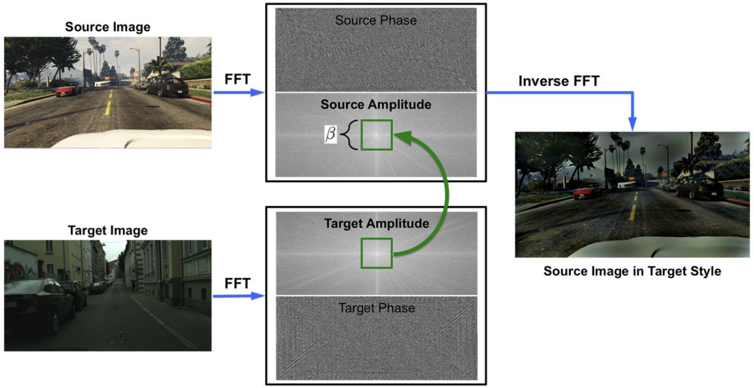
OVERVIEW

FFreeDA



Federated source-Free Domain Adaptation (FFreeDA) is a task where the clients only access their **unlabelled target** dataset while the server is **pre-trained on a labelled source** dataset.

FDA



Fourier Domain Adaptation (FDA) swap the target image **style** onto the source image to **reduce the gap** between the two distributions.

Dataset – IDDA

- Large **synthetic dataset** with over **one million** images labelled for SS
- More than **100 scenarios**, defined via **three axis**:
  - town
  - weather and illumination
  - viewpoint (car)

- We only consider a **subset**:
- 24 train clients (***Train***), each with 25 images
- 2 test clients, each with 120 images
- One with the same scenarios as *Train* (***Test Same Dom***)
- One with different scenarios (***Test Diff Dom***)

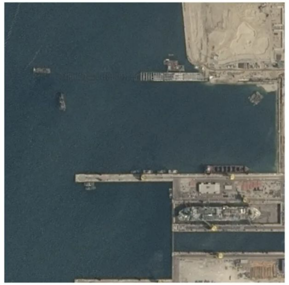
Dataset - GTAV

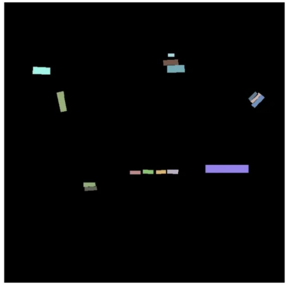
- Large **synthetic dataset** with almost **25k** labeled images.
- Rendered from the highly realistic video game Grand Theft Auto V.
- All images are from the car perspective in the street of American-style cities.
- We only consider a subset of **500 samples**.

METRICS

Pixel Accuracy (pAcc)

pAcc is a metric used in SS that calculates the **ratio** of **correctly classified** pixels to the **total** number of pixels. While very straightforward and easy to compute, it is unable to take into account **class imbalance**, possibly yielding misleading results.





Intersection-Over-Union (IoU)

IoU is the most commonly used metric in SS. While remaining pretty straightforward, it overcomes the issue of treating unbalanced classes. For a given class c, it is defined as the **ratio** between the area of **overlap** and the area of **union** between **prediction** and **ground truth**:

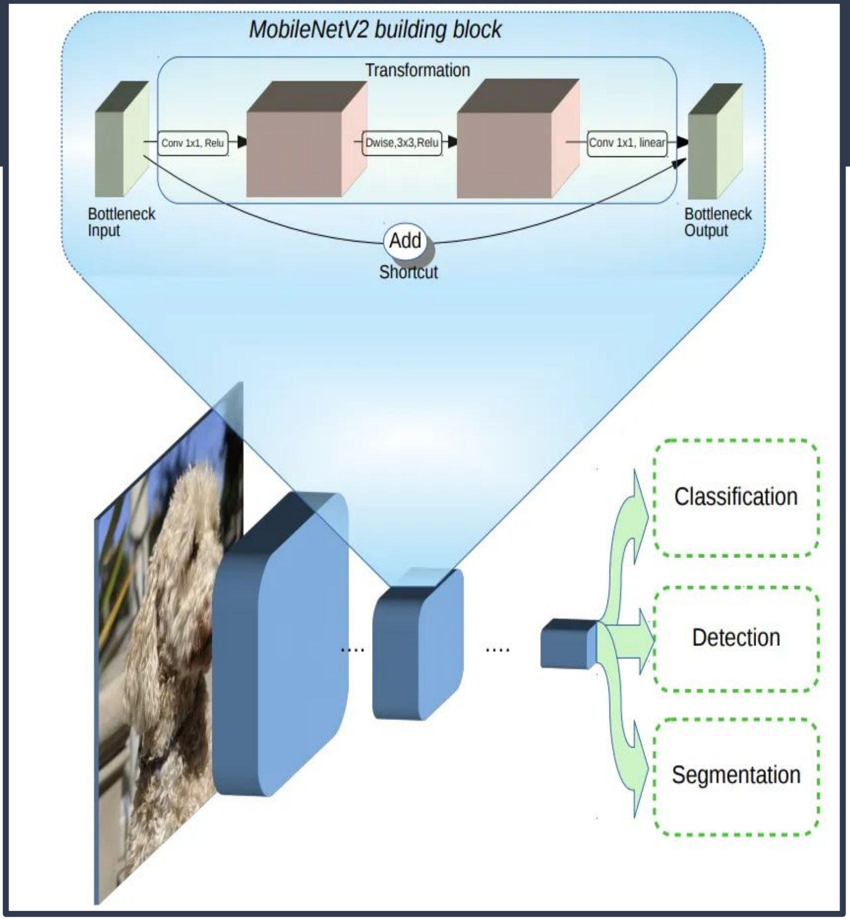
$$IoU_c = \frac{|A \cap B|}{|A \cup B|}$$

where A is the set of pixels assigned with class c and B is the set of pixels with label class c. Our main evaluation metric is the mean IoU (**mIoU**) between all 16 semantic classes.

MobileNetV2

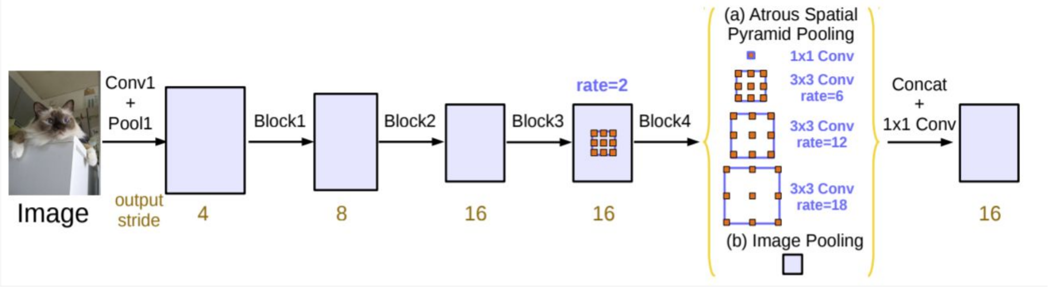
- **Lightweight CNN** that balances accuracy and computational costs.

- Employs **depth-wise convolutions** and **inverted residuals** to achieve such balance.

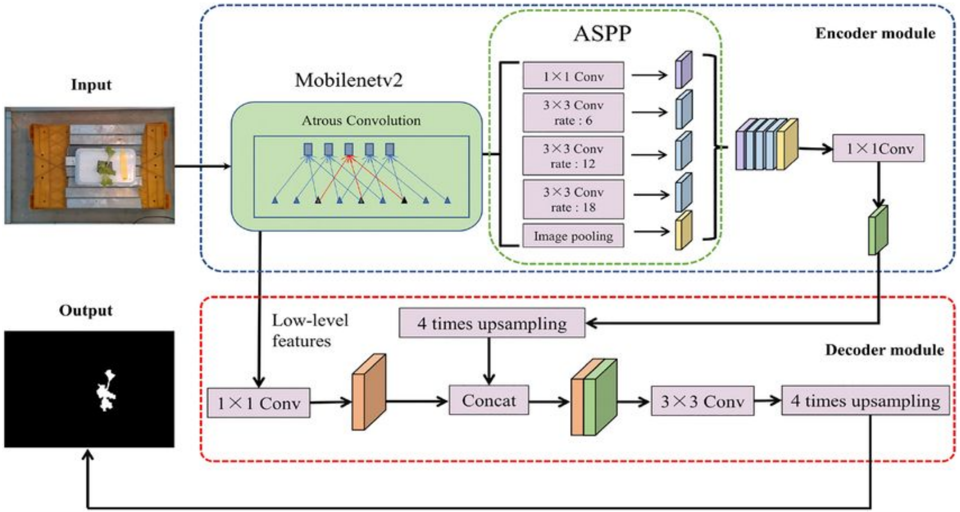


DeepLabV3

- **State-of-the-art** architecture developed by Google for SS.
- Atrous convolution (or **dilated convolution**) to control its field-of-view and efficiently capture multi-scale information.
- Coupled with a feature called atrous spatial pyramid pooling (**ASPP**) to capture objects and contexts at different scales.



Task 1
Centralized Baseline



- Method
- Results

Centralized baseline – HPO

1. First, we perform a **coarse grid search** to get a sense of the problem. In it, we try **several configurations** for a very **limited number of epochs**, and discard those that give a poor initialization.
2. Then, we train the **best configurations** for both optimizers for a larger number of epochs, thus selecting the best configuration of **lr**, **wd**, and **optimizer**.
3. Finally, we run for a **smaller number of configurations** and for a **greater number of epochs**, to select the best **lr scheduler** and **data augmentation**.

Parameter	Values
lr	[0.1, 0.01, 0.001, 0.0001]
wd	[0.0001, 0.00001, 0]
optimizers	SGD(momentum = 0.9) Adam($\beta 1 = 0.9, \beta 2 = 0.999, \epsilon = 10^{-8}$) Adam($\beta 1 = 0.9, \beta 2 = 0.99, \epsilon = 10^{-1}$) Adam($\beta 1 = 0.9, \beta 2 = 0.99, \epsilon = 10^{-5}$)
LRdecay	[StepLR, PolyLR]
DataAugmentation	basic = {RandomResizedCrop} advanced = {RandomResizedCrop,RandomHorizontalFlip, ColorJitter}

Centralized baseline – Results

Θ_c					Number of Training Epochs	Eval mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	**Wd**	**Opt**	**Sched**	**Set of Transforms**				
0.1	0.0001	SGD(m=0.9)	PolyLR	advanced	100	0.5807 ± 0.0028	0.5087 ± 0.0018	0.3211 ± 0.0031

- We identify the set of hyper-parameters $\boldsymbol{\Theta_c}$.

Centralized baseline – Results

Θ_c					Number of Training Epochs	Eval mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	**Wd**	**Opt**	**Sched**	**Set of Transforms**				
0.1	0.0001	SGD(m=0.9)	PolyLR	advanced	100	0.5807 ± 0.0028	0.5087 ± 0.0018	0.3211 ± 0.0031

- We identify the set of hyper-parameters $\boldsymbol{\Theta_c}$.
- We obtain some **good results on *Test Same Dom***, comparable to those on the Eval set.

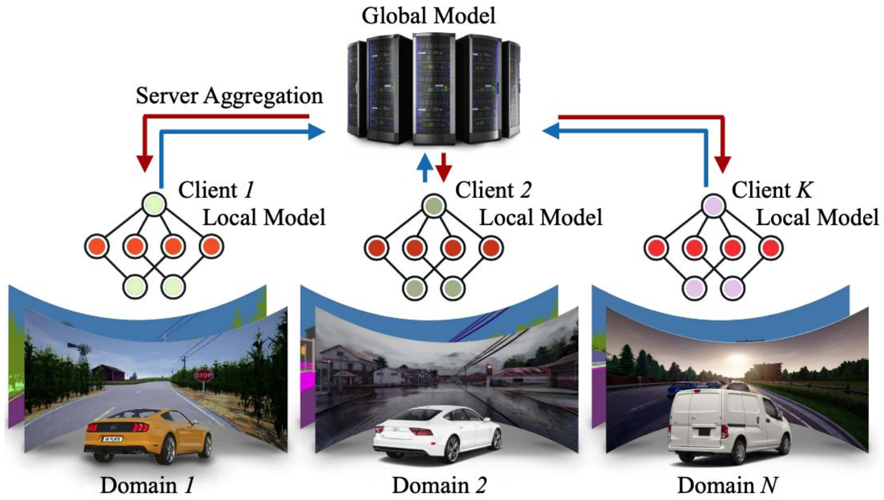
Centralized baseline – Results

Θ_c					Number of Training Epochs	Eval mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	**Wd**	**Opt**	**Sched**	**Set of Transforms**				
0.1	0.0001	SGD(m=0.9)	PolyLR	advanced	100	0.5807 ± 0.0028	0.5087 ± 0.0018	0.3211 ± 0.0031

- We identify the set of hyper-parameters $\boldsymbol{\Theta_c}$.
- We obtain some **good results on *Test Same Dom***, comparable to those on the Eval set.
- However, there is a pretty **significant reduction** in the performances **on *Test Diff Dom*.**

Task 2
Supervised FL

- Method
- Results



Supervised FL – Algorithm

Θ_c				
Lr	**Wd**	**Opt**	**Sched**	**Set of Transforms**
0.1	0.0001	SGD(m=0.9)	PolyLR	advanced

Maintaining the set of hyper-parameters $\boldsymbol{\Theta_c}$, for each round of communication $t \in$ *[1,num rounds]*:

1. The server **selects** a certain number of **clients.**
2. The server **sends** the current model **parameters** ω^{t-1} to the selected clients.
3. Each client c performs a *num_epochs* number of local epochs, obtaining the updated parameters ω^t_c.
4. At the end of each round, the server **collects** all the **client-updated parameters**. $\{\omega^t_c \mid c = 1, ..., \text{clients per round}\}$ and **aggregates** them by computing their **average**, weighted by the clients' cardinalities.
   - **equivalent** to updating the central model using the **average gradient.**

Supervised FL - Results
Fixed Clients per round

Clients per round	Number of rounds	Local Epochs	Eval mIoU (train partition)	Test Same Dom mIoU	Test Diff Dom mIoU
2	30	1	0.3109 ± 0.0109	0.2835 ± 0.0158	0.1953 ± 0.0317
		3	0.3585 ± 0.0052	0.3416 ± 0.0023	0.2470 ± 0.0085
		6	0.3868 ± 0.0032	0.3500 ± 0.0024	0.2556 ± 0.0017
4	30	1	0.3300 ± 0.0053	0.3077 ± 0.0114	0.2192 ± 0.0205
		3	0.3725 ± 0.0029	0.3472 ± 0.0057	0.2588 ± 0.0122
		6	0.3795 ± 0.0074	0.3455 ± 0.0094	0.2647 ± 0.0160
8	30	1	0.3426 ± 0.0031	0.3202 ± 0.0053	0.2382 ± 0.0095
		3	0.3773 ± 0.0020	0.3542 ± 0.0035	0.2543 ± 0.0126
		6	0.4035 ± 0.0031	0.3715 ± 0.0030	0.2682 ± 0.0089

- **Obvious improvement** when increasing either clients per round or local epochs.

Supervised FL – Results
Fixed Clients per round

Clients per round	Number of rounds	Local Epochs	Eval mIoU (train partition)	Test Same Dom mIoU	Test Diff Dom mIoU
2	30	1	0.3109 ± 0.0109	0.2835 ± 0.0158	0.1953 ± 0.0317
		3	0.3585 ± 0.0052	0.3416 ± 0.0023	0.2470 ± 0.0085
		6	0.3868 ± 0.0032	0.3500 ± 0.0024	0.2556 ± 0.0017
4	30	1	0.3300 ± 0.0053	0.3077 ± 0.0114	0.2192 ± 0.0205
		3	0.3725 ± 0.0029	0.3472 ± 0.0057	0.2588 ± 0.0122
		6	0.3795 ± 0.0074	0.3455 ± 0.0094	0.2647 ± 0.0160
8	30	1	0.3426 ± 0.0031	0.3202 ± 0.0053	0.2382 ± 0.0095
		3	0.3773 ± 0.0020	0.3542 ± 0.0035	0.2543 ± 0.0126
		6	0.4035 ± 0.0031	0.3715 ± 0.0030	0.2682 ± 0.0089

- **Obvious improvement** when increasing either clients per round or local epochs.
- **Not always** seems to **justify** the increased computational effort.

Supervised FL – Results

Fixed Clients per round

Clients per round	Number of rounds	Local Epochs	Eval mIoU (train partition)	Test Same Dom mIoU	Test Diff Dom mIoU
2	30	1	0.3109 ± 0.0109	0.2835 ± 0.0158	0.1953 ± 0.0317
		3	0.3585 ± 0.0052	0.3416 ± 0.0023	0.2470 ± 0.0085
		6	0.3868 ± 0.0032	0.3500 ± 0.0024	0.2556 ± 0.0017
4	30	1	0.3300 ± 0.0053	0.3077 ± 0.0114	0.2192 ± 0.0205
		3	0.3725 ± 0.0029	0.3472 ± 0.0057	0.2588 ± 0.0122
		6	0.3795 ± 0.0074	0.3455 ± 0.0094	0.2647 ± 0.0160
8	30	1	0.3426 ± 0.0031	0.3202 ± 0.0053	0.2382 ± 0.0095
		3	0.3773 ± 0.0020	0.3542 ± 0.0035	0.2543 ± 0.0126
		6	0.4035 ± 0.0031	0.3715 ± 0.0030	0.2682 ± 0.0089

- **Obvious improvement** when increasing either clients per round or local epochs.
- **Not always** seems to **justify** the increased computational effort.
- Particularly evident when increasing **clients per round.**

Supervised FL - Results
Long Experiments and Target mIoU

Long experiments

Clients per round	Number of rounds	Local Epochs	Eval mIoU (train partition)	Test Same Dom mIoU	Test Diff Dom mIoU
2	100	3	0.4434 ± 0.0071	0.3962 ± 0.0093	0.2667 ± 0.0178
8	100	3	0.4901 ± 0.0041	0.4485 ± 0.0038	0.2946 ± 0.0045

- If the primary concern is **time** constraints, it is clearly better to **select a larger number of clients** in each round.

Target mIoU - Train until reached

Local Epochs	Target mIoU	Client per Round	# Communications
3	0.38	2	**82**
		4	144
		8	256

- Increase in the accuracy is **not proportional** to the augment of the computational cost.
- The most **efficient** approach appears to be **keeping the clients per round low** and executing a larger number of rounds.

Supervised FL - Results
Comparison with Baseline

Centralised

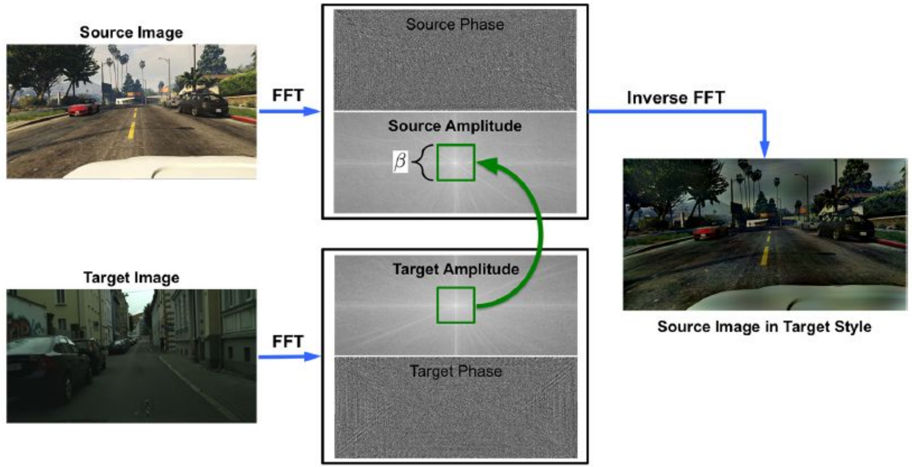
Θ_c					Number of Training Epochs	Eval mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms				
0.1	0.0001	SGD(m=0.9)	PolyLR	advanced	100	0.5807 ± 0.0028	0.5087 ± 0.0018	0.3211 ± 0.0031

Federated

Clients per round	Number of rounds	Local Epochs	Eval mIoU (train partition)	Test Same Dom mIoU	Test Diff Dom mIoU
2	100	3	0.4434 ± 0.0071	0.3962 ± 0.0093	0.2667 ± 0.0178
8	100	3	0.4901 ± 0.0041	0.4485 ± 0.0038	0.2946 ± 0.0045

Task 3
Moving Towards FFreeDA

- Motivation
- Method
- Results



T3 - Motivation and Method

- It’s unrealistic to assume to have ground truth labels on client side. The **clients** have **access** only to their **unlabelled *target*** dataset.

T3 - Motivation and Method

- It's unrealistic to assume to have ground truth labels on client side. The **clients** have **access** only to their **unlabelled *target*** dataset.

- In this task we first **train** the model on the labelled synthetic dataset **GTAV**, which is the *source* dataset, with a centralized approach.

T3 - Motivation and Method

- It's unrealistic to assume to have ground truth labels on client side. The **clients** have **access** only to their **unlabelled *target*** dataset.

- In this task we first **train** the model on the labelled synthetic dataset **GTAV**, which is the *source* dataset, with a centralized approach.

- Then, we try to **reduce the discrepancy** between IDDA (*target*) and *source* by **applying** Fourier Domain Adaptation (**FDA**) technique.

T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a. 0.000001	100	0.5824 ± 0.0056 0.5680 ± 0.0030	0.2665 ± 0.0036 0.2708 ± 0.0046	0.2633 ± 0.0036 0.2772 ± 0.0043	0.2050 ± 0.0030 0.1954 ± 0.0028

- We run again **HPO**, validating on IDDA train dataset. We follow the strategy used for the centralised baseline.

T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a. 0.000001	100	0.5824 ± 0.0056 0.5680 ± 0.0030	0.2665 ± 0.0036 0.2708 ± 0.0046	0.2633 ± 0.0036 0.2772 ± 0.0043	0.2050 ± 0.0030 0.1954 ± 0.0028

- We run again **HPO**, validating on IDDA train dataset. We follow the strategy used for the centralised baseline.
- We obtain a set of hyper-parameters $\boldsymbol{\Theta}_{pt}$

T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a.	100	0.5824 ± 0.0056	0.2665 ± 0.0036	0.2633 ± 0.0036	0.2050 ± 0.0030
					0.000001		0.5680 ± 0.0030	0.2708 ± 0.0046	0.2772 ± 0.0043	0.1954 ± 0.0028

- We run again **HPO**, validating on IDDA train dataset. We follow the strategy used for the centralised baseline..
- We obtain a set of hyper-parameters Θ_{pt}
- The results highlight a **large dip** on IDDA, especially on *Test Diff Dom*.

T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a. 0.000001	100	0.5824 ± 0.0056 0.5680 ± 0.0030	0.2665 ± 0.0036 0.2708 ± 0.0046	0.2633 ± 0.0036 0.2772 ± 0.0043	0.2050 ± 0.0030 0.1954 ± 0.0028

- Regarding the application of **FDA**, we first **tune** the hyper-parameter $\boldsymbol{\beta}$ considering a limited number of epochs.

- β represents the **size** of the frequency spectrum **window** replaced

Table 6. Beta Tuning

β	Eval mIoU
0.0000001	0.2539
0.000001	0.2679
0.00001	0.2612
0.01	0.2598

T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a. 0.000001	100	0.5824 ± 0.0056 0.5680 ± 0.0030	0.2665 ± 0.0036 0.2708 ± 0.0046	0.2633 ± 0.0036 0.2772 ± 0.0043	0.2050 ± 0.0030 0.1954 ± 0.0028

- Regarding the application of **FDA**, we first **tune** the hyper-parameter β considering a limited number of epochs.
- β represents the **size** of the frequency spectrum **window** replaced

- Once fixed the value for the beta parameter we run for a larger number of epochs.

T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a. 0.000001	100	0.5824 ± 0.0056 0.5680 ± 0.0030	0.2665 ± 0.0036 0.2708 ± 0.0046	0.2633 ± 0.0036 0.2772 ± 0.0043	0.2050 ± 0.0030 0.1954 ± 0.0028

- First, we note a slight **decrease** in performance on the **source train** dataset as the images used in training get slightly modified.

T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a. 0.000001	100	0.5824 ± 0.0056 0.5680 ± 0.0030	0.2665 ± 0.0036 0.2708 ± 0.0046	0.2633 ± 0.0036 0.2772 ± 0.0043	0.2050 ± 0.0030 0.1954 ± 0.0028

- First, we note a slight decrease in performance on the **source train** dataset as the images used in training get slightly modified.

- On the other hand, we observe an **increase** on the target test dataset, in particular on the ***Test Same Dom*** which was our aim in this task.

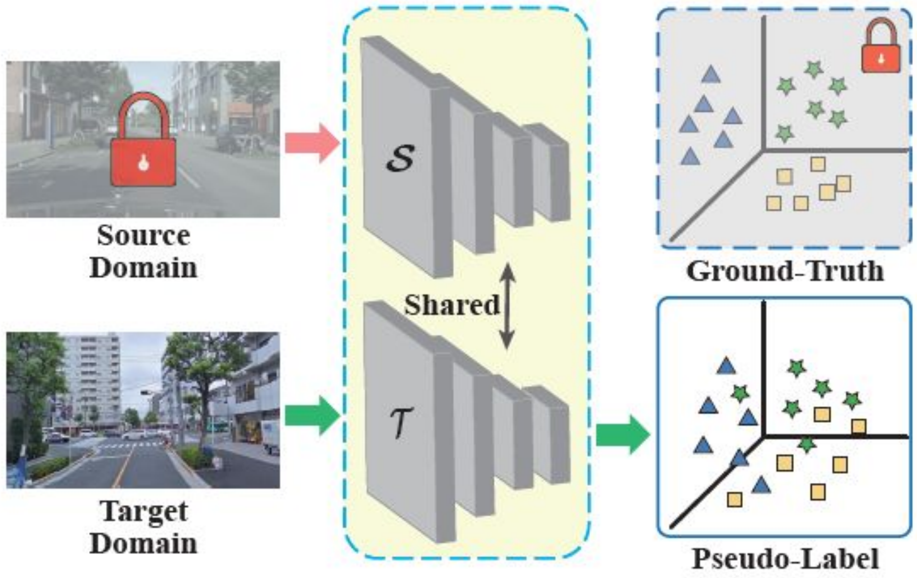
T3 – Results

Θ_{pt}					β	Training Epochs	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
Lr	Wd	Opt	Sched	Set of Transforms						
0.01	0.0001	SGD(m=0.9)	PolyLR	advanced	n.a. 0.000001	100	0.5824 ± 0.0056 0.5680 ± 0.0030	0.2665 ± 0.0036 0.2708 ± 0.0046	0.2633 ± 0.0036 0.2772 ± 0.0043	0.2050 ± 0.0030 0.1954 ± 0.0028

- First, we note a slight decrease in performance on the **source train** dataset as the images used in training get slightly modified.
- On the other hand, we observe an **increase** on the target test dataset, in particular on the ***Test Same Dom*** which was our aim in this task.
- The performance on the **Test Diff Dom** sees a small **decrease**.

Task 4
Federated Self-Training using Pseudo-Labels

- Motivation
- Method
- Results



T4 - Motivation and Method

- In this task we move toward the Self Supervised (Federated) Learning paradigm. Here we consider the **IDDA Train** dataset to be **unlabeled**.

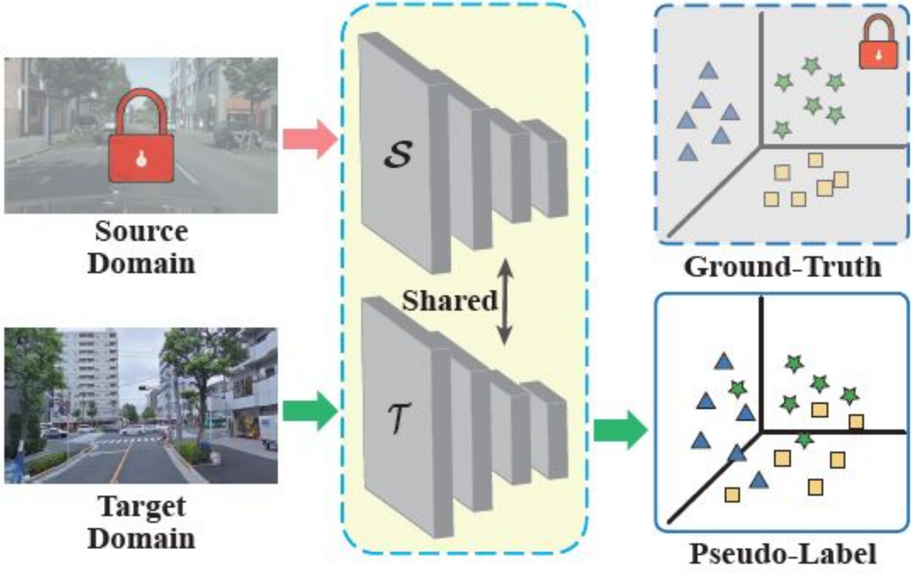
T4 - Motivation and Method

- In this task we move toward the Self Supervised (Federated) Learning paradigm. Here we consider the I**DDA Train** dataset to be **unlabeled**.

- In this framework we have two components: a ***student*** model and a ***teacher*** model.

T4 - Motivation and Method

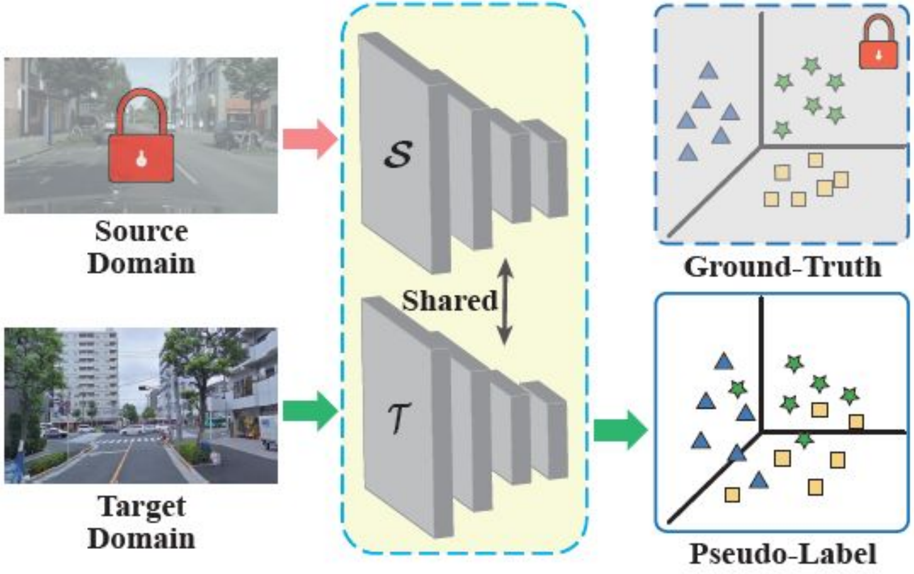
- Both the teacher and the student model are initialized using the best performing **model pre-trained** on **GTAV**.



T4 - Motivation and Method

- Both the teacher and the student model are initialized using the best performing **model pre-trained** on **GTAV**.

- The **teacher** computes the **pseudo-labels** upon which the student learns, still using the same algorithm used for **FL**.



T4 - Motivation and Method

- In this task we move toward the Self Supervised (Federated) Learning paradigm. Here we consider the I**DDA Train** dataset to be **unlabeled**.

- In this framework we have two components: a ***student*** model and a ***teacher*** model.

- We consider **three** different **strategies** to **update** the teacher model:
  - Never updated.
  - At the beginning of each FL round, set Teacher = Student.
  - Every **T** > 1 FL rounds, set Teacher = Student.

T4 - Results

Clients per round	T	lr	FDA	Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
2	$+\infty$	0.00001	X	0.3202	0.2904	0.2206
			✓	0.3350	0.2964	0.2026
8	5	0.1	X	0.0066	0.0061	0.0058
			✓	0.0160	0.0164	0.0167

- We run **two sets of experiments,** considering the best performing pre-trained model with and without FDA.

T4 - Results

Clients per round	T	lr	FDA	Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
2	$+\infty$	0.00001	X	0.3202	0.2904	0.2206
			✓	0.3350	0.2964	0.2026
8	5	0.1	X	0.0066	0.0061	0.0058
			✓	0.0160	0.0164	0.0167

- We run **two sets of experiments**, considering the best performing pre-trained model with and without FDA
- For these experiments, we use the same set of hyperparameters Θ_c, set the number of local epochs to 1 and vary the clients per round in *{2 , 8}*.

T4 - Results

Clients per round	T	lr	FDA	Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
2	$+\infty$	0.00001	X	0.3202	0.2904	0.2206
			✓	0.3350	0.2964	0.2026
8	5	0.1	X	0.0066	0.0061	0.0058
			✓	0.0160	0.0164	0.0167

- We run **two sets of experiments**, considering the best performing pre-trained model with and without FDA
- For these experiments, we use the same set of hyperparameters Θ_c, set the number of local epochs to 1 and vary the clients per round in *{2 , 8}*.
- We **test** all three **teacher update** strategies. For the third strategy, we consider two values of T, specifically T = 5 and T = 15.

T4 – Results

Clients per round	T	lr	FDA	Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
2	$+\infty$	0.00001	X	0.3202	0.2904	0.2206
			✓	0.3350	0.2964	0.2026
8	5	0.1	X	0.0066	0.0061	0.0058
			✓	0.0160	0.0164	0.0167

- Using the same set of hyperparameters Θ_c we note that the performance decays very quickly. We identify the **problem** to be a **too-large** value of **lr** = 0.1.

T4 – Results

Clients per round	T	lr	FDA	Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
2	$+\infty$	0.00001	X	0.3202	0.2904	0.2206
			✓	0.3350	0.2964	0.2026
8	5	0.1	X	0.0066	0.0061	0.0058
			✓	0.0160	0.0164	0.0167

- Using the same set of hyperparameters Θ_c we note that the performance decays very quickly. We identify the **problem** to be a **too-large** value of **lr** = 0.1.
- We then try to **tune again** the value of **lr** which we eventually set to lr = 0.00001. We run again the experiments for a considerable number of rounds.

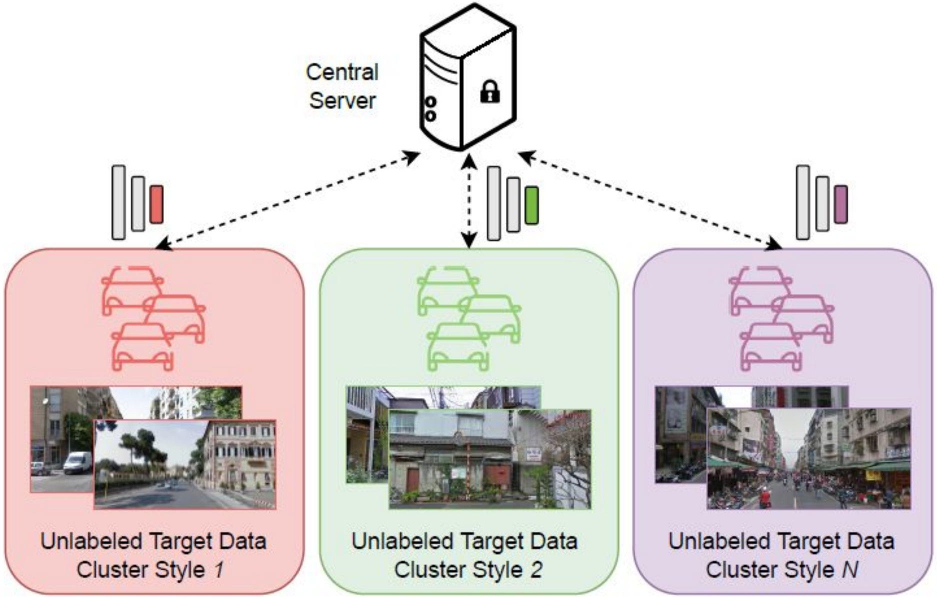
T4 - Results

Clients per round	T	lr	FDA	Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
2	$+\infty$	0.00001	X	0.3202	0.2904	0.2206
			✓	0.3350	0.2964	0.2026
8	5	0.1	X	0.0066	0.0061	0.0058
			✓	0.0160	0.0164	0.0167

- Using the same set of hyperparameters Θ_c we note that the performance decays very quickly. We identify the **problem** to be a **too-large** value of **lr** = 0.1.
- We then try to **tune again** the value of **lr** which we eventually set to lr = 0.00001. We run again the experiments for a considerable number of rounds.
- We report how across our experiments this method **does not** bring any significant **improvement**, generally being **detrimental** to the **performances**, as the pseudo-labels and the predictions are almost the same except for some small **randomicity**, which eventually **degrades** the model **performance**.

Extension 1
Ensemble Learning

- Method
- Results



Ensemble Learning – Method

- The proposed method involves creating an **ensemble** learning **model** based on a new **clustering scheme**, which originates from an existing work (**LADD**).

Ensemble Learning – Method

- The proposed method involves creating an **ensemble** learning **model** based on a new **clustering scheme**, which originates from an existing work (**LADD**).

- The different **styles** of training data are **grouped** into clusters, obtained as follows:
  - **K-means** is used to minimize the **intra-cluster** distance for various values of K
  - The best clustering is selected among the best candidates for each K based on the **silhouette score**.

Ensemble Learning – Method

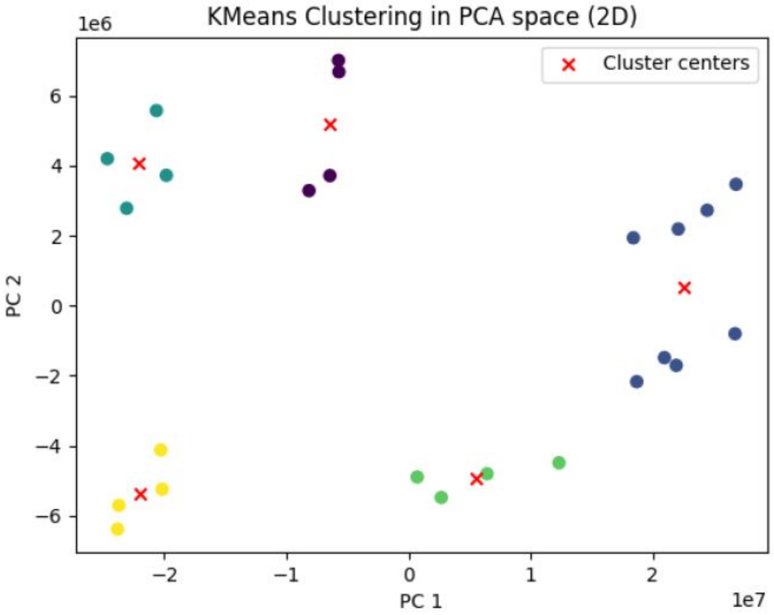
- The proposed method involves creating an **ensemble** learning **model** based on a new **clustering scheme**, which originates from an existing work (**LADD**).

- The different **styles** of training data are **grouped** into clusters, obtained as follows:
  - **K-means** is used to minimize the **intra-cluster** distance for various values of K
  - The best clustering is selected among the best candidates for each K based on the **silhouette score**.

- Individual **models** are then trained **for each** of the identified **clusters** using the FDA technique. These models are trained by applying the styles of their respective clusters to the training data.

Ensemble Learning – Method

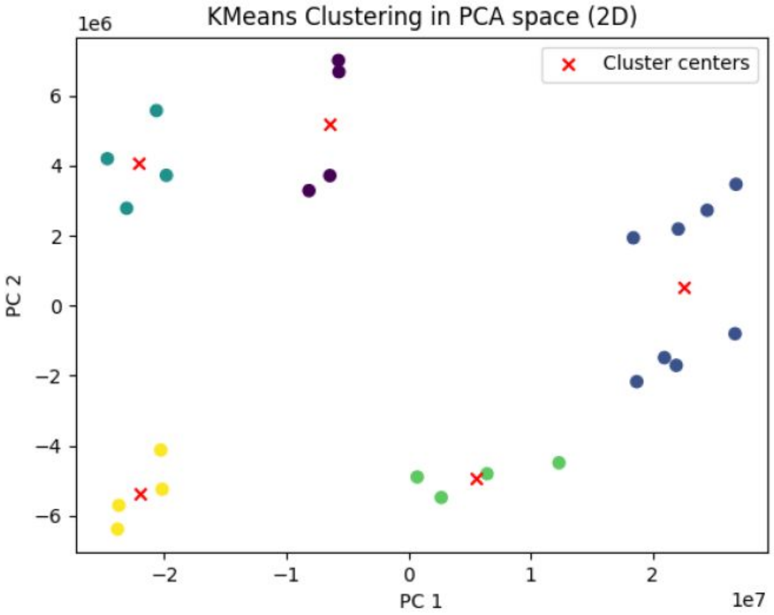
- At **inference time**, we adopt **fuzzy logic**. Instead of assigning each test image to a single cluster, we calculate **similarities** between the style of the test image and the style of each cluster.



Ensemble Learning – Method

- At **inference time**, we adopt **fuzzy logic**. Instead of assigning each test image to a single cluster, we calculate **similarities** between the style of the test image and the style of each cluster.

- These similarity scores are used to **determine the weight** with which the outputs of the models are **aggregated**.



Ensemble Learning – Method

- We consider two weighting schemes:

Ensemble Learning – Method

- We consider two weighting schemes:
  - ***Standard***: we simply take the normalized similarities as they are.

Ensemble Learning - Method

- We consider two weighting schemes:
  - ***Standard***: we simply take the normalized similarities as they are.
  - ***Skewed***: we cube the similarity scores and re-normalize them. In this way we further penalize the outputs from the clusters that are farther from the test image style.

Ensemble Learning - Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

Ensemble Learning – Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

- We consider several types of aggregation:

Ensemble Learning – Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

- We consider several types of aggregation:
  - ***Max***

Ensemble Learning – Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

- We consider several types of aggregation:
  - ***Max***
  - ***Mean***

Ensemble Learning – Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

- We consider several types of aggregation:
    - ***Max***
    - ***Mean***
    - ***Median***

Ensemble Learning - Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

- We consider several types of aggregation:
  - ***Max***
  - ***Mean***
  - ***Median***
  - ***Majority***

Ensemble Learning – Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

- We consider several types of aggregation:
  - ***Max***
  - ***Mean***
  - ***Median***
  - ***Majority***
  - ***Random by Output***

Ensemble Learning - Method

- The final output will be a weighted aggregation of the outputs from each of the K trained models.

- We consider several types of aggregation:
    - ***Max***
    - ***Mean***
    - ***Median***
    - ***Majority***
    - ***Random by Output***
    - ***Random by Pixel***

Ensemble Learning – Results

- ***Majority*** is the best performing on **Test Same Dom.**

Aggregation	Weighting Scheme	Test Same Dom	Test Diff Dom
Max	n.a	0.2524	0.2008
Mean	Standard	0.2688	0.2094
	Skewed	0.2595	0.2056
Median	Standard	0.2675	0.1936
	Skewed	0.2614	0.1674
Majority	n.a	0.2713	0.2059
Random by Output	Standard	0.2609	0.1880
	Skewed	0.2547	0.1943
Random by Pixel	Standard	0.2552	0.1887
	Skewed	0.2531	0.1949

Ensemble Learning – Results

- ***Majority*** is the best performing on **Test Same Dom.**
- **Mean** is the most effective on **Test Diff Dom.**

Aggregation	Weighting Scheme	Test Same Dom	Test Diff Dom
Max	n.a	0.2524	0.2008
Mean	Standard Skewed	0.2688 0.2595	0.2094 0.2056
Median	Standard Skewed	0.2675 0.2614	0.1936 0.1674
Majority	n.a	0.2713	0.2059
Random by Output	Standard Skewed	0.2609 0.2547	0.1880 0.1943
Random by Pixel	Standard Skewed	0.2552 0.2531	0.1887 0.1949

Ensemble Learning – Results

- ***Majority*** is the best performing on **Test Same Dom.**
- **Mean** is the most effective on **Test Diff Dom.**
- **Mean**, **Majority** and **Median** all surpass Max on both test datasets.

Aggregation	Weighting Scheme	Test Same Dom	Test Diff Dom
Max	n.a	0.2524	0.2008
Mean	Standard	0.2688	0.2094
	Skewed	0.2595	0.2056
Median	Standard	0.2675	0.1936
	Skewed	0.2614	0.1674
Majority	n.a	0.2713	0.2059
Random by Output	Standard	0.2609	0.1880
	Skewed	0.2547	0.1943
Random by Pixel	Standard	0.2552	0.1887
	Skewed	0.2531	0.1949

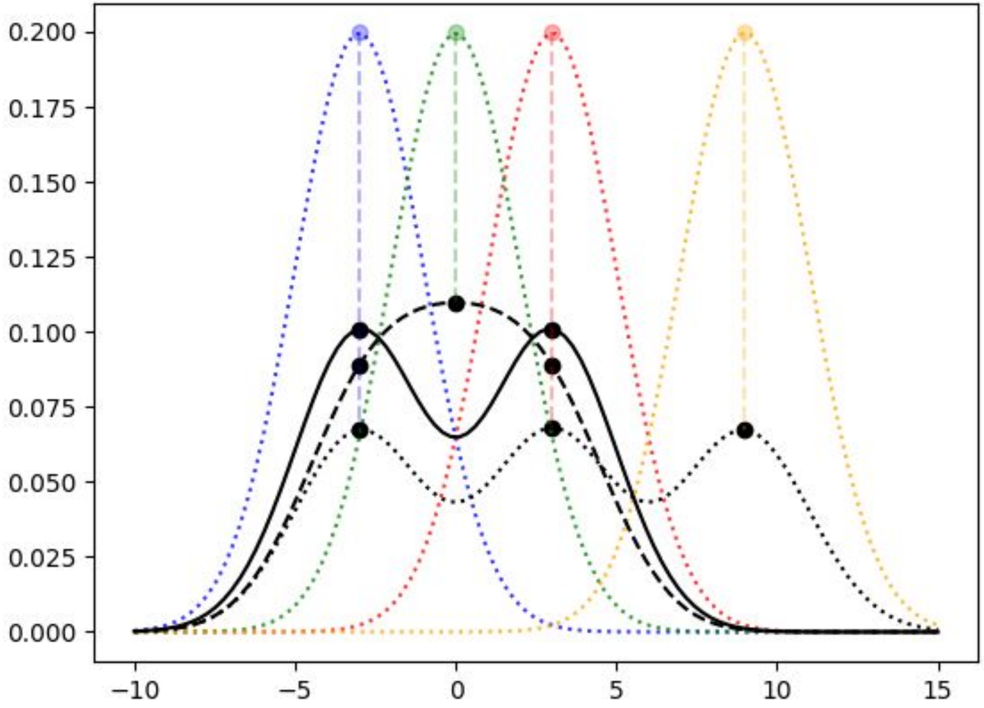
Ensemble Learning – Results

- ***Majority*** is the best performing on **Test Same Dom.**
- **Mean** is the most effective on **Test Diff Dom.**
- **Mean**, **Majority** and **Median** all surpass Max on both test datasets.
- The **skewed** weighting scheme typically **underperforms**, highlighting how the most successful aggregations appear to be those favoring balanced predictions .

Aggregation	Weighting Scheme	Test Same Dom	Test Diff Dom
Max	n.a	0.2524	0.2008
Mean	Standard	0.2688	0.2094
	Skewed	0.2595	0.2056
Median	Standard	0.2675	0.1936
	Skewed	0.2614	0.1674
Majority	n.a	0.2713	0.2059
Random by Output	Standard	0.2609	0.1880
	Skewed	0.2547	0.1943
Random by Pixel	Standard	0.2552	0.1887
	Skewed	0.2531	0.1949

Extensions 2
Simulating unseen styles

- Motivation
- Intuition
- Methodology
- Results



Simulating Unseen Styles - Motivation

- Throughout our work, we have seen that **performances** constantly **drop** when considering *Test Diff Dom* instead of *Test Same Dom*.

Centralised Baseline

Test Same Dom mIoU	Test Diff Dom mIoU
0.5087 ± 0.0018	0.3211 ± 0.0031

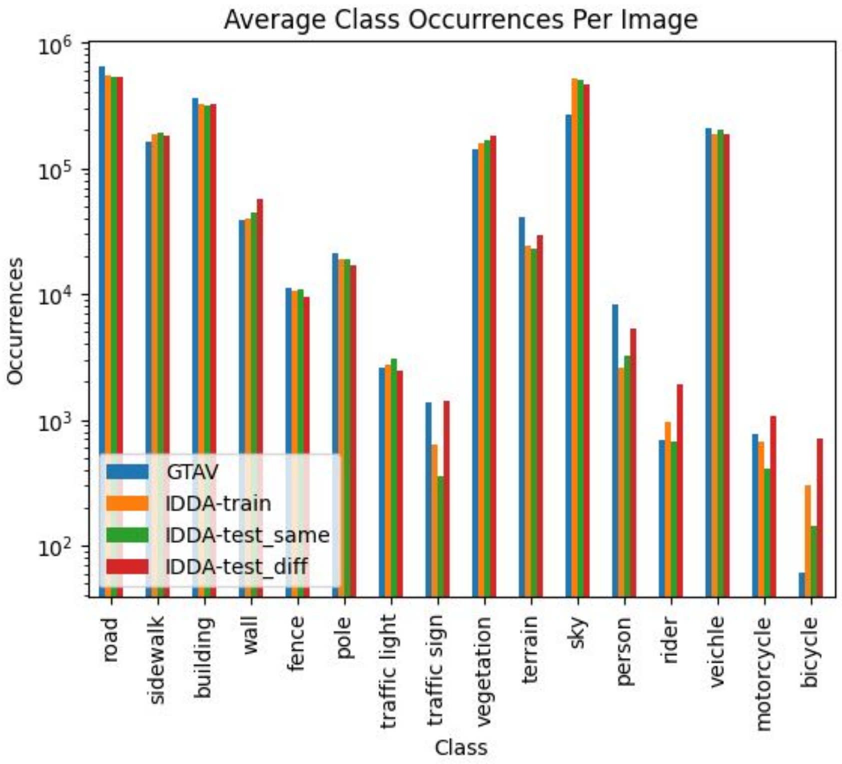
FL

Test Same Dom mIoU	Test Diff Dom mIoU
0.3962 ± 0.0093	0.2667 ± 0.0178
0.4485 ± 0.0038	0.2946 ± 0.0045

Simulating Unseen Styles - Motivation

- Throughout our work, we have seen that **performances** constantly **drop** when considering *Test Diff Dom* instead of *Test Same Dom*.

- As already said, this is caused by a **gap** between the **latent distributions** of the two.



Simulating Unseen Styles - Motivation

- Throughout our work, we have seen that **performances** constantly **drop** when considering *Test Diff Dom* instead of *Test Same Dom*.

- As already said, this is caused by a **gap** between the **latent distributions** of the two.

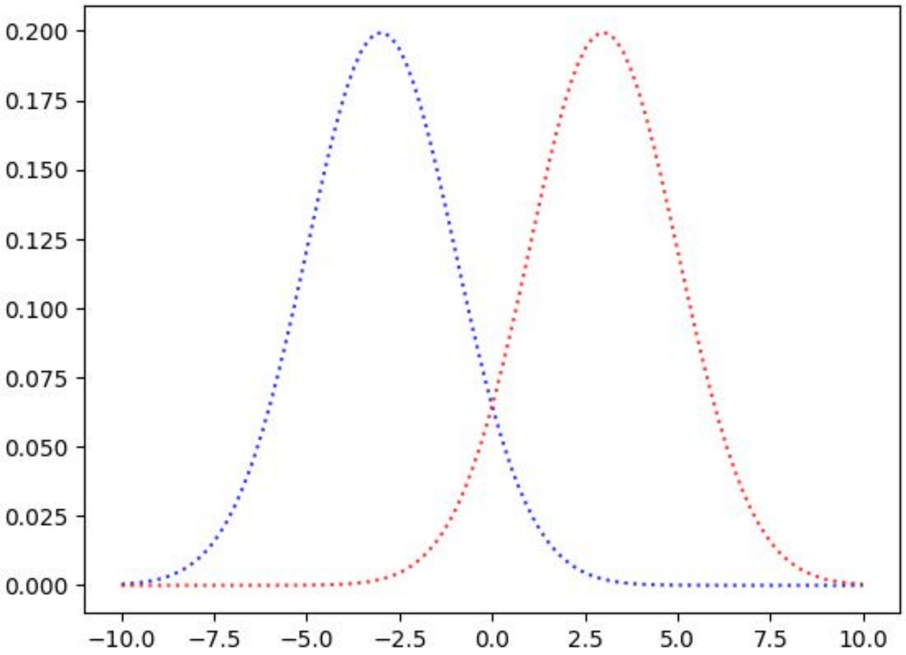
- This is still **true** when we **transfer** the **style** of IDDA onto GTAV using FDA.

Pre-train with FDA

Test Same Dom mIoU	Test Diff Dom mIoU
0.2633 ± 0.0036	0.2050 ± 0.0030
0.2772 ± 0.0043	0.1954 ± 0.0028

Simulating Unseen Styles - Intuition

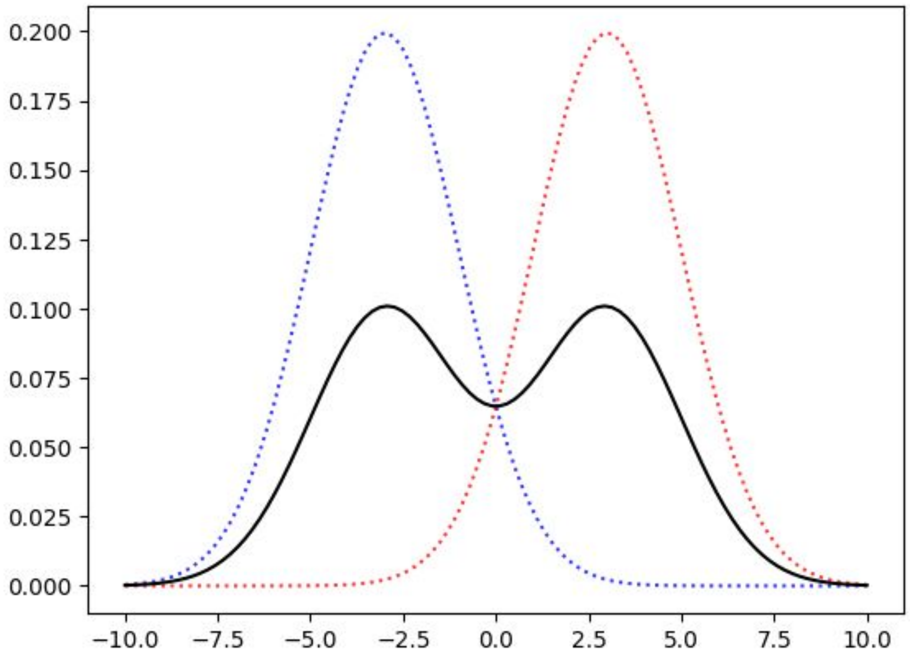
- The intuition behind our extension comes from thinking of **each client** generating the **styles** according to its own **distribution.**



Simulating Unseen Styles - Intuition

- The intuition behind our extension comes from thinking of **each client** generating the **styles** according to its own **distribution.**

- Thus, the **overall distribution** will be a **mixture** of the distributions from all the possible clients.

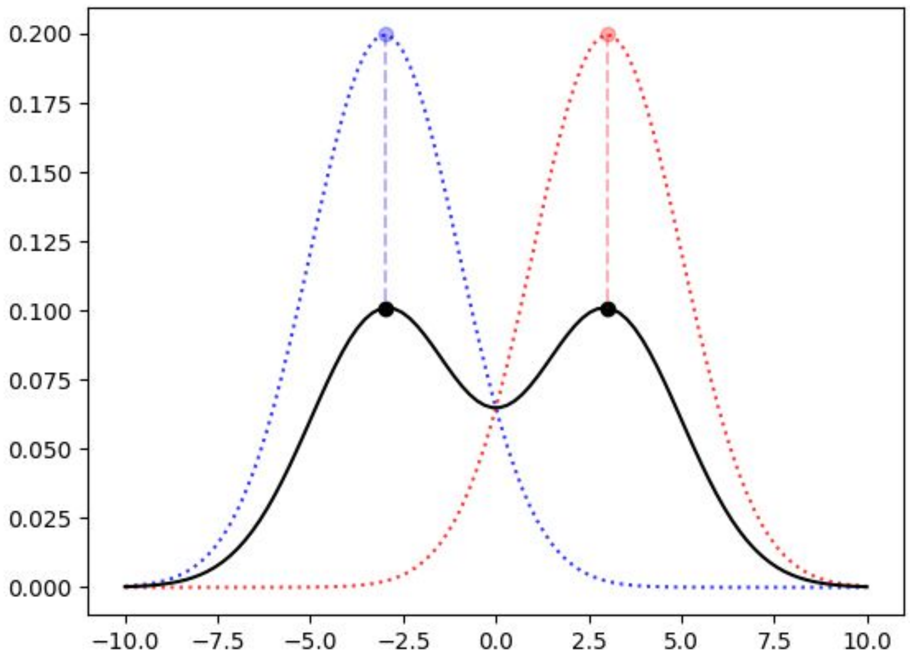


Simulating Unseen Styles - Intuition

- The intuition behind our extension comes from thinking of **each client** generating the **styles** according to its own **distribution.**

- Thus, the **overall distribution** will be a **mixture** of the distributions from all the possible clients.

- In such a setting, the previous application of FDA would consist in applying the styles identified as the mixture **components' averages.**



Simulating Unseen Styles - Intuition

- However, we are **only transferring** the **styles** that we can observe on the **train** clients, that we recall come from the same distribution *Test Same Dom*.

- This could be one of the **cause** in the **drop** in performances.

Simulating Unseen Styles - Intuition

- However, we are **only transferring** the **styles** that we can observe on the **train** clients, that we recall come from the same distribution *Test Same Dom*.

- This could be one of the **cause** in the **drop** in performances.

- We want to try to **simulate** the **unseen styles**, which will have to respect two properties:

    a. **similar enough** to the real styles, so as to come from the same mixture.
    b. **different enough** to foster better generalization.

Simulating Unseen Styles - Methodology

- We generate a new style as an **interpolation** of two observed styles, controlled by a parameter *w*.

- This should satisfy property **(a).**

$$\hat{S} = wS_1 + (1-w)S_2, 0 < w < 1$$

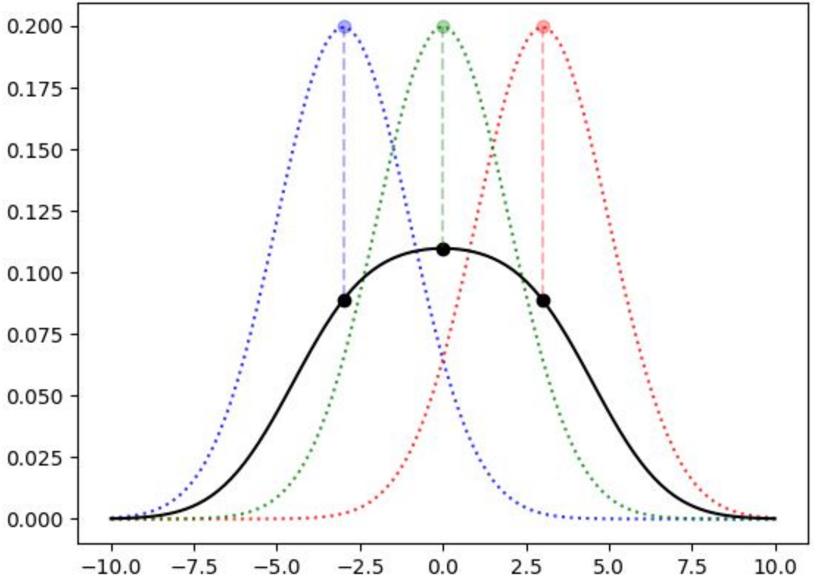
Simulating Unseen Styles - Methodology

- We generate a new style as an **interpolation** of two observed styles, controlled by a parameter *w*.

- This should satisfy property **(a).**

- However, the generated styles would be **too similar** to the observed ones, possibly making the distribution even **more centered** around them.

$$\hat{S} = wS_1 + (1 - w)S_2, 0 < w < 1$$



Simulating Unseen Styles - Methodology

- We generate a new style as an **interpolation** of two observed styles, controlled by a parameter *w*.

- This should satisfy property **(a).**

- However, the generated styles would be **too similar** to the observed ones, possibly making the distribution even **more centered** around then.

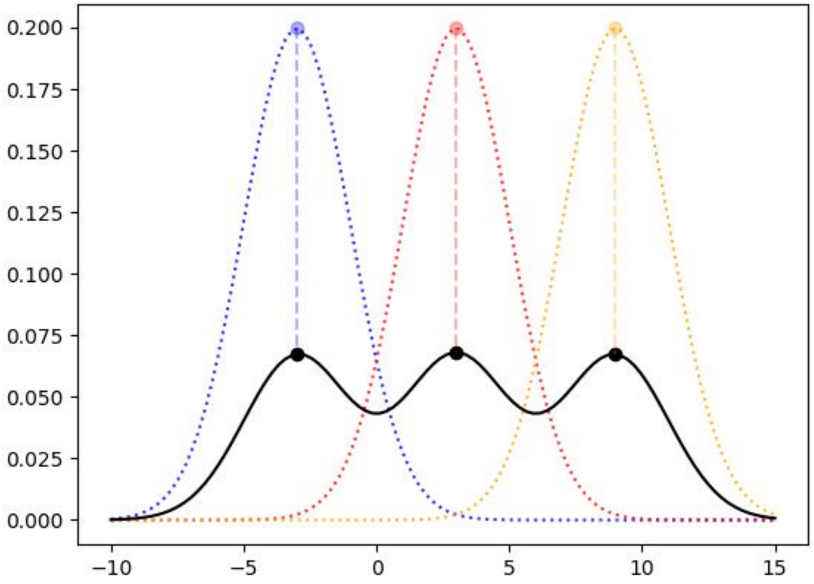
- Thus, we further imagine that the **“in-between” style** is one of the **observed** ones.

$$\hat{S} = \frac{1}{w} S_2 - \frac{1-w}{w} S_1$$

Simulating Unseen Styles - Methodology

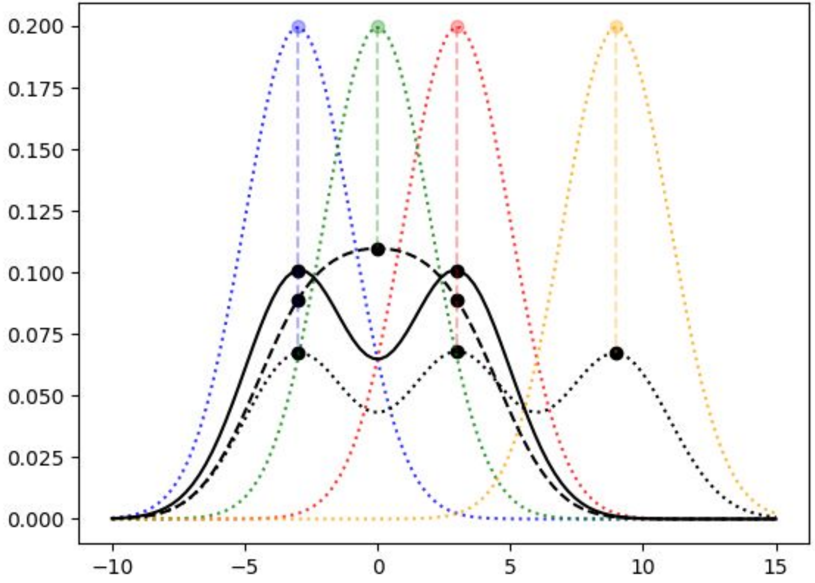
- We generate a new style as an **interpolation** of two observed styles, controlled by a parameter *w*.
- This would satisfy property **(a).**
- However, the generated styles would be **too similar** to the observed ones, possibly making the distribution even **more centered** around then.
- Thus, we further imagine that the **"in-between"** style is one of the **observed** ones.
- This allows generating styles that are **diverse** enough, satisfying property **(b).**

$$\hat{S} = \frac{1}{w} S_2 - \frac{1-w}{w} S_1$$



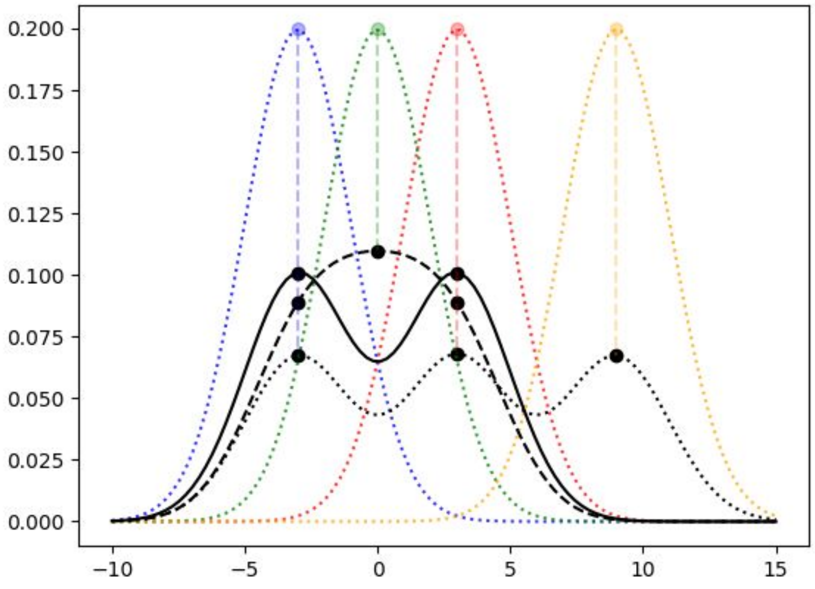
Simulating Unseen Styles - Methodology

Comparison of the **generated styles.**

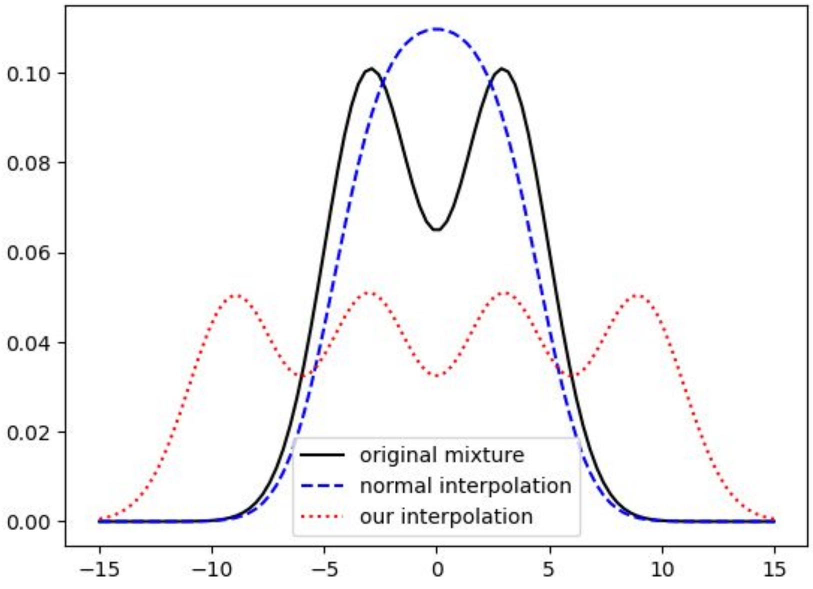


Simulating Unseen Styles - Methodology

Comparison of the **generated styles.**



Comparison of the **resulting mixtures.**



Simulating Unseen Styles – Methodology

- Given a bank of styles of cardinality *n*, the possible “in-between” interpolated styes are:

$$\binom{n}{2}$$

- This is doubled in our methodology, since either observed style can be chosen as the “in-between” one, resulting in a number of possible styles equal to:

$$2\binom{n}{2} = n^2 - n$$

Simulating Unseen Styles – Methodology

- Given a bank of styles of cardinality *n*, the possible “in-between” interpolated styes are:

$$\binom{n}{2}$$

- This is doubled in our methodology, since either observed style can be chosen as the “in-between” one, resulting in a number of possible styles equal to:

$$2\binom{n}{2} = n^2 - n$$

- Given that we cannot train for too long, we wanted to keep the number of possibly generated styles low, so we made 2 adjustments.

- Since this number grow quadratically with *n*, instead of considering the average style for each client we only consider the average style for each cluster in our bank, leveraging the clustering obtained at the previous point.

- Moreover, to further lean toward the actually observed model, we only apply a generated one with probability *p*.

Simulating Unseen Styles - Results

Type	p	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
default	n.a.	0.4315 ± 0.0015	0.2352 ± 0.0066	0.2421 ± 0.0068	0.1746 ± 0.0079
interpolation	0.2	0.4299 ± 0.0006	0.2620 ± 0.0031	0.2689 ± 0.0030	0.1909 ± 0.0032
interpolation	0.5	0.4230 ± 0.0018	0.2604 ± 0.0061	0.2639 ± 0.0059	0.1907 ± 0.0068
noise	n.a.	0.4307 ± 0.0016	0.2605 ± 0.0049	0.2648 ± 0.0049	0.1812 ± 0.0057

- Our methodology achieves some **promising results.** We considered values of *p=0.2* and *p=0.5.*

Simulating Unseen Styles - Results

Type	p	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
default	n.a.	0.4315 ± 0.0015	0.2352 ± 0.0066	0.2421 ± 0.0068	0.1746 ± 0.0079
interpolation	0.2	0.4299 ± 0.0006	0.2620 ± 0.0031	0.2689 ± 0.0030	0.1909 ± 0.0032
interpolation	0.5	0.4230 ± 0.0018	0.2604 ± 0.0061	0.2639 ± 0.0059	0.1907 ± 0.0068
noise	n.a.	0.4307 ± 0.0016	0.2605 ± 0.0049	0.2648 ± 0.0049	0.1812 ± 0.0057

- Our methodology achieves some **promising results.** We considered values of *p=0.2* and *p=0.5*.
- We see an expected drop of performance on *GTAV*, but an **increase** on *Train, Test Same Dom,* and *Test Diff Dom.* The smaller value of *p* gives a marginally larger improvement.

Simulating Unseen Styles - Results

Type	p	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
default	n.a.	0.4315 ± 0.0015	0.2352 ± 0.0066	0.2421 ± 0.0068	0.1746 ± 0.0079
interpolation	0.2	0.4299 ± 0.0006	0.2620 ± 0.0031	0.2689 ± 0.0030	0.1909 ± 0.0032
interpolation	0.5	0.4230 ± 0.0018	0.2604 ± 0.0061	0.2639 ± 0.0059	0.1907 ± 0.0068
noise	n.a.	0.4307 ± 0.0016	0.2605 ± 0.0049	0.2648 ± 0.0049	0.1812 ± 0.0057

- Our methodology achieves some **promising results.** We considered values of *p=0.2* and *p=0.5.*
- We see an expected drop of performance on *GTAV*, but an **increase** on *Train, Test Same Dom,* and *Test Diff Dom.* The smaller value of *p* gives a marginally larger improvement.
- To discern whether the improvements simply come from an increased style ambiguity, we also introduce a **control experiment** in which we **add Gaussian noise** to the style while training.

Simulating Unseen Styles - Results

Type	p	GTAV mIoU	IDDA Train mIoU	Test Same Dom mIoU	Test Diff Dom mIoU
default	n.a.	0.4315 ± 0.0015	0.2352 ± 0.0066	0.2421 ± 0.0068	0.1746 ± 0.0079
interpolation	0.2	0.4299 ± 0.0006	0.2620 ± 0.0031	0.2689 ± 0.0030	0.1909 ± 0.0032
interpolation	0.5	0.4230 ± 0.0018	0.2604 ± 0.0061	0.2639 ± 0.0059	0.1907 ± 0.0068
noise	n.a.	0.4307 ± 0.0016	0.2605 ± 0.0049	0.2648 ± 0.0049	0.1812 ± 0.0057

- Our methodology achieves some **promising results.** We considered values of *p=0.2* and *p=0.5.*
- We see an expected drop of performance on *GTAV*, but an **increase** on *Train, Test Same Dom,* and *Test Diff Dom.* The smaller value of *p* gives a marginally larger improvement.
- To discern whether the improvements simply come from an increased style ambiguity, we also introduce a **control experiment** in which we **add Gaussian noise** to the style while training.
- Adding noise brings **comparable benefits** on *Test Same Dom*, but they **don't** seem to **hold up** to the ones achieved on *Test Diff Dom*.

Thanks for the attention

Questions?

- Luca Agnese: l.agnese@studenti.polito.it
- Fabio Rizzi: s308770@studenti.polito.it
- Flavio Spuri: s303657@studenti.polito.it