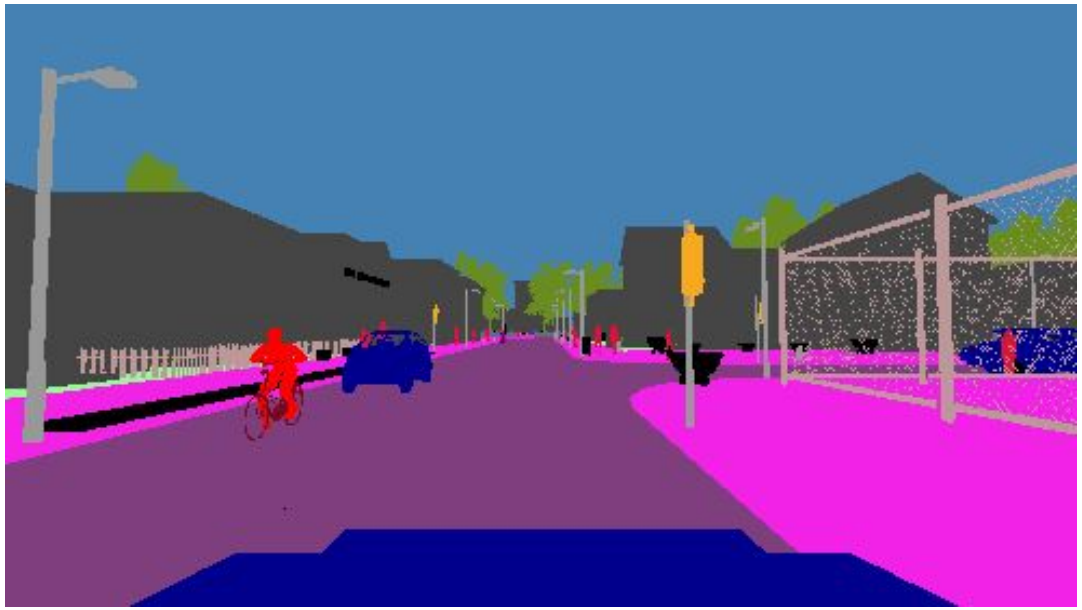
POLITECNICO
DI TORINO

Semantic Segmentation and Federated Learning: clustering and crucial classes boosting

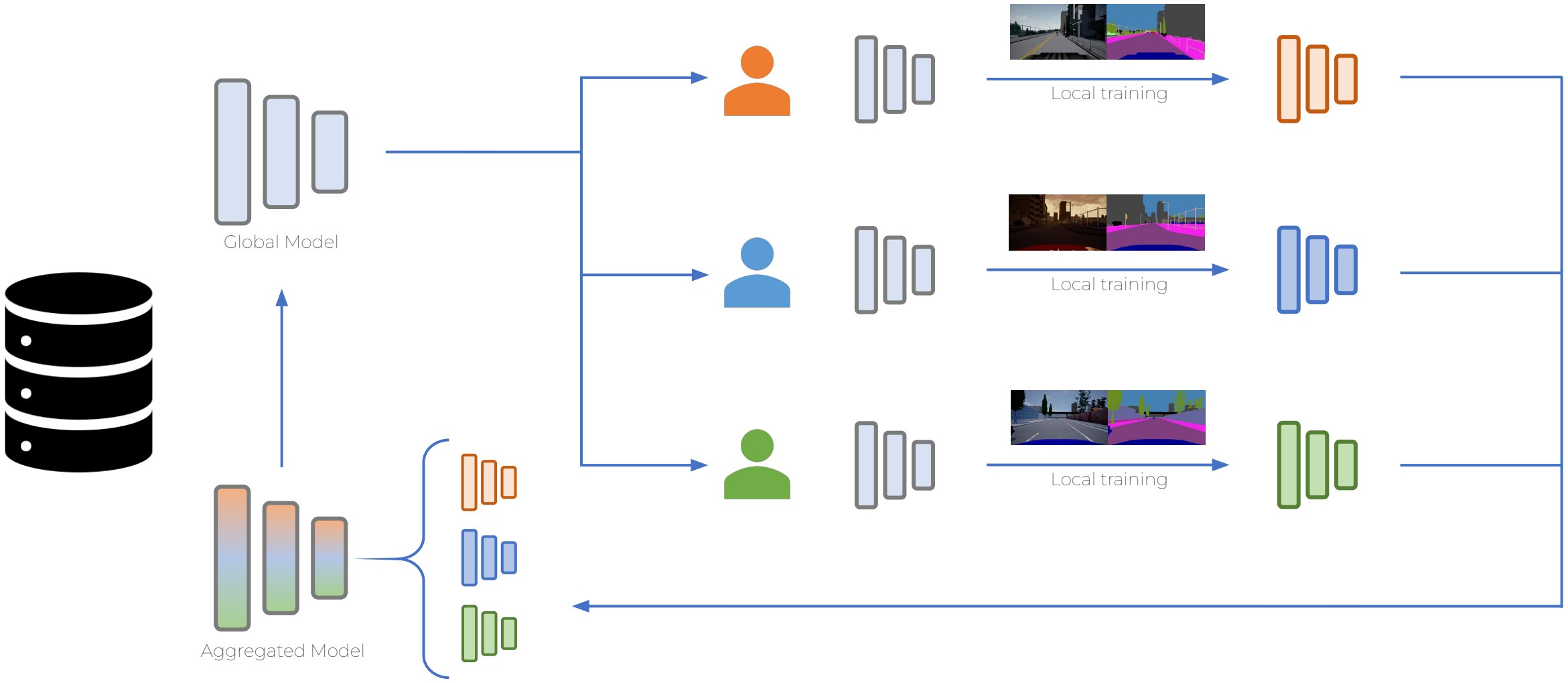
Margherita Soro, Nicola Candellieri, Lorenzo Sibille

Semantic Segmention



Assign class label to each pixel

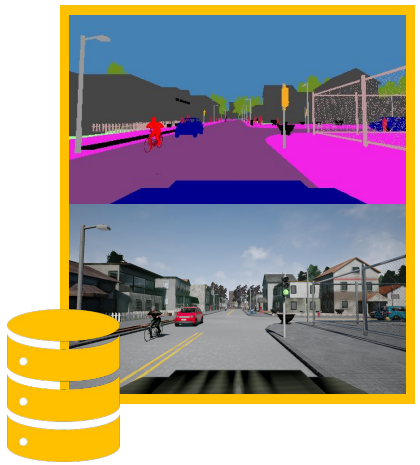
Federated Learning



Challenges

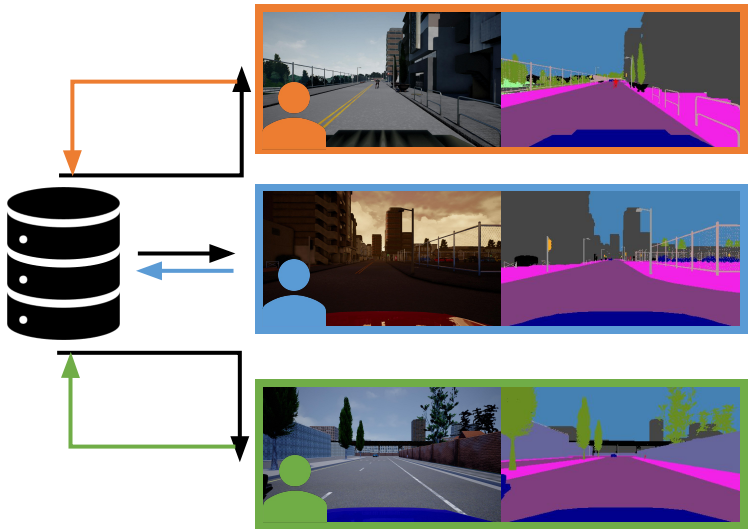
Semantic Segmentation	High quality labeled dataset needed
	Domain gaps across datasets
Federated Learning	Systems heterogeneity
	Statistical heterogeneity
	Privacy concerns
	Communication costs

Settings



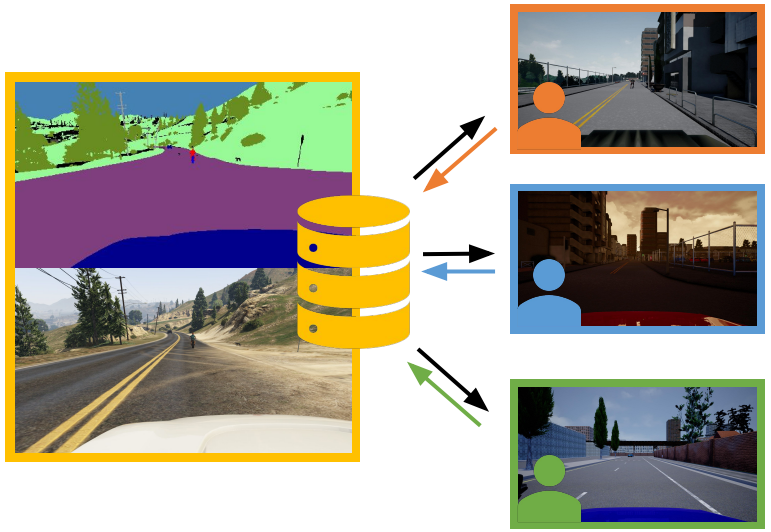
Centralized

Supervised



Federated

Supervised



Federated

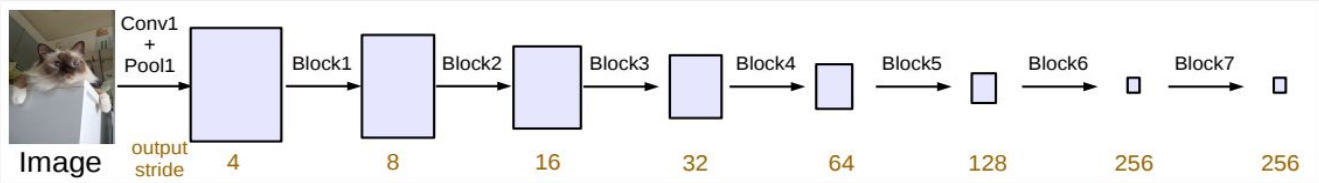
Unsupervised

Model – DeepLabV3

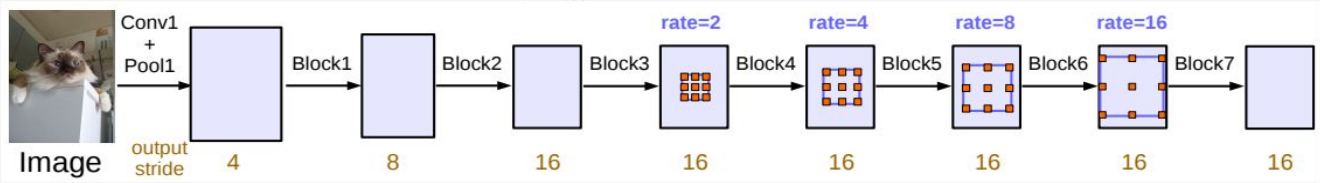
Atrous convolution captures relations without increasing kernel size or decreasing spatial resolution

Pyramid pooling handles scale variations and different sizes

MobileNetV2 light and efficient CNN as backbone



(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

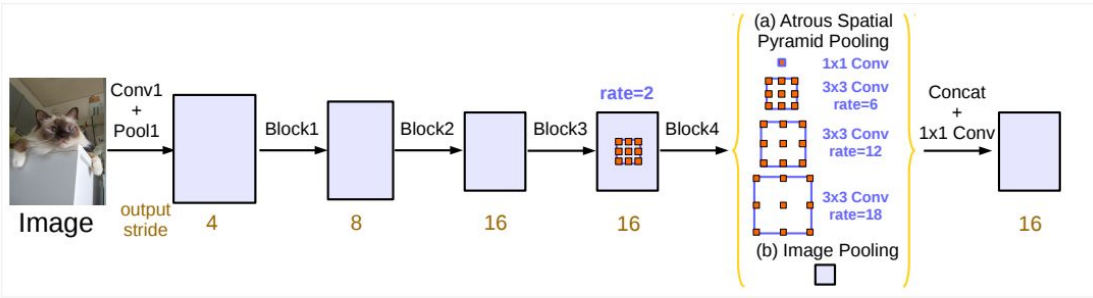


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

Datasets

GTA5

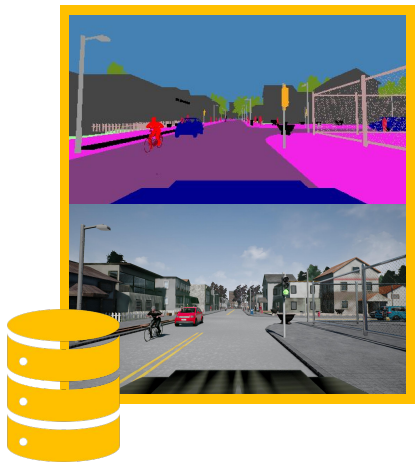
IDDA

Split into: TR (training)

TS1 (test, same dom.)

TS2 (test, diff. dom.)

Settings



Centralized

Supervised

Federated

Supervised

Federated

Unsupervised

Centralized - results

Hyperparameters and data augmentation testing

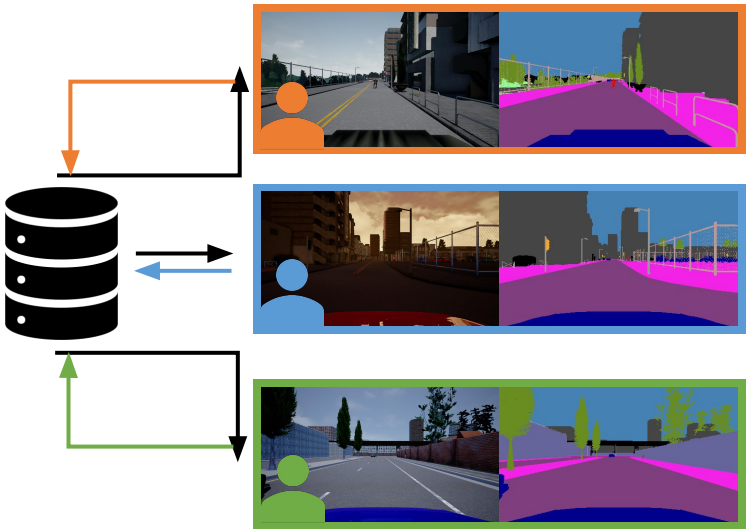
RandomCrop chosen over Rotation due to efficiency

	Transforms					
	-	Random Crop	Normalize (Imagenet)	Random Rotation	Random HorizontalFlip	Color Jitter
mIoU [*TR*]	**0.6849**	0.6480	0.6763	0.6724	0.6806	0.6834
mIoU [*TS1*]	0.5793	0.6010	0.5773	**0.6037**	0.5911	0.5754
mIoU [*TS2*]	0.4817	**0.4961**	0.4771	0.4954	0.4880	0.4786

Settings

Centralized

Supervised



Federated

Supervised

Federated

Unsupervised

Supervised FL - results

Small gap wrt centralized.

Best performances with:

few local epochs

⊕ Does not stress small devices

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

Supervised FL - results

Small gap wrt centralized.

Best performances with:

- ⊕ few local epochs
- ⊗ few clients per round

↓

Reduces parallelism

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

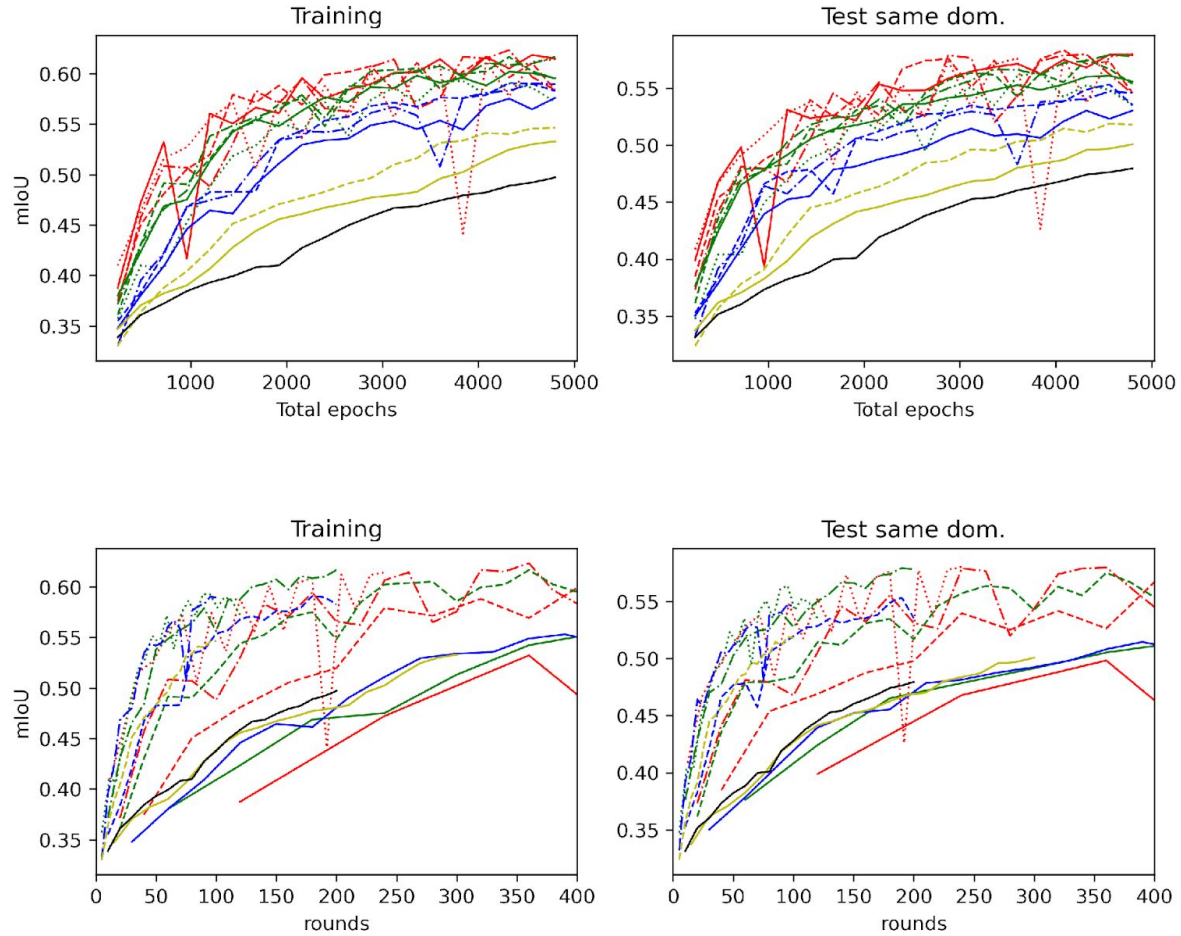
Supervised FL – clients results

Few clients:

- ⊕ better without parallelism
- ⊗ more oscillating

More clients:

- ⊕ better with parallelism
- ⊕ more stable
- ⊗ more data on network



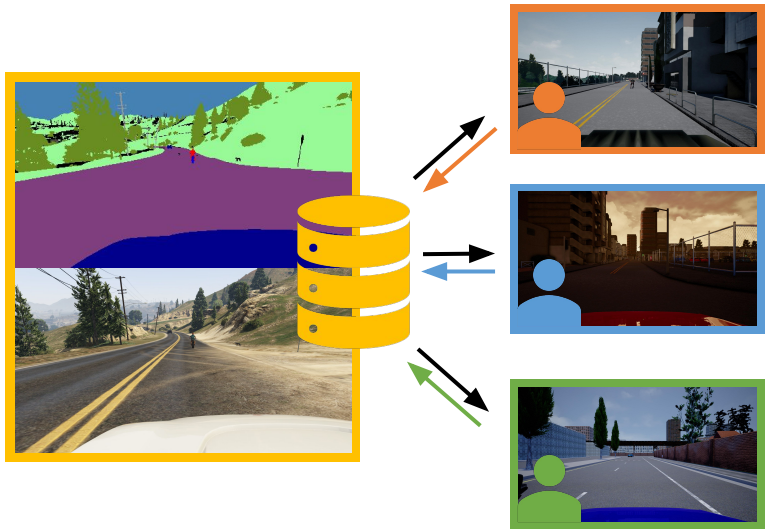
Settings

Centralized

Supervised

Federated

Supervised

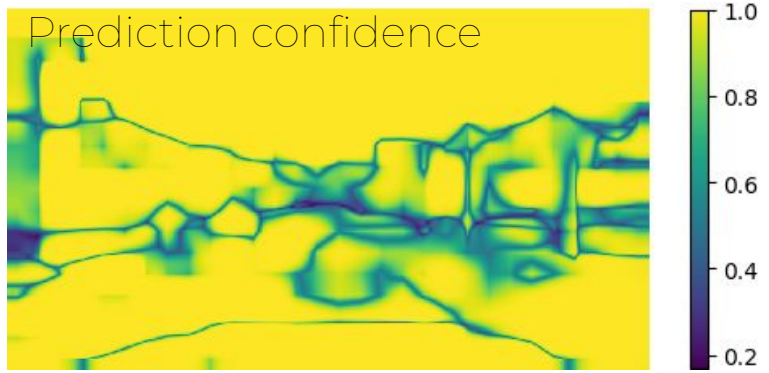


Federated

Unsupervised

Pseudo labels

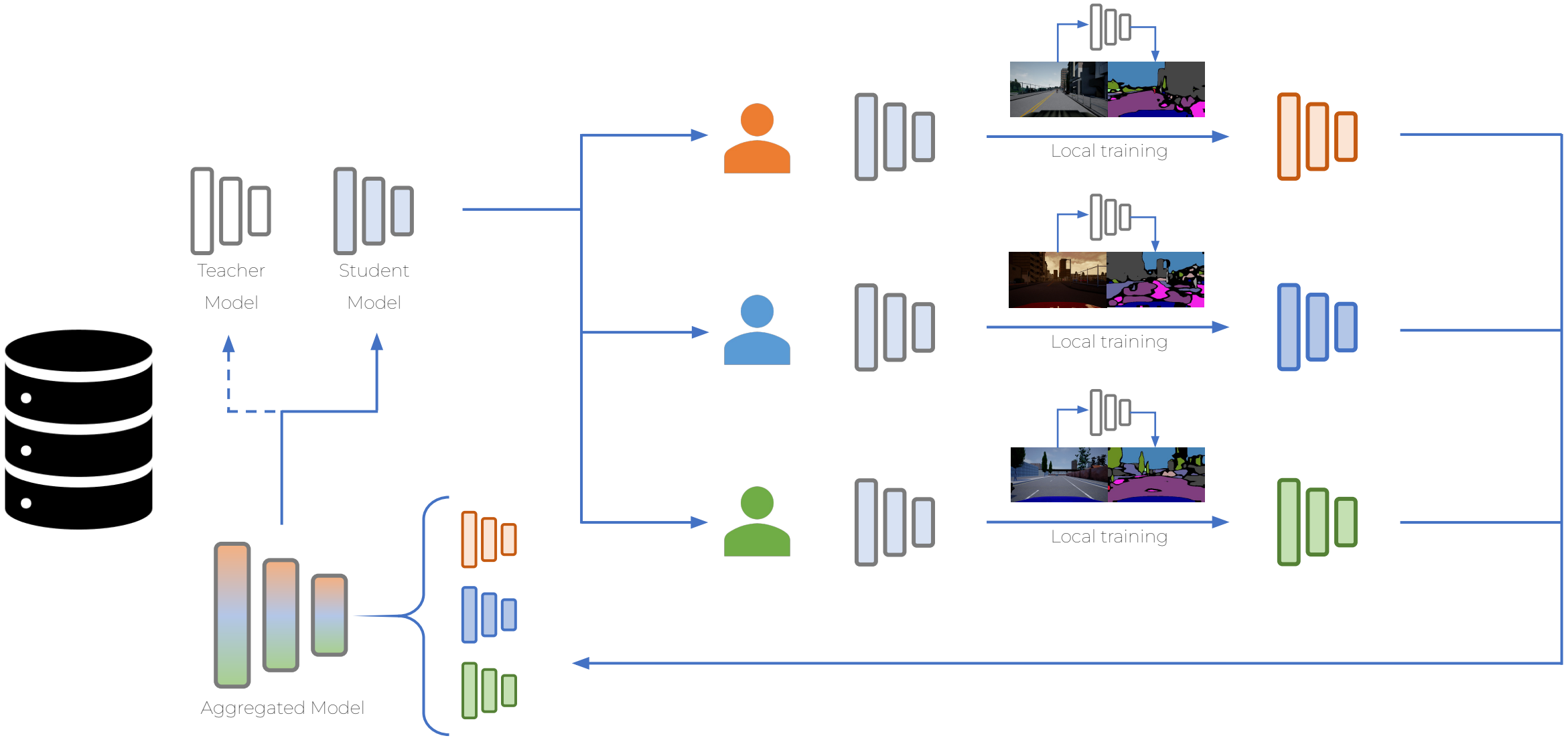
Teacher model's prediction



Pseudo-labels from masked prediction

We keep only confidence > 0.9 or 66% most confident pred. per class

Unsupervised FL

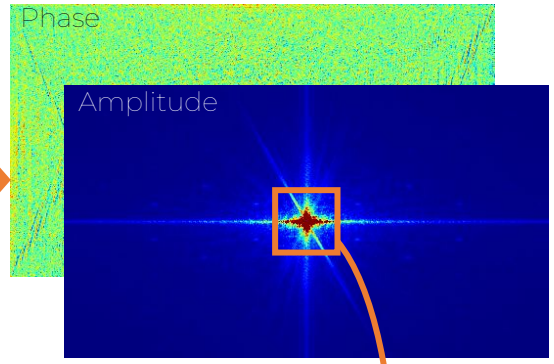


Fourier Domain Adaptation

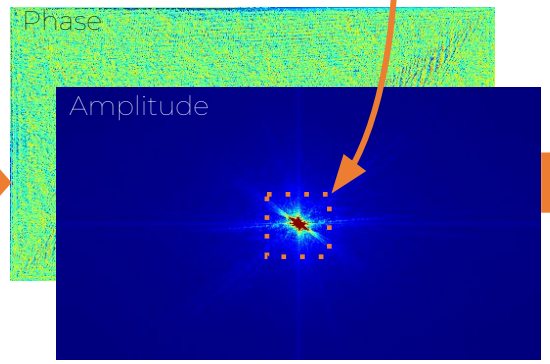
Target image

Source image

FFT



FFT



IFFT

Source image
with Target style

Pretraining on GTA

Big domain gap and performances drop

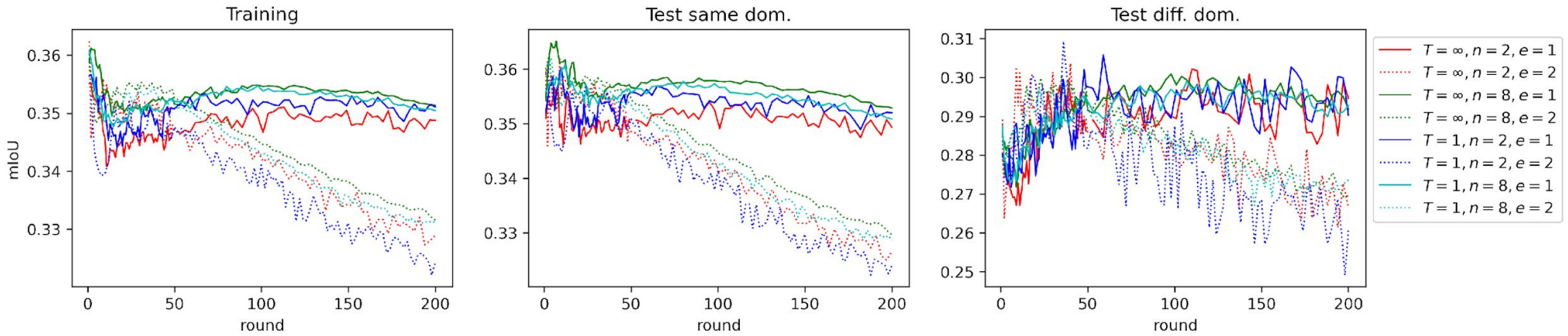
Changing «colors» worked the best, with FDA slightly worse than Jitter

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

Unsupervised FL - results

- ⊕ Improvement wrt initial model
- ⊕ Better performances on different domain
- ⊗ Immediate peak, then elbow on training
- ⊗ Terrible trend with 2 local epochs

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



Our improvements

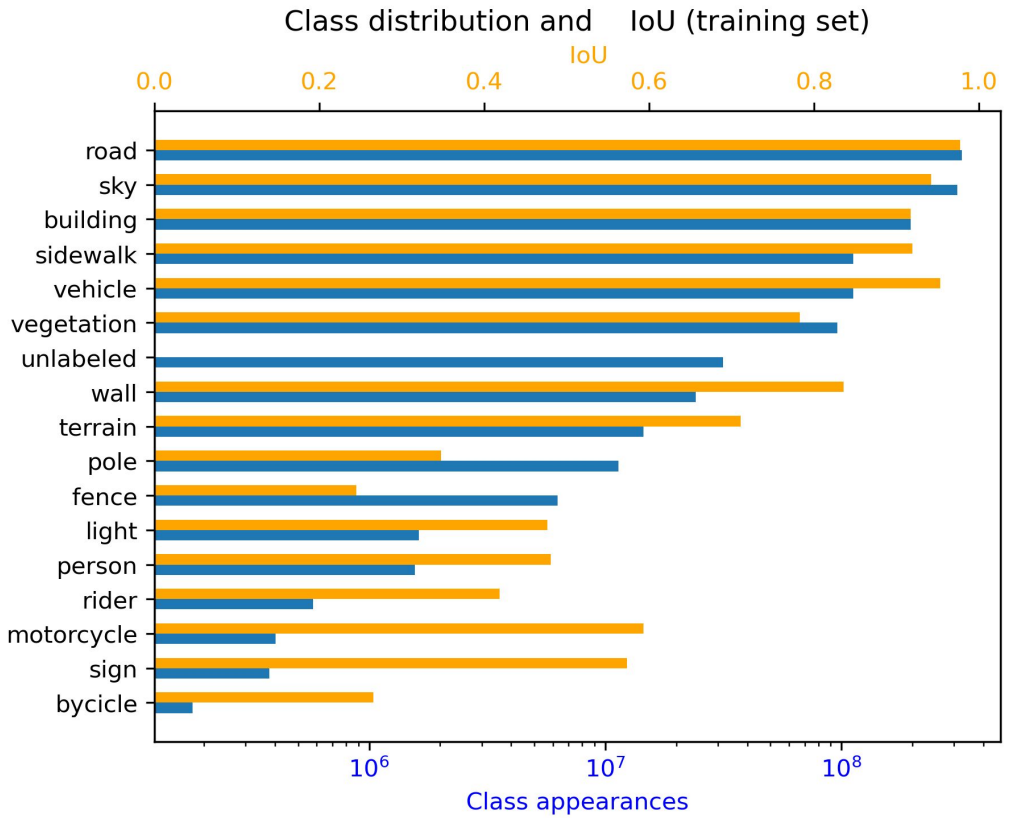
Crucial classes boosting

Different reductions for cross-entropy loss, improving performances for crucial but infrequent classes.

Clustering

Style-based clustering of clients, creating cluster-specific sub-models, enhancing performances.

Class IoU



Correlation between IoU and frequency
Crucial classes are infrequent

Reductions proposed

$$\ell_c = \sum losses\ of\ pixels\ of\ class\ c$$

$$\#c = number\ of\ pixels\ of\ class\ c$$

Mean

$$\frac{\sum \ell_c}{H \cdot W}$$

Weighted mean

$$\frac{\sum w_c \ell_c}{H \cdot W}$$

Mean by class

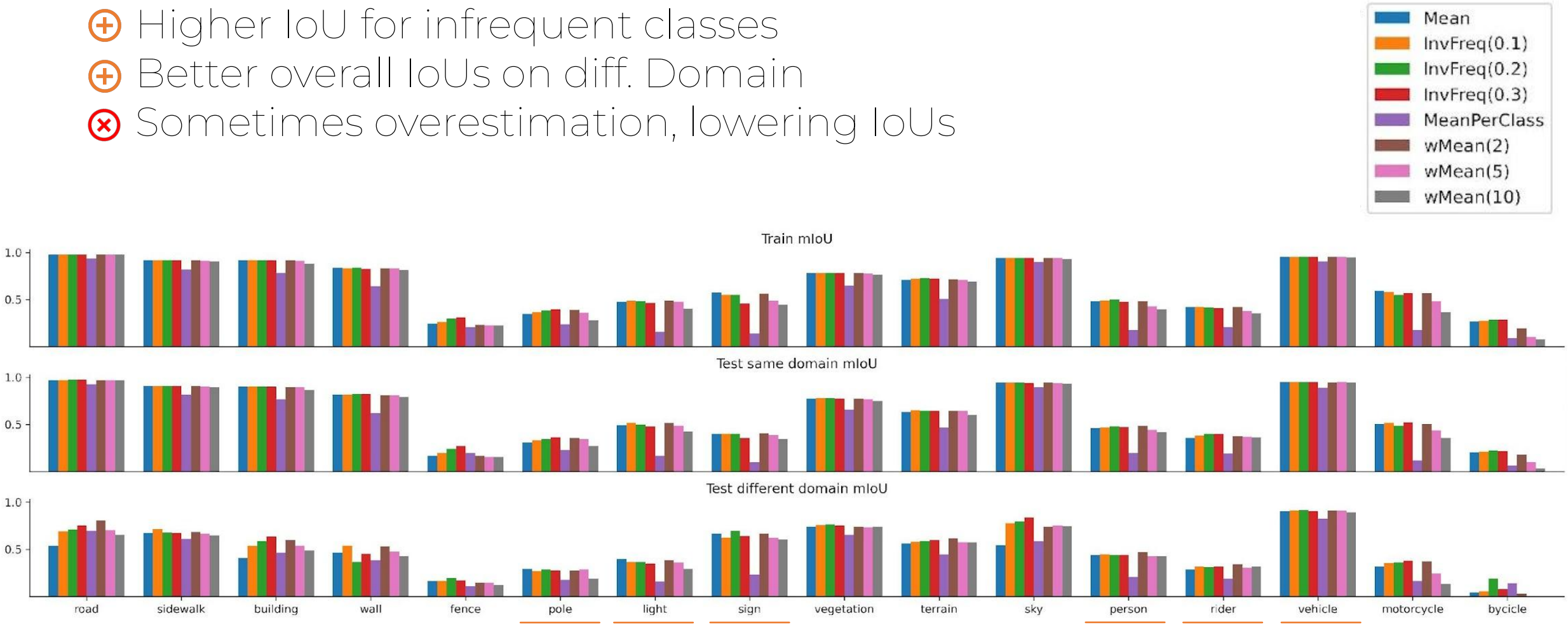
$$\frac{\sum \frac{\ell_c}{\#c}}{\#classes}$$

«Inversely» proportional to $\#c$

$$\frac{\sum \ell_c (H \cdot W / \#c)^p}{H \cdot W}$$

Quantitative results

- ⊕ Higher IoU for infrequent classes
- ⊕ Better overall IoUs on diff. Domain
- ⊗ Sometimes overestimation, lowering IoUs



Quantitative results

- Improvements in mIoU for all sets in all scenarios

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

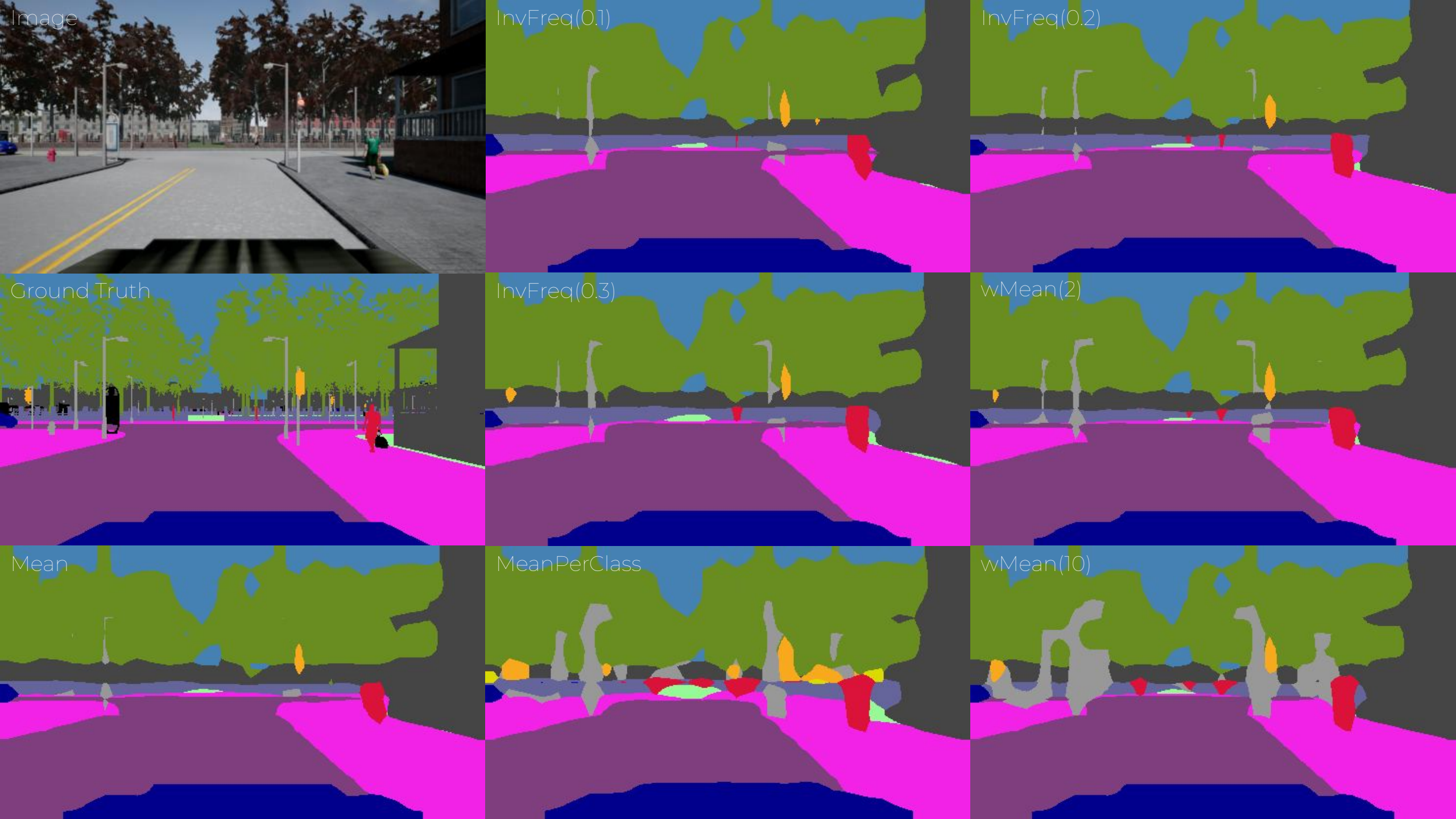


Image
InvFreq(0.1)
InvFreq(0.2)
Ground Truth
InvFreq(0.3)
wMean(2)
Mean
MeanPerClass
wMean(10)

Clustered FL

To tackle statistical heterogeneity

The process involves two steps:

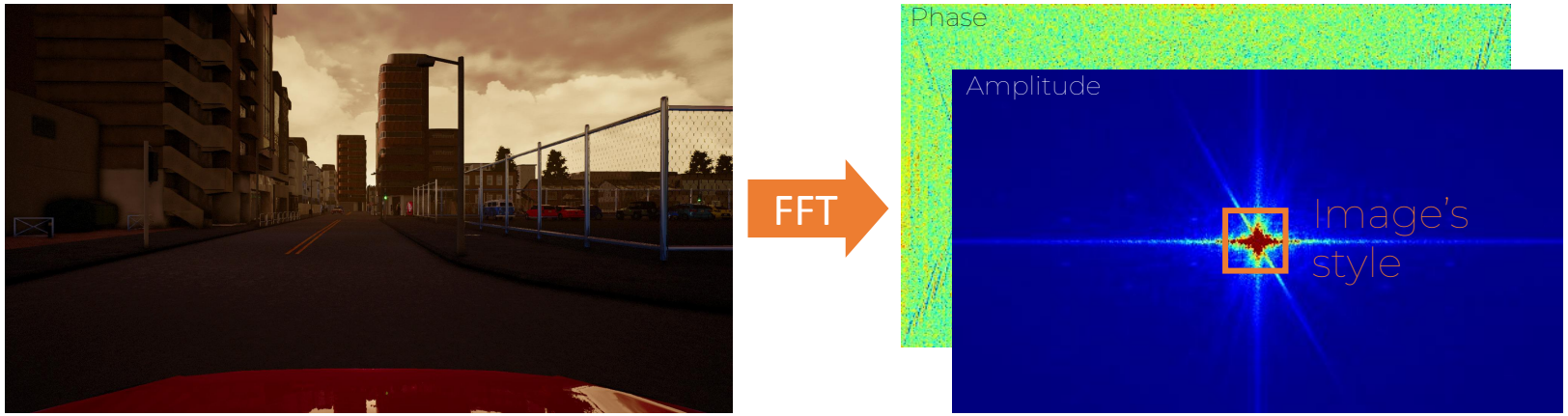
1. Clients are clustered according to their style
2. Personalized models are built for each cluster

Clustering

Style is extracted as in FDA:

Image's style corresponds to the low frequencies of FFT

Client's style is average style of its images

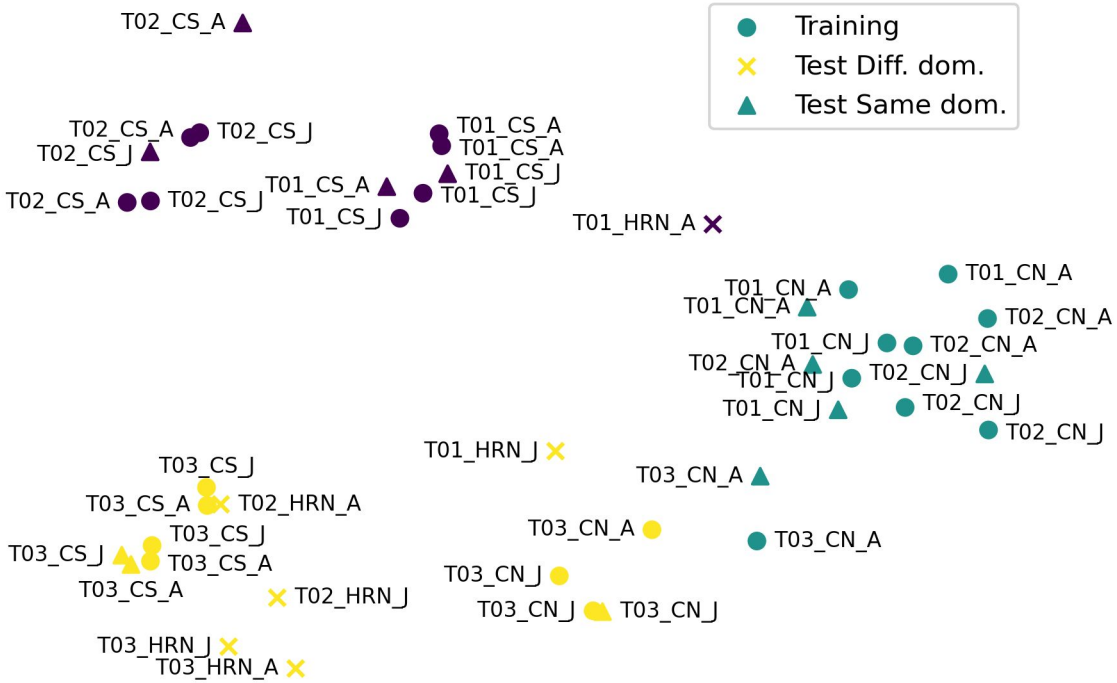


Clustering

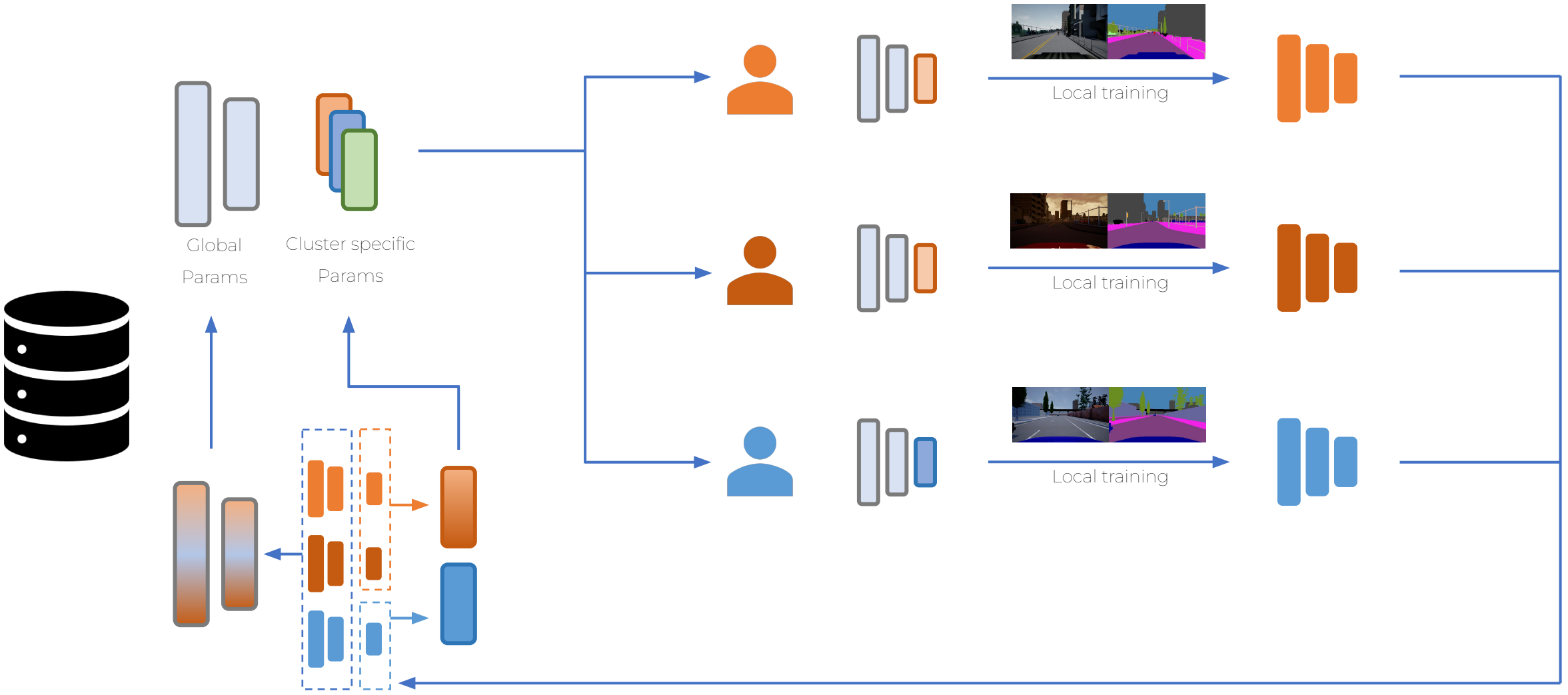
Multiple clusterings using **K-means**:

- trying multiple values of K
- using many initializations

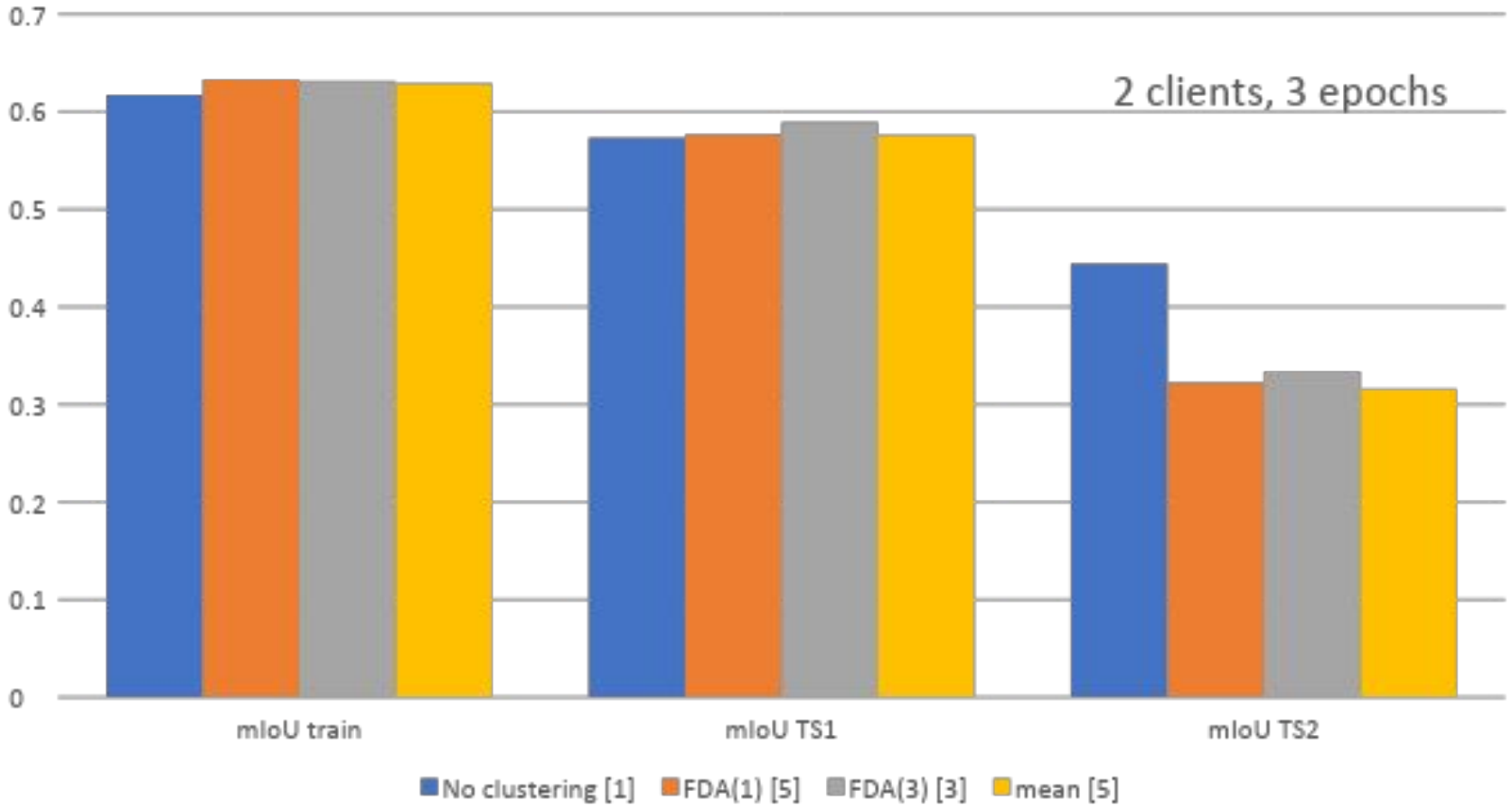
Clustering giving the best **silhouette** score is chosen



Clustered FL



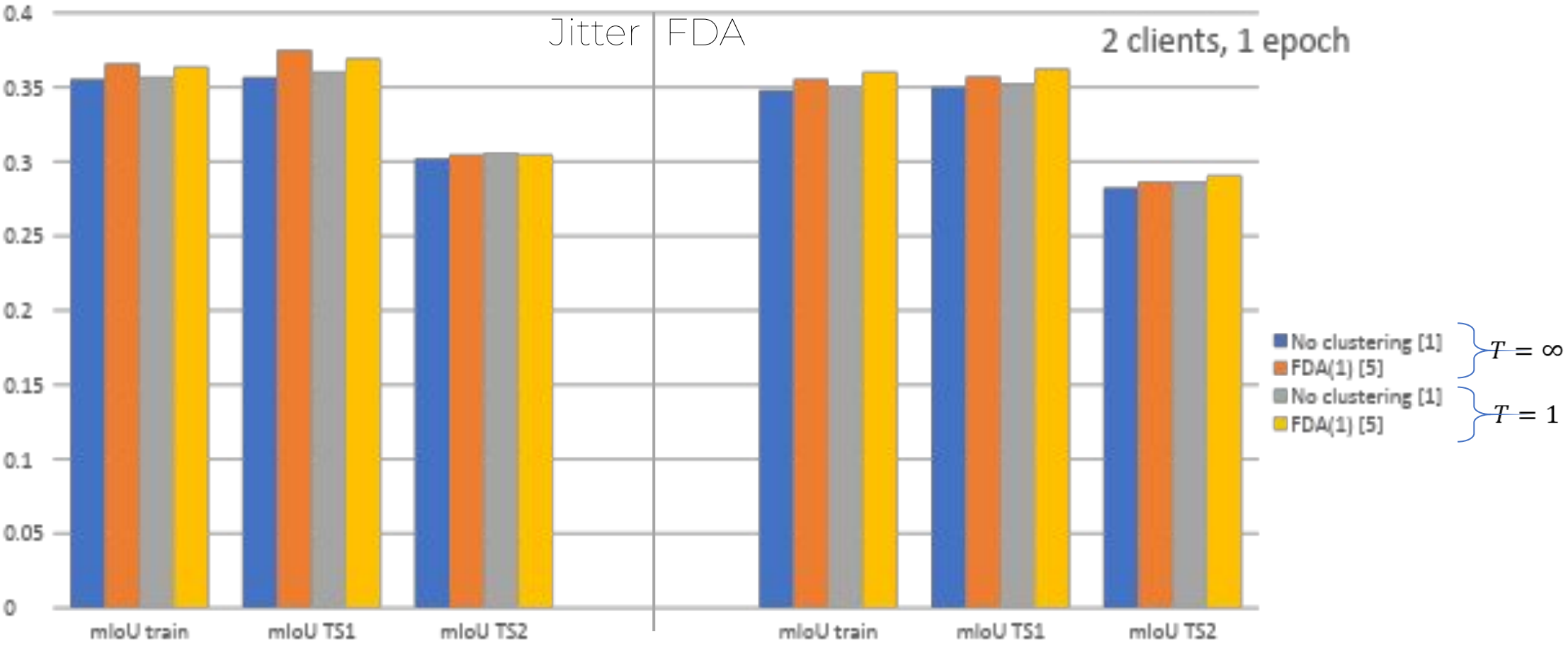
Clustering - supervised



⊕ Better in same domains

⊗ Worse in different domains

Clustering - unsupervised



⊕ Better in same domains

⊕ Slightly better in different domains

POLITECNICO
DI TORINO

Thanks for the attention!

Nicola Candellieri s314980, Lorenzo Sibille s314927,
Margherita Soro s304667

Model appendix

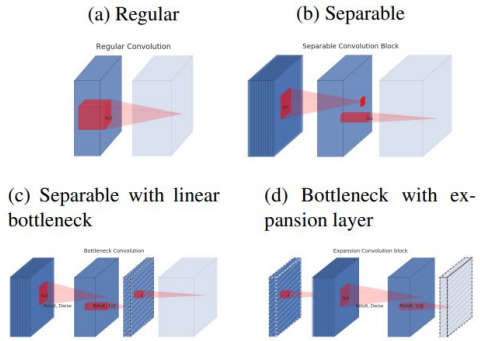
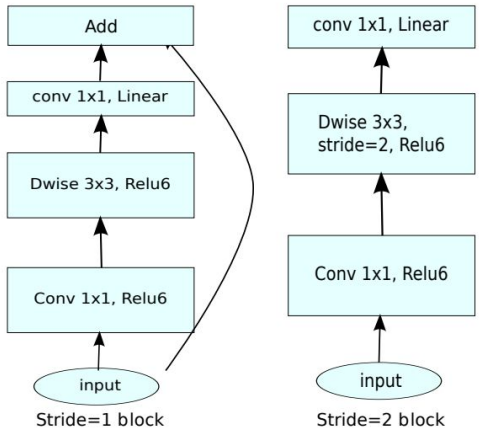


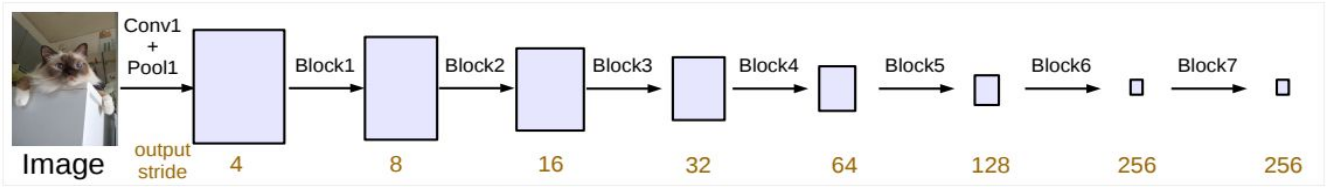
Figure 2: Evolution of separable convolution blocks. The diagonally hatched texture indicates layers that do not contain non-linearities. The last (lightly colored) layer indicates the beginning of the next block. Note: 2d and 2c are equivalent blocks when stacked. Best viewed in color.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

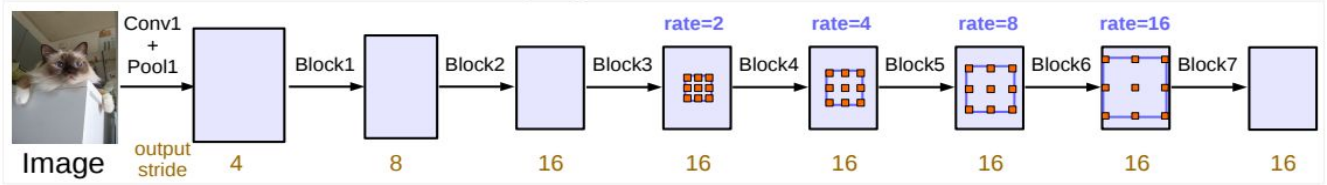
Table 2: MobileNetV2 : Each line describes a sequence of 1 or more identical (modulo stride) layers, repeated n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. All spatial convolutions use 3×3 kernels. The expansion factor t is always applied to the input size as described in Table 1.



(d) Mobilenet V2



(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

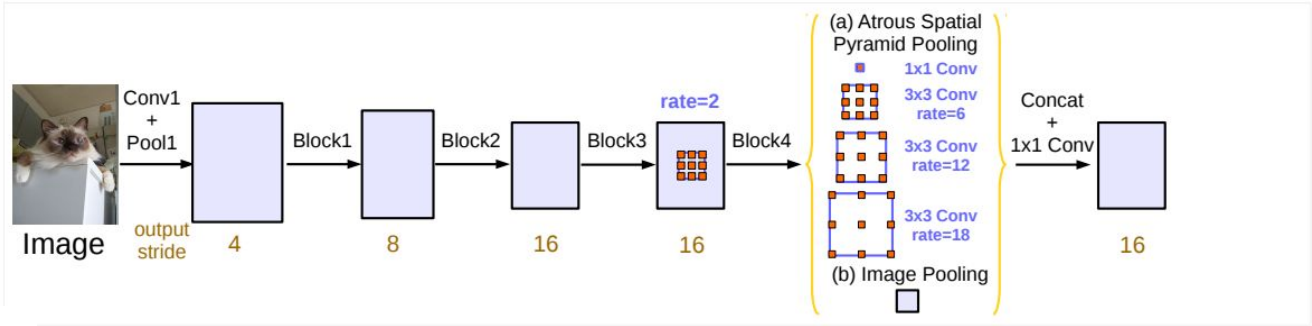


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.