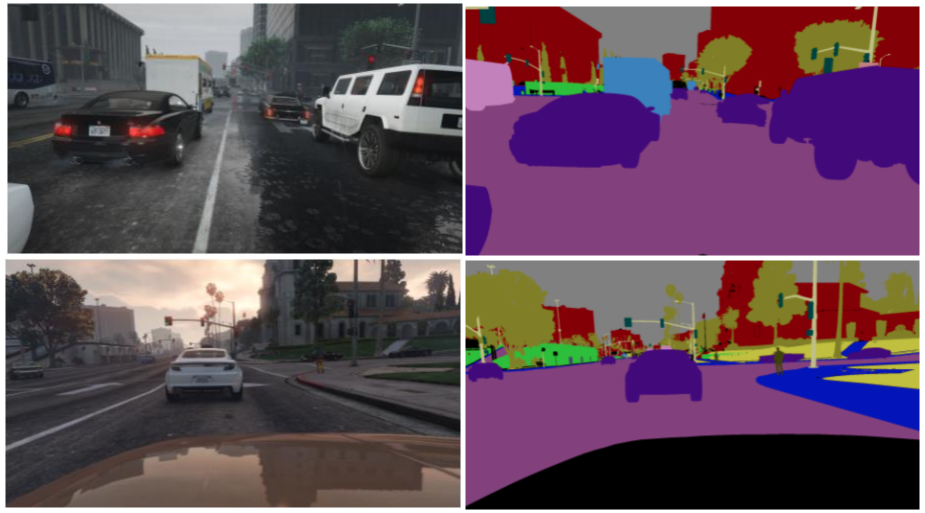
Generalize then Adapt

Source-Free Domain Adaptive Semantic Segmentation

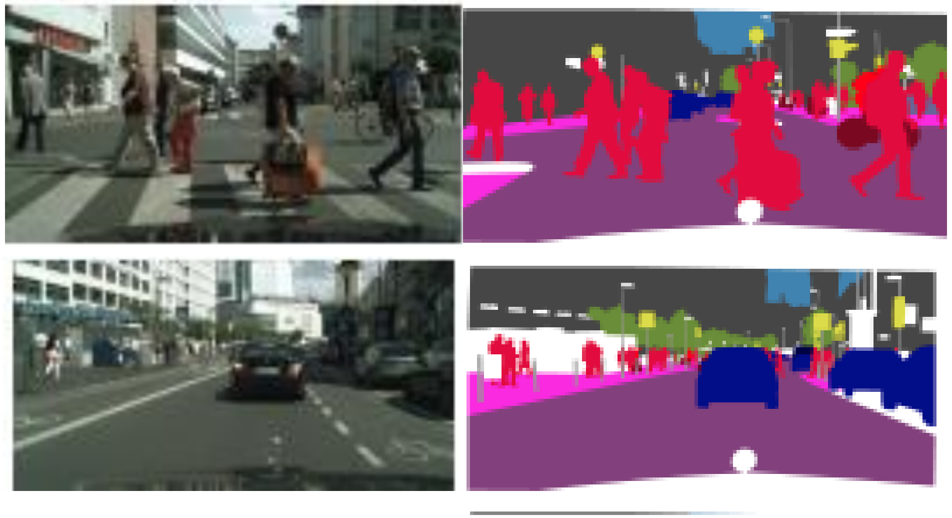
JN Kundu, A Kulkarni, A Singh, V Jampani, RV Babu

Presenters: Aditya, Sourish, Pranav
CS 8803 Machine Learning with Limited Supervision

Domain Adaptation

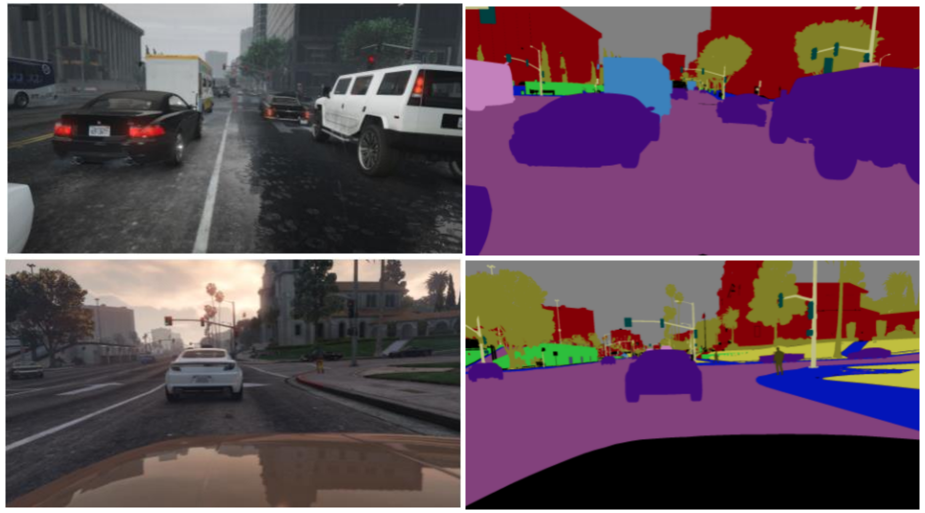


Source dataset (GTA)
Images and GT

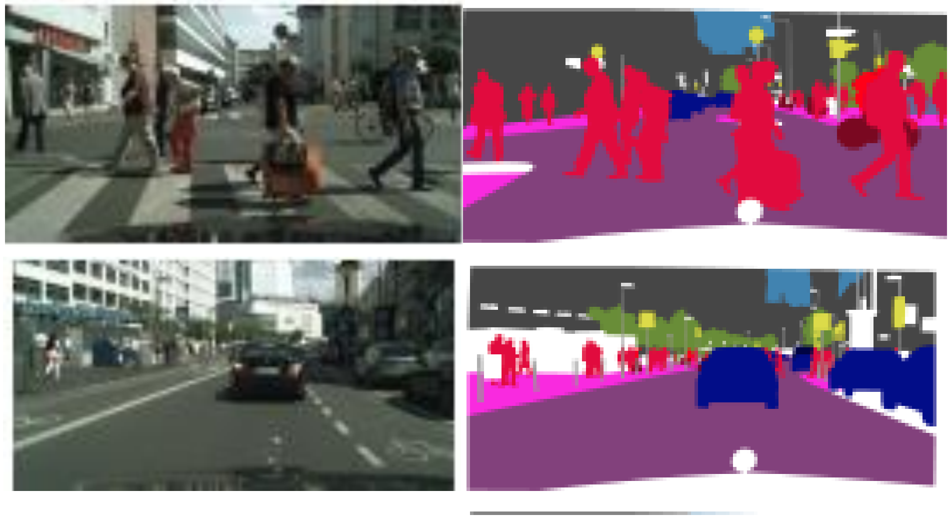


Target dataset (Cityscapes)
Images and GT

Domain Adaptation



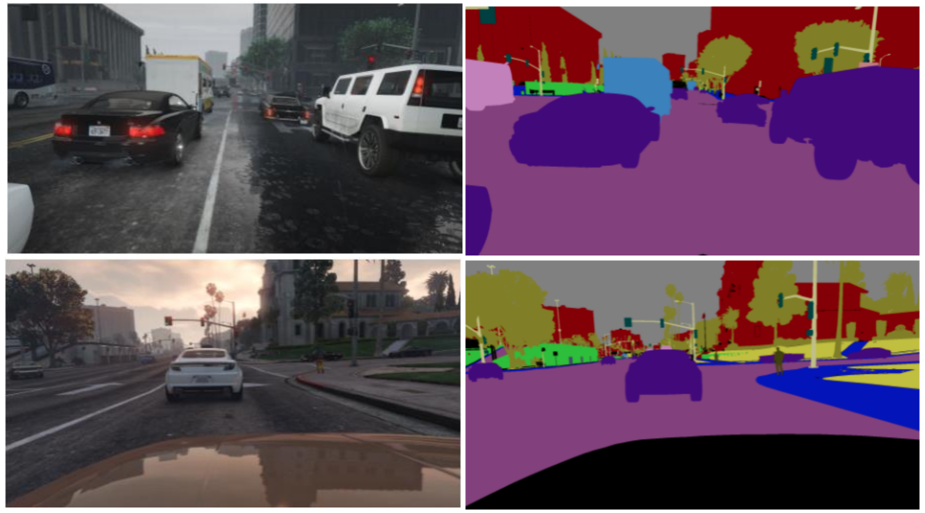
Source dataset (GTA)
Images and GT



Target dataset (Cityscapes)
Images and GT

Can we use the models learned on the source dataset to improve performance on the target dataset on the same task?

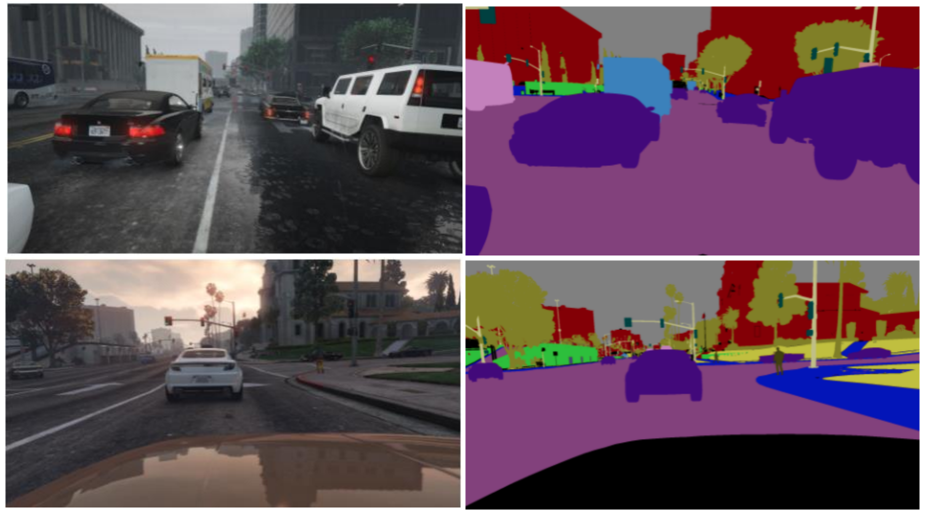
Unsupervised Domain Adaptation



Source dataset (GTA)
Images and GT

Target dataset (Cityscapes)
Images

Unsupervised Domain Adaptation



Source dataset (GTA)
Images and GT

Target dataset (Cityscapes)
Images

Can we do Domain Adaptation without target labels?

Source-free Unsupervised Domain Adaptation

SOURCE MODEL

Source dataset (GTA)
Images and GT

Target dataset (Cityscapes)
Images

Source-free Unsupervised Domain Adaptation

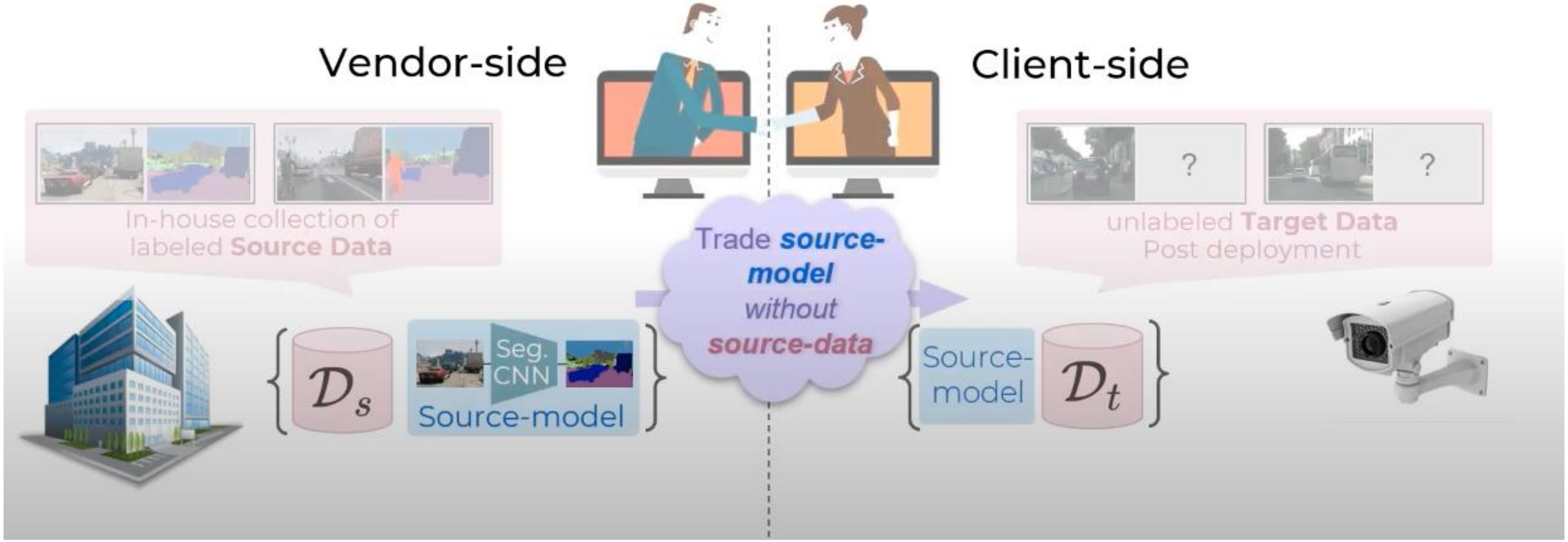
SOURCE MODEL

Source dataset (GTA)
Images and GT

Target dataset (Cityscapes)
Images

Can we do Unsupervised Domain Adaptation without having concurrent access to source / target data?

Why is SFUDA needed?



Related Work

Feature Space DA

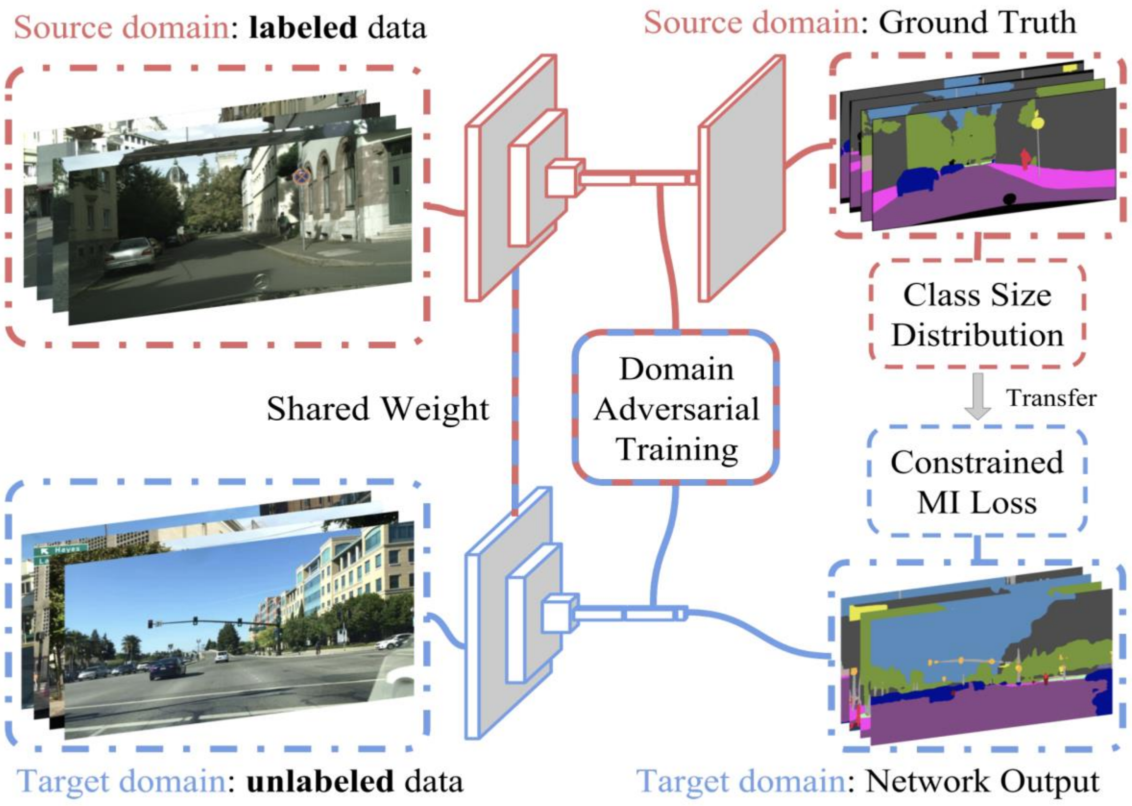


Figure 2: Overview of our pixel-level adversarial and constraint-based adaptation.

Image space DA

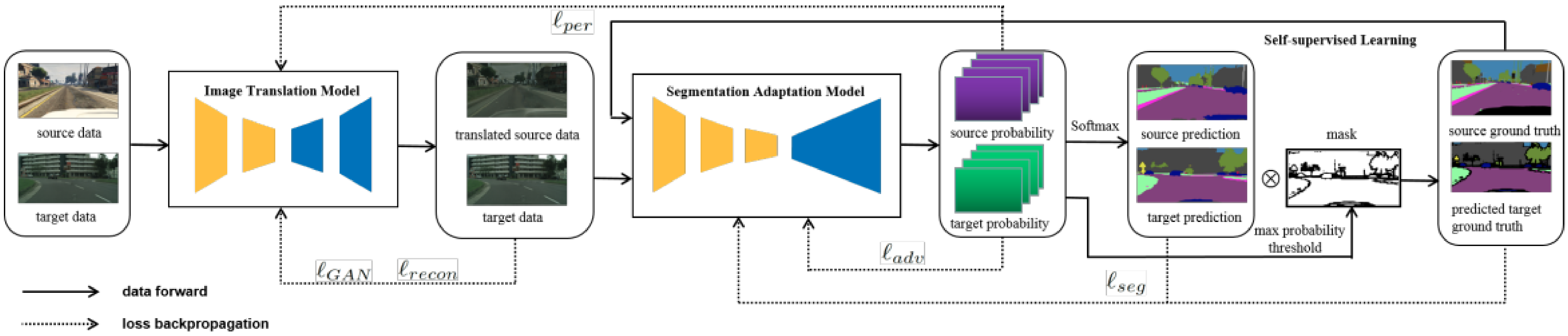


Figure 3: Network architecture and loss function

Multi-source DA

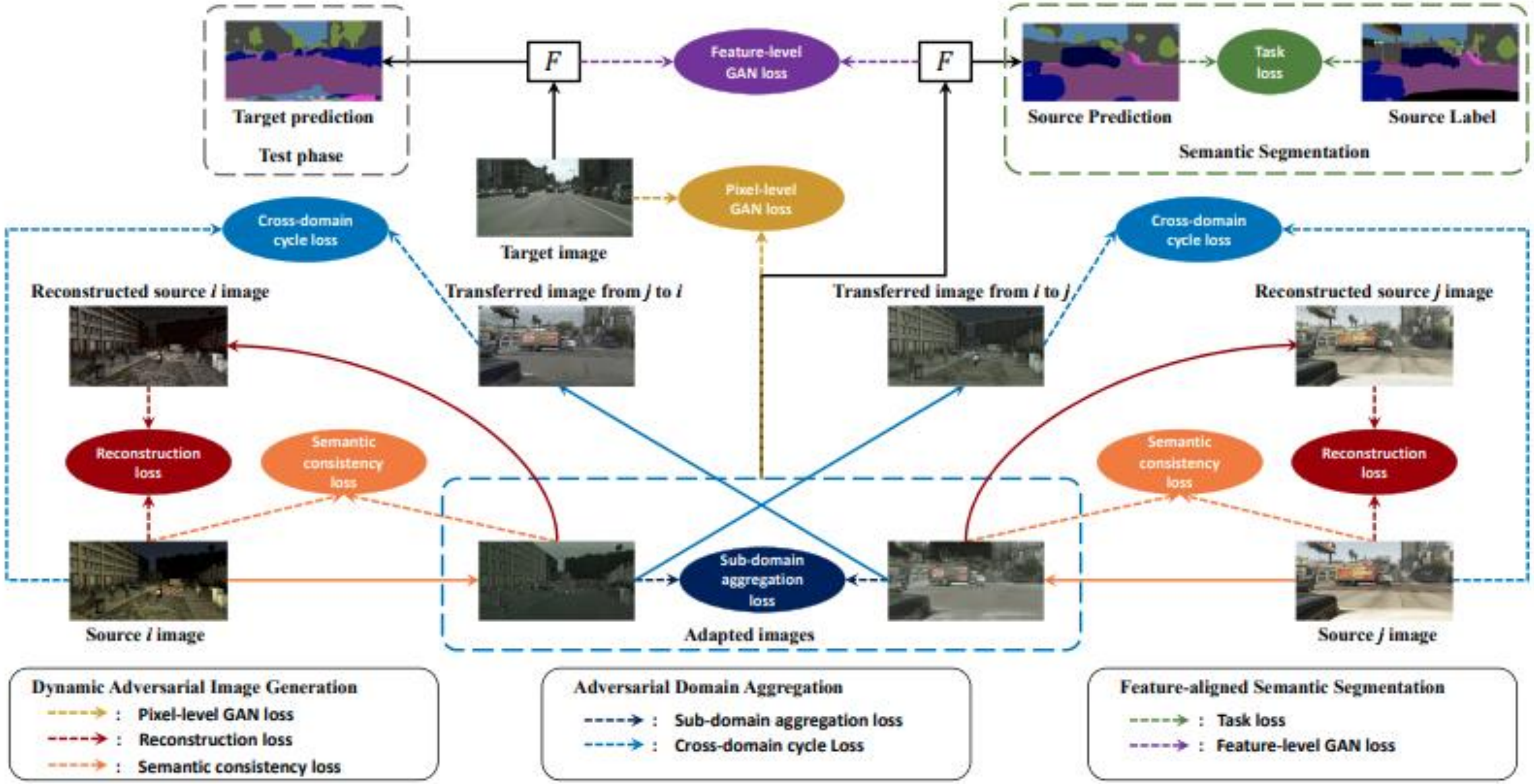


Figure 1: The framework of the proposed Multi-source Adversarial Domain Aggregation Network (MADAN). The colored solid arrows represent generators, while the black solid arrows indicate the segmentation network F. The dashed arrows correspond to different losses.

Others

Source-free

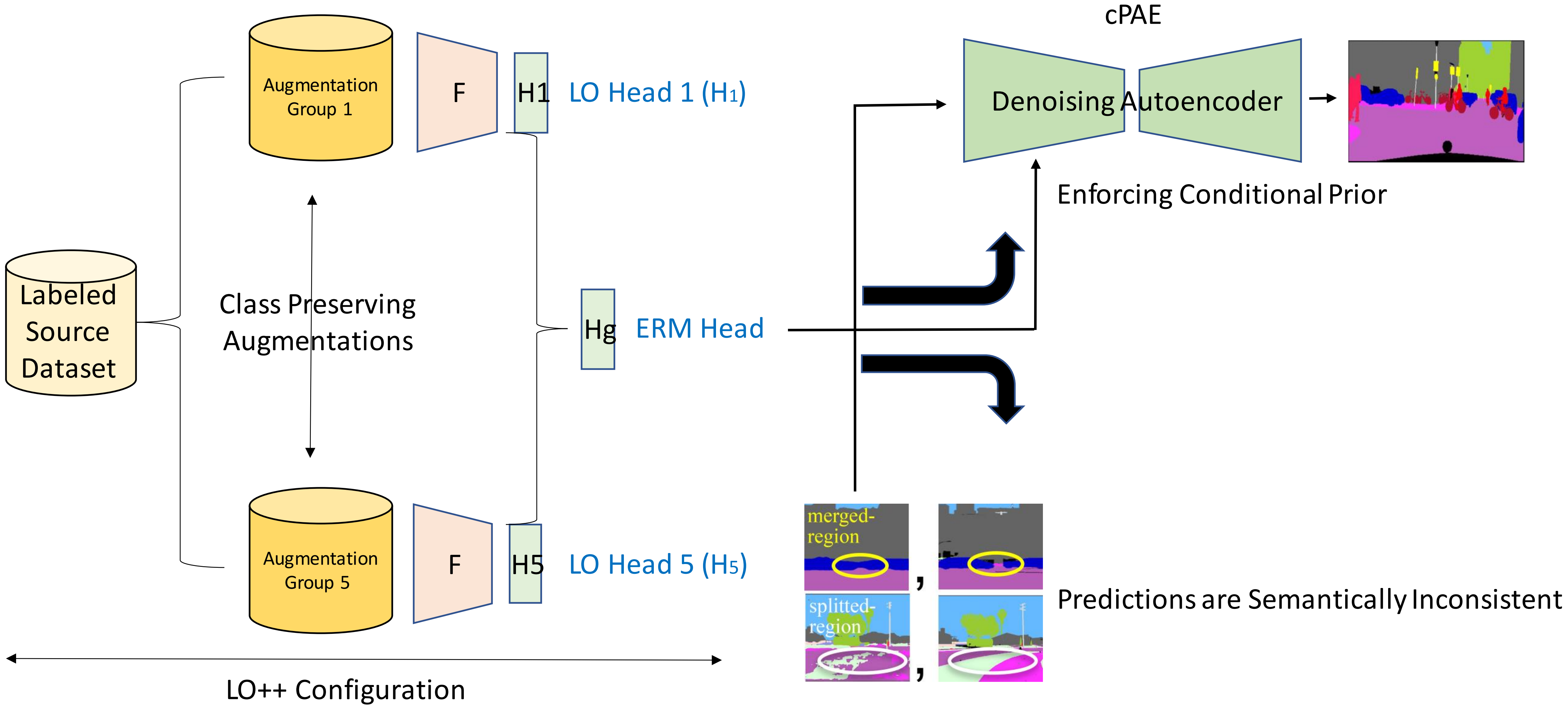
- Unsupervised loss: Entropy minimization, class-ratio alignment
- Distillation, Self-supervision

Self-training

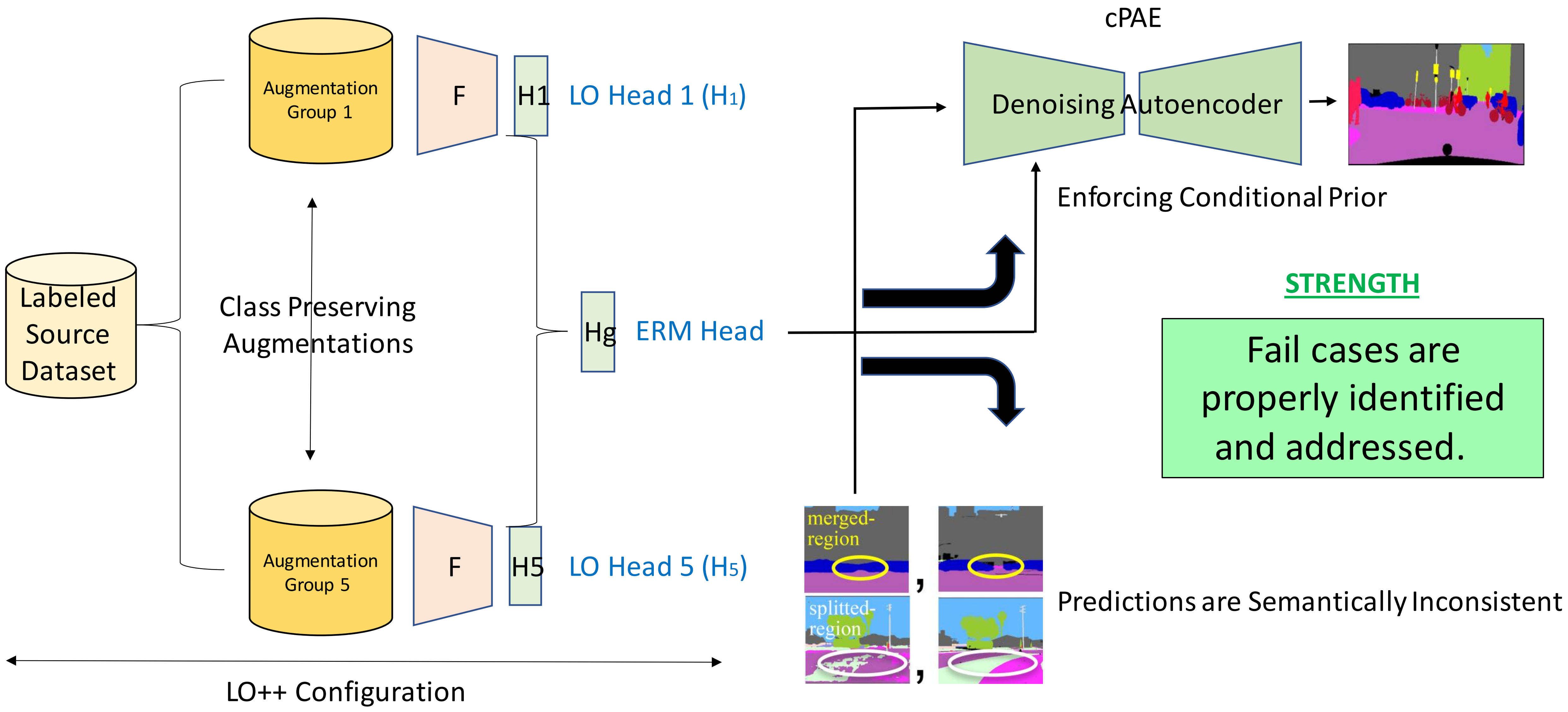
- Using highly confident pseudo-labels for target domain training

Approach

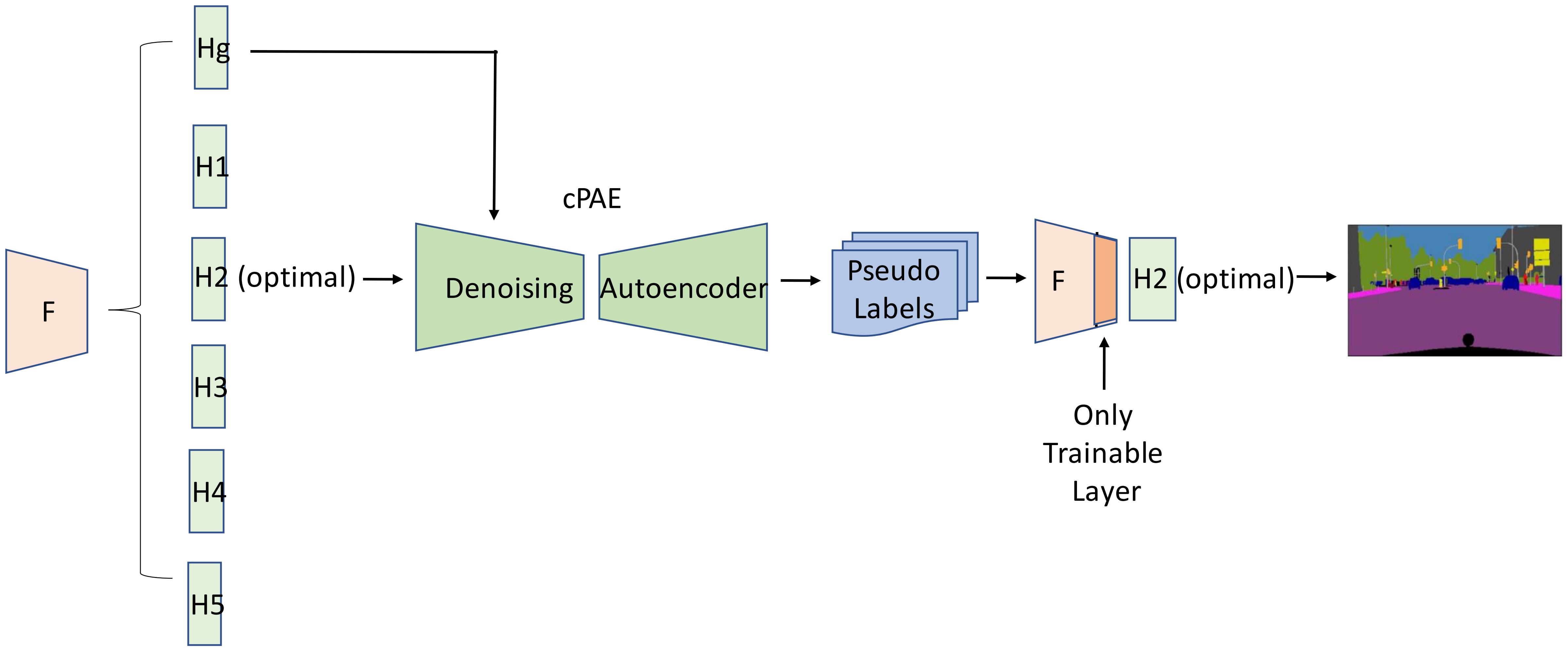
Approach – Vendor Strategy



Approach – Vendor Strategy



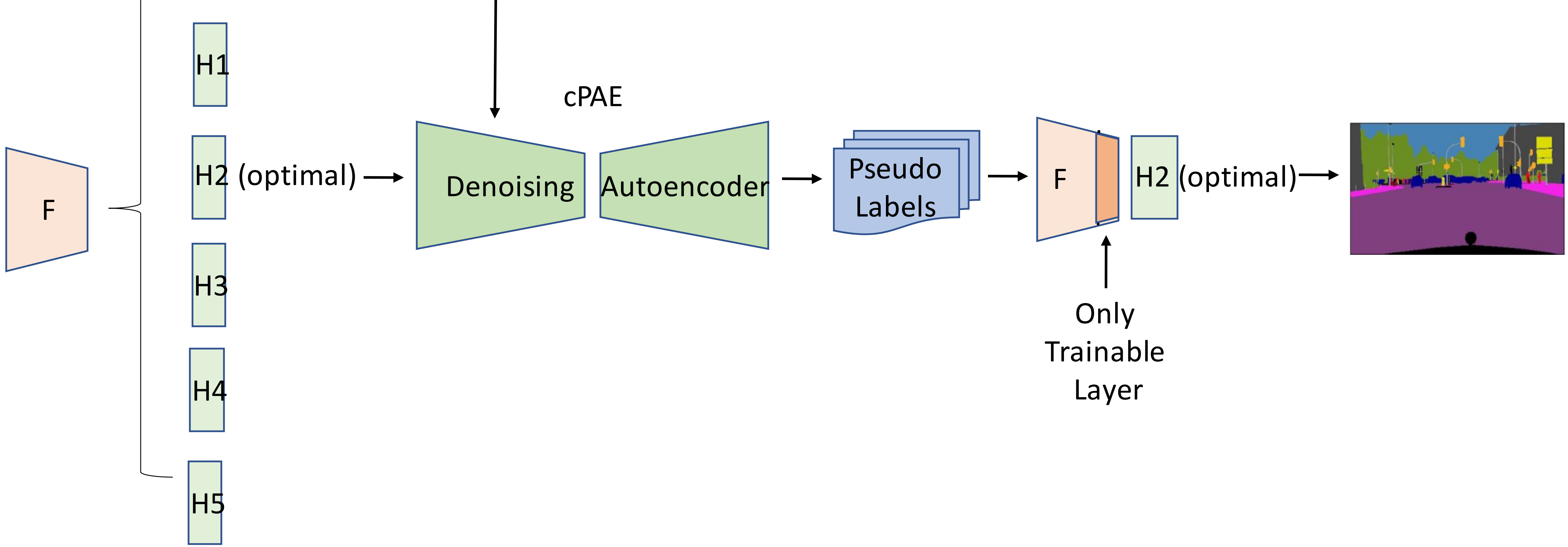
Approach – Client Strategy



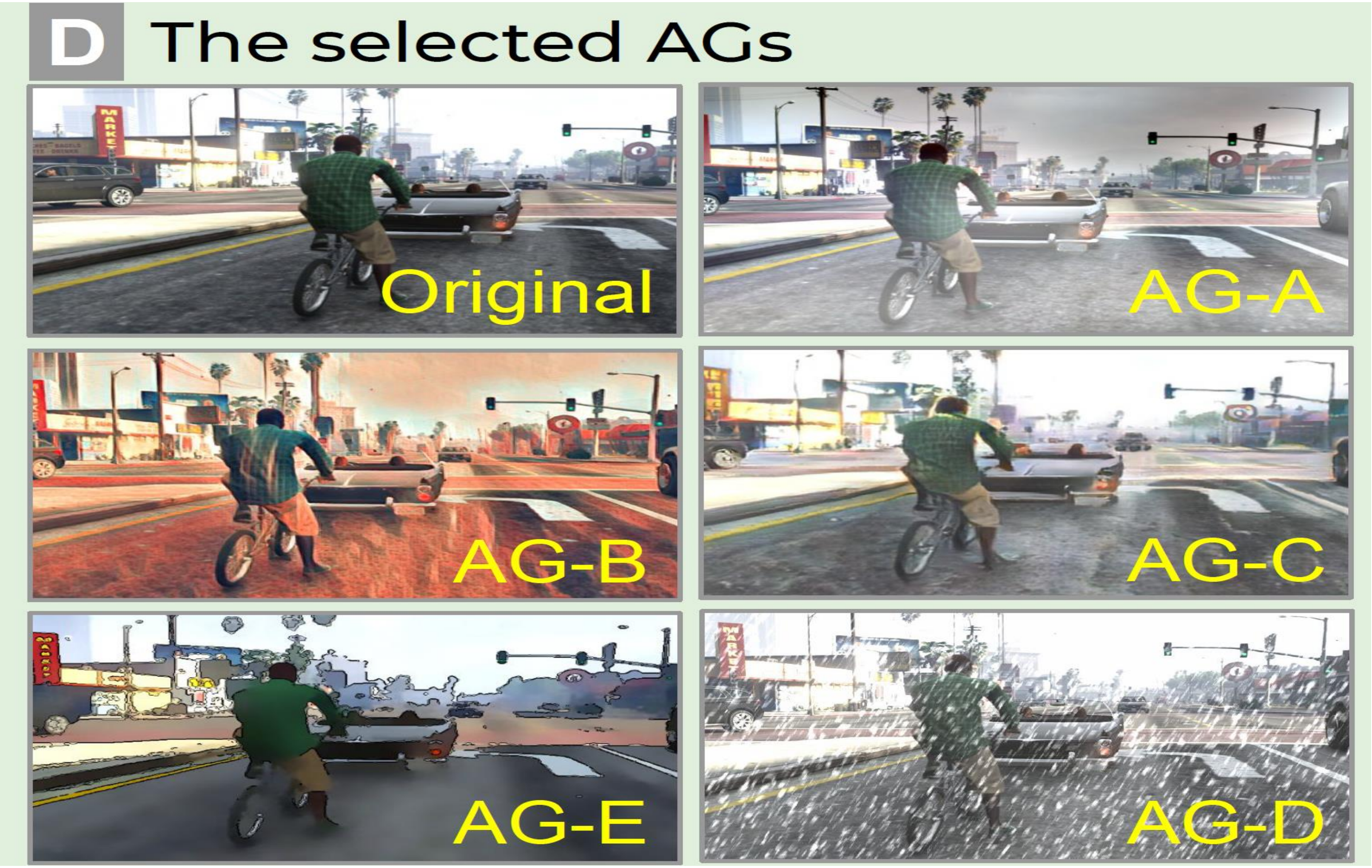
Approach – Client Strategy

STRENGTH

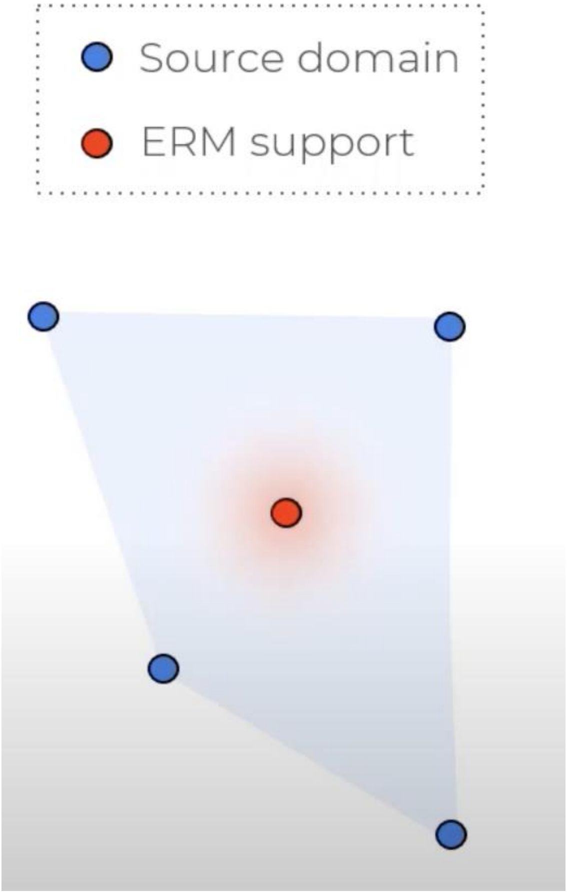
The tradeoff between "vendor-side knowledge" and "label-noise + info redundancy" in pseudo-labels is well-handled.



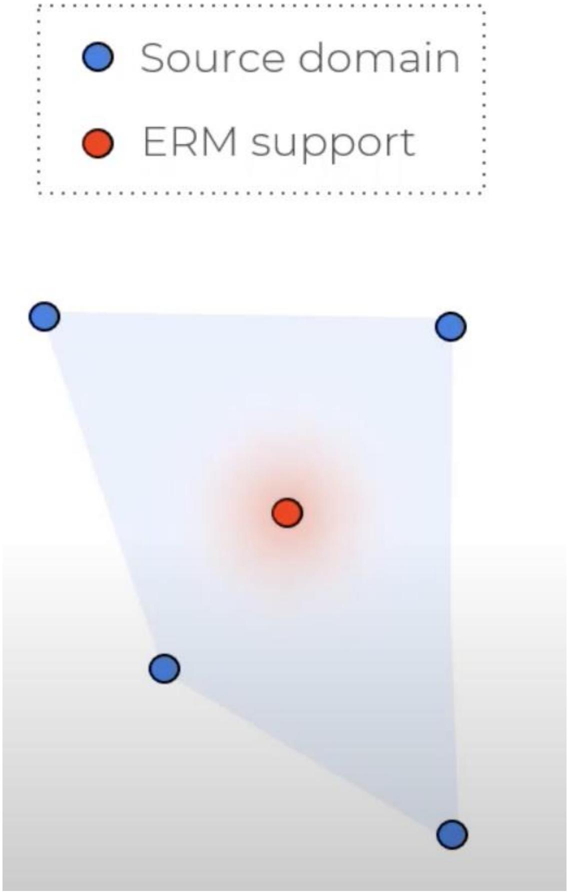
Approach - Class Preserving Augmentations

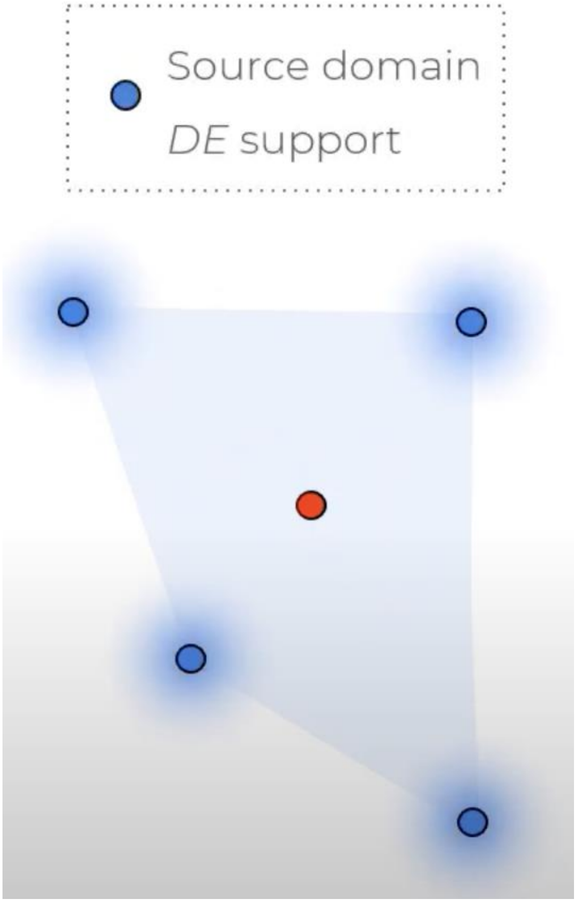


Approach - Vendor Side Training Approaches

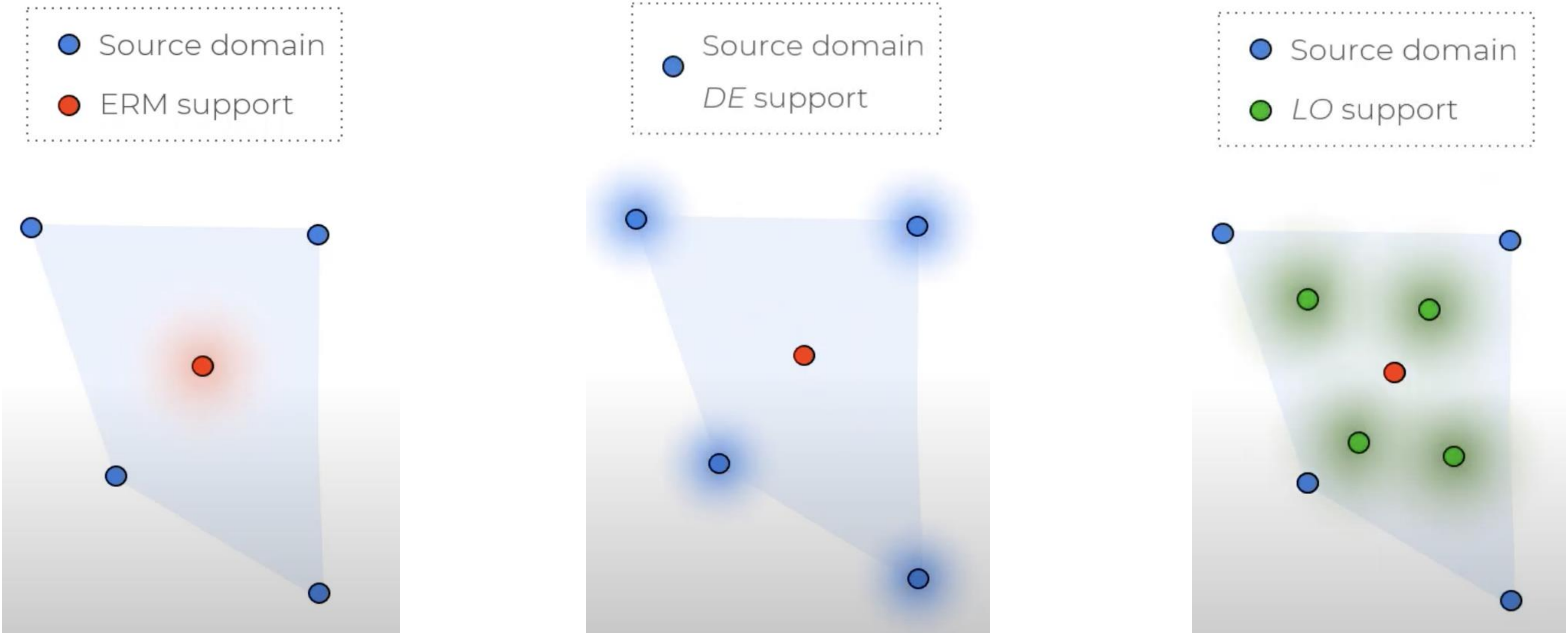


Approach - Vendor Side Training Approaches

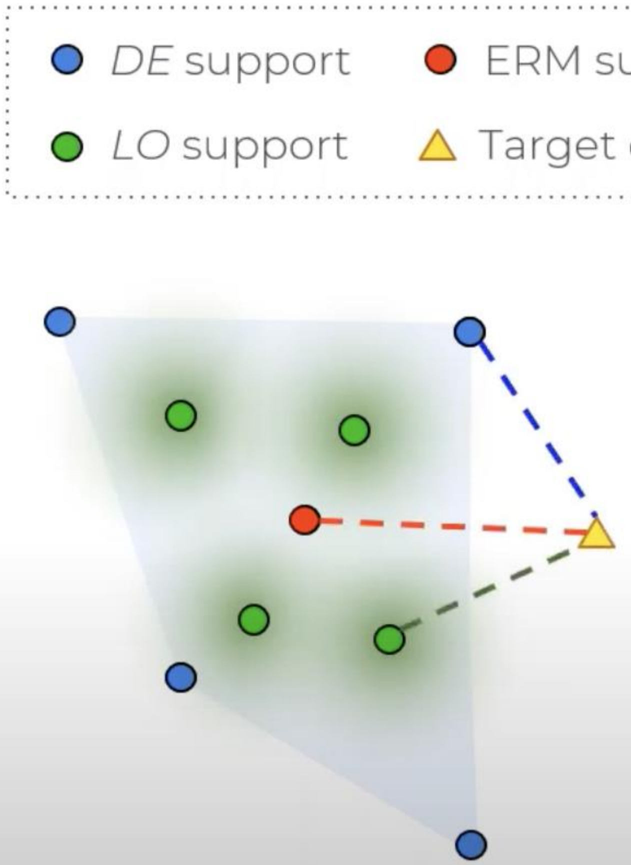




Approach - Vendor Side Training Approaches

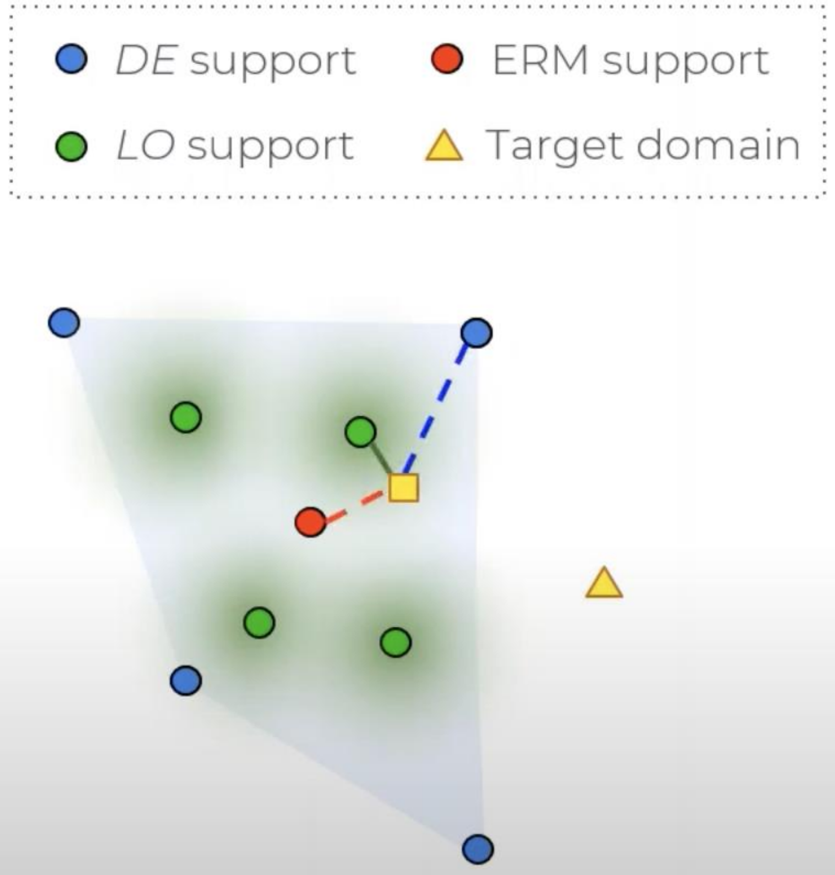


Approach – Analysis of Hypothesis Supports



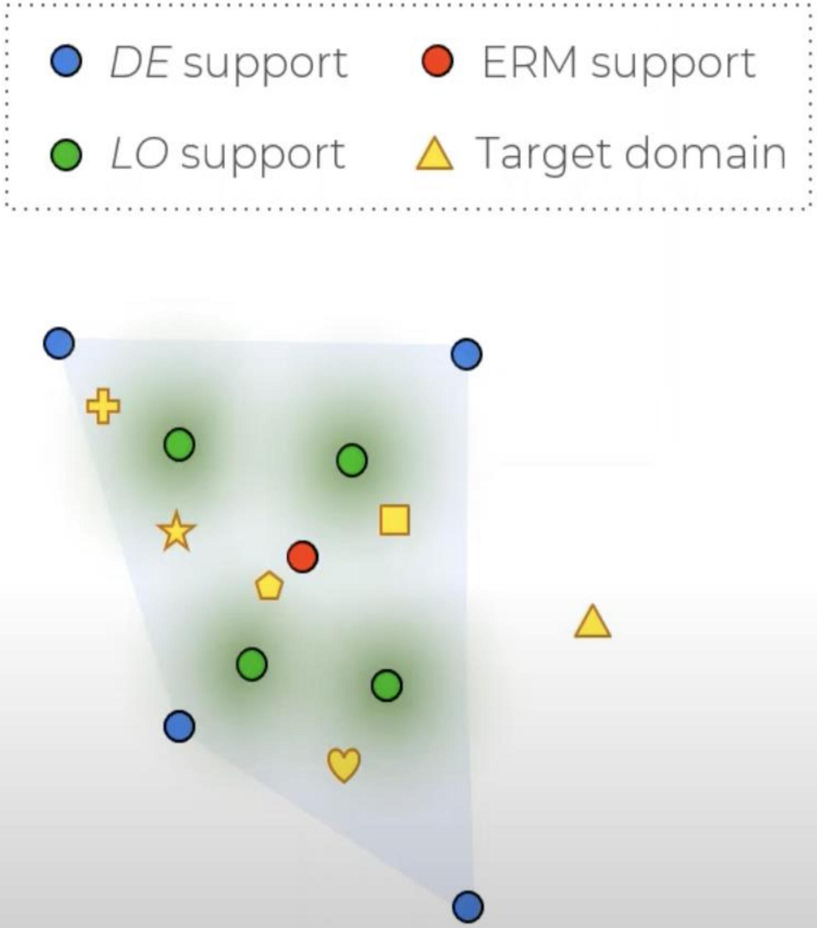
Target scenarios	Deployable vendor-side configuration		
	ERM	*DE*++	*LO*++
t_{j_1}: △	✗	✗	✓

Approach – Analysis of Hypothesis Supports



Target scenarios	Deployable vendor-side configuration		
	ERM	*DE*++	*LO*++
t_{j_1}: △	✗	✗	✓
t_{j_2}: □	✗	✗	✓

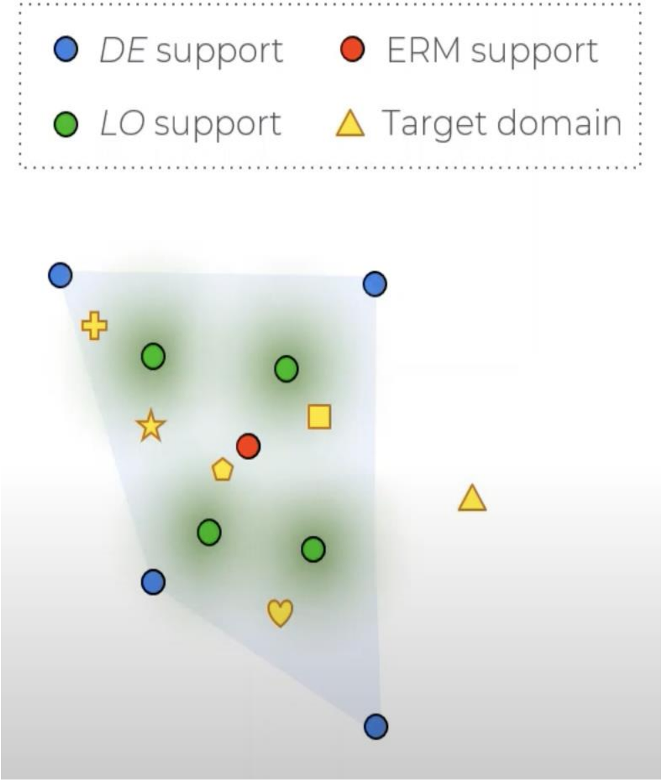
Approach – Analysis of Hypothesis Supports



Target scenarios	Deployable vendor-side configuration		
	ERM	*DE*++	*LO*++
t_{j_1}: △	✗	✗	✓
t_{j_2}: □	✗	✗	✓
t_{j_3}: ✚	✗	✓	✗
t_{j_4}: ☆	✗	✗	✓
t_{j_5}: ⬠	✓	✗	✓
t_{j_6}: ♡	✗	✗	✓

LO++ supports most target scenarios

Approach – Analysis of Hypothesis Supports



Result 1. *Consider DE++ hypothesis space $\mathcal{A}^{\mathrm{DE++}}$, LO++ hypothesis space $\mathcal{A}^{\mathrm{LO++}}$, and unseen target data $\mathcal{D}_t$. Then,*

$$\begin{aligned} \epsilon_t(h \in \mathcal{A}^{\mathrm{LO++}}) &\leq \epsilon_t(h \in \mathcal{H}^{\mathrm{ERM}}) \\ \epsilon_t(h \in \mathcal{A}^{\mathrm{DE++}}) &\leq \epsilon_t(h \in \mathcal{H}^{\mathrm{ERM}}) \end{aligned} \tag{2}$$

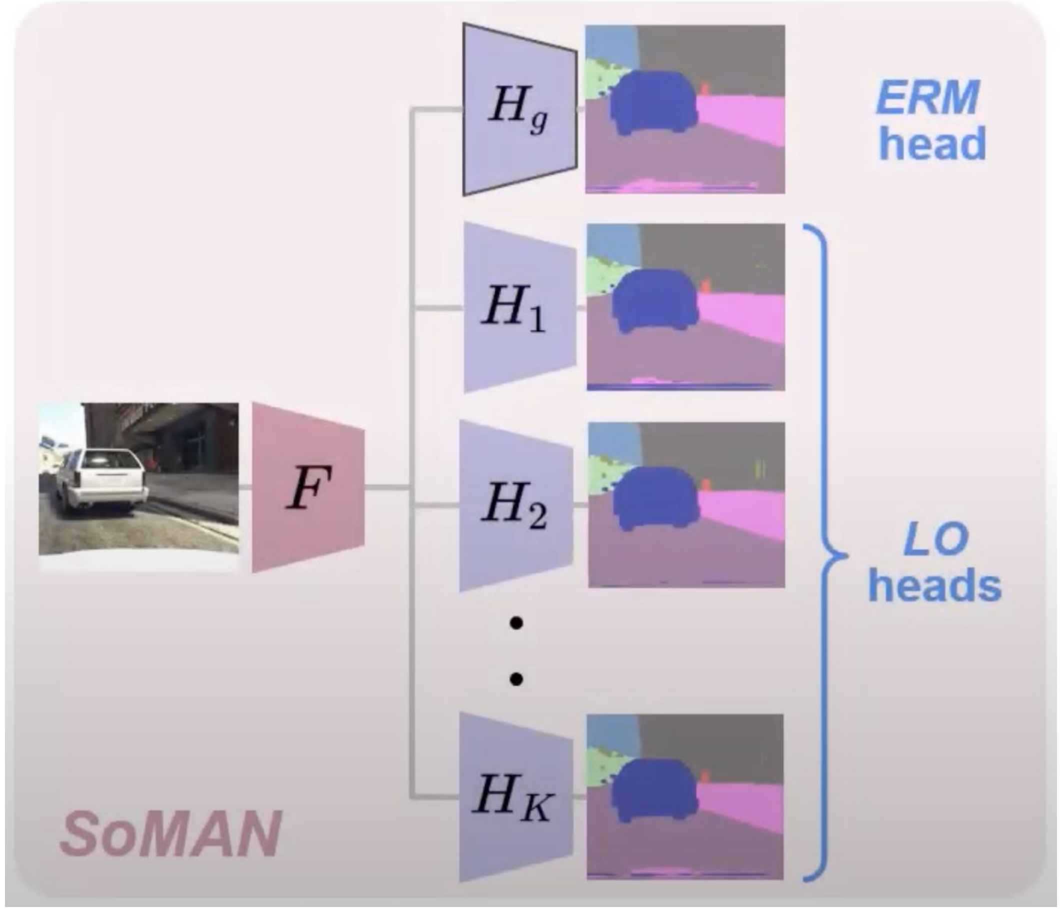
Approach – Analysis of Hypothesis Supports

Target scenarios	Deployable vendor-side configuration		
	ERM	*DE*++	*LO*++
t_{j_1}: △	✗	✗	✓
t_{j_2}: □	✗	✗	✓
t_{j_3}: ✚	✗	✓	✗
t_{j_4}: ☆	✗	✗	✓
t_{j_5}: ⬠	✓	✗	✓
t_{j_6}: ♡	✗	✗	✓

WEAKNESS

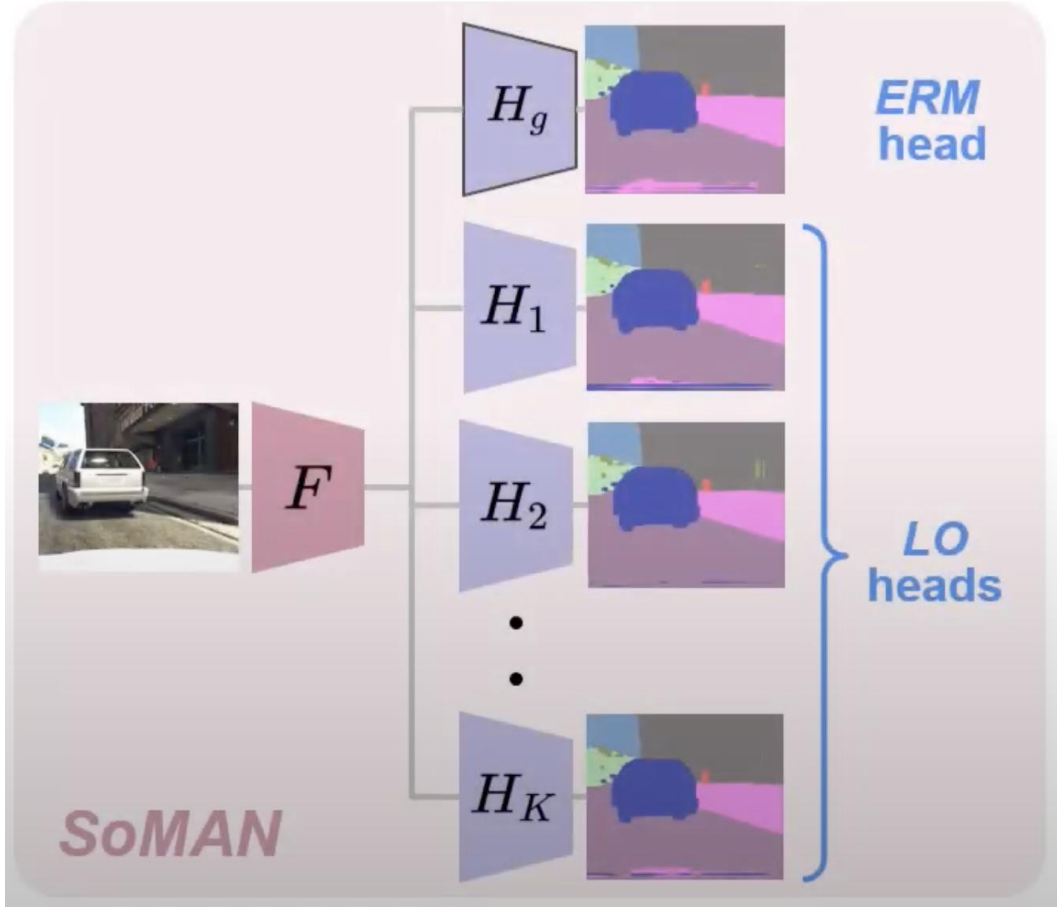
LO++ supports most target scenarios.
Is the justification correct?

Approach – SoMAN Training



LO++ Configuration

Approach – SoMAN Training



LO++ Configuration

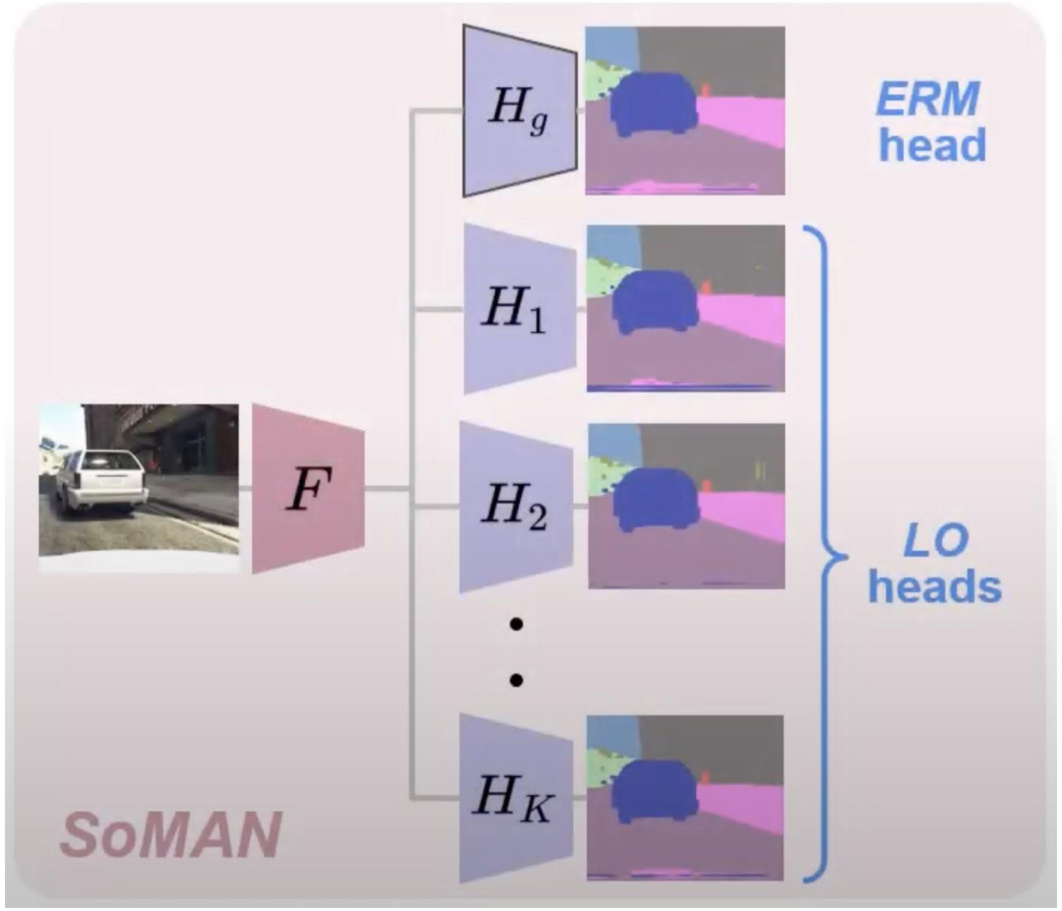
Train F on source dataset

Randomly augment and perform ERM to get Hg

Incorporate LO strategy to get H1, H2,..., Hk

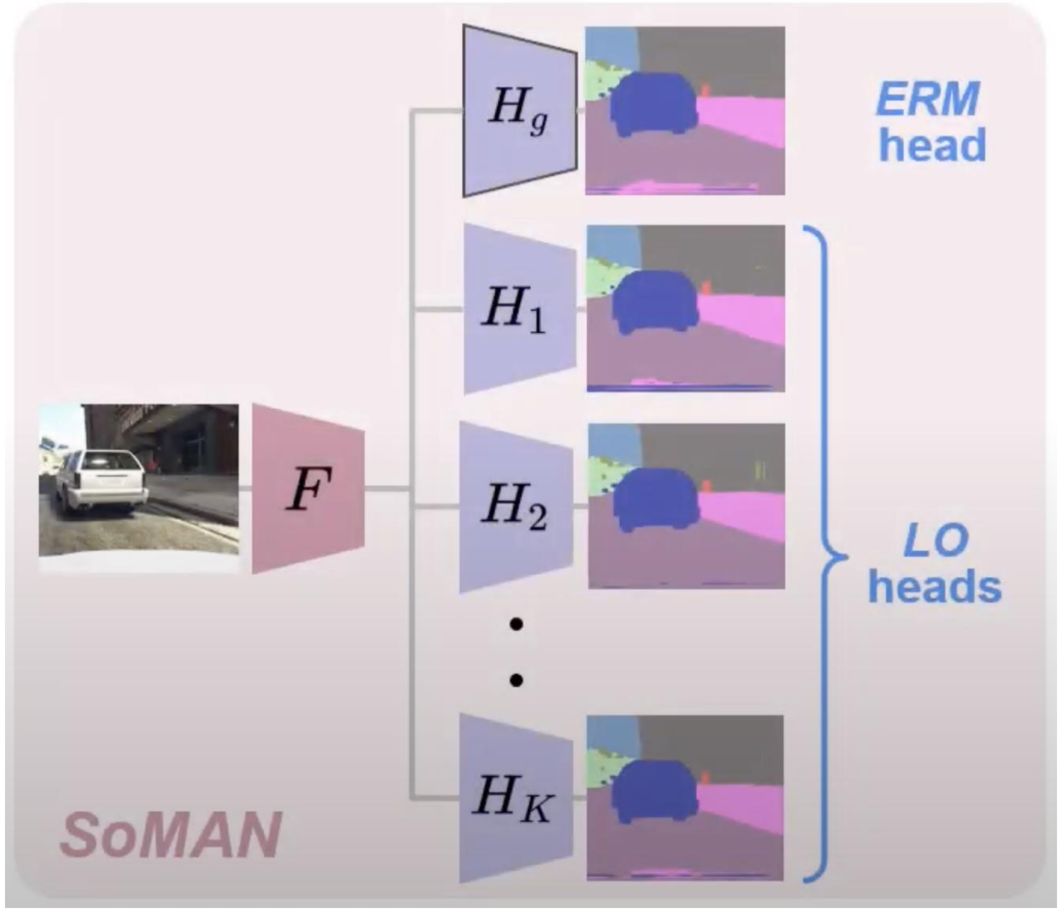
Update respective weights using CE Loss

Approach – SoMAN Training



LO++ Configuration

Approach – SoMAN Training

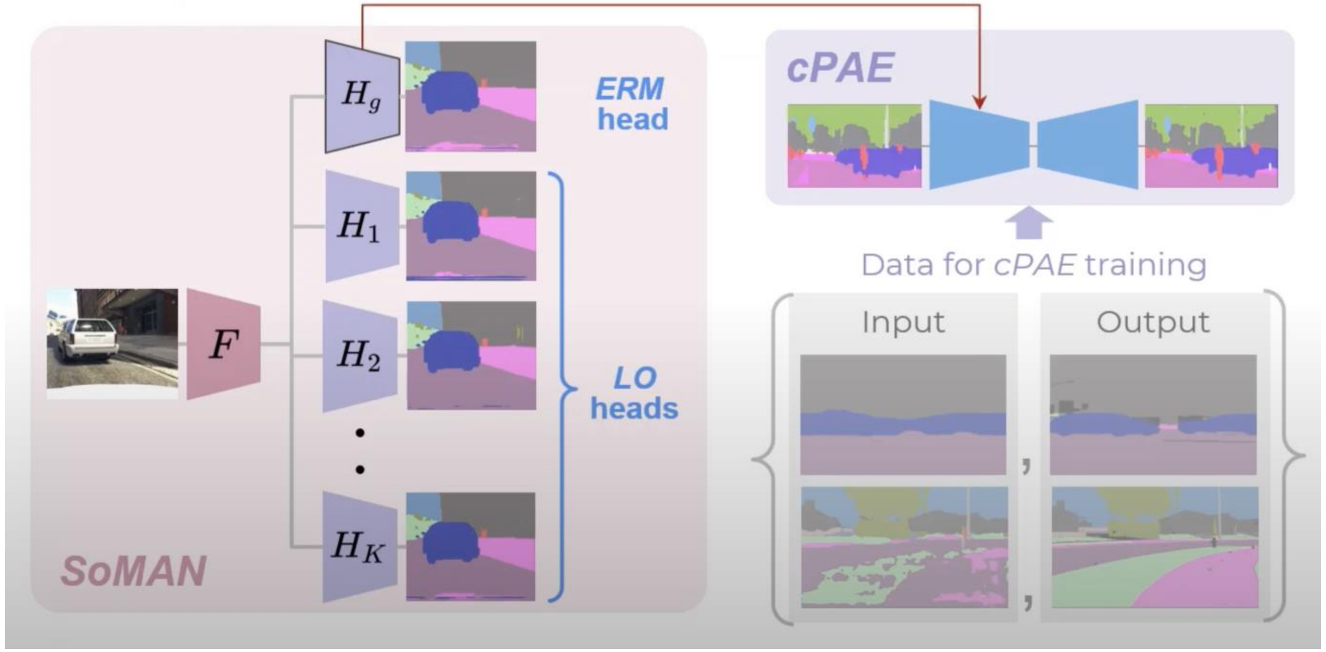


LO++ Configuration

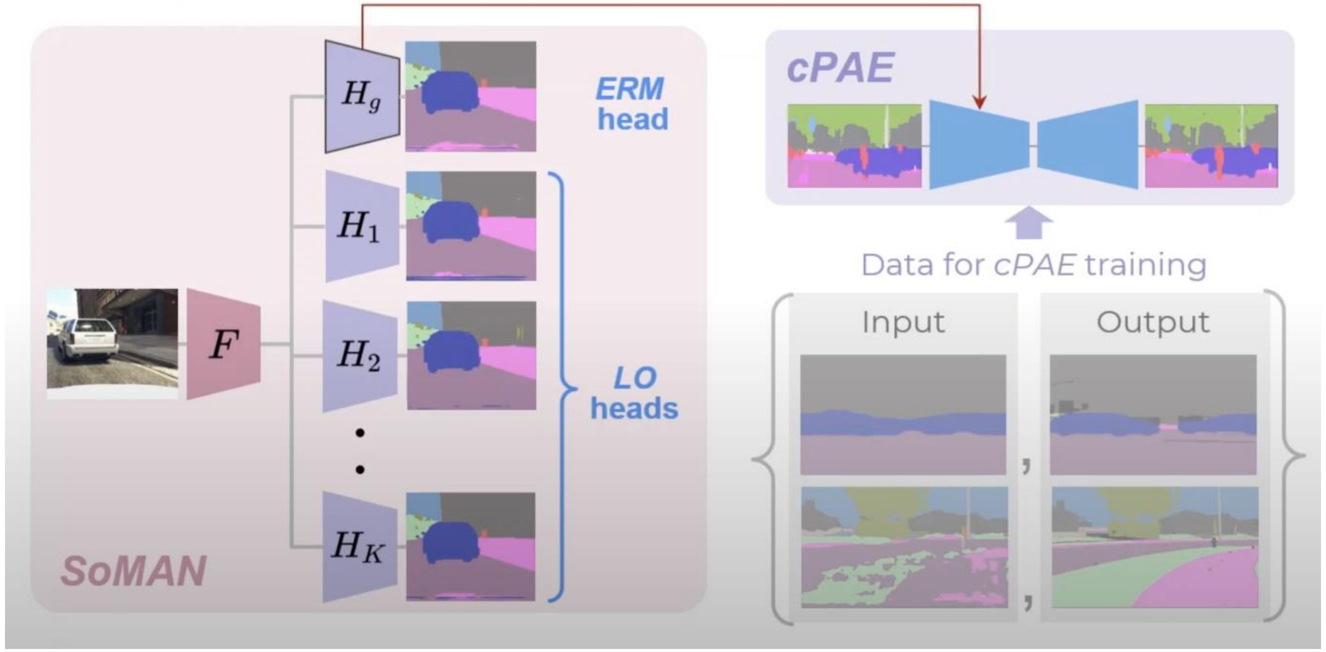
WEAKNESS

- Is this the optimal way of designing the architecture?
- Additional heads for DE supports can be inserted along with H1, H2,…,Hk and Hg to incorporate all the possible cases!
- As vendor-side training is a one-time work, the tradeoff between performance and additional computational overhead seems insignificant!

Approach – Vendor Side Training CPAE



Approach – Vendor Side Training CPAE



Freeze feature extractor and classification heads

Randomly augment and get prediction map from respective LO head to use it as noise

Refine the prediction map using domain-generic features (Fg)

Calculate CE loss between the refined prediction map and ground truth to backpropagate

Paired-data for training CPAE

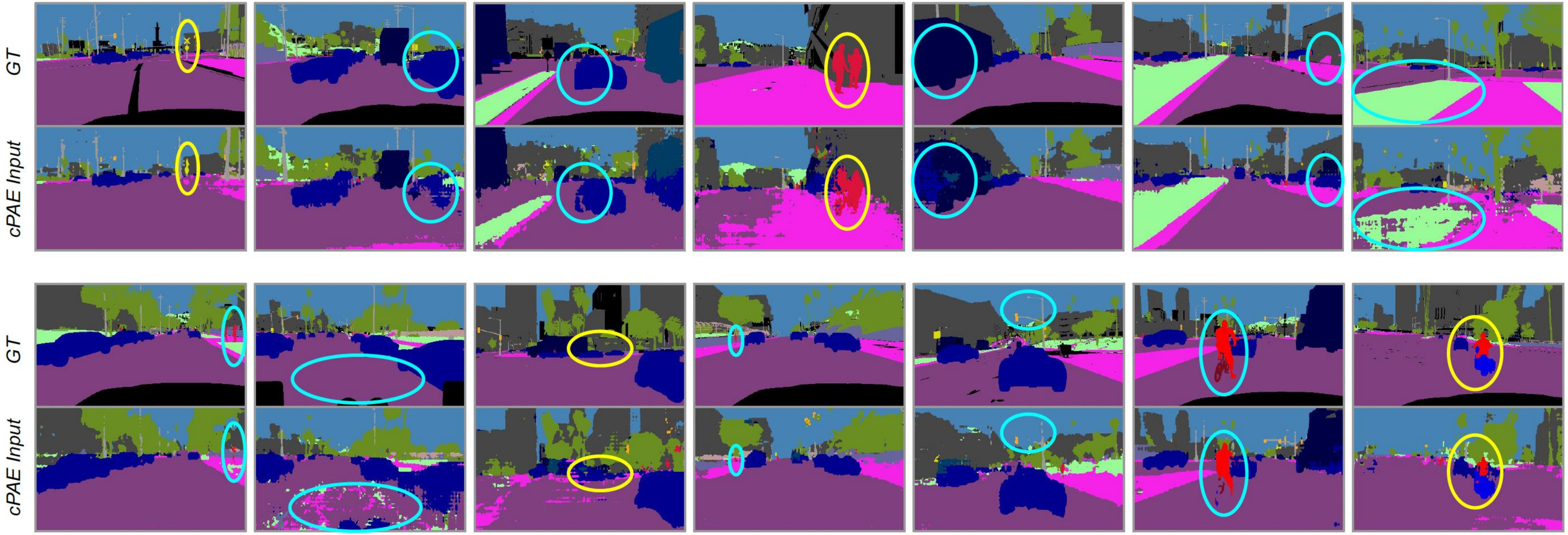
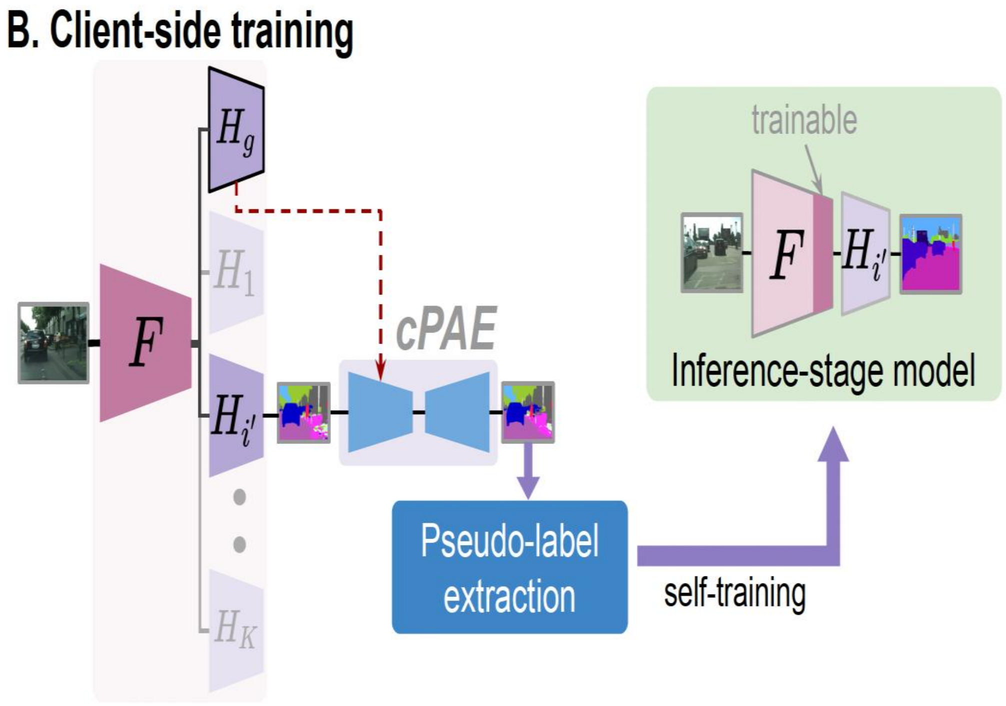
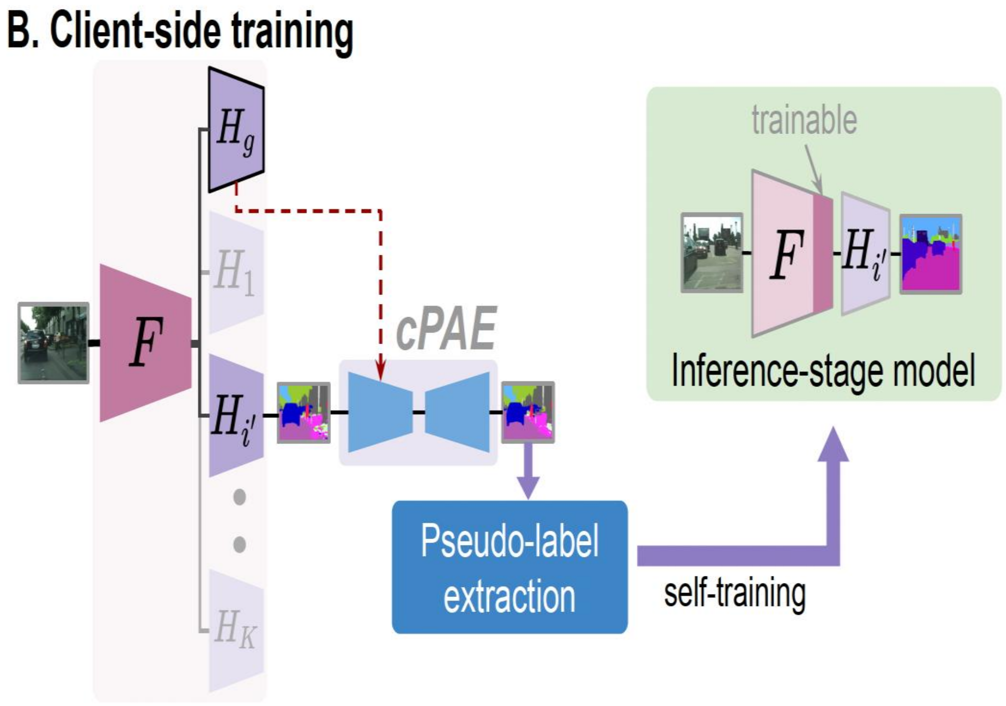


Figure 2. Paired samples for cPAE training. cPAE is trained as a denoising autoencoder to encourage structural regularity in segmentation predictions and alleviate merged-region (yellow circle) and splitted-region (blue circle) problems. *Best viewed in color.*

Approach – Client-Side Training



Approach – Client-Side Training



Identify optimal head using minimum self-entropy over target set

Calculate class probabilities using output map from cPAE

Set class-wise thresholds such that only the top 33% predictions are used as pseudo-labels

Retrieve the top 33% predictions to build the pseudo-label set

Perform training using the pseudo-labeled dataset

Experimental Setup

Experimental Setup – Network Architecture

SoMAN Architecture:

- Deeplabv2 w/ ResNet101
- FCN8s with VGG16

CPAE Architecture:

	Layer	Input	Type	Filter | Stride | Dilation	Output Size
Encoder	C_1	$\hat{y}$	Conv*	$7 \times 7, 64$ | 1 | -	$512 \times 1024 \times 64$
	C_2	C_1	Conv*	$3 \times 3, 128$ | 2 | -	$256 \times 512 \times 128$
	C_3	C_2	Conv*	$7 \times 7, 128$ | 1 | -	$256 \times 512 \times 128$
	C_4	C_3	Conv*	$3 \times 3, 256$ | 2 | -	$128 \times 256 \times 256$
	C_5	C_4	Conv*	$7 \times 7, 256$ | 1 | -	$128 \times 256 \times 256$
	C_6	C_5	Conv*	$3 \times 3, 512$ | 2 | -	$64 \times 128 \times 512$
	C_7	$C_6, F_g(x)$	$\Vert$	-	$64 \times 128 \times 2560$
	C_8	C_7	Dconv	$3 \times 3, 512$ | 1 | 2	$64 \times 128 \times 512$
	C_9	C_7	Dconv	$3 \times 3, 512$ | 1 | 4	$64 \times 128 \times 512$
	C_{10}	C_7	Dconv	$3 \times 3, 512$ | 1 | 8	$64 \times 128 \times 512$
	C_{11}	C_7	Dconv	$3 \times 3, 512$ | 1 | 16	$64 \times 128 \times 512$
	C_{12}	C_8, C_9, C_{10}, C_{11}	$\bigoplus$	-	$64 \times 128 \times 512$
	C_{13}	C_{12}	Conv+Tanh	$1 \times 1, 512$ | 1 | -	$64 \times 128 \times 512$
Decoder	C_{14}	C_{13}	Conv**	$3 \times 3, 512$ | 1 | -	$64 \times 128 \times 512$
	C_{15}	C_{14}	Conv*	$3 \times 3, 512$ | 1 | -	$64 \times 128 \times 512$
	C_{16}	C_{15}	Conv*	$7 \times 7, 256$ | 1 | -	$64 \times 128 \times 256$
	C_{17}	C_{16}	Tconv*	$3 \times 3, 256$ | 2 | -	$128 \times 256 \times 256$
	C_{18}	C_{17}	Conv*	$7 \times 7, 128$ | 1 | -	$128 \times 256 \times 128$
	C_{19}	C_{18}	Tconv*	$3 \times 3, 64$ | 2 | -	$256 \times 512 \times 64$
	C_{20}	C_{19}	Conv	$7 \times 7, 19$ | 1 | -	$256 \times 512 \times 19$
	Upsampling	C_{20}	Interpolation	-	$512 \times 1024 \times 19$

Experimental Setup - Datasets

- GTA5 dataset:
    - 24966 synthetic images with pixel-level semantic annotation

Experimental Setup - Datasets

- SYNTHIA dataset
  - 20,000+ HD images from video streams + 20,000+ HD images from snapshots
  - European style town, modern city, highway, and green areas

Experimental Setup - Datasets

- Cityscapes dataset:
  - large-scale dataset - stereo video sequences recorded in street
  - 50 different cities
  - high quality pixel-level annotations of 5000 frames + 20,000 weakly annotated frames.

Experimental Setup – Augmentation Groups

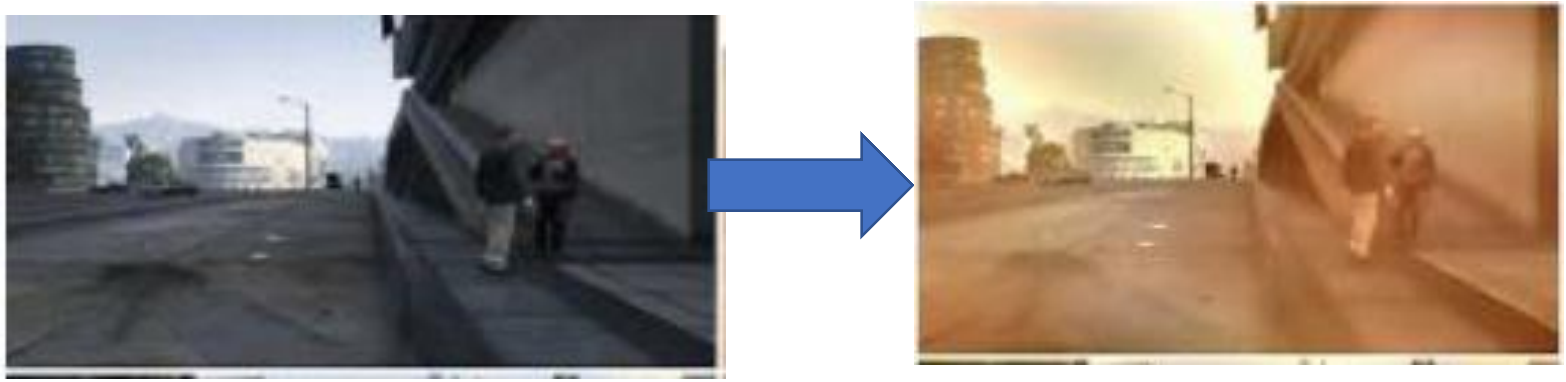
- Using equation :

$$x_{s_i} = \mathcal{T}_i(x_s) = \phi(f_y, f_i + \gamma_i f_s); \quad \gamma_i \in \mathbb{R}$$

- 5 augmentations picked.
- An augmentation $\mathcal{T}_i$ is picked, if $|\gamma_i| < 1.$ in the above equation
- Alteration in image statistics -> style gap between the two domains

Experimental Setup – Augmentation Groups

- Augmentation 1
  - **Aug-A**
  - **Fourier transform**

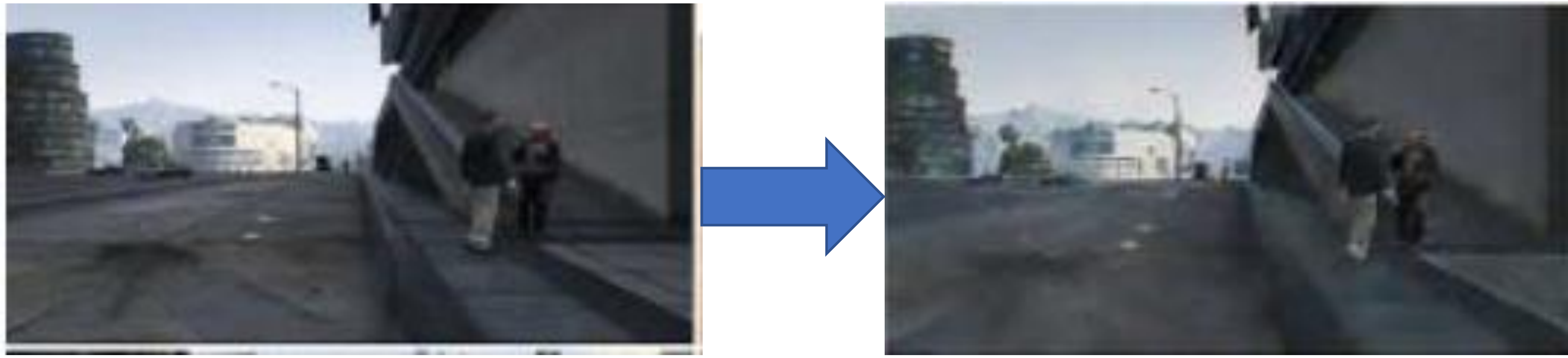


Experimental Setup – Augmentation Groups

- Augmentation 2
  - **Aug-B**
  - **Deep style transfer network for style randomization**

Experimental Setup – Augmentation Groups

- Augmentation 3
  - **Aug-C**
  - **AdaIN**



WEAKNESS

Domain shift : Non-intuitive

Experimental Setup – Augmentation Groups

- Augmentation 4
  - **Aug-D**
  - **Stylistic weather augmentations**

Experimental Setup – Augmentation Groups

- Augmentation 5
  - **Aug-E**
  - **Cartoon augmentation**

Results

Comparison with prior arts

Table 5. **Quantitative evaluation on GTA5→Cityscapes**. Performance on different segmentation architectures: A (DeepLabv2 ResNet-101), B (FCN8s VGG-16). SF indicates whether the method supports *source-free* adaptation. *Ours (V)* indicates use of our vendor-side `AG`s with prior art and * indicates reproduced by us using released code. We observe better or competitive performance on minority classes like motorcycle compared to non-source-free prior arts.

Method	Arch.	SF	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
PLCA [12]	A	×	84.0	30.4	82.4	35.3	24.8	32.2	36.8	24.5	85.5	37.2	78.6	66.9	32.8	85.5	40.4	48.0	8.8	29.8	41.8	47.7
CrCDA [8]	A	×	92.4	55.3	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	40.6	46.1	2.1	31.1	32.7	48.6
PIT [19]	A	×	87.5	43.4	78.8	31.2	30.2	36.3	39.9	**42.0**	79.2	37.1	79.3	65.4	37.5	83.2	46.0	45.6	25.7	23.5	49.9	50.6
TPLD [26]	A	×	94.2	**60.5**	82.8	36.6	16.6	39.3	29.0	25.5	85.6	44.9	84.4	60.6	27.4	84.1	37.0	47.0	**31.2**	36.1	50.3	51.2
RPT [35]	A	×	89.7	44.8	86.4	**44.2**	30.6	**41.4**	**51.7**	33.0	87.8	39.4	86.3	**65.6**	24.5	**89.0**	36.2	46.8	17.6	39.1	**58.3**	53.2
FADA [29]	A	×	91.0	50.6	86.0	43.4	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1
IAST [20]	A	×	94.1	58.8	85.4	39.7	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	50.3	24.7	35.2	40.2	52.2
Ours (V) + FADA*	A	×	91.2	51.0	**86.6**	43.6	30.3	37.1	43.7	25.2	**87.9**	40.2	88.2	64.7	**38.4**	85.5	32.0	46.8	6.6	25.9	37.5	50.6
Ours (V) + IAST*	A	×	**94.8**	59.4	86.2	40.5	29.5	25.5	43.8	34.7	85.9	34.9	89.5	63.4	30.8	88.3	42.6	50.7	25.3	35.7	40.9	52.8
URMA [28]	A	✓	92.3	55.2	81.6	30.8	18.8	37.1	17.7	12.1	84.2	35.9	83.8	57.7	24.1	81.7	27.5	44.3	6.9	24.1	40.4	45.1
SRDA* [1]	A	✓	90.5	47.1	82.8	32.8	28.0	29.9	35.9	34.8	83.3	39.7	76.1	57.3	23.6	79.5	30.7	40.2	0.0	26.6	30.9	45.8
Ours (w/o `cPAE`*)*	A	✓	90.9	48.6	85.5	35.3	31.7	36.9	34.7	34.8	86.2	47.8	88.5	61.7	32.6	85.9	46.9	50.4	0.0	38.9	52.4	51.6
Ours (w/ `cPAE`*)*	A	✓	91.7	53.4	86.1	37.6	**32.1**	37.4	38.2	35.6	86.7	**48.5**	**89.9**	62.6	34.3	87.2	**51.0**	50.8	4.2	**42.7**	53.9	**53.4**
BDL [15]	B	×	89.2	40.9	81.2	29.1	19.2	14.2	29.0	19.6	83.7	35.9	80.7	54.7	23.3	82.7	25.8	28.0	2.3	25.7	19.9	41.3
LTIR [13]	B	×	92.5	54.5	**83.9**	**34.5**	**25.5**	**31.0**	30.4	18.0	84.1	39.6	83.9	53.6	19.3	81.7	21.1	13.6	**17.7**	12.3	6.5	42.3
LDR [30]	B	×	90.1	41.2	82.2	30.3	21.3	18.3	33.5	23.0	84.1	37.5	81.4	54.2	24.3	83.0	**27.6**	32.0	8.1	29.7	26.9	43.6
FADA [29]	B	×	92.3	51.1	83.7	33.1	29.1	28.5	28.0	21.0	82.6	32.6	85.3	55.2	28.8	83.5	24.4	37.4	0.0	21.1	15.2	43.8
PCEDA [32]	B	×	90.2	44.7	82.0	28.4	28.4	24.4	33.7	**35.6**	83.7	**40.5**	75.1	54.4	28.2	80.3	23.8	39.4	0.0	22.8	30.8	44.6
SFDA [17]	B	✓	81.8	35.4	82.3	21.6	20.2	25.3	17.8	4.7	80.7	24.6	80.4	50.5	9.2	78.4	26.3	19.8	11.1	6.7	4.3	35.8
Ours (w/o `cPAE`*)*	B	✓	90.1	44.2	81.7	31.6	19.2	27.5	29.6	26.4	81.3	34.7	82.6	52.5	24.9	83.2	25.3	**41.9**	8.6	15.7	32.2	43.4
Ours (w/ `cPAE`*)*	B	✓	**92.9**	**56.9**	82.5	20.4	6.0	**30.8**	**34.7**	33.2	**84.6**	17.0	**88.9**	**62.3**	**30.7**	**85.1**	15.3	40.6	10.2	**30.1**	**50.4**	**45.9**

Comparison with prior arts

Table 6. **Quantitative evaluation on SYNTHIA→Cityscapes**. Performance on different segmentation architectures: A (DeepLabv2 ResNet-101), B (FCN8s VGG-16). mIoU and mIoU* are averaged over 16 and 13 categories respectively. SF indicates whether the method supports *source-free* adaptation.

Method	Arch.	SF	road	sidewalk	building	wall*	fence*	pole*	t-light	t-sign	vegetation	sky	person	rider	car	bus	motorcycle	bicycle	mIoU	mIoU*
CAG [34]	A	×	84.8	41.7	**85.5**	-	-	-	13.7	23.0	**86.5**	78.1	**66.3**	28.1	81.8	21.8	22.9	49.0	-	52.6
APODA [31]	A	×	86.4	41.3	79.3	-	-	-	22.6	17.3	80.3	81.6	56.9	21.0	84.1	49.1	24.6	45.7	-	53.1
PyCDA [16]	A	×	75.5	30.9	83.3	**20.8**	0.7	32.7	27.3	**33.5**	84.7	85.0	64.1	25.4	85.0	45.2	21.2	32.0	46.7	53.3
TPLD [26]	A	×	80.9	44.3	82.2	19.9	0.3	40.6	20.5	30.1	77.2	80.9	60.6	25.5	84.8	41.1	24.7	43.7	47.3	53.5
USAMR [37]	A	×	83.1	38.2	81.7	9.3	1.0	35.1	30.3	19.9	82.0	80.1	62.8	21.1	84.4	37.8	24.5	53.3	46.5	53.8
RPL [36]	A	×	87.6	41.9	83.1	14.7	1.7	36.2	31.3	19.9	81.6	80.6	63.0	21.8	86.2	40.7	23.6	53.1	47.9	54.9
IAST [20]	A	×	81.9	41.5	83.3	17.7	**4.6**	32.3	30.9	28.8	83.4	85.0	65.5	30.8	86.5	38.2	33.1	52.7	49.8	57.0
RPT [35]	A	×	89.1	47.3	84.6	14.5	0.4	**39.4**	**39.9**	30.3	86.1	86.3	60.8	25.7	**88.7**	49.0	28.4	**57.5**	51.7	59.5
URMA [28]	A	✓	59.3	24.6	77.0	14.0	1.8	31.5	18.3	32.0	83.1	80.4	46.3	17.8	76.7	17.0	18.5	34.6	39.6	45.0
Ours (w/o cPAE)	A	✓	89.0	44.6	80.1	7.8	0.7	34.4	22.0	22.9	82.0	86.5	65.4	33.2	84.8	45.8	38.4	31.7	48.1	55.5
Ours (w/ cPAE)	A	✓	**90.5**	**50.0**	81.6	13.3	2.8	34.7	25.7	33.1	83.8	**89.2**	66.0	**34.9**	85.3	**53.4**	**46.1**	46.6	**52.0**	**60.1**
PyCDA [16]	B	×	80.6	26.6	74.5	2.0	0.1	18.1	13.7	14.2	80.8	71.0	48.0	19.0	72.3	22.5	12.1	18.1	35.9	42.6
SD [6]	B	×	87.1	36.5	79.7	-	-	-	13.5	7.8	81.2	76.7	50.1	12.7	**78.0**	**35.0**	4.6	1.6	-	43.4
FADA [29]	B	×	80.4	35.9	**80.9**	2.5	0.3	**30.4**	7.9	22.3	81.8	83.6	48.9	16.8	77.7	31.1	13.5	17.9	39.5	46.0
BDL [15]	B	×	72.0	30.3	74.5	0.1	0.3	24.6	10.2	25.2	80.5	80.0	54.7	23.2	72.7	24.0	7.5	44.9	39.0	46.1
PCEDA [32]	B	×	79.7	35.2	78.7	1.4	0.6	23.1	10.0	**28.9**	79.6	81.2	51.2	**25.1**	72.2	24.1	**16.7**	**50.4**	41.1	48.7
Ours (w/o cPAE)	B	✓	88.5	45.4	79.8	2.8	2.2	27.4	18.4	25.4	82.4	83.6	55.9	12.1	72.8	25.6	3.5	12.9	40.0	46.7
Ours (w/ cPAE)	B	✓	**89.9**	**48.8**	**80.9**	**2.9**	**2.5**	28.1	**19.5**	26.2	**83.7**	**84.9**	**57.4**	17.8	75.6	28.9	4.3	17.2	**41.3**	**48.9**

Results - Ablations

Table 2. Ablation study for GTA5→Cityscapes. * indicates 3 rounds of self-training after the mentioned method. The client-side ablations begin from the best vendor-side model.

	Method	mIoU
Vendor-side	Standard single-source*	44.4
	Multi-source ERM*	47.6
	Domain-experts++ (DE++)*	48.0
	Leave-one-out++ (LO++)*	**51.6**
Client-side	w/o `cPAE`	51.6
	+ Inference via `cPAE`	52.5
	w/ `cPAE`	53.4
	+ Inference via `cPAE`	**54.2**

Results - Ablations

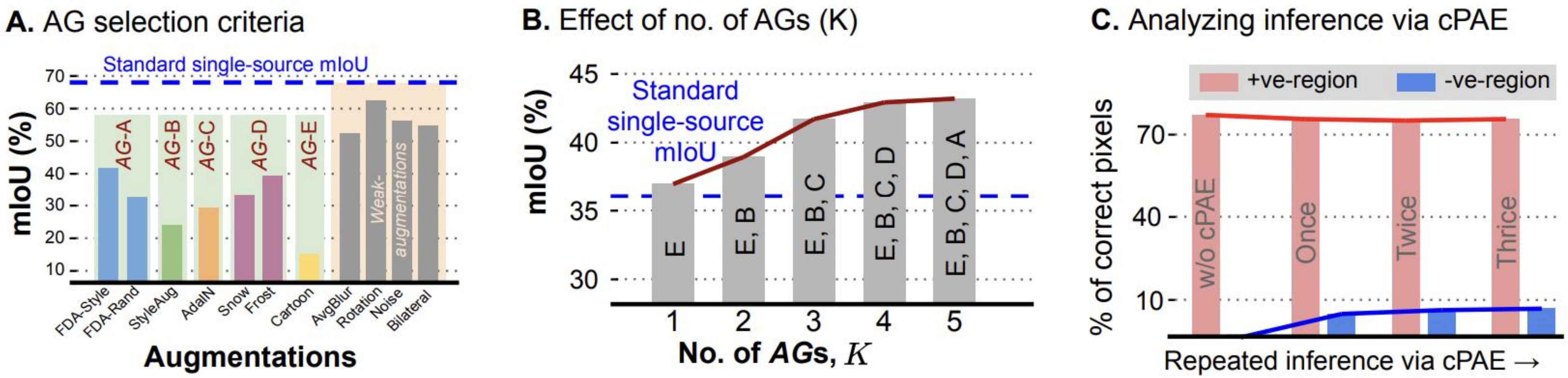


Figure 1. **A.** Selecting AGs from a set of candidate augmentations via inference through a standard single-source trained model (see Sec. 4.1). **B.** Performance of vendor-side trained models varying K on Cityscapes. Performance saturates as K reaches 5 (see Sec. 4.1). **C.** Impact of `cPAE` on correctly (+ve) and incorrectly (-ve) predicted regions on Cityscapes for a given model (see Sec. 4.3).

Results – Cross Dataset generalization

Table 5. Evaluating generalization and compatibility to online adaptation for GTA5→Cityscapes models on Foggy-Cityscapes and NTHU-Cross-City datasets. 0.005, 0.01, and 0.02 indicate the levels of fog in the dataset and GT indicates ground truth segmentation maps. * indicates experiment reproduced by us using the released code of prior arts. We also show standard Cityscapes results for reference.

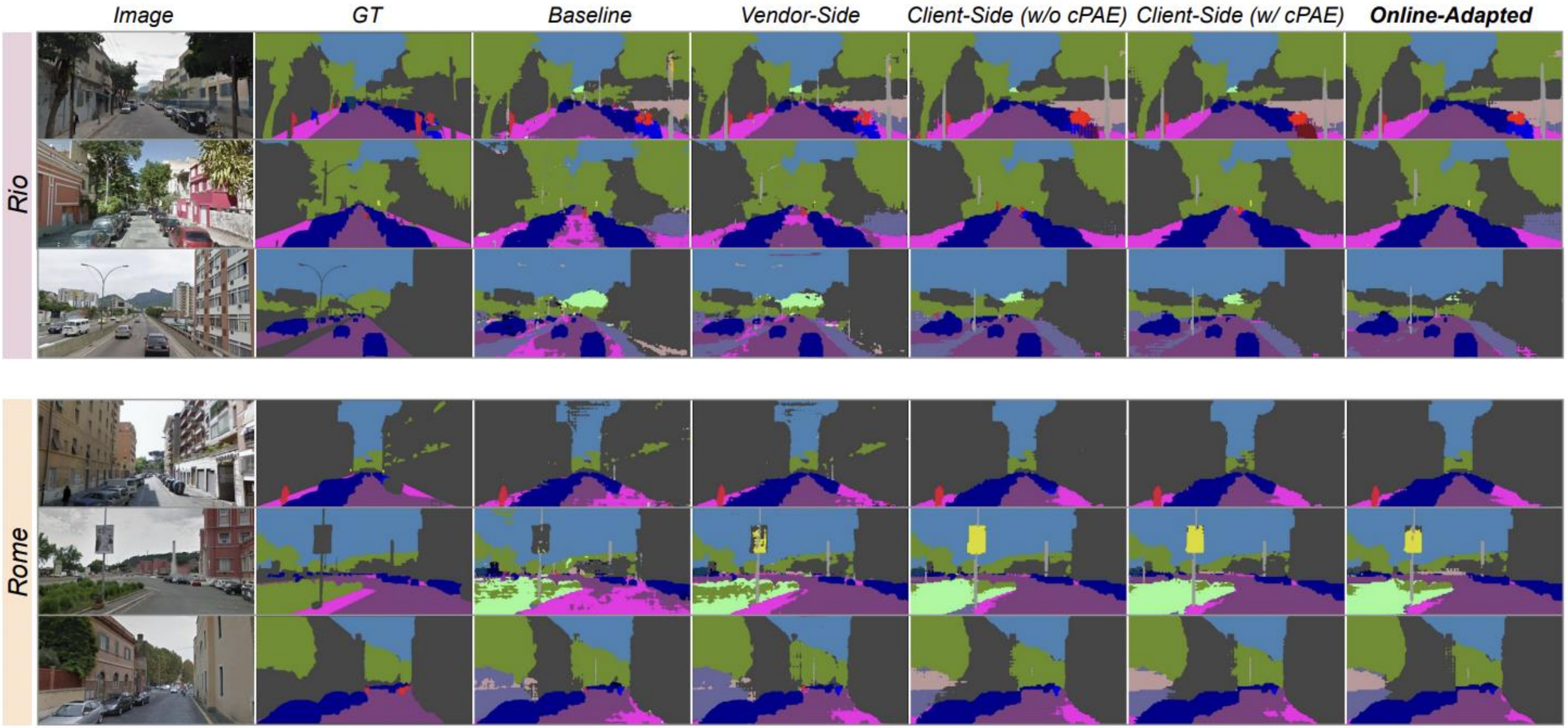
	#	Method	Access to GTA5 | Citysc.	Cityscapes		Foggy-Cityscapes (19-class)				NTHU-Cross-City (13-class)				
				19-class	13-class	0.005	0.01	0.02	Avg.	Rio	Rome	Taipei	Tokyo	Avg.
Vendor-side (GTA5)	1.	BDL (w/o ST) [41]	✓| ✓(no GT)	43.3	53.2	40.4	36.8	30.3	35.8	38.9	42.2	42.2	41.2	41.1
	2.	FDA* (w/o ST) [83]	✓| ✓(no GT)	42.7	51.9	42.1	40.3	35.3	39.2	42.2	42.3	37.5	42.3	41.0
	3.	*Ours (vendor-side)*	✓| ×	43.1	51.5	43.6	42.4	38.3	**41.4**	47.0	48.7	43.4	44.5	**45.9**
Client-side (→Citysc.)	4.	ASN [68]	✓| ✓(no GT)	42.4	51.1	41.0	38.0	31.7	36.9	41.8	44.5	37.5	41.9	41.4
	5.	MSL [5]	✓| ✓(no GT)	46.4	54.5	44.3	40.9	34.2	39.8	44.4	47.0	45.6	44.7	45.4
	6.	BDL [41]	✓| ✓(no GT)	48.5	57.7	46.0	42.6	36.3	41.6	44.1	47.1	47.5	44.3	45.7
	7.	FDA [83]	✓| ✓(no GT)	48.8	57.8	47.6	45.2	39.1	44.0	47.8	46.6	42.7	48.1	46.3
	8.	*Ours (client-side)*	× | ✓(no GT)	53.4	61.4	51.7	48.9	42.3	**47.6**	47.1	47.7	45.7	46.5	**46.7**
Online Adapt. (→FoggyC / →NTHU)	9.	CBST [95]	× | ✓(w/ GT)	-	-	-	-	-	-	52.2	53.6	50.3	48.8	51.2
	10.	MSL [5]	× | ✓(w/ GT)	-	-	-	-	-	-	53.3	54.5	50.6	50.5	52.2
	11.	CSCL [13]	× | ✓(w/ GT)	-	-	-	-	-	-	53.8	54.8	51.4	51.0	52.7
	12.	*Ours (client-side)*	× | ×	-	-	53.6	51.1	45.9	50.2	54.3	55.0	51.6	51.3	**53.0**

Results – Analysis

Table 3. Empirical evaluation of Result 1 for vendor-side SoMAN heads with mIoU for various target scenarios. LO indicates leave-one-out head while ERM is the global head. 0.005, 0.01, and 0.02 indicate the levels of fog in the dataset. We observe that different heads are optimal for different target domains.

Head	Cityscapes	Foggy-Cityscapes			NTHU-Cross-City			
		0.005	0.01	0.02	Rio	Rome	Taipei	Tokyo
ERM	43.1	43.6	42.4	**38.3**	47.0	48.7	43.4	44.5
LO-A	42.4	43.0	41.6	36.7	45.4	**48.9**	43.2	45.4
LO-B	42.2	42.2	40.7	36.1	**49.0**	47.7	42.1	46.5
LO-C	43.1	43.0	41.7	37.8	48.1	48.6	43.8	**46.7**
LO-D	**43.5**	43.4	41.7	37.0	45.6	47.9	**43.9**	45.3
LO-E	43.2	**43.9**	**42.6**	37.9	45.5	47.0	43.6	45.9

Qualitative Results



Qualitative Results

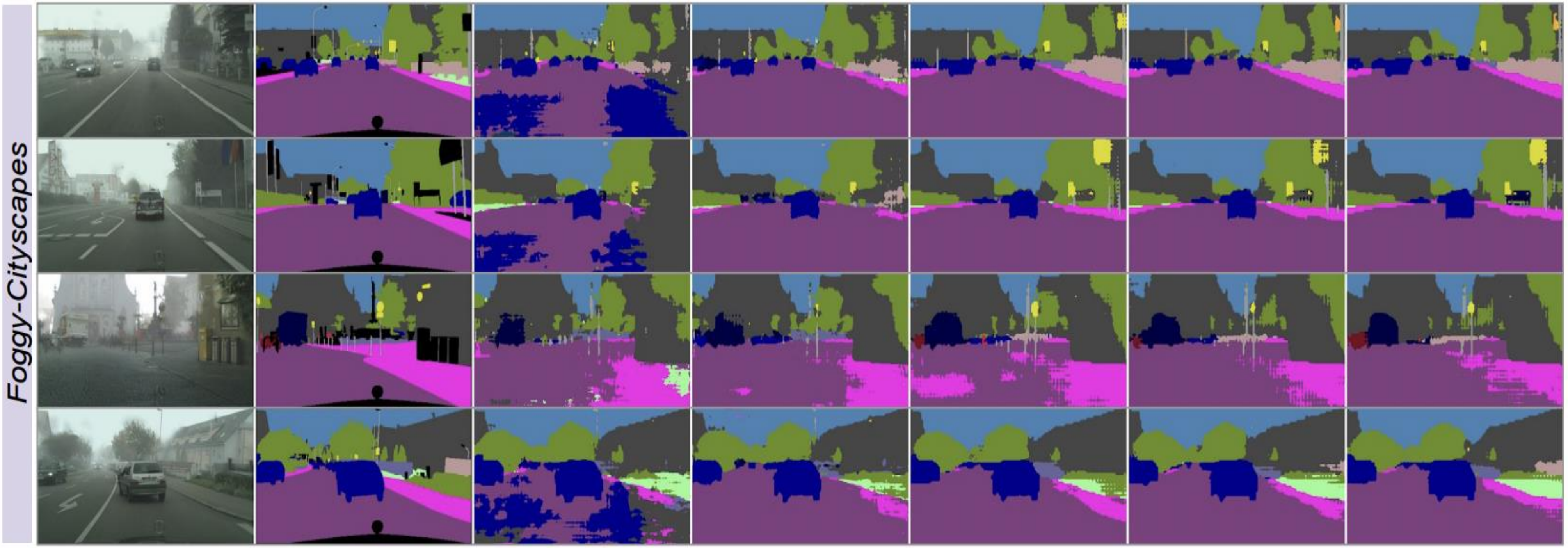


Figure 4. Qualitative evaluation of GTA5→Cityscapes and online adapted models on NTHU-Cross-City and Foggy-Cityscapes datasets. The performance generally improves from vendor-side to client-side to online-adapted model. *Best viewed in color.*

Qualitative Results

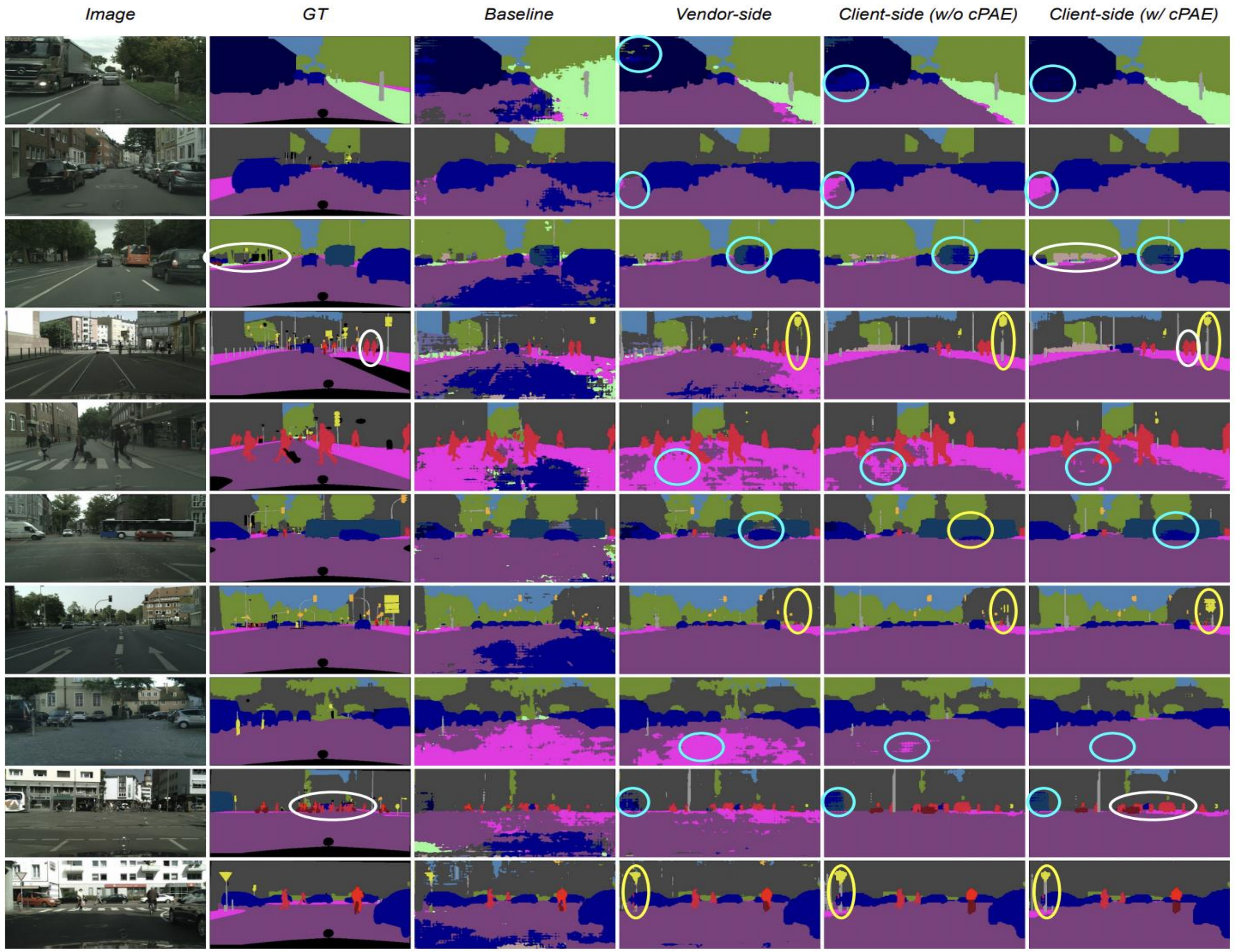


Figure 5. Qualitative evaluation of the proposed approach. Vendor-side model generalizes better than baseline but performs worse than client-side due to the domain gap. Inculcating prior knowledge from cPAE structurally regularizes the predictions and overcomes merged-region (yellow circle) and splitted-region (blue circle) problems. Some failure cases are also shown (white circle). *Best viewed in color.*

Strengths, Weaknesses, and Interesting Ideas

Strengths

- The only method (that we are aware of) that does adaptation friendly training at the vendor end, making it scalable

Strengths

- The only method (that we are aware of) that does adaptation friendly training at the vendor end, making it scalable
- Relevant problem statement with real-world applications

Strengths

- The only method (that we are aware of) that does adaptation friendly training at the vendor end, making it scalable
- Relevant problem statement with real-world applications
- Performs better than other competing **non-source-free methods**

Strengths

- The only method (that we are aware of) that does adaptation friendly training at the vendor end, making it scalable
- Relevant problem statement with real-world applications
- Performs better than other competing **non-source-free methods**
- Fail-cases (merged and split regions) properly identified and addressed

Strengths

- The only method (that we are aware of) that does adaptation friendly training at the vendor end, making it scalable
- Relevant problem statement with real-world applications
- Performs better than other competing **non-source-free methods**
- Fail-cases (merged and split regions) properly identified and addressed
- The tradeoff between "vendor-side knowledge" and "label-noise + info redundancy" in pseudo-labels is well-handled

Strengths

- The only method (that we are aware of) that does adaptation friendly training at the vendor end, making it scalable
- Relevant problem statement with real-world applications
- Performs better than other competing **non-source-free methods**
- Fail-cases (merged and split regions) properly identified and addressed
- The tradeoff between "vendor-side knowledge" and "label-noise + info redundancy" in pseudo-labels is well-handled
- Notation table and pseudo-code for the training steps clarifies the implementation details

Weaknesses

- The reasoning behind why LO++ outperforms DE++ is not justifiable from a diagram.
- The "Result 1" in the paper is intuition-based as well as without any reference.
- The exclusion of DE++ heads from the SoMAN network is wrongly justified under computational overhead.
- There is no discussion on error accumulation due to self-training
- Although code is available, the vendor-side implementations are missing (trained models are provided).

Weaknesses

- It is not an easy-to-understand paper, a lot of new terminology is introduced when it could have been done without
  - CPAE – It's a denoising encoder, the section that explains CPAE is overly complicated
  - ERM – overcomplicates the paper
- Nit – All of us were confused on what prior **arts** were

Interesting Ideas

- Generation of AGs to avoid the requirement of multi-domain labeled data on the vendor side
- The ability to tailor source/vendor training to support downstream domain adaptation is pretty interesting

Questions?