Learning to Detect and Segment Objects across Domains

Ming-Hsuan Yang

UC Merced · Google

Adaptive Vision Tasks

- Detection
  - SNoW-based face detector [NIPS99]
  - Weakly-supervised object localization with progressive domain adaption [CVPR16]
  - Every pixel matters: center-aware feature alignment for domain adaptive object detector [ECCV20]
- Tracking
  - Incremental visual tracking [NIPS04]
  - Multiple instance tracking [CVPR09]
  - Online tracking benchmark [CVPR13]
  - Tracking persons-of-interest via adaptive discriminative features [ECCV16]
- Recognition
  - Domain adaption for face recognition in unlabeled videos [ICCV17]
  - Cross-domain few-shot classification [ICLR20]
  - Generalized convolutional forest networks for domain generalization and visual recognition [ICLR20]
  - Long-tailed visual recognition from a domain adaptation perspective [CVPR20]
- Segmentation
  - Learning adaptive structured output space for semantic segmentation [CVPR18]
  - Adversarial learning for semi-supervised semantic segmentation [BMVC18]
  - Pixel-level domain transfer with cross-domain consistency [CVPR19]

Learning to Adapt Structured Output Space for Semantic Segmentation

CVPR 2018

Yi-Hsuan Tsai   Wei-Chih Hung   Samuel Schulter   Kihyuk Sohn   Ming-Hsuan Yang   Manmohan Chandraker

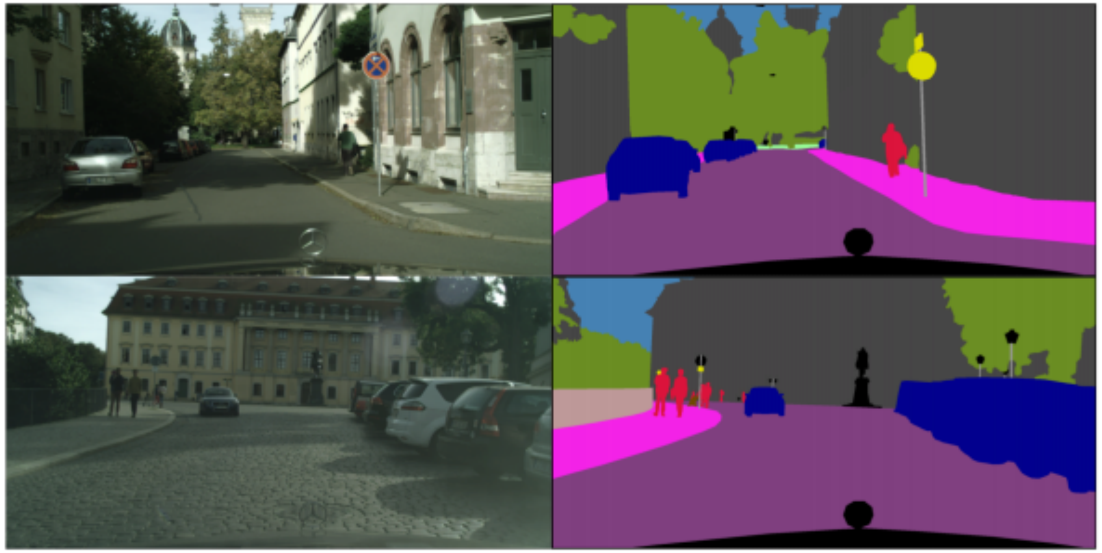
NEC Laboratories America
Relentless passion for innovation

University of California
UC Merced

Google

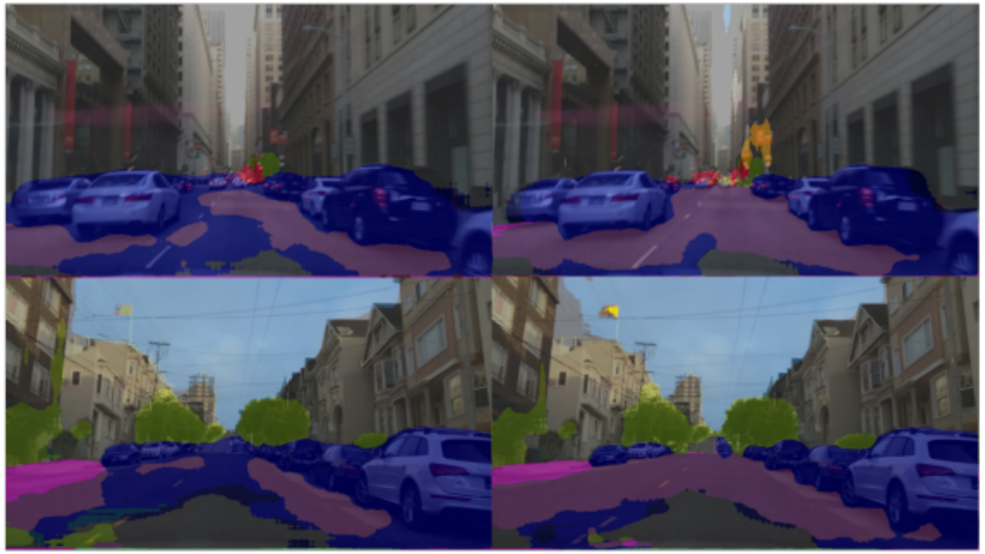
Domain Adaption

Semantic segmentation



Source domain: lots of **labeled** data

Target domain: lots of **unlabeled** data



Before Adaptation **After Adaptation**

[Hoffman, et al., arXiv 2016]

Examples

- City A -> City B
- Synthetic (source) -> Real (target)

Synthetic v.s. Real

GTA5

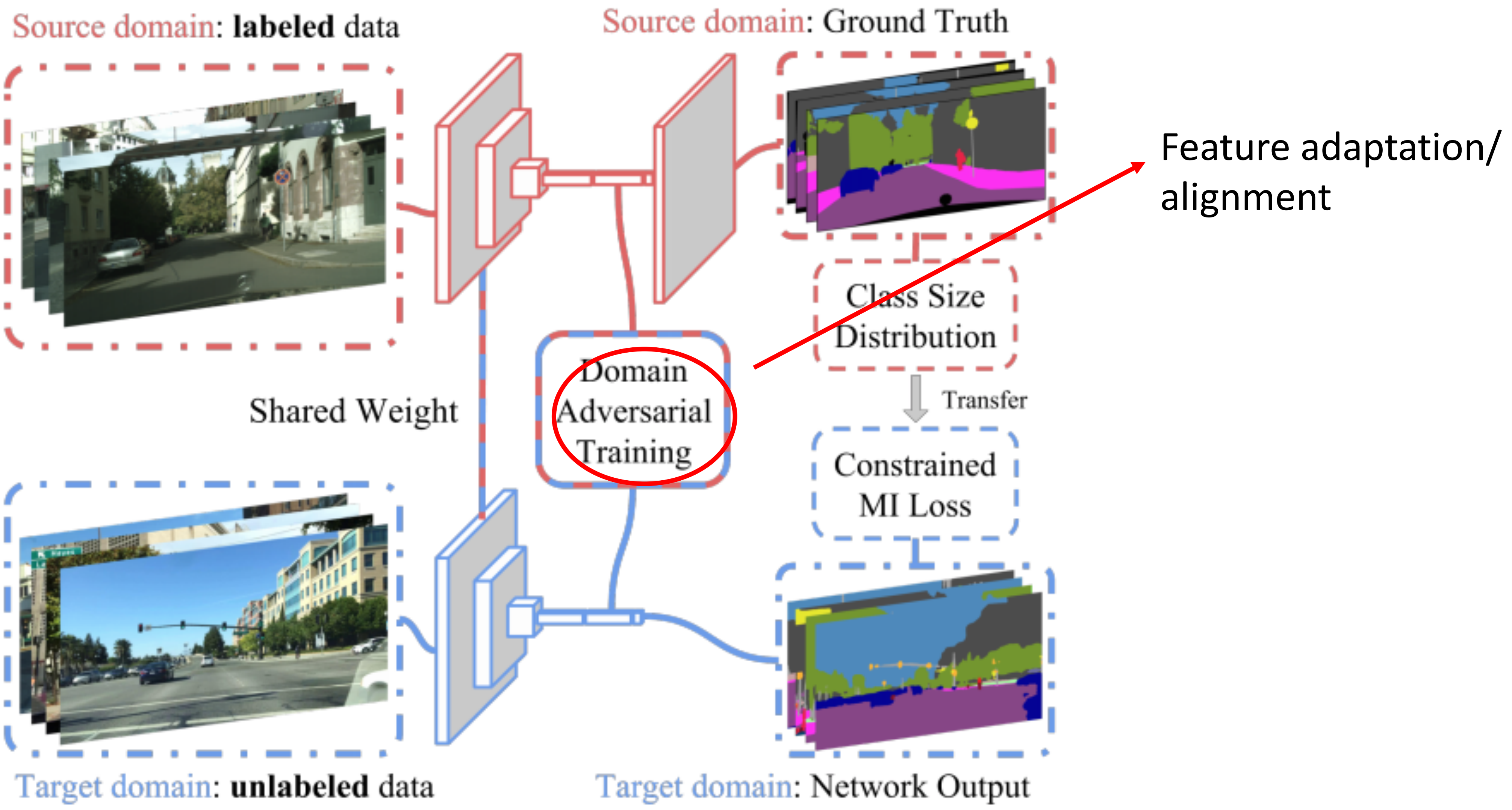
[Richter, et al., ECCV 2016]

Cityscapes

[Cordts, et al., CVPR 2016]

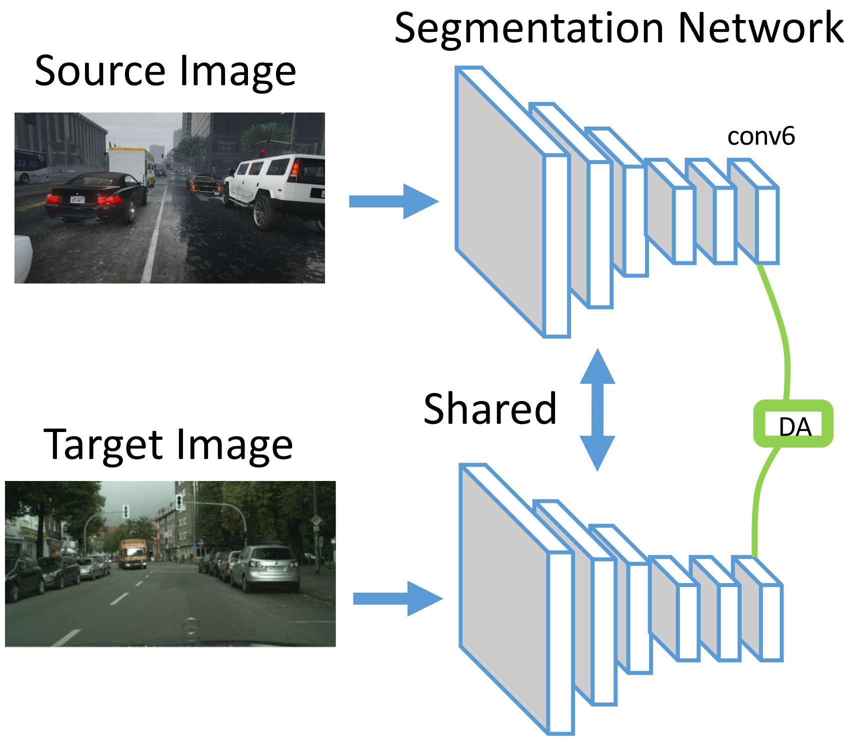
Data augmentation: rendered images by graphics engines or translation methods

Adversarial Domain Adaptation



Is feature adaptation the best choice for structured output?

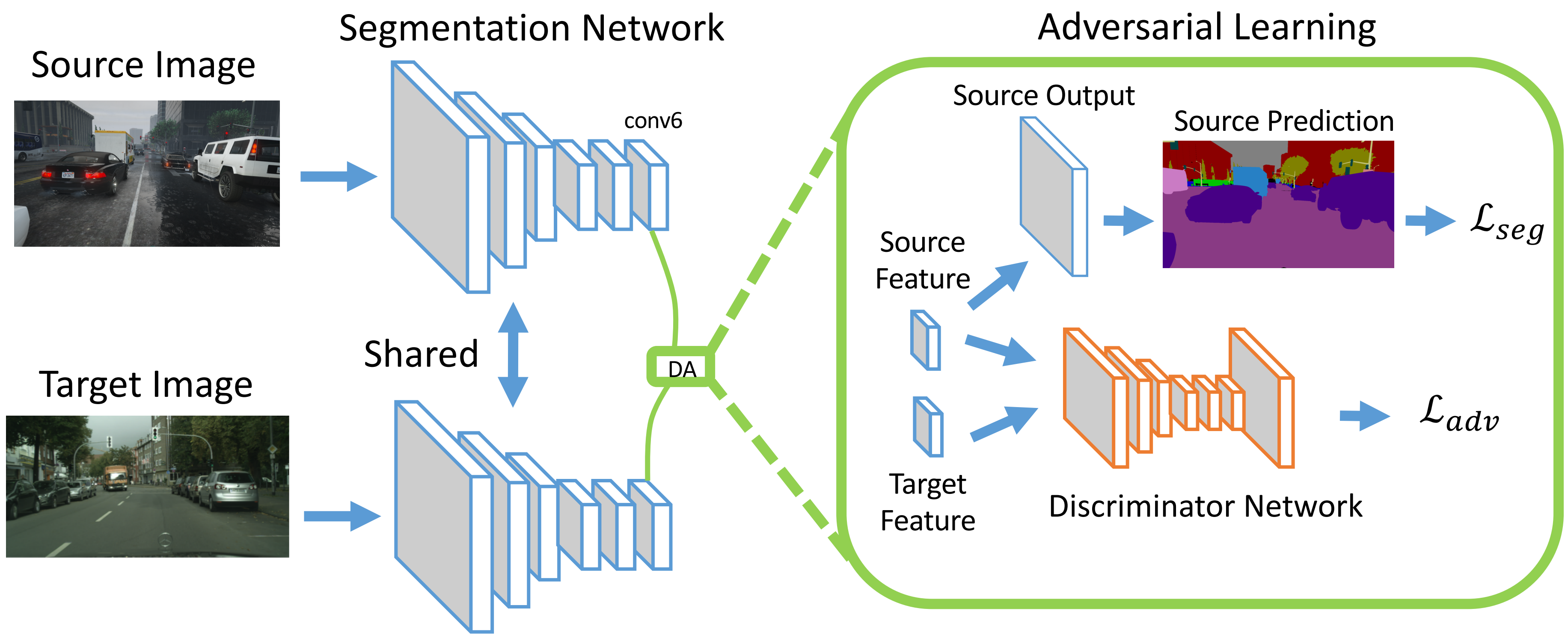
Feature Space Adaptation



Goal: align features between two domains

Feature dimensions: 1024, 2048, 4096, …

Feature Space Adaptation



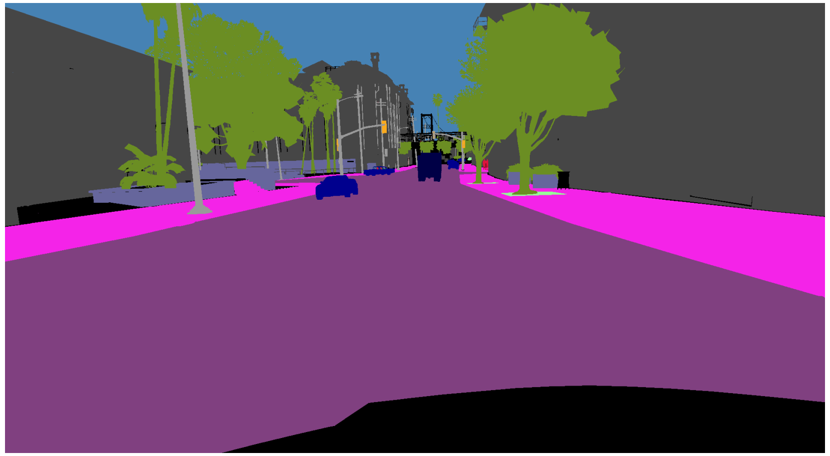
Is feature adaptation effective for semantic segmentation?

Motivation

Source Domain

Target Domain

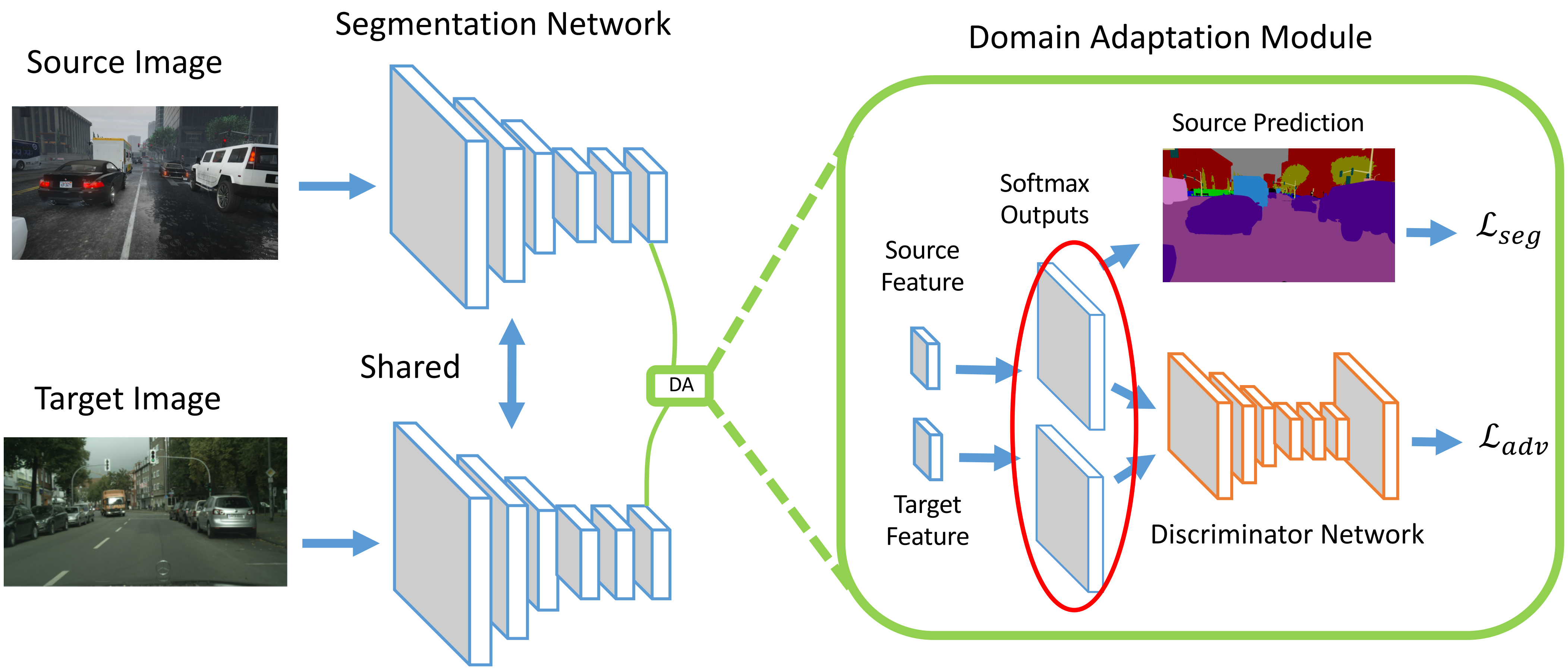
Large gap in appearance



Small gap in spatial layout

- Semantic segmentations from the source and target domains should be similar
- Consider semantic segmentation results as structured output

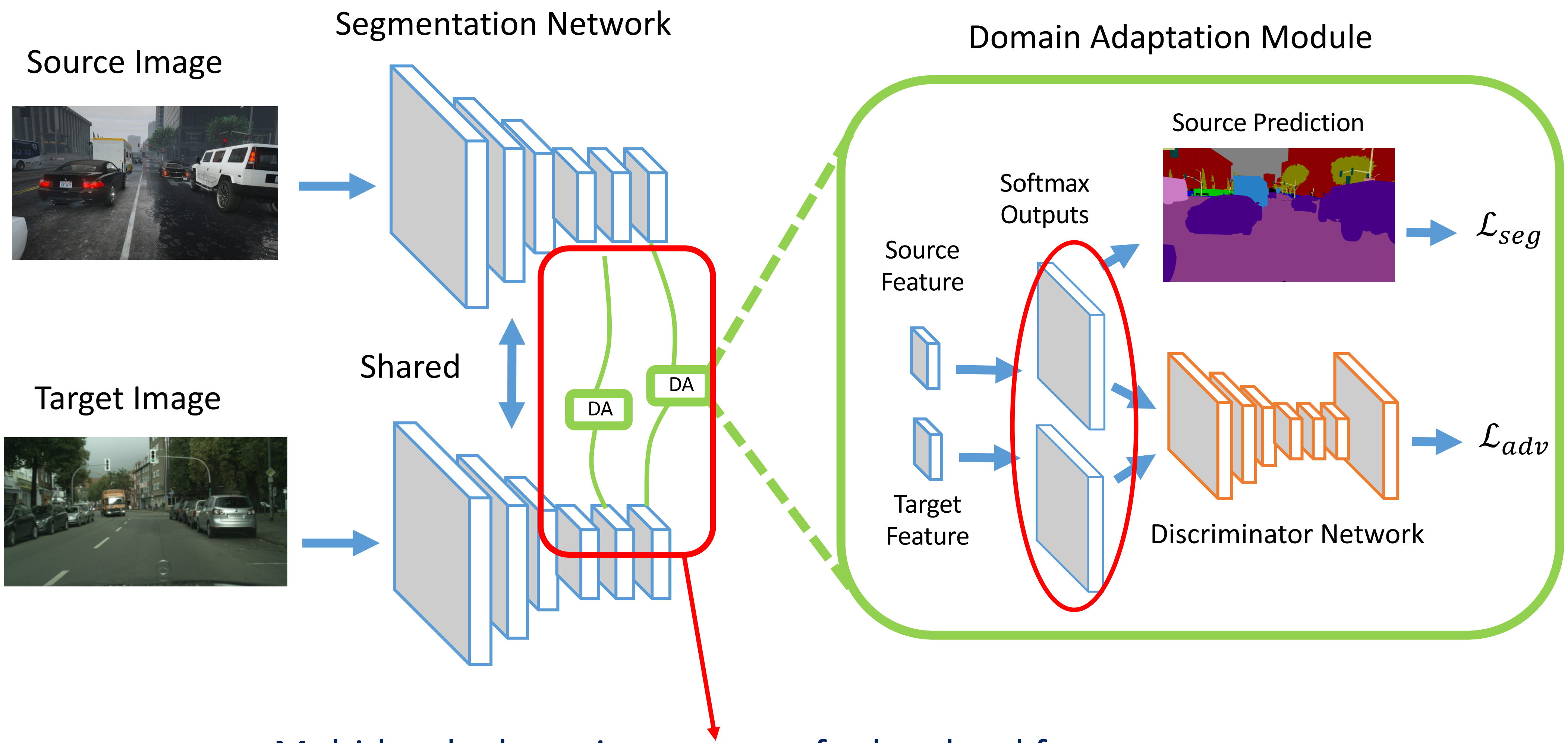
Our Method: Output Space Adaptation



Main difference: adversarial learning in the output space

Dimension of output space: 30 for Cityscapes

Our Method: Output Space Adaptation



Multi-level adaptation: account for low-level features

Multi-level Adversarial Learning

Segmentation Network (G) Training

$$\mathcal{L}(I_s, I_t) = \sum_i \lambda^i_{seg} \mathcal{L}^i_{seg}(I_s) + \sum_i \lambda^i_{adv} \mathcal{L}^i_{adv}(I_t)$$

Cross-entropy loss

Adversarial loss (only on target)

$$\mathcal{L}_{seg}(I_s) = -\sum_{h,w} \sum_{c \in C} Y_s^{(h,w,c)} \log(P_s^{(h,w,c)}),$$

$$\mathcal{L}_{adv}(I_t) = -\sum_{h,w} \log(\mathbf{D}(P_t)^{(h,w,1)})$$

$$P = G(I)$$

segmentation softmax output

Minimize loss for G

Maximize the probability of target predictions being considered as source ones

Discriminator (D) Training

Target

$$\mathcal{L}_d(P) = -\sum_{h,w} (1-z) \log(\mathbf{D}(P)^{(h,w,0)}) + z \log(\mathbf{D}(P)^{(h,w,1)}),$$

Source

Min-max objective

$$\max_{\mathbf{D}} \min_{\mathbf{G}} \mathcal{L}(I_s, I_t)$$

GTA5 (synthetic) -> Cityscapes (real)

Feature adaptation

Method	GTA5 → Cityscapes road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
FCNs in the Wild [13]	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
CDA [39]	74.9	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	**9.3**	55.2	18.8	18.9	0.0	**16.8**	**14.6**	28.9
CyCADA (feature) [12]	85.6	30.7	74.7	14.4	13.0	17.6	13.7	5.8	74.6	15.8	69.9	38.2	3.5	72.3	16.0	5.0	0.1	3.6	0.0	29.2
CyCADA (pixel) [12]	83.5	**38.3**	76.4	20.6	16.5	22.2	**26.2**	**21.9**	**80.4**	28.7	65.7	**49.4**	4.2	74.6	16.0	26.6	**2.0**	8.0	0.0	34.8
Ours (singel-level)	**87.3**	29.8	**78.6**	**21.1**	**18.2**	**22.5**	21.5	11.0	79.7	**29.6**	**71.3**	46.8	6.5	**80.1**	**23.0**	**26.9**	0.0	10.6	0.3	**35.0**

Output space adaptation

Image transform using CycleGAN

Baseline: VGG-16 -> why not use a stronger baseline?

GTA5 (synthetic) -> Cityscapes (real)

Without adaptation

	GTA5 → Cityscapes																			
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
Baseline (ResNet)	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	**36.0**	36.6
Ours (feature)	83.7	27.6	75.5	20.3	19.9	**27.4**	28.3	**27.4**	79.0	28.4	70.1	55.1	20.2	72.9	22.5	**35.7**	**8.3**	20.6	23.0	39.3
Ours (single-level)	**86.5**	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	**75.9**	57.3	26.2	**76.3**	29.8	32.1	7.2	29.5	32.5	41.4
Ours (multi-level)	**86.5**	**36.0**	**79.9**	**23.4**	**23.3**	23.9	**35.2**	14.8	**83.4**	**33.3**	75.6	**58.5**	**27.6**	73.7	**32.5**	35.4	3.9	**30.1**	28.1	**42.4**

Output space
adaptation

GTA5 (synthetic) -> Cityscapes (real)

Comparisons to upper-bounds (fully-supervised)?

GTA5 → Cityscapes				
method	Baseline	Adapt	Oracle	mIoU Gap
FCNs in the Wild [13]	VGG-16	27.1	64.6	-37.5
CDA [39]	VGG-16	28.9	60.3	-31.4
CyCADA (feature) [12]	VGG-16	29.2	60.3	-30.5
CyCADA (pixel) [12]	VGG-16	34.8	60.3	-24.9
Ours (single-level)	VGG-16	35.0	61.8	-25.2
Ours (multi-level)	ResNet-101	42.4	65.1	-22.7

Only differs a bit

GTA5 (synthetic) -> Cityscapes (real)

Comparisons to upper-bounds (fully-supervised)?

GTA5 → Cityscapes				
method	Baseline	Adapt	Oracle	mIoU Gap
FCNs in the Wild [13]	VGG-16	27.1	64.6	-37.5
CDA [39]	VGG-16	28.9	60.3	-31.4
CyCADA (feature) [12]	VGG-16	29.2	60.3	-30.5
CyCADA (pixel) [12]	VGG-16	34.8	60.3	-24.9
Ours (single-level)	VGG-16	35.0	61.8	-25.2
Ours (multi-level)	ResNet-101	42.4	65.1	-22.7

Varies a lot

GTA5 (synthetic) -> Cityscapes (real)

Training stability?

$$\mathcal{L}(I_s, I_t) = \mathcal{L}_{seg}(I_s) + \lambda_{adv}\mathcal{L}_{adv}(I_t)$$

	GTA5 → Cityscapes			
λ_{adv}	0.0005	0.001	0.002	0.004
Feature	35.3	39.3	35.9	32.8
Output Space	40.2	41.4	40.4	40.1

Varies a lot

GTA5 (synthetic) -> Cityscapes (real)

Training stability?

$$\mathcal{L}(I_s, I_t) = \mathcal{L}_{seg}(I_s) + \lambda_{adv}\mathcal{L}_{adv}(I_t)$$

	GTA5 $\rightarrow$ Cityscapes			
λ_{adv}	0.0005	0.001	0.002	0.004
Feature	35.3	39.3	35.9	32.8
Output Space	40.2	41.4	40.4	40.1

Only differs a bit

Synthia (synthetic) -> Cityscapes (real)

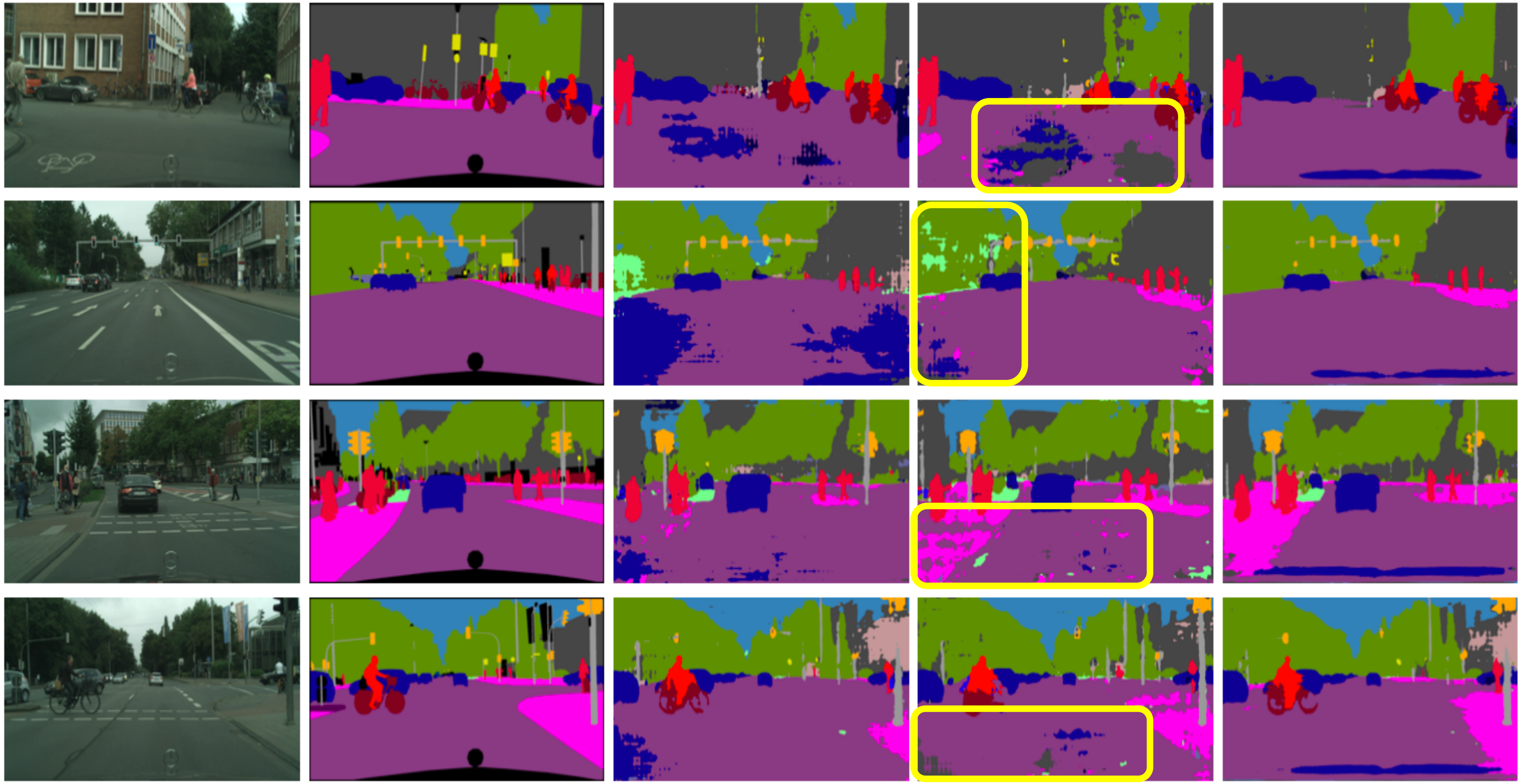
Feature adaptation

	SYNTHIA → Cityscapes													
Method	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
FCNs in the Wild [13]	11.5	19.6	30.8	0.1	**11.7**	42.3	68.7	**51.2**	3.8	54.0	3.2	0.2	0.6	22.9
CDA [39]	65.2	26.1	74.9	**3.7**	3.0	76.1	70.6	47.1	8.2	43.2	**20.7**	0.7	**13.1**	34.8
Cross-City [3]	62.7	25.6	**78.3**	1.2	5.4	**81.3**	**81.0**	37.4	6.4	63.5	16.1	1.2	4.6	35.7
Ours (single-level)	**78.9**	**29.2**	75.5	0.1	4.8	72.6	76.7	43.4	**8.8**	**71.1**	16.0	**3.6**	8.4	**37.6**
Baseline (ResNet)	55.6	23.8	74.6	6.1	**12.1**	74.8	79.0	**55.3**	19.1	39.6	23.3	13.7	25.0	38.6
Ours (feature)	62.4	21.9	76.3	**11.7**	11.4	75.3	80.9	53.7	18.5	59.7	13.7	20.6	24.0	40.8
Ours (single-level)	79.2	37.2	**78.8**	9.9	10.5	**78.2**	80.5	53.5	19.6	67.0	29.5	**21.6**	31.3	45.9
Ours (multi-level)	**84.3**	**42.7**	77.5	4.7	7.0	77.9	**82.5**	54.3	**21.0**	**72.3**	**32.2**	18.9	**32.3**	**46.7**

City A (real) -> City B (real)

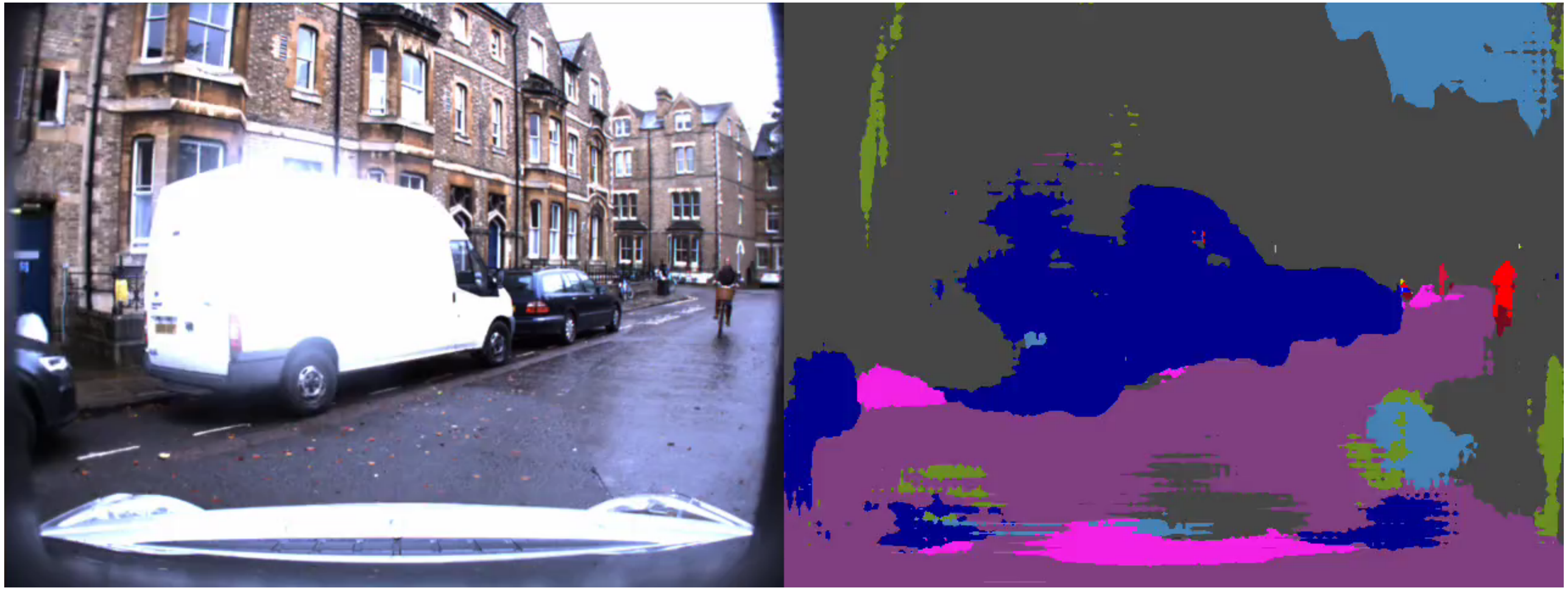
Cityscapes → Cross-City															
City	Method	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
Rome	Cross-City [3]	79.5	29.3	84.5	0.0	22.2	80.6	82.8	29.5	13.0	71.7	37.5	25.9	1.0	42.9
	Our Baseline	**83.9**	**34.3**	87.7	13.0	**41.9**	84.6	92.5	37.7	**22.4**	80.8	38.1	39.1	5.3	50.9
	Ours (feature)	78.8	28.6	85.5	16.6	40.1	85.3	79.6	42.4	20.7	79.6	**58.8**	45.5	6.1	51.4
	Ours (output space)	**83.9**	34.2	**88.3**	**18.8**	40.2	**86.2**	**93.1**	**47.8**	21.7	**80.9**	47.8	**48.3**	**8.6**	**53.8**
Rio	Cross-City [3]	74.2	43.9	79.0	2.4	7.5	77.8	69.5	39.3	10.3	67.9	**41.2**	27.9	10.9	42.5
	Our Baseline	**76.6**	**47.3**	82.5	**12.6**	22.5	77.9	86.5	43.0	19.8	**74.5**	36.8	29.4	16.7	48.2
	Ours (feature)	73.7	44.2	83.0	6.1	18.1	79.6	86.9	51.0	22.1	73.7	31.4	**48.3**	**28.4**	49.7
	Ours (output space)	76.2	44.7	**84.6**	9.3	**25.5**	**81.8**	**87.3**	**55.3**	**32.7**	74.3	28.9	43.0	27.6	**51.6**
Tokyo	Cross-City [3]	**83.4**	**35.4**	72.8	12.3	12.7	77.4	64.3	42.7	21.5	64.1	**20.8**	8.9	40.3	42.8
	Our Baseline	82.9	31.3	**78.7**	14.2	24.5	81.6	89.2	48.6	33.3	70.5	7.7	11.5	45.9	47.7
	Ours (feature)	81.5	30.8	76.6	15.3	20.2	82.0	84.0	49.4	33.3	70.5	4.5	24.3	**51.6**	48.0
	Ours (output space)	81.5	26.0	77.8	**17.8**	**26.8**	**82.7**	**90.9**	**55.8**	**38.0**	**72.1**	4.2	**24.5**	50.8	**49.9**
Taipei	Cross-City [3]	78.6	28.6	80.0	13.1	7.6	68.2	82.1	16.8	9.4	60.4	34.0	26.5	9.9	39.6
	Our Baseline	**83.5**	**33.4**	**86.6**	12.7	**16.4**	77.0	**92.1**	17.6	**13.7**	70.7	37.7	44.4	18.5	46.5
	Ours (feature)	82.1	31.9	84.1	25.7	13.2	**77.2**	81.2	28.1	12.0	67.0	35.8	43.5	20.9	46.6
	Ours (output space)	81.7	29.5	85.2	**26.4**	15.6	76.7	91.7	**31.0**	12.5	**71.5**	**41.1**	**47.3**	**27.7**	**49.1**

Qualitative Comparisons

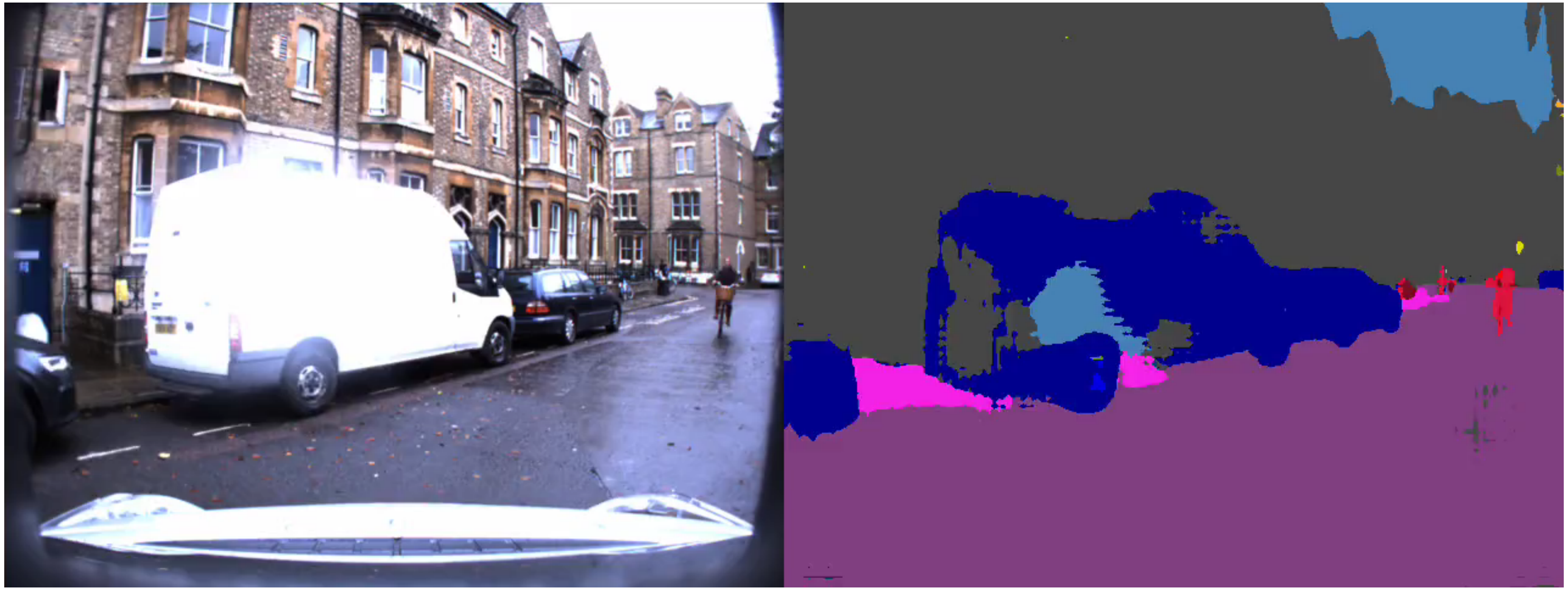


Target Image | Ground Truth | Before Adaptation | Feature Adaptation | Ours

Before adaption



Our results



Summary

- We propose a domain adaptation method for structured outputs (i.e., semantic segmentation)
  - Adversarial learning in the output space
  - Multi-level objective function
  - A strong baseline to shrink the domain gap

- Future goals: learn better feature representations
  - Different tasks? (e.g., optical flow, depth estimation)
  - Multi-tasks/domains?

- Code available at https://github.com/wasidennis/AdaptSegNet

Adversarial Learning for Semi-Supervised Semantic Segmentation

BMVC 2018

Wei-Chih Hung[1], Yi-Hsuan Tsai[2], Yan-Ting Liou[3,4], Yen-Yu Lin[4],Ming-Hsuan Yang[1,5]

[1]UC Merced [2]NEC Labs America [3]National Taiwan University

[4]Academia Sinica Taiwan [5]Google

Semi-supervised Semantic Segmentation

Small amount of labeled data

Large amount of unlabeled data

How do we exploit these data?

Motivation: Exploit Structured Context

Labeled Data

Image

Ground truth

Labeled/Unlabeled Data

Image

Model Prediction

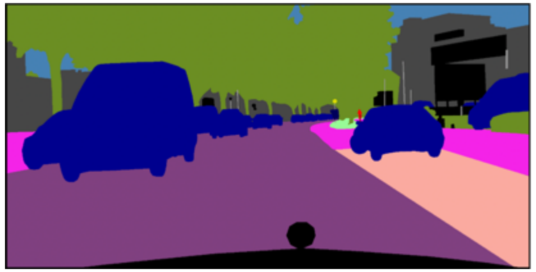
Can we push them to have similar **structure contexts**?

Apply adversarial learning to the **output space**.

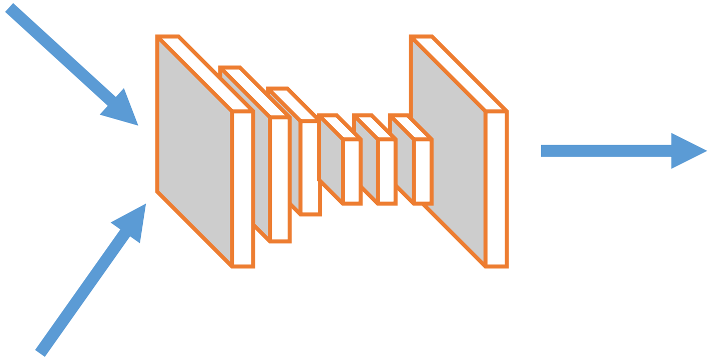
Adversarial Loss

Model Prediction

Ground truth



Discriminator Network



GT or Prediction?

$$\mathcal{L}_D = -\sum_{h,w} (1 - y_n) \log(1 - D(S(\mathbf{X}_n))^{(h,w)}) + y_n \log(D(\mathbf{Y}_n)^{(h,w)})$$

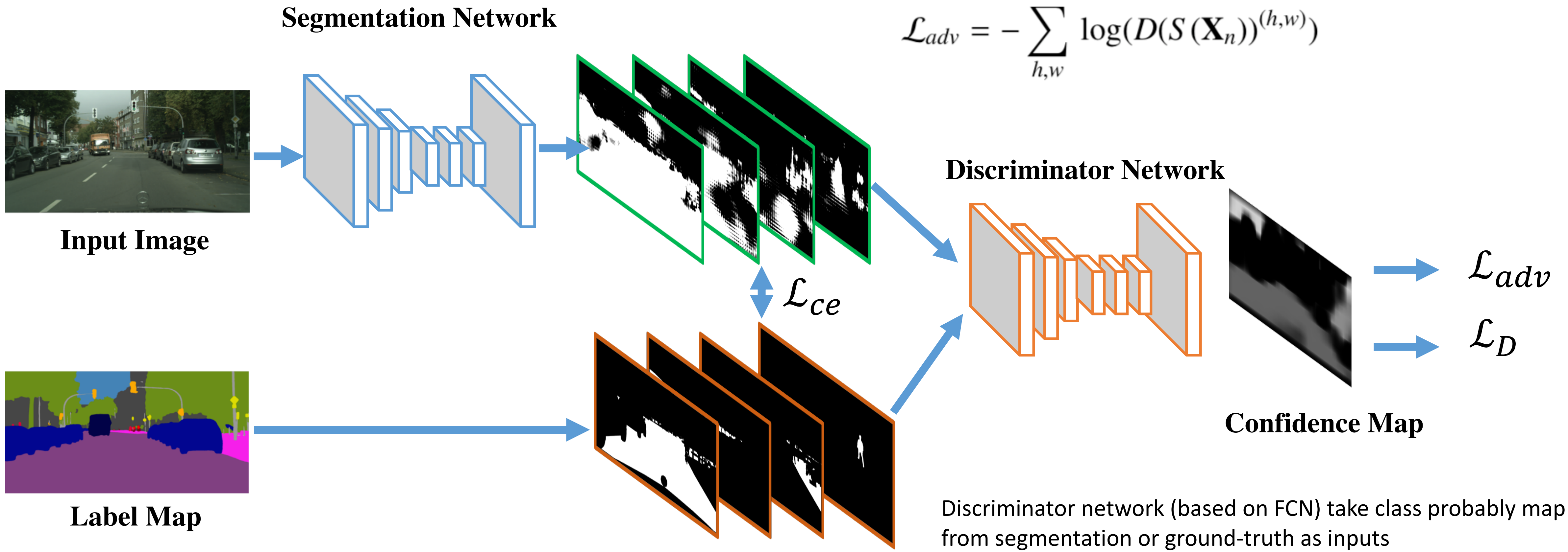
Adversarial (Inverse)

$$\mathcal{L}_{adv} = -\sum_{h,w} \log(D(S(\mathbf{X}_n))^{(h,w)})$$

Adversarial Loss: Fully Convolutional Discriminator

$$\mathcal{L}_D = -\sum_{h,w}(1-y_n)\log(1-D(S(\mathbf{X}_n))^{(h,w)}) + y_n\log(D(\mathbf{Y}_n)^{(h,w)})$$

$$\mathcal{L}_{adv} = -\sum_{h,w}\log(D(S(\mathbf{X}_n))^{(h,w)})$$



Discriminator network (based on FCN) take class probably map from segmentation or ground-truth as inputs

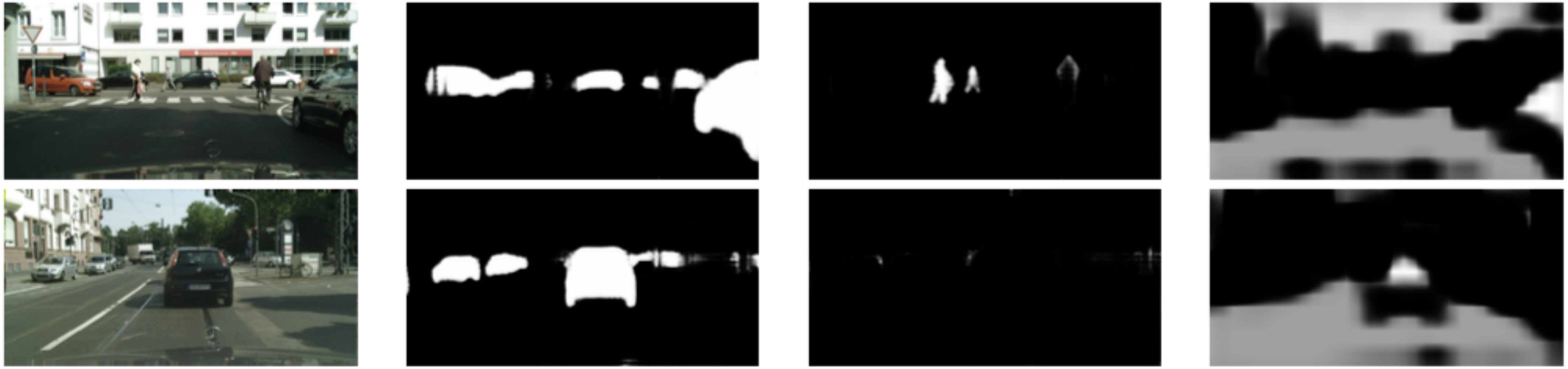
Semi-supervised Loss

- High confidence of being ground truth: trustworthy predictions
- Self-taught Learning: learn from high confidence areas

$$\mathcal{L}_{semi} = -\sum_{h,w}\sum_{c\in C} I(D(S(\mathbf{X}_n))^{(h,w)} > T_{semi}) \cdot \hat{\mathbf{Y}}_n^{(h,w,c)} \log(S(\mathbf{X}_n)^{(h,w,c)})$$

Threshold

Cross entropy with pseudo label



input image | “car” | “person” | confidence map

T_{semi} vs. Selected Prediction Accuracy

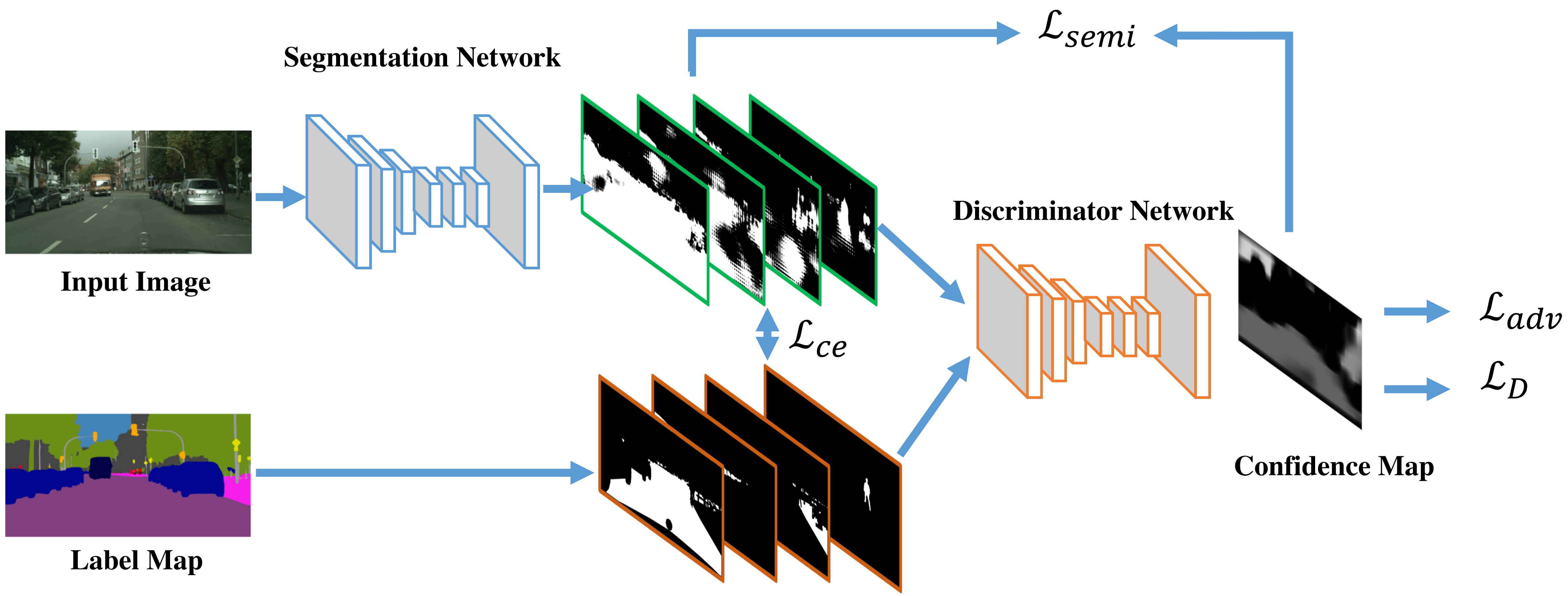
- Dataset: Cityscapes

Table 1: Selected pixel accuracy.

T_{semi}	Selected Pixels (%)	Accuracy
0	100%	92.65%
0.1	36%	99.84%
0.2	31%	99.91%
0.3	27%	99.94%

Proposed Framework

$$\mathcal{L}_{seg} = \mathcal{L}_{ce} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{semi}\mathcal{L}_{semi}$$



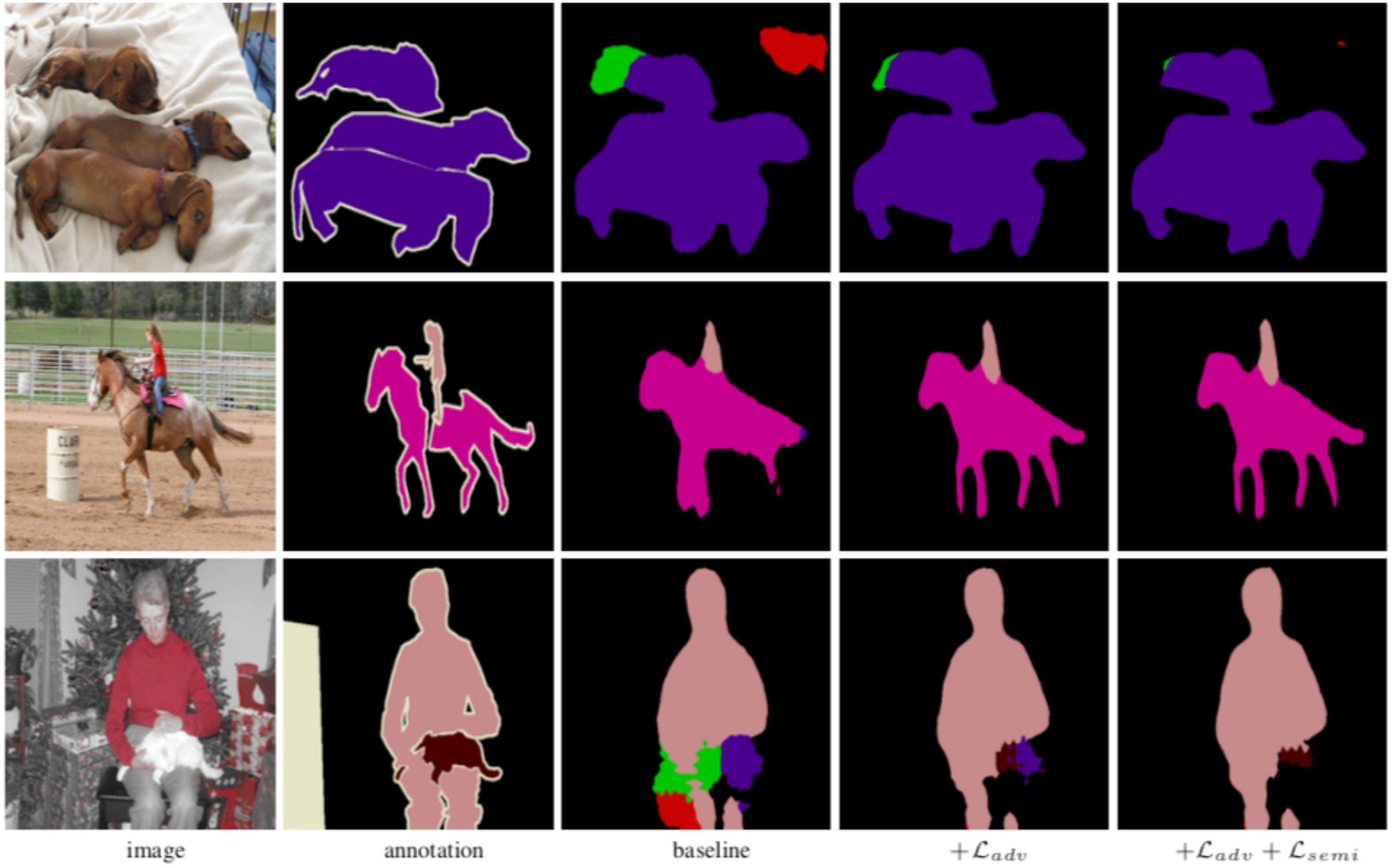
Results on PASCAL VOC 2012

Methods	Data Amount 1/8	1/4	1/2	Full
FCN-8s [46]	N/A	N/A	N/A	67.2
Dilation10 [77]	N/A	N/A	N/A	73.9
DeepLab-v2 [8]	N/A	N/A	N/A	77.7
our baseline	66.0	68.3	69.8	73.6
baseline + $\mathcal{L}_{adv}$	67.6	71.0	72.6	74.9
baseline + $\mathcal{L}_{adv}$ + $\mathcal{L}_{semi}$	68.8	71.6	73.2	N/A

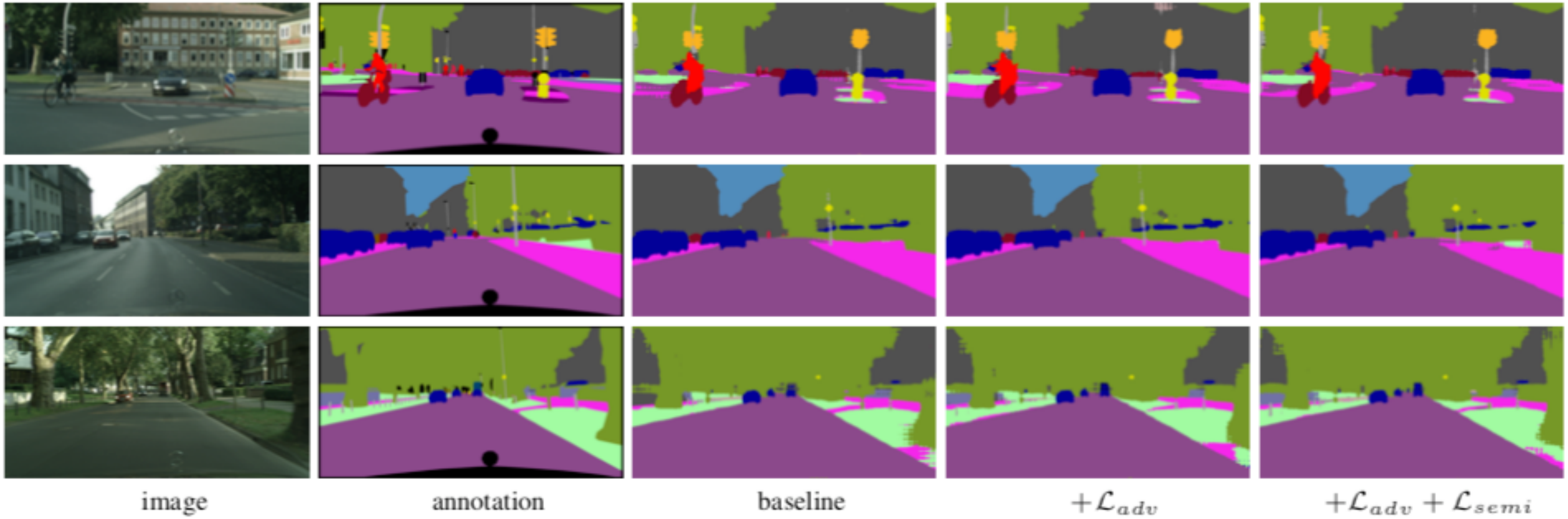
Results on Cityscapes

Methods	Data Amount 1/8	1/4	1/2	Full
FCN-8s [46]	N/A	N/A	N/A	65.3
Dilation10 [77]	N/A	N/A	N/A	67.1
DeepLab-v2 [8]	N/A	N/A	N/A	70.4
our baseline	52.4	58.3	62.6	66.4
baseline + $\mathcal{L}_{adv}$	53.8	59.1	63.7	67.7
baseline + $\mathcal{L}_{adv}$ + $\mathcal{L}_{semi}$	54.2	59.7	64.5	N/A

Qualitative Comparisons: PASCAL VOC 2012



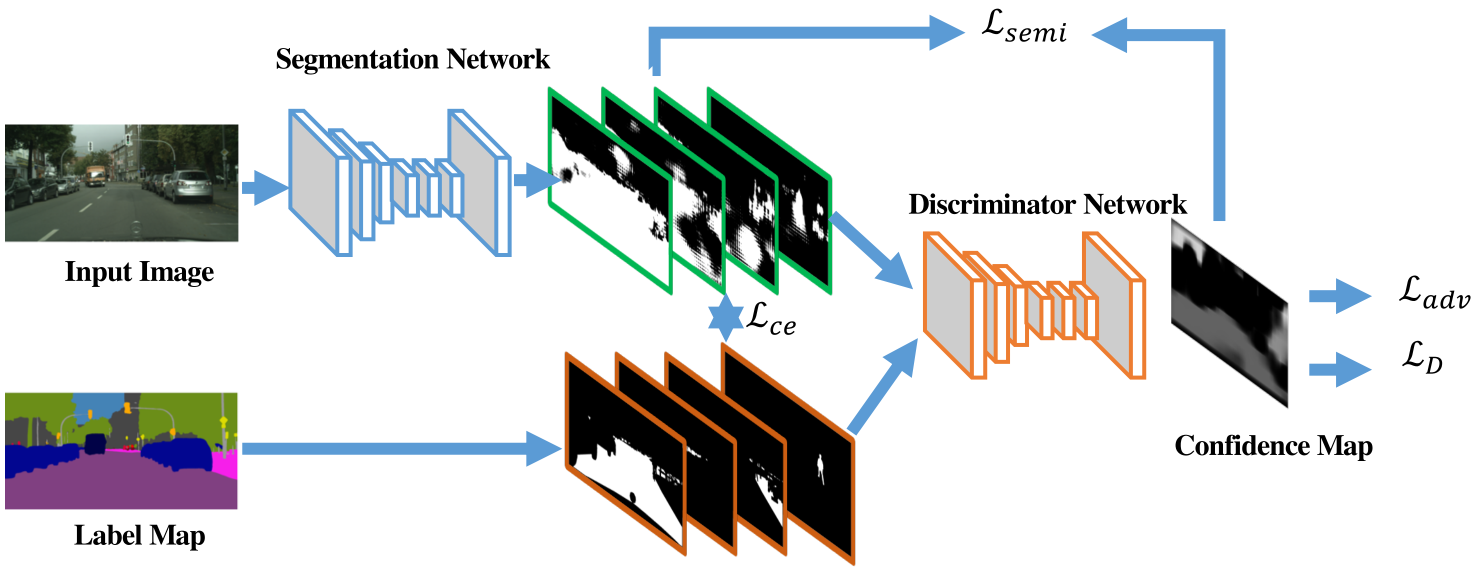
Qualitative Comparisons: Cityscapes



Summary

- Adversarial learning could be applied for Semantic segmentation
  - Performance improvement on **fully-supervised** setting
  - Exploit discriminator confidence maps of **unlabeled data**
- Code available at : https://github.com/hfslyc/AdvSemiSeg

Github



CrDoCo: Pixel-level Domain Transfer with Cross-Domain Consistency

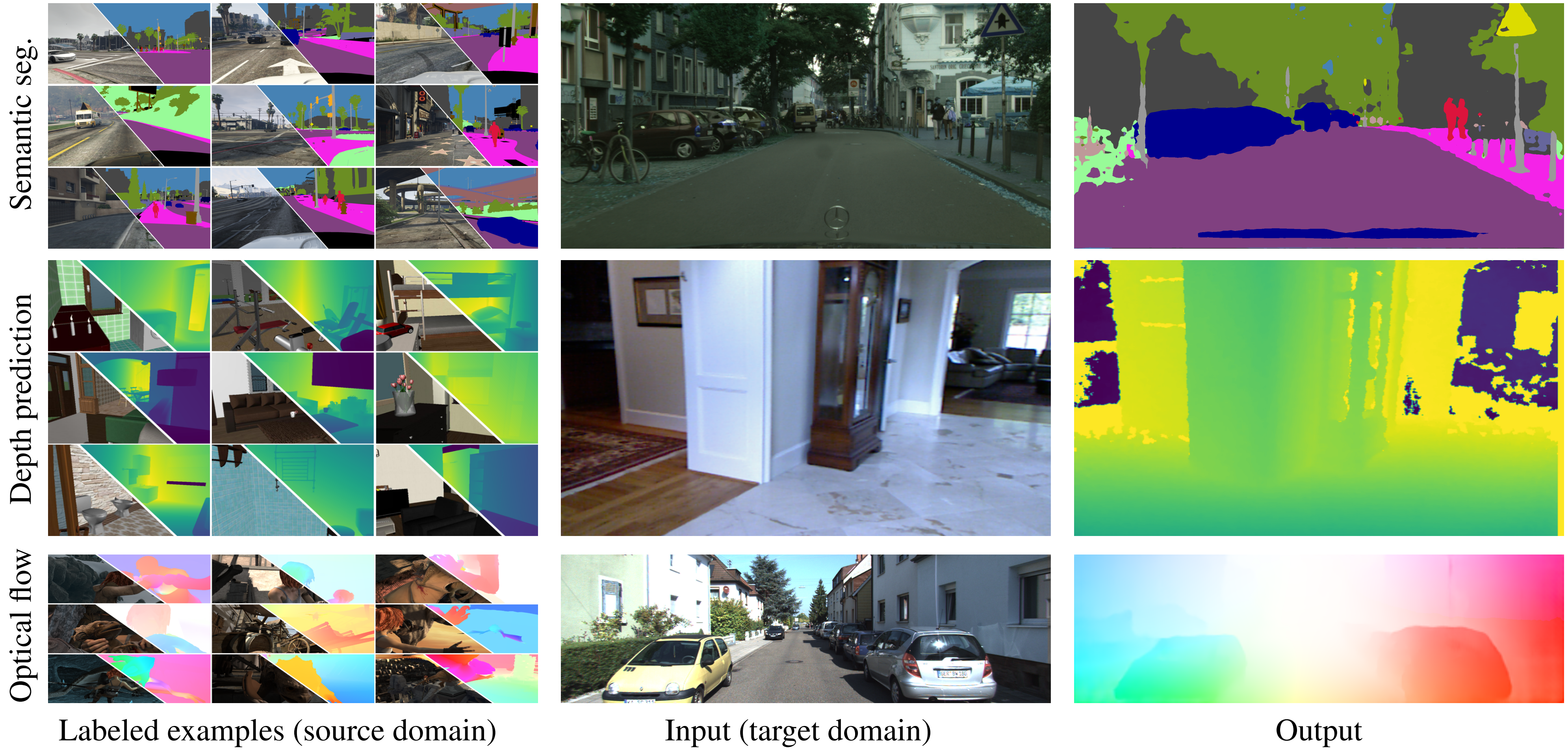
CVPR 2019

Yun-Chun Chen[1,2] Yen-Yu Lin[1] Ming-Hsuan Yang[3,4] Jia-Bin Huang[5]

[1]Academia Sinica [2]NTU [3]UC Merced [4]Google [5]Virginia Tech

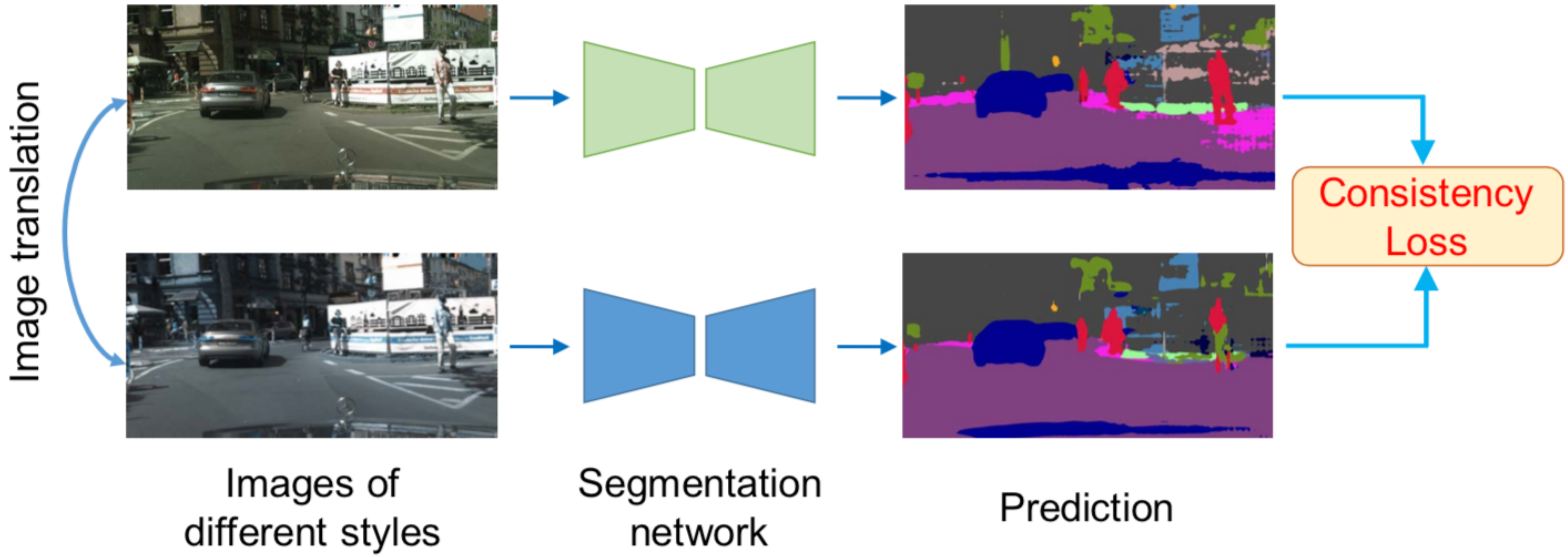
Unsupervised Domain Adaptation

- Input: A source dataset (labeled) and a target dataset (unlabeled)
- Goal: Transfer knowledge learned from source domain to target domain



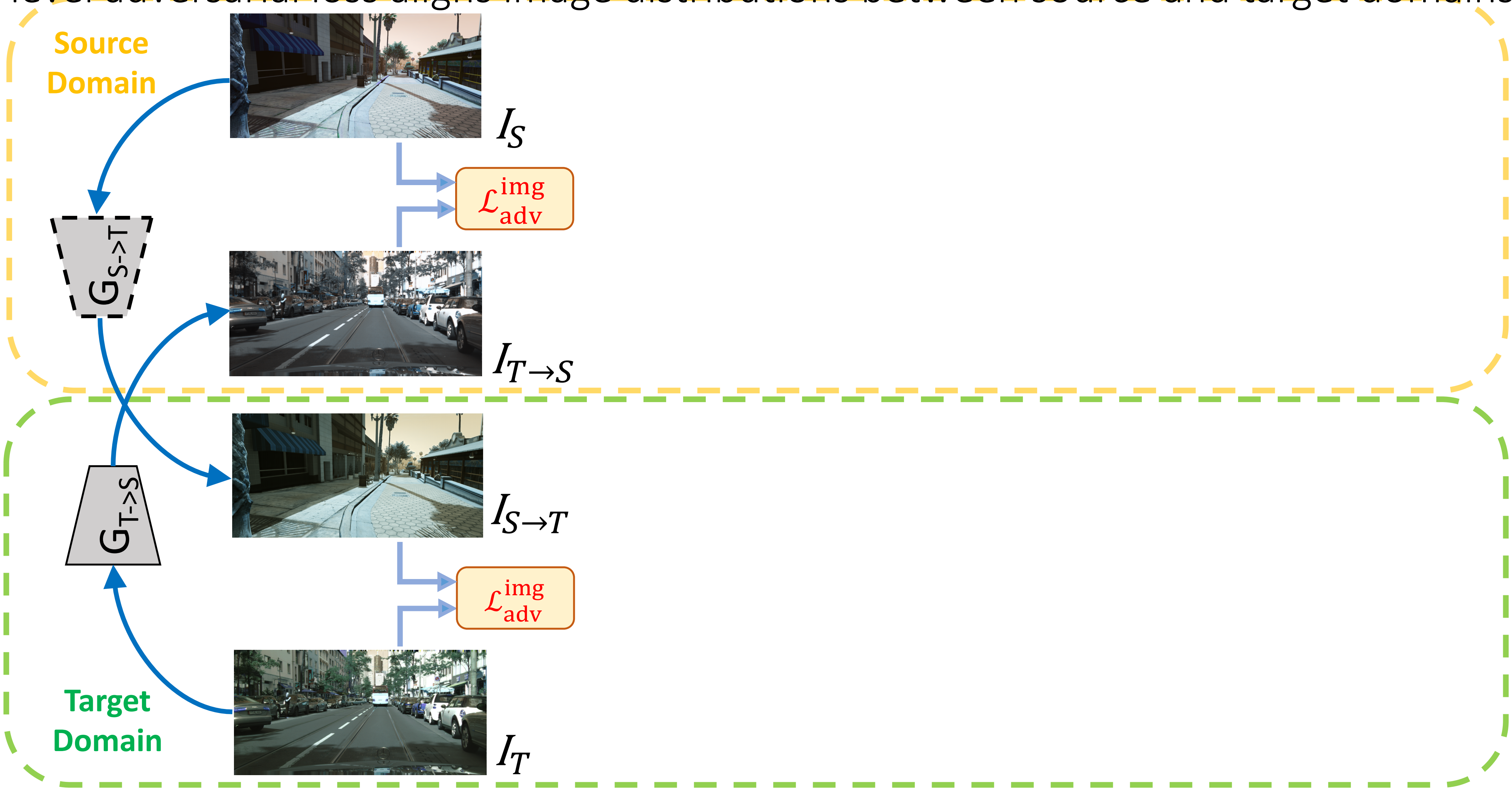
Main Idea

- Images in different domains may have different styles
- Task predictions should be the same



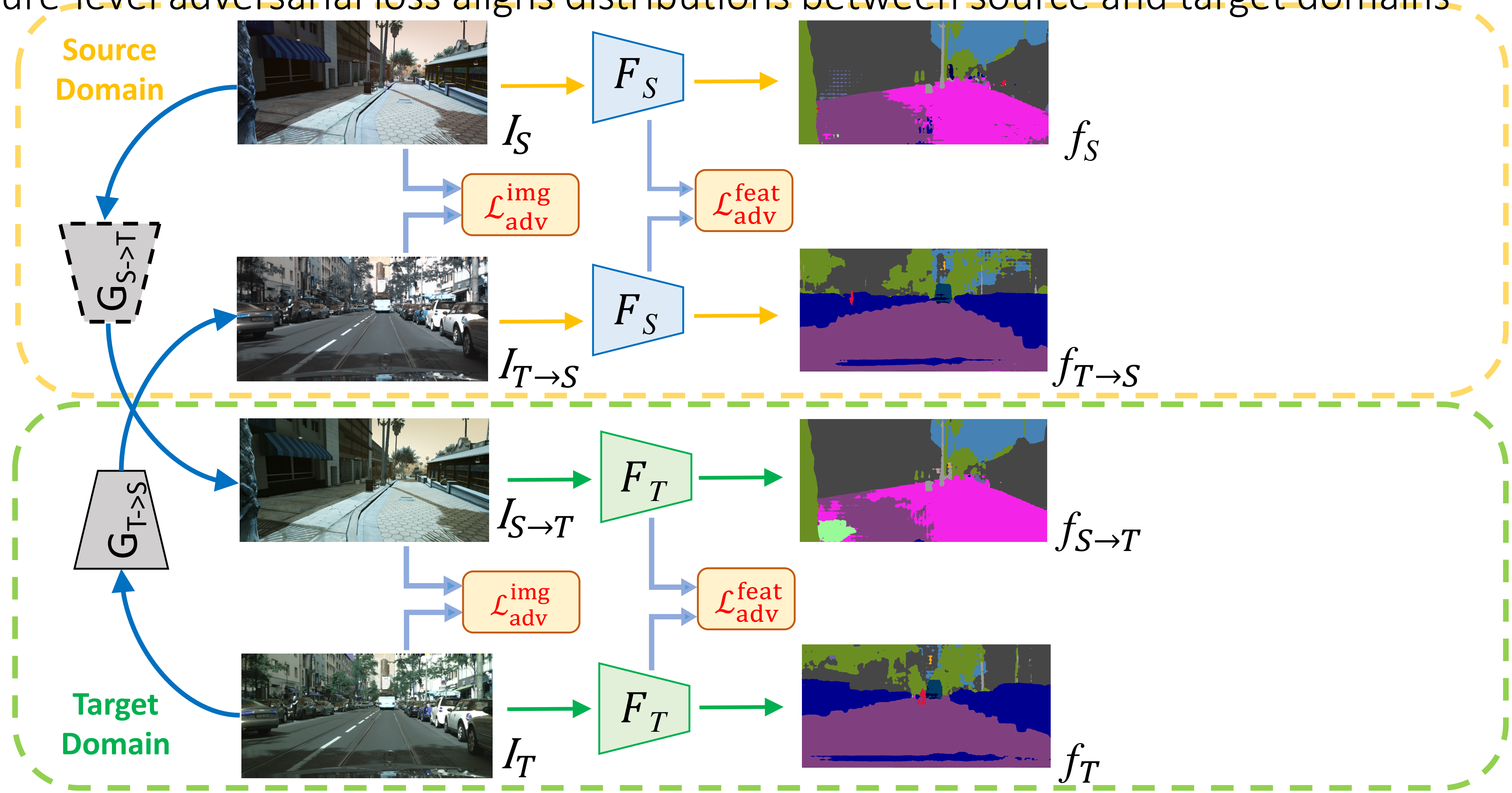
CrDoCo: Cross-Domain Consistency

- Pixel-level adversarial loss aligns image distributions between source and target domains



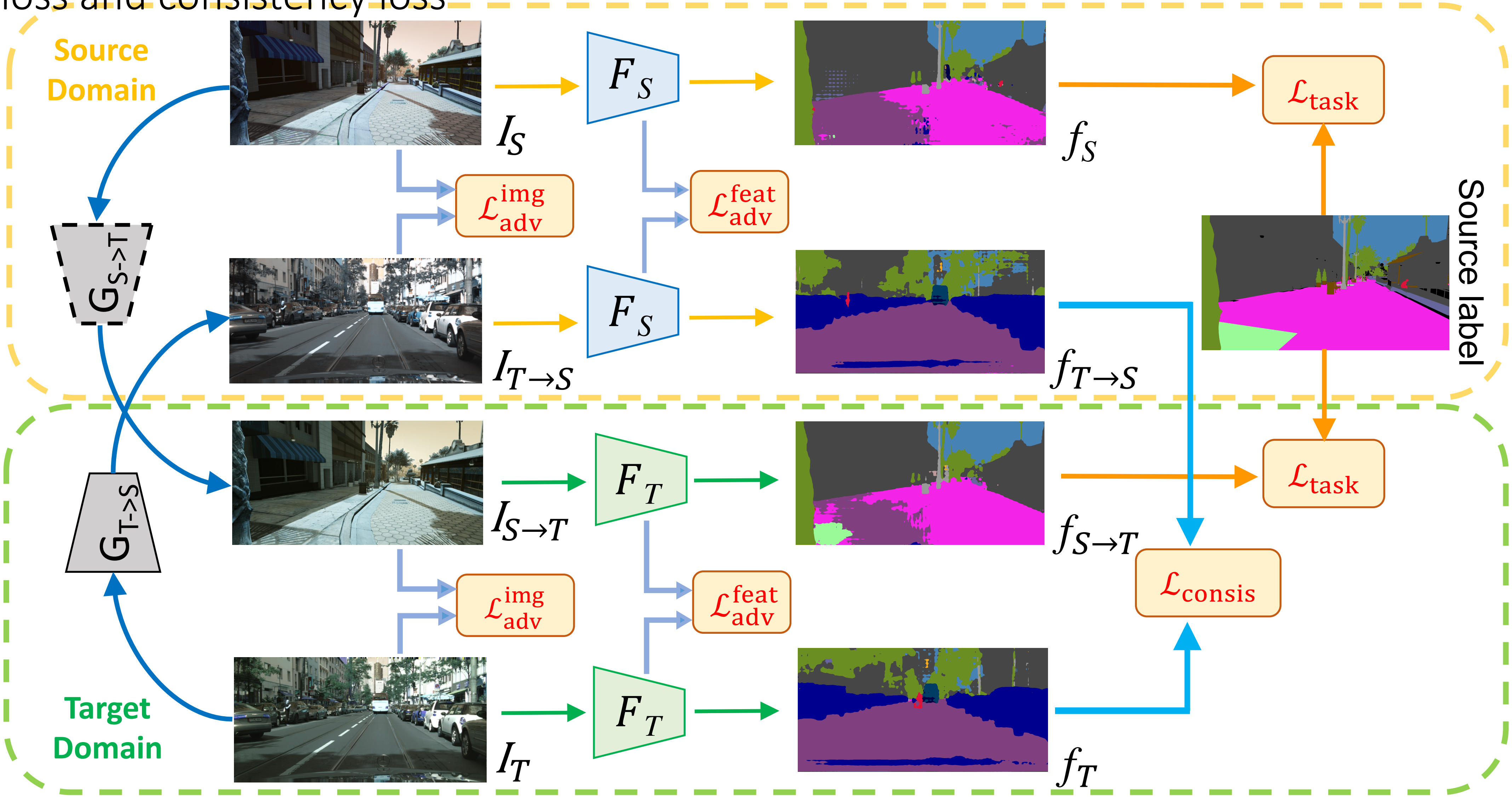
CrDoCo: Cross-Domain Consistency

- Feature-level adversarial loss aligns distributions between source and target domains

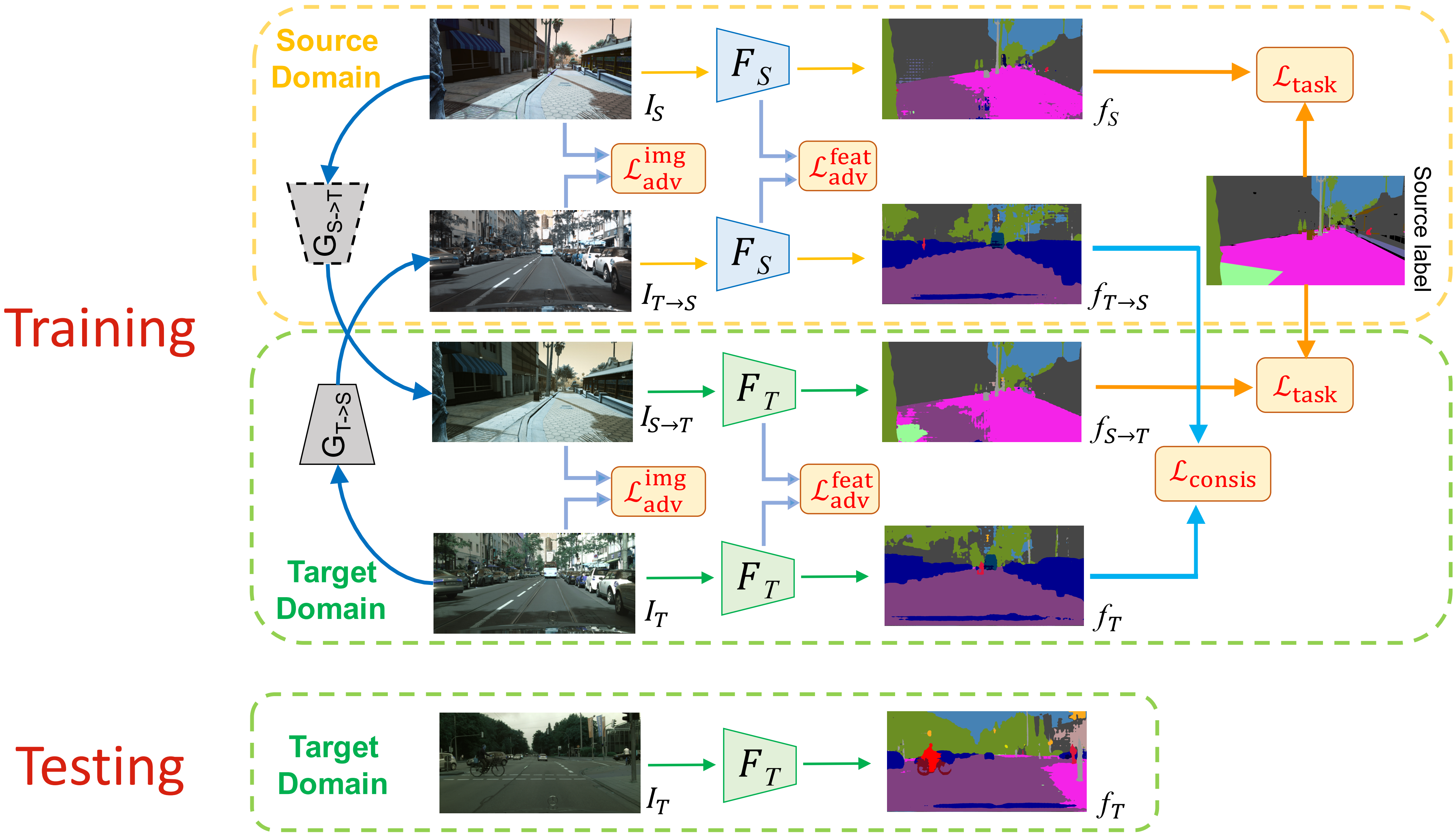


CrDoCo: Cross-Domain Consistency

- Task loss and consistency loss



CrDoCo: Cross-Domain Consistency



Experiments

- Synthetic-to-real adaptation
  - Semantic segmentation
  - Single-view depth prediction
  - Optical flow estimation

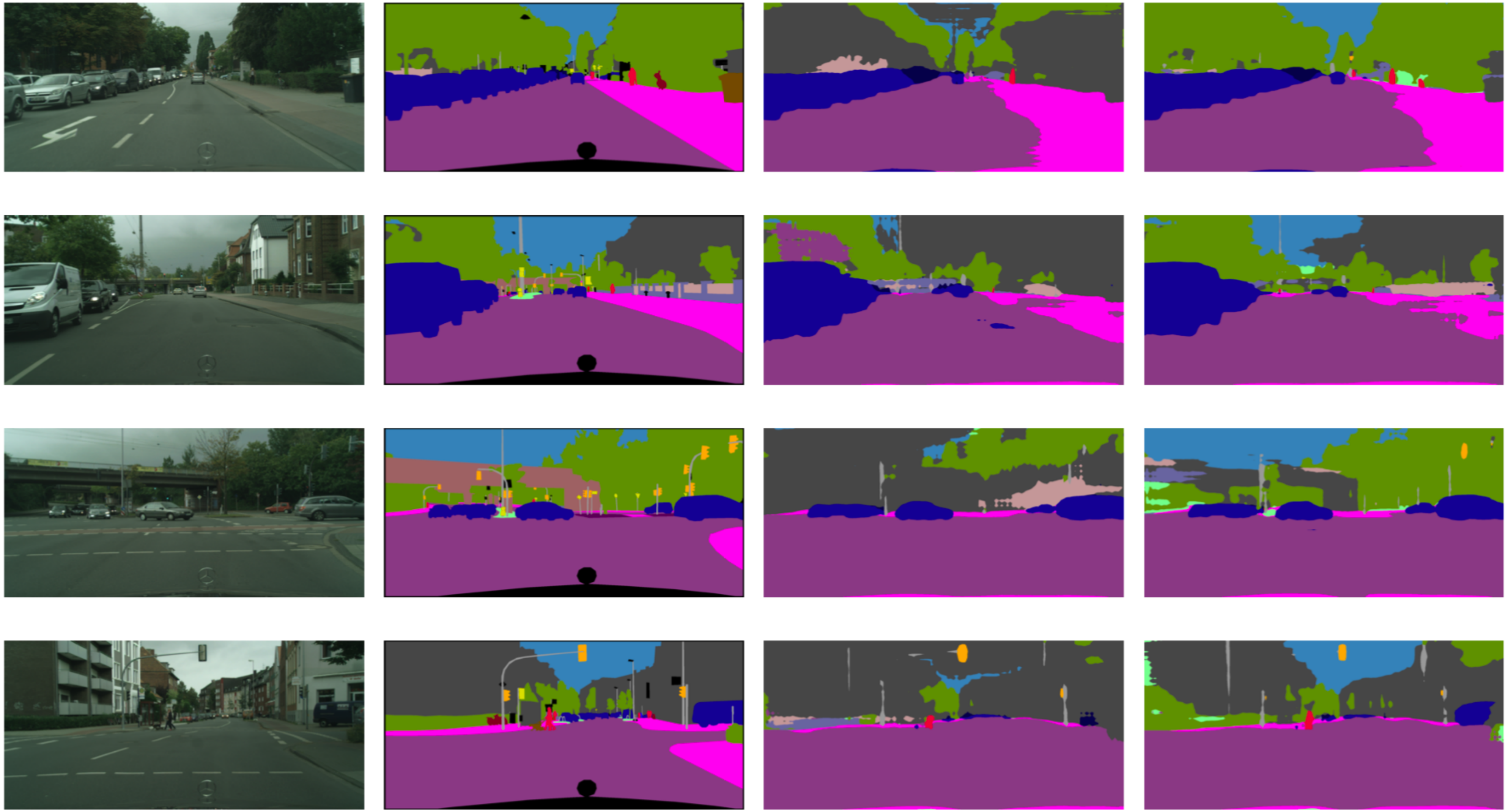
- Cross-city adaptation
  - Semantic segmentation

Synthetic-to-Real Adaptation

- Semantic segmentation

Method	GTA5 $\rightarrow$ Cityscapes mean IoU	GTA5 $\rightarrow$ Cityscapes Pixel acc.	SYNTHIA $\rightarrow$ Cityscapes mean IoU	SYNTHIA $\rightarrow$ Cityscapes Pixel acc.
Synth.	22.9	71.9	18.5	54.6
DS [Dundar arXiv 18]	38.3	<u>87.2</u>	29.5	<u>76.5</u>
UNIT [Liu NeurIPS 17]	39.1	87.1	28.0	70.8
FCNs ITW [Hoffman arXiv 17]	27.1	-	17.0	-
CyCADA [Hoffman ICML 18]	39.5	82.3	-	-
Ours w/o $\mathcal{L}_{\text{consis}}$	39.4	85.8	<u>29.8</u>	75.3
Ours	**45.1**	**89.2**	**33.4**	**79.5**

Semantic Segmentation Results



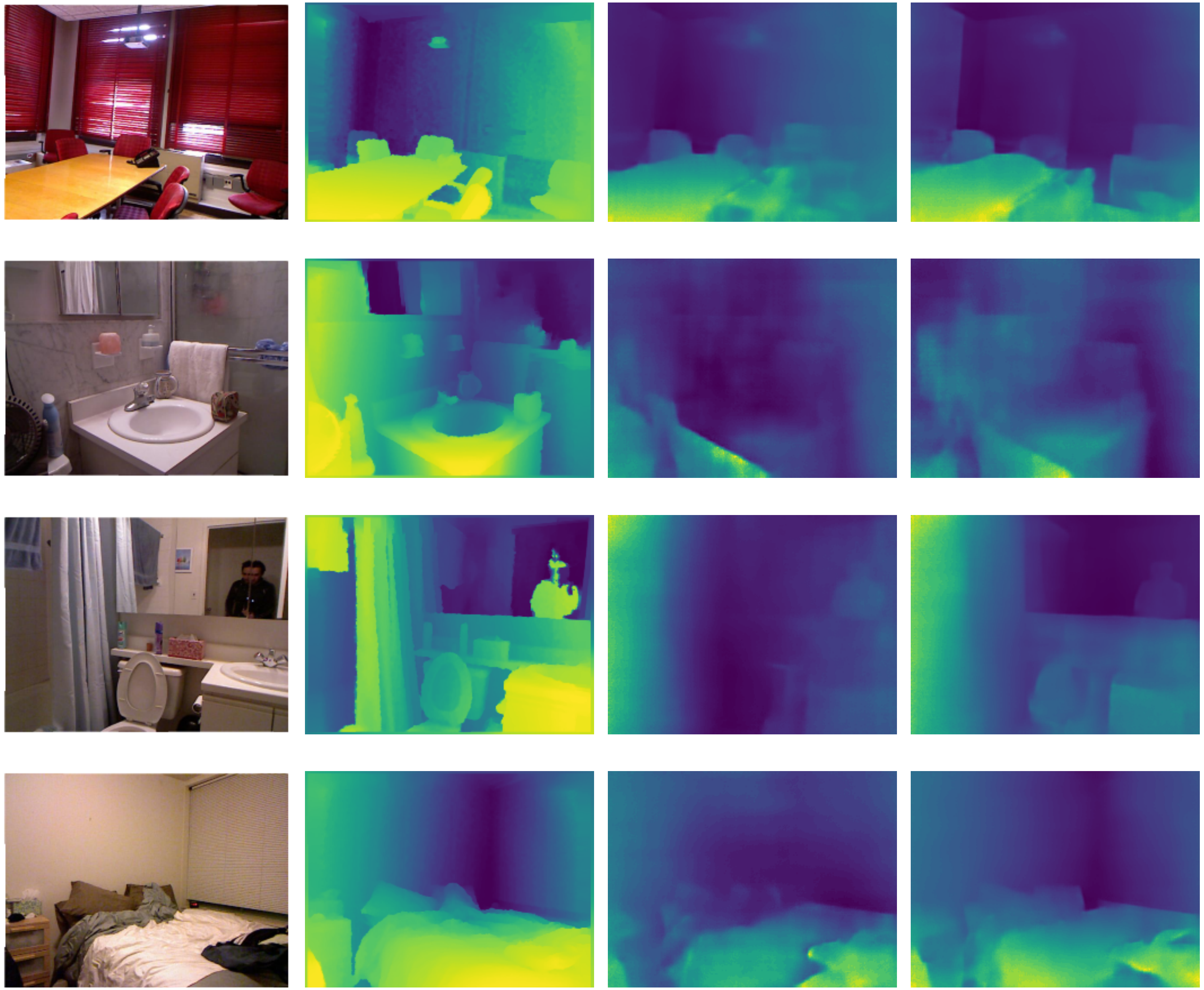
Input images | Ground truth | Ours w/o $\mathcal{L}_{\text{consis}}$ | Ours

Synthetic-to-Real Adaptation

- Single-view depth prediction

Method	SUNCG → NYUv2		
	Abs. Rel. ↓	Sq. Rel. ↓	RMSE ↓
Synth.	0.304	0.394	1.024
Baseline (train set mean)	0.439	0.641	1.148
T^2Net [Zheng ECCV 18]	0.257	<u>0.281</u>	0.915
Ours w/o $\mathcal{L}_{\mathrm{consis}}$	<u>0.254</u>	0.283	<u>0.911</u>
Ours	**0.233**	**0.272**	**0.898**

Depth Estimation Results



Input images | Ground truth | Ours w/o $\mathcal{L}_{\text{consis}}$ | Ours

Synthetic-to-Real Adaptation

- Optical flow estimation

Method	MPI Sintel → KITTI 2012			MPI Sintel → KITTI 2015		
	AEPE *train*	AEPE *test*	F1-Noc *test*	AEPE *train*	F1-all *train*	F1-all *test*
FlowNet2 [Ilg CVPR 17]	4.09	-	-	<u>10.06</u>	<u>30.37%</u>	-
PWC-Net [Sun CVPR 18]	<u>4.14</u>	<u>4.22</u>	**8.10%**	10.35	33.67%	-
Ours w/o $\mathcal{L}_{\text{consis}}$	4.16	4.92	13.52%	10.76	34.01%	<u>36.43%</u>
Ours	**2.19**	**3.16**	<u>8.57%</u>	**8.02**	**23.14%**	**25.83%**

Optical Flow Results

Cross-City Adaptation

- Semantic segmentation

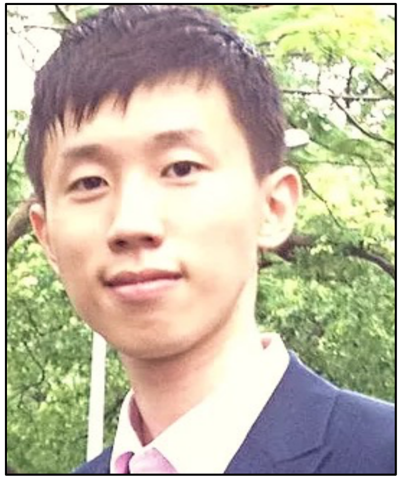
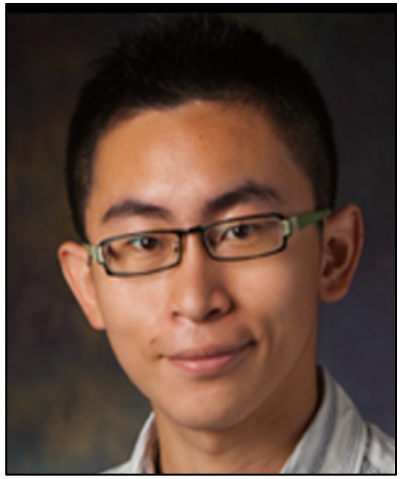
Method	Cityscapes → Cross-city			
	Rome	Rio	Tokyo	Taipei
Cross-City [Chen ICCV 17]	42.9	42.5	42.8	39.6
CBST [Zou ECCV 18]	53.6	**52.2**	48.8	**50.3**
AdaptSegNet [Tsai CVPR 18]	52.2	49.5	46.9	47.5
Ours w/o $\mathcal{L}_{\text{consis}}$	51.0	48.9	45.9	46.8
Ours	**55.1**	50.4	**51.2**	47.9

Summary

- Cross-domain consistency
    - Application agnostic
    - State-of-the-art performance

Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation

ICLR 2020


Hung-Yu Tseng
U.C. Merced

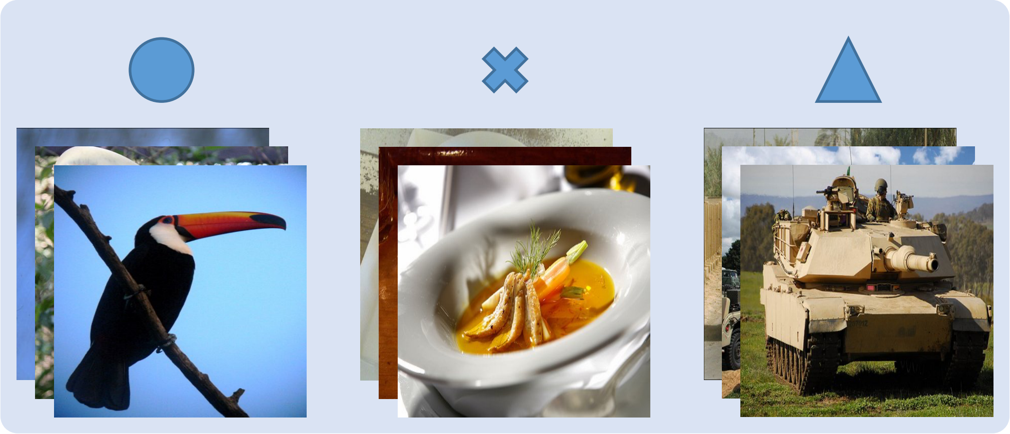
Hsin-Ying Lee
U.C. Merced

Jia-Bin Huang
Virginia Tech

Ming-Hsuan Yang
U.C. Merced
Google Research

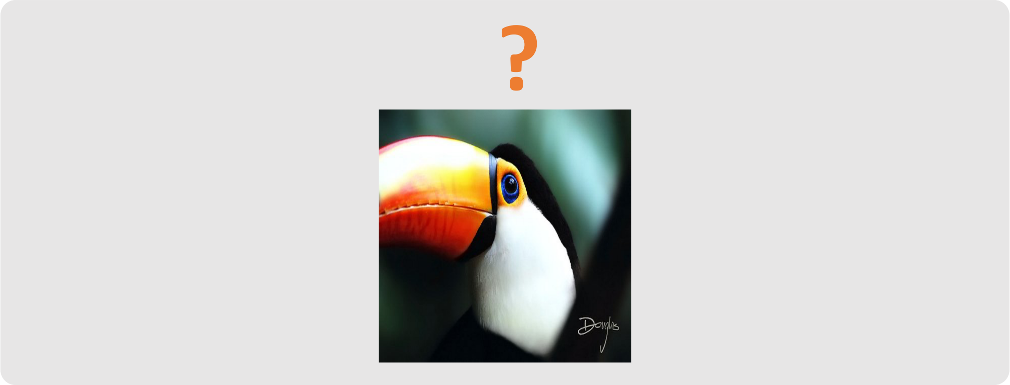
Few-Shot Classification

- Given: the few examples of novel categories (support set)
- Predict: the category of unlabeled data (query set)



Support set

$\mathbf{S} = \{(x_1, y_1), \dots, (x_k, y_k)\}$

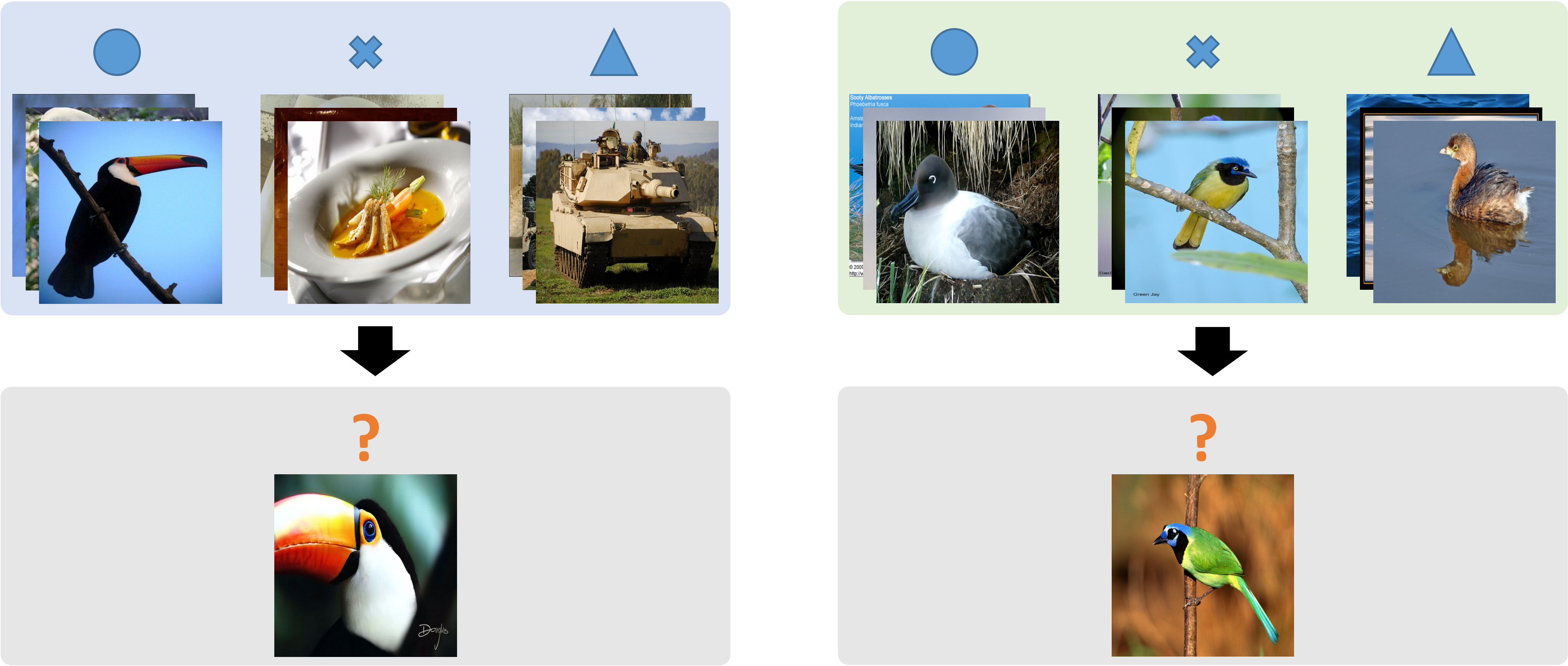


Query set

$\mathbf{Q} = \{x_{k+1}, \dots, x_l\}$

Cross-Domain Few-Shot Classification

Training domain (mini-ImageNet) Testing domain (CUB)



Metric-based few-shot methods do not perform well when the domain gap is large
Note that during the training stage, we do not have access to the data in the testing domain

Cross-Domain Few-Shot Classification

Significant performance drop!

Method	CUB 1-shot	CUB 5-shot
MatchingNet Vinyals et al. (2016)	61.16 ± 0.89	72.86 ± 0.70
ProtoNet Snell et al. (2017)	51.31 ± 0.91	70.77 ± 0.69
MAML Finn et al. (2017)	55.92 ± 0.95	72.09 ± 0.76
RelationNet Sung et al. (2018)	62.45 ± 0.98	76.11 ± 0.69

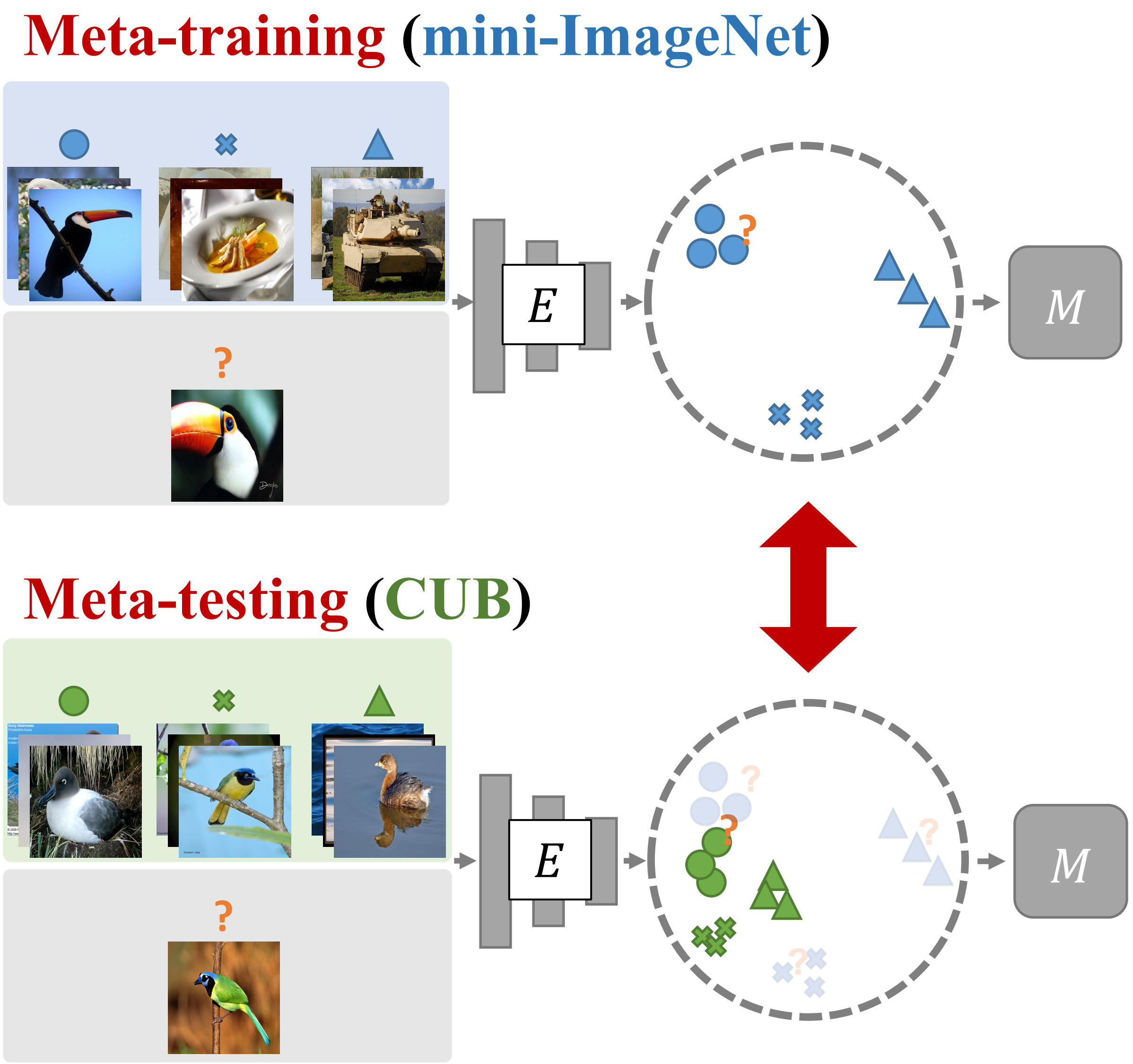
Train & test on the same domain

	***mini*-ImageNet →CUB**
MatchingNet	53.07±0.74
ProtoNet	62.02±0.70
MAML	51.34±0.72
RelationNet	57.71±0.73

Cross-domain

Chen et al. A Closer Look at Few-Shot Classification. ICLR, 2019

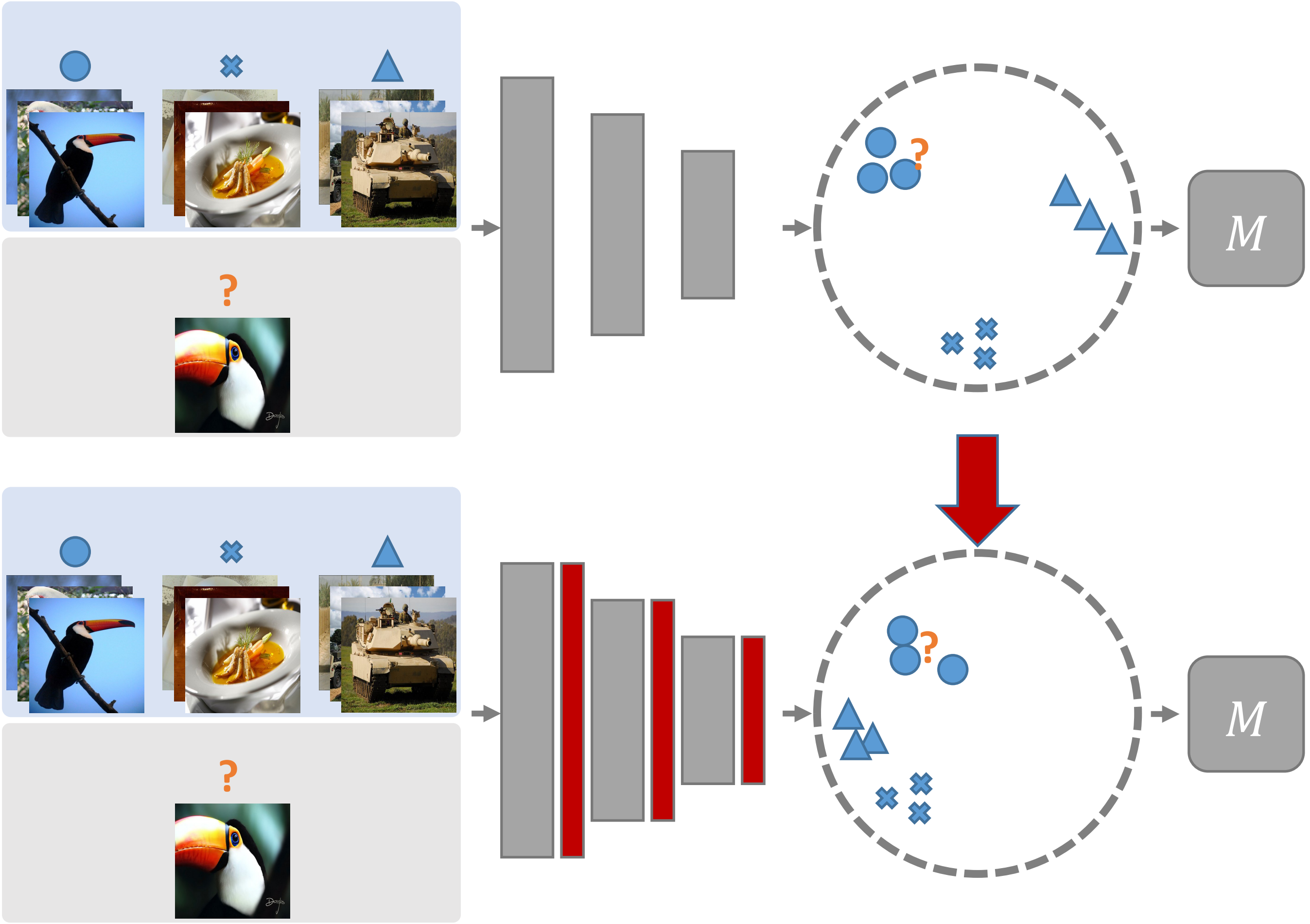
Domain Gap in Feature Space



Metric functions do not generalize to unseen feature distributions

Diversify the Feature Distribution

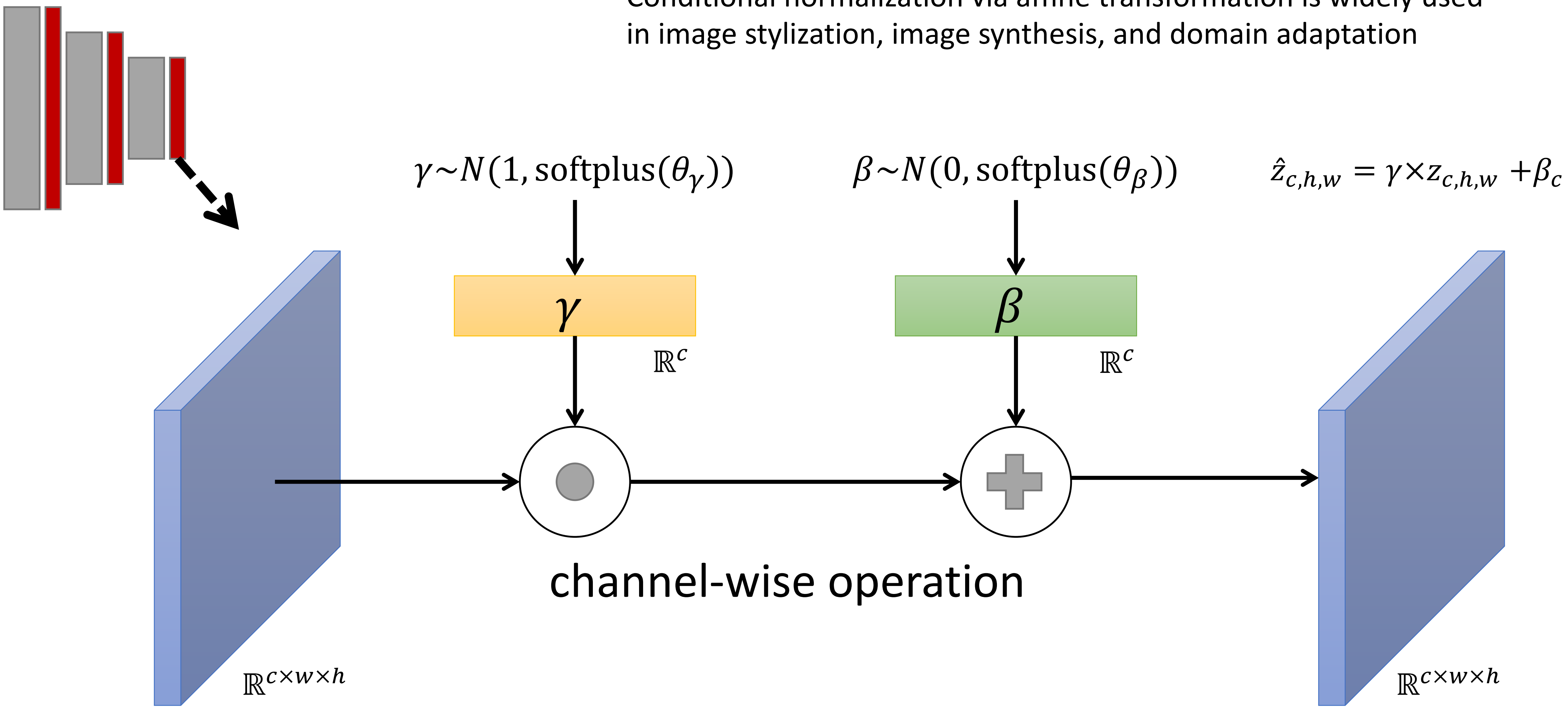
- Address few-shot classification under the domain generalization setting
- Augment features in the training domain to simulate various distributions



With the more diverse feature distribution in the training stage, we can improve the generalization ability of the metric function

Feature-Wise Transformation

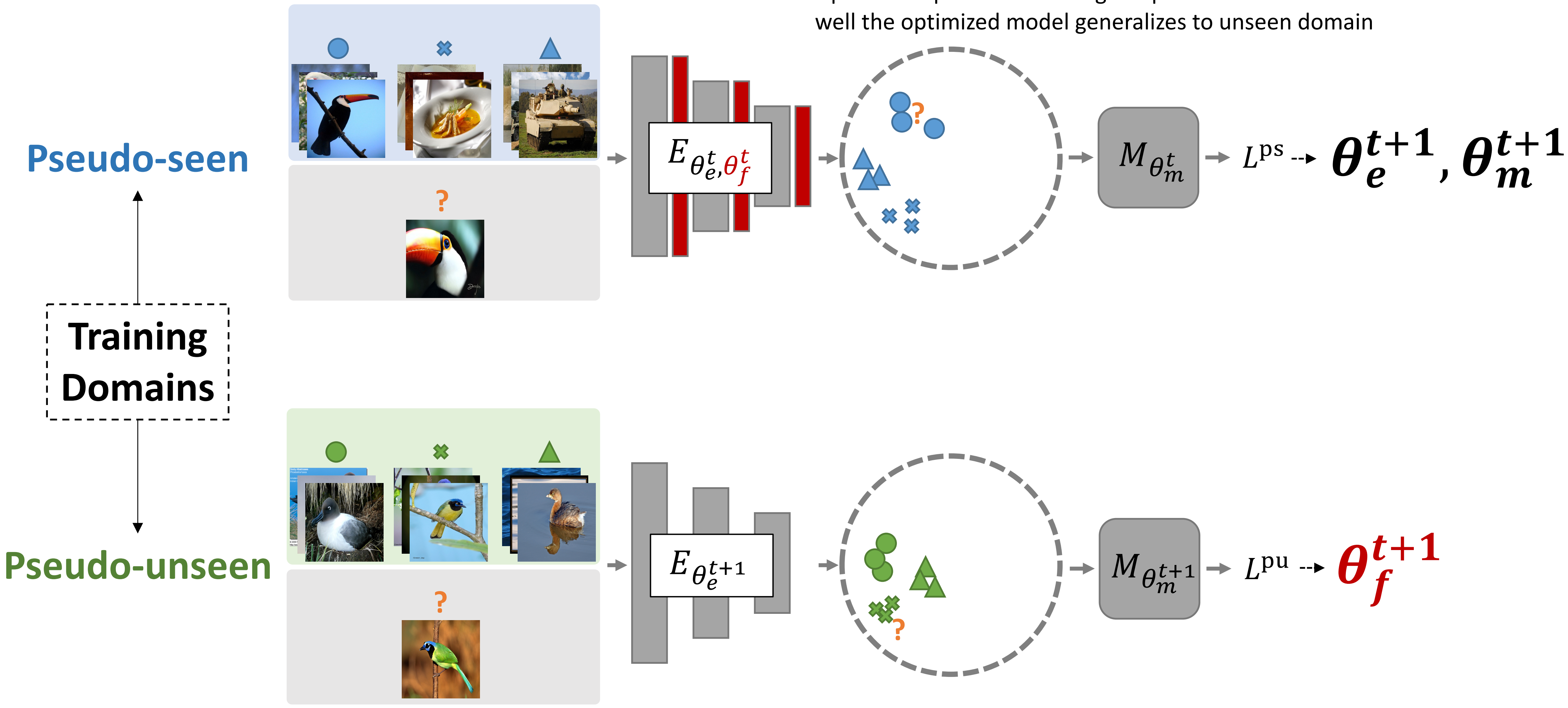
Conditional normalization via affine transformation is widely used in image stylization, image synthesis, and domain adaptation



How do we set hyper-parameters $\theta_f = \{\theta_\gamma, \theta_\beta\}$?

Learning to Generalize

1. Sample a pair of pseudo-seen and pseudo-unseen domains
2. Optimize parameters of a metric-based model with the feature-wise transformation using pseudo-seen domain
3. Remove feature-wise layer and compute loss of the optimized model on pseudo-unseen domain
4. Update the parameters using the pseudo-unseen loss as it indicates how well the optimized model generalizes to unseen domain



Experiments

- Datasets (domains): **mini-ImageNet**, CUB, Cars, Places, Plantae
- Applied methods: MatchingNet, RelationNet, GNN
  - Feature-wise transform used after batch norm in each residual block

- Scenario 1: train on mini-ImageNet 👉 test on others
  - Hand-tuned feature-wise transformation

- Scenario 2: select one as testing set 👉 train model on all other sets
  - Learning-to-learned feature-wise transformation

Scenario 1

- Train on mini-ImageNet 👉 test on others
- Hand-tuned feature-wise transformation

5-way 1-Shot	FT	mini-ImageNet	CUB	Cars	Places	Plantae
MatchingNet	-	$59.10 \pm 0.64\%$	$35.89 \pm 0.51\%$	$\mathbf{30.77 \pm 0.47\%}$	$49.86 \pm 0.79\%$	$32.70 \pm 0.60\%$
	✓	$58.76 \pm 0.61\%$	$\mathbf{36.61 \pm 0.53\%}$	$29.82 \pm 0.44\%$	$\mathbf{51.07 \pm 0.68\%}$	$\mathbf{34.48 \pm 0.50\%}$
RelationNet	-	$57.80 \pm 0.88\%$	$42.44 \pm 0.77\%$	$29.11 \pm 0.60\%$	$48.64 \pm 0.85\%$	$33.17 \pm 0.64\%$
	✓	$58.64 \pm 0.85\%$	$\mathbf{44.07 \pm 0.77\%}$	$28.63 \pm 0.59\%$	$\mathbf{50.68 \pm 0.87\%}$	$33.14 \pm 0.62\%$
GNN	-	$60.77 \pm 0.75\%$	$45.69 \pm 0.68\%$	$31.79 \pm 0.51\%$	$53.10 \pm 0.80\%$	$35.60 \pm 0.56\%$
	✓	$\boxed{\mathbf{66.32 \pm 0.80\%}}$	$\mathbf{47.47 \pm 0.75\%}$	$31.61 \pm 0.53\%$	$\mathbf{55.77 \pm 0.79\%}$	$35.95 \pm 0.58\%$
5-way 5-Shot	**FT**	**mini-ImageNet**	**CUB**	**Cars**	**Places**	**Plantae**
MatchingNet	-	$70.96 \pm 0.65\%$	$51.37 \pm 0.77\%$	$38.99 \pm 0.64\%$	$63.16 \pm 0.77\%$	$\mathbf{46.53 \pm 0.68\%}$
	✓	$\mathbf{72.53 \pm 0.69\%}$	$\mathbf{55.23 \pm 0.83\%}$	$\mathbf{41.24 \pm 0.65\%}$	$\mathbf{64.55 \pm 0.75\%}$	$41.69 \pm 0.63\%$
RelationNet	-	$71.00 \pm 0.69\%$	$57.77 \pm 0.69\%$	$37.33 \pm 0.68\%$	$63.32 \pm 0.76\%$	$44.00 \pm 0.60\%$
	✓	$\mathbf{73.78 \pm 0.64\%}$	$\mathbf{59.46 \pm 0.71\%}$	$\mathbf{39.91 \pm 0.69\%}$	$\mathbf{66.28 \pm 0.72\%}$	$\mathbf{45.08 \pm 0.59\%}$
GNN	-	$80.87 \pm 0.56\%$	$62.25 \pm 0.65\%$	$44.28 \pm 0.63\%$	$70.84 \pm 0.65\%$	$52.53 \pm 0.59\%$
	✓	$\boxed{\mathbf{81.98 \pm 0.55\%}}$	$\mathbf{66.98 \pm 0.68\%}$	$\mathbf{44.90 \pm 0.64\%}$	$\mathbf{73.94 \pm 0.67\%}$	$\mathbf{53.85 \pm 0.62\%}$

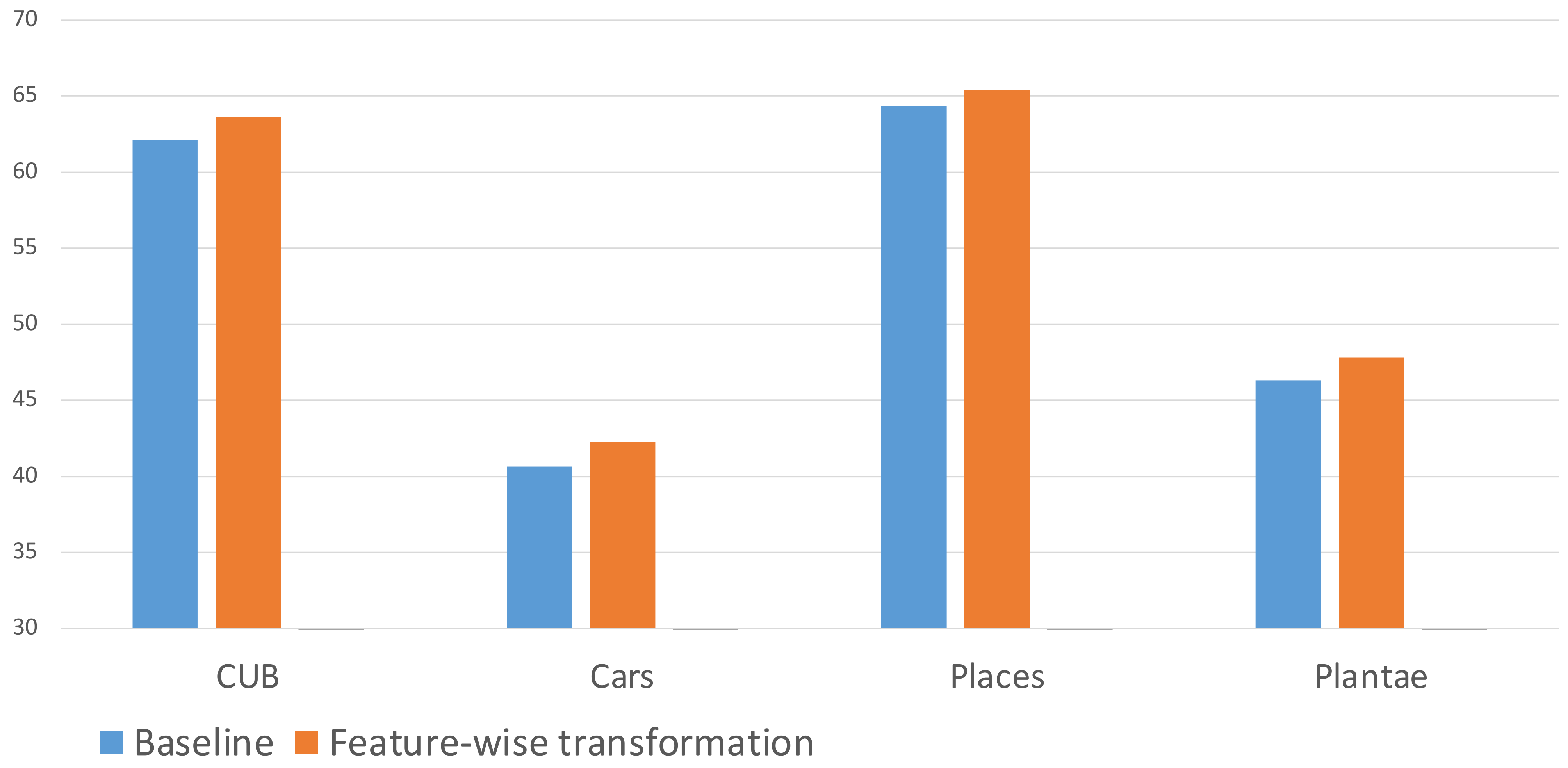
Scenario 2

- Train on multiple training sets 👉 test on one set
- LFT: use learning-to-learn method to determine parameters

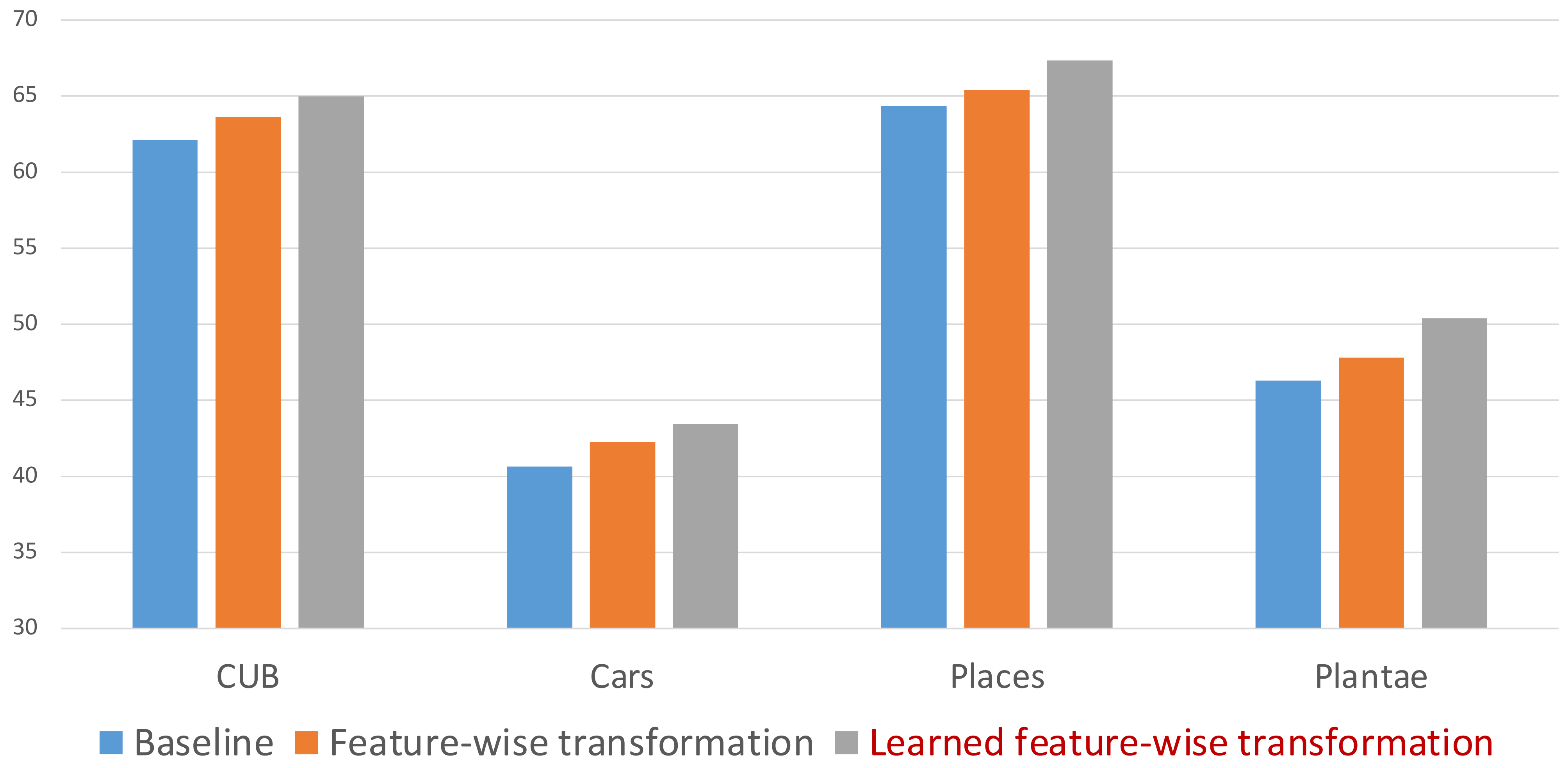
5-way 1-Shot		CUB	Cars	Places	Plantae
MatchingNet	-	37.90 ± 0.55%	28.96 ± 0.45%	49.01 ± 0.65%	33.21 ± 0.51%
	FT	41.74 ± 0.59%	28.30 ± 0.44%	48.77 ± 0.65%	32.15 ± 0.50%
	LFT	**43.29 ± 0.59%**	**30.62 ± 0.48%**	**52.51 ± 0.67%**	**35.12 ± 0.54%**
RelationNet	-	44.33 ± 0.59%	29.53 ± 0.45%	47.76 ± 0.63%	33.76 ± 0.52%
	FT	44.67 ± 0.58%	30.38 ± 0.47%	48.40 ± 0.64%	35.40 ± 0.53%
	LFT	**48.38 ± 0.63%**	**32.21 ± 0.51%**	**50.74 ± 0.66%**	35.00 ± 0.52%
GNN	-	49.46 ± 0.73%	32.95 ± 0.56%	51.39 ± 0.80%	37.15 ± 0.60%
	FT	48.24 ± 0.75%	33.26 ± 0.56%	54.81 ± 0.81%	37.54 ± 0.62%
	LFT	**51.51 ± 0.80%**	**34.12 ± 0.63%**	**56.31 ± 0.80%**	**42.09 ± 0.68%**

5-way 5-Shot		CUB	Cars	Places	Plantae
MatchingNet	-	51.92 ± 0.80%	39.87 ± 0.51%	61.82 ± 0.57%	47.29 ± 0.51%
	FT	56.29 ± 0.80%	39.58 ± 0.54%	62.32 ± 0.58%	46.48 ± 0.52%
	LFT	**61.41 ± 0.57%**	**43.08 ± 0.55%**	**64.99 ± 0.59%**	**48.32 ± 0.57%**
RelationNet	-	62.13 ± 0.74%	40.64 ± 0.54%	64.34 ± 0.57%	46.29 ± 0.56%
	FT	63.64 ± 0.77%	42.24 ± 0.57%	65.42 ± 0.58%	47.81 ± 0.51%
	LFT	**64.99 ± 0.54%**	**43.44 ± 0.59%**	**67.35 ± 0.54%**	**50.39 ± 0.52%**
GNN	-	69.26 ± 0.68%	48.91 ± 0.67%	72.59 ± 0.67%	58.36 ± 0.68%
	FT	70.37 ± 0.68%	47.68 ± 0.63%	74.48 ± 0.70%	57.85 ± 0.68%
	LFT	**73.11 ± 0.68%**	**49.88 ± 0.67%**	**77.05 ± 0.65%**	**58.84 ± 0.66%**

Scenario 2 (5-Shot Classification Results)

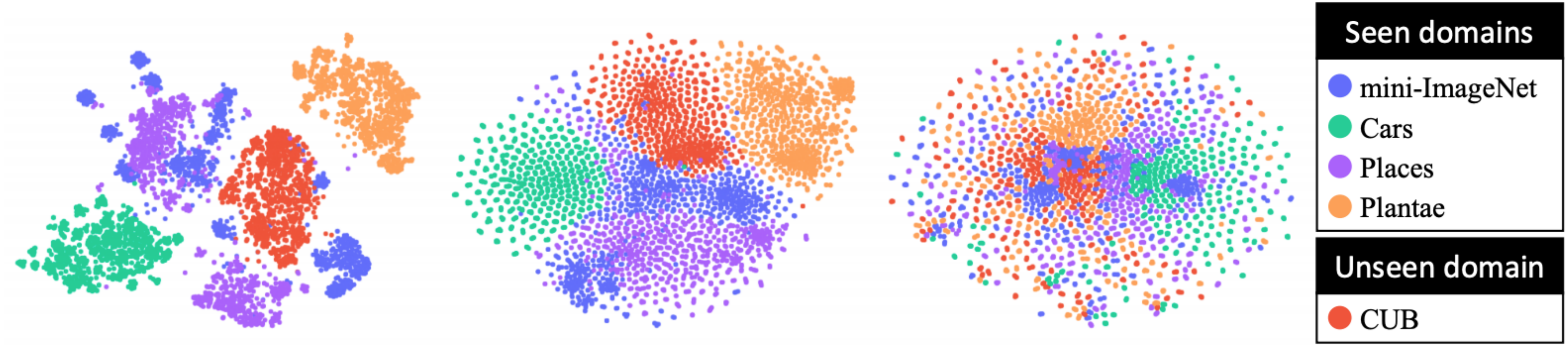


Scenario 2 (5-Shot Classification Results)



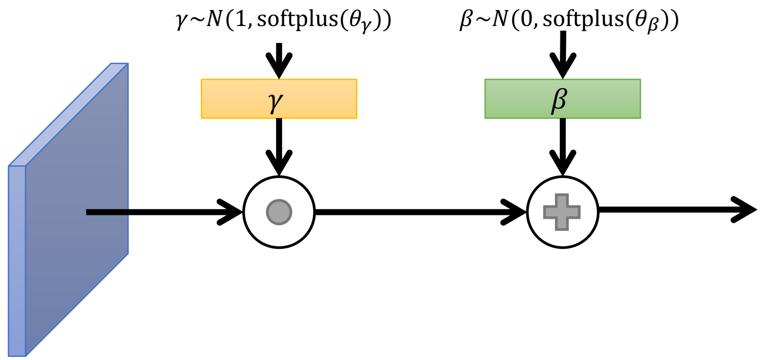
Visualization of Feature Space

w/o FT | Hand-tuned FT | Learned FT

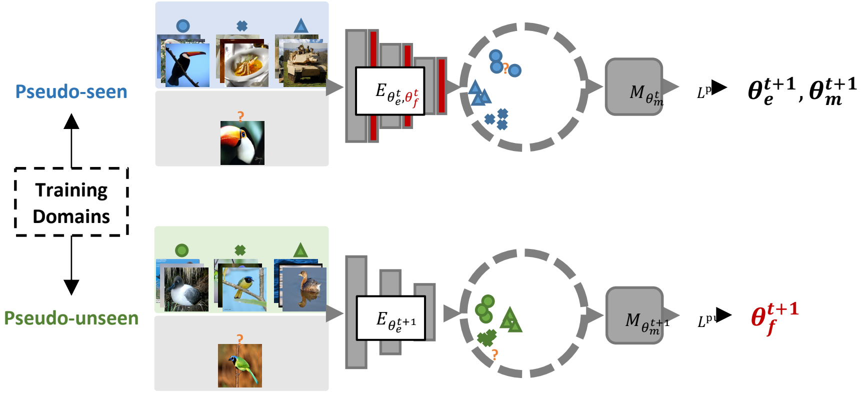


Summary

- Feature-wise transformation



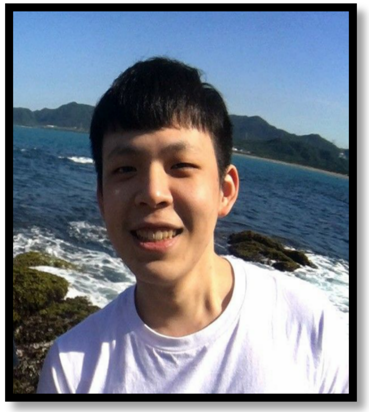
- Learning-to-generalize algorithm



- Code and dataset available at bit.ly/CrossDomainFewShot

Project Page

Every Pixel Matters: Center-aware Feature Alignment for Domain Adaptive Object Detector



Cheng-Chun Hsu
Academia Sinica

Yi-Hsuan Tsai
NEC Labs

Yen-Yu Lin
NCTU

Ming-Hsuan Yang
UC Merced/Google

NEC Laboratories America
Relentless passion for innovation

University of California
UC Merced

Problem Setting

Supervised Learning

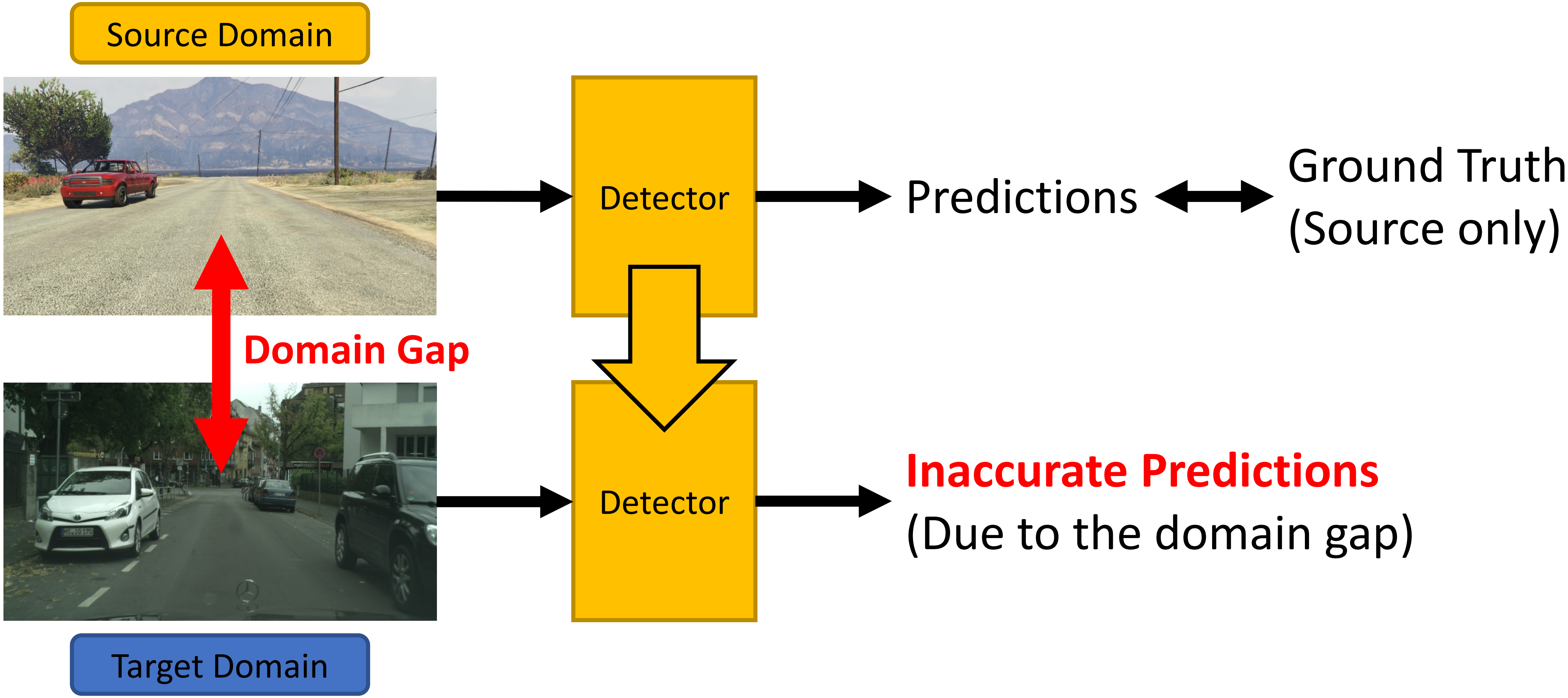
Source Domain

Detector

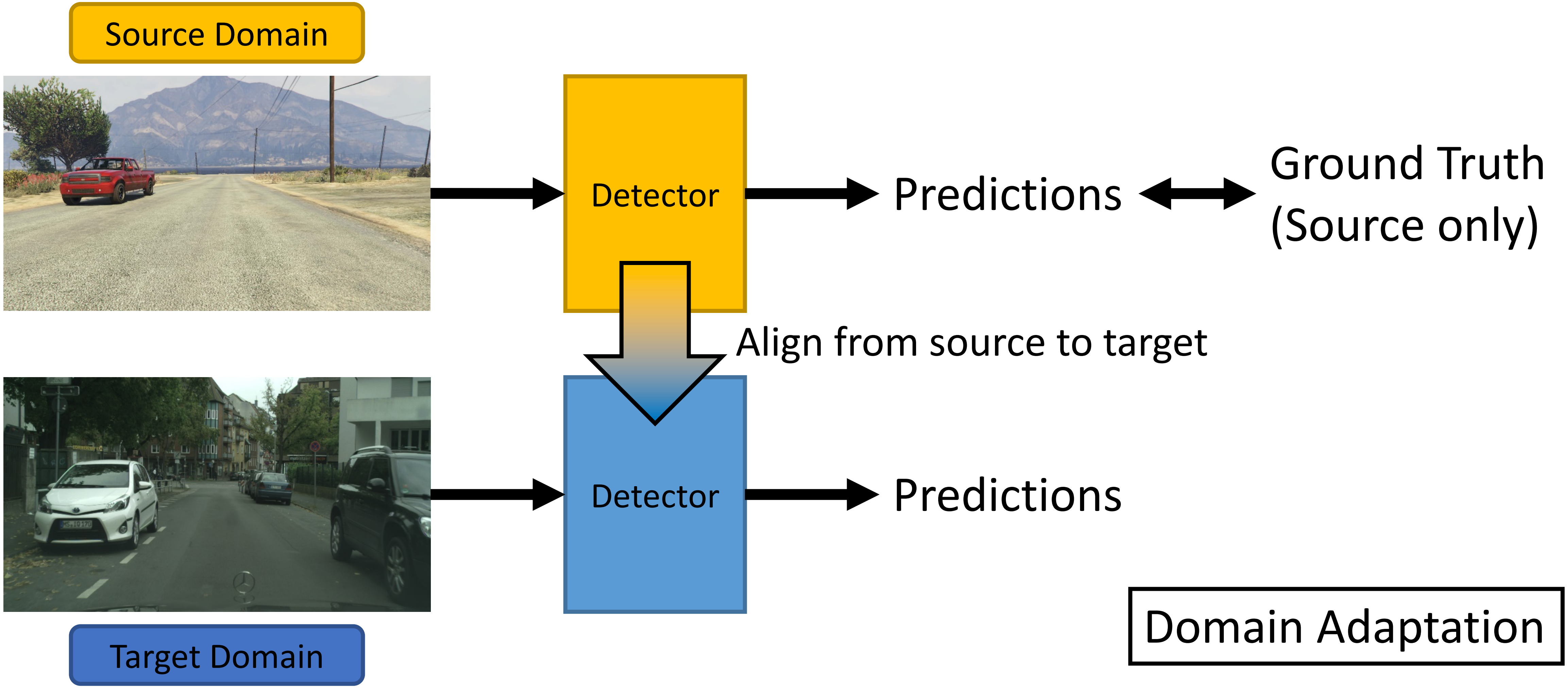
Predictions

Ground Truth
(Source only)

Problem Setting



Problem Setting



Motivation

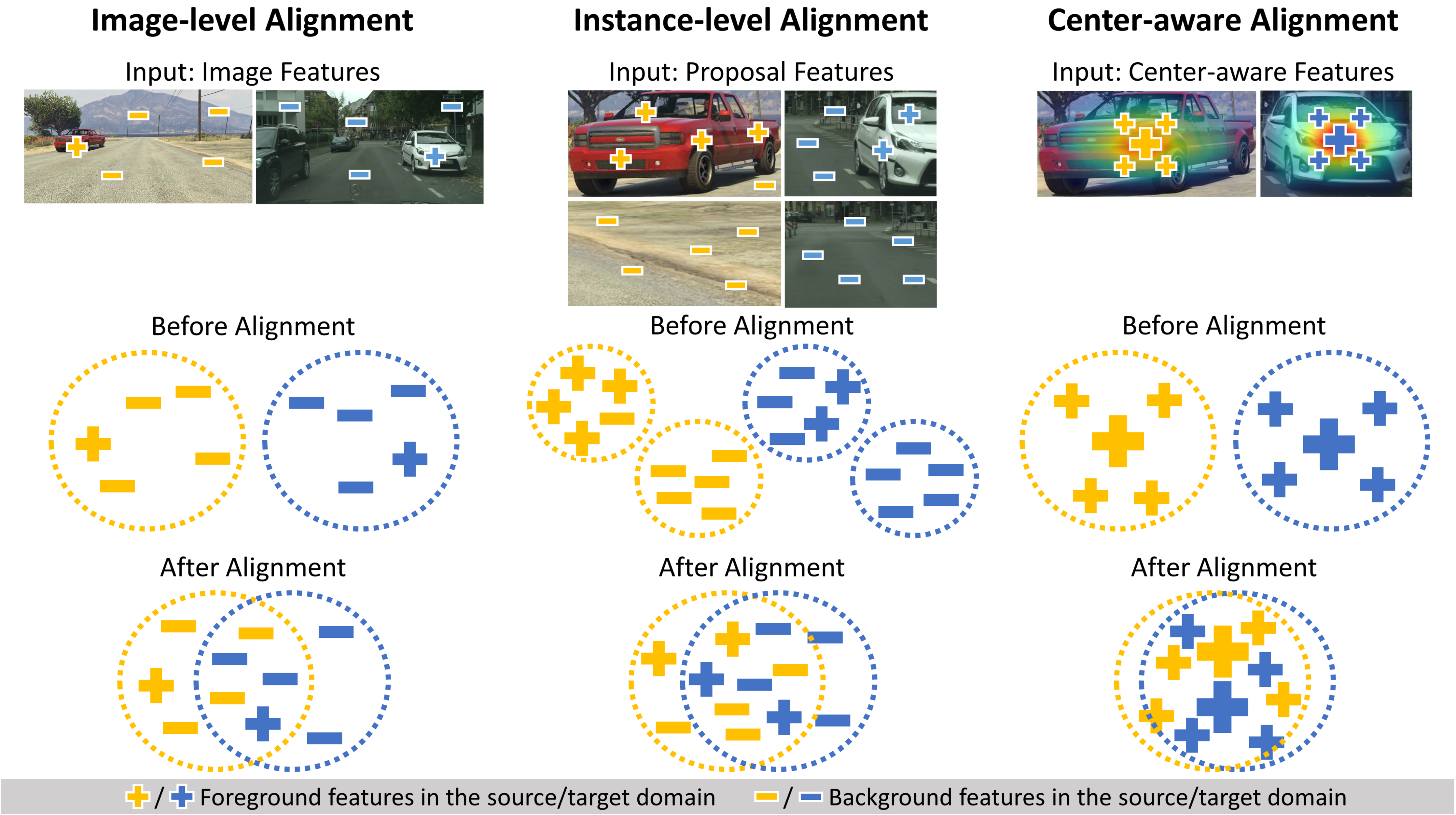
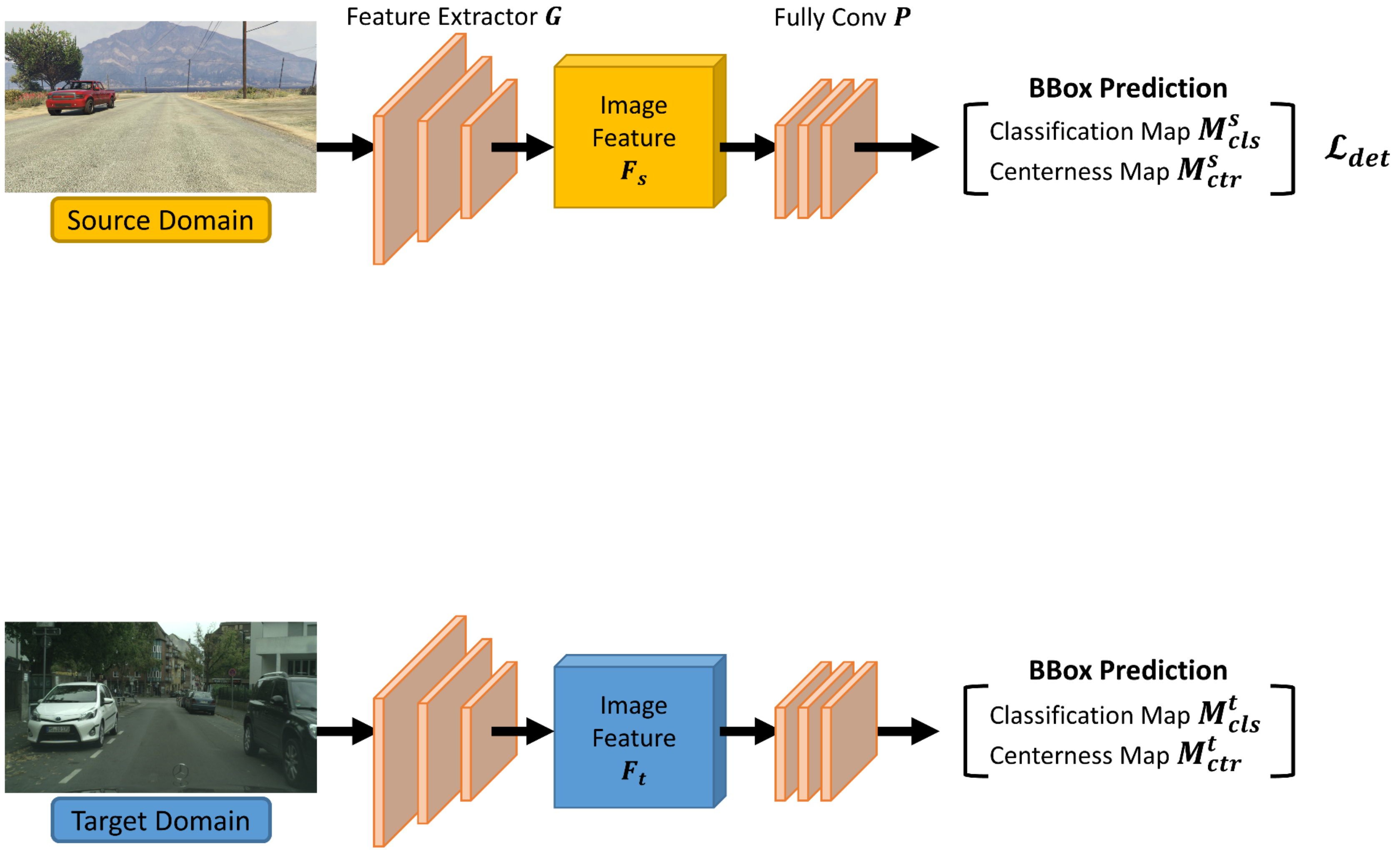


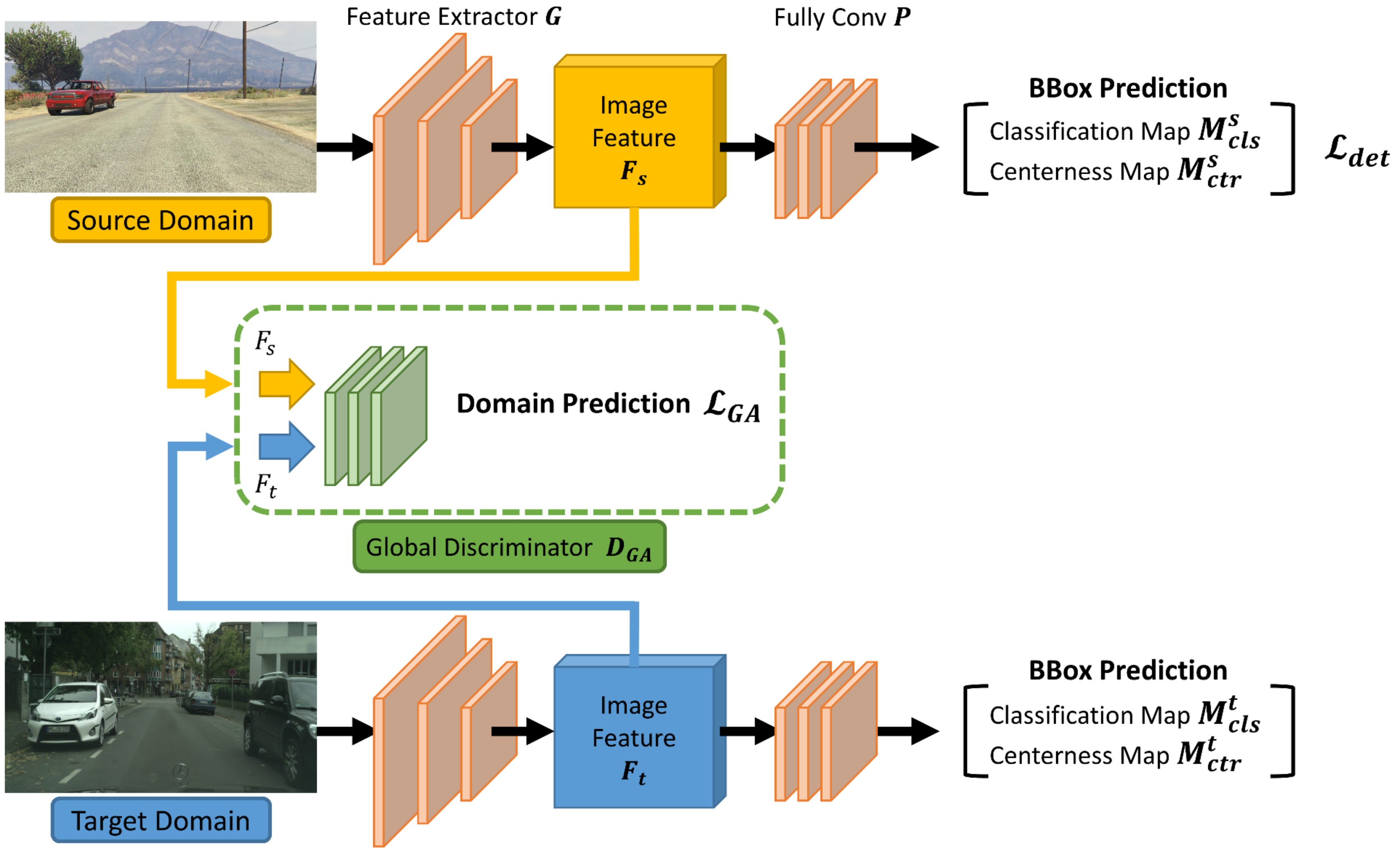
Image-level Alignment
Input: Image Features
Before Alignment
After Alignment
Instance-level Alignment
Input: Proposal Features
Before Alignment
After Alignment
Center-aware Alignment
Input: Center-aware Features
Before Alignment
After Alignment
/ Foreground features in the source/target domain
/ Background features in the source/target domain

Approach



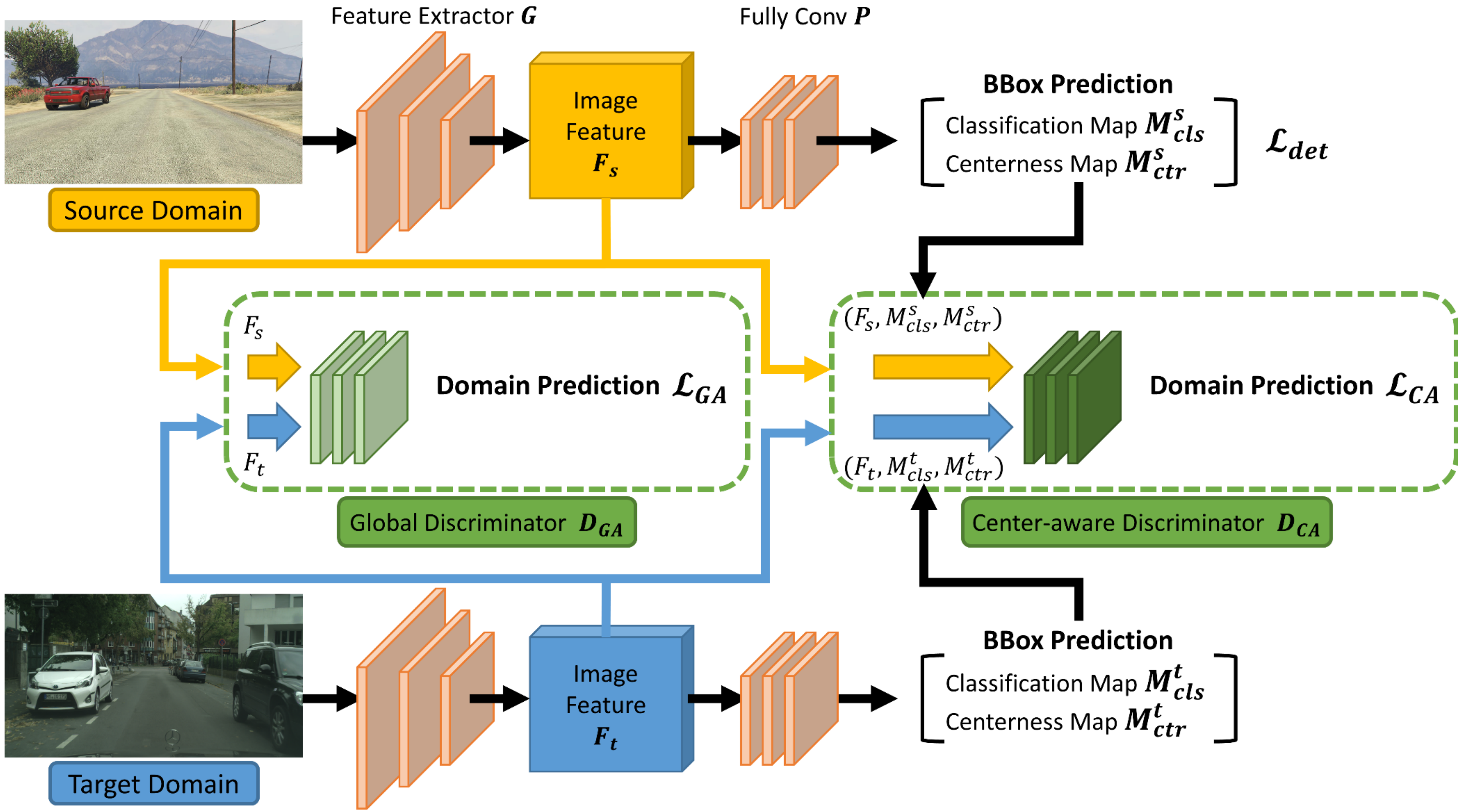
Feature Extractor G
Fully Conv P
Source Domain
Image Feature F_s
BBox Prediction
Classification Map M^s_{cls}
Centerness Map M^s_{ctr}
$\mathcal{L}_{det}$
Target Domain
Image Feature F_t
BBox Prediction
Classification Map M^t_{cls}
Centerness Map M^t_{ctr}

Approach

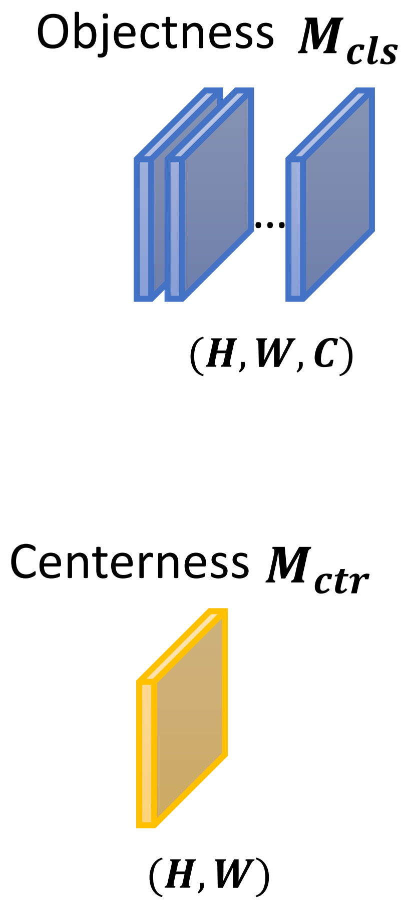


Feature Extractor G
Fully Conv P
Image Feature F_s
Source Domain
BBox Prediction
Classification Map M^s_{cls}
Centerness Map M^s_{ctr}
$\mathcal{L}_{det}$
F_s
F_t
Domain Prediction $\mathcal{L}_{GA}$
Global Discriminator D_{GA}
Image Feature F_t
Target Domain
BBox Prediction
Classification Map M^t_{cls}
Centerness Map M^t_{ctr}

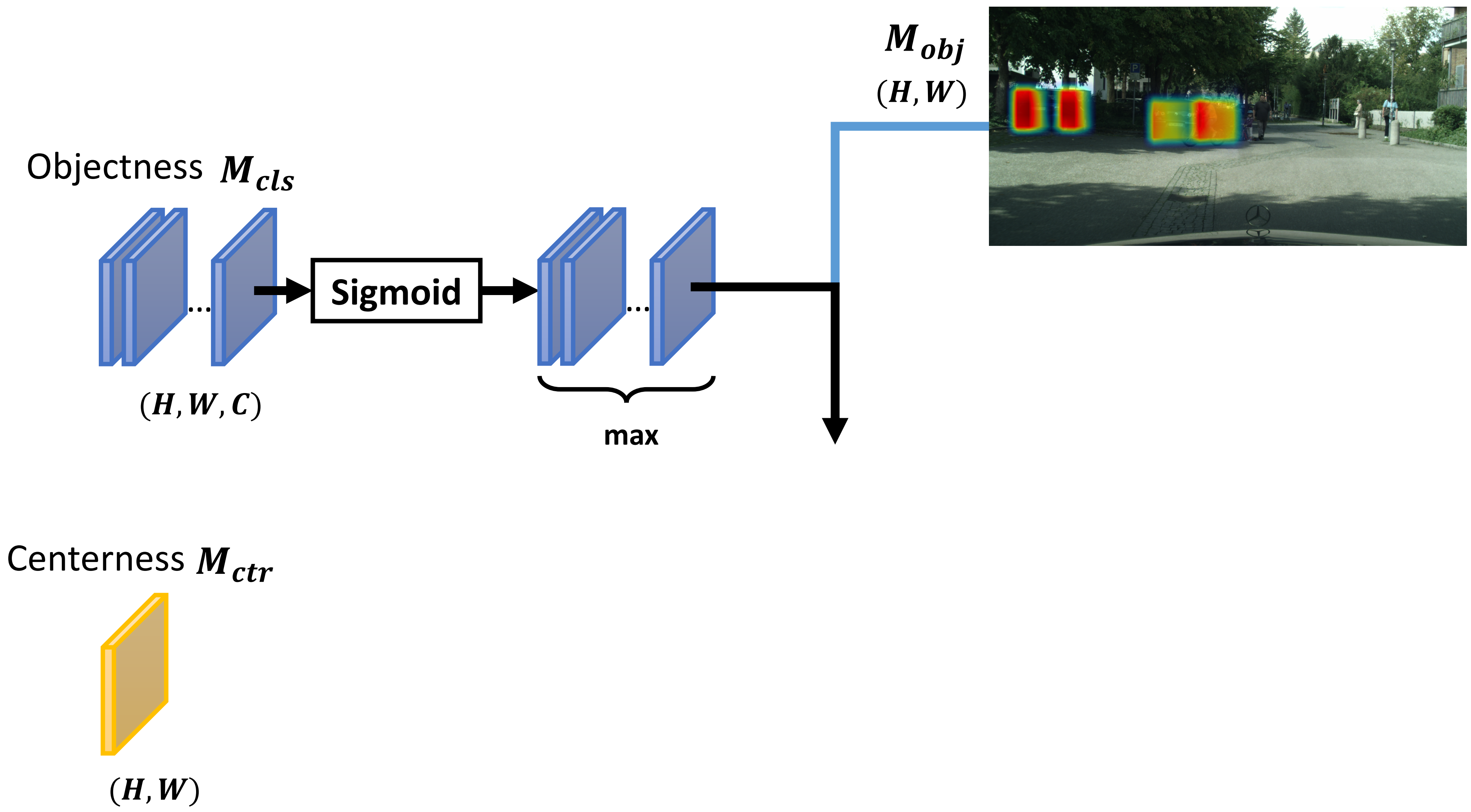
Approach

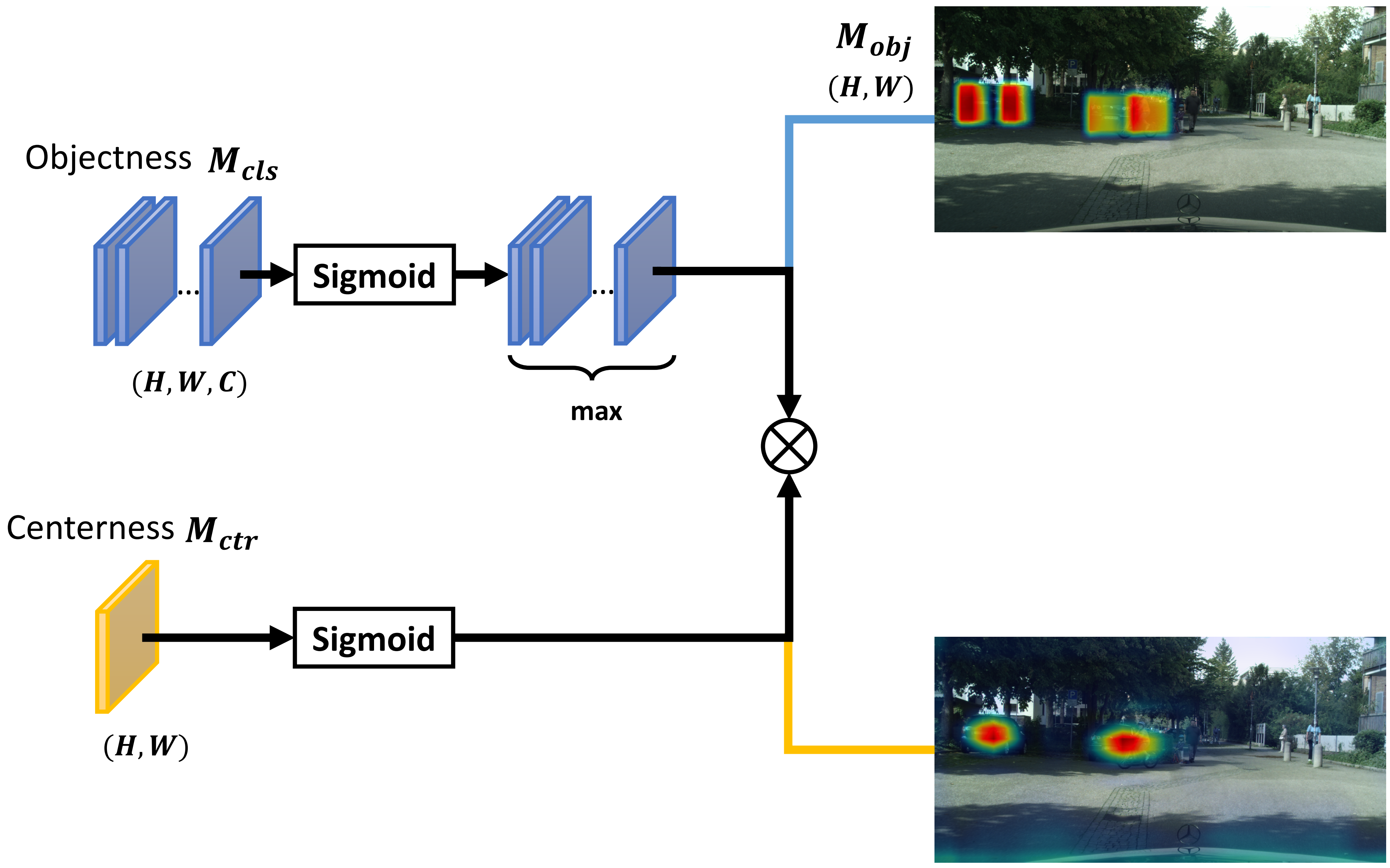


Feature Extractor G
Fully Conv P
Source Domain
Image Feature F_s
BBox Prediction
Classification Map M^s_{cls}
Centerness Map M^s_{ctr}
$\mathcal{L}_{det}$
F_s
F_t
Domain Prediction $\mathcal{L}_{GA}$
Global Discriminator D_{GA}
$(F_s, M^s_{cls}, M^s_{ctr})$
$(F_t, M^t_{cls}, M^t_{ctr})$
Domain Prediction $\mathcal{L}_{CA}$
Center-aware Discriminator D_{CA}
Target Domain
Image Feature F_t
BBox Prediction
Classification Map M^t_{cls}
Centerness Map M^t_{ctr}

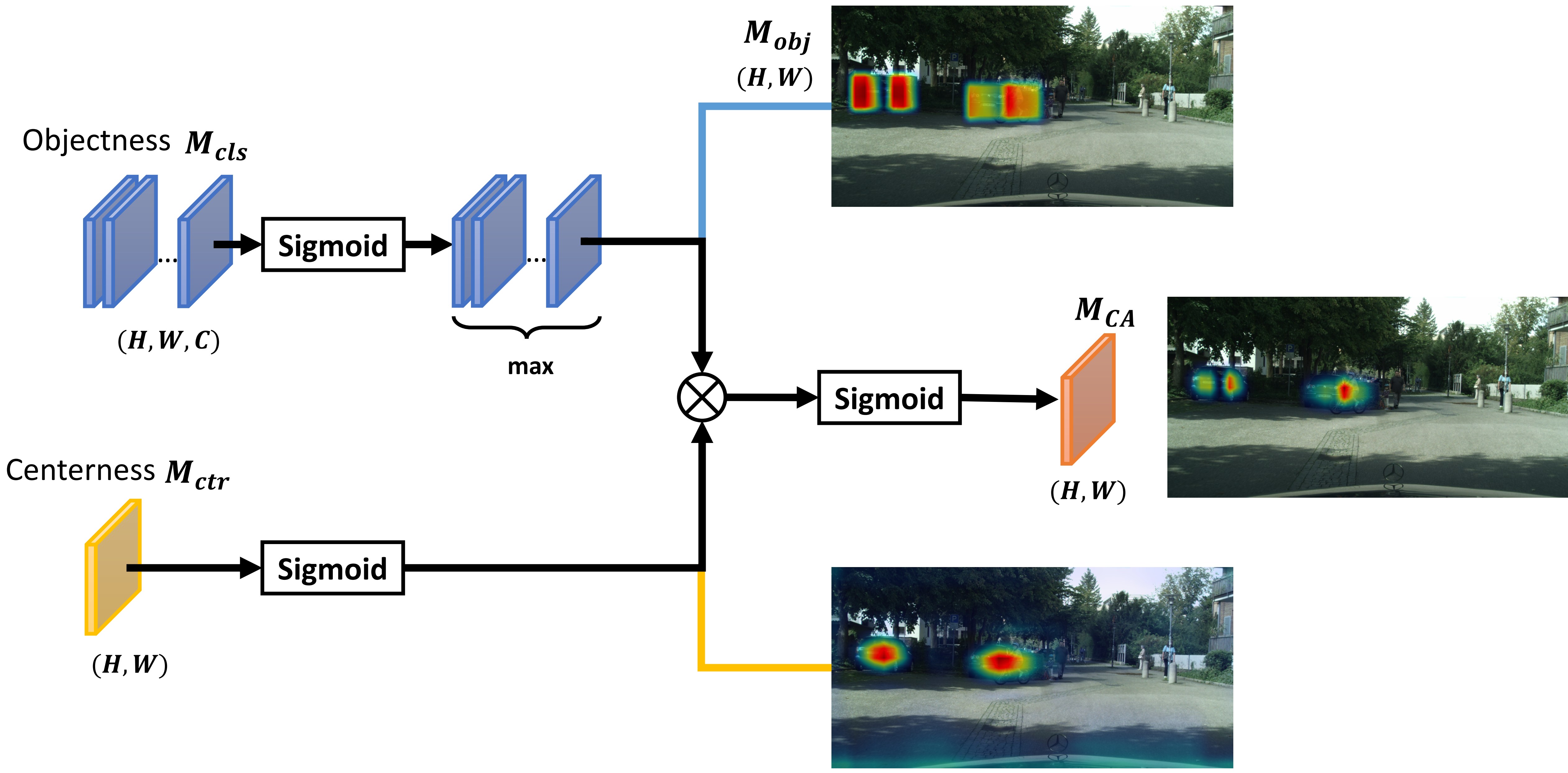


Objectness $\boldsymbol{M_{cls}}$
$(\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{C})$
Centerness $\boldsymbol{M_{ctr}}$
$(\boldsymbol{H}, \boldsymbol{W})$


Objectness M_{cls}
Sigmoid
(H, W, C)
max
M_{obj}
(H, W)
Centerness M_{ctr}
(H, W)



Objectness M_{cls}
(H, W, C)
Sigmoid
max
M_{obj}
(H, W)
Centerness M_{ctr}
(H, W)
Sigmoid



M_{obj}
(H,W)
Objectness M_{cls}
Sigmoid
(H,W,C)
max
M_{CA}
Sigmoid
(H,W)
Centerness M_{ctr}
Sigmoid
(H,W)

Experimental Results

		Cityscapes → Foggy Cityscapes								
Method	Backbone	person	rider	car	truck	bus	train	mbike	bicycle	$mAP^r_{0.5}$
Baseline (F-RCNN)	VGG-16	17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
DAF [2] CVPR'18	VGG-16	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
SC-DA [41] CVPR'19	VGG-16	33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.8
MAF [14] ICCV'19	VGG-16	28.2	39.5	43.9	23.8	39.9	33.3	**29.2**	33.9	34.0
SW-DA [32] CVPR'19	VGG-16	29.9	**42.3**	43.5	24.5	36.2	32.6	30.0	35.3	34.3
DAM [22] CVPR'19	VGG-16	30.8	40.5	44.3	**27.2**	38.4	**34.5**	28.4	32.2	34.6
Ours (w/o adapt.)	VGG-16	30.5	23.9	34.2	5.8	11.1	5.1	10.6	26.1	18.4
Ours (GA)	VGG-16	38.7	36.1	53.1	21.9	35.4	25.7	20.6	33.9	33.2
Ours (CA)	VGG-16	41.3	38.2	56.5	21.1	33.4	26.9	23.8	32.6	34.2
Ours (GA+CA)	VGG-16	**41.9**	38.7	**56.7**	22.6	**41.5**	26.8	24.6	**35.5**	**36.0**
Oracle		47.4	40.8	66.8	27.2	48.2	32.4	31.2	38.3	41.5
Ours (w/o adapt.)	ResNet-101	33.8	34.8	39.6	18.6	27.9	6.3	18.2	25.5	25.6
Ours (GA)	ResNet-101	39.4	41.1	54.6	23.8	42.5	31.2	25.1	35.1	36.6
Ours (CA)	ResNet-101	40.4	**44.9**	**57.9**	24.6	**49.6**	32.1	25.2	34.3	38.6
Ours (GA+CA)	ResNet-101	**41.5**	43.6	57.1	**29.4**	44.9	**39.7**	**29.0**	**36.1**	**40.2**
Oracle		44.7	43.9	64.7	31.5	48.8	44.0	31.0	36.7	43.2

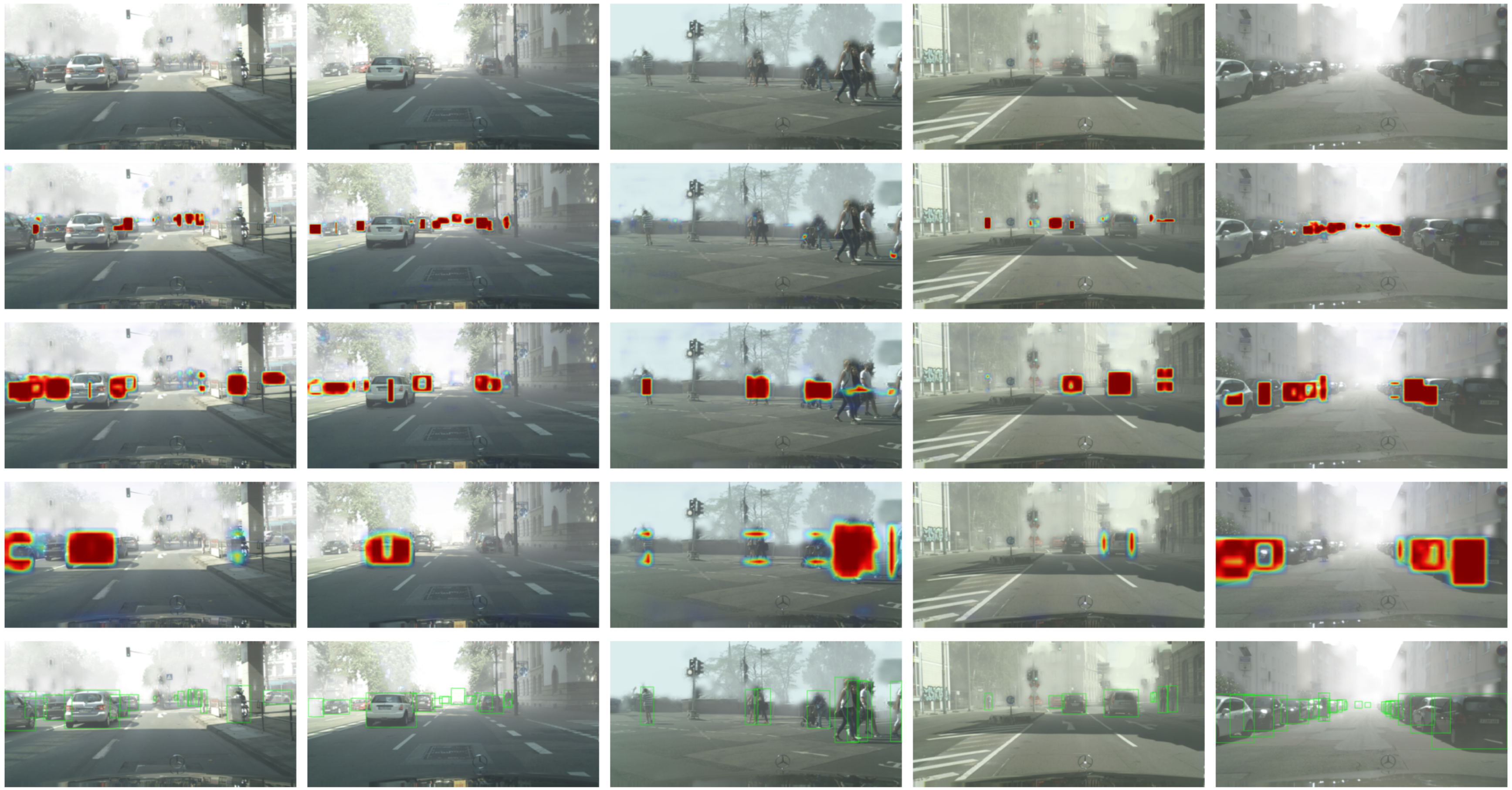
Experimental Results

		Cityscapes → Foggy Cityscapes								
Method	Backbone	person	rider	car	truck	bus	train	mbike	bicycle	$mAP^{r}_{0.5}$
Baseline (F-RCNN)	VGG-16	17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
DAF [2] CVPR'18		25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
SC-DA [41] CVPR'19		33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.8
MAF [14] ICCV'19		28.2	39.5	43.9	23.8	39.9	33.3	**29.2**	33.9	34.0
SW-DA [32] CVPR'19		29.9	**42.3**	43.5	24.5	36.2	32.6	30.0	35.3	34.3
DAM [22] CVPR'19		30.8	40.5	44.3	**27.2**	38.4	**34.5**	28.4	32.2	34.6
Ours (w/o adapt.)		30.5	23.9	34.2	5.8	11.1	5.1	10.6	26.1	18.4
Ours (GA)		38.7	36.1	53.1	21.9	35.4	25.7	20.6	33.9	33.2
Ours (CA)		41.3	38.2	56.5	21.1	33.4	26.9	23.8	32.6	34.2
Ours (GA+CA)		**41.9**	38.7	**56.7**	22.6	**41.5**	26.8	24.6	**35.5**	**36.0**
Oracle		47.4	40.8	66.8	27.2	48.2	32.4	31.2	38.3	41.5
Ours (w/o adapt.)	ResNet-101	33.8	34.8	39.6	18.6	27.9	6.3	18.2	25.5	25.6
Ours (GA)		39.4	41.1	54.6	23.8	42.5	31.2	25.1	35.1	36.6
Ours (CA)		40.4	**44.9**	**57.9**	24.6	**49.6**	32.1	25.2	34.3	38.6
Ours (GA+CA)		**41.5**	43.6	57.1	**29.4**	44.9	**39.7**	**29.0**	**36.1**	**40.2**
Oracle		44.7	43.9	64.7	31.5	48.8	44.0	31.0	36.7	43.2

Experimental Results

		Cityscapes → Foggy Cityscapes								
Method	Backbone	person	rider	car	truck	bus	train	mbike	bicycle	$mAP^r_{0.5}$
Baseline (F-RCNN)	VGG-16	17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
DAF [2] CVPR'18	VGG-16	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
SC-DA [41] CVPR'19	VGG-16	33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.8
MAF [14] ICCV'19	VGG-16	28.2	39.5	43.9	23.8	39.9	33.3	**29.2**	33.9	34.0
SW-DA [32] CVPR'19	VGG-16	29.9	**42.3**	43.5	24.5	36.2	32.6	30.0	35.3	34.3
DAM [22] CVPR'19	VGG-16	30.8	40.5	44.3	**27.2**	38.4	**34.5**	28.4	32.2	34.6
Ours (w/o adapt.)	VGG-16	30.5	23.9	34.2	5.8	11.1	5.1	10.6	26.1	18.4
Ours (GA)	VGG-16	38.7	36.1	53.1	21.9	35.4	25.7	20.6	33.9	33.2
Ours (CA)	VGG-16	41.3	38.2	56.5	21.1	33.4	26.9	23.8	32.6	34.2
Ours (GA+CA)	VGG-16	**41.9**	38.7	**56.7**	22.6	**41.5**	26.8	24.6	**35.5**	**36.0**
Oracle		47.4	40.8	66.8	27.2	48.2	32.4	31.2	38.3	41.5
Ours (w/o adapt.)	ResNet-101	33.8	34.8	39.6	18.6	27.9	6.3	18.2	25.5	25.6
Ours (GA)	ResNet-101	39.4	41.1	54.6	23.8	42.5	31.2	25.1	35.1	36.6
Ours (CA)	ResNet-101	40.4	**44.9**	**57.9**	24.6	**49.6**	32.1	25.2	34.3	38.6
Ours (GA+CA)	ResNet-101	**41.5**	43.6	57.1	**29.4**	44.9	**39.7**	**29.0**	**36.1**	**40.2**
Oracle		44.7	43.9	64.7	31.5	48.8	44.0	31.0	36.7	43.2

Experimental Results (Cityscapes → Foggy Cityscapes)



Input Image

Center-aware Map (F_3)

Center-aware Map (F_4)

Center-aware Map (F_5)

Detection Results

Experimental Results (Sim10k → Cityscapes)

Input Image

Center-aware Map (F_3)

Center-aware Map (F_4)

Center-aware Map (F_5)

Detection Results

Concluding Remarks

- Use fundamental tools for new tasks
  - Adversarial learning
  - Structured output
  - Enforcing constraints
  - Incremental learning
  - Mining high-confidence samples
- Thanks to all collaborators: Yi-Hsuan Tsai, Jia-Bin Huang, Yen-Yu Lin, Wei-Chih Hung, Hung-Yu Tseng, Hsin-Ying Lee, Cheng-Chun Hsu, Chun-Han Yao, Jongbin Ryu, Jongwoo Lim, GiTaek Kwon, Samuel Schulter, Kihyuk Sohn, Manmohan Chandraker, Han-Kai Hsu, Yan-Ting Liou, Maneesh Singh, ...