Domain Adaptation for Semantic Segmentation

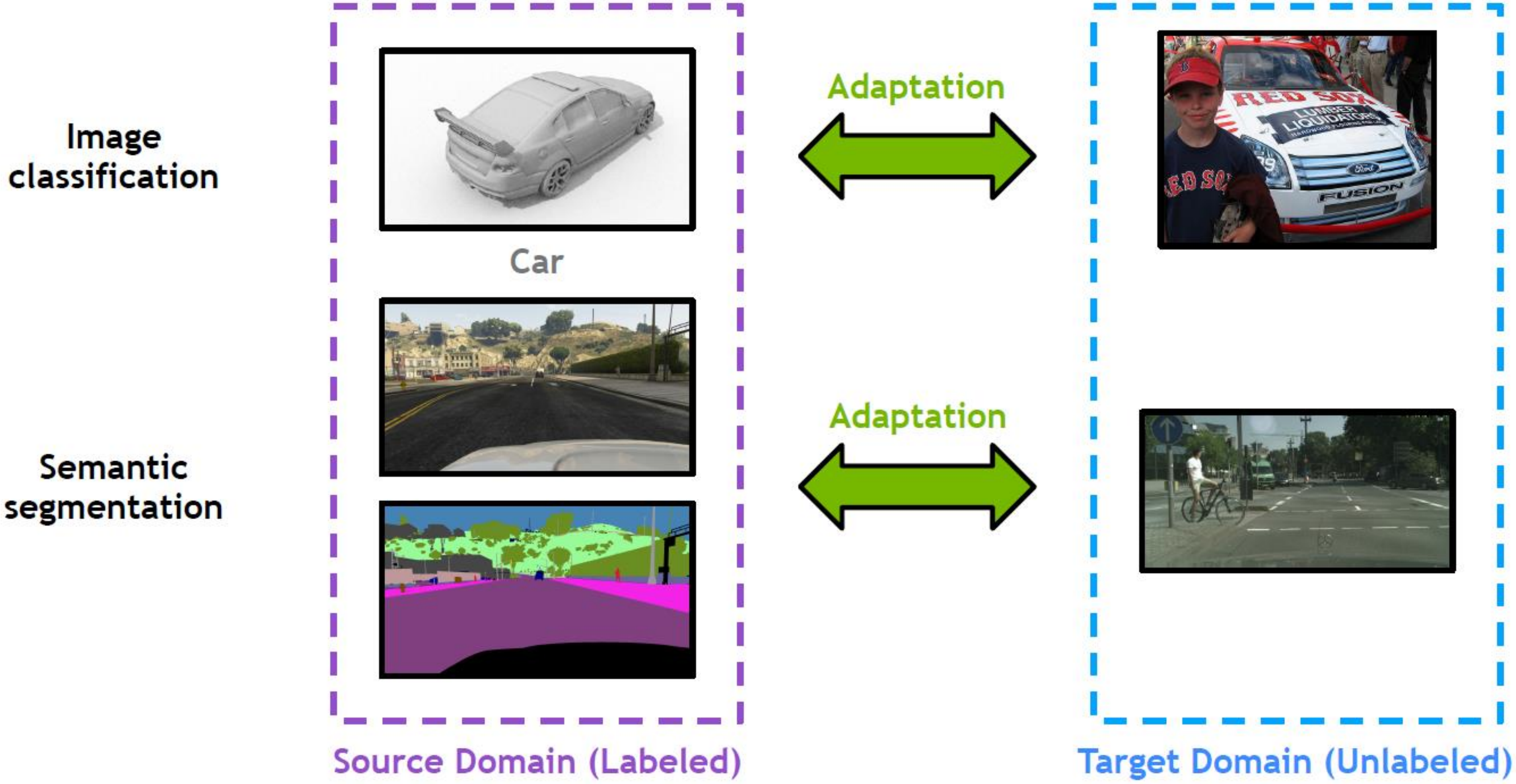


Image classification

Semantic segmentation

Car

Adaptation

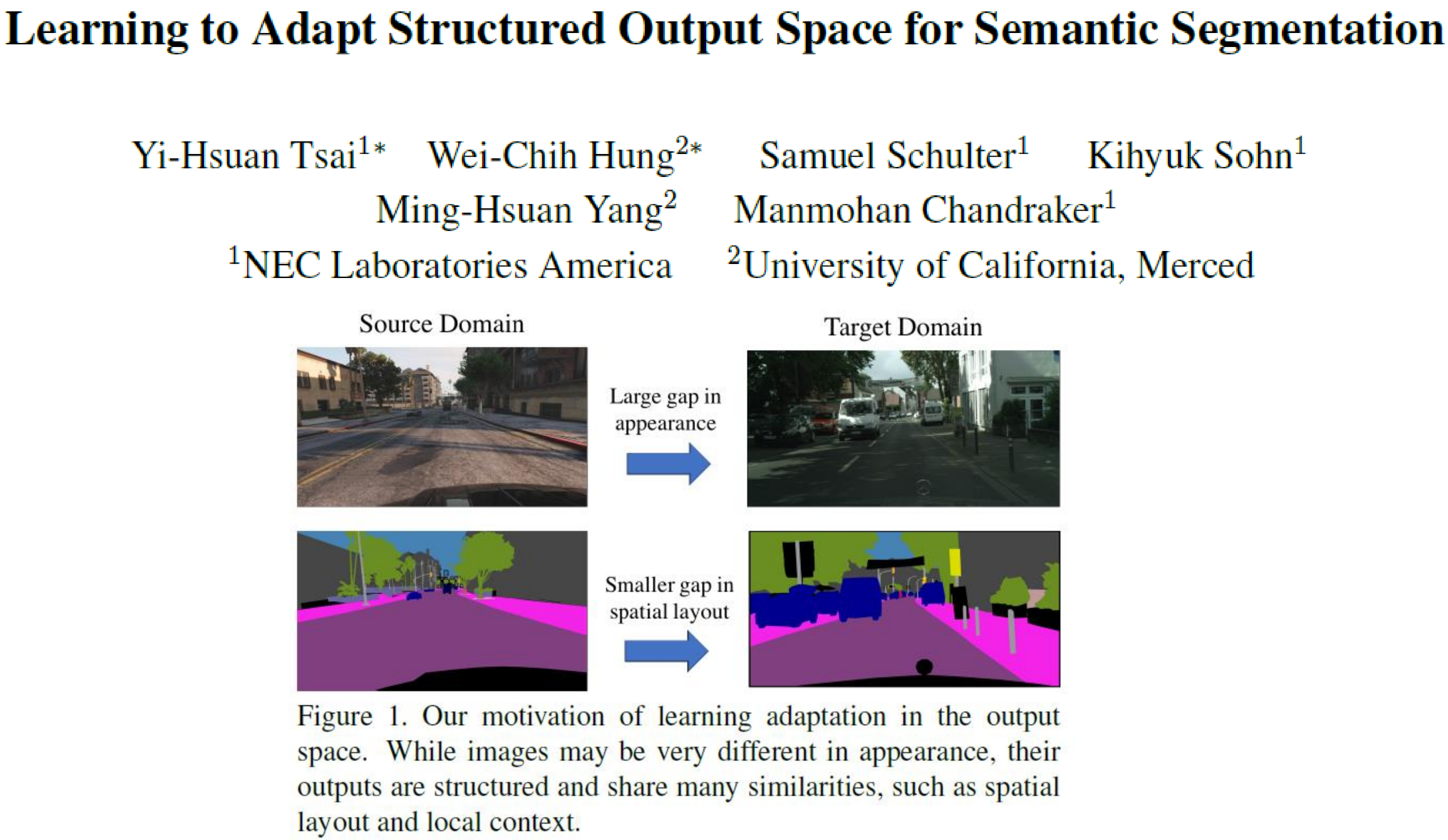
Adaptation

Source Domain (Labeled)

Target Domain (Unlabeled)

Yichao Wang
2020.4.3

AdaptSegnet



Learning to Adapt Structured Output Space for Semantic Segmentation

Yi-Hsuan Tsai[1]* Wei-Chih Hung[2]* Samuel Schulter[1] Kihyuk Sohn[1]
Ming-Hsuan Yang[2] Manmohan Chandraker[1]
[1]NEC Laboratories America [2]University of California, Merced

Source Domain
Target Domain
Large gap in appearance
Smaller gap in spatial layout

Figure 1. Our motivation of learning adaptation in the output space. While images may be very different in appearance, their outputs are structured and share many similarities, such as spatial layout and local context.

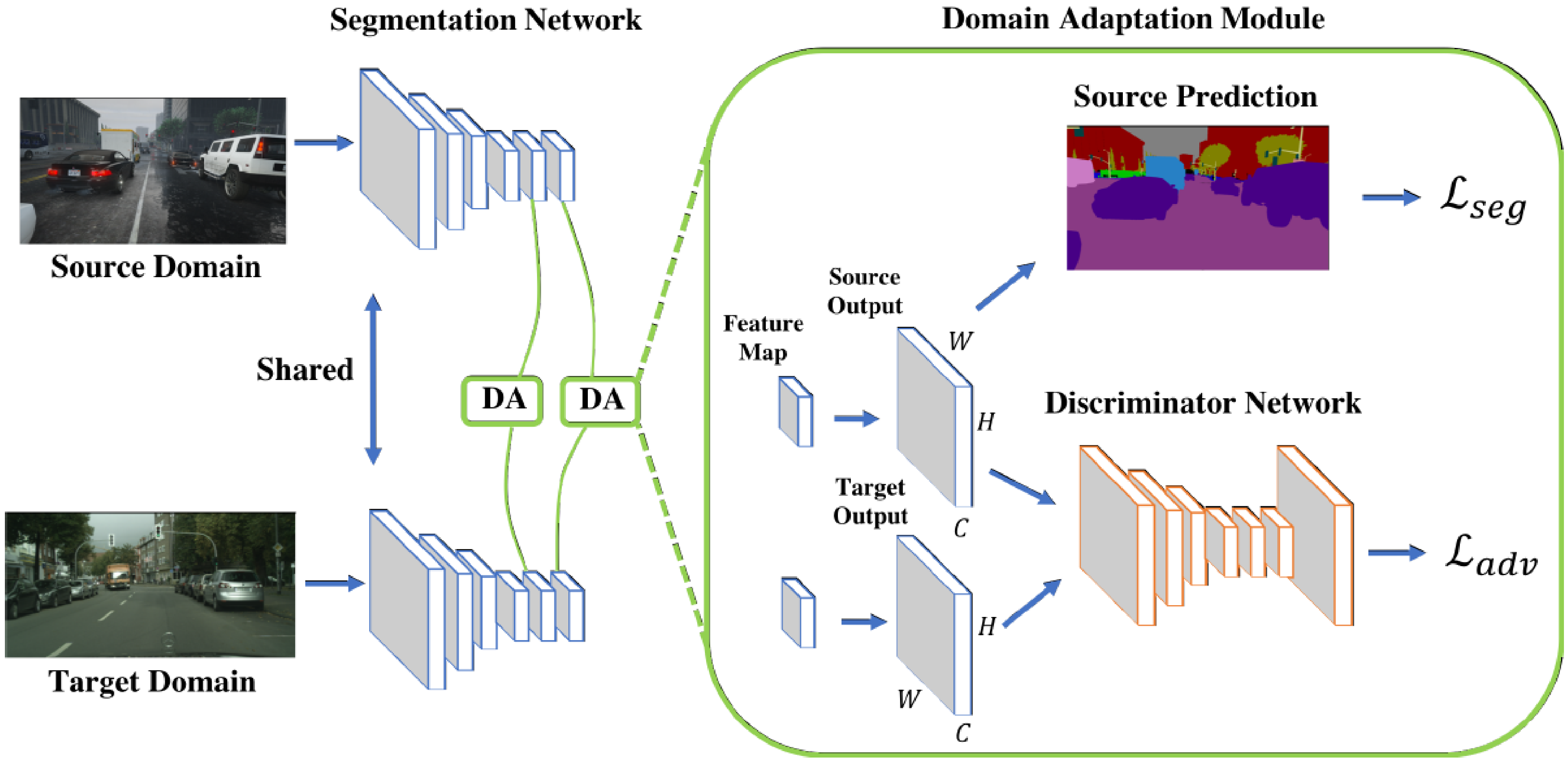
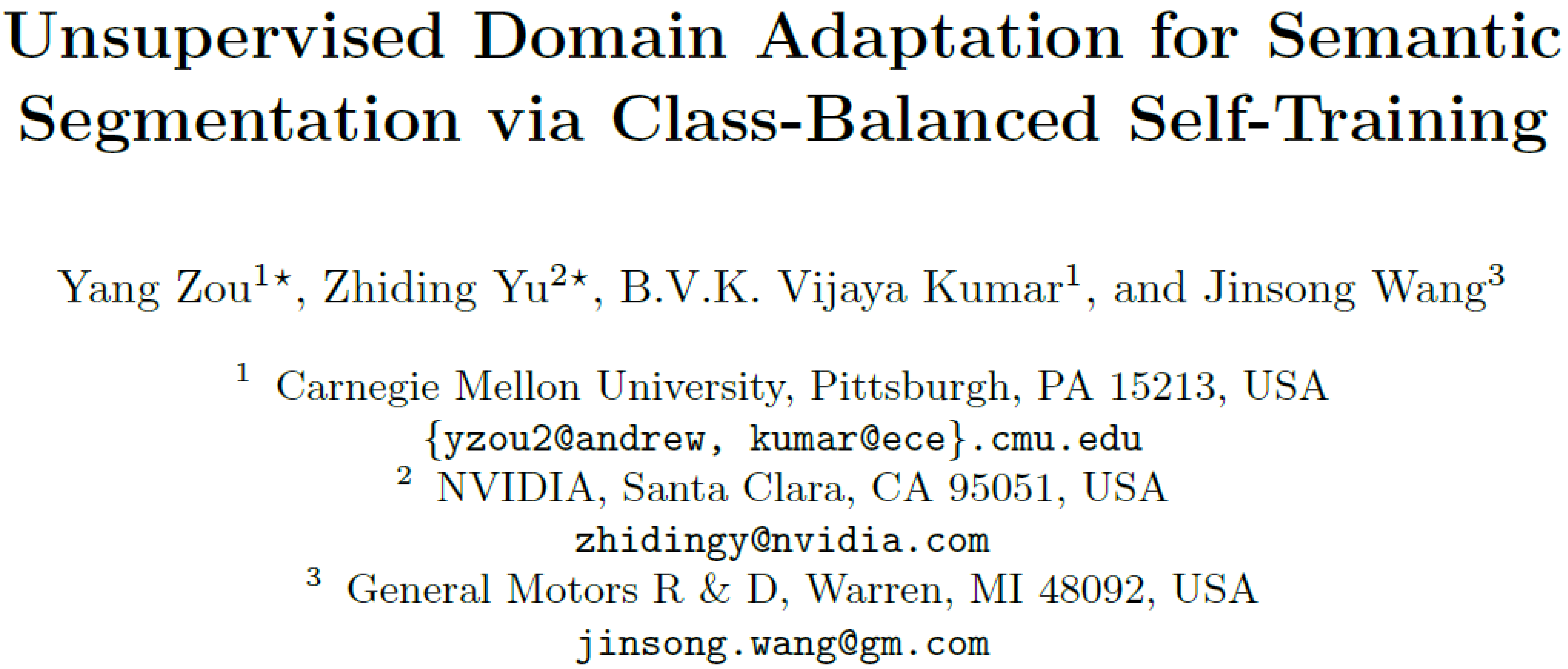


Figure 2. Algorithmic overview. Given images with the size H by W in source and target domains, we pass them through the segmentation network to obtain output predictions. For source predictions with C categories, a segmentation loss is computed based on the source ground truth. To make target predictions closer to the source ones, we utilize a discriminator to distinguish whether the input is from the source or target domain. Then an adversarial loss is calculated on the target prediction and is back-propagated to the segmentation network. We call this process as one adaptation module, and we illustrate our proposed multi-level adversarial learning by adopting two adaptation modules at two different levels here.

Self-Training



Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training

Yang Zou[1⋆], Zhiding Yu[2⋆], B.V.K. Vijaya Kumar[1], and Jinsong Wang[3]

[1] Carnegie Mellon University, Pittsburgh, PA 15213, USA
{yzou2@andrew, kumar@ece}.cmu.edu
[2] NVIDIA, Santa Clara, CA 95051, USA
zhidingy@nvidia.com
[3] General Motors R & D, Warren, MI 48092, USA
jinsong.wang@gm.com

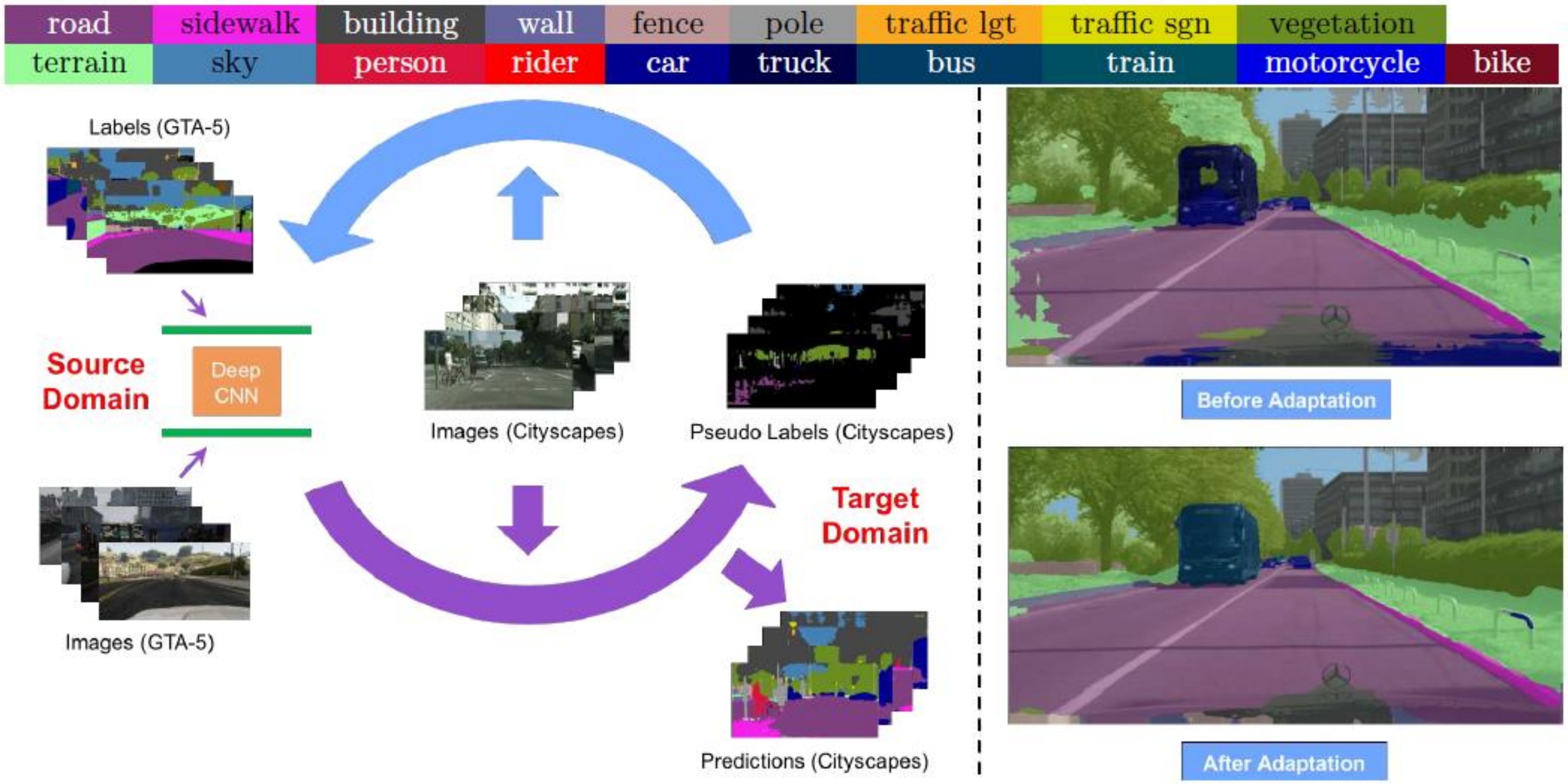
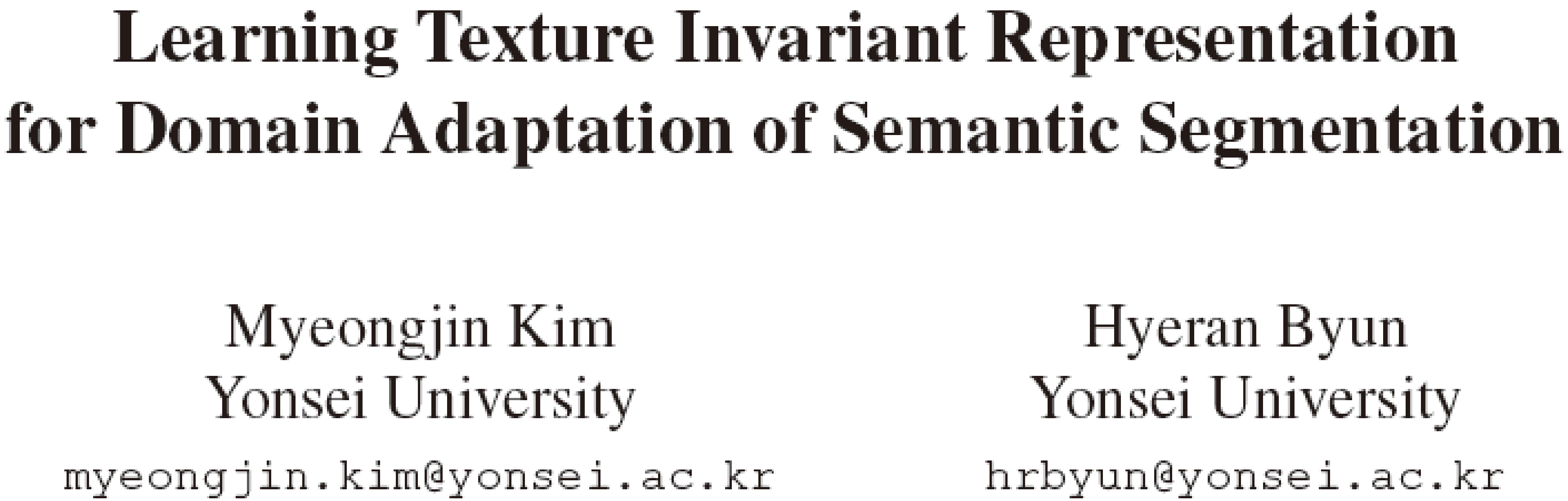


Fig. 1: Illustration of the proposed itertive self-training framework for unsupervised domain adaptation. Left: algorithm workflow. Right figure: semantic segmentation results on Cityscapes before and after adaptation.

Recent works



Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation

Myeongjin Kim
Yonsei University
myeongjin.kim@yonsei.ac.kr

Hyeran Byun
Yonsei University
hrbyun@yonsei.ac.kr

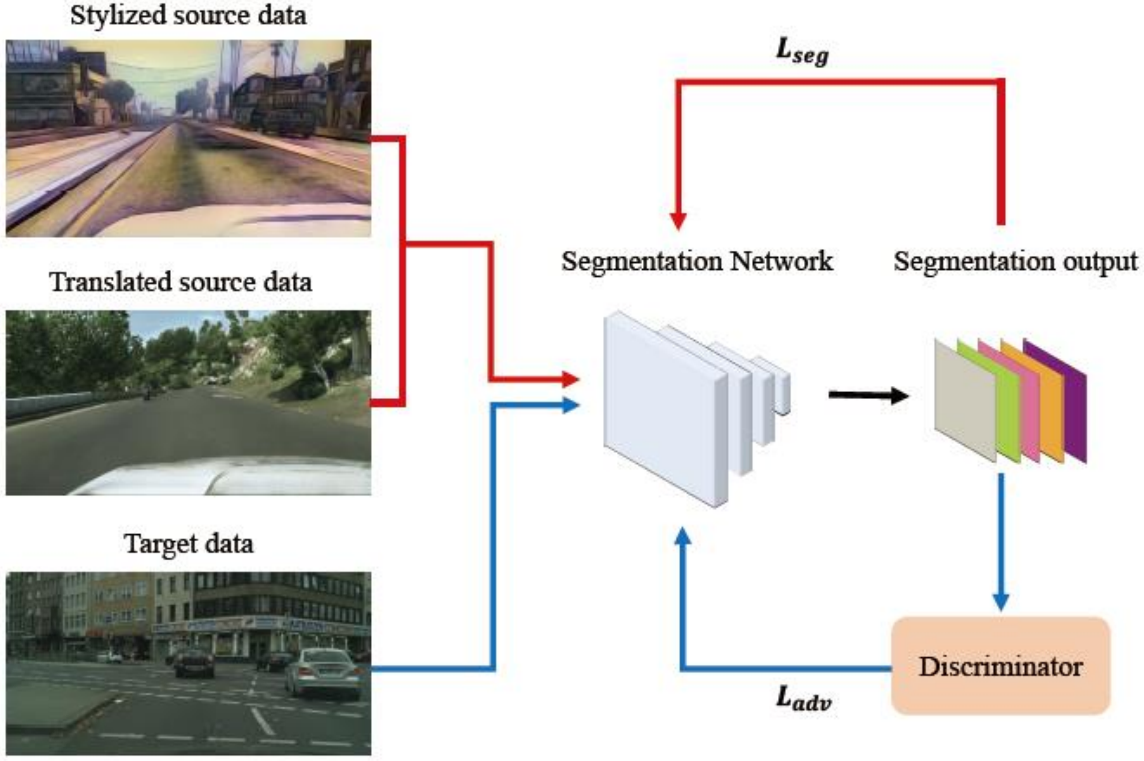


Figure 1: Process of learning texture-invariant representation. We consider both the stylized image and the translated image as the source image. The red line indicates the flow of the source image and the blue line indicates the flow of the target image. By segmentation loss of the stylized source data, the model learns texture-invariant representation. By adversarial loss, the model reduces the distribution gap in feature space.

- In this paper, considering the fundamental difference between the two domains as the texture.
- Our purpose is, using **various textures as a regularizer** preventing a model from overfitting to one specific texture, to make the segmentation model learn **texture-invariant representation**.

- First, we diversity the texture of synthetic images using a style transfer algorithm **Style-swap**.
- Then, we fine-tune the model with **self-training** to get direct supervision of the target texture.

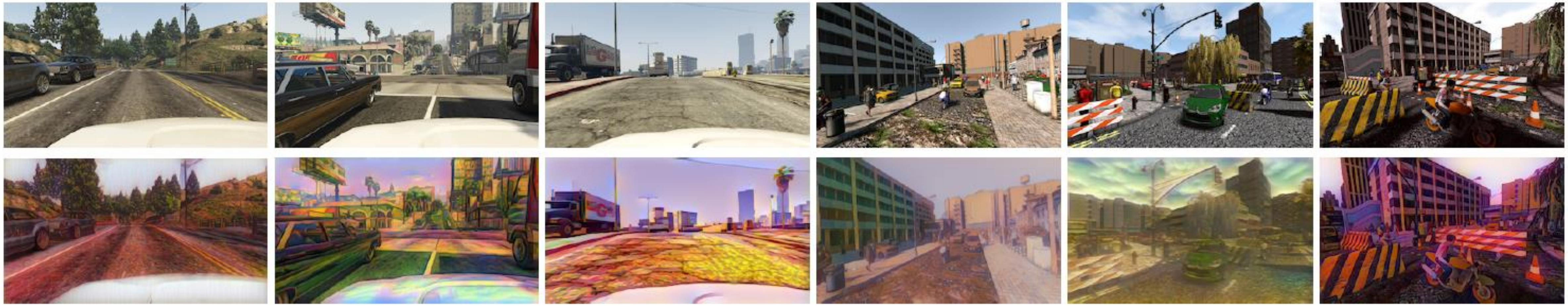


Figure 4: Examples of original images and stylized images.

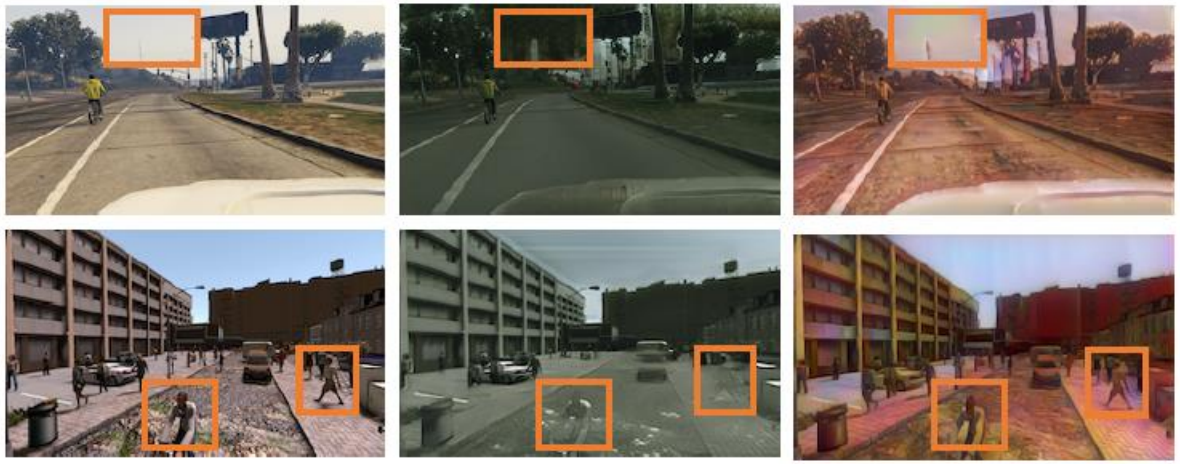


Figure 5: Inappropriate generation of CycleGAN. Original images (first column), generated images by CycleGAN (second column) and Style-swap (third column).

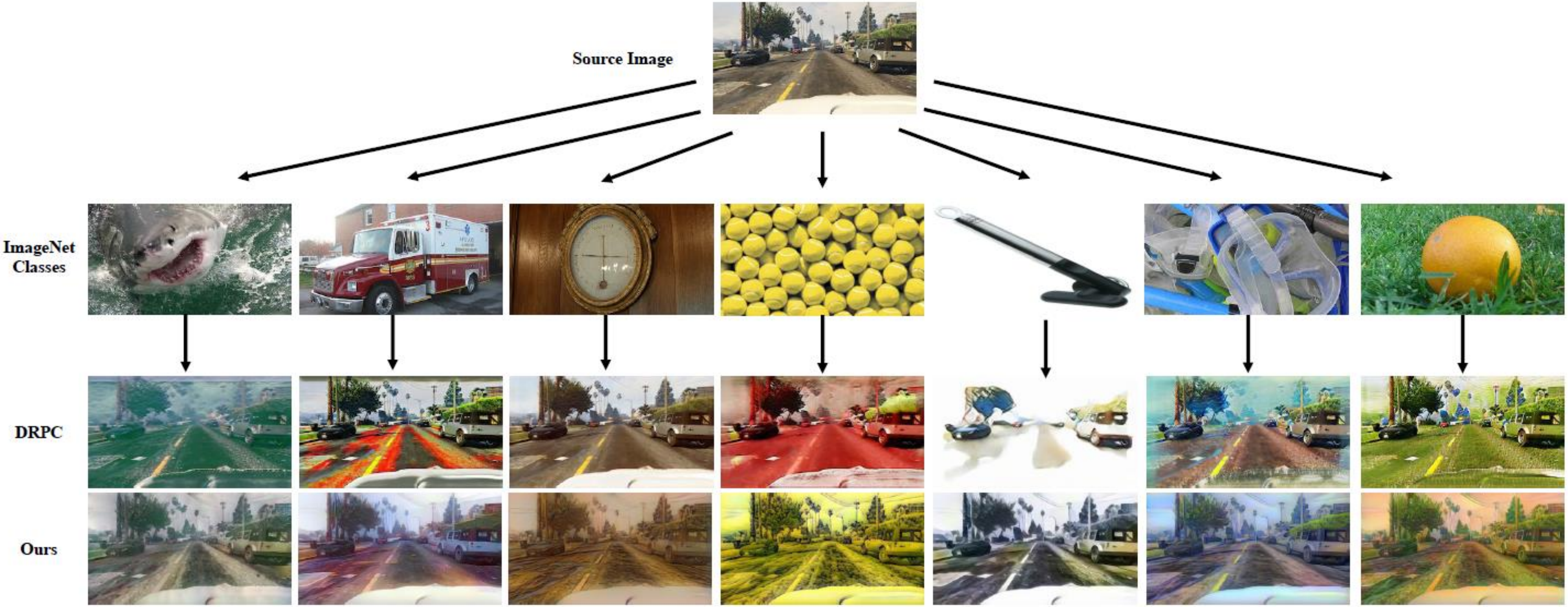


Figure 6: Stylization comparison with DRPC.

Table 1: Results on GTA5 to Cityscapes.

		GTA5 → Cityscapes																			
Base Model	Method	road	side.	buil.	wall	fence	pole	t-light	t-sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	motor	bike	mIoU
ResNet101	AdaptSegNet[23]	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
	CLAN[18]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	**6.7**	31.9	31.4	43.2
	ADVENT[25]	87.6	21.4	82.0	34.8	26.2	28.5	35.6	23.0	84.5	35.1	76.2	58.6	30.7	**84.8**	34.2	43.4	0.4	28.4	35.2	44.8
	BDL[16]	91.0	44.7	84.2	**34.6**	27.6	30.2	36.0	**36.0**	85.0	**43.6**	83.0	58.6	**31.6**	83.3	**35.3**	**49.7**	3.3	28.8	35.6	48.5
	SIBAN[17]	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
	AdaptPatch[24]	92.3	51.9	82.1	29.2	25.1	24.5	33.8	33.0	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5
	MaxSquare[3]	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	**85.3**	39.4	78.2	**63.0**	22.9	84.6	36.4	43.0	5.5	34.7	33.5	46.4
	Ours	**92.9**	**55.0**	**85.3**	34.2	**31.1**	**34.9**	**40.7**	34.0	85.2	40.1	**87.1**	61.0	31.1	82.5	32.3	42.9	0.3	**36.4**	**46.1**	**50.2**
VGG16	AdaptSegNet[23]	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
	CLAN[18]	88.0	30.6	79.2	23.4	20.5	26.1	23.0	14.8	81.6	34.5	72.0	45.8	7.9	80.5	**26.6**	29.9	0.0	10.7	0.0	36.6
	ADVENT[25]	86.8	28.5	78.1	27.6	24.2	20.7	19.3	8.9	78.8	29.3	69.0	47.9	5.9	79.8	25.9	**34.1**	0.0	11.3	0.3	35.6
	BDL[16]	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
	SIBAN[17]	83.4	13.0	77.8	20.4	17.5	24.6	22.8	9.6	81.3	29.6	77.3	42.7	10.9	76.0	22.8	17.9	5.7	14.2	2.0	34.2
	AdaptPatch[24]	87.3	35.7	79.5	32.0	14.5	21.5	24.8	13.7	80.4	32.0	70.5	50.5	16.9	81.0	20.8	28.1	4.1	15.5	4.1	37.5
	DRPC[28]	84.6	31.5	76.3	25.4	17.2	28.2	21.5	13.7	80.7	26.8	74.9	47.5	15.8	77.1	22.2	22.7	1.7	8.9	9.7	36.1
	Ours	**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**

Table 4: Ablation study on Stage 1.

GTA5 → Cityscapes	
method	mIoU
Original source only	36.6
DCAN [27]	38.5
Translated source only	41.0
DLOW [10]	42.3
Stylized source only	**42.5**
Original source + Adv loss [23]	41.4
Translated source + Adv loss [16]	42.7
Stylized source + Adv loss	43.2
Stylized/translated source + Adv loss	**44.6**

Table 5: Ablation study on Stage 2. In Stage 2-X, X means the number of iteration of self training.

GTA5 → Cityscapes	
method	mIoU
Stage 1	44.6
Stage 2-1	48.6
Stage 2-2	50.2
Stage 2-3	50.2

Table 6: Results on original and noisy validation set.

Method	AdaptSegNet[23]	Stylized source only
Original	42.4	42.5
Gaussian	22.2	**35.1**
Impulse	20.9	**32.6**
Shot	24.9	**38.2**
Speckle	32.5	**41.1**

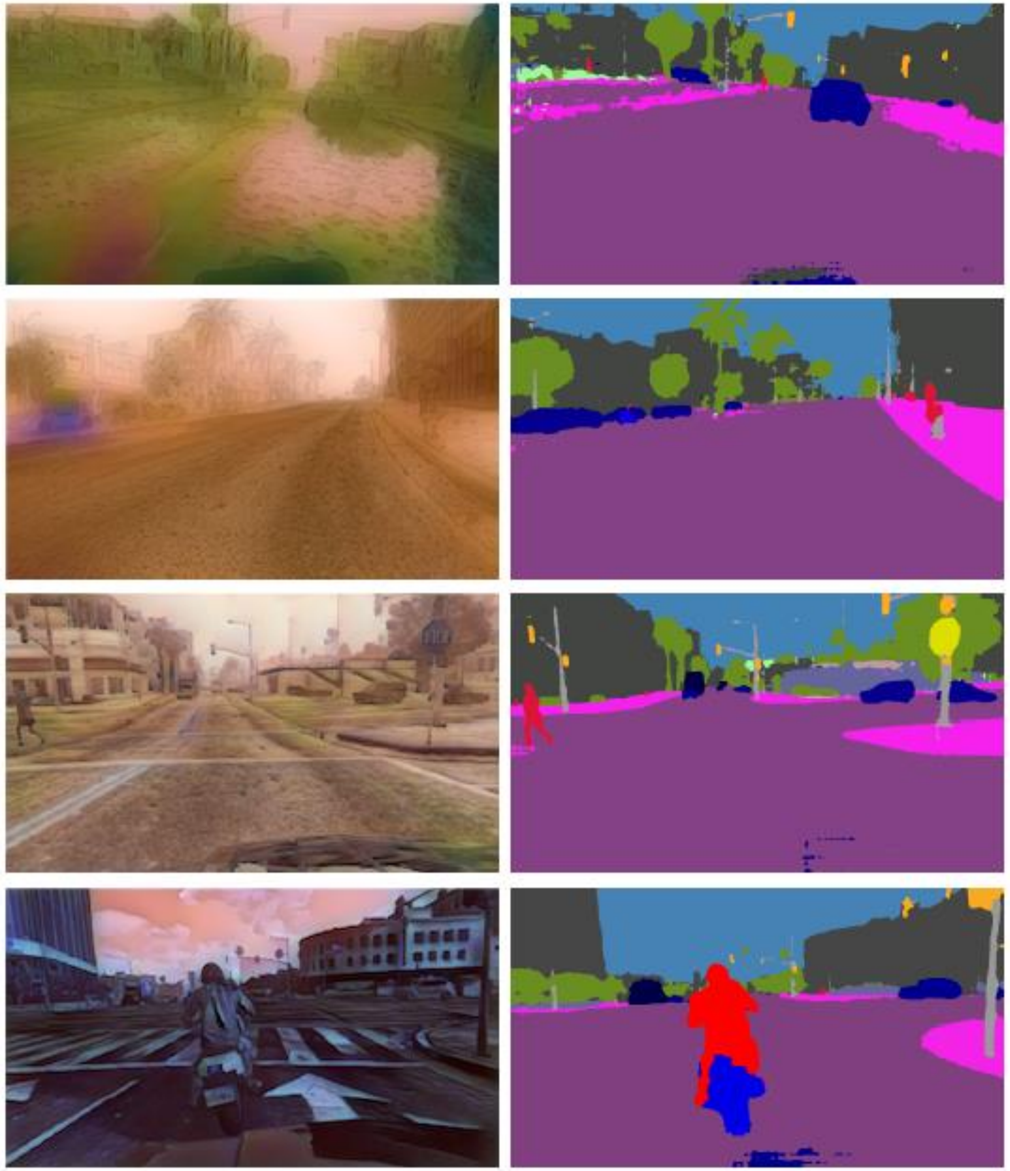
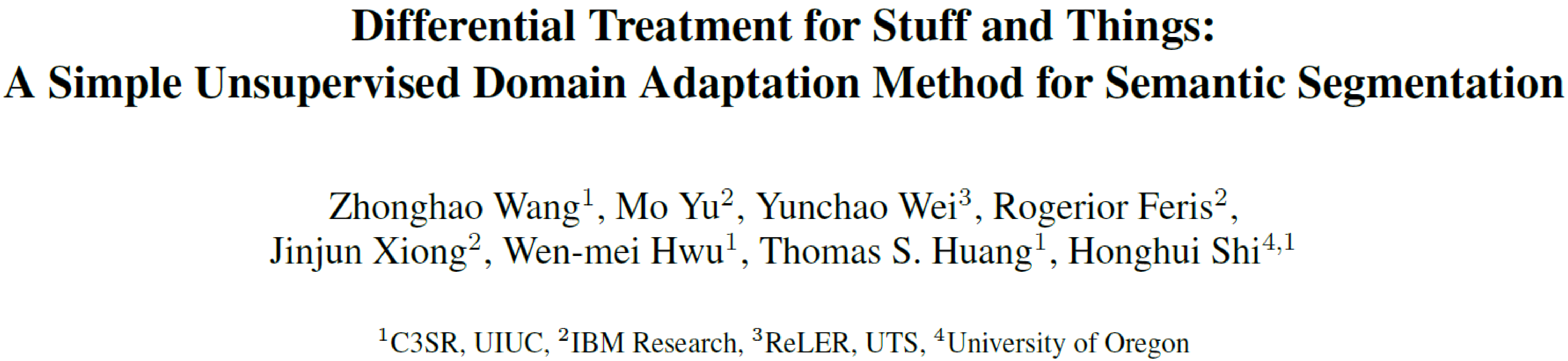


Figure 9: Results on images with various texture. Images from the Stylized GTA5 (left column) and segmentation results (right column).

Recent works



Differential Treatment for Stuff and Things: A Simple Unsupervised Domain Adaptation Method for Semantic Segmentation

Zhonghao Wang[1], Mo Yu[2], Yunchao Wei[3], Rogerior Feris[2],
Jinjun Xiong[2], Wen-mei Hwu[1], Thomas S. Huang[1], Honghui Shi[4,1]

[1]C3SR, UIUC, [2]IBM Research, [3]ReLER, UTS, [4]University of Oregon

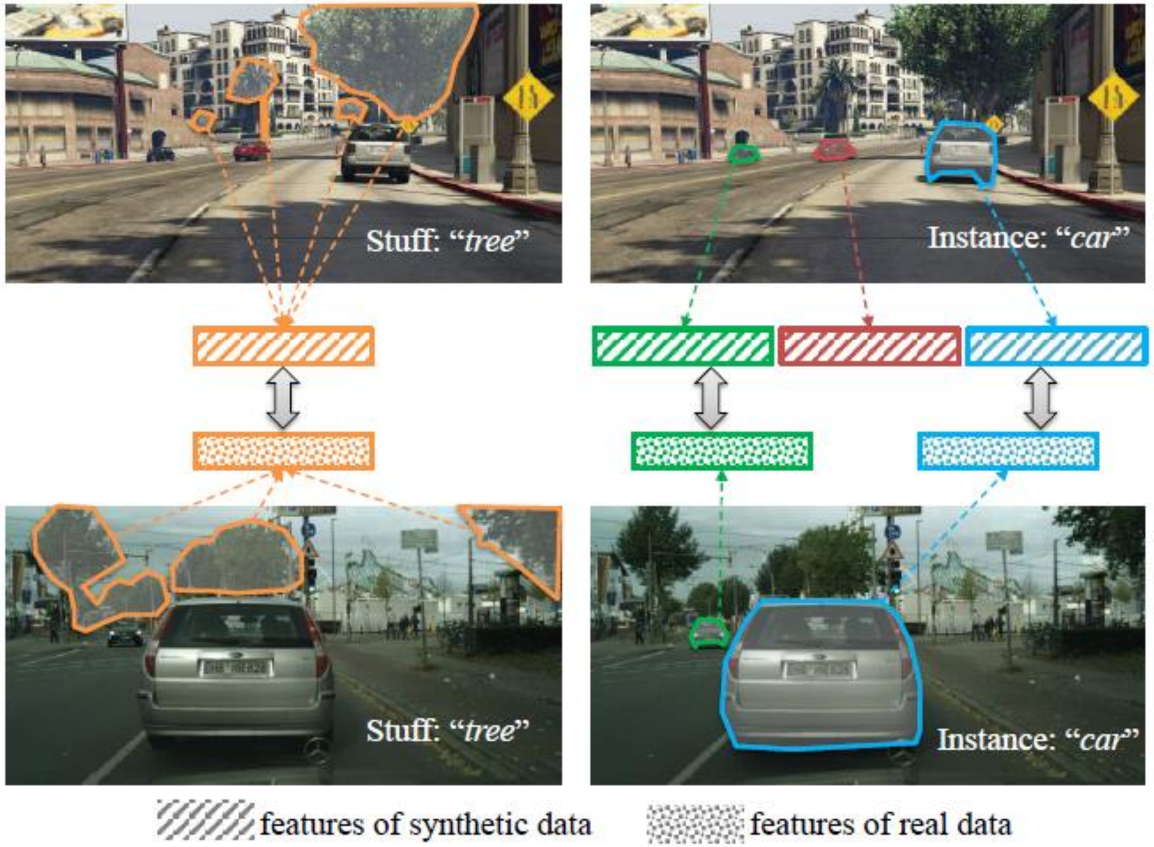


Figure 1. Illustration of the proposed Stuff Instance Matching (SIM) structure. By matching the most similar stuff regions and things (i.e., instances) with differential treatment, we can adapt the features more accurately from the source domain to the target domain.

- Based on the observation that stuff categories usually share similar appearances across images of different domains while things (i.e. object instances) have much larger differences.
- 1) for the **stuff** categories, we generate feature representation for **each class** and conduct the alignment operation from the target domain to the source domain;
- 2) for the **thing** categories, we generate feature representation for **each individual instance** and encourage the instance in the target domain to align with the most similar one in the source domain.

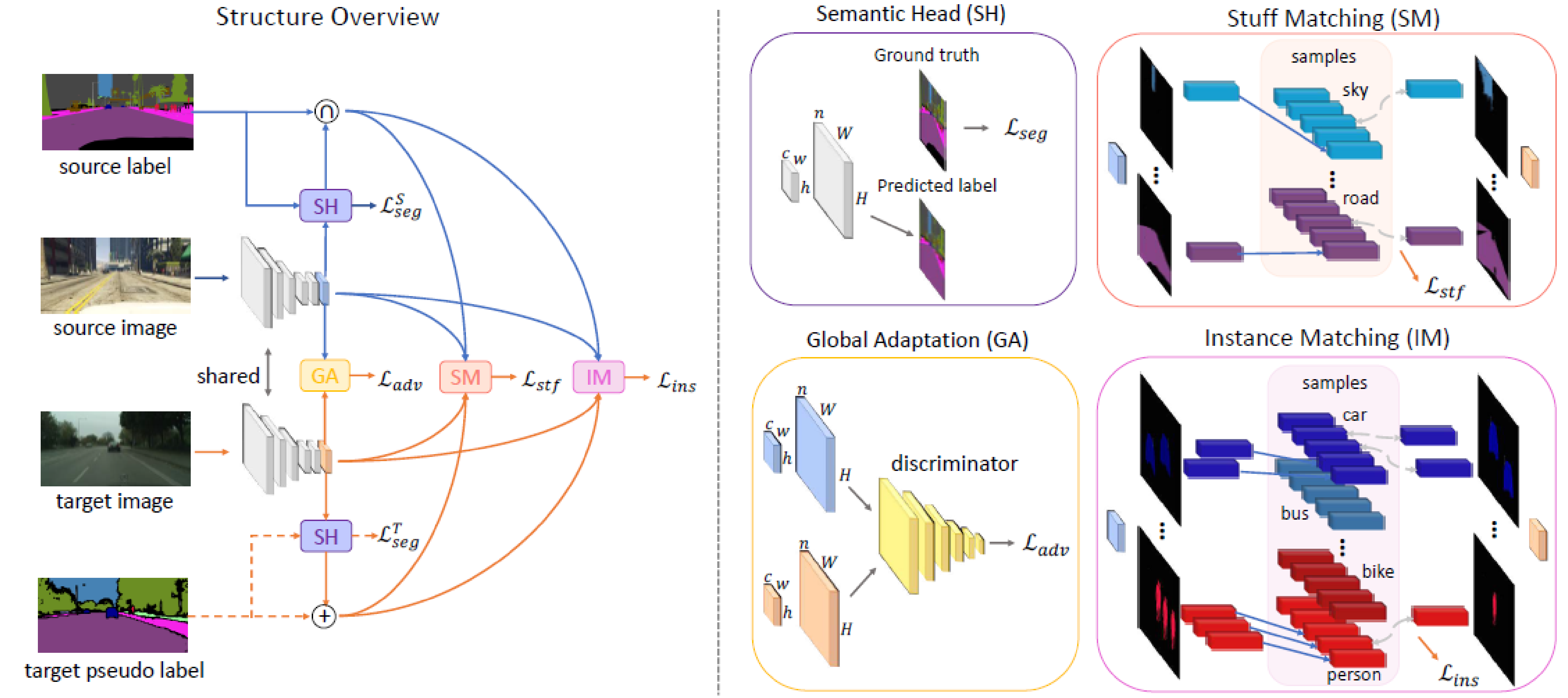


Figure 3. Framework. 1) The overall structure is shown on the left. The solid lines represent the first step training procedure in Eqn (12), and the dash lines along with the solid lines represent the second step training procedure in Eqn (13). The blue lines correspond to the flow direction of the source domain data, and the orange lines correspond to the flow direction of target domain data. $\cap$ is an operation defined in Eqn (4); $+$ is an operation defined in Eqn (11) and is only effective in the second step training procedure. 2) The specific module design is shown on the right. h, w and c represent the height, width and channels for the feature maps; H, W and n represent the height, width and class number for the output maps of the semantic head. For SH, the input ground truth label map supervise the the semantic segmentation task, and the semantic head also generates a predicted label map joining the operations of $\cap$ and $+$. For SM and IM, the grey dash lines represent the matching operation defined in Eqn (6) and (8) respectively.

Table 1. Comparison to the state-of-the-art results of adapting GTA5 to Cityscapes.

GTA5 → Cityscapes																				
Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bike	mIoU
Wu et al.[40]	85.0	30.8	81.3	25.8	21.2	22.2	25.4	26.6	83.4	36.7	76.2	58.9	24.9	80.7	29.5	42.9	2.5	26.9	11.6	41.7
Tsai et al.[37]	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
Saleh et al.[34]	79.8	29.3	77.8	24.2	21.6	6.9	23.5	44.2	80.5	38.0	76.2	52.7	22.2	83.0	32.3	41.3	**27.0**	19.3	27.7	42.5
Luo et al. [29]	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
Hong et al.[16]	89.2	**49.0**	70.7	13.5	10.9	38.5	29.4	33.7	77.9	37.6	65.8	**75.1**	32.4	77.8	39.2	45.2	0.0	25.5	35.4	44.5
Chang et al. [2]	**91.5**	47.5	82.5	31.3	25.6	33.0	33.7	25.8	82.7	28.8	82.7	62.4	30.8	85.2	27.7	34.5	6.4	25.2	24.4	45.4
Du et al. [12]	90.3	38.9	81.7	24.8	22.9	30.5	37.0	21.2	84.8	38.8	76.9	58.8	30.7	85.7	30.6	38.1	5.9	28.3	36.9	45.4
Vu et al. [38]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
Chen et al. [6]	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	85.3	39.4	78.2	63.0	22.9	84.6	36.4	43.0	5.5	**34.7**	33.5	46.4
Zou et al. [47]	89.6	58.9	78.5	33.0	22.3	**41.4**	**48.2**	**39.2**	83.6	24.3	65.4	49.3	20.2	83.3	39.0	48.6	12.5	20.3	35.3	47.0
Lian et al. [27]	90.5	36.3	84.4	32.4	**28.7**	34.6	36.4	31.5	**86.8**	37.9	78.5	62.3	21.5	**85.6**	27.9	34.8	18.0	22.9	**49.3**	47.4
Li et al. [26]	91.0	44.7	84.2	**34.6**	27.6	30.2	36.0	36.0	85.0	**43.6**	83.0	58.6	**31.6**	83.3	35.3	**49.7**	3.3	28.8	35.6	48.5
ours (ResNet101)	90.6	44.7	**84.8**	34.3	**28.7**	31.6	35.0	37.6	84.7	43.3	**85.3**	57.0	31.5	83.8	**42.6**	48.5	1.9	30.4	39.0	**49.2**
Du et al. [12]	88.7	32.1	79.5	29.9	22.0	23.8	21.7	10.7	80.8	29.8	72.5	49.5	16.1	82.1	23.2	18.1	3.5	24.4	8.1	37.7
Li et al. [26]	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
ours (VGG16)	88.1	35.8	83.1	25.8	23.9	29.2	28.8	28.6	83.0	36.7	82.3	53.7	22.8	82.3	26.4	38.6	0.0	19.6	17.1	42.4

Table 5. Comparison to the state-of-the-art results of adapting SYNTHIA to Cityscapes.

	SYNTHIA → Cityscapes													
Method	road	sidewalk	building	light	sign	vegetation	sky	person	rider	car	bus	motorbike	bike	mIoU
Luo et al. [29]	82.5	24.0	79.4	16.5	12.7	79.2	82.8	58.3	18.0	**79.3**	25.3	17.6	25.9	46.3
Tsai et al.[37]	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
Du et al. [12]	84.6	41.7	**80.8**	11.5	14.7	**80.8**	**85.3**	57.5	21.6	82.0	36.0	19.3	34.5	50.0
Li et al. [26]	**86.0**	**46.7**	80.3	14.1	11.6	79.2	81.3	54.1	27.9	73.7	**42.2**	25.7	45.3	51.4
ours (ResNet101)	83.0	44.0	80.3	**17.1**	**15.8**	80.5	81.8	**59.9**	**33.1**	70.2	37.3	**28.5**	**45.8**	**52.1**

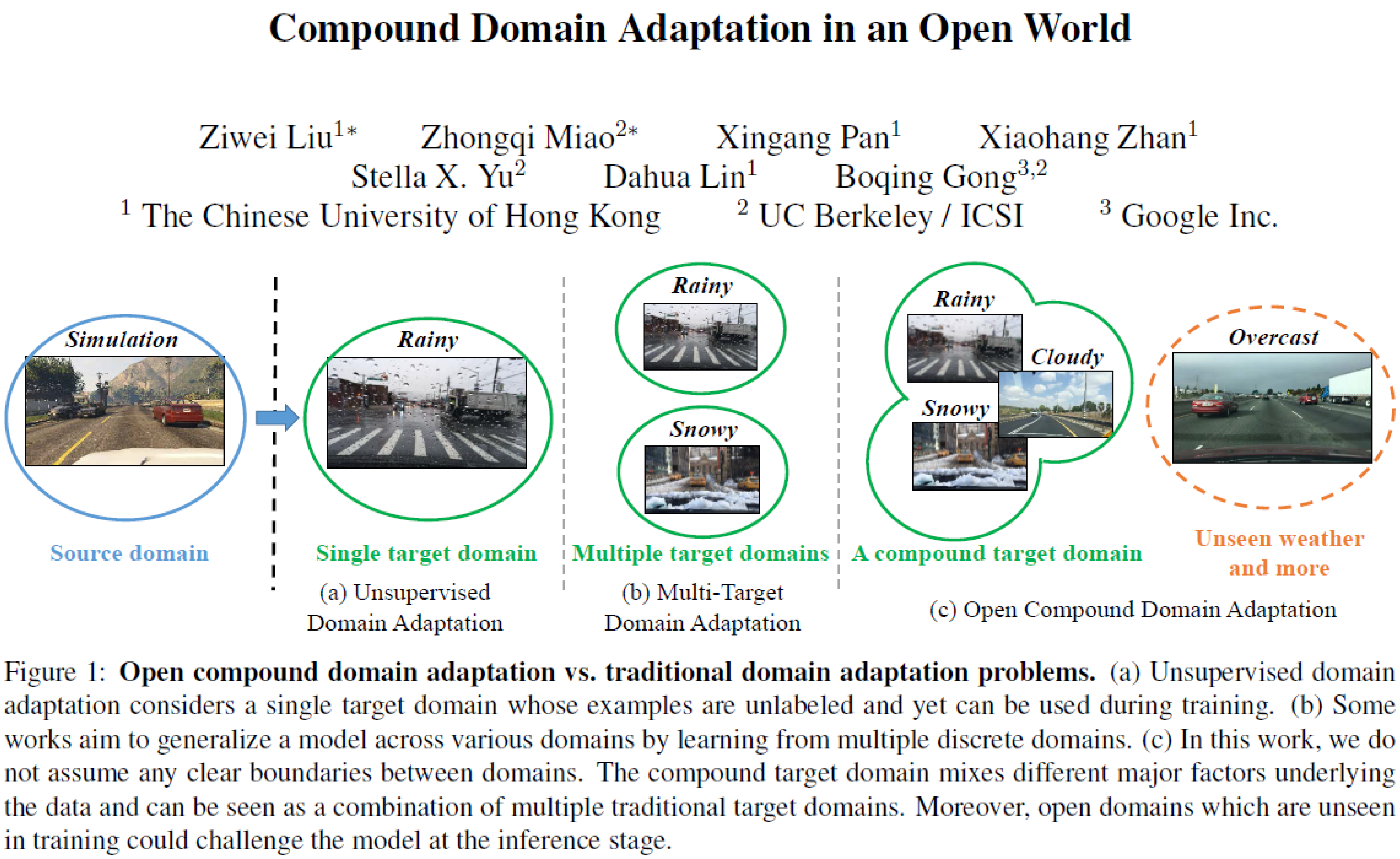
Table 2. Ablation study on the adaptation from GTA5 dataset to Cityscapes dataset. AA stands for adversarial adaptation; IT stands for image transferring; SIM stands for semantic and instance matching; SSL stands for self-supervised learning.

method	AA	IT	SIM	SSL	mIoU
source only					36.6
+ AA[37]	✓				41.4
+ IT[26]	✓	✓			44.9
+ SIM	✓	✓	✓		46.2
+ SSL	✓	✓	✓	✓	49.2
target only					65.1

Table 6. Ablation study on the adaptation from SYNTHIA dataset to Cityscapes dataset. AA stands for adversarial adaptation; IT stands for image transferring; SIM stands for semantic and instance matching; SSL stands for self-supervised learning.

method	AA	IT	SIM	SSL	mIoU
source only					38.6
+ AA[37]	✓				45.9
+ IT[26]	✓	✓			46.0
+ SIM	✓	✓	✓		47.1
+ SSL	✓	✓	✓	✓	52.1
target only					71.7

Recent works



Compound Domain Adaptation in an Open World

Ziwei Liu[1]* Zhongqi Miao[2]* Xingang Pan[1] Xiaohang Zhan[1]
Stella X. Yu[2] Dahua Lin[1] Boqing Gong[3,2]
[1] The Chinese University of Hong Kong [2] UC Berkeley / ICSI [3] Google Inc.

Simulation
Source domain
Rainy
Single target domain
Rainy
Snowy
Multiple target domains
Rainy
Cloudy
Snowy
A compound target domain
Overcast
Unseen weather and more

(a) Unsupervised Domain Adaptation

(b) Multi-Target Domain Adaptation

(c) Open Compound Domain Adaptation

Figure 1: **Open compound domain adaptation vs. traditional domain adaptation problems.** (a) Unsupervised domain adaptation considers a single target domain whose examples are unlabeled and yet can be used during training. (b) Some works aim to generalize a model across various domains by learning from multiple discrete domains. (c) In this work, we do not assume any clear boundaries between domains. The compound target domain mixes different major factors underlying the data and can be seen as a combination of multiple traditional target domains. Moreover, open domains which are unseen in training could challenge the model at the inference stage.

Three-pronged approach to tackling OCDA

1. Training algorithm （Main idea hinges on **curriculum domain adaptation**）

We schedule the (unlabeled) examples of the compound target domain according to their "**gaps**" to the labeled source domain, so that we solve the easy domain adaptation problem first, followed by increasingly harder target-domain examples.

2. Disentangling domain-focused factors

1) train a deep neural network based classifier $E_{class}(.)$ using the **labeled source domain**.
2) extract the domain-focused factors by using another encoder $E_{domain}(.)$.

Two properties:

Ⅰ: completeness

$$\boldsymbol{Decoder}\left(\mathrm{E}_{\boldsymbol{class}}(\boldsymbol{x}), \boldsymbol{E}_{\boldsymbol{domain}}(\boldsymbol{x})\right) \approx \boldsymbol{x}$$

Ⅱ : orthogonality

$$\min_{E_{domain}} \; -\sum_i z^i_{random} \log D(E_{domain}(x^i)), \qquad (1)$$

$$\min_{D} \; -\sum_i y^i \log D(E_{domain}(x^i)), \qquad (2)$$

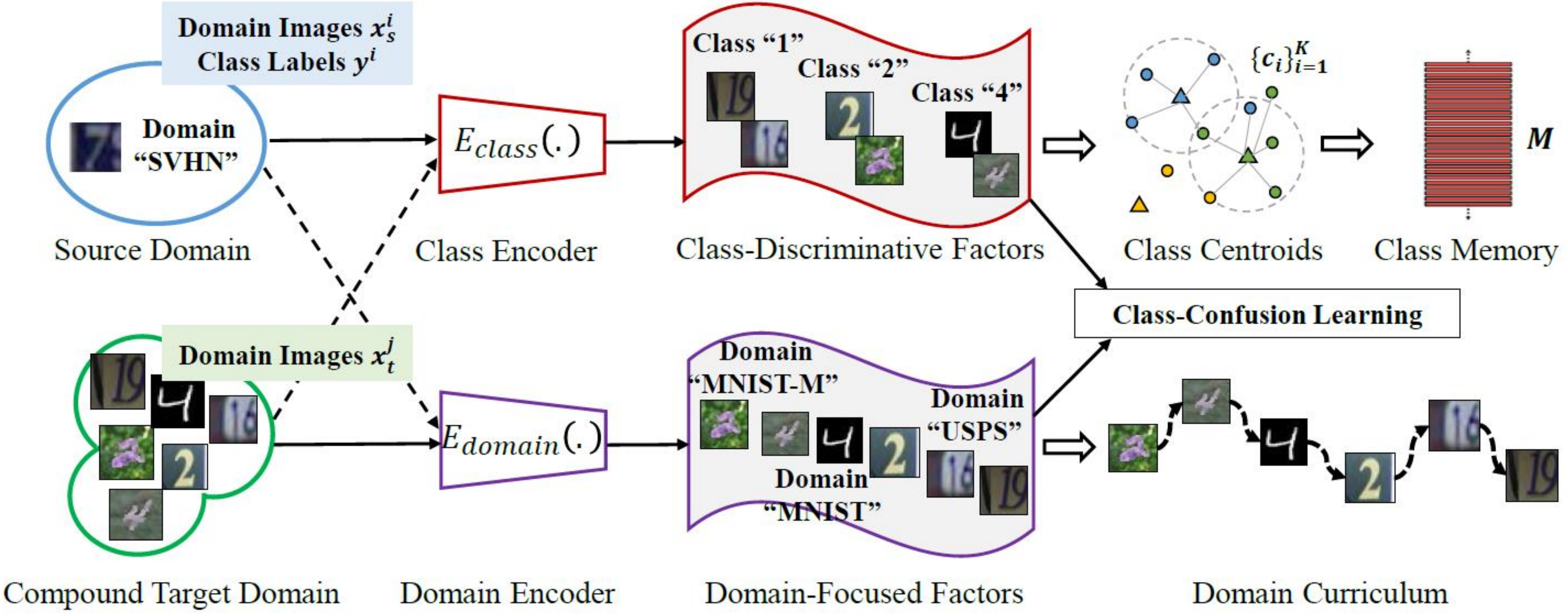
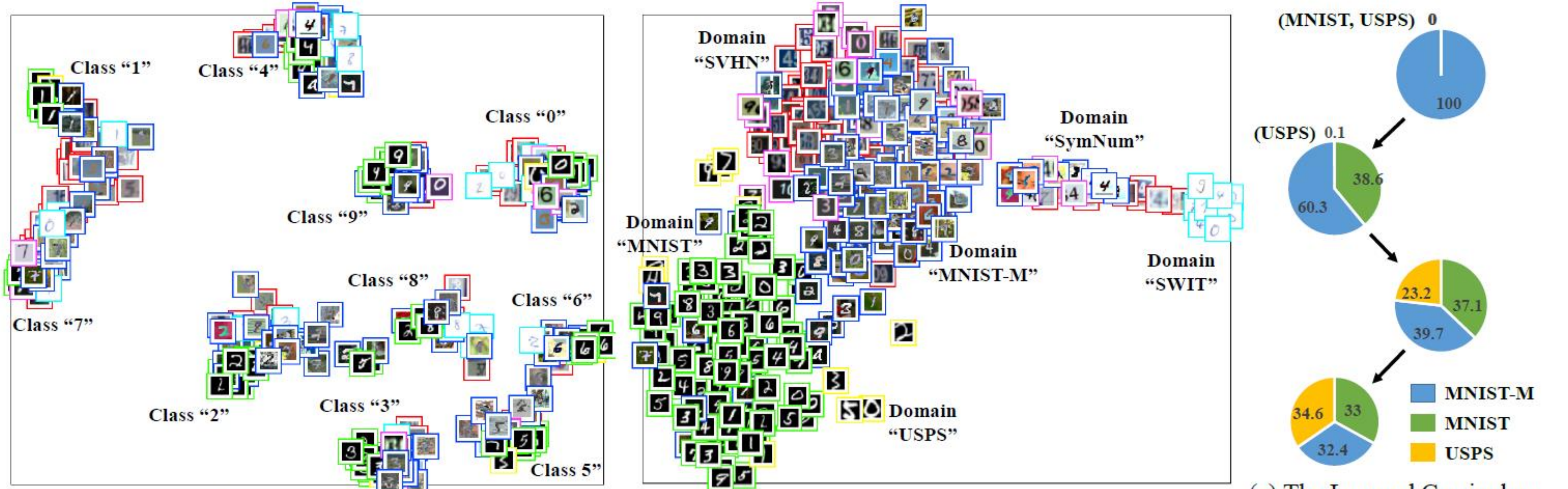


Figure 2: **Overview of disentangling domain-focused factors and curriculum domain adaptation.** We propose to separate the factors that underlie the data and contribute to the domain idiosyncrasies out of those controlling the samples' belonging to classes. It is achieved by a class-confusion algorithm in an unsupervised manner. Such domain-focused factors allow us to effectively construct the curriculum for domain-robust learning.



(a) The Learned Class-Discriminative Factors

(b) The Learned Domain-Focused Factors

(c) The Learned Curriculum along Epoches

Figure 4: **Visualization of the learned** (a) class-discriminative representations, (b) domain-focused representations through t-SNE [26], and (c) a curriculum used in our experiments.

Three-pronged approach to tackling OCDA

3. A **memory-augmented** neural network to better handle **new open domains**.

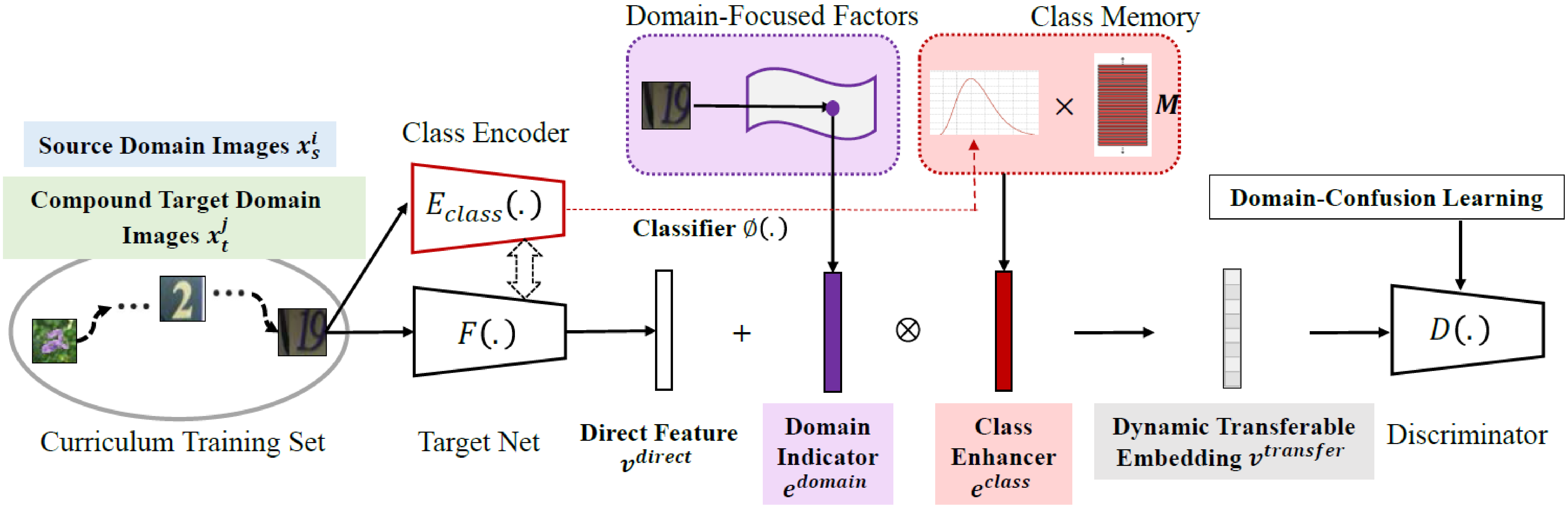
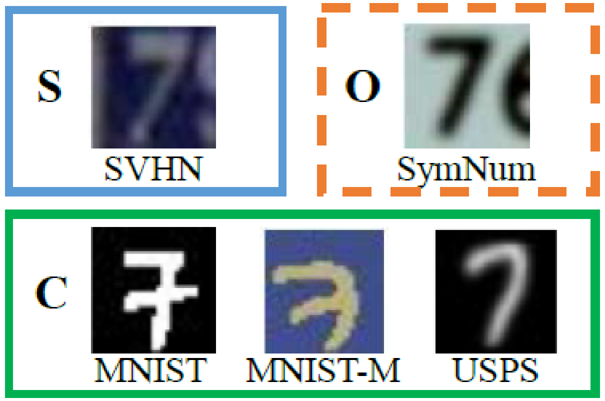

Domain-Focused Factors
Class Memory
Source Domain Images x_s^i
Compound Target Domain Images x_t^j
Class Encoder
$E_{class}(.)$
M
×
Classifier $\emptyset(.)$
Domain-Confusion Learning
$F(.)$
+
⊗
$D(.)$
Curriculum Training Set
Target Net
Direct Feature v^{direct}
Domain Indicator e^{domain}
Class Enhancer e^{class}
Dynamic Transferable Embedding $v^{transfer}$
Discriminator

Figure 3: **Overview of memory-enhanced deep neural network.** We propose to enhance the vanilla network by a memory module, which allows knowledge transfer from the source domain so that the network can dynamically balance the input-conveyed information and the memory-transferred knowledge, so as to more flexibly handle the previously unseen domains.

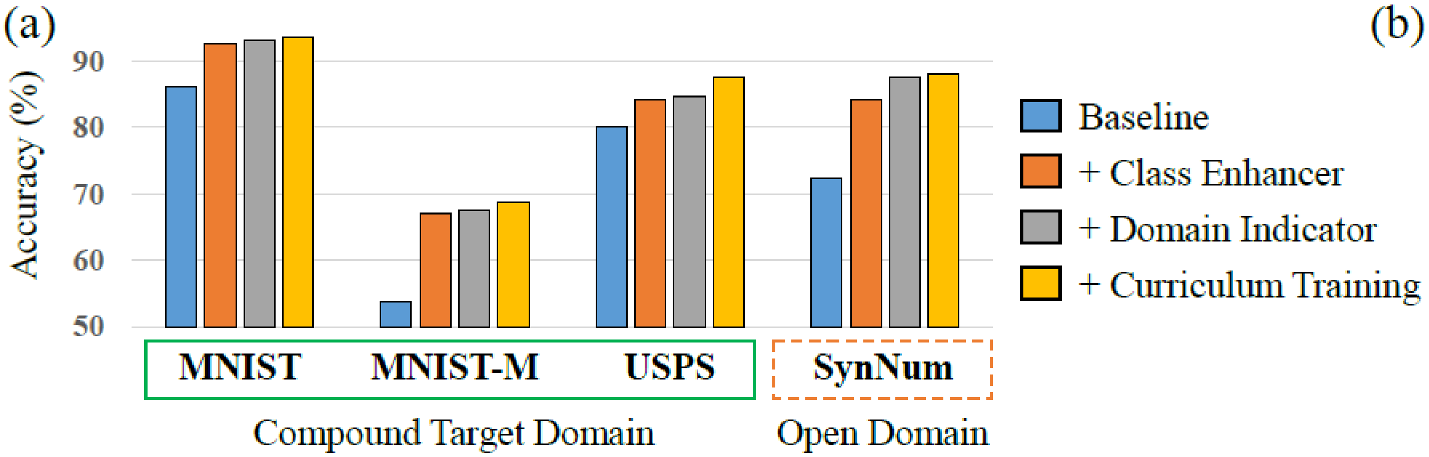
$$e^{domain} = T(E_{domain}(x))$$

$$e^{class} = (softmax(v^{direct}))^T M$$

Table 2: **Performance on the C-Digits benchmark**. The methods in gray are designed for multi-target domain adaptation. †MTDA uses domain labels, while ‡DADA uses the open domain images.



Src. Domain	Compound Domains (C)			Open (O)	Avg.	
SVHN →	**MNIST**	**MNIST-M**	**USPS**	**SynNum**	**C**	**C+O**
ADDA [42]	80.1±0.4	56.8±0.7	64.8±0.3	72.5±1.2	67.2±0.5	68.6±0.7
JAN [25]	65.1±0.1	43.0±0.1	63.5±0.2	85.6±0.0	57.2±0.1	64.3±0.1
MCD [36]	69.6±1.4	48.6±0.5	70.6±0.2	**89.8±2.9**	62.9±1.0	69.9±1.3
MTDA† [9]	84.6±0.3	65.3±0.2	70.0±0.2	-	73.3±0.2	-
DADA‡ [33]	-	-	-	-	-	80.1±0.4
Ours	**93.6±0.2**	**68.7±0.5**	**87.5±0.3**	88.0±0.8	**83.3±0.3**	**84.5±0.5**



Thank you