Unsupervised Domain Adaptation for Semantic Segmentation using Depth Distribution

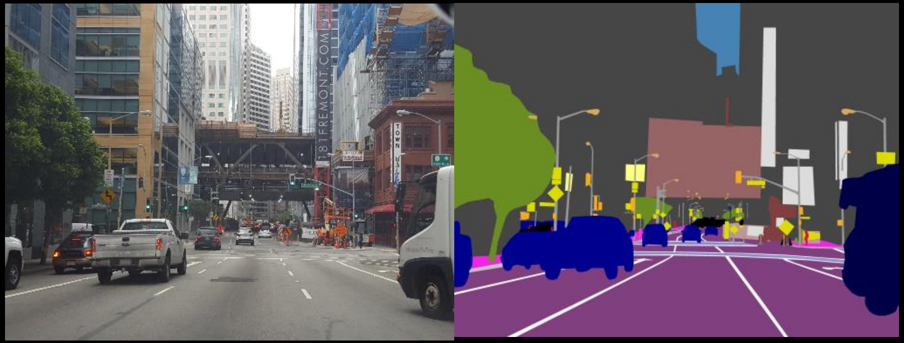
Quanliang Wu, Huajun Liu

School of Computer Science, Wuhan University

{quanliangwu, huajunliu}@whu.edu.cn

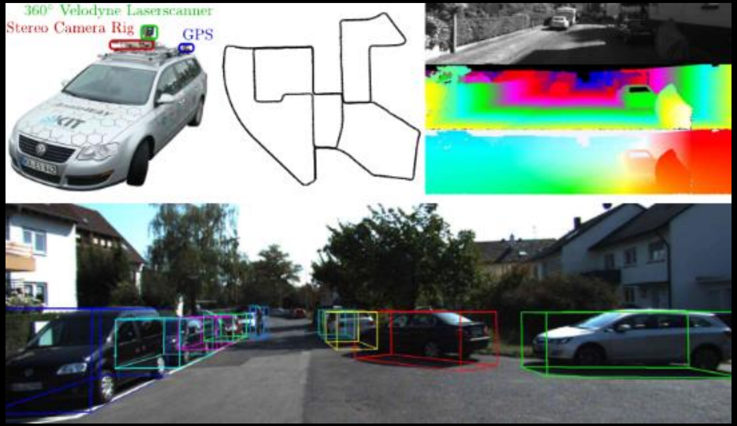
NeurIPS 2022

Semantic Segmentation in Unsupervised Domain Adaptation



Source images with annotations

Target images Predicted annotations

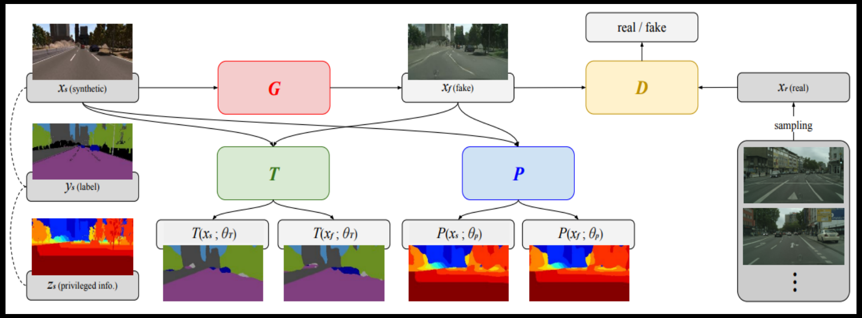


Autonomous driving

Application

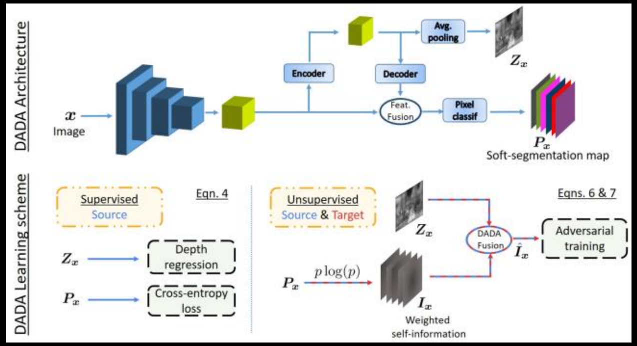
Image editing

Related Work



SPIGAN [ICLR 2019]

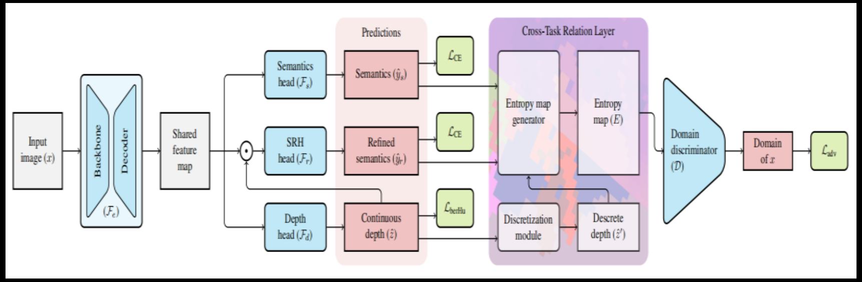
Plain way



DADA [ICCV 2019]

Plain way

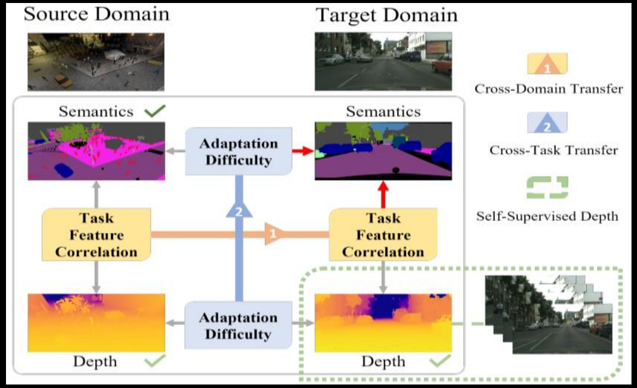
Using Depth to bridge domain gap



CTRL [CVPR 2021]

discrete depth levels

Lacking a more detailed quantitative description of depth information

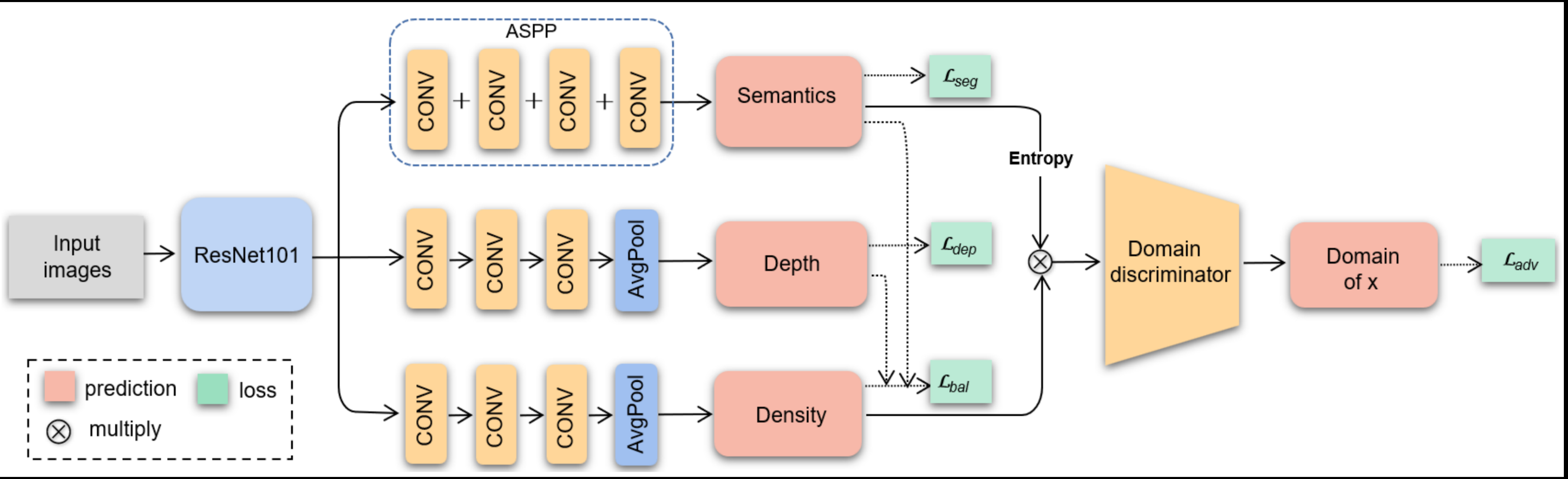


CorDA [ICCV 2021]

Obtain/generate depth in advance

*Use the **Gaussian mixture models** to build the depth distribution for different semantic classes.*

Our Framework



We use standard multi-task learning framework to obtain three sub-tasks, i.e. semantic segmentation, depth regression, and **depth distribution density estimation**.

We explore pixel aggregation priors of different classes on the source domain to help refine the pseudo-labels on the target domain for self-supervised training.

Our Loss Function

Semantics prediction

$$\mathcal{L}_{seg}(\hat{P}, P) = -\sum_{i=1}^{C} P_i \log \hat{P}_i$$

Depth regression

$$\mathcal{L}_{dep}(\hat{Z}, Z) = berHu\left(\hat{Z} - Z\right)$$

Density estimation

branch balance loss

$$\mathcal{L}_{bal}(\hat{D}, D) = berHu\left(\hat{D} - D\right)$$

Density values of each pixel can be calculated by

$$p\left(\vec{X}_i\right) = \sum_{j=1}^{K} \phi_{ij} \mathcal{N}\left(\vec{X}_i \mid \vec{\mu_{ij}}, \Sigma_{ij}\right)$$

Source domain training, ground truth depth, the predicted segmentation map and pre-constructed source domain GMMs to generate *Ds.*

Target domain training, estimated depth, the predicted segmentation map and pre-constructed source domain GMMs to generate *Dt.*

Our Loss Function

$$\min_{\theta_{net}} \mathop{\mathbb{E}}_{\mathfrak{D}^{(s)}} \left(\lambda_{seg}\mathcal{L}_{seg} + \lambda_{dep}\mathcal{L}_{dep} + \lambda_{bal}\mathcal{L}_{bal}\right),$$

$$\min_{\theta_{net}} \mathop{\mathbb{E}}_{\mathfrak{D}^{(t)}} \left(\lambda_{tar}\mathcal{L}_{bal}\right),$$

Adversarial Training

$$\min_{\theta_{\mathcal{D}}} \left\{ \mathop{\mathbb{E}}_{\mathfrak{D}_s} \left[\log \mathcal{D}\left(\hat{F}_s\right)\right] + \mathop{\mathbb{E}}_{\mathfrak{D}_t} \left[\log\left(1 - \mathcal{D}\left(\hat{F}_t\right)\right)\right] \right\}$$

$$\min_{\theta_{net}} \mathop{\mathbb{E}}_{\mathfrak{D}_t} \left[\log \mathcal{D}\left(\hat{F}_t\right)\right]$$

Hyper parameter $\lambda_{seg} = 1.0$, $\lambda_{dep} = 0.5 \times 10^{-2}$, $\lambda_{bal} = 10^{-2}$, $\lambda_{tar} = 5 \times 10^{-2}$, $\lambda_{adv} = 5 \times 10^{-2}$

Spatial Aggregation Priors for Pseudo-labels Refinement

Pixels of large objects, such as sky and road, have a large-scale aggregation in image space, while pixels of small objects, such as person and bicycle, have relatively small-scale aggregation in image space.

$$thres_i = N_{base0} + \frac{N_i - N_{min}}{N_{max} - N_{min}} \times N_{base1}$$

Algorithm 1: Spatial prior pseudo-labels refinement algorithm

Input: A target sample with predicted pseudo-labels.
Output: Refined pseudo-labels.

```
Initialize all pixels to set their flags T_wh=0.
for w=0 to W do
    for h=0 to H do
        if T_wh=0 && Confidence_wh ≥0.9 then
            Search around it for pixels that satisfy the following conditions:
                Their prediction class is the same as T_wh, and their confidence value ≥ 0.9.
            Iterate over taking these points as the fiducial points and search around them outward for the qualified points.
        Count the number of all qualified pixels, and record as N_c;
        if N_c ≥ thres_i then
            Set flags of all these pixels to 1;
Pixels labeled with 1 are reserved, and their pseudo-labels can be used for self-supervised learning.
```

Experiments and Analysis

UDA Benchmarks

SYNTHIA → Cityscapes (16 classes),
SYNTHIA → Cityscapes (7 classes),
and SYNTHIA → Mapillary (7 classes).

“mean Intersection over Union” (mIoU in %) on the 16 classes

the mIoU (%) of the 13 classes (mIoU*) excluding classes with *

Experimental Setup

a single NVIDIA 1080Ti GPU, PyTorch, ResNet-101,
Atrous Spatial Pyramid Pooling (ASPP), DC-GAN

Learning rates of the prediction and discriminator networks are set as $2.5 \times 10{-}4$ and $1.0 \times 10{-}3$ respectively.
In self-training, the parameters are: Q1 = 54*K*, Q2 = 30*K*.

Experiments and Analysis

Models	Depth	SYNTHIA → Cityscapes (16 classes)																	
		road	sidewalk	building	wall*	fence*	pole*	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU↑	mIoU*↑
SPIGAN[11]	√	71.1	29.8	71.4	3.7	0.3	**33.2**	6.4	15.6	81.2	78.9	52.7	13.1	75.9	25.5	10.0	20.5	36.8	42.4
AdaptSegnet[26]		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
AdaptPatch[37]		82.2	39.4	79.4	–	–	–	6.5	10.8	77.8	82.0	54.9	21.1	67.7	30.7	17.8	32.2	–	46.3
CLAN[38]		81.3	37.0	80.1	–	–	–	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	–	47.8
Advent[19]		87.0	44.1	79.7	9.6	0.6	24.3	4.8	7.2	80.1	83.6	56.4	23.7	72.7	32.6	12.8	33.7	40.8	47.6
DADA[12]	√	**89.2**	**44.8**	**81.4**	6.8	0.3	26.2	8.6	11.1	81.8	**84.0**	54.7	19.3	79.7	40.7	14.0	38.8	42.6	49.8
CTRL[13]	√	86.9	43.0	80.7	19.2	0.9	27.2	11.6	12.6	81.3	83.2	60.7	24.0	84.2	46.2	22.0	44.2	45.5	52.4
Ours	√	85.3	40.2	79.7	**19.6**	**1.3**	29.4	**29.7**	**32.2**	**82.5**	79.2	**64.3**	**26.7**	**85.2**	**49.4**	**22.7**	**44.9**	**48.2**	**55.5**

Table 1: The quantitative results of different methods for semantic segmentation performance (IoU and mIoU, %) on SYNTHIA→ Cityscapes(16 classes).

Experiments and Analysis

			(a) SYNTHIA → Cityscapes (7 classes)								(b) SYNTHIA → Mapillary (7 classes)							
Res.	Model	Depth	*flat*	*const*	*object*	*nature*	*sky*	*human*	*vehicle*	mIoU↑	*flat*	*const*	*object*	*nature*	*sky*	*human*	*vehicle*	mIoU↑
320*640	SPIGAN[11]	√	91.2	66.4	9.6	56.8	71.5	17.7	60.3	53.4	74.1	47.1	6.8	43.3	83.7	11.2	42.2	44.1
	Advent[19]		86.3	72.7	12.0	70.4	81.2	29.8	62.9	59.4	82.7	51.8	18.4	67.8	79.5	22.7	54.9	54.0
	DADA[12]	√	89.6	76.0	16.3	74.4	78.3	43.8	65.7	63.4	83.8	53.7	**20.5**	62.1	84.5	26.6	59.2	55.8
	CTRL[13]	√	90.8	77.5	15.7	77.1	**82.9**	45.3	68.6	65.4	**86.6**	57.4	19.7	**73.0**	**87.5**	**45.1**	**68.1**	**62.5**
	Ours	√	**92.6**	**78.2**	**23.4**	**77.2**	**82.9**	**49.6**	**69.8**	**67.7**	86.2	**58.7**	19.4	68.9	86.1	40.4	62.4	60.3
Full	Advent[19]		89.6	77.8	22.1	76.3	81.4	54.7	68.7	67.2	86.9	58.8	30.5	74.1	85.1	48.3	72.5	65.2
	DADA[12]	√	92.3	78.3	25.0	75.5	82.2	58.7	72.4	70.4	86.7	62.1	**34.9**	75.9	88.6	51.1	73.8	67.6
	CTRL[13]	√	**92.4**	80.7	27.7	78.1	**83.6**	59.0	**78.6**	71.4	**88.5**	59.2	27.8	**79.4**	85.7	**64.4**	**79.6**	69.2
	Ours	√	**92.4**	**81.8**	**34.3**	**78.9**	82.0	**64.5**	74.1	**72.6**	87.7	**68.6**	33.7	74.8	**93.0**	61.4	73.4	**70.4**

Table 2: The quantitative results of different methods for semantic segmentation performance (IoU and mIoU, %) on SYNTHIA→ Cityscapes(7 classes) and SYNTHIA → Mapillary (7 classes) in low-resolution and full-resolution.

Experiments and Analysis

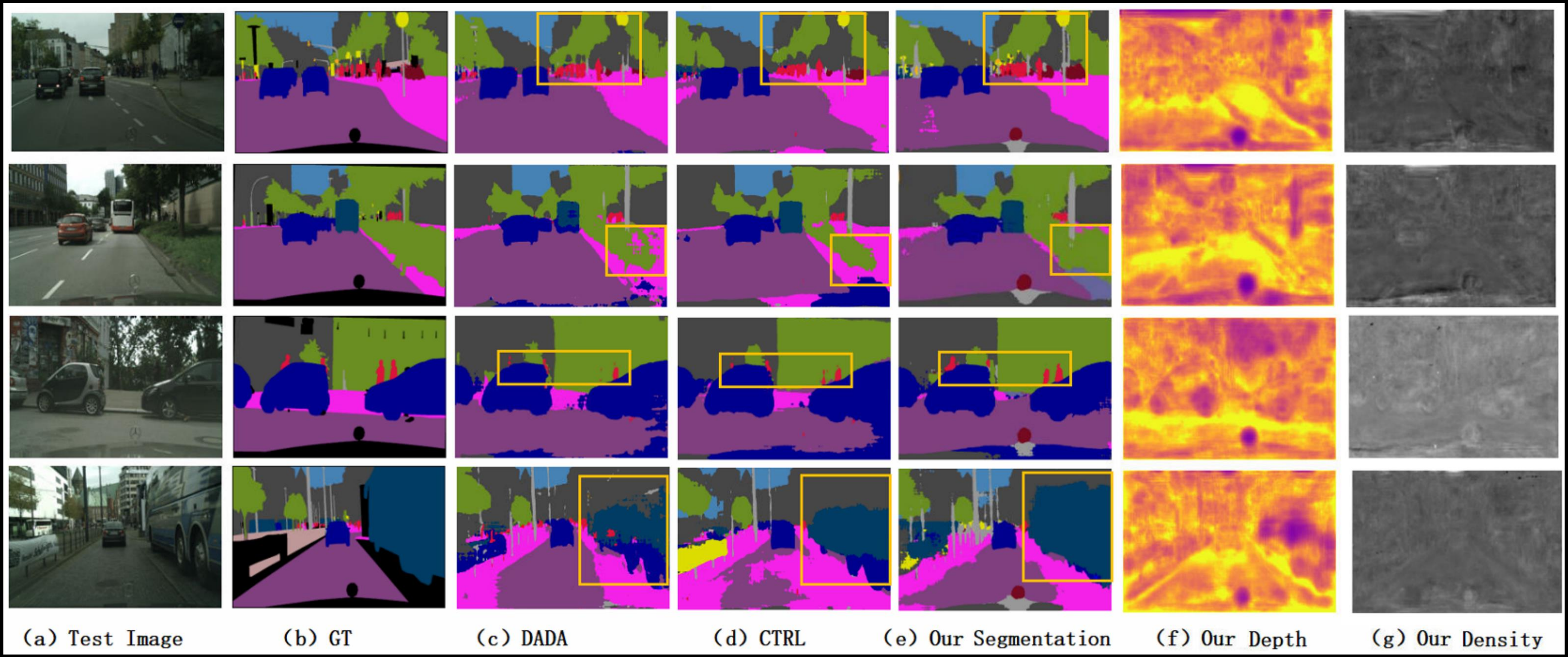


Figure 2: Qualitative results on SYNTHIA → Cityscapes (16 classes).

Experiments and Analysis

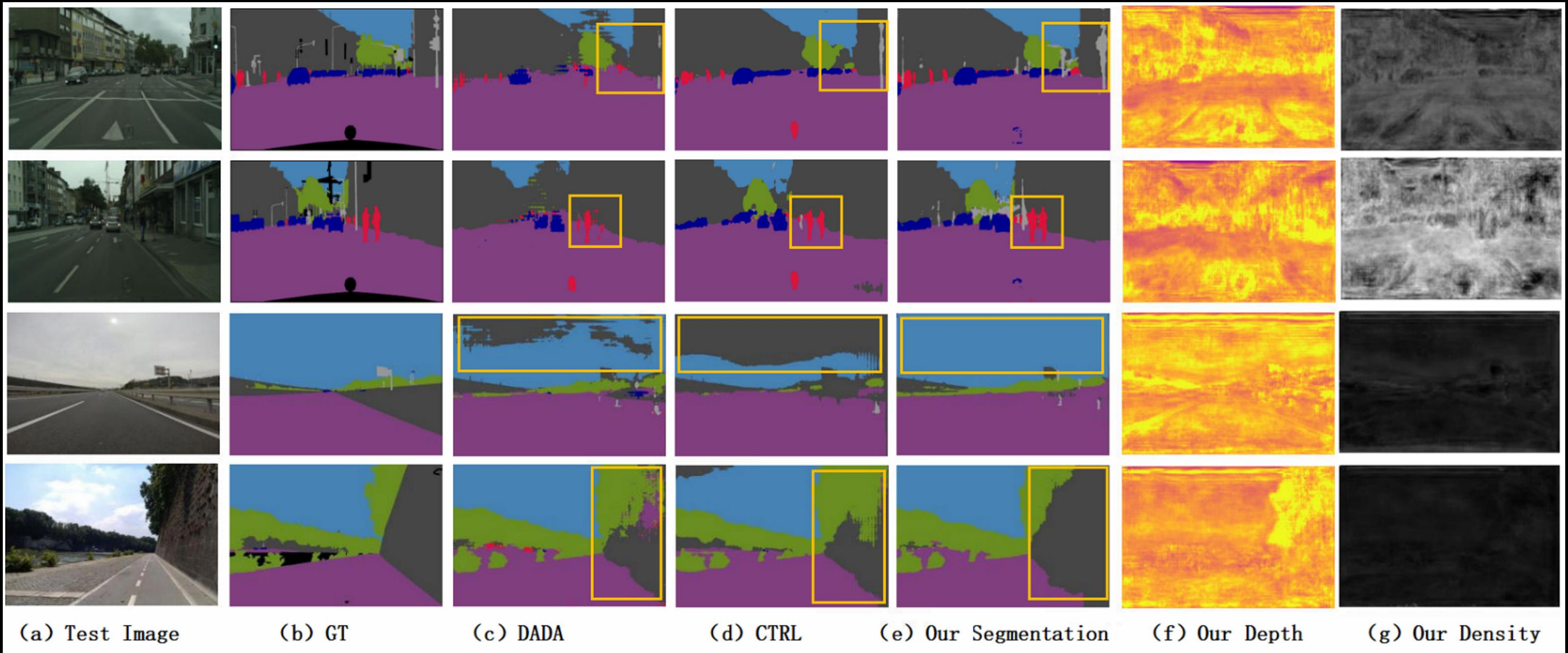


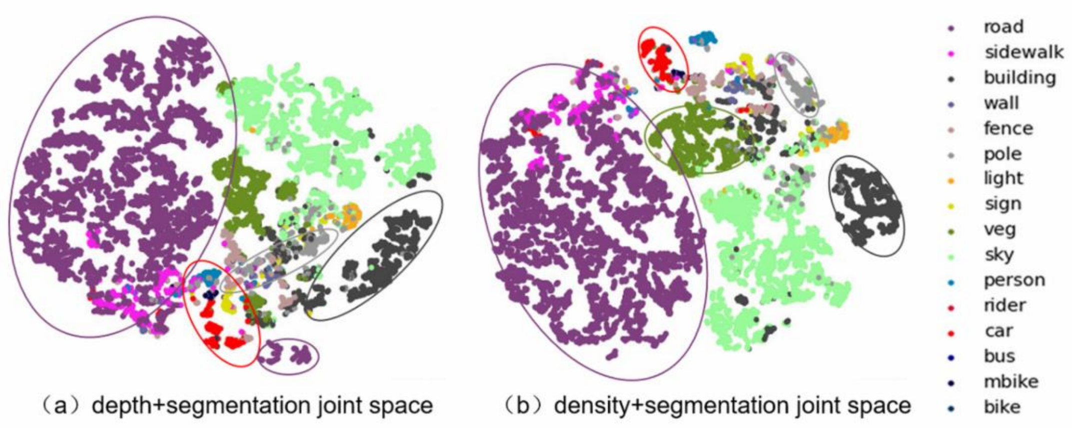
Figure 3: Qualitative results on: SYNTHIA → Cityscapes (7 classes) (upper two rows) and SYNTHIA → Mapillary (7 classes) (lower two rows).

Experiments and Analysis

Model	SegPre	DepRes	DenEst	SelfTra	SpaPri	mIoU(%)↑
M1	√	√				41.7
M2	√	√	√			44.8
M3	√	√	√	√		47.6
M4	√	√	√	√	√	**48.2**

Table 3: Ablation study of different components of our method

Situation	mIoU(%)↑
S1	44.1
S2	43.4
S3	37.8
S4	43.7
S5	**44.8**



	M1	M2
$\lvert Rel \rvert\downarrow$	0.7	**0.5**
$Rel^2\downarrow$	13.7	**9.0**
$RMS\downarrow$	20.7	**18.3**
$LRMS\downarrow$	0.9	**0.7**
$\delta_1\uparrow$	0.21	**0.26**
$\delta_2\uparrow$	0.40	**0.48**
$\delta_3\uparrow$	0.56	**0.66**

Table 4: Other analysis of different feature combinations

Experiments and Analysis

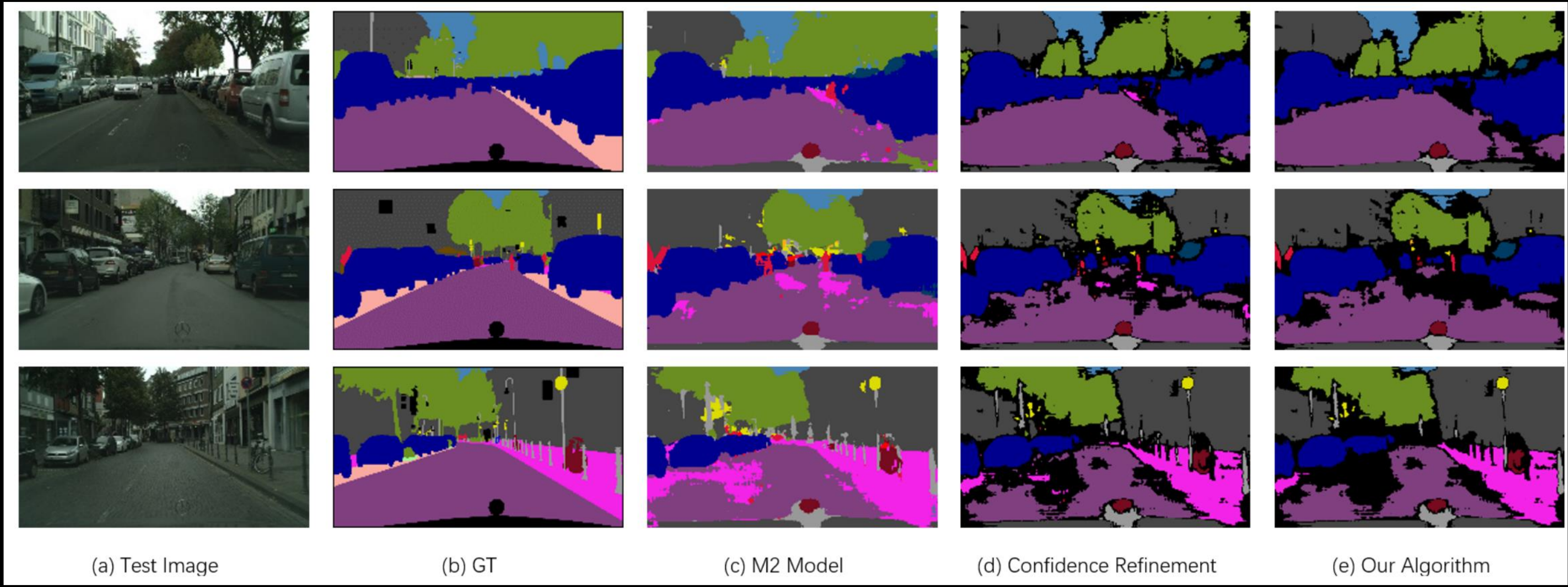


Figure 4: Comparison for qualitative results on spatial prior pseudo-labels refinement.

Thank you!