Domain Adaptation on Object Detection

Pengcheng Xu

2020, May, 8

Pipelines

Two-stage detector: Faster-RCNN

- Most of adaptation works are based on Faster-RCNN
- Two-stage adaptation: Image-level, Instance-level and regularization
- Accurate but slow

One-stage detector: SSD

- Few works are based on SSD
- Less accurate but fast

Paper list

- First one:
  - Domain Adaptive Faster R-CNN for Object Detection in the Wild
- Recent CVPRs:
  - Cross-domain Detection via Graph-induced Prototype Alignment
  - Harmonizing Transferability and Discriminability for Adapting Object Detectors

Domain Adaptive Faster R-CNN for Object Detection in the Wild

Yuhua Chen[1] Wen Li[1] Christos Sakaridis[1] Dengxin Dai[1] Luc Van Gool[1,2]

[1]Computer Vision Lab, ETH Zurich [2]VISICS, ESAT/PSI, KU Leuven

{yuhua.chen,liwen,csakarid,dai,vangool}@vision.ee.ethz.ch

Methods

- 1. Propose a probabilistic model for detection
- 2. Design adversarial domain adaptation for Image, Instance level alignment and Regularization
- 3. Use gradient reversal layer(GRL) to achieve adversarial adaptation

Probabilistic model

- DA Object detection is to learn posterior: $P(C, B|I)$ with domain shift:
  - C: classes. B : bounding box I:images

$$P_S(C, B, I) \neq P_T(C, B, I)$$

- Image level adaptation:
  - covariate shift assumption: labeling functions P(C,B|I) are same between domains. P(I) cause the distribution shift.

$$P(C, B, I) = P(C, B|I)P(I).$$

- Instance level adaptation:
  - P(C|B,I) are the same. P(B,I) cause the distribution shift.

$$P(C, B, I) = P(C|B, I)P(B, I).$$

  - P(B,I): region features based on ground truth bounding box which isn't available on target
  - P(B|I) is ideally the same but nontrivial and biased to estimate on target due to no labels.
    - Regularization for P(B|I) $P(B, I) = P(B|I)P(I)$

Probabilistic model

- Regularization for P(B|I) to force it to be domain invariant
  - Domain label: D, then Image-level domain classifier is P(D|I), Instance-level domain classifier is P(D|B,I).
  - Bayes' rule $P(D|B,I)P(B|I) = P(B|D,I)P(D|I).$
  - P(B|I) ideally domain invariant Bbox predictor
  - P(B|D,I) practically domain dependent predictor (what we have)
  - Achieve P(B|I) = P(B|D,I) by forcing P(D|B,I) = P(D|I)

Algorithms

- Image level adaptation P(I):
  - $p_i(u, v)$: output of image domain classifier for feature map activations (u,v) of the image i

$$\mathcal{L}_{img} = -\sum_{i,u,v} \left[D_i \log p_i^{(u,v)} + (1 - D_i) \log(1 - p_i^{(u,v)}) \right].$$

- Instance level adaptation P(B,I)
  - $p_{i,j}$: output of instance domain classifier for region feature j map of the image i.

$$\mathcal{L}_{ins} = -\sum_{i,j} \left[D_i \log p_{i,j} + (1 - D_i) \log(1 - p_{i,j}) \right].$$

- Regularization on P(D|B,I) = P(D|I)

$$L_{cst} = \sum_{i,j} \| \frac{1}{|I|} \sum_{u,v} p_i^{(u,v)} - p_{i,j} \|_2, \qquad (8)$$

where $|I|$ denotes the total number of activations in a feature map, and $\| \cdot \|$ is the ℓ_2 distance.

$$L = L_{det} + \lambda(L_{img} + L_{ins} + L_{cst})$$

P(c,b,i) = p(i)p(b|i)p(c|b,i)

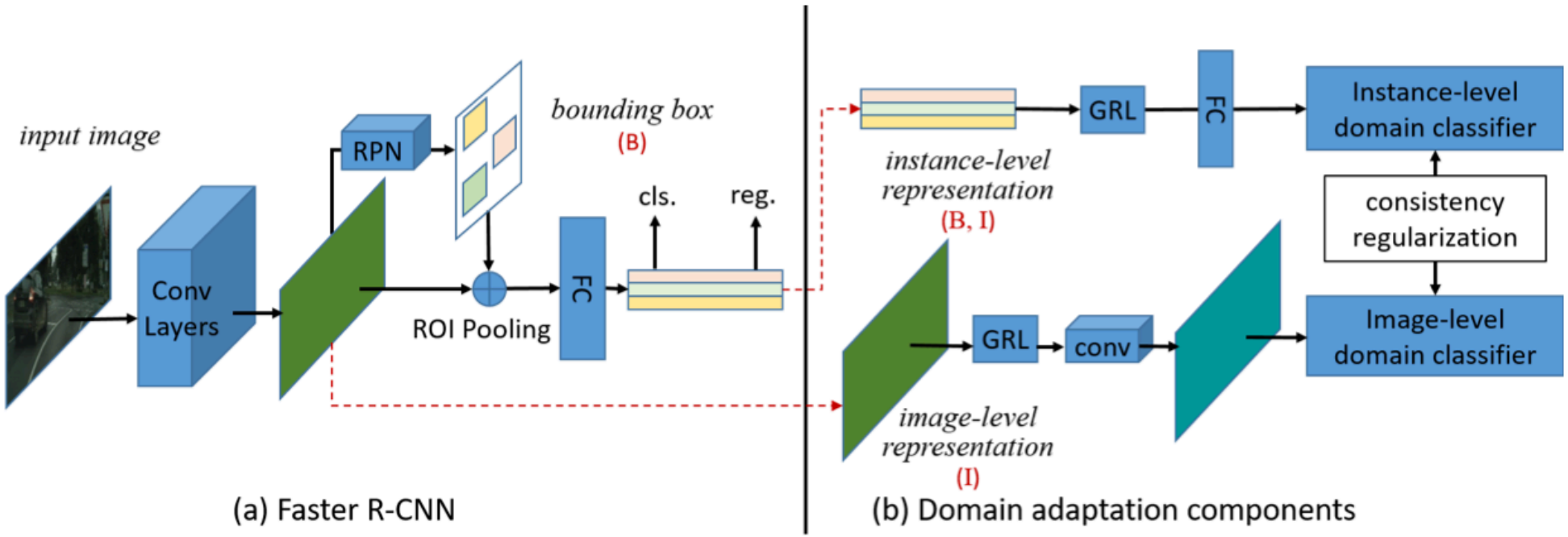


Figure 2. **An overview of our Domain Adaptive Faster R-CNN model:** we tackle the domain shift on two levels, the image level and the instance level. A domain classifier is built on each level, trained in an adversarial training manner. A consistency regularizer is incorporated within these two classifiers to learn a domain-invariant RPN for the Faster R-CNN model.

Results

	img	ins	cons	car AP
Faster R-CNN				30.12
Ours	✓			33.03
		✓		35.79
	✓	✓		37.86
	✓	✓	✓	**38.97**

Table 1. The average precision (AP) of *Car* on the *Cityscapes* validation set. The models are trained using the *SIM 10k* dataset as the source domain and the *Cityscapes* training set as the target domain. *img* is short for *image-level alignment*, *ins* for *instance-level alignment* and *cons* is short for our *consistency loss*

	img	ins	cons	K → C	C → K
Faster R-CNN				30.2	53.5
Ours	✓			36.6	60.9
		✓		34.6	57.6
	✓	✓		37.3	62.7
	✓	✓	✓	**38.5**	**64.1**

Table 3. Quantitative analysis of adaptation result between *KITTI* and *Cityscapes* . We report AP of *Car* on both directions. *e.g.* K → C and C → K.

	img	ins	cons	person	rider	car	truck	bus	train	mcycle	bicycle	mAP
Faster R-CNN				17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
Ours	✓			22.9	30.7	39.0	20.1	27.5	17.7	21.4	25.9	25.7
		✓		23.6	30.6	38.6	20.8	**40.5**	12.8	17.1	26.1	26.3
	✓	✓		24.2	**31.2**	39.1	19.1	36.2	19.2	17.1	27.0	26.6
	✓	✓	✓	**25.0**	31.0	**40.5**	**22.1**	35.3	**20.2**	**20.0**	**27.1**	**27.6**

Table 2. Quantitative results on the *Foggy Cityscapes* validation set, models are trained on the *Cityscapes* training set.

Error analysis

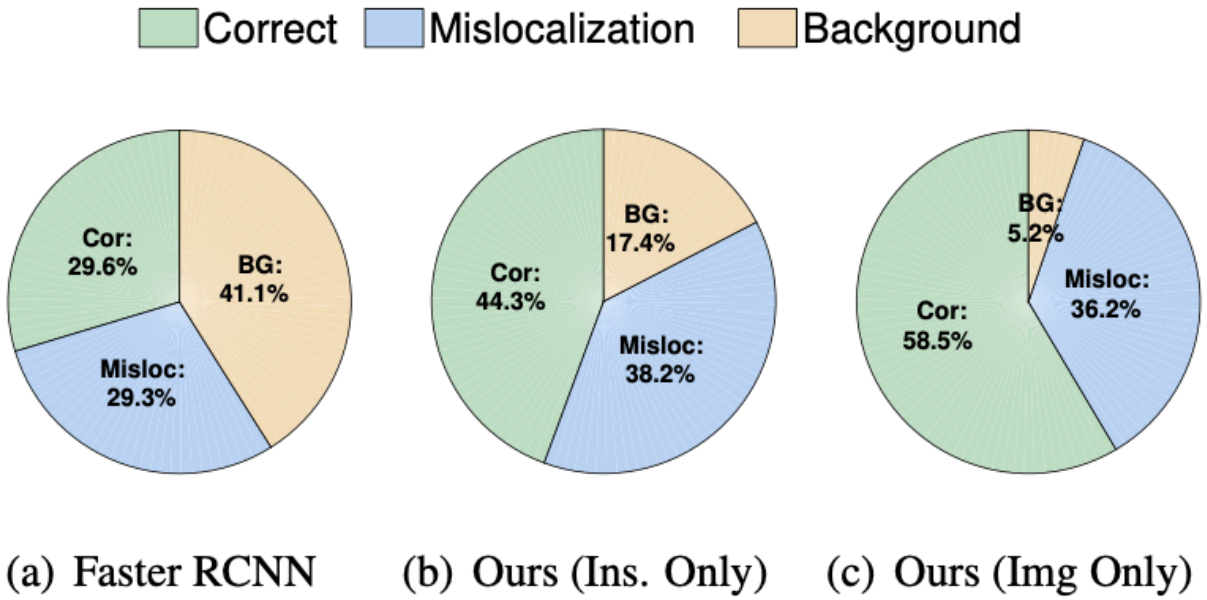


Figure 3. Error Analysis of Top Ranked Detections

Choose 2000 predictions:

Correct: >0.5 overlap with GT

Mislocalization: 0.3~0.5 overlap with GT

Background: <0.3 false positive

Harmonizing Transferability and Discriminability for Adapting Object Detectors

Chaoqi Chen[1], Zebiao Zheng[1], Xinghao Ding[1], Yue Huang[1]*, Qi Dou[2]
[1] Fujian Key Laboratory of Sensing and Computing for Smart City,
School of Informatics, Xiamen University, China
[2] Department of Computer Science and Engineering, The Chinese University of Hong Kong

Methods

- Contradiction of Transferability and Discriminability in adversarial training
  - Not all features are equally transferable
  - Adversarial adaptation potentially impair discriminability in target domain
- Methods:
  - Calibrating the transferability by hierarchically identifying and matching the transferable local region features, holistic image-level features, and ROI-based instance-level features
  - Hierarchical alignments improves the feature discriminability at multiple levels

Methods

- local region features
  - Weight up the transferable regions in one image
- Holistic image features
  - Add interpolated images generated by CycleGAN to add discriminative info on target
  - Not all interpolated images are created equally in terms of transferability
  - Weight up image features with more domain similarity
- ROI instance features
  - Lack holistic and contextual information
  - Fusion of different hierarchical features

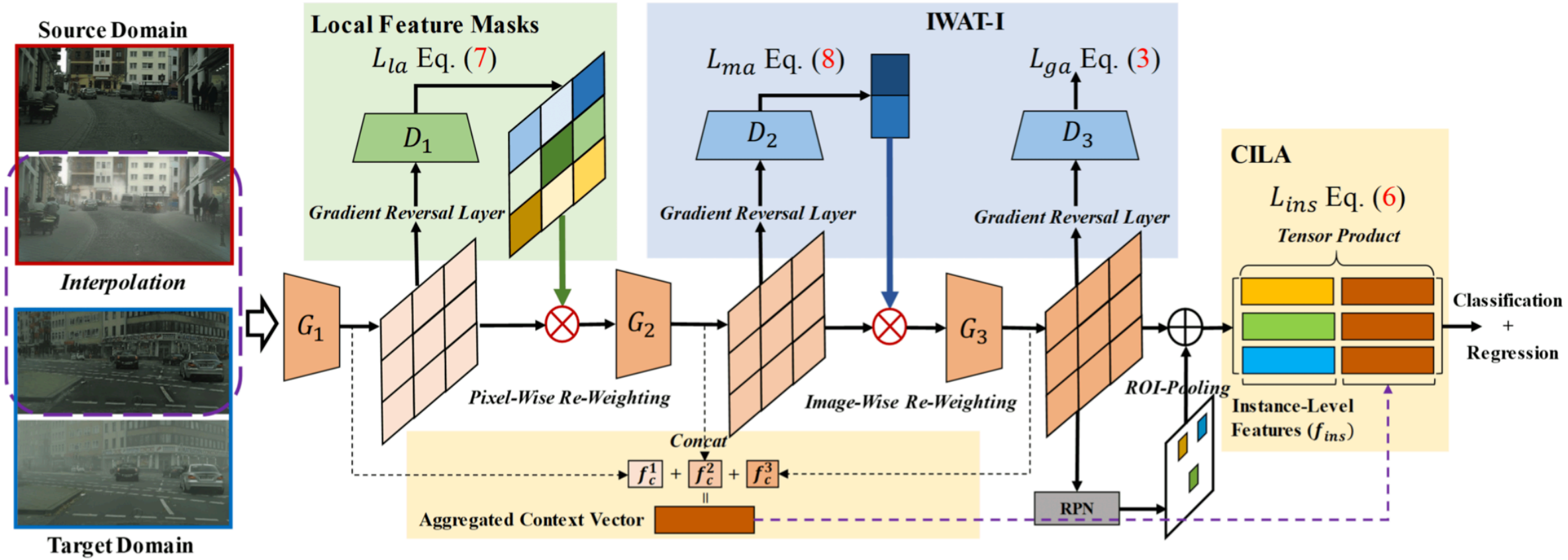


Figure 1: The overall structure of the proposed HTCN. D_1 is **pixel-wise** domain discriminator, while D_2 and D_3 are **image-wise** domain discriminator. G_1, G_2, and G_3 denote the different level feature extractors.

Importance Weighted Adversarial Training with Input Interpolation

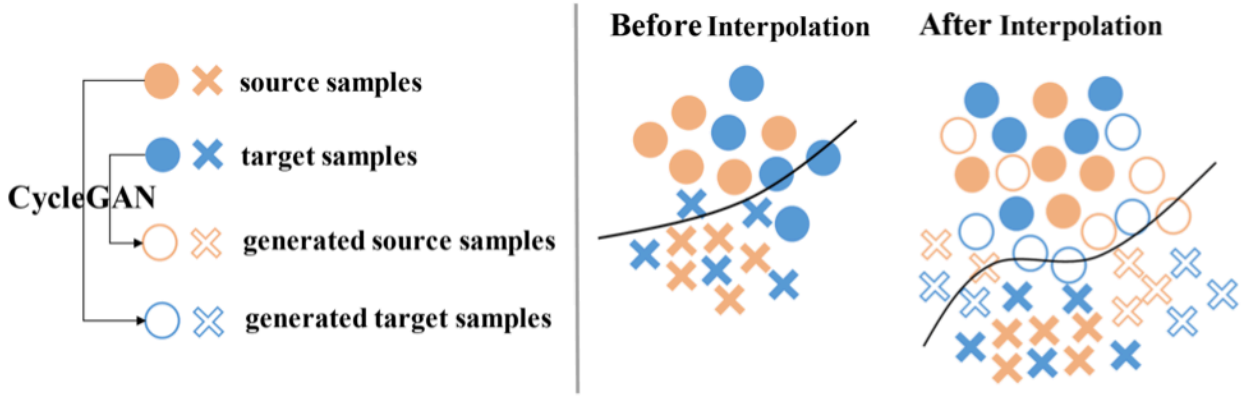


Figure 2: Motivation of the interpolation operation for improving the source-biased decision boundary through generating synthetic samples from its counterpart domain to fill in the distributional gap between domains.

1. Target domain lacks label information

2. Re-weight interpolated data with importance

Importance Weighted Adversarial Training with Input Interpolation

x_i represents image

x_i is $d_i = D_2(G_1 \circ G_2(x_i))$.

v_i represents entropy

$$v_i = H(d_i) = -d_i \cdot \log(d_i) - (1 - d_i) \cdot \log(1 - d_i) \quad (1)$$

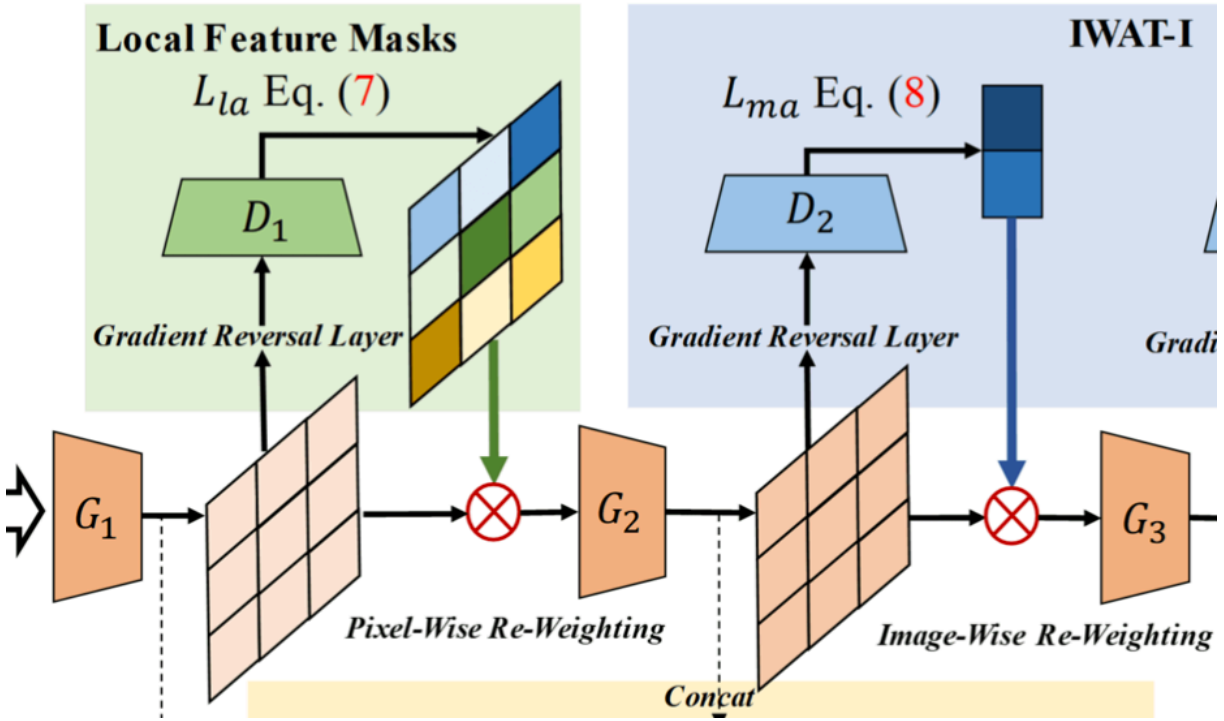
$$g_i = f_i \times (1 + v_i)$$

where f_i is the feature before feeding into D_2. The input of D_3 is $G_3(g_i)$ and its adversarial loss is defined as,

$$\mathcal{L}_{ga} = \mathbb{E}[\log(D_3(G_3(g_i^s))] + \mathbb{E}[1 - \log(D_3(G_3(g_i^t))] \quad (3)$$

Similar to attention mechanism
weighted up images features hard distinguished by D_2

not all samples are equally transferable especially after interpolation.



Context-Aware Instance-Level Alignment

- Instance-level alignment for ROI features
  - Current works alleviate local instance deviations across domains (object scale, viewpoint, deformation, and appearance)
  - Limitation: local feature info but lacks holistic context info.
  - Local features and context features are complimentary
    - Local feature distinct between domains;
    - Context features aggregated from lower layer, invariant across domains.
  - Fusion of features:
    - Concatenation of features ignores the complimentary effect; Asymmetric
    - Tensor product

Context-Aware Instance-Level Alignment

f_c^1 f_c^2 f_c^3 Represents context features of different levels

$f_{ins}^{i,j}$ Represents context features of region i from image j

$$\boldsymbol{f}_{fus} = [\boldsymbol{f}_c^1, \boldsymbol{f}_c^2, \boldsymbol{f}_c^3] \otimes \boldsymbol{f}_{ins}$$

Dimension issue: random method to get the unbiased estimate of tensor product

$$\boldsymbol{f}_{fus} = \frac{1}{\sqrt{d}}(\boldsymbol{R}_1 \boldsymbol{f}_c) \odot (\boldsymbol{R}_2 \boldsymbol{f}_{ins})$$

Formally, the CA-ILA loss is defined as follows,

$$\mathcal{L}_{ins} = -\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{i,j}\log(D_{ins}(\boldsymbol{f}_{fus}^{i,j})_s)$$

$$= -\frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{i,j}\log(1 - D_{ins}(\boldsymbol{f}_{fus}^{i,j})_t)$$

Local Feature Mask for Semantic Consistency

- Motivation: same object should be semantically invariant across domains
- Some local regions of the whole image are more descriptive and dominant than others
- Attention-like module capturing the transferable regions for semantic consistency
- Note: Previous IWAT is to weight images. Here it's to weight regions of one image. The less uncertainty regions are more informative.
- The procedure is the same as the IWAT but conducted based on pixel.

Local Feature Mask for Semantic Consistency

D_1 Pixel-level discriminator

m_f^s m_f^t Masks for source&target pixel-level features

$$\mathcal{L}_{la} = \frac{1}{N_s \cdot HW} \sum_{i=1}^{N_s} \sum_{k=1}^{HW} \log(D_1(G_1(x_i^s)_k))^2 + \frac{1}{N_t \cdot HW} \sum_{i=1}^{N_t} \sum_{k=1}^{HW} \log(1 - D_1(G_1(x_i^t)_k))^2, \quad (7)$$

$$d_i^k = D_1(r_i^k) \quad v(r_i^k) = H(d_i^k) \quad m_f^k = 2 - v(r_i^k)$$

$$\mathcal{L}_{ma} = \mathbb{E}[\log(D_2(G_2(\hat{f}_i^s))] + \mathbb{E}[1 - \log(D_2(G_2(\hat{f}_i^t))] \quad (8)$$

where $\hat{f}_i^s$ and $\hat{f}_i^t$ denote the whole pixel-wise re-weighted feature maps.

Training Loss:

$$\max_{D_1,D_2,D_3} \min_{G_1,G_2,G_3} \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} - \lambda(\mathcal{L}_{la} + \mathcal{L}_{ma} + \mathcal{L}_{ga} + \mathcal{L}_{ins}), \quad (9)$$

where λ is parameters balancing loss components.

Results

Table 1: Results on adaptation from Cityscapes to Foggy-Cityscapes. Average precision (%) is reported on the target domain. Note that the backbone of MTOR is ResNet-50, while the others are VGG-16.

Methods	Person	Rider	Car	Truck	Bus	Train	Motorbike	Bicycle	mAP
Source Only [42]	24.1	33.1	34.3	4.1	22.3	3.0	15.3	26.5	20.3
DA-Faster (CVPR'18) [7]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
SCDA (CVPR'19) [61]	**33.5**	38.0	**48.5**	26.5	39.0	23.3	28.0	33.6	33.8
MAF (ICCV'19) [18]	28.2	39.5	43.9	23.8	39.9	33.3	29.2	33.9	34.0
SWDA (CVPR'19) [44]	29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.3	34.3
DD-MRL (CVPR'19) [26]	30.8	40.5	44.3	27.2	38.4	34.5	28.4	32.2	34.6
MTOR* (CVPR'19) [4]	30.6	41.4	44.0	21.9	38.6	40.6	28.3	35.6	35.1
HTCN	33.2	**47.5**	47.9	**31.6**	**47.4**	**40.9**	**32.3**	**37.1**	**39.8**
Upper Bound	33.2	45.9	49.7	35.6	50.0	37.4	34.7	36.2	40.3

Table 2: Results on adaptation from PASCAL VOC to Clipart Dataset (%). The results of SWDA* (only G) are cited from [44], which only uses the global alignment. The backbone network is ResNet-101.

Methods	aero	bcycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	bike	prsn	plnt	sheep	sofa	train	tv	mAP
Source Only [42]	**35.6**	52.5	24.3	23.0	20.0	43.9	32.8	10.7	30.6	11.7	13.8	6.0	36.8	45.9	48.7	41.9	16.5	7.3	22.9	32.0	27.8
DA-Faster [7]	15.0	34.6	12.4	11.9	19.8	21.1	23.2	3.1	22.1	26.3	10.6	10.0	19.6	39.4	34.6	29.3	1.0	17.1	19.7	24.8	19.8
WST-BSR [25]	28.0	**64.5**	23.9	19.0	21.9	**64.3**	**43.5**	16.4	**42.2**	25.9	**30.5**	7.9	25.5	67.6	54.5	36.4	10.3	**31.2**	**57.4**	43.5	35.7
SWDA* (only G) [44]	30.5	48.5	33.6	24.8	41.2	48.9	32.4	17.2	34.5	55.0	19.0	13.6	35.1	66.2	**63.0**	45.3	12.5	22.6	45.0	38.9	36.4
SWDA [44]	26.2	48.5	32.6	**33.7**	38.5	54.3	37.1	**18.6**	34.8	**58.3**	17.0	12.5	33.8	65.5	61.6	**52.0**	9.3	24.9	54.1	49.1	38.1
HTCN	33.6	58.9	**34.0**	23.4	**45.6**	57.0	39.8	12.0	39.7	51.3	21.1	**20.1**	**39.1**	**72.8**	**63.0**	43.1	**19.3**	30.1	50.2	**51.8**	**40.3**

Results

Table 3: Results on Sim10K → Cityscapes (%). L, G, LFM, CI indicate local region alignment, global image alignment, local feature mask, and context-vector based instance-level alignment. The backbone network is VGG-16.

Methods	L	G	LFM	CI	AP on *car*
Source Only [42]	✗	✗	✗	✗	34.6
DA-Faster [7]	✓	✓	✗	✗	38.9
SWDA [44]	✓	✓	✗	✗	40.1
MAF [18]	✓	✓	✗	✗	41.1
HTCN	✓	✓	✓	✓	**42.5**

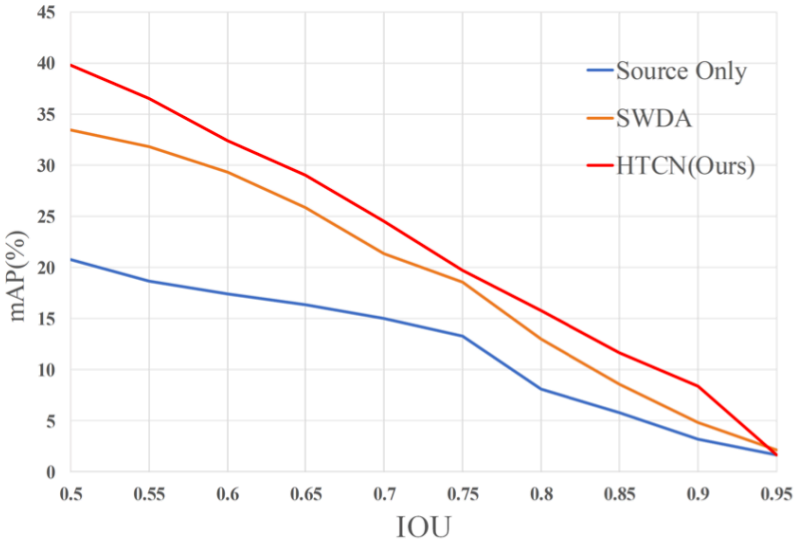


Figure 3: The performance with the variation of IOU thresholds on transfer task Cityscapes → Foggy-Cityscapes.

Ablation study

Table 4: Ablation of HTCN on Cityscapes → Foggy-Cityscapes.

Methods	Person	Rider	Car	Truck	Bus	Train	Motorbike	Bicycle	mAP
Source Only	24.1	33.1	34.3	4.1	22.3	3.0	15.3	26.5	20.3
HTCN-w/o IWAT-I	30.5	42.0	44.3	21.6	39.4	34.1	32.3	33.0	34.7
HTCN-w/o CILA	32.9	45.9	**48.5**	27.6	44.6	22.1	**34.1**	37.6	36.6
HTCN-w/o Local Feature Masks	32.9	46.2	48.2	31.1	47.3	33.3	33.0	**39.0**	38.9
HTCN-w/o Interpolation	32.8	45.6	44.8	26.5	44.3	36.9	32.0	37.1	37.5
HTCN-w/o Context Information	30.0	43.0	44.4	28.2	43.1	32.3	28.7	33.7	35.4
HTCN-w/o Tensor Product	**33.3**	46.7	47.6	28.6	46.1	36.4	32.6	37.2	38.6
HTCN (full)	33.2	**47.5**	47.9	**31.6**	**47.4**	**40.9**	32.3	37.1	**39.8**

Cross-domain Detection via Graph-induced Prototype Alignment

Minghao Xu[1,2] Hang Wang[1,2] Bingbing Ni[1,2,3*] Qi Tian[4] Wenjun Zhang[1]
[1]Shanghai Jiao Tong University, Shanghai 200240, China
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3]Huawei Hisilicon [4]Huawei Noahs Ark Lab
{xuminghao118, wang–hang, nibingbing, zhangwenjun}@sjtu.edu.cn
nibingbing@hisilicon.com tian.qi1@huawei.com

Methods

- Generated region proposals deviate from target instances
  - Relation graph to consider both location and size of proposals to integrate critical and precise features of each instance.
- Objects of a category are multi-modal
  - Confidence guided merging multi-modal instance features into prototype representation of one class.
  - Prototype-based domain alignment: narrow the intra-class distance and enlarge inter-class distance
- Class imbalance for multi-class cross domain detection
  - Weight up the sample-scarce-class

Methods

- Generate proposals
- Graph relation integrate precise instance features&confidence
- Aggregate multi-modal instance features as prototype for each class
- Weight up scarce-class prototypes for contrastive loss
- Alignment for RPN and RCNN

Cross-domain Detection via Graph-induced Prototype Alignment

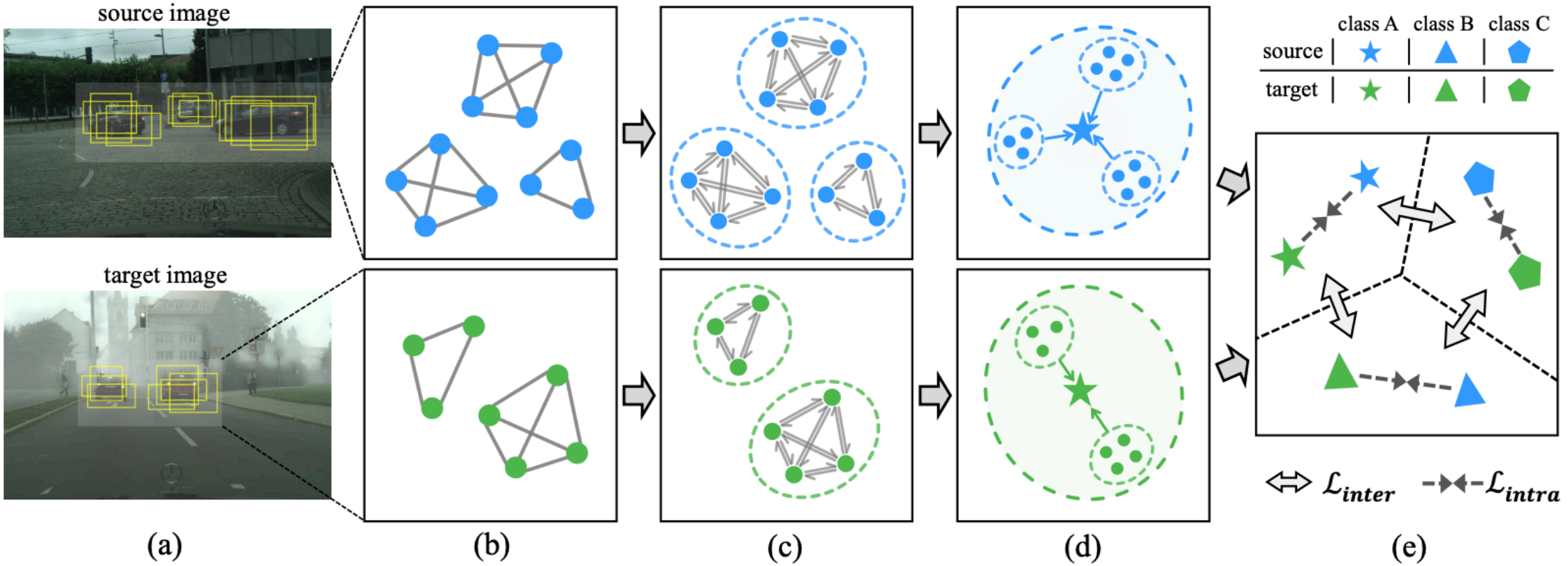


Figure 2. **Framework overview.** (a) Region proposals are generated. (b) Constructing the relation graph on produced region proposals. (c) More accurate instance-level feature representations are obtained through information propagation among proposals belonging to the same instance. (d) Prototype representation of each class is derived via confidence-guided merging. (e) Performing category-level domain alignment through enhancing intra-class compactness and inter-class separability.

Graph-based region aggregation

- Spatial matrix A for relations of proposals, σ controls the sparsity

$$\mathbf{A}_{i,j} = \exp\Big(- \frac{||o_i - o_j||_2^2}{2\sigma^2} \Big), \quad \mathbf{A}_{i,j} = \mathrm{IoU}(r_i, r_j) = \frac{r_i \bigcap r_j}{r_i \bigcup r_j},$$

- Propagate proposal features F and classification confidence P to get $\tilde{F}$ and $\tilde{P}$ with more precise instance info

$\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, $\mathbf{D} \in \mathbb{R}^{N_p \times N_p}$ $\quad \widetilde{\mathbf{F}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{F}$, $\quad \widetilde{\mathbf{F}} \in \mathbb{R}^{N_p \times d}$ and $\widetilde{\mathbf{P}} \in \mathbb{R}^{N_p \times N_c}$

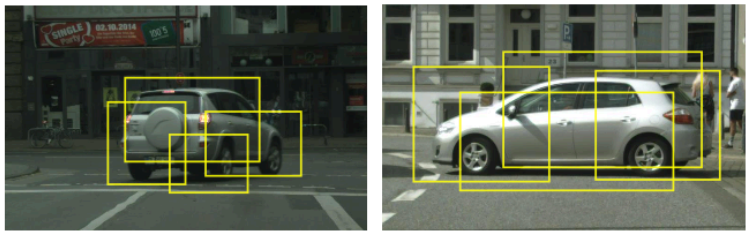
$\mathbf{F} \in \mathbb{R}^{N_p \times d}$ $\quad \mathbf{P} \in \mathbb{R}^{N_p \times N_c}$ $\quad \widetilde{\mathbf{P}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{P}$,

N_p is number of proposals; N_C is number of classes; d is feature dimension

- Confidence-guided merging instance features into prototype representations

$$c_k = \frac{\sum_{i=1}^{N_p} \widetilde{\mathbf{P}}_{ik} \cdot \widetilde{\mathbf{F}}_i^T}{\sum_{i=1}^{N_p} \widetilde{\mathbf{P}}_{ik}}, \qquad \text{where } c_k \in \mathbb{R}^d \text{ denotes the prototype of class } k.$$

To highlight the multi-modal information, we employ proposals' confidence to each class as weight during merging.



(a) (b)

Figure 1. Two vehicles and corresponding region proposals from the Cityscapes [6] dataset which serves as target domain. These two vehicles reflect multi-modal information, *e.g.* distinct scale and orientation, and the generated region proposals contain incomplete information of them.

Class-imbalance-aware Adaptation Training

- Motivation
  - The number of samples of various classes varies differently
  - The feature distribution of scarce class cannot be well aligned.
  - Put more weights on scarce-class samples
  - Calculate weights for each class based on the highest confidence of a proposal

$$p_k = \max_{1 \leqslant i \leqslant N_p} \{\widetilde{\mathbf{P}}_{ik}\}, \tag{6}$$

$$\alpha_k = \begin{cases} (1-p_k)^{\gamma} & \text{if } p_k > \frac{1}{N_c} \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

background

where p_k is the maximum confidence of class k within N_p proposals, and γ is the parameter controlling the weights among different classes. Also, we apply a hard threshold, $1/N_c$, to filter out those classes whose samples are not included in the proposal set.

Class-imbalance-aware Adaptation Training

- Construct the contrastive loss and reweight each term according to class weights

$$\mathcal{L}_{intra}(\mathcal{S},\mathcal{T}) = \frac{\sum_{i=0}^{N_c} \alpha_i^{\mathcal{S}} \alpha_i^{\mathcal{T}} \Phi(c_i^{\mathcal{S}}, c_i^{\mathcal{T}})}{\sum_{i=0}^{N_c} \alpha_i^{\mathcal{S}} \alpha_i^{\mathcal{T}}}, \tag{8}$$

$$\mathcal{L}_{inter}(\mathcal{D},\mathcal{D}') = \frac{\sum\limits_{0\leqslant i\neq j\leqslant N_c} \alpha_i^{\mathcal{D}} \alpha_j^{\mathcal{D}'} max(0, m - \Phi(c_i^{\mathcal{D}}, c_j^{\mathcal{D}'}))}{\sum\limits_{0\leqslant i\neq j\leqslant N_c} \alpha_i^{\mathcal{D}} \alpha_j^{\mathcal{D}'}}, \tag{9}$$

$$\begin{aligned}\mathcal{L}_{da} = \ & \mathcal{L}_{intra}(\mathcal{S},\mathcal{T}) + \frac{1}{3}\big(\mathcal{L}_{inter}(\mathcal{S},\mathcal{S}) \\ & + \mathcal{L}_{inter}(\mathcal{S},\mathcal{T}) + \mathcal{L}_{inter}(\mathcal{T},\mathcal{T})\big),\end{aligned} \tag{10}$$

$\Phi(x, x') = ||x - x'||_2$

$\{c_i^{\mathcal{S}}\}_{i=0}^{N_c}$, $\{c_i^{\mathcal{T}}\}_{i=0}^{N_c}$ Prototype representations

m is the margin fixed as 1.0

N_C is number of classes

Two-stage Domain Alignment

- RPN
  - RPN generate class-agnostic proposals
  - Alignment on coarse region proposals and foreground and background classes
- R-CNN
  - Fine-grained proposals via ROI
  - Alignment on accurate bbox and object classes

$$\mathcal{L}_{det} = \mathcal{L}_{cls}^{RPN} + \mathcal{L}_{loc}^{RPN} + \mathcal{L}_{cls}^{RCNN} + \mathcal{L}_{loc}^{RCNN}.$$

$$\min_{F_\theta} \mathcal{L}_{det} + \lambda_1 \mathcal{L}_{da}^{RPN} + \lambda_2 \mathcal{L}_{da}^{RCNN},$$

Results

Table 2. Experimental results (%) of *Synthetic to Real* cross-domain detection task, SIM 10k → Cityscapes.

Methods	*car* AP
Source-only	34.6
DA [5]	41.9
DivMatch [18]	43.9
SW-DA [38]	44.6
SC-DA [55]	45.1
MTOR [2]	46.6
GPA (RPN Alignment)	45.1
GPA (RCNN Alignment)	44.8
GPA (Two-stage Alignment)	**47.6**

Table 3. Experimental results (%) of *Cross Camera Adaptation* task, KITTI → Cityscapes.

Methods	*car* AP
Source-only	37.6
DA [5]	41.8
DivMatch [18]	42.7
SW-DA [38]	43.2
SC-DA [55]	43.6
GPA (RPN Alignment)	46.9
GPA (RCNN Alignment)	46.1
GPA (Two-stage Alignment)	**47.9**

Table 4. Ablation study on different manners to construct relation graph. ("ED": Euclidean distance, "LP": learnable parameter.)

ED	IoU	LP	*car* AP
			45.0
✓			46.1
✓		✓	43.2
	✓		47.6
	✓	✓	43.6

Table 1. Experimental results (%) of *Normal to Foggy* cross-domain detection task, Cityscapes → Foggy Cityscapes.

Methods	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
Source-only	26.9	38.2	35.6	18.3	32.4	9.6	25.8	28.6	26.9
DA [5]	29.2	40.4	43.4	19.7	38.3	28.5	23.7	32.7	32.0
DivMatch [18]	31.8	40.5	51.0	20.9	41.8	34.3	26.6	32.4	34.9
SW-DA [38]	31.8	44.3	48.9	21.0	43.8	28.0	28.9	35.8	35.3
SC-DA [55]	**33.8**	42.1	52.1	**26.8**	42.5	26.5	29.2	34.5	35.9
MTOR [2]	30.6	41.4	44.0	21.9	38.6	40.6	28.3	35.6	35.1
GPA (RPN Alignment)	32.5	43.1	53.3	22.7	41.4	40.8	29.4	36.4	37.4
GPA (RCNN Alignment)	33.5	44.8	52.6	26.0	41.2	37.6	29.8	35.2	37.6
GPA (Two-stage Alignment)	32.9	**46.7**	**54.1**	24.7	**45.7**	**41.1**	**32.4**	**38.7**	**39.5**

Q&A

Thank you for listening!