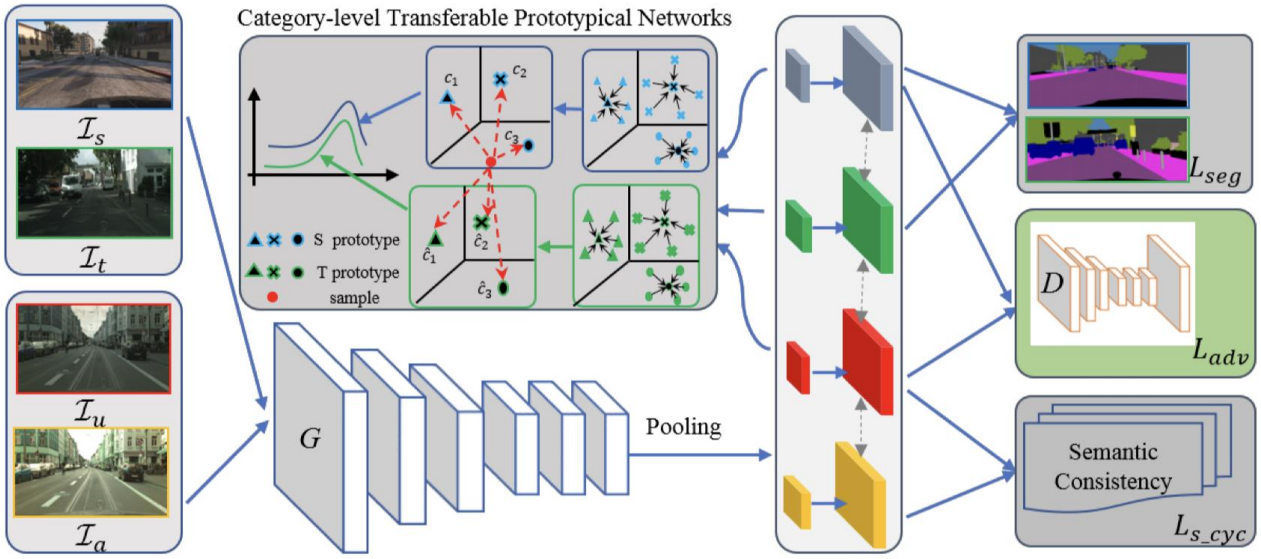
Toward Few-shot Domain Adaptation with Perturbation-invariant Representation and Transferable Prototypes

Junsong FAN, Yuxi WANG, He GUAN, Chunfeng SONG, Zhaoxiang ZHANG

Frontiers of Computer Science, DOI: 10.1007/s11704-022-2015-7

Problems & Ideas

- Problems of conventional domain adaptation approaches:
  - The target domain lacks any available data for training, resulting in much worse performance than fully supervised methods.
  - Existing approaches overlooked the balance between annotation burden and the performance.
- Ideas: A few-shot domain adaptation framework, utilizing few labels on the target domain, which balances the performance and the annotation burden.



Source images I_s, labeled target images I_t, unlabeled target images I_u and perturbed images I_a are forwarded into the segmentation networks G. The corresponding latent features are represented by cuboid with different colors. The features obtained from Is and It are trained for segmentation, and Is and I_u are used to train discriminator D. Furthermore, I_a and I_u construct the semantic consistency constraint and all the features are used to train the transferable prototypical networks.

Main Contributions

- Contributions:
  - A novel type of domain adaptation, i.e., few-shot domain adaptation (FSDA), which pursues the balance between performance and annotation burden for domain adaptation tasks.
  - A transferable prototype-based module to align the source and the target domains with only the few-shot labels.
  - A data perturbation-based approach to help alleviate the overfitting problem when only a few labeled data are available.

GTA5 → Cityscapes																					
Method	Arch.	road	side.	build.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	motor	bike	**mIoU**
Source-only	V	26.0	14.9	65.1	5.5	12.9	8.9	6.0	2.5	70.0	2.9	47.0	24.5	0.0	40.0	12.1	1.5	0.0	0.0	0.0	17.9
FCNs [2]	V	0.4	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
CyCADA [24]	V	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
MCD [3]	V	86.4	8.5	76.1	18.6	9.7	14.9	7.8	0.6	82.8	32.7	71.4	25.2	1.1	76.3	16.1	17.1	1.4	0.2	0.0	28.8
AdaptSeg [7]	V	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 [5]	V	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
Baseline	V	93.4	57.6	79.9	23.0	21.3	23.7	15.1	11.7	80.9	37.8	83.5	42.2	9.2	78.4	9.5	0.9	15.4	4.8	3.7	36.4
M_{s_cyc}	V	94.2	62.4	82.5	20.8	**30.6**	26.9	23.6	**22.9**	82.3	39.0	**87.3**	50.5	16.2	79.9	17.7	4.9	11.9	6.6	15.9	40.8
M_{proto}	V	**94.4**	**62.9**	82.2	21.4	26.3	**27.9**	**23.8**	21.5	**84.7**	38.5	85.3	**51.4**	13.9	**80.6**	14.2	4.1	3.8	3.8	**24.0**	40.3
Ours(all)	V	93.7	58.9	**82.7**	**31.4**	28.1	26.8	22.2	22.8	83.5	**40.2**	86.1	49.0	**17.1**	78.9	25.4	3.9	**20.6**	5.8	21.0	**42.1**
Source-only	R	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	36.0
AdaptSeg [7]	R	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	41.1
CLAN [5]	R	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
MRNet [42]	R	89.1	23.9	82.2	19.5	20.1	33.5	42.2	39.1	85.3	33.7	76.4	60.2	33.7	86.0	36.1	43.3	5.9	22.8	30.8	45.5
R-MRNet [43]	R	90.4	31.2	85.1	**36.9**	25.6	**37.5**	**48.8**	48.5	85.3	34.8	81.1	**64.4**	**36.8**	86.3	34.9	**52.2**	1.7	29.0	44.6	50.3
Baseline	R	93.8	59.4	79.9	21.5	19.9	26.2	22.9	18.9	83.5	40.7	84.7	58.3	25.6	86.1	**37.6**	39.8	3.7	11.3	10.2	43.4
M_{s_cyc}	R	95.2	67.6	85.0	27.0	30.5	33.0	38.2	47.8	86.6	44.3	85.9	60.3	33.8	**86.7**	20.6	14.9	24.2	15.7	56.4	50.2
M_{proto}	R	95.2	65.2	85.1	26.4	30.5	34.1	39.1	**48.7**	86.5	46.4	86.0	62.2	35.2	85.4	8.75	10.4	25.5	24.0	**58.4**	50.3
Ours(all)	R	**95.6**	**68.8**	**85.6**	27.6	**35.6**	35.4	40.2	45.2	**88.3**	**46.5**	**87.6**	61.3	36.5	86.3	30.8	10.2	**32.7**	22.4	57.2	**52.6**

Results of adaptation from GTA5 to Cityscapes. We first compare with the state-of-the-art UDA algorithms adopting the VGG16 (V) and ResNet101 (R) networks. Then, we report our results with (s_cyc)/(proto) modules respectively. We highlight the best result in each column in bold.