Dynamically Instance-Guided Adaptation:

A Backward-free Approach for Test-Time Domain Adaptive Semantic Segmentation

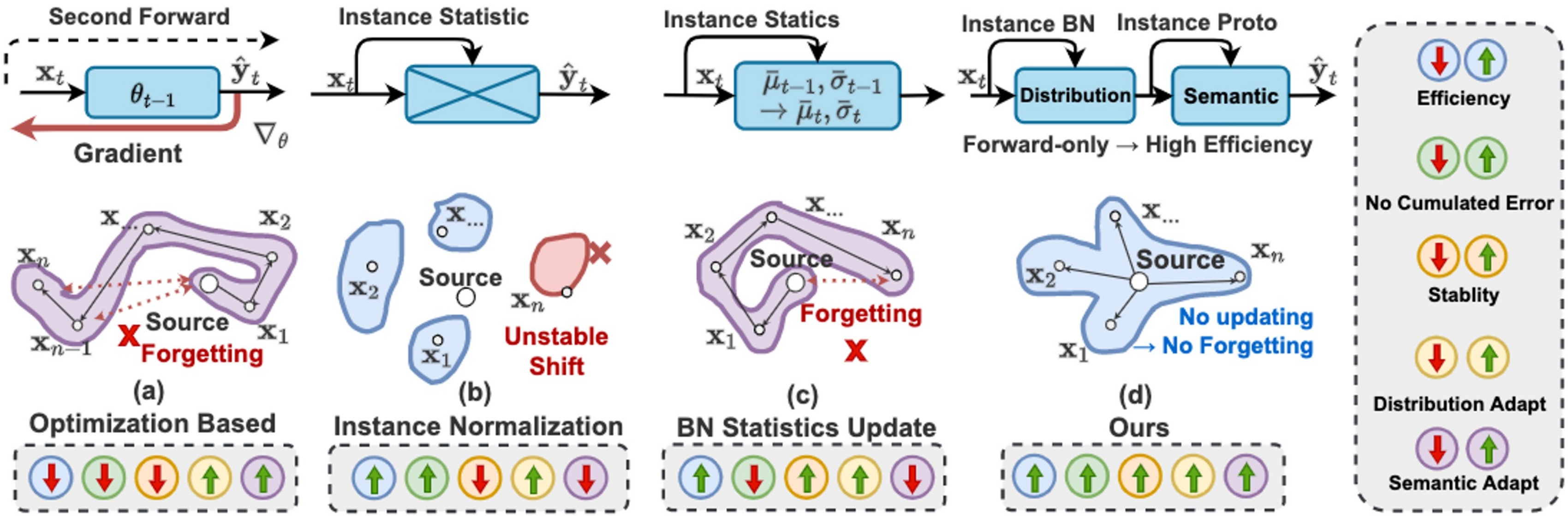
Tag: THU-PM-333
Poster number: 333

Wei Wang, Zhun Zhong, Weijie Wang, Xi Chen, Charles Ling, Boyu Wang*, Nicu Sebe

Quick preview

1. **Problem:** Adapting during test-time for segmentation
2. **Methodology:** Our efficient approach to TTDA for Semantic Segmentation eliminates the need for extra source data and improves computational efficiency.
3. **Results:** Our approach achieves superior performance with reduced computational cost and memory usage compared to existing methods.
4. **Impact:** Our research enables real-time inference in TTDA and DASS tasks, enhancing the practical application of these technologies.

Previous Approaches

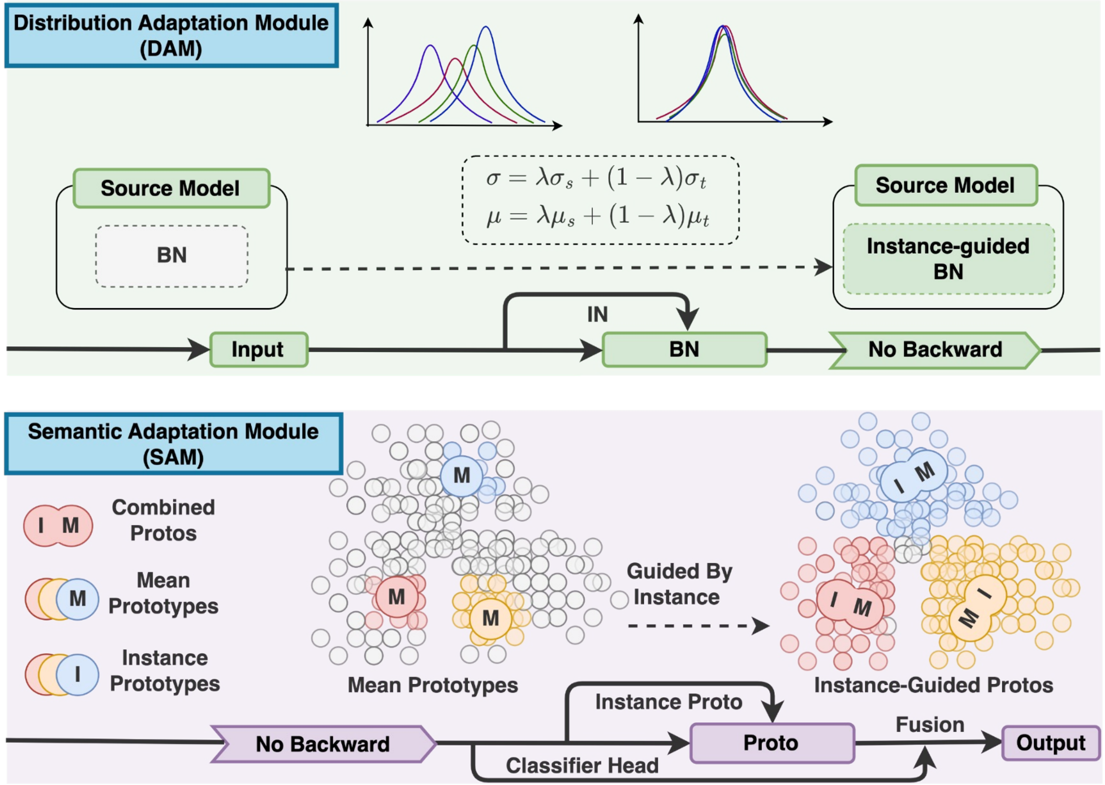


Comparison with different TTDA methods.

The proposed DIGA is a holistic method that has the properties of effectiveness (distribution & semantic adaptation and avoid unstable training & error accumulation) and efficiency (backward-free).

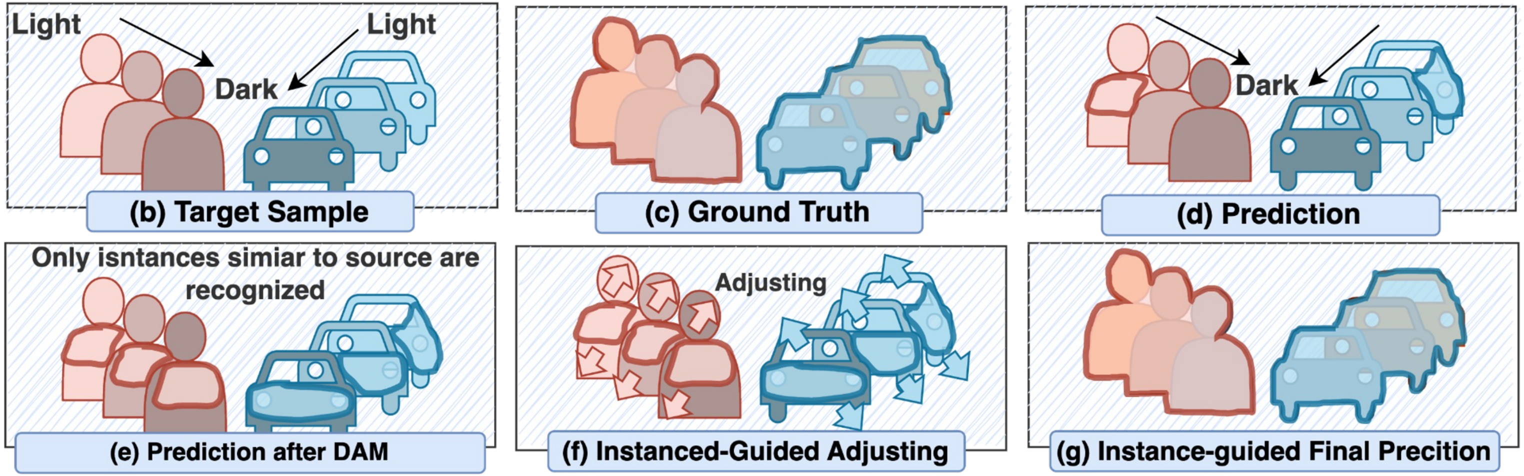
Method

Pipeline



Method

Illustration



Comparison with different TTDA methods.

The proposed DIGA is a holistic method that has the properties of effectiveness (distribution & semantic adaptation and avoid unstable training & error accumulation) and efficiency (backward-free).

Method

Algorithm

$$
\begin{aligned}
\bar{\mu}_t^T &= \lambda_{BN} \cdot \bar{\mu}^S + (1-\lambda_{BN}) \cdot \mu_t^T, \\
(\bar{\sigma}_t^T)^2 &= \lambda_{BN} \cdot (\bar{\sigma}^S)^2 + (1-\lambda_{BN}) \cdot (\sigma_t^T)^2,
\end{aligned} \quad (3)
$$

$$
\mathbf{q}_t^c = \frac{\sum^{H,W} \mathbf{z}_t^{(h,w)} \cdot \mathbb{I}(c_t^{(h,w)} = c, \max_c \hat{p}_{t,c}^{(h,w)} \geq \mathcal{P}_0)}{\sum^{H,W} \mathbb{I}(c_t^{(h,w)} = c, \max_c \hat{p}_{t,c}^{(h,w)} \geq \mathcal{P}_0)}. \quad (6)
$$

$$
\bar{\mathbf{q}}_t^c = \rho_P \cdot \bar{\mathbf{q}}_{t-1}^c + (1-\rho_P)\mathbf{q}_t^c, \ \text{with} \ \bar{\mathbf{q}}_0^c = \mathbf{q}_0^c. \quad (7)
$$

$$
\begin{aligned}
\tilde{p}^{(h,w)}(c|\mathbf{x}_t) =& \lambda_P \cdot p^{(h,w)}(c|\mathbf{x}_t, \bar{\mathbf{q}}) \\
&+ (1-\lambda_P) \cdot p^{(h,w)}(c|\mathbf{x}_t, \mathbf{q}),
\end{aligned} \quad (9)
$$

$$
p^{(h,w)} = \lambda_F \cdot \tilde{p}^{(h,w)}(c|\mathbf{x}_t) + (1-\lambda_F)\hat{p}^{(h,w)}(c|\mathbf{x}_t), \quad (10)
$$

Algorithm 1 ANAX (Testing Phase)

Input: Model f_θ, target testing sample $\mathbf{x}_t$.
Output: Prediction of $\mathbf{x}_t$.
1. Produce feature $\mathbf{z}_t$ and prediction $\hat{p}_t(\mathbf{x}_t)$ with distribution alignment of DAM (Eq. 3).
2. Calculate instance-aware prototypes $\mathbf{q}_t$ (Eq. 6).
3. Calculate historical prototypes $\bar{\mathbf{q}}_t$ (Eq. 7).
4. Calculate non-parametric predictions $\tilde{p}_t(\mathbf{x}_t)$ with SAM (Eq. 9).
5. Obtain final prediction $p(\mathbf{x}_t)$ by weighed fusion of $\hat{p}_t(\mathbf{x}_t)$ and $\tilde{p}_t(\mathbf{x}_t)$ (Eq. 10).
Return: $p(\mathbf{x}_t)$

Experiments

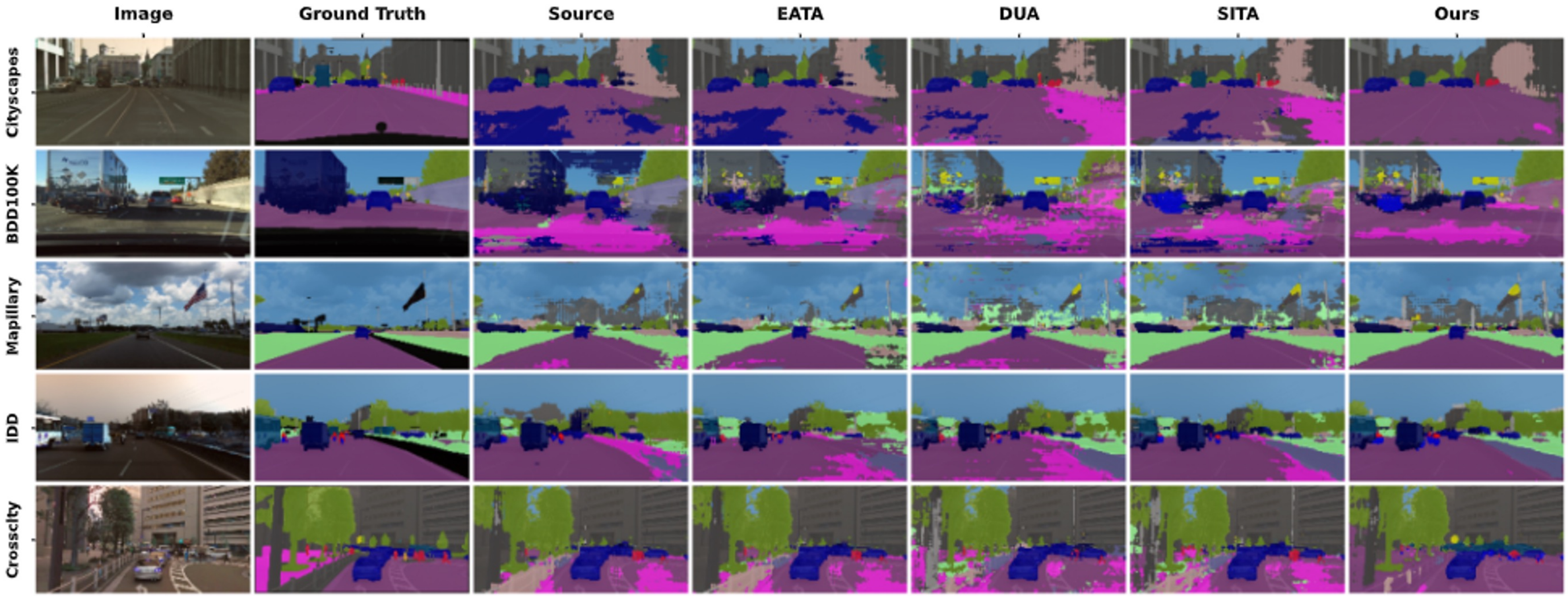
Comparison with baselines

Table 1. Comparison with state-of-the-art methods in terms of mIoU. The best score for each column is **highlighted**. CS: CityScapes, BDD: BDD100K, MA: Mapillary, IDD: IDD, CC: Cross-City. *: Use an extra augmented sample during adaptation. Avg.: Mean of mIoUs over five target domains.

Method	GTA5→						Synthia→						GTA5+Synthia →					
	CS	BDD	MA	IDD	CC	Avg.	CS	BDD	MA	IDD	CC	Avg.	CS	BDD	MA	IDD	CC	Avg.
Source [42]	35.87	29.89	38.67	38.05	30.03	34.50	30.87	21.01	31.12	26.23	31.96	28.24	37.00	28.85	41.56	39.88	32.93	36.04
Backward-based Methods																		
TENT [41]	37.30	31.53	38.29	38.96	30.59	35.33	34.89	16.99	33.46	26.23	31.68	28.65	39.39	25.19	37.32	39.51	32.84	34.85
EATA [27]	37.08	30.67	39.35	38.75	30.24	35.22	31.31	20.52	31.59	26.46	31.91	28.36	38.45	29.34	41.63	40.33	32.91	36.53
Backward-free Methods																		
IN [28]	34.25	29.64	35.01	29.8	23.87	30.51	29.53	19.33	21.92	22.08	28.24	24.22	37.09	28.81	36.02	30.99	28.63	32.31
Momentum [33]	38.12	32.42	40.79	38.74	30.2	36.05	32.84	22.51	31.12	27.24	32.23	29.45	39.61	31.72	41.79	39.88	33.17	36.66
DUA [25]	37.79	31.76	40.26	34.75	26.32	34.18	32.17	21.56	27.42	24.06	29.87	27.02	39.17	30.59	39.95	35.30	30.65	35.13
SITA* [15]	40.64	32.94	37.80	35.66	28.19	35.26	34.63	22.51	26.60	24.64	28.18	27.79	42.62	32.24	41.20	38.82	33.22	37.62
DIGA (Ours)	**45.81**	**35.78**	**44.25**	**42.73**	**33.72**	**40.46**	**41.85**	**29.09**	**36.54**	**38.36**	**36.78**	**36.52**	**46.43**	**33.87**	**43.51**	**42.08**	**34.41**	**40.06**

Experiments

Qualitative comparison of segmentation results



Experiments

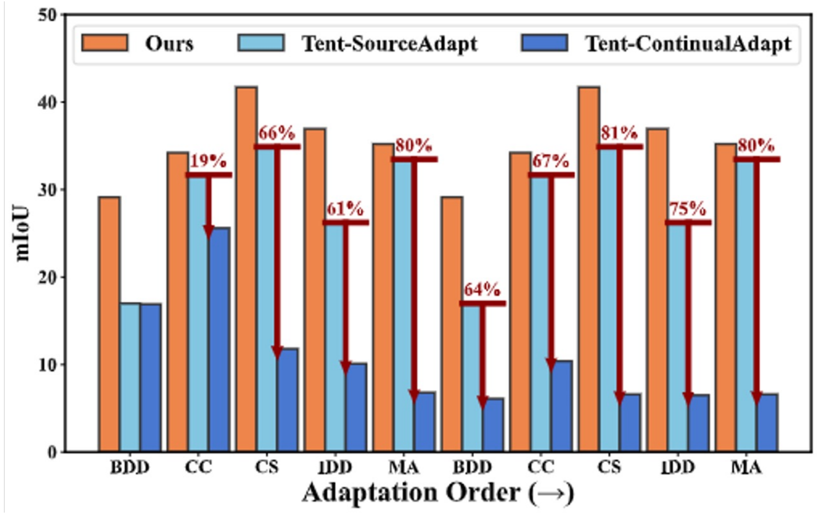
Ablation study

Modules		CS	BDD	MA	IDD	CC	Avg.
BN	Historical	35.87	29.89	38.67	**38.05**	30.03	34.50
	Instance	34.25	29.64	35.01	29.80	23.87	30.51
	DAM	**39.26**	**33.39**	**40.11**	37.23	**30.12**	**36.02**
Semantic	Historical	38.63	29.25	37.64	**41.67**	32.88	36.01
	Instance	39.16	32.26	38.68	35.09	27.98	34.63
	SAM	**42.99**	**32.69**	**41.37**	41.30	**33.49**	**38.37**
Association		**45.81**	**35.78**	**44.25**	**42.73**	**33.72**	**40.46**

- Effectiveness of DAM
- Effectiveness of SAM
- Effectiveness of Classifier Association

Experiments

Continuous adaptation



- Two version of Tent: cumulated error
- Ours is always better than Tent

Experiments

Time and space efficiency

Methods	T_{Avg}	T_{Max}	GPU Mem.	mIoU
Source-Only	134ms	141ms	3.5GB	35.87
TENT [41]	411ms	425ms	14.5GB	37.30
EATA [27]	235ms	490ms	15.6GB	37.08
Momentum [33]	**144ms**	**151ms**	**3.5GB**	37.33
DUA [25]	145ms	152ms	3.6GB	37.79
SITA [15]	253ms	256ms	5.6GB	40.64
Ours	153ms	160ms	4.0GB	**45.51**

- Space
- Time

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Thanks!

Code: https://github.com/Waybaba/DIGA

Wechat: waybaba
Twitter: waybaba1
Email: waybaba2ww@gmail.com
Homepage: https://sites.google.com/view/waybaba