KI DELTA LEARNING

Scalable AI for Automated Driving

Final Event | March 09, 2023

Domain Generalization and (Continuous) Unsupervised Domain Adaptation

Jan-Aike Termöhlen

Motivation

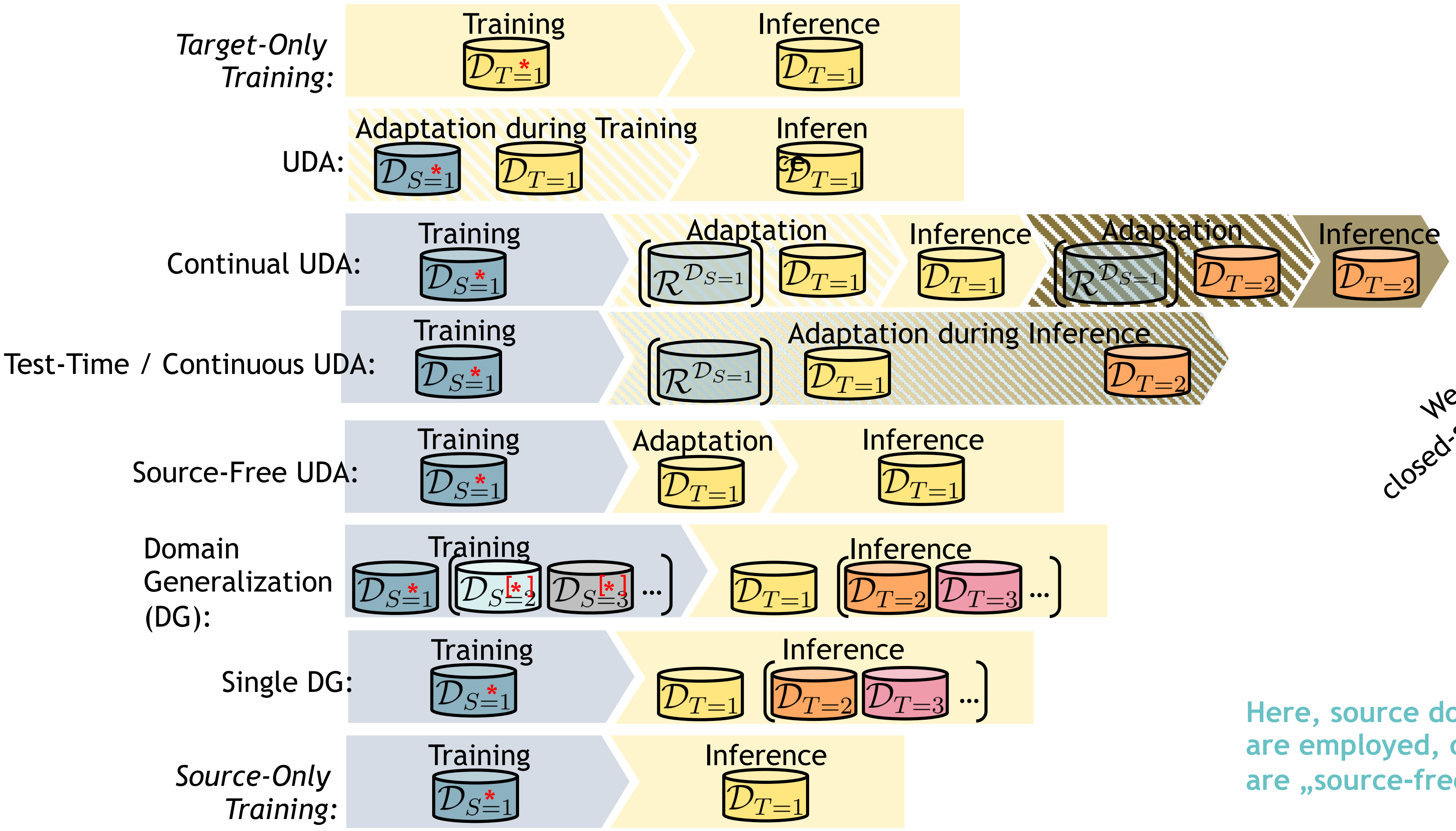
When models are trained on synthetic datasets (source domain $\mathcal{D}_S$), the domain gap to real data (target domain $\mathcal{D}_T$) typically leads to decreased performance during inference.

Training **Inference**

There are many methods to deal with this domain gap:

- Some aim at training a network to be robust without any adaptation processes
- Some adapt the mode with unlabeled samples from the target domain during training
- Some adapt the target data or the network parameters during inference

Different Task Definitions



We only consider closed-set domain adaptation!

Here, source domain representations are employed, otherwise the methods are „source-free“

Domain Generalization (DG)

ResNet101-based models | Source: $\mathcal{D}^{\text{full}}_{\text{GTA5}}$ green: better than baseline | red: worse than baseline

Method	Auxilliary Domains	mIoU (%) on $\mathcal{D}^{\text{test}*}_{\text{CS}}$	$\mathcal{D}^{\text{test}*}_{\text{MV}}$	$\mathcal{D}^{\text{test}*}_{\text{BDD}}$	$\mathcal{D}^{\text{test}*}_{\text{ACDC}}$	Mean (CS, MV, BDD)
DRPC° [1]	ImageNet	42.5	38.0	38.7	-	39.8
FSDR° [2]	ImageNet	**44.8**	43.4	41.2	-	43.1
FSDR° [2]	ImageNet	**44.8**	36.7	34.1	20.8	38.5
WildNet° [3]	ImageNet	44.6	**47.1**	**41.7**	-	**44.5**
Naive Aggregation	Synthia	41.5	46.7	38.7	**34.2**	42.3
IBN-Net [4]	-	37.1	39.6	36.0	28.2	37.6
Color Aug. (CA)	-	44.0	**47.2**	38.6	31.7	**43.3**
SAN+SAW° [5]	-	**45.3**	40.8	**41.2**	-	42.3
Baseline	-	41.0	46.0	39.2	**32.1**	41.5

DG checkpoint selection often performed on *each* of the target datasets!

only the recent WildNet improves on all datasets…

our (strong) baseline

Unsupervised Domain Adaptation (UDA)

mIoU (%) on $\mathcal{D}_{\text{CS}}^{\text{test}*}$ — ResNet101-based model

$\mathcal{D}_{\text{GTA5}}^{\text{full}}$

Method (→CS)	w/o adaptation	with adapation
AdaptSegNet° [6]	36.6	42.4
DACS° [7]	32.9	52.1
IAST° [8]	35.6	52.2
ADVENT° [9]	-	45.5
SAC° [10]	**40.8**	**53.8**

All domain adaptation approaches improve the mIoU on the target domain

DACS, IAST, and SAC perform well, all following an elaborate multi-step training process

Performance w/o adaptation varies strongl among the publications (~33% … ~41%)
(This is also the base for DG methods)

Continuous UDA

ResNet101-based models | Source: $\mathcal{D}^{\text{full}}_{\text{GTA5}}$

Task	Source-Free	Method	mIoU (%) on $\mathcal{D}^{\text{test}*}_{\text{CS}}$	$\mathcal{D}^{\text{test}*}_{\text{MV}}$	$\mathcal{D}^{\text{test}*}_{\text{BDD}}$	$\mathcal{D}^{\text{test}*}_{\text{ACDC}}$	Mean (CS, MV, BDD)	
-	yes	Baseline	41.0	**46.0**	39.2	32.1	39.6	BN statistics **frozen**
Source-Free UDA	yes	UBNA [11] (→CS)	29.2	33.3	29.1	19.8	27.9	
Continuous UDA	yes	CBNA [12]	20.4	16.4	14.2	**11.1**	15.5	
Continuous UDA	no	Online Freq. Domain Style Transfer (OFDST) [13]	**43.1**	45.9	**40.3**	**32.9**	**40.5**	
-	Yes	Baseline	33.0	38.0	37.0	22.1	32.5	BN statistics **adapted**
Source-Free UDA	Yes	UBNA [11] (→CS)	**38.6**	33.2	33.4	20.8	31.5	
Continuous UDA	Yes	CBNA [12]	34.9	**39.5**	33.5	**28.3**	34.1	
Continuous UDA	No	OFDST [13]	38.5	37.3	**37.4**	24.2	**34.4**	

6

UDA vs. Source-Free UDA

ResNet101-based model (DeepLabv2) (if #: BN statistics during GTA5 training frozen)

Source	Task	Method	mIoU (%) on $\mathcal{D}^{\text{test}*}_{\text{CS}}$	$\mathcal{D}^{\text{test}*}_{\text{MV}}$	$\mathcal{D}^{\text{test}*}_{\text{BDD}}$	$\mathcal{D}^{\text{test}*}_{\text{ACDC}}$
$\mathcal{D}^{\text{full}}_{\text{GTA5}}$	-	Baseline (#)	41.0	46.0	**39.2**	32.1
	UDA	SAC [10]▶ (→CS)	**53.8**	<u>48.9</u>	40.2	35.6
		SAC [10] (→MV)	<u>49.6</u>	**51.3**	**45.3**	**39.7**
		SAC [10] (→BDD)	45.8	46.4	<u>44.9</u>	36.5
		SAC [10] (→ACDC)	41.1	44.7	38.5	<u>36.6</u>
	-	Baseline	33.0	38.0	37.0	22.1
	Source-Free UDA	UBNA [11] (→CS)	**38.6**	33.2	33.4	20.8
		UBNA [11] (→MV)	35.7	<u>43.1</u>	<u>37.3</u>	<u>27.2</u>
		UBNA [11] (→BDD)	35.9	36.9	35.8	22.4
		UBNA [11] (→ACDC)	36.0	**43.4**	<u>37.3</u>	**28.3**

SAC performs adaptive batch normalization (ABN) during pre-training with a source-only loss.
The loss is only computed on the source samples, but the minibatches consist of source and target samples.

When adapted to MV, SAC generalizes better than when adapted to CS.

Transformer-Based UDA

ResNet101- and MiT-B5-based models

Source	Backbone	Method	mIoU [%] on $\mathcal{D}^{test*}_{CS}$	$\mathcal{D}^{test*}_{MV}$	$\mathcal{D}^{test*}_{BDD}$	$\mathcal{D}^{test*}_{ACDC}$	Mean mIoU
$\mathcal{D}^{full}_{GTA5}$	ResNet101	w/o adaptation (DeepLabv2)	41.0	46.0	39.2	32.1	39.6
		SAC [10]▸ (→CS)	53.8	48.9	40.2	35.6	44.6
	MiT-B5	w/o adaptation (SegFormer [14])	44.5	49.8	42.6	36.8	43.4
		DAFormer [15] (→CS)	**67.1**	**60.2**	**52.5**	**44.7**	**56.1**

+11.2 % rel.

+29.3 % rel.

Source	Backbone	Method	$\mathcal{D}^{test*}_{CS}$	$\mathcal{D}^{test*}_{MV}$	$\mathcal{D}^{test*}_{BDD}$	$\mathcal{D}^{test*}_{ACDC}$	Mean mIoU
$\mathcal{D}^{train}_{CS}$	ResNet101	DeepLabv2	69.3	50.3	41.3	37.8	49.7
	MiT-B5	SegFormer [14]	**76.6**	**61.0**	**53.8**	**53.0**	**61.1**

28%

still a gap

0%

Overall, the UDA methods SAC [5] and even more DAFormer [14] are also strong DG methods

There is only a 28% (relative) performance gap to target-only training …

Domain Generalization vs. (Continuous) UDA

Source Domain	mIoU (%) on $\mathcal{D}^{test*}_{CS}$	$\mathcal{D}^{test*}_{MV}$	$\mathcal{D}^{test*}_{BDD}$	$\mathcal{D}^{test*}_{ACDC}$	Mean
$\mathcal{D}^{train}_{CS}$	69.3	50.3	41.3	37.8	49.7
$\mathcal{D}^{train}_{MV}$	61.9	67.2	53.7	50.2	58.2
$\mathcal{D}^{train}_{BDD}$	53.2	53.0	55.3	41.7	50.8
$\mathcal{D}^{train}_{ACDC}$	48.1	47.9	39.5	59.7	48.8
UDA SAC [10]▸ (GTA5 →CS)	53.8	48.9	40.2	35.6	44.6
UDA SAC [10] (GTA5 →MV)	49.6	51.3	45.3	39.7	46.5
UDA DAFormer [15] (GTA5 →CS)	**67.1**	**60.2**	**52.5**	**44.7**	**56.1**
WildNet [2] **DG**	45.8	47.1	41.7	-	-
Cont. UDA [13] (GTA5→)	43.1	45.9	40.3	32.9	40.5
$\mathcal{D}^{full}_{GTA5}$	41.0	46.0	39.2	32.1	39.6

← MV is a nicely generalizing real dataset!

← strong UDA / DG methods reduce the synth-to-real ga… But not completely.

← single DG methods are still a interesting research area

← continuous methods still need more development

← source-only baseline

Conclusions

- Recent UDA methods achive very stong performance (DAFormer[]), not only on the target domain, but on multiple unseen domains.
- DG does not reach UDA performance, but it outperforms soure-only training and np target data is necessary to train the network.
- Source-free Uda can be used to adapt a network after training, but the training process is important (frozen BN statistics)
- Continuous UDA is a fresh field of research and first small perofrmanecs improvements can be achieved.

KIDELTA LEARNING
Scalable AI for Automated Driving

Jan-Aike Termöhlen | TU Braunschweig

j.termoehlen@tu-bs.de

KI Delta Learning ist ein Projekt der KI Familie. Es wurde aus der VDA Leitinitiative autonomes und vernetztes Fahren initiiert und entwickelt und wird vom Bundesministerium für Wirtschaft und Klimaschutz gefördert.

KI FAMILIE

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

References

[1] **DRPC**: [X. Yue, et al. Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization Without Accessing Target Domain Data. In Proc. of ICCV, pages 2100–2110, Seoul, Korea, Oct. 2019]

[2] **FSDR**: [J. Huang, et al. FSDR: Frequency Space Domain Randomization for Domain Generalization. In Proc. of CVPR, pages 6891–6902, virtual, June 2021]

[3] **WildNet** [S. Lee, et al. WildNet: Learning Domain Generalized Semantic Segmentation from the Wild. In Proc. of CVPR, pages 9936– 9946, New Orleans, LA, USA, June 2022]

[4] **IBN-Net** [X. Pan, et al. Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net. In Proc. of ECCV, pages 464–479, Munich, Germany, Sept. 2018]

[5] **SAN+SAW** []

[6] **AdaptSegNet** [Y.-H. Tsai, et al. Learning to Adapt Structured Output Space for Semantic Segmentation. In Proc. of CVPR, pages 7472–7481, Salt Lake City, UT, USA, June 2018]

[7] **DACS** [W. Tranheden, et al. DACS: Domain Adaptation via CrossDomain Mixed Sampling. In Proc. of WACV, pages 1379– 1389, Waikoloa, HI, USA, Jan. 2021]

[8] **IAST** [K. Mei, et al. Instance Adaptive Self-Training for Unsupervised Domain Adaptation. In Proc. of ECCV, pages 415–430, Glasgow, UK, Aug. 2020]

[9] **ADVENT** [T.-H. Vu, et al. ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation. In Proc. of CVPR, pages 2517–2526, Long Beach, CA, USA, Jun. 2019]

[10] **SAC** [N. Araslanov, S. Roth. Self-Supervised Augmentation Consistency for Adapting Semantic Segmentation. In Proc of CVPR, pages 15384–15394, virtual, Jun. 2021]

[11] **UBNA** [M. Klingner, et al. Unsupervised BatchNorm Adaptation (UBNA): A Domain Adaptation Method for Semantic Segmentation Without Using Source Domain Representations. In Proc. of WACV-Workshops, pages 210-220, Honolulu, HI, USA, Jan. 2022.]

[12] **CBNA** [M. Klingner, et al. Continual BatchNorm Adaptation (CBNA) for Semantic Segmentation. IEEE Transactions on Intelligent Transportation Systems, 23(11):20899–20911, 2022.]

[13] **OFDST** [J.-A.Termöhlen, et al. Continual Unsupervised Domain Adaptation for Semantic Segmentation by Online Frequency Domain Style Transfer. In Proc. of ITSC, pages 2881-2888, Nashville,TN, USA, Sep. 2021]

[14] **SegFormer** [E, Xie, et al. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In Proc. of NeurIPS, pages 12077-12090, Virtual, December 2021]

[15] **DAFormer** [L. Hoyer, et al. DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation. In Proc. of CVPR, pages 9924-9935, New Orleans, LA, USA June 2022]