Test-Time Adaptation for Segmentation

Klara Janouskova

About me

- PhD student at Visual Recognition Group,

  CTU in Prague, 2023-?

  Supervisor: Jiri Matas
- Previous experience
  - Equilibre Technologies: Financial time-series
  - Technion Vista Lab: Test-Time Adaptation for Segmentation with Chaim Baskin and Alex Bronstein
  - IBM Research Zurich: Model-Assisted Labelling for Visual Inspection of Bridges with Mattia Rigotti, Ioana Giurgiu and Cristiano Malossi
  - UAB Barcelona: Weakly-supervised scene-text recognition with Dimosthenis Karatzas and Lluis Gomez

Talk Outline

- Domain shift and domain adaptation
- Domain adaptation scenarios and methods
- Test-time adaptation (TTA)
- Single-Image Test-Time Adaptation for Segmentation: Our work

Collaborators:

Jiri Matas, *Visual Recognition Group, CTU in Prague*

Chaim Baskin, Tamir Shor, *Technion - Israel Institute of Technology*

Image classification

Input: RGB Image

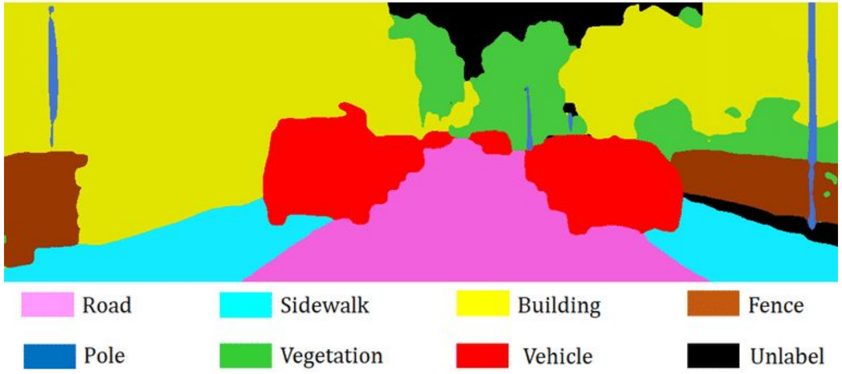
Output: Class probabilities

Dog: 0.95
Cat: 0.05
Plane: 0

Semantic segmentation

Input: RGB Image

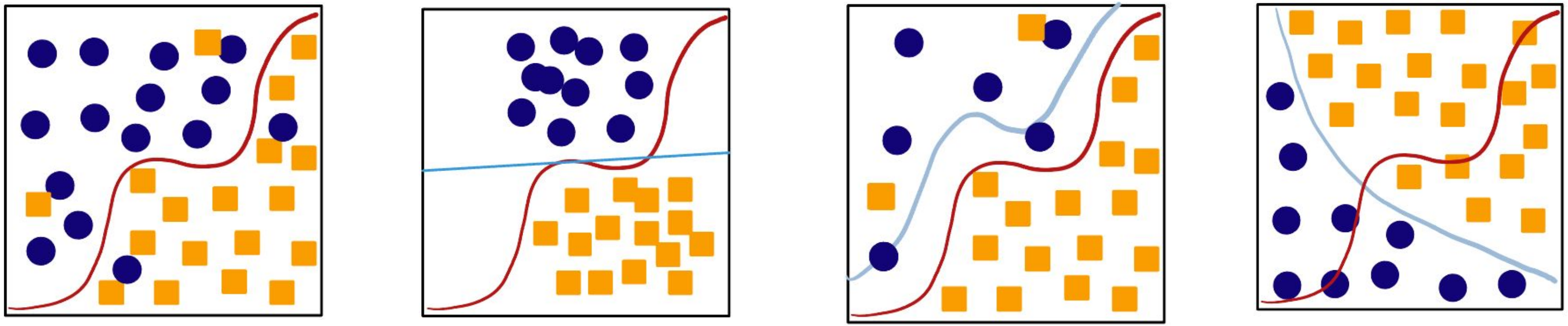
Output: Pixel level classification



Domain Shift

Change of distribution: Training (=source) $P_{\mathcal{S}} \rightarrow$ deployment (target) $P_{\mathcal{T}}$

source distribution | **covariate shift** | **prior (label) shift** | **concept shift**



X - input space, covariates
Y - output space

covariate shift: $P_{\mathcal{S}}(y|x) = P_{\mathcal{T}}(y|x)$, $P_{\mathcal{S}}(x) \neq P_{\mathcal{T}}(x)$

prior (label) shift: $P_{\mathcal{S}}(x|y) = P_{\mathcal{T}}(x|y)$, $P_{\mathcal{S}}(y) \neq P_{\mathcal{T}}(y)$

concept shift: $P_{\mathcal{S}}(y|x) \neq P_{\mathcal{T}}(y|x)$

learnt decision boundary on source and target distributions

Detecting Domain Shift

Supervised learning works well on training data distribution, but performance may drop arbitrarily under domain shift.

Detection of domain shift can be based on:

1. Performance on a subset of labelled target data → expensive, how often?
2. Input properties → is it indicative of model performance?
3. Classifier outputs properties → directly related to performance

Related: Novel class detection, anomaly detection

Suggested paper:
Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift

Dealing with Domain Shift

Options:

- Get new data, retrain (=remove the domain shift)
- Finetune on a small amount of data (=supervised domain adaptation)
- Prior shift: Prior shift adaptation ie based on confusion matrices
- Covariate shift: Most computer vision works

Link: Impossibility Theorems for Domain Adaptation Which assumptions suffice to provide performance guarantees on the success of domain adaptation algorithms?

Domain Adaptation (DA)

Motivation: Domain shift is the reason why a classifier performing well on the *evaluation set* performs poorly at deployment

Domain shift is common - few things do not change over time.

Examples:

- adapting a general LLM to medical documents
- diagnostics during an epidemic of a new disease
- people aging (personal identification system)

Prior Shift Adaptation

D - decision, Y - ground truth

Confusion matrix $C_{d|y}$ with values of P(D=i|Y=k)

$$p_{\mathcal{T}}(\mathbf{x}|Y) = p_{\mathcal{E}}(\mathbf{x}|Y) = \frac{p_{\mathcal{T}}(Y|\mathbf{x})p_{\mathcal{T}}(\mathbf{x})}{p_{\mathcal{T}}(Y)} = \frac{p_{\mathcal{E}}(Y|\mathbf{x})p_{\mathcal{E}}(\mathbf{x})}{p_{\mathcal{E}}(Y)}$$

$$p_{\mathcal{E}}(Y|\mathbf{x}) = p_{\mathcal{T}}(Y|\mathbf{x})\frac{p_{\mathcal{E}}(Y)p_{\mathcal{T}}(\mathbf{x})}{p_{\mathcal{T}}(Y)p_{\mathcal{E}}(\mathbf{x})} \propto p_{\mathcal{T}}(Y|\mathbf{x})\frac{p_{\mathcal{E}}(Y)}{p_{\mathcal{T}}(Y)}$$

$$p(D=i) = \sum_{k=1}^{K} p(D=i|Y=k)p(Y=k)$$

$$p(D) = \mathbf{C}_{d|y}p(Y)$$

$$\hat{p}_{\mathcal{E}}(Y) = \hat{\mathbf{C}}_{d|y}^{-1}\hat{p}_{\mathcal{E}}(D)$$

Milan Šulc
previous speaker

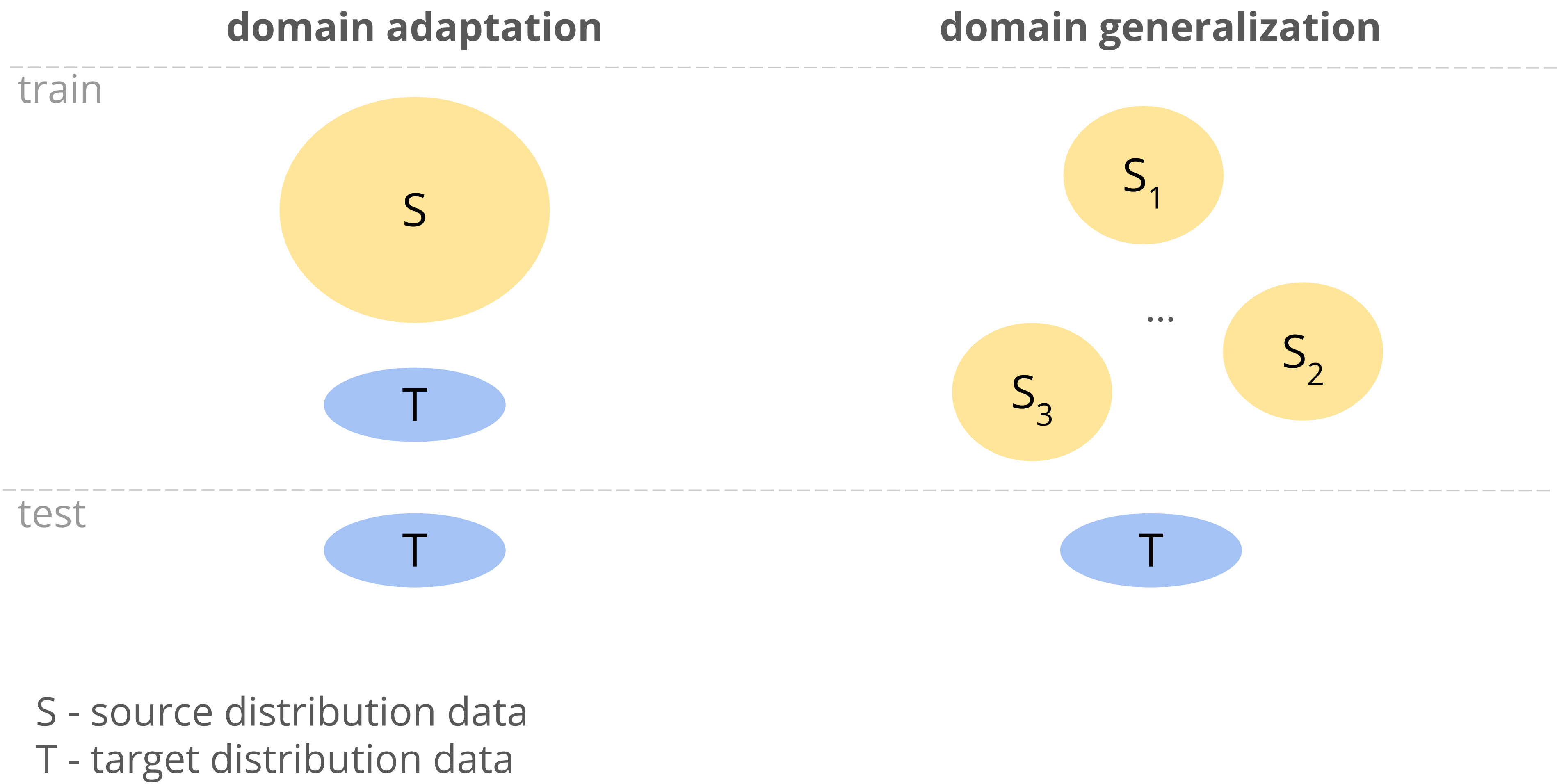
$$\boxed{\widehat{p}(\omega_i|\mathbf{x}) = \frac{\dfrac{\widehat{p}(\omega_i)}{\widehat{p}_t(\omega_i)}\widehat{p}_t(\omega_i|\mathbf{x})}{\displaystyle\sum_{j=1}^{n}\frac{\widehat{p}(\omega_j)}{\widehat{p}_t(\omega_j)}\widehat{p}_t(\omega_j|\mathbf{x})}}$$

Domain Adaptation Scenarios

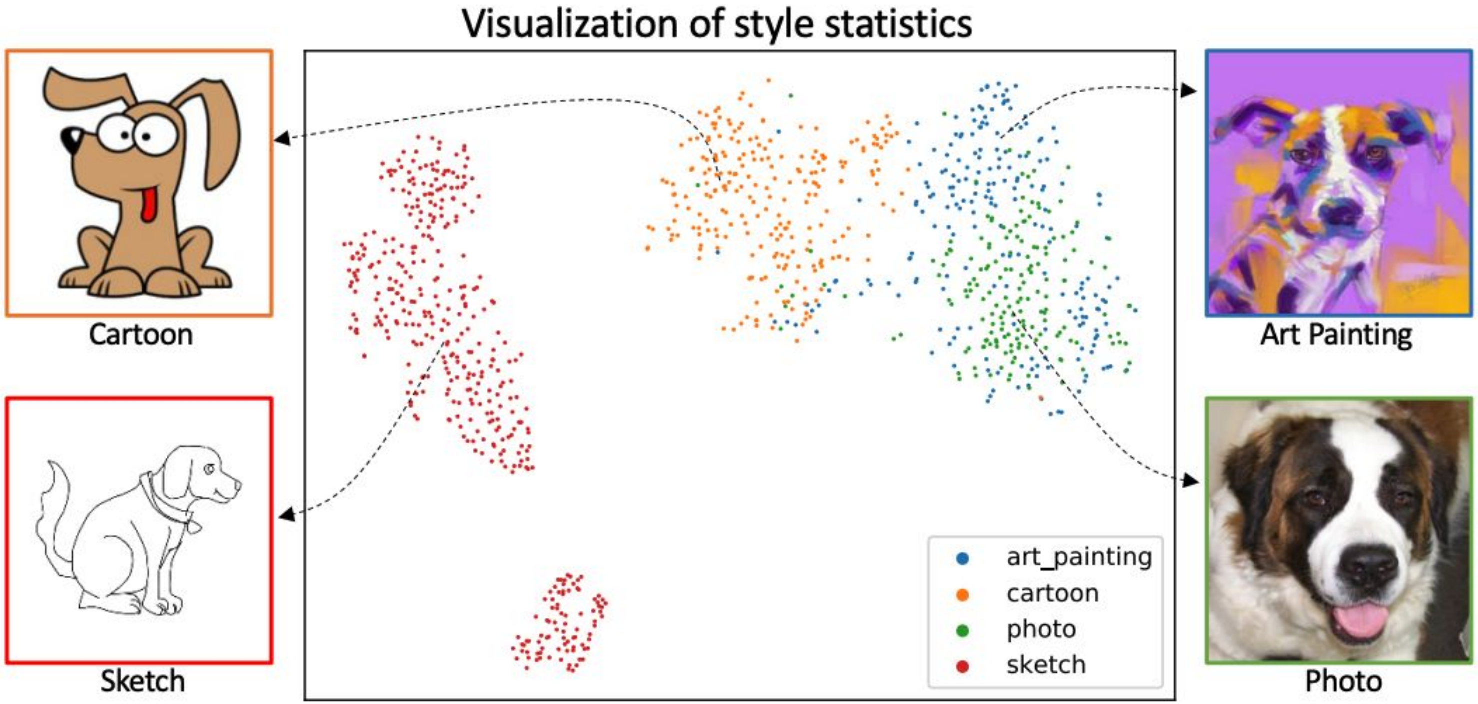
There are many realistic formulations, assuming whether

- labelled target data are available at training time - domain shift known in advance
- we have access to the training (source) data
- target distribution is static or changes continually

- samples at deployment time considered separately or all at once


domain adaptation
domain generalization
train
S
T
S_1
...
S_2
S_3
test
T
T
S - source distribution data
T - target distribution data

Domain generalization



Domain Generalization with MixStyle (arxiv)

Observation: Visual domain is closely related to style, which is encoded by bottom CNN layers.

Idea: Increase domain diversity of source data by style-mixing low-level features, inspired by adaptive instance normalization.

instance normalization

$$\mathrm{IN}(x) = \gamma \frac{x - \mu(x)}{\sigma(x)} + \beta$$

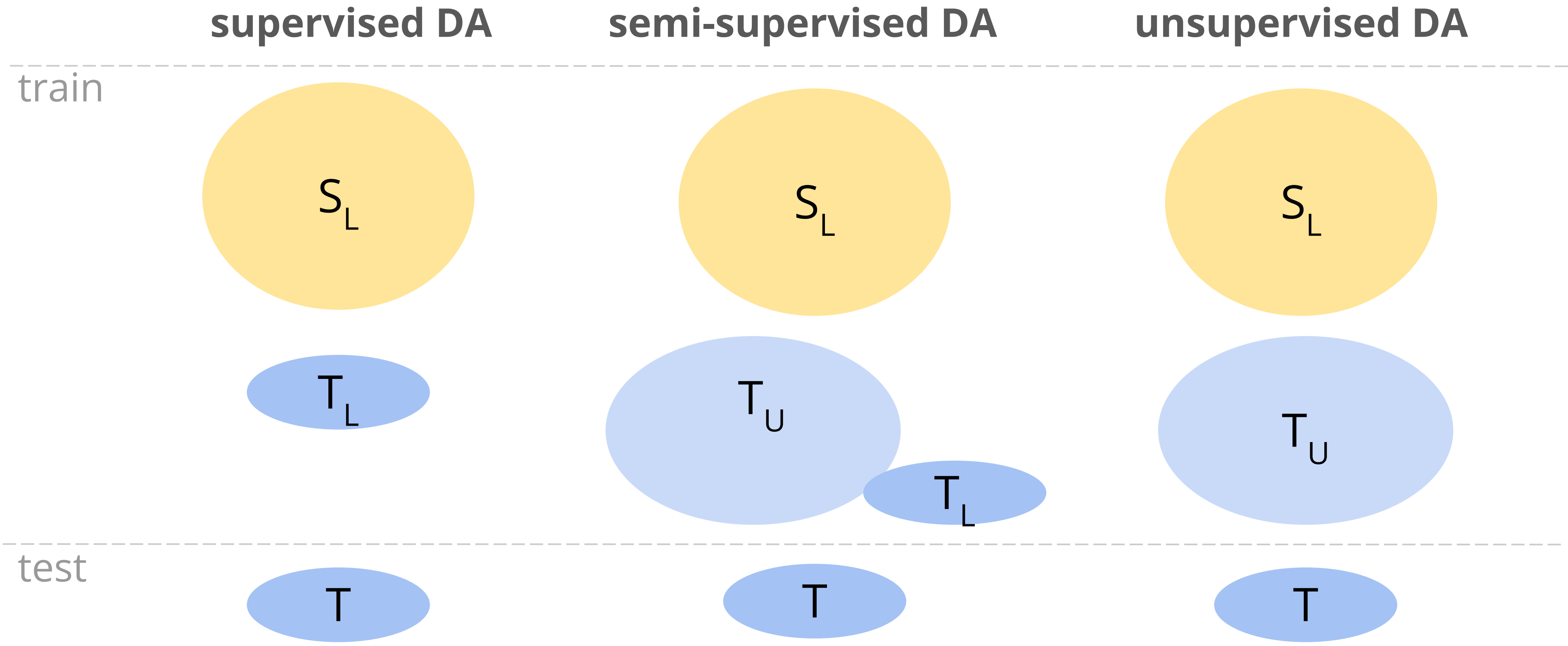
adaptive instance normalization

$$\mathrm{AdaIN}(x) = \sigma(y) \frac{x - \mu(x)}{\sigma(x)} + \mu(y).$$

mixStyle

$$\gamma_{mix} = \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x})$$
$$\beta_{mix} = \lambda\mu(x) + (1-\lambda)\mu(\tilde{x})$$

$$\mathrm{MixStyle}(x) = \gamma_{mix} \frac{x - \mu(x)}{\sigma(x)} + \beta_{mix}$$



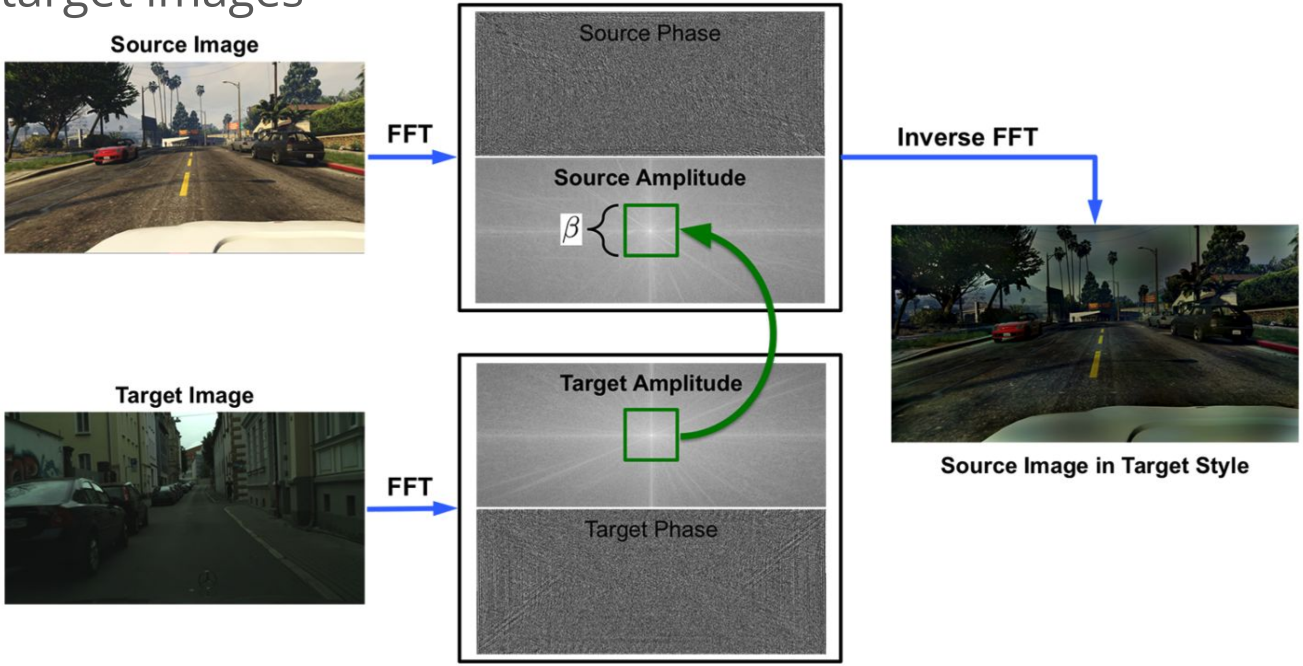
S - source distribution data
T - target distribution data

L - labelled data
U - unlabelled data

Fourier Domain Adaptation for Semantic Segmentation

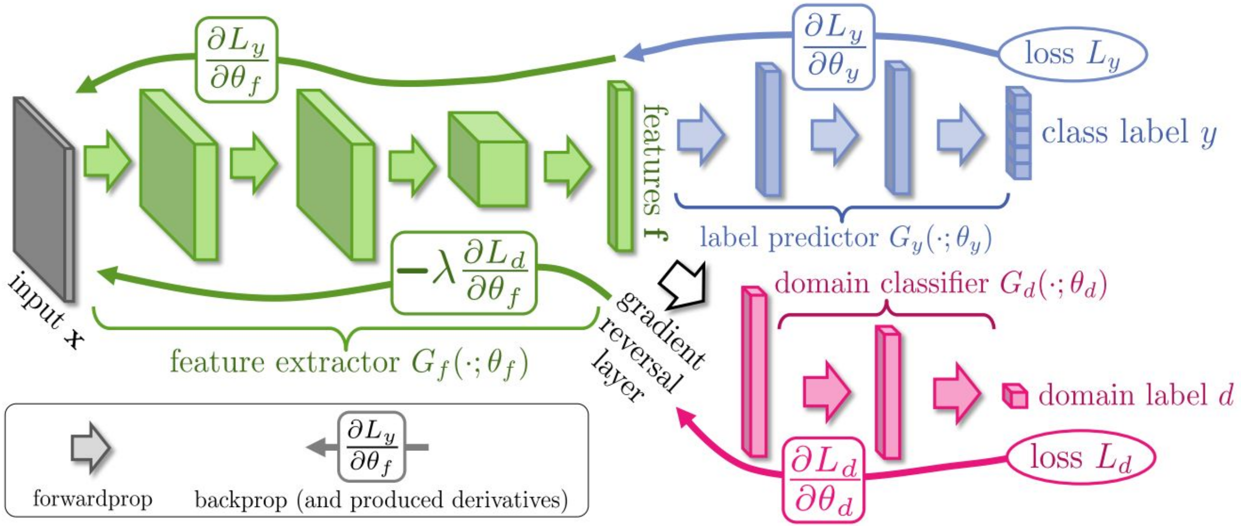
Link: [arxiv]

Unsupervised domain adaptation by replacing low-level frequencies of source images with those of target images



Unsupervised Domain Adaptation by Backpropagation

Link: https://proceedings.mlr.press/v37/ganin15.pdf



Multiply domain-classifier branch gradient to ensure similar feature distribution across domains

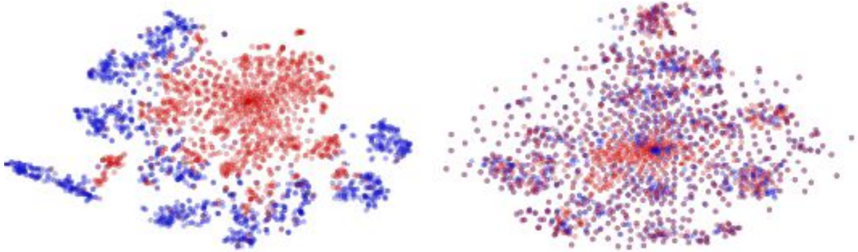
MNIST

Source

Target

MNIST-M

MNIST → MNIST-M: top feature extractor layer



(a) Non-adapted (b) Adapted

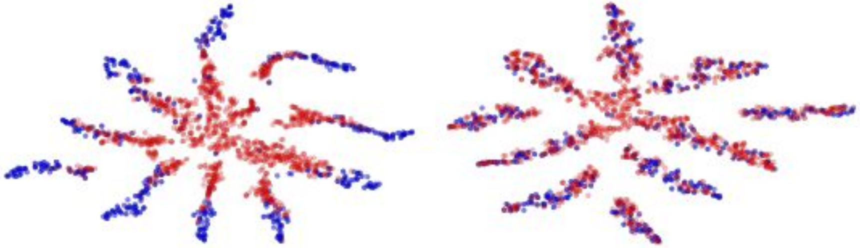
Syn Num

Source

Target

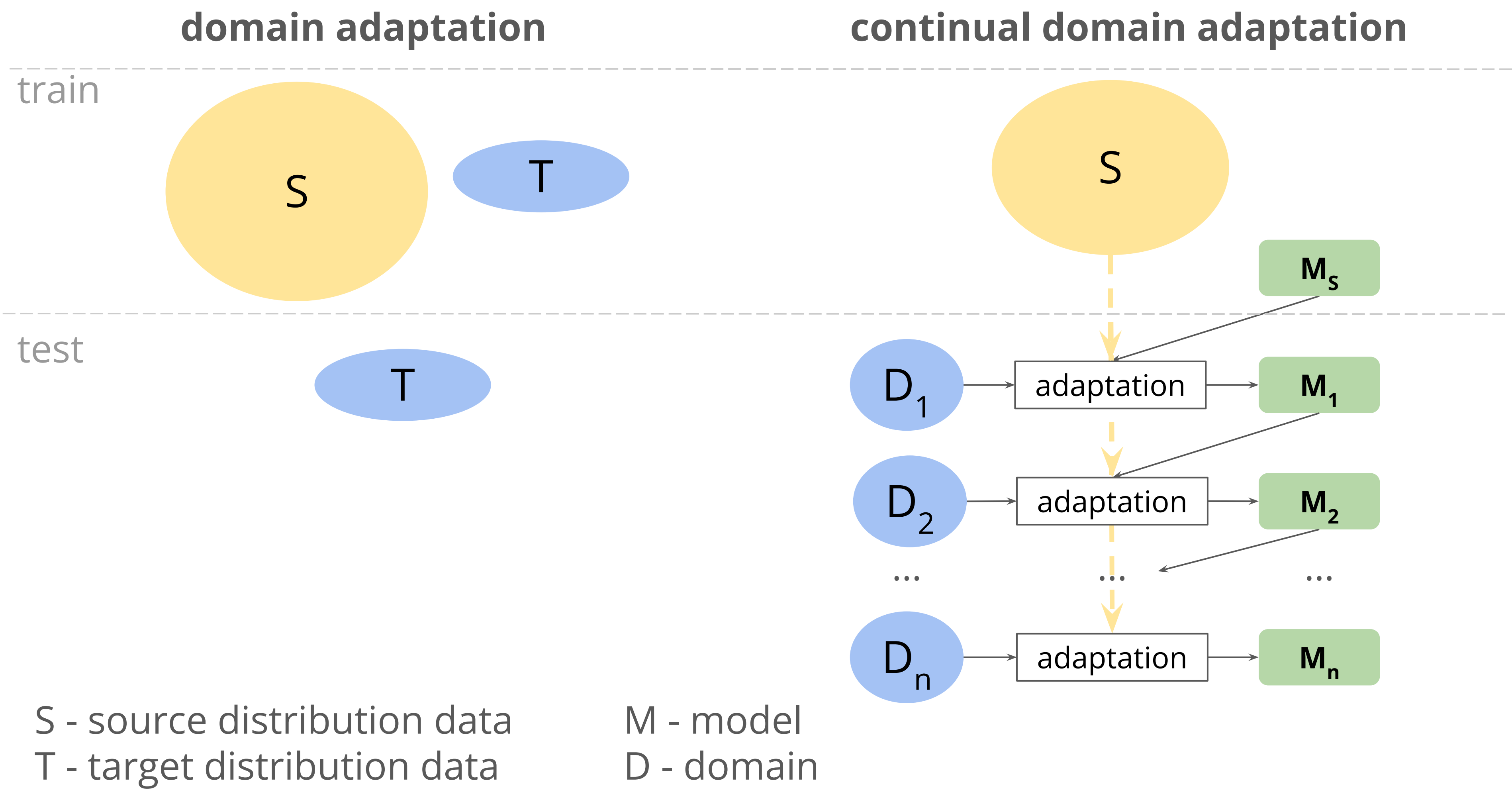
SVHN

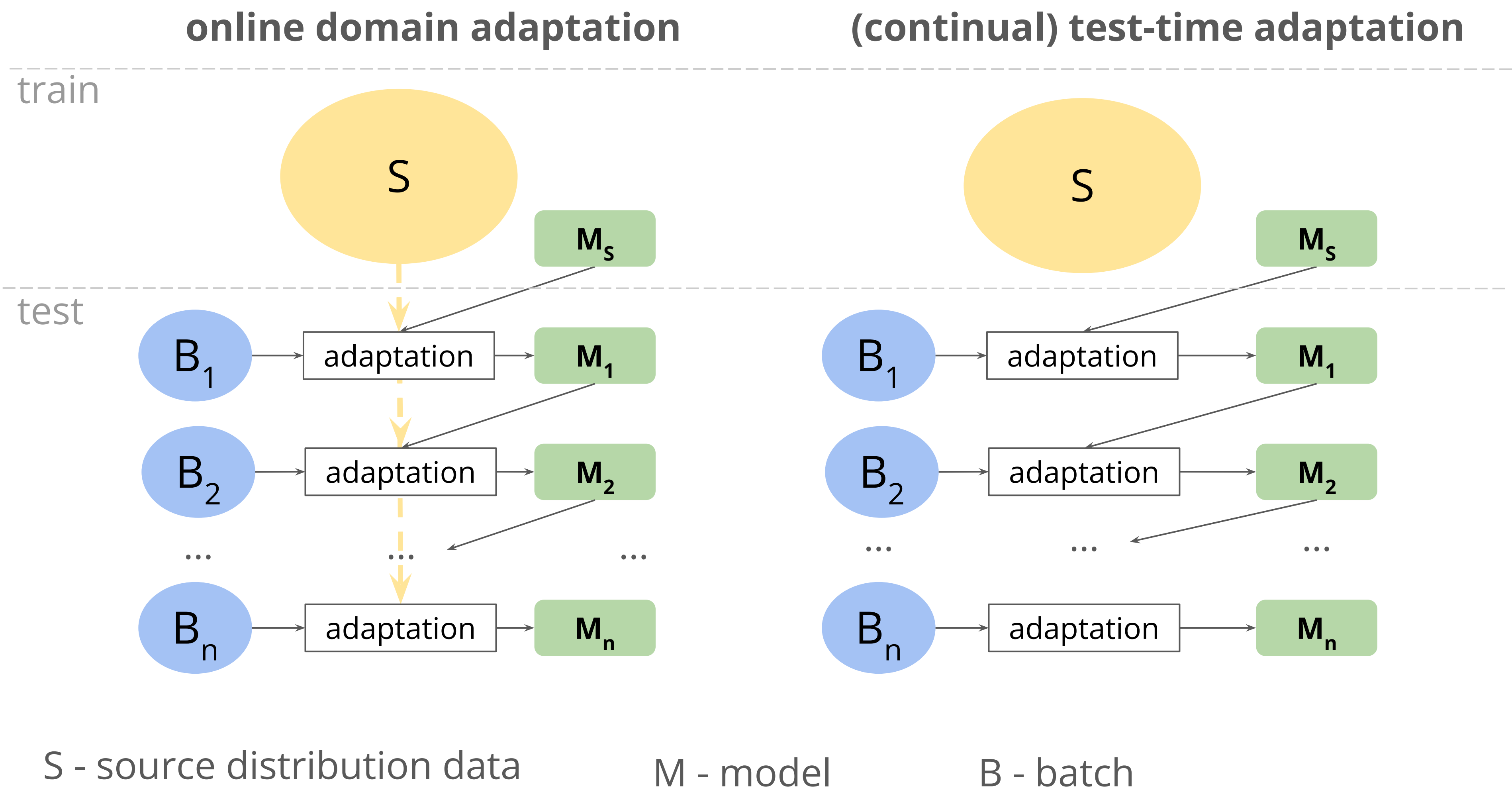
Syn Numbers → SVHN: last hidden layer of the label predictor



(a) Non-adapted (b) Adapted

source distribution samples target distribution samples

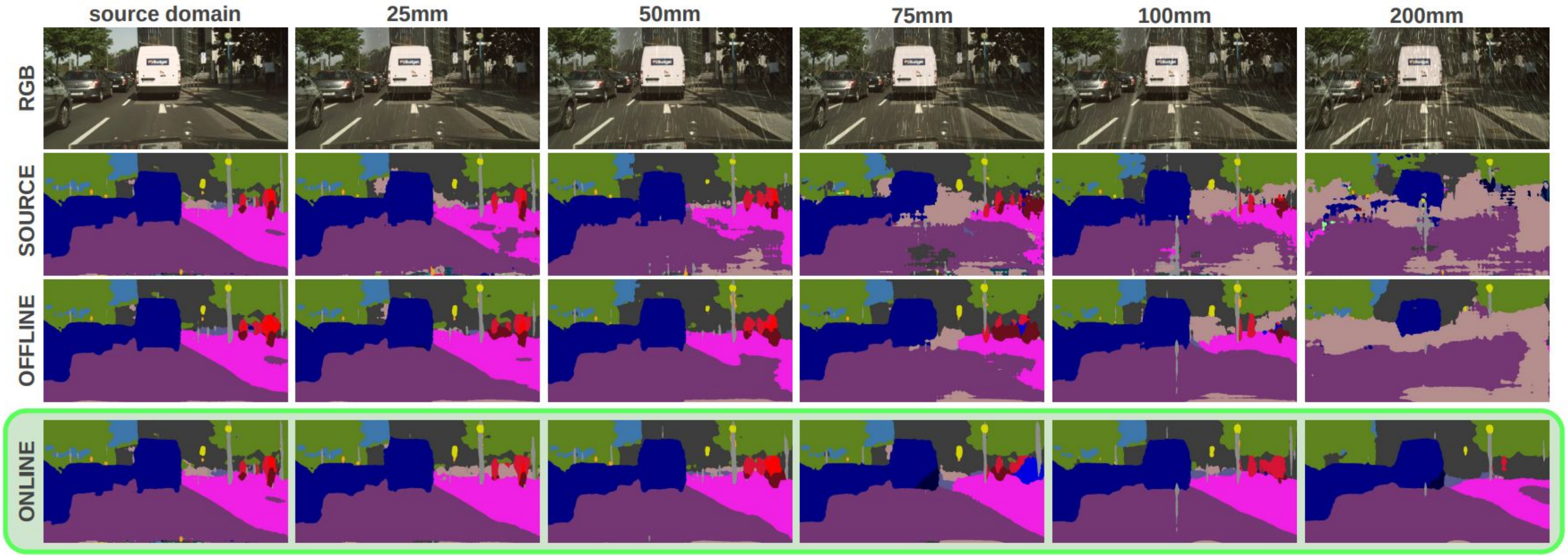

domain adaptation
continual domain adaptation
train
test
S
T
T
S
M_S
D_1
adaptation
M_1
D_2
adaptation
M_2
...
...
...
D_n
adaptation
M_n
S - source distribution data
T - target distribution data
M - model
D - domain

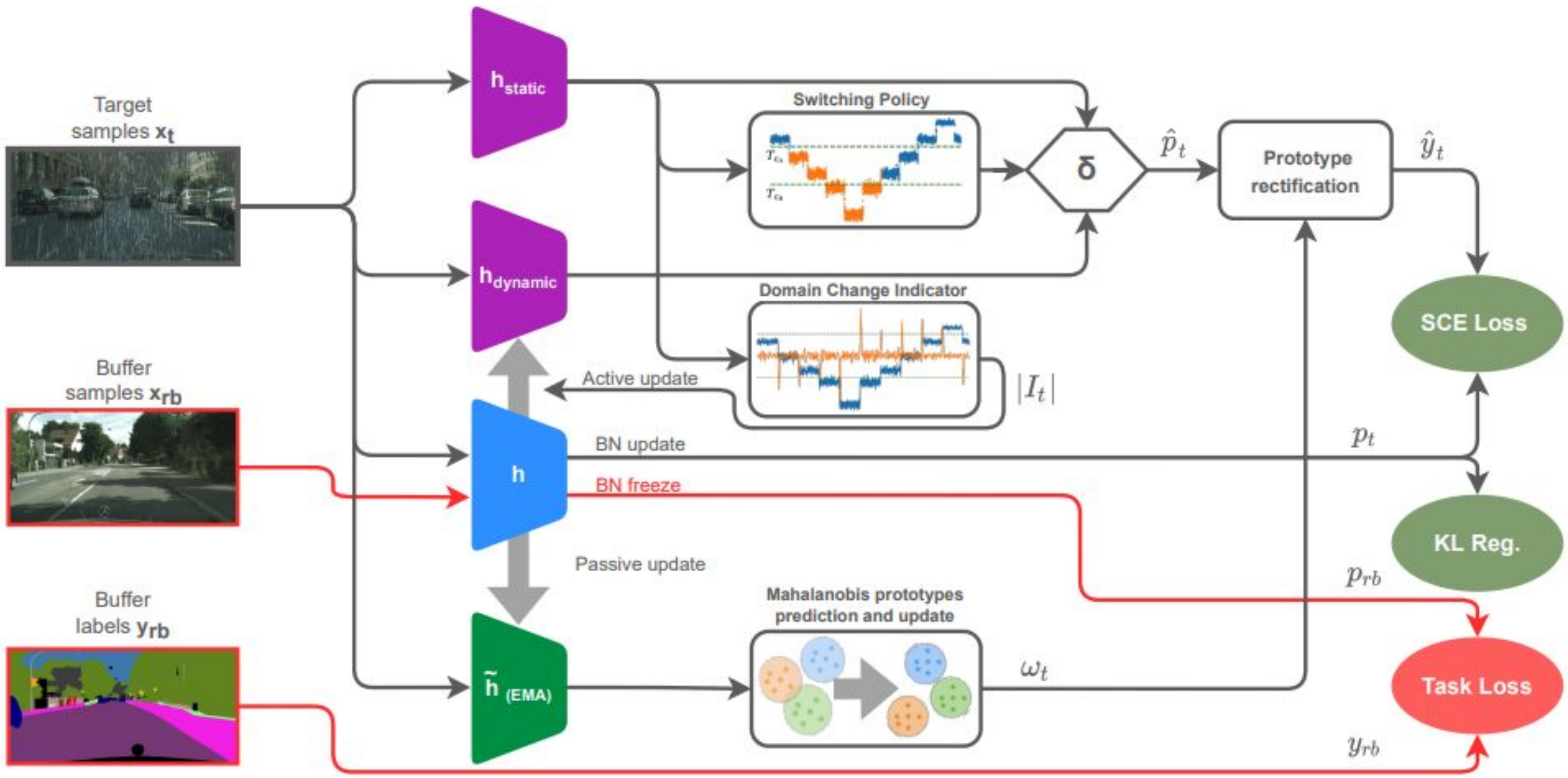


online domain adaptation
(continual) test-time adaptation
train
test
S
M$_s$
B$_1$
adaptation
M$_1$
B$_2$
adaptation
M$_2$
...
B$_n$
adaptation
M$_n$
S
M$_s$
B$_1$
adaptation
M$_1$
B$_2$
adaptation
M$_2$
...
B$_n$
adaptation
M$_n$
S - source distribution data
M - model
B - batch

Online Domain Adaptation for Semantic Segmentation in Ever-Changing Conditions

Link: https://arxiv.org/pdf/2207.10667.pdfarxiv





Complicated pipeline involving many different steps

Test-Time Adaptation (TTA)

Unsupervised, source-free (no training domain data) domain adaptation

Most methods are inspired by semi-supervised learning

Possible methods classifications:

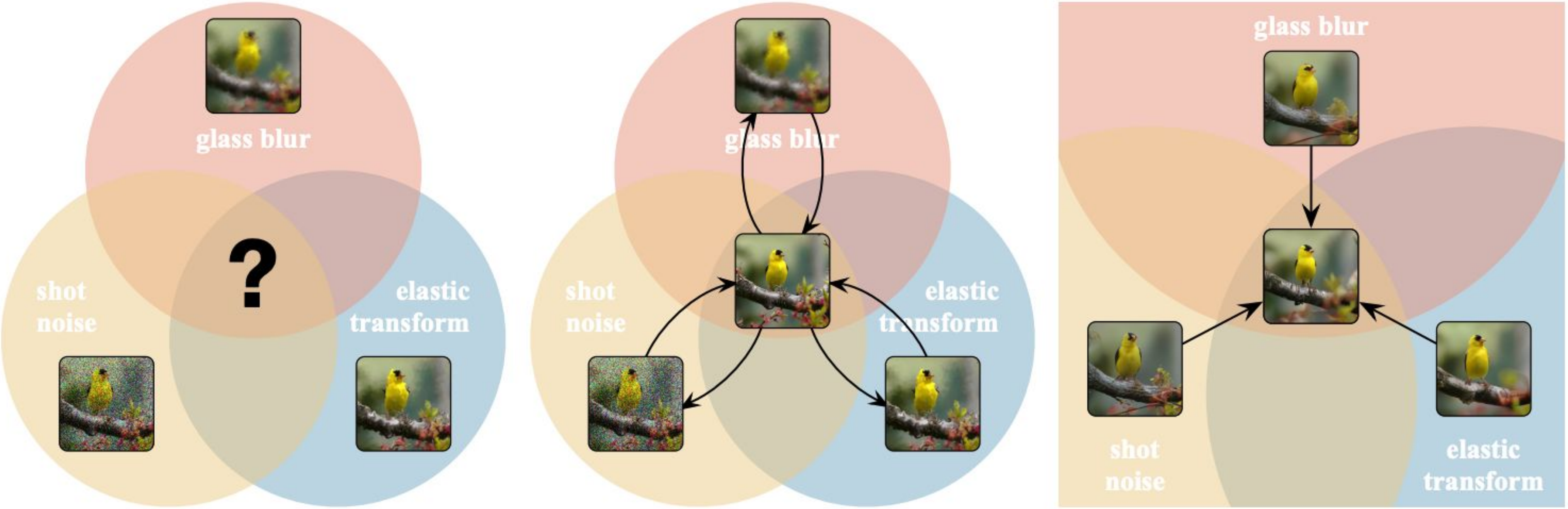
- input space adaptation
- feature space adaptation
- output space adaptation

- learnable parameter adaptation via self-supervised losses
- input/feature statistics adaptation, ie. batch-norm mean and variance
- prototype-based adaptation

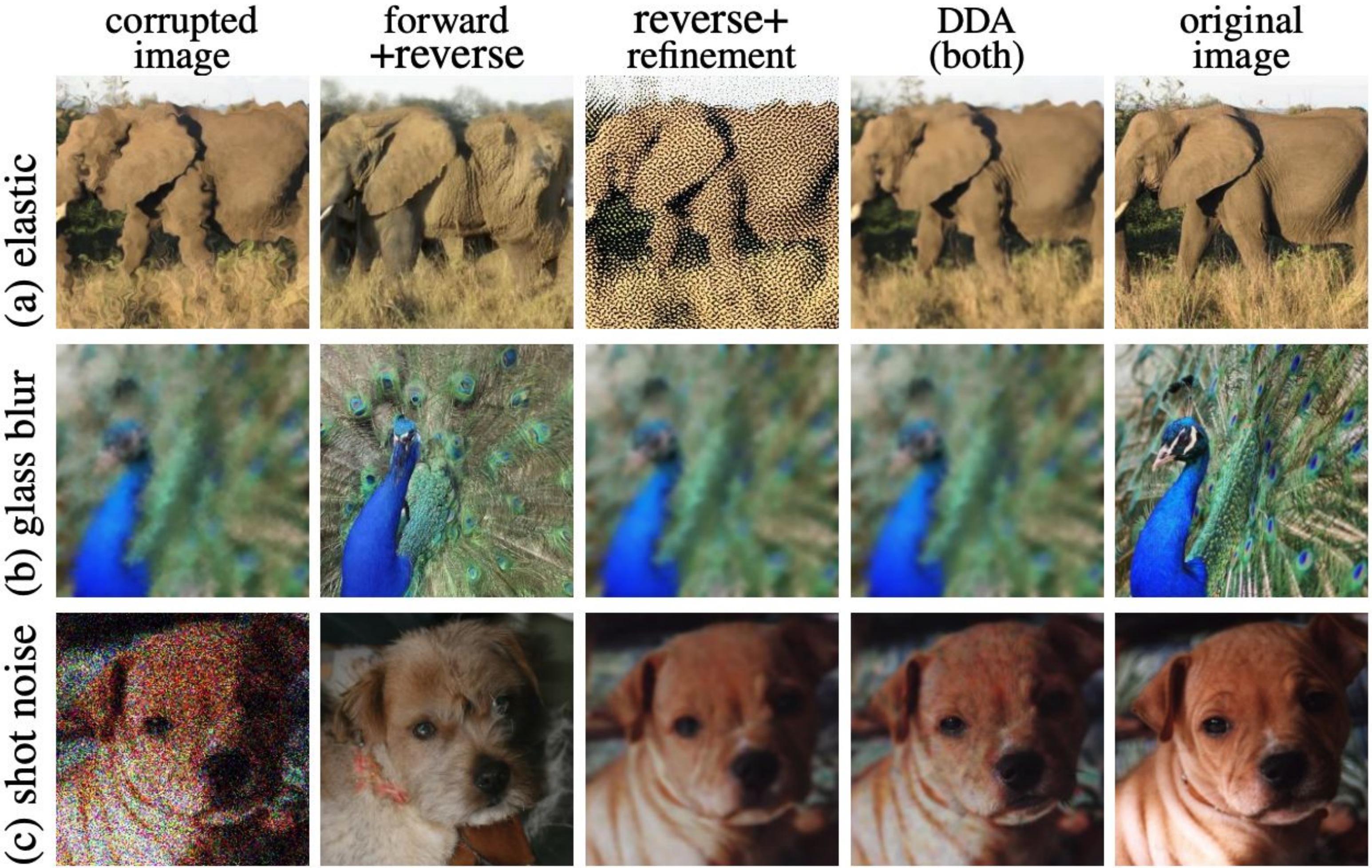
Input Space Adaptation

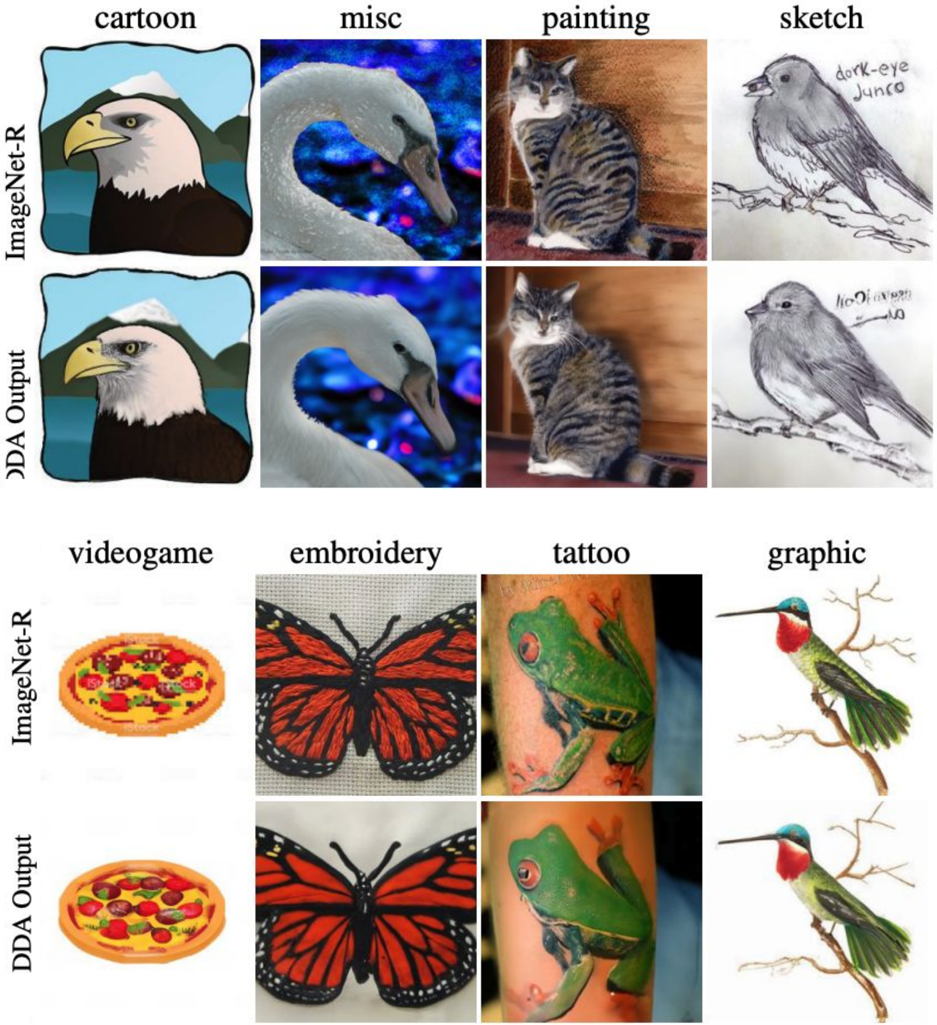
Back to the Source: Diffusion-Driven Test-Time Adaptation

Link: [arxiv]



(a) Setting: Multi-Target Adaptation (b) Cycle-Consistent Paired Translation (c) DDA (ours): Many-to-One Diffusion


corrupted image
forward +reverse
reverse+ refinement
DDA (both)
original image
(a) elastic
(b) glass blur
(c) shot noise

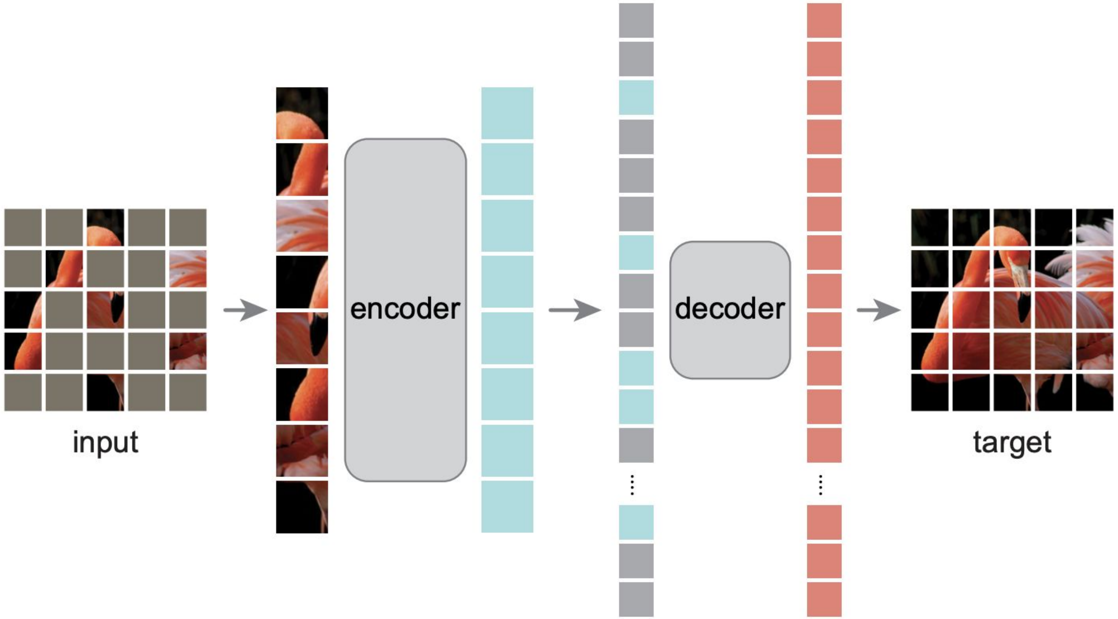


cartoon
misc
painting
sketch
ImageNet-R
DDA Output
videogame
embroidery
tattoo
graphic
ImageNet-R
DDA Output

Feature Space Adaptation

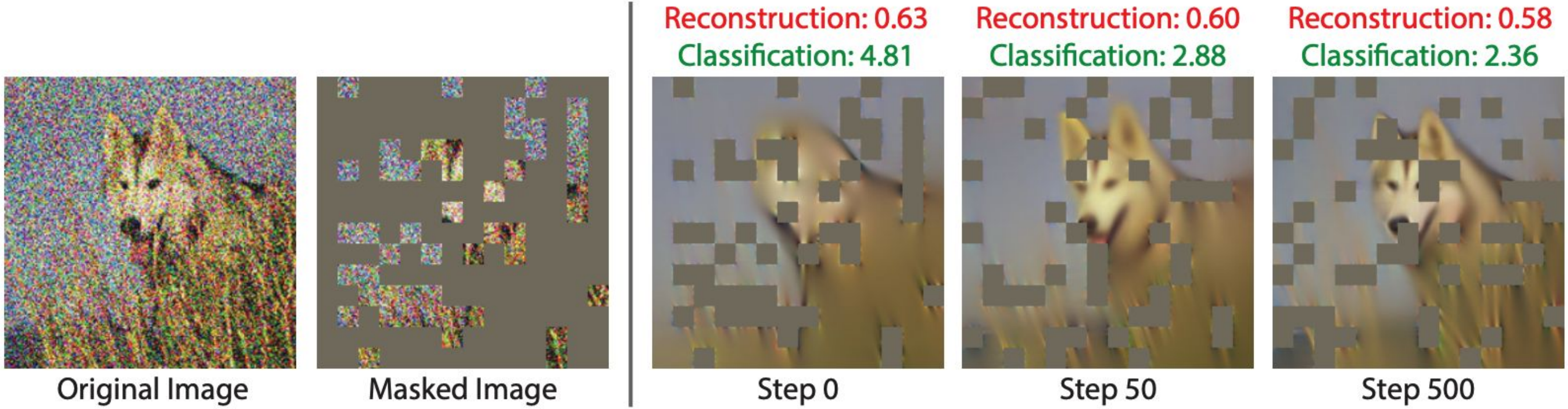
Test-Time Training with Masked Autoencoders

Link: [NeurIPS]



Source: *He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.*

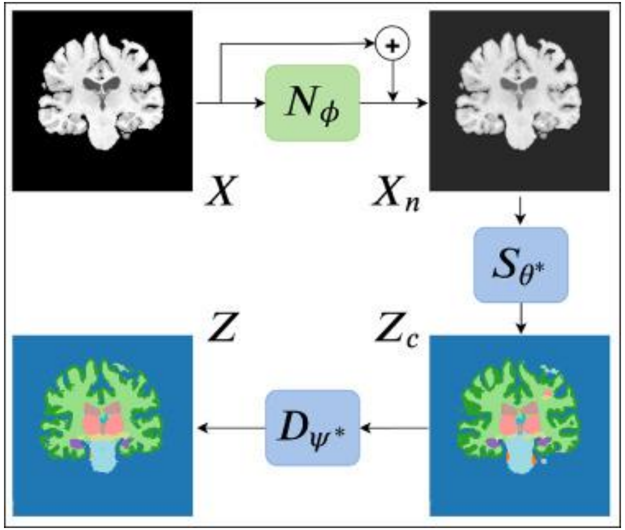
Model: Shared encoder, separate reconstruction and classification heads
Training time: Optimize classification and reconstruction loss jointly
Test time: Optimize shared encoder via reconstruction loss
Works with as little as a single image!



Output Space Adaptation

Test-time adaptable neural networks for robust medical image segmentation (link)

Learn a network translating output in the target domain to resemble outputs in the source domain. The translated output is used as supervision to update the image normalization module.

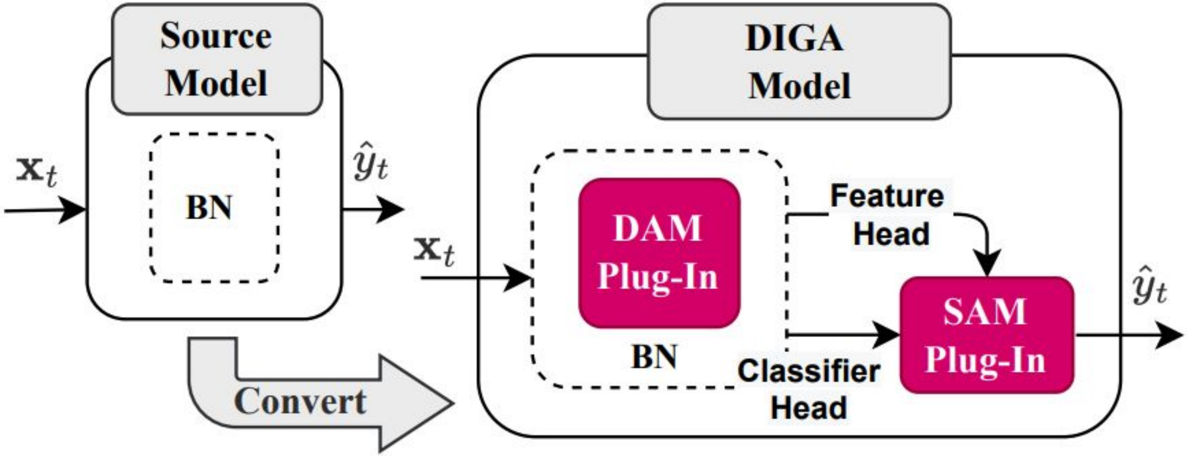


Batch-Norm (BN) Statistics and Prototype-Based Adaptation

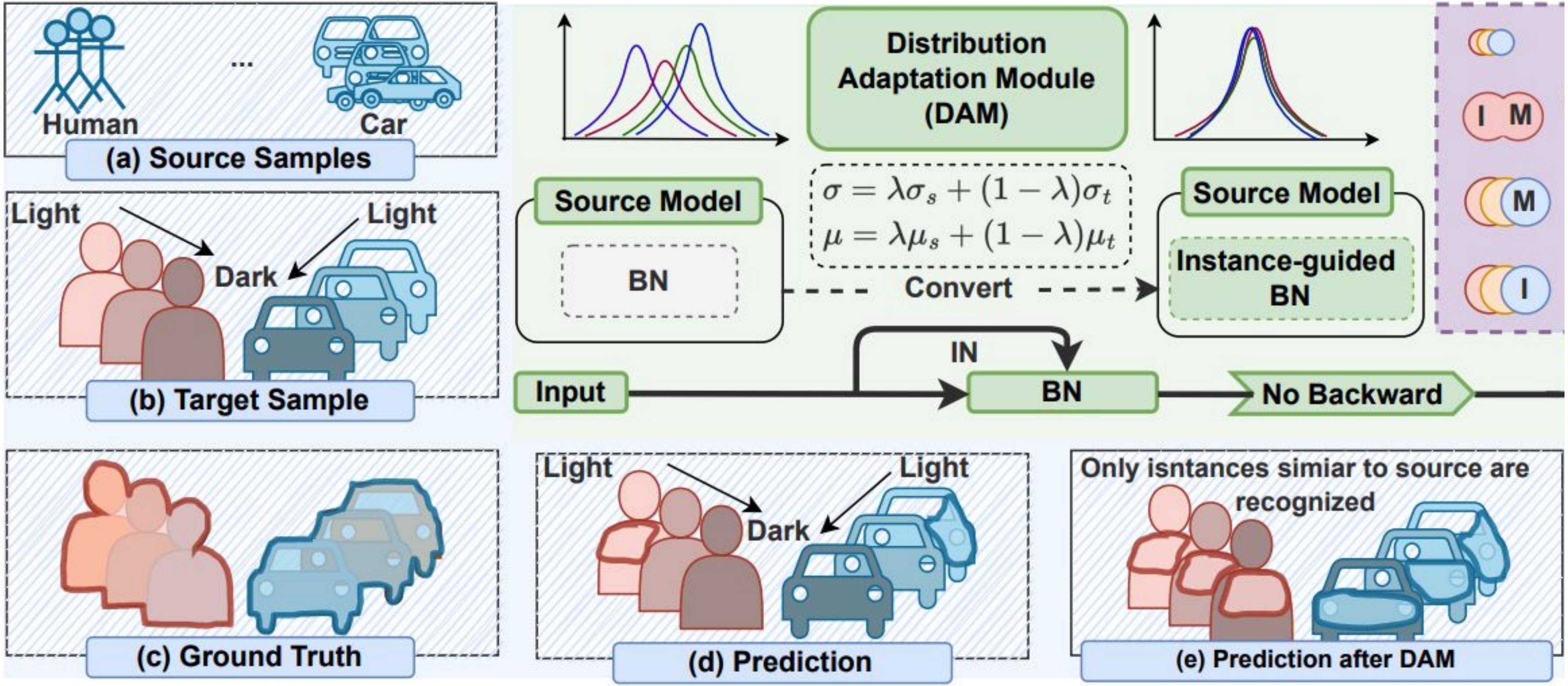
Dynamically Instance-Guided Adaptation: A Backward-free Approach for Test-Time Domain Adaptive Semantic Segmentation (link)

Distribution adaptation module - mixes instance and source BN statistics

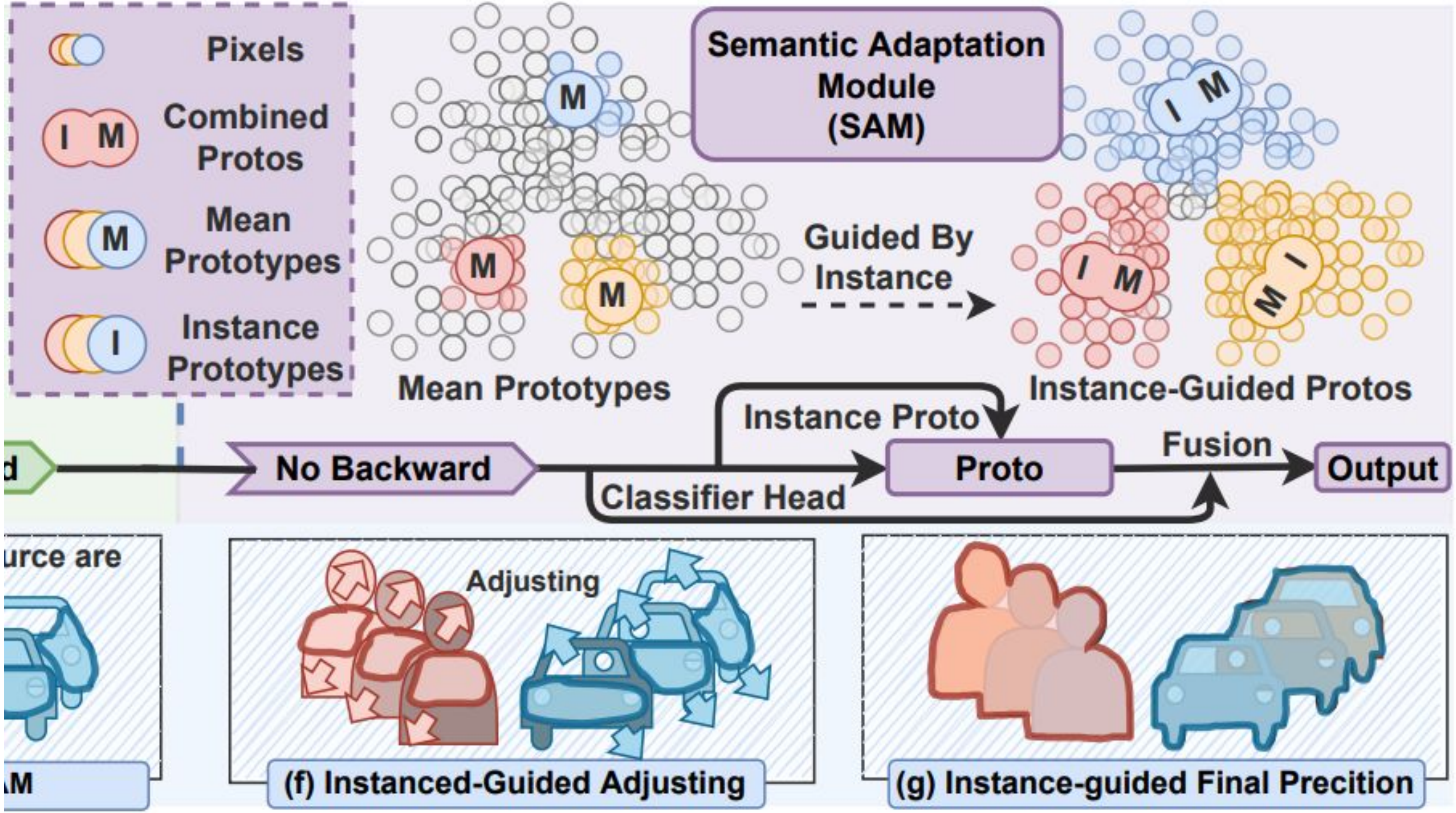
Semantic adaptation module - combines historical and instance-level prototypes to adjust predictions



Distribution adaptation



Semantic adaptation



Our Work

Single Image Test-Time Adaptation for Segmentation

State of research on Test Time Adaptation (TTA) for segmentation

- each work uses a very different setup
- comparison to few outdated baselines

Our work

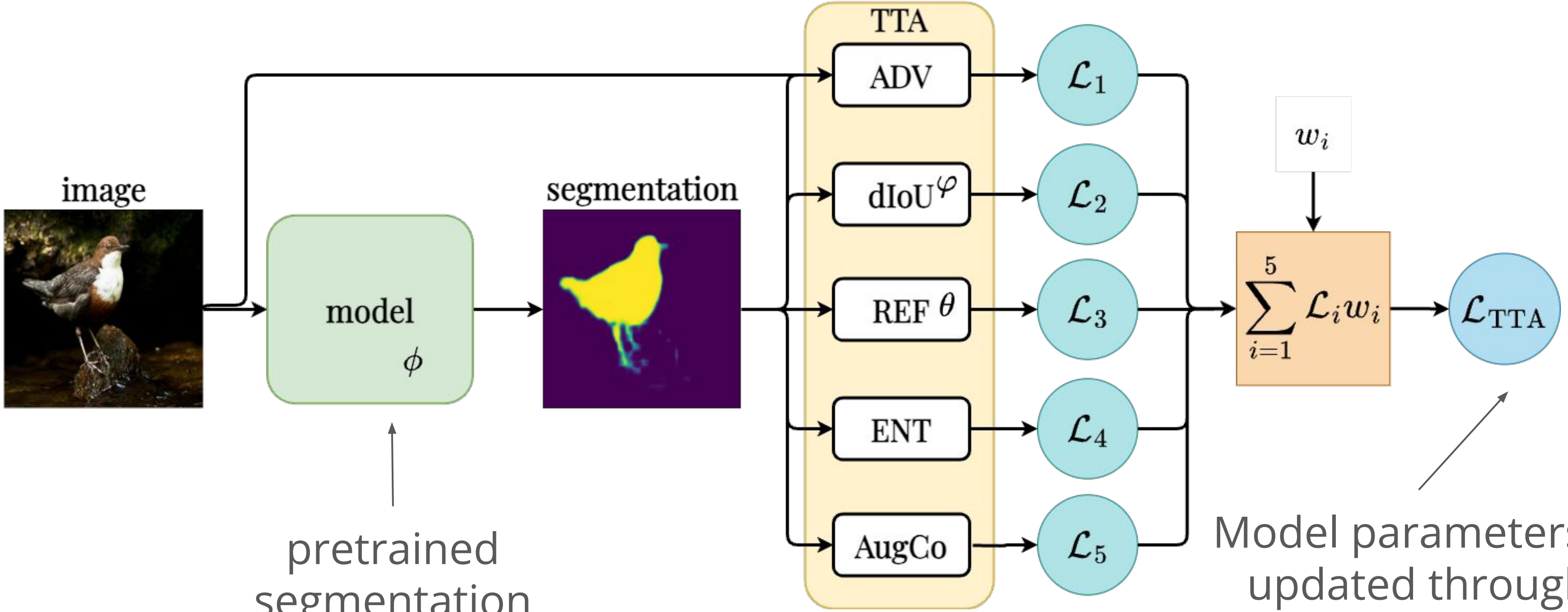
- adaptation to a single, isolated image at test-time
    - no issues with catastrophic forgetting, source parameters always restored
    - simplified setup for method analysis and comparison
- no assumptions about network architecture
    - BN-based methods can't be used
- diverse set of methods inspired by other tasks and domains
    - methods based on optimizing a self-supervised loss function

Segmentation and Domain Shift

Segmentation - assign a label to each pixel

Predicted by SAM[5]: SegmentAnything Model trained on a billion of masks released in April ‘23, SoTA

TTA with Self-Supervised Loss Functions



pretrained segmentation model

Model parameters ϕ updated through backpropagation of $\mathcal{L}_{\text{TTA}}$

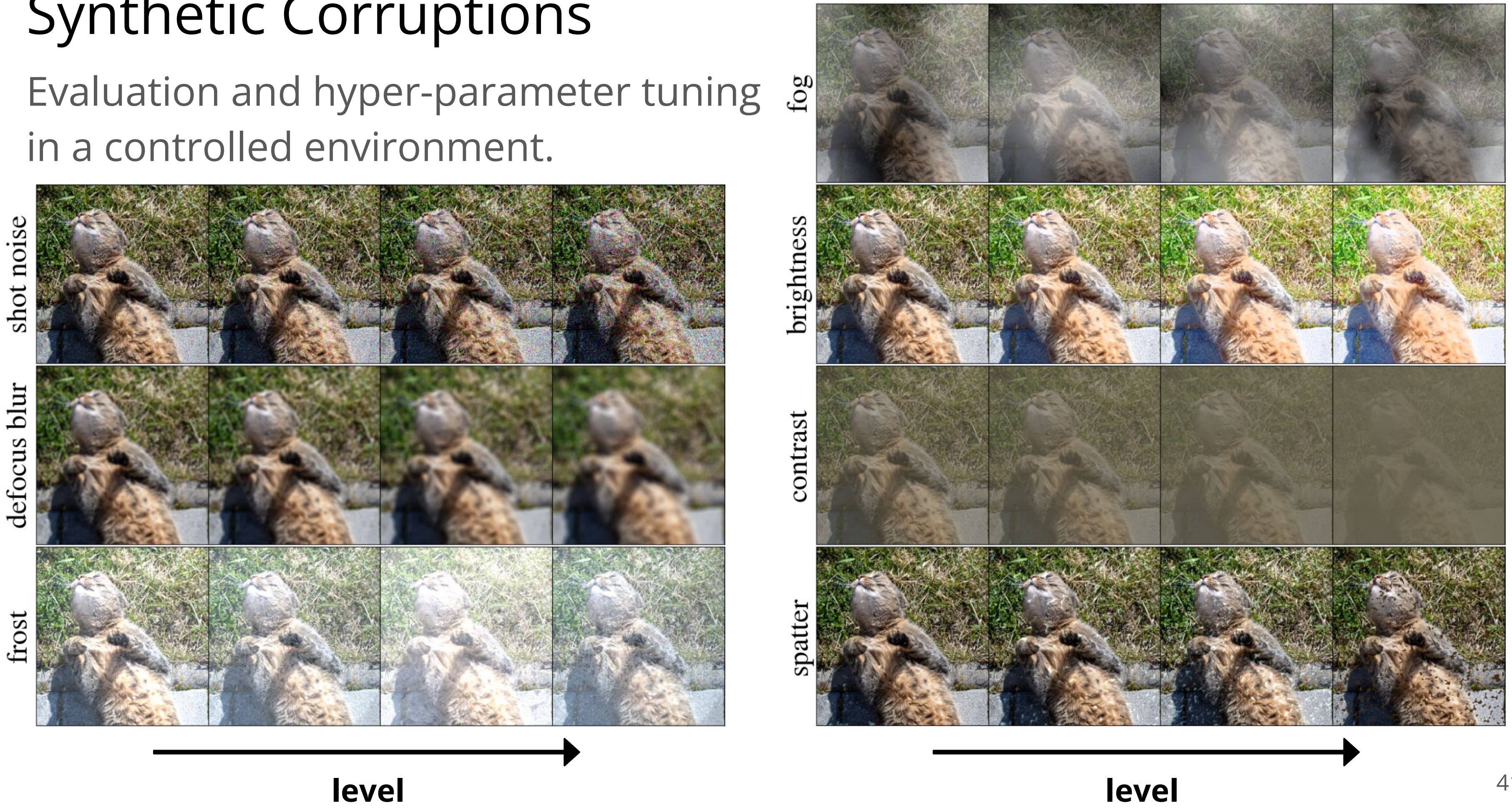
TTA hyper-parameters

Hyper-parameters considered: Number of adaptation iterations, learning rate, self-supervised loss parameters.

Deployment domain shift unknown → Use training set + synthetic corruptions

Synthetic Corruptions

Evaluation and hyper-parameter tuning in a controlled environment.

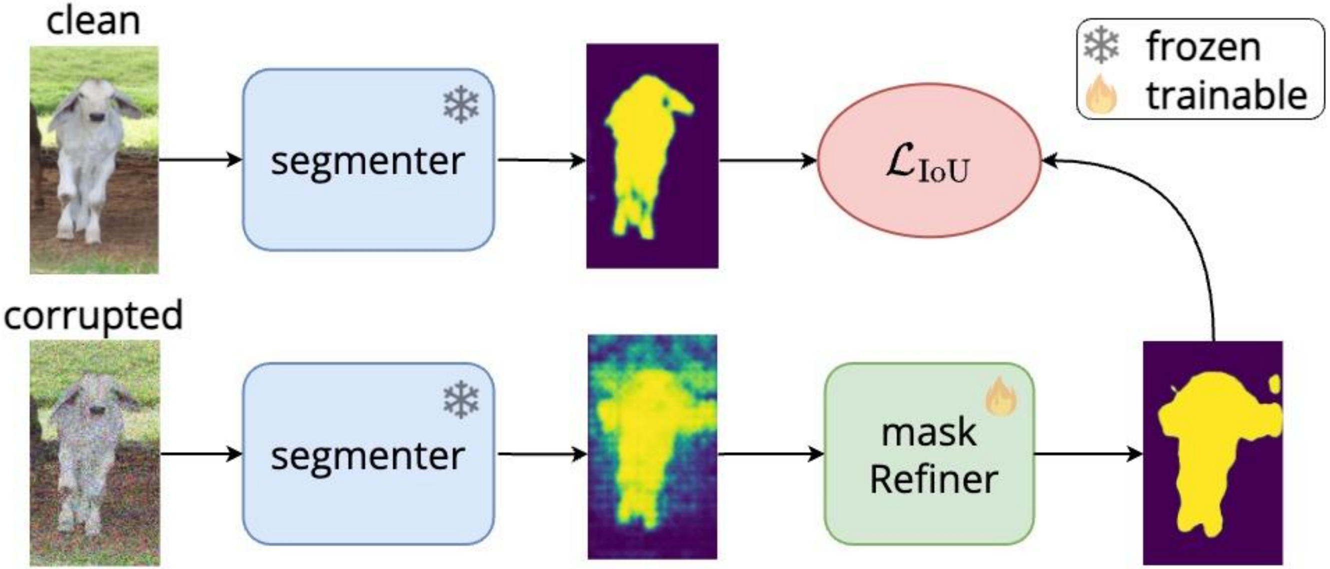


ENT: Entropy Minimization (Baseline)

$$\omega_n^\star, \varphi_n^\star = \underset{\omega_n, \varphi_n}{\operatorname{argmin}} \sum_{i=1}^{\mathrm{N}} \mathrm{s}_i \cdot \log(\mathrm{s}_i)$$

ω_n, φ_n	parameters of the normalization layers of the encoder and the decoder, respectively
$\mathrm{s} = d_s^\varphi \circ e^\omega(x)$	segmentation prediction of input image *x*
s_i	segmentation prediction for pixel *i*
N	total number of pixels

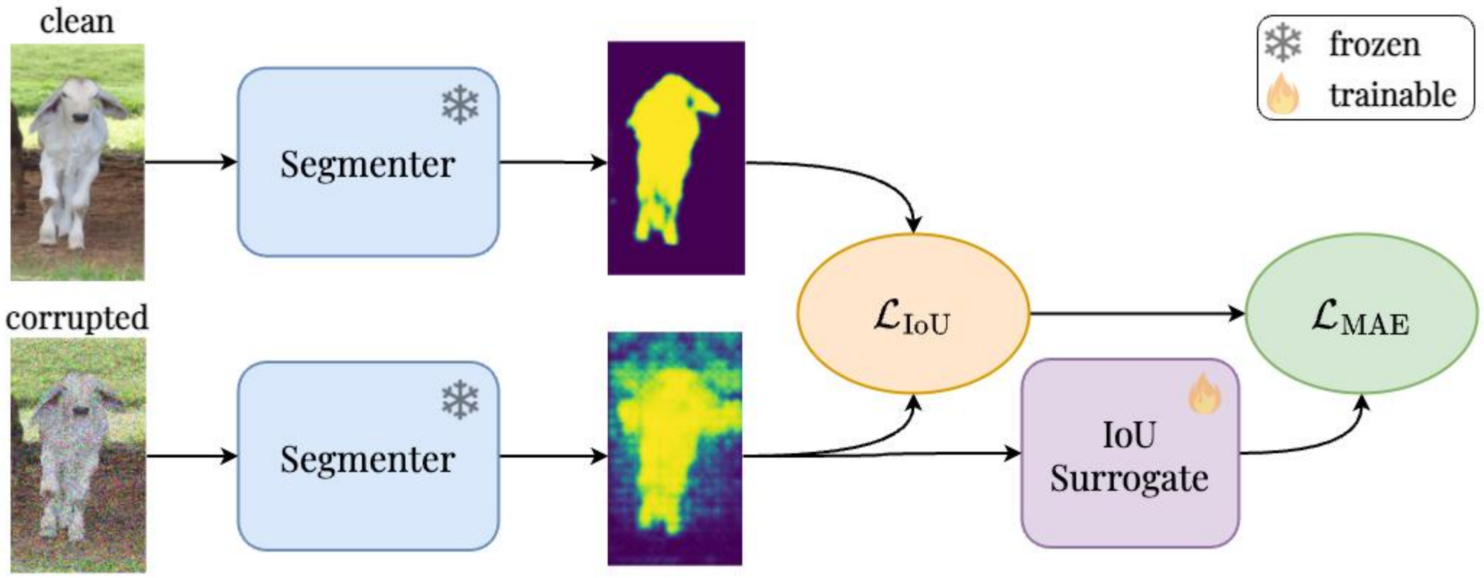
REF: Mask-Refinement-Based TTA



No prior knowledge about domain shift kind

→ images altered with targeted adversarial perturbations to produce corrupted segmentation.

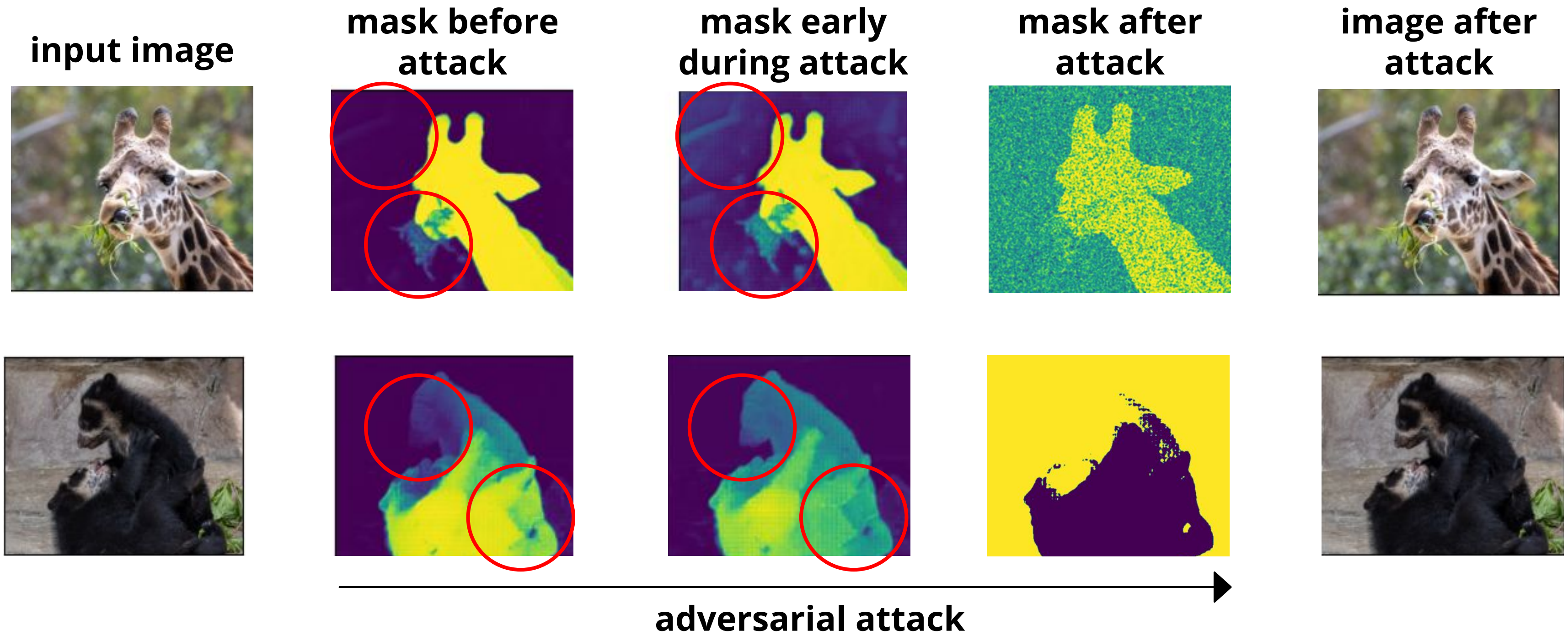
dIoU: Deeep IoU surrogate



Corruptions not known in advance - adversarial attack is used to corrupt the images!

Domain Shift Simulation: Adversarial Attack

Iteratively optimize an imperceptible perturbation of the image to change the model output. First iterations lead to very realistic mask corruptions.



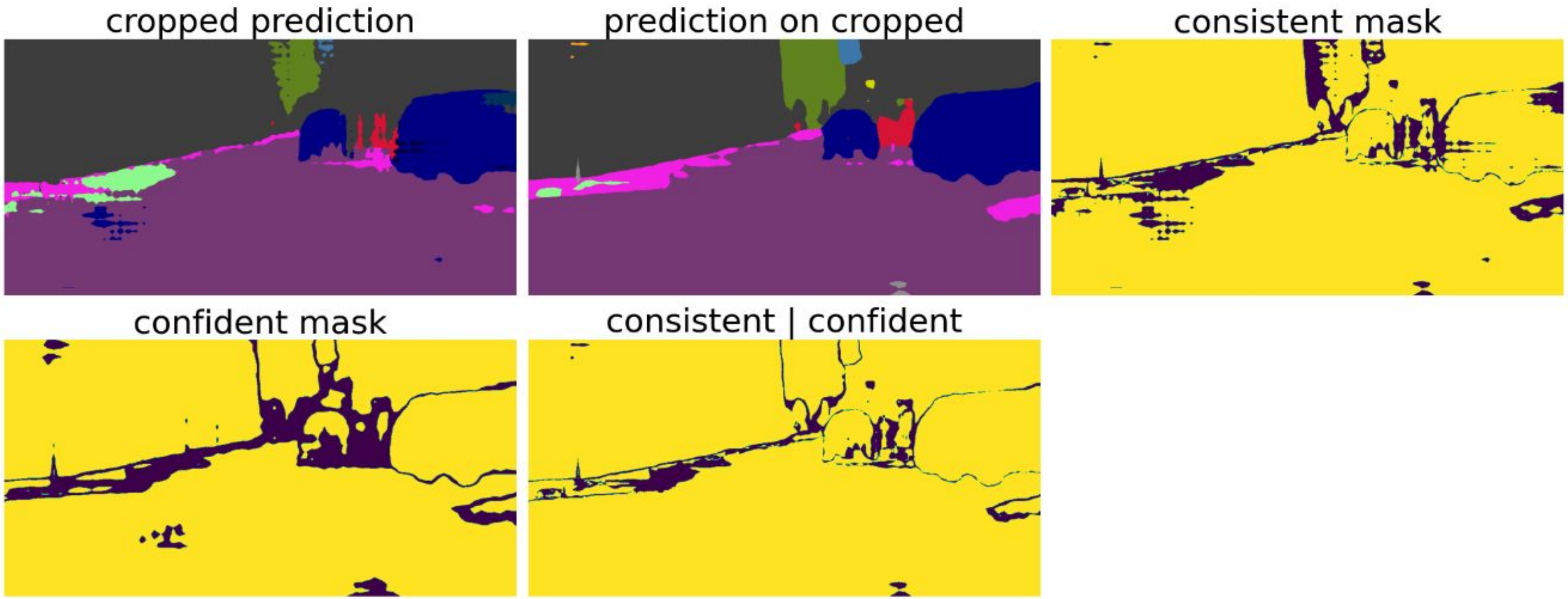
ADV: Adversarial Transformation

$$\omega_n^\star, \varphi_n^\star = \underset{\omega_n, \varphi_n}{\text{argmin}} \mathcal{L}_{\text{KL}}(\mathrm{s}, \mathrm{s}')$$

ω_n, φ_n	parameters of the normalization layers of the encoder and the decoder, respectively
$\mathrm{s} = d_s^\varphi \circ e^\omega(x)$	segmentation prediction of clean image *x*
$\mathrm{s}' = d_s^\varphi \circ e^\omega(x')$	segmentation prediction of corrupted image *x'*
$\mathcal{L}_{\text{KL}}(\mathrm{s}, \mathrm{s}')$	reverse KL divergence loss

AugCo: Augmentation Consistency

Only self-train on pixels with consistent between augmentations (crop, color jitter) or with high prediction confidence



Experiments

Training (source) domain:

Synthetic driving dataset (GTA5)

TTA learning rate, number of iterations:
Training set + synthetic corruptions

Deployment (unknown target domain):

- Real driving dataset (cityscapes)
- Real adverse weather condition driving datasets (ACDC)

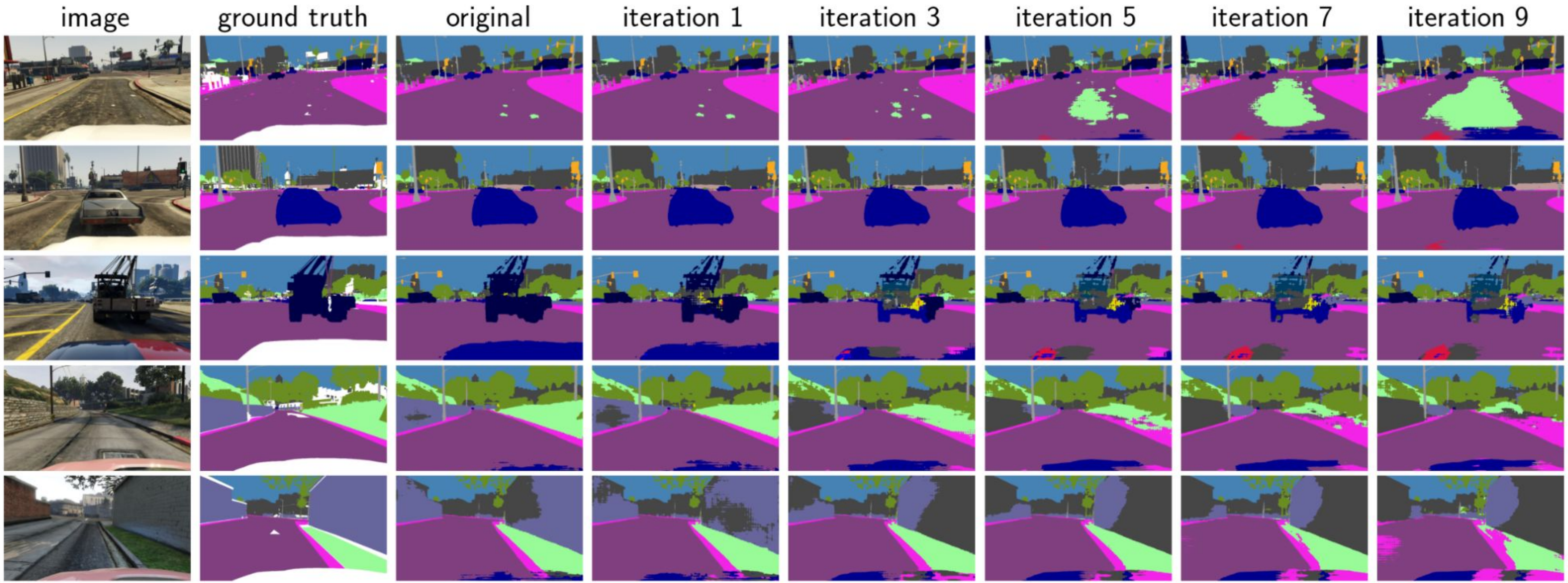
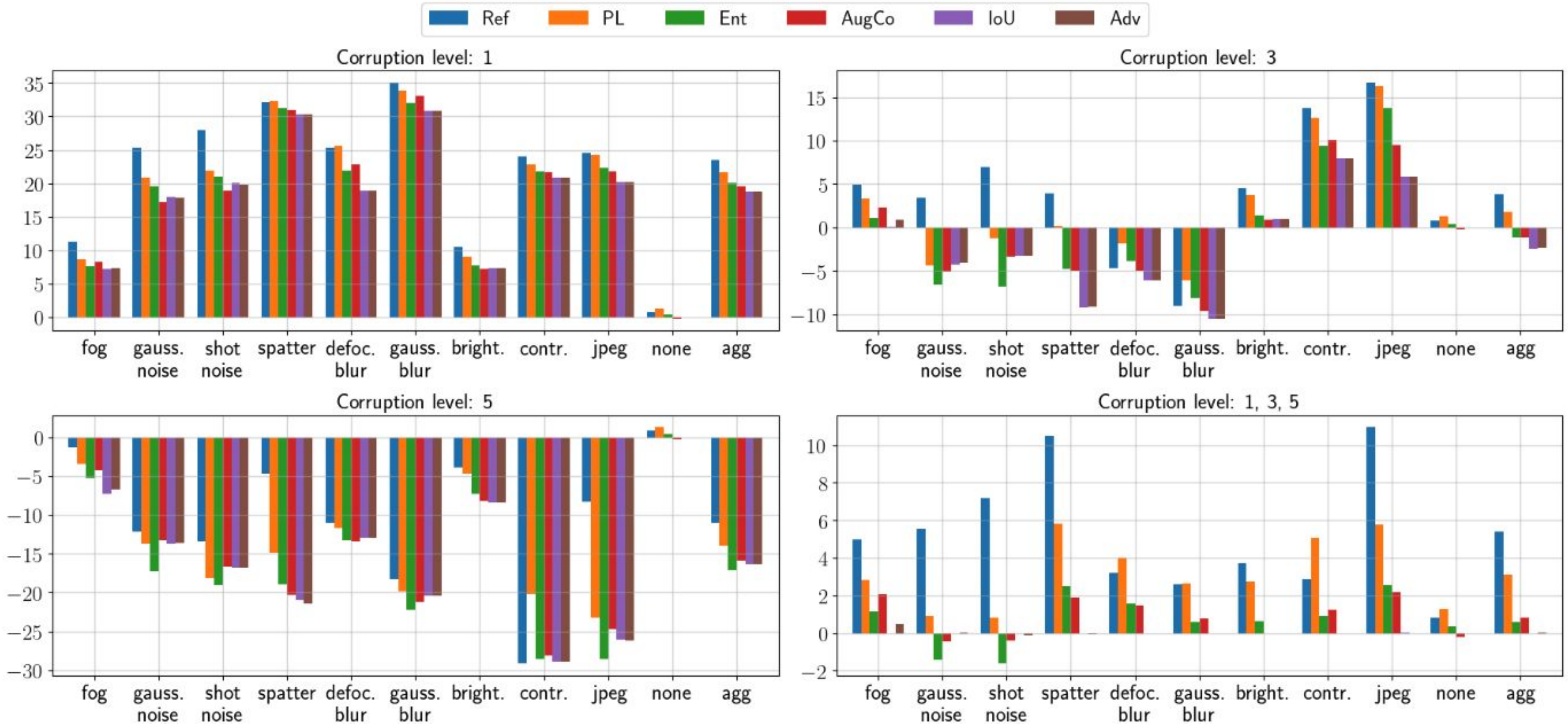


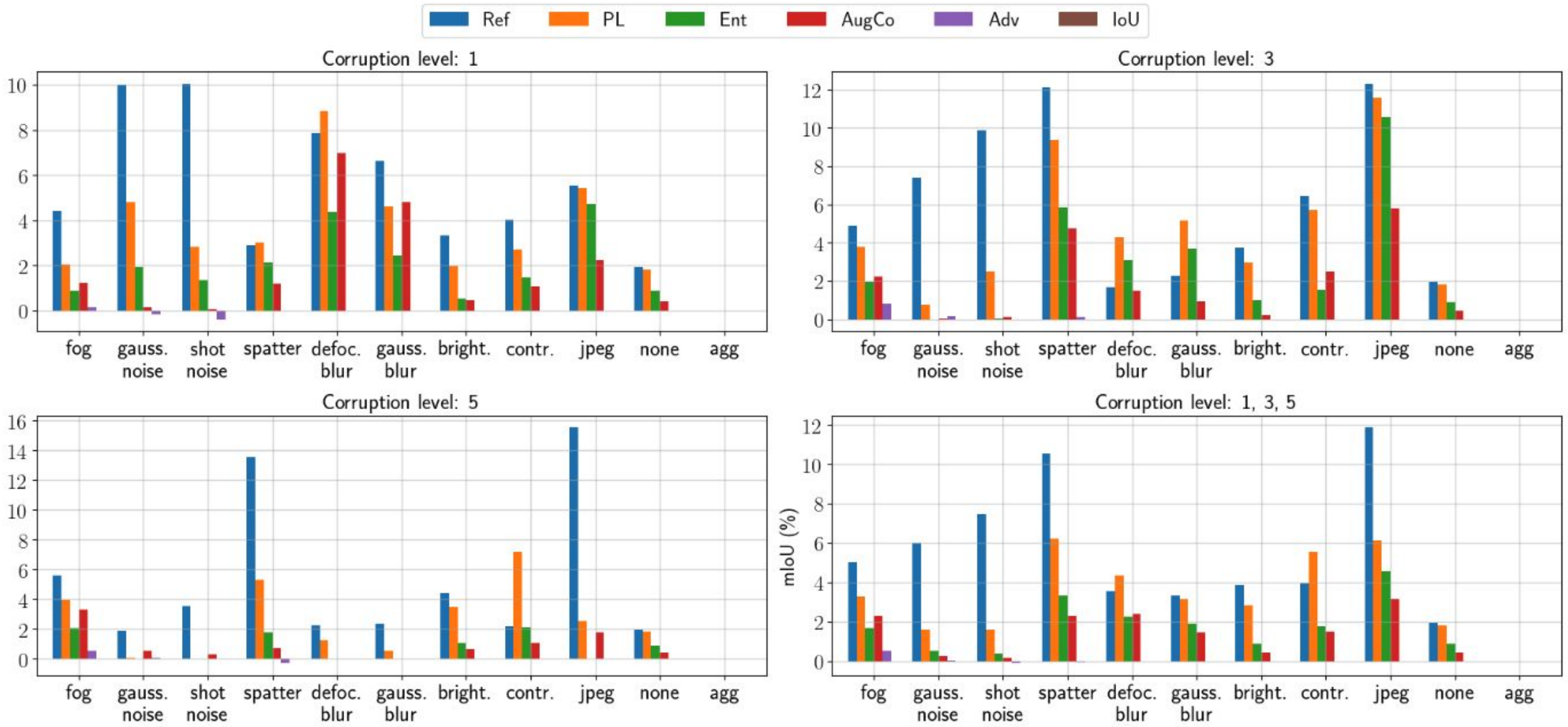
Figure 11. Evolution of masks over iterations of a projected gradient descent adversarial attack on the input image, the target being mask inversion for all of the classes. These masks serve as training data for the refinement module.

Validation: Results and Insights

Overall-optimal hyper-parameters



Per-type-per-severity-optimal hyper-parameters

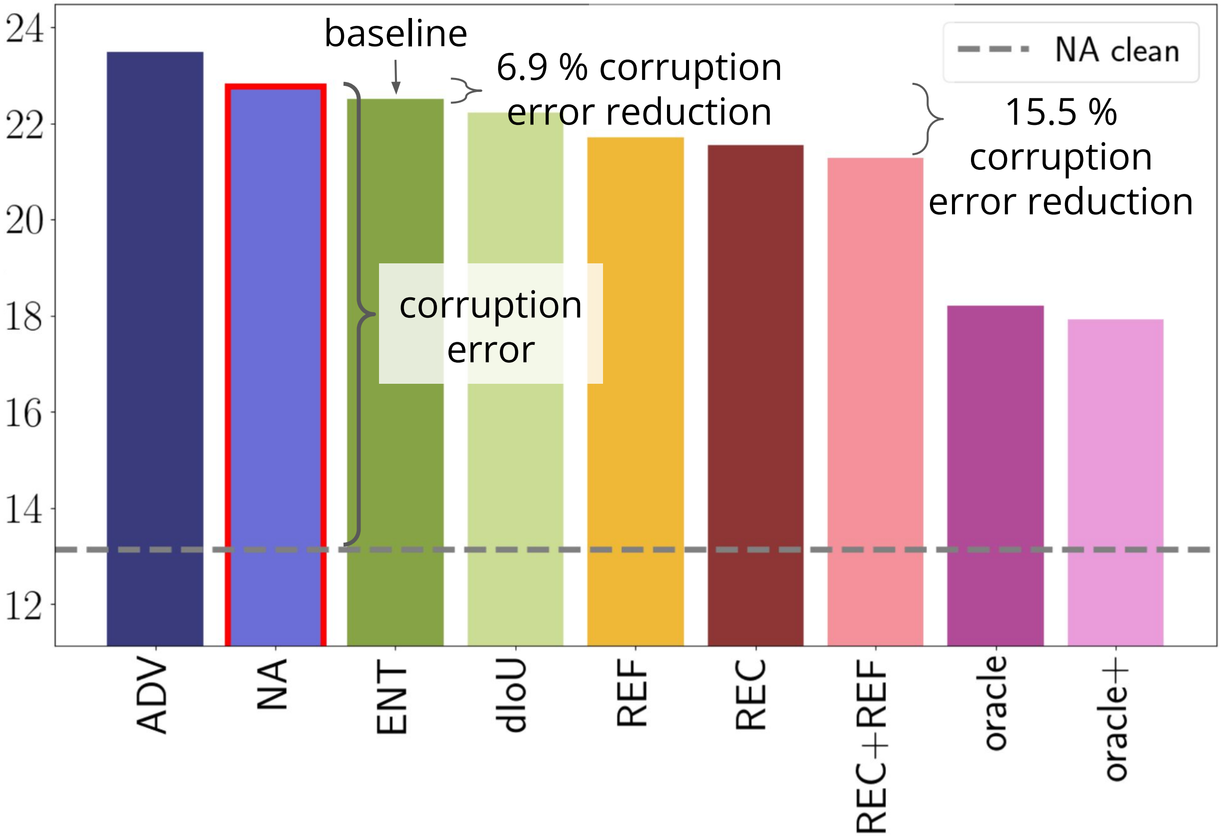


Loss functions matter

All (when applicable) baseline methods improve by using soft IoU loss instead of cross entropy, most likely because of large class imbalance.

	PL				Ref				AugCo			
params	full	full	norm	norm	full	full	norm	norm	full	full	norm	norm
loss	ce	iou	ce	iou	ce	iou	ce	iou	ce	iou	ce	iou
NA	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18
TTA_{α^*}	35.54	<u>37.21</u>	35.60	37.09	35.18	**38.69**	36.88	36.50	35.27	<u>35.66</u>	35.35	35.39
Δ_{ABS}	0.36	2.03	0.42	1.90	$-\epsilon$	3.51	1.70	1.32	0.09	0.48	0.17	0.21

IoU Error of TTA Methods on All Corruptions (~600 images)



Oracle - best method per image is known

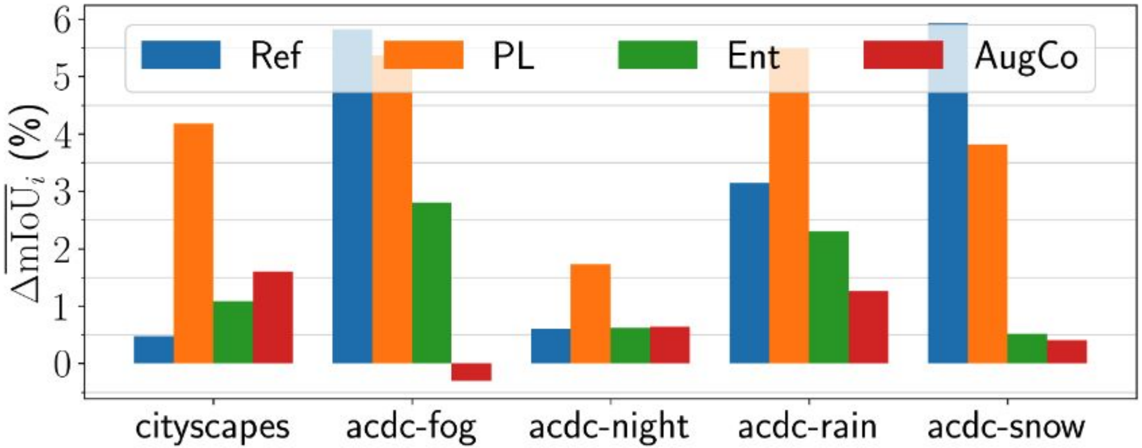
Oracle+ - best method and iteration per image is known

NA - non-adapted results

clean - non-corrupted images

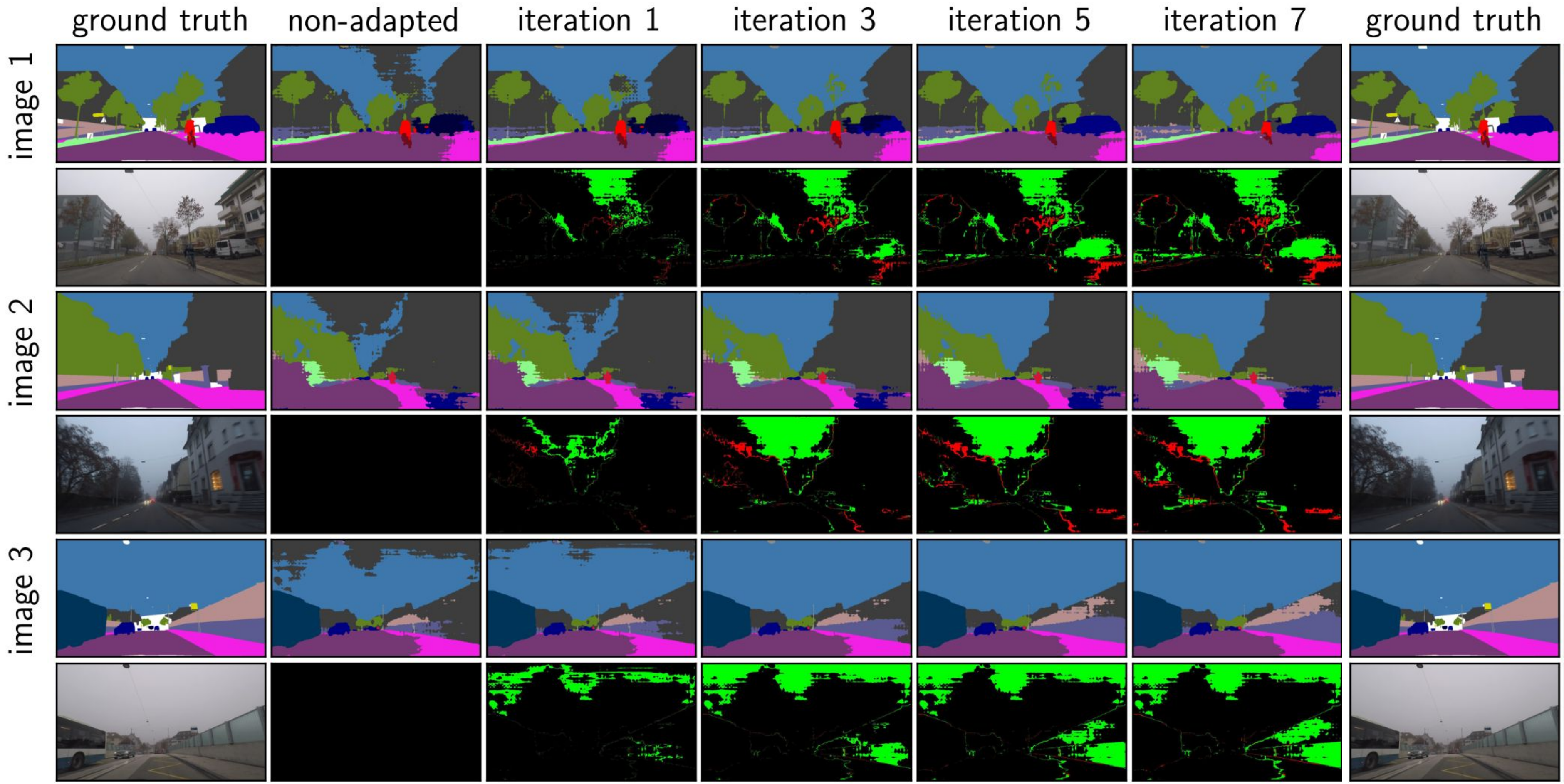
All methods except for oracle+ are evaluated in the last iteration (10) with the best overall learning rate for that method

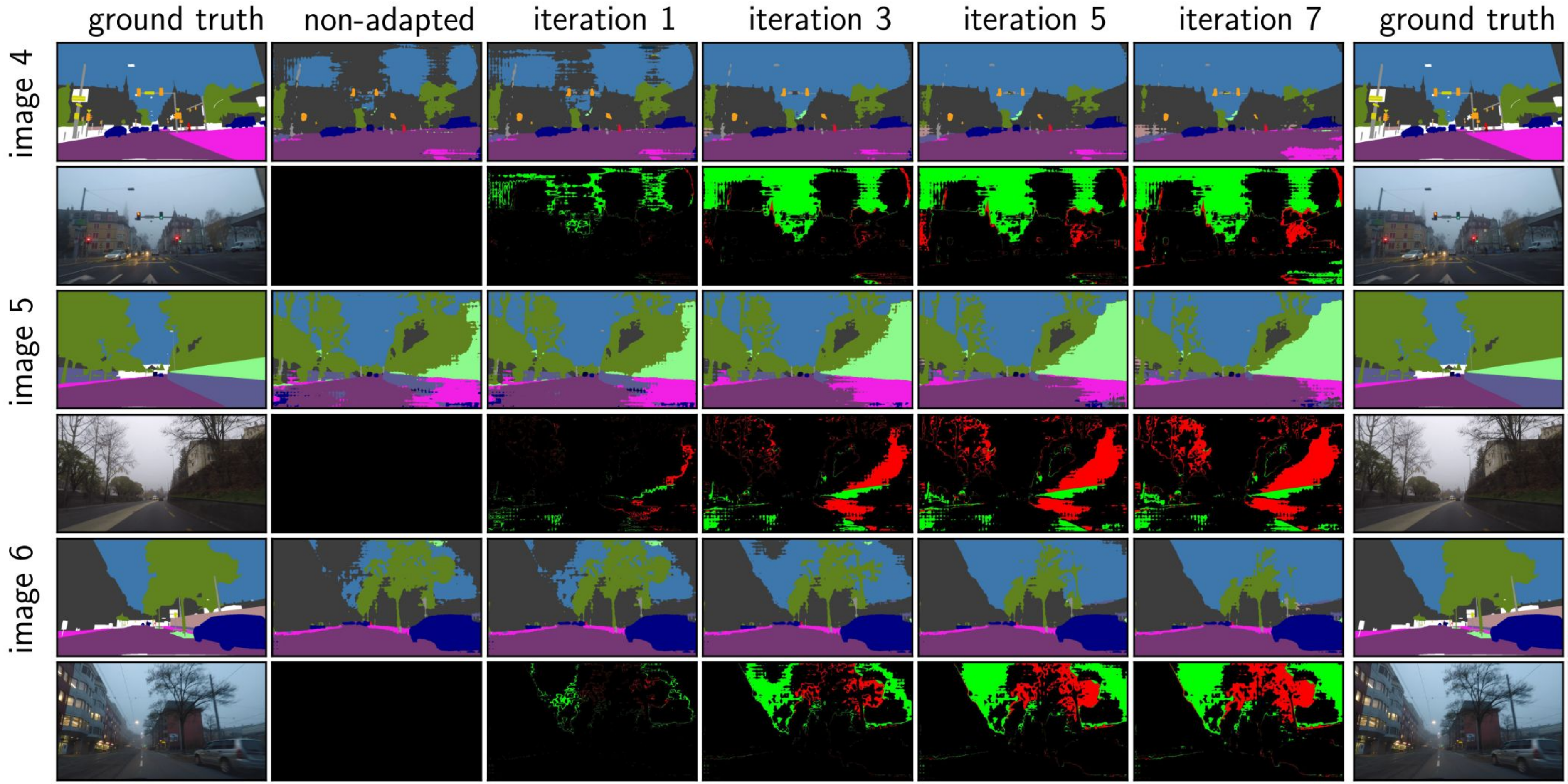
Deployment (test) Results



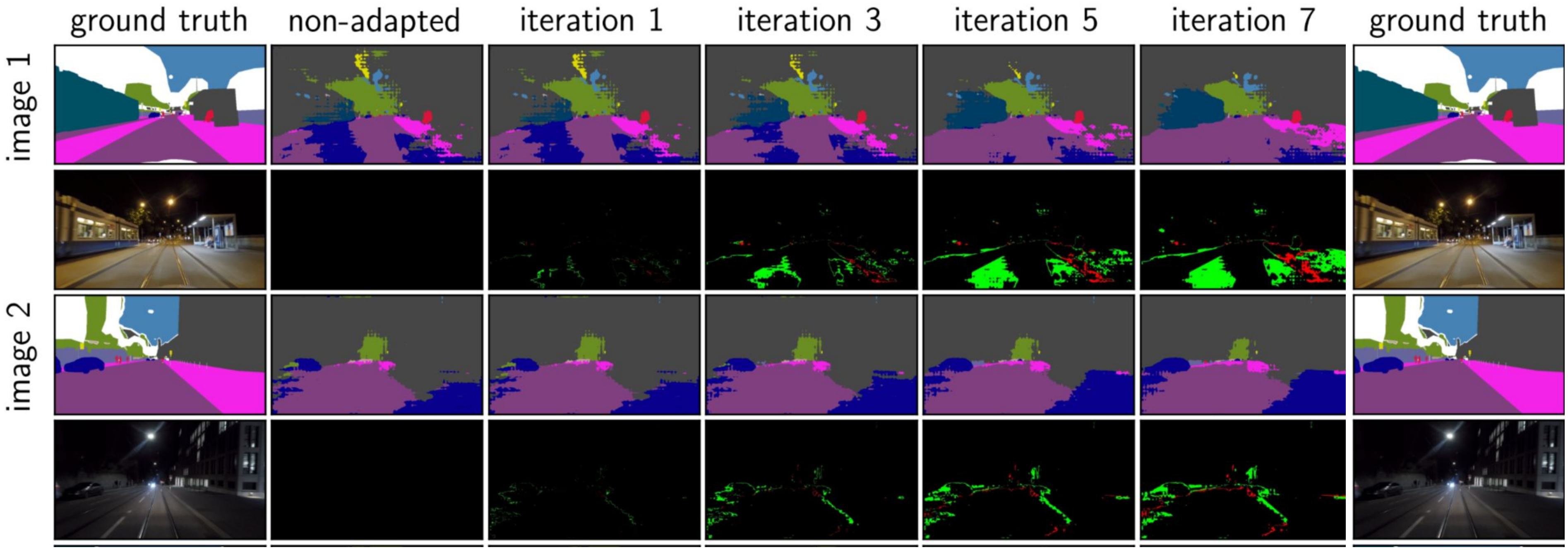
dataset	metric	NA	Ref	PL	Ent	AugCo
		method				
Cityscapes	$\text{m}\overline{\text{IoU}}_i$	34.40	34.71	**37.14**	35.11	35.45
	$\text{m}\overline{\text{IoU}}_c$	28.71	28.64	**30.70**	29.09	29.48
ACDC-fog	$\text{m}\overline{\text{IoU}}_i$	32.03	**35.98**	35.67	33.93	31.82
	$\text{m}\overline{\text{IoU}}_c$	24.87	27.29	**27.52**	26.00	24.69
ACDC-night	$\text{m}\overline{\text{IoU}}_i$	13.60	14.12	**15.09**	14.13	14.15
	$\text{m}\overline{\text{IoU}}_c$	10.77	10.96	**11.53**	10.68	11.01
ACDC-rain	$\text{m}\overline{\text{IoU}}_i$	33.52	35.61	**37.17**	35.05	34.36
	$\text{m}\overline{\text{IoU}}_c$	26.15	27.40	**28.47**	26.89	26.66
ACDC-snow	$\text{m}\overline{\text{IoU}}_i$	31.54	**35.60**	34.15	31.89	31.81
	$\text{m}\overline{\text{IoU}}_c$	25.28	**28.09**	27.17	25.39	25.45

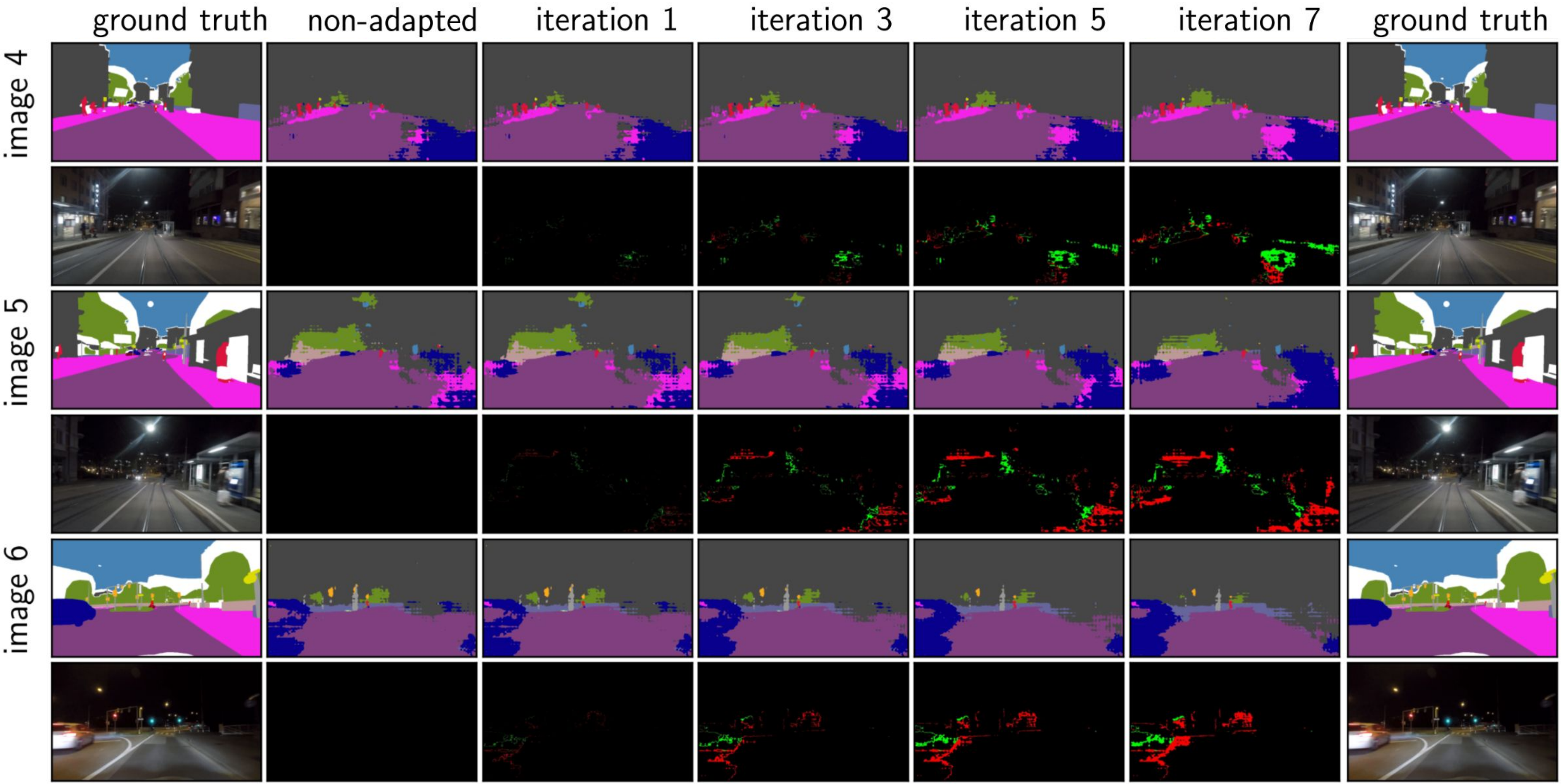
Qualitative Refinement Results


ground truth
non-adapted
iteration 1
iteration 3
iteration 5
iteration 7
ground truth
image 1
image 2
image 3



ground truth
non-adapted
iteration 1
iteration 3
iteration 5
iteration 7
ground truth
image 4
image 5
image 6


ground truth
non-adapted
iteration 1
iteration 3
iteration 5
iteration 7
ground truth
image 1
image 2


ground truth
non-adapted
iteration 1
iteration 3
iteration 5
iteration 7
ground truth
image 4
image 5
image 6

THANK YOU!