NAVER LABS
Europe

Test-time Adaptation:
Formulations, Methods and Benchmarks

Riccardo Volpi

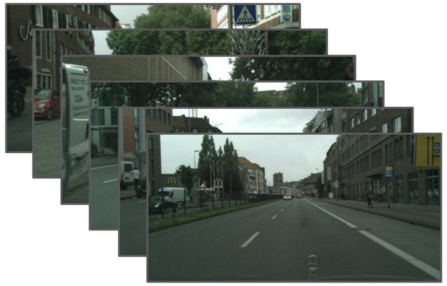
Outline

- Problem formulation
  - From “standard” to “test-time” domain adaptation
- Stationary test-time adaptation
  - Benchmarks and methods
- Continual test-time adaptation
  - Additional challenges
  - Benchmarks and methods
- Conclusions

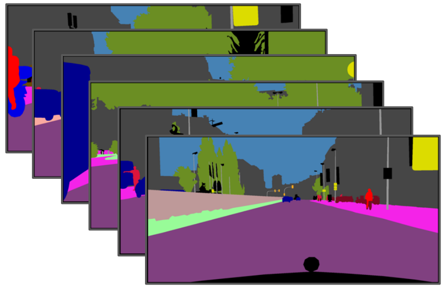
Learning i.i.d.

- **Domain shift**: the image distribution shift wrt train time ($P_X^{train} \neq P_X^{test}$)

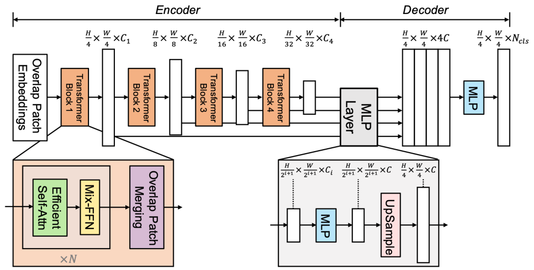
Images



Masks



Model θ_0

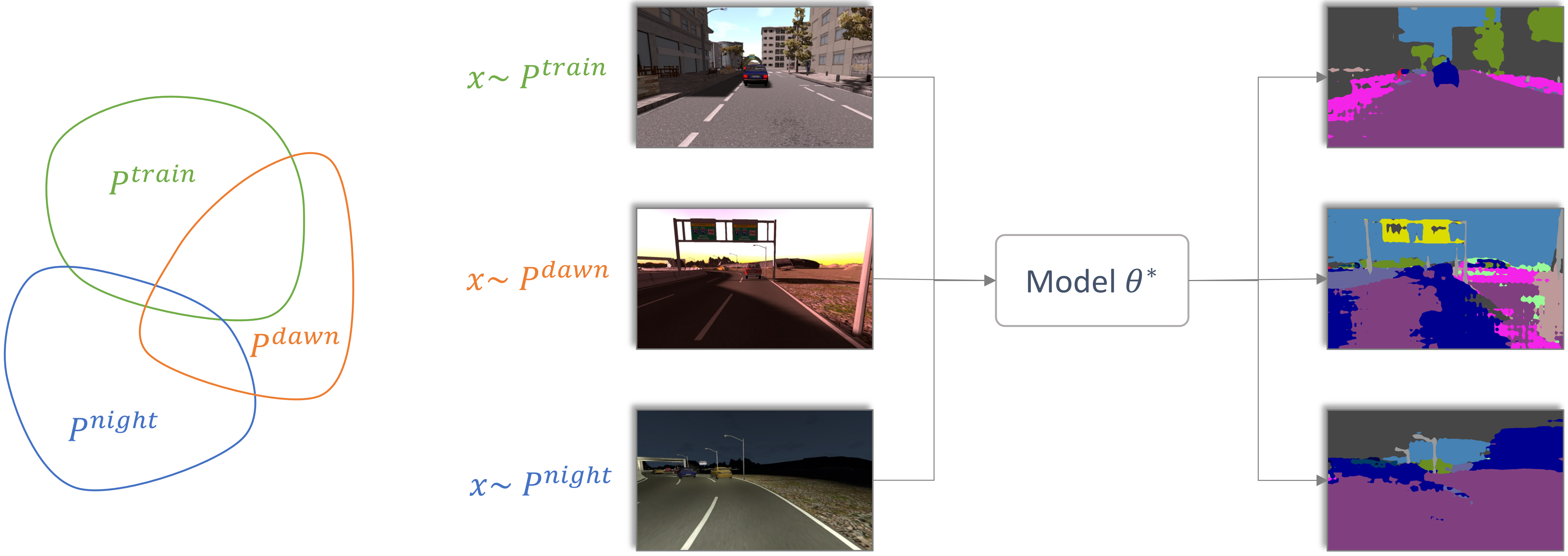


Learning algorithm

Trained model θ^*

Learning i.i.d.

- **Domain shift**: the image distribution shift wrt train time ($P_X^{train} \neq P_X^{test}$)



How to address domain shifts?

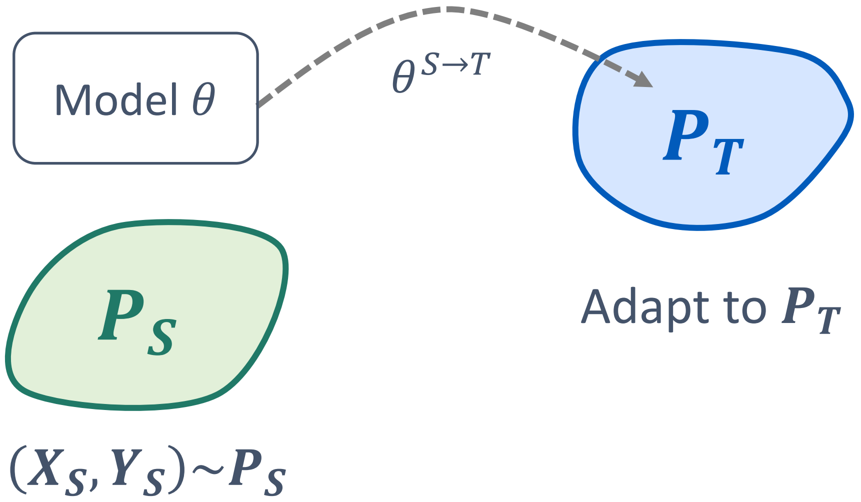
- A very large number of sub-fields
  - Supervised domain adaptation
  - Semi-supervised domain adaptation
  - Unsupervised domain adaptation
  - Domain generalization
  - …

- We focus here on **test-time adaptation**

Problem formulation

Problem formulation

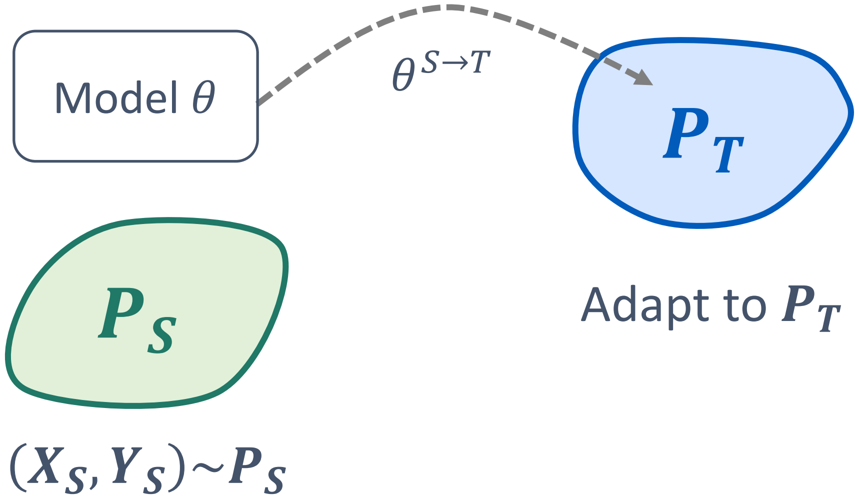
- **"Standard" UDA:** adapt from one or few **source** domains to one or few **target** domains



- Adaptation happens **offline**

Problem formulation

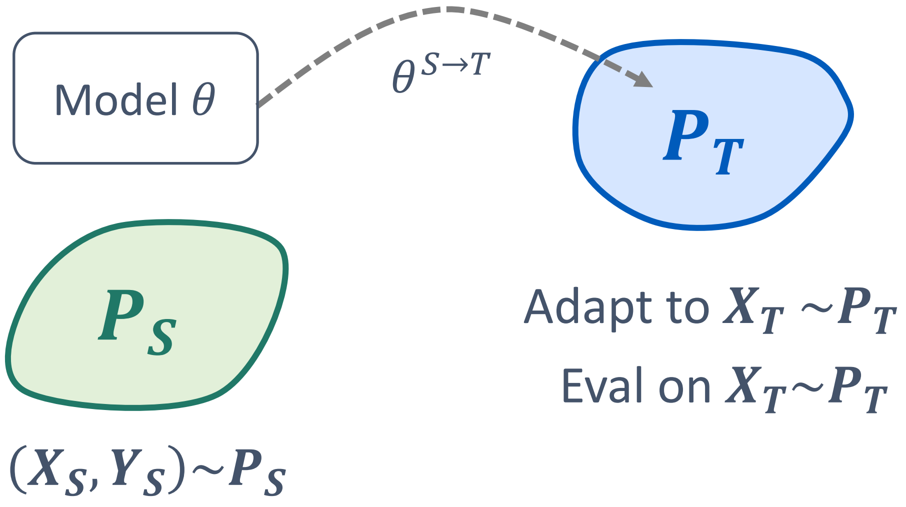
- **“Standard” UDA:** adapt from one or few **source** domains to one or few **target** domains



- Adaptation happens **offline**
- Can be
  - **Transductive** (adapt/test on same data)
  - **Inductive** (adapt/test on different data)

Problem formulation

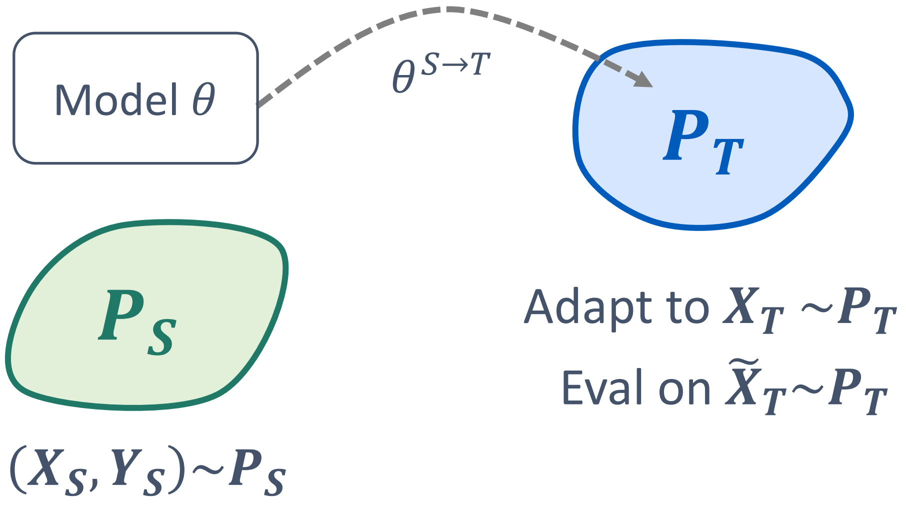
- **“Standard” UDA:** adapt from one or few **source** domains to one or few **target** domains



- Adaptation happens **offline**
- Can be
  - <u>**Transductive**</u> (adapt/test on same data)
  - **Inductive** (adapt/test on different data)

Problem formulation

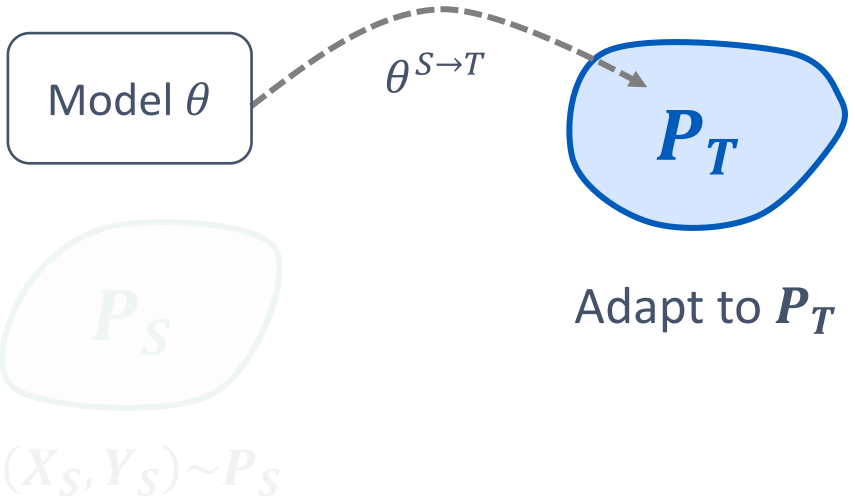
- **"Standard" UDA:** adapt from one or few **source** domains to one or few **target** domains



- Adaptation happens **offline**
- Can be
  - **Transductive** (adapt/test on same data)
  - <u>**Inductive**</u> (adapt/test on different data)

Problem formulation

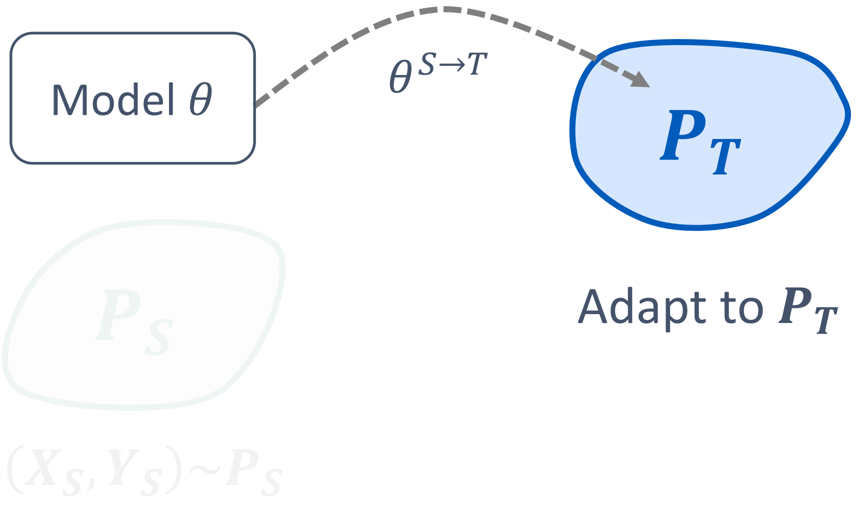
- **“Source-free” UDA:** adapt from one or few **source** domains to one or few **target** domains



- Adaptation happens **offline**
- Can be
  - **Transductive** (adapt/test on same data)
  - **Inductive** (adapt/test on different data)
- <u>No access to the source dataset</u>

Problem formulation

- **“Test-time Adaptation”**

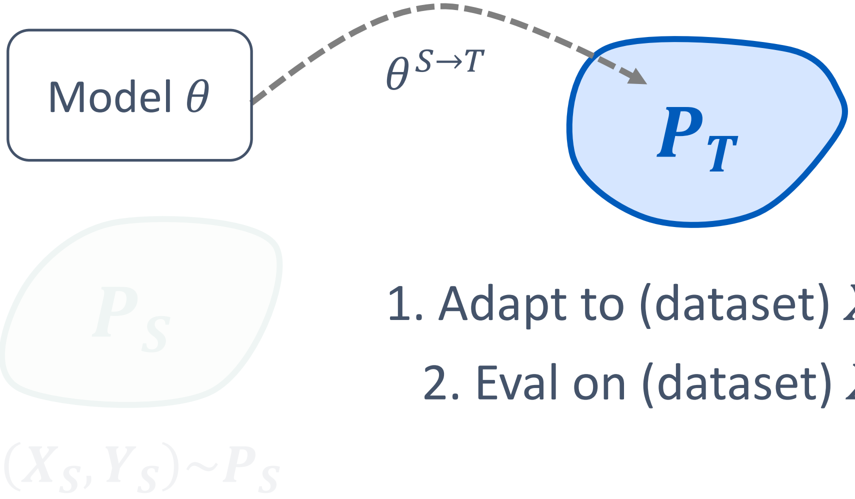


- Adaptation can happen
  - **Offline**
  - **Online**

- No access to the source dataset

Problem formulation

- **“Test-time Adaptation” = “Source-free Adaptation”**



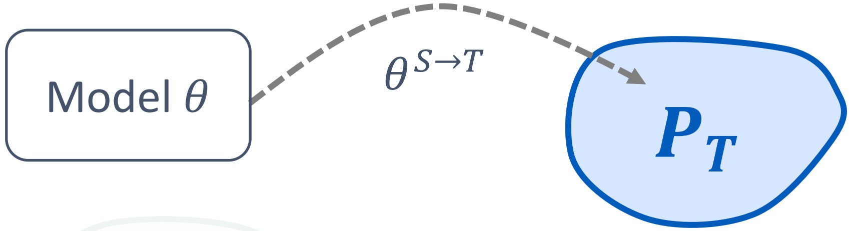
1. Adapt to (dataset) $X_T \sim P_T$
2. Eval on (dataset) $X_T \sim P_T$

- Adaptation can happen
  - **Offline**
  - **Online**
- No access to the source dataset

Wang et al., “**Tent: Fully Test-time Adaptation by Entropy Minimization**” ICLR 2021
Chidlovskii et al., **“Domain Adaptaiton in the Absence of Source Domain Data”, SIGKDD 2016**

Problem formulation

- **“Test-time Adaptation”**



1. $i = 0$

2. Adapt to (batch) $\boldsymbol{X_T^i} \sim \boldsymbol{P_T}$

3. Eval on (batch) $\boldsymbol{X_T^i} \sim \boldsymbol{P_T}$

4. i++, back to 2.

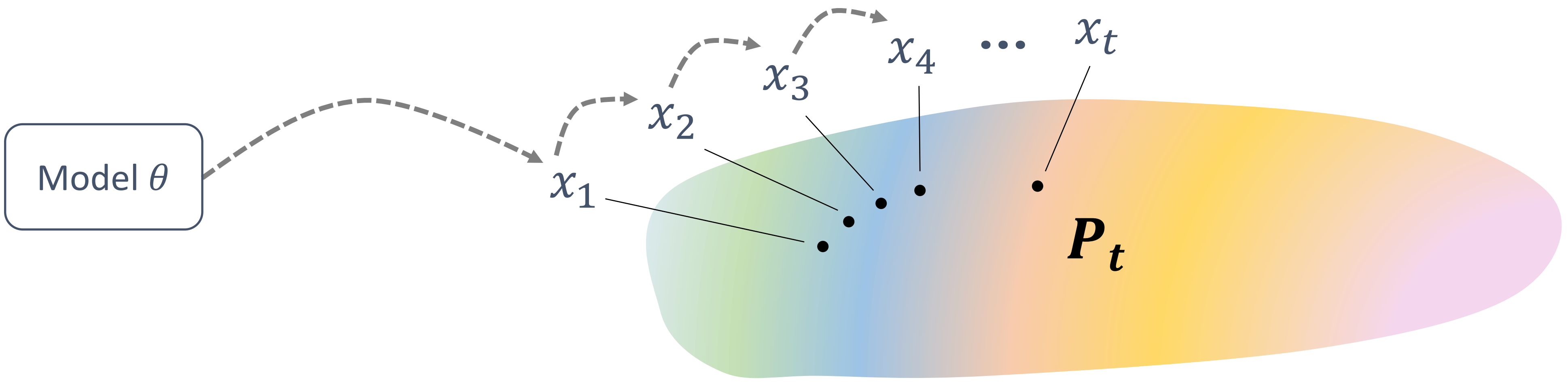
- Adaptation can happen
  - **Offline**
  - **Online**

We can also relax this assumption

- No access to the source dataset

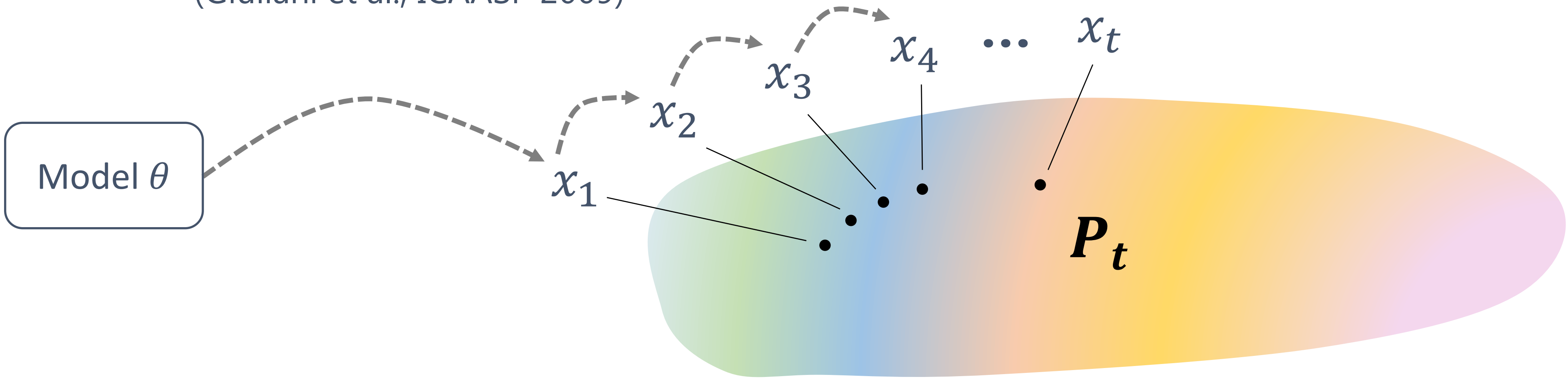
Problem formulation

- **"Continual TTA": frame-by-frame** adaptation with **continuous shifts**
  - Samples are drawn from an **ever-changing distribution** $\longrightarrow$ $(\boldsymbol{x_t})_{\boldsymbol{0}}^{\infty} \sim \boldsymbol{P_t}$
  - Each sample/batch $\boldsymbol{X_t}$ represents an **adaptation problem in itself**



Problem formulation

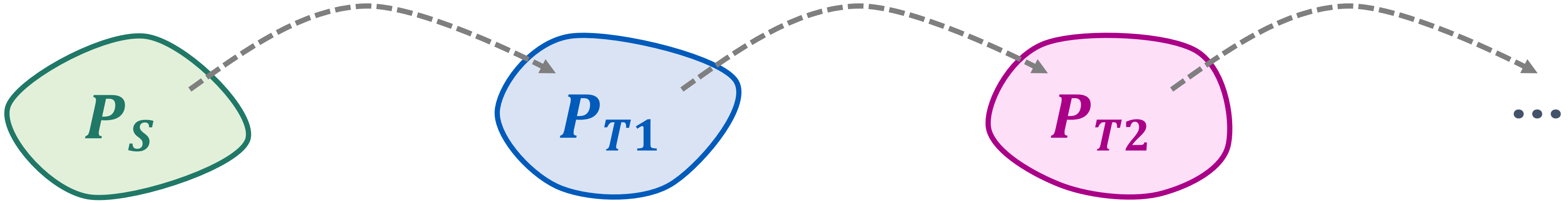
- **"Continual TTA": frame-by-frame** adaptation with **continuous shifts**
  - Seminal works in this setting are from the NLP literature
    - (Dredzer and Crammer, EMNLP 2009)
    - (Giuliani et al., ICAASP 2009)



Dredzer and Crammer, "**Online Methods for Multi-domain Learning and Adaptation**", EMNLP 2009
Giuliani et al., "**On-line speaker adaptation on telephony speech data with adaptively trained acoustic models**", ICAASP 2009

(Related) Problem formulations

- **Incremental UDA:** offline adaption to **sequential target domains** at different stages



$(X_S, Y_S) \sim P_S$ $X_{T1} \sim P_{T1}$ $X_{T2} \sim P_{T2}$

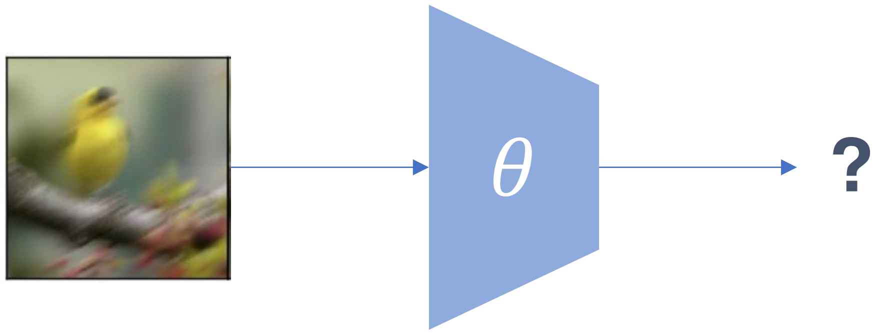
(Related) Problem formulations

- **Domain generalization:** there is no adaptation at all, we train on one (or more) domains and test on different ones

Methods

Methods

- Overall goal: adapting a given model to new batches of data
  - Extreme case: single-sample adaptation

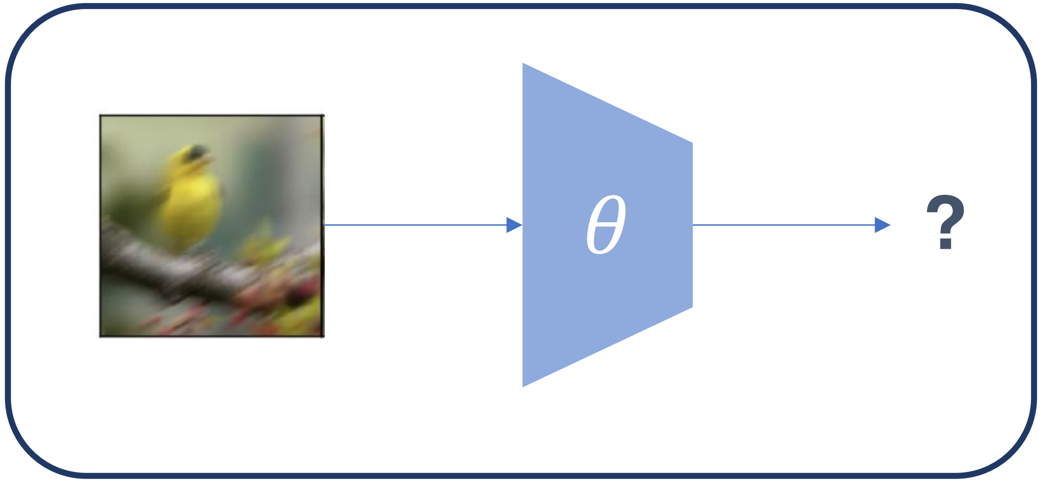


- Update model weights in order to maximize our chances of correct prediction
- We're not given any label

Image from Hendrycks and Dietterich **"Benchmarking Neural Network Robustness to Common Corruptions and Perturbations"**, ICLR 2019

Methods

- Self-training with pseudo-labels
- BatchNorm statistics adaptation
- BatchNorm parameters adaptation
- Self-supervised training
- Data augmentation



Methods

- **Self-training with pseudo-labels**

- **Standard recipe**
  - Trust (some of) your model's predictions
  - Use them as ground truth to update your model
  - Repeat

- Originally for **semi-supervised learning**
  - Large application in DA
  - Standard **baseline in TTA**

Methods

- **BatchNorm statistics adaptation**

- In BN layers we generally use the statistics from the training set

- We can update them with the target's
  - **Online** [Mancini et al. 2018]
  - **Offline** [Schneider et al. 2020]

- Often important not to completely replace the training ones (weighted)

$$\widehat{F^l(x_i^t)} = \gamma \cdot \frac{F^l(x_i^t) - \mu_l}{\sigma_l^2} + \beta$$

$$\mu_l := (1-\alpha) \cdot \mu_l + \alpha \cdot \mathbb{E}\{F^l(x_i^t)\}$$
$$\sigma_l^2 := (1-\alpha) \cdot \sigma^2 + \alpha \cdot \mathbb{E}\Big\{\big(F^l(x_i^t) - \mathbb{E}\{F^l(x_i^t)\}\big)^2\Big\}$$

Methods

- **(Batch)Norm parameters adaptation**

- **Entropy minimization** is another standard technique from semi-supervised learning
- But updating all network parameters cause **huge drifts from the original model**
- We can just **update the BatchNorm parameters** (or LayerNorm, etc.) via entropy minimization
- At the same time, we can update statistics

$$\operatorname*{argmin}_{\beta,\gamma} \mathcal{L}_H := -\sum_{p \in x_i^t} \sum_{c}^{C} \hat{y}_{i,c}^{p} \log \hat{y}_{i,c}^{p}$$

Wang et al., "**Tent: Fully Test-time Adaptation by Entropy Minimization**" ICLR 2021

Methods

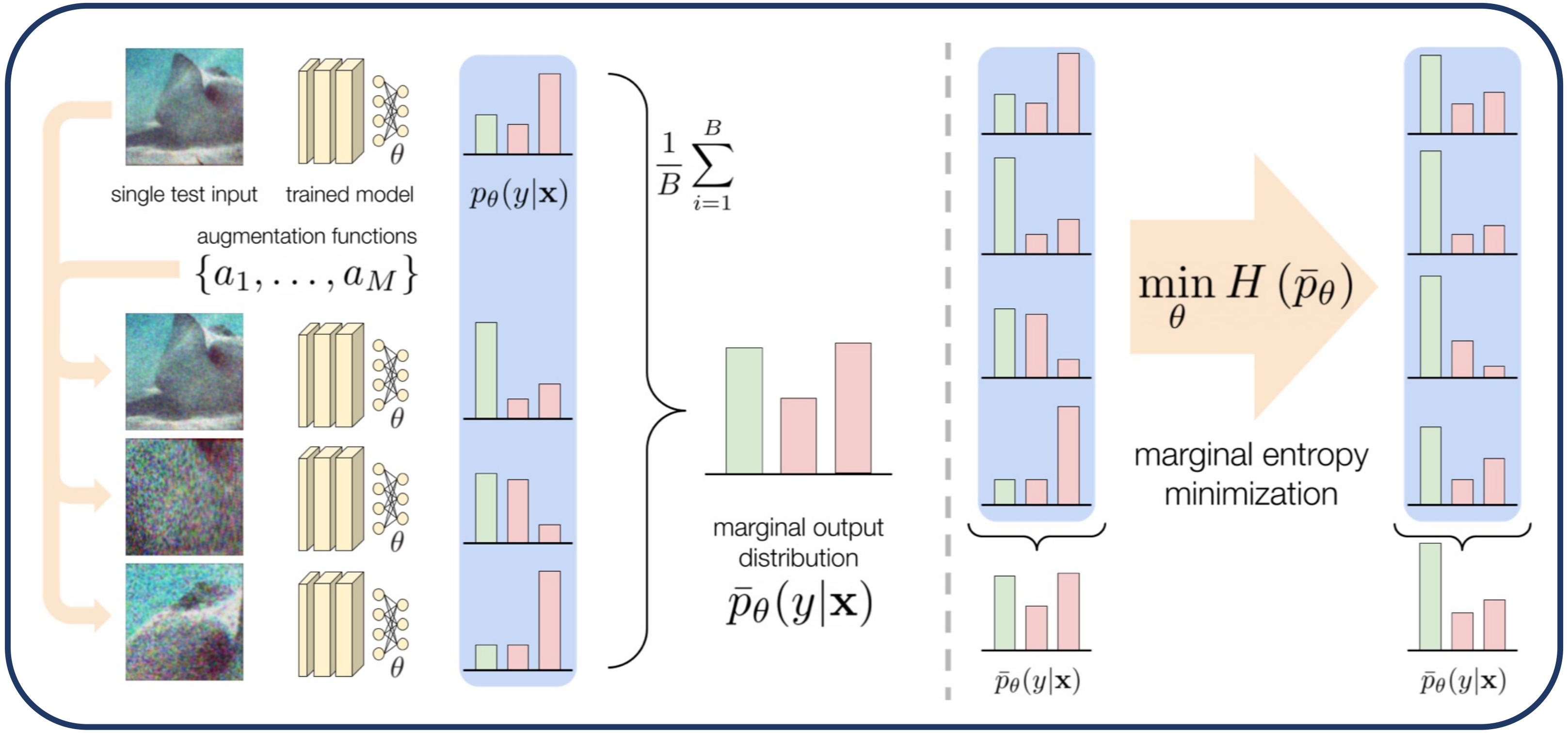
- **Self-supervised learning**

- We can solve a SSL objective using the test data
- Given a test-sample or a batch, we solve a SSL problem before making a prediction
- Note: SSL pre-training itself helps robustness
  - See Hendrycks et al., "**Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty**", NeurIPS 2019

Methods

- **Data augmentation**

- We can generate several copies of the current batch and use some of the previously mentioned objectives (e.g. entropy minimization)



Benchmarks

- In general, train on one dataset and adapt to another one

- Researchers have mostly played with
  - ImageNet to ImageNet-C/A/R
  - CIFAR10 to CIFAR10-C
  - CIFAR100 to CIFAR100-C

- The only constraint, is that the set of classes need to be the same
  - TTA does not fit class-incremental purposes
  - We *could* have new classes, but we would be helpless

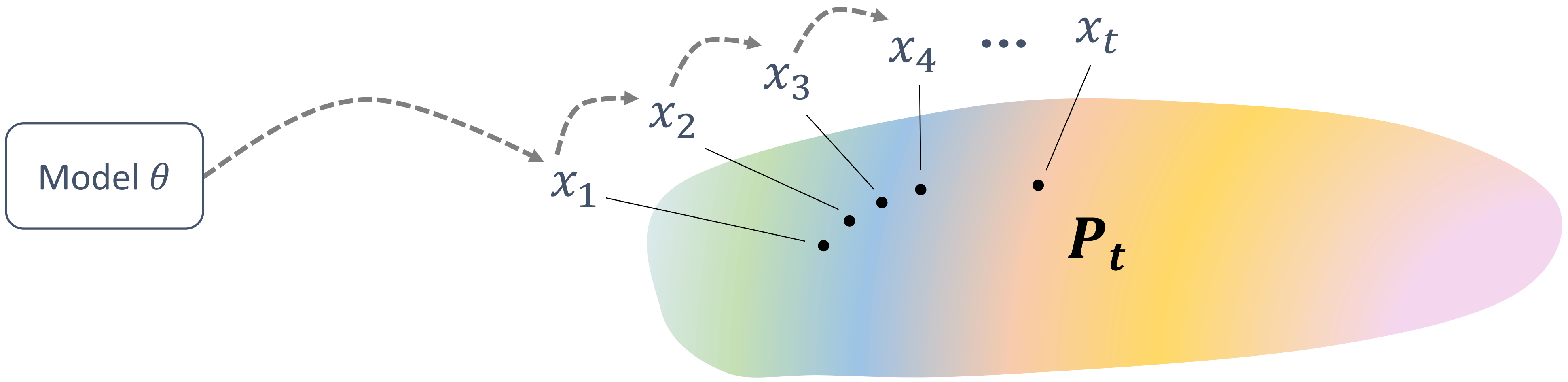
Benchmarks

	ImageNet-C mCE ↓	ImageNet-R Error (%)	ImageNet-A Error (%)
Baseline ResNet-50 [11]	76.7	63.9	100.0
+ TTA	77.9 (+1.2)	61.3 (−2.6)	98.4 (−1.6)
+ Single point BN	71.4 (−5.3)	61.1 (−2.8)	99.4 (−0.6)
+ MEMO (ours)	69.9 (−6.8)	58.8 (−5.1)	99.1 (−0.9)
+ BN ($N = 256, n = 256$)	61.6 (−15.1)	59.7 (−4.2)	99.8 (−0.2)
+ Tent (online) [46]	54.4 (−22.3)	57.7 (−6.2)	99.8 (−0.2)
+ Tent (episodic)	64.7 (−12.0)	61.0 (−2.9)	99.7 (−0.3)

From Zhang et al., "**NEMO: Test Time Robustness via Adaptation and Augmentation**" NeurIPS 2022

Continual TTA

- Addressing TTA in a continually evolving environment



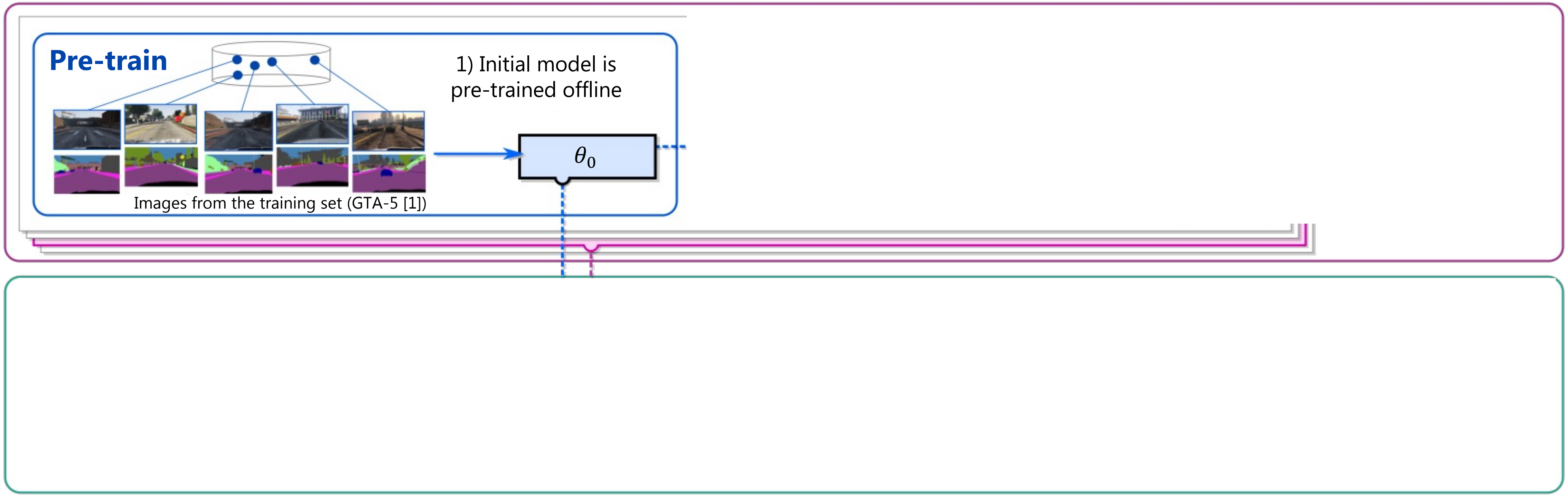
- Additional challenge: **catastrophic forgetting**

The OASIS benchmark

- (2022) Lack of benchmarks to assess segmentation models in these setting
- We introduced one
  - **Image-by-image adaptation** in sequences of **temporally correlated** frames
  - Fair and realistic **pre-train/validate/deploy** pipeline
  - Need to overcome **catastrophic forgetting**

The OASIS benchmark

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself



Methods

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

- Baselines:
  - **Self-training with pseudo-labels**
  - **BN statistics adaptation**
  - **BN parameters adaptation**
  - **Self-supervised training**

Methods

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

- Baselines:
  - <u>**Self-training with pseudo-labels**</u>
  - **BN statistics adaptation**
  - **BN parameters adaptation**
  - **Self-supervised training**

1. Trust (some of) your model's predictions
2. Use them as ground truth to update your model
3. Repeat

Methods

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

- Baselines:
  - **Self-training with pseudo-labels**
  - **BN statistics adaptation**
  - **BN parameters adaptation**
  - **Self-supervised training**

$$\widehat{F^l(x_i^t)} = \gamma \cdot \frac{F^l(x_i^t) - \mu_l}{\sigma_l^2} + \beta$$

$$\mu_l := (1-\alpha) \cdot \mu_l + \alpha \cdot \mathbb{E}\{F^l(x_i^t)\}$$
$$\sigma_l^2 := (1-\alpha) \cdot \sigma^2 + \alpha \cdot \mathbb{E}\big\{(F^l(x_i^t) - \mathbb{E}\{F^l(x_i^t)\})^2\big\}$$

Mancini et al., "**Kitting in the Wild through Online Domain Adaptation**" IROS 2018
Schneider et al., "**Improving robustness against common corruptions by covariate shift adaptation**", NeurIPS 2020

Methods

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

- Baselines:
  - **Self-training with pseudo-labels**
  - **BN statistics adaptation**
  - **<u>BN parameters adaptation</u>**
  - **Self-supervised training**

BN statistics adaptation

+

$$\operatorname*{argmin}_{\beta,\gamma} \mathcal{L}_H := -\sum_{p \in x_i^t} \sum_{c}^{C} \hat{y}_{i,c}^{p} \log \hat{y}_{i,c}^{p}$$

Methods

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

- Baselines:
  - **Self-training with pseudo-labels**
  - **BN statistics adaptation**
  - **BN parameters adaptation**
  - **Self-supervised training**

Solve a side SSL objective on the target samples

Catastrophic forgetting

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

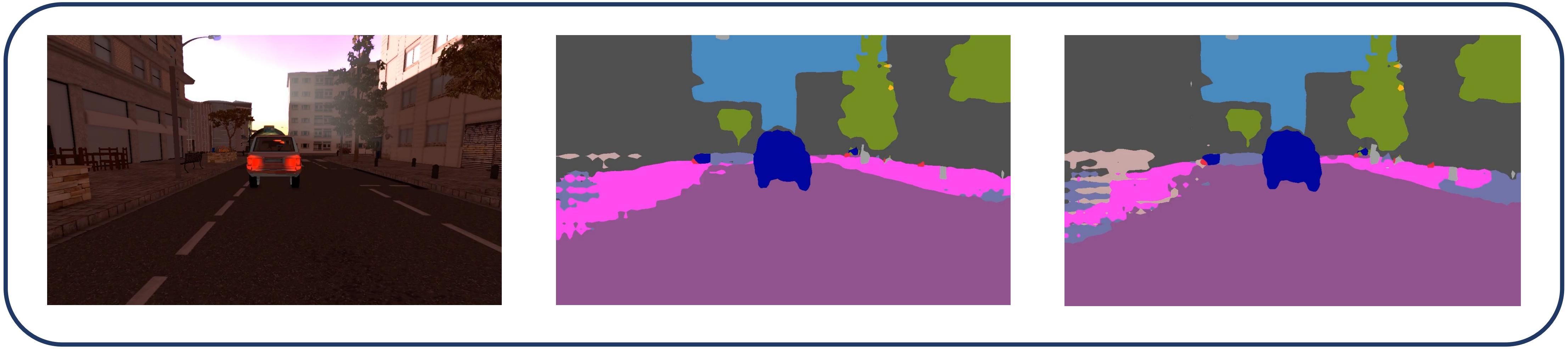
- **Main problem**: like often in continual learning, **catastrophic forgetting**
- We're learning in an unsupervised way, so it's not trivial how to avoid the model to forget classes.
- Classes that are more rare will disappear, leaving their space to the more abundant ones

- **Example:** in urban street segmentation, it's easy to forget about ***things*** (countable objects), overtaken by the more abundant ***stuff*** (street, sky, buildings, etc.)

Catastrophic forgetting

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

- **Main problem**: like often in continual learning, **catastrophic forgetting**



Catastrophic forgetting

- The goal is adapting **frame-by-frame** to streams of **temporally correlated, unlabeled samples**
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself

- **Some solutions:**
  - **"Naive" learning:** instead of doing continual learning, at each frame re-start from the original model
  - **Memories:** keep rehearsing the original (labelled) training samples to the model
  - **Reset strategies:** use the original model as a checkpoint, and reset when some thershold is met

Results

- **Evaluation**

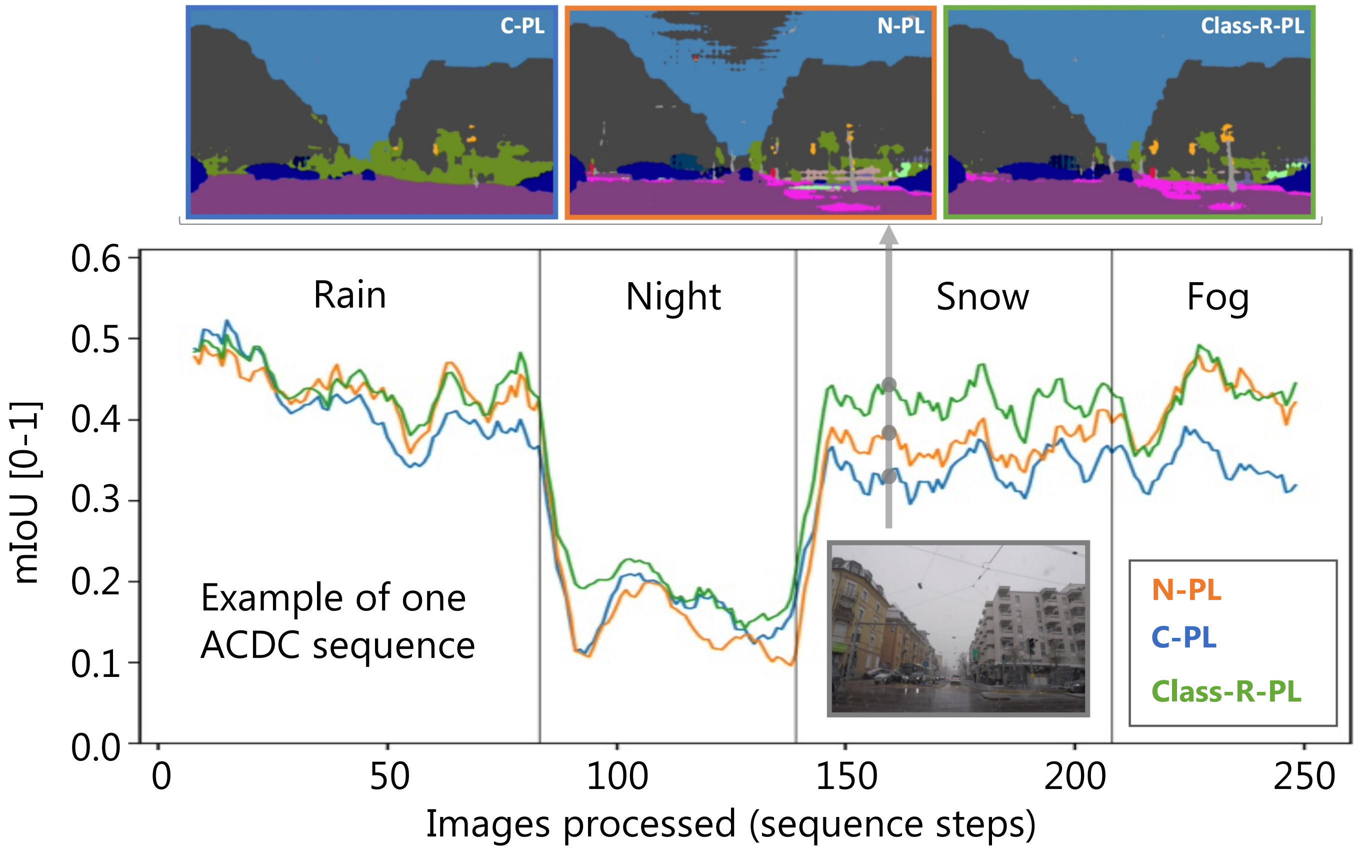
1. Compute mIoU for each frame
2. Average across each sequence
3. Average across dataset

- **Effect of pre-training** (no adaptation)

Training	SYNTHIA	ACDC	Cityscapes A.W.	Cityscapes O.
ERM	35.9 ± 2.5	29.5 ± 2.5	35.6 ± 1.9	40.3 ± 0.9
DR↑	34.3 ± 3.3	29.5 ± 2.4	36.2 ± 2.3	41.2 ± 1.0
DR↑↑	$\mathbf{39.8 \pm 3.0}$	$\mathbf{33.6 \pm 2.5}$	$\mathbf{38.3 \pm 2.6}$	$\mathbf{45.2 \pm 1.0}$
DR↑↑↑	31.9 ± 3.0	26.7 ± 2.3	33.2 ± 2.5	37.7 ± 1.1

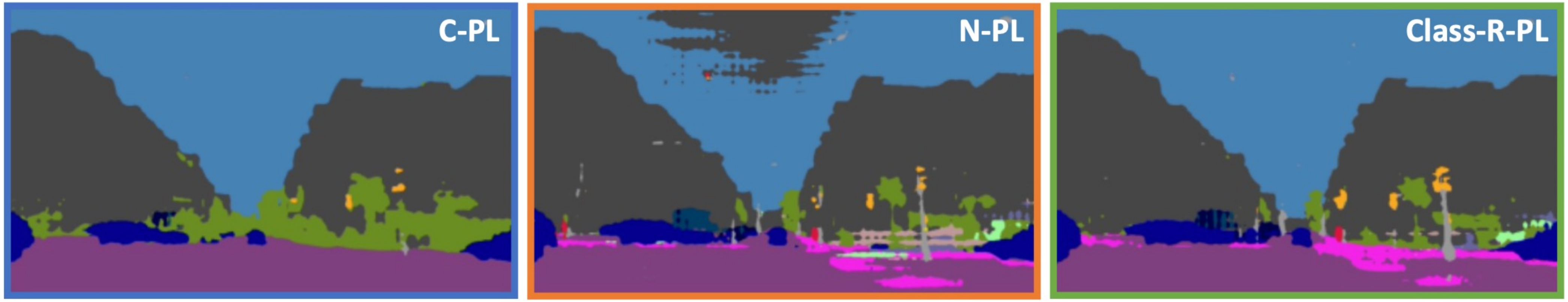
A.W. = Artificial Weather O. = Original

Results

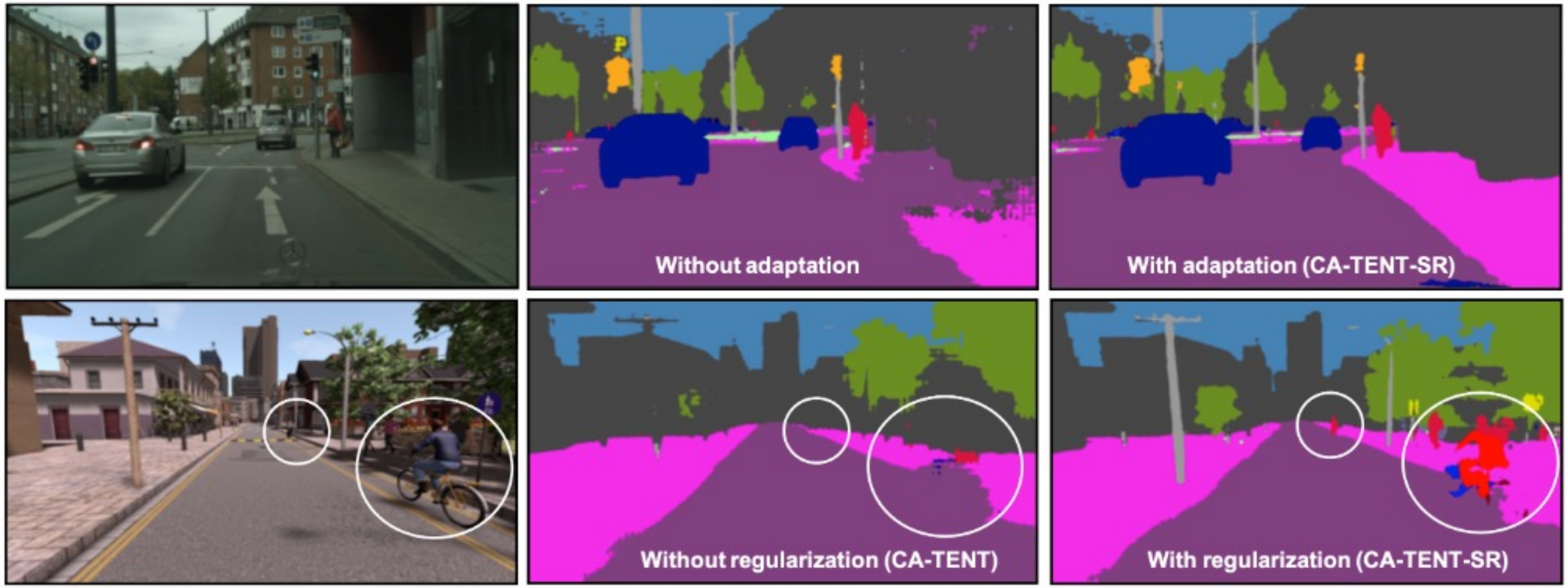


C-PL
N-PL
Class-R-PL
Rain
Night
Snow
Fog
0.6
0.5
0.4
0.3
0.2
0.1
0.0
mIoU [0-1]
0
50
100
150
200
250
Images processed (sequence steps)
Example of one
ACDC sequence
N-PL
C-PL
Class-R-PL

Results



Results

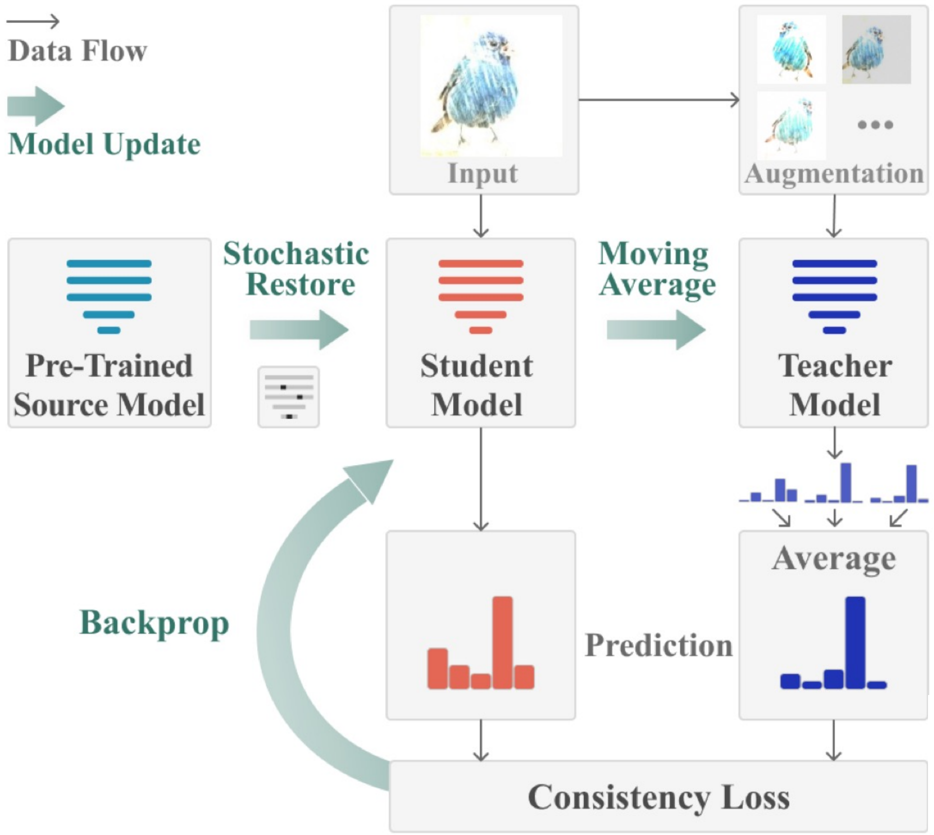

Without adaptation
With adaptation (CA-TENT-SR)
Without regularization (CA-TENT)
With regularization (CA-TENT-SR)

Results

		Validation	Test (Deploy)				
		SYNTHIA	ACDC	Cityscapes A.W.	Cityscapes O.		
	No adapt. baseline (NA)	39.8 ±3.0	33.6 ±2.5	38.3 ±2.6	45.2 ±1.0		
	Method	Improvements				**Add. computation**	**Add. memory**
Style trans.	N-ST (random)	+0.7% ±1.7	−7.4% ±2.6	+4.1% ±1.7	+0.4% ±0.8	ST optim. (++)	Source set (++)
Style trans.	N-ST (NN)					ST optim. & NN (+++)	Source set (++)
Naive adapt.	N-BN					BN stat. update (*)	-
Naive adapt.	N-PL					$\mathcal{O}(\texttt{trainsteps})$ (+)	-
Naive adapt.	N-TENT					$\mathcal{O}(\texttt{trainsteps})$ (+)	-
CL Vanilla	C-BN					BN stat. update (*)	-
CL Vanilla	C-PL					$\mathcal{O}(\texttt{trainsteps})$ (+)	-
CL Vanilla	C-TENT					$\mathcal{O}(\texttt{trainsteps})$ (+)	-
CL SrcReg	C-PL-SR					$\mathcal{O}(\texttt{trainsteps})$ (+)	Source set (++)
CL SrcReg	C-TENT-SR					$\mathcal{O}(\texttt{trainsteps})$ (+)	Source set (++)
CL Reset	Class-R-PL					$\mathcal{O}(\texttt{trainsteps})$ (+)	Backup net (+)
CL Reset	Class-R-TENT					$\mathcal{O}(\texttt{trainsteps})$ (+)	Backup net (+)
CL Oracle	Oracle-R-PL	+10.8% ±4.5	+11.6% ±3.8	+12.7% ±5.6	+2.9% ±1.4	$\mathcal{O}(\texttt{trainsteps})$ (+)	Backup net (+)
CL Oracle	Oracle-R-TENT	+11.4% ±4.4	+10.9% ±4.1	+12.2% ±5.9	+1.9% ±1.4	$\mathcal{O}(\texttt{trainsteps})$ (+)	Backup net (+)

(Image from "The Bear", FX)

More cont. TTA methods and benchmarks



- **CoTTA**
  - Pseudo-labeling
  - Augmentations
  - Random weight reset
- **Benchmarks**
  - CIFAR10 to CIFAR10-C
  - CIFAR100 to CIFAR100-C
  - **ImageNet to ImageNet-C**
  - Cityscapes to ACDC

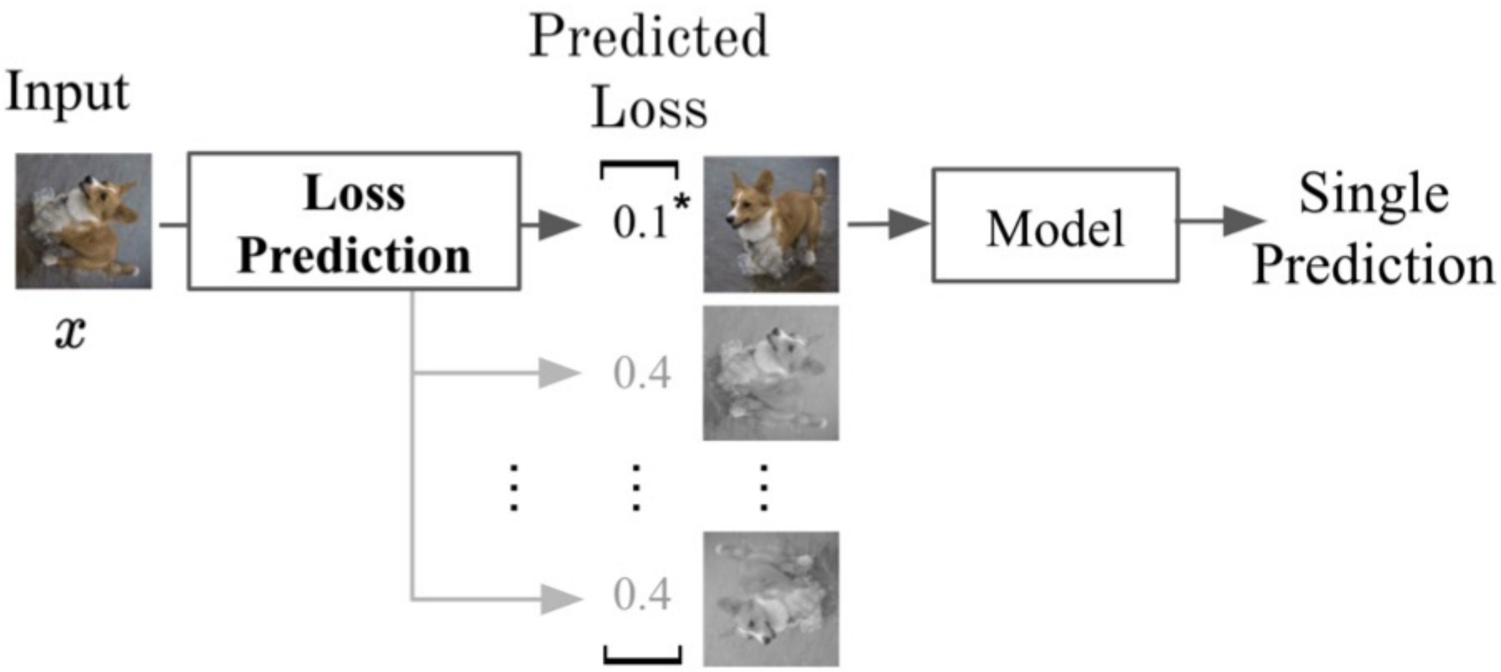
Avg. Error (%)	Source	BN Adapt	Test Aug [5]	TENT [58]	CoTTA
ImageNet-C	82.4	72.1	71.4	66.5	63.0 $\pm$ 1.8 (0.1)

Continual TTA in related areas

- We focused on **2D tasks** here… but there's more
- Online adaptation for **kitting**
  - Mancini et al., Kitting in the Wild through Online Domain Adaptation, IROS 2018
- Online adaptation for **depth estimation**
  - Tonioni et al., Learning to Adapt for Stereo, CVPR 2019
  - Tonioni et al., Real-time Self-Adaptive Deep Stereo, CVPR 2019
- Continual TTA for **3D lidar segmentation** tasks
  - Saltori et al., GIPSO: Geometrically Informed Propagation for Online Adaptation in 3D LiDAR Segmentation, ECCV 2022

Test-time augmentations

- (Active) test-time augmentation can be framed as test-time adaptation



Kim et al., "**Learning Loss for Test-Time Augmentation**" NeurIPS 2020

Conclusions

- Test-time adaptation is a recent and active research area

- Yet, its roots are from well established fields
  - Domain adaptation
  - Online learning
  - Self-training

- Its continual counterpart introduces additional challenges
  - Catastrophic forgetting
  - Evaluating in a ever-changing environments

References

No representation learning

- (NLP) Dredze and Crammer, Online Methods for Multi-Domain Learning and Adaptation, EMNLP 2008
- (NLP) Giuliani et al., On-line speaker adaptation on telephony speech data with adaptively trained acoustic models, ICASSP 2009
- (supervised) Zao and Hoi, OTL: A Framework of Online Transfer Learning, ICML 2010
- Hoffman et al., Continuous Manifold Based Adaptation For Evolving Visual Domains, CVPR 2014
- (supervised) Xu et al., Incremental Domain Adaptation of Deformable Part-based Models, BMVC 2014
- Lampert, Predicting the Future Behavior of a Time-Varying Probability Distribution, CVPR 2015
- Soleymani et al., Incremental Evolving Domain Adaptation, IEEE Transactions on Knowledge and Data Engineering 2016
- Li et al., Domain Generalization and Adaptation Using Low Rank Exemplar SVMs, TPAMI 2018
- Moon et al., Multi-step Online Unsupervised Domain Adaptation, ICASSP 2020

Deep learning-based

- Mancini et al., Kitting in the Wild through Online Domain Adaptation, IROS 2018
- Zhang et al., Online Adaptation through Meta-Learning for Stereo Depth Estimation, arXiv 2019
- Ashukha et al., Pitfalls of in-Domain Uncertainty Estimation and Ensembling in Deep Learning, ICLR 2020
- Sun et al., Test-Time Training with Self-Supervision for Generalization under Distribution Shifts, ICML 2020
- Schneider et al., Improving robustness against common corruptions by covariate shift adaptation, NeurIPS 2020
- Wang et al., Tent: Fully Test-time Adaptation by Entropy Minimization, ICLR 2021
- Ikasawa and Matsuo, Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization, NeurIPS 2021
- Liu et al., TTT++: When Does Self-Supervised Test-Time Training Fail or Thrive?, NeurIPS 2021

References

Deep learning-based

- Nado et al., Evaluating Prediction-Time Batch Normalization for Robustness under Covariate Shift, ICML 2020 Woskshops
- Karani et al., A Field of Experts Prior for Adapting Neural Networks at Test Time, arXiv 2022
- Xiao et al., Learning to Generalize across Domains on Single Test Samples, ICLR 2022
- Volpi et al., On the Road to Online Adaptation for Semantic Image Segmentation, CVPR 2022
- Wange et al., Continual Test-Time Domain Adaptation, CVPR 2022
- Klingner et al., Continual BatchNorm Adaptation (CBNA) for Semantic Segmentation, IEEE T. on Intelligent Transportation Systems 2022
- Chen et al., Contrastive Test-Time Adaptation, CVPR 2022
- Valanarasu et al., On-the-Fly Test-time Adaptation for Medical Image Segmentation, MIDL 2023
- Yang et al., Test-time Batch Normalization, arXiv 2022
- Bateson et al., Test-Time Adaptation with Shape Moments for Image Segmentation, MICCAI 2022
- Jung et al., CAFA: Class-Aware Feature Alignment for Test-Time Adaptation, arXiv 2022
- Gao et al., Back to the Source: Diffusion-Driven Test-Time Adaptation, CVPR 2023
- Rusak et al., If your data distribution shifts, use self-learning, TMLR 2022
- Niu et al., Efficient Test-Time Model Adaptation without Forgetting, ICML 2022
- Choi et al., Improving Test-Time Adaptation via Shift-agnostic Weight Regularization and Nearest Source Prototypes, ECCV 2022
- Liu et al., Single-domain Generalization in Medical Image Segmentation via Test-time Adaptation from Shape Dictionary, AAAI 2022
- Kojima et al., Robustifying Vision Transformer without Retraining from Scratch by Test-Time Class-Conditional Feature Alignment, IJCAI 2022

References

Deep learning-based

- Thopalli et al., Domain Alignment Meets Fully Test-Time Adaptation, ACML 2022
- Ma et al., Test-time Adaptation with Calibration of Medical Image Classification Nets for Label Distribution Shift, MICCAI 2022
- Saltori et al., GIPSO: Geometrically Informed Propagation for Online Adaptation in 3D LiDAR Segmentation, ECCV 2022
- Cordier et al., Test-Time Adaptation with Principal Component Analysis, ECML/PKDD workshops 2022
- Frey et al., Continual Adaptation of Semantic Segmentation using Complementary 2D-3D Data Representations, RAL 2022
- Boudiaf et al., Parameter-free Online Test-time Adaptation, CVPR 2022
- Gandelsman et al., Test-Time Training with Masked Autoencoders, NeurIPS 2022
- Zhang et al., MEMO: Test Time Robustness via Adaptation and Augmentation, NeurIPS 2022
- Shu et al., Test-Time Prompt Tuning for Zero-Shot Generalization in Vision-Language Models, NeurIPS 2022
- Goyal et al., Test-time Adaptation via Conjugate Pseudo-labels, NeurIPS 2022
- Sinha et al., TeST: Test-time Self-Training under Distribution Shift, WACV 2023
- Khurana et al., SITA: Single Image Test-time Adaptation, arXiv 2021
- Lin et al., Video Test-Time Adaptation for Action Recognition, CVPR 2023
- Yu et al., Mitigating Forgetting in Online Continual Learning via Contrasting Semantically Distinct Augmentations, arXiv 2022
- Lim et al., TTN: A Domain-Shift Aware Batch Normalization in Test-Time Adaptation, ICLR 2023
- Gaillochet et al., TAAL: Test-time Augmentation for Active Learning in Medical Image Segmentation, MICCAI-DALI 2022
- Han et al., Rethinking Precision of Pseudo Label: Test-Time Adaptation via Complementary Learning, arXiv 2023

References

Deep learning-based
- Ma et al., Test-time Adaptation with Calibration of Medical Image Classification Nets for Label Distribution Shift, MICCAI 2022
- Qian and del Hougne, Noise-Adaptive Intelligent Programmable Meta-Imager, arXiv 2022
- Jung et al., CAFA: Class-Aware Feature Alignment for Test-Time Adaptation, arXiv 2023
- Das et al., TransAdapt: A Transformative Framework for Online Test Time Adaptive Semantic Segmentation, ICASSP 2023
- Yang et al., AUTO: Adaptive Outlier Optimization for Online Test-Time OOD Detection, arXiv 2023
- Liang et al., A Comprehensive Survey on Test-Time Adaptation under Distribution Shifts, arXiv 2023
- Yu et al., Benchmarking Test-Time Adaptation against Distribution Shifts in Image Classification, arXiv 2023
- Lim et al., TTN: A Domain-Shift Aware Batch Normalization in Test-Time Adaptation, ICLR 2023
- Li et al., On the Robustness of Open-World Test-Time Training: Self-Training with Dynamic Prototype Expansion, ICCV 2023
- Zhang et al., DomainAdaptor: A Novel Approach to Test-time Adaptation, arXiv 2023
- Hakim et al., ClusT3: Information Invariant Test-Time Training, ICCV 2023
- Bertrand et al., Test-time Training for Matching-based Video Object Segmentation, NeurIPS 2023

Many works surely missing, please also check

- **https://github.com/tim-learn/awesome-test-time-adaptation**
- **https://github.com/YuejiangLIU/awesome-source-free-test-time-adaptation**

Acknowledgments


Gabriela Csurka


Cesar de Souza


Pau de Jorge


Tyler Hayes


Diane Larlus


Grégory Rogez

NAVER LABS
Europe