Continual Learning: The State of the Art?

Riccardo Volpi
riccardo.volpi@naverlabs.com

17 May 2022
University of Verona

NAVER LABS Europe

Our plan for today

- Moving on from the “standard” learning paradigm.
- The continual learning formulation
  - Incremental learning
  - Streaming/online learning
  - Learning new domains vs learning new classes/tasks
- Continual learning benchmarks
- Continual learning methods
  - Regularization methods
  - Memory-based methods
  - Architecture growing methods
- Online unsupervised domain adaptation
- Continual learning @ Naver Labs Europe


(image from Shutterstock)

Naver Labs Europe

Computer Vision

3D Vision

Machine Learning & Optimization

Search & Recommendation

Natural Language Processing

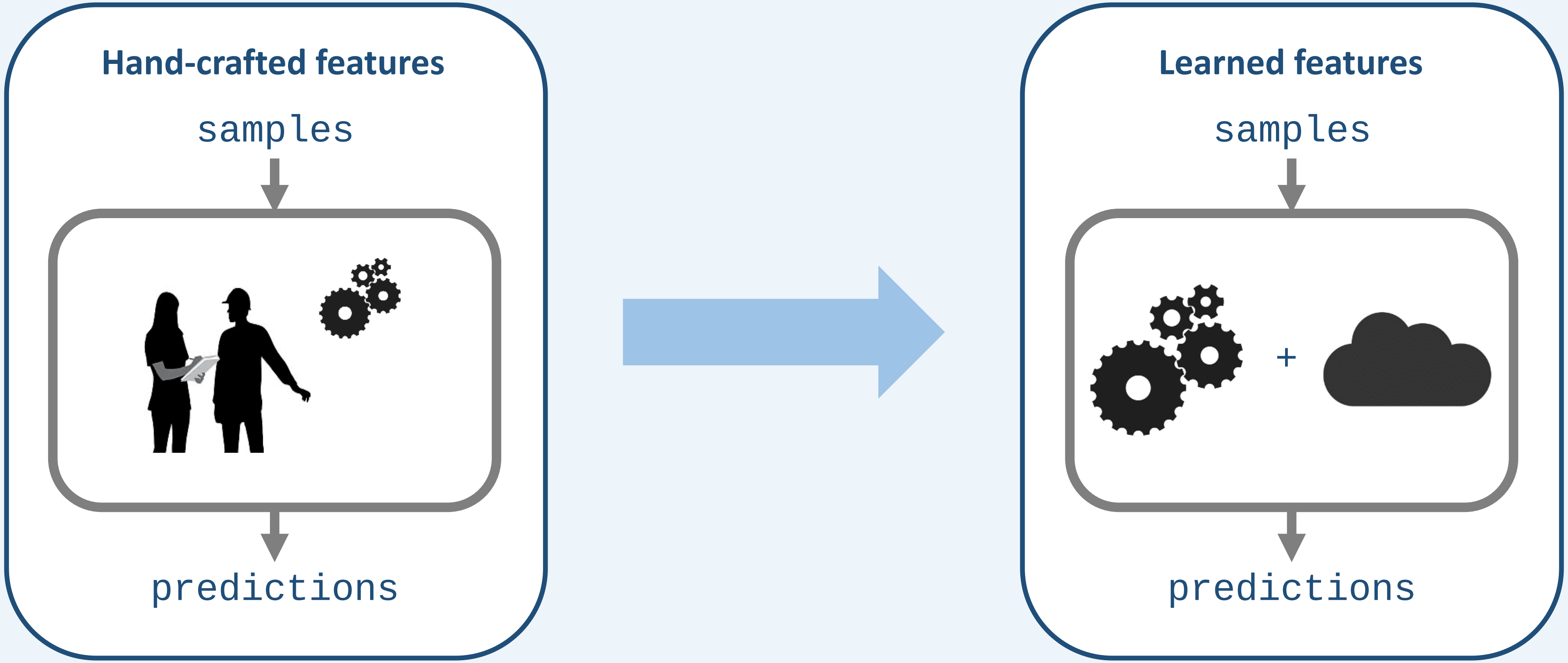
UX & Ethnography

Year-round internships!

The “standard” supervised learning paradigm

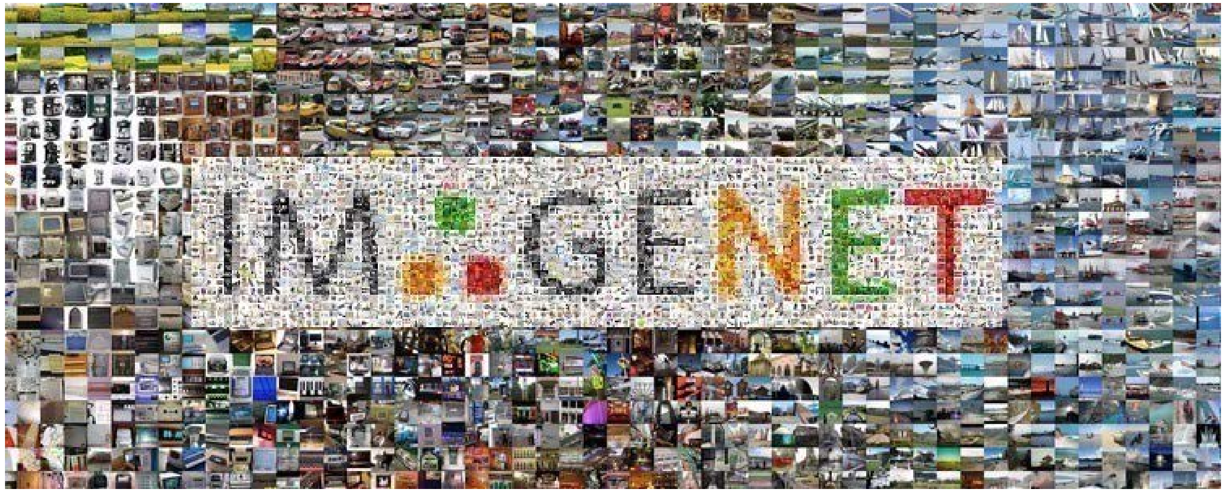
The “standard” supervised learning paradigm

A revolution happened by switching from **hand-crafted** to **data-driven** solutions



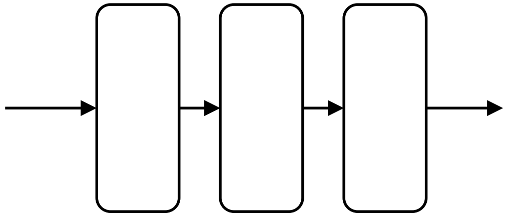
The “standard” supervised learning paradigm

A revolution happened by switching from **hand-crafted** to **data-driven** solutions



+

(your favorite neural net)



PyTorch

TensorFlow

The “standard” supervised learning paradigm

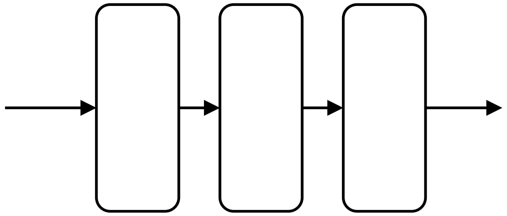
A revolution happened by switching from **hand-crafted** to **data-driven** solutions

Typical ingredients

- **Huge data availability (often labeled)**
- **Time and computational resources**
- **Offline training / seeing each sample again and again**
- **Assumption of an iid world**

+

(your favorite neural net)



The self-supervised learning paradigm

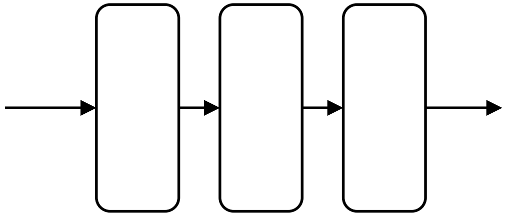
A revolution happened by switching from **hand-crafted** to **data-driven** solutions

Typical ingredients

- **Huge data availability (unlabeled)** ~~(often labeled)~~
- **Time and computational resources**
- **Offline training / seeing each sample again and again**
- **Assumption of an iid world**

+

(your favorite neural net)



A parallel with human learning

In contrast, humans **learn continuously** by processing **streams of samples**.

We are very good **few-shot learners**, and can **generalize to unfamiliar conditions**.

We do extensive **unsupervised learning**.

In short, we don't learn visual categories by staring again and again at annotated photos.

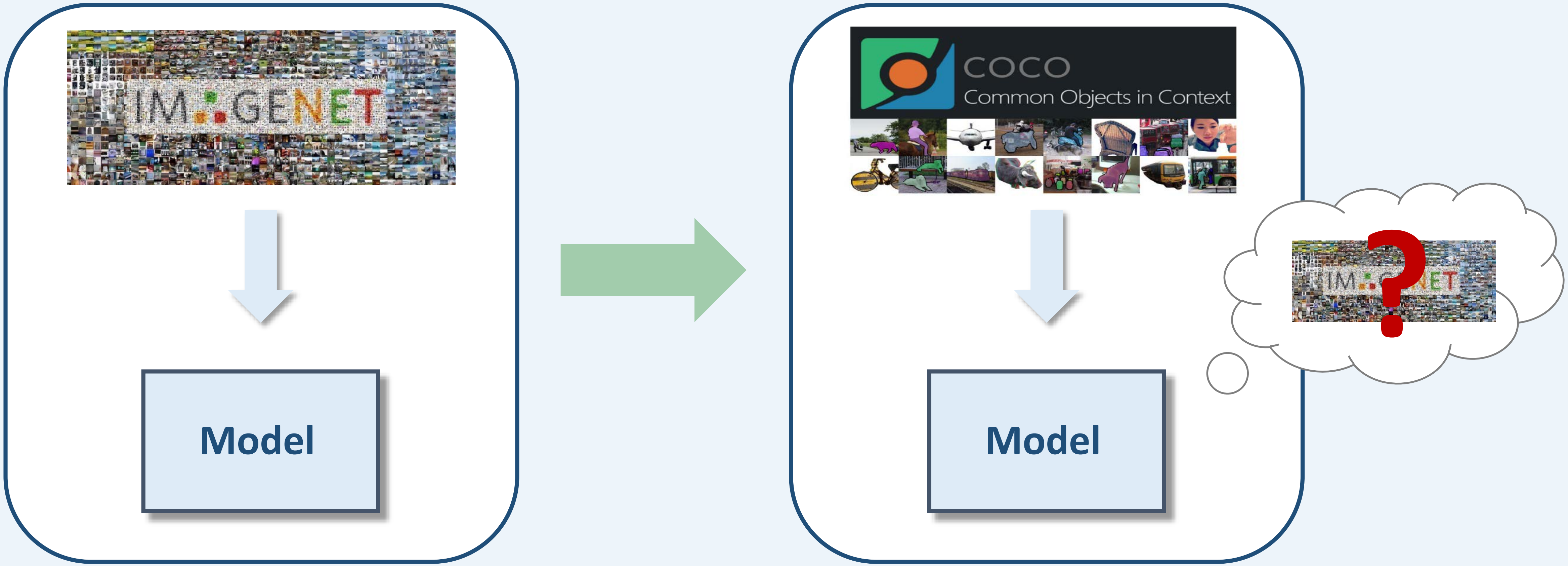
Note: I'm not assuming we can replicate human intelligence with the models we are currently using, but we need to apprehend some of these capabilities, very important e.g. in robotics.

Simple continual learning examples

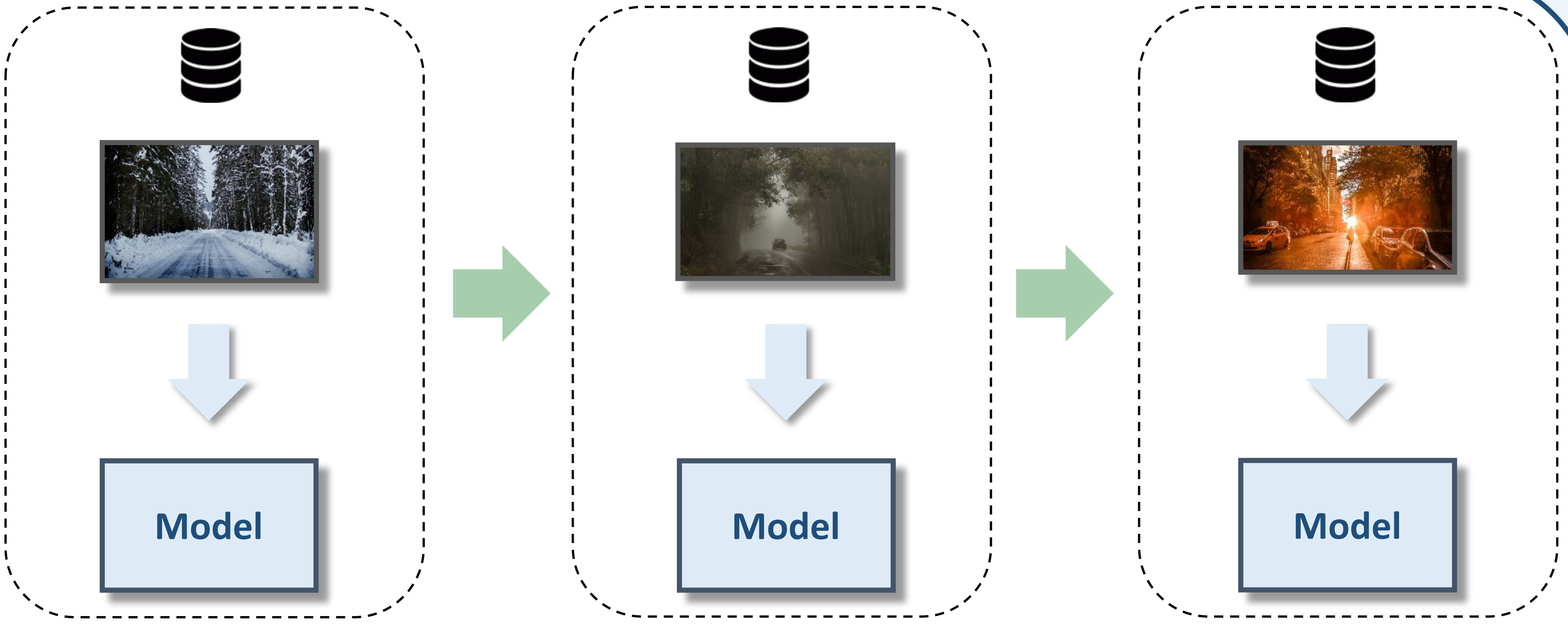
…what can (**and will**) go wrong.

Simple continual learning examples

…what can (and will) go wrong.



Simple continual learning examples



Catastrophic forgetting: models will forget previously learned patterns and tasks.

Continual Learning Formulations

Incremental batch learning

Incremental batch learning

- In **“vanilla” supervised learning**, we are given a task T, that generally involves training a model M using a learning algorithm + a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$.

- In incremental learning, we desire to solve **different tasks** that are given **sequentially**

$$\mathrm{T}_1 \rightarrow \mathrm{T}_2 \rightarrow \mathrm{T}_3 \rightarrow \ldots \rightarrow \mathrm{T}_i \rightarrow \ldots \mathrm{T}_N$$

with the associated **datasets**

$$D_1 \rightarrow D_2 \rightarrow D_3 \rightarrow \ldots \rightarrow D_i \rightarrow \ldots D_N$$

- **Catastrophic forgetting:** after learning task T_i, we under-perfom on tasks $\mathrm{T}_{i-1,\ldots,1}$ (**negative backward transfer**)

- **Positive transfer:** ideally, at task T_i we would like to improve on tasks $\mathrm{T}_{i-1,\ldots,1}$, as well as facilitating learning on $\mathrm{T}_{i+1,\ldots,N}$

Incremental batch learning

- What is a **"task"**?

$$\mathrm{T}_1 \rightarrow \mathrm{T}_2 \rightarrow \mathrm{T}_3 \rightarrow \ldots \rightarrow \mathrm{T}_i \rightarrow \ldots \mathrm{T}_N$$

with the associated **datasets**

$$D_1 \rightarrow D_2 \rightarrow D_3 \rightarrow \ldots \rightarrow D_i \rightarrow \ldots D_N$$

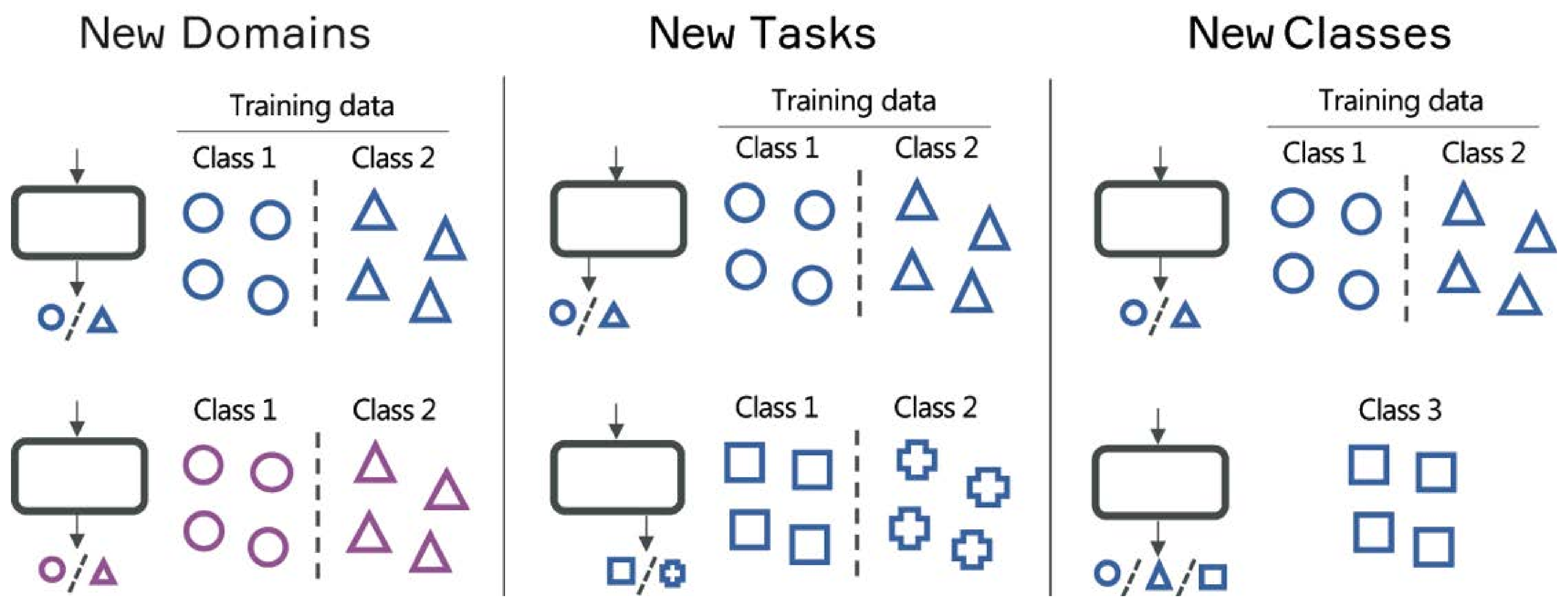
- Two main incremental learning problems in the literature:
  - **Domain**-incremental learning
  - **Class**-incremental learning

- **Domain-incremental learning (AKA Continual Domain Adaptation)**
  - Task does not change $\mathrm{T}_1 = \mathrm{T}_2 = \ldots = \mathrm{T}$ (for example, can be the same class. problem)
  - The domain each dataset is drawn from change $D_1 \sim P_1,\ D_1 \sim P_2, \ldots,$ with $P_1 \neq P_2 \neq \cdots$

- **Class-incremental learning**
  - Task changes $\mathrm{T}_1 \neq \mathrm{T}_2 \neq \ldots \neq \mathrm{T}_N$ (for example, each task associated with different classification problems)
  - The domain each dataset is drawn from may change or not (we may draw samples from different class from same distribution).

- **"Task label"**: at deployment, we should avoid relying on information re: the specific task samples come from.

Incremental batch learning



Streaming/online learning

Streaming/online learning

- We desire to learn from **one sample at a time**
  - We (kind of) lose the notion of “sub-tasks” here

- Let P_t be a **time-dependent distribution**: in **streaming learning**, we process sequences of samples drawn from it

- Such sequence can be written as $((x_t, y_t))_{t=1}^{\infty} \sim P_t$
  - Potentially **never-ending**
  - **Nonstationary**
  - Samples can be **temporally correlated**, if P_t is the real world

- For example, we may receive streams of samples to train a classifier.
  - If the **number of classes is fixed** since the start, we can define the task as **T**
  - If **new classes can arise over time**, the task is also time-dependent, $\mathbf{T_t}$

- How heavily the distribution P_t depends on time, depends on the specific applications
  - May be close to stationary for a warehouse robot, but will vary significantly for an agent exposed to the outdoor.

Continual Learning Benchmarks

Examples of benchmarks

Incremental batch learning

- **MNIST (domain):** learning from ten different versions of MNIST where pixels are randomly permuted.
- **MNIST (class):** divide MNIST in ten different tasks, with ten different label sets.
- **CIFAR-10/100:** divide CIFAR-10/100 in ten different tasks, with ten different label sets.
- **ImageNet**: divide ImageNet in ten different tasks, with ten different label sets.

Streaming learning

- **ImageNet**: divide ImageNet in ten different tasks, with ten label sets. See each sample **once**.
- **iCubWorld**
- **CoRE-50**
- **Stream-51**

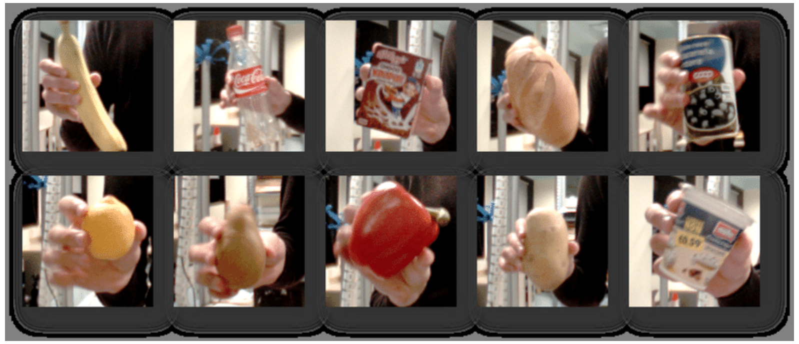
Examples of benchmarks

Incremental batch learning

- **MNIST (domain):** learning from ten different versions of MNIST where pixels are randomly permuted.
- **MNIST (class):** divide MNIST in ten different tasks, with ten different label sets.
- **CIFAR-10/100:** divide CIFAR-10/100 in ten different tasks, with ten different label sets.
- **ImageNet**: divide ImageNet in ten different tasks, with ten different label sets.

Streaming learning

- **ImageNet**: divide ImageNet in ten different tasks, with ten label sets. See each sample **once**.
- **iCubWorld**
- **CoRE-50**
- **Stream-51**



[Pasquale et al., JMLR 2015]

Examples of benchmarks

Incremental batch learning

- **MNIST (domain):** learning from ten different versions of MNIST where pixels are randomly permuted.
- **MNIST (class):** divide MNIST in ten different tasks, with ten different label sets.
- **CIFAR-10/100:** divide CIFAR-10/100 in ten different tasks, with ten different label sets.
- **ImageNet**: divide ImageNet in ten different tasks, with ten different label sets.

Streaming learning

- **ImageNet**: divide ImageNet in ten different tasks, with ten label sets. See each sample **once**.
- **iCubWorld**
- **CoRE-50**
- **Stream-51**

[Lomonaco et al., CoRL 2017]

Examples of benchmarks

Incremental batch learning

- **MNIST (domain):** learning from ten different versions of MNIST where pixels are randomly permuted.
- **MNIST (class):** divide MNIST in ten different tasks, with ten different label sets.
- **CIFAR-10/100:** divide CIFAR-10/100 in ten different tasks, with ten different label sets.
- **ImageNet**: divide ImageNet in ten different tasks, with ten different label sets.

Streaming learning

- **ImageNet**: divide ImageNet in ten different tasks, with ten label sets. See each sample **once**.
- **iCubWorld**
- **CoRE-50**
- **Stream-51**

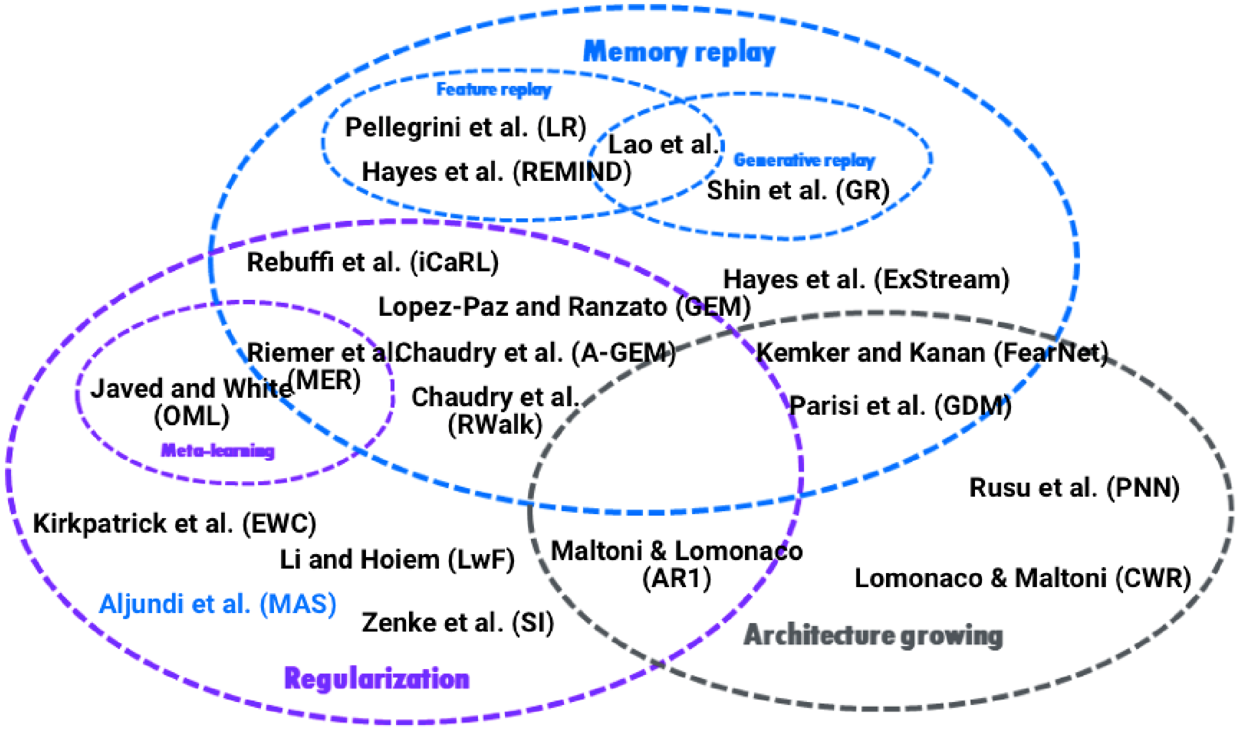
[Roady et al., CVPRW 2020]

Continual learning methods

- We follow Parisi et al. [1] and Maltoni and Lomonaco [2] in categorizing CL approaches.
- We also wrote a more informal blog post about this, find it at [this link].

Three main bodies of work

1. **Regularization**
2. **Memory-based**
3. **Architecture growing**



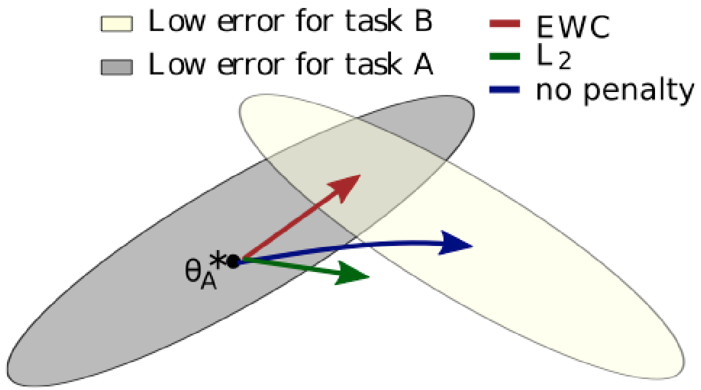
Regularization methods

- Avoid catastrophic forgetting by regularizing the loss at hand (for example, the cross-entropy loss)
- **Pros:** **Principled** approaches, trying to improve the objectives we optimize in deep learning.
- **Cons:** These methods generally work well if provided with the **task label**.

Elastic Weight Consolidation (EWC)

- Moving from task A to task B

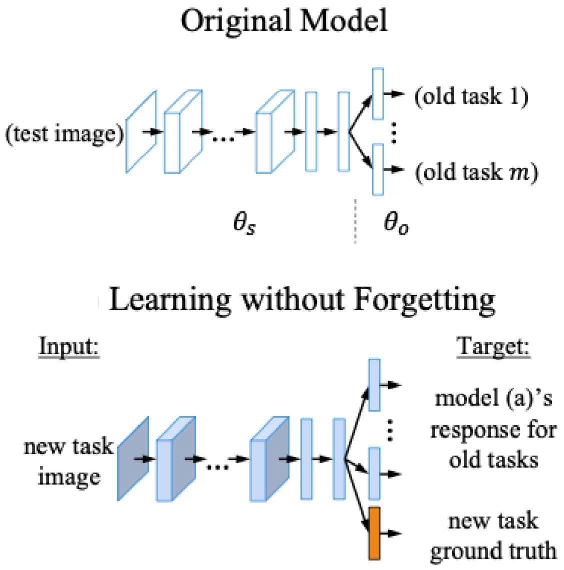
$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2$$



Learning without Forgetting (LwF)

Distillation

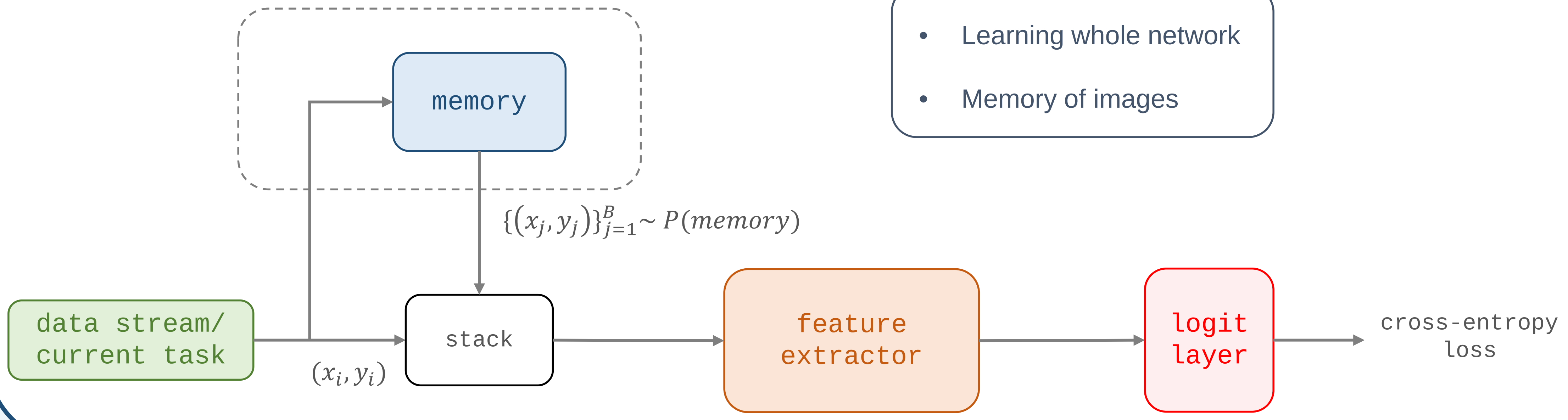
- Model performing well on tasks $1, \dots, m$
- Transferring the model to task $m+1$



[EWC] Kirkpatrick et al., "Overcoming Catastrophic Forgetting in Deep Neural Networks", PNAS 2017
[LwC] Li and Hoiem, "Learning without Forgetting", ECCV 2016

Memory-based methods

- Avoid catastrophic forgetting by rehearsing old samples, stored in memory.
- **Pros:** **Best performing algorithms**.
- **Cons:** Possible memory/privacy issues. Public benchmarks are very small / mostly tested iid.

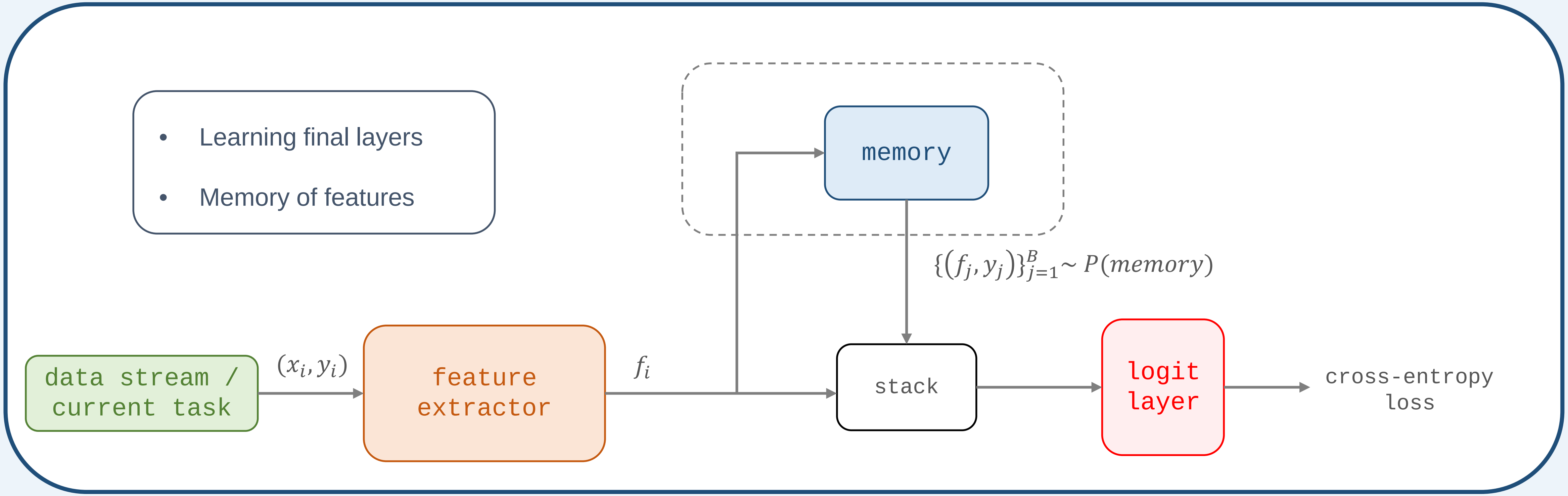


- Learning whole network
- Memory of images

[iCaRl] Rebuffi et al., “iCaRl: Incremental Classifier and Representation Learning”, CVPR 2017
Chaudry et al., “On Tiny Episodic Memories in Continual Learning”, ICML Workshops 2019

Memory-based methods

- Avoid catastrophic forgetting by rehearsing old samples, stored in memory.
- **Pros:** **Best performing algorithms**.
- **Cons:** Possible memory/privacy issues. Public benchmarks are very small / mostly tested iid.

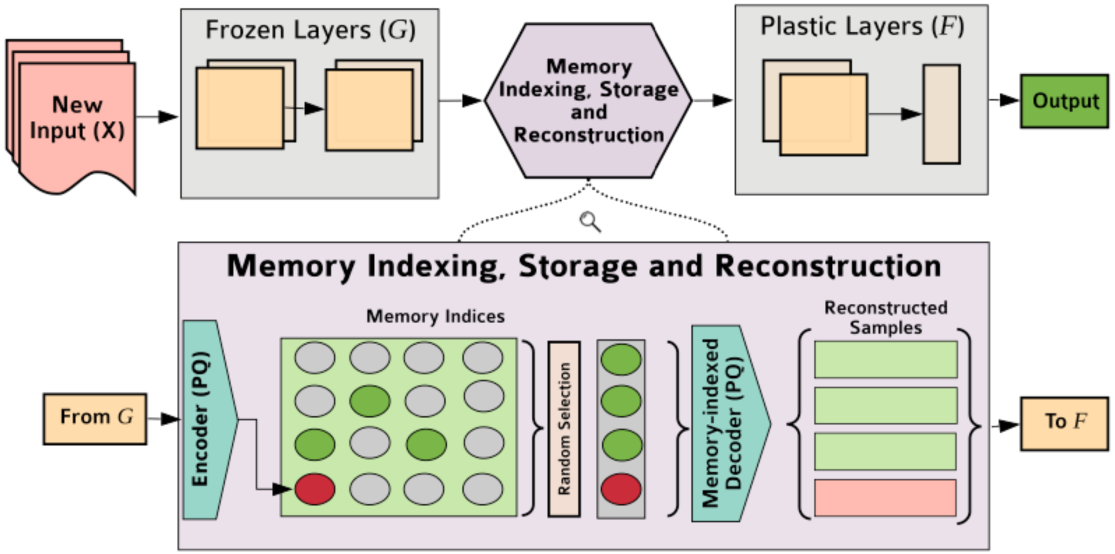


[REMIND] Hayes et al., "REMIND your Network to Prevent Catastrophic Forgetting", ECCV 2020

Pellegrini et al., "Latent Replay for Real-time Continual Learning", arXiv 2019

Memory-based methods

- Avoid catastrophic forgetting by rehearsing old samples, stored in memory.
- **Pros:** **Best performing algorithms**.
- **Cons:** Possible memory/privacy issues. Public benchmarks are very small / mostly tested iid.



REMIND method (ECCV 2020)

- PQ to compress features
- Very low reconstruction error
- Can store many more samples
- Does not do representation learning

[REMIND] Hayes et al., "REMIND your Network to Prevent Catastrophic Forgetting", ECCV 2020

Memory-based methods

- Avoid catastrophic forgetting by rehearsing old samples, stored in memory.
- **Pros:** **Best performing algorithms**.
- **Cons:** Possible memory/privacy issues. Public benchmarks are very small / mostly tested iid.

My opinion:

How to select **which samples to store in memory**
is a very interesting research direction.

Especially when focusing on ML system that
receive millions/billions of samples per day.

<u>Random/reservoir VERY strong baselines.</u>

[REMIND] **Hayes et al., "REMIND your Network to Prevent Catastrophic Forgetting", ECCV 2020**

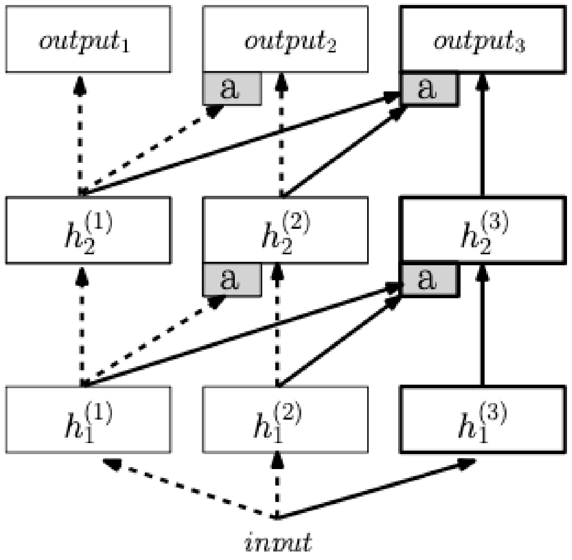
Architecture growing methods

- Avoid catastrophic forgetting by extending the network parameters.
- **Pros:** Freezing some weights effectively copes with catastrophic forgetting.
- **Cons:** Scalability.

Progressive Neural Networks (PNN)

- Used to learn **sequential RL tasks**.
- Define a new neural network (**“column”**) per task.
- **Lateral connections** to exploit previous knowledge.

$$h_i^{(k)} = f\left(W_i^{(k)} h_{i-1}^{(k)} + \sum_{j<k} U_i^{(k:j)} h_{i-1}^{(j)}\right)$$



[PSS] Rusu et al., “Progressive Neural Networks”, arXiv 2016

Hybrid approaches

- Combining different techniques in tandem.

Gradient Episodic Memory (GEM)

- Regularization + Episodic memory
- Enforces **positive backward transfer**, via gradient agreement

$$\begin{aligned} \text{minimize}_\theta \quad & \ell(f_\theta(x,t),y) \\ \text{subject to} \quad & \ell(f_\theta,\mathcal{M}_k) \leq \ell(f_\theta^{t-1},\mathcal{M}_k) \text{ for all } k<t, \end{aligned}$$

$$\langle g, g_k \rangle := \left\langle \frac{\partial \ell(f_\theta(x,t),y)}{\partial \theta}, \frac{\partial \ell(f_\theta,\mathcal{M}_k)}{\partial \theta} \right\rangle \geq 0, \text{ for all } k<t.$$

- Need to solve a QP. Improved in **A-GEM**.

[GEM] Lopez-Paz and Ranzato, "Gradient Episodic Memory for Continual Learning", NeurIPS 2017
[A-GEM] Chaudry et al., "Efficient Lifelong Learning with A-GEM", ICLR 2019

Some results

- From **REMIND** paper

Model	ImageNet	CORe50			
	CLS IID	IID	CLS IID	INST	CLS INST
Fine-Tune (θ_F)	0.288	0.961	0.334	0.851	0.334
ExStream	0.569	0.953	0.873	0.933	0.854
SLDA	0.752	0.976	0.958	0.963	0.959
iCaRL	0.306	-	0.690	-	0.644
Unified	0.614	-	0.510	-	0.527
BiC	0.440	-	0.410	-	0.415
REMIND	**0.855**	**0.985**	**0.978**	**0.980**	**0.979**
Offline (θ_F)	0.929	0.989	0.984	0.985	0.985
Offline	1.000	1.000	1.000	1.000	1.000

	CORe50			
	CLS IID		CLS INST	
Model	TL	No TL	TL	No TL
SI	0.895	0.417	0.905	0.416
EWC	0.893	0.413	0.903	0.413
MAS	0.897	0.415	0.905	0.421
RWALK	0.903	0.410	0.912	0.417
A-GEM	0.925	0.417	0.916	0.421
REMIND	**0.995**	**0.978**	**0.995**	**0.979**
Offline	1.000	1.000	1.000	1.000

[REMIND] Hayes et al., "REMIND your Network to Prevent Catastrophic Forgetting", ECCV 2020

Methods summary…

Family of methods	Pros	Cons
Regularization	Lightweight, Privacy	Difficult to avoid forgetting
Architecture growing	Easy to avoid forgetting, Privacy	Increasing memory (↑↑)
Memory replay	Easy to avoid forgetting	Increasing memory (↑), Privacy

Volpi R., Larlus D., Rogez G., "The Short Memory of Artificial Neural Networks", NLE's blog.

Continual learning @Naver Labs Europe

- Worked on **continual learning** and **continual domain adaptation**
  - Volpi et al., “Continual Adaptation of Visual Representations via Domain Randomization and Meta-learning”, CVPR 2021 (Oral)
  - Volpi et al., “On the Road to Online Adaptation for Semantic Image Segmentation”, CVPR 2022

- We have been particularly interested in **online/unsupervised domain adaptation**.

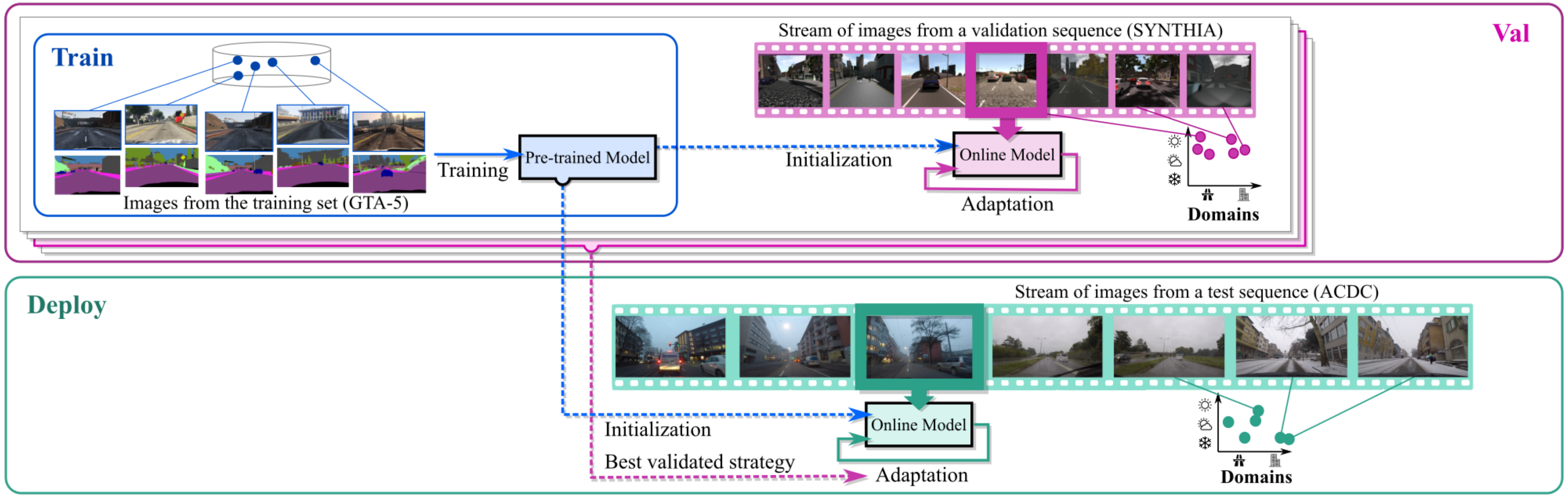
- We have been targeting more challenging computer vision tasks
  - In particular, we have been focusing a lot on **semantic image segmentation**.

Online unsupervised domain adaptation

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

Online unsupervised domain adaptation

- The goal is adapting **frame-by-frame** to sequences of **temporally correlated, unlabeled samples**.
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself.
- We proposed the **OASIS benchmark** **(Online Adaptation for Semantic Image Segmentation)**



Online unsupervised domain adaptation

- The goal is adapting **frame-by-frame** to sequences of **temporally correlated, unlabeled samples**.
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself.
- We proposed the **OASIS benchmark** **(Online Adaptation for Semantic Image Segmentation)**

- Differences with
  - **(Standard) unsupervised domain adaptation**: no access to target(s) distribution(s).
  - **Domain generalization:** possibility to adapt/possily store some target samples.
  - **Other continual learning frameworks:** unlabelled.
  - **Test-time adaptation:** in principle none, but different evaluation/hypotheses.

Online unsupervised domain adaptation

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

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

[OASIS] Volpi R., De Jorge P., Larlus D. and Csurka G., "On the Road to Online Adaptation for Semantic Image Segmentation", *to appear at CVPR 2022*

Online unsupervised domain adaptation

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

- Some methods:
  - **Self-training with pseudo-labels**
  - **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

[Lee et al., "**Pseudo-label: The Simple and Efficient Semi-supervised Learning Method for Deep Neural Networks**" ICMLW 2013]

[OASIS] **Volpi R., De Jorge P., Larlus D. and Csurka G., "On the Road to Online Adaptation for Semantic Image Segmentation",** ***to appear at CVPR 2022***

Online unsupervised domain adaptation

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

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

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

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

[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]

Online unsupervised domain adaptation

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

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

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

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

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

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

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

Online unsupervised domain adaptation

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

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

Solve a side SSL objective on the target samples

[Sun et al., "**Test-Time Training with Self-Supervision for Generalization under Distribution Shifts**" ICML 2020]

Online unsupervised domain adaptation

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

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

- **"Test-time adaptation"** is becoming an active research field, with different methods published ~monthly
  - Mostly for robustness against corruptions, but all these methods can be used by a continual learner

[OASIS] Volpi R., De Jorge P., Larlus D. and Csurka G., "On the Road to Online Adaptation for Semantic Image Segmentation", *to appear at CVPR 2022*

Online unsupervised domain adaptation

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

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

- **"Test-time adaptation"** is becoming an active research field, with different methods published ~monthly
  - Mostly for robustness against corruptions, but all these methods can be used by a continual learner

- Similar formulations studied for 3D depth estimation

  [Tonioni et al., "**Real-time Self-adaptive Deep Stereo**", CVPR 2019]

  [Poggi et al., "**Continual Adaptation for Deep Stereo**", TPAMI 2021]

[OASIS] Volpi R., De Jorge P., Larlus D. and Csurka G., **"On the Road to Online Adaptation for Semantic Image Segmentation"**, *to appear at CVPR 2022*

Online unsupervised domain adaptation

- The goal is adapting **frame-by-frame** to sequences 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: also here, **catastrophic forgetting**!
- We're learning in an unsupervised way, so it's not trivial how to avoid the model to forget classes.
- In practical terms, classes that are more rare will disappear, leaving their space to the more abundant ones.

- **Example:** if we're adapting a segmentation model for urban street segmentation, it's very easy to forget about ***things*** (countable objects), overtaken by the more abundant ***stuff*** (street, sky, buildings, etc.)

[OASIS] **Volpi R., De Jorge P., Larlus D. and Csurka G., "On the Road to Online Adaptation for Semantic Image Segmentation",** ***to appear at CVPR 2022***

Online unsupervised domain adaptation

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

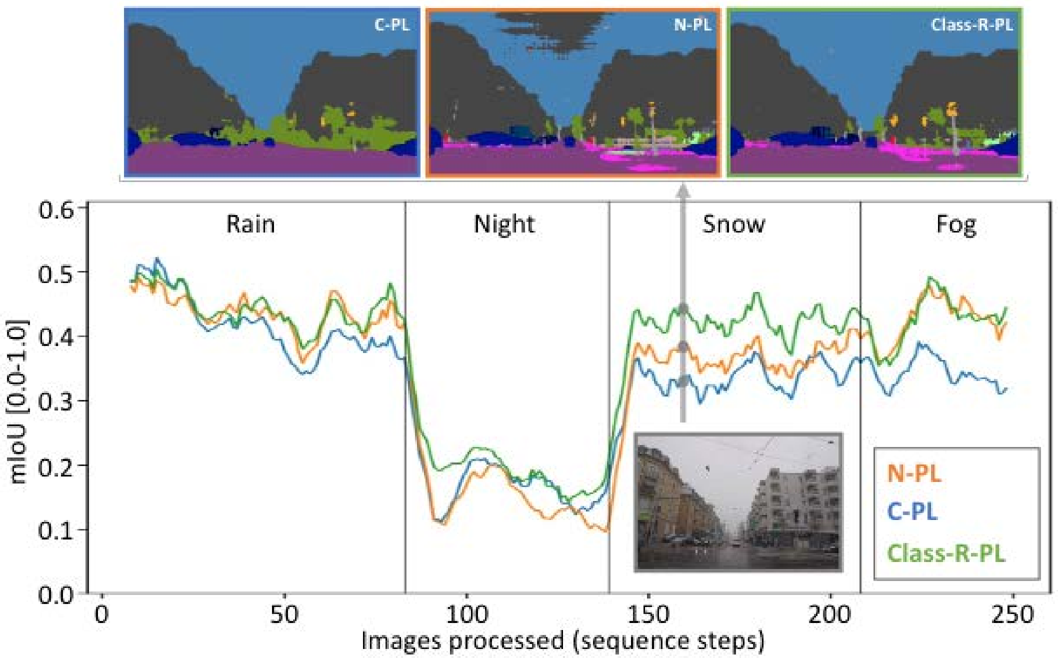
- Main problem: also here, **catastrophic forgetting**!
- We're learning in an unsupervised way, so it's not trivial how to avoid the model to forget classes.
- In practical terms, classes that are more rare will disappear, leaving their space to the more abundant ones.

- **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.

[OASIS] Volpi R., De Jorge P., Larlus D. and Csurka G., "On the Road to Online Adaptation for Semantic Image Segmentation", *to appear at CVPR 2022*

Online unsupervised domain adaptation

- The goal is adapting **frame-by-frame** to sequences of **temporally correlated, unlabeled samples**.
- Each sample from the sequence $(x_t)_{t=1}^{\infty} \sim P_t$ represents an adaptation problem itself.
- We don't care about performance on previous tasks here: it's a continuous stream, and **we only care about the present**.



Online unsupervised domain adaptation

		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 (++)
	N-ST (NN)	$+0.7\%$ ± 1.7	-5.1% ± 0.8	$+2.9\%$ ± 1.1	$+1.0\%$ ± 0.3	ST optim. & NN (+++)	Source set (++)
Naive adapt.	N-BN	$+2.7\%$ ± 0.8	$+2.4\%$ ± 0.6	$+1.9\%$ ± 0.8	$+1.2\%$ ± 0.1	BN stat. update (*)	-
	N-PL	$+3.5\%$ ± 1.0	$+2.9\%$ ± 0.6	$+2.4\%$ ± 1.0	$\mathbf{+1.4\%}$ $\pm \mathbf{0.2}$	$\mathcal{O}$(trainsteps) (+)	-
	N-TENT	$\mathbf{+8.5\%}$ $\pm \mathbf{3.1}$	$+4.9\%$ ± 2.0	$+3.1\%$ ± 3.6	-1.2% ± 0.7	$\mathcal{O}$(trainsteps) (+)	-
CL Vanilla	C-BN	$+6.1\%$ ± 3.7	$+6.8\%$ ± 3.6	$+7.7\%$ ± 4.3	-0.1% ± 1.4	BN stat. update (*)	-
	C-PL	-19.9% ± 12.0	-11.7% ± 8.1	-9.4% ± 8.9	-17.4% ± 3.2	$\mathcal{O}$(trainsteps) (+)	-
	C-TENT	$+2.5\%$ ± 6.8	$+2.7\%$ ± 6.7	$+6.4\%$ ± 5.6	-0.9% ± 1.2	$\mathcal{O}$(trainsteps) (+)	-
CL SrcReg	C-PL-SR	$+4.9\%$ ± 3.9	$+2.8\%$ ± 2.9	$+3.9\%$ ± 3.2	$+0.5\%$ ± 0.5	$\mathcal{O}$(trainsteps) (+)	Source set (++)
	C-TENT-SR	$+7.2\%$ ± 4.0	$+5.8\%$ ± 3.7	$+4.7\%$ ± 3.7	$+0.2\%$ ± 0.5	$\mathcal{O}$(trainsteps) (+)	Source set (++)
CL Reset	Class-R-PL	$+7.2\%$ ± 3.9	$\mathbf{+8.2\%}$ $\pm \mathbf{3.4}$	$\mathbf{+9.0\%}$ $\pm \mathbf{5.1}$	$+0.0\%$ ± 1.4	$\mathcal{O}$(trainsteps) (+)	Backup net (+)
	Class-R-TENT	$\mathbf{+8.3\%}$ $\pm \mathbf{4.2}$	$+7.3\%$ ± 3.9	$\mathbf{+9.1\%}$ $\pm \mathbf{4.9}$	$+0.9\%$ ± 1.3	$\mathcal{O}$(trainsteps) (+)	Backup net (+)
CL Oracle	Oracle-R-PL	$+10.8\%$ ± 4.5	$+11.6\%$ ± 3.8	$+12.7\%$ ± 5.6	$+2.9\%$ ± 1.4	$\mathcal{O}$(trainsteps) (+)	Backup net (+)
	Oracle-R-TENT	$+11.4\%$ ± 4.4	$+10.9\%$ ± 4.1	$+12.2\%$ ± 5.9	$+1.9\%$ ± 1.4	$\mathcal{O}$(trainsteps) (+)	Backup net (+)

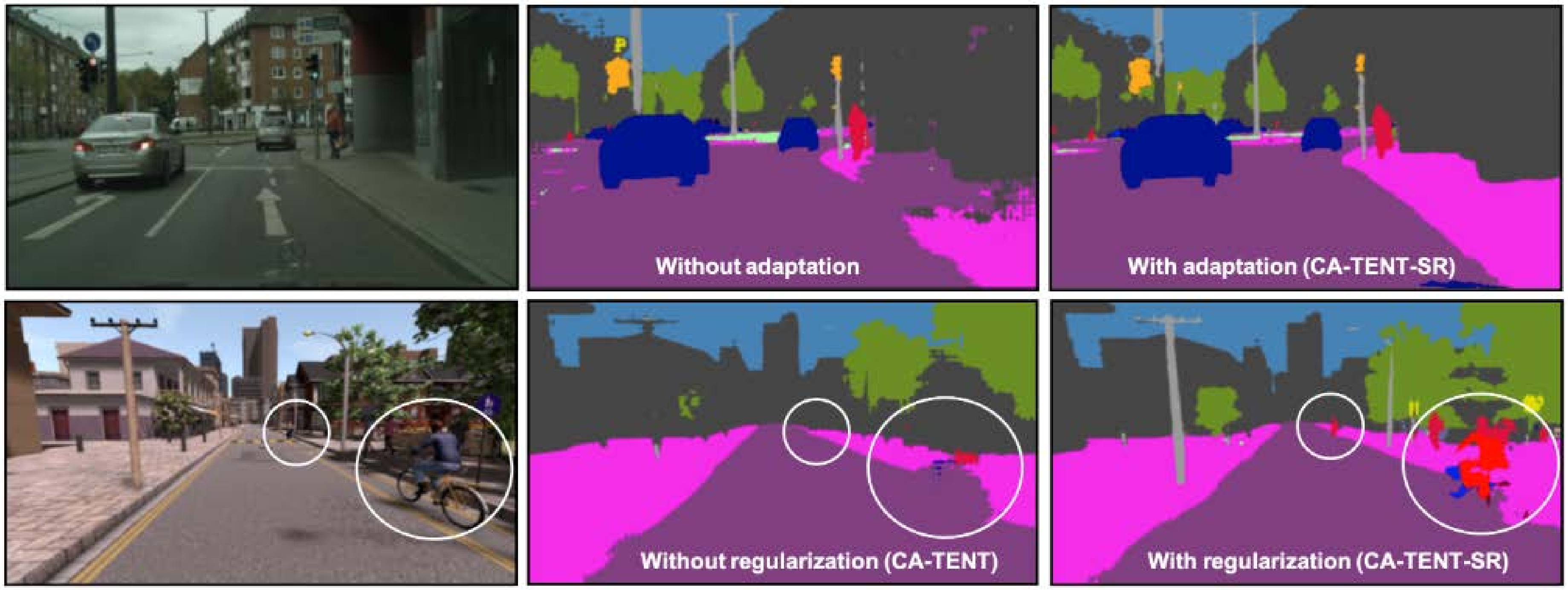
Code available at
github.com/naver/oasis

[OASIS] Volpi R., De Jorge P., Larlus D. and Csurka G., "On the Road to Online Adaptation for Semantic Image Segmentation", *to appear at CVPR 2022*

Online unsupervised domain adaptation

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

Some results…



[OASIS] Volpi R., De Jorge P., Larlus D. and Csurka G., "On the Road to Online Adaptation for Semantic Image Segmentation", *to appear at CVPR 2022*

Wrapping-up…

- Many **formulations** and **family of methods** to start from in **continual learning**.
- The main issue that the community is focusing on is still **catastrophic forgetting**.
- Methods that use **episodic memories** perform better by large margins. Yet, trade-offs with memory overheads and privacy issues.
- We have found several CL sub-problems that are interesting within the **computer vision** sphere.
- We don't plan to stop researching on CL any soon, **feel free to reach out about this** ;-)

(image from Shutterstock)

Acknowledgments


Cesar de Souza


Diane Larlus


Gabriela Csurka


Grégory Rogez


Yannis Kalantidis


Pau de Jorge

MIAI Grenoble Alpes
Multidisciplinary Institute In Artificial Intelligence

NAVER LABS Europe


(image from Shutterstock)