Continual Learning with Deep Networks

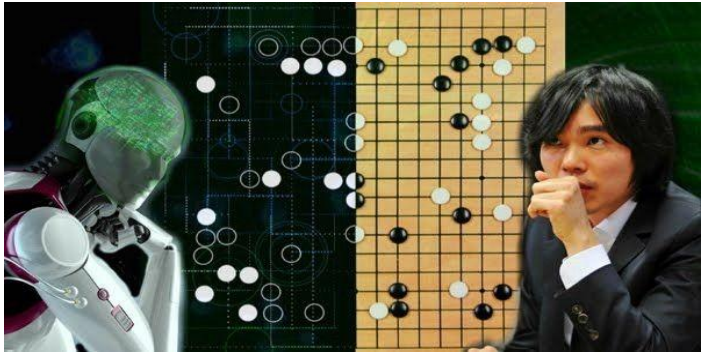
Advanced Computer Vision

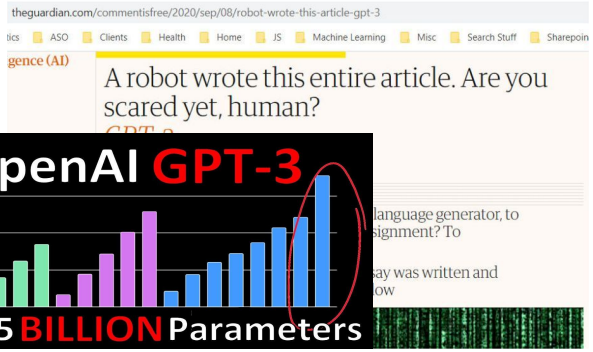
Join Zoom Meeting
https://unitn.zoom.us/j/2852627866

Meeting ID: 285 262 7866
Passcode: 474605

Some slides are taken from Irina Rish

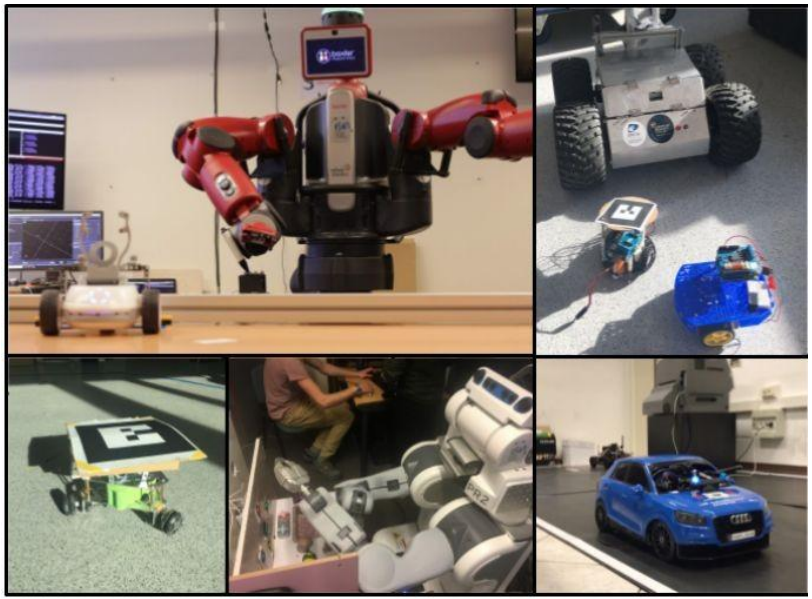
AI Today: Impressive... but (Still) “Narrow"





Challenge

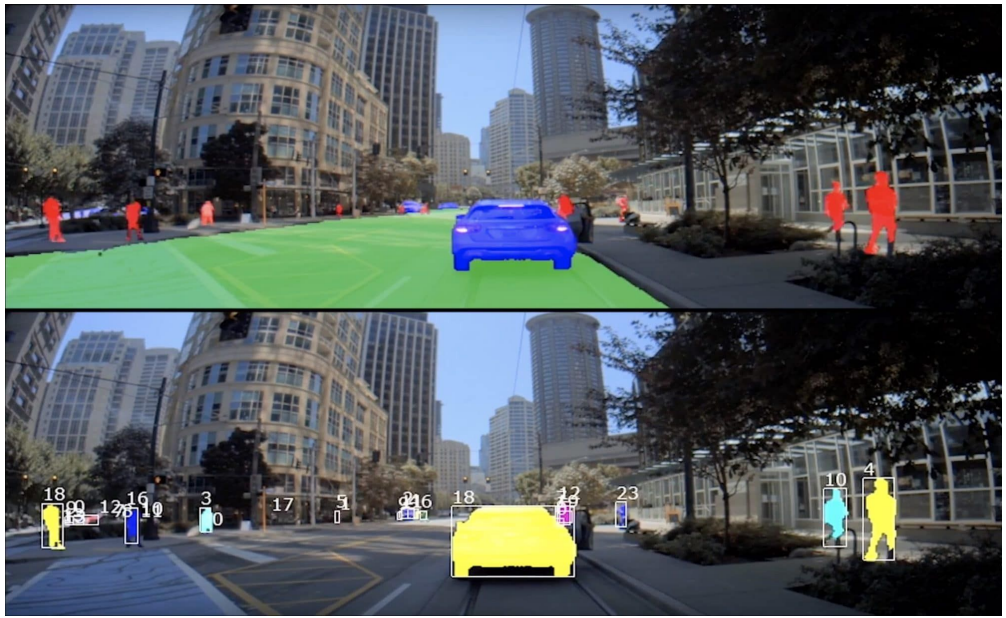
A robot acquiring **new skills** in different environment, adapting to new situations, learning **new tasks**



S. Thrun and T. Mitchell. Lifelong robot learning. *Robotics and Autonomous Systems*, 15:25-46, 1995.

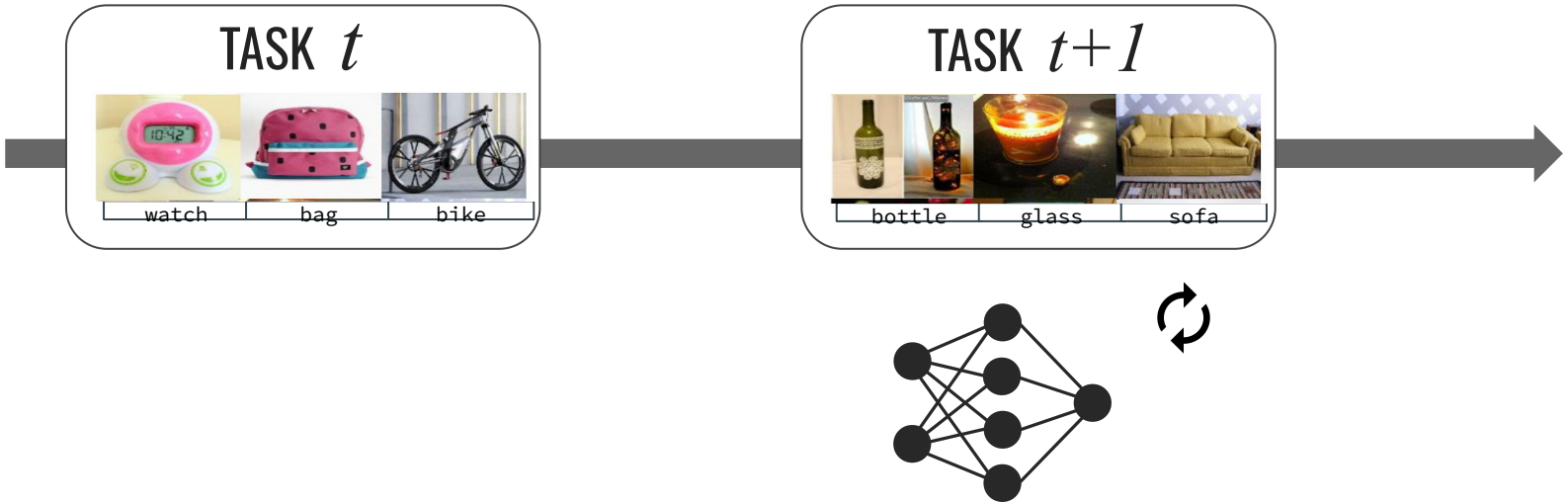
Challenge

A self-driving car adapting to **different environments** (e.g. from a country road to a city)



Continual Learning (CL)

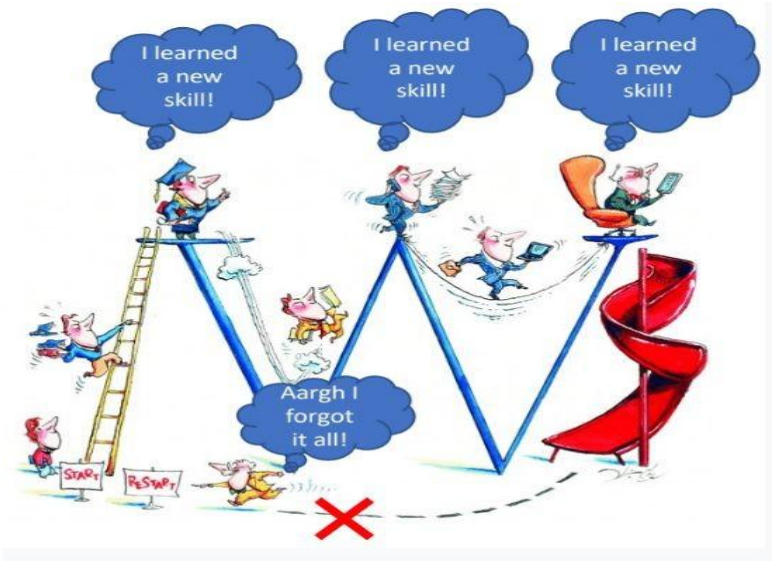
Emulating the human ability to **sequentially learn new tasks** while being able to retain knowledge obtained from past experiences.



Catastrophic forgetting

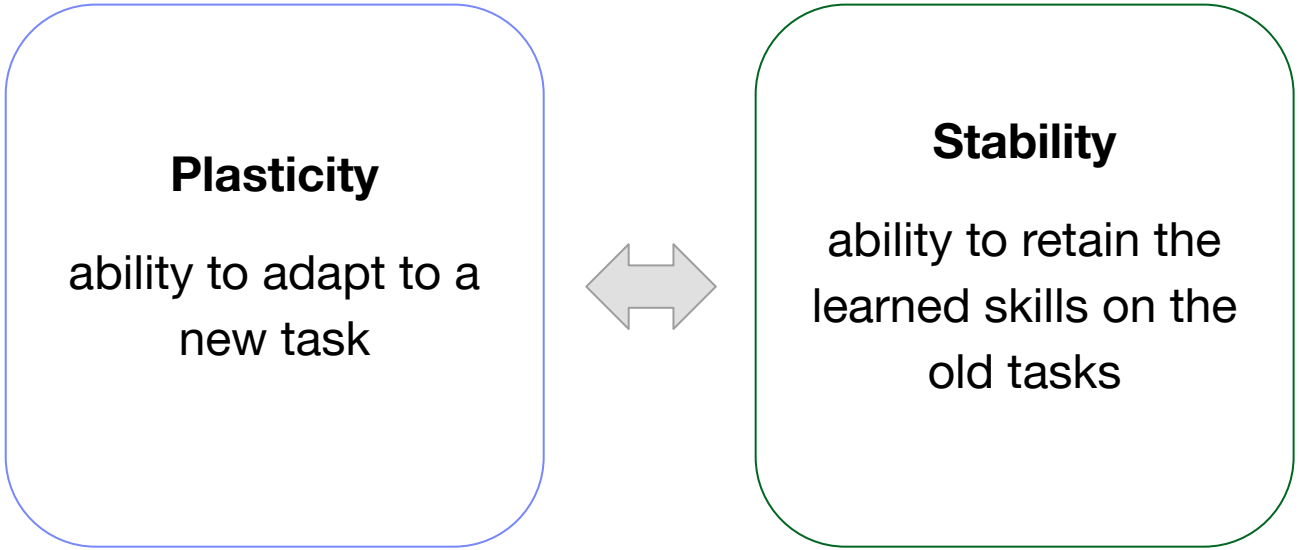
Challenge: Catastrophic Forgetting

"...the process of learning a new set of patterns suddenly and completely erase the network's knowledge of what it had already learned." (French, 1999).



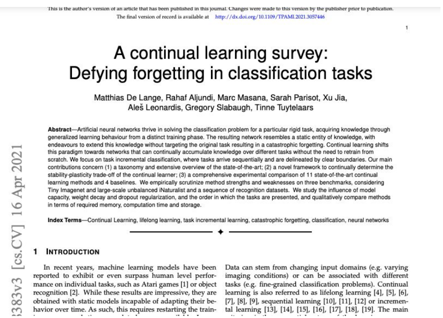
French, Robert M. "Catastrophic forgetting in connectionist networks." *Trends in cognitive sciences* 3.4 (1999): 128-135.

General Problem: Stability vs Plasticity



Grossberg. S. (1982) Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.
Carpenter, G. and Grossberg, S. (1987) ART 2: Self-organization of stable category recognition codes for analog input pattern

CL Methods



This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TPAMI.2021.3057446

1

A continual learning survey: Defying forgetting in classification tasks

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, Tinne Tuytelaars

Abstract—Artificial neural networks thrive in solving the classification problem for a particular rigid task, acquiring knowledge through generalized learning behaviour from a distinct training phase. The resulting network resembles a static entity of knowledge, with endeavours to extend this knowledge without targeting the original task resulting in a catastrophic forgetting. Continual learning shifts this paradigm towards networks that can continually accumulate knowledge over different tasks without the need to retrain from scratch. We focus on task incremental classification, where tasks arrive sequentially and are delineated by clear boundaries. Our main contributions concern (1) a taxonomy and extensive overview of the state-of-the-art; (2) a novel framework to continually determine the stability-plasticity trade-off of the continual learner; (3) a comprehensive experimental comparison of 11 state-of-the-art continual learning methods and 4 baselines. We empirically scrutinize method strengths and weaknesses on three benchmarks, considering Tiny Imagenet and large-scale unbalanced iNaturalist and a sequence of recognition datasets. We study the influence of model capacity, weight decay and dropout regularization, and the order in which the tasks are presented, and qualitatively compare methods in terms of required memory, computation time and storage.

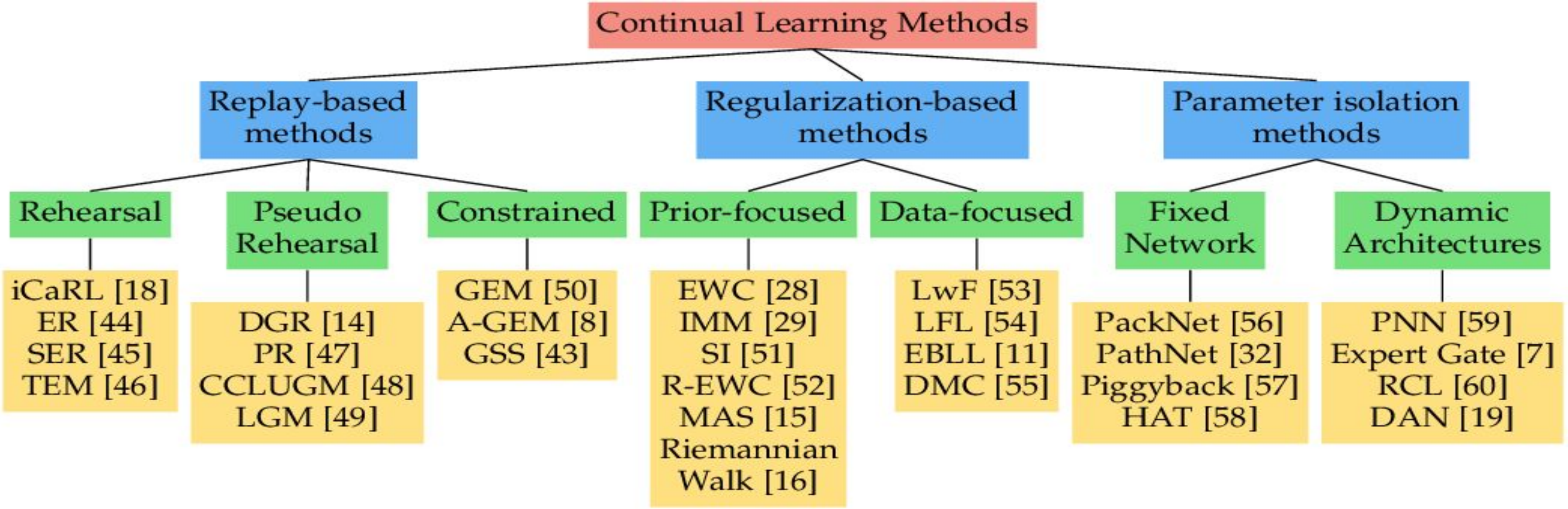
Index Terms—Continual Learning, lifelong learning, task incremental learning, catastrophic forgetting, classification, neural networks

16 Apr 2021 [cs.CV]

1 Introduction

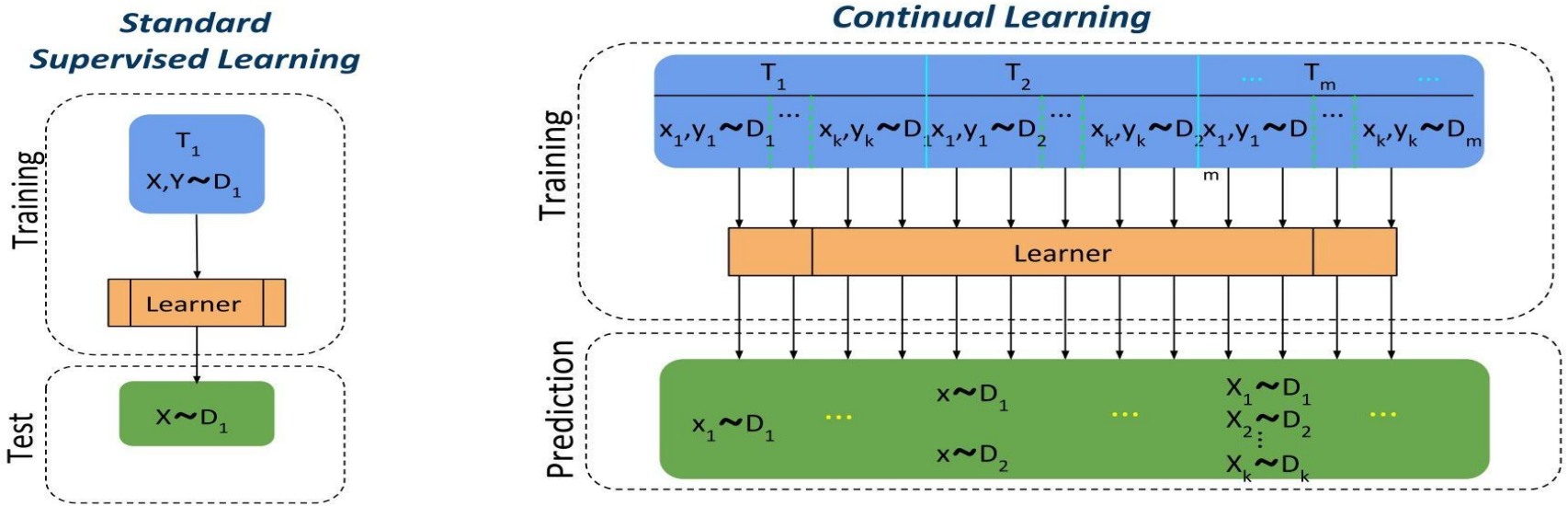
In recent years, machine learning models have been reported to exhibit or even surpass human level performance on individual tasks, such as Atari games [1] or object recognition [2]. While these results are impressive, they are obtained with static models incapable of adapting their behavior over time. As such, this requires restarting the train-

Data can stem from changing input domains (e.g. varying imaging conditions) or can be associated with different tasks (e.g. fine-grained classification problems). Continual learning is also referred to as lifelong learning [4], [5], [6], [7], [8], [9], sequential learning [10], [11], [12] or incremental learning [13], [14], [15], [16], [17], [18], [19]. The main



De Lange, Matthias, et al. "A continual learning survey: Defying forgetting in classification tasks." *IEEE TPAMI* 44.7 (2021): 3366-3385.

Supervised Continual Learning



Continual Learning Settings

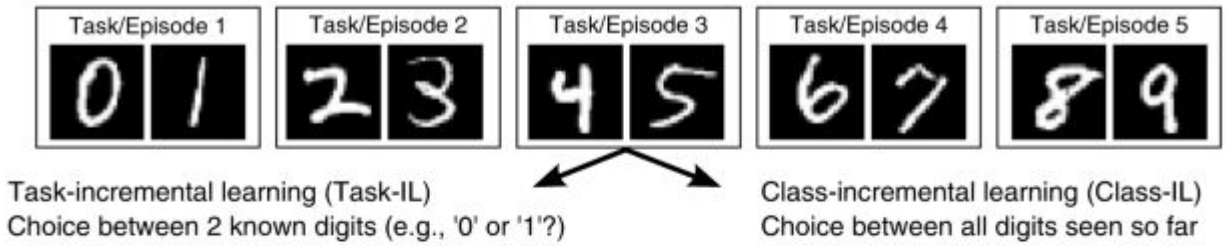
X – input data

Y – class label

T – task (context)

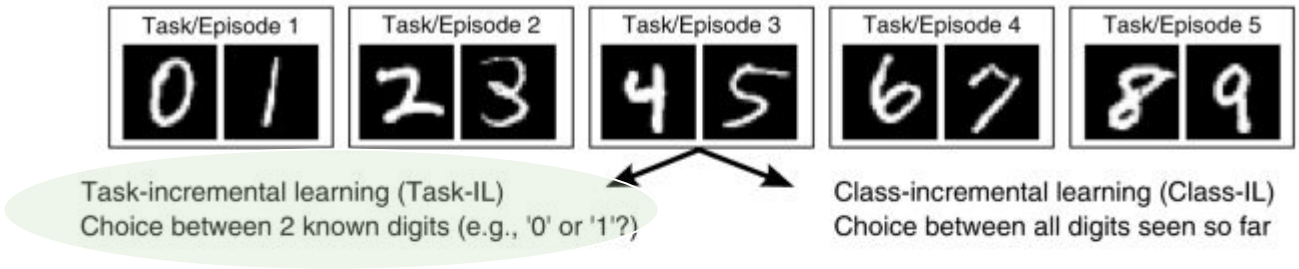
Task ID observed at training:

- T observed at test: **task-incremental CL**
- T not observed at test: **class-incremental CL**



Task-Incremental CL

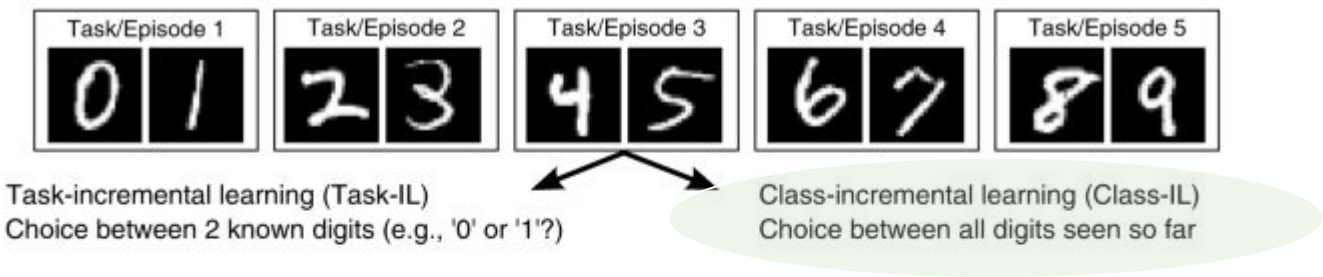
- Models are always informed about which task needs to be performed (both at train and test time)
- The easiest continual learning scenario
- Possible to train models with task-specific components
- A typical NN architecture: “multihead” output layer
  - Each task has its own output units but the rest of the network may be shared between tasks



Hsu et al (2018). Re-evaluating continual learning scenarios: A categorization and case for strong baselines.

Class-Incremental CL

- Models must be able not only to solve each task seen so far, but also to **infer which task** they are presented with.
- Includes protocols in which new classes need to be learned incrementally.



Hsu et al (2018). Re-evaluating continual learning scenarios: A categorization and case for strong baselines.

Continual Learning Settings

X – input data

Y – class label

T – task (context) defines $P(\mathbf{X},\mathbf{Y}|\ \mathbf{T})$

Task ID/boundary is not known at training:

- **Task-agnostic CL**

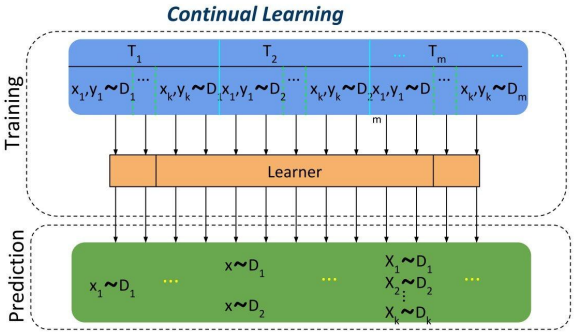
Real world Datasets: CORe50

- 50 domestic objects belonging to 10 categories
- Classification can be performed at object level (50 classes) or at category level (10 classes)
- 11 distinct collecting sessions (8 indoor and 3 outdoor) with different backgrounds and lighting
- For each session and for each object, a 15 seconds video (at 20 fps) has been recorded with a Kinect 2.0 sensor delivering 300 RGB-D frames.

Continual Learning Objective

Data $(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})$ is randomly drawn from distribution $D^{(t)}$, with $\mathcal{X}^{(t)}$ a set of data samples for task t, and $\mathcal{Y}^{(t)}$ the corresponding ground truth labels. The goal is to control the statistical risk of all seen tasks given limited or no access to data $(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})$ from previous tasks $t < \mathcal{T}$:

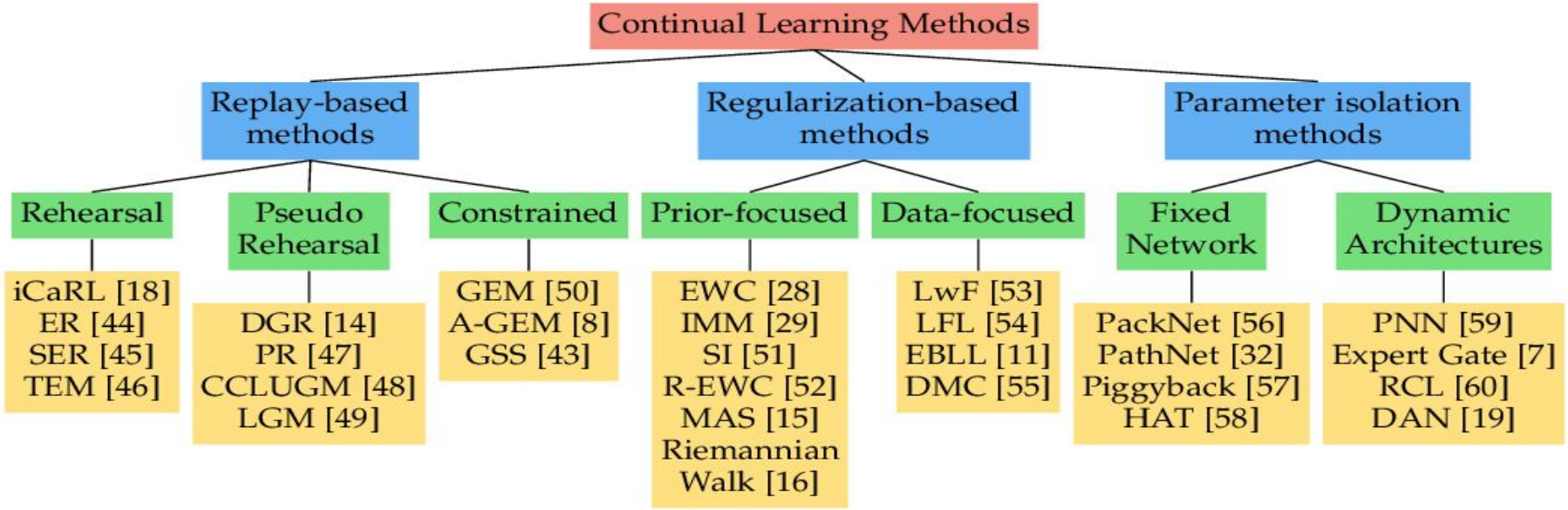
$$\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})}[\ell(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$$



But we have **no access** to the previous tasks, thus cannot compute this empirical risk exactly

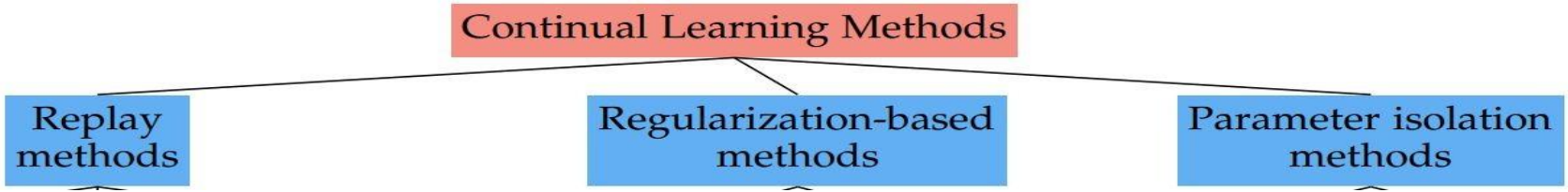
De Lange, Matthias, et al. "A continual learning survey: Defying forgetting in classification tasks." *IEEE TPAMI* 44.7 (2021): 3366-3385.

Continual Learning Taxonomy



De Lange, Matthias, et al. "A continual learning survey: Defying forgetting in classification tasks." *IEEE TPAMI* 44.7 (2021): 3366-3385.

Continual Learning Taxonomy



Memory-based

Maintain a subset of (representative) samples from previous tasks (raw samples or pseudo-samples – e.g., use a generative model).

Replay: reuse these samples as additional inputs in the future, or use them to constrain the new task loss.

No (explicit) memory

Instead, add regularization term to the loss function, consolidating previous knowledge when learning on new data.

Better for privacy, minimizing memory size and access latency.

Architecture-based

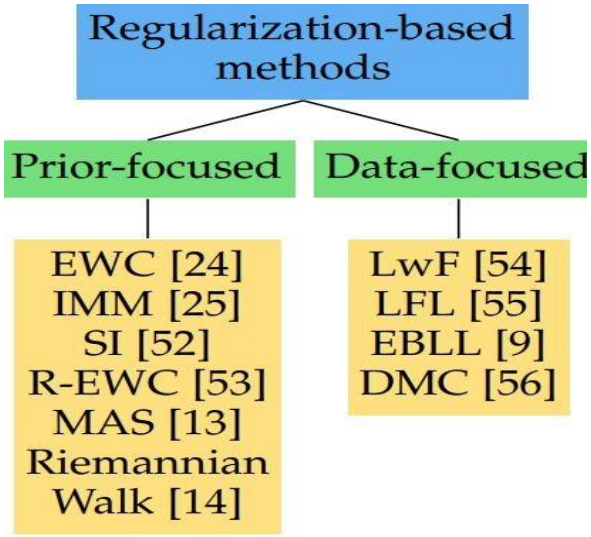
Each task will use different model parameters. Can use a static architecture or keep expanding it.

How will the overall performance scale with the model size?

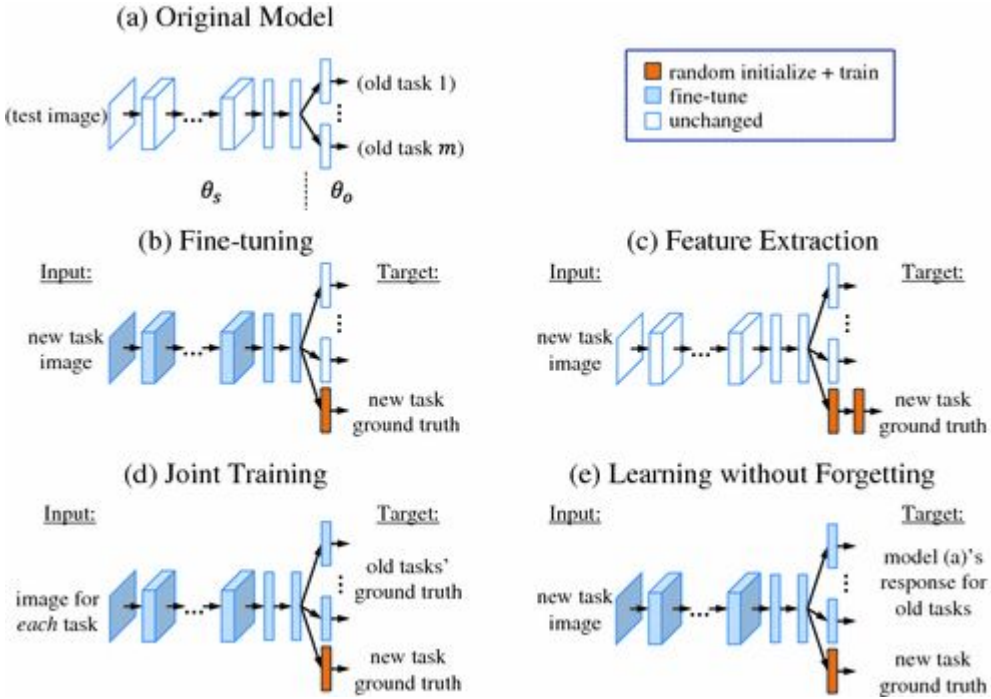
Regularization Based Methods

Data focused: Knowledge distillation from a previous model (trained on a previous task) to the model being trained on the new data

Prior focused: Use an estimated distribution over the model parameters as prior when learning from new data; penalize changes to parameters important for the past tasks



Learning Without Forgetting (LWF)



Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." *IEEE TPAMI* 40.12 (2017): 2935-2947.

Learning Without Forgetting (LWF)

LEARNINGWITHOUTFORGETTING:

Start with:

θ_s: shared parameters

θ_o: task specific parameters for each old task

X_n, Y_n: training data and ground truth on the new task

Initialize:

$Y_o \leftarrow \text{CNN}(X_n, \theta_s, \theta_o)$ // *compute output of old tasks for new data*

$\theta_n \leftarrow \text{RANDINIT}(|\theta_n|)$ // *randomly initialize new parameters*

Train:

Define $\hat{Y}_o \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_o)$ // *old task output*

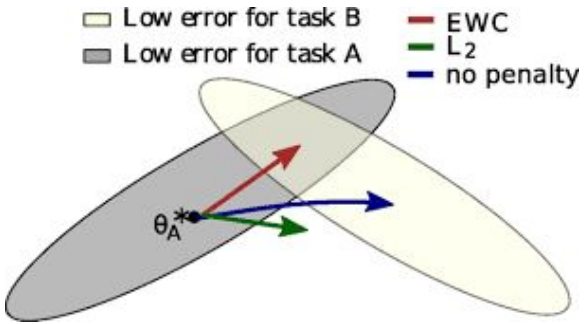
Define $\hat{Y}_n \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_n)$ // *new task output*

$$\theta_s^*,\ \theta_o^*,\ \theta_n^* \leftarrow \underset{\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n}{\text{argmin}} \left(\lambda_o \mathcal{L}_{old}(Y_o, \hat{Y}_o) + \mathcal{L}_{new}(Y_n, \hat{Y}_n) + \mathcal{R}(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n) \right)$$

Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." *IEEE TPAMI* 40.12 (2017): 2935-2947.

Elastic Weight Consolidation (EWC)

- Consider weight importance: Fisher information matrix
- While learning new task constrain the weight to stay in a region of low error for old task

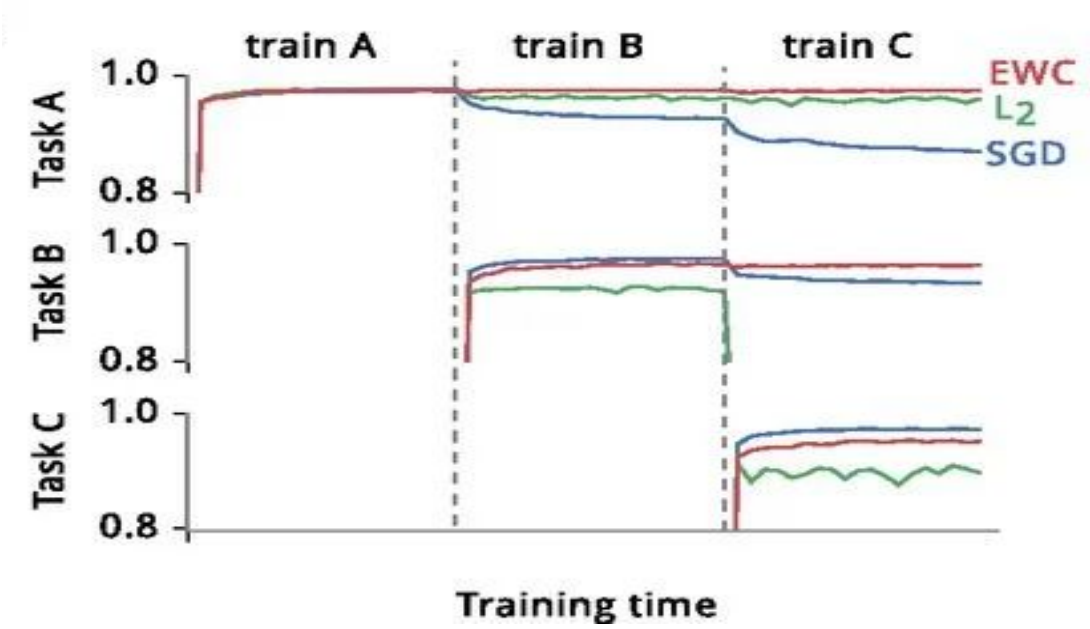


$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2$$

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. PNAS, 114(13), 3521-3526.

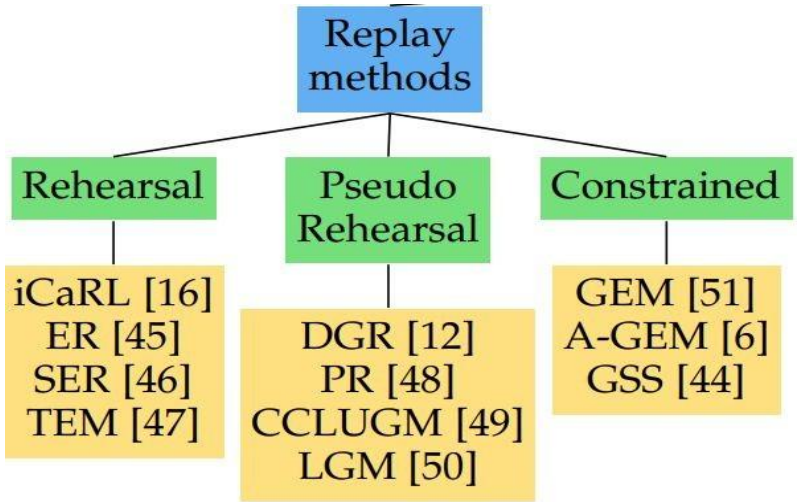
Elastic Weight Consolidation (EWC)

Experiments on permuted MNIST



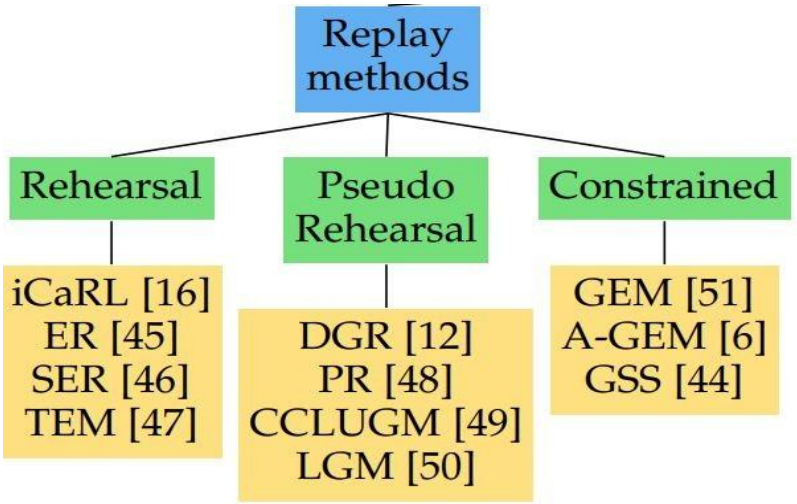
Replay methods

Replay-based methods alleviate catastrophic forgetting by either storing or by artificially generating samples of previous tasks, often referred to as **exemplars.**



Replay methods

- **Rehearsal methods:** retrain current model on a subset of stored samples jointly with new tasks
- **Pseudo rehearsal methods:** feed random input to previous models, use the output as a pseudo-sample. Generative models are also used but add training complexity
- **Constrained optimization:** minimize interference with old tasks by constraining updates on the new task. (e.g. in GEM projects the estimated gradient direction on the feasible region determined by previous task gradients)



iCaRL

- Store a set of training samples (**exemplar**)
- Combine classification loss and distillation loss
- Maintain the size of the exemplar set constant

Algorithm 2 iCaRL INCREMENTALTRAIN

input $X^s, \dots, X^t$ // training examples in per-class sets
input K // memory size
require Θ // current model parameters
require $\mathcal{P} = (P_1, \dots, P_{s-1})$ // current exemplar sets
$\Theta \leftarrow$ UPDATEREPRESENTATION$(X^s, \dots, X^t; \mathcal{P}, \Theta)$
$m \leftarrow K/t$ // number of exemplars per class
for $y = 1, \dots, s-1$ **do**
  $P_y \leftarrow$ REDUCEEXEMPLARSET(P_y, m)
end for
for $y = s, \dots, t$ **do**
  $P_y \leftarrow$ CONSTRUCTEXEMPLARSET(X_y, m, Θ)
end for
$\mathcal{P} \leftarrow (P_1, \dots, P_t)$ // new exemplar sets

Algorithm 3 iCaRL UPDATEREPRESENTATION

input $X^s, \dots, X^t$ // training images of classes $s, \dots, t$
require $\mathcal{P} = (P_1, \dots, P_{s-1})$ // exemplar sets
require Θ // current model parameters
// form combined training set:

$$\mathcal{D} \leftarrow \bigcup_{y=s,\dots,t} \{(x,y) : x \in X^y\} \cup \bigcup_{y=1,\dots,s-1} \{(x,y) : x \in P^y\}$$

// store network outputs with pre-update parameters:
for $y = 1, \dots, s-1$ **do**
  $q_i^y \leftarrow g_y(x_i)$ for all $(x_i, \cdot) \in \mathcal{D}$
end for
run network training (*e.g.* BackProp) with loss function

$$\ell(\Theta) = -\sum_{(x_i,y_i)\in\mathcal{D}} \Big[\sum_{y=s}^{t} \delta_{y=y_i} \log g_y(x_i) + \delta_{y\neq y_i} \log(1-g_y(x_i)) + \sum_{y=1}^{s-1} q_i^y \log g_y(x_i) + (1-q_i^y) \log(1-g_y(x_i)) \Big]$$

that consists of *classification* and *distillation* terms.

Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. CVPR

Gradient Episodic Memory (GEM)

Measure the concept of Backward transfer (BWT) and Forward transfer (FWT)

1. *Backward transfer* (BWT), which is the influence that learning a task t has on the performance on a previous task $k \prec t$. On the one hand, there exists *positive* backward transfer when learning about some task t increases the performance on some preceding task k. On the other hand, there exists *negative* backward transfer when learning about some task t decreases the performance on some preceding task k. Large negative backward transfer is also known as *(catastrophic) forgetting*.
2. *Forward transfer* (FWT), which is the influence that learning a task t has on the performance on a future task $k \succ t$. In particular, *positive* forward transfer is possible when the model is able to perform "zero-shot" learning, perhaps by exploiting the structure available in the task descriptors.

Lopez-Paz, David, and Marc'Aurelio Ranzato. "Gradient episodic memory for continual learning." *NIPS* (2017)

Gradient Episodic Memory (GEM)

- Introduces the notion of **episodic memory** M□ that stores a subset of the observed examples from task t.
- The samples in the memory are used to find the predictors $f(x.k)$

$$l(f_\theta, M_k) = \frac{1}{|M_k|} \Sigma_{(x_i,k,y_i) \in M_k} l(f_\theta(x_i, k), y_i)$$

- Problems:
  - minimizing the loss using only the examples stored in the episodic memory leads to overfitting to those examples only
  - keeping the predictions of the past tasks invariant by distillation makes positive backward transfer impossible.

Lopez-Paz, David, and Marc'Aurelio Ranzato. "Gradient episodic memory for continual learning." *NIPS* 30 (2017): 6467-6476.

GEM

- GEM find the predictor for the current task t by changing the parameters to ensure that the **loss of the current predictor** is less than or equal to **loss of all the previous predictors**:

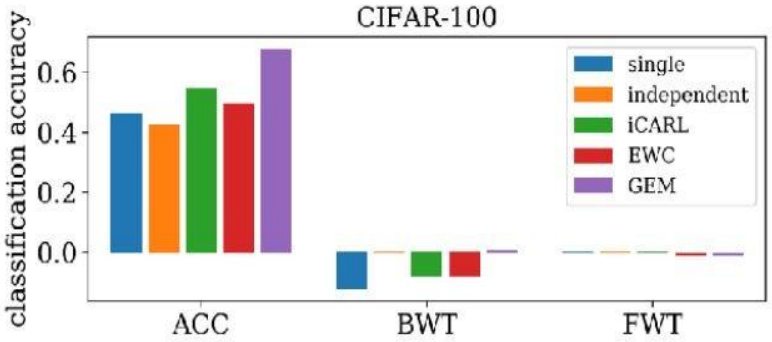
$$\text{minimize}_\theta \; l(f_\theta(x,t),y) \text{ subject to } l(f_\theta, M_k) \leq l(f_\theta^{t-1}, m_k) \text{ for all } k < t$$

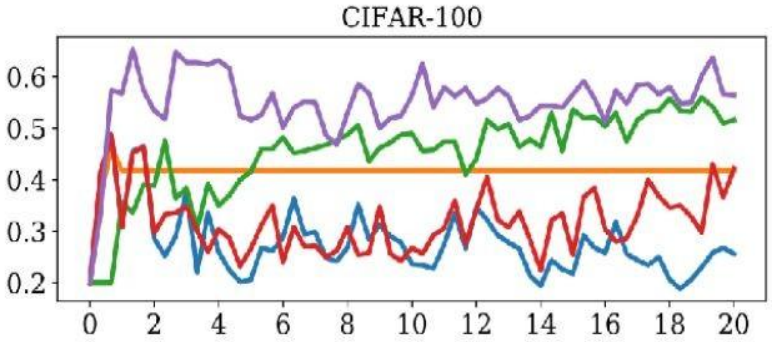
- This can be implemented without storing the old predictors but only reasoning on gradients (see papers for details)
- The problem becomes

$$\text{minimize}_{\tilde{g}} \; \frac{1}{2}||g-\tilde{g}||_2^2 \text{ subject to } \langle \tilde{g}, g_k \rangle \geq 0 \text{ for all } k < t$$

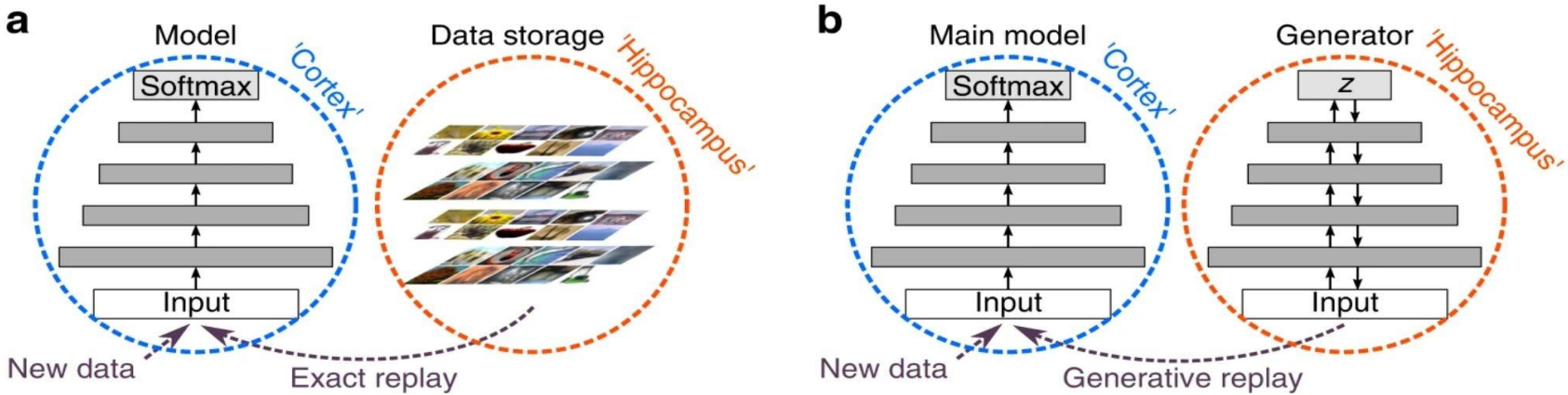
which is a QP problem and can be solved efficiently in the dual

Results


CIFAR-100
classification accuracy
0.6
0.4
0.2
0.0
single
independent
iCARL
EWC
GEM
ACC
BWT
FWT


CIFAR-100
0.6
0.5
0.4
0.3
0.2
0
2
4
6
8
10
12
14
16
18
20

Deep Generative Replay



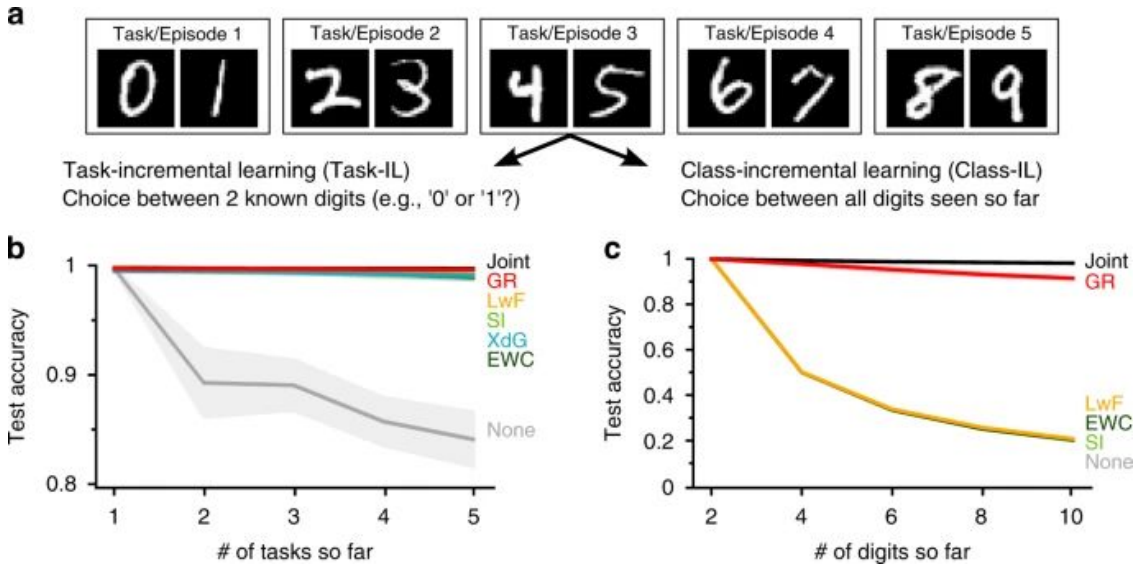
a Exact or experience replay, which views the hippocampus as a memory buffer in which experiences can simply be stored, akin to traditional views of episodic memory[77,78]. **b** Generative replay with a separate generative model, which views the hippocampus as a generative neural network and replay as a generative process[62,79].

SOTA performance on challenging CL benchmarks with many tasks (≥100) or complex inputs (natural images) without storing data

van de Ven et al., Brain-inspired replay for continual learning with artificial neural networks, Nature communications, 2020

Results

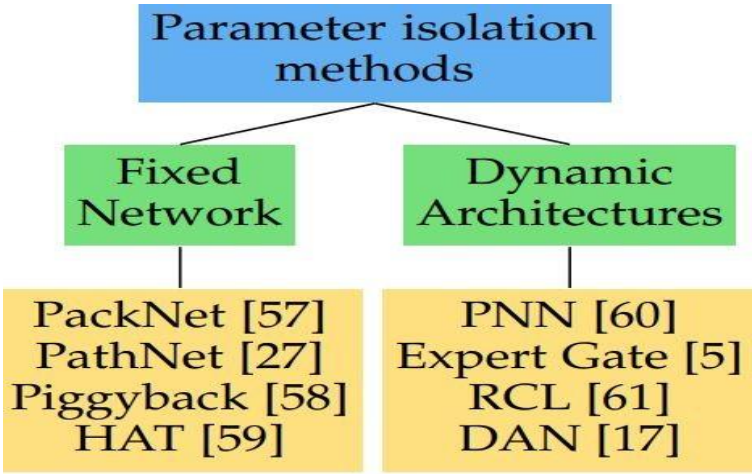
In the task-incremental learning scenario, all CL methods perform very well. In the class-incremental learning scenario, only deep Generative Replay (GR) prevents catastrophic forgetting



Parameter Isolation Methods

Idea: avoid forgetting by using different parameters for each task

Best-suited for: task-incremental setting, unconstrained model capacity, when performance is the priority.



Parameter Isolation Methods

Fixed Network Methods:

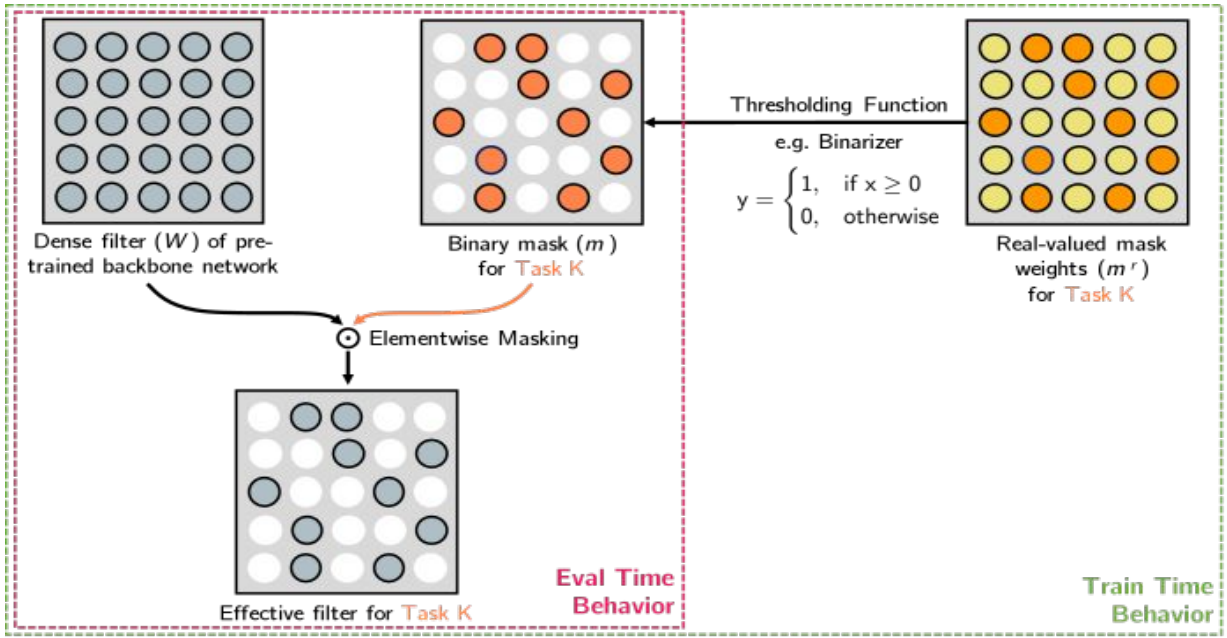
Network parts used for previous tasks are masked out when learning new tasks

Dynamic Architecture Methods

When model size is not constrained: grow new branches for new tasks while freezing previous task parameters or dedicate a model copy to each task

Piggyback

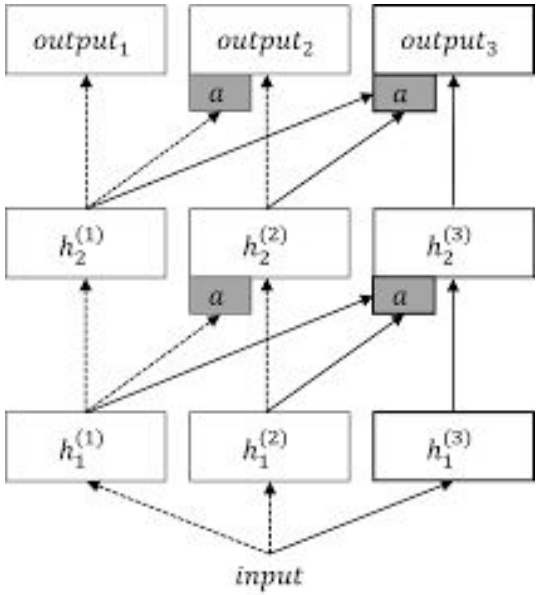
Learn task-specific binary masks



Mallya, A., Davis, D., & Lazebnik, S. (2018). Piggyback: Adapting a single network to multiple tasks by learning to mask weights. ECCV

Progressive NN

Instantiate a NN for each task being solved.
Add lateral connections and adapters.



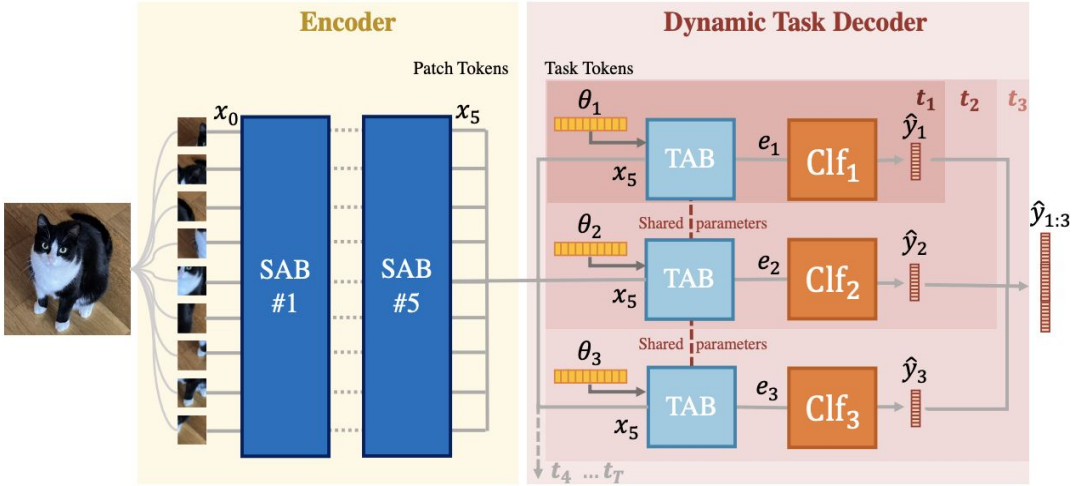
$$h_i^0 = f\left(W_i^0\, h_{i-1}^0\right)$$

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

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., ... & Hadsell, R. (2016). Progressive neural networks. arXiv preprint arXiv:1606.04671.

DYnamic TOken eXpansion (DyTox)

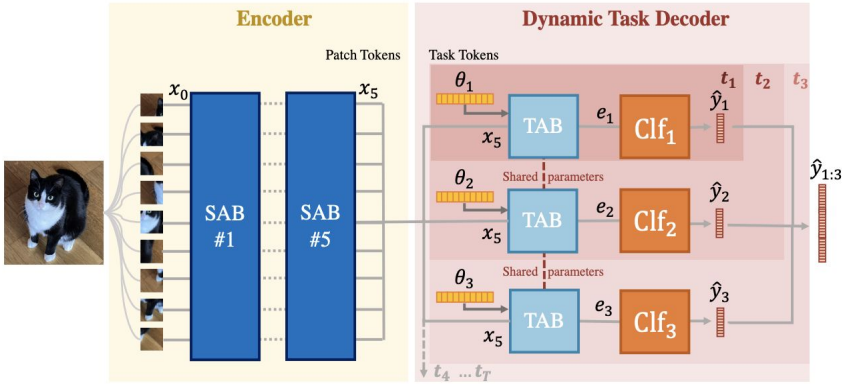
- Encoder and decoder shared among all tasks.
- Dynamic expansion of special tokens permits to specialize to different tasks
- The memory growth of the dynamic network is extremely limited



Douillard, A., Ramé, A., Couairon, G., & Cord, M. Dytox: Transformers for continual learning with dynamic token expansion. CVPR, 2022

DYnamic TOken eXpansion (DyTox)

- An image is split into patches, embedded with a linear projection.
- The patch tokens are processed by five Self-Attention Blocks (SAB).
- For each task, the processed patch tokens are then given to the Task-Attention Block (TAB): each forward through the TAB is modified by a different **task-specialized token**.
- The T final embeddings are given separately to independent classifiers, each predicting their task's classes.



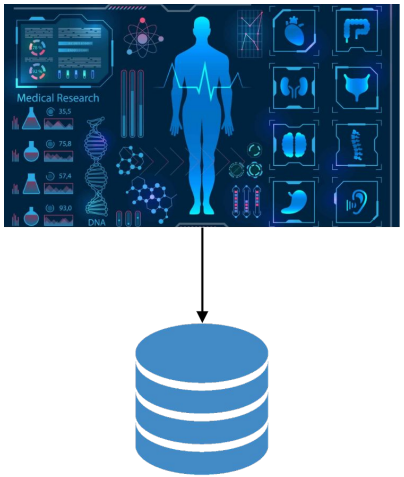
Summary

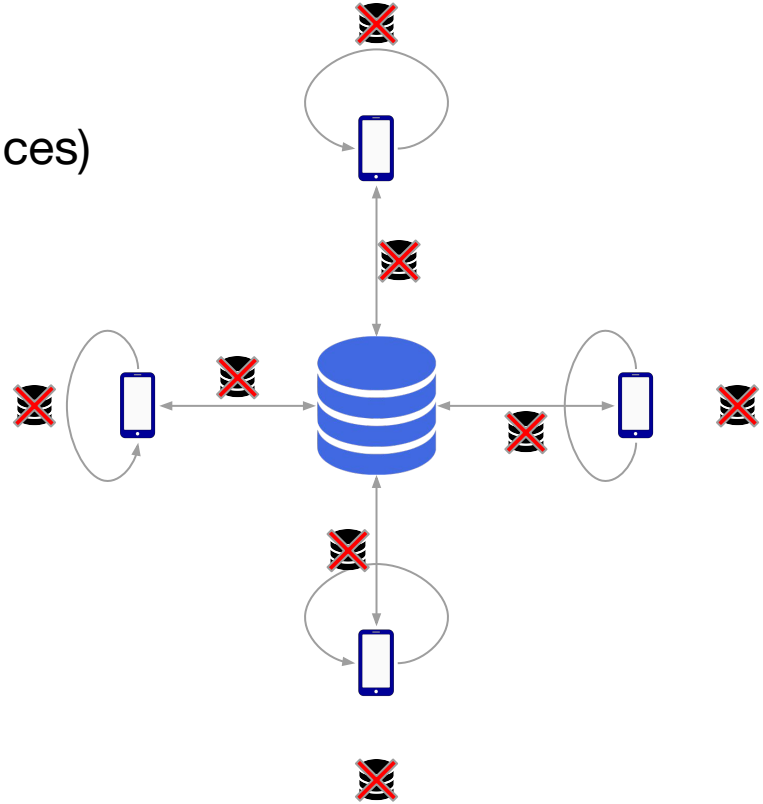
- CL is widely studied.
- Large variety of approaches with different strategies.
- Better performances thanks to state-of-the art architectures.
- Very powerful methods under some (strict) assumptions: only classification, available labeled data, no constraints in terms of memory, etc.

Continual Learning if...

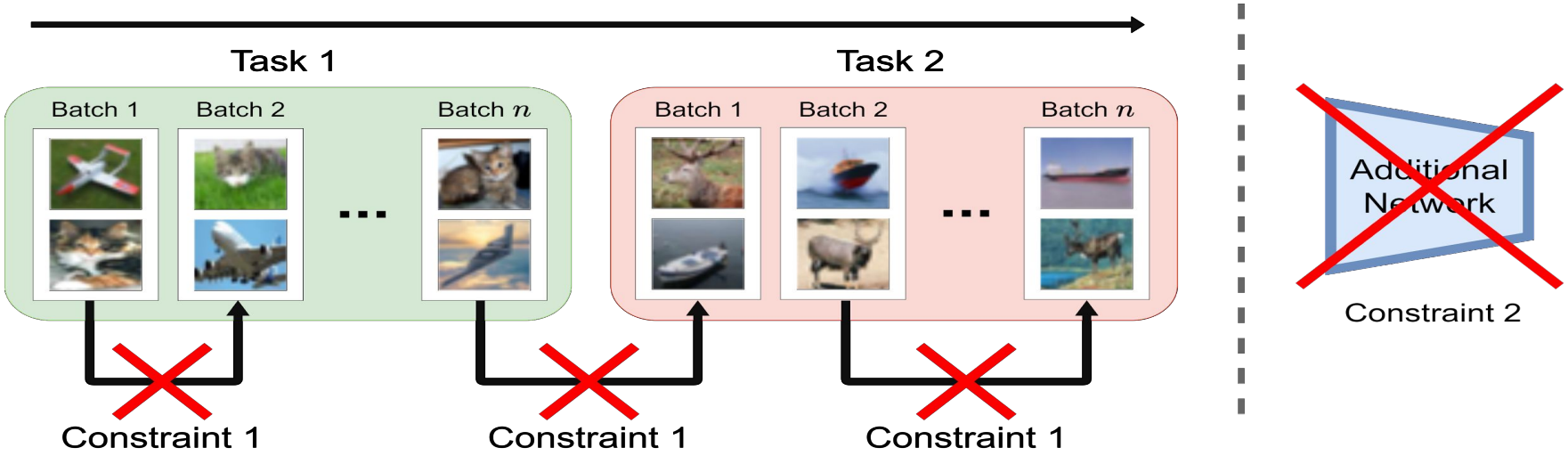
… we have constraints

- Scarcity of Resources (Mobile and IoT Devices)
- Privacy (Medical Applications)





Memory-Constrained Online Continual Learning (MC-OCL)

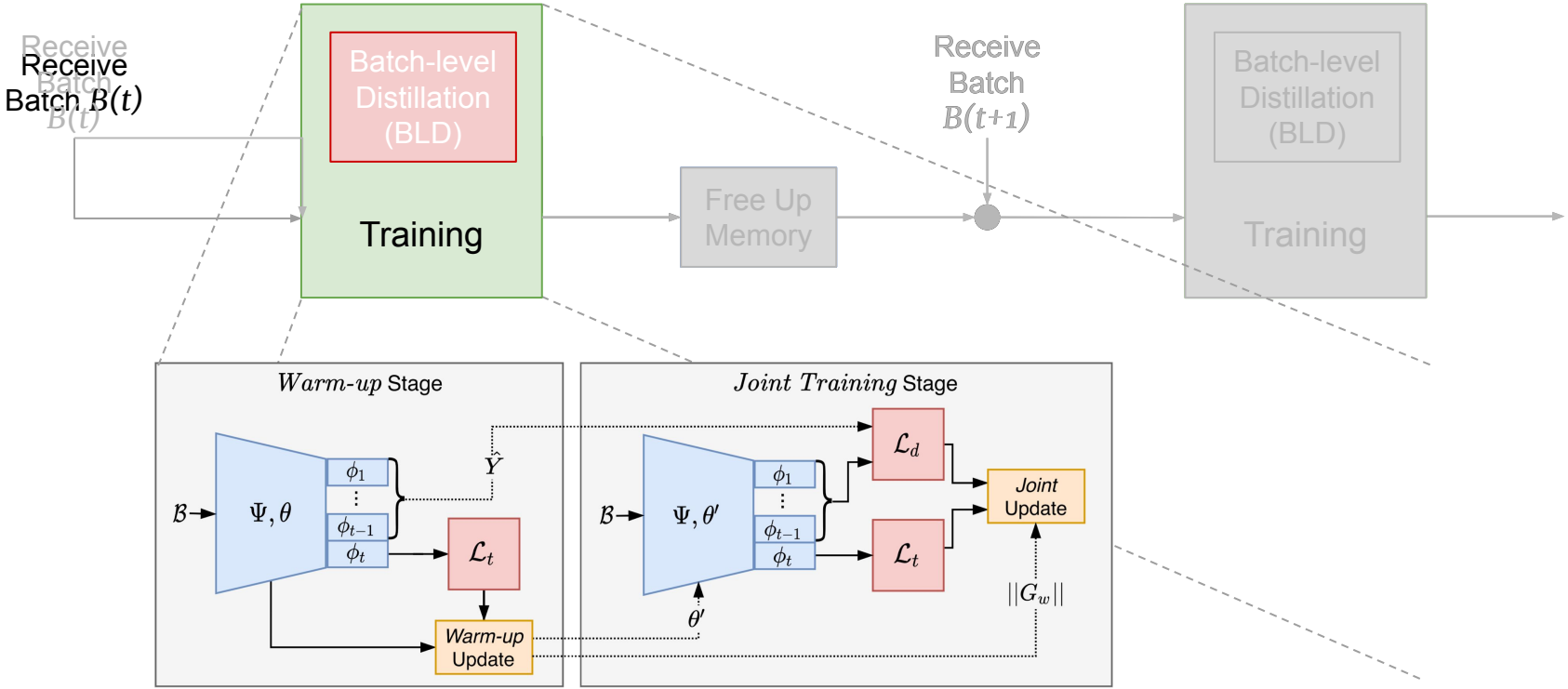


Fini, Enrico, et al. "Online continual learning under extreme memory constraints." *ECCV* 2020.

Memory-Constrained Online Continual Learning (MC-OCL)

- Existing CL solutions are not compatible with this new scenario.
- **Replay-based methods**, need to either store previous training samples (which violates constraint (1)) or train a generator (which is not allowed by constraint (2)).
- **Regularization-based methods** are also not suitable because they need to store Importance weights or output probabilities (violating constraint (1)) or task-specific networks (violating constraint (2)).
- **Parameter Isolation** methods are not compatible because they rely either on retaining masks or network expansions (constraint(2)).

Batch-level Distillation (BLD)



$$G_w = \frac{\partial \mathcal{L}_t}{\partial \theta}$$

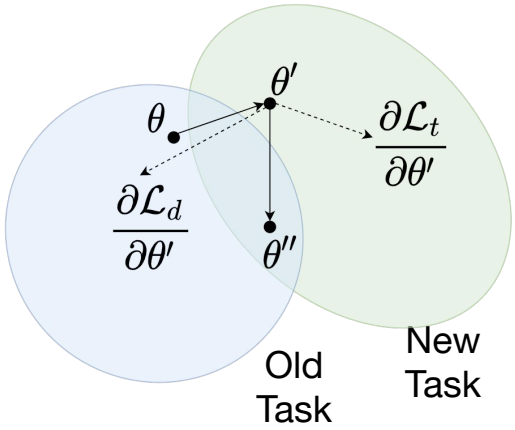
$$\theta' = \theta - \alpha_w G_w$$

$$G_j = \lambda \frac{||G_w||}{||\frac{\partial \mathcal{L}_d}{\partial \theta'}||} \frac{\partial \mathcal{L}_d}{\partial \theta'} + \frac{\partial \mathcal{L}_t}{\partial \theta'}$$

$$\theta'' = \theta' - \alpha_j G_j$$

Batch-level Distillation (BLD)

- When a batch of data of the new task is received the parameters minimizes the loss on the old task.
- In the **warm up stage** an unconstrained step towards the new task is taken, which results in a decrease of accuracy on the old task. However, using distillation we can estimate the direction in parameter space that will lead back to the old task.
- The gradient is then scaled according to the **magnitude of the warm-up gradient**
- The two objectives are optimized jointly, causing the parameters to land in a configuration that yields good performance on both tasks.



$$G_w = \frac{\partial \mathcal{L}_t}{\partial \theta}$$

$$\theta' = \theta - \alpha_w G_w$$

Warm-up Stage

$$G_j = \lambda \frac{\|G_w\|}{\| \frac{\partial \mathcal{L}_d}{\partial \theta'} \|} \frac{\partial \mathcal{L}_d}{\partial \theta'} + \frac{\partial \mathcal{L}_t}{\partial \theta'}$$

$$\theta'' = \theta' - \alpha_j G_j$$

Joint Training Stage

Results

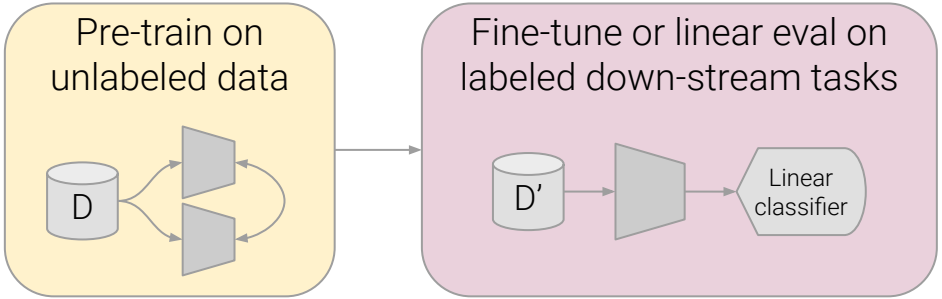
Method		CIFAR10						Memory Overhead		
		T0	T1	T2	T3	T4	Avg.	Intra-batch	Inter-batch	Data Storage
MC-OCL	*Finetune*	59.6	58.2	66.8	80.2	97.0	72.3	-	-	-
	L2	75.5	65.3	73.5	81.3	96.8	78.5	44.8MB	-	-
	BLD	83.4	83.2	79.5	88.1	97.0	**86.2**	32kB	-	-
Single-pass	*LwF*	81.2	83.6	81.1	88.5	96.5	86.2	320kB	320kB	36.8MB
Offline	*LwF*	93.8	94.1	91.6	96.2	98.3	94.8	320kB	320kB	36.8MB

Continual Learning if....

	Why?
...we have **no labels**	Availability of labels when learning online or sequentially is quite unlikely
... we focus on **learning good visual representations**	Training a linear classifier or even fine-tuning a pre-trained feature extractor on a down-stream task takes literally seconds with modern hardware

Self-Supervised Learning (SSL)

Self-Supervised Learning exploits the intrinsic structure of the data to pretrain strong feature extractors



The pre-training objective is to find parameters such as:

$$\underset{\theta}{\operatorname{argmin}}\ \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathcal{L}_{SSL}\left(\boldsymbol{z}^{A},\boldsymbol{z}^{B}\right)\right]$$

SSL Methods

Contrastive-based

instance discrimination, invariance to augmentations + negatives

Examples: SimCLR, MoCo V2+, …
Loss: Contrastive

$$-\log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}_i^A, \boldsymbol{z}_i^B\right)/\tau\right)}{\sum_{\boldsymbol{z}_j \in \eta(i)} \exp\left(\text{sim}\left(\boldsymbol{z}_i^A, \boldsymbol{z}_j\right)/\tau\right)}$$

Consistency-based

no negatives, only invariance to augmentations

Examples: BYOL, SimSiam
Loss: MSE

$$-||\boldsymbol{q}^A - \boldsymbol{z}^B||_2^2$$

Decorrelation-based

invariance to augmentations and redundancy reduction

Examples: BarlowTwins, VICReg, W-MSE
Loss: Cross-correlation

$$\sum_u \left(1 - \mathcal{C}_{uv}\right)^2 + \lambda \sum_u \sum_{v \neq u} \mathcal{C}_{uv}^2$$

Clustering-based

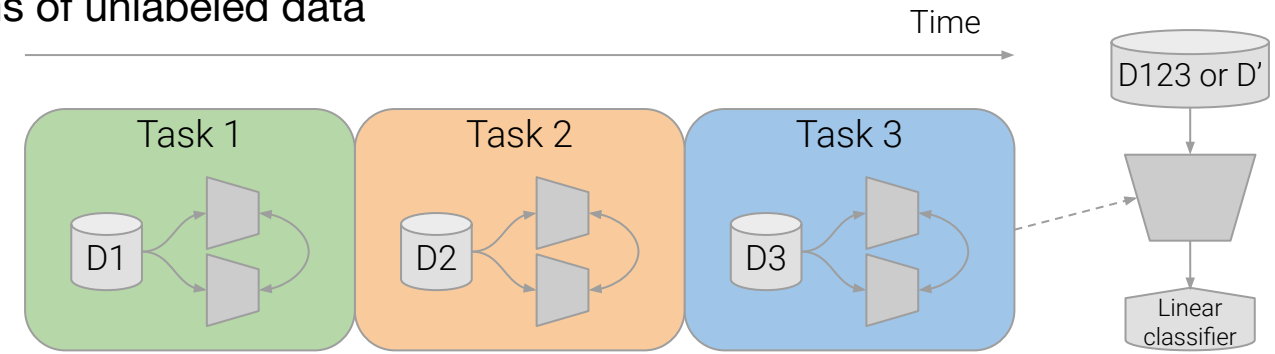
Predicted cluster assignments are invariant to augmentations

Examples: DeepCluster V2, SwAV, DINO
Loss: Cross-entropy (with pseudo-labels)

$$-\sum_d \boldsymbol{a}_d^B \log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}^A, \boldsymbol{c}_d\right)/\tau\right)}{\sum_k \exp\left(\text{sim}\left(\boldsymbol{z}^A, \boldsymbol{c}_k\right)/\tau\right)}$$

Continual Self-Supervised Learning (CSSL)

Continual Self-Supervised Learning is the problem of learning strong feature extractors from streams of unlabeled data



The continual pre-training objective is to find parameters such as:

$$\underset{\theta}{\operatorname{argmin}} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}_t} \left[\mathcal{L}_{SSL} \left(\boldsymbol{z}^A, \boldsymbol{z}^B \right) \right]$$

Continual Self-Supervised Learning (CSSL)

- SSL benefits from longer training, the evolution of representations should not be overly constrained
- SSL methods exhibit different losses and feature normalizations that interfere with CL regularization losses and vice-versa

Methods	Loss	Equation
SimCLR MoCo NNCLR	InfoNCE	$-\log \frac{\exp(\text{sim}(\boldsymbol{z}_i^A, \boldsymbol{z}_i^B)/\tau)}{\sum_{\boldsymbol{z}_j \in \eta(i)} \exp(\text{sim}(\boldsymbol{z}_i^A, \boldsymbol{z}_j)/\tau)}$
BYOL SimSiam VICReg	MSE	$-\|\|\boldsymbol{q}^A - \boldsymbol{z}^B\|\|_2^2$
SwAV DCV2 DINO	Cross-entropy	$-\sum_d \boldsymbol{a}_d^B \log \frac{\exp(\text{sim}(\boldsymbol{z}^A, \boldsymbol{c}_d)/\tau)}{\sum_k \exp(\text{sim}(\boldsymbol{z}^A, \boldsymbol{c}_k)/\tau)}$
Barlow Twins VICReg	Cross-correlation	$\sum_u (1 - \mathcal{C}_{uv})^2 + \lambda \sum_u \sum_{v \neq u} \mathcal{C}_{uv}^2$

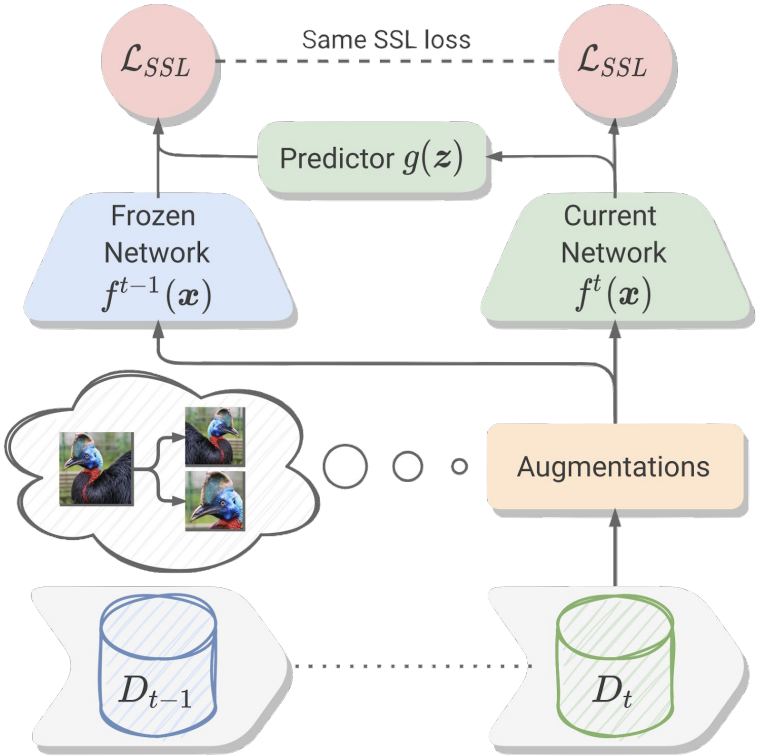
CaSSLe

Basically, two simple ideas:

- a predictor network that maps the current state of the representations to their past state
- a family of adaptable distillation losses inherited from the SSL literature

… and an important key insight: use the same loss for distillation and representation learning!

$$\mathcal{L} = \mathcal{L}_{SSL}(z^A, z^B) + \mathcal{L}_D(z^A, \bar{z}^A)$$
$$= \mathcal{L}_{SSL}(z^A, z^B) + \mathcal{L}_{SSL}(g(z^A), \bar{z}^A)$$



Fini, Enrico, et al. "Self-supervised models are continual learners." *CVPR* 2022

Experiments

We train six SSL models:

- Barlow Twins
- SwAV
- BYOL
- VICReg
- MoCoV2+
- SimCLR

We evaluate three CL settings:

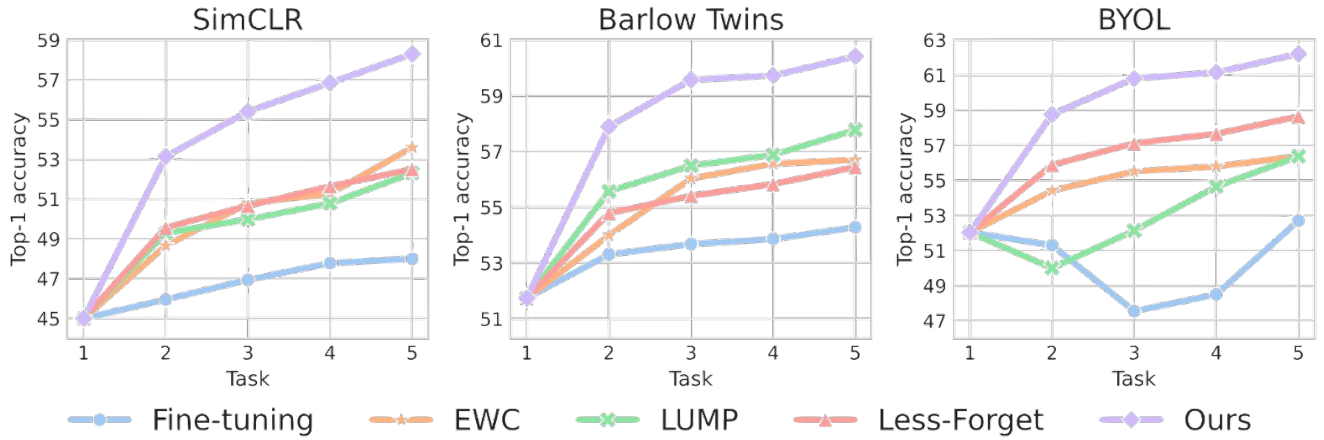
- **Class-incremental:** each task contains a new set of classes
- **Data-incremental:** each task contains new samples of the same classes
- **Domain-incremental:** each task contains new domain

On three widely used datasets:

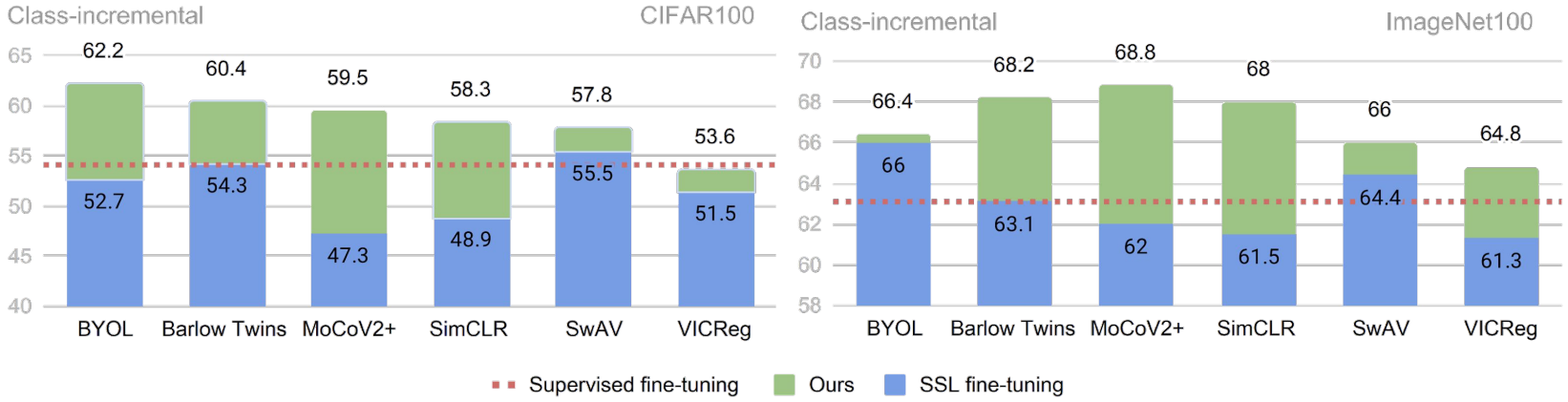
- CIFAR100
- ImageNet100
- DomainNet

Results

Comparison with CL methods – CIFAR100



Results


Class-incremental
CIFAR100
65
60
55
50
45
40
62.2
60.4
59.5
58.3
57.8
53.6
52.7
54.3
47.3
48.9
55.5
51.5
BYOL
Barlow Twins
MoCoV2+
SimCLR
SwAV
VICReg
Class-incremental
ImageNet100
70
68
66
64
62
60
58
66.4
68.2
68.8
68
66
64.8
66
63.1
62
61.5
64.4
61.3
BYOL
Barlow Twins
MoCoV2+
SimCLR
SwAV
VICReg
Supervised fine-tuning
Ours
SSL fine-tuning

Results

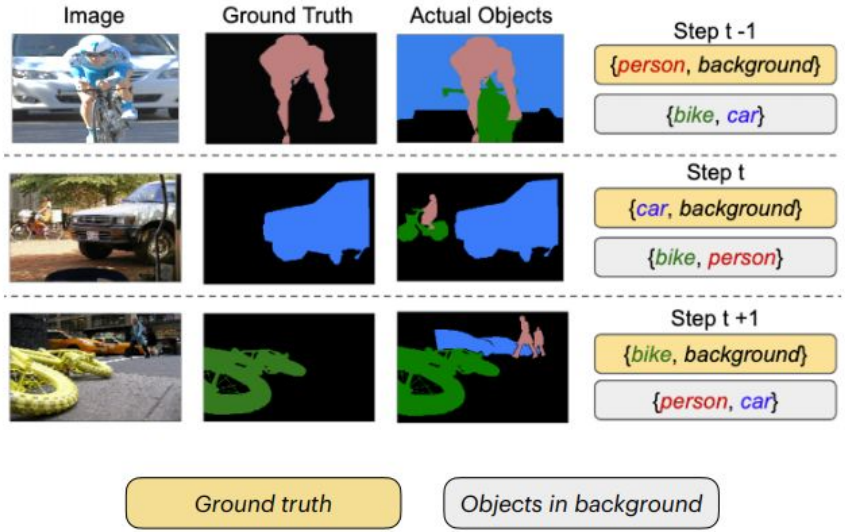
Top-1 accuracy on DomainNet: 6 tasks, domain-incremental setting

Method	Strategy	Real		Quickdraw		Painting		Sketch		Infograph		Clipart		Avg.	
		Aw.	Ag.	Aw.	Ag.	Aw.	Ag.	Aw.	Ag.	Aw.	Ag.	Aw.	Ag.	Aw.	Ag.
Barlow Twins	Finetuning	56.3	50.9	54.1	45.8	42.7	35.9	49.0	41.9	22.0	17.4	59.0	52.5	50.3	43.7
	CaSSLe	**62.7**	**57.1**	**59.1**	**50.6**	**49.2**	**42.1**	**53.8**	**47.7**	**25.5**	**20.6**	**61.9**	**55.6**	**55.5**	**48.9**
	Offline	67.1	63.0	60.3	53.9	52.4	46.3	51.9	46.9	25.9	21.0	58.8	52.6	57.2	51.8
SwAV	Finetuning	57.7	52.3	53.2	43.5	43.0	35.9	46.1	39.0	21.6	16.5	53.4	46.6	49.6	42.5
	CaSSLe	**62.8**	**57.8**	**59.5**	**50.2**	**47.5**	**41.2**	**49.5**	**42.5**	**22.5**	**17.9**	**56.5**	**49.6**	**54.3**	**47.5**
	Offline	64.1	59.5	60.6	53.6	47.6	42.9	47.7	42.1	23.3	18.9	53.6	47.3	54.6	49.1
BYOL	Finetuning	58.7	53.2	51.7	41.6	44.0	37.4	49.6	43.9	23.5	19.0	58.6	53.5	50.6	43.8
	CaSSLe	**63.7**	**60.5**	**59.3**	**50.9**	**48.6**	**44.1**	**50.4**	**45.2**	**24.1**	**19.4**	**59.0**	**54.4**	**55.1**	**49.7**
	Offline	67.2	64.0	60.2	53.3	51.5	47.3	50.4	46.2	24.5	20.8	57.0	51.5	56.6	51.9
VICReg	Finetuning	54.7	49.6	53.0	44.9	42.1	34.7	49.0	41.9	21.1	16.4	58.5	52.6	49.3	42.8
	CaSSLe	**59.0**	**53.2**	**56.4**	**47.8**	**46.0**	**38.9**	**52.3**	**45.6**	**23.9**	**18.5**	**60.9**	**55.3**	**52.9**	**46.1**
	Offline	66.4	62.7	59.2	53.5	52.4	47.2	53.2	48.1	25.3	20.7	58.3	53.2	56.7	51.9
SimCLR	Finetuning	52.5	47.6	48.2	38.1	37.5	31.7	42.8	35.7	18.8	14.4	50.9	46.8	45.1	38.4
	CaSSLe	**58.4**	**43.4**	**54.2**	**44.7**	**43.9**	**37.7**	**47.6**	**41.9**	**22.0**	**17.8**	**54.9**	**50.5**	**50.0**	**44.2**
	Offline	62.1	59.5	58.3	52.9	46.1	42.5	45.6	41.3	22.1	18.8	51.0	45.9	52.6	48.6
MoCoV2+	Finetuning	50.9	45.5	45.8	37.5	36.0	29.3	39.5	32.1	17.9	13.5	50.3	**44.5**	43.2	36.7
	CaSSLe	**56.0**	**50.3**	**48.7**	**40.0**	**40.4**	**33.6**	**42.0**	**35.0**	**19.9**	**15.2**	**51.7**	**44.5**	**46.7**	**38.8**
	Offline	65.2	61.3	57.9	51.3	48.7	43.1	44.7	39.1	23.4	19.0	51.3	44.8	53.7	48.4
Supervised Contrastive	Finetuning	57.7	52.6	55.3	45.5	44.9	38.0	51.7	45.0	22.6	18.3	64.0	60.0	52.1	45.4
	CaSSLe	**63.4**	**58.8**	**59.7**	**51.3**	**50.1**	**44.7**	**55.9**	**50.3**	**26.9**	**22.4**	**65.0**	**61.3**	**56.7**	**50.9**
	Offline	67.4	65.3	65.8	63.0	53.6	50.9	56.0	53.1	28.0	25.7	62.8	59.6	60.0	57.4
Supervised	Finetuning	63.0	58.2	56.9	47.6	49.1	44.0	55.7	50.3	27.7	23.3	68.6	63.5	55.9	49.8
	Offline	74.7	73.2	68.5	67.8	62.0	59.3	65.7	63.7	33.7	34.5	72.3	69.3	66.4	65.0

Beyond Image Recognition

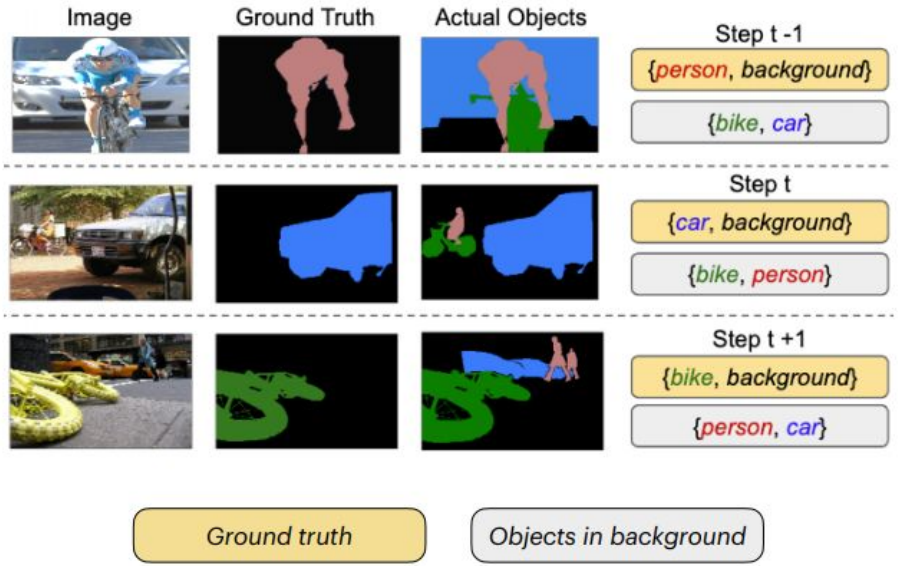
Beyond Recognition: CL for Semantic Segmentation

- Goal: Update the segmentation network as new classes are made available. Every step we consider pixels associated to a restricted set of classes and annotations are given.
- All the other pixels are considered background.



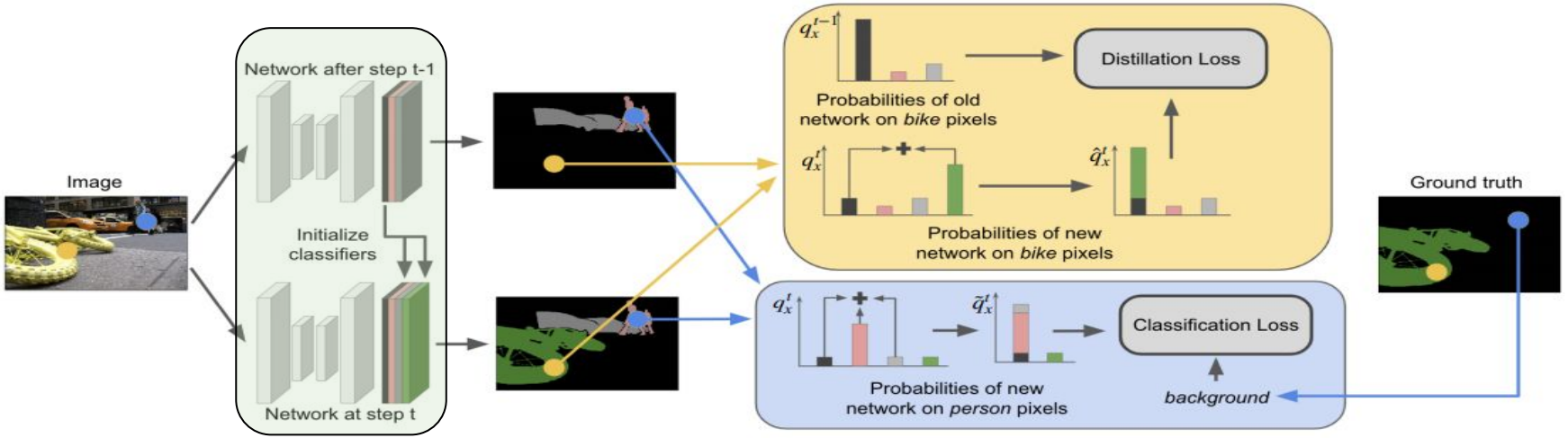
Background Shift

The BG semantic changes at every step, exacerbating catastrophic forgetting.

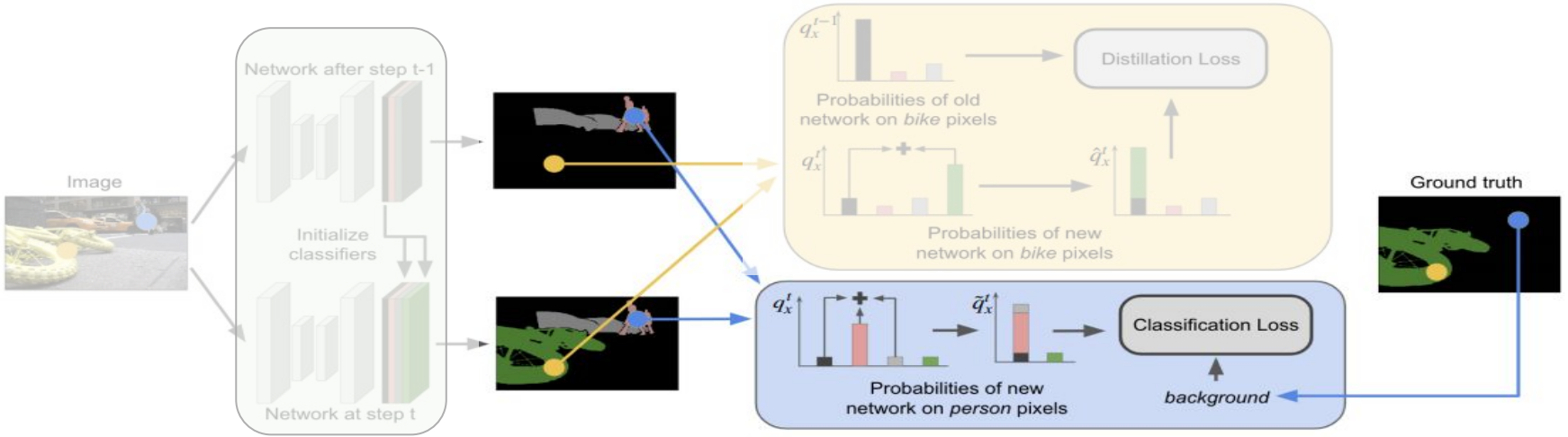


Cermelli, Fabio, et al. "Modeling the background for incremental learning in semantic segmentation." *CVPR*. 2020.

Modelling the Background (MiB)



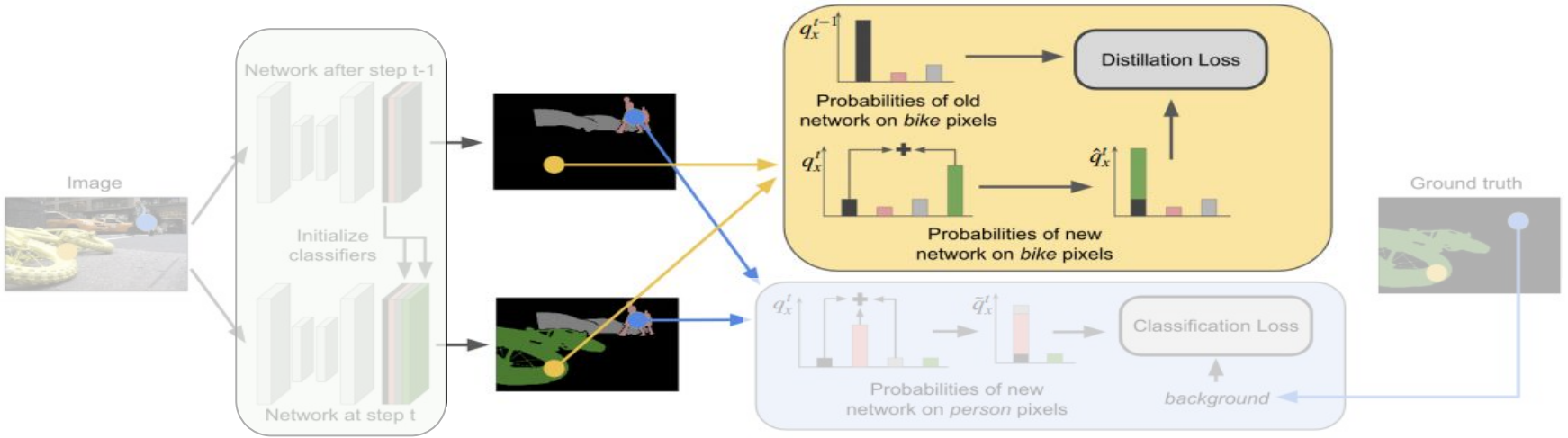
Modelling the Background (MiB)



Classification loss modeling the presence of classes seen in previous learning step in the background.

$$\ell_{ce}^{\theta^t}(x,y) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \log \tilde{q}_x^t(i,y_i) \qquad \tilde{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq \texttt{b} \\ \sum_{k\in\mathcal{Y}^{t-1}} q_x^t(i,k) & \text{if } c = \texttt{b} \end{cases}$$

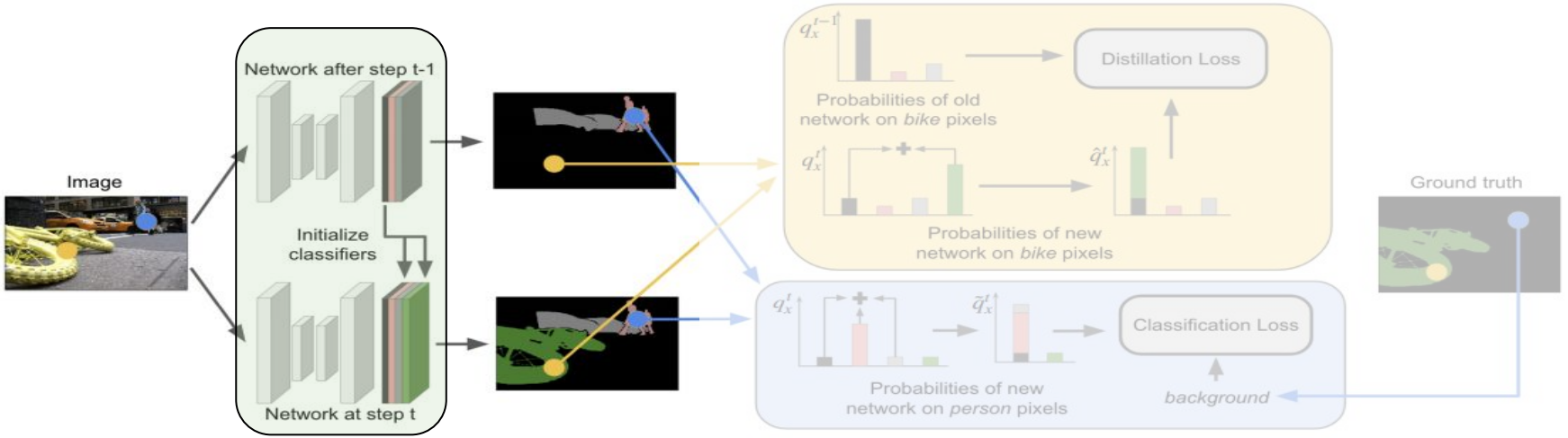
Modelling the Background (MiB)



Distillation loss: to account that in previous steps the BG might have contained the classes which are currentlv learned.

$$\ell_{kd}^{\theta^t}(x,y) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{Y}^{t-1}} q_x^{t-1}(i,c)\log\hat{q}_x^t(i,c) \qquad \hat{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq \mathtt{b} \\ \sum_{k\in\mathcal{C}^t} q_x^t(i,k) & \text{if } c = \mathtt{b} \end{cases}$$

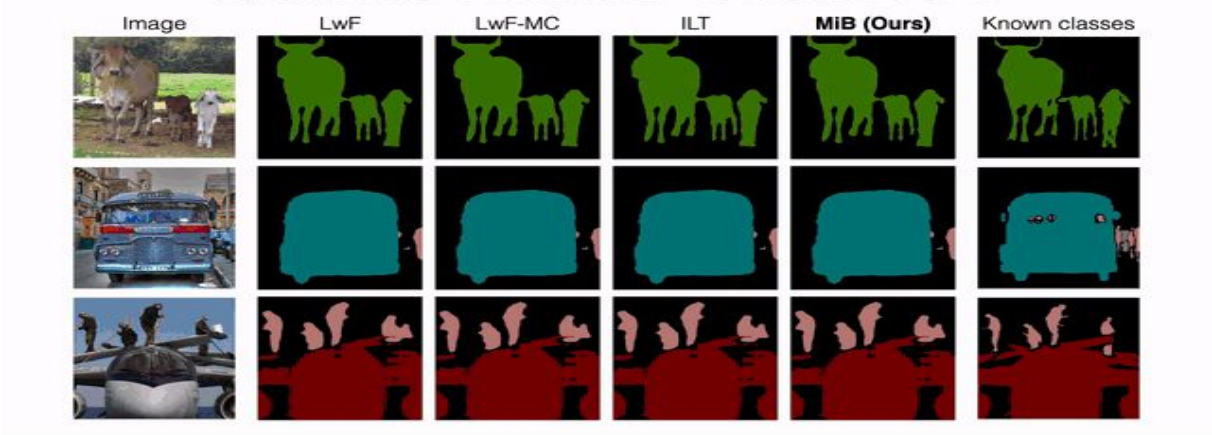
Modelling the Background (MiB)



Initialization: compensate for the fact that novel classes at a given step will likely be classified as BG.

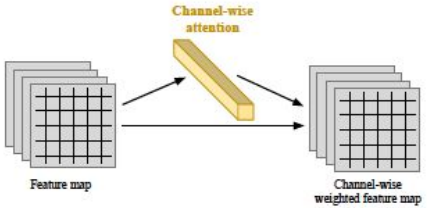
Results - Pascal VOC2012

	19-1						15-5						15-1					
	Disjoint			Overlapped			Disjoint			Overlapped			Disjoint			Overlapped		
Method	*1-19*	*20*	*all*	*1-19*	*20*	*all*	*1-15*	*16-20*	*all*	*1-15*	*16-20*	*all*	*1-15*	*16-20*	*all*	*1-15*	*16-20*	*all*
FT	5.8	12.3	6.2	6.8	12.9	7.1	1.1	33.6	9.2	2.1	33.1	9.8	0.2	1.8	0.6	0.2	1.8	0.6
PI [2]	5.4	14.1	5.9	7.5	14.0	7.8	1.3	34.1	9.5	1.6	33.3	9.5	0.0	1.8	0.4	0.0	1.8	0.5
EWC [3]	23.2	16.0	22.9	26.9	14.0	26.3	26.7	37.7	29.4	24.3	35.5	27.1	0.3	4.3	1.3	0.3	4.3	1.3
RW [4]	19.4	15.7	19.2	23.3	14.2	22.9	17.9	36.9	22.7	16.6	34.9	21.2	0.2	5.4	1.5	0.0	5.2	1.3
LwF [1]	53.0	9.1	50.8	51.2	8.5	49.1	58.4	37.4	53.1	58.9	36.6	53.3	0.8	3.6	1.5	1.0	3.9	1.8
LwF-MC [5]	63.0	13.2	60.5	64.4	13.3	61.9	67.2	41.2	60.7	58.1	35.0	52.3	4.5	7.0	5.2	6.4	8.4	6.9
ILT [6]	69.1	16.4	66.4	67.1	12.3	64.4	63.2	39.5	57.3	66.3	40.6	59.9	3.7	5.7	4.2	4.9	7.8	5.7
MiB	**69.6**	**25.6**	**67.4**	**70.2**	**22.1**	**67.8**	**71.8**	**43.3**	**64.7**	**75.5**	**49.4**	**69.0**	**46.2**	**12.9**	**37.9**	**35.1**	**13.5**	**29.7**
Joint	77.4	78.0	77.4	77.4	78.0	77.4	79.1	72.6	77.4	79.1	72.6	77.4	79.1	72.6	77.4	79.1	72.6	77.4

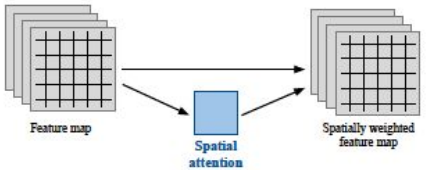


An Aside: Spatial and Channel Attention

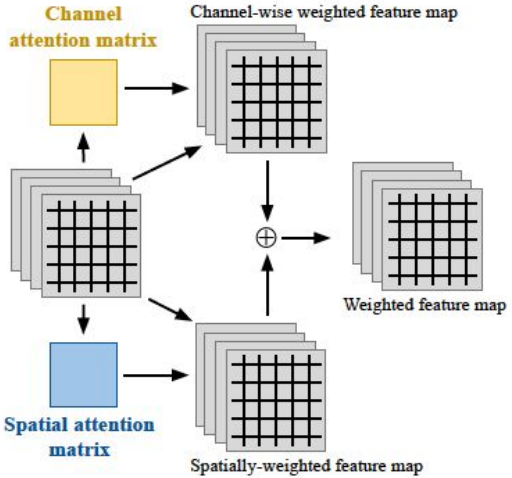
- Attention mechanisms widely popular in semantic segmentation.
- Usually, attention is used to exploit long range dependencies between pixels.



(a) Channel-wise attention.



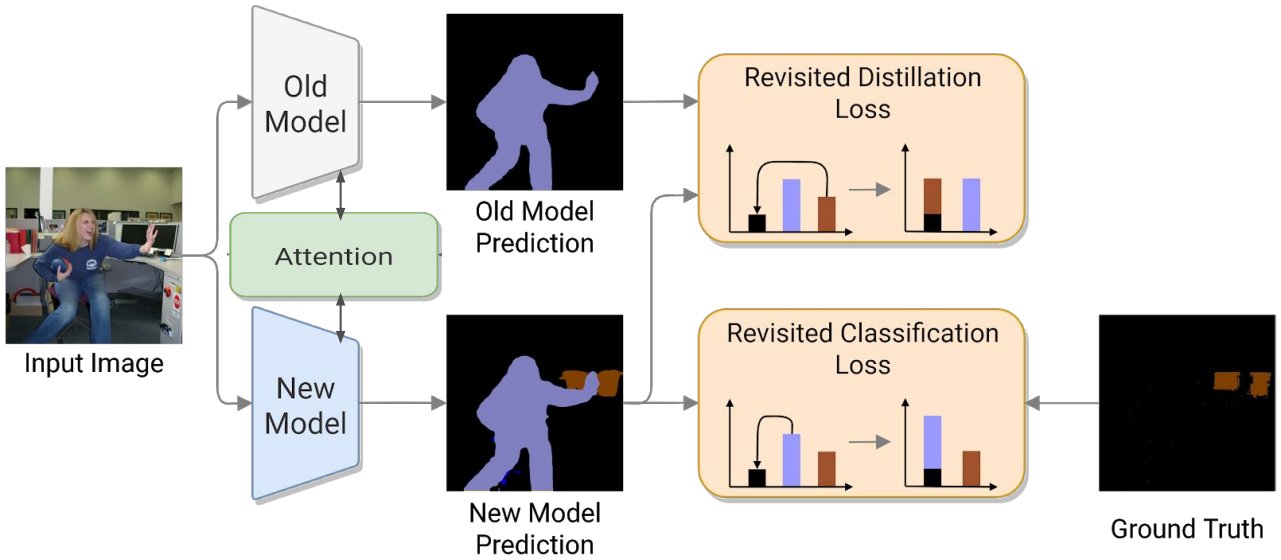
(b) Spatial attention.



(c) Separate spatial and channel attention.

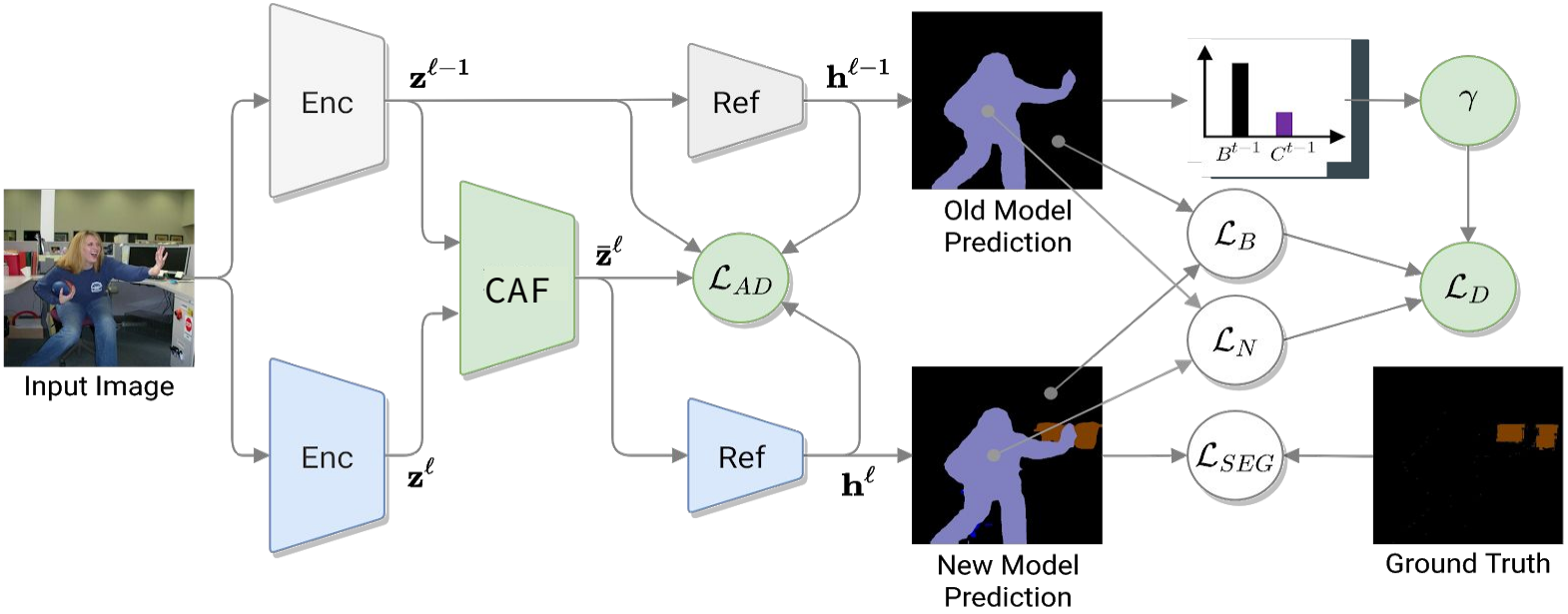
What about Attention?

Attention to model relationships between networks trained on **different tasks**

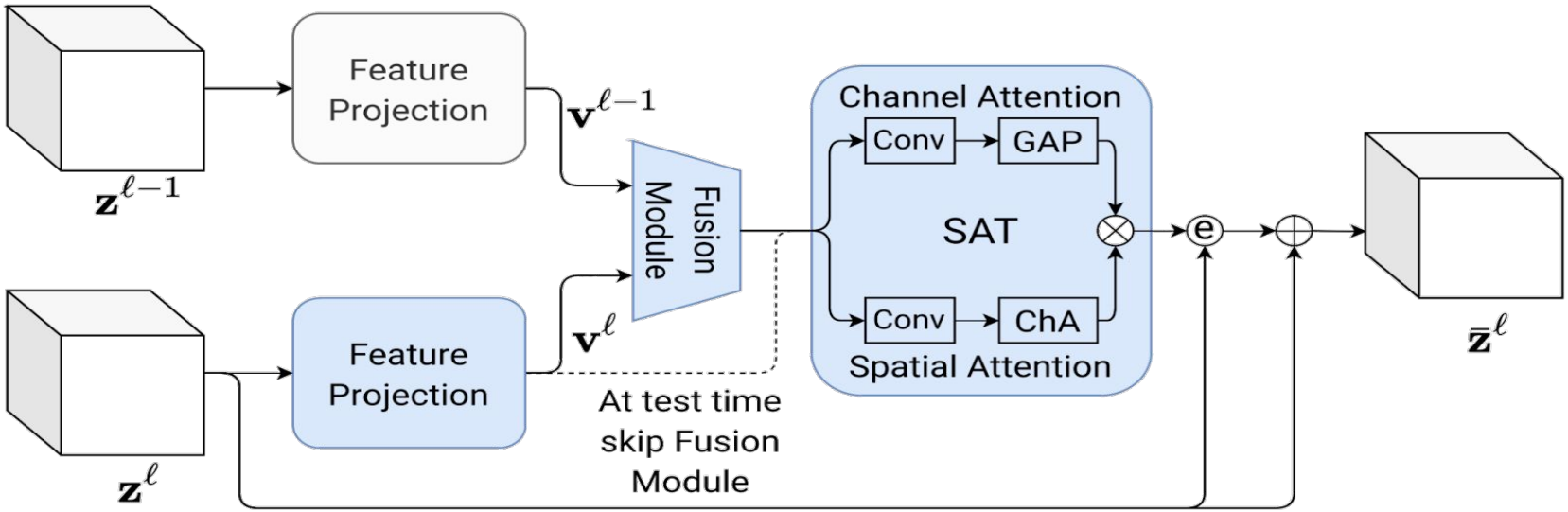


Yang, G., Fini, E., Xu, D., Rota, P., Ding, M., Hao, T., ... & Ricci, E. (2022). Continual Attentive Fusion for Incremental Learning in Semantic Segmentation. IEEE Transactions on Multimedia.

Continual Attentive Fusion

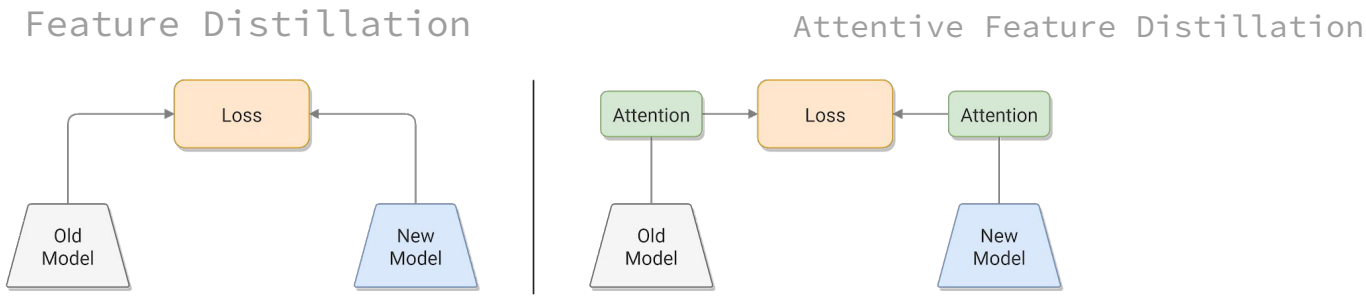


CAF Module


$\mathbf{z}^{\ell-1}$
Feature Projection
$\mathbf{v}^{\ell-1}$
Fusion Module
Channel Attention
Conv
GAP
SAT
Conv
ChA
Spatial Attention
$\bar{\mathbf{z}}^{\ell}$
$\mathbf{z}^{\ell}$
Feature Projection
$\mathbf{v}^{\ell}$
At test time skip Fusion Module

Attentive Feature Distillation

- Previous works:
  - **Distill output probabilities**
  - All channels and spatial locations are **equally weighted**
- We propose to **distill intermediate features** using **attentive feature distillation**



Attentive Feature Distillation

We use both channel and spatial attention:

$$\mathrm{AD}_{\mathrm{CH}}\left(\mathbf{m}\right) = \psi\left(\omega_2^{\mathbf{m}} * \sigma\left(\omega_1^{\mathbf{m}} * \mathrm{AvgPool}(\mathbf{m})\right)\right)$$

$$\mathrm{AD}_{\mathrm{SP}}\left(\mathbf{m}\right) = \frac{\sum_{j=1}^{C} \mathbf{m}_j^2}{\left\| \sum_{j=1}^{C} \mathbf{m}_j^2 \right\|_{\mathrm{F}}}$$

The two contributions are merged together via tensor product:

$$\mathrm{AD}\left(\mathbf{m}\right) = \left(\mathrm{AD}_{\mathrm{CH}}\left(\mathbf{m}\right) \otimes \mathrm{AD}_{\mathrm{SP}}\left(\mathbf{m}\right) + 1\right)\ \mathbf{m},$$

Attentive Feature Distillation loss:

$$\mathcal{L}_{\mathrm{AD}} = \|\mathrm{AD}(\bar{\mathbf{z}}^{\ell}) - \mathrm{AD}(\mathbf{z}^{\ell-1})\|_{\mathrm{F}}^2 + \| \mathrm{AD}(\mathbf{h}^{\ell}) - \mathrm{AD}(\mathbf{h}^{\ell-1})\|_{\mathrm{F}}^2$$

Balanced Knowledge Distillation

- Knowledge Distillation is imbalanced: the background is usually the most represented class by far
- We propose a Balanced Knowledge Distillation:

$$\mathcal{L}_{\mathrm{D}} = \gamma \mathcal{L}_{\mathrm{B}} + \mathcal{L}_{\mathrm{N}}$$

Background

Non-Background

- Where the balancing parameter only depends on the inferred ratio of background vs. non-background pixels:

$$\gamma = \frac{\sum_{s \in \mathcal{S}_{t-1} / \{\mathrm{B}_{\ell-1}\}} \mathrm{Softmax}\left(\mathrm{AvgPool}\left(\phi_{\omega}^{\ell-1}(\mathbf{x})[s]\right)\right)}{\mathrm{Softmax}\left(\mathrm{AvgPool}\left(\phi_{\omega}^{\ell-1}(\mathbf{x})[\mathrm{B}_{\ell-1}]\right)\right)}$$

Results

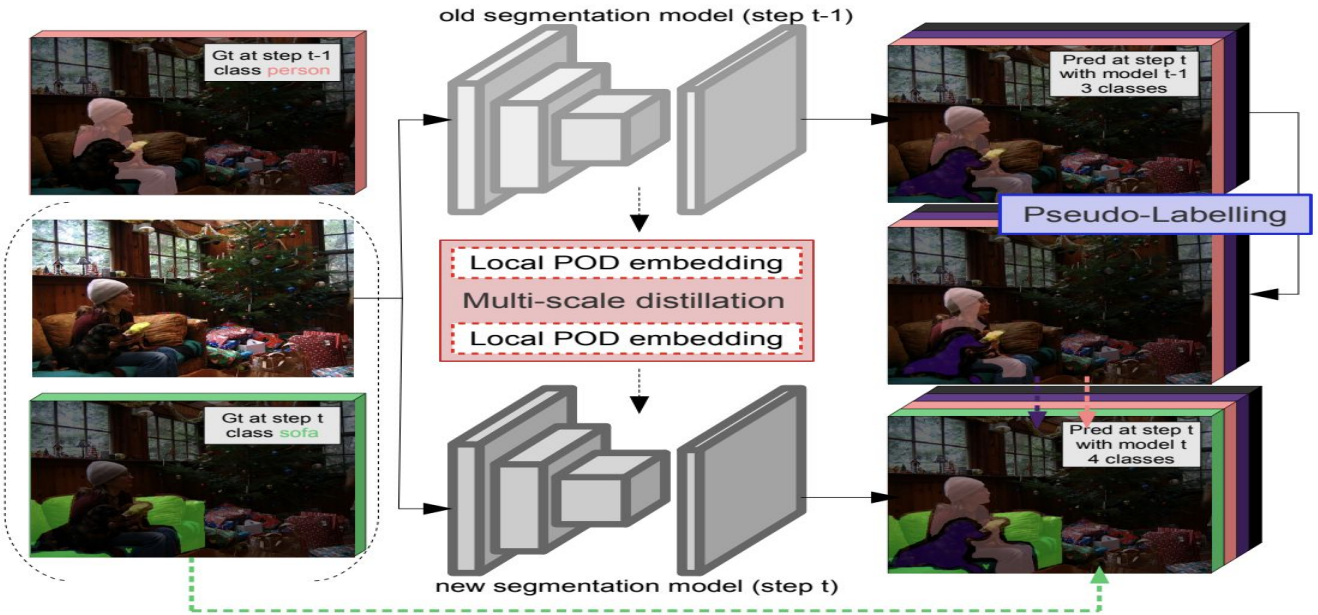
Method	19-1						15-5						15-1					
	Disjoint			Overlapped			Disjoint			Overlapped			Disjoint			Overlapped		
	1-19	20	all	1-19	20	all	1-15	16-20	all	1-15	16-20	all	1-15	16-20	all	1-15	16-20	all
FT	5.8	12.3	6.2	6.8	12.9	7.1	1.1	33.6	9.2	2.1	33.1	9.8	0.2	1.8	0.6	0.2	1.8	0.6
PI [45]	5.4	14.1	5.9	7.5	14.0	7.8	1.3	34.1	9.5	1.6	33.3	9.5	0.0	1.8	0.4	0.0	1.8	0.4
EWC [19]	23.2	16.0	22.9	26.9	14.0	26.3	26.7	37.7	29.4	24.3	35.5	27.1	0.3	4.3	1.3	0.3	4.3	1.3
RW [4]	19.4	15.7	19.2	23.3	14.2	22.9	17.9	36.9	22.7	16.6	34.9	21.2	0.2	5.4	1.5	0.0	5.2	1.3
LwF [20]	53.0	9.1	50.8	51.2	8.5	49.1	58.4	37.4	53.1	58.9	36.6	53.3	0.8	3.6	1.5	1.0	3.9	1.8
LwF-MC [32]	63.0	13.2	60.5	64.4	13.3	61.9	67.2	41.2	60.7	58.1	35.0	52.3	4.5	7.0	5.2	6.4	8.4	6.9
ILT [27]	69.1	16.4	66.4	67.1	12.3	64.4	63.2	39.5	57.3	66.3	40.6	59.9	3.7	5.7	4.2	4.9	7.8	5.7
MiB [3]	69.6	25.6	67.4	70.2	22.1	67.8	71.8	**43.3**	64.7	75.5	49.4	69.0	46.2	12.9	37.9	35.1	13.5	29.7
CMA	**75.5**	**30.8**	**73.3**	**75.5**	**34.8**	**73.4**	**76.8**	42.1	**65.2**	**77.2**	**49.9**	**70.4**	**57.2**	**15.5**	**46.7**	**55.7**	**14.1**	**45.3**
Joint	77.4	78.0	77.4	77.4	78.0	77.4	79.1	72.6	77.4	79.1	72.6	77.4	79.1	72.6	77.4	79.1	72.6	77.4

Method	15-5 disjoint			19-1 disjoint		
	1-15	16-20	all	1-19	20	all
Baseline	59.3	34.1	53.0	69.7	24.7	67.4
+ FD	63.2	39.5	57.3	60.3	16.3	58.1
+ AD_{SP}	71.5	42.7	64.3	74.2	29.2	71.9
+ AD_{CH}	10.5	11.2	10.6	1.7	16.2	2.4
+ AD_{SP} & AD_{CH}	**72.5**	**41.6**	**64.8**	**75.1**	**30.0**	**72.8**

Attention distillation helps!

PLOP

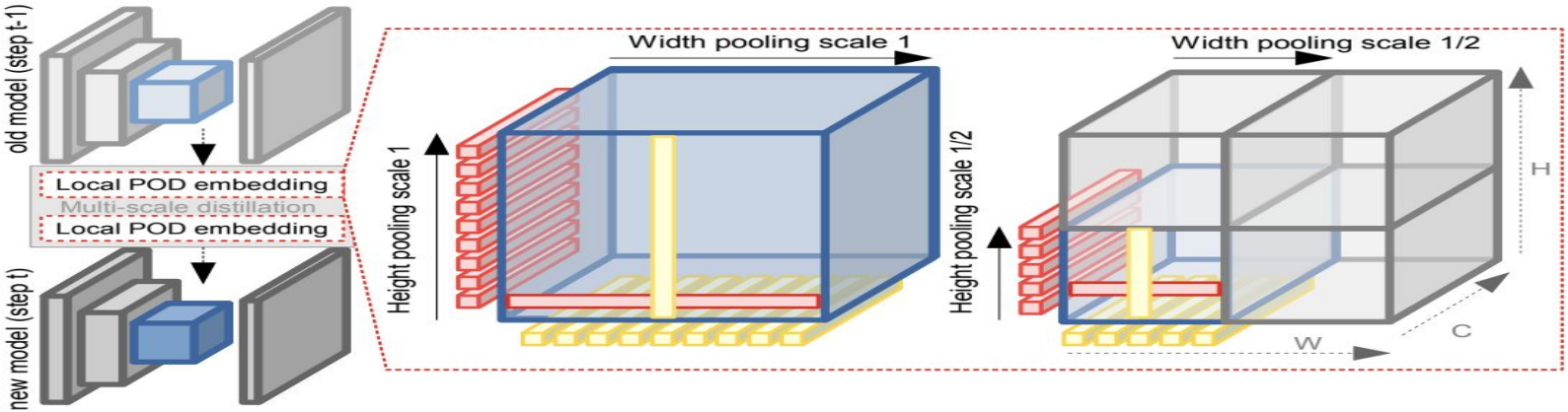
Multi-scale POD feature distillation. Pseudolabeling for BG shift.



Douillard, A., Chen, Y., Dapogny, A., & Cord, M. (2021). Plop: Learning without forgetting for continual semantic segmentation. CVPR

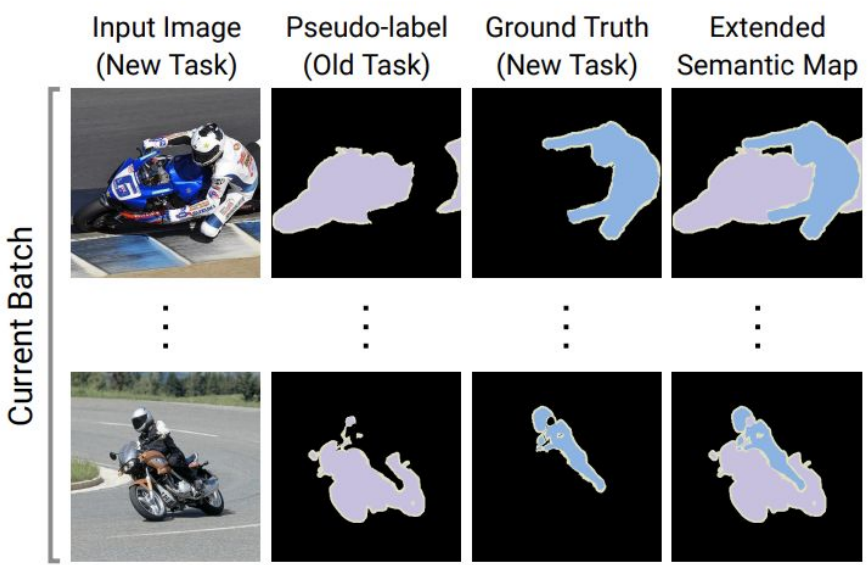
PLOP

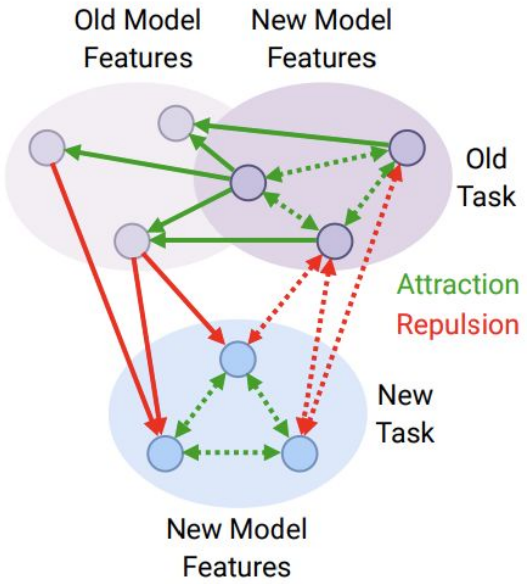
Pooling Distillation



Douillard, A., Chen, Y., Dapogny, A., & Cord, M. (2021). Plop: Learning without forgetting for continual semantic segmentation. CVPR

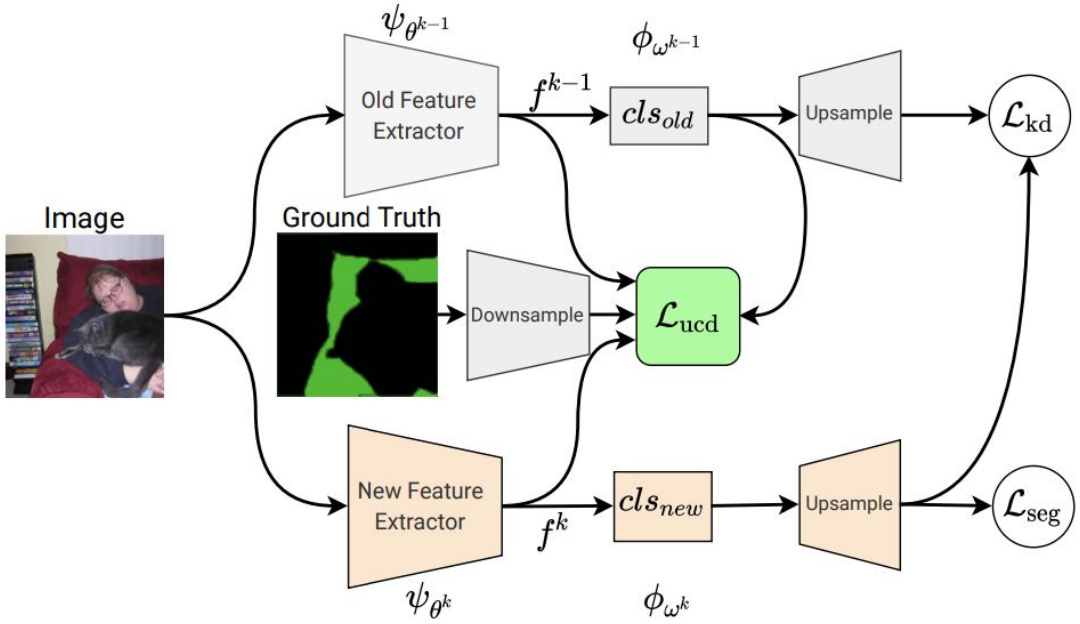
Uncertainty-aware Contrastive Distillation (UCD)



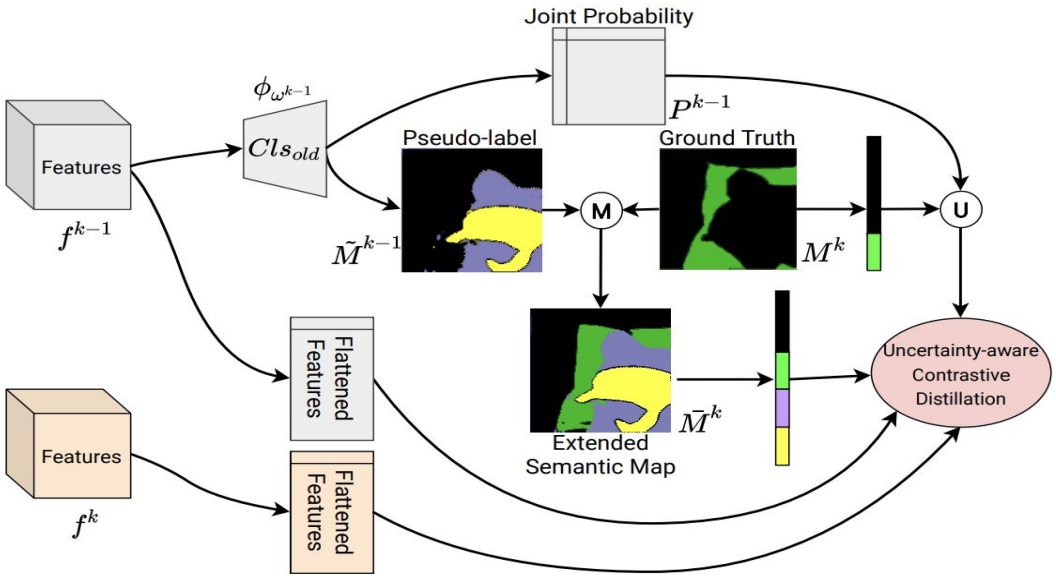


Yang, G., Fini, E., Xu, D., Rota, P., Ding, M., Nabi, M., ... & Ricci, E. (2022). Uncertainty-aware Contrastive Distillation for Incremental Semantic Segmentation. IEEE TPAMI

Uncertainty-aware Contrastive Distillation (UCD)

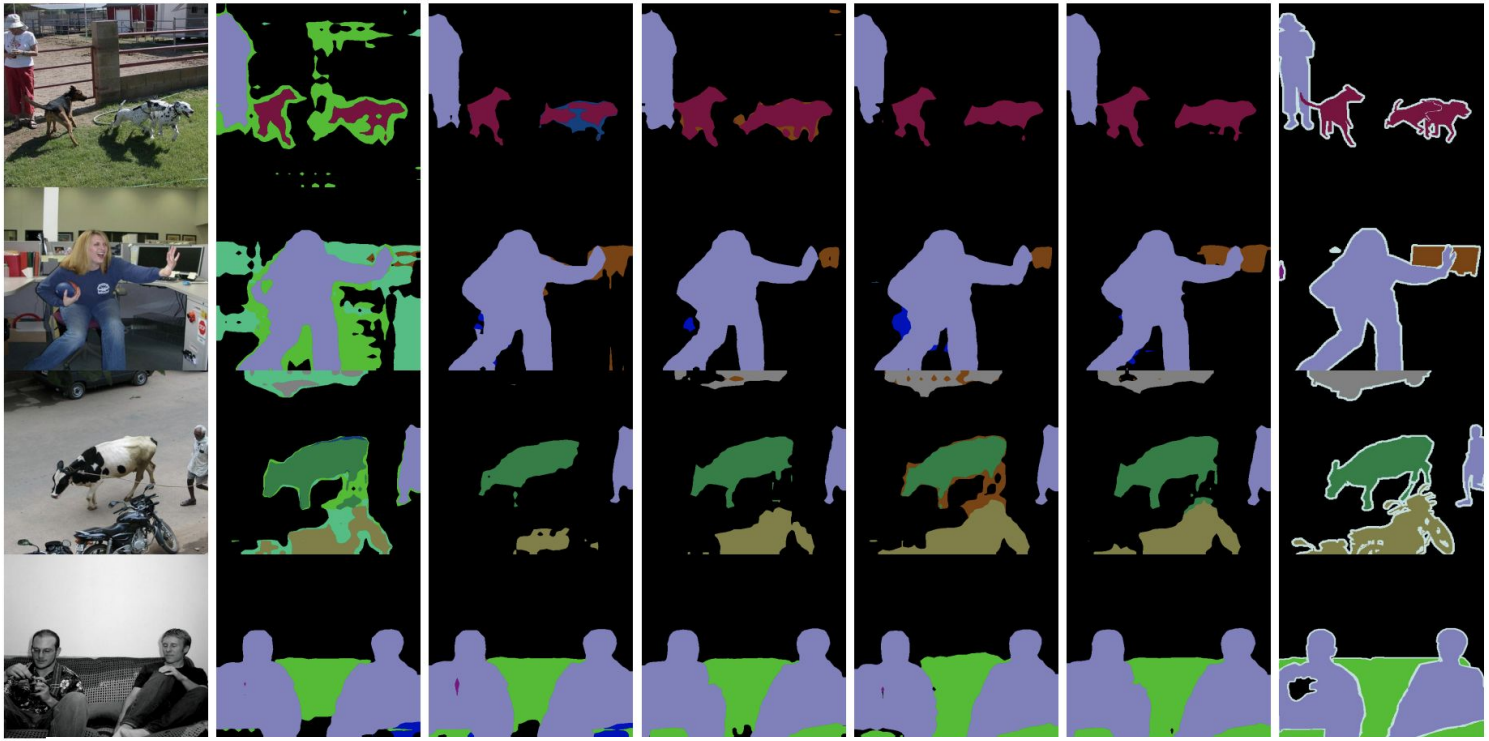


Uncertainty-aware Contrastive Distillation (UCD)



$$\log \frac{\exp\left(\delta(f_a^k, f^+)/\tau\right)}{\sum_{f^- \in H^k(a)} \exp\left(\delta(f_a^k, f^-)/\tau\right)}$$

Results



Results

Method	100-50			100-10							50-50			
	1-100	101-150	all	1-100	100-110	110-120	120-130	130-140	140-150	all	1-50	51-100	101-150	all
FT	0.0	24.9	8.3	0.0	0.0	0.0	0.0	0.0	16.6	1.1	0.0	0.0	22.0	7.3
LwF [14]	21.1	25.6	22.6	0.1	0.0	0.4	2.6	4.6	16.9	1.7	5.7	12.9	22.8	13.9
LwF-MC [16]	34.2	10.5	26.3	18.7	2.5	8.7	4.1	6.5	5.1	14.3	27.8	7.0	10.4	15.1
ILT [19]	22.9	18.9	21.6	0.3	0.0	1.0	2.1	4.6	10.7	1.4	8.4	9.7	14.3	10.8
Inc.Seg [49]	36.6	0.4	24.6	32.4	0.0	0.2	0.0	0.0	0.0	21.7	40.2	1.3	0.3	14.1
SDR [51]	37.4	24.8	33.2	28.9	-	-	-	-	-	21.7	40.9	-	-	29.5
UCD	40.4	27.3	36.0	28.6	13.0	13.1	9.2	10.7	16.1	23.2	39.3	25.3	19.1	27.9
MiB [23]	37.9	27.9	34.6	31.8	10.4	14.8	**12.8**	**13.6**	18.7	25.9	35.5	22.2	**23.6**	27.0
MiB + SDR [51]	37.5	25.5	33.5	28.9	-	-	-	-	-	23.2	**42.9**	-	-	**31.3**
MiB + UCD	**40.5**	**28.1**	**36.4**	33.4	**15.2**	**15.3**	10.8	12.5	**18.8**	**27.1**	40.2	**25.9**	19.5	28.5
PLOP* [29]	29.8	4.2	22.2	32.1	1.9	10.0	0.8	1.2	0.1	22.3	19.2	0.4	0.4	6.6
PLOP + UCD	33.2	4.7	23.7	**35.7**	2.1	11.1	0.9	1.4	0.1	24.8	21.3	0.4	0.4	7.4
Joint	44.3	28.2	38.9	44.3	26.1	42.8	26.7	28.1	17.3	38.9	51.1	38.3	28.2	38.9

Summary

- Significant and growing interest in the last few years on continual learning within deep learning
- Significant improvements over standard benchmark but focus still mostly on simplified scenarios.
- Huge space of possible and significant explorations.
- CL has the potential to push for the next generation of truly intelligent robust and autonomous AI systems: efficient, effective, scalable, hence sustainable.

Questions?