heuritech

SCIENCES SORBONNE UNIVERSITÉ

Small-Task Incremental Learning

arXiv Preprint

Arthur Douillard
Matthieu Cord
Charles Ollion
Thomas Robert
Eduardo Valle

22/05/2020

LIP6

MLIA Machine Learning & Deep Learning for Information Access

@Ar_Douillard
arthurdouillard.com

Who

Who Are We

Arthur Douillard

1^{st} year PhD student
Research Scientist

Matthieu Cord

Professor
Senior Research Scientist

Charles Ollion

Head of Research

Thomas Robert

Research Scientist

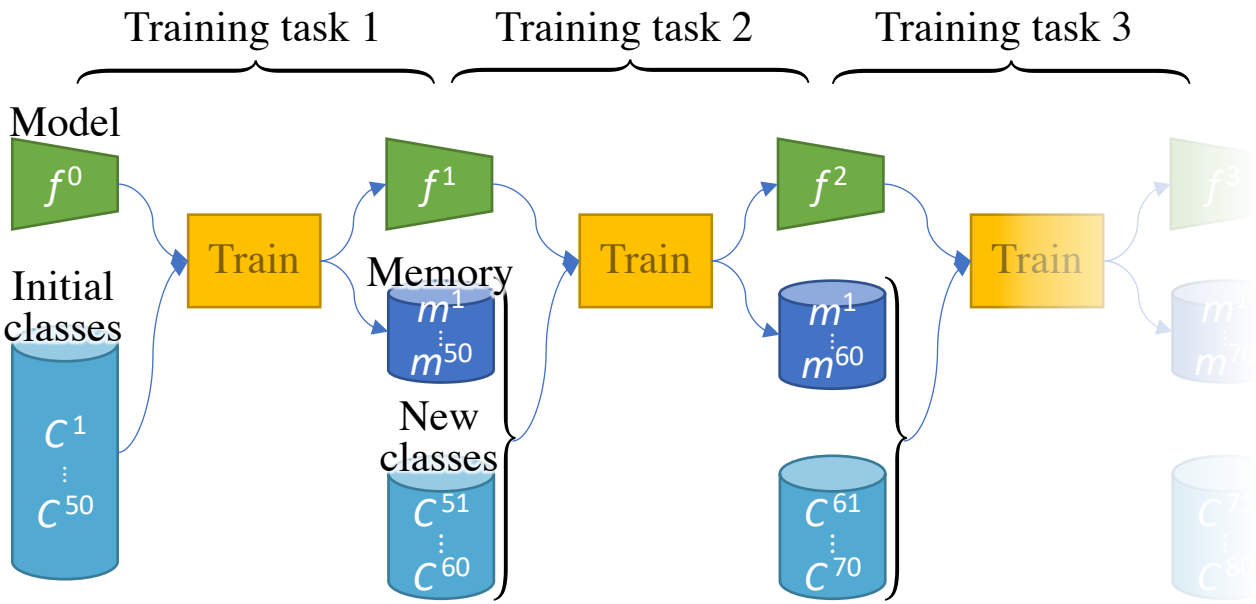
Eduardo Valle

Professor

The Task

The Task

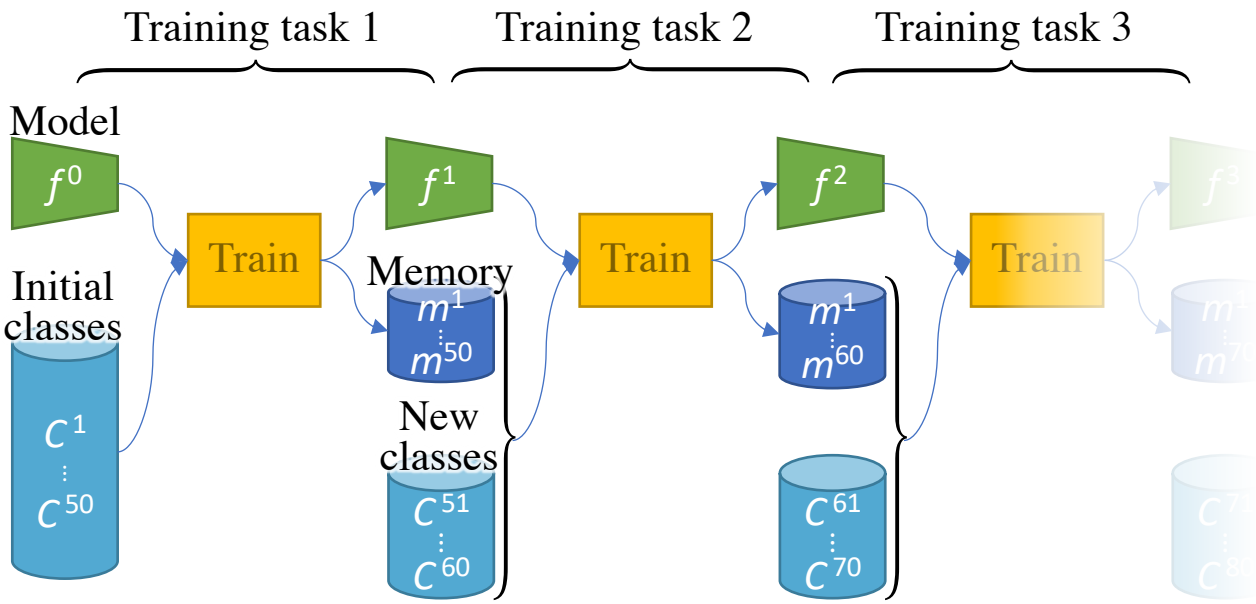
- **New Classes (NC)** setting [1] where each new task brings new classes
- A very limited subset of previous classes data is preserved into a **memory** for rehearsal [2]
- Model is from task T is copied for task T+1



[1]: CORe50: a New Dataset and Benchmark for Continuous Object Recognition, Lomonaco et al., 2017, PMLR
[2]: Catastrophic Forgetting, Rehearsal, and Pseudorehearsal, Robins, 1995, Connection Science

Evaluation

- After each task, the model is evaluated on **all seen classes**
- We don't have access to task id in inference
- Final score is the average of all task accuracies [1]



$$AvgIncAcc = \frac{1}{N_{tasks}}(Acc_{0:50} + Acc_{0:60} + Acc_{0:70} + \ldots)$$

[1]: iCaRL: Incremental Classifier and Representation Learning, Rebuffi et al. CVPR 2017

Datasets & Increments

heuritech

SCIENCES SORBONNE UNIVERSITÉ

- Following *Hou et al.* [1], we evaluate on **CIFAR100**, **ImageNet100**, and **ImageNet1000**.

- We use a fixed amount of memory $M_{per} = 20$
  - More challenging than iCaRL setting $M_{total} = 2000$

- We also train the model on **half the total classes**, then incrementally add more classes

- In our case, we focus on **large amount of very small tasks**
  - Up to tasks made of a single new class

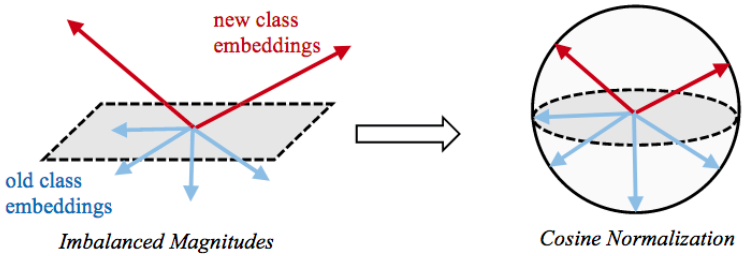
[1]: Learning an Unified Classifier Incrementally via Rebalancing, Hou et al. CVPR 2019

Baseline

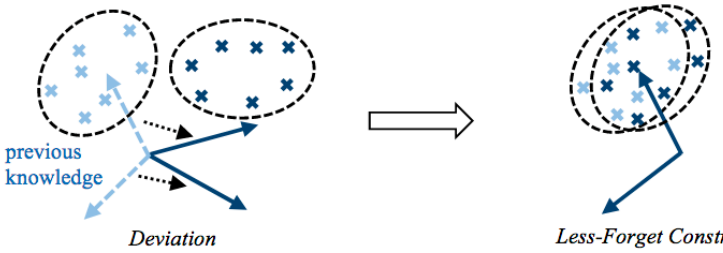
UCIR

- Our baseline is *Hou et al.*'s UCIR [1]

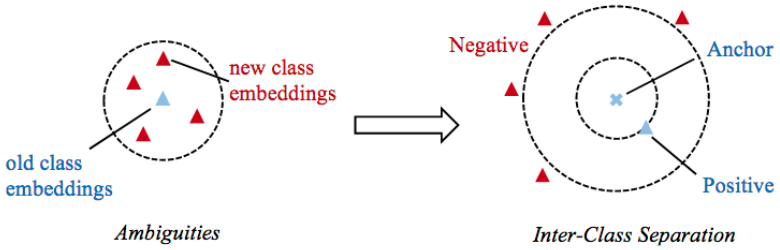
Cosine classifier



Cosine constraint on final embedding as distillation



Hinge-based regularization



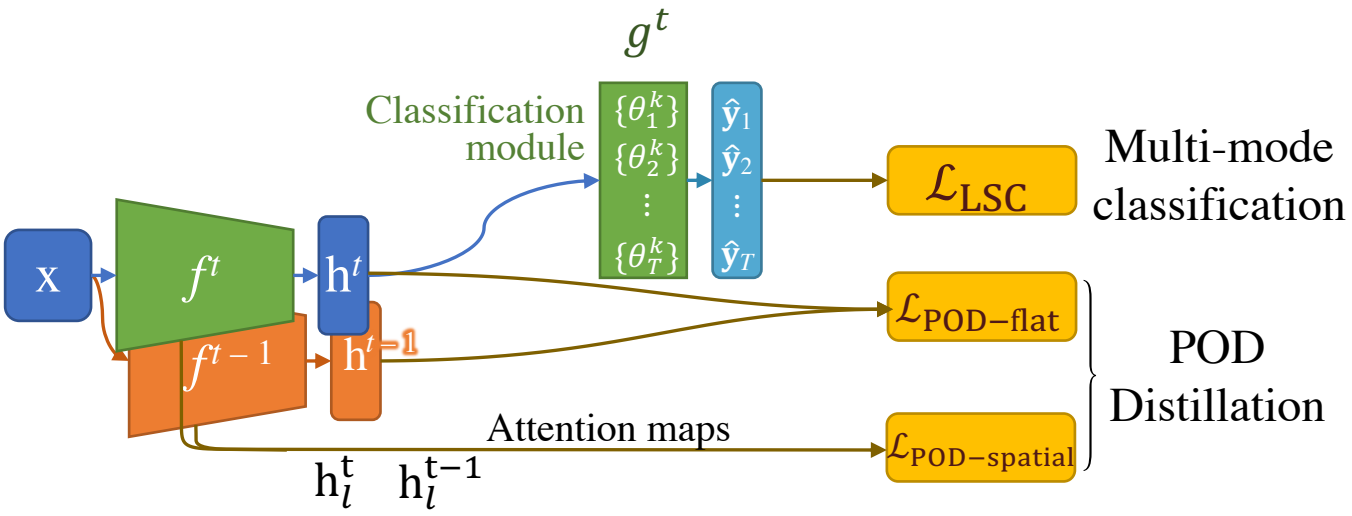
[1]: Learning an Unified Classifier Incrementally via Rebalancing, Hou et al. CVPR 2019

Our Model:

PODNet

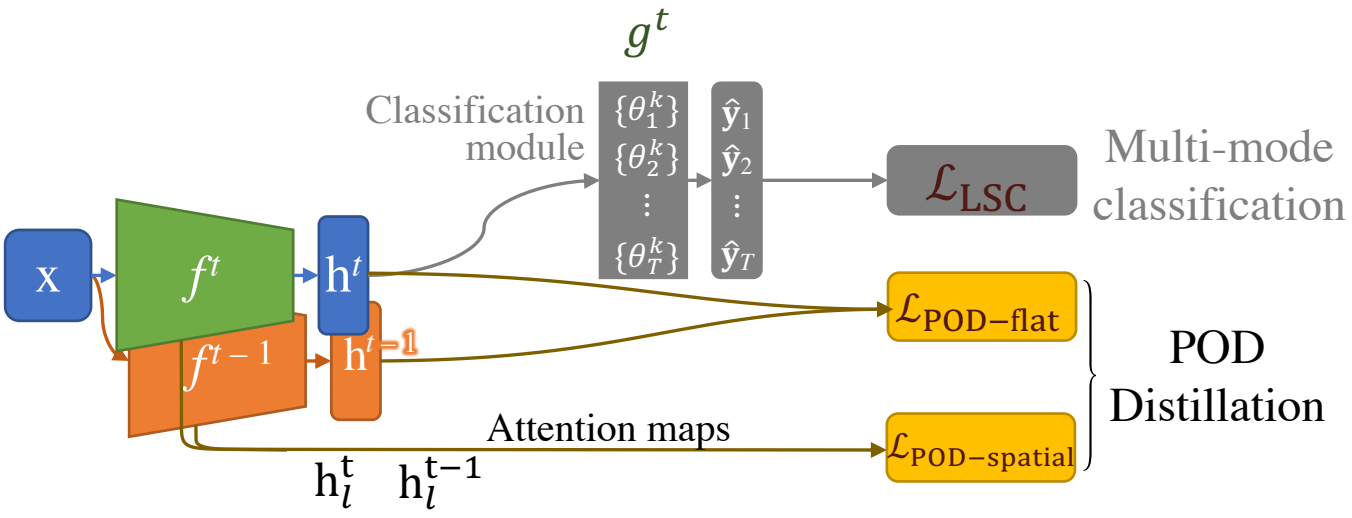
The Model

- **A classification loss**, to discriminate classes
- **Two distillation losses**, to reduce catastrophic forgetting



POD Distillation

- **A classification loss**, to discriminate classes
- **Two distillation losses**, to reduce catastrophic forgetting



Shortcoming

$$\mathcal{L}_{\text{Less-Forget}}(\mathbf{h}^{t-1}, \mathbf{h}^{t}) = \sum_{d=1}^{D} \left\| \mathbf{h}_d^{t-1} - \mathbf{h}_d^{t} \right\|^2$$

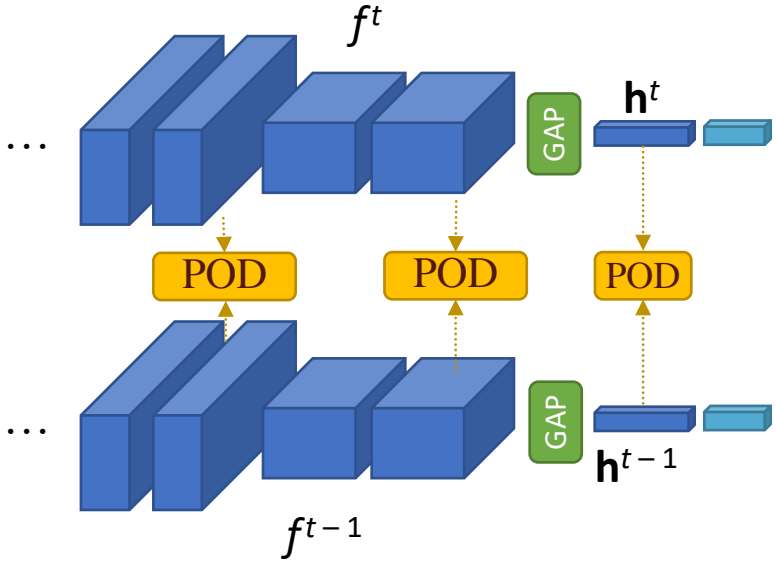
- We found that *Hou et al.* loss was too rigid:
  - **Forgetting** was alleviated, with a high loss factor
  - **Plasticity** was hurt, as it was difficult to learn new classes

Shortcoming

$$\mathcal{L}_{\text{Less-Forget}}(\mathbf{h}^{t-1}, \mathbf{h}^{t}) = \sum_{d=1}^{D} \left\| \mathbf{h}_d^{t-1} - \mathbf{h}_d^{t} \right\|^2$$

- We found that *Hou et al.* loss was too rigid:
  - **Forgetting** was alleviated, with a high loss factor
  - **Plasticity** was hurt, as it was difficult to learn new classes

- Furthermore, we only constraint the final embedding:
  - Cannot we exploit **intermediary embeddings**?
  - Can we design a loss explicitly for **images** as prior?



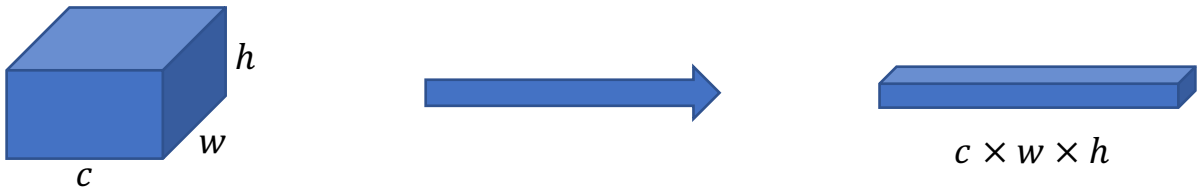
POD Distillation

heuritech SCIENCES SORBONNE UNIVERSITÉ

$$\mathcal{L}_{\text{Less-Forget}}(\mathbf{h}^{t-1}, \mathbf{h}^{t}) = \sum_{d=1}^{D} \left\| \mathbf{h}_d^{t-1} - \mathbf{h}_d^{t} \right\|^2$$

- Naïve generalization of *Hou et al.*'s loss to spatial features:

$$\mathcal{L}_{\text{POD-pixel}}(\mathbf{h}_\ell^{t-1}, \mathbf{h}_\ell^{t}) = \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} \left\| \mathbf{h}_{\ell,c,w,h}^{t-1} - \mathbf{h}_{\ell,c,w,h}^{t} \right\|^2$$



POD Distillation

$$\mathcal{L}_{\text{Less-Forget}}(\mathbf{h}^{t-1}, \mathbf{h}^{t}) = \sum_{d=1}^{D} \left\| \mathbf{h}_d^{t-1} - \mathbf{h}_d^{t} \right\|^2$$

- Naïve generalization of *Hou et al.*'s loss to spatial features:

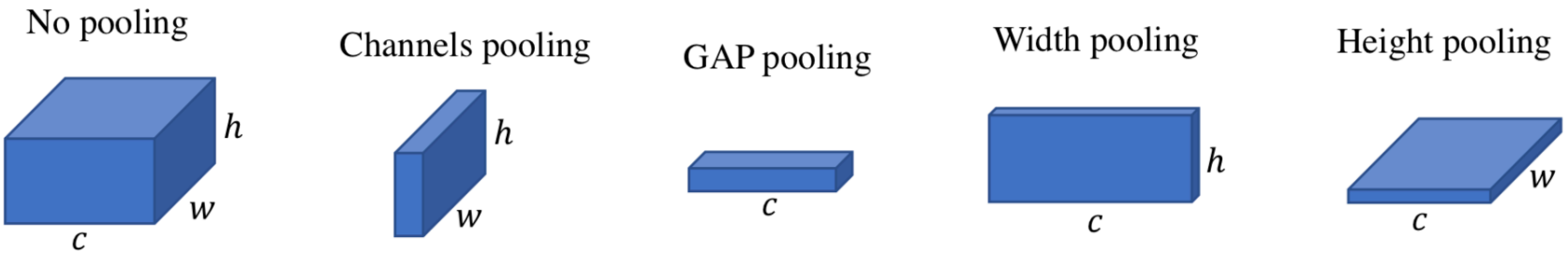
$$\mathcal{L}_{\text{POD-pixel}}(\mathbf{h}_\ell^{t-1}, \mathbf{h}_\ell^{t}) = \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} \left\| \mathbf{h}_{\ell,c,w,h}^{t-1} - \mathbf{h}_{\ell,c,w,h}^{t} \right\|^2$$

- However:
  - **No plasticity is left**, the loss is sensitive to pixel outliers
  - We don't really exploit the multiple dimensions of an image

POD Distillation

$$\mathcal{L}_{\text{POD-pixel}}(\mathbf{h}_\ell^{t-1}, \mathbf{h}_\ell^t) = \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} \left\| \mathbf{h}_{\ell,c,w,h}^{t-1} - \mathbf{h}_{\ell,c,w,h}^{t} \right\|^2$$

- More permissive loss by **pooling along a particular axis** before distilling:
  - **Not enforcing pixel-wise match but similar statistics**



Note that Channels-pooling+L2 distance is equivalent to Zagoruyko and Komodakis 2017's loss.

POD Distillation

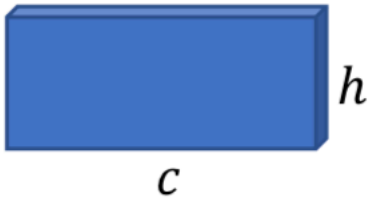
- The best **pooling** found, trading best the **stability** with **plasticity** is to pool along the spatial dimension:

$$\mathcal{L}_{\text{POD-width}}(\mathbf{h}_\ell^{t-1}, \mathbf{h}_\ell^t) = \sum_{c=1}^{C} \sum_{h=1}^{H} \left\| \sum_{w=1}^{W} \mathbf{h}_{\ell,c,w,h}^{t-1} - \sum_{w=1}^{W} \mathbf{h}_{\ell,c,w,h}^{t} \right\|^2$$

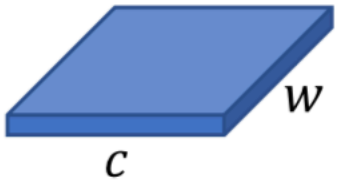
- Likewise, for the height, and then using both:

$$\mathcal{L}_{\text{POD-spatial}}(\mathbf{h}_\ell^{t-1}, \mathbf{h}_\ell^t) = \mathcal{L}_{\text{POD-width}}(\mathbf{h}_\ell^{t-1}, \mathbf{h}_\ell^t) + \mathcal{L}_{\text{POD-height}}(\mathbf{h}_\ell^{t-1}, \mathbf{h}_\ell^t)$$

Width pooling



Height pooling



POD Results

heuritech

SCIENCES SORBONNE UNIVERSITÉ

- POD-pixel is equivalent to *Hou et al.* [1]'s loss applied to spatial features
- GradCam is used in *Dhar et al.* [2]

- While they may work with large increments, they don't with **large amount of small increments**

Forgetting is heavy, thus plasticity is often sacrificed to get a okay performance

Spatial statistics are **more robust** and **less rigid** than pixel-wise distillations.

[1]: Learning an Unified Classifier Incrementally via Rebalancing, Hou et al. CVPR 2019
[2]: Learning without Memorizing, Dhar et al. CVPR 2019

POD Results

heuritech SCIENCES SORBONNE UNIVERSITÉ

With POD-flat

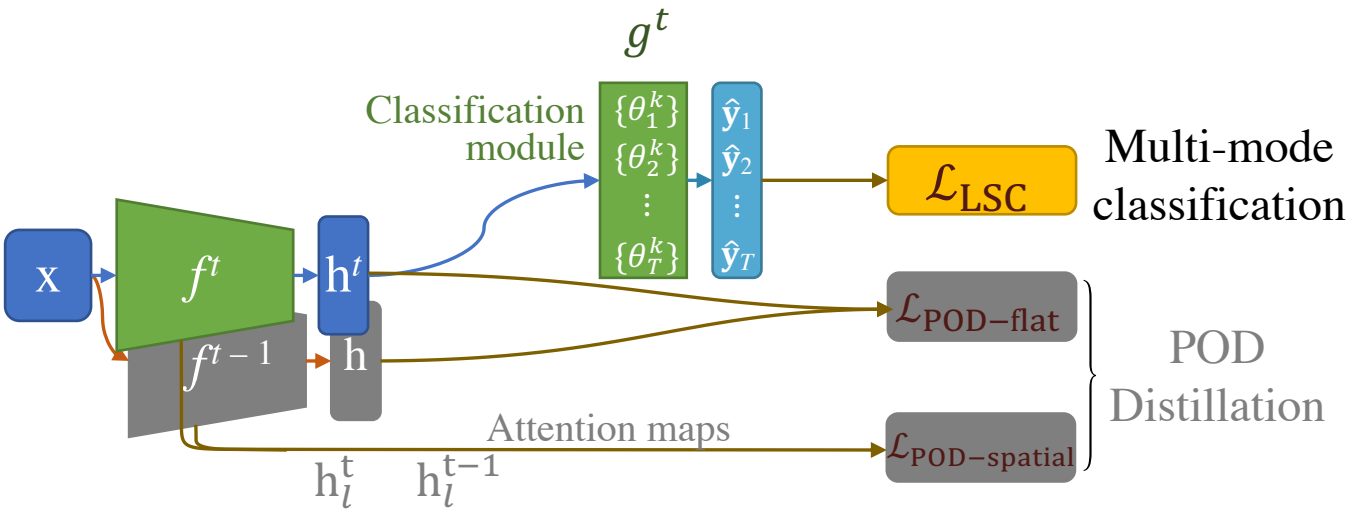
Loss	NME	CNN
None	53.29	52.98
POD-pixels	49.74	52.34
POD-channels	57.21	54.64
POD-gap	58.80	55.95
POD-width	60.92	57.51
POD-height	60.64	57.50
POD-spatial	**61.40**	**57.98**
GradCam [4]	54.13	52.48
Perceptual Style [13]	51.01	52.25

Without POD-flat

Loss	NME	CNN
None	41.56	40.76
POD-pixels	42.21	40.81
POD-channels	55.91	50.34
POD-gap	57.25	53.87
POD-width	61.25	57.51
POD-height	61.24	57.50
POD-spatial	**61.42**	**57.64**
GradCam [4]	41.89	42.07
Perceptual Style [13]	41.74	40.80

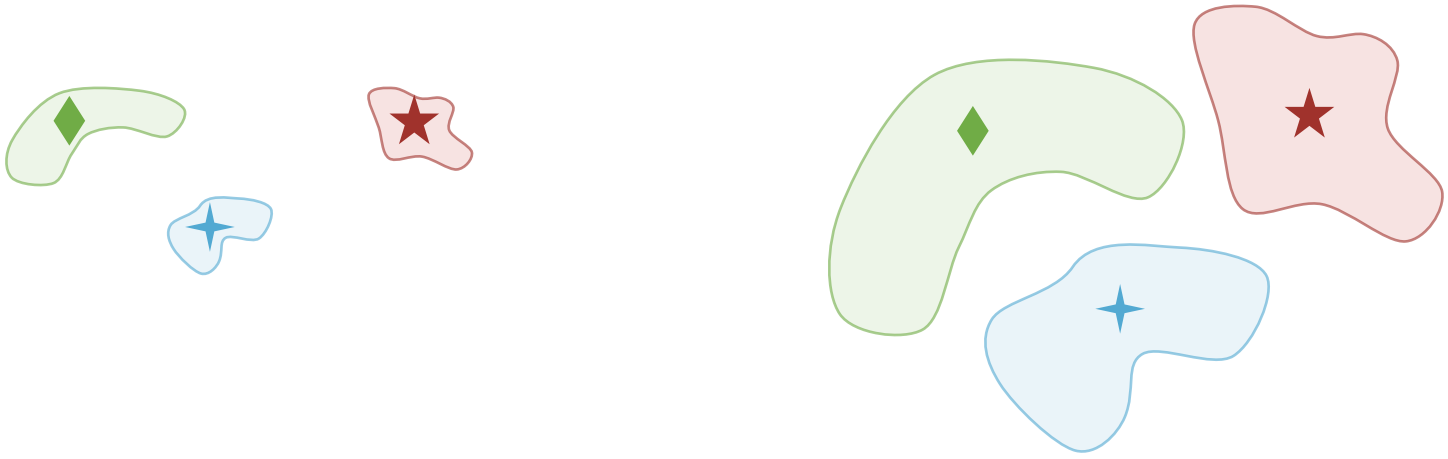
The Model

- **A classification loss**, to discriminate classes
- **Two distillation losses**, to reduce catastrophic forgetting



The Model

- Even with distillation losses, the **embedding distribution change** a little
- We found that each class distribution become **stretched**



Task 1

Task N

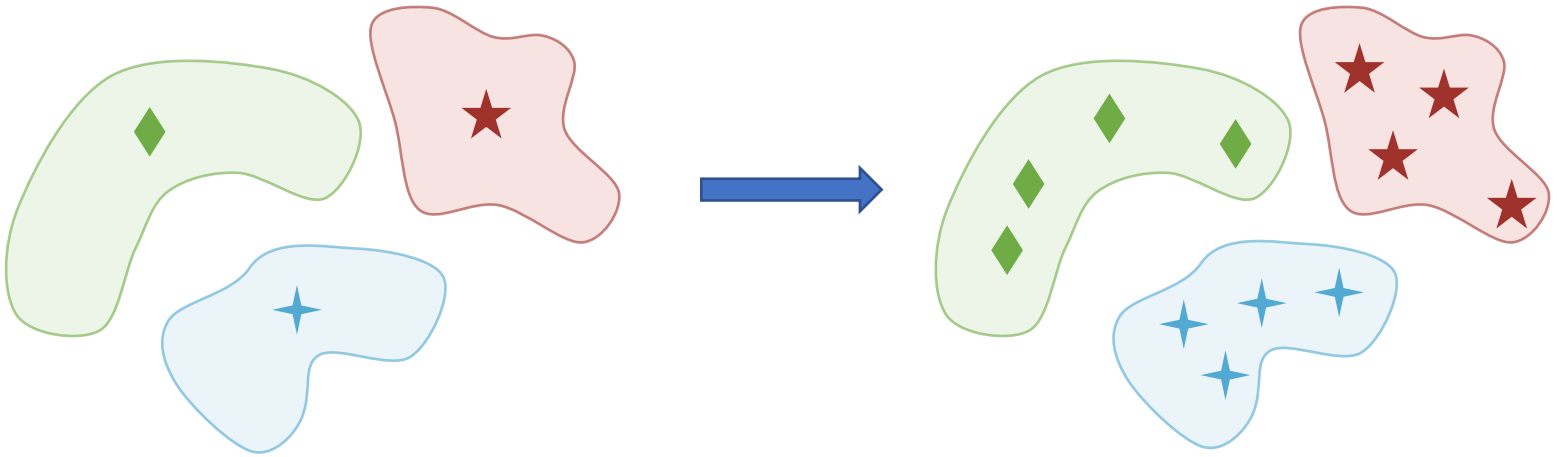
The Model

- Even with distillation losses, the **embedding distribution change** a little
- We found that each class distribution become **stretched**
- The **cosine classifier** is sensitive to those changes, as it models a **unique majority mode** per class through its class proxies

Task 1 Task N

The Model

- Multi-modes makes the classifier more **robust to distribution change**



One mode per class

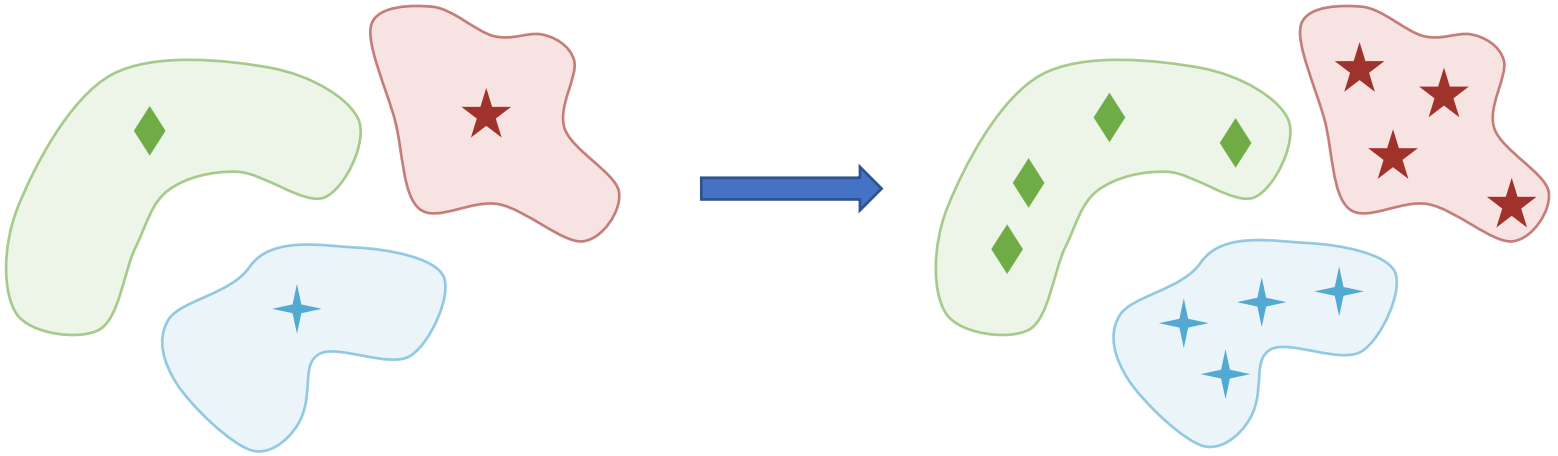
Four modes per class

The Model

- Multi-modes classifier is a **weighted average of local mode similarity**:

$$\hat{\mathbf{y}}_c = \frac{\exp\left(\eta \langle \boldsymbol{\theta}_c, \mathbf{h} \rangle\right)}{\sum_i \exp\left(\eta \langle \boldsymbol{\theta}_i, \mathbf{h} \rangle\right)}$$

$$s_{c,k} = \frac{\exp \langle \boldsymbol{\theta}_{c,k}, \mathbf{h} \rangle}{\sum_i \exp \langle \boldsymbol{\theta}_{c,i}, \mathbf{h} \rangle}, \qquad \hat{\mathbf{y}}_c = \sum_k s_{c,k} \langle \boldsymbol{\theta}_{c,k}, \mathbf{h} \rangle$$



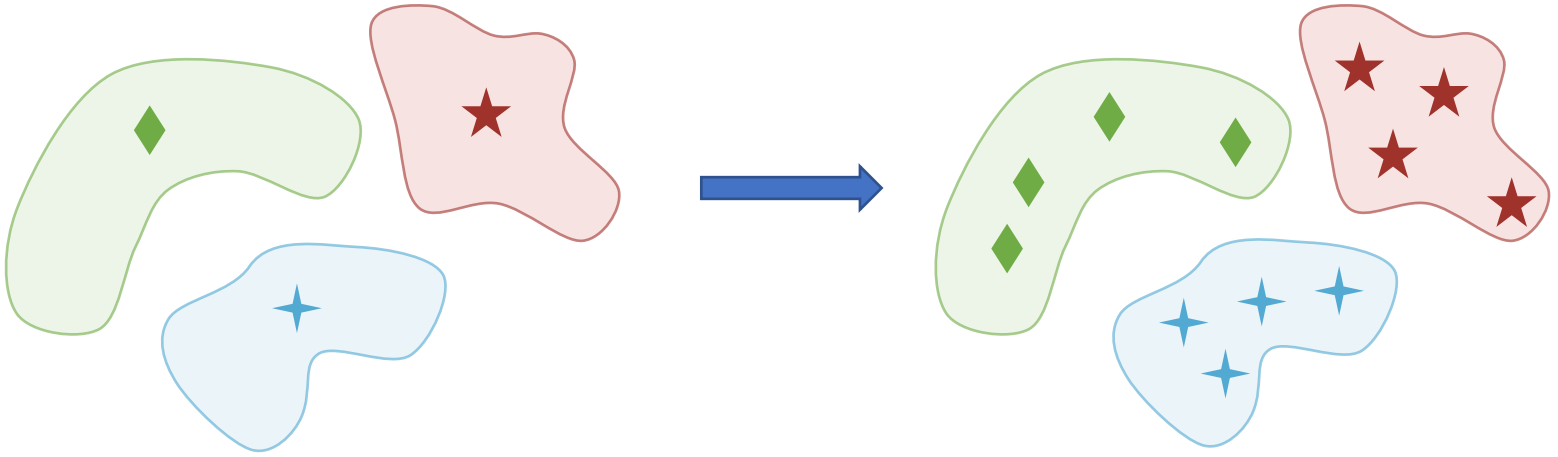
One mode per class

Four modes per class

The Model

- Multi-modes makes the classifier more **robust to distribution change**

- Compared to single-mode:
  - No significant gain in new classes accuracies
  - Gain of up to 2 points in **old class accuracies**



One mode per class

Four modes per class

Results

Evaluation type

heuritech

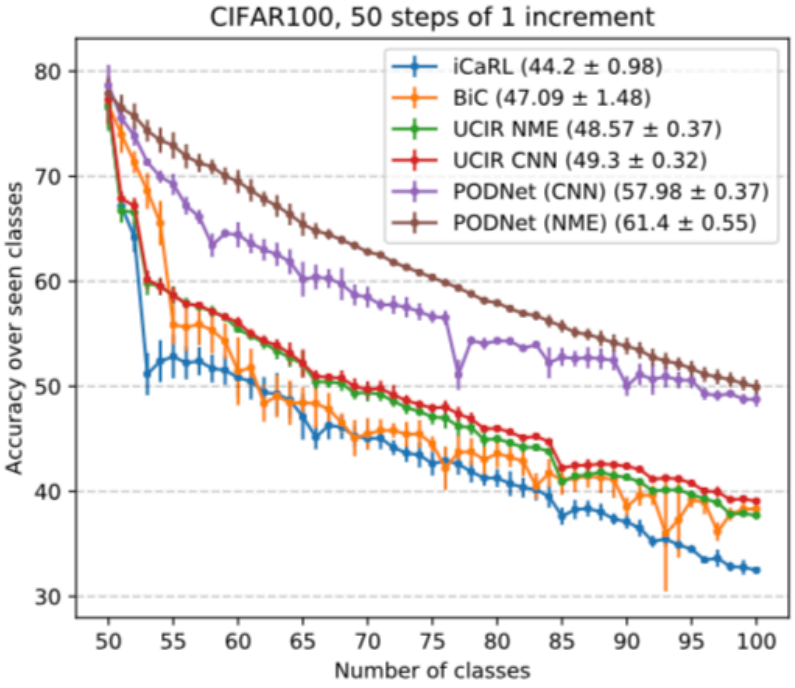
SCIENCES SORBONNE UNIVERSITÉ

- As *Hou et al.* we evaluate our model with two methods:
  - Nearest Mean Exemplar (**NME**): classifying with a KNN on the embedding
  - **CNN**: classifying with classifier logits + argmax

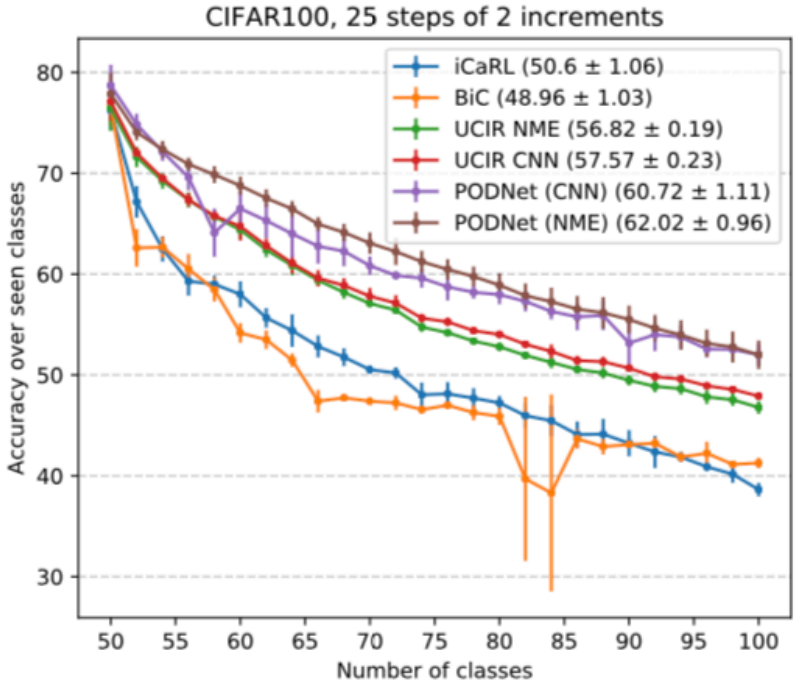
CIFAR100

	CIFAR100			
	50 steps	25 steps	10 steps	5 steps
New classes per step	1	2	5	10
*iCaRL** [28]	—	—	52.57	57.17
iCaRL	44.20 ± 0.98	50.60 ± 1.06	53.78 ± 1.16	58.08 ± 0.59
BiC [32]	47.09 ± 1.48	48.96 ± 1.03	53.21 ± 1.01	56.86 ± 0.46
*UCIR (NME)** [12]	—	—	60.12	63.12
UCIR (NME)	48.57 ± 0.37	56.82 ± 0.19	60.83 ± 0.70	63.63 ± 0.87
*UCIR (CNN)** [12]	—	—	60.18	63.42
UCIR (CNN)	49.30 ± 0.32	57.57 ± 0.23	61.22 ± 0.69	64.01 ± 0.91
PODNet (NME)	**61.40 ± 0.68**	**62.71 ± 1.26**	**64.03 ± 1.30**	**64.48 ± 1.32**
PODNet (CNN)	**57.98 ± 0.46**	**60.72 ± 1.36**	**63.19 ± 1.16**	**64.83 ± 0.98**

CIFAR100



(a) 50 steps, 1 class / step



(b) 25 steps, 2 classes / step

ImageNet

heuritech SCIENCES SORBONNE UNIVERSITÉ

	ImageNet100				Imagenet1000	
	50 steps	25 steps	10 steps	5 steps	10 steps	5 steps
New classes per step	1	2	5	10	50	100
iCaRL* [29]	—	—	59.53	65.04	46.72	51.36
iCaRL [29]	54.97	54.56	60.90	65.56	—	—
BiC [33]	46.49	59.65	65.14	68.97	44.31	45.72
UCIR (NME)* [12]	—	—	66.16	68.43	59.92	61.56
UCIR (NME) [12]	55.44	60.81	65.83	69.07	—	—
UCIR (CNN)* [12]	—	—	68.09	70.47	61.28	64.34
UCIR (CNN) [12]	57.25	62.94	67.82	71.04	—	—
PODNet (CNN)	**62.08**	**67.28**	**73.14**	**75.82**	**64.13**	**66.95**

Robustness Tests

heuritech

SCIENCES SORBONNE UNIVERSITÉ

Table 4. Effect of the memory size per class M_{per} on the models performance. Results from CIFAR100 with 50 steps, we report the average incremental accuracy

M_{per}	5	10	**20**	50	100	200
iCaRL	16.44	28.57	44.20	48.29	54.10	57.82
BiC	20.84	21.97	47.09	55.01	62.23	**67.47**
UCIR (NME)	21.81	41.92	48.57	56.09	60.31	64.24
UCIR (CNN)	22.17	42.70	49.30	57.02	61.37	65.99
PODNet (NME)	**48.37**	**57.20**	**61.40**	**62.27**	**63.14**	63.63
PODNet (CNN)	**35.59**	**48.54**	**57.98**	**63.69**	**66.48**	**67.62**

Table 5. Effect of the initial task size and the M_{total} on the models performance. We report the average incremental accuracy

(a) Evaluation of an easier memory constraint ($M_{\text{total}} = 2000$)

	Nb. steps	
Loss	50	10
iCaRL [29]	42.34	56.52
BiC [33]	48.44	55.03
UCIR (NME) [12]	54.08	62.89
UCIR (CNN) [12]	55.20	63.62
PODNet (NME)	**62.47**	**64.60**
PODNet (CNN)	**61.87**	**64.68**

(b) Varying initial task size for 50 steps with $M_{\text{per}} = 20$

	Initial task size				
Loss	10	20	30	40	**50**
iCaRL	40.97	41.28	43.38	44.35	44.20
BiC	41.58	40.95	42.27	45.18	47.09
UCIR (NME)	42.33	40.81	46.80	46.71	48.57
UCIR (CNN)	43.25	41.69	47.85	47.51	49.30
PODNet (NME)	**45.09**	**49.03**	**55.30**	**57.89**	**61.40**
PODNet (CNN)	**44.95**	**47.68**	**52.88**	**55.42**	**57.98**

Ablations

(a) Comparison of the performance of the model when disabling parts of the complete PODNet loss

Classifier	POD-flat	POD-spatial	NME	CNN
Cosine			40.76	37.93
Cosine	✓		50.06	46.73
Cosine		✓	59.01	57.27
Cosine	✓	✓	59.50	55.72
LSC-CE	✓	✓	59.86	57.45
LSC			41.56	40.76
LSC	✓		53.29	52.98
LSC		✓	**61.42**	57.64
LSC	✓	✓	61.40	**57.98**

Thanks for attending!

What are your questions?