Apprentissage continu appliqué à la classification d'images

Tutoriel PFIA, le 3 juillet 2025

Eva Feillet*, Adrian Popescu°, Céline Hudelot+

*LAMSADE, Université Paris Dauphine-PSL | °CEA list, Université Paris-Saclay

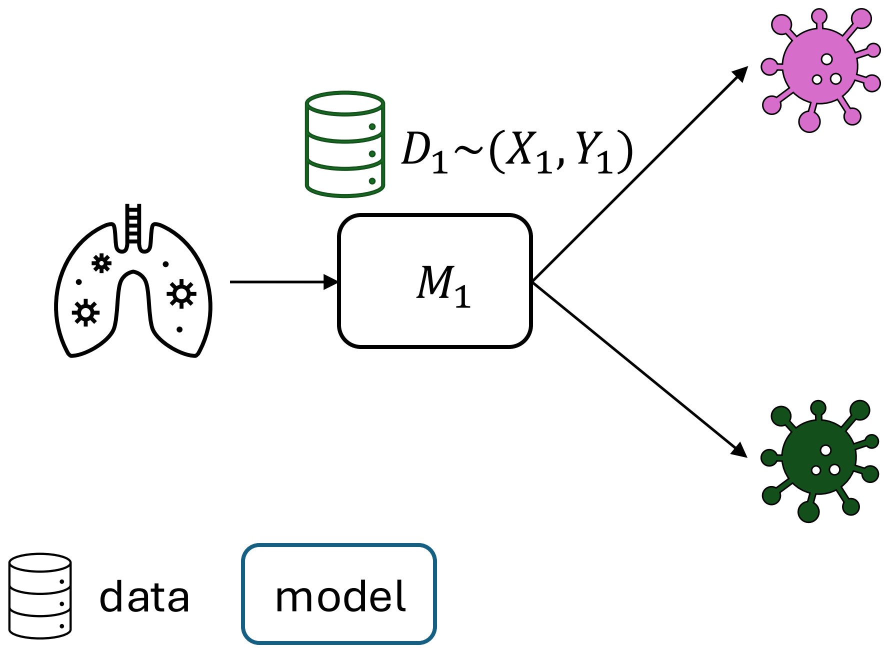
+MICS, CentraleSupélec, Université Paris-Saclay

Introduction

Why continual learning ?

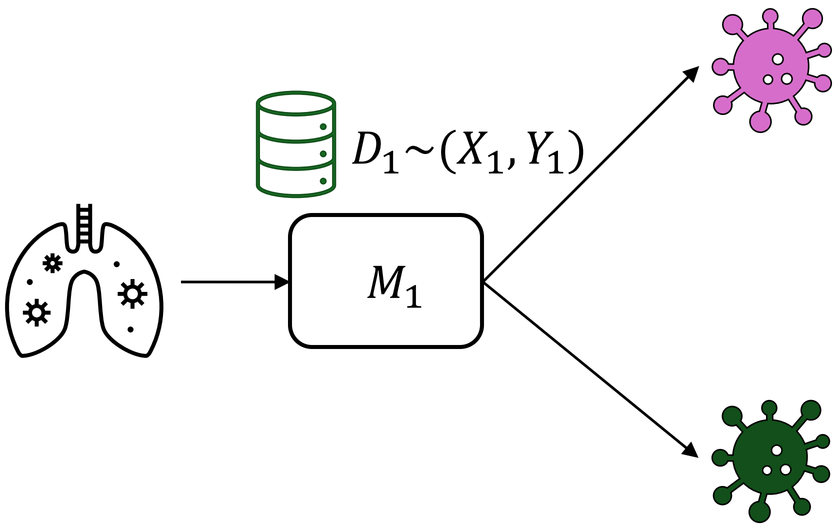
Continual Learning for Adaptive Models

Classic static supervised learning: Solve a specific task by learning from a fixed data distribution.

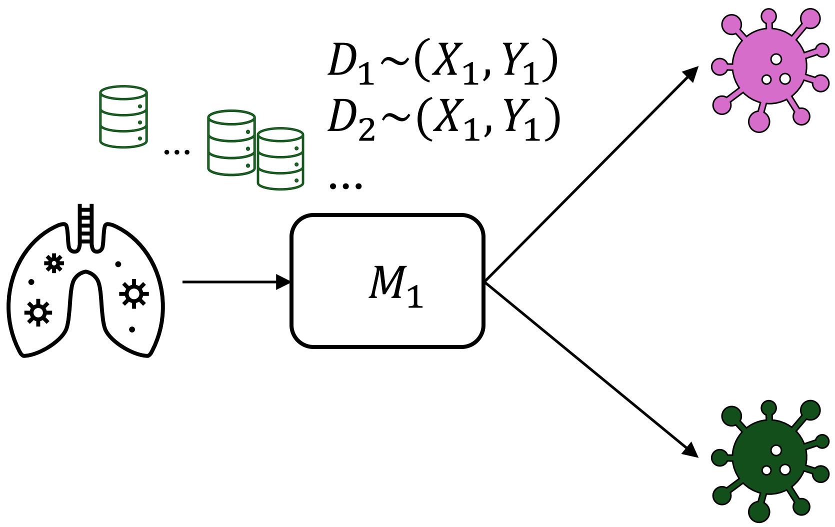


Continual Learning for Adaptive Models

Classic static supervised learning: Solve a specific task by learning from a fixed data distribution.



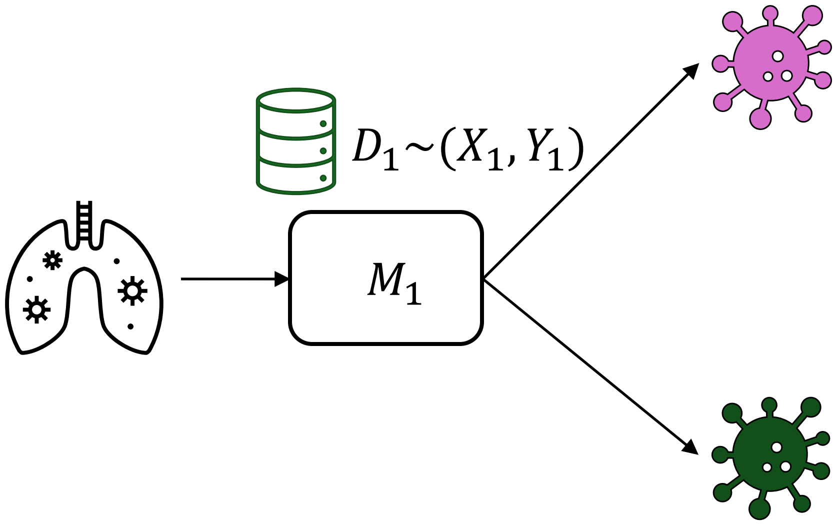
⁉️ What if… the training data comes as **a stream**?

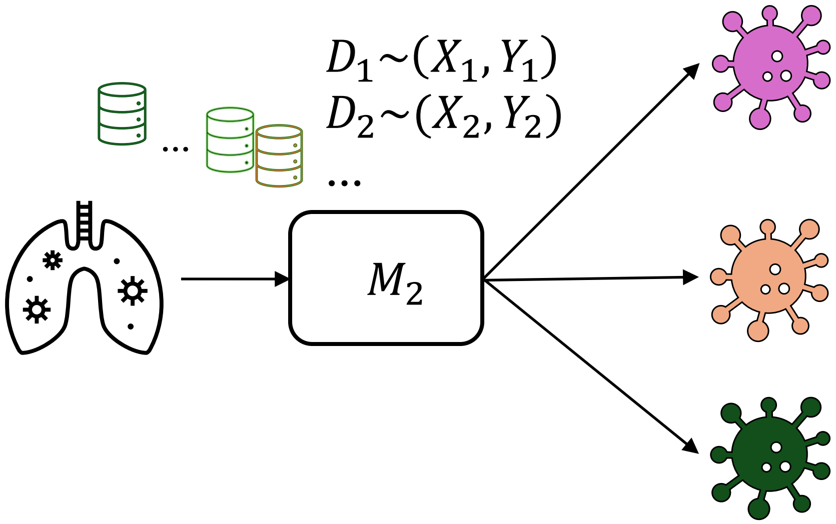


Continual Learning for Adaptive Models

Classic static supervised learning: Solve a specific task by learning from a fixed data distribution.

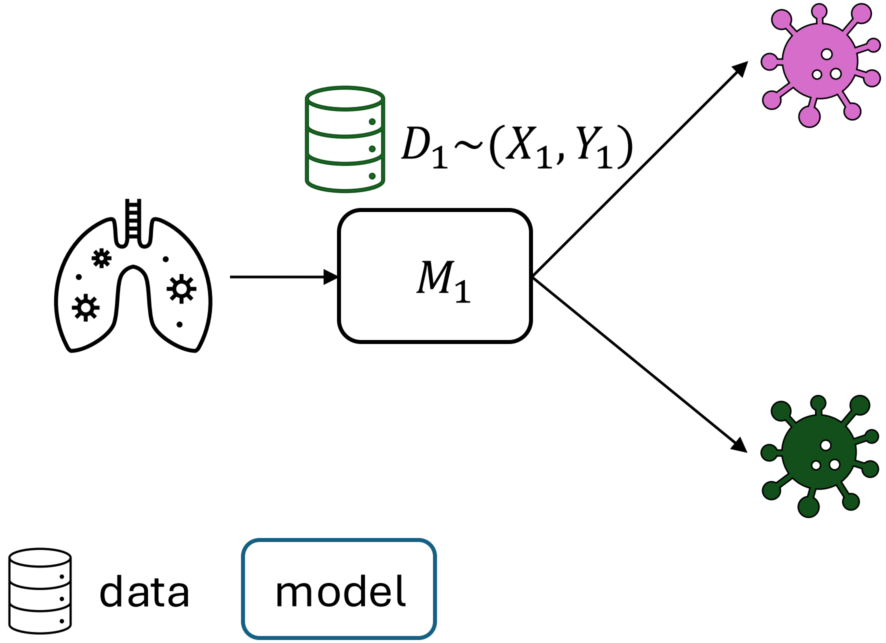
⁉️ What if… the training data comes as **a stream**? and if **the distribution changes** over time?



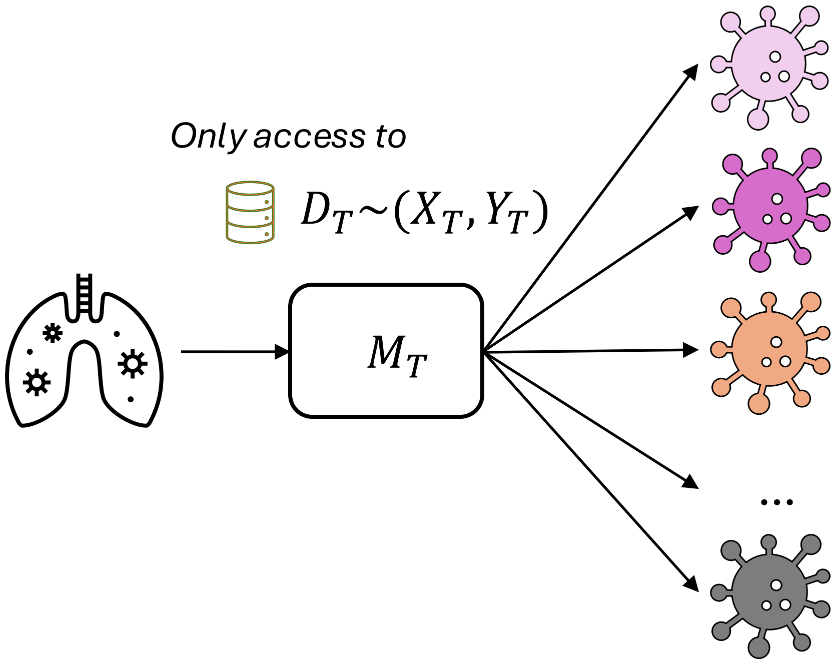


Continual Learning for Adaptive Models

Classic static supervised learning: Solve a specific task by learning from a fixed data distribution.



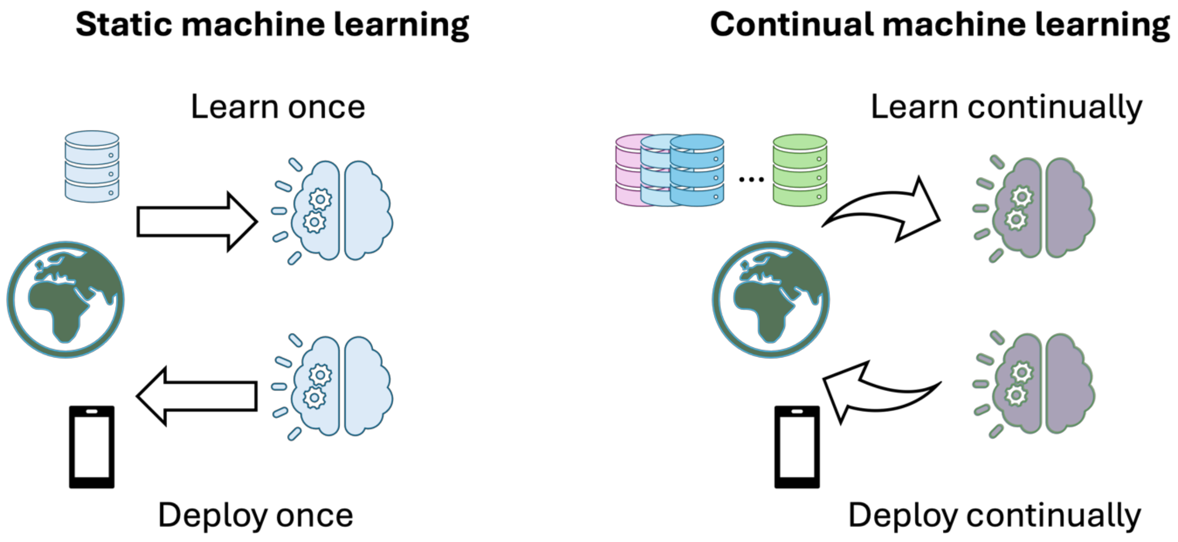
⁉️ What if… the training data comes as **a stream**? and if **the distribution changes** over time**, continuously,** and **without access to past data ?**



Continual Learning for Adaptive Models

Continual learning aims at:

- Learning **continuously and adaptively** about the external world
- Autonomously developing **more complex skills and knowledge**
- Suited for **constrained applications (storage, privacy, computation, …)**
- A more **sustainable** way of training and deploying machine learning models



(Lomonaco et al. 2020, Hayes et al. 2022)

The incremental learning framework

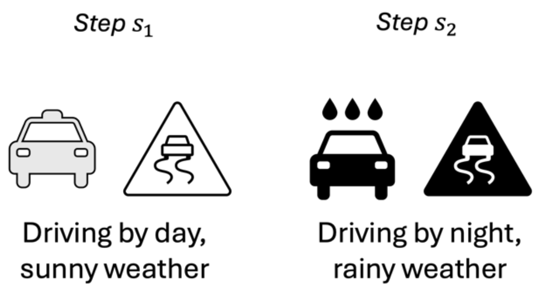
Focus on Class-Incremental Learning

Types of Incremental Learning

Input $\mathrm{x} \in \mathcal{X}$, label $\mathrm{y} \in \mathcal{Y}$,
Task identifier $\mathrm{c} \in \mathcal{C}$.

Domain-incremental learning

Learn $f : \mathcal{X} \rightarrow \mathcal{Y}$
Increasing number of domains

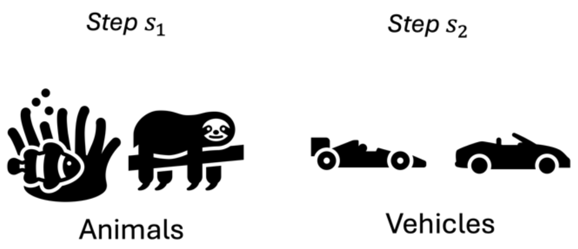


handling an increasing number of accents in an ASR system

(van de Ven et al., 2022)

Task-incremental learning

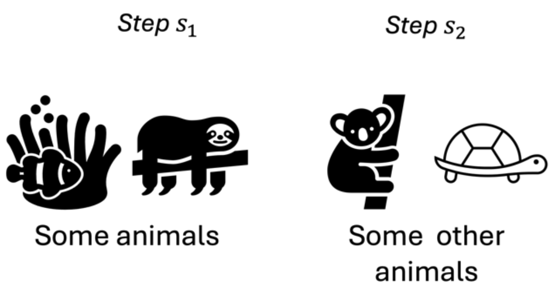
Learn $f : \mathcal{X}, \mathcal{C} \rightarrow \mathcal{Y}$
Increasing number of tasks (and classes)



learning an increasing number of tasks (intent classif. then emotion reco.)

Class-incremental learning

Learn $f : \mathcal{X} \rightarrow \mathcal{Y}$
Increasing number of classes, no task label



recognizing an increasing number of speakers

Hypotheses

- Availability of task labels ?
- Availability of class labels ?
- Task boundaries ?
- Batches of data vs true stream/online

→ Focus on supervised Class-Incremental Learning in this tutorial

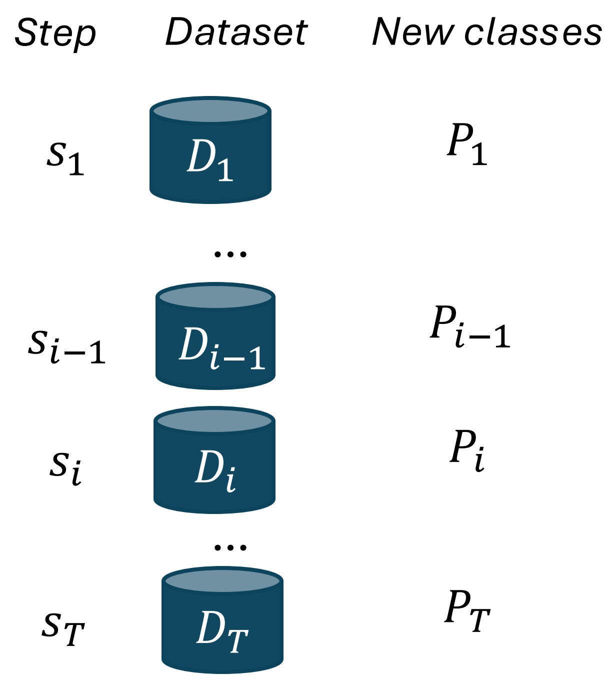
Class-Incremental Learning

<u>Notations and Hypotheses</u>

A **sequential learning process** composed of T non-overlapping learning steps $s_1, s_2, \dots, s_T$

To each step s_i is associated a subset of data D_i corresponding to a set of classes P_i. All class sets $P_1, \dots P_T$ are **disjoint**, i.e.

$$\forall\ (i,j) \in [1,T], i \neq j, P_i \cap P_j = \emptyset.$$



(Li and Hoiem, 2016; Rebuffi, 2017)

Class-Incremental Learning

Notations and Hypotheses

A **sequential learning process** composed of T non-overlapping learning steps $s_1, s_2, \dots, s_T$

To each step s_i is associated a subset of data D_i corresponding to a set of classes P_i. All class sets $P_1, \dots P_T$ are **disjoint**, i.e.

$$\forall\,(i,j) \in [1,T], i \neq j, P_i \cap P_j = \emptyset.$$

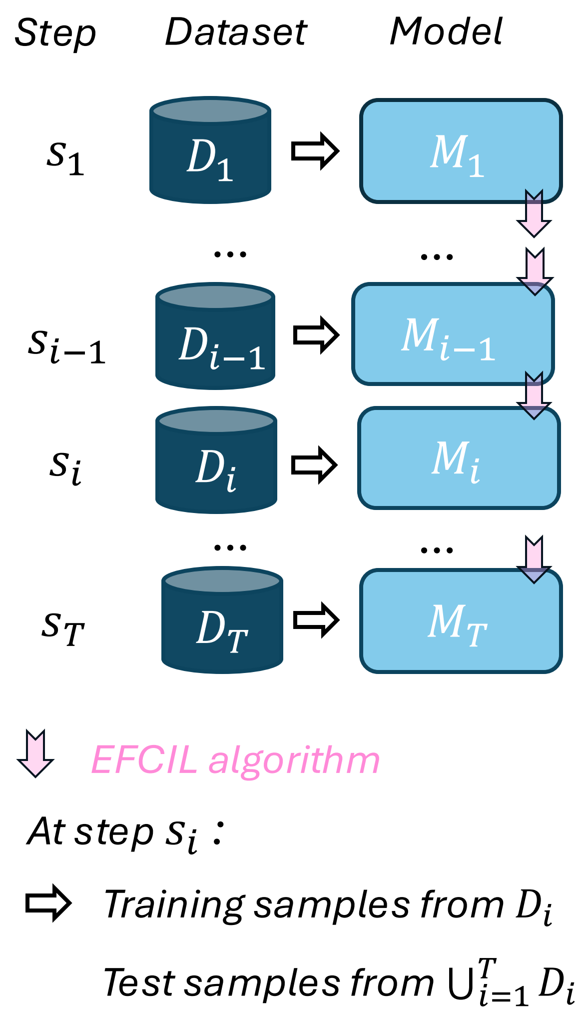
Training

At the first step s_1, train the model M_1 using the dataset D_1.

For $i = 2, 3, \dots, T$, at the step s_i, M_i first **recovers the weights** from M_{i-1} that was obtained in the previous step s_{i-1}.

Train $\boldsymbol{M_i}$ using **the examples of the dataset $\boldsymbol{D_i}$** with the objective to **recognize all the classes from $\boldsymbol{P_1 \cup P_2 \cup \cdots P_i}$**. Optionally, use a memory buffer $\boldsymbol{B_i} \subset \boldsymbol{D_1} \cup \boldsymbol{D_2} \dots \cup \boldsymbol{D_i}$ and train on $\boldsymbol{D_i} \cup \boldsymbol{B_i}$.

(Li and Hoiem, 2016; Rebuffi, 2017)



CIL – Evaluation

Average incremental accuracy $\boldsymbol{A}$

For a data stream $D = D_1 \cup D_2 \ldots \cup D_T$ composed of T batches of classes:

$$A = \frac{1}{T} \sum_{i=1}^{T} Acc(M_i, D_1 \cup D_2 \cup \ldots D_i)$$

The average of the classification accuracies of the model M_i on the **cumulated test set $\boldsymbol{D_1} \cup \boldsymbol{D_2} \cup \ldots \boldsymbol{D_i}$**.

Test set at step...

s_1	D_1					$A_{i,[1,1]}$
s_2	D_1	D_2				$A_{i,[1,2]}$
s_3	D_1	D_2	D_3			$A_{i,[1,3]}$
s_4	D_1	D_2	D_3	D_4		$A_{i,[1,4]}$
s_5	D_1	D_2	D_3	D_4	D_5	$A_{i,[1,5]}$

$$A = \frac{1}{5} \sum_{i=1}^{5} A_{i,[1,i]}$$

$$A_{i,[1,i]} = Acc(M_i, D_1 \cup D_2 \cup \ldots D_i)$$

(Rebuffi et al., 2017)

CIL – Evaluation

Average forgetting $\boldsymbol{F}$

$$F = \frac{1}{T-1}\sum_{i=1}^{T-1} \max_{i \le j \le T} Acc(M_j, D_i) - Acc(M_T, D_i)$$

The average value of the maximum accuracy drop over the incremental process **for a given subset $\boldsymbol{D_i}$.**

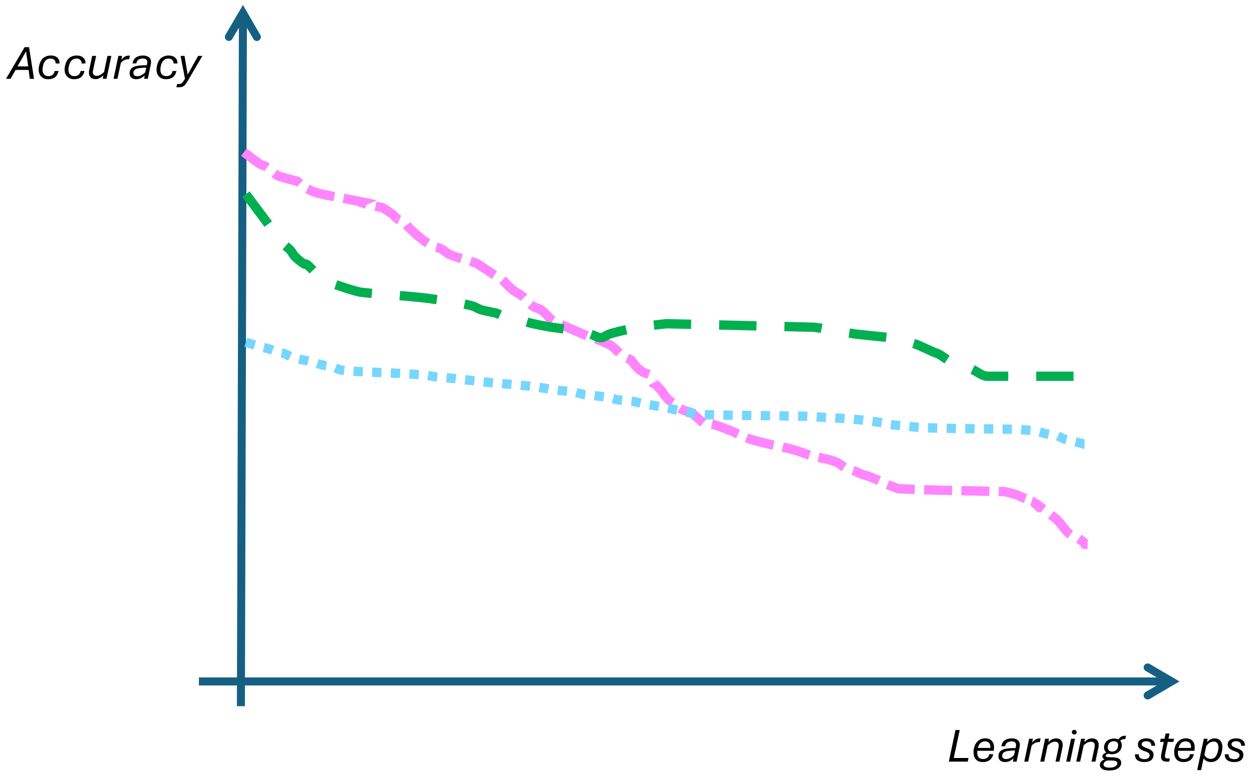
	D_1	D_2	D_3	D_4	D_5
s_1	$A_{1,1}$				
s_2	$A_{2,1}$	$A_{2,2}$			
s_3	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$		
s_4	$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$	
s_5	$A_{5,1}$	$A_{5,2}$	$A_{5,3}$	$A_{5,4}$	$A_{5,5}$

$$F = \frac{1}{4}\sum_{i=1}^{4} \max_{i \le j \le 5} Acc(M_j, D_i) - Acc(M_5, D_i)$$

(Chaudhry et al., 2018)

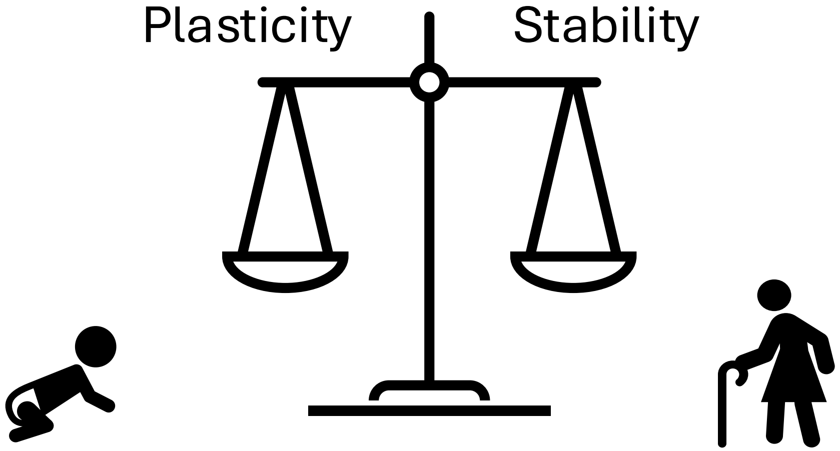
CIL – Evaluation

Complementarity of evaluating accuracy and forgetting



Stability-Plasticity trade-off

EFCIL algorithms need to balance **stability** and **plasticity**

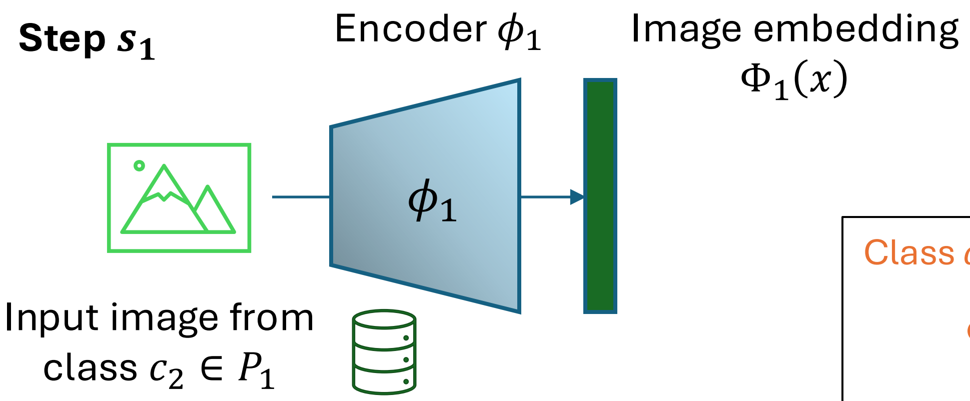


Catastrophic forgetting
(McCloskey et al. 1989)

Plasticity loss
(Dohare et al. 2024)

(Mermillod et al. 2013)

Challenges of CIL: naïve fine-tuning



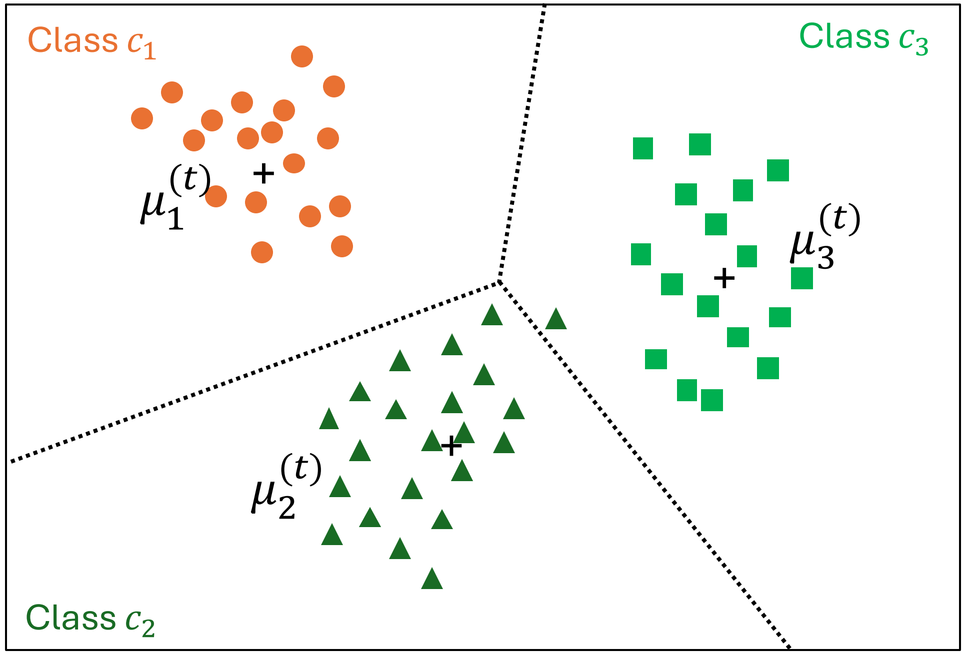
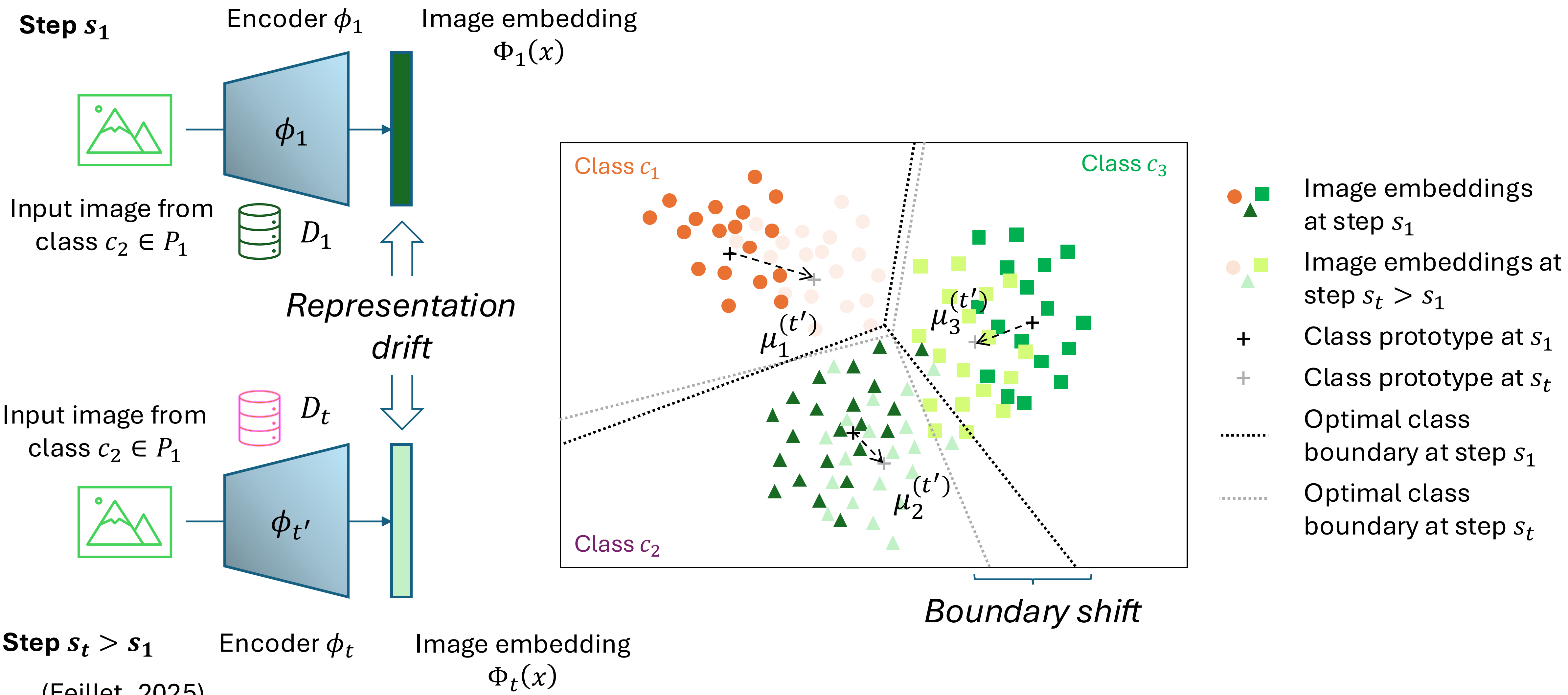


Image embeddings at step s_1

$+$ Class prototype at s_1

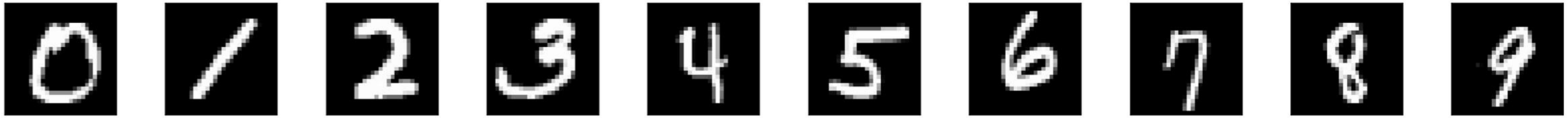
Optimal class boundary at step s_1

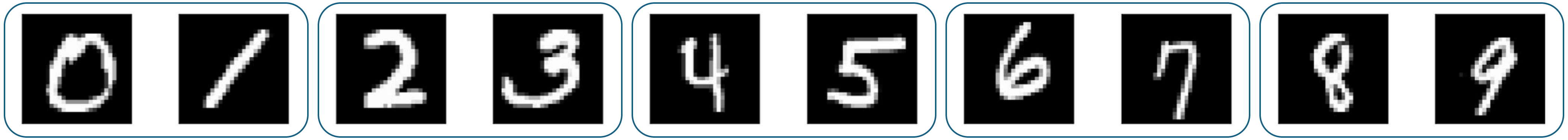
Challenges of CIL: naïve fine-tuning



(Feillet, 2025)

Illustration: MNIST 2 by 2 classes





Step s_1 **Step s_2** **Step s_3** **Step s_4** **Step s_5**

\> python joint_expe.py

\> python vanilla_expe.py

$$\boldsymbol{L_{CE}}(x) = \sum_{k=1}^{N} -\delta_{y=k} \log\big(p_k(x)\big), \ \mathrm{x} \in D$$

$$\boldsymbol{L_{CE,t}}(x) = \sum_{k=1}^{N_{1:t}} -\delta_{y=k} \log\left(p_k^t(x)\right), \ x \in D_t$$

$N_{1:t}$: number of classes seen up to step s_t

$p_k^t(x)$: softmax score of model M_t for input x and class k

An overview of CIL algorithms

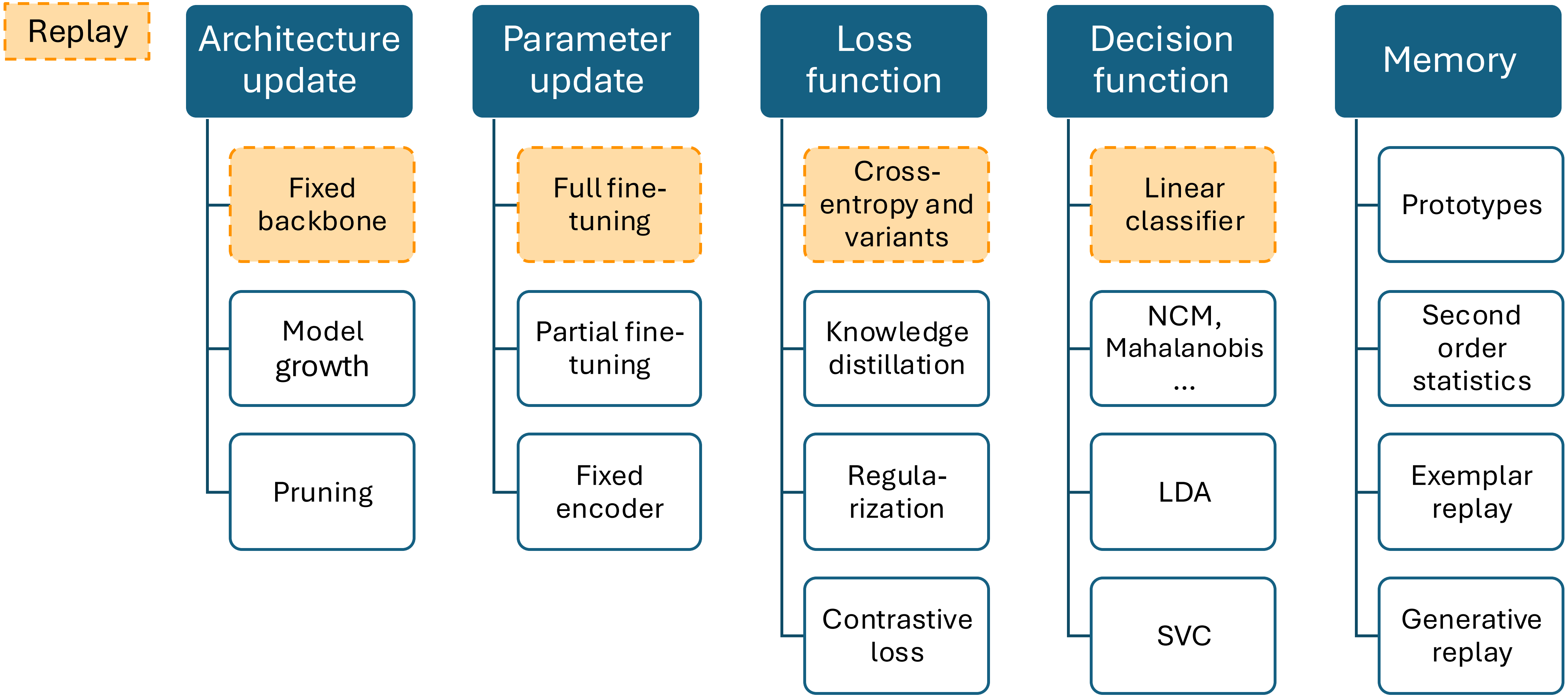
- **Architecture update**
  - Fixed backbone
  - Model growth
  - Pruning
- **Parameter update**
  - Full fine-tuning
  - Fixed encoder
  - Partial fine-tuning
- **Loss function**
  - Cross-entropy and variants
  - Knowledge distillation
  - Regula-rization
  - Contrastive loss
- **Decision function**
  - Linear classifier
  - NCM, Mahalanobis …
  - LDA
  - SVC
- **Memory**
  - Prototypes
  - Second order statistics
  - Exemplar replay
  - Generative replay

Tutorial

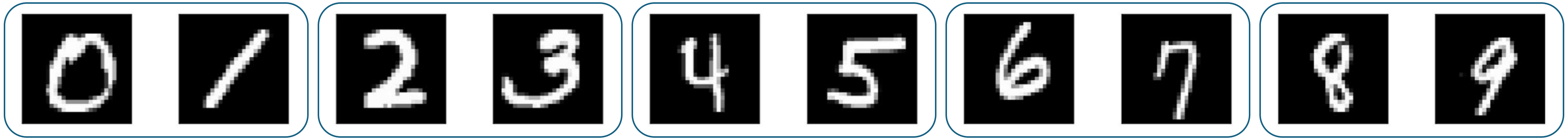
Fine-tuning based CIL methods

Methods that update the parameters of the encoder as well as the classifier

Replaying past samples



MNIST 2 by 2 classes - Replay



Step s_1 **Step s_2** **Step s_3** **Step s_4** **Step s_5**

> python replay_expe.py

$$\boldsymbol{L_{CE,t}}(x) = \sum_{k=1}^{N_{1:t}} -\delta_{y=k} \log\left(q_k^t(x)\right), \quad x \in D_t \cup B_t$$

Replaying past samples – How to select samples ?

- At random
- “Herding” (Rebuffi et al. 2017) in iCaRL : rank samples by distance to their class prototype and maintain a memory buffer of fixed size
- Compress samples : Most informative pixels, edge maps
- …

→ Use your favorite imbalanced classification method

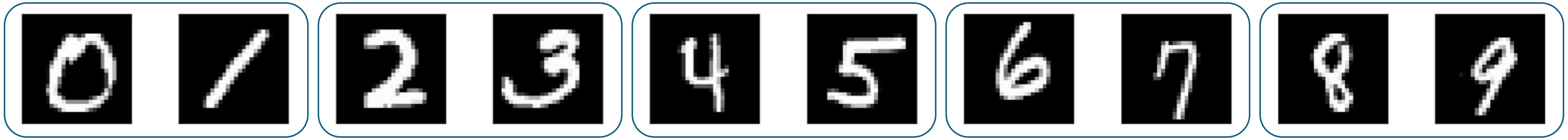
NB: to evaluate methods, choose either a fixed total number of samples in the memory buffer, or a fixed number of samples per class.

Learning without forgetting (Li and Hoeim, 2017)

LwF (Li et Hoeim, 2016)

- **Architecture update**
  - Fixed backbone
  - Model growth
  - Pruning
- **Parameter update**
  - Full fine-tuning
  - Partial fine-tuning
  - Fixed encoder
- **Loss function**
  - Cross-entropy and variants
  - Knowledge distillation
  - Regula-rization
  - Contrastive loss
- **Decision function**
  - Linear classifier
  - Support Vector Classifiers
  - NCM, Mahalanobis …
  - LDA
- **Memory**
  - Prototypes
  - Second order statistics
  - Generative replay
  - Exemplar replay

MNIST 2 by 2 classes - LwF (Li and Hoeim, 2017)



Step s_1 **Step s_2** **Step s_3** **Step s_4** **Step s_5**

> python distil_expe.py

$$L = (1-\rho)L_{CE} + \rho L_{KD} + R_\theta$$

New classes

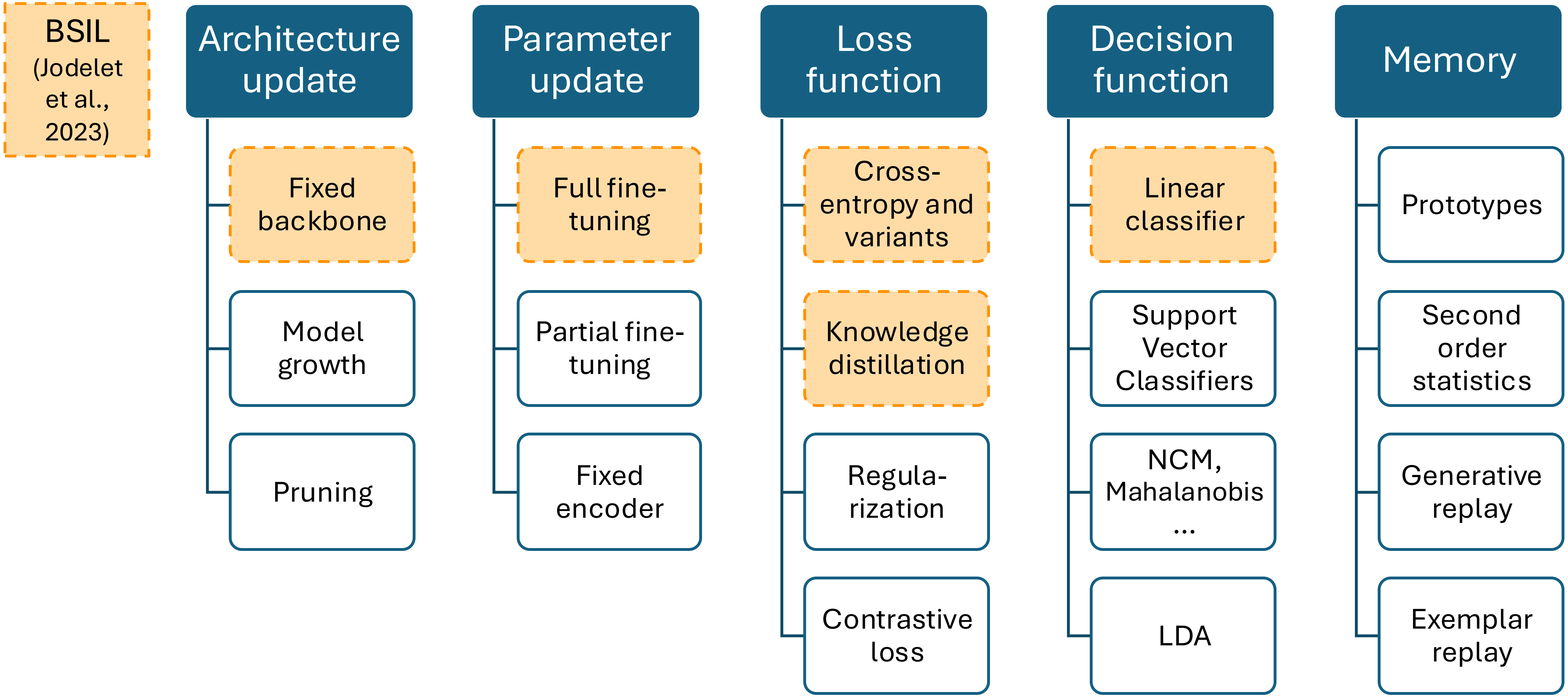
$$L_{CE}(x) = \sum_{k=N_{1:t-1}}^{N_{1:t}} -\delta_{y=k} \log\left(p_k^t(x)\right)$$

Old classes

$$L_{KD}(x) = \sum_{k=1}^{N_{1:t-1}} -p_k^{t-1}(x) \log\left(p_k^t(x)\right) \tau^2$$

https://www.nature.com/articles/s42256-022-00568-3/tables/2

Balanced Softmax for Incremental Learning (Jodelet et al. 2023)



Balanced Softmax for Incremental Learning (Jodelet et al. 2023)

Fixed architecture, continual fine-tuning, linear classifier. Training from scratch. Loss function:

$$\boldsymbol{L} = (1-\rho)\boldsymbol{L_{CE}} + \rho\boldsymbol{L_{KD}}, \qquad \rho = N_{1:t-1}/N_{1:t}$$

$$\boldsymbol{L_{CE}}(x) = \sum_{k=1}^{N_{1:t}} -\delta_{y=k} \log\left(q_k^t(x)\right)$$

Balanced cross-entropy loss proposed by (Ren et al. 2020) for long-tail classification, adapted to EFCIL by Jodelet et al. (2023).

$$\boldsymbol{L_{KD}}(x) = \sum_{k=1}^{N_{1:t-1}} -p_k^{t-1}(x) \log\left(p_k^t(x)\right) \tau^2$$

For a given input, the KD loss constrains the output of the current model to be similar to the output obtained by the previous model

$$q_k^t(x) = \frac{\lambda_k e^{z_k^t(x)}}{\sum_{i=1}^{N_{1:t}} \lambda_i e^{z_i^t(x)}} \qquad \lambda_k = \begin{cases} \epsilon > 0 \ if \ k \in [1, N_{t-1}] \\ n_k \ else \end{cases}$$

$N_{1:t}$: total number of classes encountered from learning steps s_1 to s_t.
ϵ : small positive value.
n_k: number of training samples of class k.

CIL with a fixed encoder

Methods that update only the classifier

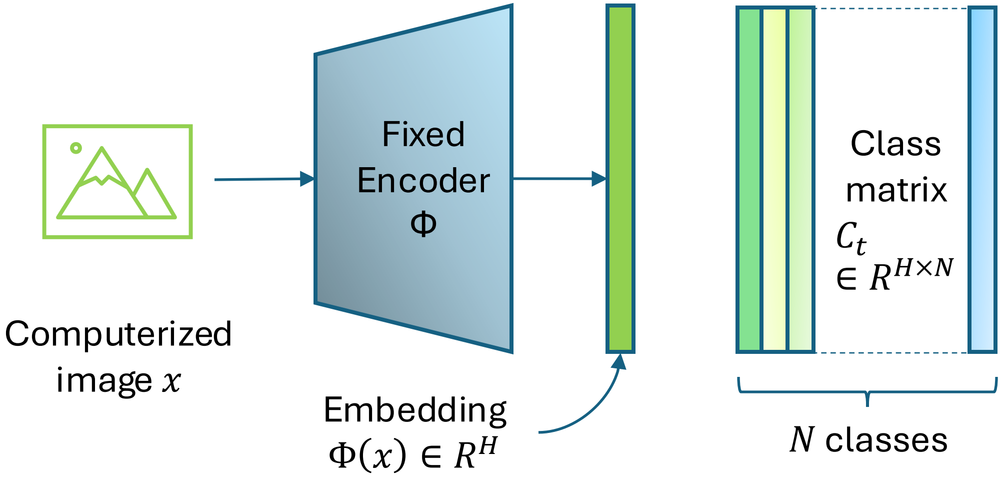
Nearest Class Mean (Rebuffi et al. 2017)

NCM (Rebuffi et al., 2017)

- **Architecture update**
  - Fixed backbone
  - Model growth
  - Pruning
- **Parameter update**
  - Full fine-tuning
  - Partial fine-tuning
  - Fixed encoder
- **Loss function**
  - Cross-entropy and variants
  - Knowledge distillation
  - Regula-rization
  - Contrastive loss
- **Decision function**
  - Linear classifier
  - L2 or cosine distance
  - LDA
  - SVC
- **Memory**
  - Prototypes
  - Second order statistics
  - Exemplar replay
  - Generative replay

Nearest Class Mean (Rebuffi et al. 2017)

Fixed architecture and encoder, prototype-based classifier.



$$y_{pred} = \underset{1 \le c \le N}{\operatorname{argmin}} \, dist(\mu_c, \phi(x))$$

NB: Can be a strong baseline with a pre-trained encoder (Ostapenko et al. 2022)

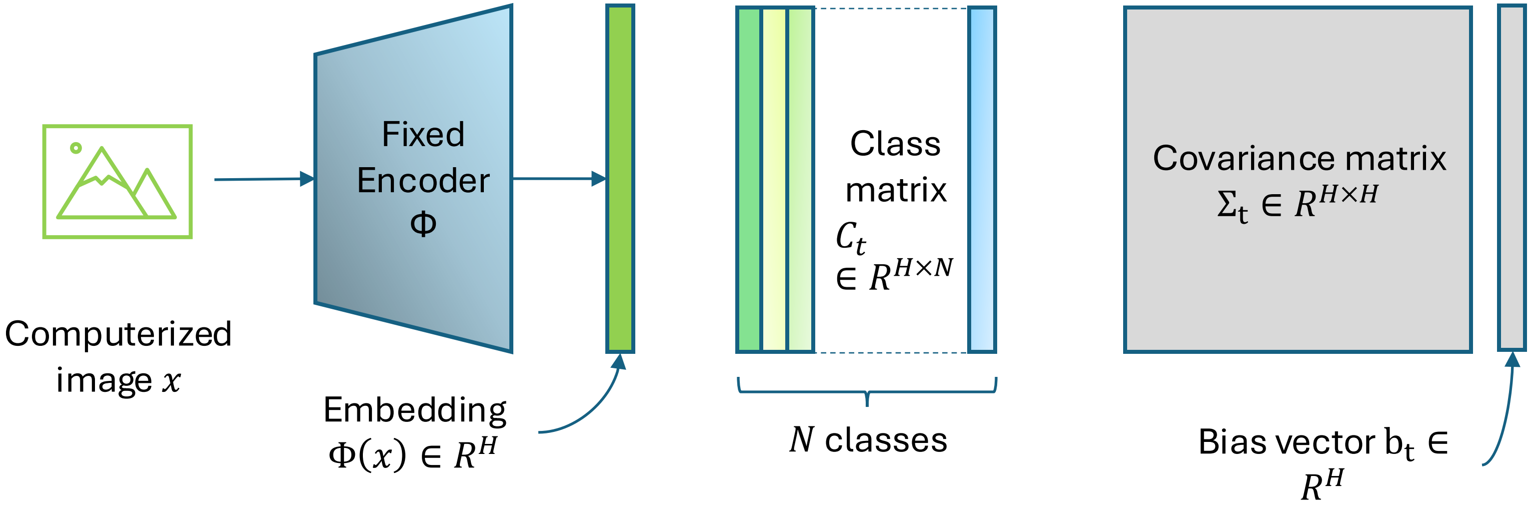
Deep Streaming LDA (Hayes et al., 2020)

DSLDA (Hayes et al., 2020)

- **Architecture update**
  - Fixed backbone
  - (Model growth)
  - (Pruning)
- **Parameter update**
  - Full fine-tuning
  - Partial fine-tuning
  - Fixed encoder
- **Loss function**
  - Cross-entropy and variants
  - Knowledge distillation
  - Regularization
  - Contrastive loss
- **Decision function**
  - Linear classifier
  - NCM, Mahalanobis …
  - LDA
  - SVC
- **Memory**
  - Prototypes
  - Second order statistics
  - (Generative replay)
  - (Exemplar replay)

Deep Streaming LDA (Hayes et al., 2020)

Fixed architecture and encoder, prototype-based classifier.



$$y_{pred} = \underset{1\le c\le N}{\operatorname{argmin}}\, \Lambda\mu_c + b_c$$

With $\Lambda = [(1-\epsilon)\Sigma_t + \epsilon I]^{-1}$

(precision matrix)

$$b_c = -\frac{1}{2}(\mu_c \cdot \mu_c \Lambda)$$

(bias vector)

FeCAM (Goswami et al., 2024)

FeCAM (Goswami et al., 2024)

Architecture update
- Fixed backbone
- Model growth
- Pruning

Parameter update
- Full fine-tuning
- Partial fine-tuning
- Fixed encoder

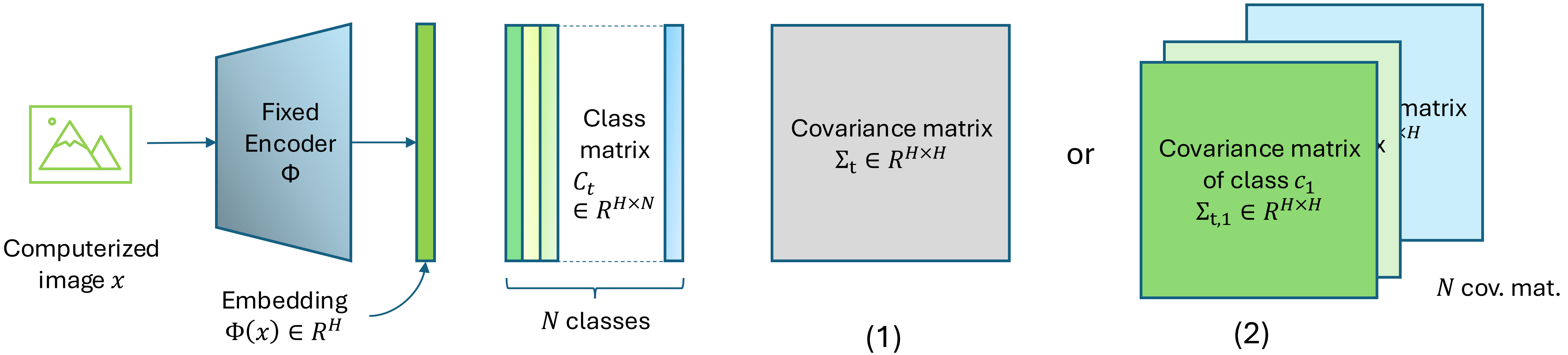
Loss function
- Cross-entropy and variants
- Knowledge distillation
- Regularizati on
- Contrastive loss

Decision function
- Linear classifier
- NCM, Mahalanobis …
- LDA
- SVC

Memory
- Prototypes
- Second order statistics
- Generative replay
- Exemplar replay

FeCAM (Goswami et al., 2024)

Fixed architecture and encoder, prototype-based classifier.



$$(1)\ y_{pred} = \underset{1 \le c \le N}{\operatorname{argmin}} (\phi(x) - \mu_c)^\intercal\ \Sigma_t^{-1} (\phi(x) - \mu_c)$$

$$(2)\ y_{pred} = \underset{1 \le c \le N}{\operatorname{argmin}} (\phi(x) - \mu_c)^\intercal\ \overline{\Sigma_{t,c}^{-1}} (\phi(x) - \mu_c)$$

NB: a single covariance matrix is preferable if few samples per class are available

Further challenges of CIL

Impact of the incremental learning scenario

When CIL algorithms are tested in different incremental settings, no method outperforms all others

Scenario (a): memory budget of 1.5M parameters, 20 steps x 5 classes each.

Scenario (b): memory budget of 3.0M parameters, 4 steps x 25 classes each.

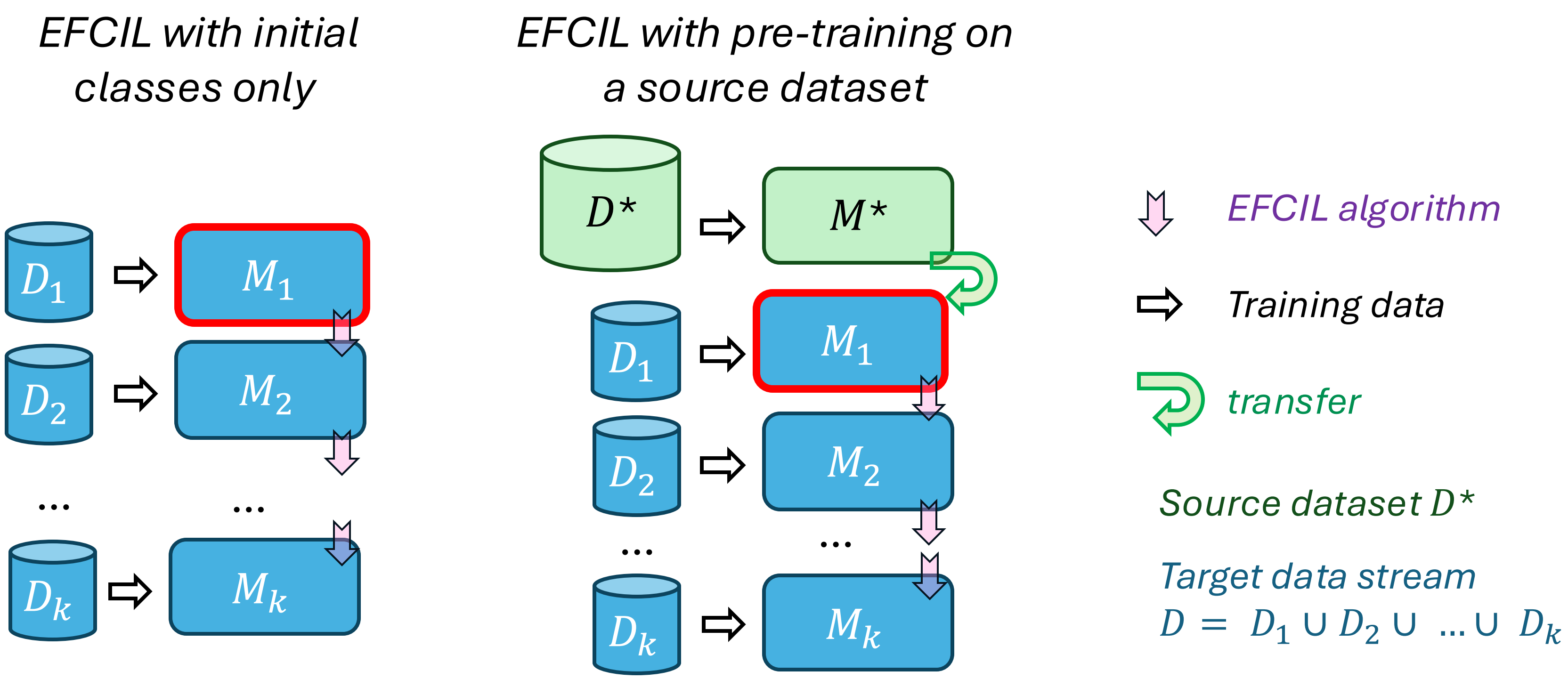
Classification performance in percent for various combinations of CIL algorithm and backbone network, averaged over five reference datasets containing 100 classes each in total. ***Best combination for each scenario is highlighted in purple.***

⇒ Need for a recommendation method to select **the best combination of CIL algorithm and backbone network depending on the scenario**.

(Illustration from ‘AdvisIL,’ Feillet et al. 2022) (surveys: Belouadah et al. 2021, Masana et al. 2022)

Impact of the initial training strategy

Different EFCIL methods may train the initial model differently: how does it impact performance?



(Petit, Soumm, Feillet et al. 2024)

Modeling causal effects

Goal: **Identify the primary factors that influence the performance of EFCIL algorithms.**

Method: A statistical analysis using linear regressions (Ordinary Least Squares framework) to model EFCIL performance metrics as a function of the experimental settings e.g.

$$\boldsymbol{\overline{Acc} = \beta_0 + \beta_1 Train + \beta_2 Incr + \beta_3 Data + \cdots + \epsilon}$$

→ Short notation: $\boldsymbol{\overline{Acc} \sim Train + Incr + Data}$

- $\overline{Acc}$: avg incr acc
- F: average forgetting
- $Train$: initial training strategy
- $Incr$: EFCIL algo
- $Data$: data stream
- Acc_1 : initial accuracy

Target/endogenous variables

Explanatory/exogenous variables

(Petit, Soumm, Feillet et al. 2024)

Modeling causal effects

Key findings

- the most significant factor affecting the average incremental accuracy $\overline{Acc}$ is the choice of initial training strategy $Train$.
- Upon controlling the impact of initial accuracy Acc_1, the selected incremental algorithm $Incr$ has a greater importance.
- Regarding forgetting F, the incremental algorithm $Incr$ is the most influential factor.

Choosing the right initial model is highly important for the accuracy of EFCIL models.
The EFCIL algorithm mostly impacts the stability of the performance (ability to retain previous knowledge while integrating new knowledge).

Model	R^2	variable	η^2
$\overline{Acc} \sim Incr + Train + Data$	0.69	$Train$	0.32
		$Data$	0.24
		$Incr$	0.11
$\overline{Acc} \sim Acc_1 + Incr + Train + Data$	0.81	Acc_1	0.25
		$Incr$	0.22
		$Train$	0.10
		$Data$	0.06
$F \sim Incr + Train + Data$	0.71	$Incr$	0.61
		$Train$	0.06
		$Data$	0.03

ANOVA results for each considered regression. Variables are significant at p < 0.05 and ordered by decreasing importance.

(Petit, Soumm, Feillet et al. 2024)

Conclusion

Recap

- **Architecture update**
  - Fixed backbone
  - Model growth
  - Pruning
- **Parameter update**
  - Full fine-tuning
  - Fixed encoder
  - Partial fine-tuning
- **Loss function**
  - Cross-entropy and variants
  - Knowledge distillation
  - Regula-rization
  - Contrastive loss
- **Decision function**
  - Linear classifier
  - NCM, Mahalanobis …
  - LDA
  - SVC
- **Memory**
  - Prototypes
  - Second order statistics
  - Exemplar replay
  - Generative replay

See list of references in appendix

Perspectives

- Back to frugality: focus less on memory and more on compute
- Synergies with domain adaptation, online learning / shallow methods, novelty detection, few-shot CIL
- Explainability tools to track forgetting
- Continual learning for fondation models

Hands-on : visit
https://github.com/EvaJF/continual_tuto

Appendix

References

Belouadah, E., Popescu, A., and Kanellos, I. (2021). A comprehensive study of class incremental learning algorithms for visual tasks. Neural Networks, 135:38–54.

Bucilua, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. Proceedings of the 12 th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 3.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. IEEE transactions on pattern analysis and machine intelligence, 44(7):3366–3385.

Dohare, S., Sutton, R. S., and Mahmood, A. R. (2021). Continual backprop: Stochastic gradient descent with persistent randomness. arXiv preprint arXiv:2108.06325.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. (2020). Podnet: Pooled outputs distillation for small-tasks incremental learning. In Computer vision-ECCV 2020-16th European conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX, volume 12365, pages 86–102. Springer.

Feillet, E., Petit, G., Popescu, A., Reyboz, M., and Hudelot, C. (2023). Advisil - a class-incremental learning advisor. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2400–2409.

References

Feillet, Eva, Adrian Popescu, and Céline Hudelot. "A Reality Check on Pre-training for Exemplar-free Class-Incremental Learning." 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE, 2025.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. Trends in cognitive sciences, 3(4):128–135.

Goswami, D., Liu, Y., Twardowski, B., and van de Weijer, J. (2024). Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. Advances in Neural Information Processing Systems, 36.

Hayes, T. L. and Kanan, C. (2020). Lifelong machine learning with deep streaming linear discriminant analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 220–221.

Hinton, G. E., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. CoRR, abs/1503.02531.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 831–839.

References

Jodelet, Q., Liu, X., and Murata, T. (2022). Balanced softmax cross-entropy for incremental learning with and without memory. Computer Vision and Image Understanding, 225:103582.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13):3521–3526.

Lange, M. D., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G. G., and Tuytelaars, T. (2019). Continual learning: A comparative study on how to defy forgetting in classification tasks. CoRR, abs/1909.08383.

Li, Z. and Hoiem, D. (2016). Learning without forgetting. In European Conference on Computer Vision, ECCV.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and Van De Weijer, J. (2022). Class-incremental learning: survey and performance evaluation on image classification. IEEE TPAMI, 45(5):5513–5533.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. The Psychology of Learning and Motivation, 24:104–169.

References

McDonnell, M. D., Gong, D., Parvaneh, A., Abbasnejad, E., and van den Hengel, A. (2024). Ranpac: Random projections and pre-trained models for continual learning. Advances in Neural Information Processing Systems, 36.

Mermillod, M., Bugaiska, A., and Bonin, P. (2013). The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. Frontiers in Psychology, 4:504–504.

Ostapenko, O., Lesort, T., Rodriguez, P., Arefin, M. R., Douillard, A., Rish, I., and Charlin, L. (2022).

Continual learning with foundation models: An empirical study of latent replay. In Chandar, S., Pascanu, R., and Precup, D., editors, Proceedings of The 1st Conference on Lifelong Learning Agents, volume 199 of Proceedings of Machine Learning Research, pages 60–91. PMLR.

Petit, G., Soumm, M., Feillet, E., et al. (2024) "An analysis of initial training strategies for exemplar-free class-incremental learning." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.

Petit, G., Popescu, A., Schindler, H., Picard, D., and Delezoide, B. (2023). Fetril: Feature translation for exemplar-free class-incremental learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.

References

Prabhu, A., Torr, P. H., and Dokania, P. K. (2020). Gdumb: A simple approach that questions our progress in continual learning. In European Conference on Computer Vision, pages 524–540. Springer.

Rebuffi, S., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In Conference on Computer Vision and Pattern Recognition, CVPR.

Ring, M. B. (1997). Child: A first step towards continual learning. Machine Learning, 28(1):77–104.

Smith, J. S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., and Kira, Z. (2023a). Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11909–11919.

Thrun, S. (1995). A lifelong learning perspective for mobile robot control. In Intelligent robots and systems, pages 201–214. Elsevier.

References

van de Ven, G. M., Tuytelaars, T., and Tolias, A. S. (2022). Three types of incremental learning. Nature Machine Intelligence, 4(12):1185–1197.

Verwimp, Eli, et al. "Continual Learning: Applications and the Road Forward." Transactions on Machine Learning Research (2024).

Wang, Q.-W., Zhou, D.-W., Zhang, Y.-K., Zhan, D.-C., and Ye, H.-J. (2024). Few-shot class-incremental learning via training-free prototype calibration. Advances in Neural Information Processing Systems, 36.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. (2022). Learning to prompt for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 139–149.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. (2019). Large scale incremental learning. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 374–382.