Continual Learning with Deep Architectures

Tutorial @ ICML 2021

Part 1: Introduction and State-of-the-Art

Irina Rish
University of Montreal & Mila
irina.rish@mila.quebec

Irina Rish

Associate Professor @
University of Montreal

Core member @ Mila -
Quebec AI Institute

Canada Excellence Research
Chair (CERC) in Autonomous AI

CIFAR chair

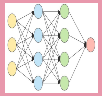
Outline

01

Motivation & History

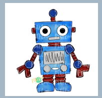
02

Inspirations from Neuroscience



03

Supervised Continual Learning



04

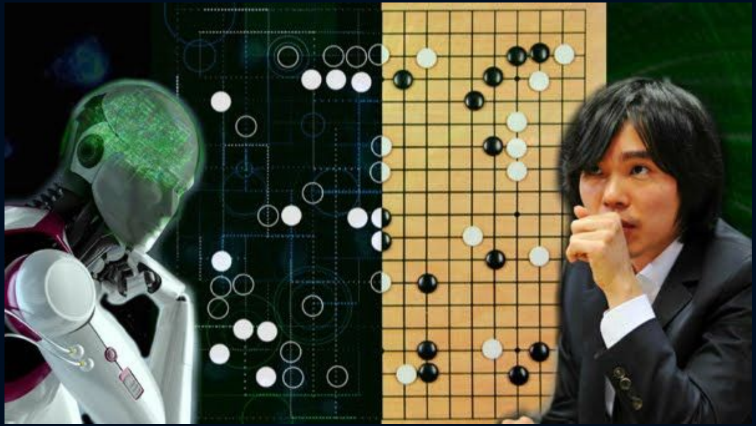
Continual Reinforcement Learning

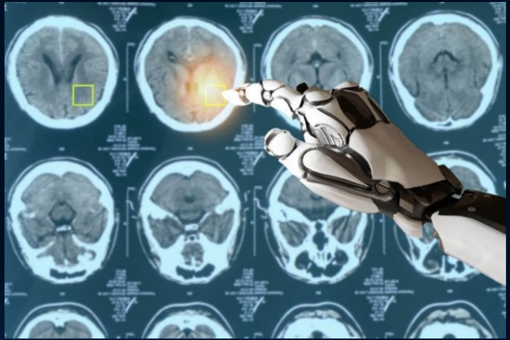
05

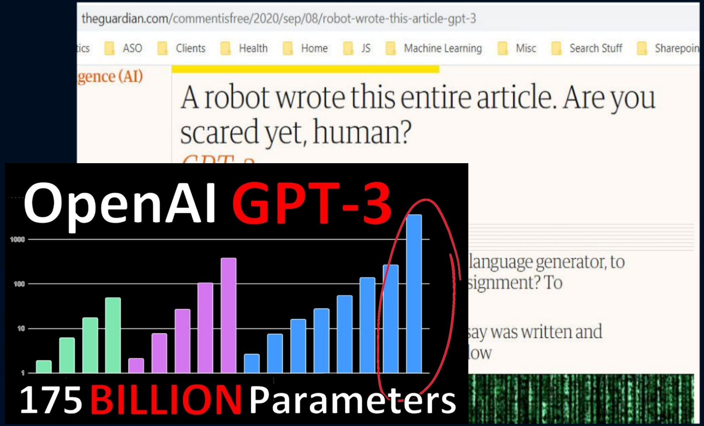
Summary

Motivation & History

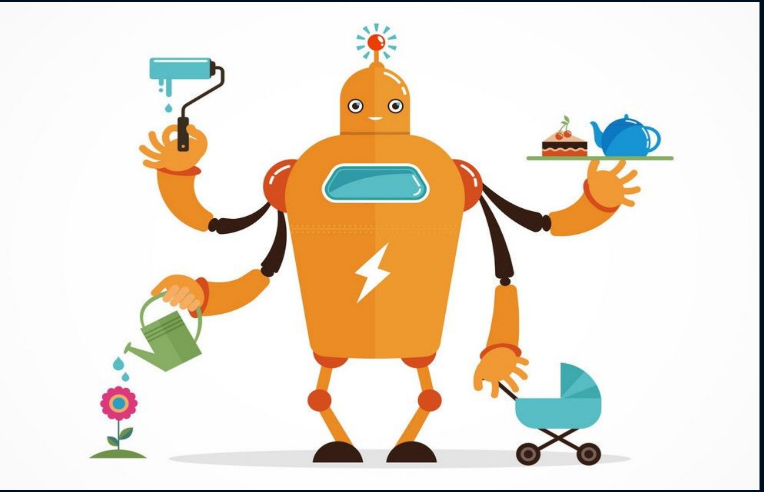
AI Today: Impressive… but (Still) "Narrow"





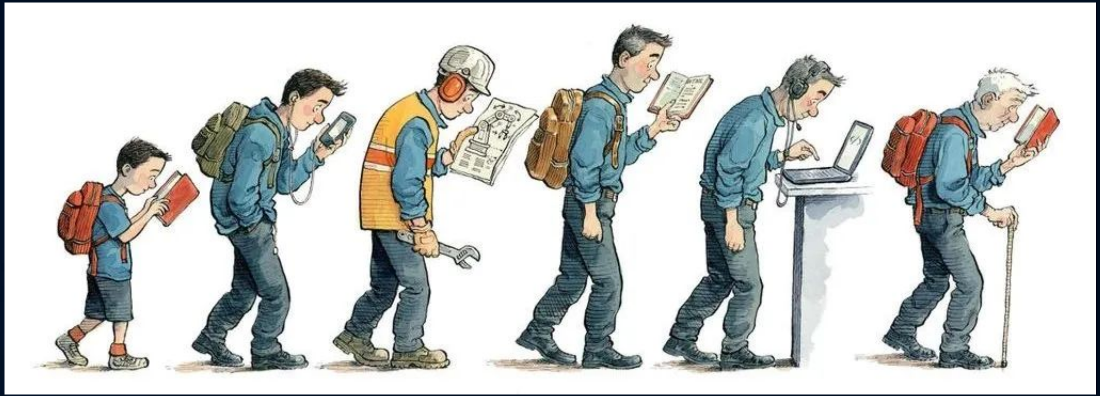


Human-Level AI: "Broad" – Versatile, Multi-Task



How Do We Achieve This?

Lifelong, Continual Learning

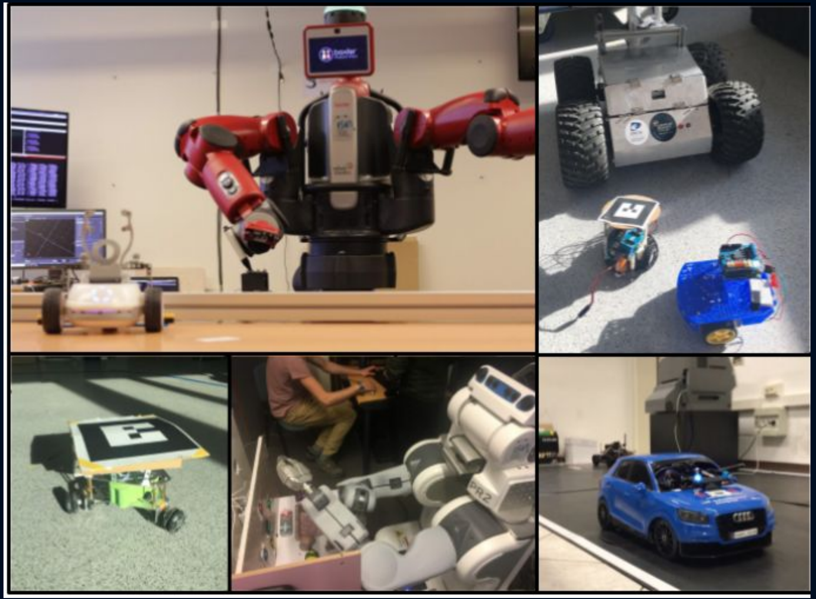


"Continual learning is the constant development of increasingly complex behaviors; the process of building more complicated skills on top of those already developed."

Ring (1997). CHILD: A First Step Towards Continual Learning.

Continual Learning Challenges in Practical Applications

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.

Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D. and Díaz-Rodríguez, N., Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 2020.

Continual Learning Challenges in Practical Applications

A self-driving car adapting to different environments (from a country road to a highway to a city)

Continual Learning Challenges in Practical Applications

Conversational agents adapting to different users, situations, tasks

Continual Learning Challenges in Practical Applications

Medical applications:

adapting to new patients, hos

conditions

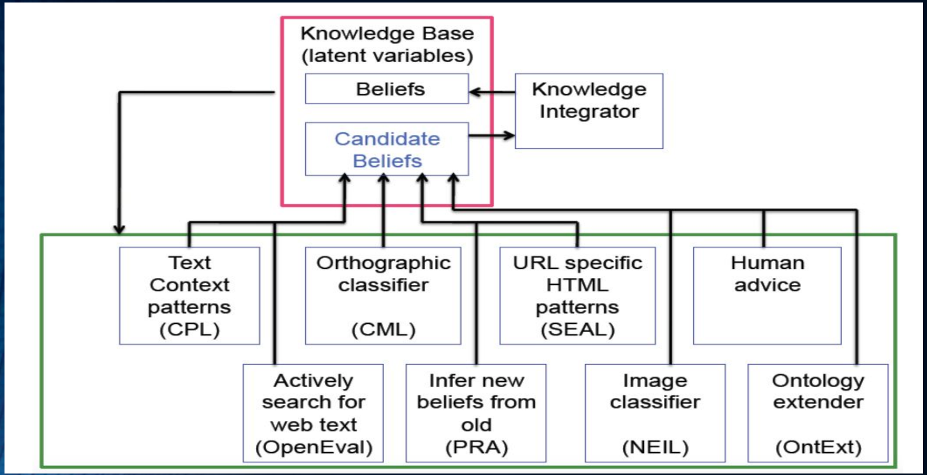
Continual Learning Challenges in Practical Applications

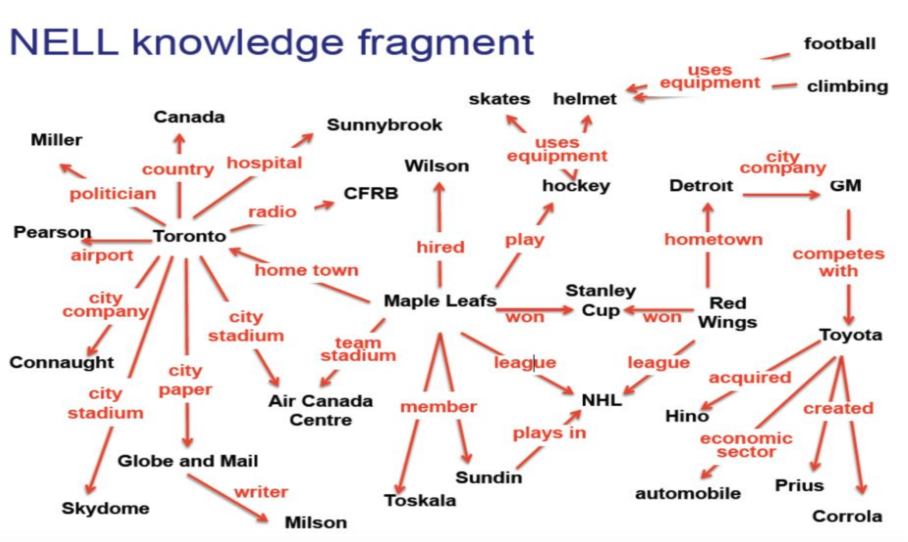
Multi-game environments

(e.g. OpenAI gym)

Example: Never-Ending Language Learner (NELL)

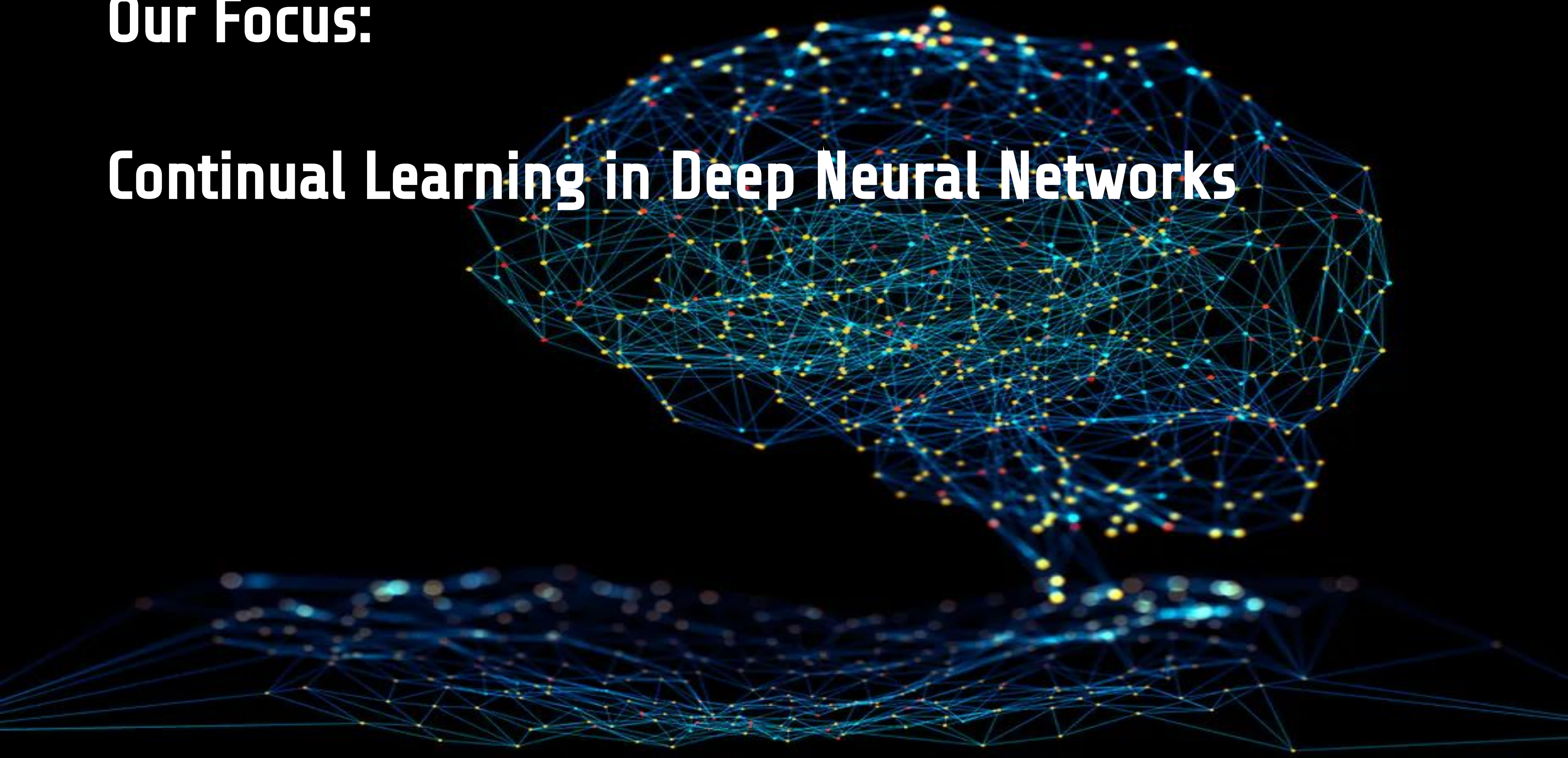
Mitchell et al, Never-Ending Learning, AAAI-2015 http://rtw.ml.cmu.edu/rtw/





Our Focus:

Continual Learning in Deep Neural Networks

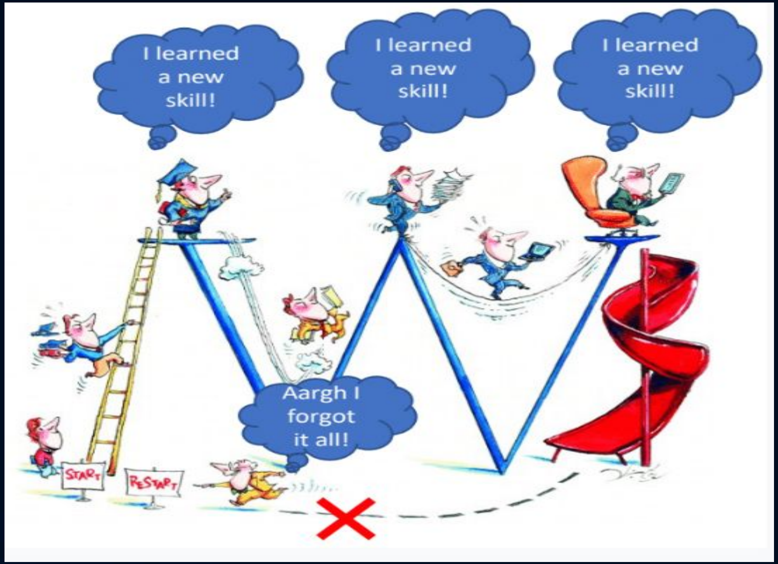


Challenge: Catastrophic Forgetting (McCloskey and Cohen, 1989)

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

CF was identified by (McCloskey and Cohen, 1989):

- A neural net trained with backprop learned a set of "one's addition facts" (i.e., the 17 sums 1+1 through 9+1 and 1+2 through 1+9)
- Then the network learned the 17 "two's addition facts" (2+1 through 2+9, 1+2 through 9+2).
- Within 1-5 two's learning trials, accuracy on task 1 had dropped from 100% to 20%, in 5 more trials, to 1%; by 15 trials, to 0%.



McCloskey, M. and Cohen, N.J., 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation* (Vol. 24, pp. 109-165). Academic Press.

French, R.M., 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, *3*(4).

More Generally: Stability vs Plasticity (Carpenter and Grossberg, 1987)

Plasticity ⇔ ability to adapt to a new task

Stability ⇔ ability to retain the learned skills on the old tasks

"Catastrophic interference is a radical manifestation of a more general problem for connectionist models of memory — in fact, for any model of memory — the so-called "stability-plasticity" problem [1,2]. The problem is how to design a system that is simultaneously sensitive to, but not radically disrupted by, new input." (French, 1999)

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

Brief History: CL in Neural Networks

- **1949** — Sensitivity-Stability (Hebb)
- **1987** — Stability-Plasticity (Carpenter & Grossberg)
- **1990** — Catastrophic forgetting (McCloskey & Cohen; Ratcliff)
- **1991** — Semi-distributed representations (French)
- **1993** — Pre-training (McRae & Hetherington)
- **1995** — Rehearsal (replay) & pseudo-rehearsal (Robins)
- **1995** — Lifelong robot learning (Thrun & Mitchell)
- **1997** — Pseudo-recurrent ("dual") networks a la neocortex/ hippocampus (French)
- **1999** — Survey on "Catastrophic forgetting in connectionist networks" (French)
- **2013** — "An empirical investigation of catastrophic forgetting in gradient-based neural networks" (Goodfellow et al.)

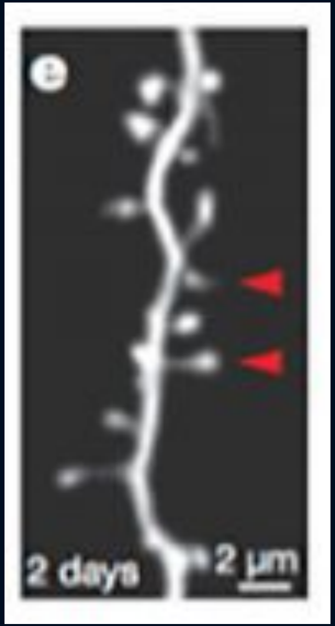
Recent work on CL

Inspirations from Neuroscience

Synaptic Plasticity Regulation for Retaining Knowledge

Learning -> enlargement of (some) dendritic spines -> decreased plasticity of the corresponding synapses. (Cichon&Gan, 2015; Yang et al., 2009)

Persistent change (months), despite learning new tasks. If these changes removed via synaptic optogenetics, the task is forgotten (Hayashi-Takagi et al., 2015).



Two-photon data (structural imaging)

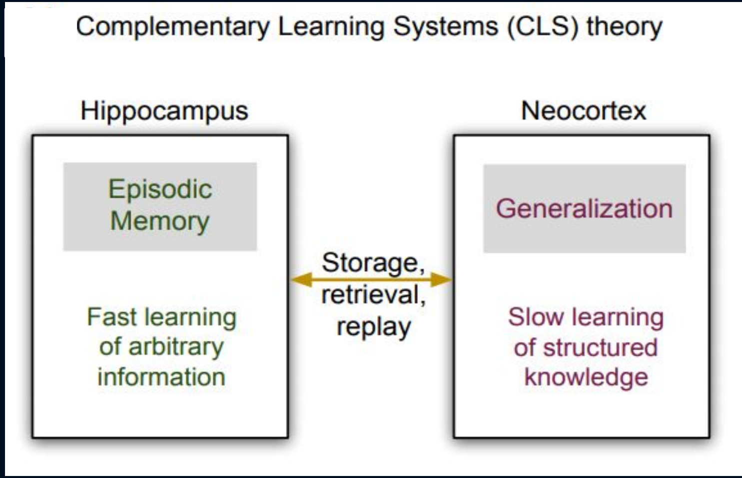
Hassabis et al (2017). Neuroscience-inspired artificial intelligence.

An inspiration for Elastic Weight Consolidation (EWC), Synaptic Intelligence (SI) and other regularization methods for preserving task-important weights.

Complementary Learning Systems and Experience Replay

CLS theory (McClelland et al. 1995):

- hippocampus: fast (one-shot) learning of episodic information, consolidated to the neocortex in sleep (or resting periods) via “replay” of neural activity patterns associated with the episode
- neocortex: slow learning of structured knowledge; efficient representation for generalization.



Parisi et al. (2019) Continual lifelong learning with neural networks: A review.

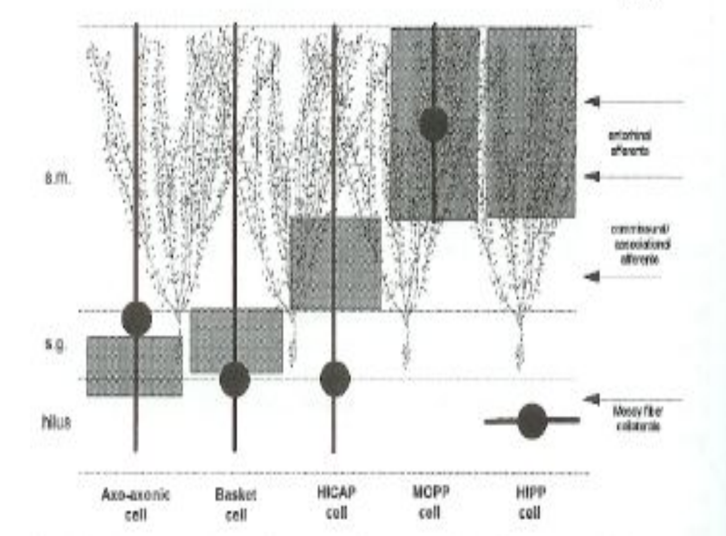
An inspiration for rehearsal/pseudo-rehearsal (Robins, 1995), pseudo-recurrent (“dual”) networks (French, 1997) and many modern experience replay methods.

Structural Plasticity via Neurogenesis

Adult neurogenesis: generation of new neurons in adult brains throughout life, balanced by death of unused neurons (“use it or lose it”)

In humans, it occurs primarily in the dentate gyrus of the hippocampus

Increased neurogenesis is associated with better adaptation to new environments.



An inspiration for adaptive, expanding neural architecture methods.

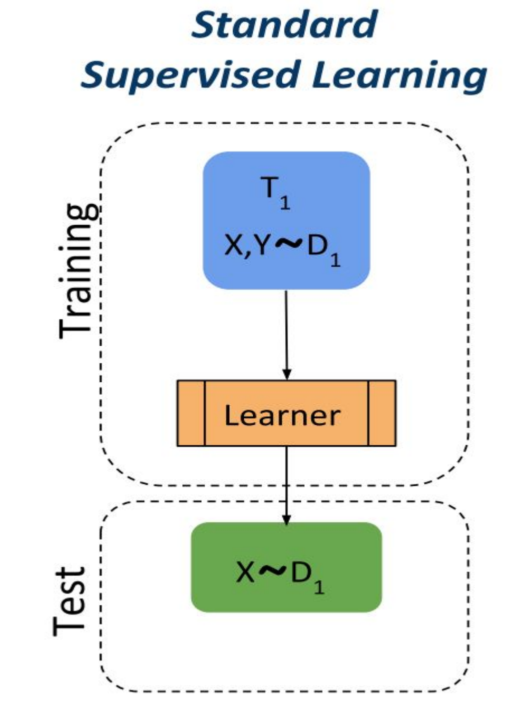
Supervised Continual Learning

Supervised Continual Learning

Non-stationary data comes one example at a time in a stream:

$(x_1, y_1, t_1), \ldots, (x_i, y_i, t_i), \ldots, (x_{i+j}, y_{i+j}, t_{i+j})$

Our data is *locally i.i.d.* – samples for a task are drawn from the same unknown joint probability distribution $x_i, y_i \sim P_t(x, y)$.



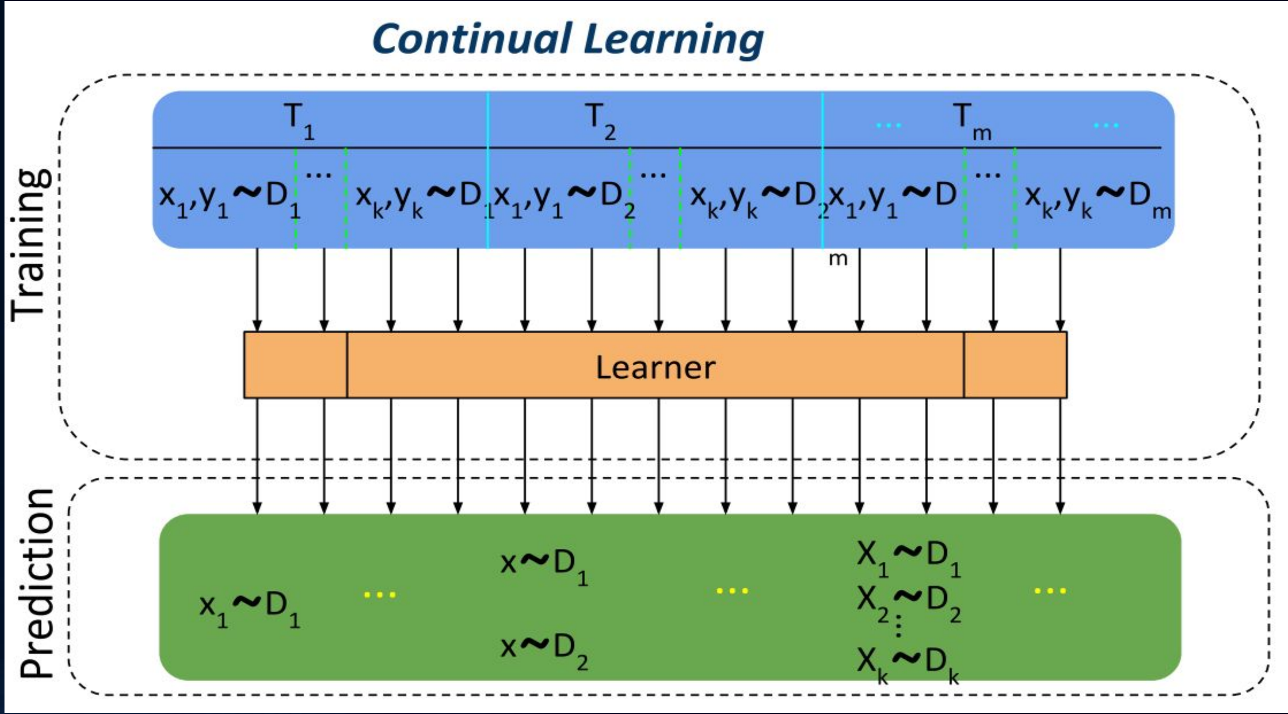
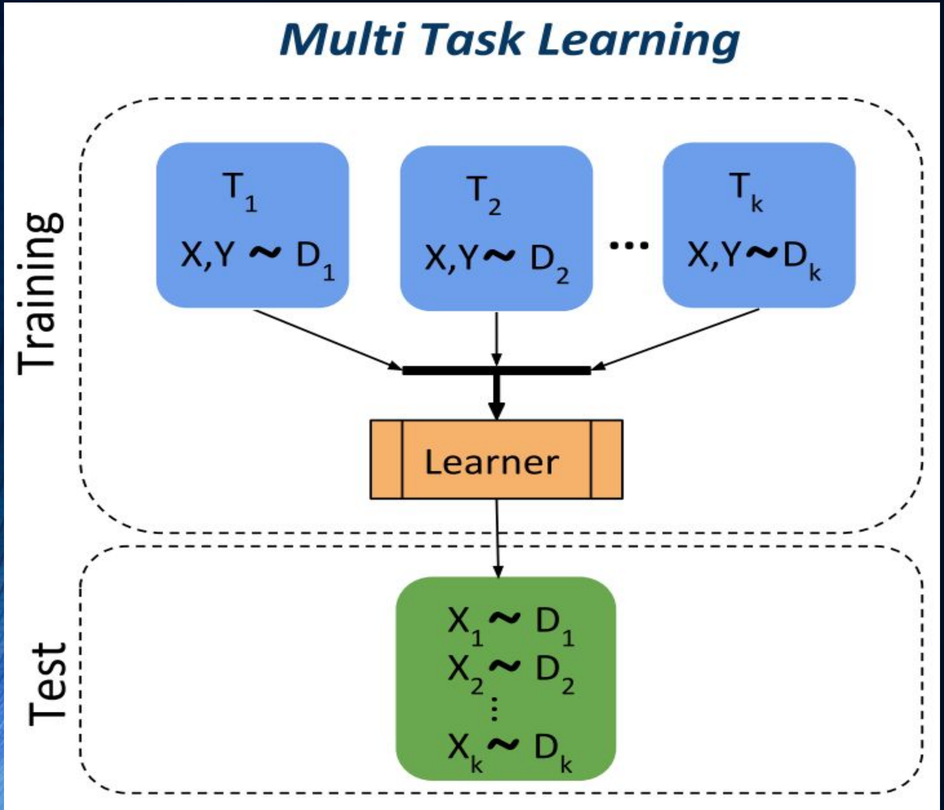


Image credit: de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Continual vs Multi-Task Learning



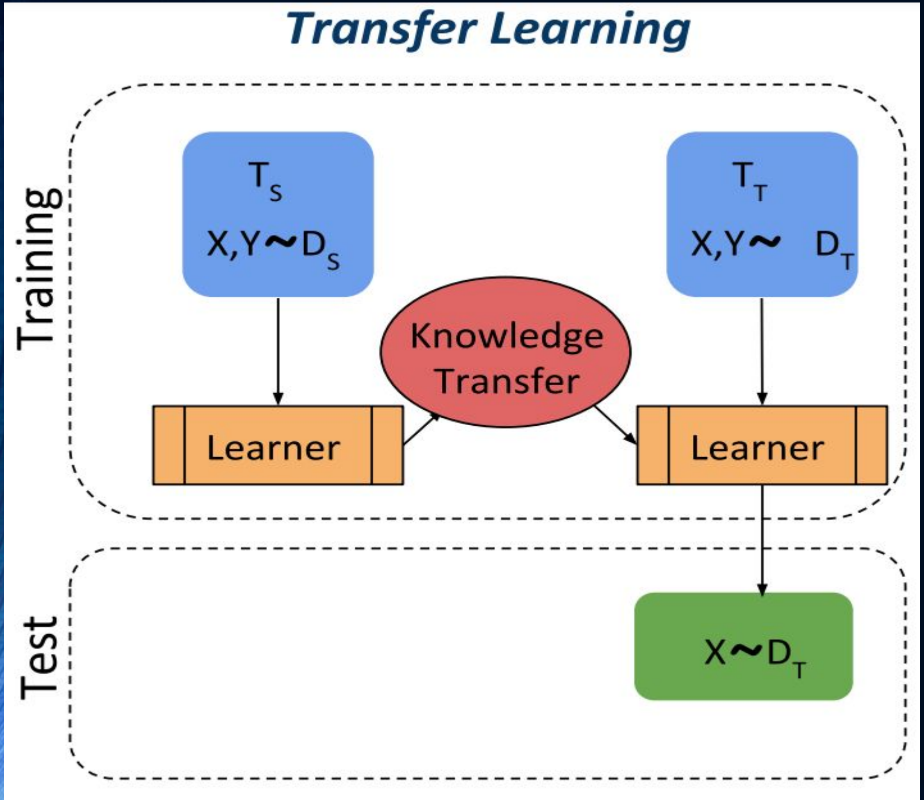
Learning of multiple related tasks offline, simultaneously

Using a set or subset of shared parameters

No continual model adaptation

Image credit: de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Continual vs Transfer Learning



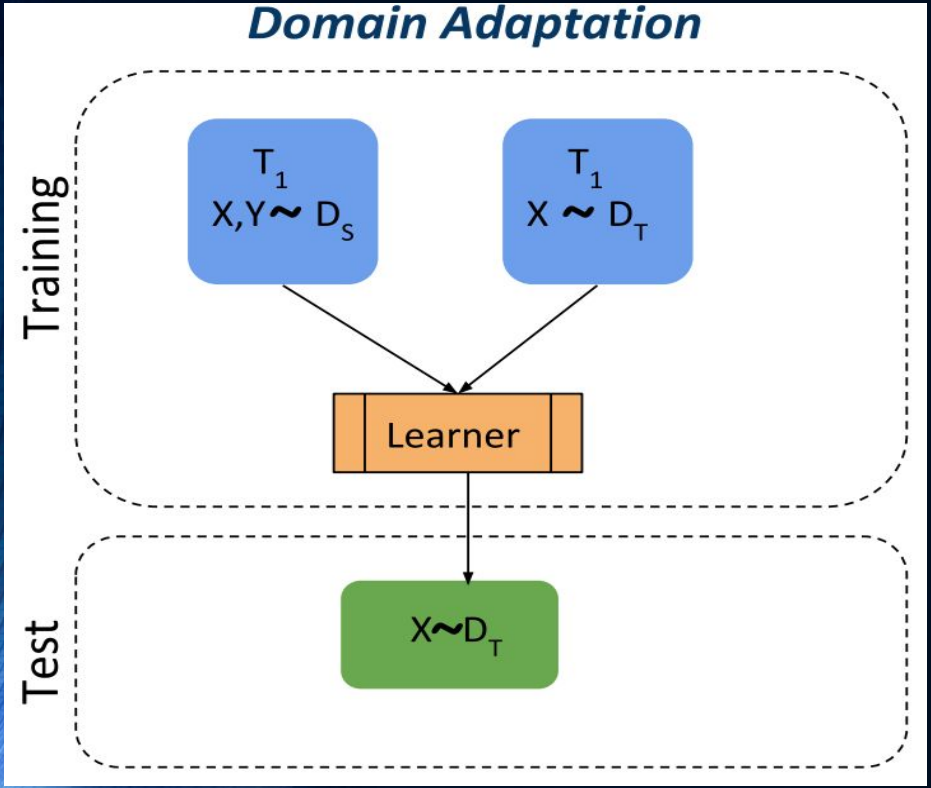
Help learning the target task using model trained on the source task

No continuous adaptation after learning the target task

Performance on the source task(s) is not taken into account

Image credit: de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Continual Learning vs Domain Adaptation



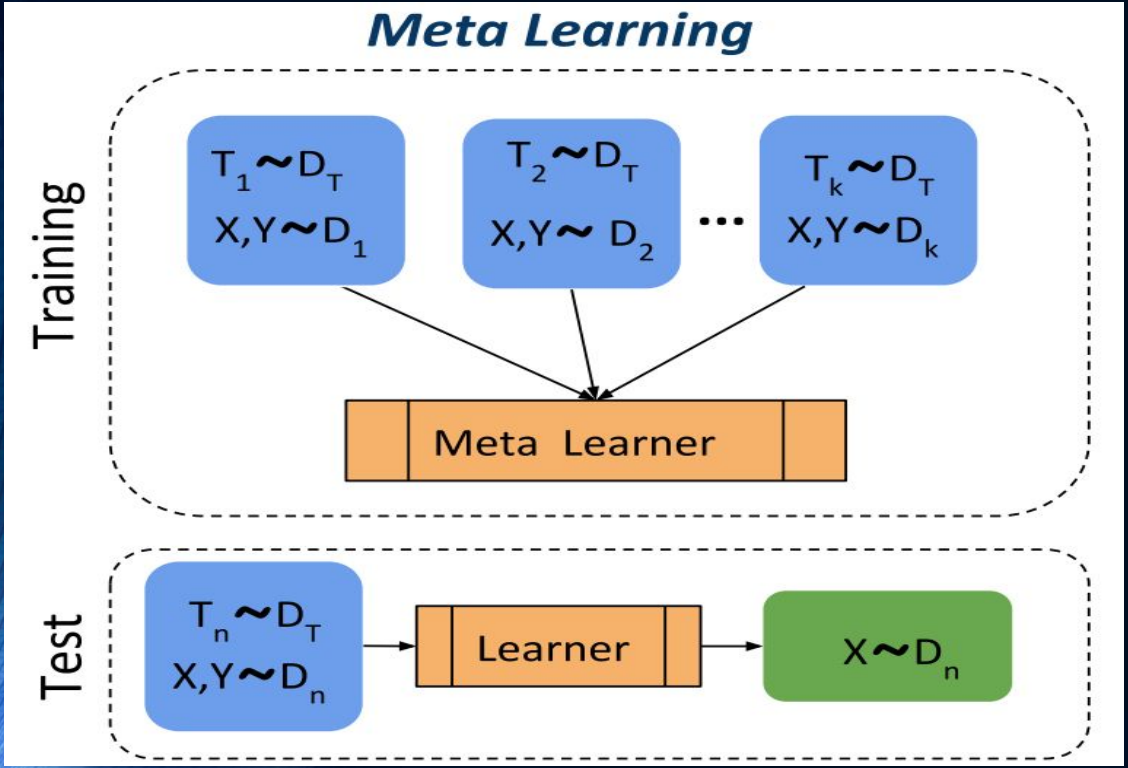
Transfer learning with same source and target tasks, but from different input domains

Trains on the source domain, adapts model to the (with no or only a few labels).

Unidirectional; does not involve any accumulation of knowledge

Image credit: de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Continual vs Meta-Learning



Faster adaptation on a task given a large number of training tasks

Offline training: a set of training tasks available at the same time

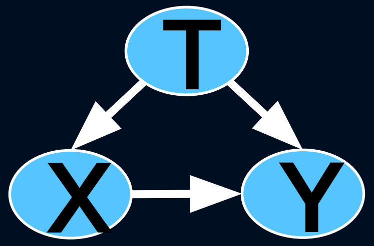
Image credit: de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Continual Learning Settings

X – input vector

Y – class label

T – task (context) defines P(X,Y|T)



Task ID observed at training:

- T observed at test: task-incremental CL
- T not observed at test: class-incremental or domain-incremental CL

Task ID/boundary is not known at training:

- Task-agnostic 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 the tasks
- Assumptions: $\{\mathcal{Y}^{(t)}\} \neq \{\mathcal{Y}^{(t+1)}\}$

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

Task-Incremental CL

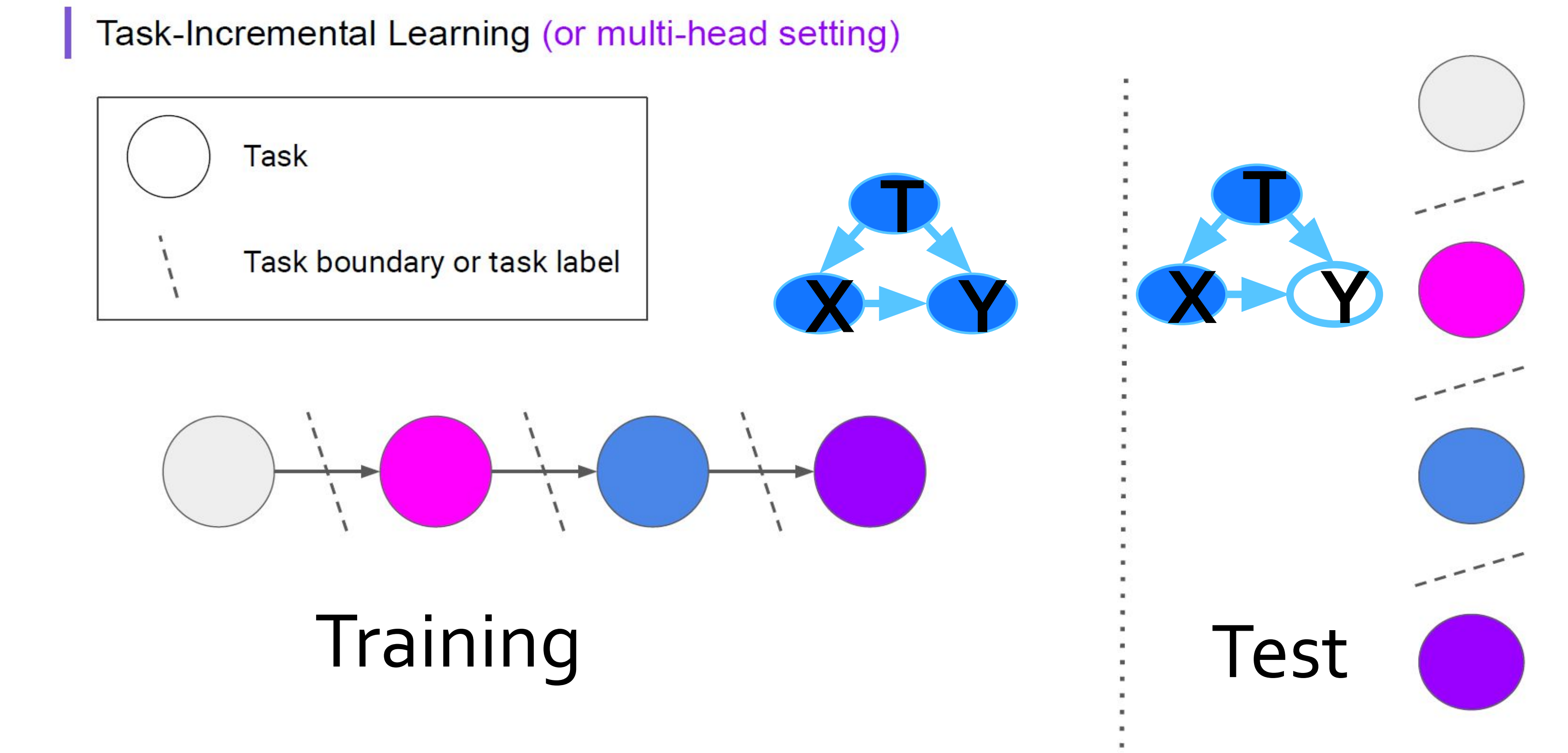


Figure credit: Massimo Caccia

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. An example: sequentially learning MNIST digits (split-MNIST)
- Assumptions:

$$P(\mathcal{X}^{(t)}) \neq P(\mathcal{X}^{(t+1)})$$

$$\{\mathcal{Y}^{(t)}\} = \{\mathcal{Y}^{(t+1)}\}$$

$$P(\mathcal{Y}^{(t)}) \neq P(\mathcal{Y}^{(t+1)})$$

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

Class-Incremental CL

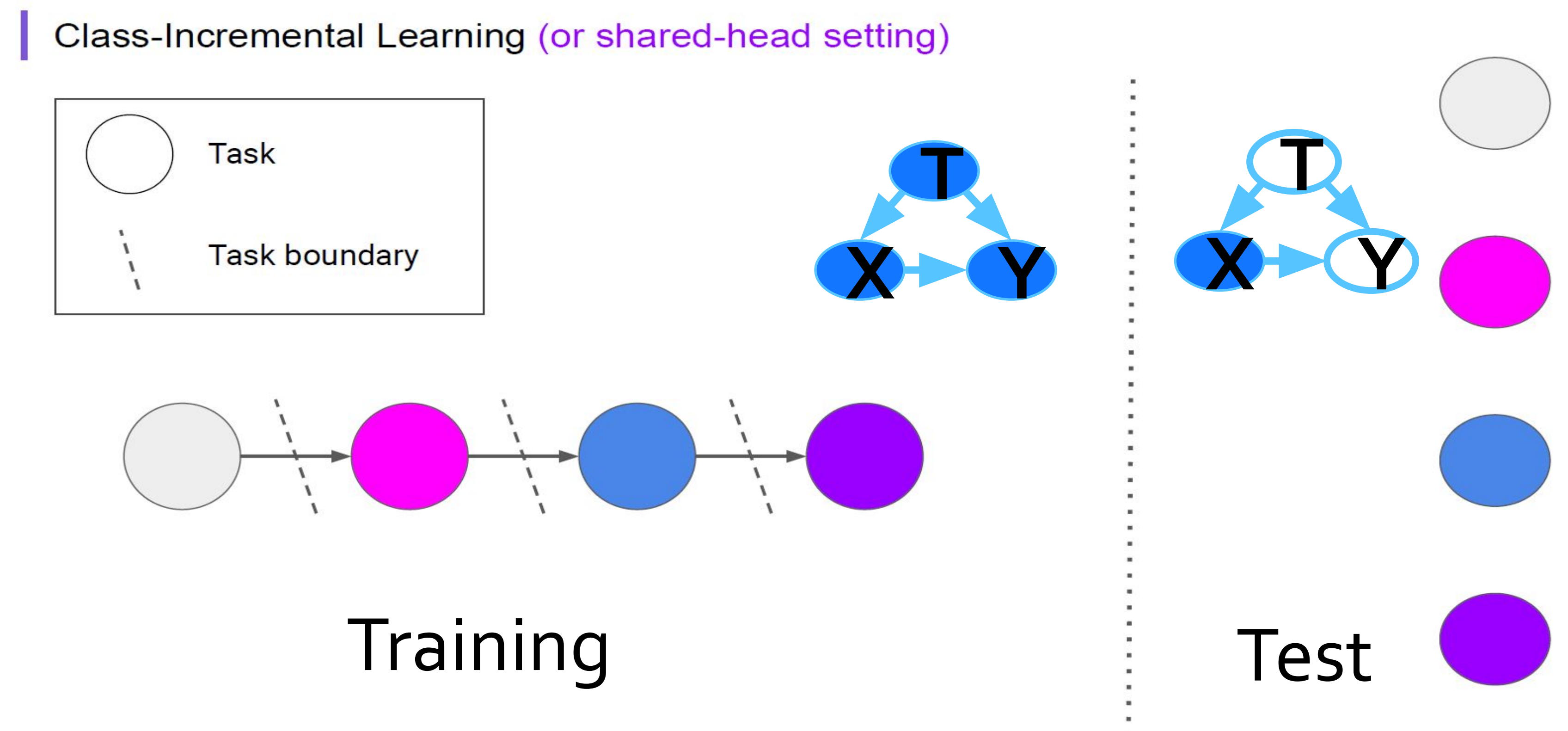


Figure credit: Massimo Caccia

Domain-Incremental CL

- Task identity is not available at test time
- Models however only need to solve the task at hand; they are not required to infer which task it is
- Typical examples of this scenario: the structure of the tasks is always the same, but the input-distribution is changing (e.g., 'permuted MNIST')
- Assumptions: similar to class-incremental, except for last one:

$$P(\mathcal{X}^{(t)}) \neq P(\mathcal{X}^{(t+1)})$$

$$\{\mathcal{Y}^{(t)}\} = \{\mathcal{Y}^{(t+1)}\}$$

$$\boxed{P(\mathcal{Y}^{(t)}) = P(\mathcal{Y}^{(t+1)})}$$

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

Example: Split MNIST

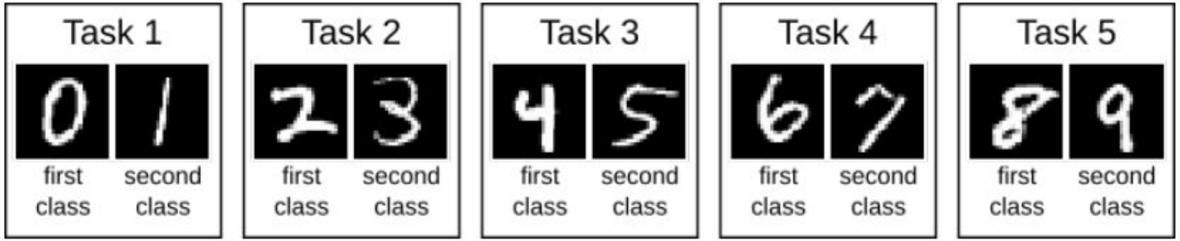


Figure 1: Schematic of the split MNIST task protocol.

Table 1: The split MNIST task protocol according to each continual learning scenario.

Incremental task learning	With task given, is it the first or second class? (e.g., ‘0’ or ‘1’)
Incremental domain learning	With task unknown, is it a first or second class? (e.g., in [‘0’,‘2’,‘4’,‘6’,‘8’] or in [‘1’,‘3’,‘5’,‘7’,‘9’])
Incremental class learning	With task unknown, which digit is it? (choice from ‘0’ to ‘9’)

Gido van de Ven and Andreas S. Tolias. Three scenarios for continual learning. *arXiv:1904.07734*, 2019.

Example: Permuted MNIST

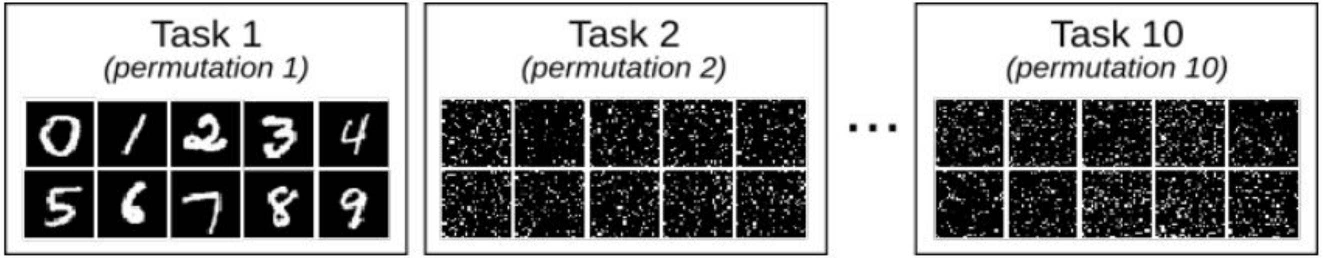


Figure 2: Schematic of the permuted MNIST task protocol.

Table 2: The permuted MNIST task protocol according to each continual learning scenario.

Incremental task learning	Given permutation X was applied, which digit is it?
Incremental domain learning	With permutation unknown, which digit is it?
Incremental class learning	Which digit is it *and* which permutation was applied?

Gido van de Ven and Andreas S. Tolias. Three scenarios for continual learning. *arXiv:1904.07734*, 2019

Task-Agnostic CL: Most Challenging

Task identity is not available even at training time!

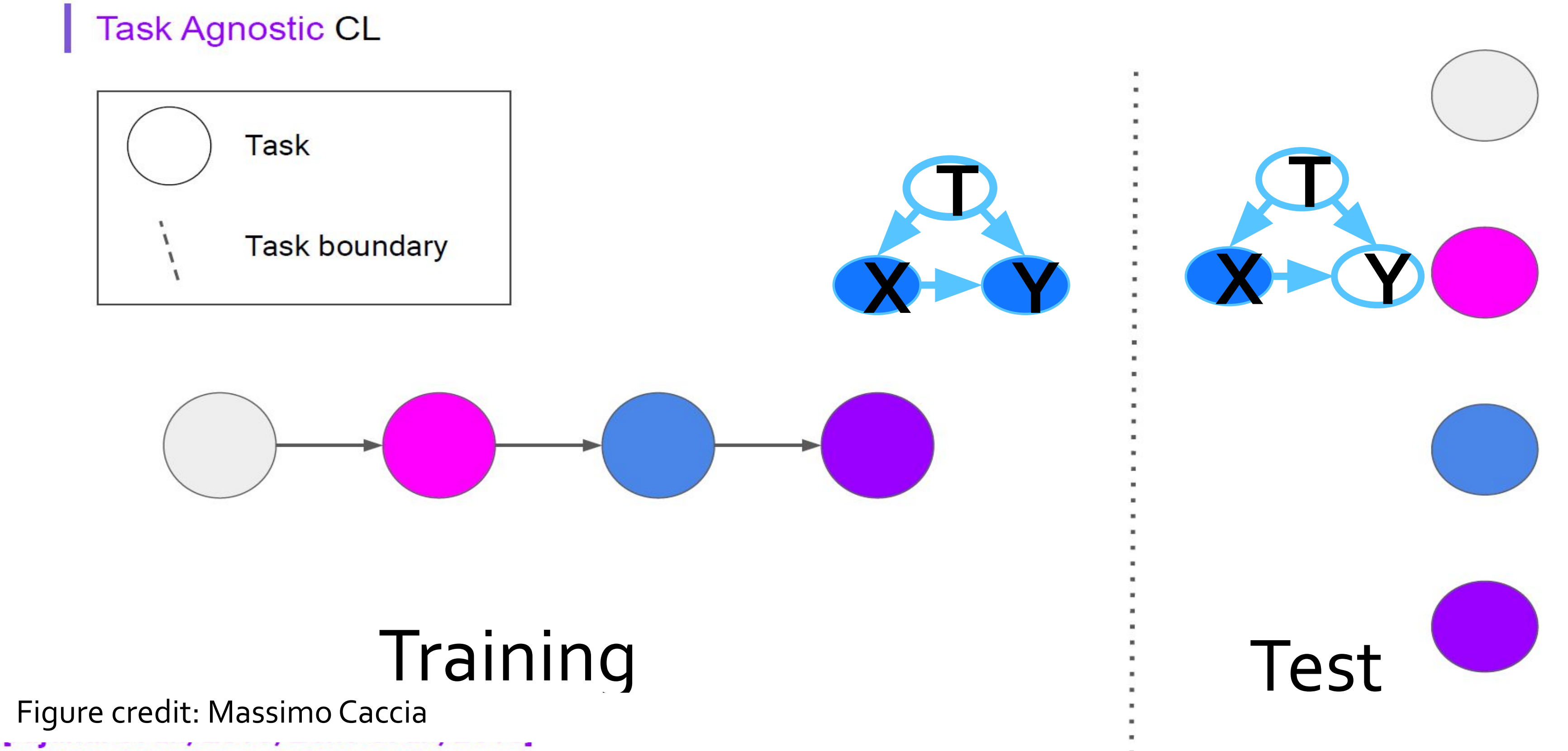


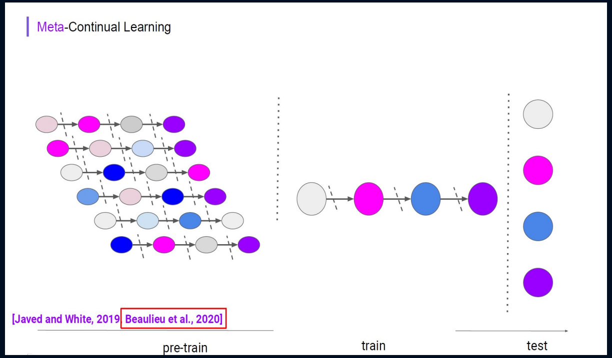
Figure credit: Massimo Caccia

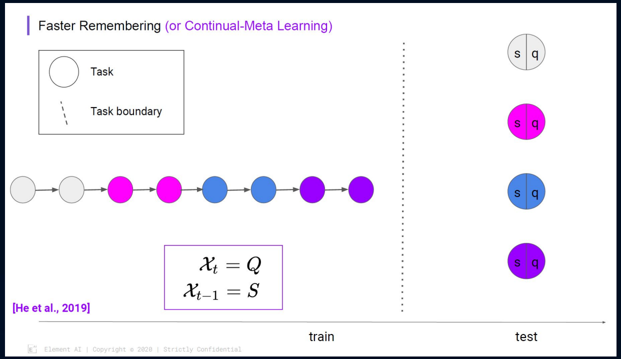
Towards A Unified Framework for Continual Learning?

Caccia, M et al (2020) Online fast adaptation and knowledge accumulation (**OSAKA**)

- Task-agnostic setting
- Combines continual-meta [1] and meta-continual [2] learning
  - Continual-MAML:

    an online extension of MAML



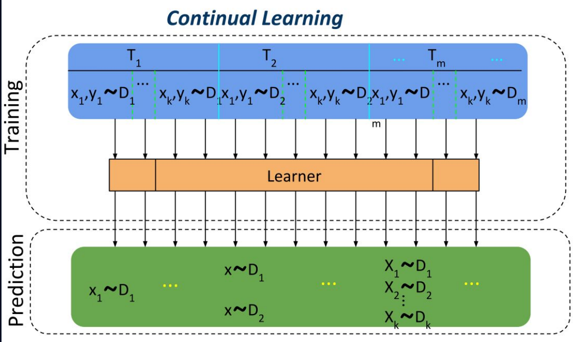


[1] He et al. Task agnostic continual learning via meta learning. 2019
[2] K. Javed and M. White. Meta-learning representations for continual learning. 2019.

Objective

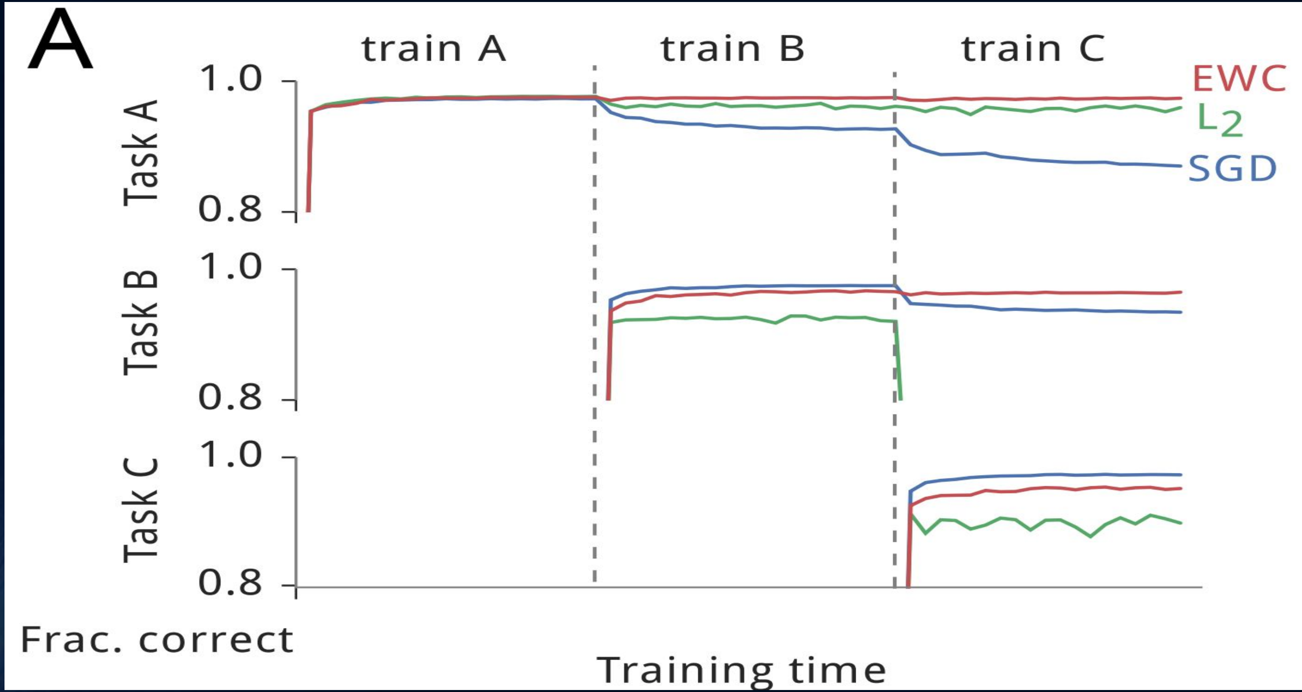
Data $(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})$ is randomly drawn from distribution $D^{(t)}$, with $\mathcal{X}^{(t)}$ a set of data samples for task t, and $\mathcal{Y}^{(t)}$ the corresponding ground truth labels. The goal is to control the statistical risk of all seen tasks given limited or no access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

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



However, we have no access to the previous tasks, and thus cannot compute this empirical risk exactly.

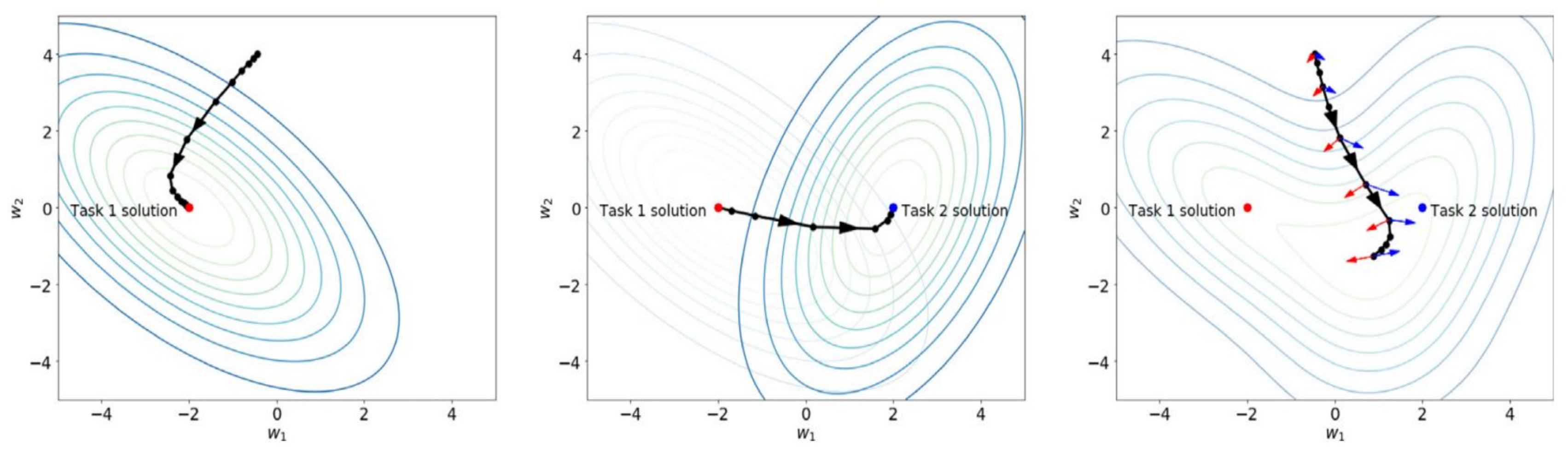
The Curse of Catastrophic Forgetting



Standard learning methods such as SGD quickly “forget” old knowledge (parameters) when the data/task change (i.e., adapt too well).

Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks, PNAS 2017.

Multi-Task Gradient Dynamics: Tug-of-War



Loss(Task1) Loss(Task2) Loss(Task1) +Loss(Task2)

However, the tasks are not available simultaneously in CL!

Need to use some form of memory, or to modify the gradients, to still take into account what solutions are good for previous tasks

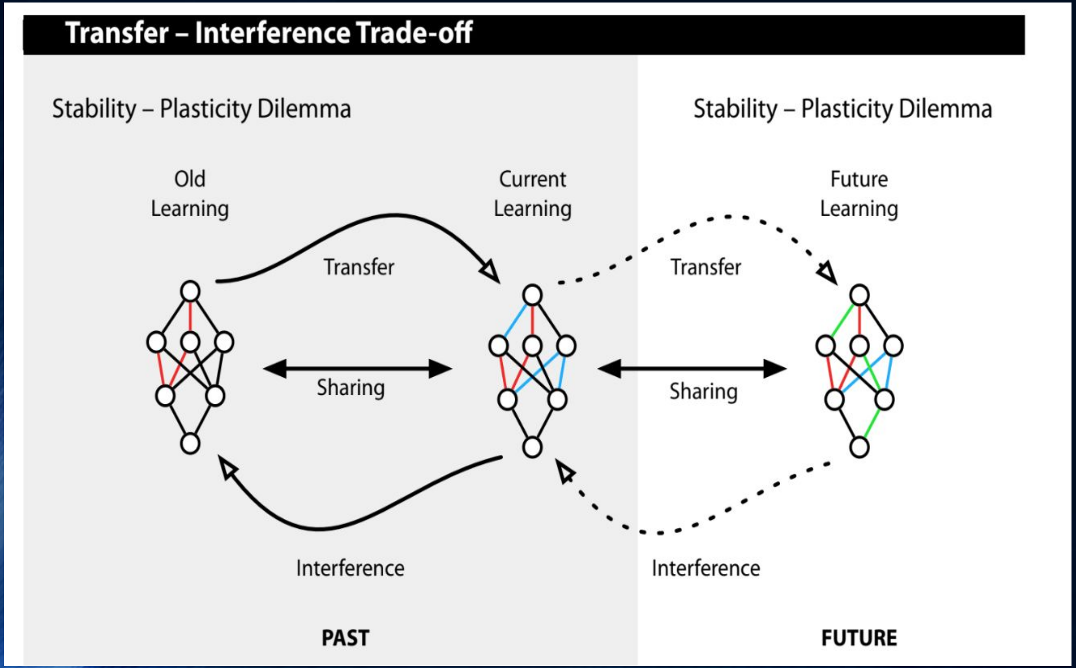
Image credit: Hadsell et al. Embracing Change: Continual Learning in Deep Neural Networks, 2020.

Transfer vs Interference

Transfer from task A to task B ⇔ improved performance on B after learning A

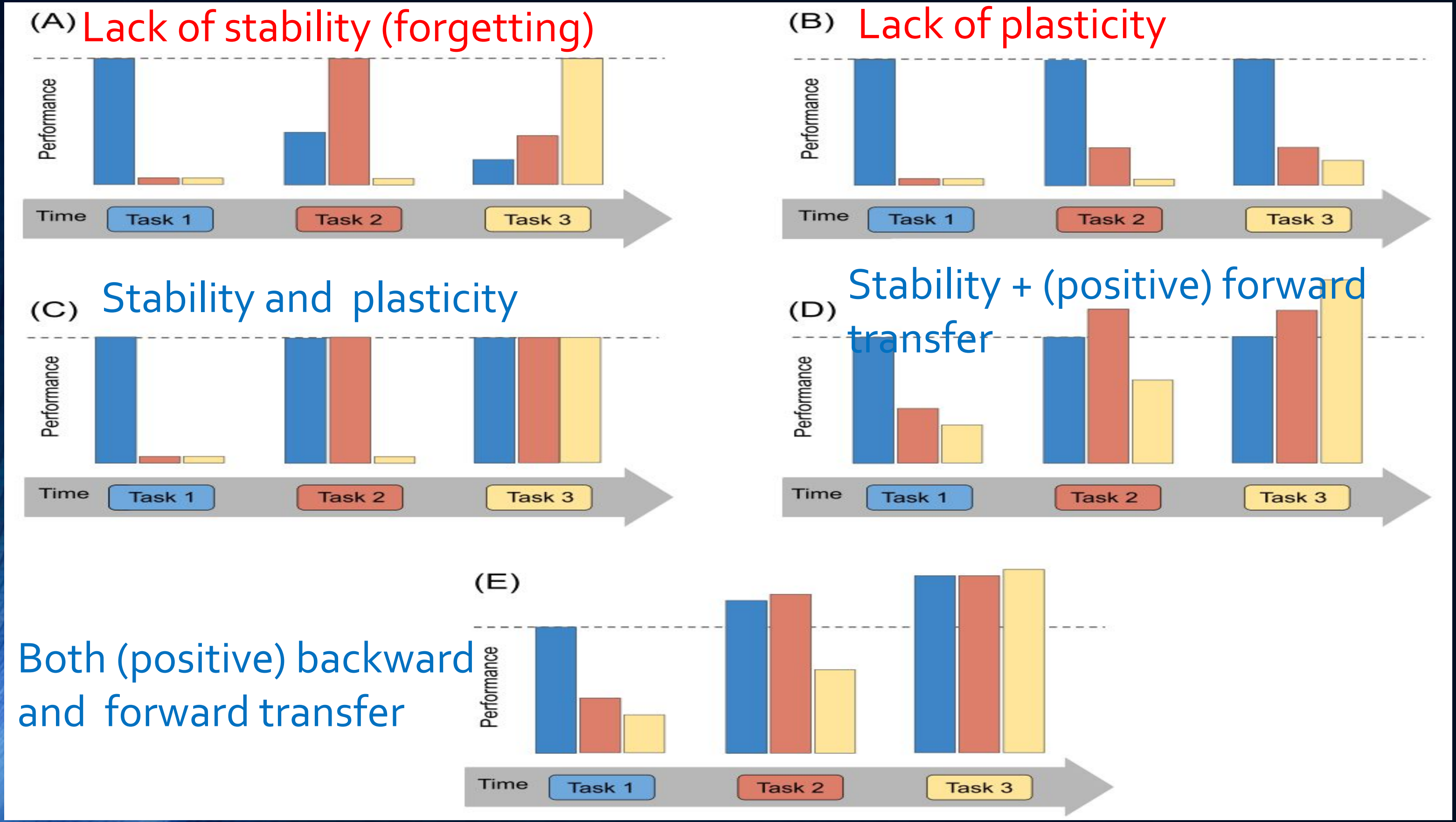
Interference = negative transfer (i.e., decrease in performance)

Weight-sharing can cause both ⇔ finding a good trade-off is the key!



The reason for catastrophic forgetting is "the very thing — a single set of shared weights — that gave the networks their remarkable abilities to generalize and degrade gracefully." (French, 1999)

Possible Scenarios in CL



Bad

Good

Image credit: Hadsell et al. Embracing Change: Continual Learning in Deep Neural Networks, 2020.

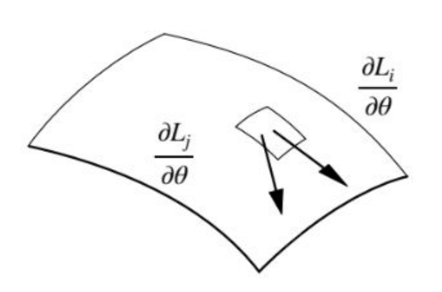
Transfer vs Interference ("Negative Transfer")

Riemer et al (2019) Learning to Learn without Forgetting By Maximizing Transfer and Minimizing Interference

Transfer:

$$\frac{\partial L(x_i, y_i)}{\partial \theta} \cdot \frac{\partial L(x_j, y_j)}{\partial \theta} > 0.$$

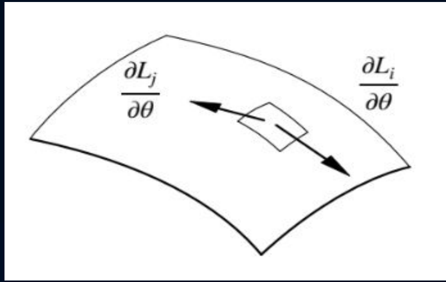
- When we train on one we improve on the other (a form of generalization)
- Analogous to positive transfer in either the forward or backward direction

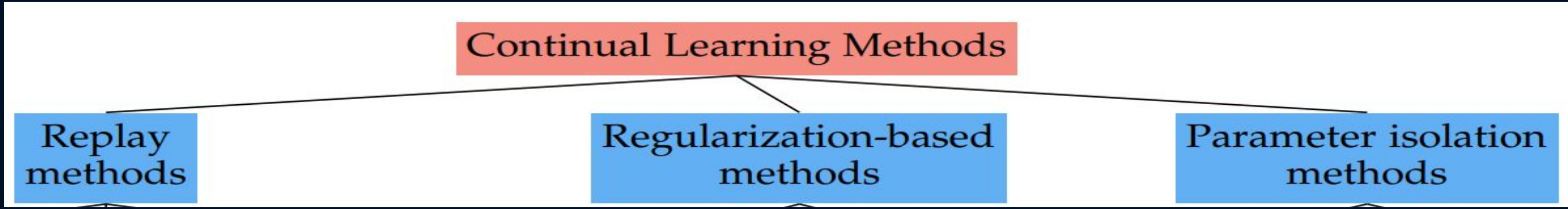


Interference:

$$\frac{\partial L(x_i, y_i)}{\partial \theta} \cdot \frac{\partial L(x_j, y_j)}{\partial \theta} < 0.$$

- When we train on one we get worse at the other
- Analogous to negative transfer in either the forward or backward direction





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.

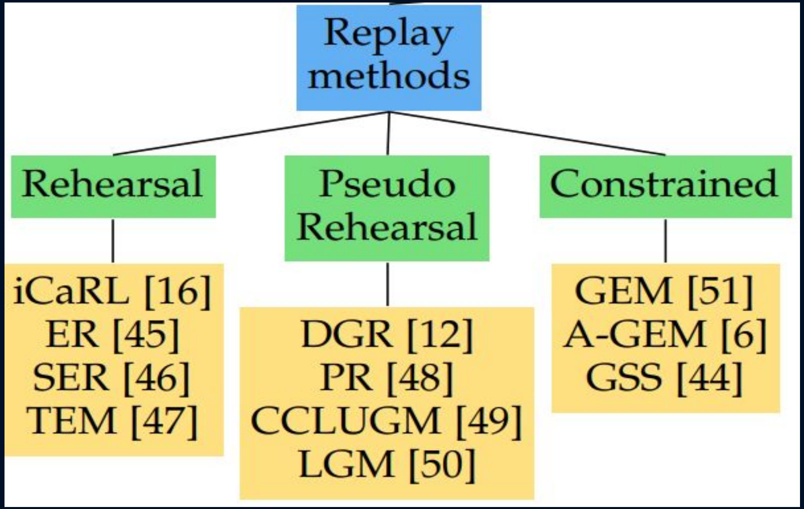
Better for privacy, minimizing memory size and access latency.

Architecture-based

Each task will use different model parameters (no forgetting!). Can use a static architecture or keep expanding it.

How will the overall performance scale with the model size?

de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.



Rehearsal methods:

retrain current model on a subset of stored samples jointly with new tasks (e.g., reservoir sampling [45])

Pseudo rehearsal methods:

Feed random input to previous models, use the output as a pseudo-sample [48]. 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., GEM, in task-incremental setting, projects the estimated gradient direction on the feasible region determined by previous task gradients, etc. More recent work (A-GEM, MER, etc).

de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Example: Replay + Constraints (GEM)

Lopez-Paz and Ranzato (2017). Gradient episodic memory for Continual Learning.

Idea:

(1) store small amount of data per task in **memory**

(2) when making updates for new tasks, ensure that they don't **unlearn** previous tasks

How do we accomplish (2)?

learning predictor $y_t = f_\theta(x_t, z_t)$ memory: $\mathcal{M}_k$ for task z_k

For $t = 0, \ldots, T$

minimize $\mathcal{L}(\ f_\theta(\cdot\ , z_t)\ ,\ (x_t, y_t)\ )$

subject to $\mathcal{L}(\ f_\theta\ ,\ \mathcal{M}_k\ ) \leq \mathcal{L}(\ f_\theta^{t-1}\ ,\ \mathcal{M}_k\ )$ for all $z_k < z_t$

(i.e. s.t. loss on previous tasks doesn't get worse)

Assume local linearity:

$$\langle g_t, g_k \rangle := \left\langle \frac{\partial \mathcal{L}(\ f_\theta\ ,\ (x_t, y_t)\ )}{\partial \theta}, \frac{\mathcal{L}(\ f_\theta\ ,\ \mathcal{M}_k\ )}{\partial \theta} \right\rangle \geq 0 \quad \text{for all } z_k < z_t$$

Can formulate & solve as a QP.

Lopez-Paz & Ranzato. Gradient Episodic Memory for Continual Learning. NeurIPS '17

Lopez-Paz and Ranzato (2017). Gradient episodic memory for Continual Learning.

Experiments

Problems:

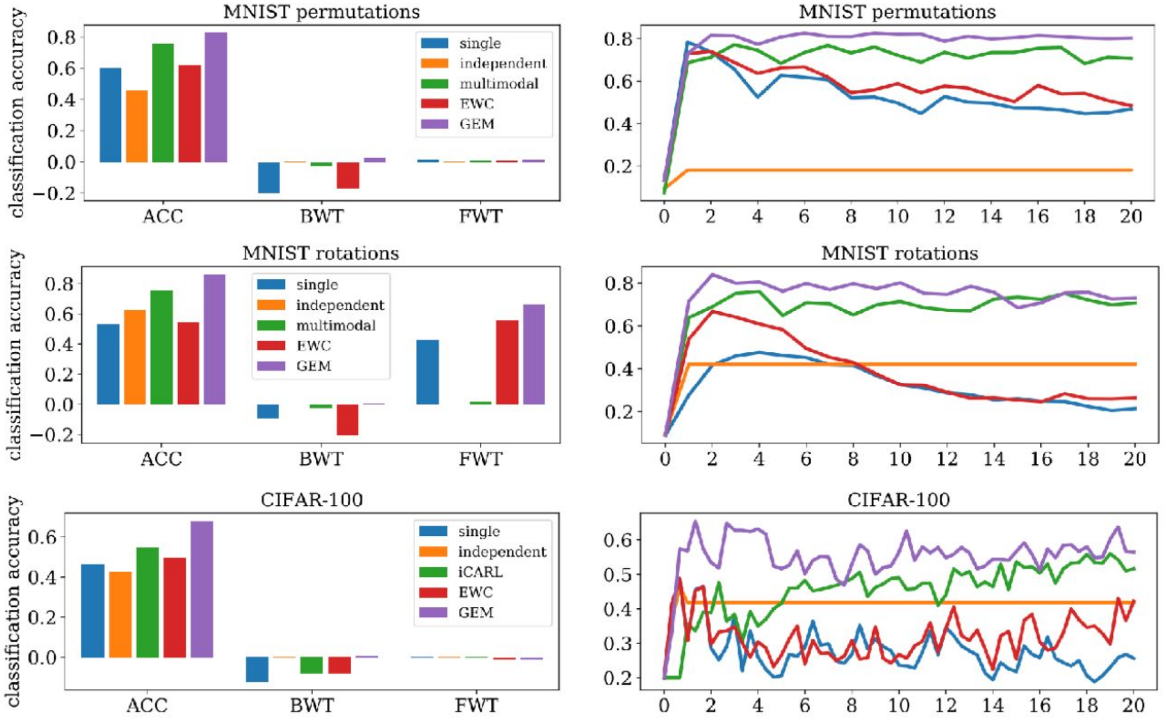
- MNIST permutations
- MNIST rotations
- CIFAR-100 (5 new classes/task)

BWT: backward transfer,

FWT: forward transfer

Total memory size:

5012 examples



Meta-Experience Replay (MER) Approach

Riemer et al (2019) Learning to Learn without Forgetting By Maximizing Transfer and Minimizing Interference

Standard offline training objective with dataset D:

$$\theta = arg\min_{\theta} \mathbb{E}_{(x,y)\in D}[L(x,y)],$$

Modifying it to also learn to maximize transfer and minimize interference in either direction:

$$\theta = arg\min_{\theta} \mathbb{E}_{(x_i,y_i)\&(x_j,y_j)\in D}[L(x_i,y_i) + L(x_j,y_j) \boxed{- \alpha \frac{\partial L(x_i,y_i)}{\partial \theta} \cdot \frac{\partial L(x_j,y_j)}{\partial \theta}}],$$

Meta-learning perspective: we would like to learn to learn each example in a way that generalizes to other examples from the overall distribution.

Replay + Meta-Learning: Meta-Experience Replay

Reptile [1] is an efficient meta-learning algorithm that approximates the same objective as MAML.

Reptile can be extended to continual learning by integrating with ER! ☺

✓ Results from [1] still hold to the extent that our buffer captures the full variation of the distribution of examples seen.

✓ We can separate an ER batch into SGD steps over individual examples and apply a Reptile parameter meta-update.

✓ We also note that it is important to prioritize the current example before moving on as it may not be added to M.

Approximate Objective (s batches with k examples each):

$$\theta = arg\min_{\theta} \mathbb{E}_{(x_{11},y_{11}),\ldots,(x_{sk},y_{sk})\in M}[2\sum_{i=1}^{s}\sum_{j=1}^{k}[L(x_{ij},y_{ij})\boxed{-\sum_{q=1}^{i-1}\sum_{r=1}^{j-1}\alpha\frac{\partial L(x_{ij},y_{ij})}{\partial\theta}\cdot\frac{\partial L(x_{qr},y_{qr})}{\partial\theta}}]].$$

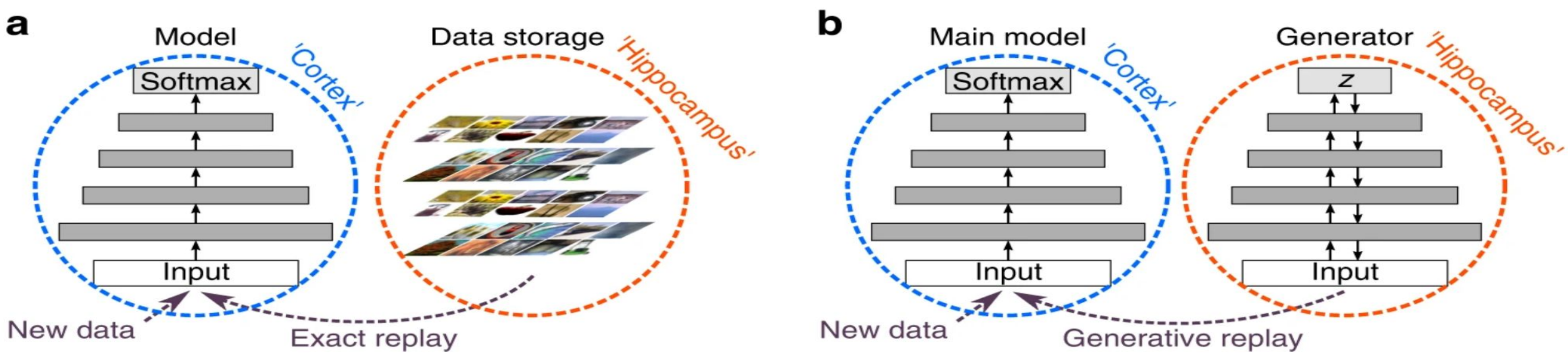
Retained Accuracy After Training on All Tasks

Model	Buffer Size	MNIST Rotations	MNIST Permutations	Many Permutations	Incremental Omniglot
Online	N/A	46.40 ± 0.78	55.42 ± 0.65	32.62 ± 0.43	4.36 ± 0.37
EwC	N/A	57.96 ± 1.33	62.32 ± 1.34	33.46 ± 0.46	4.63 ± 0.14
GEM	5120	87.12 ± 0.44	82.50 ± 0.42	56.76 ± 0.29	18.03 ± 0.15
	500	72.08 ± 1.29	69.26 ± 0.66	32.14 ± 0.50	-
	200	66.88 ± 0.72	55.42 ± 1.10	-	-
MER	5120	**89.56** ± 0.11	**85.50** ± 0.16	**61.84** ± 0.25	**75.23** ± 0.52
	500	**81.82** ± 0.52	**77.40** ± 0.38	**47.40** ± 0.35	**32.05** ± 0.69
	200	**77.24** ± 0.47	**72.74** ± 0.46	-	-

Episodic (exact) vs Generative Replay

van de Ven et al (2020). Brain-inspired replay for continual learning with artificial neural networks

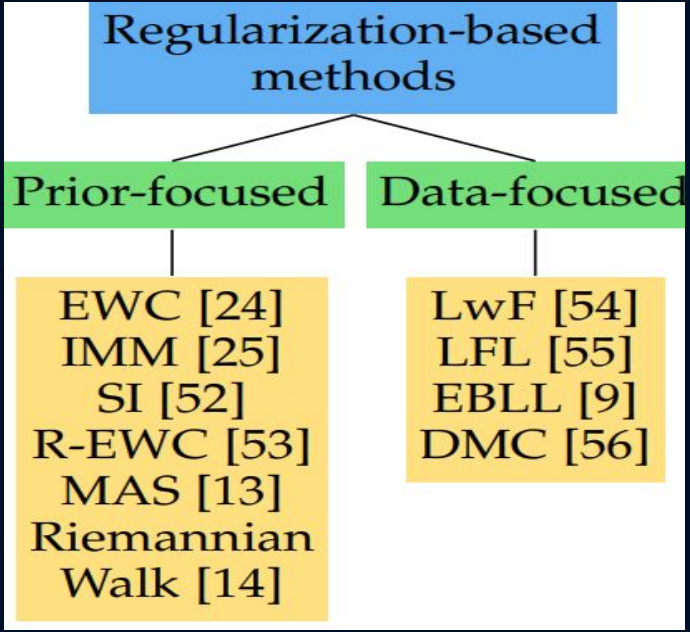
From: Brain-inspired replay for continual learning with artificial neural networks



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].

Novel GR method: internal or hidden representations are replayed that are generated by the network's own, context-modulated feedback connections.

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



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

Data-focused:

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

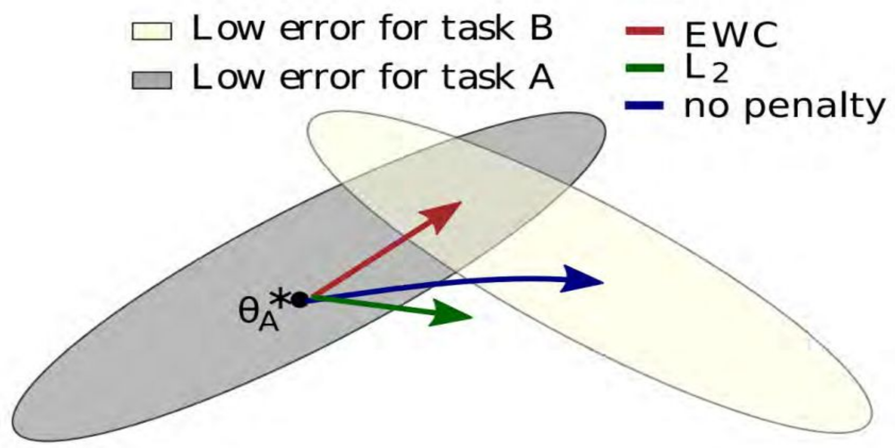
Prior-focused:

Use an estimated distribution over the model parameters as prior when learning from new data; penalize changes to parameters important for the past tasks (e.g. EWC and later work).

de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Elastic Weight Consolidation (EWC)

[Kirkpatrick et al, PNAS 2017]



Idea: Don't let *important* parameters change drastically (reduce plasticity)

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

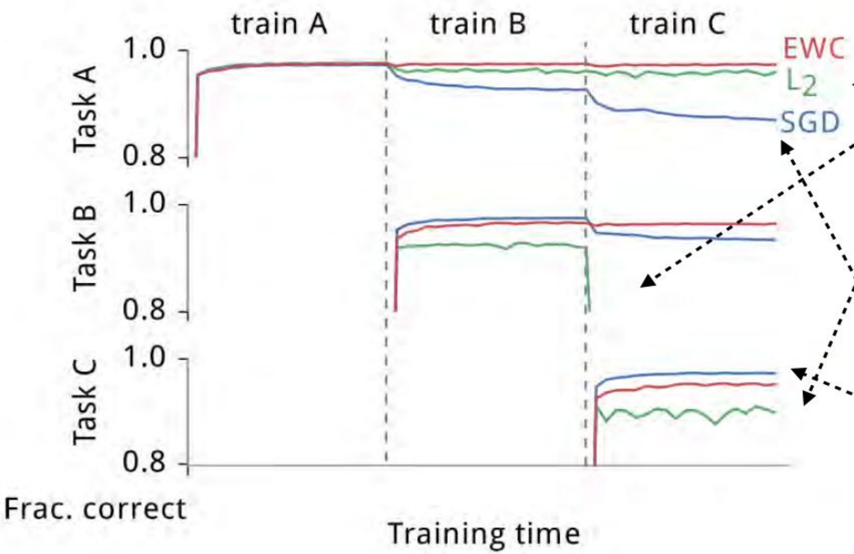
Task B Loss

- Inspired by research on synaptic consolidation

Slide credit: ICML 2019 Tutorial on Never-Ending Learning by Tom Mitchell and Partha Talukdar

Elastic Weight Consolidation (EWC)

[Kirkpatrick et al., PNAS 2017]

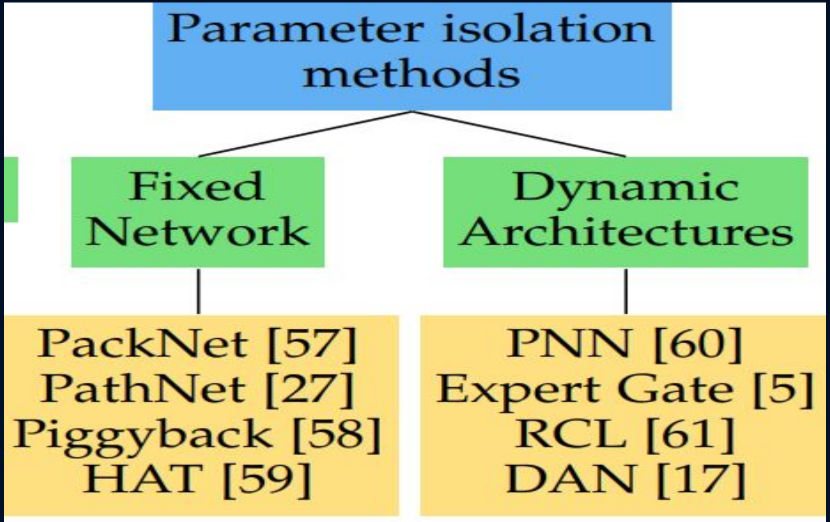


L2 is too rigid, doesn't allow learning on new tasks => parameter weighting is important

Catastrophic Forgetting in SGD

MNIST experiments. New tasks are random pixel permutations.

Slide credit: ICML 2019 Tutorial on Never-Ending Learning by Tom Mitchell and Partha Talukdar



Idea: avoid forgetting by using different parameters for each task

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

Fixed Network Methods:

Network parts used for previous tasks are masked out when learning new tasks (e.g., at neuronal level (HAT) or at parameter level (PackNet, PathNet)

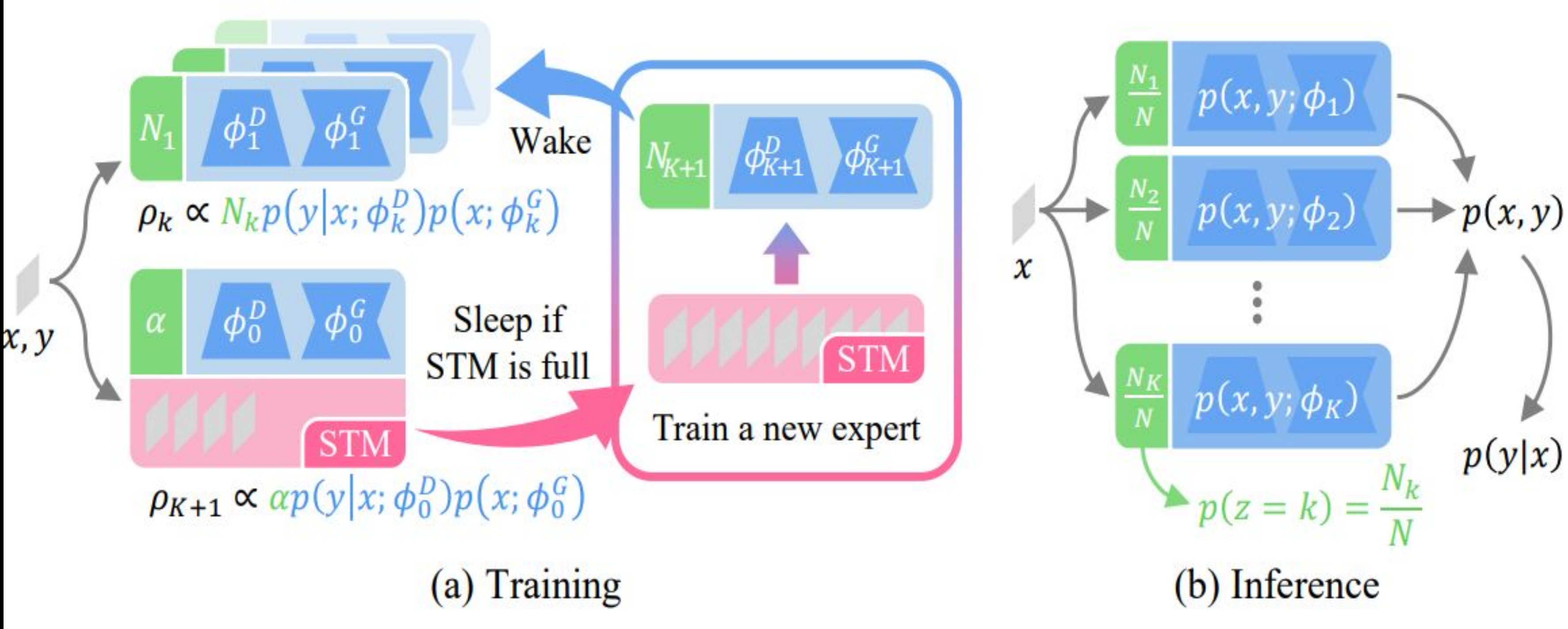
Dynamic Architecture Methods:

When model size is not constrained: grow new branches for new tasks, while freezing previous task parameters (RCL), or dedicate a model copy to each task (Expert Gate), etc.

de Lange et al. Continual learning: A comparative study on how to defy forgetting in classification tasks, 2019.

Example: Architectural Approaches

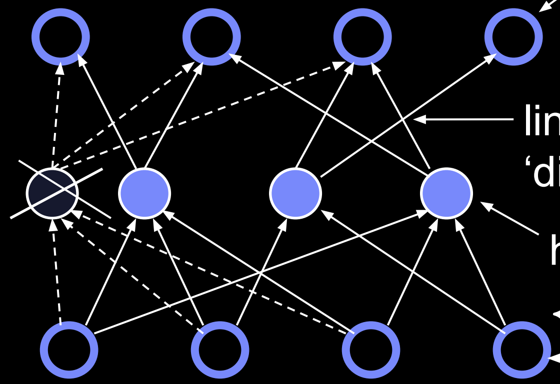
Lee at al (2020) A Neural Dirichlet Process Mixture Model for Task-Free Continual Learning. ICLR2020



Neurogenetic Autoencoder (Online Dictionary Learner)

Garg et al (2017) Neurogenesis-inspired dictionary learning.

- A sparse autoencoder model (a.k.a. dictionary learning)
- Neuronal "birth": adding random hidden units
-
- Neuronal "death": using l1/l2 (group sparsity) regularize

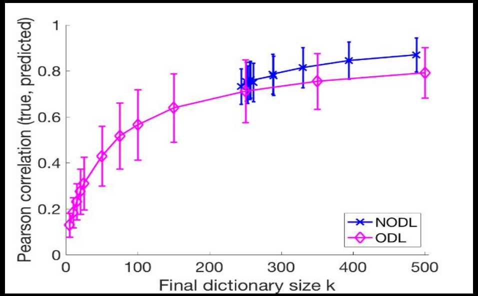


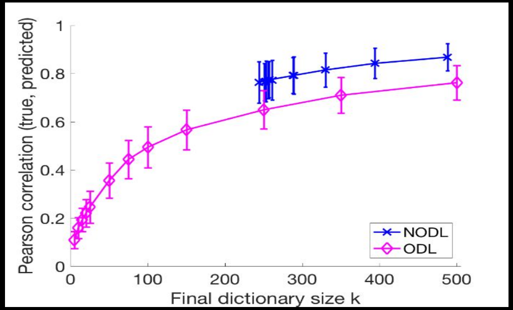
$$\hat{f}_t(D) = \underbrace{\frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}||\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i||_2^2}_{\text{reconstruction error}} + \underbrace{\lambda_c||\boldsymbol{\alpha}_i||_1}_{\text{sparsity on codings}} + \underbrace{\lambda_g\sum_j ||\boldsymbol{d}_j||_2}_{L_1/L_2\text{ group sparsity}} + \underbrace{\sum_j \lambda_j||\boldsymbol{d}_j||_1}_{\text{sparse elements}}$$

“Neurogenesis” Helps CL

Garg et al (2017) Neurogenesis-inspired dictionary learning.

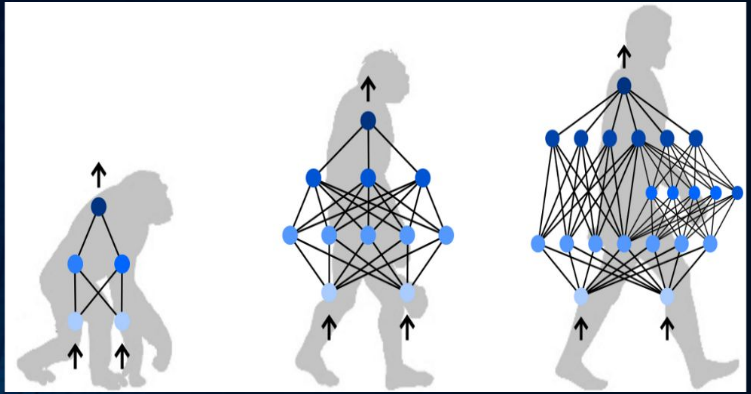
From urban (“Oxford”) to nature (flowers, dogs,cats) images

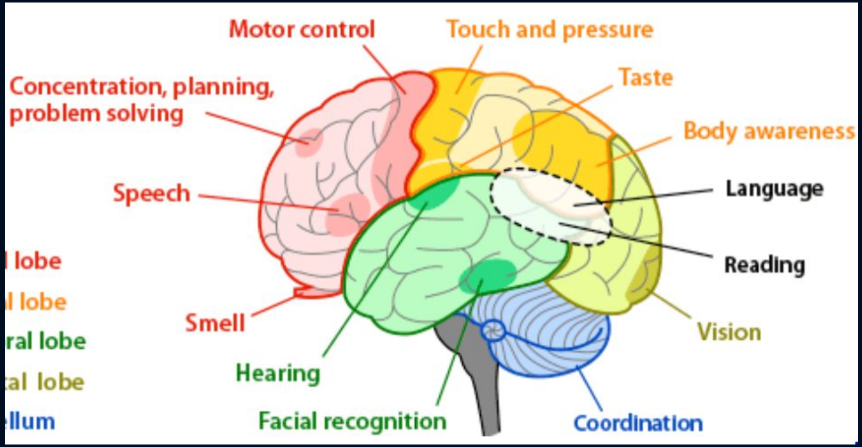




Neurogenetic Online Dictionary Learner (NODL) improves reconstruction accuracy over standard ODL on BOTH old and new data (i.e. avoids forgetting while adapting), and learns more compact representations.

Role of CL: Evolving a “Library” of “Basis Functions” ?





Network components ⇔ finite functional “basis”

$$\{ h_1(x), \ldots, h_k(x) \}$$

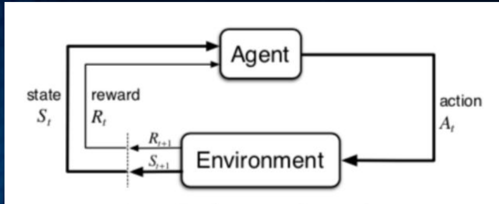
$$\overbrace{f_1(x) \qquad f_2(x) \qquad f_3(x) \qquad \ldots \qquad f_n(x) \qquad \ldots}$$

Infinite stream of changing environments and tasks

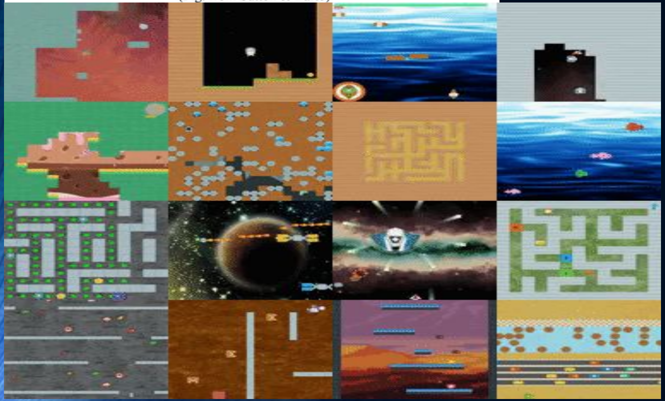
Continual Reinforcment Learning

Continual Reinforcement Learning

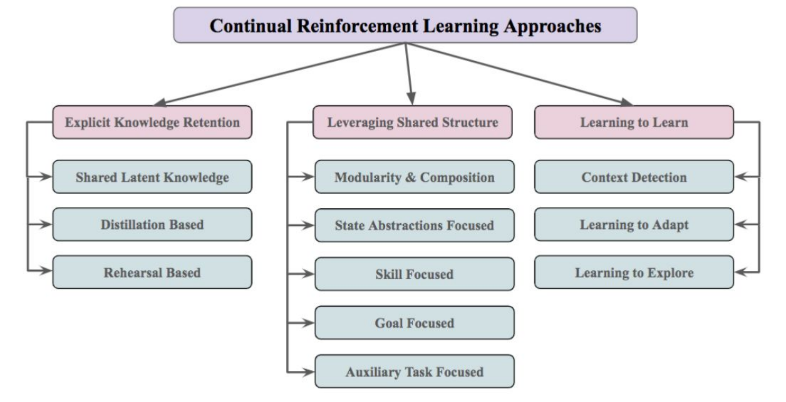
Khimya Khetarpal*, Matthew Riemer*, Irina Rish, Doina Precup (2020).
Towards Continual Reinforcement Learning: A Review and Perspectives.

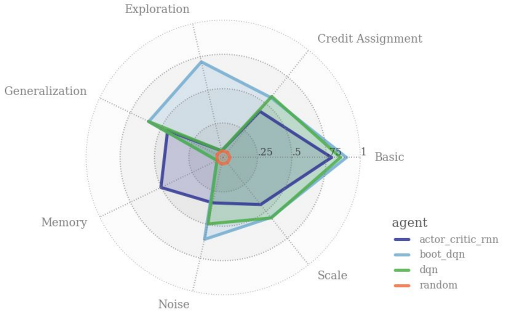


Agent-Environment Interaction
(Fig. from Sutton & Barto)



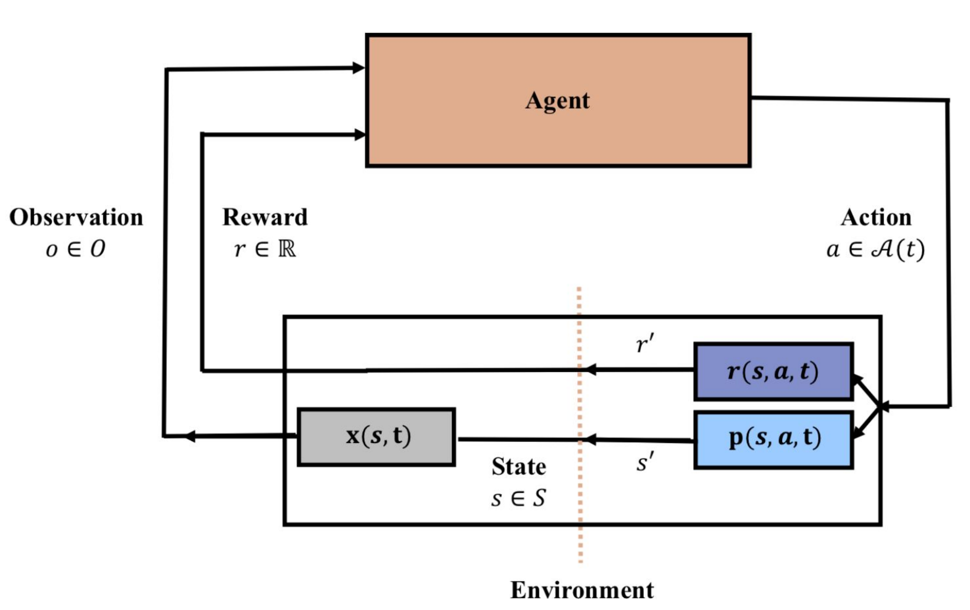
Procgen: A benchmark for procedurally generated set of environments to measure generalization.

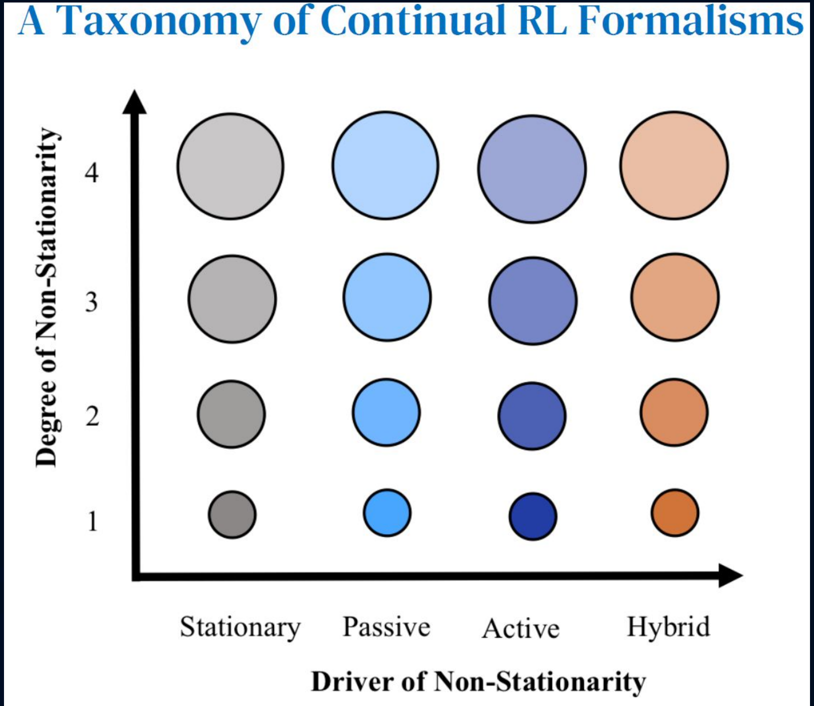




Bsuite: is a collection of carefully-designed experiments that investigate core capabilities of a reinforcement learning (RL) agent.

Continual Reinforcement Learning





Drivers of Nonstationarity

Passive Non-stationarity: In passive non-stationary environments, we assume that the non-stationary behavior (i.e. the evolution of tasks) does not depend on the behavior of the agent itself when interacting with the environment.

- ❒ E.g. "*Hidden-Mode Markov Decision Processes for Nonstationary Sequential Decision Making*"

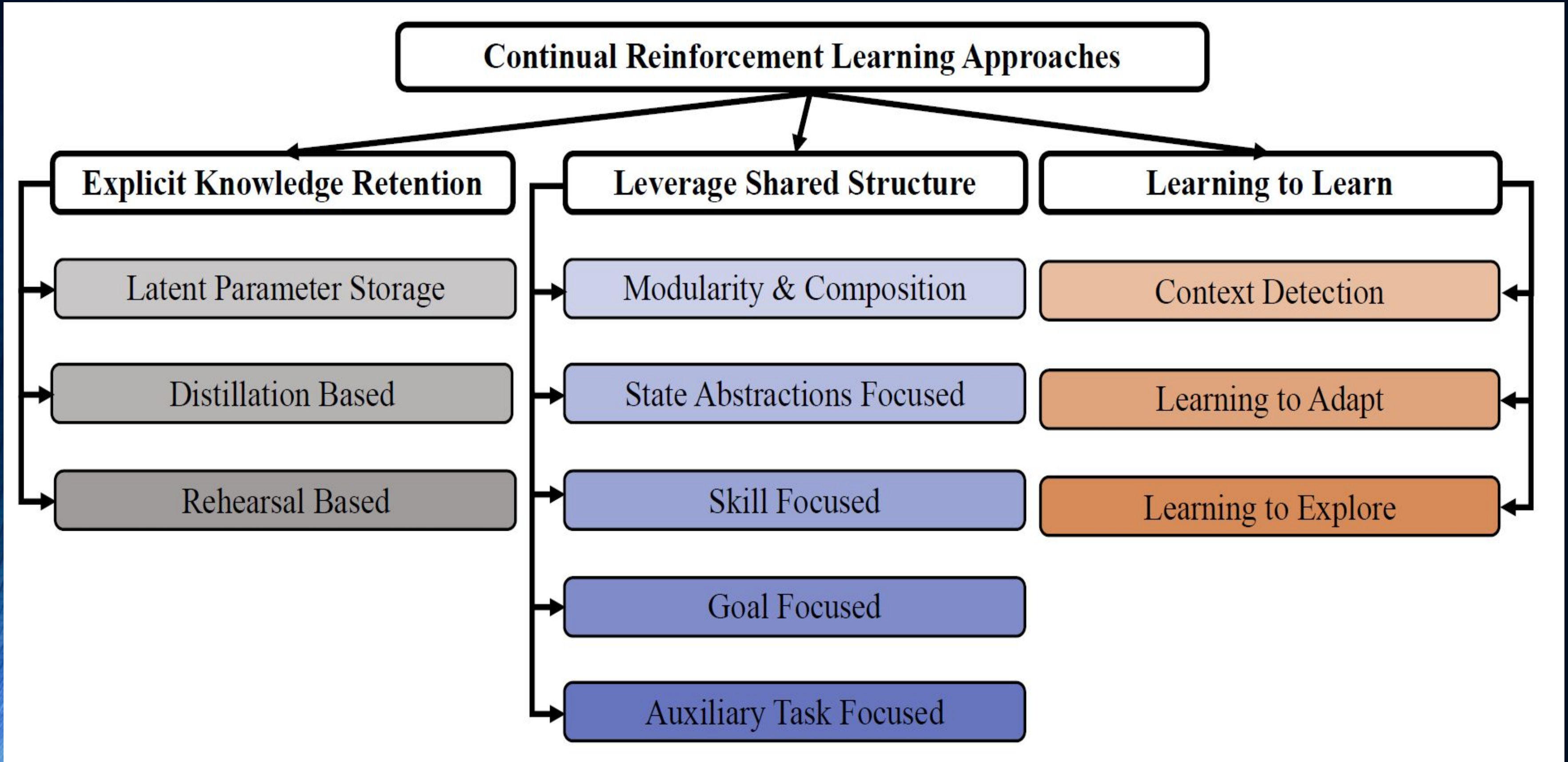
- ❒ The evolution of tasks depends on a stochastic function P(z'|z) as in without having to consider the effects of our own changing policy on this distribution

Active Non-stationarity: In active non-stationary environments, we consider that the agent's behavior may have an impact on the nature of the non-stationarity in the environment.

- ❒ Eg: Intrinsic motivation, curriculum learning

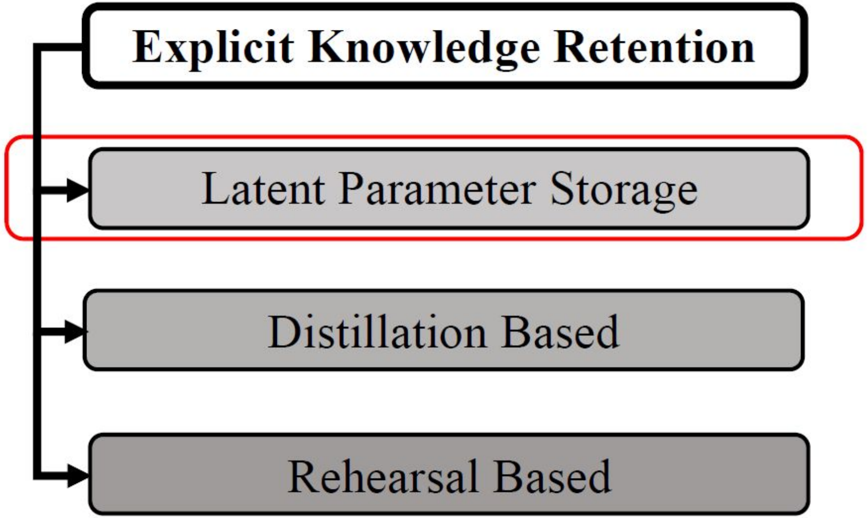
- ❒ This setting is foundational to work studying intrinsic motivation or the agent setting its own curriculum.

Hybrid Non-stationarity: Combining both active and passive sources of non-stationarity.


Continual Reinforcement Learning Approaches
Explicit Knowledge Retention
Leverage Shared Structure
Learning to Learn
Latent Parameter Storage
Distillation Based
Rehearsal Based
Modularity & Composition
State Abstractions Focused
Skill Focused
Goal Focused
Auxiliary Task Focused
Context Detection
Learning to Adapt
Learning to Explore

Explicit Knowledge Retention

- Latent Parameter Storage



Shared components: Ammar et al., 2014 : shared latent basis that captures reusable components

Borsa et al., 2016: explicitly model a shared abstraction of state-action space for multi-task setting

Prior representations: Rusu et al., 2016: provide representations of networks trained on previous tasks as inputs for subsequent tasks.

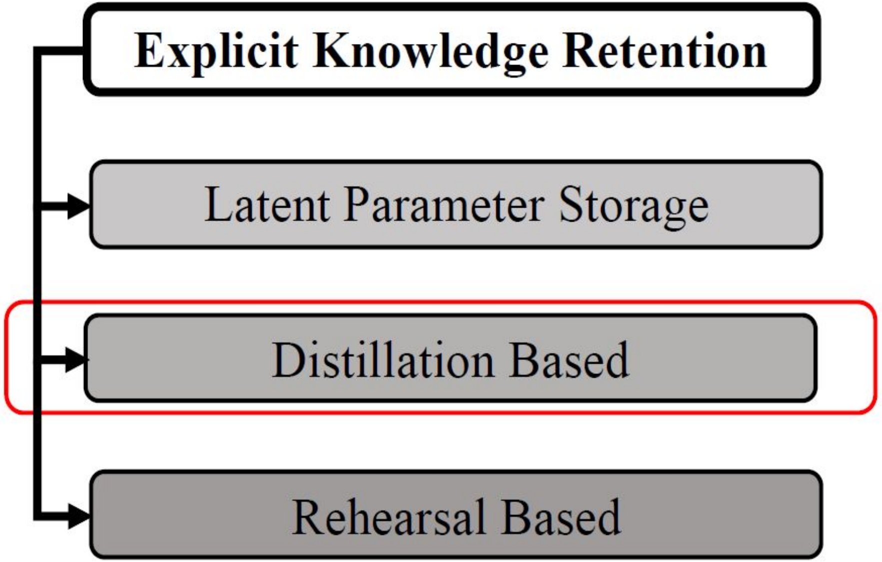
Kirkpatrick et al., 2017: store a prior about the extent of past usage of each parameter during learning to preserve important old knowledge

Single shared representation: Maurer et al. 2016, extracted features for multiple tasks in a single low-dim shared representation

D'Eramo et al. 2016 derived theoretical bounds for AVI and API showing that learning a shared representation significantly decreases the error propagation.

Explicit Knowledge Retention

❒ **Distillation Based**



Idea: Leveraging knowledge distillation which dates back to Bucilua et al., 2006 and renewed interested and success in Hinton et al. 2015

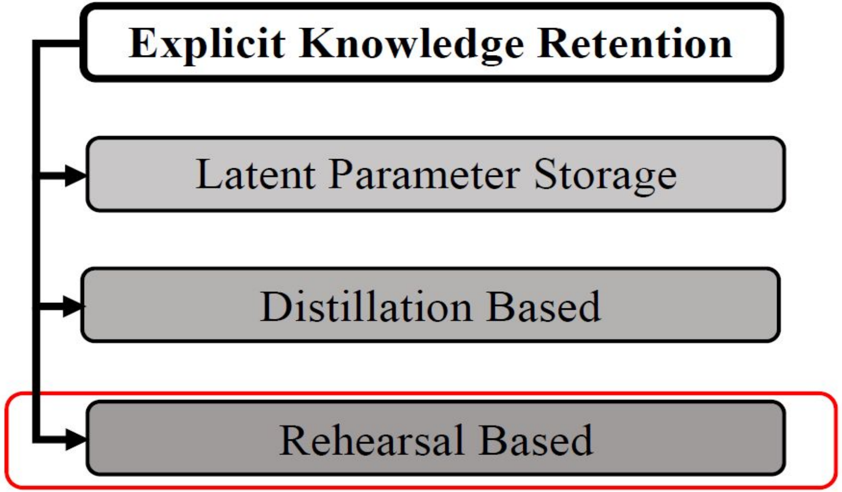
Distill knowledge from past policies when learning a new task as in Rusu et al., 2015; Li and Hoiem, 2016; Riemer et al., 2016; Espeholt et al., 2018; Schwarz et al., 2018; Berseth et al., 2018; Kaplanis et al., 2019; Traoré et al., 2019; Tirumala et al., 2019.

An additional benefit: ultimately learns a separate model for each task.

However, this results in a need for a knowledge compression strategy to scale truly to many many tasks.

Explicit Knowledge Retention

❐ **Rehearsal Based**



Idea: Reinforce the importance of experiences from the past distribution using experience replay
(Lin, 1992)

Replay experiences can help correct the bias in our objective function towards the short term to the extent that the past is a good proxy for the future.
A very successful approach for tackling continual RL as shown in (Isele and Cosgun, 2018; Riemer et al., 2019; Rolnick et al., 2019).

Replay might result in significant storage requirements, and it is not always clear how to prioritize data in replay (the length, the recency, rewards).

Besides they struggle to effectively leverage past data if the shift in distribution is drastic.

Idea: replace replay with *pseudo-rehearsals* sampled from a trained generative model of the environment.

Explored in Robins, 1995, Atkinson et al.,2018

On Evaluation of Continual RL Agents: Benchmarks

A desired CRL Benchmark should allow for

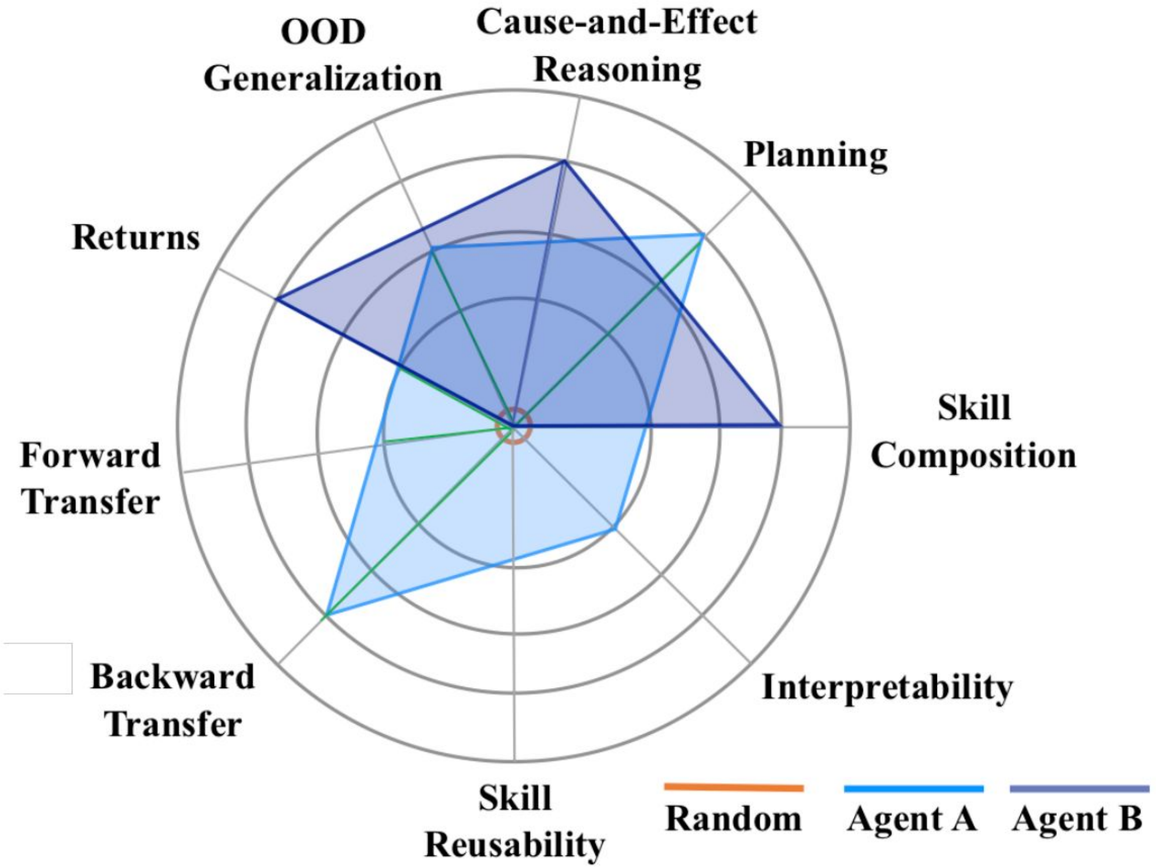
- ❐ a range of degrees of non-stationarity (from 1 to 4)
- ❐ training in progressive fashion
- ❐ discovery and composition of skills
- ❐ generate more complex tasks/scenarios with increasing difficulty
- ❐ learning casual relationships including affordances associated with objects
- ❐ embodied agents
- ❐ rich parallel streams of data which are multi modal

Procgen: A benchmark for procedurally generated set of environments to measure generalization.

Open Problems and Challenges

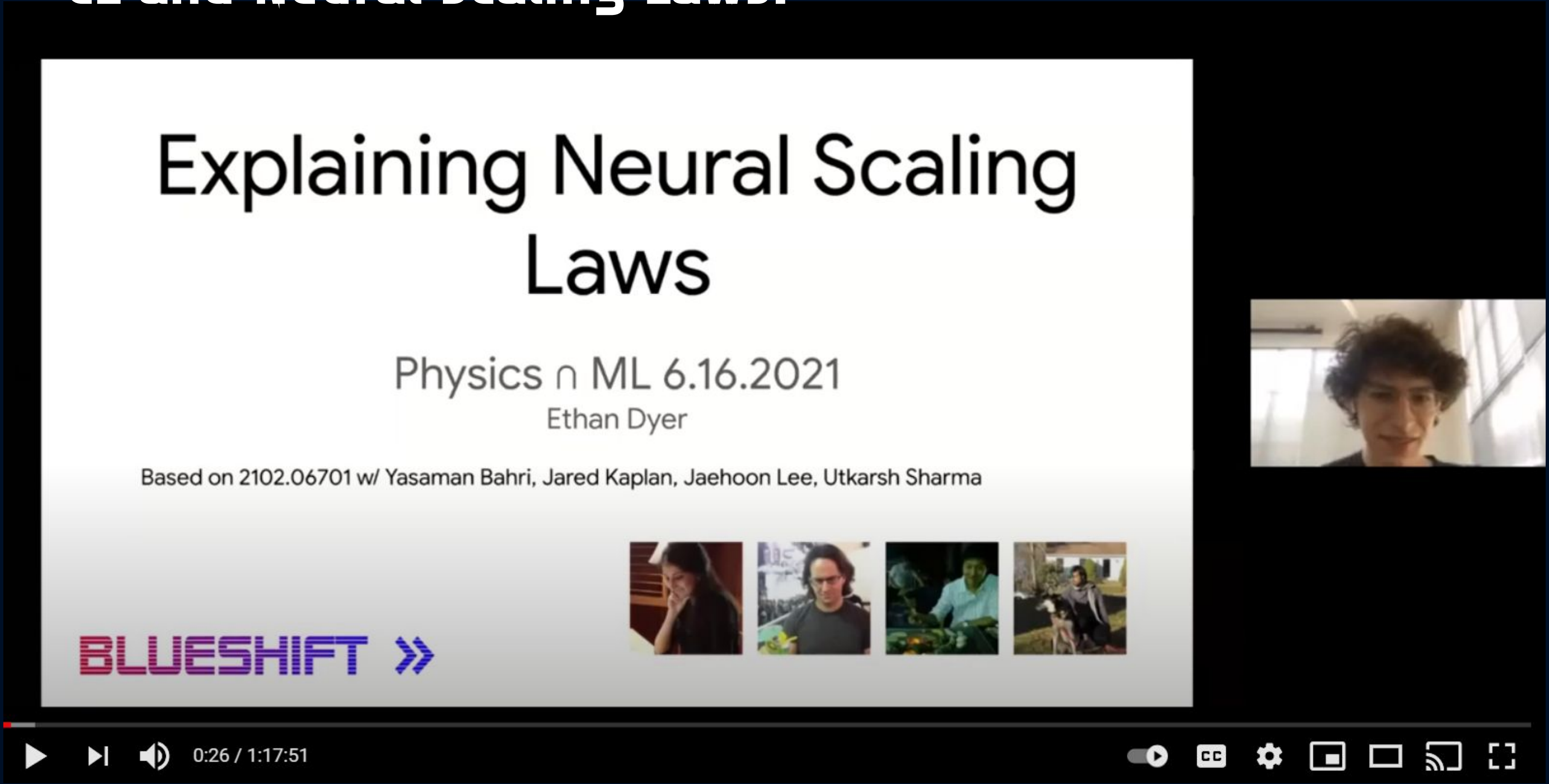
- Finding the right inductive biases
- Task specification and formalism
- Understanding the agent-environment boundary
- Experimental design and evaluation i.e. training and testing
- Interpreting discovered behaviors
- Learning at scale – scaling laws for Continual RL?

Key Metrics for Continual RL



CL and Neural Scaling Laws?

https://youtu.be/V8FEFw50lg4



Neural Scaling Laws

Kaplan et al (2020). Scaling laws for neural language models.

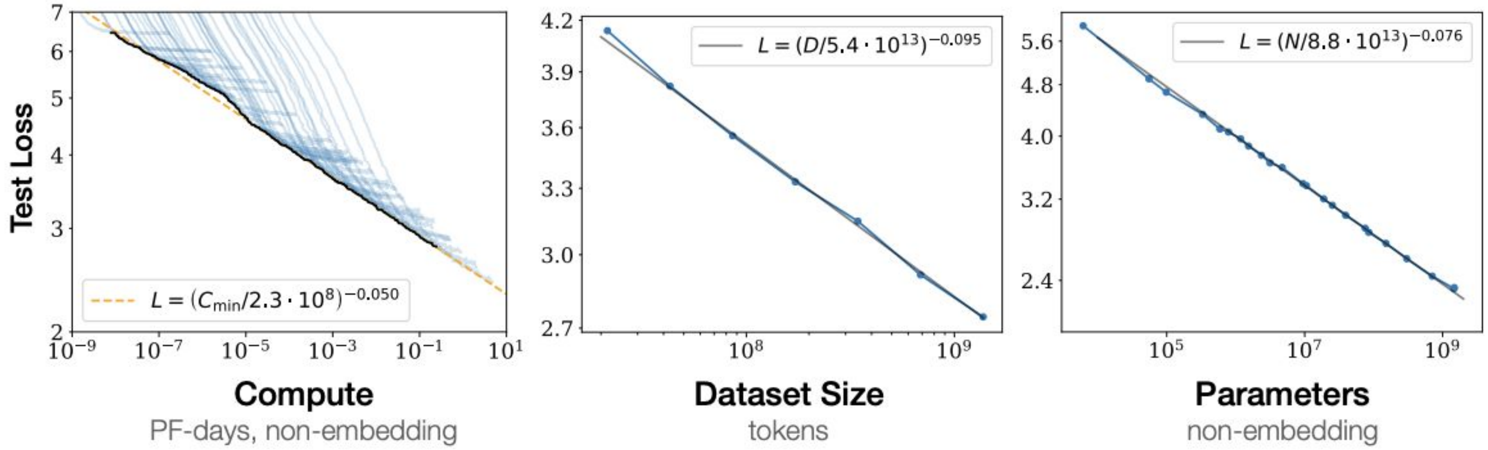
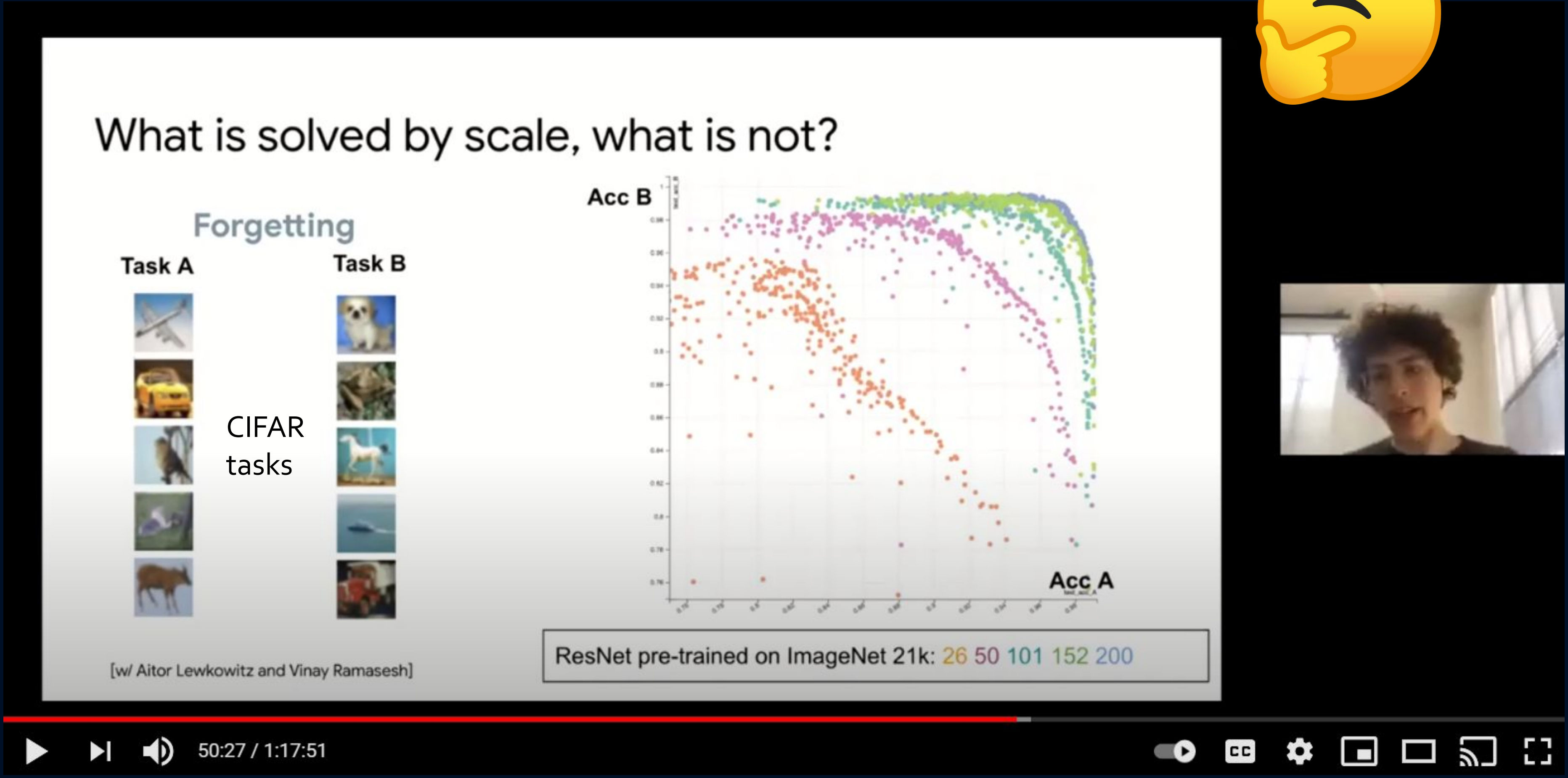


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

What aspects of models, algorithms and data have the largest effect on scaling laws?
How can we design learning approaches with better scaling (e.g., scaling exponent)?

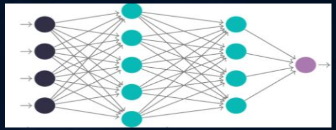
Can Scaling Solve Catastrophic Forgetting?



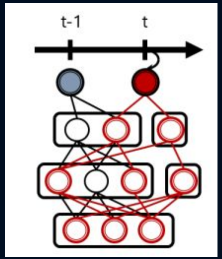
Environment Complexity vs Model Capacity

Representation Capacity

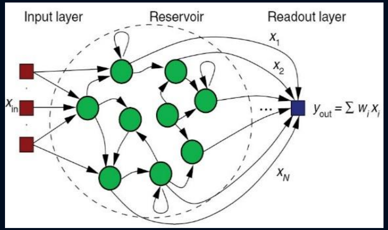
Fixed-size ANNs



Expanding ANNs:



3rd-generation ANNs?



Space-time coding?

Environment Complexity

Summary: Desirable Properties of CL Systems

- Constant memory (infinite data/task stream)
- No task boundary info
- Online learning (without offline training on large batches/tasks)
- Forward transfer (e.g., OOD generalization)
- Backward transfer (beyond not forgetting)
- Problem agnostic (e.g., not limited to classification)
- Adaptively learning from any partial data (e.g., semi-supervised)
- No test time oracle
- Task revisiting to strengthen prior knowledge
- Graceful forgetting (compression) to balance stability and plasticity

Recent Surveys on Continual Learning

Hadsell et al. (2020) Embracing Change: Continual Learning in Deep Neural Networks.

Khetarpal et al. (2020). Towards Continual Reinforcement Learning: A Review and Perspectives.

Mundt et al. (2020) A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning.

De Lange et al. (2019) Continual learning: A comparative study on how to defy forgetting in classification tasks.

Parisi et al. (2019) Continual lifelong learning with neural networks: A review.

Chen & Liu (2018). Lifelong Machine Learning.

Soltoggio et al. (2017) Born to learn: the inspiration, progress, and future of evolved plastic artificial neural networks.

Thank you!

10 minutes

Break

Future Directions and Open Challenges

Continual Learning with Deep Architectures
Tutorial @ ICML 2021 – Part 2

Vincenzo Lomonaco
University of Pisa & ContinualAI
vincenzo.lomonaco@unipi.it

Vincenzo Lomonaco

Assistant Professor @
University of Pisa

Co-founding President and *Lab Director @* ContinualAI.org

Co-founder & Board Member @ AIforPeople.org

Focus on Deep Neural Networks

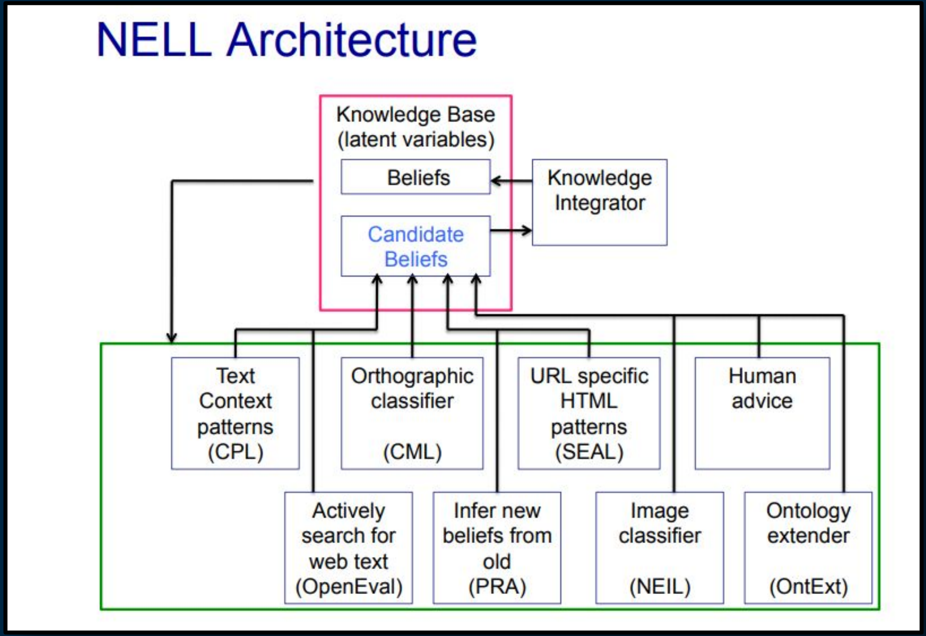
TABLE OF CONTENTS

Continual Unsupervised Learning

NELL: Never-Ending Language Learning

Key Ideas

- *Semi-Supervised Learning System*
- Ran 24x7, from January, 2010 to September 2018
- Combination of many learning algorithms (CPL, CML,SEAL, OpenEval, PRA, NEIL)
- Intended as a case-study for a **never-ending agent**



T. Mitchell et al. ***Never-Ending Learning***. Communications of the ACM, 2018.

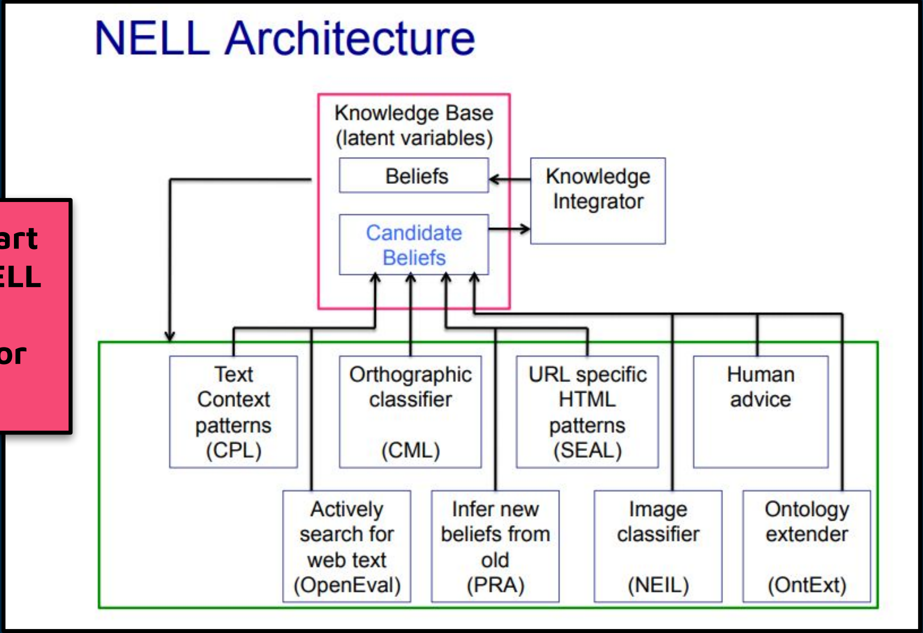
NELL: Never-Ending Language Learning

Key Ideas

- *Semi-Supervised Learning System*

In his 2019 book "Human Compatible", Stuart Russell commented that "Unfortunately NELL has confidence in only 3% of its beliefs and relies on human experts to clean out false or meaningless beliefs on a regular basis."

- Intended as a case-study for a **never-ending agent**



T. Mitchell et al. ***Never-Ending Learning***. Communications of the ACM, 2018.

NEIL: Extracting Visual Knowledge from Web Data

Key Ideas

- *Semi-Supervised Learning System*
- **Cumulative approach**: not incremental, SVM as main learning algorithm
- Feature: GIST, SIFT, HOG, Lab color space, and Texton
- 2.5 months on 200 core cluster: 16 iterations, 400K self-labeled instances, 1152 object, categories, 1034 scene categories

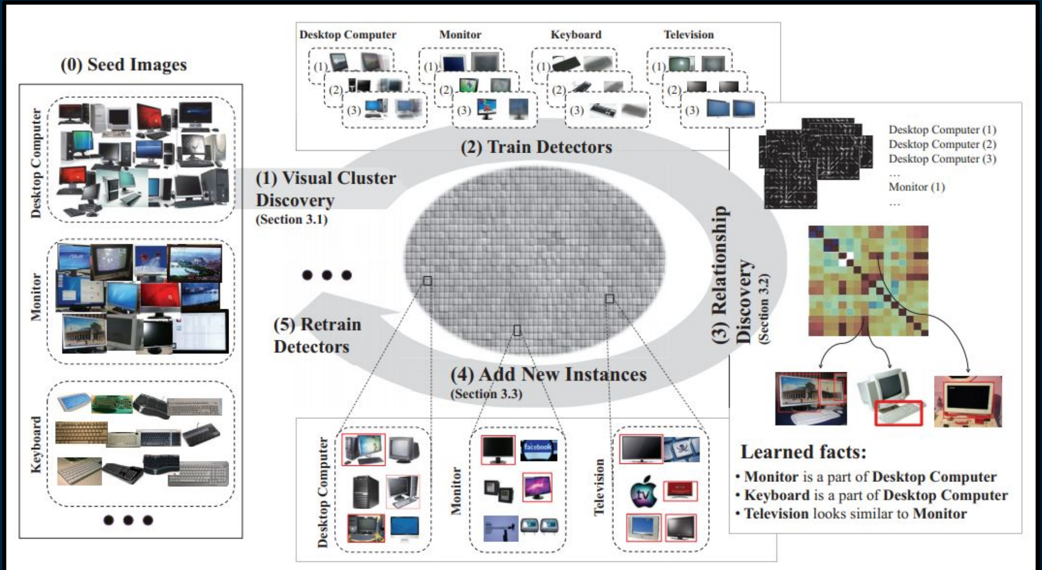


Figure 2. Outline of Iterative Approach

X. Chen et al. ***NEIL: Extracting Visual Knowledge from Web Data***. ICCV, 2013.

Lifelong Topic Modeling

Key Ideas

- *Traditionally an unsupervised learning task.*
- **The "topics" produced by topic modeling techniques are clusters of similar words.**
- Set of shared words among some topics generated from multiple domains are **more likely to be coherent** for a particular topic.
- **Focus**: knowledge accumulation rather than than learning an incremental function

Algorithm 1 PriorTopicsGeneration(D)

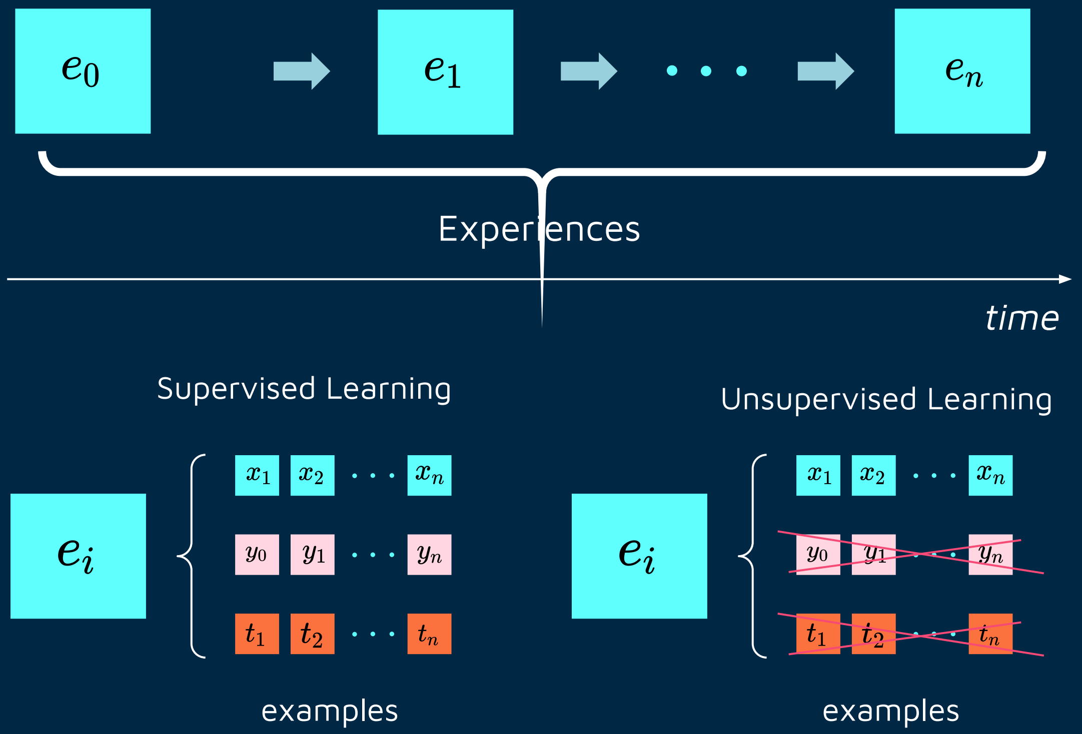
for $r = 0$ **to** R **do**
    for each domain corpus $D_i \in D$ **do**
        if $r = 0$ **then**
            $S_i \leftarrow \mathrm{LDA}(D_i)$;
        else
            $S_i \leftarrow \mathrm{LTM}(D_i, S)$;
        end if
    end for
    $S \leftarrow \cup_i S_i$;
end for

Algorithm 2 LTM(D^t, S)

$A^t \leftarrow$ GibbsSampling(D^t, $\emptyset$, N); // Run N Gibbs iterations with no knowledge (equivalent to LDA).
for $i = 1$ **to** N **do**
    $K^t \leftarrow$ KnowledgeMining(A^t, S);
    $A^t \leftarrow$ GibbsSampling(D^t, K^t, 1); // Run with knowledge K^t.
end for

Z. Chen et al. ***Topic modeling using topics from many domains, lifelong learning and big data***. ICML, 2014.
P. Gupta et al. **Neural topic modeling with continual lifelong learning**. ICML, 2020.

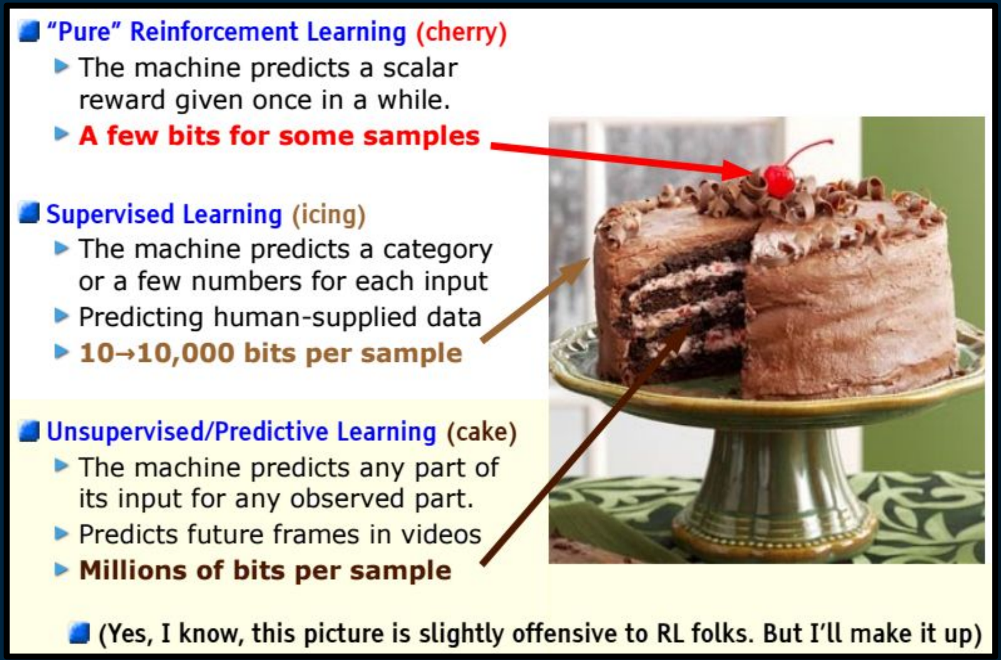
Continual Unsupervised Learning



Continual Unsupervised Learning

Ideal Paradigm to Combine with CL

- **No Continual Labeling**
- **Less Bias**
- **Why this is still not the case?**
    - *Changing the paradigm*: More Data, Less Supervision
    - Less impactful applications (for now)



Y. LeCun, NeuIPS 2016

Continual Unsupervised Representation Learning

Key Ideas

- **Fully Generative Approach**
- **y** can be interpreted as representing some discrete clusters in the data
- **Mixture of Gaussian with Dynamic Expansion**
- *Difficult to scale*: tested only on MNIST and Omniglot

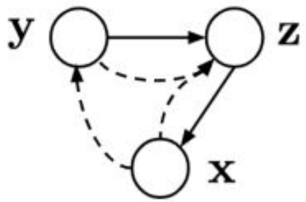


Figure 1: Graphical model for CURL. The categorical task variable **y** is used to instantiate a latent mixture-of-Gaussians **z**, which is then decoded to **x**.

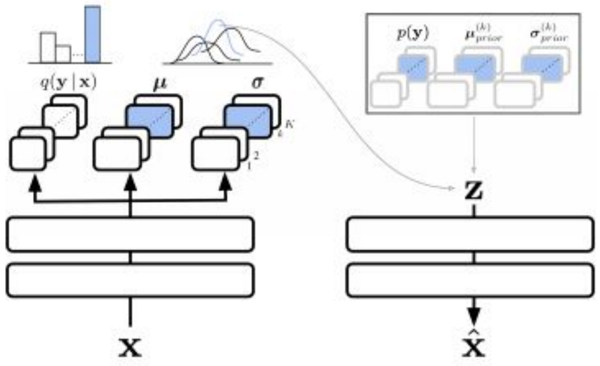


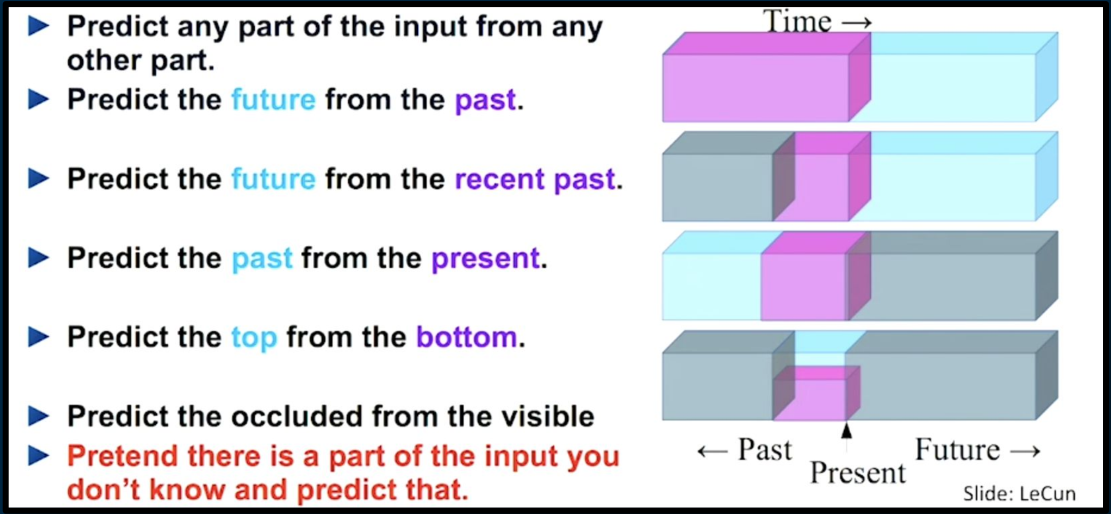
Figure 2: Diagram of the proposed approach, showing the inference procedure and architectural components used.

D. Rao et al. ***Continual Unsupervised Representation Learning***. NeurIPS 2019.

Continual Unsupervised Learning

Huge Exploration Opportunities

- **Self-Supervised Learning**
- Sequence Learning
- Contrastive Learning
- Hebbian-like Learning
- Active Learning
- Weakly/Semi-Supervised Learning
- Randomized Networks

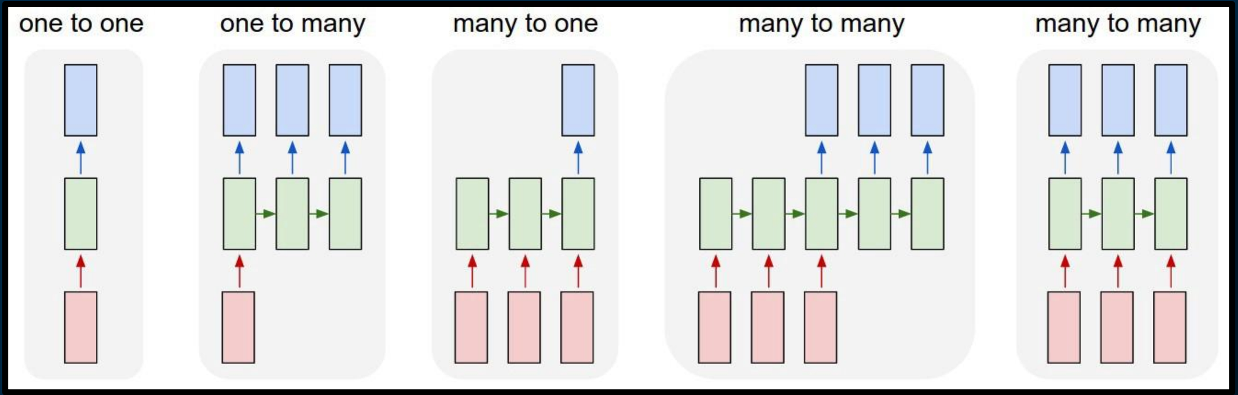


S. Zhang et Al. ***Self-Supervised Learning Aided Class-Incremental Lifelong Learning***. CLVision Workshop Findings at CVPR, 2021.
J. Gallardo et Al. ***Self-Supervised Training Enhances Online Continual Learning***. arXiv, 2021.

Continual Unsupervised Learning

Huge Exploration Opportunities

- Self-Supervised Learning
- **Sequence Learning**
- Contrastive Learning
- Hebbian-like Learning
- Active Learning
- Weakly/Semi-Supervised Learning
- Randomized Networks



Y. Cui et al. ***Continuous online sequence learning with an unsupervised neural network model***. *Neural Computation, 2016.*
A. Cossu et Al. ***Continual Learning for Recurrent Neural Networks: an Empirical Evaluation***. Elsevier Neural Networks, 2021.
B. Ehret et Al. ***Continual learning in Recurrent Neural Networks.*** ICLR 2021.

Continual Unsupervised Learning

Huge Exploration Opportunities

- Self-Supervised Learning
- Sequence Learning
- **Contrastive Learning**
- Hebbian-like Learning
- Active Learning
- Weakly/Semi-Supervised Learning
- Randomized Networks

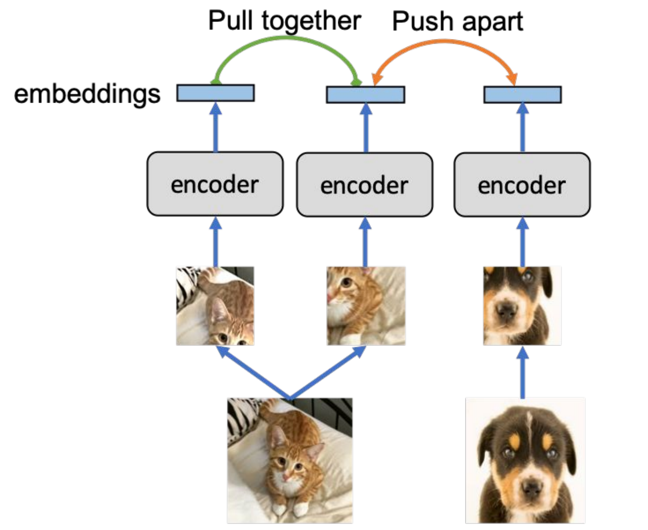


Figure 1 (a)

Junnan Li, 2020

M. Zheda et al. ***Supervised Contrastive Replay: Revisiting the Nearest Class Mean Classifier in Online Class-Incremental Continual Learning***. CLVision Workshop at CVPR 2021.
C. Hyuntak et al. ***CO2L: Contrastive Continual Learning***. arXiv, 2021.

Continual Unsupervised Learning

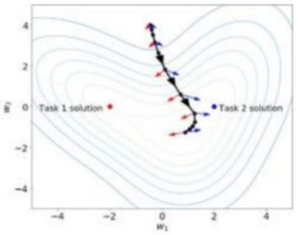
Huge Exploration Opportunities

- Self-Supervised Learning
- Sequence Learning
- Contrastive Learning
- **Hebbian-like Learning**
- Active Learning
- Weakly/Semi-Supervised Learning
- Randomized Networks

Gradient-based optimization and tug-of-war dynamics

- Continual Learning is a huge challenge for deep learning models because of **gradient-based optimization**.
- Gradient-based learning is effective and cheap, the de rigeur method for training neural networks for close to 4 decades.
- However, a close look at the learning dynamics reveals a problem.
- Each training sample produces a gradient for each parameter in the network that votes to make the parameter bigger or smaller.
- In a mini-batch, a gradient is produced by each sample in parallel and they are summed to decide the winning direction.

➡ The result is a **tug-of-war** over the direction of change of each parameter.



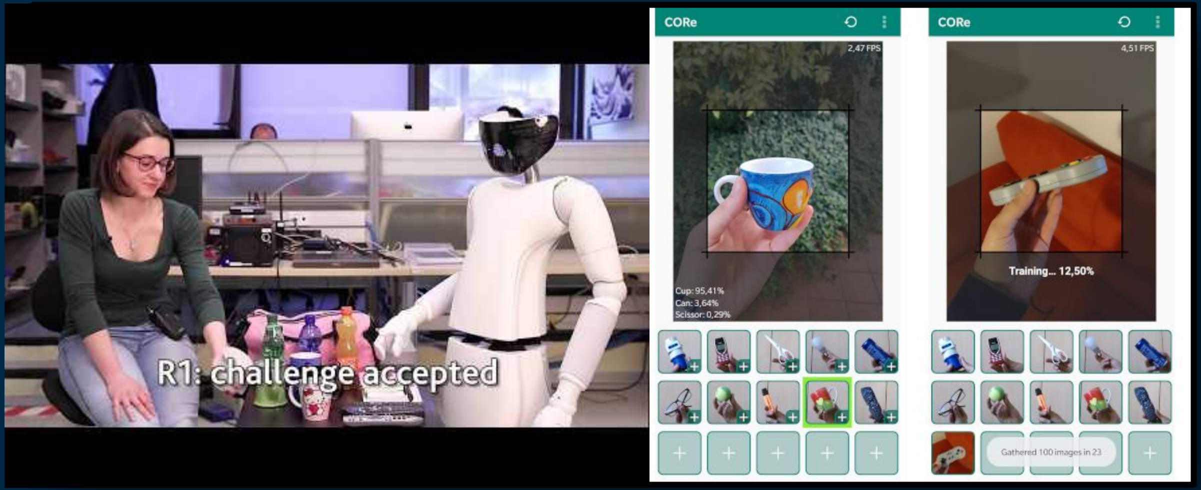
R. Pascanu, 2021

G. Parisi et al. ***Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization***. Frontiers in neurorobotics, 2018.
P. Bashivan et al. ***Continual learning with self-organizing maps***. CL Workshop at NeurIPS 2018

Continual Unsupervised Learning

Huge Exploration Opportunities

- Self-Supervised Learning
- Sequence Learning
- Contrastive Learning
- Hebbian-like Learning
- **Active Learning**
- Weakly/Semi-Supervised Learning
- Randomized Networks

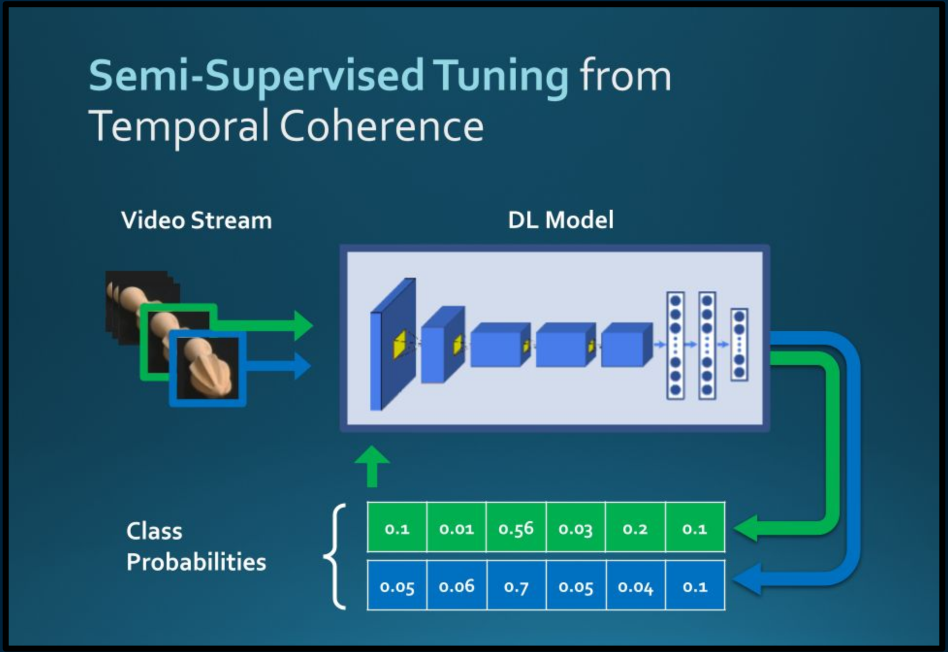


L. Pellegrini et al. ***Continual Learning at the Edge: Real-Time Training on Smartphone Devices***. ESANN, 2021.
R. Camoriano et al. ***Incremental robot learning of new objects with fixed update time***. ICRA, 2017.

Continual Unsupervised Learning

Huge Exploration Opportunities

- Self-Supervised Learning
- Sequence Learning
- Contrastive Learning
- Hebbian-like Learning
- Active Learning
- **Weakly/Semi-Supervised Learning**
- Randomized Networks



Lomonaco V. and Maltoni D. ***Semi-Supervised Tuning from Temporal Coherence***. ICPR 2016.
L. Wang et al. **Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning**. CVPR 2021.

Continual Unsupervised Learning

Huge Exploration Opportunities

- Self-Supervised Learning
- Sequence Learning
- Contrastive Learning
- Hebbian-like Learning
- Active Learning
- Weakly/Semi-Supervised Learning
- **Randomized Networks**

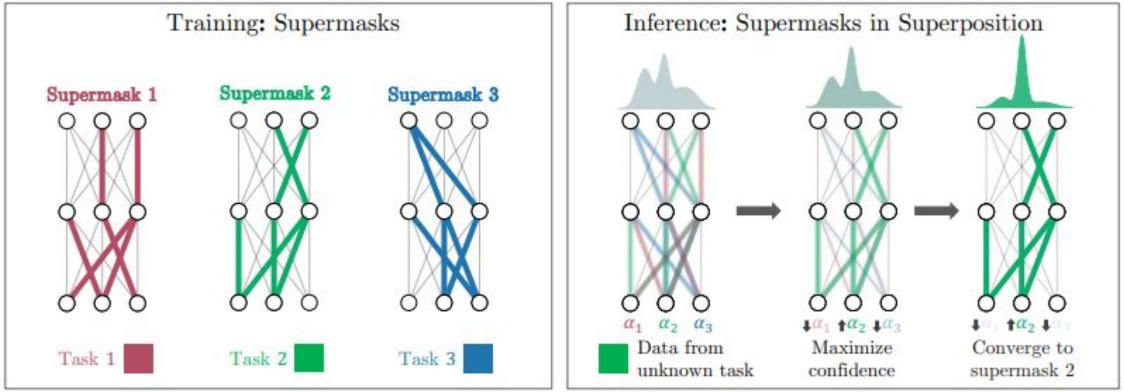


Figure 1: **(left)** During training SupSup learns a separate supermask (subnetwork) for each task. **(right)** At inference time, SupSup can infer task identity by superimposing all supermasks, each weighted by an α_i, and using gradients to maximize confidence.

M. Wortsman, 2020

A. Cossu et al. ***Continual Learning with Echo State Networks***. ESANN 2021.
M. Wortsman et al. ***Supermasks in superposition***. NeurIPS 2020.

Continual Unsupervised Learning

Other relevant works in this area

- A. Bertugli et al. ***Few-Shot Unsupervised Continual Learning through Meta-Examples***. Workshop on Meta-Learning at NeurIPS 2020.
- I. Muñoz-Martín et al. ***Unsupervised learning to overcome catastrophic forgetting in neural networks***. IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, 2019.
- L. Caccia et al. ***SPeCiaL: Self-Supervised Pretraining for Continual Learning***, arXiv 2021.
- W. Sun et al. ***ILCOC: An Incremental Learning Framework Based on Contrastive One-Class Classifiers.*** CLVision Workshop at CVPR 2021.
- J. He et al. ***Unsupervised Continual Learning Via Pseudo Labels.*** arXiv 2020.
- S. Khar et al. ***Unsupervised Class-Incremental Learning through Confusion***. arXiv 2021.

Continual Learning

Applications

Continual Learning Applications

Main Possibilities

- **Edge**
  - **Embedded systems and Robotics:** +privacy, +efficiency, +fast adaptation, +on the edge, -Internet connection (e.g. Autonomous Cars, Robotics Arms/Hands)
- **Cloud**
  - **AutoML and CI systems for AI models:** +scalability, +efficiency, +fast adaptation, -energy consumption, -$$$ (e.g. Recommendation Systems)
- **Continuum Edge-Cloud**
  - **Pervasive AI systems**: Efficient Communication, fluid & dynamic computation
  - **Neural Patches**: +security patches, +fairness patches, +fast update
  - **Continual Distributed Learning**: understudied relationship with parallel and federated learning

On-Device Personalization without Forgetting

L. Pellegrini et al. ***Latent Replay for Real-Time Continual Learning***, IROS 2020.
L. Pellegrini et al. ***Continual Learning at the Edge: Real-Time Training on Smartphone Devices***. ESANN, 2021.
G. Demosthenous et al. ***Continual Learning on the Edge with TensorFlow Lite***. *arXiv 2021.*
L. Ravaglia et al. ***Memory-Latency-Accuracy Trade-offs for Continual Learning on a RISC-V Extreme-Edge Node***. *SiPS 2020.*

AR1: a Flexible Hybrid Strategy for CL

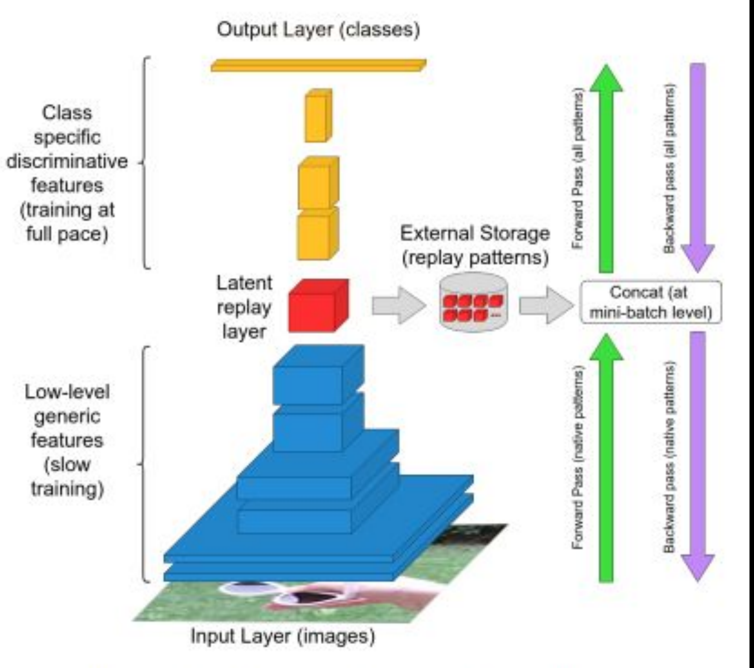


Figure 1: Architectural diagram of Latent Replay.

Key Ideas

- ***Architectural, Regularization and Replay components:***
    - *CWR* for the output layer (arch)*
    - *Online Synaptic Intelligence (reg)*
    - *Latent Replay (replay)*

$$\tilde{L}_\mu = L_\mu + \lambda \sum_k \Omega_k^\mu \left(\bar{\theta}_k - \theta_k\right)^2$$

$$w_k^v = \int_{t^{\mu-1}}^{t^\mu} \frac{\partial L}{\partial \theta_k} \cdot \frac{\partial \theta_k}{\partial t}$$

D. Maltoni et al. ***Continuous Learning in Single-Incremental-Task Scenarios***, Neural Networks, 2019.
L. Pellegrini et al. ***Latent Replay for Real-Time Continual Learning***, IROS 2020.
V. Lomonaco et al. ***Rehearsal-Free Continual Learning over small I.I.D Batches***. CLVision at CVPR 2020.

Continual Learning in Production

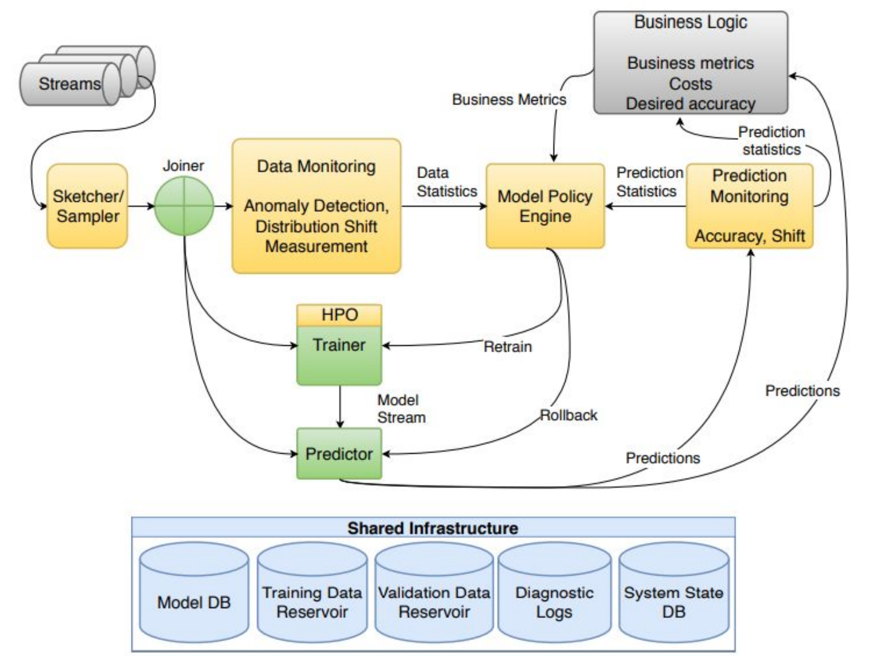
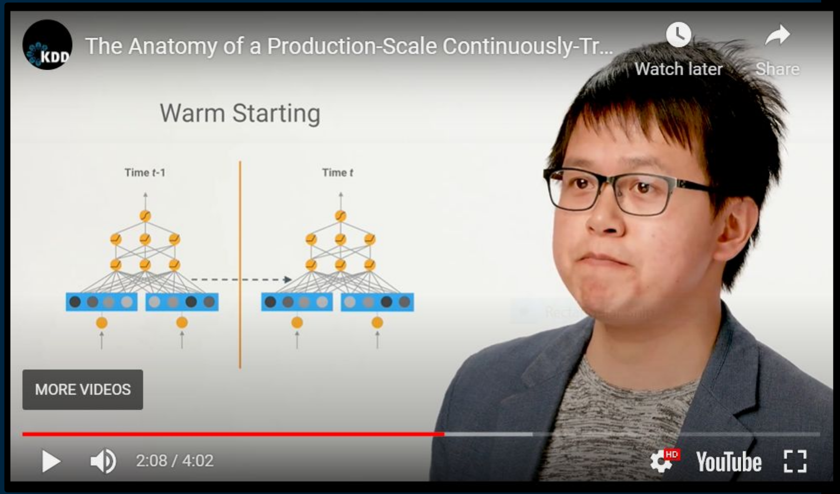
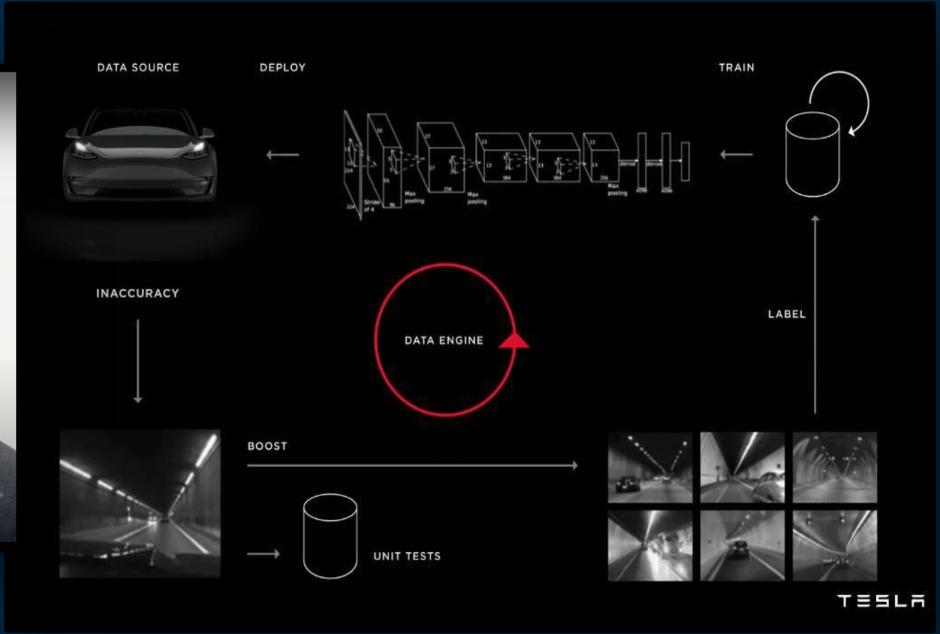


Figure 1: Data flow in the Auto-Adaptive Machine Learning architecture. See text for details.

T. Diethe et al. ***Continual Learning in Practise***. Continual Learning Workshop at NeurIPS 2018.

Use-Cases: Google Play and Tesla





V. Lomonaco. ***Continual Learning for Production Systems: The new "Agile" in the Machine Learning Era***. ContinualAI Publication, 2019.
D. Baylor et al. ***FX: A TensorFlow-Based Production-Scale Machine Learning Platform***. KDD, 2017.
A.Karpathy. ***Building the Software 2.0 Stack***. Spark+AI Summit, 2018.

Some Startups: Cogitai, Neurala, Gantry

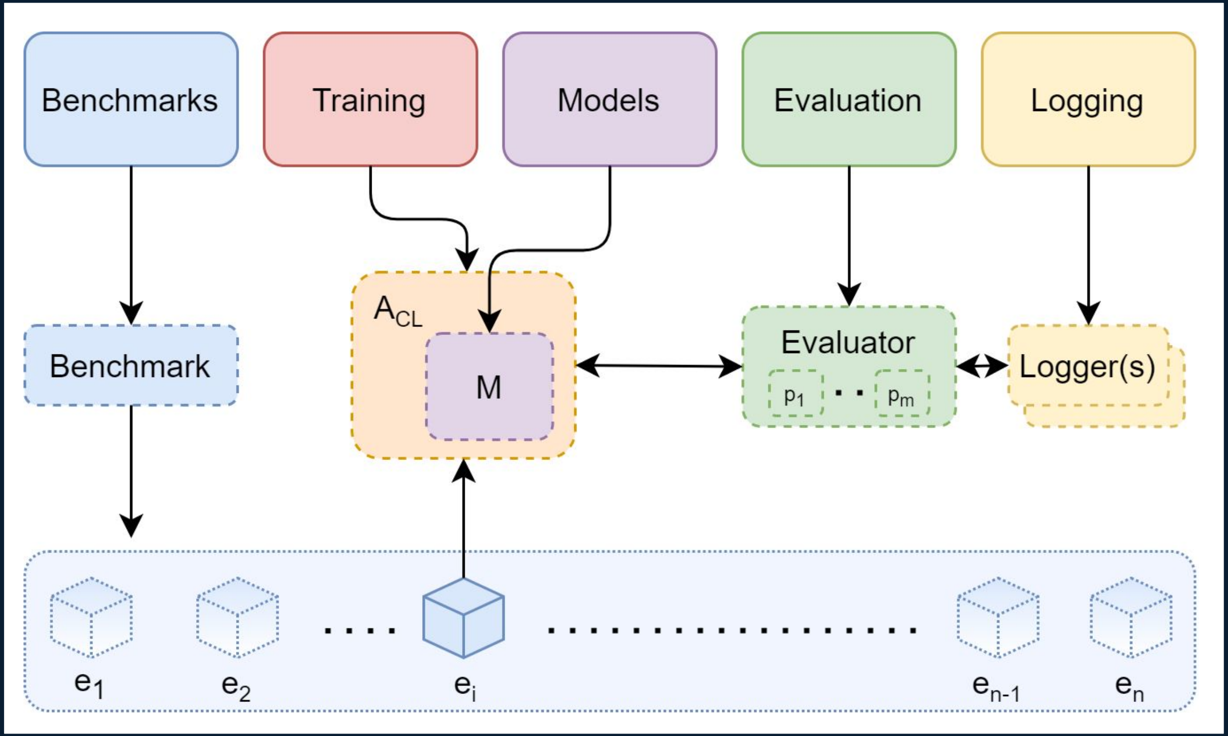
https://www.neurala.com
https://gantry.io
https://www.cogitai.com

Continual Learning Tools

Research & Development Tools

- V. Lomonaco et al. ***Avalanche: an End-to-End Library for Continual Learning***. CLVision Workshop at CVPR 2021.
- A. Douillard et al. ***Continuum: Simple management of complex continual learning scenarios***. CLVision Workshop at CVPR 2021.
- S.I. Mirzadeh et al. ***CL-Gym: Full-Featured PyTorch Library for Continual Learning***. CLVision Workshop at CVPR 2021.
- F. Normandin et al. ***Sequoia: A Software Framework to Unify Continual Learning Research***. CLVision Workshop at CVPR 2021.

Avalanche for R&D



V. Lomonaco et al. ***Avalanche: an End-to-End Library for Continual Learning***. CLVision Workshop at CVPR 2021.

Avalanche for R&D

Avalanche Key Links

- ***Avalanche Official Website***: https://avalanche.continualai.org
- ***Avalanche GitHub***: https://github.com/ContinualAI/avalanche
- ***Avalanche API-DOC***: https://avalanche-api.continualai.org
- ***Avalanche ContinualAI Slack***: #avalanche channel

With Avalanche | Without Avalanche

```python
import torch
from torch.nn import CrossEntropyLoss
from torch.optim import SGD

from avalanche.benchmarks.classic import PermutedMNIST
from avalanche.extras.models import SimpleMLP
from avalanche.training.strategies import Naive

# Config
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

# model
model = SimpleMLP(num_classes=10)

# CL Benchmark Creation
perm_mnist = PermutedMNIST(n_experiences=3)
train_stream = perm_mnist.train_stream
test_stream = perm_mnist.test_stream

# Prepare for training & testing
optimizer = SGD(model.parameters(), lr=0.001, momentum=0.9)
criterion = CrossEntropyLoss()

# Continual learning strategy
cl_strategy = Naive(
    model, optimizer, criterion, train_mb_size=32, train_epochs=2,
    eval_mb_size=32, device=device)

# train and test loop
results = []
for train_task in train_stream:
    cl_strategy.train(train_task, num_workers=4)
    results.append(cl_strategy.eval(test_stream))
```

V. Lomonaco et al. ***Avalanche: an End-to-End Library for Continual Learning***. CLVision Workshop at CVPR 2021.

Avalanche for R&D

```
replay = ReplayPlugin(mem_size)
ewc = EWCPlugin(ewc_lambda)
strategy = BaseStrategy(
    model, optimizer,
    criterion, mem_size,
    plugins=[replay, ewc])
```

V. Lomonaco et al. ***Avalanche: an End-to-End Library for Continual Learning***. CLVision Workshop at CVPR 2021.

Impact on Sustainable AI

Sustainable AI Principles

General Principles

- **Accuracy & Robustness**
- **Explainability, Transparency & Accountability**
- **Bias**
- **Fairness**
- **Privacy & Security**
- **Human, Social and Environmental Wellbeing**

L. Royakkers et al. ***Societal and ethical issues of digitization***. Ethics and Information Technology, 2018.
B.D. Mittelstadt et al. ***The ethics of algorithms: Mapping the debate***. Big Data & Society, 2016.
A. Jobin et al. ***The global landscape of AI ethics guidelines***. Nature Machine Intelligence, 2019.
https://www.aiforpeople.org/ethical-ai/

Continual Learning Impact

...On each Principle:

- **Accuracy & Robustness ➞** Robustness & Autonomy, Continual & Fast Improvement
- **Bias ➞** CL as the new Agile: Bias Patches
- **Fairness ➞** Efficient Fairness Patches
- **Privacy & Security ➞** Security Patches
- **Human, Social and Environmental Wellbeing ➞** improved efficiency & scalability: less energy consumption, CO_2 emission; sustainable & "progressive" by design
- **Explainability, Transparency & Accountability ➞** Neuroscience-grounded, Human-centered AI

L. Royakkers et al. ***Societal and ethical issues of digitization***. Ethics and Information Technology, 2018.
B.D. Mittelstadt et al. ***The ethics of algorithms: Mapping the debate***. Big Data & Society, 2016.
A. Jobin et al. ***The global landscape of AI ethics guidelines***. Nature Machine Intelligence, 2019.
https://www.aiforpeople.org/ethical-ai/

Open Questions

Open Questions (1/2)

1. Is it possible to learn **robust**, **deep representations** **continually**?
2. Are currently addressed **scenarios** and **eval metrics** **enough**?
3. What is the right **level of supervision**?
4. How to know **what to forget** and **what to remember**?
5. What's the relationship with **concept drift**?
6. Is **replay** a research direction worth pursuing?
7. Is **computation** more important than **memory**?
8. Is **gradient descent** the right algorithm to learn continually?
9. **Continual Meta-Learning** & **Meta-Continual Learning**: what's the right relationship?
10. What is the relationship with **Sequence** and **Continual** **Learning**?

N. Díaz-Rodríguez et al. **Don't Forget, There is More than Forgetting: new Metrics for Continual Learning. CL Workshop at NeurIPS 2018.**
A Prabhu. **Gdumb: A simple approach that questions our progress in continual learning. ECML, 2020.**

Open Questions (2/2)

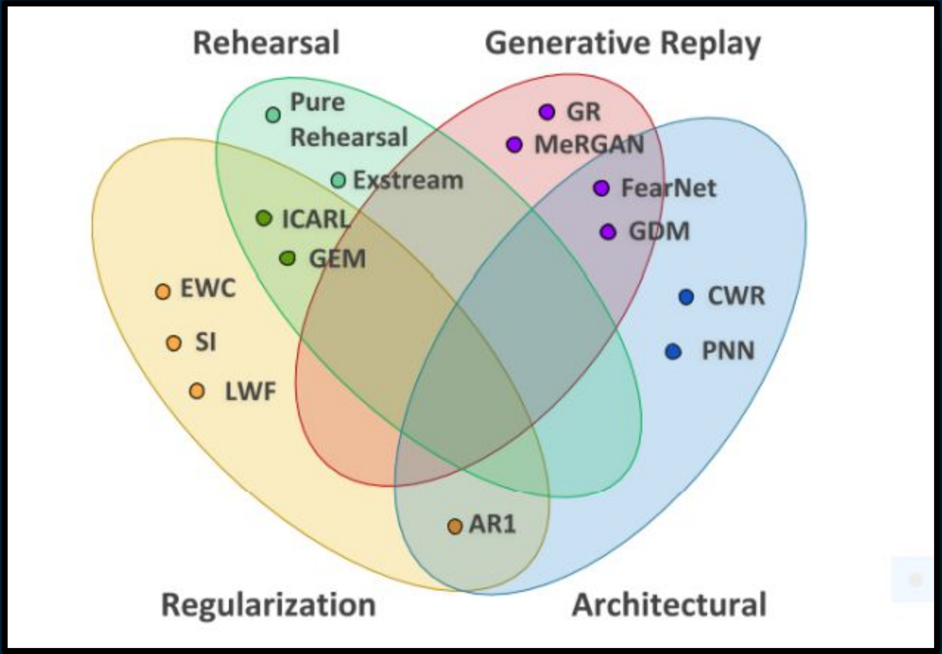
1. Is **curiosity** important for **continual learning**?
2. What about **Curriculum Learning**?
3. **Compositionality** is a key aspect of human intelligence: what to expect for CL Systems?
4. **Self-Reflection***: accuracy of learned functions, given only unlabeled data?
5. **Self-reflection** that can detect every possible shortcoming (called impasse) of the agent*
6. (External) **Knowledge and Reasoning***

…and much more!

*T. Mitchell and P. Talukdar. ***Never-Ending Learning***. Tutorial at ICML 2019.
J.A. Mendez et al. ***Lifelong learning of compositional structures***. ICLR 2021.

On the Future of CL (Short-Medium Term)

1. **More Natural Scenarios**
   - **Domain**, **Task** and **Class-Incremental** are not enough.
   - Longer streams of "*experiences*".
   - More metrics, focus on scalability.
2. **Move towards unsupervised training**
   - Mostly **Semi-Supervised**, **Self-Supervised** and **Sequence Learning**.
3. **Hybrid Continual Learning Strategies**
4. **Continual Learning Applications**



T. Lesort et al. *Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges*. Information Fusion, 2020.
G.I. Parisi et al. ***Online Continual Learning on Sequences.*** Studies in Computational Intelligence, 2020.

On the Future of CL (Long-Term)

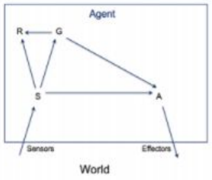
1. Fundamentally a question of **agent architecture***

2. **Two main paths for (deep) CL**

    a. **Neuroscience-Inspired**

    b. **Distributed Continual Learning**

What should a theory of Learning Agents answer?



might model learning agent A as tuple <S,E,M,F,G,L>

- S = sensors
- E = effectors
- F = set of functions
- M = set of memory units
- G = graph specifying data flow among F, M, S, E
- L = learning mechanism

might model L as another agent L = <S_L,E_L,M_L,F_L,G_L>

- where S_L, E_L sense and act on Agent, especially its F, M, G

*T. Mitchell and P. Talukdar. **Never-Ending Learning**. Tutorial at ICML 2019.

Conclusions

Conclusions

What we have seen

- 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** and forgetting centered metrics.
- **Huge space of possible and significant explorations**.

Take-Home Messages

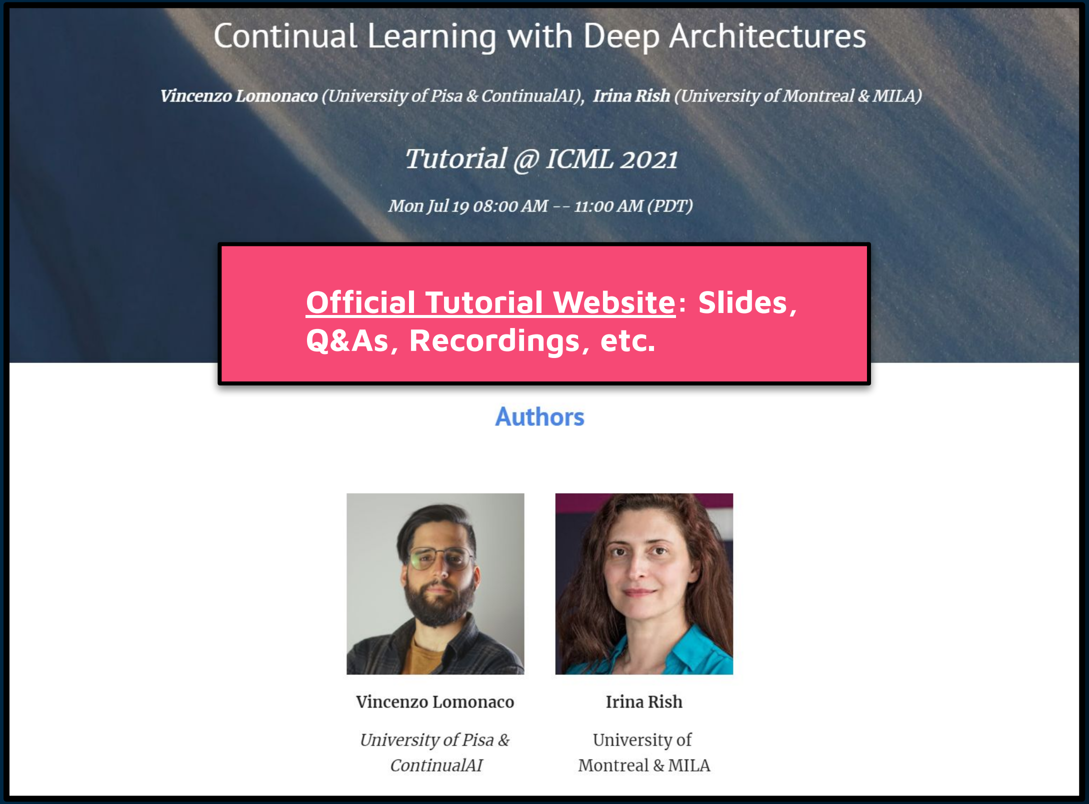
1. Continual Learning is a **paradigm-changing approach** trying to break the fundamental i.i.d. assumption in statistical learning.
2. CL pushes for the **next step in Neuroscience-grounded approaches** to learning
3. CL pushes for the next generation of truly intelligent robust and autonomous AI systems: **efficient, effective, scalable, hence sustainable**.

"We are not looking for incremental improvements in state-of-the-art AI and neural networks, but rather paradigm-changing approaches to machine learning that will enable systems to continuously improve based on experience."

– Hava Siegelmann, 2018

Resources

Additional Resources (1/3)

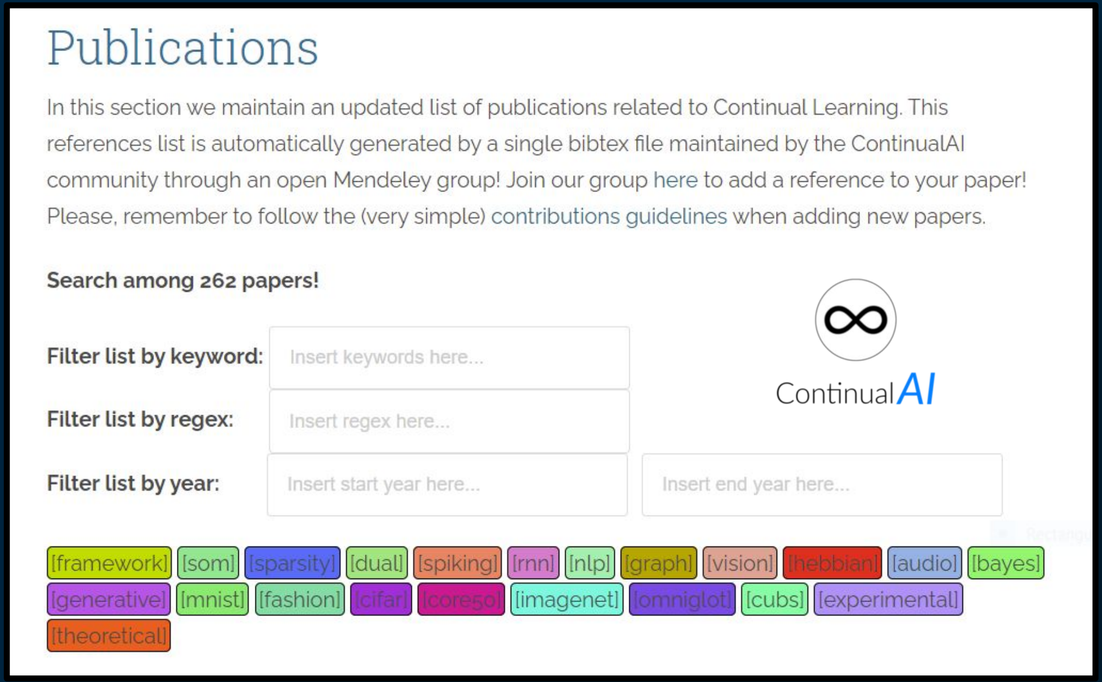


V. Lomonaco, I. Rish. ***Continual Learning with Deep Architectures***. ICML Tutorial, 2021.

ContinualAI.org

Additional Resources (2/3)

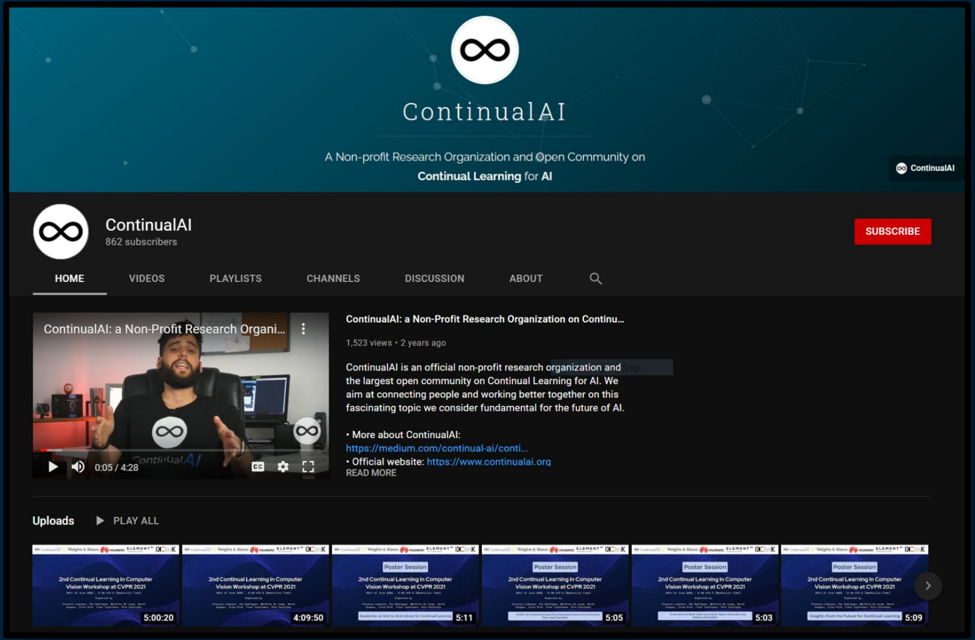
- **ContinualAI Wiki**: a shared and collaboratively maintained *knowledge base* for Continual Learning: tutorials, workshops, demos, tutorials, courses, etc.
- **Continual Learning Papers**: curated list of CL papers & books with meta-data by ContinualAI
- **ContinualAI Forum** + Slack: discussions / Q&As about Continual Learning
- **ContinualAI Research Consortium**: networks of Top CL Labs across the world.



You can find more at: www.continualai.org

Additional Resources (3/3)

- **ContinualAI Publication**: a curated list of original blog posts on CL.
- **Continual Learning & AI Mailing List**+: curated list of CL papers & books with meta-data by ContinualAI.
- **ContinualAI Newsletter**:news from the ContinualAI community and the CL World in one place.
- **ContinualAI Seminars**: weekly invited talks on CL.
- **ContinualAI YouTube**: collection of videos about CL.



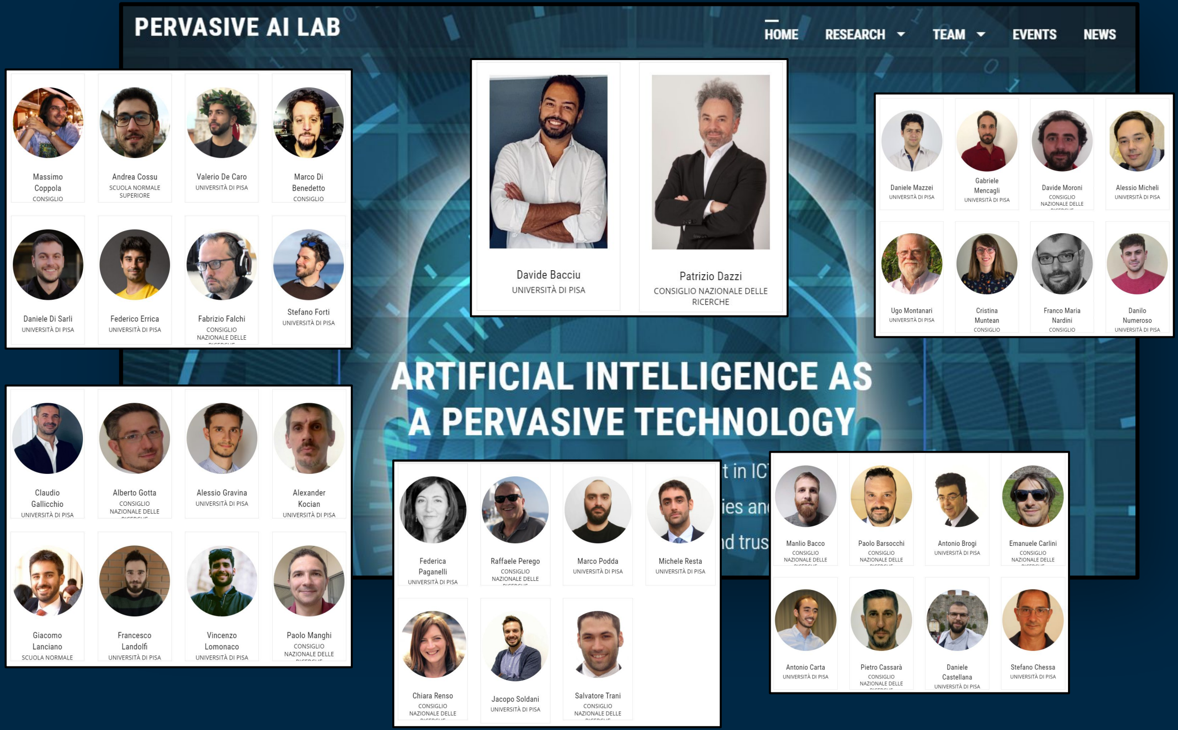
You can find more at: www.continualai.org

Continual Learning: On Machines that Can Learn Continually

1st Official Open-Access Course on CL Offered by **University of Pisa** & **ContinualAI**

01

Join our Pervasive AI Lab!



Teaching & Supervision

Research

Spin-off

Consultancy

The lab is in Pisa, Italy! Feel free to visit and get in touch with us anytime! Official website: Pervasive AI Lab (unipi.it)

Do you have any questions?

vincenzo.lomonaco@unipi.it
vincenzolomonaco.com
University of Pisa

THANKS