Continual Learning for *Online Behavioral Analytics*

Yasas Senarath · Marcos Zampieri · Hemant Purohit

Information Sciences and Technology (IST) Department
George Mason University, Fairfax, Virginia

Tutorial at The 20th International AAAI Conference on Web and Social Media (ICWSM) 2026

Last updated: May 2026

Acknowledgement

This tutorial builds upon numerous excellent research papers and prior tutorials in continual learning. All sources are credited on their respective slides.

Outline

I. Introduction and Background
II. Experience-Replay-based Methods
III. Regularization-based Methods
IV. Beyond Traditional Settings
V. Open Challenges and Future Directions

Introduction and Background

Section I

Behavioral Analytics

Buy or sell a product

Issue an emergency alert

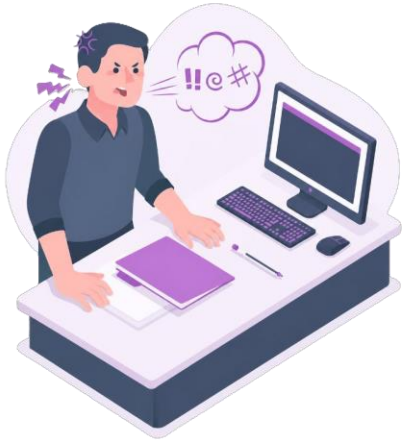
Suggest a mental health intervention

Thanks: Freepik for images

AI in Behavioral Analytics

Sentiment Analysis

Stance Detection



Hate Speech Detection

Anomaly Detection

Emergency Incident Detection

Thanks: Freepik for images

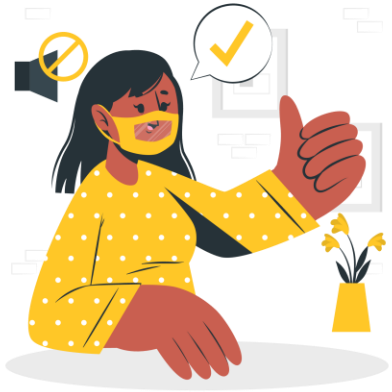
Where Do We Need Continual Learning?

- We live in a dynamic and ever changing world

Time Drift	**Domain Drift**	**Language Drift**
facts change (news, science, policies)	enterprise or specialized sectors evolve	new slang and multilingual corpora appear

Wu, T., Luo, L., Li, Y. F., Pan, S., Vu, T. T., & Haffari, G. (2024). Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

Example: COVID-19 Stance Detection



2020 · PHASE **A**

Lockdowns & masks

Targets: lockdown policy, mask mandates, travel bans, school closures

|*"shutting it all down hurts more than it helps" - against*

2021 · PHASE **B**

Vaccines & mandates

Targets: vaccine efficacy, mandates, passports, employer rules

|*"got my booster today — small price for everyone's safety" - favor*

2022+ · PHASE **C**

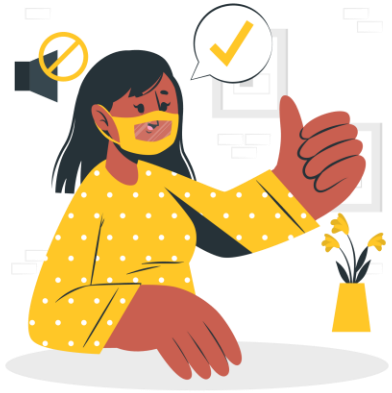
Variants & long COVID

Targets: new variants, booster fatigue, long-COVID policy, masks return

|*"another booster? I am out - variants will keep coming anyway" - against*

Thanks: Freepik for images

Traditional Stance Detection



2020 · PHASE **A**

Lockdowns & masks

2021 · PHASE **B**

Vaccines & mandates

2022+ · PHASE **C**

Variants & long COVID

Model 1

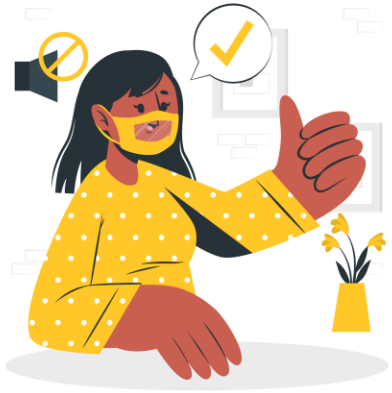
Model 2

Model 3

*We will compare different other existing paradigms later in this session

Thanks: Freepik for images

Continual Stance Detection



2020 · PHASE **A**

Lockdowns & masks

Model 1

2021 · PHASE **B**

Vaccines & mandates

Model 2

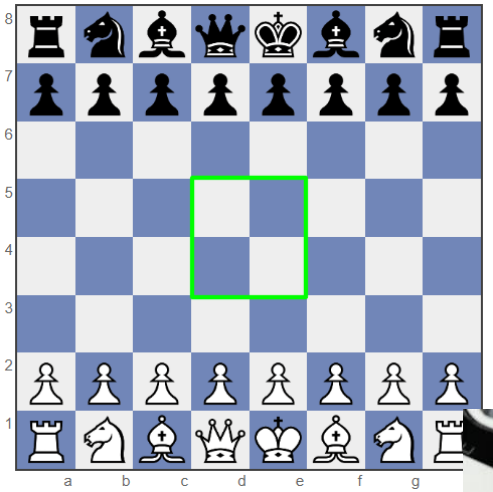
2022+ · PHASE **C**

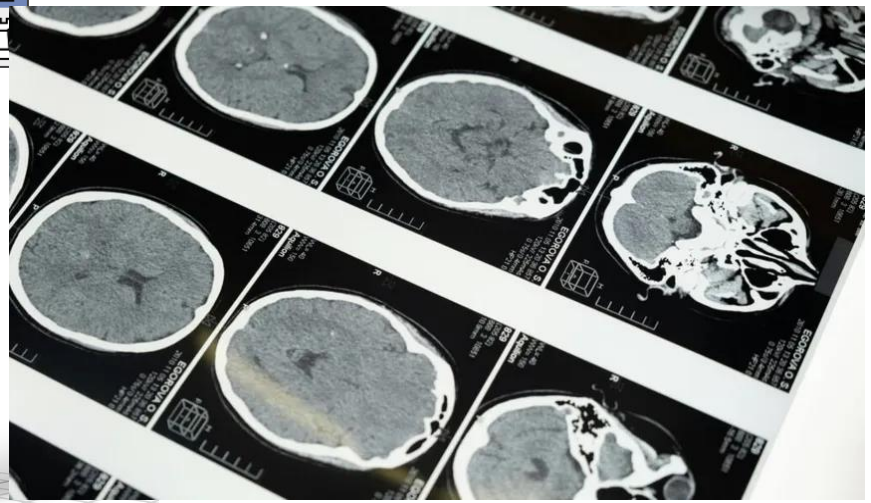
Variants & long COVID

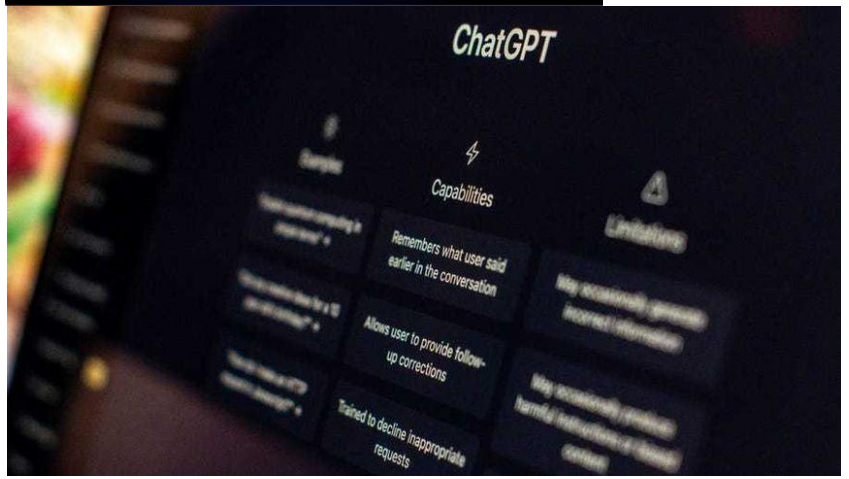
Model 3

Thanks: Freepik for images

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







Lomonaco, V., & Rish, I. (2021). *Continual learning with deep architectures* [Tutorial]. International Conference on Machine Learning (ICML 2021). https://icml.cc/virtual/2021/tutorial/10833

Human-Level AI: “Broad” - Versatile, Multi-Task

One of the most important feature in human learning is to learn and *adapt new knowledge continuously* without forgetting previous knowledge.

Traditional Deep Learning	Continual Learning
NO	**YES**

Fit in any real-world applications facing a continuous stream of non-stationary data when it is a bad idea to retrain from scratch.

Lomonaco, V., & Rish, I. (2021). *Continual learning with deep architectures* [Tutorial]. International Conference on Machine Learning (ICML 2021). https://icml.cc/virtual/2021/tutorial/10833

Definition of Continual Learning

> Continual learning is a machine learning paradigm where an algorithm receives the data from tasks sequentially without the access to previous ones to learn a model that performs the best for all tasks.

Sequential tasks	Test for all tasks	No access to previous task's data	Infinite sequence of tasks
• Non-stationary data: tasks from different distributions (difficult!)	• Difficult	• Memory cost / potential violence to privacy • Not joint training. Problems: computation cost, induction bias	• Never know the future challenges

Our Focus

- Continual Learning in Deep Neural Network
- Examples from Behavioral Analytics
- Supervised Continual Learning
- Pre-trained & Large Language Models

Challenge: Catastrophic Forgetting

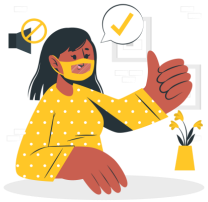
> "…the process of learning a new set of patterns suddenly and completely erased a network's knowledge of what it had already learned"
>
> (French, 1999)

- CF was identified by (McCloskey and Cohen, 1989)
- Why?
  - Distributed Representations – Shared Weights

[1] 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.

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

Challenge: Catastrophic Forgetting



2020 · PHASE A

Lockdowns & masks

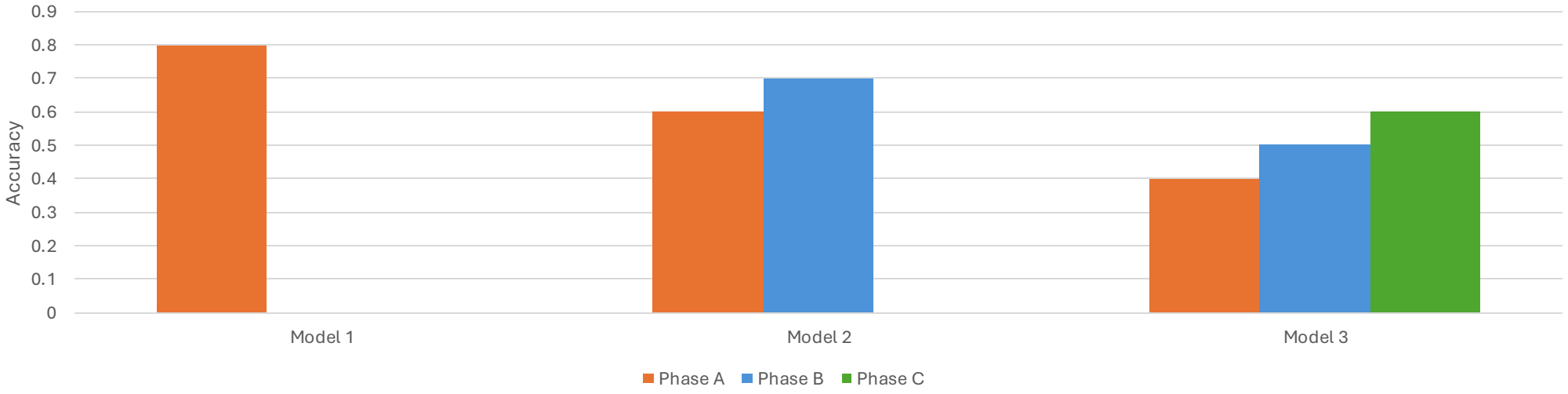
2021 · PHASE B

Vaccines & mandates

2022+ · PHASE C

Variants & long COVID

Model 1 → Model 2 → Model 3



More Generally: Stability vs Plasticity

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.

[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.

More Generally: Stability vs Plasticity

STABILITY ↑

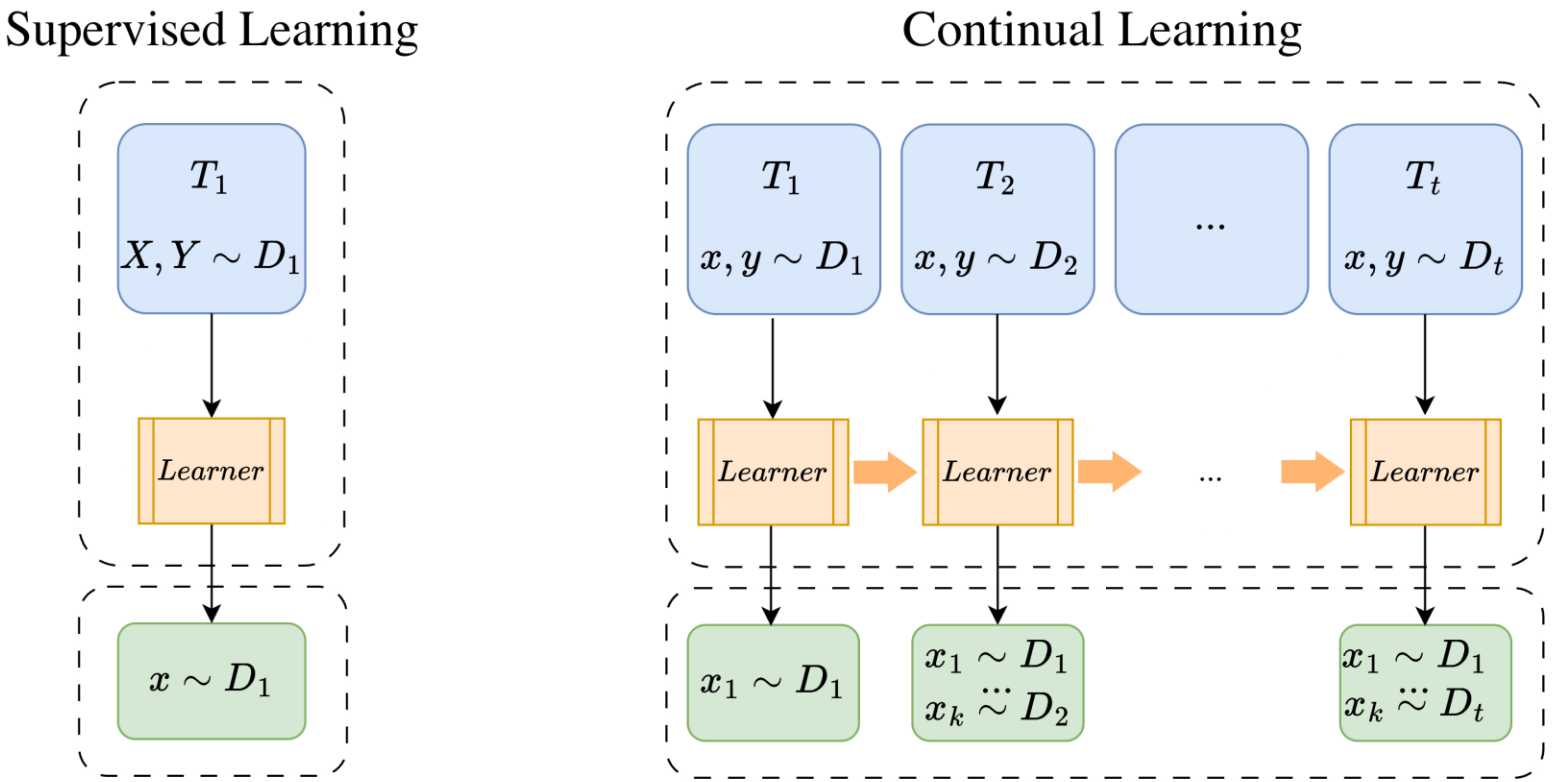
Protect what is already learned.
Risk: a frozen model that ages out, every new phase erodes its relevance

PLASTICITY ↑

Absorb the new distribution fast.
Risk: overfit to last week and forget last year — catastrophic forgetting

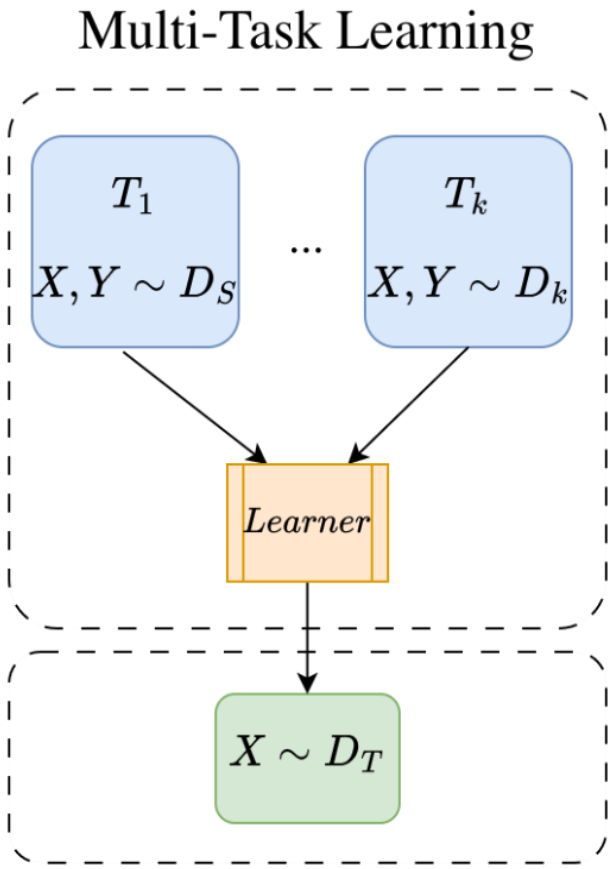
[1] Grossberg. S. (1982) Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.
[2] Carpenter, G. and Grossberg, S. (1987) ART 2: Self-organization of stable category recognition codes for analog input patterns.

Supervised Continual Learning



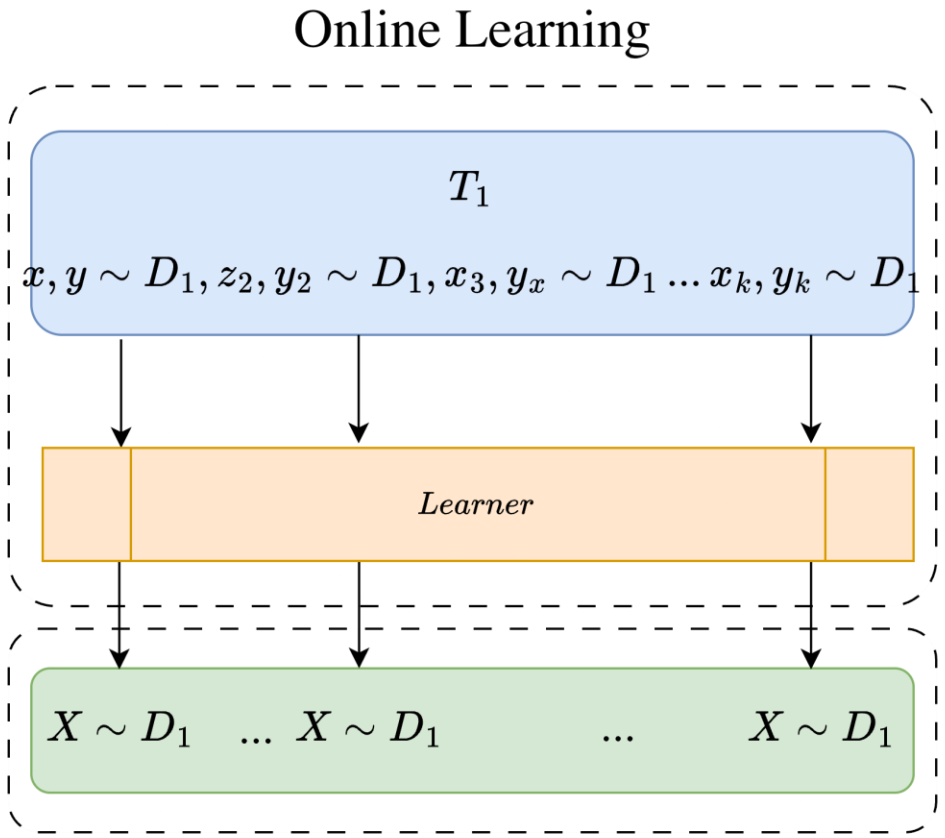
Continual vs Multi-Task Learning

- Learning of multiple related tasks offline, simultaneously
- Using a set or subset of shared parameters
- No continual model adaptation



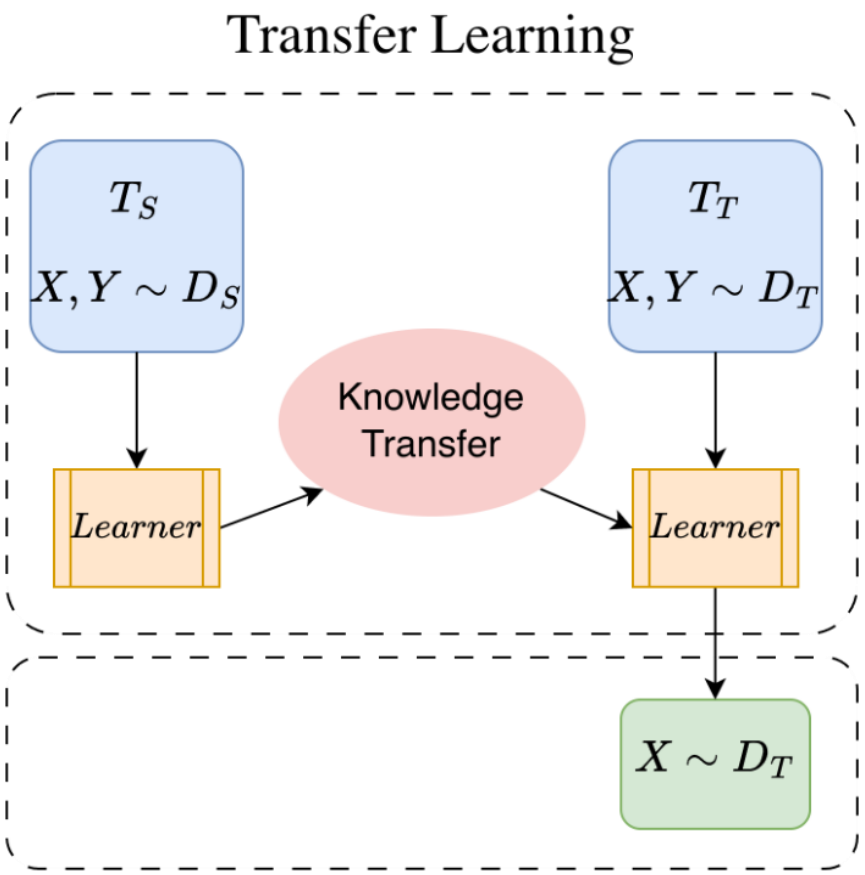
Continual vs Online Learning

- Same / similar task
- Observe data in sequence
- Main goal is to adapt to new distributions of the same problem



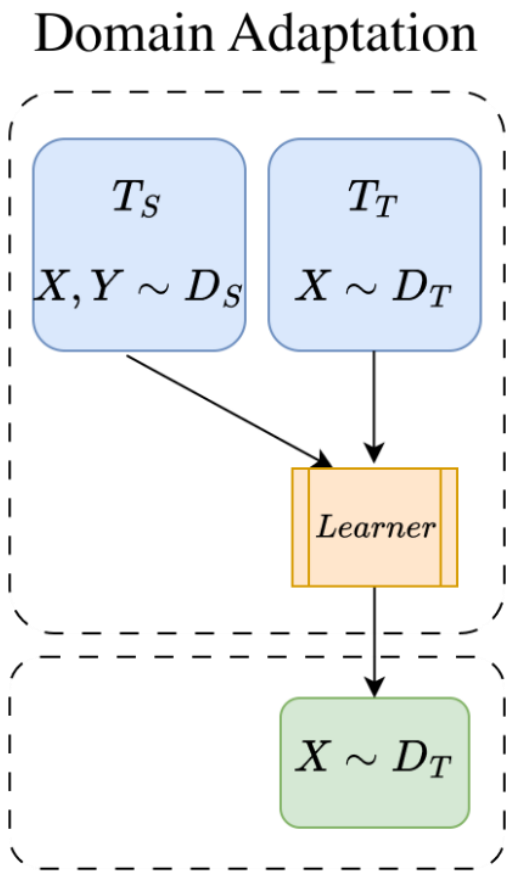
Continual vs Transfer Learning

- 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



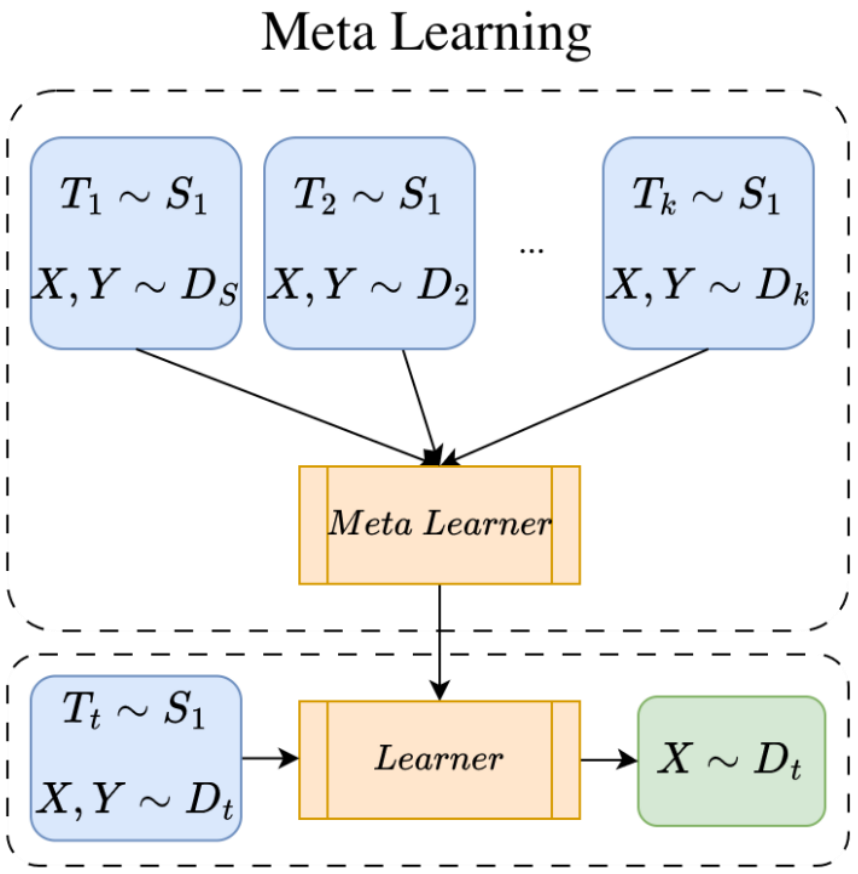
Continual Learning vs Domain Adaptation

- 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



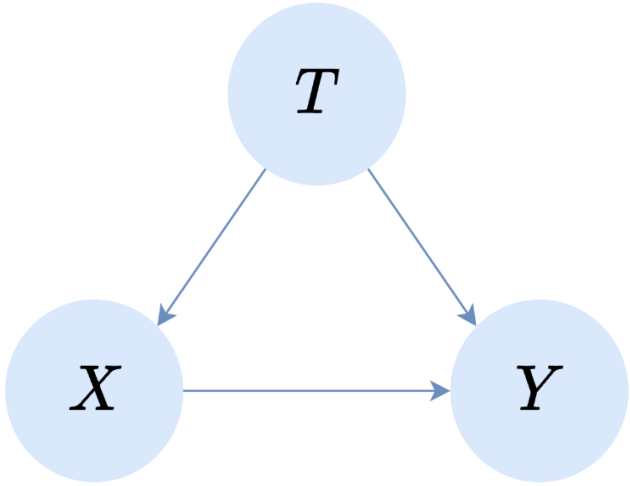
Continual vs Meta-Learning

- Faster adaptation on a task given a large number of training tasks
- Offline training: a set of training tasks available at the same time



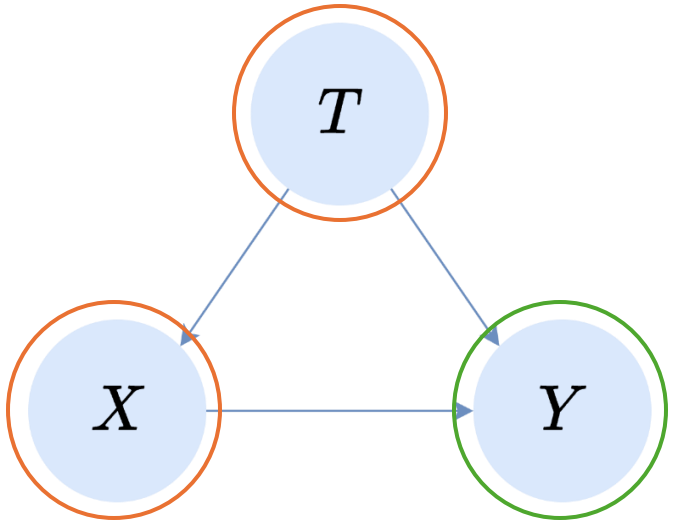
Continual Learning Settings

- X – Input Vector / Text
- Y – class label
- T – task (context) defines P(X, Y| T)

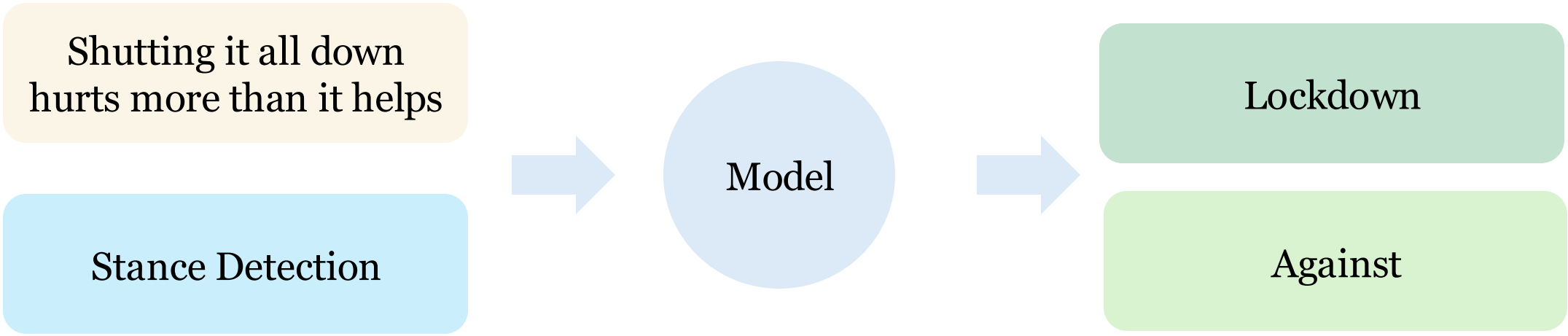


Task-Incremental CL

- 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

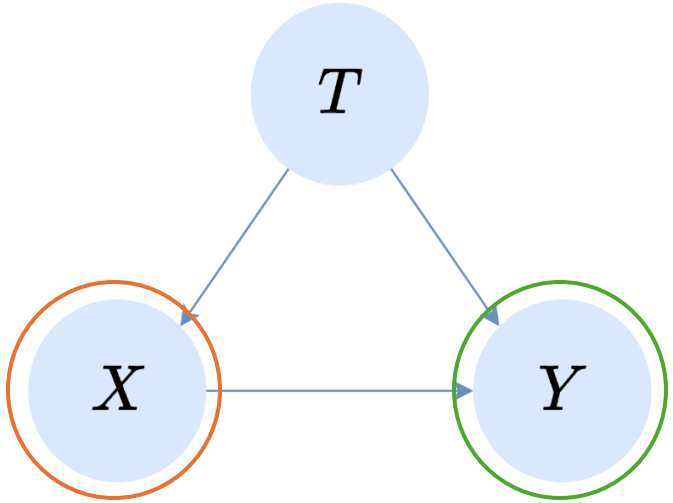


Task-Incremental CL- Example

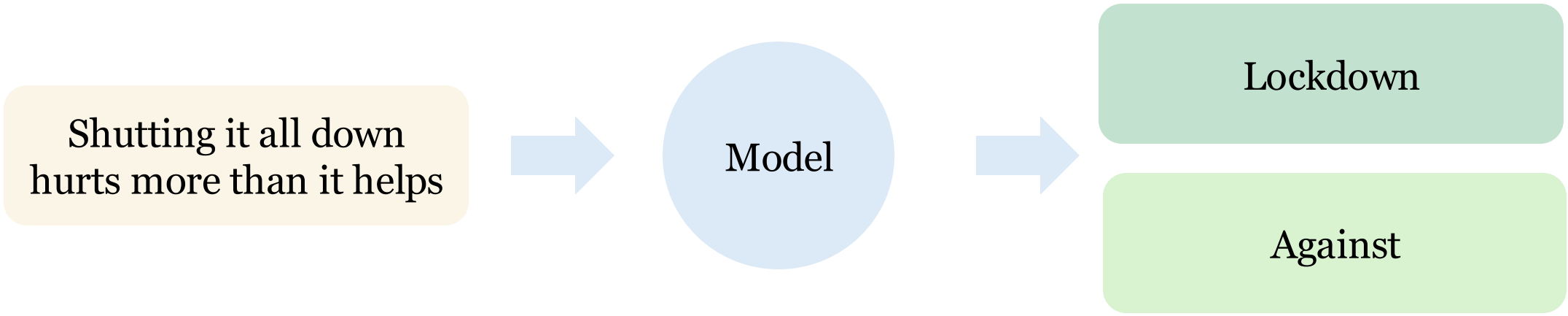


Class-Incremental CL

- 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

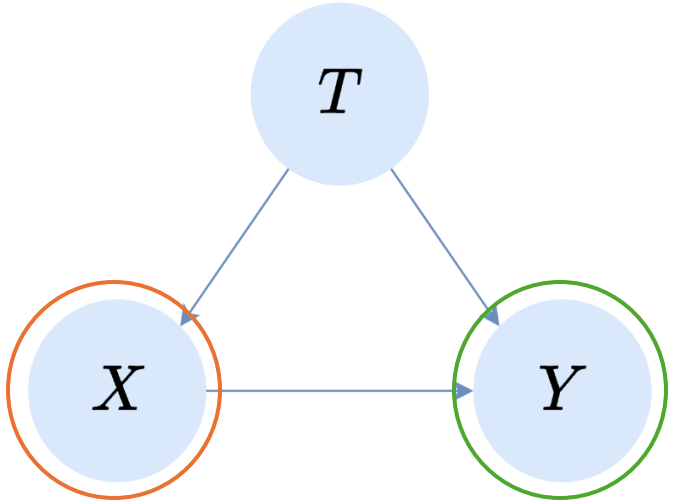


Class-Incremental CL- Example

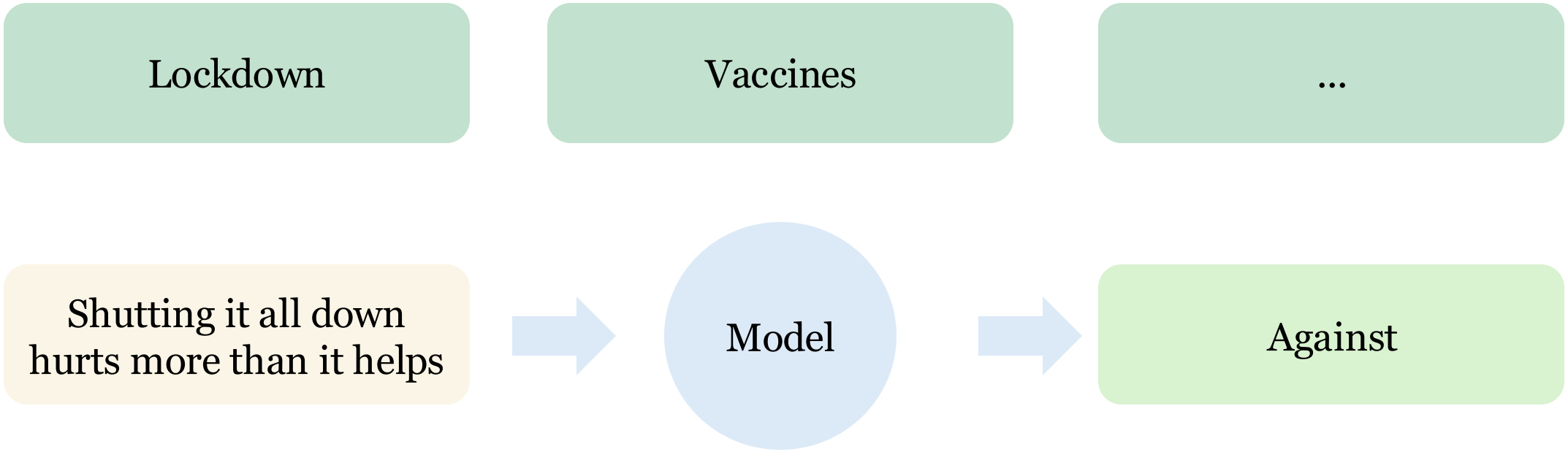


Domain-Incremental CL

- 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



Domain-Incremental CL - Example



Task-agnostic CL

- Receives a stream of data continuously
- Has no explicit task boundaries provided
- Has no task ID at training or test time
- Must autonomously discover structure in the data stream

Formal Definition of CL Classification

- In continual learning classification problem, we have:
  - Tasks: $t = 1, 2, \cdots$
  - Training data of tasks: $\mathcal{D}_{\text{train}}^{(t)} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_t} \in (\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})$
  - Testing data of tasks as well: $\mathcal{D}_{\text{test}}^{(t)} \in (\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})$
- We aim to develop an algorithm which trains the model $f^{(t-1)}$ to $f^{(t)}$ at the time for task t:
  - With access to $\mathcal{D}_{\text{train}}^{(t)}$ only
  - To perform well on all seen tasks $\mathcal{D}_{\text{test}}^{(1)}, \cdots, \mathcal{D}_{\text{test}}^{(t)}$

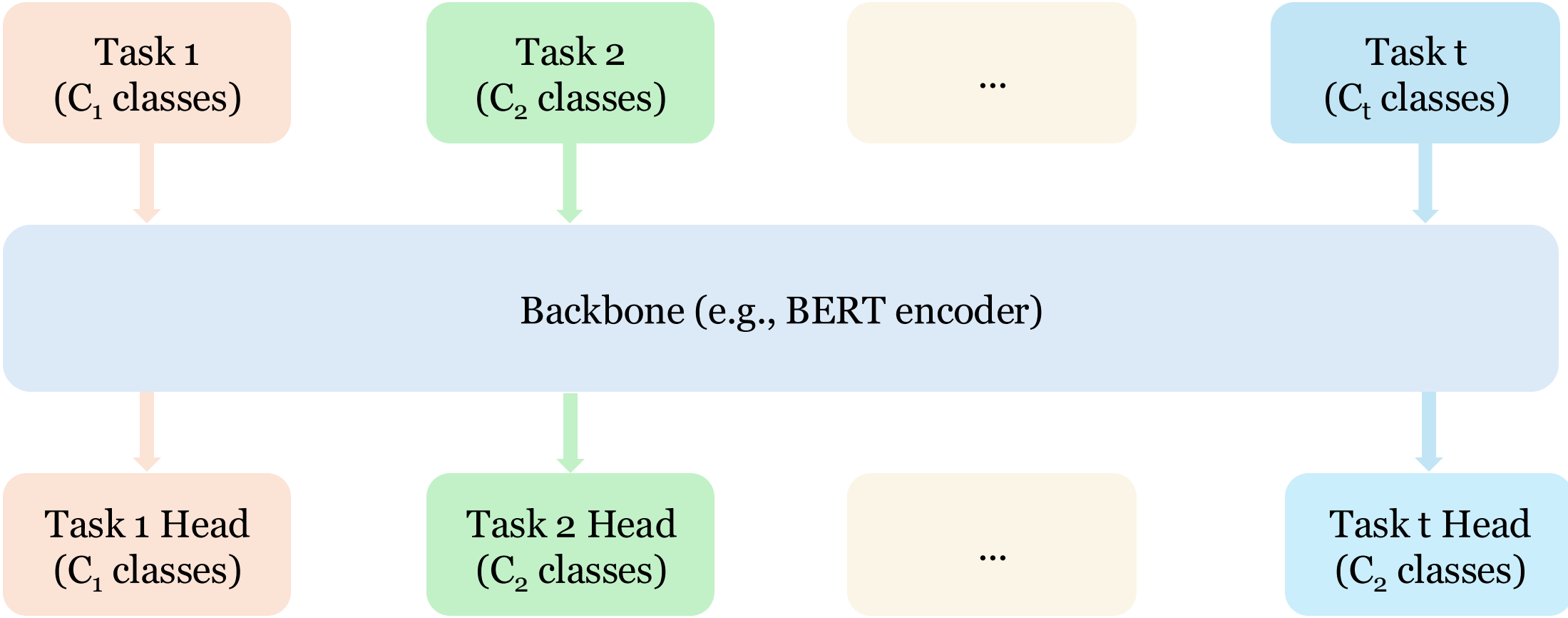
https://pengxiang-wang.com/slides/slides-continual-learning-beginners-guide.pdf

Neural Network: The Multi-head Classifier

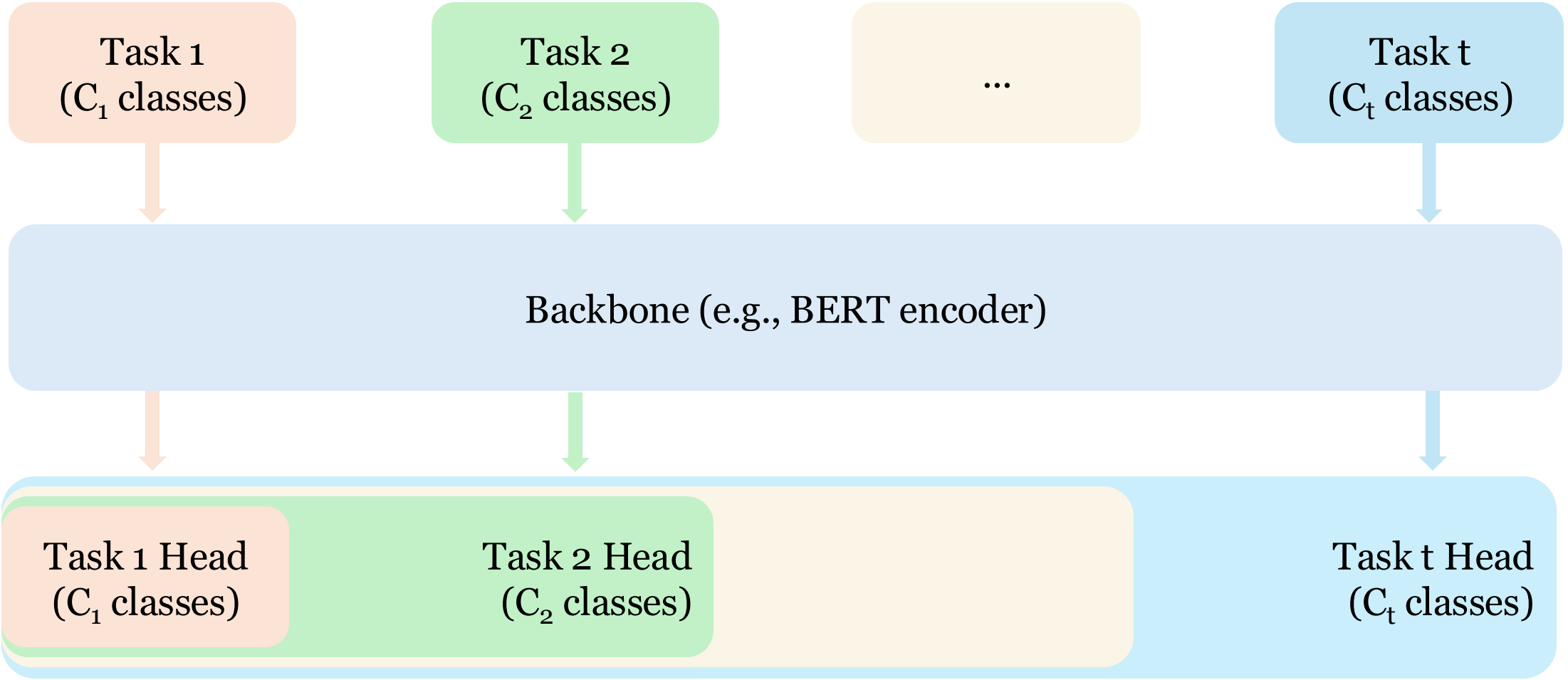
- CL Model (f(t)) = Backbone Network + Multi-head Classifier
- Backbone Network
  - Encodes the input
- Multi-head classifier
  - Output heads assigned to different tasks
  - A head = simply a linear output layer outputting logits of classes
  - New head is initialized and trained along with backbone as new task come in - For Class-IL and Task-IL problem

https://pengxiang-wang.com/slides/slides-continual-learning-beginners-guide.pdf

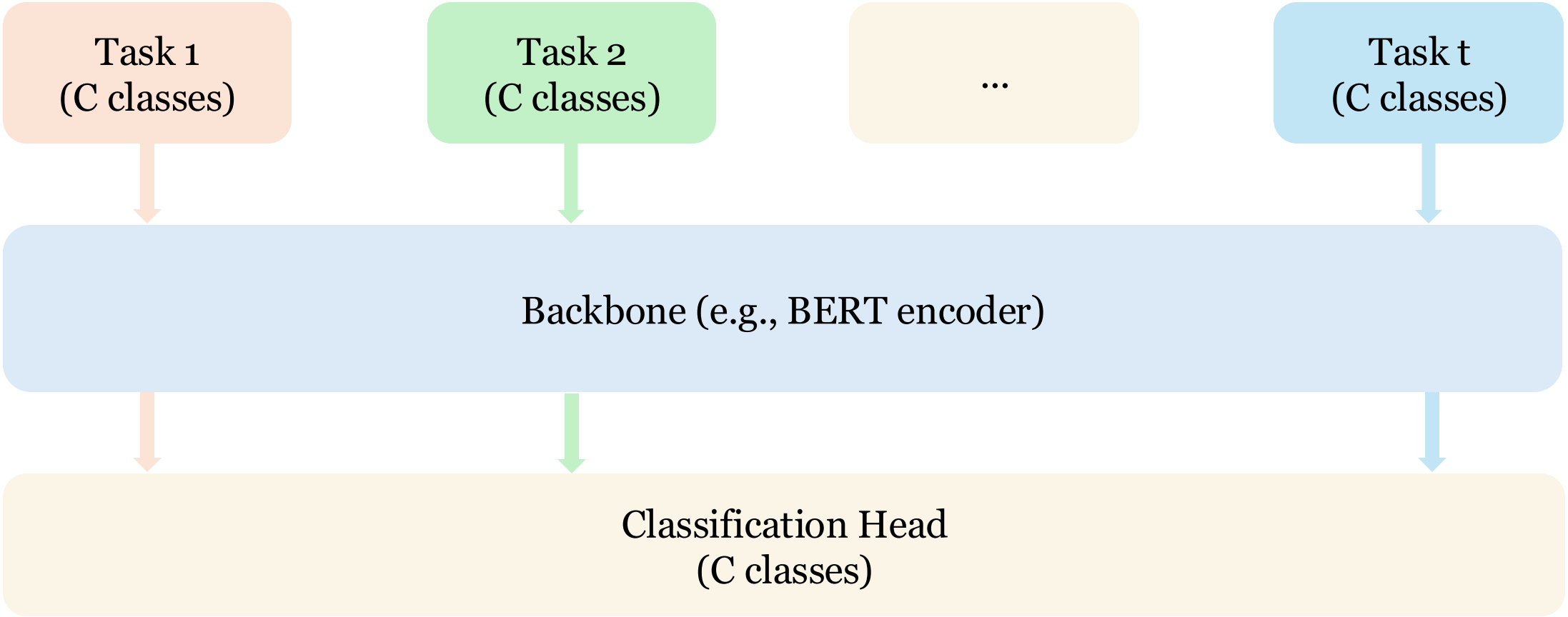
The Multi-head Classifier (Task-IL)



The Multi-head Classifier (Class-IL)



The Multi-head Classifier (Domain-IL)



Metrics: What CL Cares About

- After sequentially learning all T tasks

$$\mathbf{R} = \begin{pmatrix} \mathcal{A}_{1,1} & & & & \\ \mathcal{A}_{2,1} & \mathcal{A}_{2,2} & & & \\ \mathcal{A}_{3,1} & \mathcal{A}_{3,2} & \mathcal{A}_{3,3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathcal{A}_{T,1} & \mathcal{A}_{T,2} & \mathcal{A}_{T,3} & \cdots & \mathcal{A}_{T,T} \end{pmatrix}$$

- Average Accuracy: $\mathrm{AA} = \frac{1}{T} \sum_{k=1}^{T} \mathcal{A}_{T,k}$

Metrics: What CL Cares About

- After sequentially learning all T tasks

$$\mathbf{R} = \begin{pmatrix} \mathcal{A}_{1,1} & & & & \\ \mathcal{A}_{2,1} & \mathcal{A}_{2,2} & & & \\ \mathcal{A}_{3,1} & \mathcal{A}_{3,2} & \mathcal{A}_{3,3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathcal{A}_{T,1} & \mathcal{A}_{T,2} & \mathcal{A}_{T,3} & \cdots & \mathcal{A}_{T,T} \end{pmatrix}$$

- Forgetting Metric:

$$\mathcal{F} = \frac{1}{T-1} \sum_{k=1}^{T-1} f_k^T$$

$$f_k^T = \max_{t \in \{k,\ldots,T-1\}} \mathcal{A}_{t,k} - \mathcal{A}_{T,k}$$

Break

10 min break

Continual Learning Methods

Replay Methods

Stores exemplars / use generative model to replay prior tasks

Regularization-Based Methods

Adds regularization term to loss function instead of storing data

Parameter Isolation Method

Separate model parameters per task

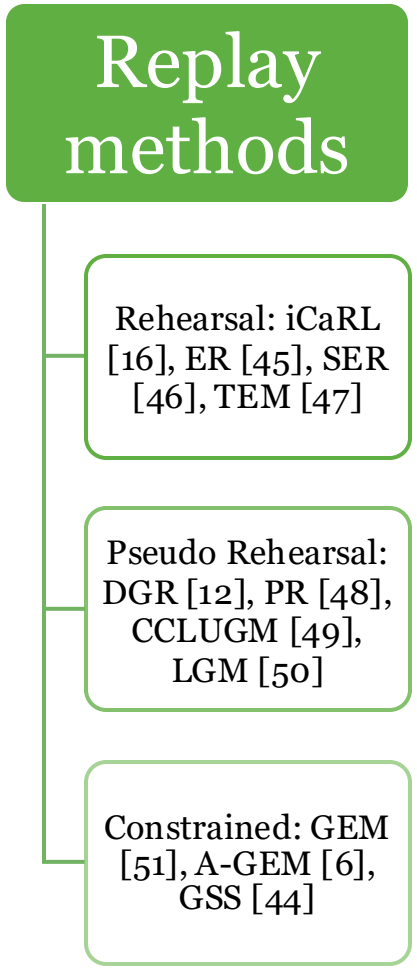
Demo I – The Problem

https://github.com/human-info-lab/ICWSM-2026-Continual-Learning-Tutorial-Code

Experience-Replay-based Methods

Section II

Replay-based Methods



- 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).

The Replay Idea

Keep a small *buffer*. Mix it back in.

PAST DATA

Replay Buffer $|M| = k$

A bounded memory of representative exemplars.

JOINT TRAINING

Model Adaptation

Buffer + new batch, weighted & regularized.

OUTPUT

Updated Model θ_{t+1}

Competent on past and present.

Q1 What enters the buffer? → Exemplar Selection

Q2 How is the buffer used at training time? → Model Adaptation

Data Replay - Simplified

- Replay is just two losses:

$$L_{replay} = L_{task}(\theta; D_t) + \lambda \cdot L_{task}(\theta; \mathrm{M})$$

- Large λ → Optimizer attends to past examples
  → preserved old performance, slower adaptation
- Small λ → optimizer focuses on current batch
  → faster adaptation, more forgetting risk

Exemplar Selection

1. Random task-aware sampling
2. Reservoir sampling (Vitter, 1985)
3. Class-balanced reservoir (CBRS) (Chrysakis & Moens, 2020)
4. Clustering-based selection (Rebuffi et al., 2017)
5. Influence-based selection (Aljundi et al., 2019)

Random task-aware Sampling

- At the end of task t, reserve $t/(t+1) \cdot |M|$ slots for prior tasks and fill the rest with a random sample of task k
- No assumptions about example importance
- Strong baseline whenever your stream has rough class & topic balance

Reservoir Sampling

(Vitter, 1985)

- After n examples, every example has equal probability k/n of being in the buffer
- Use When: Task boundaries are blurry

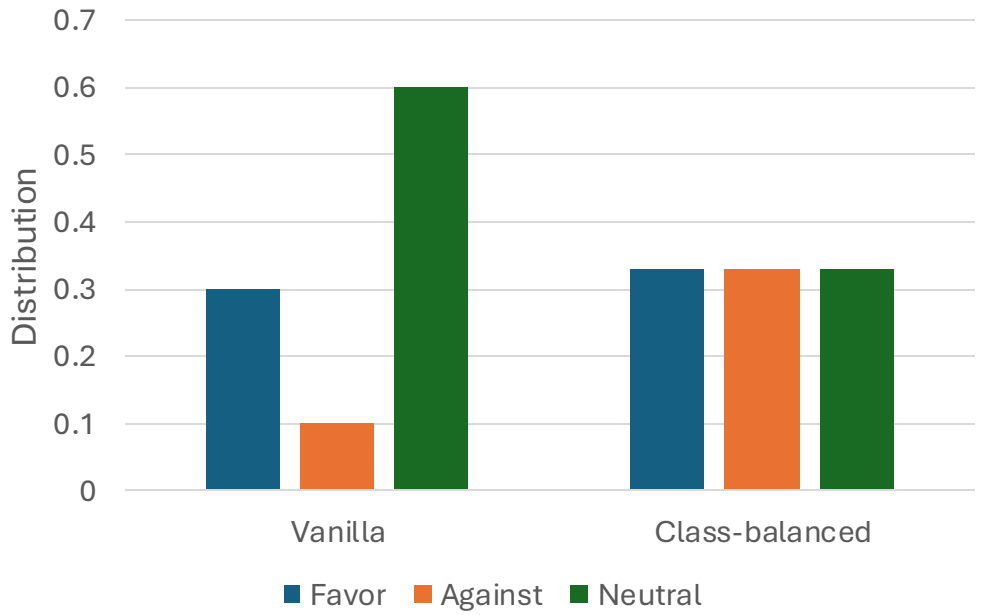
```
fun res_update(M, x, n, k):
  # M: buffer of size k
  # x: new instance, n: index
  if len(M) < k:
    M.append(x)
  else:
    j = random_int(0, n)
    if j < k:
      M[j] = x
  return M
```

Vitter, J. S. (1985). Random sampling with a reservoir. ACM Transactions on Mathematical Software (TOMS), 11(1), 37-57.

Class-balanced Reservoir Sampling

(Chrysakis & Moens, 2020)

- Partition the buffer per class. Run reservoir within each partition
- Use When: Data is imbalanced

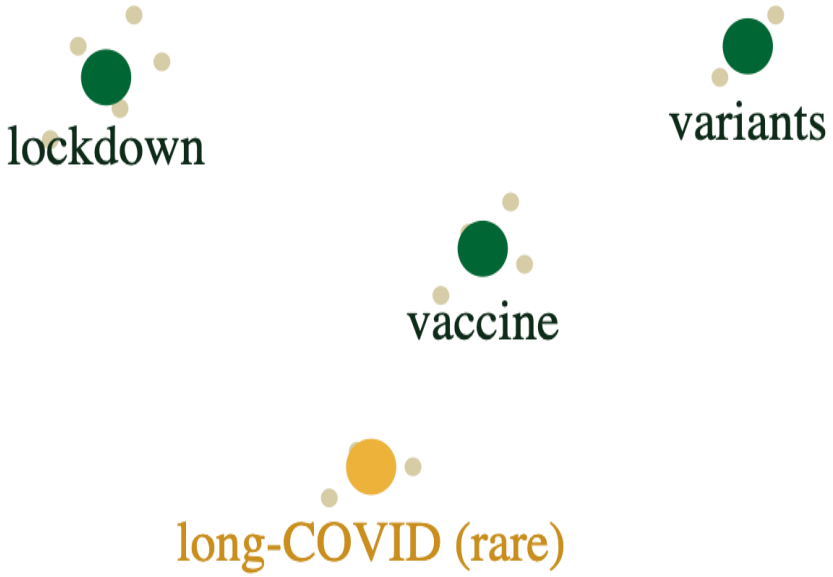


Chrysakis, A., & Moens, M. F. (2020, November). Online continual learning from imbalanced data. In International Conference on Machine Learning (pp. 1952-1961). PMLR.

Clustering-based Selection

(Rebuffi, 2017)

- Embed past examples with the current model
- K-means in feature space
- Keep the example nearest each centroid (a.k.a. herding)



Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 2001-2010).

Influence-based Selection

(Aljundi et al., 2019)

- Select instances with maximum *diversity* of samples in the replay buffer with *parameters gradient* as the feature
- Computationally expensive
- Useful in combination with specific model adaptation method
  - e.g., data regularization*

* Will be discussed in later slides

Aljundi, R., Lin, M., Goujaud, B., & Bengio, Y. (2019). Gradient based sample selection for online continual learning. Advances in neural information processing systems, 32.

Replay-based Model Adaptation

1. Concatenation
2. Class-balanced sampling
3. Data regularization
4. Generative Replay

Concatenation

- Train on the *union* of *buffer* and *new batch*

$$L_{total} = L_{CE}(\theta; D_k) + \lambda L_{CE}(\theta; M)$$

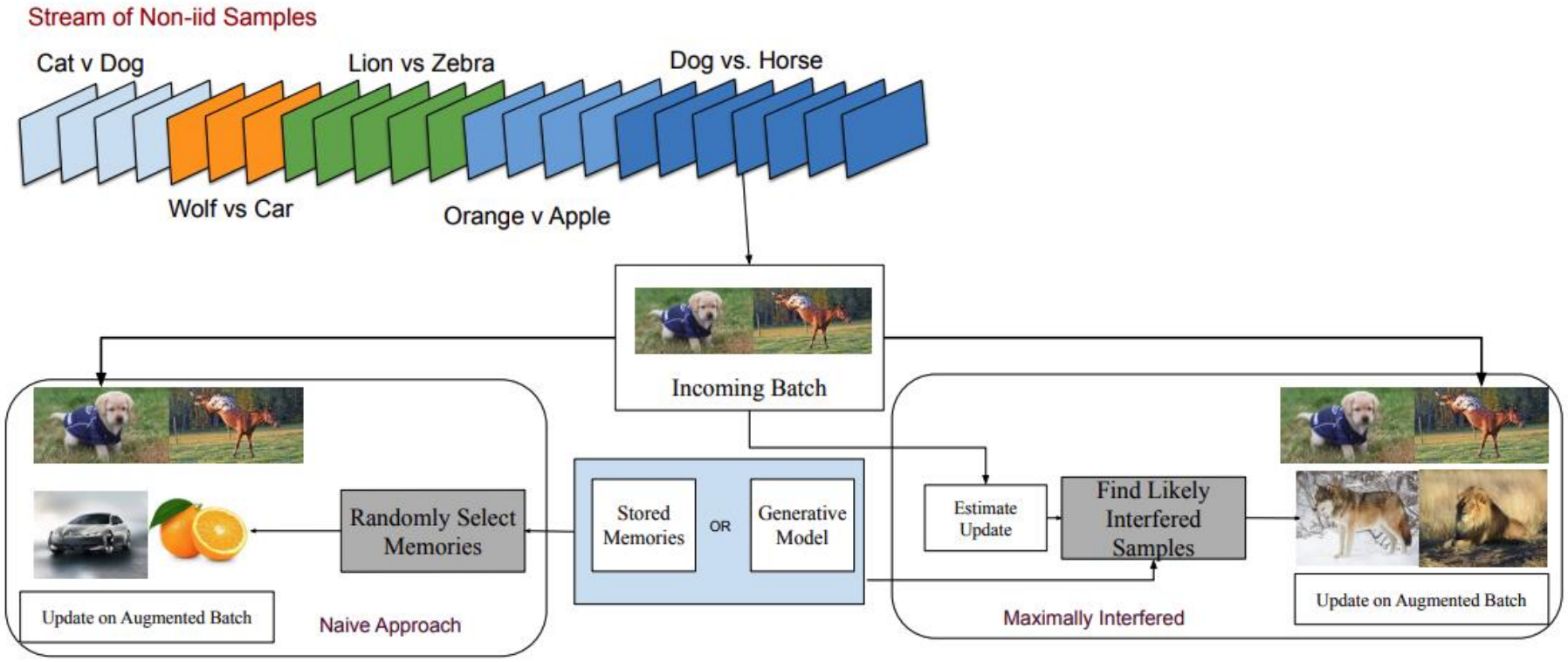
PROS

- Zero extra hyperparameters beyond λ
- Trivial to implement
- Surprisingly strong on balanced streams

CONS

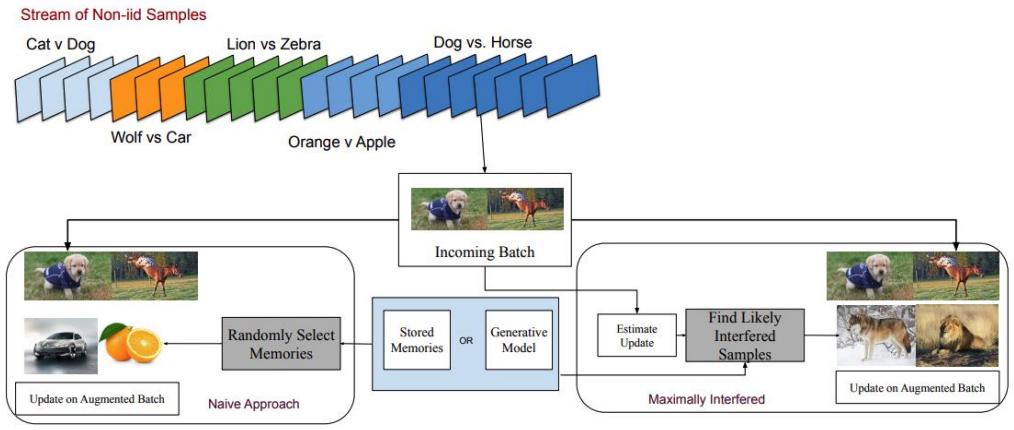
- Dominant class drowns out the minority

Maximally Interfered Sampling



Aljundi, R., Belilovsky, E., Tuytelaars, T., Charlin, L., Caccia, M., Lin, M., & Page-Caccia, L. (2019). Online continual learning with maximal interfered retrieval. Advances in neural information processing systems, 32.

Maximally Interfered Sampling



Algorithm 1: Experience MIR (ER-MIR)

Input: Learning rate α, Subset size C; Budget $\mathcal{B}$

Initialize: Memory $\mathcal{M}$; θ

for $t \in 1..T$ **do**

  for $B_n \sim D_t$ **do**

    %%Virtual Update

    $\theta^v \leftarrow \texttt{SGD}(B_n, \alpha)$

    %Select C samples

    $B_C \sim \mathcal{M}$

    %Select based on score

    $S \leftarrow sort(s_{MI}(B_C))$

    $B_{\mathcal{M}_C} \leftarrow \{S_i\}_{i=1}^{\mathcal{B}}$

    $\theta \leftarrow SGD(B_n \cup B_{\mathcal{M}_C}, \alpha)$

    %Add samples to memory

    $\mathcal{M} \leftarrow UpdateMemory(B_n)$;

  end

end

Aljundi, R., Belilovsky, E., Tuytelaars, T., Charlin, L., Caccia, M., Lin, M., & Page-Caccia, L. (2019). Online continual learning with maximal interfered retrieval. Advances in neural information processing systems, 32.

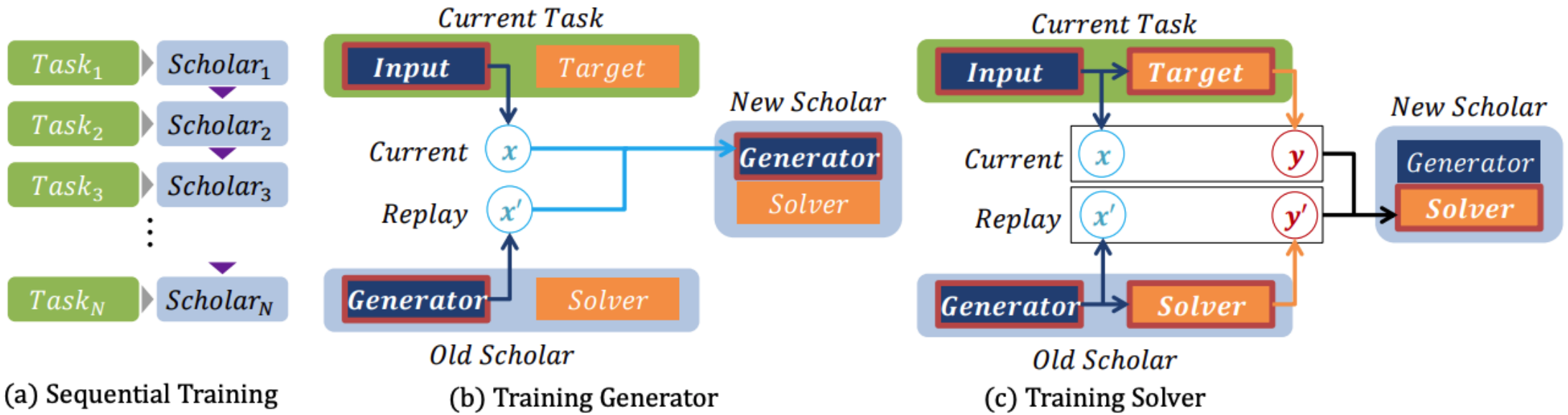
Data Regularization

- Gradient Episodic Memory (GEM), Averaged GEM (A-GEM)
- From Rehearsal to Constraint

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

$$\langle g, g_k \rangle := \left\langle \underbrace{\frac{\partial \ell(f_\theta(x,t),y)}{\partial \theta}}_{\text{Gradient on the new task}}, \underbrace{\frac{\partial \ell(f_\theta,\mathcal{M}_k)}{\partial \theta}}_{\text{Gradient on the buffer instances}} \right\rangle \geq 0, \text{ for all } k<t.$$

Generative Replay



Shin, H., Lee, J. K., Kim, J., & Kim, J. (2017). Continual learning with deep generative replay. Advances in neural information processing systems, 30.

Regularization-based Methods

Section III

Regularization-based Methods

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

Parameter regularization

1. Identify which weights mattered for past tasks
2. Penalize movement in those directions

Elastic Weight Consolidation (EWC)

Feature (function) regularization

1. Identify features from the past tasks
2. Penalize change on those features

Learning without Forgetting (LwF)

Regularization-based Methods

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

Parameter regularization

1. Identify which weights mattered for past tasks
2. Penalize movement in those directions

Elastic Weight Consolidation (EWC)

Feature (function) regularization

1. Identify features from the past tasks
2. Penalize change on those features

Learning without Forgetting (LwF)

Elastic Weight Consolidation (EWC)

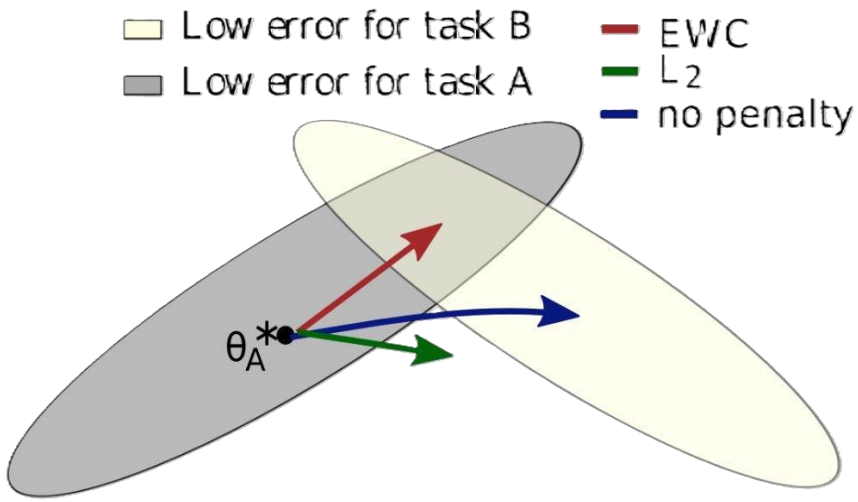
- Do not drift far from the parameters that mattered

Parameter Importance

Change in Parameters

$$L_{EWC}(\theta) = L_B(\theta; D_k) + \Sigma_i \, \lambda/2 \cdot F_i(\theta_i - \theta^*_{A,i})^2$$

Regularization Loss



Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcomingcatastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13), 3521-3526.

Elastic Weight Consolidation (EWC)

- How to calculate the parameter importance?
  - Fisher importance matrix

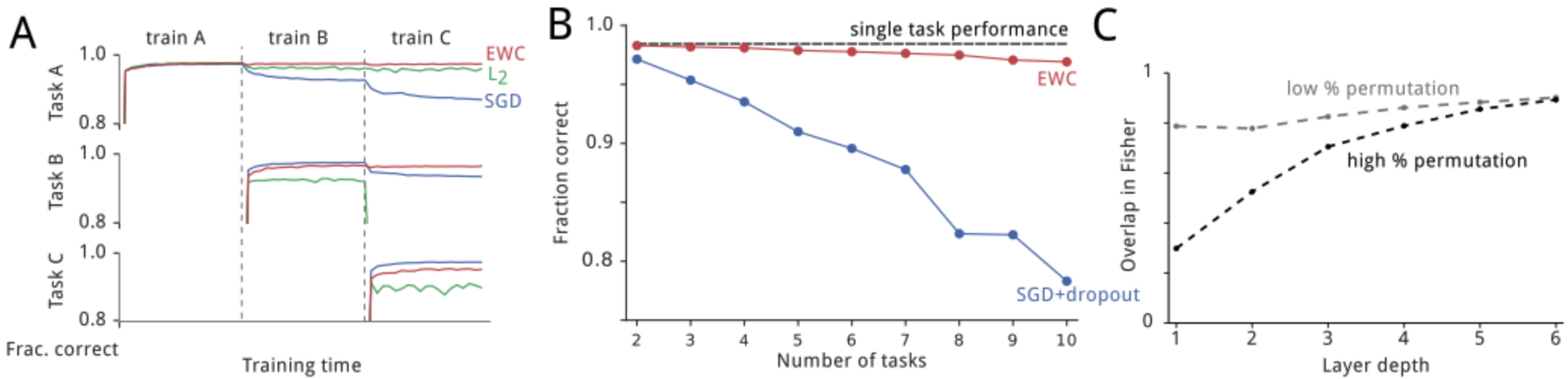
$$F_i = \mathbb{E}_{x \sim \mathcal{D}} \left[\left(\partial \log p(y \mid x, \theta) / \partial \theta_i \right)^2 \right]$$

High F_i =
Small perturbations of θ cause large changes in predictions on D

Low F_i =
Small perturbations of θ cause small changes in predictions on D

EWC – Results on MNIST task

- Performance on MNIST task as presented in the original paper

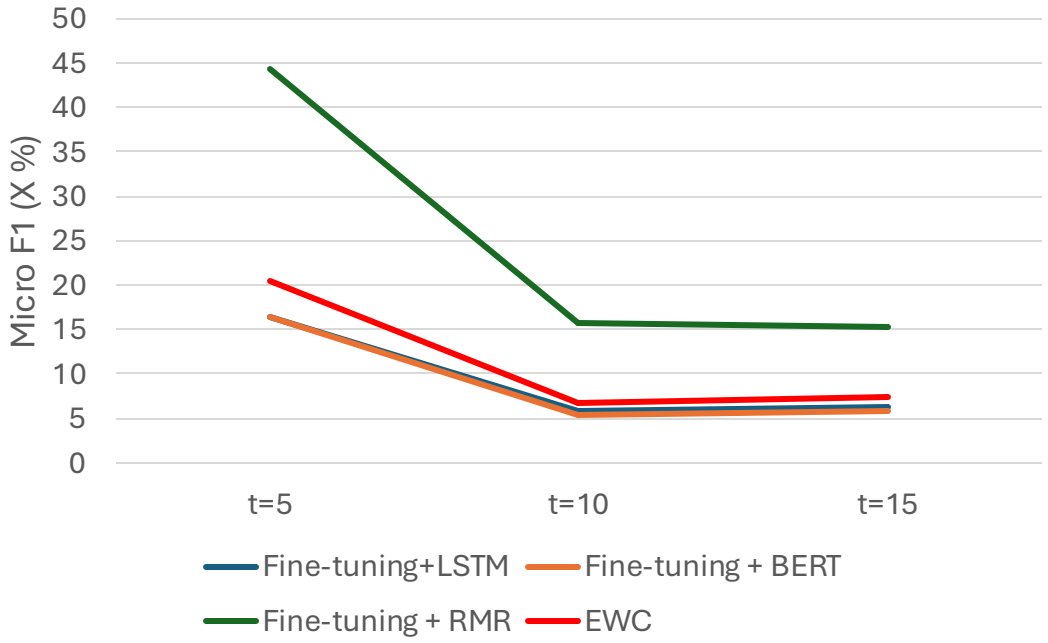


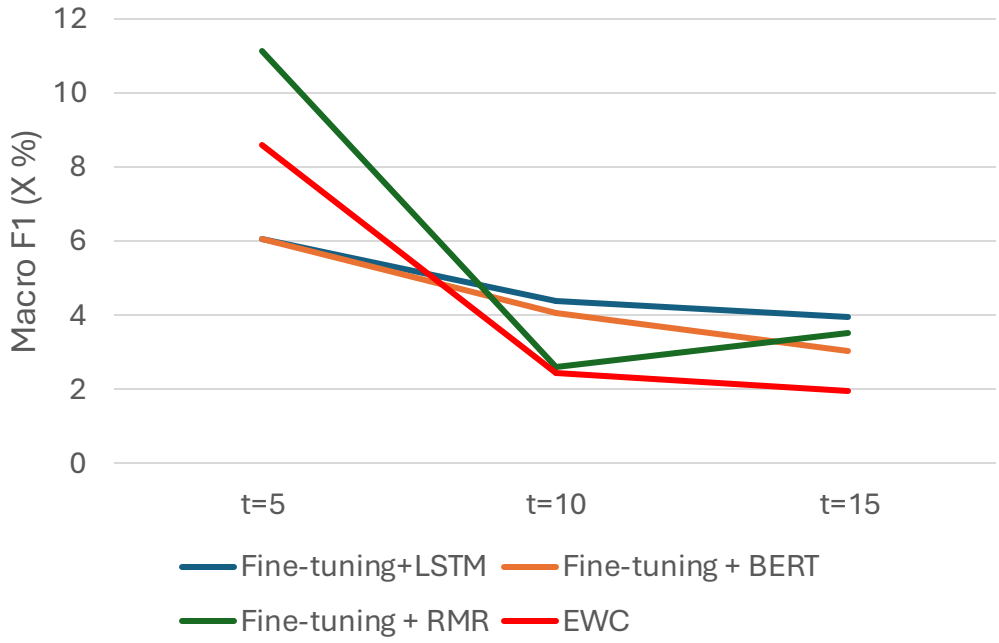
Results of the permuted MNIST task.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcomingcatastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13), 3521-3526.

EWC – Results on Behavioral Tasks

- Performance on Hate Speech Detection task (Qian et al., 2021)





Qian, J., Wang, H., El Sherief, M., & Yan, X. (2021, June). Lifelong learning of hate speech classification on social media. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 2304-2314).

EWC – Results on Behavioral Tasks

- EWC preserves previous tasks by slowing parameter updates but works best when tasks are similar
- Since our task sequence involves frequent shifts in vocabulary, topic, and data distribution, memory replay is a more effective approach

Qian, J., Wang, H., ElSherief, M., & Yan, X. (2021, June). Lifelong learning of hate speech classification on social media. In Proceedings of the 2021Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 2304-2314).

Regularization-based Methods

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

Parameter regularization

1. Identify which weights mattered for past tasks
2. Penalize movement in those directions

Elastic Weight Consolidation (EWC)

Feature (function) regularization

1. Identify features from the past tasks
2. Penalize change on those features

Learning without Forgetting (LwF)

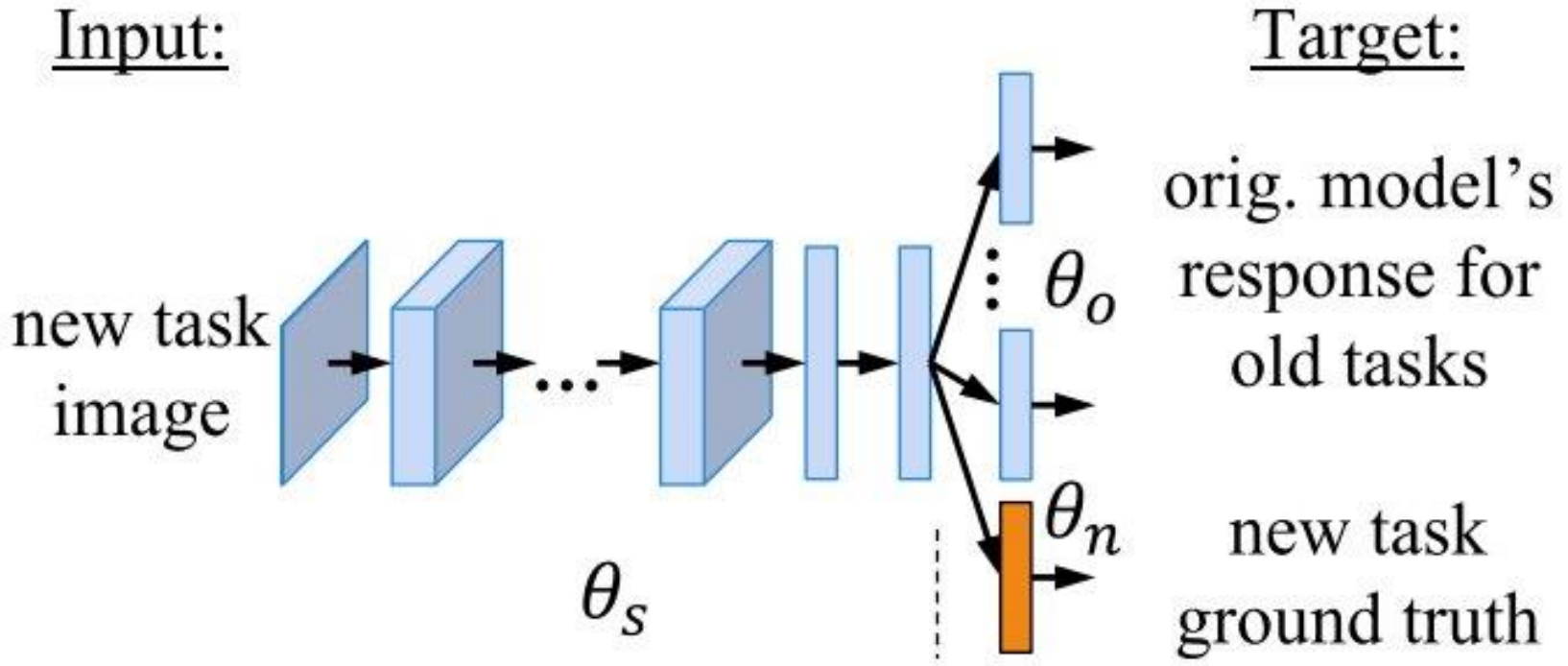
Learning without Forgetting (LwF)

- Different parameter configurations can produce similar features
- Move the constraint to where the meaning lives
  - Preserve feature/output behavior, not weight
- Solution: Distillation from old to new task

Li, Z., & Hoiem, D. (2017). Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12), 2935-2947.

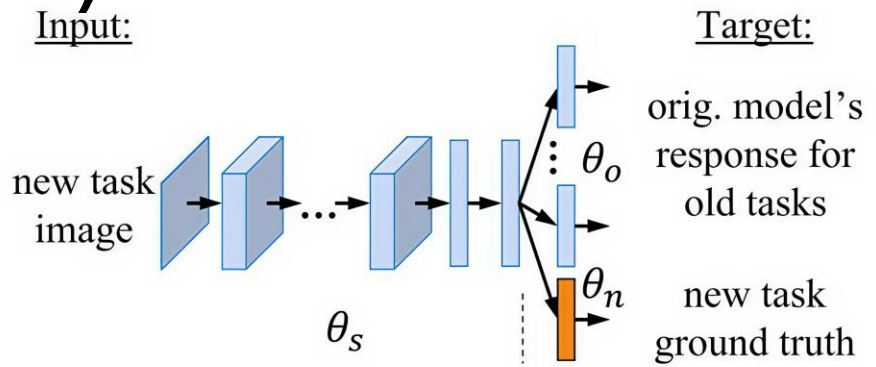
Learning without Forgetting (LwF)

- Use a task-head for each task



Li, Z., & Hoiem, D. (2017). Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12), 2935-2947.

Learning without Forgetting (LwF)



LEARNINGWITHOUTFORGETTING:

Start with:

θ_s: shared parameters

θ_o: task specific parameters for each old task

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

Initialize:

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

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

Train:

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

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

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

Li, Z., & Hoiem, D. (2017). Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12), 2935-2947.

Break

15 min break

Beyond Traditional Settings

Section IV

Continual Learning of LLMs

- Recall: the world changes continuously
  - Facts, Language, Code, and Norms
- LLMs are trained on static data, but needs to be adapted to the evolving world
- Retraining is slow and expensive
  - Huge clusters of GPUs/TPUs over weeks or months
  - Thousands to millions of GPU hours
- CL enables LLMs to update with new information, adapt to new domains and learn behaviors incrementally, while retaining the old knowledge

Continual Learning of LLMs

Parametric

Update Model Weights

Pretraining
Fine-tuning

Non-Parametric

Update Structure

External memory
Communication Graph
Prompt

Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025, October). From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning* (pp. 21497-21515). PMLR.

Continual Learning of LLMs

Parametric

Update Model Weights

Pretraining
Fine-tuning

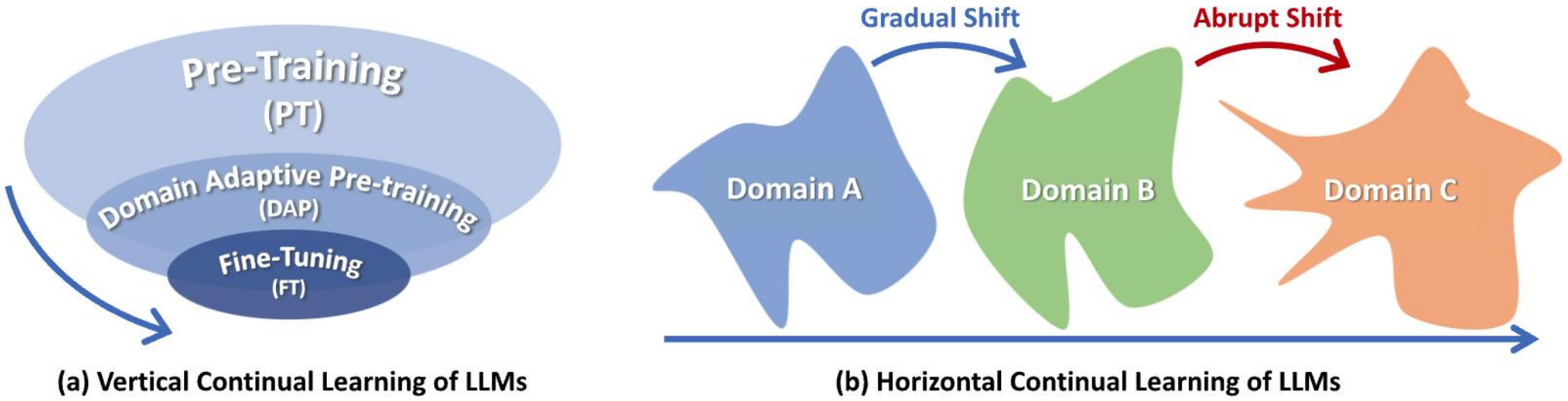
Non-Parametric

Update Structure

External memory
Communication Graph
Prompt

Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025, October). From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning* (pp. 21497-21515). PMLR.

Continual Learning of LLMs - Dimensions

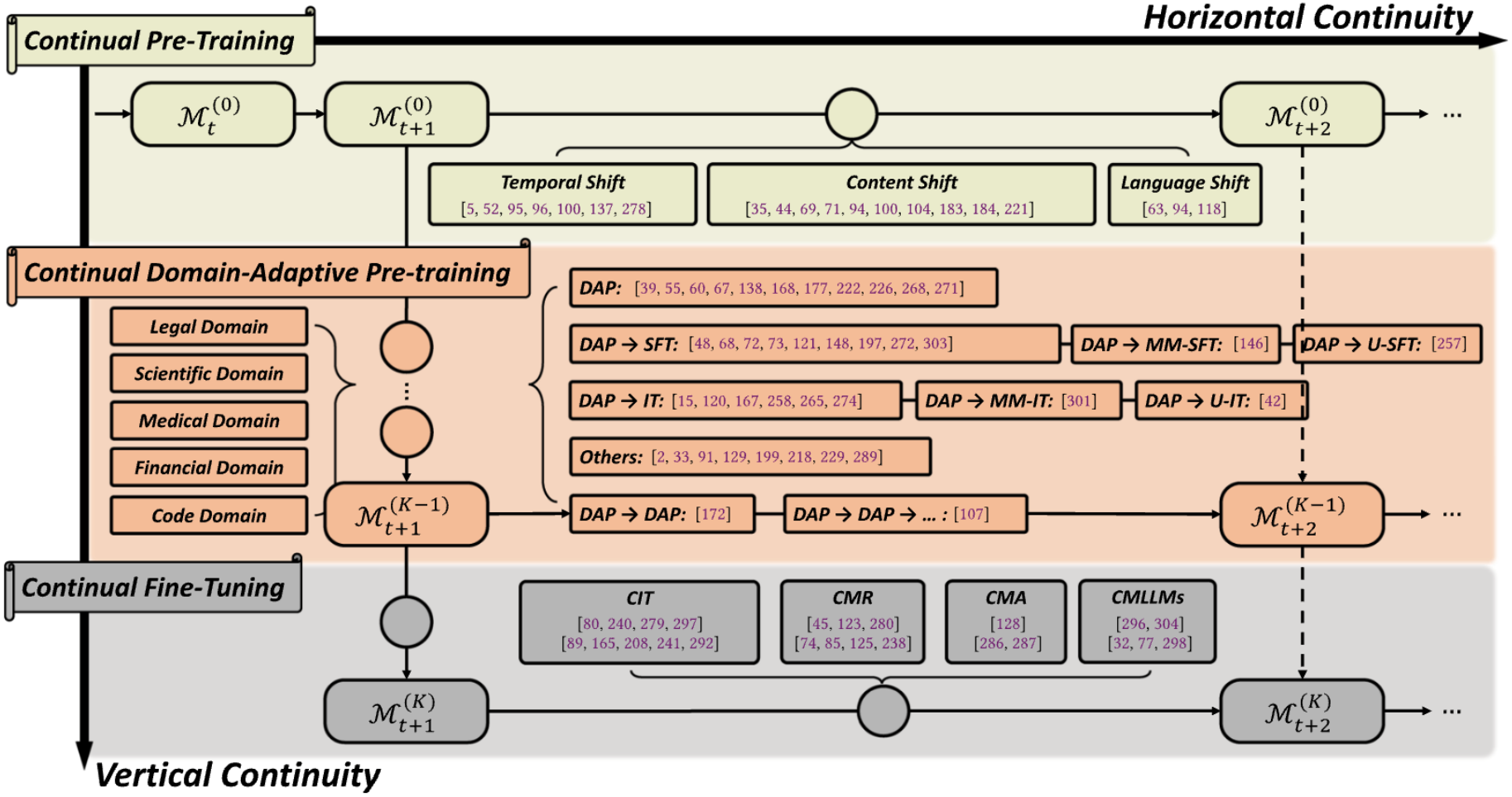


(a) Vertical Continual Learning of LLMs

(b) Horizontal Continual Learning of LLMs

Shi, H., Xu, Z., Wang, H., Qin, W., Wang, W., Wang, Y., ... & Wang, H. (2025). Continual learning of large language models: A comprehensive survey. ACM Computing Surveys, 58(5), 1-42.

Continual Learning of LLMs - Dimensions



Shi, H., Xu, Z., Wang, H., Qin, W., Wang, W., Wang, Y., ... & Wang, H. (2025). Continual learning of large language models: A comprehensive survey. ACM Computing Surveys, 58(5), 1-42.

Continual Instruction Tuning (CIT)

- Tune on new instructions as a stream
- Forgetting of the previously learned instructions

CTo

Improves instruction retention by replay

Key-Part Information Gain (KPIG)

Use masked parts to dynamically select replay data

SSR

Uses LLM to generate synthetic instances for replay

CTo

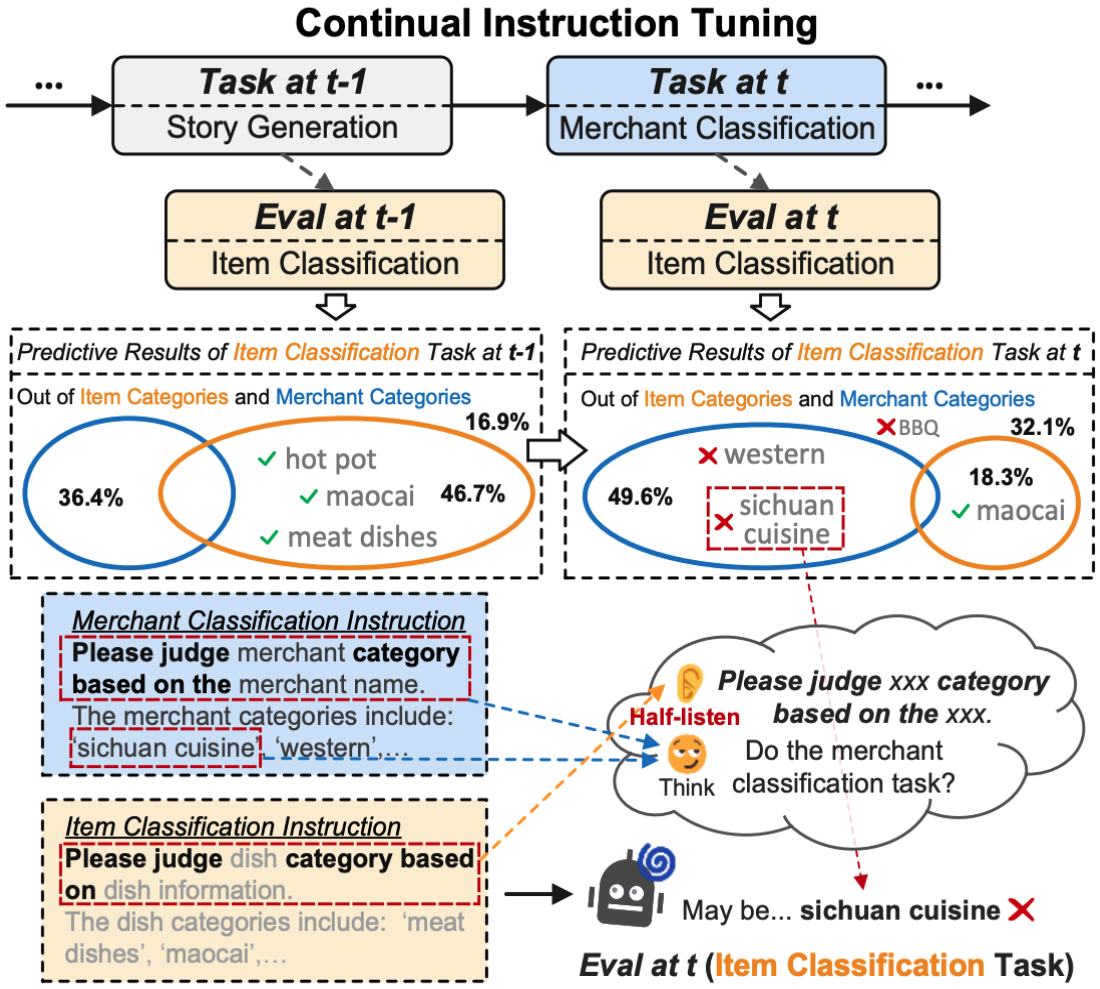
- Fine-tune on new instructions
  - Tasks => Instructions

Text Smpfl(Simp)	Instruction	Make this text simpler: "A Georgian inscription around the drum attests his name."
	Output	*A Georgian writing on the drum is his name.*
Headline Generation (HGen)	Instruction	Make a title for this article that begins with "protesters": police arrested five anti-nuclear protesters thursday after they sought to disrupt loading of a french antarctic research and supply vessel , a spokesman for the protesters said .
	Output	*protesters target french research ship*
Haiku Gen (Haiku)	Instruction	Generate a haiku about 'Seagulls crying high'
	Output	*Seagulls crying high / the air smelling of sea salt / Or is it my tears?*
Covid QA (CQA)	Instruction	In the context of the COVID pandemic,who is at greater risk of dying from COVID19?
	Output	*patients with underlying medical conditions and the elderly*

- Solution: Rehearsal based Continual Learning
  - Do not train on whole dataset

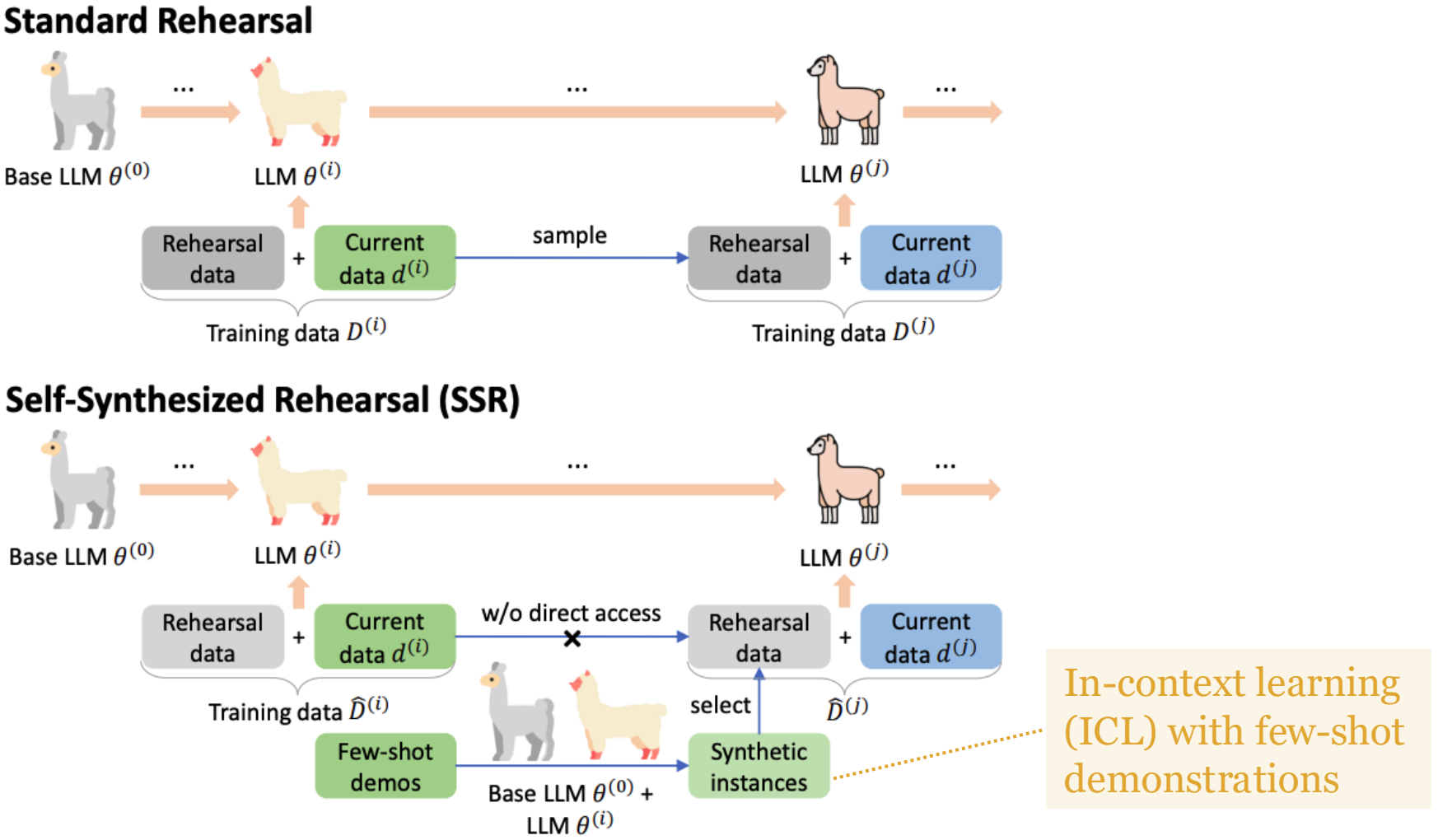
Scialom, T., Chakrabarty, T., & Muresan, S. (2022, December). Fine-tuned language models are continual learners. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (pp. 6107-6122).

Key-Part Information Gain (KPIG)



He, Y., Zhang, W., Huang, X., Zhang, P., Meng, L., Zhou, X., Zeng, K., & Cai, X. (2025). Don't half-listen: Capturing key-part information in continual instruction tuning. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 23649–23668). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.1153

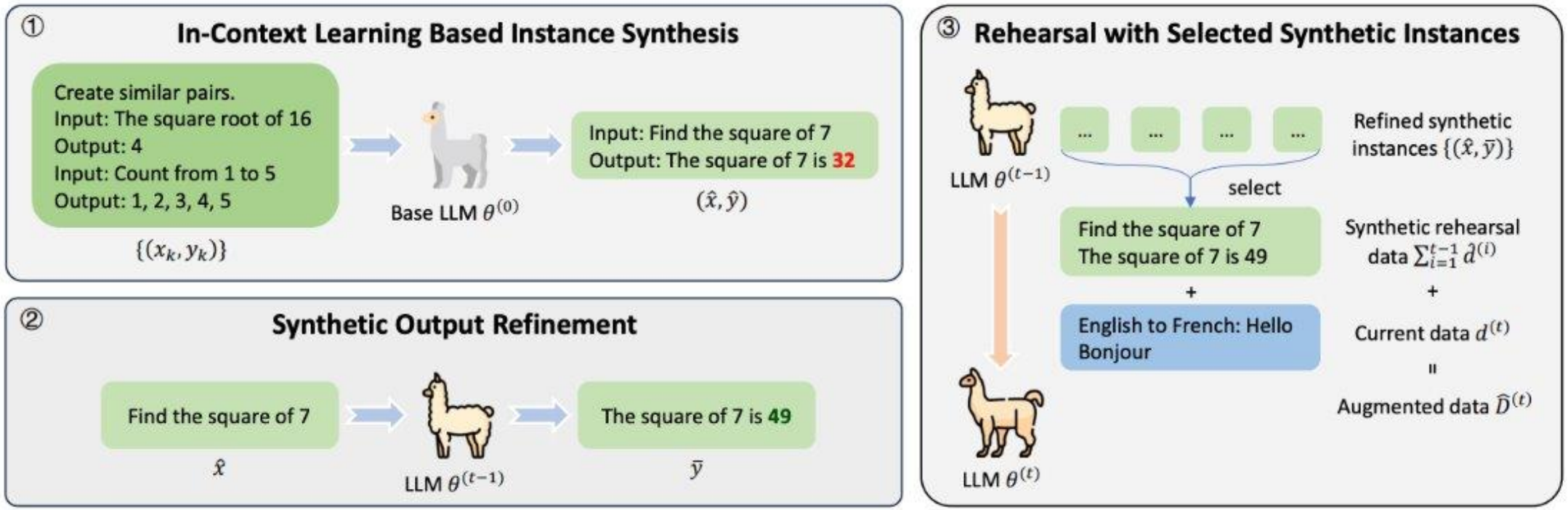
Self-Synthesized Rehearsal (SSR)



In-context learning (ICL) with few-shot demonstrations

Huang, J., Cui, L., Wang, A., Yang, C., Liao, X., Song, L., ... & Su, J. (2024, August). Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1416-1428).

SSR - Example



Huang, J., Cui, L., Wang, A., Yang, C., Liao, X., Song, L., ... & Su, J. (2024, August). Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1416-1428).

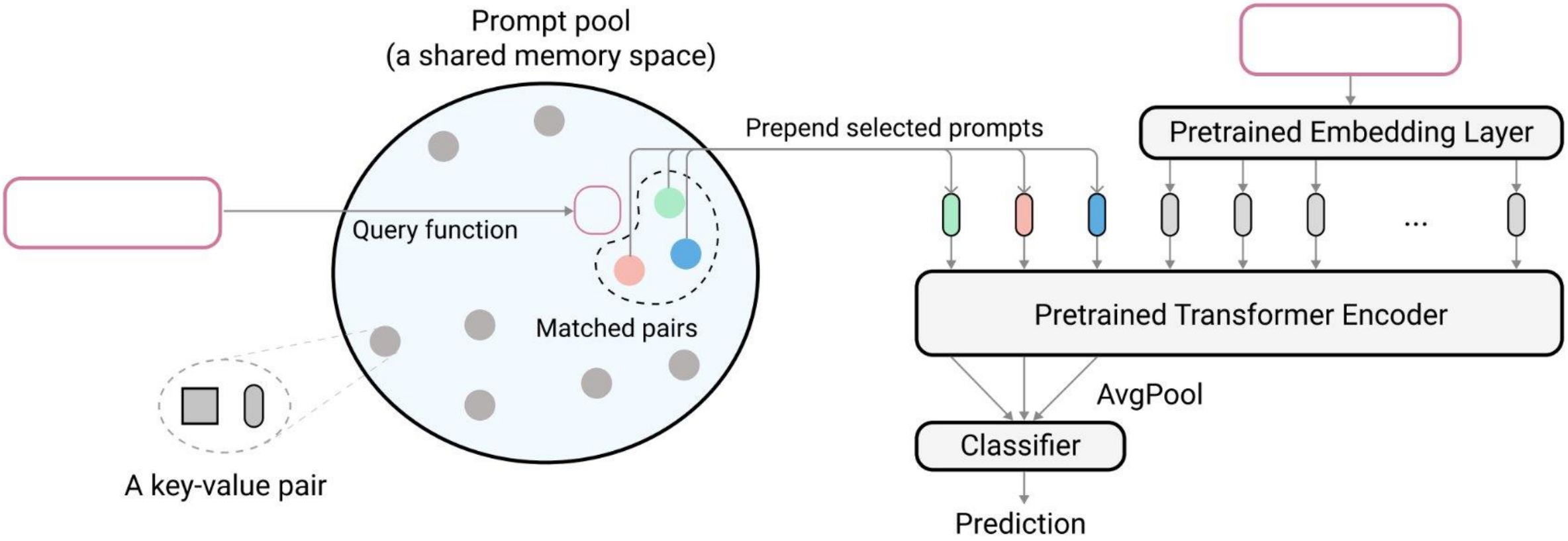
Prompt-based CL - L2P

- Learning to Prompt (L2P) for Continual Learning
- Solves the problem of cost to fine-tune

- Shared = pre-trained model
- Task-specific = prompts foreach task in CL

Wang, Z., Zhang, Z., Lee, C. Y., Zhang, H., Sun, R., Ren, X., ... & Pfister, T. (2022). Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 139-149).

Prompt-based CL - L2P



Wang, Z., Zhang, Z., Lee, C. Y., Zhang, H., Sun, R., Ren, X., ... & Pfister, T. (2022). Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 139-149).

Continual Learning of LLMs

Parametric

Update Model Weights

Pretraining
Fine-tuning

Non-Parametric

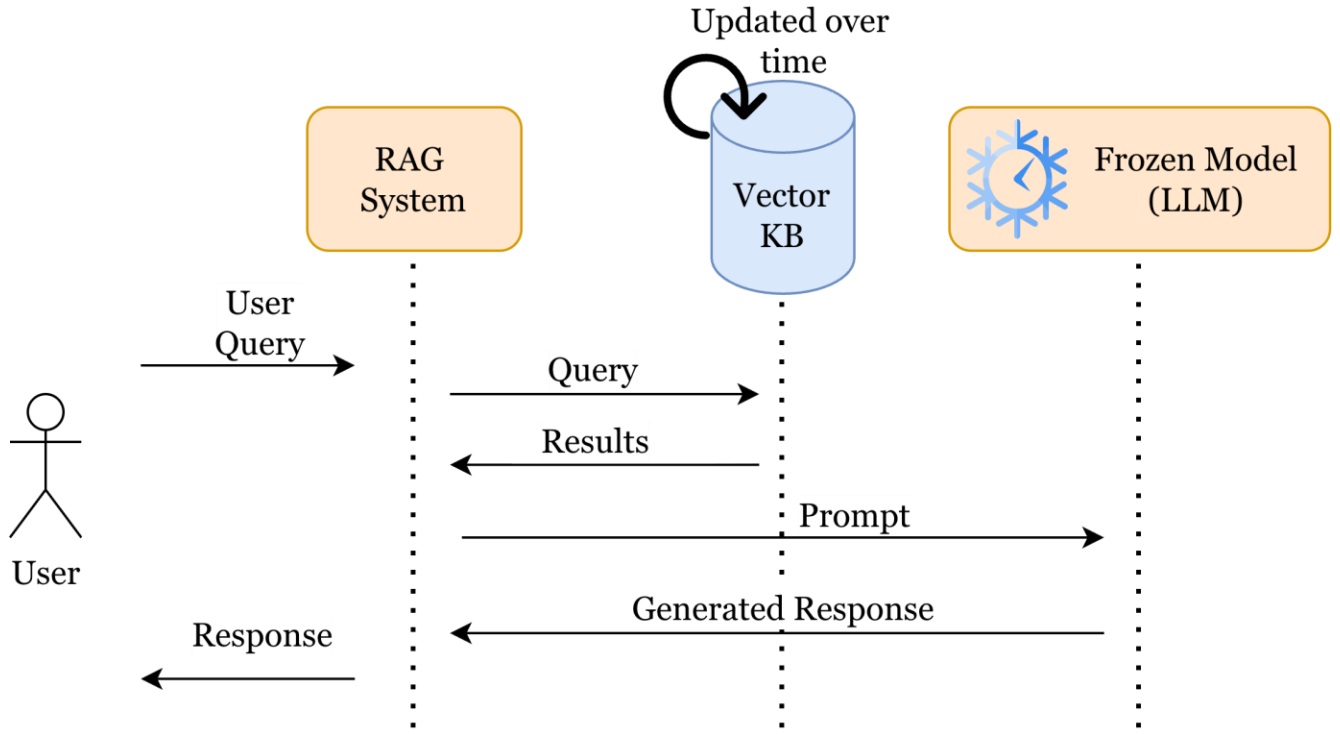
Update Structure

External memory
Communication Graph
Prompt

Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025, October). From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning* (pp. 21497-21515). PMLR.

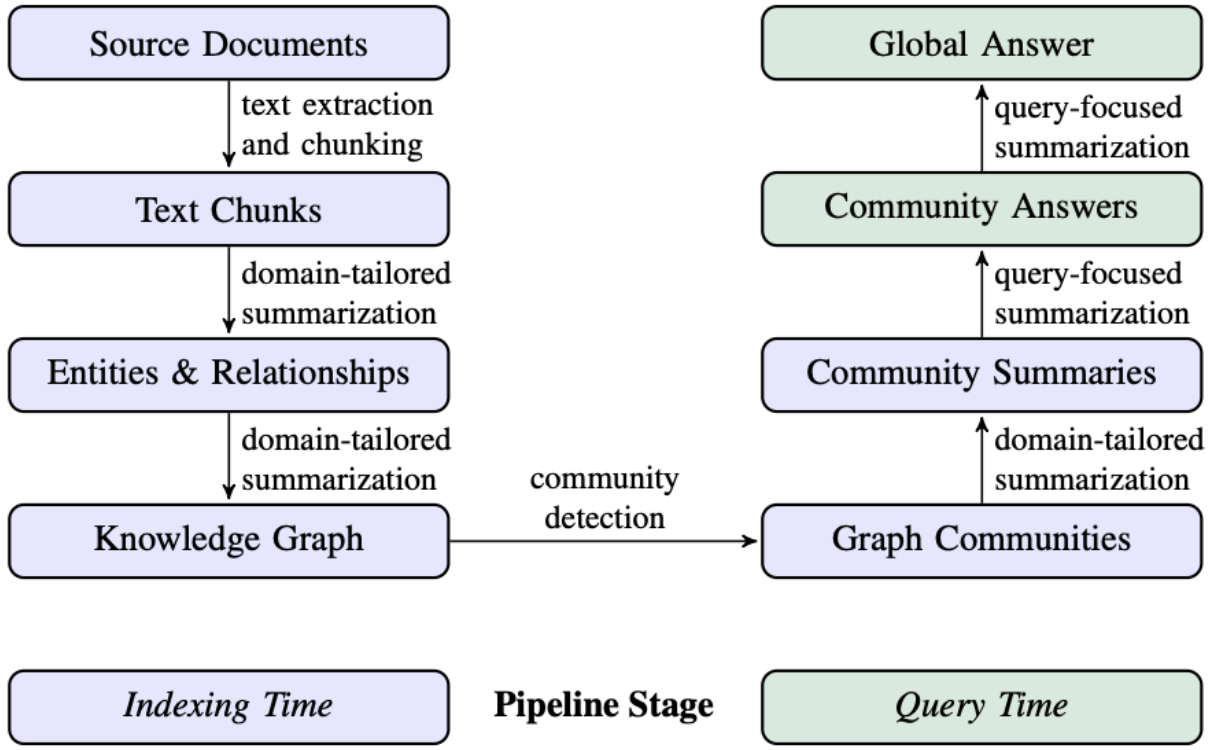
Retrieval-Augmented Generation (RAG)

- A vector KB grows continuously, no model retraining needed



Graph-RAG

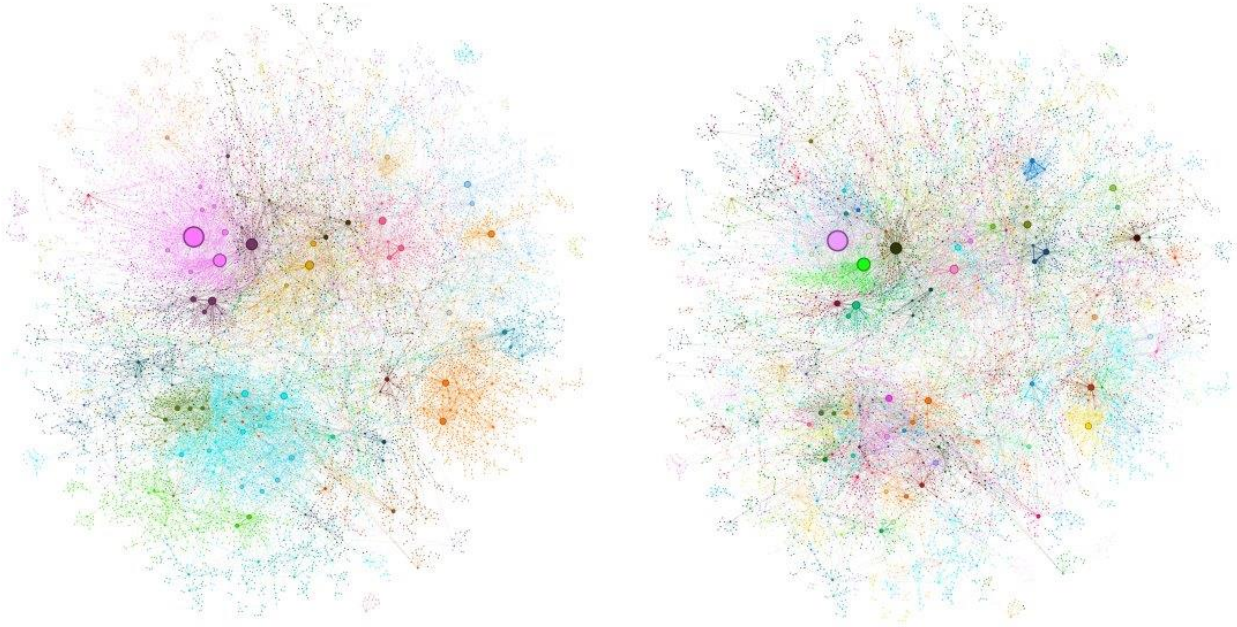
- Task: Query-Focused Summarization (QFS)
- Evolves from vector retrieval to graph-based routing and summarization
- Enables multi-hop reasoning and scalable knowledge organization



Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025, October). From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning* (pp. 21497-21515). PMLR.

Graph-RAG

- Improved
  - Comprehensiveness
  - Diversity
- Directness Tradeoff: Vector RAG produces the most direct responses
- Root-level community summaries token efficiency
  - Less tokens needed for Level 0 queries



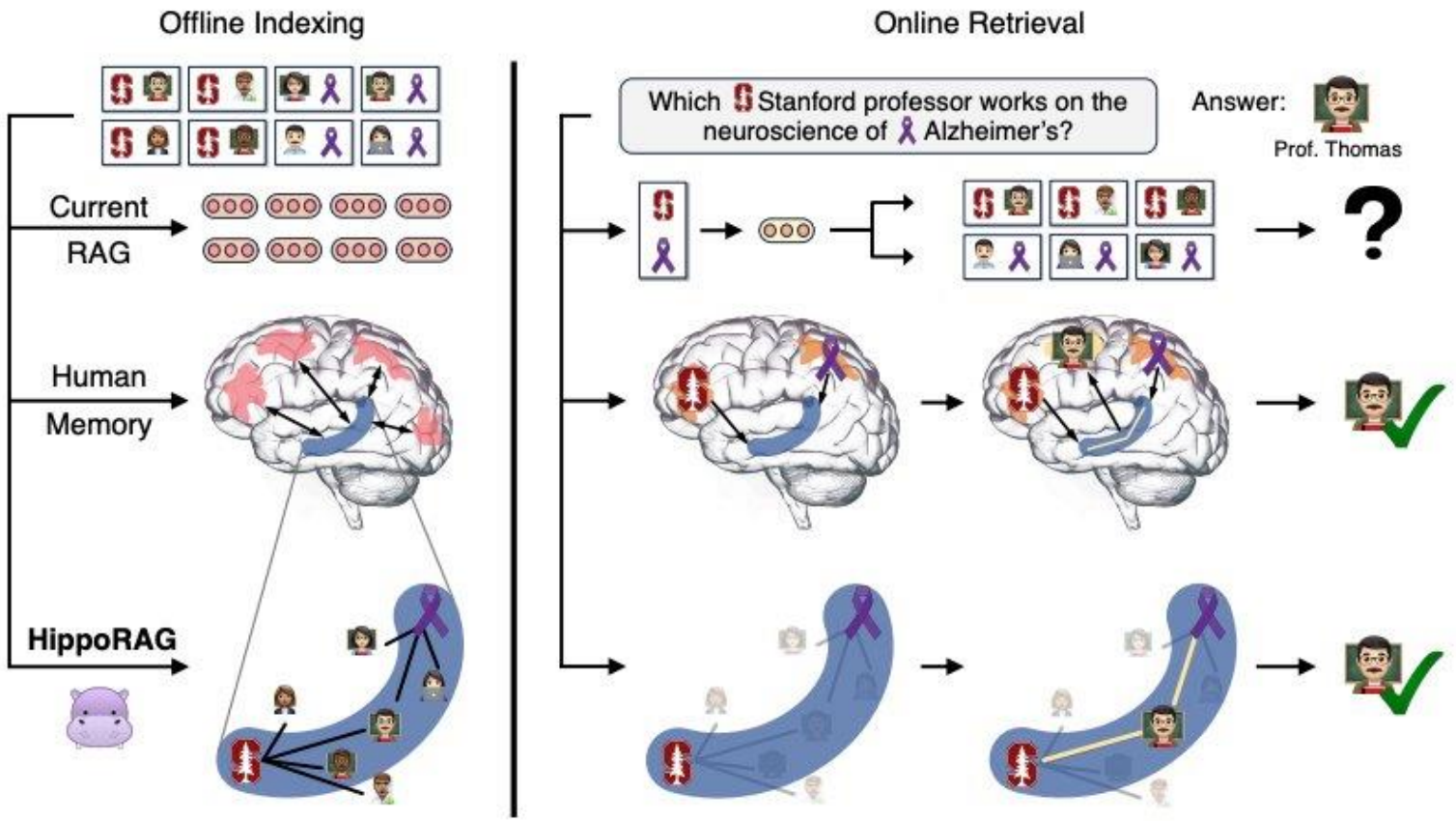
(a) Root communities at level 0 (b) Sub-communities at level 1

Leiden community detection

Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025, October). From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning* (pp. 21497-21515). PMLR.

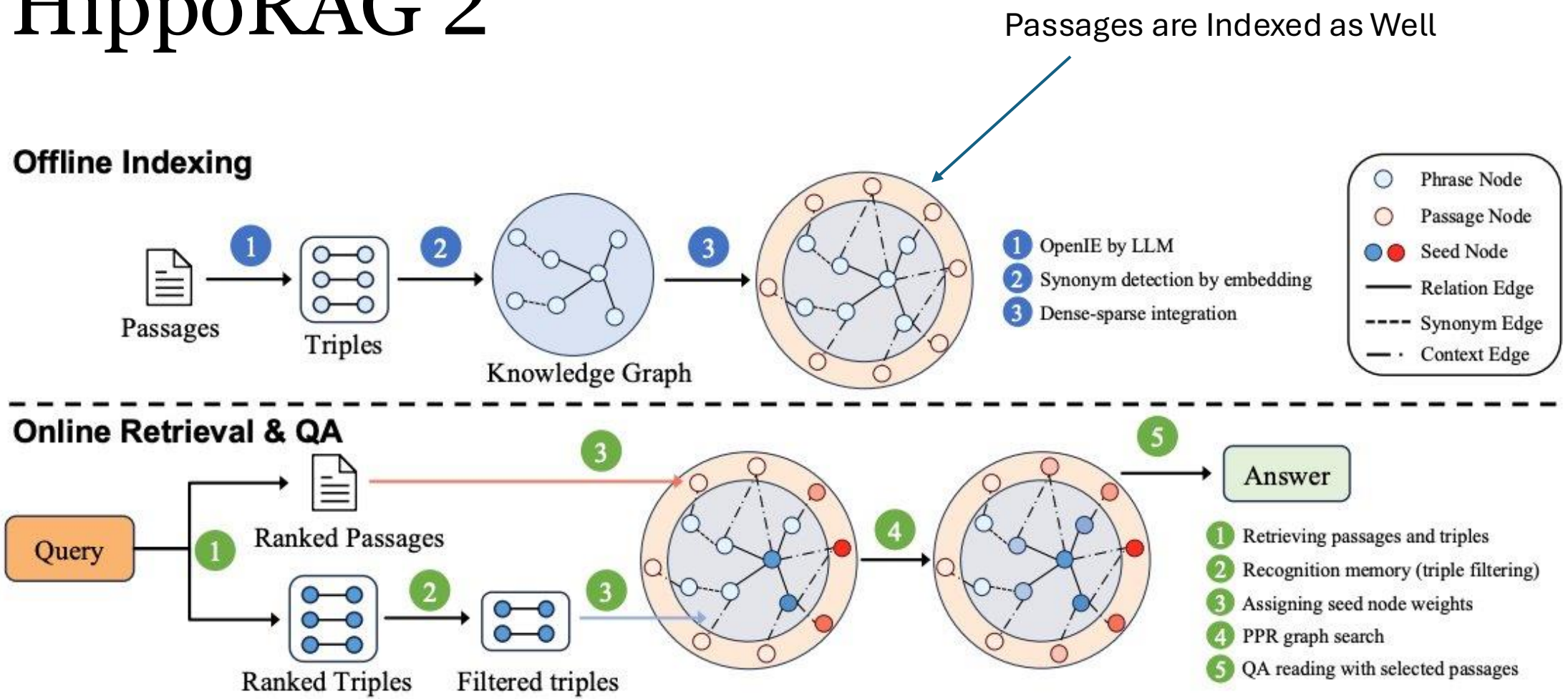
HippoRAG

- Task: Multi-hop Question Answering (QA)
- Personalized Page Rank (PPR)
- Mimic human memory
  - Neocortex
  - Hippocampus



Gutiérrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., & Su, Y. (2024). HippoRAG: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, *37*, 59532-59569.

HippoRAG 2



Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025, October). From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In International Conference on Machine Learning (pp. 21497-21515). PMLR.

RAG Comparison

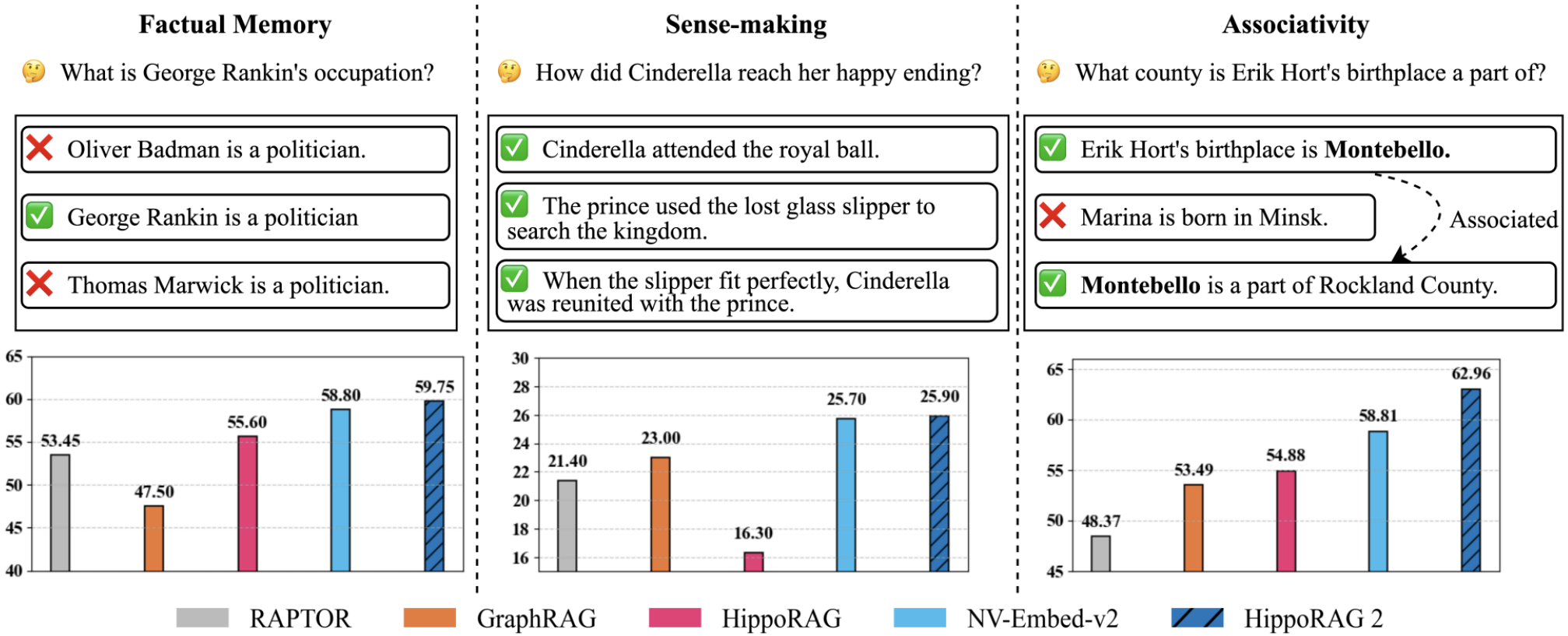
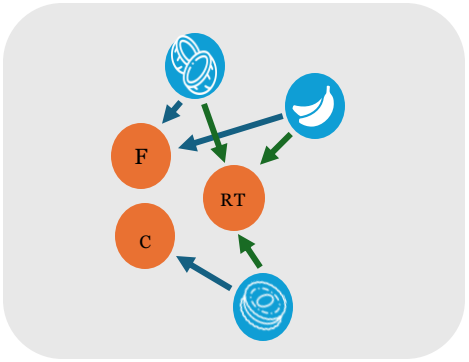


Figure 1. Evaluation of continual learning capabilities across three key dimensions: factual memory (NaturalQuestions, PopQA), sense-making (NarrativeQA), and associativity (MuSiQue, 2Wiki, HotpotQA, and LV-Eval). HippoRAG 2 surpasses other methods across all benchmark categories, bringing it one step closer to a true long-term memory system.

Knowledge-guided Continual Learning (KGCL)

- External structured knowledge help to improve the performance of CL approaches like reply



External Knowledge (Knowledge Graph)

My cousin is such an **oreo** now. #Traitor

Internal Knowledge (Memory Buffer)

Shallow Augmentation

My cousin is such an oreo now. #Traitor

My cousin is such an Race Traitor now. #Traitor

…

My cousin is such an Cookie now. #Traitor

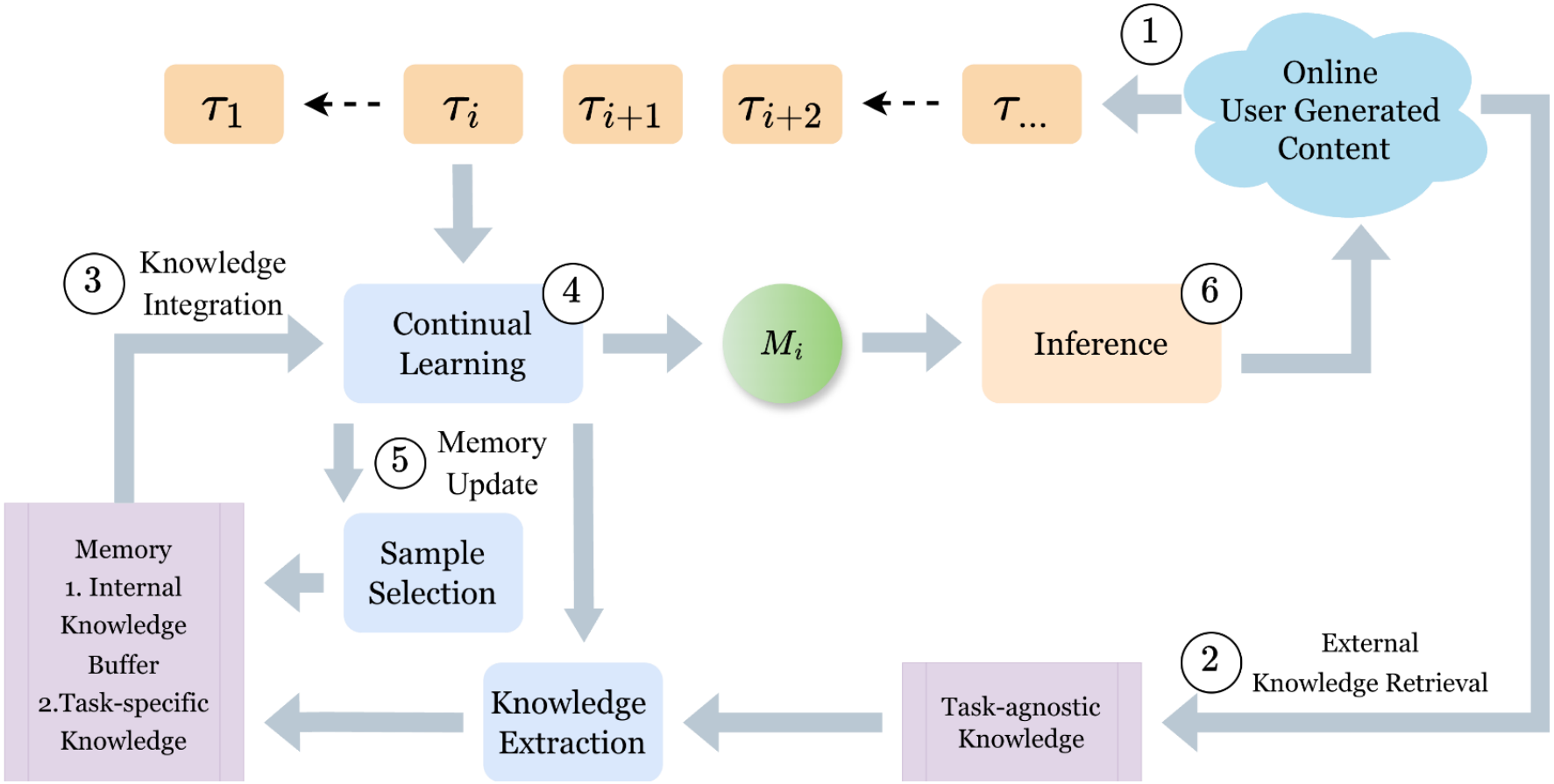
T_2: Religion-based Threats

New Model (M_2) Training

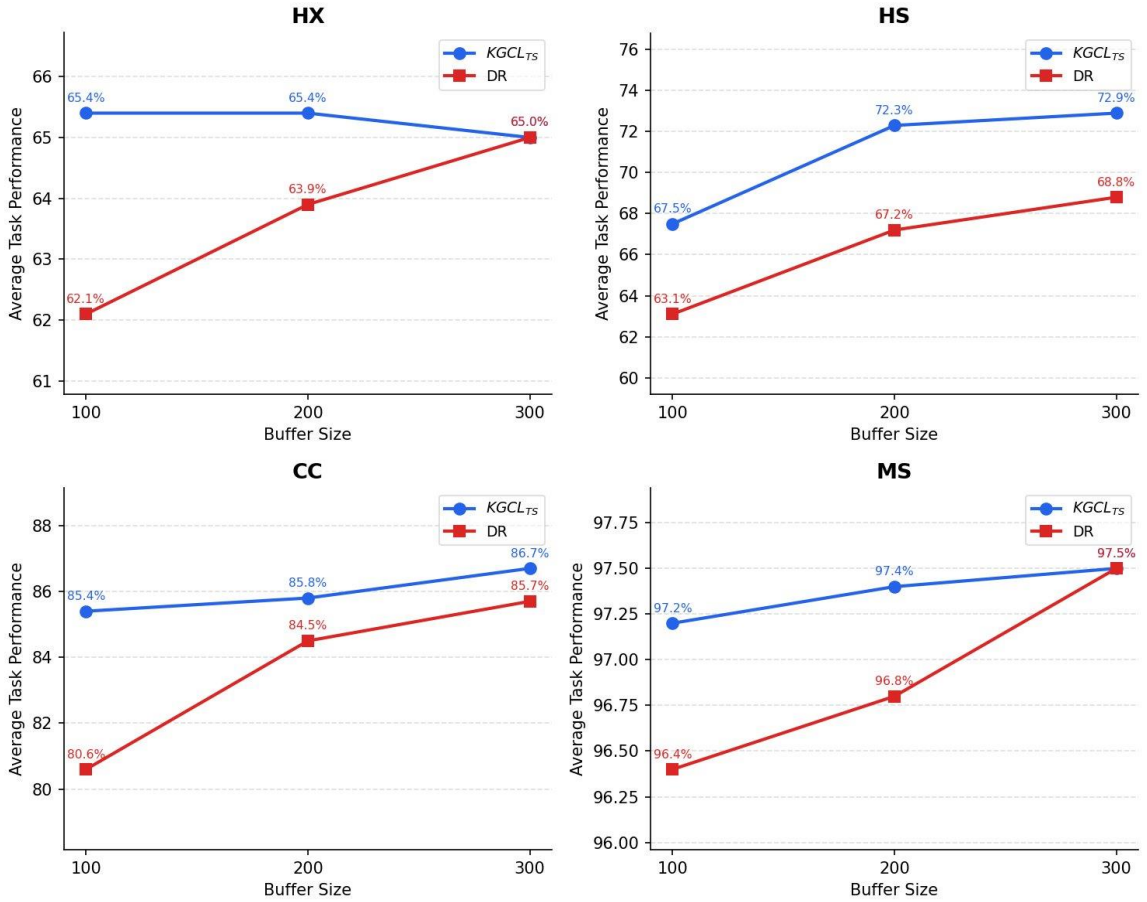
Incorrect - Knowing relevance of knowledge relations to tasks can help to avoid this

Senarath, Y., & Purohit, H. (2025). Knowledge-guided continual learning for behavioral analytics systems. 2025 IEEE 7th International Conference on Cognitive Machine Intelligence (CogMI), 1–11. https://doi.org/10.1109/CogMI67134.2025.00011

Knowledge-guided Continual Learning (KGCL)



Knowledge-guided Continual Learning (KGCL)



Demo II – CL Methods

https://github.com/human-info-lab/ICWSM-2026-Continual-Learning-Tutorial-Code

Open Challenges and Future Directions

Section V

Open Challenges and Future Directions

Note to readers: This part will be updated in the tutorial website soon!

Continual Learning Tools

- Avalanche: https://avalanche.continualai.org/
  - "Avalanche is an End-to-End Continual Learning Library based on PyTorch, born within ContinualAI with the goal of providing a shared and collaborative open-source (MIT licensed) codebase for fast prototyping, training and reproducible evaluation of continual learning algorithms."
- Continuum: https://continuum.readthedocs.io/en/latest/
  - "Continuum is the library you need for Continual Learning. It supports many datasets and most scenarios (NC, NI, NIC, etc.)."
- PyCIL: https://github.com/LAMDA-CL/PyCIL
  - Presents itself as "toolbox for class-incremental learning with the most implemented methods"

Srijith, P. K., Satapara, S., & Chandar, S. (n.d.). Continual learning in large language models: Foundations to frontiers. In B. Heinzerling & L.-W. Ku (Eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: Tutorial Abstract*. Association for Computational Linguistics.

Continual Learning Tools

- Renate: https://renate.readthedocs.io/en/latest/index.html
  - "Python package for automatic retraining of neural networks models. It uses advanced Continual Learning and Lifelong Learning algorithms to achieve this purpose."
- SequeL: https://nik-dim.github.io/sequel-site/
  - Provides a simple and easy to use framework for continual learning. Written in PyTorch and JAX. The library is still in development!
- ContinualLM: https://github.com/UIC-Liu-Lab/ContinualLM
  - "Imagine an LM that not only effortlessly acquires new knowledge but also retains its mastery of skills, all while successfully transferring knowledge. Is it even possible?"
  - Good for NLP!

Srijith, P. K., Satapara, S., & Chandar, S. (n.d.). Continual learning in large language models: Foundations to frontiers. In B. Heinzerling & L.-W. Ku (Eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: Tutorial Abstract*. Association for Computational Linguistics.

Thank you.

Questions and Discussion

Yasas Senarath | Ph.D. | yasas.mason@gmail.com

Marcos Zampieri · Hemant Purohit
Information Sciences and Technology (IST) Department
George Mason University, Fairfax, Virginia