



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Continual learning: Introduction and Rehearsal free-continual learning

Machine Learning group, BKAI
Presenter: Ngô Văn Linh

ONE LOVE. ONE FUTURE.

1. Phân biệt một số kiểu học (learning paradigms)
2. Học liên tục (Continual learning)
3. Rehearsal-free continual learning
4. References

1. Phân biệt một số kiểu học (learning paradigms)

• Lifelong learning [1]

• *Nhận xét về cách học của con người:*

- Con người tích lũy kiến thức theo thời gian.
- Khi gặp vấn đề mới, một số kiến thức đã học từ trước phù hợp và có liên quan được sử dụng để giải quyết nó.
- Qua thời gian, khi đã có nhiều kiến thức hơn thì việc học trở nên dễ dàng hơn.

⇒ Mục tiêu của lifelong learning là **bắt chước cách học của con người.**

[1] Zhiyuan Chen, Bing Liu, Ronald Brachman, Peter Stone, and Francesca Rossi. 2018. Lifelong Machine Learning (2nd. ed.). Morgan &

1. Phân biệt một số kiểu học (learning paradigms)

- Lifelong learning

- **Định nghĩa:**

- Tại thời điểm thứ t :

- **Previous tasks** gồm $N - 1$ tasks T_1, \dots, T_{N-1} tương ứng với các bộ dữ liệu D_1, \dots, D_{N-1} .

- Các tasks này có thể thuộc các loại khác nhau và các lĩnh vực khác nhau (***different types and different domains***).

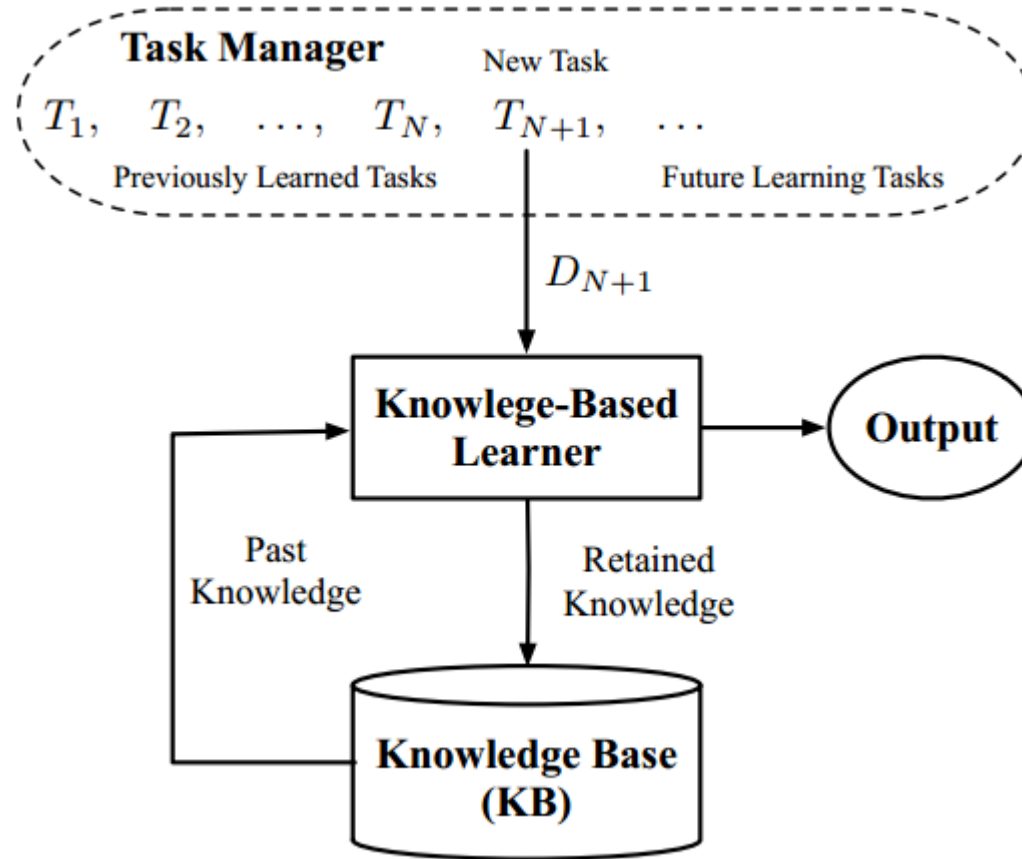
- **Current task** (new task) là T_N với dữ liệu D_N .

- Tận dụng kiến thức đã học (***past knowledge***) trong ***knowledge base (KB)*** để học task mới này.

- Sau khi học xong task mới, cập nhật tri thức mới học được vào KB.

1. Phân biệt một số kiểu học (learning paradigms)

- Lifelong learning



Kiến trúc hệ thống lifelong learning [1]

1. Phân biệt một số kiểu học (learning paradigms)

- **Lifelong learning**

- **3 đặc điểm chính (key characteristics):**

- Là quá trình học liên tục (**continuous learning process**)
- Lưu giữ và tích lũy (**retain and accumulate**) kiến thức theo thời gian.
- Sử dụng kiến thức đã học được để giúp học những công việc mới (**use previous knowledge to help new learning tasks**).

⇒ Một số kiểu học khác có đặc điểm liên quan: ***transfer learning, multi-task learning, meta learning, online learning.***

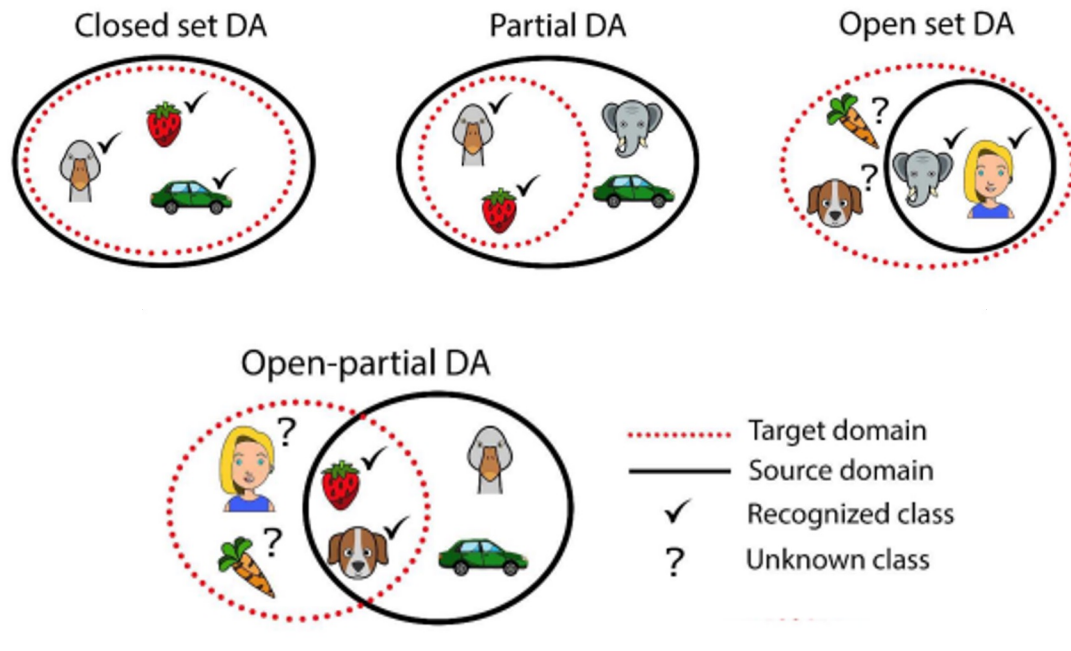
1. Phân biệt một số kiểu học (learning paradigms)

- **Transfer learning [1]**

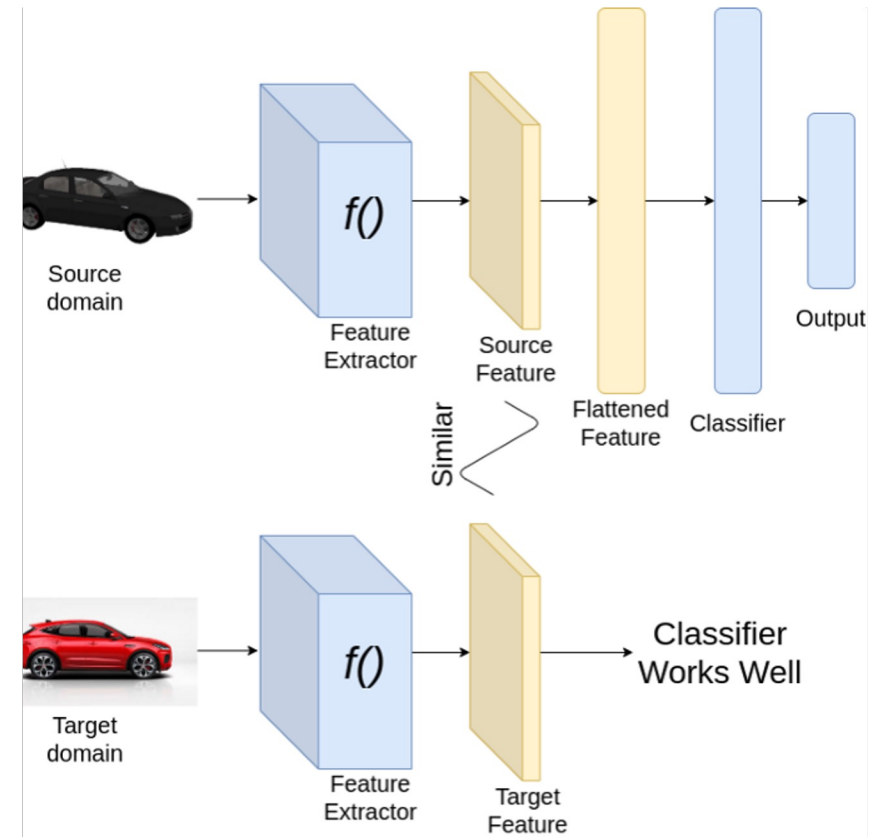
- Trong NLP, CV còn được biết đến là **domain adaptation**.
- Thường liên quan đến 2 domains: **source domain** và **target domain**, cụ thể:
 - **Source domain**: một lượng lớn dữ liệu được gán nhãn.
 - **Target domain**: ít dữ liệu được gán nhãn hoặc không có.
 - 2 domains này được giả sử là tương tự nhau (similar domains) và thường được chọn bởi con người.
- **Mục tiêu của transfer learning**: sử dụng dữ liệu được gán nhãn ở source domain để giúp việc học trong target domain.

1. Phân biệt một số kiểu học (learning paradigms)

• Domain adaptation



Phân loại domain adaptation theo nhãn lớp [2]



Ví dụ về kiến trúc học trong DA [2]

1. Phân biệt một số kiểu học (learning paradigms)

- So sánh giữa lifelong learning và transfer learning

	Transfer learning	Lifelong learning
Giống nhau	Sử dụng kiến thức trước đó để học kiến thức mới.	
Khác nhau	- Không phải quá trình học liên tục.	- Là quá trình học liên tục.
	- Không tích lũy kiến thức.	- Tích lũy kiến thức.
	- 1 chiều (unidirectional): chỉ có kiến thức từ source domain được dùng để học target domain.	- Kết quả học từ domain hay task mới có thể dùng để cải thiện việc học trong domains hay tasks trước đó.
	- Thường chỉ liên quan tới 2 domains: một source domain và một target domain.	- Quan tâm tới một lượng lớn tasks/ domains.
	- Giả sử source domain và target domain là tương tự nhau (similar domains).	- Không có giả sử nào như transfer learning.

1. Phân biệt một số kiểu học (learning paradigms)

- **Multi-task learning (Batch multi-task learning) [1]**
 - Quan tâm đến việc học nhiều tasks liên quan đồng thời.
 - **Mục tiêu của multi-task learning:** maximize hiệu năng trên tất cả các tasks.
 - Mở rộng của batch multi-task learning là **online multi-task learning:**
 - Học tasks một cách tuần tự.
 - Tích lũy kiến thức qua thời gian để áp dụng học tasks mới (hoặc cải thiện tasks trước đó).
 - **Online multi-task learning là lifelong learning.**

1. Phân biệt một số kiểu học (learning paradigms)

- Multi-task learning và lifelong learning

	Multi-task learning	Lifelong learning
Giống nhau	Sử dụng một vài thông tin được chia sẻ qua các tasks để giúp cho việc học.	
Khác nhau	- Không phải quá trình học liên tục.	- Là quá trình học liên tục.
	- Không tích lũy kiến thức.	- Tích lũy kiến thức.

- Một số cách tiếp cận của multi-task learning được sử dụng trong lifelong learning: architecture-based approach (Parameter Sharing), Gradient-based multi-objective optimization

1. Phân biệt một số kiểu học (learning paradigms)

- **Online learning/streaming learning/incremental learning [1]**
 - Dữ liệu đến theo dạng chuỗi.
 - **Mục tiêu của online learning:** tối ưu hiệu năng trên task cần học (given learning task).
 - Thường được sử dụng khi:
 - Việc huấn luyện trên toàn bộ tập dữ liệu là không khả thi.
 - Ứng dụng thực tế không thể chờ đến khi có đủ lượng dữ liệu được thu thập.

1. Phân biệt một số kiểu học (learning paradigms)

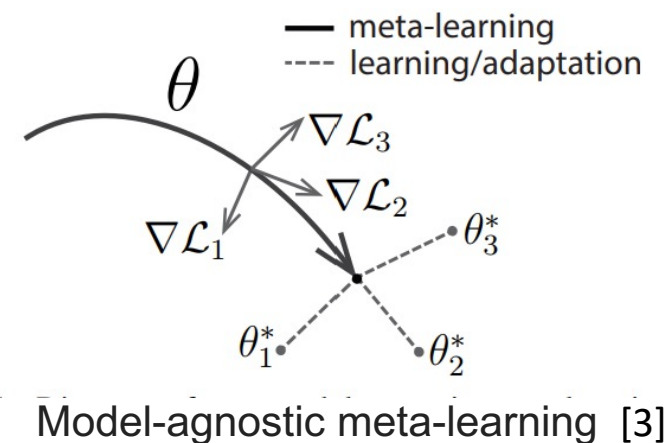
• Online learning và lifelong learning

	Online learning	Lifelong learning
Giống nhau	Quá trình học liên tục.	
Khác nhau	- Hầu hết các nghiên cứu về online learning hướng tới xử lý cùng một domain/task theo thời gian.	- Hướng tới học từ một chuỗi các tasks/domains khác nhau.
	- Không tích lũy kiến thức để học tasks mới.	- Tích lũy kiến thức để giúp việc học tasks mới. - Sử dụng kiến thức cũ để giúp việc học tasks mới.

- Lifelong learning sử dụng cách tiếp cận hiệu chỉnh (regularization/prior-based approach) của online learning cho các shared parameters để chuyển giao tri thức giữa các tasks

1. Phân biệt một số kiểu học (learning paradigms)

- **Meta learning [1]** còn được gọi là “học cách học” (learning to learn)
 - **Mục tiêu của meta learning:** học task mới chỉ với lượng nhỏ dữ liệu huấn luyện bằng việc sử dụng một model đã được huấn luyện trên nhiều những tasks (ví dụ: N-way-k-shot classification)
 - Hệ thống meta learning gồm 2 phần:
 - Meta learner (slow learner)
 - Base learner (quick learner/adaptation module)



⇒ model học được từ meta learner giúp base learner học hiệu quả chỉ với một lượng rất ít dữ liệu huấn luyện.

1. Phân biệt một số kiểu học (learning paradigms)

• Meta learning và lifelong learning

	Meta learning	Lifelong learning
Giống nhau	- Sử dụng kiến thức từ tasks này để học task mới.	
Khác nhau	<ul style="list-style-type: none">- Hầu hết các nghiên cứu về meta learning đưa ra giả sử tập dữ liệu tương tự nhau- Lấy mẫu tasks trong bước meta learner từ tập tasks cố định- Hướng tới thích nghi nhanh với task mới	<ul style="list-style-type: none">- Lifelong learning hướng tới giải quyết các tasks khác nhau từ những domains khác nhau.- Các task được học liên tục không lặp lại- Hướng đến cả thích nghi nhanh task mới và tránh quên task cũ

2. Học liên tục (Continual learning)

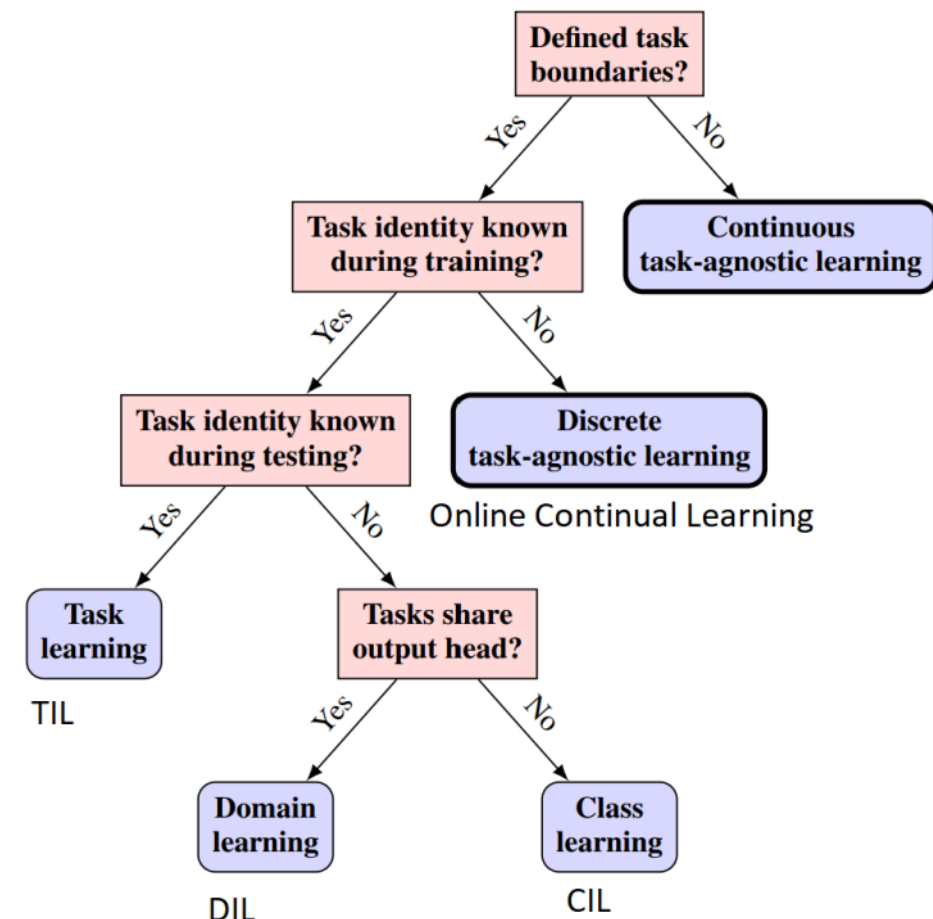
- Ban đầu, Lifelong learning trong deep learning thường được gọi là *continual learning* [1]
- Continual learning quan tâm đến vấn đề quên nghiêm trọng (*catastrophic forgetting*) [7]
 - Khi huấn luyện trên tasks mới, neural network có xu hướng quên thông tin đã học được từ những tasks trước.

⇒ Làm giảm hiệu suất mô hình trên những tasks cũ.

- Hiện tượng này liên quan chặt chẽ đến hiện tượng *stability-plasticity dilemma* [8].
 - Nếu model quá ổn định thì khó khăn học thông tin mới.
 - Nếu model quá mềm dẻo thì gặp vấn đề quên đi thông tin đã học được.
- *Continual learning được xem xét trong các ML models: SVM [4], Topic models [5], decision tree [6]*

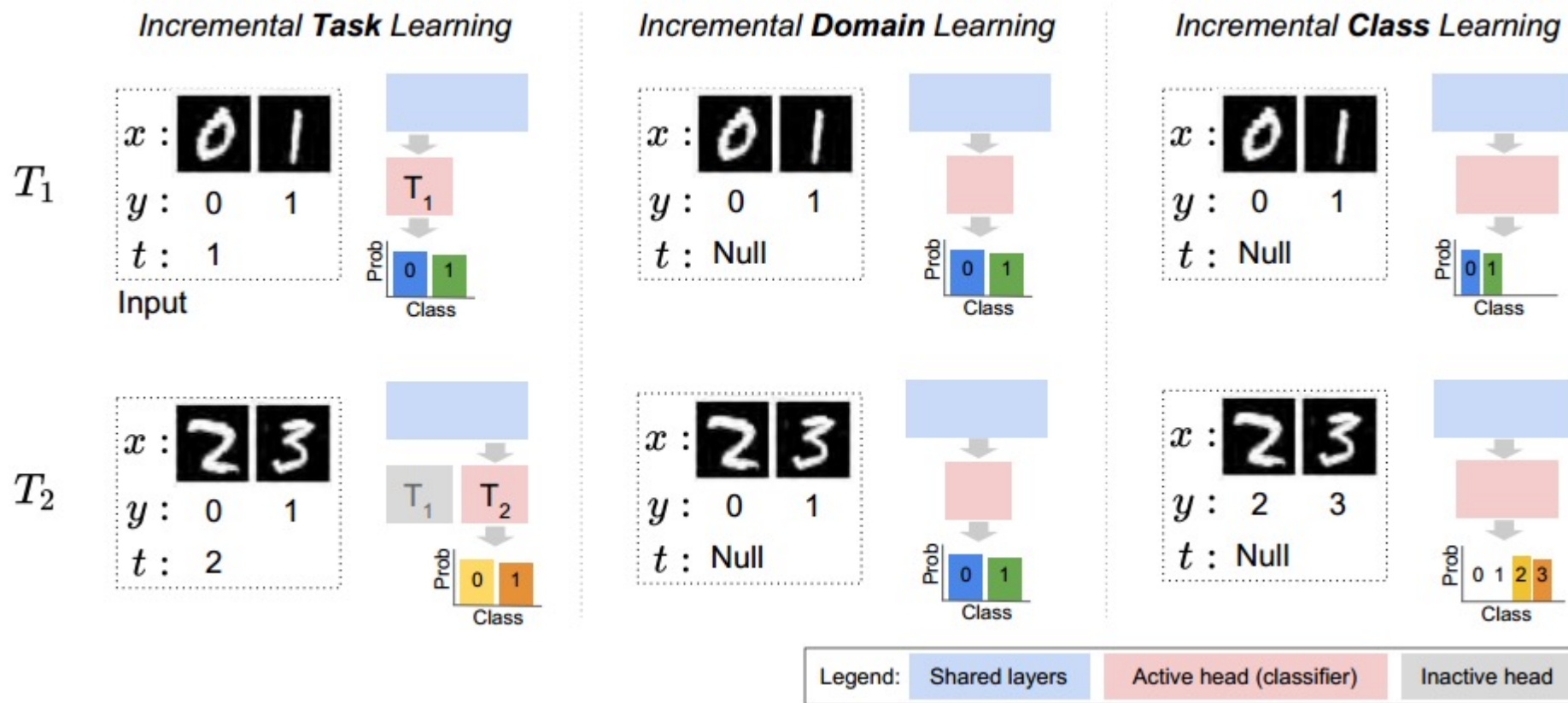
2. Học liên tục (Continual learning)

- Continual learning: Học chuỗi các tasks liên tục
- 3 kịch bản phổ biến [9]
 - Task-Incremental Learning (TIL).
 - Domain-Incremental Learning (DIL).
 - Class-Incremental Learning (CIL).
- Online continual learning: Nghiên cứu continual learning trong **online setup** (dữ liệu đến liên tục, phân phối của dữ liệu thường xuyên thay đổi).



Một số kịch bản chính trong CL [10]

2. Học liên tục (Continual learning)



Minh họa các kịch bản CL [9]

2. Học liên tục (Continual learning): Evaluation Metrics

- Sử dụng tập hold-out test cho mỗi task trong T tasks
- Huấn luyện lần lượt các task và đánh giá trên tập hold-out test của các tasks: Xây dựng matrix $R_{T \times T}$ trong đó $R_{i,j}$ là accuracy của mô hình trên task j khi học xong task i .
- 3 độ đo phổ biến: Average Accuracy (ACC), Backward Transfer (BWT), Forward Transfer (FWT)

- $ACC = \frac{1}{T} \sum_{i=1}^T R_{T,i}$

- $BWT = \frac{1}{T-1} \sum_{i=1}^T (R_{T,i} - R_{i,i})$

- $FWT = \frac{1}{T-1} \sum_{i=1}^T R_{i-1,i}$

2. Học liên tục (Continual learning): Một số cách tiếp cận

- Ba cách tiếp cận phổ biến:
 - Prior/Regularization-based approach
 - Memory-based approach
 - Architecture-based approach

2. Học liên tục (Continual learning): Prior/Regularization-based approach

- Bổ sung đại lượng regularization hoặc distillation loss vào hàm loss hiện tại để giữ tham số mô hình hiện tại “gần” với tham số mô hình ở task trước

$$\mathcal{L}_k(\theta) = L_k(\theta) + \lambda(\theta - \theta^{t-1})^T \Omega^{t-1}(\theta - \theta^{t-1})$$

với θ là tham số mô hình (weights), L_k là hàm loss cho task k , Ω^{t-1} mã hóa độ quan trọng của weights

- Nhiều cách khác nhau mã hóa độ quan trọng
 - Elastic Weight Consolidation (EWC [11]) sử dụng Fisher Information
 - Synaptic Intelligence (SI) [12] dựa trên giảm của hàm loss
 - Memory Aware Synapses (MAS) [13] dựa trên biến đổi của đầu ra (output)

2. Học liên tục (Continual learning): Prior/Regularization-based approach

- Variational Continual Learning (VCL) [14] sử dụng cách tiếp cận Bayes:

$$P(\theta|D_1, D_2, \dots, D_t) \propto P(D_t|\theta)P(\theta|D_1, D_2, \dots, D_{t-1})$$

Với D_t là dữ liệu huấn luyện của task t

- Tri thức học được ở những task trước đóng vai trò là tiên nghiệm (prior) cho task hiện tại:

$$P(\theta|D_1, D_2, \dots, D_{t-1}) = q_{t-1}(\theta)$$

- Sử dụng online variational inference

$$E_{q(\theta)} \log P(D_t|\theta) - KL(q(\theta)||q_{t-1}(\theta))$$

với $q(\theta)$ là xấp xỉ phân phối hậu nghiệm ở task hiện tại

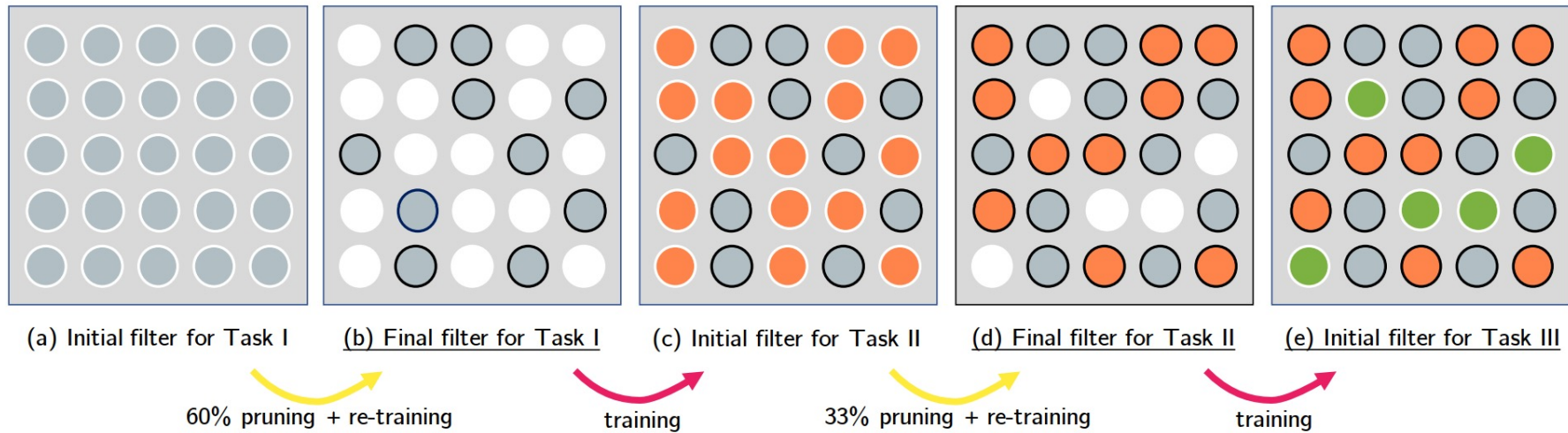
2. Học liên tục (Continual learning): Memory-based approach

- Sử dụng bộ nhớ nhỏ (memory buffer) [15] để lưu lại data từ các tasks trước và huấn luyện lại cùng với task hiện tại (rehearsal)
- Nhiều chiến lược lựa chọn data đặc trưng được đề xuất
 - Chọn data đại diện dựa theo lớp (herding algorithm)
 - Chọn data “khó”: Nằm gần biên phân loại (decision boundary) hoặc gây ra loss lớn
 - Sử dụng các chiến lược bi-level optimization
- Xây dựng các mô hình generative models [16] để sinh lại dữ liệu cũ hoặc sinh lại biểu diễn của dữ liệu cũ
- Gradient Episodic Memory (GEM) [17] and A-GEM [18] lưu dữ liệu để ngăn cập nhật weight theo hướng gradient làm giảm loss trên dữ liệu cũ.

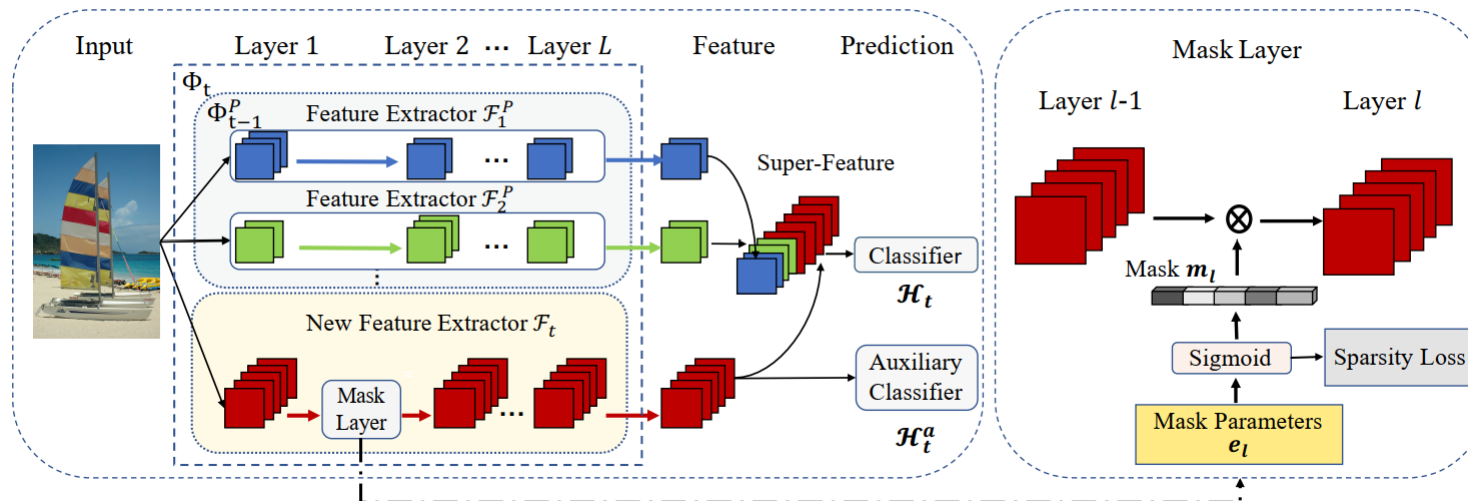
2. Học liên tục (Continual learning): Architecture-based approach

- Học một phần kiến trúc (sub-network) [19] cho mỗi task và lưu sub-network
- Đóng băng phần kiến trúc này và mở rộng mạng để học task mới
- Lặp lại 2 bước
 - Network expansion
 - Sparse learning
- Với TIL, tiếp cận dựa trên kiến trúc không bị quên task cũ
- Với DIL và CIL, cần bước xác định task boundary khi inference

2. Học liên tục (Continual learning): Architecture-based approach



PackNet [25]



DER [26]

3. Rehearsal-free continual learning

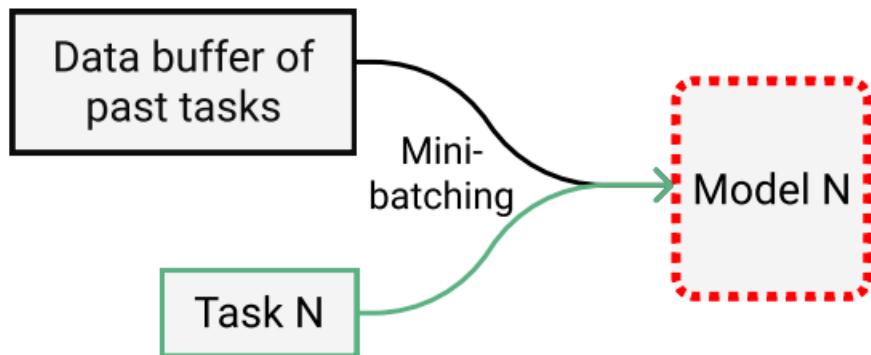
- Sử dụng memory buffer để lưu data của previous tasks gặp phải vấn đề data privacy => Động lực Rehearsal-free continual learning
- Cách tiếp cận dựa trên regularization/prior hoặc architecture có thể không yêu cầu sử dụng memory buffer.
 - Chỉ làm việc tốt với TIL
 - Thường vẫn yêu cầu memory buffer trong CIL
- Cách tiếp cận dựa trên generative model có thể không yêu cầu lưu raw data của task trước
 - Gặp vấn đề quên trong chính generative model
 - Yêu cầu lưu nhiều generative models

3. Rehearsal-free continual learning


- Hướng tới giải pháp bộ nhớ tạm (episodic memory) thông minh hơn, nhỏ hơn thay vì lưu data
- Tận dụng sức mạnh của các pretrained model (như ViT, Bert,...) và kỹ thuật prompt/prefix tuning trong NLP khi finetune các pretrained model
- Giải pháp:
 - Learning to prompt (L2P) [20] hướng tới xây dựng prompt pool chứa các prompts cho tất cả các task
 - Dualprompt [21] hướng tới học shared prompt cho tất cả các tasks và task-specific prompt
 - Coda-prompt [22] hướng tới học cơ chế attention để kết hợp các prompt
 - SPP (Steering prototypes with prompt-tuning) [23] và EPI (Efficient Parameter Isolation) [24] hướng tới loại bỏ shared parameters

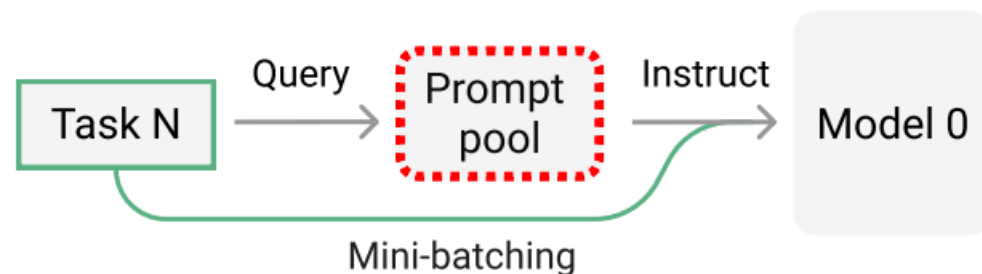
3. Rehearsal-free continual learning: L2P [20]

Rehearsal-based methods:
Fine-tuning



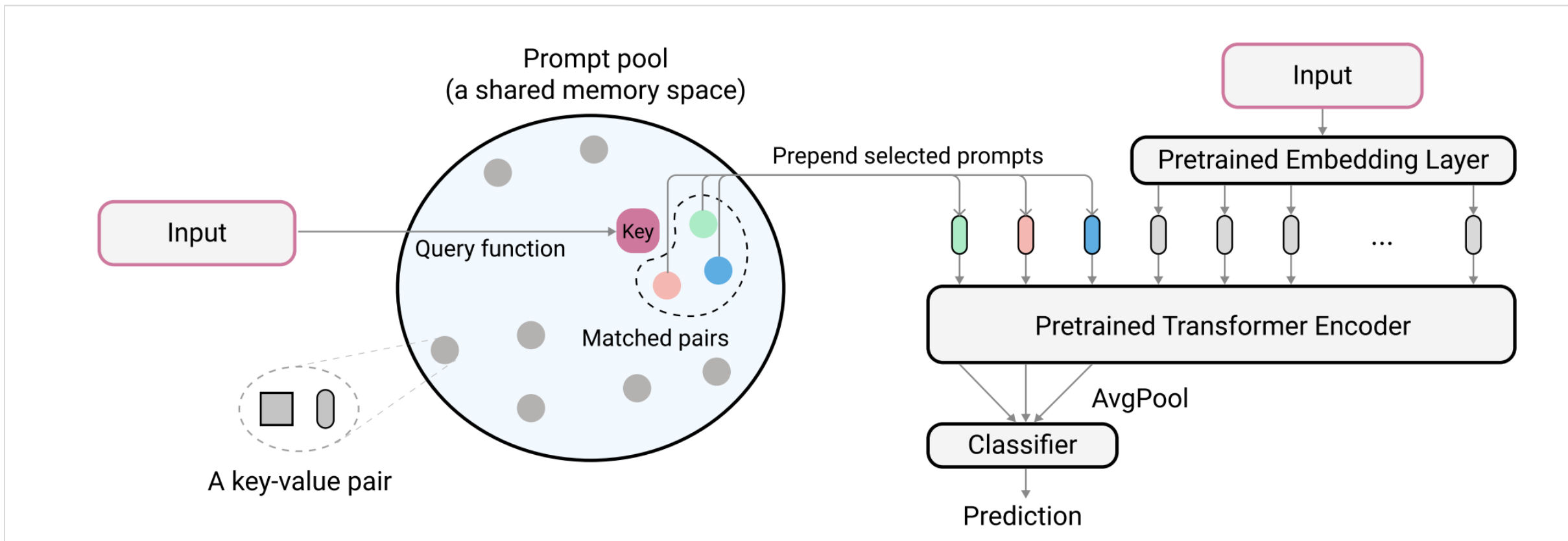
Our method:
Prompt selection + tuning

 : Trainable



L2P đóng băng pre-trained model và học prompt pool. Kiến thức cụ thể cho từng task được lưu trữ bên trong các prompts trong prompt pool. L2P tự động chọn và cập nhật các prompt từ prompt pool khi học task mới [20].

3. Rehearsal-free continual learning: L2P



Minh họa L2P [20]

3. Rehearsal-free continual learning: L2P

- Cho ảnh đầu vào $x \in R^{H \times W \times C}$ và pretrained vision transformer (ViT):

$$f = f_r \circ f_e,$$

$x_e = f_e(x)$ là một embedding, sau đó được đưa qua f_r (f_r thể hiện cho stack của các attention layers).

- Sử dụng kỹ thuật prompt tuning: Prompt $P_e \in R^{L \times D}$ được nối vào embedding x_e để tạo biểu diễn mới $x_p = [P_e; x_e]$, sau đó qua lớp attention thu được $f_r(x_p)$
- Prompt pool được định nghĩa như sau: $P = \{P_1, P_2, \dots, P_M\}$, trong đó
 - M = tổng số prompts trong prompt pool
 - $P_j \in R^{L_P \times D}$ là một prompt duy nhất với độ dài token L_P và cùng kích thước nhúng D như x_e

3. Rehearsal-free continual learning: L2P

- Mỗi data x sẽ lấy N prompts từ prompt pool. Ký hiệu $\{s_i\}_{i=1}^N$ là một tập con của N chỉ số từ $[1, M]$, đầu vào được biểu diễn như sau:

$$x_p = [P_{s_1}; P_{s_2}; P_{s_3}; \dots P_{s_N}; x_e]$$

trong đó ; biểu thị phép nối dọc theo chiều kích thước của token.

- Với mỗi sample x , làm sao lựa chọn N prompts từ prompt pool?
 - Dựa trên truy vấn query để tìm key và lấy value
 - Mỗi prompt là giá trị cho một khóa có thể học được: $\{(k_1, P_1), (k_2, P_2), \dots, (k_M, P_M)\}$, trong đó $k_i \in R^{D_k}$
 - Một hàm truy vấn $q: R^{H \times W \times C} \rightarrow R^{D_k}$ để mã hóa đầu vào x với cùng kích thước như khóa. Sử dụng pre-trained model: $q(x) = f(x)[0]$ (vector đặc trưng tương ứng với [class] token)

3. Rehearsal-free continual learning: L2P

- Với đầu vào x , sử dụng $q(x)$ để tìm top-N keys bằng hàm mục tiêu đơn giản sau:

$$\mathbf{K}_x = \underset{\{s_i\}_{i=1}^N \subseteq [1, M]}{\operatorname{argmin}} \sum_{i=1}^N \gamma(q(x), \mathbf{k}_{s_i})$$

trong đó K_x là tập gồm N keys phù hợp nhất với x từ prompt pool. γ là hàm khoảng cách (sử dụng cosine distance).

- Trong quá trình huấn luyện task t , khuyến khích cơ chế truy vấn chọn các prompt đa dạng trong prompt pool
- Sử dụng duy trì một bảng tần số prompt $H_t = [h_1, h_2, \dots, h_M]$, trong đó tần suất chuẩn hóa của prompt P_i được chọn cho đến nhiệm vụ $t - 1$.

$$\mathbf{K}_x = \underset{\{s_i\}_{i=1}^N \subseteq [1, M]}{\operatorname{argmin}} \sum_{i=1}^N \gamma(q(x), \mathbf{k}_{s_i}) \cdot h_{s_i}$$

3. Rehearsal-free continual learning: L2P

- Ở mỗi bước huấn luyện, sau khi chọn N prompts, tính biểu diễn

$$\mathbf{x}_p = [P_{s1}; P_{s2}; P_{s3}; \dots P_{sN}; \mathbf{x}_e]$$

- \mathbf{x}_p được đưa qua pre-trained model ViT f_r và bộ phân loại cuối cùng g_ϕ được tham số hóa bởi ϕ .
- Tối thiểu hóa hàm loss khi huấn luyện :

$$\min_{\mathbf{P}, \mathbf{K}, \phi} \mathcal{L}(g_\phi(f_r^{\text{avg}}(\mathbf{x}_p)), y) + \lambda \sum_{\mathbf{K}_x} \gamma(q(\mathbf{x}), \mathbf{k}_{s_i})$$

trong đó, f_r^{avg} là output mà các hidden vectors tương ứng tại N. L_p của prompt đã được lấy trung bình trước khi đưa vào đầu phân loại g_ϕ

3. Rehearsal-free continual learning: L2P

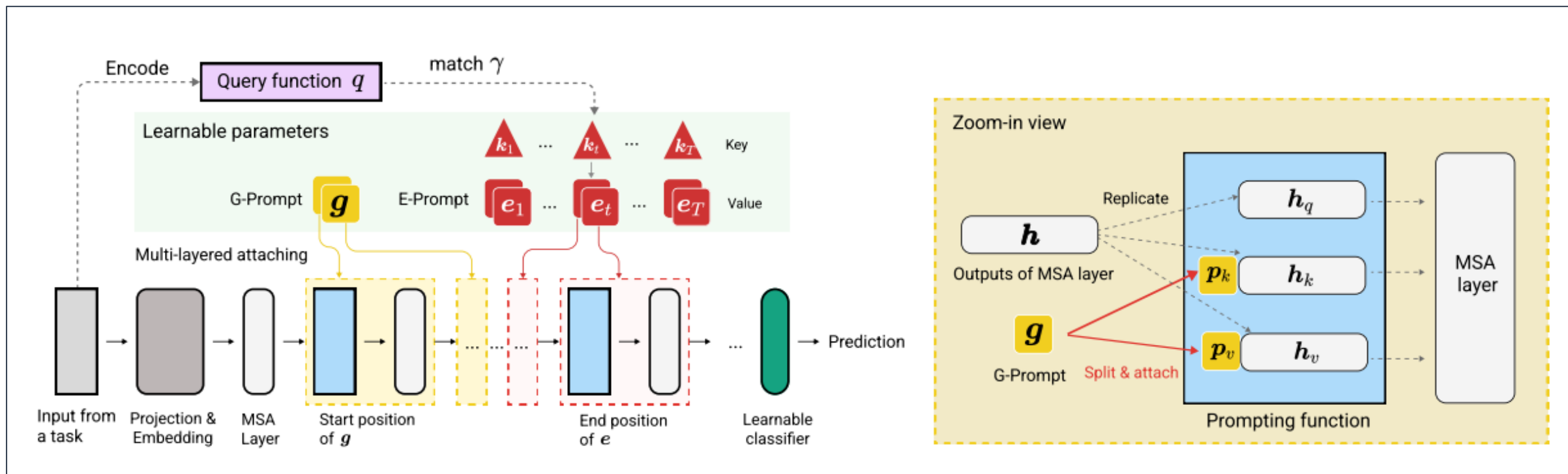
Kết quả thử nghiệm so sánh các phương pháp trong kịch bản CIL [20]

Method	Buffer size	Split CIFAR-100		Buffer size	5-datasets	
		Average Acc (\uparrow)	Forgetting (\downarrow)		Average Acc (\uparrow)	Forgetting (\downarrow)
FT-seq-frozen	0	17.72 \pm 0.34	59.09 \pm 0.25	0	39.49 \pm 0.12	42.62 \pm 0.20
FT-seq		33.61 \pm 0.85	86.87 \pm 0.20		20.12 \pm 0.42	94.63 \pm 0.68
EWC [21]		47.01 \pm 0.29	33.27 \pm 1.17		50.93 \pm 0.09	34.94 \pm 0.07
LwF [28]		60.69 \pm 0.63	27.77 \pm 2.17		47.91 \pm 0.33	38.01 \pm 0.28
L2P (ours)		83.83\pm0.04	7.63\pm0.30		81.14 \pm0.93	4.64 \pm0.52
ER [8]	10/class	67.87 \pm 0.57	33.33 \pm 1.28	5/class	80.32 \pm 0.55	15.69 \pm 0.89
GDumb [46]		67.14 \pm 0.37	-		56.99 \pm 0.06	-
BiC [61]		66.11 \pm 1.76	35.24 \pm 1.64		78.74 \pm 1.41	21.15 \pm 1.00
DER++ [3]		61.06 \pm 0.87	39.87 \pm 0.99		80.81 \pm 0.07	14.38 \pm 0.35
Co ² L [4]		72.15 \pm 1.32	28.55 \pm 1.56		82.25 \pm 1.17	17.52 \pm 1.35
L2P-R (ours)		84.21\pm0.53	7.72\pm0.77		85.56\pm0.95	4.22\pm0.03
ER [8]	50/class	82.53 \pm 0.17	16.46 \pm 0.25	10/class	84.26 \pm 0.84	12.85 \pm 0.62
GDumb [46]		81.67 \pm 0.02	-		70.76 \pm 0.12	-
BiC [61]		81.42 \pm 0.85	17.31 \pm 1.02		85.53 \pm 2.06	10.27 \pm 1.32
DER++ [3]		83.94 \pm 0.34	14.55 \pm 0.73		84.88 \pm 0.57	10.46 \pm 1.02
Co ² L [4]		82.49 \pm 0.89	17.48 \pm 1.80		86.05 \pm 1.03	12.28 \pm 1.44
L2P-R (ours)		86.31\pm0.59	5.83\pm0.61		88.95\pm0.78	4.92\pm0.71
Upper-bound	-	90.85 \pm 0.12	-	-	93.93 \pm 0.18	-

3. Rehearsal-free continual learning: Dualprompt [21]

- L2P sử dụng Prompt pool có thể học để mã hóa kiến thức một cách ngắn gọn hơn so với bộ đệm trong rehearsal-based approach.
- Hạn chế
 - Sử dụng nhiều shared prompts giữa các tasks
 - Kết quả còn thấp hơn với một số rehearsal-based methods
- DualPrompt là học:
 - G(eneral)-Prompt dùng chung cho tất cả các task
 - E(xpert)-Prompt (task-specific prompt) đặc trưng riêng cho mỗi task
- Phương pháp này tách rời không gian prompt ở mức cao hơn, giúp tiết kiệm bộ nhớ và tăng hiệu suất hơn so với các phương pháp tập trung vào không gian biểu diễn ẩn ở mức thấp hơn.

3. Rehearsal-free continual learning: Dualprompt



Trong khi G-prompt được dùng chung cho tất cả các task, E-prompt sẽ đặc trưng cho mỗi task. G-prompt và E-prompt sẽ được đặt ở các layers khác nhau theo prefix-tuning hoặc prompt-tuning [21].

3. Rehearsal-free continual learning: Dualprompt

- Một mạng pretrained ViT với N tầng MSA liên tiếp, ký hiệu đặc trưng nhúng đầu vào input của lớp MSA thứ i là $h^{(i)}$ với $i = 1, 2, \dots, N$.
- G-Prompt: $g \in \mathbb{R}^{L_g \times D}$ với độ dài prompt L_g và chiều nhúng D , là một tham số được chia sẻ cho tất cả các nhiệm vụ.
- Giả sử muốn đính kèm G-Prompt vào lớp MSA thứ i , G-Prompt biến đổi $h^{(i)}$ thông qua một hàm:

$$h_g^{(i)} = f_{\text{prompt}}(g, h^{(i)})$$

trong đó, f_{prompt} xác định cách đính kèm prompt vào những embedding ẩn.

3. Rehearsal-free continual learning: Dualprompt

- E-Prompt: $E = \{e_t\}_{t=1}^T$ là tập hợp các tham số cho task t tương ứng, trong đó $e_t \in \mathbb{R}^{L_e \times D}$ có chiều dài chuỗi L_e và chiều nhúng D giống với G-Prompt, và T là tổng số nhiệm vụ.
- Khác với G-Prompt được chia sẻ, mỗi e_t được liên kết với một key cụ thể $k_t \in \mathbb{R}^D$ của task t
- Đối với một ví dụ đầu vào từ task t , đính kèm E-Prompt vào lớp MSA thứ j :

$$h_e^{(j)} = f_{\text{prompt}}(e_t, h^{(j)})$$

trong đó, f_{prompt} xác định cách đính kèm prompt vào những embedding ẩn.

- Tương tự trong L2P, hàm loss này cho việc truy vấn E-prompt trực tiếp sử dụng toàn bộ mô hình pretrained làm hàm truy vấn: $q(x) = f(x)[0]$ (vector đặc trưng tương ứng với [class] token), và độ khoảng cách cosine γ

$$L_{\text{match}}(x, k_t) = \gamma(q(x), k_t), \quad x \in D_t$$

- k_t được xem là trung bình (prototype) của task t

3. Rehearsal-free continual learning: Dualprompt

WHERE: Cần sự linh hoạt khi đính kèm các prompt vào các vị trí phù hợp nhất một cách độc lập

- Đính kèm G-Prompt $g^{(l)}$ từ lớp $start_g$ đến lớp end_g
- Đính kèm E-Prompt $e_t^{(l)}$ từ lớp $start_e$ đến lớp end_e .

Với $(start_g, end_g)$ và $(start_e, end_e)$ có thể hoàn toàn khác nhau hoặc không giao nhau.

- Trong các thực nghiệm, sử dụng grid search thực nghiệm một tập hợp cụ thể của $start_g, end_g, start_e, end_e$ trên tập validation

HOW: Việc sử dụng prompt có thể được coi như việc sửa đổi đầu vào của các lớp MSA.

Đầu vào của lớp MSA là $h \in \mathbb{R}^{L \times D}$, và chúng ta tiếp tục ký hiệu đầu vào cho query, key, value của lớp MSA lần lượt là h_Q, h_K, h_V .

- Prompt Tuning (Pro-T): $f_{\text{prompt}}^{\text{Pro-T}}(\mathbf{p}, \mathbf{h}) = \text{MSA}([\mathbf{p}; \mathbf{h}_Q], [\mathbf{p}; \mathbf{h}_K], [\mathbf{p}; \mathbf{h}_V]),$

- Prefix Tuning (Pre-T): $f_{\text{prompt}}^{\text{Pre-T}}(\mathbf{p}, \mathbf{h}) = \text{MSA}(\mathbf{h}_Q, [\mathbf{p}_k; \mathbf{h}_K], [\mathbf{p}_v; \mathbf{h}_V]).$

3. Rehearsal-free continual learning: Dualprompt

- Ký hiệu:
 - Kiến trúc với các hướng dẫn được đính kèm là f_{g,e_t}
 - f_ϕ tầng phân loại được tham số hóa bởi ϕ
- Tối thiểu hóa hàm mất mát huấn luyện:

$$\min_{g, e_t, k_t, \phi} \mathcal{L}(f_\phi(f_{g, e_t}(\mathbf{x})), y) + \lambda \mathcal{L}_{\text{match}}(\mathbf{x}, \mathbf{k}_t), \quad \mathbf{x} \in \mathcal{D}_t,$$

Trong đó, L là hàm mất mát cross-entropy, L_{match} là loss liên quan đến chọn prompt phù hợp, và λ là một hệ số cân bằng scalar.

3. Rehearsal-free continual learning: Dualprompt

Kết quả thử nghiệm so sánh các phương pháp trong kịch bản CIL [21]

Method	Buffer size	Split CIFAR-100		Buffer size	Split ImageNet-R	
		Avg. Acc (↑)	Forgetting (↓)		Avg. Acc (↑)	Forgetting (↓)
ER [8]	1000	67.87±0.57	33.33±1.28	1000	55.13±1.29	35.38±0.52
BiC [62]		66.11±1.76	35.24±1.64		52.14±1.08	36.70±1.05
GDumb [47]		67.14±0.37	-		38.32±0.55	-
DER++ [4]		61.06±0.87	39.87±0.99		55.47±1.31	34.64±1.50
Co ² L [5]		72.15±1.32	28.55±1.56		53.45±1.55	37.30±1.81
ER [8]	5000	82.53±0.17	16.46±0.25	5000	65.18±0.40	23.31±0.89
BiC [62]		81.42±0.85	17.31±1.02		64.63±1.27	22.25±1.73
GDumb [47]		81.67±0.02	-		65.90±0.28	-
DER++ [4]		83.94±0.34	14.55±0.73		66.73±0.87	20.67±1.24
Co ² L [5]		82.49±0.89	17.48±1.80		65.90±0.14	23.36±0.71
FT-seq	0	33.61±0.85	86.87±0.20	0	28.87±1.36	63.80±1.50
EWC [20]		47.01±0.29	33.27±1.17		35.00±0.43	56.16±0.88
LwF [28]		60.69±0.63	27.77±2.17		38.54±1.23	52.37±0.64
L2P [60]		83.86±0.28	7.35±0.38		61.57±0.66	9.73±0.47
DualPrompt		86.51±0.33	5.16±0.09		68.13±0.49	4.68±0.20
Upper-bound	-	90.85±0.12	-	-	79.13±0.18	-

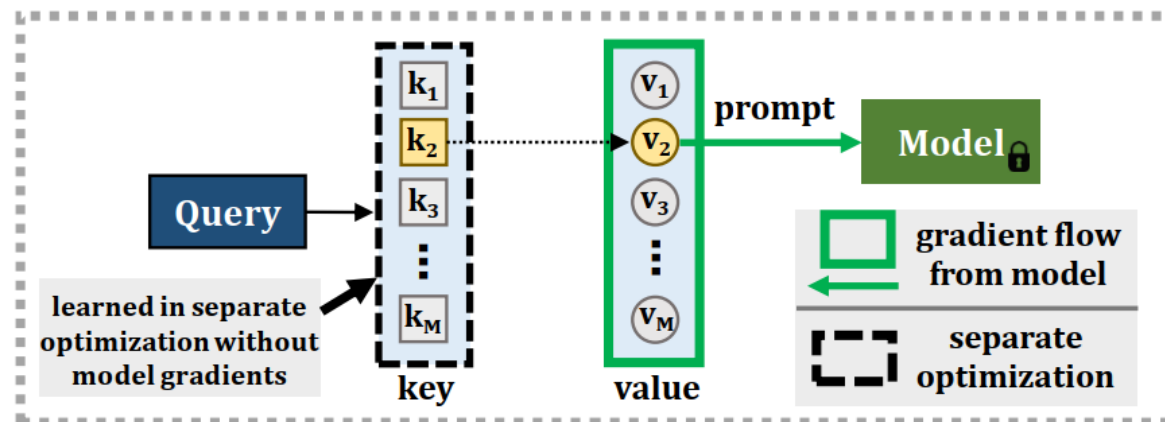
3. Rehearsal-free continual learning: Coda-prompt [22]

- Tạo ra một tập hợp các prompts và kết hợp chúng dựa trên một vector trọng số α dựa trên query $q(x)$, thay vì một chỉ số prompts lấy ra từ prompt pool.

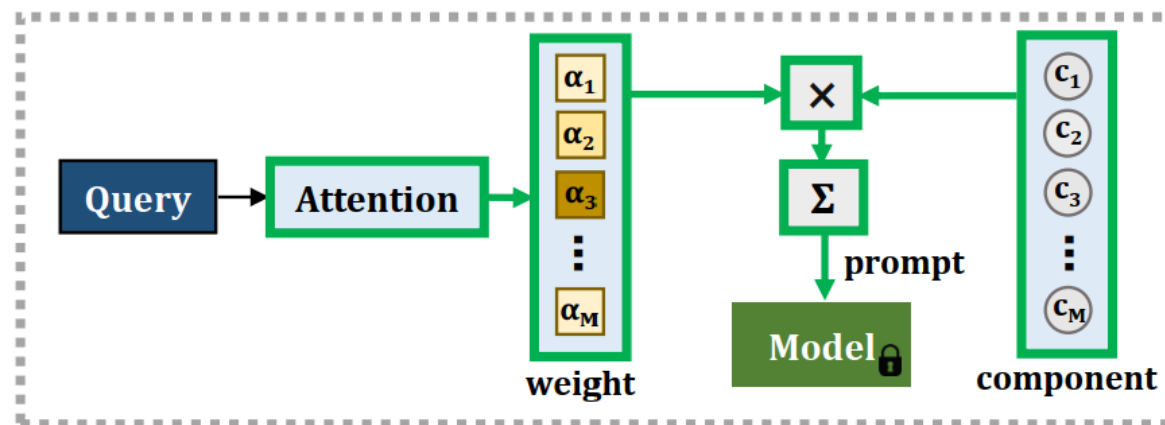
$$p = \sum_m \alpha_m P_m$$

Trong đó, $P \in R^{L_p \times D \times M}$ là tập hợp các prompts, M là số prompts (P gồm M prompt thành phần, mỗi thành phần là $P_m \in R^{L_p \times D}$) và α là vector trọng số thể hiện sự đóng góp của từng prompt thành phần.

- Đóng băng các prompt cũ và mở rộng thêm các prompt mới khi học task mới



Previous work



Coda-prompt

3. Rehearsal-free continual learning: Coda-prompt

- Độ tương đồng cosine giữa một query và các key để tạo ra vector trọng số:

$$\begin{aligned}\alpha &= \gamma(q(\mathbf{x}), \mathbf{K}) \\ &= \{\gamma(q(\mathbf{x}), \mathbf{K}_1), \gamma(q(\mathbf{x}), \mathbf{K}_2), \dots, \gamma(q(\mathbf{x}), \mathbf{K}_M)\}\end{aligned}$$

trong đó, $K \in R^{D \times M}$ là các khóa tương ứng đối với tập các prompt thành phần.

- Việc khớp giữa query và key có thể được coi như việc gom cụm trong một không gian chiều D, đó là một vấn đề đã biết và khó.
- Thêm thành phần khác vào việc khớp giữa key và query:
 - Mỗi P_m có một vector chú ý tương ứng A_m cùng với key $K_m \Rightarrow$ Hướng tới truy vấn tập trung vào các đặc điểm cụ thể từ truy vấn $q(x)$ có số chiều cao.

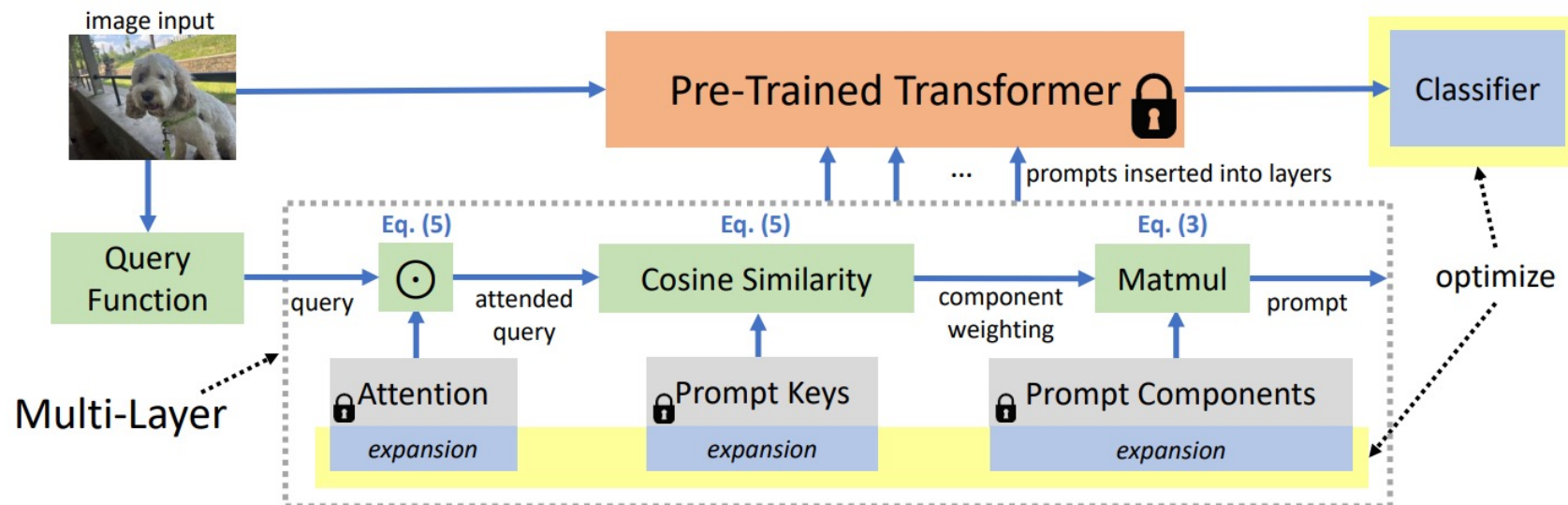
$$\begin{aligned}\alpha &= \gamma(q(\mathbf{x}) \odot \mathbf{A}, \mathbf{K}) \\ &= \{\gamma(q(\mathbf{x}) \odot \mathbf{A}_1, \mathbf{K}_1), \dots, \gamma(q(\mathbf{x}) \odot \mathbf{A}_M, \mathbf{K}_M)\}\end{aligned}$$

trong đó, $A \in R^{D \times M}$ chứa các tham số có thể học được (attention vector)

3. Rehearsal-free continual learning: Coda-prompt

- Khi chuyển đến một nhiệm vụ mới, CODA đóng băng các thành phần hiện tại của mình và mở rộng tập hợp, chỉ cập nhật các thành phần mới

=> giảm thiểu hiện tượng quên đột ngột là tránh việc ghi đè lên kiến thức đã học được từ các nhiệm vụ trước đó.



Từ ảnh đầu vào, coda-prompt sử dụng cơ chế attention để tổng hợp các prompts lại và kết hợp vào các layers

3. Rehearsal-free continual learning: Coda-prompt

- Toàn bộ hàm mất mát huấn luyện :

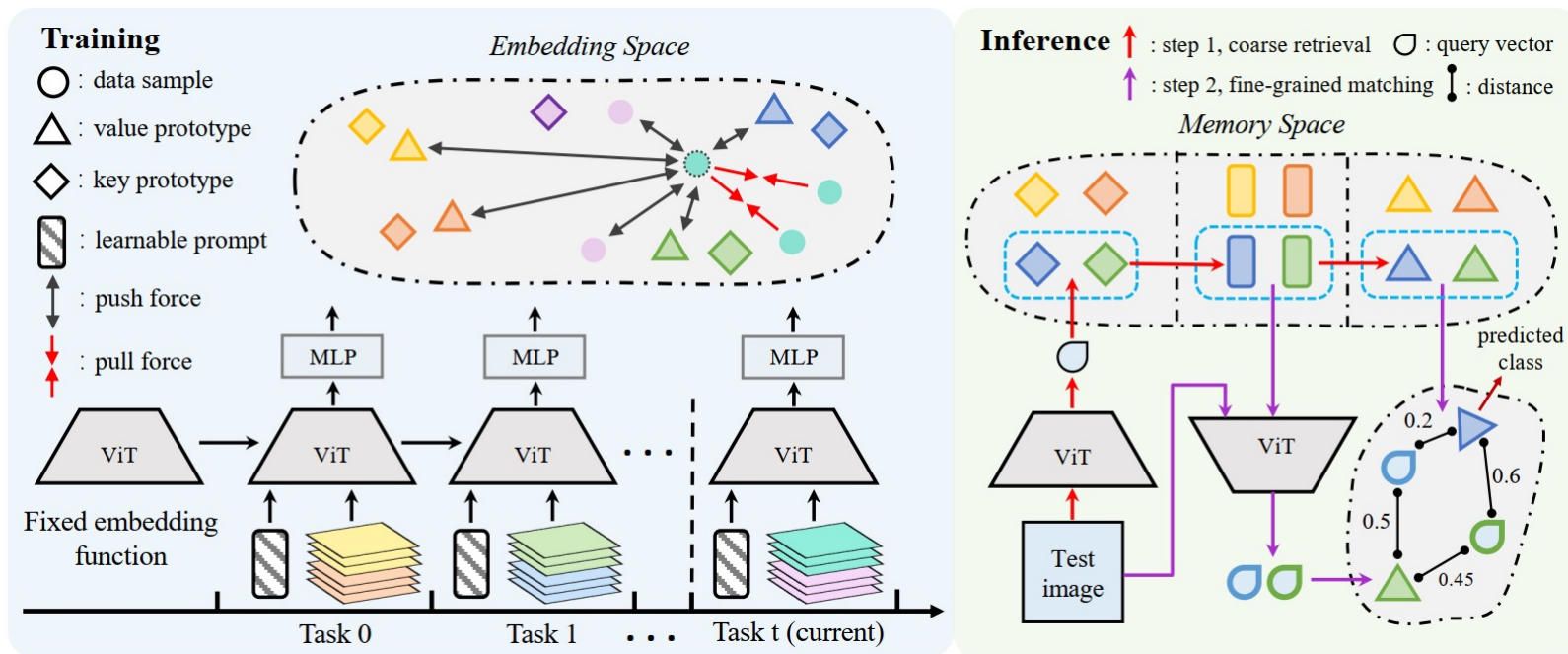
$$\min_{\mathbf{P}^n, \mathbf{K}^n, \mathbf{A}^n, \phi_n} \mathcal{L}(f_\phi(f_{\theta, \mathbf{P}, \mathbf{K}, \mathbf{A}}(\mathbf{x})), y) + \lambda(\mathcal{L}_{ortho}(\mathbf{P}) + \mathcal{L}_{ortho}(\mathbf{K}) + \mathcal{L}_{ortho}(\mathbf{A}))$$

trong đó, $\mathbf{P}^n, \mathbf{K}^n, \mathbf{A}^n$ đề cập đến các thành phần gợi ý và các khóa, vector tập trung tương ứng của chúng, được giữ nguyên và huấn luyện trong quá trình nhiệm vụ thứ n (task- n), và λ là một siêu tham số cân bằng sự mất mát về tính trực giao.

- Ràng buộc về tính trực giao cho \mathbf{P} , \mathbf{K} và \mathbf{A} giúp giảm gây xung đột lẫn nhau giữa kiến thức task cũ và mới

3. Rehearsal-free continual learning: SPP [23]

- Kết hợp ý tưởng architecture-based approach: Chỉ sử dụng task-specific prompt và không sử dụng shared prompts



- Lưu prototype (class mean) của các lớp cũ và sử dụng contrastive learning để phân tách biểu diễn của các class mới với class trước trong khi học task-specific prompt
- Xác định task id khi test dựa trên khoảng giữa biểu diễn của data với prototypes

3. Rehearsal-free continual learning: Một số vấn đề chính

- Rehearsal-free approaches vẫn gặp phải các vấn đề cơ bản trong CIL
 - Thiếu cơ chế tránh quên khi shared parameters giữa các tasks trong L2P [20], Dualprompt [21], coda-prompt [23]
- Các kỹ thuật xác định task ID đơn giản
 - Dựa trên các kỹ thuật học không giám sát (unsupervised mechanism) như trong L2P
 - Dựa trên prototype của task (task mean) như trong Dualprompt
 - Dựa trên khoảng cách gần nhất tới prototype của một lớp trong task đó như trong SPP [24]

4. References

- [1] Zhiyuan Chen, Bing Liu, Ronald Brachman, Peter Stone, and Francesca Rossi. 2018. Lifelong Machine Learning (2nd. ed.). Morgan & Claypool Publishers.
- [2] Toldo, M., Maracani, A., Michieli, U., & Zanuttigh, P. (2020). Unsupervised domain adaptation in semantic segmentation: a review. *Technologies*, 8(2), 35.
- [3] Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning* (pp. 1126-1135). PMLR.
- [4] Benavides Prado, D. (2020, July). Beyond catastrophic forgetting in continual learning: An attempt with svm. The International Conference on Machine Learning (ICML).
- [5] Nguyen, T., Mai, T., Nguyen, N., Van, L. N., & Than, K. (2022). Balancing stability and plasticity when learning topic models from short and noisy text streams. *Neurocomputing*, 505, 30-43.
- [6] Korycki, Ł., & Krawczyk, B. (2021). Streaming decision trees for lifelong learning. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21* (pp. 502-518). Springer International Publishing.
- [7] McCloskey, Michael ; Cohen, Neal J. / Catastrophic Interference in Connectionist Networks : The Sequential Learning Problem. In: Psychology of Learning and Motivation - Advances in Research and Theory. 1989 ; Vol. 24, No. C. pp. 109-165.
- [8] Abraham WC, Robins A. Memory retention--the synaptic stability versus plasticity dilemma. *Trends Neurosci.* 2005;28(2):73-78. doi:10.1016/j.tins.2004.12.003

4. References

- [9] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. arXiv preprint arXiv:1904.07734, 2019.
- [10] Zeno, C., Golan, I., Hoffer, E., & Soudry, D. (2021). Task-agnostic continual learning using online variational bayes with fixed-point updates. *Neural Computation*, 33(11), 3139-3177.
- [11] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 3521-3526.
- [12] Zenke, F., Poole, B., & Ganguli, S. (2017, July). Continual learning through synaptic intelligence. In *International conference on machine learning* (pp. 3987-3995). PMLR.
- [13] Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., & Tuytelaars, T. (2018). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 139-154).
- [14] Loo, N., Swaroop, S., & Turner, R. E. (2020). Generalized variational continual learning. *arXiv preprint arXiv:2011.12328*.
- [15] W. Hu, Z. Lin, B. Liu, C. Tao, Z. Tao, J. Ma, D. Zhao, and R. Yan, "Overcoming catastrophic forgetting for continual learning via model adaptation," in *International Conference on Learning Representations*, 2018
- [16] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems*, pp. 2990-2999, 2017.

4. References

- [17] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in neural information processing systems*, pp. 6467-6476, 2017.
- [18] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *International Conference on Learning Representation*, 2019
- [19] W. Hu, Z. Lin, B. Liu, C. Tao, Z. Tao, J. Ma, D. Zhao, and R. Yan, "Overcoming catastrophic forgetting for continual learning via model adaptation," in *International Conference on Learning Representations*, 2018
- [20] 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).
- [21] Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C. Y., ... & Pfister, T. (2022, October). Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision* (pp. 631-648). Cham: Springer Nature Switzerland.
- [22] Smith, J. S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., ... & Kira, Z. (2023). CODA-Prompt: COntinual DEcomposed Attention-based Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11909-11919).
- [23] Li, Z., Zhao, L., Zhang, Z., Zhang, H., Liu, D., Liu, T., & Metaxas, D. N. (2023). Steering Prototype with Prompt-tuning for Rehearsal-free Continual Learning. *arXiv preprint arXiv:2303.09447*.
- [24] Wang, Z., Liu, Y., Ji, T., Wang, X., Wu, Y., Jiang, C., ... & Zeng, W. (2023, July). Rehearsal-free Continual Language Learning via Efficient Parameter Isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 10933-10946).

References

- [25] Mallya, A., & Lazebnik, S. (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 7765-7773).