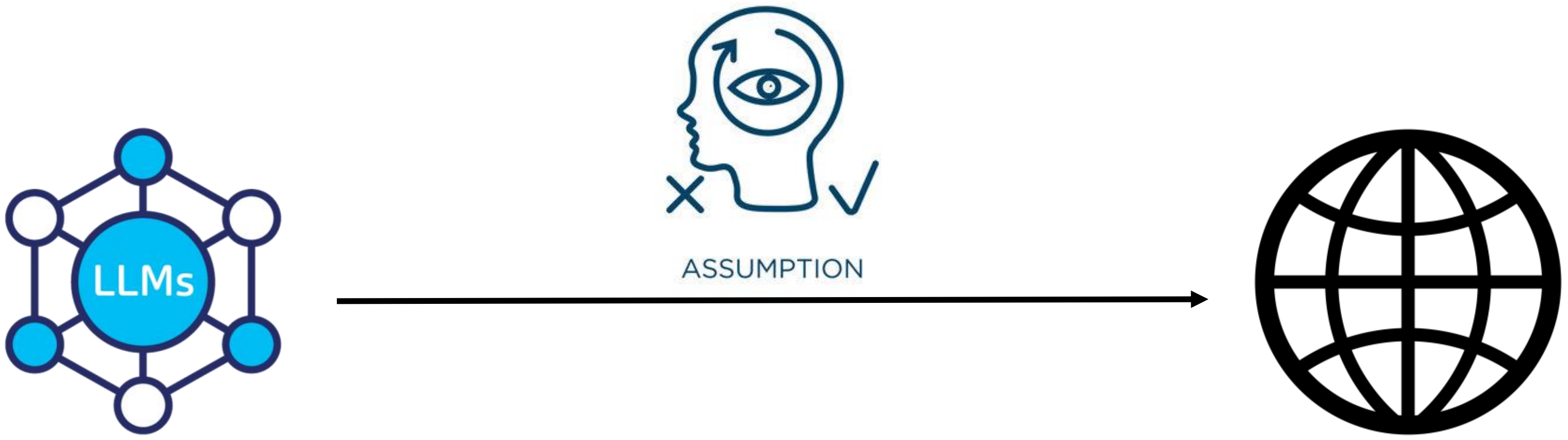
Enhancing Large Language Models for a Dynamic World

Presenter: Zixuan Ke
https://vincent950129.github.io/

LLM

Packed with
knowledge and excels
in many tasks

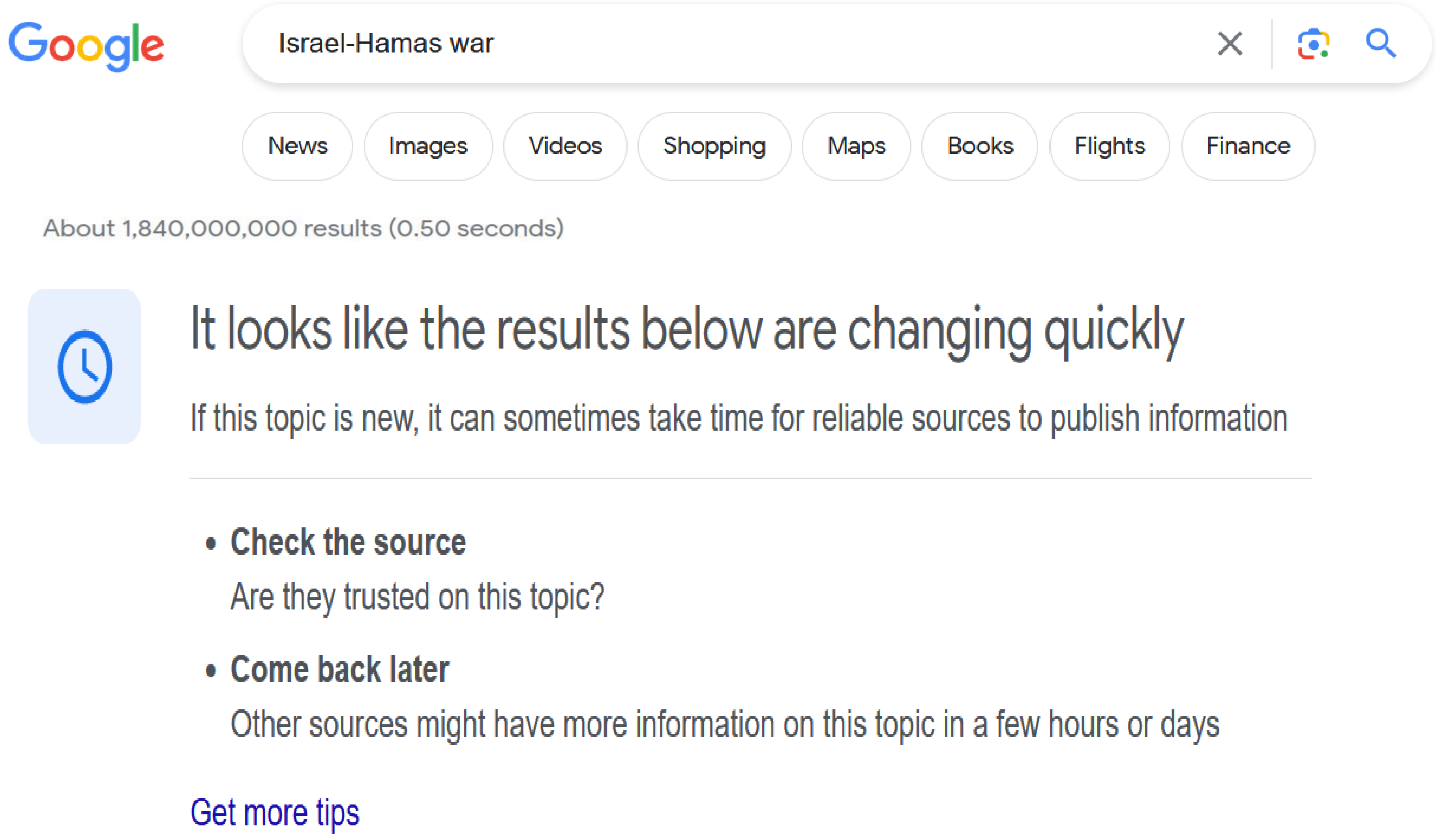
LLM in A Fixed World



Packed with knowledge and excels in many tasks

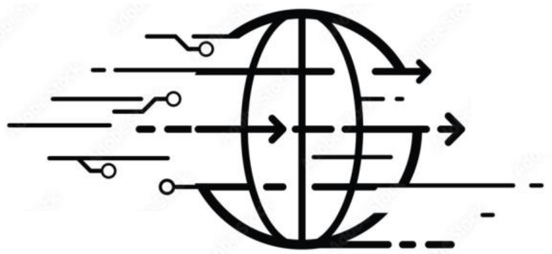
The world is **fixed** (i.i.d)

The World Changes Quickly



LLM in A Dynamic World

Packed with knowledge and excels in many tasks

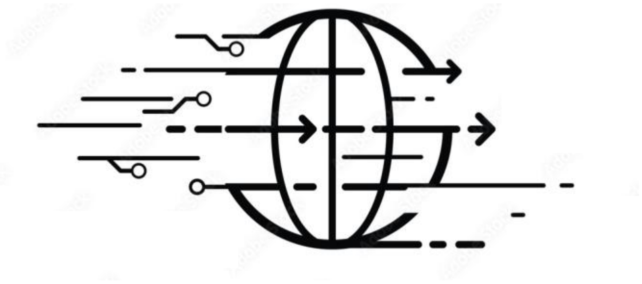


The world is **ever-changing**

LLM in A Dynamic World

How to make knowledge in LLM more **reusable** and **updatable** in the **dynamic** world?

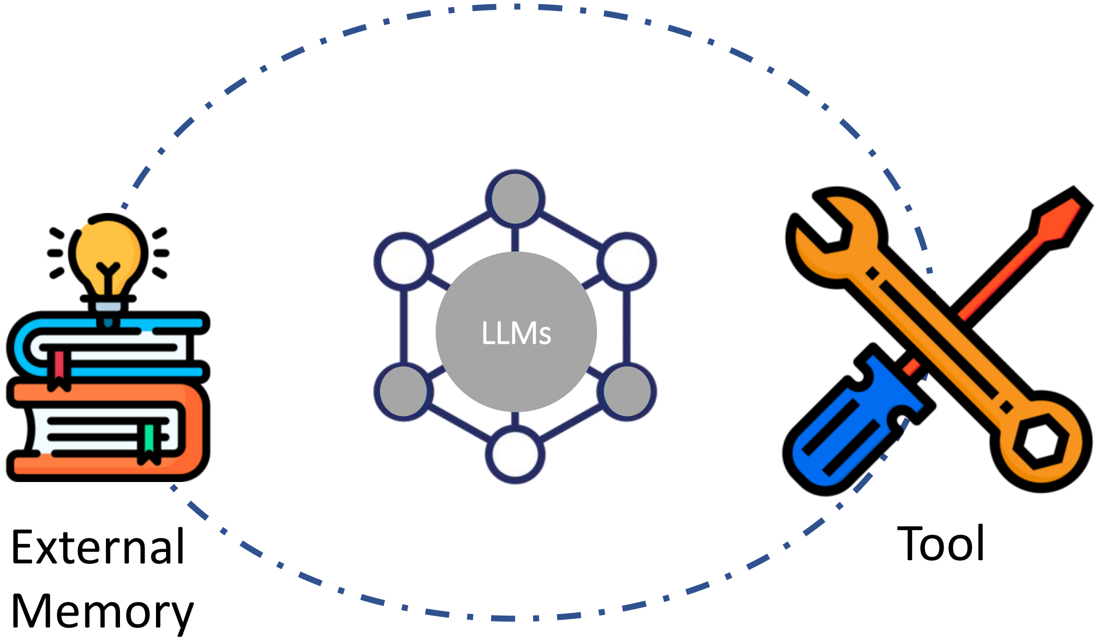
Packed with knowledge and excels in many tasks



The world is **ever-changed**

LLM in A Dynamic World

How to make knowledge in LLM more **reusable** and **updatable** in the **dynamic** world?

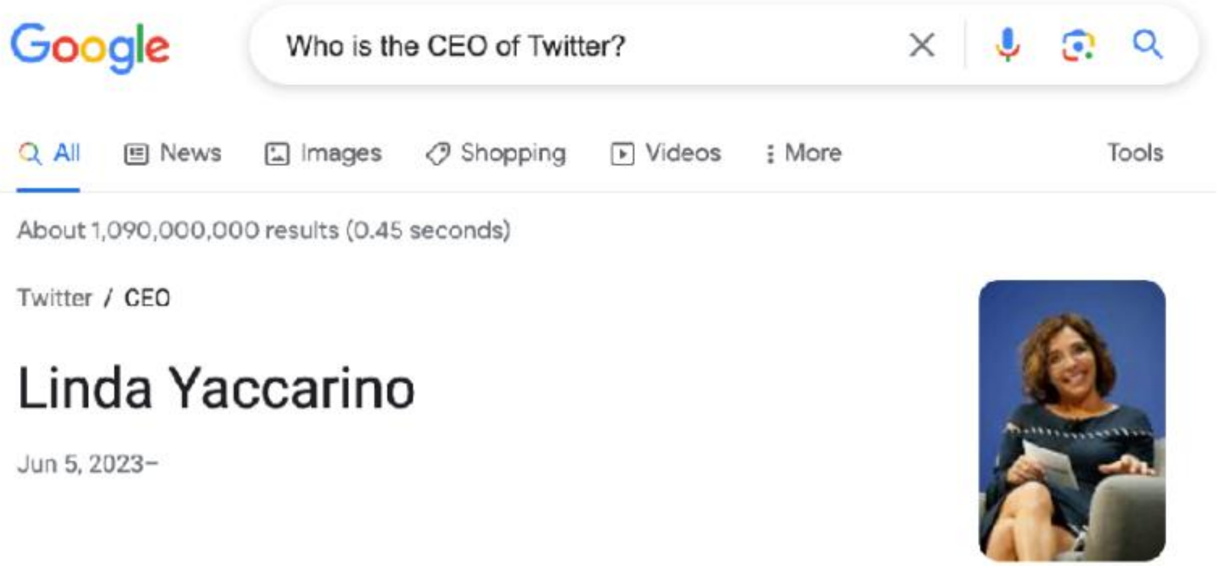


Retrieval-augmented LLM

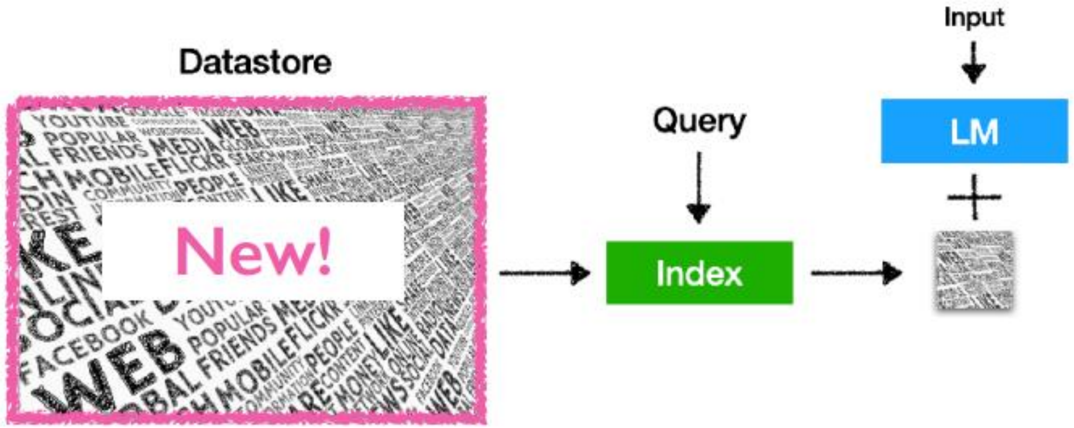
Who is the CEO of Twitter?

ChatGPT

As of my **knowledge cutoff in September 2021**, the CEO of Twitter is **Jack Dorsey**….



- The datastore can be easily **updated** and **expanded** - even without retraining!



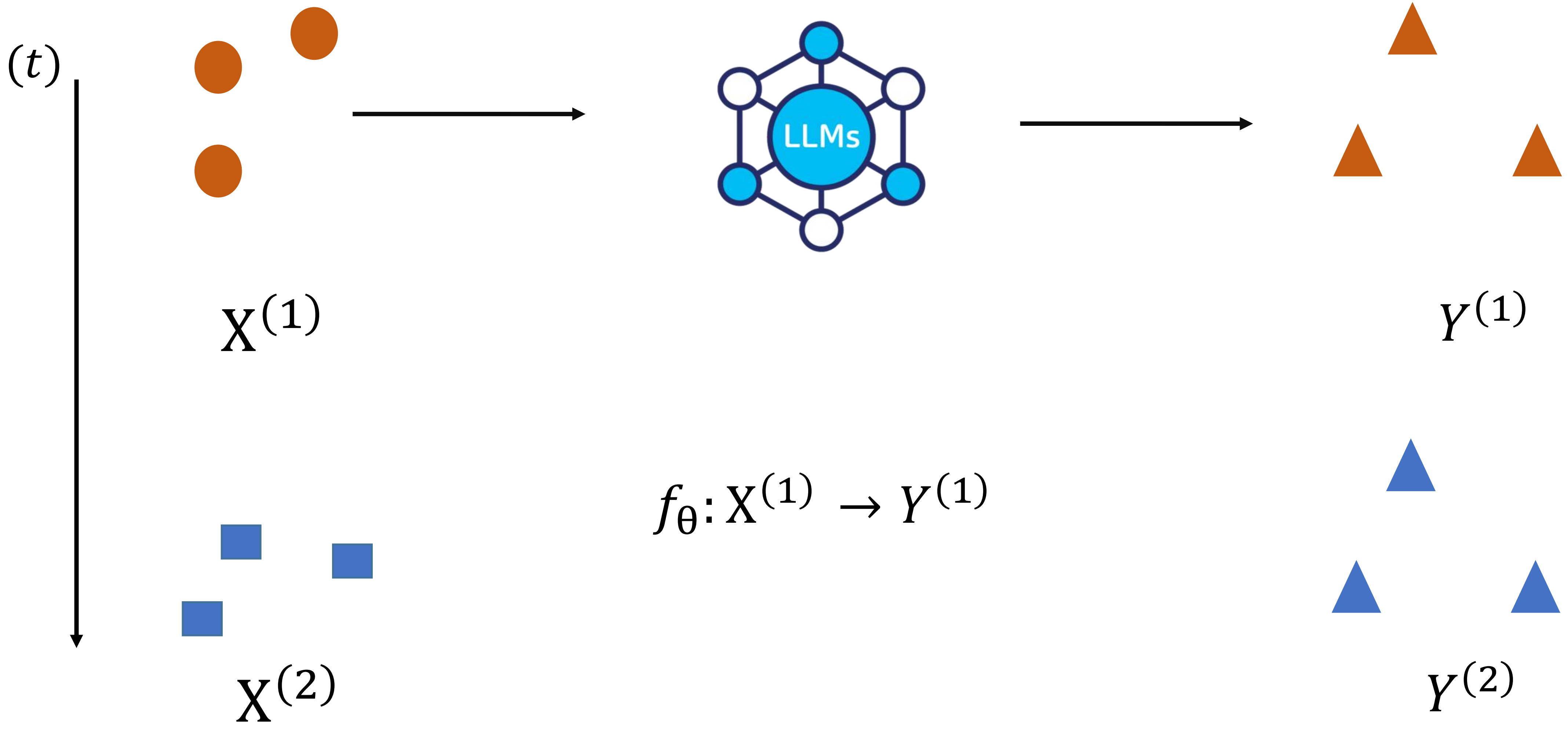
LLM in A Dynamic World

How to make knowledge in LLM more **reusable** and **updatable** in the **dynamic** world?

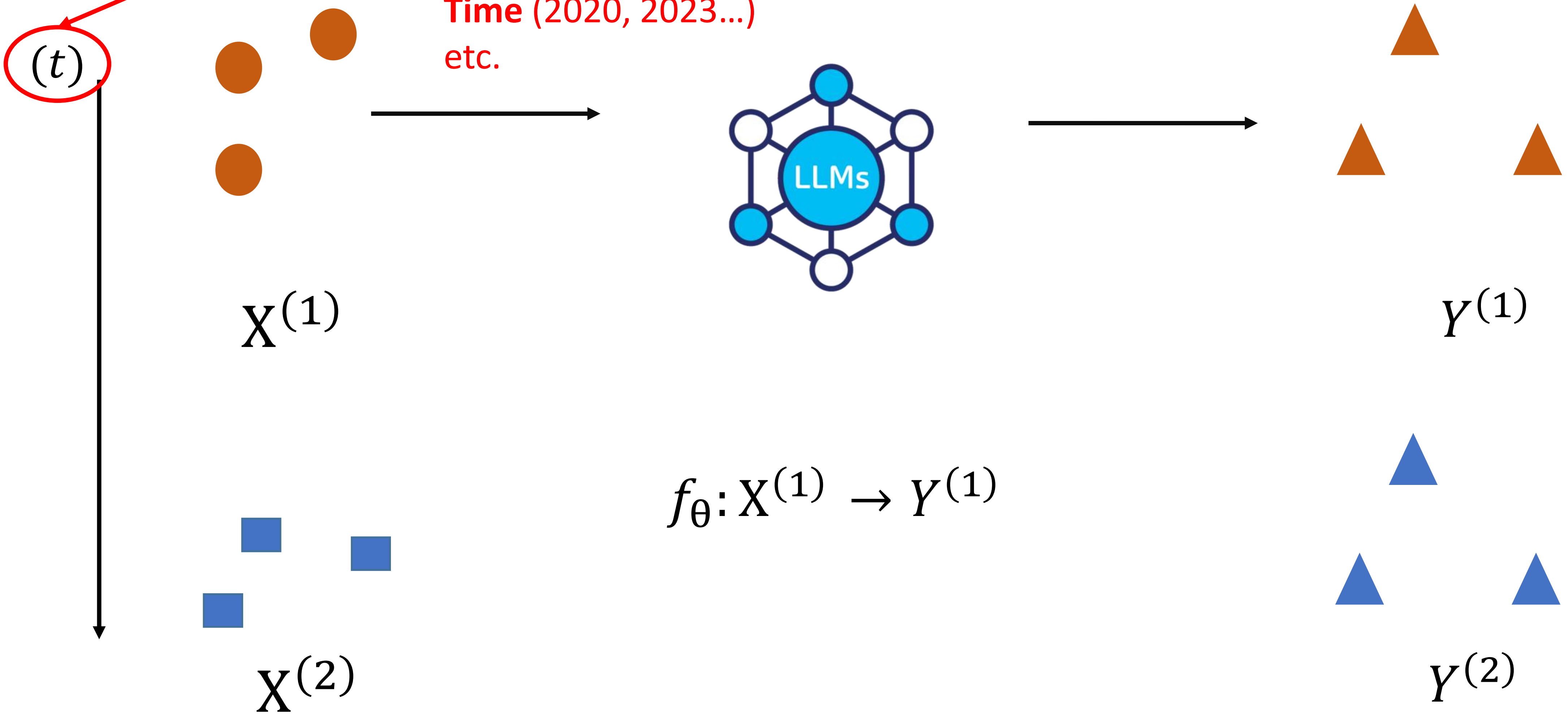
Continual Learning

What will happen if we **update the LLM** in a **changing world**?

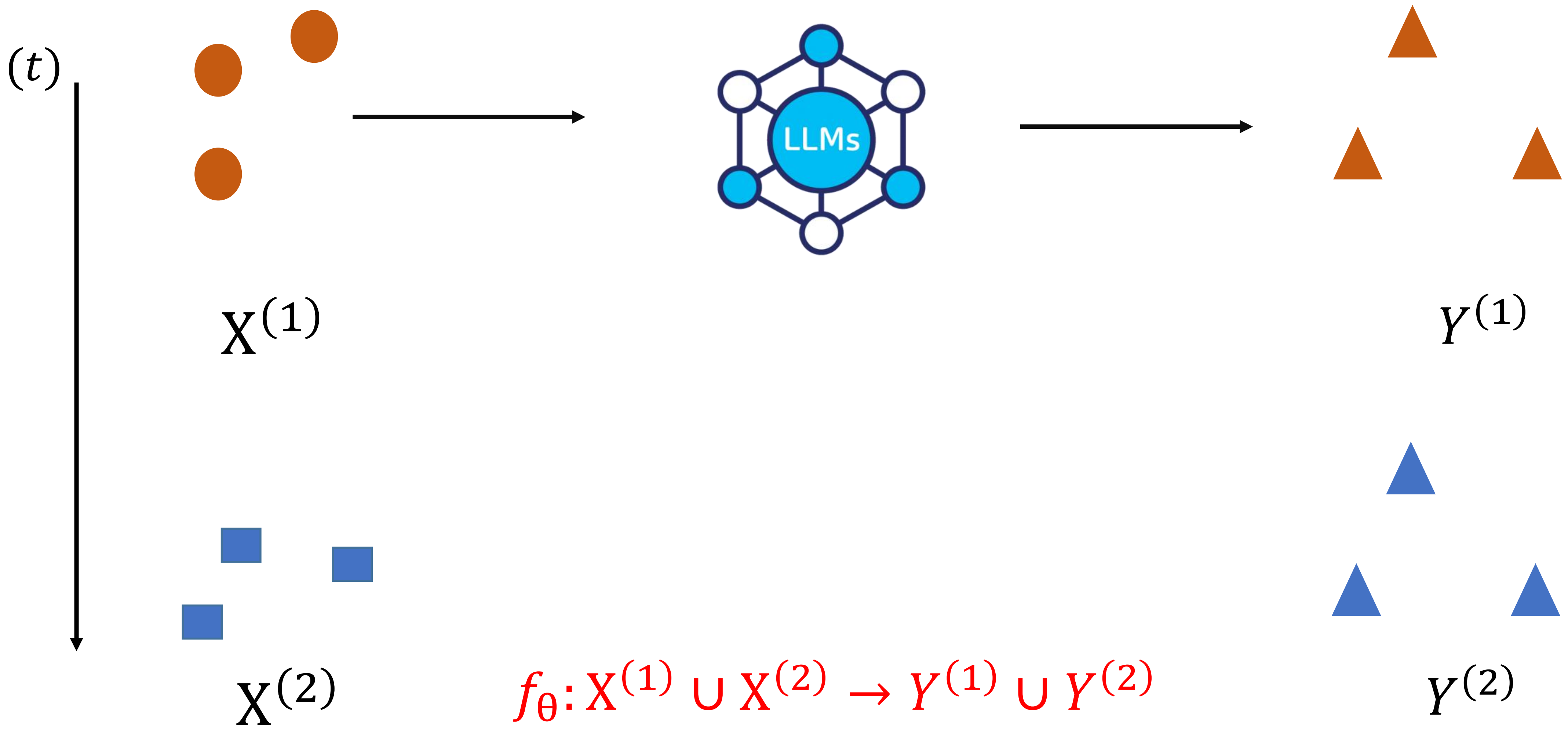
X $\quad f_{\theta}: X \to Y \quad Y$


(t)
LLMs
X(1)
Y(1)
f_θ: X(1) → Y(1)
X(2)
Y(2)

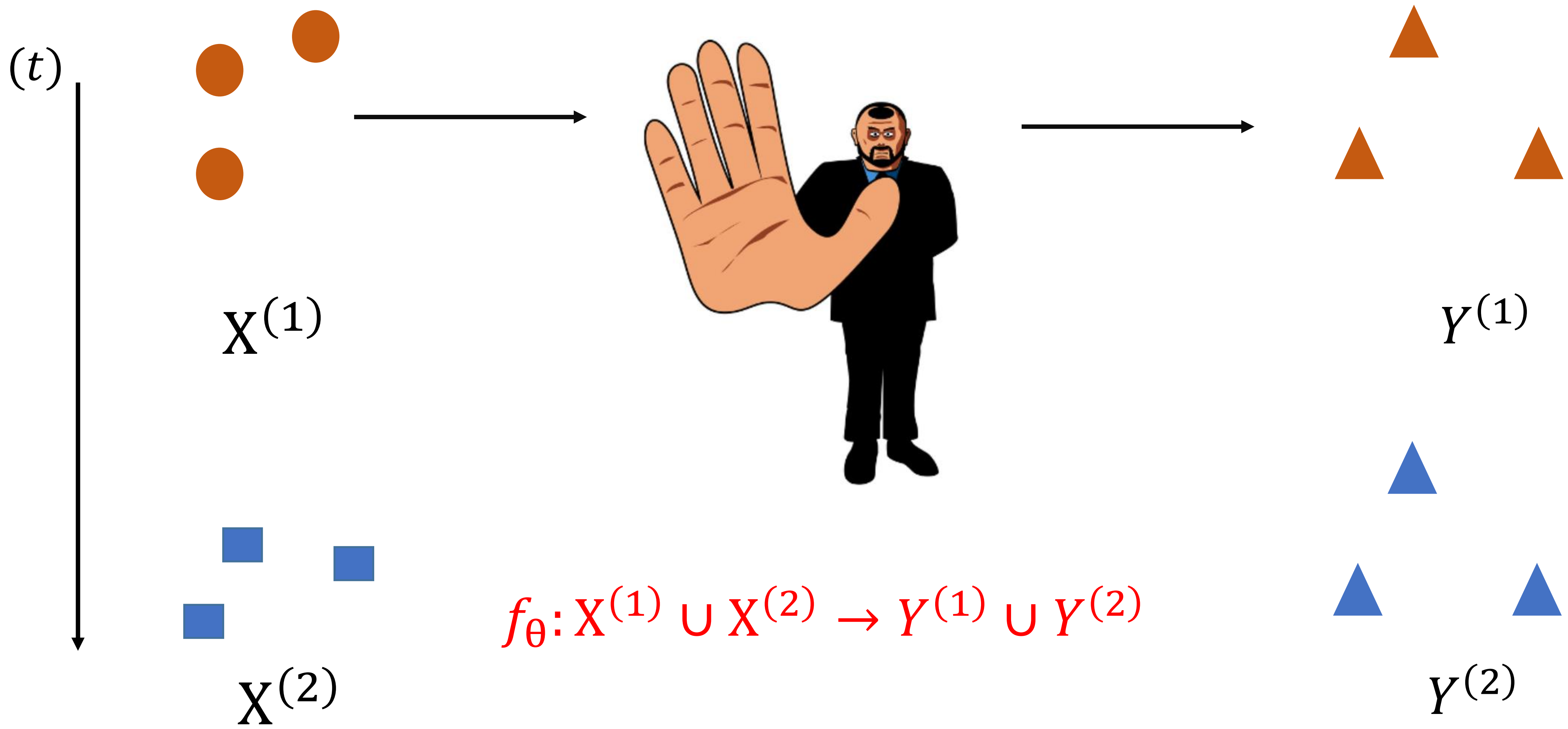
Another dimension, can be
Task (sentiment classification, news classification…)
Domain (restaurant, phone, camera…)
Class (dog, cat, car, horse…)
Time (2020, 2023…)
etc.



The dimension *t* can be called "task" in continual learning



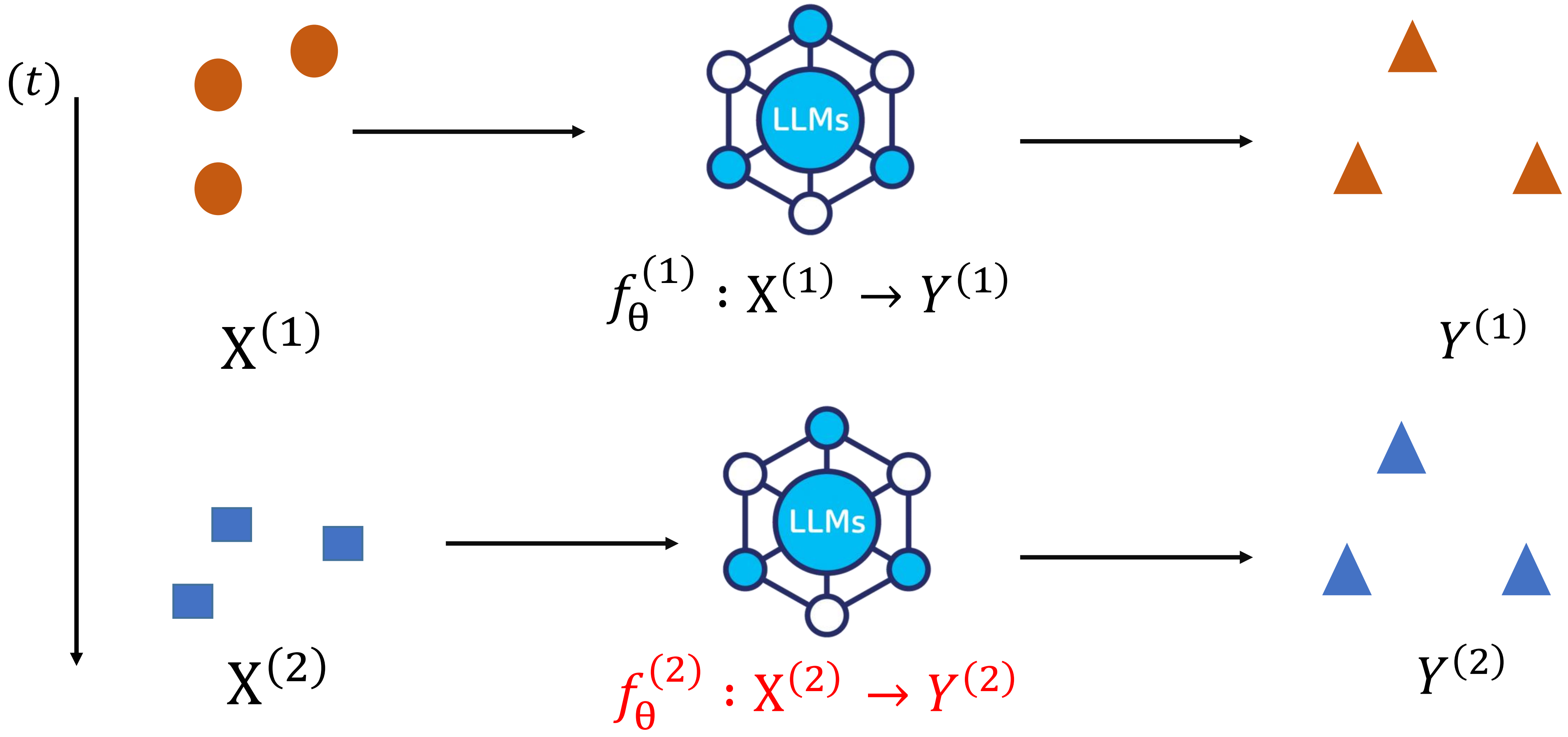
$$f_\theta : X^{(1)} \cup X^{(2)} \rightarrow Y^{(1)} \cup Y^{(2)}$$



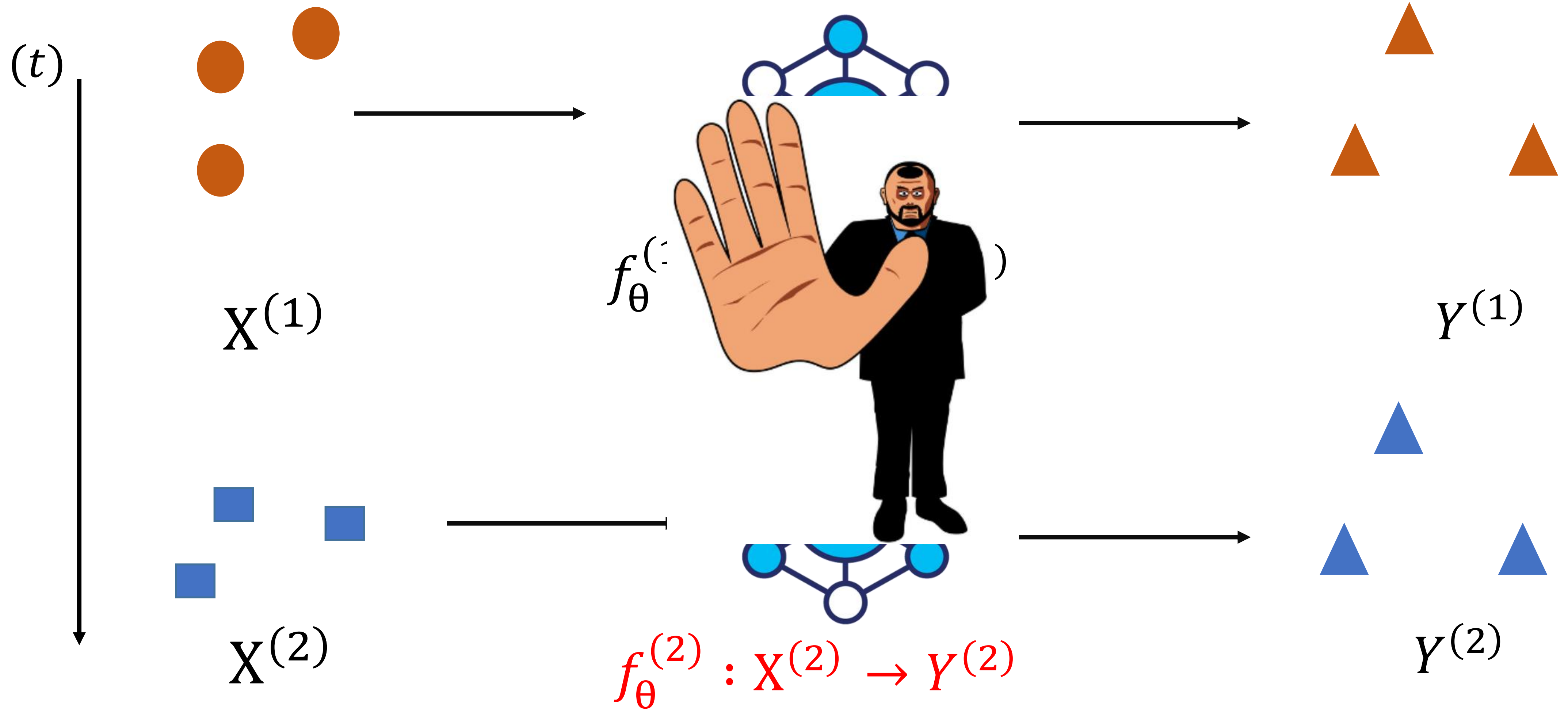
$$f_\theta : X^{(1)} \cup X^{(2)} \rightarrow Y^{(1)} \cup Y^{(2)}$$

Cons

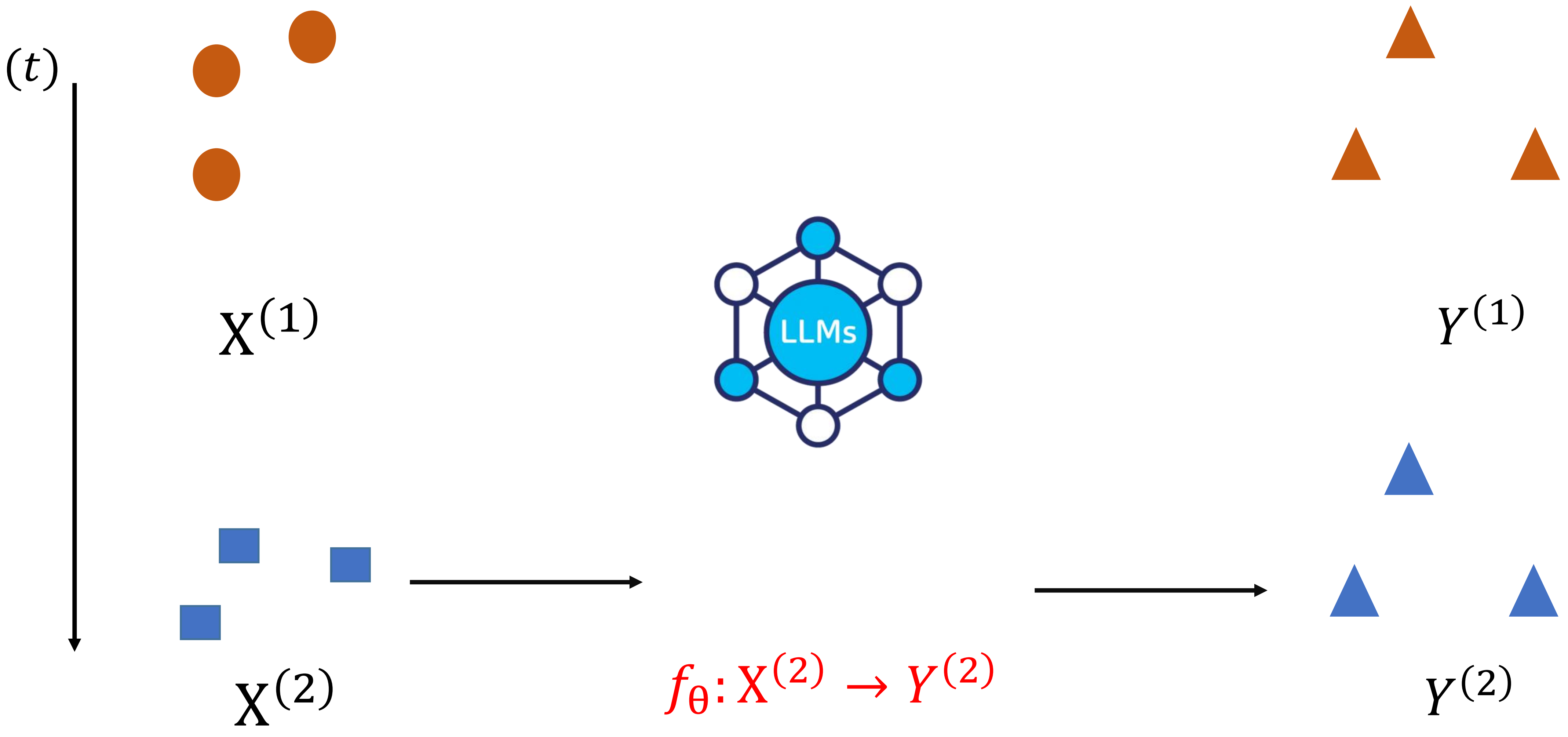
- Train from scratch whenever update is needed
- Save all the past data
- Privacy concern (user may not want to share their personal data)
- etc.



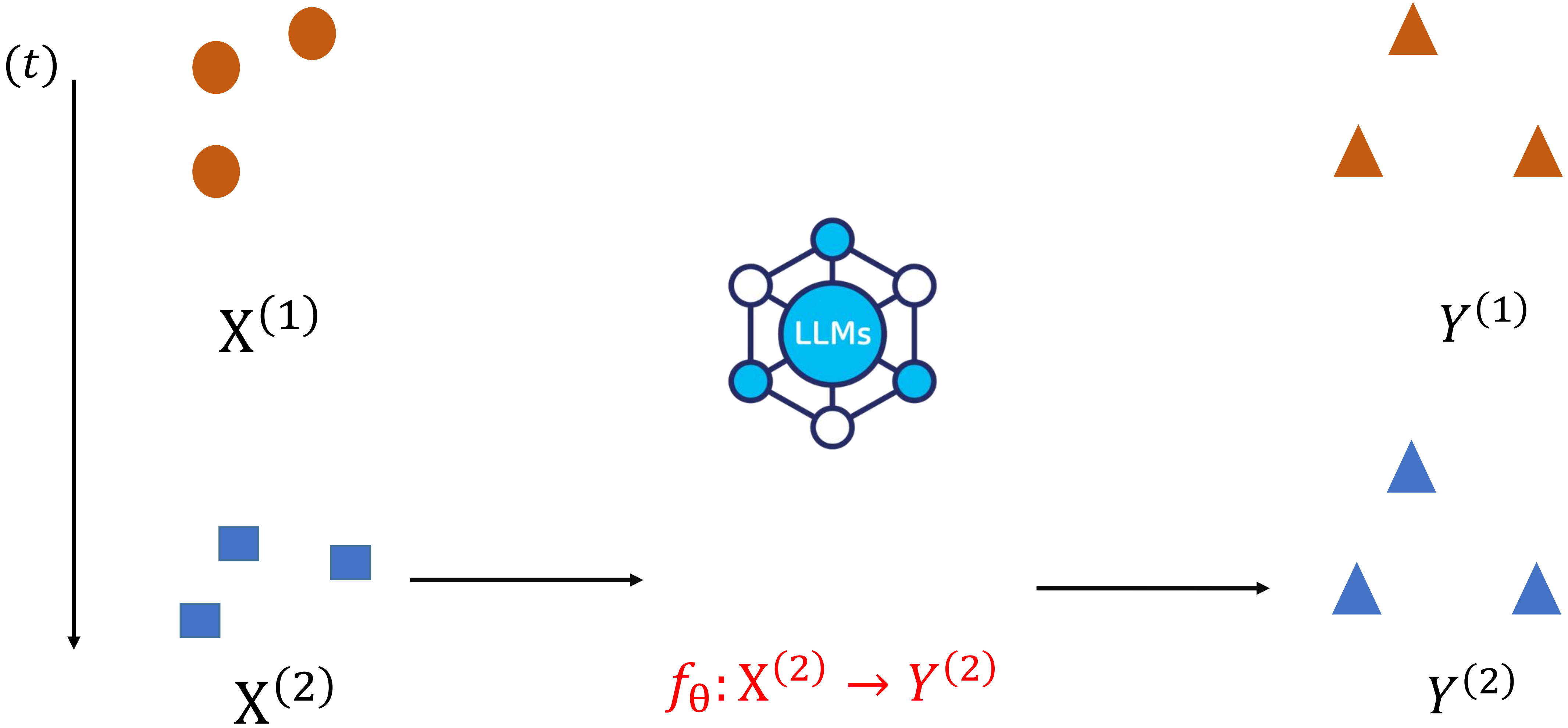
(t)
LLMs
$f_\theta^{(1)} : X^{(1)} \rightarrow Y^{(1)}$
$X^{(1)}$
$Y^{(1)}$
LLMs
$X^{(2)}$
$f_\theta^{(2)} : X^{(2)} \rightarrow Y^{(2)}$
$Y^{(2)}$



Cons
- Models are isolated. They are hard to help each other
- Need to know the task belonging in testing
- Need to train a new model whenever update is needed
- etc.



(t)
$X^{(1)}$
$X^{(2)}$
LLMs
$f_\theta: X^{(2)} \rightarrow Y^{(2)}$
$Y^{(1)}$
$Y^{(2)}$



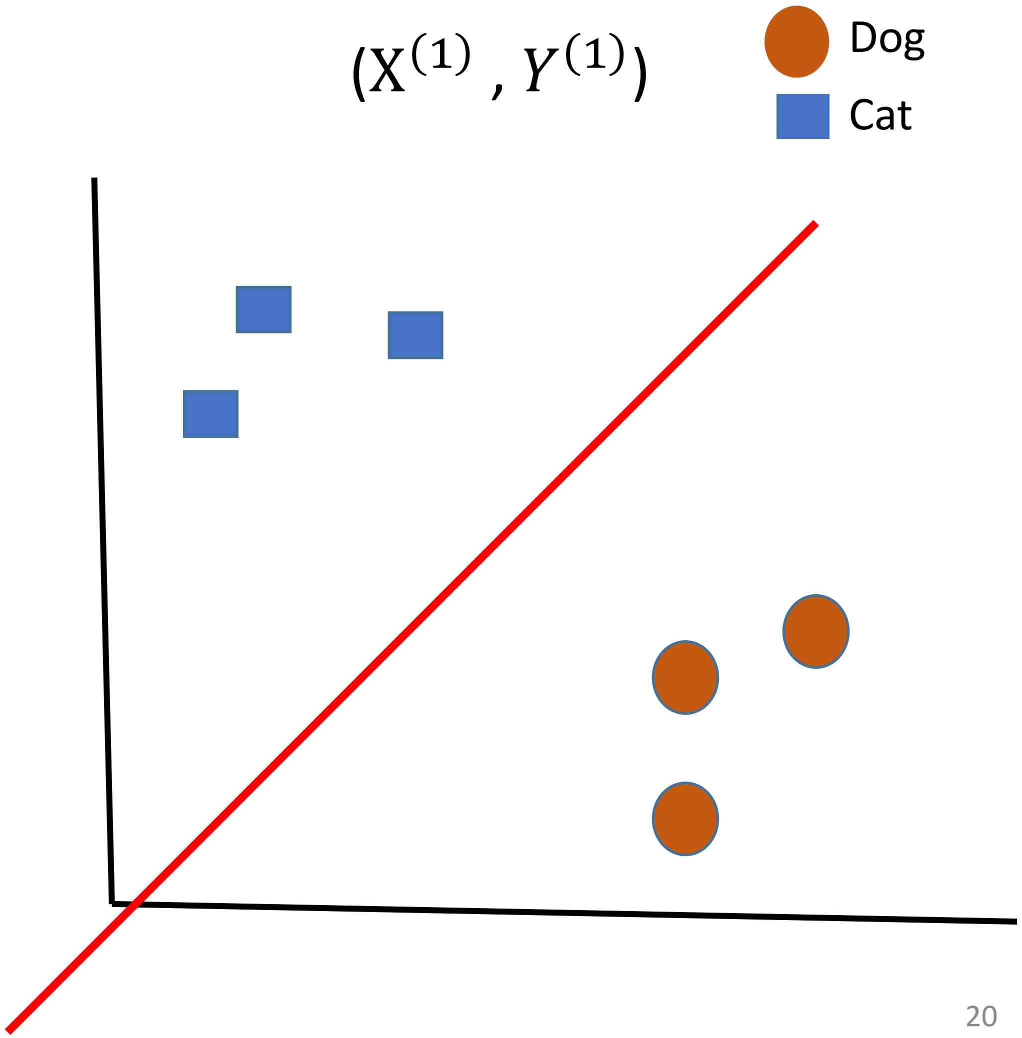
Challenges:

Catastrophic Forgetting (CF)

Knowledge Transfer (KT)

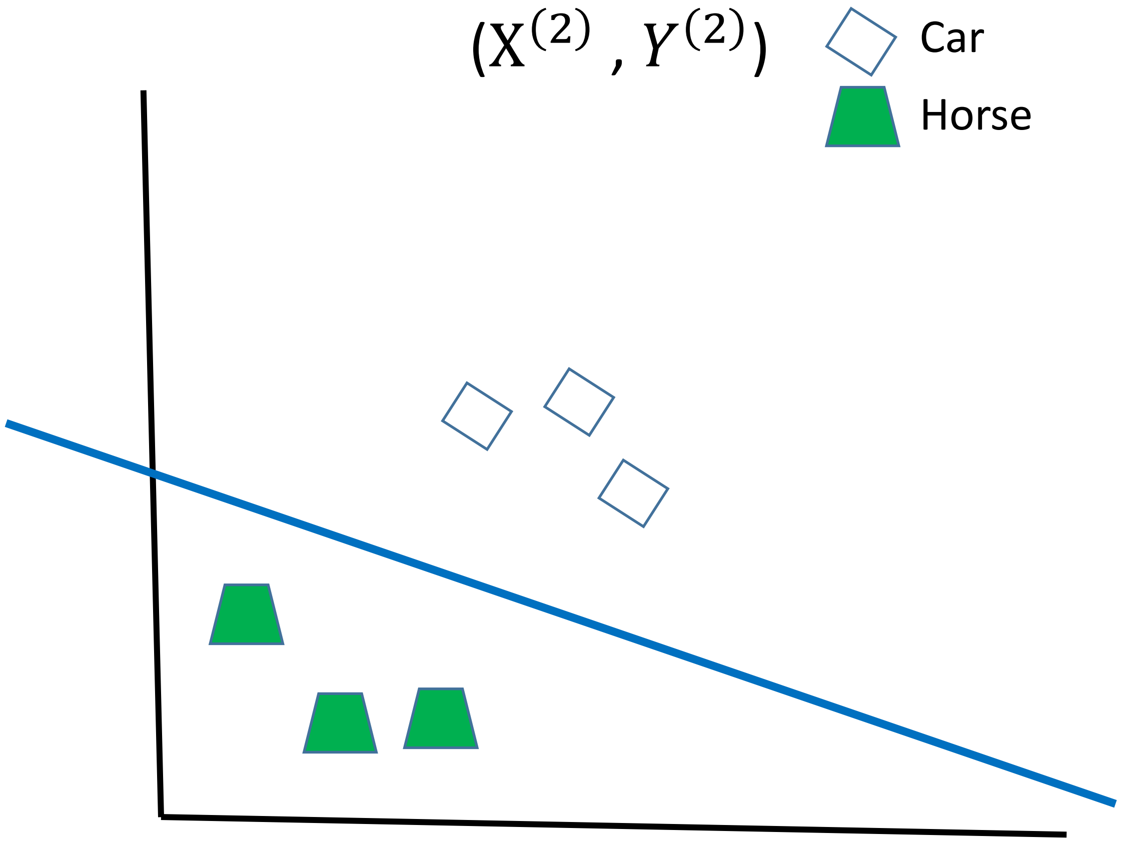
Catastrophic Forgetting

- Simple case:
  - 2 features
  - Learn a line in 2D plane
- You learn a perfect line for task 1



Catastrophic Forgetting

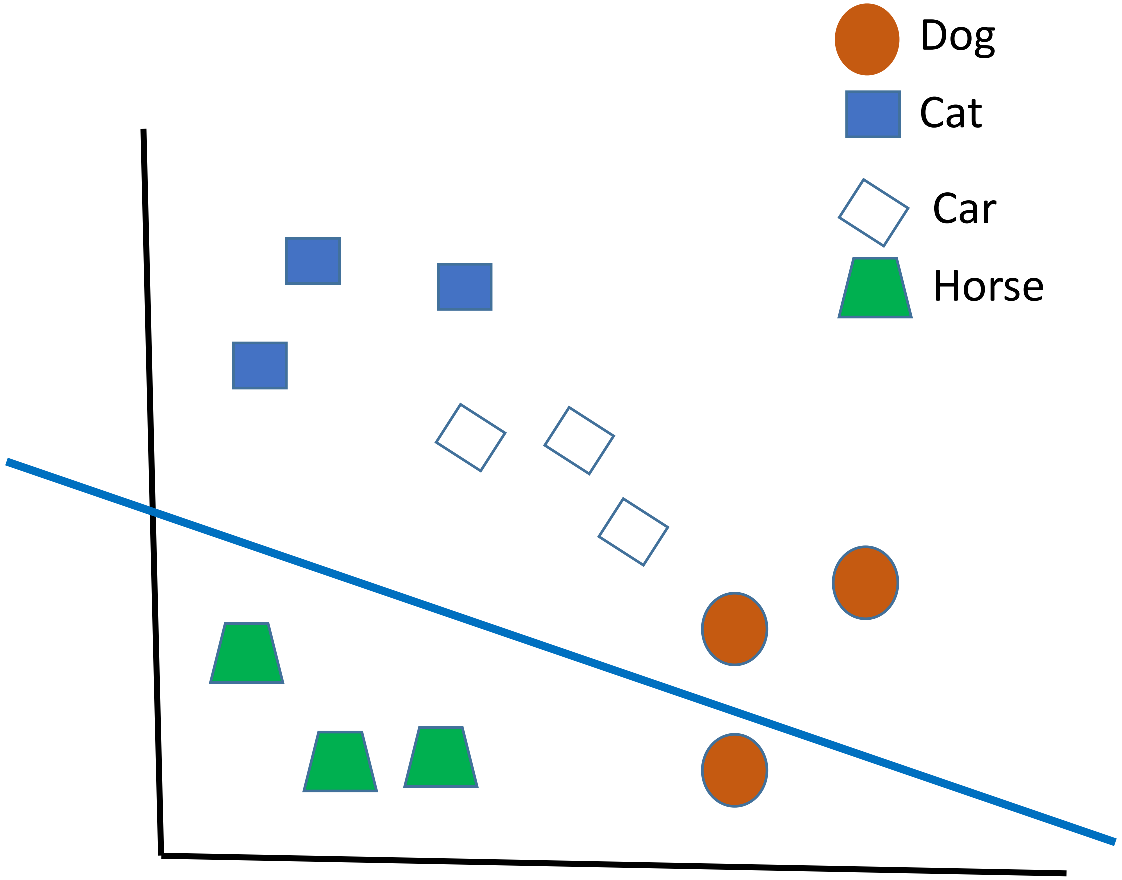
- Simple case:
  - 2 features
  - Learn a line in 2D plane
- **Update** the learned parameters, and learn **another** perfect separate line for task 2



Catastrophic Forgetting

- Simple case:
  - Evaluate the final model on **both** learned tasks (assuming task id unknown)

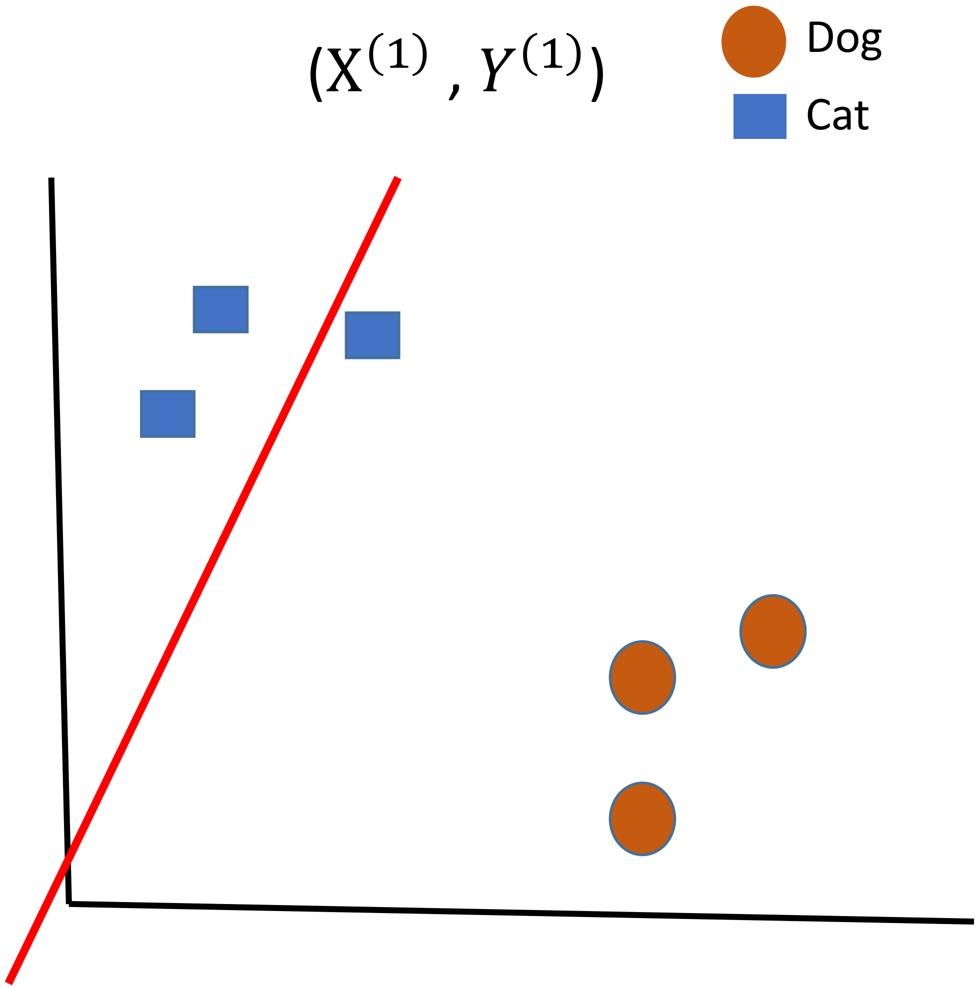
- After learning a second task, you **forget** how to deal with the first task!



Catastrophic Forgetting (CF)

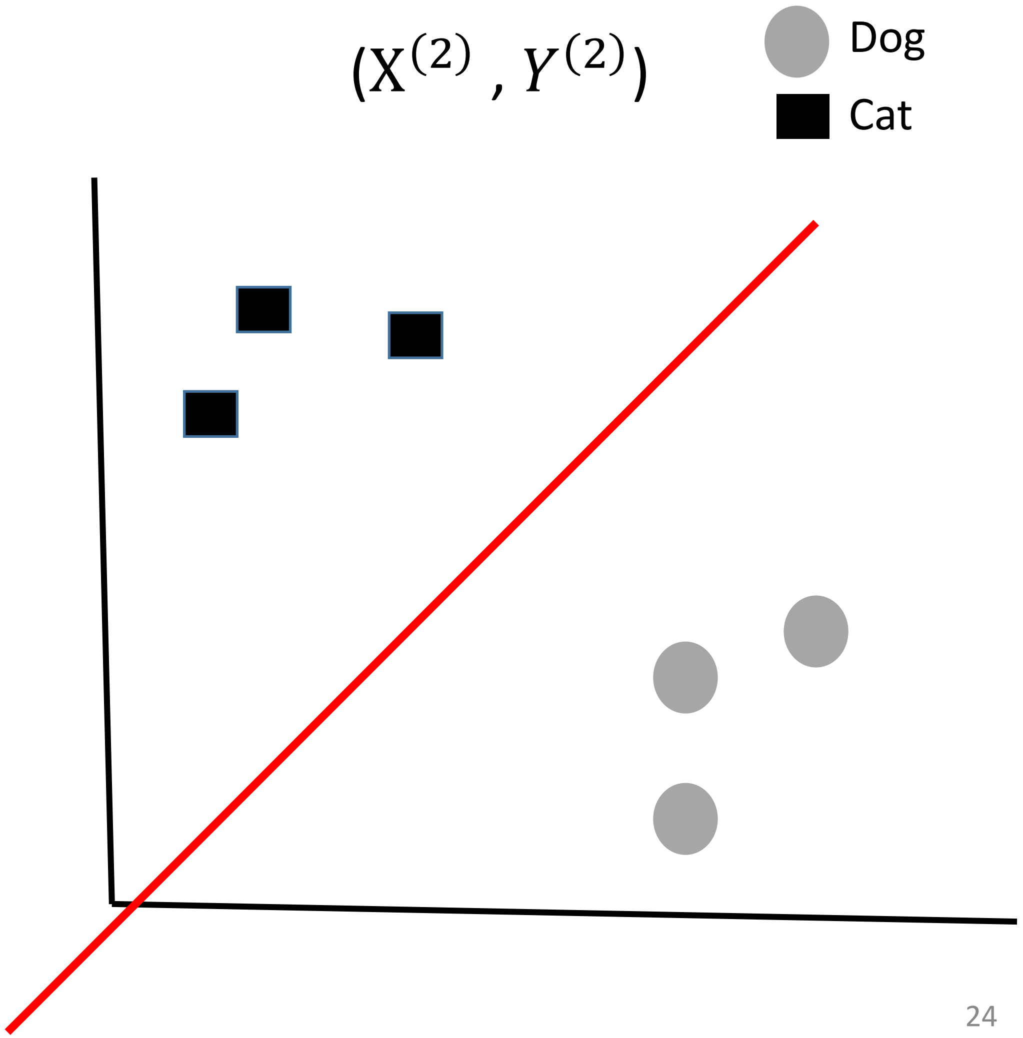
Knowledge Transfer

- Simple case:
  - 2 features
  - Learn a line in 2D plane
- This time, you learn an **imperfect** line for task 1



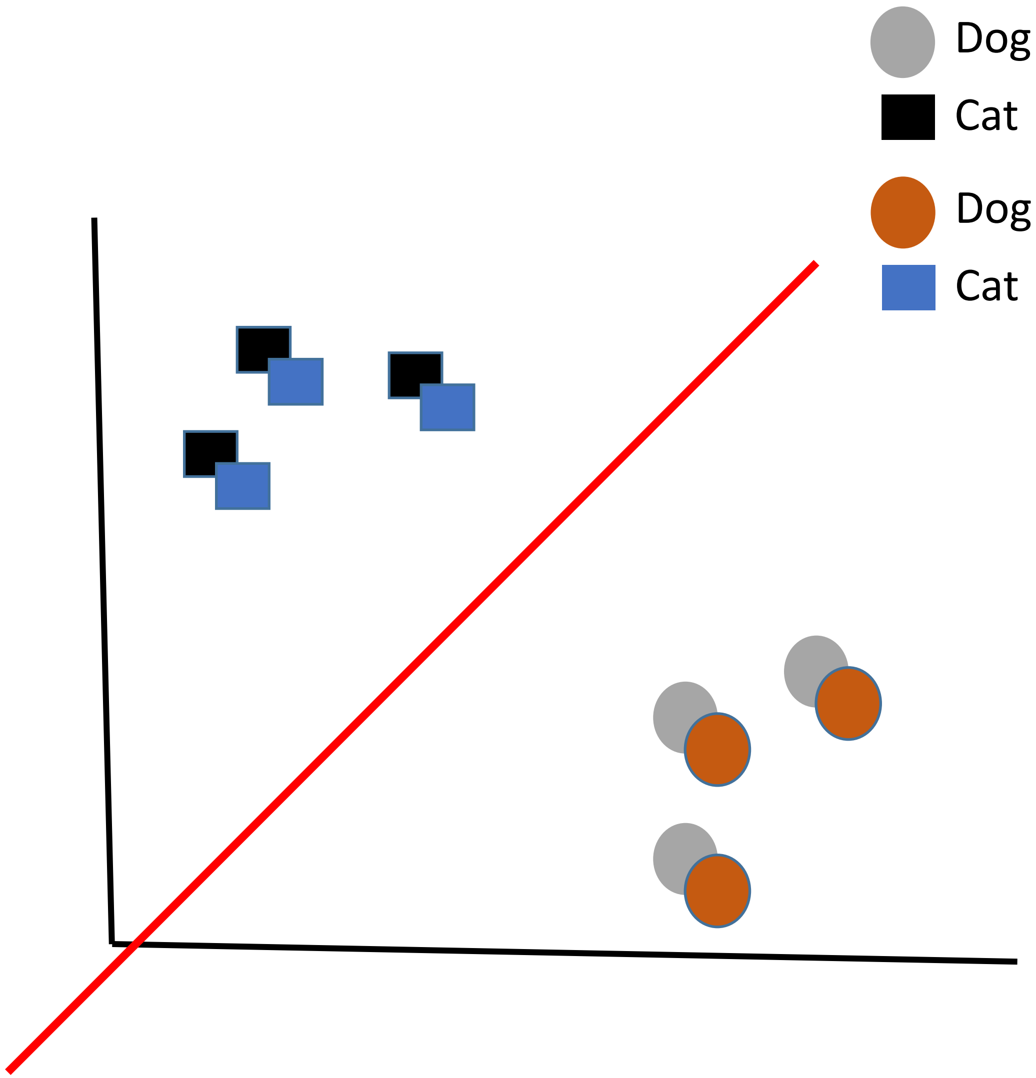
Knowledge Transfer

- Simple case:
  - 2 features
  - Learn a line in 2D plane
- Task 2 has **similar label** as task 1, but the input images becomes **binary**
- You learn a **perfect** line for task 2



Knowledge Transfer

- Simple case:
  - Evaluate the final model on **both** learned tasks
- After Learning a second task, the **old task improved**
- Because the knowledge from task 2 is **helpful** to task 1



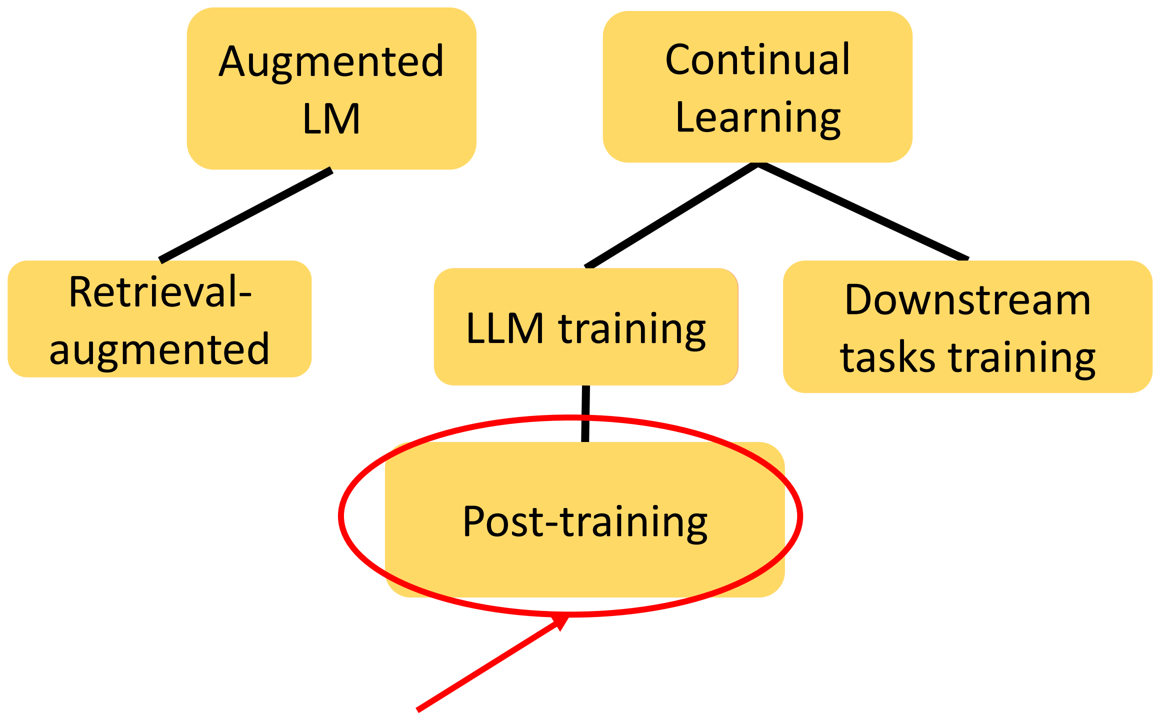
Knowledge Transfer (KT)

Continual Learning

How to
(1) **Mitigate forgetting**, i.e., perform reasonably well on what has been learned
(2) **Knowledge transfer**, i.e., relevant tasks can help each other

Enhancing LLM for A Dynamic World

How to make knowledge in LLM more **reusable** and **updatable**?



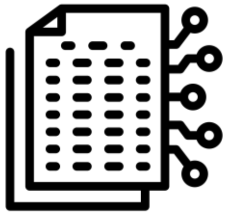
Continual Post-training of Language Models, Ke et al., ICLR 2023

Plan

- Motivation
- Introduction
    - Continual Learning
- Continual Post-training of Language model
- Conclusion and future work

Continual Post-training of Language Model

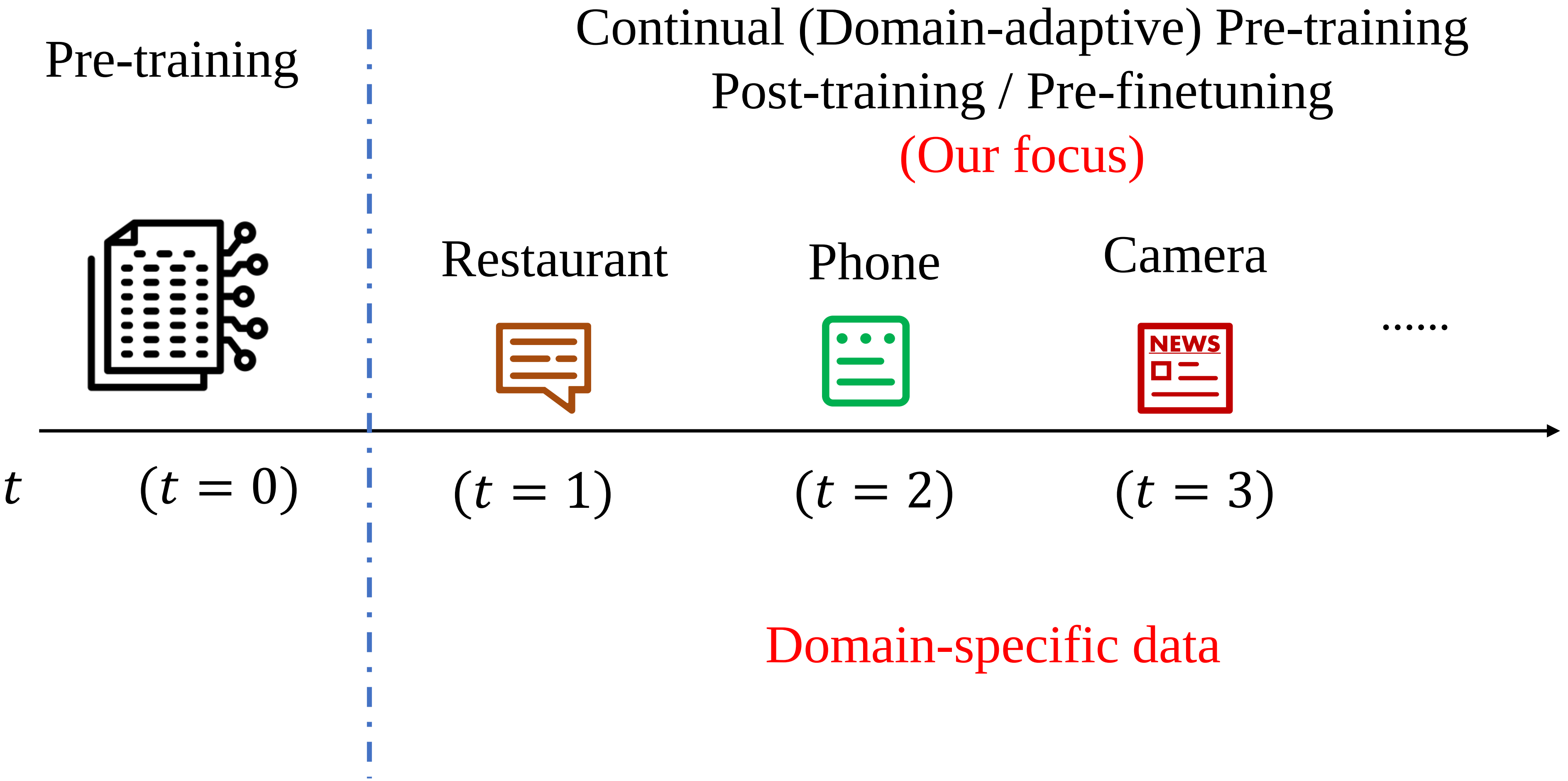
Pre-training



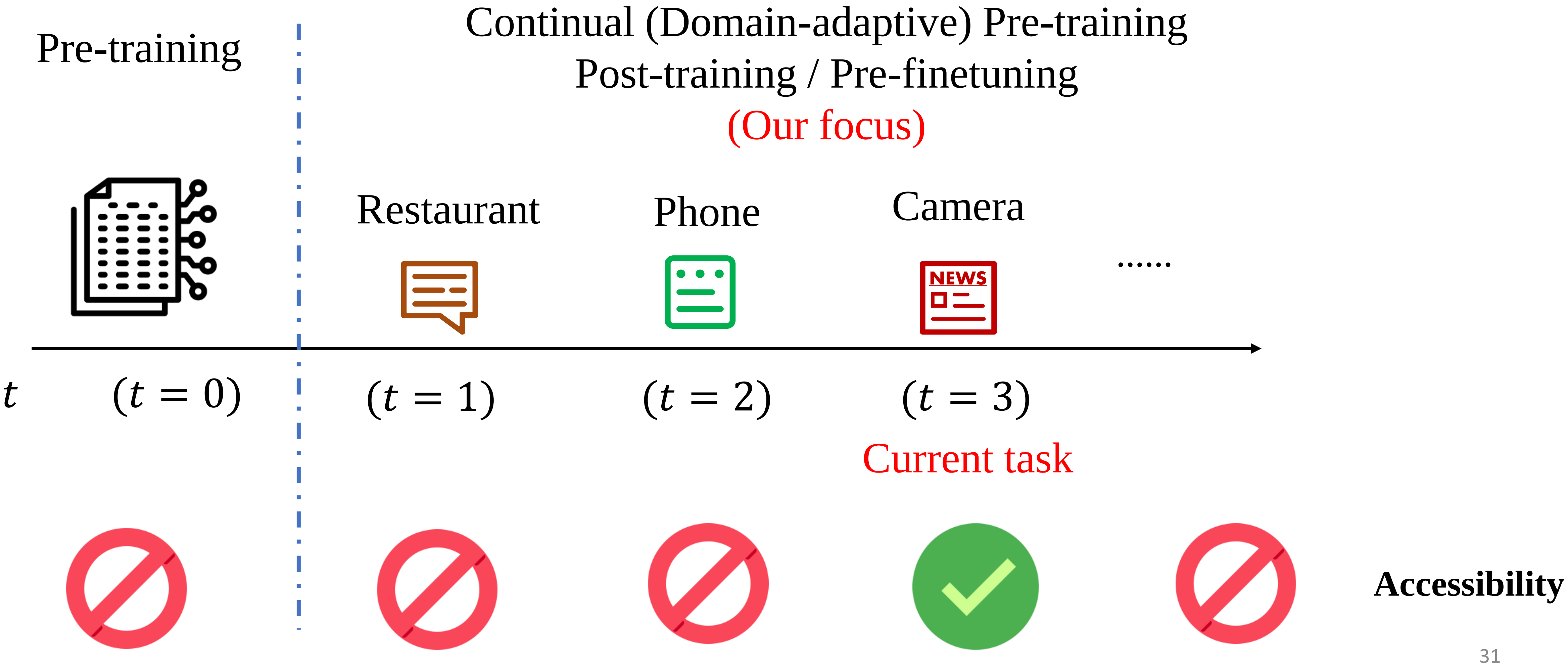
t $(t = 0)$

Huge amount
of general data

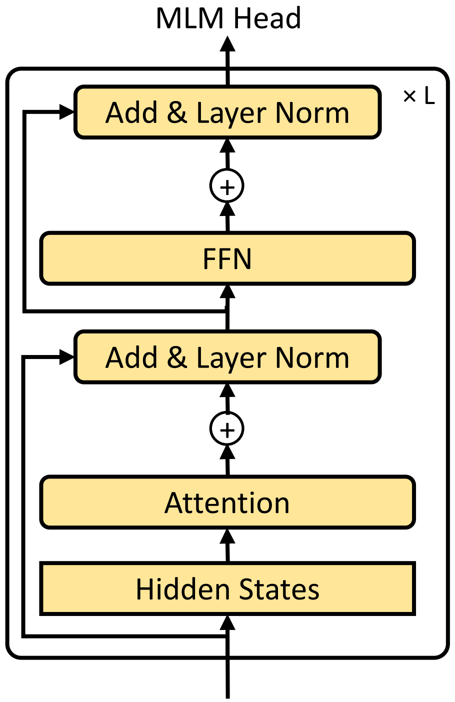
Continual Post-training of Language Model



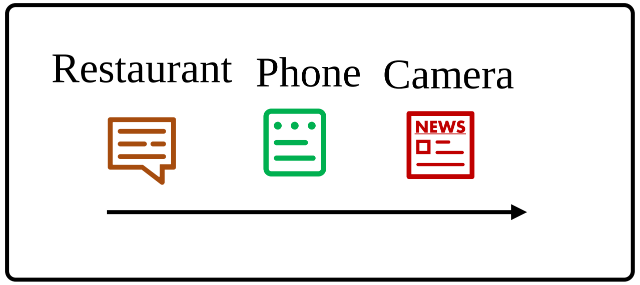
Continual Post-training of Language Model



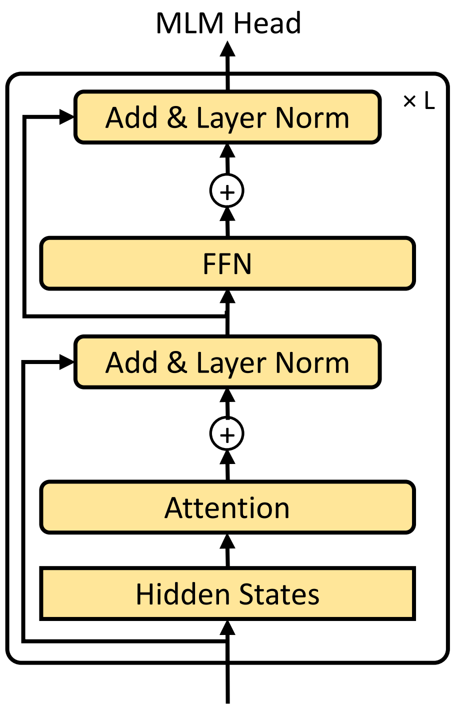
(A) Continual Post-training



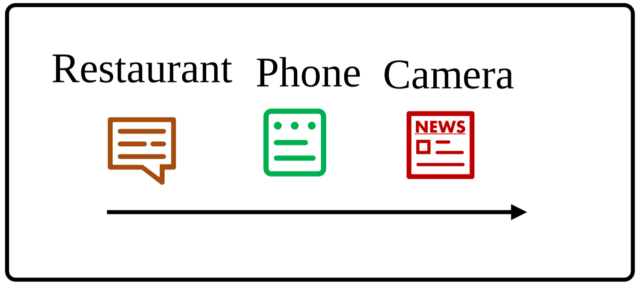
(We use RoBERTa in this work)



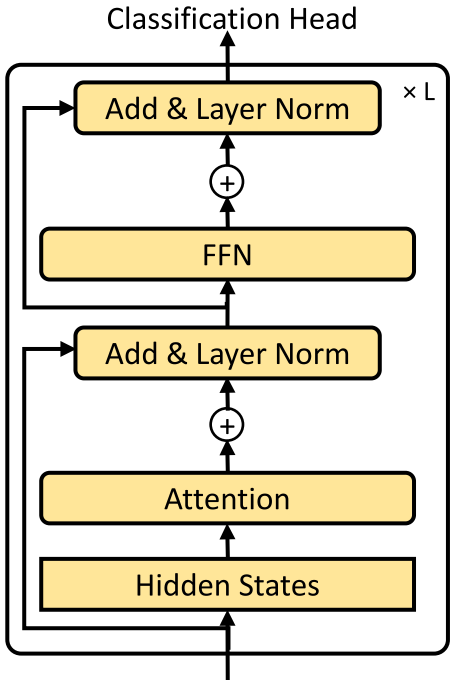
First, we continually post-trains **a sequence of domains**



(A) Continual Post-training



First, we continually pre-trains **a sequence of domains**



(B) Individual Fine-tuning

End-tasks

ASC-Restaurant

ASC-Phone

ASC-Camera

ASC: Aspect Sentiment Classification

After (A), the performance is **evaluated** by end-tasks

Each end-task **corresponding** to one domain and has its **own** training and testing set. It is trained individually and **will not** affect the continual learning

Continual Post-training of Language Model

Unlabelde Domain Datasets			End-Task Classification Datasets				
Source	Dataset/Domain	Size	Dataset/Domain	Task	#Training	#Testing	#Classes
Reviews	Yelp Restaurant	758MB	Restaurant	Aspect Sentiment Classification (ASC)	3,452	1,120	3
	Amazon Phone	724MB	Phone	Aspect Sentiment Classification (ASC)	239	553	2
	Amazon Camera	319MB	Camera	Aspect Sentiment Classification (ASC)	230	626	2
Academic Papers	ACL Papers	867MB	ACL	Citation Intent Classification	1,520	421	6
	AI Papers	507MB	AI	Relation Classification	2,260	2,388	7
	PubMed Papers	989MB	PubMed	Chemical-protein Interaction Prediction	2,667	7,398	13

6 domains

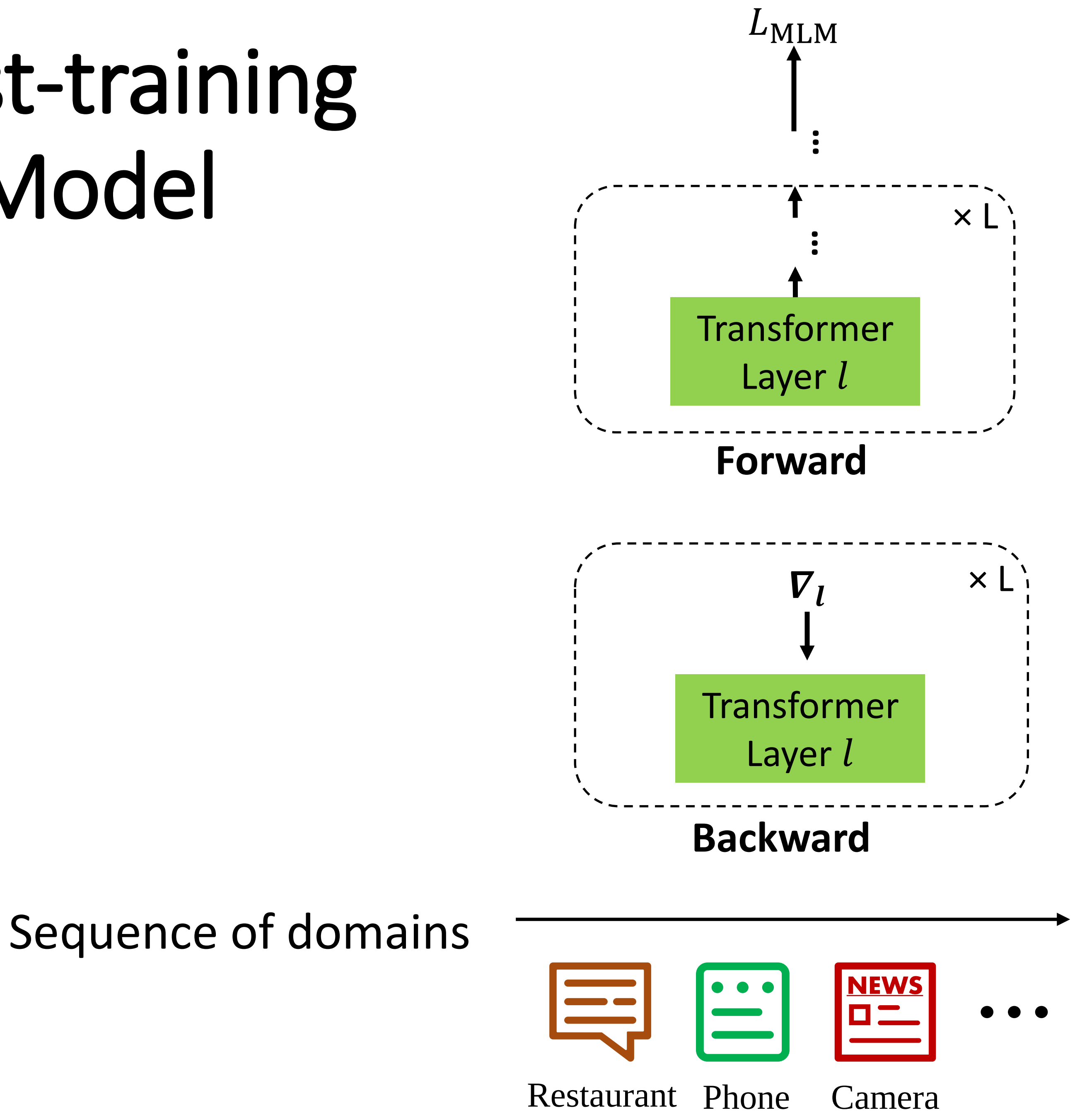
Continual post-training

Individual Fine-tuning

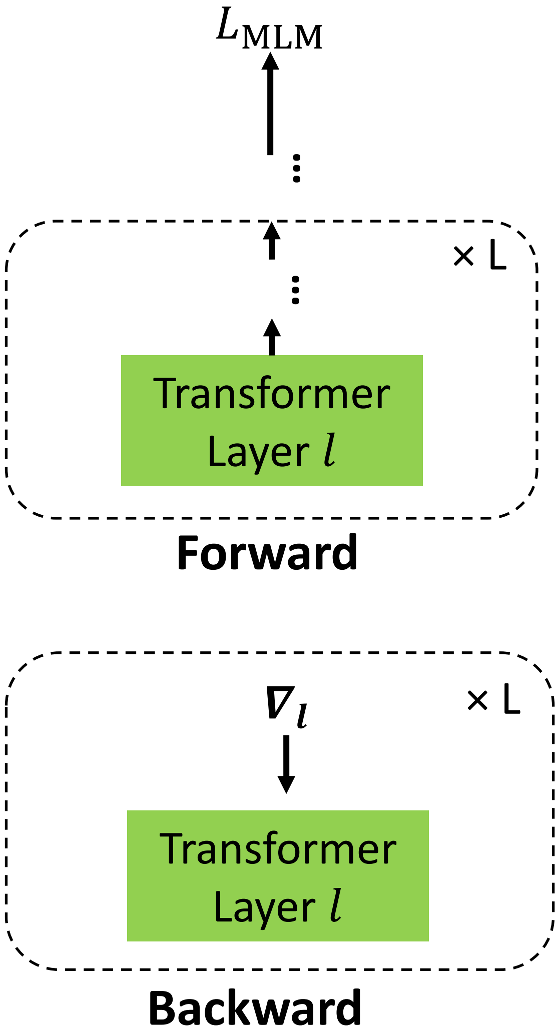
Continual Post-training of Language Model

- Setting
  - Post-train a sequence of domains **without** access to the data that used in **pre-training** and **previously learned domains**
  - End-task doesn't know its domain belonging
- Goals
  - CF prevention
  - KT (backward and forward)
- Related Work
  - There are CL and Post-training work **but** none directly on continual post-training.
- Approach
  - **C**ontinual **P**ost-training with **S**oft-masking (**CPS**)

Continual Post-training of Language Model



Continual Post-training of Language Model

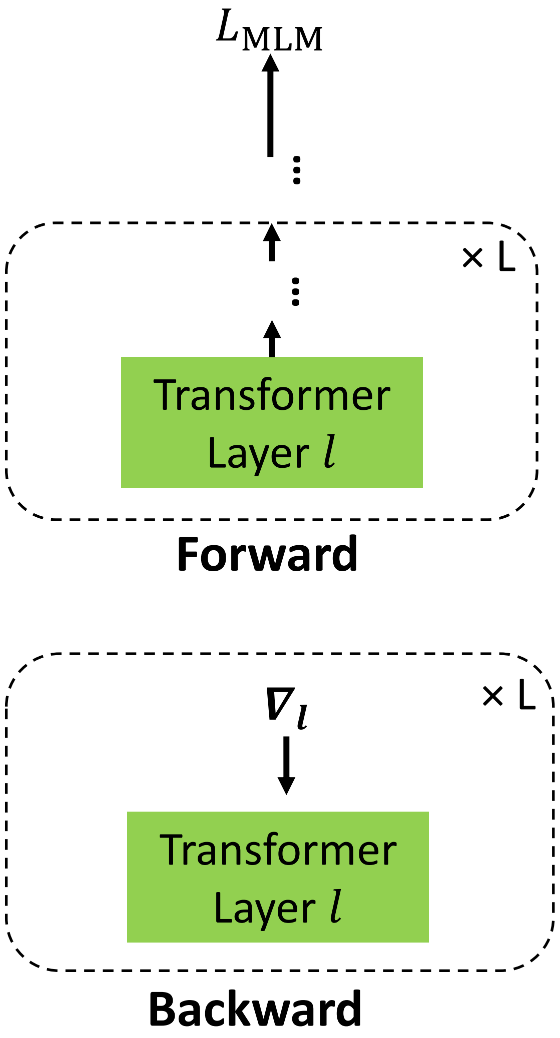


1st Issue: CF on the general knowledge

General knowledge means the knowledge in the **original pre-trained** LM

The knowledge learned from each domain alone **will not be sufficient** to recover it and give good end-task performances

Continual Post-training of Language Model



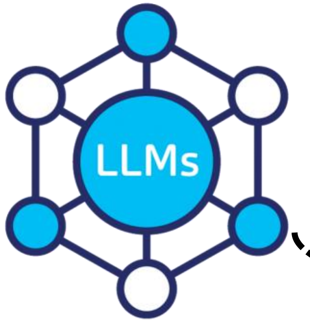
2nd Issue: CF on the previously learned domain knowledge

Because we post-train a sequence of domains

Continual Post-training of Language Model

Pre-trained
LM

Continual Post-training of Language Model



$t = 1$

No training

Importance Computation

1) Detect importance of units (attention heads and neurons) for general and domain knowledge

1) How to detect importance for the two types of knowledge

Continual Post-training of Language Model

$t = 1$

No training — Importance Computation

Post-training — Soft-masking

Backward

Key Idea

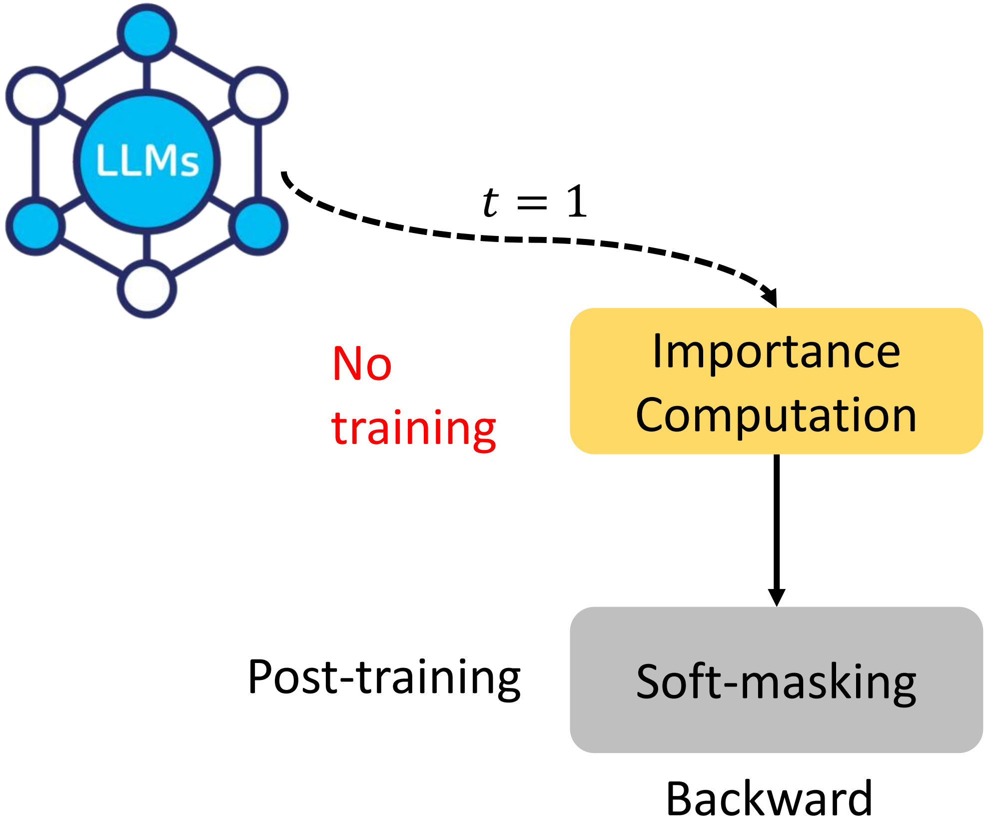
1) Detect importance of units for general and domain knowledge

2) Soft-masking the important units when training new tasks

1) How to detect importance for the two types of knowledge

2) How to soft-mask

Continual Post-training of Language Model

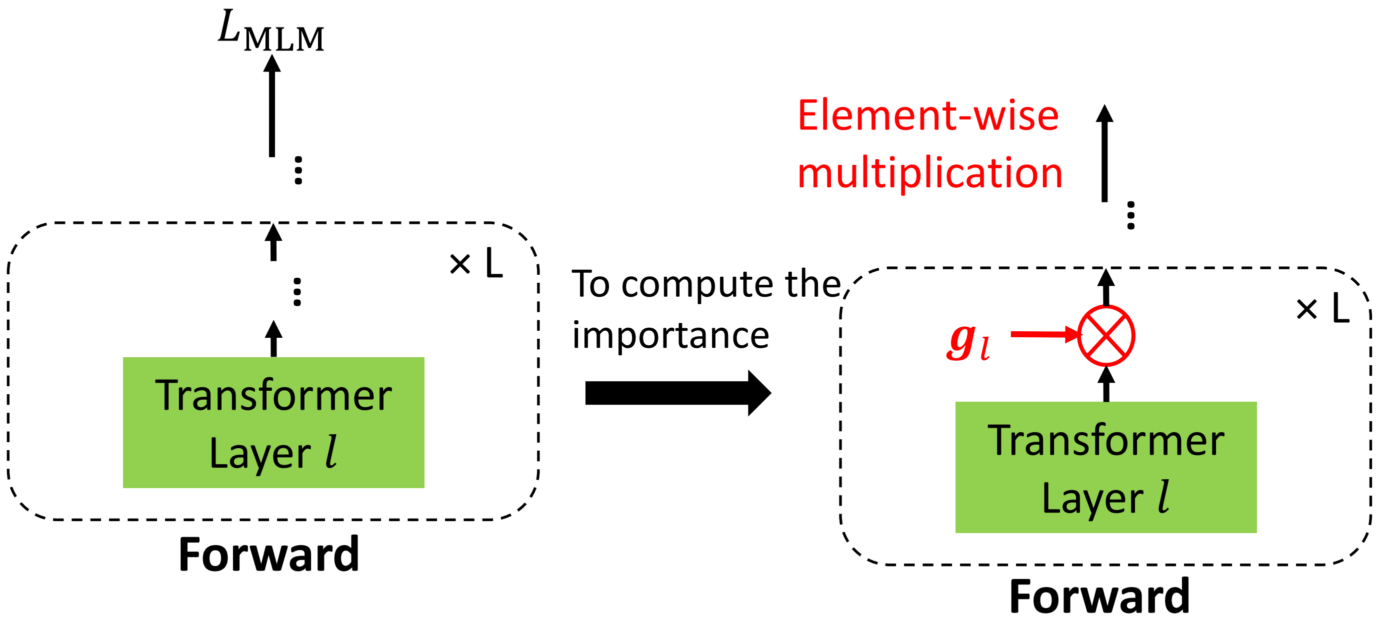


Goal: Compute the importance of units for **general** (and domain) knowledge

Why?
1) Not all units are important
2) Given the important units, we can protect them afterward

No training involved. We only need the importance

Importance Computation



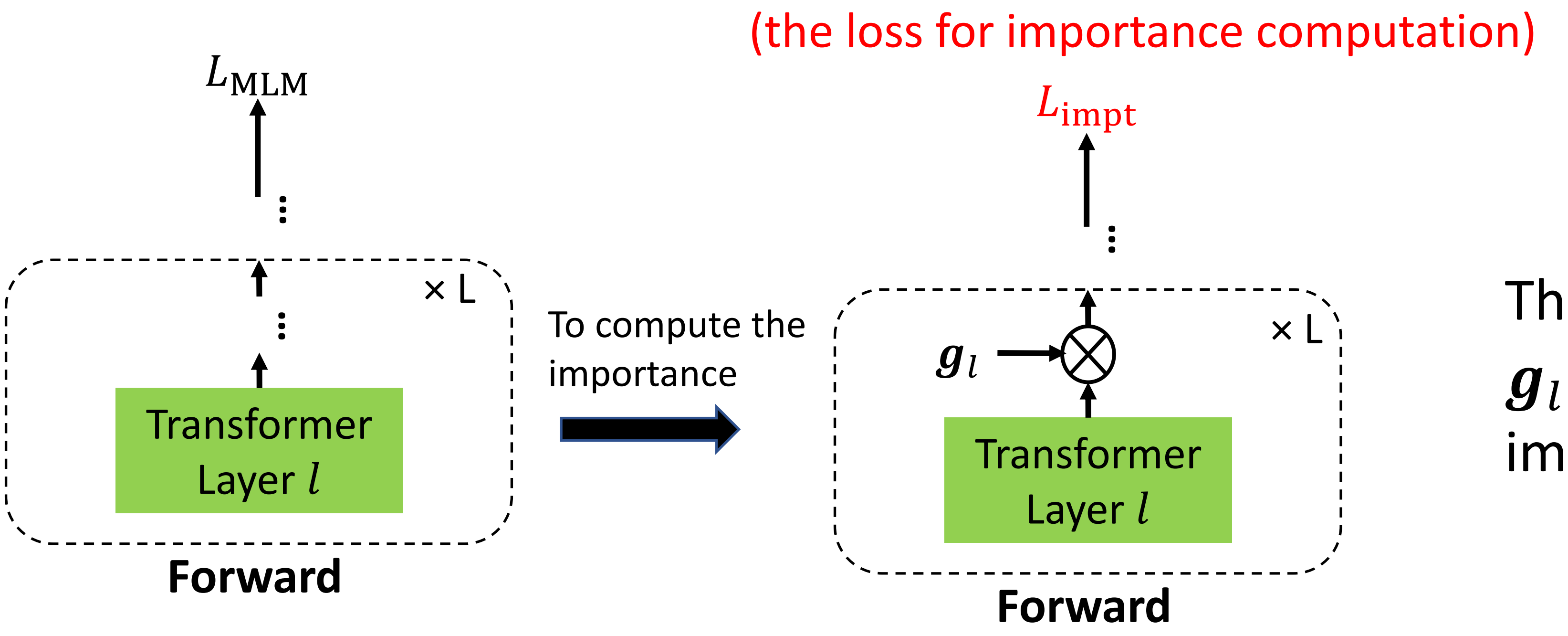
First, we added **virtual parameters** $\boldsymbol{g}_l$.

$\boldsymbol{g}_l$ is the **virtual parameters.** Each virtual parameter $g_{l,i}$ in $\boldsymbol{g}_l$ corresponding to an attention head or neurons (units)

It is **initialized as all 1's**, and has its gradient but will **never change**.

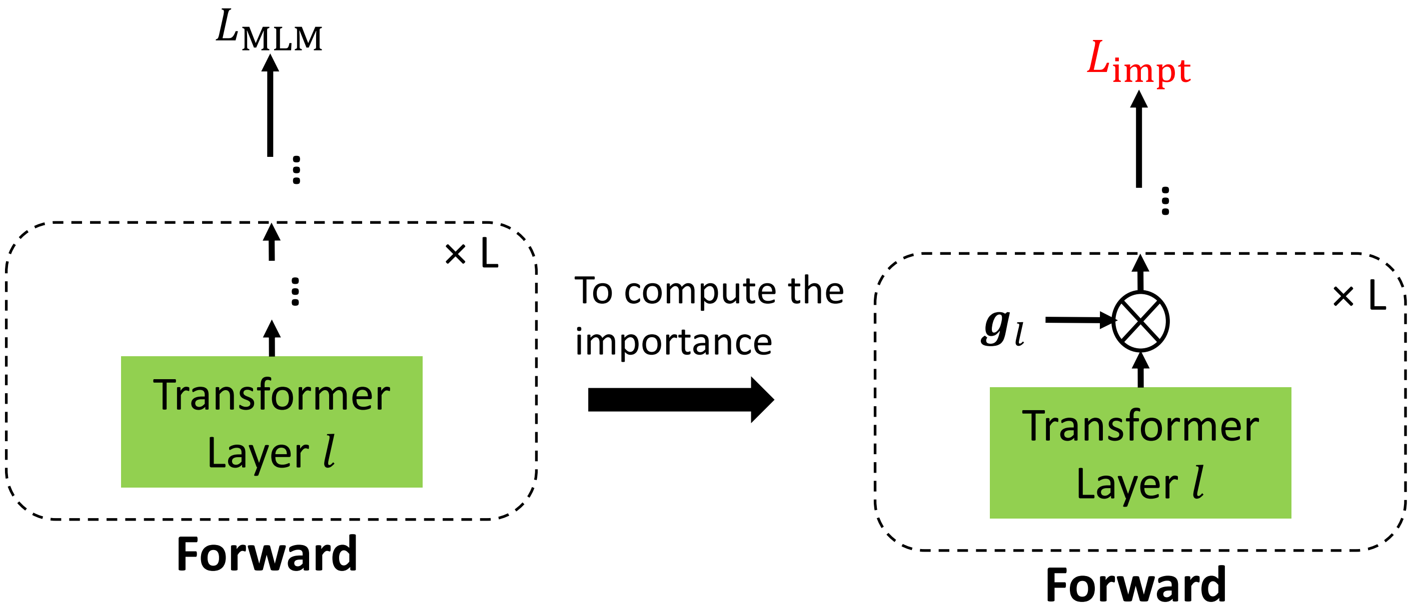
Why? We only use its gradient to compute importance

Importance Computation



The gradient of $\boldsymbol{L}_{impt}$ w.r.t $\boldsymbol{g}_l$ will be used to compute importance.

Importance Computation



For **domain knowledge,**

$L_{\text{impt}} = L_{\text{MLM}}$

$$\nabla_{g_l}^{m} = \frac{\partial L_{\text{impt}}(\boldsymbol{x}_m^{(t)}, \boldsymbol{y}_m^{(t)})}{\partial_{\boldsymbol{g}_l}}$$

$$\boldsymbol{I}_l^{(t)} = \frac{1}{M} \sum_M |\nabla_{g_l}^{m}|$$

Use **absolute gradient** to indicate importance[1]

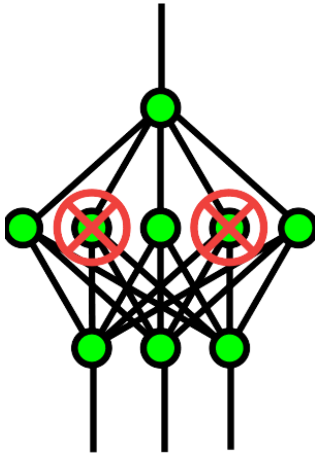
[1] Michel et al. Are sixteen heads really better than one? NeurIPS, 2019.

Importance Computation

However, for **general knowledge**, we cannot do $L_{\text{impt}} = L_{\text{MLM}}$ as we do not have the pre-training data.

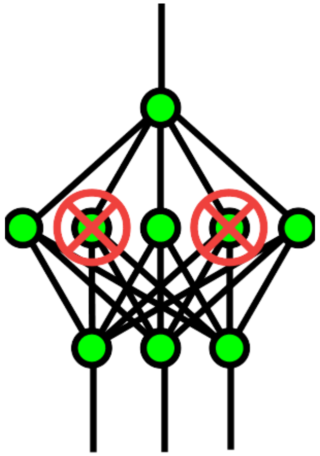
We need another L_{impt}

Importance Computation

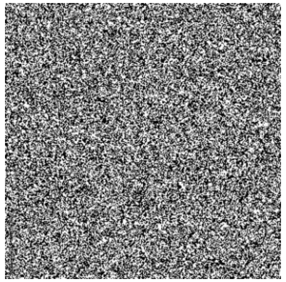


Random dropout

Importance Computation



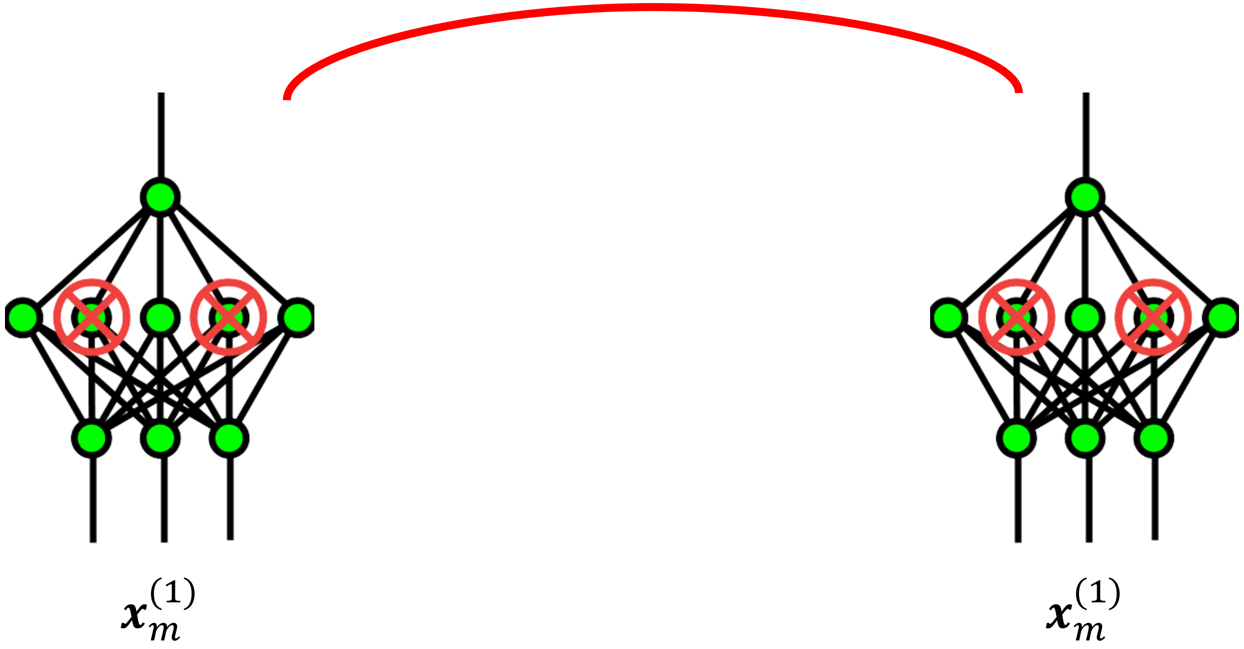
Random dropout



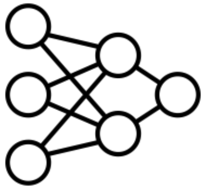
Random noise

Importance Computation

Same input but different output representation
The distance indicate the **robustness**

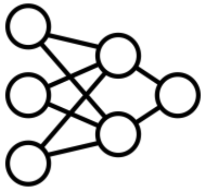


Importance Computation



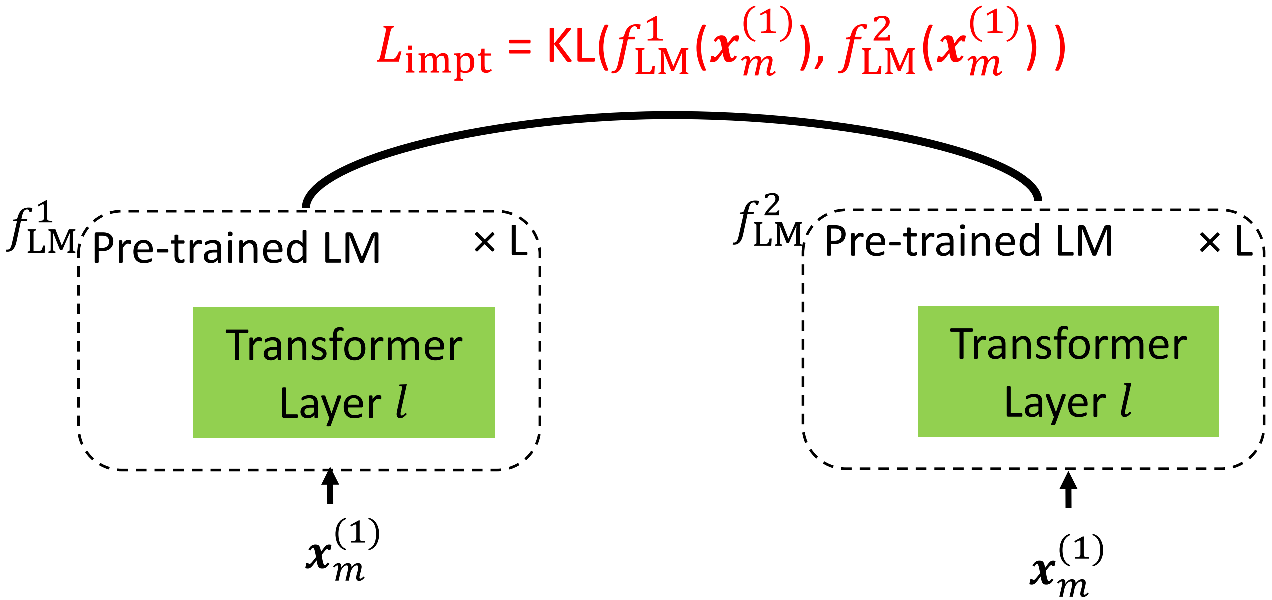
Units that are important to the robustness

their changes will cause the pre-trained LM to change significantly



Units that are important to the pre-trained/general knowledge

Importance Computation



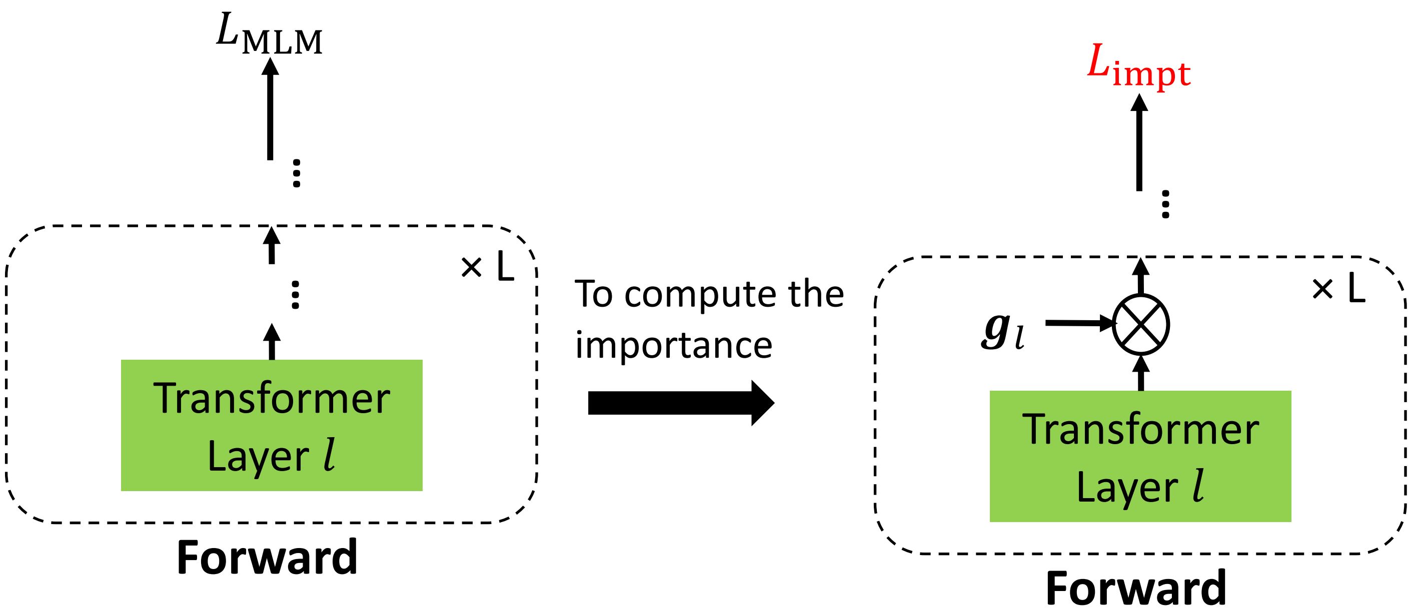
Based on the intuition, we propose another L_{impt}, which do not need pre-training data

KL: how different given two representations

$\boldsymbol{f^1_{LM}}$/ $\boldsymbol{f^2_{LM}}$: Transformer with different dropouts

$\boldsymbol{x^{(1)}_m}$: We only use first domain data because we want the importance of units for the pre-trained knowledge

Importance Computation



For **general knowledge,**

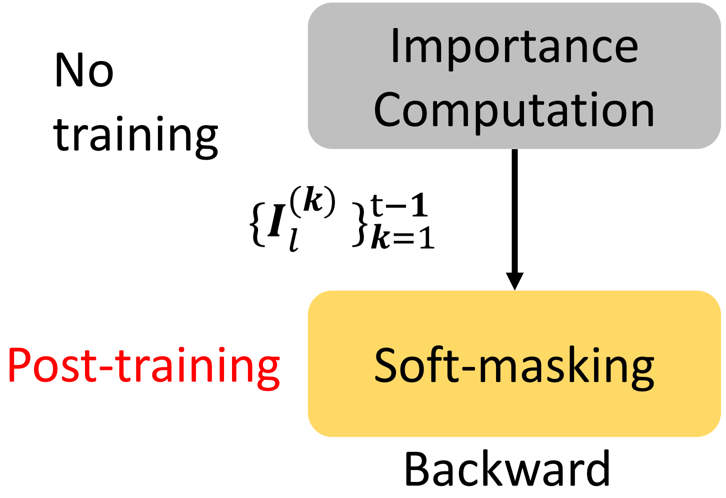
$$L_{\text{impt}} = \text{KL}(f^1_{\text{LM}}(\boldsymbol{x}^{(1)}_m),\ f^2_{\text{LM}}(\boldsymbol{x}^{(1)}_m)\ )$$

$$\boldsymbol{\nabla}^m_{\boldsymbol{g}_l} = \frac{\partial L_{\text{impt}}(\boldsymbol{x}^{(1)}_m)}{\partial_{\boldsymbol{g}_l}}$$

$$\boldsymbol{I}^{(0)}_l = \frac{1}{M} \sum_M |\boldsymbol{\nabla}^m_{\boldsymbol{g}_l}|$$

Importance of units for general knowledge

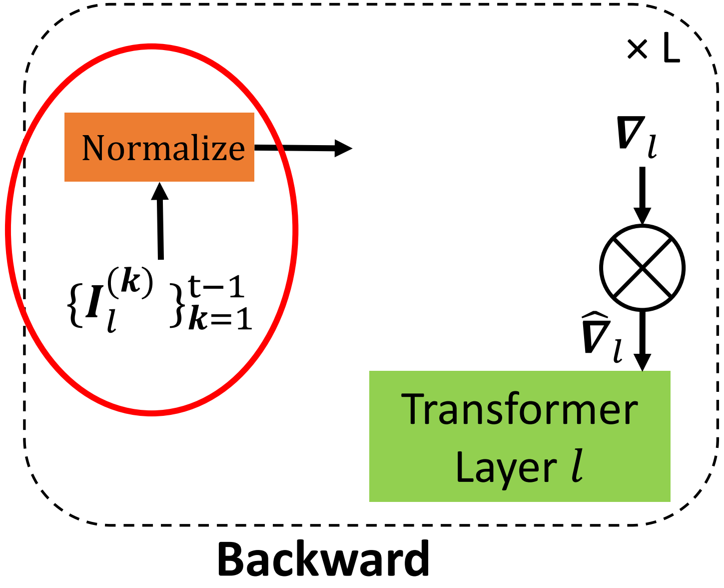
Continual Post-training of Language Model



Goal: Soft-mask the **gradient** based on the importance

Why?
1) We need to protect them when training new domain
2) We want to allow knowledge transfer

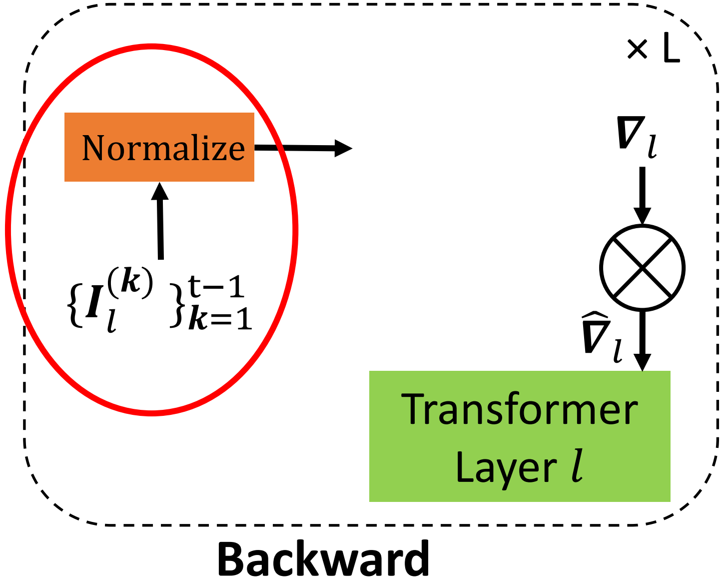
Soft-masking



First, we normalized the importance so that they are comparable

$$\boldsymbol{I}_l^{(k)} = |\mathrm{Tanh}(\mathrm{Norm}(\boldsymbol{I}_l^{(k)}))|$$

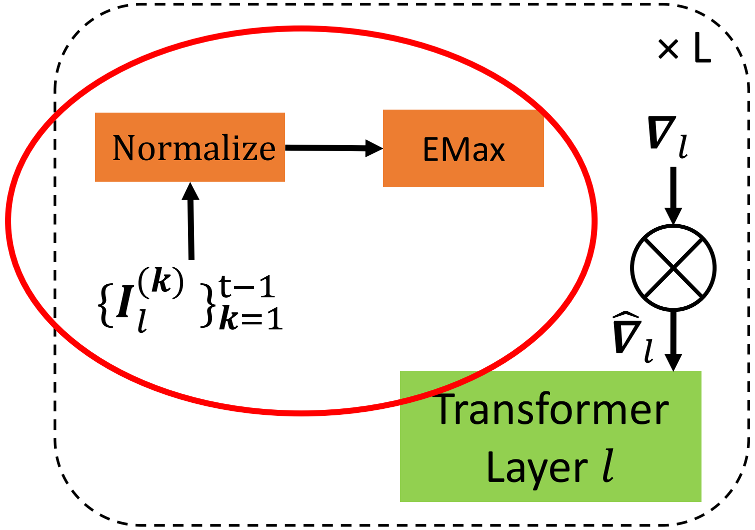
Soft-masking



First, we normalized the importance so that they are comparable

$$\boldsymbol{I}_l^{(k)} = |\mathrm{Tanh}(\mathrm{Norm}(\boldsymbol{I}_l^{(k)}))|$$ make sure the importance is [0,1]

Soft-masking



Backward

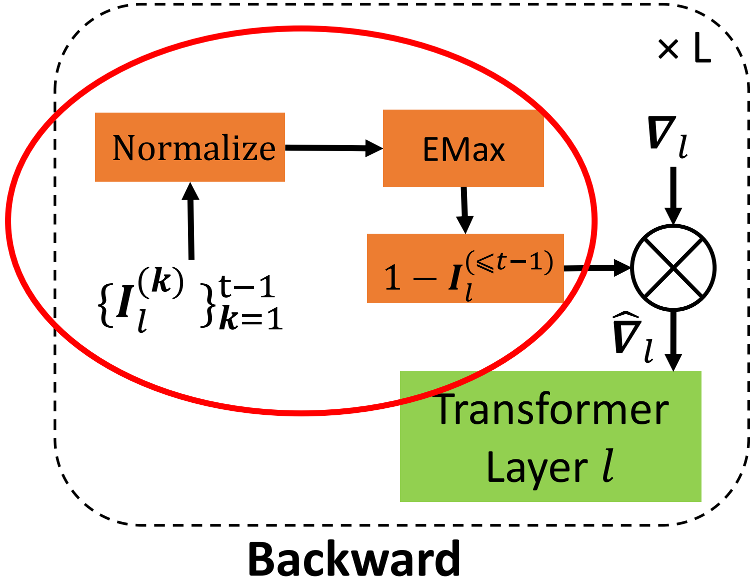
First, we normalized the importance so that they are comparable

$$\boldsymbol{I}_l^{(k)} = |\mathrm{Tanh}(\mathrm{Norm}(\boldsymbol{I}_l^{(k)}))|$$

Second, we accumulate all importance before current domain t

$$\boldsymbol{I}_l^{(\leqslant t-1)} = \mathrm{EMax}(\{\boldsymbol{I}_l^{(t-1)}, \boldsymbol{I}_l^{(t-2)}\})$$

Soft-masking



First, we normalized the importance so that they are comparable

$$\boldsymbol{I}_l^{(k)} = |\mathrm{Tanh}(\mathrm{Norm}(\boldsymbol{I}_l^{(k)}))|$$

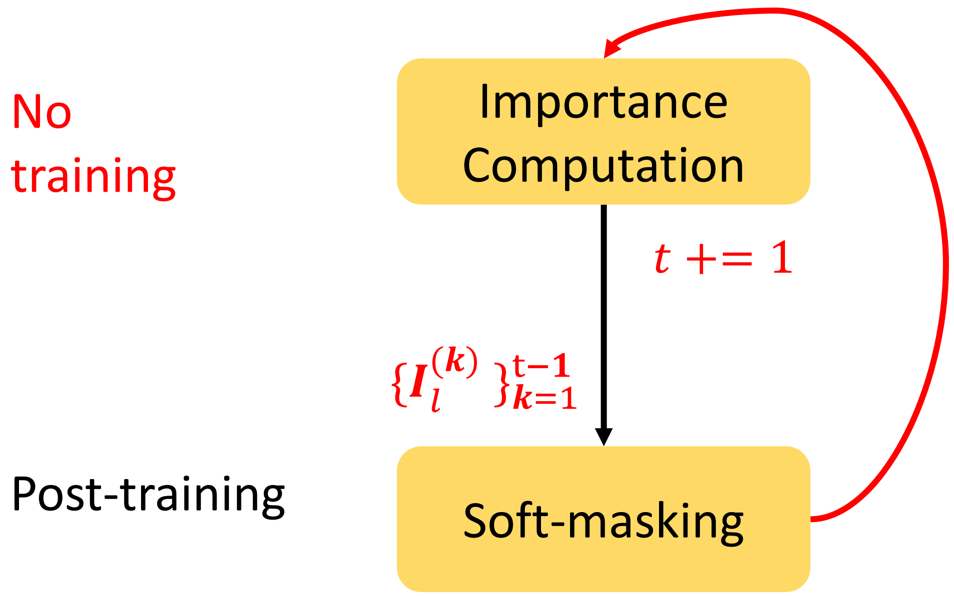
Second, we accumulate the importance

$$\boldsymbol{I}_l^{(\leqslant t-1)} = \mathrm{EMax}(\{\boldsymbol{I}_l^{(t-1)}, \boldsymbol{I}_l^{(t-2)}\})$$

Third, we soft-mask the gradient (in backward pass)

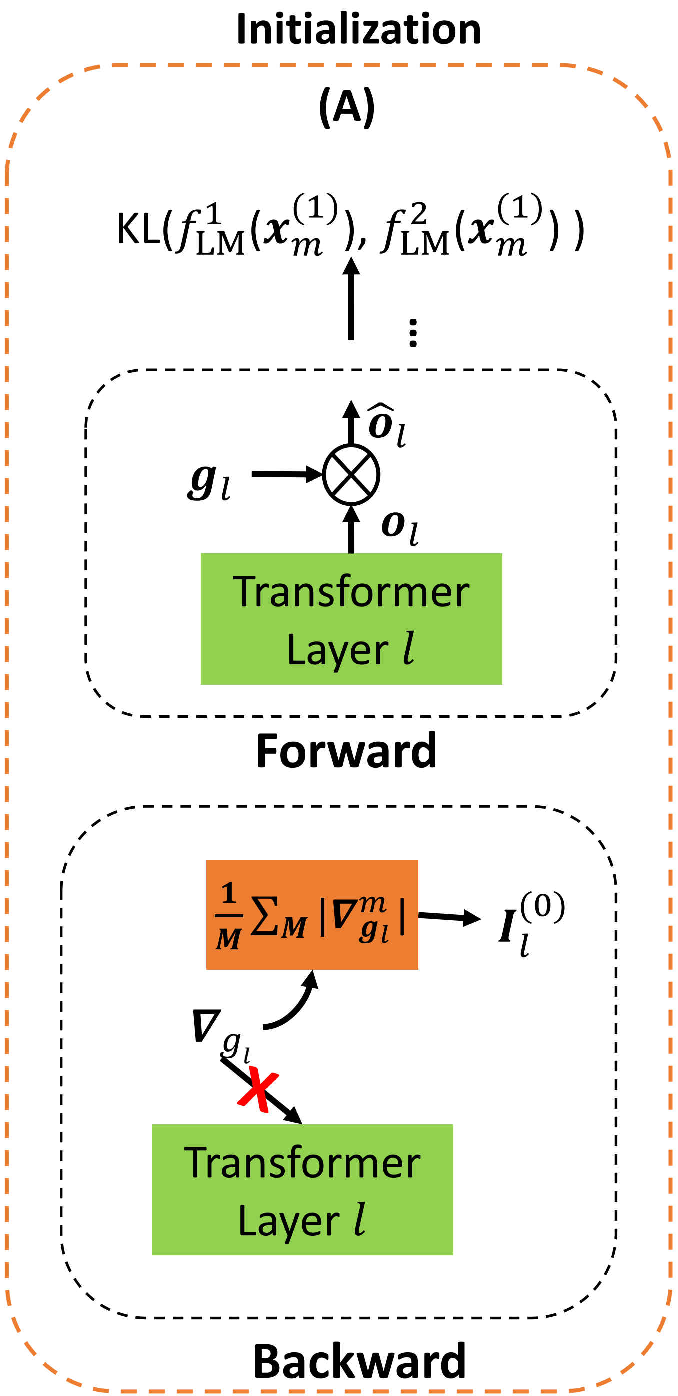
$$\boldsymbol{\nabla}'_l = \left(1 - \boldsymbol{I}_l^{(\leqslant t-1)}\right) \otimes \boldsymbol{\nabla}_l$$

Continual Post-training of Language Model



$$\boldsymbol{\nabla}'_l = \left(1 - \boldsymbol{I}_l^{(\leqslant t-1)}\right) \otimes \boldsymbol{\nabla}_l$$

Backward

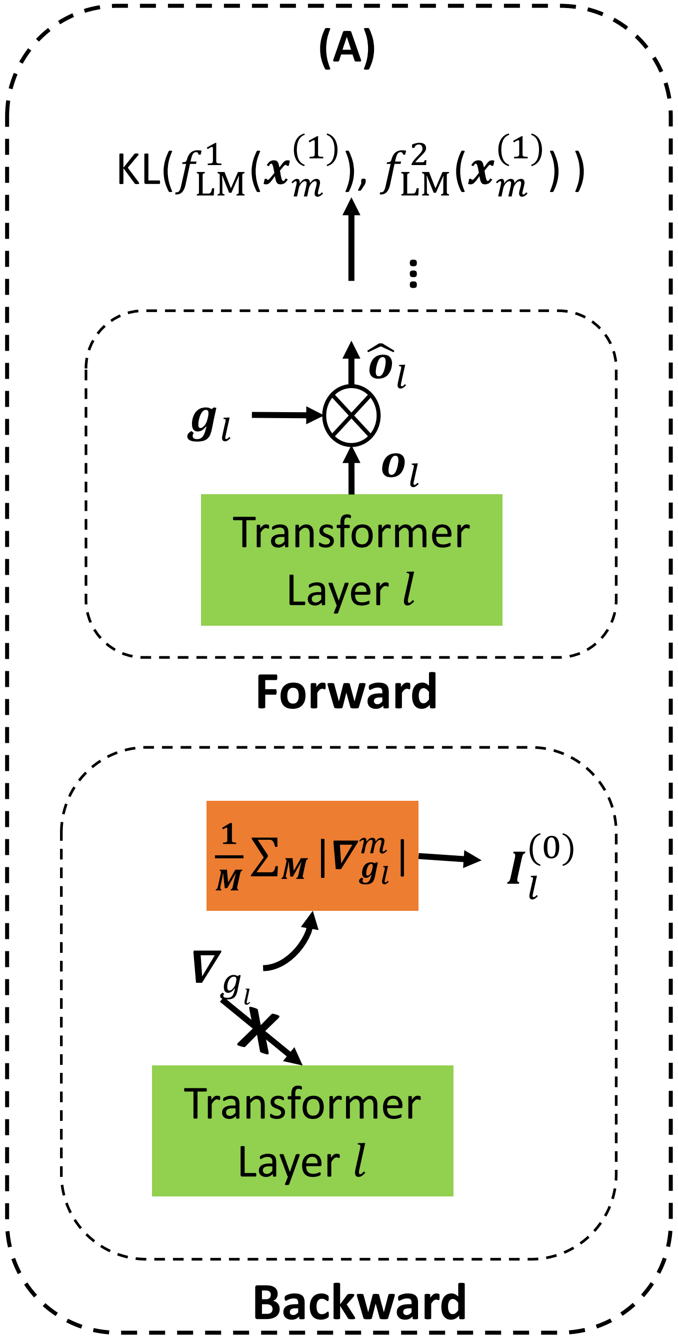


KL loss as L_{impt}

Use gradient to indicate importance, but the gradient does not optimize the layer

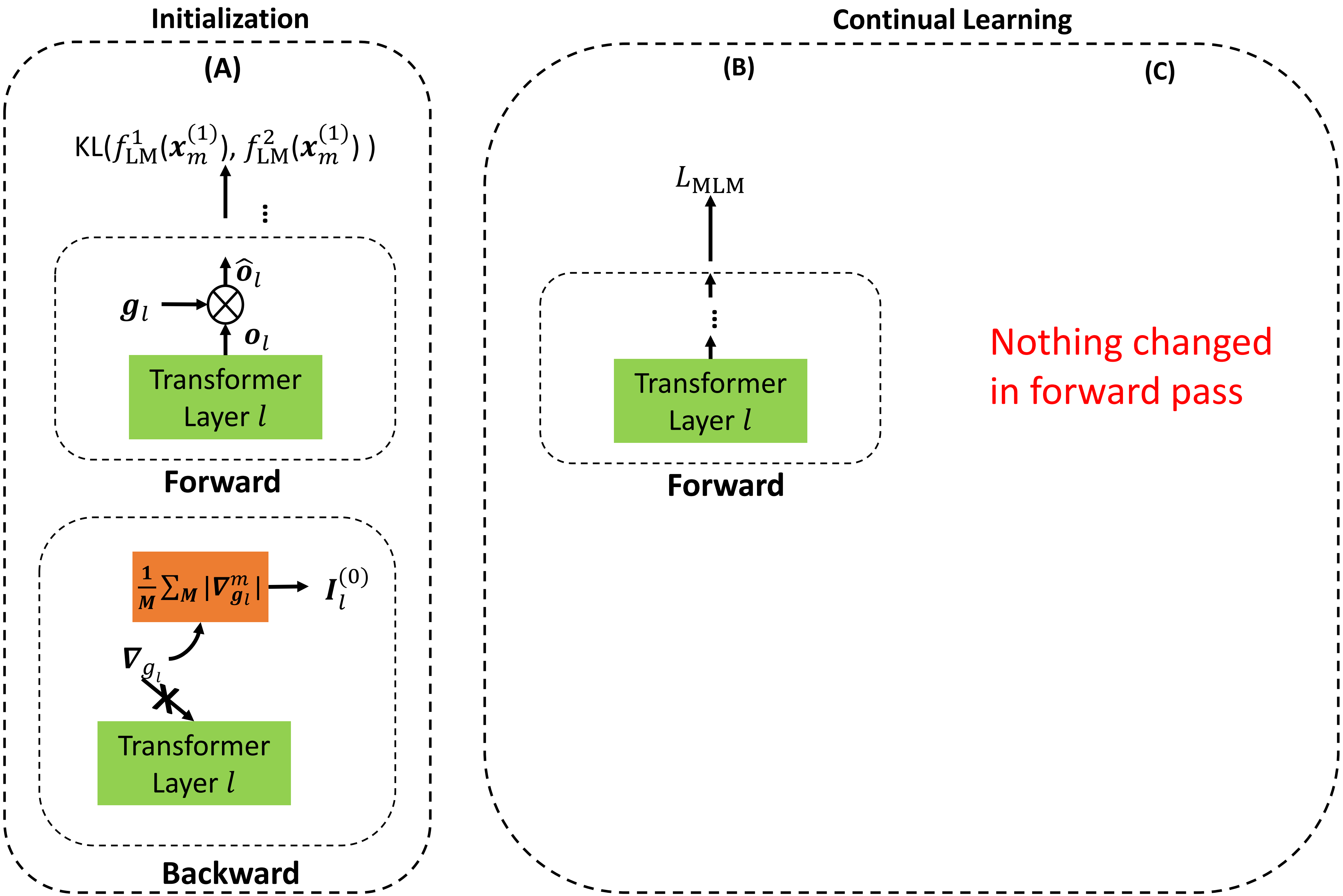
$\boldsymbol{I}_l^{(0)}$ indicates the importance for general knowledge

Initialization

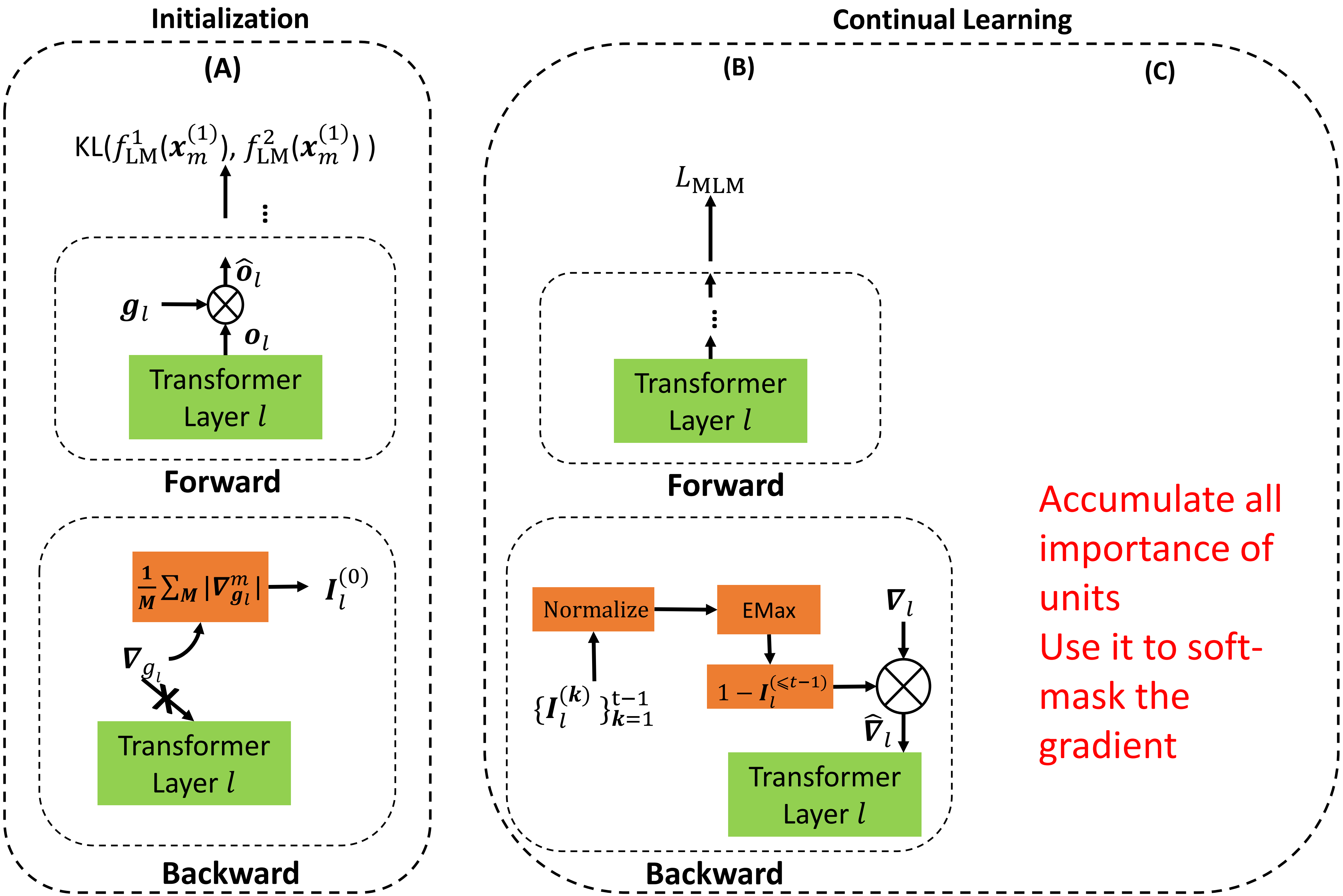


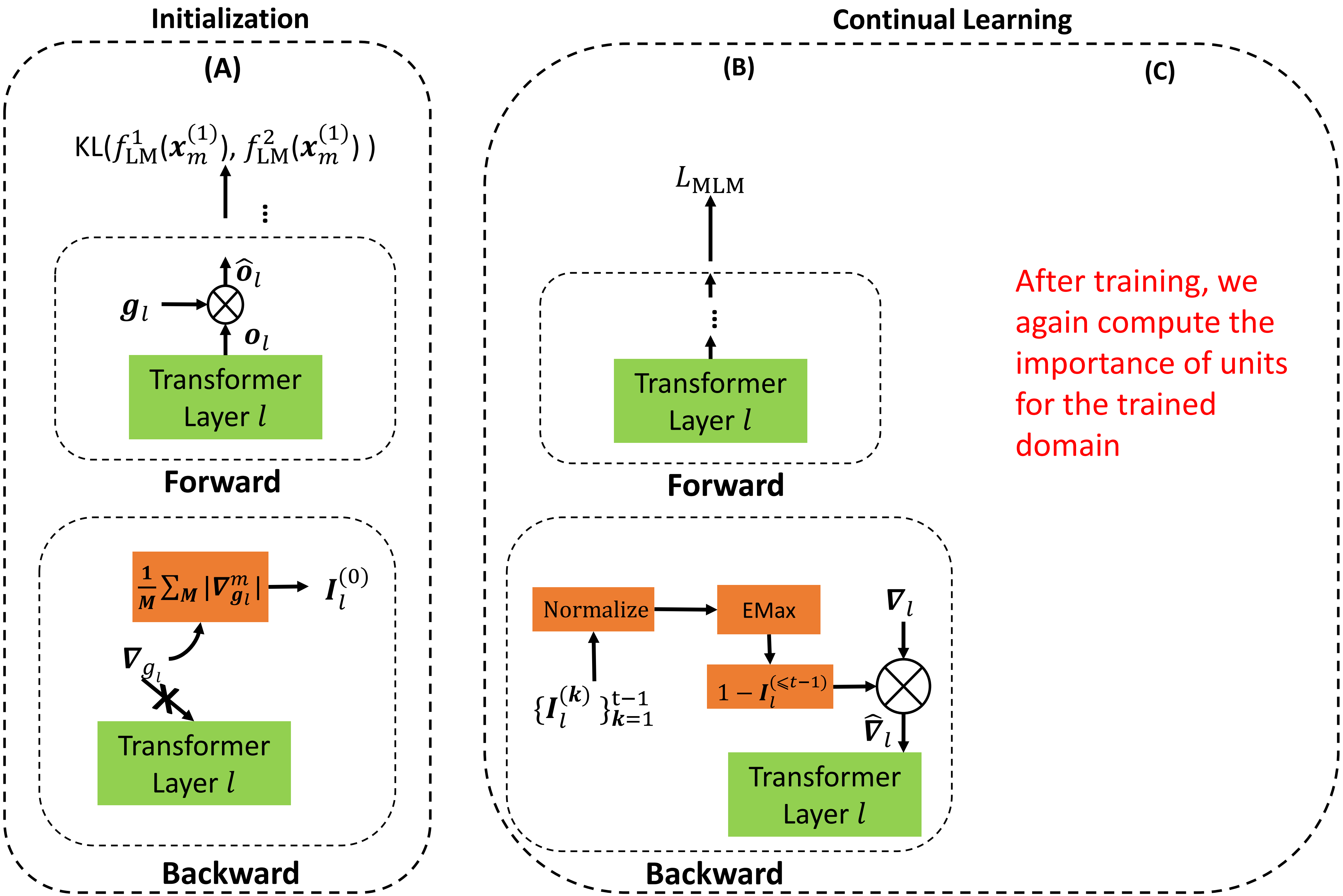
Continual Learning

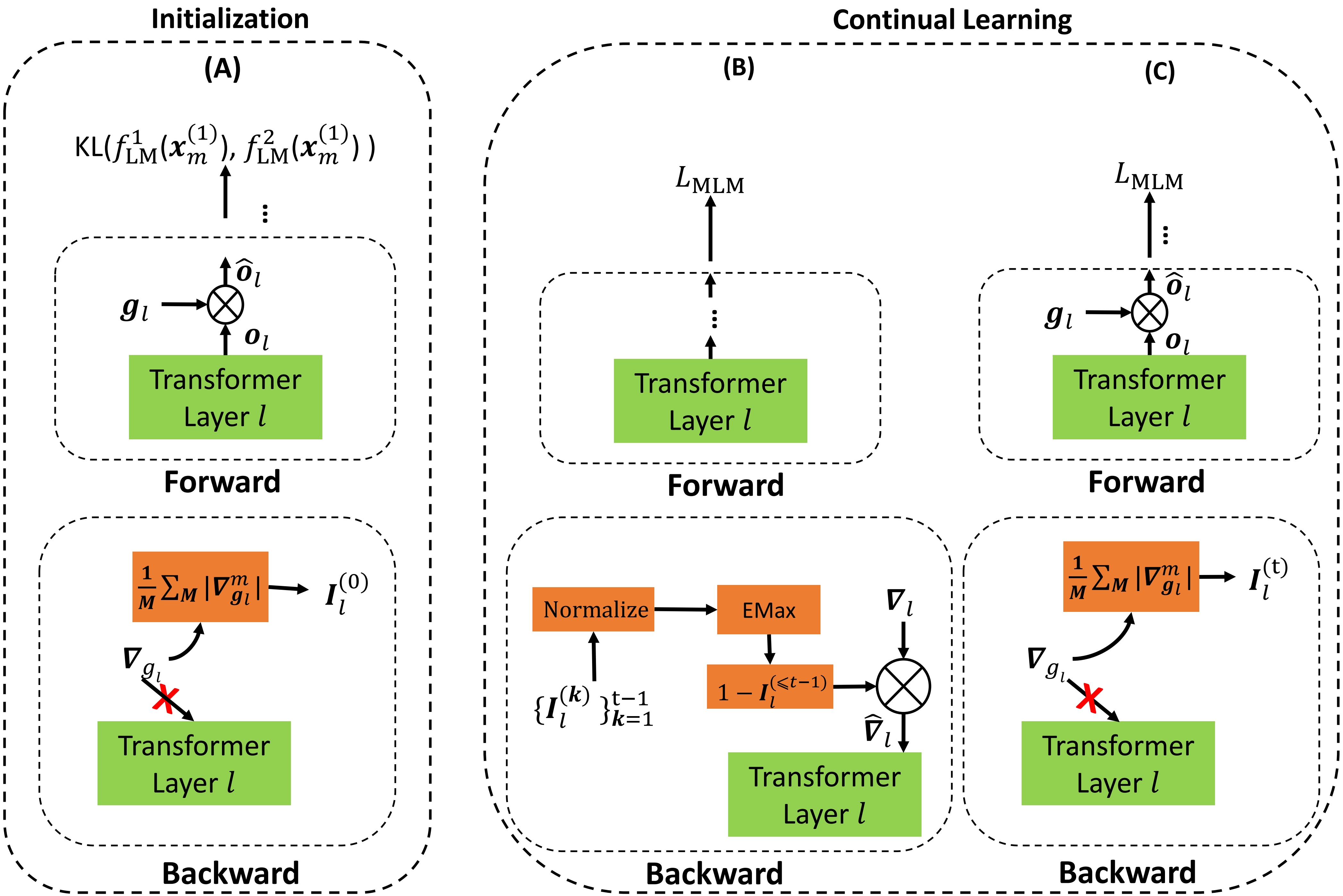
Next, we start **continual learning**



Initialization
Continual Learning
(A)
(B)
(C)
$\mathrm{KL}(f_{\mathrm{LM}}^{1}(\boldsymbol{x}_m^{(1)}), f_{\mathrm{LM}}^{2}(\boldsymbol{x}_m^{(1)}))$
$\hat{\boldsymbol{o}}_l$
$\boldsymbol{g}_l$
$\boldsymbol{o}_l$
Transformer Layer l
Forward
$\frac{1}{M}\sum_M |\nabla_{g_l}^m|$
$\boldsymbol{I}_l^{(0)}$
∇_{g_l}
Transformer Layer l
Backward
L_{MLM}
Transformer Layer l
Forward
Nothing changed in forward pass


Initialization
Continual Learning
(A)
(B)
(C)
KL($f_{\mathrm{LM}}^1(\boldsymbol{x}_m^{(1)})$, $f_{\mathrm{LM}}^2(\boldsymbol{x}_m^{(1)})$)
$\hat{\boldsymbol{o}}_l$
$\boldsymbol{g}_l$
$\boldsymbol{o}_l$
Transformer Layer l
Forward
$\frac{1}{M}\sum_M |\nabla_{g_l}^m|$
$I_l^{(0)}$
∇_{g_l}
Transformer Layer l
Backward
L_{MLM}
Transformer Layer l
Forward
Normalize
EMax
∇_l
$\{I_l^{(k)}\}_{k=1}^{t-1}$
$1 - I_l^{(\leqslant t-1)}$
$\hat{\nabla}_l$
Transformer Layer l
Backward
Accumulate all importance of units
Use it to soft-mask the gradient


Initialization
Continual Learning
(A)
(B)
(C)
$\mathrm{KL}(f_{\mathrm{LM}}^{1}(\boldsymbol{x}_m^{(1)}), f_{\mathrm{LM}}^{2}(\boldsymbol{x}_m^{(1)}))$
$\hat{\boldsymbol{o}}_l$
$\boldsymbol{g}_l$
$\boldsymbol{o}_l$
Transformer Layer l
Forward
$\frac{1}{M}\sum_M |\nabla_{\boldsymbol{g}_l}^{m}|$
$\boldsymbol{I}_l^{(0)}$
∇_{g_l}
Transformer Layer l
Backward
L_{MLM}
Transformer Layer l
Forward
Normalize
EMax
∇_l
$\{\boldsymbol{I}_l^{(k)}\}_{k=1}^{t-1}$
$1 - \boldsymbol{I}_l^{(\leqslant t-1)}$
$\hat{\nabla}_l$
Transformer Layer l
Backward
After training, we again compute the importance of units for the trained domain



Same as initialization, but L_{impt} is L_{MLM} instead of KL

$\boldsymbol{I}^{(t)}_l$ will be used in domain t+1

Evaluation

Goals

CF Prevention

Knowledge Transfer

Metrics

Forgetting Rate

Final Performance

Metrics

End-tasks of the domains

Restaurant

Domains that have post-trained

Restaurant

$A_{1,1}$

Metrics

End-tasks of the domains

Domains that have post-trained

	Restaurant	ACL
Restaurant	$A_{1,1}$	
ACL	$A_{2,1}$	$A_{2,2}$

Metrics

End-tasks of the domains

Domains that have post-trained

	Restaurant	ACL	AI
Restaurant	$A_{1,1}$		
ACL	$A_{2,1}$	$A_{2,2}$	
AI	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$

Metrics

End-tasks of the domains

Domains that have post-trained

	Restaurant	ACL	AI	Phone	PubMed	……
Restaurant	$A_{1,1}$					
ACL	$A_{2,1}$	$A_{2,2}$				
AI	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$			
Phone	$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$		
PubMed	$A_{5,1}$	$A_{5,2}$	$A_{5,3}$	$A_{5,4}$	$A_{5,5}$	
⋮						
	$A_{t,1}$	……				$A_{t,t}$

Final Performance

End-tasks of the domains

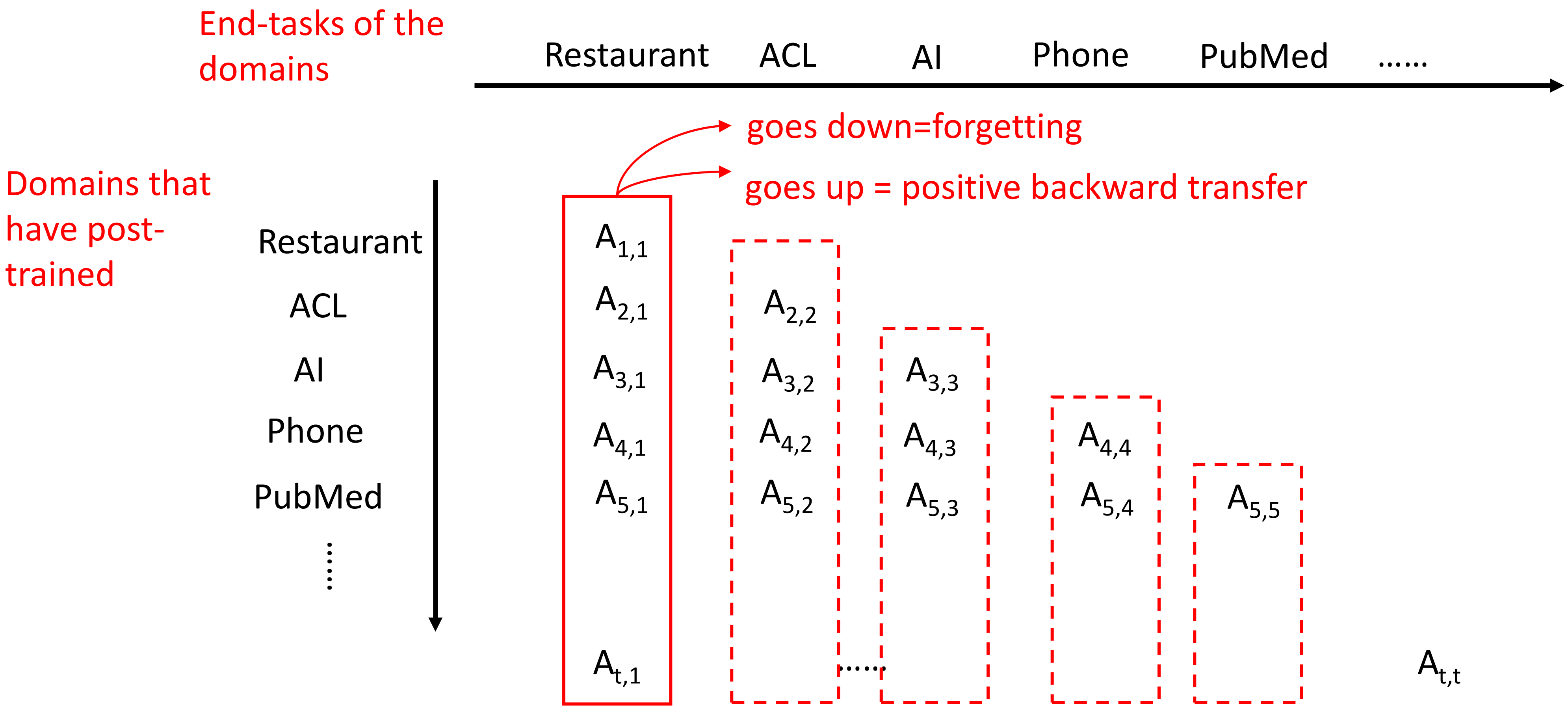
Domains that have post-trained

	Restaurant	ACL	AI	Phone	PubMed	……
Restaurant	$A_{1,1}$					
ACL	$A_{2,1}$	$A_{2,2}$				
AI	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$			
Phone	$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$		
PubMed	$A_{5,1}$	$A_{5,2}$	$A_{5,3}$	$A_{5,4}$	$A_{5,5}$	
⋮						
	$A_{t,1}$	……				$A_{t,t}$

Final Performance: $\frac{1}{T}\sum_{i=1}^{T} R_{t,i}$

The higher, the better. The most popular metric

Forgetting Rate



Forgetting Rate: $\frac{1}{T-1}\sum_{k=1}^{t-1} A_{k,k} - A_{t,k}$

The difference between a task **first learned** performance and its **final** performance
Positive=forgetting; **Negative**=positive backward transfer

Overall Performance

Non-Continual Learning

Without post-train (directly fine-tune the LM)

Individual post-training

	Restaurant	ACL	AI	Phone	PubMed	Camera	Average
Without post-train	79.81	66.11	60.98	**83.75**	72.38	78.82	73.64
Individual post-training	**80.84**	**68.75**	**68.97**	82.59	**72.84**	**84.39**	**76.4**

w/o Pre-trained < Individual Post-trained

This is not surprising, as post-training has been demonstrated to improve performance in the literature.

Overall Performance

Now we can look at continual learning

	Restaurant	ACL	AI	Phone	PubMed	Camera	Average
Without post-train (directly fine-tune the LM)	79.81	66.11	60.98	83.75	72.38	78.82	73.64
Individual post-training	**80.84**	68.75	68.97	82.59	**72.84**	84.39	76.4
Our continual post-training method (**CPS**)	80.34	**69.36**	**70.93**	**85.99**	72.8	**88.16**	**77.93**

w/o Pre-trained < Individual Post-trained < CPS

CPS is better than individual post-training
CPS can not only mitigate forgetting but also encourage knowledge transfer

Overall Performance

Continual Learning **v.s. CPS**

Forgetting Rate: $\frac{1}{T-1}\sum_{k=1}^{t-1} A_{k,k} - A_{t,k}$

		Restaurant	ACL	AI	Phone	PubMed	Camera	Average	Forgetting Rate
Non-Continual-learning	No post-train	79.81	66.11	60.98	83.75	72.38	78.82	73.64	---
	Individual post-train	80.84	68.75	68.97	82.59	72.84	84.39	76.4	---
Naïve continual learning (**NCL**): continual learning without any specific technique		79.52	68.39	67.94	84.1	72.49	85.71	76.36	1.14
CPS		80.34	**69.36**	**70.93**	**85.99**	**72.8**	**88.16**	**77.93**	**-1.09**

+ forgetting rate in NCL, indicates it does suffer from forgetting

- forgetting rate in CPS, indicating it has positive transfer

		Restaurant	ACL	AI	Phone	PubMed	Camera	Average	Forgetting Rate
Non-Continual-learning	No post-train	79.81	66.11	60.98	83.75	72.38	78.82	73.64	---
	Individual post-train	80.84	68.75	68.97	82.59	72.84	84.39	76.4	---
Naïve continual post-training		79.52	68.39	67.94	84.1	72.49	85.71	76.36	1.14
SoTA continual learning baselines	EWC	80.98	65.94	65.04	82.32	71.43	83.35	74.84	0.02
	DER++	79	67.2	63.96	83.22	72.58	87.1	75.51	2.36
	HAT	79.29	68.25	64.84	81.44	71.61	82.37	74.63	-0.23
	BCL	78.97	70.71	66.26	81.7	71.99	85.06	75.78	-0.06
	CPS	80.34	**69.36**	**70.93**	**85.99**	**72.8**	**88.16**	**77.93**	**-1.09**

✓ CPS outperforms SoTA

✗ Most of the SoTA only focus on mitigating forgetting, which is not enough

✗ Even replay-based method (DER++) is not good as post-training need much more replay data

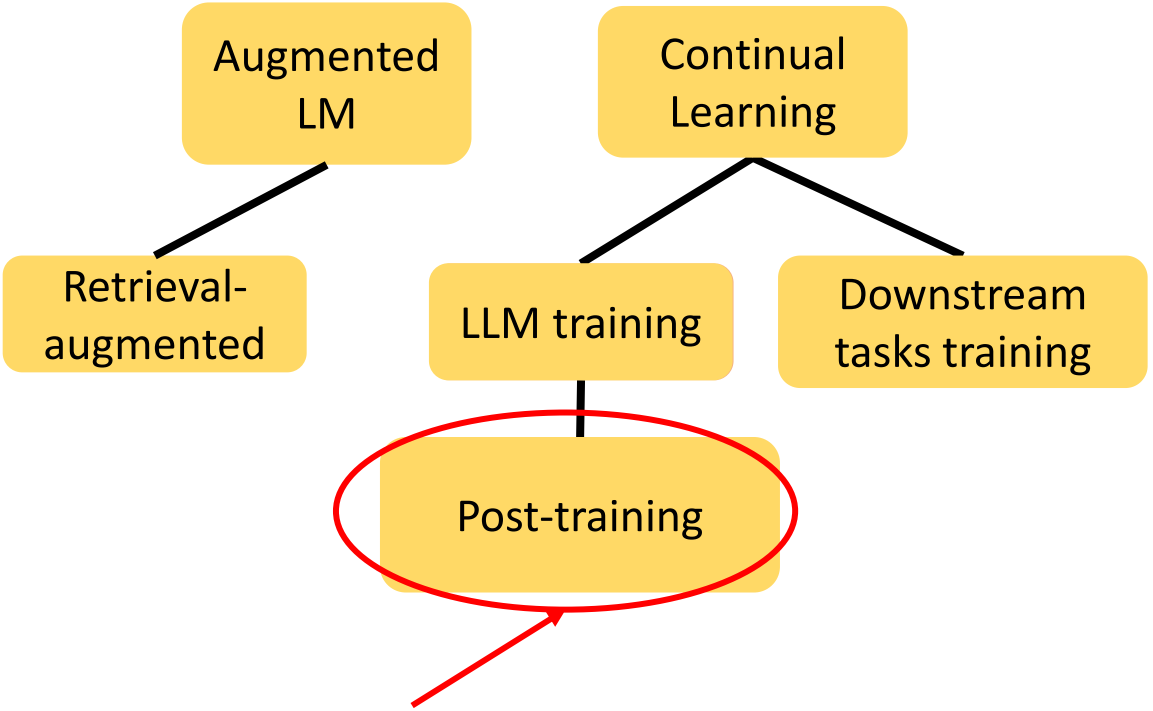
Continual Post-training of Language Model

- Computing **importance** of units for general and domain knowledge, with **different** L_{impt}

- **Soft-masking** the backward propagation based on importance (which help CF and KT)

Enhancing LLM for A Dynamic World

How to make knowledge in LLM more **reusable** and **updatable**?



Continual Post-training of Language Models, Ke et al., ICLR 2023

Enhancing LLM for A Dynamic World

Why it could be increasingly important?

Enhancing LLM for A Dynamic World

Why it could be increasingly important?

- The fixed world assumption is way too limited!

Over just a few months, ChatGPT went from correctly answering a simple math problem 98% of the time to just 2%, study finds

BY PAOLO CONFINO
July 19, 2023 at 6:29 PM CDT

How Is ChatGPT's Behavior Changing over Time?

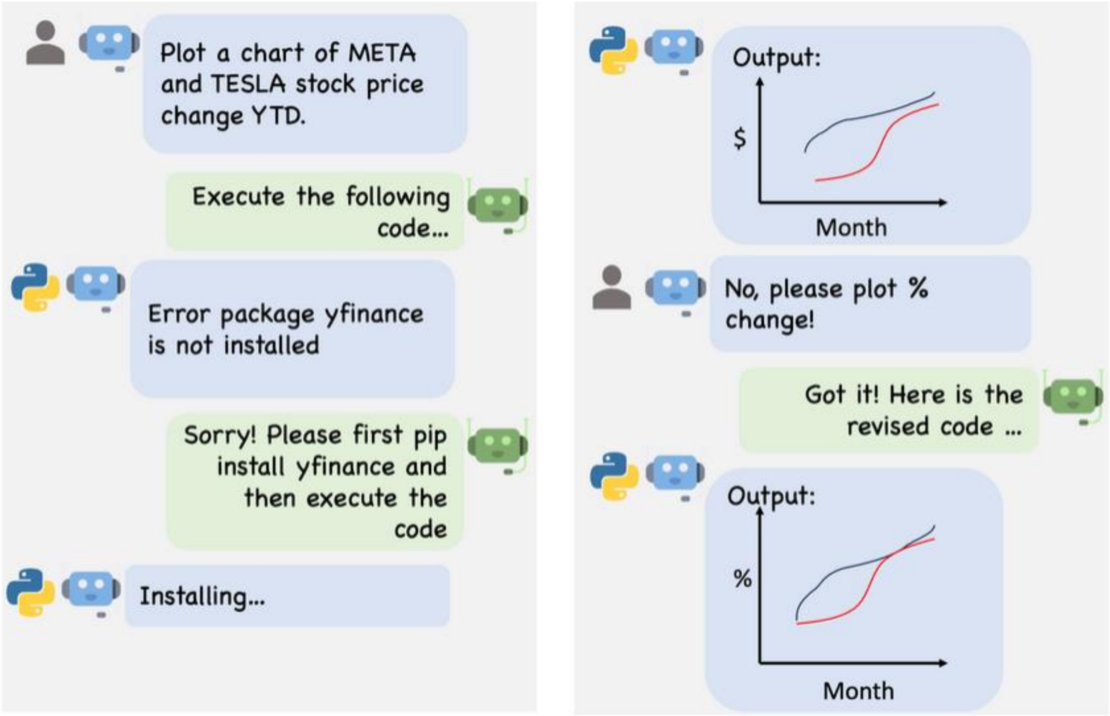
Lingjiao Chen†, Matei Zaharia‡, James Zou†

†Stanford University ‡UC Berkeley

Enhancing LLM for A Dynamic World

Why it could be increasingly important?

- The fixed world assumption is way too limited!
- LLMs are increasingly replacing/eliminating building blocks and memorizing more and more knowledge, **yet** these still depends on human efforts. A more ambitious goal is to make this **fully autonomous**, which require LLMs to **self-initiate** and **adapt to new circumstances**.

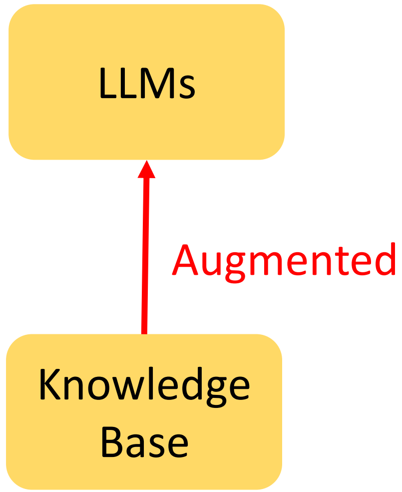


Enhancing LLM for A Dynamic World

Why it could be increasingly important?

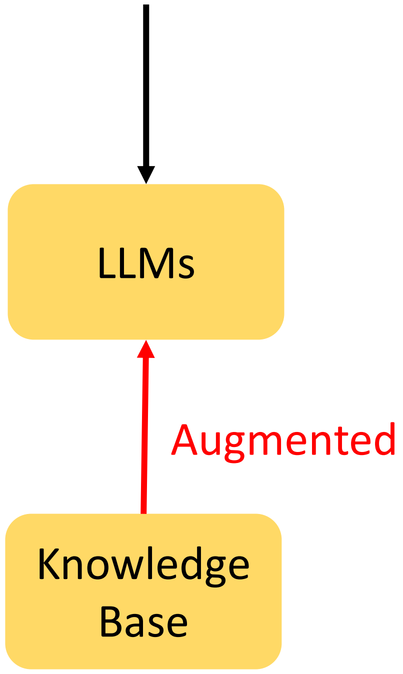
- The fixed world assumption is way too limited!
- LLMs are increasingly replacing/eliminating building blocks and memorizing more and more knowledge, these still depends on human efforts. A more ambitious vision is to make this **fully autonomous**, which require LLMs to **self-initiate** and **adapt to new circumstances**.
- It is still cutting-edge, and still, plenty of room to improve (see next!)

What research questions can lead us toward a more autonomous LLMs?



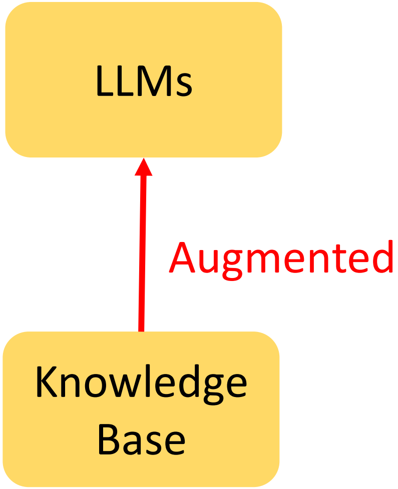
What research questions can lead us toward a more autonomous LLMs?

Complete the sentence in Trump's tone: "Between a wall and an egg that breaks it, I will always stand on the side of "



Retrieve Trump's speeches from the KB and augment the learner's working memory (context).

What research questions can lead us toward a more autonomous LLMs?

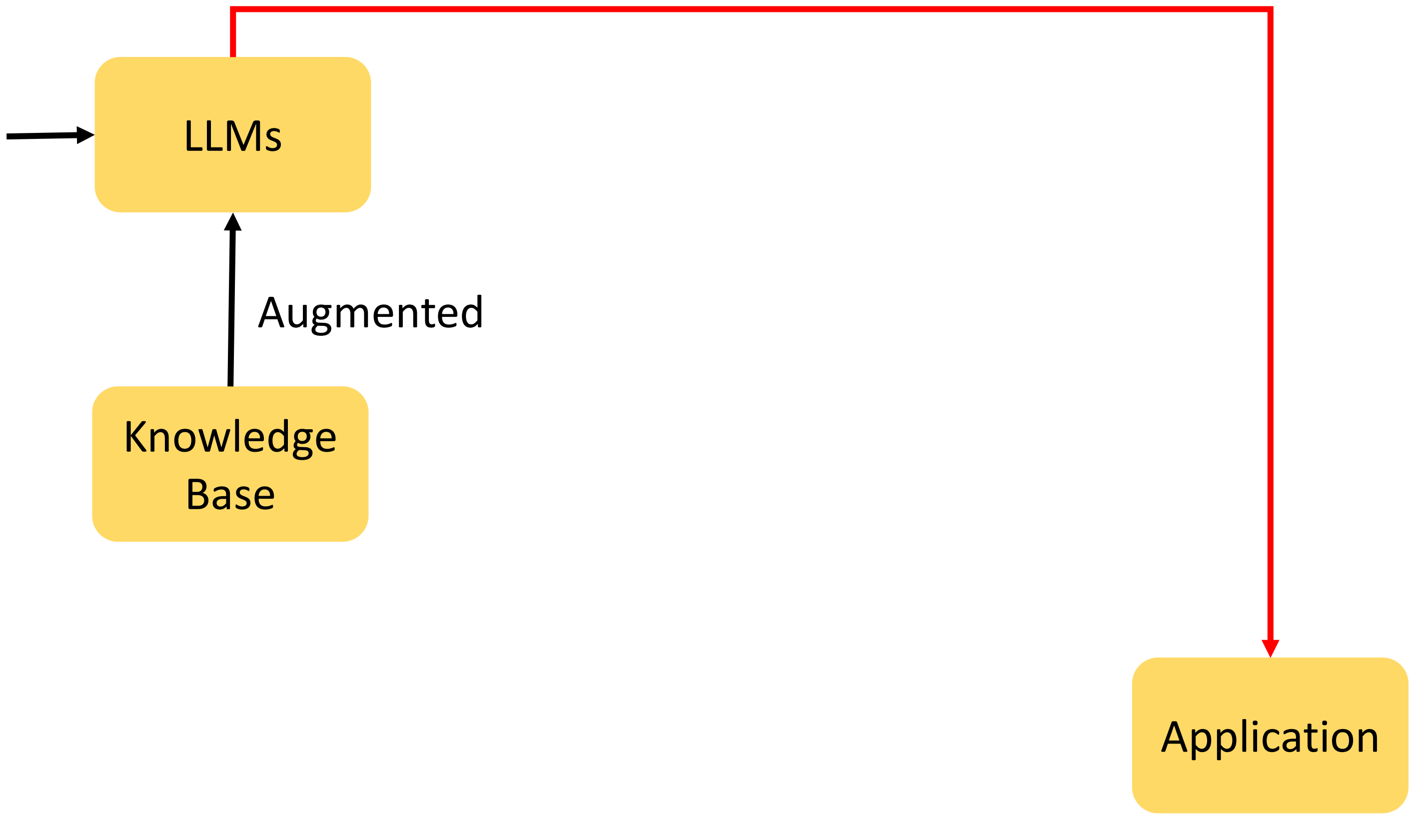


Research questions:

- What to retrieve (rerank/selection…)?
- How to better combine the retriever and LLM?
- When to use retrieval and when to update/use LLMs’ parameters?

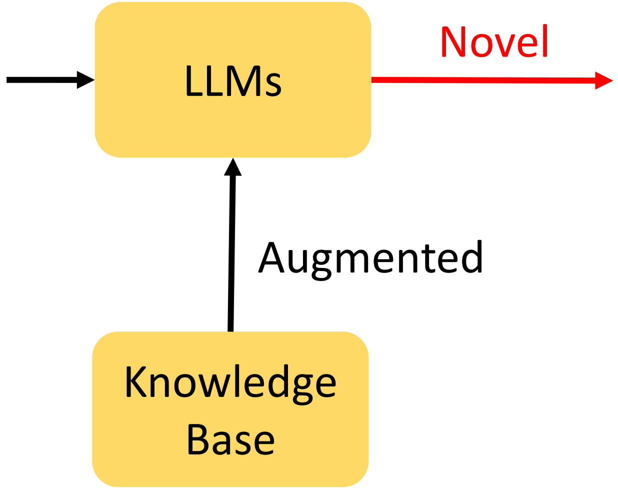
What research questions can lead us toward a more autonomous LLMs?

Complete the sentence in Trump's tone: "Between a wall and an egg that breaks it, I will always stand on the side of "



What research questions can lead us toward a more autonomous LLMs?

Complete the sentence in ***Zixuan's*** *tone: "Between a wall and an egg that breaks it, I will always stand on the side of "*

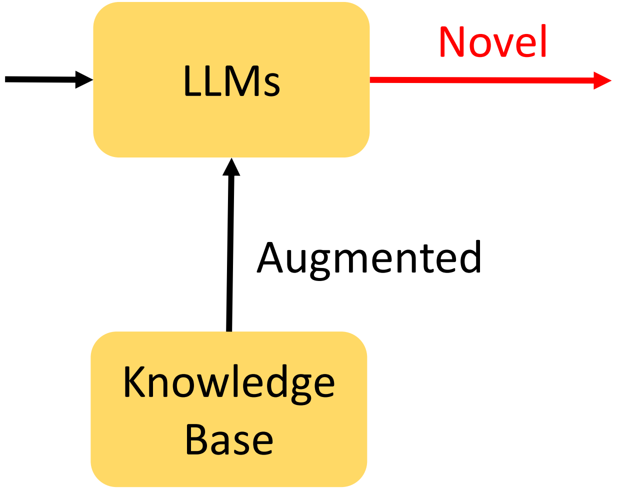


In some instances, there could be “novelty”

Novelty/Unknown/Unexpected/Unclear: anything that the LLM does not fully understand in order to accomplish the task

What research questions can lead us toward a more autonomous LLMs?

<u>Complete the sentence in Trump's tone</u>:
"Between a wall and an egg that breaks it, I will always stand on the side of "

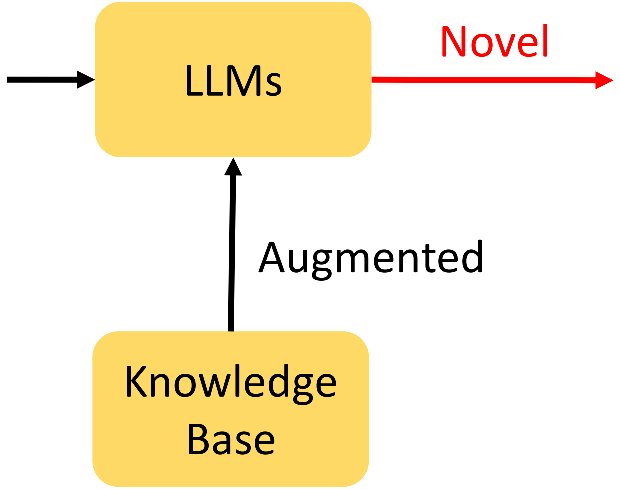


It could also be unclear to the LLM what '*in one's tone*' means or what aspects should be focused on.

Novelty/Unknown/Unexpected/Unclear: anything that the LLM does not fully understand in order to accomplish the task

What research questions can lead us toward a more autonomous LLMs?

*Complete the sentence in **Zixuan's** tone: "Between a wall and an egg that breaks it, I will always stand on the side of "*



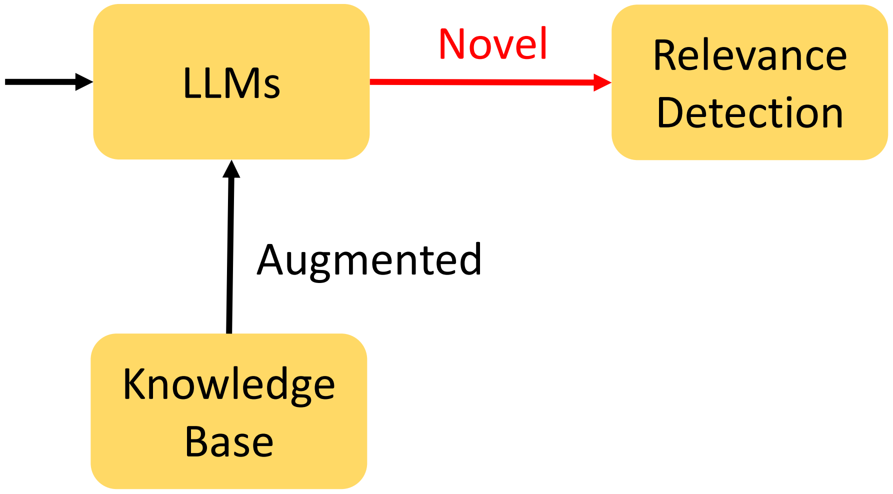
Research question:

How to detect novelty (knowledge that LLM does not already know)?

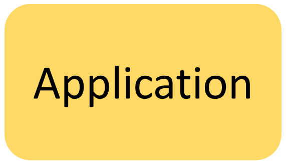
Application

What research questions can lead us toward a more autonomous LLMs?

Hello, my name is ***<u>Vincent Bing.</u>*** *Please complete the sentence in* ***<u>Zixuan's</u>*** *tone: "Between a wall and an egg that breaks it, I will always stand on the side of "*

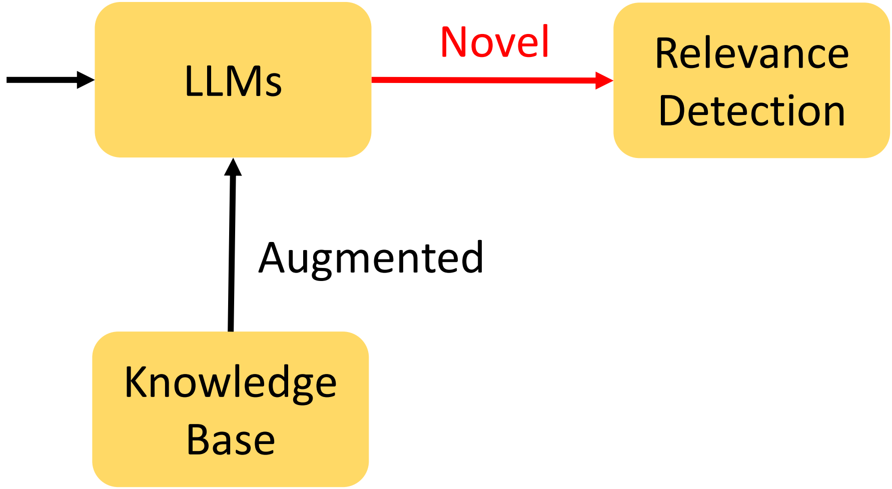


It is possible that the novelty occurs but is not related to the application.



What research questions can lead us toward a more autonomous LLMs?

<u>Hello, my name is Bing.</u>
Please complete the sentence in ***<u>Zixuan's</u>*** *tone: "Between a wall and an egg that breaks it, I will always stand on the side of "*



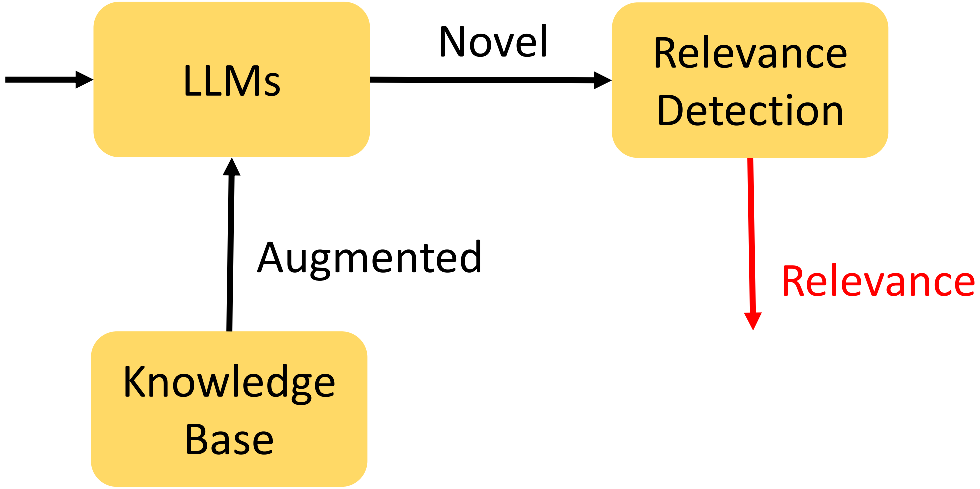
Research question:

How can we determine if the novelty is relevant to the final application? (there could be some noise!)

Application

What research questions can lead us toward a more autonomous LLMs?

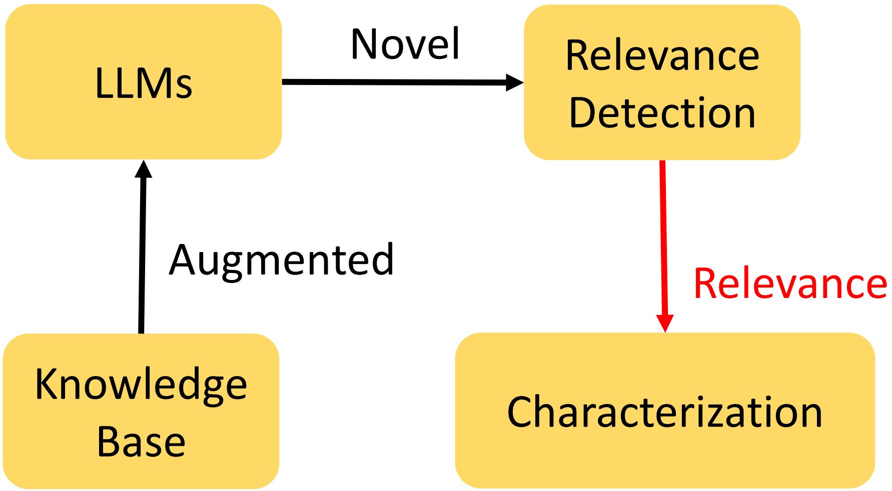
Complete the sentence in ***<u>Zixuan's</u>*** *tone: "Between a wall and an egg that breaks it, I will always stand on the side of "*



We still need to know that it is the part "Zixuan" that the LLM does not know

Application

What research questions can lead us toward a more autonomous LLMs?



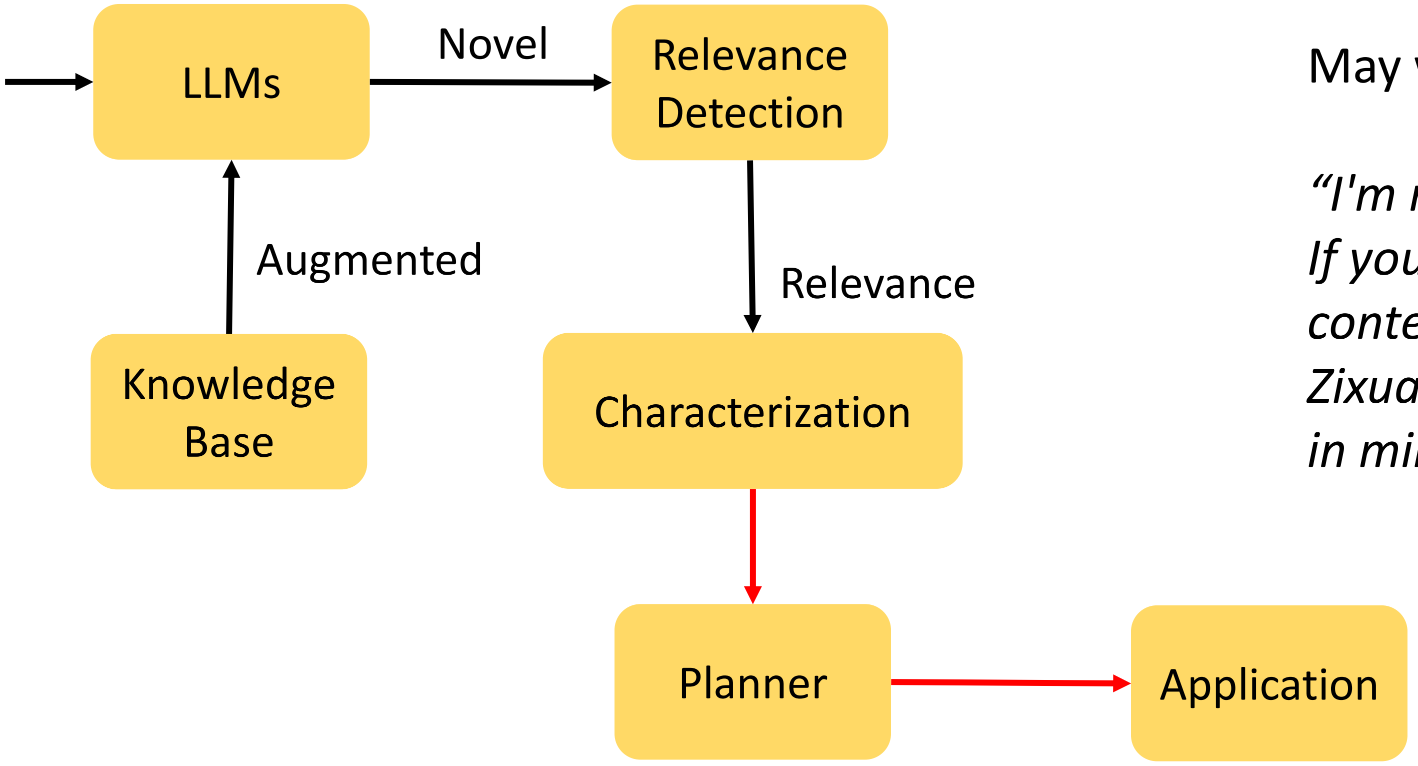
Research question:

How to specify parts with novel knowledge?

Application

What research questions can lead us toward a more autonomous LLMs?

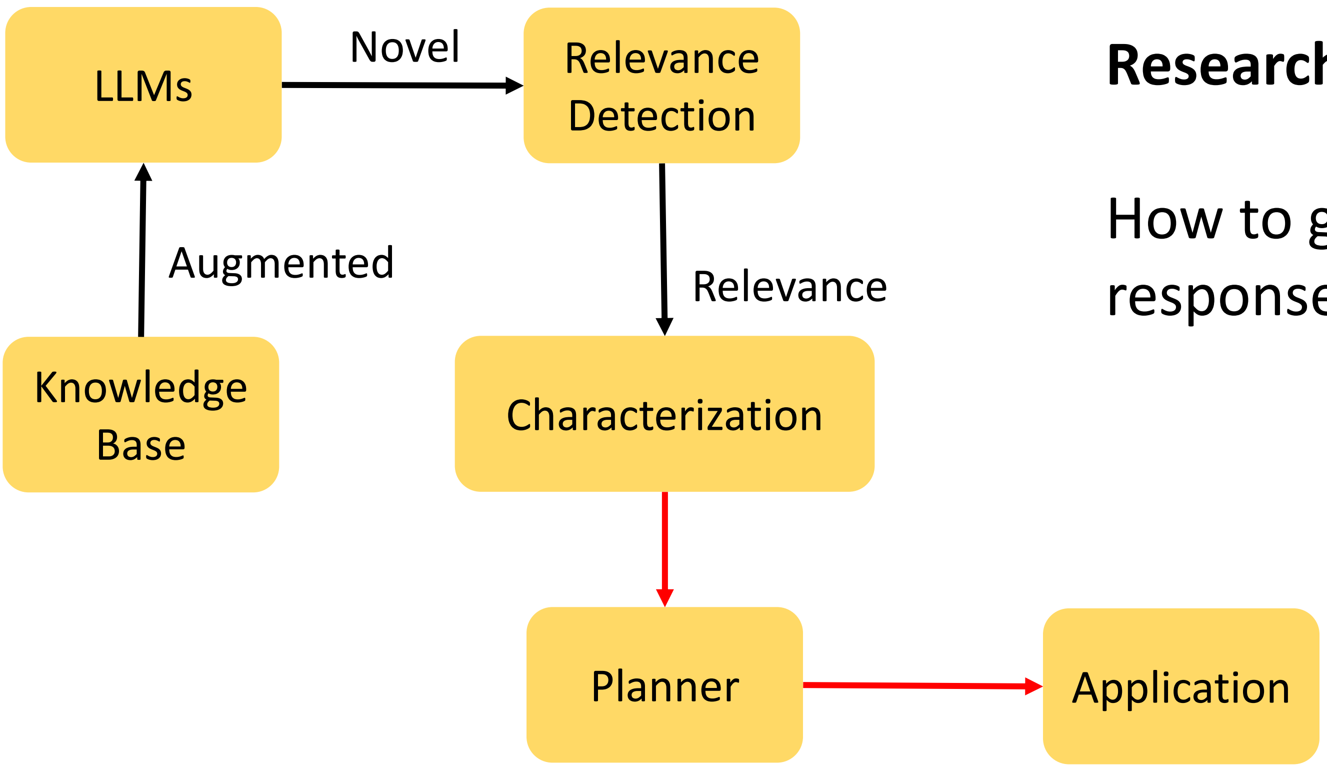
Complete the sentence in ***<u>Zixuan's</u>*** *tone: "Between a wall and an egg that breaks it, I will always stand on the side of "*



May want to response with:

"I'm not familiar with "Zixuan". If you could provide more context or details about who Zixuan is or the style you have in mind, I will be happy to help

What research questions can lead us toward a more autonomous LLMs?

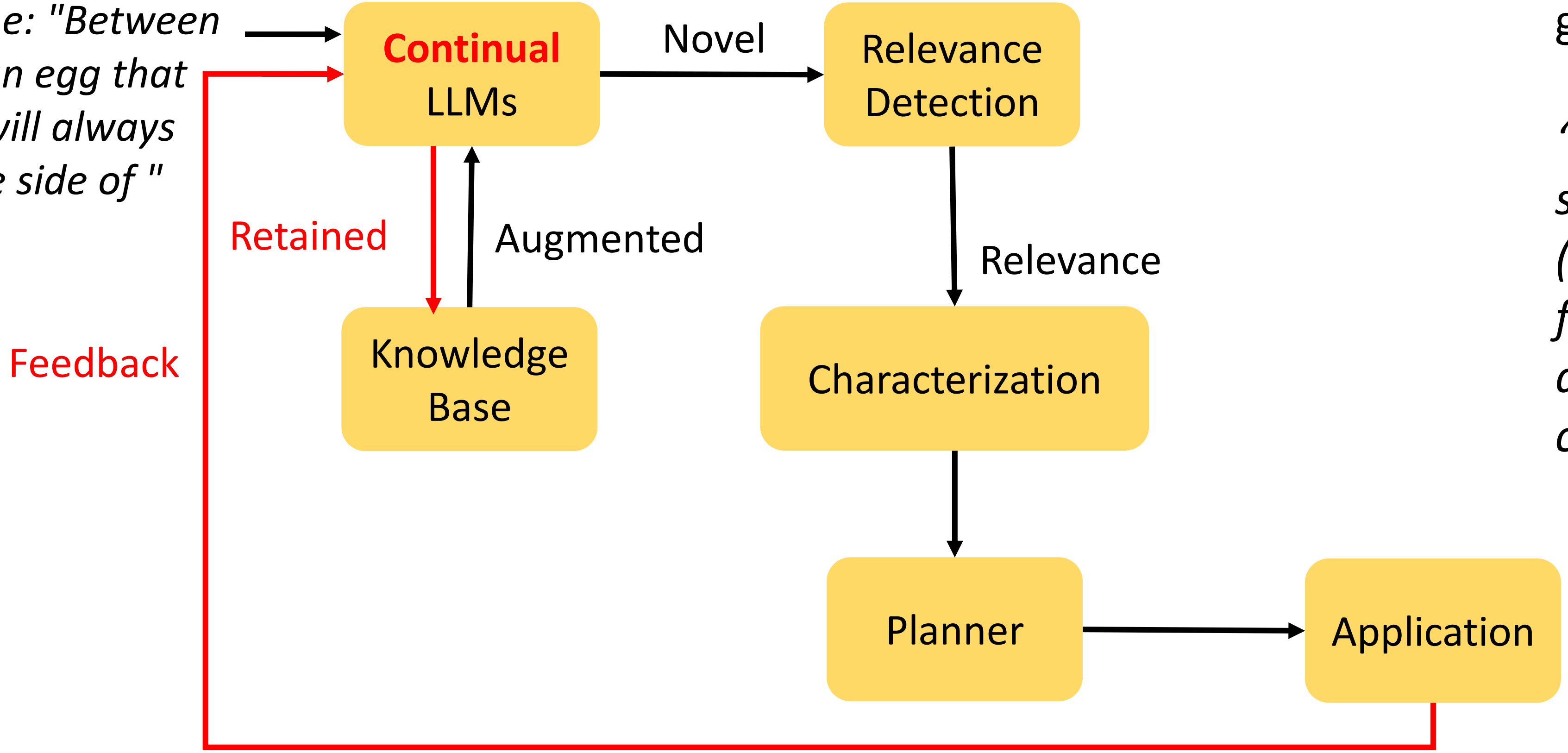


Research question:

How to generate a good response given the novel part?

What research questions can lead us toward a more autonomous LLMs?

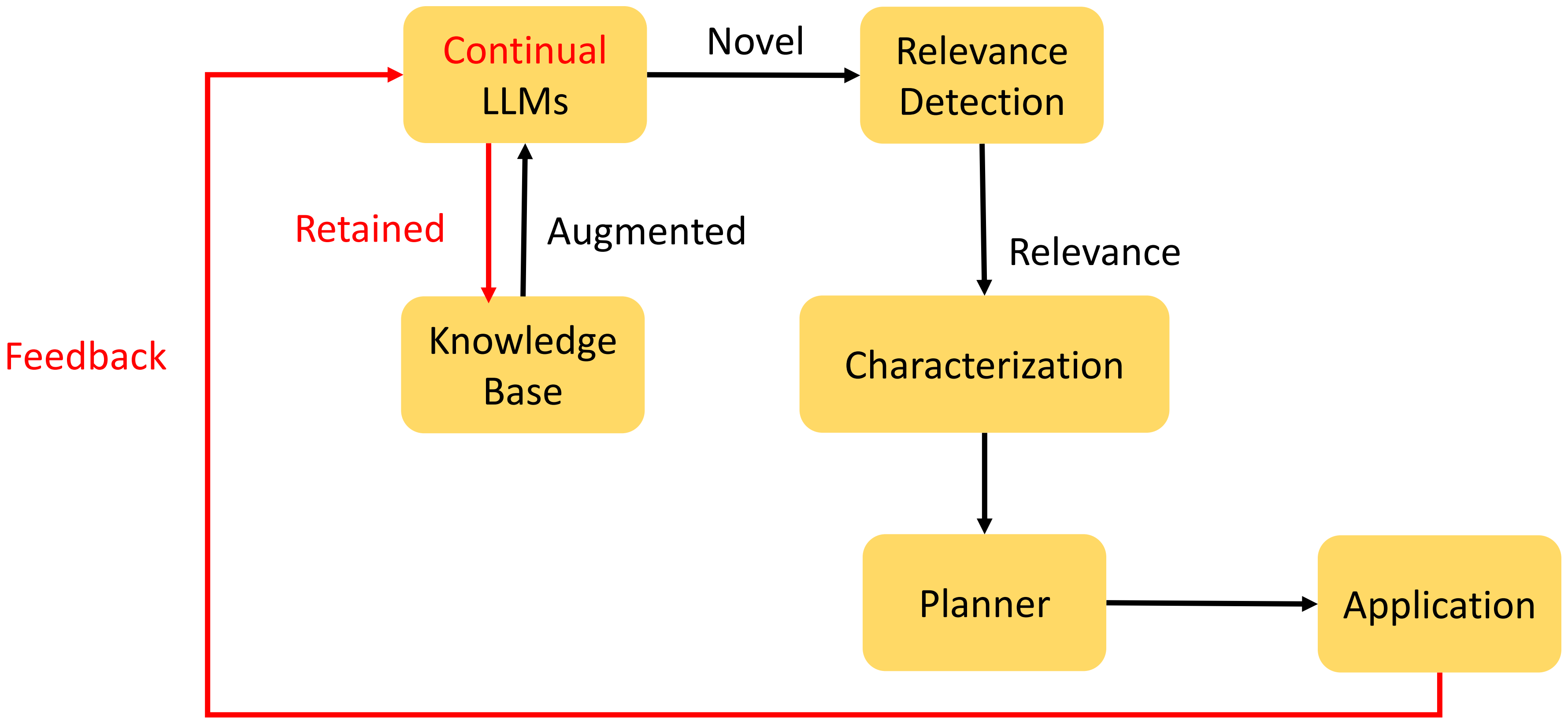
*Complete the sentence in **<u>Zixuan's</u>** tone: "Between a wall and an egg that breaks it, I will always stand on the side of "*



User/another agent may give feedback:

"Zixuan is a final-year PhD student working on LLM (pretrain, post-train, frontiers like retrieval-augmented LLM) and continual learning...."

What research questions can lead us toward a more autonomous LLMs?

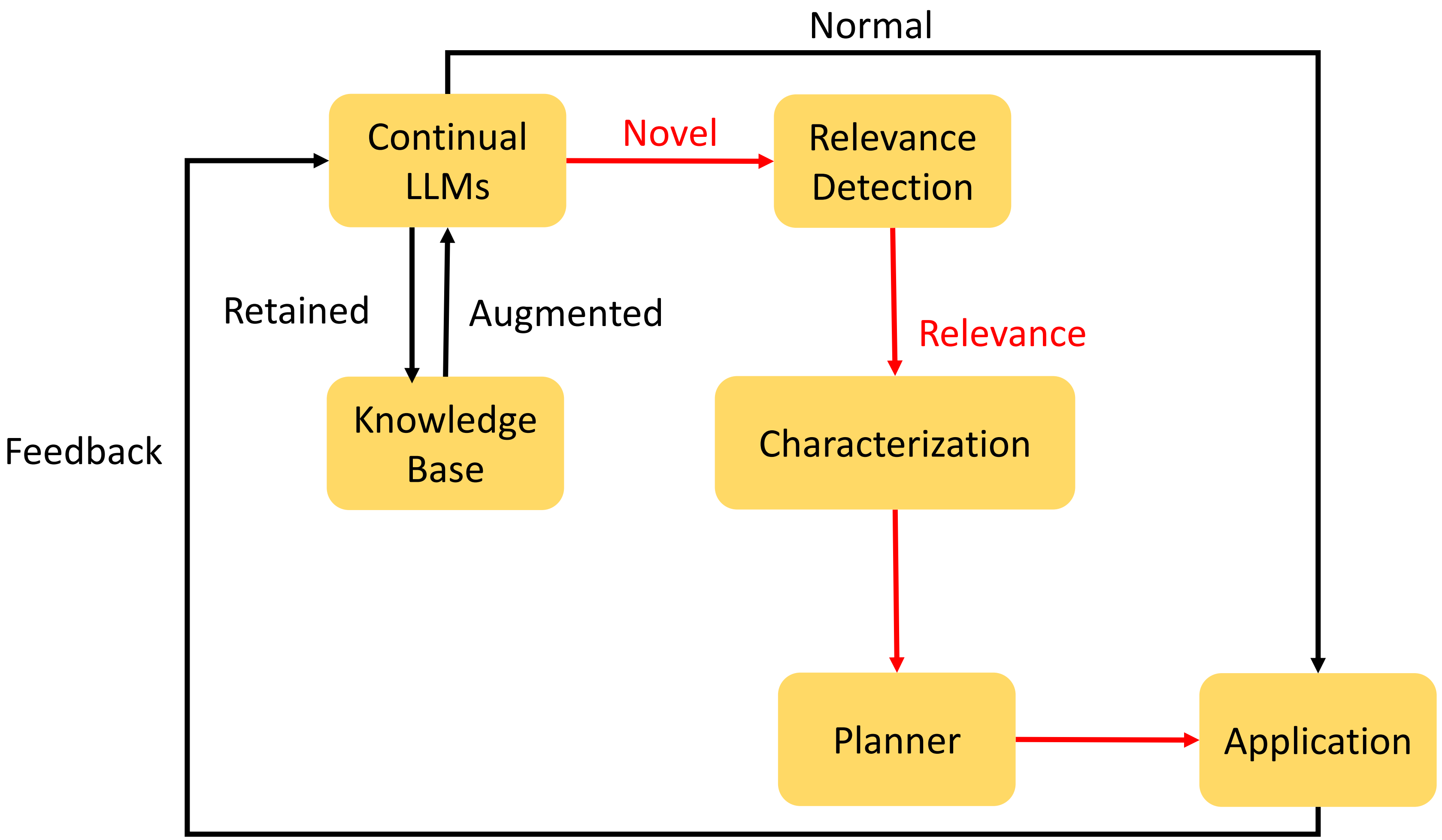


Research question:

How to use the feedback to continually update the LLM and retain the useful knowledge?

Using external memory, working memory (context), or updating the LLM?

What research questions can lead us toward a more autonomous LLMs?

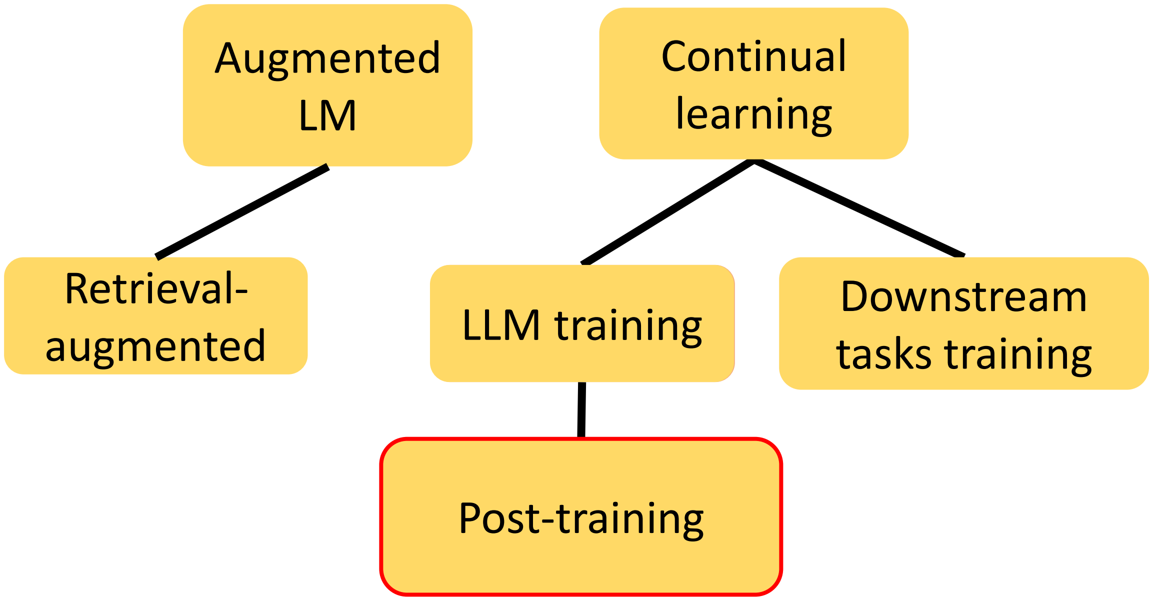


Most existing works are dedicated to the black part, which includes active research areas like retrieval-augmented LLM and continual learning.

The other component remains largely unexplored!

Enhancing LLM for A Dynamic World

How to make knowledge in LLM more **reusable** and **updatable**?



Ambitious goal: Fully autonomous LLM

Thank you