KDD2023 Research Track

Task Relation-aware Continual User Representation Learning

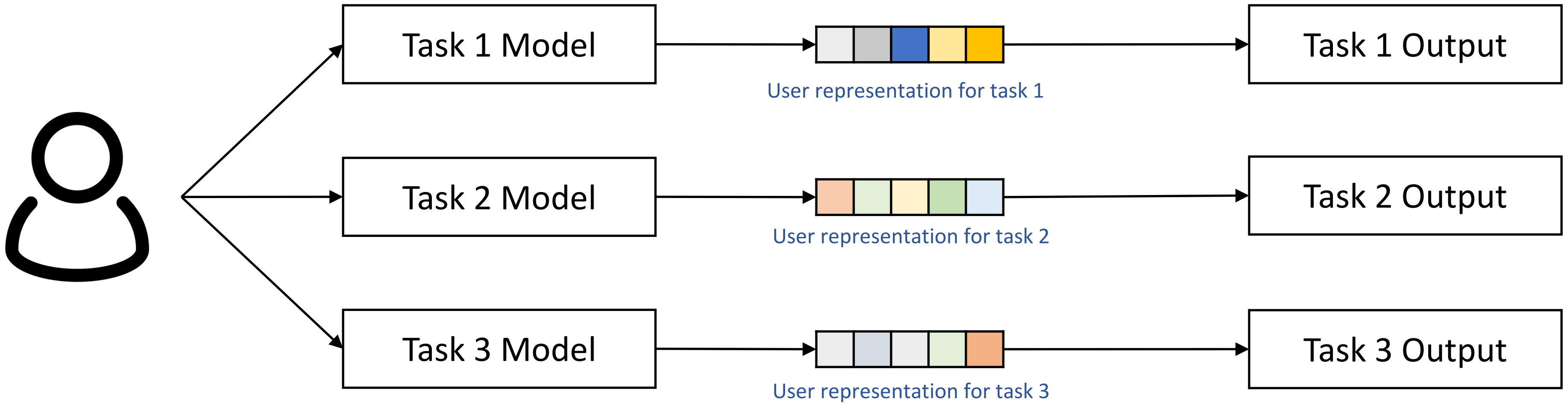
**Sein Kim, Namkyeong Lee,
Donghyun Kim, Minchul Yang, Chanyoung Park**

Korean Advanced Institute of Science and Technology (KAIST)
NAVER Corporation

DSAIL Data Science & Artificial Intelligence

Research Background Problems on User Modeling

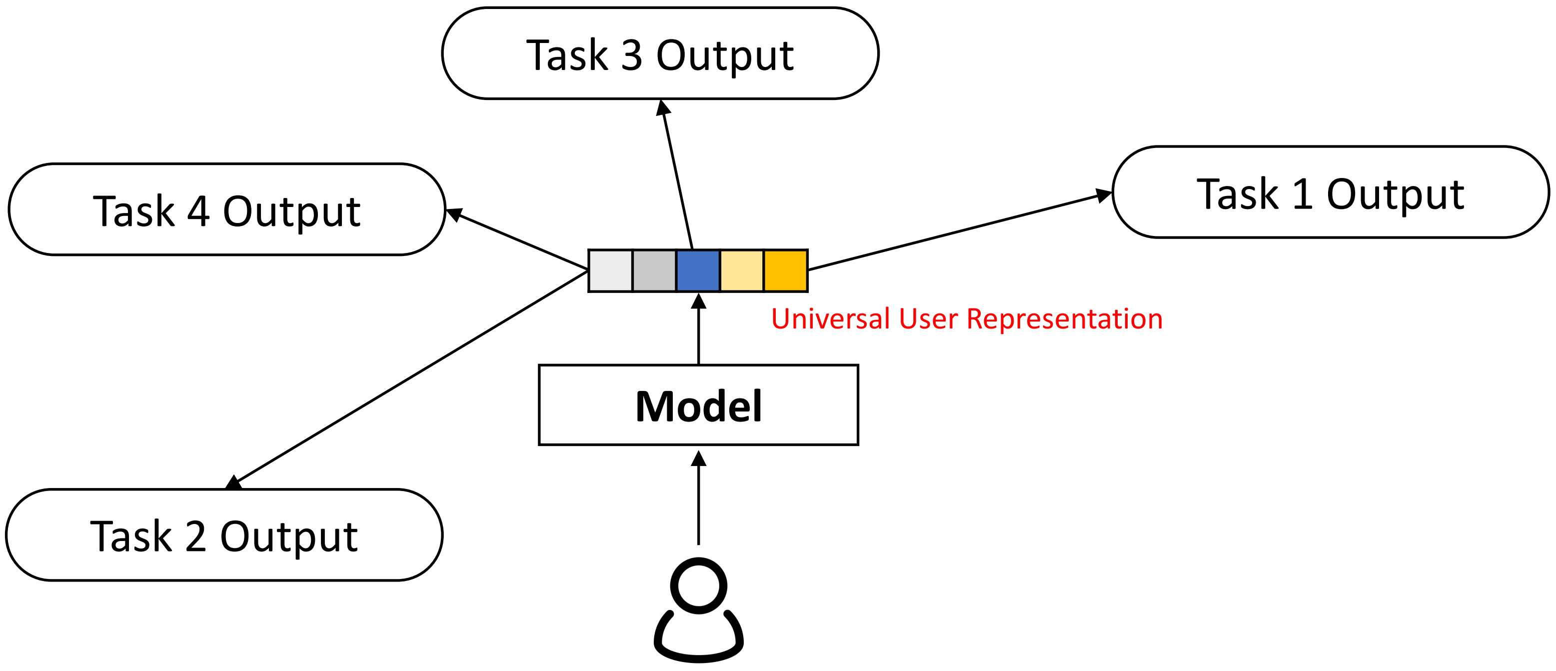
- **Inefficient**: “**Create**” and “**Train**” new models for **each new task**
- **Loss of positive transfer**: **disregard inter-task relationships** and hinder potential positive transfer



<Example of model operation for each task>

Research Background Problems on User Modeling

- **Inefficient**: "**Create**" and "**Train**" new models for **each new task**
- **Loss of positive transfer**: **disregard inter-task relationships** and hinder potential positive transfer



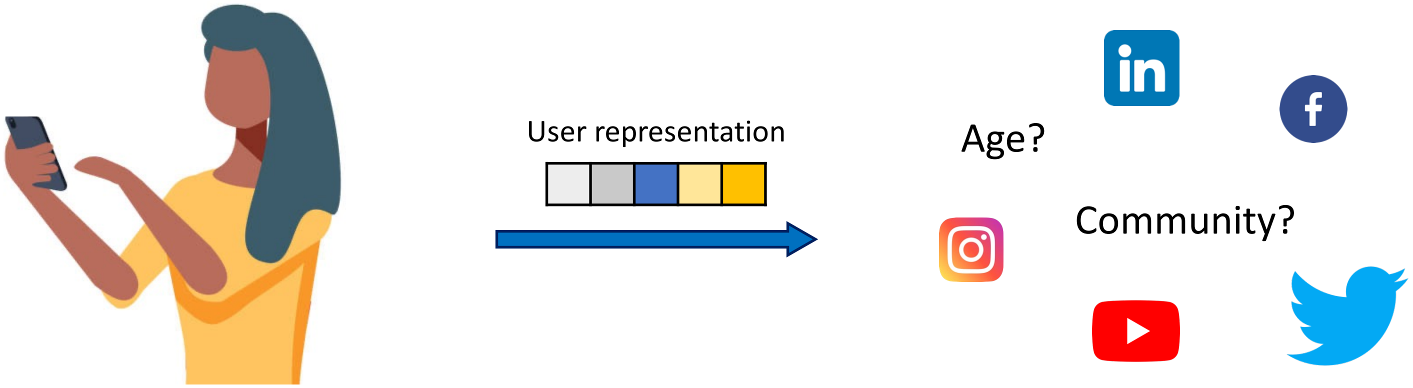
<Research Objective>

Research Objective: Solve various tasks through a **Universal user representation**
Maintain competitive performance across tasks using single universal user representation

Universal User Representation

Universal User Representation?

- A **single** user representation that can be **utilized for various tasks**
- Universal user representation should contain “**general**” and “**representative**” information that can perform well in various tasks



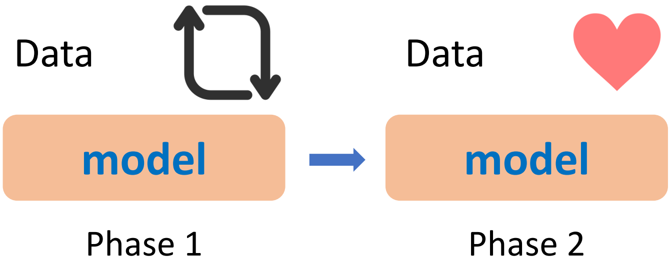
Universal User Representation Previous Works

Universal User Representation?

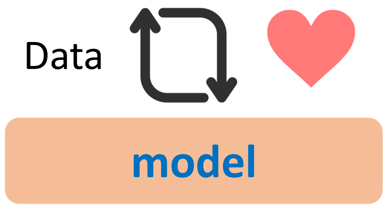
- A **single** user representation that can be **utilized for various tasks**
- Universal user representation should contain “**general**” and “**representative**” information that can perform well in various tasks

Previous studies have been focused on **Transfer** Learning / **Multi-task** Learning

Transfer Learning (TL)

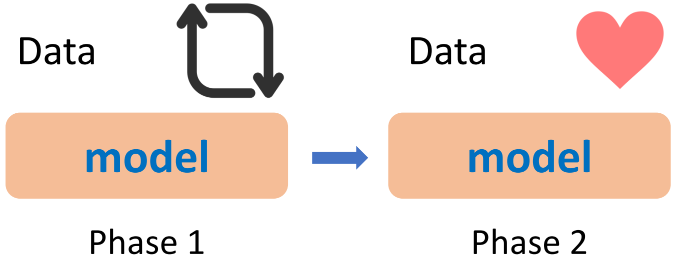


Multi-task Learning (MTL)



Universal User Representation Previous approaches

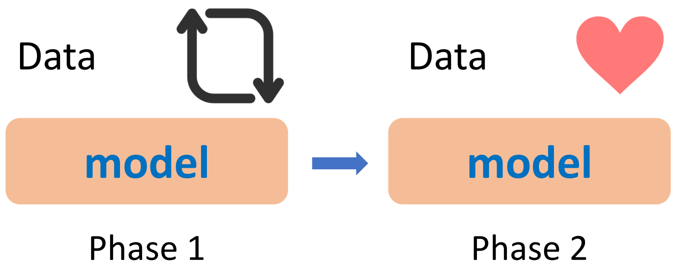
Transfer Learning (TL)



Only applicable when a **pair of tasks** (source and target) is given
→ **Different models** are required for **each (target) task**

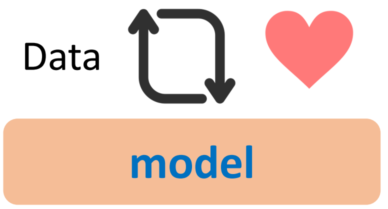
Universal User Representation Previous approaches

Transfer Learning (TL)



Only applicable when a **pair of tasks** (source and target) is given
→ **Different models** are required for **each (target) task**

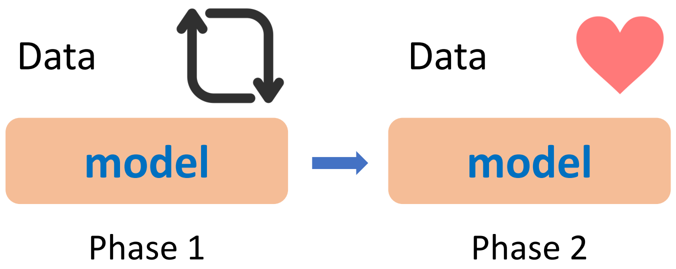
Multi-task Learning (MTL)



Requires **all the tasks** and their associated **data** to be **available** in advance
→ The model should be **retrained** with **all the data across tasks** to train **new service (task)**

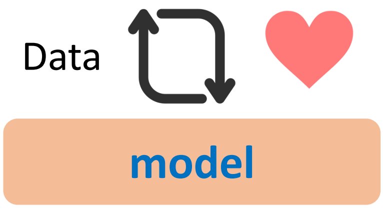
Universal User Representation Previous approaches

Transfer Learning (TL)



Only applicable when a **pair of tasks** (source and target) is given
→ **Different models** are required for **each (target) task**

Multi-task Learning (MTL)



Requires **all the tasks** and their associated **data** to be **available** in advance
→ The model should be **retrained** with **all the data across tasks** to train **new service (task)**

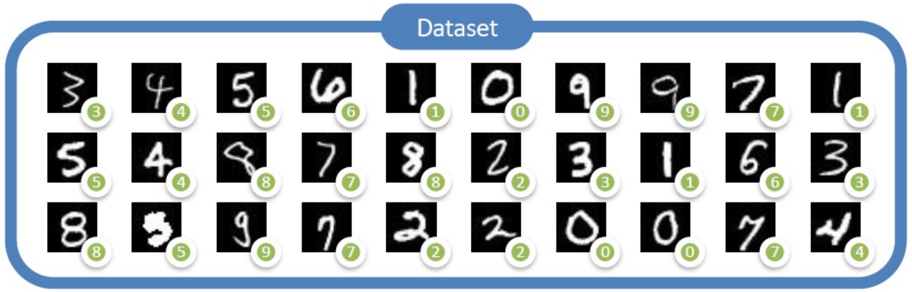
Both learning methods necessitate **Large Scale Datasets to exist simultaneously**
In domains with **continuous** influx of **new users** and launching of **new services**, TL & MTL may **not be suitable**

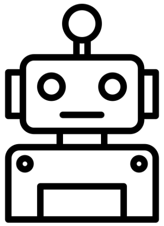
Prompting the need for a new learning approach

Continual Learning

Continual Learning?

The method of **sequentially learning** new knowledge in a **single model** while handling **multiple tasks**.
The key aspect here is to **maintain** the **knowledge** acquired from **previous tasks**!





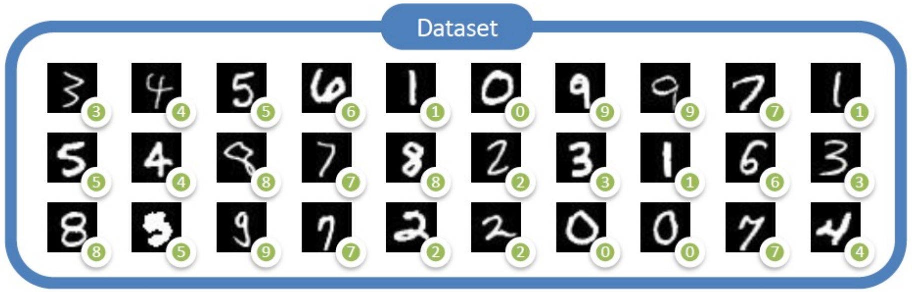
General Machine Learning

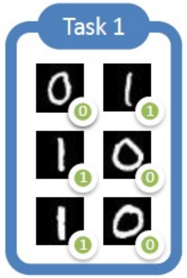
In traditional Machine Learning, it assumes i.i.d. samples from a fixed data distribution.

Continual Learning

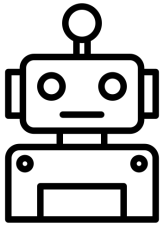
Continual Learning?

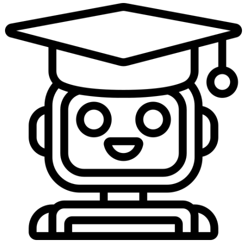
The method of **sequentially learning** new knowledge in a **single model** while handling **multiple tasks**.
The key aspect here is to **maintain** the **knowledge** acquired from **previous tasks**!





Train





General Machine Learning

In traditional Machine Learning, it assumes i.i.d. samples from a fixed data distribution.

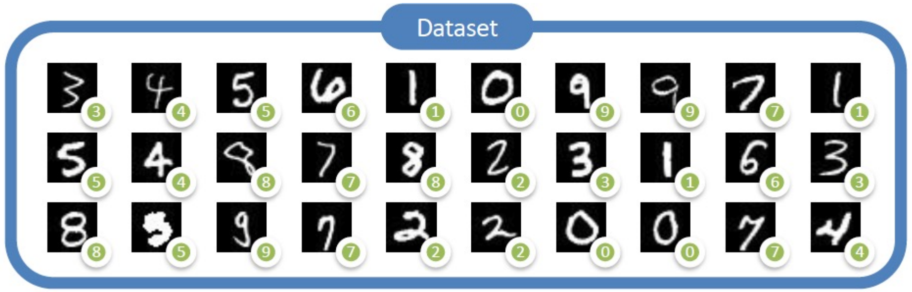
Continual Learning

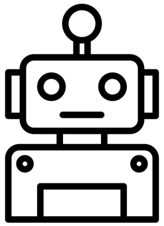
Non i.i.d. stream

Continual Learning

Continual Learning?

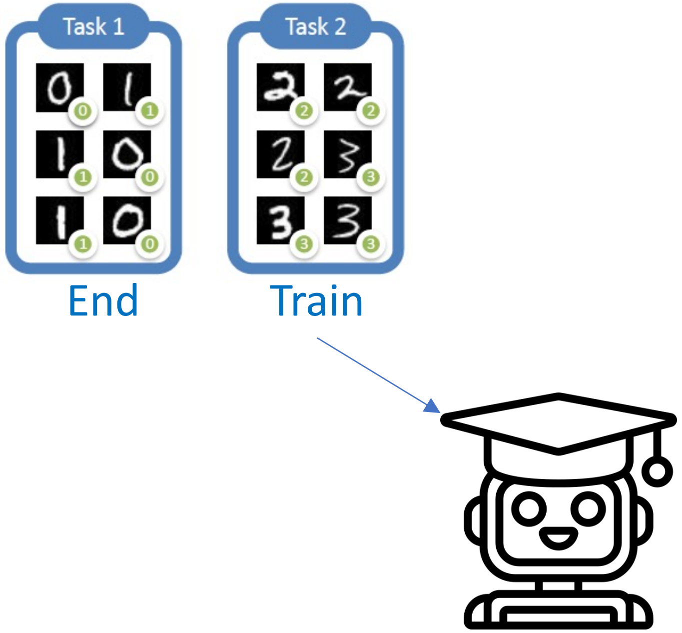
The method of **sequentially learning** new knowledge in a **single model** while handling **multiple tasks**.
The key aspect here is to **maintain** the **knowledge** acquired from **previous tasks**!





General Machine Learning

In traditional Machine Learning, it assumes i.i.d. samples from a fixed data distribution.



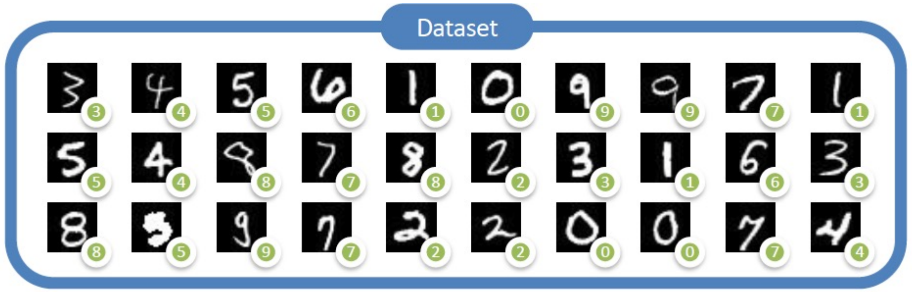
Continual Learning

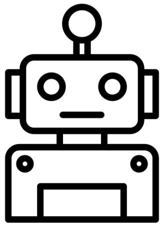
Non i.i.d. stream

Continual Learning

Continual Learning?

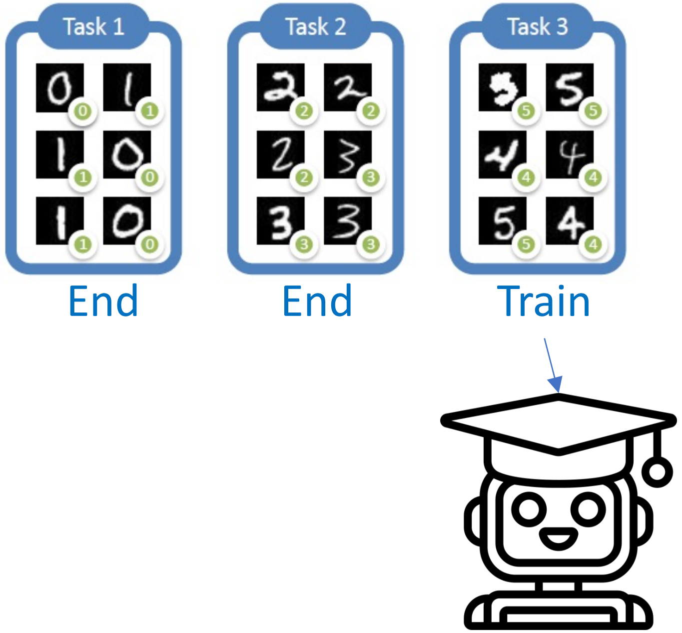
The method of **sequentially learning** new knowledge in a **single model** while handling **multiple tasks**.
The key aspect here is to **maintain** the **knowledge** acquired from **previous tasks**!





General Machine Learning

In traditional Machine Learning, it assumes i.i.d. samples from a fixed data distribution.



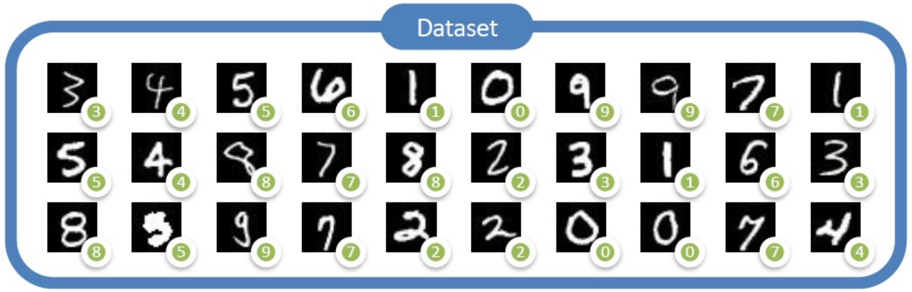
Continual Learning

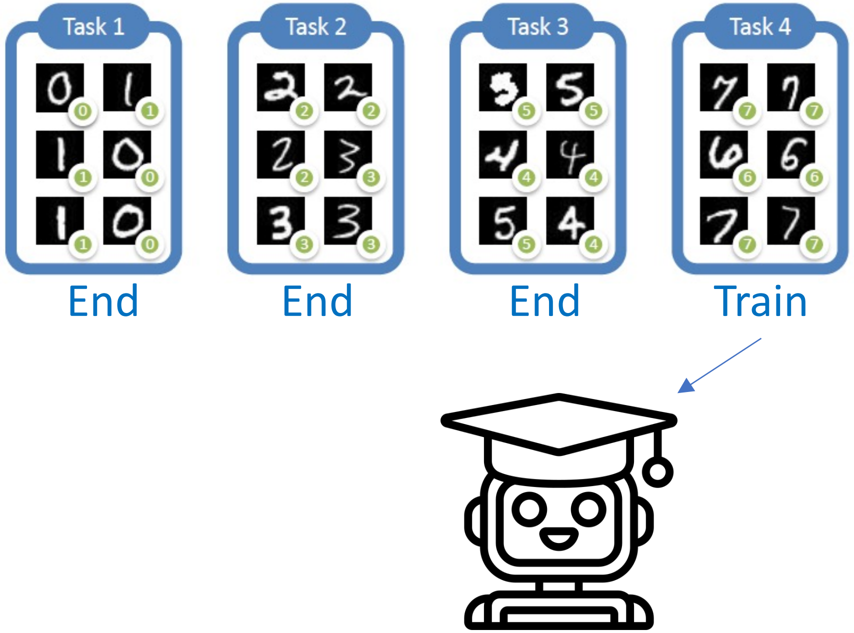
Non i.i.d. stream

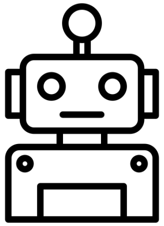
Continual Learning

Continual Learning?

The method of **sequentially learning** new knowledge in a **single model** while handling **multiple tasks**.
The key aspect here is to **maintain** the **knowledge** acquired from **previous tasks**!







General Machine Learning

In traditional Machine Learning, it assumes i.i.d. samples from a fixed data distribution.

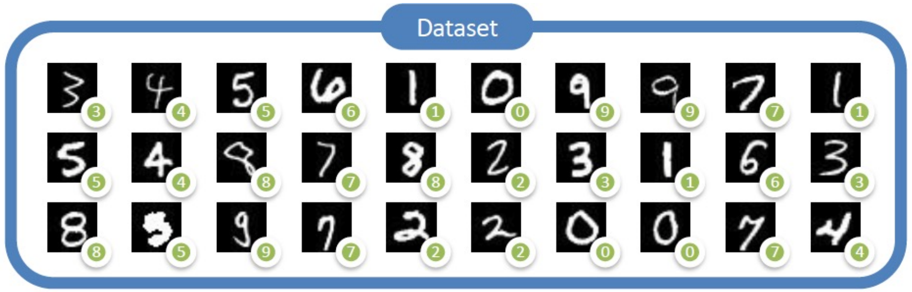
Continual Learning

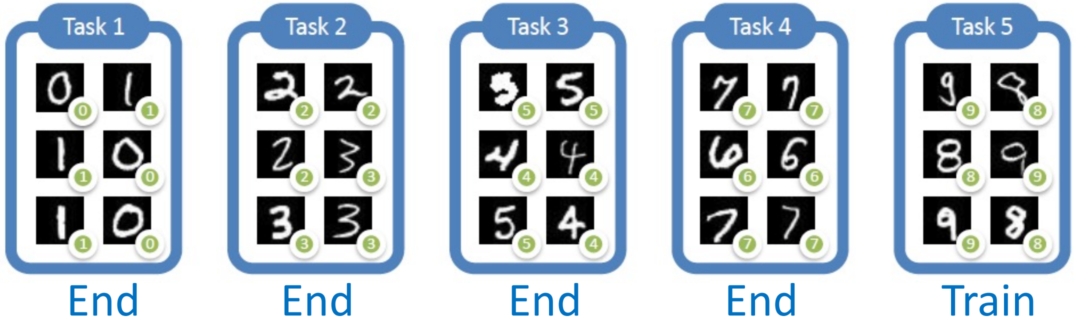
Non i.i.d. stream

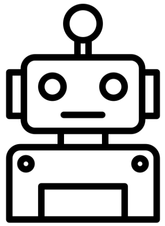
Continual Learning

Continual Learning?

The method of **sequentially learning** new knowledge in a **single model** while handling **multiple tasks**.
The key aspect here is to **maintain** the **knowledge** acquired from **previous tasks**!

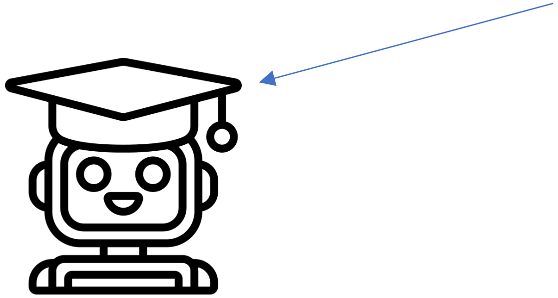






General Machine Learning

In traditional Machine Learning, it assumes i.i.d. samples from a fixed data distribution.



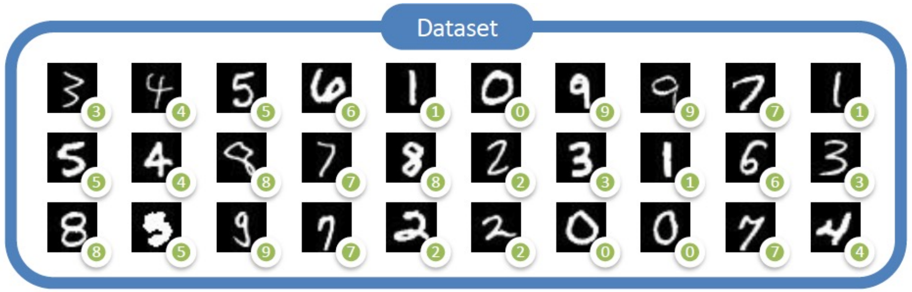
Continual Learning

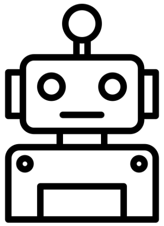
Non i.i.d. stream

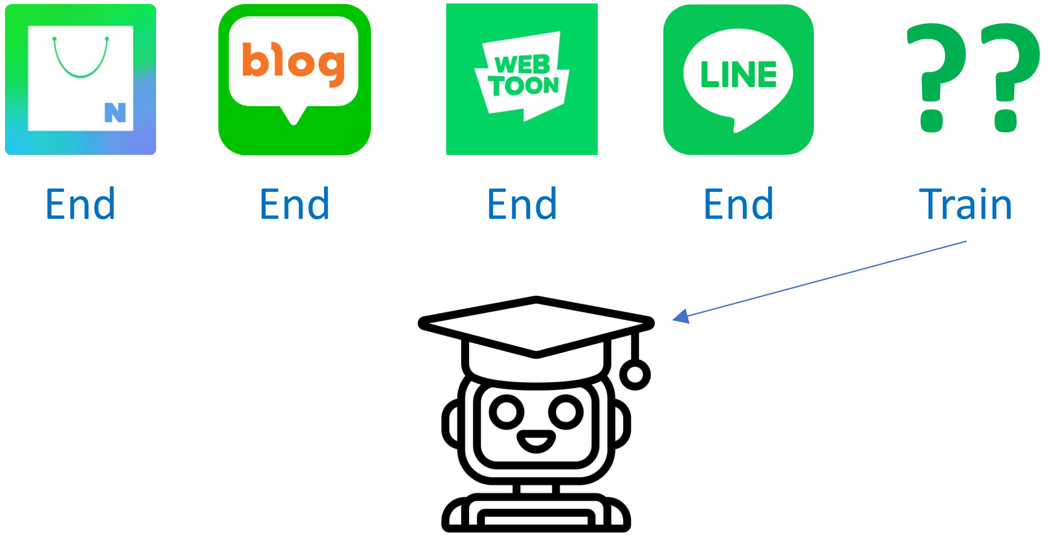
Continual Learning

Continual Learning?

The method of **sequentially learning** new knowledge in a **single model** while handling **multiple tasks**.
The key aspect here is to **maintain** the **knowledge** acquired from **previous tasks**!







General Machine Learning

In traditional Machine Learning, it assumes i.i.d. samples from a fixed data distribution.

Continual Learning

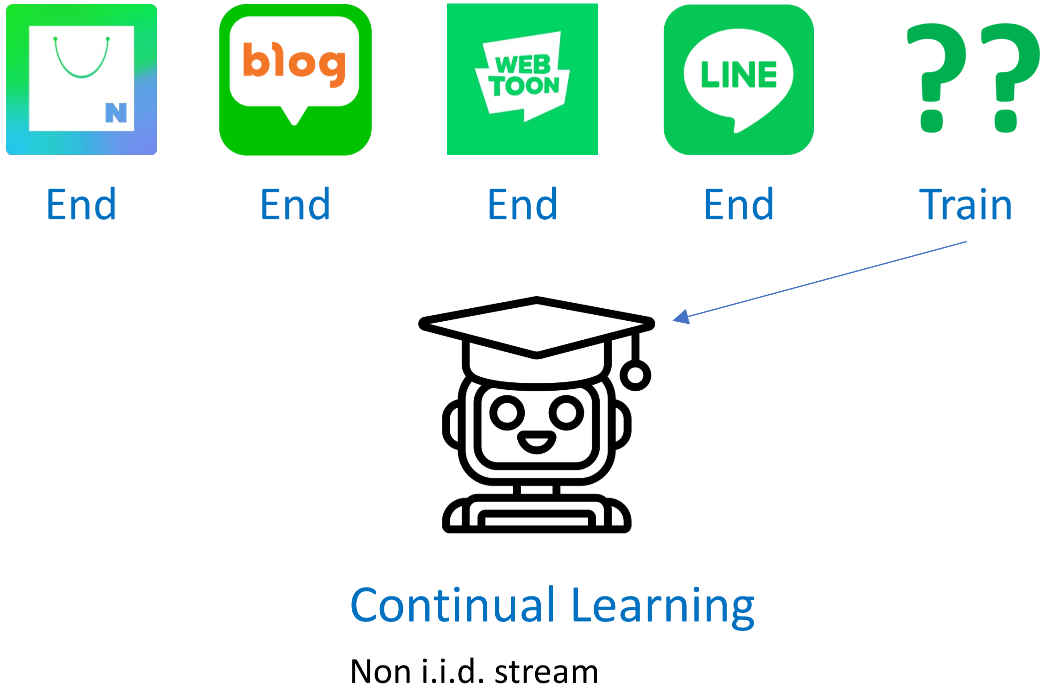
Non i.i.d. stream

Continual Learning

➢ **Task-incremental Learning**
- **Inference** is performed for a **specific task** under the context of **knowing the task** to be **executed**

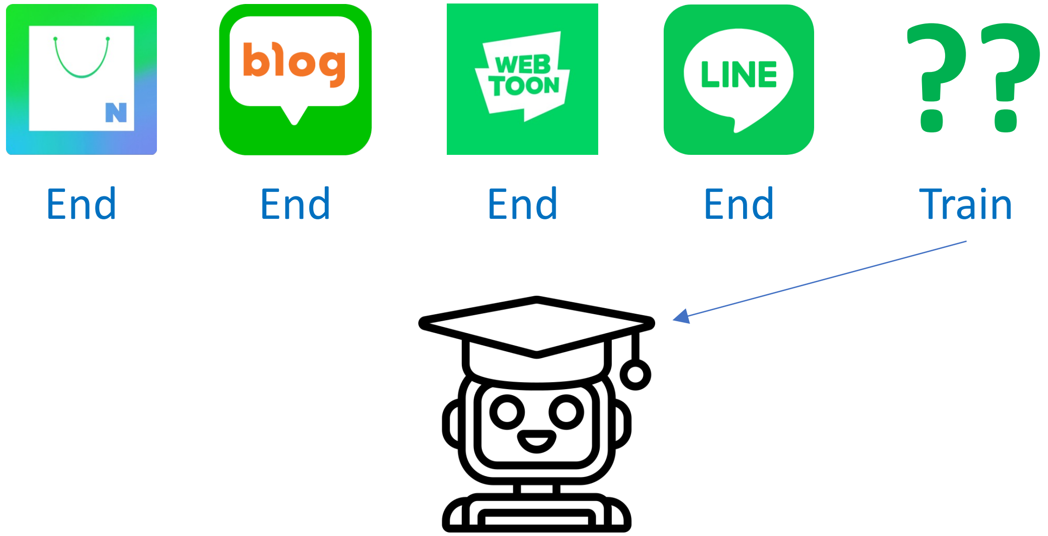
➢ **Domain-incremental Learning**
- All tasks have the **same labels**, but **different input** domains

➢ **Class-incremental Learning**
- **Inference** is conducted **simultaneously** for **all** learned **tasks**



Continual Learning

- **Task-incremental Learning**
  - **Inference** is performed for a **specific task** under the context of **knowing the task** to be **executed**
- **Domain-incremental Learning**
  - All tasks have the **same labels**, but **different input domains**
- **Class-incremental Learning**
  - **Inference** is conducted **simultaneously** for **all** learned **tasks**
- Necessity of **Continual Learning**
  - In domains with **continuous** influx of **new users** and launching of **new services**, **continual learning** structure is **suitable**
  - **Task-incremental learning** is applied in the user modeling process to accommodate the **diverse services** available on online platforms



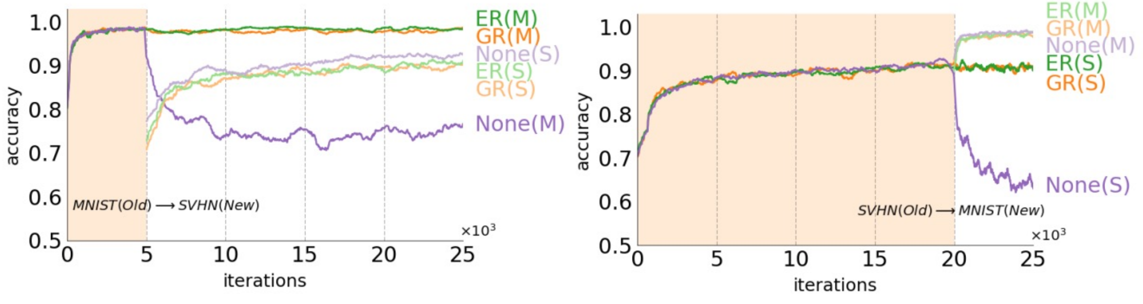
Continual Learning

Non i.i.d. stream

Continual Learning Challenges

Catastrophic Forgetting

When training a model in a **Continual Setting**, there is a situation where it becomes **biased** towards the **recent data distribution**



(a) MNIST to SVHN (b) SVHN to MNIST

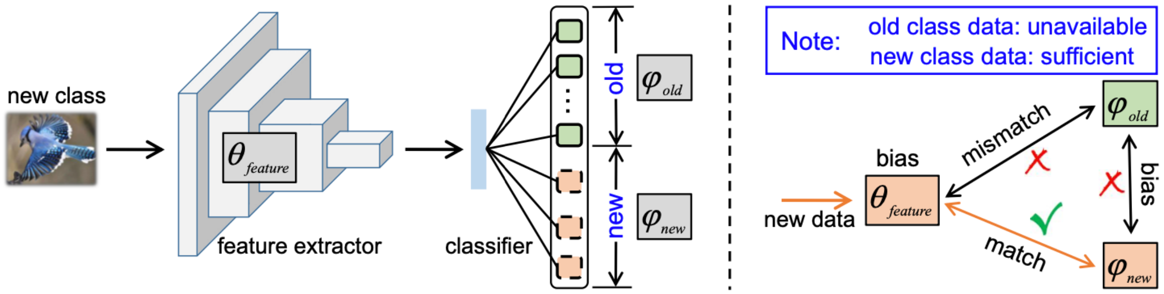


Figure 1: Two inherent problems in Class-IL: representation bias and classifier bias.

Positive Transfer

- Positive **Forward** Transfer
  The knowledge learned **from the previous task** should be **beneficial** for the **next task**

- Positive **Backward** Transfer :
  The knowledge learned **from the next task** should also be **helpful** for improving the performance of the **previous task**

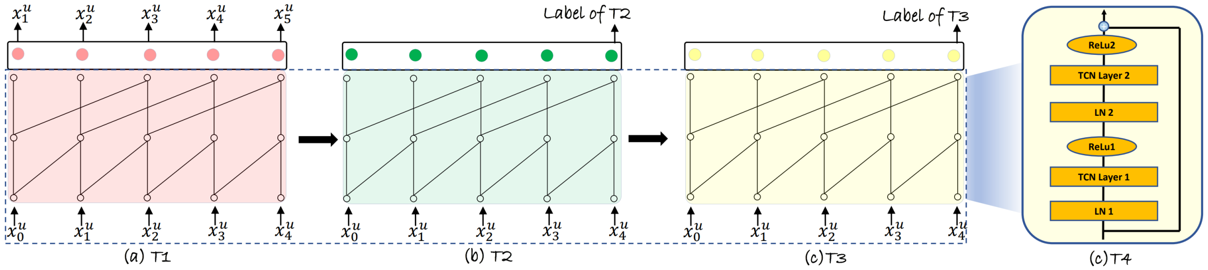
SHIN, Hanul, et al. Continual learning with deep generative replay. *Advances in neural information processing systems*, 2017, 30.
Zhu, Fei, et al. "Class-incremental learning via dual augmentation." *Advances in Neural Information Processing Systems* 34 (2021): 14306-14318.

Previous works: CONURE One Person, One Model, One World: Learning Continual User Representation without Forgetting

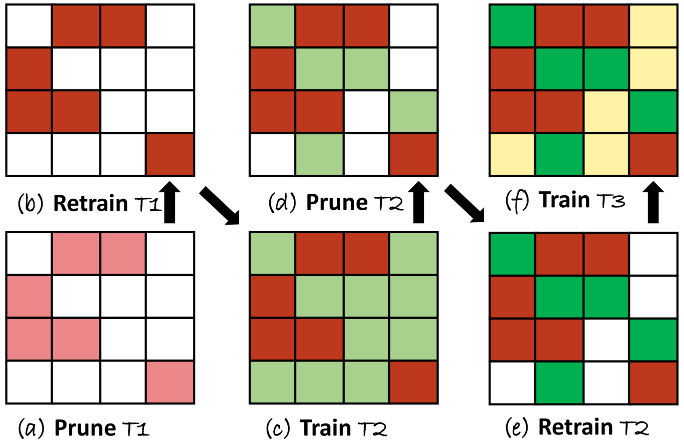
Key Idea: Parameter Isolation

When given a single model, **assign specific parameters for each task**

After training on a single task, **select important parameters**
→ Retrain and freeze only the important parameters for the next task's learning



Backbone network: NextItNet

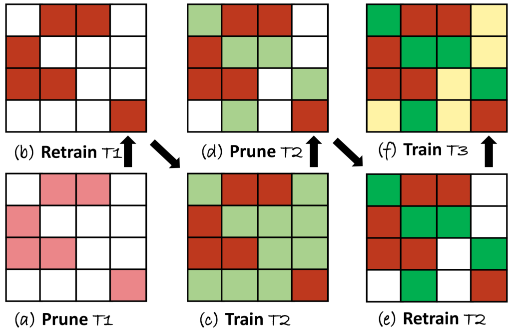


Example of Model Training Process

Previous works: CONURE One Person, One Model, One World: Learning Continual User Representation without Forgetting

Limitations of the previous work:

- Limitation of the **parameter isolation**-based methodology:
  - As tasks are sequentially learned, **the number of parameters** available for learning **new tasks decreases** and the **performance** of the model may **degrade** for **tasks** that come **later in the sequence**
  - Once **all parameters are used**, it is **no longer possible** to learn **additional tasks**
  - Since **parameters** from previous tasks are **fixed**, positive **backward transfer does not occur**



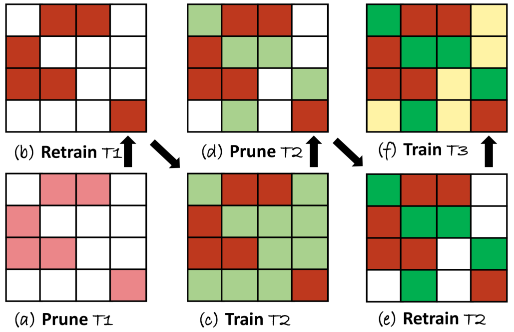
Parameter Isolation

Previous works: CONURE One Person, One Model, One World: Learning Continual User Representation without Forgetting

Limitations of the previous work:

➢ Limitation of the **parameter isolation**-based methodology:

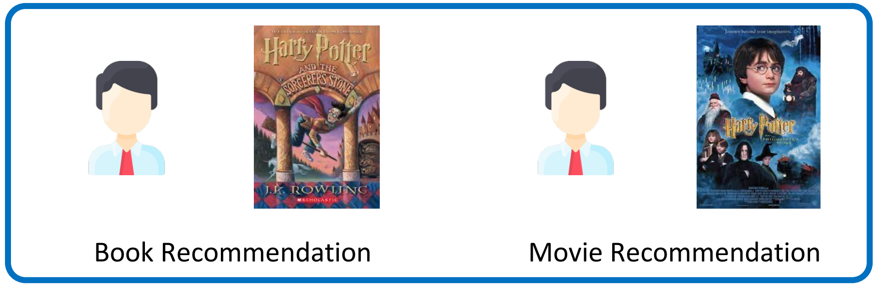
- As tasks are sequentially learned, **the number of parameters** available for learning **new tasks decreases** and the **performance** of the model may **degrade** for **tasks** that come **later in the sequence**
- Once **all parameters are used**, it is **no longer possible** to learn **additional tasks**
- Since **parameters** from previous tasks are **fixed**, positive **backward transfer does not occur**

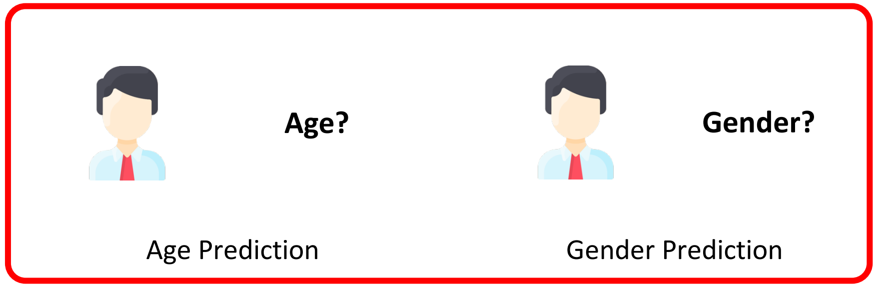


Parameter Isolation

➢ Limitation of **not considering** the **relationship** between tasks:

- There exist **specific relationships between tasks**, for example, positively related tasks and negatively related tasks
- By considering the relationships between tasks, **positive transfer becomes** possible, and **negative transfer** can be **prevented**
- Existing work disregards task relationships, leading to their inability to capture potential performance improvements of the model





TERACON Task Relation-aware Continual User Representation Learning

Motivation

Through the **continual learning**, train a **single model** capable of performing **multiple** sequence of **tasks**

By considering the **relationships** between tasks, we **maximize positive transfer** and **minimize negative transfer**

TERACON Task Relation-aware Continual User Representation Learning

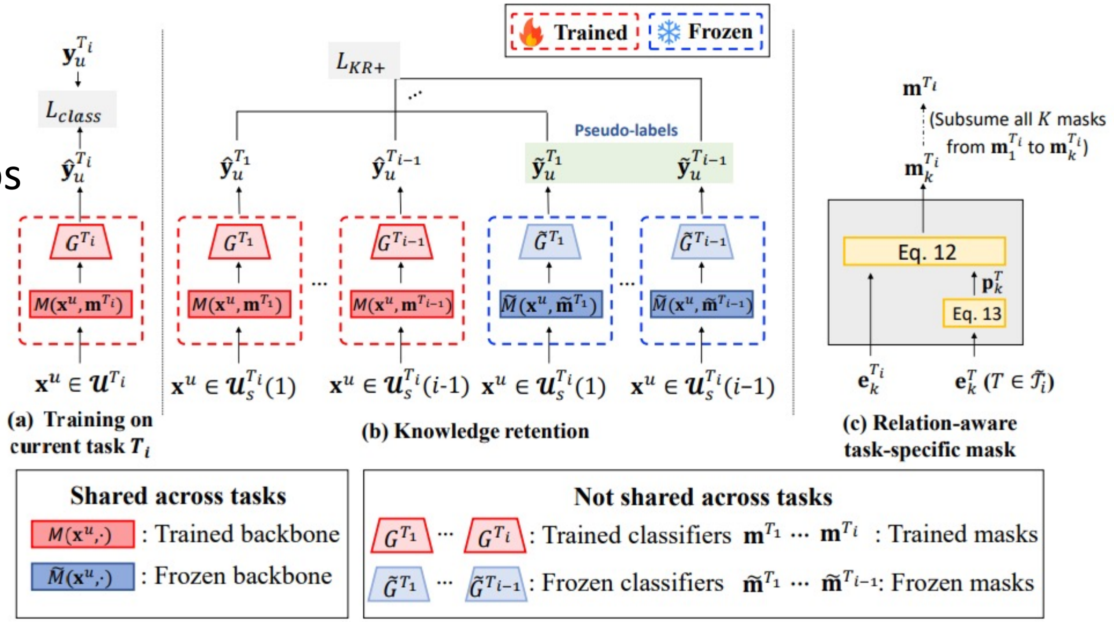
Motivation

Through the **continual learning**, train a **single model** capable of performing **multiple** sequence of **tasks**
By considering the **relationships** between tasks, we **maximize positive transfer** and **minimize negative transfer**

Key Idea

Task Embedding-guided Relation-Aware CONtinual learning (TERACON)

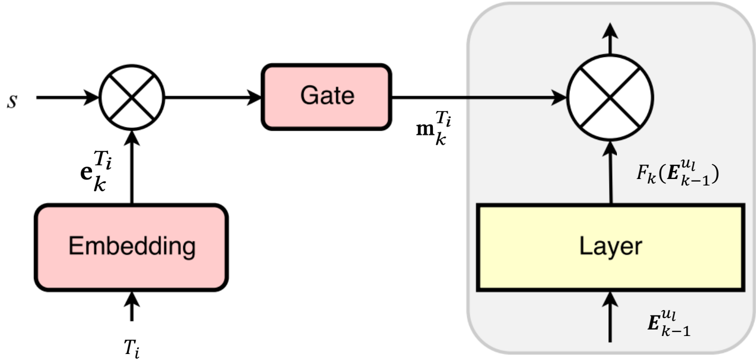
- **Task Embedding**: Learn task-specific masks (Soft masking)
- **Pseudo labeling**: Prevent catastrophic forgetting
- **Relation-aware Task-specific Mask**: Capture the task relationships



TERACON Task Relation-aware Continual User Representation Learning

Learning Task-specific Mask via Task Embedding

- Generate **task embeddings** (randomly initialized) for each layer of the model
- Generate Task masking using a positive scaling hyper-parameter (denote s) and a sigmoid function (σ)
- Perform **element-wise multiplication** of the **mask** with each **output of the model's layers**



$$\mathbf{m}_k^{T_i} = \sigma(s \cdot \mathbf{e}_k^{T_i})$$

$$F_k(\mathbf{E}_{k-1}^{u_l}) \odot \mathbf{m}_k^{T_i}$$

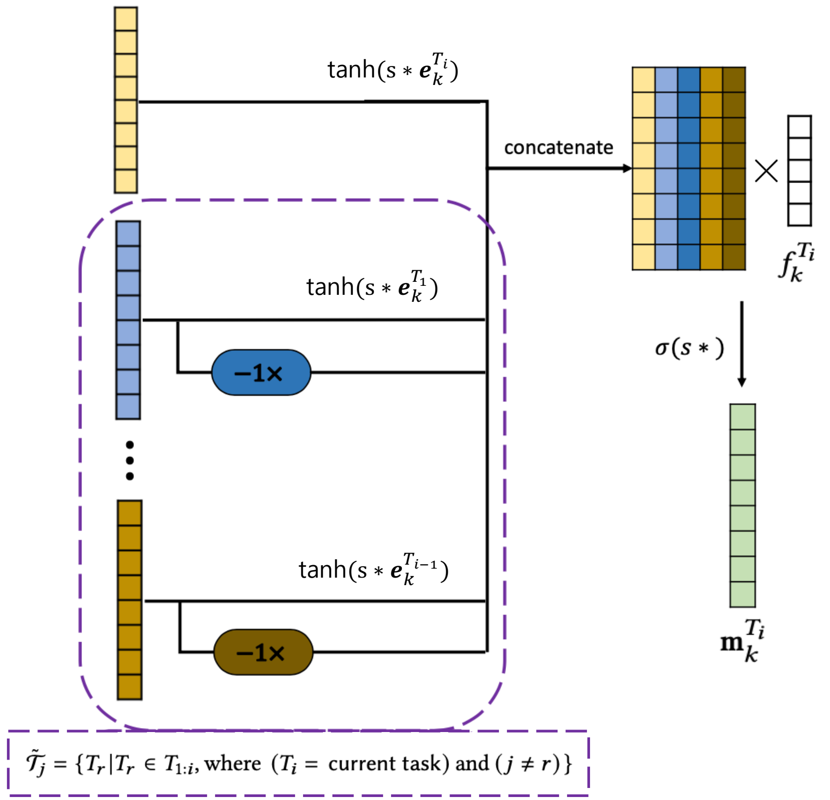
Figure from Serra, Joan, et al.

Task masking determines **how much to amplify or reduce** the layer output at each position (soft masking)
→ This allows **identification of important output** for specific tasks

Serra, Joan, et al. "Overcoming catastrophic forgetting with hard attention to the task." *International conference on machine learning*. PMLR, 2018.

TERACON Task Relation-aware Continual User Representation Learning

Relation-aware Task-specific Mask



$$\mathbf{m}_k^{T_i} = \sigma\left(s \cdot f_k^{T_i}\left[\tanh(s \cdot \mathbf{e}_k^{T_i}) \parallel (\parallel_{T \in \tilde{\mathcal{T}}_i} \mathbf{p}_k^{T})\right]\right) \in \mathbb{R}^f$$

$$\tilde{\mathcal{T}}_j = \{T_r | T_r \in T_{1:i}, \text{where } (T_i = \text{ current task}) \text{ and } (j \neq r)\}$$

$$\mathbf{p}_k^{T} = \left[\tanh(s \cdot \mathbf{e}_k^{T}) \parallel \tanh(-s \cdot \mathbf{e}_k^{T})\right] \in \mathbb{R}^{2\times f}$$

- To capture the **task relation**, TERACON **aggregate** the **information** from the **past tasks and the current task**
- Using 1-layer MLP $f_k^{T_i}$, create relation-aware task-specific mask
- $f_k^{T_i}$ is used to learn **how to amplify or diminish the information** from a **specific task** while training the current task
  →Learn to use **only** the information from **tasks** that **provide positive transfer** to the current task

TERACON Task Relation-aware Continual User Representation Learning

Overcoming Catastrophic Forgetting via Knowledge Retention

- If the **previous task** has been **adequately learned**, it is possible to **generate pseudo-labels** for previous task using current task's input

TERACON Task Relation-aware Continual User Representation Learning

Overcoming Catastrophic Forgetting via Knowledge Retention

- If the **previous task** has been **adequately learned**, it is possible to **generate pseudo-labels** for previous task using current task's input

- E.g.,
  Past Task: Learn the **age** of users A, B, C
  Current Task: Predict the **gender** of users C, D, E
  current input (users C, D, E) → Generates **pseudo-labels** for the **ages** of C, D, E
  - By training on these pseudo-labels, it is possible **to retain the age** information **for C** and **learn the age** information of **D, E**

- By training on pseudo-labels, the **knowledge** from **past tasks** can be **preserved**

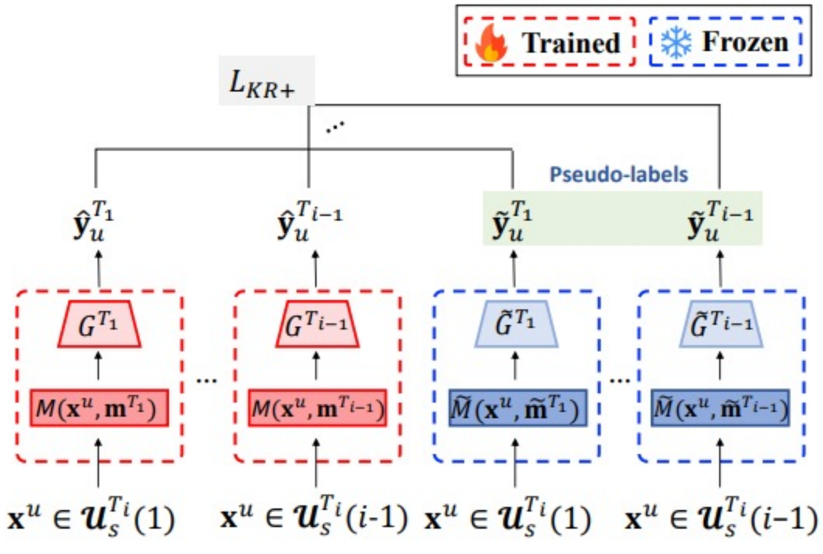
TERACON Task Relation-aware Continual User Representation Learning

Overcoming Catastrophic Forgetting via Knowledge Retention

- If the **previous task** has been **adequately learned**, it is possible to **generate pseudo-labels** for previous task using current task's input

- E.g.,
  Past Task: Learn the **age** of users A, B, C
  Current Task: Predict the **gender** of users C, D, E
  current input (users C, D, E) → Generates **pseudo-labels** for the **ages** of C, D, E
  ➢ By training on these pseudo-labels, it is possible **to retain the age** information **for C** and **learn the age** information of **D, E**

- By training on pseudo-labels, the **knowledge** from **past tasks** can be **preserved**



(b) Knowledge retention

$$\mathcal{L}_{\mathrm{KR}} = \mathop{\mathbb{E}}_{1 \le j < i} \left[\mathop{\mathbb{E}}_{u_l \in \mathcal{U}^{T_i}} \left[L_{\mathrm{MSE}}\left(G^{T_j}\left(\mathcal{M}(\mathbf{x}^{u_l}; \mathbf{m}^{T_j})\right), \tilde{\mathbf{y}}_{u_l}^{T_j}\right) \right] \right]$$

TERACON Task Relation-aware Continual User Representation Learning

Relation-aware User Sampling Strategy

- Using the **entire input** to create pseudo-labels is **inefficient**

TERACON Task Relation-aware Continual User Representation Learning

Relation-aware User Sampling Strategy

- Using the **entire input** to create pseudo-labels is **inefficient**
- Therefore, create pseudo-labels using **only a portion** of the **input** $\mathcal{U}_s^{T_i}(j) \leftarrow \text{sample}(\mathcal{U}^{T_i}, \rho_{i,j})$

TERACON Task Relation-aware Continual User Representation Learning

Relation-aware User Sampling Strategy

- Using the **entire input** to create pseudo-labels is **inefficient**
- Therefore, create pseudo-labels using **only a portion** of the **input**
- **Idea**: in continual learning, **positive transfer exists**
  → Task with positive transfer (**similar** task):
  knowledge retention can be achieved with **fewer samples**
  → Task with negative transfer (**dissimilar** task):
  more samples are required to retain knowledge and prevent catastrophic forgetting

$$\mathcal{U}_s^{T_i}(j) \leftarrow \text{sample}(\mathcal{U}^{T_i}, \rho_{i,j})$$

$$\rho_{i,j} = 1 - \frac{1}{K}\sum_{k=1}^{K} \sigma(c \times \underline{\cos(\mathbf{m}_k^{T_i}, \tilde{\mathbf{m}}_k^{T_j})})$$

$\boldsymbol{m}_k^{T_i}$: mask of current task (T_i)

$\tilde{\boldsymbol{m}}_k^{T_j}$: mask of T_j prior to training T_i

TERACON Task Relation-aware Continual User Representation Learning

Relation-aware User Sampling Strategy

- Using the **entire input** to create pseudo-labels is **inefficient**
- Therefore, create pseudo-labels using **only a portion** of the **input**
- **Idea**: in continual learning, **positive transfer exists**
  → Task with positive transfer (**similar** task):
  knowledge retention can be achieved with **fewer samples**
  → Task with negative transfer (**dissimilar** task):
  more samples are required to retain knowledge and prevent catastrophic forgetting

$$\mathcal{U}_s^{T_i}(j) \leftarrow \text{sample}(\mathcal{U}^{T_i}, \rho_{i,j})$$

$$\rho_{i,j} = 1 - \frac{1}{K}\sum_{k=1}^{K}\sigma(c \times \underline{\cos(\mathbf{m}_k^{T_i}, \tilde{\mathbf{m}}_k^{T_j})})$$

$\boldsymbol{m}_k^{T_i}$: mask of current task (T_i)
$\tilde{\boldsymbol{m}}_k^{T_j}$: mask of T_j prior to training T_i

$$\mathcal{L}_{\text{KR}} = \mathop{\mathbb{E}}_{1\le j<i}\left[\mathop{\mathbb{E}}_{u_l\in\mathcal{U}^{T_i}}\left[L_{\text{MSE}}(G^{T_j}(\mathcal{M}(\mathbf{x}^{u_l};\mathbf{m}^{T_j})), \tilde{\mathbf{y}}_{u_l}^{T_j})\right]\right]$$

$$\downarrow$$

$$\mathcal{L}_{\text{KR+}} = \mathop{\mathbb{E}}_{1\le j<i}\left[\frac{\rho_{i,j}}{\sum_{k=1}^{i-1}\rho_{i,k}}\sum_{u_l\in\mathcal{U}_s^{T_i}(j)}\mathcal{L}_{\text{MSE}}(G^{T_j}(\mathcal{M}(\mathbf{x}^{u_l};\mathbf{m}^{T_j})), \tilde{\mathbf{y}}_{u_l}^{T_j})\right]$$

TERACON Experiments

Datasets & Tasks Descriptions

Dataset	Task 1 (T_1)		Task 2 (T_2)		Task 3 (T_3)		Task 4 (T_4)		Task 5 (T_5)		Task 6 (T_6)	
	$\|\mathcal{U}^{T_1}\|$	$\|\mathcal{Y}^{T_1}\|$	$\|\mathcal{U}^{T_2}\|$	$\|\mathcal{Y}^{T_2}\|$	$\|\mathcal{U}^{T_3}\|$	$\|\mathcal{Y}^{T_3}\|$	$\|\mathcal{U}^{T_4}\|$	$\|\mathcal{Y}^{T_4}\|$	$\|\mathcal{U}^{T_5}\|$	$\|\mathcal{Y}^{T_5}\|$	$\|\mathcal{U}^{T_6}\|$	$\|\mathcal{Y}^{T_6}\|$
TTL	Watching		Clicking		Thumb-up		Age		Gender		Life status	
	1.47M	0.64M	1.39M	17K	0.25M	7K	1.47M	8	1.46M	2	1M	6
ML	Clicking		4-star		5-star		-		-		-	
	0.74M	54K	0.67M	26K	0.35M	16K						
NAVER Shopping	Search Query		Search Query		Item Category		Item Category		Gender		Age	
	0.9M	0.58M	0.59M	0.51M	0.15M	4K	0.15M	10	0.82M	2	0.82M	9

Tencent TL (TTL) dataset

T_1 → (userID, recent 100 news & video on QQ browser platform) — **Sequential Recommendation**

T_2 → (userID, one of clicking interactions on the Kandian platform)
T_3 → (userID, one of thumb-up interactions on the Kandian platform) — **Item Recommendation**

T_4 → (userID, age)
T_5 → (userID, gender)
T_6 → (userID, Life status categories) — **Profile Prediction**

ML (Movie Lens) dataset

T_1 → (userID, recent 30 clicking interactions) — **Sequential Recommendation**

T_2 → (userID, an item that is rated higher than 4)
T_3 → (userID, one of 5-star items) — **Item Recommendation**

NAVER Shopping dataset

T_1 → (userID, recent 60 search queries in NAVER browser platform)
T_2 → (userID, next five search queries after T_1 in NAVER browser platform) — **Sequential Learning**

T_3 → (userID, minor categories of user-purchased items in NAVER shopping platform)
T_4 → (userID, major categories of user-purchased items in NAVER shopping platform) — **Item Recommendation**

T_5 → (userID, gender)
T_6 → (userID, age) — **Profile Prediction**

Conduct diverse tasks using the user's previous search history
Search history holds the most comprehensive information about the user → Generates a meaningful Universal user presentation

TERACON Experiments

Overall Performance

	TTL						ML			NAVER Shopping					
	T_1	T_2	T_3	T_4	T_5	T_6	T_1	T_2	T_3	T_1	T_2	T_3	T_4	T_5	T_6
SinMo	0.0446	0.0104	0.0168	0.4475	0.8901	0.4376	0.0566	0.0186	0.0314	0.0349	0.0265	0.0292	0.1984	0.5742	0.2985
FineAll	0.0446	0.0144	0.0218	0.5232	0.8851	0.4596	0.0566	0.0224	0.0328	0.0349	0.0318	0.0332	0.2367	0.6204	0.3247
PeterRec	0.0446	0.0147	0.0224	0.5469	0.8841	0.4749	0.0566	0.0224	0.0308	0.0349	0.0317	0.0322	0.2370	0.6257	0.3258
MTL	-	0.0102	0.0142	0.4672	0.8012	0.3993	-	0.0144	0.0267	-	0.0143	0.0266	0.1372	0.4998	0.2322
Piggyback	0.0446	0.0157	0.0236	0.5931	0.8990	0.5100	0.0566	0.0214	0.0302	0.0349	0.0314	0.0322	0.2349	0.6188	0.3129
HAT	0.0424	0.0174	0.0279	0.5880	0.9002	0.5126	0.0543	0.0227	0.0372	0.0344	0.0356	0.0317	0.2411	0.6294	0.3296
CONURE	0.0457	0.0169	0.0276	0.5546	0.8967	0.5230	**0.0598**	0.0244	0.0384	**0.0361**	0.0322	0.0305	0.2403	**0.6391**	0.3340
TERACON	**0.0474**	**0.0189**	**0.0316**	**0.6066**	**0.9048**	**0.5386**	0.0577	**0.0270**	**0.0459**	**0.0361**	**0.0359**	**0.0337**	**0.2444**	0.6381	**0.3354**

Trains a single model for each task from scratch

Transfer Learning (T_1 → T_i)

Multi-task Learning

Continual Learning

Model performance

Observations

- **Positive transfer** occurs between tasks (SinMo vs. others)
  SinMo means single model → Learns tasks separately (# of model = # of tasks)
- Continual learning-based methods perform better than other universal user representation method
- TERACON outperforms the continual learning-based approaches
  → Modeling the **relationship** between tasks is **crucial**

TERACON Experiments

Overall Performance

(a) Original	T_1			T_2			T_3			T_4			T_5			T_6		
	MRR@5	BWT	FWT	MRR@5	BWT	FWT	MRR@5	BWT	FWT	ACC	BWT	FWT	ACC	BWT	FWT	ACC	BWT	FWT
HAT	0.0424	-11.30%	-	0.0174	-7.45%	80.77%	0.0279	-0.71%	67.25%	0.5880	-2.52%	34.79%	0.9002	-1.98%	3.17%	0.5126	-	17.14%
CONURE	0.0457	-	-	0.0169	-	62.50%	0.0276	-	64.29%	0.5546	-	23.93%	0.8967	-	0.74%	0.5230	-	19.52%
TERACON	**0.0474**	-0.83%	-	**0.0189**	0.0%	81.73%	**0.0316**	3.27%	82.13%	**0.6066**	1.23%	33.91%	**0.9048**	0.01%	1.64%	**0.5386**	-	23.08%

(b) Reversed	T_1			T_6			T_5			T_4			T_3			T_2		
	MRR@5	BWT	FWT	ACC	BWT	FWT	ACC	BWT	FWT	ACC	BWT	FWT	MRR@5	BWT	FWT	MRR@5	BWT	FWT
HAT	0.0422	-11.72%	-	0.5025	-4.70%	20.49%	0.8980	-0.33%	1.22%	0.5770	-1.72%	31.19%	0.0269	-0.37%	60.71%	0.0184	-	76.92%
CONURE	0.0457	-	-	0.5322	-	21.62%	0.8849	-	-0.58%	0.5546	-	23.93%	0.0164	-	-2.38%	0.0119	-	14.42%
TERACON	**0.0474**	-0.83%	-	**0.5365**	1.84%	20.38%	**0.9039**	0.93%	0.61%	**0.6042**	0.07%	34.92%	**0.0313**	2.62%	81.55%	**0.0190**	-	82.69%

Reversed task sequence experiment

BWT: Backward Transfer
FWT: Forward Transfer

Observations

- CONURE is a **parameter isolation**-based method that **prevents Catastrophic forgetting** even when learning a new task

- TERACON exhibits **Positive Backward** Transfer
  - →Because TERACON allows the **entire model parameters** to be **modified** during the entire training sequence, enabling the **knowledge** obtained **from the new tasks** to be **transferred** to the **previous tasks**

- TERACON is **robust** to the change of **task orders**
  - → Sequence of tasks **cannot** be **arbitrarily determined** in the real world, highlighting the importance of robustness on task order
  - →The change of task order significantly deteriorates the performance of CONURE (parameter isolation-based method)
  - →TERACON considers task relationships, is **not parameter isolation-based method** → Robust on the task order

TERACON Experiments

Overall Performance

	TTL						
	T_1	T_2	T_3	T'	T_4	T_5	T_6
HAT	0.0411 (-3.06 %)	0.0165 (-5.17 %)	0.0259 (-7.16 %)	-	0.5424 (-7.76 %)	0.8870 (-1.47 %)	0.4873 (-4.94 %)
CONURE	0.0457 (0.0 %)	0.0169 (0.0 %)	0.0276 (0.0 %)	-	0.5245 (-5.43 %)	0.8663 (-3.39 %)	0.4469 (-14.55 %)
TERACON	**0.0472** (-0.42 %)	**0.0189** (0.0 %)	**0.0314** (-0.63 %)	-	**0.6022** (-0.73 %)	**0.9014** (-0.38 %)	**0.5312** (-1.37 %)

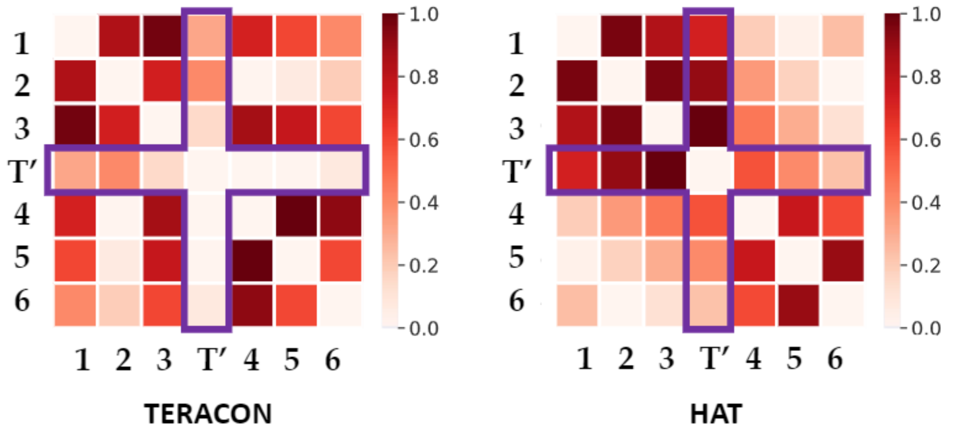
	NAVER Shopping						
	T_1	T_2	T'	T_3	T_4	T_5	T_6
HAT	0.0314 (-8.72%)	0.0302 (-15.16%)	-	0.0309 (-2.52%)	0.2357 (-2.24%)	0.6219 (-1.19%)	0.3180 (-3.51%)
CONURE	**0.0361** (0.0%)	0.0322 (0.0%)	-	0.0291 (-4.59%)	0.2231 (-7.16%)	0.6202 (-2.95%)	0.3122 (-6.53%)
TERACON	0.0346 (-4.15%)	**0.0336** (-6.41%)	-	**0.0329** (-2.37%)	**0.2378** (-2.7%)	**0.6348** (-0.52%)	**0.3329** (-0.75%)

Performance degradation ratio after training on a noisy task

Noisy task: randomly sample 50% of users and generate a random label of 50 classes for each user

Observations

- TERACON is **robust** to the **negative transfer**
  - → Automatically disregard the information from the noisy task
  - → The task-specific **masks** of **task T'** learned by TERACON exhibit **low similarity** with that of other tasks

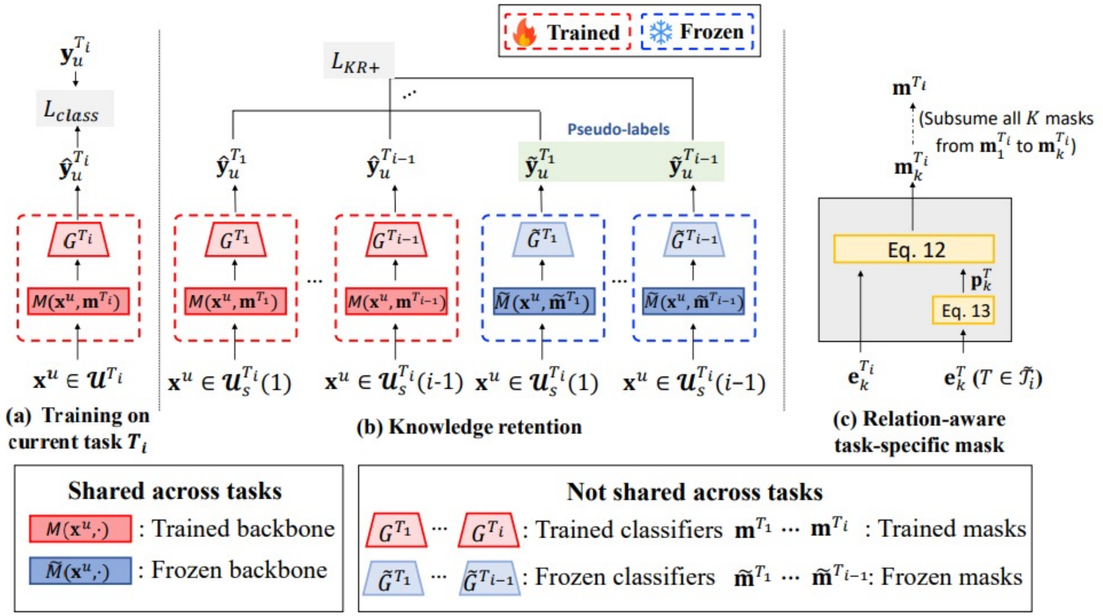


Task embedding Heatmap

Conclusion

Propose a **continual learning-based** universal user representation method

Considers the **relationships** between tasks to induce positive transfer and prevent catastrophic forgetting



Key Idea
- **Task Embedding**: Learn task-specific masks (Soft masking)
- **Pseudo labeling**: Prevent catastrophic forgetting
- **Relation-aware Task-specific Mask**: Capture the task relationships

Extensive experiments show
- Occurrence of positive backward transfer
- Improvement of performance of universal user representation
- Robustness on task order and negative transfer
- Analysis on task embedding and negative related tasks

Thank you!

[Full Paper] https://arxiv.org/abs/2306.01792

[Source Code] https://github.com/Sein-Kim/TERACON

[Lab Homepage] http://dsail.kaist.ac.kr

[Email] rlatpdlsgns@kaist.ac.kr

Paper

Code

TERACON Experiments

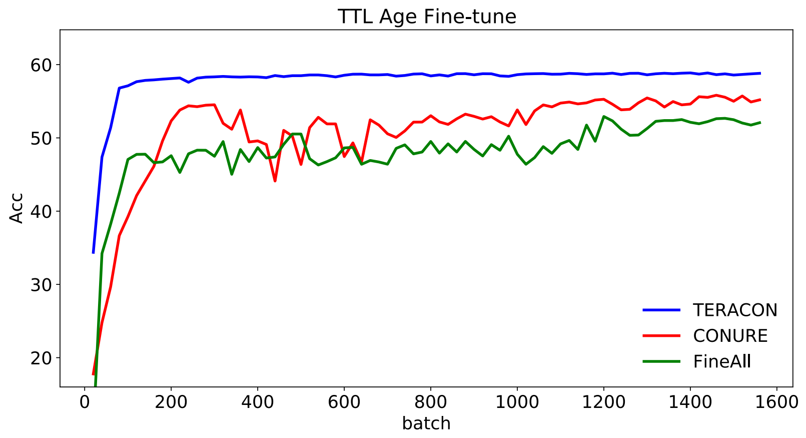
TERACON is efficient learner

	Sampling	T_1	T_2	T_3	T_4	T_5	T_6
$\rho_{i,j} = \rho_{min}$	✓	0.0470 (-)	0.0184 (625.47)	0.0280 (77.82)	0.6027 (417.65)	0.9007 (510.80)	0.5385 (414.44)
$\rho_{i,j} =$ Eq.15	✓	0.0474 (-)	0.0189 (625.47)	**0.0316** (90.79)	0.6066 (504.3)	**0.9048** (583.77)	0.5386 (494.14)
$\rho_{i,j} = 1.0$	✗	**0.0475** (-)	**0.0190** (1146.70)	0.0313 (151.32)	**0.6143** (1179.31)	0.9047 (1355.18)	**0.5403** (797.09)

User sampling

Observations

By sampling users while considering the relationship between tasks, TERACON can achieve both efficiency and performance



Model Convergence

Observations

Compared to the existing models, TERACON converges in fewer epochs
→ TERACON provides a better initial point for new tasks