SIGIR-25 Full Papers Track

Dynamic Time-aware Continual User Representation Learning

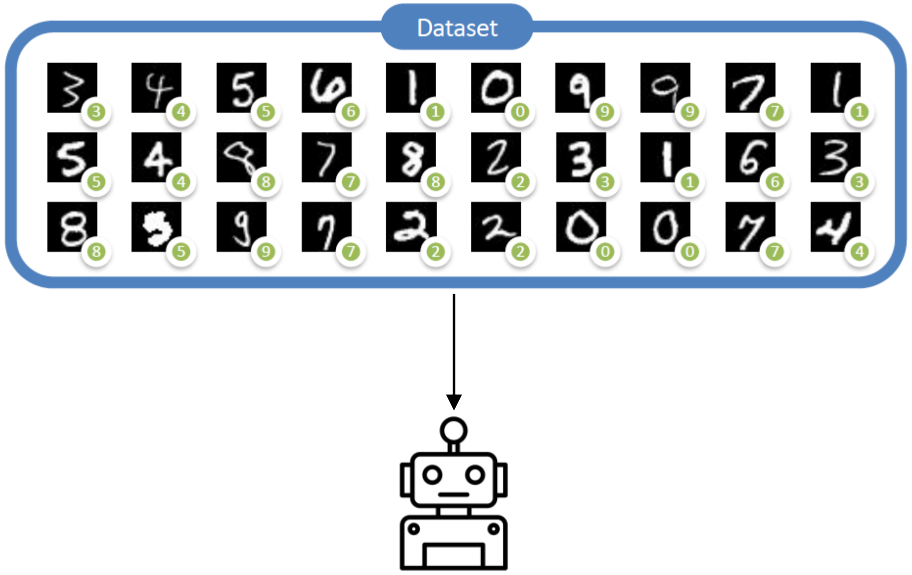
Seungyoon Choi, Sein Kim, Hongseok Kang, Wonjoong Kim, Chanyoung Park

Korean Advanced Institute of Science and Technology (KAIST)

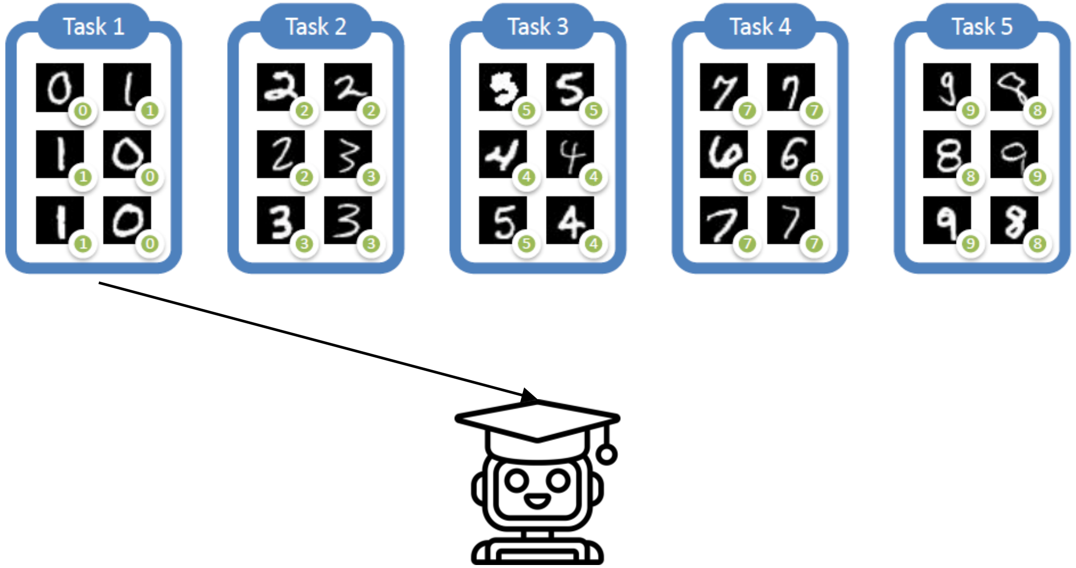
Continual Learning

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

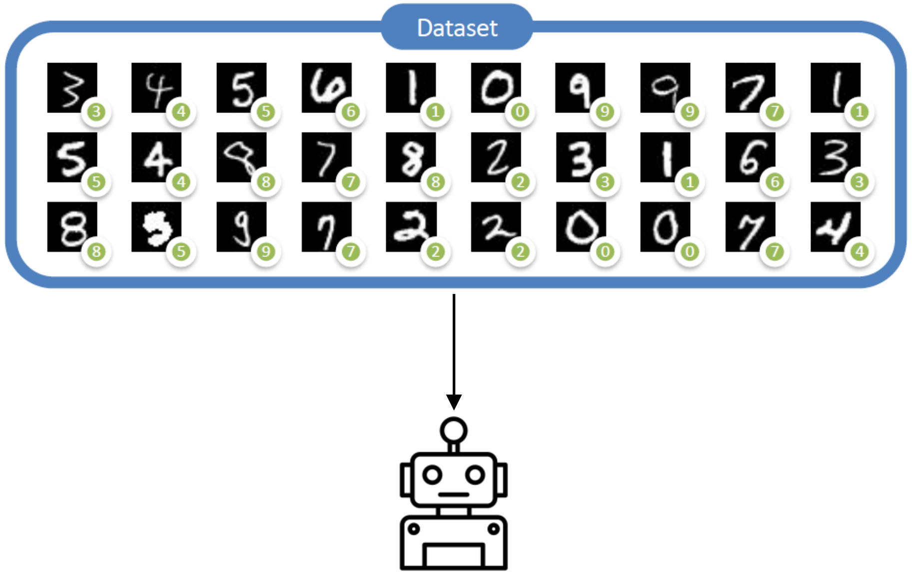


Continual Learning

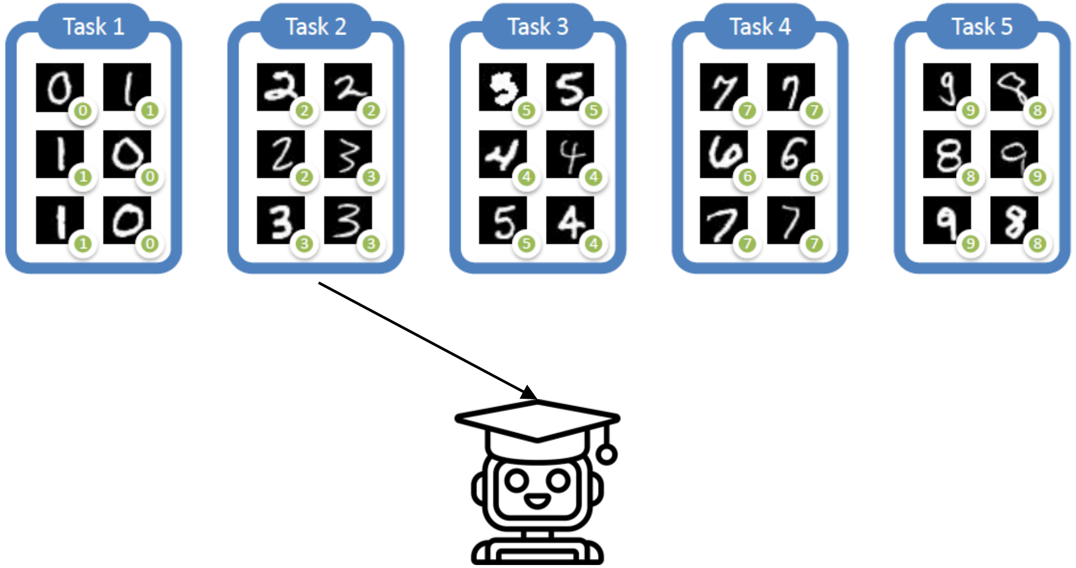
Continual Learning

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

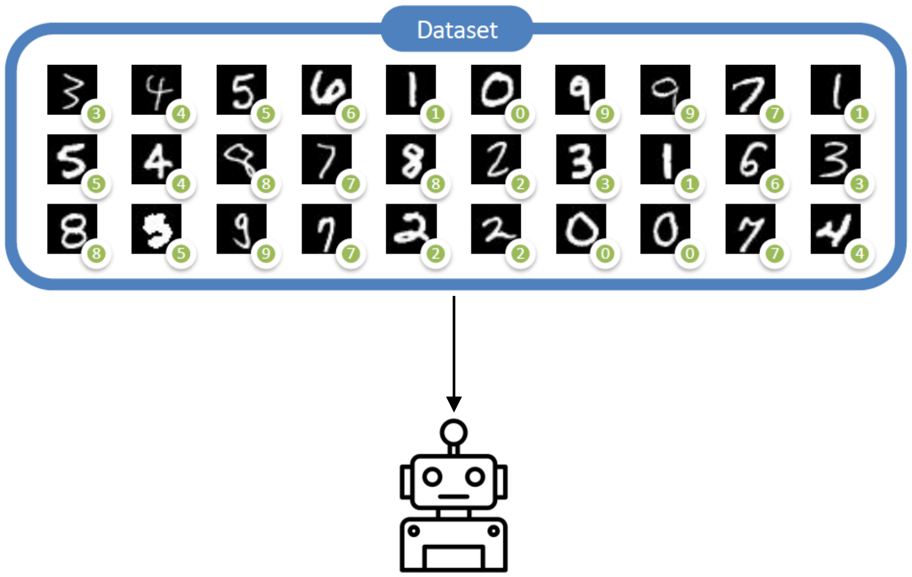


Continual Learning

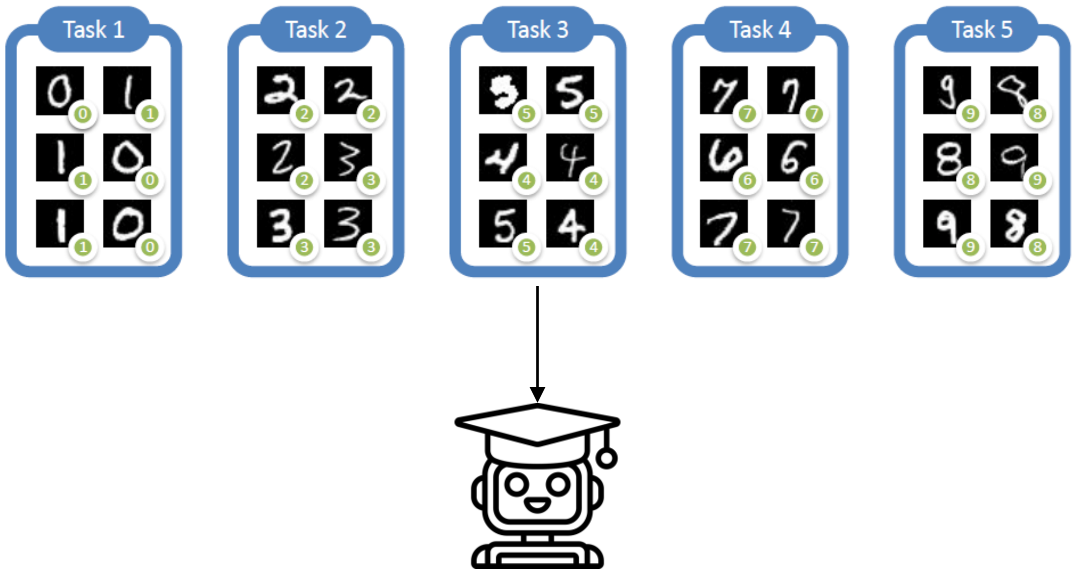
Continual Learning

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

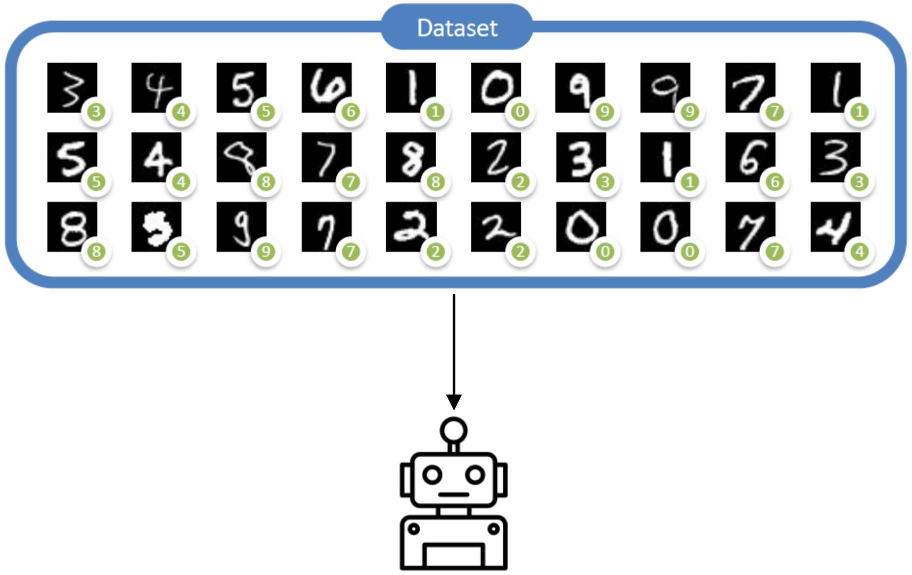


Continual Learning

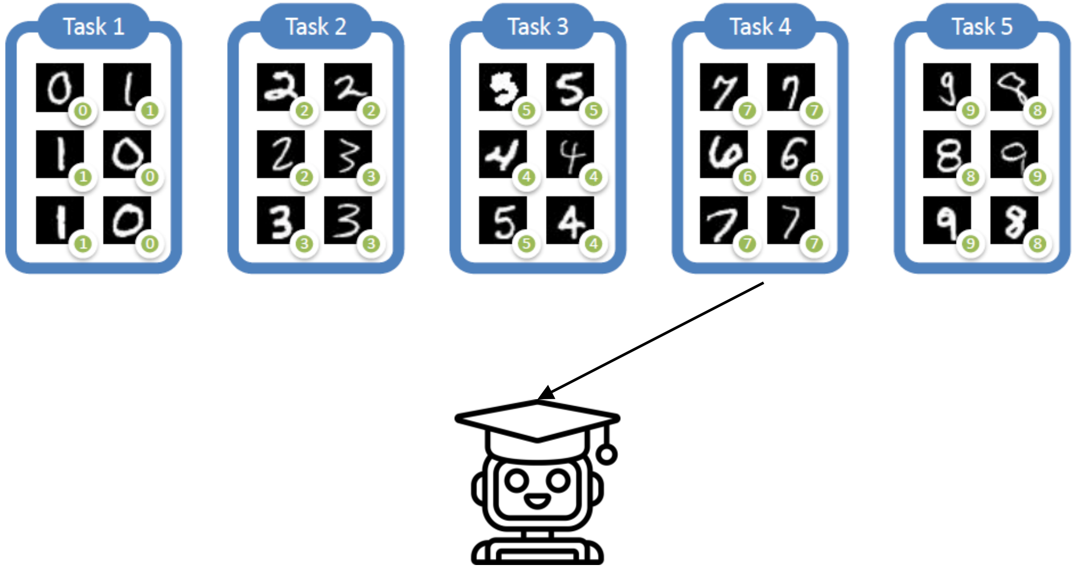
Continual Learning

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

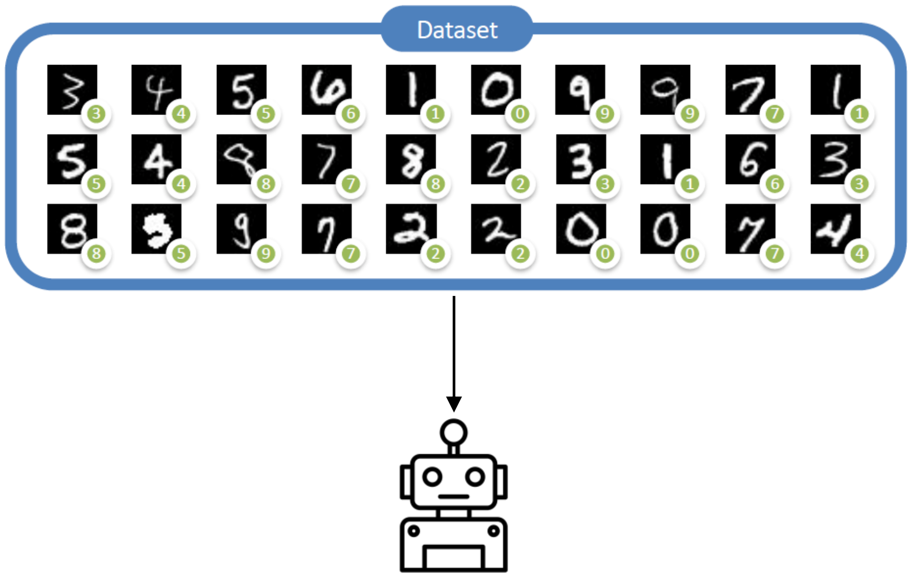


Continual Learning

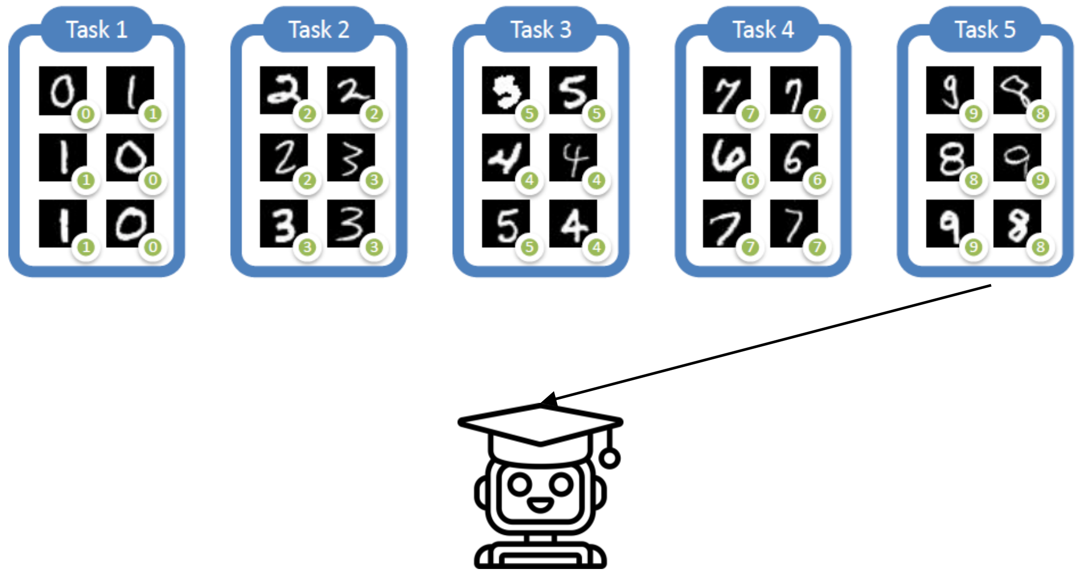
Continual Learning

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

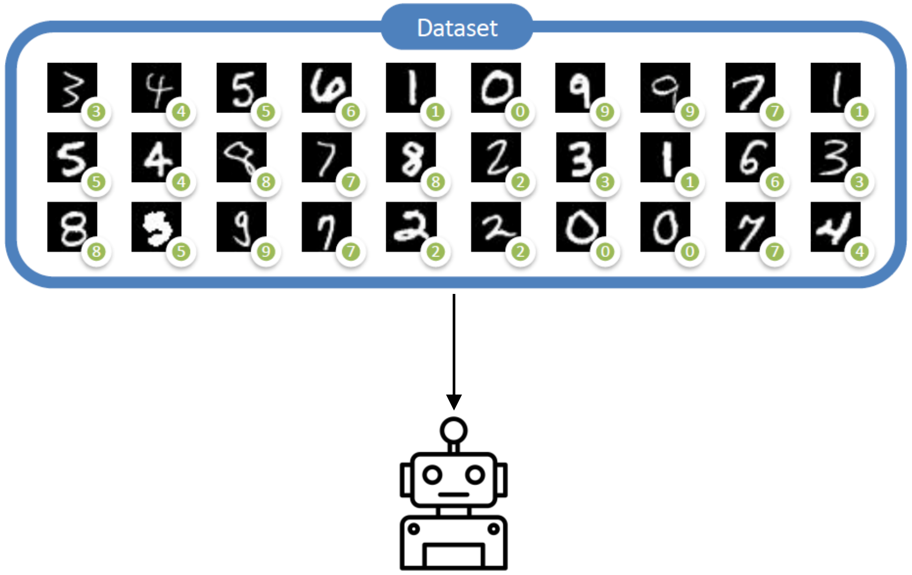


Continual Learning

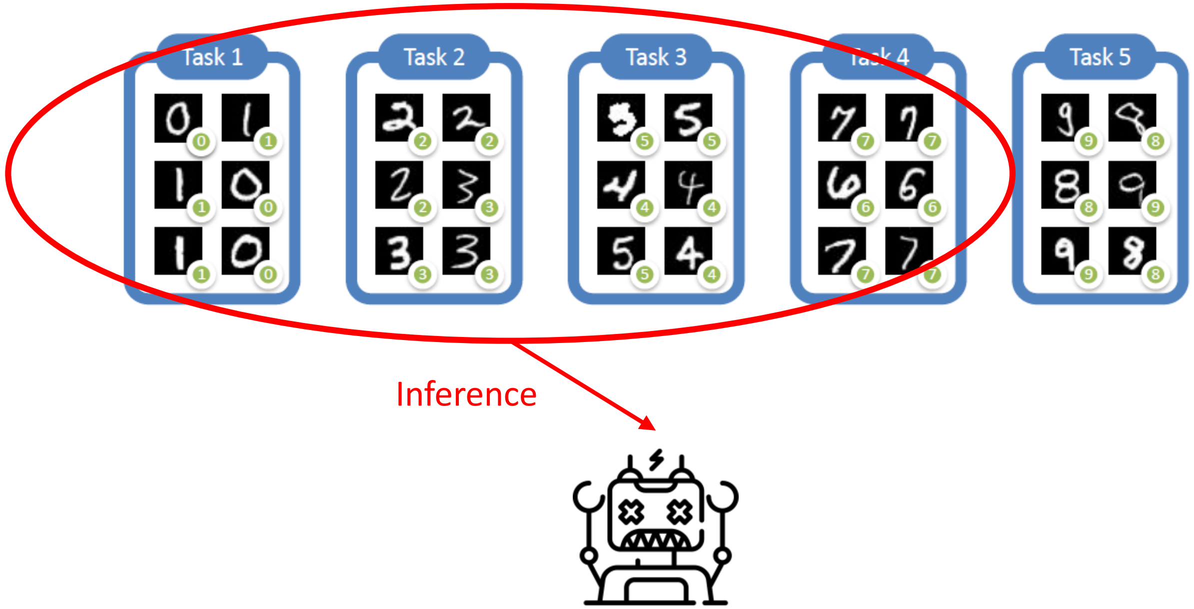
Continual Learning

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.



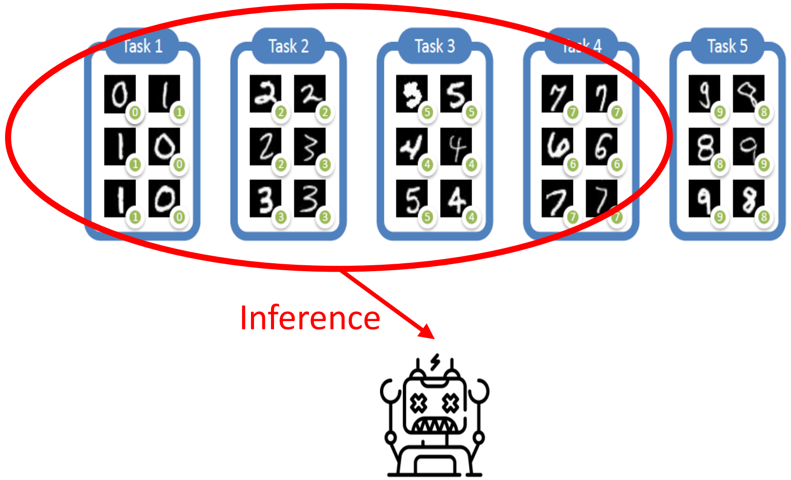
General Machine Learning



Continual Learning

Continual Learning

Challenges of Continual Learning



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

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

Task 1 → **Task 2** → **Task 3** → **Task 4** → **Continual Learning**

Forward Knowledge Transfer

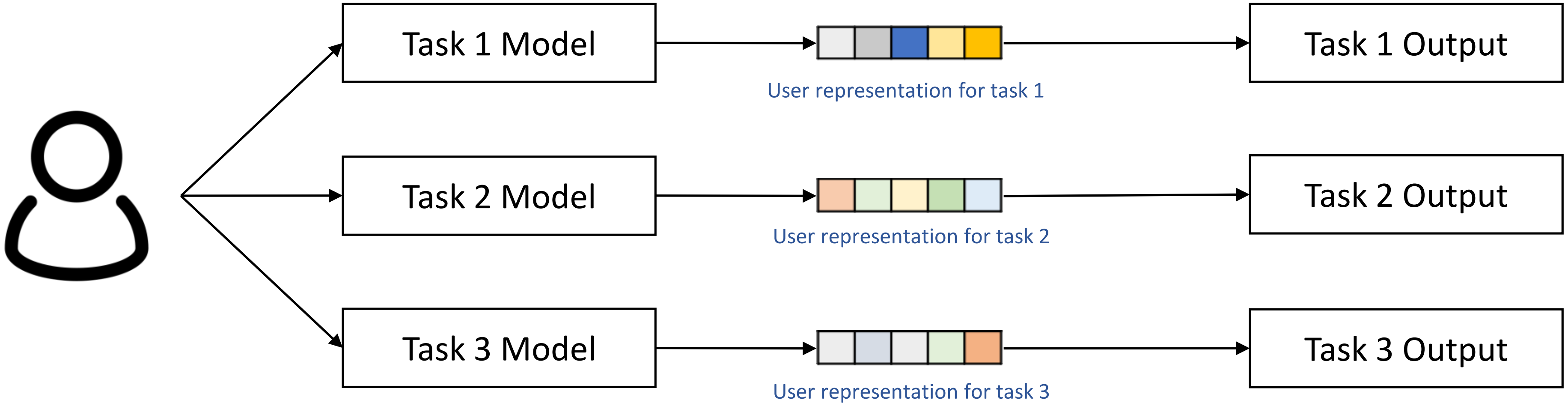
  - Positive **Backward** Transfer
    - The knowledge learned from the next task should also be helpful for improving the performance of the previous task

Task 1 → **Task 2** → **Task 3** → **Task 4** → **Continual Learning**

Backward Knowledge Transfer

User Modeling

Generating **user representations** for each task to enable personalized recommendations

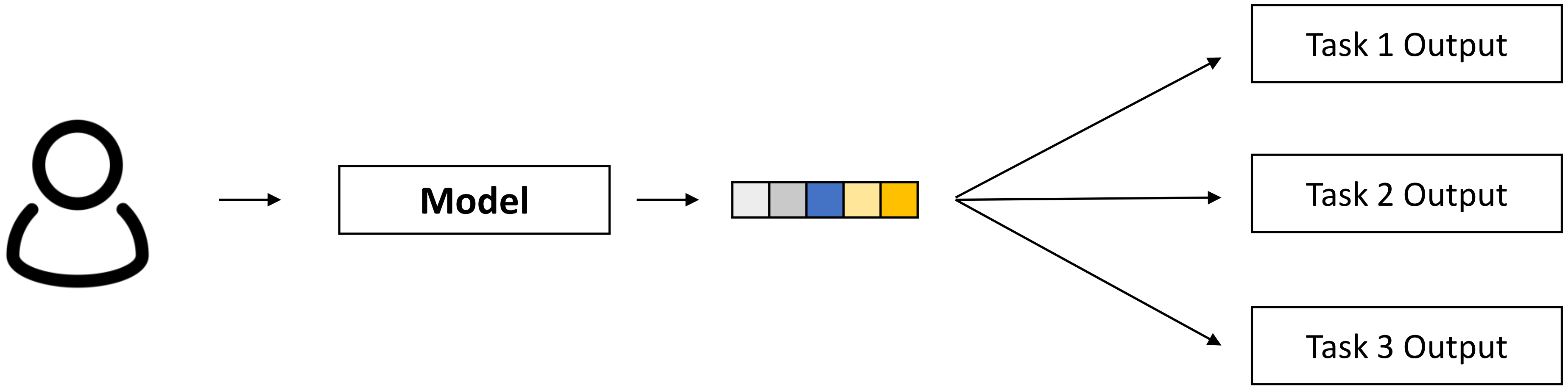


<Example of model operation for each task>

Universal User Representation Learning

Problems on User Modeling

- Inefficient : Create and train **new models** for each **new task**

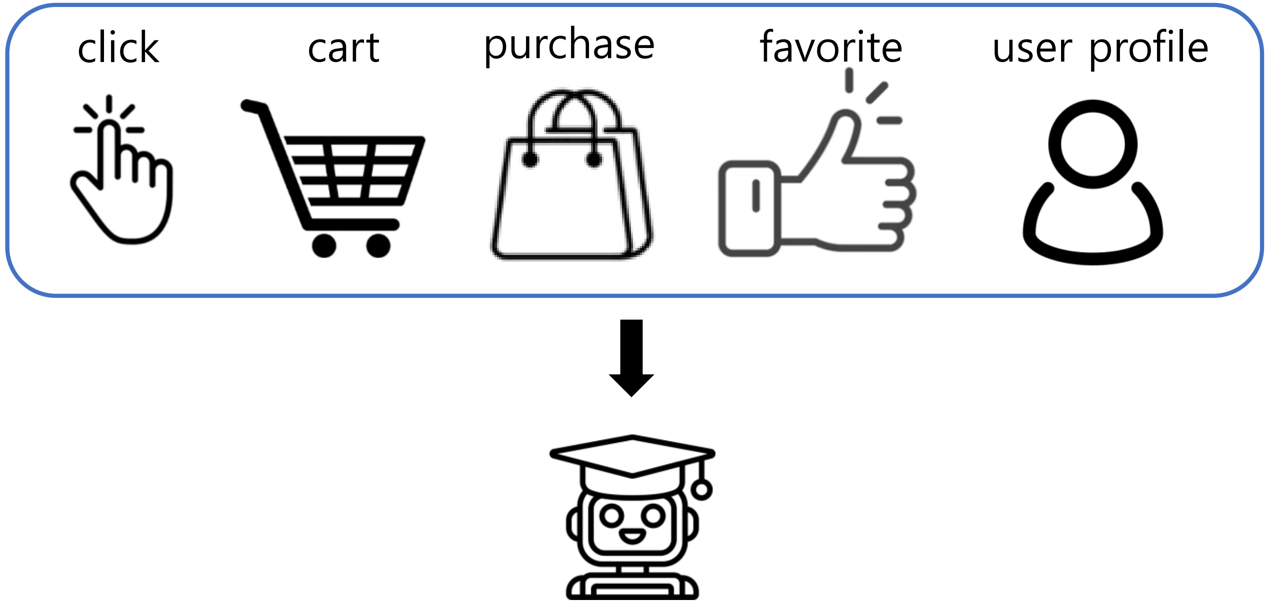


Research Objective

- Solve various tasks through a **Universal User Representation**
- Maintain competitive performance across tasks using **single universal user representation**

Continual User Representation Learning

Sequentially solving the user modeling tasks



Continual User Representation Learning

Motivation

Limitations of the previous work

Limitation of not considering the **flow of time** as the task progresses:

- In the real world, time passes as the task progresses
- **New users** with **new items** emerge

Tmall Dataset	Num. items given on 8/11	The number of new items emerged					
		8/11 ~ 8/26	8/26 ~ 9/11	9/11 ~ 9/26	9/26 ~ 10/11	10/11 ~ 10/26	10/26~ 11/12
Click	570.6K	65.0K	79.0K	61.0K	58.9K	77.0K	171.3K
Cart	6.2K	1.1K	1.9K	1.7K	1.9K	3.2K	27.1K
Purchase	153.3K	16.8K	26.5K	19.2K	18.4K	21.3K	117.3K
Favorite	195.2K	27.2K	34.4K	28.1K	28.2K	39.1K	93.1K

<Number of **new items** over time in Tmall dataset>

Motivation

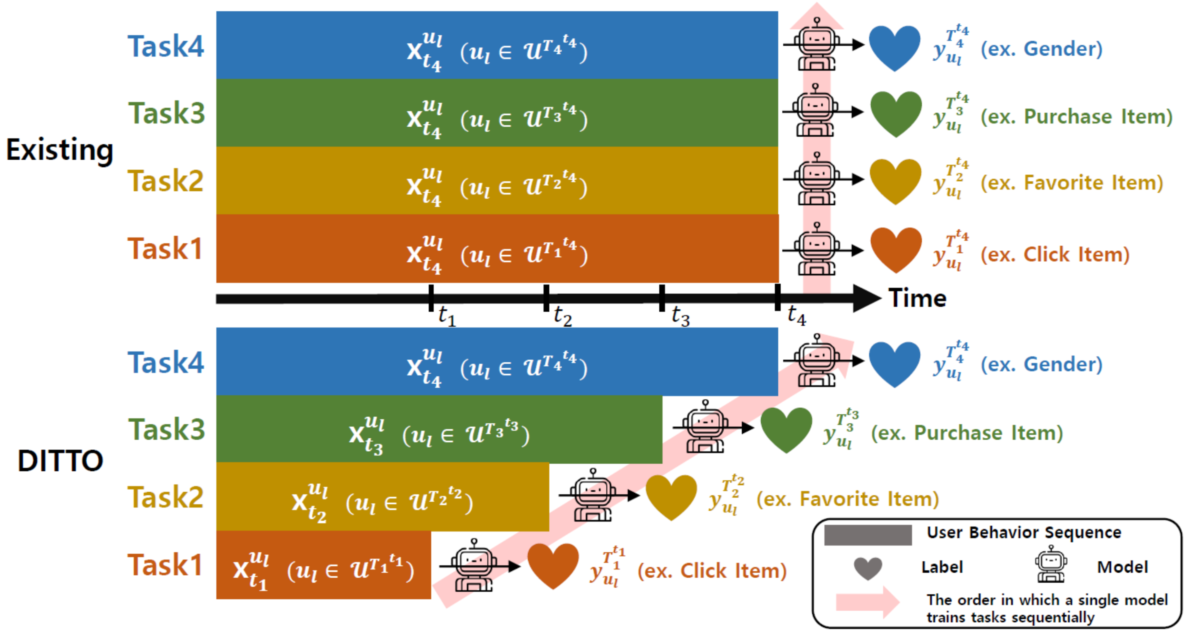
Proposed Evaluation Scenario

Training Phase

- **Time** passes as tasks progress

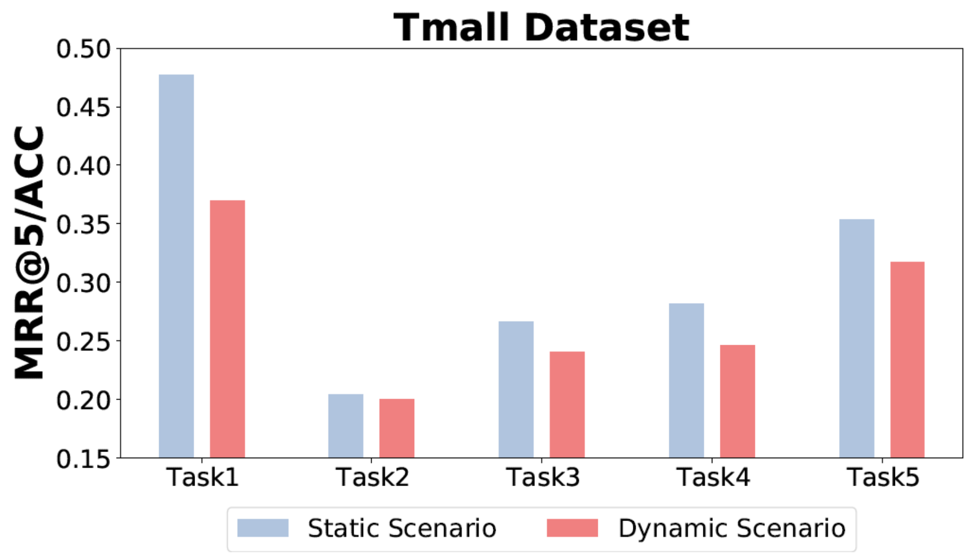
Inference Phase

- Evaluation is conducted on all preceding tasks at the point when the **final task is completed**



Motivation

Existing Method



TERACON utilize the data for the **entire time period equally** to all tasks → **unrealistic!**

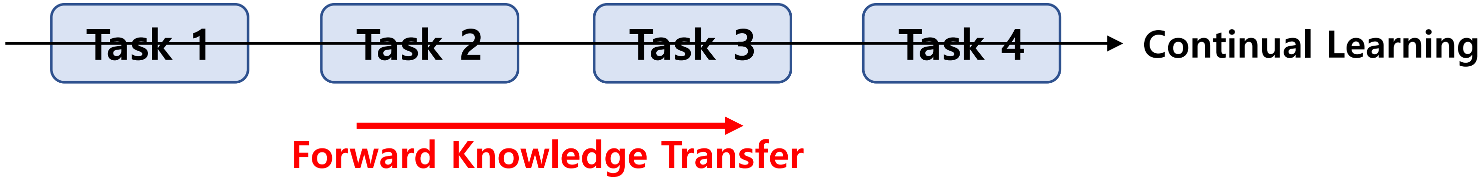
The failure to perform proper knowledge transfer in the **shifted distribution** → **Catastrophic forgetting!**

Kim, Sein, et al. "Task relation-aware continual user representation learning." *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 2023.

Challenges

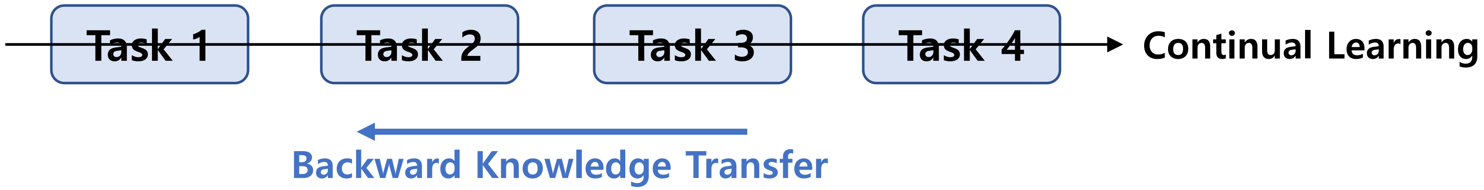
Forward Knowledge Transfer

- Transferring the knowledge learned from past tasks to the current task to ensure that the knowledge acquired from previous tasks is not forgotten → avoid Catastrophic Forgetting



Backward Knowledge Transfer

- As new items are added over time, the distribution shifts
- The knowledge from the current task is transferred to the past task to help it adapt to the current distribution



➔ **D**ynam**I**c **T**ime-aware Con**T**inual User Representati**O**n Learning (**DITTO**)

  - Investigates how/what/why to forward/backward transfer under continuous shifts in item distribution

Preliminaries

Notation

- Sequence of Tasks : $\mathcal{T} = \{T_1, T_2, ..., T_i, ..., T_M\}$

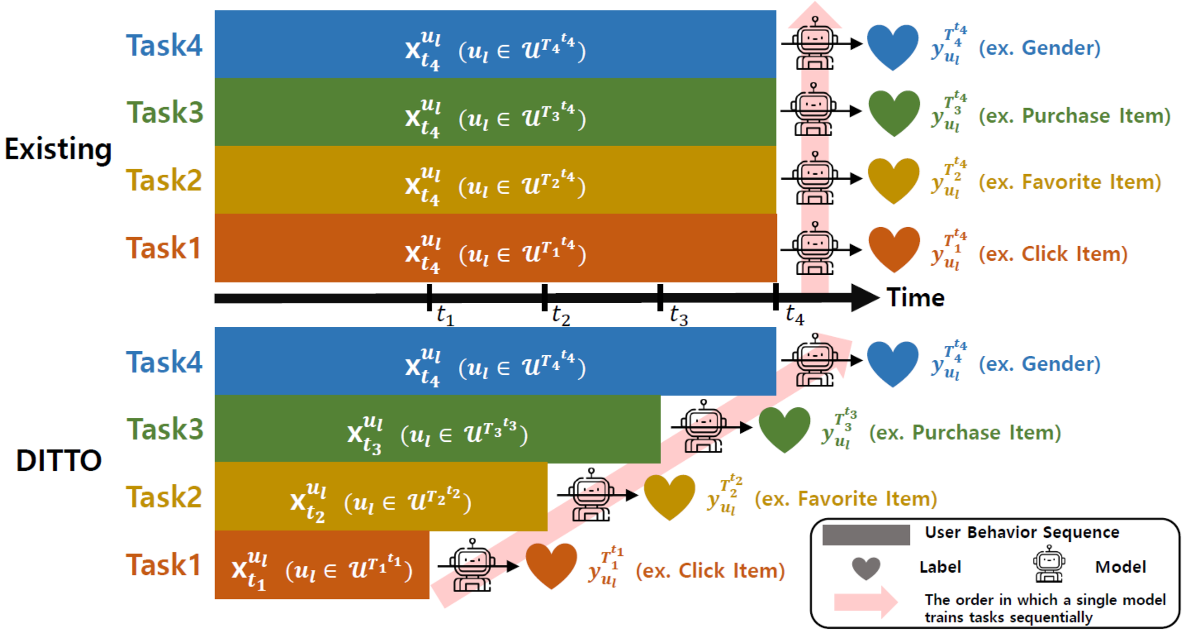
$$\Big\downarrow \quad \text{timestamp} \quad t = \{t_1, t_2, ..., t_j, ..., t_M\}$$

$$\mathcal{T}^t = \left\{T_1^{t_1}, T_2^{t_2}, ..., T_k^{t_k}, ..., T_M^{t_M}\right\}$$

- User across entire tasks : $\mathcal{U} = \{u_1, u_2, ..., u_N\}$

- Set of items : $\mathcal{I}$

- Behavior sequence of u_l up to the time point t_j : $\mathbf{x}_{t_j}^{u_l} = \{x_1^{u_l}, x_2^{u_l}, ..., x_n^{u_l}\}$



Proposed Method

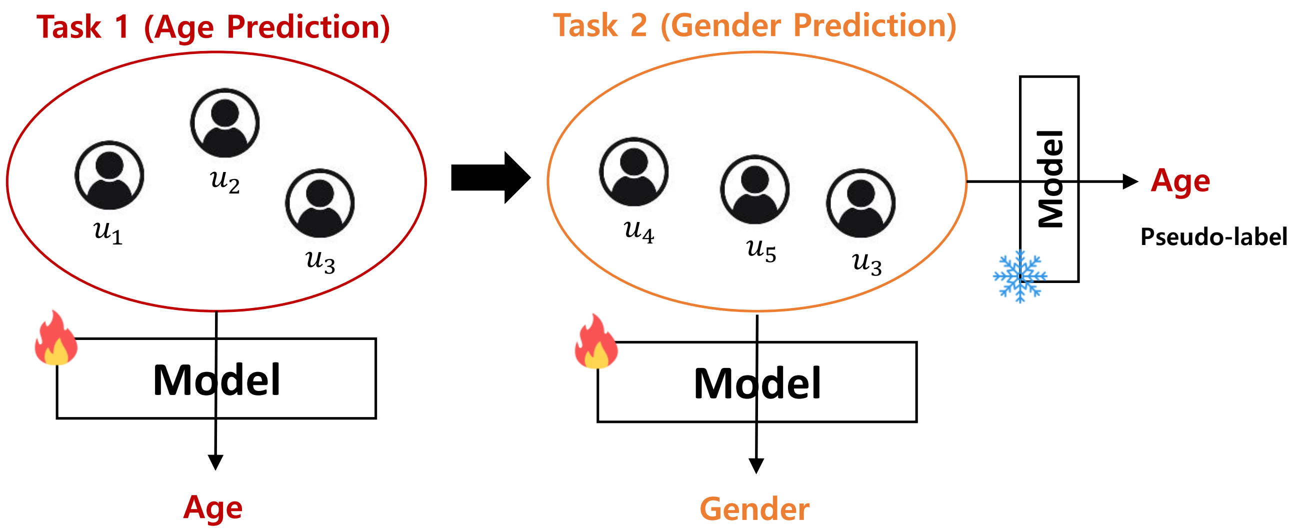
Distribution-aware Forward Knowledge Transfer (FKT)

- Transferring knowledge **from previous tasks to the current task** by considering the relationship between tasks

- Previous work used **Pseudo-labeling**!
  - 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.,
    Task 1 : Learn the age of users u_1, u_2, u_3
    Task 2: Predict the gender of users u_3, u_4, u_5
    current input → Generates pseudo-labels
    for the ages of u_3, u_4, u_5
    By training on these pseudo-labels,
    it is possible to retain the age information for u_3
    and learn the age information of u_4, u_5

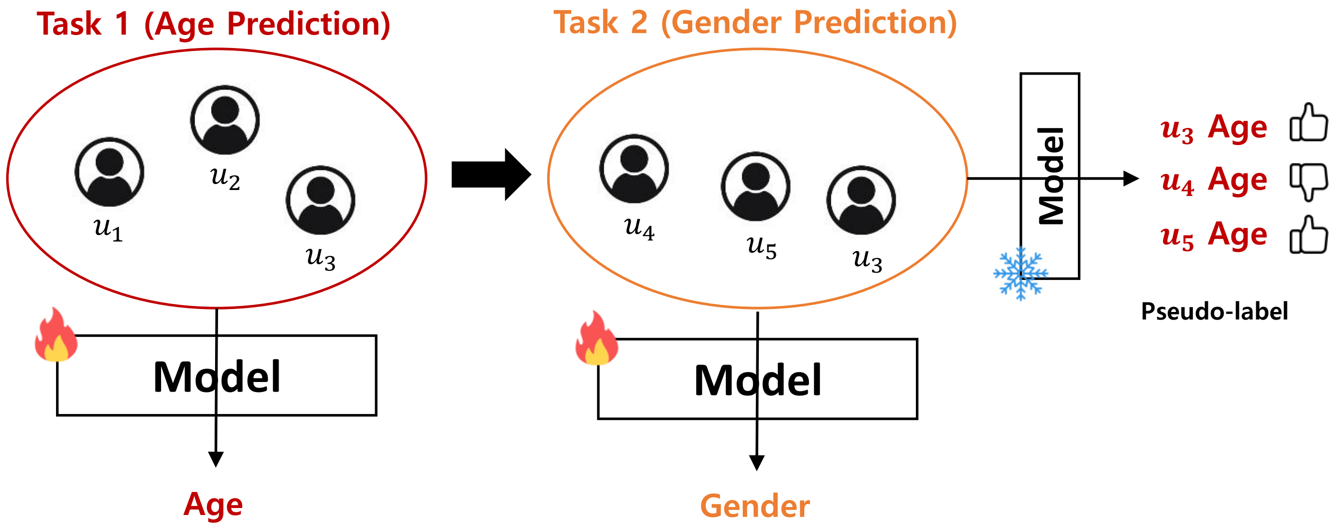
  - By training on pseudo-labels,
    the knowledge from past tasks can be preserved



Proposed Method

Distribution-aware Forward Knowledge Transfer (FKT)

- Reliability of Pseudo-labels
  - Due to the continuous emergence of new items as the task progresses, **reliable pseudo-labels cannot be generated**



➔ User behavior sequences that allow for generating reliable pseudo-labels through **distribution-aware sampling strategy**!

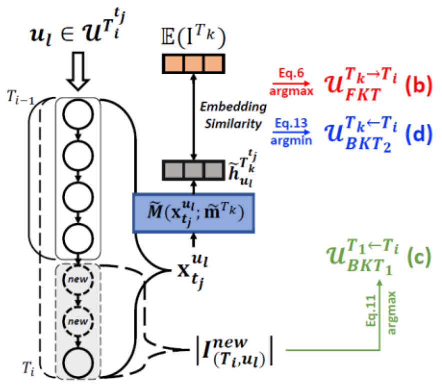
	Tmall				
	T_1	T_2	T_3	T_4	T_5
w/o. sampling	-0.013	0.0271	0.0258	0.0277	0.0138
w. sampling	0.3912	0.3413	0.3974	0.4078	0.3827

Proposed Method

Distribution-aware Forward Knowledge Transfer (FKT)

- Distribution-aware User Sampling
  - Sampling user behavior sequences that have a **distribution similar** to the one during the learning of past tasks
  - Sampling the Top K **most similar** user behavior sequences for past tasks by comparing the representation of the user behavior sequence with the average of the item embeddings after the past task has been learned

$$\mathcal{U}_{FKT}^{T_k \to T_i} = \underset{u_l}{\mathrm{argmax}}^{(S_{i,k})} \cos\left(\tilde{\mathrm{h}}_{u_l}^{T_k^{t_j}}, \mathbb{E}[\mathrm{I}^{T_k}]\right), \quad u_l \in \mathcal{U}_{T_i}^{t_j}$$



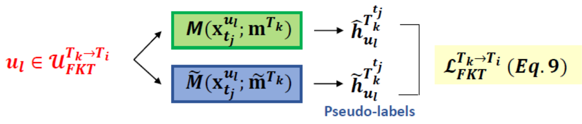
(a) User Sampling Strategy

➔ The reliability of the pseudo-labels improves when compared to the actual labels

	Tmall				
	T_1	T_2	T_3	T_4	T_5
w/o. sampling	-0.013	0.0271	0.0258	0.0277	0.0138
w. sampling	0.3912	0.3413	0.3974	0.4078	0.3827

➔ The model is able to effectively retain the past knowledge that needs to be remembered, excluding the shifted distribution!

$$\mathcal{L}_{FKT}^{T_k \to T_i} = \mathbb{E}_{u_l}\left[L_{MSE}\left(\mathcal{M}(\mathrm{x}_{t_j}^{u_l}; \mathrm{m}^{T_k}), \tilde{\mathrm{h}}_{u_l}^{T_k^{t_j}}\right)\right], \quad u_l \in \mathcal{U}_{FKT}^{T_k \to T_i}$$

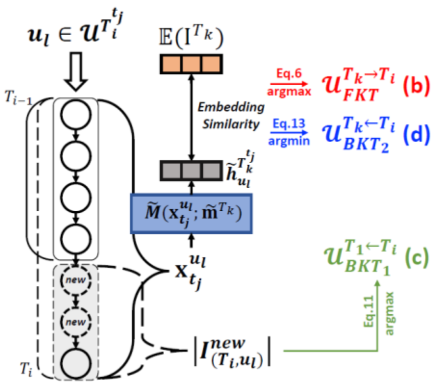


Proposed Method

Distribution-aware Backward Knowledge Transfer (BKT)

- Transferring knowledge to allow **past tasks to adapt** to the shifted distribution as new items are added
- Distribution-aware User Sampling
  - Sampling user behavior sequences that have a **distribution different** from the one during the learning of past tasks
  - Sampling the Top K **most dissimilar** user behavior sequences for past tasks by comparing the representation of the user behavior sequence with the average of the item embeddings after the past task has been learned

$$\mathcal{U}_{BKT_2}^{T_k \leftarrow T_i} = \underset{u_l}{\operatorname{argmin}}^{(S_{i,k})} \cos\left(\tilde{\mathbf{h}}_{u_l}^{T_k^{t_j}}, \mathbb{E}[\mathbf{I}^{T_k}]\right), \quad u_l \in \mathcal{U}_{T_i}^{t_j}$$

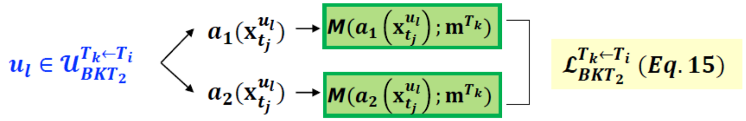


(a) User Sampling Strategy

- Backward Knowledge Transfer through Contrastive Learning
  - Since reliable pseudo-labels cannot be generated, an **unsupervised learning** approach is utilized
  - Applying contrastive learning by augmenting the sampled user behavior sequence with two different views

$$\mathcal{L}_{cl}^{u_l} = -\log \frac{\exp\left(\mathrm{sim}\left(a_1(\mathbf{x}_{t_j}^{u_l}), a_2(\mathbf{x}_{t_j}^{u_l})\right)\right)}{\exp\left(\mathrm{sim}\left(a_1(\mathbf{x}_{t_j}^{u_l}), a_2(\mathbf{x}_{t_j}^{u_l})\right)\right) + \sum_{s^- \in S^-} \exp\left(\mathrm{sim}\left(a_1(\mathbf{x}_{t_j}^{u_l}), s^-\right)\right)}$$

$$\mathcal{L}_{BKT_2}^{T_k \leftarrow T_i} = \mathbb{E}_{u_l}\left[\mathcal{L}_{cl}^{u_l}\right], \quad u_l \in \mathcal{U}_{BKT_2}^{T_k \leftarrow T_i}$$



➔ The past task can adapt to the current shifted distribution!

Experiments

Datasets & Tasks Descriptions

- Tmall
  - User Behavior Modeling
    - T_1 : (userID, recent 100 clicking interactions)
  - Item Recommendation
    - T_2 : (userID, the item that the user put in the cart)
    - T_3 : (userID, the item purchased by user)
    - T_4 : (userID, the item favored by the user)
  - Profile Prediction
    - T_5 : (userID, age)
    - T_6 : (userID, gender)
- ML (Movie Lens)
  - User behavior modeling
    - T_1 : (userID, recent 30 clicking interactions)
  - Item Recommendation
    - T_2 : (userID, an item that is rated higher than 4)
    - T_3 : (userID, one of 5-star items)
- Taobao
  - User behavior modeling
    - T_1 : (userID, recent 50 page-view interactions)
  - Item Recommendation
    - T_2 : (userID, the item that the user put in the cart)
    - T_3 : (userID, the item favored by the user)
    - T_4 : (userID, the item purchased by user)

Dataset	Task 1 ($T_1^{t_1}$)		Task 2 ($T_2^{t_2}$)		Task 3 ($T_3^{t_3}$)		Task 4 ($T_4^{t_4}$)		Task 5 ($T_5^{t_5}$)		Task 6 ($T_6^{t_6}$)	
	$\lvert\mathcal{U}^{T_1^{t_1}}\rvert$	$\lvert\mathcal{Y}^{T_1^{t_1}}\rvert$	$\lvert\mathcal{U}^{T_2^{t_2}}\rvert$	$\lvert\mathcal{Y}^{T_2^{t_2}}\rvert$	$\lvert\mathcal{U}^{T_3^{t_3}}\rvert$	$\lvert\mathcal{Y}^{T_3^{t_3}}\rvert$	$\lvert\mathcal{U}^{T_4^{t_4}}\rvert$	$\lvert\mathcal{Y}^{T_4^{t_4}}\rvert$	$\lvert\mathcal{U}^{T_5^{t_5}}\rvert$	$\lvert\mathcal{Y}^{T_5^{t_5}}\rvert$	$\lvert\mathcal{U}^{T_6^{t_6}}\rvert$	$\lvert\mathcal{Y}^{T_6^{t_6}}\rvert$
Tmall	Click		Cart		Purchase		Favorite		Age		Gender	
	355K	525K	1.29K	526K	65K	591K	54K	648K	393K	6	402K	2
ML	Click		4-star		5-star		-		-		-	
	53K	4K	1.2K	4.5K	6.1K	6.8K						
Taobao	Page View		Cart		Favorite		Buy		-		-	
	434K	1.47M	311K	1.63M	295K	1.83M	321K	2.00M				

Experiments

Overall Performance

		Tmall						ML			Taobao			
		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
Trains a single model for each task from scratch	SinMo	0.3002	0.1032	0.2234	0.2135	0.3327	0.7432	0.3603	0.0804	0.4531	0.3340	0.2087	0.2417	0.3526
Transfer Learning ($T_1 \rightarrow T_i$)	FineAll	0.3022	0.1101	0.1568	0.1116	0.2831	0.7219	0.3603	0.0671	0.4015	0.3340	0.3081	0.3288	0.3442
	PeterRec	0.3022	0.1096	0.1588	0.1135	0.3165	0.7458	0.3603	0.0785	0.4565	0.3340	0.3393	0.3268	0.3691
Multi-task Learning	MTL	-	0.1327	0.2392	0.1367	0.2674	0.7070	-	0.0685	0.4261	-	0.2523	0.2737	0.3120
Continual Learning	Piggyback	0.3022	0.1106	0.0944	0.0680	0.2638	0.6993	0.3603	0.0741	0.4108	0.3340	0.1848	0.1800	0.2868
	HAT	0.3597	0.1712	0.1726	0.1555	**0.3557**	0.7378	0.3517	0.0518	0.4592	0.2439	0.3001	0.3599	0.3488
	CONURE	0.3366	0.1417	0.2145	0.1927	0.3103	0.6994	0.3635	0.0625	0.4779	0.3241	0.3349	0.3569	0.4469
	TERACON	0.3698	0.2002	0.2405	0.2462	0.3170	0.7411	0.3755	0.0816	0.5094	0.2803	0.3272	0.4843	0.4858
	DITTO	**0.6102**	**0.2764**	**0.3058**	**0.3345**	0.3209	**0.7496**	**0.4168**	**0.0890**	**0.5107**	**0.4291**	**0.3407**	**0.4903**	**0.4960**

- Transferring knowledge without considering the passage of time can lead to negative transfer (SinMo vs. TL/CL)
- Continual learning-based methods perform better than Transfer learning-based method
- DITTO outperforms the SinMo & Continual learning-based method
    - Modeling by considering the **relation between tasks and the passage of time**

Experiments

Ablation Study

- Overall performance with and without the FKT and BKT modules,
  and performance for (users who interacted only with existing items / users who interacted with new items)

Row	Component		T_1	T_2	T_3	T_4	T_5	T_6
	FKT	BKT						
(1)	✗	✗	0.2572 (0.2744 / 0.2517)	0.1994 (0.2056 / 0.1975)	0.2309 (0.2487 / 0.2220)	0.2170 (0.2217 / 0.2162)	0.2281 (0.2317 / 0.2175)	0.7446 (0.7446 / -)
(2)	✓	✗	0.2925 (0.3903 / 0.2627)	0.1955 (0.2248 / 0.1825)	0.2371 (0.2685 / 0.2162)	0.2401 (0.2652 / 0.2226)	**0.3240** (0.3484 / 0.3016)	0.7392 (0.7392 / -)
(3)	✗	✓	0.5717 (0.5624 / 0.5763)	0.2514 (0.2423 / 0.2524)	0.2799 (0.2780 / 0.2807)	0.3069 (0.3013 / 0.3105)	0.3061 (0.3059 / 0.3067)	0.7445 (0.7445 / -)
(4)-1	✓	✓ (only $\mathcal{L}_{BKT_1}$)	0.6073 (0.6092 / 0.6059)	0.2506 (0.2538 / 0.2475)	0.2942 (0.3063 / 0.2869)	0.3010 (0.3165 / 0.2886)	0.3184 (0.3163 / 0.3251)	0.7450 (0.7450 / -)
(4)-2	✓	✓ (only $\mathcal{L}_{BKT_2}$)	0.2816 (0.3878 / 0.2500)	0.2039 (0.2095 / 0.1984)	0.2455 (0.2516 / 0.2322)	0.2266 (0.2371 / 0.2233)	0.2836 (0.2885 / 0.2792)	0.7478 (0.7478 / -)
(5)	✓ (*random*)	✓ (*random*)	0.5968 (0.5804 / 0.6066)	0.1221 (0.1242 / 0.1190)	0.1418 (0.1491 / 0.1288)	0.1579 (0.1652 / 0.1437)	0.3182 (0.3212 / 0.3173)	0.7477 (0.7477 / -)
(6)	✓	✓	**0.6102** (0.6044 / 0.6099)	**0.2764** (0.2801 / 0.2722)	**0.3058** (0.3162 / 0.3013)	**0.3345** (0.3375 / 0.3338)	0.3209 (0.3195 / 0.3256)	**0.7496** (0.7496 / -)

- FKT module is beneficial for users who interacted with **existing items** (Row (1) vs. (2))
- BKT module is beneficial for users who have interacted with **newly emerged items** (Row (1) vs. (3))
- Distribution-aware sampling strategy helps in utilizing users that are suitable for the objectives of each module (Row (5) vs. (6))

Experiments

Order Robustness

(a) Original	T_1		T_2		T_3		T_4		T_5		T_6	
	MRR@5	KT	MRR@5	KT	MRR@5	KT	ACC	KT	ACC	KT	ACC	KT
SinMo	0.3002	-	0.1032	-	0.2234	-	0.2135	-	0.3327	-	0.7432	-
HAT	0.3597	19.82%	0.1712	65.89%	0.1726	-22.74%	0.1555	-27.17%	0.3557	6.91%	0.7378	-0.73%
CONURE	0.3366	12.13%	0.1417	37.31%	0.2145	-3.98%	0.1927	-9.74%	0.3103	-6.73%	0.6994	-5.89%
TERACON	0.3698	23.18%	0.2002	93.99%	0.2405	7.65%	0.2462	15.32%	0.3170	-4.72%	0.7411	-0.28%
DITTO	**0.6102**	103.26%	**0.2764**	167.83%	**0.3058**	36.88%	**0.3345**	56.67%	**0.3209**	-3.55%	**0.7496**	0.86%

(b) Reversed	T_1		T_6		T_5		T_4		T_3		T_2	
	MRR@5	KT	ACC	KT	ACC	KT	ACC	KT	MRR@5	KT	MRR@5	KT
SinMo	0.3222	-	0.7133	-	**0.3768**	-	0.2913	-	0.3360	-	0.4062	-
HAT	0.4309	33.74%	0.6695	-6.14%	0.3522	-6.53%	0.4465	53.27%	0.4625	37.65%	0.4695	15.58%
CONURE	0.3366	4.47%	0.7188	0.77%	0.3678	-2.39%	0.2460	-15.55%	0.3136	-6.67%	0.4174	2.76%
TERACON	0.4874	51.27%	0.6768	-5.12%	0.3680	-2.34%	0.4340	48.99%	0.4507	34.14%	0.4504	10.88%
DITTO	**0.5430**	68.53%	**0.7678**	7.64%	0.3700	-1.80%	**0.4816**	65.33%	**0.4677**	39.20%	**0.4710**	15.95%

- DITTO maintains its superiority even when the **order of tasks is changed**
  - Verifying the effectiveness of maximizing positive knowledge transfer between tasks regardless of the task order
- TERACON fails to generate reliable pseudo-labels for users with shifted item distributions, leading to negative transfer

➔ DITTO is order robust framework

Experiments

Noise Robustness

- Inserting a noisy task (i.e., T_{noise}) between T_3 and T_4, which contains noisy labels

	Tmall						
	T_1	T_2	T_3	T_{noise}	T_4	T_5	T_6
HAT	0.3495 (-2.84 %)	0.1612 (-5.84 %)	0.1553 (-10.02 %)	-	0.1021 (-34.34 %)	0.3145 (-11.58 %)	0.7211 (-2.26 %)
CONURE	0.3366 (0.0 %)	0.1417 (0.0 %)	0.2145 (0.0 %)	-	0.1526 (-20.81 %)	0.3020 (-2.67 %)	0.6901 (-1.33 %)
TERACON	0.3458 (-6.49 %)	0.1875 (-6.34 %)	0.1951 (-18.88 %)	-	0.2261 (-8.16 %)	0.3004 (-5.24 %)	0.7343 (-0.92 %)
DITTO	**0.5967** (-2.21 %)	**0.2697** (-2.42 %)	**0.2854** (-6.67 %)	-	**0.3125** (-6.58 %)	**0.3137** (-2.24 %)	**0.7455** (-0.55 %)

- Only user behavior sequences, **not label information**, are used for pseudo-label generation

  → forward knowledge transfer is possible without being affected by noisy labels
- Backward knowledge transfer is performed in an **unsupervised manner**, it is not affected by noisy labels

➔ DITTO is noise robust framework

Conclusion

Summary

- Continual user representation learning framework that maximizes **knowledge transfer** between tasks under **time-driven distribution shifts**

Contribution

- Propose a continual user representation learning scenario that accounts for the **passage of time**

- Develop **distribution-aware user sampling strategies and transfer methods** tailored to each objective to maximize positive knowledge transfer between tasks

- Extensive experiments demonstrate the effectiveness and efficiency of DITTO

Thank you!

[Full Paper] https://arxiv.org/abs/2504.16501

[Source Code] https://github.com/seungyoon-Choi/DITTO_official

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

[Email] csyoon08@kaist.ac.kr

Paper

Code