The Forty-first International Conference on Machine Learning

NEURAI

# Gradual Divergence for Seamless Adaptation: A Novel Domain Incremental Learning Method

Kishaan Jeeveswaran, Elahe Arani*, Bahram Zonooz*

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

ICML International Conference On Machine Learning

# Introduction

- **Domain incremental learning** (DIL) is a subset of continual learning that addresses the challenge of acquiring knowledge from new domains or tasks in an incremental manner.

- Catastrophic forgetting, phenomenon wherein the network forgets information about previous tasks, makes learning new domains challenging.

- Though different approaches like experience replay, regularization, and architecture expansion have been proposed to mitigate catastrophic forgetting, they don’t explicitly address representation drift caused by the weight updates.

# Motivation

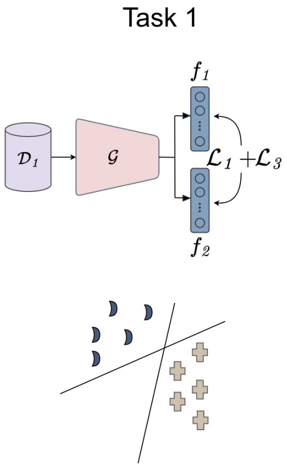
- Representation drift at task boundaries are caused by the disruption of clustered representations corresponding to previously learned tasks while learning a new task/domain*.

- Representation drift is directly correlated with performance drop on old tasks and contributes significantly to catastrophic forgetting and it remains unexplored in DIL literature.

- To address this, we suggest a three-stage training process to gradually adapt the learning model to new domain sample representations.

Caccia, Lucas, et al. "New insights on reducing abrupt representation change in online continual learning." *arXiv preprint arXiv:2104.05025* (2021).

# DARE - Methodology



Task 1

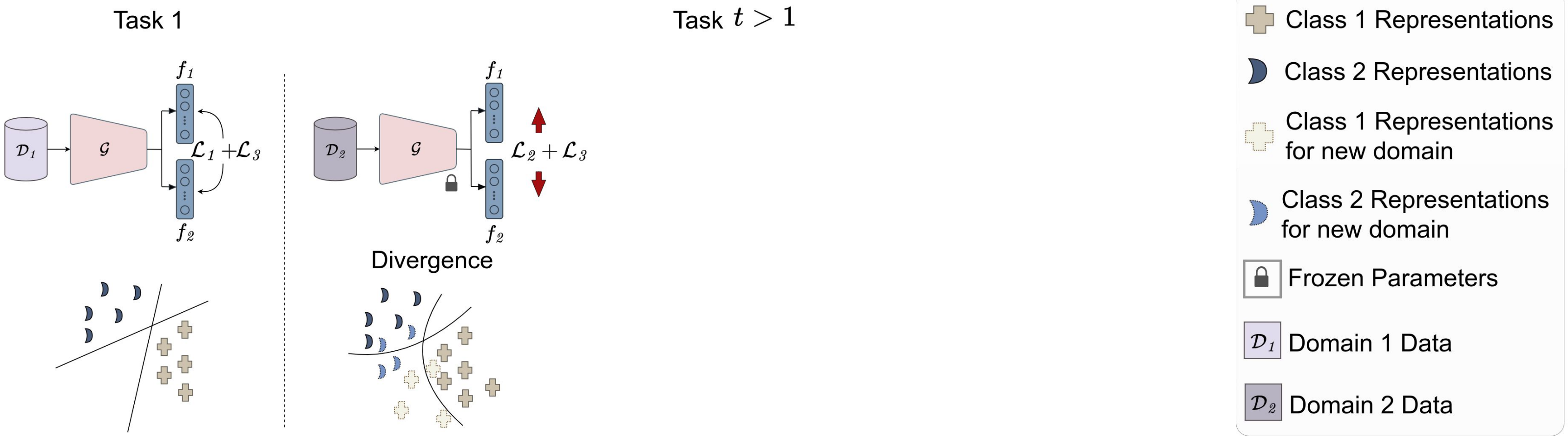
- ✚ Class 1 Representations
- ☽ Class 2 Representations
- ✚ Class 1 Representations for new domain
- ☽ Class 2 Representations for new domain
- 🔒 Frozen Parameters
- $\mathcal{D}_1$ Domain 1 Data
- $\mathcal{D}_2$ Domain 2 Data

- Experience replay has been shown to effectively mitigate catastrophic forgetting. However, it lacks mechanism to explicitly address drift in representations at task boundaries.

- We adopt a **two-classifier architecture** which learns on the first task similar to other experience replay methods* using cross-entropy loss ($L_1$) and consistency loss on predicted logits ($L_3$).

Buzzega, Pietro, et al. "Dark experience for general continual learning: a strong, simple baseline." *Advances in neural information processing systems* 33 (2020): 15920-15930.

# DARE - Methodology



Class 1 Representations

Class 2 Representations

Class 1 Representations for new domain

Class 2 Representations for new domain

Frozen Parameters

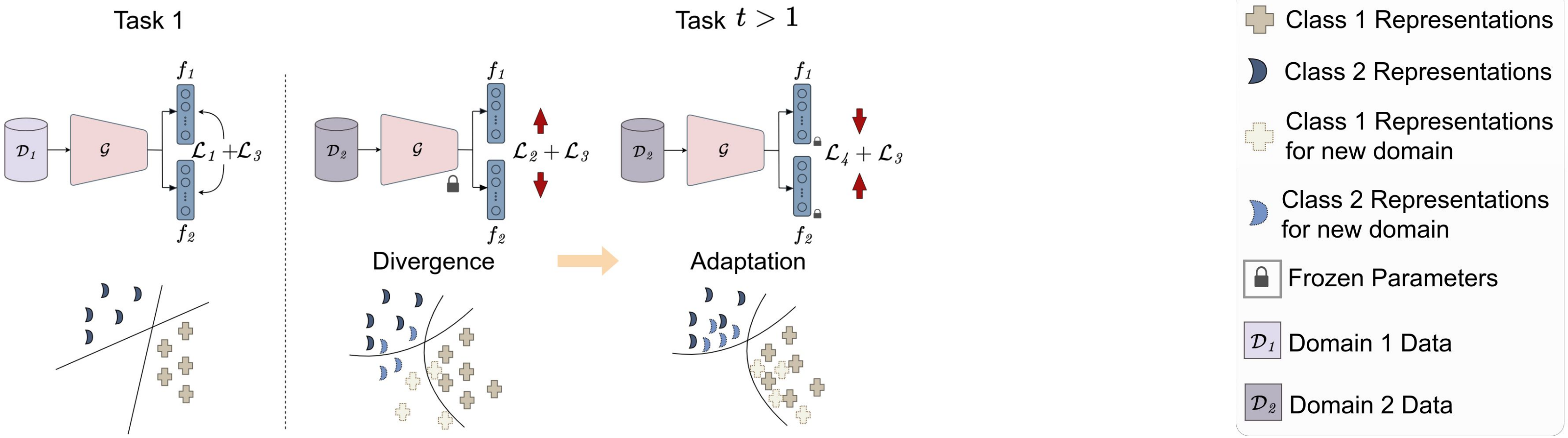
$\mathcal{D}_1$ Domain 1 Data

$\mathcal{D}_2$ Domain 2 Data

- DARE proposes a novel **three-stage training process** (**D**ivergence, **A**daptation, and **RE**finement) to adapt the representations of new domain samples into the feature space spanned by the old domains while learning new domains.

- The **'Divergence'** stage aims to tighten the decision boundaries of the two classifiers around the representation space spanned by the samples of already learned tasks. This involves maximizing the divergence between the two classifiers ($L_2$).

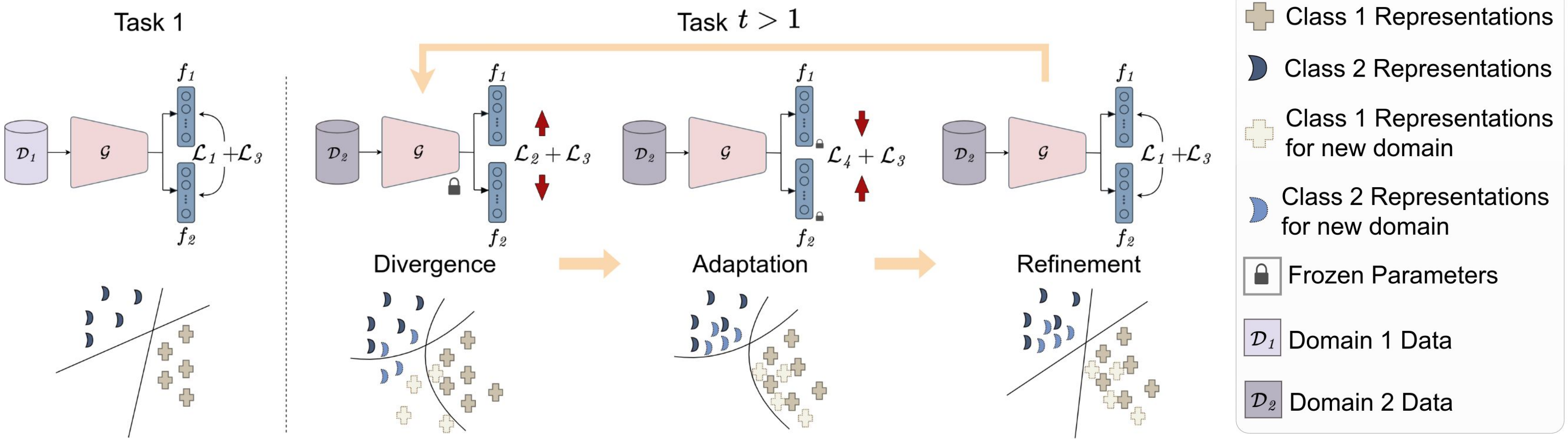
# DARE - Methodology


Task 1
Task $t > 1$
$\mathcal{D}_1$
$\mathcal{G}$
$f_1$
$f_2$
$\mathcal{L}_1 + \mathcal{L}_3$
$\mathcal{D}_2$
$\mathcal{L}_2 + \mathcal{L}_3$
Divergence
$\mathcal{L}_4 + \mathcal{L}_3$
Adaptation
Class 1 Representations
Class 2 Representations
Class 1 Representations for new domain
Class 2 Representations for new domain
Frozen Parameters
$\mathcal{D}_1$ Domain 1 Data
$\mathcal{D}_2$ Domain 2 Data

- Then, the **'Adaptation'** stage aims to adapt the encoder *g* so that the representations of the new task samples are adapted within the subspace spanned by the already learned representations of previous tasks.

- The goal is to learn a consolidated representation space that supports the samples of the new tasks while remaining close to the optimal representations for the previously learned tasks. This is achieved by freezing the classifiers, and minimizing the discrepancy between their predictions ($L_4$).

# DARE - Methodology


Task 1
Task $t > 1$
$f_1$
$f_2$
$\mathcal{D}_1$
$\mathcal{G}$
$\mathcal{L}_1 + \mathcal{L}_3$
$\mathcal{D}_2$
$\mathcal{L}_2 + \mathcal{L}_3$
$\mathcal{L}_4 + \mathcal{L}_3$
Divergence
Adaptation
Refinement
Class 1 Representations
Class 2 Representations
Class 1 Representations for new domain
Class 2 Representations for new domain
Frozen Parameters
$\mathcal{D}_1$ Domain 1 Data
$\mathcal{D}_2$ Domain 2 Data

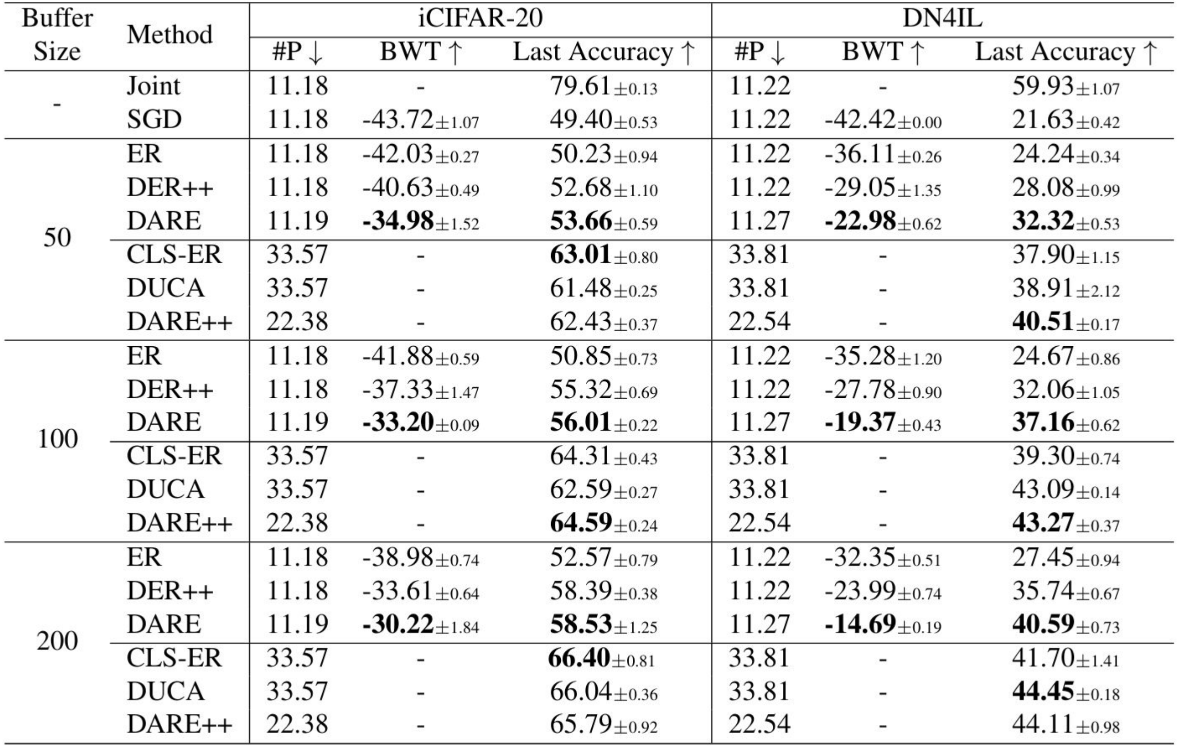
- **‘Refinement’** stage aims to refine the encoder and classifiers to effectively consolidate the new task information with the previously learned knowledge such that the consolidated representation space and decision boundary perform well on all the tasks.
- We iterate through the three stages multiple times while learning each task to consolidate the information in a structured manner.

# Intermediary Reservoir Sampling

- **Reservoir Sampling*** is a popular strategy to sample data points from tasks using uniform distribution and store them in the memory buffer.
- There is a nontrivial probability that logits are stored in the buffer at the very beginning or end of learning a task, leading to suboptimal performance.
- To improve this, we propose the "**Intermediary Reservoir Sampling (IRS)**" strategy, which employs a normal distribution over the learning trajectory of each task. The mean of the distribution is set to the intermediate stages, and the buffer is populated accordingly.
- This incentivizes the storage of logits with more "dark knowledge" about the current task, which in turn propagates the knowledge across future tasks through distillation.

Vitter, Jeffrey S. "Random sampling with a reservoir." *ACM Transactions on Mathematical Software (TOMS)* 11.1 (1985): 37-57.

# Experimental Results

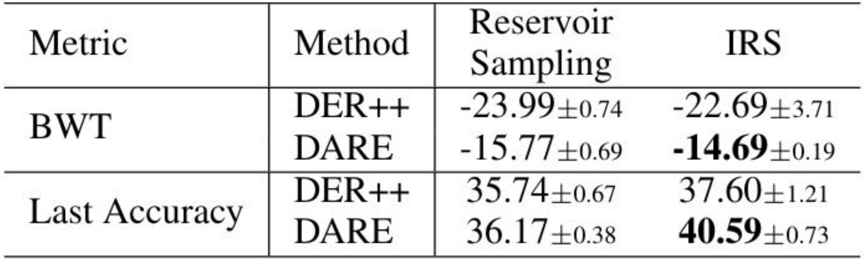


| Buffer Size | Method | iCIFAR-20 |  |  | DN4IL |  |  |
|---|---|---|---|---|---|---|---|
|  |  | #P ↓ | BWT ↑ | Last Accuracy ↑ | #P ↓ | BWT ↑ | Last Accuracy ↑ |
| - | Joint | 11.18 | - | 79.61±0.13 | 11.22 | - | 59.93±1.07 |
|  | SGD | 11.18 | -43.72±1.07 | 49.40±0.53 | 11.22 | -42.42±0.00 | 21.63±0.42 |
| 50 | ER | 11.18 | -42.03±0.27 | 50.23±0.94 | 11.22 | -36.11±0.26 | 24.24±0.34 |
|  | DER++ | 11.18 | -40.63±0.49 | 52.68±1.10 | 11.22 | -29.05±1.35 | 28.08±0.99 |
|  | DARE | 11.19 | **-34.98**±1.52 | **53.66**±0.59 | 11.27 | **-22.98**±0.62 | **32.32**±0.53 |
|  | CLS-ER | 33.57 | - | **63.01**±0.80 | 33.81 | - | 37.90±1.15 |
|  | DUCA | 33.57 | - | 61.48±0.25 | 33.81 | - | 38.91±2.12 |
|  | DARE++ | 22.38 | - | 62.43±0.37 | 22.54 | - | **40.51**±0.17 |
| 100 | ER | 11.18 | -41.88±0.59 | 50.85±0.73 | 11.22 | -35.28±1.20 | 24.67±0.86 |
|  | DER++ | 11.18 | -37.33±1.47 | 55.32±0.69 | 11.22 | -27.78±0.90 | 32.06±1.05 |
|  | DARE | 11.19 | **-33.20**±0.09 | **56.01**±0.22 | 11.27 | **-19.37**±0.43 | **37.16**±0.62 |
|  | CLS-ER | 33.57 | - | 64.31±0.43 | 33.81 | - | 39.30±0.74 |
|  | DUCA | 33.57 | - | 62.59±0.27 | 33.81 | - | 43.09±0.14 |
|  | DARE++ | 22.38 | - | **64.59**±0.24 | 22.54 | - | **43.27**±0.37 |
| 200 | ER | 11.18 | -38.98±0.74 | 52.57±0.79 | 11.22 | -32.35±0.51 | 27.45±0.94 |
|  | DER++ | 11.18 | -33.61±0.64 | 58.39±0.38 | 11.22 | -23.99±0.74 | 35.74±0.67 |
|  | DARE | 11.19 | **-30.22**±1.84 | **58.53**±1.25 | 11.27 | **-14.69**±0.19 | **40.59**±0.73 |
|  | CLS-ER | 33.57 | - | **66.40**±0.81 | 33.81 | - | 41.70±1.41 |
|  | DUCA | 33.57 | - | 66.04±0.36 | 33.81 | - | **44.45**±0.18 |
|  | DARE++ | 22.38 | - | 65.79±0.92 | 22.54 | - | 44.11±0.98 |

DARE archives:

- consistent improvements in last accuracy and BWT across different buffer sizes
- DARE++, which includes an EMA model, is more memory efficient than CLS-ER and DUCA while being competitive
- DARE and DARE++ outperform their counterparts with smaller buffers

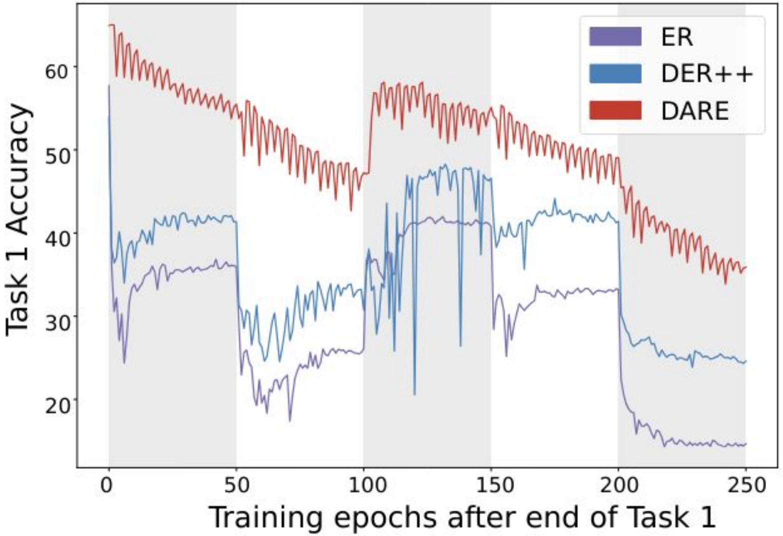
## Intermediary Reservoir Sampling Strategy

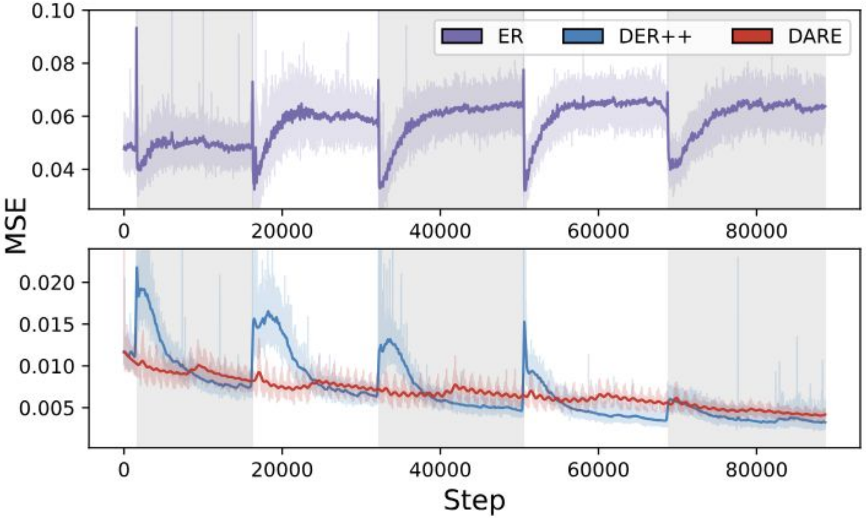


| Metric | Method | Reservoir Sampling | IRS |
|---|---|---|---|
| BWT | DER++ | -23.99±0.74 | -22.69±3.71 |
| | DARE | -15.77±0.69 | **-14.69**±0.19 |
| Last Accuracy | DER++ | 35.74±0.67 | 37.60±1.21 |
| | DARE | 36.17±0.38 | **40.59**±0.73 |

- IRS helps the method achieve better last accuracy and backward transfer.
- Storing the samples in the intermediate stages helps store more information about the task being learned in the buffer.
- Replaying samples stored in such manner helps the model consolidate more knowledge about old tasks.

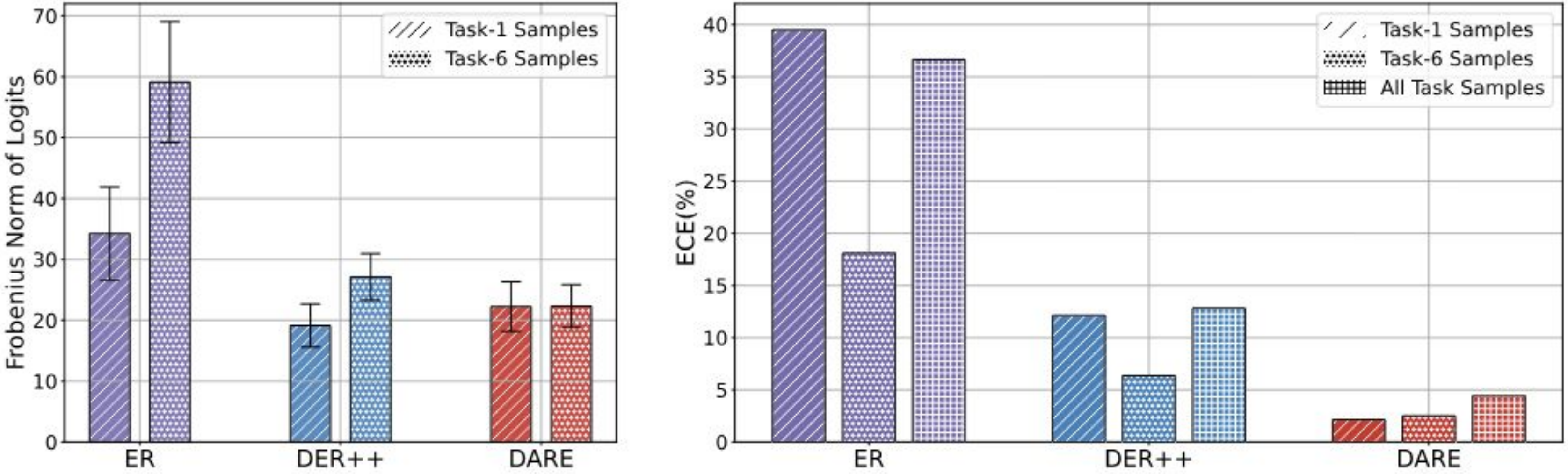
# Representation Drift Analysis





Left: Epoch-wise accuracy on Task 1 samples, while learning future tasks (shaded regions indicate new tasks). Right: Iteration-wise drifts for buffered samples for CL methods trained with a buffer size of 50. It is evident that DARE effectively reduces representation drift compared to other methods.

# Task Recency and Model Calibration Analysis



Task recency bias and model calibration analyses of different CL approaches learned with buffer size 200. Left: Logit norm analysis shows that DARE predicts logits with magnitudes smaller than DER++ (less overconfident) for recent task samples. Right: DARE has a lower calibration error compared to DER++ on samples belonging to different tasks

# Conclusions

- We proposed a novel method to **address representation drift in domain-incremental learning**.
- Our proposed method, DARE
  - outperforms existing methods across different DIL benchmarks
  - mitigates representation drift at task boundaries and effectively assimilates new domain information into the feature space of old task samples
  - **exhibits efficient memory and computational usage**
- The inclusion of an effective buffer sampling strategy (IRS) allows the preservation of the knowledge learned on old tasks.

# THANKS

Contact: Kishaan Jeeveswaran
Email: j.kishaan@tue.nl
Website: github.com/NeurAI-Lab