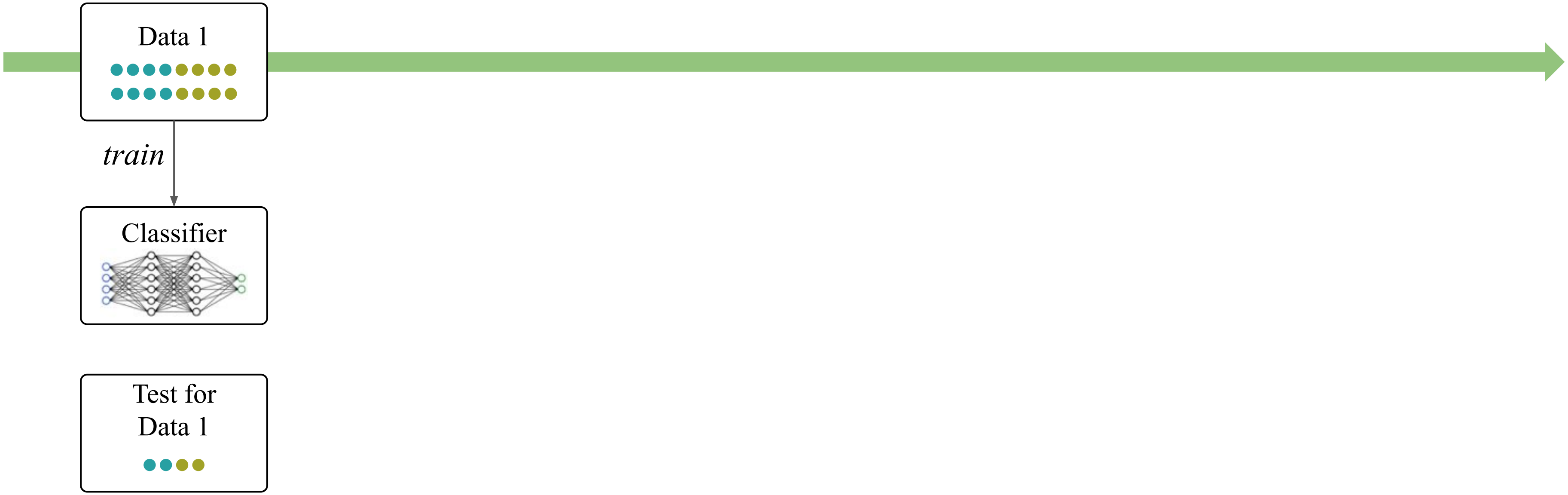
RMM: Reinforced Memory Management for Class-Incremental Learning

Yaoyao Liu,[1] Bernt Schiele,[1] Qianru Sun[2]

[1] Max Planck Institute for Informatics [2] Singapore Management University

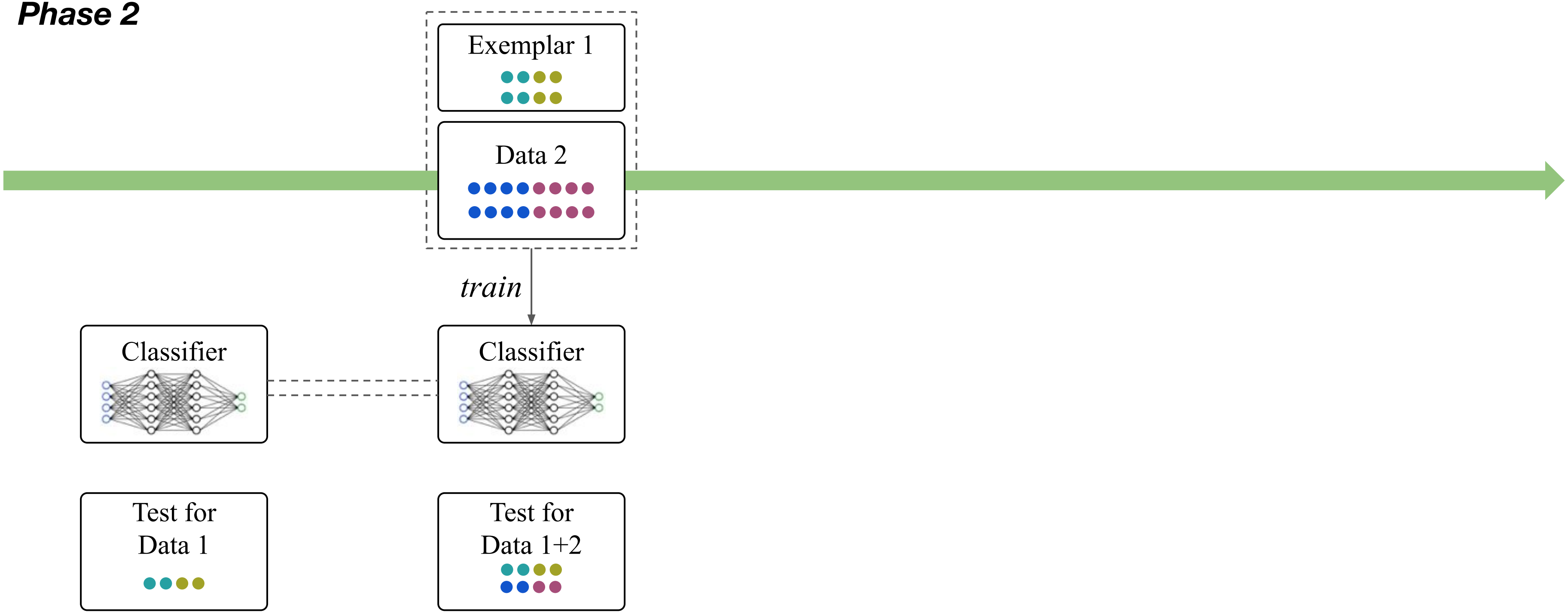
Research background: Class-Incremental Learning (CIL)

Phase 1



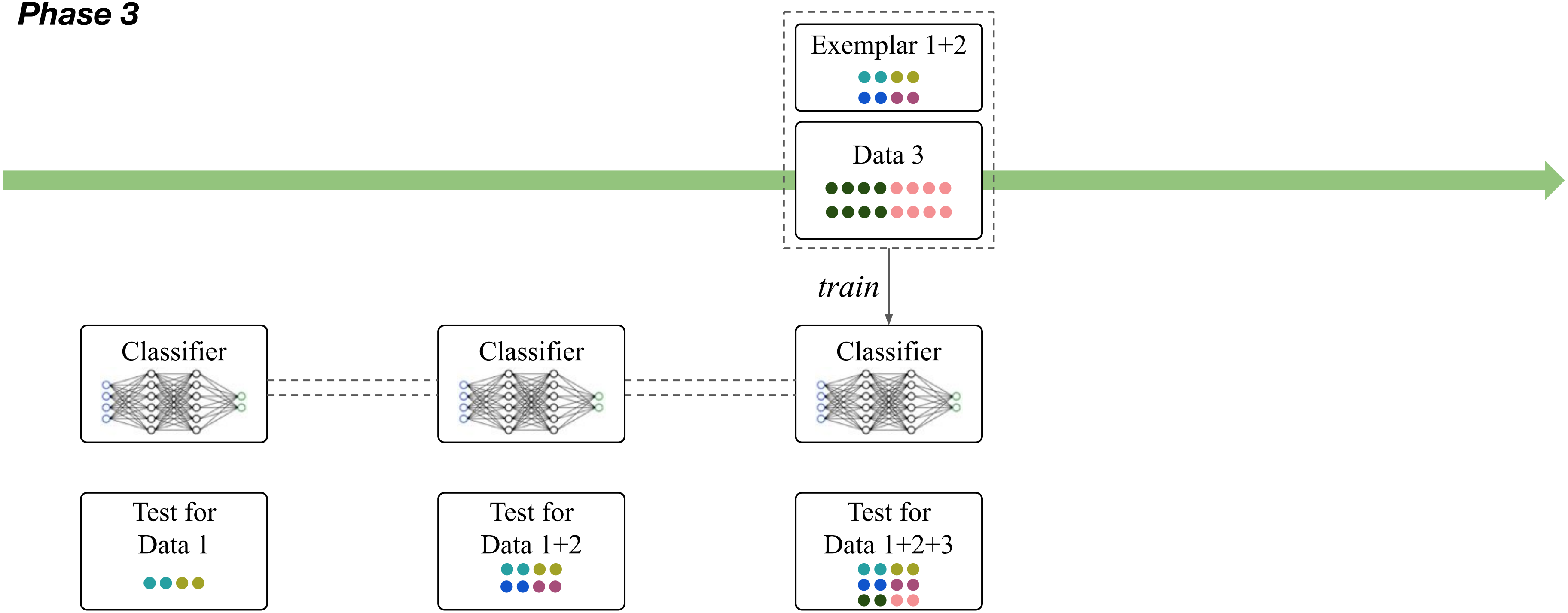
Research background: Class-Incremental Learning (CIL)

Phase 2

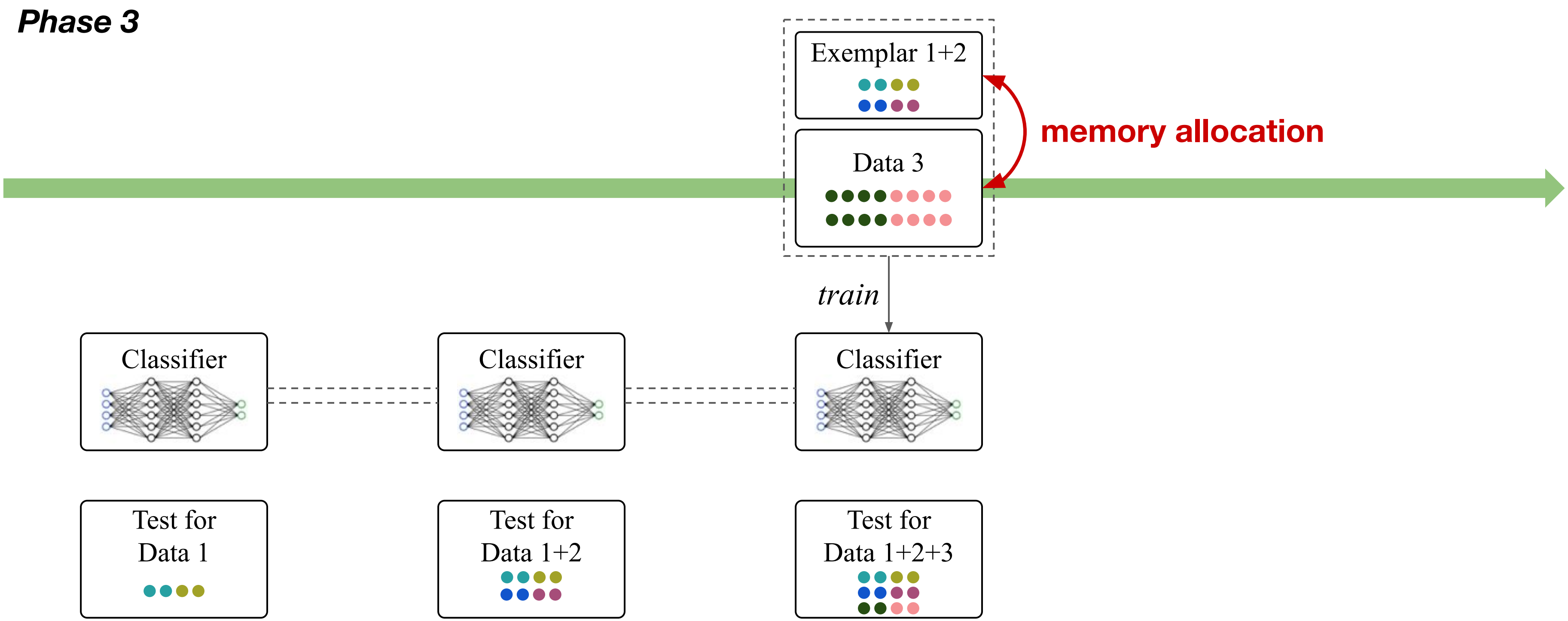


Research background: Class-Incremental Learning (CIL)

Phase 3

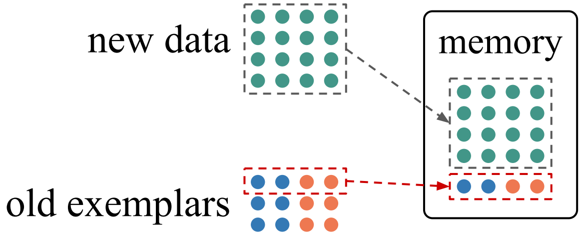


Research background: Class-Incremental Learning (CIL)



How to allocate the memory between old and new data?

Existing methods [1,2,3]
Allocate as much memory as possible for the new-class data



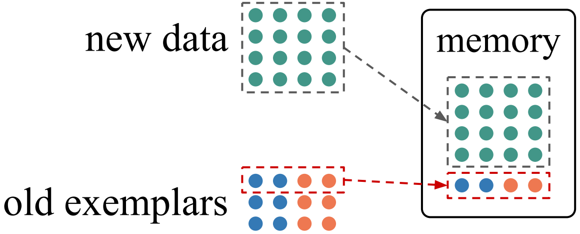
Limitations:
- Imbalance between old and new classes
- Catastrophic forgetting problem

References
[1] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." CVPR 2017;
[2] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019;
[3] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." TPAMI 2017.

How to allocate the memory between old and new data?

Existing methods [1,2,3]

Allocate as much memory as possible for the new-class data



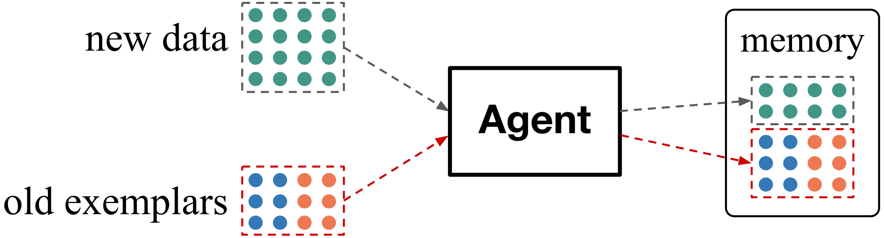
Limitations:

- Imbalance between old and new classes

- Catastrophic forgetting problem

Our method: Reinforced Memory Management (RMM)

Learn an agent using reinforcement learning to manage the memory allocation



Benefits:

+ Balancing the old and new classes

+ Overcoming the forgetting problem

References

[1] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." CVPR 2017;

[2] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019;

[3] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." TPAMI 2017.

max planck institut
informatik

SMU
SINGAPORE MANAGEMENT UNIVERSITY

What is a reinforcement learning (RL) system?

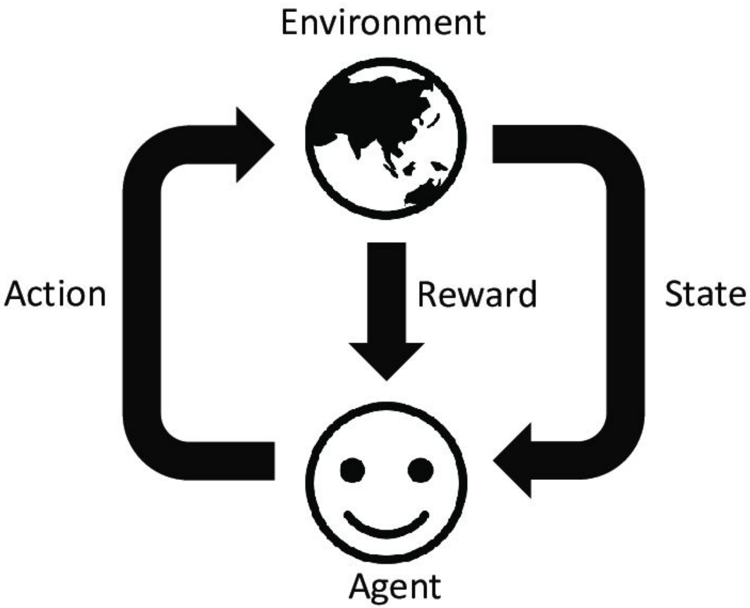
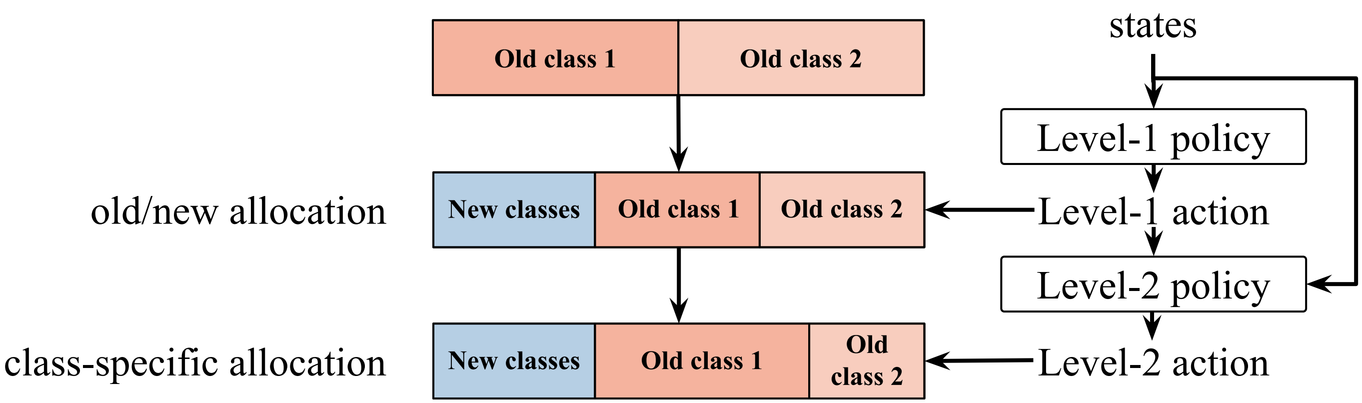


Image from https://europepmc.org/article/PMC/6767655

How to define the RL system for our CIL task?

- ***Actions***
  - Level-1: coarse (old/new) allocation
  - Level-2: fine-grained (class-specific) allocation



How to define the RL system for our CIL task?

- *Actions*
- ***States***
  - Distinct in each incremental phase
  - Transferable between CIL tasks

$$s_i = \left(\frac{\#\text{ new classes}}{\#\text{ old classes}}, \frac{\text{memory for old exemplars}}{\text{total memory}}\right)$$

How to define the RL system for our CIL task?

- *Actions*
- *States*
- ***Rewards: the validation accuracy***

How to define the RL system for our CIL task?

- *Actions*
- *States*
- *Rewards*
- ***Training data points***
  Due to the CIL protocol, we're not allowed to use the ***historical*** and ***future*** data

  Our solution: create many ***pseudo CIL tasks***, and train the RL system on them

How to create the pseudo CIL tasks?

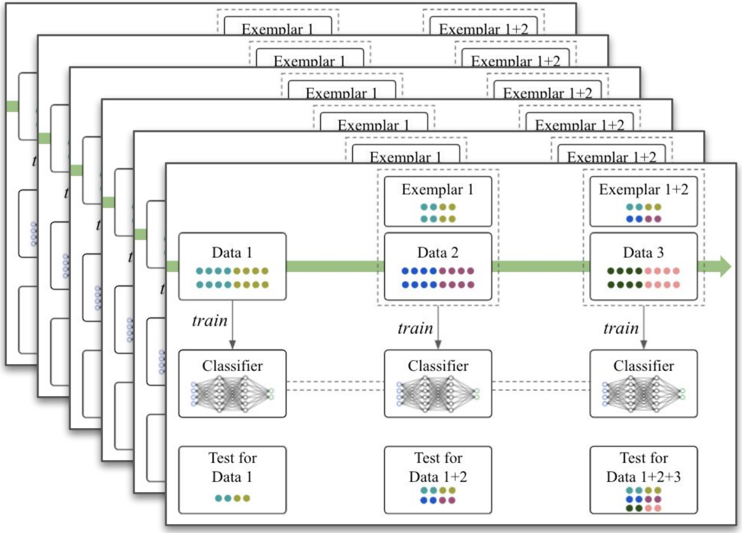
The data in the initial phase

OR

Another dataset

Create

Pseudo CIL tasks



Train

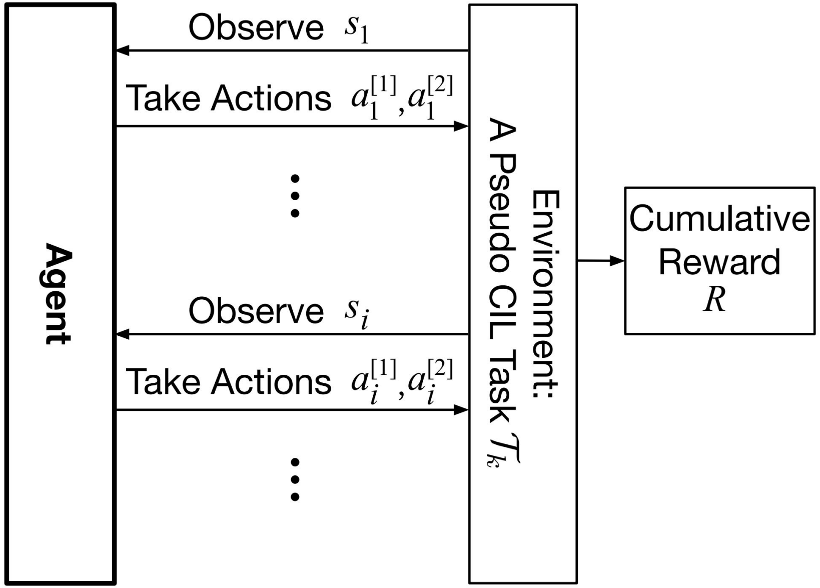
Agent

How to define the RL system for our CIL task?

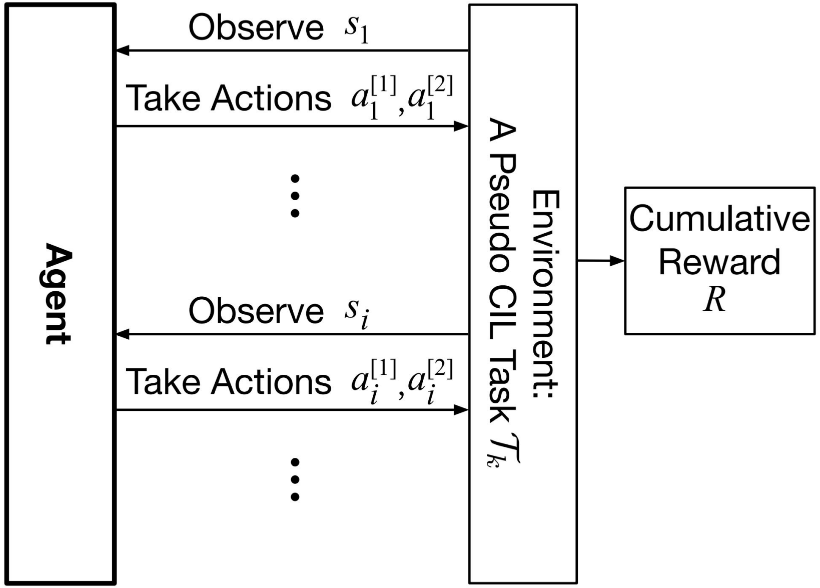
- *Actions*
- *States*
- *Rewards*
- *Training data points*
- ***Algorithm: the REINFORCE algorithm***[4]

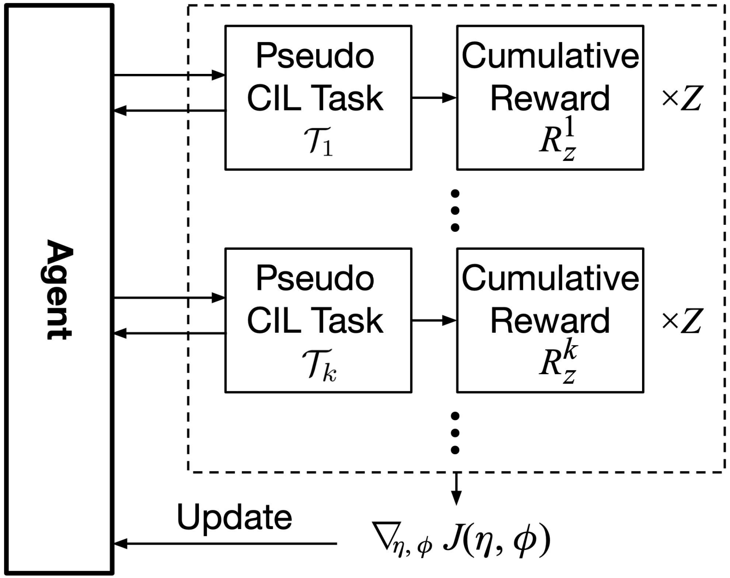
Reference
[4] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256, 1992.

How to learn the RL system using the REINFORCE algorithm?



How to learn the RL system using the REINFORCE algorithm?





Our RMM achieves SOTA performance

Method	CIFAR-100			ImageNet-Subset			ImageNet-Full		
	N=5	10	25	5	10	25	5	10	25
LwF	56.79	53.05	50.44	58.83	53.60	50.16	52.00	47.87	47.49
iCaRL	60.48	56.04	52.07	67.33	62.42	57.04	50.57	48.27	49.44
LUCIR	63.34	62.47	59.69	71.21	68.21	64.15	65.16	62.34	57.37
Mnemonics	64.59	62.59	61.02	72.60	71.66	70.52	65.40	64.02	62.05
PODNet	64.60	63.13	61.96	76.45	74.66	70.15	66.80	64.89	60.28
LUCIR-AANets	66.88	65.53	63.92	72.80	69.71	68.07	65.31	62.99	61.21
w/ RMM (ours)	68.42	67.17	64.56	73.58	72.83	72.30	65.81	64.10	62.23
POD-AANets	66.61	64.61	62.63	77.36	75.83	72.18	67.97	65.03	62.03
w/ RMM (ours)	**68.86**	**67.61**	**66.21**	**79.52**	**78.47**	**76.54**	**69.21**	**67.45**	**63.93**

References
[1] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." CVPR 2017;
[2] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019;
[3] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." TPAMI 2017;
[5] Liu, Yaoyao, et al. "Mnemonics training: Multi-class incremental learning without forgetting." CVPR 2020;
[6] Douillard, Arthur, et al. "Podnet: Pooled outputs distillation for small-tasks incremental learning." ECCV 2020;
[7] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "Adaptive aggregation networks for class-incremental learning." CVPR 2021.

Ablation results: two-level RL performs better than one-level RL

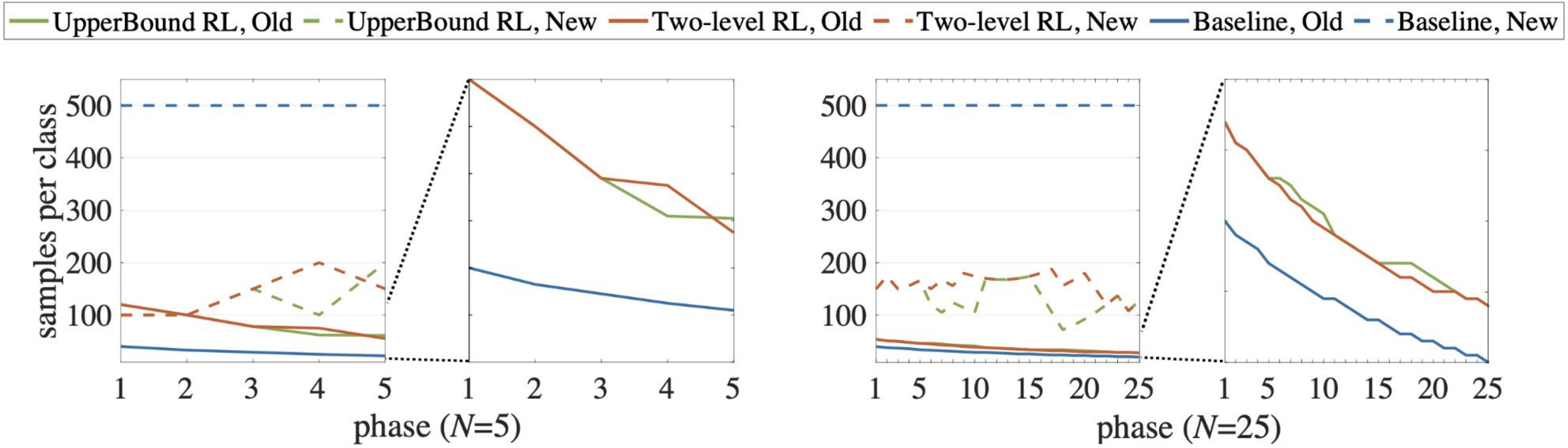
Ablation Setting	CIFAR-100						ImagNet-Subset					
	N=5		10		25		5		10		25	
	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
1 BaseRow	66.61	57.81	64.61	55.70	62.63	52.53	77.36	70.02	75.83	68.97	72.18	63.89
2 One-level RL	67.92	58.61	66.94	58.31	65.95	56.44	78.50	72.00	78.15	71.00	75.47	67.47
3 Two-level RL (Used)	68.86	59.00	67.61	59.03	66.21	56.50	79.52	73.80	78.47	71.40	76.54	68.84
margin	+2.3	+1.2	+3	+3.3	+3.6	+4	+2.1	+3.8	+2.6	+2.4	+4.4	+5
4 Two-level RL (T.P.)	68.62	59.40	67.22	58.20	65.82	56.20	78.81	72.42	77.68	70.77	75.29	68.81
margin	+2	+1.6	+2.6	+2.5	+3.2	+3.7	+1.5	+2.4	+1.9	+1.8	+3.1	+4.9
5 UpperBound RL	70.00	61.12	68.36	60.00	66.56	56.74	80.01	74.31	78.95	71.97	76.99	69.14
6 CrossVal Fixed	67.50	58.48	66.69	57.19	65.73	55.51	77.96	70.31	76.70	69.08	74.18	66.10

Ablation results: transferred policy achieves comparable performance

Ablation Setting	CIFAR-100						ImagNet-Subset					
	N=5		10		25		5		10		25	
	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
1 BaseRow	66.61	57.81	64.61	55.70	62.63	52.53	77.36	70.02	75.83	68.97	72.18	63.89
2 One-level RL	67.92	58.61	66.94	58.31	65.95	56.44	78.50	72.00	78.15	71.00	75.47	67.47
3 Two-level RL (Used)	68.86	59.00	67.61	59.03	66.21	56.50	79.52	73.80	78.47	71.40	76.54	68.84
margin	+2.3	+1.2	+3	+3.3	+3.6	+4	+2.1	+3.8	+2.6	+2.4	+4.4	+5
4 Two-level RL (T.P.)	68.62	59.40	67.22	58.20	65.82	56.20	78.81	72.42	77.68	70.77	75.29	68.81
margin	+2	+1.6	+2.6	+2.5	+3.2	+3.7	+1.5	+2.4	+1.9	+1.8	+3.1	+4.9
5 UpperBound RL	70.00	61.12	68.36	60.00	66.56	56.74	80.01	74.31	78.95	71.97	76.99	69.14
6 CrossVal Fixed	67.50	58.48	66.69	57.19	65.73	55.51	77.96	70.31	76.70	69.08	74.18	66.10

“**T.P.**” denotes our results using the **P**olicy functions **T**ransferred from another dataset.

Allocated memory: RMM achieves more balanced memory allocation



Thanks!


Adaptive Aggregation Networks for Class-Incremental Learning

RMM: Reinforced Memory Management for Class-Incremental Learning

Webpage: https://class-il.mpi-inf.mpg.de/rmm/

Code: https://gitlab.mpi-klsb.mpg.de/yaoyaoliu/rmm/