max planck institut
informatik

RMM: Reinforced Memory Management for Class-Incremental Learning

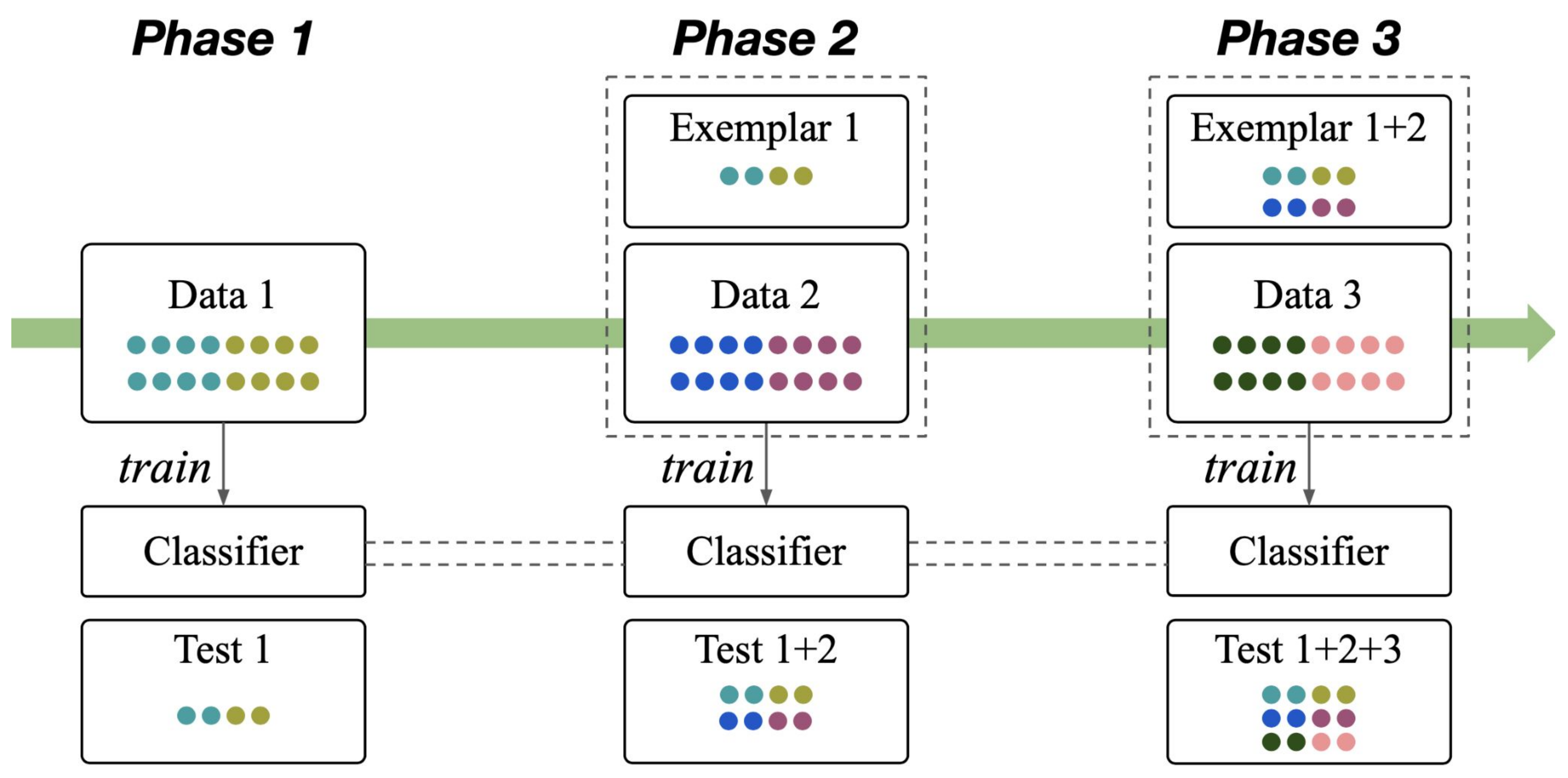
Yaoyao Liu[1] Bernt Schiele[1] Qianru Sun[2]

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

SMU SINGAPORE MANAGEMENT UNIVERSITY

Task & Challenge & Contributions

- **Task: Class-Incremental Learning[1]**
  - Different classes arrive in different phases;
  - At any time, it provides a classifier for the classes observed so far;
  - The memory is limited.

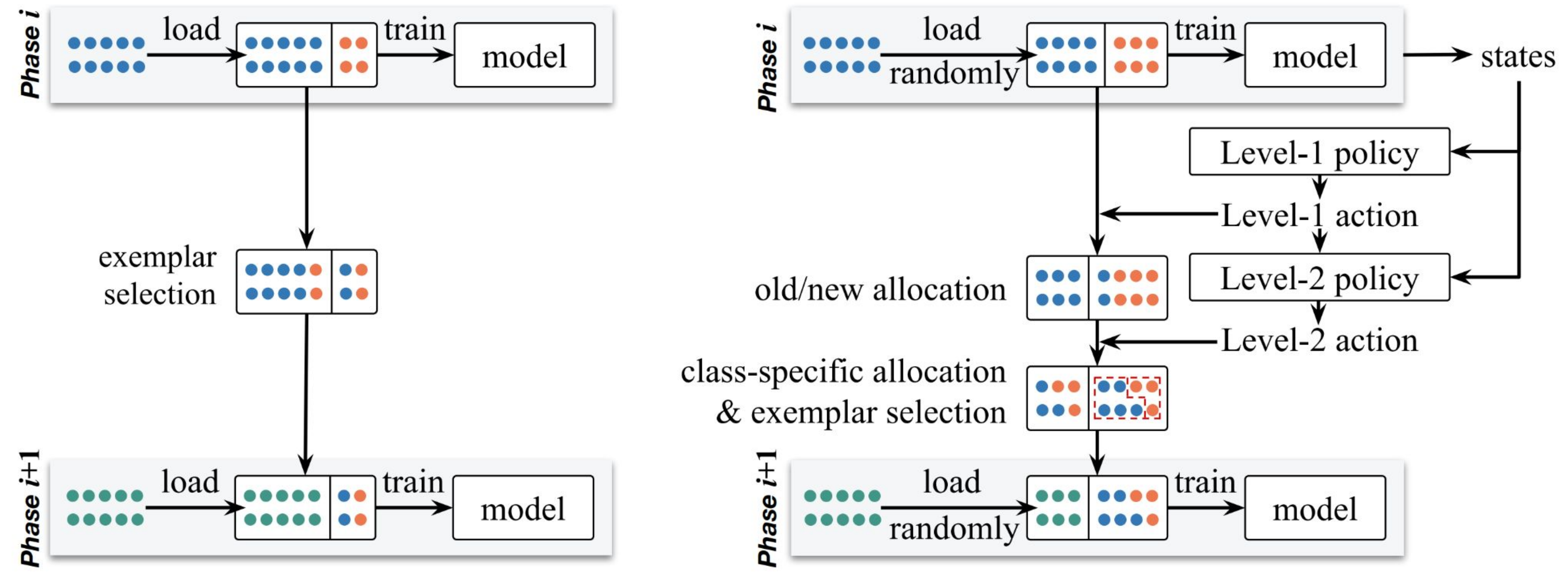


- **Challenge: Memory Allocation between Old and New classes**
  - High-plasticity models easily forget old classes;
  - High-stability models are weak to learn new classes.

- **Contributions**
  - ***A hierarchical reinforcement learning algorithm*** *called RMM to manage the memory in a way that can be conveniently modified through incremental phases and for different classes;*
  - ***A pseudo task generation strategy*** *that requires only in-domain available data (small-scale) or cross-domain datasets (large-scale), relieving the data incompatibility between reinforcement learning and class-incremental learning;*
  - ***Extensive experiments, visualization, and interpretation*** *for RMM in three CIL benchmarks and using two top models as baselines.*

Framework & Optimization Steps

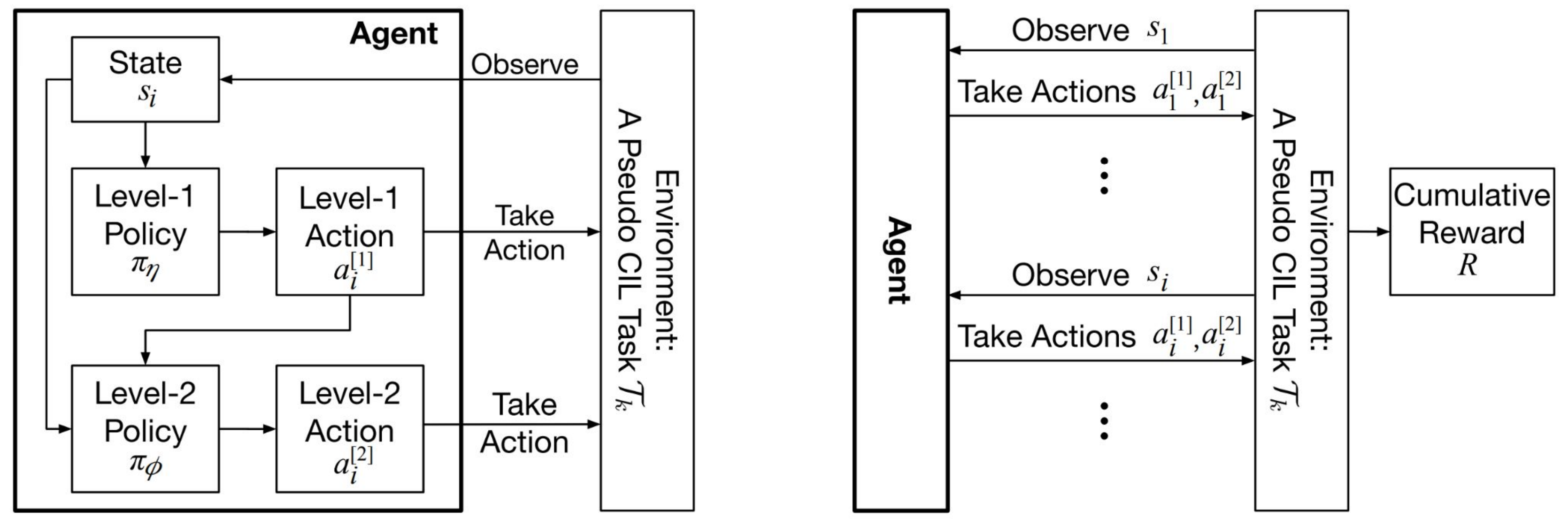
- **Conceptual Illustrations**



(a) Baseline

(b) Reinforced Memory Management (ours)

- **The RL system**



(a) The i-th incremental phase of the k-th pseudo CIL task

(b) The k-th pseudo CIL task

- **Optimization Steps**
  - The expected cumulative reward

$$J(\eta, \phi) = \mathbb{E}_{\mathcal{T}} \mathbb{E}_{\pi_\eta, \pi_\phi}[R].$$

  - Using the policy gradient theorem

$$\nabla_{\eta,\phi} J(\eta, \phi) = \mathbb{E}_{\mathcal{T}} \left[\sum_{i=1}^{N} \mathbb{E}_{\pi_\eta, \pi_\phi} [\nabla_{\eta,\phi} \log(\pi_\eta(a_i^{[1]}|s_i)\pi_\phi(a_i^{[2]}|s_i, a_i^{[1]}))R] \right]$$

  - Following the REINFORCE algorithm

$$\nabla_{\eta,\phi} J(\eta, \phi) = \frac{1}{ZK} \sum_{k=1}^{K} \sum_{z=1}^{Z} \sum_{i=1}^{N} \nabla_{\eta,\phi} \log(\pi_\eta(a_i^{[1]}|s_i)\pi_\phi(a_i^{[2]}|s_i, a_i^{[1]}))(R_z^k - b)$$

Experiment Results

- **Ablation Study**

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

- **Comparing w/ SOTA**

Method	*CIFAR-100*			*ImageNet-Subset*			*ImageNet-Full*		
	N=5	10	25	5	10	25	5	10	25
LUCIR [18]	63.34	62.47	59.69	71.21	68.21	64.15	65.16	62.34	57.37
Mnemonics [29]	64.59	62.59	61.02	72.60	71.66	70.52	65.40	64.02	62.05
PODNet [13]	64.60	63.13	61.96	76.45	74.66	70.15	66.80	64.89	60.28
LUCIR-AANets [28]	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 [28]	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**

- **The Memory Allocated for “Old” and “New”**