DESIGN AUTOMATION CONFERENCE

59

JULY 10 - 14, 2022

MOSCONE WEST CENTER

SAN FRANCISCO, CA, USA

CarM: Hierarchical Episodic Memory for Continual Learning

Soobee Lee, Minindu Weerakoon, Jonghyun Choi,
Minjia Zhang, Di Wang, Myeongjae Jeon

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

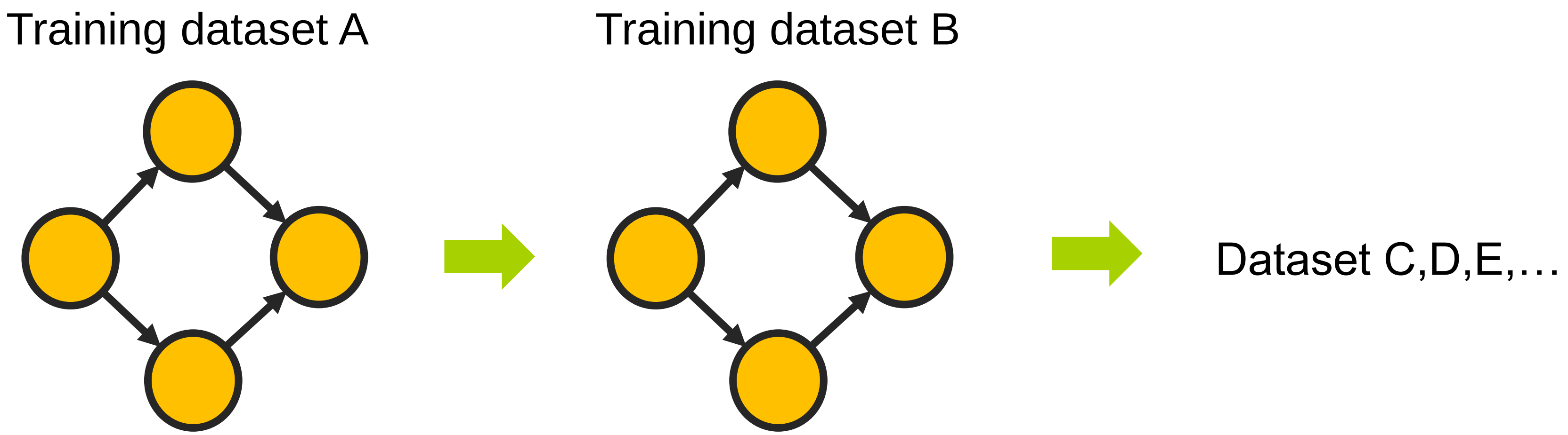
YONSEI UNIVERSITY

Microsoft

Background

Continual Learning(CL)

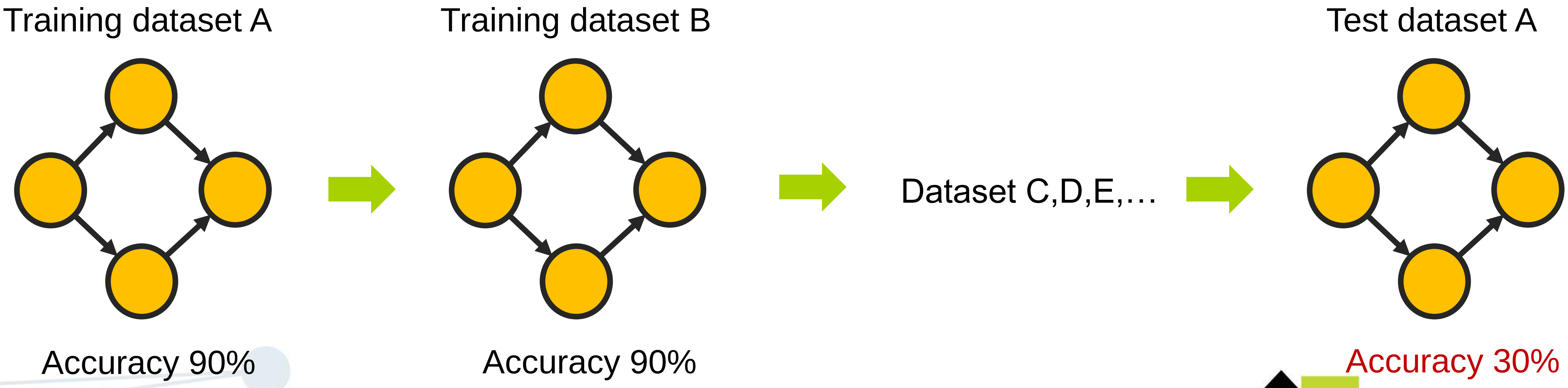
- Data in the real world is dynamically changing
- Deep neural networks are required to learn incrementally with dynamic incoming data



Challenge

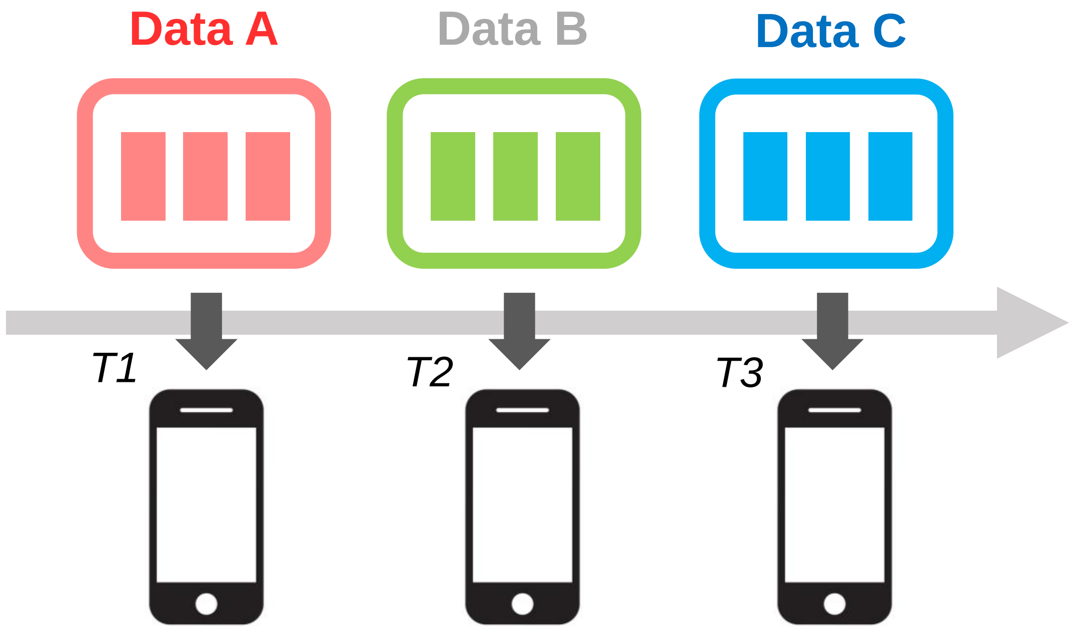
Catastrophic Forgetting

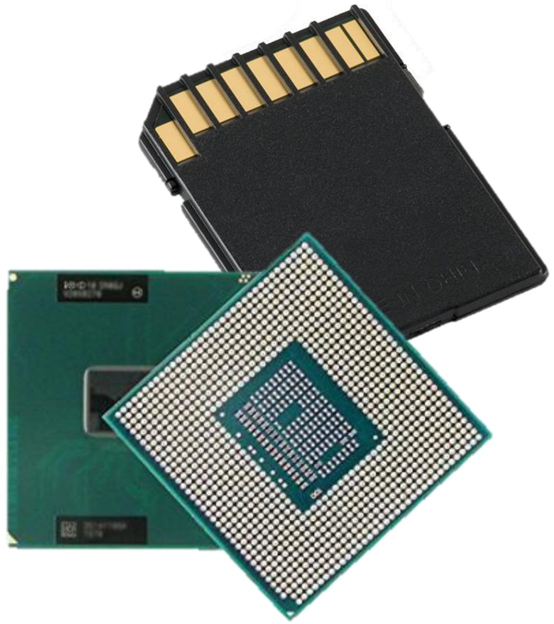
When trained on new data, standard neural networks forget most information related to old data previously learned



Challenge

Catastrophic forgetting is a main challenge especially in edge devices





Stream of non-i.i.d. data

Resource constraints

Motivation

To solve the forgetting issue, prior CL methods suggest...

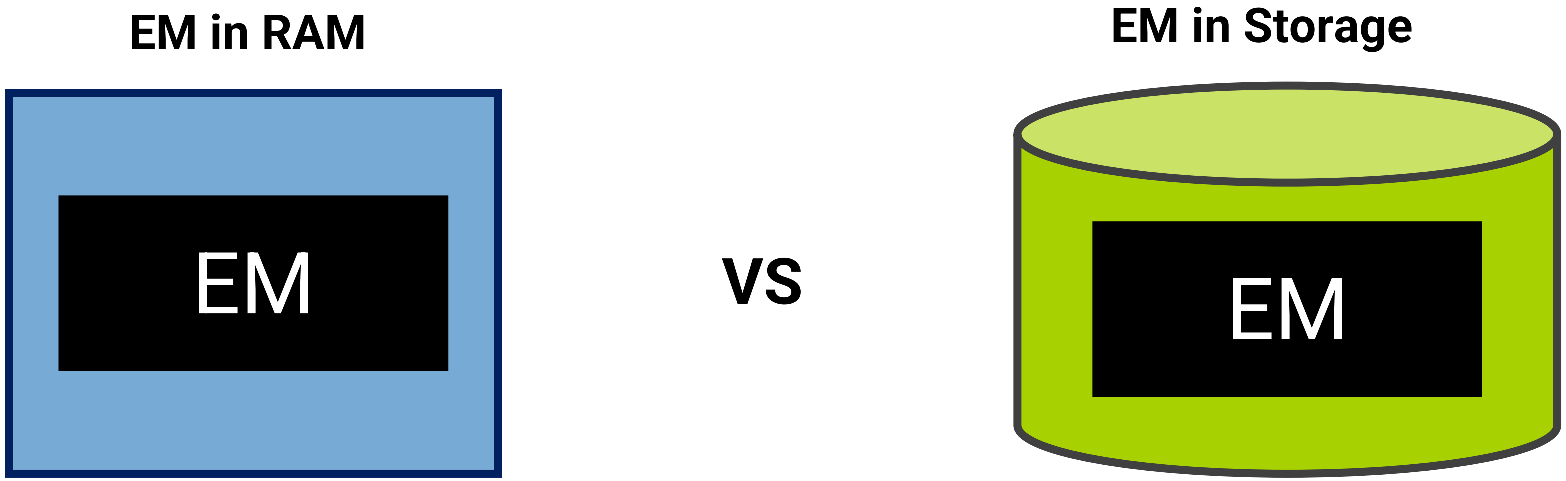
- Algorithmic approach only (*e.g.*, regularization, distillation, *etc.*)
- Algorithmic approach + **Episodic Memory(EM)**

Since EM ensures learning the previous representation constantly,

many works develop their own algorithms on top of EM to improve the accuracy

Limitation of existing works

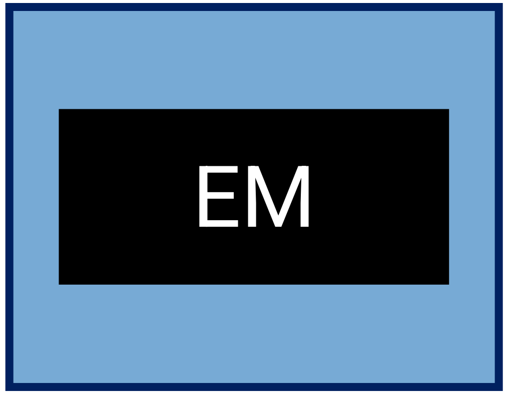
However, prior works are simulation-based, which implemented single-level EM



Limitation of existing works

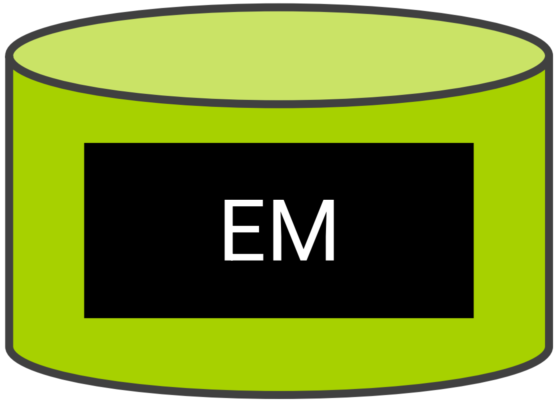
Each design has its own advantages, but…

EM in RAM



VS

EM in Storage



(+) Fast access to old data

(–) Limited memory capacity

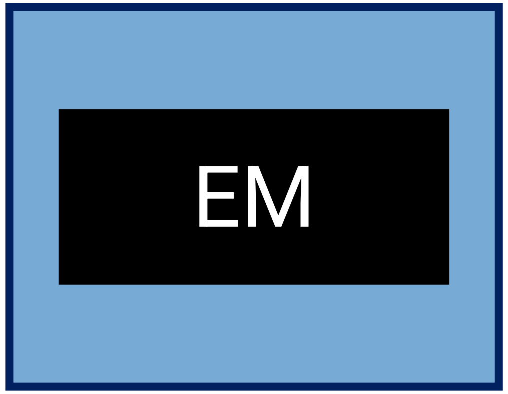
(+) Abundant old data

(–) Slow access to old data

DESIGN AUTOMATION CONFERENCE 59

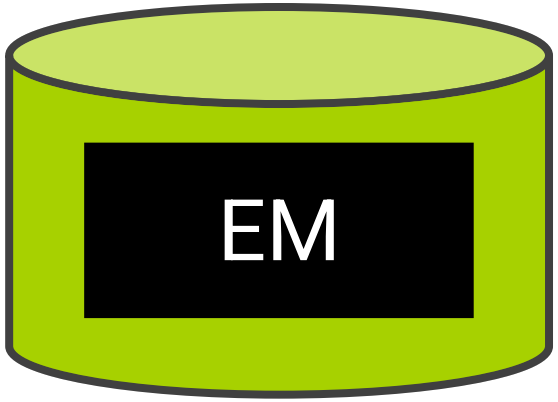
Limitation of existing works

EM in RAM



VS

EM in Storage



(+) Fast access to old data

(−) Limited memory capacity

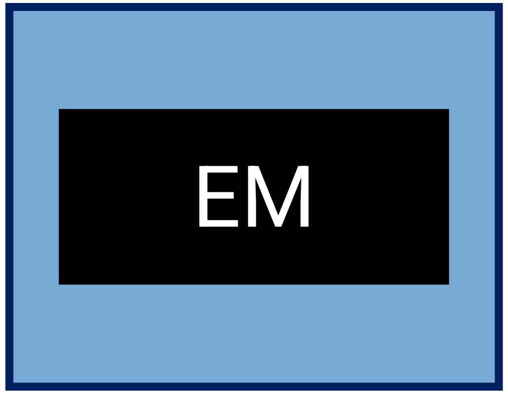
(+) Abundant old data

(−) Slow access to old data

They cannot meet the performance requirements for real-world applications!

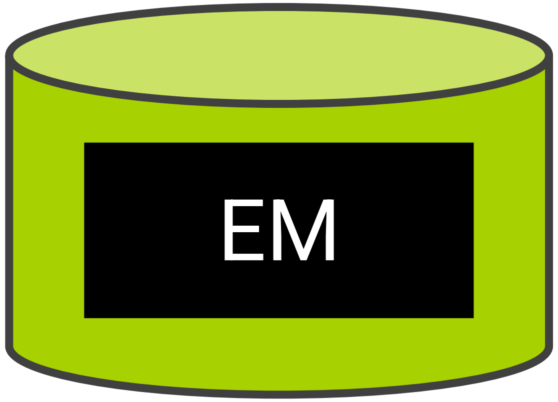
Limitation of existing works

EM in RAM



VS

EM in Storage



(+) Fast access to old data

(–) Limited memory capacity

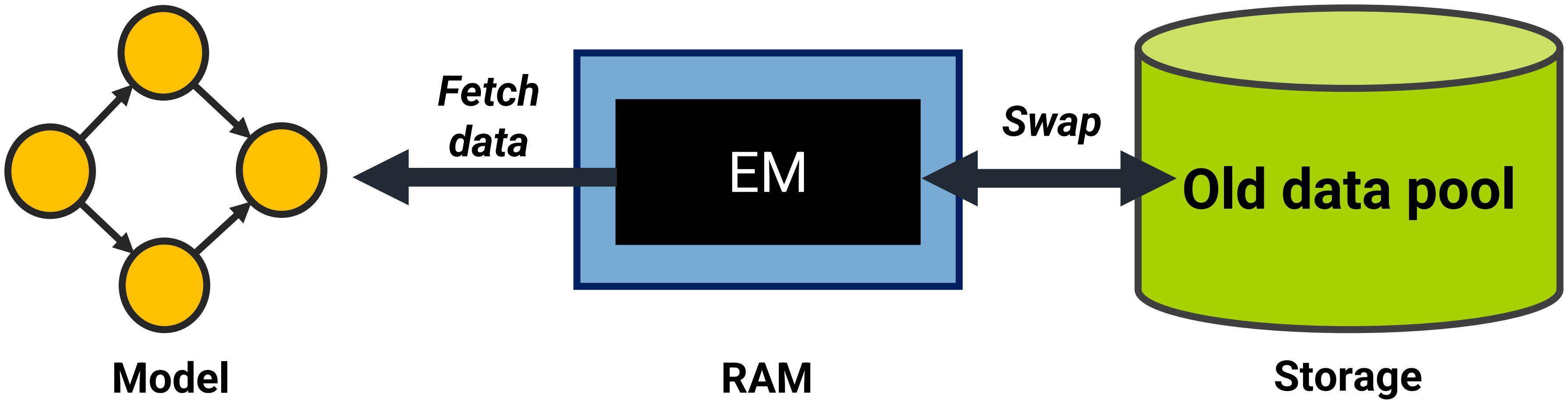
(+) Abundant old data

(–) Slow access to old data

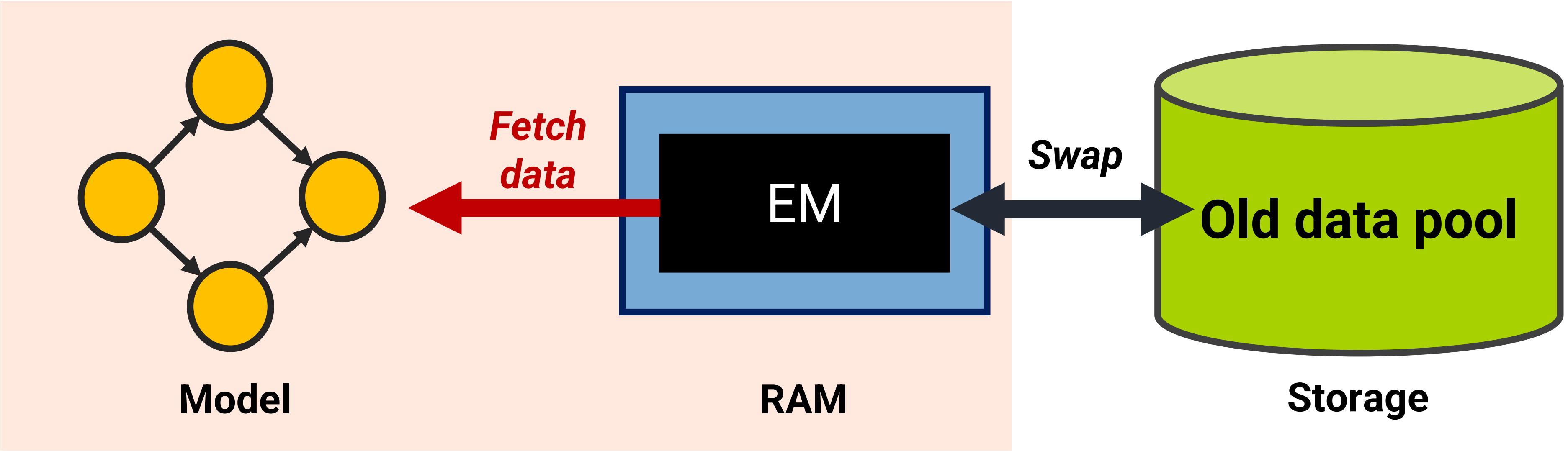
How can we make the best of both worlds?

Carousel Memory (CarM)

We propose Carousel Memory that combines the best of both worlds by hierarchical memory-aware data swapping for continual learning EM

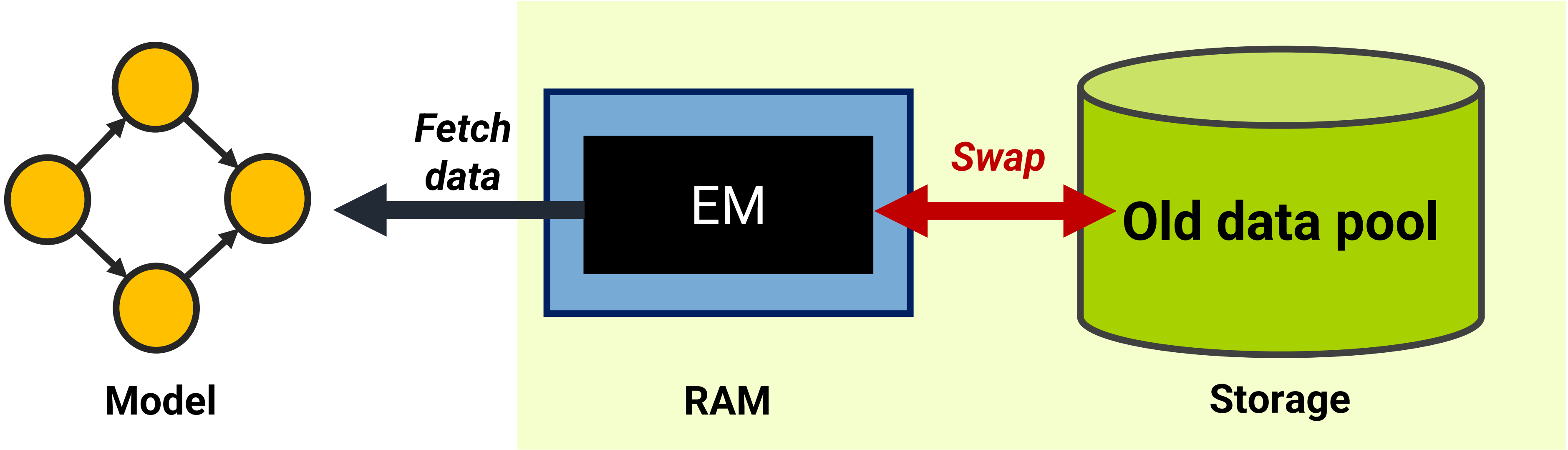


Carousel Memory (CarM)



CarM accesses **data only in memory** during training,
promising **high-speed training**

Carousel Memory (CarM)



CarM simultaneously leverages **abundant old data in storage by data swapping**, addressing the **forgetting issue**

Design principle of CarM

Goal 1 : System efficiency

Drawing in-storage data does not incur significant I/O overhead that affects the overall system efficiency

Goal 2 : Model accuracy

CarM improves the model accuracy by exploiting in-storage data more effectively for training

Design principle of CarM

Goal 1 : System efficiency

Drawing in-storage data does not incur significant I/O overhead that affects the overall system efficiency

↓

Swapping mechanism by asynchronous sample retrieval

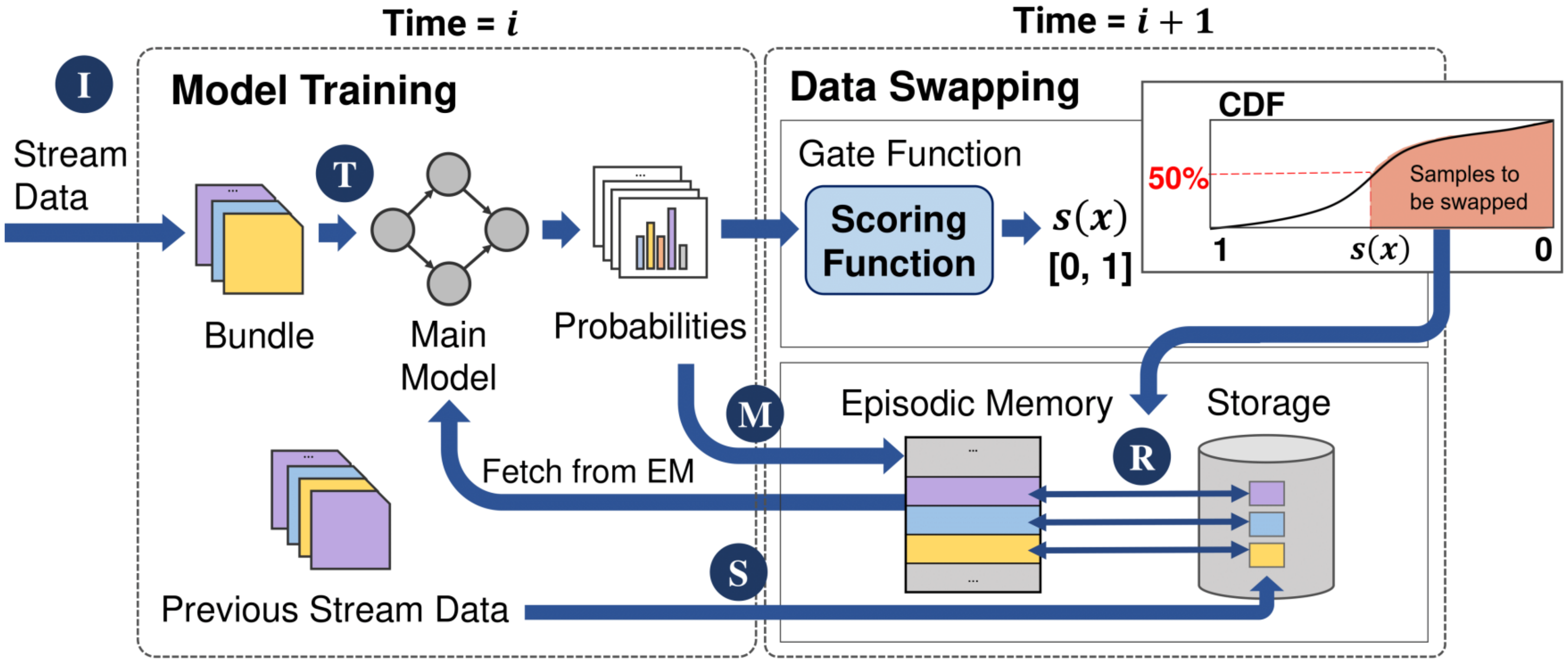
Goal 2 : Model accuracy

CarM improves the model accuracy by exploiting in-storage data more effectively for training

↓

Swapping policy by gate function

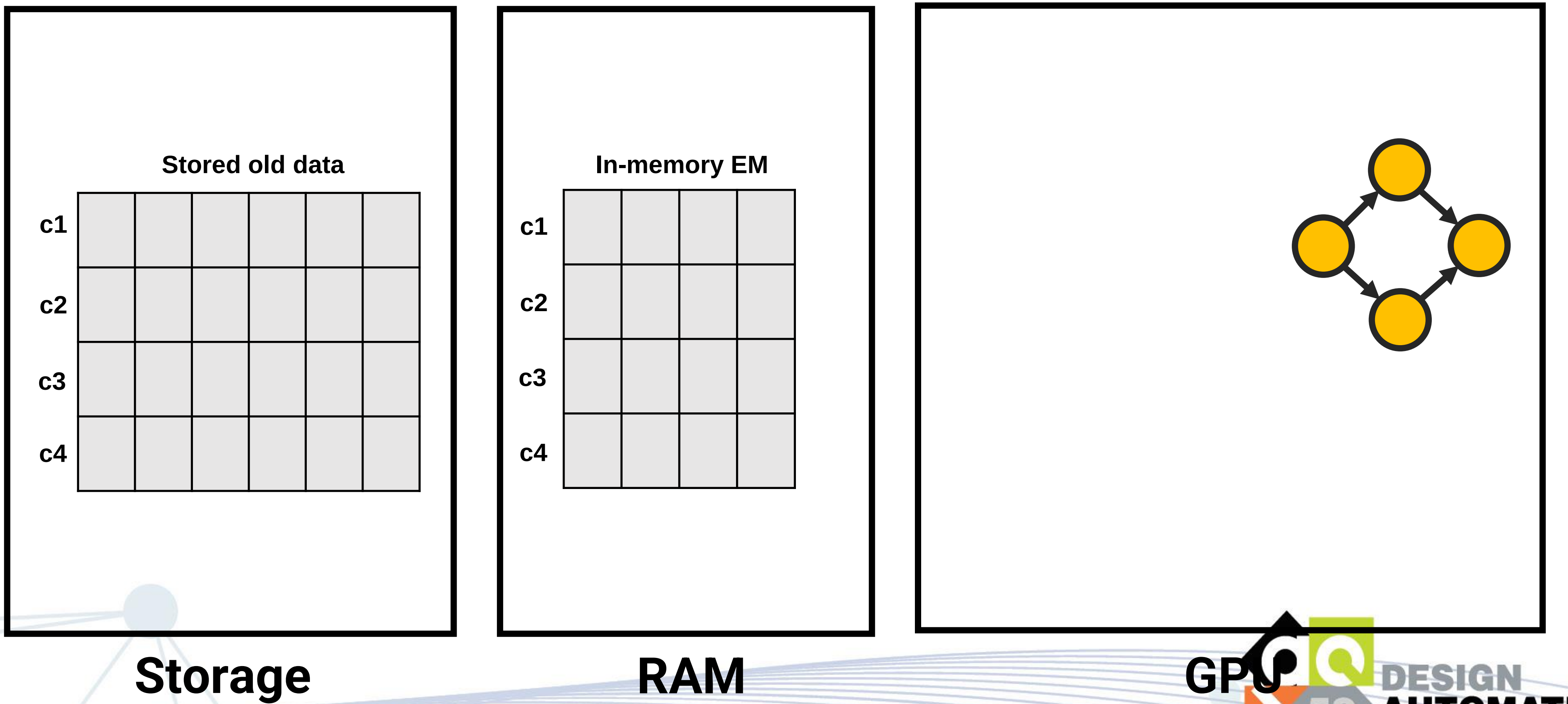
Architecture of CarM



- **I** Data incoming
- **T** Training
- **R** Storage sample retrieving
- **M** EM updating
- **S** Storage updating

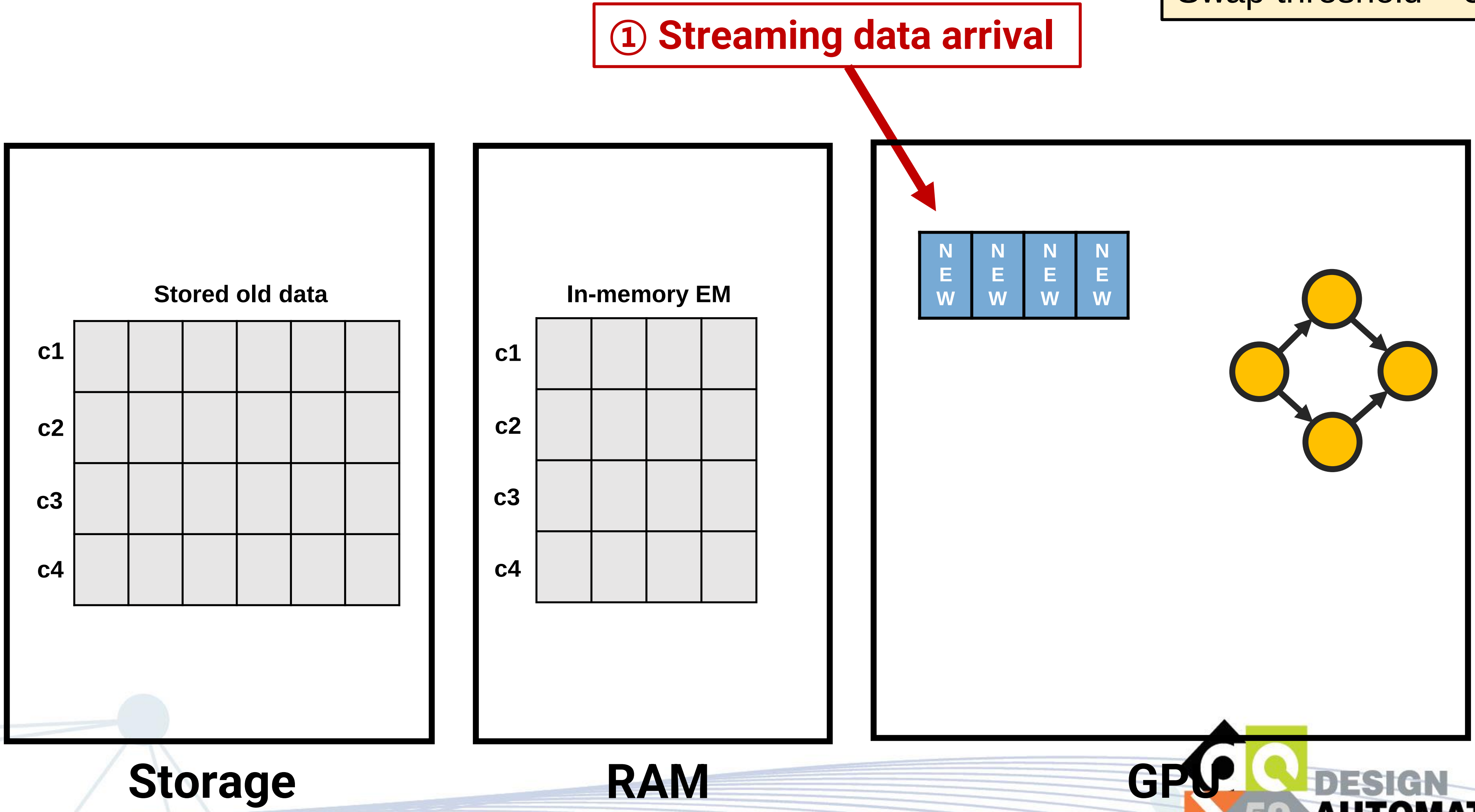
Workflow of CarM

Config
Batch size = 4
In-memory EM size = 16
Swap threshold = 50%



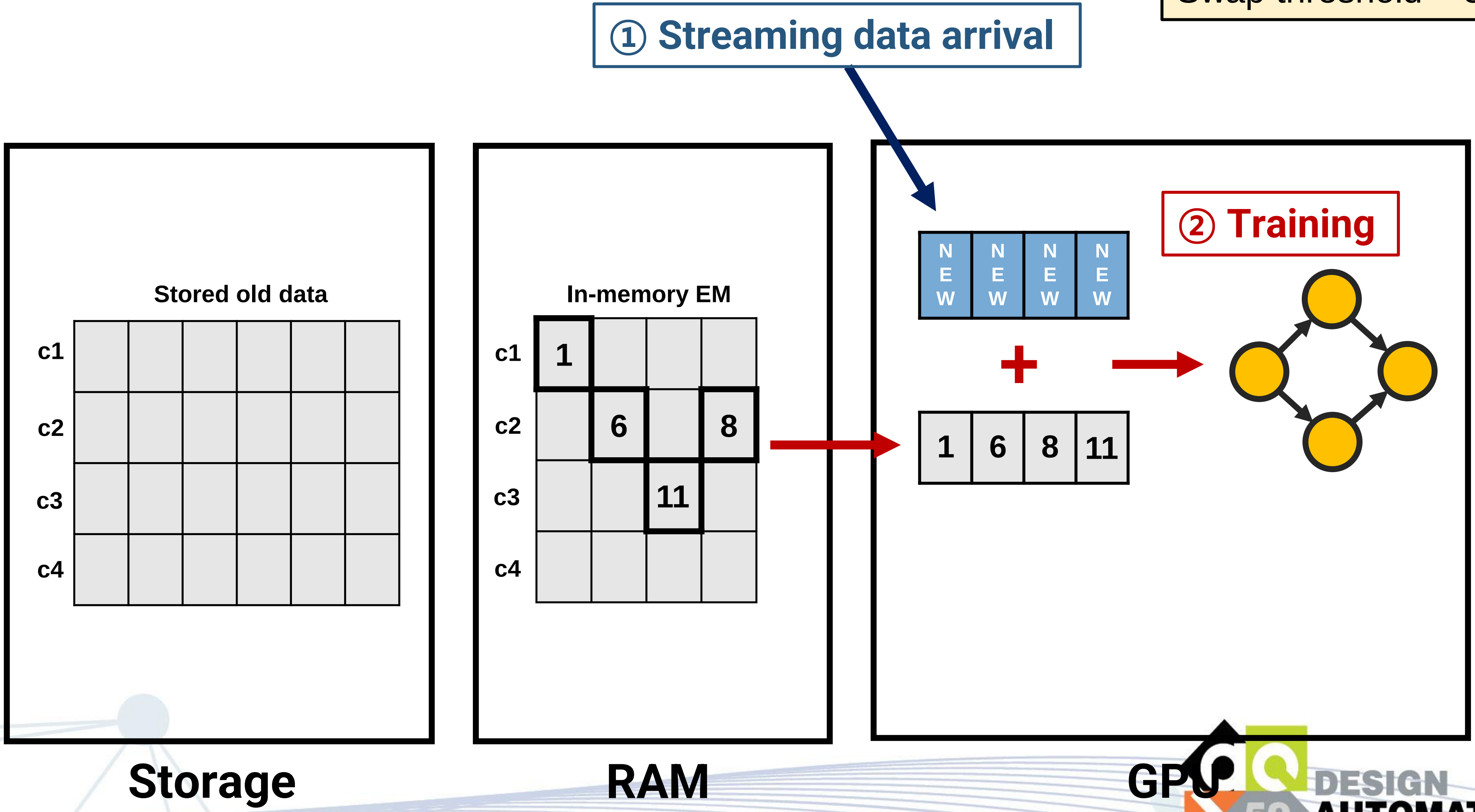
Workflow of CarM

Config
Batch size = 4
In-memory EM size = 16
Swap threshold = 50%



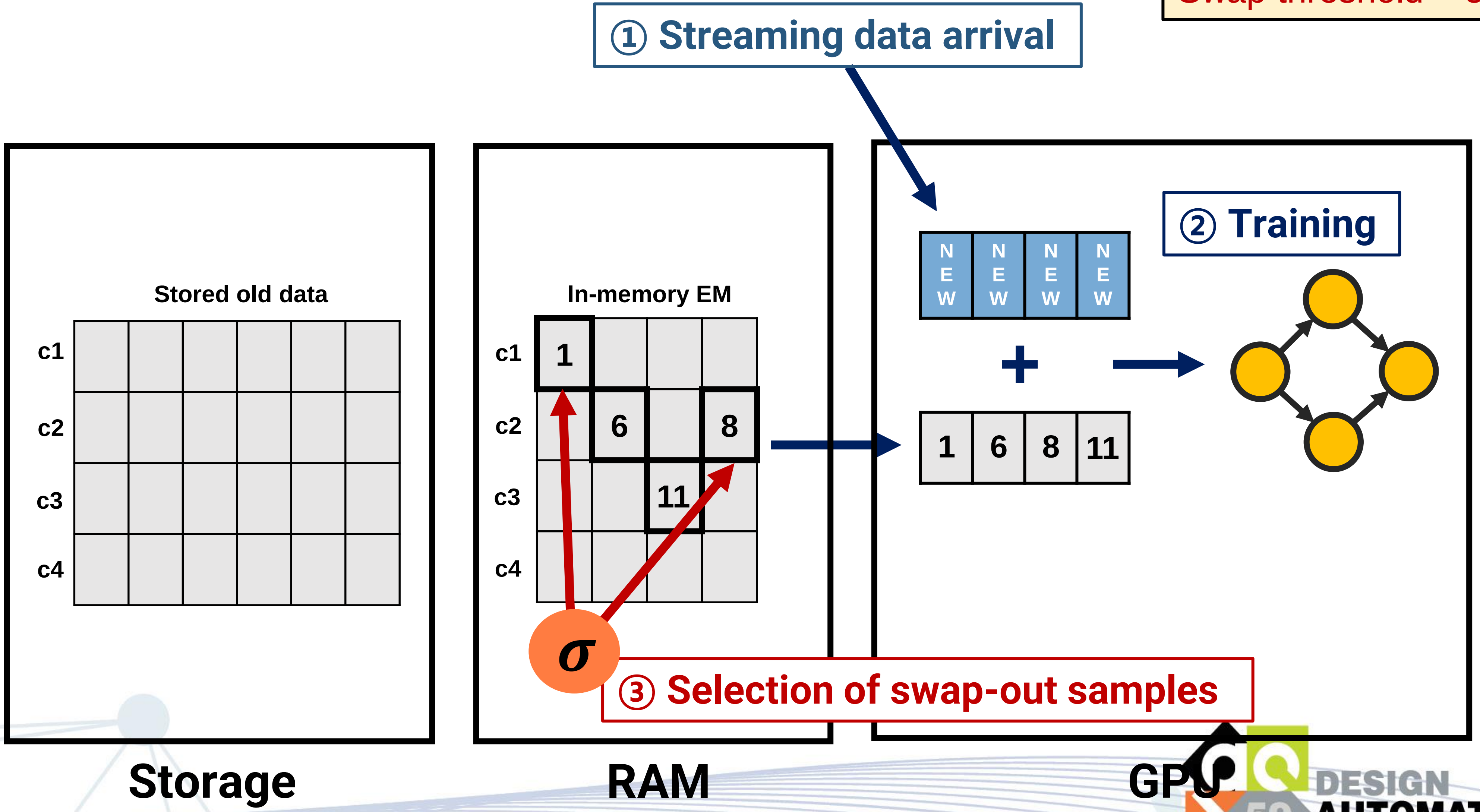
Workflow of CarM

Config
Batch size = 4
In-memory EM size = 16
Swap threshold = 50%



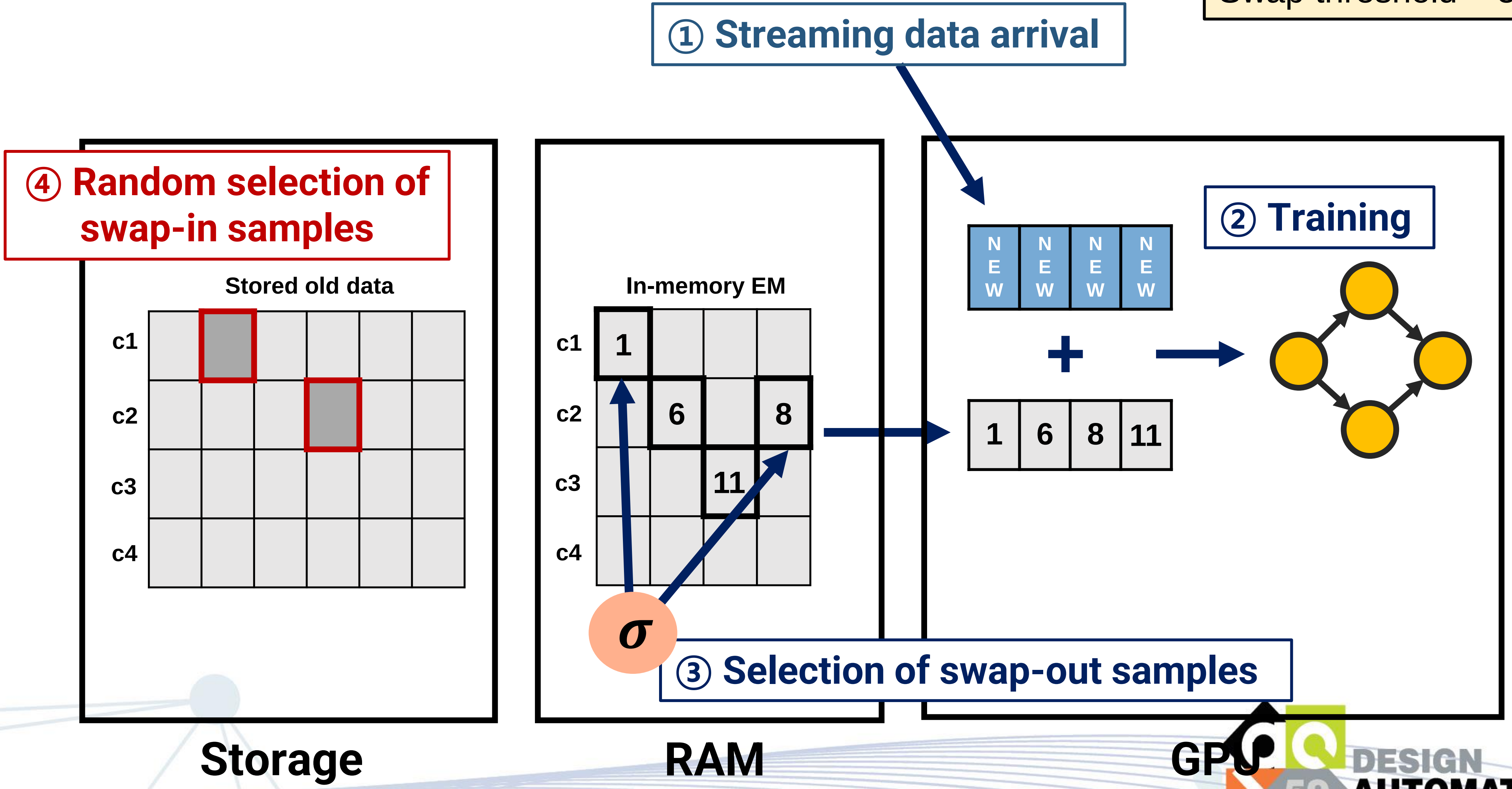
Workflow of CarM

Config
Batch size = 4
In-memory EM size = 16
Swap threshold = 50%



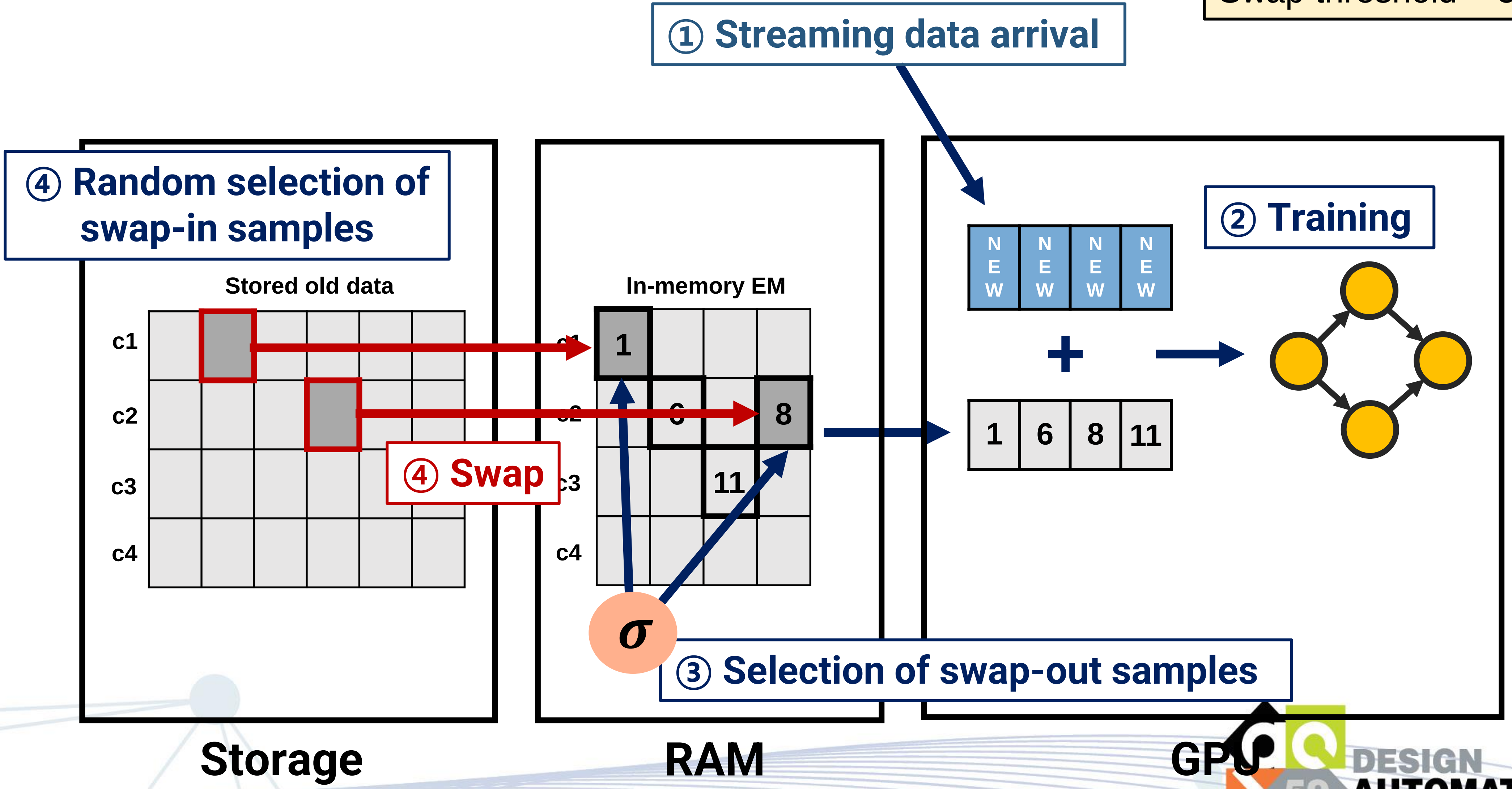
Workflow of CarM

Config
Batch size = 4
In-memory EM size = 16
Swap threshold = 50%



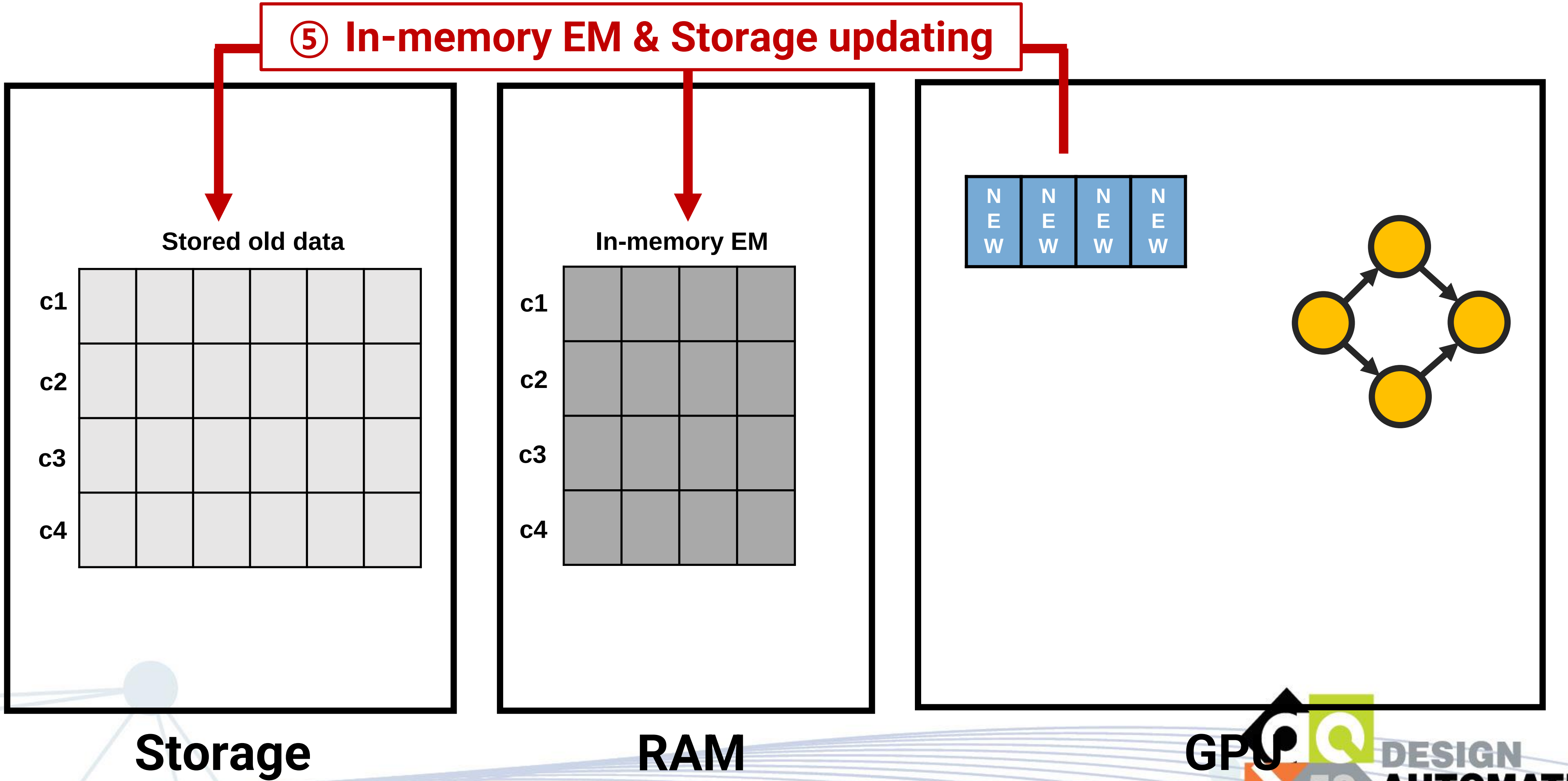
Workflow of CarM

Config
Batch size = 4
In-memory EM size = 16
Swap threshold = 50%



Workflow of CarM

Config
Batch size = 4
In-memory EM size = 16
Swap threshold = 50%

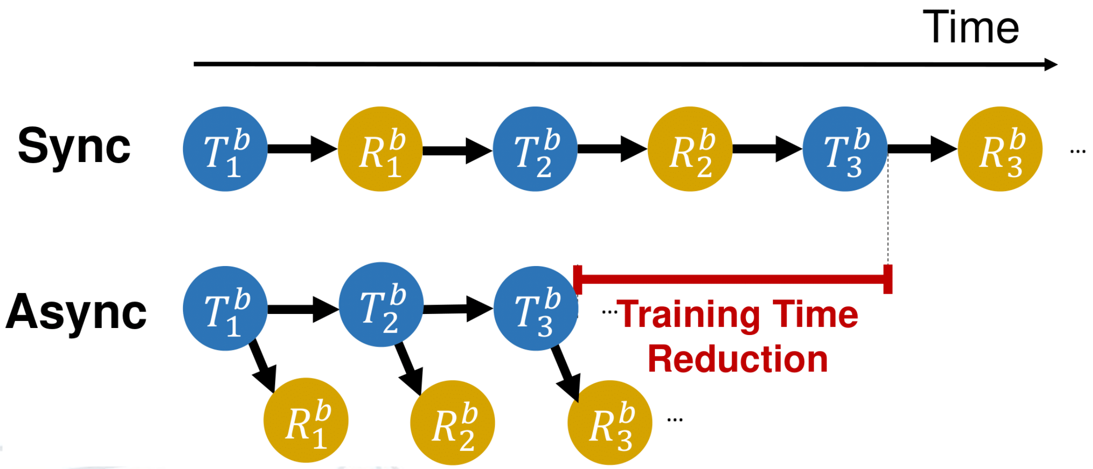


Data swapping method of CarM

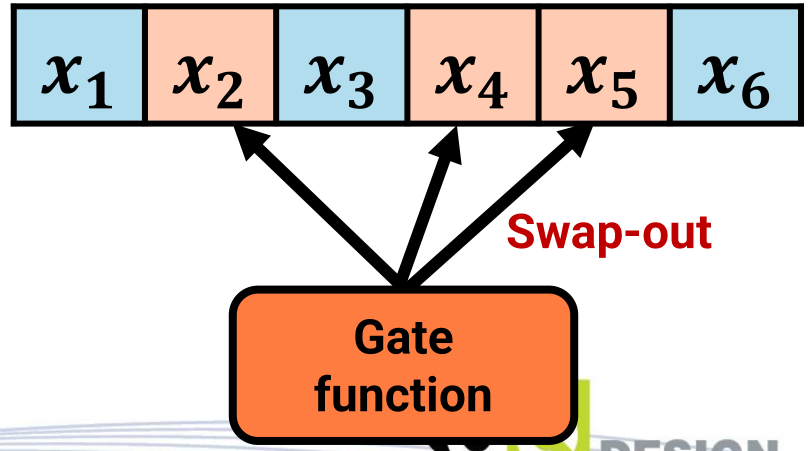
CarM includes the swap stages

- Samples in EM are replaced by samples preserved in storage
- It must be system-efficient, while improving the accuracy

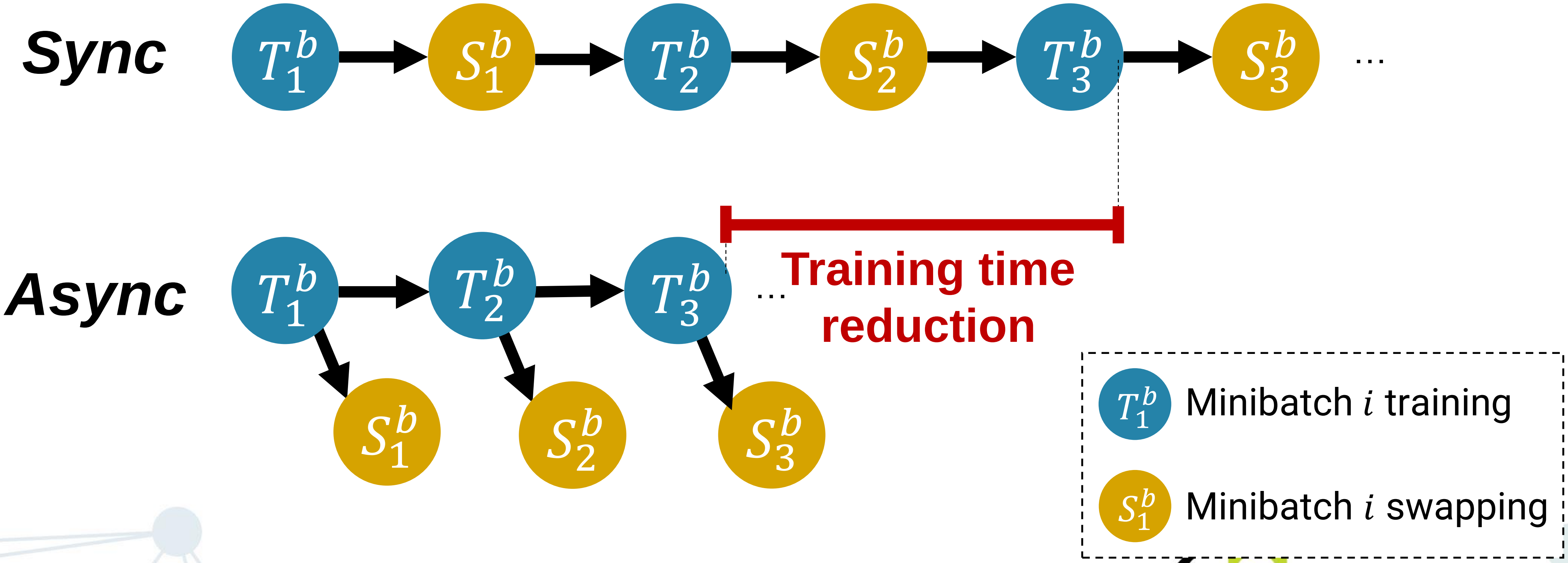
1. Swapping mechanism



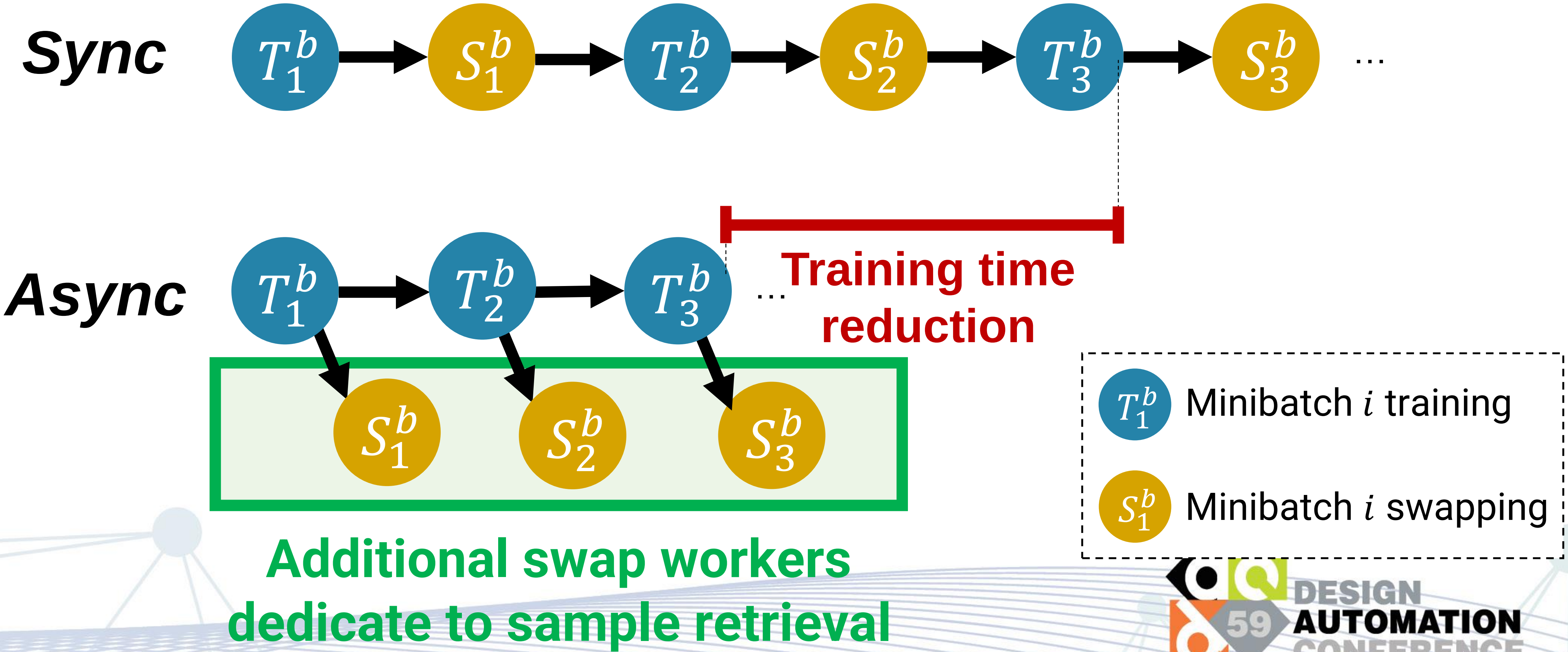
2. Swapping policy



Swapping mechanism



Swapping mechanism



Swapping policy

- To adjust I/O traffic and keep important samples in EM, CarM selects a subset of a mini-batch to swap out from EM
- CarM's gate function decides swap-out samples based on the score values

Gate function $\sigma_i = \mathbb{1}(s(x_i) > \tau)$

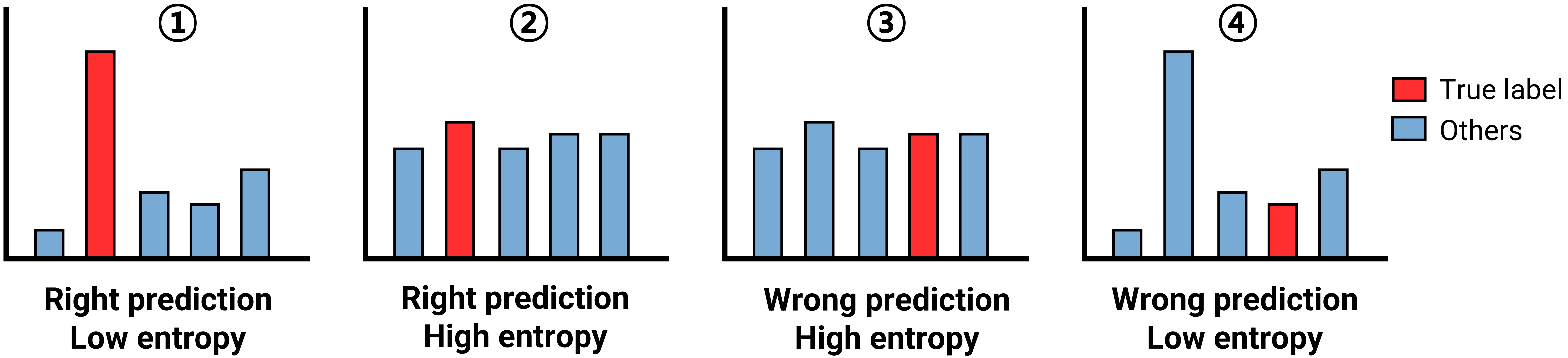
Entropy value [0,1]

Score function $s(x_i) = \frac{1}{2u}\left[g(x_i)H(f(x_i)) + (1 - g(x_i))\,(2u - H(f(x_i)))\right]$

Prediction flip (0 or 1)

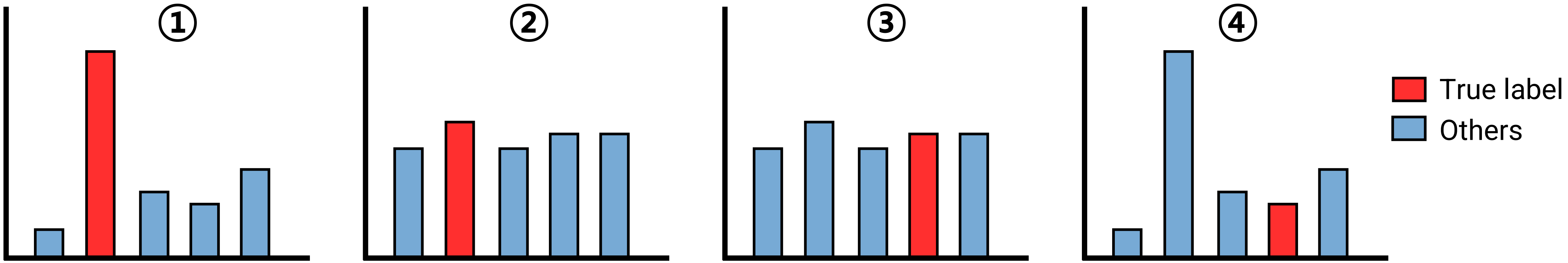
Gate function

Assume that we have 4 swap candidates with the following predictions…



Gate function

Assume that we have 4 swap candidates with the following predictions…



Right prediction **Low entropy**	**Right prediction** **High entropy**	**Wrong prediction** **High entropy**	**Wrong prediction** **Low entropy**

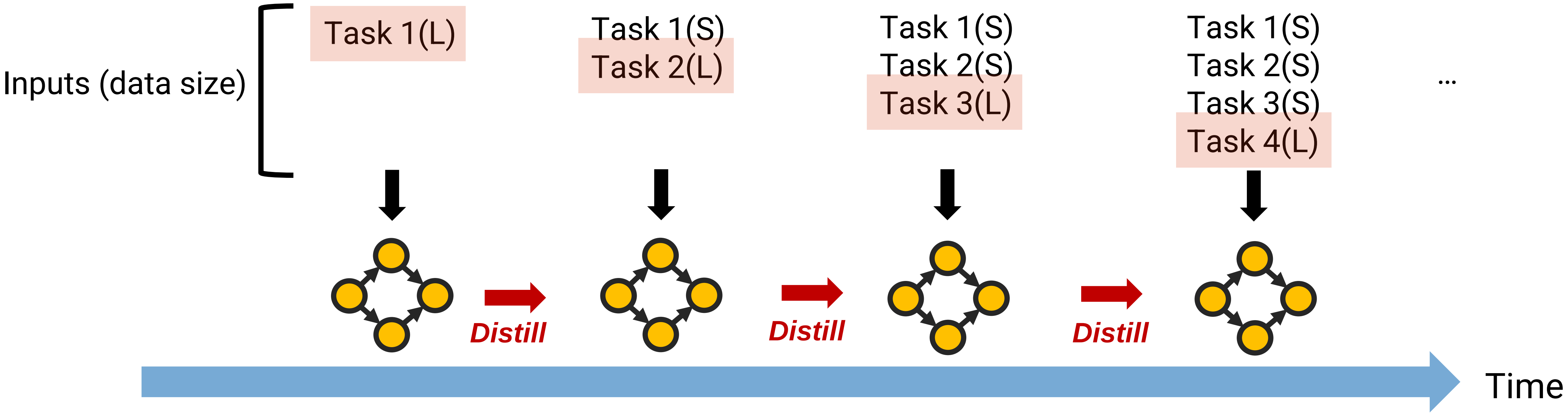
Low $s(x)$ → Easy sample

High $s(x)$ → Hard sample

Swap-out from EM

Keep in EM

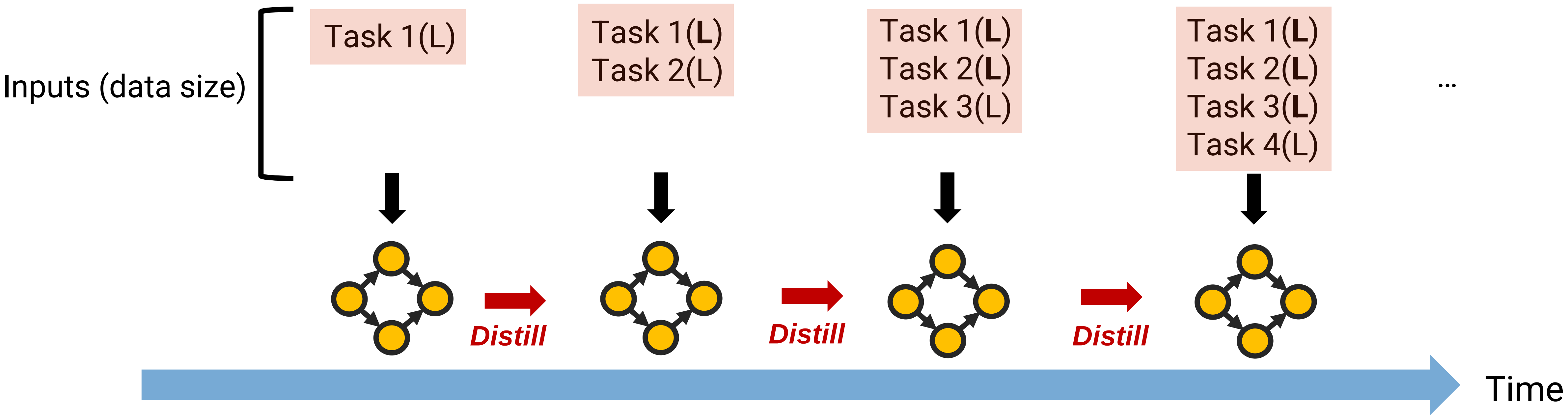
Analysis – Knowledge distillation on CarM



Why use knowledge distillation(KD) in CL?

KD transfers the knowledge of the old model trained with sufficient old data to the current model trained with a limited amount of old data

Analysis – Knowledge distillation on CarM



Why isn't knowledge distillation(KD) effective on CarM?

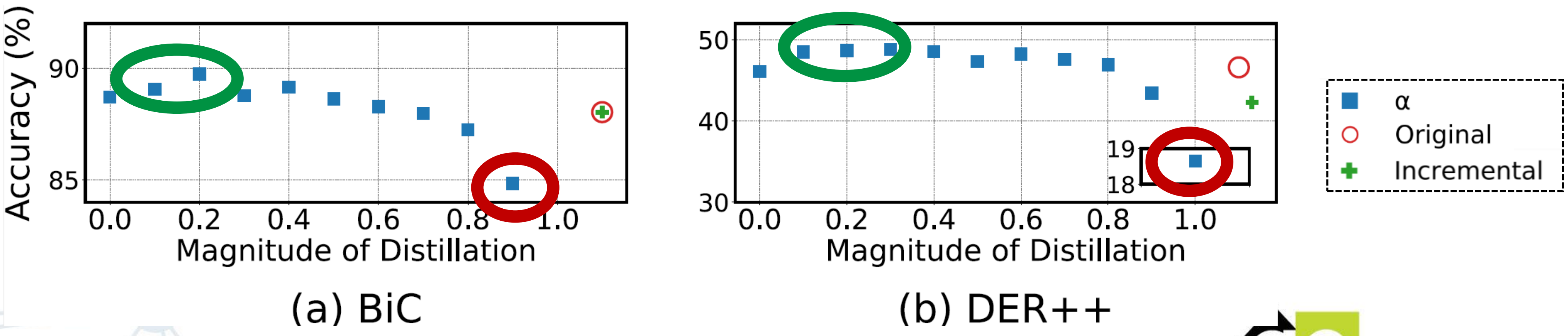
- Drawback of KD is that old models might not be sufficiently generalized for old tasks
- If distillation is extensive, data swapping which enables training with abundant old data can be interfered with the knowledge from the old models

Analysis – Knowledge distillation on CarM

Recent KD methods further combine soft labels and hard labels to update the model for old data

$$L = \alpha \times distillation\ loss + (1 - \alpha) \times hard\ label\ loss$$

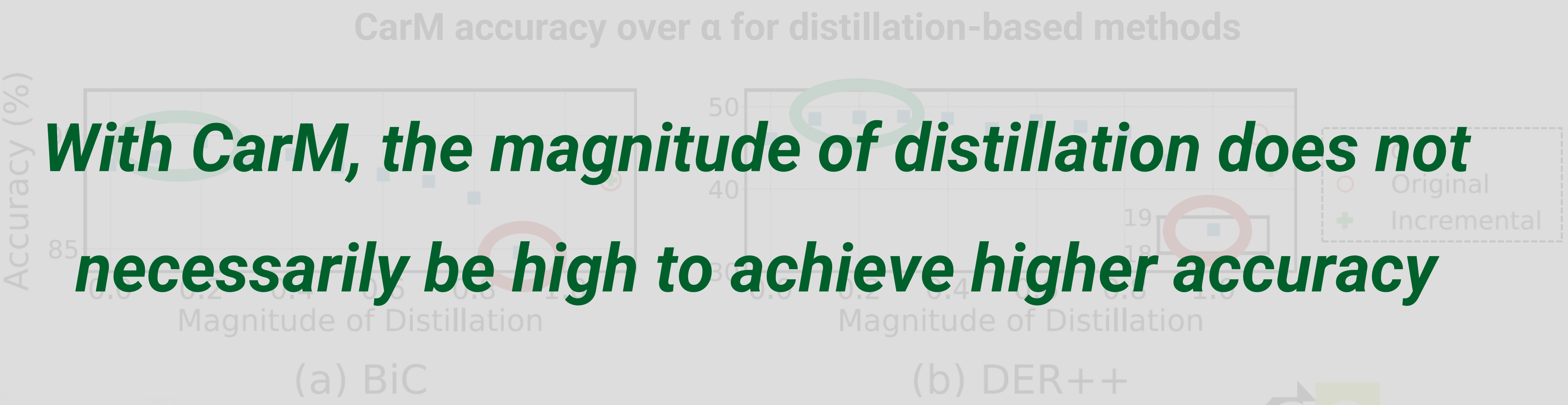
CarM accuracy over α for distillation-based methods



Analysis – Knowledge distillation on CarM

Recent KD methods further combine soft labels and hard labels to update the model for old data

$$L = \alpha \times distillation\ loss + (1 - \alpha) \times hard\ label\ loss$$



With CarM, the magnitude of distillation does not necessarily be high to achieve higher accuracy

Evaluation

Setup

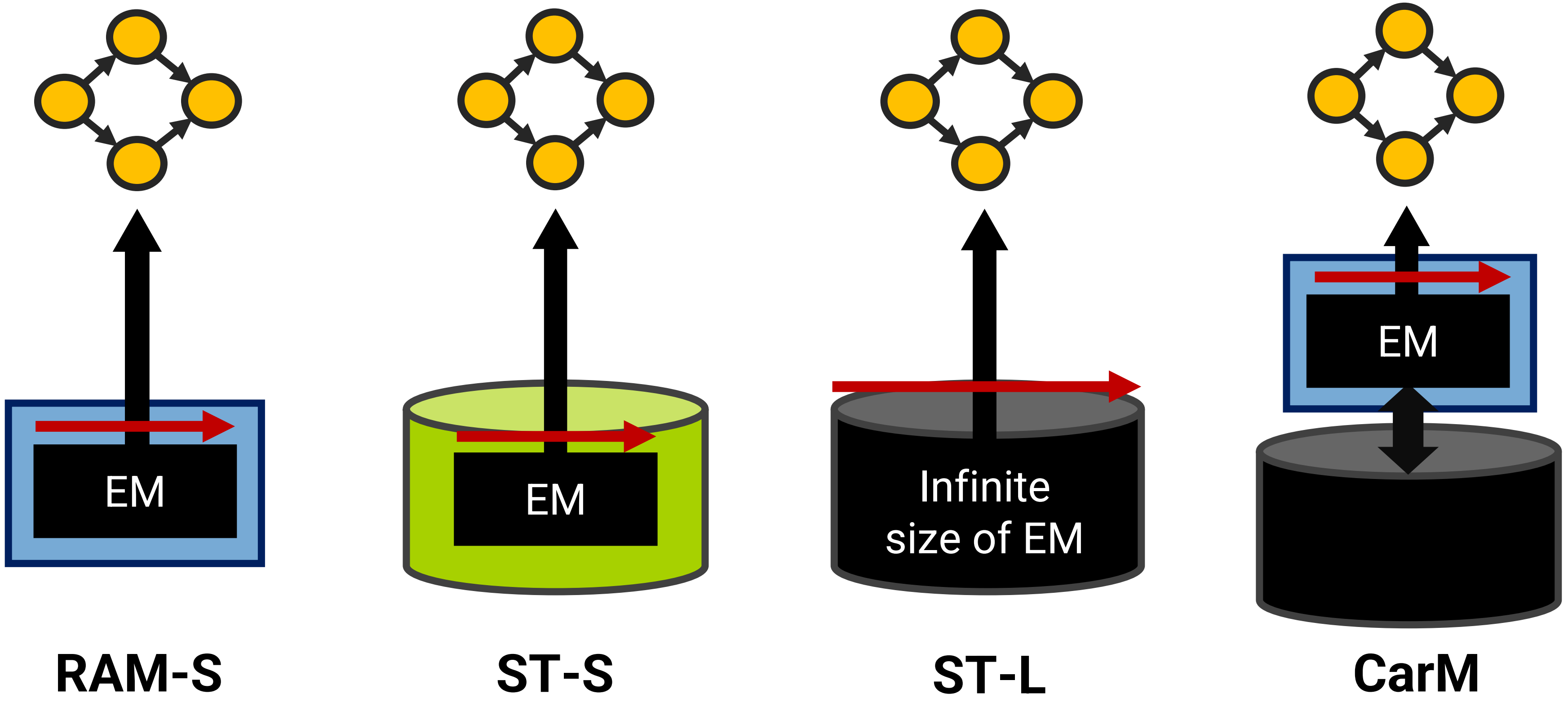
- NVIDIA 2080 Ti GPU
- Intel Xeon Gold 6226 12 cores
- 128GB RAM, 480GB SSD drive

Baselines & Datasets

We measure the performance with and without CarM in the existing methods of their own setups as used in the original works

	BiC	DER++	RM
CIFAR Subset	CIFAR100	CIFAR10	CIFAR10
	2000(6MB)	500(2MB)	500(2MB)
ImageNet Subset	ImageNet100	Tiny-ImageNet	Tiny-ImageNet
	2000(1GB)	4500(53MB)	500(6MB)

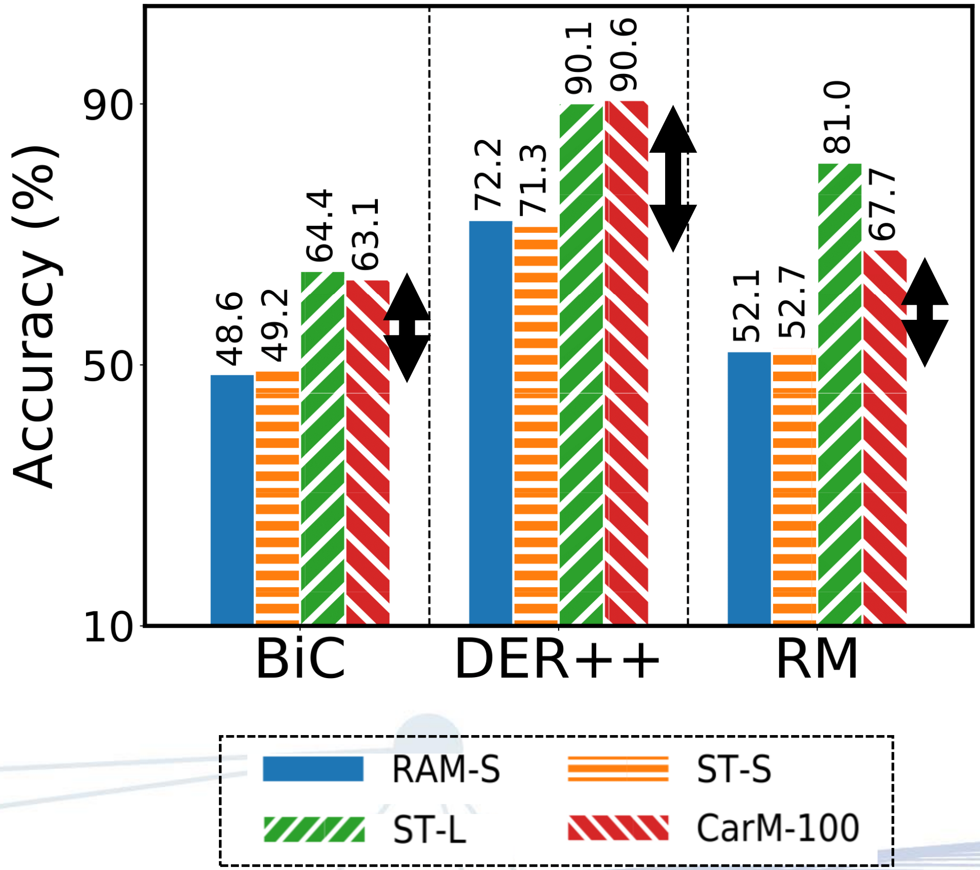
Result 1 – Comparison to other EM designs



We compare CarM with <u>non-hierarchical EM setups</u>

Result 1 – Comparison to other EM designs

Accuracy



Overall, ST-L shows the best accuracy due to its infinite capacity of EM

CarM improves the accuracy over RAM-S and ST-S

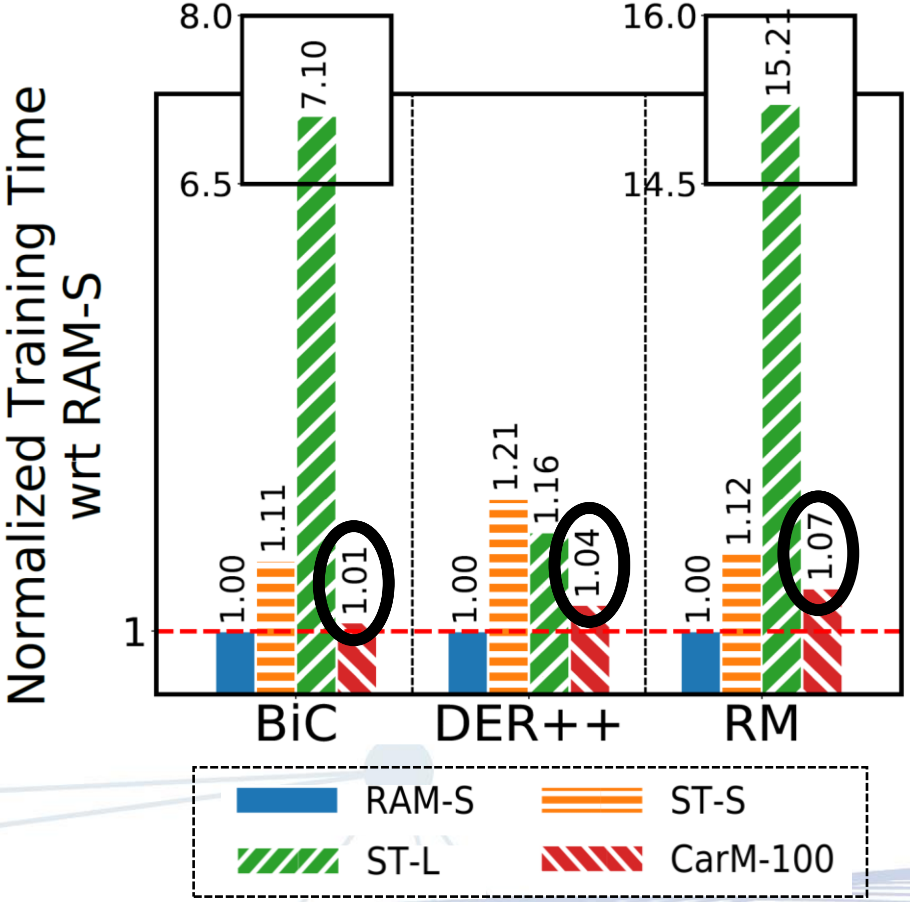
- 18.4% and 19.3% accuracy gain in DER++

Avoid catastrophic forgetting!

Accuracy : RAM-S ~ ST-S <<< CarM < ST-L

Result 1 – Comparison to other EM designs

Training time wrt RAM-S



CarM has negligible slowdowns compared to RAM-S

However, ST-L shows the slowdown up to 15.21X

Slowdown : RAM-S ~ CarM < ST-S <<< ST-L

Favorable for device learning!

DESIGN AUTOMATION CONFERENCE

Result 2 – Effect of asynchronous swapping

The percentage increase of training time for CarM-50 w.r.t. RAM-S (CIFAR/ImageNet)

Method	BiC	DER++	RM
Async	-0.3%/+1.7%	+0.3%/+2.4%	+2.3%/-0.9%
Sync	+20.0%/+33.8%	+71.6%/+38.8%	+25.9%/+2.6%

Asynchronous swapping has the marginal overhead,

whereas synchronous swapping slows down training time up to 71.6%,

Additional experiments

Varying EM sizes, data swapping ratios, and storage capacities

- Small EM size with CarM is better than larger EM size with original setup RAM-S
- 20% swapping is comparable to 100% swapping
- Storing only half of the whole dataset is still effective

CarM with large-scale dataset (ImageNet-1000)

- CarM is effective with large-scale dataset (up to 26%)

CarM on NVIDIA Jetson TX2

- CarM is effective on small device with quad-cores and 8GB RAM

Conclusion

- CarM suggests an effective **hierarchical EM** design providing the best of both worlds of memory and storage by data swapping
- CarM is largely **complementary to existing CL methods,** and we demonstrate its high performance, **mitigating forgetting without significant slowdown**
- Based on **careful analysis of distillation on CarM**, we found a way to assort with existing algorithmic optimizations
- CarM provides more **practical EM management in on-device** continual learning