Caching in Deep Learning Systems

Redwan Ibne Seraj Khan
PhD Candidate
Computer Science and Applications
Virginia Tech

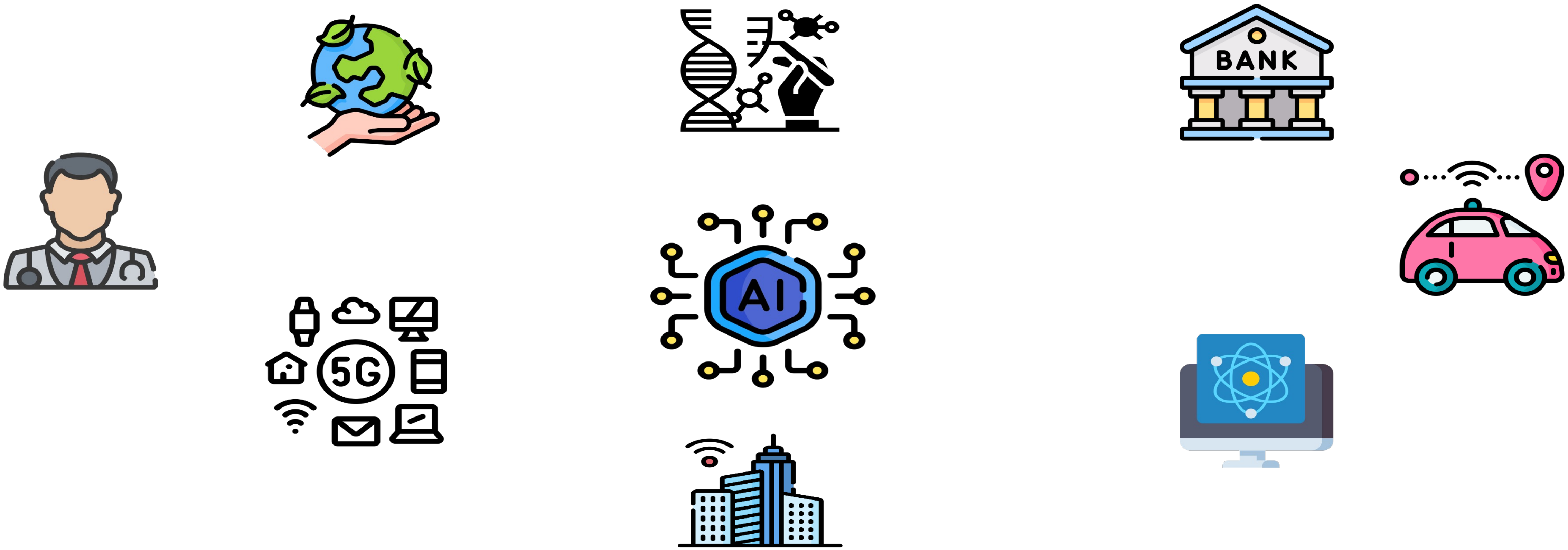
March 18, 2024

Today's Agenda

We need **flexible data storage systems** for modern workloads that **can adapt** itself to the **ever-changing workload requirements** by considering the underlying **workload and system characteristics**.

"Characteristic Awareness"

Modern workloads are ML-based or using ML

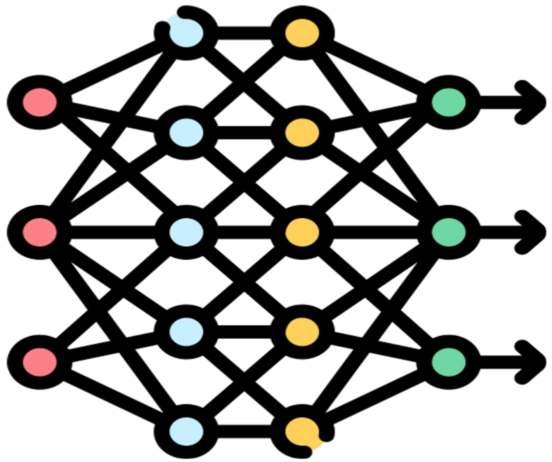


Applications are using Computer Vision, NLP, Reinforcement Learning, etc.
Market expected to reach 12 billion dollars by 2025!

Exponential growth ~19% annually

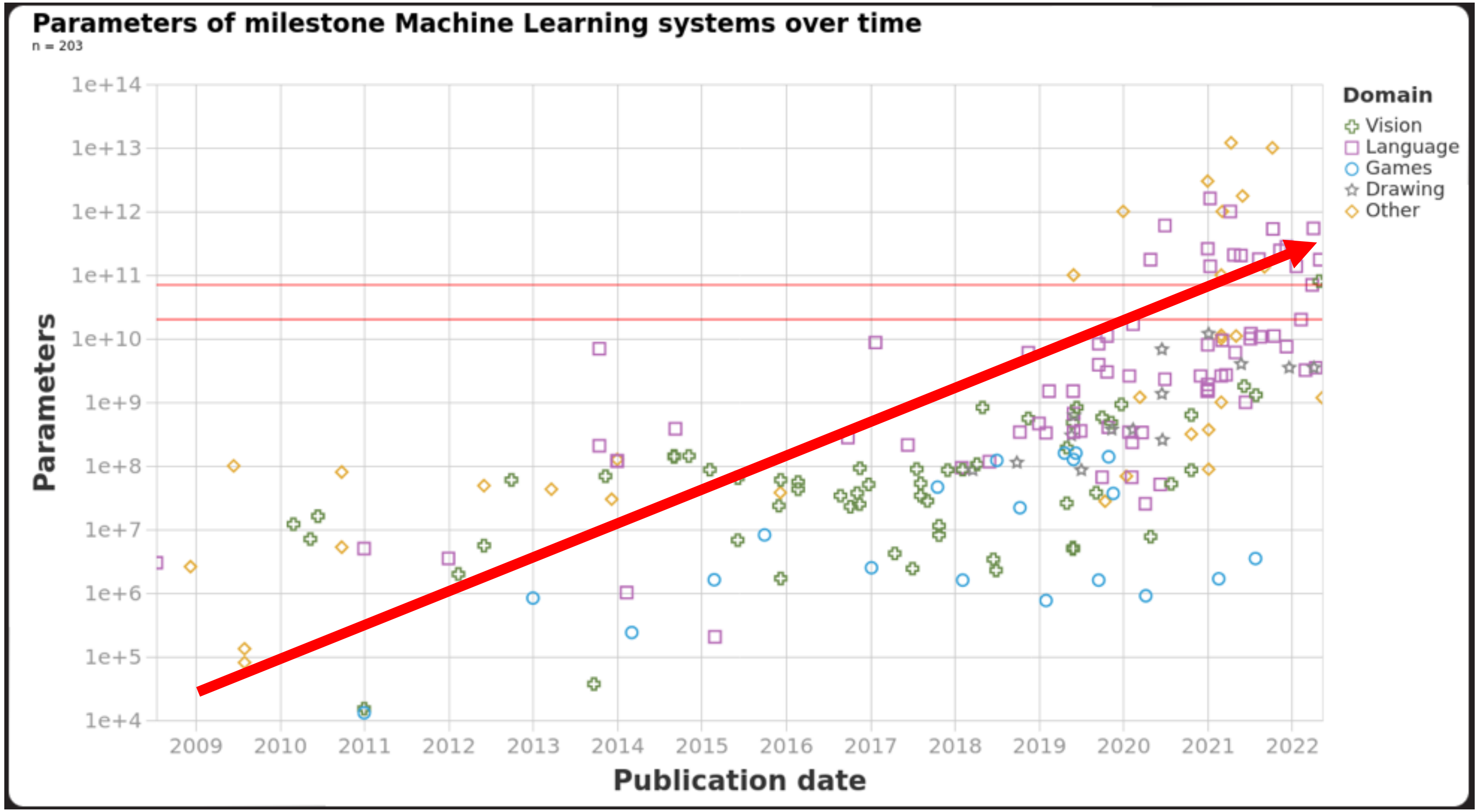
Global Conversational AI Market Report 2021. https://tinyurl.com/Global-AI-Market-Report-2021

Problem 1: ML workloads are Compute Intensive



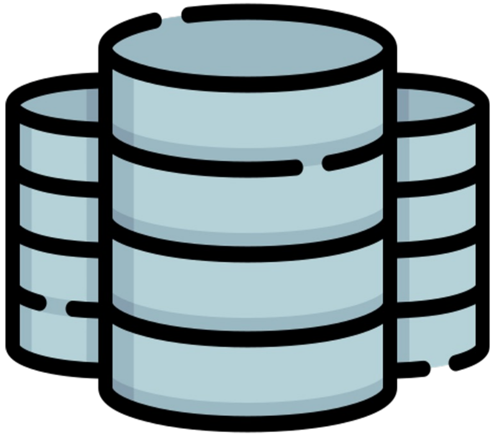
Model Size ↑

Computations ↑



https://epochai.org/blog/machine-learning-model-sizes-and-the-parameter-gap

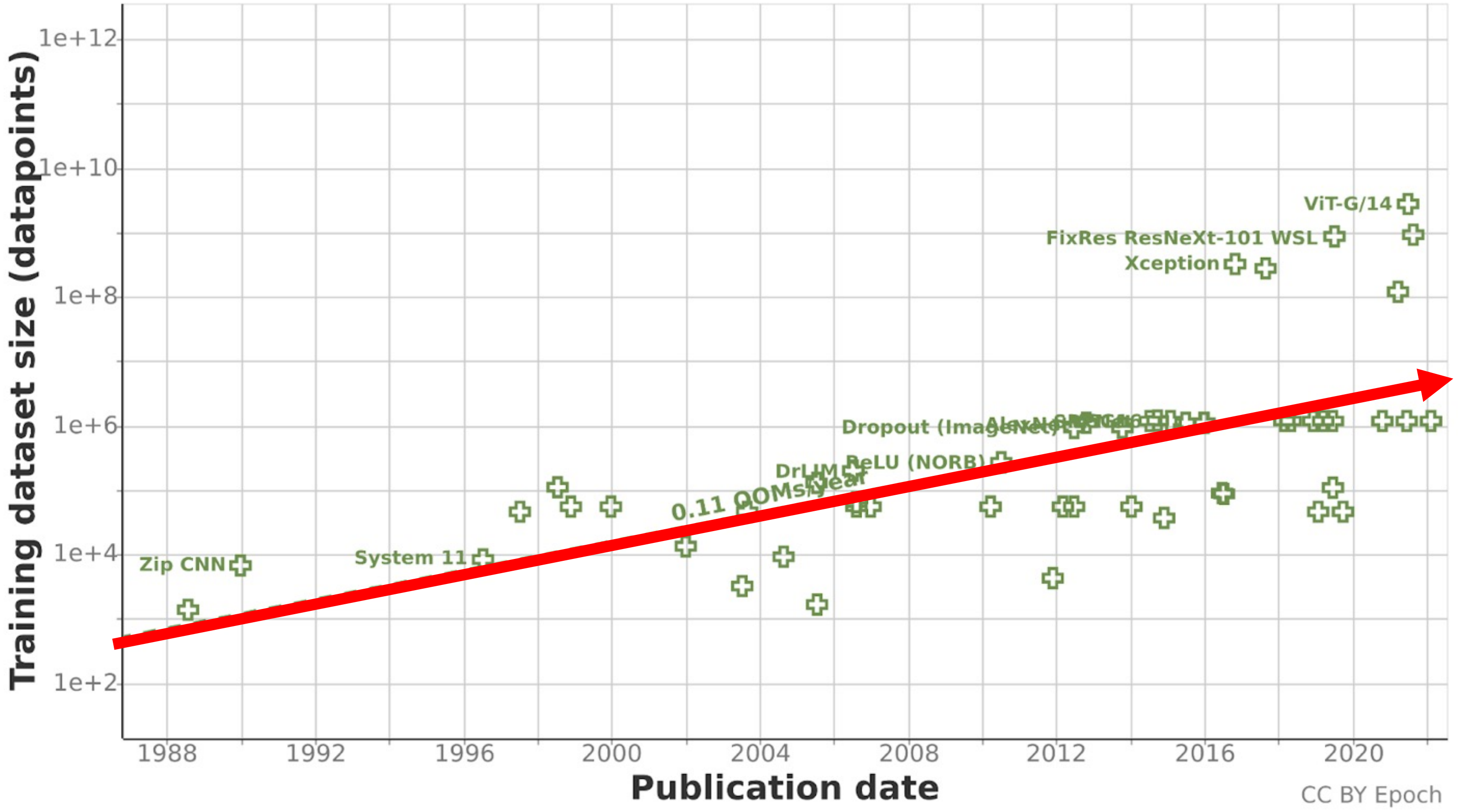
Problem 2: ML workloads are Data-Intensive



Samples

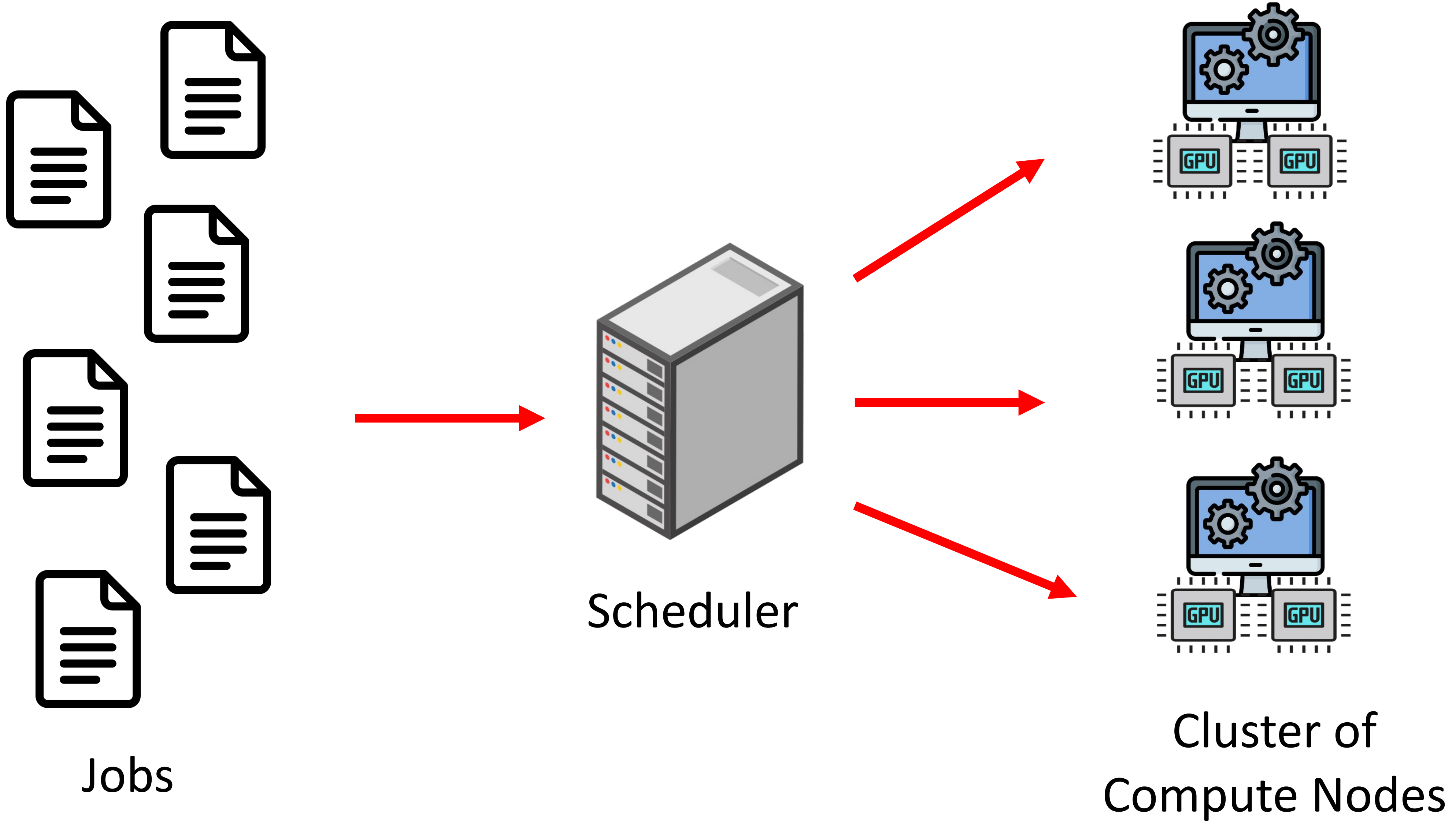
Size

I/O

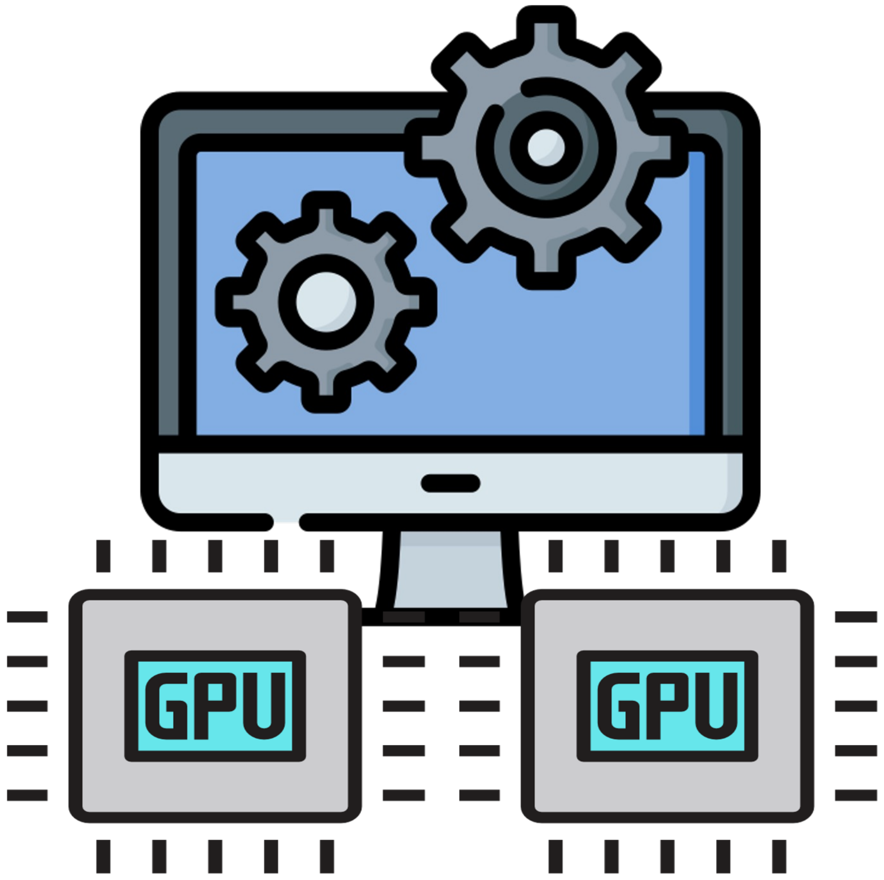


https://epochai.org/blog/trends-in-training-dataset-sizes

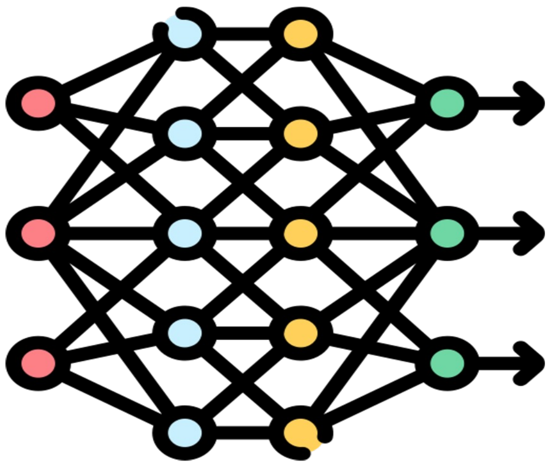
Problem 3: Modern Workloads are Resource-intensive



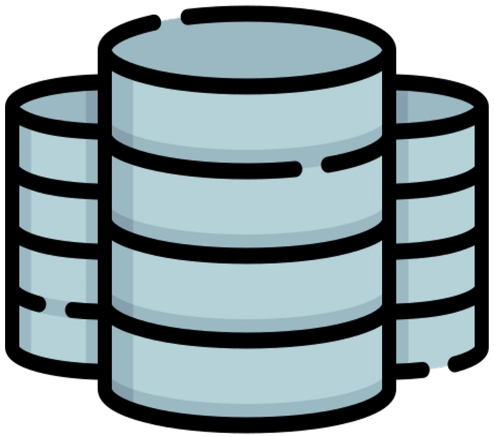
Challenge: Matching App Needs with System Resources



compute-intensive

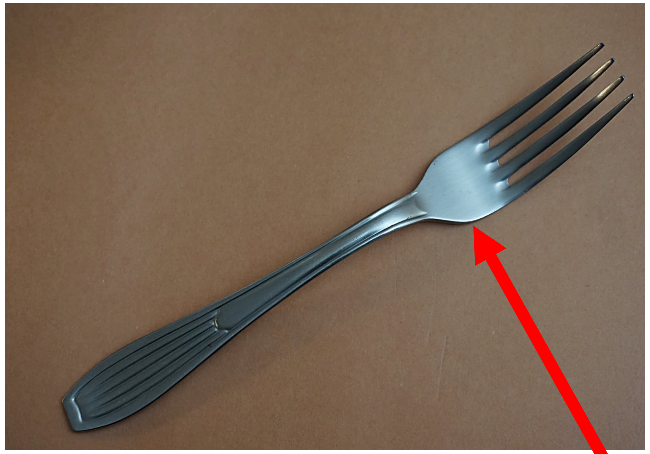


Applications



data-intensive

A Practical Example



Modern ML workloads

Traditional Systems

Traditional Workloads

New Systems

Solution: Become App + System Characteristic-aware

Application Characteristics → Data Characteristics, Training Characteristics, Model Characteristics

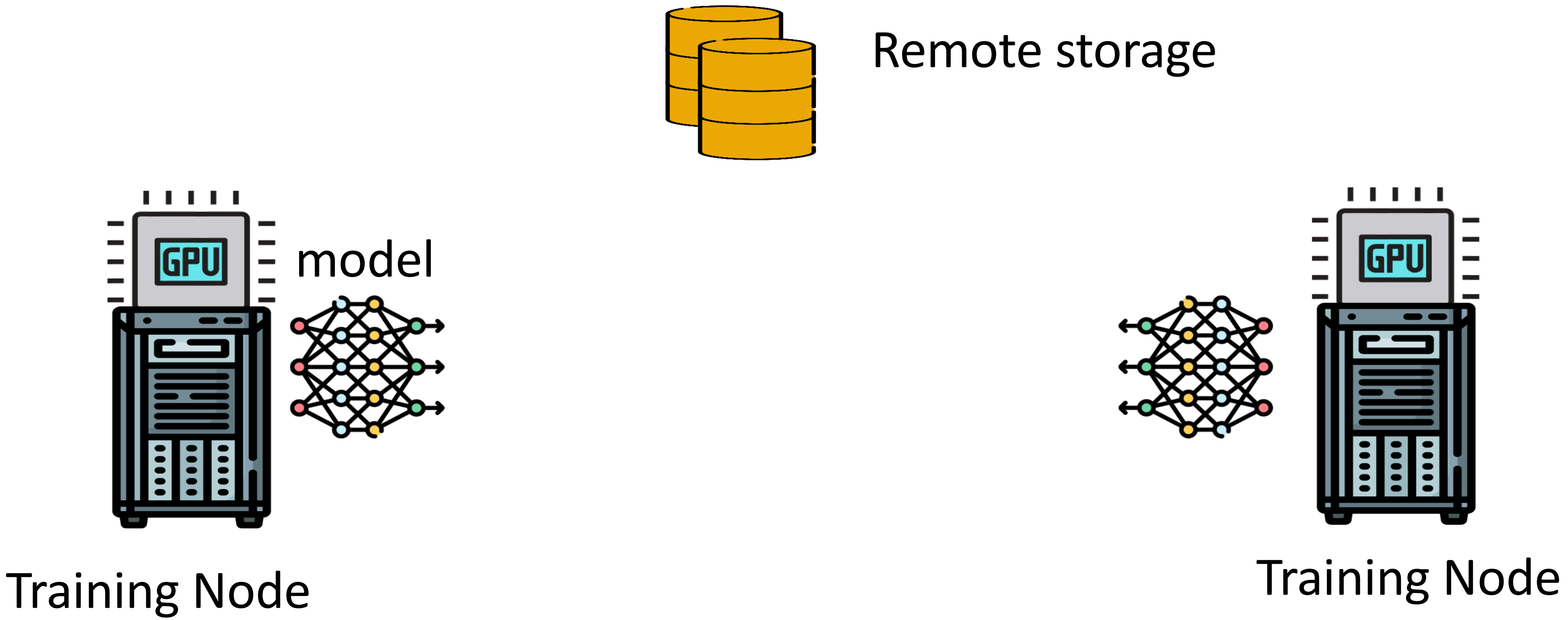
System Characteristics → I/O characteristics, Compute Characteristics, Cluster characteristics

Solution: Become App + System Characteristic-aware

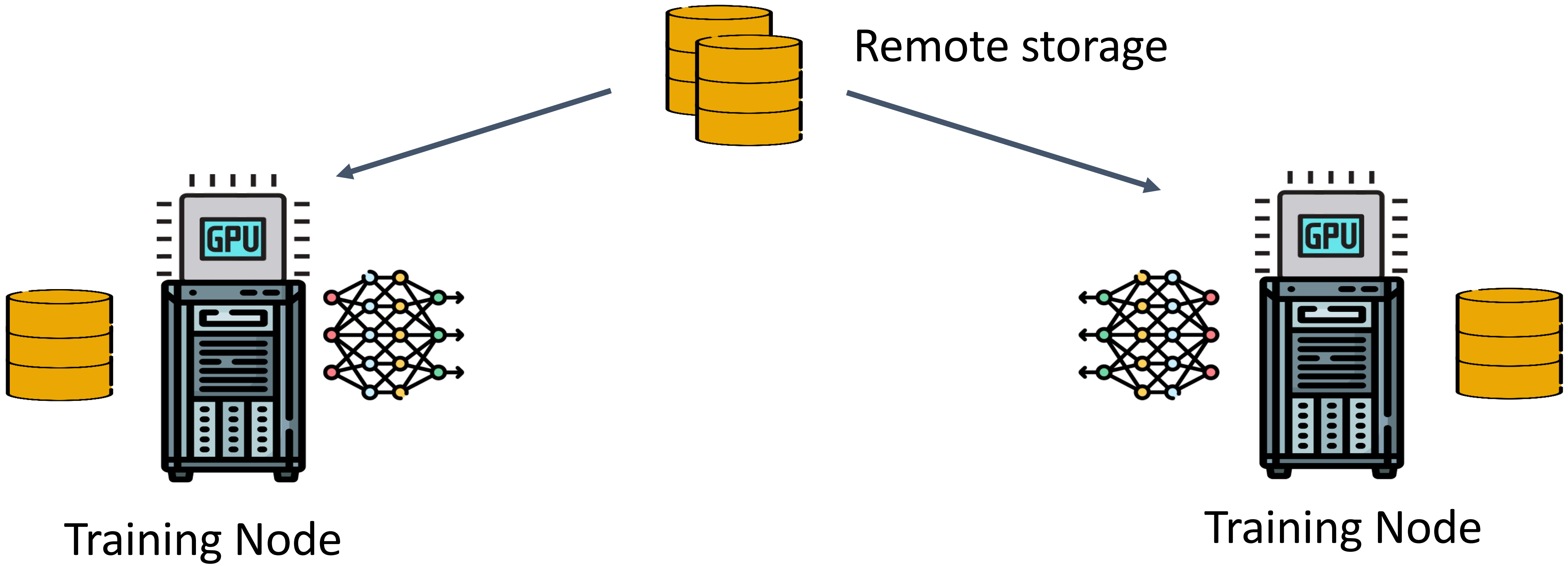
Application Characteristics
- Data Characteristics
- Training Characteristics
- Model Characteristics

System Characteristics
- I/O characteristics
- Compute Characteristics
- Cluster characteristics

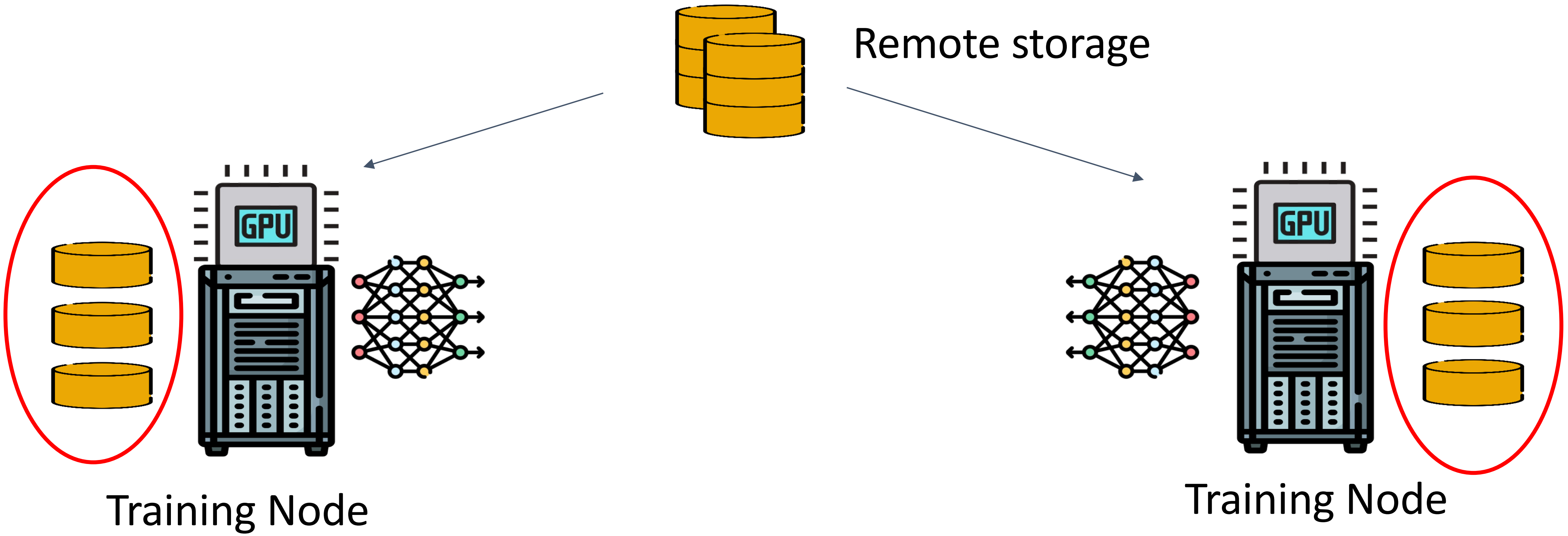
Data Storage Impacts the Performance of DL Training



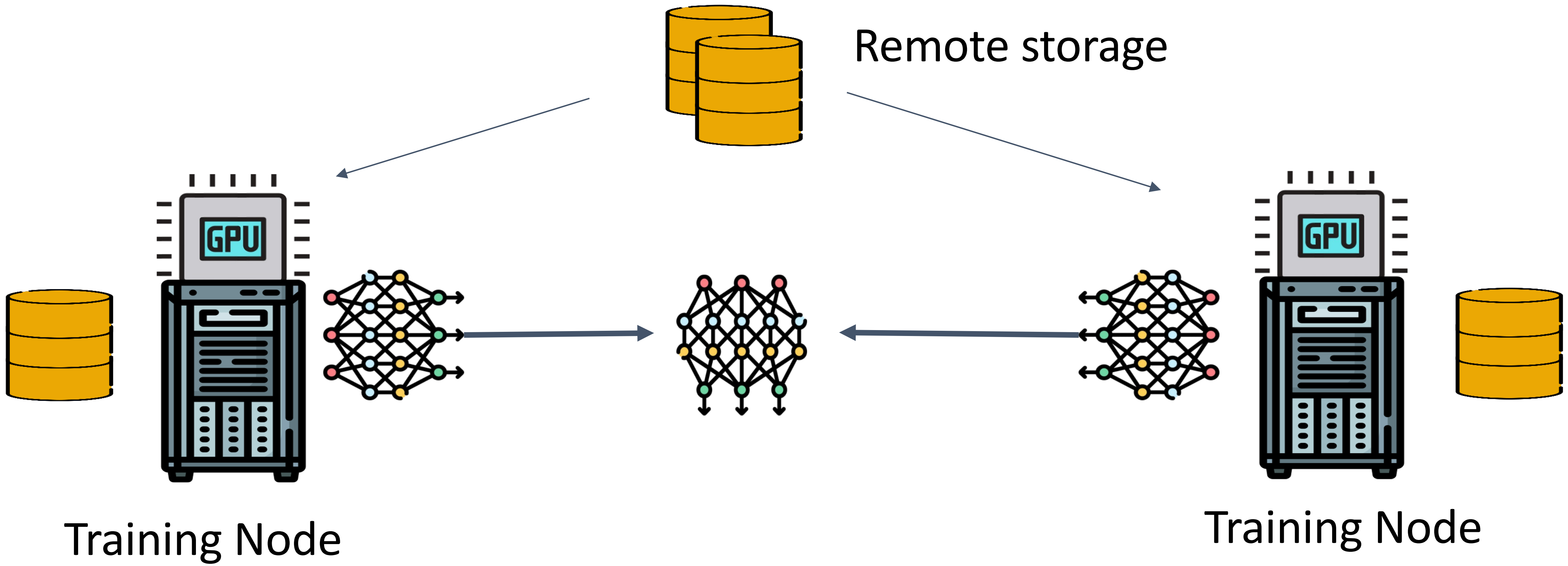
Data Storage Impacts the Performance of DL Training



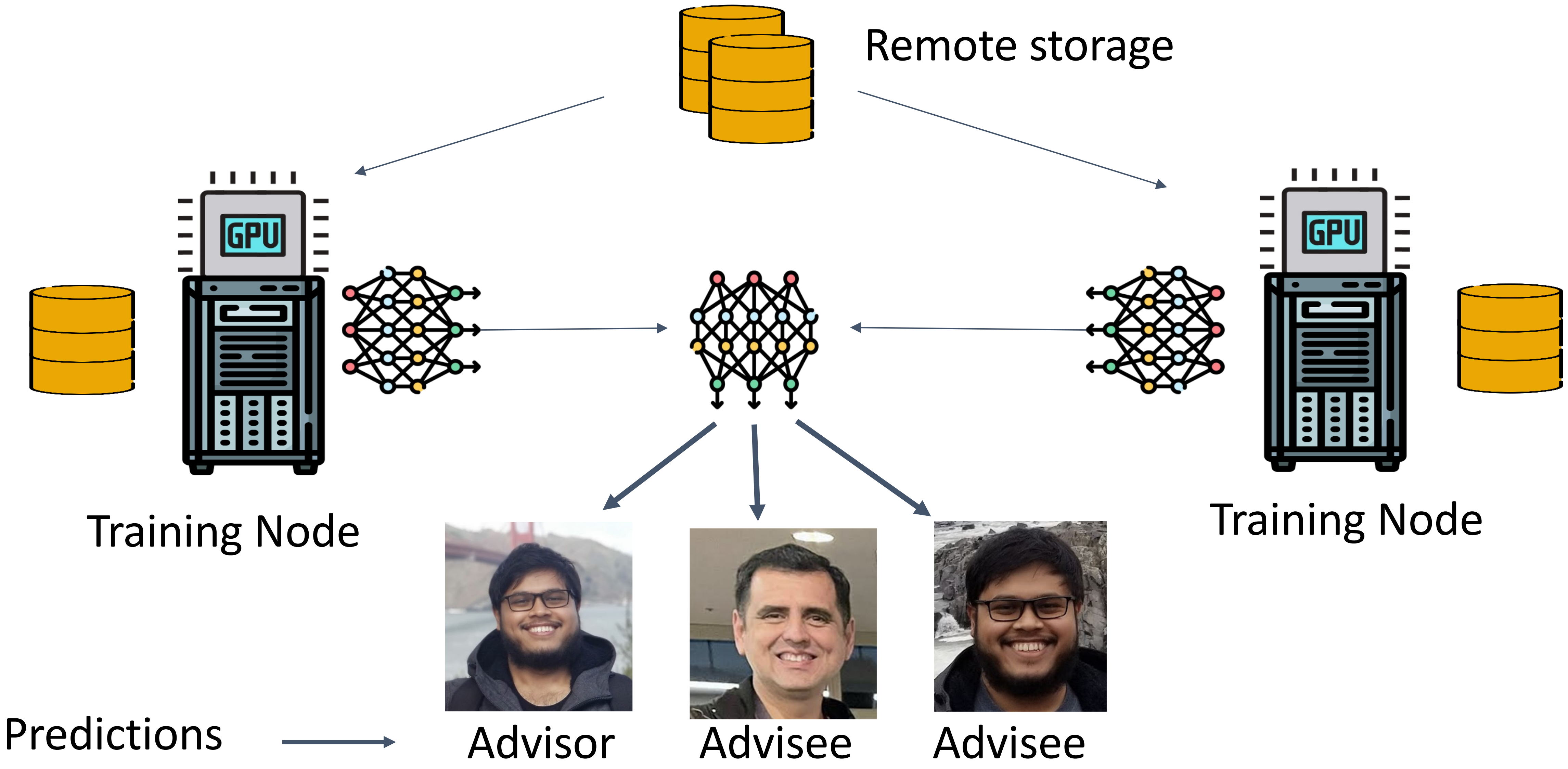
Data Storage Impacts the Performance of DL Training



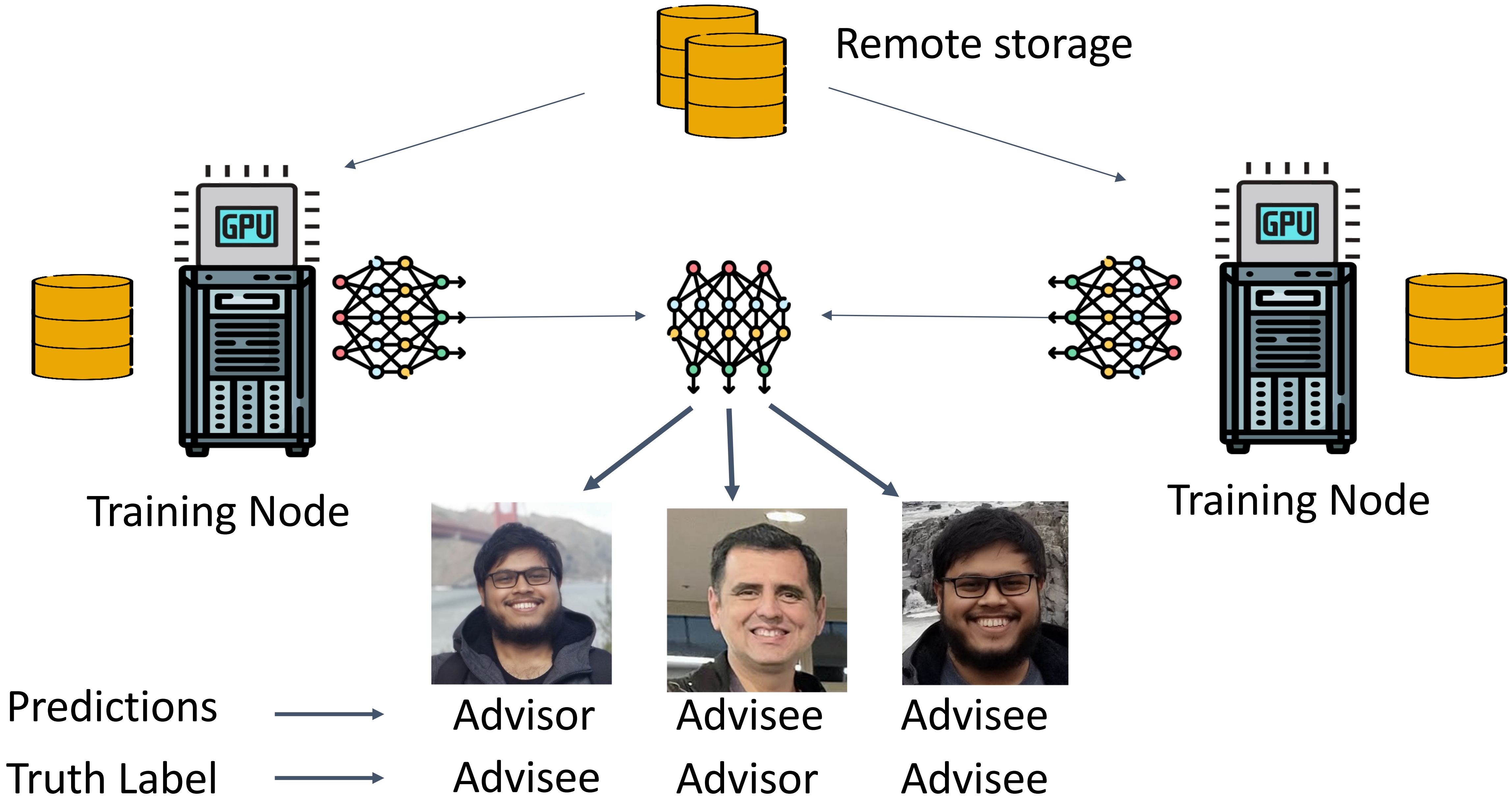
Data Storage Impacts the Performance of DL Training



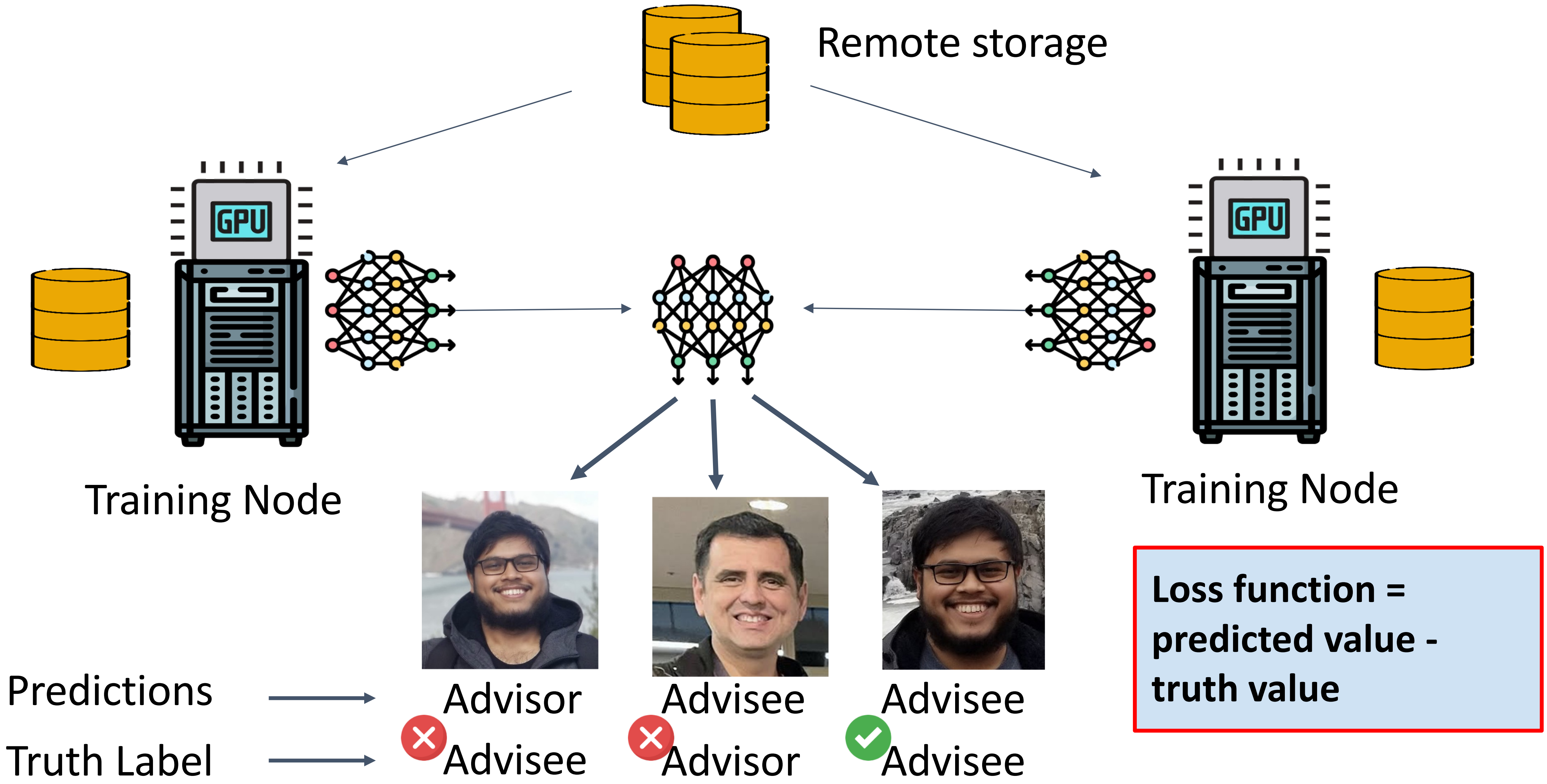
Data Storage Impacts the Performance of DL Training



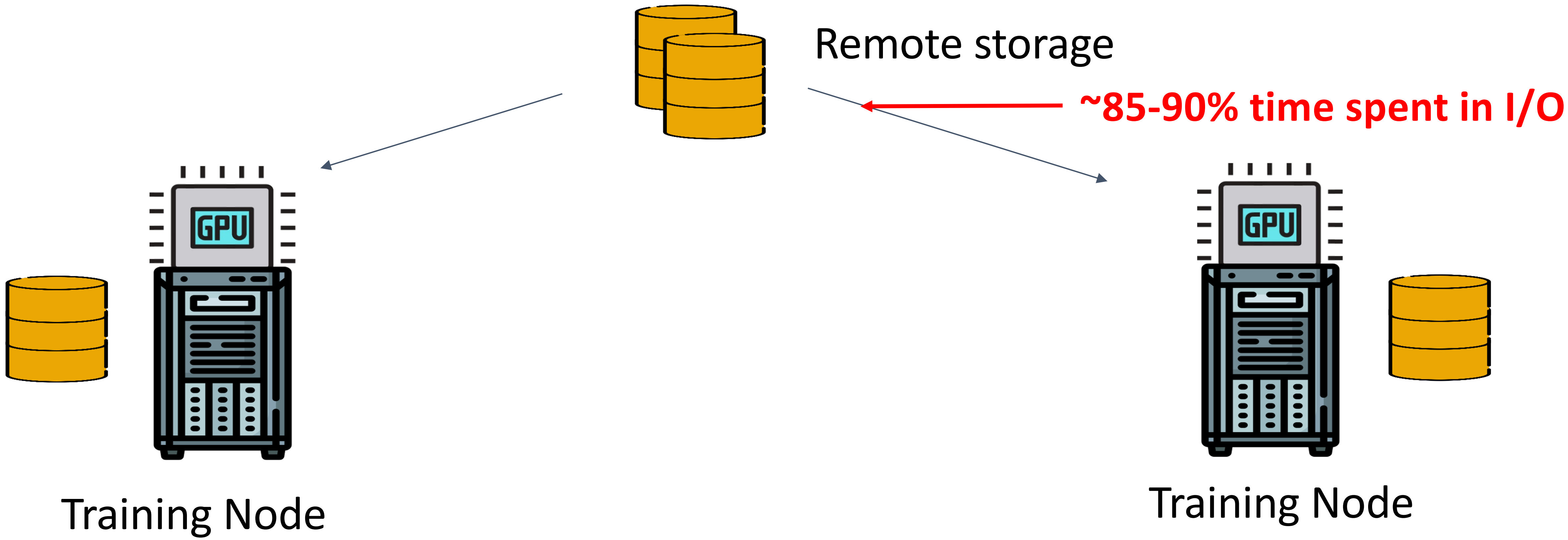
Data Storage Impacts the Performance of DL Training



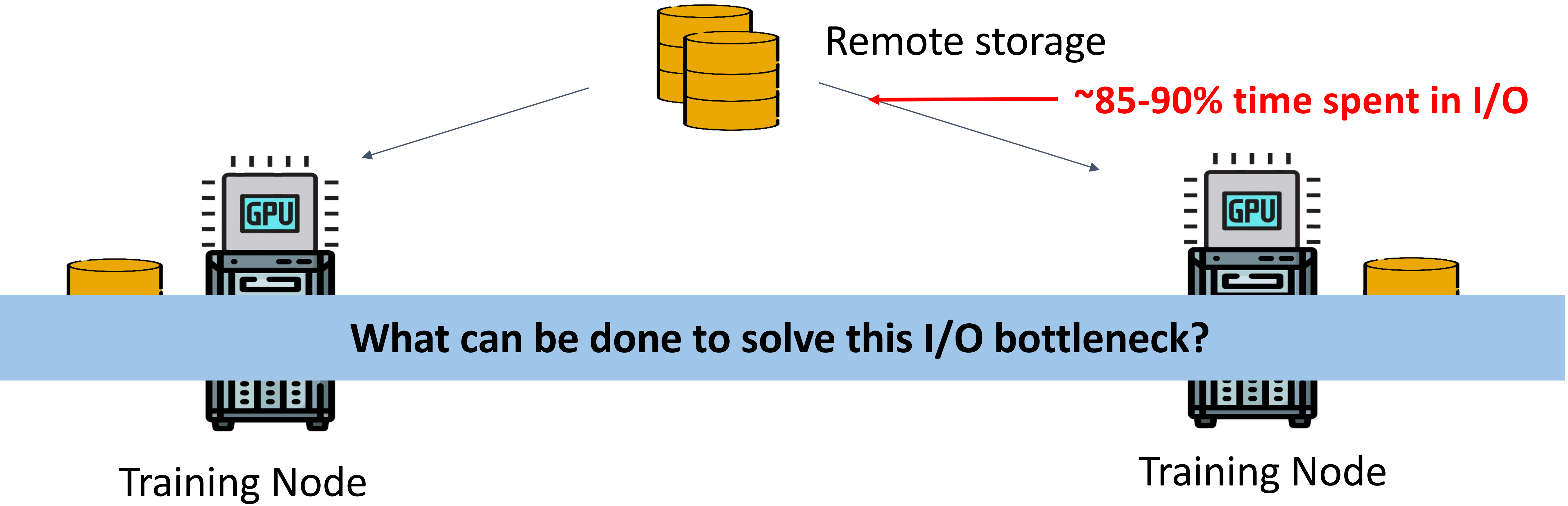
Data Storage Impacts the Performance of DL Training



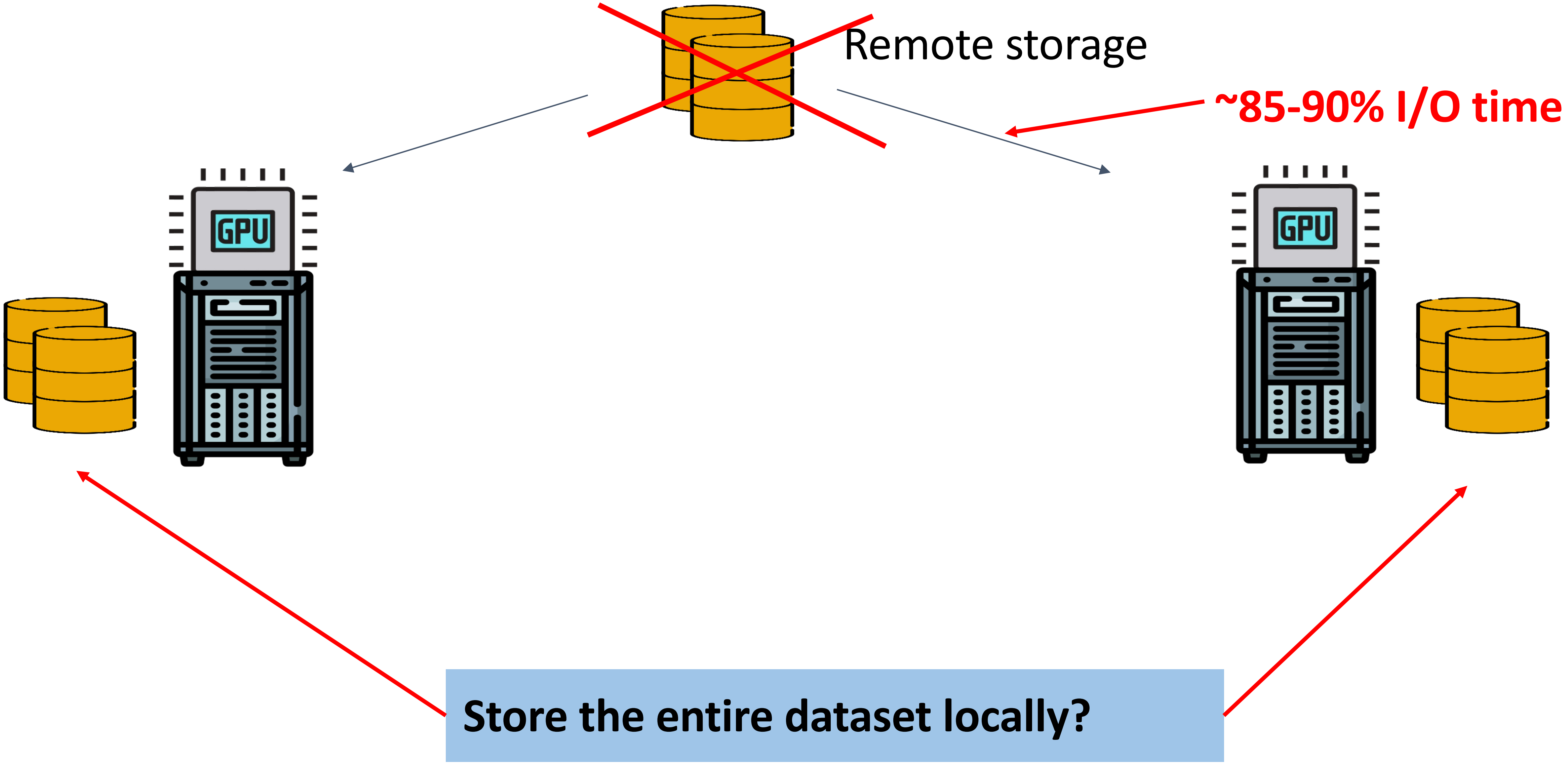
I/O is the Bottleneck in Distributed DL Training



I/O is the Bottleneck in Distributed DL Training

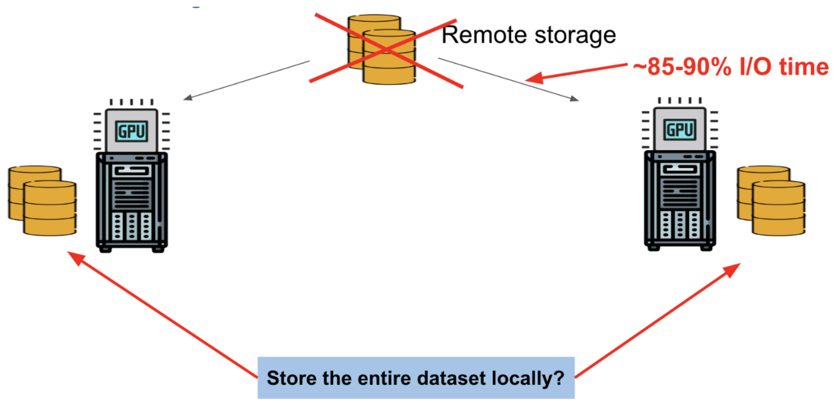


A Possible Solution - Local Storage



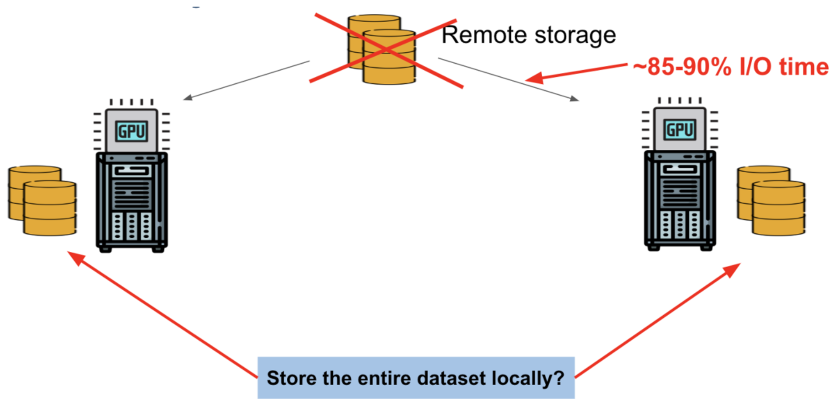
Problems with Storing Data Locally for DL Training

1. GPU VMs can lead to loss of local storage state



Problems with Storing Data Locally for DL Training

1. GPU VMs can lead to loss of local storage state
2. RAM size is very small



Exploiting System Characteristics - Can We Cache?

Goal: Improve performance using a small working set size (WSS)

Exploiting System Characteristics - Can We Cache?

Goal: Improve performance using a small working set size (WSS)

Caching DL workloads is non-trivial

- Data samples are fetched randomly
- All samples are accessed every epoch

Exploiting System Characteristics - Can We Cache?

Goal: Improve performance using a small working set size (WSS).

Caching DL workloads is non-trivial.

- Data samples are fetched randomly
- All samples are accessed every epoch

Which samples should we cache?

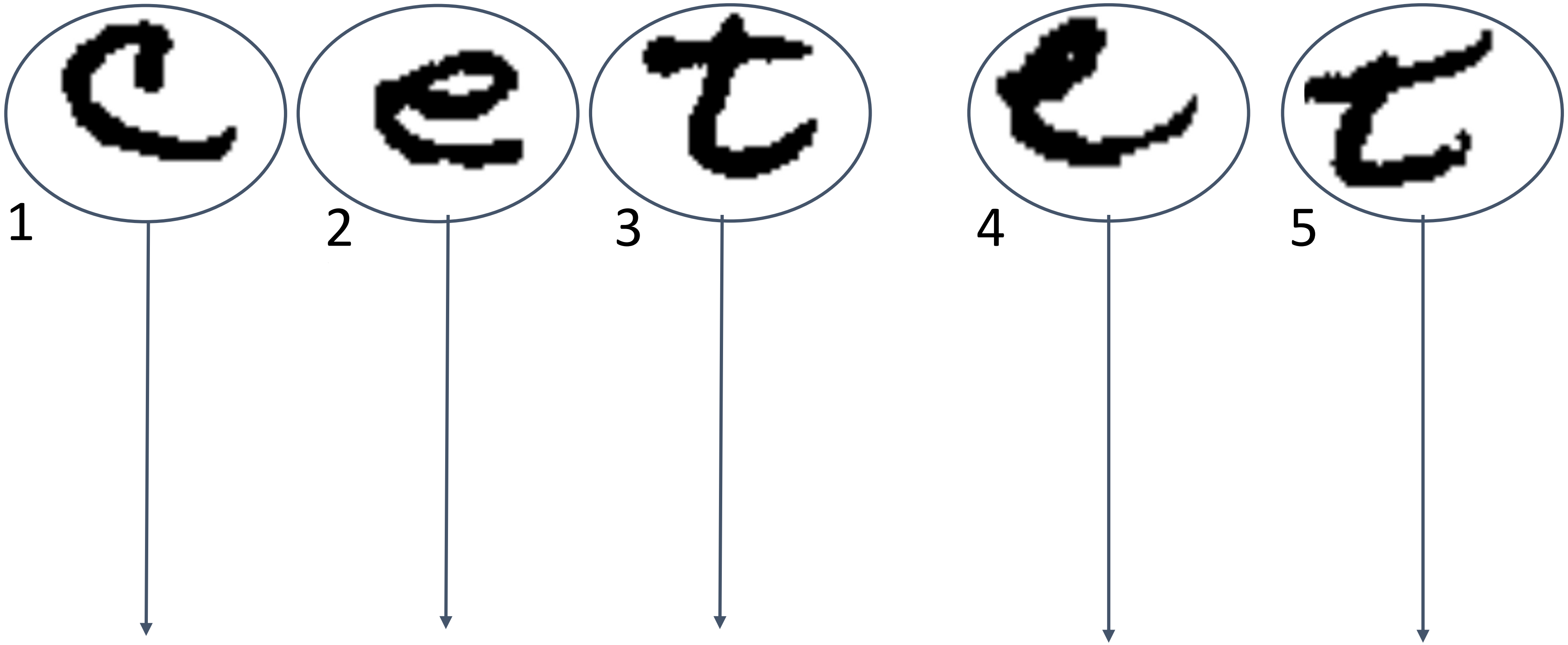
Exploiting Data Characteristics – Find the important ones!

Not all samples are equal[1,2]. Some are more important than others as they contribute more towards increasing the accuracy of a DL model.

1. Angelos Katharopoulos and Francois Fleuret. Not all samples are created equal: Deep learning with importance sampling. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 2525–2534. PMLR, 10–15 Jul 2018
2. Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. arXiv preprint arXiv:1511.06343, 2015.

Inequality Among Sample Contribution

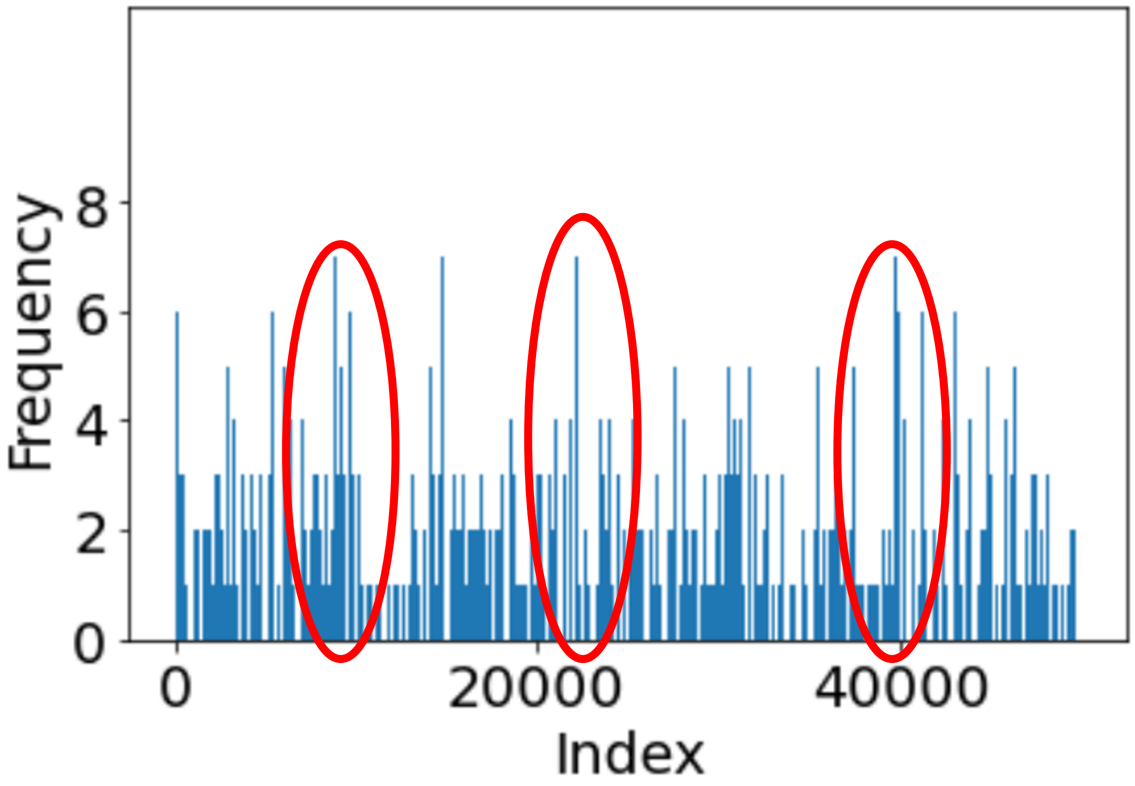
Hard-to-learn samples are the most important
Example: FEMNIST dataset



Not Important (easier-to-learn)

Important (harder-to-learn)

Repetitive Access Patterns are Amenable to Caching



Observations

~26% samples accessed more than once
~10% samples accessed more than three times

Key Insight of New System - SHADE

DL Training can treat different samples differently. If we can predict the I/O accesses, we can cache the samples to fundamentally improve the I/O efficiency.

Making Important Samples Cacheable is Challenging

1. Identify per-sample importance
2. Avoid making the model biased
3. Track importance scores

Challenge 1: Coarse-grained Importance Scores

Importance scores are generally coarse-grained and thus inaccurate.

Minibatch samples		Importance Score
(4, 5, 6)	→	0.4

Score(4) = Score(5) = Score(6) → Not accurate

Individual samples have their own contribution towards improving the accuracy of the model.

Challenge 1: Coarse-grained Importance Scores

How can we compare the importance across minibatches?

	Minibatch samples	Importance Scores
B1	(4, 5, 6)	(0.3, **0.5**, 0.4)
B2	(7, 8, 9)	(0.6, **0.9**, 0.8)

Most important (8, 9, 7, | 5, 6, 4) Least important

Relative comparison across minibatches become difficult.

Challenge 2: Biased Models

Minibatch samples

(6, **8, 8, 8**, 9)

Importance Scores

(0.5, 0.8, 0.8, 0.8, 0.7)

Let's use repeated samples to get more hits!

Challenge 2: Biased Models

Minibatch samples

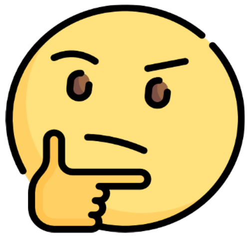
(6, **8, 8, 8**, 9)

Importance Scores

(0.5, 0.8, 0.8, 0.8, 0.7)

Let's use repeated samples to get more hits!

But, wait a sec!



Would training on repeated samples lead to biased results?

Challenge 3: Constant Change of Importance Scores

Importance scores are dynamic

Minibatch samples

(6, **7**, **8**, 9)

Current score →

Importance Scores

(0.6, **0.8**, **0.9**, 0.6)

Let's put 7 and 8 in cache!

Challenge 3: Constant Change of Importance Scores

Importance scores are dynamic

Minibatch samples

(6, 7, 8, 9)

Current score

Future score

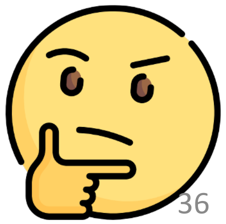
Importance Scores

(0.6, **0.8**, **0.9**, 0.6)

Let's put 7 and 8 in cache!

(**0.5**, 0.3, 0.4, **0.6**)

If cache is not updated, most important samples will not be in cache.



Making Important Samples Amenable to Caching

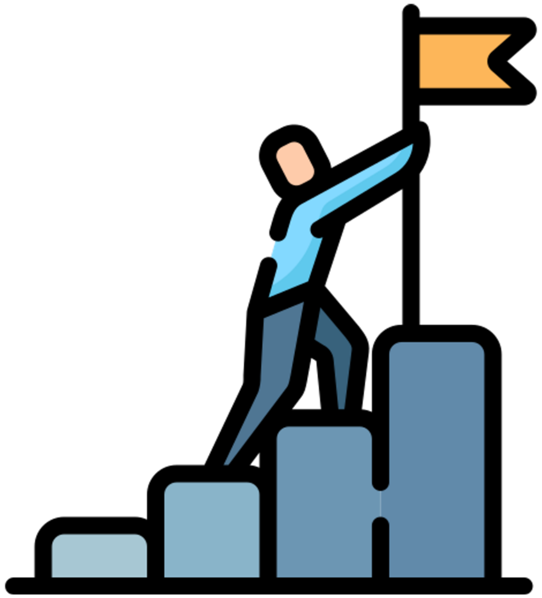
Loss Decomposition

Rank-based Score

Priority-based Adaptive Sampling (*PADS*)

Adaptive Priority-aware Prediction (*APP*) Cache

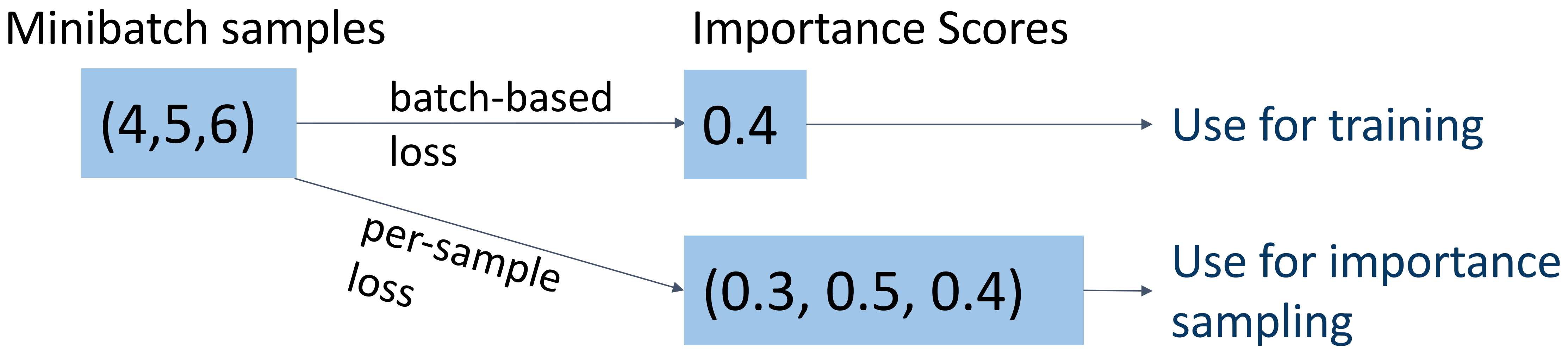
SHADE



Khan, Redwan Ibne Seraj, et al. "{SHADE}: Enable Fundamental Cacheability for Distributed Deep Learning Training." *21st USENIX Conference on File and Storage Technologies (FAST 23)*. 2023.

Technique 1: Loss Decomposition

Tackling coarse-grained nature of importance scores



Use coarse-grained loss for training, fine-grained loss for importance sampling

Technique 2: Rank-based Importance Score

Adapting importance scores for priority-based caching

	Minibatch samples		Raw scores		Ranked scores
B1	(4,5,6)	→	(0.3, 0.5, 0.4)	→	(0, 2, 1)
B2	(7,8,9)	→	(0.6, 0.9, 0.8)	→	(0, 2, 1)

Rank fine-grained losses to detect relative importance.

Technique 3: Priority-based Adaptive Sampling (*PADS*)

Adapting sampling based on loss/accuracy change to avoid bias

Samples

4, 7, 3, 5, 3, 5, 2, 2, 1, 10 → Create samples list with repetitions based on importance.
Higher repetitions → Higher importance

3, 3, 5, 5, 2 → Send optimal list for training and caching
Cache → (3,5)
Hits → (3,3,5,5); Hit rate → 80%

Monitor the loss + accuracy in real time

Increase hit rate at will while keeping the accuracy improvement in check

Technique 4: Adaptive Priority-aware Prediction (*APP*)

Keep the most important samples in cache

Priority Queue(PQ) → importance of currently cached samples
Ghost Cache → importance of all trained samples.

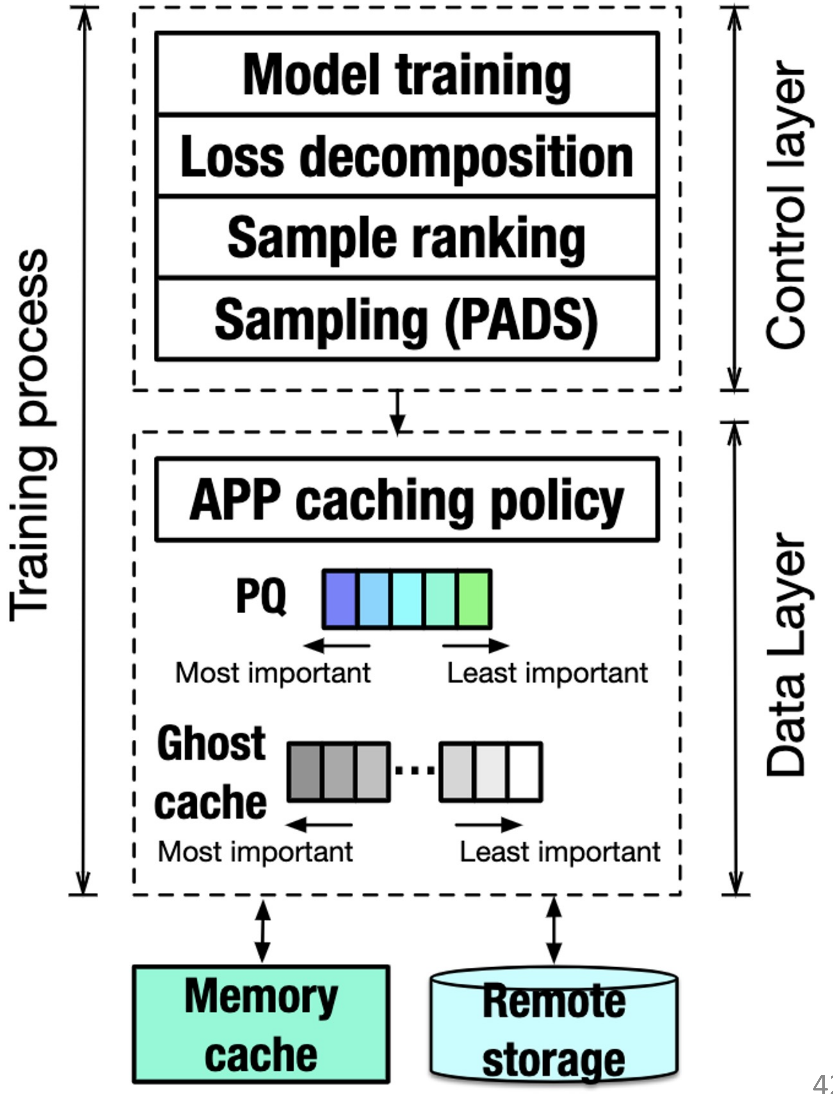
Compare(importance of cached sample, importance of incoming sample)

Get from PQ

Get from ghost cache

Compare importance scores to keep most important samples in cache

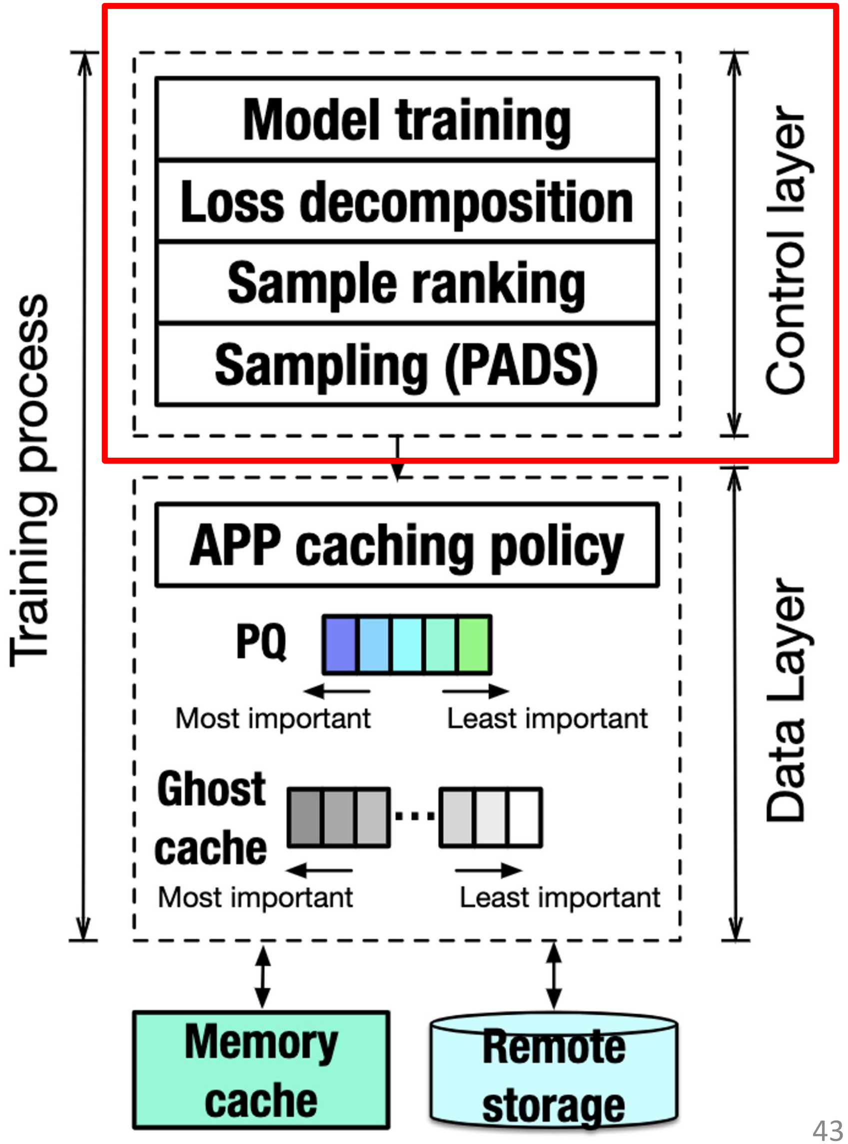
SHADE Architecture



SHADE Architecture

SHADE Control Layer

- Finding fine-grained importance, ranking them, and sampling



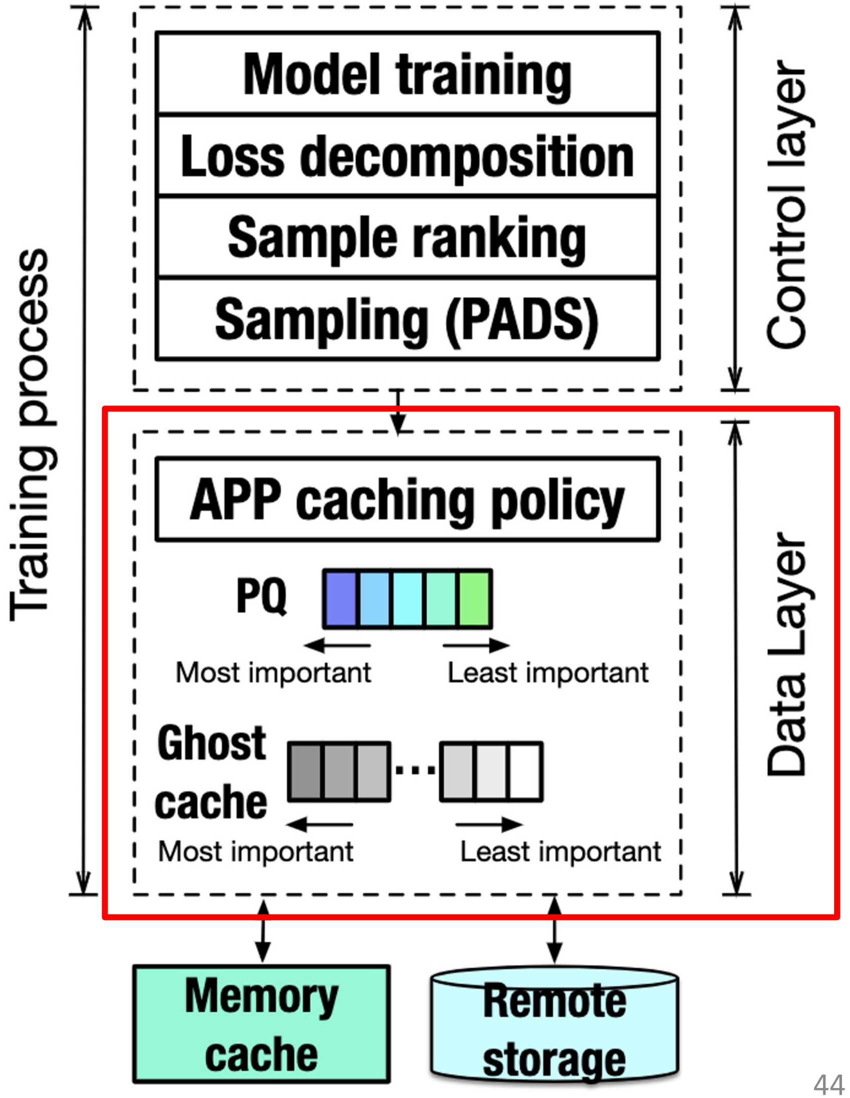
SHADE Architecture

SHADE Control Layer

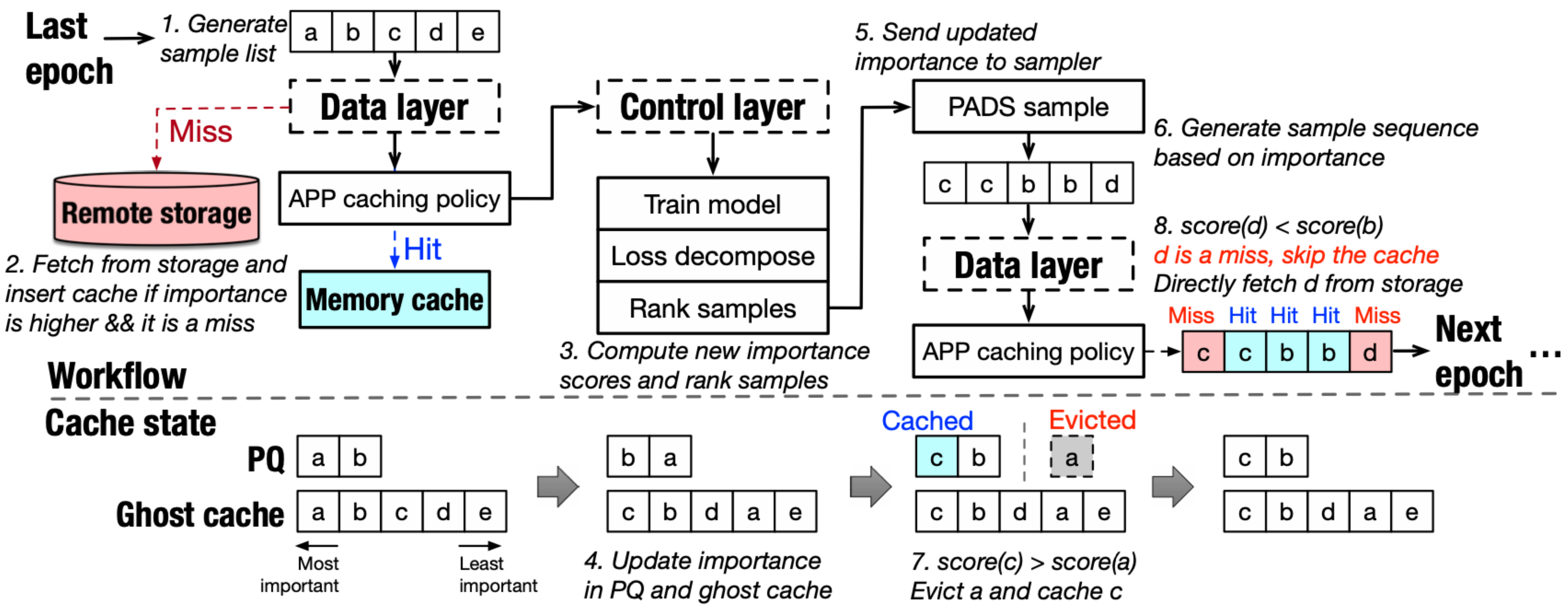
- Finds fine-grained importance, ranks, and samples data

SHADE Data Layer

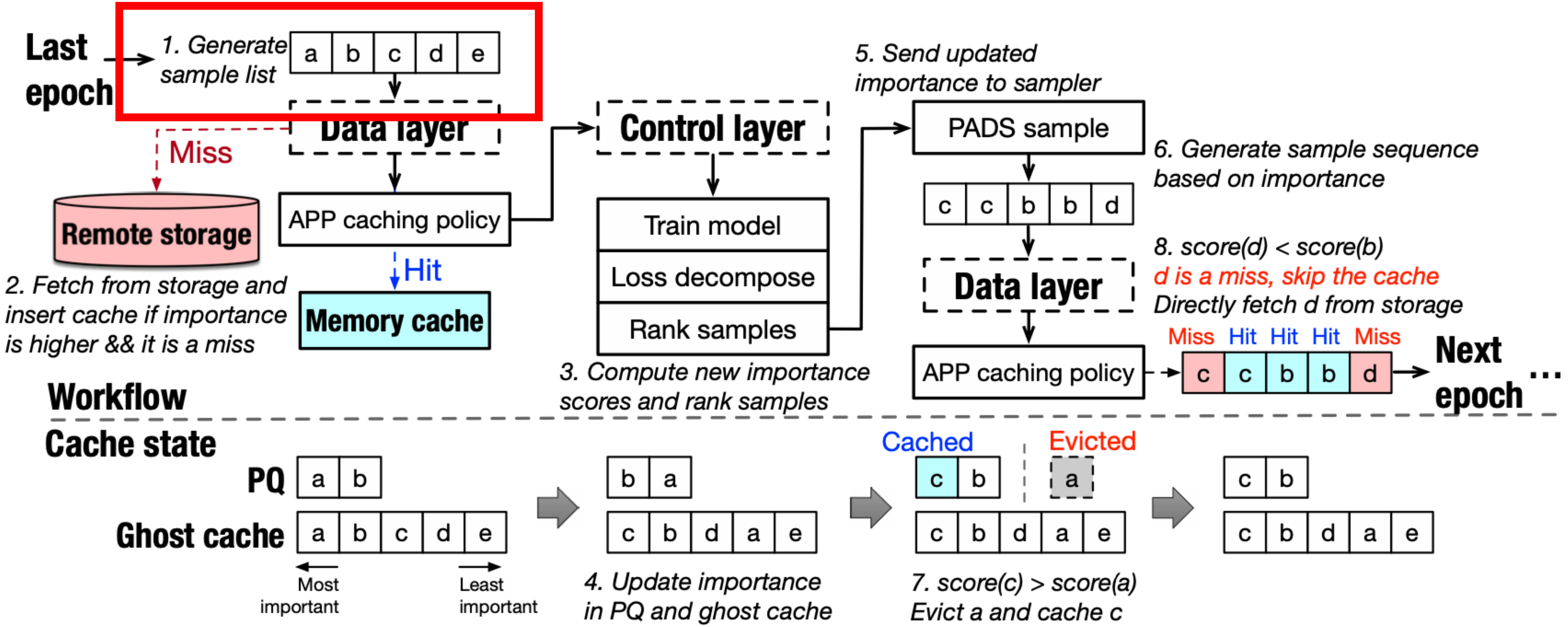
- Stores and retrieves from cache
- Updates the cache



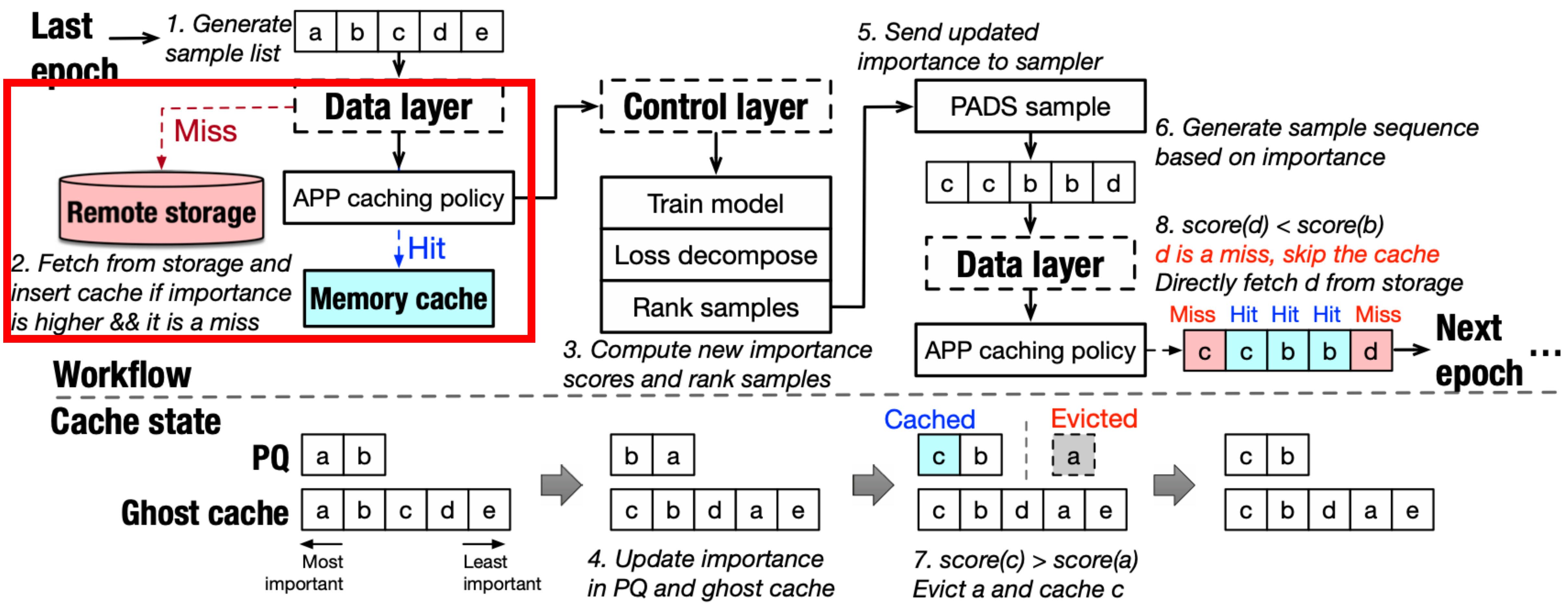
SHADE Workflow



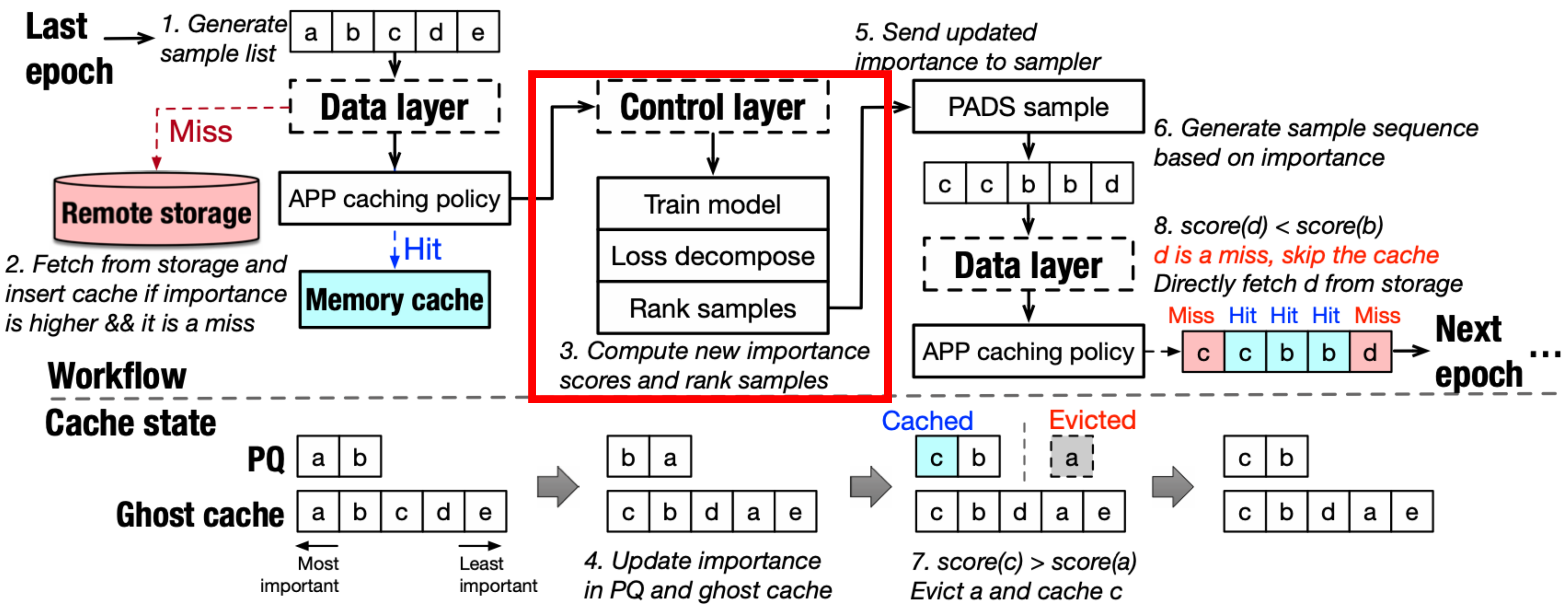
SHADE Workflow



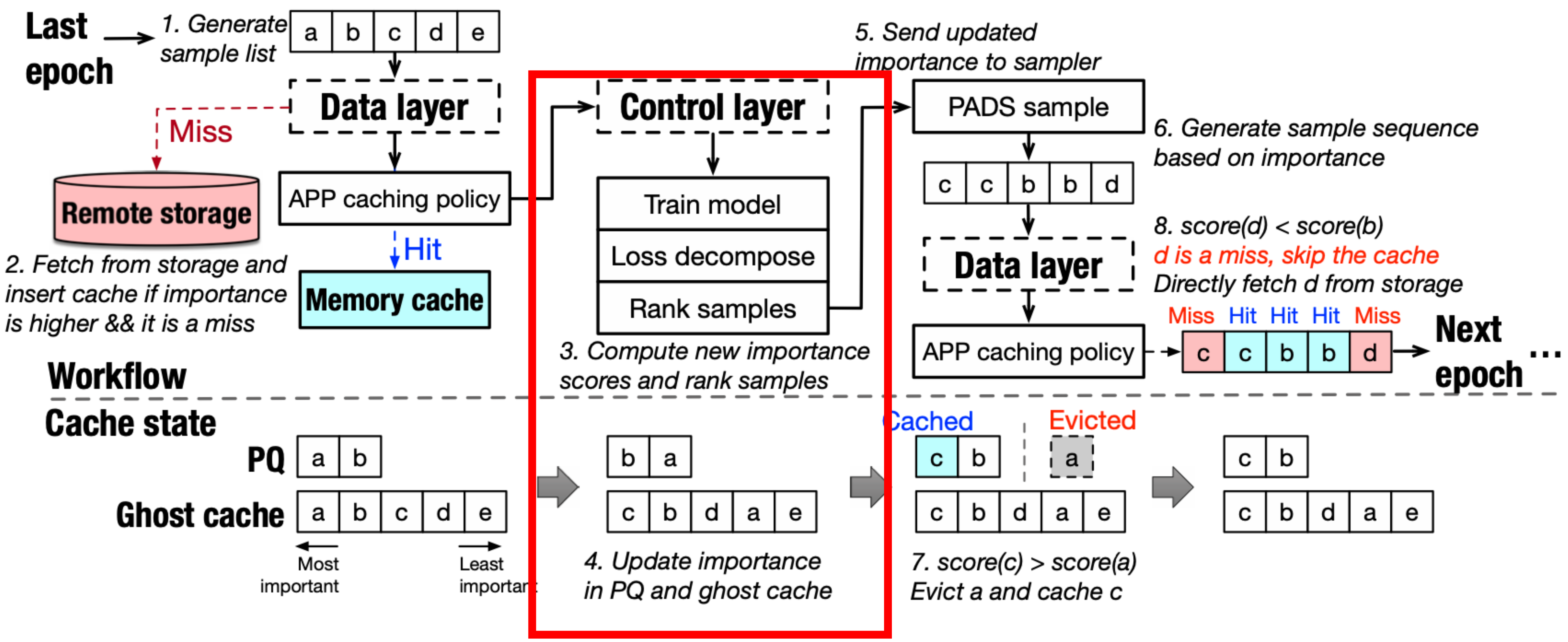
SHADE Workflow



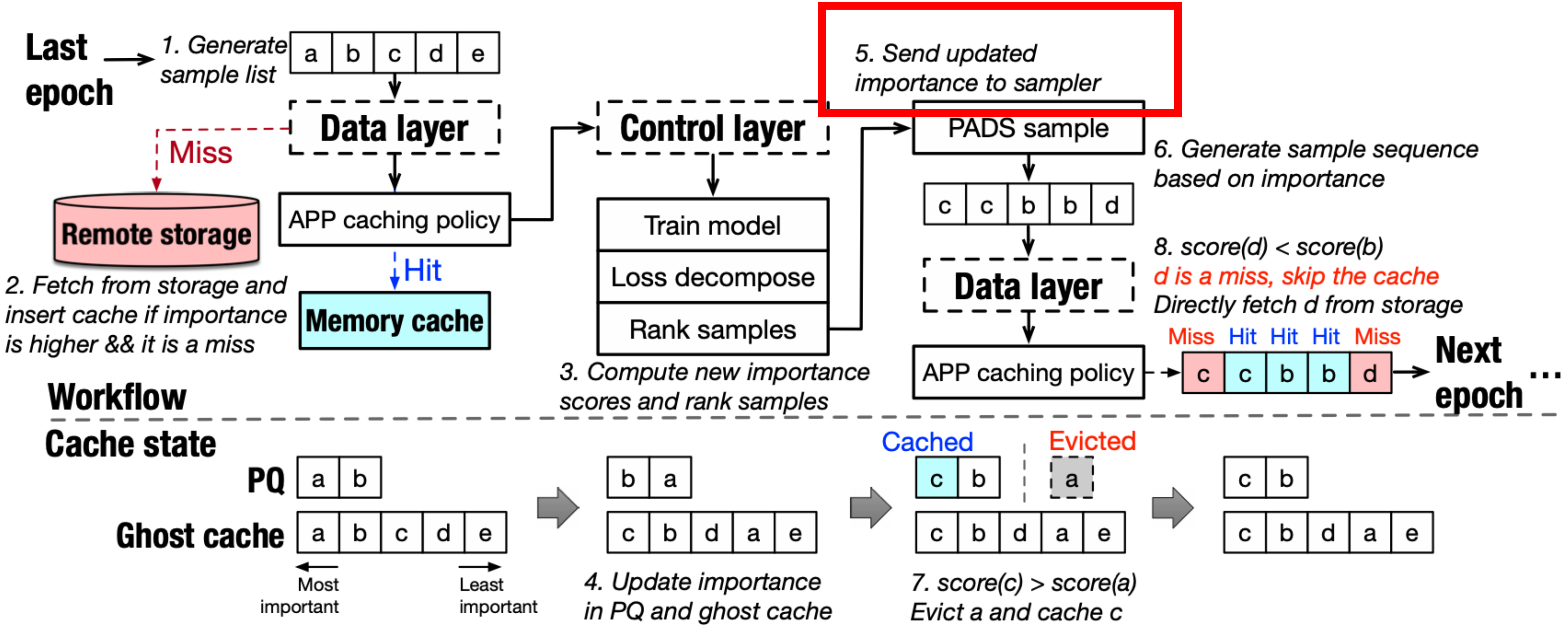
SHADE Workflow



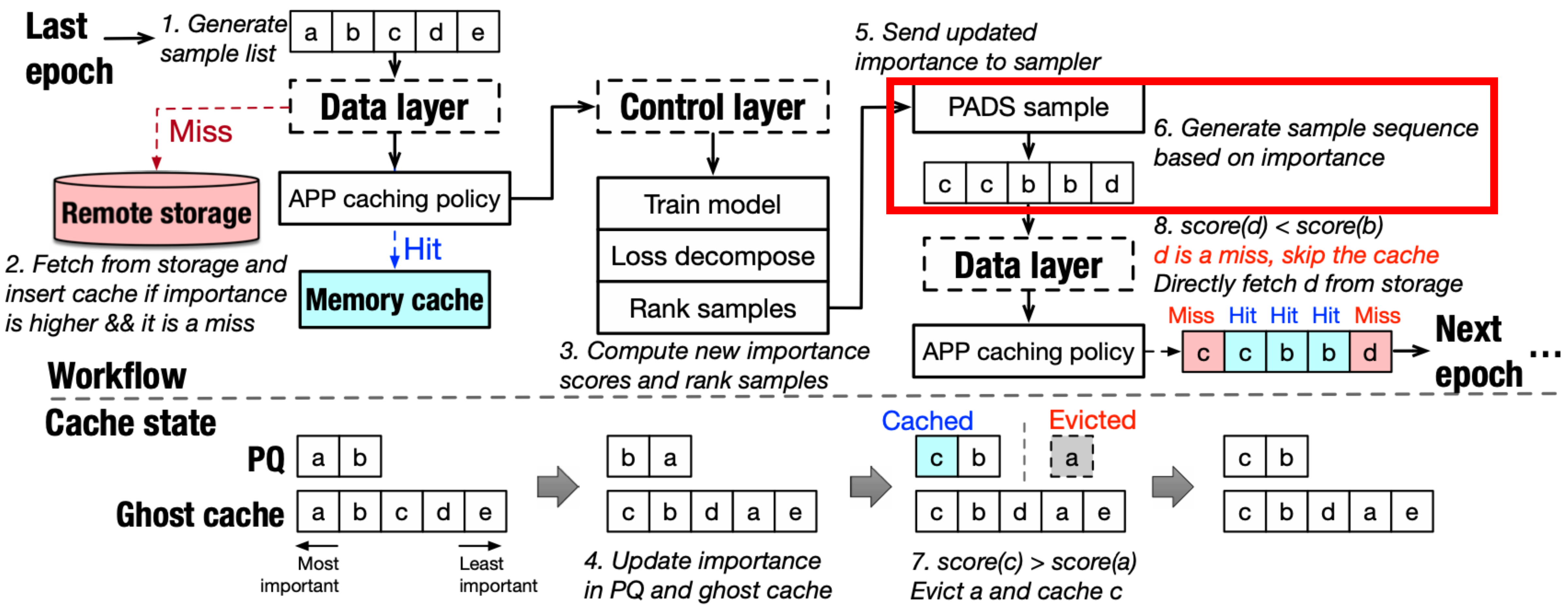
SHADE Workflow



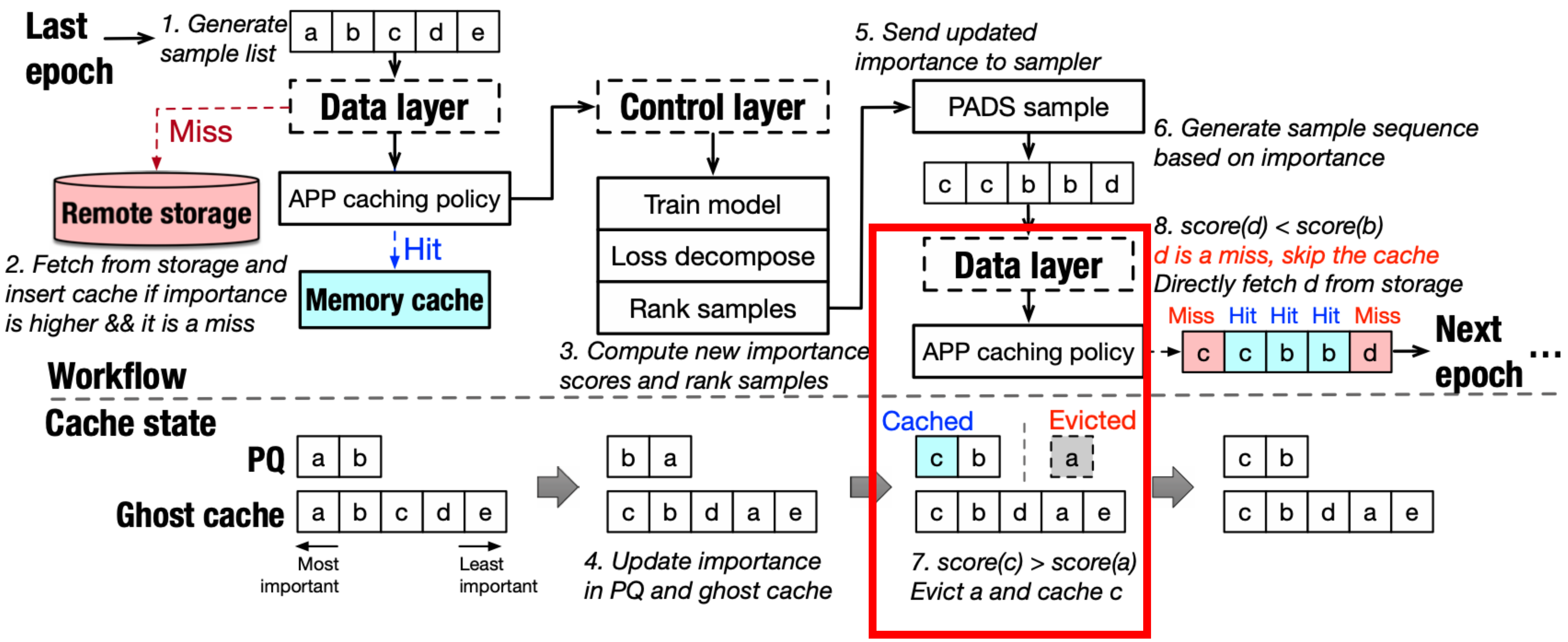
SHADE Workflow



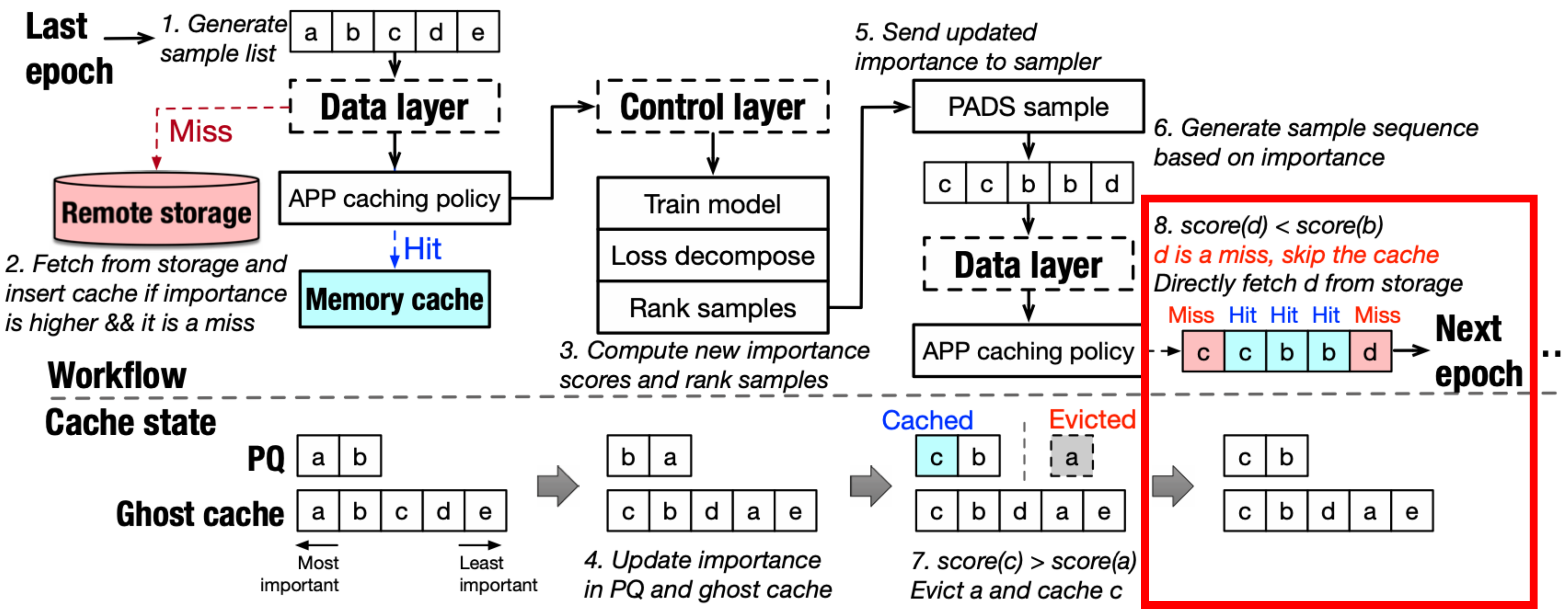
SHADE Workflow



SHADE Workflow



SHADE Workflow



SHADE Feature: Ease of Deployment

Default PyTorch

```
train_dataset = datasets.ImageFolder(
                        train_directory,
                transform_function)
val_dataset = datasets.ImageFolder(
                        val_directory,
                        transform_function)
train_sampler =
DistributedSampler(train_dataset)
Train()
Validate()
```

SHADE

```
train_dataset = ShadeDataset(
                        train_directory,
                transform_function,...)
val_dataset = ShadeValDataset(
                        val_directory,
                        transform_function)
train_sampler =
ShadeSampler(train_dataset)
Train()
Validate()
```

Evaluation Setup

Cluster

- 4 nodes each having 2 P100 GPUs
- HDD-based NFS server as a remote storage

Datasets

- ImageNet-1K
- CIFAR-10

Models

- AlexNet, ResNet-18, ResNet-50, VggNet

Baseline

- LRU incorporated PyTorch Distributed Training

Evaluation Objectives

1. Read Hit Ratio
2. Accuracy vs. Time
3. Throughput
4. Minibatch Load Time

Belady’s Optimal Cache Replacement Algorithm - MIN

- Pages are replaced which would not be used for the longest duration in the future
- Optimal i.e. perfect policy as it has knowledge about the future.
- Not possible in practice as OS cannot know future requests.
- Used for setting up benchmarks so that other replacement algorithms can be analyzed against it.

Belady’s MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3
Hit or Miss?														

Cache Space

Example adapted and modiified from Geeksforgeeks.

Belady’s MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space														
	7													
Hit or Miss?	Miss													

Belady’s MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

		0												
	7	7												
Hit or Miss?	Miss	Miss												

Cache Space

Belady's MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

			1											
		0	0											
	7	7	7											
Hit or Miss?	Miss	Miss	Miss											

Cache Space

Belady’s MIN example

Time stamp	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2										
			1	1										
		0	0	0										
	7	7	7	7										
Hit or Miss?	Miss	Miss	Miss	Miss										

Cache Space

Belady’s MIN example

Time stamp	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2									
			1	1	1									
		0	0	0	0									
	7	7	7	7	7									
Hit or Miss?	Miss	Miss	Miss	Miss	Hit									

Cache Space

Belady's MIN example

Time difference between current and future usage of pages in cache

- 2 → 9 - 6 = 3
- 1 → 7 - 6 = 1
- 0 → 11 - 6 = 5
- 7 → inf – 6 = inf [EVICT]

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2								
			1	1	1	1								
		0	0	0	0	0								
	7	7	7	7	7	3								
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss								

Belady’s MIN example

Time stamp	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2							
			1	1	1	1	1							
		0	0	0	0	0	0							
	7	7	7	7	7	3	3							
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit							

Cache Space

Belady’s MIN example

Time difference between current and future usage of pages in cache

- 2 → 9 - 8 = 1
- 1 → inf - 8 = inf [EVICT]
- 0 → 11 - 8 = 3
- 3 → 10 – 8 = 2

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2						
			1	1	1	1	1	4						
		0	0	0	0	0	0	0						
	7	7	7	7	7	3	3	3						
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss						

Cache Space

Belady's MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2	2					
			1	1	1	1	1	4	4					
		0	0	0	0	0	0	0	0					
	7	7	7	7	7	3	3	3	3					
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit					

Cache Space

Belady’s MIN example

Time stamp	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2	2	2				
			1	1	1	1	1	4	4	4				
		0	0	0	0	0	0	0	0	0				
	7	7	7	7	7	3	3	3	3	3				
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit				

Cache Space

Belady’s MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2	2	2	2			
			1	1	1	1	1	4	4	4	4			
		0	0	0	0	0	0	0	0	0	0			
	7	7	7	7	7	3	3	3	3	3	3			
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit			

Cache Space

Belady’s MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2	2	2	2	2		
			1	1	1	1	1	4	4	4	4	4		
		0	0	0	0	0	0	0	0	0	0	0		
	7	7	7	7	7	3	3	3	3	3	3	3		
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit		

Cache Space

Belady’s MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2	2	2	2	2	2	
			1	1	1	1	1	4	4	4	4	4	4	
		0	0	0	0	0	0	0	0	0	0	0	0	
	7	7	7	7	7	3	3	3	3	3	3	3	3	
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit	Hit	

Cache Space

Belady’s MIN example

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2	2	2	2	2	2	2
			1	1	1	1	1	4	4	4	4	4	4	4
		0	0	0	0	0	0	0	0	0	0	0	0	0
	7	7	7	7	7	3	3	3	3	3	3	3	3	3
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit	Hit	Hit

Cache Space

Belady’s MIN example

Total Page Faults = 6

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	2	2	2	2	2	2	2
			1	1	1	1	1	4	4	4	4	4	4	4
		0	0	0	0	0	0	0	0	0	0	0	0	0
	7	7	7	7	7	3	3	3	3	3	3	3	3	3
Hit or Miss?	**Miss**	**Miss**	**Miss**	**Miss**	Hit	**Miss**	Hit	**Miss**	Hit	Hit	Hit	Hit	Hit	Hit

Cache Space

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

															Cache Space
Hit or Miss?															

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

	7													
Hit or Miss?	Miss													

Cache Space

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

		0												
	7	7												
Hit or Miss?	Miss	Miss												

Cache Space

Comparison with LRU

Time stamp	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

			1											
		0	0											
	7	7	7											
Hit or Miss?	Miss	Miss	Miss											

Cache Space

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2										
			1	1										
		0	0	0										
	7	7	7	7										
Hit or Miss?	Miss	Miss	Miss	Miss										

Cache Space

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2									
			1	1	1									
		0	0	0	0									
	7	7	7	7	7									
Hit or Miss?	Miss	Miss	Miss	Miss	Hit									

Comparison with LRU

Last Used Timestamp

- 2 → 4
- 1 → 3
- 0 → 5
- 7 → 1 [Least recently used] [EVICT]

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2								
			1	1	1	1								
		0	0	0	0	0								
	7	7	7	7	7	3								
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss								

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2	2							
			1	1	1	1	1							
		0	0	0	0	0	0							
	7	7	7	7	7	3	3							
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit							

Comparison with LRU

Last Used Timestamp

- 2 → 4 [Least Recently Used] [Evict]
- 1 → 7
- 0 → 5
- 3 → 6

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2	2	4						
			1	1	1	1	1	1						
		0	0	0	0	0	0	0						
	7	7	7	7	7	3	3	3						
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss						

Comparison with LRU

Last Used Timestamp

- 4 → 8
- 1 → 7
- 0 → 5 [Least Recently Used] [Evict]
- 3 → 6

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2	2	4	4					
			1	1	1	1	1	1	1					
		0	0	0	0	0	0	0	2					
	7	7	7	7	7	3	3	3	3					
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Miss					

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	4	4	4				
			1	1	1	1	1	1	1	1				
		0	0	0	0	0	0	0	2	2				
	7	7	7	7	7	3	3	3	3	3				
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Miss	Hit				

Cache Space

Comparison with LRU

Last Used Timestamp

- 4 → 8
- 1 → 7 [Least Recently Used][EVICT]
- 2 → 9
- 3 → 10

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	4	4	4	4			
			1	1	1	1	1	1	1	1	0			
		0	0	0	0	0	0	0	2	2	2			
	7	7	7	7	7	3	3	3	3	3	3			
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Miss	Hit	Miss			

Cache Space

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2	2	4	4	4	4	4		
			1	1	1	1	1	1	1	1	0	0		
		0	0	0	0	0	0	0	2	2	2	2		
	7	7	7	7	7	3	3	3	3	3	3	3		
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Miss	Hit	Miss	Hit		

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	4	4	4	4	4	4	
			1	1	1	1	1	1	1	1	0	0	0	
		0	0	0	0	0	0	0	2	2	2	2	2	
	7	7	7	7	7	3	3	3	3	3	3	3	3	
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Miss	Hit	Miss	Hit	Hit	

Cache Space

Comparison with LRU

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

				2	2	2	2	4	4	4	4	4	4	4
			1	1	1	1	1	1	1	1	0	0	0	0
		0	0	0	0	0	0	0	2	2	2	2	2	2
	7	7	7	7	7	3	3	3	3	3	3	3	3	3
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Miss	Hit	Miss	Hit	Hit	Hit

Cache Space

Comparison with LRU

Total Page Faults = 8

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2	2	4	4	4	4	4	4	4
			1	1	1	1	1	1	1	1	0	0	0	0
		0	0	0	0	0	0	0	2	2	2	2	2	2
	7	7	7	7	7	3	3	3	3	3	3	3	3	3
Hit or Miss?	**Miss**	**Miss**	**Miss**	**Miss**	Hit	**Miss**	Hit	**Miss**	**Miss**	Hit	**Miss**	Hit	Hit	Hit

Belady's MIN example

Total Page Faults = 6

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Page Reference	7	0	1	2	0	3	1	4	2	3	0	3	2	3

Cache Space				2	2	2	2	2	2	2	2	2	2	2
			1	1	1	1	1	4	4	4	4	4	4	4
		0	0	0	0	0	0	0	0	0	0	0	0	0
	7	7	7	7	7	3	3	3	3	3	3	3	3	3
Hit or Miss?	**Miss**	**Miss**	**Miss**	**Miss**	Hit	**Miss**	Hit	**Miss**	Hit	Hit	Hit	Hit	Hit	Hit

SHADE Magic!

Total sample misses = 4

Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Sample ID	~~7~~ 3	0	1	2	0	3	1	4 0	2	3	0	3	2	3

Cache Space				2	2	2	2	2	2	2	2	2	2	2
			1	1	1	1	1	4	4	4	4	4	4	4
		0	0	0	0	0	0	0	0	0	0	0	0	0
	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Hit or Miss?	**Miss**	**Miss**	**Miss**	**Miss**	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit

SHADE Magic!

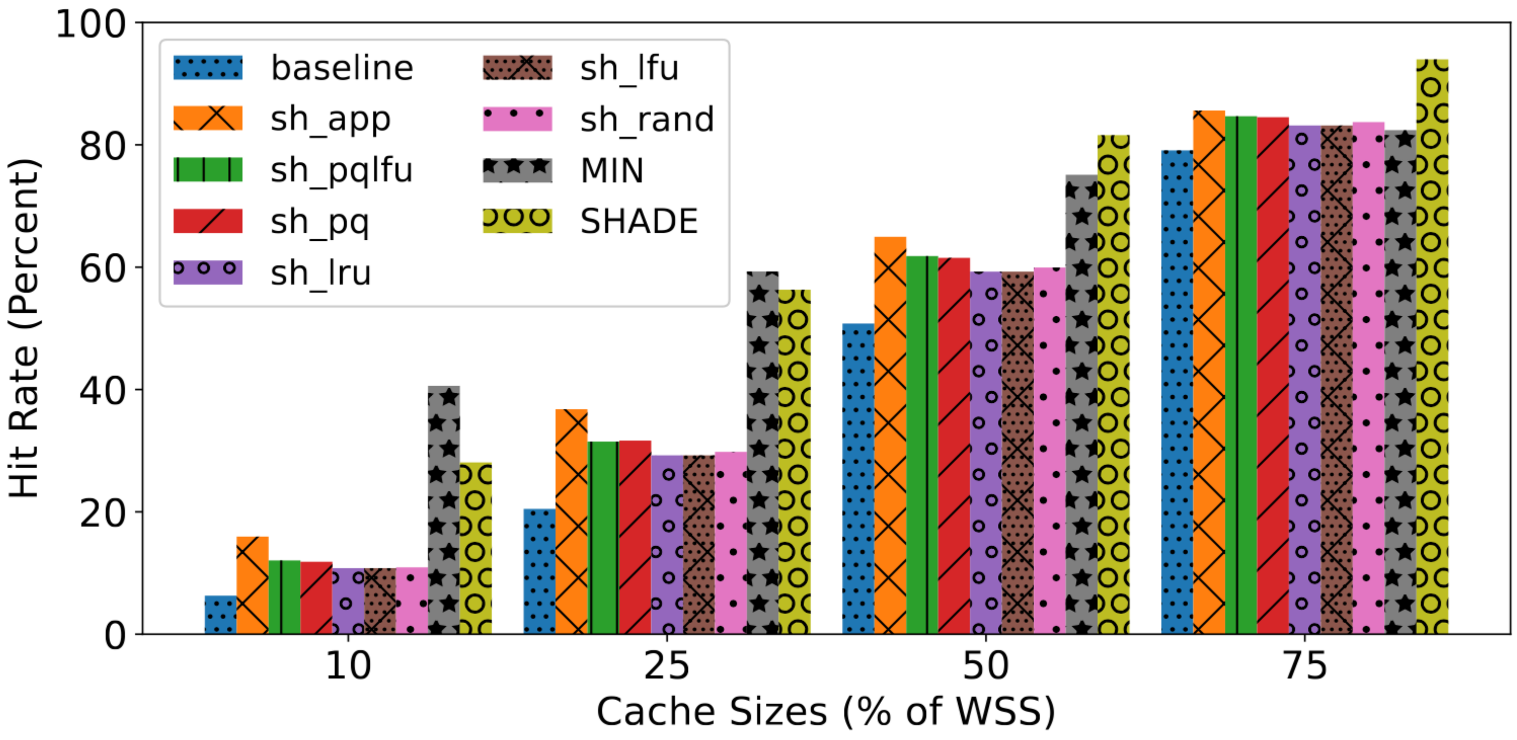
Total sample misses = 4

Belady's MIN knows the future, SHADE creates it.

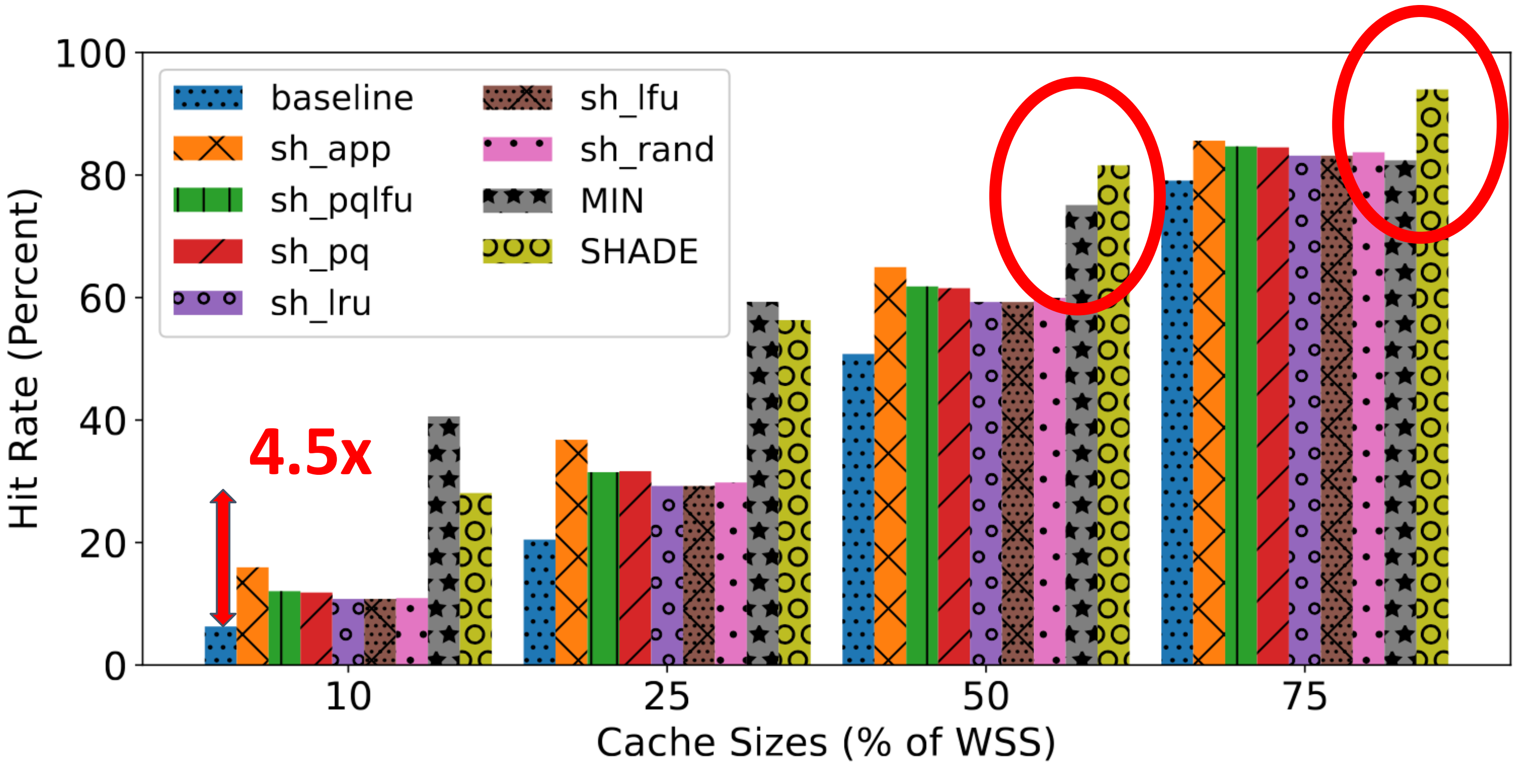
Time stamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Sample ID	~~7~~ 3	0	1	2	0	3	1	~~4~~ 0	2	3	0	3	2	3

Cache Space				2	2	2	2	2	2	2	2	2	2	2
			1	1	1	1	1	4	4	4	4	4	4	4
		0	0	0	0	0	0	0	0	0	0	0	0	0
	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Hit or Miss?	Miss	Miss	Miss	Miss	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit

Impact on Hit Ratio

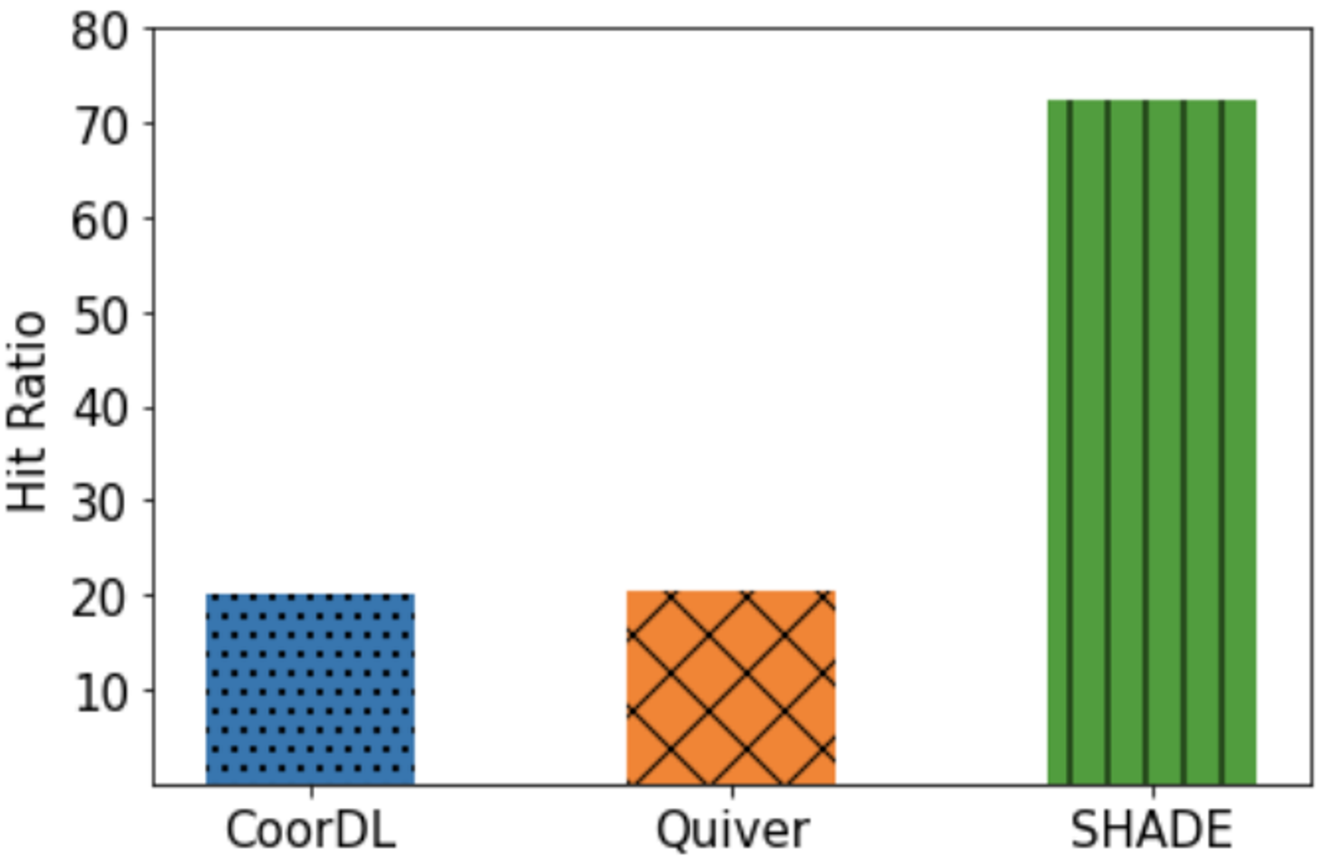


Impact on Hit Ratio



SHADE can achieve read hit ratio up to 4.5x compared to LRU and outperforms MIN in some cases

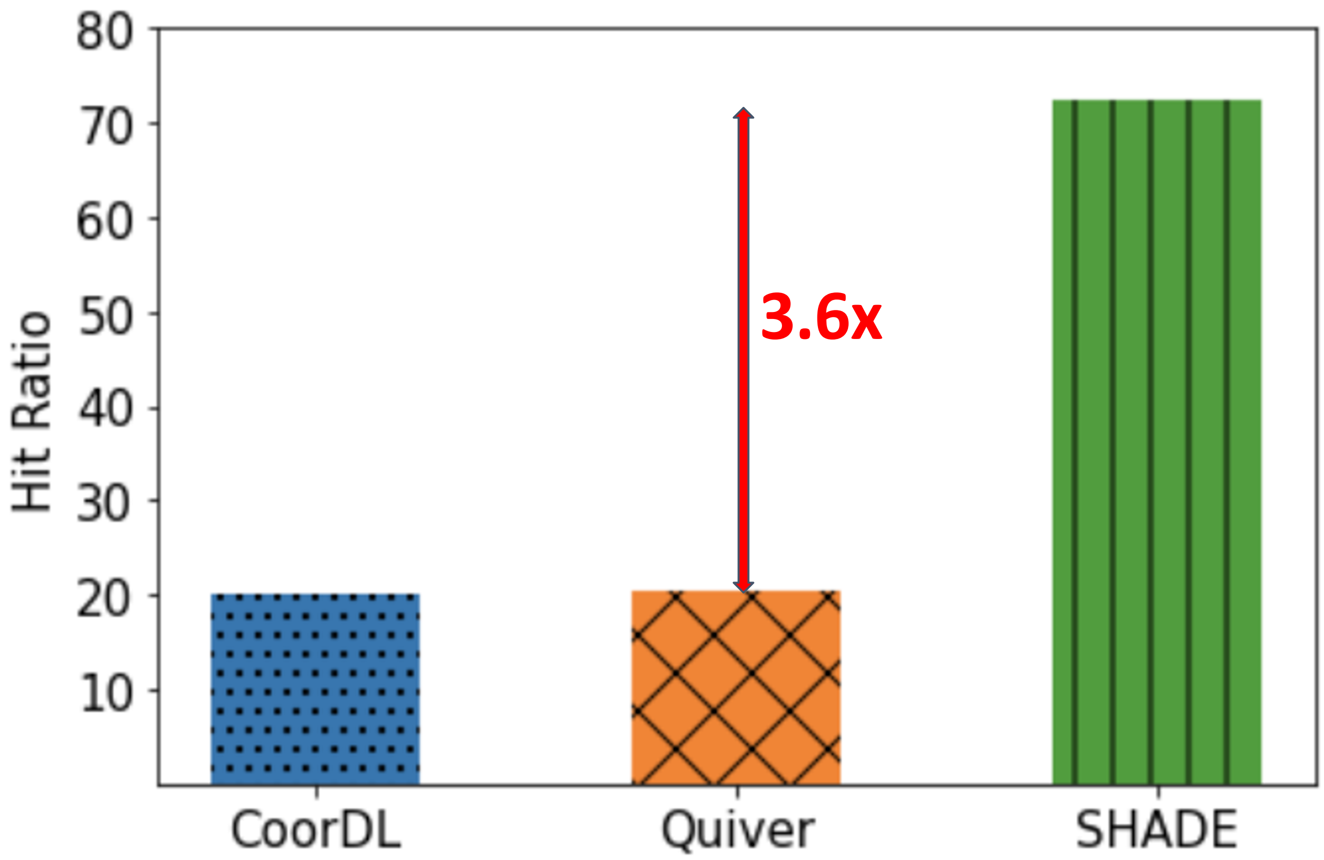
Impact on Hit Ratio



20% WSS of CIFAR-10 + ResNet-18

1. Abhishek Vijaya Kumar and Muthian Sivathanu. Quiver: An informed storage cache for deep learning. In 18th USENIX Conference on File and Storage Technologies (FAST 20), pages 283–296, 2020.
2. Jayashree Mohan, Amar Phanishayee, Ashish Raniwala, and Vijay Chidambaram. Analyzing and mitigating data stalls in dnn training. arXiv preprint arXiv:2007.06775, 2020

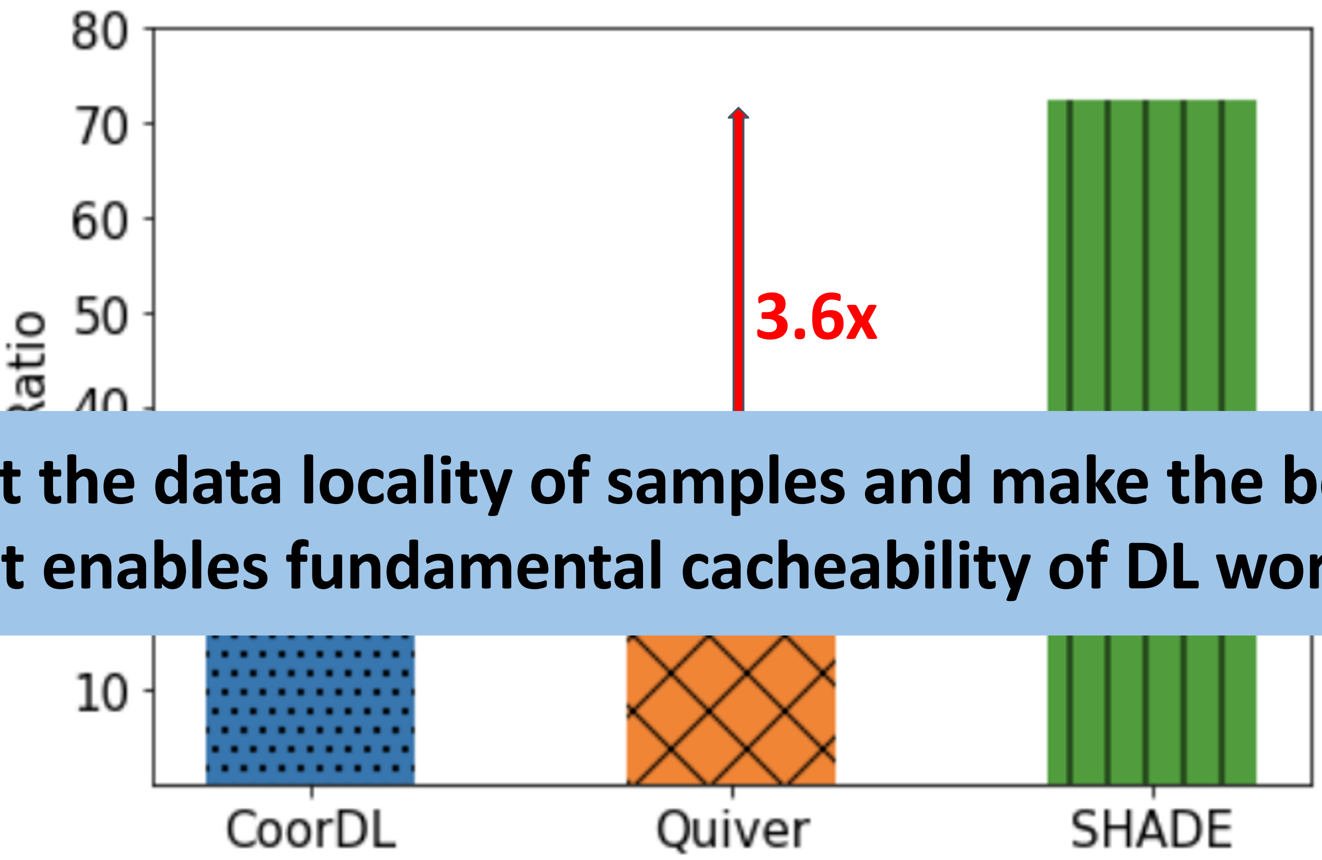
Higher Exploitation of Data Locality



20% WSS of CIFAR-10 + ResNet-18

1. Abhishek Vijaya Kumar and Muthian Sivathanu. Quiver: An informed storage cache for deep learning. In 18th USENIX Conference on File and Storage Technologies (FAST 20), pages 283–296, 2020.
2. Jayashree Mohan, Amar Phanishayee, Ashish Raniwala, and Vijay Chidambaram. Analyzing and mitigating data stalls in dnn training. arXiv preprint arXiv:2007.06775, 2020

Higher Exploitation of Data Locality

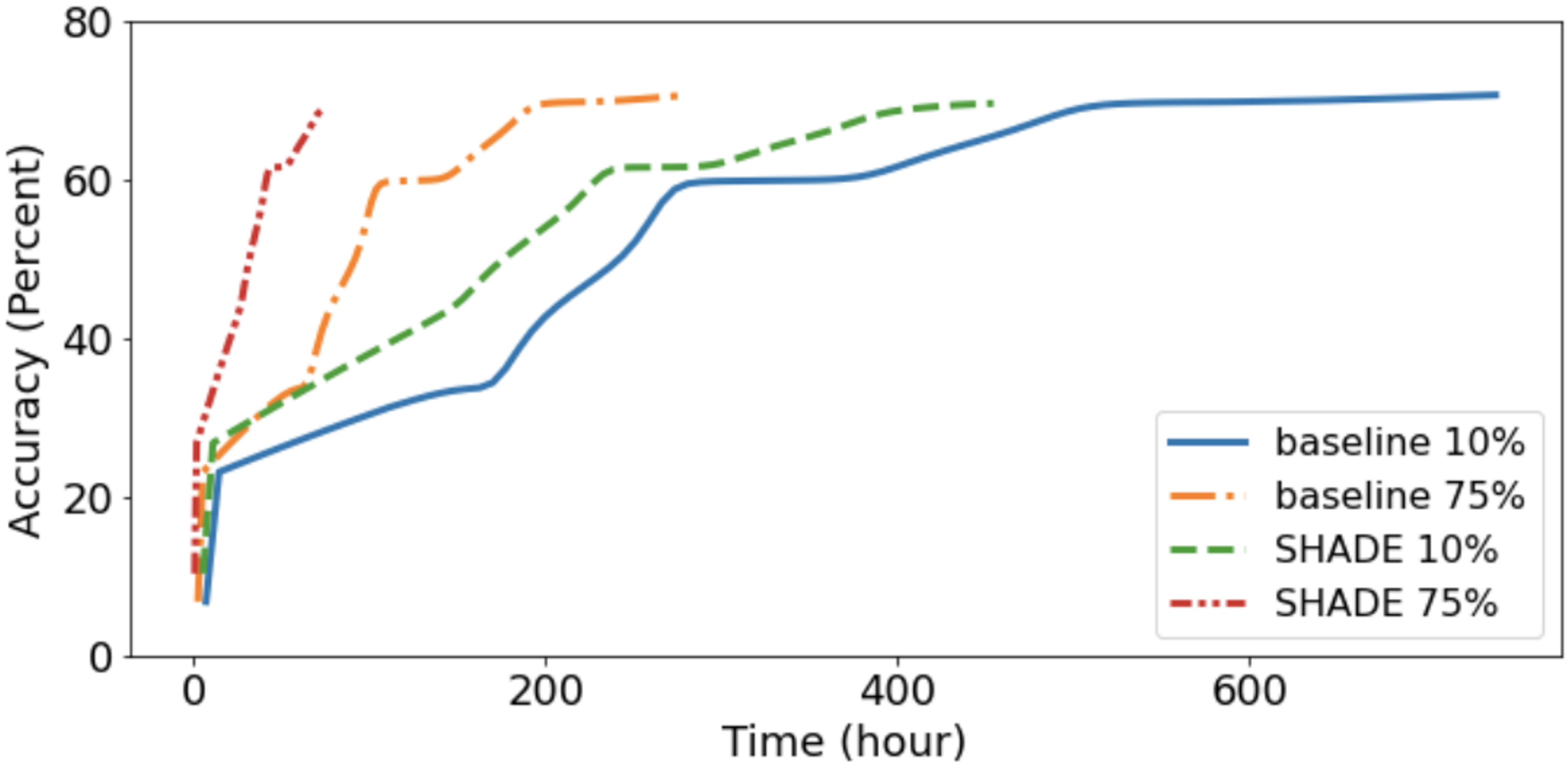


SHADE can exploit the data locality of samples and make the best utilization of a small cache, i.e., it enables fundamental cacheability of DL workloads.

20% WSS of CIFAR-10 + ResNet-18

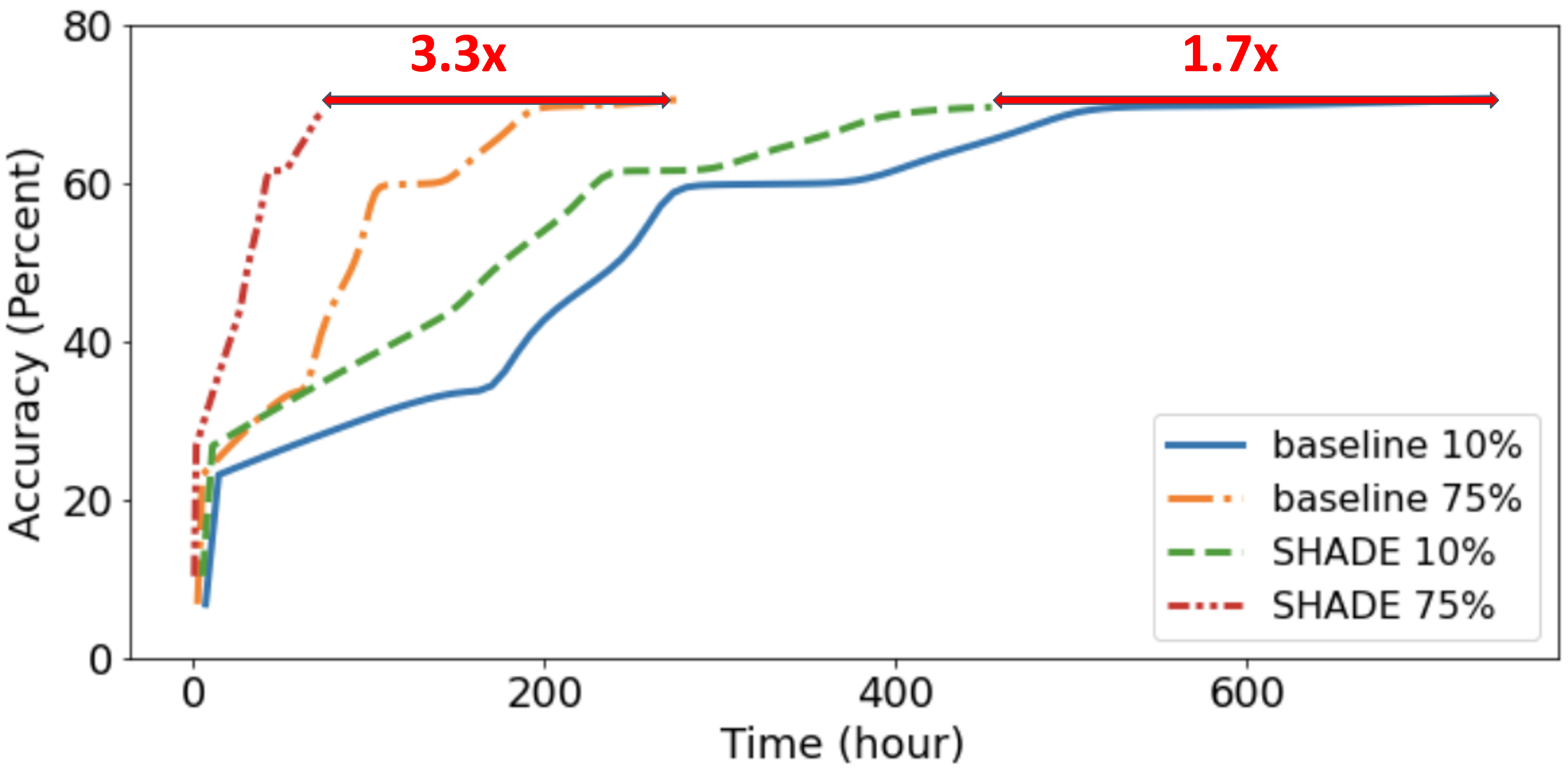
1. Abhishek Vijaya Kumar and Muthian Sivathanu. Quiver: An informed storage cache for deep learning. In 18th USENIX Conference on File and Storage Technologies (FAST 20), pages 283–296, 2020.
2. Jayashree Mohan, Amar Phanishayee, Ashish Raniwala, and Vijay Chidambaram. Analyzing and mitigating data stalls in dnn training. arXiv preprint arXiv:2007.06775, 2020

Accuracy vs. Time



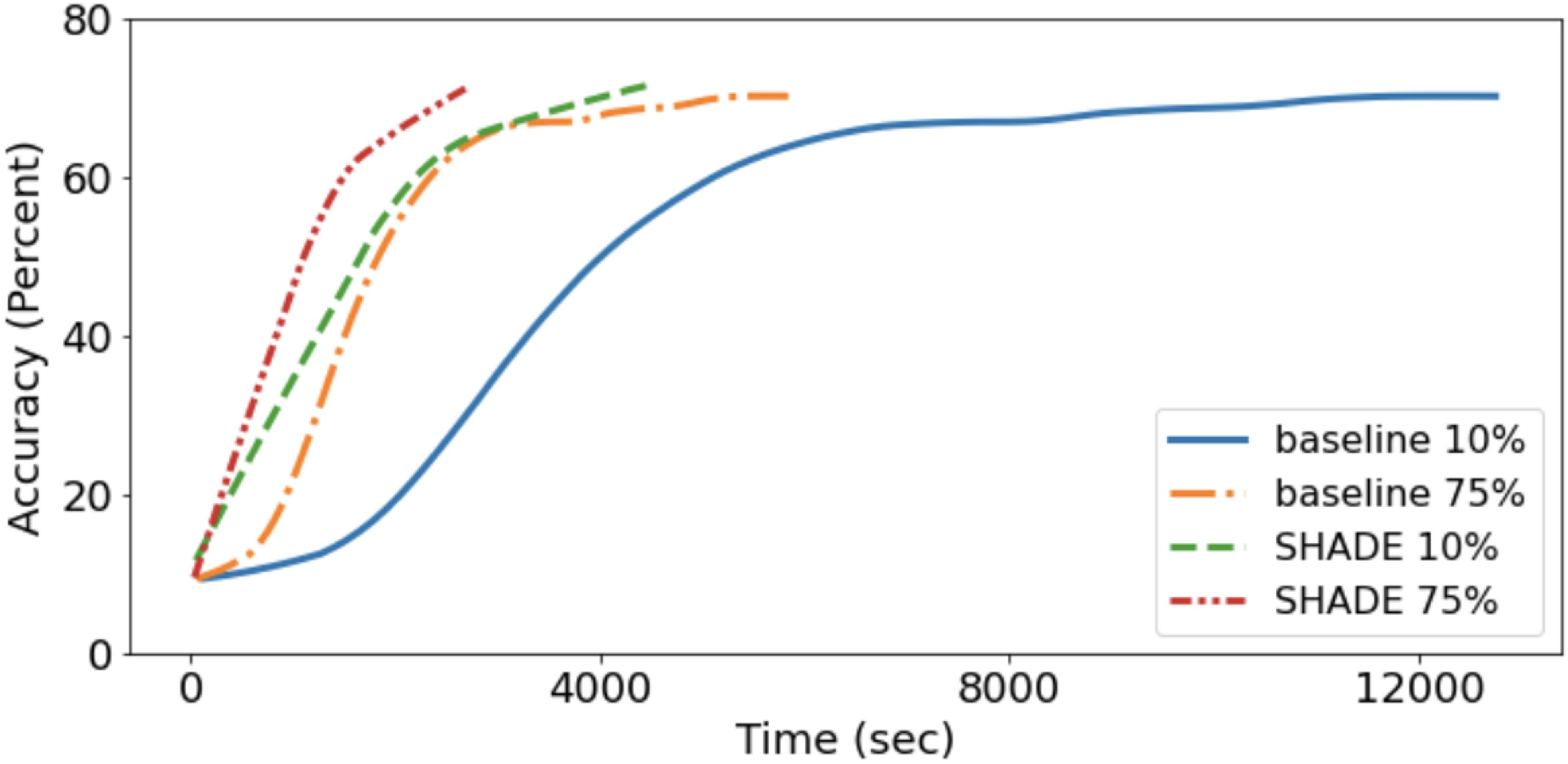
ResNet-50 model + ImageNet Dataset

Faster Accuracy Convergence



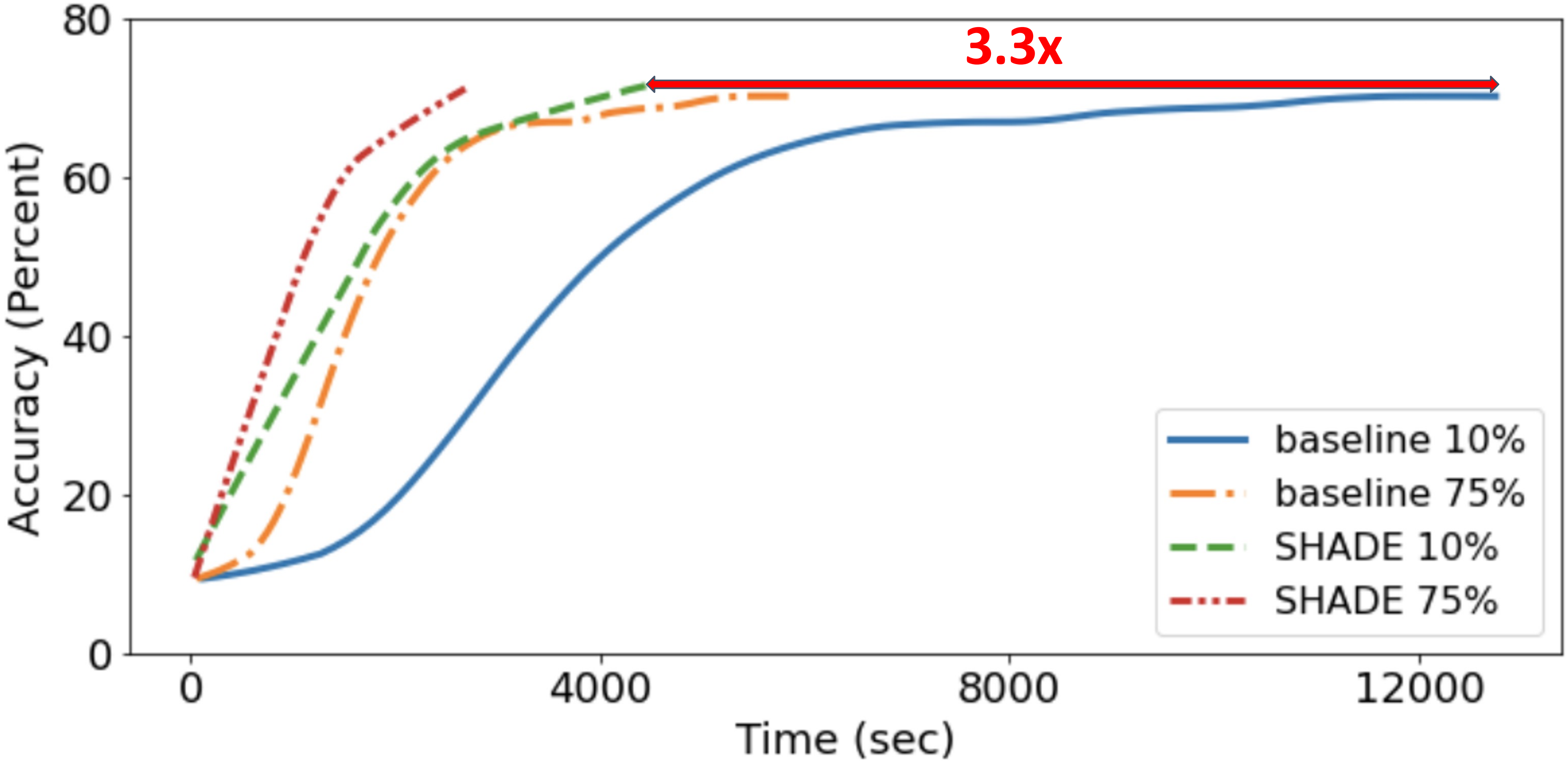
ResNet-50 model + ImageNet Dataset

Accuracy vs. Time



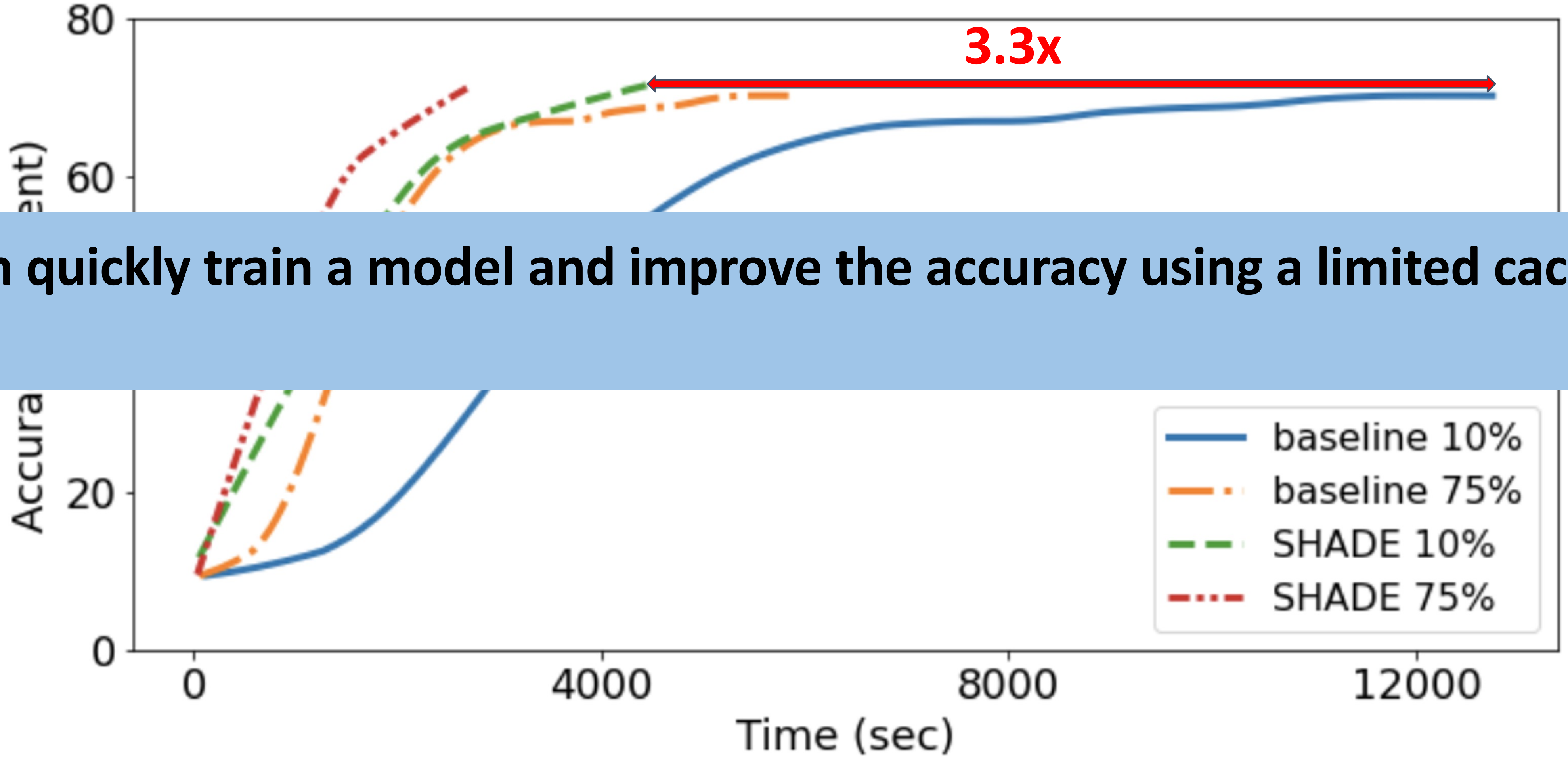
AlexNet model + CIFAR-10 Dataset

Faster Accuracy Convergence



AlexNet model + CIFAR-10 Dataset

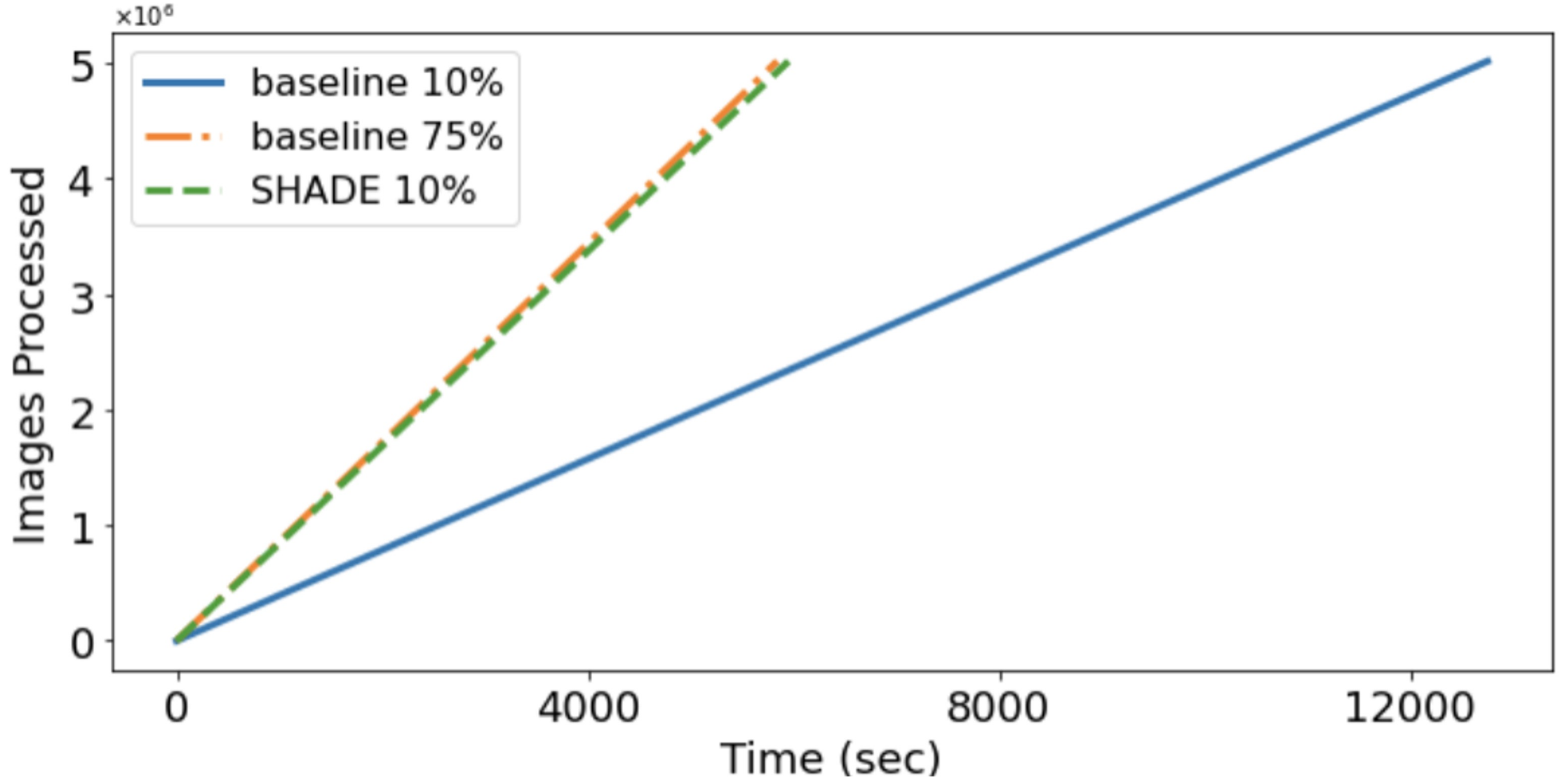
Faster Accuracy Convergence



SHADE can quickly train a model and improve the accuracy using a limited cache space.

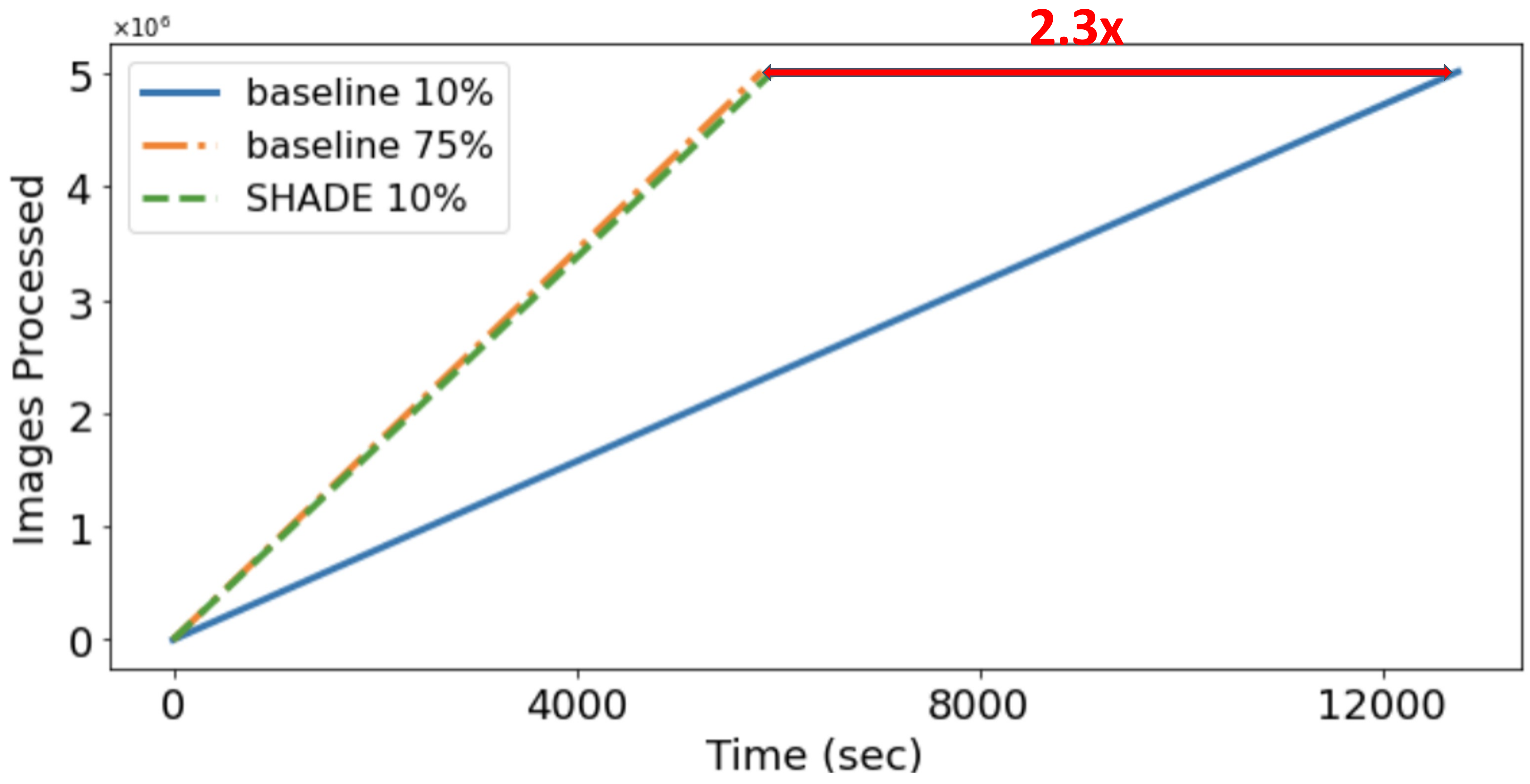
AlexNet model + CIFAR-10 Dataset

Impact on Throughput



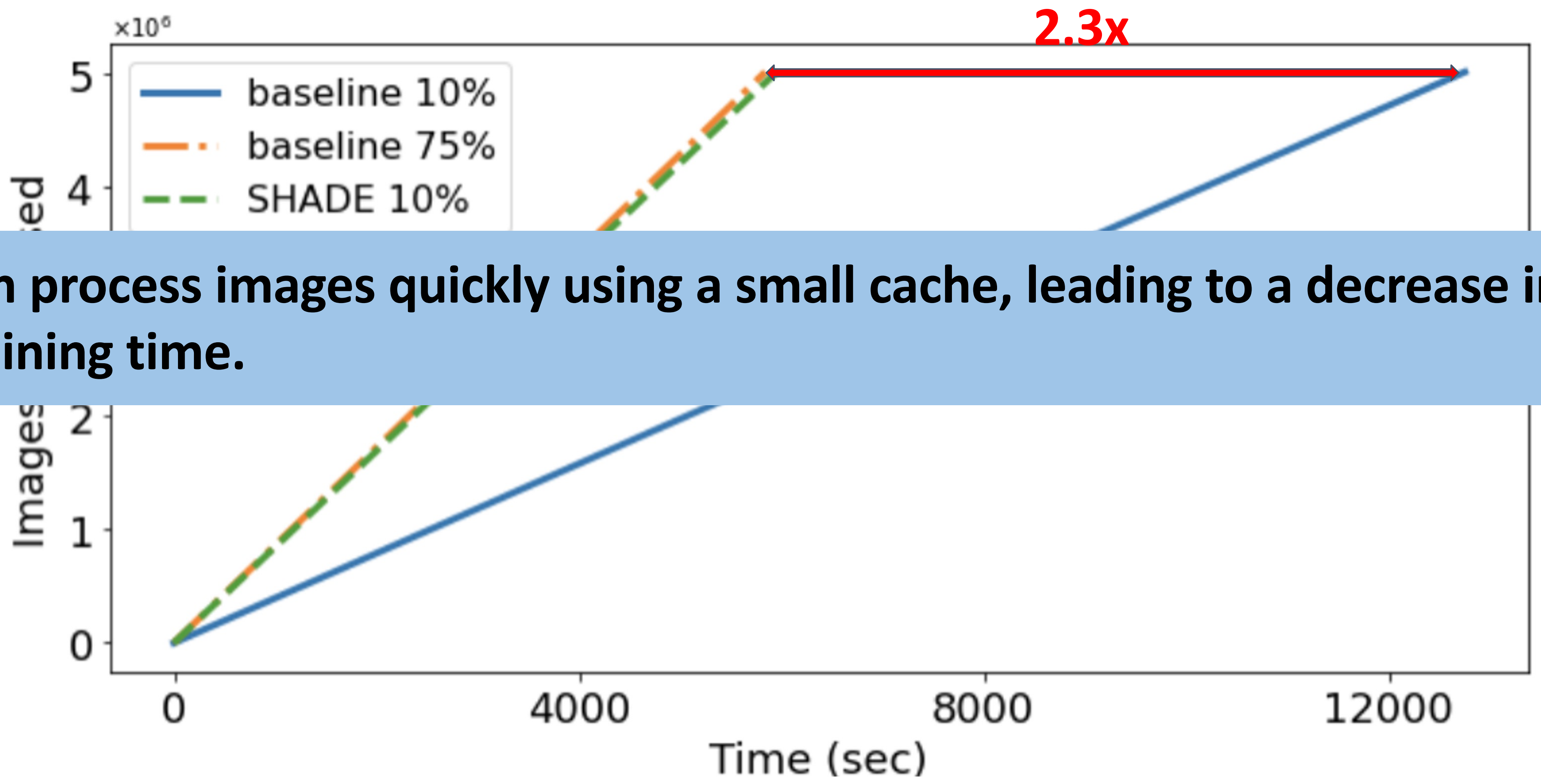
AlexNet model + CIFAR-10 Dataset

Higher Throughput



AlexNet model + CIFAR-10 Dataset

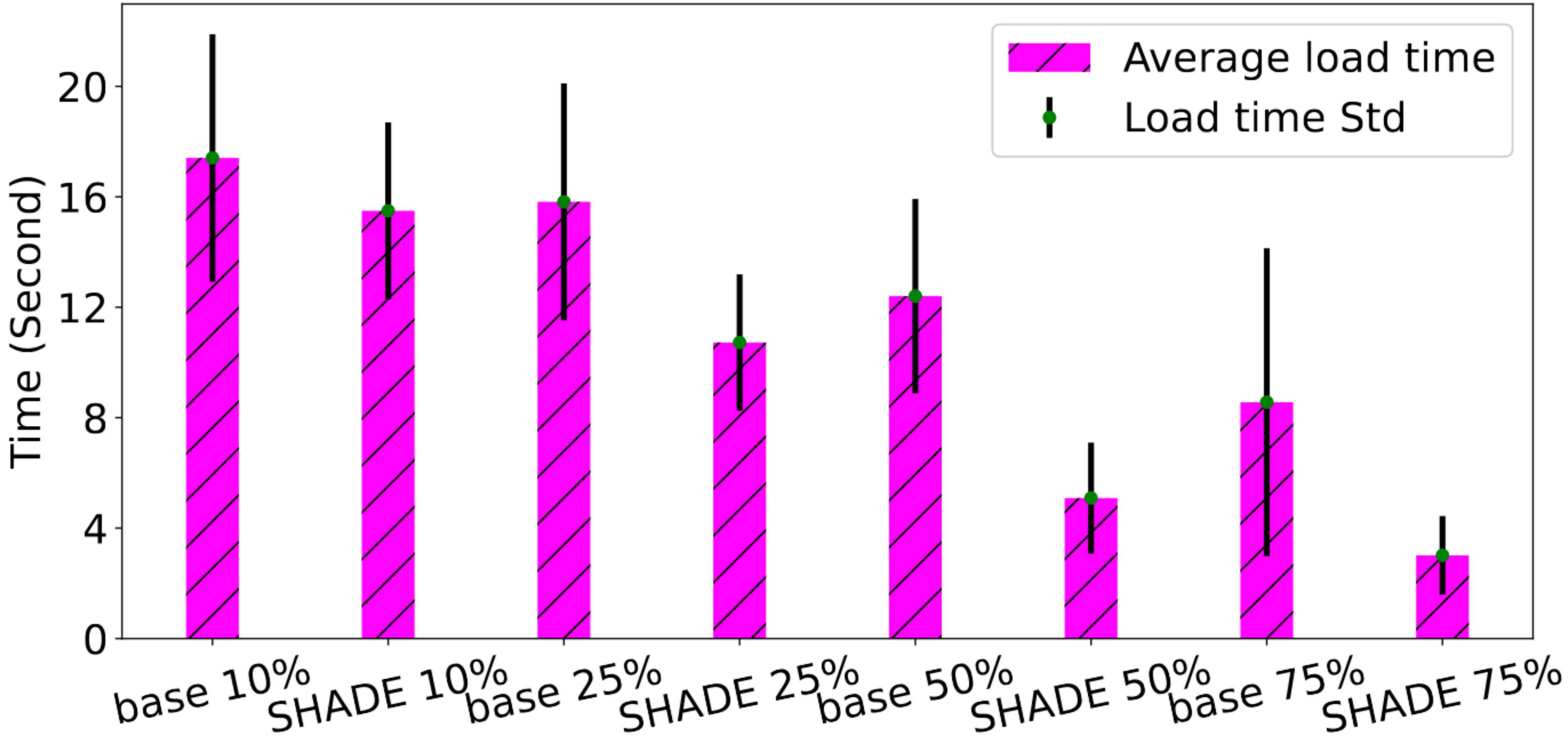
Higher Throughput



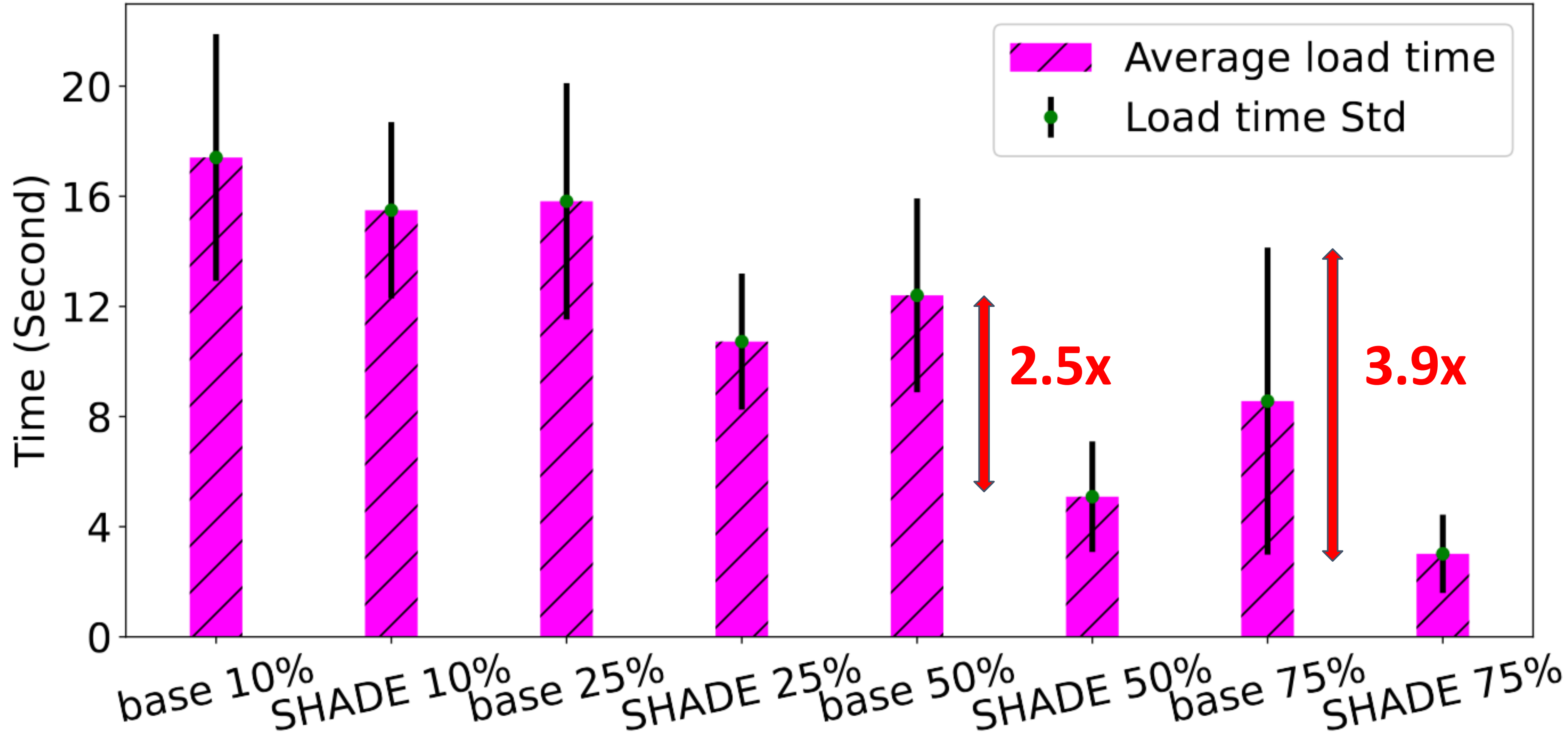
SHADE can process images quickly using a small cache, leading to a decrease in overall training time.

AlexNet model + CIFAR-10 Dataset

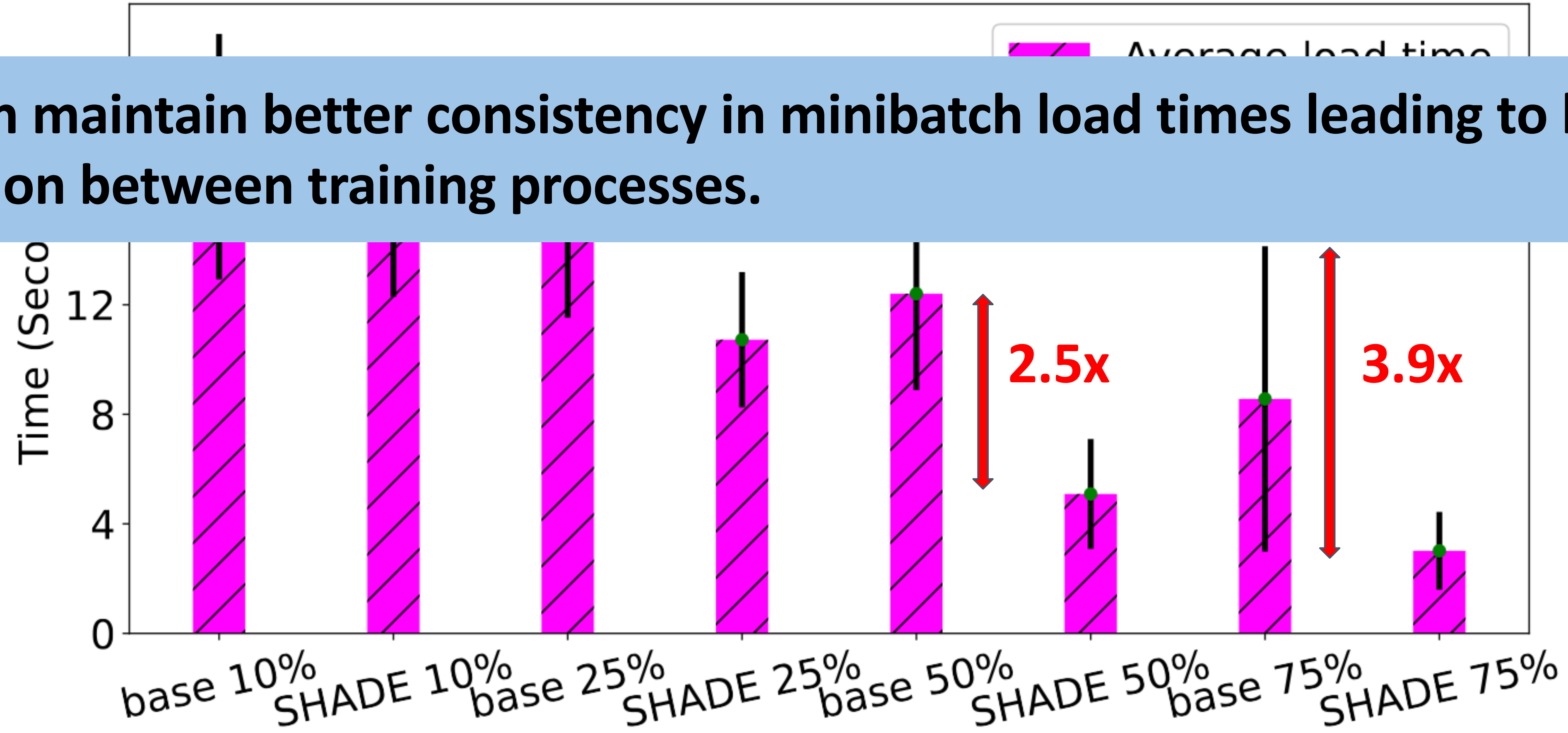
Minibatch Load Time



Minibatch Load Time



Minibatch Load Time

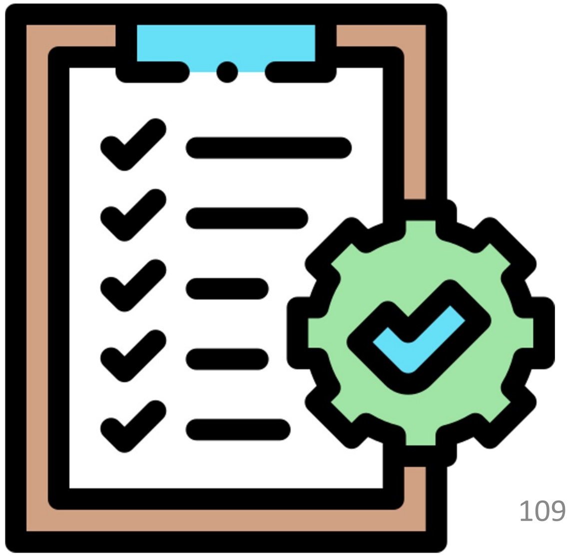


SHADE can maintain better consistency in minibatch load times leading to better coordination between training processes.

Summary

SHADE is a caching system that **exploits data and system characteristics** in DLT.

- Provides the ability to train more on hard-to-learn samples (Ranking fine-grained importance *+ PADS policy*)
- Retains the most important samples in cache (*APP* Cache)
- Increases hit rate in cache leading to faster accuracy convergence (~3.3x) and increased throughput (~2.3x)
- Enables fundamental cacheability of DL Training outperforming optimal Belady's MIN policy in DLT context



SHADE is available at https://github.com/R-I-S-Khan/SHADE

Shoot me an email at redwan@vt.edu

Check out my recent works at https://r-i-s-khan.github.io