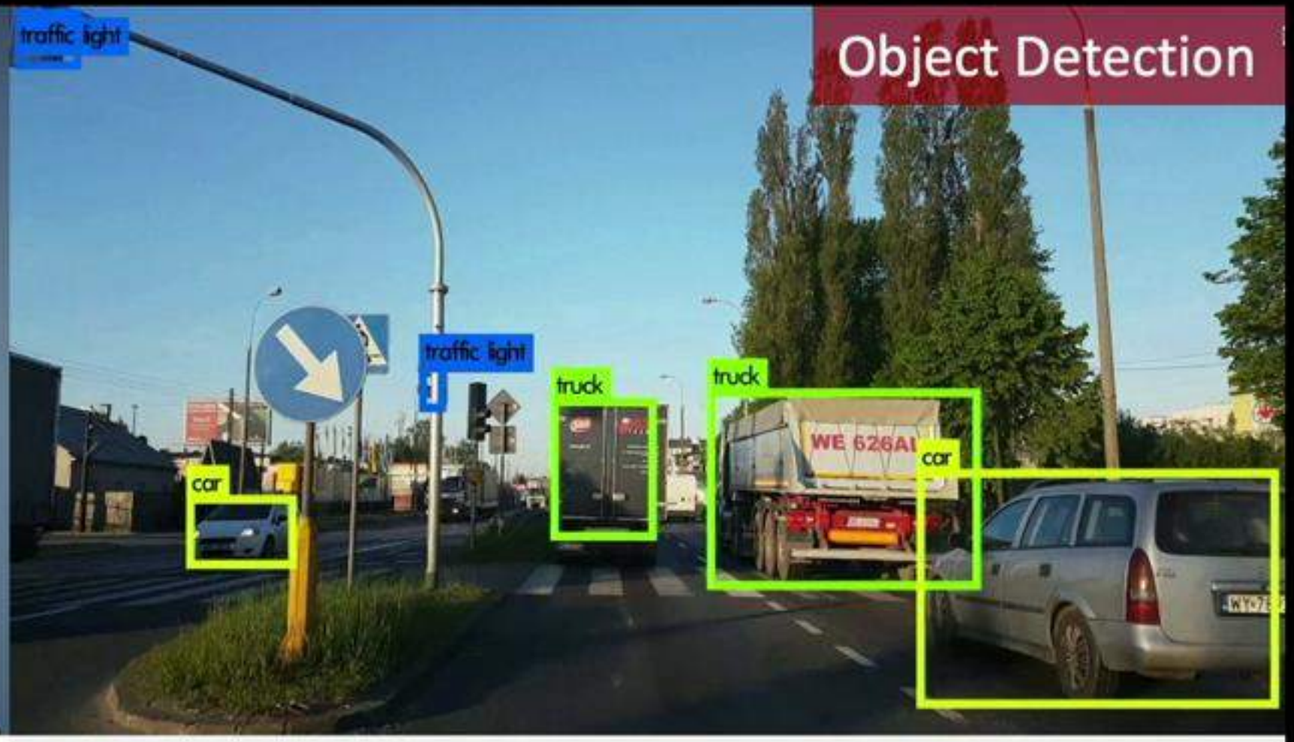
Machine Learning Systems for Highly Distributed and Rapidly Growing Data

Kevin Hsieh

March 28, 2019

Carnegie Mellon University

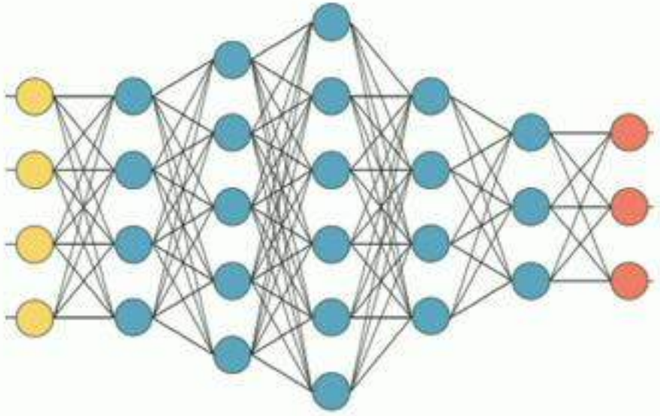

Face Recognition


Object Detection
traffic light
traffic light
truck
truck
car
car


Self-driving car


こんにちは!
Ciao
Hi!
Language translation

At Their Core: ML Training and Serving



At Their Core: ML Training and Serving

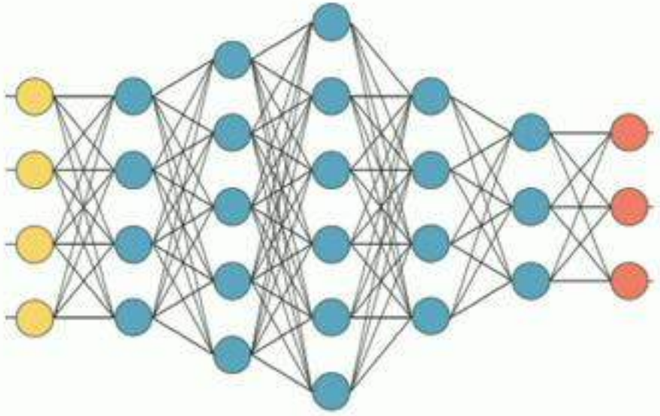


Image classification

At Their Core: ML Training and Serving

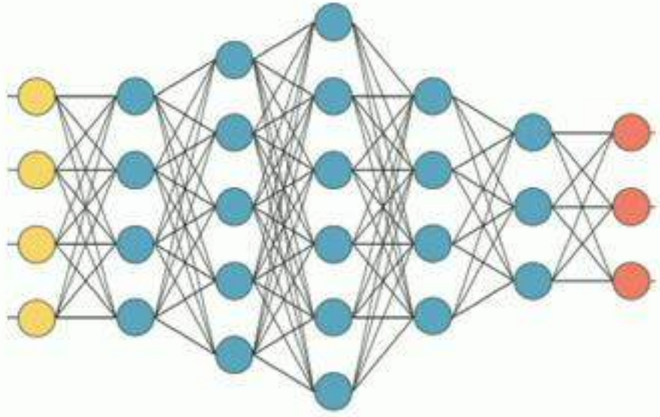


Image classification

Key Objectives: Low Latency and Low Cost

Training Data

Serving Data

ML on Real-World, Large-Scale Data

ML on Real-World, Large-Scale Data

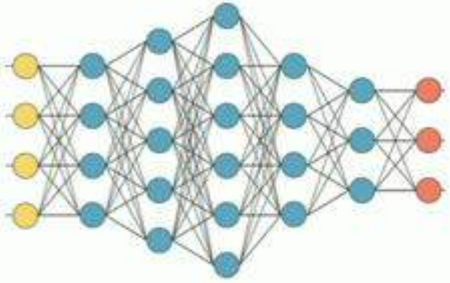
Many ML Data are

Highly Distributed and Rapidly Growing

Challenge #1: ML Serving on Rapidly Growing Data

Traffic cameras in a city

ML Serving System



Find video frames with trucks

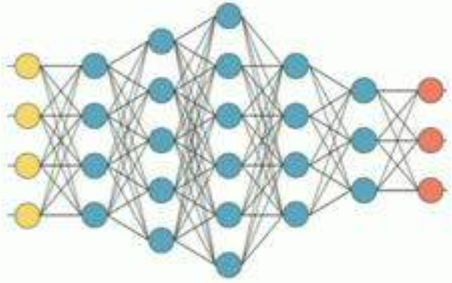
Challenge #1: ML Serving on Rapidly Growing Data

High Cost to Process Data at Ingest Time

Long Latency to Process Data at Query Time

Traffic cameras in a city

ML Serving System

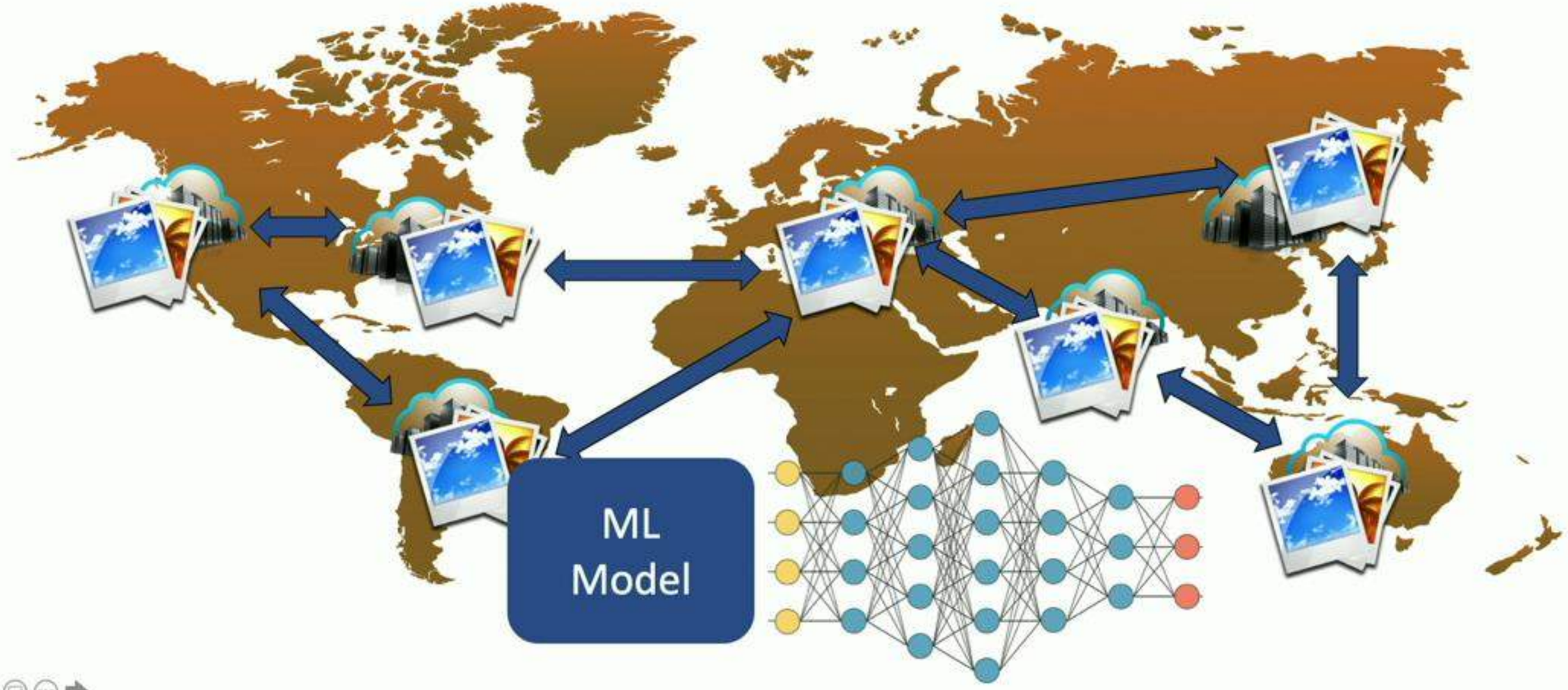


Find video frames with trucks

Challenge #2: ML Training on Highly Distributed Data

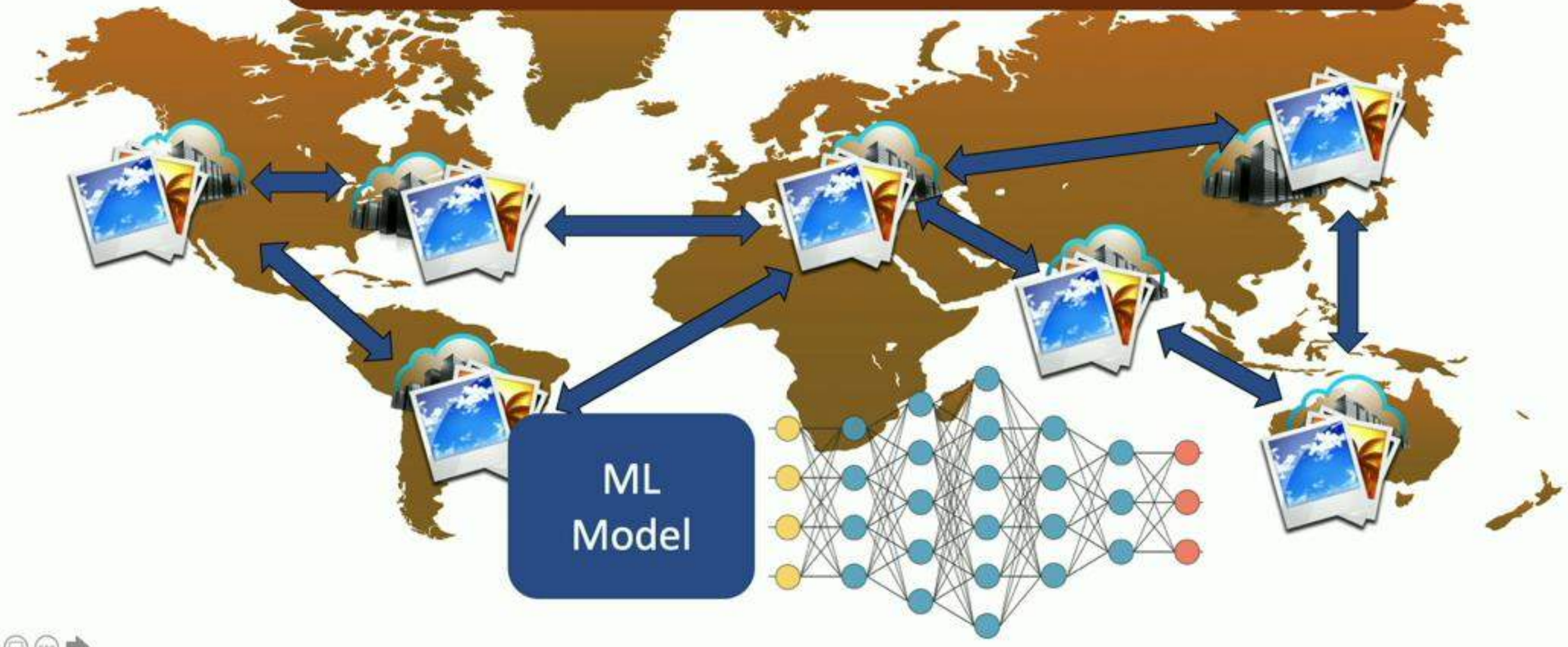
Challenge #2: ML Training on Highly Distributed Data

Challenge #2: ML Training on Highly Distributed Data



Challenge #2: ML Training on Highly Distributed Data

High Cost and Long Latency to
Train ML Models over Geo-Distributed Data



In This Talk

Focus: ML Serving for Rapidly Growing Data [OSDI'18]

Gaia: ML Training for Geo-Distributed Data [NSDI'17]

On-going and Future Work

In This Talk

Focus: ML Serving for Rapidly Growing Data [OSDI'18]

Gaia: ML Training for Geo-Distributed Data [NSDI'17]

On-going and Future Work

Video Data are Rapidly Growing

Massive video recordings are happening everywhere

Querying Objects in Videos using ML Serving

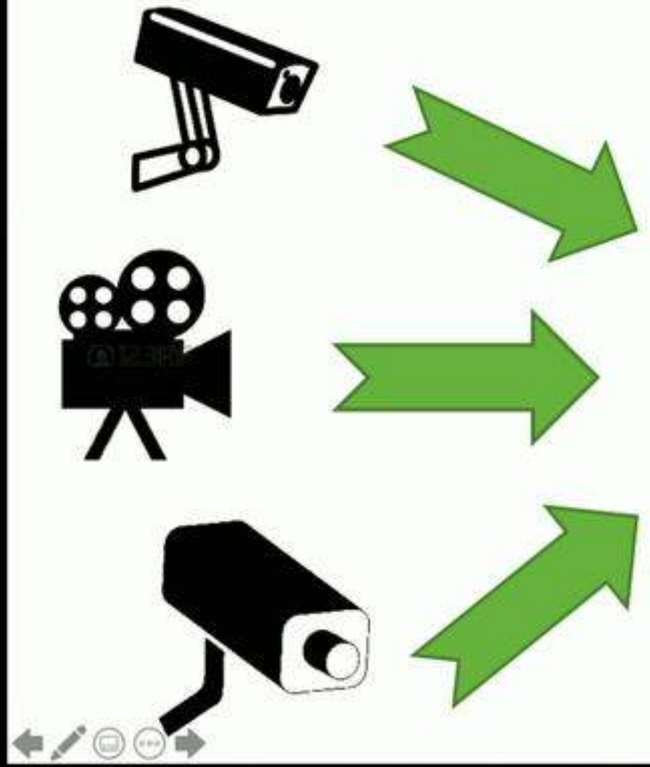
- Find all trucks among traffic videos in a city last week
- Find all people in garage videos in a company last night

→ *Query execution requires running detector & classifier CNNs*

→ *It is slow and costly on massive videos*

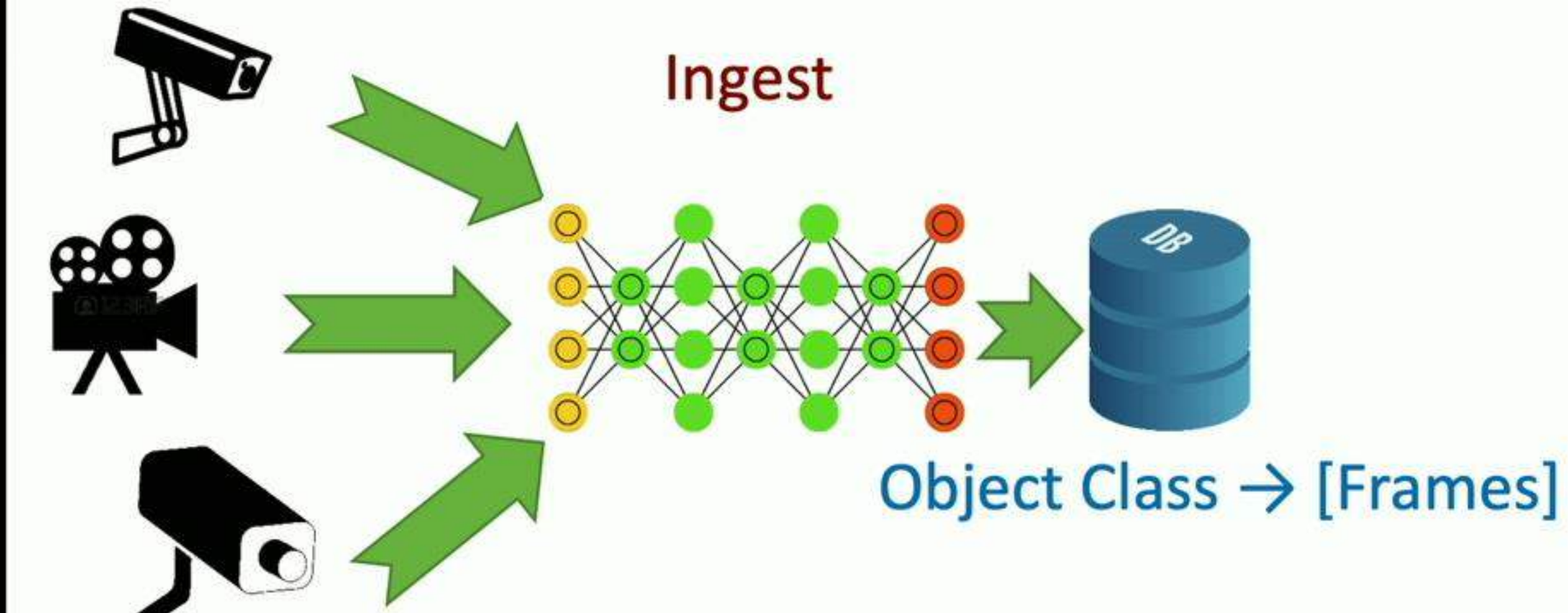
Ingest Time Analysis: Too Costly

- Analyzing live videos at ingest time can make query fast



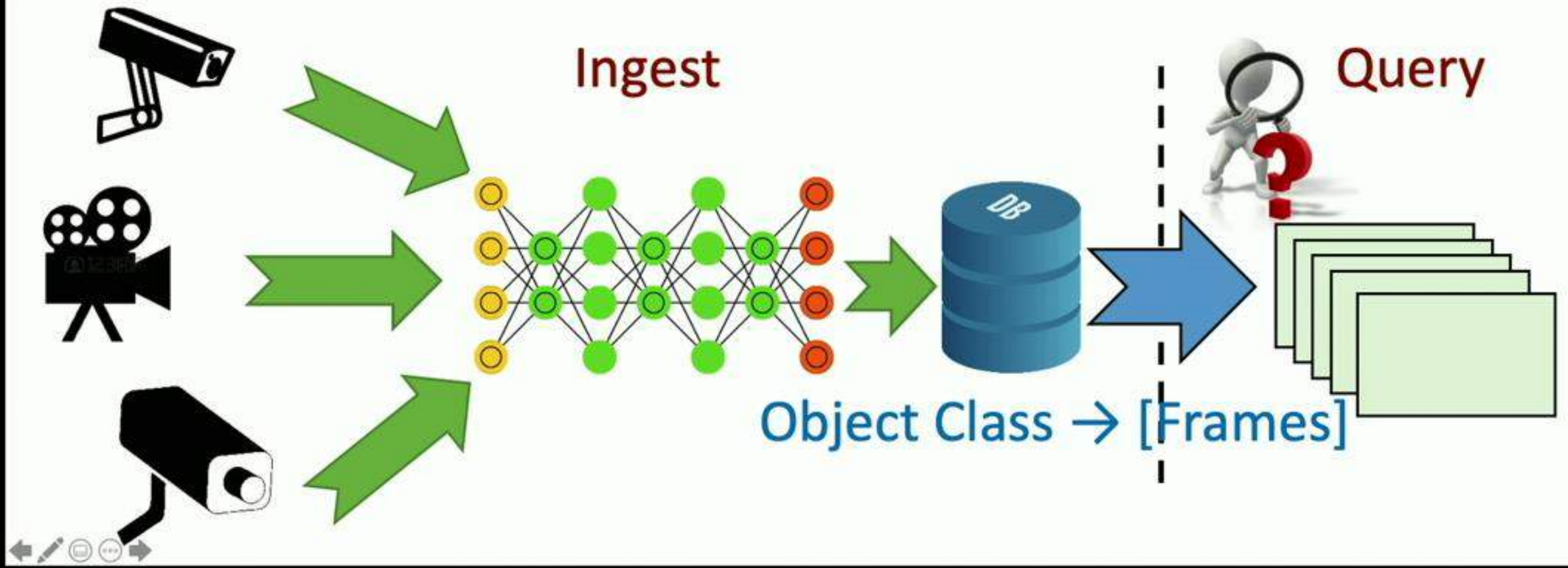
Ingest Time Analysis: Too Costly

- Analyzing live videos at ingest time can make query fast



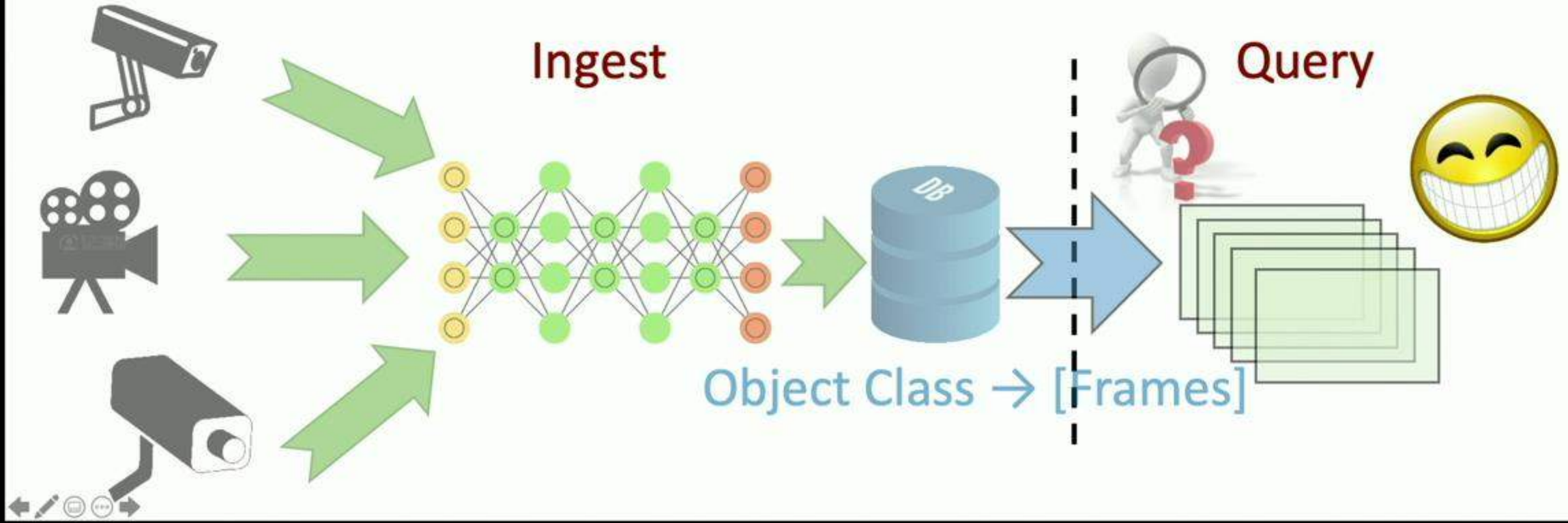
Ingest Time Analysis: Too Costly

- Analyzing live videos at ingest time can make query fast



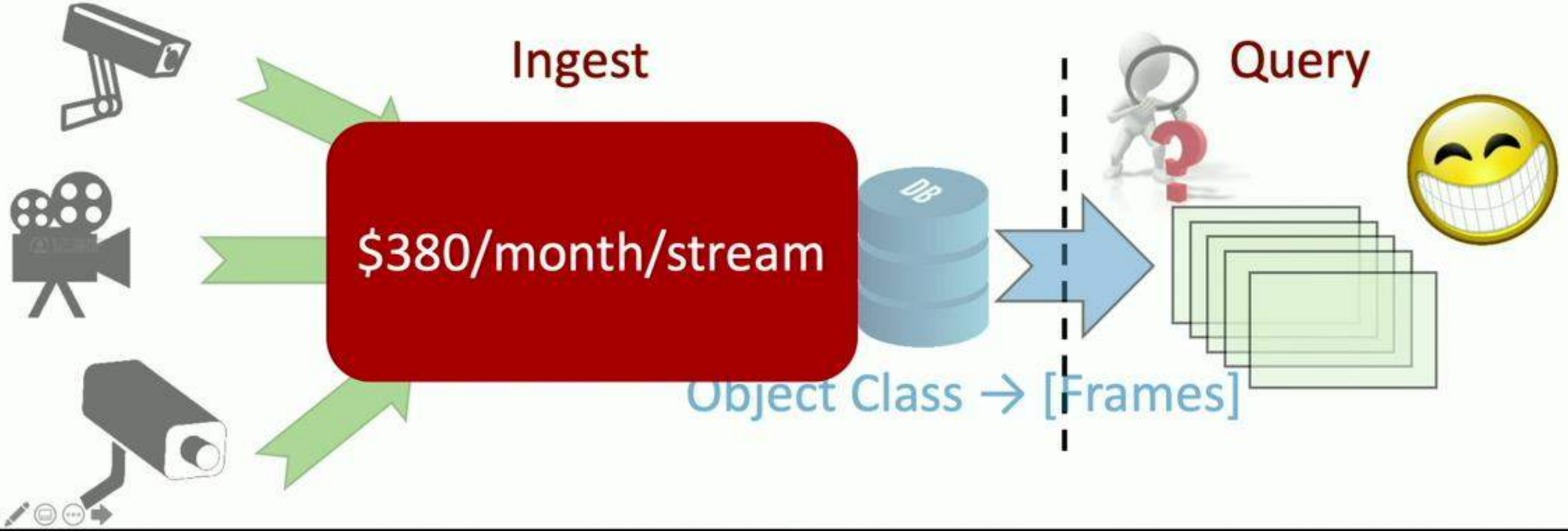
Ingest Time Analysis: Too Costly

- Analyzing live videos at ingest time can make query fast



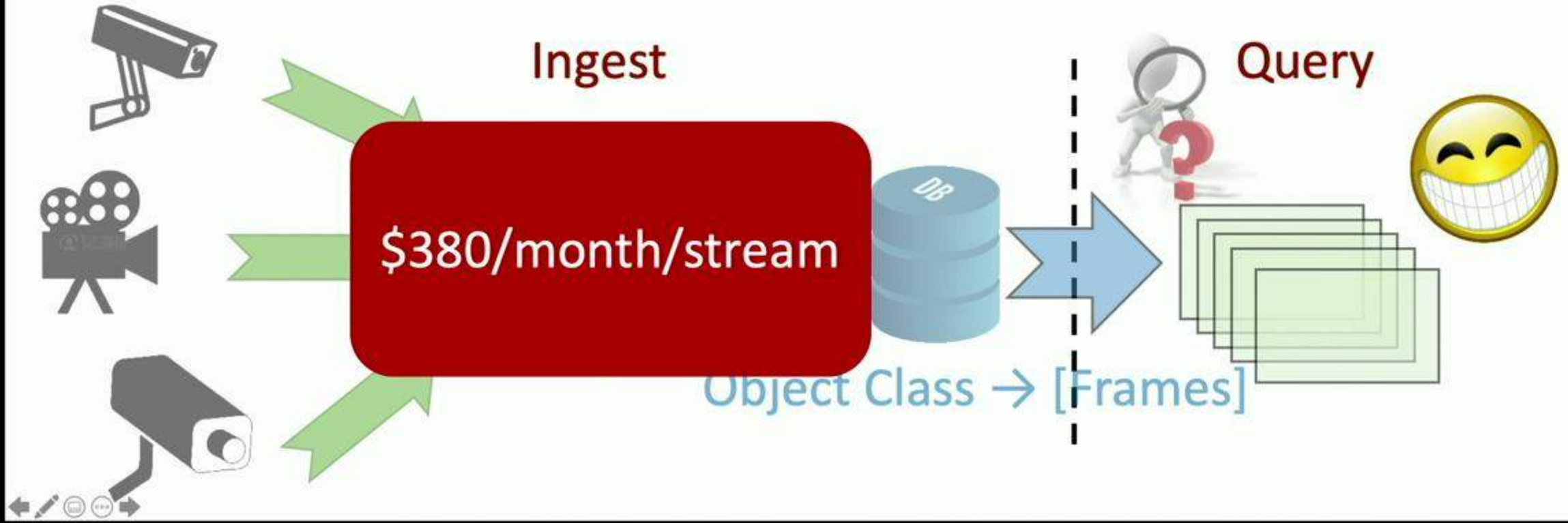
Ingest Time Analysis: Too Costly

- Analyzing live videos at ingest time can make query fast
  - But it is costly



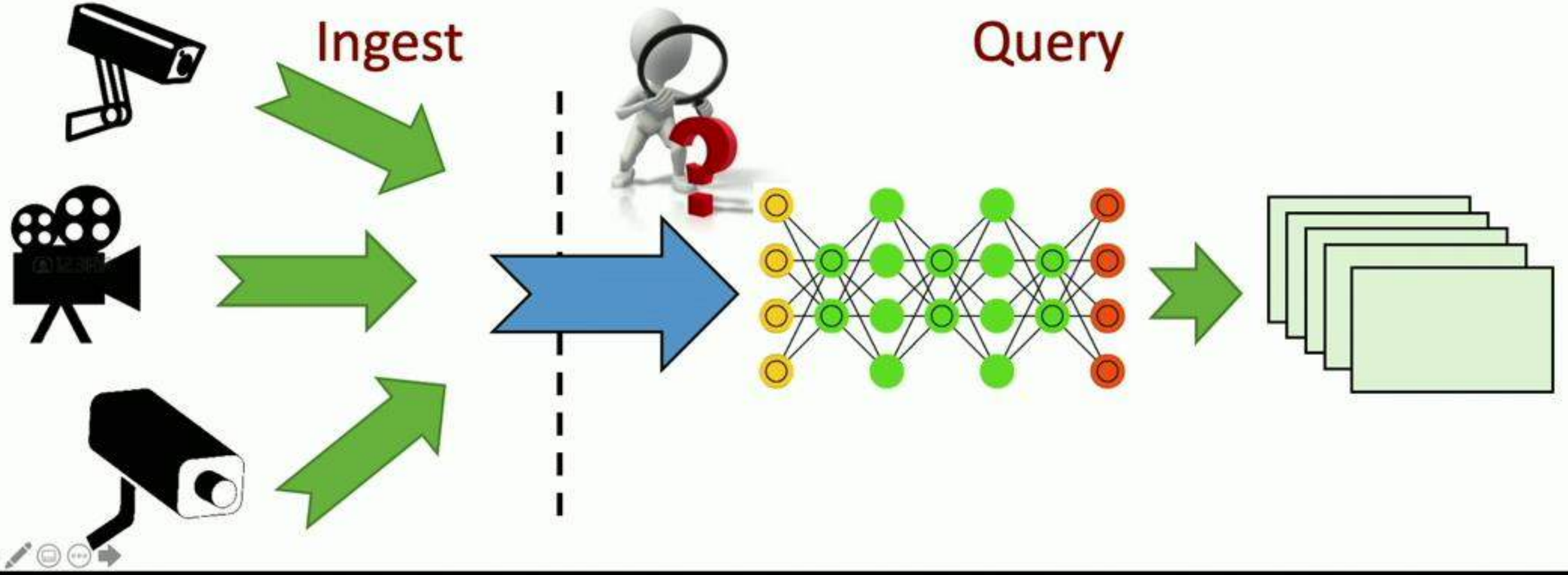
Ingest Time Analysis: Too Costly

- Analyzing live videos at ingest time can make query fast
  - But it is costly
  - Potentially wasteful (ingest all garage cameras vs. query one)



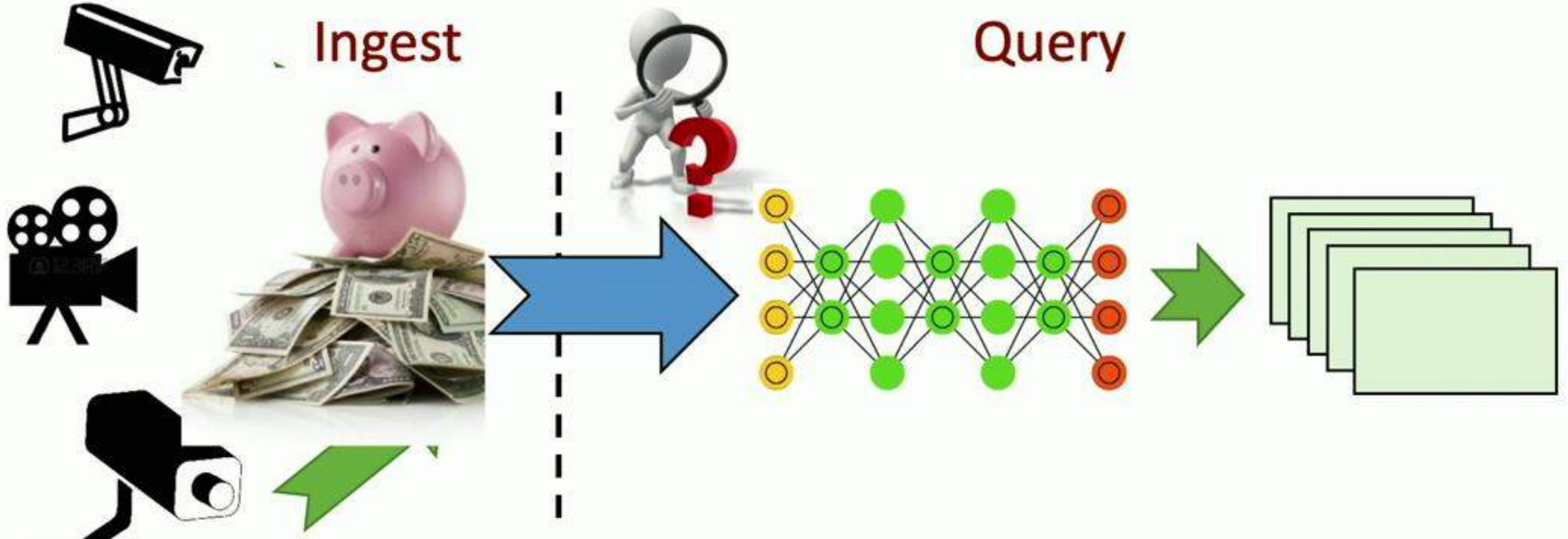
Query Time Analysis: Too Slow

- Analyzing videos at query time can save cost



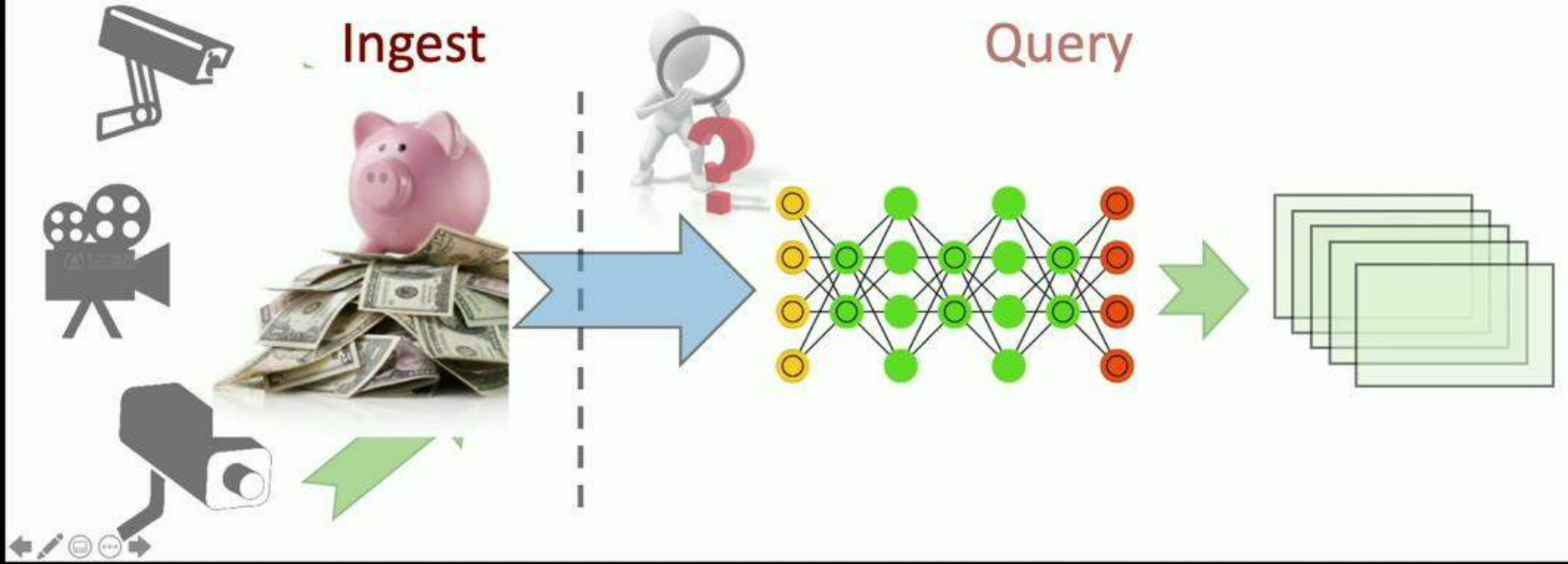
Query Time Analysis: Too Slow

- Analyzing videos at query time can save cost



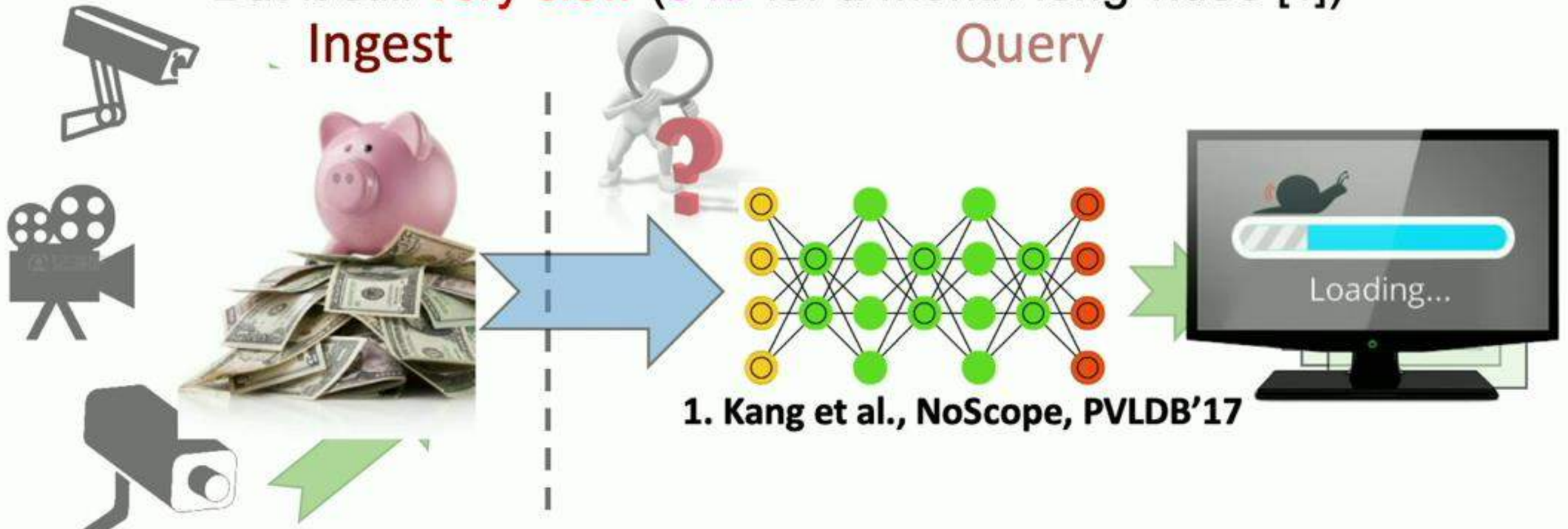
Query Time Analysis: Too Slow

- Analyzing videos at query time can save cost
  - Frame down-sampling / skipping
  - CNN specialization / cascading



Query Time Analysis: Too Slow

- Analyzing videos at query time can save cost
  - Frame down-sampling / skipping
  - CNN specialization / cascading
  - But it still very slow (5 hr for a month-long video [1])



1. Kang et al., NoScope, PVLDB'17

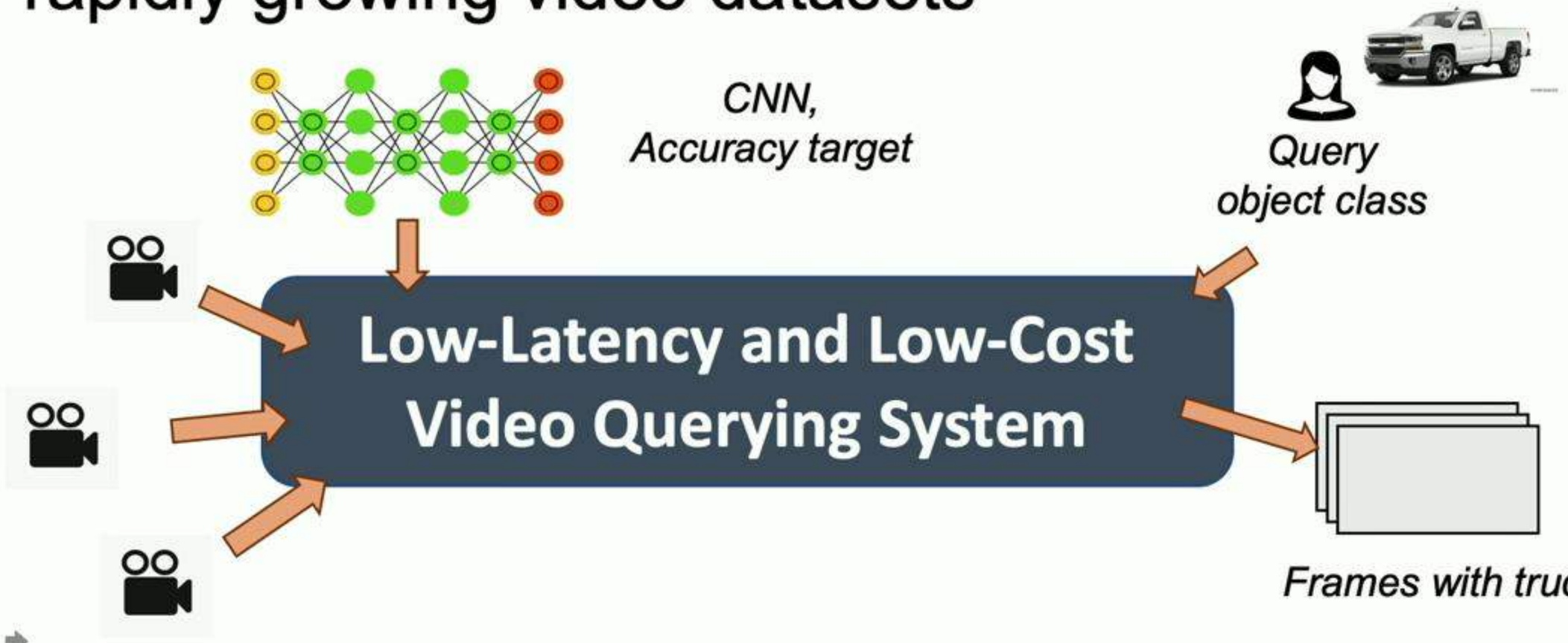
Our Goal

Enable low-latency and low-cost querying over rapidly growing video datasets

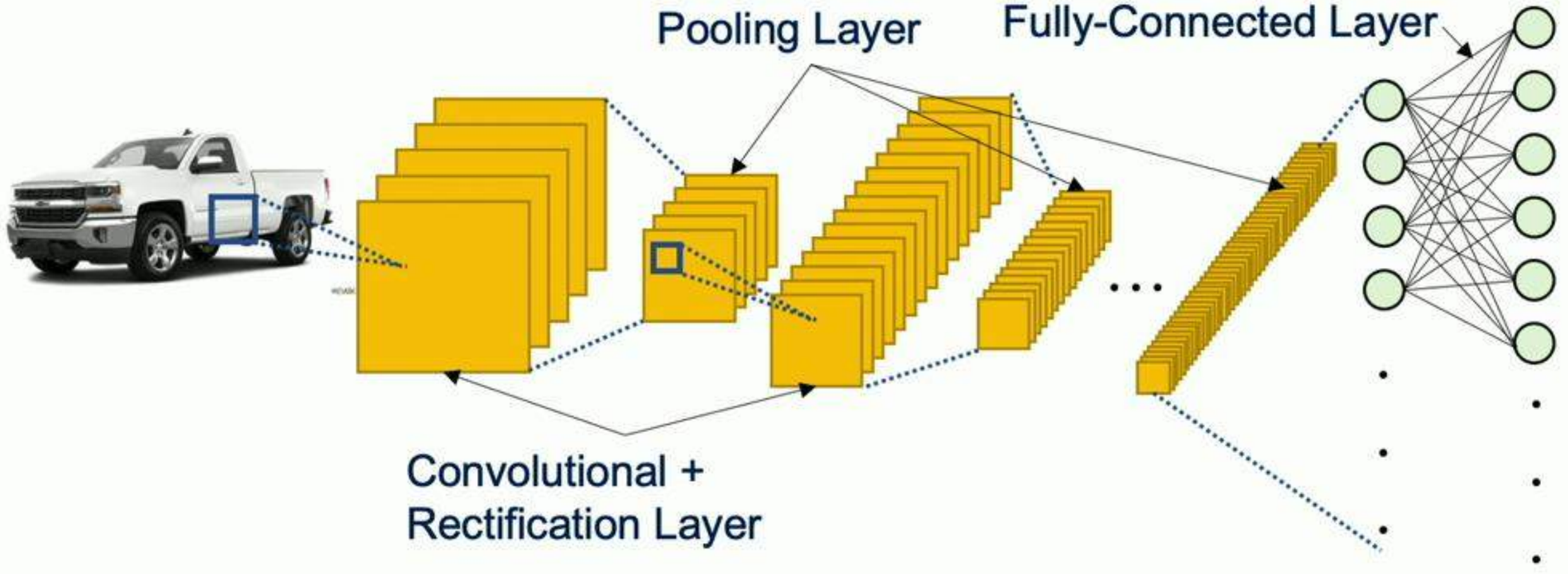
Low-Latency and Low-Cost Video Querying System

Our Goal

Enable low-latency and low-cost querying over rapidly growing video datasets

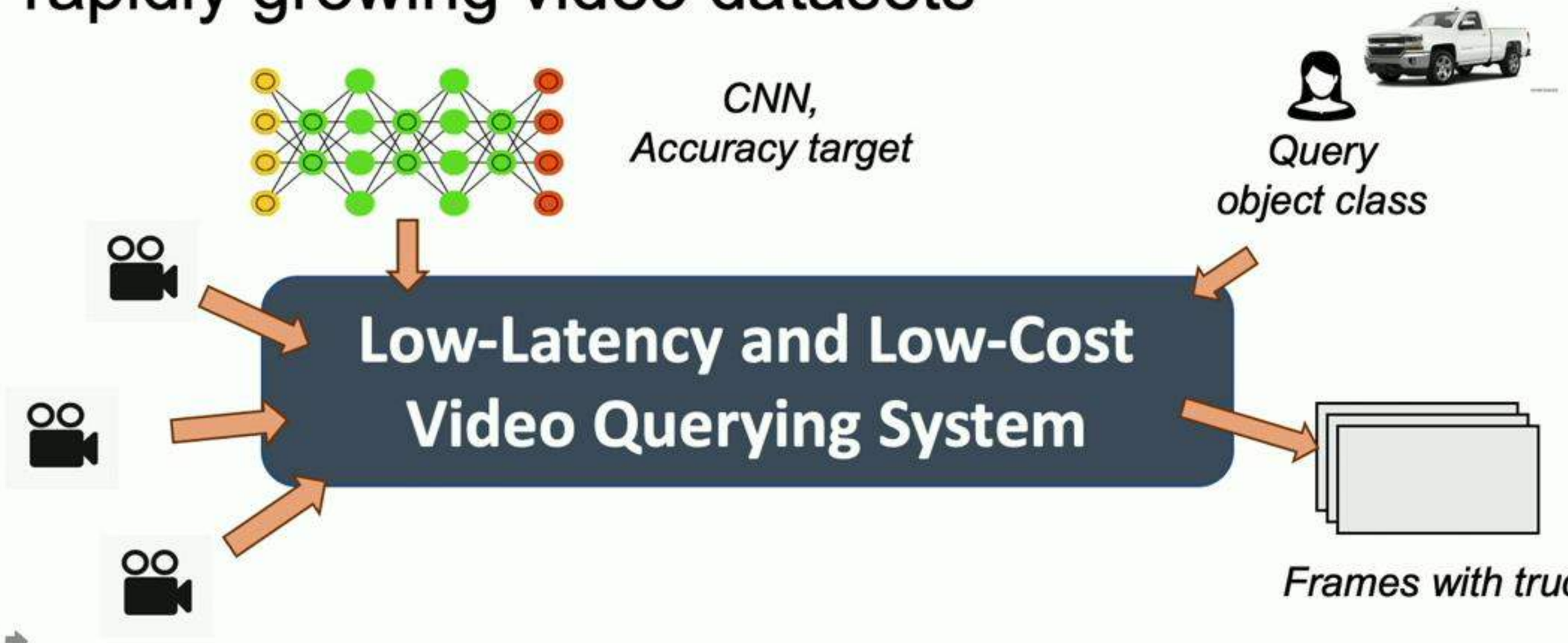


Background: Convolutional Neural Networks

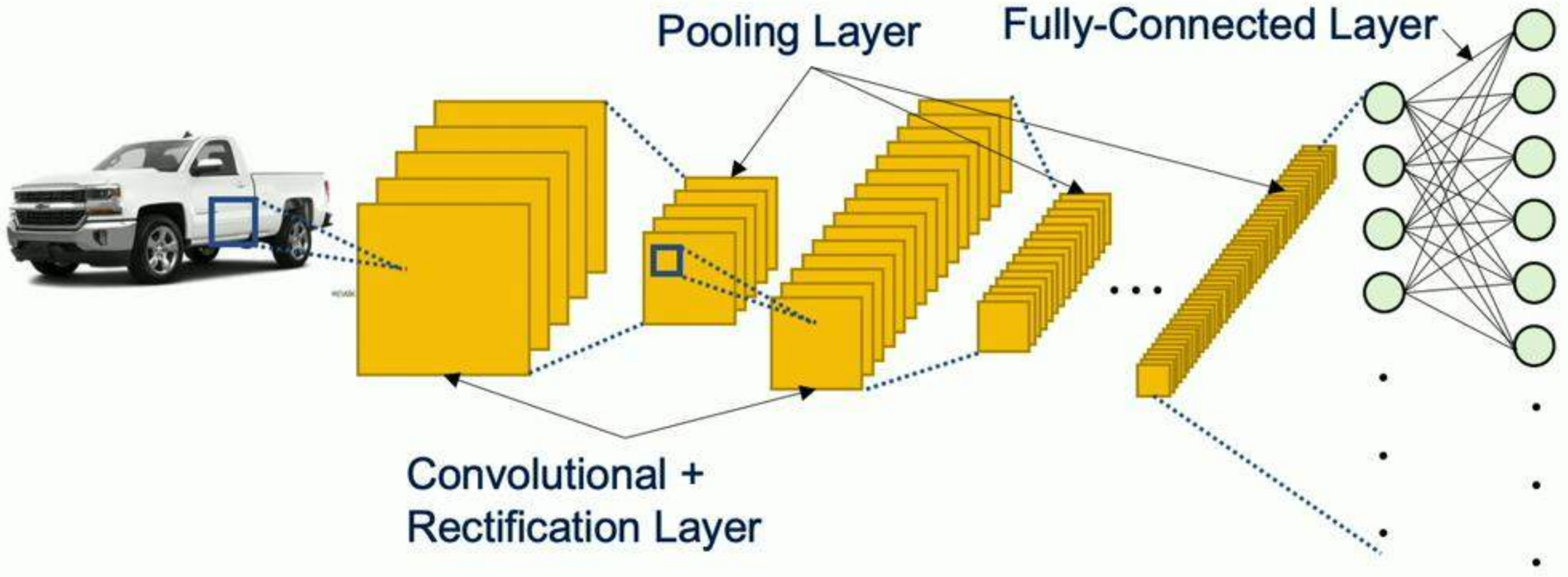


Our Goal

Enable low-latency and low-cost querying over rapidly growing video datasets

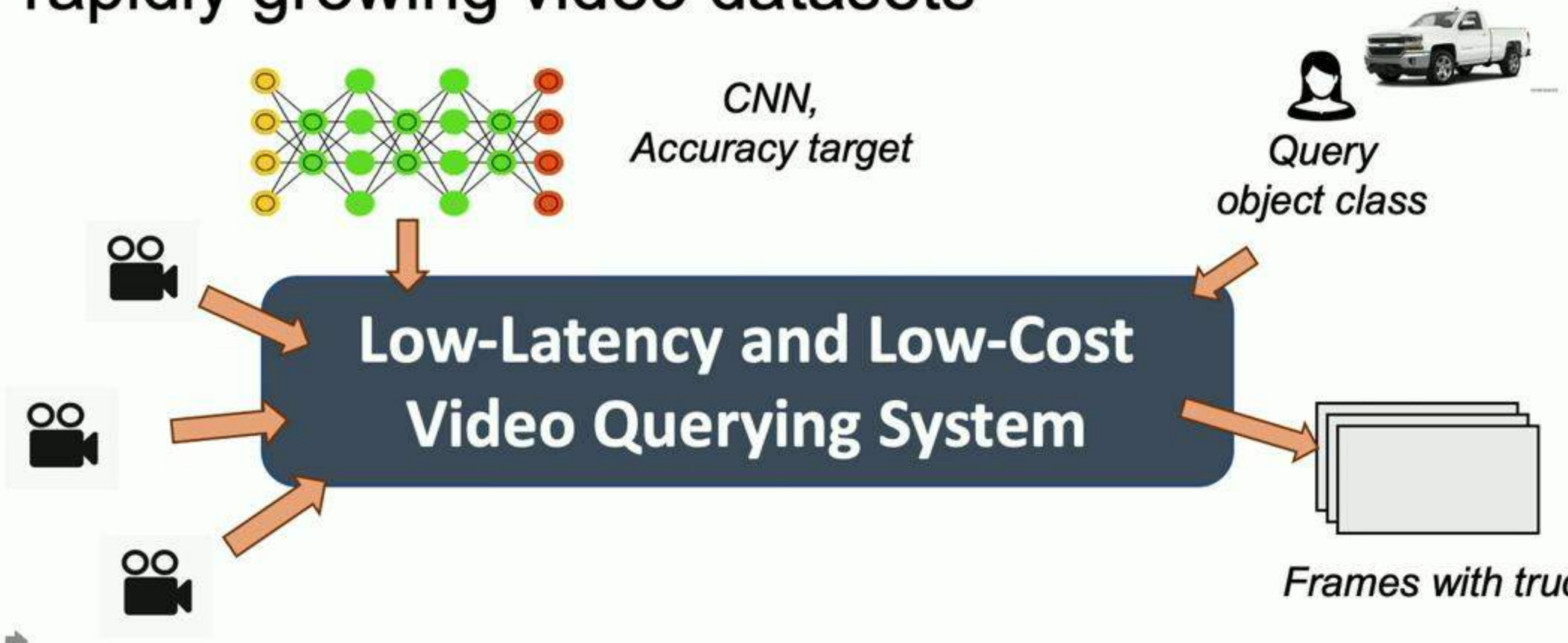


Background: Convolutional Neural Networks

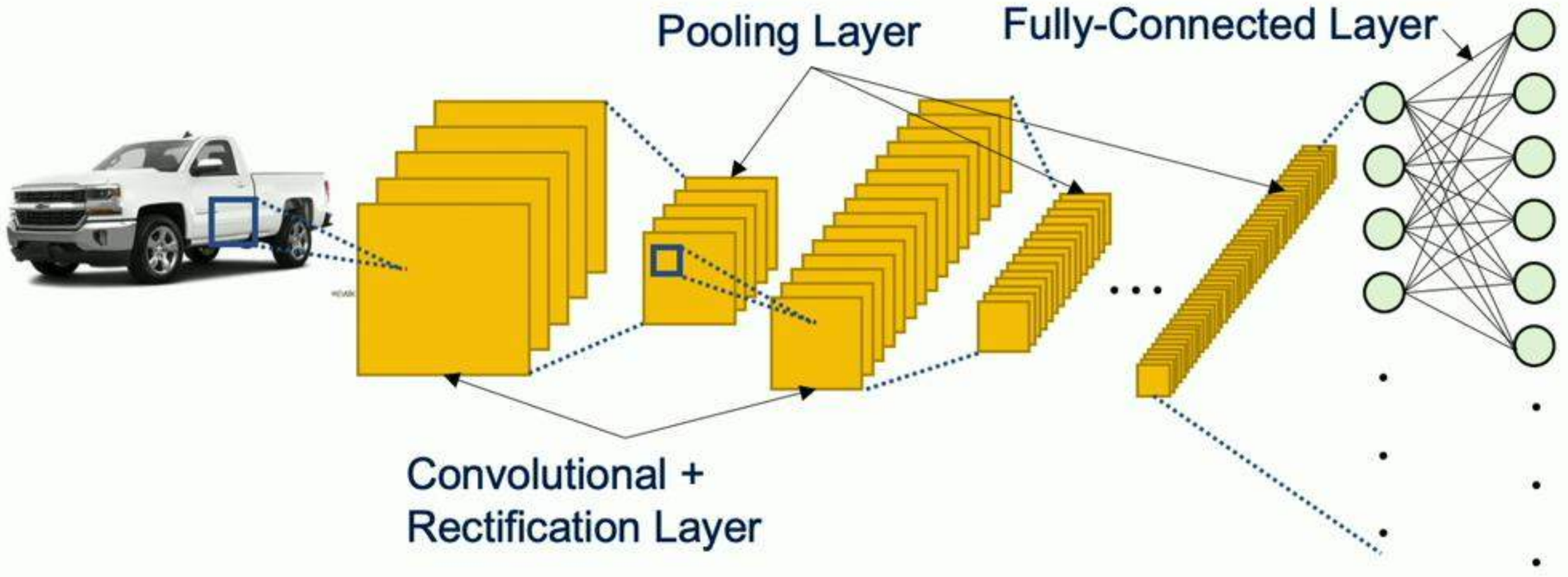


Our Goal

Enable low-latency and low-cost querying over rapidly growing video datasets

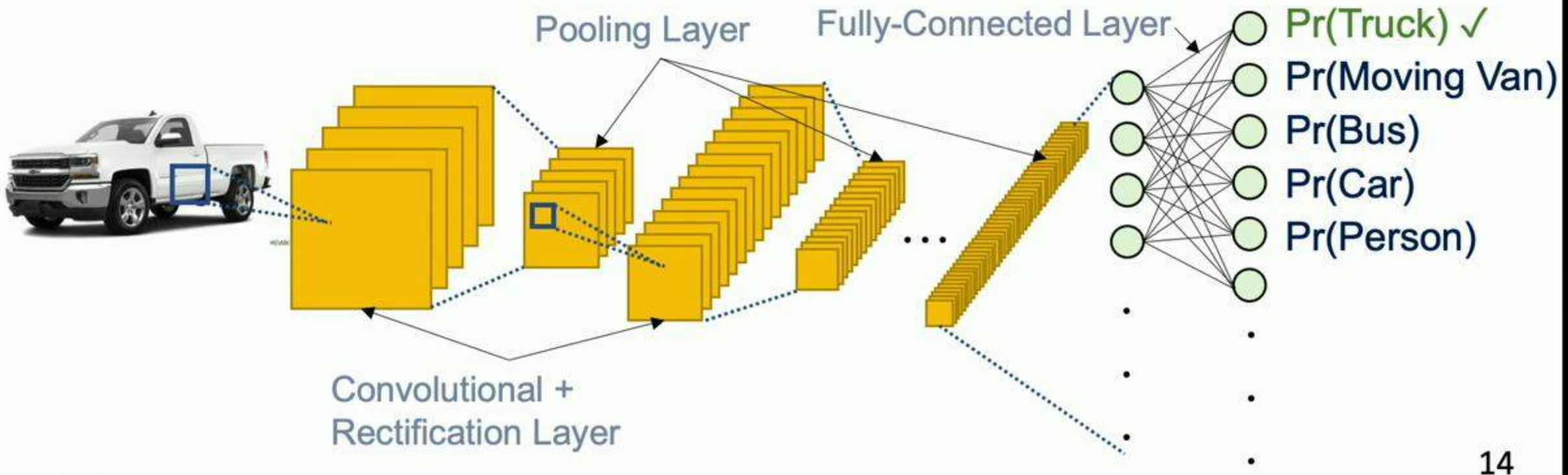


Background: Convolutional Neural Networks



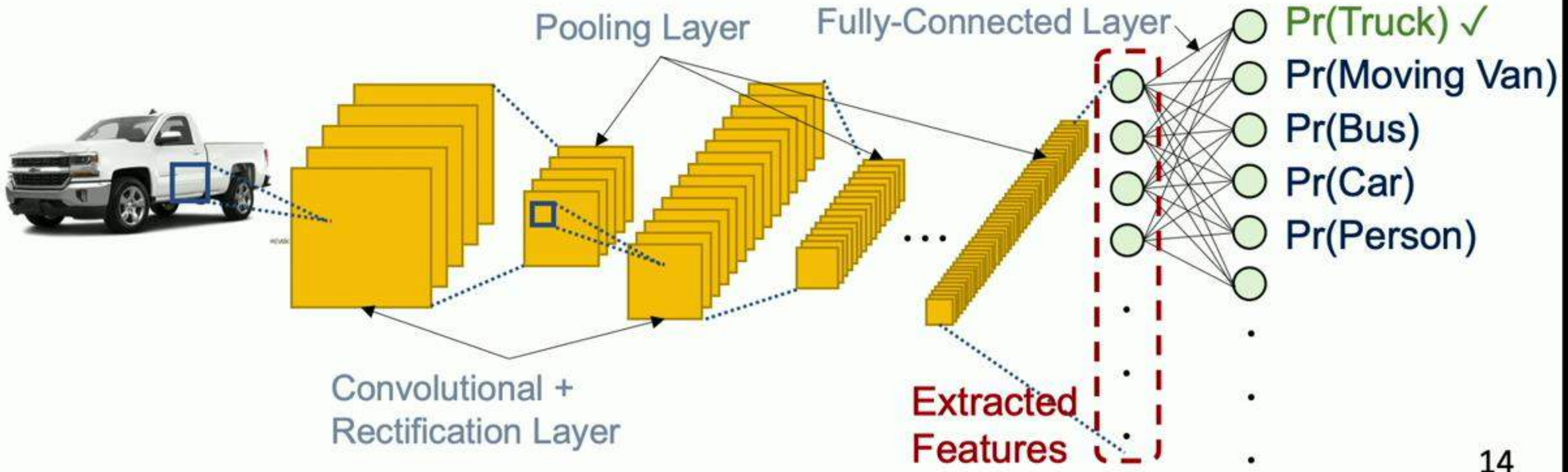
Background: Convolutional Neural Networks

- A Convolutional Neural Network (CNN) outputs the probability of each class



Background: Convolutional Neural Networks

- A Convolutional Neural Network (CNN) outputs the probability of each class
- Based on the extracted features (high-level representation)



Focus System: Low-latency query with low-cost ingest

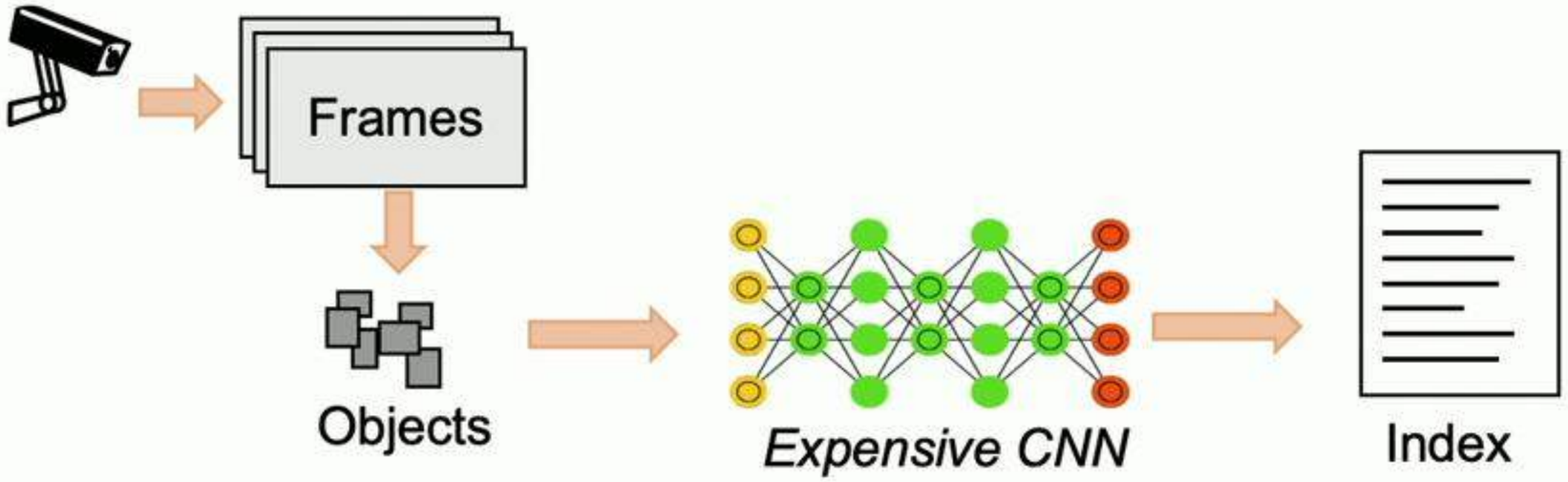
- Approximate indexing via cheap ingest
- Redundancy elimination for fast query
- Trading off ingest cost vs. query latency

Low-Cost Ingestion: Cheaper CNNs

- Process video frames with a cheap CNN at ingest time
  - Compressed and Specialized CNN: fewer layers / weights and are specialized for each video stream

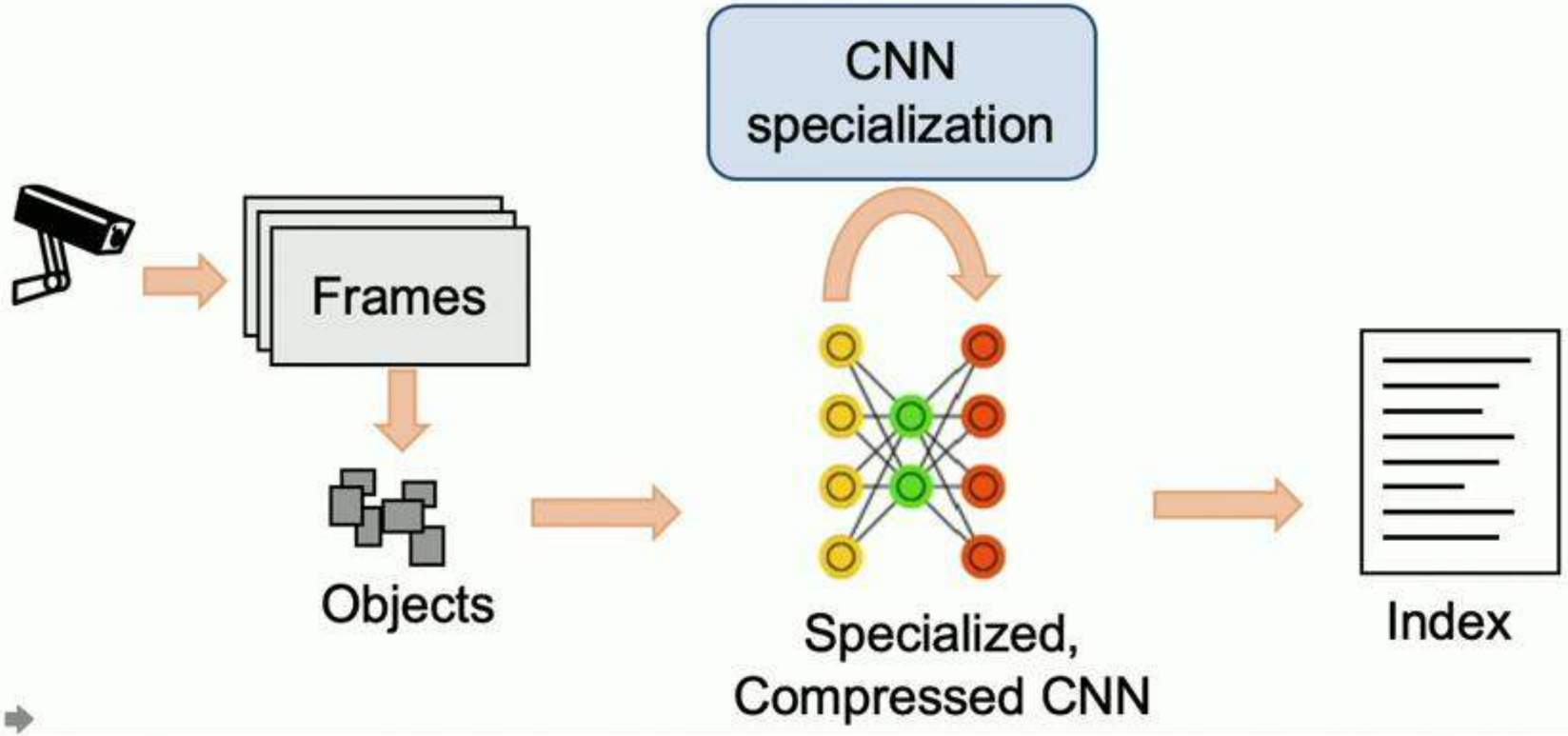
Low-Cost Ingestion: Cheaper CNNs

- Process video frames with a cheap CNN at ingest time
  - Compressed and Specialized CNN: fewer layers / weights and are specialized for each video stream



Low-Cost Ingestion: Cheaper CNNs

- Process video frames with a cheap CNN at ingest time
  - Compressed and Specialized CNN: fewer layers / weights and are specialized for each video stream



Challenge: Cheap CNNs are Less Accurate

- Cheaper CNNs are less accurate than the expensive CNNs

Rank	Expensive CNN	Cheap CNN
1	Truck	Moving Van

Challenge: Cheap CNNs are Less Accurate

- Cheaper CNNs are less accurate than the expensive CNNs

The best result from the expensive CNN is within the top-K results of the cheaper CNN

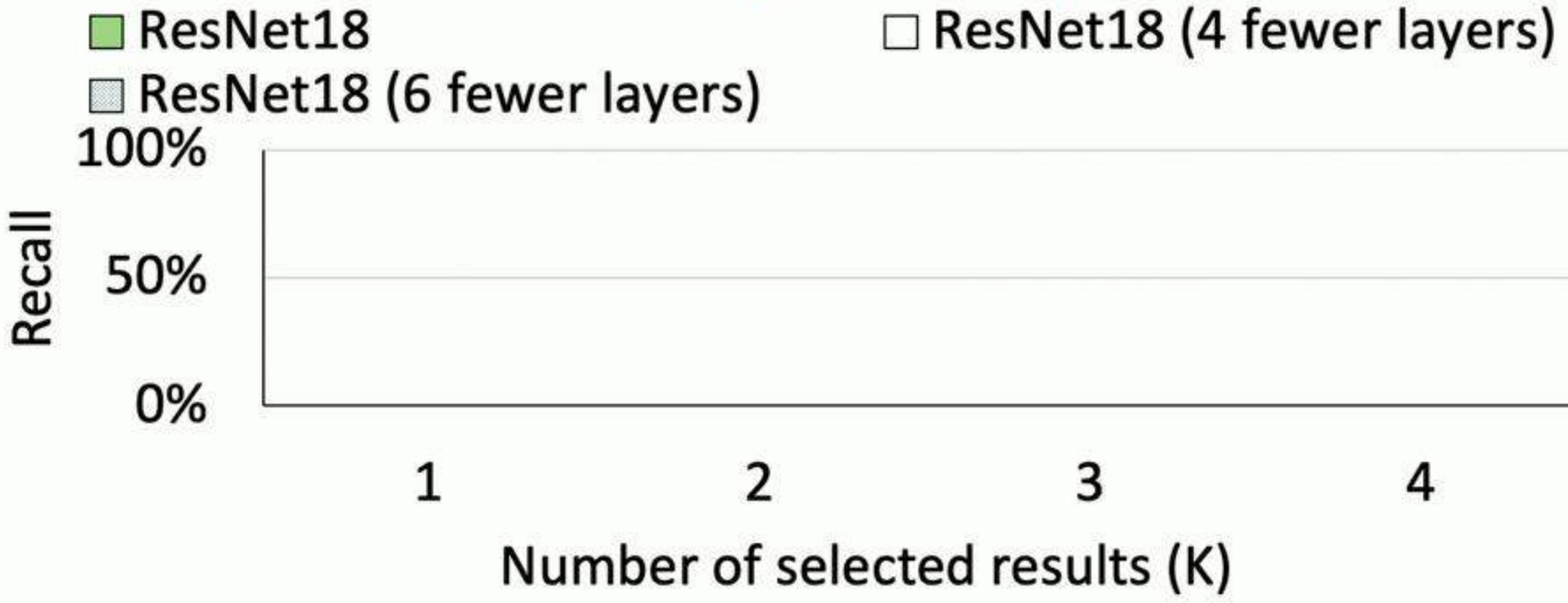
Rank	Expensive CNN	Cheap CNN
1	Truck	Moving Van ✗
2	Moving Van	Airplane
3	Passenger Car	Truck ✓
4	Recreational vehicle	Passenger Car

Recall, Precision and Top-K Results

Recall: Fraction of relevant objects that are selected

Precision: Fraction of selected objects that are relevant

Ground-truth CNN: YOLOv2 (80 classes)

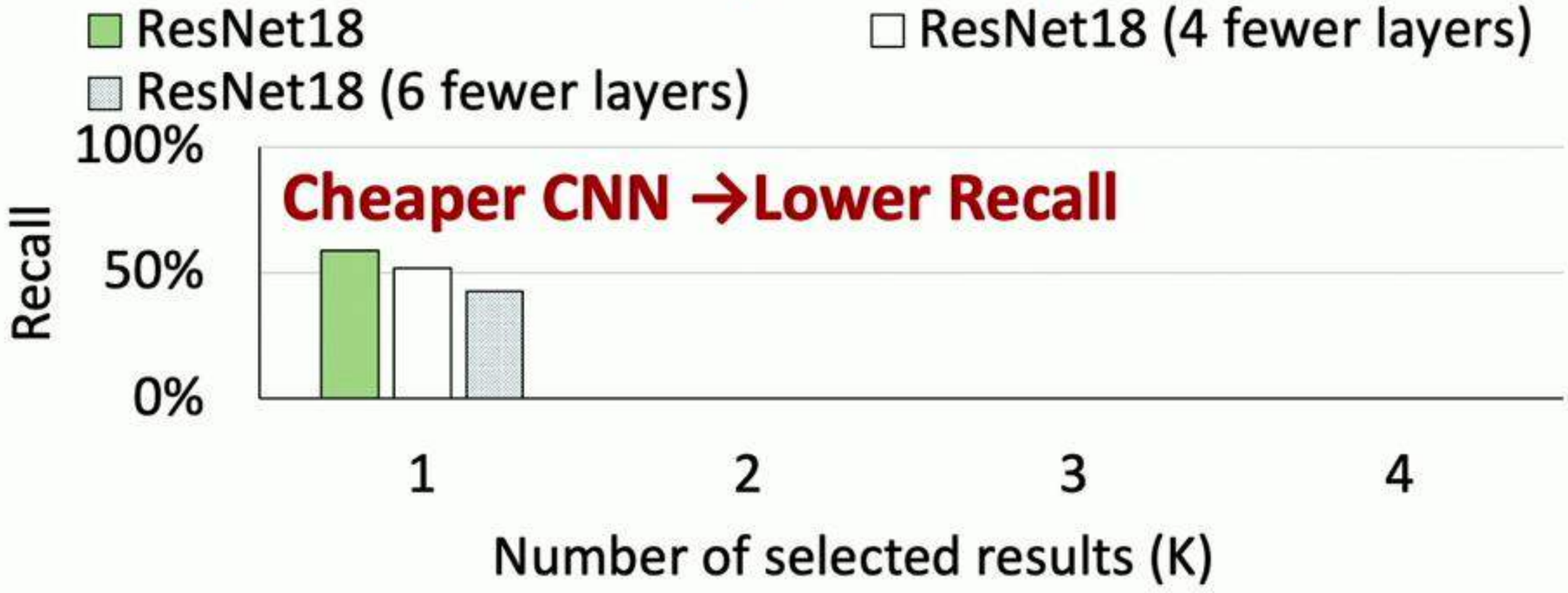


Recall, Precision and Top-K Results

Recall: Fraction of relevant objects that are selected

Precision: Fraction of selected objects that are relevant

Ground-truth CNN: YOLOv2 (80 classes)

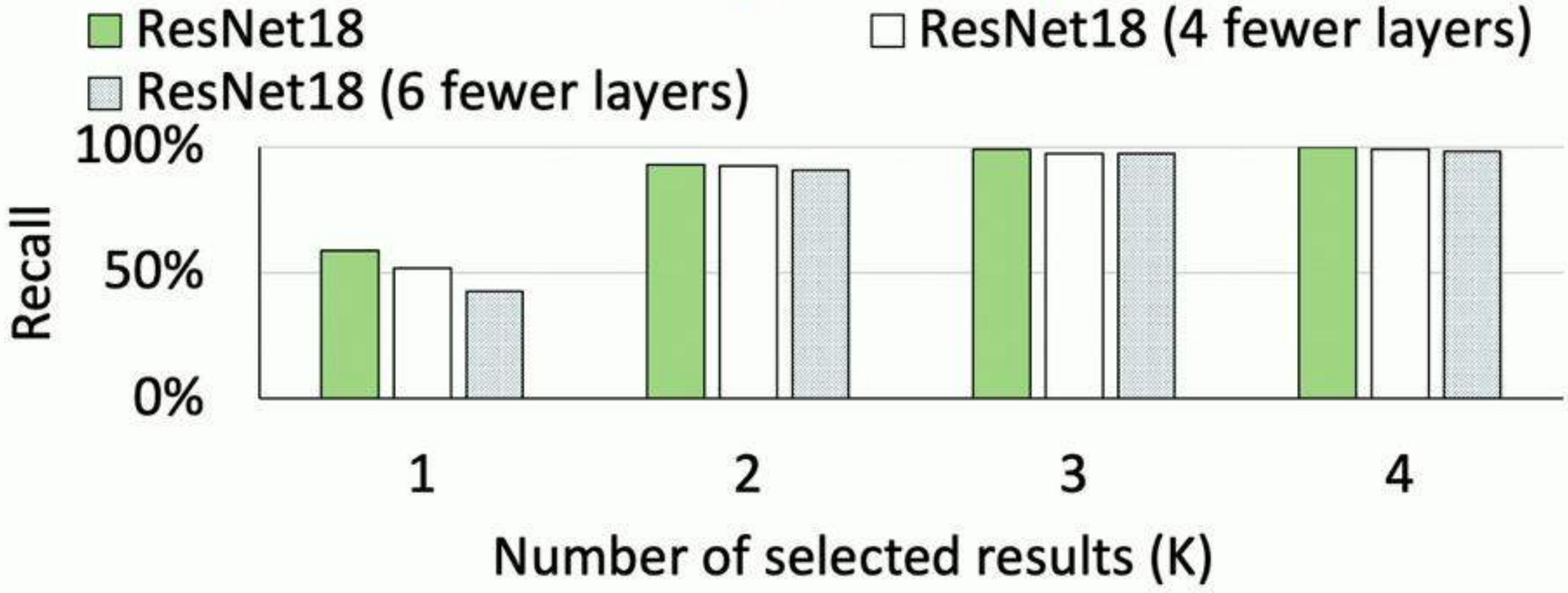


Recall, Precision and Top-K Results

Recall: Fraction of relevant objects that are selected

Precision: Fraction of selected objects that are relevant

Ground-truth CNN: YOLOv2 (80 classes)

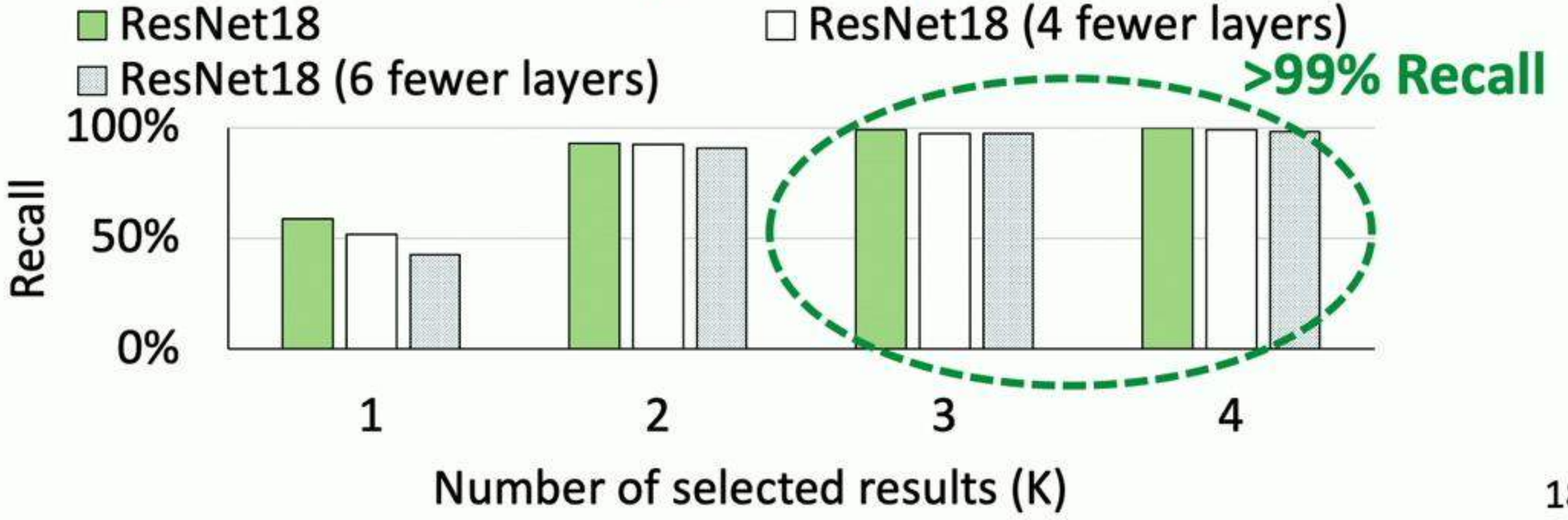


Recall, Precision and Top-K Results

Recall: Fraction of relevant objects that are selected

Precision: Fraction of selected objects that are relevant

Ground-truth CNN: YOLOv2 (80 classes)

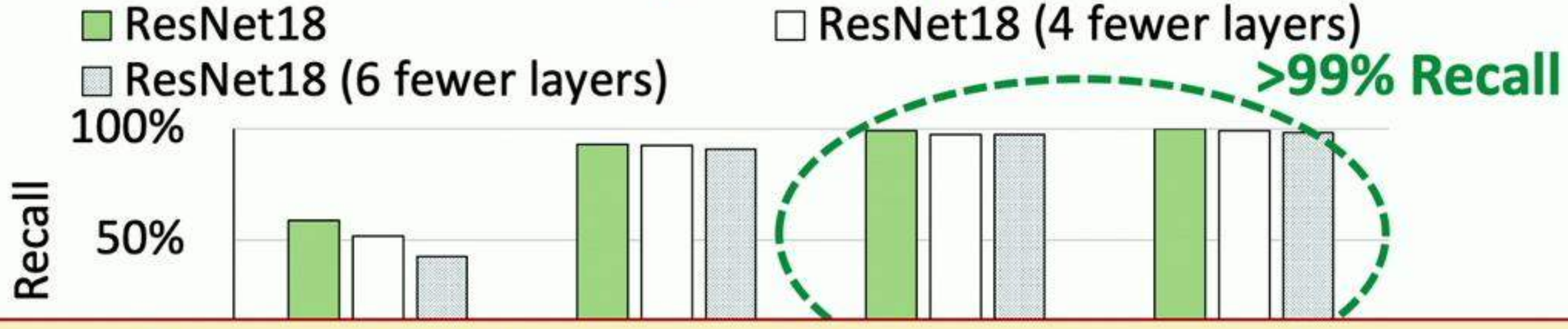


Recall, Precision and Top-K Results

Recall: Fraction of relevant objects that are selected

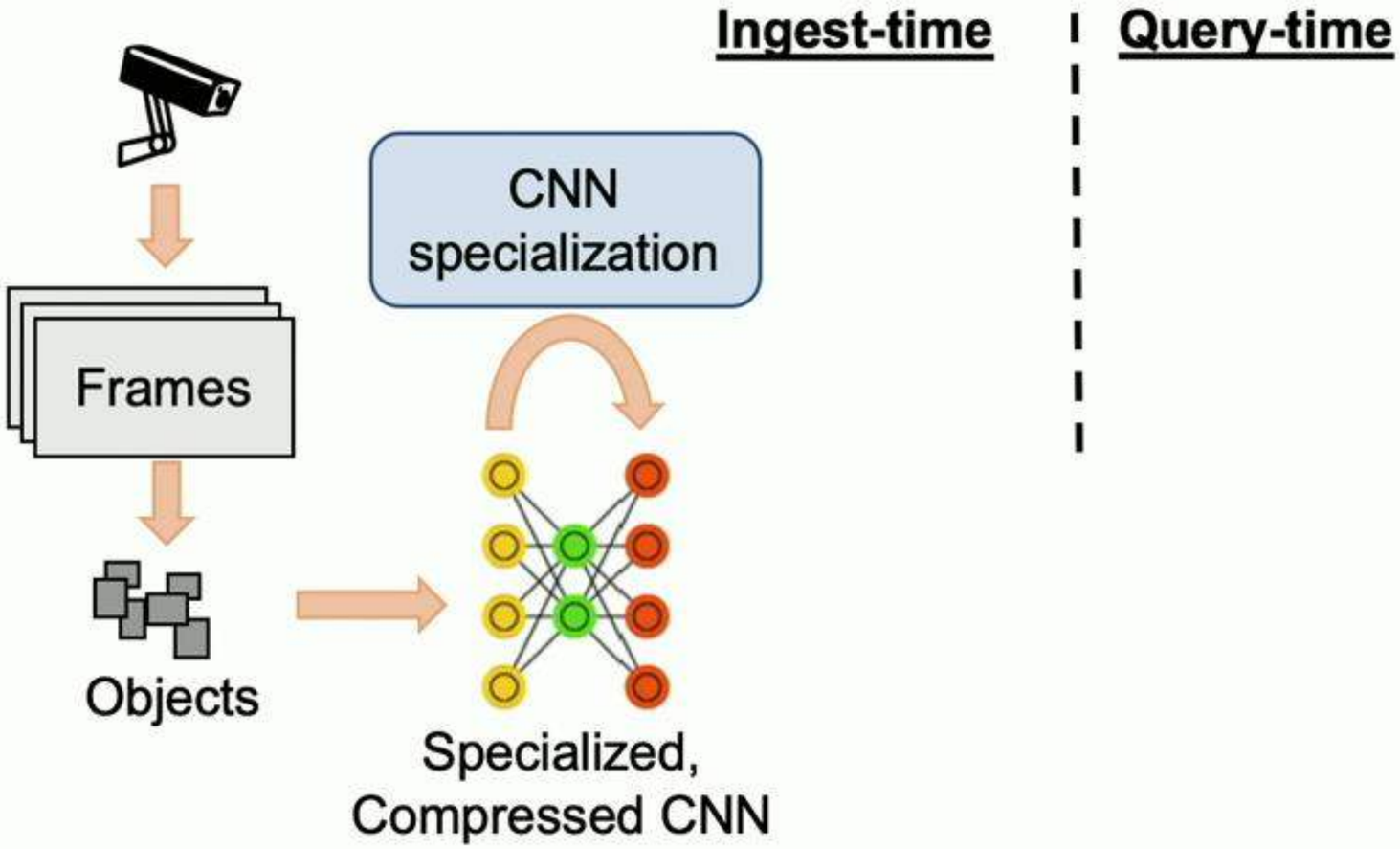
Precision: Fraction of selected objects that are relevant

Ground-truth CNN: YOLOv2 (80 classes)

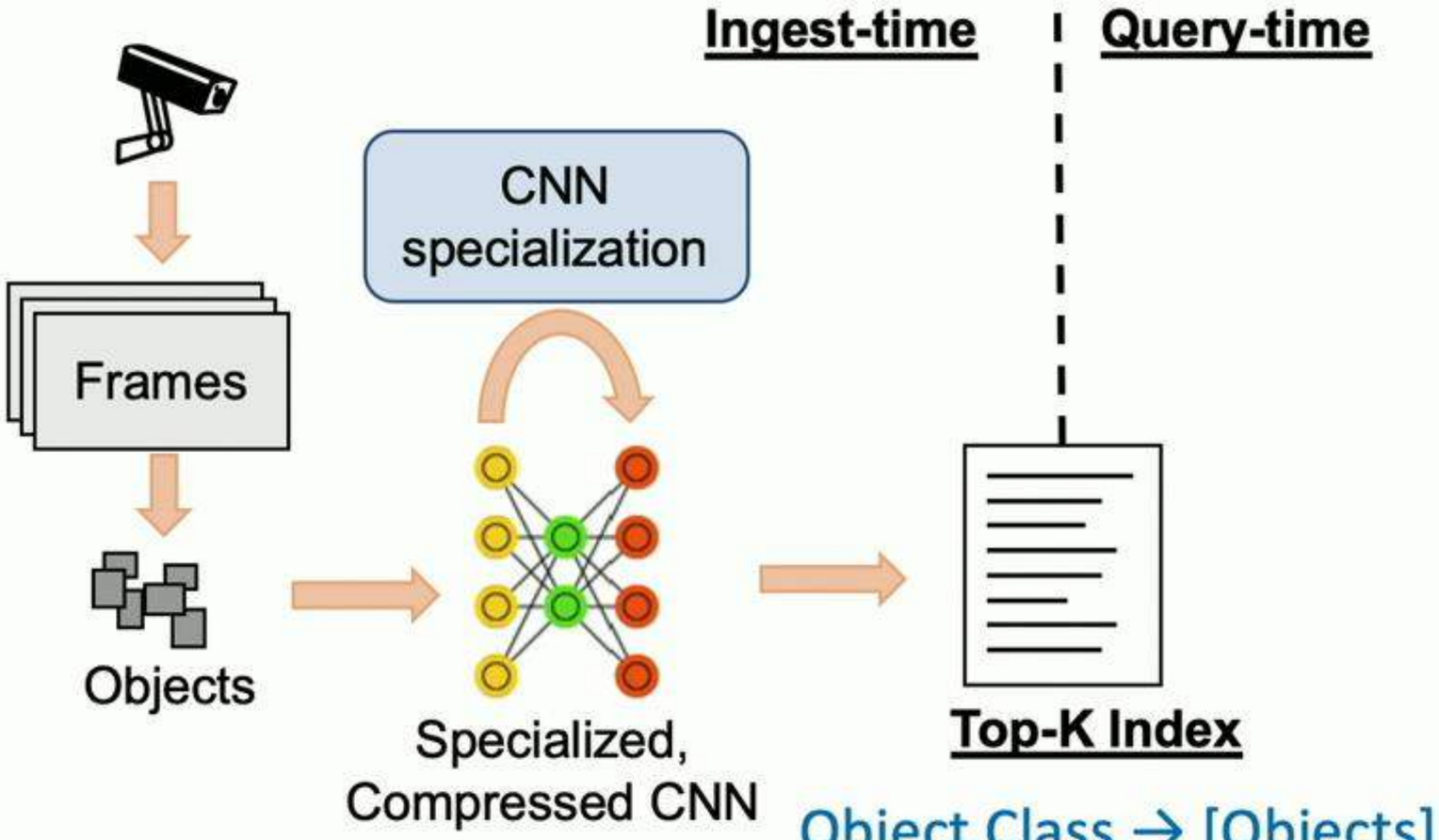


Cheap CNNs can achieve high recall with small top-K results

Solution: Split Ingest- and Query-time Work



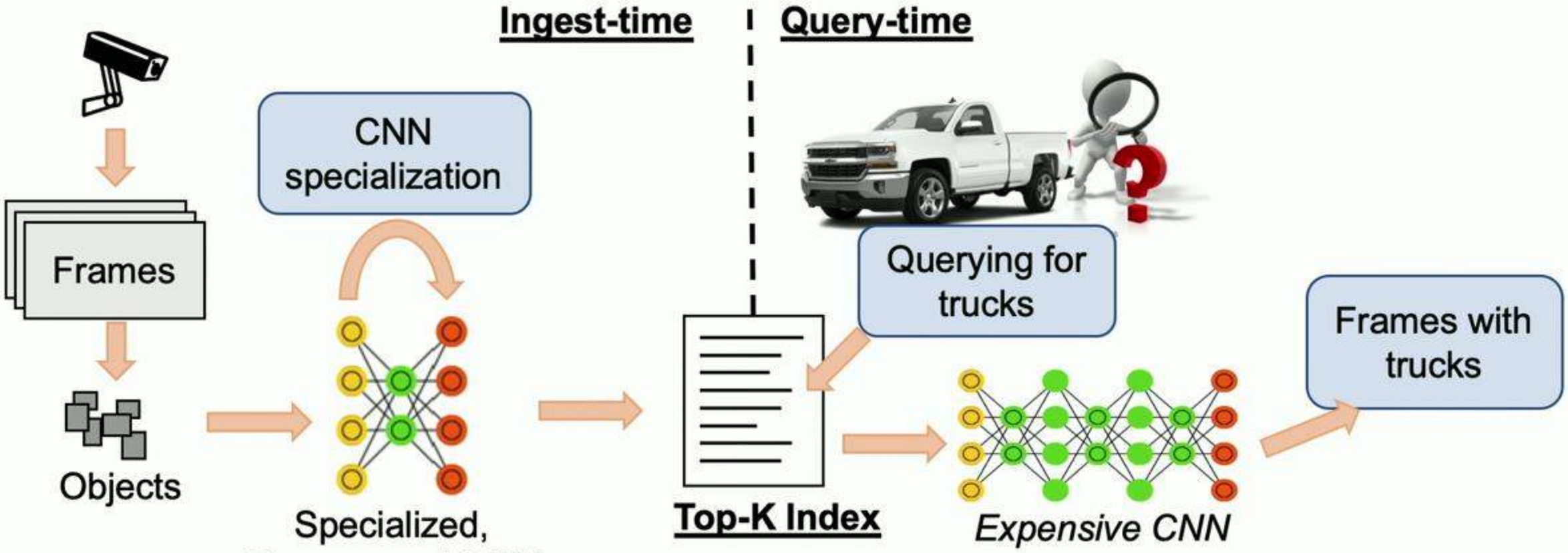
Solution: Split Ingest- and Query-time Work



Object Class → [Objects]

Object → Frame

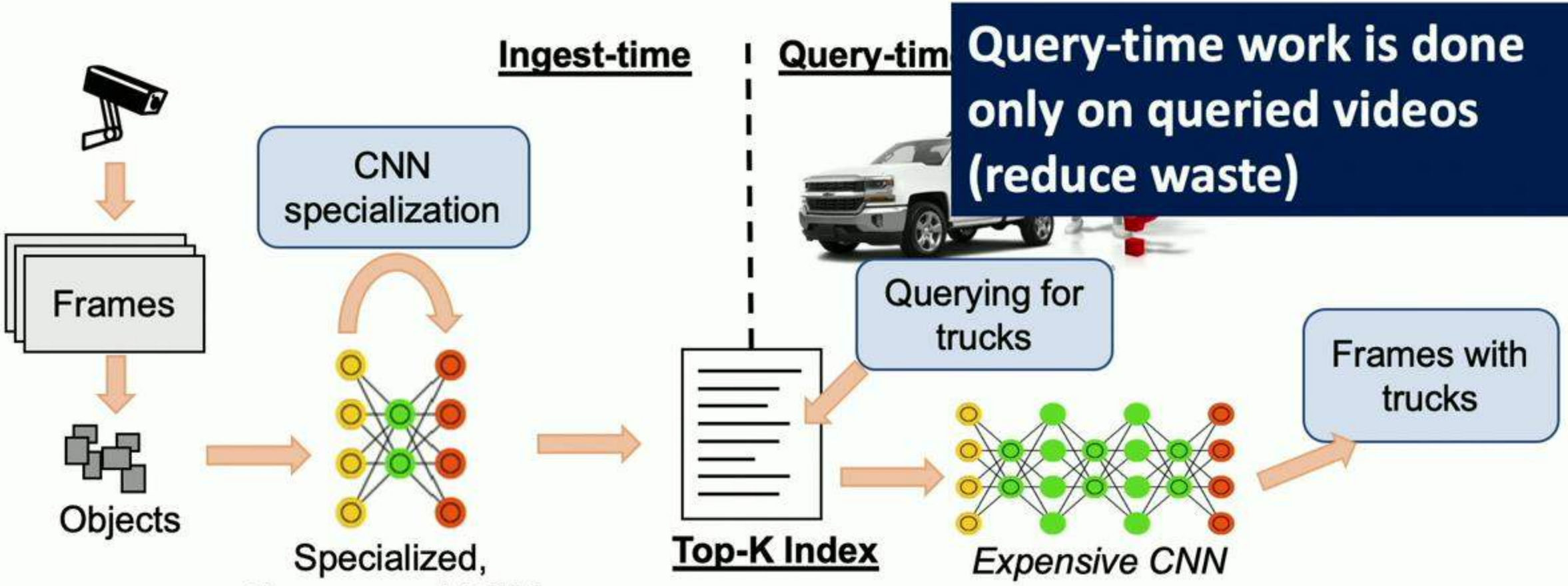
Solution: Split Ingest- and Query-time Work



Object Class → [Objects]

Object → Frame

Solution: Split Ingest- and Query-time Work

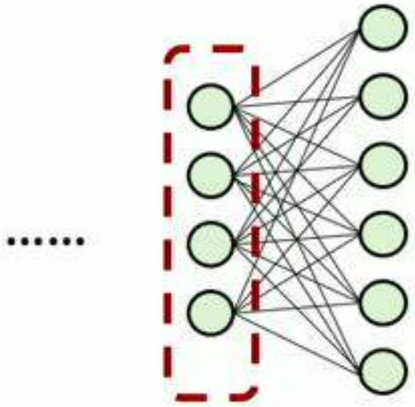


Focus System: Low-latency query with low-cost ingest

- Approximate indexing via cheap ingest
- Redundancy elimination for fast query
- Trading off ingest cost vs. query latency

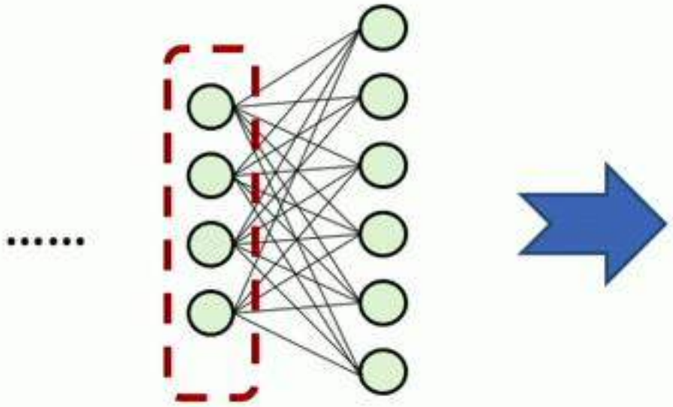
Low-Latency Query: Redundancy Elimination

- Approximate indexing ➔ non-trivial work at query time
  - A larger K ➔ more query-time work
- Images with similar feature vectors are visually similar



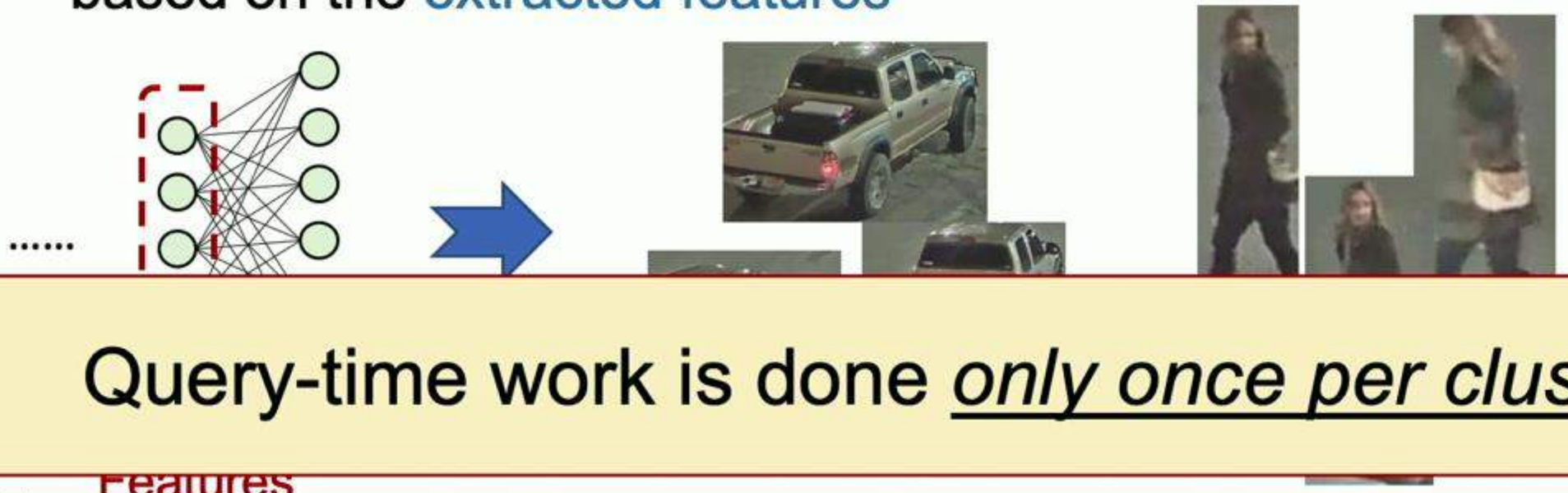
Low-Latency Query: Redundancy Elimination

- Approximate indexing ➔ non-trivial work at query time
  - A larger K ➔ more query-time work
- Images with similar feature vectors are visually similar
- Minimize the work at query time ➔ clustering similar objects based on the extracted features



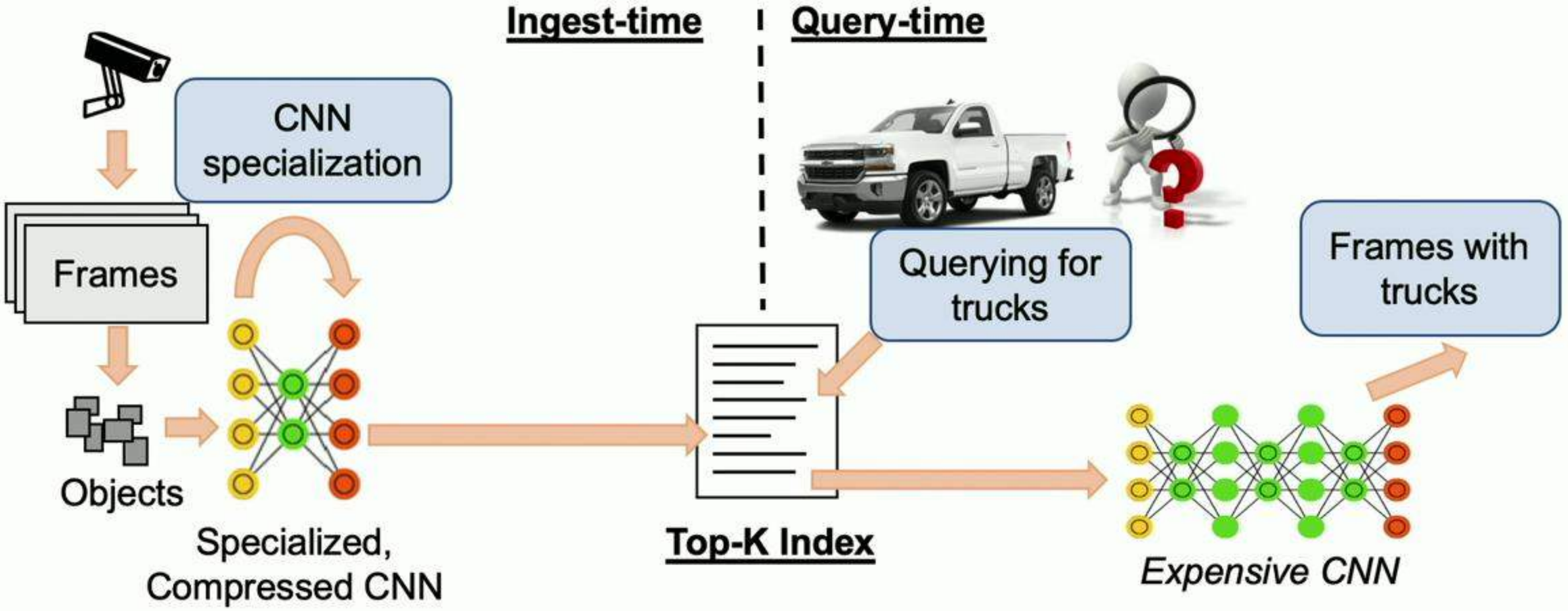
Low-Latency Query: Redundancy Elimination

- Approximate indexing ➔ non-trivial work at query time
  - A larger K ➔ more query-time work
- Images with similar feature vectors are visually similar
- Minimize the work at query time ➔ clustering similar objects based on the extracted features

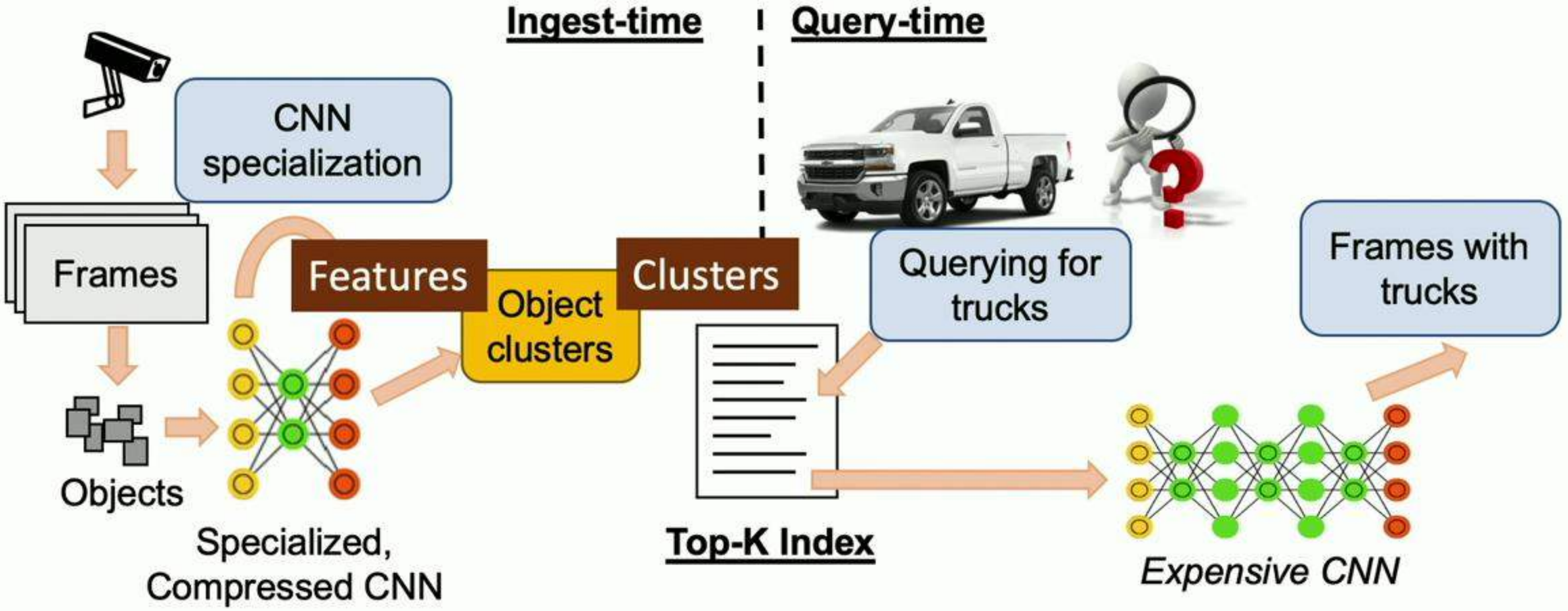


Query-time work is done *only once per cluster*

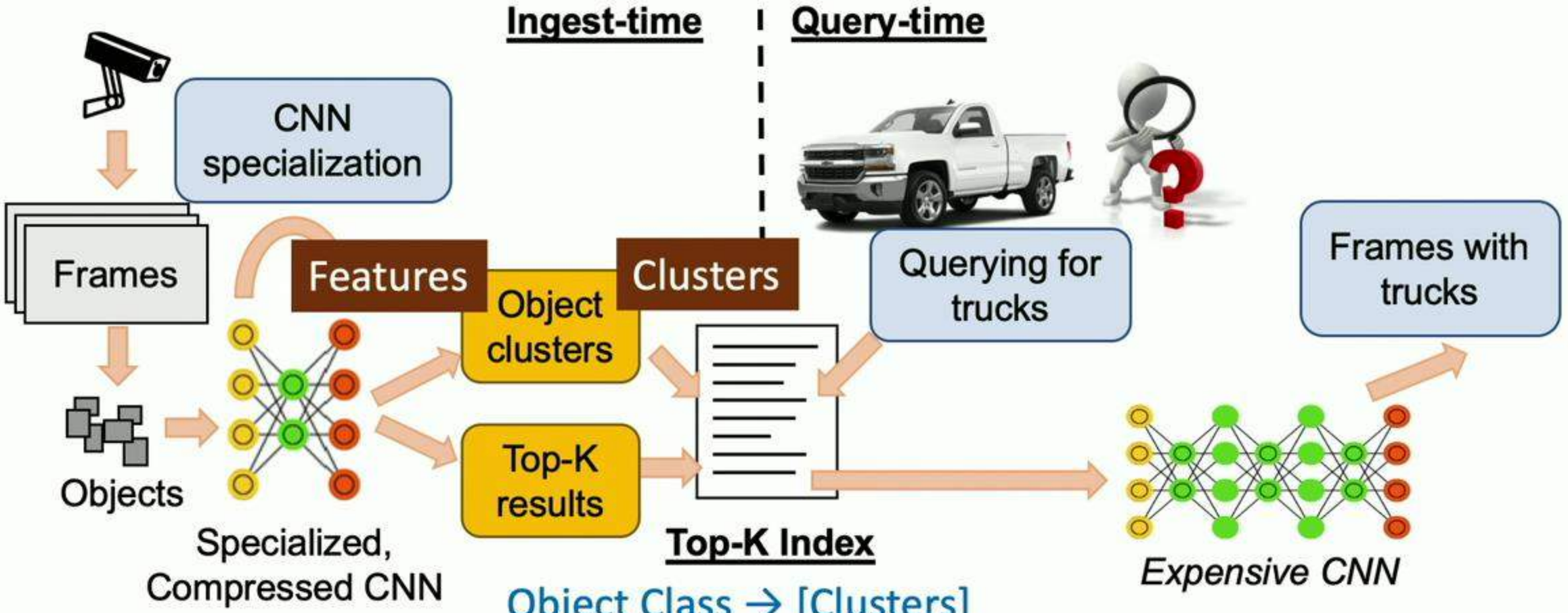
Adding Feature-based Clustering



Adding Feature-based Clustering



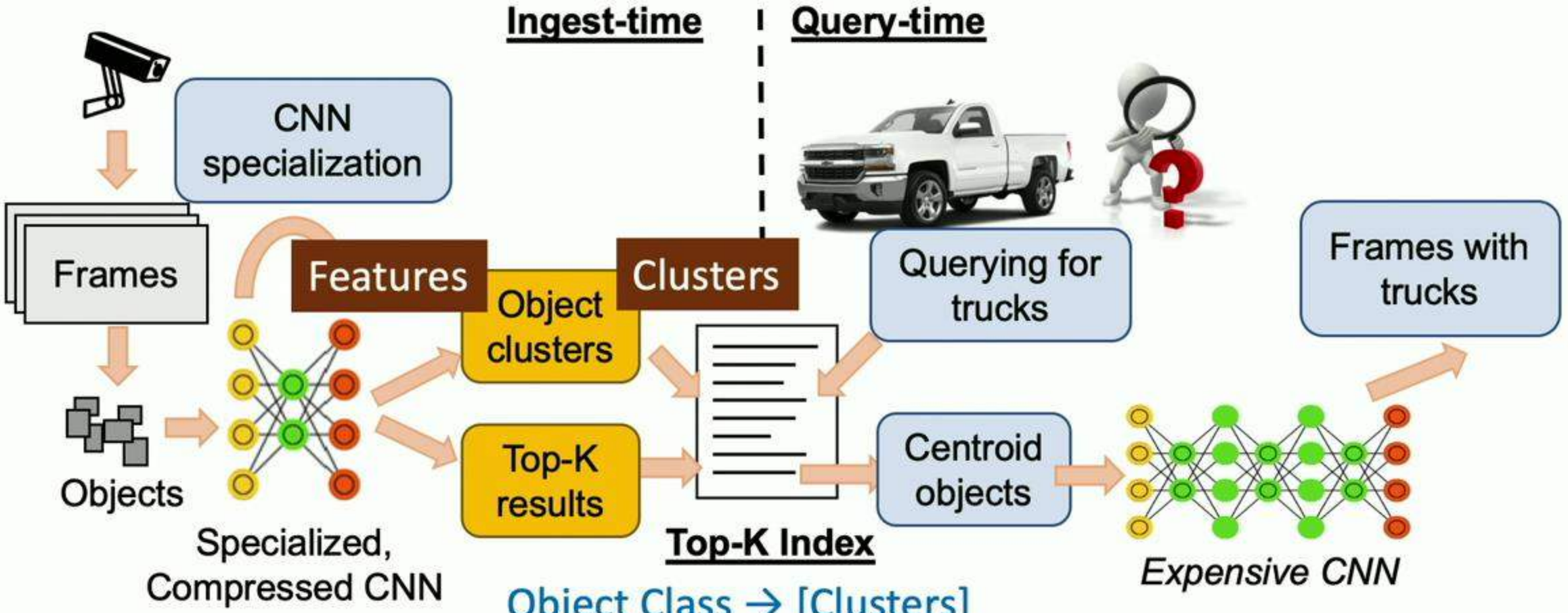
Adding Feature-based Clustering



Object Class → [Clusters]

Cluster → Centroid, [Objects], [Frames]

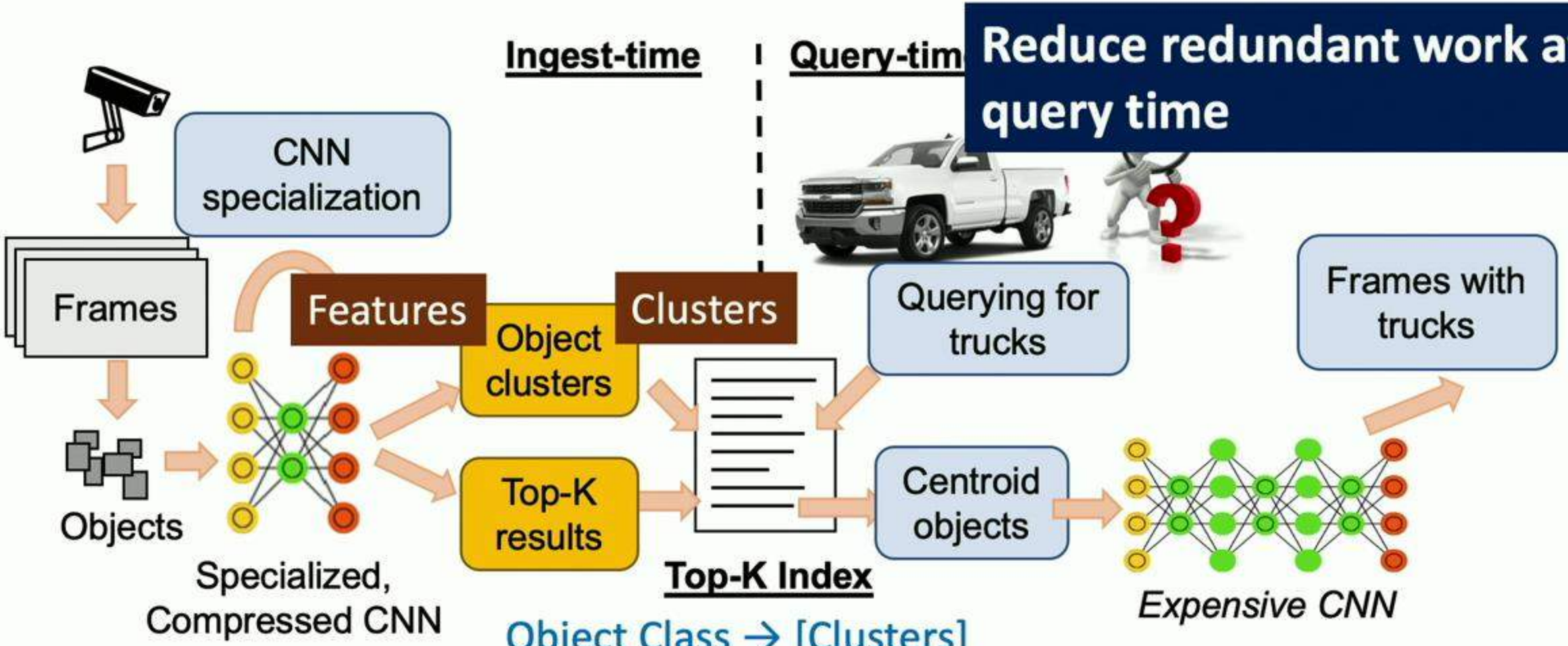
Adding Feature-based Clustering



Object Class → [Clusters]

Cluster → Centroid, [Objects], [Frames]

Adding Feature-based Clustering



Experimental Setup

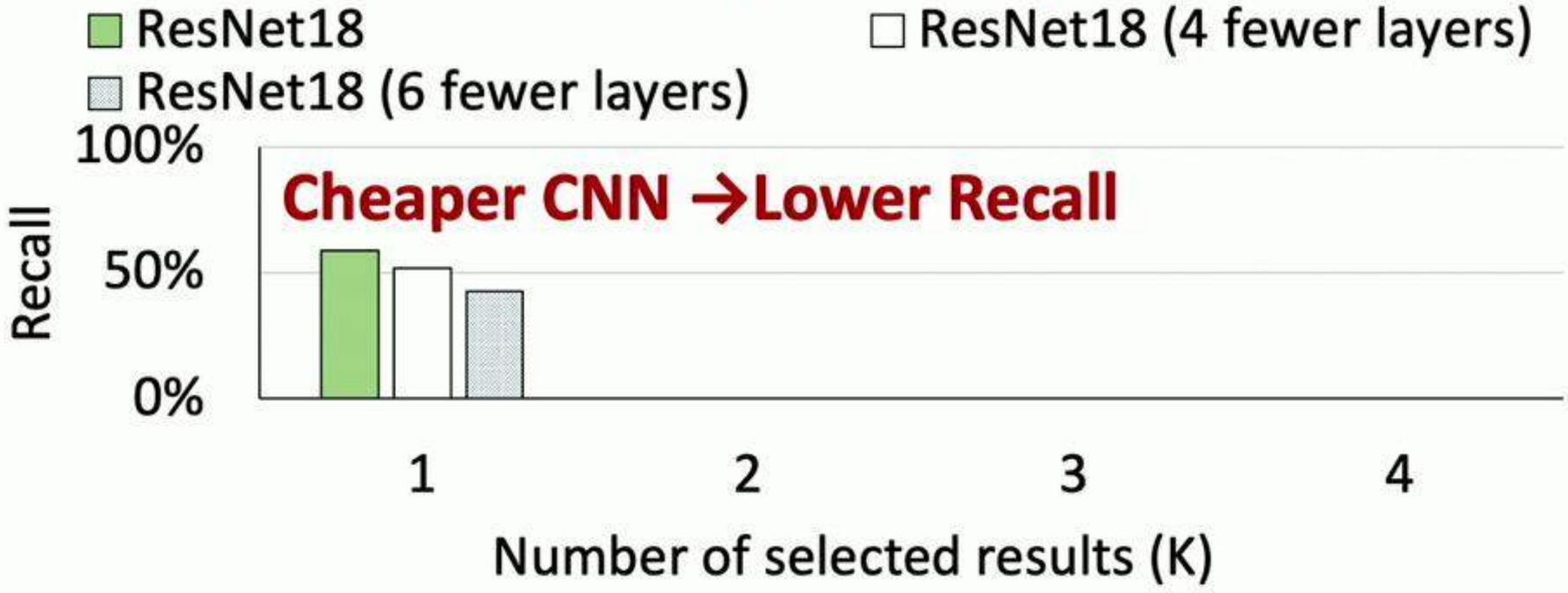
- **Video Datasets**
  - 11 live traffic and enterprise videos
  - Each video stream is evaluated for 12 hours
- **Accuracy Targets**
  - 99% recall and 99% precision w.r.t. YOLOv2
- **Baselines**
  - Ingest-heavy: Analyzes all frames with YOLOv2 at ingest time and stores the inverted index for query
  - NoScope [VLDB'17]: A query-optimized system that analyzes frames only at query time

Recall, Precision and Top-K Results

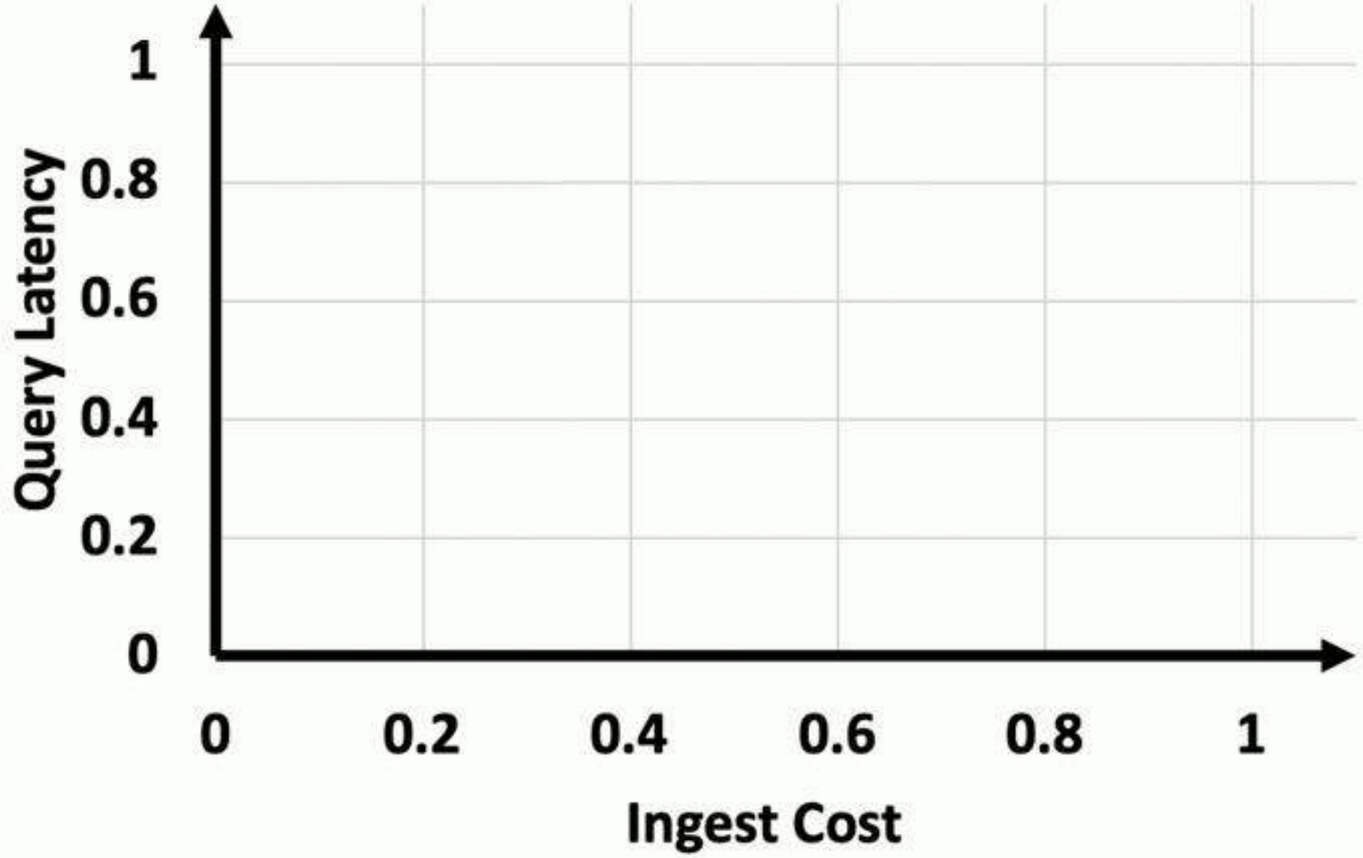
Recall: Fraction of relevant objects that are selected

Precision: Fraction of selected objects that are relevant

Ground-truth CNN: YOLOv2 (80 classes)

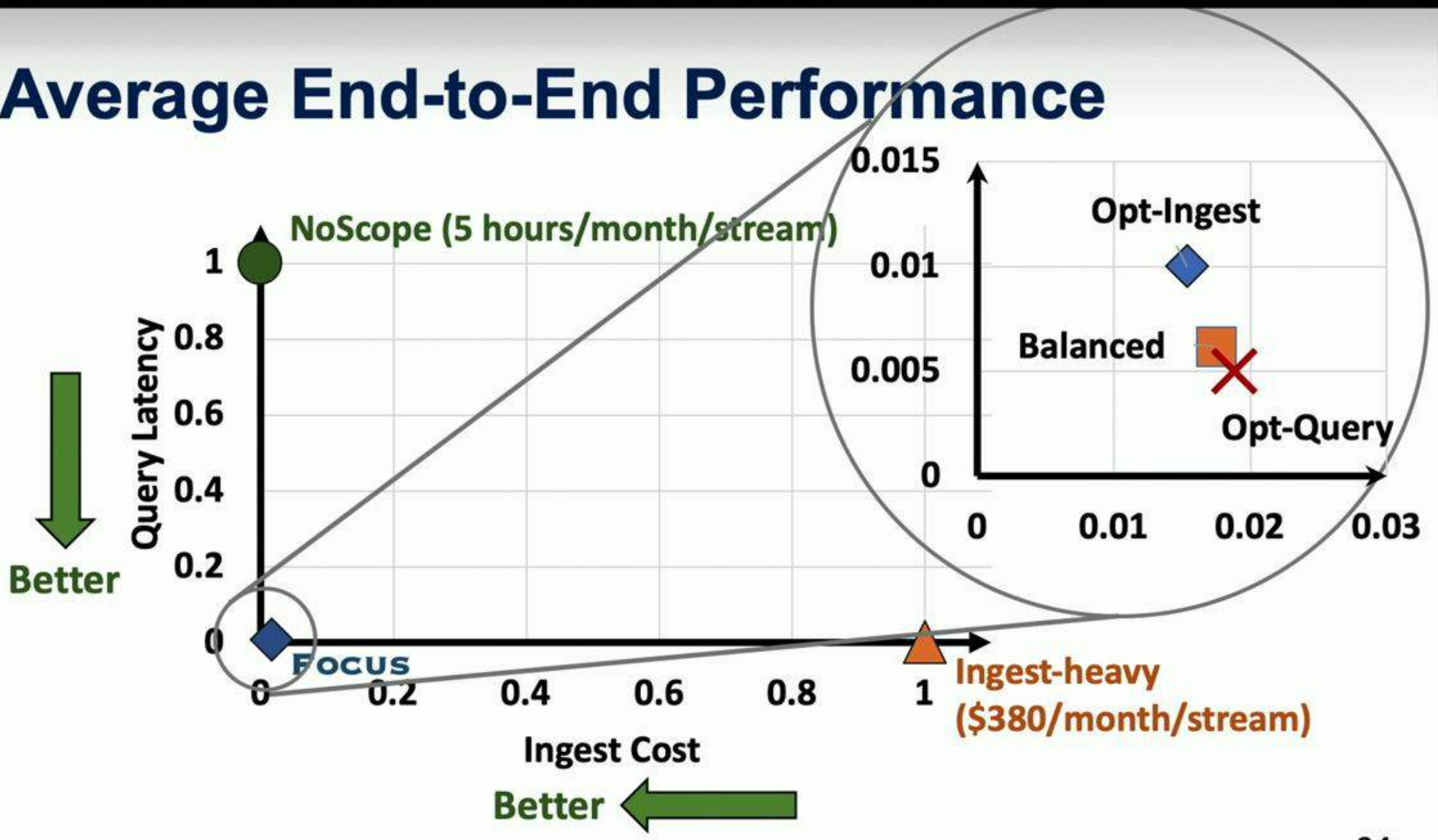


Average End-to-End Performance

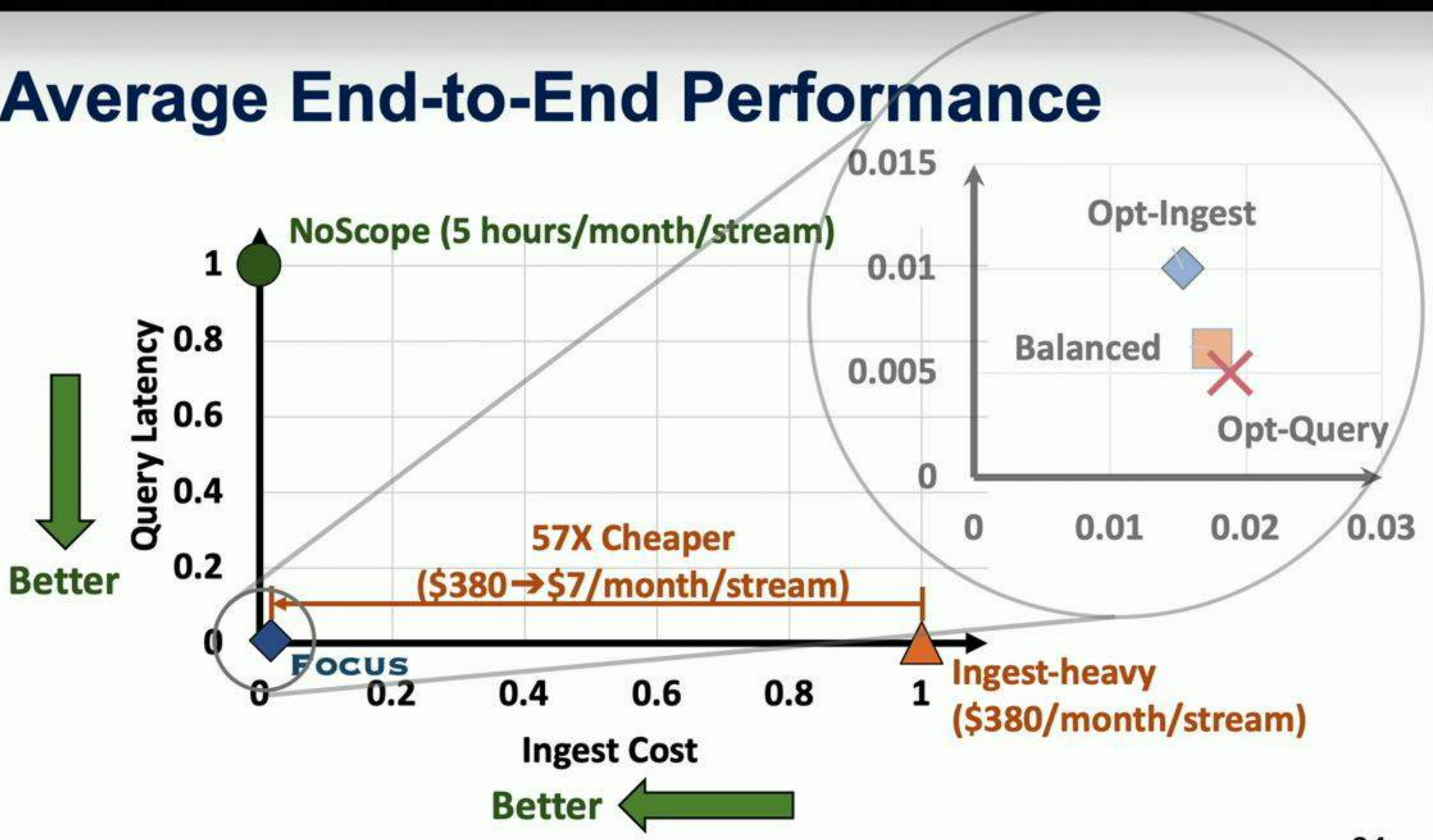


Average End-to-End Performance

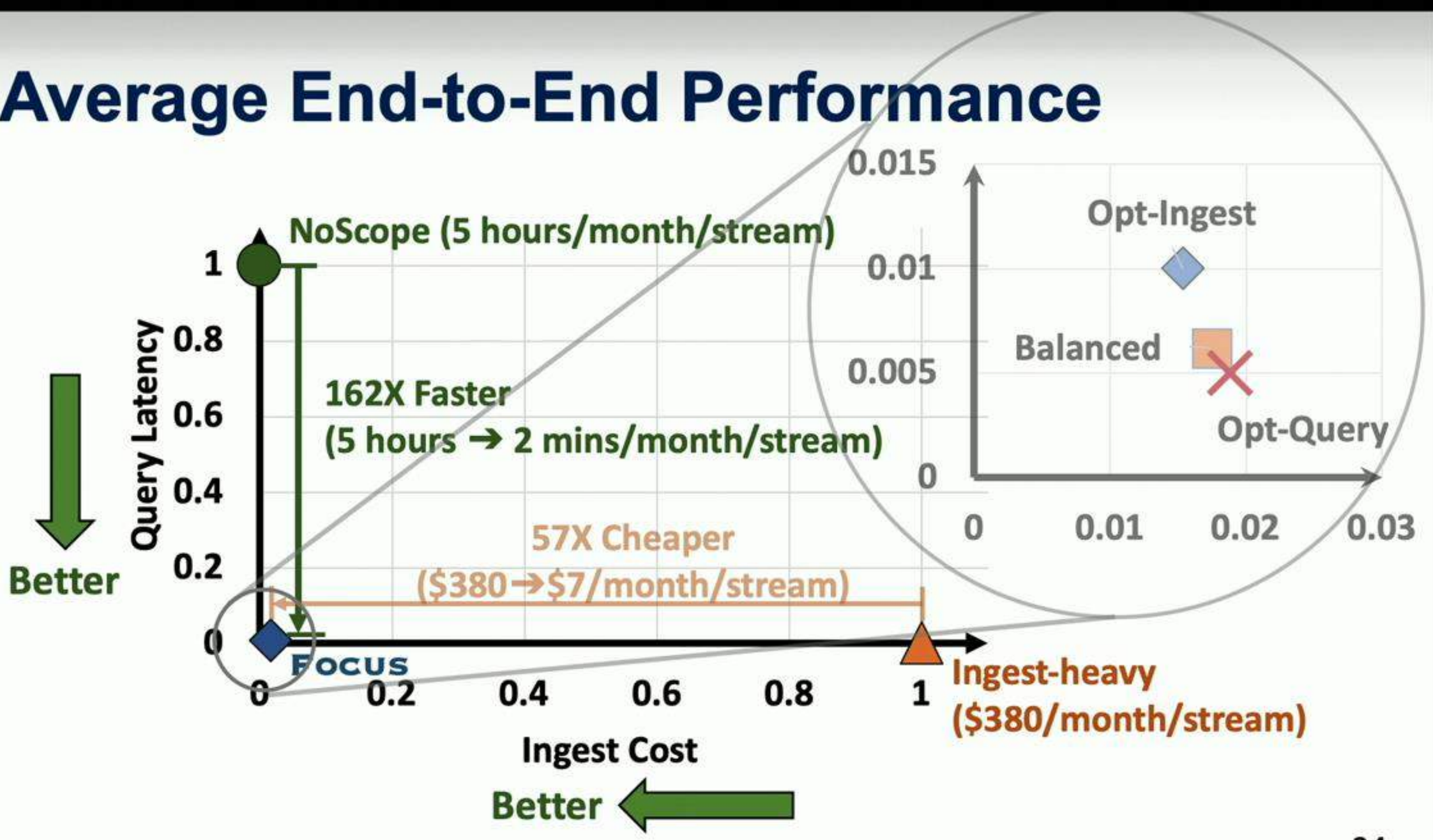
Average End-to-End Performance



Average End-to-End Performance



Average End-to-End Performance

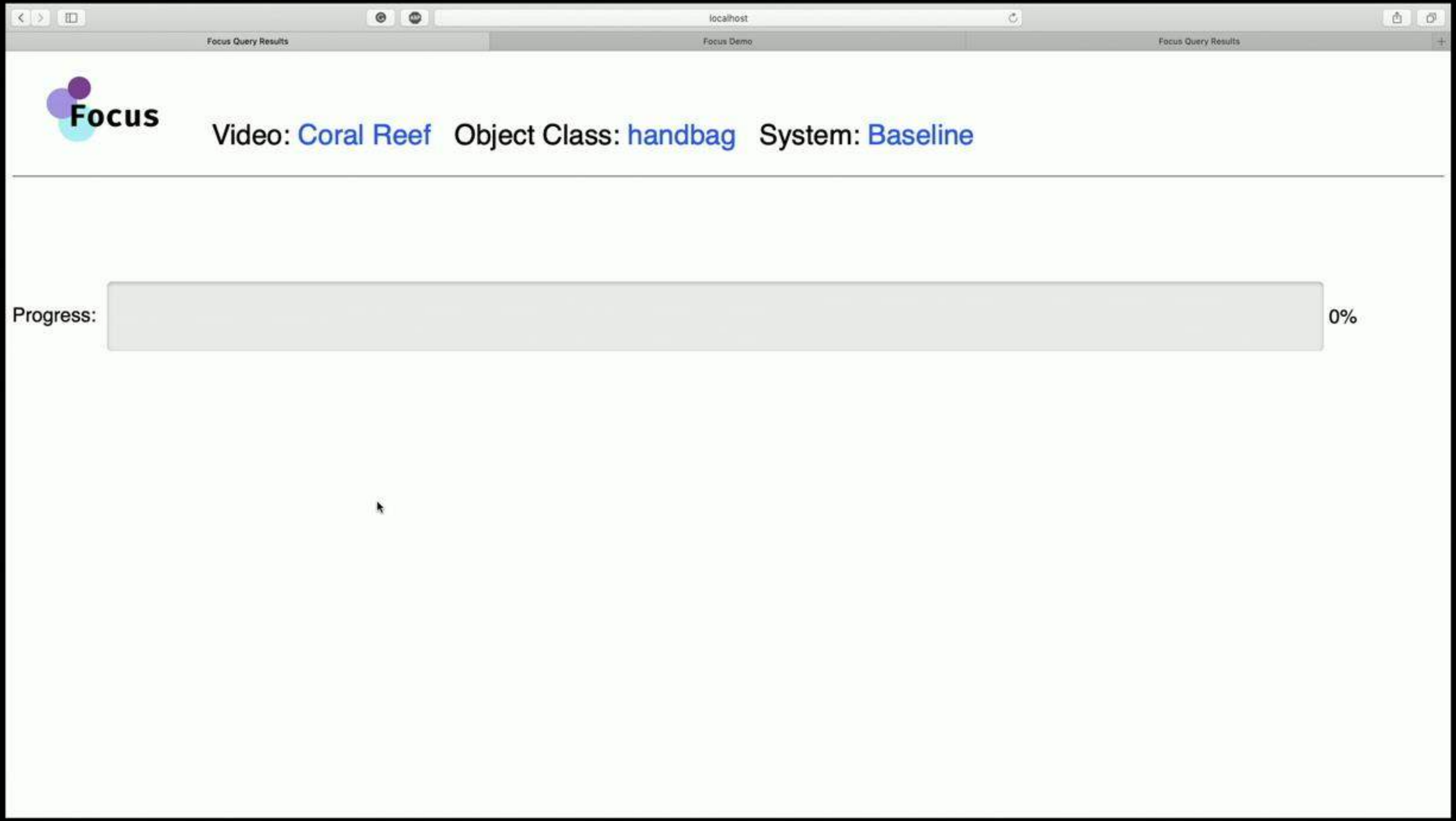


Focus

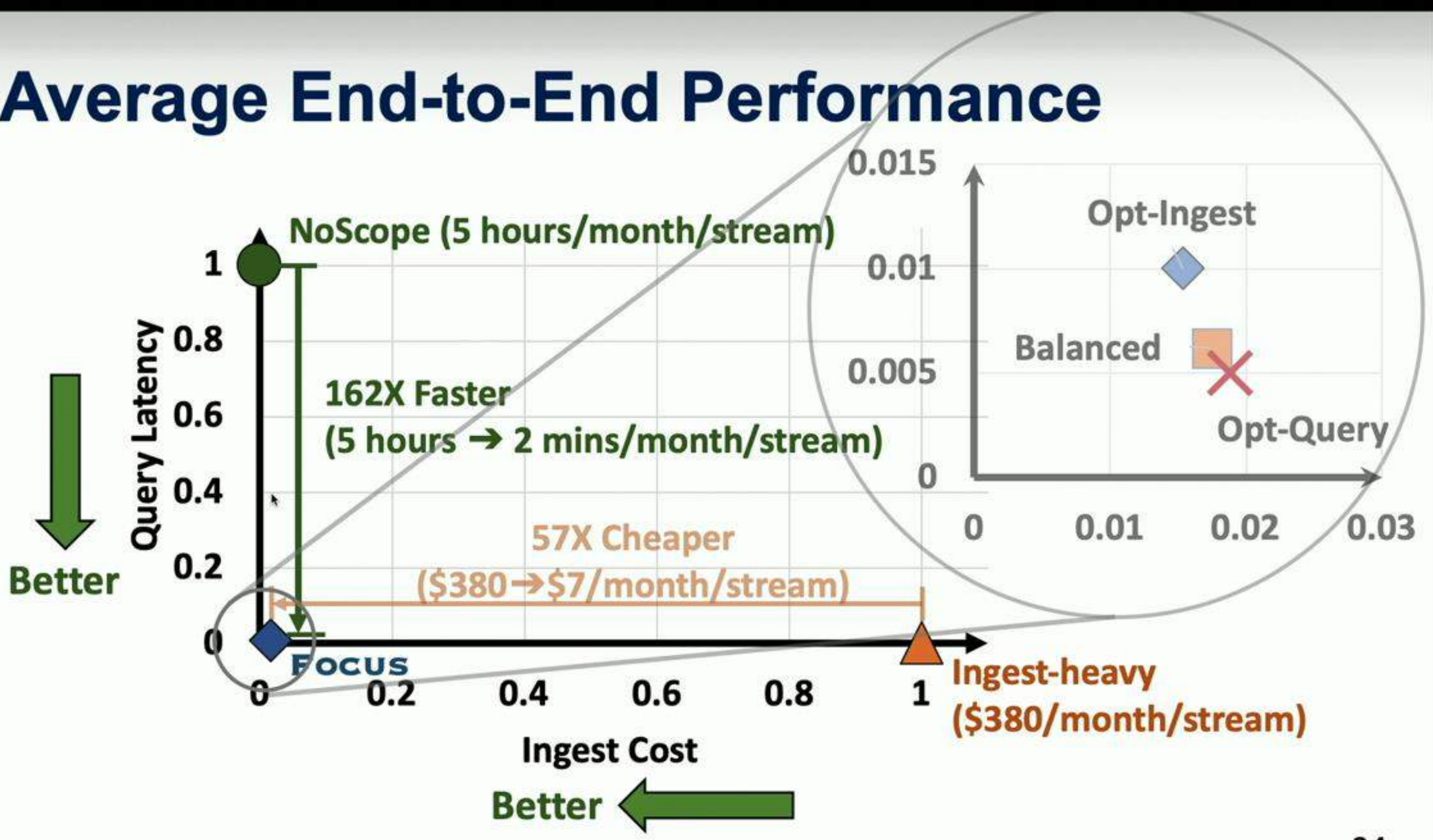
Video: Coral Reef Object: handbag

Focus Baseline Run

Video Duration: 4:59:50


localhost
Focus Query Results
Focus Demo
Focus Query Results
Focus
Video: Coral Reef Object Class: handbag System: Baseline
Progress:
0%

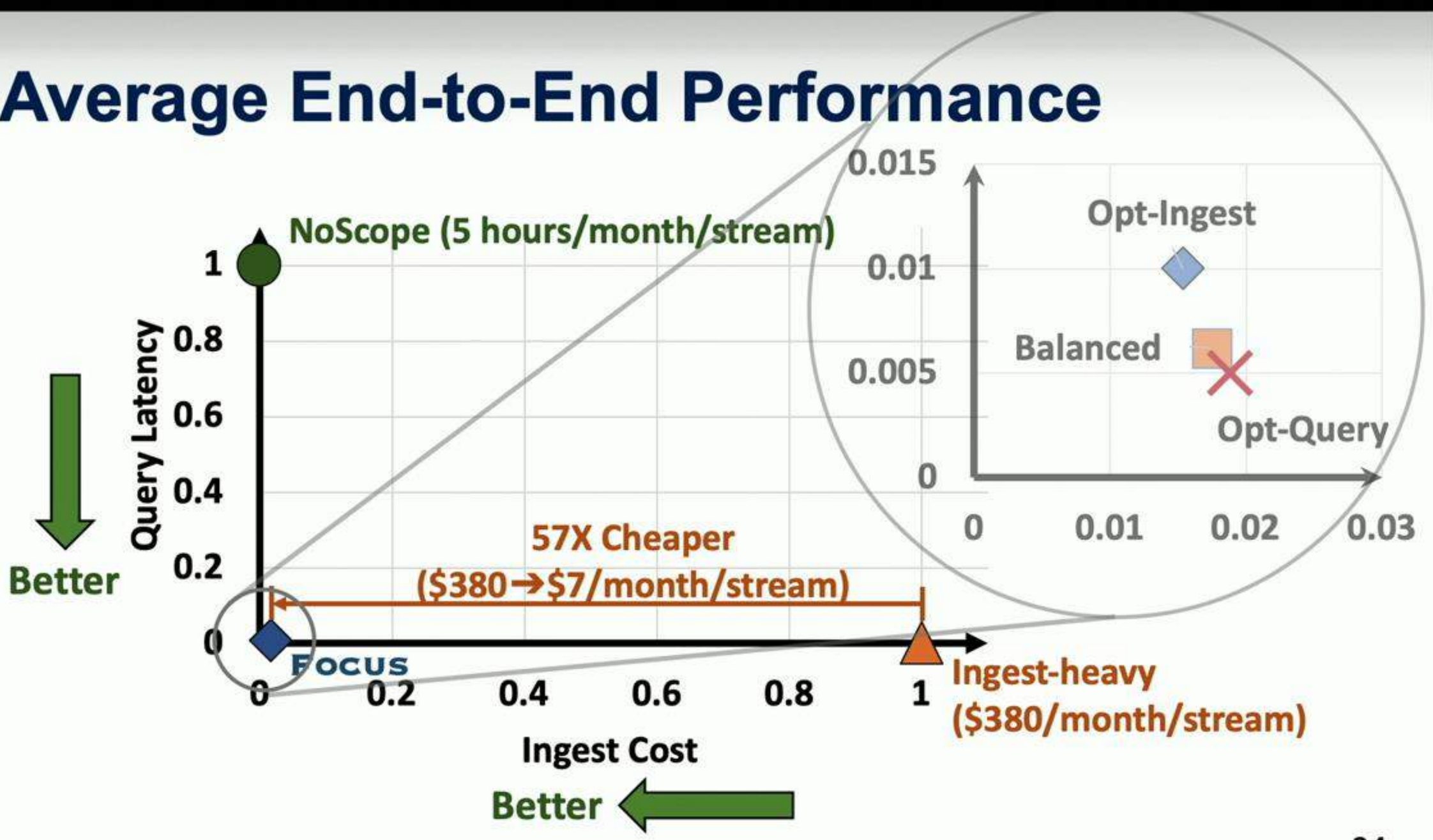
Average End-to-End Performance



Experimental Setup

- **Video Datasets**
  - 11 live traffic and enterprise videos
  - Each video stream is evaluated for 12 hours
- **Accuracy Targets**
  - 99% recall and 99% precision w.r.t. YOLOv2
- **Baselines**
  - Ingest-heavy: Analyzes all frames with YOLOv2 at ingest time and stores the inverted index for query
  - NoScope [VLDB'17]: A query-optimized system that analyzes frames only at query time

Average End-to-End Performance



Other Applications

Process large and growing data with CNNs to answer "after the fact" queries

Other Video Apps
- Face Recognition
- Emotion Detection

Audio
find audio segments with a word

Bioinformatics
- Brain signals
- Medical images

Geoinformatics
- Satellite images
- …

In This Talk

Focus: ML Serving for Rapidly Growing Data [OSDI'18]

Gaia: ML Training for Geo-Distributed Data [NSDI'17]

On-going and Future Work

Other Applications

Process large and growing data with CNNs to answer "after the fact" queries

Other Video Apps
- Face Recognition
- Emotion Detection

Audio
find audio segments with a word

Bioinformatics
- Brain signals
- Medical images

Geoinformatics
- Satellite images
- …

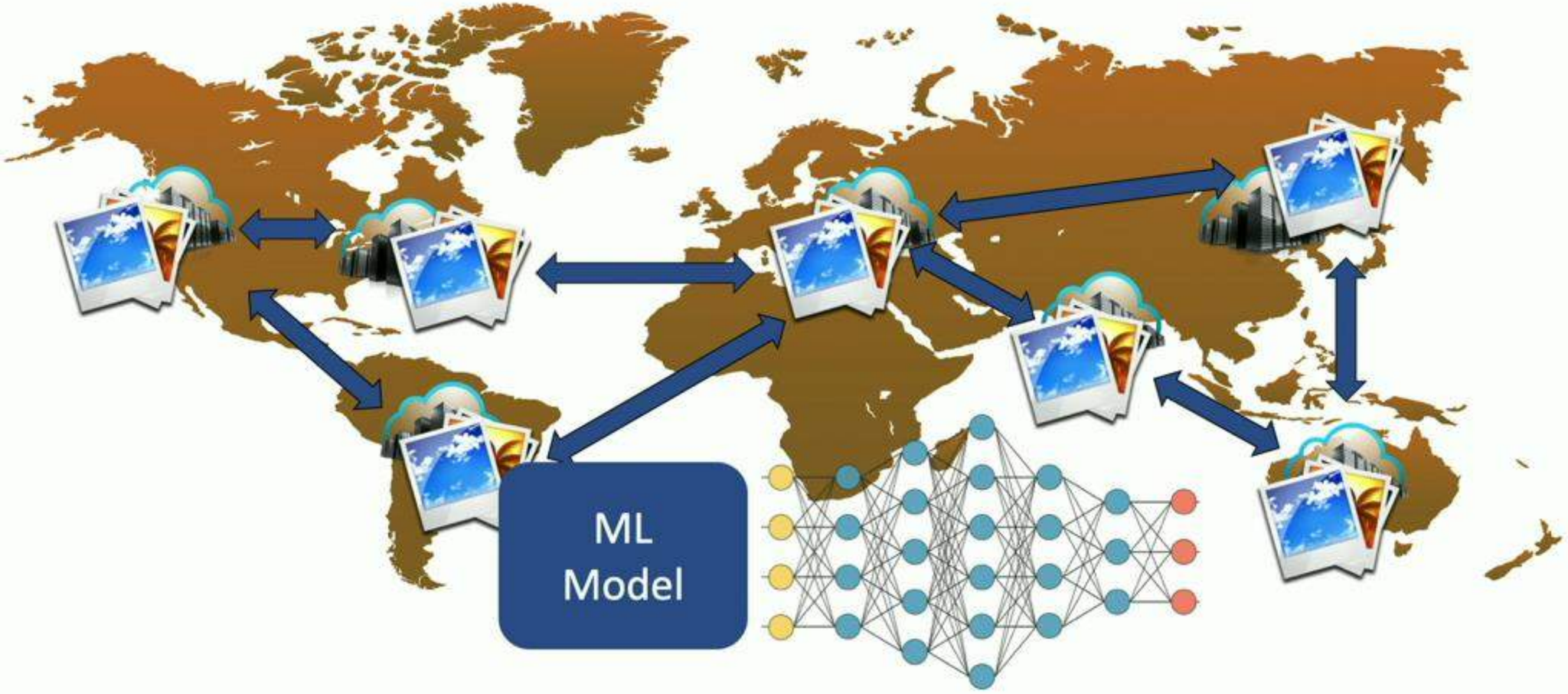
In This Talk

Focus: ML Serving for Rapidly Growing Data [OSDI'18]

Gaia: ML Training for Geo-Distributed Data [NSDI'17]

On-going and Future Work

ML Training on Geo-Distributed Data



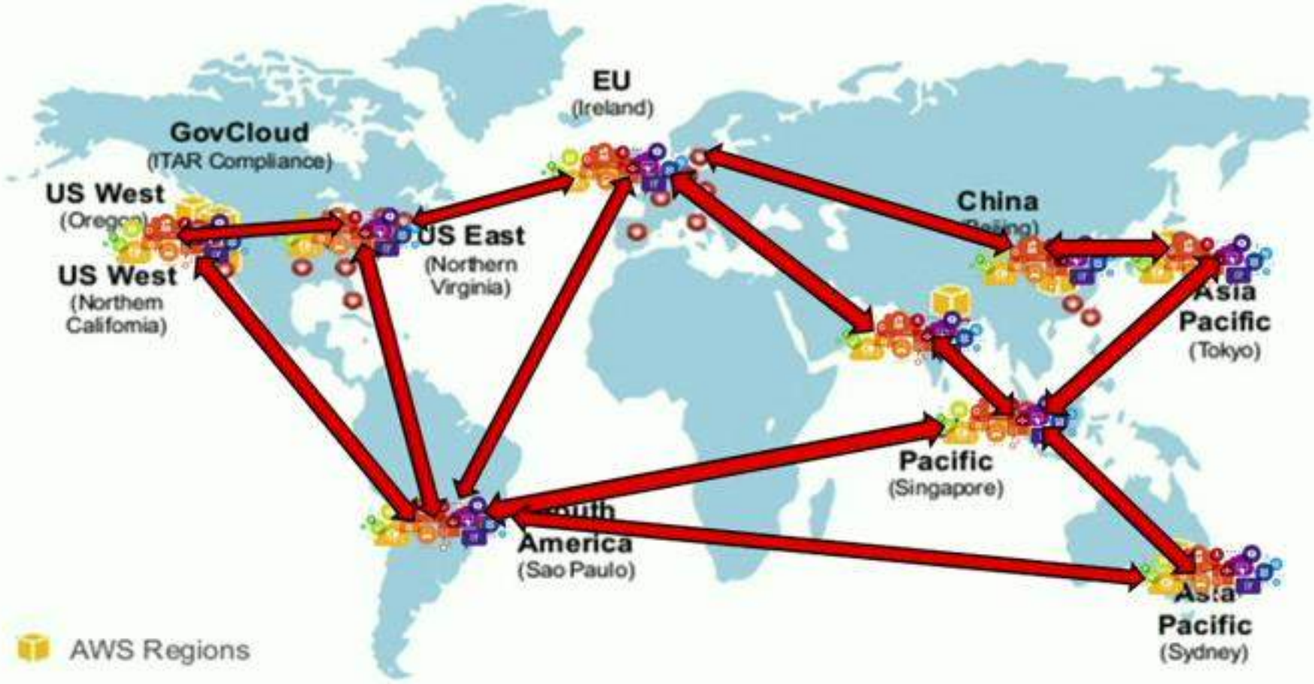
Centralizing Data is Infeasible [1, 2, 3]

- Moving data over wide-area networks (WANs) can be extremely slow
- It is also subject to data sovereignty laws

1. Vulimiri et al., NSDI'15
2. Pu et al., SIGCOMM'15
3. Viswanathan et al., OSDI'16

Geo-distributed ML is Challenging

- No ML system is designed to run across data centers (up to 53X slowdown in our study)

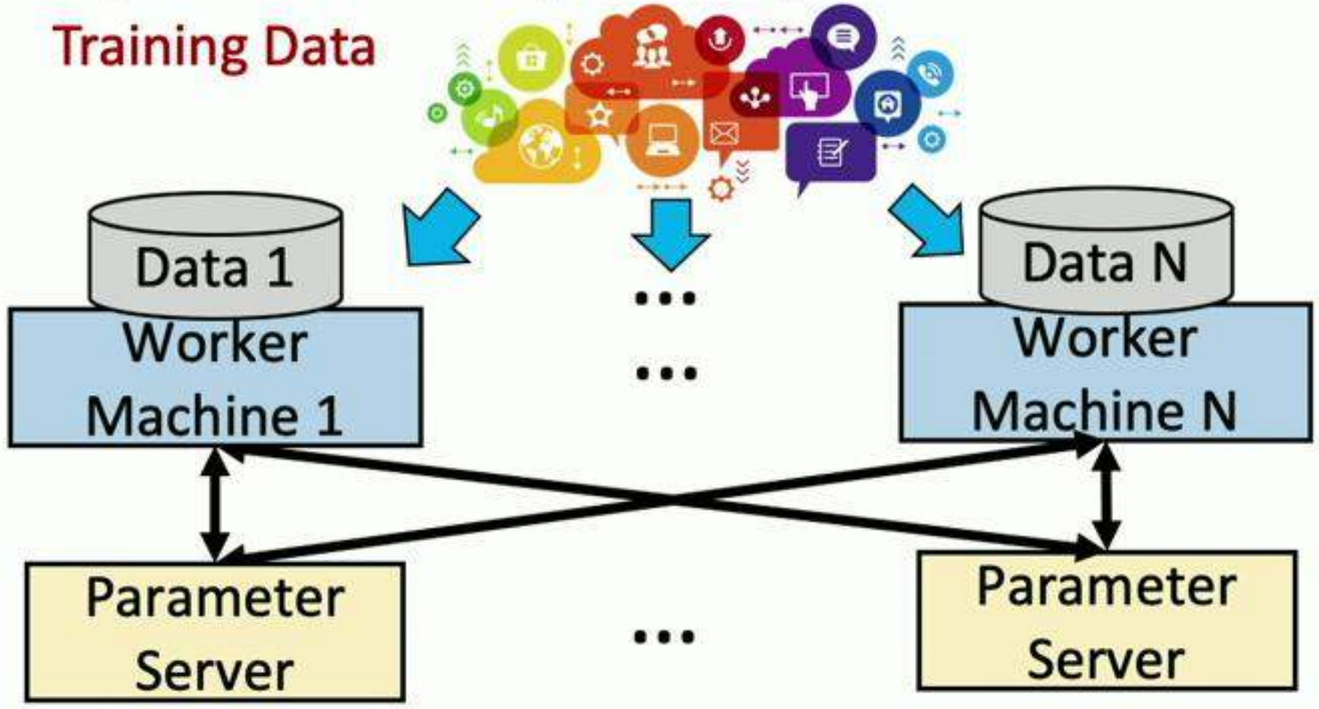


Our Goal

- Develop a geo-distributed ML system
  - Minimize communication over wide-area networks
  - Retain the accuracy and correctness of ML algorithms
  - Without requiring changes to the algorithms

Background: Parameter Server Architecture

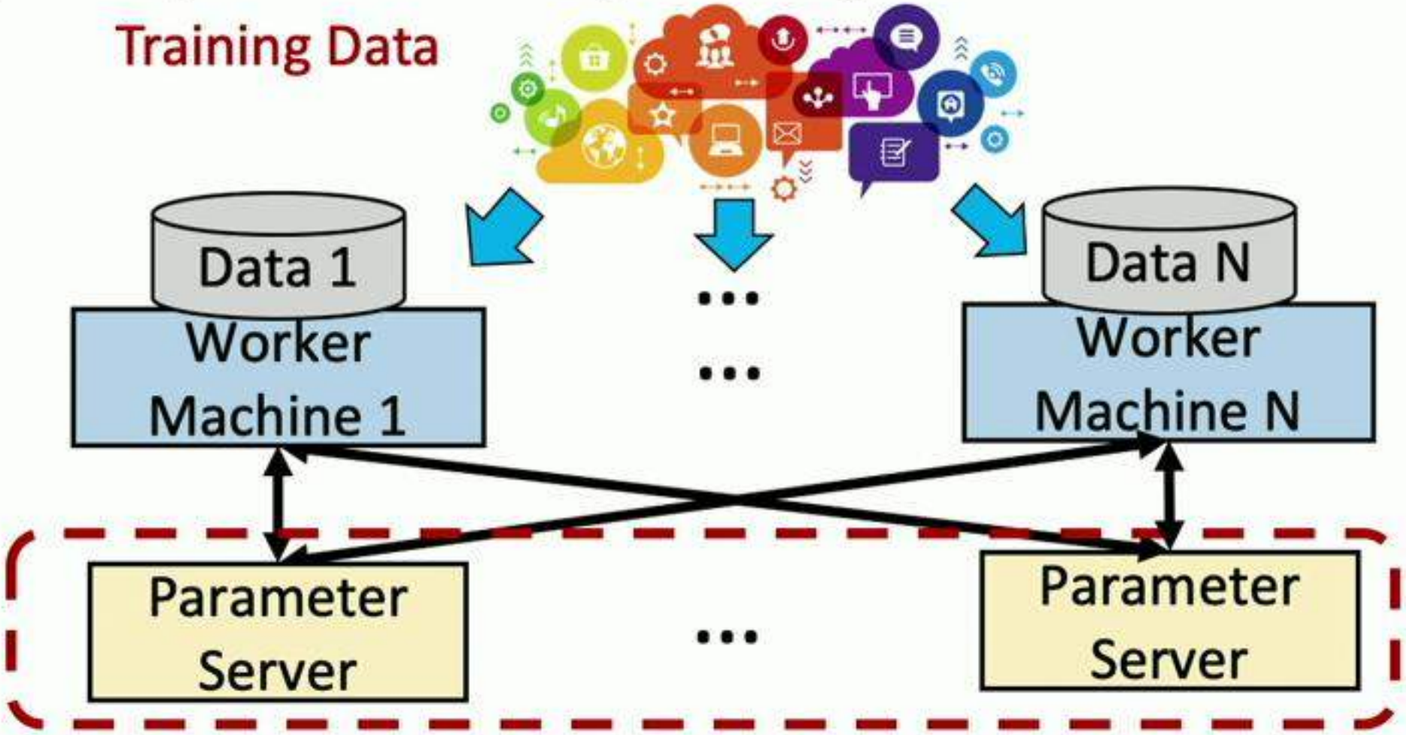
- The parameter server architecture has been widely adopted in many ML systems



ML Model

Background: Parameter Server Architecture

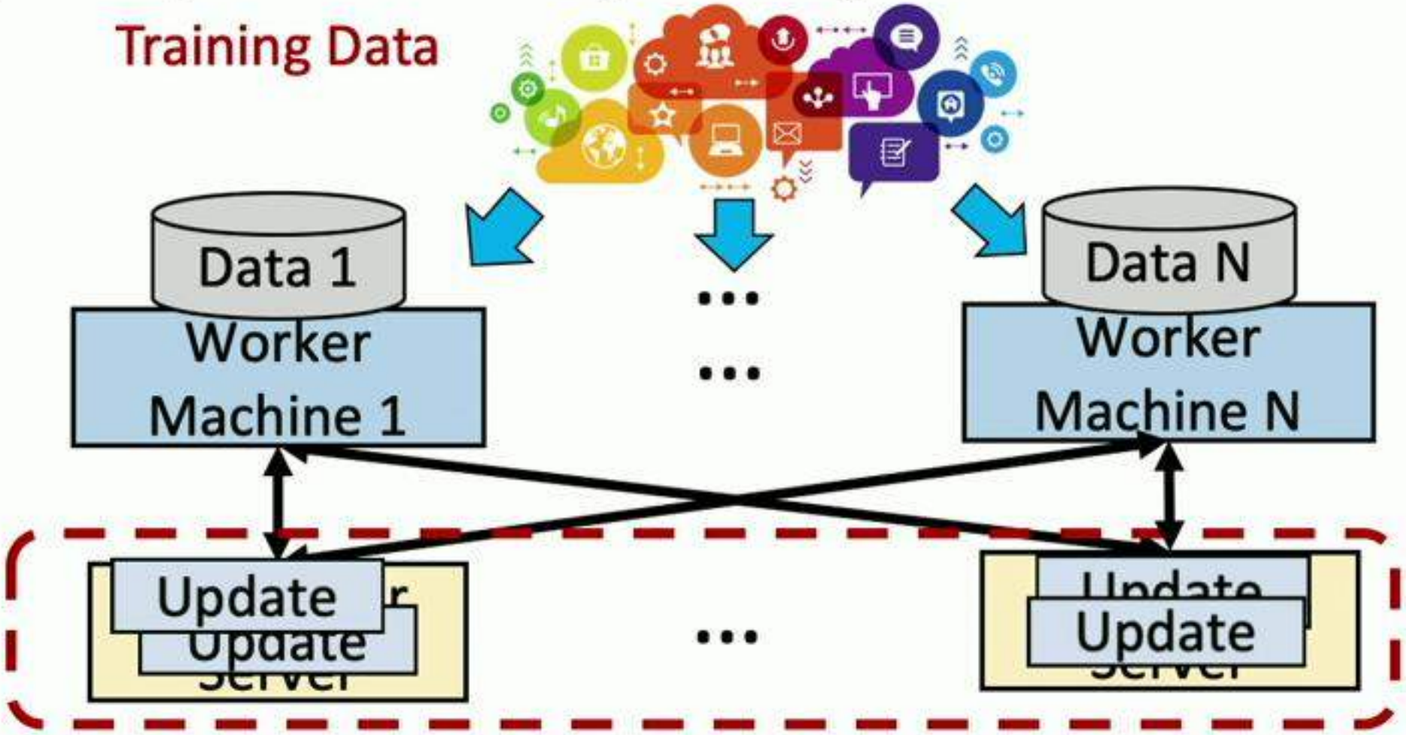
- The parameter server architecture has been widely adopted in many ML systems



ML Model

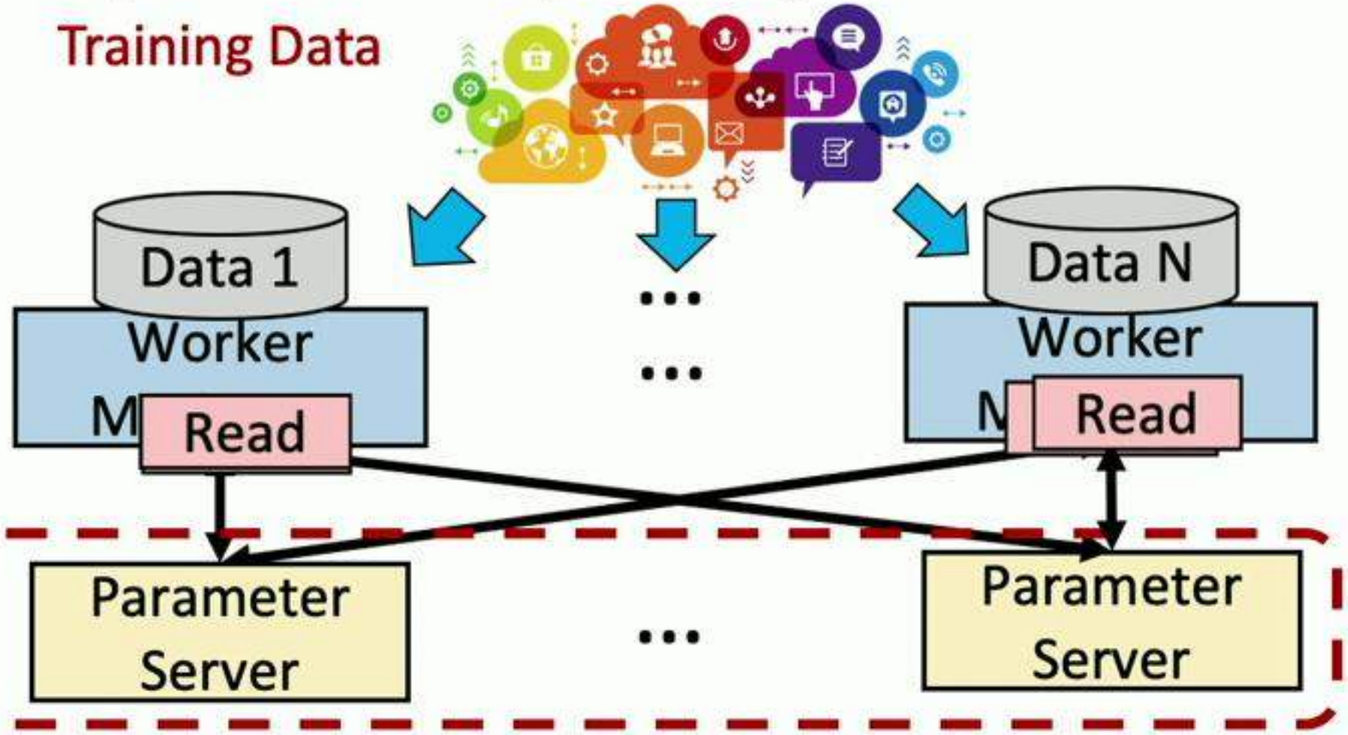
Background: Parameter Server Architecture

- The parameter server architecture has been widely adopted in many ML systems



Background: Parameter Server Architecture

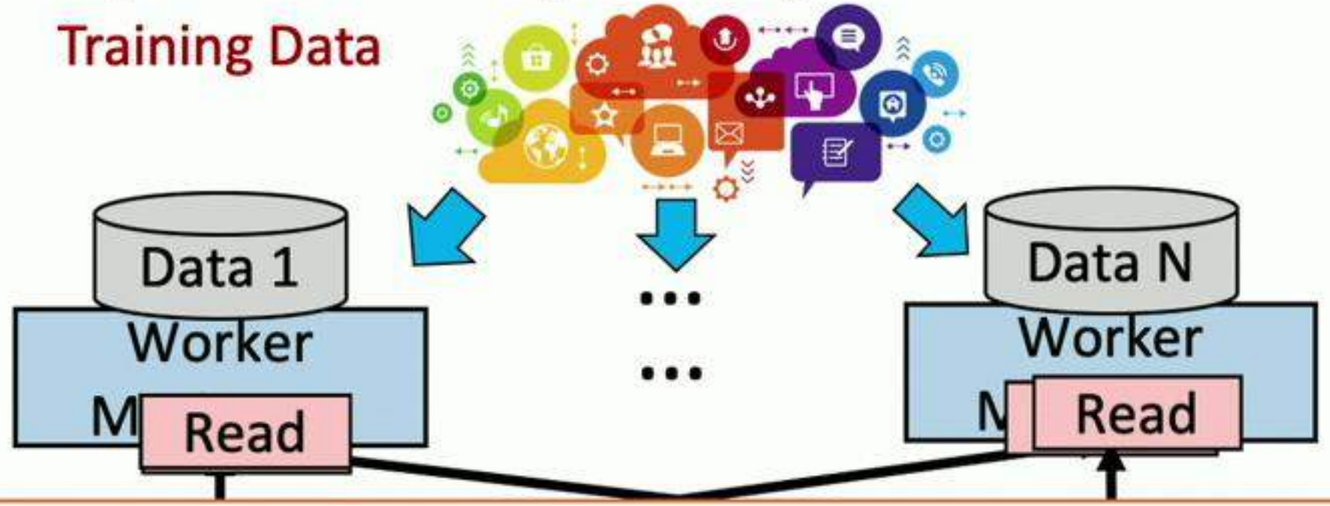
- The parameter server architecture has been widely adopted in many ML systems



ML Model

Background: Parameter Server Architecture

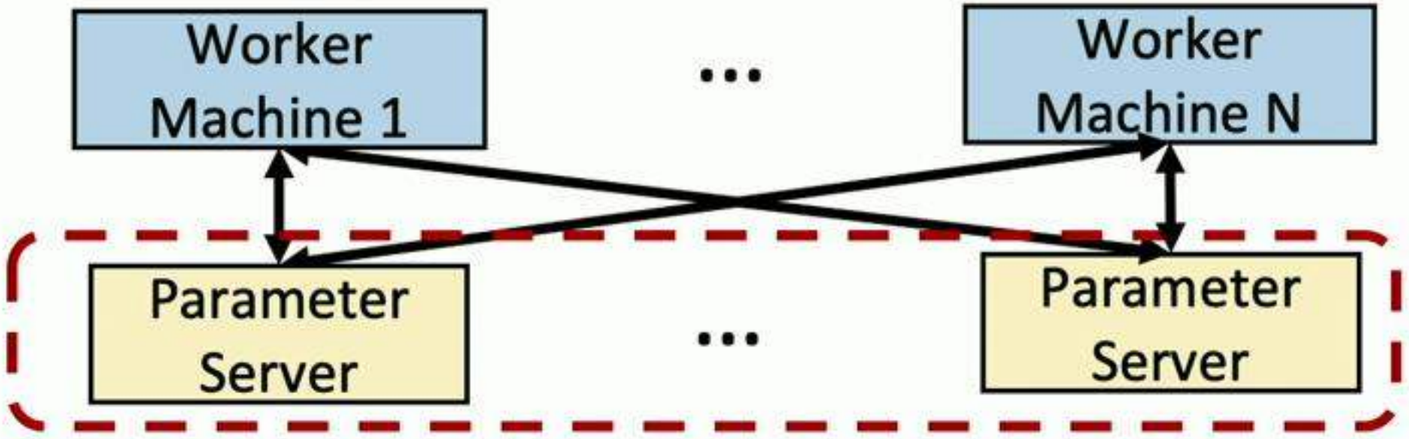
- The parameter server architecture has been widely adopted in many ML systems



Synchronization is critical to the accuracy and correctness of ML algorithms

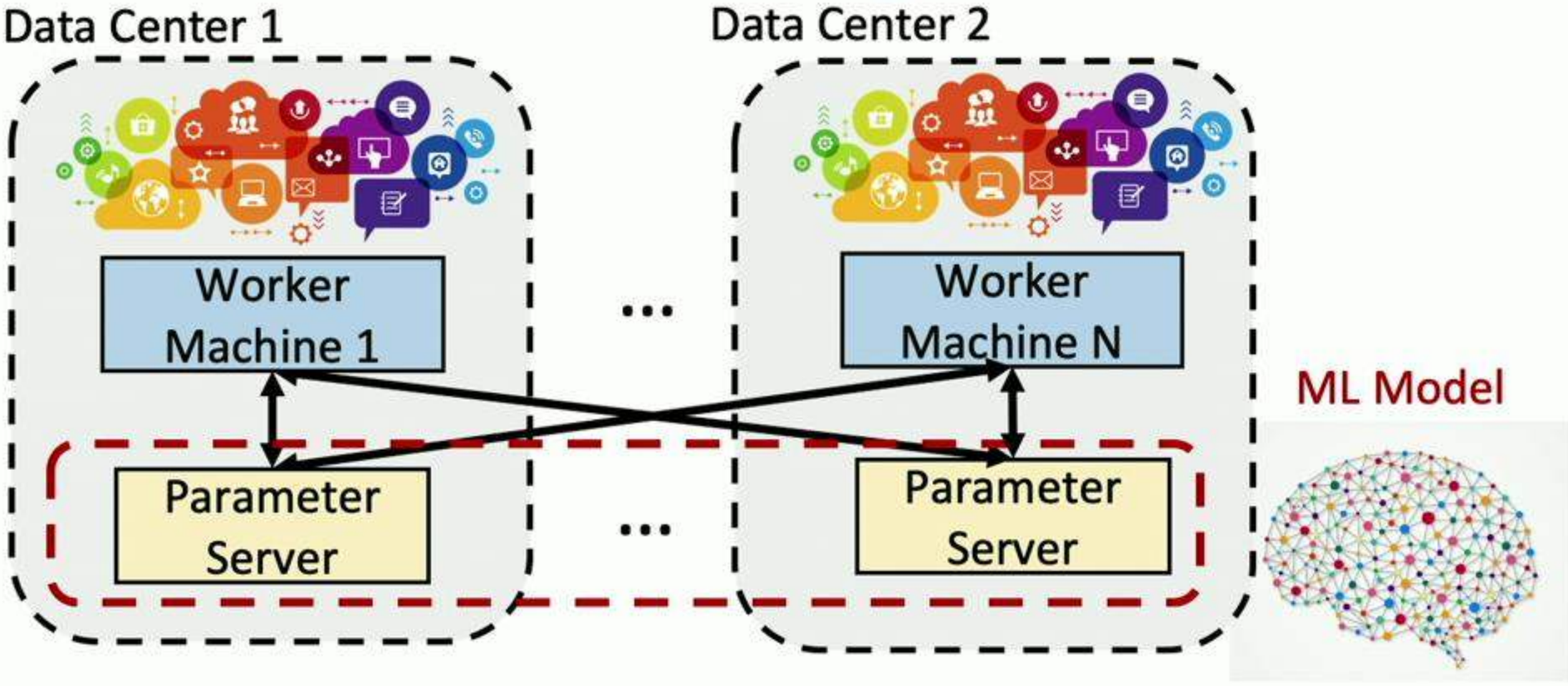
Deploy Parameter Servers on WANs

- Deploying parameter servers across data centers requires a lot of communication over WANs (up to 53X slowdown)



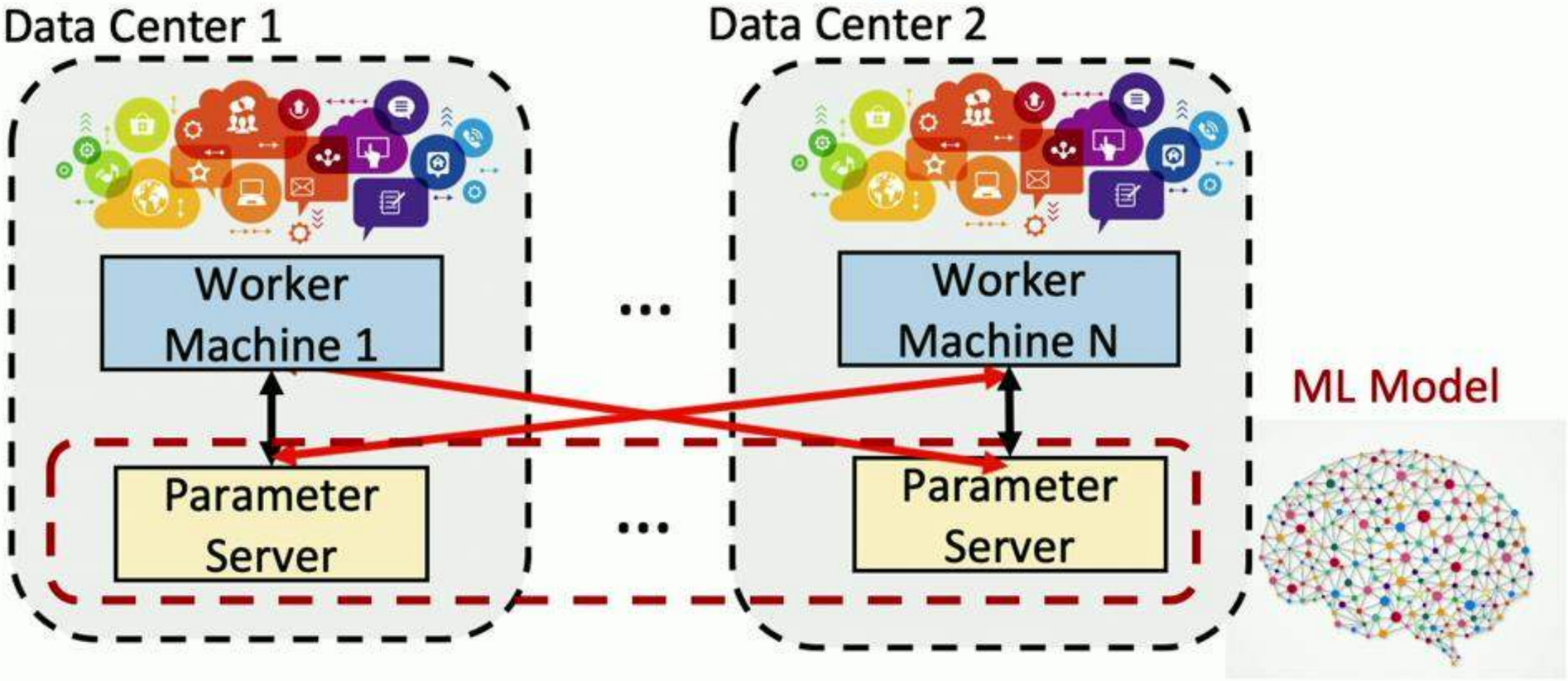
Deploy Parameter Servers on WANs

- Deploying parameter servers across data centers requires a lot of communication over WANs (up to 53X slowdown)



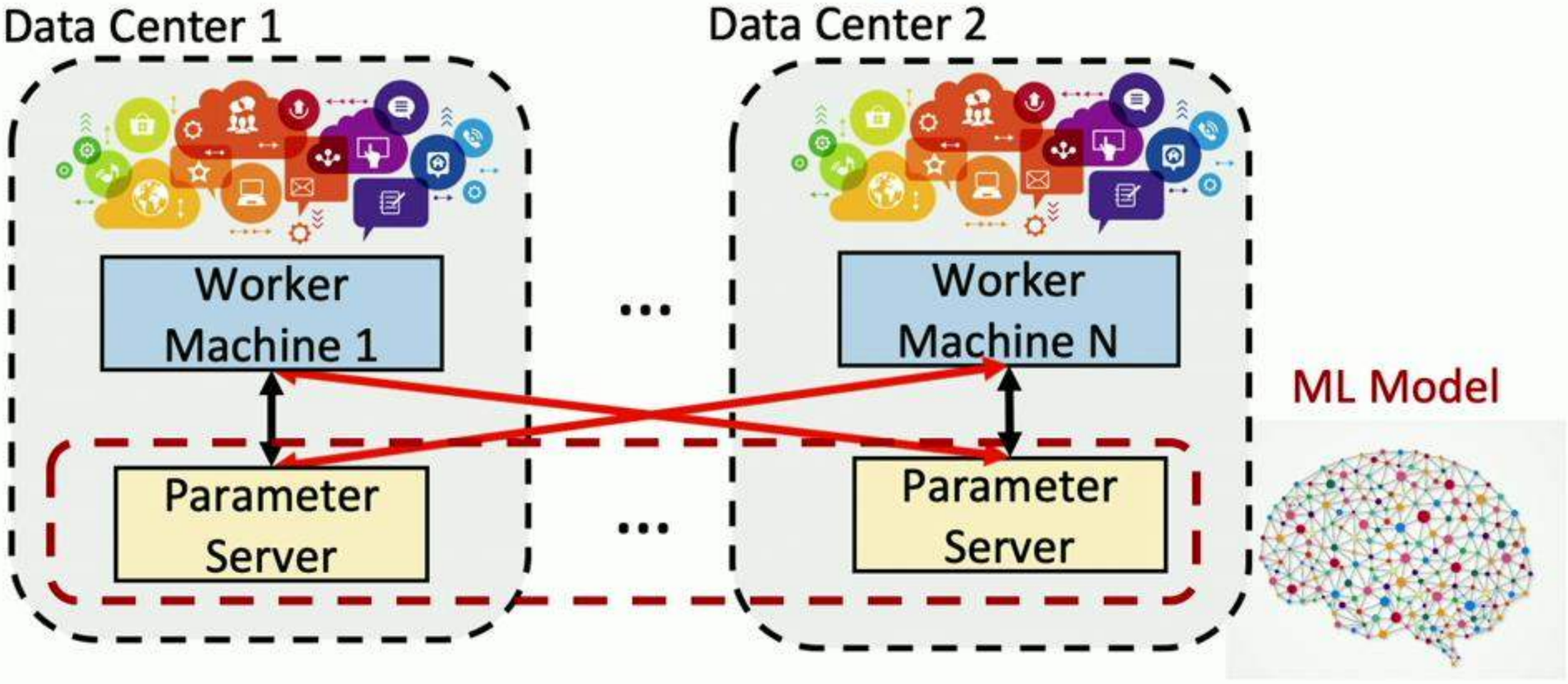
Deploy Parameter Servers on WANs

- Deploying parameter servers across data centers requires a lot of communication over WANs (up to 53X slowdown)



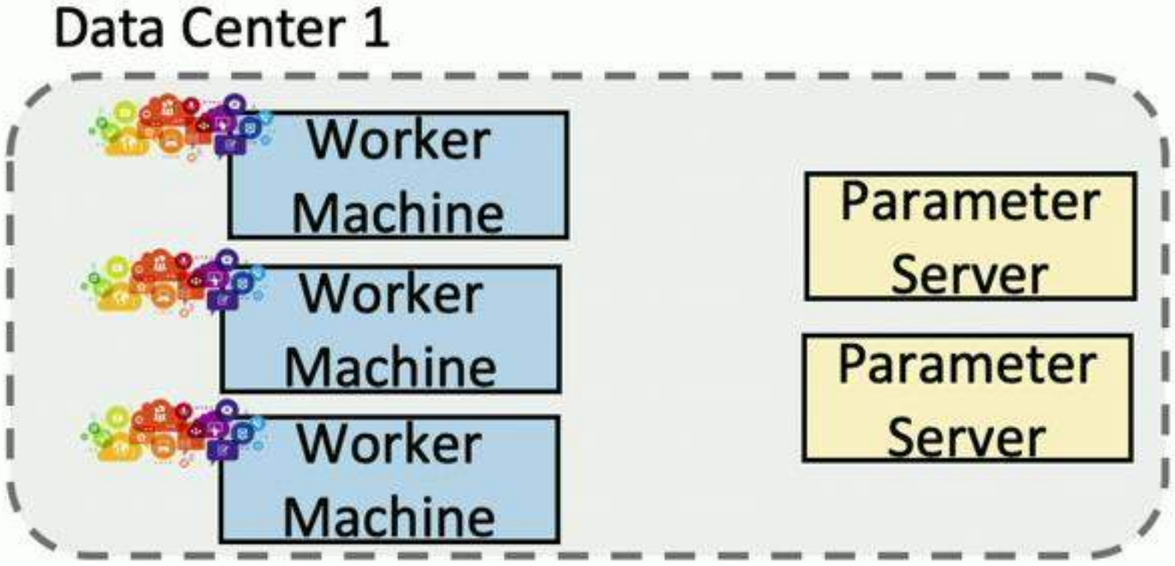
Deploy Parameter Servers on WANs

- Deploying parameter servers across data centers requires a lot of communication over WANs (up to 53X slowdown)



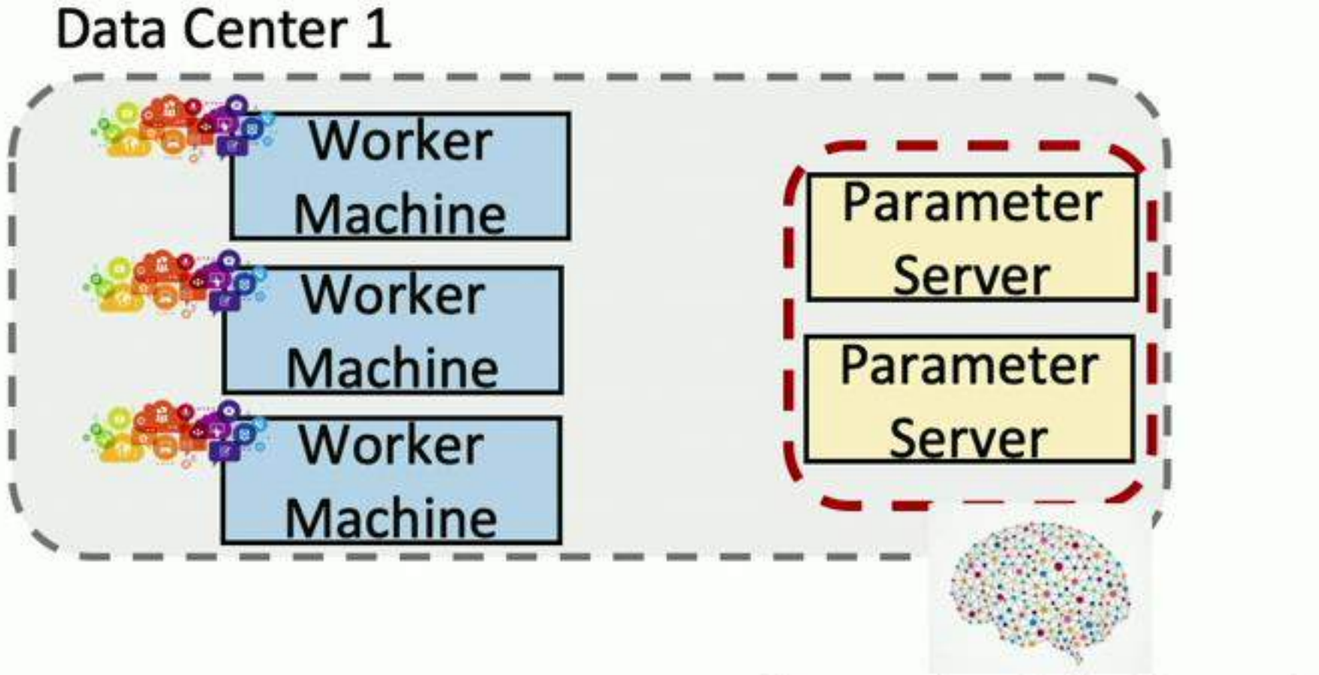
Gaia System Overview

- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers



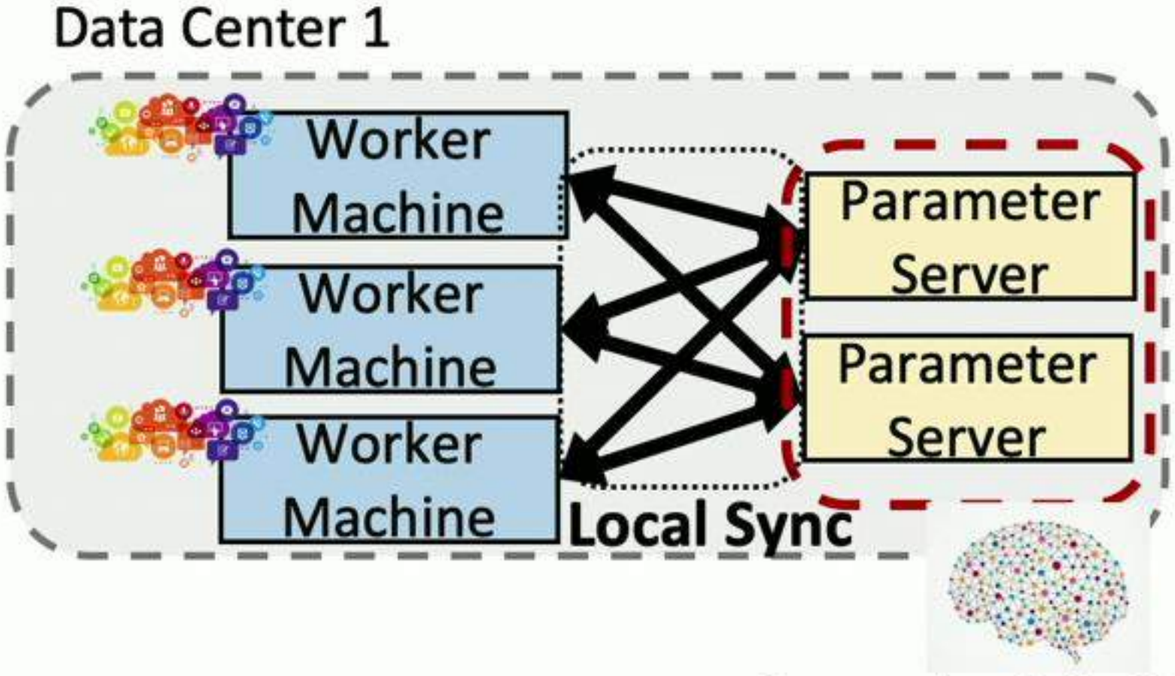
Gaia System Overview

- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers



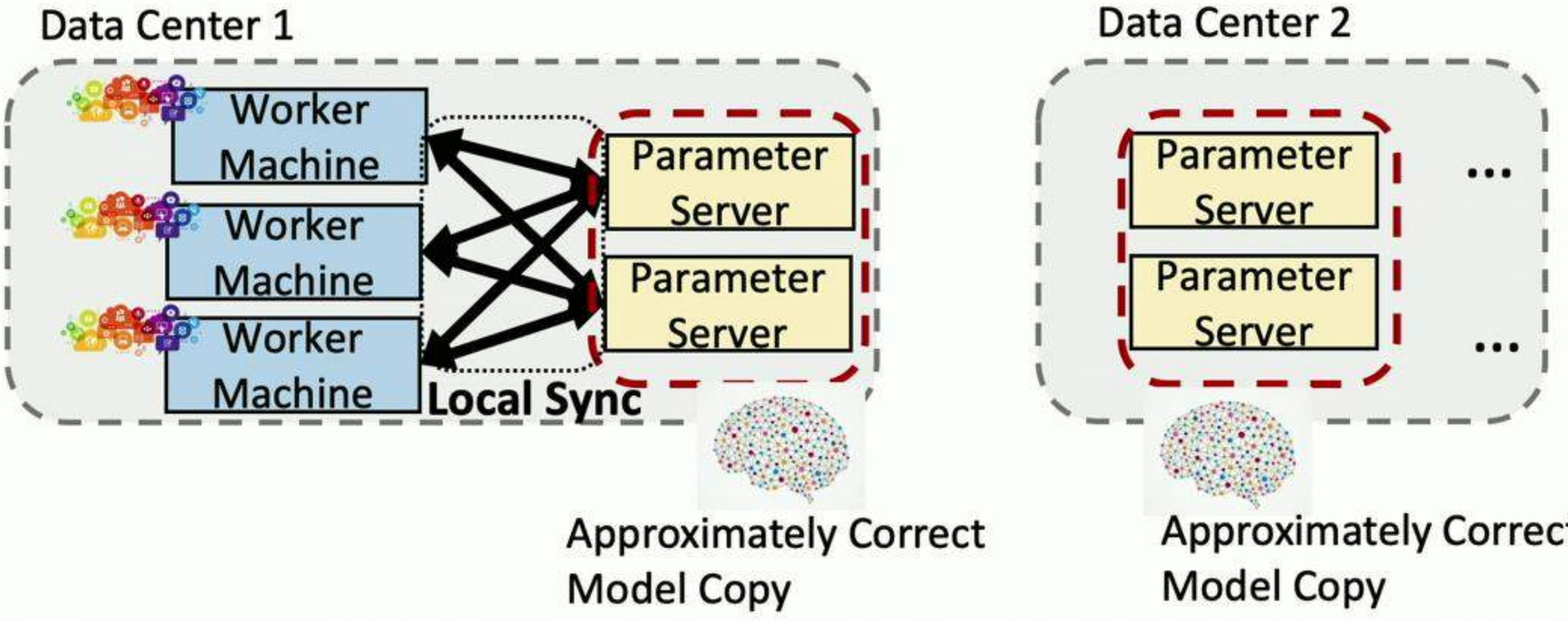
Gaia System Overview

- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers



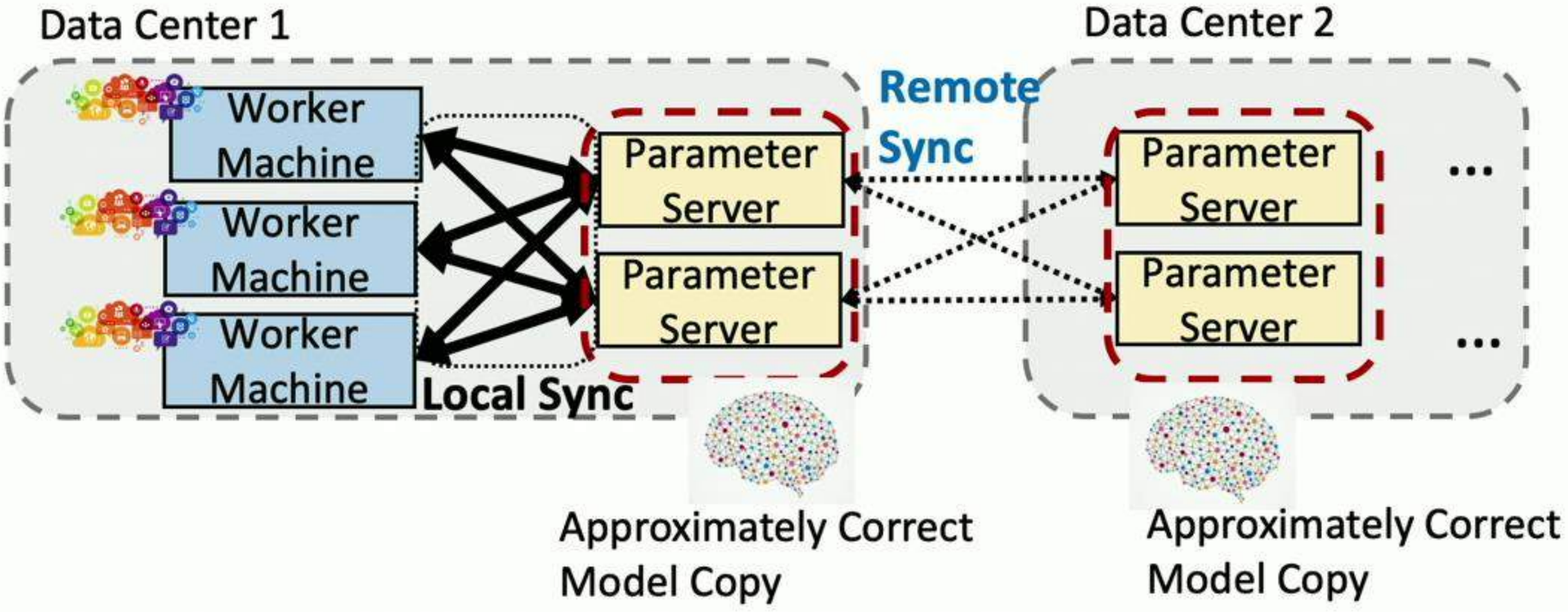
Gaia System Overview

- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers



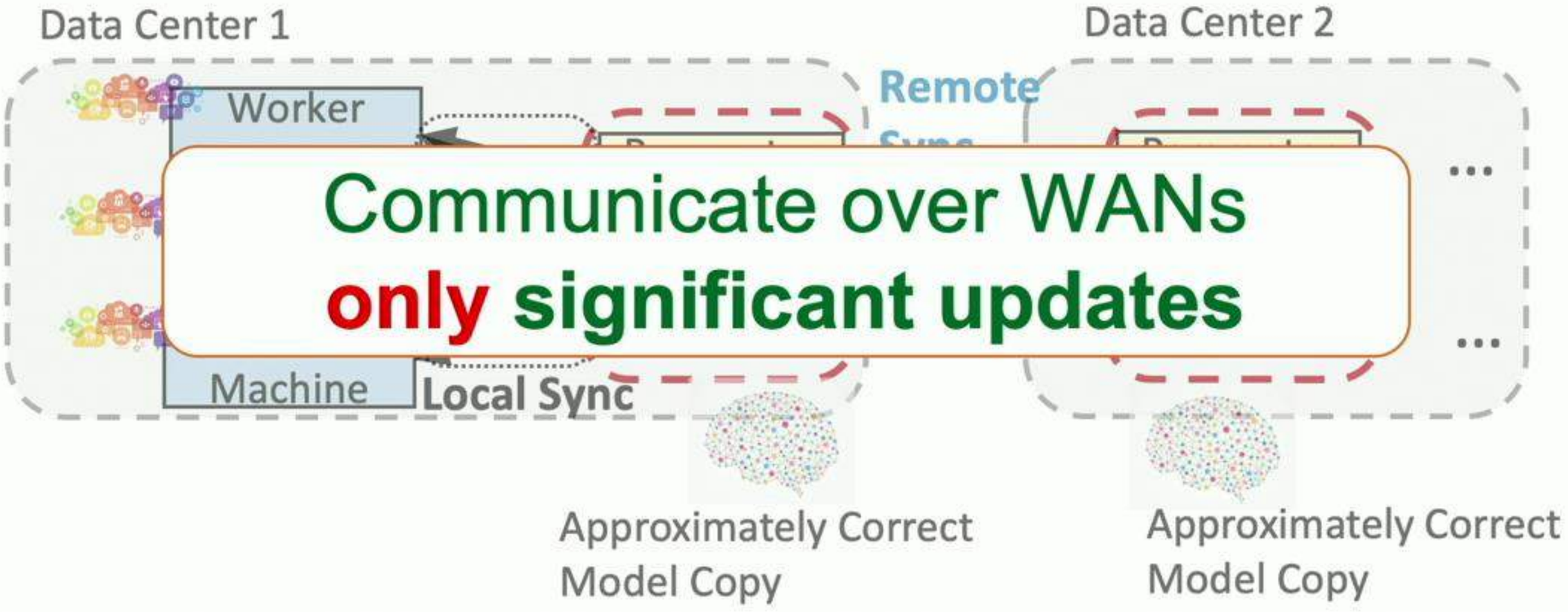
Gaia System Overview

- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers

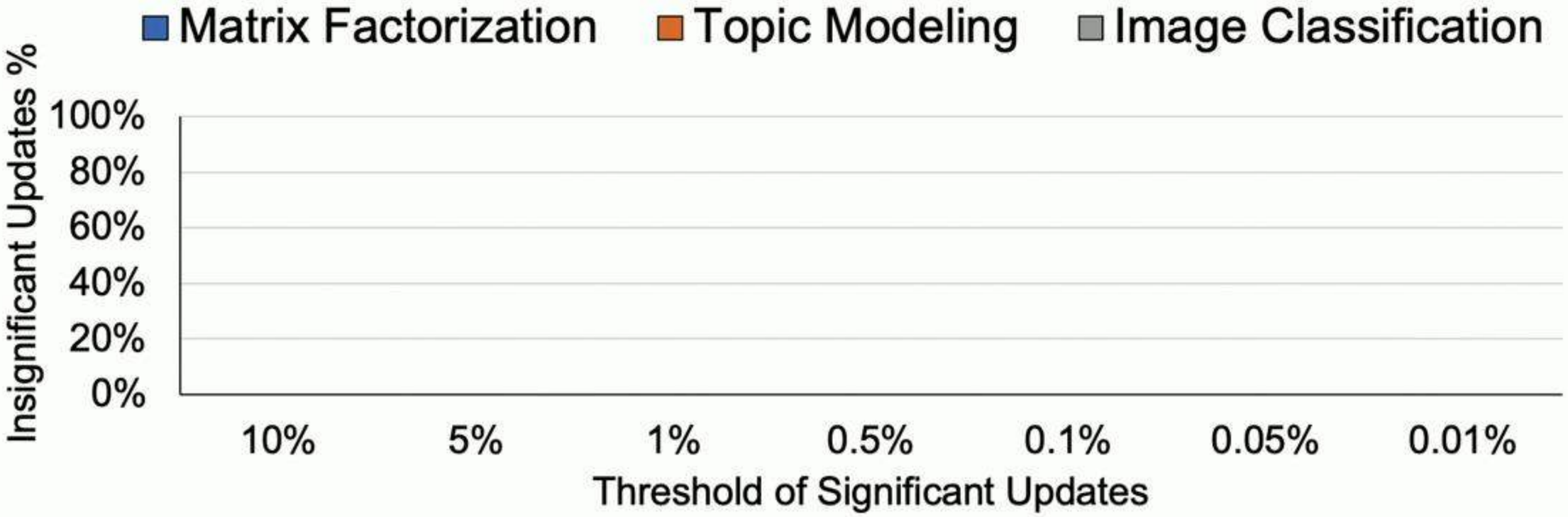


Gaia System Overview

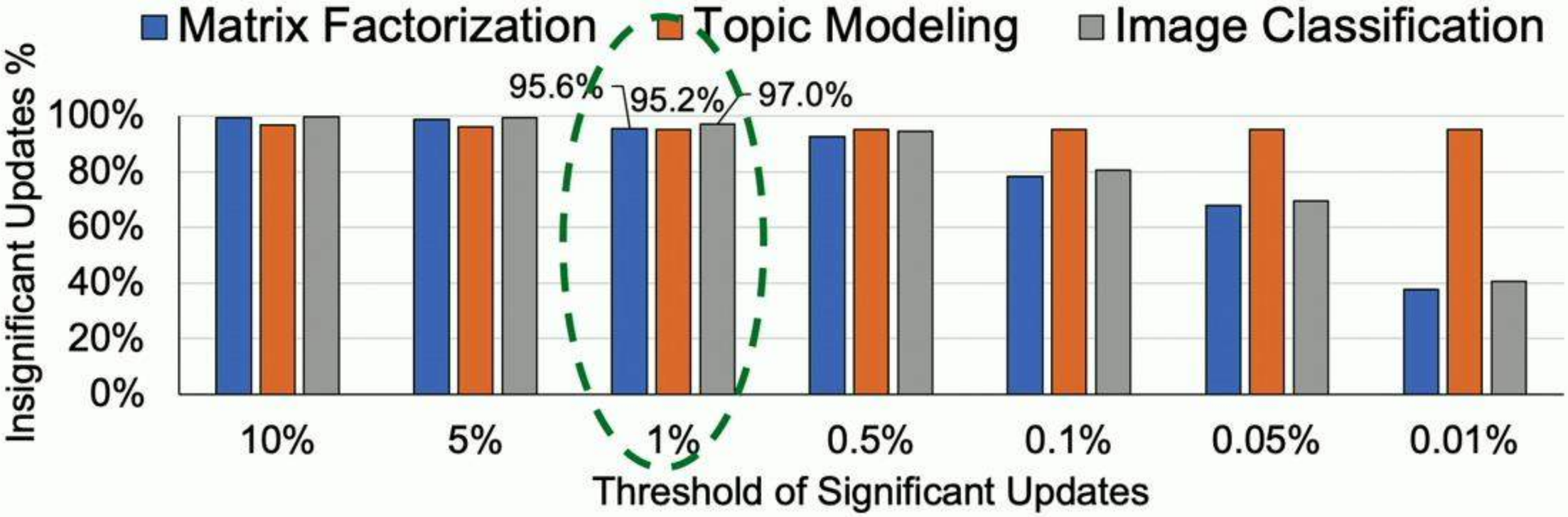
- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers



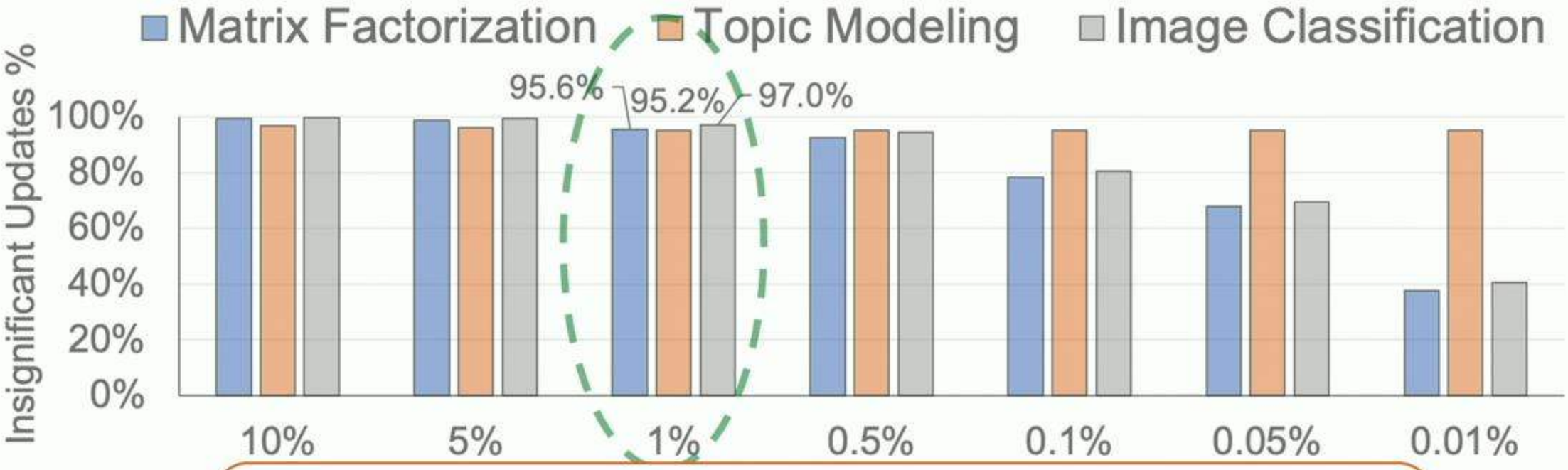
Key Finding: Study of Update Significance



Key Finding: Study of Update Significance



Key Finding: Study of Update Significance



The vast majority of updates are ***insignificant***

Approximate Synchronous Parallel

The significance filter

- Filter updates based on their significance

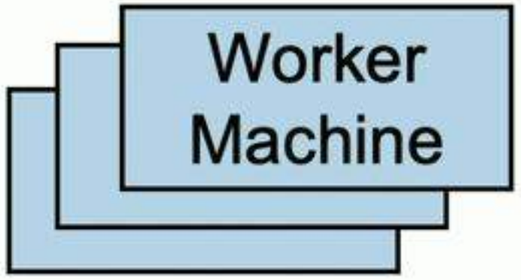
ASP selective barrier

- Ensure significant updates are read in time

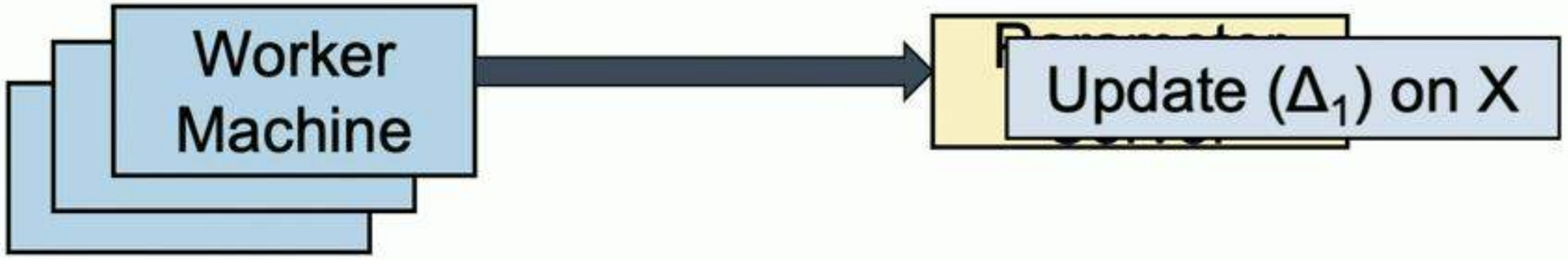
Mirror clock

- Safe guard for pathological cases

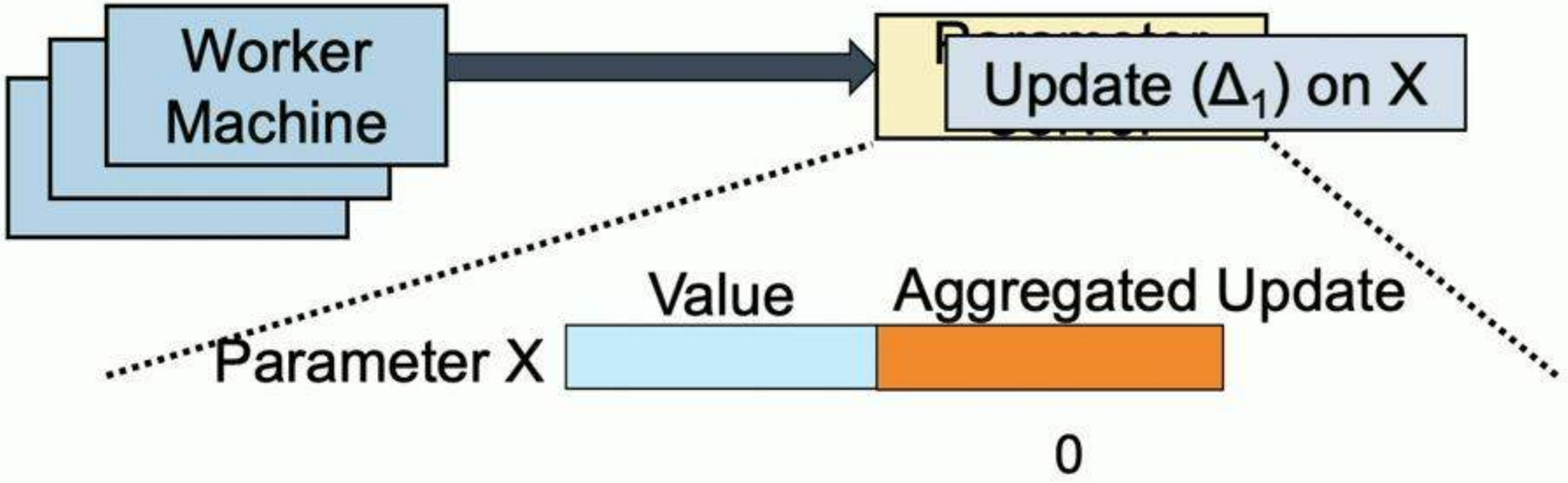
The Significance Filter



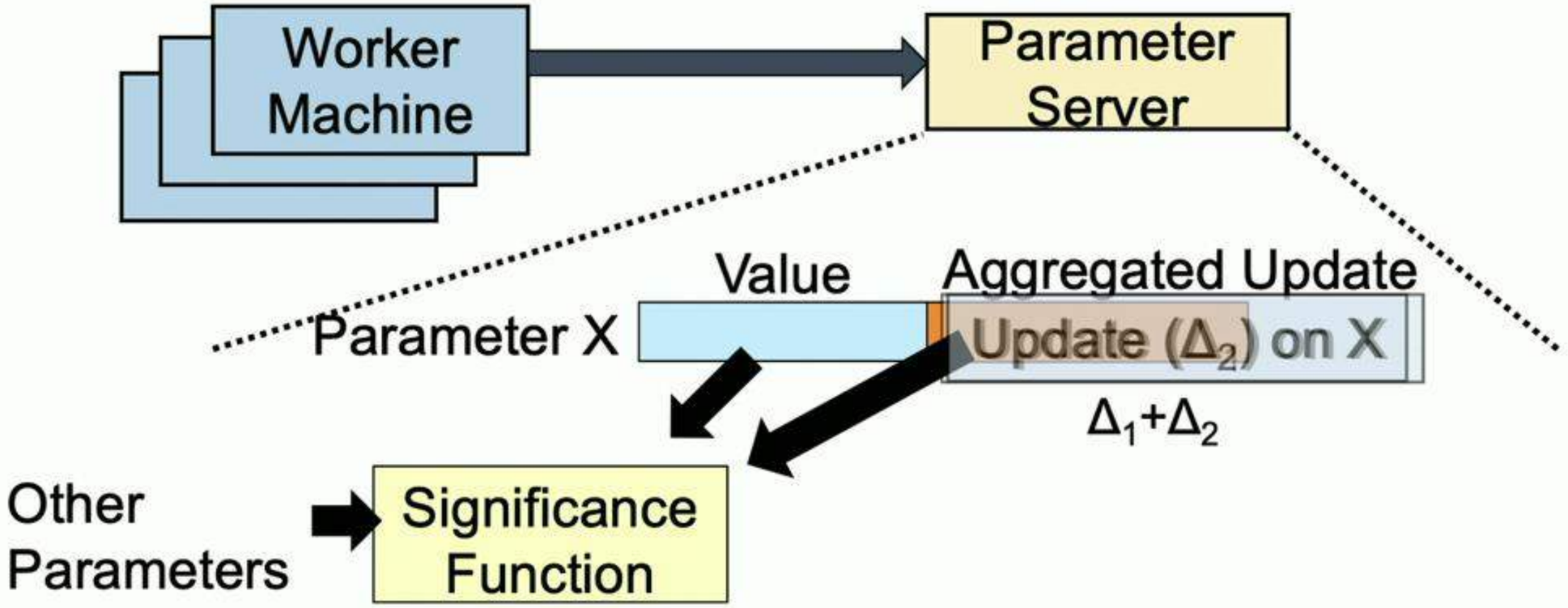
The Significance Filter



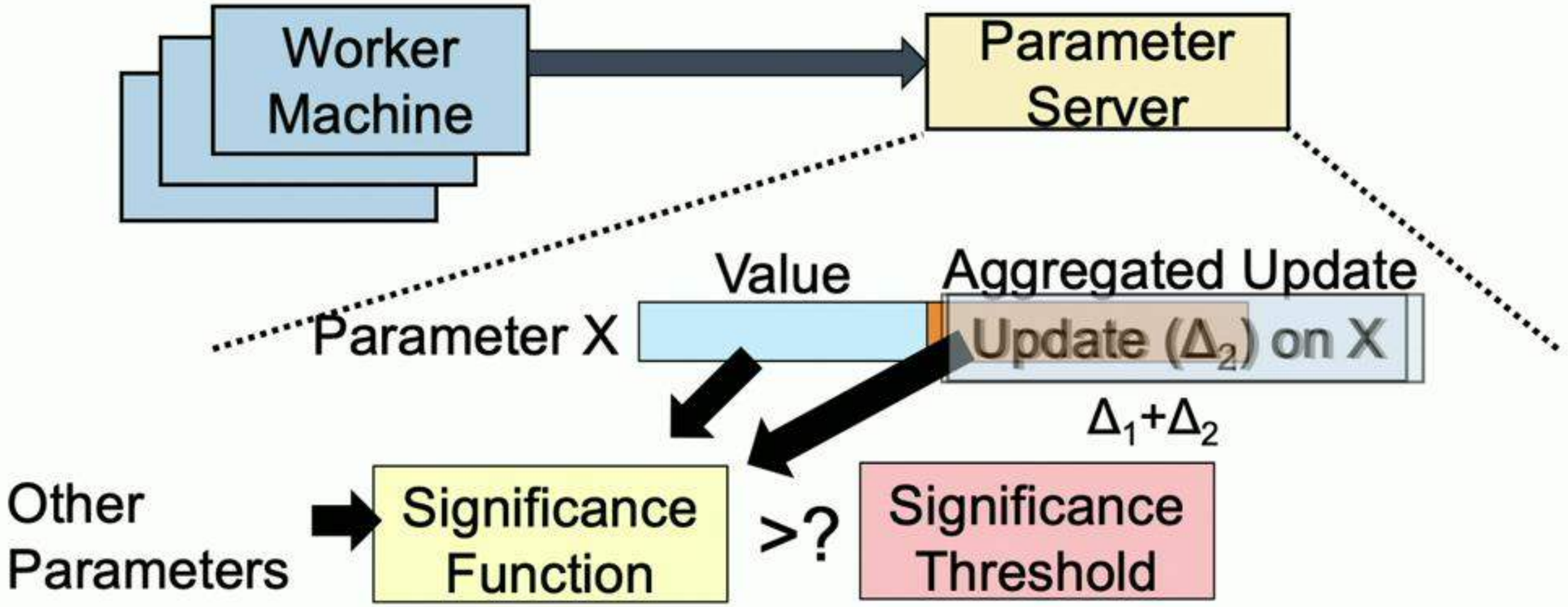
The Significance Filter



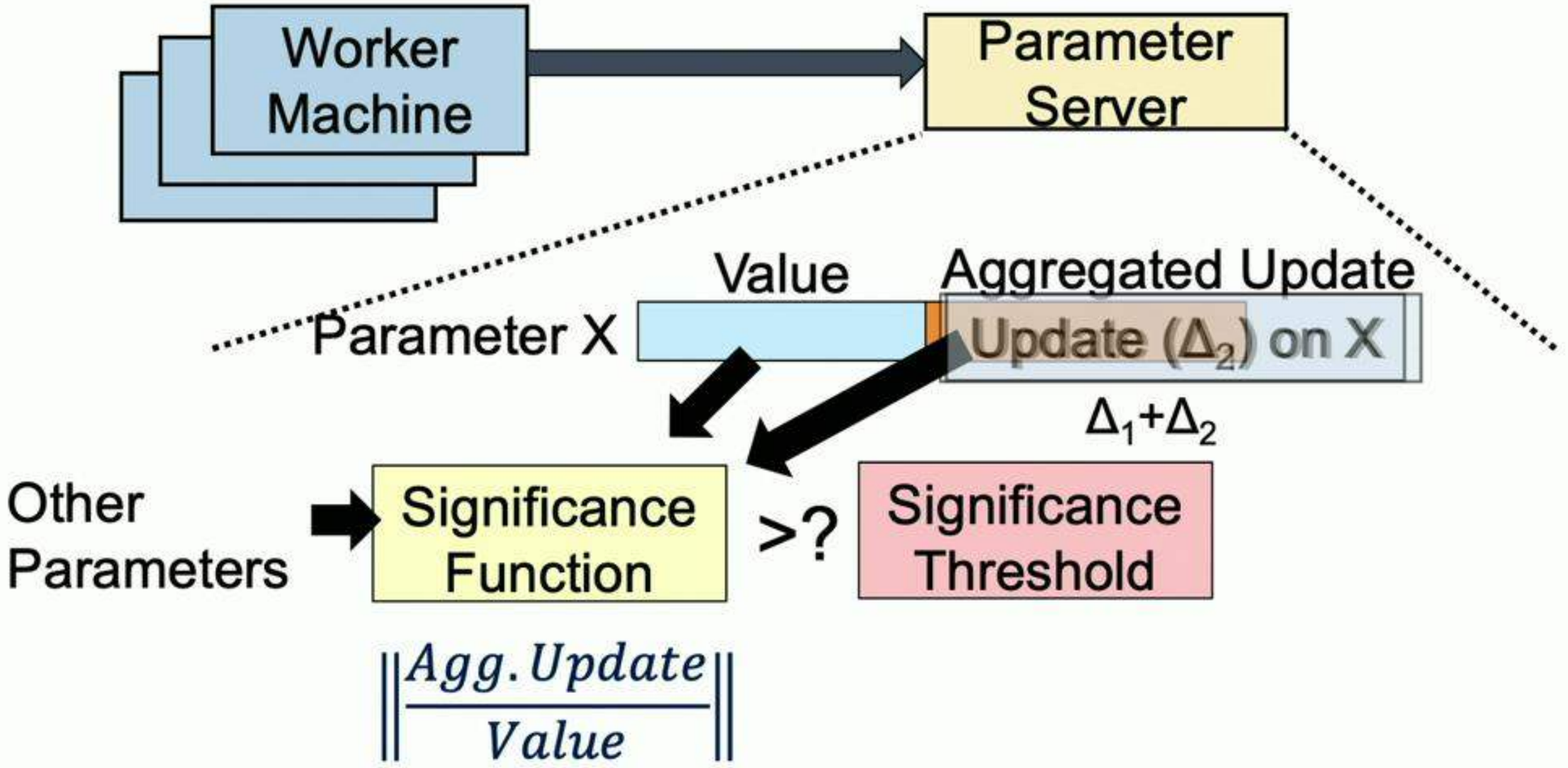
The Significance Filter



The Significance Filter



The Significance Filter



Approximate Synchronous Parallel

The significance filter

- Filter updates based on their significance

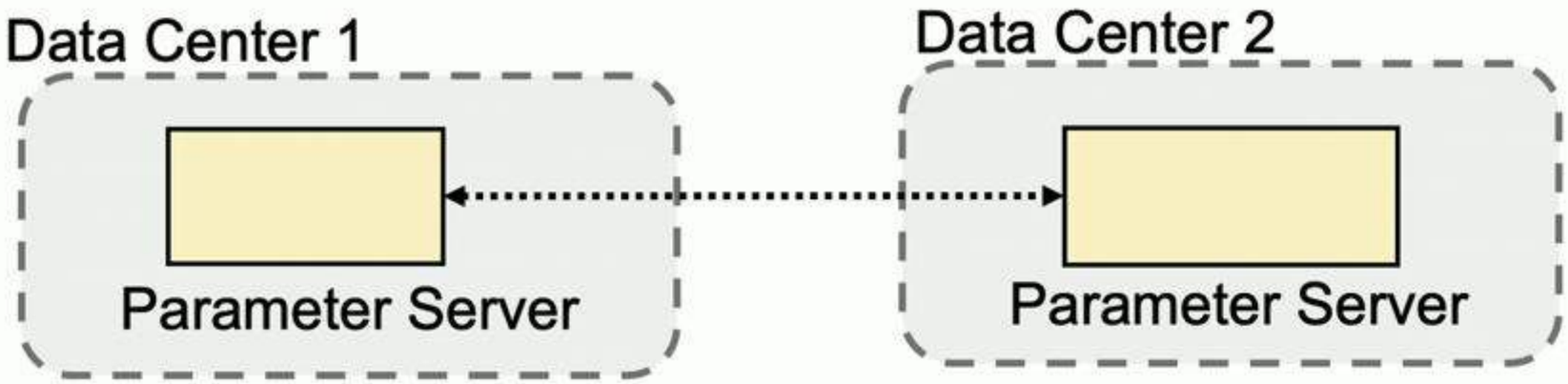
ASP selective barrier

- Ensure significant updates are read in time

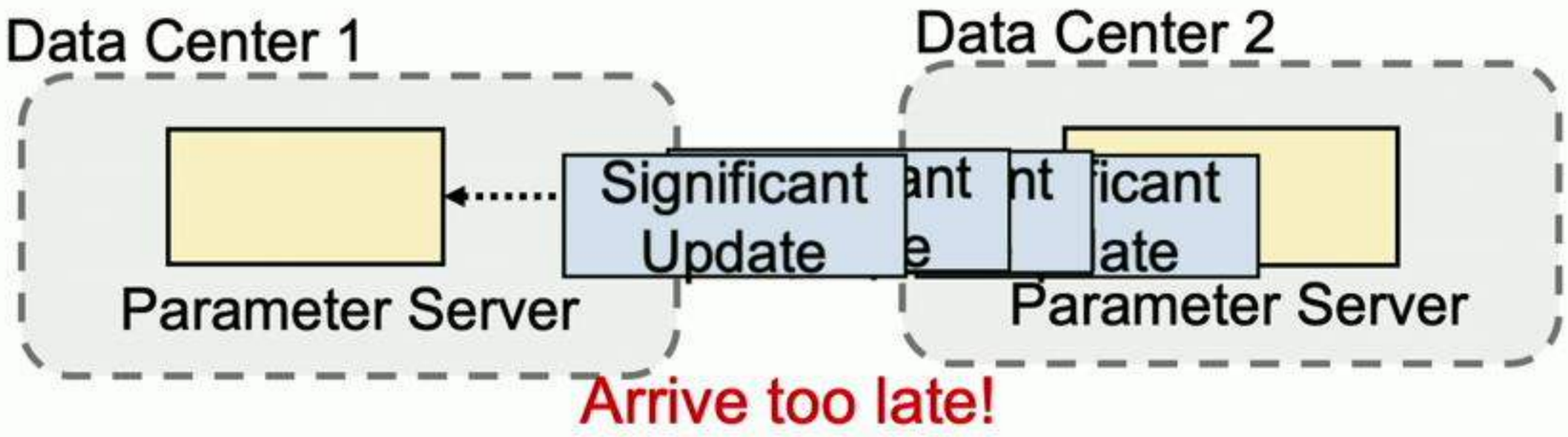
Mirror clock

- Safeguard for pathological cases

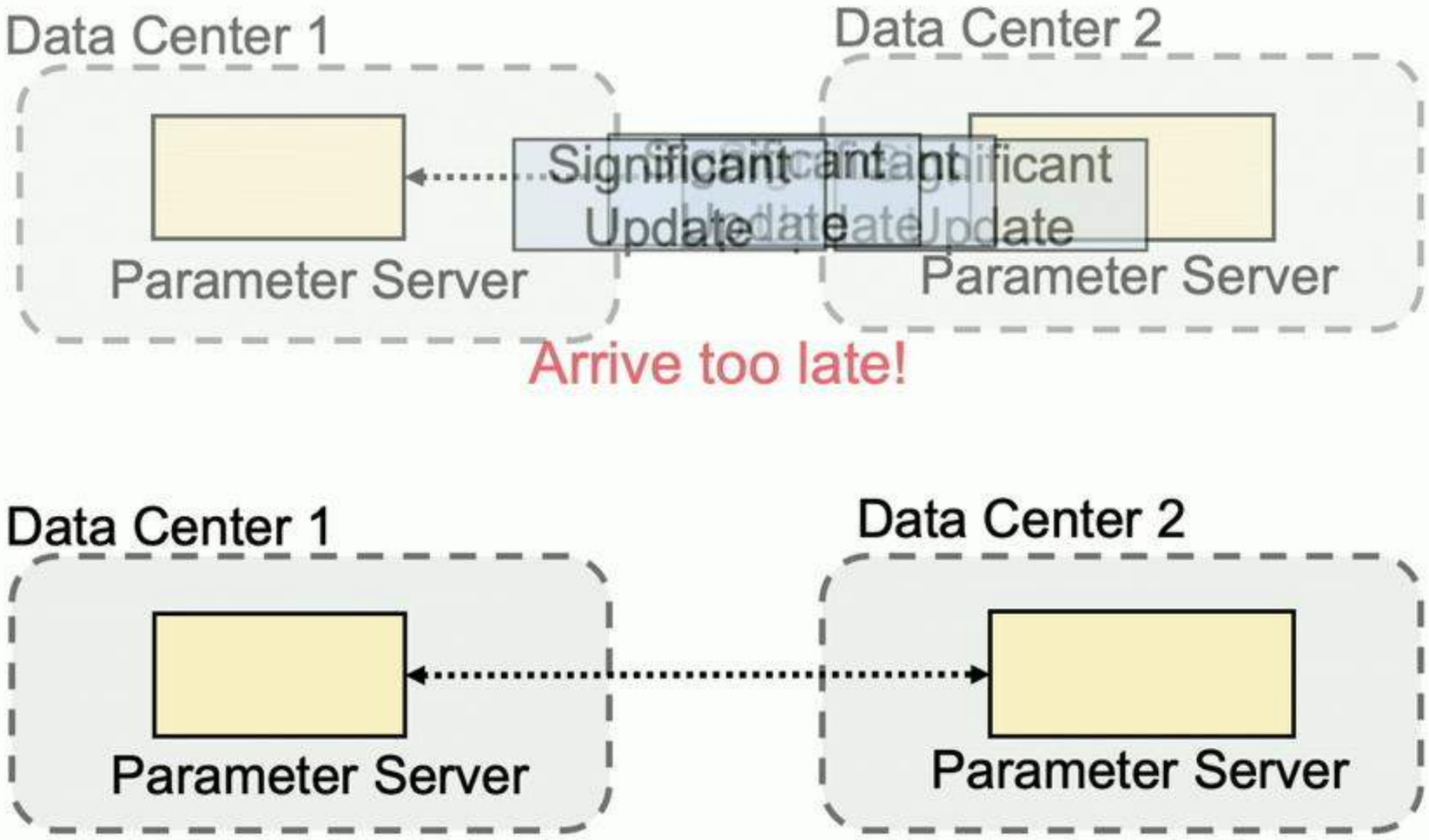
ASP Selective Barrier



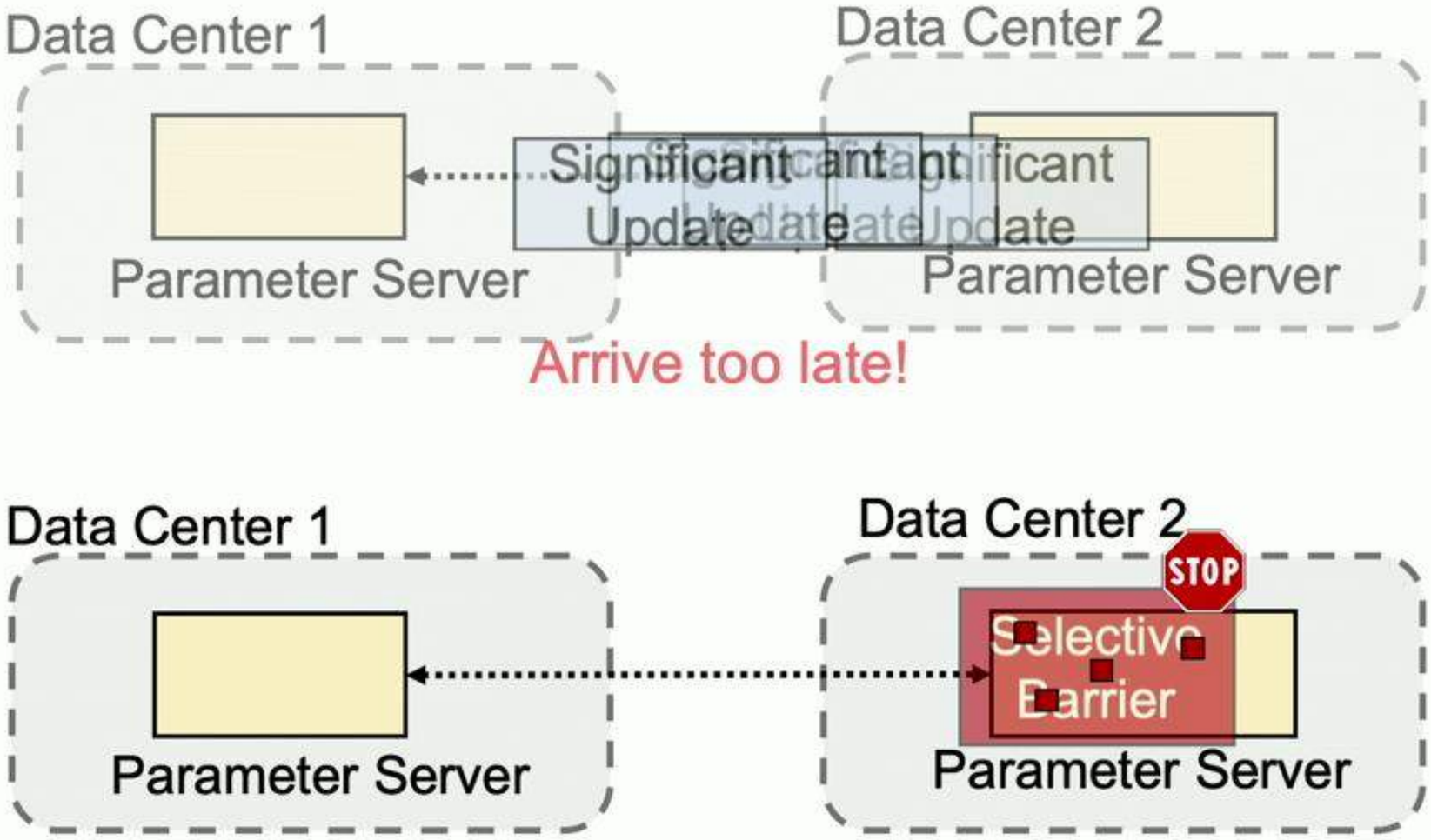
ASP Selective Barrier



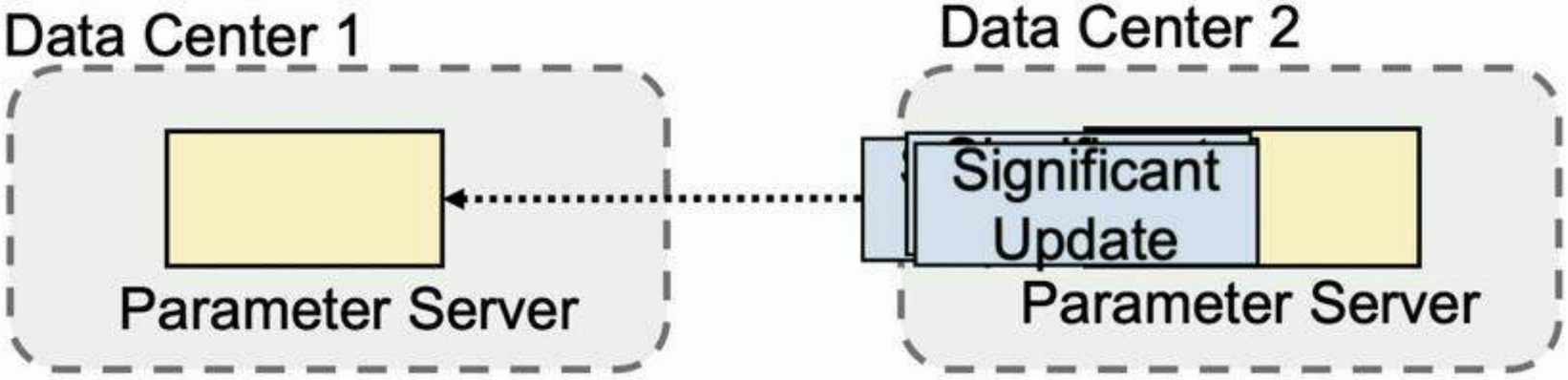
ASP Selective Barrier



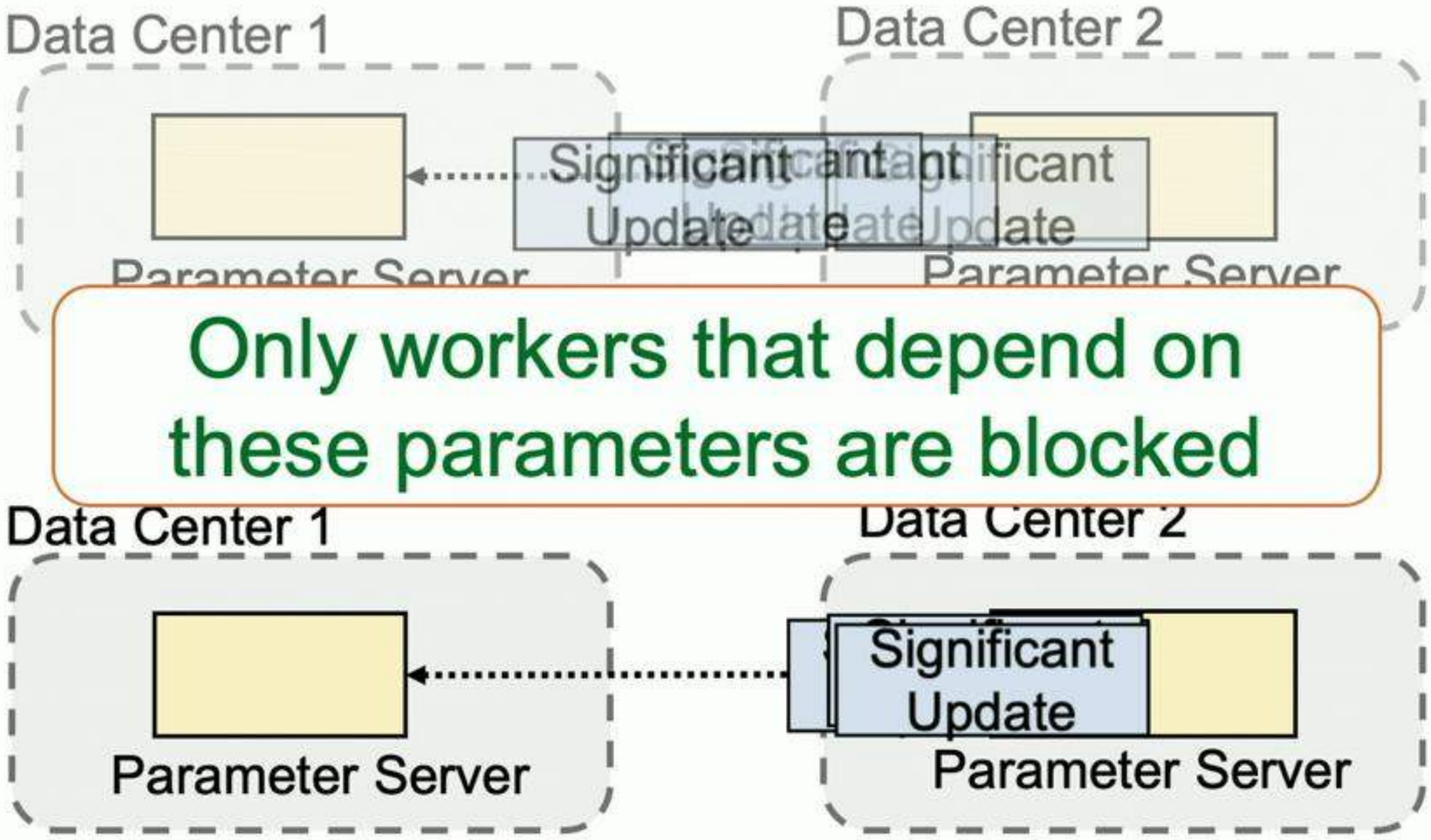
ASP Selective Barrier



ASP Selective Barrier



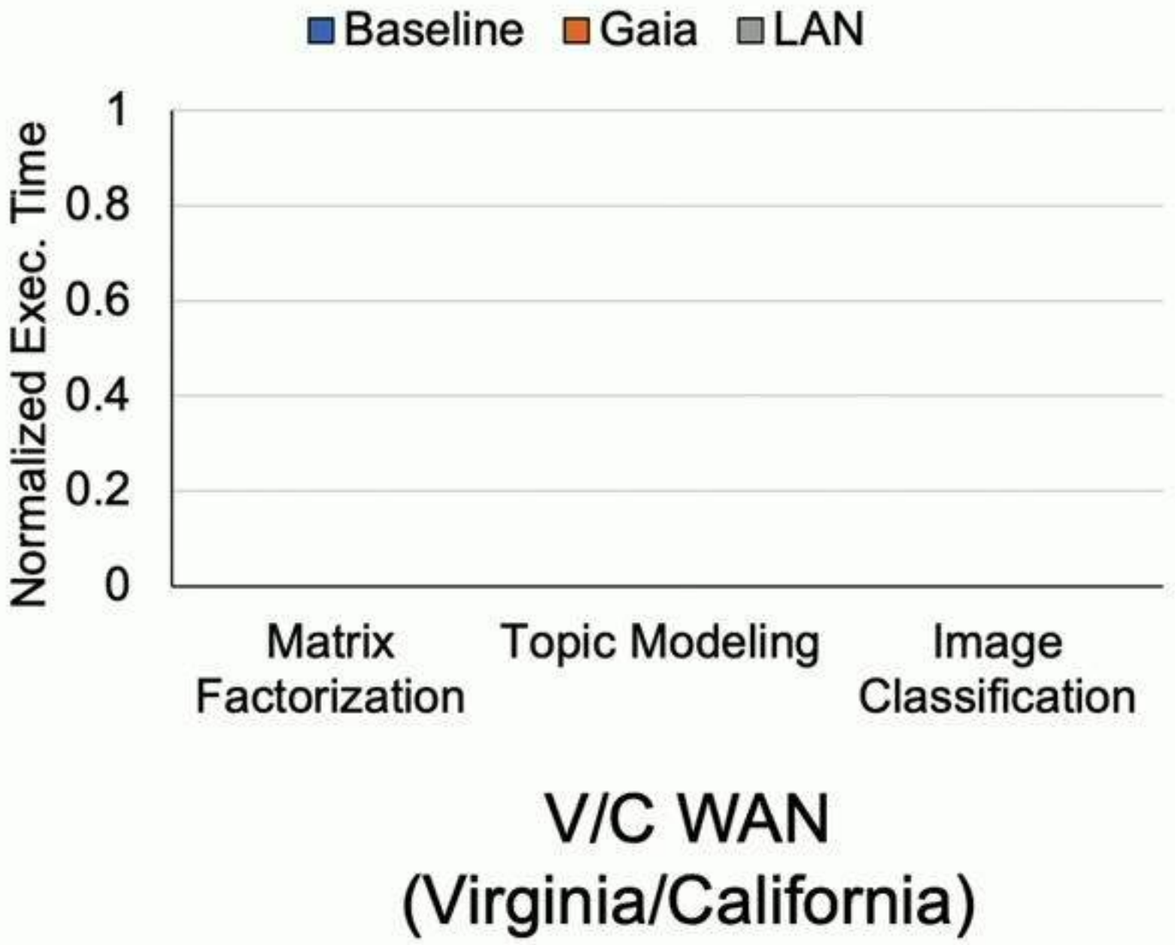
ASP Selective Barrier



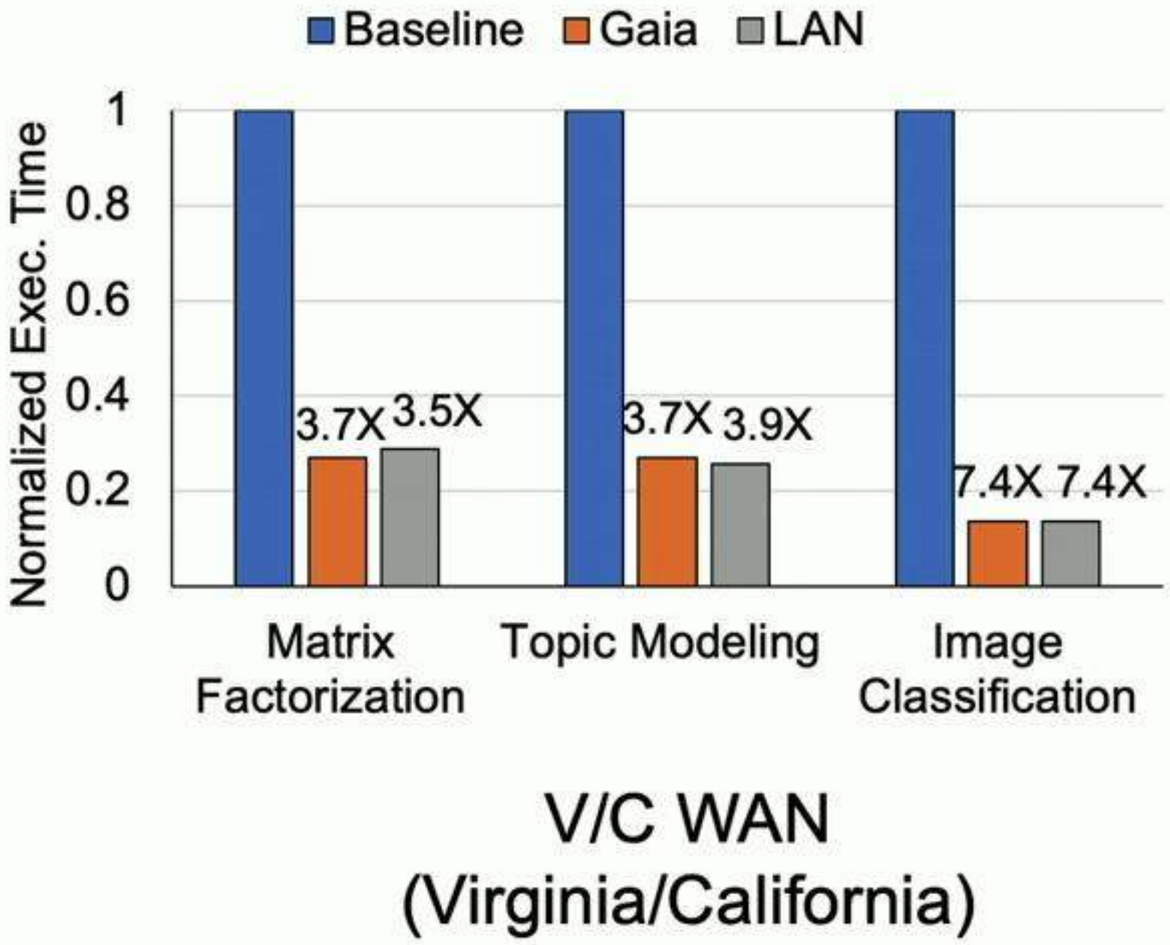
Experimental Setup

- **Applications**
  - Matrix Factorization with the *Netflix* dataset
  - Topic Modeling with the *Nytimes* dataset
  - Image Classification with the *ILSVRC12* dataset
- **Hardware platform**
  - 22 machines with emulated EC2 WAN bandwidth
  - We validated the performance with a real EC2 deployment
- **Baseline**
  - IterStore (Cui et al., SoCC'14) and GeePS (Cui et al., EuroSys'16) on WAN
- **Performance metrics**
  - Execution time until algorithm convergence
  - Monetary cost of algorithm convergence

Performance and WAN Bandwidth

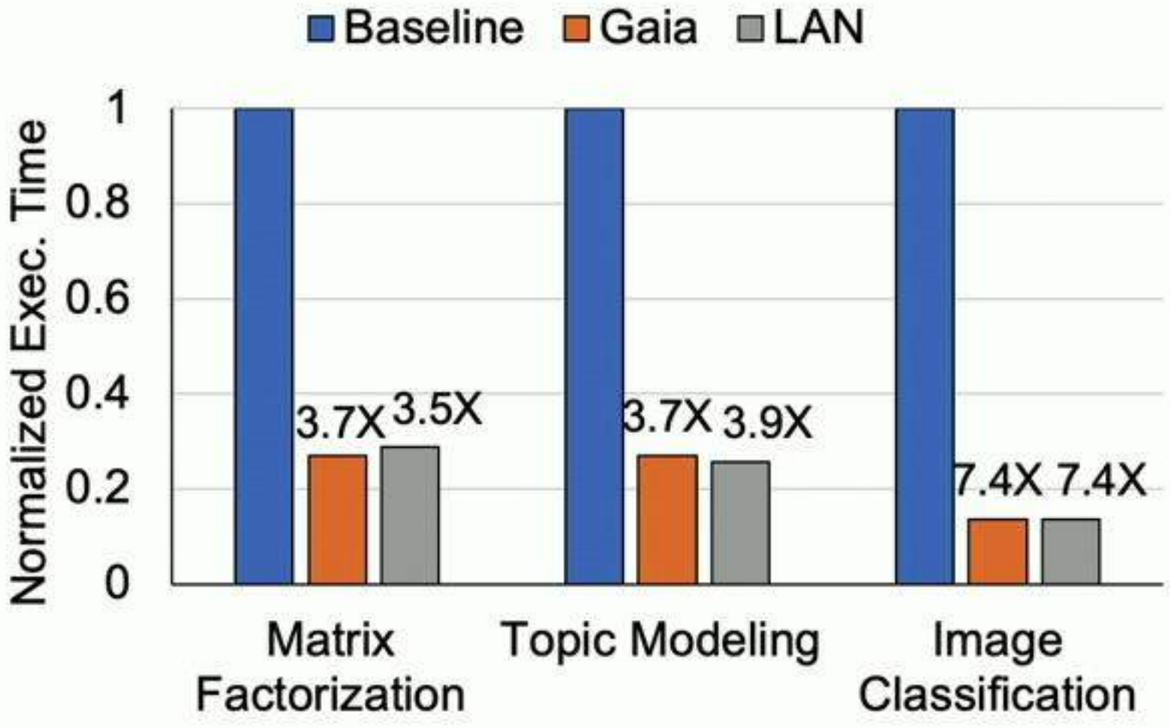


Performance and WAN Bandwidth

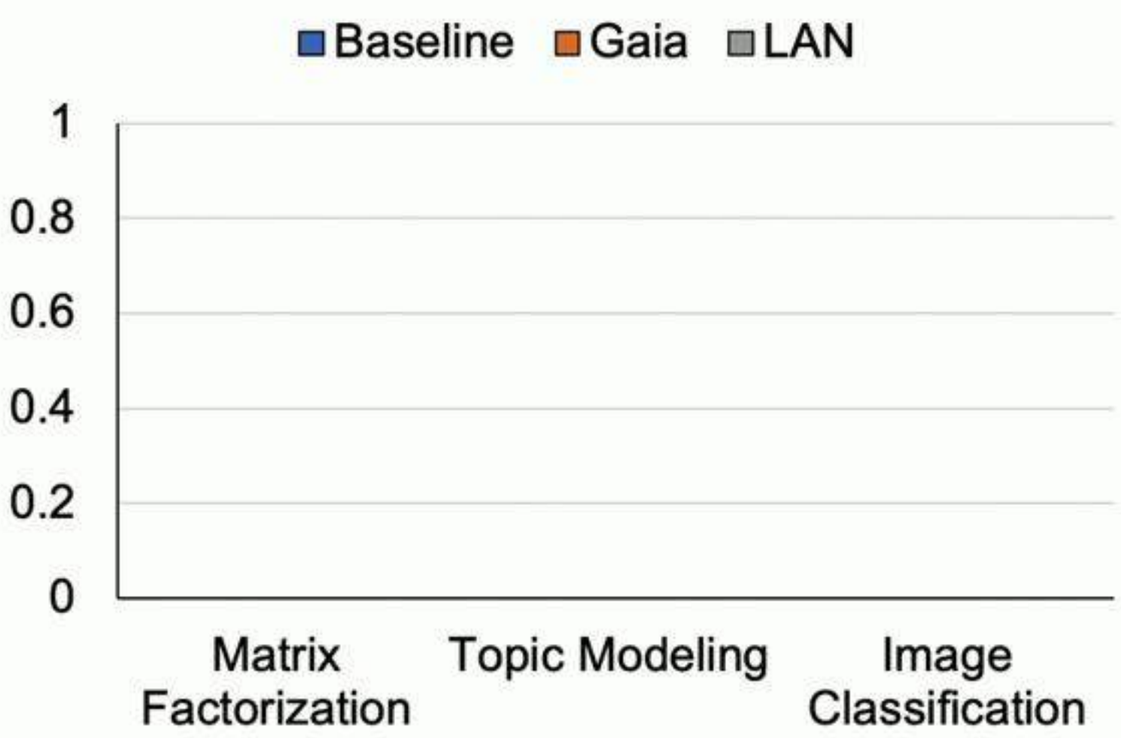


V/C WAN
(Virginia/California)

Performance and WAN Bandwidth

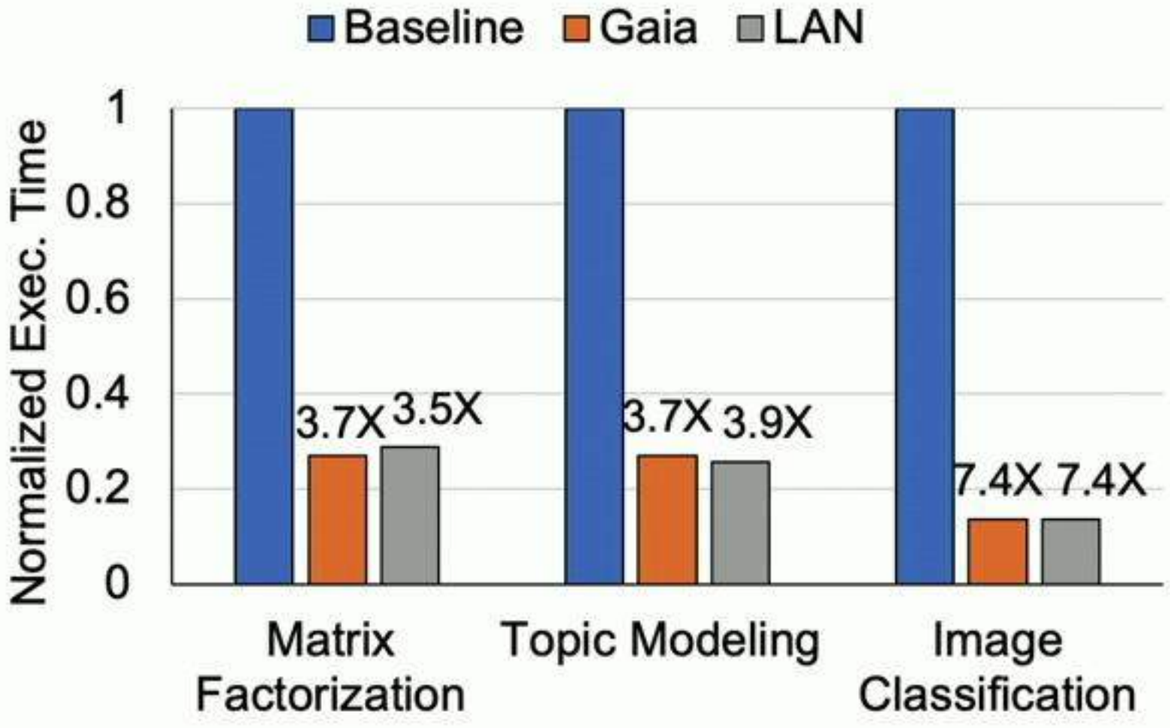


V/C WAN
(Virginia/California)

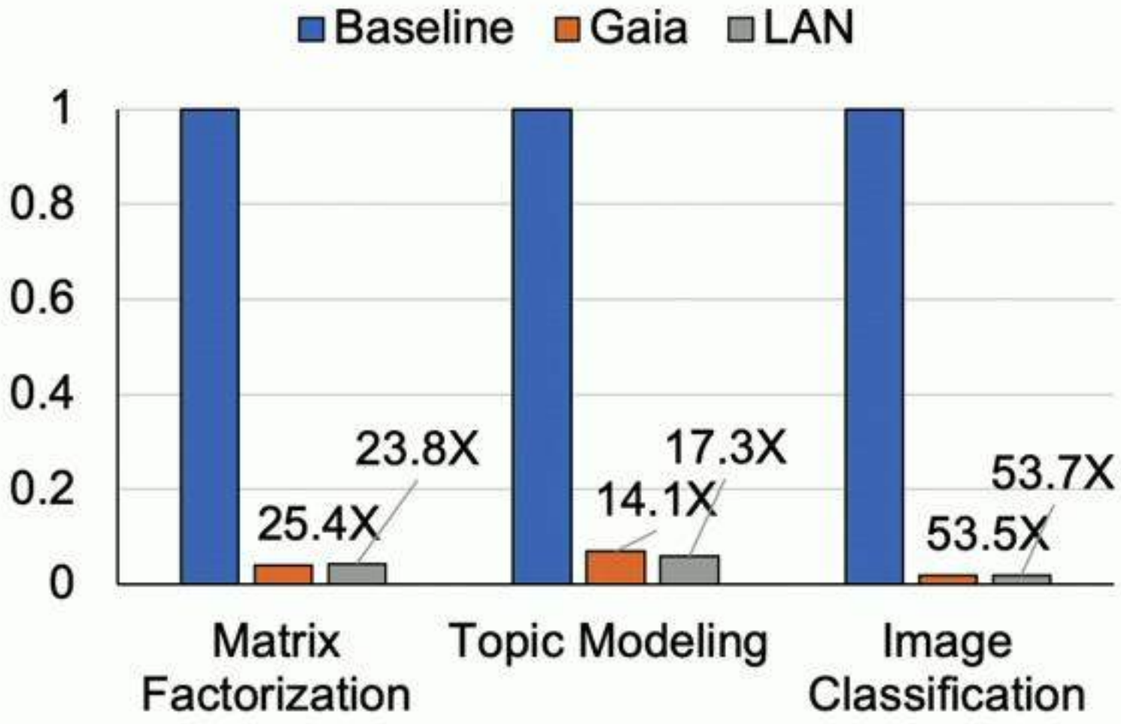


S/S WAN
(Singapore/São Paulo)

Performance and WAN Bandwidth

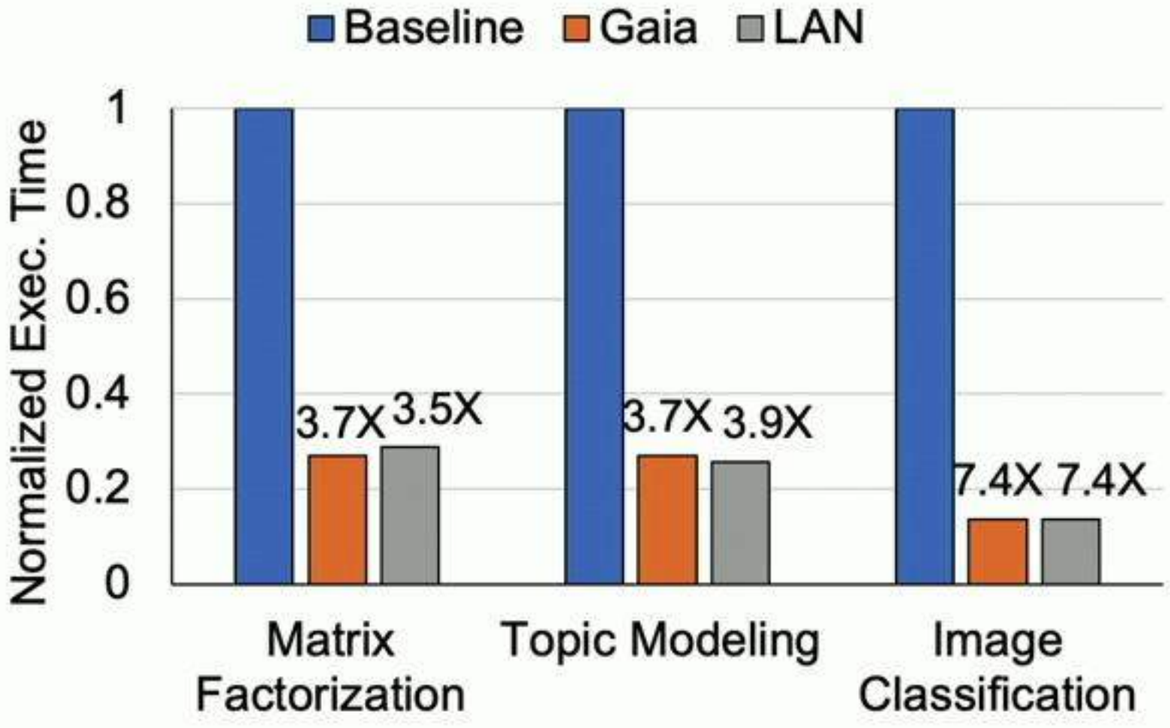


V/C WAN
(Virginia/California)

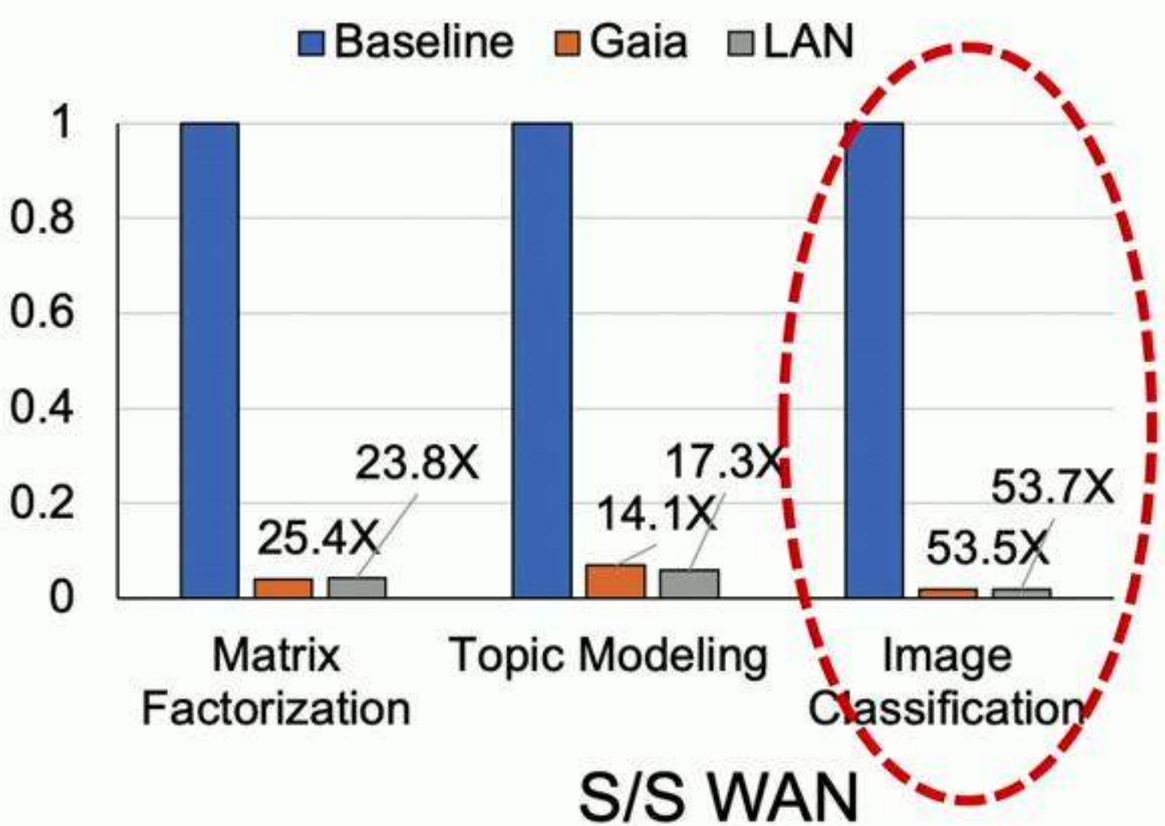


S/S WAN
(Singapore/São Paulo)

Performance and WAN Bandwidth

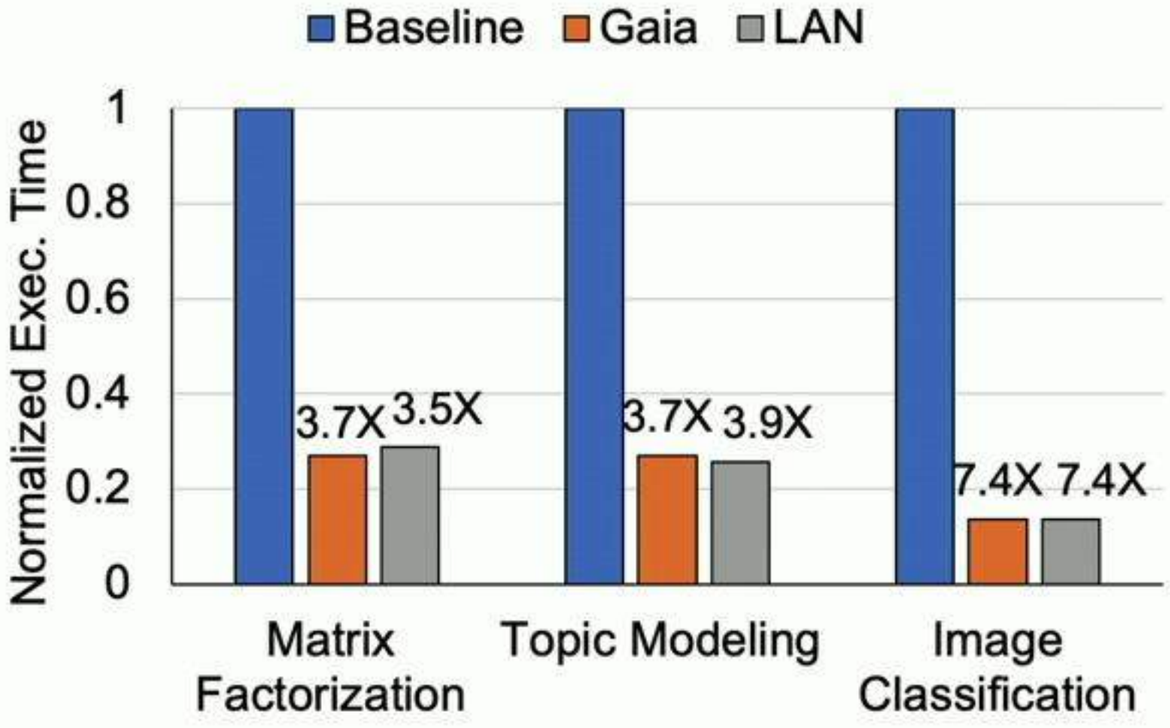


V/C WAN
(Virginia/California)

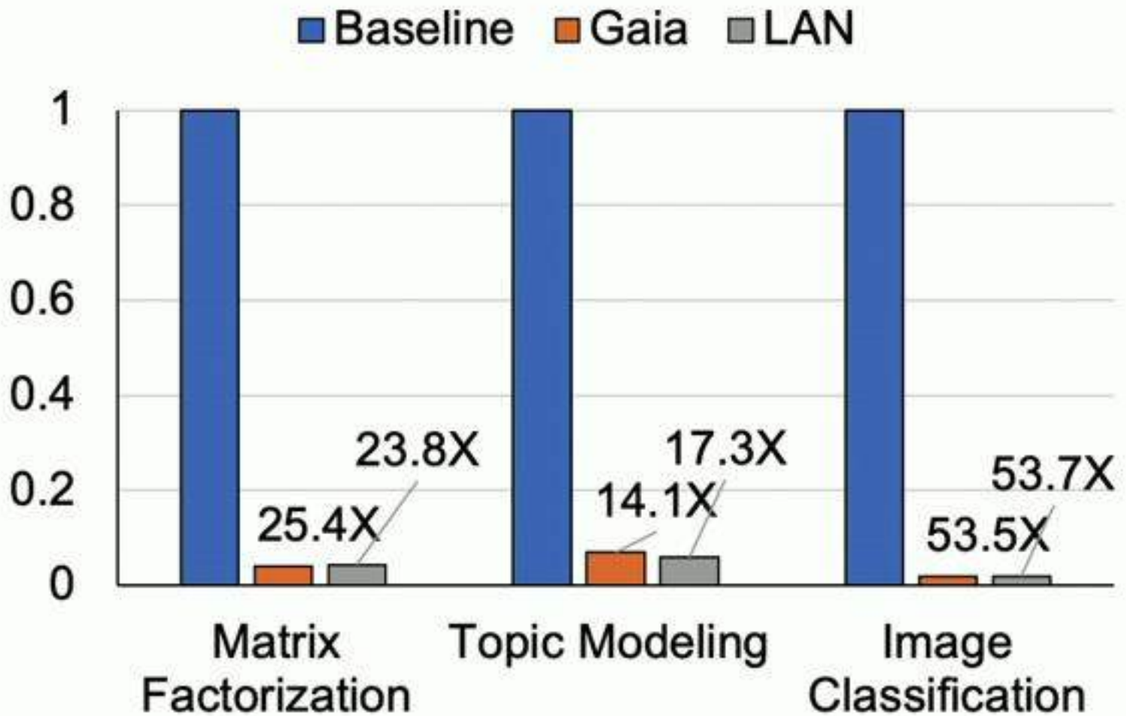


S/S WAN
(Singapore/São Paulo)

Performance and WAN Bandwidth



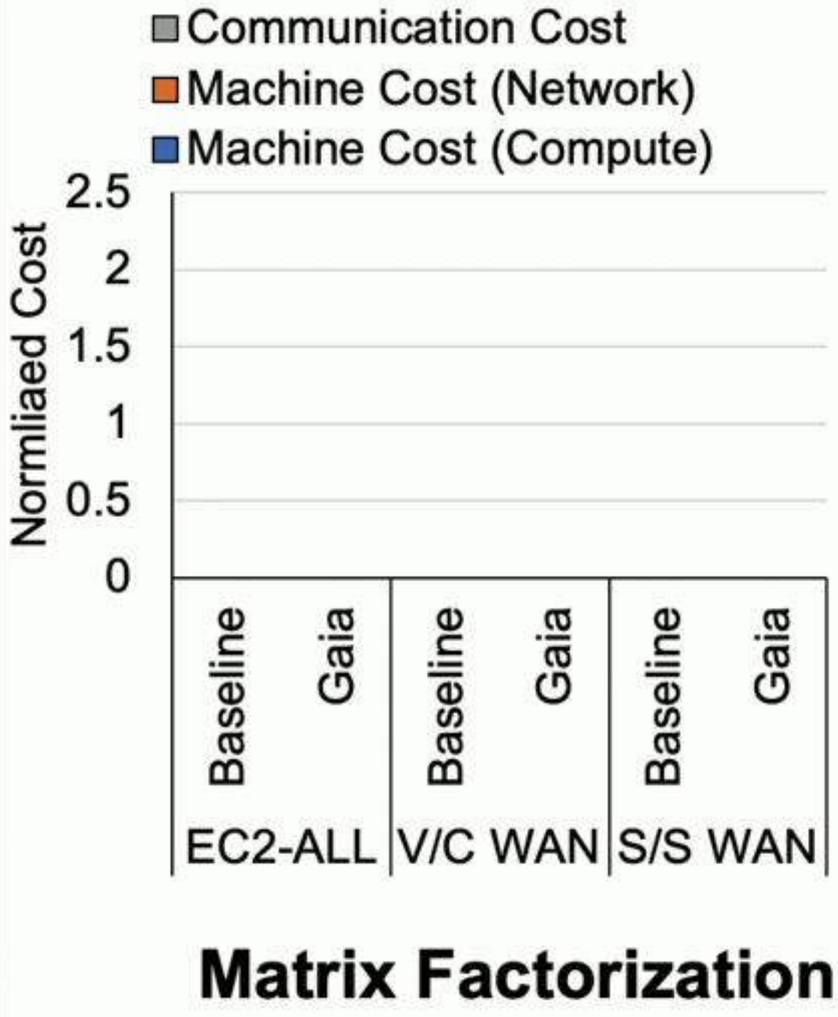
V/C WAN



S/S WAN

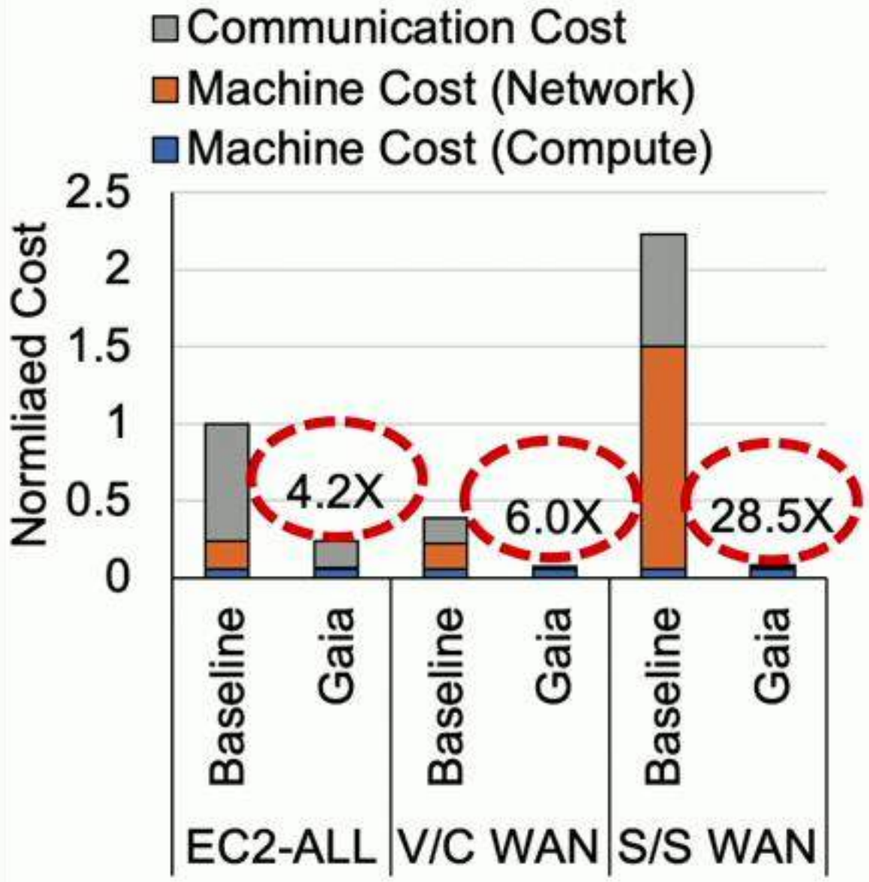
Gaia is at most 1.23X of LAN speeds

Results – EC2 Monetary Cost



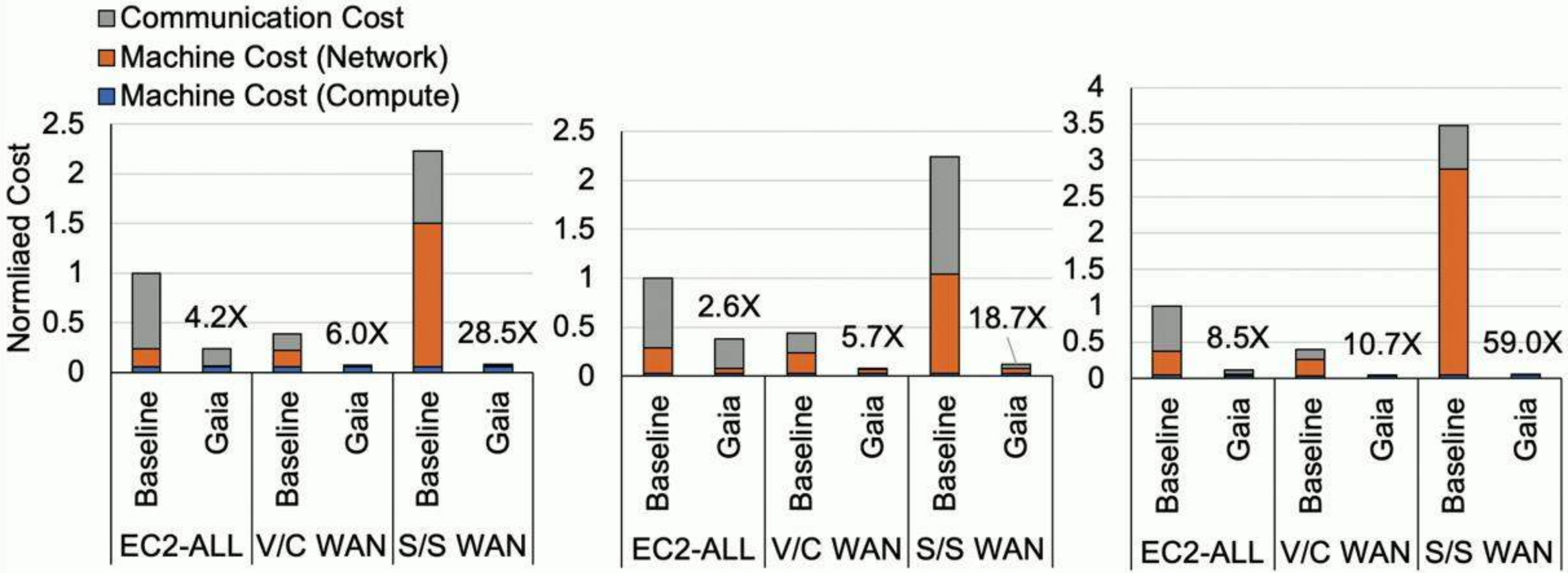
Matrix Factorization

Results – EC2 Monetary Cost



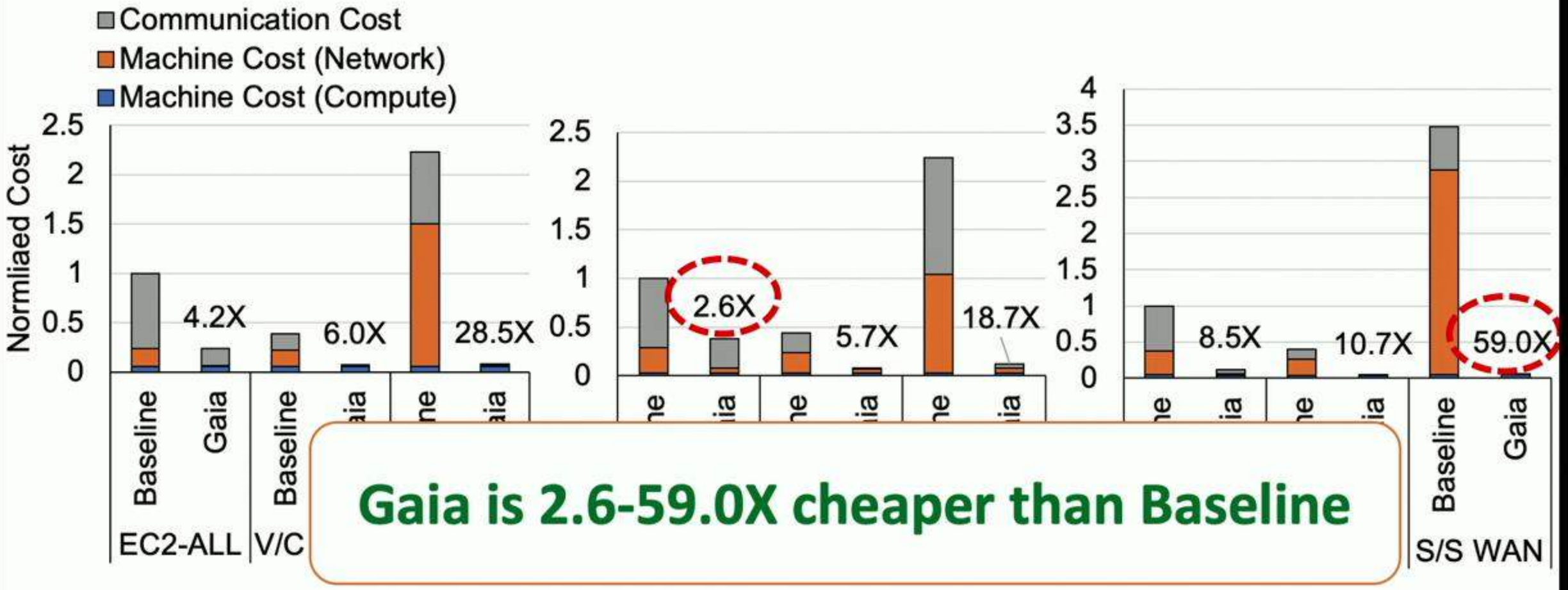
Matrix Factorization

Results – EC2 Monetary Cost



Matrix Factorization **Topic Modeling** **Image Classification**

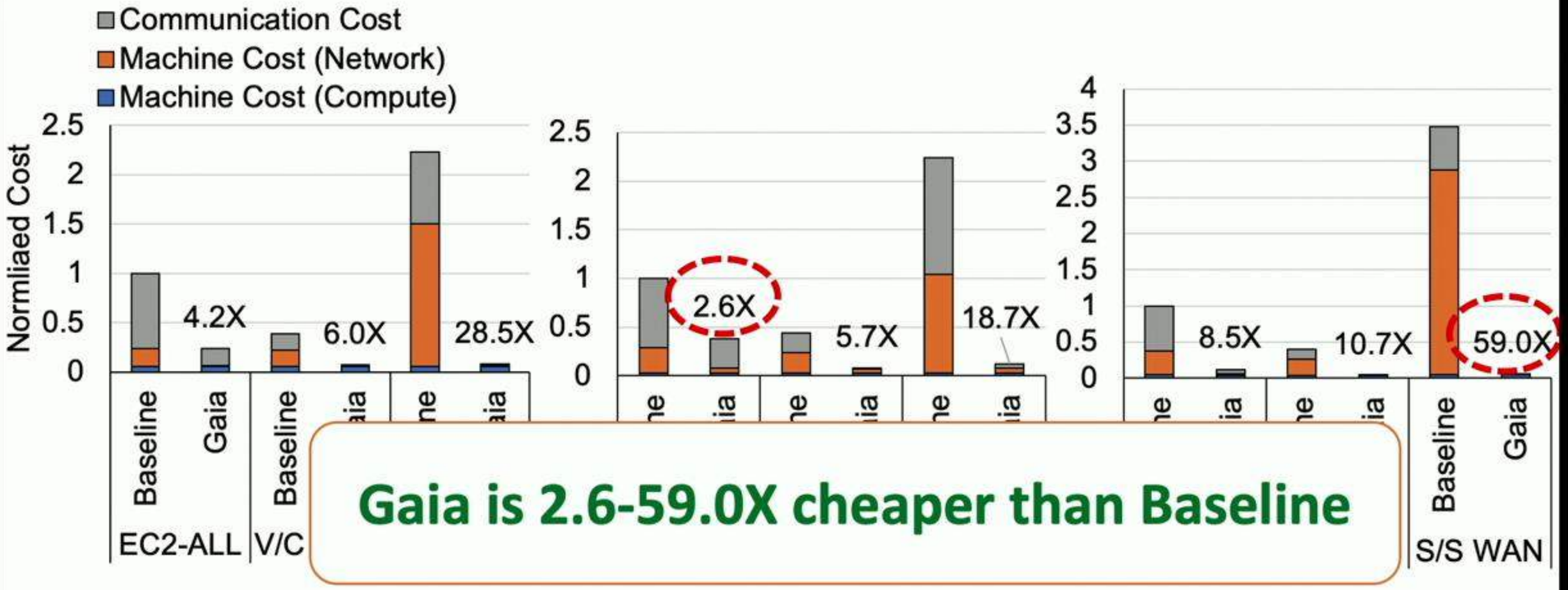
Results – EC2 Monetary Cost



Key Takeaways

Takeaways

Results – EC2 Monetary Cost



Compare Against Centralizing Approach

		Gaia Speedup over Centralize	Gaia to Centralize Cost Ratio
Matrix Factorization	EC2-ALL	1.11	3.54
	V/C WAN	1.22	1.00
	S/S WAN	2.13	1.17
Topic Modeling	EC2-ALL	0.80	6.14
	V/C WAN	1.02	1.26
	S/S WAN	1.25	1.92
Image Classification	EC2-ALL	0.76	3.33
	V/C WAN	1.12	1.07
	S/S WAN	1.86	1.08

Takeaways

- ML serving and training over highly-distributed and rapidly-growing are challenging
- Our approach: designing ML systems that exploit the characteristics of:
  - ML algorithm (e.g., most training updates are not significant),
  - ML model structures (e.g., top-K results in DNN classifiers),
  - ML training/serving data (e.g., object similarities in videos)
- Improve the latency and cost of ML serving and training by one to two orders of magnitude

Other Contributions

Processing-in-Memory

- **Automatic offloading [ISCA'16, SC'17]**
- **Pointer chasing accelerator [ICCD'16]**
- Cache coherence [CAL'17, ISCA'19]
- Bulk bit-wise ops [CAL'16]

Cross-layer abstractions

- Expressive Memory [ISCA'18]
- Locality descriptor in GPUs [ISCA'18]
- GPU programmability [MICRO'16]

Memory

- Variable DRAM latency [SIGMETRICS'16]

In This Talk

Focus: ML Serving for Rapidly Growing Data [OSDI'18]

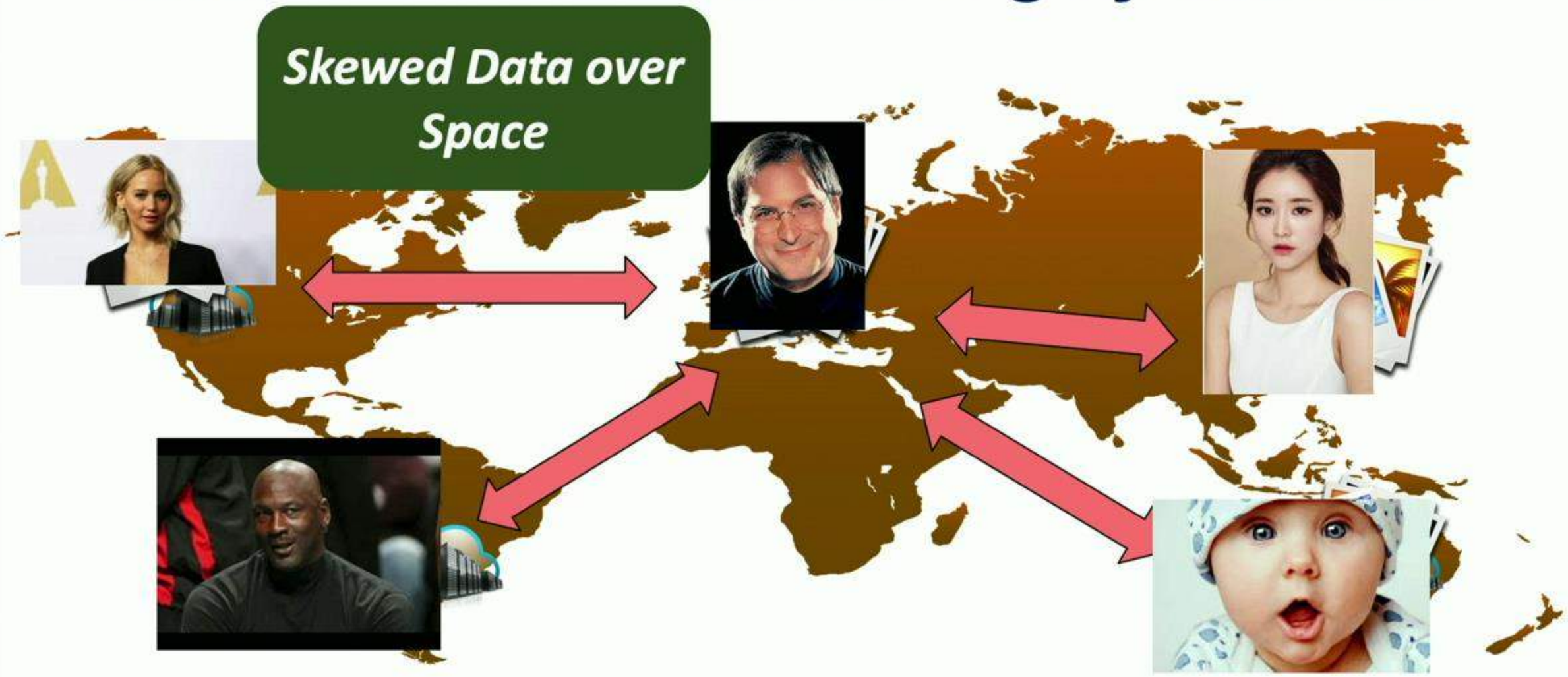
Gaia: ML Training for Geo-Distributed Data [NSDI'17]

On-going and Future Work

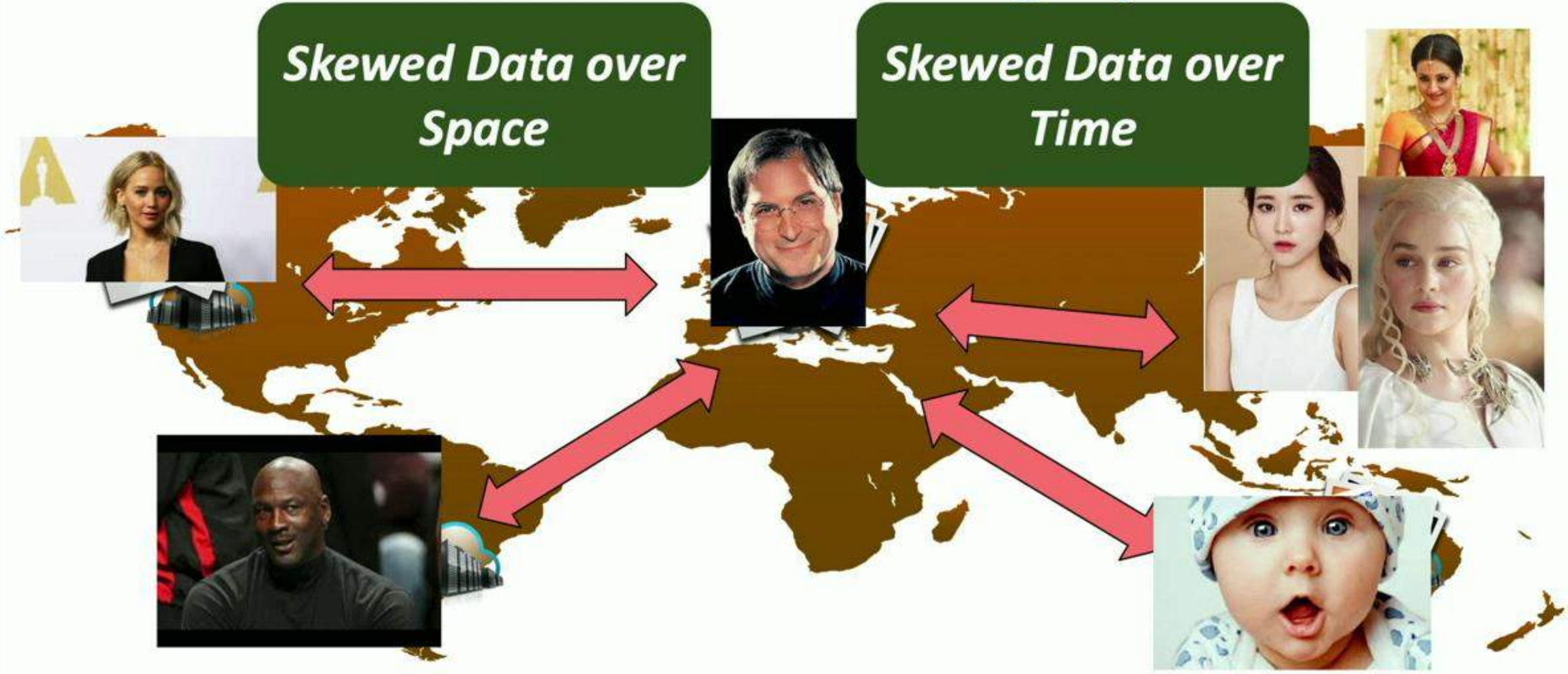
Real-World Data can be Highly Skewed

Real-World Data can be Highly Skewed

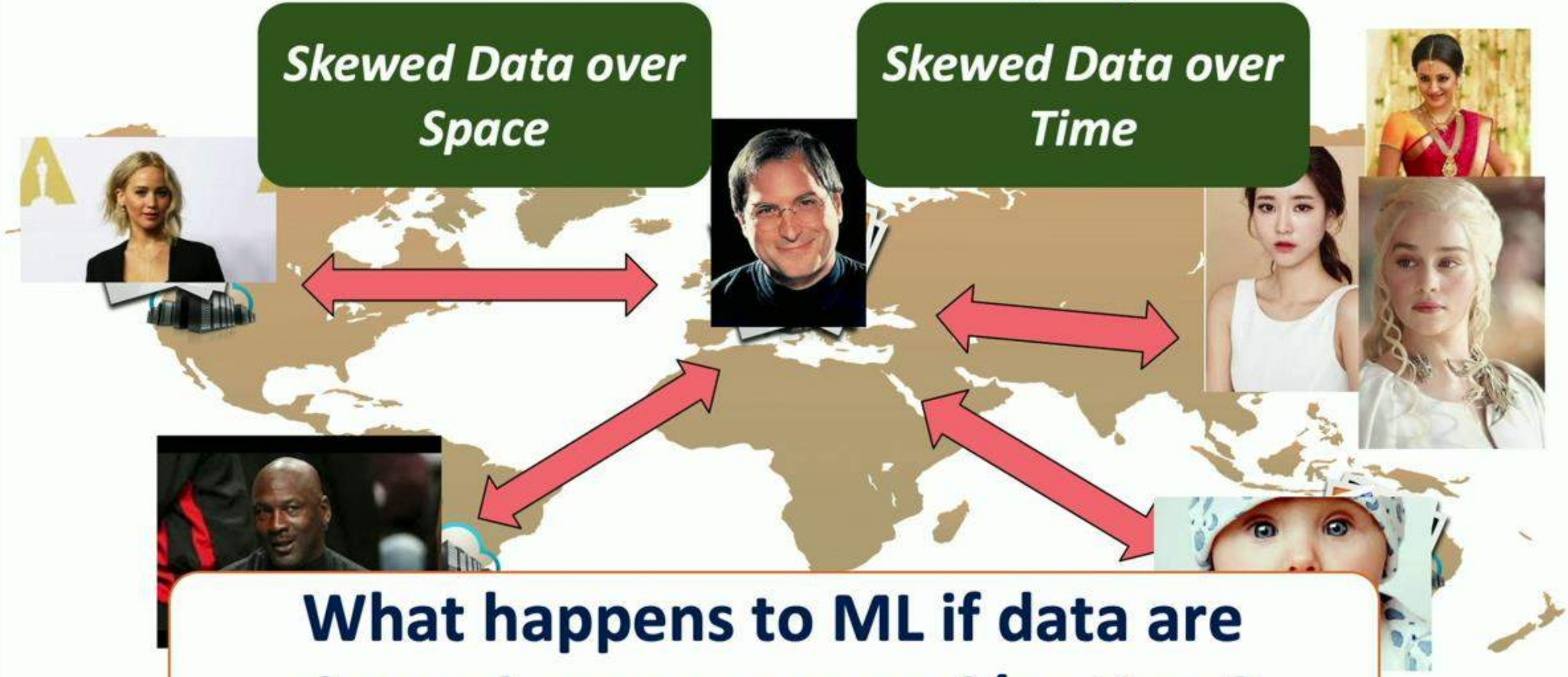
Real-World Data can be Highly Skewed



Real-World Data can be Highly Skewed



Real-World Data can be Highly Skewed



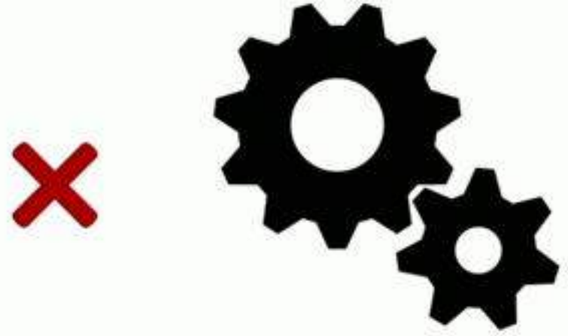
What happens to ML if data are skewed over space and/or time?

Understanding Skewed Data over Space

ML Application

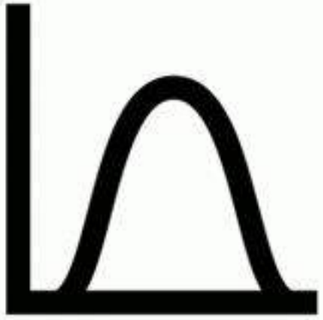
- Image Classification (with various DNNs and datasets)
- Face recognition

Decentralized Learning Approach



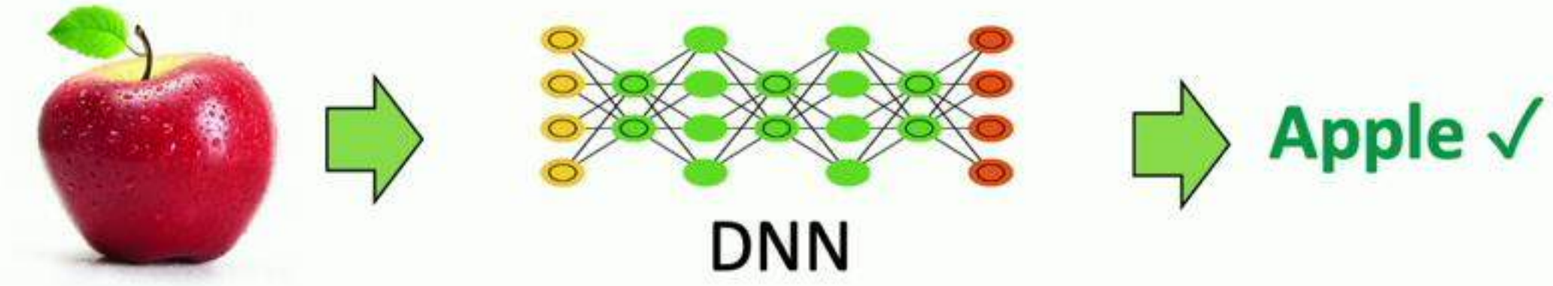
- Gaia [NSDI'17]
- FederatedAveraging [AISTATS'17]
- DeepGradientCompression [ICLR'18]

Skewness of Data Partitions



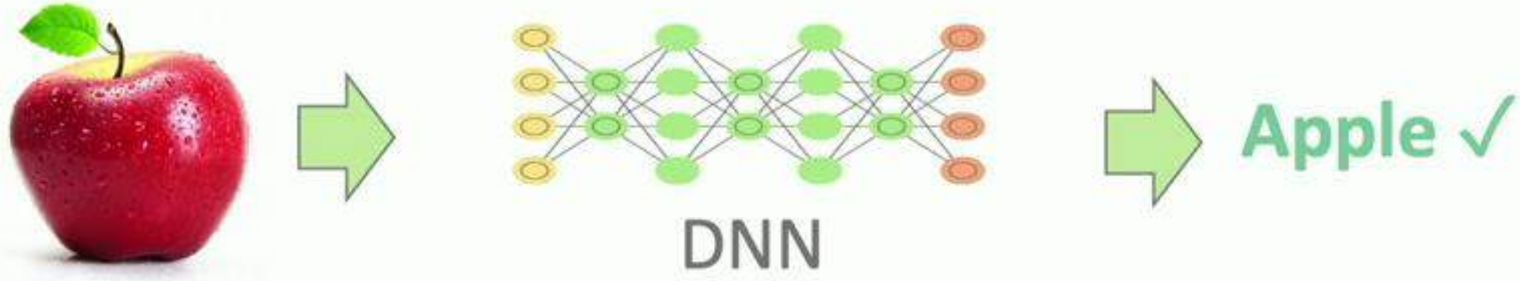
Skewed Data over Space: Setup

- Task: Classify an image into one of the object classes



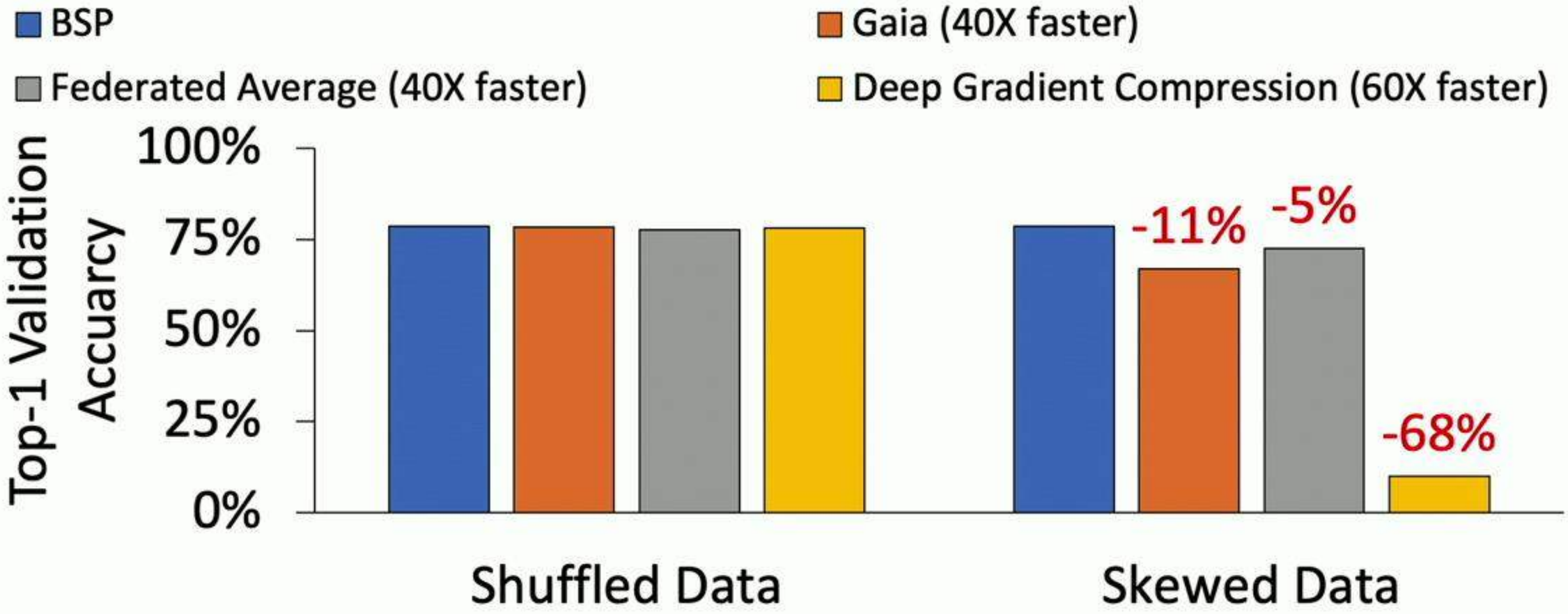
Skewed Data over Space: Setup

- Task: Classify an image into one of the object classes

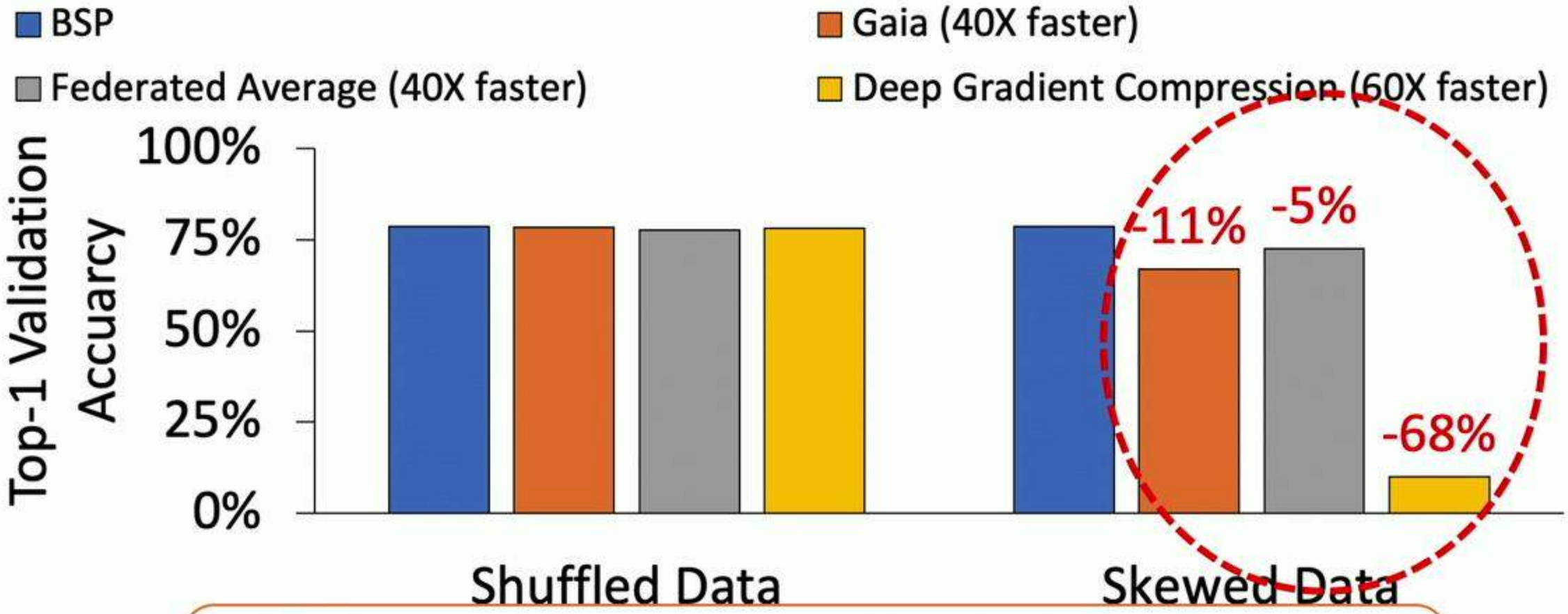


- Each data center has some classes of images

Results: GoogleNet over CIFAR-10

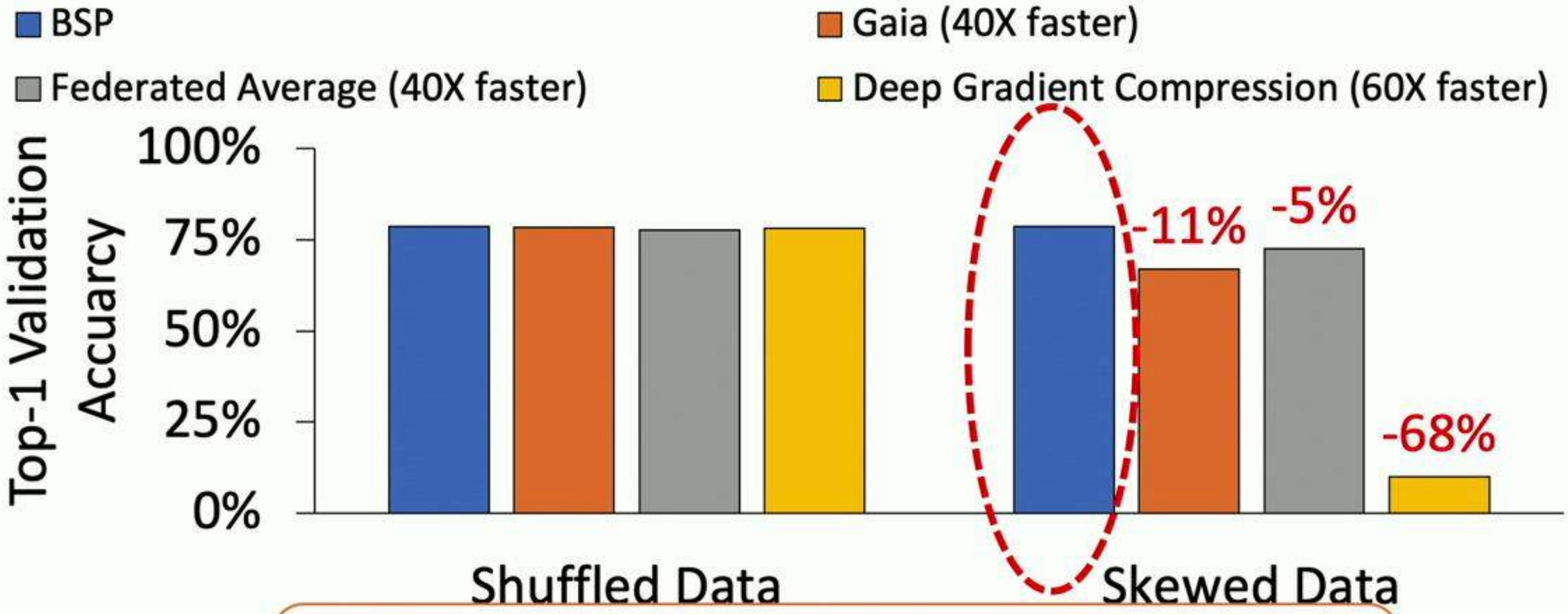


Results: GoogleNet over CIFAR-10



All decentralized learning approaches lose significant accuracy

Results: GoogleNet over CIFAR-10



Tight synchronization (BSP) is accurate but too slow

Similar Results across the Board

Image Classification (CIFAR-10)

Similar Results across the Board

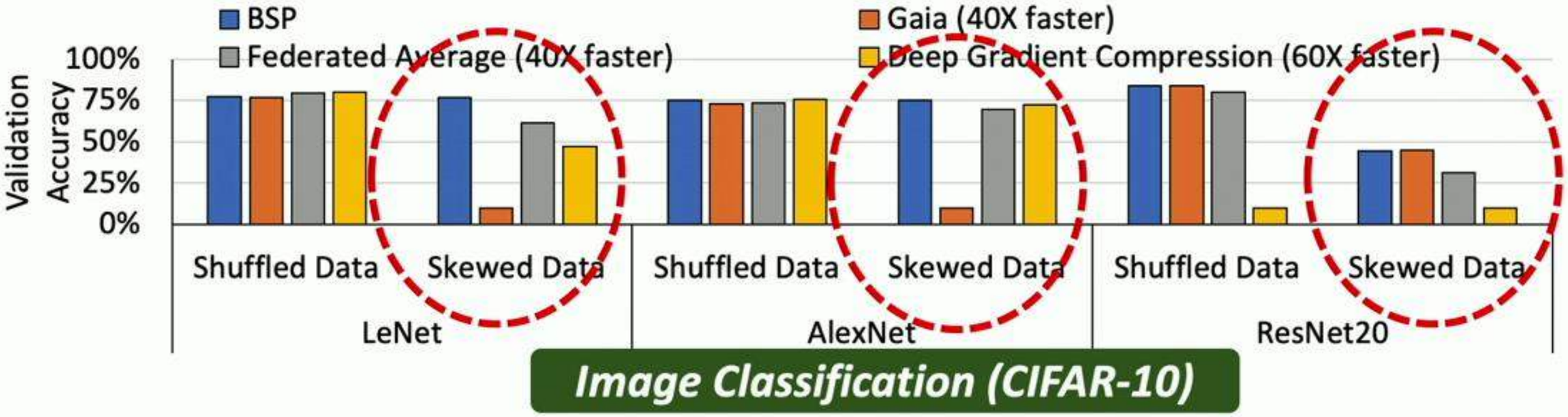


Image Classification (CIFAR-10)

Similar Results across the Board

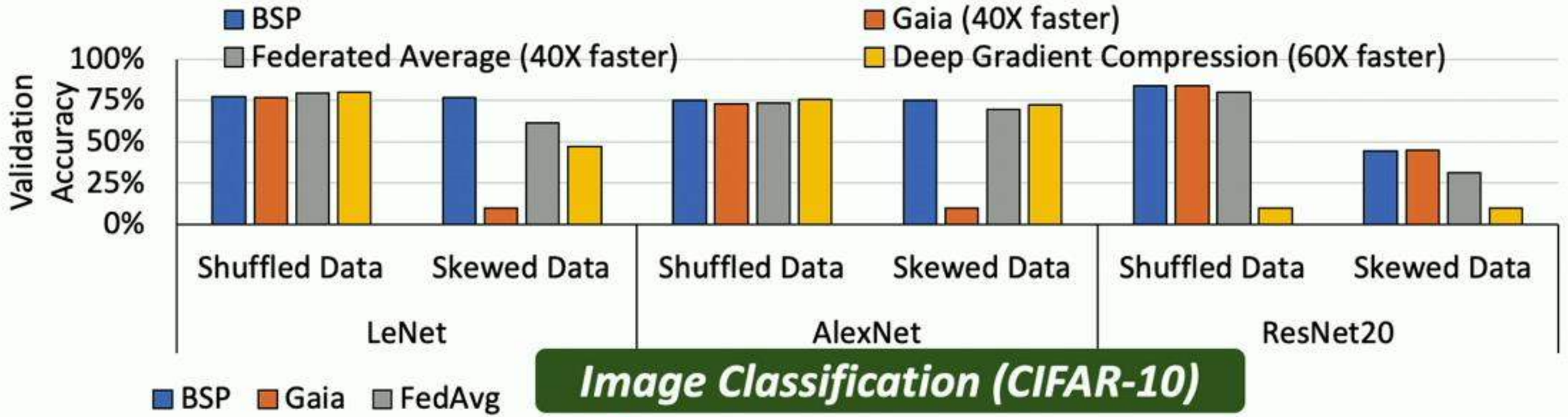


Image Classification (CIFAR-10)

Image Classification (ImageNet)

Similar Results across the Board

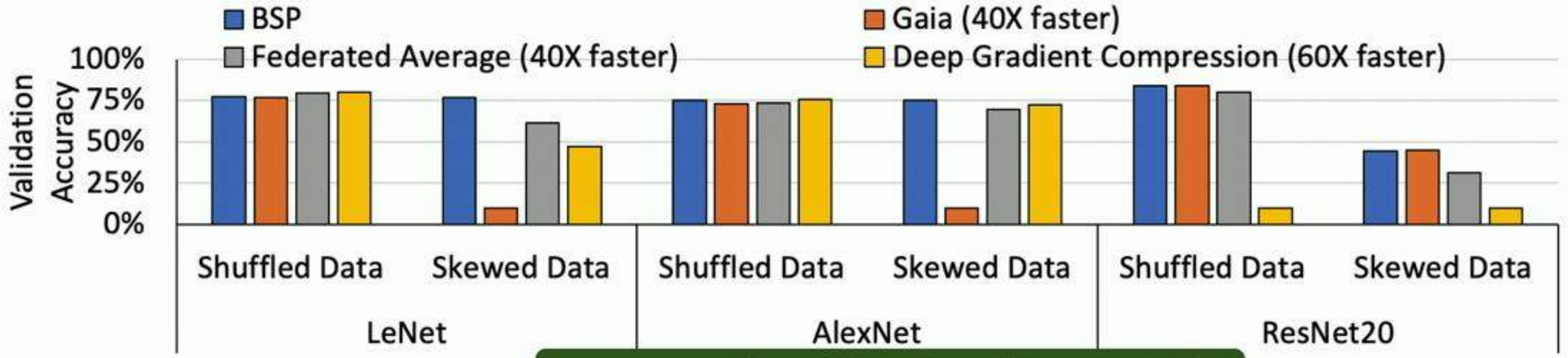


Image Classification (CIFAR-10)

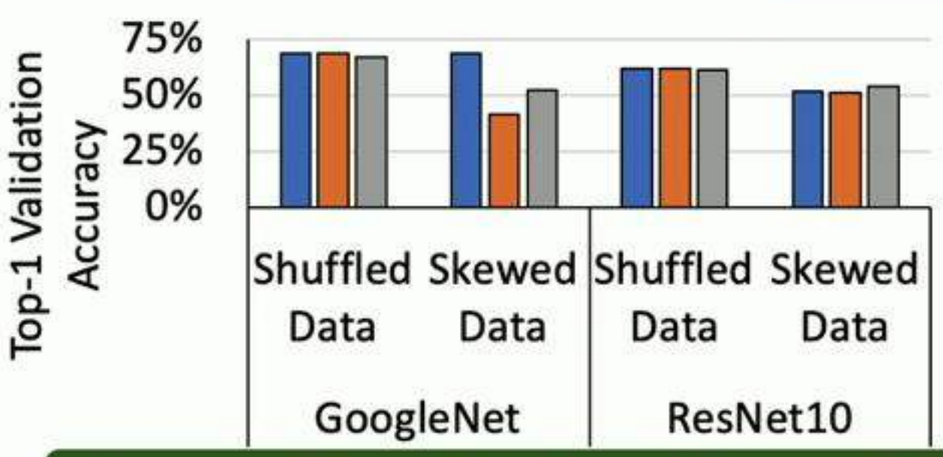
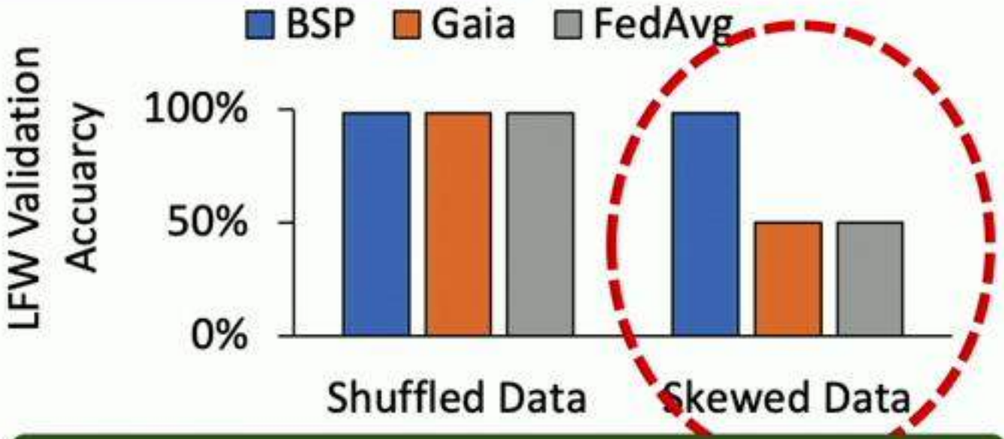
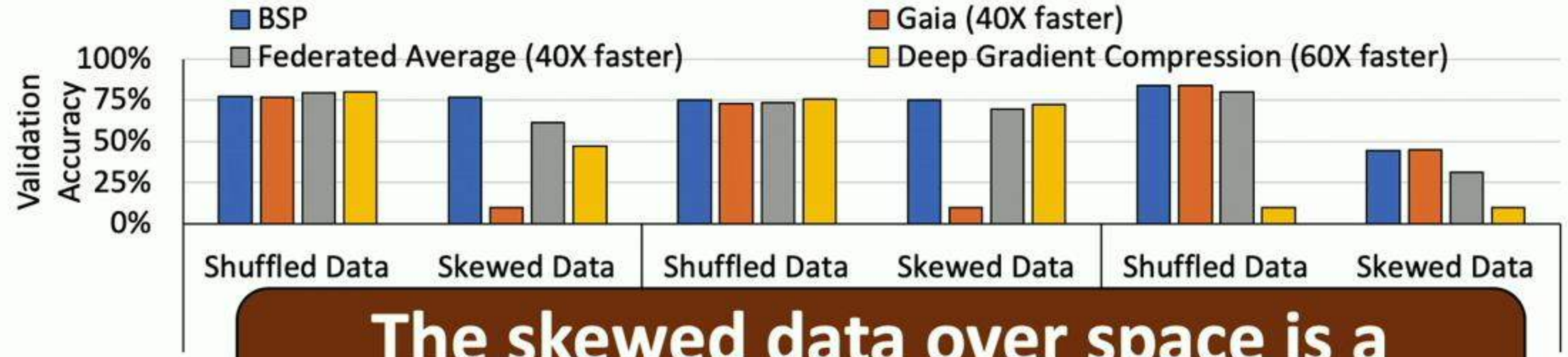


Image Classification (ImageNet)



Face Recognition

Similar Results across the Board



The skewed data over space is a pervasive and fundamental problem

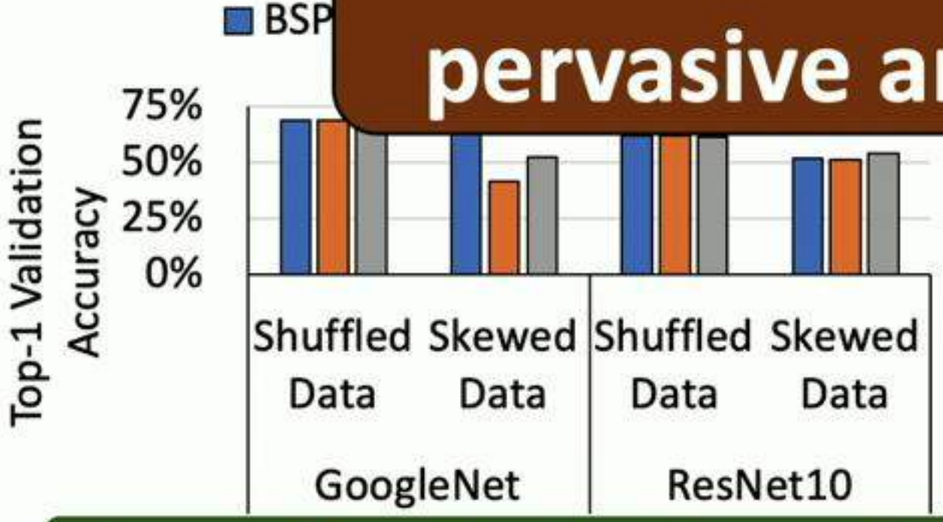
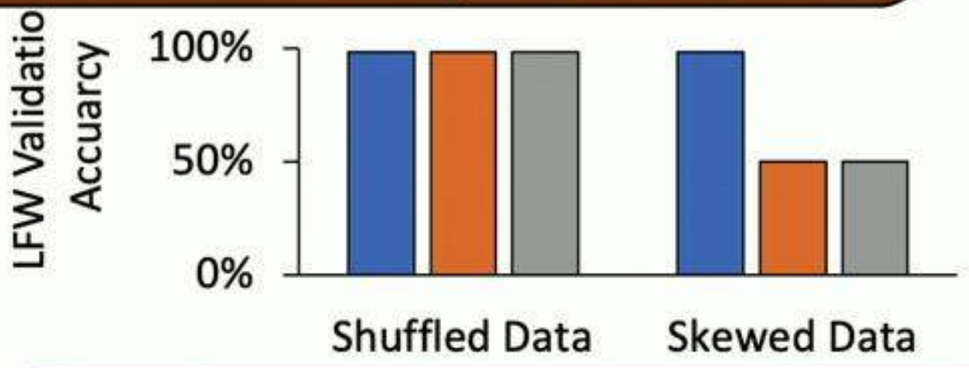


Image Classification (ImageNet)



Face Recognition

Even BSP cannot solve this problem for DNNs with Batch Normalization Layers

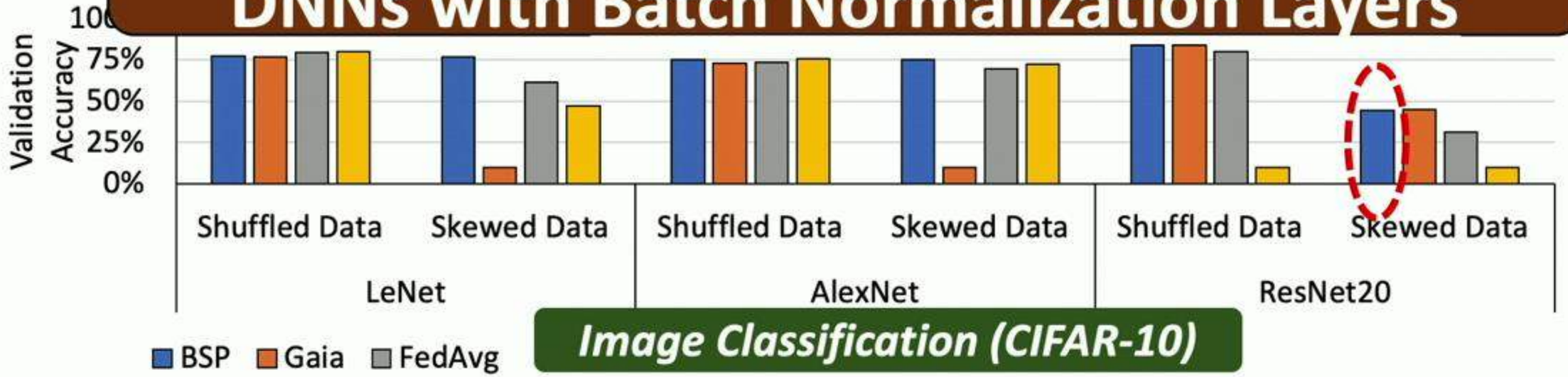


Image Classification (CIFAR-10)

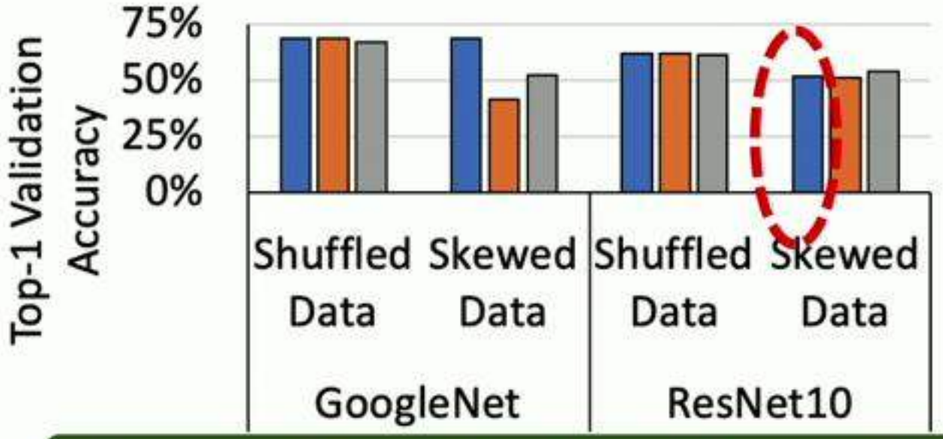
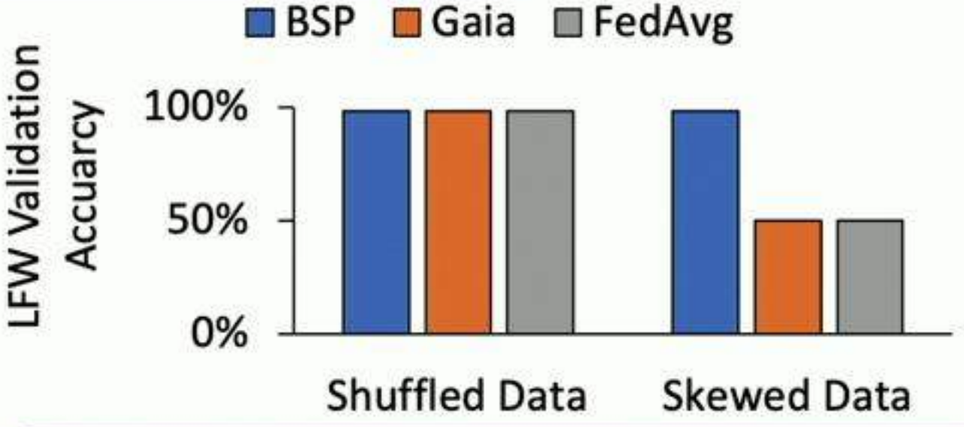


Image Classification (ImageNet)



Face Recognition

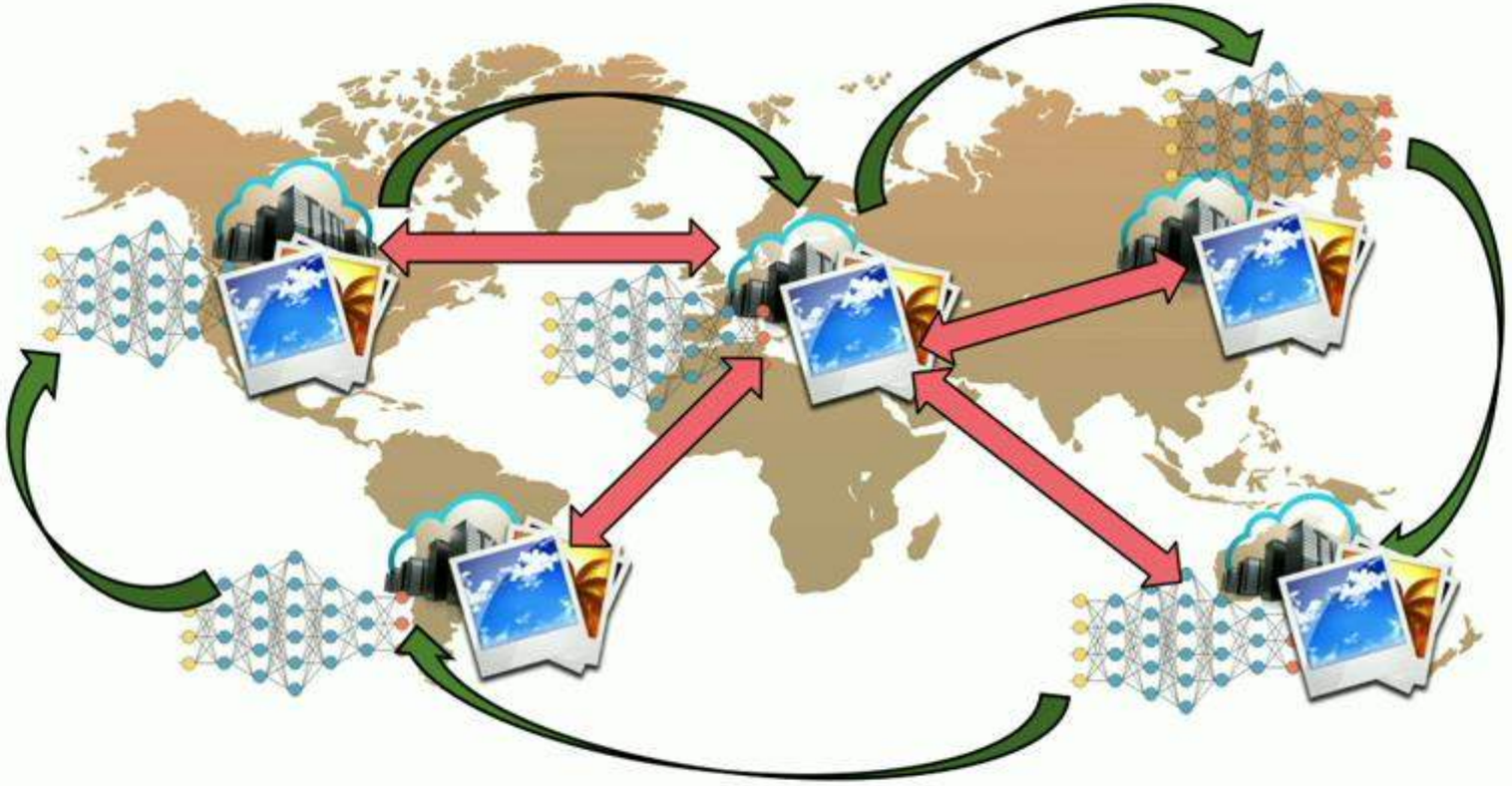
Direction #1: Skewed Data over Space

Direction #1: Skewed Data over Space

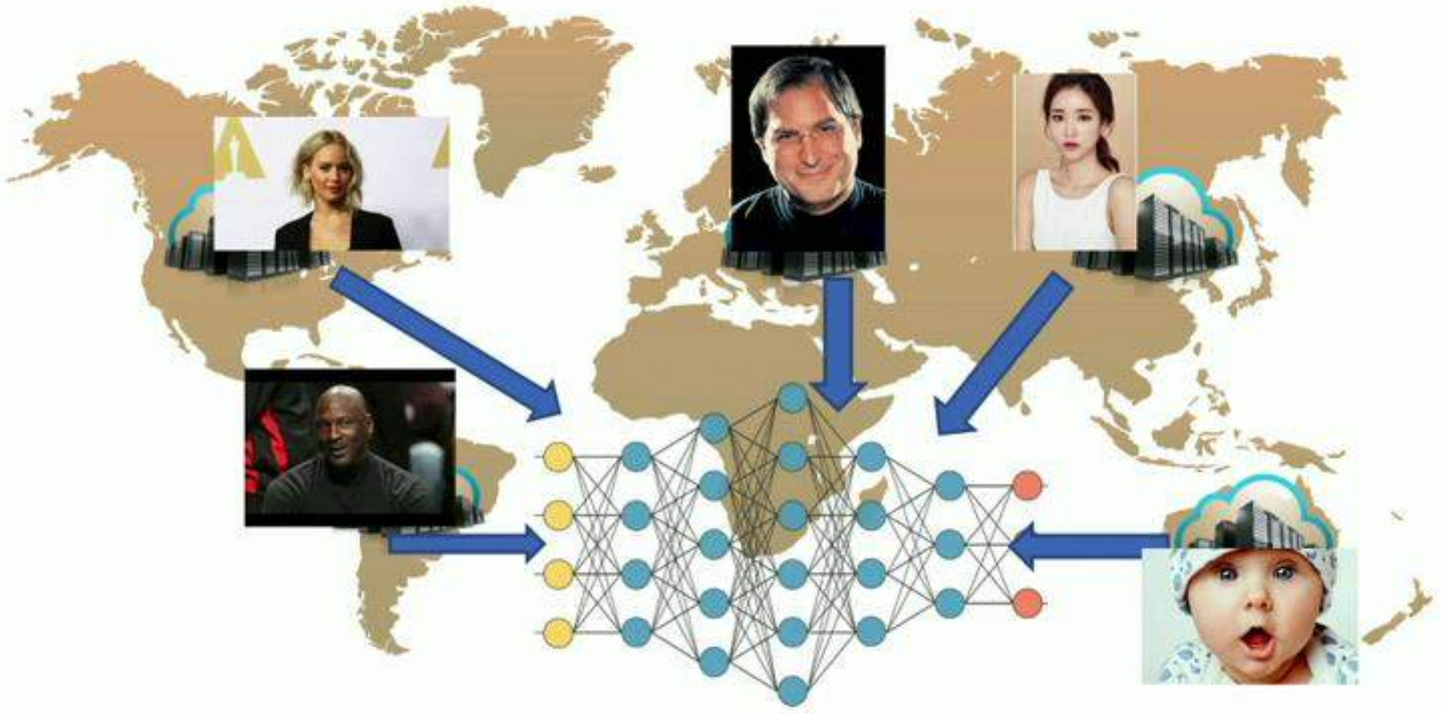
1. **Periodical model travelling**
   - Accuracy gap
   - Underperforming data

Direction #1: Skewed Data over Space

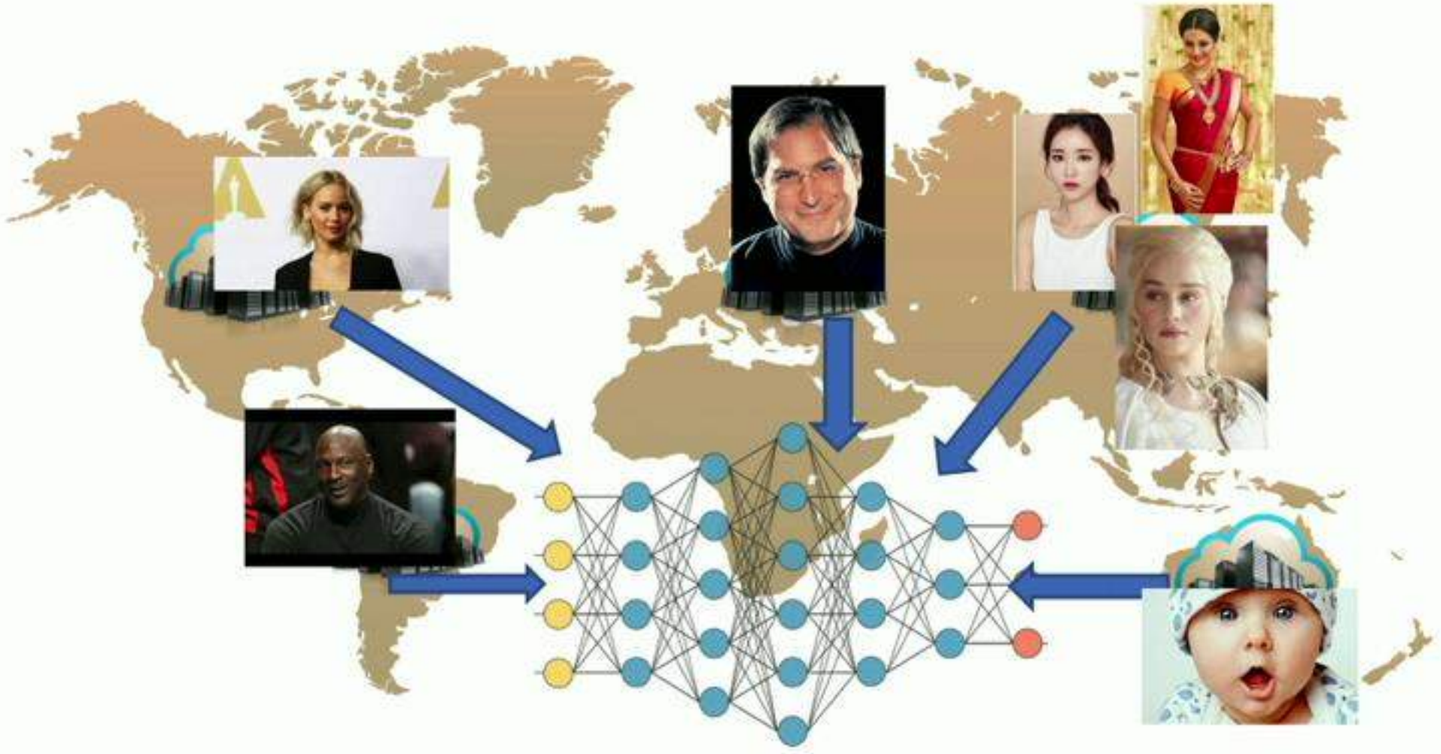
1. **Periodical model travelling**
   - Accuracy gap
   - Underperforming data
2. **Communication tightness control**
3. **Selective data shuffling (active learning)**
4. **Solutions for DNNs with normalizations**



Direction #2: Skewed Data over Spacetime



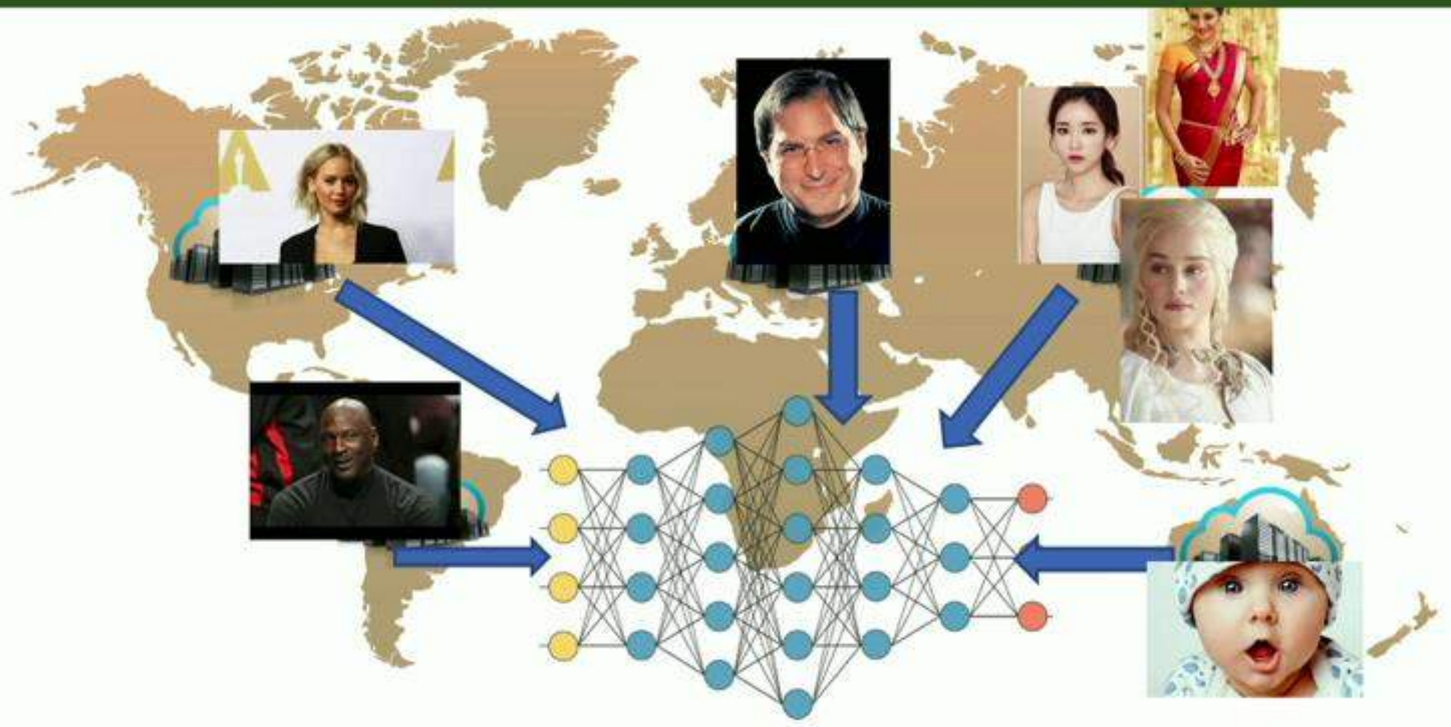
Direction #2: Skewed Data over Spacetime



Direction #2: Skewed Data over Spacetime

1. Continuous learning on skewed data at different time and places

- *Detect data change over time in each space*
- *Efficiently and incrementally update the global model*
- *Tailored to application's requirement on history data*



Direction #2: Skewed Data over Spacetime

1. Continuous learning on skewed data at different time and places

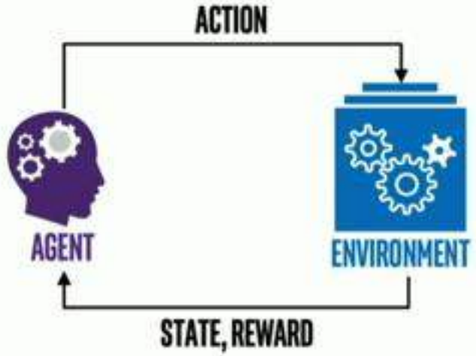
- *Detect data change over time in each space*
- *Efficiently and incrementally update the global model*
- *Tailored to application's requirement on history data*

2. Train the global and local models at the same time

- *Efficient multi-task learning over space and time*

Direction #3: New ML Paradigms over Real-World Data

Reinforcement Learning



AutoML

Cloud AutoML Vision

Rapidly Growing

Distributed and Skewed

Direction #3: New ML Paradigms over Real-World Data

Reinforcement Learning

- Communication efficient RL
- RL adapts to skewed data over spacetime

AutoML

Cloud AutoML Vision

AUTO KERAS

Rapidly Growing

Distributed and Skewed

Direction #3: New ML Paradigms over Real-World Data

Reinforcement Learning

- Communication efficient RL
- RL adapts to skewed data over spacetime

Rapidly Growing

AutoML

Cloud AutoML Vision

- Cost efficient AutoML over distributed data
- AutoML vs decentralized learning approaches

Distributed and Skewed

Machine Learning Systems for Highly Distributed and Rapidly Growing Data

Acknowledgements

Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodik, Amirali Boroumand, Kevin K. Chang, Niladrish Chatterjee, Eiman Ebrahimi, Gregory R. Ganger, Saugata Ghose, Phillip B. Gibbons, Nastaran Hajinazar, Aaron Harlap, Hasan Hassan, Abhilasha Jain, Adwait Jog, Abhijith Kashyap,Gwangsun Kim, Samira Khan, Dimitris Konomis, Michael A. Kozuch, Donghyuk Lee, Tianshi Li, Brandon Lucia, Diptesh Majumdar, Krishna T. Malladi, Todd C. Mowry, Onur Mutlu, Mike O'Connor, Minesh Patel, Gennady Pekhimenko, Amar Phanishayee, Matthai Philipose, Vivek Seshadri, Ashish Shrestha, Shivaram Venkataraman, Nandita Vijaykumar, Hongzhong Zheng