CoEdge: A Cooperative Edge System for Distributed Real-Time Deep Learning Tasks

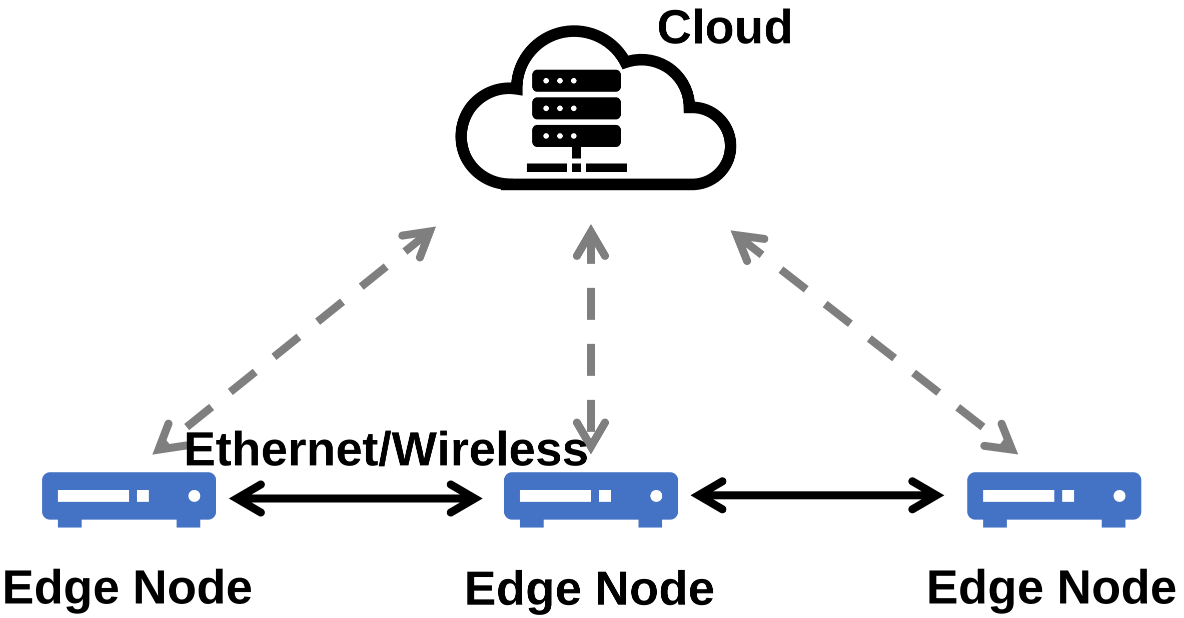
Zhehao Jiang*, **Neiwen Ling*,** Xuan Huang, Shuyao Shi, Chenhao Wu, Xiaoguang Zhao, Zhenyu Yan, and Guoliang Xing

* Co-first author, Presenter

The Chinese University of Hong Kong

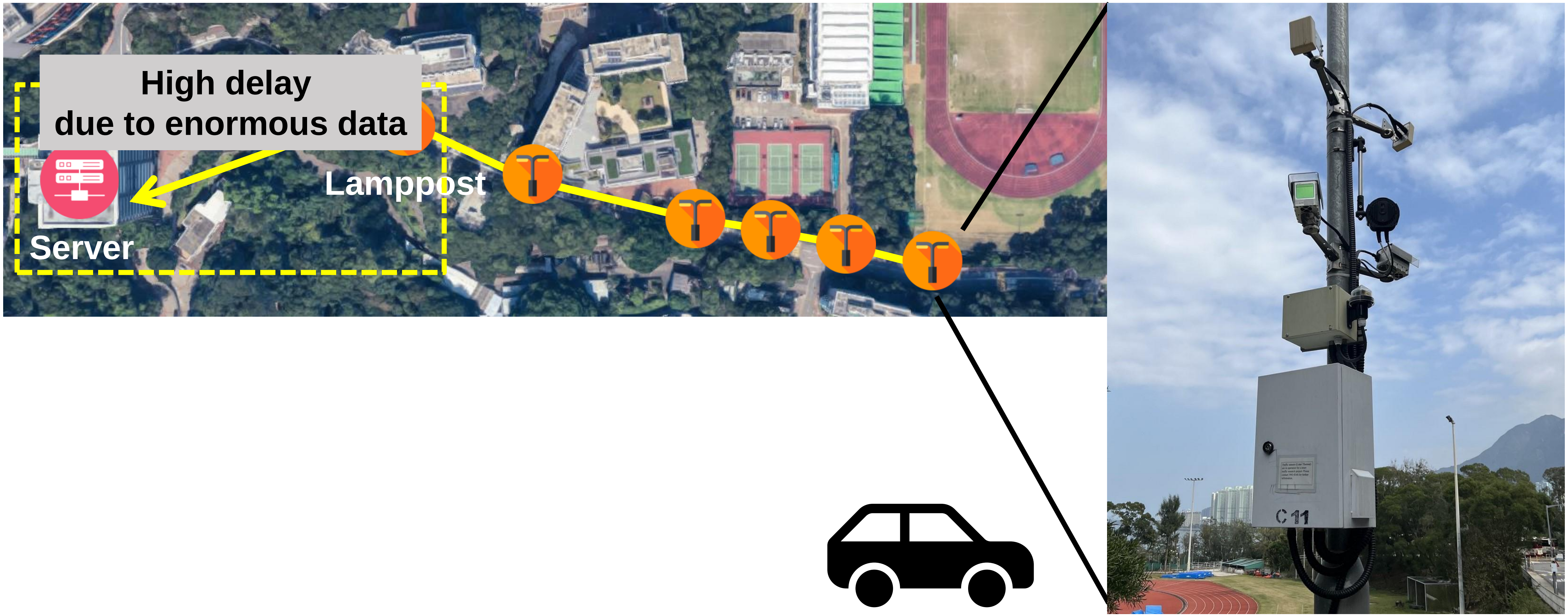
ACM/IEEE IPSN 2023

Cooperative edge systems

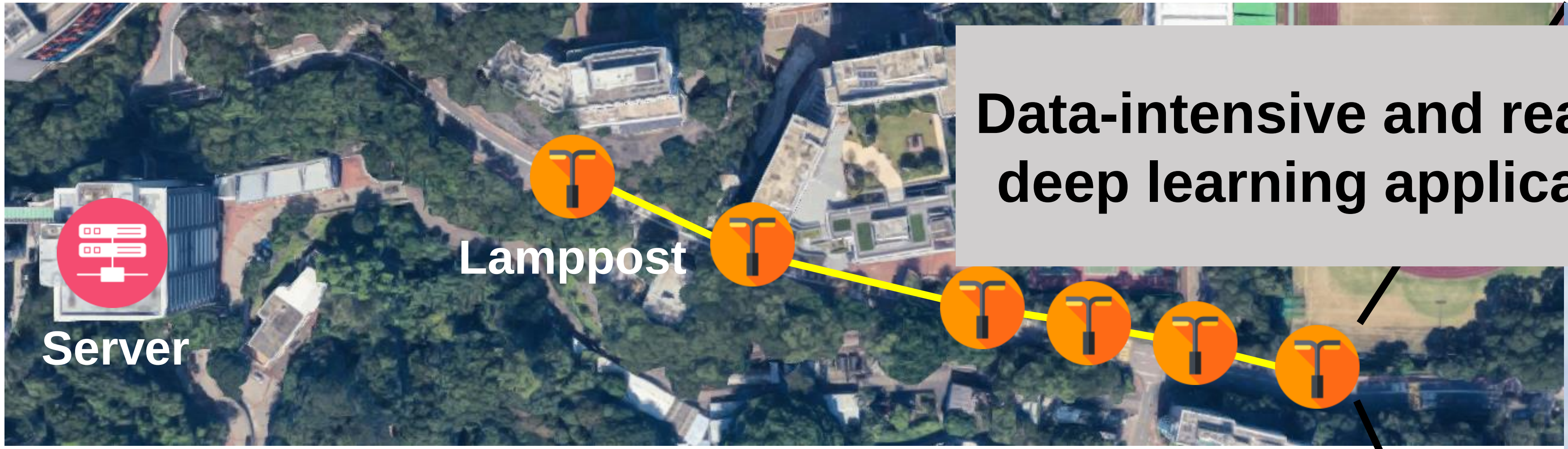


- Cloud-edge connectivity
- Local peer-to-peer connectivity

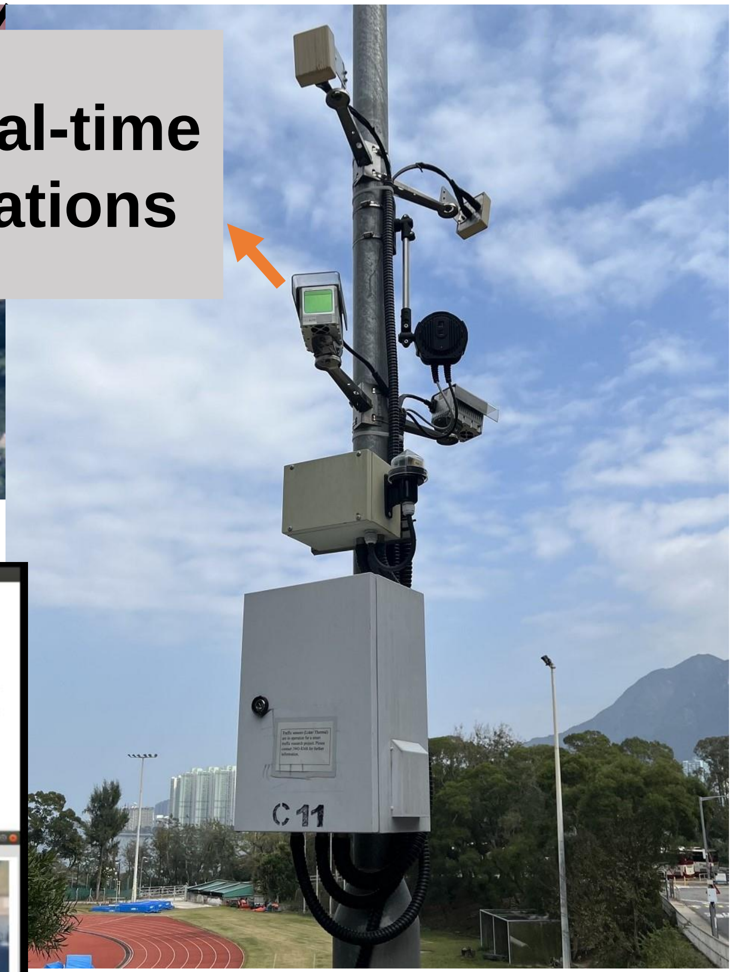
An example of cooperative edge

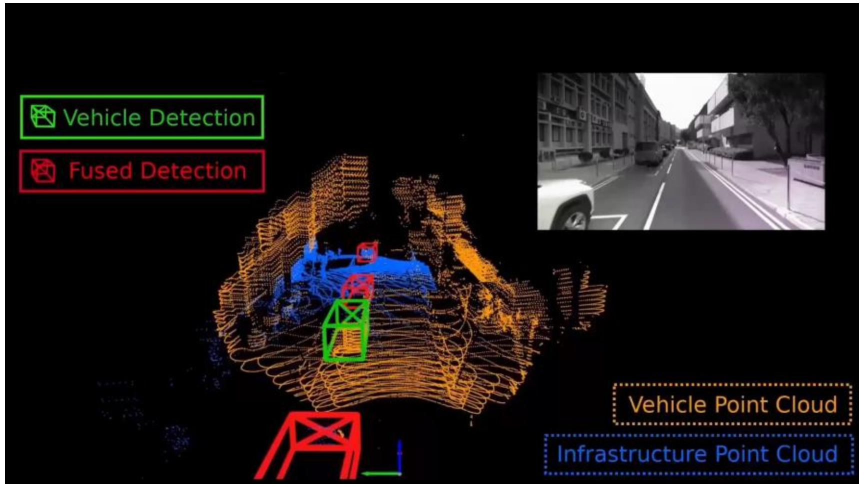


An example of cooperative edge



Data-intensive and real-time deep learning applications





Vehicle detection

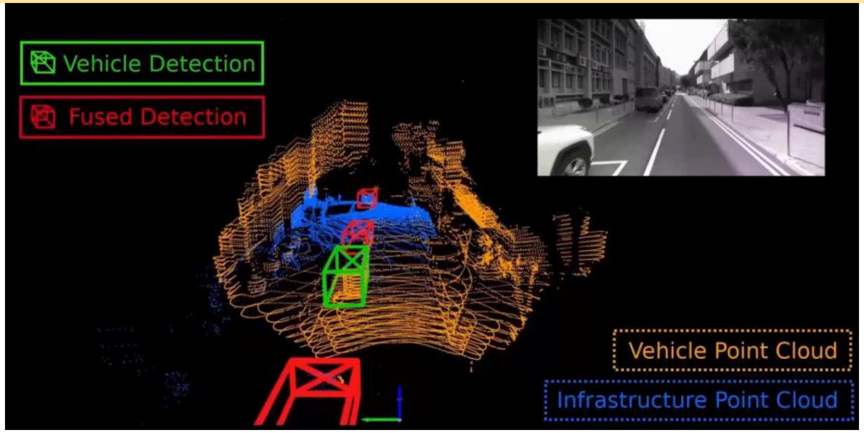
Traffic flow mapping

Video source: CUHK AIoT Lab

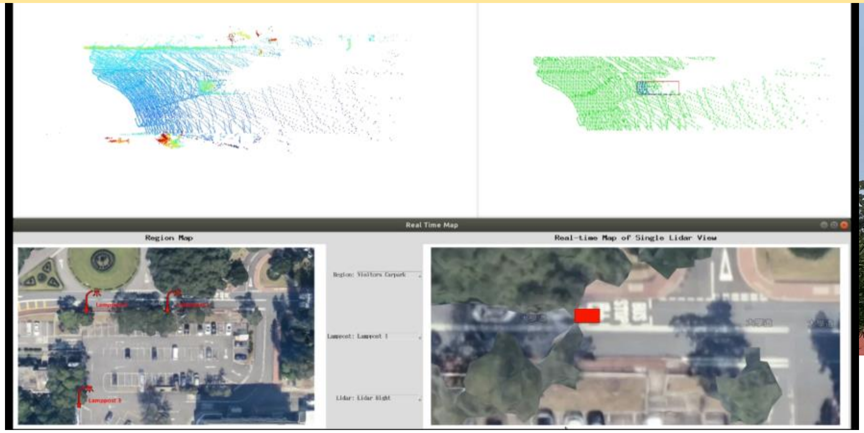
An example of cooperative edge

Data-intensive and real-time deep learning applications

How to support multiple, distributed **deep learning** tasks on **cooperative edge**?



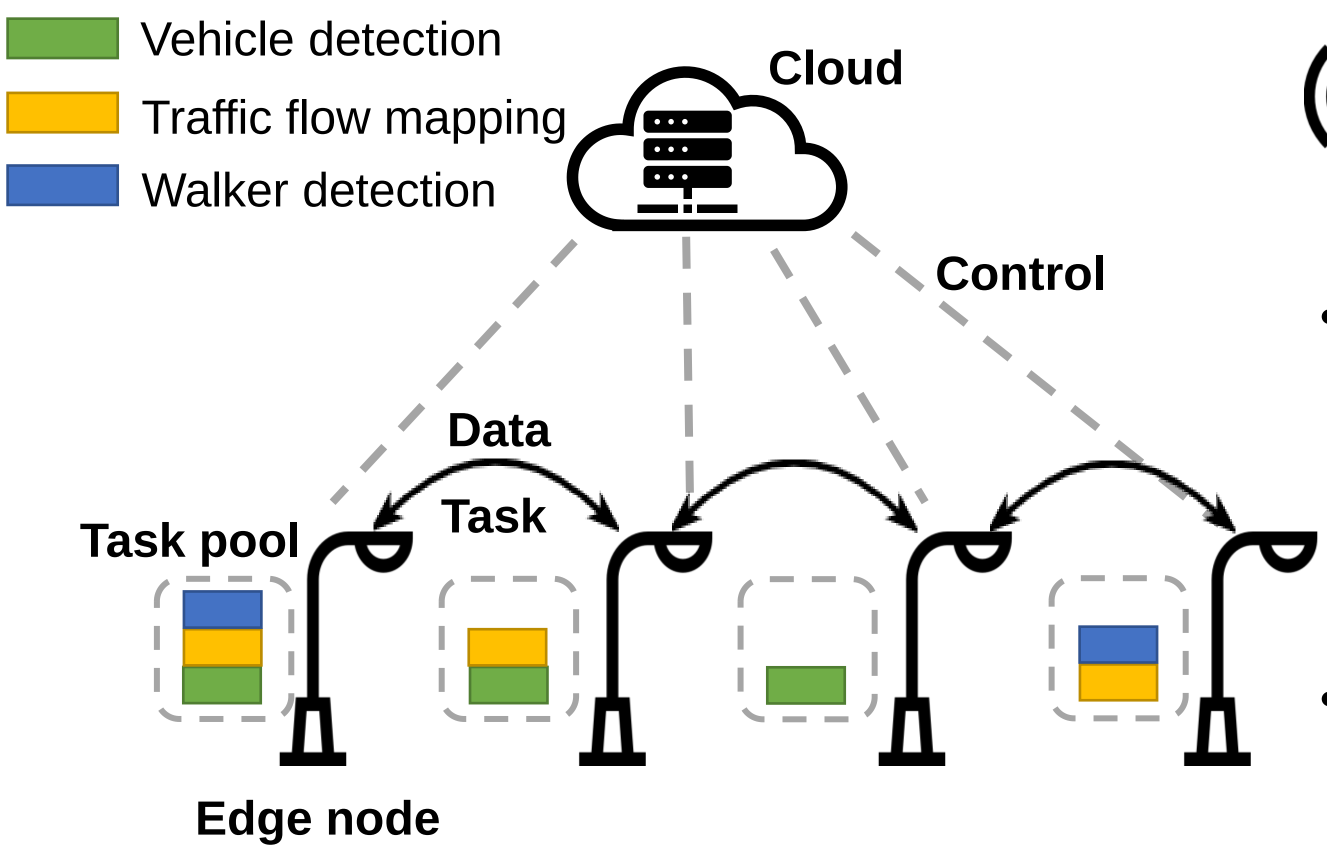
Vehicle detection



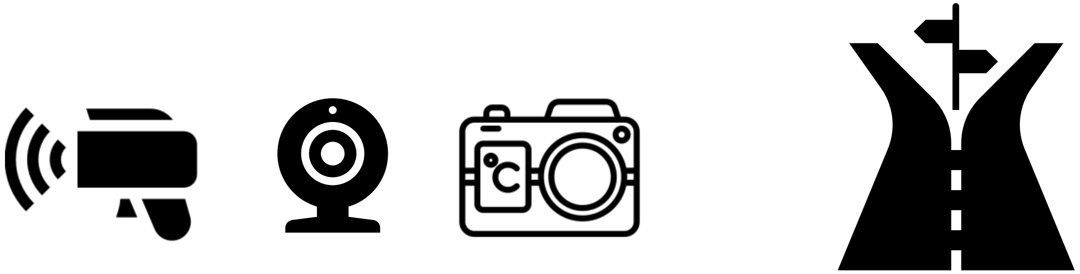
Traffic flow mapping

Video source: CUHK AIoT Lab

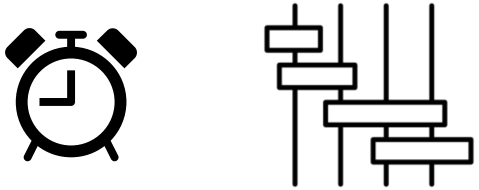
Deep learning on cooperative edge



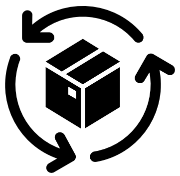
- **Geo-distributed, uneven workloads**



- **Real-time and concurrent DNN execution**



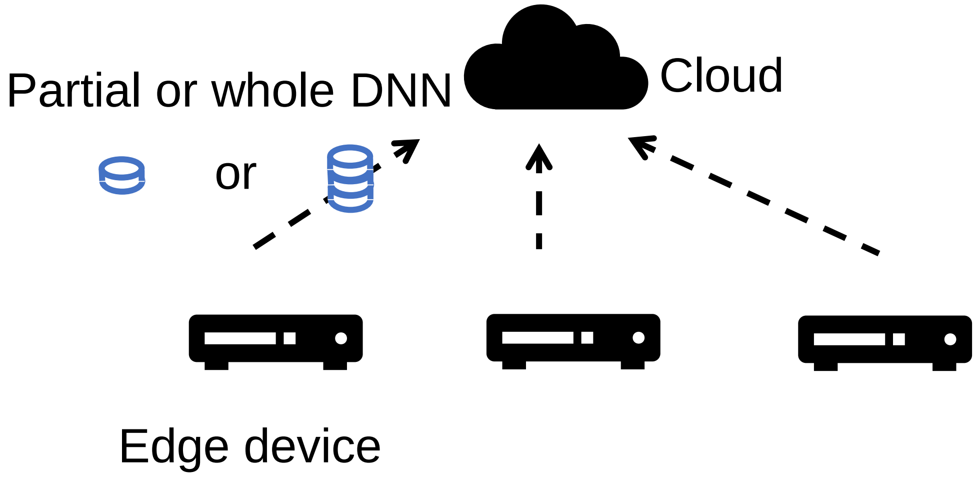
- **Diverse software environments**



Existing approaches

- DL Task Offloading

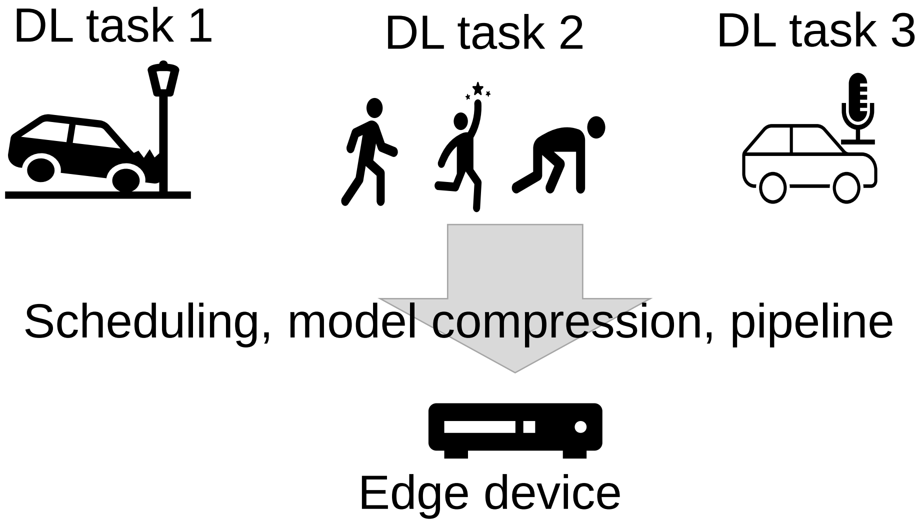
  [MobiCom ’20, IoTDI ’21]



- Rely on high-bandwidth communication
- Privacy leakage

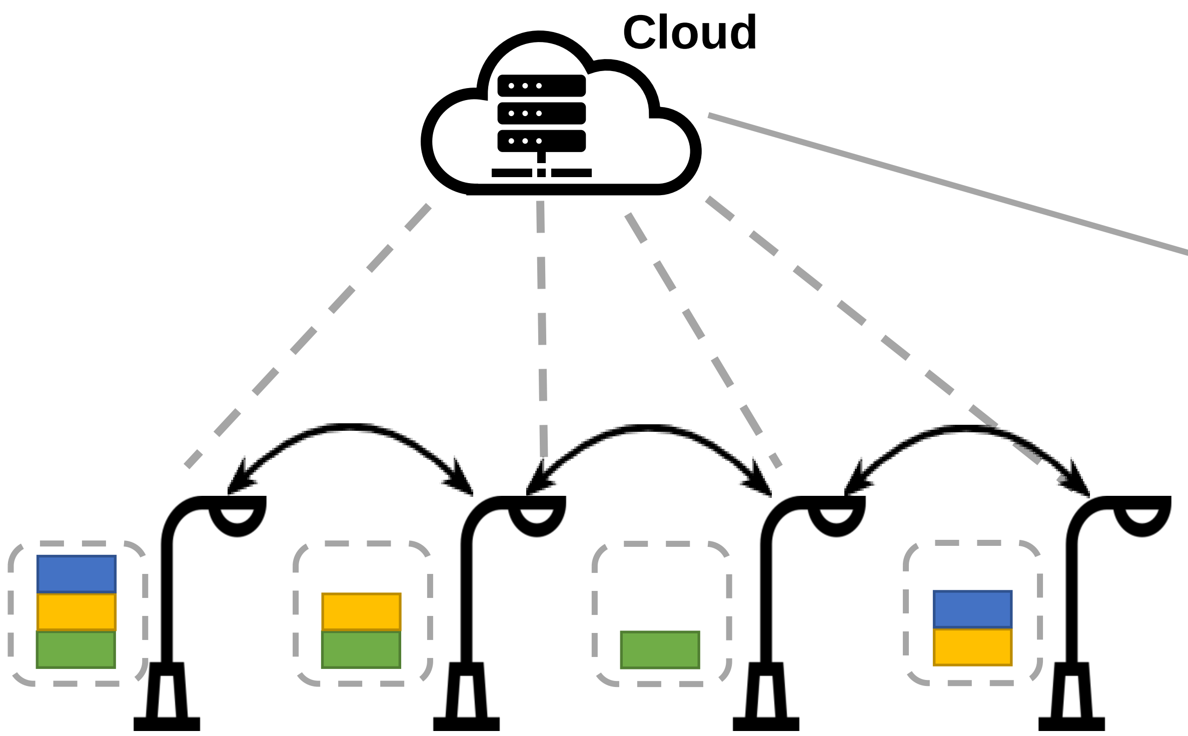
- Concurrent DL Tasks on the Edge

  [RTSS ’ 19, SenSys ’21]

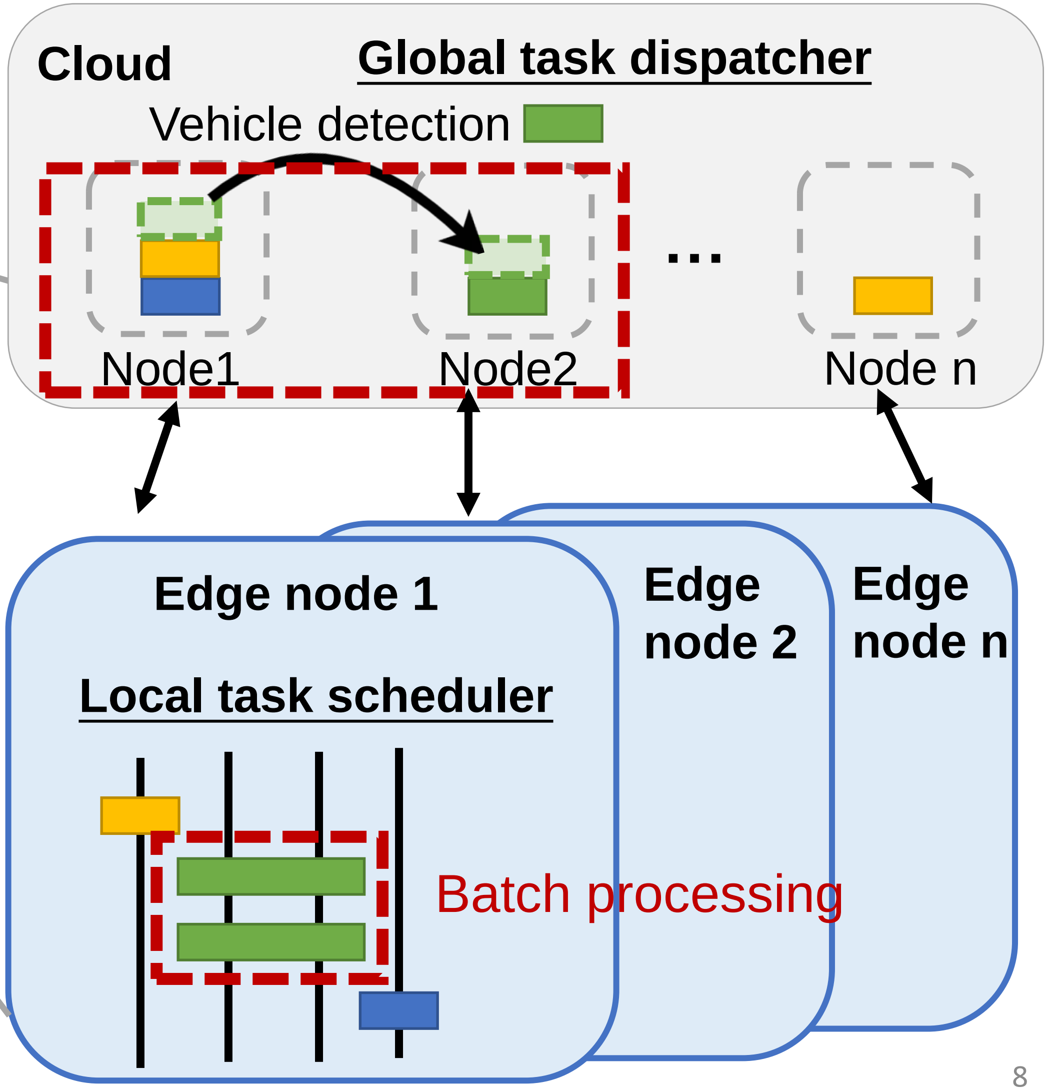


- Focus on single edge device
- Do not consider the tasks in a cooperative manner

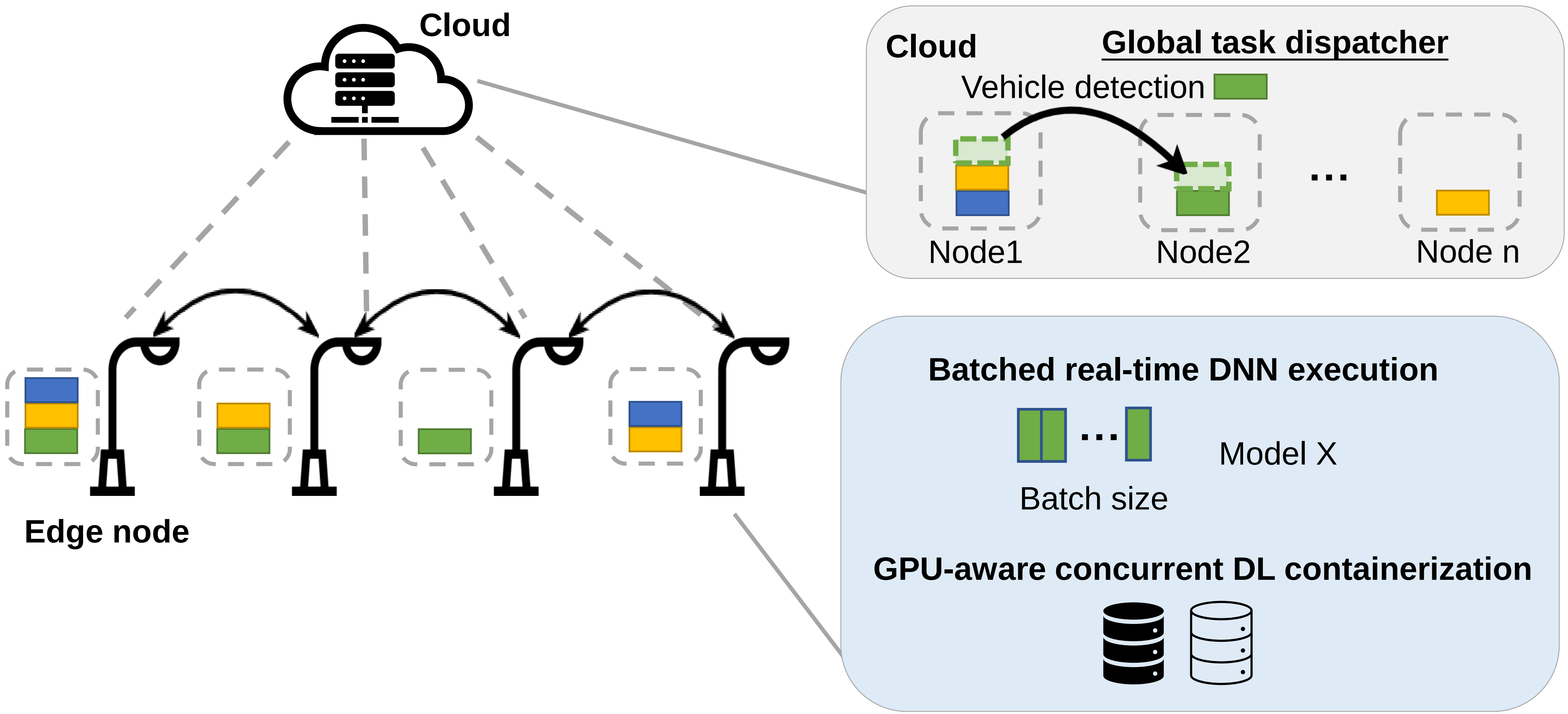
CoEdge: a new cooperative edge system



Hierarchical DL task scheduling



CoEdge: a new cooperative edge system



Global task dispatcher

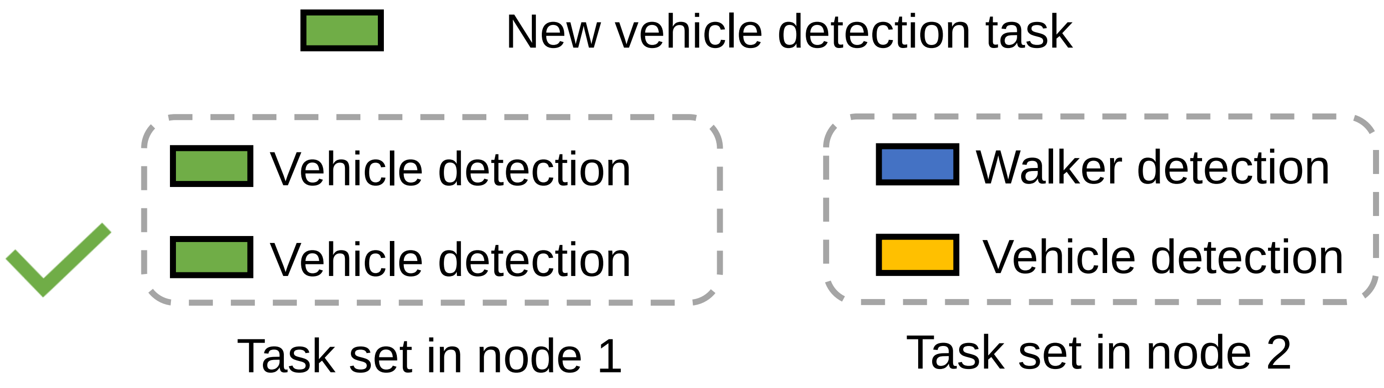
Task deployment rules

1. Estimated end-to-end delay < deadline?

$$T^{comm} + T^{exe} < DDL$$

communication time (T^{comm}) — execution time (T^{exe})

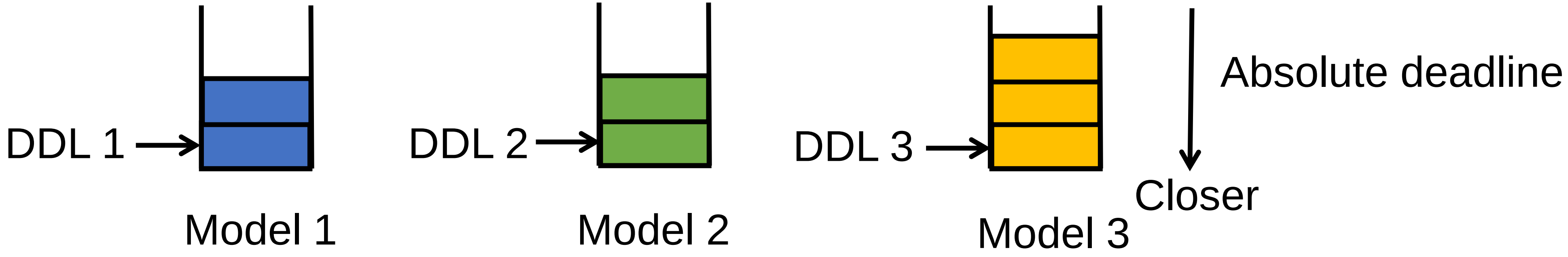
2. Calculate the similarity of new task and existing task set



Batched real-time DNN execution

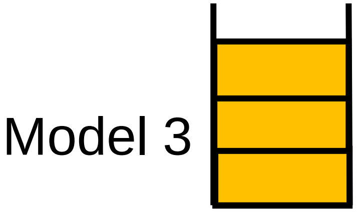
Larger batch size → Higher GPU utilization → More deadline missing

Model queues



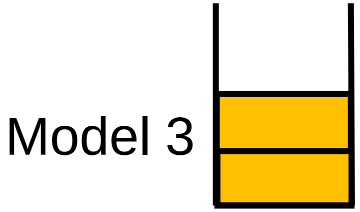
1. Determine the most urgent model

$$argmin(DDL\ 1, DDL\ 2, DDL\ 3)$$



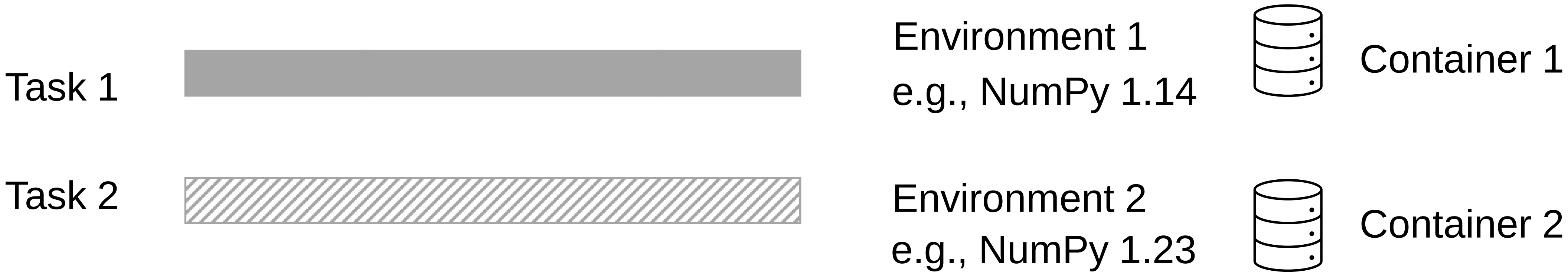
2. Select batch size for the executed model

$$argmax\{CurTime + ExeTime(batch\ size, model\ i) < DDL\ i\}$$



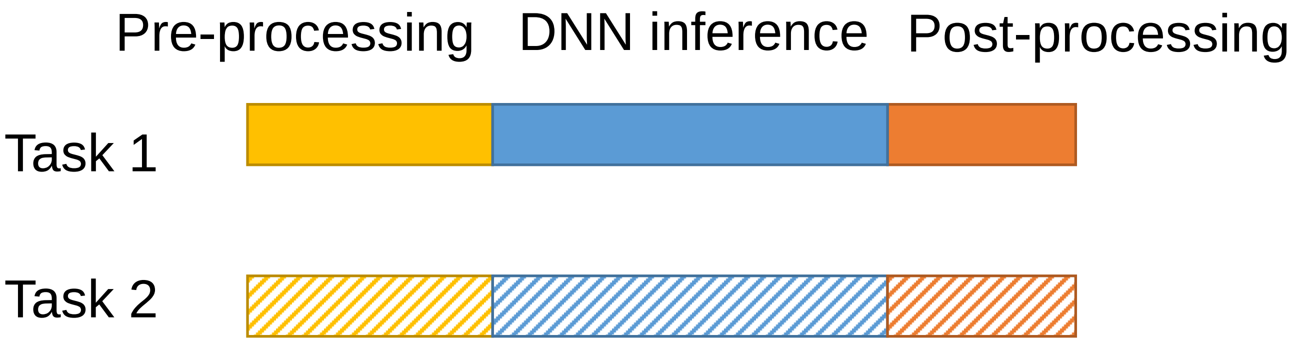
GPU-aware concurrent DL containerization

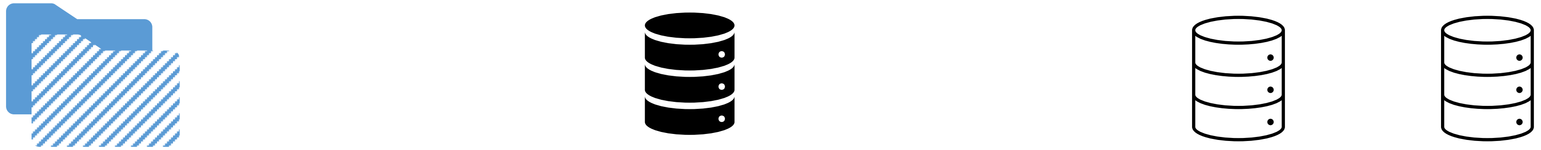
Challenges: multiple containers cannot access the same edge GPU at the same time



GPU-aware concurrent DL containerization

Challenges: multiple containers cannot access the same edge GPU at the same time





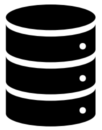
GPU-aware concurrent DL containerization

Challenges: multiple containers cannot access the same edge GPU at the same time

Pre-processing DNN inference Post-processing

- **Improve GPU utilization**
- **Support different environments**

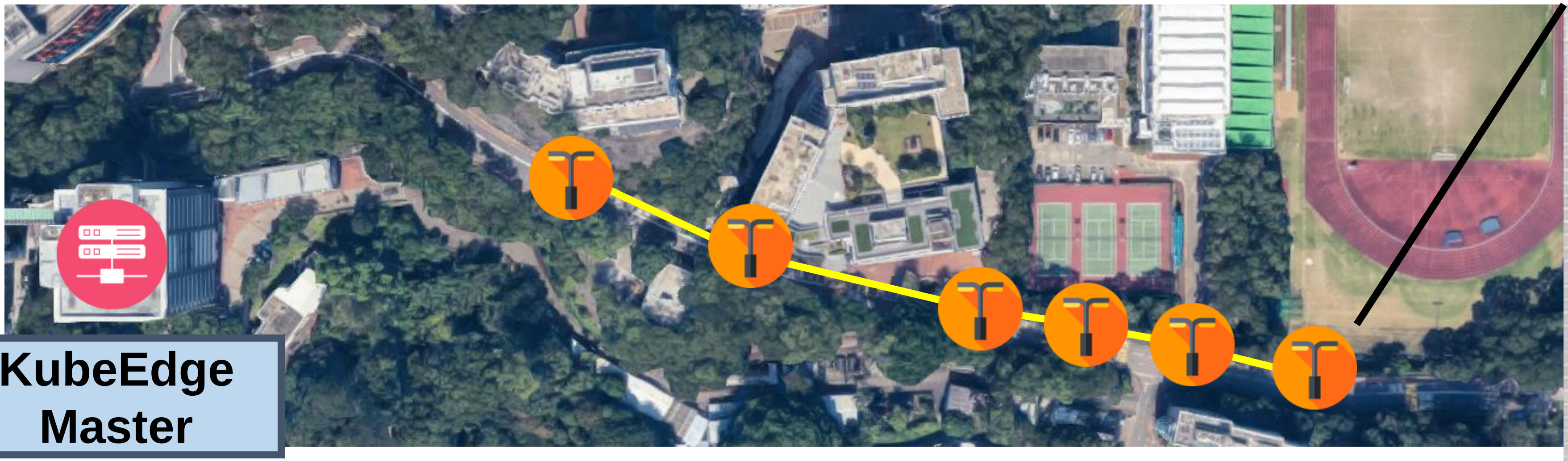
ONNX transformation



Shared container

Independent task container

Implementation of CoEdge



KubeEdge Master

Integrate open-source components

Master node: collaborates with the KubeEdge master for global task dispatching

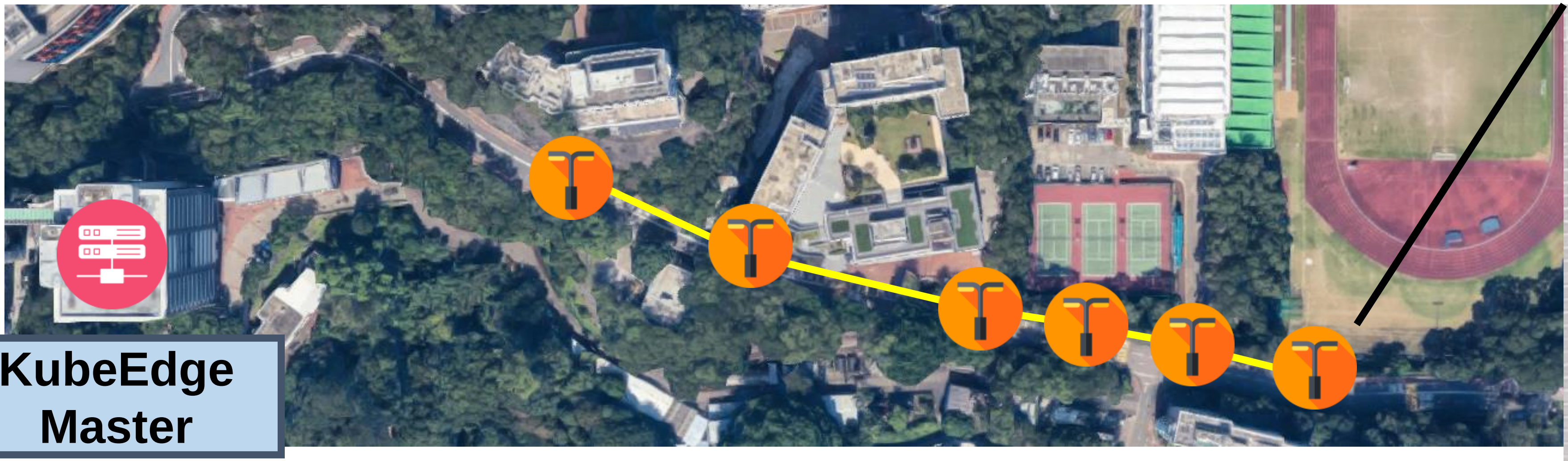
Edge node: Work containers for inference, pre/post processing, data containers, KubeEdge worker

KubeEdge Worker

Work containers

Data containers(ROS2)

Implementation of CoEdge



KubeEdge Master

Integrate open-source components

ROS2

Master ... eEdge master

for glol ...

More autonomous task deployment

Edge r ... nce, pre/post

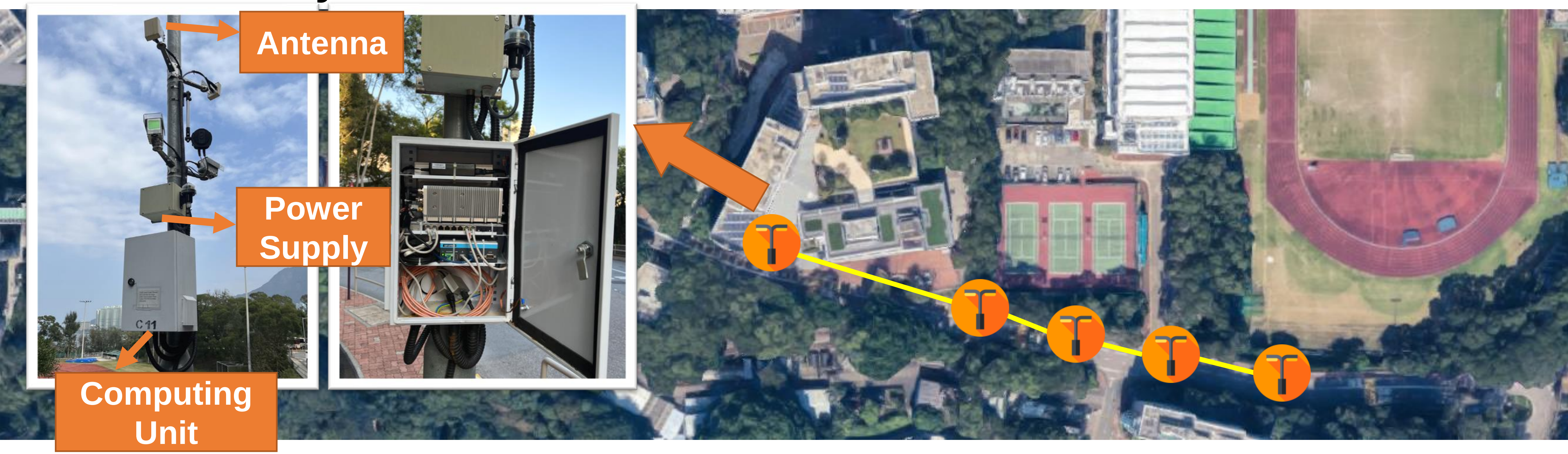
processing, data containers, KubeEdge worker

KubeEdge Worker

Work containers

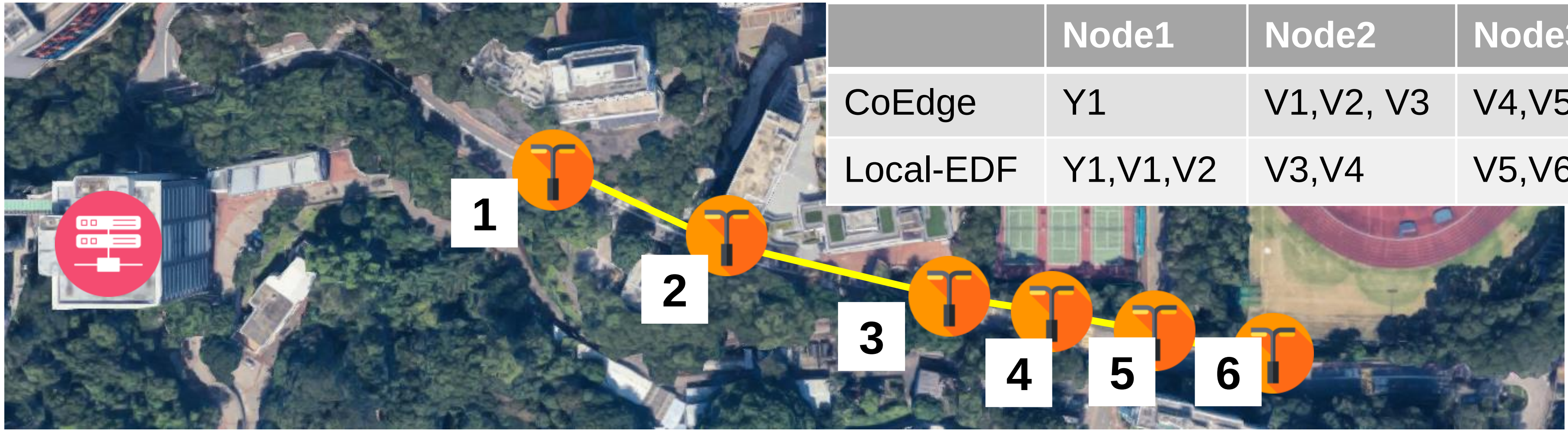
Data containers(ROS2)

End-to-end system evaluation



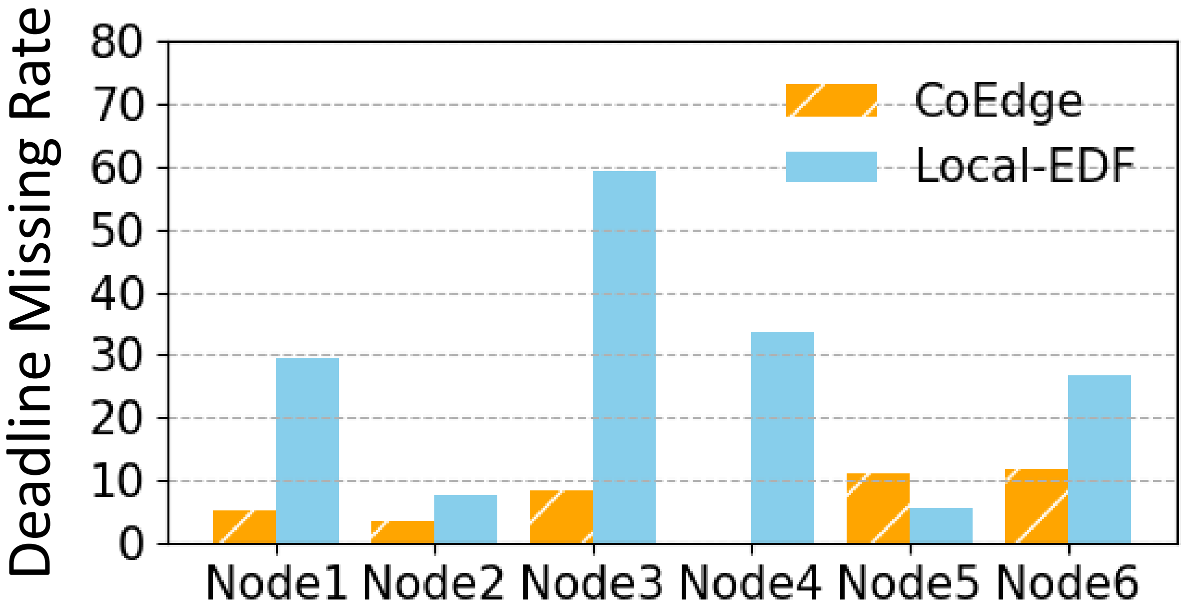
- Deploy CoEdge on an **outdoor smart lamppost network** located at CUHK
- Designed for supporting smart traffic in campus, operational **for over two years**
- DL Apps: real-time traffic monitoring, walker recognition, and vehicle recognition

End-to-end system evaluation



	Node1	Node2	Node3	Node4	Node5	Node6
CoEdge	Y1	V1,V2, V3	V4,V5,V6	Y2	R1,R2	Y3
Local-EDF	Y1,V1,V2	V3,V4	V5,V6	Y2	R1	Y3,R2

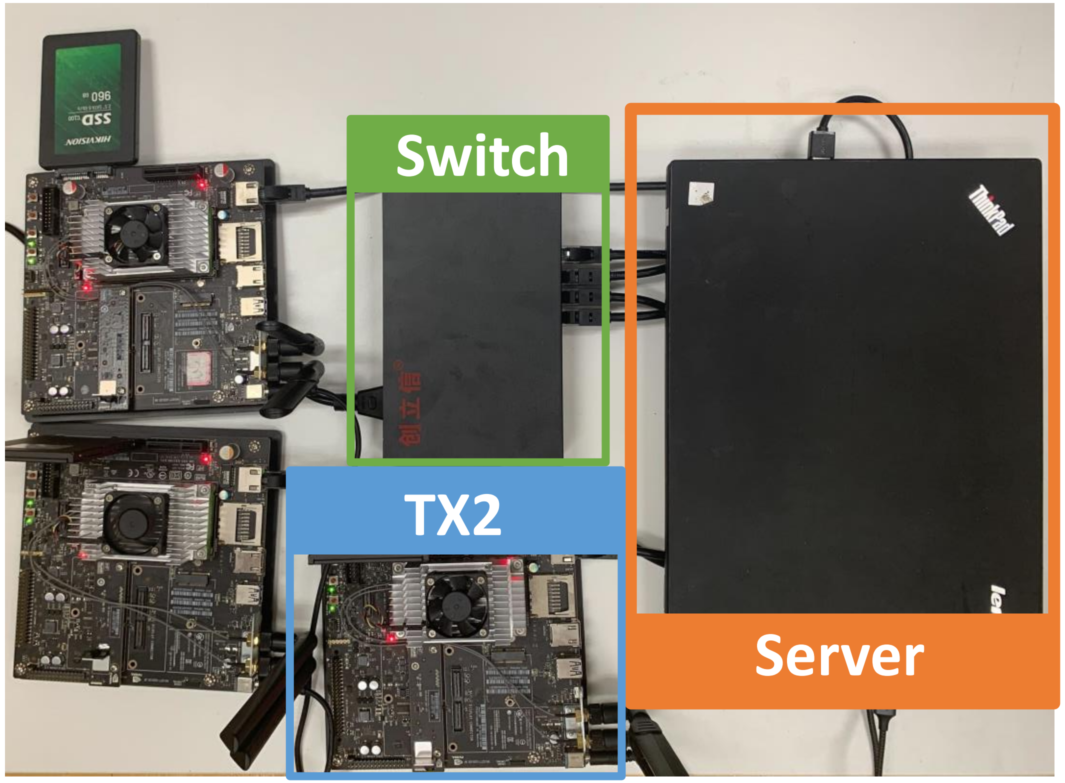
Y: Traffic monitoring
V: Walker recognition
R: Vehicle recognition



- Reduce deadline missing rate up to 50.99% without any accuracy loss

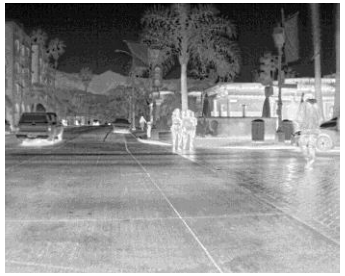
Indoor experiment setup

- **Indoor platform**

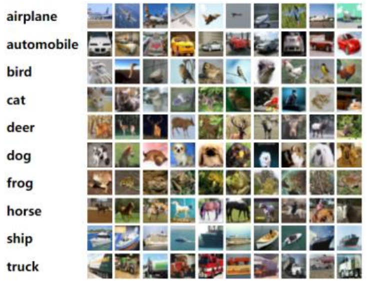


- **4 types of DL tasks**
- **3 DNN models: YOLO, VGG, ResNet**
- **3 datasets: COCO, CIFAR10, Teledyne FLIR ADAS**

COCO



Teledyne FLIR ADAS



CIFAR10

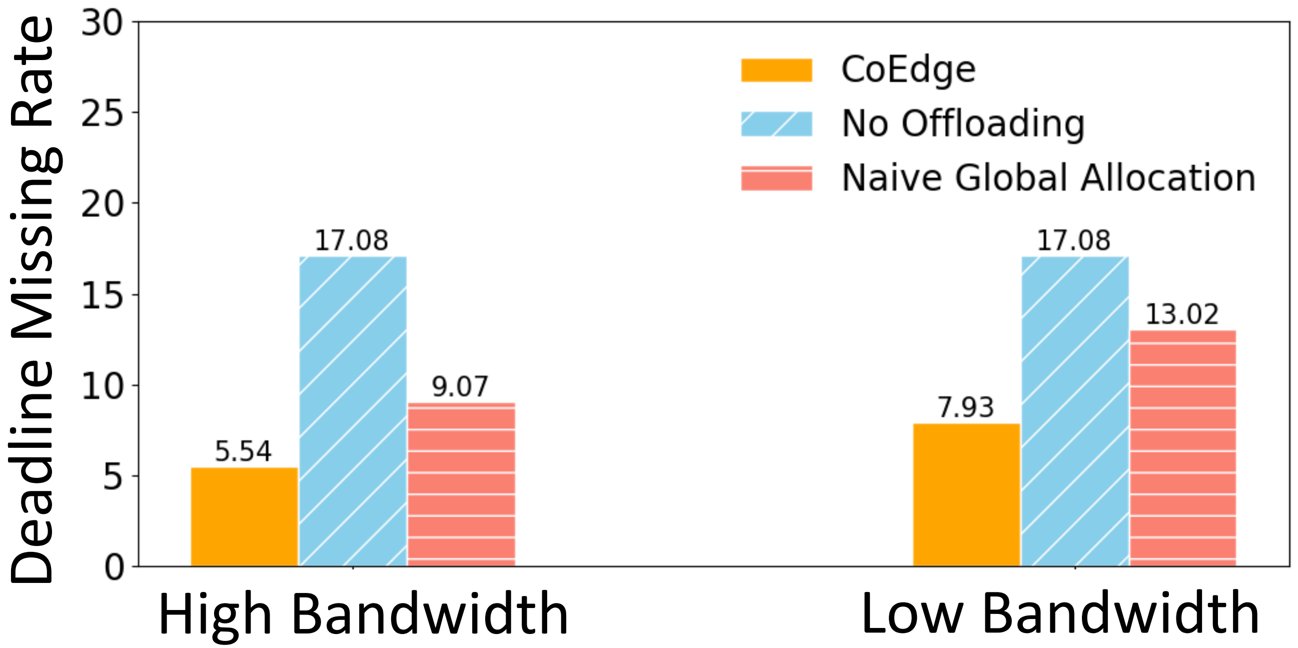
CoEdge under different network bandwidth

YB: YOLO-Big, YS: YOLO-Small

	Node1	Node2	Node3
No offloading	YB1, YB2, YS1, YS2	YS3	YS4
CoEdge	YB1,YB2	YS1, YS2	YS3, YS4

Task with smaller input

Baselines
- No offloading
execute DL tasks **locally**
- Naïve global allocation:
allocate DL tasks only based on estimated real-time performance



CoEdge
- Adapt to different bandwidth
- Maintain a deadline missing rate below 10%

Conclusion

❑ CoEdge

- ➢ A **hierarchical DL task scheduling** framework for efficient distributed DL
- ➢ **Batched** DNN execution mechanism
- ➢ GPU-aware concurrent DL **containerization** approach
- ➢ Implementation on a self-deployed **smart lamppost testbed**

❑ Future Work

- ➢ Integration with edge-cloud offloading
- ➢ Scalability to large-scale applications

Thanks for listening!

Any questions?

- *Check our paper:* CoEdge: A Cooperative Edge System for Distributed Real-Time Deep Learning Tasks
- *Authors:* Zhehao Jiang*, **Neiwen Ling***, Xuan Huang, Shuyao Shi, Chenhao Wu, Xiaoguang Zhao, Zhenyu Yan, Guoliang Xing
- *Website:* http://aiot.ie.cuhk.edu.hk