RT-mDL: Supporting Real-Time Mixed Deep Learning Tasks on Edge Platforms

Neiwen Ling[1], Kai Wang[1], Yuze He[1], Guoliang Xing[1], Daqi Xie[2]

The Chinese University of Hong Kong[1], Edge Cloud Innovation Lab, Huawei[2]

ACM SenSys 2021, November 15-17, 2021

Deep Learning models increasingly run on the edge

Autonomous driving

Image source: Getty Images

Smart roadside infrastructure

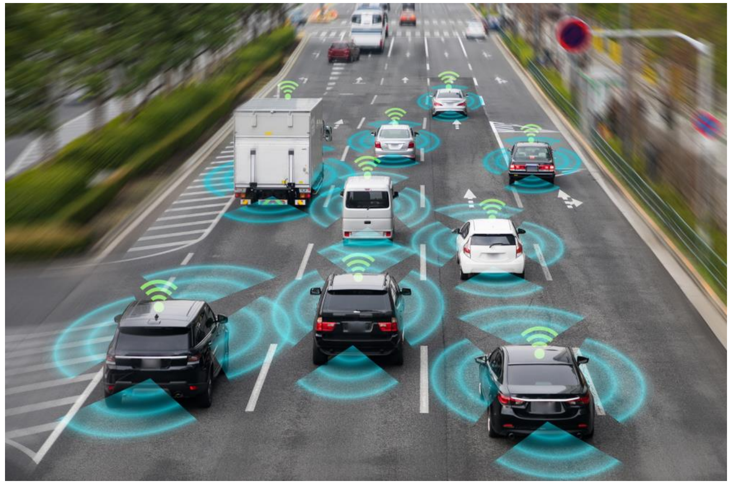


Image source: Shutterstock

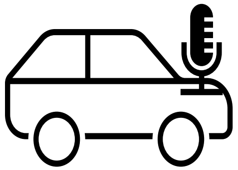
Embedded computer vision

Image source: Bosanski.com

Multiple concurrent DL tasks need to be executed on single edge platform

On-road collisions detection:
SMD-NN

Speech recognition for voice control:
SBCNN



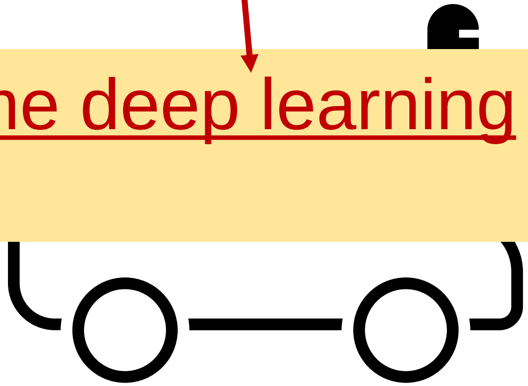
Pedestrian Action Recognition:
3D-ResNet-R18

Current challenge on the edge

- Different DNN model types (YOLO, ResNet…)
- Highly diverse real-time/accuracy requirements

On-road collisions detection

Speech recognition
for voice control



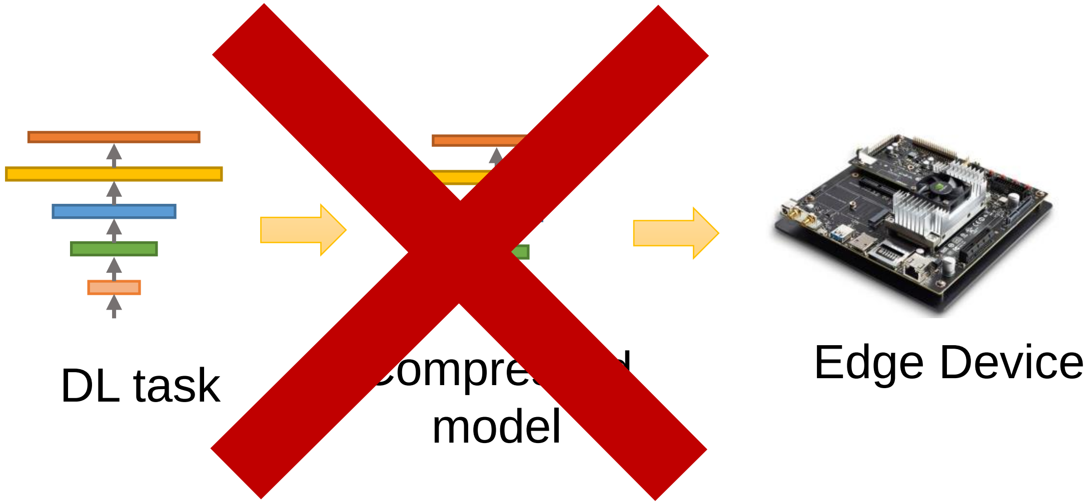
How to efficiently support mixed real-time deep learning
tasks on the edge?

- tight deadline
- low probability of deadline missing
- high accuracy

- loose deadline
- more tolerable

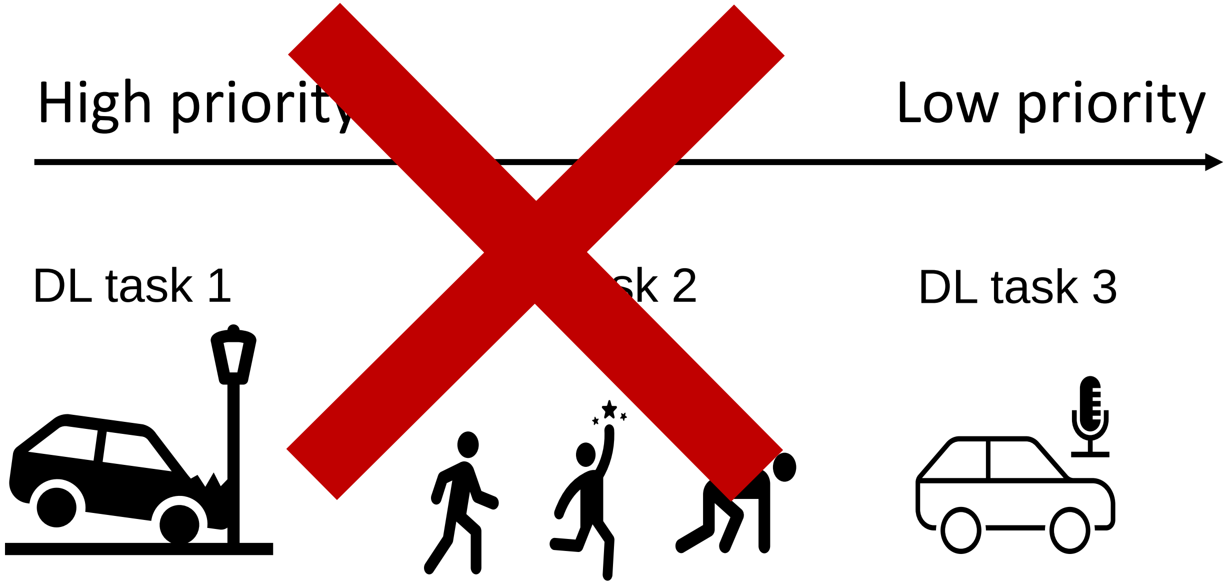
Existing approach

❑ Individual model compression



Doesn't consider
resource contention
among mixed DL tasks

❑ Assign priorities to different tasks

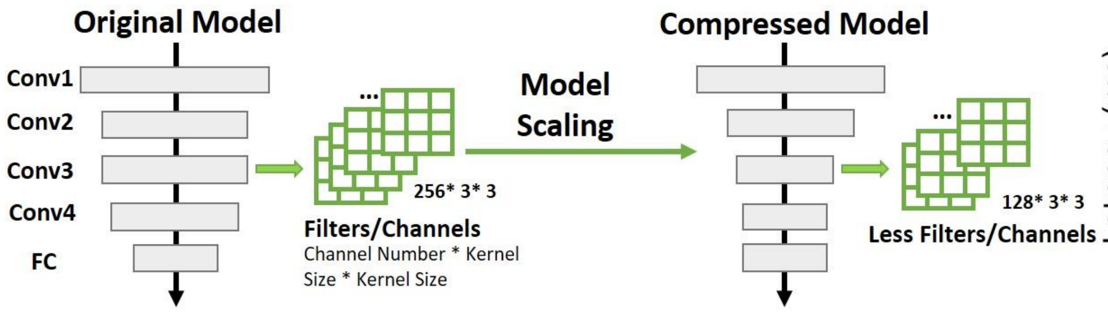


- Treat each task as "black box"
- Low resource utilization

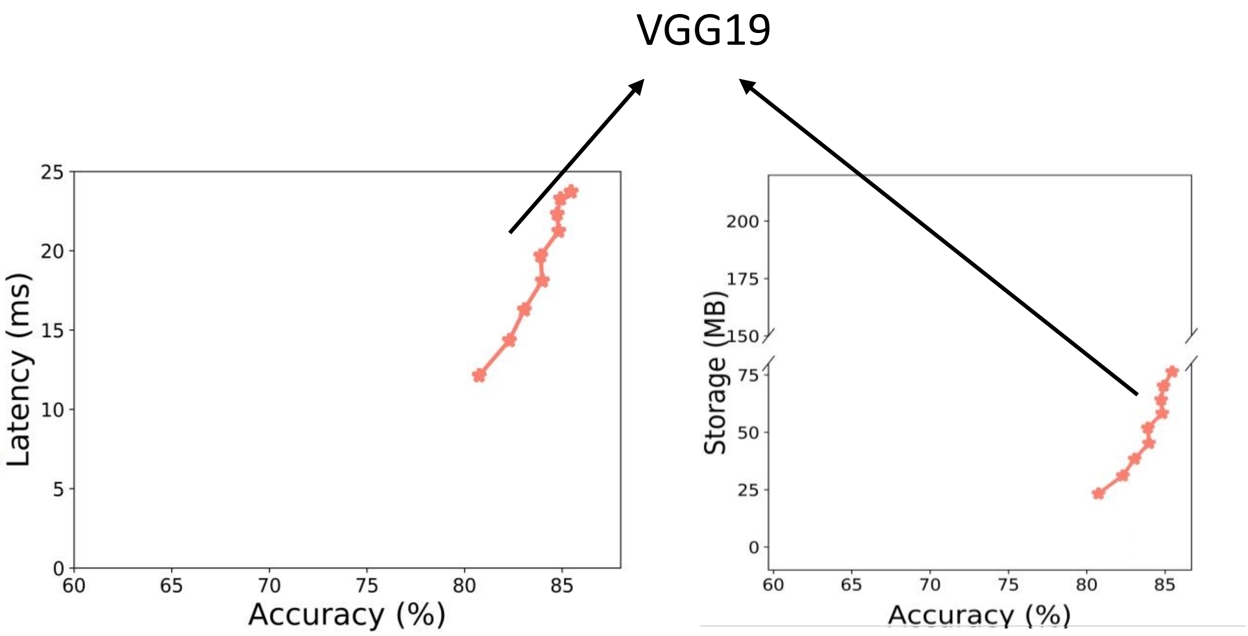
Motivation: Unique characteristics of mixed DL tasks

DNN Model Compressibility

- **Significant trade-off** between latency and accuracy
- Highly **diverse** across different DNN models



❖ DNN model scaling



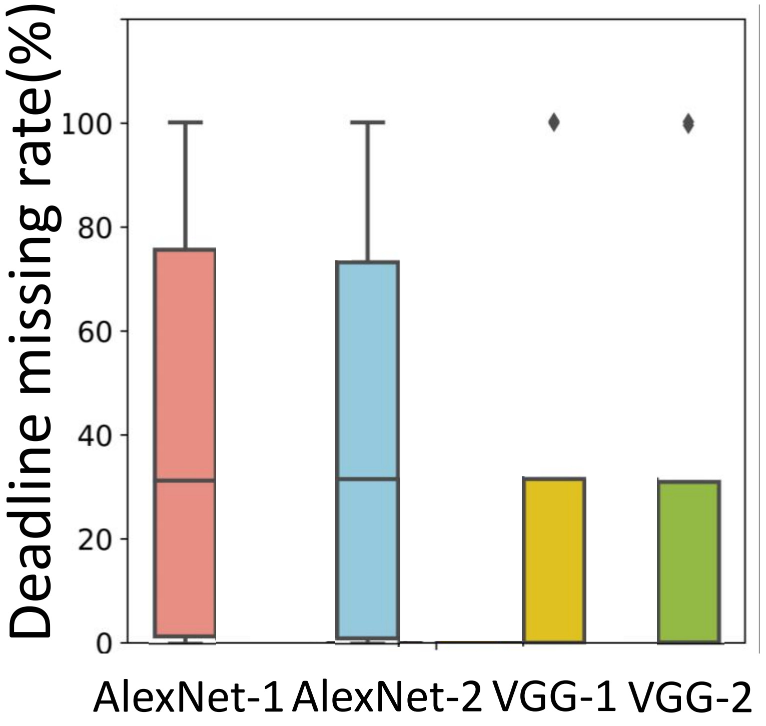
❖ Latency-accuracy trade-off for different DNN models on Xavier

Motivation: Unique characteristics of mixed DL tasks

Real-time performance of mixed DL tasks

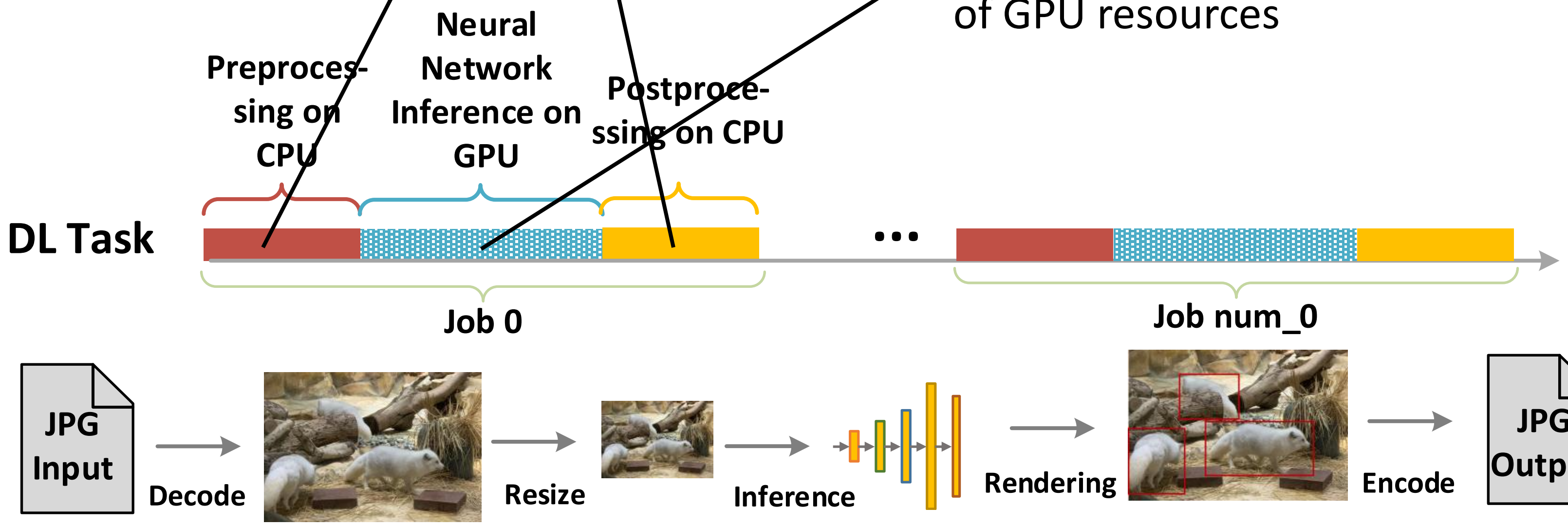
- Highly relies on the **scheduling policy**
- **Task-based scheduling alone is not enough**

High variation under different priorities

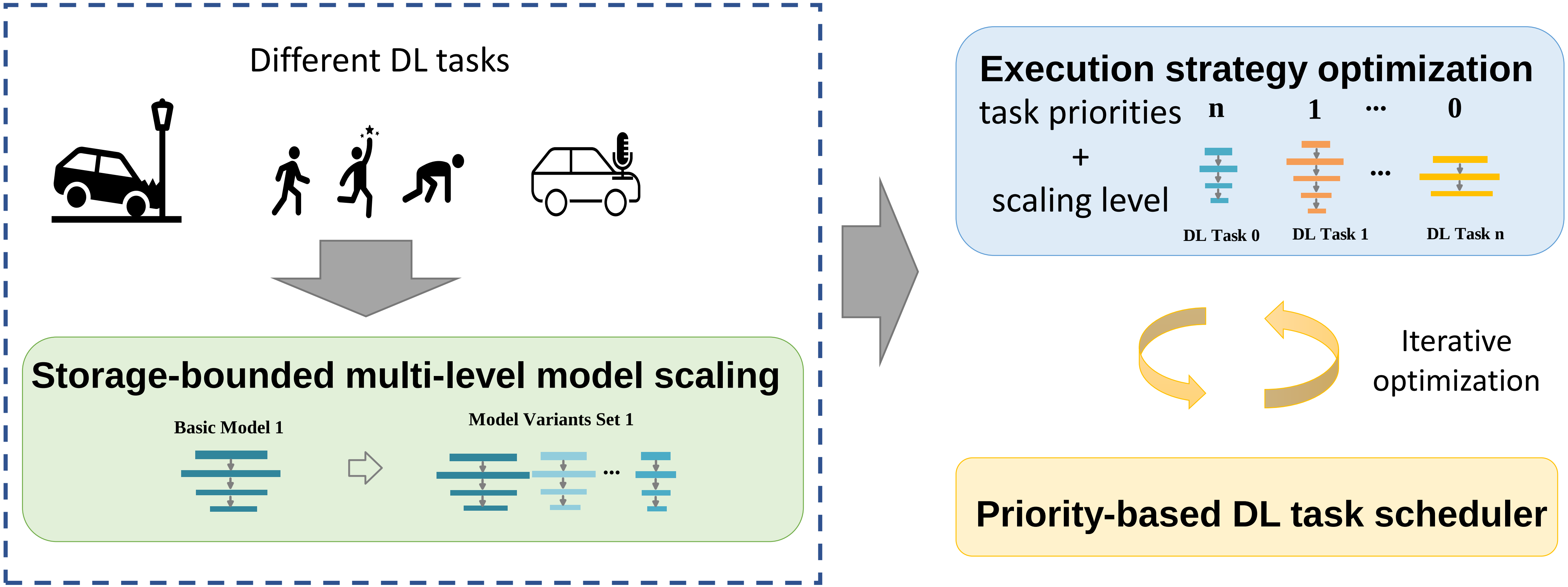


❖ CPU portion in task delay: 30% for AlexNet

❖ 14.54%-36.25% waste of GPU resources



RT-mDL: a new real-time DL framework



Problem Formulation

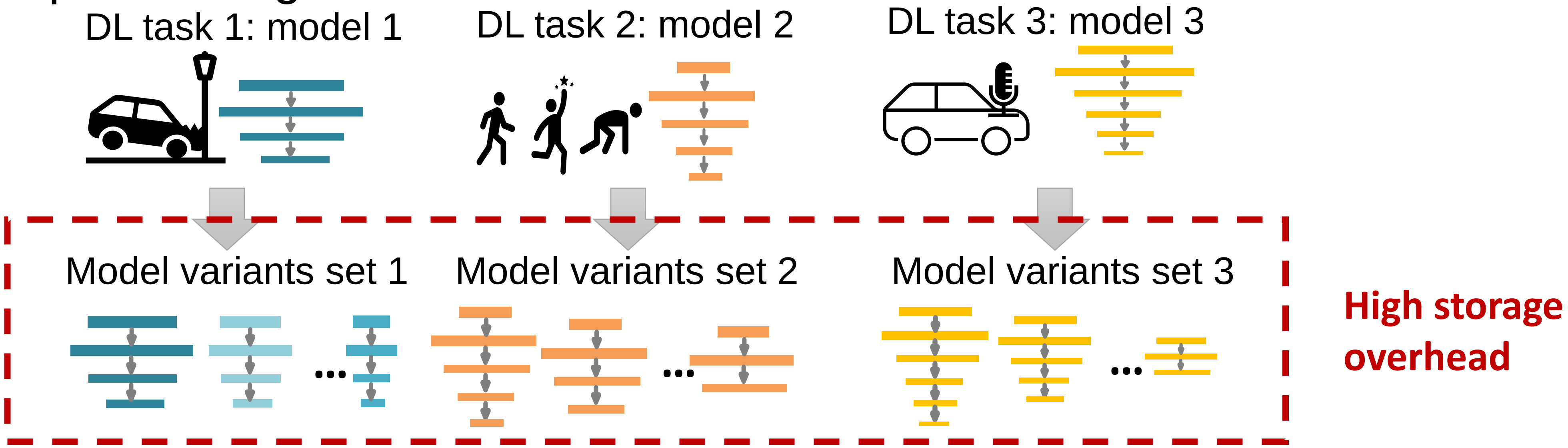
$$\min_{\mathbf{s}} \ \mathrm{LOSS}(\mathbf{s}) = \sum_i \frac{\mathrm{LOSS}_i(\mathbf{s})}{\mathrm{ACC}_i^{max}}$$

$$s.t. \ \ \mathrm{MIS}_i(\mathbf{s}) \le \zeta_i, \quad \sum_i \sum_k Storage(\tilde{N}_{i,k}) \le \overline{Storage}$$

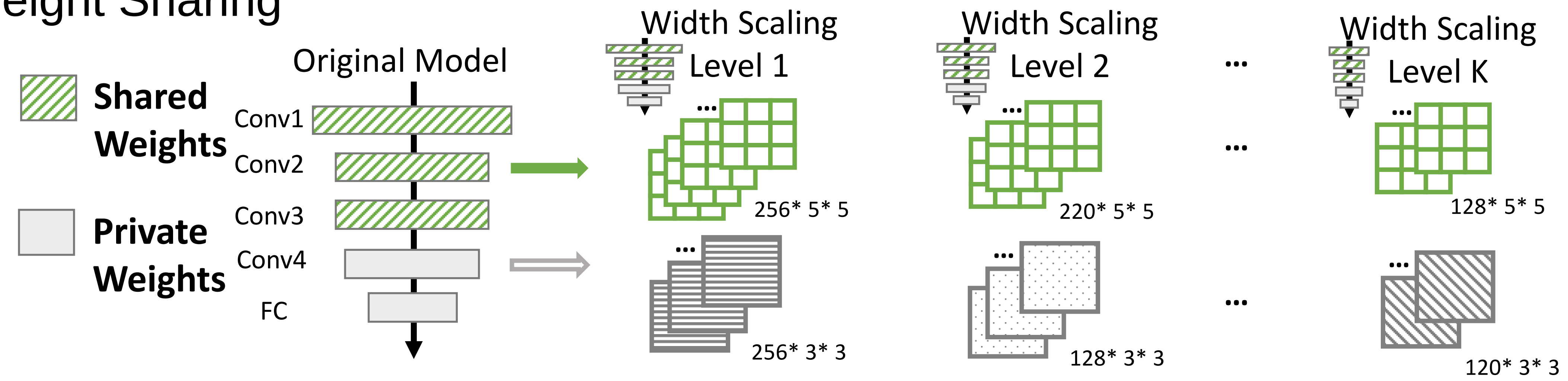
Minimizes the accuracy loss for all DL tasks under the constraints of deadline missing rate and storage bound

RT-mDL component 1: Storage-bounded multi-level model scaling

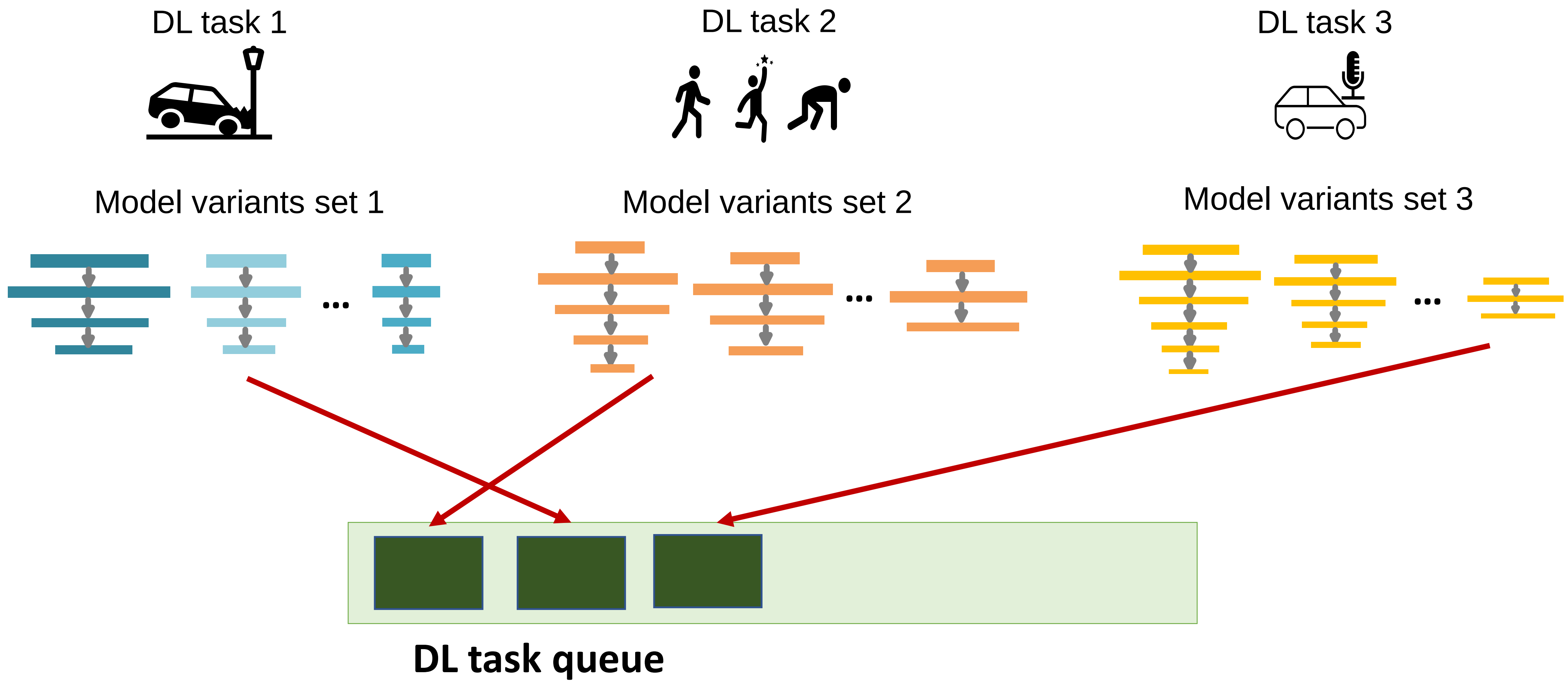
- Width and Depth Scaling



- Weight Sharing



Joint optimization of **scaling levels** and task priorities



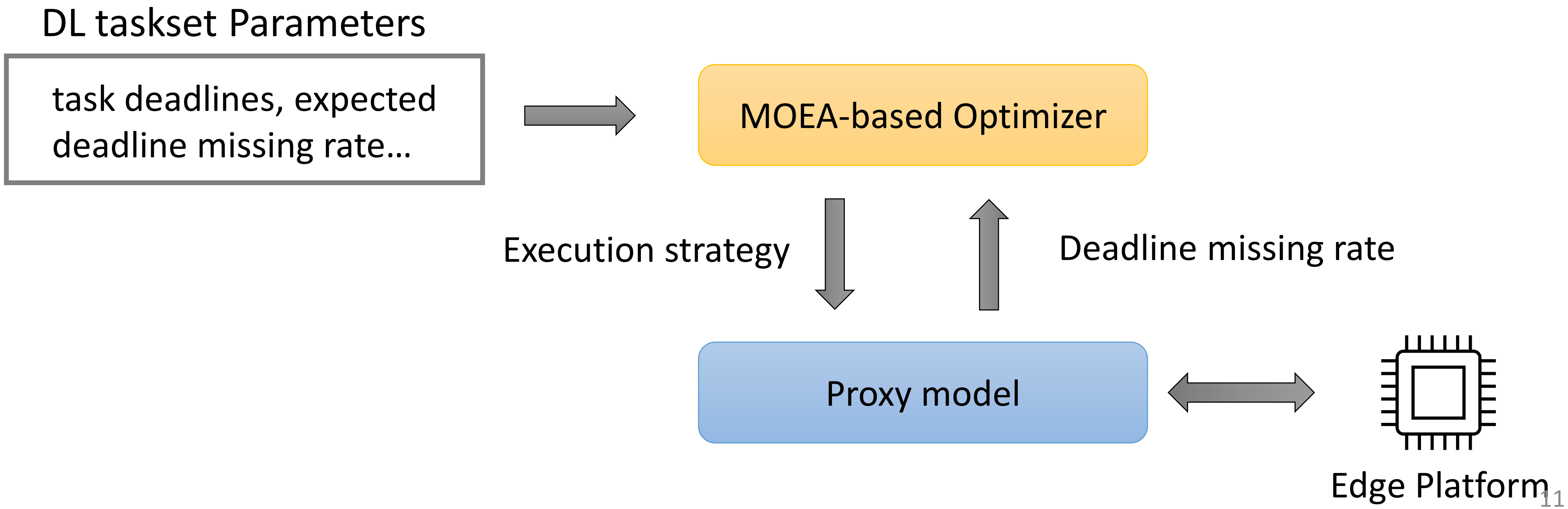
RT-mDL component 2: Execution strategy optimization

Task deadline missing rate

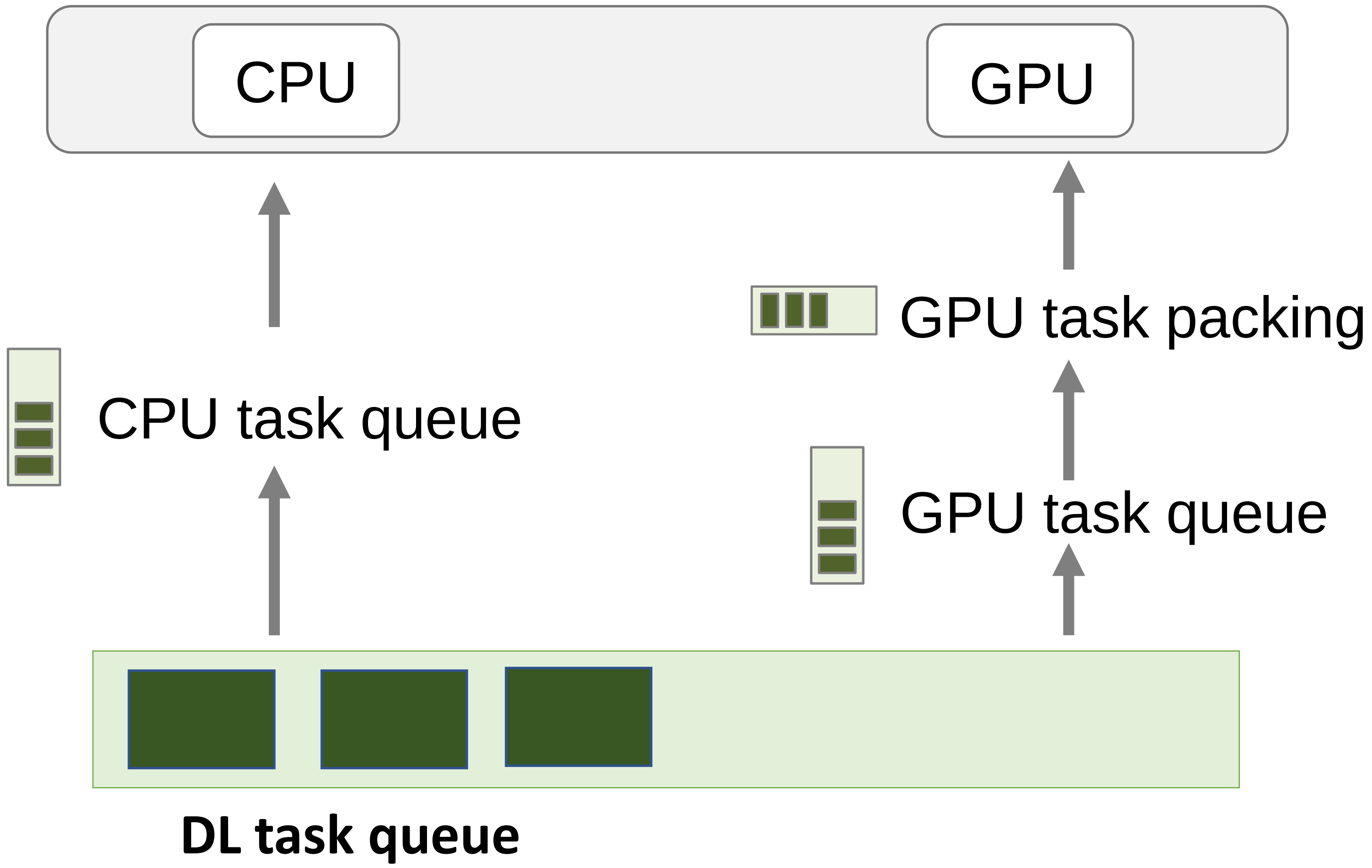
No closed form expressions

Pessimistic response time analysis

Our approach



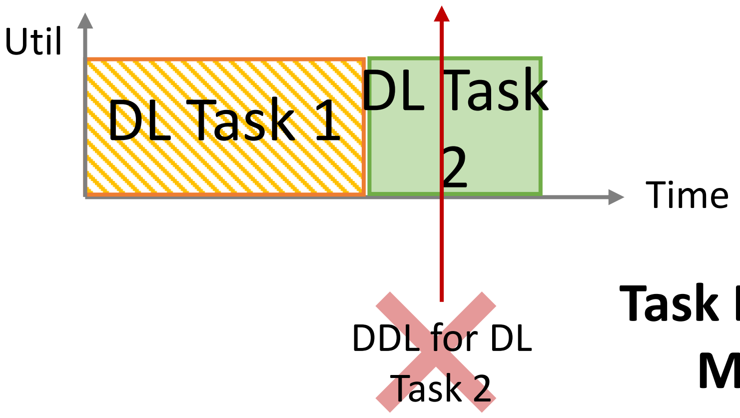
RT-mDL component 3: A new priority-based DL task scheduler



Improve CPU/GPU spatiotemporal utilization

RT-mDL component 3: A new priority-based DL task scheduler

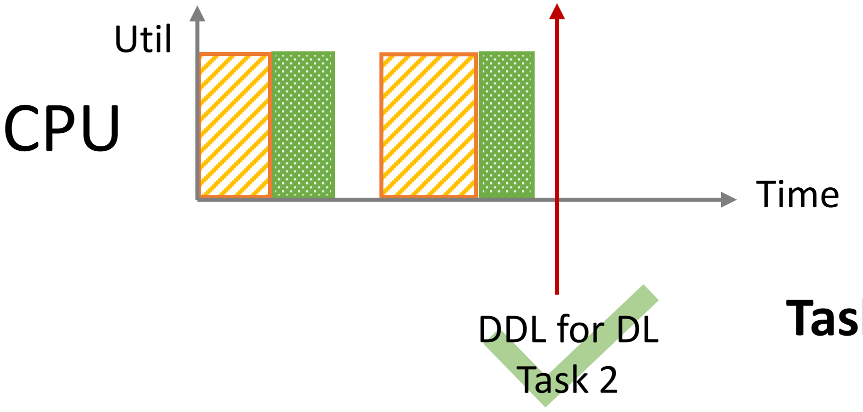
treat each DL task as a single task



Task Deadline Missed

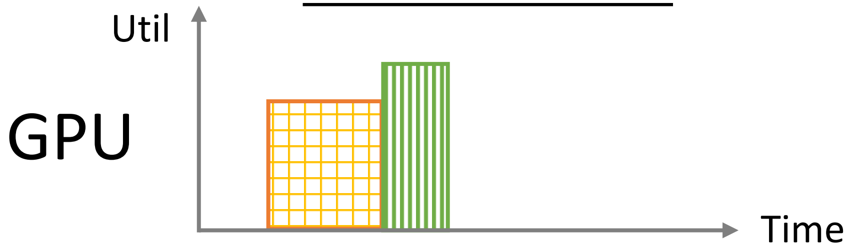
divide each DL task into CPU/GPU sub-task

<u>CPU sub-tasks</u>

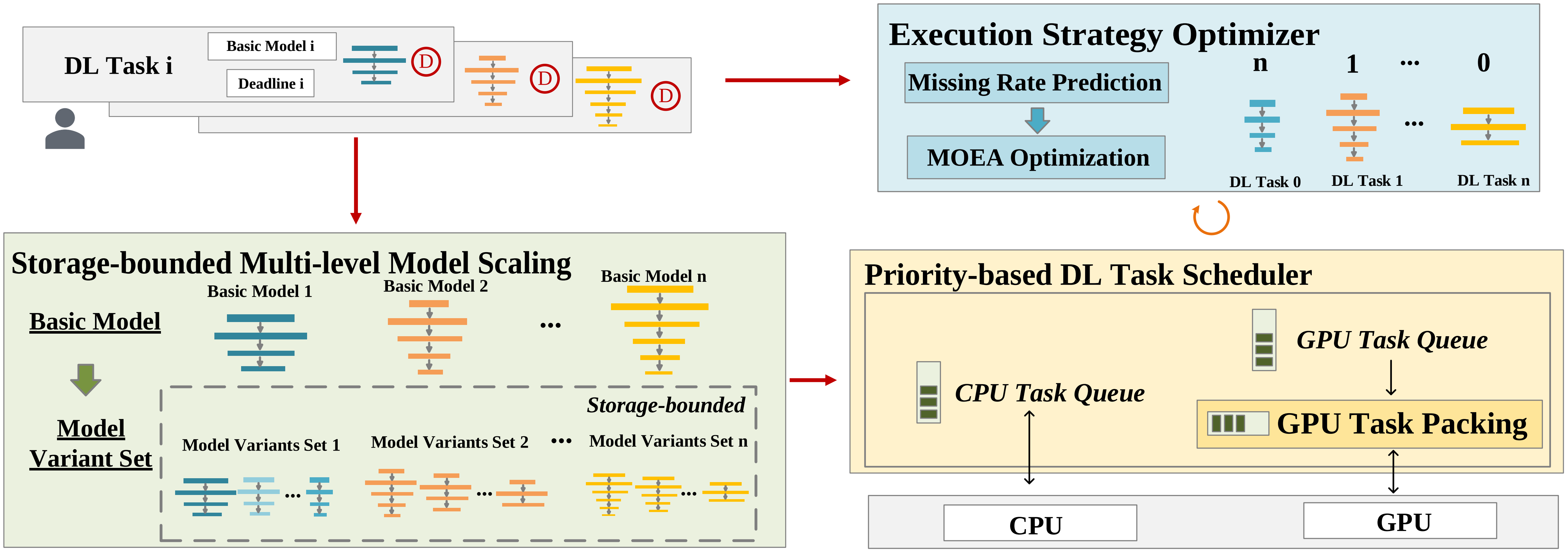


Task Deadline Met

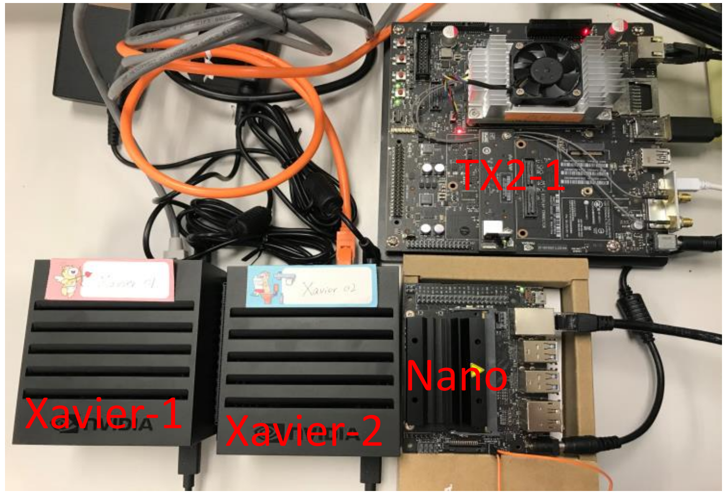
<u>GPU sub-tasks</u>



Recap of RT-mDL



RT-mDL implementation

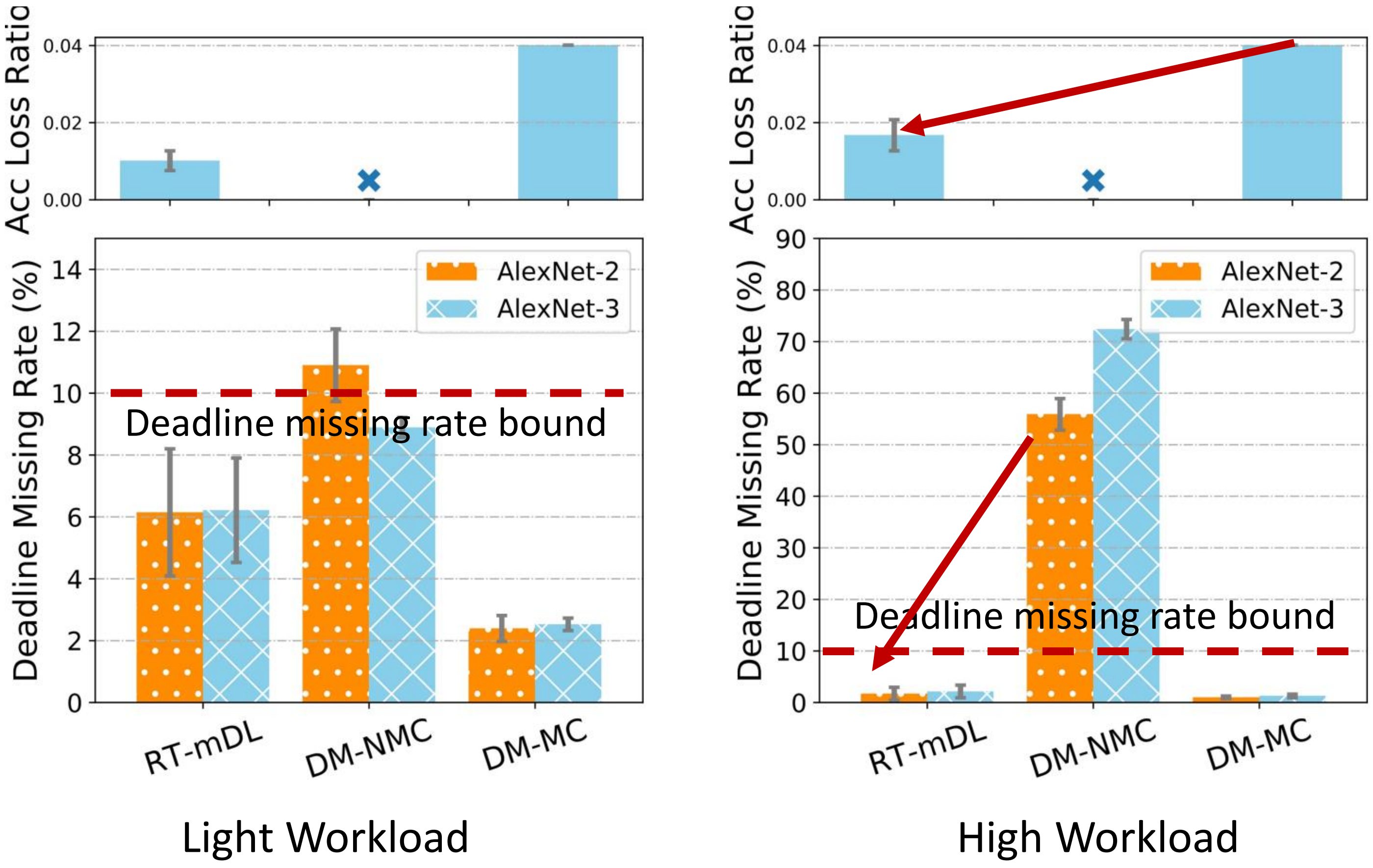


- **5 edge platforms**
- **5 types of DL task, 6 DNN models, 5 datasets**

Platform	GPU	CPU	Memory	Storage
NVIDIA AGX Xavier	512-core Volta	8-core ARMv8.2	16GB	32GB
NVIDIA Jetson TX2	256-core Pascal	2-core ARM Denver + 4-core ARM A57	8GB	32GB
NVIDIA Jetson Nano	128-core Maxwell	4-core ARM A57	4GB	microSD
Desktop	RTX2080	8-core Intel i9-9900K	32GB	5TB
Laptop	Intel Iris Plus	4-core Intel	16GB	500GB

DL Task Type	Dataset	DNN model	Pre/Post Processing
Image Classification	CIFAR10	AlexNet, VGG11/13/16/19, ResNet18/34	Data Fetch, Image Resizing
Sign Recognition	GTSRB	VGG11/19, ResNet18/34	
Object Detection	Self-collected traffic light dataset	tiny-YOLO	Image reading from camera, bounding box regression
Sound Classification	UrbanSound	SBCNN	Down-sampling, MFCC feature extraction
Emotion Recognition	Ravdess	LSTM	

Experimental results: Joint model scaling and scheduling



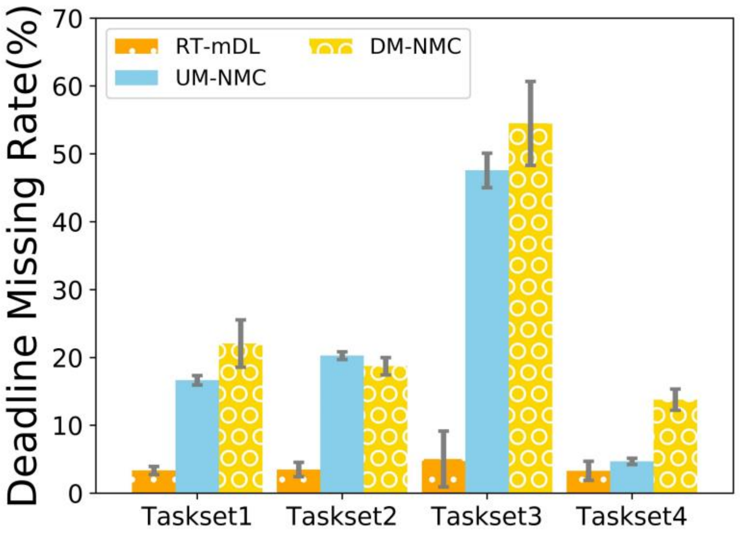
Two Baselines

- DM-NMC: Deadline Monotonic Scheduling + Original Model
- DM-MC: Deadline Monotonic Scheduling + Compressed Model

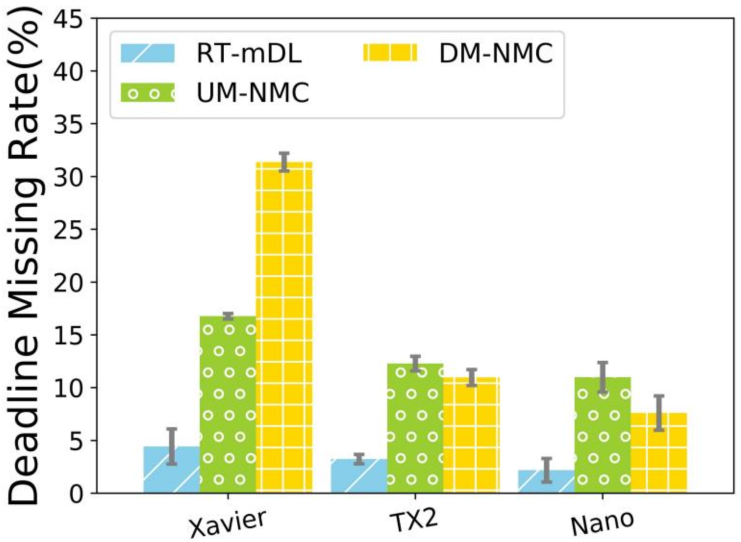
RT-mDL

- reduce deadline missing rate by **40.12%**
- while only sacrificing **1.7%** model accuracy loss.

Experimental results: Generality of RT-mDL



DNN Model Combination



Different edge platform

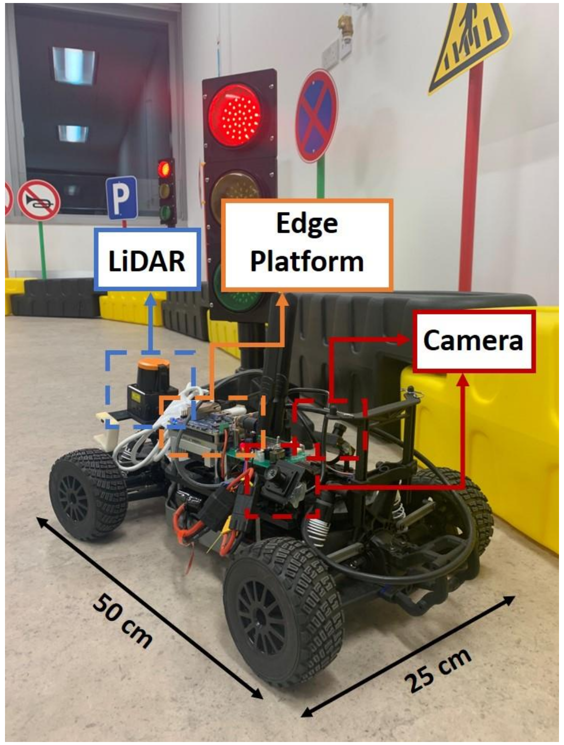
Two Baselines

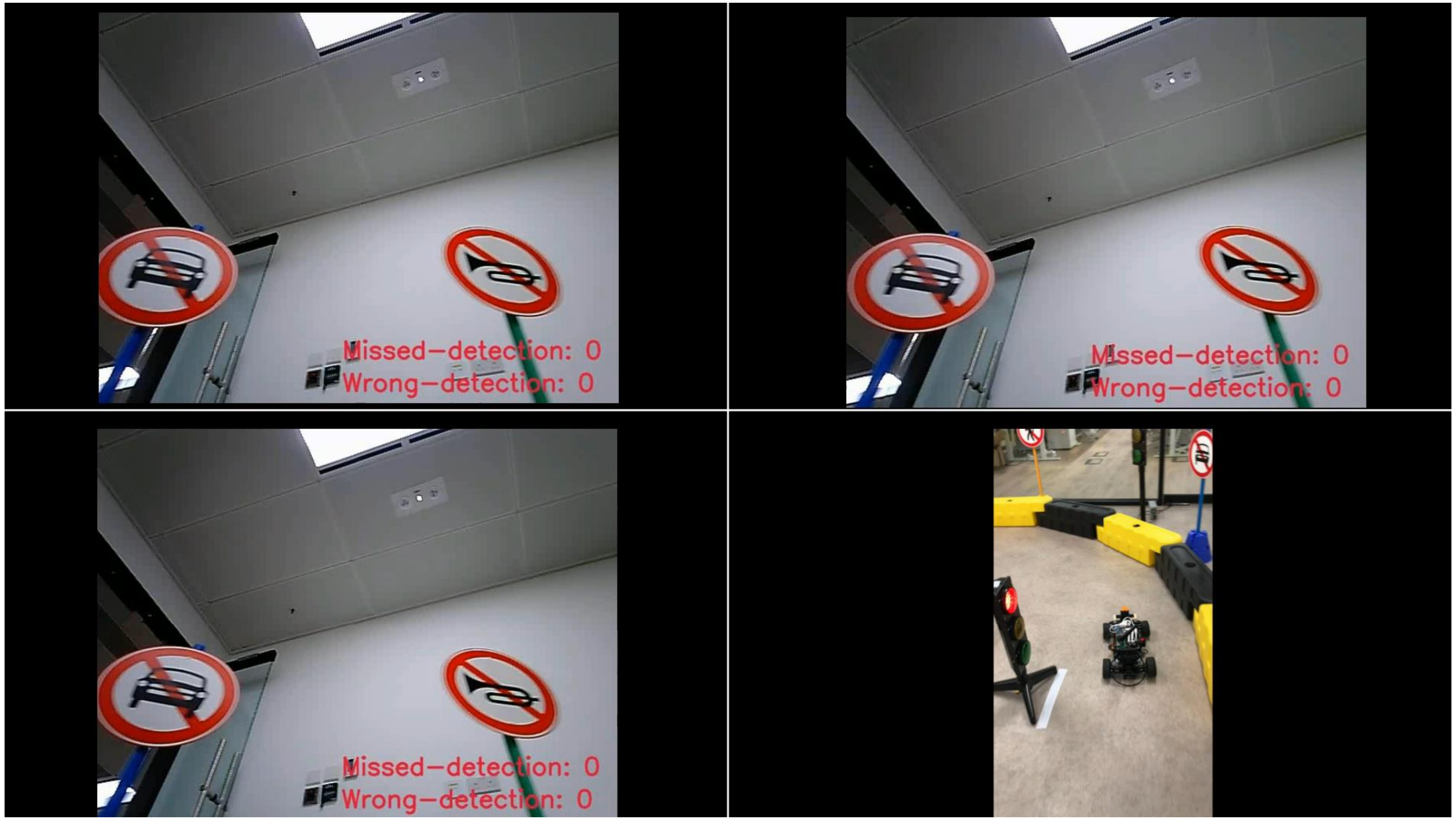
- UM-NMC: Utilization Monotonic Scheduling + Original Model
- DM-NMC: Deadline Monotonic Scheduling + Original Model

RT-mDL

- Superior performance across various DNN model combinations
- Adapt to different edge platforms

Experimental results: End-to-end system evaluation





GB RB GG RR BB GR RG

- RT-mDL
- Baseline1: Utilization Monotonic Scheduling + Original Model
- Baseline2: Deadline Monotonic Scheduling + Compressed Model

Conclusion

❑ RT-mDL

- ➢ The first framework to support mixed real-time DL task execution
  - Joint optimization of model scaling and tasks scheduling
  - A new priority-based real-time task scheduler for edge platform with CPU and GPU resource

- ➢ Implementation on an F1/10 autonomous driving testbed

❑ Future Work

- ➢ Handling uncertain workloads

- ➢ Integration with dynamic scheduling

Thanks for listening!

Any questions?

Visit CUHK AIoT Lab

RT-mDL: Supporting Real-Time Mixed Deep Learning Tasks on Edge Platforms

Neiwen Ling, Kai Wang, Yuze He, Guoliang Xing, Daqi Xie