AdaDrone: Quality of Navigation Based Neural Adaptive Scheduling for Edge-Assisted Drones

Haowei Chen, Liekang Zeng, Xiaoxi Zhang, Xu Chen

Sun Yat-sen University

ICDCS 2022

Background and Motivation

- Drones have been widely used in real world

Disaster rescue

Smart agriculture

Delivery services

The need for drone autonomy

+

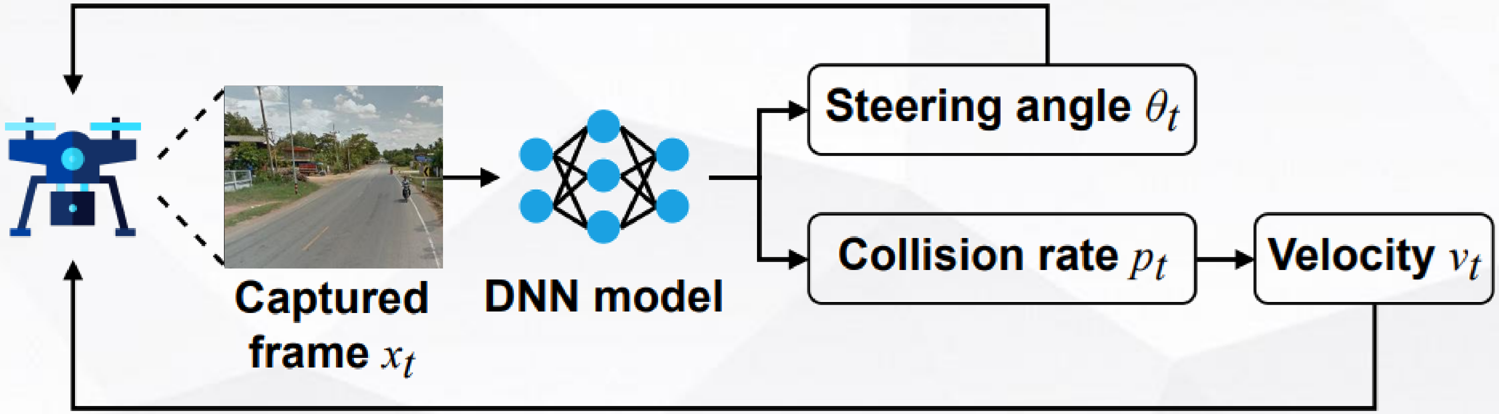
Development of deep learning

promote →

Self-navigation technology for drone

Self-navigation based on deep learning

- Capture images and call a DNN model to export navigation decisions
- DNN's outputs are sent to the drone for execution, forming a closed loop of control



A typical control loop for navigation [1]

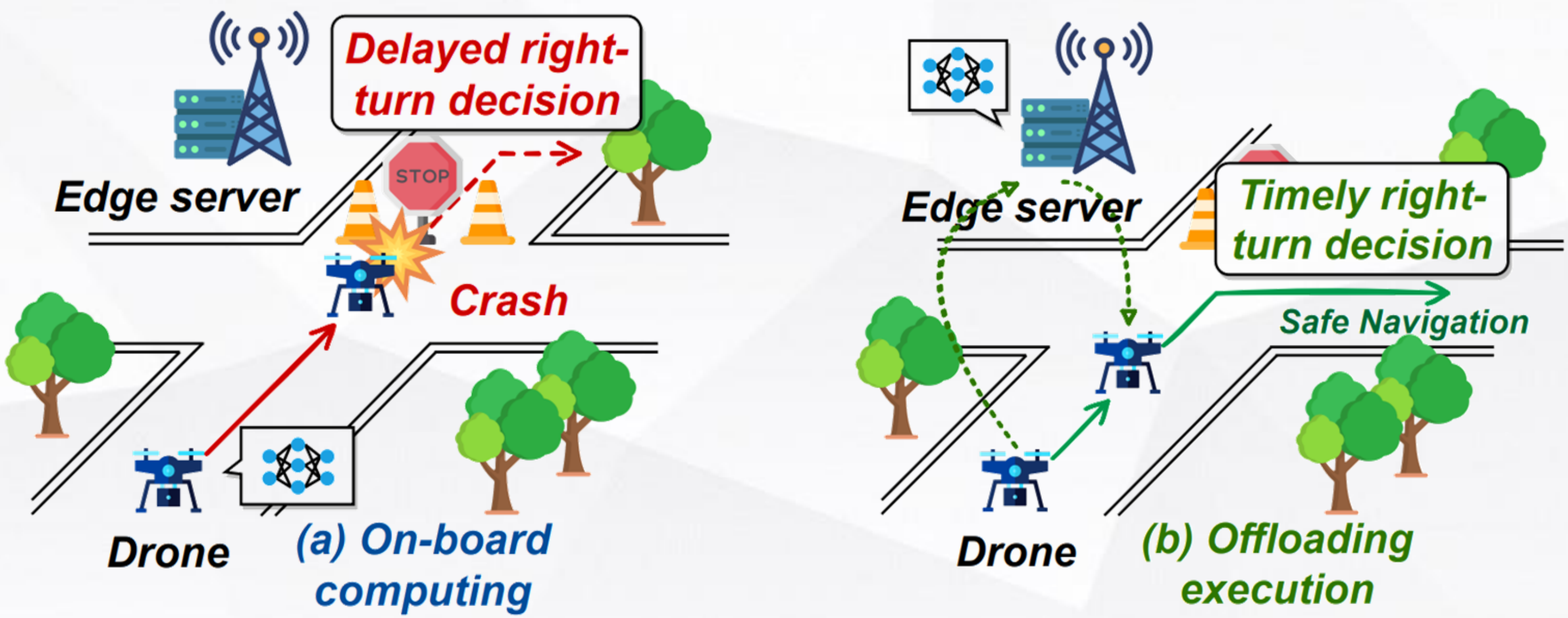
[1] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," IEEE RA-L, vol. 3, no. 2, pp. 1088–1095, 2018.

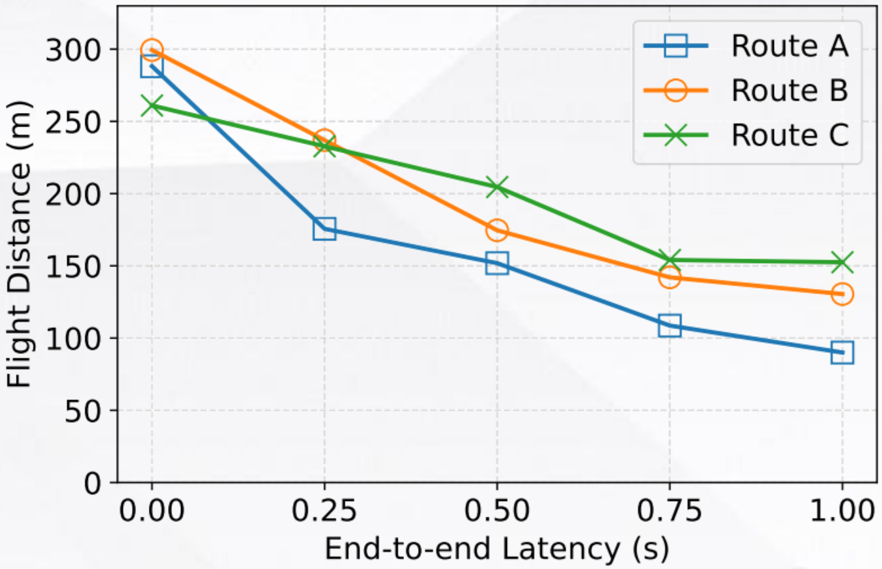
Challenge 1: Constrained computing capability of drones

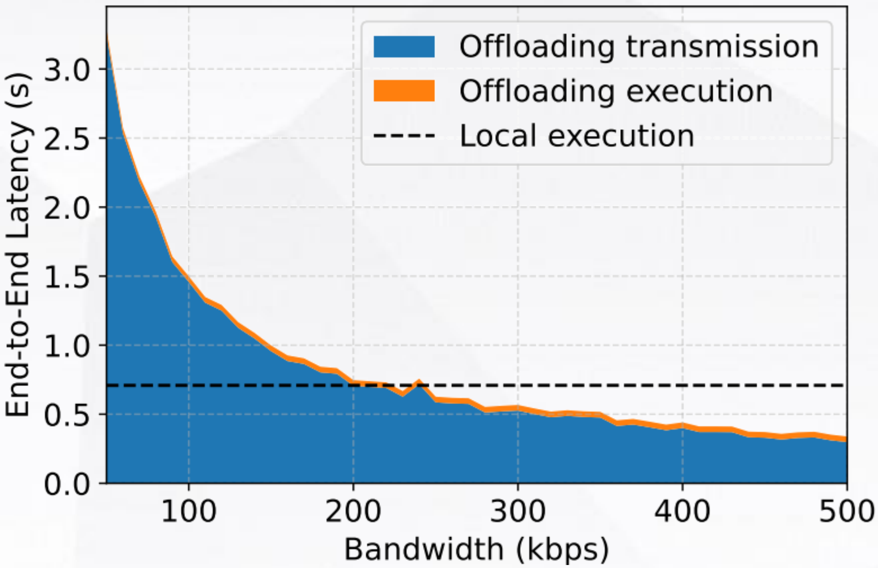
- Yielding high computing latency when performing DNN inference

Challenge 2: Drone's navigation is latency-sensitive

- Delayed navigation decision may lead the drone to an unexpected crash







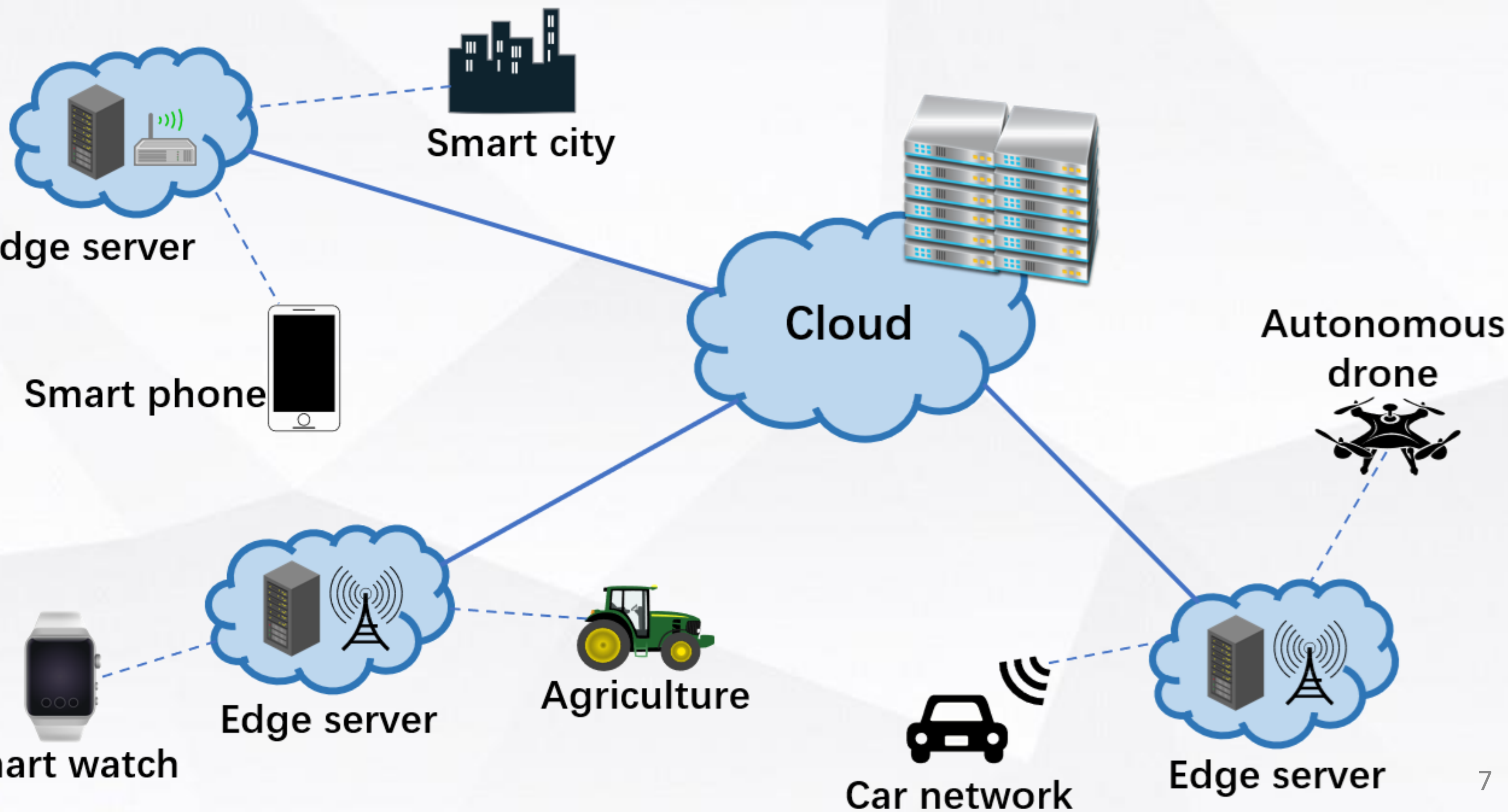
As the end-to-end latency of navigation decision increases, the achieved flight distance dramatically decreases

End-to-end latency of offloading and local execution, where the offloading latency breaks down in communication and computation

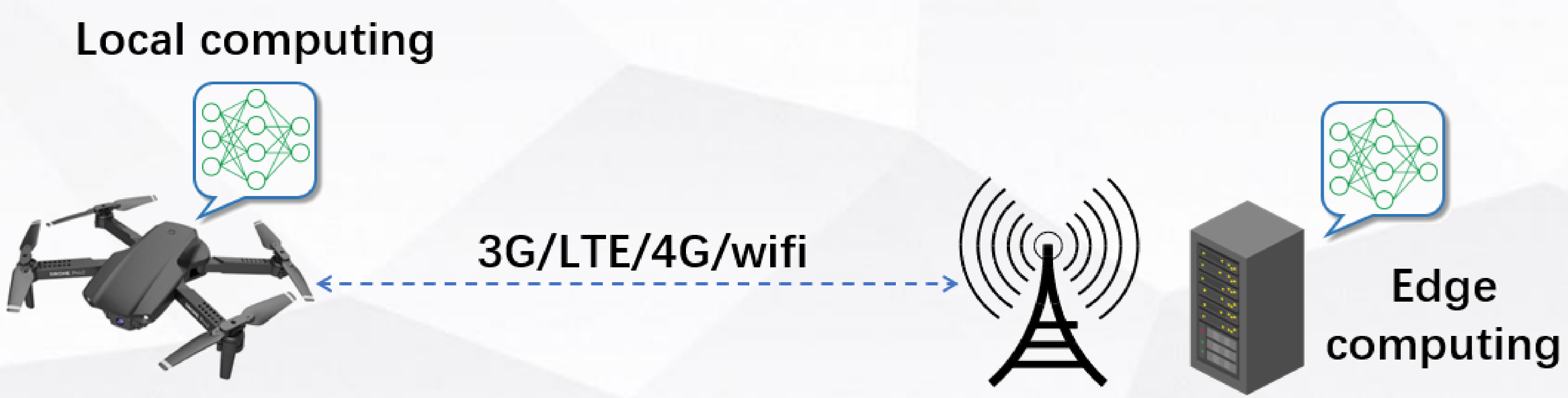
Constrained computing capability ⟹ **Conflict** ⟸ Latency-sensitive task

⇓ Promote

Edge computing

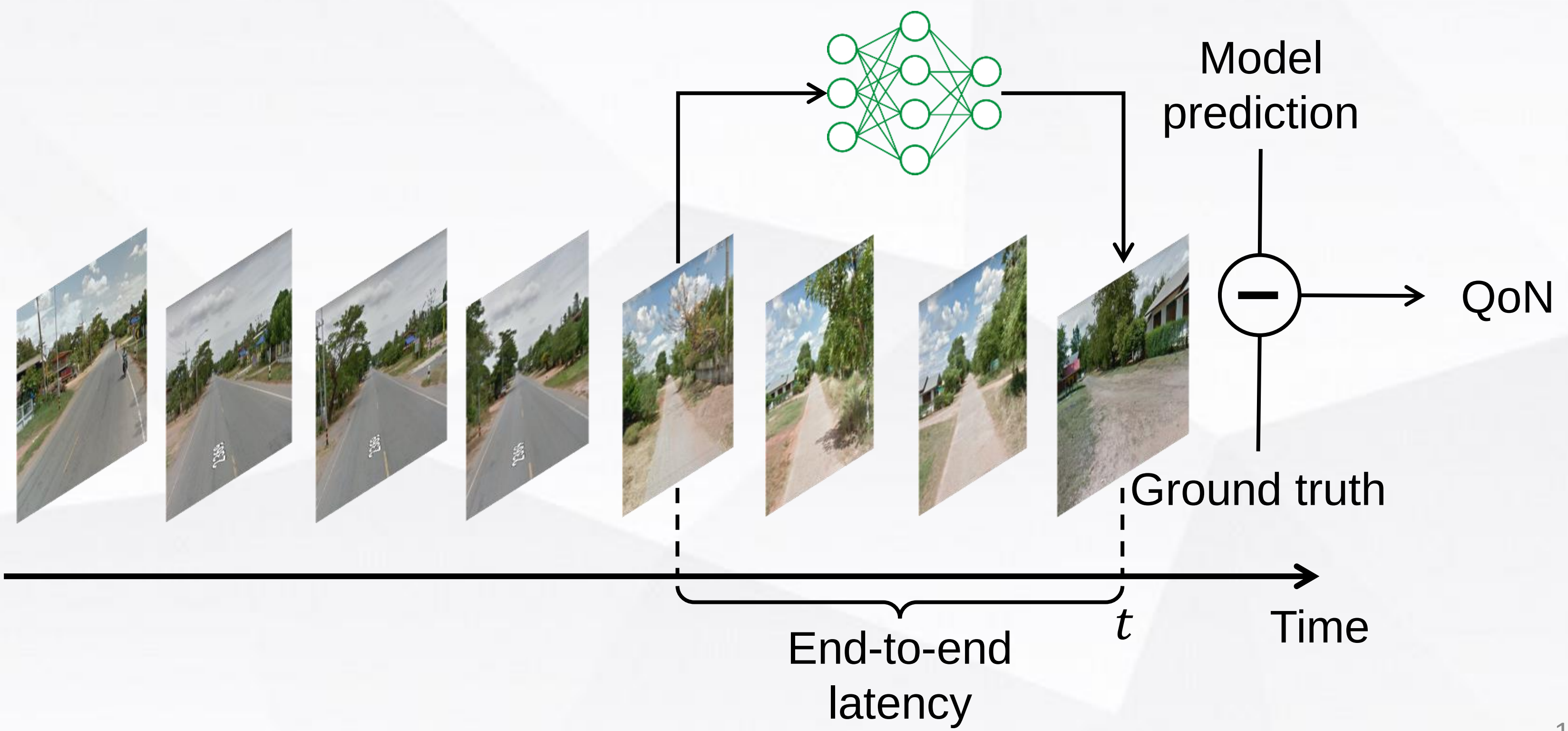


- Designing a framework for edge-assisted drone to optimize existing deep learning-based navigation techniques to improve the navigation performance



Problem Analyst and Method

- Treat adaptive navigation as a service
- Define *Quality of Navigation* (QoN), a new metric which can effectively shape the navigation performance in terms of **end-to-end latency** and **model accuracy**



Definition of Quality of Navigation(QoN)

- At any time t, the drone receives the model prediction on the turning angle θ_{pre}^{t}
- θ_{pre}^{t} is based on the scene at some previous moment t_{pre}
- θ_{gt}^{t}: ground truth of the current scene at time t
- τ: number of times that θ_{pre}^{t} are received in a time period

$$\left|\theta_{pre}^{t} - \theta_{gt}^{t}\right| \leq \varepsilon (1)$$

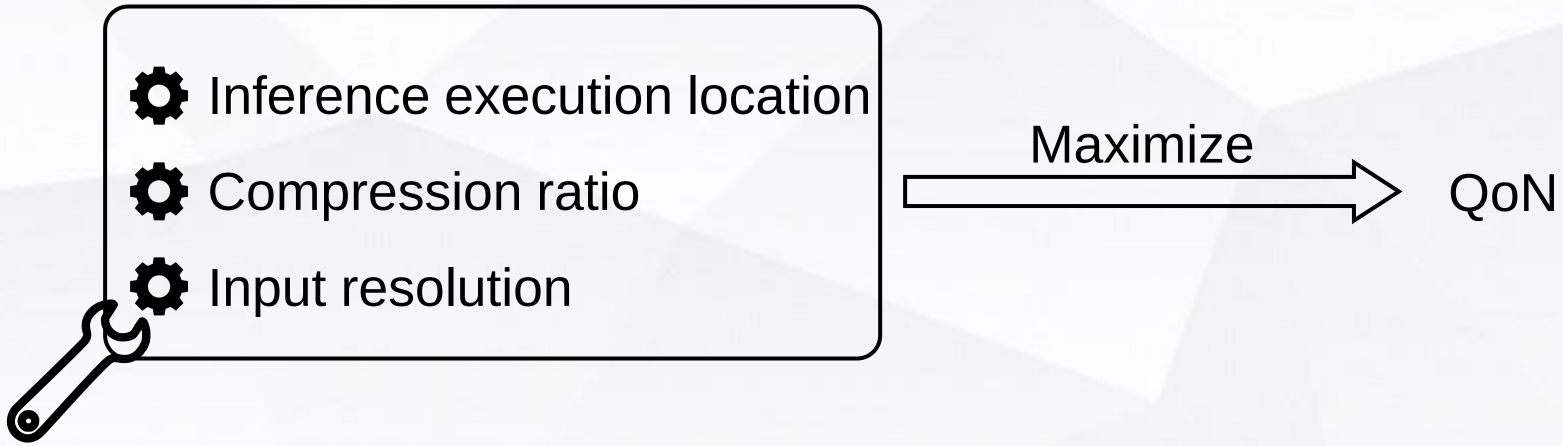
$$QoN = \sum_{t}^{\tau} I\left(\left|\theta_{pre}^{t} - \theta_{gt}^{t}\right| \leq \varepsilon\right)/\tau$$

where $I(\cdot)$ is an indicator function

2 PROBLEM ANALYST

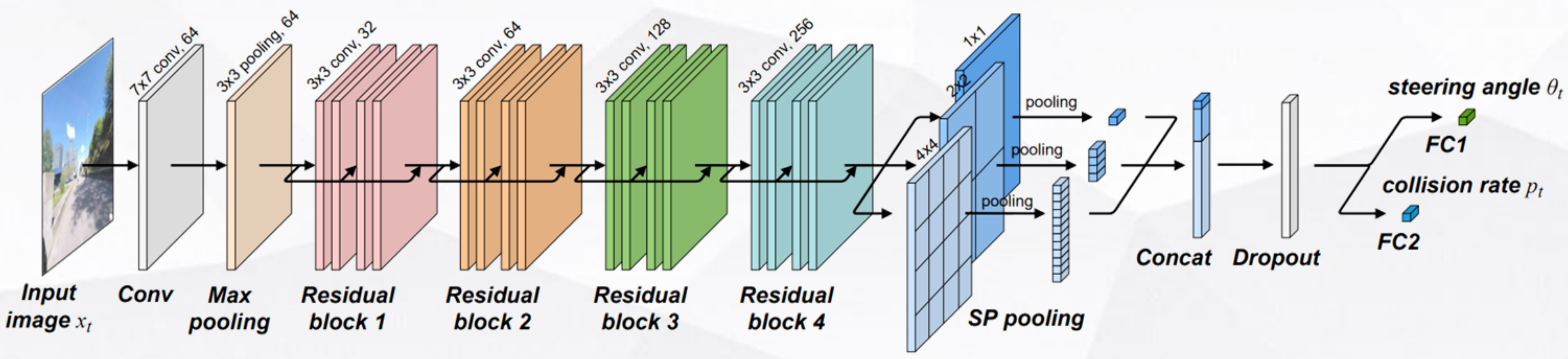
Goal: Maximize QoN

- Configuration 1: Inference execution location (Local or edge)
- Configuration 2: Compression ratio. Tune the trade off between data size and image's quality when offloading
- Configuration 3: Input resolution. Resize the input image to a lower resolution to reduce the computation workload.



Enable dynamic input resolution

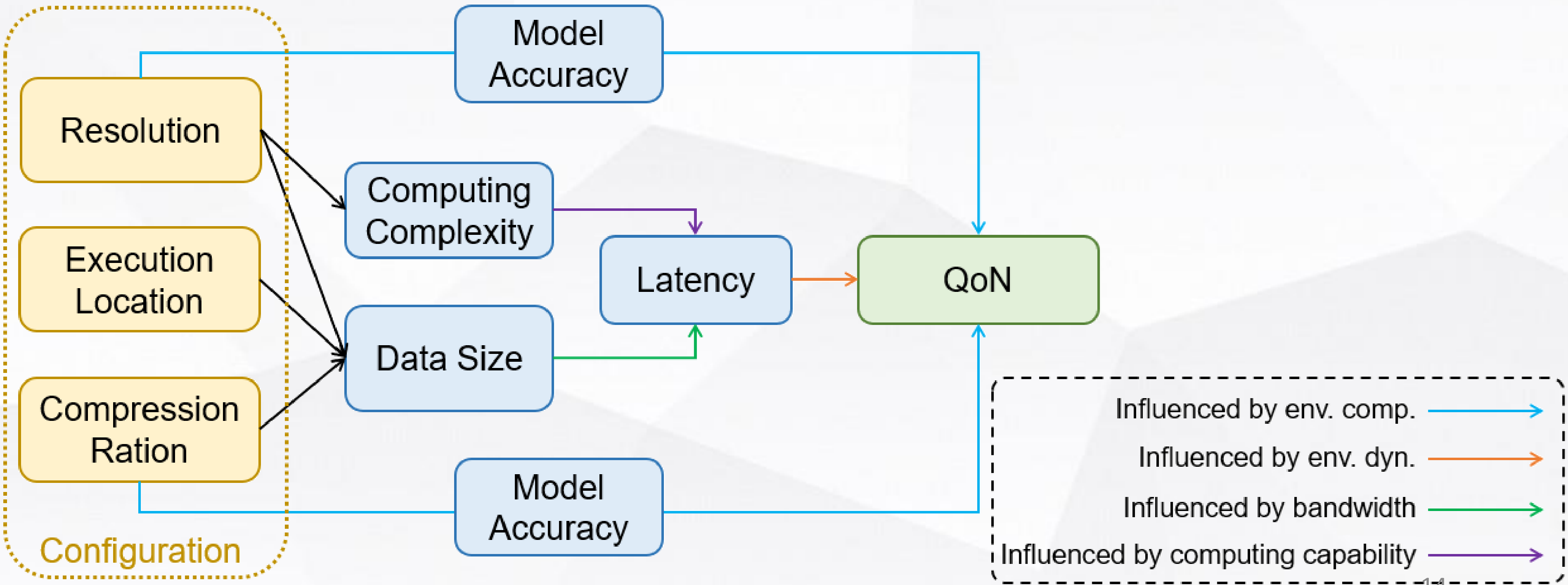
- We leverage the Spatial Pyramid Pooling mechanism
- It enables the model to input arbitrary resolution, and the computational complexity of the model is proportional to the input resolution



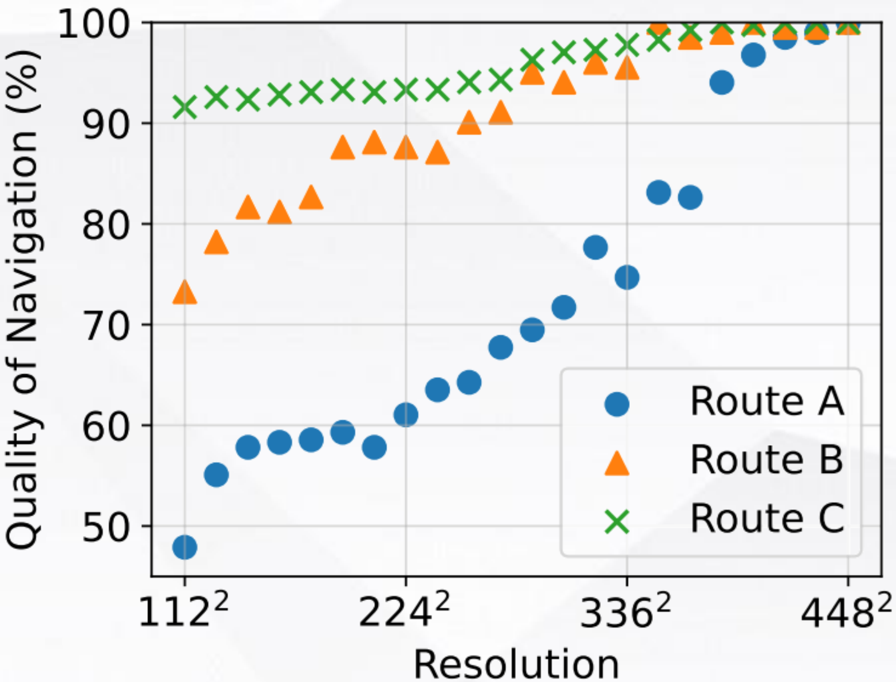
Architecture of navigation model with spatial pyramid pooling

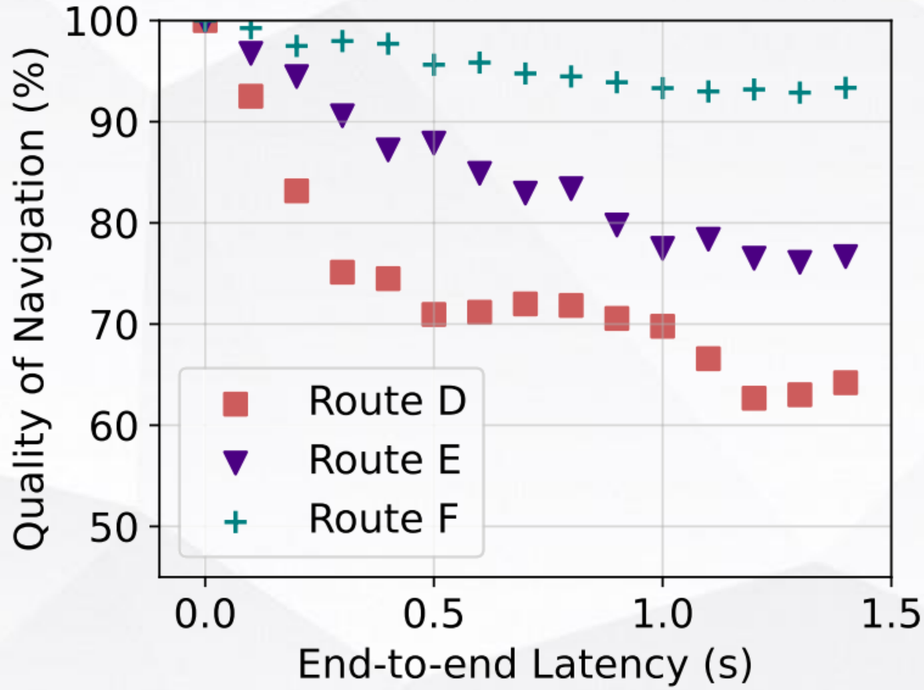
Challenges of Scheduling Adaptive Navigation

1. Composite optimization objective
2. Complex nexus of schedulable configurations
3. Dynamic environmental information



Dynamic environmental information

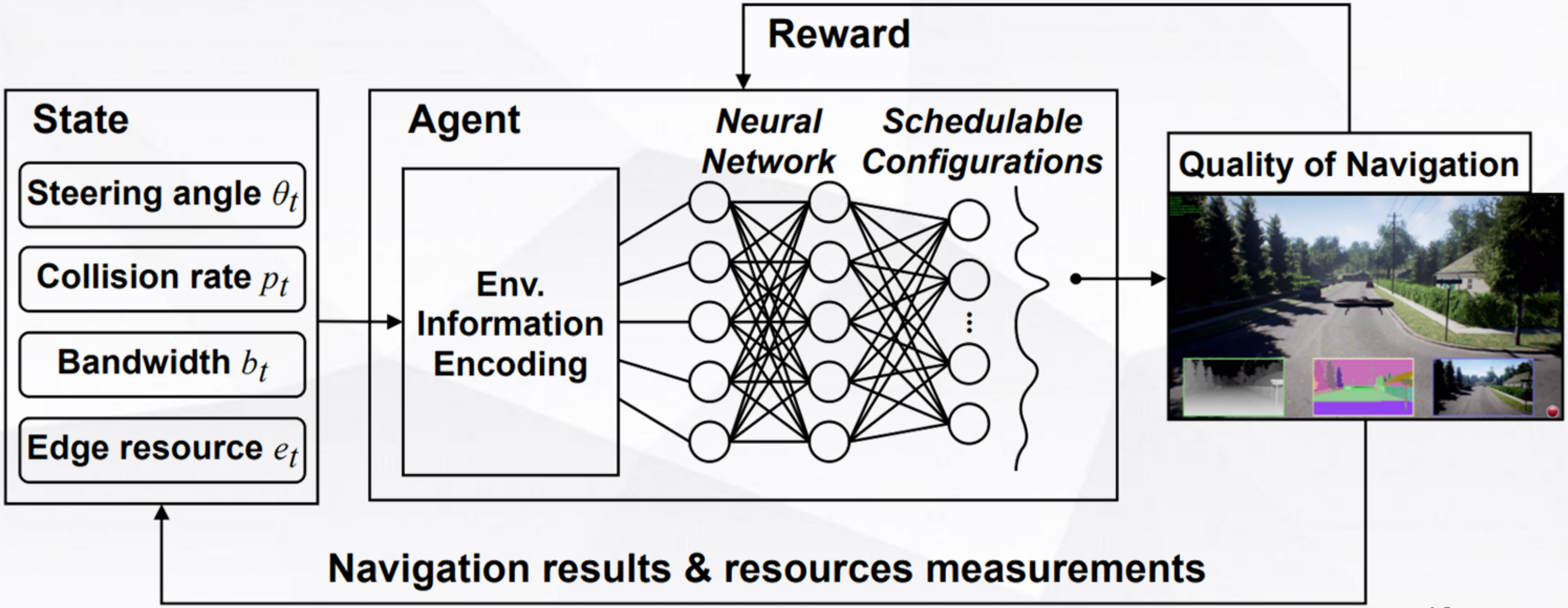


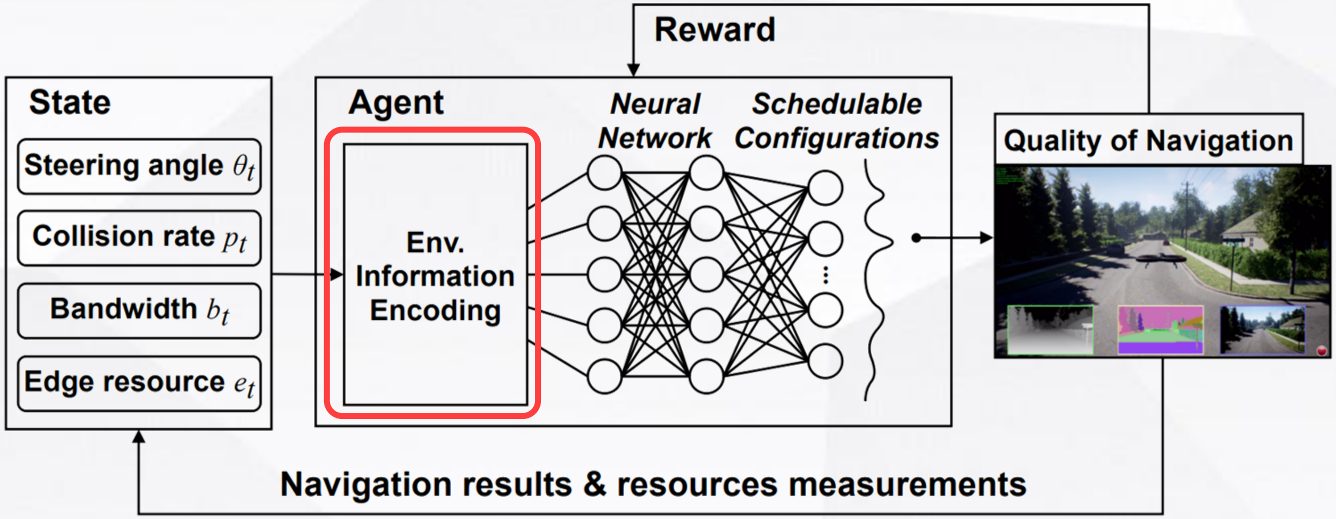


The measured Quality of Navigation varies in different routes with respect to the changes of resolution (left) and end-to-end latency (right).

DRL Neural Scheduler

- State: Outputs of navigation model, bandwidth, edge computing resource
- Action: Schedulable configurations
- Reward: Quality of Navigation





- Scheduler needs to deal with dynamic environmental information
- Original state cannot directly characterize this information

Environmental Information Encoding

1. *Environment complexity*: Reflect how sensitive the QoN is to the change of input resolutions
2. *Environment dynamics*: Characterize how rapidly the content of captured images changes

Environment complexity $\boldsymbol{c}$

- θ_h, p_h: Model output corresponding to images in the highest resolution
- θ_l, p_l: Model output corresponding to images in the lowest resolution

$$c = |\theta_h - \theta_l| + \alpha|p_h - p_l|$$

α is a hyper-parameter that keeps θ and p at the same order of magnitude

Environment dynamics $\boldsymbol{d}$

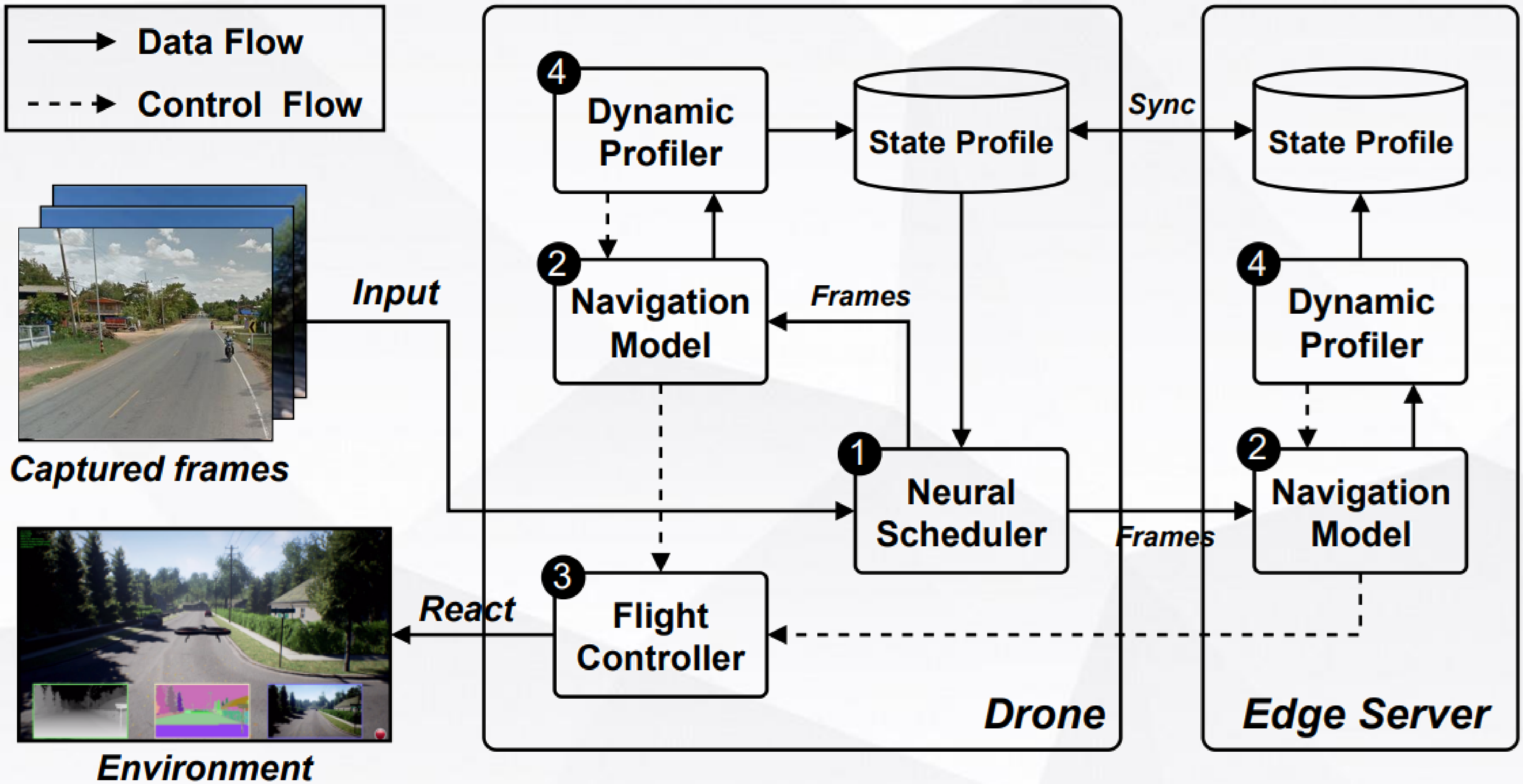
- $\boldsymbol{\theta}, \boldsymbol{p}$: Model outputs within the latest navigation epoch

$$d = \sigma(\boldsymbol{\theta}) + \beta\sigma(\boldsymbol{p})$$

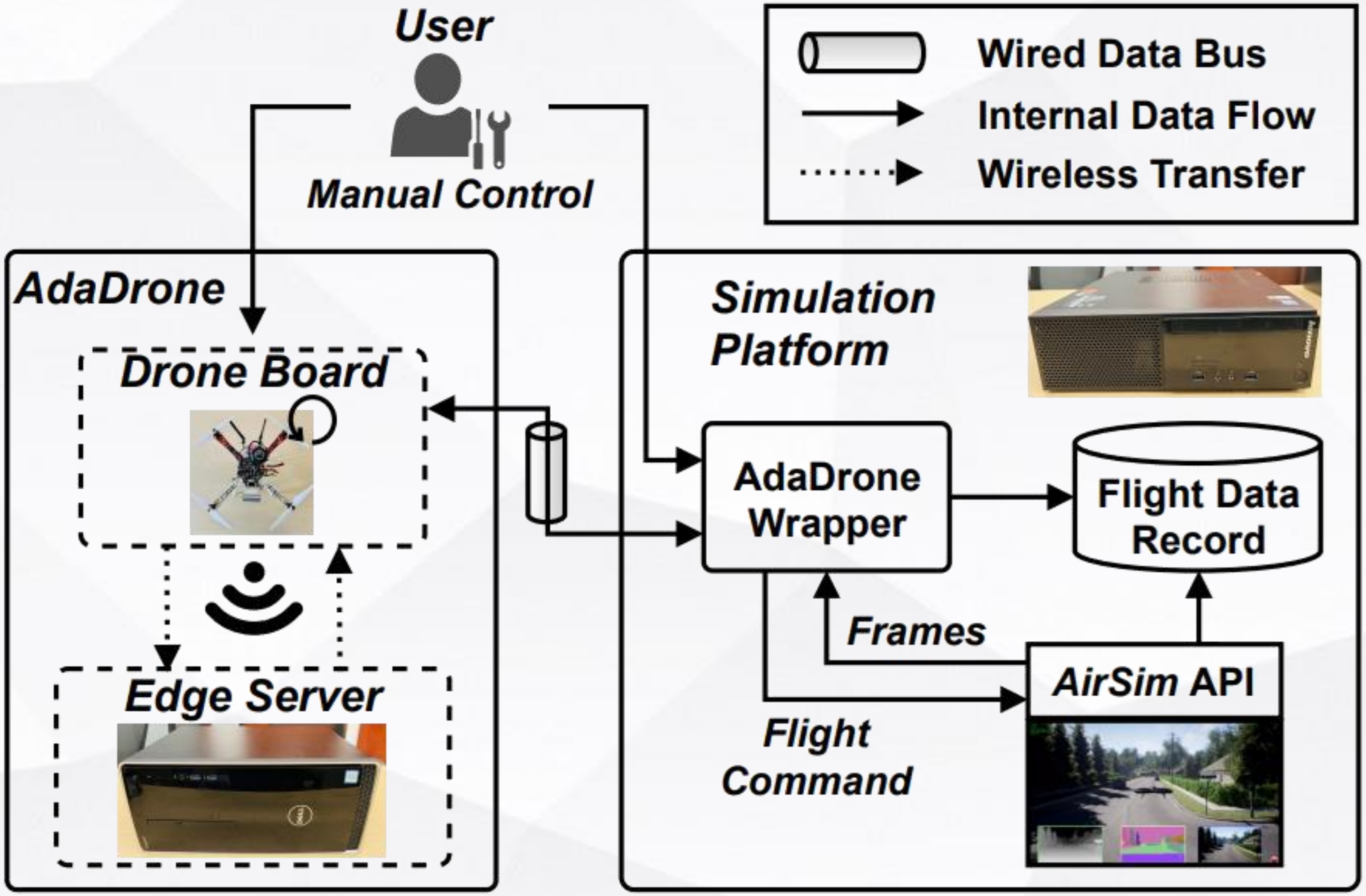
$\sigma(\cdot)$ reckons the standard deviation

Framework Design and Evaluation

AdaDrone framework overview



- Validate the performance of AdaDrone in AirSim drone simulator

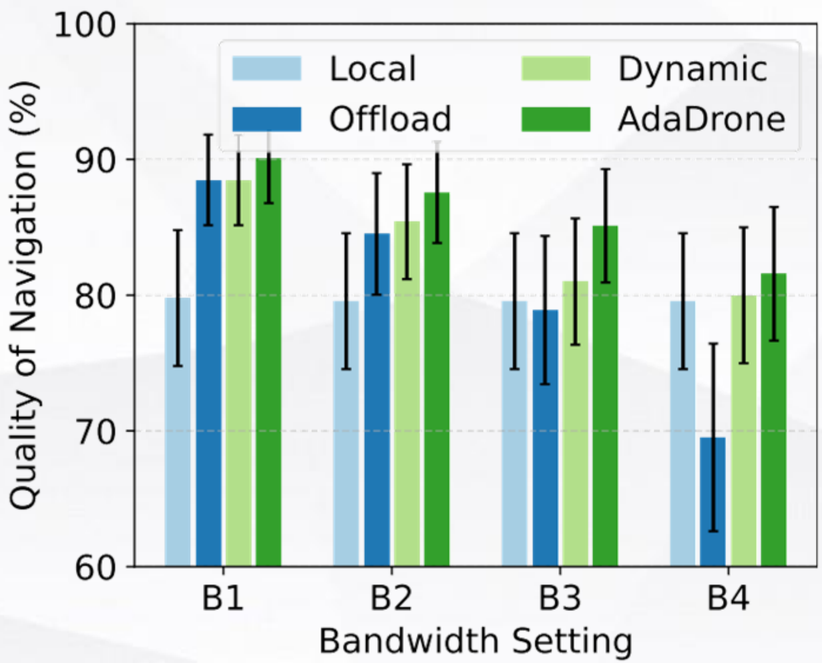


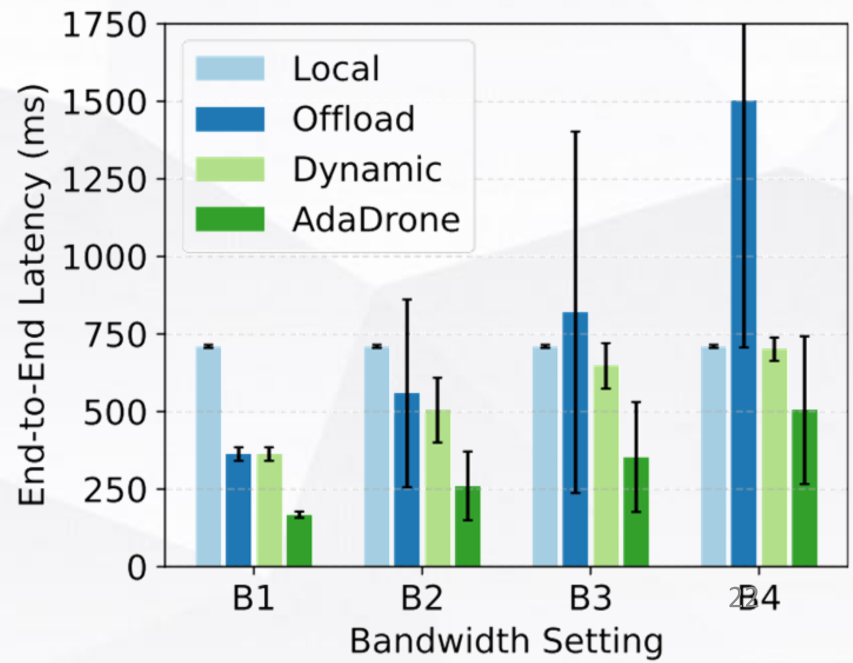
AdaDrone integration with AirSim

3 Evaluation

Baselines

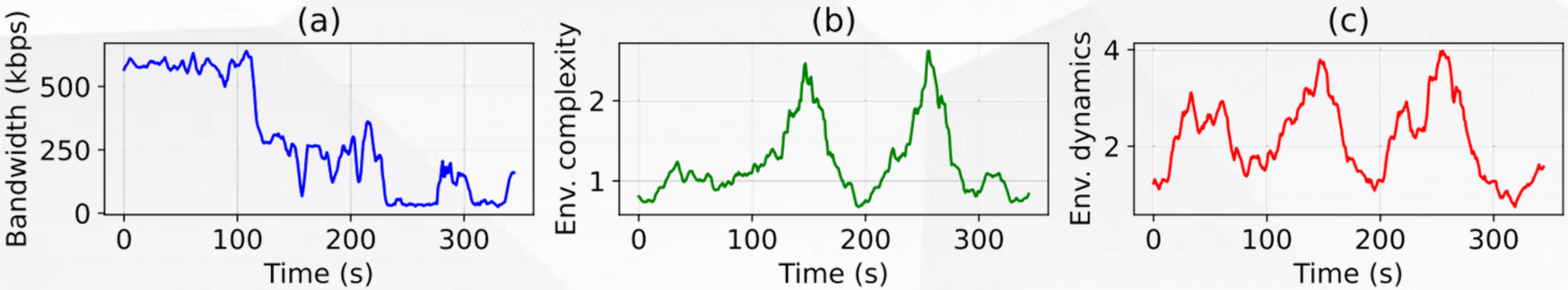
- *Local*: Place the navigation model on the onboard computing device for execution at any moment
- *Offload*: Place the navigation model on the server for execution at any moment
- *Dynamic Offload*: Merely optimizes QoN by adapting the inference execution location



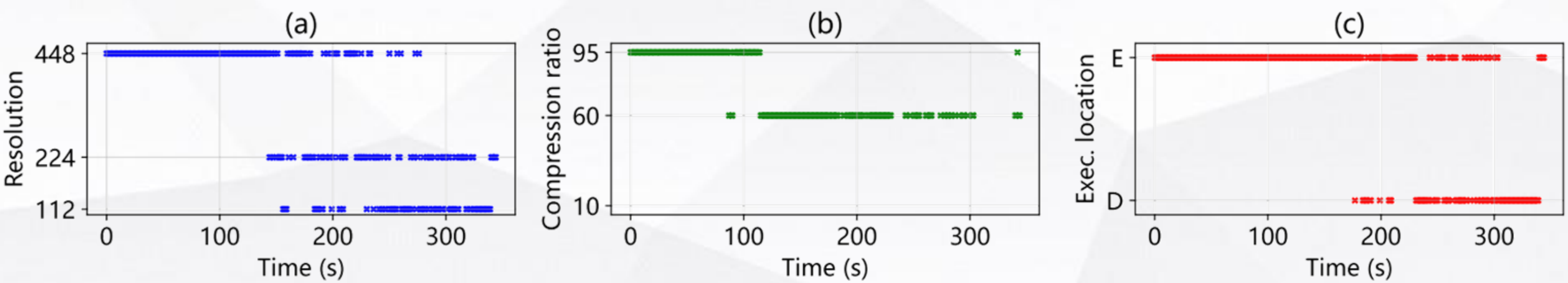


Case study

- We investigate how AdaDrone makes dynamic decisions

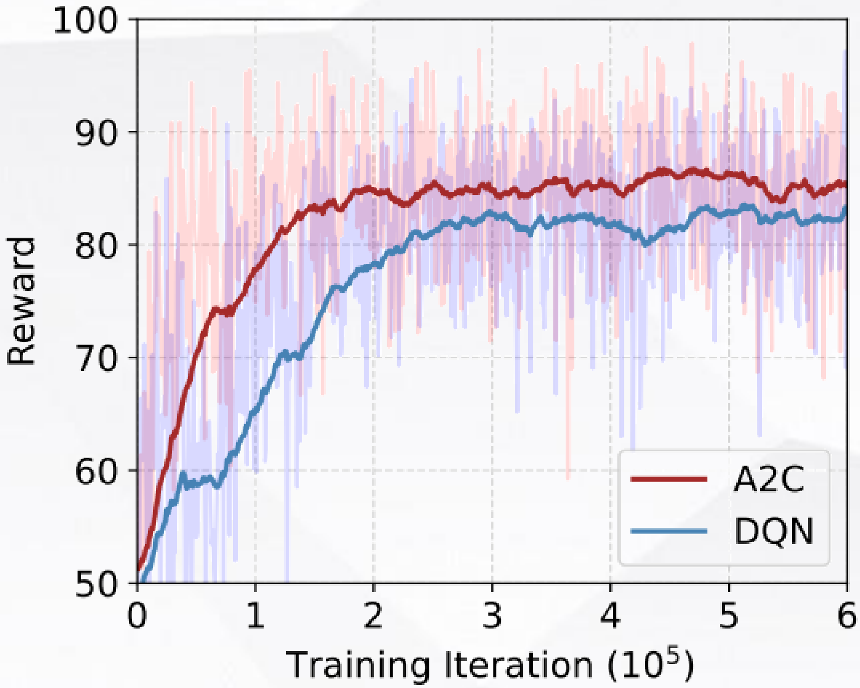


Trajectories of bandwidth, environment complexity and dynamics

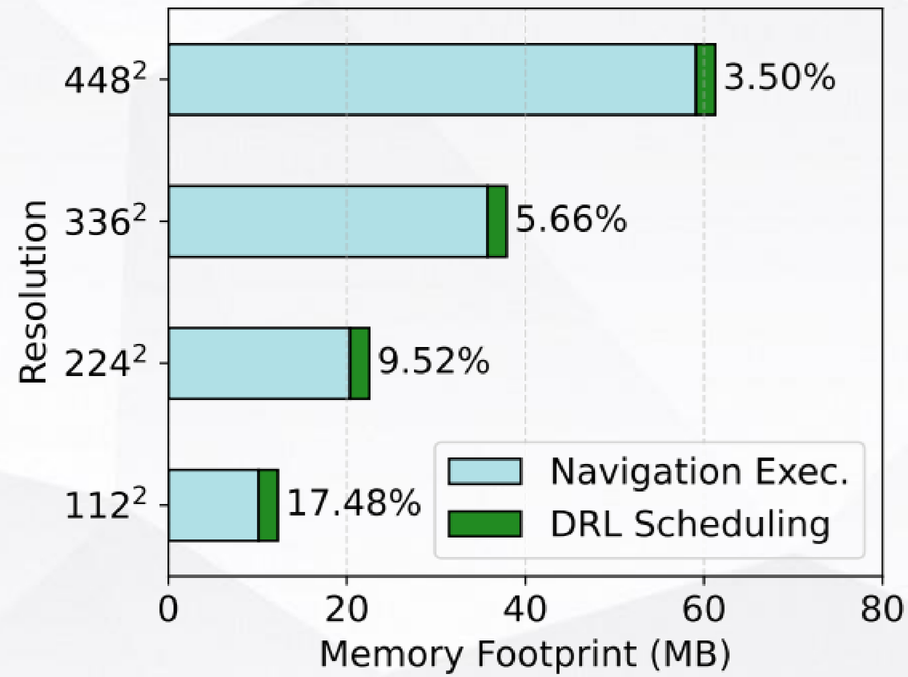


Selection of three decision variables of AdaDrone

Performance of DRL Neural Scheduler



The training curves of the neural scheduler with different DRL algorithms



The memory footprint of the neural scheduler and the navigation model

THANKS

THANK YOU