Dynamic Distributed Computing for Autonomous Vehicles in 5G Infrastructures

Marco Levorato
Professor
CS - University of California, Irvine
levorato@uci.edu

Earth Day Workshop 2024

Marco Levorato
Professor
University of California, Irvine
Computer Science Dept.

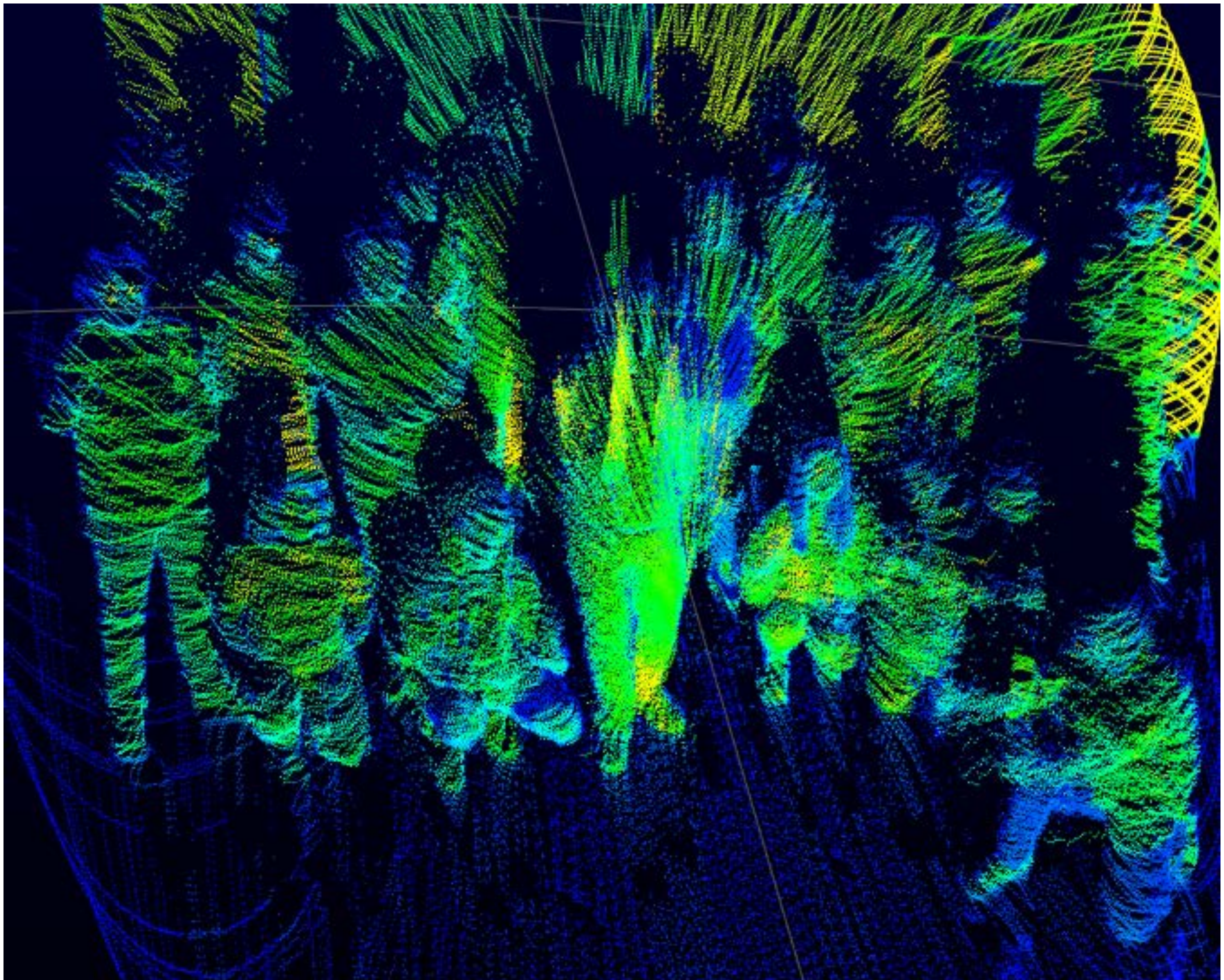
Faculty - Computer Science
University of California, Irvine

PostDoc - Electrical Engineering
Stanford University
University of Southern California

Ph.D - Telecommunication Engineering
University' degli Studi di Padova (Italy)

Marco Levorato
Professor
University of California, Irvine
Computer Science Dept.

Intelligent and
Autonomous Systems Lab

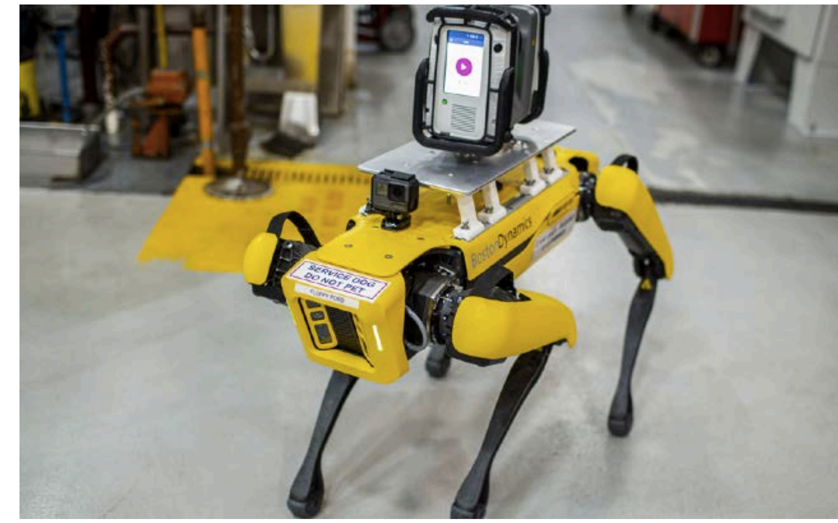


Dynamic, Distributed and Resilient Computing for Extreme Mobile Applications

Autonomous cars

Autonomous drones

Collaborative robots



Smart manufacturing

Augmented Reality

Mobile HealthCare

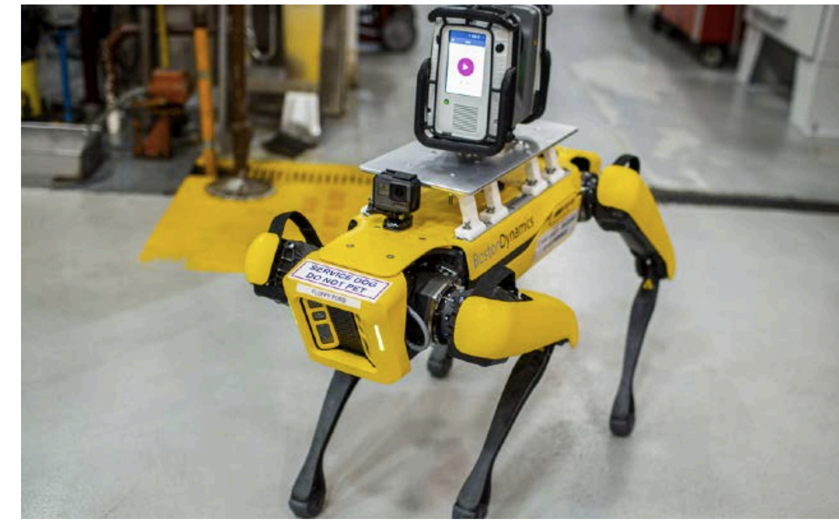
Funded by

Dynamic, Distributed and Resilient Computing for Extreme Mobile Applications

Autonomous cars

Autonomous drones

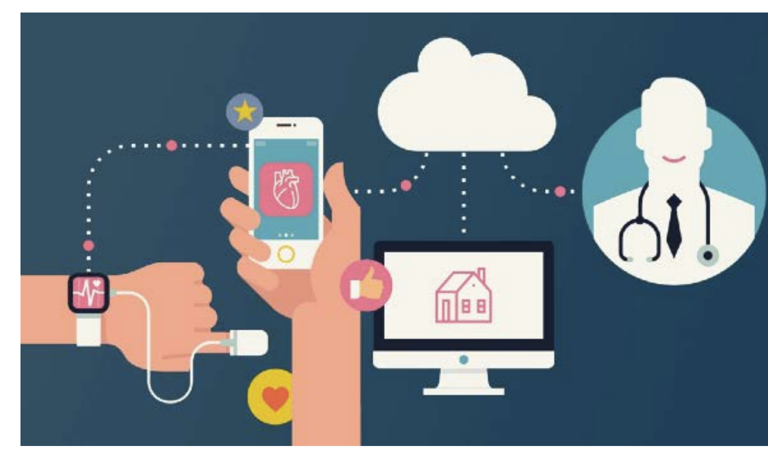
Collaborative robots



Smart manufacturing

Augmented Reality

Mobile HealthCare



Challenge

Mismatch between **computing needs** (e.g., continuous stream of complex computing tasks), **requirements** (accuracy, latency) and **resource** availability (energy, computing power)

Multi-Faceted Challenge

Algorithm level

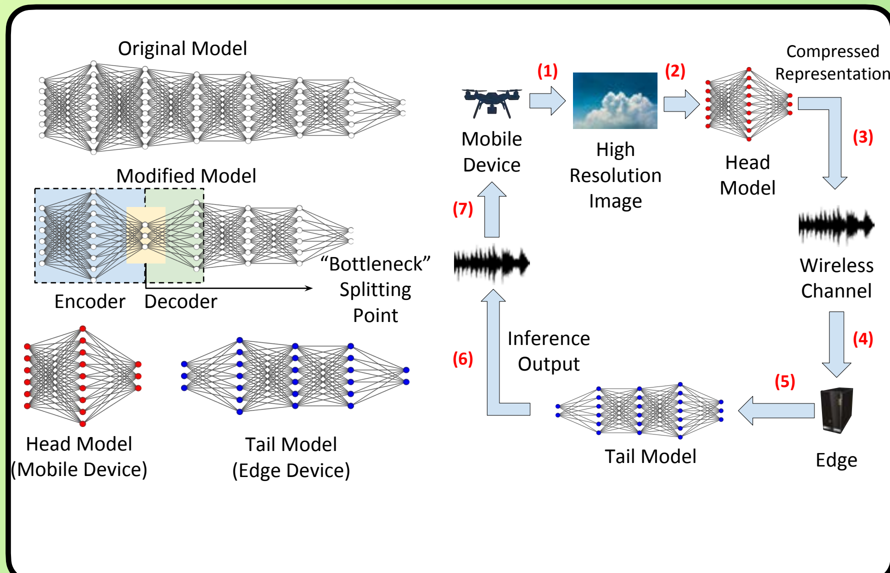
Split, dynamic, multi-branched models designed for efficiency and and distributed execution

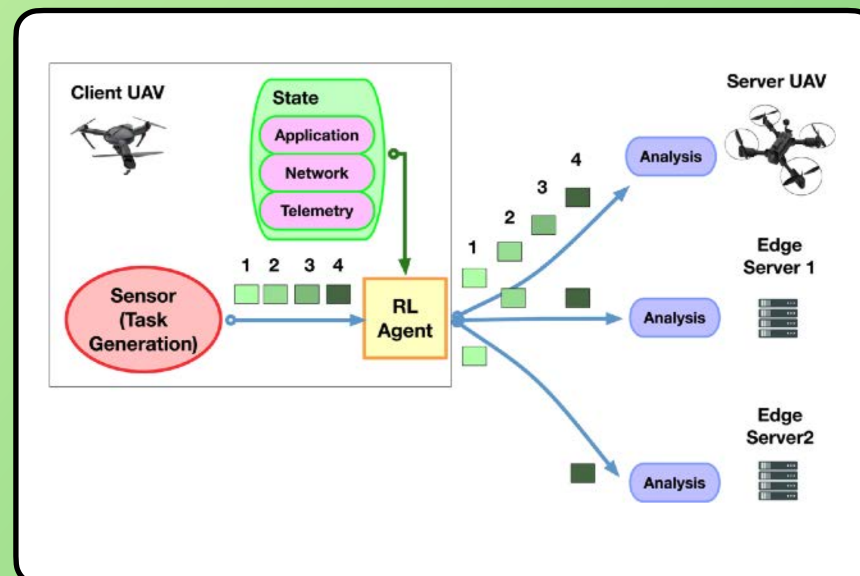
Device-level

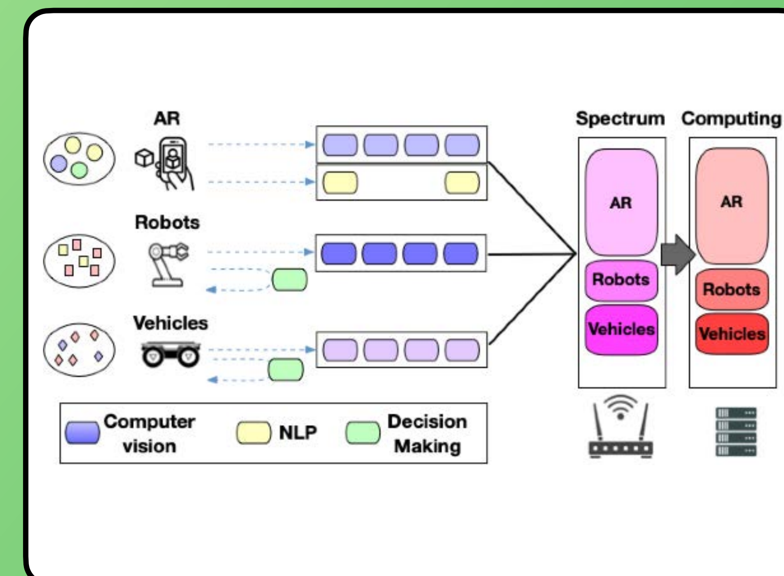
Context-aware computing strategies and semantic communication strategies

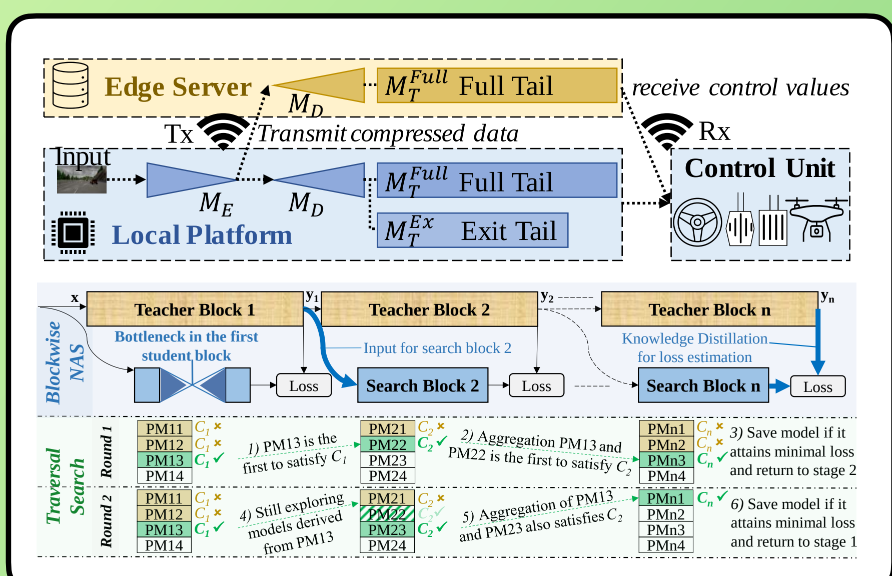
Infrastructure-level

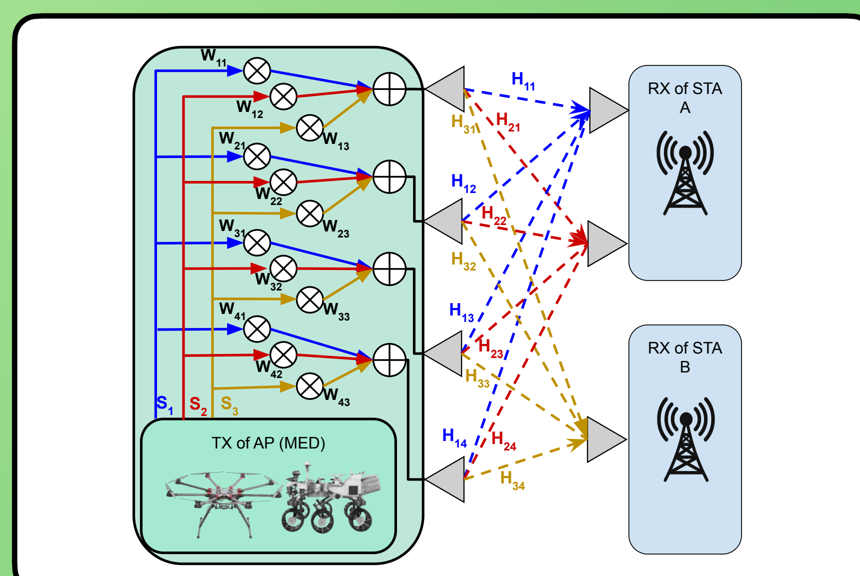
AI for semantic network and edge slicing, generalization and knowledge accumulation

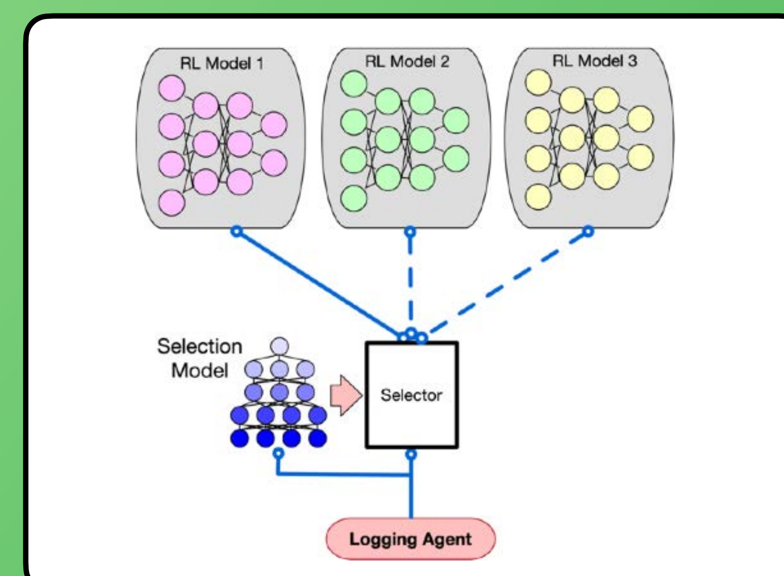






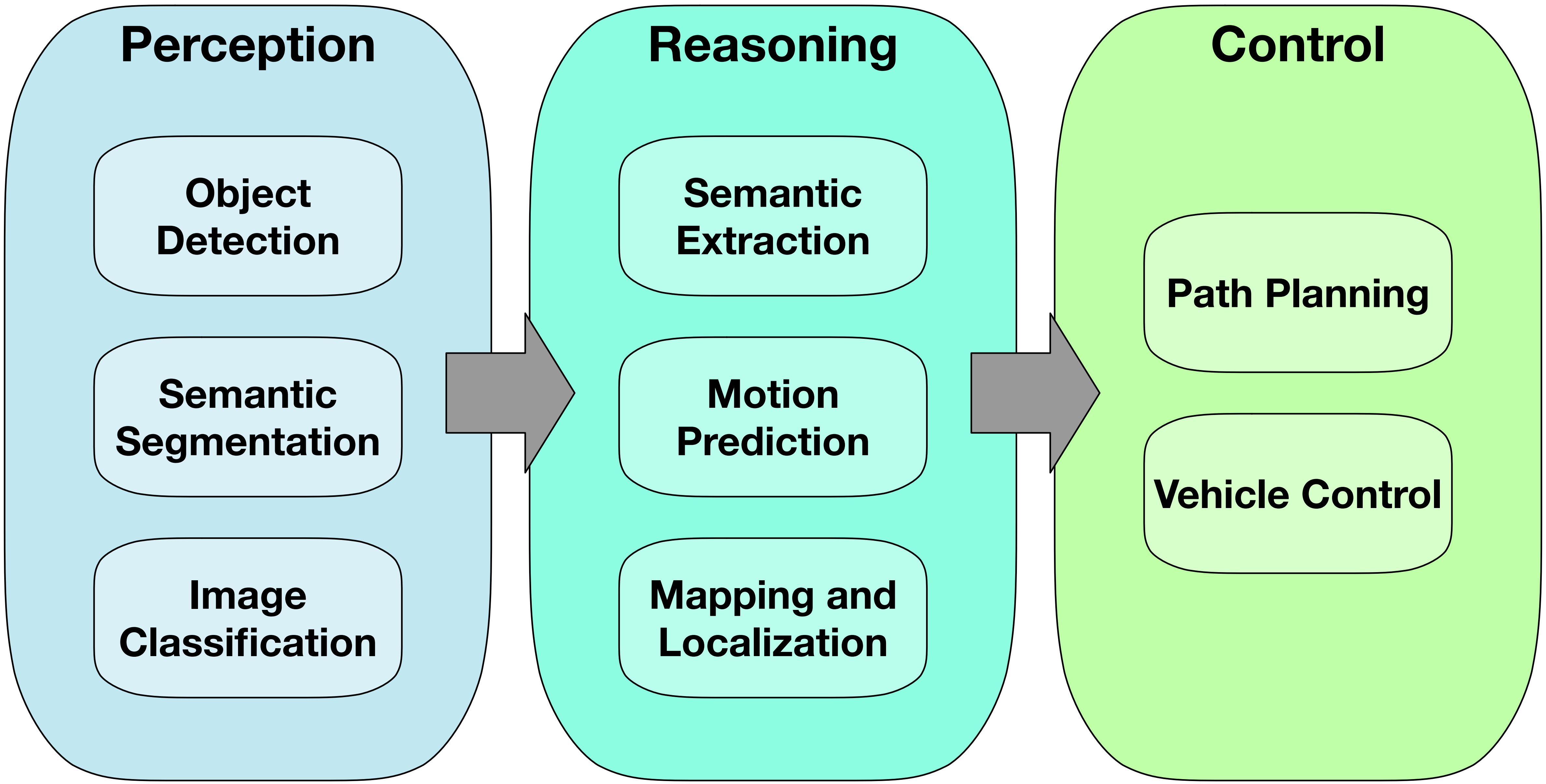






Autonomy

Autonomy Stack



Concatenation of complex Modules (often neural networks)

Autonomous cars and other large vehicles can support the stack, at the price of an increased cost and power expense

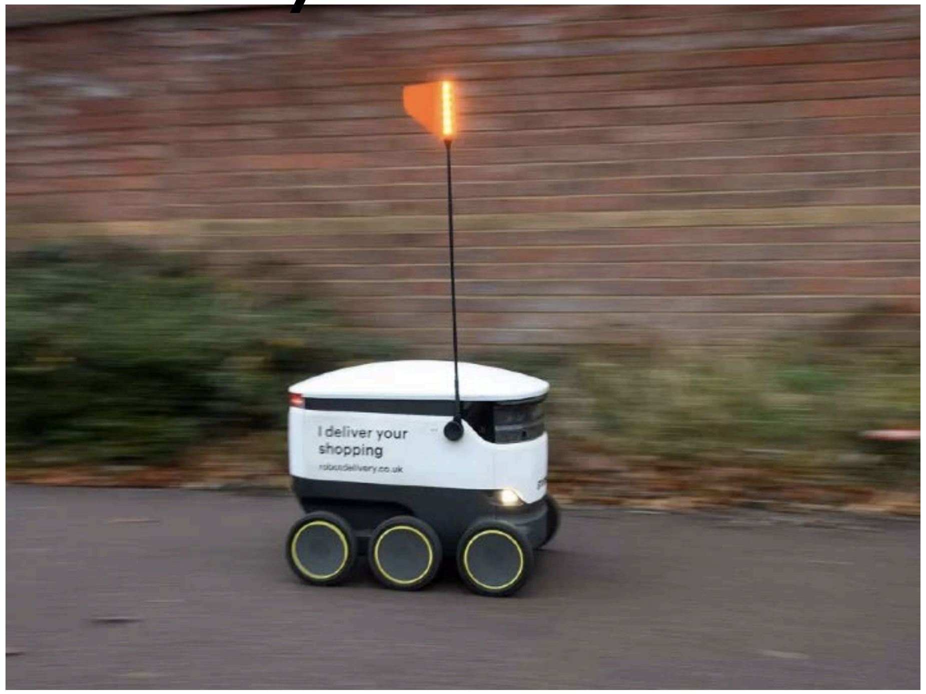
Autonomous cars and other large vehicles can support the stack, at the price of an increased cost and power expense

What if we want to empower small scale vehicles with advanced autonomy functionalities?

Defense

Logistics

Delivery



Or even just advanced computer vision…

Low-orbit monitoring

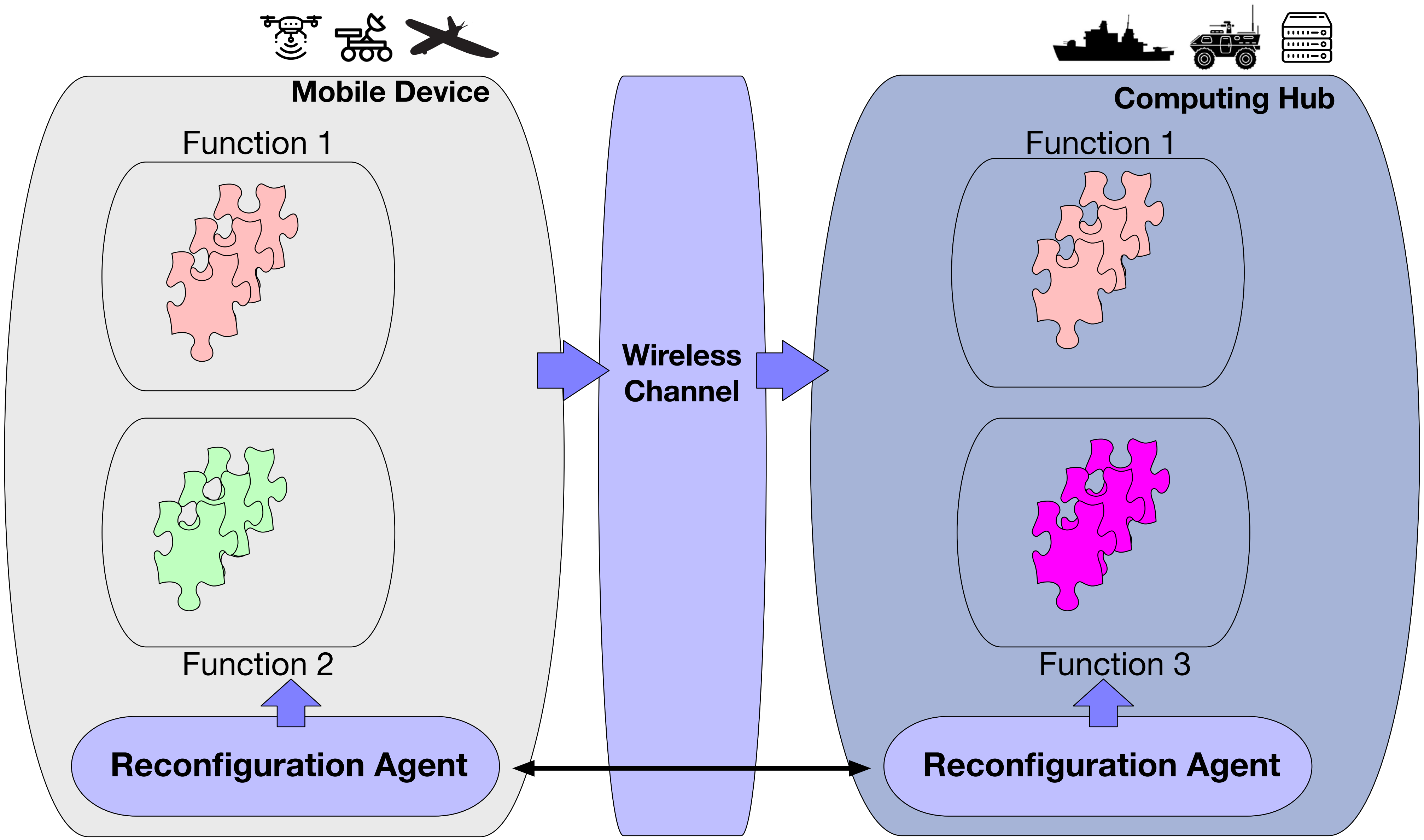
Traffic/City Monitoring

Objectives and Approach

Dynamic-Distributed AI pipelines that adapt to data and system context

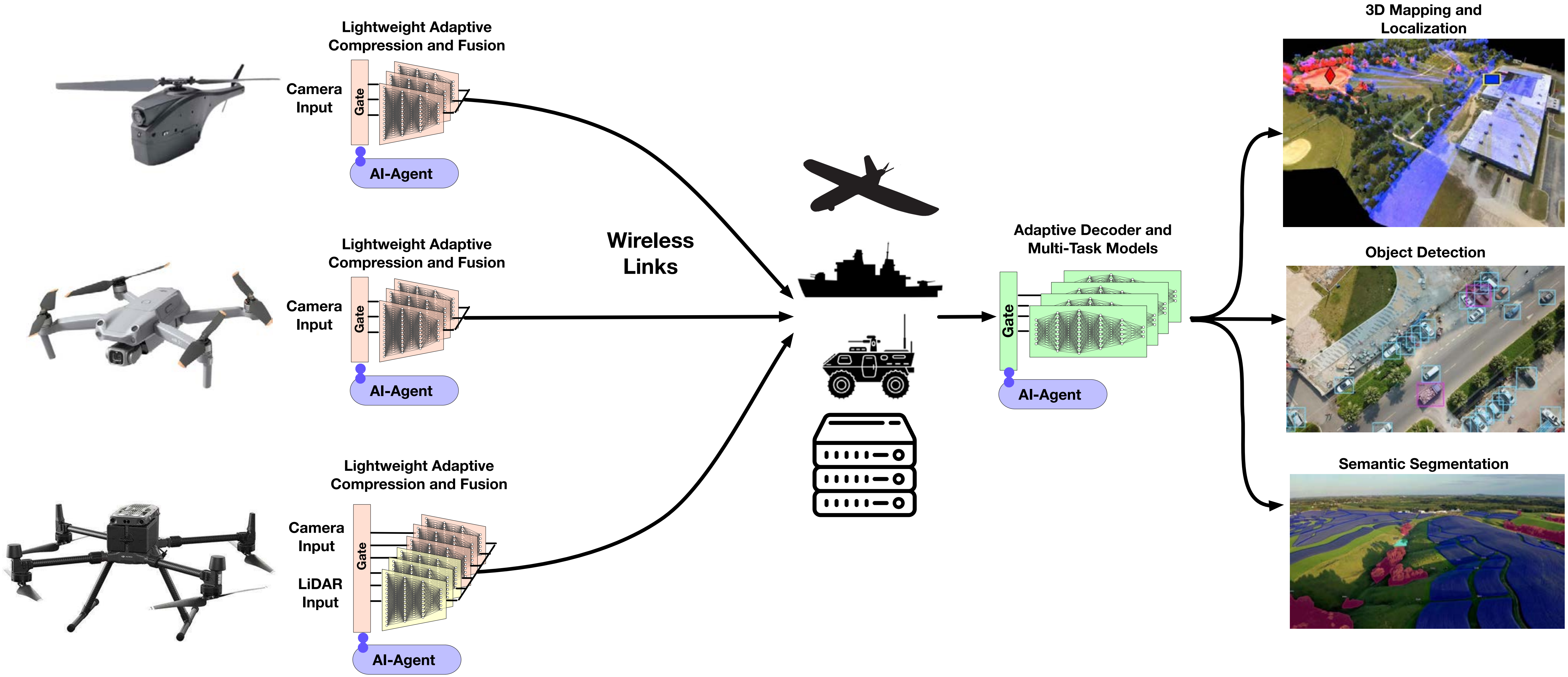
Objectives and Approach

Dynamic-Distributed Composable AI pipelines that adapt to data and system context



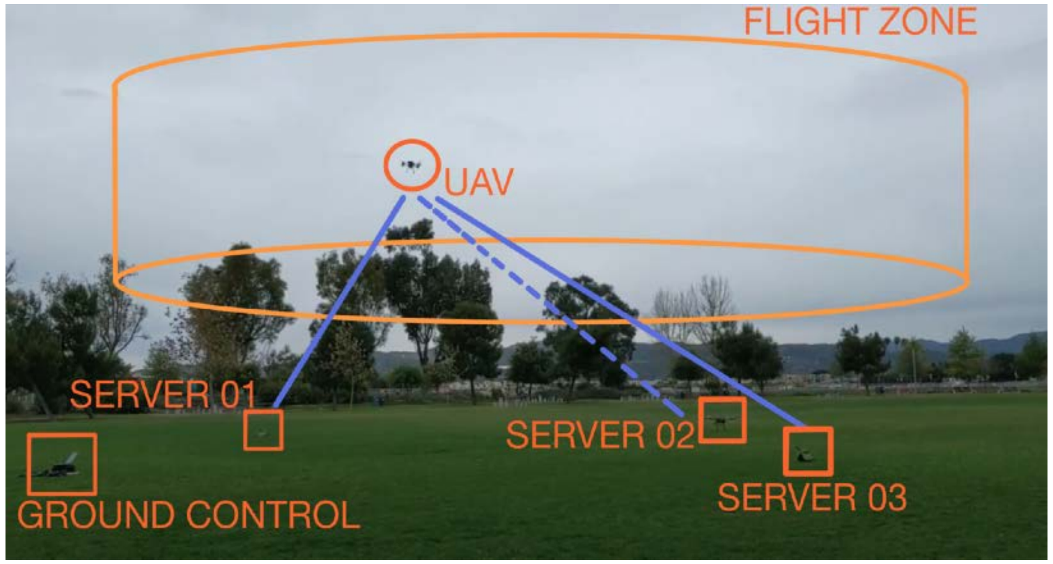
Objectives and Approach

Dynamic-Distributed Composable AI pipelines that adapt to data and system context



Hydra - Testbed

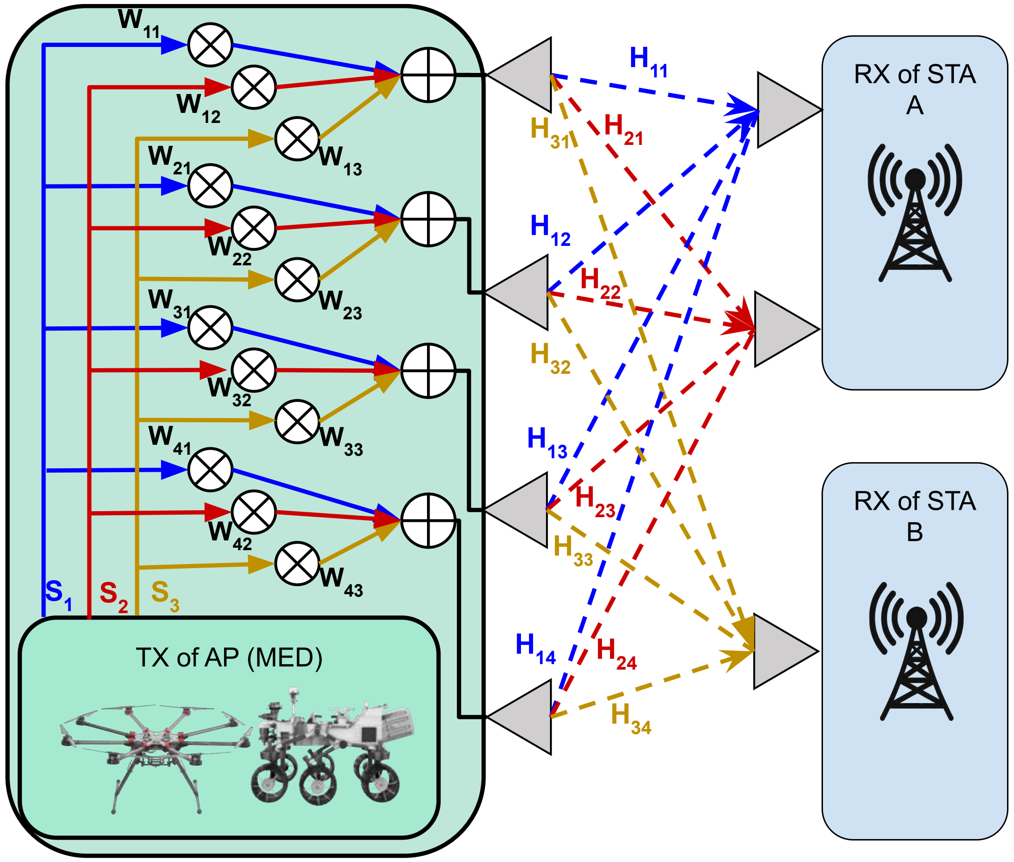
- Middleware for AI-controlled dynamic edge computing
- Airborne and ground autonomous vehicles connected to multiple edge servers
- Real-time telemetry, network and application logging for decision making
- Control of how many and which edge servers



Hydra - Testbed Current development

Advanced comms

- MU-MIMO comms
- Semantic control of comms parameters (bandwidth, number of antennas, connectivity) and packetization (replication vs parallelization)



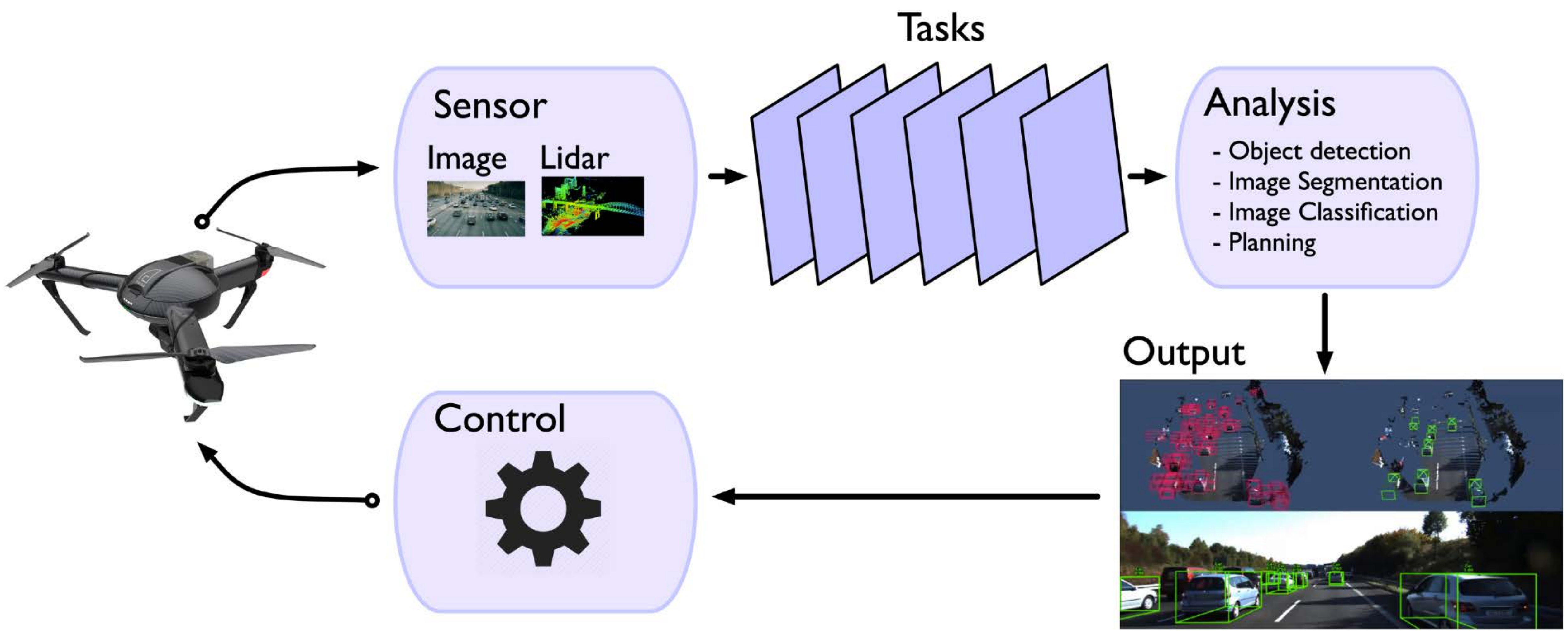
Indoor-Outdoor Navigation

- Dynamic reconfiguration of autonomy stack across "layers"
- Dynamic neural sensor fusion

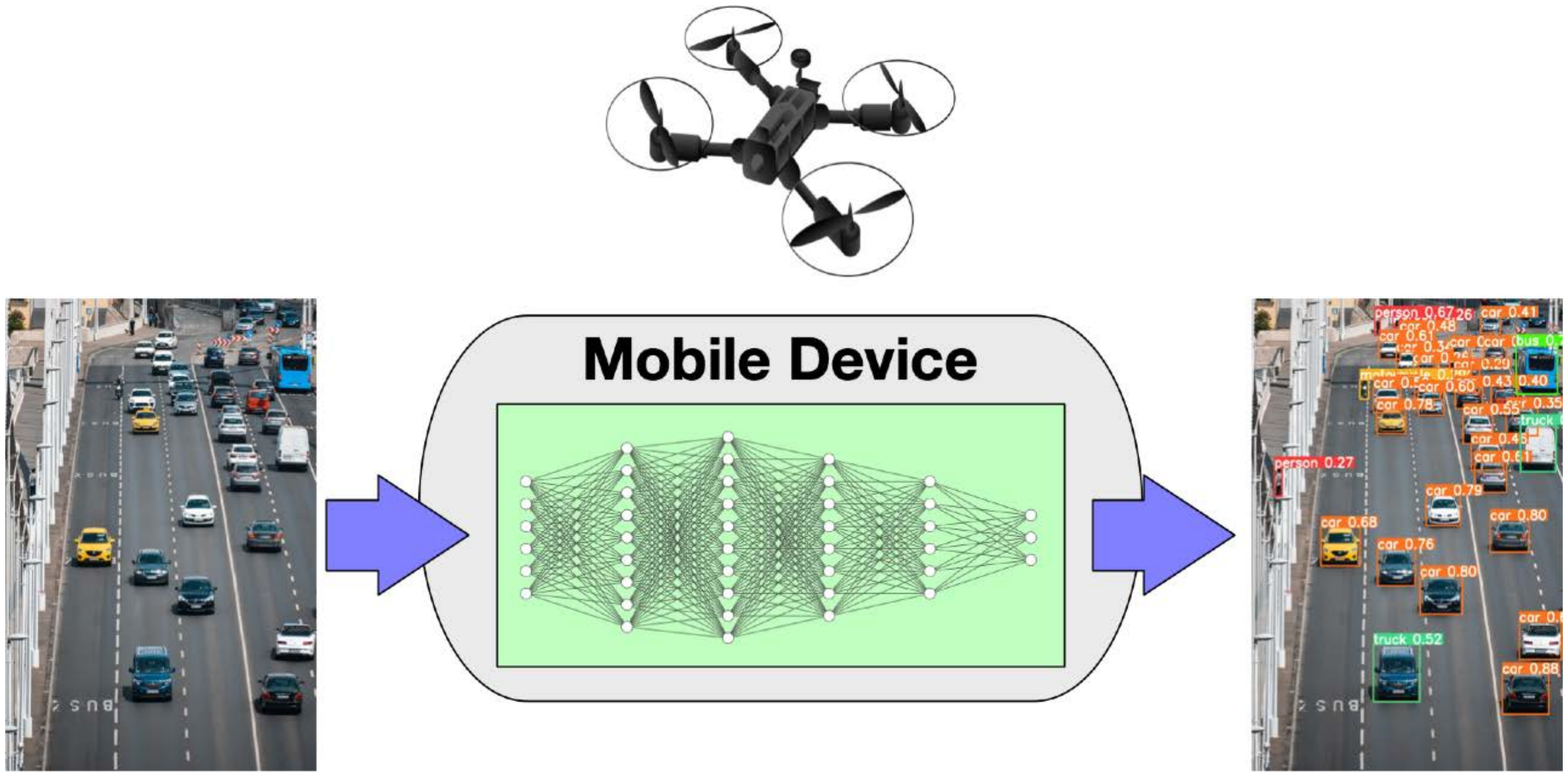
Preliminaries

Problem Illustration

Stream of data from multiple sensors to be analyzed to support mission functions and autonomous navigation

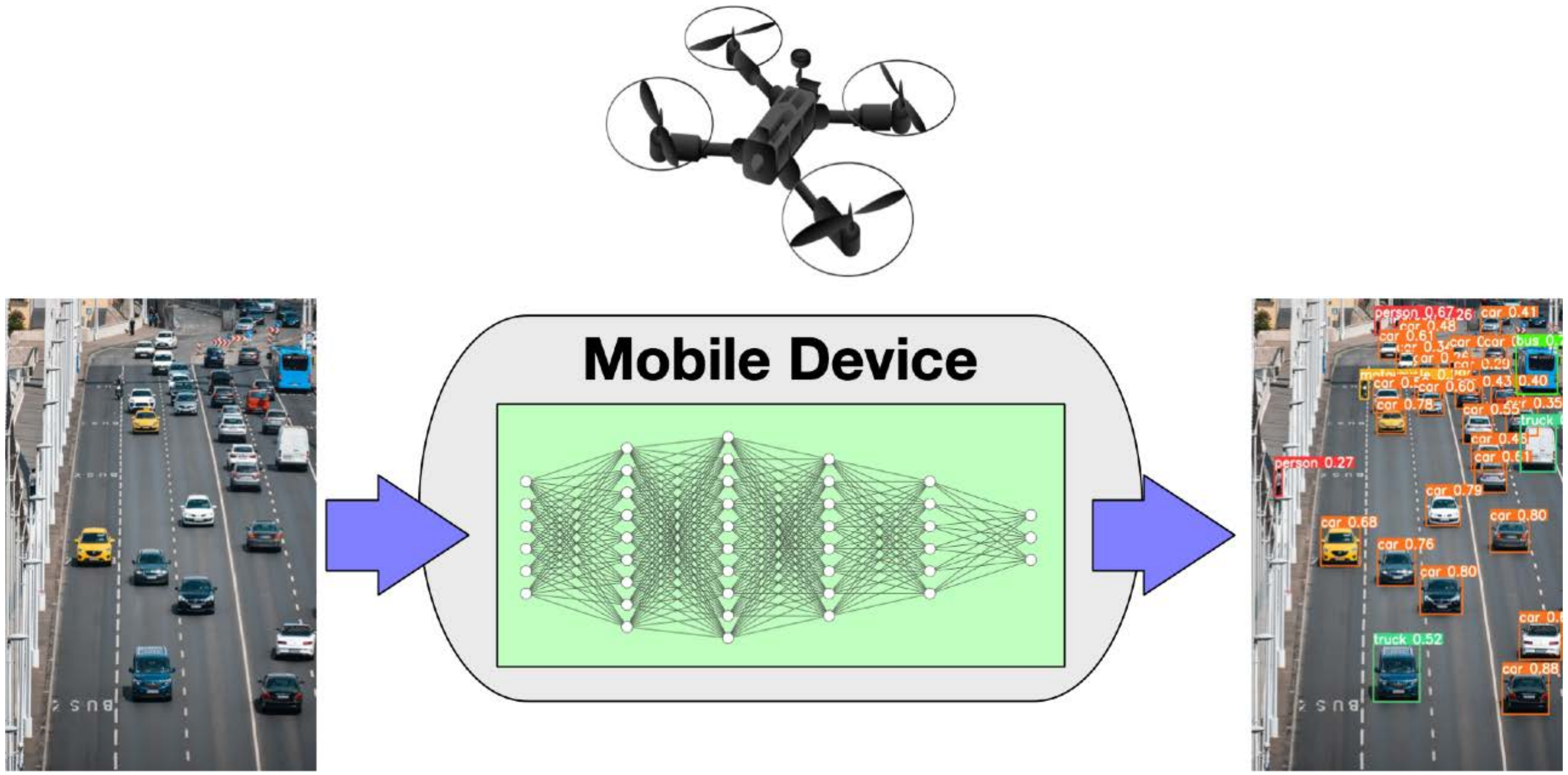


Onboard Computing



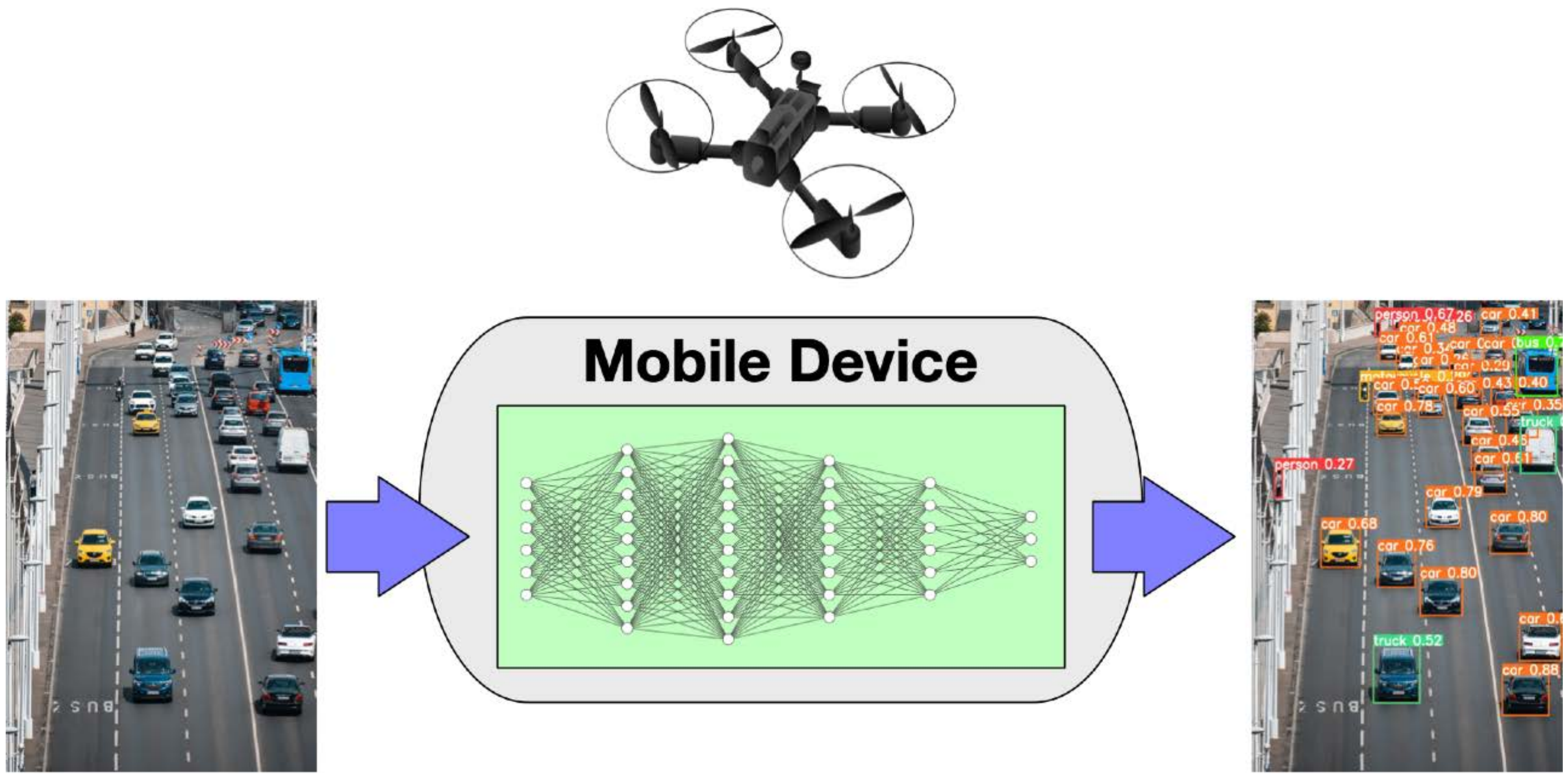
"Compressed" models deployed on the vehicle/robot
Quantization, distillation, pruning

Onboard Computing



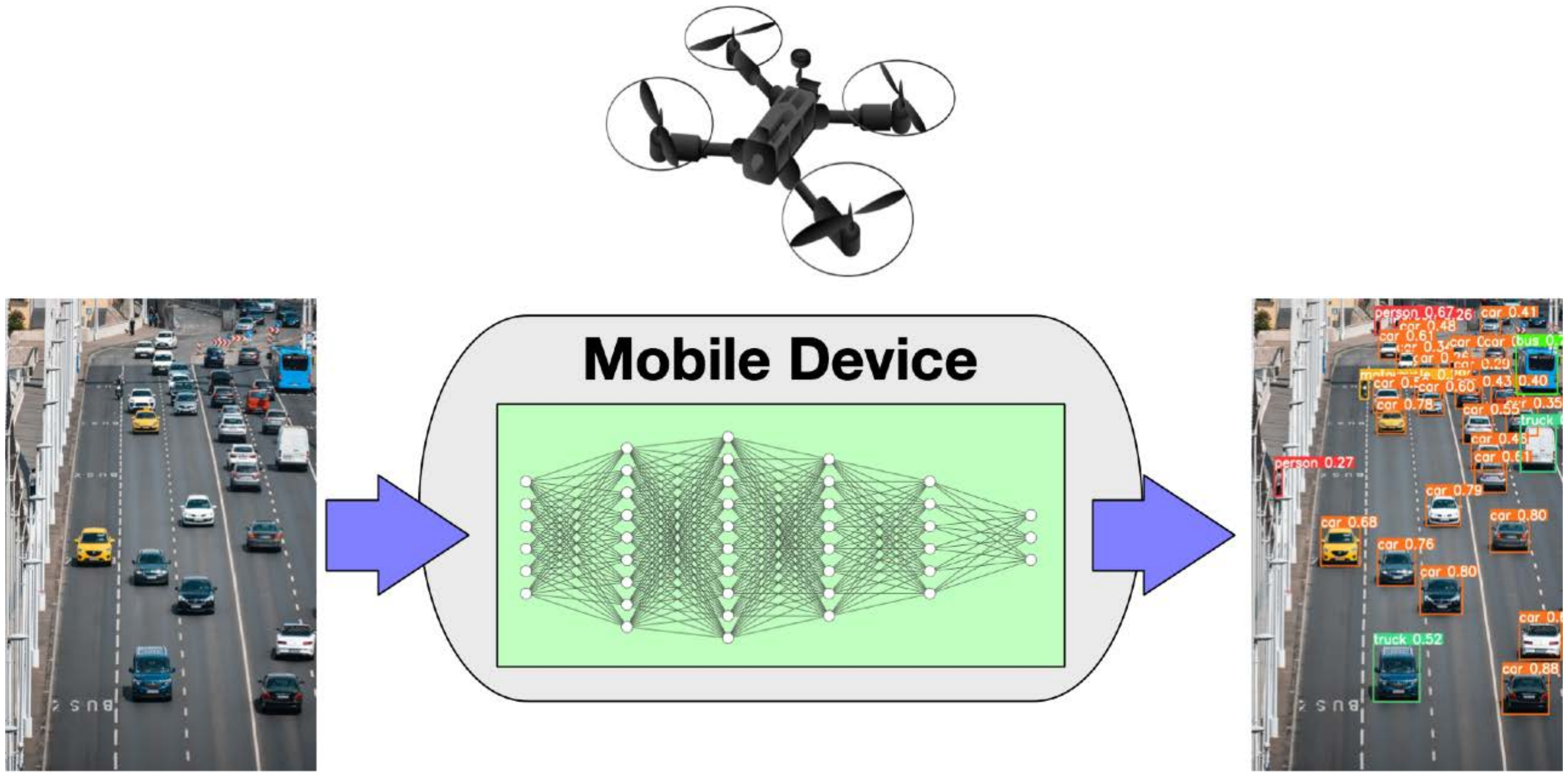
• **Hardware cost**

Onboard Computing



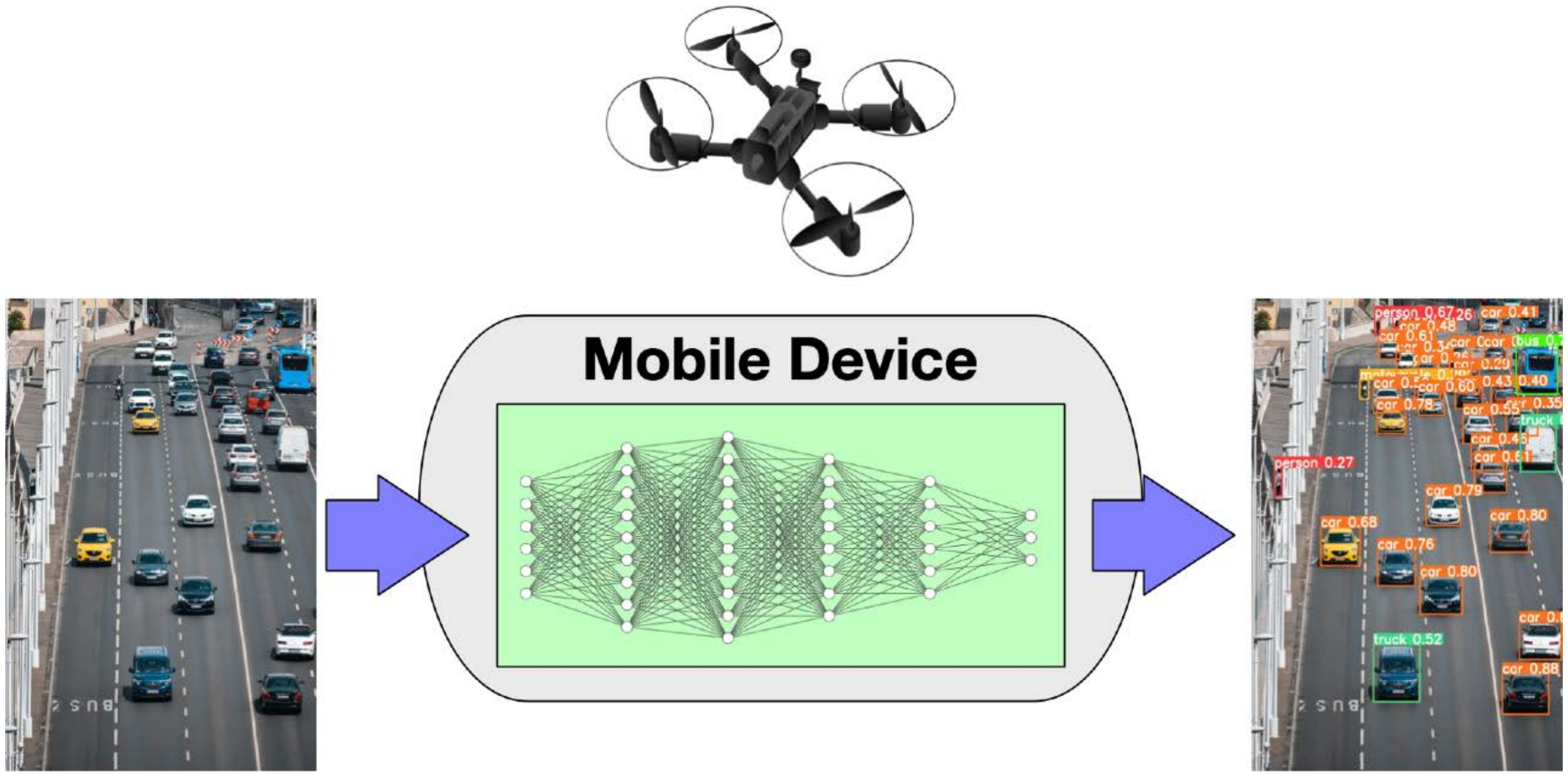
- **Hardware cost**
- **Task performance**
  - Model compression
  - Tradeoff between MAP/latency

Onboard Computing



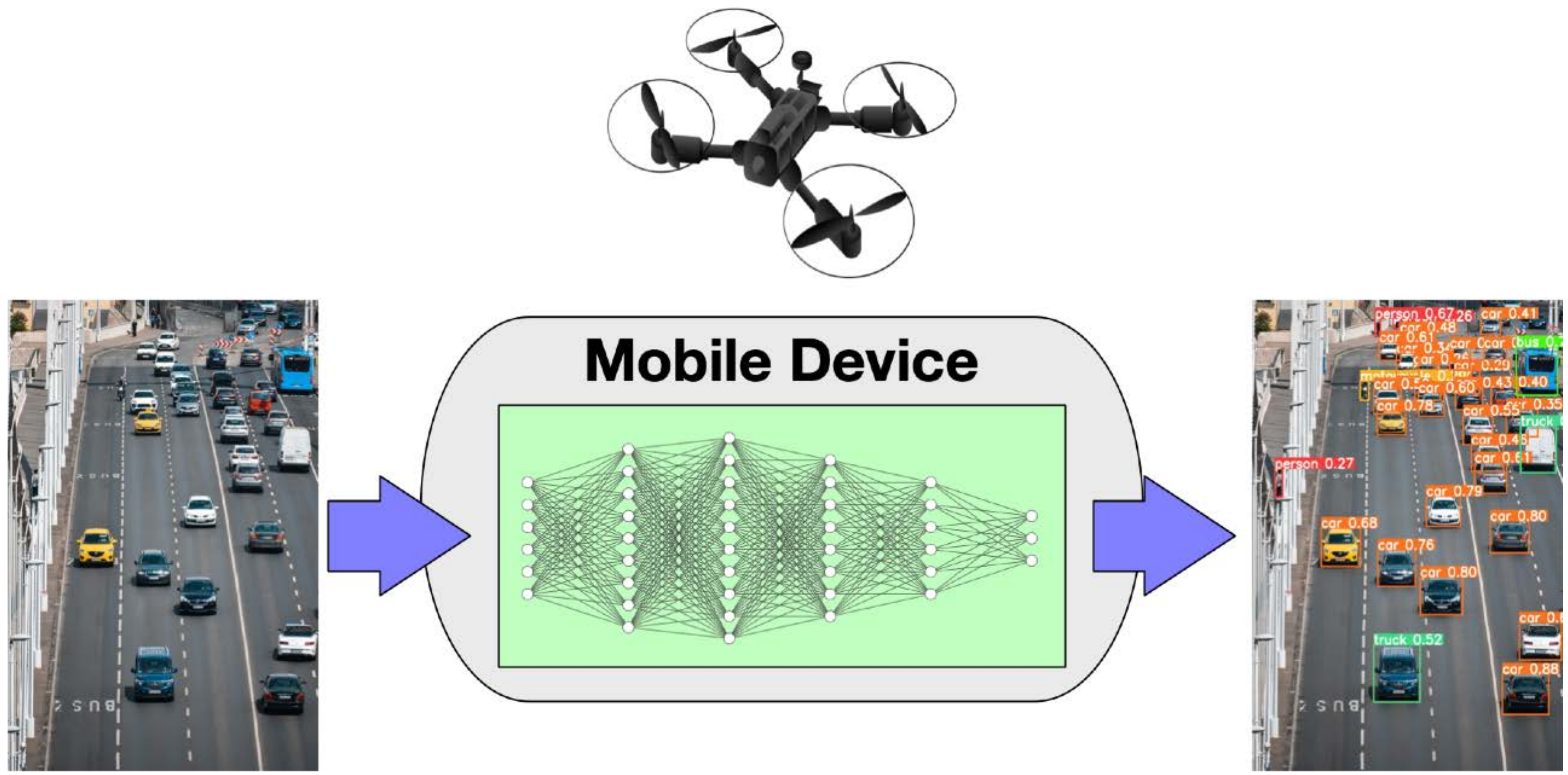
- **Hardware cost**
- **Task performance**
  - Model compression
  - Tradeoff between MAP/latency
- **Energy**

Onboard Computing



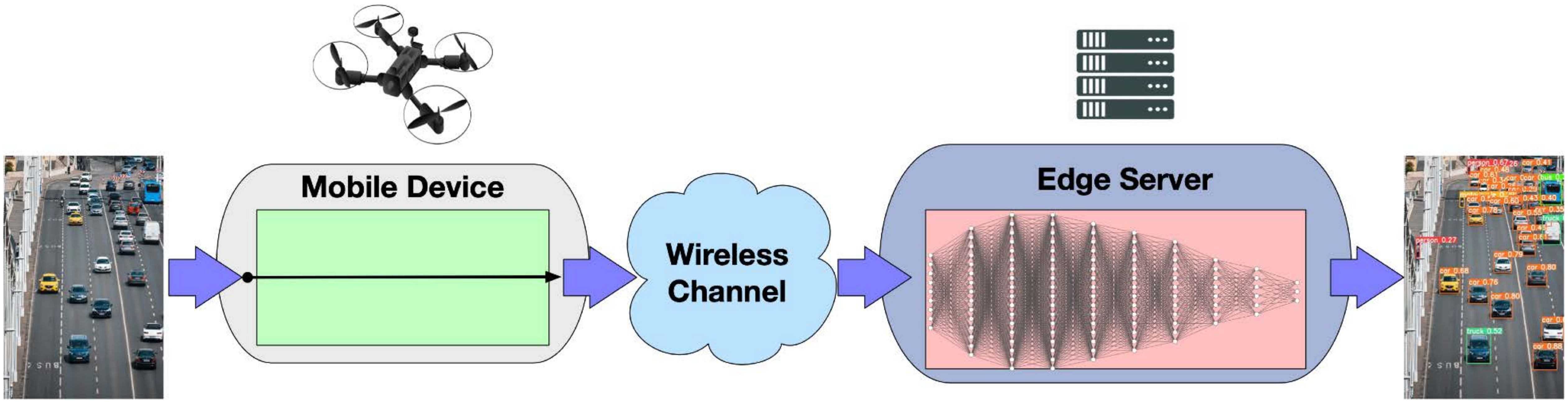
- **Hardware cost**
- **Task performance**
  - Model compression
  - Tradeoff between MAP/latency
- **Energy**
- **Hardware Degradation**

Onboard Computing



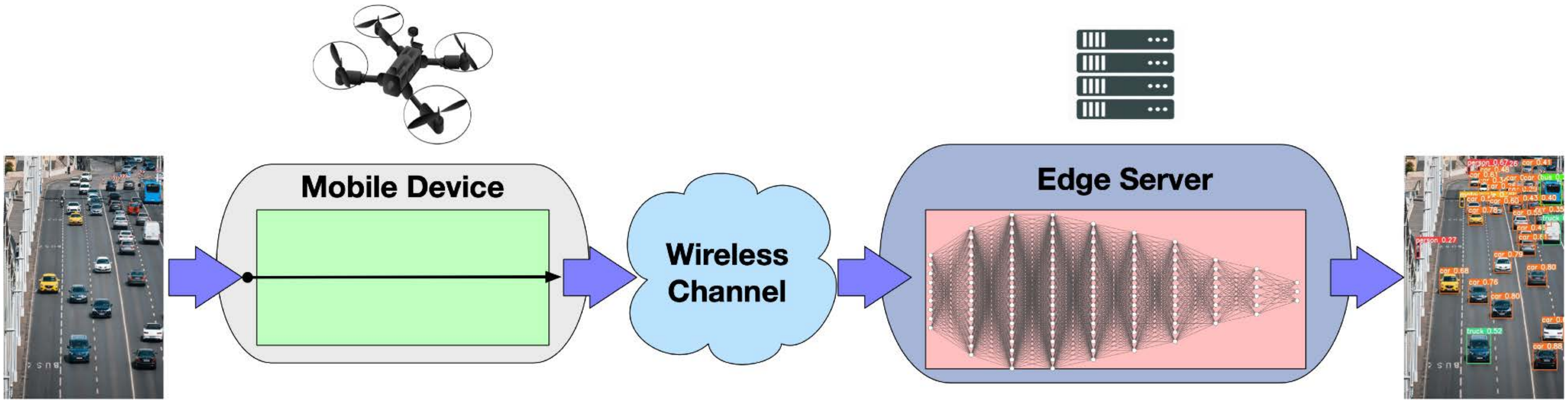
- **Hardware cost**
- **Task performance**
  - Model compression
  - Tradeoff between MAP/latency
- **Energy**
- **Hardware Degradation**
- **Limited functionalities**

Edge Computing



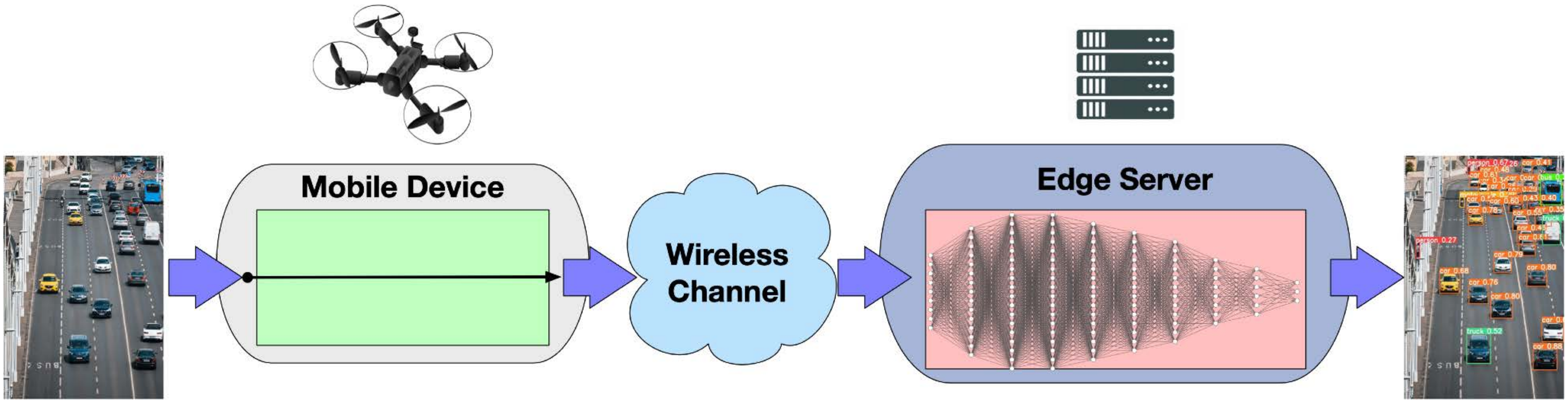
Data sent to a compute-capable device taking over the tasks

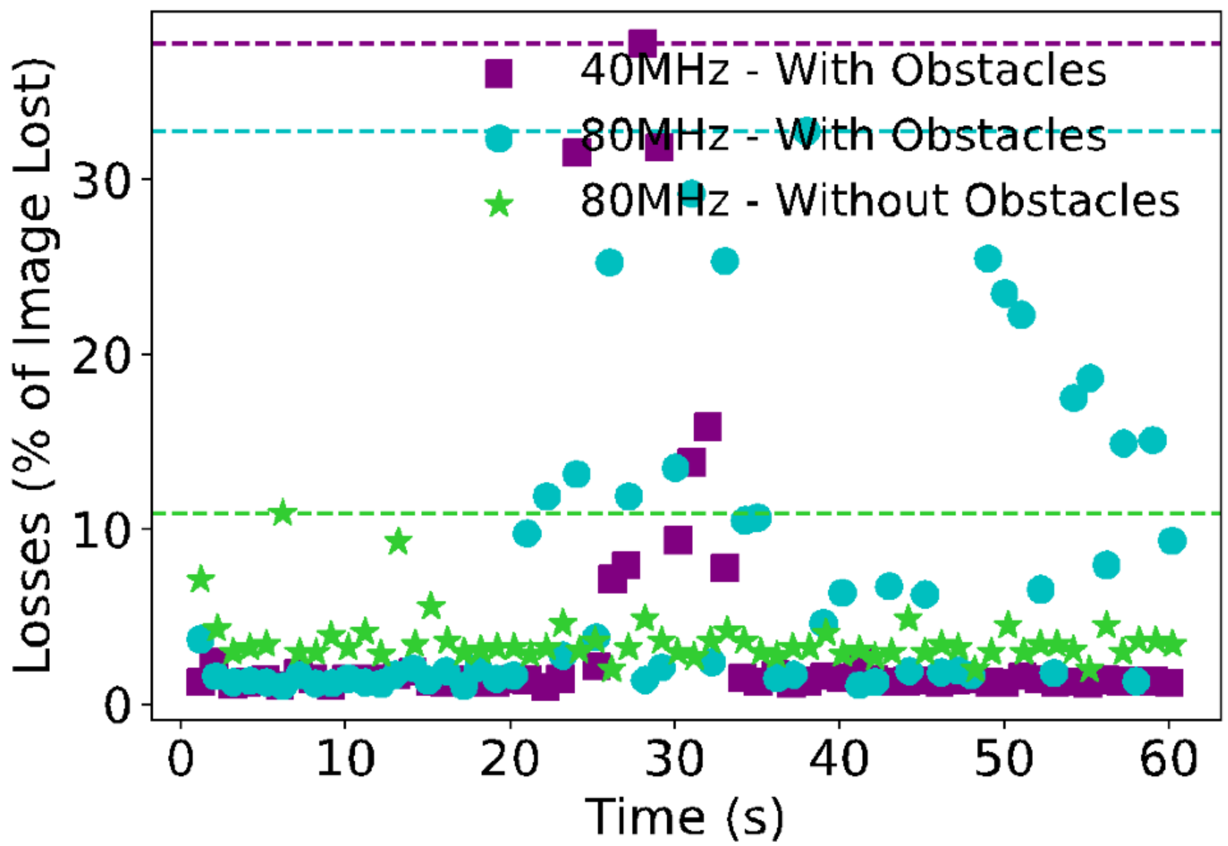
Edge Computing



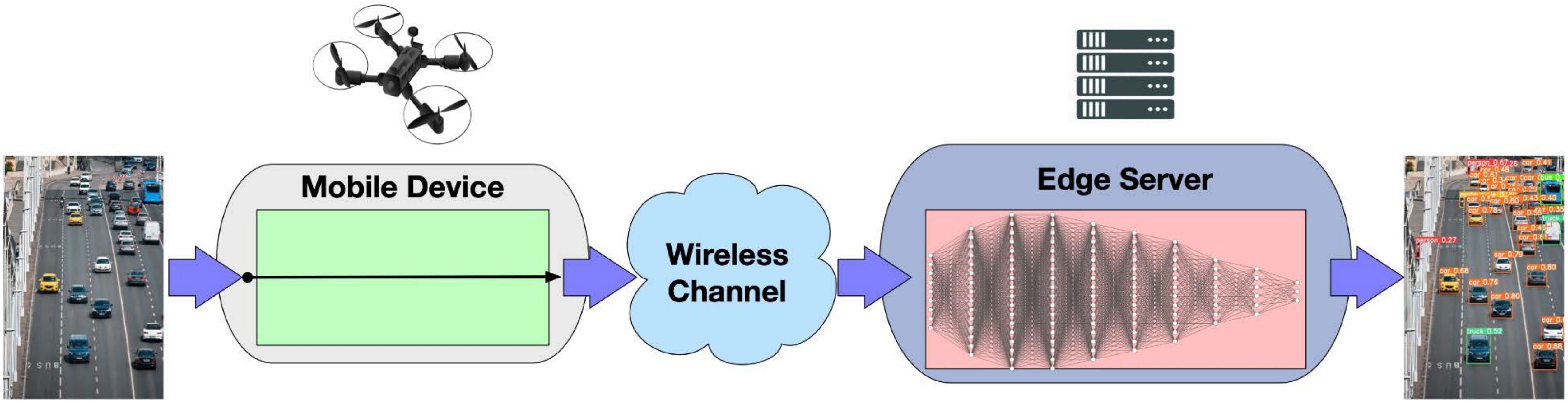
- **Latency/Latency variance**

Edge Computing



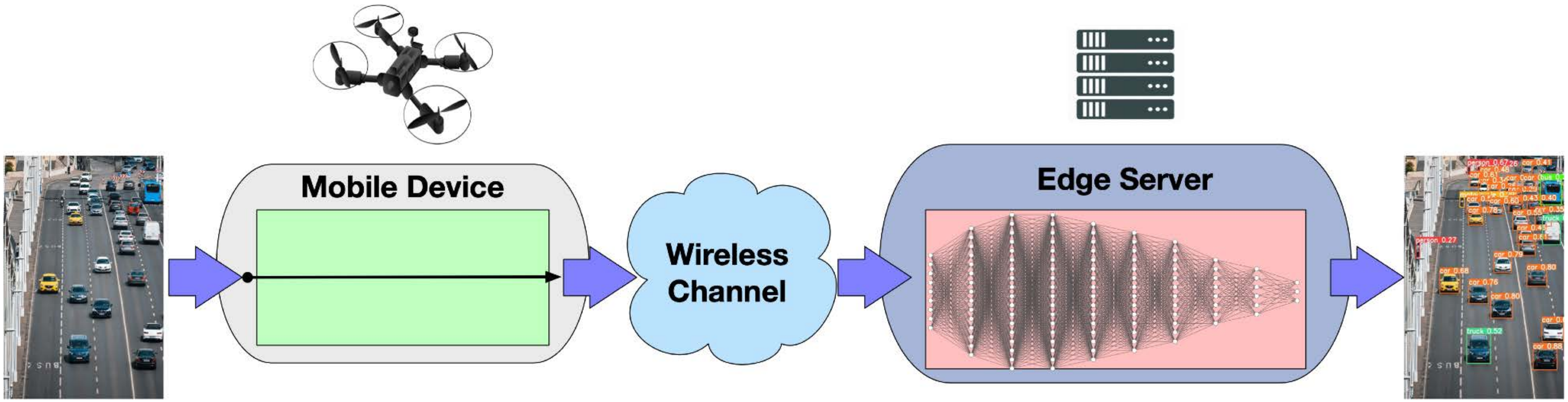


Edge Computing



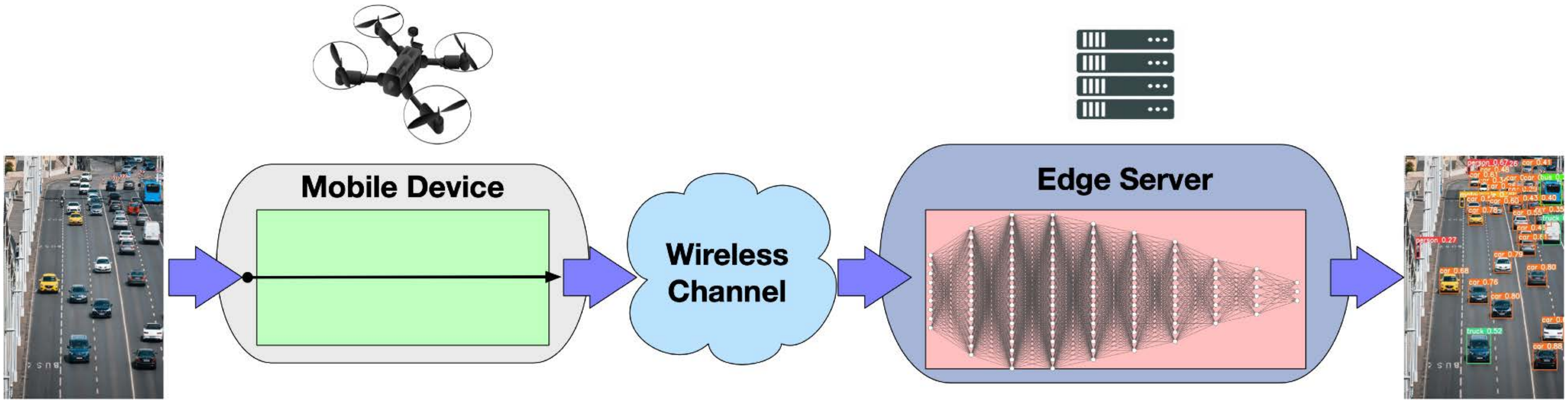
- **Latency/Latency variance**
- **Uncertainty**

Edge Computing



- **Latency/Latency variance**
- **Uncertainty**
- **Bandwidth usage**
  - Sharing with other users and services

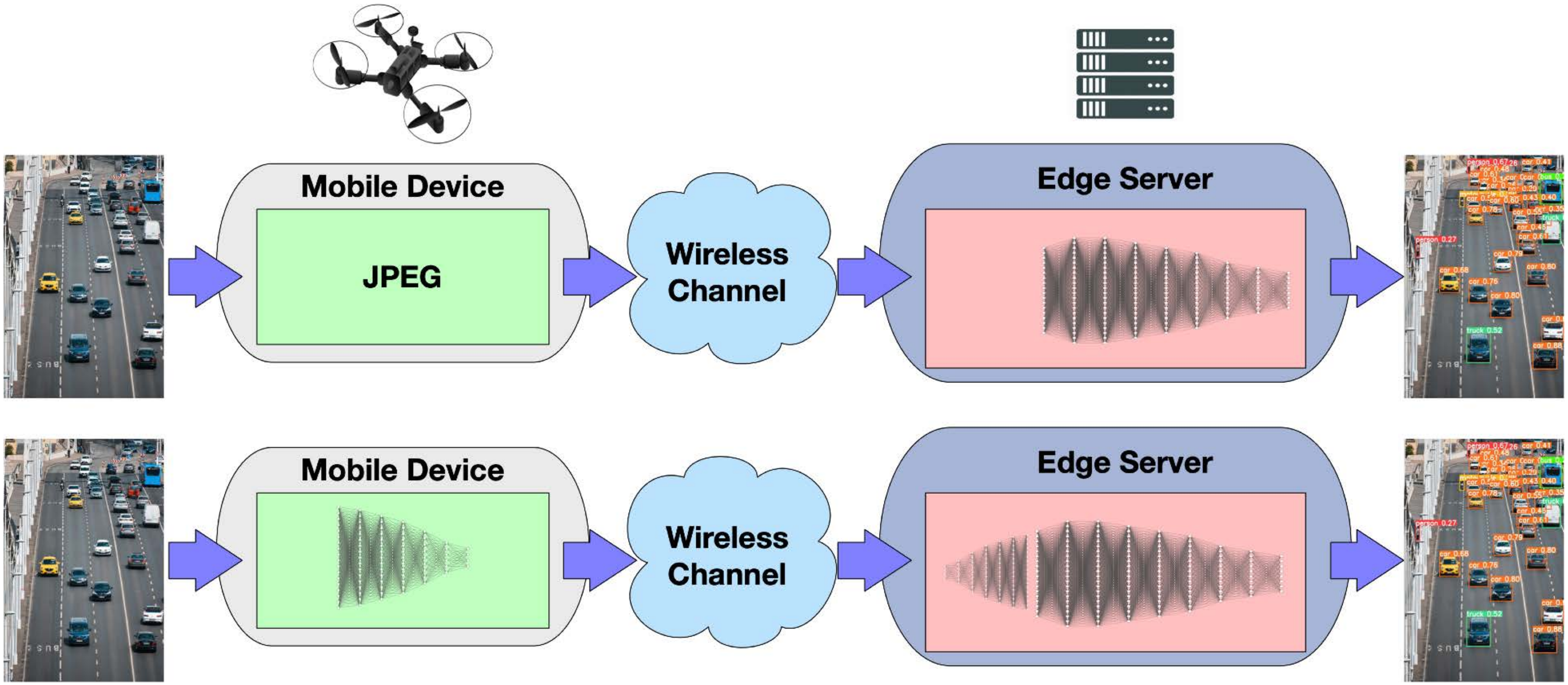
Edge Computing



- **Latency/Latency variance**
- **Uncertainty**
- **Bandwidth usage**
  - Sharing with other users and services
- **Hardware degradation**
  - Servers are more resilient

Distributed AI

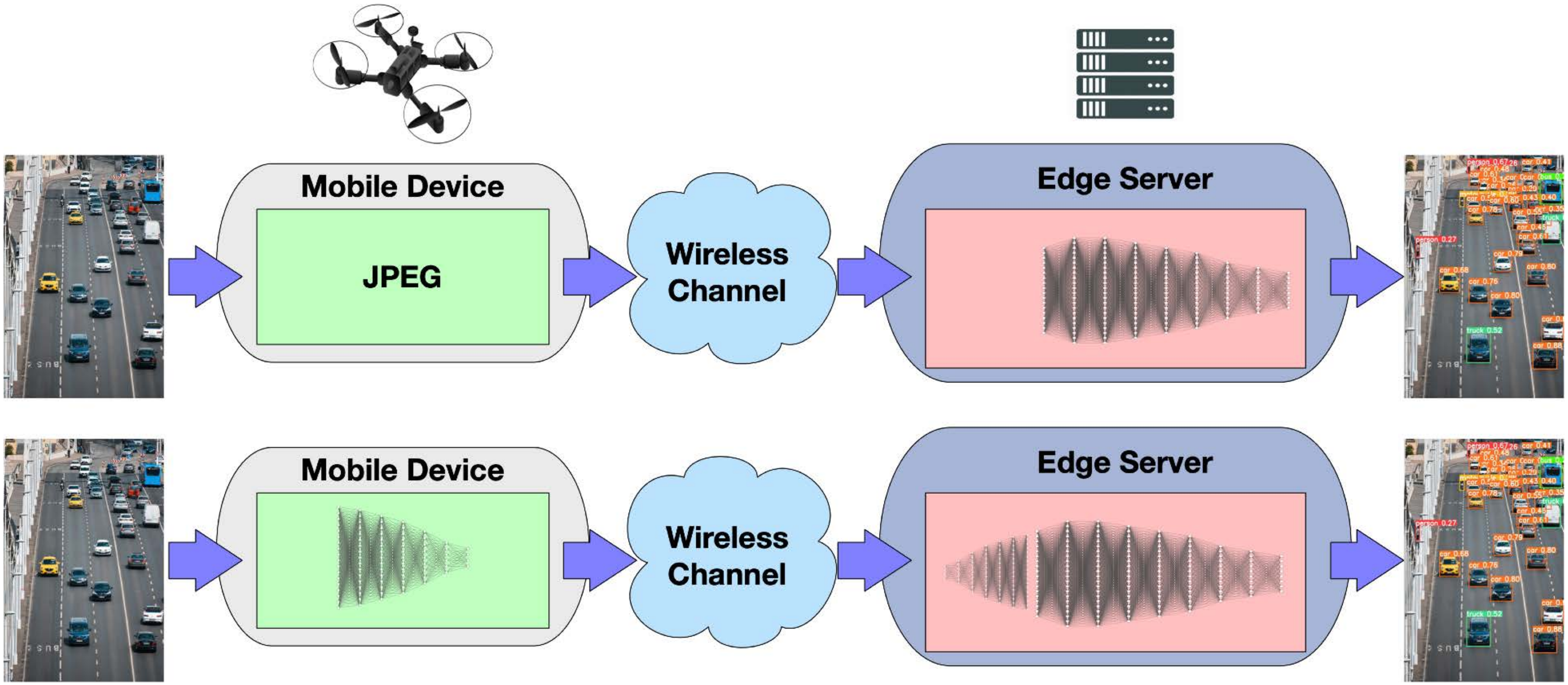
Compression



JPEG

- Low complexity
- Bad rate-distortion curve (high compression gain)
- Designed for human perception

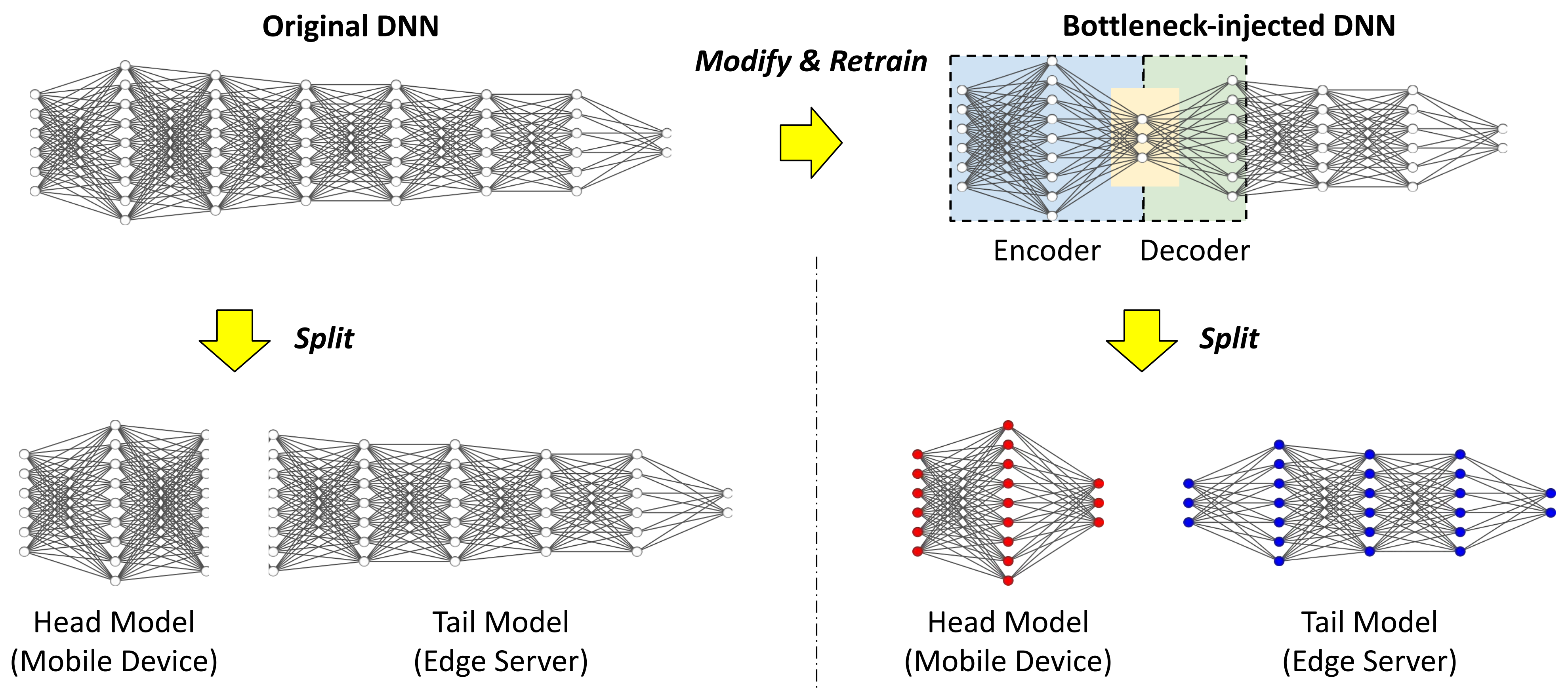
Compression



Neural Encoders

- High complexity (mobile device and server)
- High performance
- Designed to reconstruct the input image

Split Deep Neural Networks

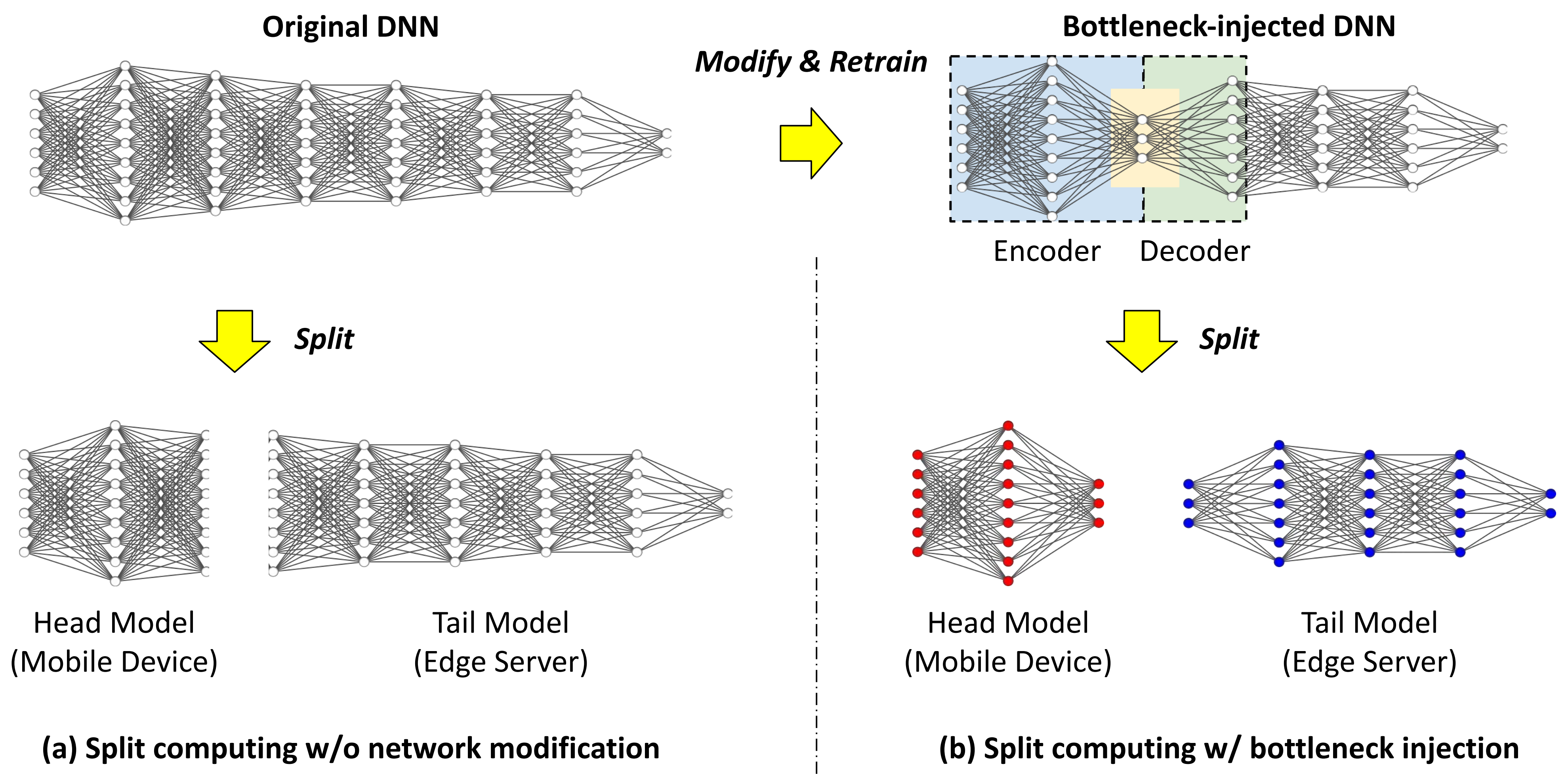


(a) Split computing w/o network modification **(b) Split computing w/ bottleneck injection**

Trivial Split DNN

- Distribution of computing load
- Compression only if split point is toward the end of the model
- Optimal latency often at extreme point

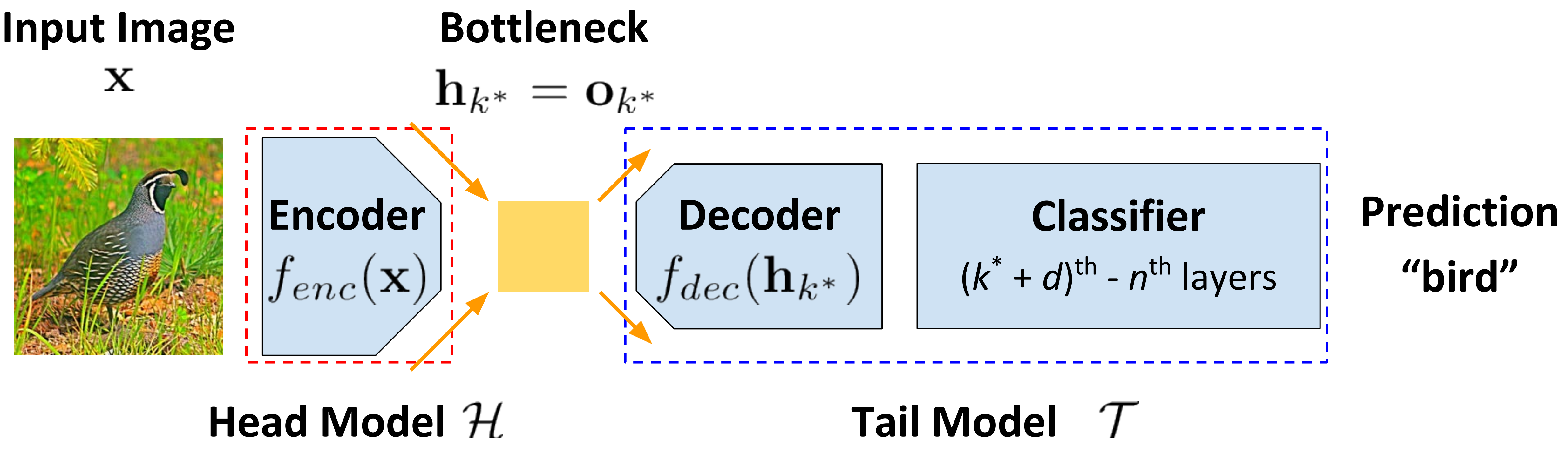
Split Deep Neural Networks



"Artificial" Bottleneck

- Architecture altered to incorporate a bottleneck (in-model compression)
- Objective: minimal complexity - maximum compression - maximum task performance
- Specialized training

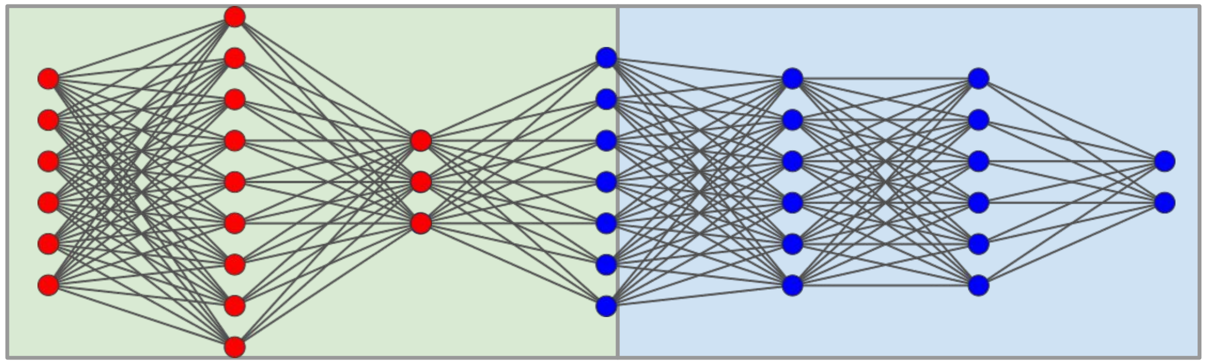
Supervised Compression



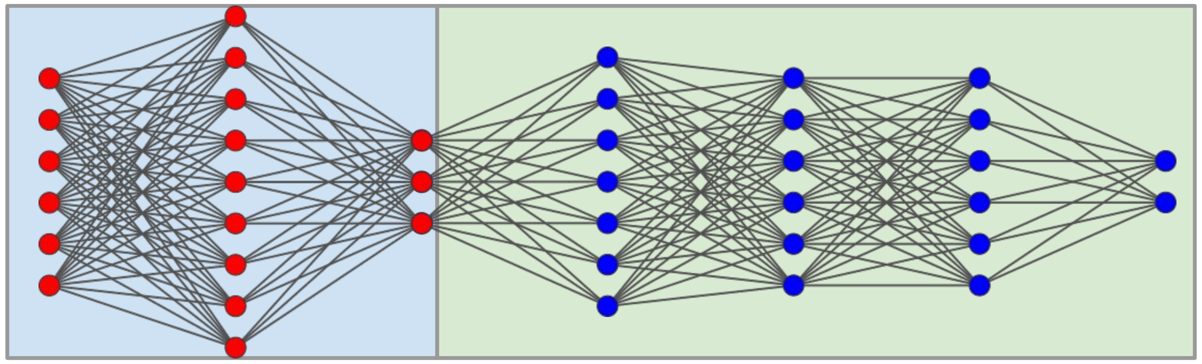
- Encoder/decoder-like structure within the model
- Semantic in-model compression obtained at the splitting point

Training

Stage 1: **Encoder/Decoder Training**



Stage 2: **Fine-Tuning to Task**



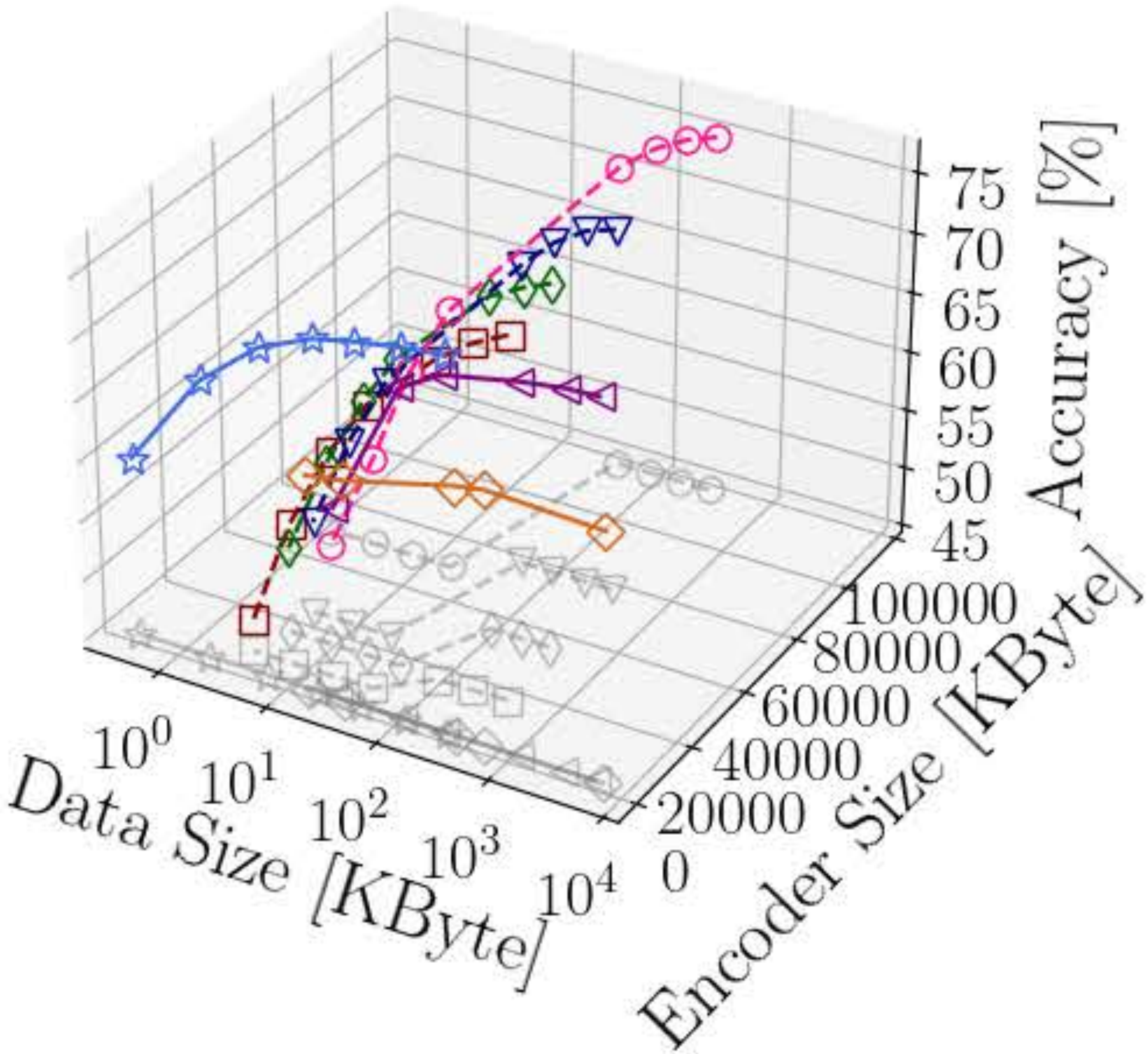
Trainable   Frozen

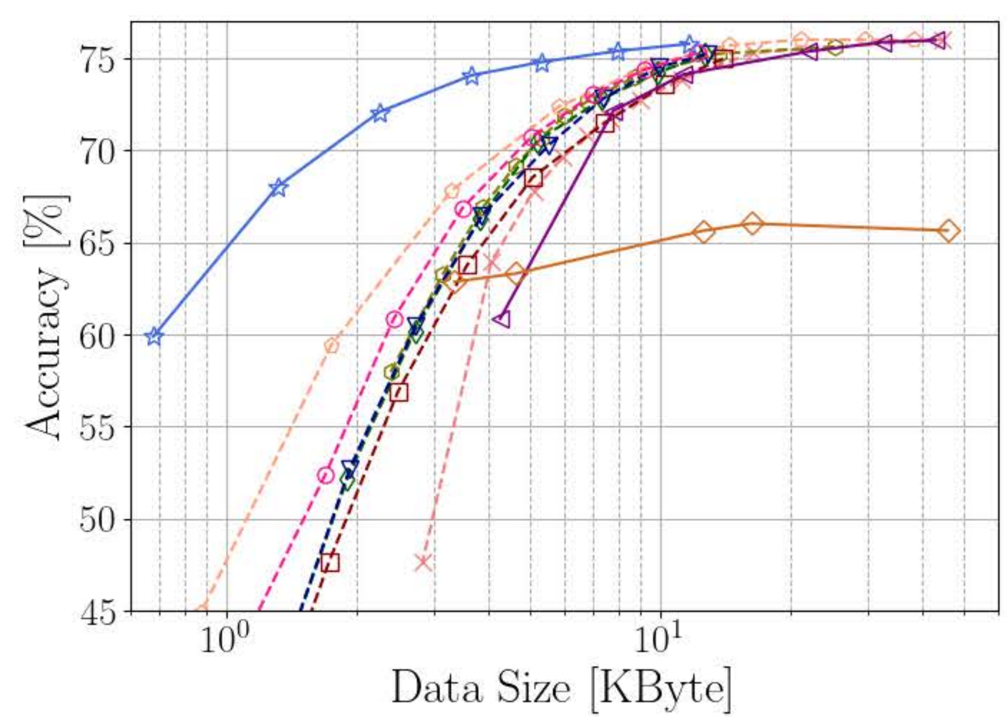
Multistage Training

- Encoder-decoder trained to reproduce an intermediate layer of the original model
- Supervised task-oriented compression: representation if trained to shed irrelevant bits

Performance

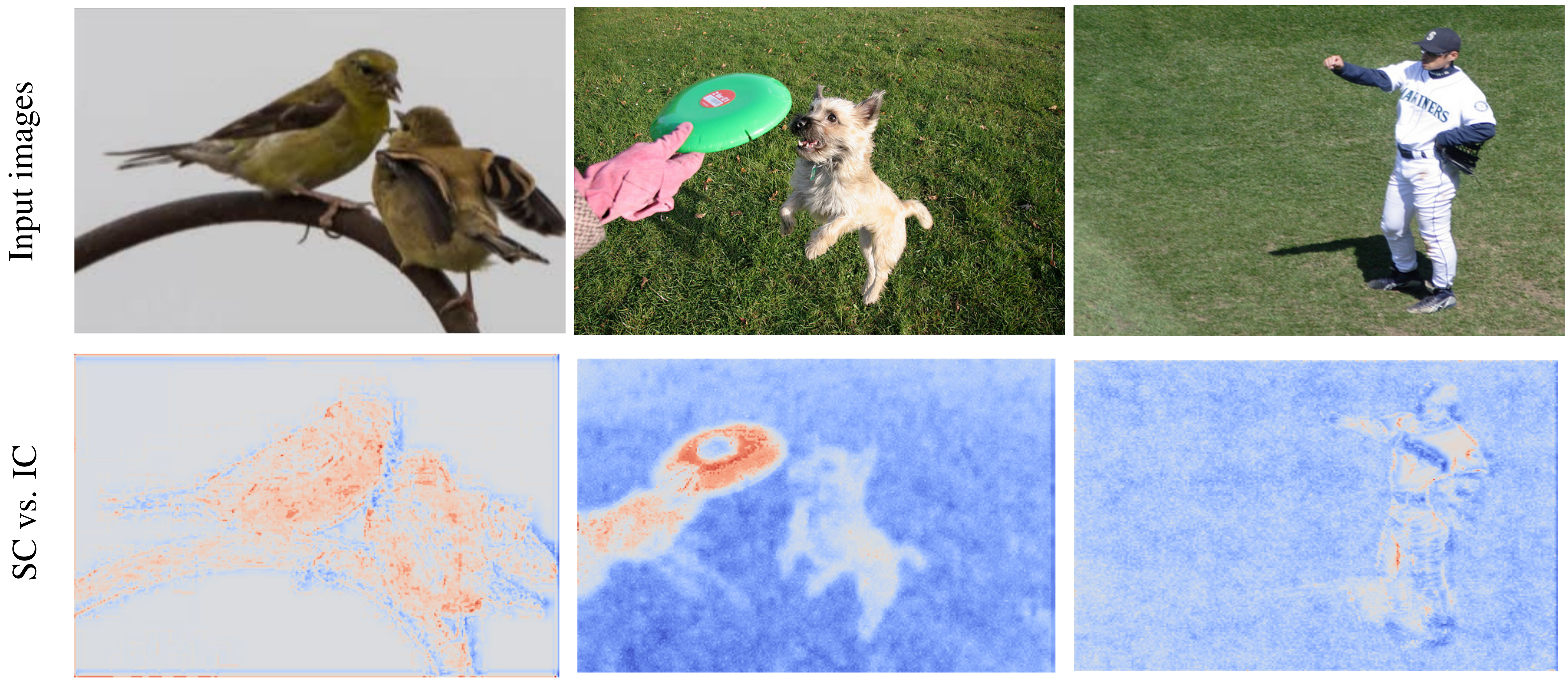
Rate-Distortion-Complexity Curve





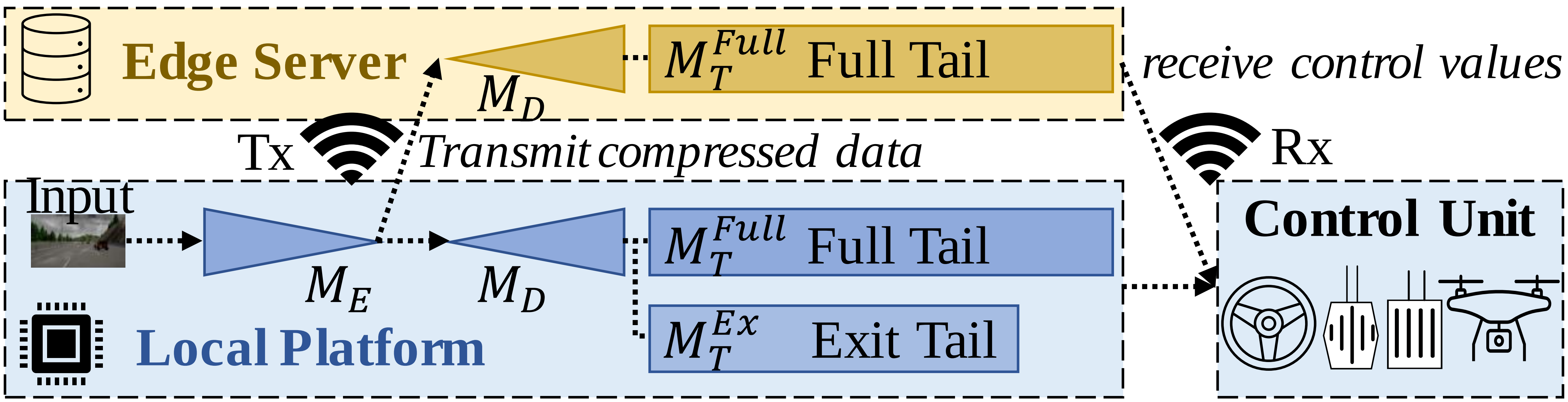
Bit Allocation

Visualization: bit allocation with respect to a variational autoencoder



Multi-Branched Split Architecture

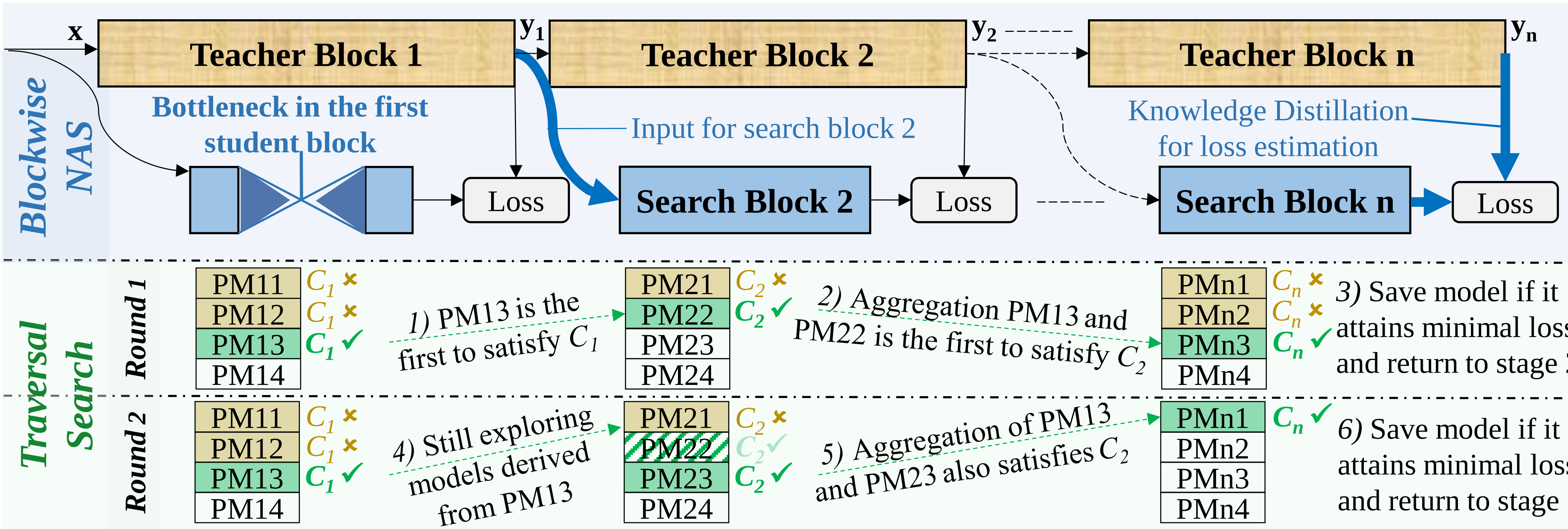
Dynamic Neural Network for Autonomous Vehicles



Local and Edge pipelines dynamically selected based on sample and system parameters

- Computing path dynamically controlled by a lightweight AI agent
- Predictive logics based on DRL

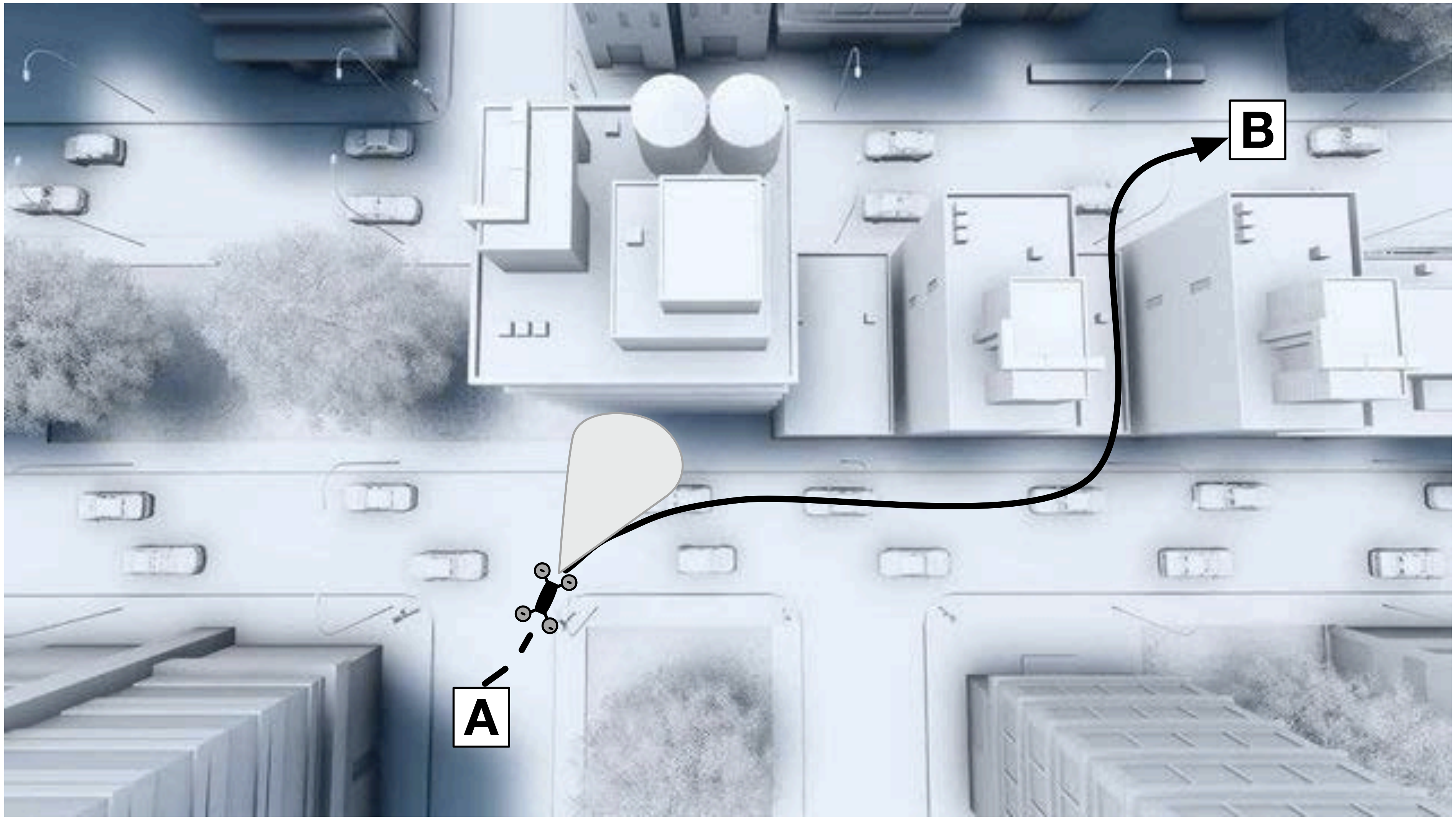
Dynamic Neural Network for Autonomous Vehicles



- Architecture automatically generated using Neural Architecture Search based on system parameters
- Optimized position and shape of the bottleneck

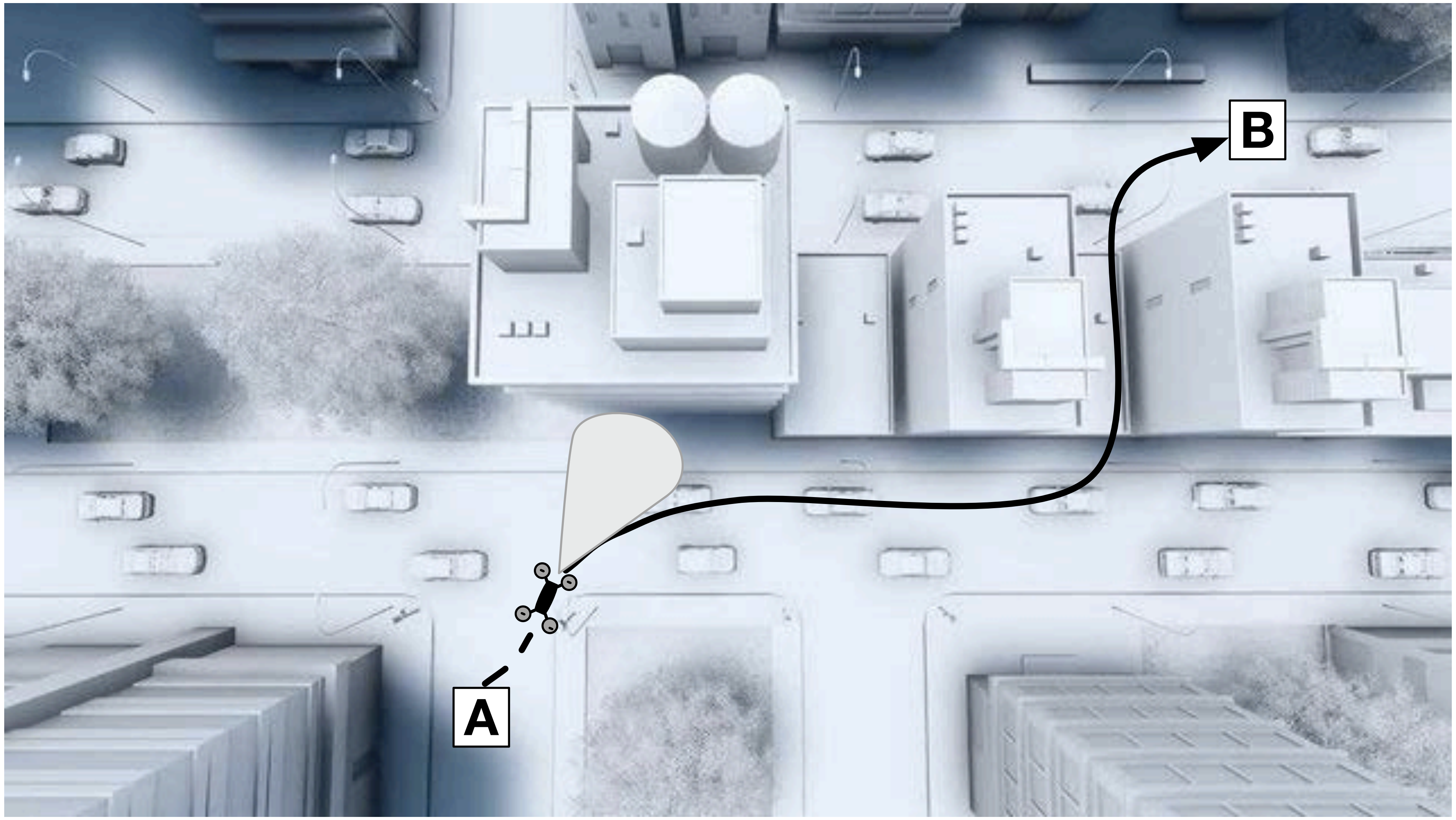
Split Self-Adaptive AI for Navigation

NaviSplit



Navigation problem: nano/microdrone autonomously determines path to reach point B from point A in an unknown environment

NaviSplit

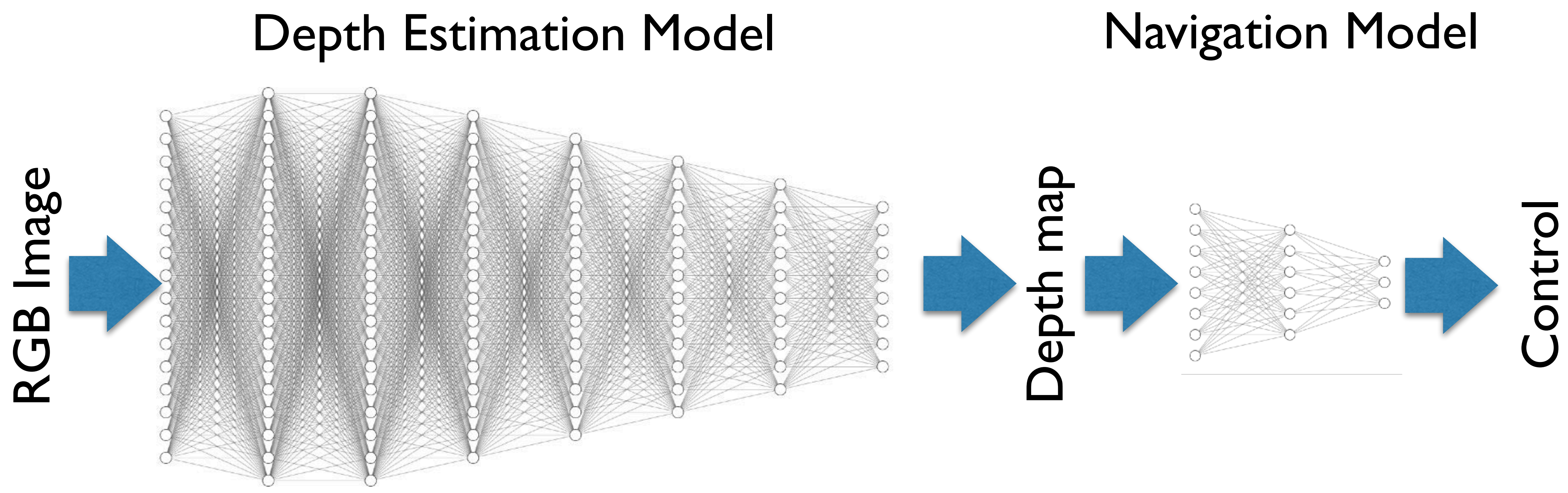


Input: (a) GPS, (b) RGB image (no depth)

NaviSplit

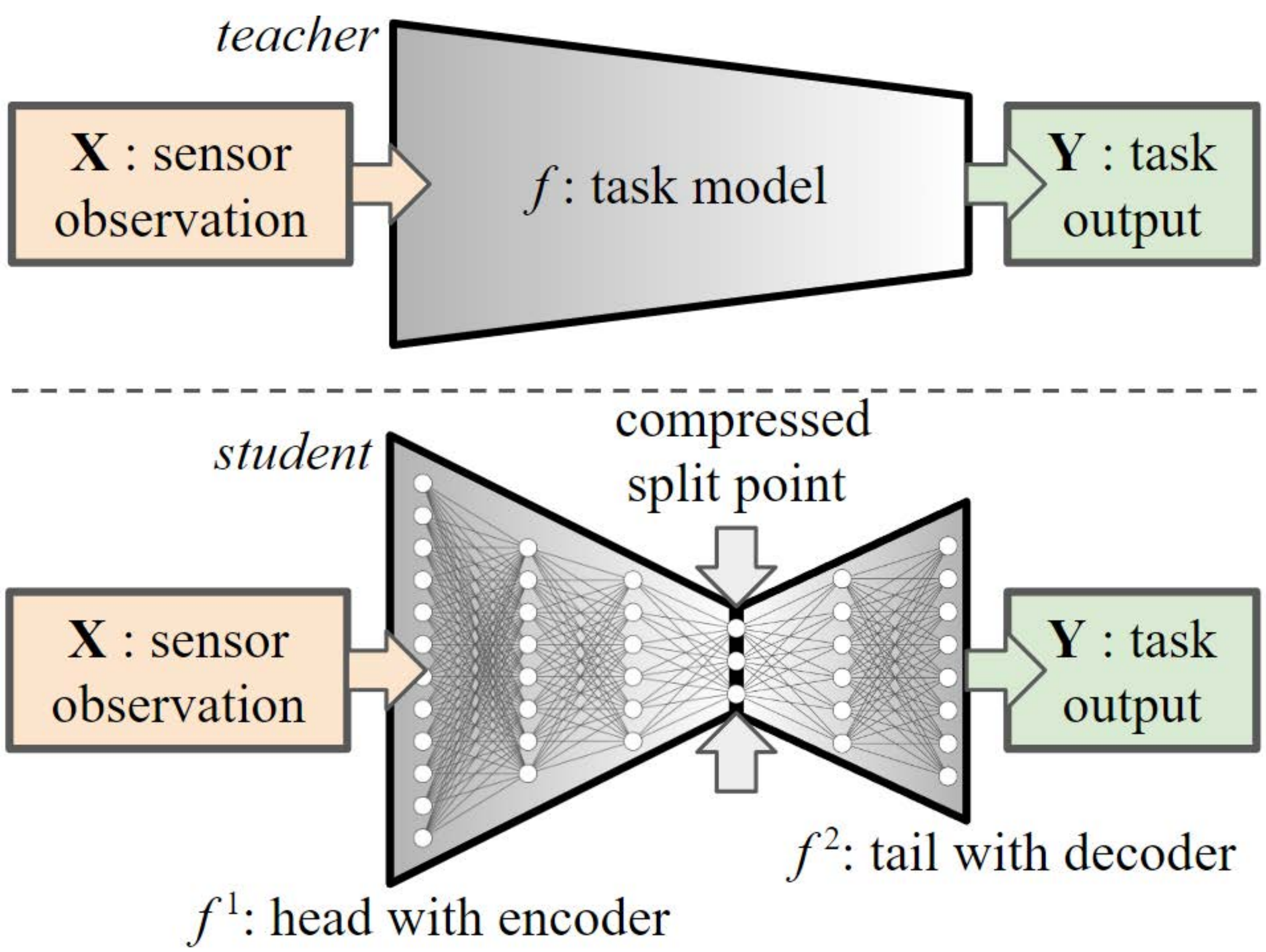
Two neural models:

- **Depth estimation:** neural model transforms the RGB image into a depth map
- **Navigation:** neural model transforms the depth map into motion commands

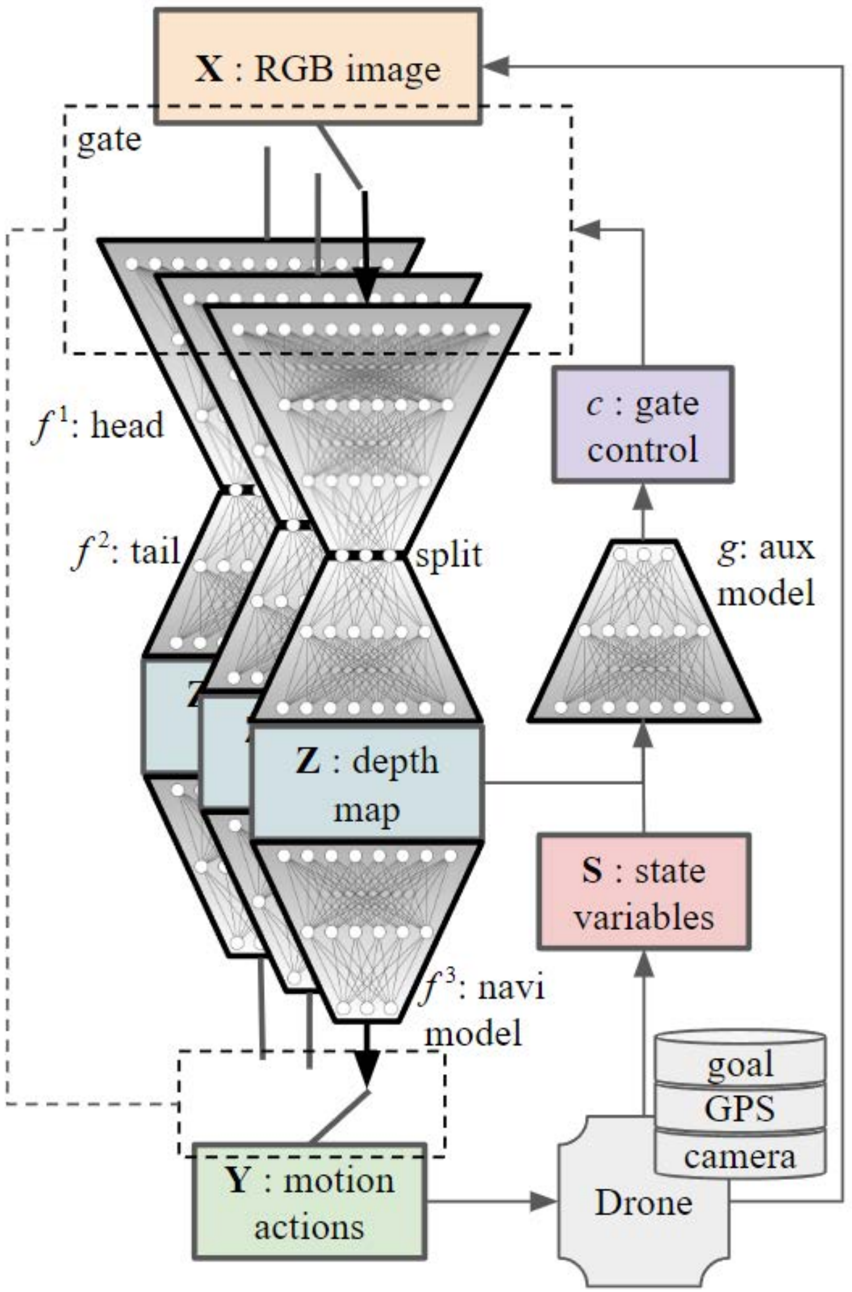


NaviSplit

Supervised training based on knowledge distillation to **split** the depth estimation model



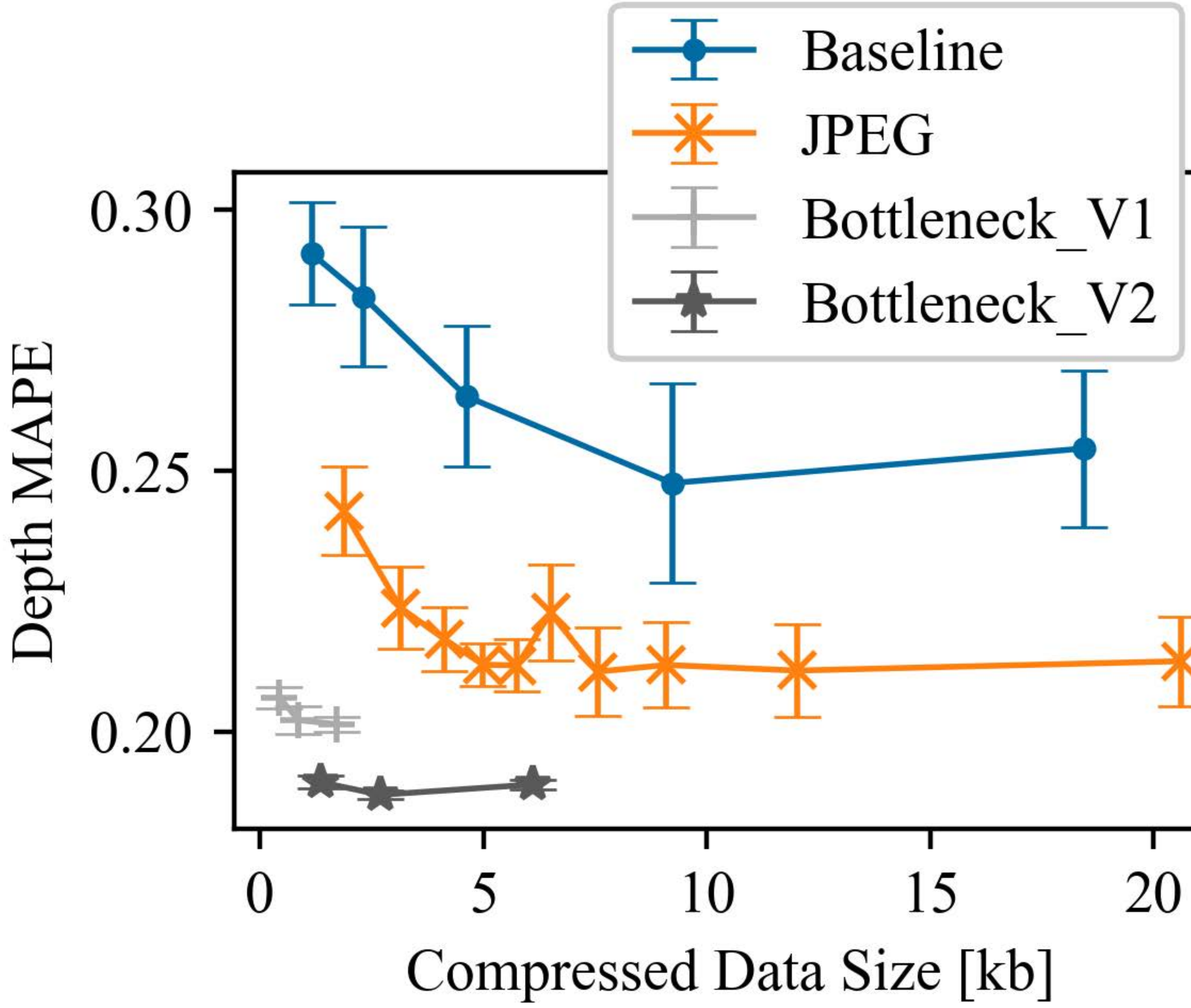
NaviSplit



Our solution:

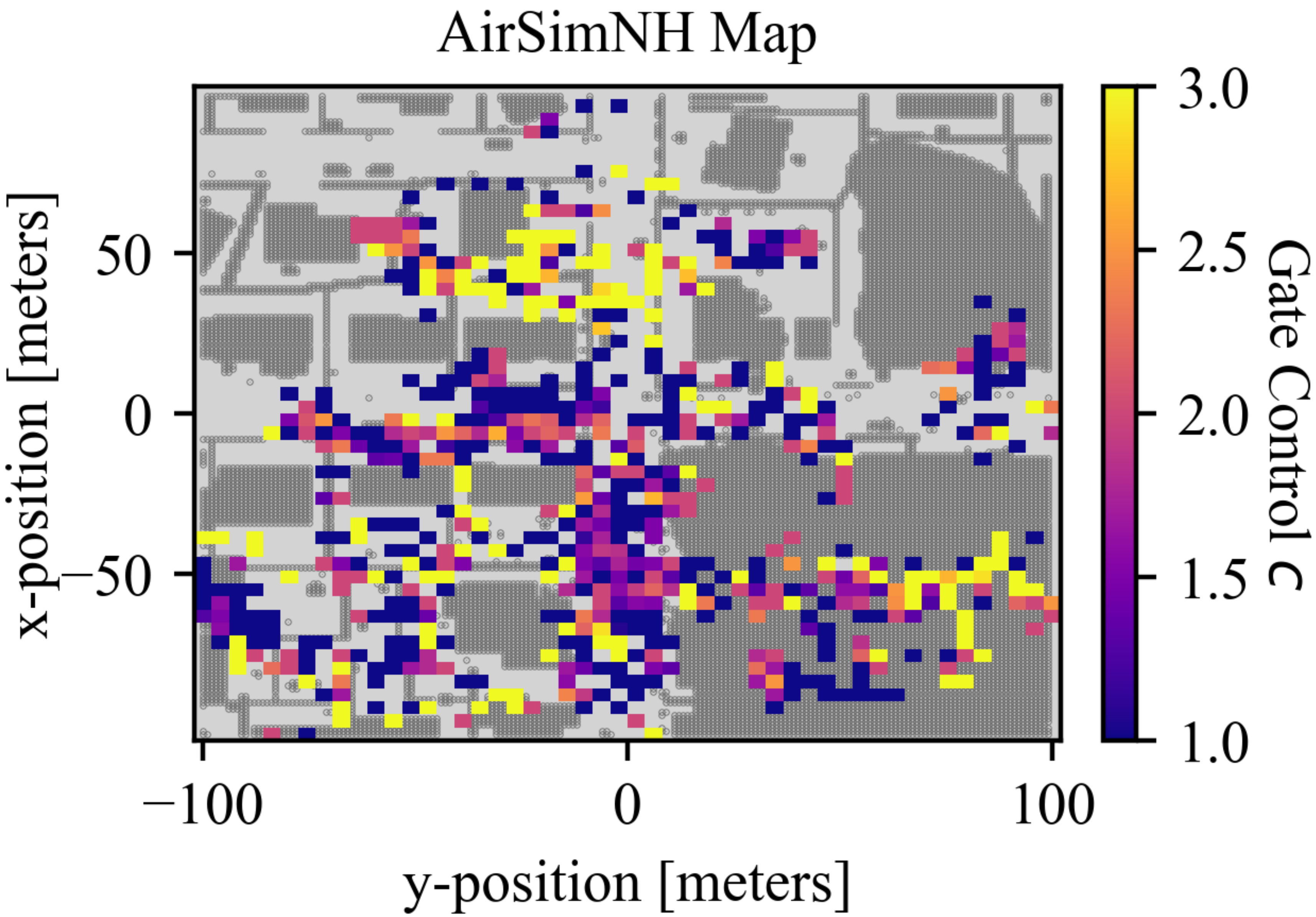
- **Split depth estimation:** depth model is split to minimize data transmission and computing effort at the drone
- **Navigation:** neural model transforms the depth map into motion commands
- **Adaptation:** auxiliary neural model takes mission parameters and depth map as input to determine the optimal representation

Performance



Mean absolute percent error vs data size

Performance



Adaptation

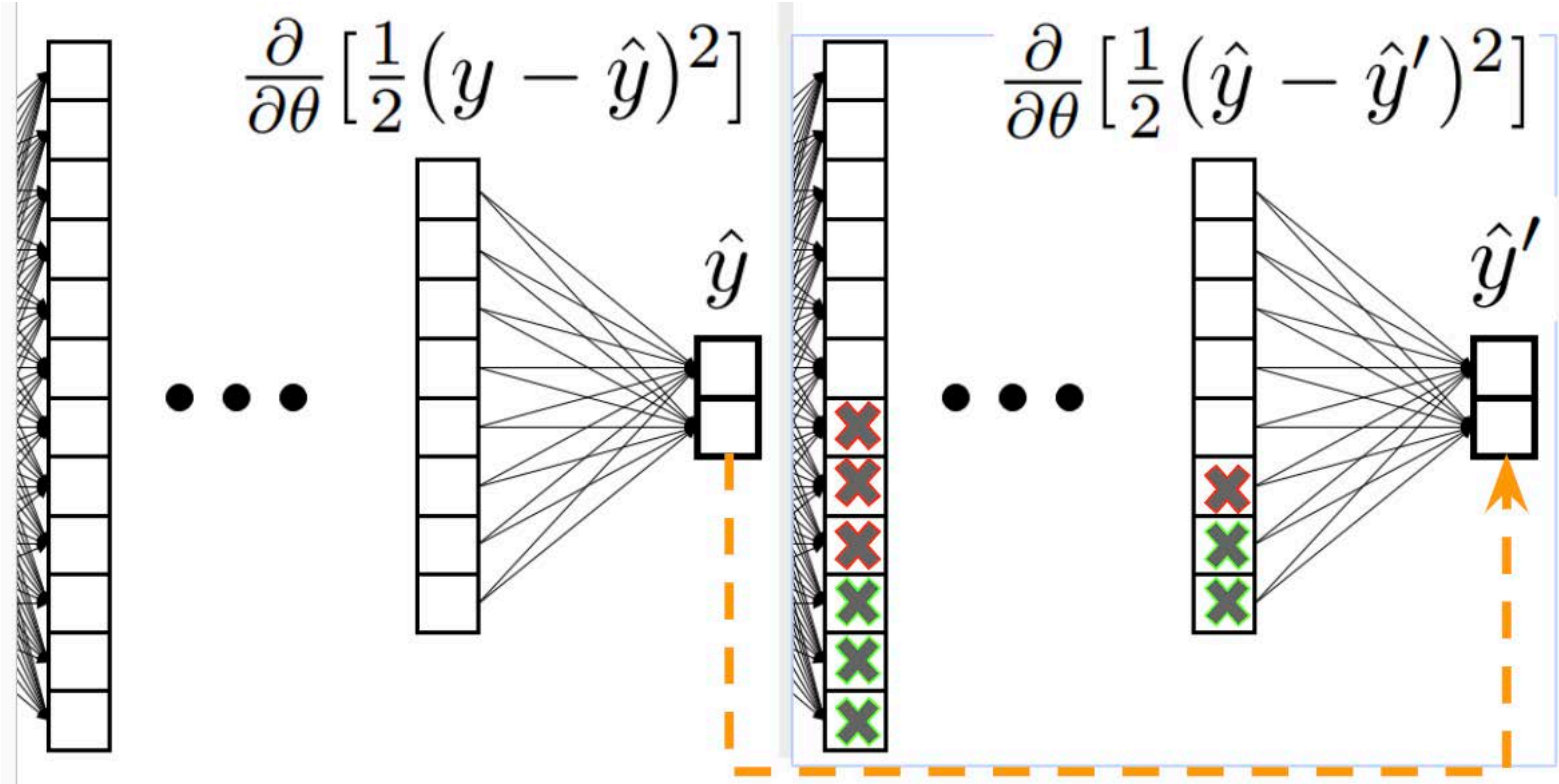
Self-Adaptive Low-Complexity AI

Slimmable Neural Networks

Networks whose width can be reduced at runtime

Super Network

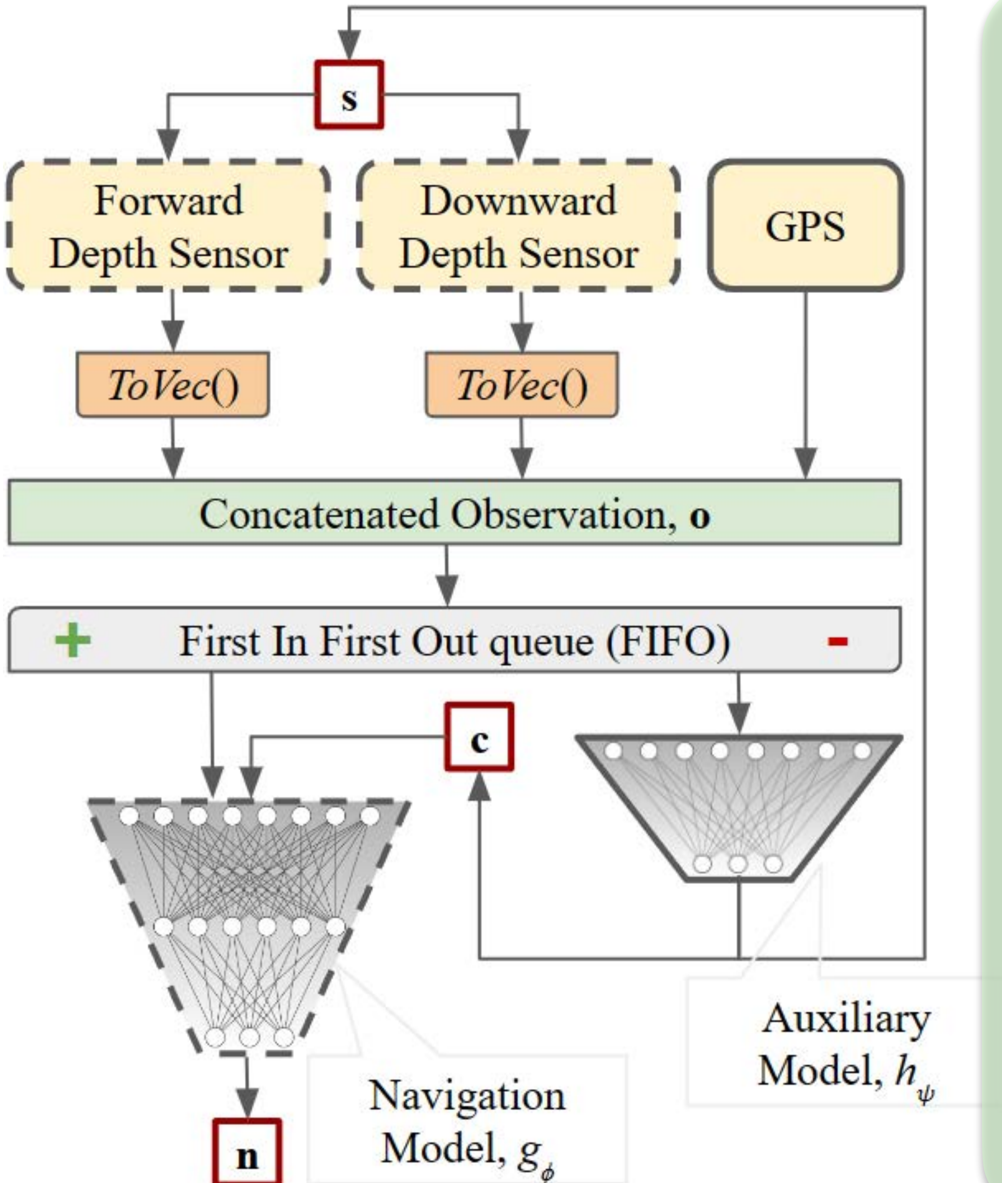
Sub Networks



Knowledge distillation:
sub networks learn to mimic the super network

Dynamic Neural Navigation for Microdrones

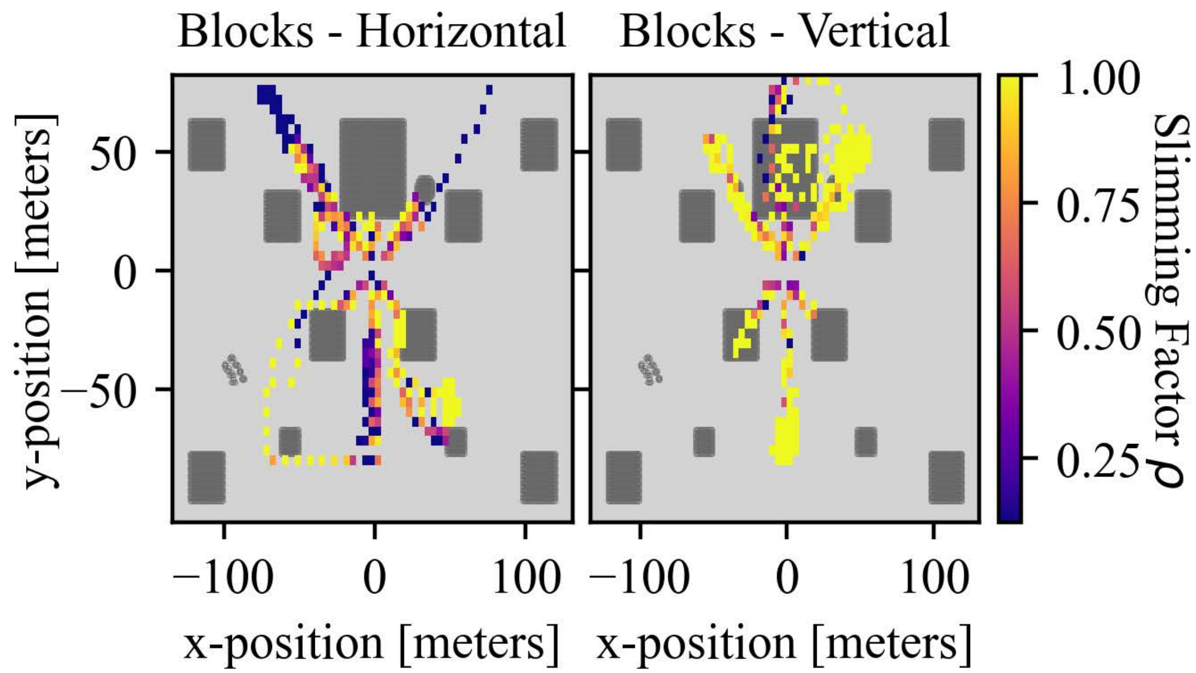
New architecture realizes a
gated dynamic slimmable network for navigation



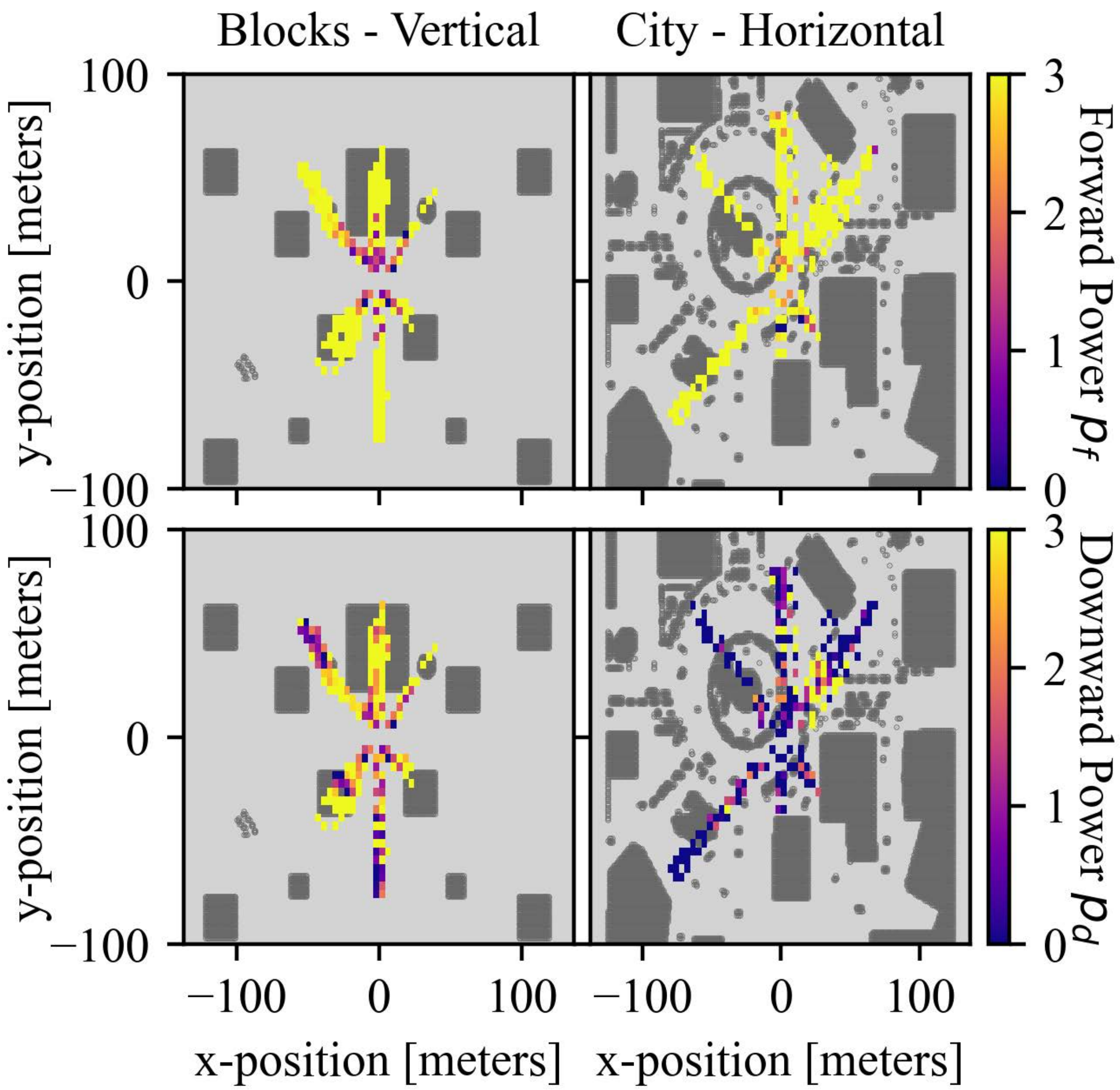
Auxiliary neural gate controls the slices of a main navigation model decision by decision
- Number of operations
- Sensor selection and resolution

Specialized multi-stage training uses
- Knowledge distillation
- Curriculum learning
- Deep reinforcement learning

Dynamic Adaptation

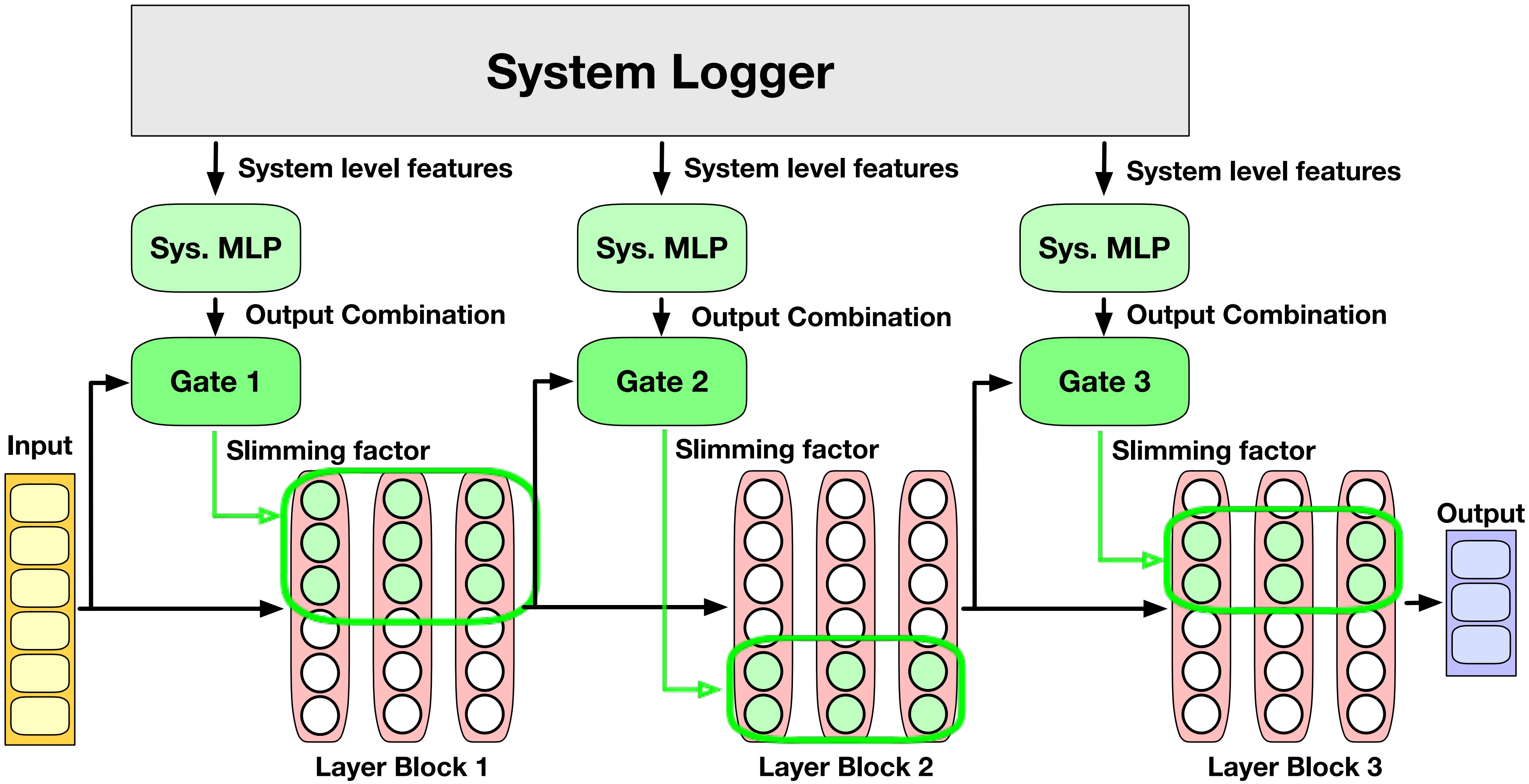


- Complexity slimming factors



- Sensing slimming factors

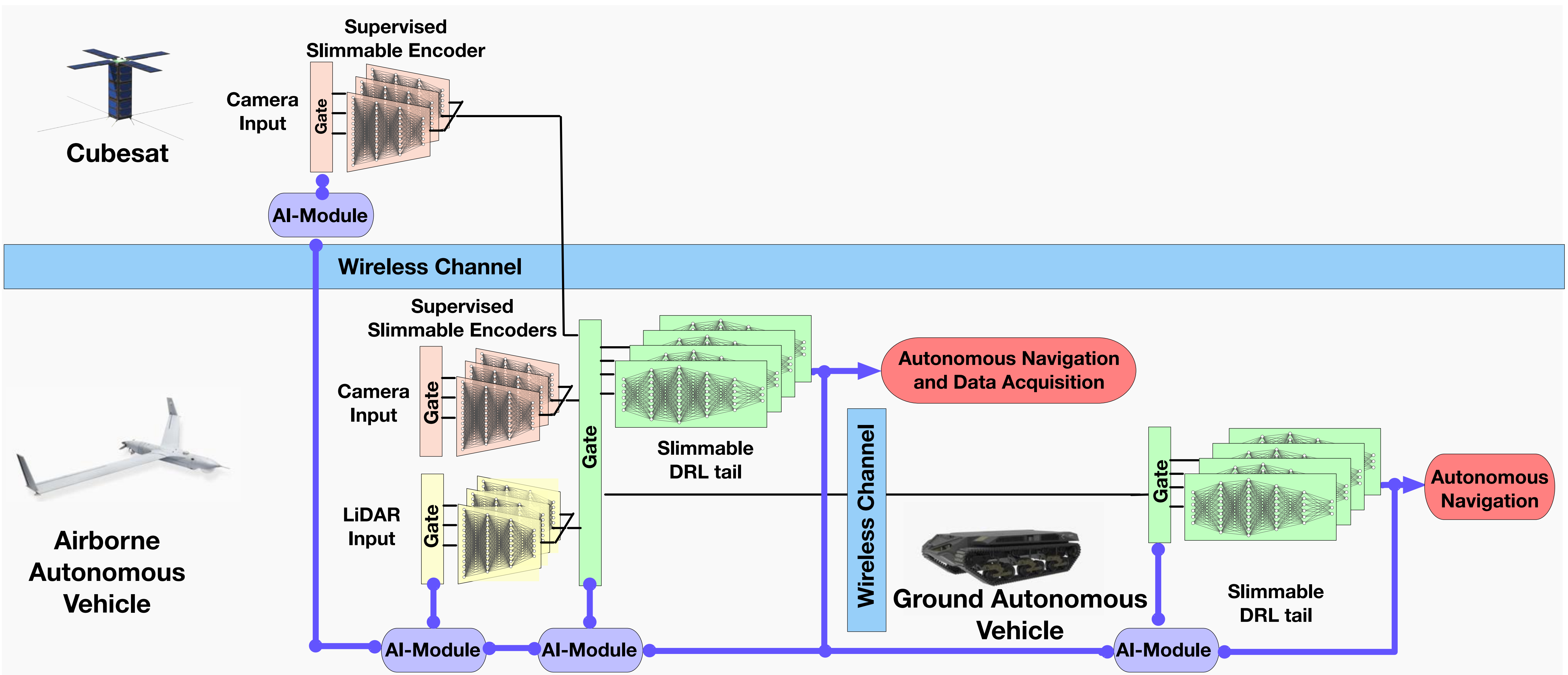
Dynamic Slimmable Networks



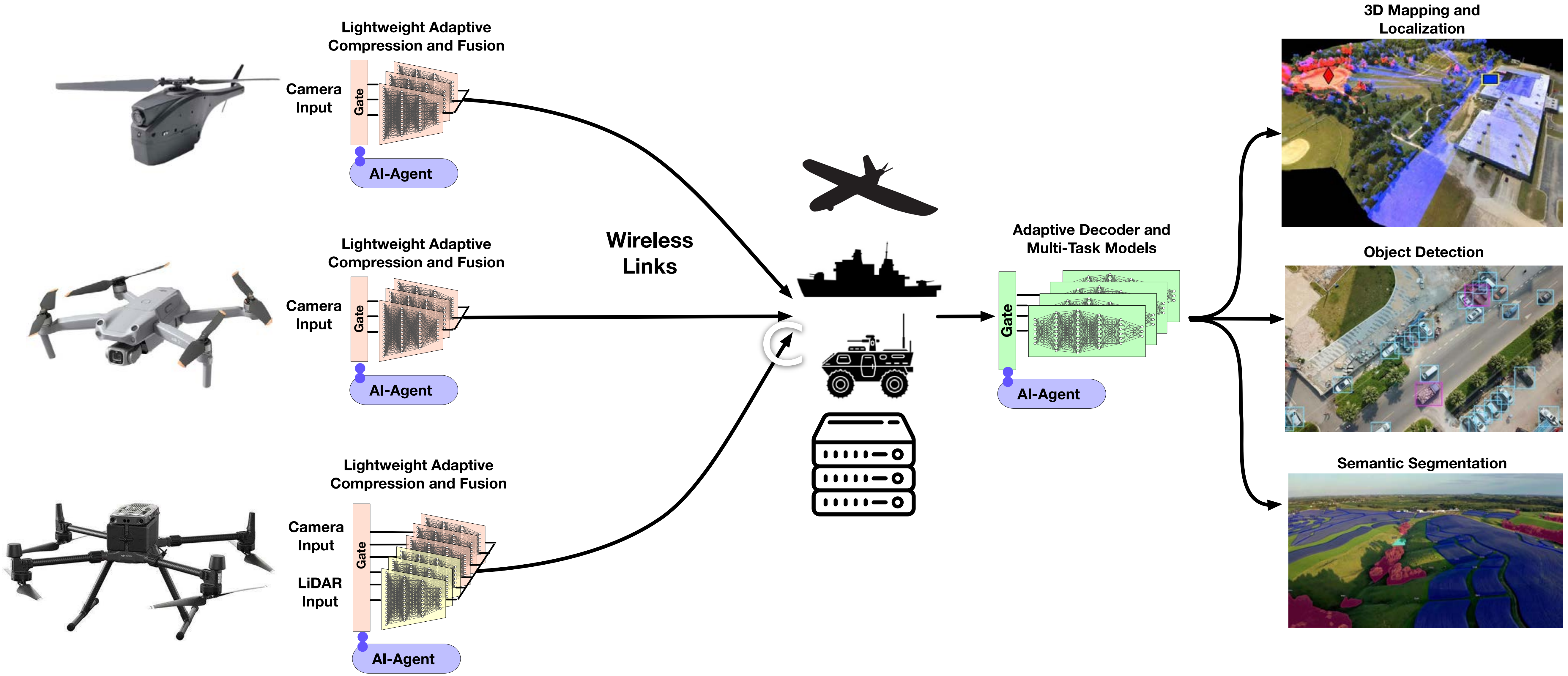
- Runtime adaptation
- Context AND system-aware
- Designed to be distributed (slimmable encoders are a component of it)

Vision

Layered Collaboration



FlexAI - Lightweight Swarm Intelligence



Dynamic Distributed Computing for Autonomous Vehicles in 5G Infrastructures

Marco Levorato
Professor
CS - University of California, Irvine
levorato@uci.edu