香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)
信息枢纽
INFORMATION HUB
人工智能学域
ARTIFICIAL INTELLIGENCE

Embodied AI System Overview

Course AIAA 4220

Week 1 - Lecture 2

Changhao Chen

Assistant Professor

HKUST (GZ)

What is "Embodiment"?

A Book vs. A Person

- **A book** *contains* information about how to ride a bike. It has diagrams, text, and physics equations. But the book itself cannot ride a bike. It is **disembodied** intelligence.

- **A person** *has* a body. They learn to ride by feeling the balance, scraping their knees, and coordinating their muscles. Their intelligence is **embodied**.

What is "Embodiment"?

We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried.

——<Computing Machinery and Intelligence> Alan Turing

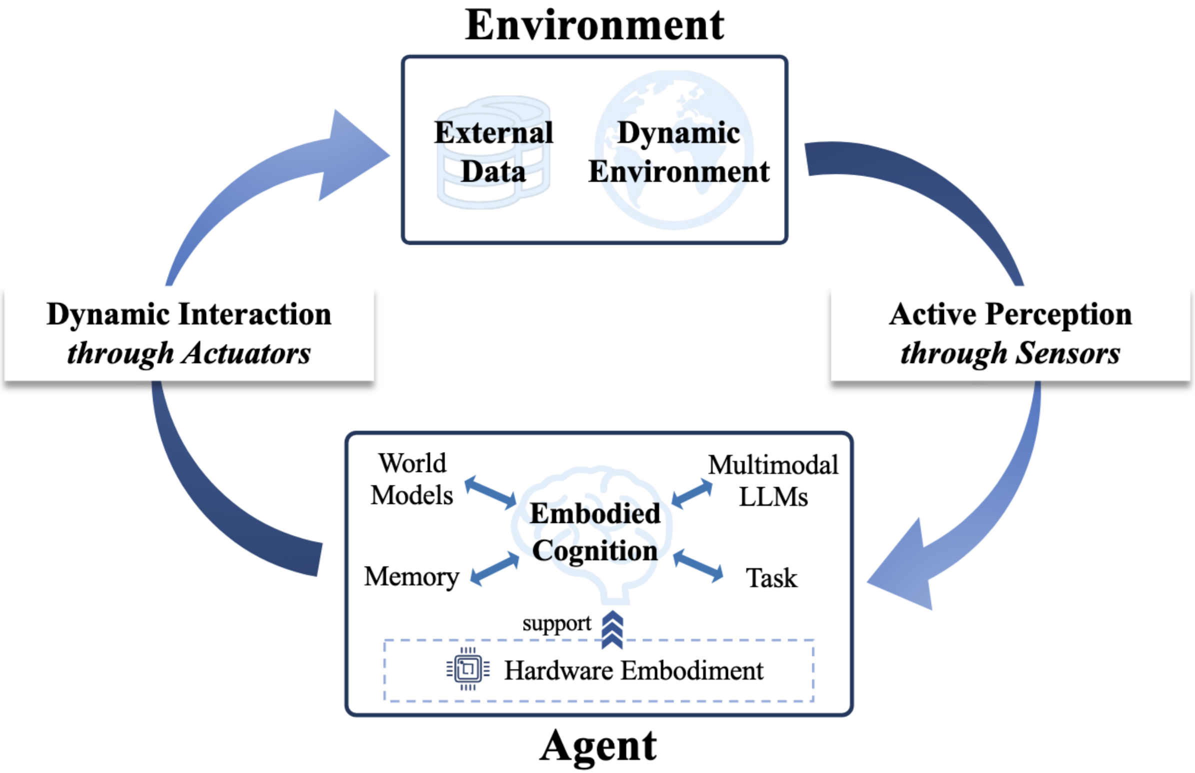
What is "Embodiment"?

Embodiment is the two-way relationship between an agent (an AI, a robot, an animal) and its environment, mediated through a physical form.

It’s not just having a body. It’s about how the body's characteristics (its shape, sensors, actuators, materials) shape:

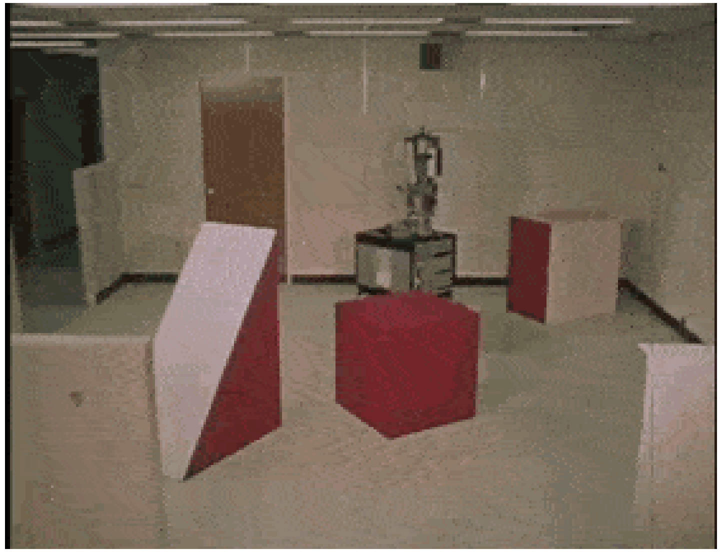
- How it perceives the world.
- What actions it can take.
- How it learns and thinks.

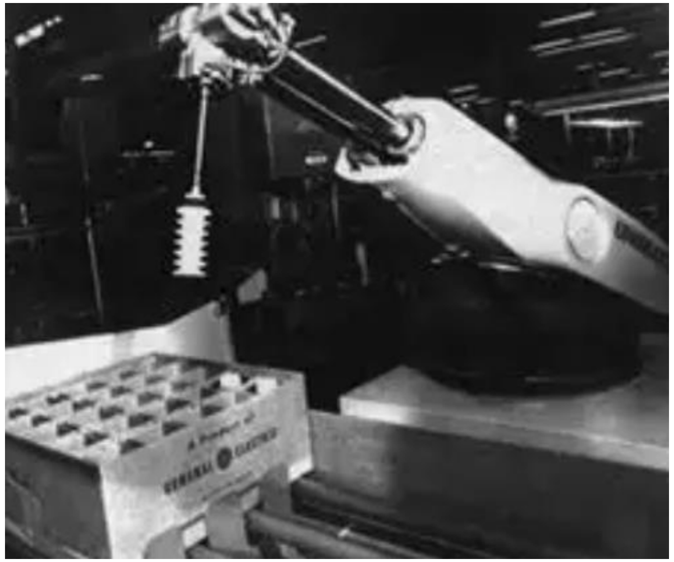
The body is not just a vessel for the brain; it is a fundamental part of the intelligence system.



History of Robotics (1950s - 1980s)

- The First Industrial Robot: Unimate (1954)
- Invented: By George Devol and Joseph Engelberger (the "Father of Robotics").
- Deployed: 1961 at a General Motors plant for die casting.
- Function: A programmable, hydraulic arm that lifted hot metal parts and stacked them. It was disembodied—it had no sensors and simply repeated pre-programmed motions.
- Widespread Adoption: Robotics became essential in automotive and manufacturing industries for tasks that are dull, dirty, or dangerous (e.g., welding, assembly, material handling).





History of Robotics (1980s - 2000s)

Sensor-Guided Reaction:
Sensors- Integration of vision systems, force/torque sensors, and lidar.
Computing- More powerful and smaller computers enabled real-time processing.

Robots moved from blind, pre-programmed machines to machines that could perceive and react to their environment (e.g., selecting parts from a conveyor belt using vision).

Examples:
PUMA (Programmable Universal Machine for Assembly): A seminal electric-arm robot used widely in research and industry.

First Autonomous Vehicles: Landmark projects like CMU's Navlab and Ernst Dickmanns' VaMoRs car demonstrated the potential of embodied AI on wheels.

History of Robotics (2000s - Present)

The Modern Era: Embodiment and Autonomy

The AI Revolution: The convergence of machine learning, massive computational power (GPUs), and advanced sensors.

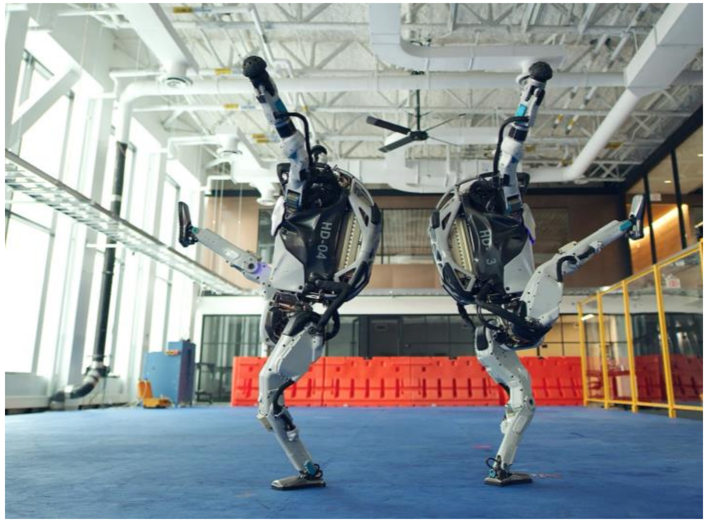
Beyond the Factory Floor: Robots moved into dynamic, unstructured environments:

Space: NASA's Mars Rovers (Spirit, Opportunity, Curiosity, Perseverance).

Consumer: iRobot's Roomba (2002) brought robotics into homes.

Research: Boston Dynamics pushed the boundaries of dynamic locomotion and mobility with BigDog, Atlas, and Spot.

AI Integration: The rise of Embodied AI, where robots learn tasks through simulation and reinforcement learning, connecting perception to action intelligently.



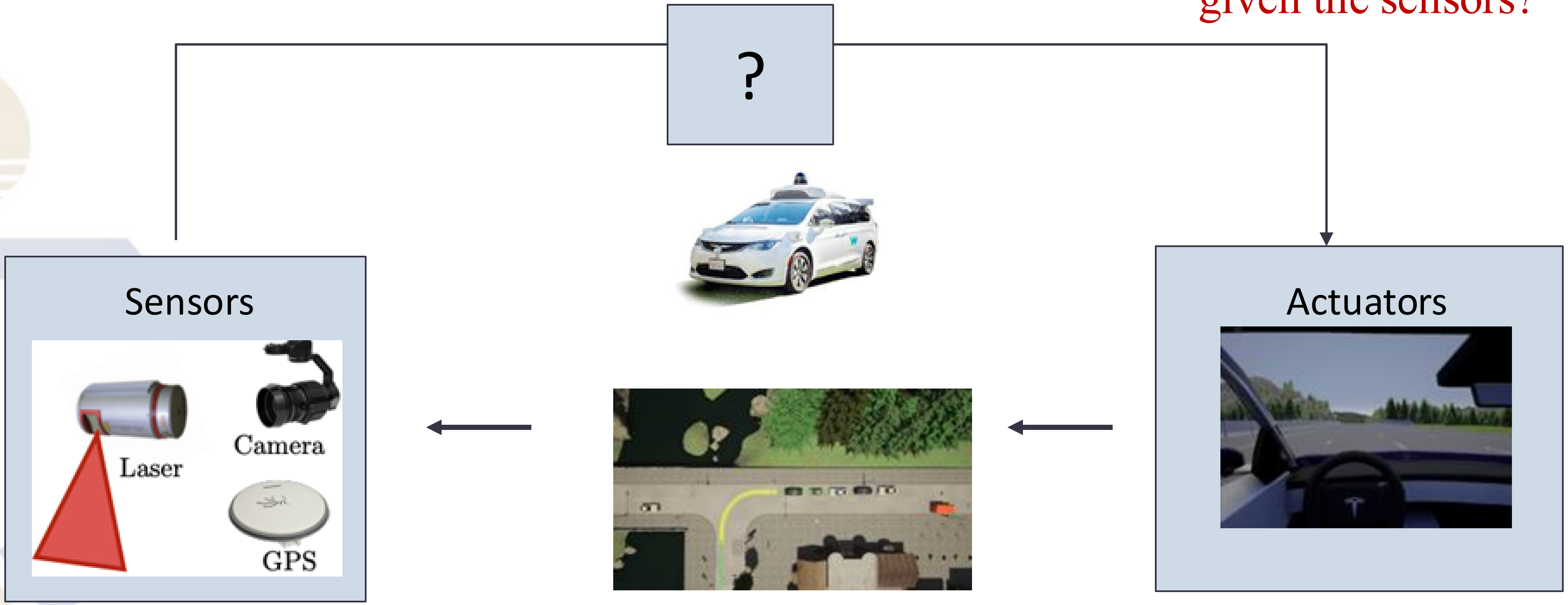
The Embodied AI Paradigm

- Let’s start with - How do we make a self-driving vehicle?
  - Given
    - Starting location, Goal location
    - Map of the city
    - Sensors - Camera, LIDAR, etc.
  - Objective
    - Minimize distance to goal
  - Constraint
    - Follow the road
    - Follow traffic rules
    - No crashing into other objects

The Embodied AI Paradigm

- How do we make a self-driving vehicle?
    - Actions includes steering wheel and pedals

- Key questions
    - Q1: Assume we know everything about the world. What commands should we send to the actuator?
    - Q2: How do we update the world knowledge given the sensors?



?

Sensors

Actuators

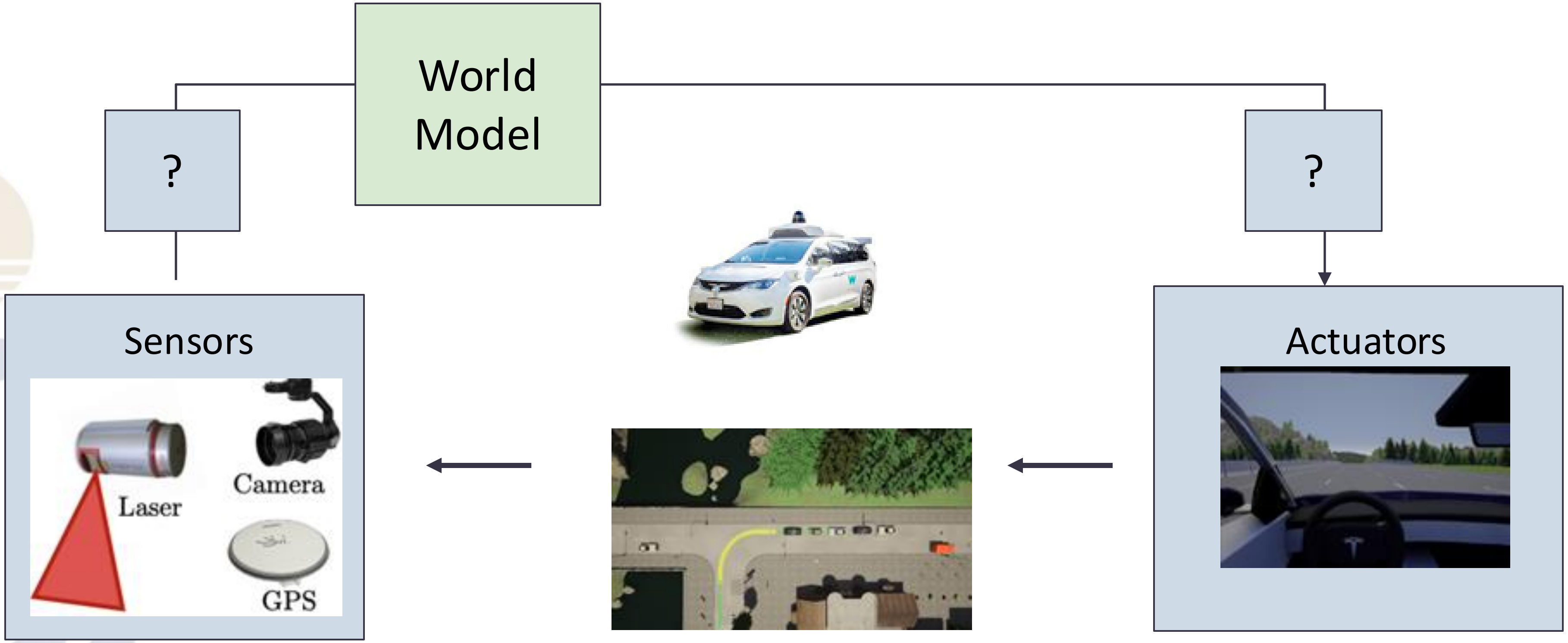
Robot interacts with environment

Steering wheel, pedals

The Embodied AI Paradigm

- Suppose we have a **world model**

Note in robotics, people use “model” to refer to a representation of the physical system. In ML, the model refers to an algorithm (and its parameters).



World Model

?

?

Sensors

Actuators

Robot interacts with environment

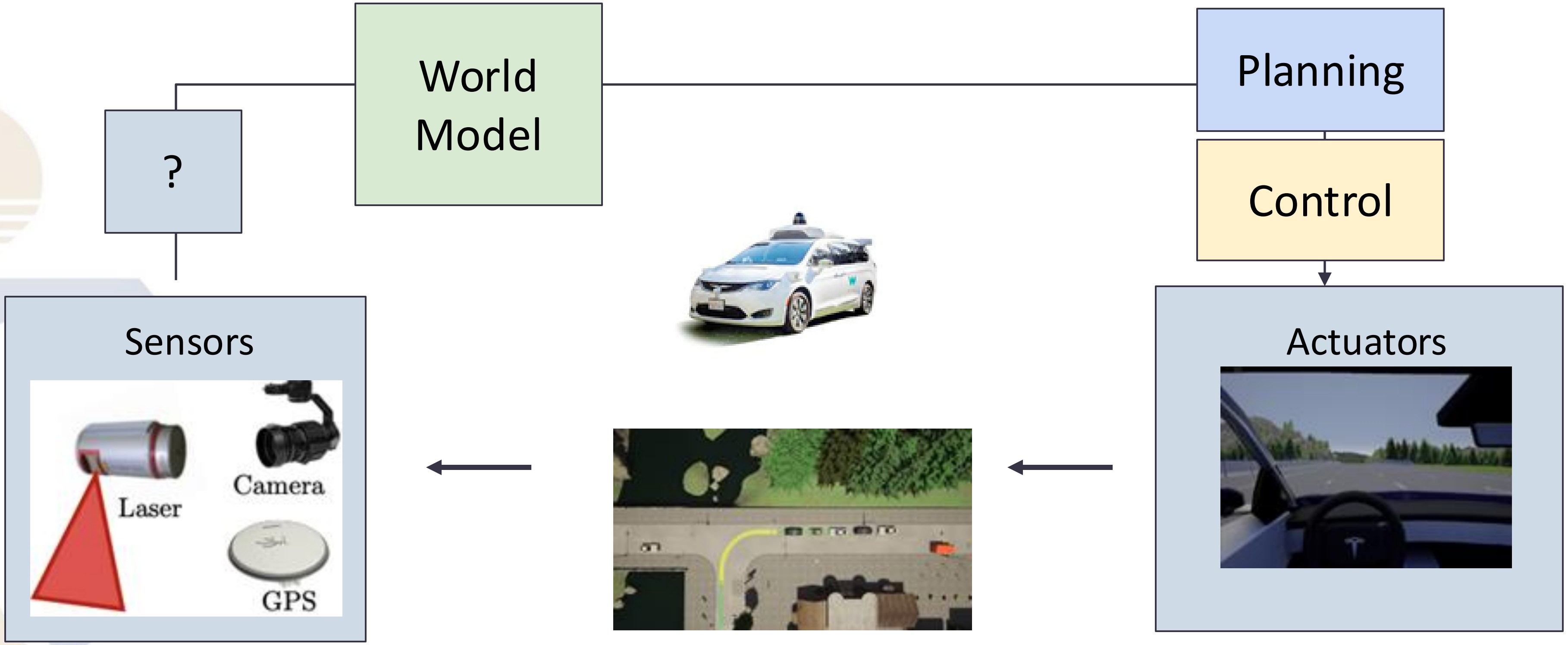
Steering wheel, pedals

The Embodied AI Paradigm

- What is the world model? It includes:
  - **Where is the robot/vehicle? What is its current status?**
  - What are obstacles (sidewalk, grass, trees) and what are available to drive (road)?
  - What objects to avoid (other vehicle, pedestrians)?
  - What are the properties of the vehicle? (Car Model)
    - How fast it can go? Acceleration rate?
    - Wet ground?
  - Nowadays, world model means “pixel and action prediction” model. So the vision generative models are considered world model

The Embodied AI Paradigm

- What is planning?
  - Planning is an optimization problem in which we find the optimal sequence of actions towards minimizing a cost function
  - While satisfying constraints (no crashing, follow traffic rules)
  - There are global planning (global route) and local planning (change lane? Turn?)



World Model

?

Planning

Control

Sensors

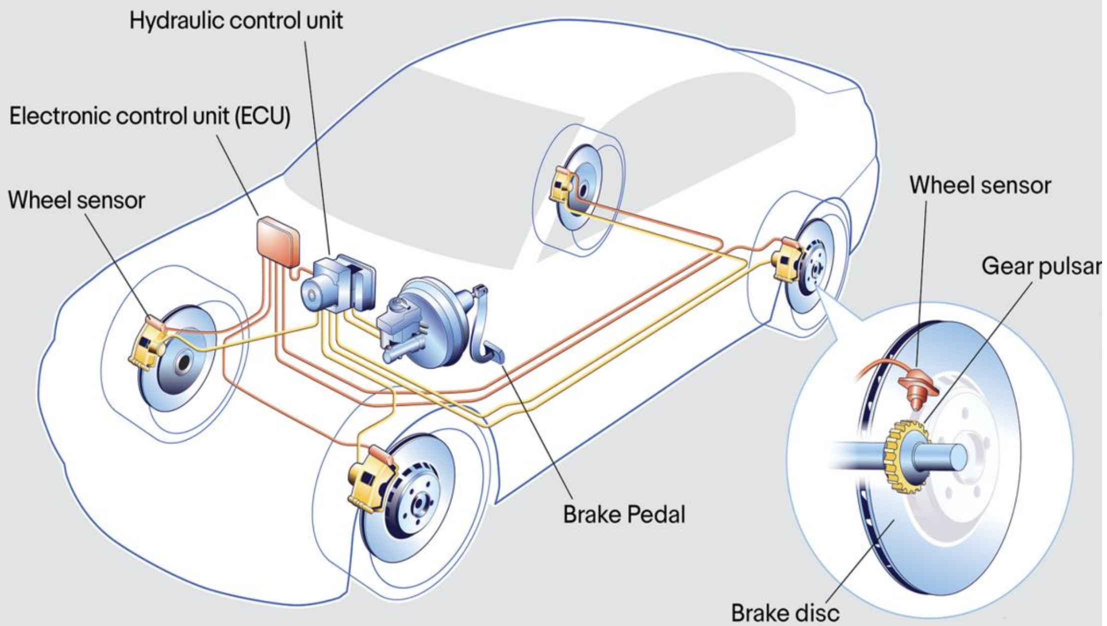
Actuators

Robot interacts with environment

Steering wheel, pedals

The Embodied AI Paradigm

- What is planning and **control**?
  - There are global (global route) and local planning (change lane? Turn?)
  - **A controller will execute the plan by giving specific commands to the actuators**
    - For example, convert the trajectory/motion plan (which way the car should go and how fast) into the **steering** and **pedal** command

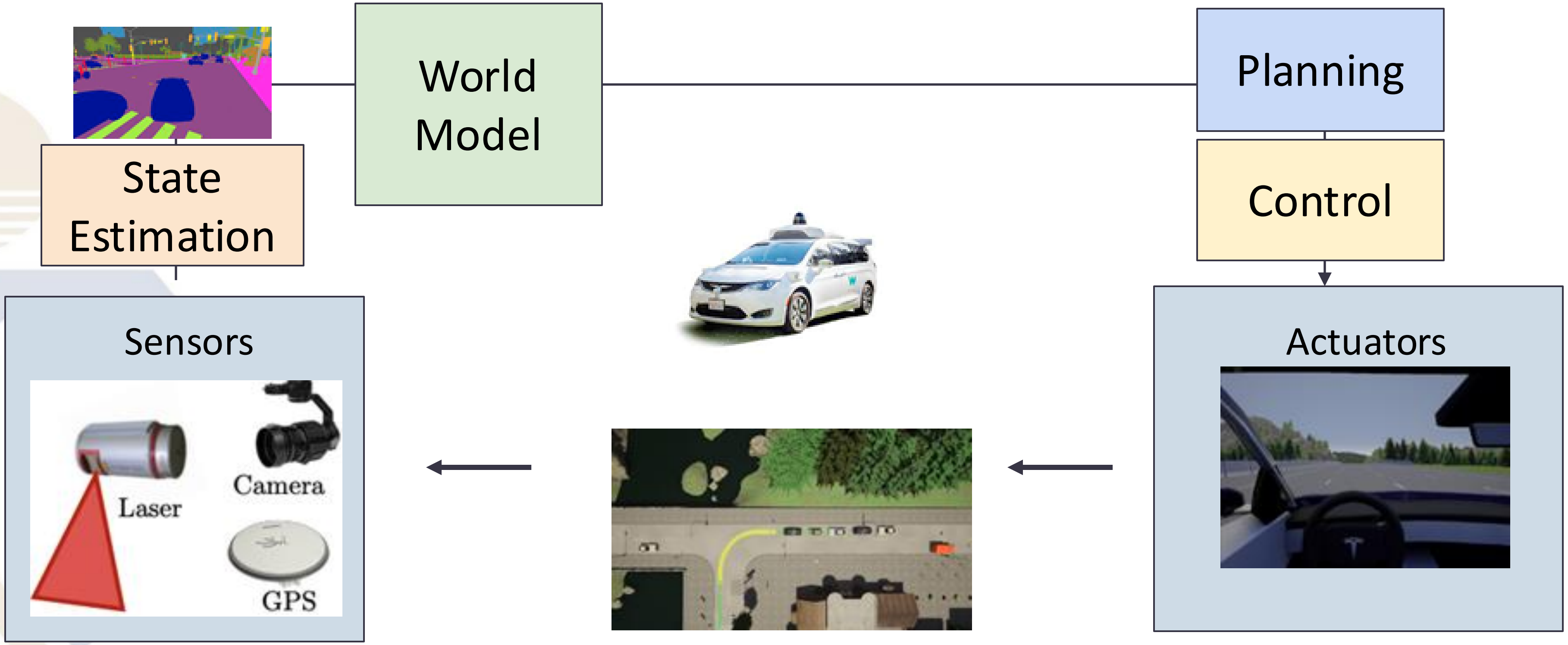


The Embodied AI Paradigm

- Key questions
  - Q1: Assume we know everything about the world. What commands should we send to the actuator?
    - Done. We know how to control the car if we know about the world
  - **Q2: How do we update the world knowledge given the sensors?**

The Embodied AI Paradigm

- **State Estimation -** Update the world knowledge given the sensors
  - Given the sensor data, the perception module
    - extracts necessary information (about the surroundings)
    - localizes the robot in the world



World Model

Planning

State Estimation

Control

Sensors

Actuators

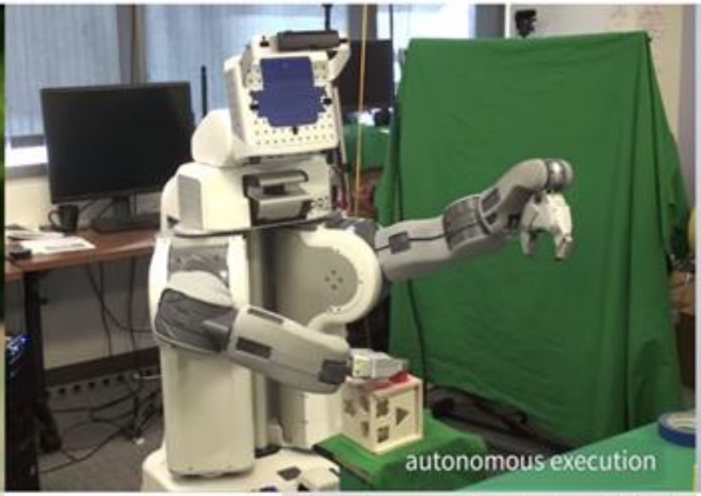
Robot interacts with environment

Steering wheel, pedals

The Embodied AI Paradigm

- Besides self-driving vehicle, we can also generalize this paradigm to other robot systems
  - Drones, robotic arms, legged robots

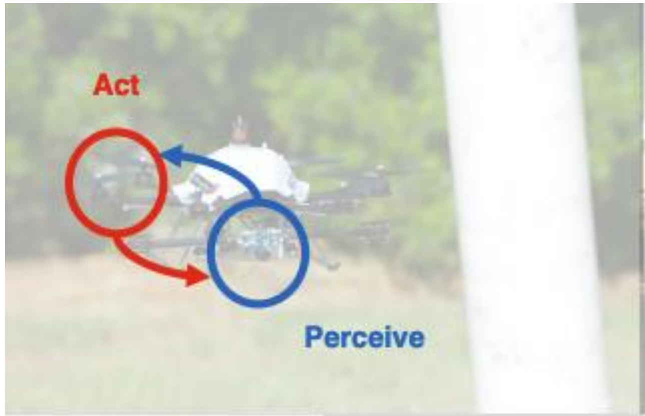
[Sa et al. IROS 2014]



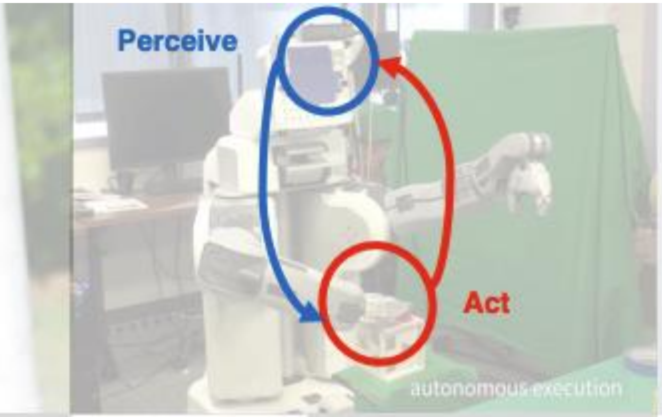
[Levine et al. JMLR 2016]

[Bohg et al. ICRA 2018]

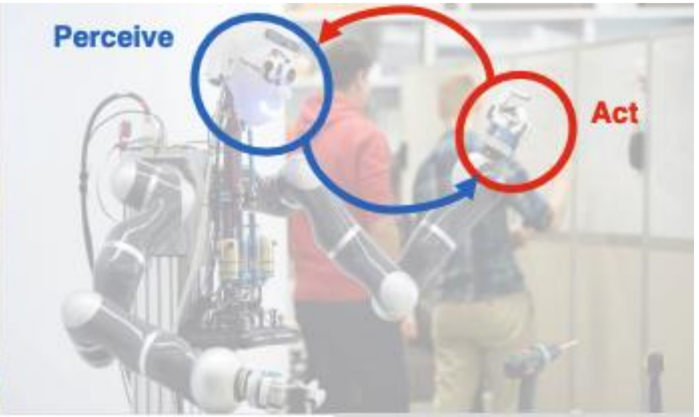
The Embodied AI Paradigm



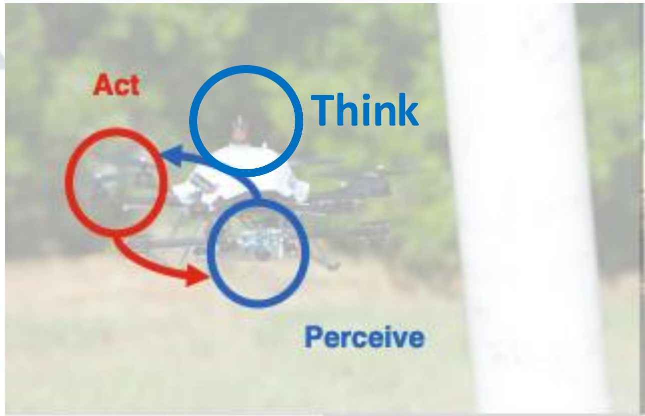
[Sa et al. IROS 2014]



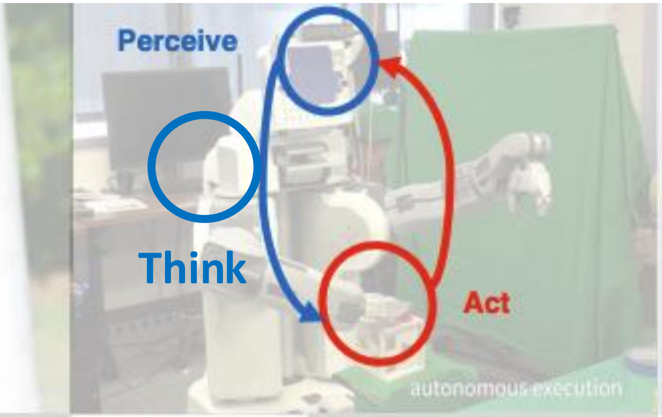
[Levine et al. JMLR 2016]



[Bohg et al. ICRA 2018]



[Sa et al. IROS 2014]

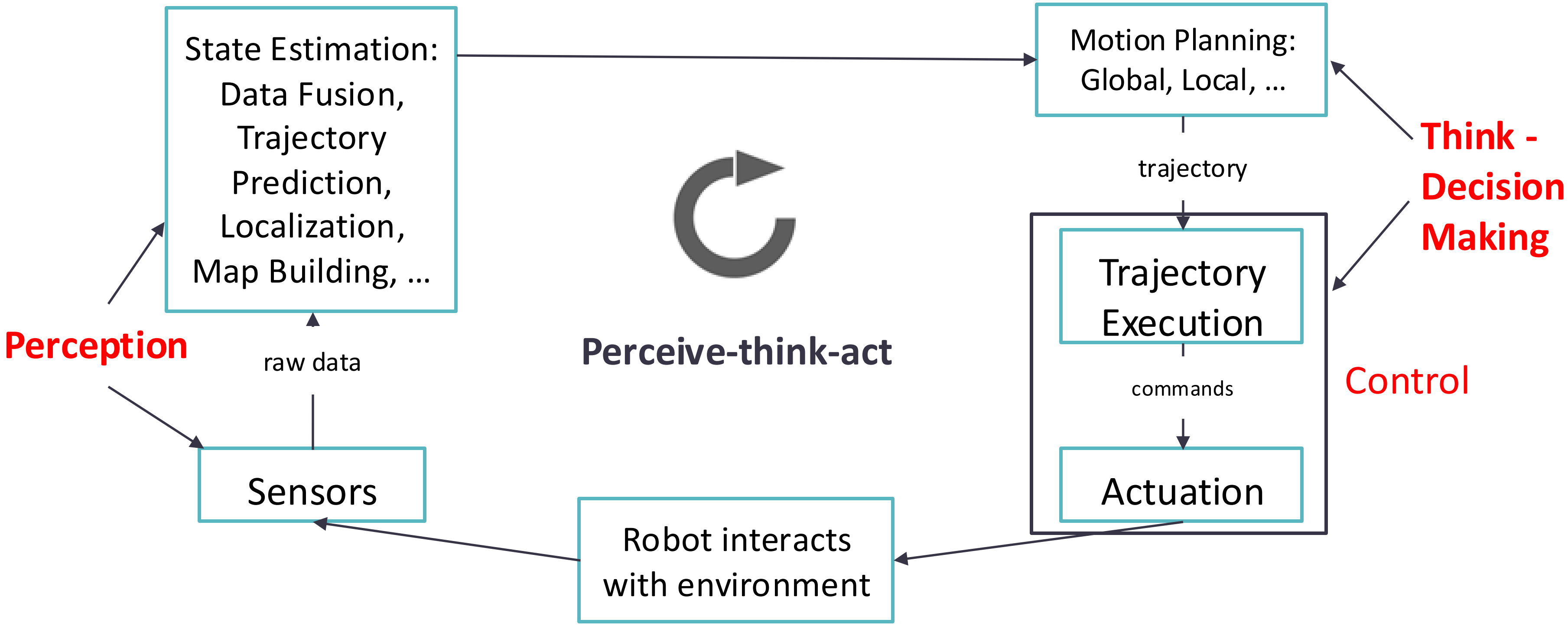


[Levine et al. JMLR 2016]

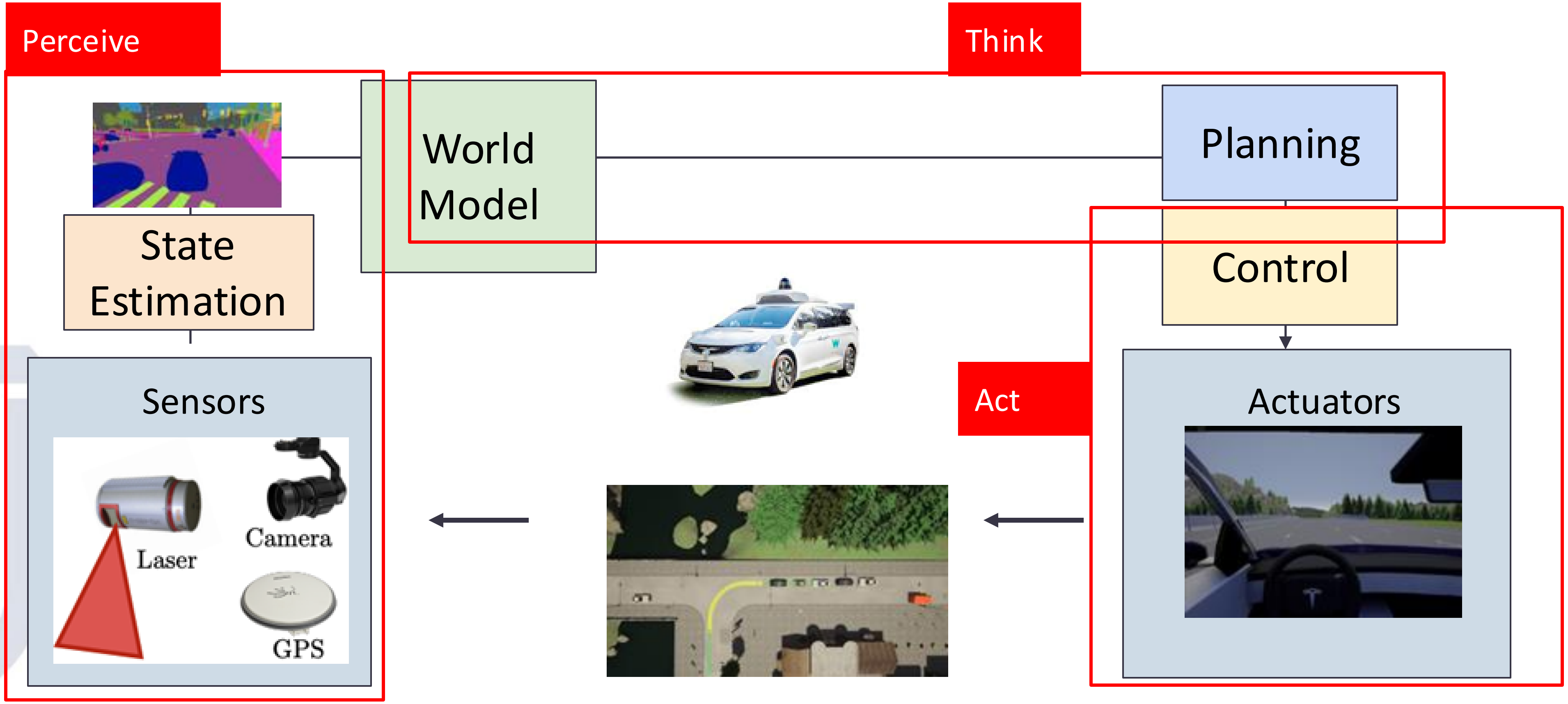
[Bohg et al. ICRA 2018]

The Embodied AI Paradigm

- A more generalized paradigm - The Perceive-think-act circle


State Estimation: Data Fusion, Trajectory Prediction, Localization, Map Building, …
Motion Planning: Global, Local, …
trajectory
Think - Decision Making
Trajectory Execution
commands
Actuation
Control
Perception
raw data
Sensors
Perceive-think-act
Robot interacts with environment

The Perceive-Think-Act Cycle

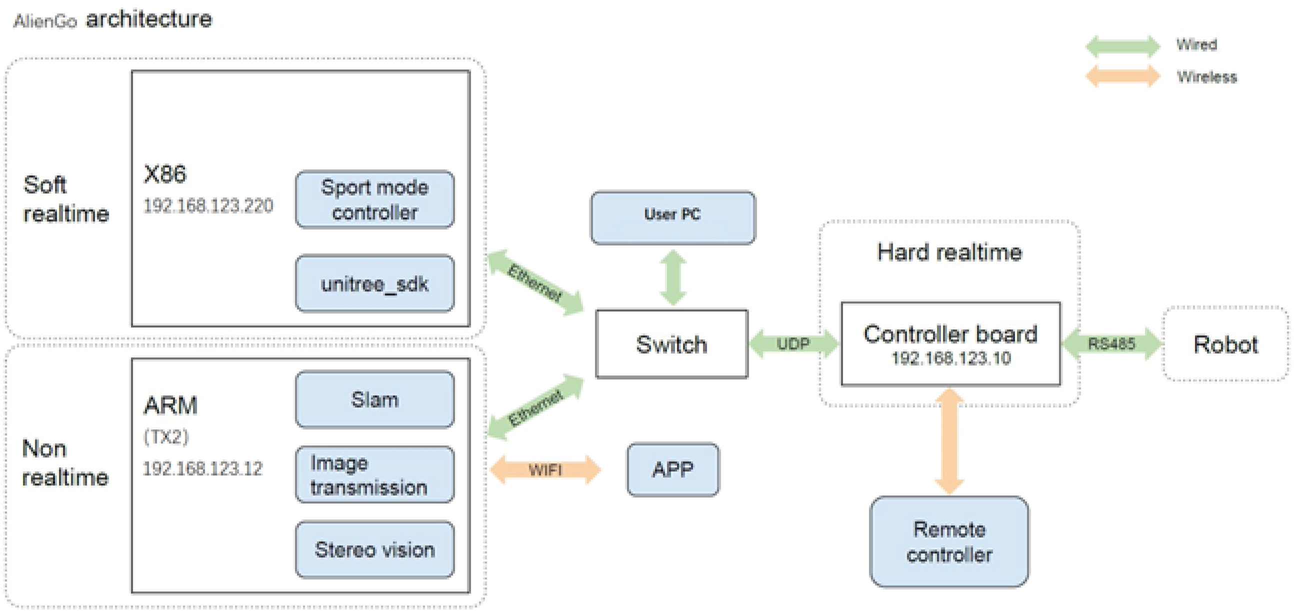


Robot interacts with environment

Steering wheel, pedals

Embodied System

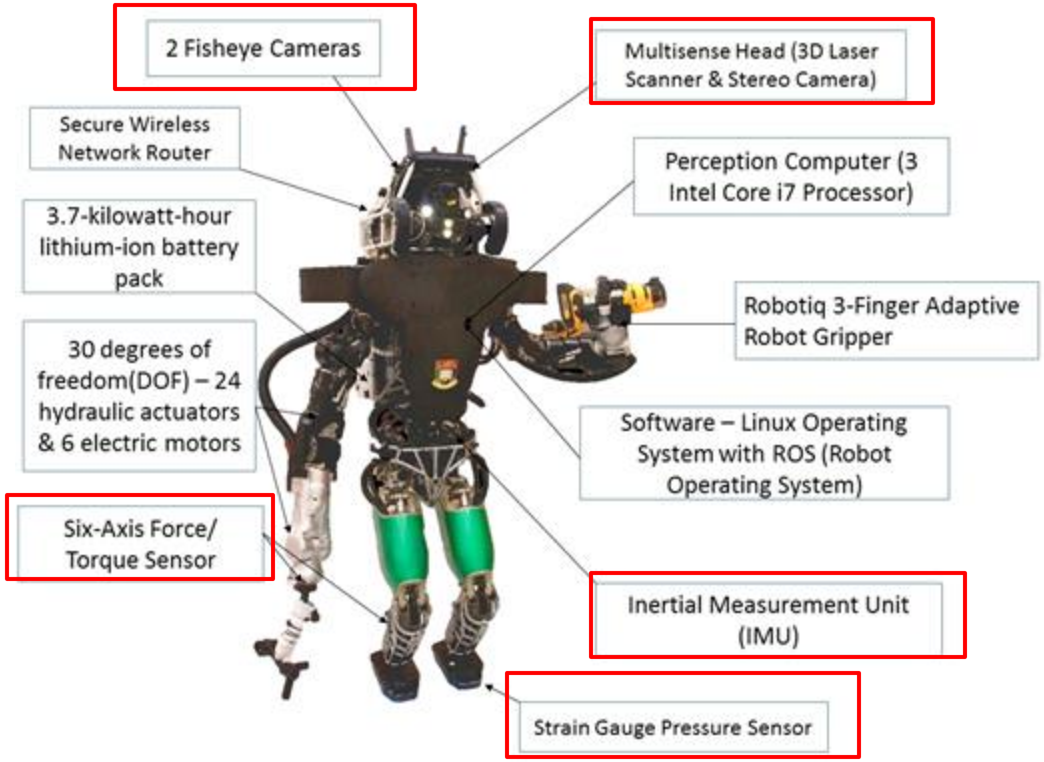
- The hardware
  - Sensors
  - Actuators
- The software
  - Lowlevel: Control Unit Drivers
  - Middleware: Robot Operating System (ROS)
    - Communication between modules
    - Data logging
    - Visualization
  - Perception models
  - Motion planning models
- Principles
  - Follow the Perceive-think-act paradigm
  - Modular, robust, data logging and playback



The architecture of Unitree's four legged-robot

Embodied System

- Multimodal sensors are key to robot perception
  - Example: DARPA robotic challenge 2015
    - Fisheye cameras
    - LIDAR
    - Inertial Measurement Unit (IMU)
    - Torque Sensor
    - Pressure Sensor



2 Fisheye Cameras
Multisense Head (3D Laser Scanner & Stereo Camera)
Secure Wireless Network Router
Perception Computer (3 Intel Core i7 Processor)
3.7-kilowatt-hour lithium-ion battery pack
Robotiq 3-Finger Adaptive Robot Gripper
30 degrees of freedom(DOF) – 24 hydraulic actuators & 6 electric motors
Software – Linux Operating System with ROS (Robot Operating System)
Six-Axis Force/ Torque Sensor
Inertial Measurement Unit (IMU)
Strain Gauge Pressure Sensor

[Source: HKU Advanced Robotics Laboratory]

Key Techniques - Sensors

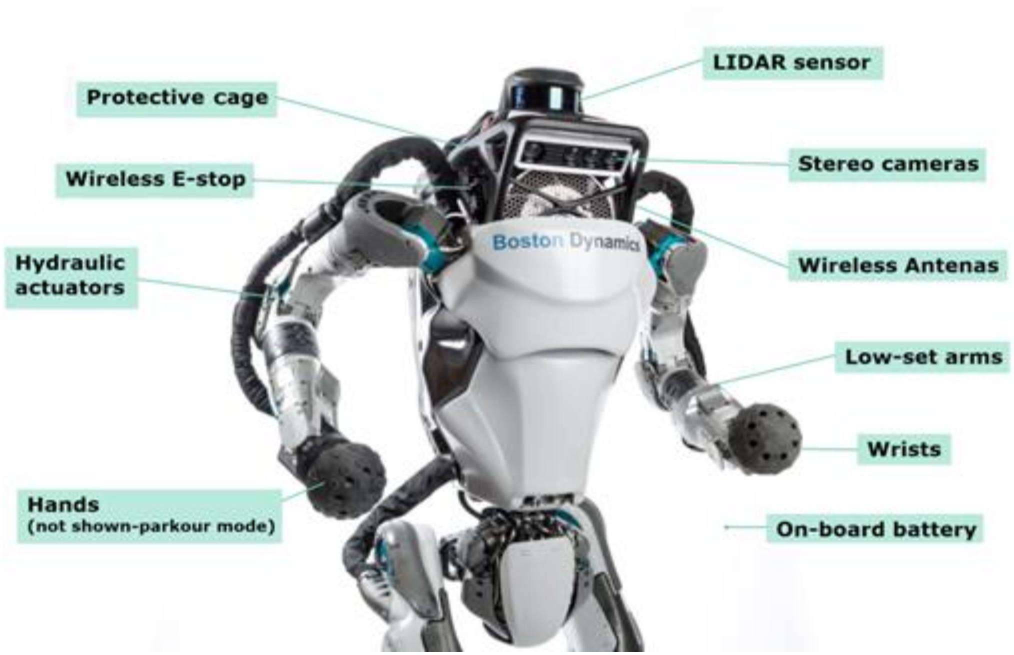
Sensors are the hardware that collects raw data about the environment. Each type has unique strengths and weaknesses, making them complementary.

Cameras:

Provide rich semantic information (color, texture, text). They are essential for reading traffic signs, recognizing traffic light colors, detecting lane markings, and classifying objects (e.g., pedestriar vs. cyclist).

Challenge:

They produce 2D images, making it difficult to accurately judge distance and speed, and their performance degrades in poor lighting (darkness, glare, fog).



Key Techniques - Sensors

Monocular Camera:

Single lens, low-cost

Rich information

Limitation: lacks depth perception

Stereo Camera:

Mimics human eyes → depth from disparity

Useful for near-field obstacle detection

More expensive & calibration-sensitive

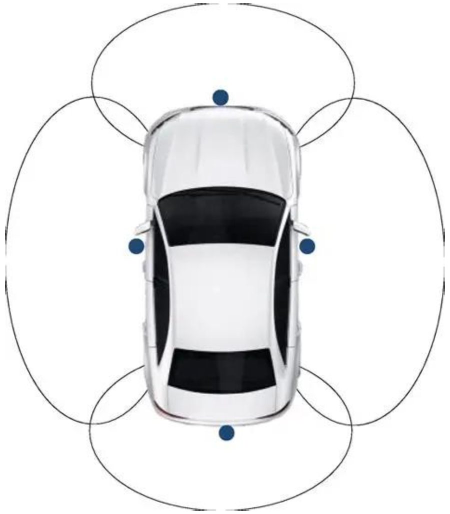
Key Techniques - Sensors

Fisheye & Surround-View Cameras:

Wide field of view (180°)

Typically used for parking, blind-spot monitoring

Enable bird’s-eye surround-view systems

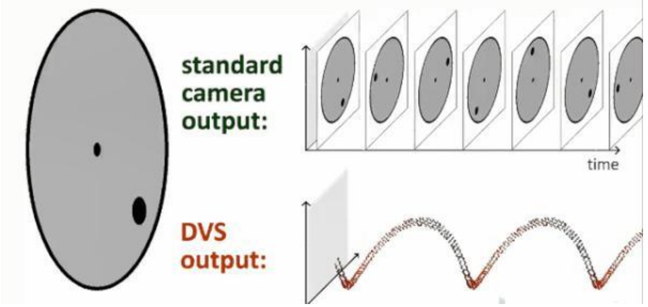


Thermal & Event Cameras:

Thermal cameras: robust to low light, fog, night driving

Event cameras: capture fast motion with low latency

Emerging but not yet mainstream



Key Techniques - Sensors

GPS (Global Positioning System):

Primary Role: Provides absolute global positioning (latitude, longitude, and altitude).

Key Strength: It gives a vehicle its precise location on a global map, which is essential for determining the overall route from origin to destination.

Critical Weakness: Its signal is unreliable in urban canyons, tunnels, and dense cities. Tall buildings can block, reflect, or multipath signals, leading to inaccurate or lost positioning. It also updates at a slow rate (~1-10 Hz), making it insufficient for real-time vehicle control.

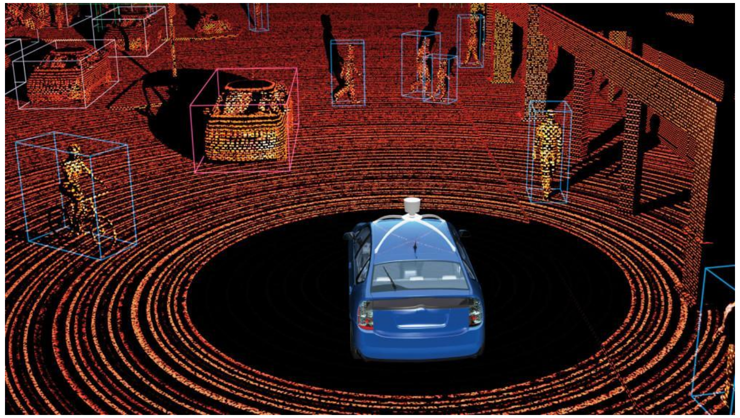
Key Techniques - Sensors

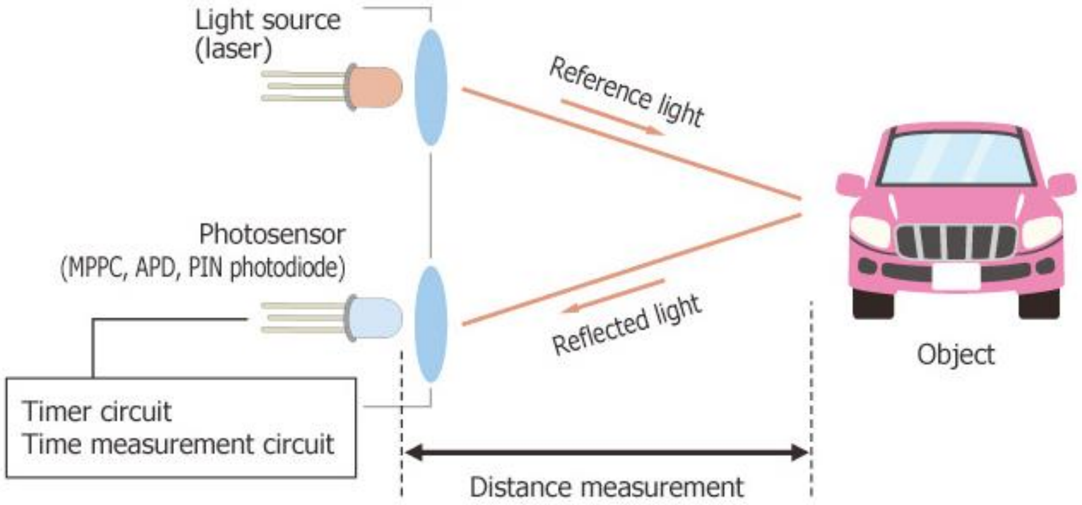
LiDAR (Light Detection and Ranging):

Primary Role: Creates a high-resolution 3D point cloud map of the immediate environment by measuring the distance to objects with laser pulse

Key Strength: Provides extremely accurate geomet information. It is excellent for understanding the precise shape, size, and distance of obstacles, othe vehicles, and road contours. It works well in the dark.

Critical Weakness: Historically expensive and can be bulky. Performance can be degraded by heavy rain, fog, or snow, which scatter the laser beams. Lower-resolution LiDARs may struggle with fine details..





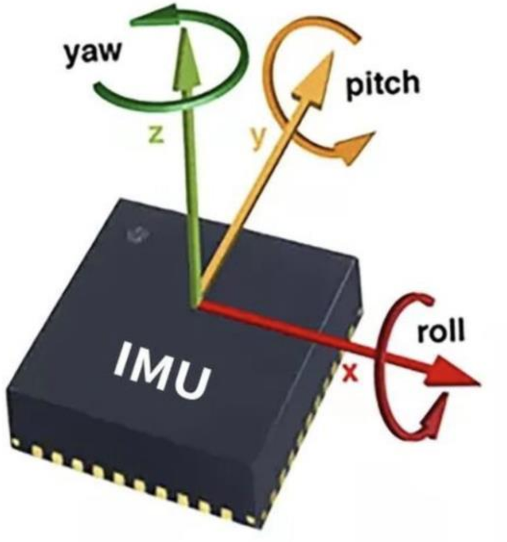
Key Techniques - Sensors

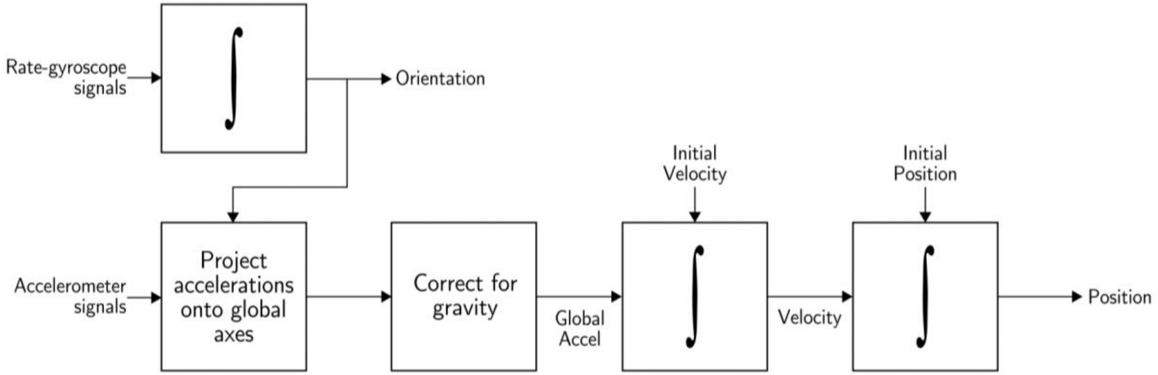
IMU (Inertial Measurement Unit):

Measures the vehicle's short-term motion and dynamics.

Key Strength: It provides high-frequency data (~100-1000 Hz) on acceleration and rotational rate. This is crucial for tracking the vehicle's movement between updates from slower sensors like GPS or cameras (e.g., during quick turns or sudden braking). It is not affected by external environmental conditions like weather or lighting.

Critical Weakness: Its estimates drift over time. Any small error in measuring acceleration or rotation is integrated into the position estimate, leading to exponentially growing inaccuracies if not corrected by other sensors.





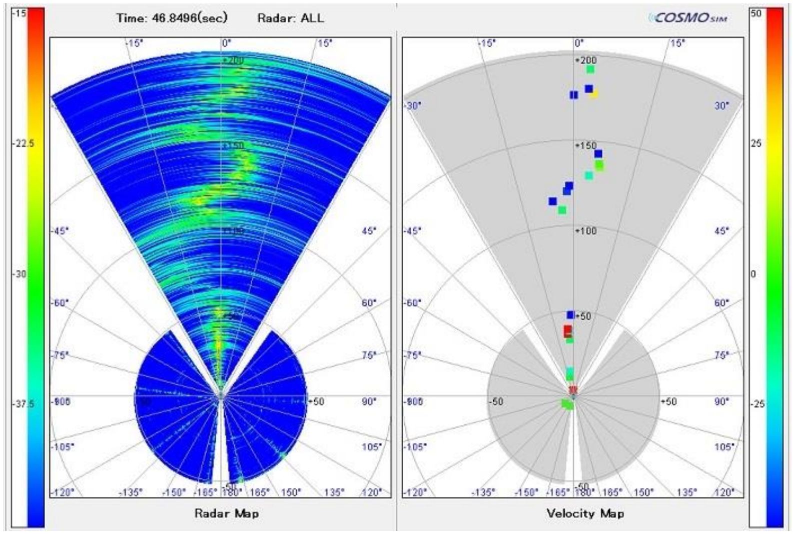
Key Techniques - Sensors

Radar (Radio Detection and Ranging):

Detects objects and precisely measures their relative distance and velocity.

Key Strength: Highly robust in adverse weather conditions like rain, fog, and snow. It directly measures the speed of approaching objects using the Doppler effect, making it exceptional for adaptive cruise control and collision warning.

Critical Weakness: Offers very low resolution compared to LiDAR and cameras. It typically struggles to discern the exact shape or type of an object (e.g., distinguishing a car from a guardrail) and can have difficulty with stationary objects.



Key Techniques – Computer Vision

Turning raw pixels into understanding:

- Humans drive mainly through vision
- Cameras replicate human perception for vehicles
- Enables understanding of lanes, obstacles, traffic signs

Object Detection and Recognition

Semantic Segmentation

Visual Localization and Mapping

Depth Estimation

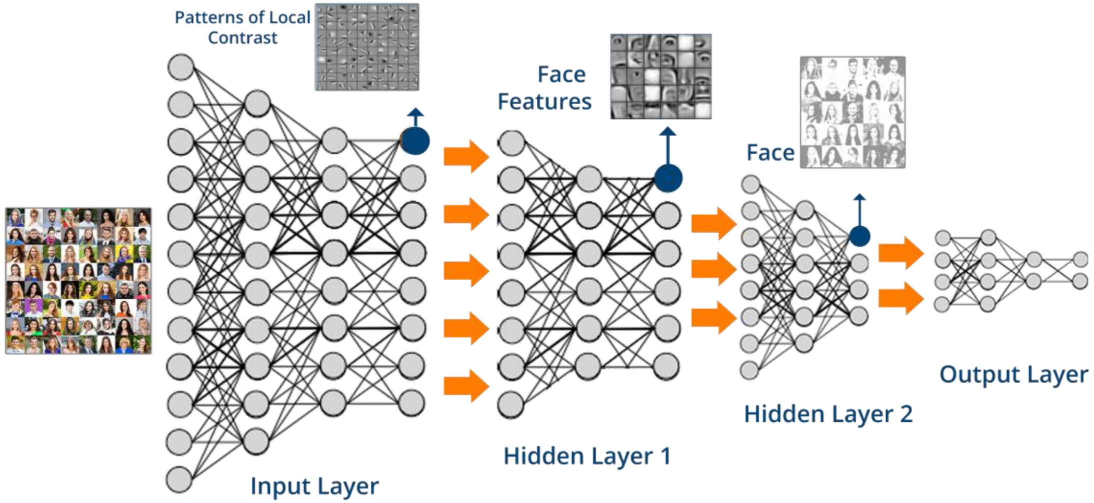
Etc.

Robot Vision vs. Computer Vision

- Robot vision is **embodied, active** and **environmentally situated**
  - Like human ego-vision

Key Techniques – AI and Deep Learning

AI and especially Deep Learning provide the "brain" that learns to use those tools effectively. Instead of relying on hand-coded rules, these systems learn directly from vast amounts of data.



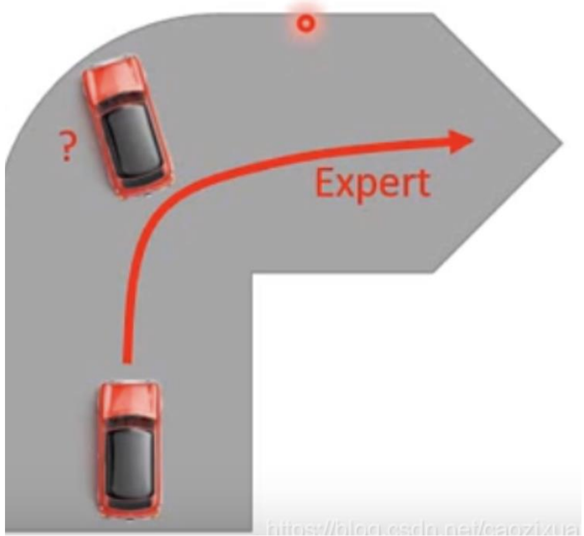
Convolutional Neural Networks

Reinforcement Learning

Imitation Learning

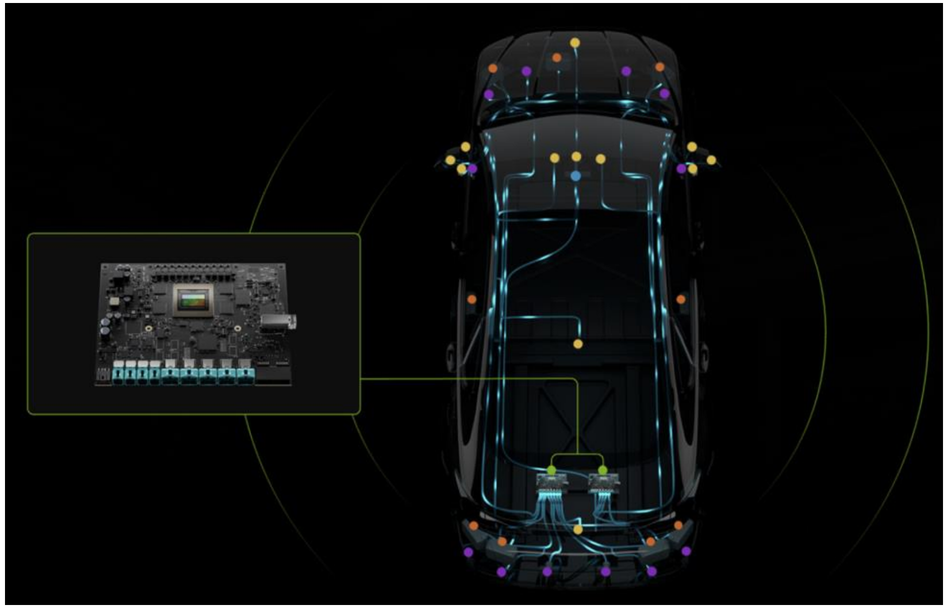
Transformers

Visual-Language Action Model



Key Techniques – Computing Hardware

The sophisticated algorithms for perception and decision-making require immense computational power. The computing hardware is the central nervous system of the intelligent vehicle, responsible for processing massive data streams in real-time under rigorous constraints.

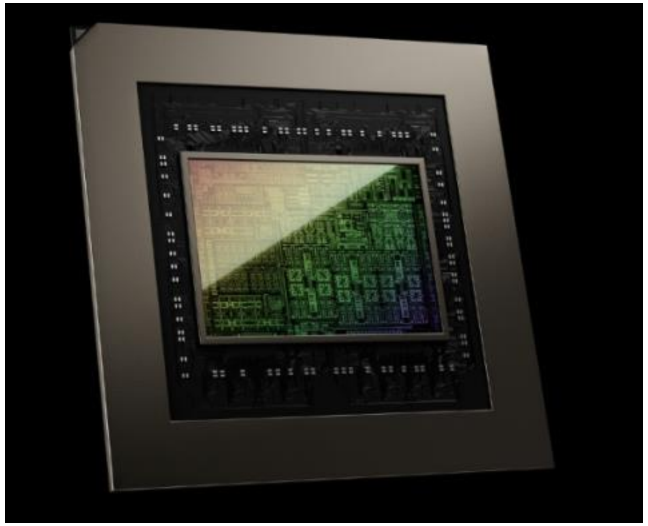


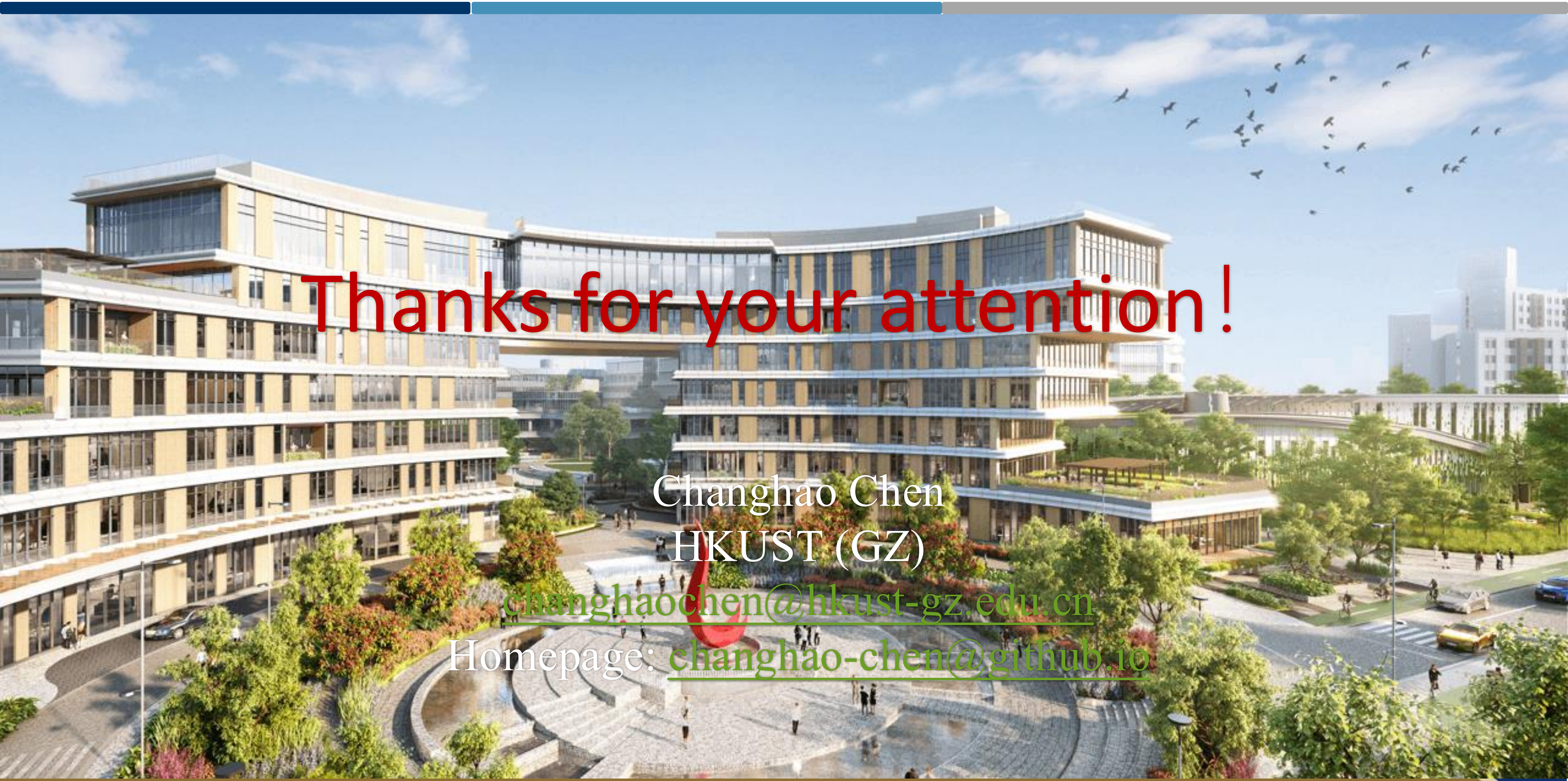
High-Performance Computing

AI Accelerators (GPUs & NPUs)

System-on-a-Chip (SoC) Integration

Edge Computing





Thanks for your attention！

Changhao Chen
HKUST (GZ)
changhaochen@hkust-gz.edu.cn
Homepage: changhao-chen@github.io