Privileged Knowledge Distillation and *Hierarchical Framework* in Reinforcement Learning

CS586 – Robot Motion Planning and Applications
Speaker: Taegeun Yang
2025.04.09

SGVR KAIST

Content

Recap: Reinforcement Learning

Privileged Knowledge Distillation

- Concept
- Teacher-Student Framework
- Regularized Online Adaptation (ROA)

Hierarchical Reinforcement Learning (HRL)

- Concept
- Related Works – *Manipulation, Locomotion, Navigation*

Our Work

- Efficient Navigation Among Movable Obstacles using a Mobile Manipulator via Hierarchical Policy Learning

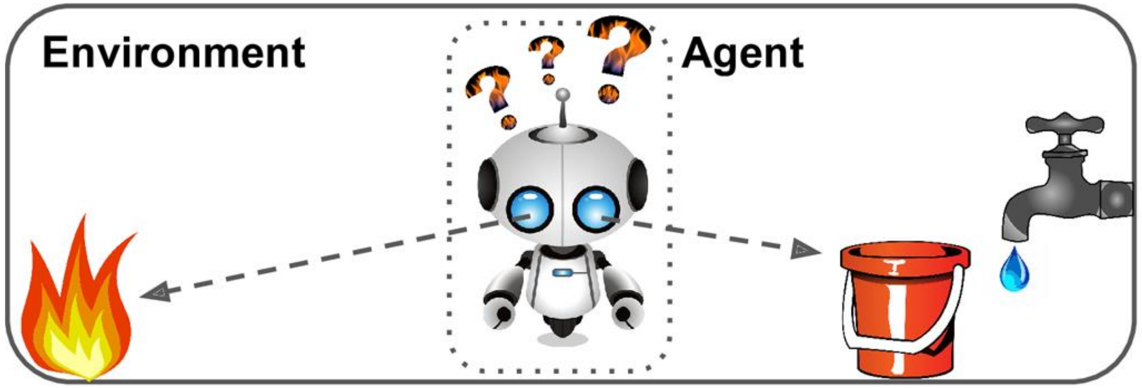
Reinforcement Learning (RL)

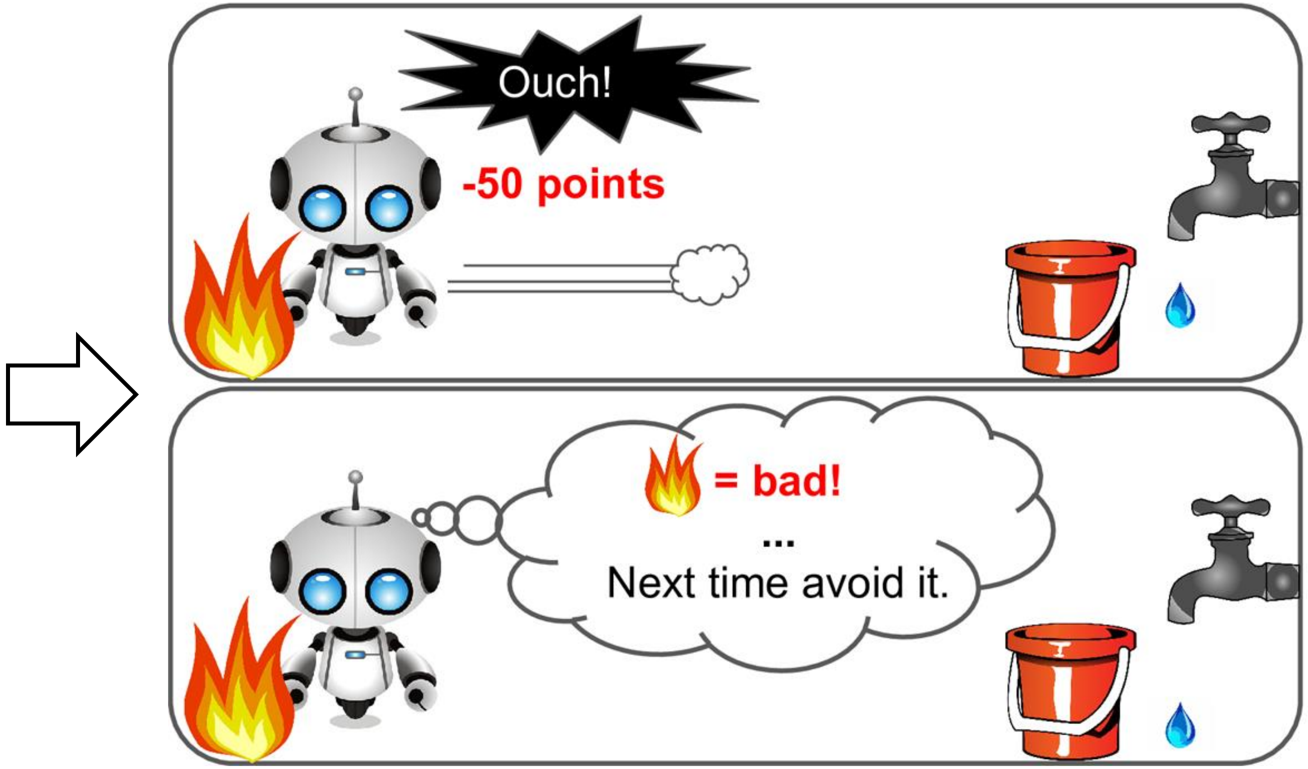
Reinforcement Learning (RL)

Agent learns to make decisions by <u>interacting with an environments</u>

- <u>Trial and error</u> interactions with an environment

Task: extinguish the fire





Reinforcement Learning (RL)

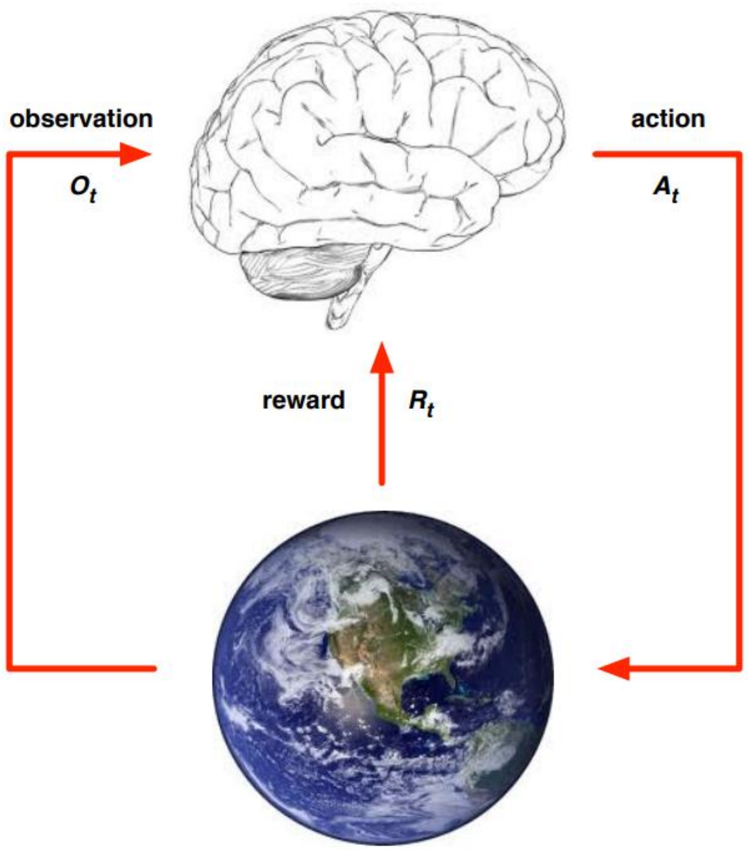
Agent learns to make decisions by <u>interacting with an environments</u>

- <u>Trial and error</u> interactions with an environment

At each time step *t*

- **Agent**:
  - Receives observation O_t *(e.g., sensing)*
  - Executes action A_t *(e.g., move forward)*
  - Receives reward R_t *(task related)*
- **Environment**:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits reward R_{t+1}

Agent (robot)



Environment
(physics simulation, real world)

Reinforcement Learning (RL)

Problem Formulation

- Markov Decision Problem (MDP)
  - $< \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{T}, \boldsymbol{\gamma} >$
  - S: state space
  - A: action space
  - R: reward function $\quad R: S \times A \times S \to \mathbb{R}$
  - T: state transition function $\quad T: S \times A \to S$
  - γ: discount factor $\quad \gamma \in [0,1)$

Objective

- Maximize expected cumulative reward *(return)* : $\sum_{t=0}^{\infty} \gamma^t r_t$
  - Policy $\pi_\theta : S \to A$
  - $\theta^* = \underset{\theta}{\operatorname{argmax}} \, \mathbb{E}_{a \sim \pi_\theta(s),\, s' \sim T(s,a)} \left[\sum_{t=0}^{\infty} \gamma^t R(s, a, s')\right]$

Reinforcement Learning (RL)

Example 1: Atari Example

- S: state space − image
- A: action space − move left/right
- R: reward function
  - Breaking a block: +1
  - Missing a ball: -1
- T: state transition function
  - How the ball moves
  - Block breaks when hit by ball
- γ: discount factor $\gamma \in [0,1)$

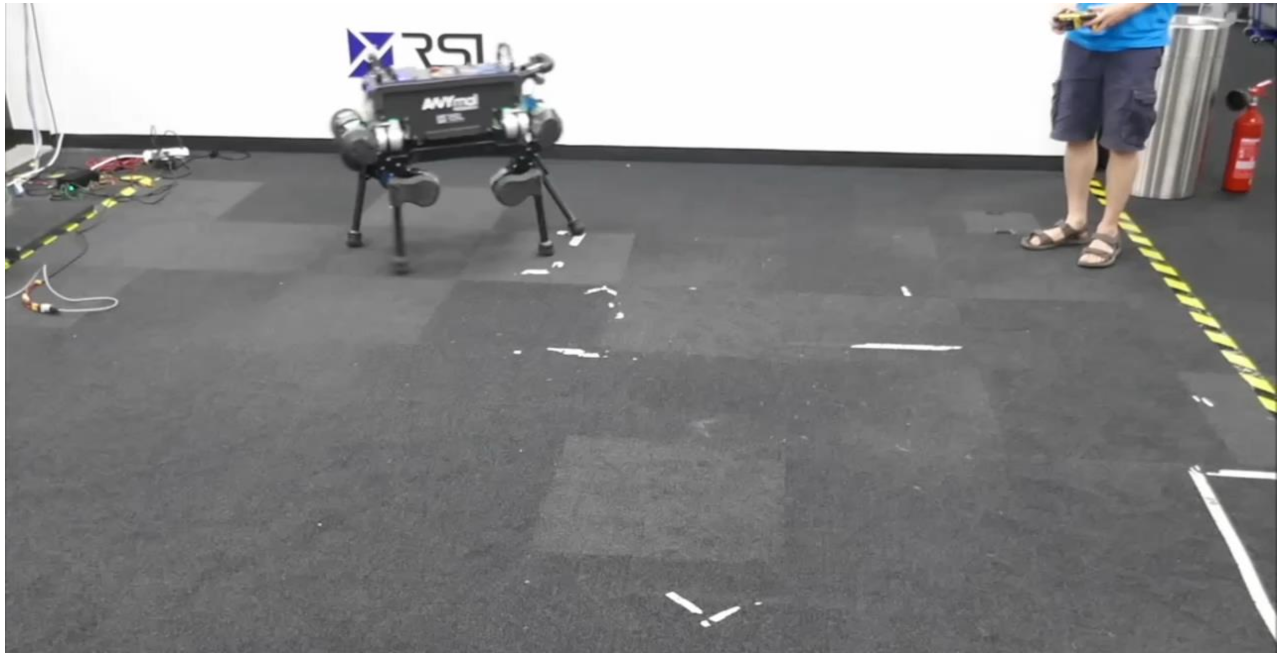
- **Policy** determines where the paddle (agent) moves based on the image input

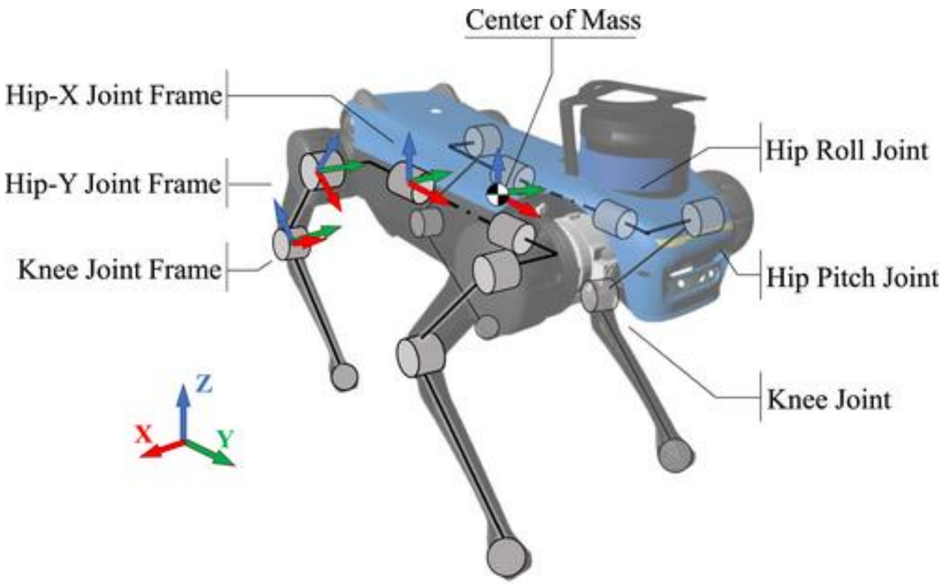
dhttps://gsurma.medium.com/atari-reinforcement-learning-in-depth-part-1-ddqn-ceaa762a546f

Reinforcement Learning (RL)

Example 2: Quadruped Robot Locomotion

- Given movement command: forward velocity, lateral velocity, and angular velocity (yaw)
- Policy output: Action – target joint position *(12-dim)*





https://www.youtube.com/watch?v=aTDkYFZFWug
https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/csy2.12062

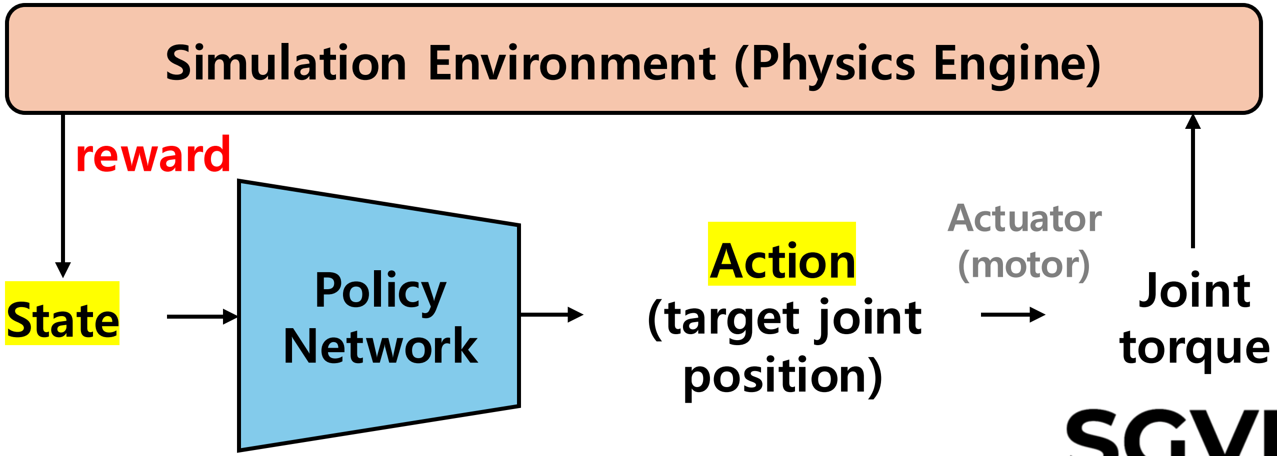
Reinforcement Learning (RL)

Example 2: Quadruped Robot Locomotion

- Given movement command: forward velocity, lateral velocity, and angular velocity (yaw)

- Policy *input* : **State** – current robot state, target movement command, etc.
- Policy *output* : **Action** – target joint position *(12-dim)*

- **Reward** – command tracking, stable locomotion (i.e., avoid falling), etc.
- **Transition** – Physics Engine (dynamics)



Privileged Knowledge Distillation

Privileged Knowledge Distillation

Privileged Knowledge

- Information that is available during training but not during deployment (testing)
- In the context of RL
  - Extra observations or state variables
  - E.g., ground friction, forces *(not be accessible in the real-world)*
- Provide more complete understanding of the environment

Knowledge Distillation

- Training technique - weaker policy learns by mimicking the behavior of stronger policy
  student **teacher**
- Strong policy: trained with full state information *(w/ privileged information)*
- Weak policy: trained with partial state information *(w/o privileged information)*

Privileged Knowledge Distillation

Example 1: Teacher-Student Framework

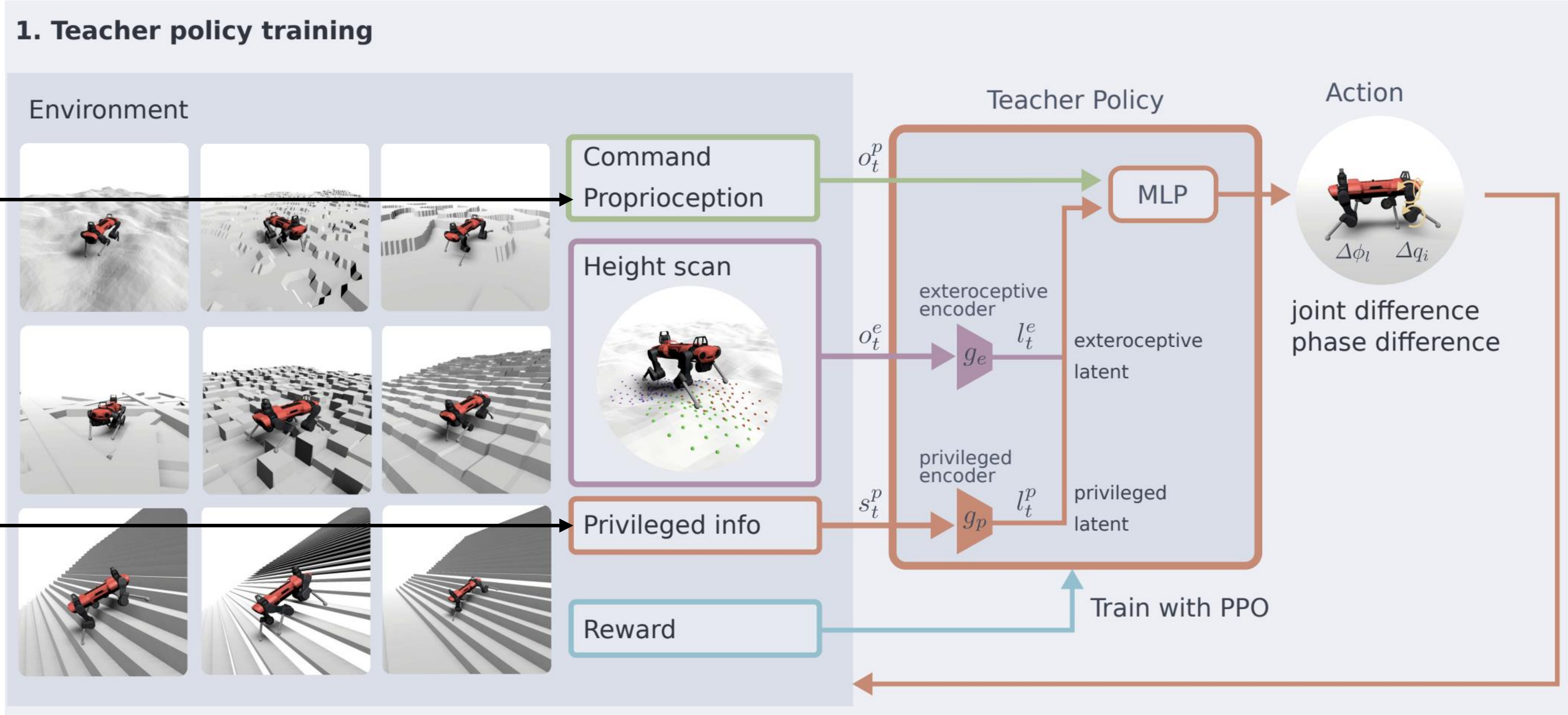
- Train **teacher** policy with privileged information **(RL)**

Robot's internal sensory data - Obtainable

Body velocity
Body orientation
Joint position
Joint velocity
…

Contact states
Contact forces
Contact normal
Friction coef.
…

Unobtainable in real-world

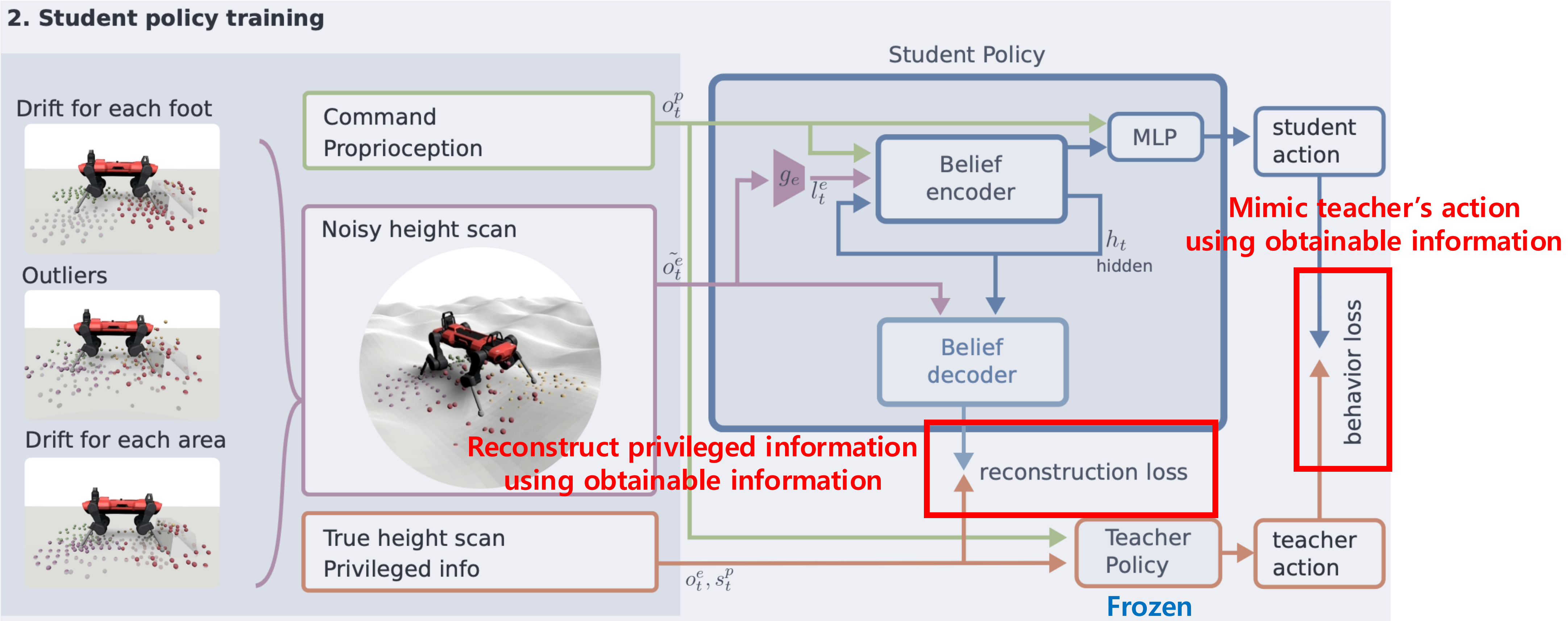


Learning robust perceptive locomotion for quadrupedal robots in the wild (Science Robotics, 2022)

Privileged Knowledge Distillation

Example 1: Teacher-Student Framework

- Train **student** policy without privileged information **(Supervised Learning)**



Privileged Knowledge Distillation

Example 1: Teacher-Student Framework

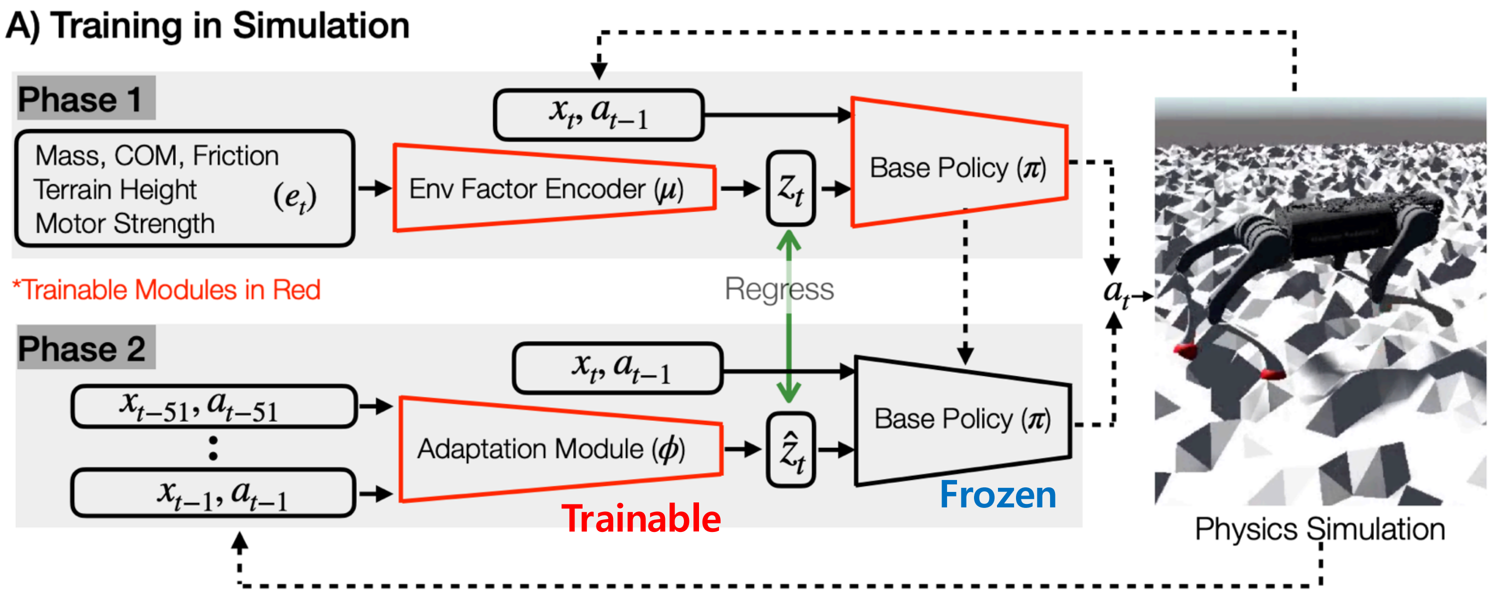
- Deploy using **student** policy *(real-world)*

Learning robust perceptive locomotion for quadrupedal robots in the wild (Science Robotics, 2022)

Privileged Knowledge Distillation

Example 2: Adaptation Module

- Estimate *encoded* privileged information using history of the robot's state
  - Perform robust and adaptive locomotion



RMA: Rapid Motor Adaptation for Legged Robots (RSS, 2021)

Privileged Knowledge Distillation

Example 3: Regularized Online Adaptation (ROA)

- 2-step training framework → teacher policy may not provide supervision that student can learn
- Remove 2-step framework
  - Training Loss: $L(\theta_\pi, \theta_\mu, \theta_\phi) = \underbrace{-J(\theta_\pi, \theta_\mu)}_{\text{RL objective}} + \underbrace{\lambda}_{\text{Increase from 0 to 1}} \| z^\mu - \mathrm{sg}[z^\phi] \|_2 + \| \mathrm{sg}[z^\mu] - z^\phi \|_2$

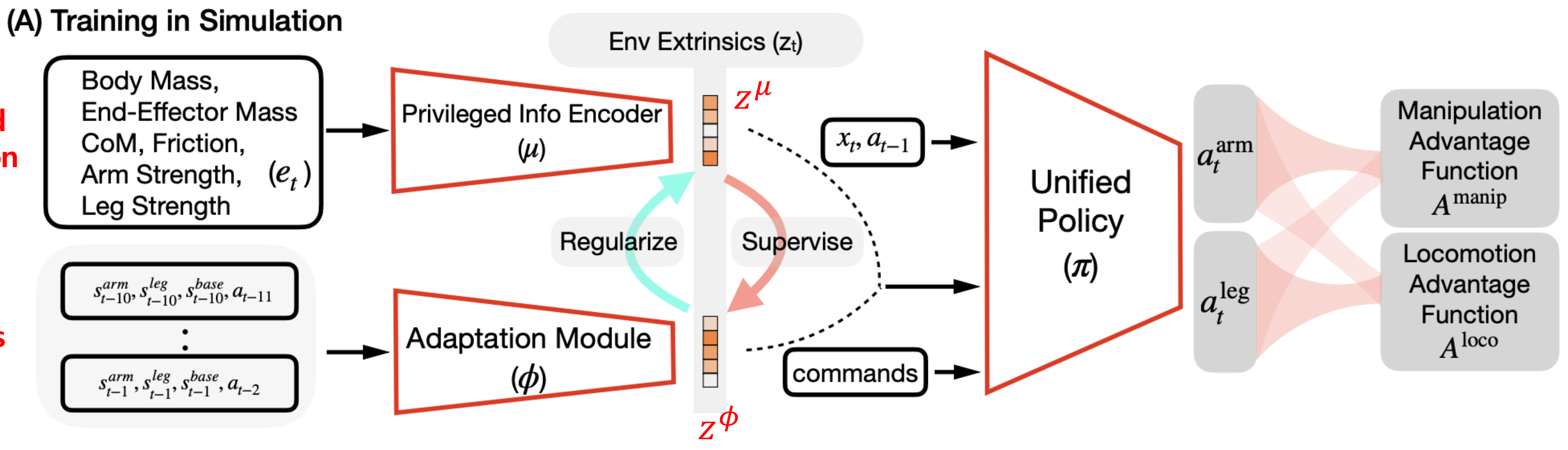
sg: stop gradient

RL objective **Increase from 0 to 1**

RL policy uses privileged information

Privileged Information

Recent observation history from robot's onboard sensors



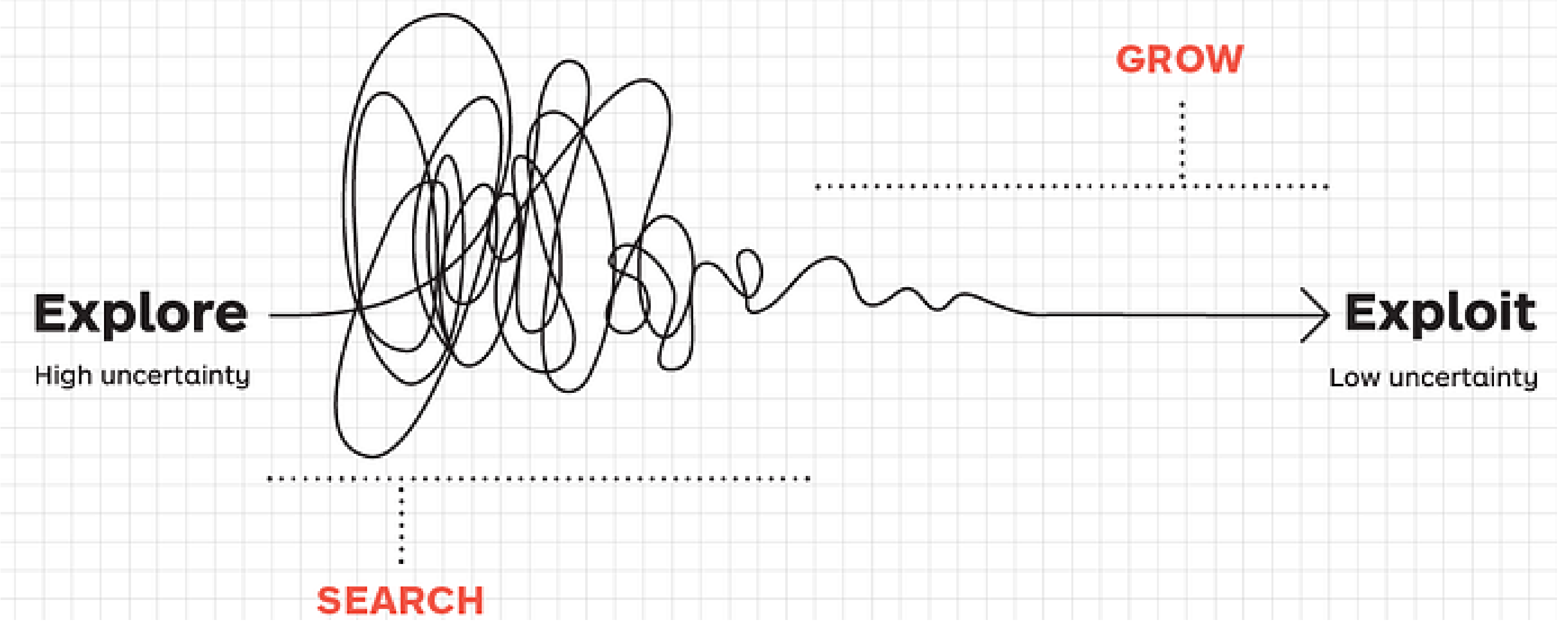
Deep Whole-Body Control: Learning a Unified Policy for Manipulation and Locomotion (CoRL, 2022)

Hierarchical Reinforcement Learning (HRL)

Hierarchical Reinforcement Learning (HRL)

Exploration **in Reinforcement Learning (RL)**

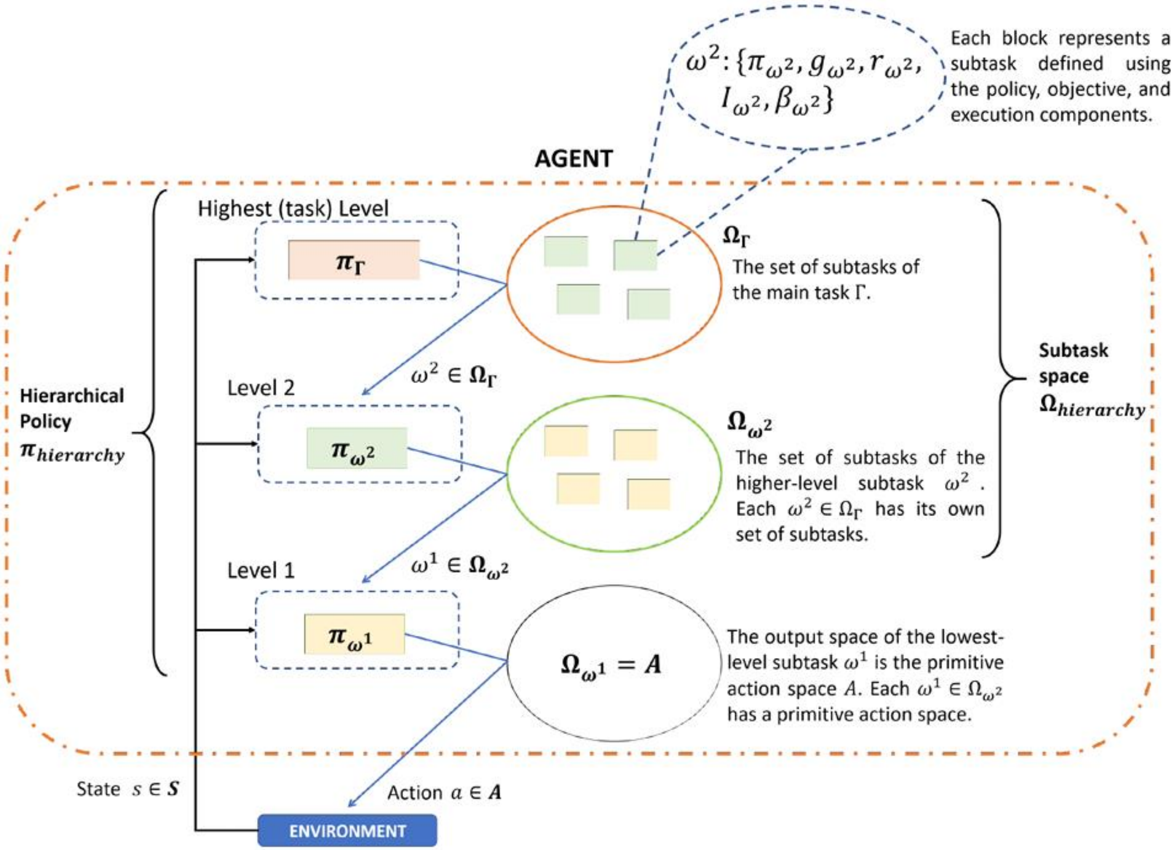
- <u>Essential for learning optimal policies</u> in uncertain environments
  - w/o exploration: agent may become stuck in local optima
- Become more challenging as the observation *(input)* & action *(output)* space expands



Hierarchical Reinforcement Learning (HRL)

Exploration **in Reinforcement Learning (RL)**

- Observation & Action space ↑ → difficulty of effective exploration ↑
- Hierarchical structure
  - Break down a complex problem into multiple sub-problems

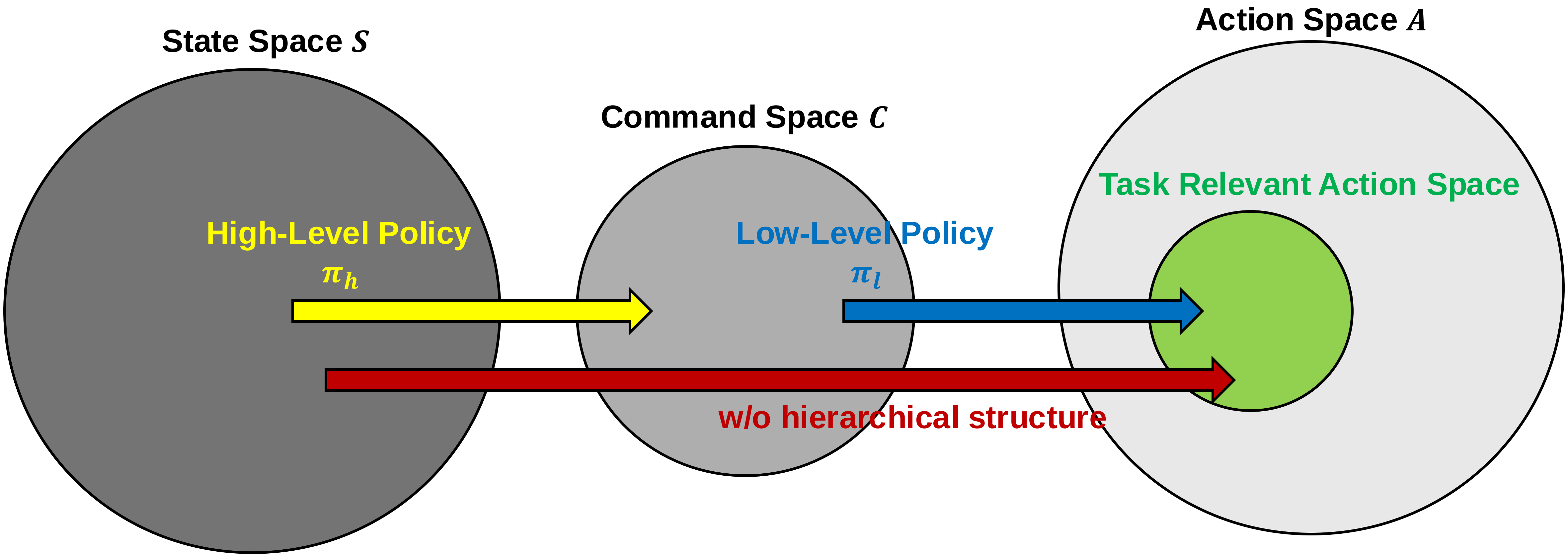


SGVR KAIST

Hierarchical Reinforcement Learning (HRL)

Recent Hierarchical Framework in RL

- **High-Level Policy** : decide task-relevant command *(decision making)*
- **Low-Level Policy** : execute given command *(control)*

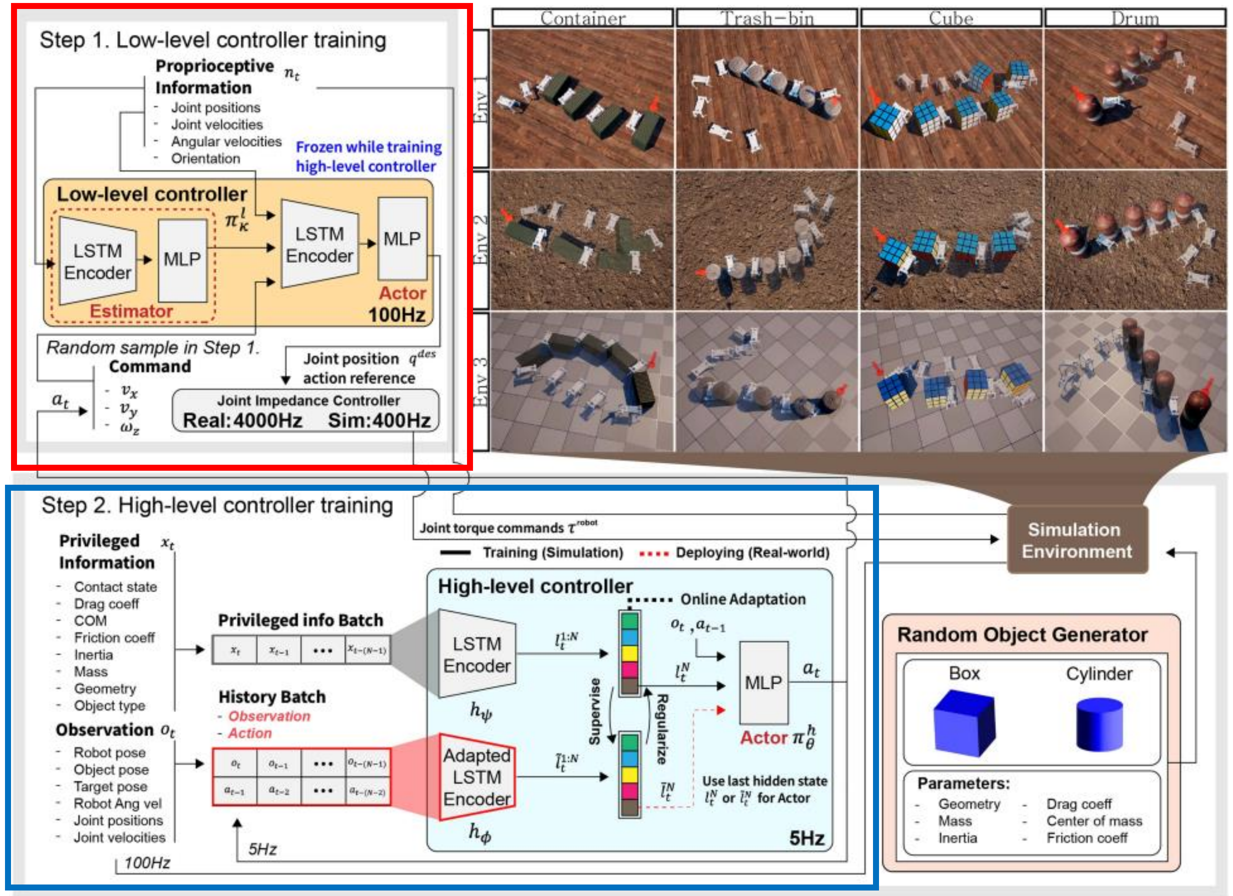


Hierarchical Reinforcement Learning (HRL)

Example 1: Manipulation

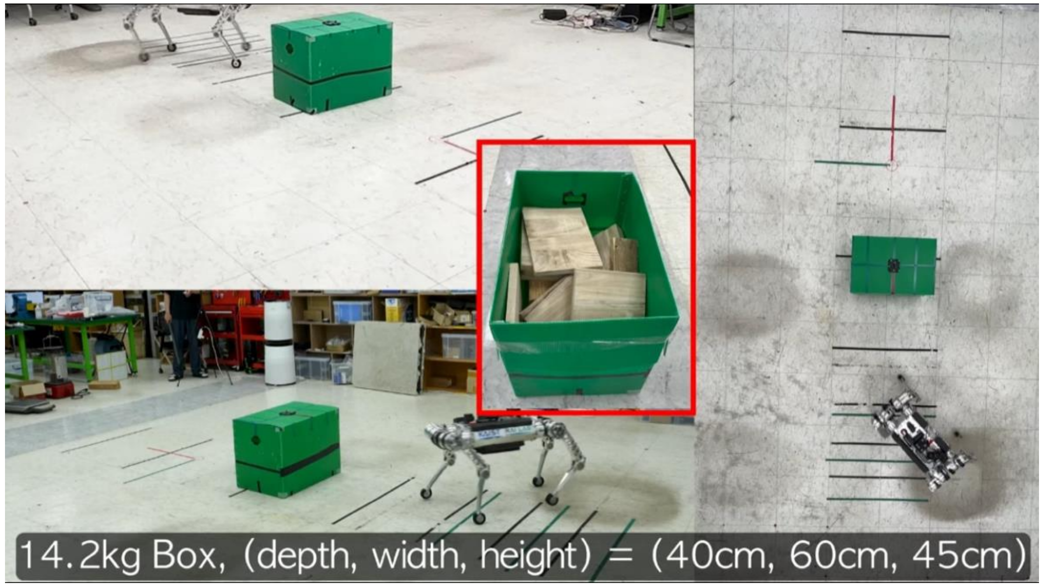
- *High-Level Policy* : decide robot's velocity *(command)* to push the object to the target location

Low-Level
- *Low-Level Policy* : execute given command *(velocity command → target joint positions)*



High-Level w/ privileged learning (ROA)

* During High-Level training,
Low-Level policy keeps frozen



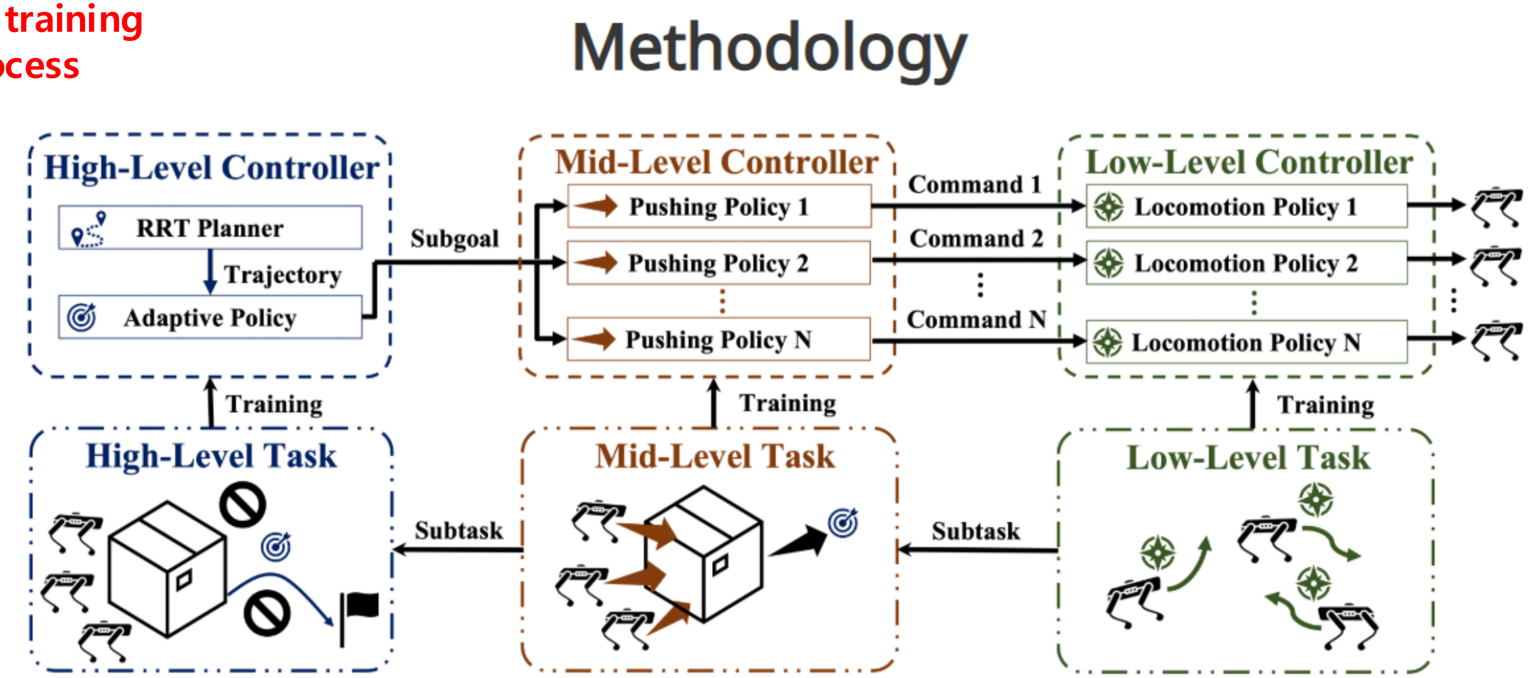
Learning whole-body manipulation for quadrupedal robot (RA-L, 2023)

Hierarchical Reinforcement Learning (HRL)

Example 2: Manipulation

- *High-Level Policy* : generate subgoals for the object
- *Mid-Level Policy* : decide robots' velocity *(command)* to push the object to the subgoal
- *Low-Level Policy* : execute given velocity command

3-step training process



Learning Multi-Agent Loco-Manipulation for Long-Horizon Quadrupedal pushing (arXiv, 2024)

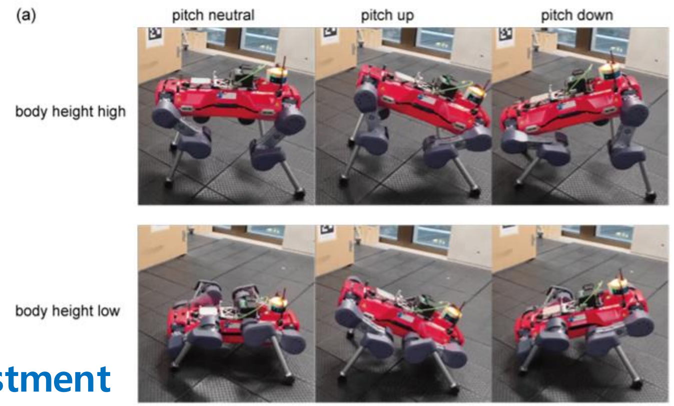
Hierarchical Reinforcement Learning (HRL)

Example 3: Locomotion

- Follow given velocity command given by human in confined space

  (v_x, v_y, w_z)

- Requires body adjustments to avoid collision



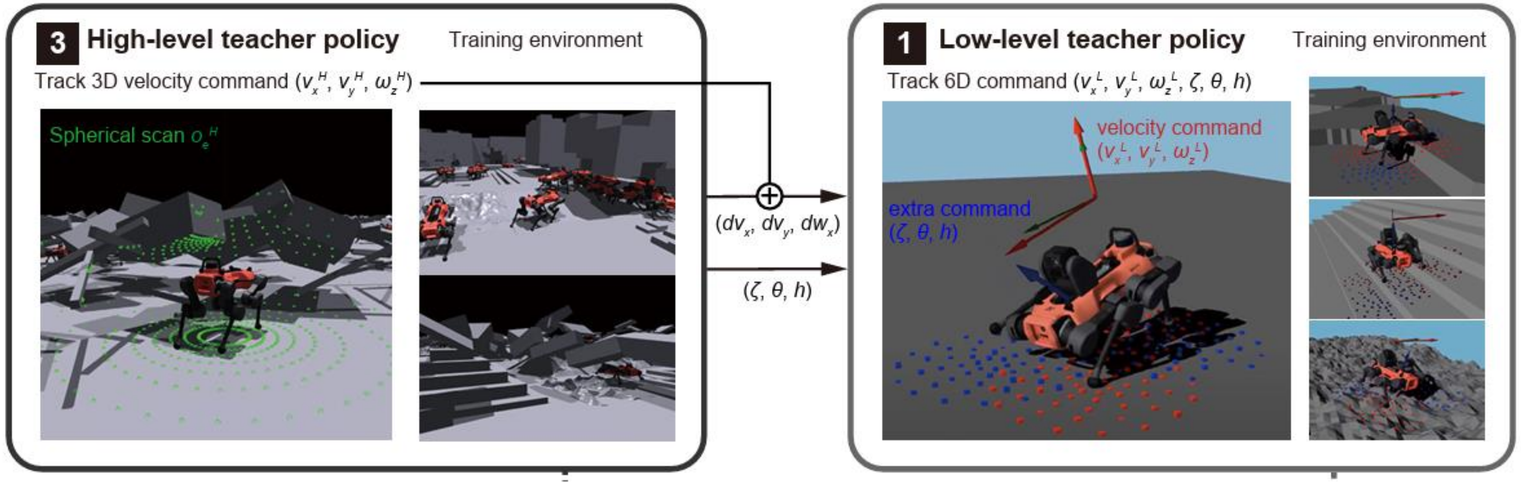
Body adjustment

Learning to walk in confined spaces using 3d representations (ICRA, 2024)

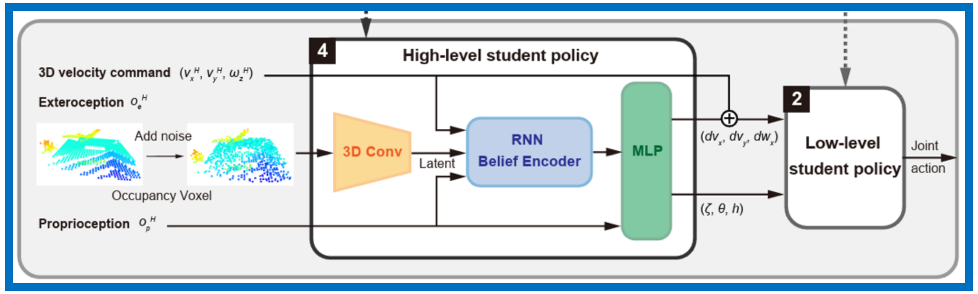
Hierarchical Reinforcement Learning (HRL)

Example 3: Locomotion Body velocity & Body adjustment motion

- *High-Level Policy* : generate 6-dim command → avoid collision, track 3-dim velocity command
- *Low-Level Policy* : execute given 6-dim command *(command → joint action)*



Hierarchical Structure



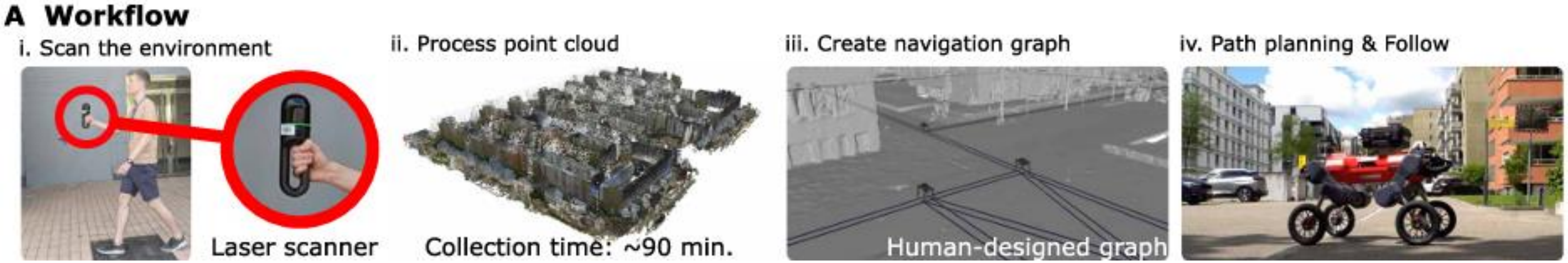
Knowledge Distillation

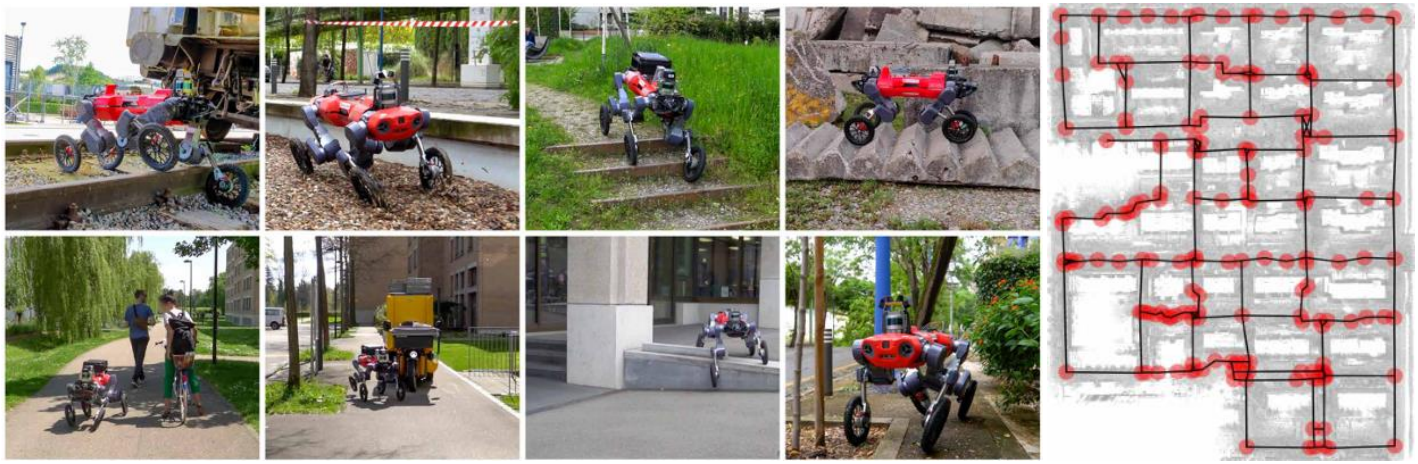
Learning to walk in confined spaces using 3d representations (ICRA, 2024)

Hierarchical Reinforcement Learning (HRL)

Example 4: Navigation

- Autonomous navigation *(map → planning → follow)*
- Sensor data *(e.g., LiDAR, Camera)* → Joint action *(avoid collision, path following, etc.)* **: Hard to learn**



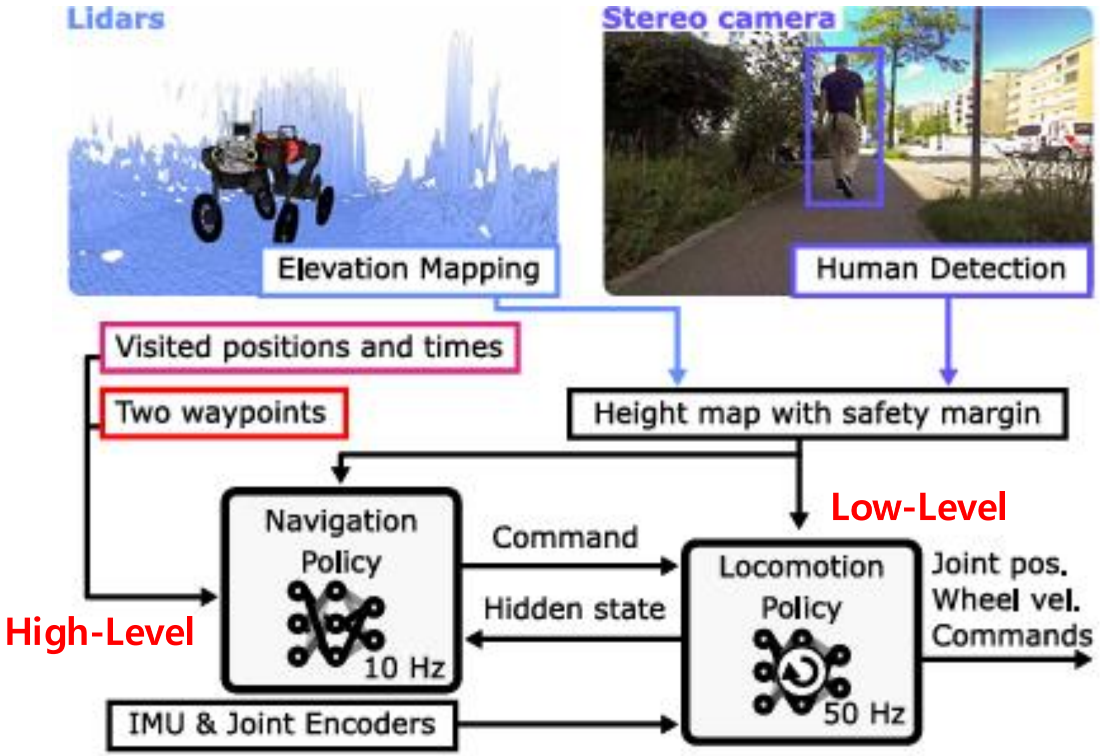


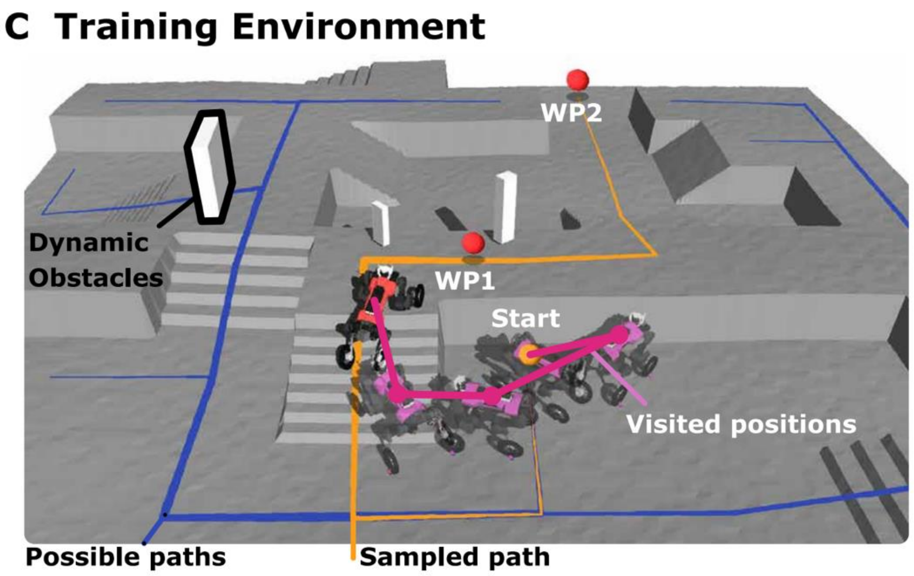
Learning robust autonomous navigation and locomotion for wheeled-legged robots (Science Robotics, 2024)

Hierarchical Reinforcement Learning (HRL)

Example 4: Navigation

- *High-Level Policy* : consider surrounding environment & path → generate velocity command (v_x, v_y, w_z)
- *Low-Level Policy* : execute given command *(velocity command → target joint position/velocity)*





Learning robust autonomous navigation and locomotion for wheeled-legged robots (Science Robotics, 2024)

SGVR KAIST

Thanks for your attention

Any question will be welcome

CS586 – Robot Motion Planning and Applications
Speaker: Taegeun Yang
2025.04.09