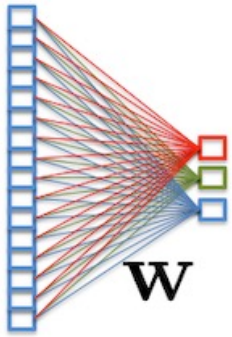
AI and Data Analytics

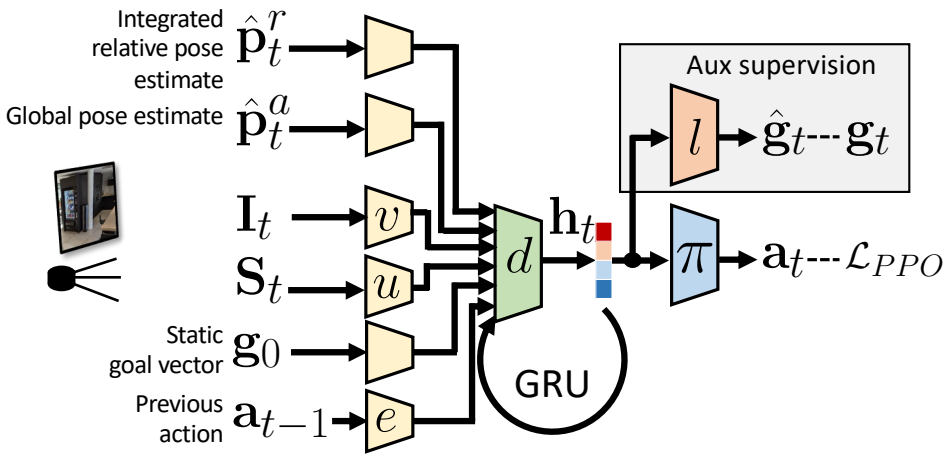
3.3 Reinforcement Learning – a concrete example

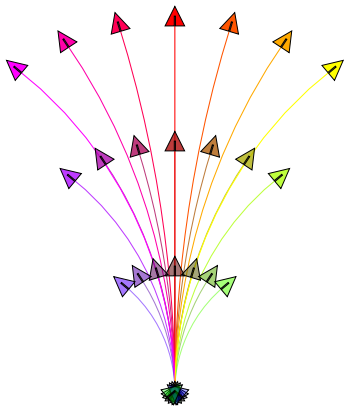


Christian Wolf

Agent architecture

CVPR 2024
G.Bono, H. Poirier, L. Antsfeld, G. Monaci, B. Chidlovskii, C. Wolf





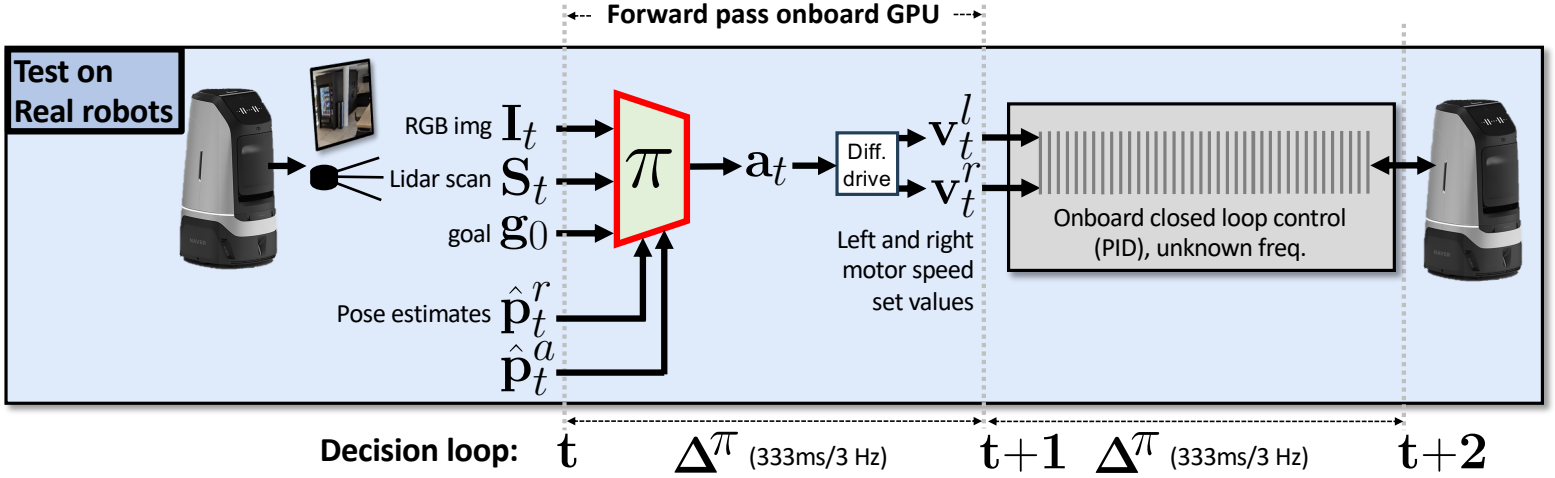
Input: RGB, Lidar, pose estimates from onboard measurements.
The agent does **not** have access to a map.

28 discrete actions
(pairs of linear+angular velocities).

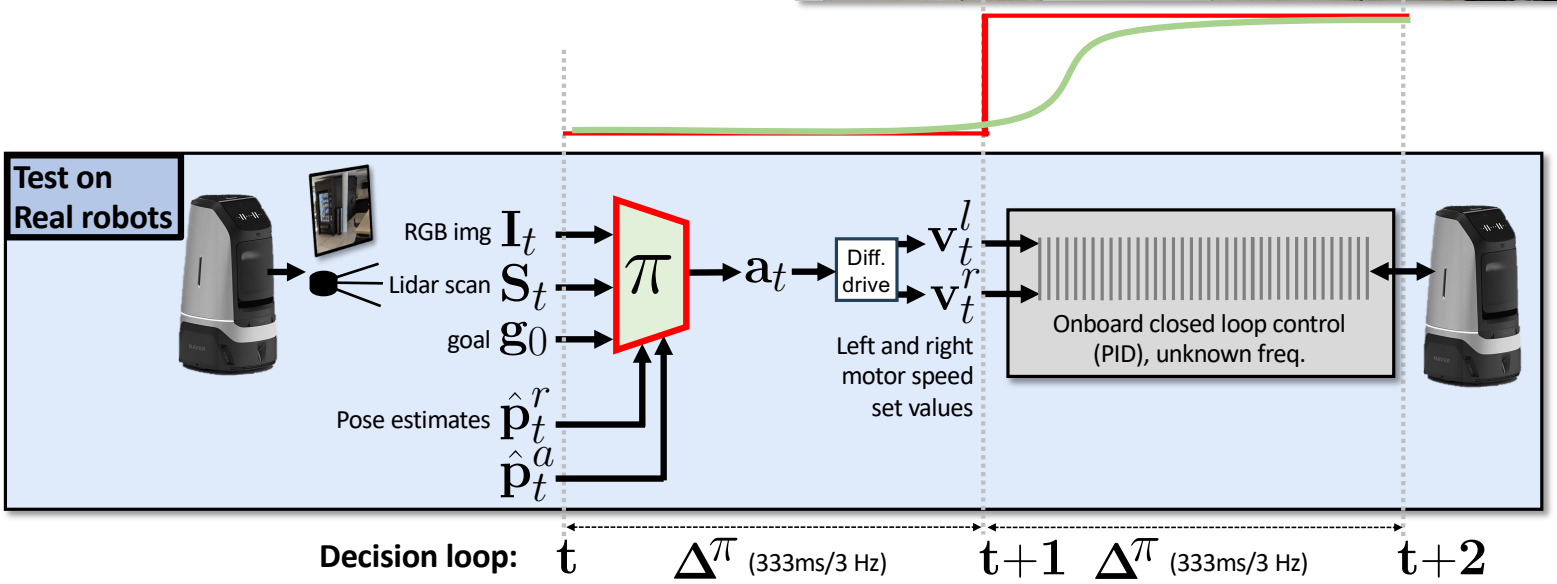
Testing configuration

CVPR 2024
G.Bono, H. Poirier, L. Antsfeld, G. Monaci, B. Chidlovskii, C. Wolf

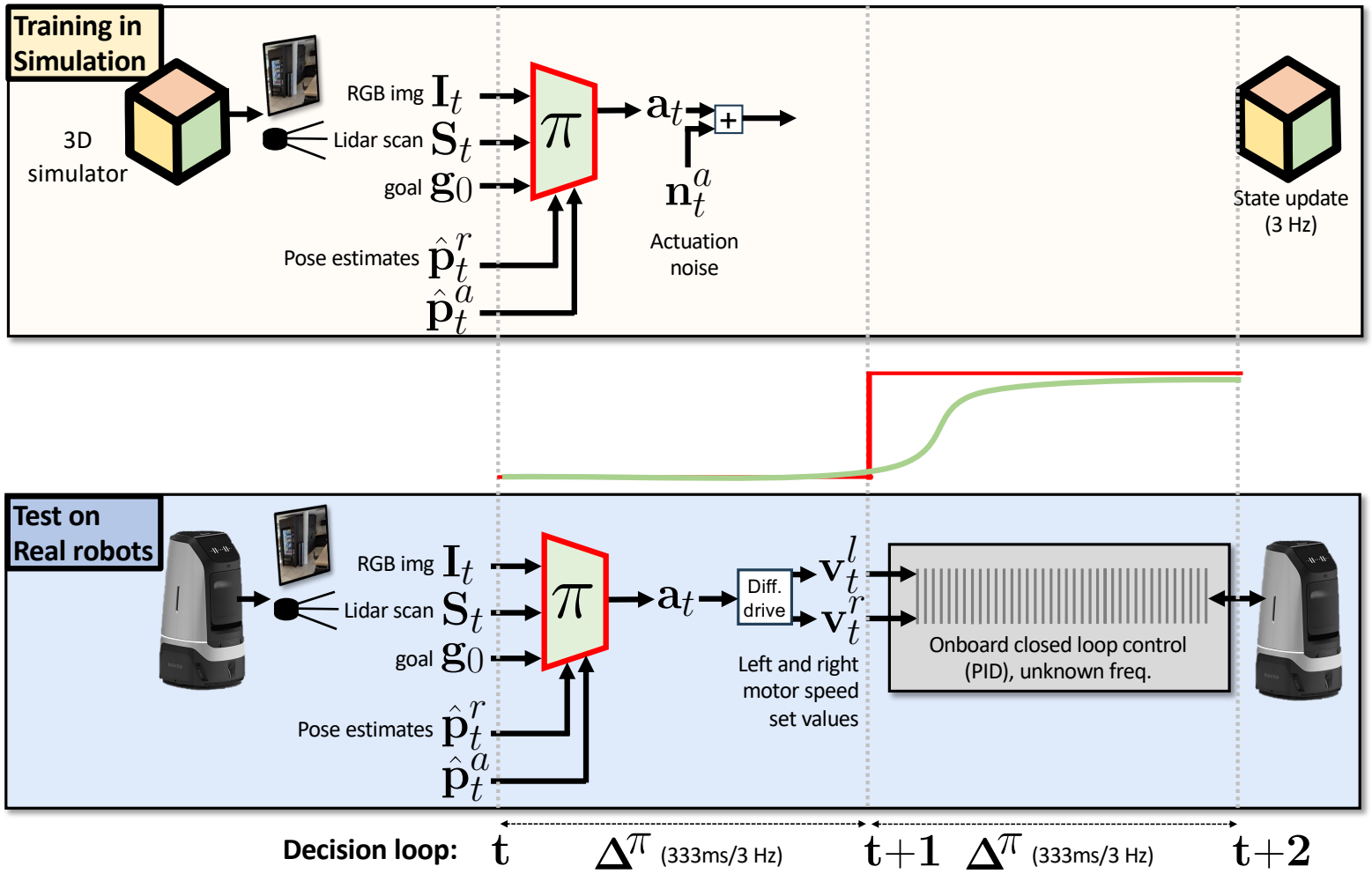
RGB Camera
4 depth cameras
Onboard GPU (Nvidia Jetson Orin)



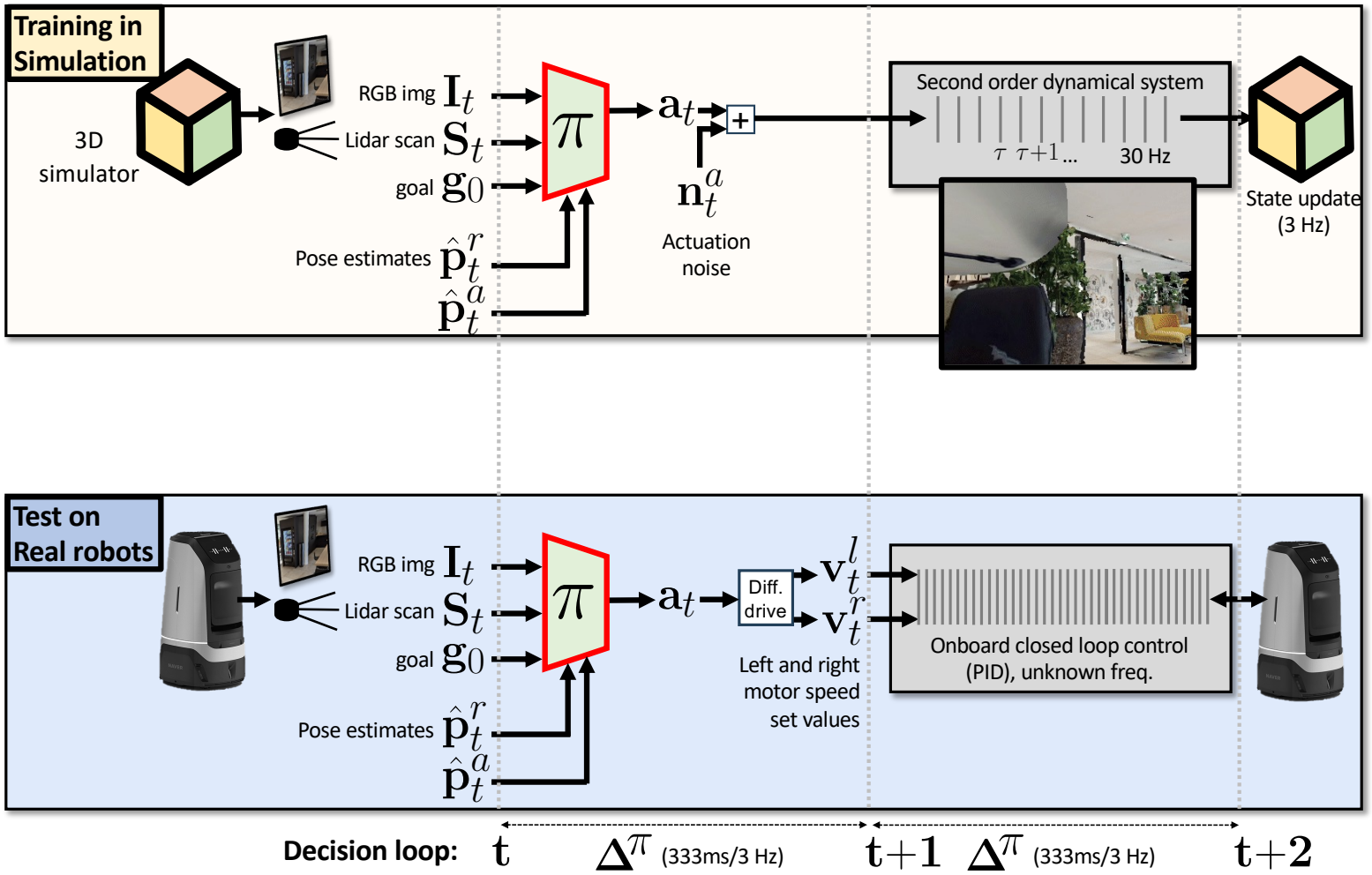
Realistic motion



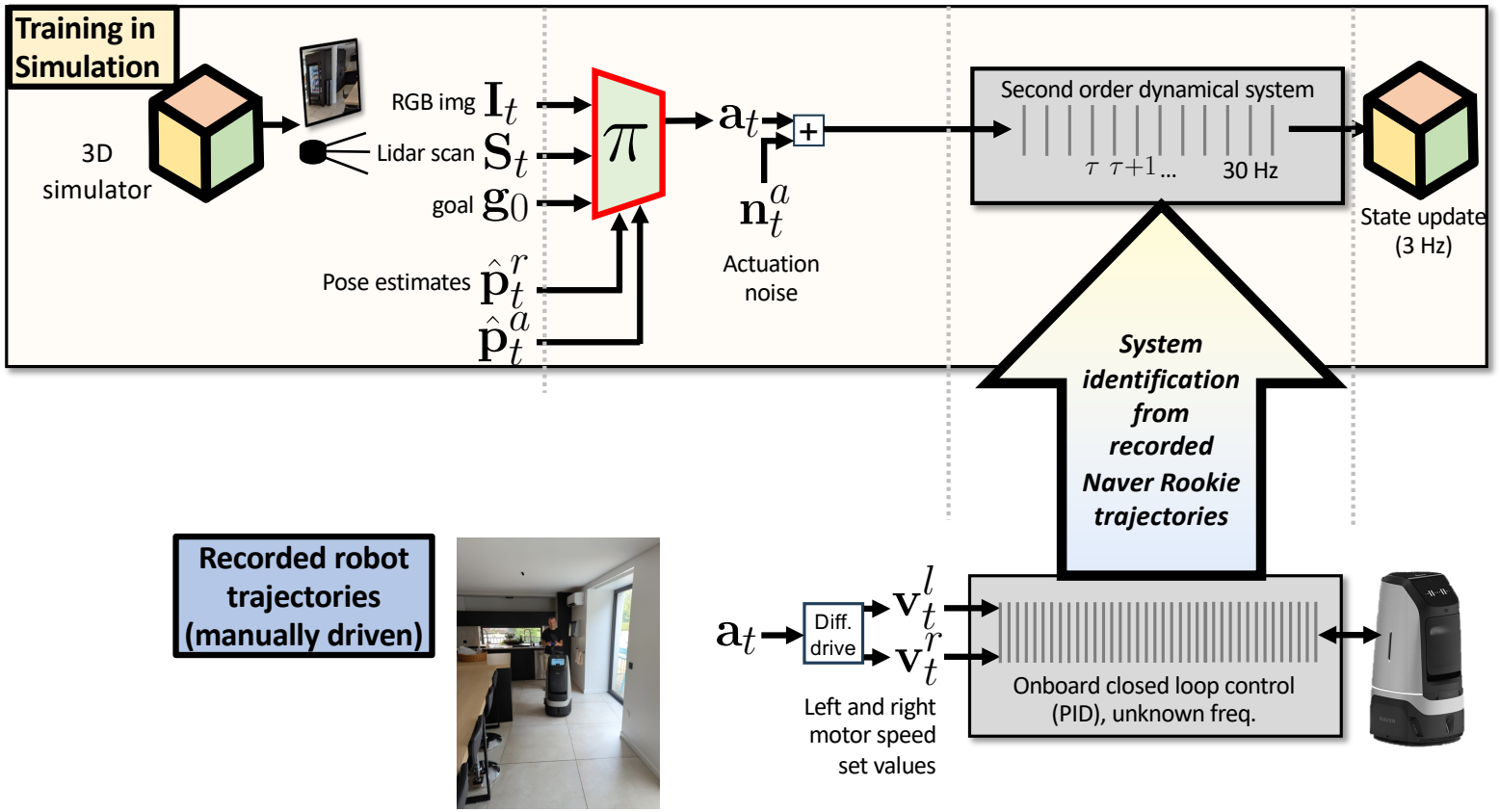
Simulated realistic motion



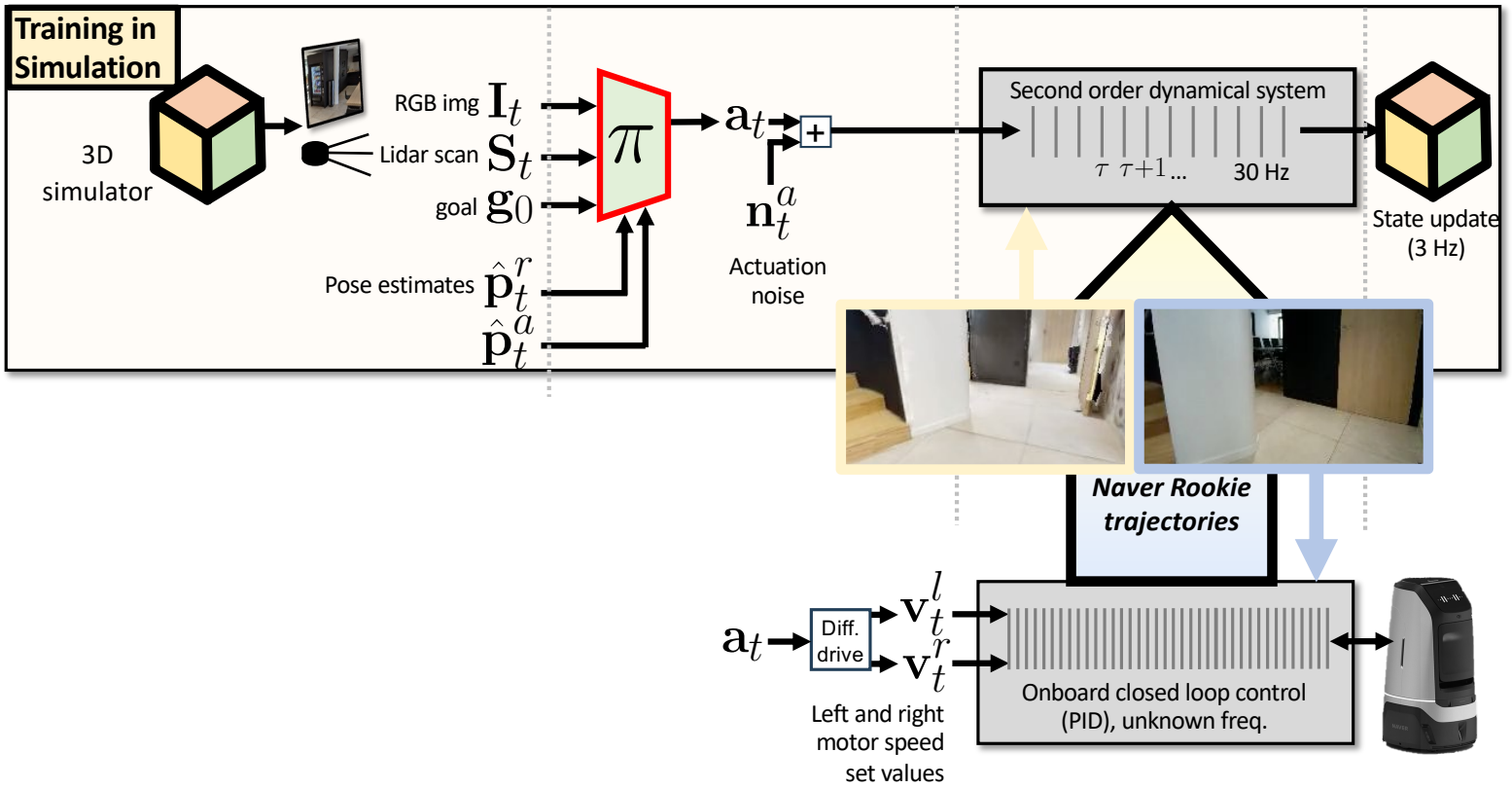
Second order dynamical model



System identification



System identification



Training with Reinforcement Learning (PPO)

Reward function:

$$r_t = \underbrace{\mathrm{S} \cdot \mathbb{I}_{\text{success}}}_{\text{Reward}} - \underbrace{\Delta_t^{\text{Geo}}}_{\text{Geodesic Distance}} - \underbrace{\lambda}_{\text{Step Penalty}} - \underbrace{\mathrm{C} \cdot \mathbb{I}_{\text{collision}}}_{\text{Collision Cost}}$$

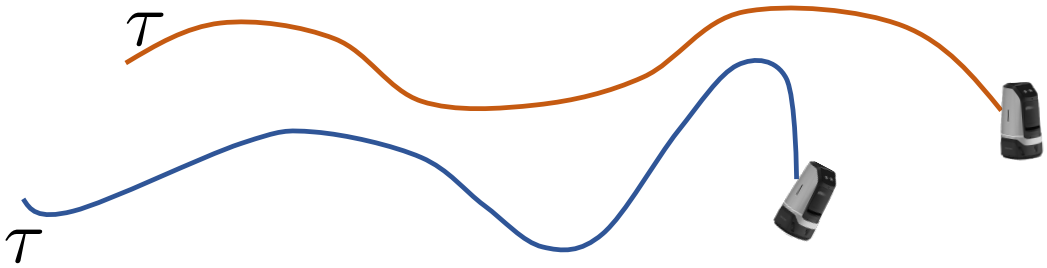
PPO algorithm: we train the policy to optimize cumulated reward:

$$\begin{aligned} J(\pi_\theta) &= \mathop{\mathbb{E}}_{\tau \sim \pi_\theta}[R(\tau)] \\ &= \mathop{\mathbb{E}}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T-1} r_{t+1} | \tau\right] \end{aligned}$$

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, Proximal Policy Optimization Algorithms, arXiv:1707.06347, 2017.

Training with Reinforcement Learning (PPO)

PPO collection step: interact with the environment with the current policy to collect rollouts τ.



PPO update step (*simplified*) : gradient update of the policy, nudge it to repeat actions that led to better-than-expected rewards.

$$L^{PG}(\theta) = \hat{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\underbrace{\log \pi_\theta \left(a_t \mid s_t \right)}_{\text{LogL of cur. } a_t} \quad \underbrace{\hat{A}_t}_{\text{Advantage of } a_t} \right]$$

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, Proximal Policy Optimization Algorithms, arXiv:1707.06347, 2017.

Datasets!

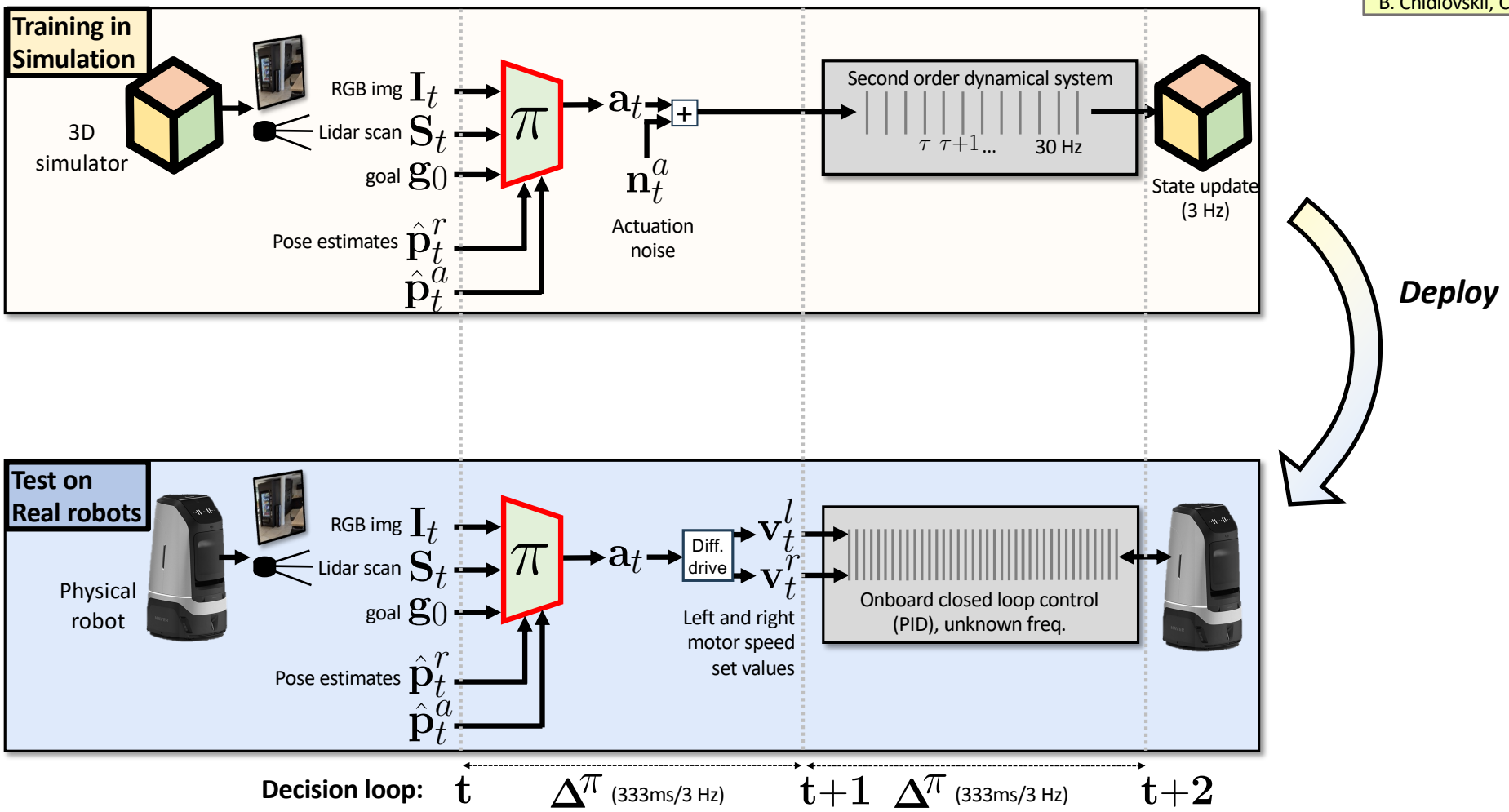


John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, Proximal Policy Optimization Algorithms, arXiv:1707.06347, 2017.

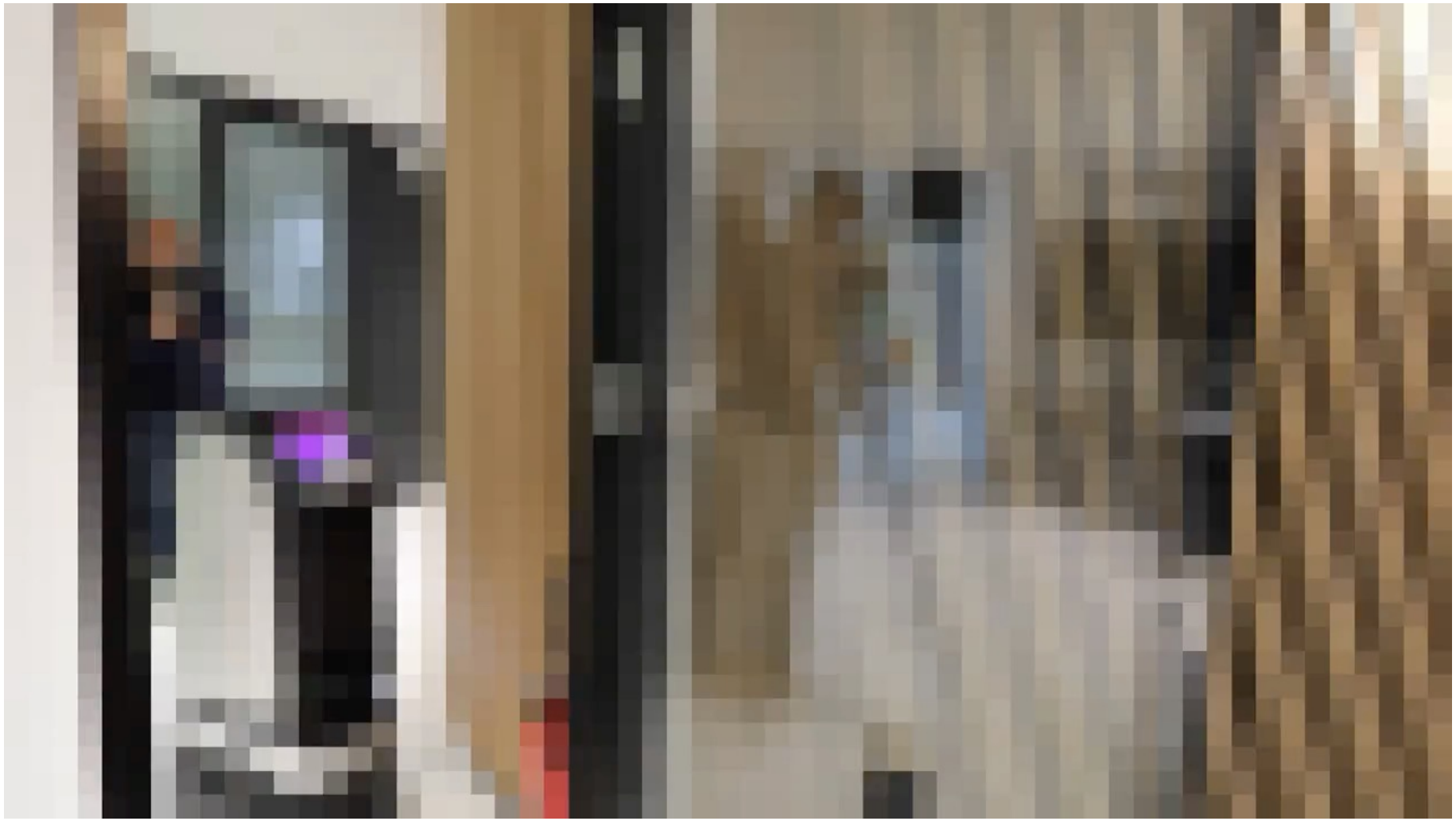
Deploy to real robot (all calculations onboard)

CVPR 2024

G.Bono, H. Poirier, L. Antsfeld, G. Monaci, B. Chidlovskii, C. Wolf



Example episode



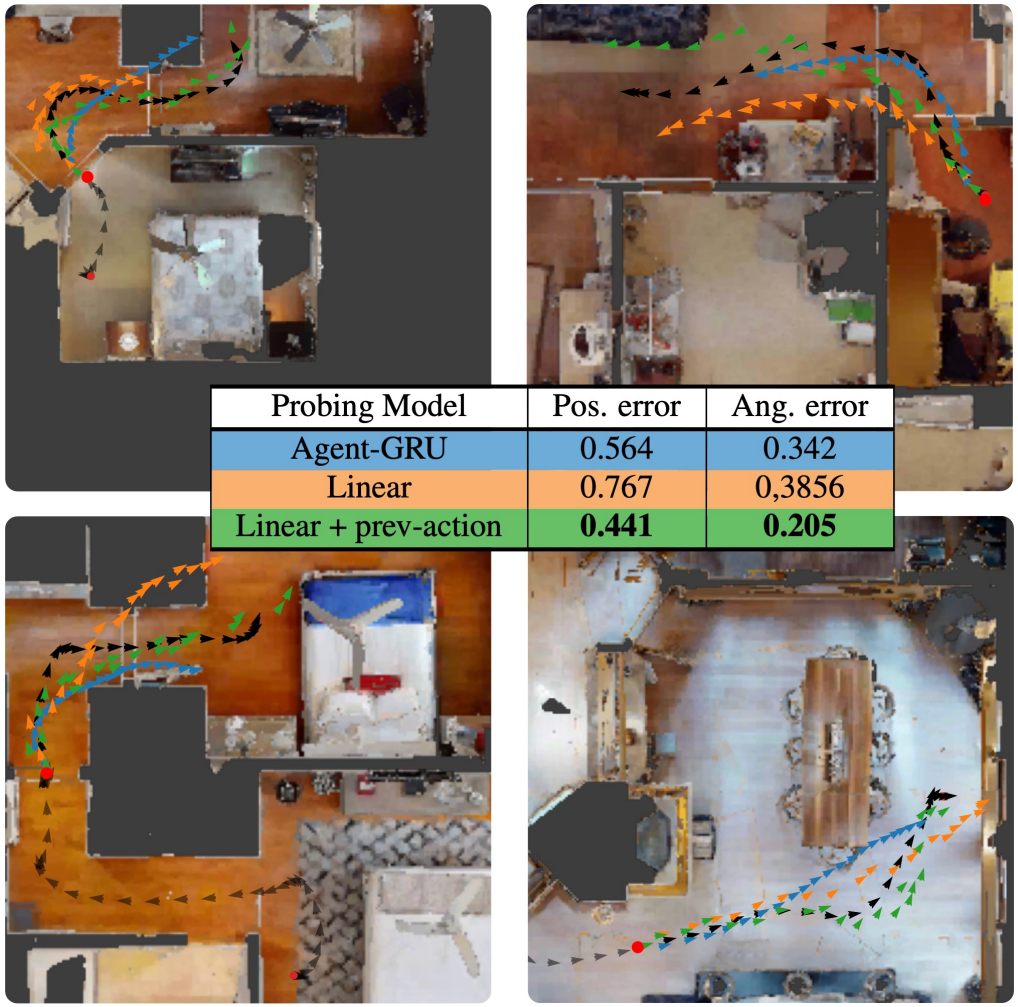
Social aspects of navigation

AI for Robotics Workshop at Naver Labs Europe, Nov. 2025

Social aspects of navigation

Internal demo at Naver Labs Europe

Does the agent plan? (1) local planning



Probing Model	Pos. error	Ang. error
Agent-GRU	0.564	0.342
Linear	0.767	0,3856
Linear + prev-action	**0.441**	**0.205**

We probe whether the future agent path is encoded in the recurrent memory of an end-to-end trained agent.

TLDR: yes it is, up to a certain point!

Does the agent plan? (2) strategies

CVPR 2025
S. Janny, H. Poirier, L. Antsfeld, G.Bono, G. Monaci, B. Chidlovskii, F. Giuliari, A. del Blue, C. Wolf

① Starting point

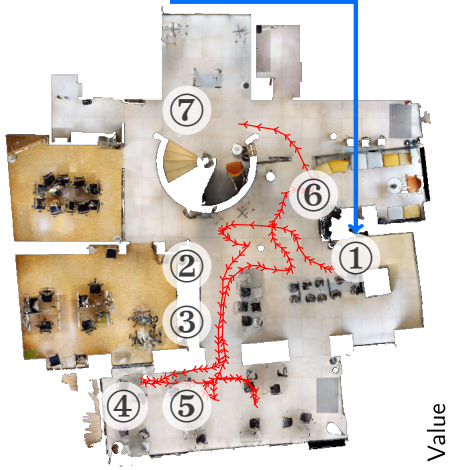
The robot wants to reach ⑦.

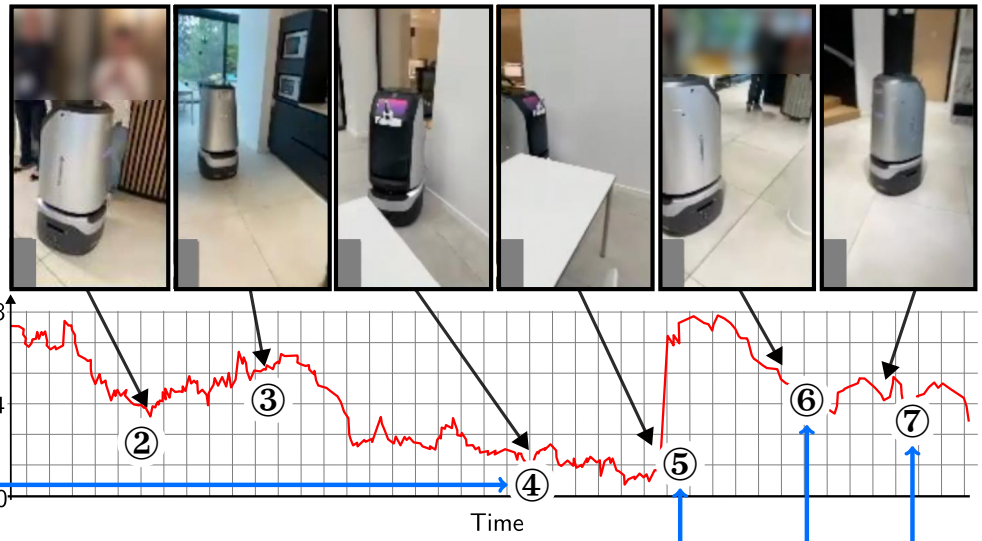
② Shortest path is blocked

People block the door through the goal. The agent needs to find another path.

③ Replan route...

Tries an alternative path south of the map, value estimate drops.





④ Oops, a dead end...

Route south turns out to be a dead end. The value function drops.

⑤ Replan again...

The agent abandon this strategy and re-plan north. Value estimate pikes, the agent anticipate positive reward.

⑥ / ⑦ Reach the goal

The agent tries the other door, and reaches the goal. Value function converges to 2.5, i.e. the final reward for a successful episode.