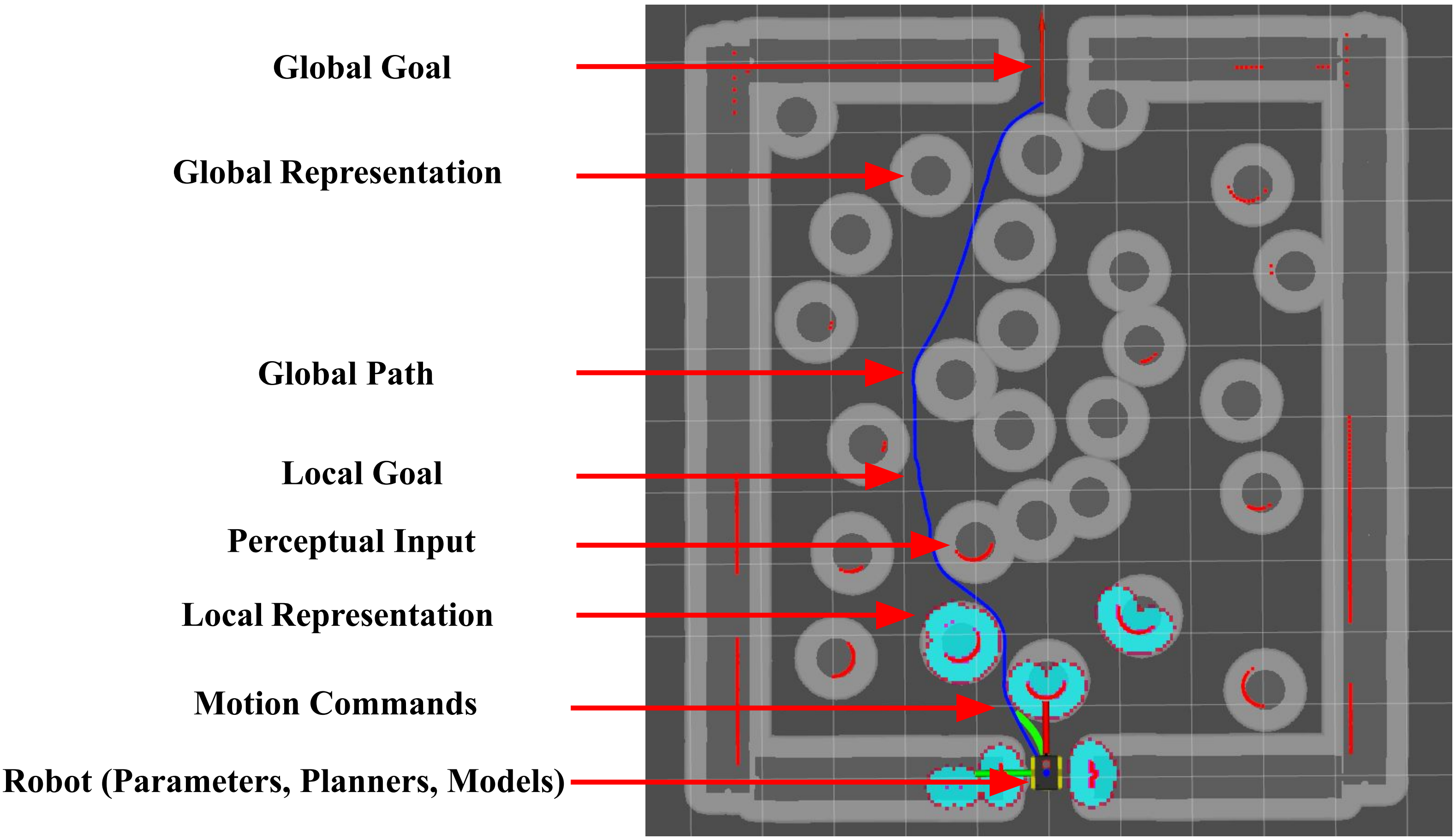
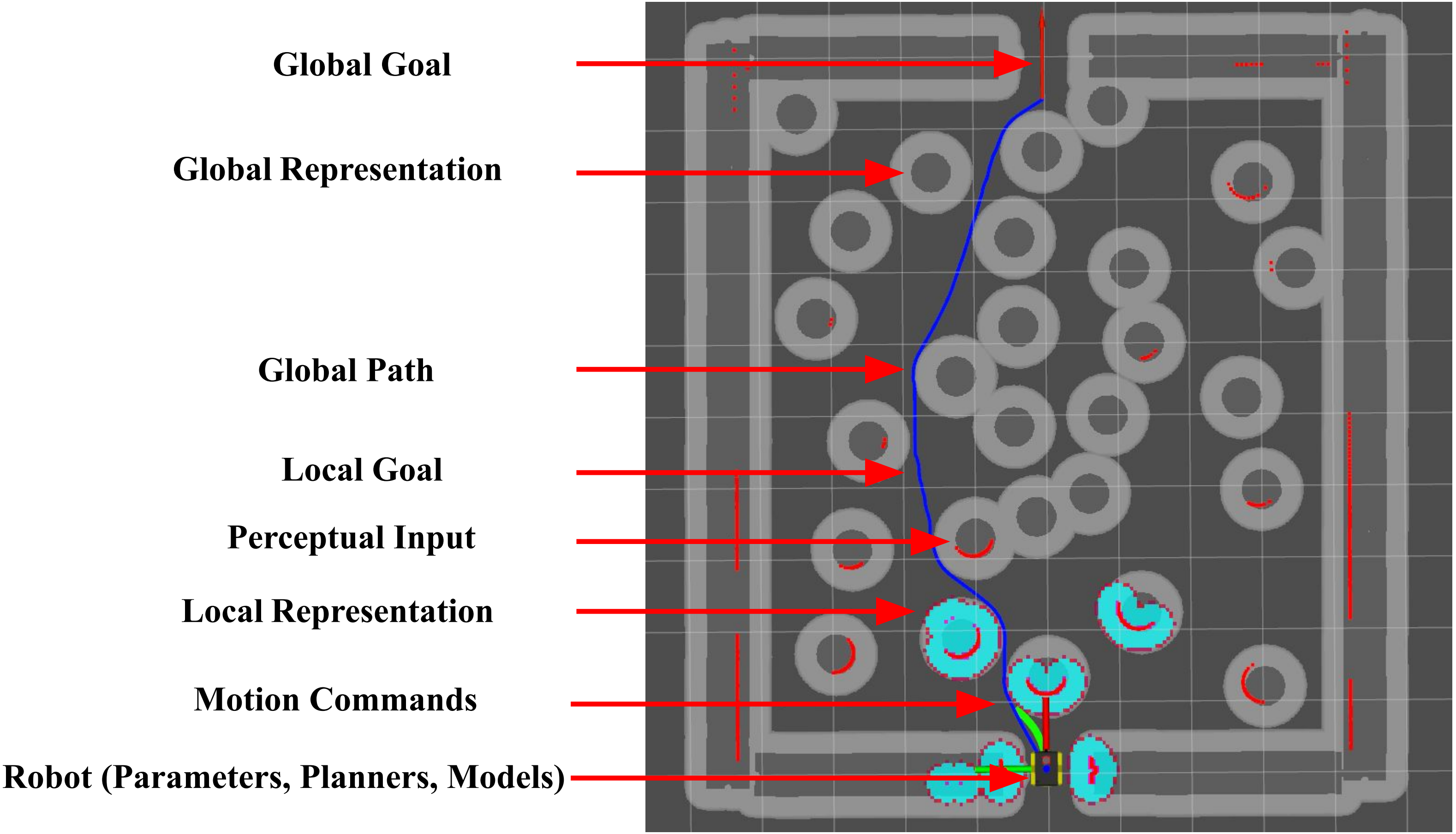
The classical navigation problem: moving a robot from one point to another without collision with any obstacle



[*Xiao et al., AuRo22*]

The classical navigation problem: moving a robot from one point to another without collision with any obstacle

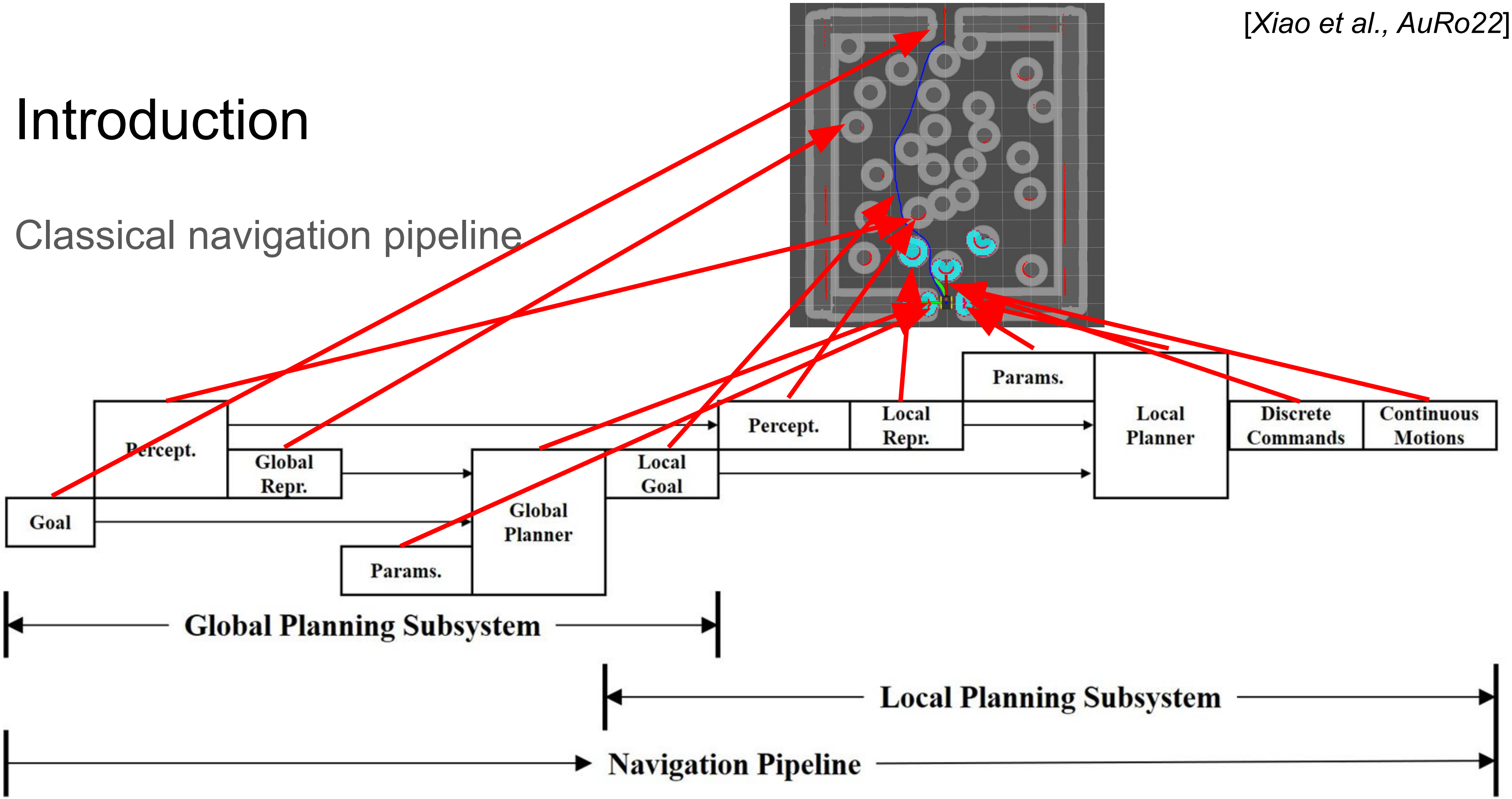


[*Xiao et al., AuRo22*]

[Xiao et al., AuRo22]

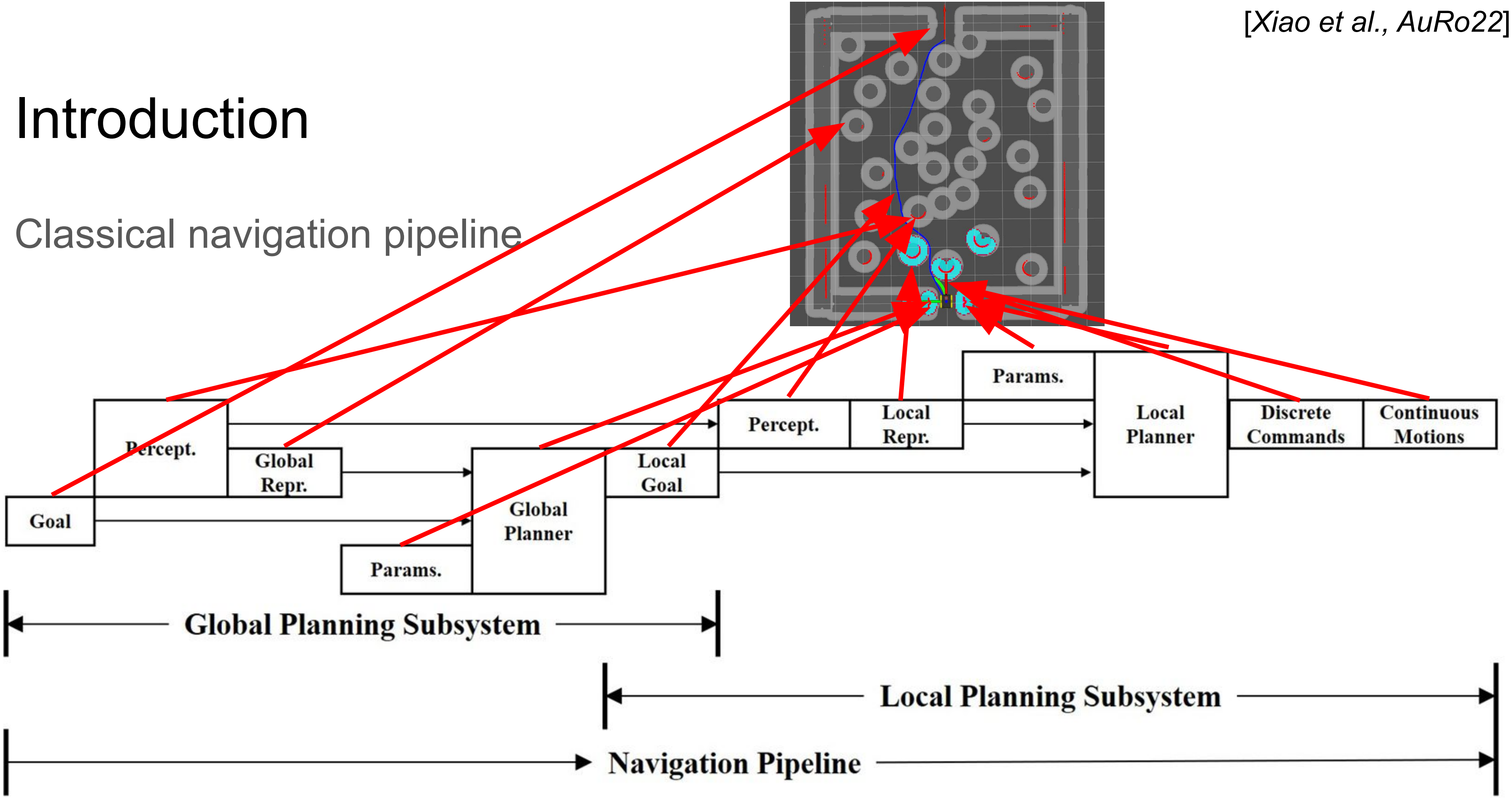
Introduction

Classical navigation pipeline



Introduction

Classical navigation pipeline



Classical Navigation vs. End-to-End Learning

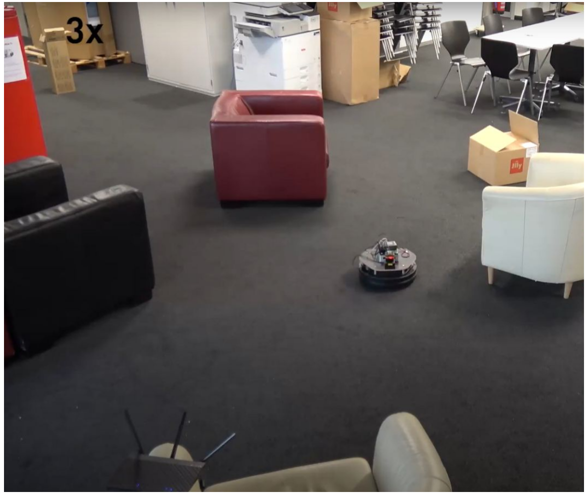
Strengths of classical navigation:

- Verifiable safety assurance
- Explainable components for debugging
- Generalizability to different scenarios

Weaknesses of classical navigation:

- Requires extensive engineering effort
- System performance won’t improve without manual intervention
- Propagation of errors through multiple components

Learning navigation in highly constrained spaces



vs.

Normal Environment
[*Pfeiffer et al., RA-L18*]

Highly-Constrained Environments
[*Xiao et al., RA-L21*]

Learning navigation in highly constrained spaces

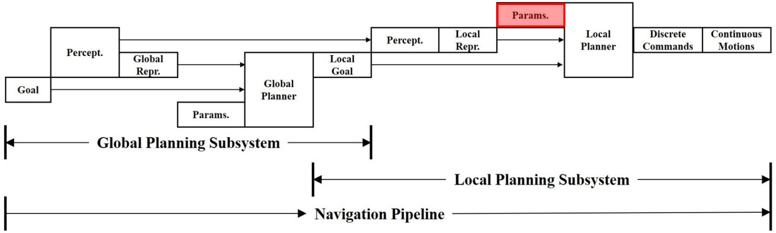
- Classical motion planners require **increased computation**.
  - Sampling-based methods require more samples to generate feasible motion. [*Kavraki et al., TRA96, Fox et al., RAM97, LaVelle, TechReport98*]
  - Trajectory-optimization-based methods require more optimization iterations. [*Quinlan et al., 93, Zucker et al., IJRR13, Zhou et al,. RA-L21*]
- Learning-based planners suffer from **lack of good-quality training data.**
  - Demonstration is difficult to acquire for Imitation Learning. [*Pfeiffer et al., ICRA17, Tai et al., IROS16*]
  - Trial-and-error is expensive for Reinforcement Learning. [*Tai et al., IROS17, Chiang et al., RA-L19*]

Challenge:
https://www.cs.utexas.edu/~xiao/BARN_Challenge/BARN_Challenge.html

Dataset:
https://www.cs.utexas.edu/~xiao/BARN/BARN.html

Learning navigation in highly constrained spaces

- Adaptive Planner Parameter Learning (APPL)
    - Learning local planners' parameters
    - Learning from non-expert humans using different interaction modalities



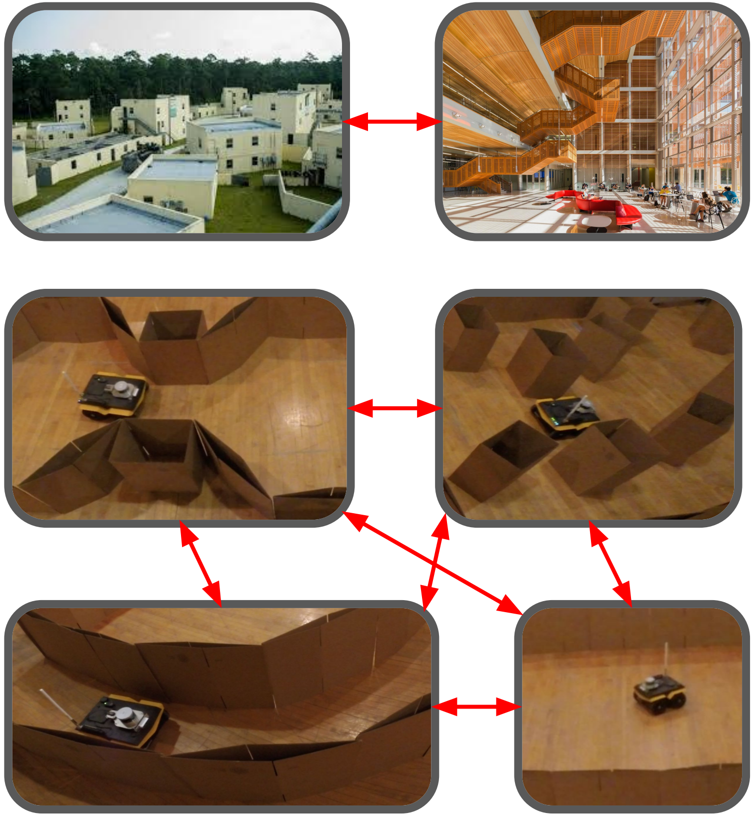
- Learning from Hallucination (LfH)
    - Learning a local planner
    - Learning from self-supervised experiences

Adaptive Planner Parameter Learning (APPL)

Robots need to face entirely different obstacle configurations.

Classical navigation systems require expert roboticists to fine-tune planner parameters.

max_vel_x: 0.5
min_vel_x: 0.1
max_vel_theta: 1.57
min_vel_theta: -1.57
vx_samples: 6
vtheta_samples: 20
occdist_scale: 0.1
pdist_scale: 0.75
gdist_scale: 1.0
Inflation_radius: 0.30
......

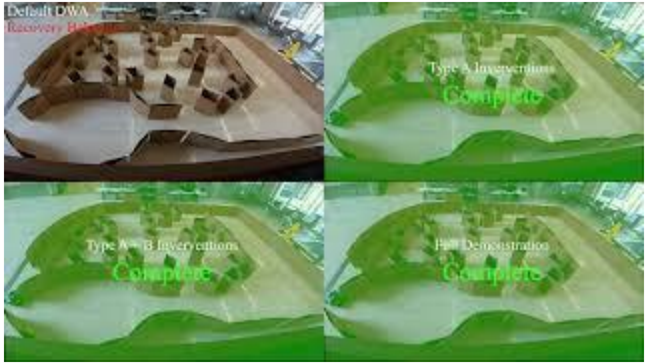


Adaptive Planner Parameter Learning (APPL)

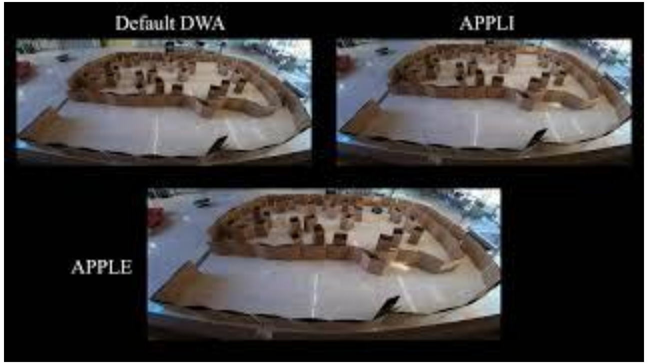
Inspiration: Most humans are not robotics experts, but they are navigation experts.

Teleoperated Demonstration

Corrective Interventions

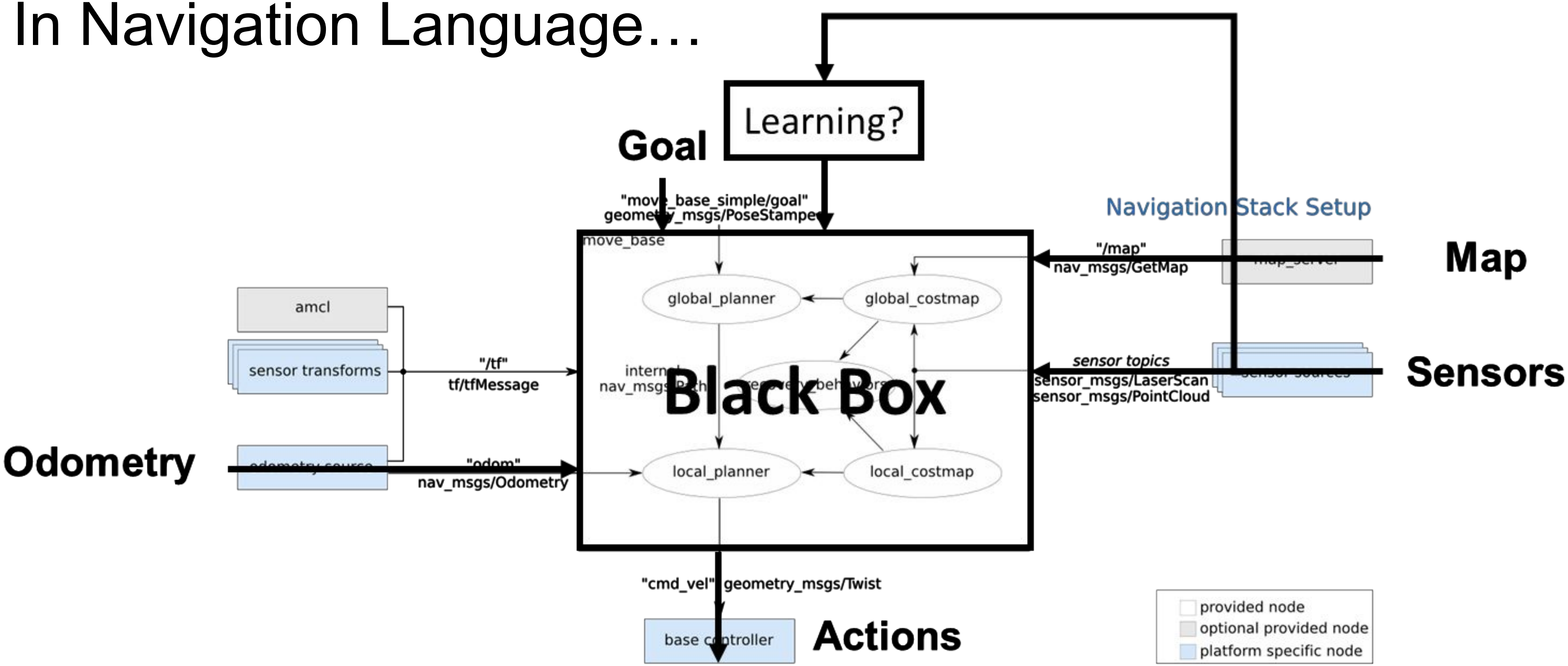


Evaluative Feedback



Reinforcement Learning

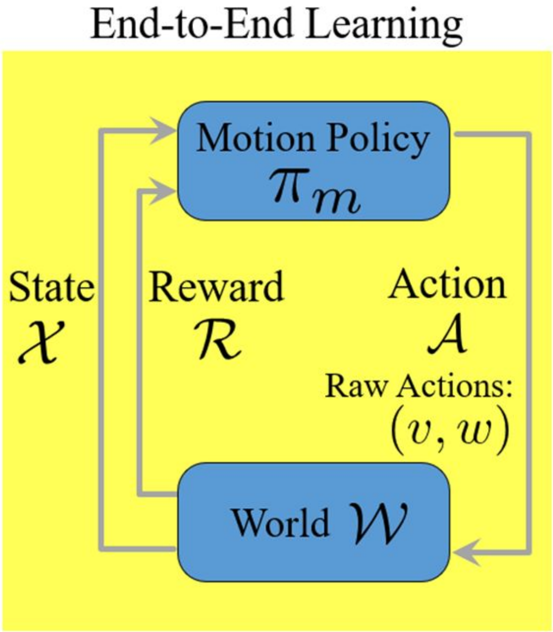
In Navigation Language…



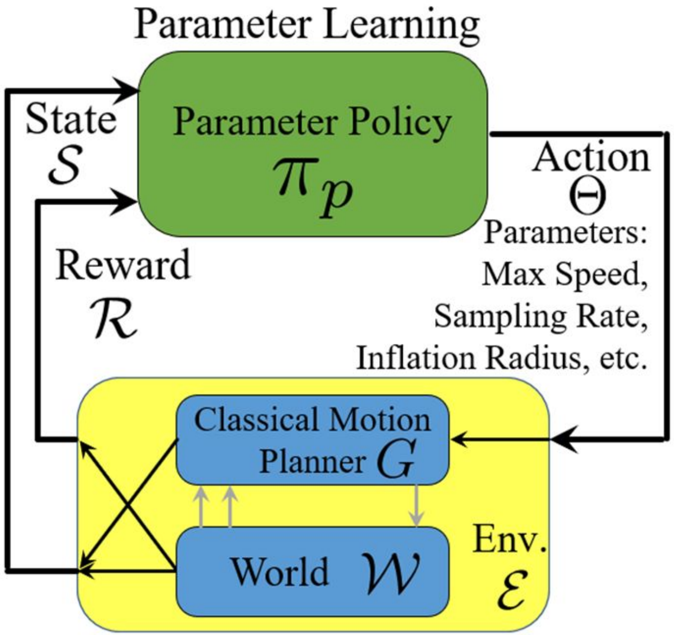
ROS **move_base** navigation stack

[*ROS move_base*]

In Learning Language…

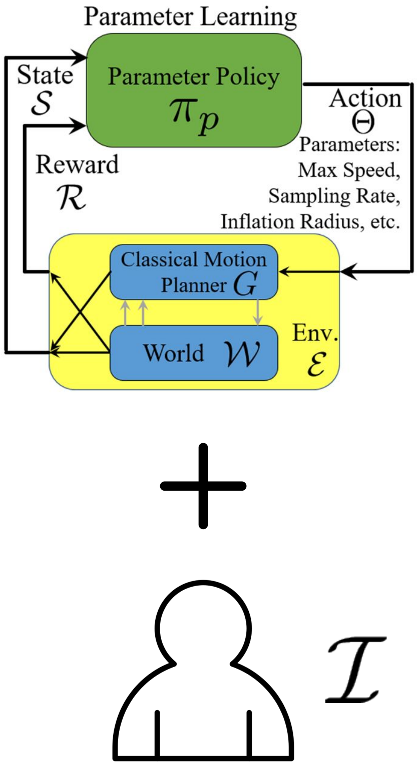


[*Gao et al., CoRL17, Pfeiffer et al., ICRA17, Chiang et al., RA-L19, Xiao et al., RA-L21*]



[*Xiao et al., RA-L20, Wang et al., ICRA21, Wang et al., RA-L21, Xu et al., ICRA21*]

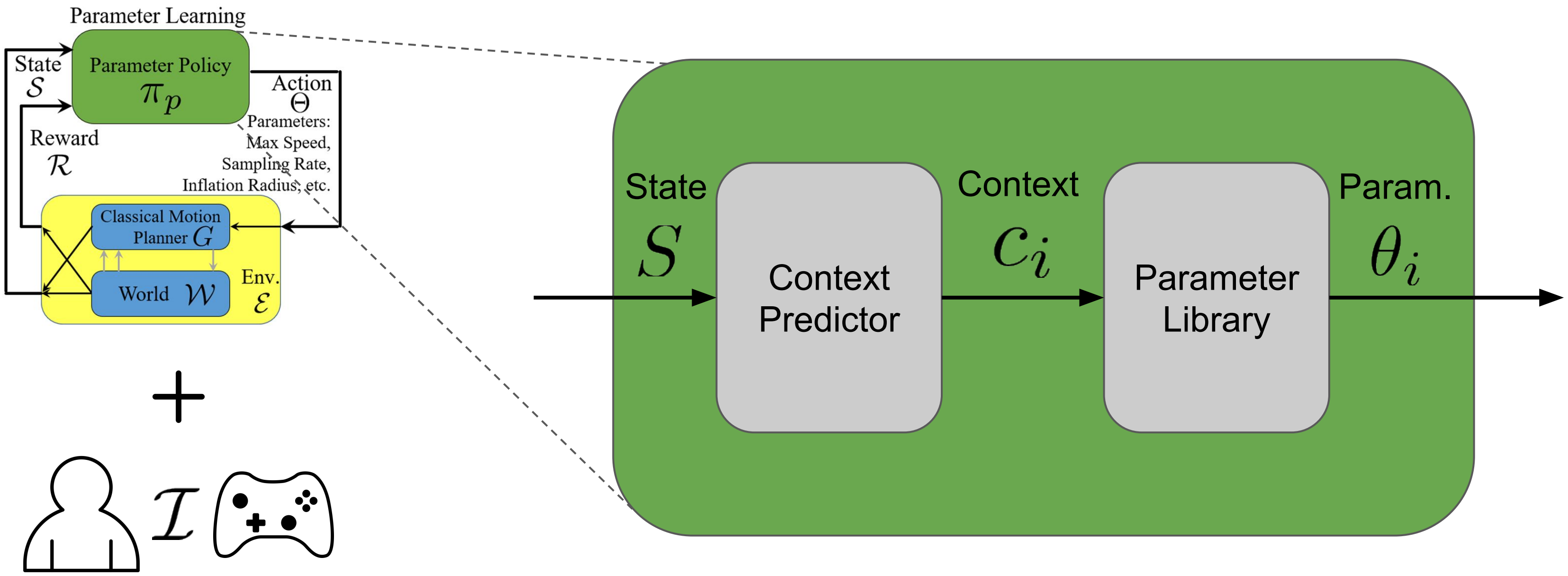
APPL from Human Interactions [*Xiao et al., RAS22*]




```
Algorithm 1 APPL
// Training
Input: human interaction 𝓘, space of possible parameters Θ, and navigation stack G.
π = LearnParameterPolicy(𝓘, Θ, G).
// Deployment
Input: navigation stack G, parameter policy π.
for t = 1 : T do
    construct meta-state s_t from x_t and θ_{t−1}.
    θ_t = π(s_t).
    Navigate with G_{θ_t}(x_t).
end for
```

Adaptive Planner Parameter Learning from Demonstration (APPLD) [*Xiao et al., RA-L20*]

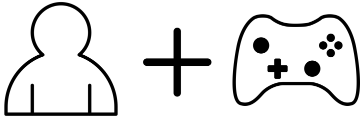
APPLD imposes an internal structure to the general parameter policy



Adaptive Planner Parameter Learning from Demonstration (APPLD) [*Xiao et al., RA-L20*]

Context Predictor:

1. Collect demonstration



$$\longrightarrow \mathcal{I} = \mathcal{D} = \{x_i, u_i\}_{i=1}^{N}$$

2. Perform automatic segmentation (e.g., using CHAMP [*Niekum et al. ICRA15*])

$$\{\mathcal{D}_k = \{x_i, u_i \,|\, \tau_{k-1} \leq i < \tau_k\}\}_{k=1}^{K}$$

3. Train online context predictor

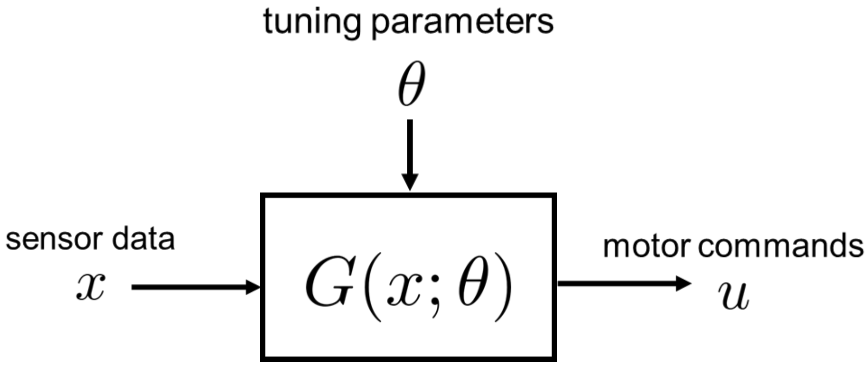
$$c_i = B_\phi(x_t)$$

$$\phi^* = \underset{\phi}{\operatorname{argmax}} \sum_{i=1}^{N} \log \frac{\exp\left(f_\phi(x_i^D)[c_i]\right)}{\sum_{c=1}^{K} \exp\left(f_\phi(x_i^D)[c]\right)}$$

$$B_\phi(x_t) = \operatorname{mode}\left\{ \underset{c}{\operatorname{argmax}}\, f_\phi(x_i)[c],\ t - p < i \leq t \right\}$$

Adaptive Planner Parameter Learning from Demonstration (APPLD) [*Xiao et al., RA-L20*]

Parameter Library: For each context, use behavior cloning to construct each element of the parameter library

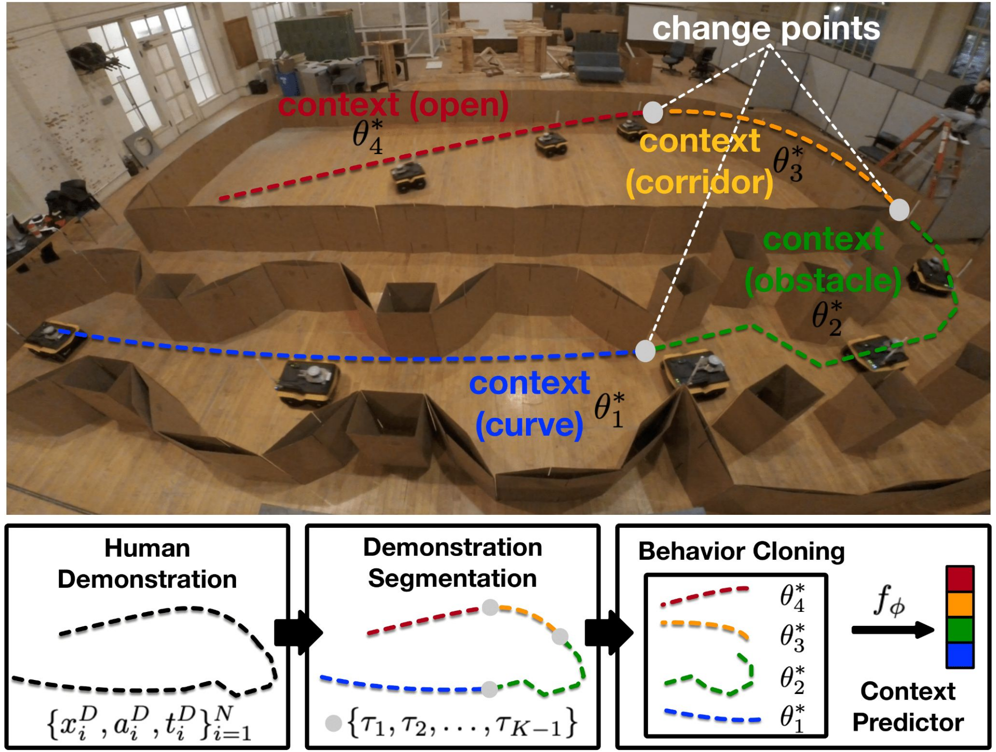


max_vel_x: 0.5
min_vel_x: 0.1
max_vel_theta: 1.57
min_vel_theta: -1.57
vx_samples: 6
vtheta_samples: 20
occdist_scale: 0.1
pdist_scale: 0.75
gdist_scale: 1.0
Inflation_radius: 0.30
......

Behavioral Cloning: Learn parameters from a demonstration using supervised learning.

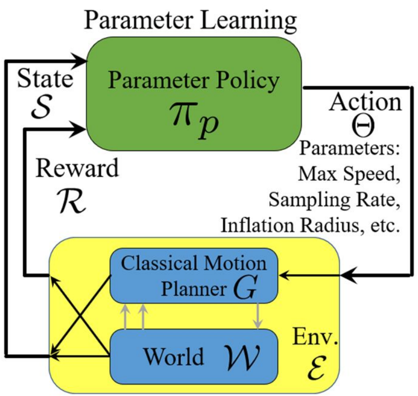
$$\theta^* = \arg\min_\theta \sum_i \ell(G(x_i;\theta), u_i)$$

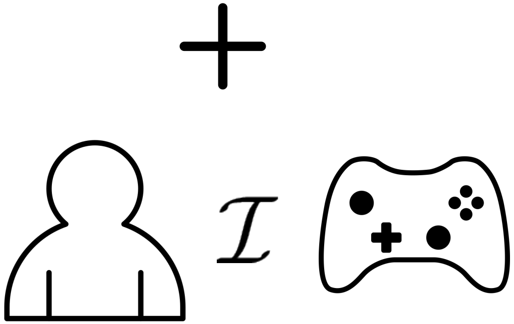
Adaptive Planner Parameter Learning from Demonstration (APPLD) [*Xiao et al., RA-L20*]



Adaptive Planner Parameter Learning from Demonstration (APPLD) [*Xiao et al., RA-L20*]

APPL from Human Interactions [*Xiao et al., RAS22*]





Algorithm 1 APPL

// Training

Input: human interaction $\mathcal{I}$, space of possible parameters Θ, and navigation stack G.

$\pi = LearnParameterPolicy(\mathcal{I}, \Theta, G)$.

// Deployment

Input: navigation stack G, parameter policy π.

for $t = 1 : T$ **do**

  construct meta-state s_t from x_t and θ_{t-1}.

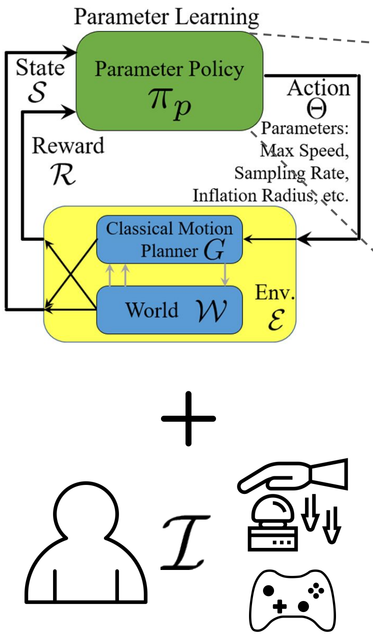
  $\theta_t = \pi(s_t)$.

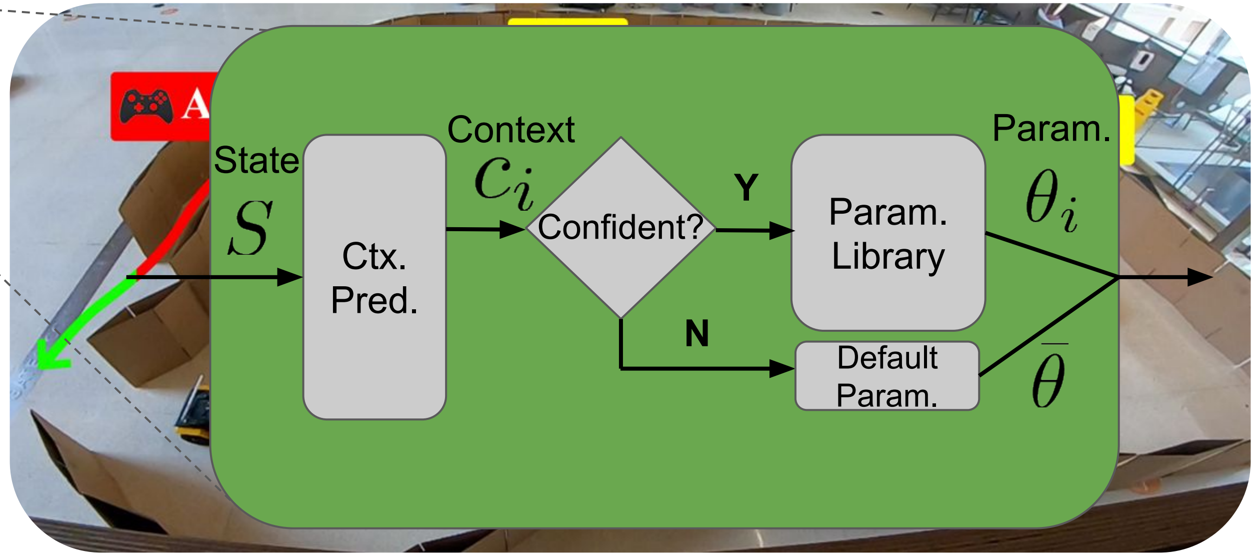
  Navigate with $G_{\theta_t}(x_t)$.

end for

Adaptive Planner Parameter Learning from Interventions (APPLI) [*Wang et al., ICRA21*]

Robots do not behave suboptimally everywhere: **Intervention** when necessary

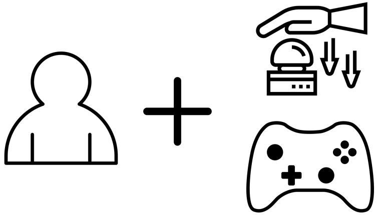




Adaptive Planner Parameter Learning from Interventions (APPLI) [*Wang et al., ICRA21*]

Context Predictor:

1. Collect (naturally segmented) interventions



$$\longrightarrow \mathcal{I} = I_{1:N} = \left\{ \{x_t, u_t\}_{t=1}^{T_i} \right\}_{1:N}$$

2. Train context predictor with Evidential Deep Learning (EDL)
[*Sensoy et al. NeurIPS18*]

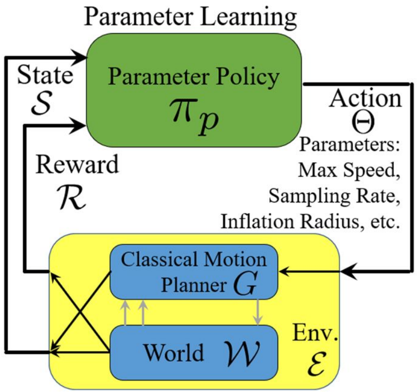
$$f_\phi(x_i) = (c_i, u_i) \qquad g_\phi(x_i) = c_i 1(u_i \geq \epsilon_u)$$

$$B_\phi(x_t) = \text{mode}\left\{ g_\phi(x_i),\ t - p < i \leq t \right\}$$

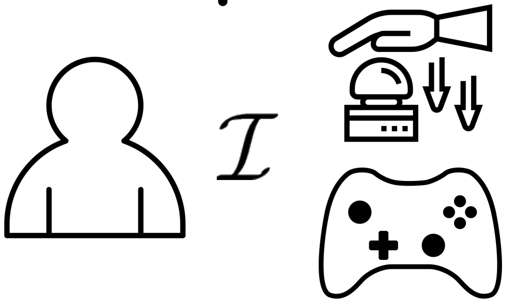
Parameter Library

3. Behavior clone parameters for each intervention $\theta^* = \arg\min_\theta \sum_i \ell(G(x_i;\theta), u_i)$

APPL from Human Interactions *[Xiao et al., RAS]*



+



Algorithm 1 APPL

// Training

Input: human interaction $\mathcal{I}$, space of possible parameters Θ, and navigation stack G.

$\boxed{\pi = \mathit{LearnParameterPolicy}(\mathcal{I}, \Theta, G).}$

// Deployment

Input: navigation stack G, parameter policy π.

for $t = 1 : T$ **do**

  construct meta-state s_t from x_t and θ_{t-1}.

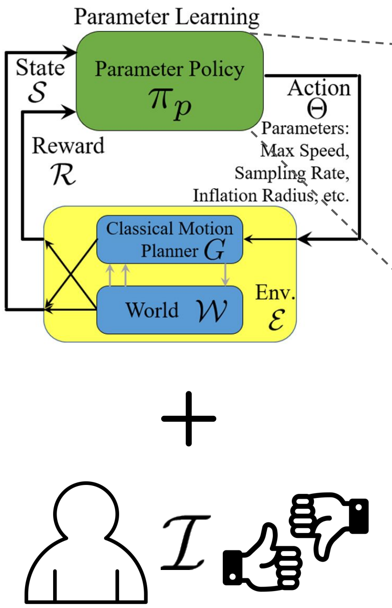
  $\theta_t = \pi(s_t)$.

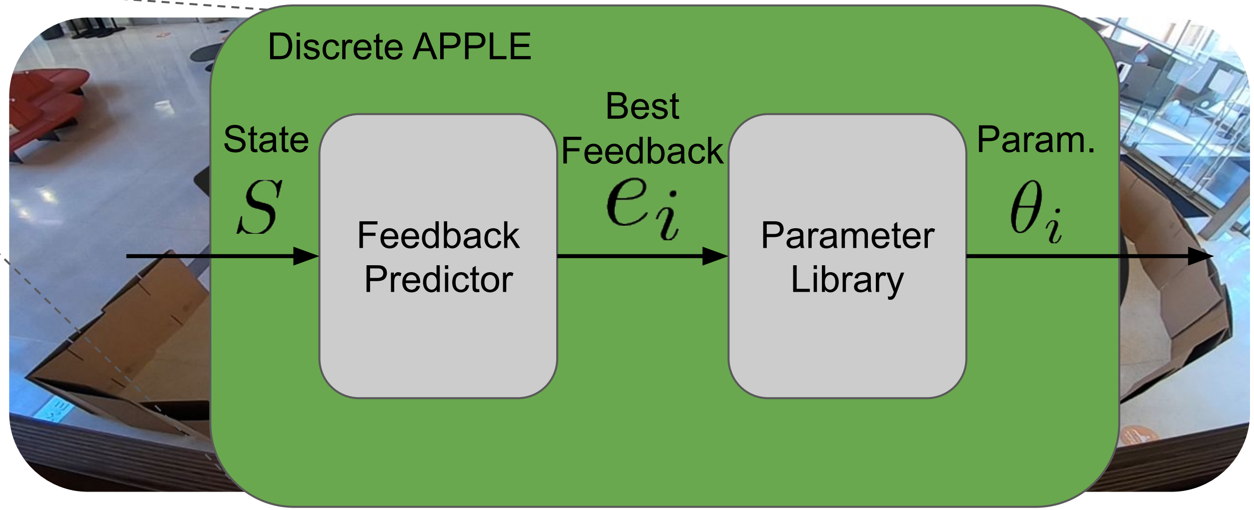
  Navigate with $G_{\theta_t}(x_t)$.

end for

Adaptive Planner Parameter Learning from Evaluative Feedback (APPLE) [*Wang et al., RA-L21*]

Non-expert users may not be able to take control of the robot: **Evaluative feedback**





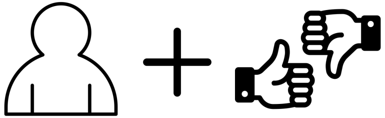
Discrete Adaptive Planner Parameter Learning from Evaluative Feedback (APPLE) [*Wang et al., RA-L21*]

Existing **Parameter Library** (from default, manually tuned, APPLD, APPLI, etc.)

$\mathcal{L}$

Feedback Predictor:

1. Collect feedback set



$$\longrightarrow \mathcal{I} = \mathcal{F} = \{x_j, \theta_j, e_j\}_{j=1}^{N} \ (\theta_j \in \mathcal{L})$$

2. Train feedback predictor

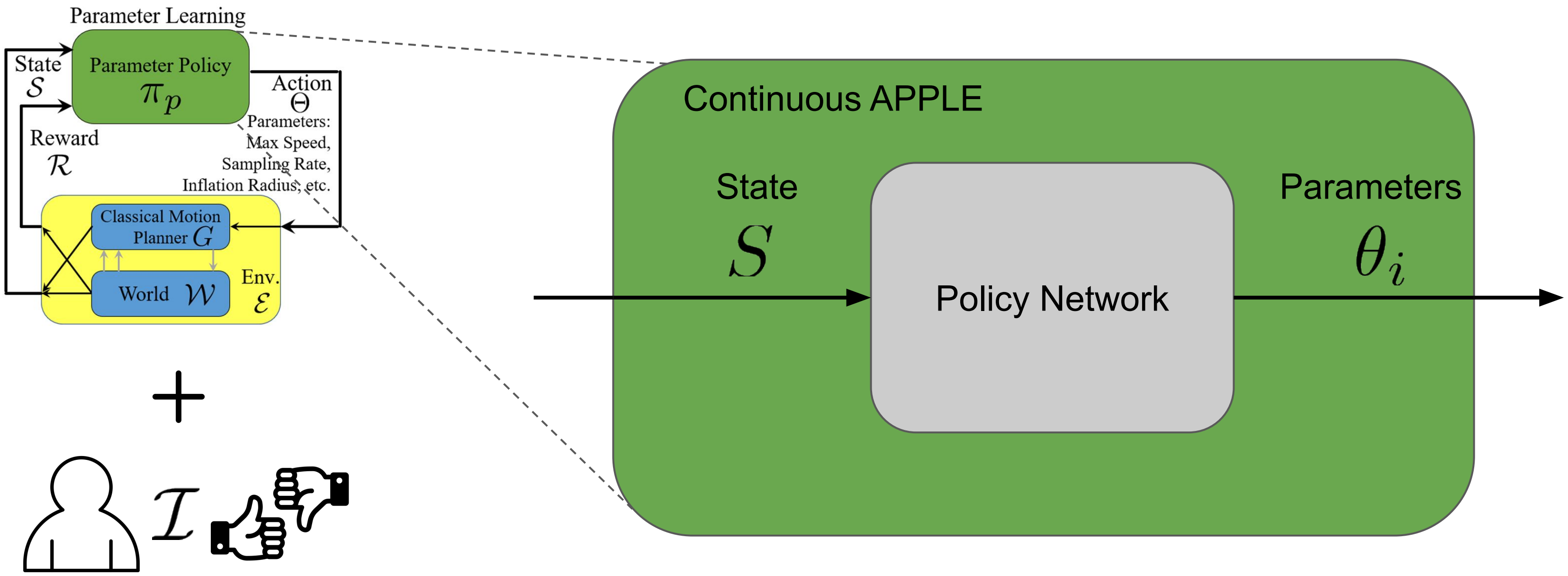
$$\phi^* = \underset{\phi}{\operatorname{argmin}} \underset{(x_j,\theta_j,e_j)\sim\mathcal{F}}{\mathbb{E}} \ell(F_\phi(x_j,\theta_j), e_j)$$

3. Deploy parameter policy

$$\pi(\cdot|x) = \underset{\theta\in\mathcal{L}}{\operatorname{argmax}} F_{\phi^*}(x,\theta)$$

Adaptive Planner Parameter Learning from Evaluative Feedback (APPLE) [*Wang et al., RA-L21*]

Non-expert users may not be able to take control of the robot: **Evaluative feedback**

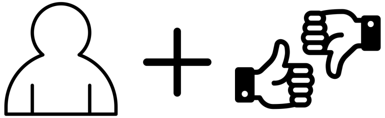


Continuous Adaptive Planner Parameter Learning from Evaluative Feedback (APPLE) [*Wang et al., RA-L21*]

Parameter Space instead of Parameter Library

Policy Network:

Train in actor-critic style



$$\longrightarrow \mathcal{I} = \mathcal{F} = \{x_j, \theta_j, e_j\}_{j=1}^{N}$$

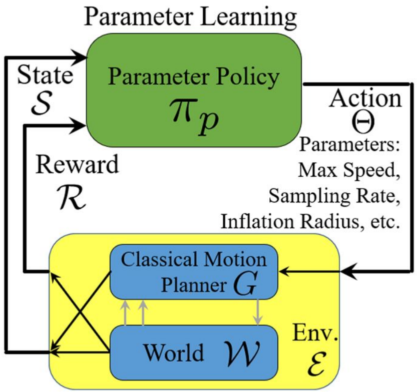
Critic

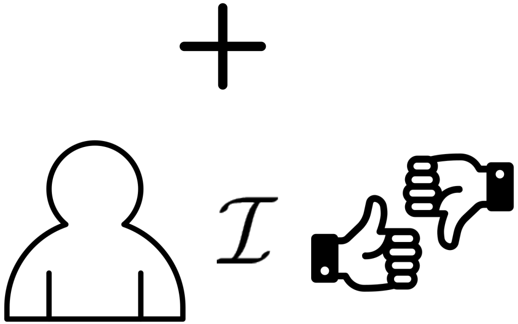
$$\phi^* = \underset{\phi}{\operatorname{argmin}} \underset{(x_j, \theta_j, e_j) \sim \mathcal{F}}{\mathbb{E}} \ell(F_\phi(x_j, \theta_j), e_j)$$

Actor

$$\psi^* = \underset{\psi}{\operatorname{argmin}} \underset{\substack{x_j \in \mathcal{F} \\ \tilde{\theta}_j \sim \pi_\psi(\cdot|x_j)}}{\mathbb{E}} \left[-F_\phi(x_j, \tilde{\theta}_j) + \alpha \log \pi_\psi(\tilde{\theta}_j | x_j) \right]$$

APPL from Human Interactions *[Xiao et al., RAS22]*



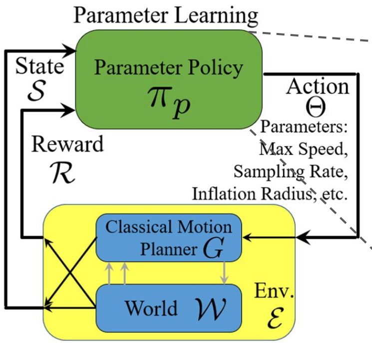


Algorithm 1 APPL

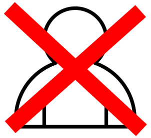
// Training
Input: human interaction $\mathcal{I}$, space of possible parameters Θ, and navigation stack G.
$\pi = LearnParameterPolicy(\mathcal{I}, \Theta, G)$.
// Deployment
Input: navigation stack G, parameter policy π.
for $t = 1 : T$ **do**
    construct meta-state s_t from x_t and θ_{t-1}.
    $\theta_t = \pi(s_t)$.
    Navigate with $G_{\theta_t}(x_t)$.
end for

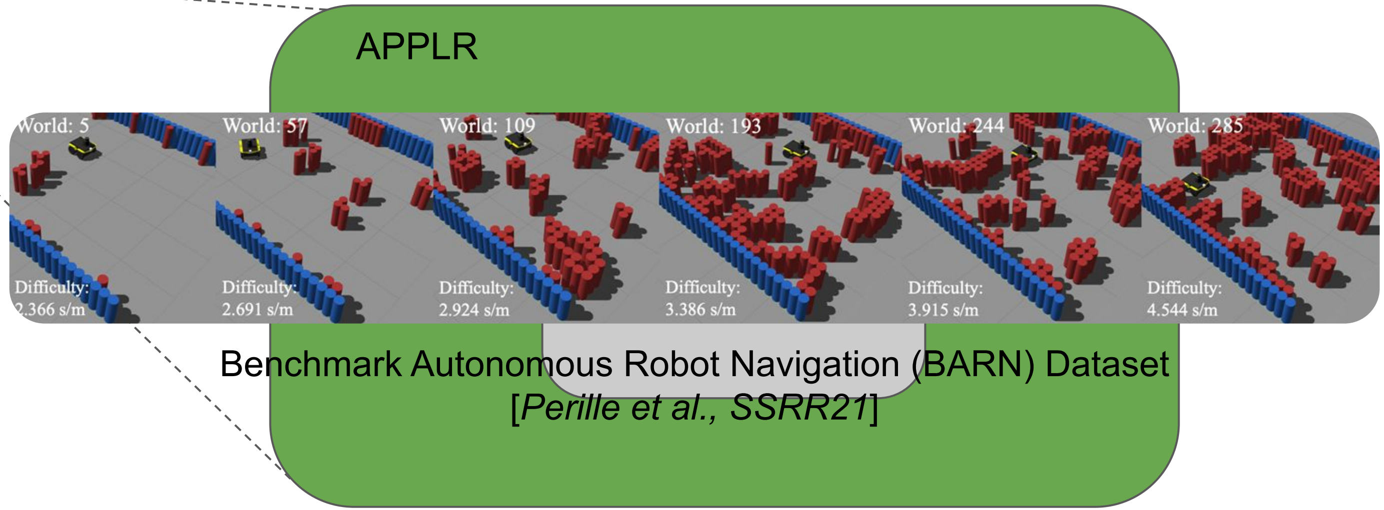
Adaptive Planner Parameter Learning from Reinforcement (APPLR) [*Xu et al., ICRA21*]

What about no humans at all? **Reinforcement Learning**



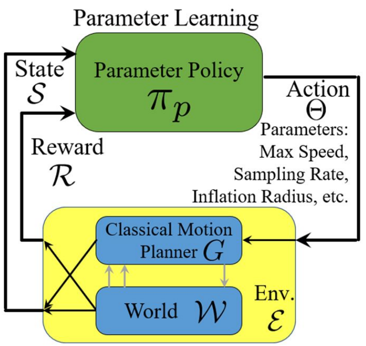
+





Benchmark Autonomous Robot Navigation (BARN) Dataset [*Perille et al., SSRR21*]

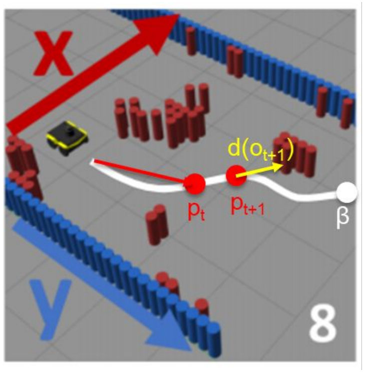
Adaptive Planner Parameter Learning from Reinforcement (APPLR) [*Xu et al., ICRA21*]



$$
\begin{aligned}
\text{State} \quad & \mathcal{S}: s_t = (o_t, \phi_t, \theta_{t-1}) \\
\text{Action} \quad & \mathcal{A}: a_t = \theta_t \\
\text{Reward} \quad & \mathcal{R}: R_t(s_t, a_t, s_{t+1}) \\
\text{Transition} \quad & \mathcal{T}: o_{t+1}, \phi_{t+1} \sim \mathcal{T}(\cdot | s_t, \theta_t)
\end{aligned}
$$

$$
\begin{aligned}
\text{Sensory input} \quad & \mathcal{O}: o_t \\
\text{Local goal} \quad & \mathcal{G}: \phi_t \\
\text{Parameter set} \quad & \Theta: \theta_t
\end{aligned}
$$

Optimization Objective: $\max_{\pi} J^{\pi} = \mathbb{E}_{s_0, \theta_t \sim \pi(s_t), s_{t+1} \sim \mathcal{T}(s_t, \theta_t)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$

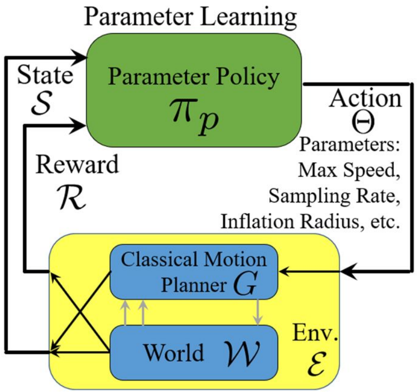


Reward Design: $R_t(s_t, a_t, s_{t+1}) = R_f + 0.5 R_p + 0.05 R_c$

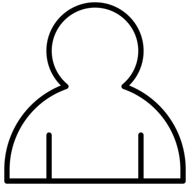
$$R_f(s_t, a_t) = \mathbb{1}(s_t \text{ is terminal}) - 1$$

$$R_p = \frac{(p_{t+1} - p_t) \cdot (\beta - p_t)}{|\beta - p_t|} \qquad R_c = -1/d(o_{t+1})$$

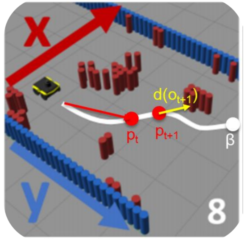
APPL from Human Interactions [*Xiao et al., RAS22*]



+



$\mathcal{I}$



Algorithm 1 APPL

// Training

Input: human interaction $\mathcal{I}$, space of possible parameters Θ, and navigation stack G.

$\pi = LearnParameterPolicy(\mathcal{I}, \Theta, G)$.

// Deployment

Input: navigation stack G, parameter policy π.

for $t = 1 : T$ **do**

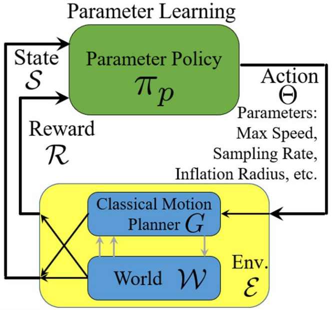
  construct meta-state s_t from x_t and θ_{t-1}.

  $\theta_t = \pi(s_t)$.

  Navigate with $G_{\theta_t}(x_t)$.

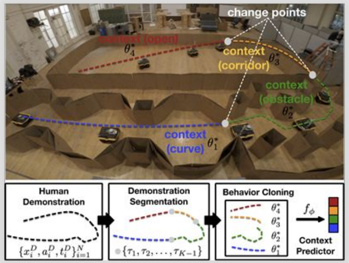
end for

Cycle-of-Learning from APPL [*Xiao et al., RAS22*]



[*Xu et al., ICRA21*]

APPLD



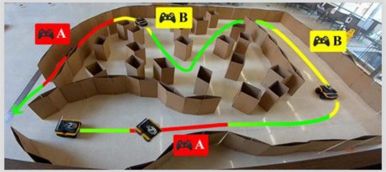
[*Xiao et al., RA-L20*]

APPLR

APPLI

[*Wang et al., RA-L21*]

APPLE



[*Wang et al., ICRA21*]

References

Xiao, Xuesu, Bo Liu, Garrett Warnell, and Peter Stone. "Motion planning and control for mobile robot navigation using machine learning: a survey." Autonomous Robots (2022): 1-29.

Gao, Wei, David Hsu, Wee Sun Lee, Shengmei Shen, and Karthikk Subramanian. "Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation." In Conference on Robot Learning, pp. 185-194. PMLR, 2017.

Pfeiffer, Mark, Michael Schaeuble, Juan Nieto, Roland Siegwart, and Cesar Cadena. "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots." In 2017 ieee international conference on robotics and automation (icra), pp. 1527-1533. IEEE, 2017.

Chiang, Hao-Tien Lewis, Aleksandra Faust, Marek Fiser, and Anthony Francis. "Learning navigation behaviors end-to-end with autorl." IEEE Robotics and Automation Letters 4, no. 2 (2019): 2007-2014.

Xiao, Xuesu, Bo Liu, Garrett Warnell, and Peter Stone. "Toward agile maneuvers in highly constrained spaces: Learning from hallucination." IEEE Robotics and Automation Letters 6, no. 2 (2021): 1503-1510.

Xiao, Xuesu, Bo Liu, Garrett Warnell, Jonathan Fink, and Peter Stone. "APPLD: Adaptive planner parameter learning from demonstration." IEEE Robotics and Automation Letters 5, no. 3 (2020): 4541-4547.

Wang, Zizhao, Xuesu Xiao, Bo Liu, Garrett Warnell, and Peter Stone. "APPLI: Adaptive planner parameter learning from interventions." In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 6079-6085. IEEE, 2021.

Wang, Zizhao, Xuesu Xiao, Garrett Warnell, and Peter Stone. "APPLE: Adaptive planner parameter learning from evaluative feedback." IEEE Robotics and Automation Letters 6, no. 4 (2021): 7744-7749.

Xu, Zifan, Gauraang Dhamankar, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Bo Liu, Zizhao Wang, and Peter Stone. "APPLR: Adaptive planner parameter learning from reinforcement." In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 6086-6092. IEEE, 2021.

Niekum, Scott, Sarah Osentoski, Christopher G. Atkeson, and Andrew G. Barto. "Online bayesian changepoint detection for articulated motion models." In 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1468-1475. IEEE, 2015.

Sensoy, Murat, Lance Kaplan, and Melih Kandemir. "Evidential deep learning to quantify classification uncertainty." arXiv preprint arXiv:1806.01768 (2018).