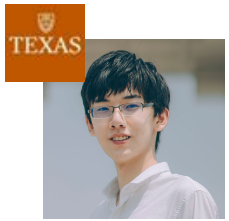
APPLE: Adaptive Planner Parameter Learning from Evaluative Feedback

Zizhao Wang[1], Xuesu Xiao[1], Garrett Warnell[2], and Peter Stone[1, 3]




[1]The University of Texas at Austin
[2]Army Research Laboratory
[3]Sony AI

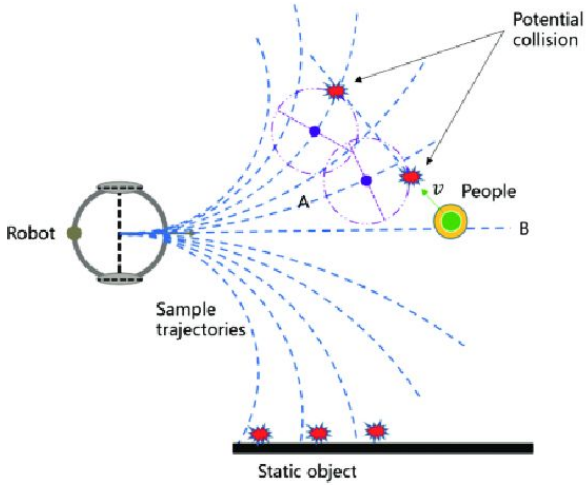
Motivation

Deploying a classical autonomous navigation system in new environments requires adaptivity by tuning the parameters.

Motivation

Deploying a classical autonomous navigation system in new environments requires adaptivity by tuning the parameters.

Otherwise, it may produce suboptimal behaviors or even fail.



Dynamic Window Approach (DWA) planner
- max velocity
- max angular velocity
- number of velocity samples
- number of angular velocity samples
- weight for avoiding obstacles
- weight for staying close to global path
- weight for staying close to local goal
- ...

Dieter, et al., *The Dynamic Window Approach to Collision Avoidance*, Robotics & Automation Magazine, 1997

Motivation

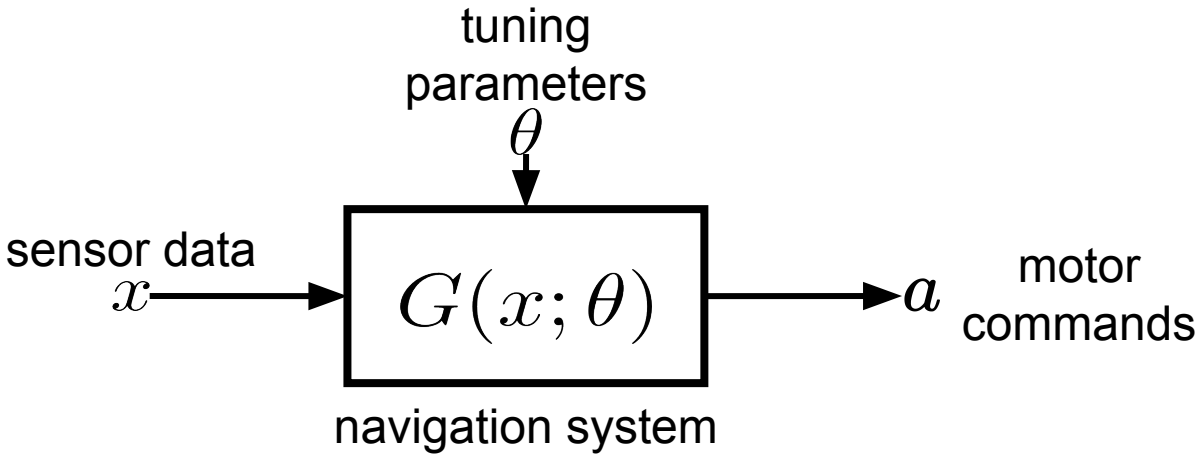
Manually re-tuning those parameters requires expert knowledge.

However, it’s easy for even non-expert to provide evaluative feedback.

Background

APPL: Adaptive Planner Parameter Learning

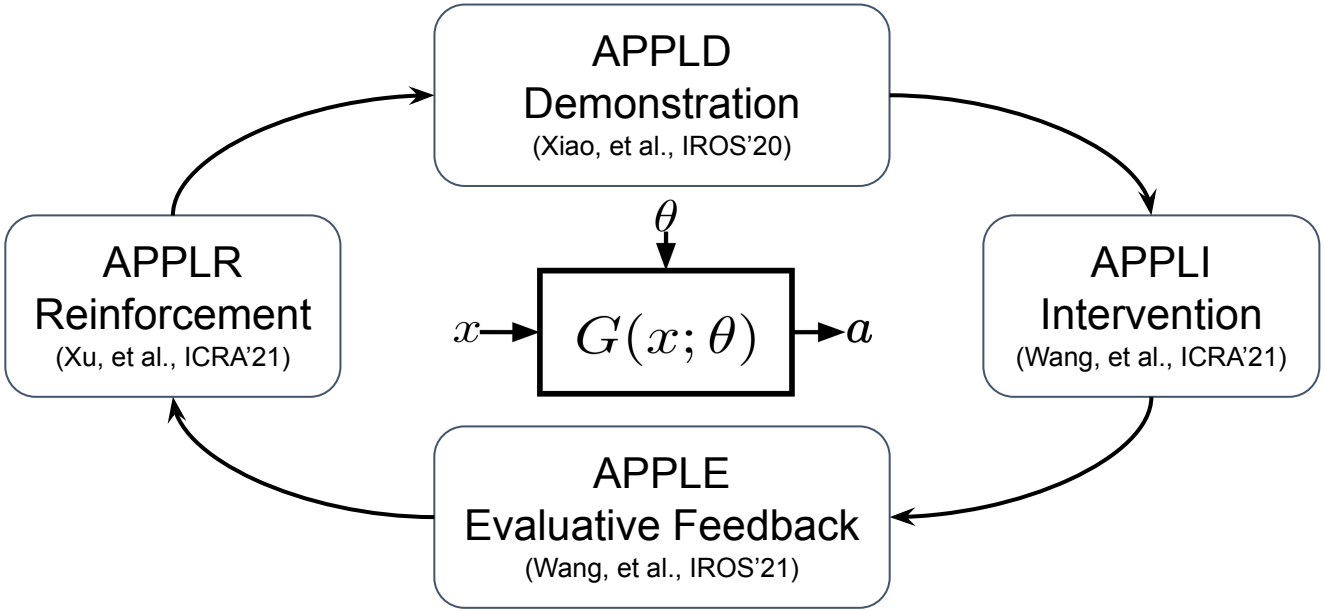
- Use different human interactions to “tune” any navigation system.



Background

APPL: Adaptive Planner Parameter Learning

- Use different human interactions to “tune” any navigation system.



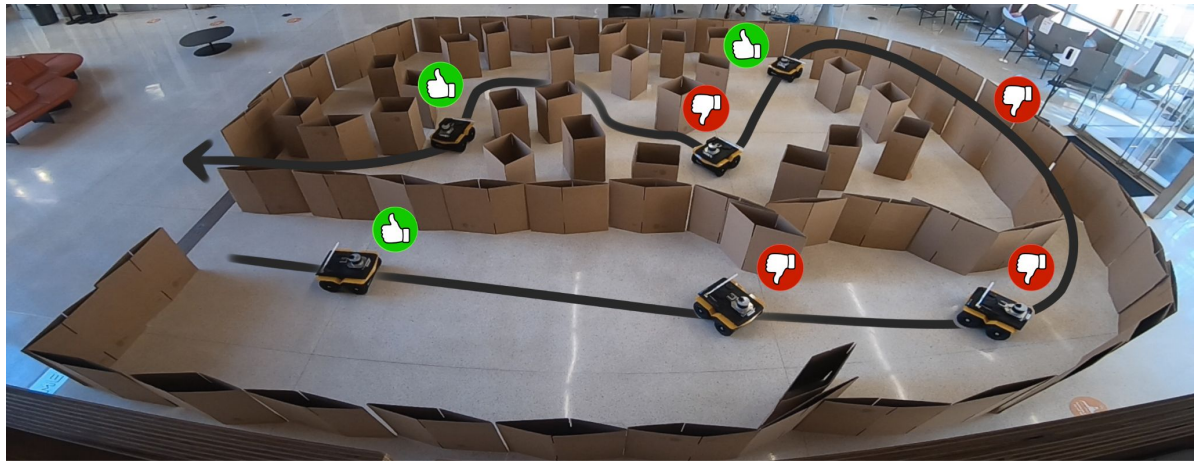
Cycle of Learning

(Xiao, et al., *APPL: Adaptive Planner Parameter Learning*, 2021)

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Contributions

- Compared to demonstrations/interventions that require users to take control of the robot, evaluative feedback is easier to collect.
- APPLE outperforms APPLD/I by selecting the planner parameters based on a performance-based metric.



APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Procedures (at every time step)

1. Collect discrete/continuous evaluative feedback.

$$\longrightarrow (x_t, \theta_t, e_t)$$

2. Train the feedback predictor F_ϕ with feedback collected so far.
   - Minimize cross entropy loss for discrete feedback.
   - Minimize mean squared error for continuous feedback.

$$(x_t, \theta_t) \xrightarrow{F_\phi} \hat{e}_t$$

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

3. Train the discrete/continuous parameter policy π_ψ.
   - Assumption: human will not only consider current results but also future consequences when giving feedback.
   - Maximizing the current feedback is enough.

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

3. Train the discrete/continuous parameter policy π_ψ.
   - Assumption: human will not only consider current results but also future consequences when giving feedback.
   - Maximizing the current feedback is enough.
   - Discrete parameter policy: select from a parameter library $\mathcal{L} = \{\theta^i, \ldots, \theta^K\}$.
     - DQN style

$$\pi(\cdot|x) = \arg\max_{\theta \in \mathcal{L}} F_\phi(x, \theta)$$

   - Continuous parameter policy: select from continuous parameter spaces.
     - Soft actor-critic style

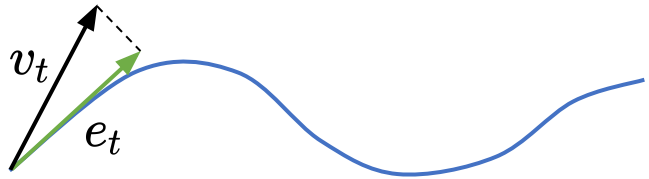
$$\psi^* = \arg\max_\psi \mathop{\mathbb{E}}_{\tilde{\theta} \sim \pi_\psi(\cdot|x)} \left[F_\phi(x, \tilde{\theta}) - \alpha \log \pi_\psi(\tilde{\theta}|x) \right]$$

entropy regularization
to improve exploration

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Setup

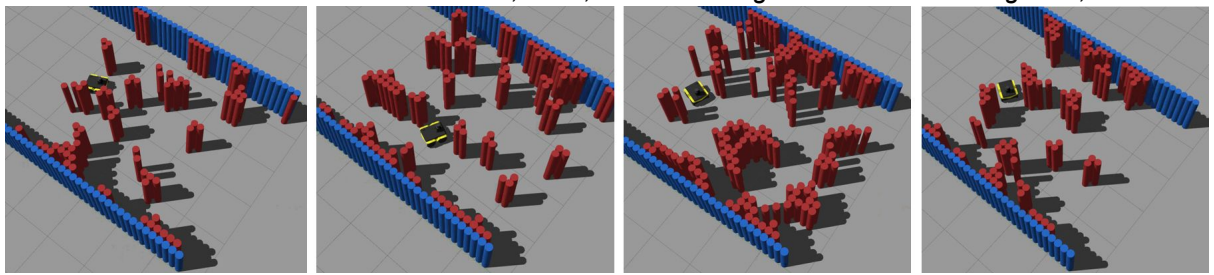
Robot: Clearpath Jackal (Velodyne Puck lidar)



Feedback: (suboptimal) simulated feedback

- Velocity along the local path (drive as fast as possible)
- Discretized to **different numbers of levels**
- Provided at 1 Hz

Perille, et al., *Benchmarking Metric Ground Navigation*, SSRR'20

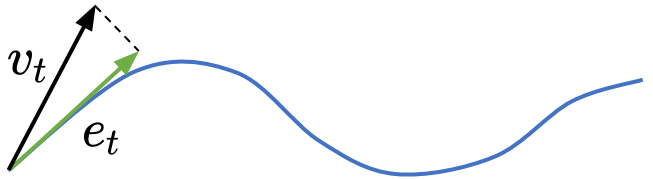


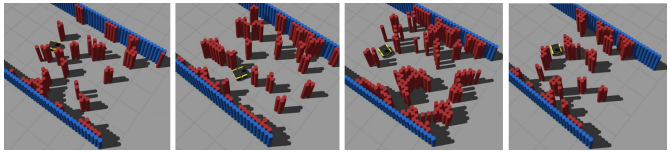
Environment: 250 for training, 50 for testing

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Questions to Answer

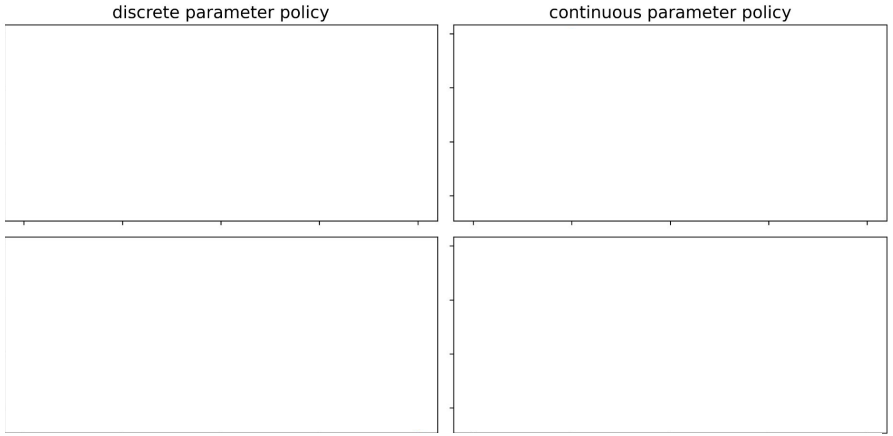
- Performance at different numbers of feedback levels
- APPLE vs baselines
  - Baseline 1: planner with default parameters
  - Baseline 2: APPL-Intervention (richer interaction modality)
  - Baseline 3: APPL-Reinforcement (greatest capacity)
  - To test its performance limits, APPLE is trained with 2.5M feedback signals.
  - Will show APPLE is still practical with physical experiments soon.
- Continuous APPLE vs discrete APPLE.
  - Will continuous policy work well with low-resolution feedback?





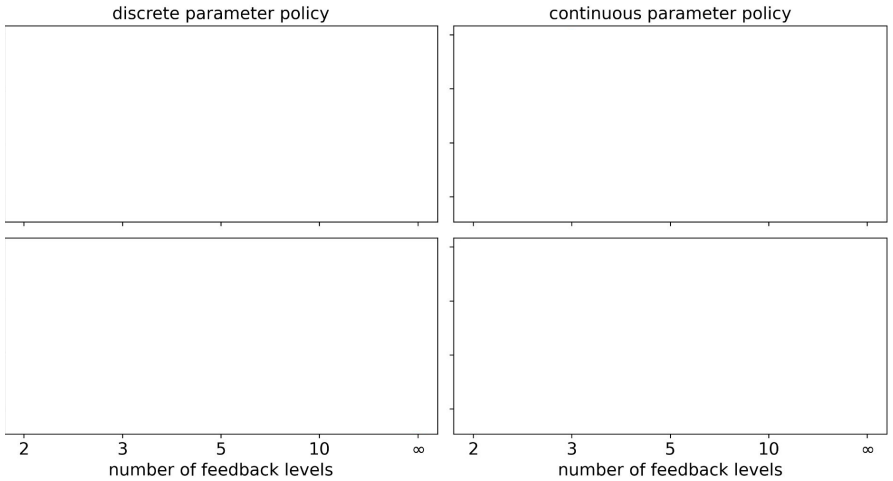
APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Results



APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Results

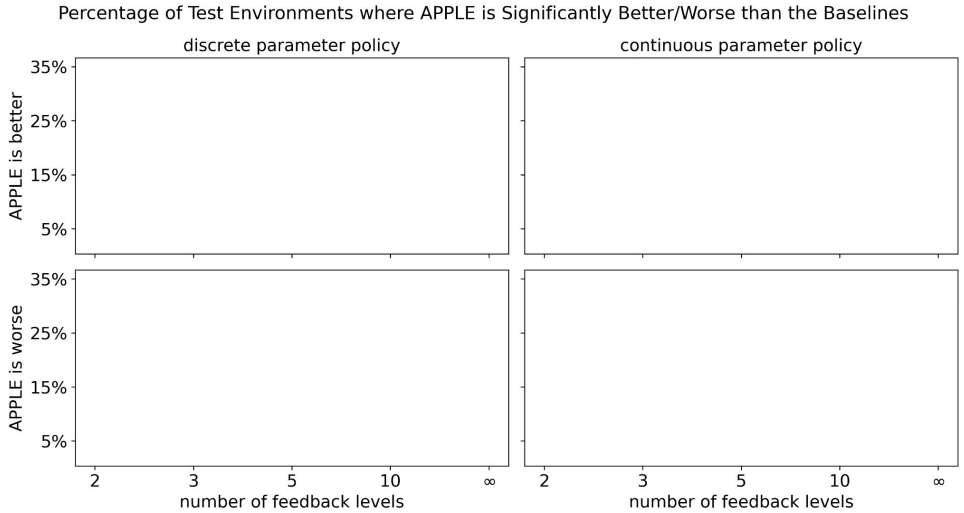


Implementation details

∞ feedback levels mean using continuous feedback.

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Results



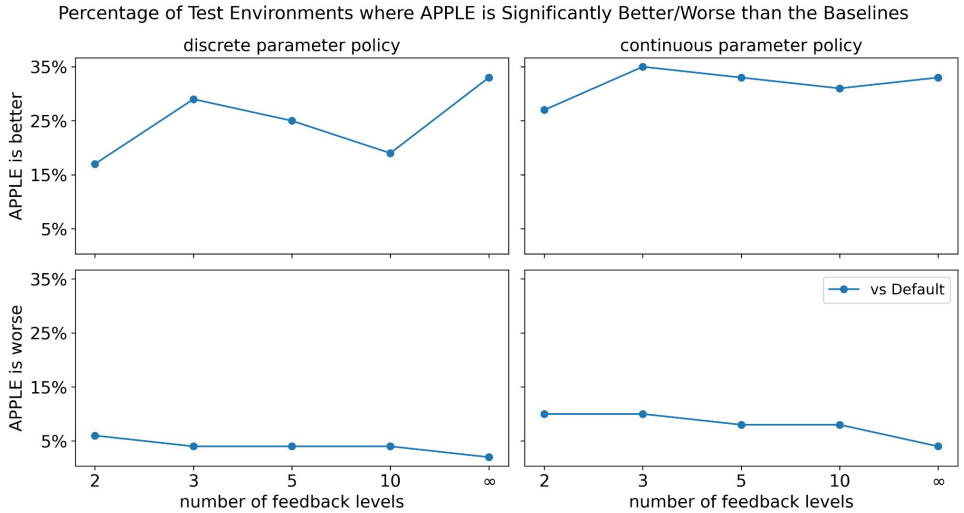
Implementation details

∞ feedback levels mean using continuous feedback.

Significantly better/worse: t-test on 20 runs per environment with $p < 0.05$.

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Results



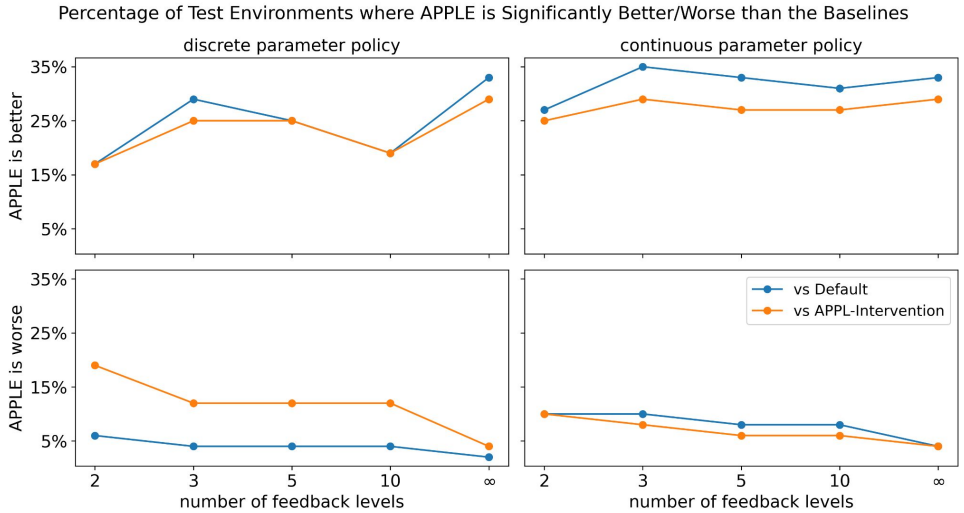
Implementation details

∞ feedback levels mean using continuous feedback.

Significantly better/worse: t-test on 20 runs per environment with $p < 0.05$.

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Results



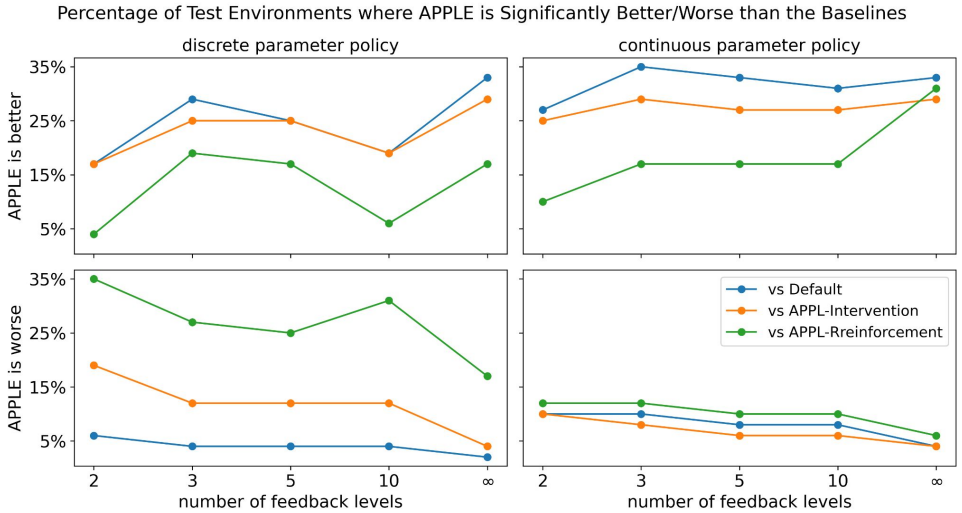
Implementation details

∞ feedback levels mean using continuous feedback.

Significantly better/worse: t-test on 20 runs per environment with $p < 0.05$.

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Simulated Experiment - Results



Implementation details

∞ feedback levels mean using continuous feedback.

Significantly better/worse: t-test on 20 runs per environment with $p < 0.05$.

- Feedback resolution
  Three feedback levels are good enough, and it is easy to collect.
- APPLE outperforms APPLI.
  Select parameter based on performance (feedback) is better than based on similarity with intervened environments.
- Continuous APPLE outperforms discrete APPLE.
  Because continuous APPLE has larger capacity in the parameter space.

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

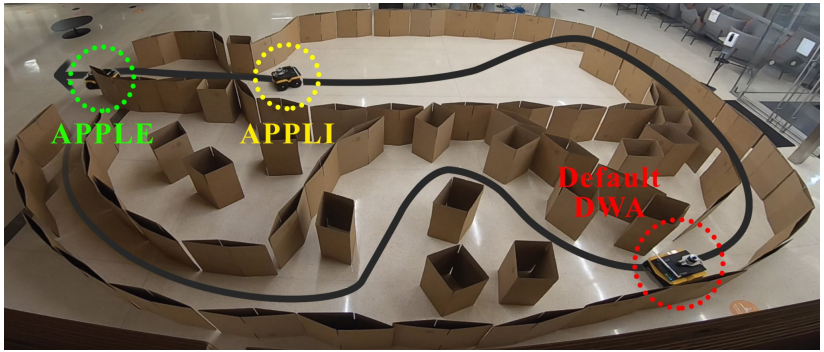
Physical Experiment Setup

Robot: Clearpath Jackal (Velodyne Puck lidar)

Human Supervisor: an author providing feedback via Xbox controller

- Providing binary feedback (“good job” or “bad job”)
- Only need to give “bad job” feedback, otherwise assumed “good job” is given.
- Feedback given at 2 Hz for 30 min

Training Environment



Unseen Environment

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

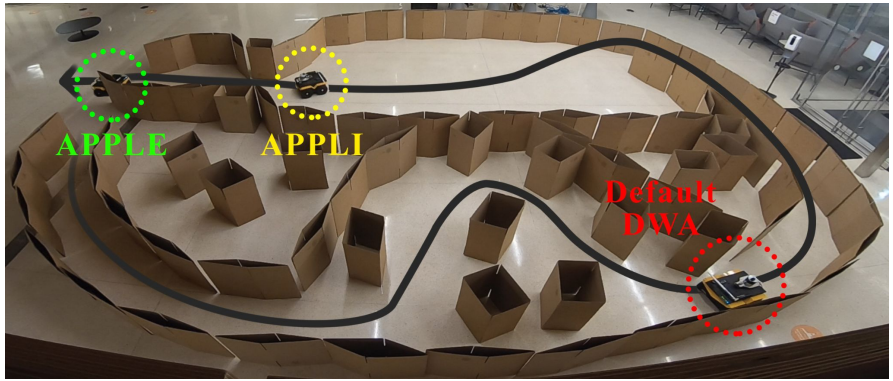
Physical Experiment Results

Traversal Time in Training and Unseen Environment

	Default	APPLI	APPLE (disc.)
Training	143.1±20.0s	79.8±8.1s	75.2±4.1s
Unseen	150.5±24.0s	86.4±1.1s	83.9±4.6s

Implementation details

Only evaluate discrete APPLE as it require less feedback. Discrete APPLE still uses the same parameter library as APPLI. APPLR is not evaluated as it requires infeasible amount of trial and error.

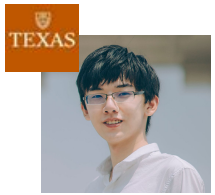


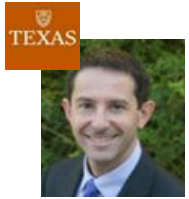
Future Work

- How little feedback is needed to train a good APPLE?
- Human with different expertise levels?
- Human with different feedback criteria?
- How to handle feedback latency?

APPLE: Adaptive Planner Parameter Learning From Evaluative Feedback

Zizhao Wang[1], Xuesu Xiao[1], Garrett Warnell[2], and Peter Stone[1, 3]



[1]The University of Texas at Austin
[2]Army Research Laboratory
[3]Sony AI

Contact Information:
Zizhao Wang: zizhao.wang@utexas.edu
Xuesu Xiao: xiao@cs.utexas.edu

Link to the Paper: https://arxiv.org/pdf/2108.09801.pdf

Scan to access the paper

Related Work

Learning for Navigation

- End-to-end learning (Tai, et al., IROS'17; Zhang, et al., IROS'17)
- Learning for subsystems
  - Global planning (Yao, et al., IROS'19)
  - Local planning (Gao, et al., CoRL'17; Faust, et al., ICRA'18)
- Learning for components in subsystems
  - Cost function (Shiarlis, et al., ICRA'17), cost map (Luber, et al., IROS'12), ...
  - Planner parameters

Reference: Xiao, et al., *Motion Control for Mobile Robot Navigation Using Machine Learning: a Survey*