JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Deep CPT-RL: Imparting Human-Like Risk Sensitivity to Artificial Agents

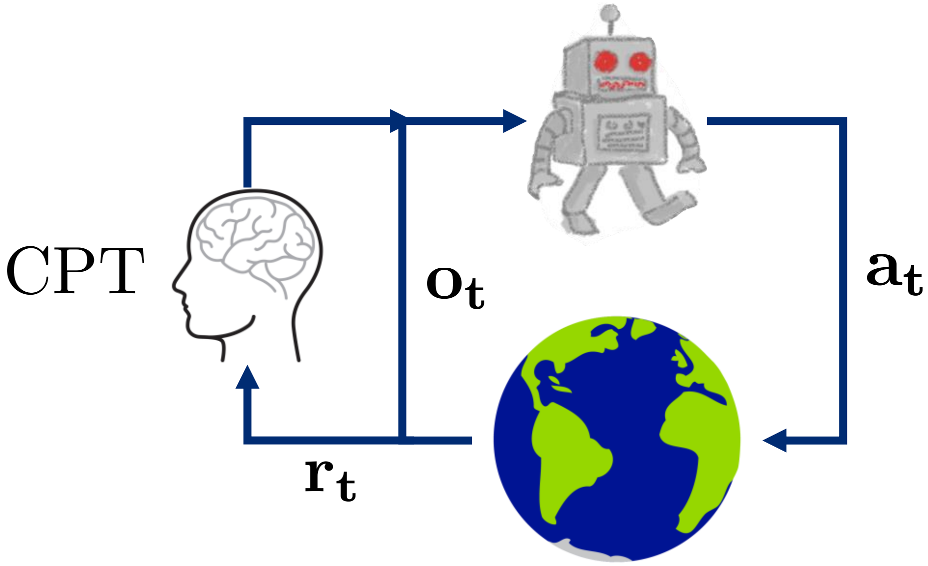
Jared Markowitz, Marie Chau, I-Jeng Wang

SafeAI 2021

February 8, 2021

Motivation and Background

- One contributor to unsafe AI is its clumsy, non-human handling of risk:
  - It does not properly consider rare but potentially catastrophic outcomes
  - It does not asymmetrically value losses and gains
- Cumulative Prospect Theory (CPT) [1, 2] is a leading empirical model of human risk-processing from behavioral economics.
- We seek to incorporate CPT into ***deep*** RL, producing agents that process risk more intelligently.



Methods

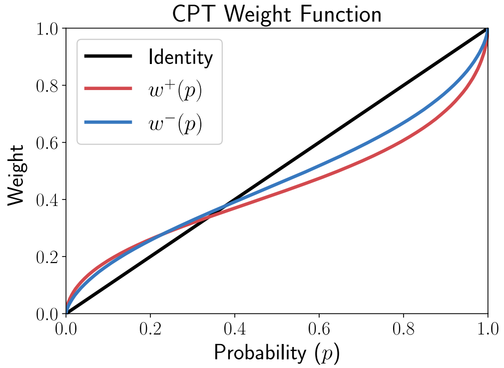
Standard RL

$$\max_{\theta \in \Theta} \int r(\tau) p_\theta(\tau) d\tau$$

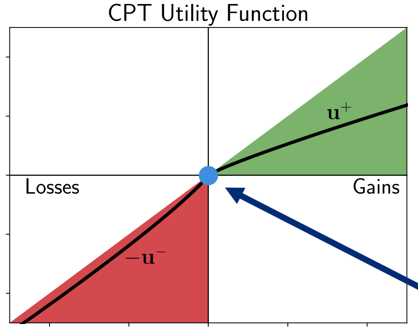
- A single (often convex) reward function makes it difficult to enact risk-sensitive strategies
- Unweighted averaging means rare events have minimal impact (regardless of consequences)

CPT-RL

$$\max_{\theta \in \Theta} \left[\int \left(-u^-(r) \frac{d}{dr}(w^-(P_\theta(r))) + u^+(r) \frac{d}{dr}(-w^+(1 - P_\theta(r))) \right) dr \right]$$



Nonlinear probability scaling emphasizes rare events



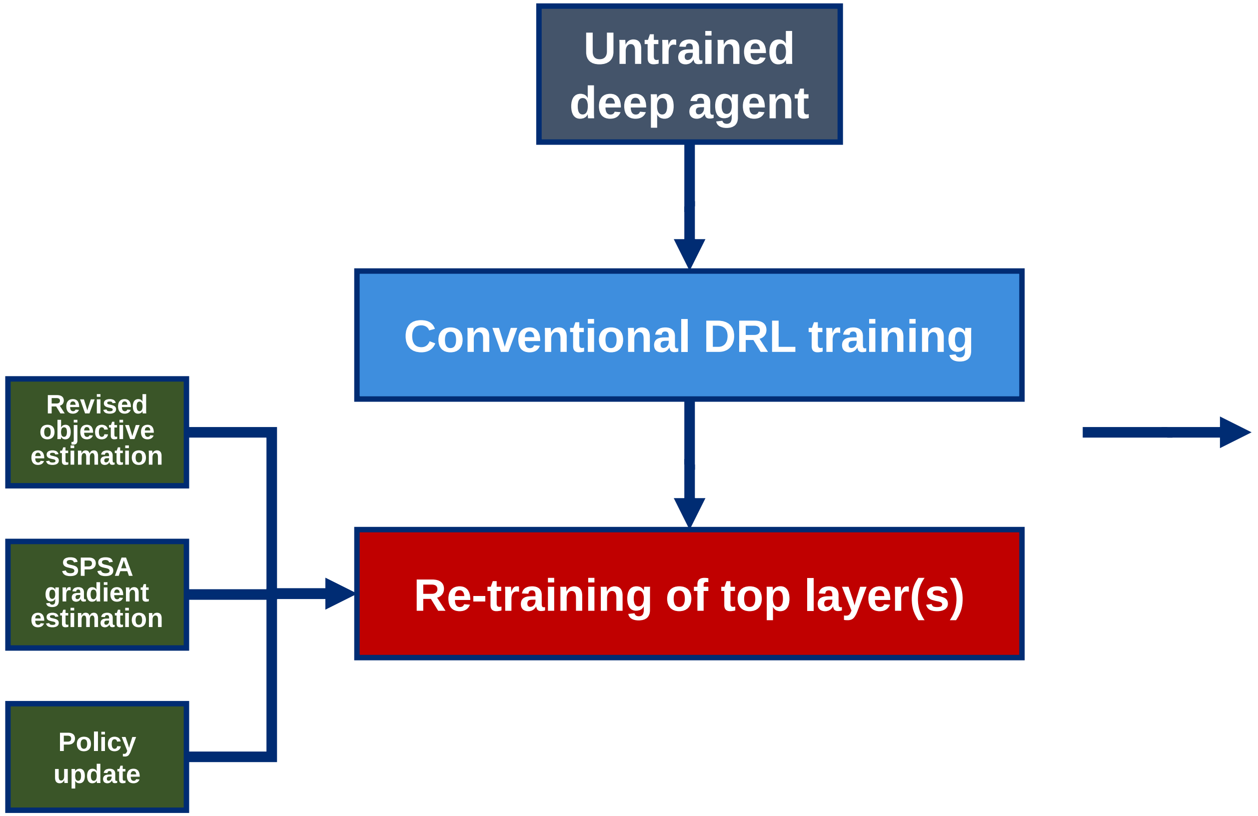
Risk-averse

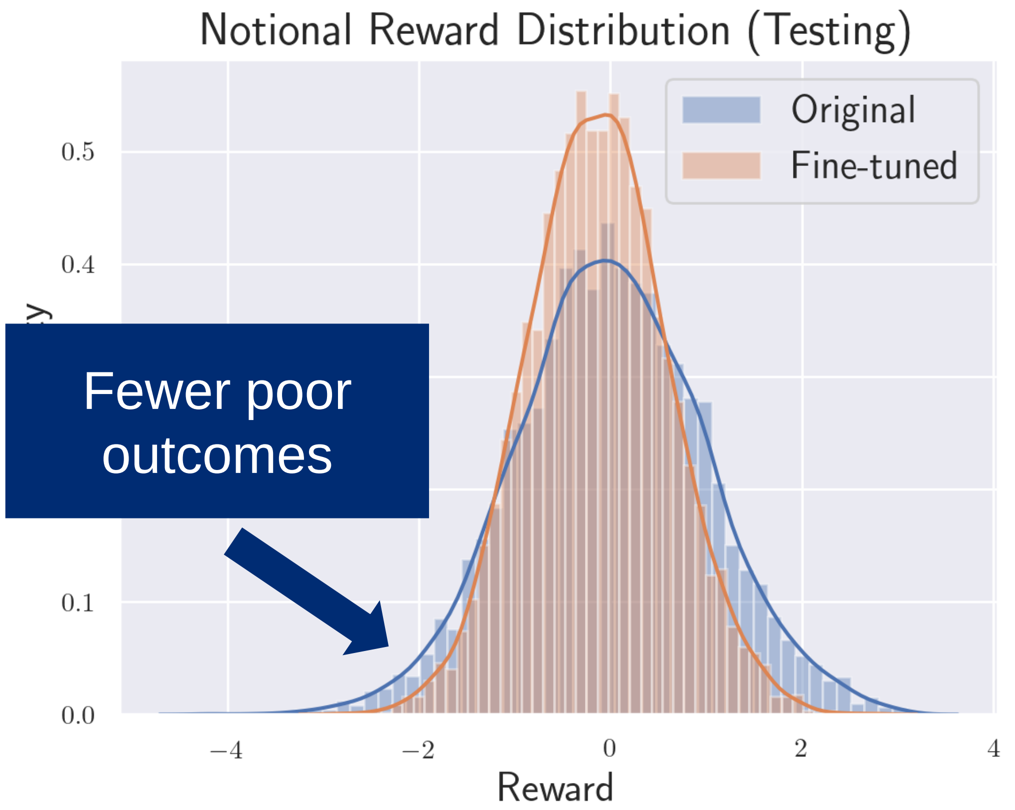
Reference point

Risk-seeking

- We build on work [3] from UMD that allows agents to optimize the CPT value instead of expected reward.
  - The UMD method does not apply to ***deep*** networks.
- We introduce Deep CPT-RL, a method for fine-tuning trained DRL networks [4] to optimize CPT value.
- Our method allows other distributional shaping strategies (e.g. Conditional Value at Risk (CVaR)).

A Two-Stage Approach to Modifying Reward Distributions





We seek to shift the distribution of outcomes in order to mitigate negative outcomes.

CPT Value Estimation [3]

Algorithm 1 CPT-value estimation for Hölder continuous weights

1: Simulate n i.i.d. samples from the distribution of X.

2: Order the samples and label them as follows: $X_{[1]}, X_{[2]}, \ldots, X_{[n]}$. Note that $u^+(X_{[1]}), \ldots, u^+(X_{[n]})$ are also in ascending order.

3: Let

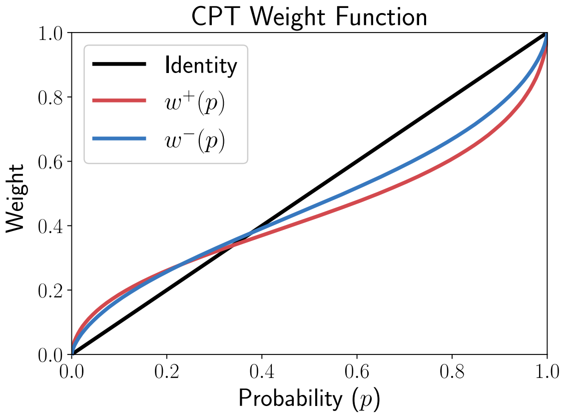
$$\overline{\mathbb{C}}_n^+ := \sum_{i=1}^{n} u^+(X_{[i]}) \left(w^+\left(\frac{n+1-i}{n}\right) - w^+\left(\frac{n-i}{n}\right)\right).$$

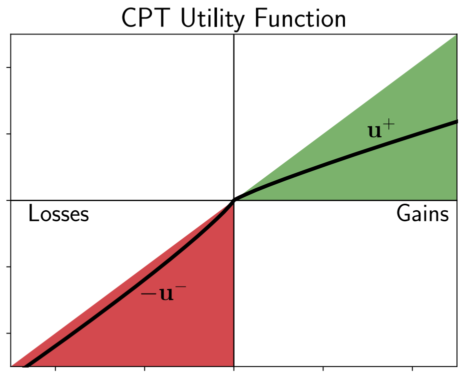
4: Apply u^- on the sequence $\{X_{[1]}, X_{[2]}, \ldots, X_{[n]}\}$; notice that $u^-(X_{[i]})$ is in descending order since u^- is a decreasing function.

5: Let

$$\overline{\mathbb{C}}_n^- := \sum_{i=1}^{n} u^-(X_{[i]}) \left(w^-\left(\frac{i}{n}\right) - w^-\left(\frac{i-1}{n}\right)\right).$$

6: Return $\overline{\mathbb{C}}_n = \overline{\mathbb{C}}_n^+ - \overline{\mathbb{C}}_n^-$.





- This procedure allows a numerical estimation of CPT value via sampling.

Simultaneous Perturbation Stochastic Approximation

- SPSA [5] is an efficient method for numerical gradient estimation.
- Simultaneously perturbs each parameter, rather than doing them one at a time (as in finite differences (FDSA)).
- Gives more noisy but much more efficient gradient estimates.
- Gradient Estimation:

$$\hat{\nabla}_i \mathcal{C}(X^{\theta}) = \frac{\bar{\mathcal{C}}_n^{\theta_n + \delta_n \Delta_n} - \bar{\mathcal{C}}_n^{\theta_n - \delta_n \Delta_n}}{2\delta_n \Delta_n^i}$$

- Parameter Update:

$$\theta_{n+1}^i = \theta_n^i + \gamma_n \hat{\nabla}_i \mathcal{C}(X^{\theta_n})$$

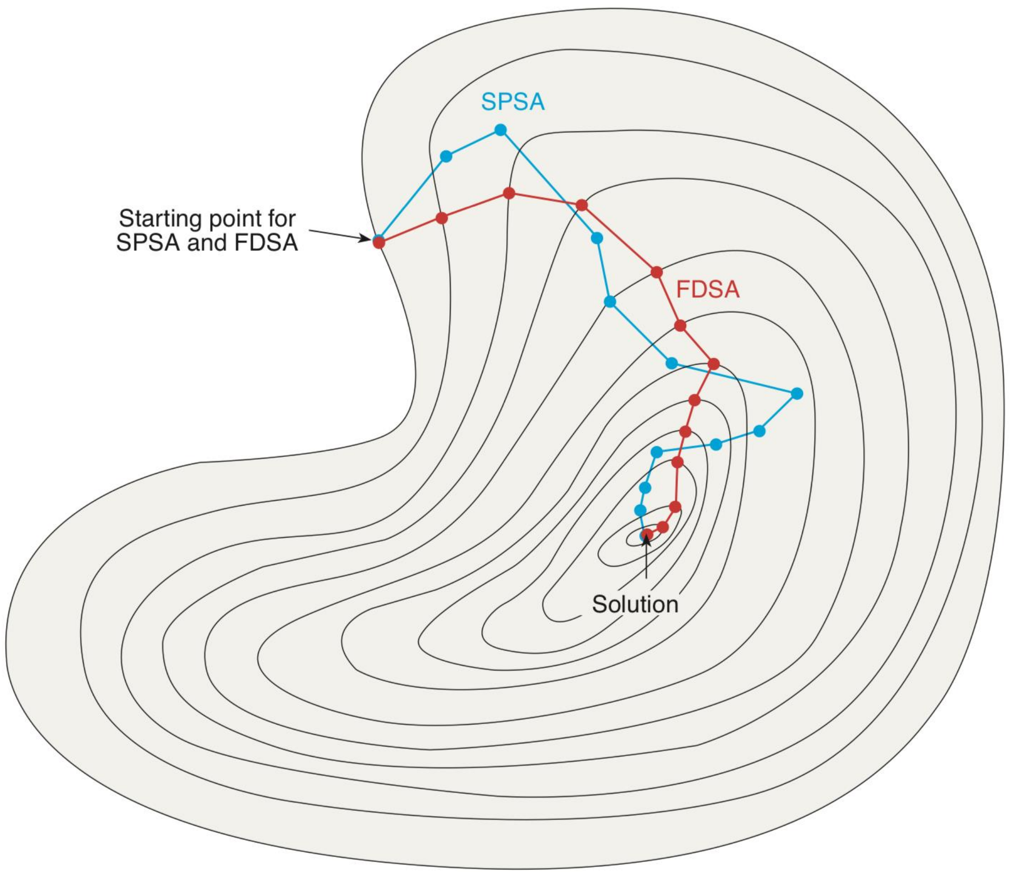
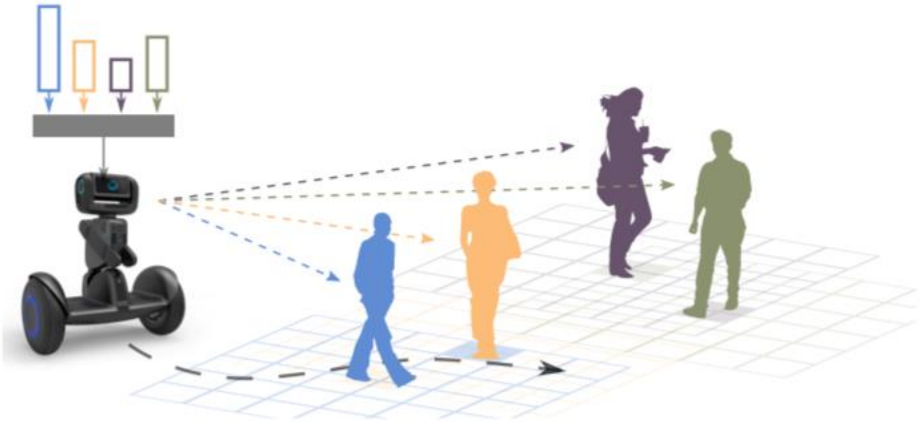


Figure Credit: [5]

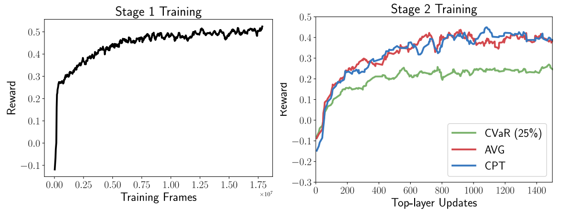
Crowd Navigation Simulation

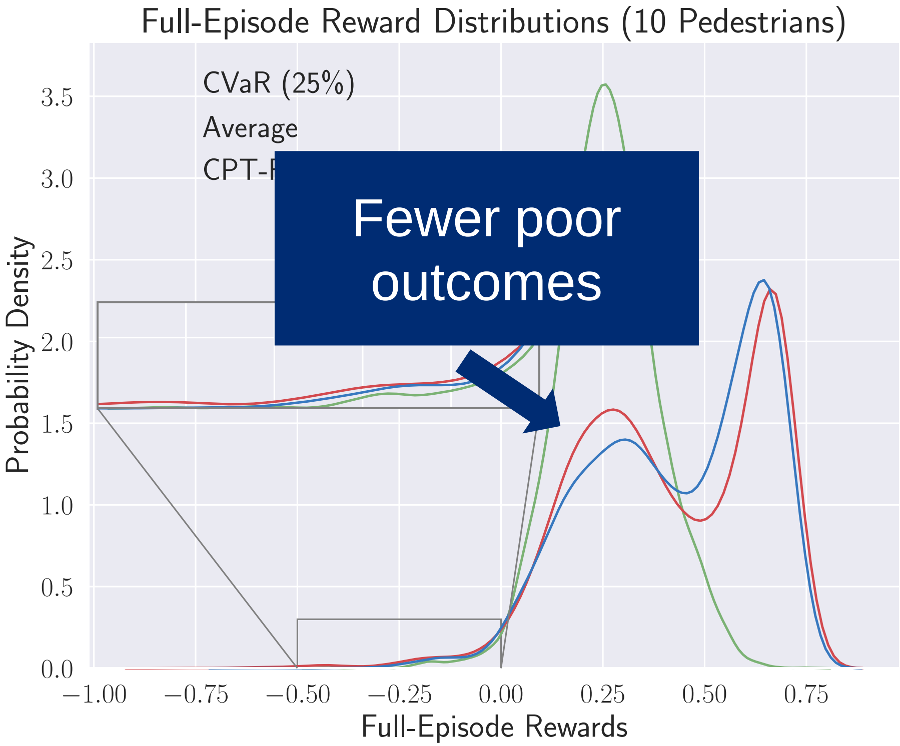
- In the CrowdSim environment [6], a single **robot navigates from a starting location to a goal location, trying to avoid people** who are passing through.
- The people in the simulation proceed from randomized starting points to randomized goal points, trying to avoid collisions with each other.
- In our configuration, the robot is invisible to the people and the episode ends when a collision occurs.
- Here, **risk is measured in the willingness of the agent to risk collisions in the pursuit of speed**.

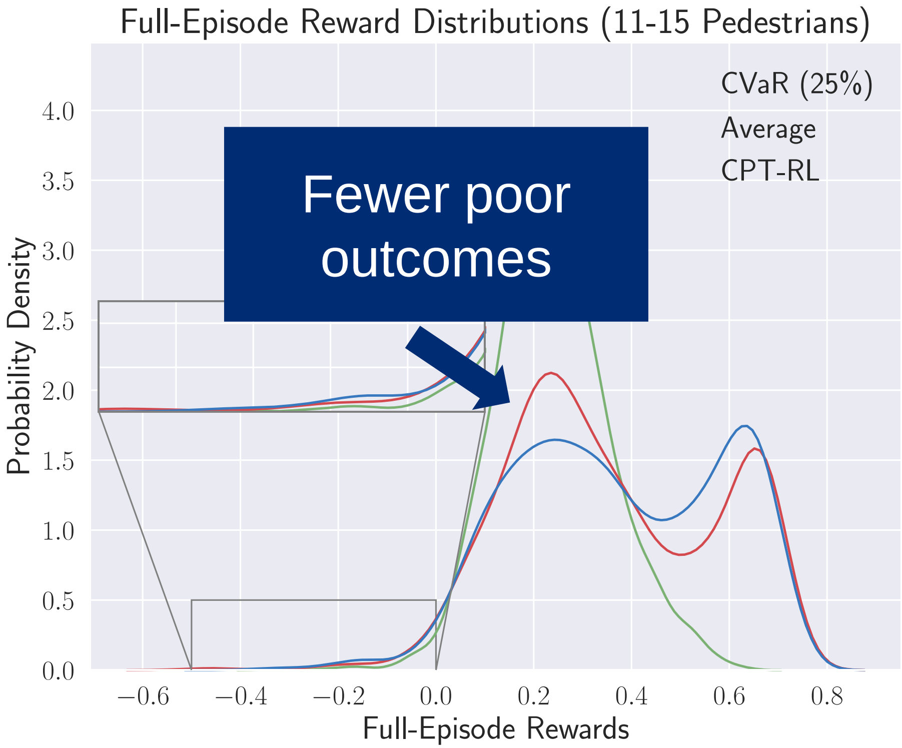


$$r(t) = C_{\text{progress}}\left(d_{\text{goal}}(t-1) - d_{\text{goal}}(t)\right) - C_{\text{time}}$$

Results







	Rewards, 10 Pedestrians			Rewards, 11-15 Pedestrians		
Method	Mean	Median	0.01-quantile	Mean	Median	0.01-quantile
CVaR	0.262 ± 0.002	0.260	-0.029	0.239 ± 0.002	0.232	-0.020
AVG	0.418 ± 0.003	0.414	-0.172	0.358 ± 0.003	0.320	-0.066
CPT	0.432 ± 0.003	0.461	-0.085	0.375 ± 0.003	0.360	-0.123

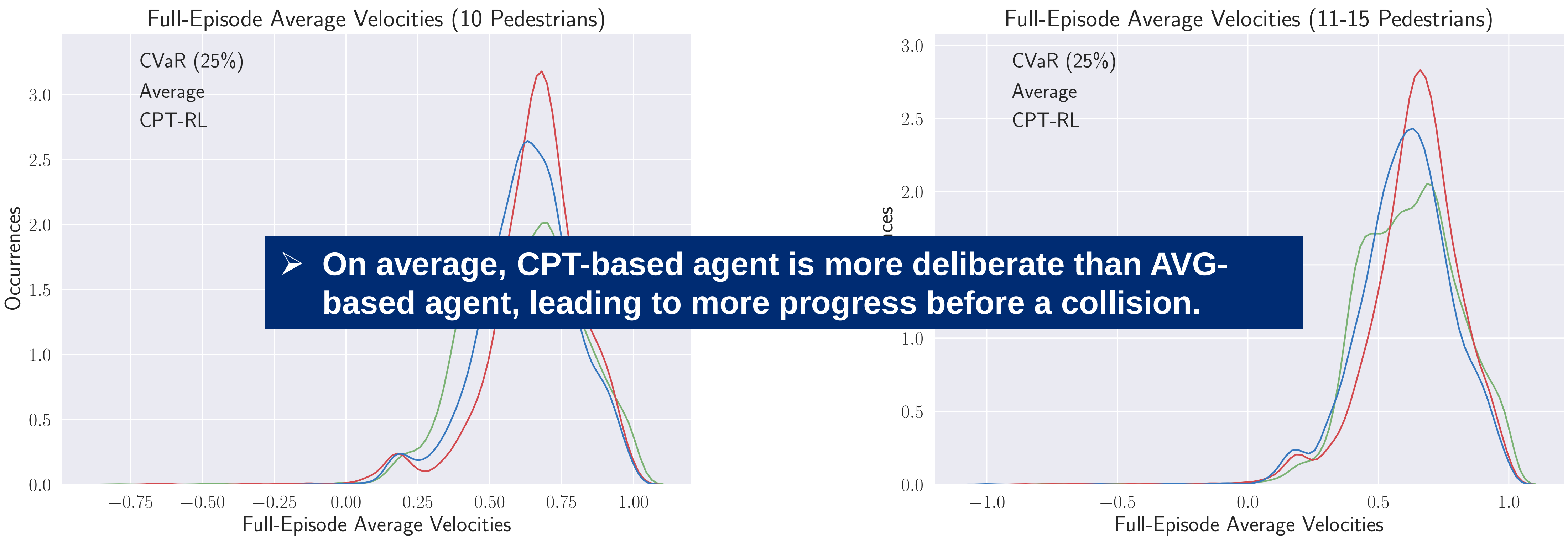
Quantitatively Different Behavior

$p(t) = d_{\text{robot to goal}}(t) - d_{\text{robot to goal}}(0)$

Episode time: T

Episode progress: $p(T)$

Episode velocity: $\frac{p(T)}{T}$



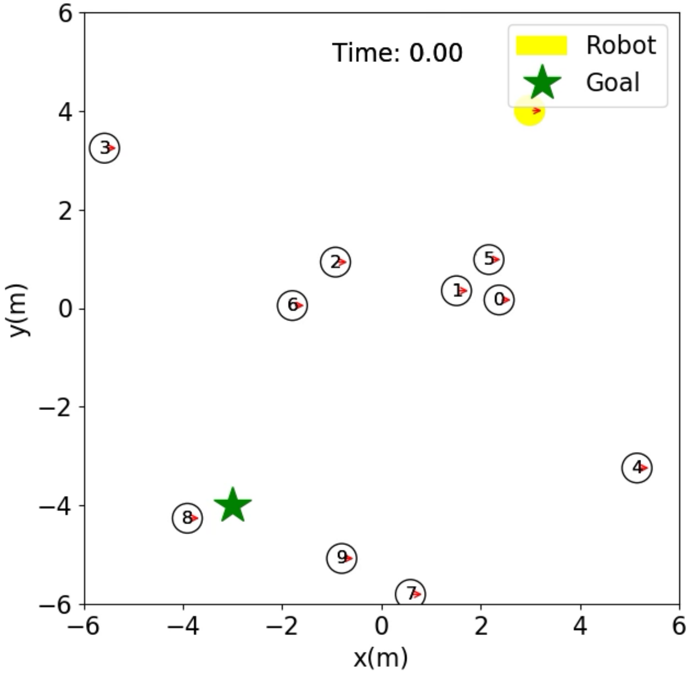
➢ **On average, CPT-based agent is more deliberate than AVG-based agent, leading to more progress before a collision.**

	10 Pedestrians			11-15 Pedestrians		
Method	Progress	Time	Velocity	Progress	Time	Velocity
CVaR	4.35 ± 0.03	8.61 ± 0.10	0.617 ± 0.003	3.81 ± 0.03	7.14 ± 0.08	0.621 ± 0.003
AVG	6.23 ± 0.04	10.26 ± 0.09	0.661 ± 0.002	5.39 ± 0.04	9.07 ± 0.09	0.640 ± 0.002
CPT	6.63 ± 0.04	11.52 ± 0.10	0.631 ± 0.002	5.86 ± 0.04	10.55 ± 0.10	0.605 ± 0.002

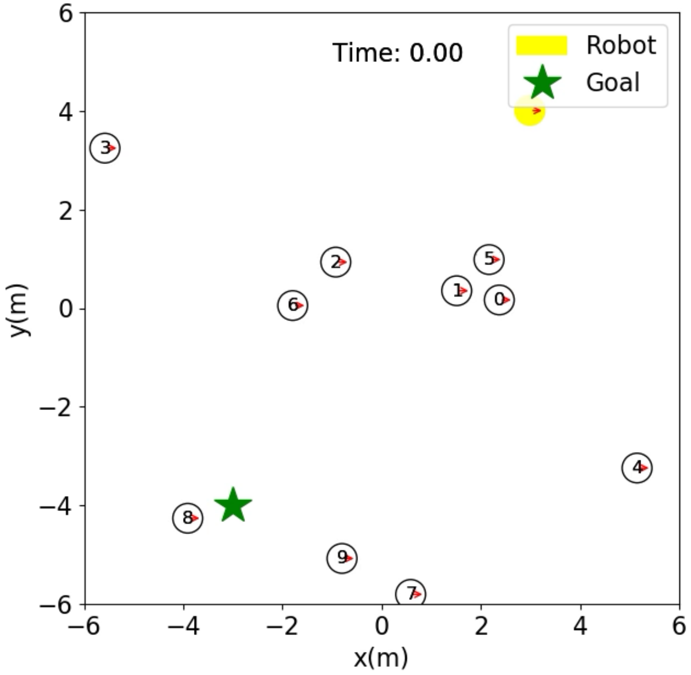
Illustrative Example 1

Avoiding an early crash

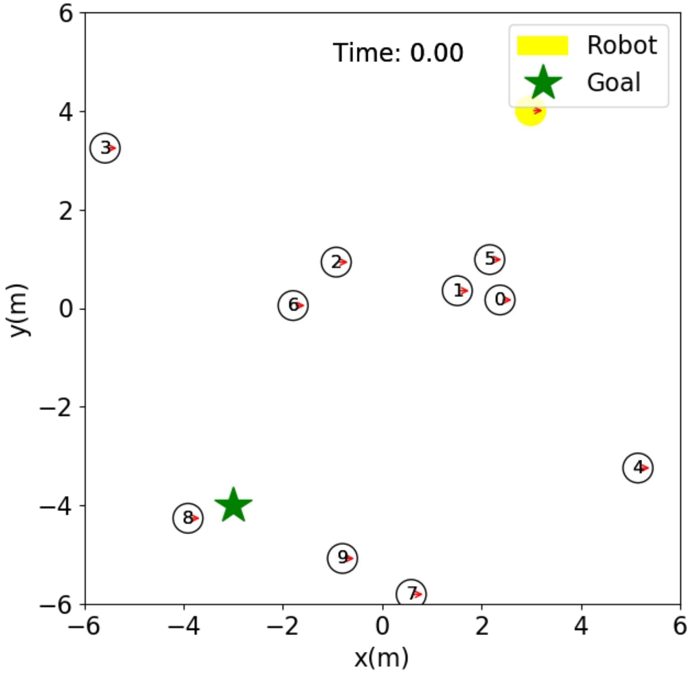
$$r(t) = C_{\text{progress}}\left(d_{\text{goal}}(t-1) - d_{\text{goal}}(t)\right) - C_{\text{time}}$$



AVG



CVaR(25%)

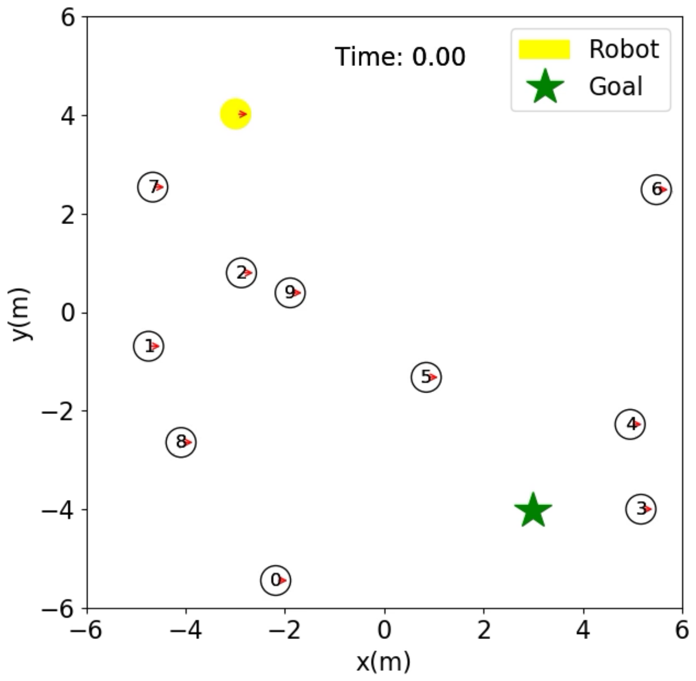


CPT

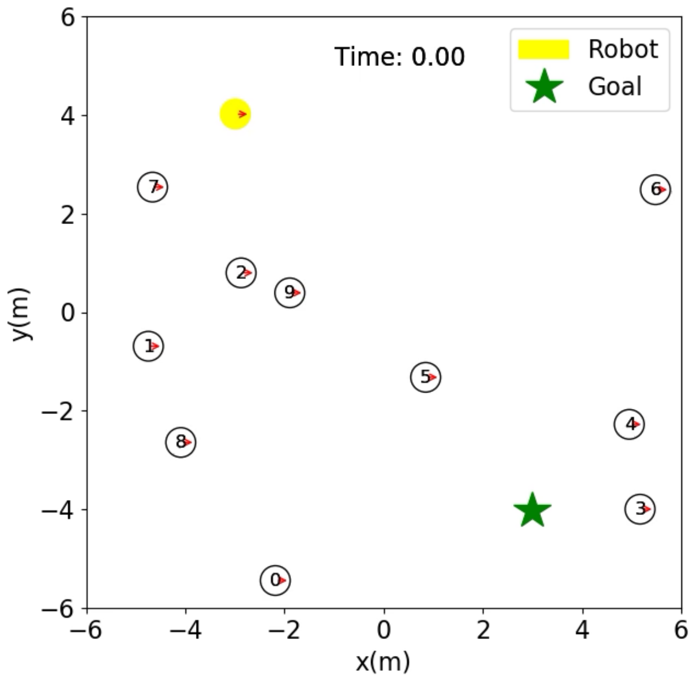
Illustrative Example 2

Reaching the goal

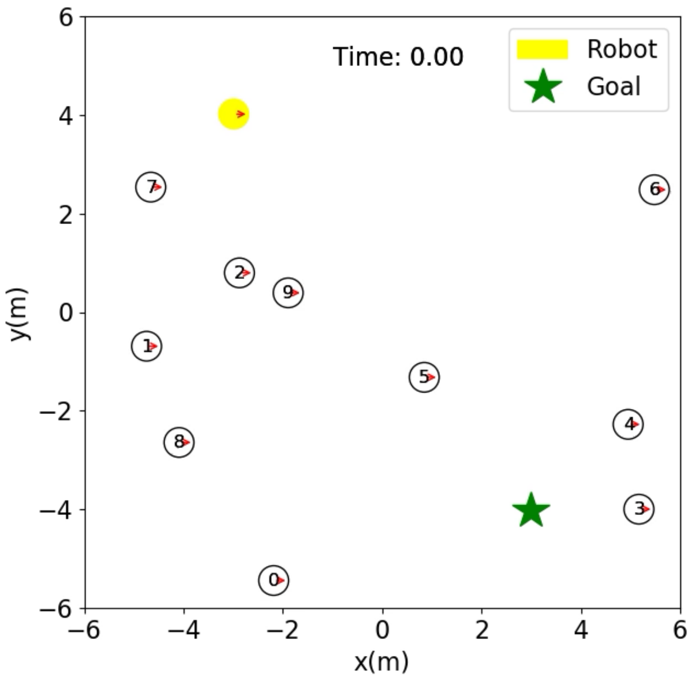
$$r(t) = C_{\text{progress}}\left(d_{\text{goal}}(t-1) - d_{\text{goal}}(t)\right) - C_{\text{time}}$$



AVG



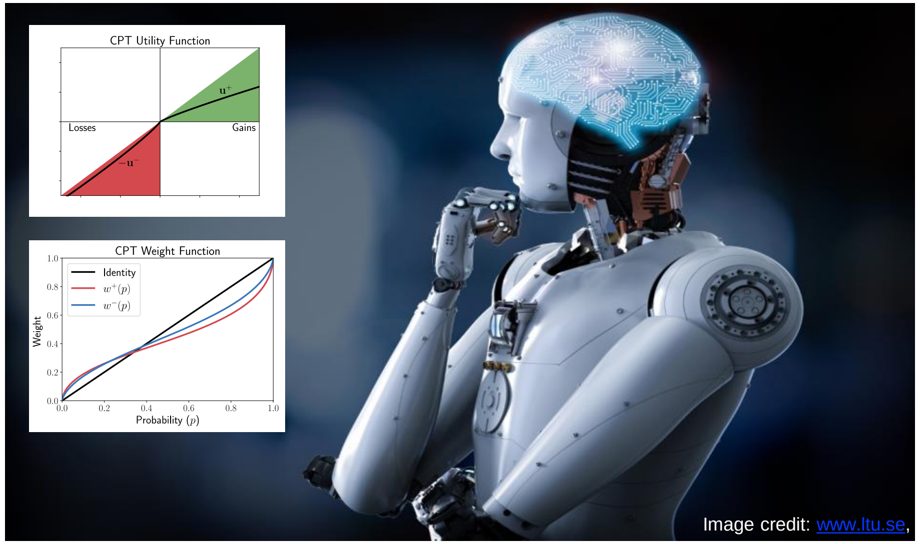
CVaR(25%)



CPT

Summary and Future Work

- We have developed a method for modifying the distribution of outcomes for DRL agents.
- Our approach allows for optimization of quantities beyond expected reward.
- Agents trained to maximize CPT value demonstrate quantitatively different behavior than those trained to maximize average total reward.
- Areas of current and future research include
  - Methods for making this learning more robust
  - Exploration of behaviors induced by different distributional objectives
  - Application to more complex and realistic environments



Work supported by APL IRAD and the Johns Hopkins Institute for Assured Autonomy (IAA)

References

[1] Daniel Kahneman and Amos Tversky. “Prospect Theory: An Analysis of Decision under Risk” *Econometrica* Vol. 47 No. 2 pp. 263-291 (1979).

[2] Amos Tversky and Daniel Kahneman. “Advances in prospect theory: Cumulative representation of uncertainty” *Journal of Risk and Uncertainty* Volume 5, Issue 4, pp. 297-323 (1992).

[3] Prashanth L.A. et al. “Cumulative Prospect Theory Meets Reinforcement Learning: Prediction and Control” *ICML* (2016).

[4] J. Schulman et al. “Proximal Policy Optimization Algorithms” *arXiv:1707.06347.*

[5] James Spall, “An Overview of the Simultaneous Perturbation Method for Efficient Optimization” *Johns Hopkins APL Technical Digest* Volume 19, Number 4 (1998).

[6] Changan Chen et al. “Crowd-Robot Interaction: Crowd-aware Robot Navigation with Attention-based Deep Reinforcement Learning” *ICRA* (2019).

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY