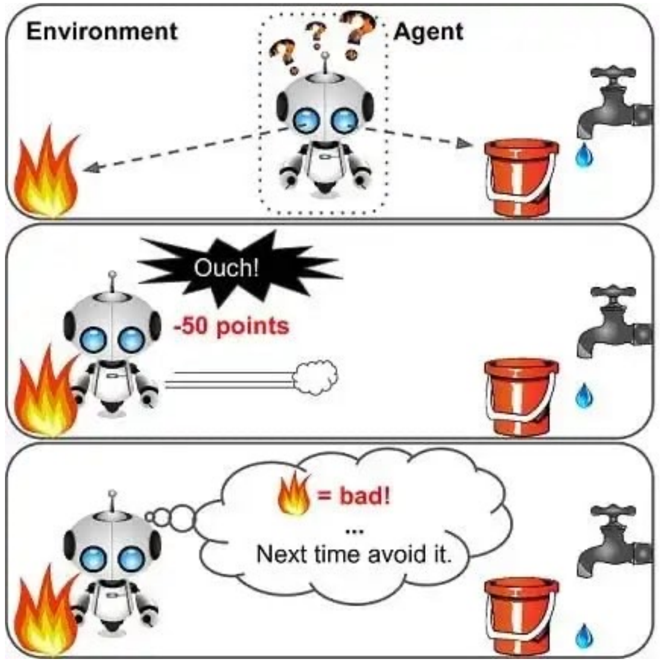
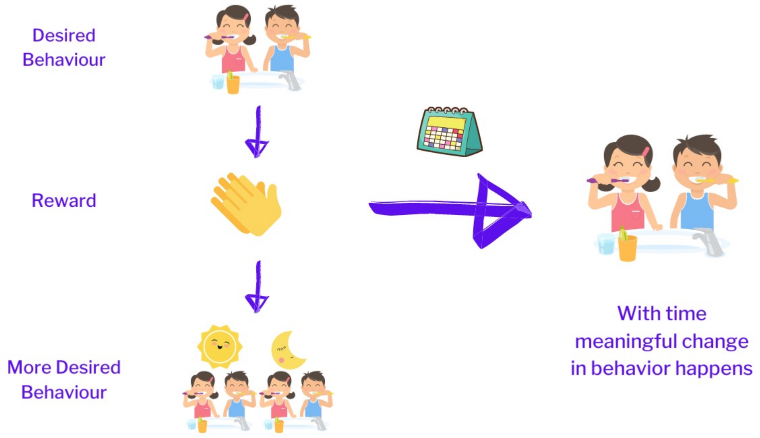
Evolving AI Decision-Making: From Safe Reinforcement Learning to Intelligent Systems with Language Models

Ali Baheri

March 25, 2024

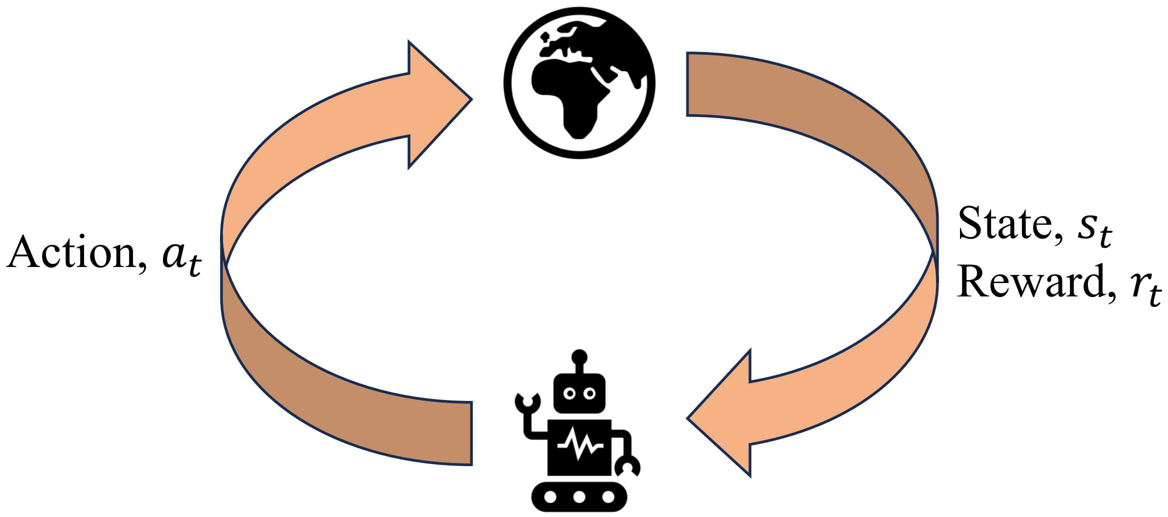
Reinforcement Learning Intro





Reinforcement Learning Intro

- RL is a type of machine learning where an agent learns to make decisions by taking actions in an environment to maximize some notion of cumulative reward.



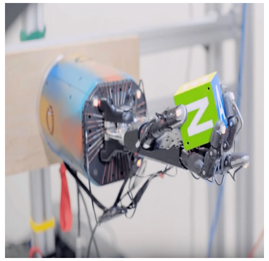
Safety in Reinforcement Learning

- Safety in RL is defined by the system's ability to attain the environmental objectives while adhering to safety constraints.

RL in simulated world

Games

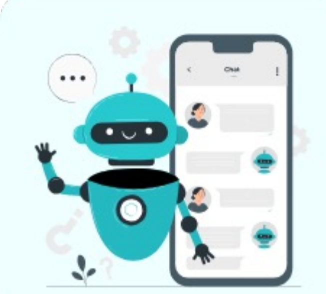
Protein folding



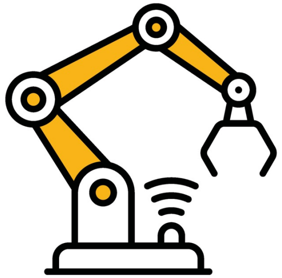
Robotics

RL in physical world

Autonomous driving



Chatbot



Robotics

Safety Constraints

- Safety constraints are rules or limitations specific to an environment, designed to prevent harmful outcomes by an RL agent, ensure ethical compliance, and mitigate risks while maximizing environmental objectives.
- Overall goal of constrained RL: **maximize expected return** subject to the environment specific **safety constraints**

Safety Constraints in Autonomous Driving

Maximize expected return

$$\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\right]$$

subject to

Safety constraints

Maximize average velocity while driving to destination

subject to

- Adhere to speed limits
- Obey traffic signs
- Maintain safe following distance

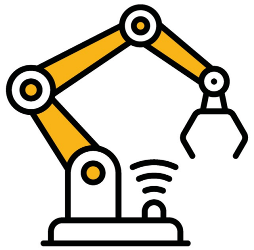
Safety Constraints in Robotics

Maximize expected return

$$\mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k}R_{t+k+1}\right]$$

subject to

Safety constraints



Assist humans in a collaborative environment

subject to

- Maintain a safe distance from humans
- Adhere to power/velocity limits
- Operate within designated envelope

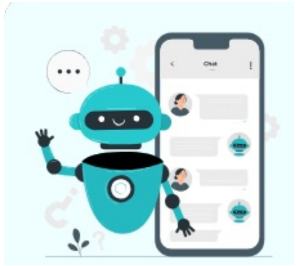
Safety Constraints in Chatbots

Maximize expected return

$$\mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \right]$$

subject to

Safety constraints



Generate responses to user prompts

subject to

- Avoid discriminatory/biased/offensive responses
- Filter inappropriate text
- Limit misinformation

Defining Safety Constraints

- These safety constraints are often defined in prior works using:
  - Expert knowledge
  - Computational methods from data

- Predefined safety constraint may not always be adequate in dynamic and complex environments.
  - Outdated expert knowledge/information
  - The need for extensive historical data
  - Their static nature

Challenges of Static Safety Constraints

- Static, predefined safety constraints lack flexibility in dynamic environments where conditions and parameters are subject to frequent changes
- Consider the frozen-lake environment

Initial state *Environment evolving through time* *Further changes occurring…*

Challenges of Static Safety Constraints

- Uber Autonomous Vehicle Incident, 2018

A frame from the Dash cam footage released by Uber Inc.

Reports claim that the death of Elaine Herzberg in March 2018 was caused by a self-driving vehicle system that could not detect "*jaywalkers*" and failed to classify Herzberg as a pedestrian. *the system design did not include consideration for jaywalking pedestrians*.

Lack of Predefined Safety Constraints

- In some instances, predefined safety constraints may not be unavailable and impossible to acquire
  - In environments that are uncharted and never before explored
  - In environments that are too dangerous to explore repeatedly to have a good idea of the safety constraints
  - In environments where the collection of extensive historical data poses potential risks.

Problem Statement

- We consider the problem of safe RL policy synthesis in an environment where safety constraints are unknown *a priori*
- Our ultimate objective is to concurrently:

    1. Optimize *parameters of a safety specification* to closely mirror the true environmental safety constraints

    2. Solve a constrained optimization problem to obtain an *optimal policy* such that the policy adheres to the learned STL safety constraint while maximizing returns

This Talk

- Our contributions:
    1. A framework for concurrently learning safety constraints and RL control policy
    2. An adaptation of the TD3-Lagrangian RL algorithm to compute costs from an STL specification
    3. Proving the efficacy of our framework through evaluations in various safety critical environments
- Outline

Background → Our Approach → Case studies → Results & Discussion → Summary & Future Work

Signal Temporal Logic (STL)

- STL is a formal language used for specifying properties of signals over time.
- STL grammar is given by:

$$\phi \coloneqq \top \mid \mu(x) < c \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \, U_{[t_1,t_2]} \phi_2$$

True Predicate Not And Until

- From which additional logical and temporal operators were derived:

$\phi_1 \wedge \phi_2$, Or

$\boldsymbol{F}_{[t_1,t_2]} \, \phi$, Eventually

$\boldsymbol{G}_{[t_1,t_2]} \, \phi$, Always

$\phi_1 \Rightarrow \phi_2$, Implies

Example: $\boldsymbol{\phi} = G_{[0,3]} \, (x < 5) \wedge (y > 3)$

Qualitative Semantics

- Qualitative semantics (Boolean semantics) of STL indicate weather or not a signal satisfies an STL formula (True/False)
- Quantitative semantics indicate how well a signal satisfies an STL formula through a robustness degree

STL Quantitative semantics

Formula	**Robustness value**
$\rho(s_t, >)$	$\rho_{\max}$
$\rho(s_t, \mu_c)$	$\mu(x_t) - c$
$\rho(s_t, \neg\phi_1)$	$-\rho(s_t, \phi_1)$
$\rho(s_t, \phi_1 \wedge \phi_2)$	$\min(\rho(s_t, \phi_1), \rho(s_t, \phi_2))$
$\rho(s_t, \phi_1 \vee \phi_2)$	$\max(\rho(s_t, \phi_1), \rho(s_t, \phi_2))$
$\rho(s_t, \phi_1 \Rightarrow \phi_2)$	$\max(-\rho(s_t, \phi_1), \rho(s_t, \phi_2))$
$\rho(s_t, \mathrm{F}_{[a,b]}\phi_1)$	$\max_{t' \in [t+a, t+b]} \rho(s_{t'}, \phi_1)$
$\rho(s_t, \mathrm{G}_{[a,b]}\phi_1)$	$\min_{t' \in [t+a, t+b]} \rho(s_{t'}, \phi_1)$
$\rho(s_t, \phi_1 \mathcal{U}_{[a,b]} \phi_2)$	$\max_{t' \in [t+a, t+b]} \Big(\min\{\rho(s_{t'}, \phi_2), \min_{t'' \in [t, t']} \rho(s_{t''}, \phi_1)\} \Big)$

Parametric STL (pSTL)

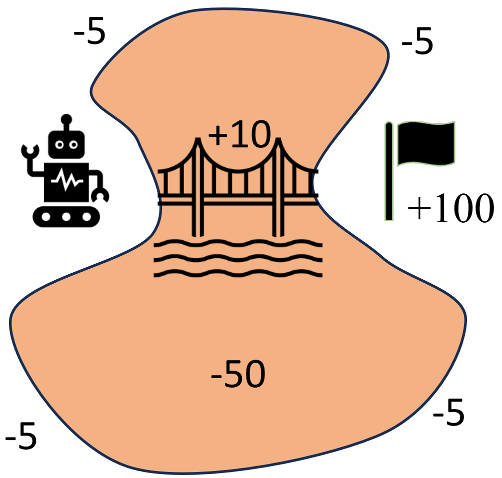
- pSTL is an extension of STL where only the structure/template of the STL formula is given, i.e., the STL formula is parameterized
  - The time-bounds $[t1, t2]$ for temporal operators
  - The constants μ for inequality predicates are replaced by free parameters

Example: $\boldsymbol{\phi} = G_{[t_1,t_2]}\,(x < \mu_1) \land (y > \mu_2)$

RL vs. Constrained RL

- The RL objective is to maximize cumulative discounted rewards within an episode

$$\max \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$



- The constrained RL objective is to maximize reward while also satisfying environmental safety constraints

$$\max J^R(\pi_\theta)$$
$$s.t. \quad J^C(\pi_\theta) \le d$$

J^R is the reward objective function, J^C is the constraint function, and d is the cost limit.

Bayesian Optimization

- BO is an optimization strategy for black-box functions that are intractable to analyze
  - Non-convex, non-linear, and/or computationally expensive to evaluate
- A technique to find the global optimum of an objective function by building a probabilistic model of the objective function, known as the surrogate function.
- Expected Improvement (EI) acquisition function:

$$EI(p) = \mathbb{E}\left[\max\left(0, f_{min}(p) - f(p)\right) \mid p, D\right]$$

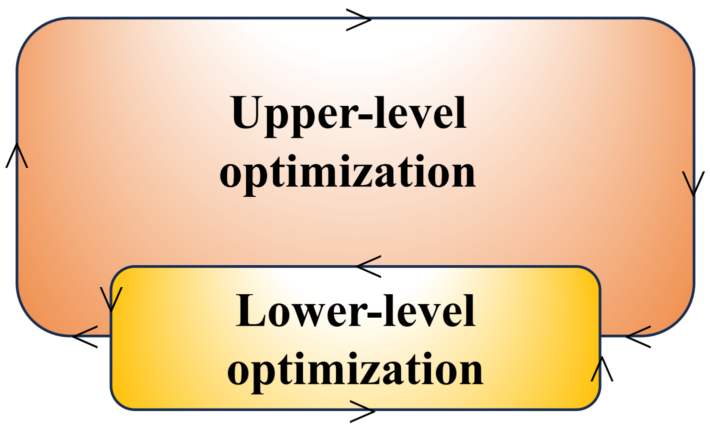
p is the parameter set, D represents the current observations, and f_{min} is the minimum value observed so far

Our Proposed Approach

- We propose a framework for concurrently learning safe RL policies and STL safety constraint parameters in an environment where safety constraints are not defined *a priori*
- Begins with:
  1. A small set of labeled data, D_s and D_{us}
  2. A pSTL specification, ϕ_p
- We frame this concurrent learning problem as a bi-level optimization,
  - upper-level ⟹ pSTL parameter synthesis
  - lower-level ⟹ constrained RL policy optimization
  - assistance of a human expert

Bi-level Optimization

- An optimization approach that contains two levels of optimization tasks where one optimization task, the lower level, is nested within the other, the upper level.



$$\arg\min_{p} f\left(\phi_{v(p)}, \pi^{*}\left(\phi_{v(p)}\right)\right),$$

$$s.t. \quad \pi^{*}\left(\phi_{v(p)}\right) \in \arg\max_{\pi_\theta \in \pi_c} J^{R}\left(\pi_\theta(\phi_{v(p)})\right)$$

f is the upper-level objective function with optimization variable p and π is the lower-level optimization objective with optimization variable θ.

STL Parameter Learning

- Upper-level optimization
- A Bayesian optimization process designed to obtain the optimal parameters p^* of a given pSTL formula ϕ_p using the labeled safe and unsafe datasets D_s and D_{us}
- The final STL $\phi_{v(p^*)}$ best classifies between D_s and D_{us} such that:
  - Traces labeled "safe" by the human expert, x_s $\longrightarrow$ $\rho(\phi_{v(p^*)}, x_s) > 0$
  - Traces labeled "unsafe" by the human expert, x_{us} $\longrightarrow$ $\rho(\phi_{v(p^*)}, x_{us}) < 0$

STL Parameter Learning

- Objective function:

$$f\left(\phi_{v(p)}\right) = \frac{1}{2}\left(\frac{N_{\rho\left(\phi_{v(p)}\right)^{-}|x_s}}{N_{x_s}} + \frac{N_{\rho\left(\phi_{v(p)}\right)^{+}|x_{us}}}{N_{x_{us}}}\right)$$

False Negative Rate

False Positive Rate

x_s and x_{us} are safe and unsafe trajectories, respectively, sampled from their respective datasets

- “Balanced” misclassification rate (MCR)
- Goal: minimize f,
- Output: $\phi_{v(p*)} \cong "\phi_{cost}"$

Policy Learning: twin delayed deep deterministic policy gradient (TD3)

- A class of actor-critic RL algorithms that is designed to address the overestimation bias in the deep deterministic policy gradient (DDPG) algorithm
- How?
    - Clipped double-Q learning
    - Delayed policy update
    - Target policy smoothing

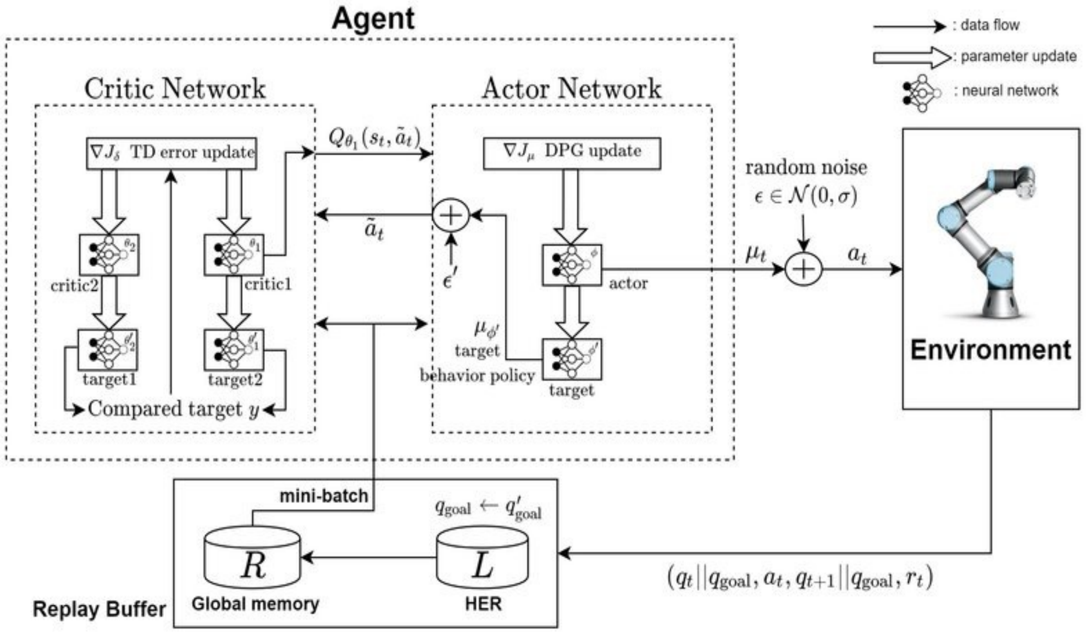


Image credit: Google search

Policy Learning- TD3 Structure

- Lagrange multiplier method

  - Transforms a constrained optimization problem into an equivalent unconstrained optimization problem through Lagrangian relaxation procedure that introduces Lagrange coefficient λ

$$\max_{\pi_\theta \in \pi_C} J^R(\pi_\theta) \quad \boldsymbol{s.t.} \quad J^C(\pi_\theta) \le d$$

$$\Downarrow$$

$$\max_{\theta} \min_{\lambda \ge 0} \mathcal{L}(\theta, \lambda) = J^R(\pi_\theta) - \lambda\,(J^R(\pi_\theta) - d)$$

Goal: Find optimal values θ^* and λ^*

Policy Learning- TD3 Structure

- TD3-Lagrangian:

$$L = -Q^V(\pi_\theta, s) + \lambda \cdot Q^C(\pi_\theta, s)$$

  Q^V is the minimum value of the two reward critic network outputs, Q^C is the value of cost critic network, and π is the policy network.

- Lagrange coefficient update rule

$$\lambda' = \lambda + \eta(J^C(\pi_\theta) - d)$$

  η is the when J^C exceeds the constraint threshold d, λ is increased to prioritize cost minimization

Logically-Constrained TD3

- Cost assignment

  - We propose a novel modification to the TD3-Lagrangian architecture redefining the cost function *logically*, using the learned STL specification ϕ_{cost}

  - Cost at each step:

$$c\ (s_t, a_t) = \begin{cases} 1, & \text{if } \rho(\phi_{cost}) < 0 \\ 0, & \text{if } \rho(\phi_{cost}) \geq 0 \end{cases}$$

$-\ \rho(\phi_{cost}) < 0 \Longrightarrow s_t$ does not satisfy ϕ_{cost}

$-\ \rho(\phi_{cost}) \geq 0 \Longrightarrow s_t$ satisfies ϕ_{cost}

Human Feedback Mechanism

- A human expert iteratively provides labels to the rollout traces generated through the execution of π^*
- Why?
  - Because acquiring an extensive, diverse labeled dataset is often impractical
- Our strategy focuses on attaining <u>sufficiently accurate pSTL parameters</u> with the <u>minimal necessary amount of data</u>

  - Iteratively expanding the “small” initial dataset of labeled data at each loop
  - Refining the parameter assignment for the pSTL using the updated dataset

Human Feedback Mechanism

- Automation of human labeling for the purpose of experimentation:

  - Computing the robustness value of each trace within the rollout set with respect to the **True STL safety constraint $\boldsymbol{\psi}$**

  - The use of ψ is only for automation purposes, and in real-world applications the actual safety constraint remains unknown to the algorithm

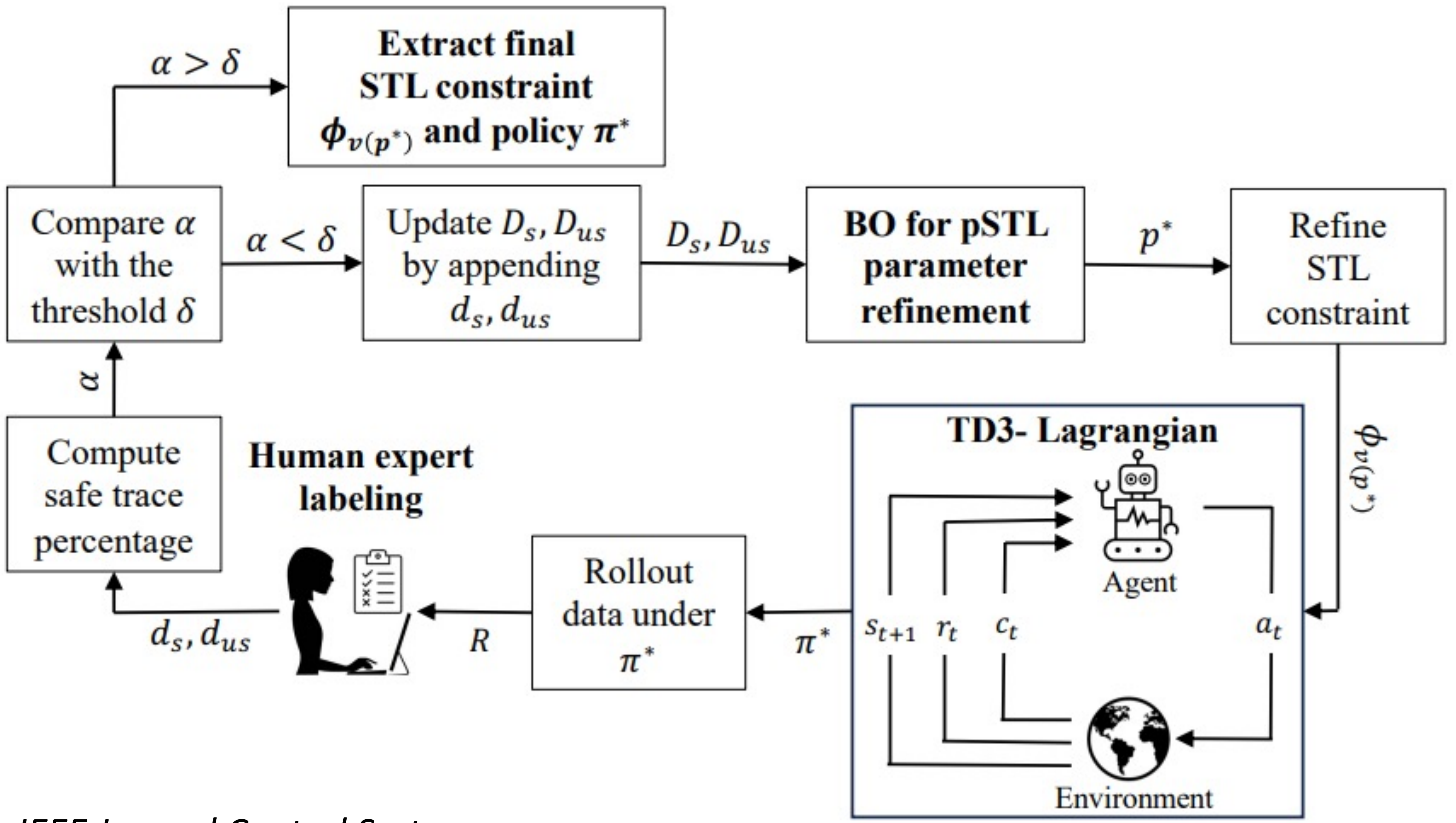
"Satisfies / models" $\equiv \rho(\psi, x) \geq 0$

$$L(x) = \begin{cases} 1, if\ x \vDash \psi \\ 0, if\ x \nvDash \psi \end{cases}$$

"Does not satisfy" $\equiv \rho(\psi, x) < 0$

  - Traces labeled safe are append to D_s,Traces labeled unsafe are append to D_{us}

Our Proposed Framework



Under review for the IEEE Journal Control Systems

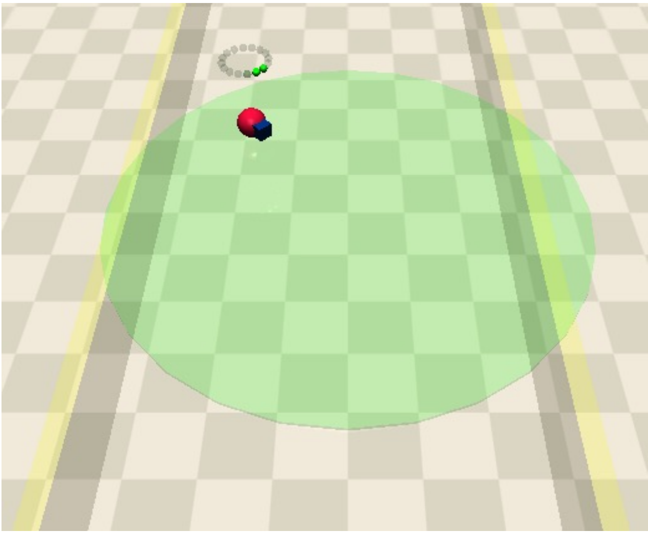
Case Study 1: Safe Navigation- Circle

- Goal: agent needs to move in a circular motion within the circle area (green), while also attempting to stay at the outermost circumference of the circle

$$r_t = \frac{1}{r_a - r_c} \cdot \frac{-uy + vx}{r_a}$$

- Constraint: avoid going outside safety boundaries that intersect with the circle (yellow)

$$\phi_p = G\left(\neg\left((x_a < x_{\tau^-}) \bigvee (x_a < x_{\tau^+})\right)\right)$$

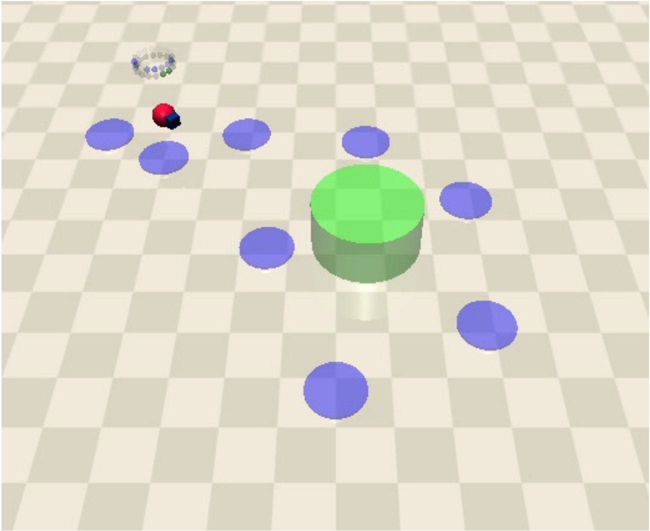


- Unknown Constraint : The x coordinates of the boundaries
- 2 safety parameters to learn

Case Study 2 : Safe Navigation- Goal

- Goal: agent needs to navigate towards a designated goal location (green) starting from a random initial state. New goal randomly assigned upon reaching the goal

$$r_t = (d_{t-1} - d_t) \cdot \beta$$



- Constraint: avoid collision with the hazard areas (blue)

$$\phi_p = G\left(\neg\left(\bigvee_{i=1}^{8} \sqrt{\left(x_a - x_{h,i},\right)^2 + \left(y_a - y_{h,i},\right)^2} < r_h\right)\right)$$

- Unknown Constraint : The x-y coordinates of the hazards-
- 16 safety parameters to learn

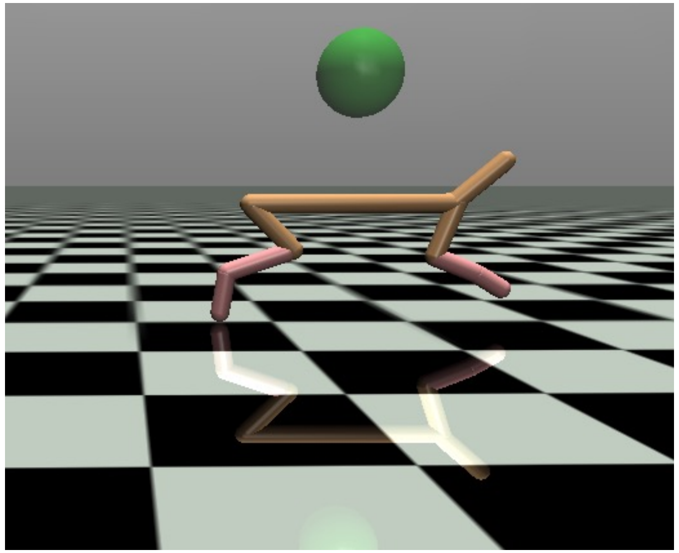
Case Study 3: Half Cheetah

- Goal: agent needs to apply torque on the joints to make the cheetah run in the forward direction to achieve maximum speed

$$r = (w_f \cdot \frac{x_{t-1} - x_t}{d_t}) - (w_c \cdot \sum(a_t^2))$$

- Constraint: stay below the maximum allowable x-velocity, u_{max}

$$\phi_p = G(\neg(u_a > u_{max}))$$



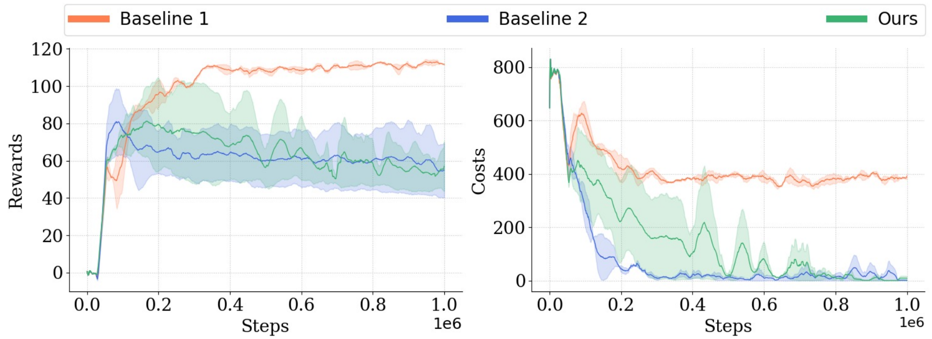
- Unknown Constraint : the x-velocity threshold
- 1 safety parameter to learn

Evaluation

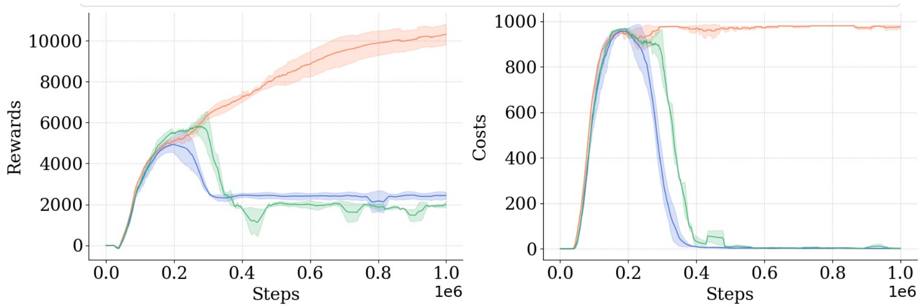
- We evaluate key performance metrics of two primary tasks:
  1. Safe policy optimization
  2. pSTL parameter synthesis
- Compare results with two baselines:
  1. **Baseline 1:** unconstrained RL policy optimization in an environment in which safety constraints are unknown
  2. **Baseline 2:** constrained RL policy optimization in an environment with known STL safety constraint

Results and Discussion

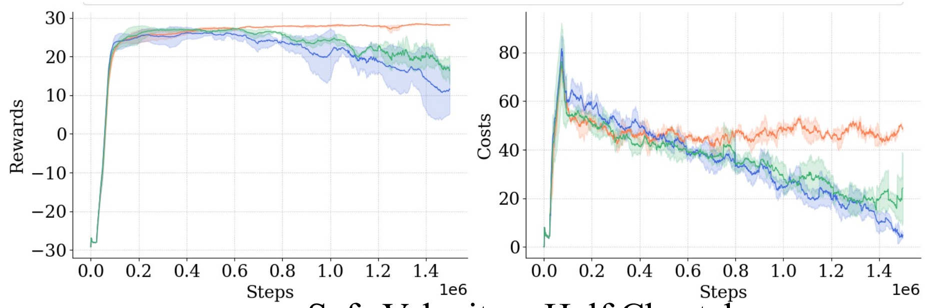
- A trade-off between rewards and costs (not trivially safe)
- Baseline 1 achieves the highest reward, yet it concurrently incurs the highest cost
- Our algorithm exhibits a reduction in rewards compared to baseline 1; however, it succeeds in reducing costs substantially across all case studies
- The performance of our algorithm closely mirrors that of baseline 2



a. Safe Navigation - Circle



b. Safe Navigation - Goal



c. Safe Velocity – Half Cheetah

Results and Discussion

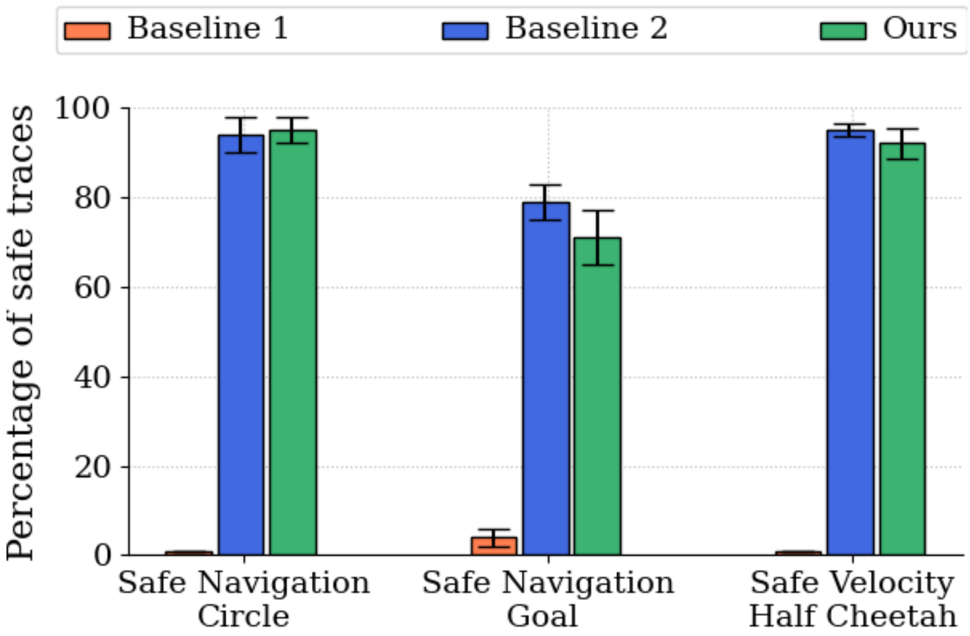
Metrics from the conclusion of training averaged over 3 random seeds

	Baseline 1		Baseline 2		Ours	
	$\overline{\mathcal{J}}_R$	$\overline{\mathcal{J}}_c$	$\overline{\mathcal{J}}_R$	$\overline{\mathcal{J}}_c$	$\overline{\mathcal{J}}_R$	$\overline{\mathcal{J}}_c$
Safe Navigation Circle	111.3	390.3	54.90	1.41	57.02	8.39
Safe Navigation Goal	28.2	48.8	11.5	4.9	16.5	24.3
Safe Velocity Half Cheetah	10371.1	957.6	2676.1	1.67	2114.7	0.62

- Qualitative counterpart to the learning curves presented previously

Results and Discussion

- The policy optimized under baseline 1 fails to produce safe trajectories in case studies 2 and 3, with only a few safe trajectories in case study 2
- In contrast, the policy optimized through our framework yields a number of safe trajectories comparable to baseline 2, which had complete knowledge of the safety constraints from the start



Results and Discussion

	MCR	
	Baseline 2	Ours
Safe Navigation Circle	0.0	0.0251
Safe Navigation Goal	0.0	0.0534
Safe Velocity Half Cheetah	0.0	0.0

- We assessed the STL's quality by its ability to accurately classify labeled data, and then benchmarked these results against the performance of the True STL used in baseline 2
- The true STL safety specification (as expected) classifies all traces with an MCR of zero
- The STL derived through our algorithm closely parallels this standard

Limitations

- Reliance on pre-existing datasets of safe and unsafe trajectories, however small, as well as an STL safety specification template
- The requirement for human expert manual labeling of trajectories
- No guarantees of a safe policy

RL + Foundation Models

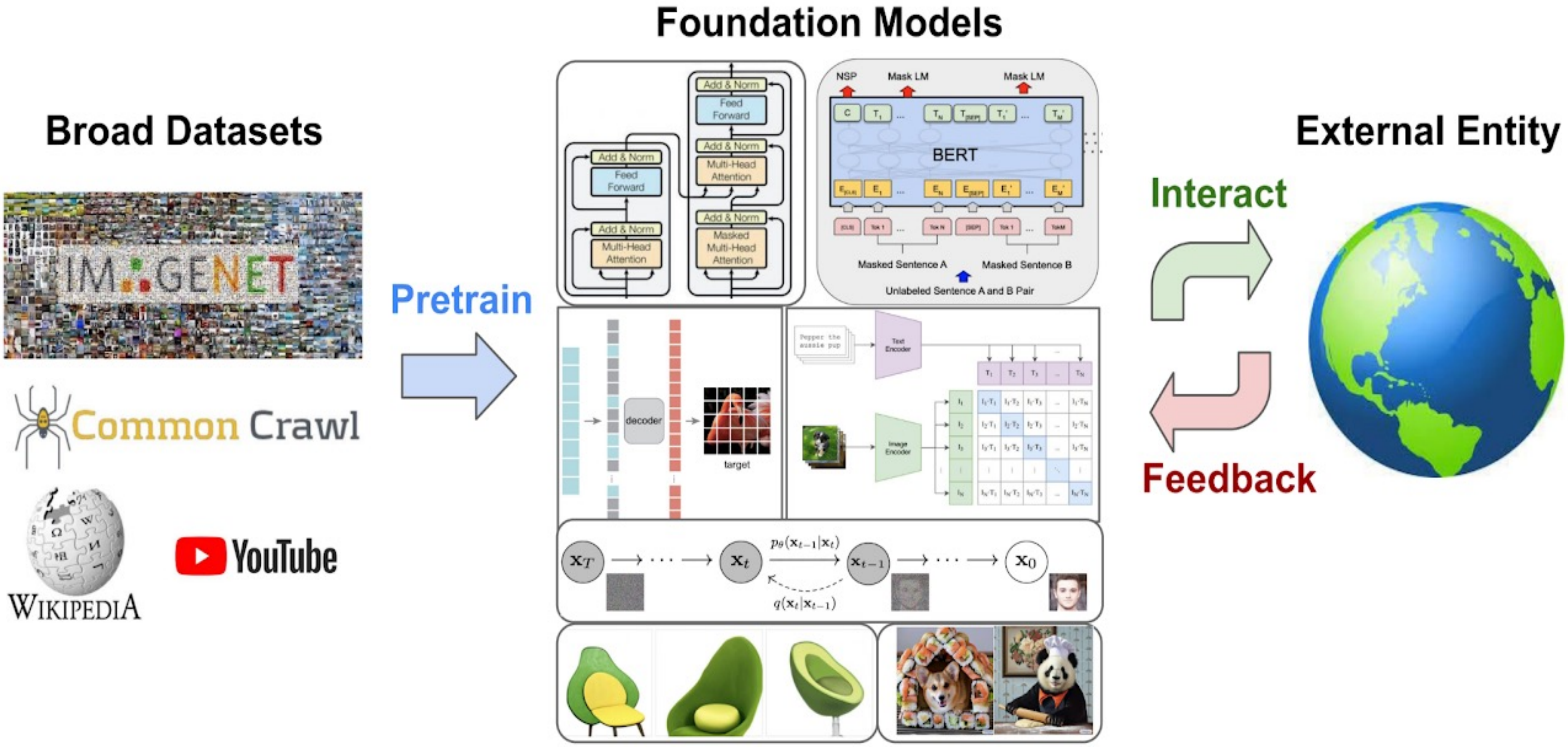
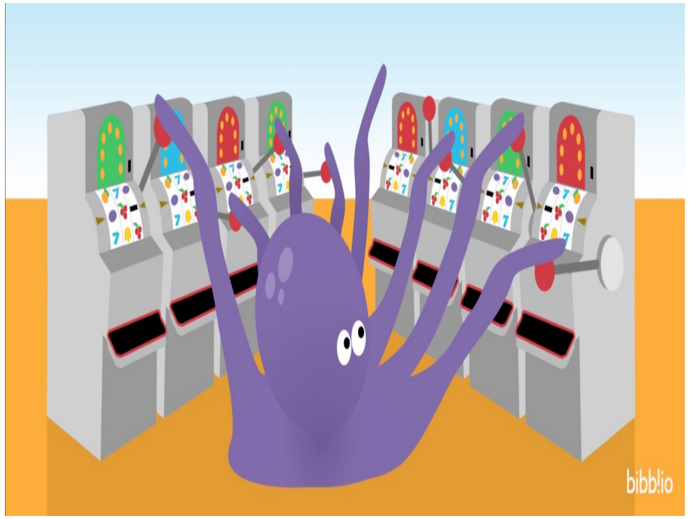


Image credit: Foundation Models for Decision Making Website

LLMs-augmented Contextual Bandit

Ali Baheri
Department of Mechanical Engineering
Rochester Institute of Technology
akbeme@rit.edu

Cecilia O. Alm
Department of Psychology
Rochester Institute of Technology
cecilia.o.alm@rit.edu



Foundation Models for Decision Making Workshop at NeurIPS 2023

Thank you!