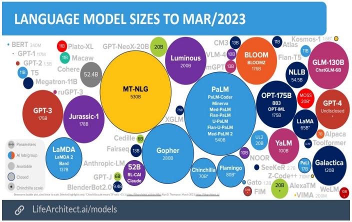
Trustworthy Reinforcement Learning: Explainability and Risk-Awareness

Bo Liu

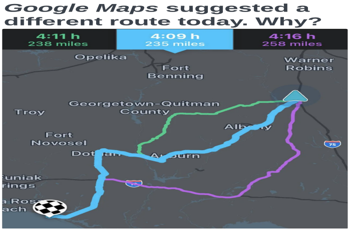
Amazon.com, WA, USA

Artificial Intelligence is the New Electricity



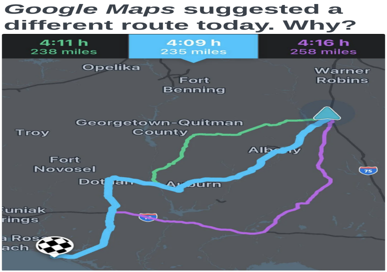
- **Artificial Intelligence** becomes ubiquitous.
- **Ubiquitous** $\neq$ **Trustworthy**!

Artificial Intelligence is the New Electricity



- **Artificial Intelligence** becomes ubiquitous.
- **Ubiquitous** $\neq$ **Trustworthy**!
  - **Why** should I trust it?
  - **When** should I trust it?

Challenge I: Explainability



- **Why** should I trust it?
- Explainability refers to understanding the predictions, decisions, and behaviors.

Challenge I: Explaining the Behaviors

- Question:
  How to explain agents' behaviors clearly?

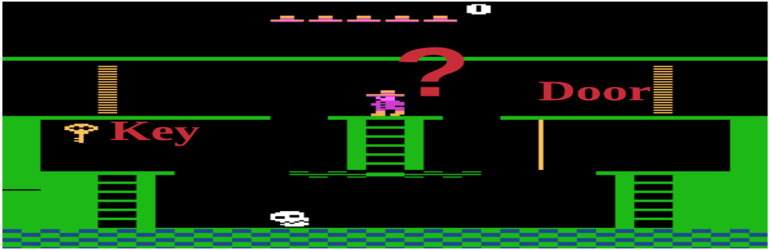
Challenge I: Explaining the Behaviors

- Question:
  How to explain agents' behaviors clearly?
- Default Answer:
  1. Build a policy model.
  2. Explain the model directly.
  3. Done.

Challenge I: Explaining the Behaviors

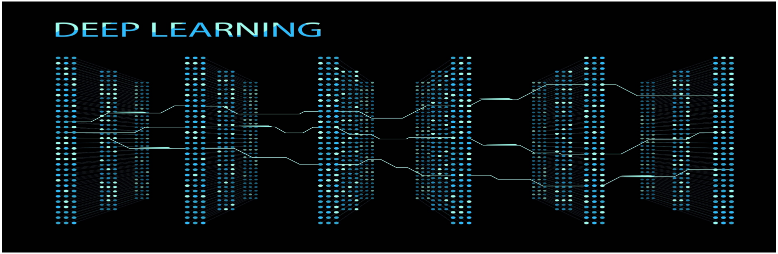
- Behaviors are highly uncertain.

Challenge I: Explaining the Behaviors



- Behaviors are highly uncertain.
- Behaviors have long-term temporal dependencies.

Challenge I: Explaining the Behaviors



- Behaviors are highly uncertain.
- Behaviors have long-term temporal dependencies.
- Behavior models are high-dimensional.

How to explain agents' behaviors clearly?

Challenge II: Risk-Awareness

Tesla Autopilot crash driver 'was playing video game'

26 February 2020

The driver of the Tesla Model X died shortly after the crash

- **When** should I trust it?
- Risk refers to the volatility of future returns.

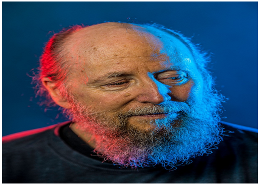
Challenge II: Risk-Awareness in Diverse Scenarios

- Question:
  How to ensure risk-awareness?

Challenge II: Risk-Awareness in Diverse Scenarios

- Question:
  How to ensure risk-awareness?
- Default Answer:
  1. Read the risk-oblivious algorithm.
  2. Design a risk-aware version.
  3. Done.

Challenge II: Risk-Awareness in Diverse Scenarios



The deadly triad

- The risk of divergence arises whenever we combine three things:
  1. Function approximation
     significantly generalizing from large numbers of examples
  2. Bootstrapping
     learning value estimates from other value estimates, as in dynamic programming and temporal-difference learning
  3. Off-policy learning (Why is dynamic programming off-policy?)
     learning about a policy from data not due to that policy, as in Q-learning, where we learn about the greedy policy from data with a necessarily more exploratory policy

- Diverse decision-making scenarios.
  1. Online vs. Offline
  2. Bootstrap vs. Monte-Carlo
  3. Tabular vs. Function Approximation (e.g., DNNs)
  4. Sequential vs. Non-sequential (i.e., Bandits)
  5. Markovian vs. Non-Markovian
  6. $\cdots$ and many more $\cdots$

Challenge II: Risk-Awareness in Diverse Scenarios

Algorithm	Description	Policy	Action space	State space	Operator
DDPG	Deep Deterministic Policy Gradient	Off-policy	Continuous	Continuous	Action-value
A3C	Asynchronous Advantage Actor-Critic Algorithm	On-policy	Discrete	Continuous	Advantage (=action-value - state-value)
TRPO	Trust Region Policy Optimization	On-policy	Continuous or Discrete	Continuous	Advantage
PPO	Proximal Policy Optimization	On-policy	Continuous or Discrete	Continuous	Advantage
TD3	Twin Delayed Deep Deterministic Policy Gradient	Off-policy	Continuous	Continuous	Action-value
SAC	Soft Actor-Critic	Off-policy	Continuous	Continuous	Advantage
DSAC[1][2][3]	Distributional Soft Actor Critic	Off-policy	Continuous	Continuous	Action-value distribution

- Diverse decision-making scenarios.
- Diverse fundamentally different algorithms.
- Significant nontransferable effort to "robustify" each algorithm.

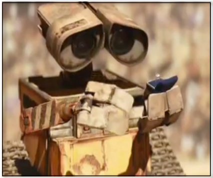
How to ensure risk-awareness across diverse scenarios?

Summary of Challenges

- How to explain agents' behaviors clearly?
- How to ensure risk-awareness across diverse scenarios?

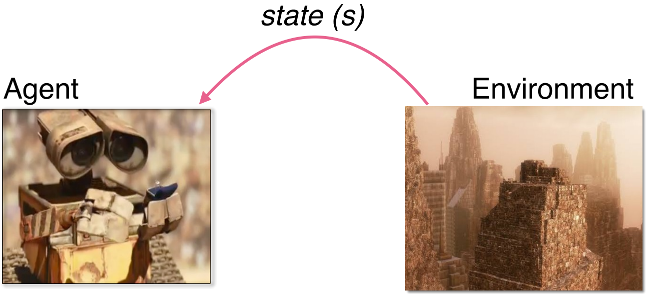
Background: Reinforcement Learning

Agent

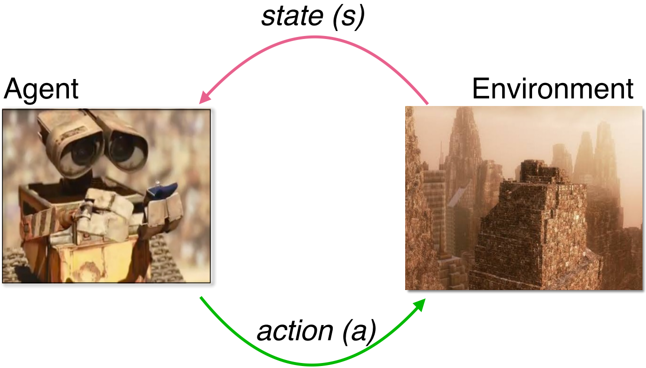


Environment

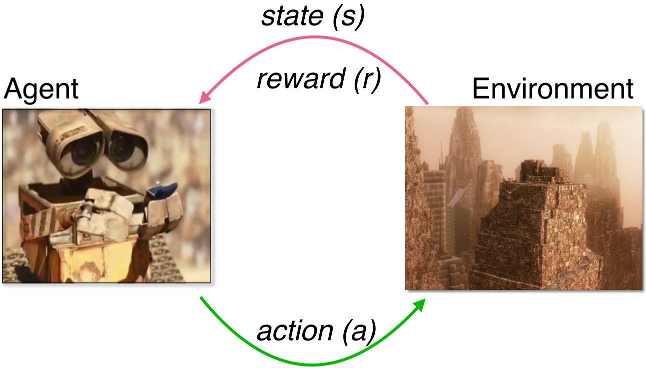
Background: Reinforcement Learning



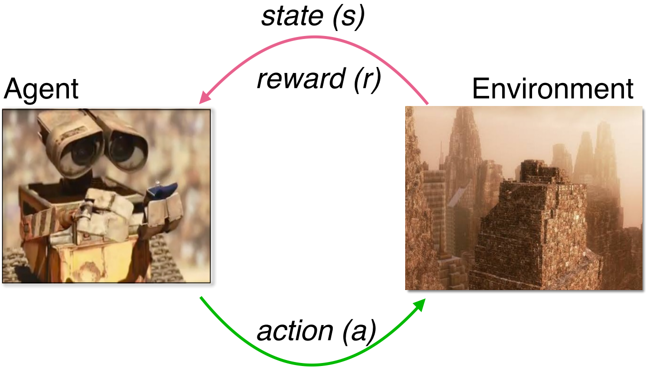
Background: Reinforcement Learning



Background: Reinforcement Learning



Background: Reinforcement Learning



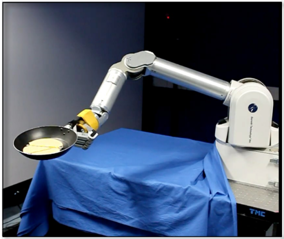
- The goal of **Reinforcement Learning (RL)** is to learn a policy $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ that maximizes $J(\pi) := \mathbb{E}_\pi[R]$, where R is the *cumulative rewards*.

$$R := \sum_{i=0}^{T} \gamma^i r_{s_i,a_i}$$

Examples: Reinforcement Learning in Robotics

Robotics

- **State**: vector about the robot's current configuration.
- **Action**: how to move each joint.
- **Reward**: $+1$ if flip is successful; 0 if pancake falls off the pan.

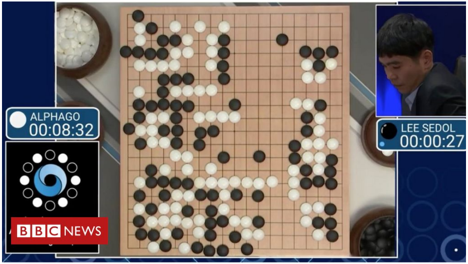


(Kormushev et al., IROS 2010)

Examples: Reinforcement Learning in Games

AlphaGo

- **State**: current board configuration and game context.
- **Action**: selection of a move on the board.
- **Reward**: $+1$ for winning, -1 for losing; 0 otherwise.

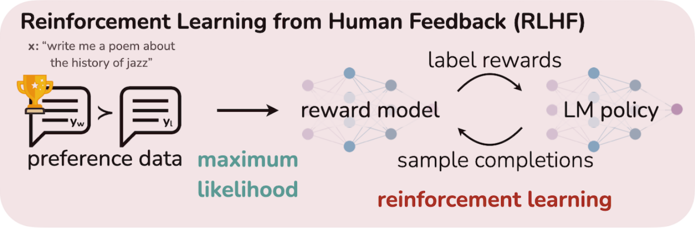


(Silver et al., Nature 2016)

Examples: Reinforcement Learning from Human Feedback

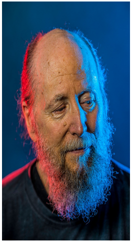
Large Language Models (LLMs)

- **State**: vector by tokenizing the prompt.
- **Action**: the response to the given prompt.
- **Preference**: human-provided pairwise comparisons or list-wise rankings of the responses.
- **Reward**: A reward model is learned from the preferences.



(Rafailov et al., NeurIPS 2023)

Core Problems: Diverse Scenarios of RL



The deadly triad

- The risk of divergence arises whenever we combine three things:

1. Function approximation

   significantly generalizing from large numbers of examples

2. Bootstrapping

   learning value estimates from other value estimates,
   as in dynamic programming and temporal-difference learning

3. Off-policy learning (Why is dynamic programming off-policy?)

   learning about a policy from data not due to that policy,
   as in Q-learning, where we learn about the greedy policy from
   data with a necessarily more exploratory policy

Algorithm	Description	Policy	Action space	State space	Operator
DDPG	Deep Deterministic Policy Gradient	Off-policy	Continuous	Continuous	Action-value
A3C	Asynchronous Advantage Actor-Critic Algorithm	On-policy	Discrete	Continuous	Advantage (=action-value - state-value)
TRPO	Trust Region Policy Optimization	On-policy	Continuous or Discrete	Continuous	Advantage
PPO	Proximal Policy Optimization	On-policy	Continuous or Discrete	Continuous	Advantage
TD3	Twin Delayed Deep Deterministic Policy Gradient	Off-policy	Continuous	Continuous	Action-value
SAC	Soft Actor-Critic	Off-policy	Continuous	Continuous	Advantage
DSAC[1][2][3]	Distributional Soft Actor Critic	Off-policy	Continuous	Continuous	Action-value distribution

- RL has diverse scenarios.
- Diverse scenarios lead to diverse algorithms.

Core Problems: Universal Tasks of RL

Two basic learning tasks of RL are for any scenarios and algorithms.

- **Policy Evaluation (PE)** is to *evaluate* π

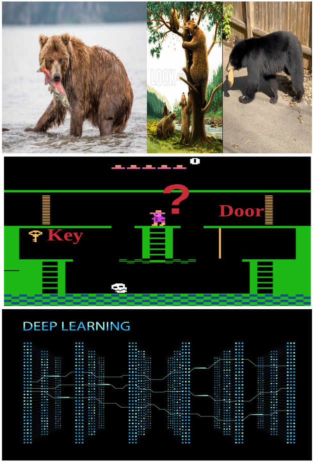
$$\mathbb{E}_\pi[R] = \frac{1}{(1-\gamma)}\mathbb{E}_\pi[r_{s,a}]$$

- **Policy Optimization (PO)** is to *improve* π

$$\pi^* = \arg\max_\pi \mathbb{E}_\pi[R] = \arg\max_\pi \mathbb{E}_\pi[r_{s,a}].$$

- Both are about the policy π!

Challenge I: Explaining the Behaviors



- Behaviors are highly uncertain.
- Behaviors have long-term temporal dependencies.
- Behavior models are high-dimensional.

Challenge I: Explaining the Behaviors

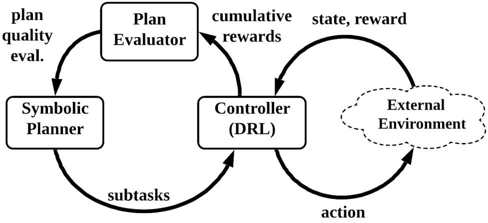
Question:

- How to explain agents' behaviors?

Research Motif:

- **Clear tasks, clear deeds.**

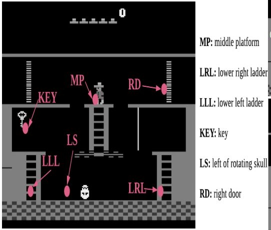
TERL: Task-Level Explainable Reinforcement Learning

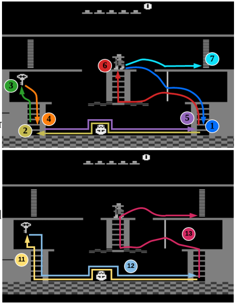


- **Symbolic Planner**: orchestrates a sequence of *subtasks* via a symbolic plan.
- **Controller**: fulfills each subtask.
- **Plan Evaluator**: evaluates the plan's rewarding level.

Task Discovery: Construct Subtasks

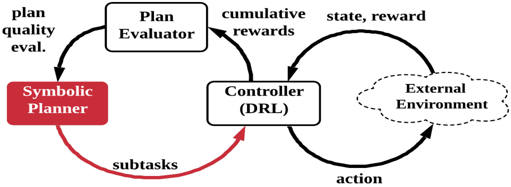
- A subtask dictionary is constructed containing intuitive subtasks.
  1. Location identification via bounding-box segmentation.
  2. Connecting different locations.
  3. Symbolic AI for subtask representation.





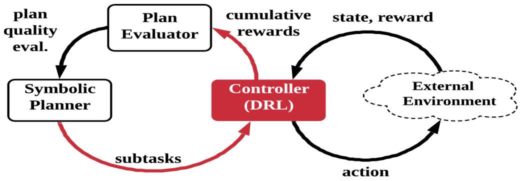
No.	subtask
1	MP to LRL, no key
2	LRL to LLL, no key
3	LLL to key, no key
4	key to LLL, with key
5	LLL to LRL, with or without key
6	LRL to MP, with or without key
7	MP to RD, with key
8	LRL to LS, with or without key
9	LS to key, with or without key
10	MP to RD, no key
11	LRL to key, with or without
12	key to LRL, with key
13	LRL to RD, with key

Symbolic Planner: Generate a Plan



- A **plan** is a sequence of several subtasks.
- At the t-th iteration, Symbolic Planner generates a new plan Φ_t that
  - **explores** new subtasks,
  - **exploits** more rewarding subtasks.

Controller: Fulfill Subtasks of a Plan



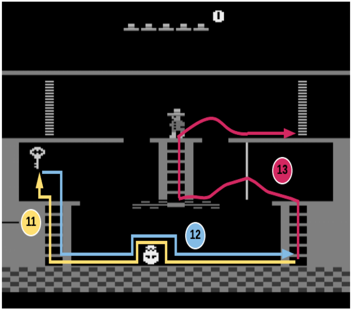
- DRL Controller learns to fulfill subtasks and adapts to uncertainties.
- Any off-the-shelf RL model can be adopted.
- For the i-th subtask g_i, **cumulative rewards** R_i and **success ratio** ϵ_i measure the rewarding level and "hardness". E.g., given a threshold,

$$R_i \leftarrow \begin{cases} 0 & \epsilon_i < threshold \\ R_i & otherwise \end{cases}$$

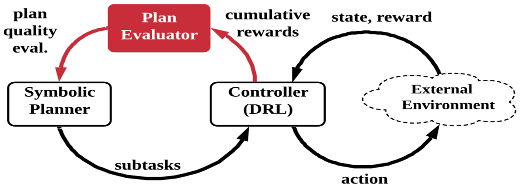
Controller: Fulfill Subtasks of a Plan

- DRL controller prunes infeasible subtasks.

No.	subtask	policy learned	in optimal plan
1	MP to LRL, no key	√	√
2	LRL to LLL, no key	√	√
3	LLL to key, no key	√	√
4	key to LLL, with key	√	√
5	LLL to LRL, with or without key	√	√
6	LRL to MP, with or without key	√	√
7	MP to RD, with key	√	√
8	LRL to LS, with or without key	√	
9	LS to key, with or without key	√	
10	RD to MP, no key	√	
11	~~LRL to key, with or without key~~		
12	~~key to LRL, with key~~		
13	~~LRL to RD, with key~~		



Plan Evaluator: Evaluate a Plan



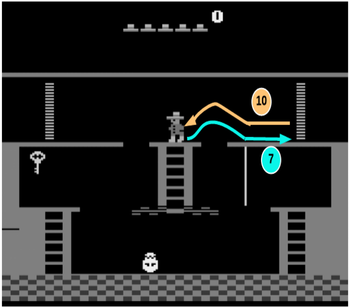
- Plan Evaluator evaluates the plan Φ_t,

$$\textbf{Evaluation}(\Phi_t) = \sum_{g_i \in \Phi_t} R_i.$$

Symbolic Planner + Plan Evaluator

- DRL controller prunes infeasible tasks.
- Symbolic planner and plan evaluator jointly prune non-essential subtasks, i.e., irrelevant and redundant.

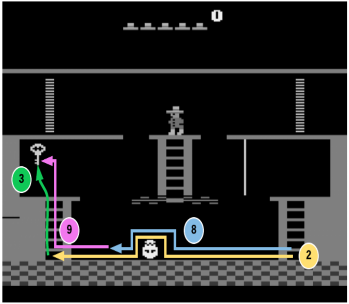
No.	subtask	policy learned	in optimal plan
1	MP to LRL, no key	√	√
2	LRL to LLL, no key	√	√
3	LLL to key, no key	√	√
4	key to LLL, with key	√	√
5	LLL to LRL, with or without key	√	√
6	LRL to MP, with or without key	√	√
7	MP to RD, with key	√	√
8	LRL to LS, with or without key	√	
9	LS to key, with or without key	√	
10	~~RD to MP, no key~~	√	
11	LRL to key, with or without key		
12	key to LRL, with key		
13	LRL to RD, with key		



Symbolic Planner + Plan Evaluator

- DRL controller prunes infeasible tasks.
- Symbolic planner and plan evaluator jointly prune non-essential subtasks, i.e., irrelevant and redundant.

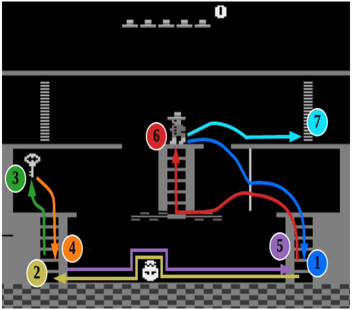
No.	subtask	policy learned	in optimal plan
1	MP to LRL, no key	√	√
2	LRL to LLL, no key	√	√
3	LLL to key, no key	√	√
4	key to LLL, with key	√	√
5	LLL to LRL, with or without key	√	√
6	LRL to MP, with or without key	√	√
7	MP to RD, with key	√	√
8	~~LRL to LS, with or without key~~	√	
9	~~LS to key, with or without key~~	√	
10	RD to MP, no key	√	
11	LRL to key, with or without key		
12	key to LRL, with key		
13	LRL to RD, with key		



Experimental Results

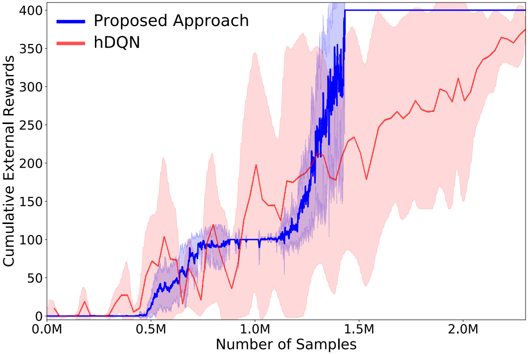
- DRL controller prunes infeasible tasks.
- Symbolic planner and plan evaluator jointly prune non-essential subtasks, i.e., irrelevant and redundant.
- The final plan's subtasks are intuitive, feasible, and essential!

No.	subtask	policy learned	in optimal plan
1	MP to LRL, no key	√	√
2	LRL to LLL, no key	√	√
3	LLL to key, no key	√	√
4	key to LLL, with key	√	√
5	LLL to LRL, with or without key	√	√
6	LRL to MP, with or without key	√	√
7	MP to RD, with key	√	√
8	~~LRL to LS, with or without key~~	√	
9	~~LS to key, with or without key~~	√	
10	~~RD to MP, no key~~	√	
11	~~LRL to key, with or without key~~		
12	~~key to LRL, with key~~		
13	~~LRL to RD, with key~~		

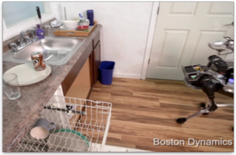


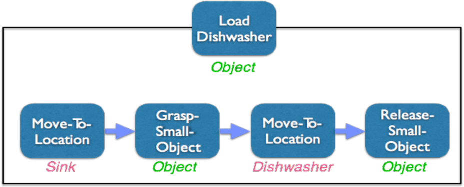
Experimental Results

- TERL saves more than 0.5 million training samples compared with hDQN.
- TERL can greatly reduce the variance.



Ongoing: LLM-Aided Task Discovery





Different methods for subtask discovery:

- Vision-based location segmentation.
- Transformer-based tokenization + time series change-point.
- LLM for subtask description.

Highlights

Task-level **E**xplainable **RL** (**TERL**) [**IJCAI'18,AAAI'19,ICLP'19,IEEE-TETCI'21,IEEE-TNNLS'23,CISS'2023, Liu**].

- Question: How to explain agents' behaviors?
- Research motif: **Clear tasks, clear deeds.**
- Key ingredient: Neuro-symbolic AI for data-efficiency and explainability.
- Key ingredient: Task-level explainability.

Impacts

- **Inspiring later research:**

Learning Neuro-Symbolic Skills for Bilevel Planning

Tom Silver, Ashay Athalye
Joshua B. Tenenbaum, Tomás Lozano-Pérez, Leslie Pack Kaelbling
MIT Computer Science and Artificial Intelligence Laboratory
{tslvr, ashay, jbt, tlp, lpk}@mit.edu

iCORPP: Interleaved Commonsense Reasoning and Probabilistic Planning on Robots

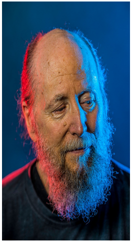
Shiqi Zhang[a,*], Piyush Khandelwal[b] and Peter Stone[c,b]
[a]The State University of New York at Binghamton, 4400 Vestal Parkway East, Binghamton, 13902, NY, USA
[b]Sony AI, Sony Corporation of America, 25 Madison Avenue, New York, 10010, NY, USA
[c]The University of Texas at Austin, 2317 Speedway, Stop D9500, Austin, 78712, TX, USA

Deep Explainable Relational Reinforcement Learning: A Neuro-Symbolic Approach

Rishi Hazra[1] ✉ and Luc De Raedt[1,2]
[1] Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden
[2] Department of Computer Science, KU Leuven, Belgium
{rishi.hazra,luc.de-raedt}@oru.se

- **Featured in surveys:**
  - *Neural Networks*.
  - *AI Magazine*.
  - *ACM Computing Surveys*.
  - ...

Challenge II: Risk-Awareness in Diverse Scenarios



The deadly triad

- The risk of divergence arises whenever we combine three things:

1. Function approximation

   significantly generalizing from large numbers of examples

2. Bootstrapping

   learning value estimates from other value estimates,
   as in dynamic programming and temporal-difference learning

3. Off-policy learning (Why is dynamic programming off-policy?)

   learning about a policy from data not due to that policy,
   as in Q-learning, where we learn about the greedy policy from
   data with a necessarily more exploratory policy

Algorithm	Description	Policy	Action space	State space	Operator
DDPG	Deep Deterministic Policy Gradient	Off-policy	Continuous	Continuous	Action-value
A3C	Asynchronous Advantage Actor-Critic Algorithm	On-policy	Discrete	Continuous	Advantage (=action-value - state-value)
TRPO	Trust Region Policy Optimization	On-policy	Continuous or Discrete	Continuous	Advantage
PPO	Proximal Policy Optimization	On-policy	Continuous or Discrete	Continuous	Advantage
TD3	Twin Delayed Deep Deterministic Policy Gradient	Off-policy	Continuous	Continuous	Action-value
SAC	Soft Actor-Critic	Off-policy	Continuous	Continuous	Advantage
DSAC[1][2][3]	Distributional Soft Actor Critic	Off-policy	Continuous	Continuous	Action-value distribution

- RL has diverse scenarios.
- Diverse scenarios lead to diverse risk-oblivious algorithms.

Challenge II: Risk-Awareness in Diverse Scenarios

Question:

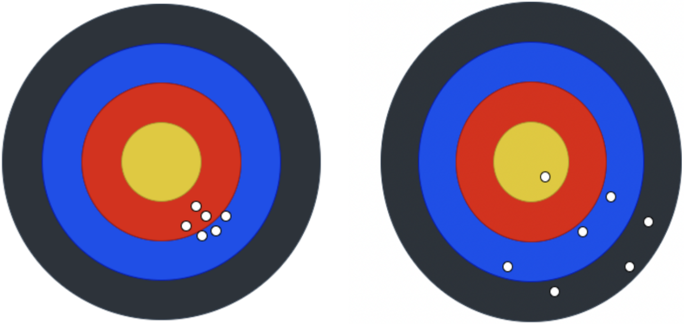
- How to ensure risk-awareness across diverse scenarios?

Research Motif:

- **Risk-oblivious + "Universally Robustified" = Risk-aware**

Mean Optimization

- $R := \sum_{i=0}^{\tau} \gamma^i r_{s_i,a_i}$. There is $\mathbb{E}_\pi[R] = \frac{1}{(1-\gamma)}\mathbb{E}_\pi[r_{s,a}]$.
- Conventional RL is Mean-Optimization $\max_\pi \mathbb{E}_\pi[R] \sim \mathbb{E}_\pi[r_{s,a}]$.



Mean-Variance Optimization

- $R := \sum_{i=0}^{\tau} \gamma^i r_{s_i,a_i}$. There is $\mathbb{E}_\pi[R] = \frac{1}{(1-\gamma)} \mathbb{E}_\pi[r_{s,a}]$.
- Conventional RL is Mean-Optimization $\max_\pi \mathbb{E}_\pi[R] \sim \mathbb{E}_\pi[r_{s,a}]$.
- Mean-Variance Optimization

$$\max_{\pi} \quad J_\lambda(\pi) := \mathbb{E}_\pi[r_{s,a}] - \lambda \mathrm{Var}_\pi(r_{s,a}),$$

  where $\lambda \geq 0$ is a variance-controlling hyper-parameter.
- **Variance reflects volatility and risks**.

A Duality Perspective of Mean-Variance

- $J_\lambda(\pi)$ is

$$\begin{aligned}
J_\lambda(\pi) =& \mathbb{E}_\pi[r_{s,a}] - \lambda \mathrm{Var}_\pi(r_{s,a}) \\
=& \mathbb{E}_\pi[r_{s,a}] - \lambda \mathbb{E}_\pi[r_{s,a}^2] + \lambda \big(\mathbb{E}_\pi[r_{s,a}]\big)^2
\end{aligned}$$

A Duality Perspective of Mean-Variance

- $J_\lambda(\pi)$ is

$$
\begin{aligned}
J_\lambda(\pi) =& \mathbb{E}_\pi[r_{s,a}] - \lambda \mathrm{Var}_\pi(r_{s,a}) \\
=& \mathbb{E}_\pi[r_{s,a}] - \lambda \mathbb{E}_\pi[r_{s,a}^2] + \lambda \big(\mathbb{E}_\pi[r_{s,a}]\big)^2
\end{aligned}
$$

- Use the fact $\forall x, x^2 = \max_y \left(2xy - y^2\right)$ and let $x = \mathbb{E}_\pi[r_{s,a}]$,

$$
\begin{aligned}
J_\lambda(\pi) =& \mathbb{E}_\pi[r_{s,a}] - \lambda \mathbb{E}_\pi[r_{s,a}^2] + \lambda \max_y \left(2\mathbb{E}_\pi[r_{s,a}]y - y^2\right) \\
=& \max_y \Bigg(\underbrace{\mathbb{E}_\pi[r_{s,a}] - \lambda \mathbb{E}_\pi[r_{s,a}^2] + \lambda \left(2\mathbb{E}_\pi[r_{s,a}]y - y^2\right)}_{J_\lambda(\pi, y)} \Bigg)
\end{aligned}
$$

A Duality Perspective of Mean-Variance

- Now $J_\lambda(\pi) = \max_y J_\lambda(\pi, y)$, where

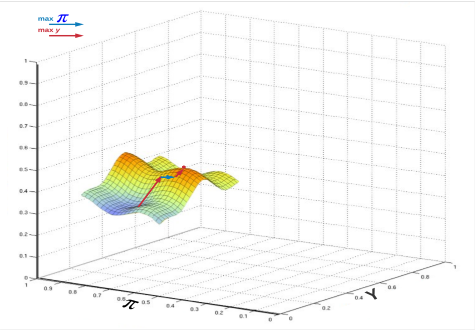
$$J_\lambda(\pi, y) := \mathbb{E}_\pi[r_{s,a}] - \lambda \mathbb{E}_\pi[r_{s,a}^2] + \lambda\big(2\mathbb{E}_\pi[r_{s,a}]y - y^2\big)$$

Coordinate-Maximization

- *Coordinate-Maximization* alternates optimization among variables (i.e., coordinates).

$$y_{t+1} = \arg\max_{y} J_\lambda(\pi_t, y),$$

$$\pi_{t+1} = \arg\max_{\pi} J_\lambda(\pi, y_{t+1}).$$



Coordinate Maximization

- Sub-problems are much lower-dimensional!
- Sub-problems may be well-studied already!

Mean-Variance Policy Iteration (Meta-algorithm)

$$\max_{\pi,y} J_\lambda(\pi, y) = \underbrace{-\lambda}_{A} y^2 + \underbrace{2\lambda\mathbb{E}_\pi\left[r_{s,a}\right]}_{B} y + \mathbb{E}_\pi\left[r_{s,a} - \lambda r_{s,a}^2\right]$$

Algorithm: MVPI (Meta-algorithm)

for $t = 1, \ldots$ **do**

  Step 1: $y_{t+1} = \arg\max_y J_\lambda(\pi_t, y)$

  Step 2: $\pi_{t+1} = \arg\max_\pi J_\lambda(\pi, y_{t+1})$

Mean-Variance Policy Iteration (Meta-algorithm)

$$\max_{\pi,y} J_\lambda(\pi, y) = \underbrace{-\lambda}_{A} y^2 + \underbrace{2\lambda\mathbb{E}_\pi\left[r_{s,a}\right]}_{B} y + \mathbb{E}_\pi\left[r_{s,a} - \lambda r_{s,a}^2\right]$$

$$\arg\max_{y} J_\lambda(\pi, y) = -\frac{B}{2A} = \mathbb{E}_{\pi_t}[r_{s,a}]$$

Algorithm: MVPI (Meta-algorithm)

for $t = 1, \ldots$ **do**

  Step 1: $y_{t+1} = \arg\max_y J_\lambda(\pi_t, y) = \mathbb{E}_{\pi_t}[r_{s,a}]$

  Step 2: $\pi_{t+1} = \arg\max_\pi J_\lambda(\pi, y_{t+1})$

Background: Universal Tasks of RL

Two basic learning tasks of RL are for any scenarios and algorithms.

- **Policy Evaluation (PE)** is to *evaluate* π

$$\mathbb{E}_\pi[R] = \frac{1}{(1-\gamma)}\mathbb{E}_\pi[r_{s,a}]$$

- **Policy Optimization (PO)** is to *improve* π

$$\pi^* = \arg\max_\pi \mathbb{E}_\pi[R] = \arg\max_\pi \mathbb{E}_\pi[r_{s,a}].$$

- Both are about the policy π!

Mean-Variance Policy Iteration (Meta-algorithm)

$$\max_{\pi,y} J_\lambda(\pi, y) = \underbrace{-\lambda}_{A} y^2 + \underbrace{2\lambda\mathbb{E}_\pi\left[r_{s,a}\right]}_{B} y + \mathbb{E}_\pi\left[r_{s,a} - \lambda r_{s,a}^2\right]$$

$$\arg\max_y J_\lambda(\pi, y) = -\frac{B}{2A} = \mathbb{E}_{\pi_t}[r_{s,a}]$$

Algorithm: MVPI (Meta-algorithm)

for $t = 1, \ldots$ **do**

- **Step 1**: $y_{t+1} = \arg\max_y J_\lambda(\pi_t, y) = \mathbb{E}_{\pi_t}[r_{s,a}]$
  $= (1-\gamma)\mathbb{E}_\pi[R] = (1-\gamma)\mathbf{PE}(\pi_t)$
- Plug&play any policy evaluation (PE).
- **Step 2**: $\pi_{t+1} = \arg\max_\pi J_\lambda(\pi, y_{t+1})$

Mean-Variance Policy Iteration (Meta-algorithm)

$$\max_{\pi,y} J_\lambda(\pi, y) = \mathbb{E}_\pi\left[\underbrace{r_{s,a} - \lambda r_{s,a}^2 + 2\lambda r_{s,a} y}_{\bar{r}_{s,a}(y)}\right] - \lambda y^2.$$

Algorithm: MVPI (Meta-algorithm)

for $t = 1, \ldots$ **do**

- **Step 1**: $y_{t+1} = (1-\gamma)\mathbf{PE}(\pi_t)$
  Plug&play any policy evaluation (PE).
- **Step 2**: $\pi_{t+1} = \arg\max_\pi J_\lambda(\pi, y_{t+1})$
  $= \arg\max_\pi \mathbb{E}_\pi[\bar{r}_{s,a}(y_{t+1})]$

Background: Universal Tasks of RL

Two basic learning tasks of RL are for any scenarios and algorithms.

- **Policy Evaluation (PE)** is to *evaluate* π

$$\mathbb{E}_{\pi}[R] = \frac{1}{(1-\gamma)}\mathbb{E}_{\pi}[r_{s,a}]$$

- **Policy Optimization (PO)** is to *improve* π

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}[R] = \arg\max_{\pi} \mathbb{E}_{\pi}[r_{s,a}].$$

- Both are about the policy π!

Mean-Variance Policy Iteration (Meta-algorithm)

$$\max_{\pi,y} J_\lambda(\pi, y) = \mathbb{E}_\pi\Big[\underbrace{r_{s,a} - \lambda r_{s,a}^2 + 2\lambda r_{s,a} y}_{\bar{r}_{s,a}(y)}\Big] - \lambda y^2.$$

Algorithm: MVPI (Meta-algorithm)

for $t = 1, \ldots$ **do**

- **Step 1**: $y_{t+1} = (1-\gamma)\mathbf{PE}(\pi_t)$
  Plug&play any policy evaluation (PE).
- **Step 2**: $\pi_{t+1} = \arg\max_\pi J_\lambda(\pi, y_{t+1})$
  $= \arg\max_\pi \mathbb{E}_\pi[\bar{r}_{s,a}(y_{t+1})]$
  $= \mathbf{PO}(y_{t+1})$
  Plug&play any policy optimization (PO).

From a Meta-Algorithm to (Many) Algorithms

General meta-algorithm:

Algorithm: MVPI (Meta-algorithm)

for $t = 1, \ldots$ **do**

Step 1: $y_{t+1} = (1-\gamma)\mathbf{PE}(\pi_t)$

Plug&play any policy evaluation (PE).

Step 2: $\pi_{t+1} = \mathbf{PO}(y_{t+1})$

Plug&play any policy optimization (PO).

From a Meta-Algorithm to (Many) Algorithms

An example algorithm:

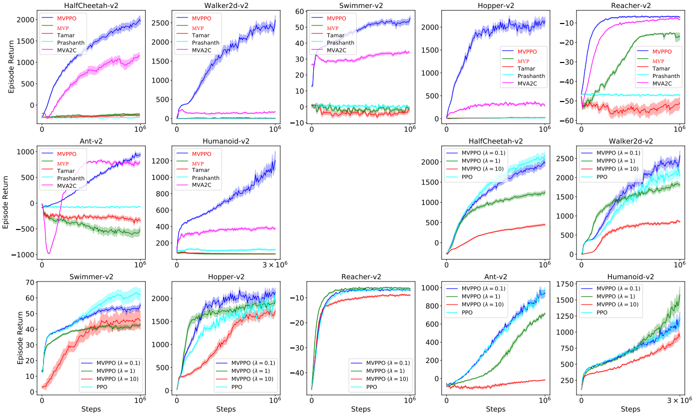
Algorithm: Mean-Variance Proximal Policy Optimization

for $t = 1, \ldots$ **do**

- **Step 1**: $y_{t+1} = (1 - \gamma)\mathbf{PE}(\pi_t)$
  Plug&play Backward-TD method. [**AAAI'24, Liu**]
- **Step 2**: $\pi_{t+1} = \mathbf{PO}(y_{t+1})$
  Plug&play Proximal Policy Optimization (PPO).

Experimental Results

- A better scalability compared with other risk-aware RL algorithms.
- A better variance control compared with vanilla risk-oblivious methods.



Highlights

Mean-**V**ariance **P**olicy Iteration(**MVP**) [**NeurIPS'2018, JAIR'2018, ICML'2020a,b, AAAI'2021, AAMAS'2022, AAAI'2024,Liu**].

- Question: How to ensure risk-awareness across diverse scenarios?
- Research motif:
  Risk-oblivious + "Universally Robustified" = Risk-aware
- Key ingredient: Coordinate-Maximization formulation.
- Key ingredient: Black-box plug&play!

Impacts

- **Inspiring later research:**

Average-Reward Off-Policy Policy Evaluation with Function Approximation

Shangtong Zhang[1*] Yi Wan[2*] Richard S. Sutton[2] Shimon Whiteson[1]

Risk-Aware Transfer in Reinforcement Learning using Successor Features

Michael Gimelfarb* University of Toronto mike.gimelfarb@mail.utoronto.ca

André Barreto DeepMind andrebarreto@google.com

Scott Sanner* University of Toronto ssanner@mie.utoronto.ca

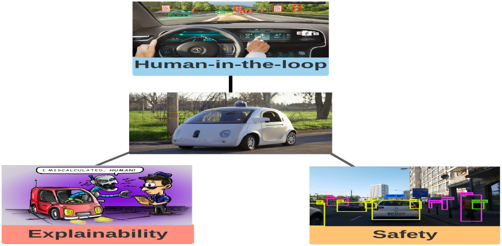
Chi-Guhn Lee University of Toronto cglee@mie.utoronto.ca

An Alternative to Variance: Gini Deviation for Risk-averse Policy Gradient

Yudong Luo[1,4], Guiliang Liu[2], Pascal Poupart[1,4], Yangchen Pan[3]
[1]University of Waterloo, [2]Chinese Universiry of HongKong, Shenzhen, [3]University of Oxford, [4]Vector Institute
yudong.luo@uwaterloo.ca, liuguiliang@cuhk.edu.cn, ppoupart@uwaterloo.ca,yangchen.pan@eng.ox.ac.uk

- **Featured in surveys:**
  - *IEEE Transactions on Neural Networks and Learning Systems (IEEE-TNNLS).*
  - *NeurIPS 2020 Tutorial by Sergy Levine (UC-Berkeley).*
  - *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE-TPAMI).*
  - ...

Future Plan: Explainable and Safe Autonomous Driving

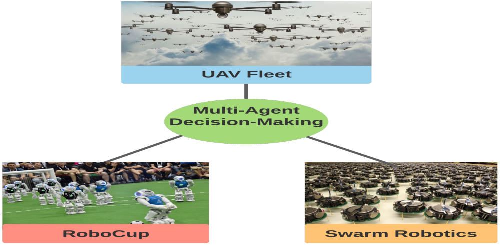


- **Explore** human-in-the-loop to improve understanding and safety.
- **Investigate** challenges such as human interventions, scalability, and real-time.
- **Apply** to ground-true Advanced Driver-Assistance Systems (ADAS).

Future Plan: Trustworthy AI for Social Good

- **Explore** privacy, personalization, and fairness.
- **Investigate** challenges such as LLM integration and multi-modality.
- **Apply** to precision healthcare, service robotics, etc.

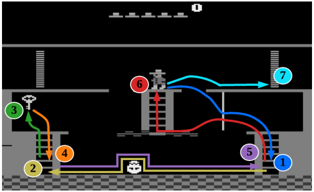
Future Plan: Collective Embodied AI

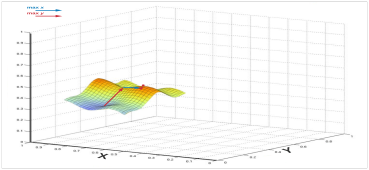


- **Explore** key issues in multi-agent AI, such as decentralization, federated learning, and fairness.
- **Investigate** challenges such as multi-agent, multi-objective, multi-distribution.
- **Apply** to UAVs, multi-robots, etc.

Take-Away Message

- Trustworthy AI primarily involves *Explainability* and *Risk-Awareness*.
- **Clear tasks, clear deeds.**
  - **T**ask-level **E**xplainable **RL** (**TERL**).
  - Key ingredients: Neuro-symbolic AI, task-level explainability.
- **Risk-oblivious + "Universally Robustified" = Risk-aware**
  - **M**ean-**V**ariance **P**olicy Iteration (**MVP**).
  - Key ingredients: Coordinate-maximization, plug&play of universal RL tasks.





Coordinate Maximization

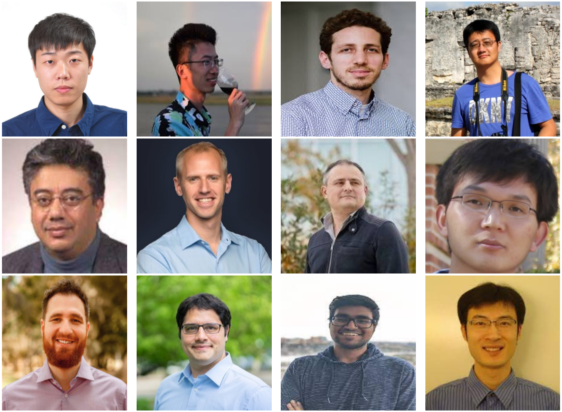
Related Publications

- **[JAIR'18] Liu, B.**, Gemp, I., Ghavamzadeh, M., Liu, J., Mahadevan, S., and Petrik, M. (2018). Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity. *Journal of Artificial Intelligence Research (**JAIR**).*
- **[NeurIPS'18] Liu, B.***, Xie, T.*, Xu, Y., Ghavamzadeh, M., Chow, Y., Lyu, D., and Yoon, D. (2018). A block coordinate ascent algorithm for mean-variance optimization. In *Advances in Neural Information Processing Systems (**NeurIPS**)*. (* equal contributions)
- **[IJCAI'18]** Yang, F., Lyu, D., **Liu, B.**, and Gustafson, S. (2018). PEORL: Integrating symbolic planning and hierarchical reinforcement learning for robust decision-making. In *International Joint Conference of Artificial Intelligence (**IJCAI**)*.
- **[AAAI'19]** Lyu, D., Yang, F., **Liu, B.**, and Gustafson, S. (2019). SDRL: Interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In *AAAI Conference on Artificial Intelligence* (**AAAI**).
- **[ICLP'19]** Lyu, D., Yang, F., **Liu, B.**, and Gustafson, S. (2019). A human-centered data-driven planner-actor-critic architecture via logic programming. In *35th International Conference on Logic Programming (**ICLP**)*.
- **[ICML'20a]** Zhang, S., **Liu, B.**, and Whiteson, S. (2020). Gradientdice: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning (**ICML**)*.
- **[ICML'20b]** Zhang, S., **Liu, B.**, and Whiteson, S. (2020). Provably Convergent Two-Timescale Off-Policy Actor-Critic with Function Approximation. In *International Conference on Machine Learning (**ICML**)*.
- **[AAAI'21]** Zhang, S., **Liu, B.**, and Whiteson, S. (2021). Mean-variance policy iteration for risk-averse reinforcement learning. In *AAAI Conference on Artificial Intelligence (**AAAI**)*.
- **[IEEE-TETCI'21]** Lyu, D., Yang, F., Kwon, H., Dong, W., Yilmaz, **Liu, B.** (2021). TDM: Trustworthy Decision-Making via Interpretability Enhancement. *IEEE Transactions on Emerging Topics in Computational Intelligence (**IEEE-TETCI**)*.

Related Publications

- **[AAMAS'22]** Xu, L., Lyu, D., Pan, Y., Jiang, A., **Liu, B.** (2022). TOPS: Transition-Based Volatility-Reduced Policy Search. *Autonomous Agents and Multiagent Systems (**AAMAS**)*. Best and Visionary Papers Award.
- **[CISS'23]** Xu, C., **Liu, B.**, Zhao, Y. (2023). Offline Reinforcement Learning for Price-Based Demand Response Program Design. *Proc. 57th Annual Conference on Information Sciences and Systems (**CISS**)*.
- **[IEEE-TNNLS'23]** Zhang, Z., Yilmaz, L., **Liu, B.** (2023). A Critical Review of Inductive Logic Programming Techniques for Explainable AI. *IEEE Transactions on Neural Networks and Learning Systems (**IEEE-TNNLS**)*.
- **[AAAI'24]** Gupta, D., Jordan, S., Chaudhari, S., **Liu, B.**, Thomas, P., da Silva, B. C. (2024). From Past to Future: Rethinking Eligibility Traces. *Thirty-Eighth AAAI Conference on Artificial Intelligence (**AAAI**)*.

Collaborators



Thanks for Your Time!

More questions?

Email: ergodic.markovian@gmail.com