Causal Reinforcement Learning

By Anna Zhang

Part I Intro to RL

Main ideas

What is
Reinforcement Learning?

A computational approach to learning whereby an agent tries to maximize the total amount of reward it receives while interacting with a complex and uncertain environment.

What is
Reinforcement Learning?

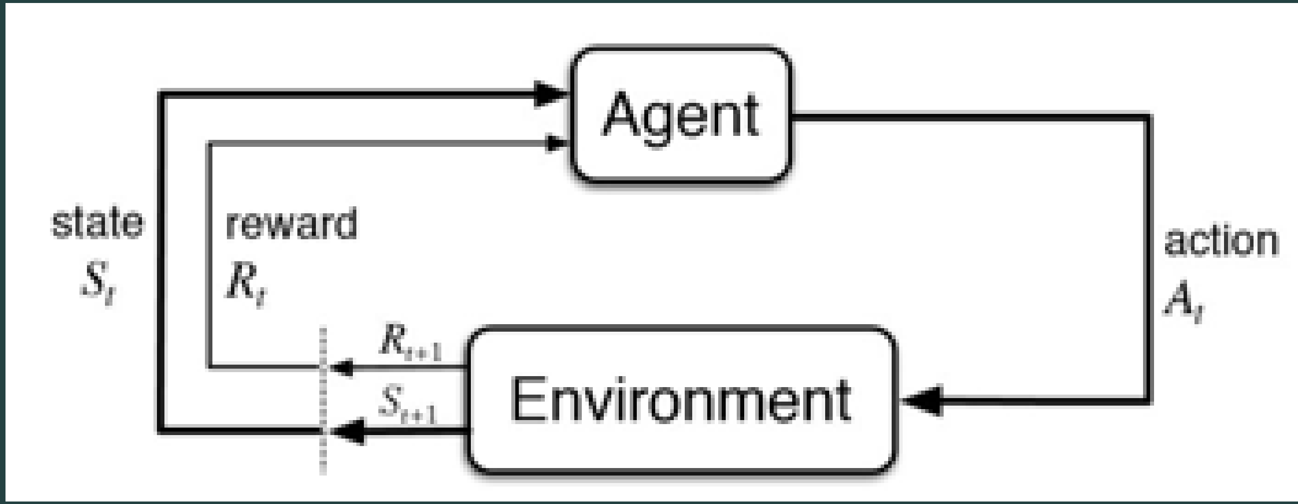
A computational approach to learning whereby an agent tries to maximize the total amount of reward it receives while interacting with a complex and uncertain environment.

What is
Reinforcement Learning?

A computational approach to learning whereby an agent tries to maximize the total amount of reward it receives while interacting with a complex and uncertain environment.

Fundamental Framework for RL

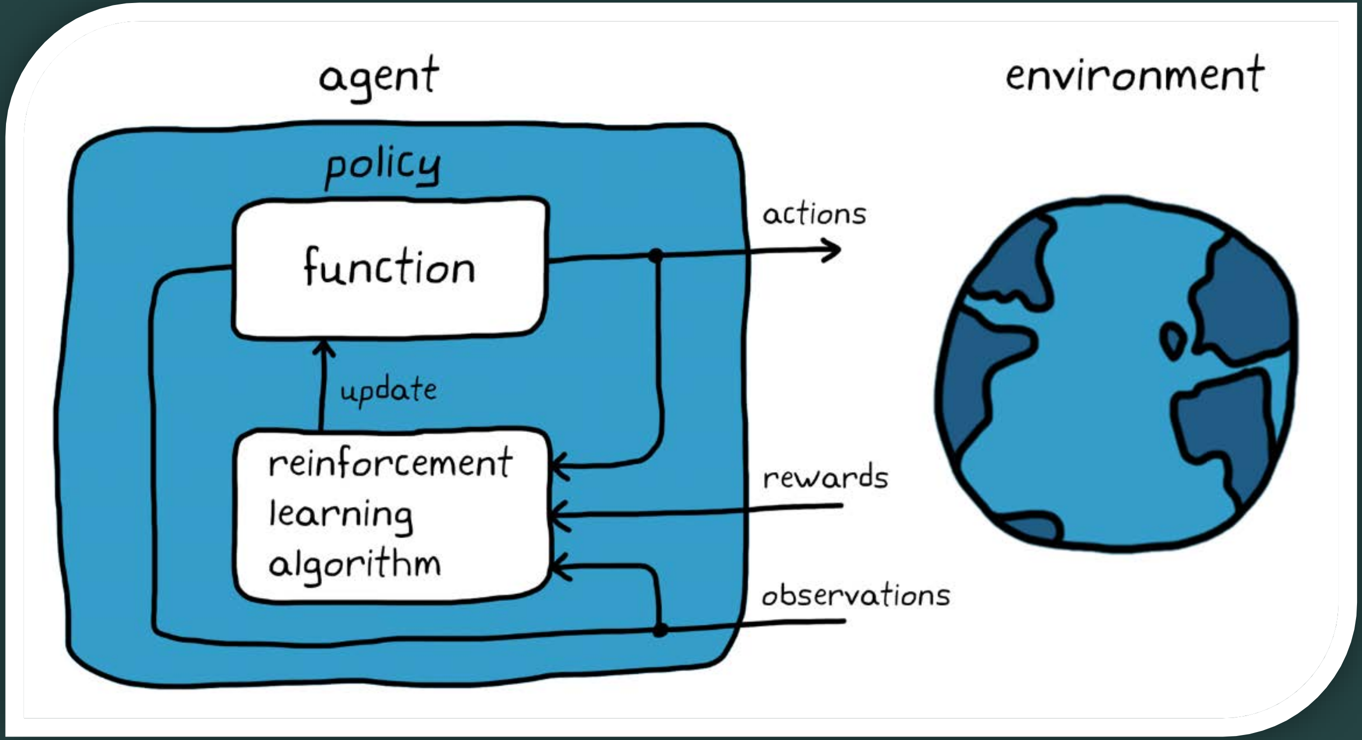
- Agent
- Env



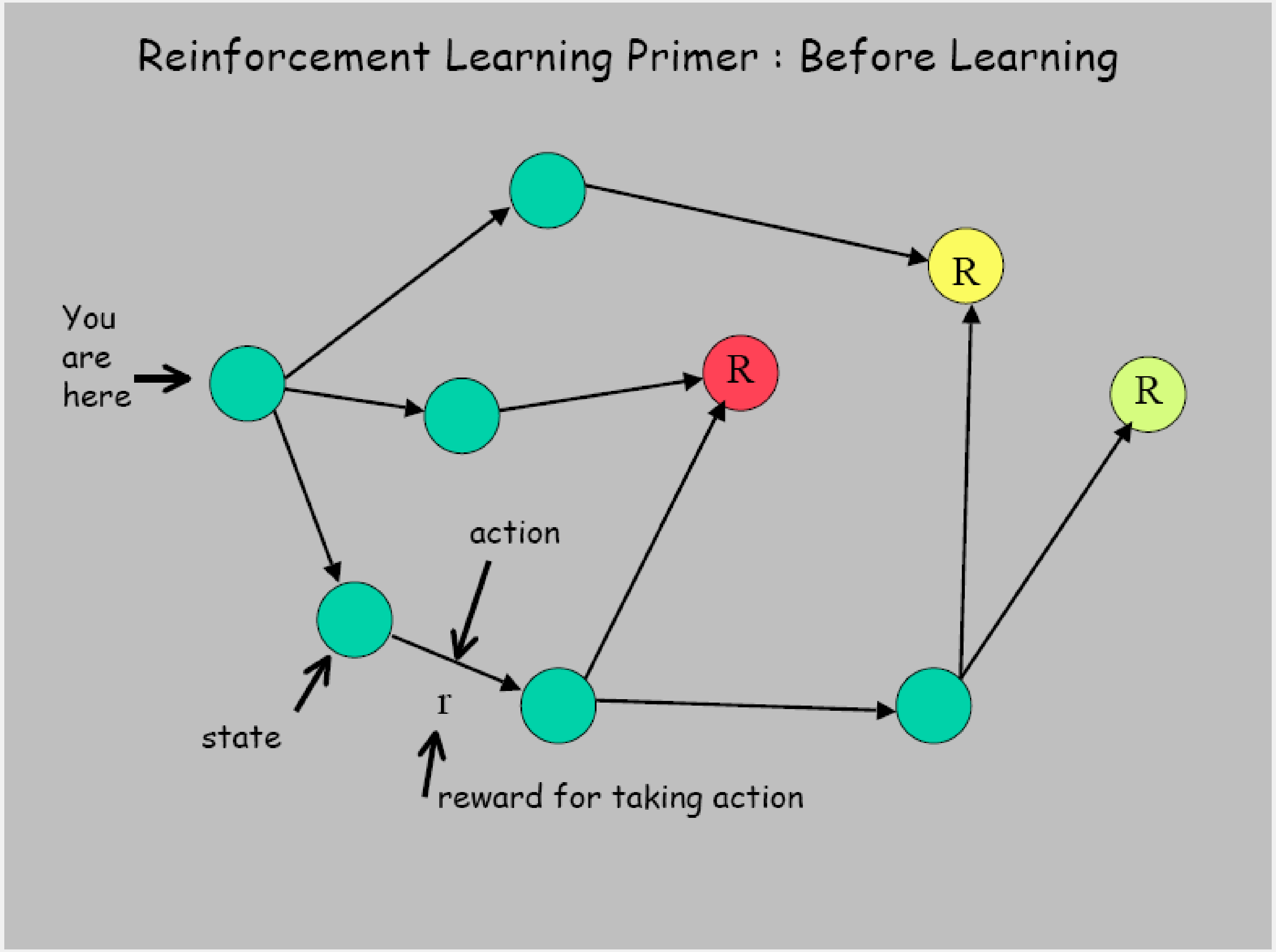
- State
- Action
- Reward

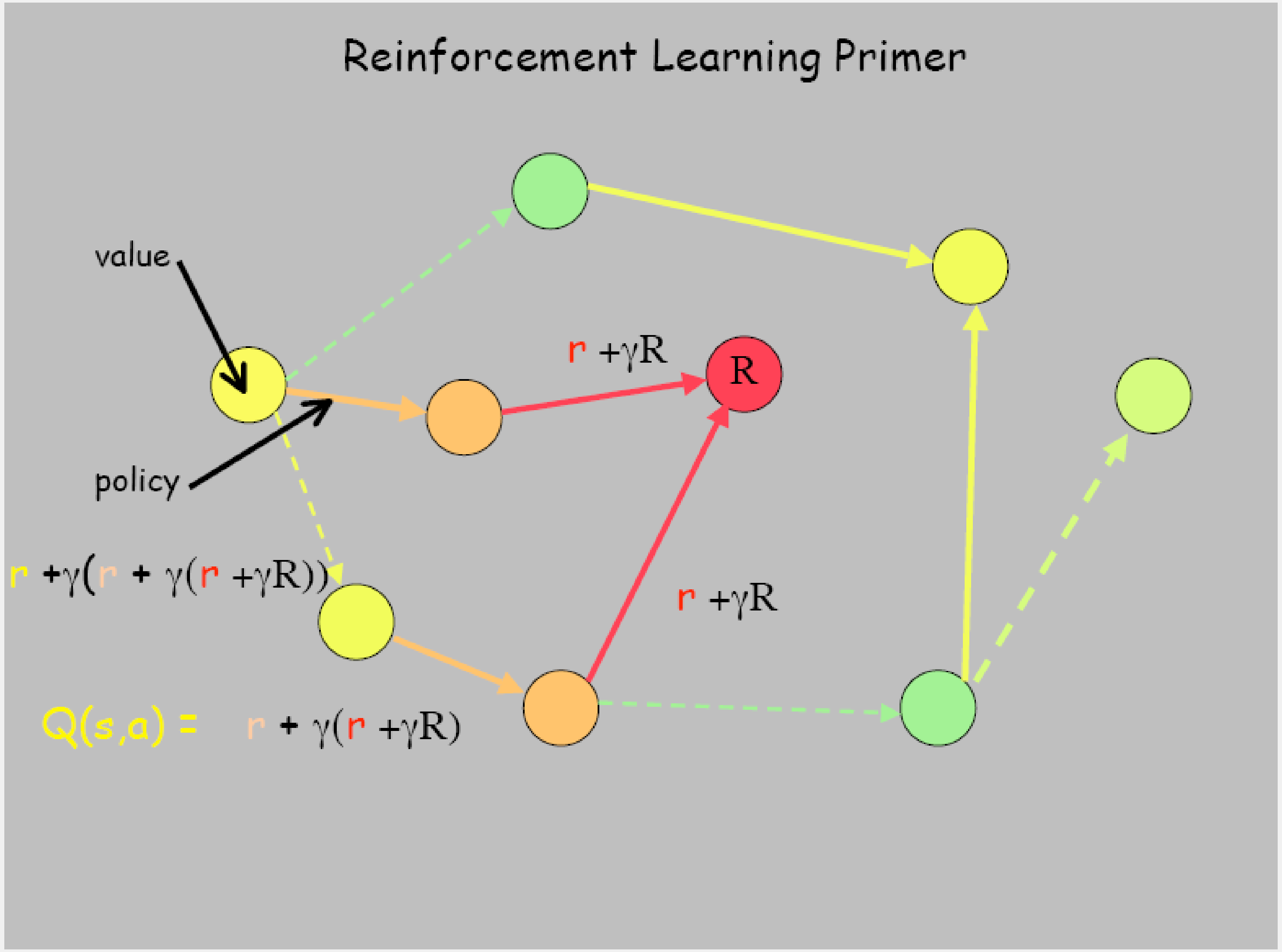
Fundamental Framework for RL

- Agent
- Env



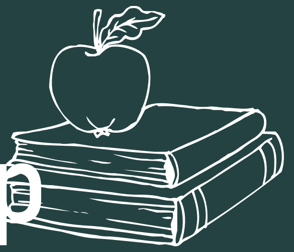
- State
- Action
- Reward
- Policy


Reinforcement Learning Primer : Before Learning
You are here
R
R
R
action
state
r
reward for taking action



Reinforcement Learning Primer
value
policy
r +γR
R
r +γ(r + γ(r +γR))
r +γR
Q(s,a) = r + γ(r +γR)

RL's position in Machine Learning Map

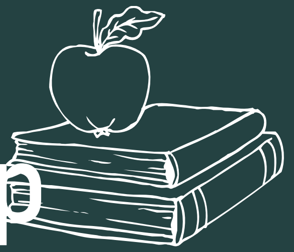


- Heuristic algorithm
- Statistical learning
- Deep learning
- Reinforcement learning

RL's position in Machine Learning Map

- Heuristic algorithm
- Statistical learning
- Deep learning
- ~~Reinforcement learning~~

RL’s position in Machine Learning Map



Classification via Algo

- Heuristic algorithm
- Statistical learning
- Deep learning

Classification via problem

- Supervised learning
- Unsupervised learning
- Reinforcement learning

RL's position in Machine Learning Map

Classification via Algo

- Heuristic algorithm
- Statistical learning
- Deep learning

Classification via problem

- Supervised learning
- Unsupervised learning
- Reinforcement learning

RL's position in Machine Learning Map

Features by

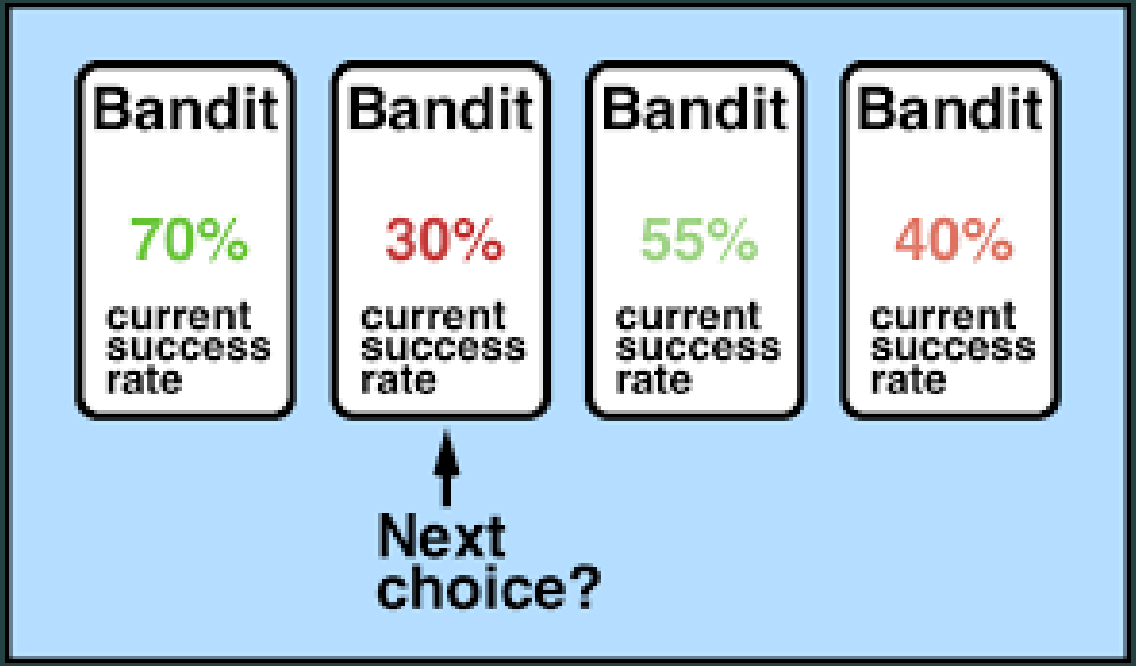
- ➢ Supervised learning: learning from labels
- ➢ Unsupervised learning: find hidden structures
- ➢ Reinforcement learning: learning from environments
  - Rewards are correlated time series, not i.i.d. samples
  - no supervisor, only a delayed reward signal
  - Agent is not told which actions to take, discover the most-rewarded actions by trying them.

Features of Reinforcement Learning

- Delayed reward
- Time matters (sequential data, non i.i.d data)
- Agent's actions affect the subsequent data it receives (agent's action changes the environment)
- Trial-and-error exploration

An Example for Introduction

Multi-armed bandit problem



Exploration-exploitation dilemma

Tasks in Machine Learning

- **Unsupervised Learning**
  - Clustering
  - Dimension Reduction
- **Supervised Learning**
  - Classification
  - Prediction
- **Reinforcement Learning**
  - Learn from environment
  - Learn a "policy"

[1] Dudik, M., Langford, J., Li, L. Doubly robust policy evaluation and learning. In Proceedings of 28th International Conference on Machine Learning. 2011.

[2] Bareinboim, E., Forney, A., Pearl, J. Bandits with Unobserved Confounders: A Causal Approach. In Proceedings of the 28th Annual Conference on Neural Information Processing Systems, 2015.

[3] Zhang, J., Bareinboim, E. Designing Optimal Dynamic Treatment Regimes: A Causal Reinforcement Learning Approach. In Proceedings of the 37th International Conference on Machine Learning. 2020.

Part II Causal RL

Reinforcement learning environments with causal structures

Why Causal + RL?

- Critical concerns in ML:
  - Overfitting
  - bias-variance trade-off
  - Robustness
- Key to causal inference:
  - confounders bias
  - Causal discovery: structural learning
- Bandits with Unobserved Confounders:
- A Causal Approach

- Reinforcement Learning
  - Real-world challenges
  - Promising applications
  - A step further into AGI

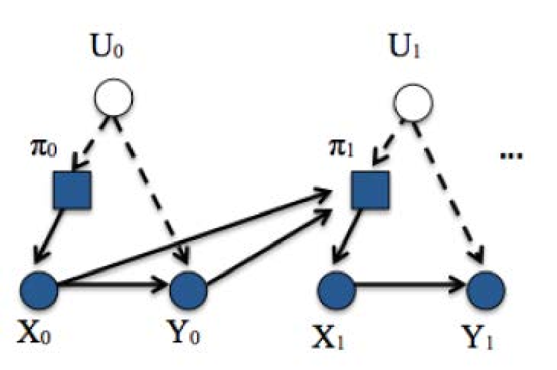
See bandits from a Causal Perspective: Unobserved Confounders

Definition 3.1. (Structural Causal Model) ([Pea00, Ch. 7]) A structural causal model M is a 4-tuple $\langle U, V, f, P(u) \rangle$ where:

1. U is a set of background variables (also called exogenous), that are determined by factors outside of the model,
2. V is a set $\{V_1, V_2, ..., V_n\}$ of observable variables (also called endogenous), that are determined by variables in the model (i.e., determined by variables in $U \cup V$),
3. F is a set of functions $\{f_1, f_2, ..., f_n\}$ such that each f_i is a mapping from the respective domains of $U_i \cup PA_i$ to V_i, where $U_i \subseteq U$ and $PA_i \subseteq V \setminus V_i$ and the entire set F forms a mapping from U to V. In other words, each f_i in $v_i \leftarrow f_i(pa_i, u_i), i = 1, ..., n$, assigns a value to V_i that depends on the values of the select set of variables ($U_i \cup PA_i$), and
4. $P(u)$ is a probability distribution over the exogenous variables.

Definition 3.2. (K-Armed Bandits with Unobserved Confounders) A K-Armed bandit problem with unobserved confounders is defined as a model M with a reward distribution over $P(u)$ where:

1. $X_t \in \{x_1, ..., x_k\}$ is an observable variable encoding player's arm choice from one of k arms, decided by Nature in the observational case, and $do(X_t = \pi(x_0, y_0, ..., x_{t-1}, y_{t-1}))$, for strategy π in the experimental case (i.e., when the strategy decides the choice),
2. U_t represents the unobserved variable encoding the payout rate of arm x_t as well as the propensity to choose x_t, and
3. $Y_t \in 0, 1$ is a reward (0 for losing, 1 for winning) from choosing arm x_t under unobserved confounder state u_t decided by $y_t = f_y(x_t, u_t)$.



See bandits from a Causal Perspective: Unobserved Confounders

Algorithm 1 Causal Thompson Sampling (TS^C)

procedure $\text{TS}^C(P_{obs}, \text{T})$	
$\quad E(Y_{X=a} \vert X) \leftarrow P_{obs}(y \vert X)$	(seed distribution)
$\quad$ **for** $t = [1, ..., T]$ **do**	
$\quad\quad x \leftarrow intuition(t)$	(get intuition for trial)
$\quad\quad Q_1 \leftarrow E(Y_{X=x'} \vert X = x)$	(estimated payout for counter-intuition)
$\quad\quad Q_2 \leftarrow P(y \vert X = x)$	(estimated payout for intuition)
$\quad\quad w \leftarrow [1, 1]$	(initialize weights)
$\quad\quad bias \leftarrow 1 - \vert Q_1 - Q_2 \vert$	(compute weighting strength)
$\quad\quad$ **if** $Q_1 > Q_2$ **then** $w[x] \leftarrow bias$ **else** $w[x'] \leftarrow bias$	(choose arm to bias)
$\quad\quad a \leftarrow max(\beta(s_{M_1,x}, f_{M_1,x}) \times w[1], \beta(s_{M_2,x}, f_{M_2,x}) \times w[2])$	(choose arm) [6]
$\quad\quad y \leftarrow pull(a)$	(receive reward)
$\quad\quad E(Y_{X=a} \vert X = x) \vert\leftarrow y \vert a, x$	(update)

Recall: Regression Method

- Model assumption:

$$E(Y \mid Z, \boldsymbol{X}) = \alpha_0 + \alpha_Z Z + \boldsymbol{X}^T \alpha_X$$

- Treatment effect:

$$\Delta = E\{E(Y \mid Z=1, \boldsymbol{X}) - E(Y \mid Z=0, \boldsymbol{X})\} = \alpha_Z$$

- Calculus treatment effect by fitting a regression model (OLS) $\widehat{\Delta} = \widehat{\alpha}_Z$
- Binary outcome: logistic regression

$$\widehat{\Delta} = n^{-1} \sum_{i=1}^{n} \left\{ \frac{\exp\left(\widehat{\alpha}_0 + \widehat{\alpha}_Z + \boldsymbol{X}_i^T \widehat{\alpha}_X\right)}{1 + \exp\left(\widehat{\alpha}_0 + \widehat{\alpha}_Z + \boldsymbol{X}_i^T \widehat{\alpha}_X\right)} - \frac{\exp\left(\widehat{\alpha}_0 + \boldsymbol{X}_i^T \widehat{\alpha}_X\right)}{1 + \exp\left(\widehat{\alpha}_0 + \boldsymbol{X}_i^T \widehat{\alpha}_X\right)} \right\}$$

- Adjustment by Regression

Recall: Propensity score Method

- Propensity score: Probability of treatment given covariates

$$e(\boldsymbol{X}) = P(Z=1 \mid \boldsymbol{X}) = E\{I(Z=1) \mid \boldsymbol{X}\} = E(Z \mid \boldsymbol{X})$$

- Assumption: $\boldsymbol{X} \perp Z \mid e(\boldsymbol{X})$

  $(Y_0, Y_1) \perp Z \mid e(\boldsymbol{X})$

- Model for propensity score: $P(Z=1 \mid \boldsymbol{X}) = e(\boldsymbol{X}, \boldsymbol{\beta}) = \dfrac{\exp\left(\beta_0 + \boldsymbol{X}^T \beta_1\right)}{1 + \exp\left(\beta_0 + \boldsymbol{X}^T \beta_1\right)}$

- Treatment Estimation: $\widehat{\Delta}_{IPW,1} = n^{-1} \sum_{i=1}^{n} \dfrac{Z_i Y_i}{e\left(\boldsymbol{X}_i, \widehat{\boldsymbol{\beta}}\right)} - n^{-1} \sum_{i=1}^{n} \dfrac{(1 - Z_i) Y_i}{1 - e\left(\boldsymbol{X}_i, \widehat{\boldsymbol{\beta}}\right)}$

Doubly robust estimator

- Modified estimator:

$$\begin{aligned}\widehat{\Delta}_{DR} =& n^{-1}\sum_{i=1}^{n}\left[\frac{Z_iY_i}{e\left(\boldsymbol{X}_i,\widehat{\boldsymbol{\beta}}\right)} - \frac{\left\{Z_i - e\left(\boldsymbol{X}_i,\widehat{\boldsymbol{\beta}}\right)\right\}}{e\left(\boldsymbol{X}_i,\widehat{\boldsymbol{\beta}}\right)}m_1(\boldsymbol{X}_i,\widehat{\boldsymbol{\alpha}}_1)\right] \\ & - n^{-1}\sum_{i=1}^{n}\left[\frac{(1-Z_i)Y_i}{1-e\left(\boldsymbol{X}_i,\widehat{\boldsymbol{\beta}}\right)} + \frac{\left\{Z_i - e\left(\boldsymbol{X}_i,\widehat{\boldsymbol{\beta}}\right)\right\}}{1-e\left(\boldsymbol{X}_i,\widehat{\boldsymbol{\beta}}\right)}m_0(\boldsymbol{X}_i,\widehat{\boldsymbol{\alpha}}_0)\right] \\ =& \widehat{\mu}_{1,DR} - \widehat{\mu}_{0,DR}\end{aligned}$$

$$\widehat{\mu}_{1,DR}: \; E(Y_1) + E\left[\frac{\{Z - e(\boldsymbol{X},\boldsymbol{\beta})\}}{e(\boldsymbol{X},\boldsymbol{\beta})}\{Y_1 - m_1(\boldsymbol{X},\boldsymbol{\alpha}_1)\}\right]$$

Doubly robust estimator

Scenario 1: *Postulated propensity score model* $e(\boldsymbol{X}, \boldsymbol{\beta})$ is *correct*, but *postulated regression model* $m_1(\boldsymbol{X}, \boldsymbol{\alpha}_1)$ is *not*, i.e.,

- $e(\boldsymbol{X}, \boldsymbol{\beta}) = e(\boldsymbol{X}) = E(Z|\boldsymbol{X})$ ($= E(Z|Y_1, \boldsymbol{X})$ by *no unmeasured confounders*)
- $m_1(\boldsymbol{X}, \boldsymbol{\alpha}_1) \neq E(Y|Z = 1, \boldsymbol{X})$

Doubly robust estimator

Scenario 2: *Postulated regression model* $m_1(\boldsymbol{X}, \boldsymbol{\alpha}_1)$ is *correct*, but *postulated propensity score model* $e(\boldsymbol{X}, \boldsymbol{\beta})$ is *not*

- $e(\boldsymbol{X}, \boldsymbol{\beta}) \neq e(\boldsymbol{X}) = E(Z|\boldsymbol{X})$
- $m_1(\boldsymbol{X}, \boldsymbol{\alpha}_1) = E(Y|Z = 1, \boldsymbol{X})$ ($= E(Y_1|\boldsymbol{X})$ by *no unmeasured confounders*)

Doubly Robust Policy Evaluation and Learning

$\widehat{\Delta}_{DR}$ is consistent estimator if:

- Assumption of Scenario 1 holds

or

- Assumption of Scenario 2 holds

Doubly Robust Policy Evaluation

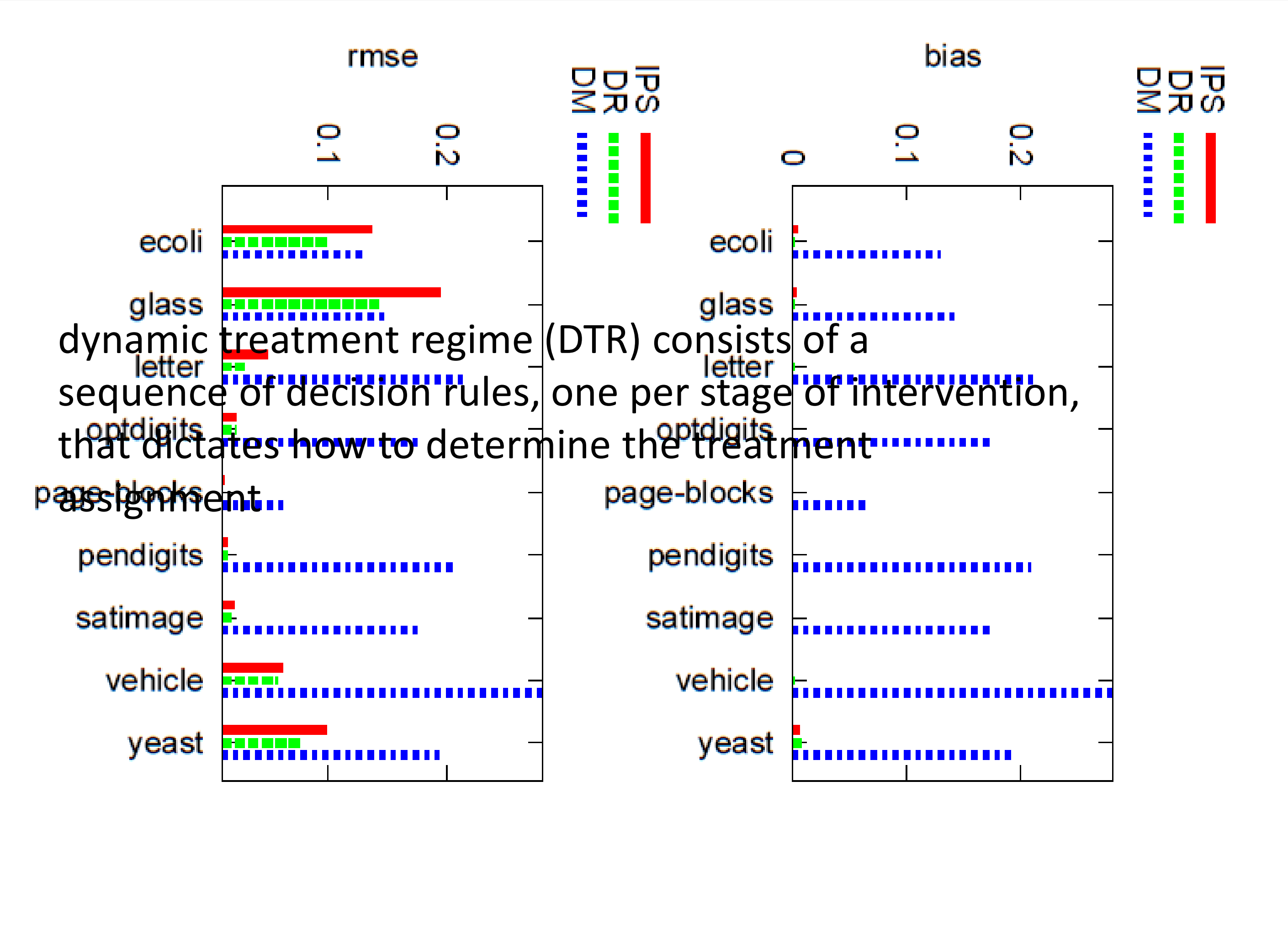
- Apply the doubly robust technique to policy value estimation

expected reward

$$\hat{V}^{\pi}_{\mathrm{DR}} = \frac{1}{|S|} \sum_{(x,h,a,r_a)\in S} \left[\frac{(r_a - \hat{\varrho}_a(x))\mathbf{I}(\pi(x)=a)}{\hat{p}(a \mid x, h)} + \hat{\varrho}_{\pi(x)}(x) \right]$$

the estimate of action probabilities

Doubly Robust Policy Evaluation



dynamic treatment regime (DTR) consists of a sequence of decision rules, one per stage of intervention, that dictates how to determine the treatment assignment

Dynamic Treatment Regime

- Goal of DTR:
  - Determine a sequence of decision rules, one per stage of intervention, that dictates how to determine the treatment assignment

Dynamic Treatment Regime

- Reinforcement learning with causal bond

Theorem 6. *Given $[\![\mathcal{G}, \Pi, Y]\!]$ and causal bounds $\mathcal{C}$, fix a $\delta \in (0, 1)$. W.p. at least $1 - \delta$, it holds for any $T > 1$, the regret of* `OFU-DTR` *is bounded by*

$$R(T, M^*) \leq \Delta(T, \mathcal{C}, \delta) + 2|\boldsymbol{S}|\sqrt{T \log(2|\boldsymbol{S}|T/\delta)},$$

where function $\Delta(T, \mathcal{C}, \delta)$ is defined as

$$\sum_{S_k \in \boldsymbol{S}} \min\left\{|\mathcal{C}_{S_k}|T, 17\sqrt{|\mathcal{D}_{\bar{\boldsymbol{S}}_k \cup \bar{\boldsymbol{X}}_k}|T \log(|\boldsymbol{S}|T/\delta)}\right\}.$$

Algorithm 2 `OFU-DTR`

Input: Signature $[\![\mathcal{G}, \Pi, Y]\!]$, $\delta \in (0, 1)$.
Initialization: Let $\Pi = \texttt{Reduce}(\mathcal{G}, \Pi, Y)$ and let $\mathcal{G} = \texttt{Proj}(\mathcal{G}, \{\boldsymbol{S}, \boldsymbol{X}, Y\})$.
for all episodes $t = 1, 2, \ldots$ **do**
  Define counts $n^t(\boldsymbol{z})$ for any event $\boldsymbol{Z} = \boldsymbol{z}$ prior to episode t as $n^t(\boldsymbol{z}) = \sum_{i=1}^{t-1} I_{\{\boldsymbol{Z}^i = \boldsymbol{z}\}}$.
  For any $S_k \in \boldsymbol{S}$, compute estimates

$$\hat{P}^t_{\bar{\boldsymbol{x}}_k}(s_k | \bar{\boldsymbol{s}}_k \setminus \{s_k\}) = \frac{n^t(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{s}}_k)}{\max\left\{n^t(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{s}}_k \setminus \{s_k\}), 1\right\}}.$$

  Let $\mathcal{P}_t$ denote a set of distributions $P_{\boldsymbol{x}}(\boldsymbol{s})$ such that its factor $P_{\bar{\boldsymbol{x}}_k}(s_k | \bar{\boldsymbol{s}}_k \setminus \{s_k\})$ in Eq. (2) satisfies

$$\left\| P_{\bar{\boldsymbol{x}}_k}(\cdot | \bar{\boldsymbol{s}}_k \setminus \{s_k\}) - \hat{P}^t_{\bar{\boldsymbol{x}}_k}(\cdot | \bar{\boldsymbol{s}}_k \setminus \{s_k\}) \right\|_1 \leq f_{S_k}(t, \delta),$$

  where $f_{S_k}(t, \delta)$ is a function defined as

$$f_{S_k}(t, \delta) = \sqrt{\frac{6|\mathcal{D}_{S_k}| \log(2|\boldsymbol{S}||\mathcal{D}_{(\bar{\boldsymbol{S}}_k \cup \bar{\boldsymbol{X}}_k) \setminus \{S_k\}}|t/\delta)}{\max\left\{n^t(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{s}}_k \setminus \{s_k\}), 1\right\}}}.$$

  Find the optimistic policy $\sigma^t_{\boldsymbol{X}}$ such that

$$\sigma^t_{\boldsymbol{X}} = \arg\max_{\sigma_{\boldsymbol{X}} \in \Pi} \max_{P^t_{\boldsymbol{x}}(\boldsymbol{s}) \in \mathcal{P}_t} V_{\sigma_{\boldsymbol{X}}}(P^t_{\boldsymbol{x}}(\boldsymbol{s})) \qquad (3)$$

  Perform $do(\sigma^t_{\boldsymbol{X}})$ and observe $\boldsymbol{X}^t, \boldsymbol{S}^t$.
end for

[4] Khalil, Elias, et al. "Learning combinatorial optimization algorithms over graphs." Advances in Neural Information Processing Systems. 2017.

[5] Zhu, Shengyu, Ignavier Ng, and Zhitang Chen. "Causal discovery with reinforcement learning." arXiv preprint arXiv:1906.04477 (2019).

Part III
RL for Causal Discovery

RL for Combinational Optimization on graph

Algorithm 1 Q-learning for the Greedy Algorithm

Initialize experience replay memory $\mathcal{M}$ to capacity N
for episode $e = 1$ **to** L **do**
  Draw graph G from distribution $\mathbb{D}$
  Initialize the state to empty $S_1 = ()$
  for step $t = 1$ **to** T **do**
    $v_t = \begin{cases} \text{random node } v \in \overline{S}_t, & \text{w.p. } \epsilon \\ \text{argmax}_{v \in \overline{S}_t} \widehat{Q}(h(S_t), v; \Theta), & \text{otherwise} \end{cases}$
    Add v_t to partial solution: $S_{t+1} := (S_t, v_t)$
    if $t \geq n$ **then**
      Add tuple $(S_{t-n}, v_{t-n}, R_{t-n,t}, S_t)$ to $\mathcal{M}$
      Sample random batch from $B \overset{iid.}{\sim} \mathcal{M}$
      Update Θ by SGD over $\boxed{6}$ for B
    end if
  end for
end for
return Θ

Recall: Causal Discovery

- Constraint based methods
  - Markov assumptions
  - Model joint distributions for observed variables
  - Directed Acyclic Graph→Markov equivalence classes
- SEM: functional causal models
  - Additional assumptions
  - Distinguish DAGs in same Markov equivalence class
    - Linear non-Gaussian acyclic model(LiNGAM)
    - Nonlinear Additive Noise Model(ANM)
    - Post-nonlinear causal model(PNL)
- Score based methods
  - Evaluate the DAG and observed dataset
  - Bayesian Information Criterion (BIC) or Minimum Description Length(MDL) etc…

Recall: Causal Discovery

- Goal: search for the DAG with the best scoring.

$$\min_{\mathcal{G}} \ \mathcal{S}(\mathcal{G}), \ \text{subject to } \mathcal{G} \in \text{DAGs}.$$

- Challenge: large search space
  - 3e6(6-node DAG)
  - 5e26(12-node DAG)
- Alternative method: transfer the problem into continuous space
  - Zheng, Xun, et al. "DAGs with NO TEARS: Continuous optimization for structure learning." *Advances in Neural Information Processing Systems*. 2018.

Recall: Causal Discovery

- ICLR 2020
- Zhu, Shengyu, Ignavier Ng, and Zhitang Chen. "Causal discovery with reinforcement learning." arXiv preprint arXiv:1906.04477 (2019).

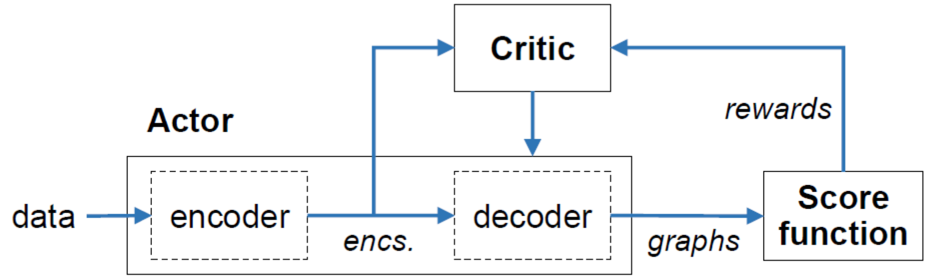


Figure 1: Reinforcement learning for score-based causal discovery.

- Main idea: use Reinforcement Learning (RL) to search for the DAG with the best scoring.

Actor-Critic

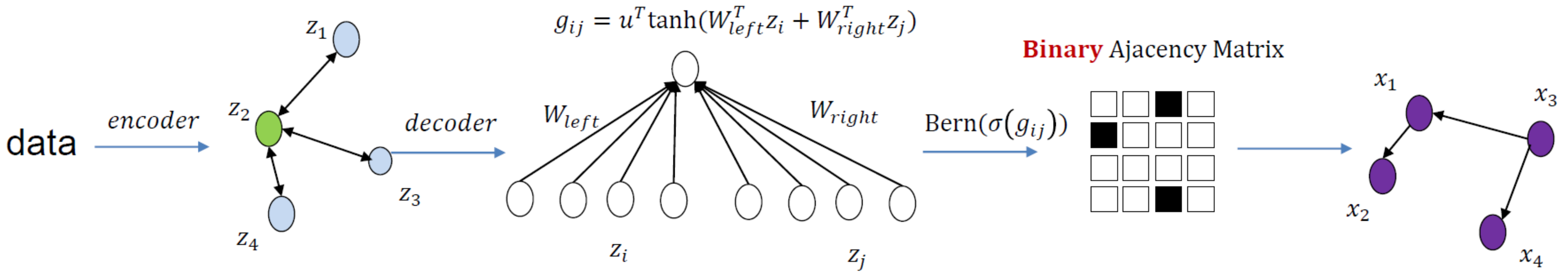
Policy Gradient $$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) Q^{\pi_\theta}(s,a)]$$

Actor-Critic $$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) Q_w(s,a)]$$

$$J(\psi \mid \mathbf{s}) = \mathbb{E}_{A\sim\pi(\cdot\mid\mathbf{s})}\{-[\mathcal{S}(\mathcal{G}) + \lambda_1 \mathbf{I}(\mathcal{G} \notin \mathrm{DAGs}) + \lambda_2 h(A)]\}$$

Reinforcement-learning for Causal Discovery

Encoder-Decoder for generating directed graphs



Reward

$$\text{reward} = -(\text{score function} + \text{DAGness})$$

how a directed graph
fits the observed data

to enforce acyclicity

Reinforcement-learning for Causal Discovery

Algorithm 1 The proposed RL approach to score-based causal discovery

Require: score parameters: $\mathcal{S}_L$, $\mathcal{S}_U$, and $\mathcal{S}_0$; penalty parameters: λ_1, Δ_1, λ_2, Δ_2, and Λ_2; iteration number for parameter update: t_0.

for $t = 1, 2, \ldots$ **do**
    Run actor-critic algorithm, with score adjustment by $\mathcal{S} \leftarrow \mathcal{S}_0(\mathcal{S} - \mathcal{S}_L)/(\mathcal{S}_U - \mathcal{S}_L)$
    if $t \pmod{t_0} = 0$ **then**
        if the maximum reward corresponds to a DAG with score $\mathcal{S}_{\min}$ **then**
            update $\mathcal{S}_U \leftarrow \min(\mathcal{S}_U, \mathcal{S}_{\min})$
        end if
        update $\lambda_1 \leftarrow \min(\lambda_1 + \Delta_1, \mathcal{S}_U)$ and $\lambda_2 \leftarrow \min(\lambda_2 \Delta_2, \Lambda_2)$
        update recorded rewards according to new λ_1 and λ_2
    end if
end for

RL + Causality

- A promising research area
- Various real-world applications
- Ultimate Goal:
  - Train an agent that learns causality from real environment

RL + Causality

TASK 1

Generalized Policy Learning

combining online + offline learning

Learn policy ∏ by systematically combining offline (L_1) and online (L_2) modes of interaction.

TASK 2

When and Where to Intervene?

refining the policy space

Identify subset of L_2 to refine the policy space do(∏(X)) based on topological constraints implied by *M* on *G*.

TASK 3

Counterfactual Decision-Making

changing optimization function based on intentionality, free will, and autonomy

Optimization criterion based on counterfactuals and L_3-based randomization (instead of L_2/do()-counterpart).

TASK 4

Generalizability & Robustness of Causal Claims

transportability & structural invariances

Generalize policy based on structural invariances shared across training (SCM *M*) and deployment environments (*M**).

TASK 5

Learning Causal Models

discovering the causal structure with observation and experiments

Learn the causal graph *G* (of *M*) by systematically combining observations (L_1) and experimentation (L_2).

TASK 6

Causal Imitation Learning

policy learning with unobserved rewards

Construct L_2-policy based on partially observable L_1-data coming from an expert with unknown reward function.

Thanks!