# Bridging Distributional and Risk-Sensitive Reinforcement Learning:

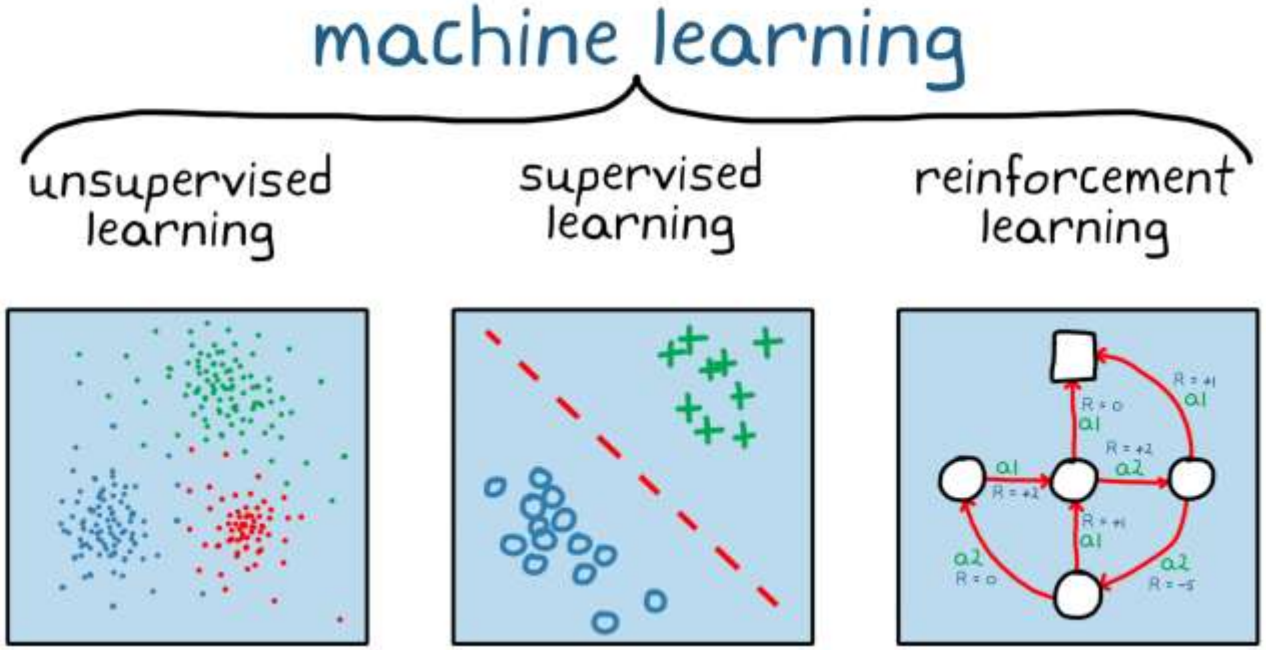
## Balancing Statistical, Computational, and Risk Considerations

**Hao Liang**

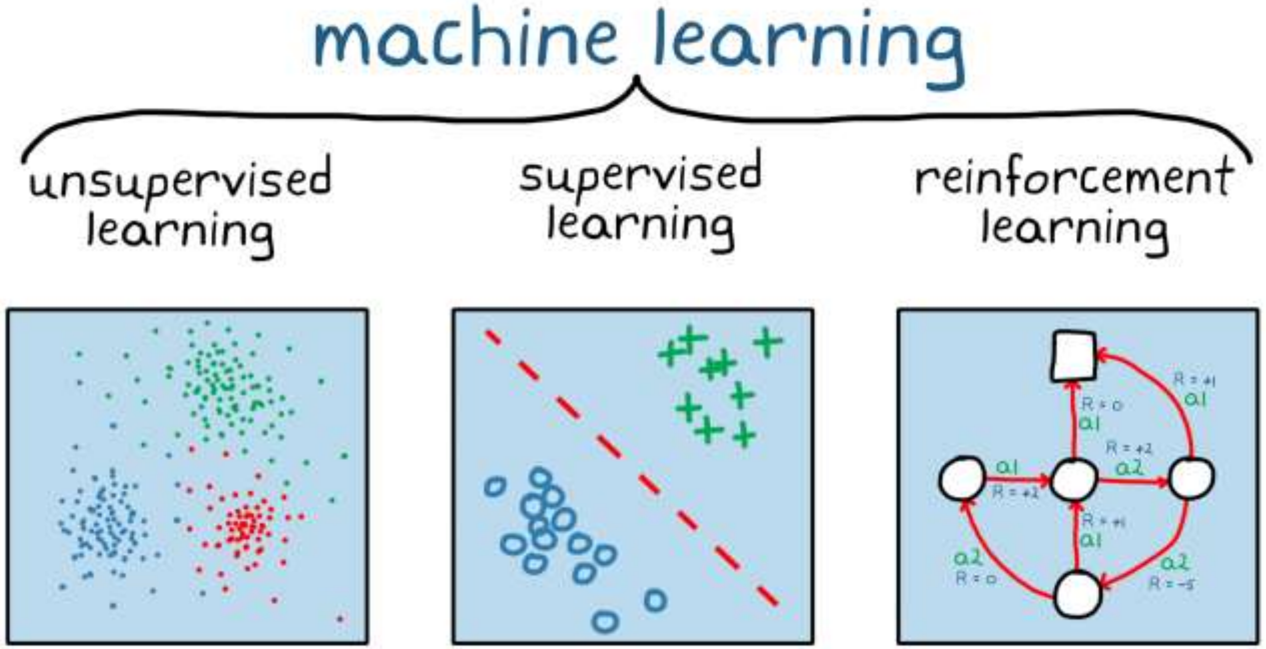
The Chinese University of Hong Kong, Shenzhen

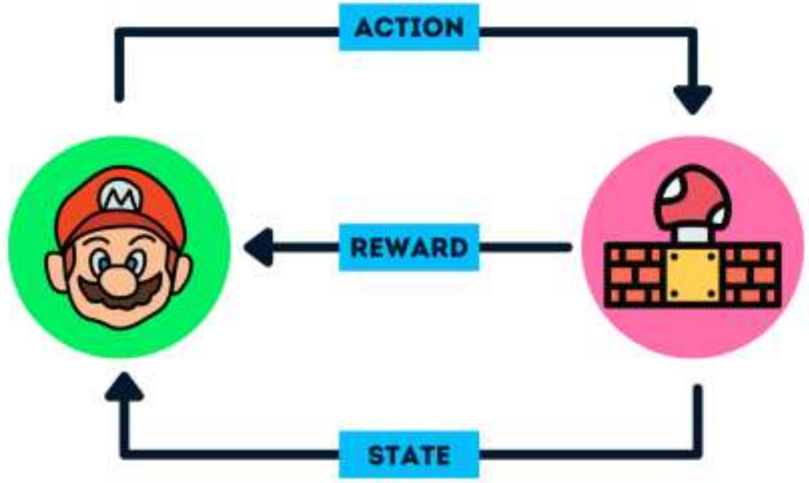
- Hao Liang, and Zhi-Quan Luo. "**Bridging distributional and risk-sensitive reinforcement learning with provable regret bounds**." arXiv preprint arXiv:2210.14051v3 (2022). Under review at *Journal of Machine Learning Research*.
- Hao Liang, and Zhi-Quan Luo. "**A distribution optimization framework for confidence bounds of risk measures**." *International Conference on Machine Learning*. PMLR, 2023.

# What Is Reinforcement Learning (RL)?



# What Is Reinforcement Learning (RL)?



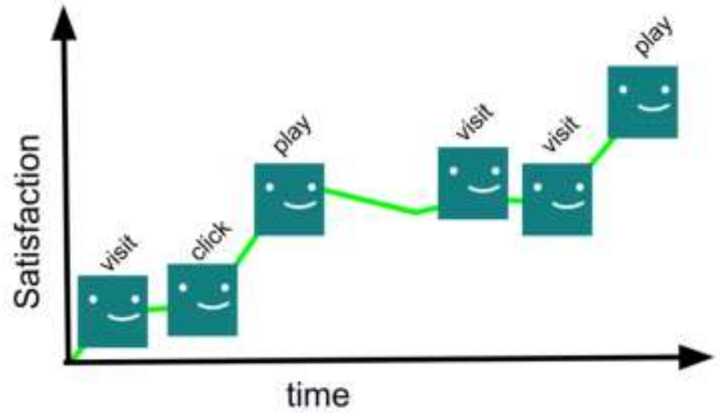

- Sequential decision-making under uncertainty
- The goal is to maximize expected cumulative rewards

$$E\left[\sum_t \text{Reward}_t\right]$$

# Standard RL is Risk-neutral

**Online recommendation**

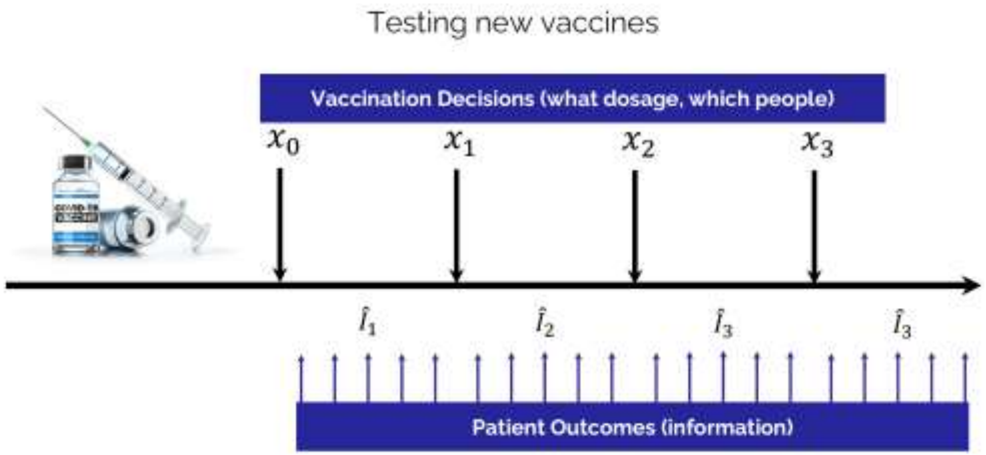

Maximize average “satisfaction”

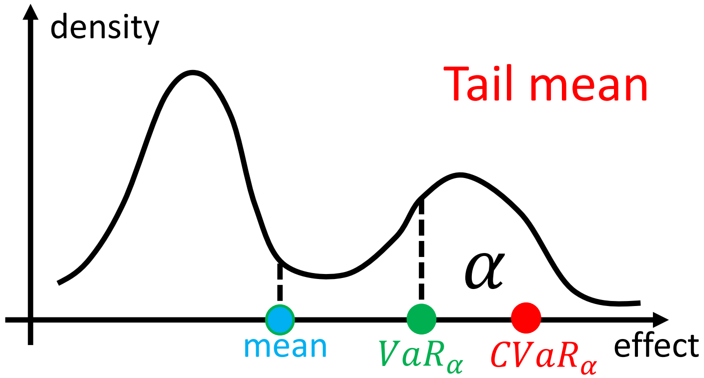



Standard RL aims to maximize the expected return

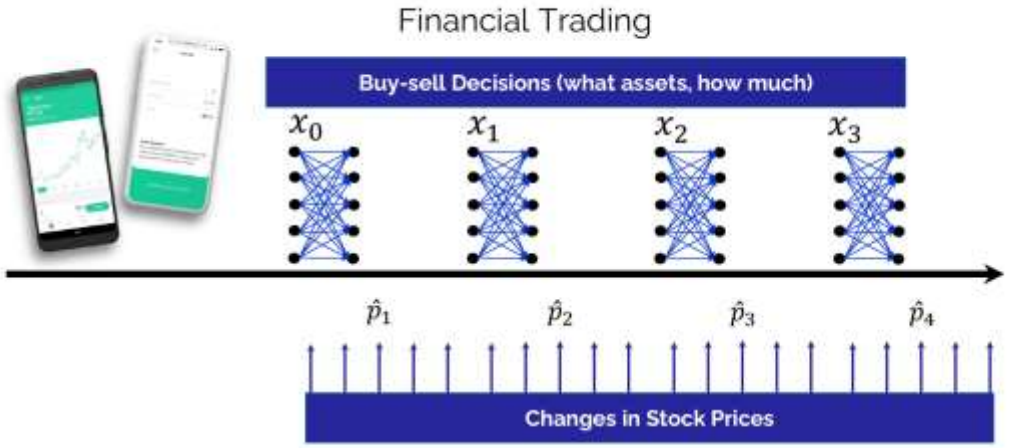
# Risk-sensitive RL (RSRL)

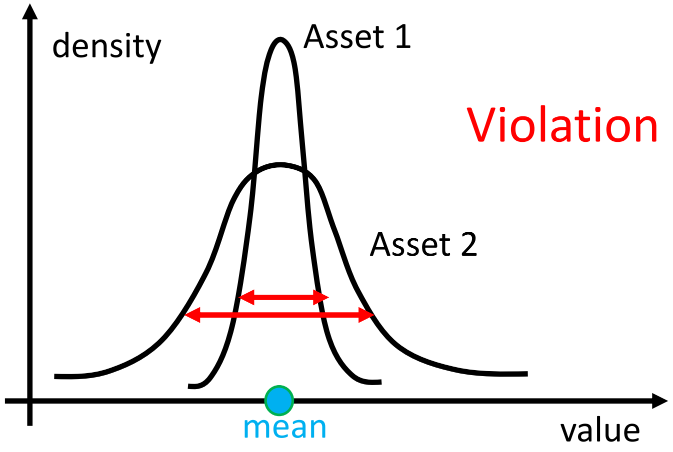
## Healthcare/Clinical trial



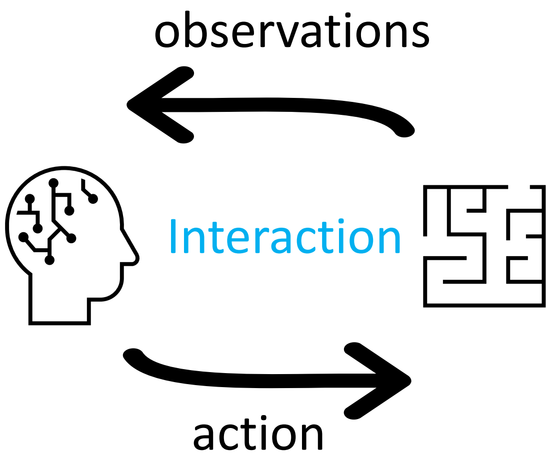

## Finance




RSRL captures other distributional characteristics of return

# Towards Sample-efficient RSRL



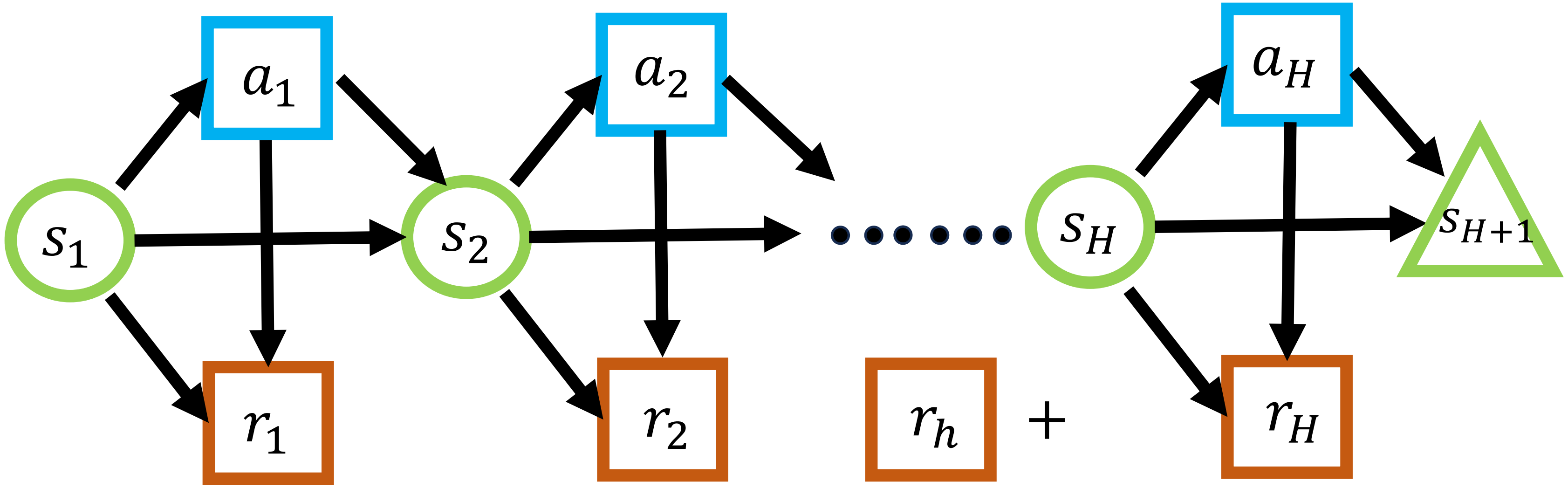

Sample efficiency is critical in RSRL!

- Healthcare monitoring systems
- Financial trading
- Online advertising

**Question: How to attain sample-efficient RSRL?**

**Principle: Distributional Perspective**

# Markov Decision Process (MDP)

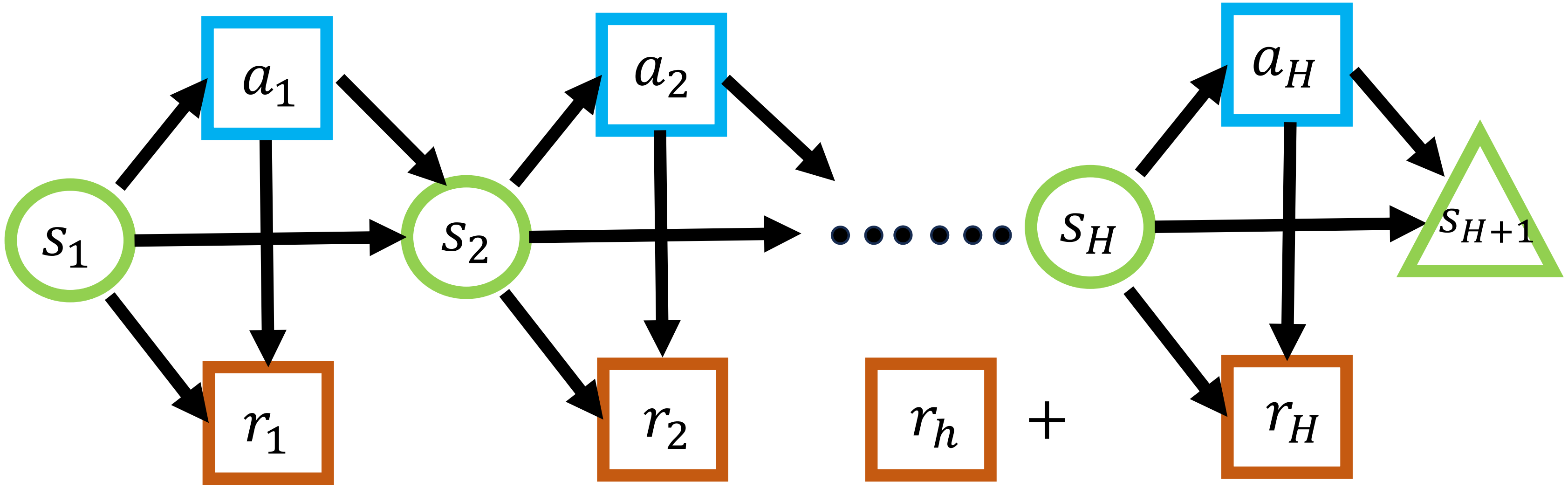



Tabular MDP $\boldsymbol{M} = (S, A, P, r, H)$

- Finite state space $S$, action space $A$
- Transition kernel $P_h(s, a)$

  $$s' \sim P_h(s, a)$$
- Reward function $r_h(s, a)$
- Horizon $H$

# Markov Decision Process (MDP)




Tabular MDP $\boldsymbol{M} = (S, A, P, r, H)$

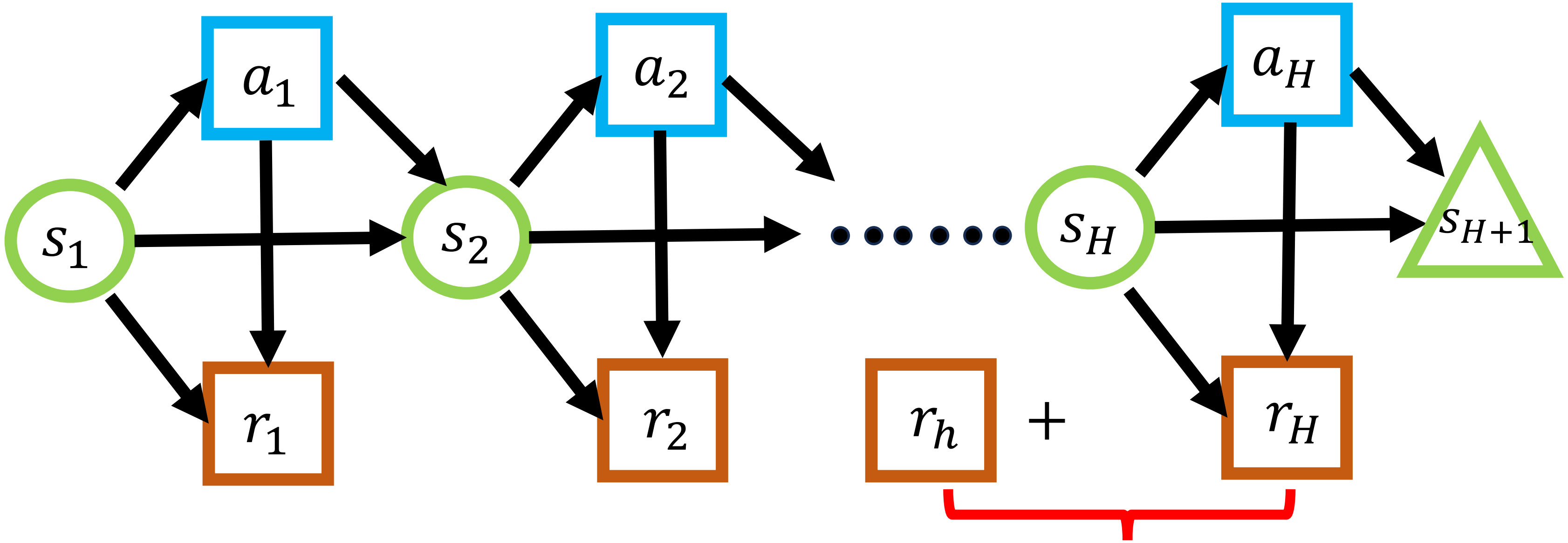
- Finite state space $S$, action space $A$
- Transition kernel $P_h(s, a)$

  $s' \sim P_h(s, a)$
- Reward function $r_h(s, a)$
- Horizon $H$

- Policy $\pi = (\pi_h)_{h \in [H]}$

  $\pi_h : S \to A \in \Pi$

# Markov Decision Process (MDP)




$Z_h^\pi$ **Random Variable**

Tabular MDP $\boldsymbol{M} = (S, A, P, r, H)$

- Finite state space $S$, action space $A$
- Transition kernel $P_h(s, a)$

  $s' \sim P_h(s, a)$
- Reward function $r_h(s, a)$
- Horizon $H$

- Policy $\pi = (\pi_h)_{h \in [H]}$

  $\Pi \ni \pi_h : S \to A$

Return = cumulative reward

$Z_h^\pi = r_h(s_h, a_h) + \cdots + r_H(s_H, a_H),$

$a_h = \pi_h(s_h), s_{h+1} \sim P_h(s_h, a_h)$

# Risk-neutral MDP vs. Risk-sensitive MDP

**Risk-neutral MDP**

$$\max \quad \mathbb{E}[Z_1^\pi]$$

**Risk-sensitive MDP**

$$\max \quad \boldsymbol{\rho}(Z_1^\pi)$$

Risk measure $\boldsymbol{\rho}$: R.V./distribution $\rightarrow \mathbb{R}$
reflects the risk preference towards the uncertainty

# Risk-neutral MDP vs. Risk-sensitive MDP

**Risk-neutral MDP**

$$\max \quad \mathbf{E}[Z_1^\pi]$$

**Risk-sensitive MDP**

$$\max \quad \rho(Z_1^\pi)$$

MANAGEMENT SCIENCE
Vol. 18, No. 7, March, 1972
*Printed in U.S.A.*

RISK-SENSITIVE MARKOV DECISION PROCESSES*

RONALD A. HOWARD† AND JAMES E. MATHESON‡§

**Entropic risk measure (ERM)** [1]

$$U_\beta(X) := \frac{1}{\beta}\log \mathbf{E}[\exp(\beta X)] = \mathbf{E}[X] + \frac{\beta}{2}\mathbf{V}[X] + O(|\beta|^2)$$

[1] Howard, Ronald A., and James E. Matheson. "**Risk-sensitive Markov decision processes**." *Management science* 18.7 (1972): 356-369.

# Risk-neutral MDP vs. Risk-sensitive MDP

**Risk-neutral MDP**

$$\max \quad \mathbb{E}[Z_1^{\pi}]$$

**Risk-sensitive MDP**

$$\max \quad \boldsymbol{\rho}(Z_1^{\pi})$$

MANAGEMENT SCIENCE
Vol. 18, No. 7, March, 1972
*Printed in U.S.A.*

**RISK-SENSITIVE MARKOV DECISION PROCESSES***

RONALD A. HOWARD† AND JAMES E. MATHESON‡§

**Entropic risk measure (ERM) [1]**

$$\boldsymbol{U}_{\boldsymbol{\beta}}(X) := \frac{1}{\beta}\log \mathbf{E}[\exp(\beta X)] = \mathbf{E}[X] + \frac{\beta}{2}\mathbf{V}[X] + O(|\beta|^2)$$

$\beta$ controls risk preference

- Risk-seeking $\beta > 0$: favoring high uncertainty
- Risk-averse $\beta < 0$: favoring low uncertainty
- Risk-neutral $\beta \to 0$

[1] Howard, Ronald A., and James E. Matheson. "**Risk-sensitive Markov decision processes**." *Management science* 18.7 (1972): 356-369.

# Risk-sensitive MDP: Optimality

- Find the optimal policy $\pi^* = (\pi_1^*, \cdots, \pi_H^*)$

$$\pi^* := \operatorname{argmax}_{(\pi_1, \cdots, \pi_H)} V_1^{\pi_1 \cdots \pi_H}(s_1) := U_\beta(Z_1^{\pi_1 \cdots \pi_H})$$

- A multi-stage maximization problem

$$\pi = (\pi_1, \cdots, \pi_H) \in \Pi^H$$

- Direct search suffers exponential computational complexity [2]

$$|\Pi^H| = |\Pi|^H$$

[2] Bertsekas, Dimitri. **Dynamic programming and optimal control: Volume I.** Vol. 4. Athena scientific, 2012.

# Risk-sensitive MDP: Optimality

- Find the optimal policy

$$\pi^* := \mathrm{argmax}_{(\pi_1,\cdots,\pi_H)} V_1^{\pi_1\cdots\pi_H}(s_1) = \boldsymbol{U_\beta}(Z_1^{\pi_1\cdots\pi_H})$$

- A multi-stage maximization problem

$$\pi = (\pi_1, \cdots, \pi_H) \in \Pi^H$$

- Direct search suffers exponential computational complexity

$$|\Pi^H| = |\Pi|^H$$

## Risk-neutral optimality equation

$$Q_h^*(s,a) = r_h(s,a) + \sum P_h(s'|s,a) V_{h+1}^*(s')$$
$$V_h^*(s) = \max_a Q_h^*(s,a), V_{H+1}^*(s) = 0$$

# Risk-sensitive MDP: Optimality

- Find the optimal policy

$$\pi^* := \operatorname{argmax}_{(\pi_1,\cdots,\pi_H)} V_1^{\pi_1\cdots\pi_H}(s_1) = \boldsymbol{U_\beta}(Z_1^{\pi_1\cdots\pi_H})$$

- A multi-stage maximization problem

$$\pi = (\pi_1, \cdots, \pi_H) \in \Pi^H$$

- Direct search suffers exponential computational complexity

$$|\Pi^H| = |\Pi|^H$$

**Risk-neutral optimality equation** → **Optimal substructure**

$$Q_h^*(s,a) = r_h(s,a) + \sum P_h(s'|s,a) V_{h+1}^*(s')$$
$$V_h^*(s) = \max_a Q_h^*(s,a), V_{H+1}^*(s) = 0$$

- Break into multiple single-stage problems
- Recursion of value functions

# Risk-sensitive MDP: Optimality

- Find the optimal policy

$$\pi^* := \operatorname{argmax}_{(\pi_1,\cdots,\pi_H)} V_1^{\pi_1\cdots\pi_H}(s_1) = \boldsymbol{U_\beta}(Z_1^{\pi_1\cdots\pi_H})$$

- A multi-stage maximization problem

$$\pi = (\pi_1, \cdots, \pi_H) \in \Pi^H$$

- Direct search suffers exponential computational complexity

$$|\Pi^H| = |\Pi|^H$$

**Risk-neutral optimality equation** → **Optimal substructure**

$$Q_h^*(s,a) = r_h(s,a) + \sum P_h(s'|s,a) V_{h+1}^*(s')$$
$$V_h^*(s) = \max_a Q_h^*(s,a), V_{H+1}^*(s) = 0$$

- Break into multiple single-stage problems
- Recursion of value functions

**Optimal substructure for risk-sensitive MDP?**

# Risk-sensitive MDP: Optimality

- Find the optimal policy

$$\pi^* := \operatorname{argmax}_{(\pi_1,\cdots,\pi_H)} V_1^{\pi_1\cdots\pi_H}(s_1) = \boldsymbol{U_\beta}(Z_1^{\pi_1\cdots\pi_H})$$

- A multi-stage maximization problem

$$\pi = (\pi_1, \cdots, \pi_H) \in \Pi^H$$

- Direct search suffers exponential computational complexity

$$|\Pi^H| = |\Pi|^H$$

**Risk-neutral optimality equation** $\longrightarrow$ **Optimal substructure**

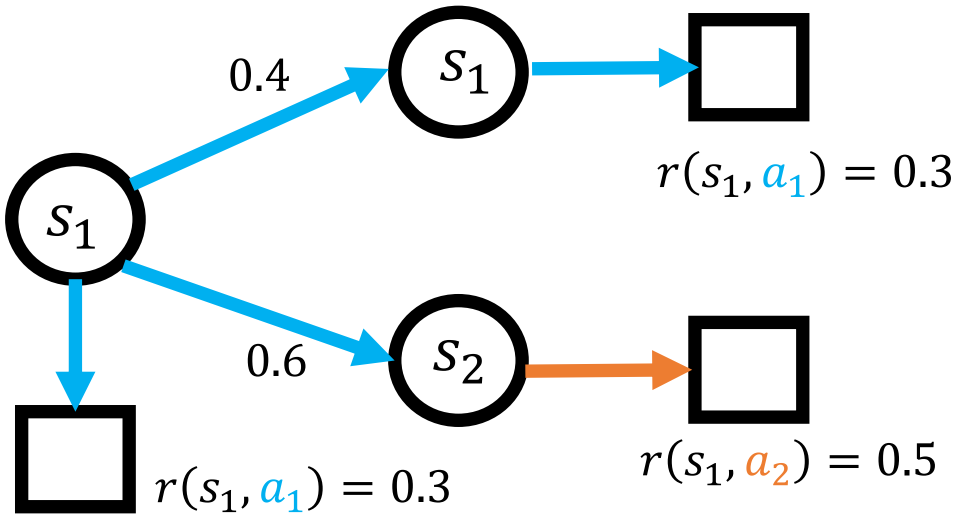
$$Q_h^*(s,a) = r_h(s,a) + \sum P_h(s'|s,a) V_{h+1}^*(s')$$
$$V_h^*(s) = \max_a Q_h^*(s,a), V_{H+1}^*(s) = 0$$

- Break into multiple single-stage problems
- Recursion of value functions

**Optimal substructure for risk-sensitive MDP?**

**Yes. Distributional dynamic programing**

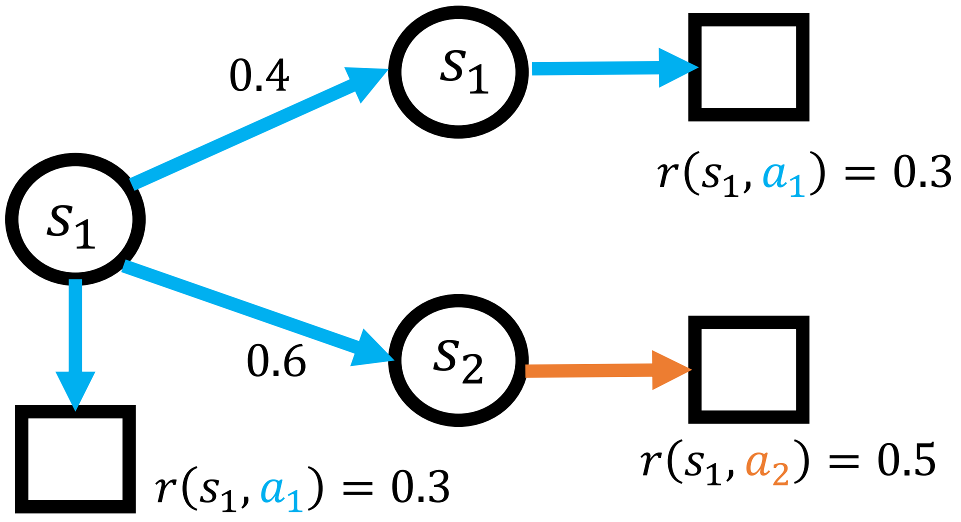
# First Attempt: Distributional Policy Evaluation




Given policy $\pi$
$\pi_1(s_1) = a_1,$
$\pi_2(s_1) = a_1, \pi_2(s_2) = a_2$

**Task**
Determine the **r.v.** $Z_1^{\pi}(s_1)$

# First Attempt: Distributional Policy Evaluation



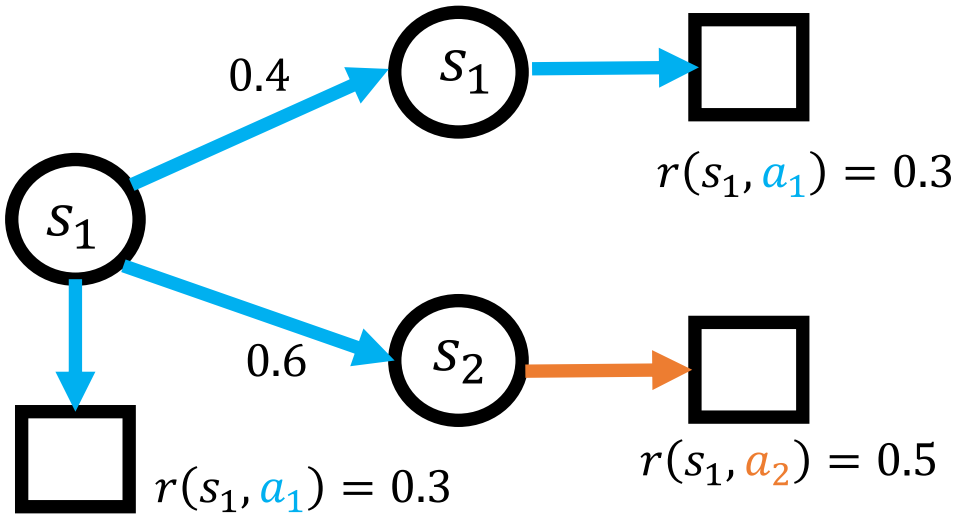

Given policy $\pi$
$\pi_1(s_1) = a_1,$
$\pi_2(s_1) = a_1, \pi_2(s_2) = a_2$

**Task**
Determine the **r.v.** $Z_1^\pi(s_1)$

## Recursion of random variables

Denote by $Y_h(s) \coloneqq Z_h(s, \pi_h(s))$

# First Attempt: Distributional Policy Evaluation




**Given policy $\pi$**

$\pi_1(s_1) = a_1,$

$\pi_2(s_1) = a_1, \pi_2(s_2) = a_2$

**Task**

Determine the **r.v.** $Z_1^\pi(s_1)$

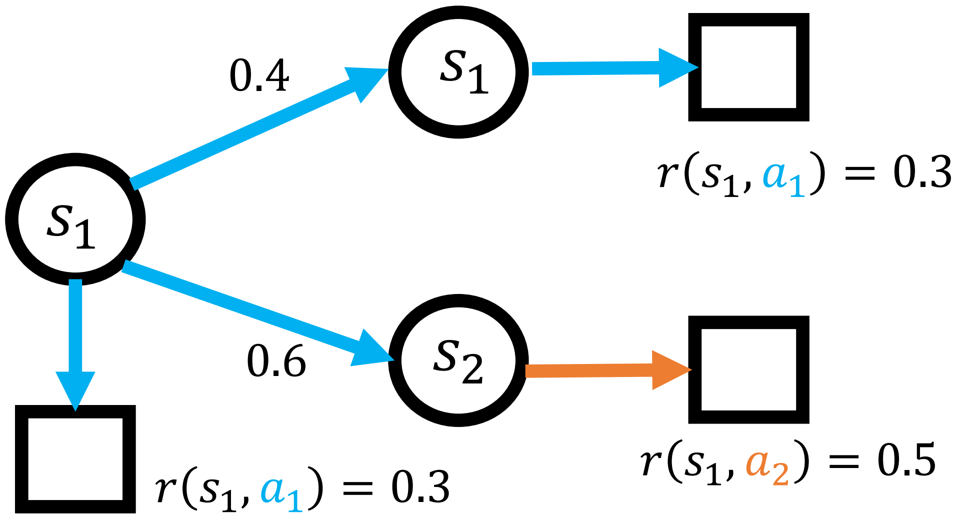
## Recursion of random variables

$$Y_2(s_1) = 0.3$$
$$Y_2(s_2) = 0.5$$

Deterministic final reward

$$S' | s_1, a_1 \sim \begin{cases} s_1, w.p.\, 0.4 \\ s_2, w.p.\, 0.6 \end{cases}$$

# First Attempt: Distributional Policy Evaluation




Given policy $\pi$

$\pi_1(s_1) = a_1,$

$\pi_2(s_1) = a_1, \pi_2(s_2) = a_2$

**Task**

Determine the **r.v.** $Z_1^\pi(s_1)$

## Recursion of random variables

$Y_2(s_1) = 0.3$

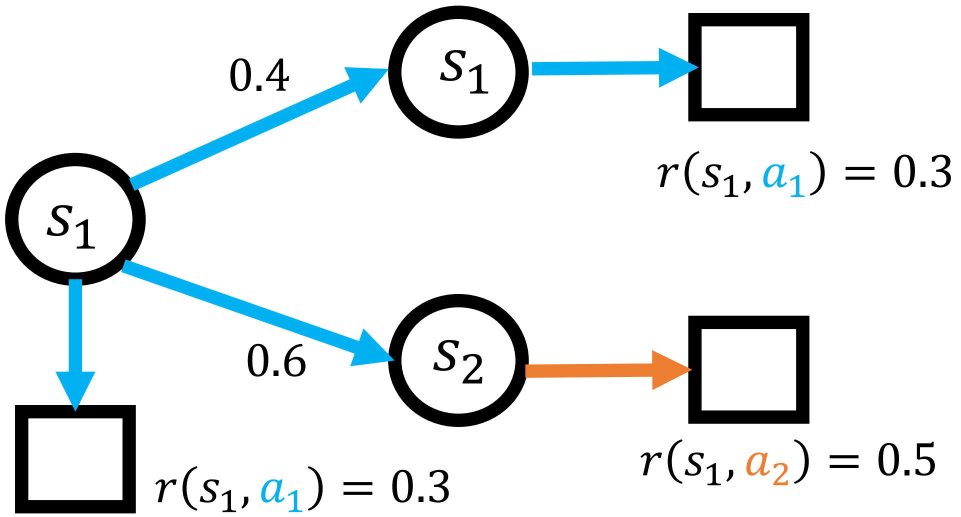
$Y_2(s_2) = 0.5$

$$\longrightarrow \quad Y_2(S')|s_1, a_1 \sim \begin{cases} 0.3, w.p. 0.4 \\ 0.5, w.p. 0.6 \end{cases}$$

Random future return

$$S'|s_1, a_1 \sim \begin{cases} s_1, w.p. 0.4 \\ s_2, w.p. 0.6 \end{cases}$$

# First Attempt: Distributional Policy Evaluation




Given policy $\pi$

$\pi_1(s_1) = a_1,$

$\pi_2(s_1) = a_1, \pi_2(s_2) = a_2$

**Task**

Determine the **r.v.** $Z_1^{\pi}(s_1)$

## Recursion of random variables

$$Y_2(s_1) = 0.3$$
$$Y_2(s_2) = 0.5$$

$$S' | s_1, a_1 \sim \begin{cases} s_1, w.p. \, 0.4 \\ s_2, w.p. \, 0.6 \end{cases}$$

$\longrightarrow$

$$Y_2(S') | s_1, a_1 \sim \begin{cases} 0.3, w.p. \, 0.4 \\ 0.5, w.p. \, 0.6 \end{cases}$$

$$Y_1(s_1) = r_1(s_1, a_1) + Y_2(S') \sim \begin{cases} 0.3 + 0.3, w.p. \, 0.4 \\ 0.3 + 0.5, w.p. \, 0.6 \end{cases}$$

Random initial return

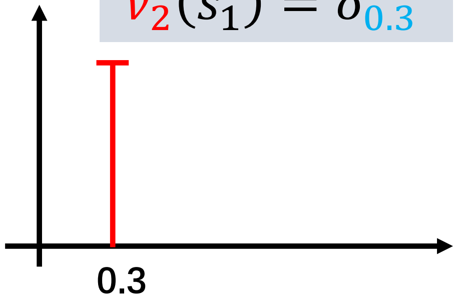
# Recursion of Distributions

## Recursion of r.v.s

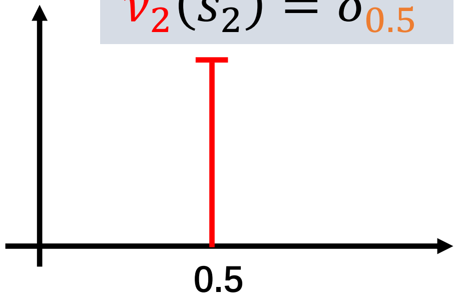
$Y_2(s_1) = 0.3$
$Y_2(s_2) = 0.5$

**Dirac**

$\nu_2(s_1) = \delta_{0.3}$




$\nu_2(s_2) = \delta_{0.5}$




Denote by $\nu_h(s)$ the distribution of $Y_h(s)$

$$Y_h(s) \sim \nu_h(s)$$

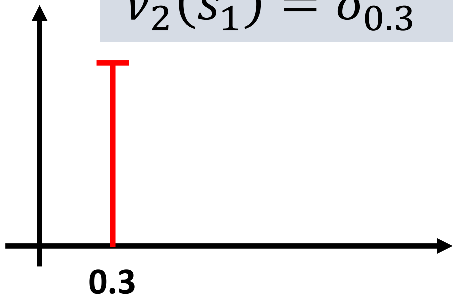
# Recursion of Distributions

## Recursion of r.v.s

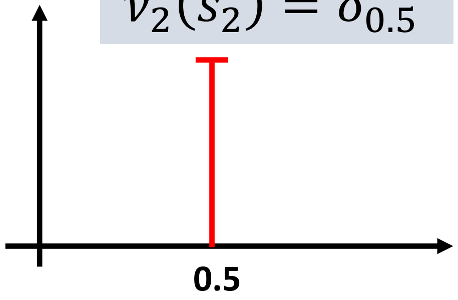
$$Y_2(s_1) = 0.3$$
$$Y_2(s_2) = 0.5$$

Dirac

$$\nu_2(s_1) = \delta_{0.3}$$



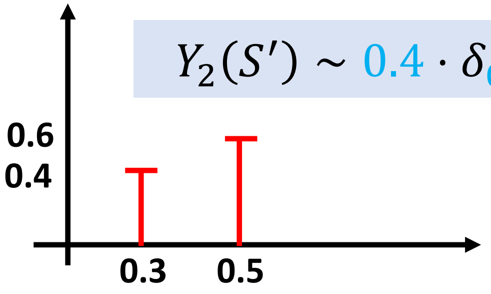

$$\nu_2(s_2) = \delta_{0.5}$$




$$Y_2(S')|s_1, a_1 \sim \begin{cases} 0.3, w.p. 0.4 \\ 0.5, w.p. 0.6 \end{cases}$$

Mixture

$$Y_2(S') \sim 0.4 \cdot \delta_{0.3} + 0.6 \cdot \delta_{0.5}$$



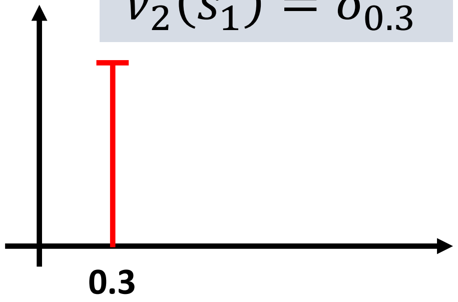
# Recursion of Distributions

## Recursion of r.v.s

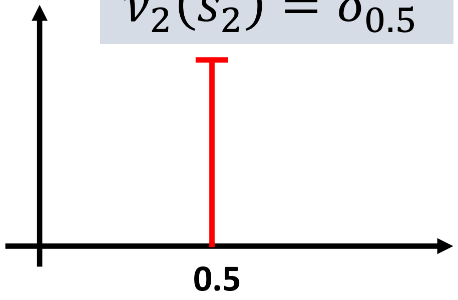
$$Y_2(s_1) = 0.3$$
$$Y_2(s_2) = 0.5$$

**Dirac**

$$\nu_2(s_1) = \delta_{0.3}$$



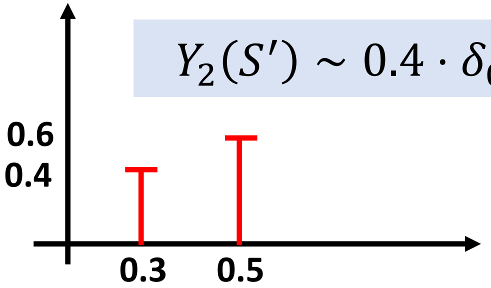

$$\nu_2(s_2) = \delta_{0.5}$$




$$Y_2(S')|s_1, a_1 \sim \begin{cases} 0.3, w.p. \, 0.4 \\ 0.5, w.p. \, 0.6 \end{cases}$$

**Mixture**

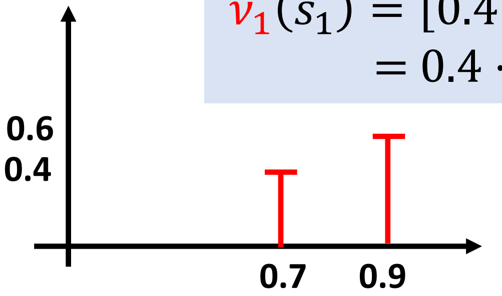
$$Y_2(S') \sim 0.4 \cdot \delta_{0.3} + 0.6 \cdot \delta_{0.5}$$




$$Y_1(s_1) = r_1(s_1, a_1) + Y_2(S')$$
$$\sim \begin{cases} 0.4 + 0.3, w.p. \, 0.4 \\ 0.4 + 0.5, w.p. \, 0.6 \end{cases}$$

**Shift**

$$\nu_1(s_1) = [0.4 \cdot \delta_{0.3} + 0.6 \cdot \delta_{0.5}](\cdot - 0.4)$$
$$= 0.4 \cdot \delta_{0.7} + 0.6 \cdot \delta_{0.9}$$



# Distributional Dynamic Programming: Policy Evaluation

Fact 1: mixture distribution

$$X_i \sim F_i, P(I = i) = p_i \Longrightarrow X_I \sim \sum_i p_i F_i$$

Fact 2: shift

$$X \sim F \Longrightarrow X + c \sim F(\cdot - c)$$

# Distributional Dynamic Programming: Policy Evaluation

Fact 1: mixture distribution

$$X_i \sim F_i, P(I = i) = p_i \Rightarrow X_I \sim \sum_i p_i F_i$$

Fact 2: shift

$$X \sim F \Rightarrow X + c \sim F(\cdot - c)$$

Recursion of R.V.s

$$Z_h(s,a) = r_h(s,a) + Y_{h+1}(S')$$

$$S' \sim P_h(\cdot \mid s,a)$$

$$Y_h(s) = Z_h(s, \pi_h(s))$$

# Distributional Dynamic Programming: Policy Evaluation

Fact 1: mixture distribution

$$X_i \sim F_i, P(I = i) = p_i \Longrightarrow X_I \sim \sum_i p_i F_i$$

Fact 2: shift

$$X \sim F \Longrightarrow X + c \sim F(\cdot - c)$$

Recursion of R.V.s

$$Z_h(s,a) = r_h(s,a) + Y_{h+1}(S')$$
$$S' \sim P_h(\cdot \mid s,a)$$
$$Y_h(s) = Z_h(s, \pi_h(s))$$

**Two facts**

Recursion of distributions

$$\eta_h(s,a) = \sum P_h(s' \mid s,a) \nu_{h+1}(s')(\cdot - r_h(s,a))$$
$$\nu_h(s) = \eta_h(s, \pi_h(s))$$

**mixture** **shift**

# Distributional Dynamic Programming: Policy Evaluation

**Fact 1**: mixture distribution

$$X_i \sim F_i, P(I = i) = p_i \Longrightarrow X_I \sim \sum_i p_i F_i$$

**Fact 2**: shift

$$X \sim F \Longrightarrow X + c \sim F(\cdot - c)$$

Recursion of R.V.s

$$Z_h(s,a) = r_h(s,a) + Y_{h+1}(S')$$
$$S' \sim P_h(\cdot|s,a)$$
$$Y_h(s) = Z_h(s, \pi_h(s))$$

**Two facts**

Recursion of distributions

$$\eta_h(s,a) = \sum P_h(s'|s,a) \nu_{h+1}(s')(\cdot - r_h(s,a))$$
$$\nu_h(s) = \eta_h(s, \pi_h(s))$$

**Distributional Bellman Operator $\mathrm{T}_d$: $P(R)^S \to P(R)^{S \times A}$**

$$\eta_h(s,a) = [\mathrm{T}_d \nu_{h+1}](s,a)$$
$$:= \sum P_h(s'|s,a) \nu_{h+1}(s')(\cdot - r_h(s,a))$$

# Distributional Dynamic Programming: Risk-sensitive Control

**Policy Evaluation**

Given $\pi$, determine $Z_1^\pi$

**Risk-sensitive Control**

$$\max_\pi \quad \boldsymbol{U_\beta}\left(Z_1^\pi\right)$$

# Distributional Dynamic Programming: Risk-sensitive Control

**Risk-sensitive Control**

$$\max_{\pi} \quad \boldsymbol{U_\beta}\,(Z_1^\pi)$$

**Key property 1: Additivity**

$$\boldsymbol{U_\beta}(X + c) = \boldsymbol{U_\beta}(X) + c$$

**Key property 2: Independence**

$$\boldsymbol{U_\beta}(F_2) \leq \boldsymbol{U_\beta}(F_1) \Longrightarrow$$
$$\boldsymbol{U_\beta}\big((1-\theta)F_2 + \theta G\big) \leq \boldsymbol{U_\beta}\big((1-\theta)F_1 + \theta G\big)$$

# Distributional Dynamic Programming: Risk-sensitive Control

**Risk-sensitive Control**

$$\max_{\pi} \quad \boldsymbol{U_\beta}\,(Z_1^\pi)$$

**Key property 1: Additivity**

$$\boldsymbol{U_\beta}(X+c) = \boldsymbol{U_\beta}(X) + c$$

**Key property 2: Independence**

$$\boldsymbol{U_\beta}(F_2) \le \boldsymbol{U_\beta}(F_1) \Longrightarrow$$
$$\boldsymbol{U_\beta}\big((1-\theta)F_2 + \theta G\big) \le \boldsymbol{U_\beta}\big((1-\theta)F_1 + \theta G\big)$$

<u>**Distributional Bellman Optimality Equation**</u>

$$\eta_h^*(s,a) = [\mathbf{T}_{\boldsymbol{d}} \nu_{h+1}^*](s,a)$$
$$\pi_h^*(s) = \mathrm{argmax}_a \boldsymbol{U_\beta}(\eta_h^*(s,a))$$
$$\nu_h^*(s) = \eta_h^*\big(s, \pi_h^*(s)\big)$$

**backward recursion**

# Distributional Dynamic Programming: Risk-sensitive Control

**Risk-sensitive Control**

$$\max_{\pi} \quad \boldsymbol{U}_{\boldsymbol{\beta}}\left(Z_1^{\pi}\right)$$

**Key property 1: Additivity**

$$\boldsymbol{U}_{\boldsymbol{\beta}}(X+c)=\boldsymbol{U}_{\boldsymbol{\beta}}(X)+c$$

**Key property 2: Independence**

$$\boldsymbol{U}_{\boldsymbol{\beta}}(F_2)\leq \boldsymbol{U}_{\boldsymbol{\beta}}(F_1)\Longrightarrow$$
$$\boldsymbol{U}_{\boldsymbol{\beta}}\big((1-\theta)F_2+\theta G\big)\leq \boldsymbol{U}_{\boldsymbol{\beta}}\big((1-\theta)F_1+\theta G\big)$$

**Distributional Bellman Optimality Equation**

$$\eta_h^*(s,a)=[\mathbf{T}_{\boldsymbol{d}}\nu_{h+1}^*](s,a)$$
$$\pi_h^*(s)=\mathrm{argmax}_a \boldsymbol{U}_{\boldsymbol{\beta}}(\eta_h^*(s,a))$$
$$\nu_h^*(s)=\eta_h^*\big(s,\pi_h^*(s)\big)$$

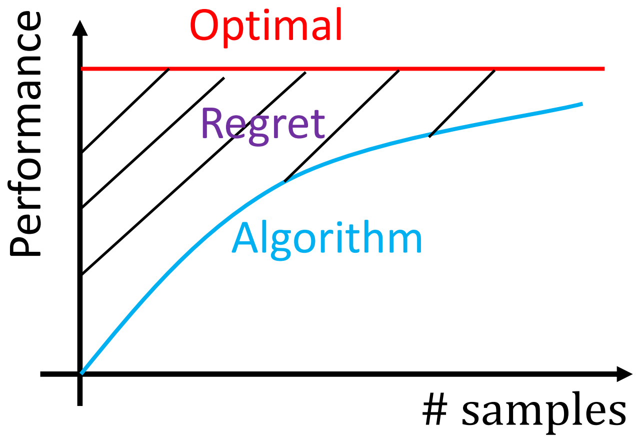
**backward recursion**

**greedy is optimal**

# From DP to RL

- DP requires the knowledge of transition
- RL deals with unknown transition
- Learn from interactions/samples
- Expensive samples

**Fundamental problem of RL**
Statistical/sample efficiency

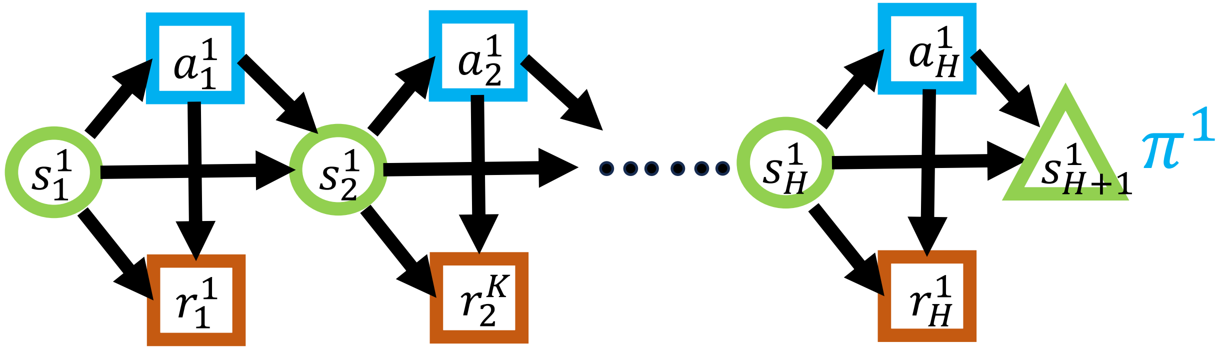



**Low regret**

↓

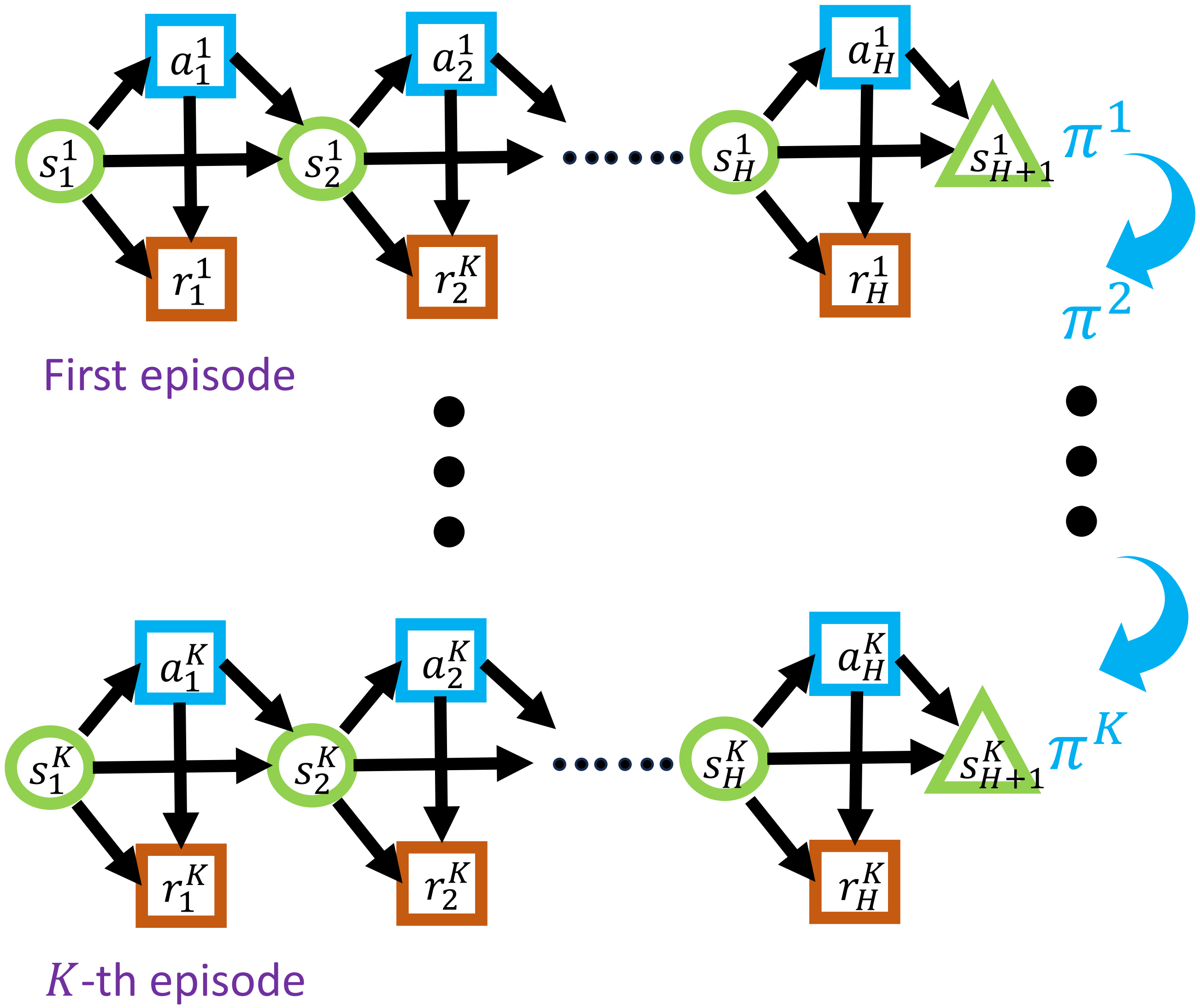
**High sample efficiency**

# Episodic MDP

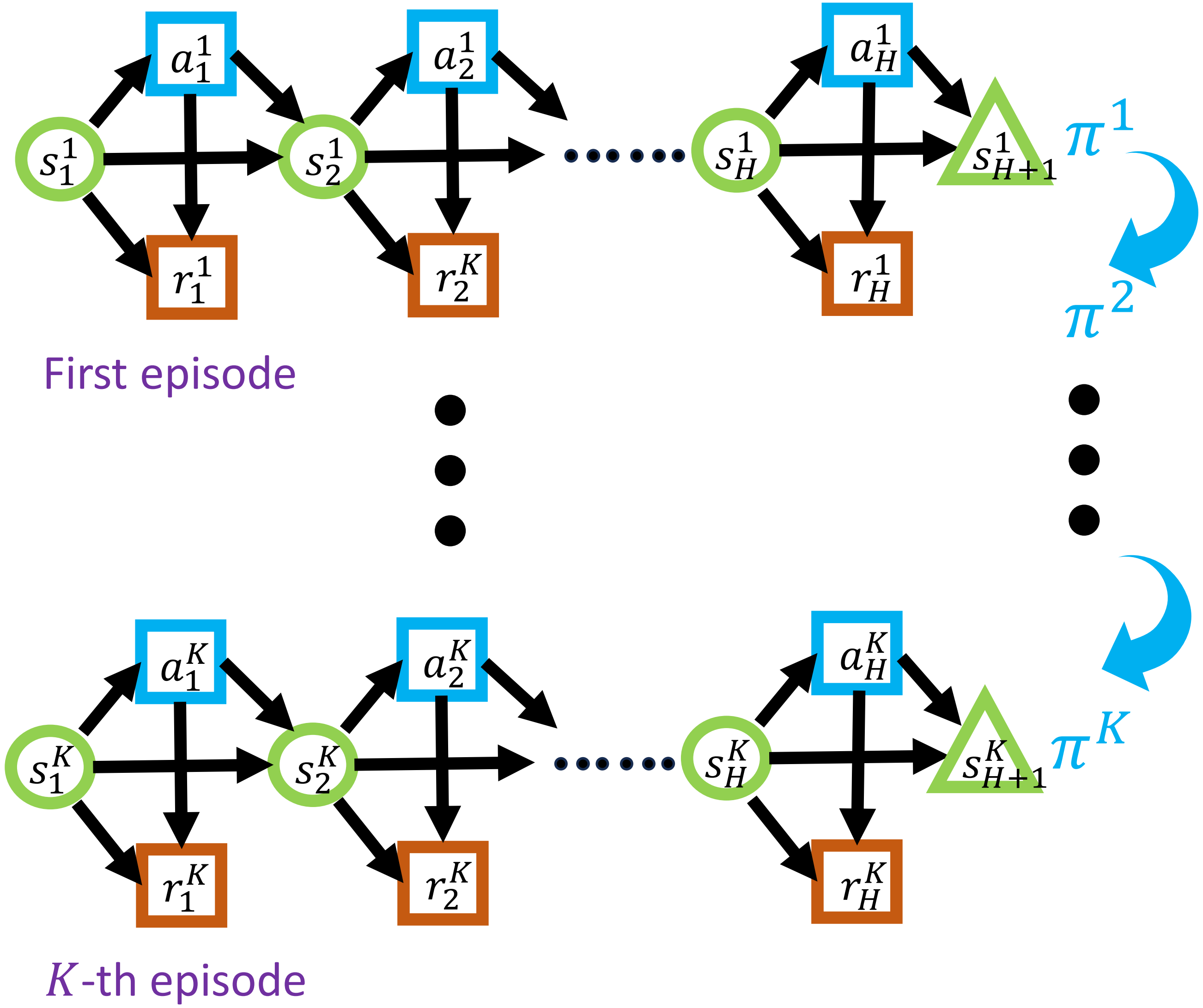


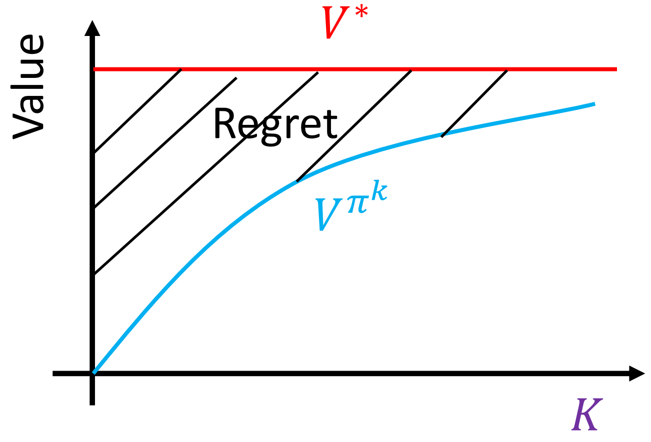

First episode

# Episodic MDP



# Episodic MDP



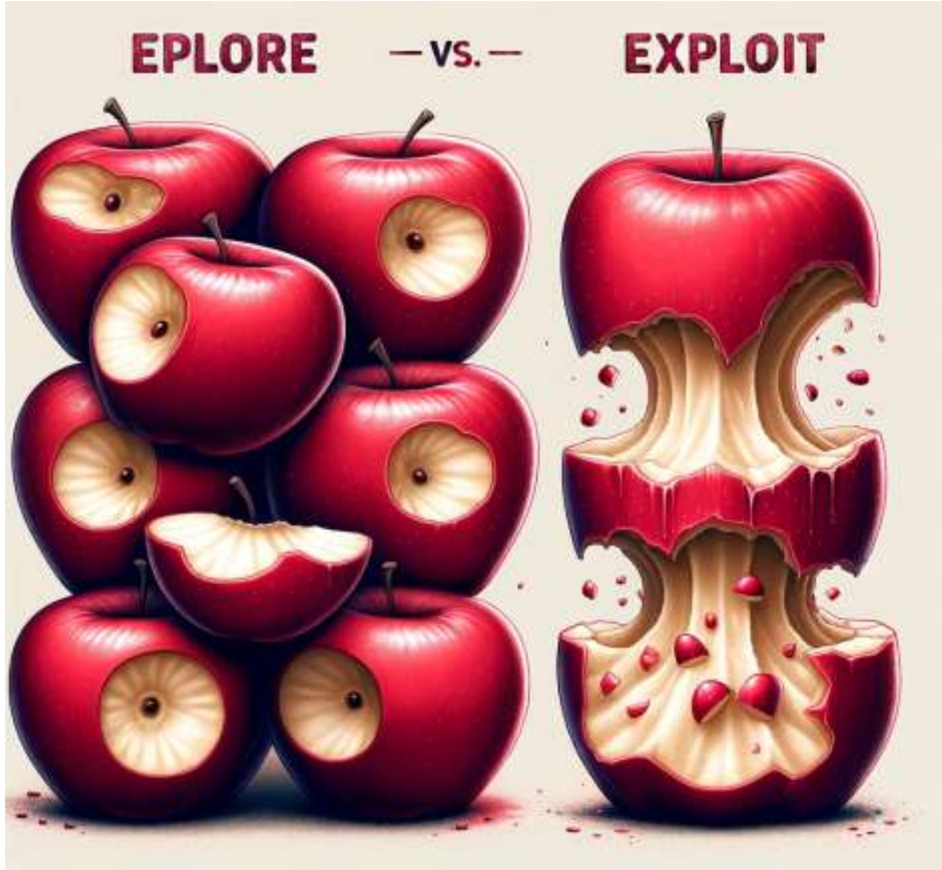

Goal : minimize the regret over $K$ episodes

$$\text{Regret}(\text{alg}, K) := \sum_{k=1}^{K} V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k)$$

# Exploration vs. Exploitation Dilemma

EXPLORATION

Try different actions
to gather info




EXPLOITATION

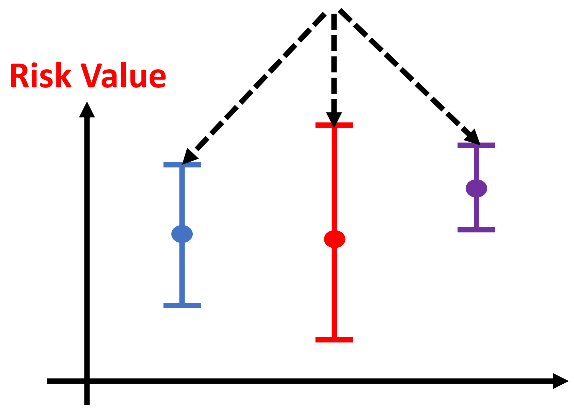
Use the best-known action

**Sample efficiency requires balance between exploration and exploitation!**

# Risk-sensitive Optimism in Face of Uncertainty

- An effective principle for efficient exploration

  Optimism in Face of Uncertainty (OFU)
- Act greedily w.r.t. **Upper Confidence Bound** of **Risk Value**
- Encourages exploring actions with the best possible outcomes




**Tighter UCB**

↓

**Higher sample efficiency**

# Risk-sensitive Optimistic Distribution Iteration (RODI)

**Empirical model**

$$\hat{P}_h^k(s'|s,a) = N_h^k(s,a,s')/\ N_h^k(s,a)$$

**Approximate Distributional Bellman operator**

$$\left[\widehat{\mathbf{T}_d}^k \nu_{h+1}^k\right](s,a) := \sum \hat{P}_h^k(s'|s,a)\nu_{h+1}^k(s')(\cdot - r_h\ (s,a))$$

**Approximate Bellman recursion**

$$\eta_h^k \leftarrow \widehat{\mathbf{T}_d}^k \nu_{h+1}^k$$

**Distributional Optimism Operator**

$$\eta_h^k \leftarrow \mathbf{O}_{c^k}\eta_h^k$$

**Policy in episode $k$**

$$\pi_h^k(s) \leftarrow \operatorname{argmax}_a U_\beta(\eta_h^k(s,a))$$

# Risk-sensitive Optimistic Distribution Iteration (RODI)

**Empirical model**

$$\hat{P}_h^k(s'|s,a) = N_h^k(s,a,s')/\ N_h^k(s,a)$$

**Approximate Distributional Bellman operator**

$$\left[\widehat{\mathbf{T}_d}^k \nu_{h+1}^k\right](s,a) \coloneqq \sum \hat{P}_h^k(s'|s,a)\nu_{h+1}^k(s')(\cdot - r_h\ (s,a))$$

**Approximate Bellman recursion**

$$\eta_h^k \leftarrow \widehat{\mathbf{T}_d}^k \nu_{h+1}^k$$

**Distributional Optimism Operator**

$$\eta_h^k \leftarrow \mathbf{O}_{c^k} \eta_h^k$$

**Policy in episode *k***

$$\pi_h^k(s) \leftarrow \operatorname{argmax}_a U_\beta(\eta_h^k(s,a))$$

**RODI in one line**

$$\eta_h^k \leftarrow \mathbf{O}_{c^k} \widehat{\mathbf{T}_d}^k \nu_{h+1}^k$$

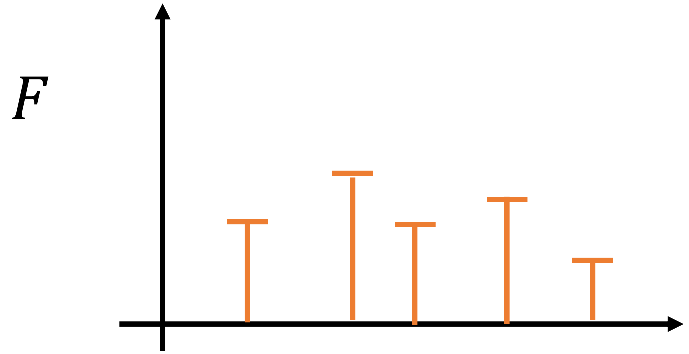
**Optimism**

$$\boldsymbol{U}_{\boldsymbol{\beta}}(\eta_h^k(s,a)) \geq \boldsymbol{U}_{\boldsymbol{\beta}}(\eta_h^*(s,a))$$
$$\forall\ (s,a,k,h)$$

# Distributional Optimism

**Def.** A distribution $F$ is optimistic over $G$ if
$$\boldsymbol{U}_{\boldsymbol{\beta}}(F) \geq \boldsymbol{U}_{\boldsymbol{\beta}}(G)$$
Denote by $F \geq \mathrm{G}$

Given a confidence ball of distributions
$$B(F, c) \coloneqq \{G \,|\, \|G - F\| \leq c\}$$



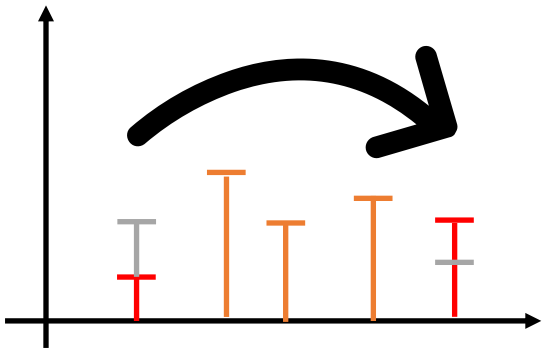
# Distributional Optimism

**Def.** A distribution $F$ is optimistic over $G$ if
$$\boldsymbol{U}_{\boldsymbol{\beta}}(F) \geq \boldsymbol{U}_{\boldsymbol{\beta}}(G)$$
Denote by $F \geq \mathrm{G}$

Given confidence ball of distributions
$$B(F,c) := \{G \,|\, \|G - F\| \leq c\}$$

**Distributional optimism operator**
$$\mathbf{O}_c F \geq G, \forall G \in B(F,c)$$

$\mathbf{O}_c F$



**Intuition** move probability mass from the lower tail to the maximum value

- Keramati, Ramtin, et al. "**Being optimistic to be conservative: Quickly learning a CVaR policy**." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. No. 04. 2020.
- Hao Liang, and Zhi-Quan Luo. "**A distribution optimization framework for confidence bounds of risk measures.**" *International Conference on Machine Learning*. PMLR, 2023.

# Optimism of RODI

**RODI**

$$\eta_h \leftarrow \mathbf{O}_c \widehat{\mathbf{T}}_d \nu_{h+1}$$

**Induction** $\nu_{h+1} \geq \nu_{h+1}^*$

$$\left\| \widehat{\mathbf{T}}_d \nu_{h+1} - \mathbf{T}_d \nu_{h+1} \right\| \leq c_h \Longrightarrow \mathbf{T}_d \nu_{h+1} \in B\left(\widehat{\mathbf{T}}_d \nu_{h+1}, c_h\right)$$

$$\Big\downarrow \ \textbf{Distributional optimism operator}$$

$$\eta_h = \mathbf{O}_c \widehat{\mathbf{T}}_d \nu_{h+1} \geq \mathbf{T}_d \nu_{h+1}$$

$$\Big\downarrow \ \textbf{Induction}$$

$$\eta_h \geq \mathbf{T}_d \nu_{h+1} \geq \mathbf{T}_d \nu_{h+1}^* = \eta_h^* \longrightarrow \nu_h \geq \nu_h^*$$

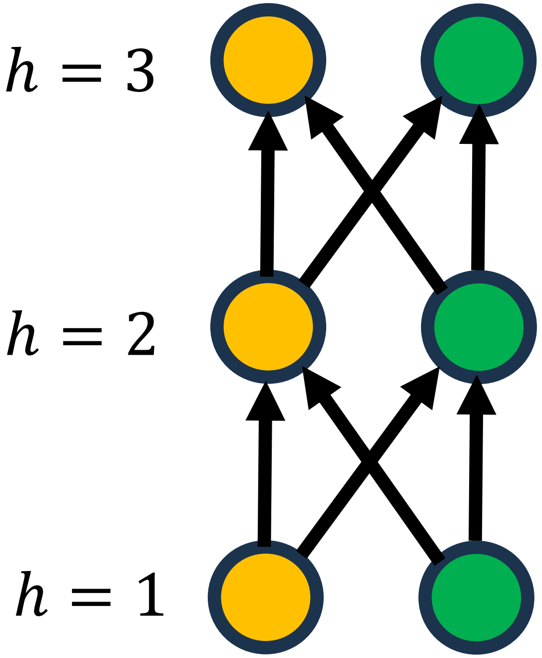
$$\Big\downarrow \ \textbf{Def.}$$

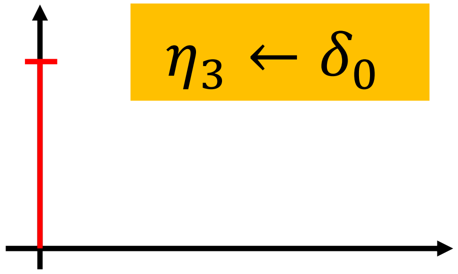
**Optimism of risk value**

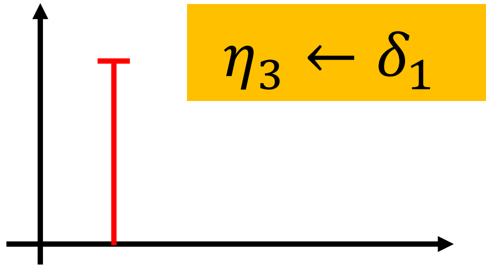
$$\boldsymbol{U}_{\boldsymbol{\beta}}(\eta_h(s,a)) \geq \boldsymbol{U}_{\boldsymbol{\beta}}(\eta_h^*(s,a)) \longrightarrow Q_h(s,a) \geq Q_h^*(s,a)$$

# Computational Inefficiency of RODI

State space = $\{\,,\,\}$
Uniform transition
$r(\,) = 0, r(\,) = 1$



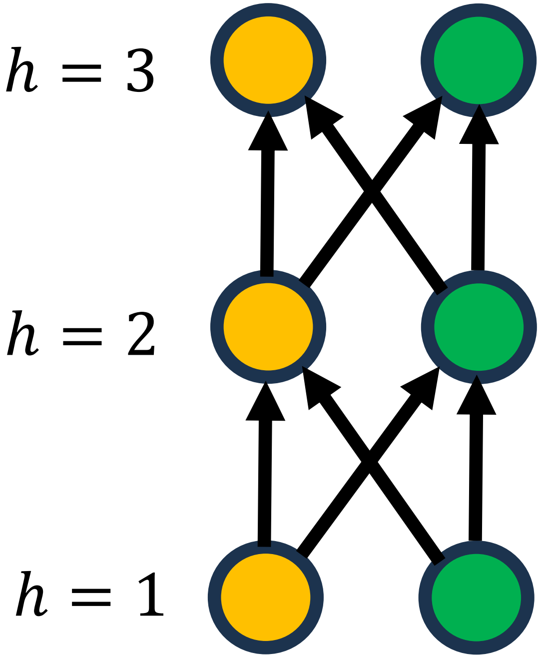



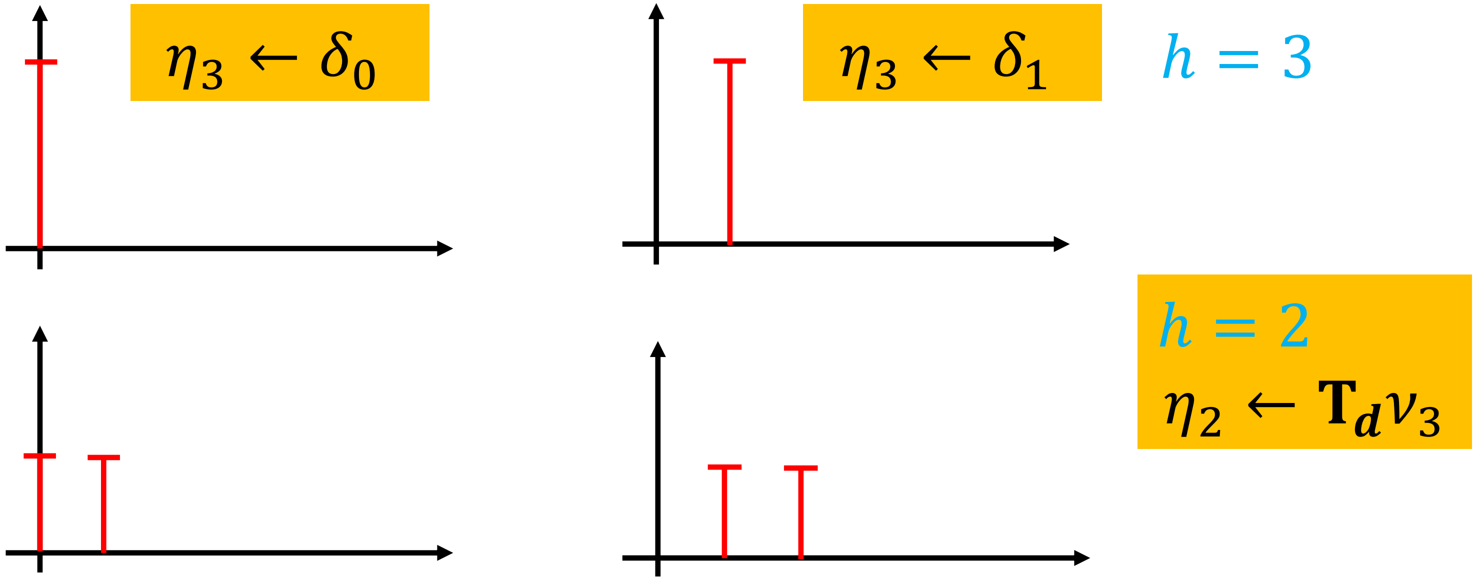




$h = 3$

# Computational Inefficiency of RODI

State space = $\{\ ,\ \}$
Uniform transition
$r(\ ) = 0, r(\ ) = 1$



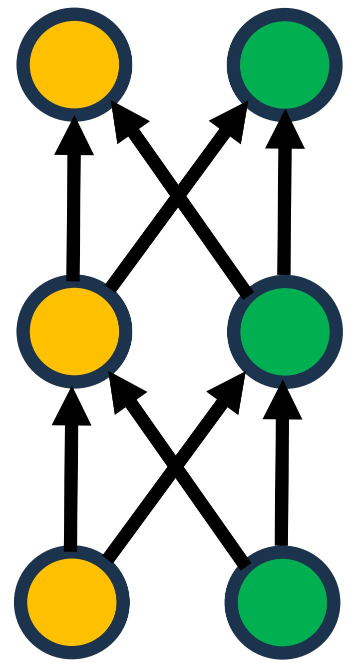
# Computational Inefficiency of RODI

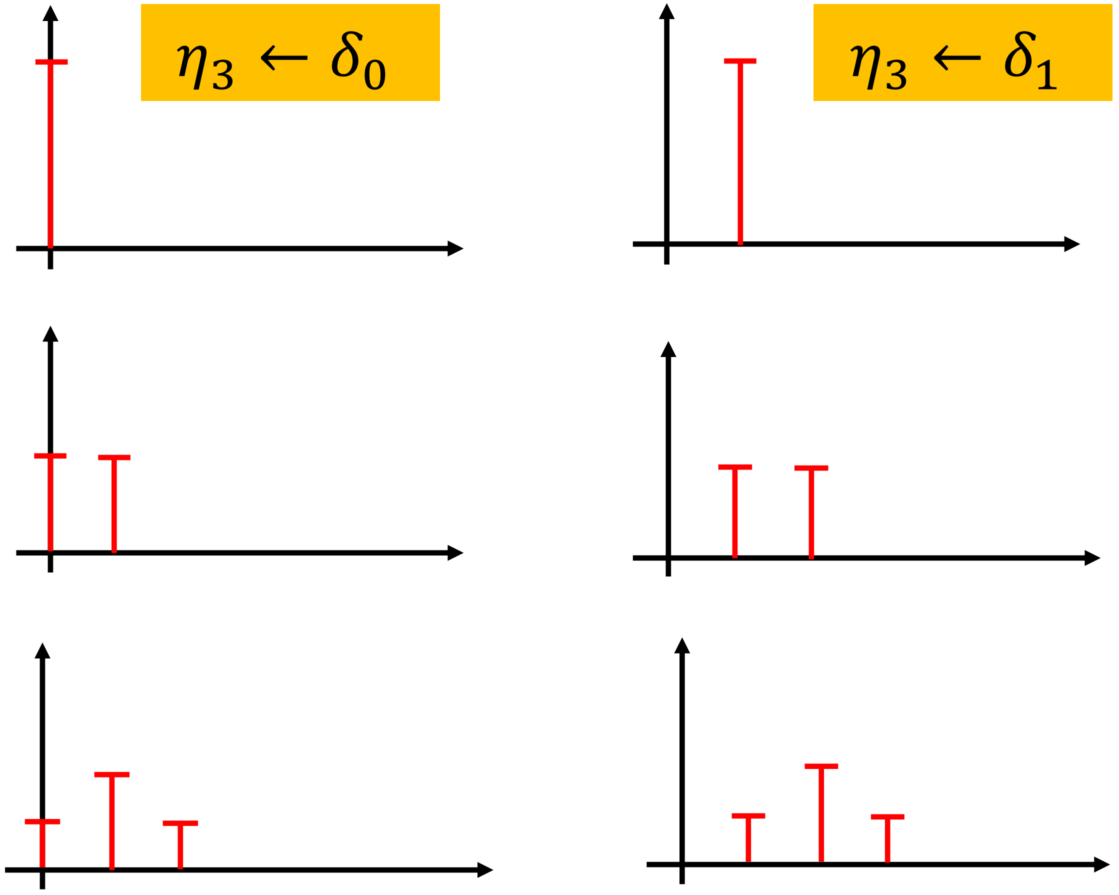
State space = { , }
Uniform transition
$r(\ ) = 0, r(\ ) = 1$

$h = 3$

$h = 2$

$h = 1$





$\eta_3 \leftarrow \delta_0$

$\eta_3 \leftarrow \delta_1$

$h = 3$

$h = 2$
$\eta_2 \leftarrow \mathbf{T}_d \nu_3$

$h = 1$
$\eta_1 \leftarrow \mathbf{T}_d \nu_2$

Operator $\mathbf{T}_d$ increases support size exponentially!

# RODI with Distribution Representation (RODI-Rep)

Operator $\mathbf{T}_{\boldsymbol{d}}$ increases support size exponentially

$$\eta_h \leftarrow \mathbf{T}_{\boldsymbol{d}} \nu_{h+1} \quad \longrightarrow \quad |\eta_h| = S \cdot |\nu_{h+1}|$$

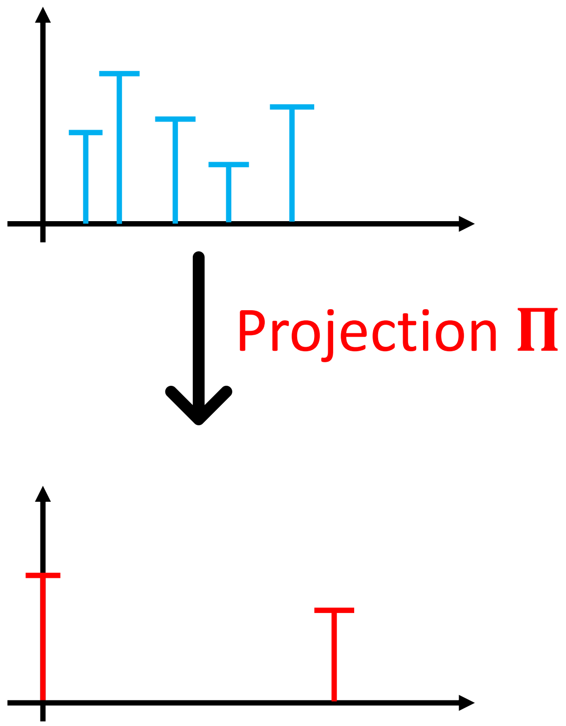
# RODI with Distribution Representation (RODI-Rep)

Operator $\mathbf{T}_d$ increases support size exponentially

$$\eta_h \leftarrow \mathbf{T}_d \nu_{h+1} \longrightarrow |\eta_h| = S \cdot |\nu_{h+1}|$$

Represent distribution with fixed support via projection

$$\eta_h \leftarrow \mathbf{\Pi} \mathbf{T}_d \nu_{h+1} \longrightarrow |\eta_h| = |\nu_{h+1}| = |\eta_{h+1}|$$




Bernoulli representation

# RODI with Distribution Representation (RODI-Rep)

Operator $\mathbf{T}_d$ increases support size exponentially

$$\eta_h \leftarrow \mathbf{T}_d \nu_{h+1} \longrightarrow |\eta_h| = S \cdot |\nu_{h+1}|$$

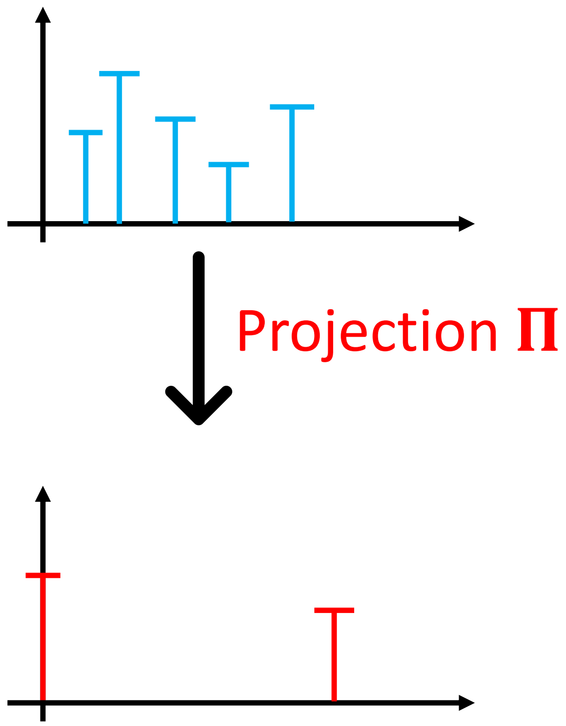
**Represent distribution with fixed support via projection**

$$\eta_h \leftarrow \mathbf{\Pi}\mathbf{T}_d \nu_{h+1} \longrightarrow |\eta_h| = |\nu_{h+1}| = |\eta_{h+1}|$$

**RODI-Rep**

$$\eta_h \leftarrow \mathbf{\Pi}\mathbf{O}_c \widehat{\mathbf{T}}_d \nu_{h+1} \longrightarrow |\eta_h| = |\nu_{h+1}|$$

**Ensure optimism while maintaining computational efficiency**




Bernoulli representation

# Regret Lower Bound

$T := KH$ total time steps

For any algorithm, exists an MDP instance

$$\mathrm{E}[\mathrm{Regret}(K)] \geq \Omega\left(\frac{\exp(\beta H/6) - 1}{\beta}\sqrt{SAT}\right)$$

Corrected result in [3]

$$\mathrm{E}[\mathrm{Regret}(K)] \geq \Omega\left(\frac{\exp(|\beta| H/2) - 1}{|\beta|}\sqrt{K\log K}\right)$$

**Fundamental trade-off between risk sensitivity and sample complexity**

- Fix and tighten the previous lower bound
- Recover tight lower bound under risk-neutral setting
- Hold for $\beta > 0$

**Missing $S, A$**
**Loose dependency on $H$**

- Reduction to 2-armed bandit
- Proof has errors

[3] Fei, Yingjie, et al. "**Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret**." *Advances in Neural Information Processing Systems* 33 (2020): 22384-22395.

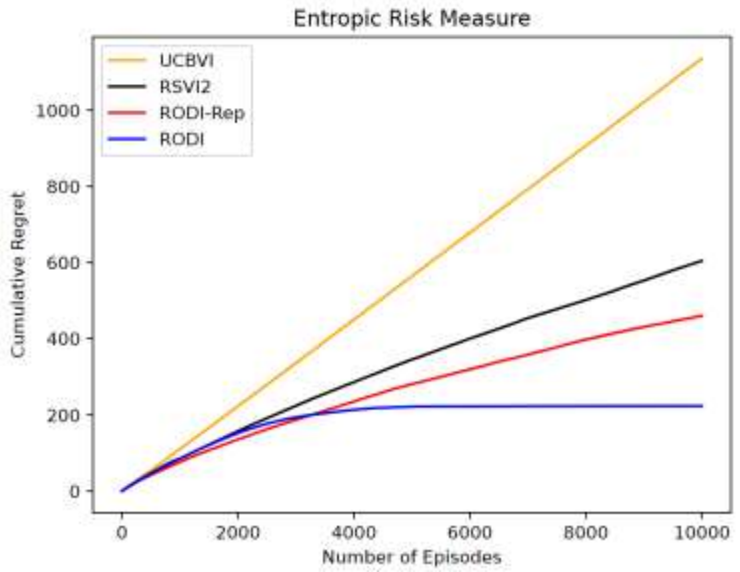
# Regret Upper Bound

| Algorithm | Regret bound | Time | Space |
|---|---|---|---|
| RSVI | $\tilde{\mathcal{O}}\left(\exp(\lvert\beta\rvert H^2)\frac{\exp(\lvert\beta\rvert H)-1}{\lvert\beta\rvert}\sqrt{HS^2AT}\right)$ | $\mathcal{O}\left(TS^2A\right)$ | $\mathcal{O}\left(HSA+T\right)$ |
| RSVI2 | $\tilde{\mathcal{O}}\left(\frac{\exp(\lvert\beta\rvert H)-1}{\lvert\beta\rvert}\sqrt{HS^2AT}\right)$ | $\mathcal{O}\left(TS^2A\right)$ | $\mathcal{O}\left(HSA+T\right)$ |
| RODI-Rep | $\tilde{\mathcal{O}}\left(\frac{\exp(\lvert\beta\rvert H)-1}{\lvert\beta\rvert}\sqrt{HS^2AT}\right)$ | $\mathcal{O}\left(TS^2A\right)$ | $\mathcal{O}\left(HSA+T\right)$ |
| RODI | $\tilde{\mathcal{O}}\left(\frac{\exp(\lvert\beta\rvert H)-1}{\lvert\beta\rvert}\sqrt{HS^2AT}\right)$ | $\mathcal{O}(KS^H)$ | $\mathcal{O}(S^H)$ |
| lower bound | $\Omega\left(\frac{\exp(\beta H/6)-1}{\beta}\sqrt{SAT}\right)$ | - | - |

**RODI and RODI-Rep satisfies**

$$\mathbf{Regret}(K) \le O\left(\frac{\exp(|\beta|H)-1}{|\beta|}\sqrt{HS^2AT}\right)$$

- Matching the best known result in [4]
- Clean and interpretable analysis
- Outperform RSVI2 [4] empirically




[4] Fei, Yingjie, et al. "**Exponential bellman equation and improved regret bounds for risk-sensitive reinforcement learning**." *Advances in Neural Information Processing Systems* 34 (2021): 20436-20446.

# Summary

## Main Contributions

- Risk-Sensitive Distributional Dynamic Programming framework
- Computationally efficient DRL algorithm with near-optimal regret guarantee
- Tight regret lower bound

# Summary

## Main Contributions

- Risk-Sensitive Distributional Dynamic Programming framework
- Computationally efficient DRL algorithm with near-optimal regret guarantee
- Tight regret lower bound

## Future Directions

- Scalability issues with large state-action spaces: approximation techniques
- Multi-Agent RSRL: coordination and cooperation
- Theoretical foundations: connections to robust control
- Application domains: cyber-physical systems

# Thank You