

Recent Advances in Reward Modeling for Reinforcement Learning



Masashi Sugiyama

RIKEN/The University of Tokyo

<http://www.ms.k.u-tokyo.ac.jp/sugi/>



About Myself

2



■ Masashi Sugiyama:

- Director: RIKEN AIP, Japan
- Professor: University of Tokyo, Japan
- Consultant: several local startups

■ Interests: Machine learning (ML)

- ML theory & algorithm →
- ML applications (signal, image, language, brain, robot, mobility, advertisement, biology, medicine, education...)

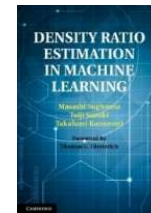
■ Academic activities:

- **Keynote:** ICLR2023, WCCI2024, CoLLAs2026...
- **Program chair:** NeurIPS2015, AISTATS2019, ACML2010/2020, IEEE-CAI2024...
- **Board:** NeurIPS (2016-2022, 2023- Advisory), ICML (2025-), ACML (2010-2021)

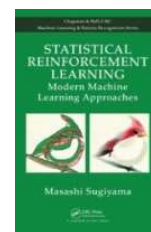
Sugiyama & Kawanabe,
**Machine Learning
in Non-Stationary
Environments**,
MIT Press, 2012



Sugiyama, Suzuki &
Kanamori, **Density Ratio
Estimation in Machine
Learning**, Cambridge
University Press, 2012



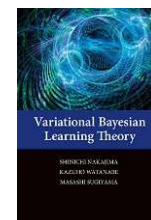
Sugiyama,
**Statistical Reinforcement
Learning**,
Chapman and Hall/CRC,
2015



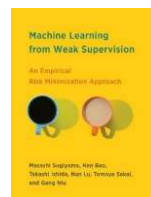
Sugiyama,
**Introduction to Statistical
Machine Learning**,
Morgan Kaufmann,
2015

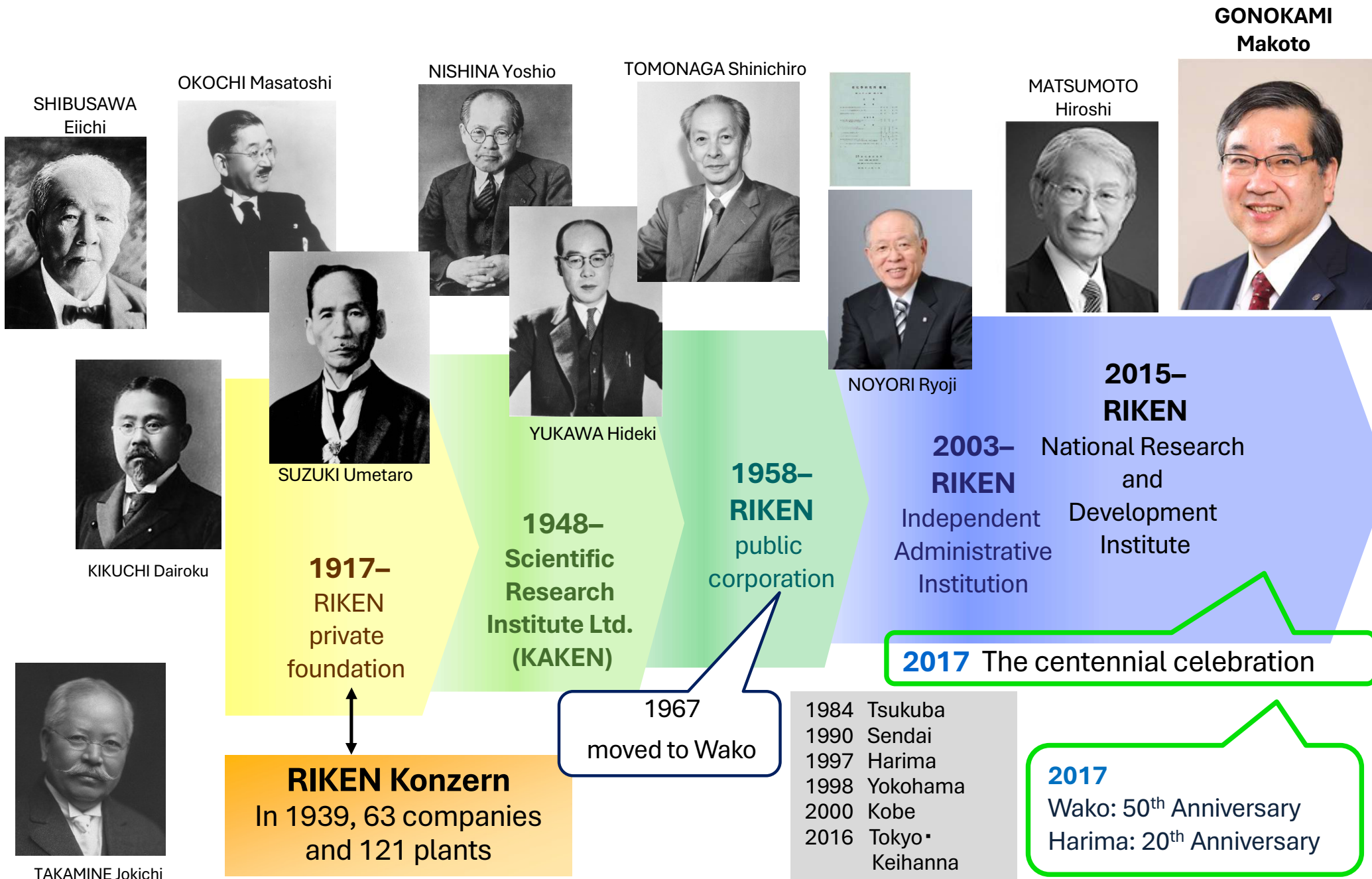


Nakajima, Watanabe &
Sugiyama, **Variational
Bayesian Learning Theory**,
Cambridge University
Press, 2019



Sugiyama, Bao, Ishida,
Lu, Sakai & Niu.
**Machine Learning from
Weak Supervision**,
MIT Press, 2022.





KAWASAKI Masashi



Data and Computational Sciences
Integration Research Program (CoRe)
Advanced General Intelligence
for Science Program (AGIS)
Fundamental Quantum Science Program (FQSP)
Program for Drug Discovery
and Medical Technology Platforms (DMP)
Advanced Semiconductor
Science Program (ASSP)
Industrial Co-creation Program (ICoP)
Baton Zone Program (BZP)

TOKURA Yoshinori



Center for Emergent Matter
Science (CEMS)
Center for Advanced Photonics (RAP)
Nishina Center for
Accelerator-Based Science (RNC)
SPRING-8 Center (RSC)

SAITO Kazuki



Center for Sustainable Resource Science (CSRS)
(BioResource Research Center (BRC))

**Mathematical,
Computational and
Information Science**

**Pioneering
Science**

RIKEN TRIP

**Physical
Science**

**Life
Science**

**Sustainability
Science**

KOTANI Motoko



Pioneering Research Institute (PRI)

HATSUDA Tetsuo



Center for Interdisciplinary Theoretical
and Mathematical Sciences (iTHEMS)
Center for Computational Science (R-CCS)
Center for Quantum Computing (RQC)
Center for Advanced Intelligence Project (AIP)
RIKEN Information R&D
and Strategy Headquarters (R-IH)

NISHIDA Eisuke



Center for Integrative Medical Sciences (IMS)
Center for Biosystems Dynamics Research (BDR)
Center for Brain Science (CBS)
BioResource Research Center (BRC)

*TRIP (Transformative Research Innovation Platform of RIKEN platforms)

■ RIKEN founded Center for Advanced Intelligence Project (AIP) in 2016.

- 117 employed researchers (45% international, 20% female)
- 244 visiting researchers
- 105 domestic students
- 241 international interns (total)

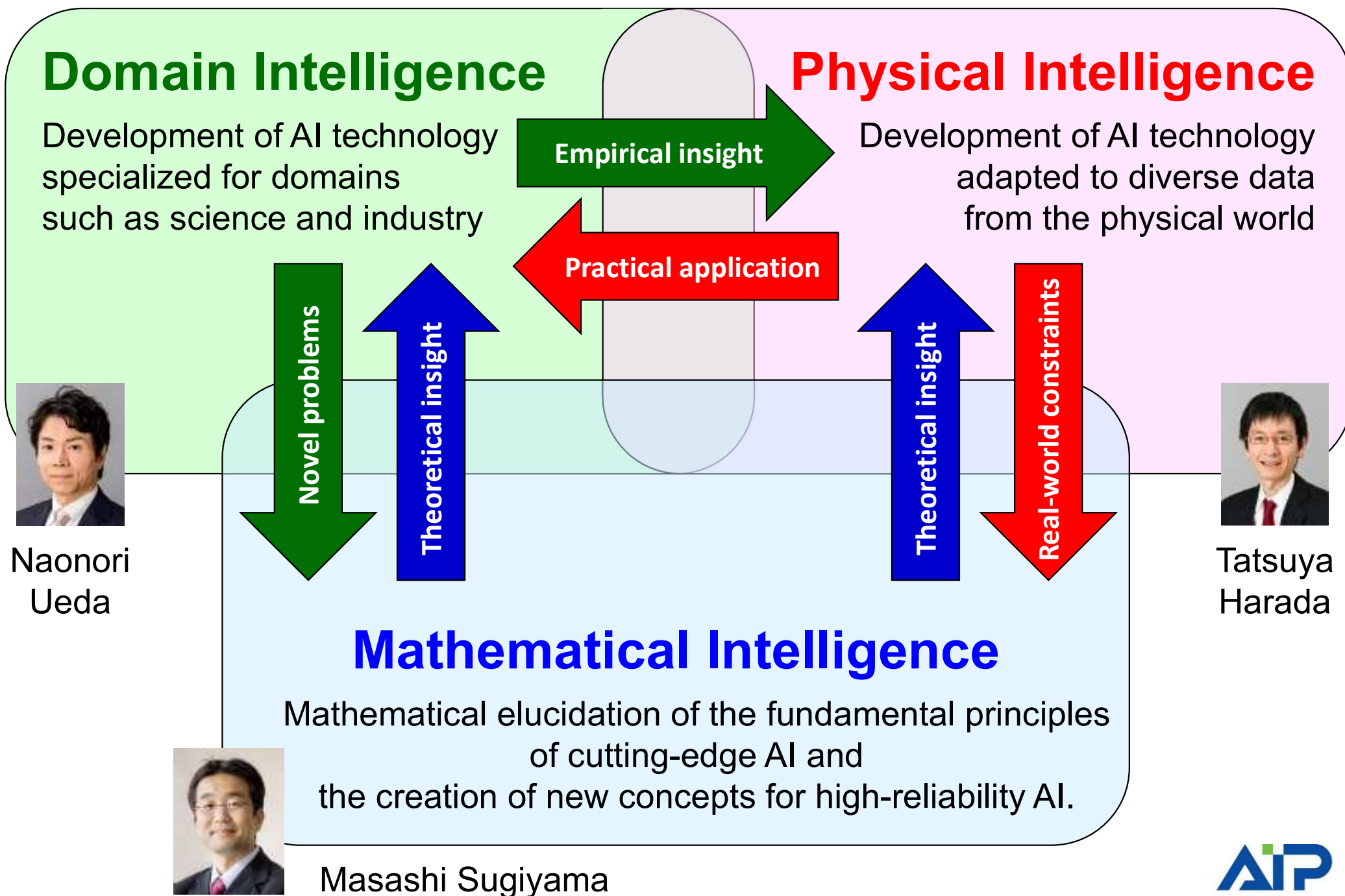


Distributed office across Japan



3 Research Groups

6



Director
Deputy Director
Office of the Center Director
Mathematical Intelligence Group

Group Director : Masashi Sugiyama

Imperfect Information Learning Team	Dynamical Systems Learning Team	Tensor Learning Team	Succinct Information Processing Team	Deep Learning Theory Team	Computational Learning Theory Team
 Masashi Sugiyama	 Yoshinobu Kawahara	 Qibin Zhao	 Yasuo Tabei	 Taiji Suzuki	 Kohei Hatano
Causal Inference Team	Adaptive Bayesian Intelligence Team	Continuous Optimization Team	Mathematical Science Team	Business and Economic Information Fusion Analysis Team	AI Safety and Reliability Unit
 Shohei Shimizu	 Mohammad Emtiyaz Khan	 Akiko Takeda	 Kenichi Bannai	 Takahiro Hoshino	 Hiromi Arai
High-Dimensional Structure Theory Team	Sequential Decision Making Team	Uncertainty Quantification Team	Information Statistical Mechanics and Dynamics Team	Computational Social Choice Team	
 Masaaki Imaizumi	 Shinji Ito	 Futoshi Futami	 Ayaka Sakata	 Ayumi Igarashi	

Domain Intelligence Group

Group Director : Naonori Ueda

AI Medical Engineering Team	Molecular Informatics Team	Cognitive Behavioral Assistive Technology Team	Disaster Resilience Science Team	Multimodal Visual Intelligence Team	Data-Driven Experimental Design Team
 Ryuji Hamamoto	 Koji Tsuda	 Mihoko Otake	 Naonori Ueda	 Takayuki Okatani	 Ichiro Takeuchi
Statistical Genetics Team	Biomedical Spatial Science Team	Natural Language Understanding Team	Geoinformatics Team	Music Information Intelligence Team	Computational Physics Machine Learning Team
 Gen Tamiya	 Yoichiro Yamamoto	 Kentaro Inui	 Naoto Yokoya	 Masatoshi Hamanaka	 Takaharu Yaguchi
Statistical Astrophysics Team	Chemical Reaction Informatics Team				
 Naoki Yoshida	 Ichigaku Takigawa				

Physical Intelligence Group

Group Director : Tatsuya Harada

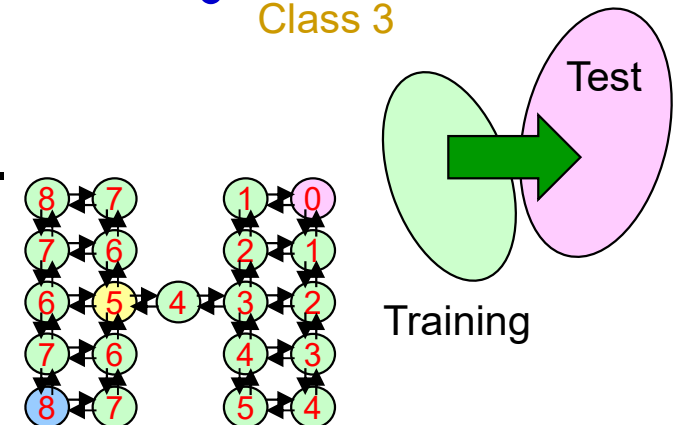
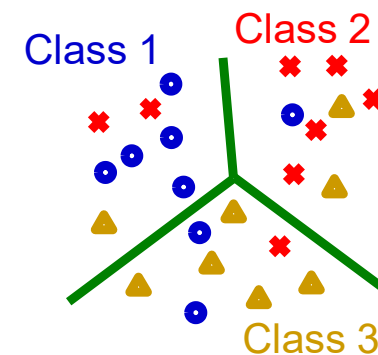
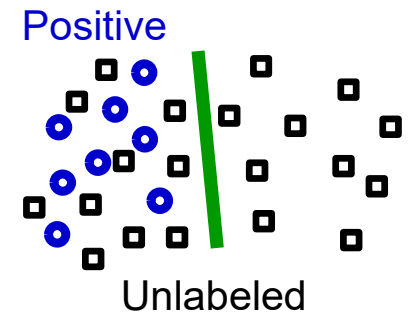
Real-World Intelligence Team	AI Security and Privacy Team	Robot Learning Team	Robot System Team	AI Computing Team	3D Environmental Information Understanding Team
 Tatsuya Harada	 Jun Sakuma	 Takayuki Osa	 Kei Okada	 Shinya Takamaeda	 Asako Kanezaki
Explainable AI Team					
 Hitomi Yanaka					

Computing Support and Operations Unit

Machine Learning from Imperfect Information

■ Reliability of ML systems depends on training data:

- **Weakly supervised learning:**
Training data contains insufficient supervision.
- **Label-noise robust learning:**
Labels are prone to observation error and human mistakes.
- **Distribution shift adaptation:**
Training data is biased due to changing environments and privacy.
- **Reinforcement learning:**
Only indirect feedback (reward) is given.



■ Improving reliability is a long-standing challenge!

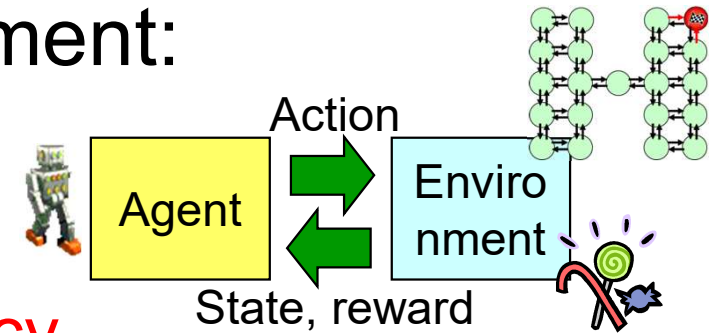
Reinforcement Learning (RL)

9

Sutton & Barto (MIT Press 2018)

■ An agent interacts with the environment:

- Take an action, make a transition, and receive a reward.
- Learn an optimal decision-making policy that maximizes aggregated rewards.



■ Many successful applications:

- Games, robots, language models,...

■ User Input: Rewards

- **Success:** Positive reward (“Praising”)
- **Failure:** Negative reward (“Punishing”)

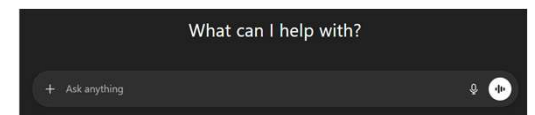
■ Designing (or learning) the rewards has been a long-standing challenge!



<https://www.itmedia.co.jp/news/articles/1705/23/news122.html>



Sugimoto+ (IEEE-RAM2016)



<https://chatgpt.com/>



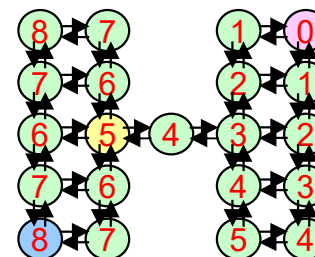
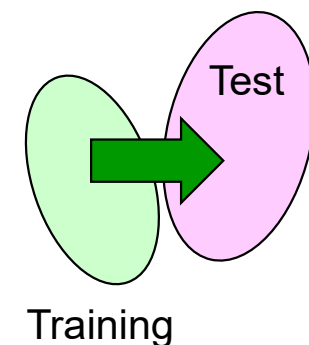
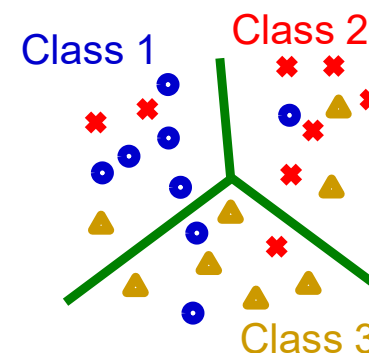
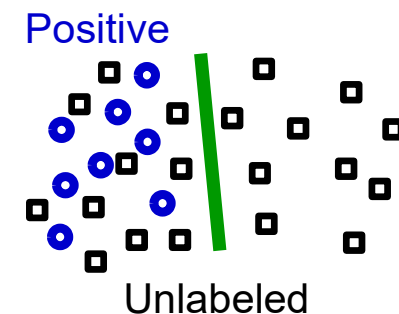
Contents

10

1. Background materials

- A) Weakly supervised learning
- B) Label-noise robust learning
- C) Distribution shift adaptation

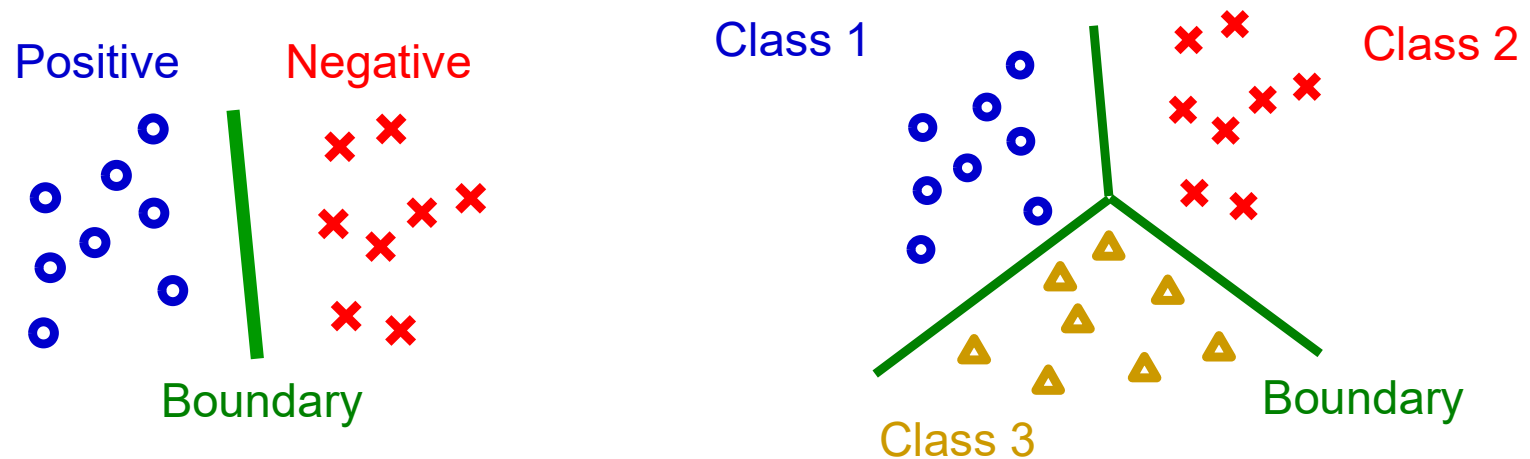
2. Reward modeling in RL



Weakly Supervised Learning

11

- Supervised classification from big labeled data is successful: speech, image, language, ...



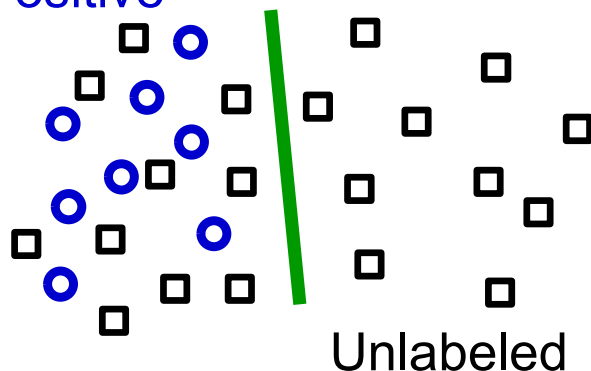
- However, there are many applications where **big labeled data is not available**:
 - Medicine, disaster, robot, brain, ...
- We want to utilize “**weak**” supervision that can be collected easily!

Weakly Supervised Learning

12

Positive-Unlabeled

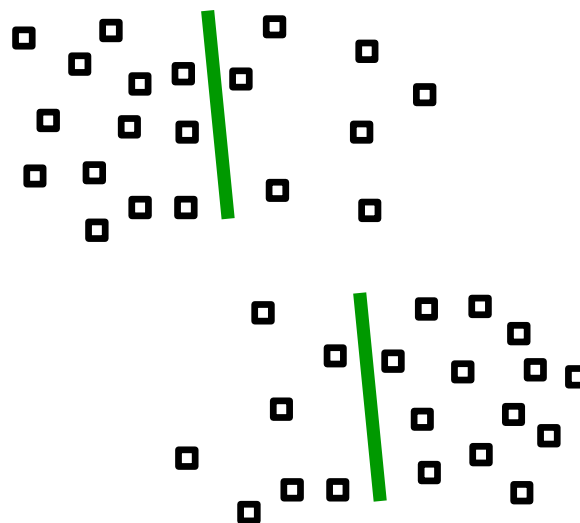
Positive



du Plessis+ (NeurIPS2014, ICML2015, MLJ2017),
Niu+ (NeurIPS2016), Kiryo+ (NeurIPS2017), Hsieh+ (ICML2019)

Ex.: Click Prediction

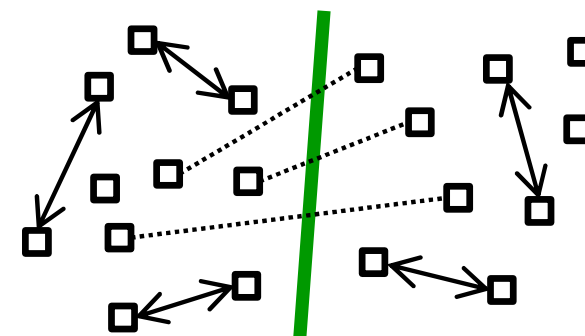
Unlabeled-Unlabeled



du Plessis+ (TAAI2013), Lu+ (ICLR2019, AISTATS2020),
Charoenphakdee+ (ICML2019), Lei+ (ICML2021)

Ex.: Multi-populations

Similar-Dissimilar-Unlabeled



Bao+ (ICML2018), Shimada+ (NeCo2021),
Dan+ (ECMLPKDD2021), Cao+ (ICML2021),
Feng+ (ICML2021)

Ex.: Sensitive
Information Prediction

Multi-class extensions:

- Noisy labels, ambiguous labels, complementary labels...

Ishida+ (NeurIPS2017, ICML2019), Chou+ (ICML2020), Feng+ (ICML2020, NeurIPS2020), Lv+ (ICML2020), Cao+ (arXiv2021)

Applicable to any loss, classifier, optimizer & regularizer!

Recent advances:

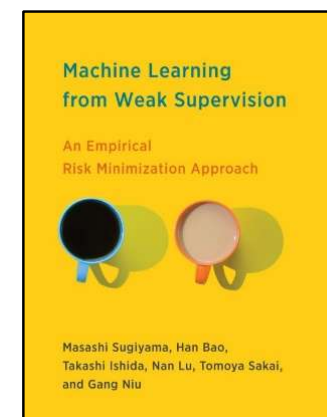
- Unified framework, new problems and algorithms...

Chiang+ (TMLR2025), Chen+ (ICML2024), Lv+ (NeurIPS2024), Wang+ (NeurIPS2023, ICML2024, ICLR2025),

- Speech denoising, reinforcement and imitation learning, large language models...

Ito & Sugiyama (ICASSP2023, Best Paper Award), Cai+ (NeurIPS2023), Nishimori+ (RLC2025), Zhang+ (ICML2024), Li+ (MLJ2025)

Sugiyama, Bao, Ishida,
Lu, Sakai & Niu,
MIT Press, 320 pages, 2022.





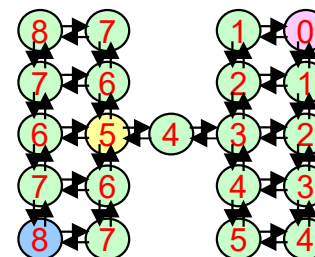
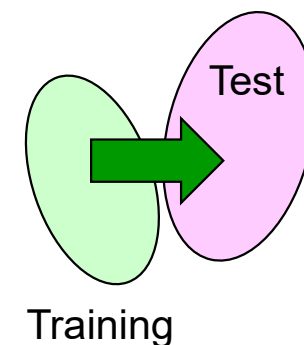
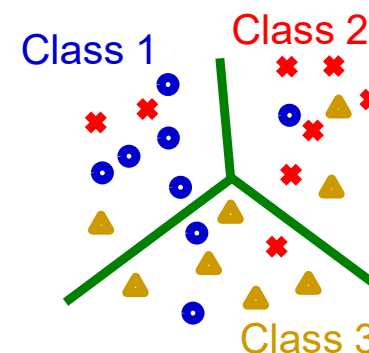
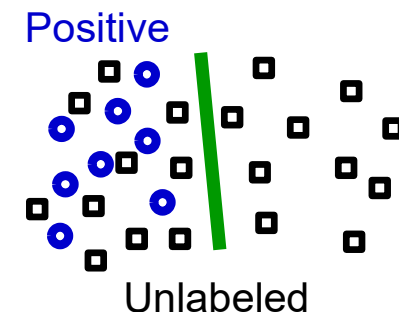
Contents

16

1. Background materials

- A) Weakly supervised learning
- B) Label-noise robust learning
- C) Distribution shift adaptation

2. Reward modeling in RL



Distribution Shift Adaptation

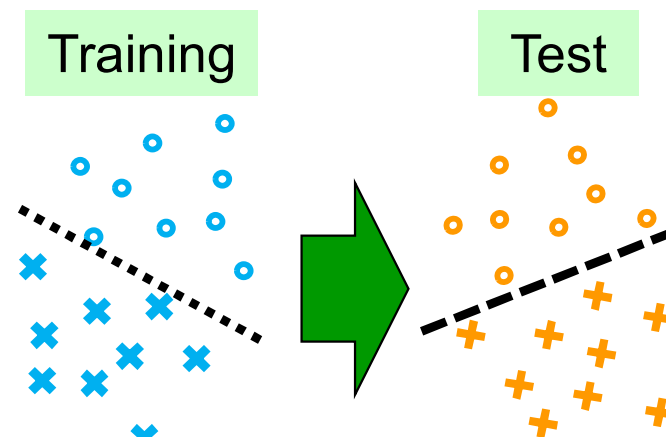
17

■ Training/test data often follow **different distributions**:

- Changing environments,
- Sample selection bias (privacy).

■ **Distribution shift adaptation**:

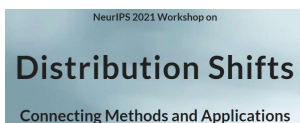
- Train a test-domain predictor using training data from different domains.



Quiñonero-Candela+
(MIT Press 2009)

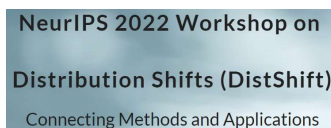


• **New names**: Distribution shift, fine-tuning...



NeurIPS 2023 Workshop on Distribution Shifts (DistShift)

New Frontiers with Foundation Models



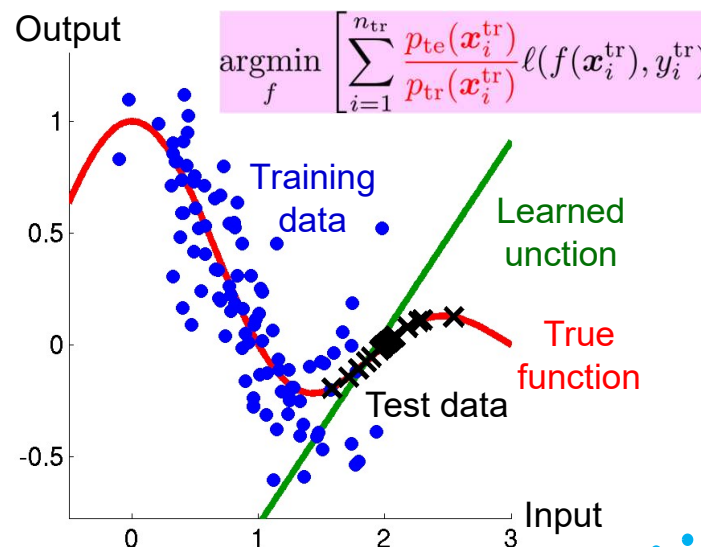
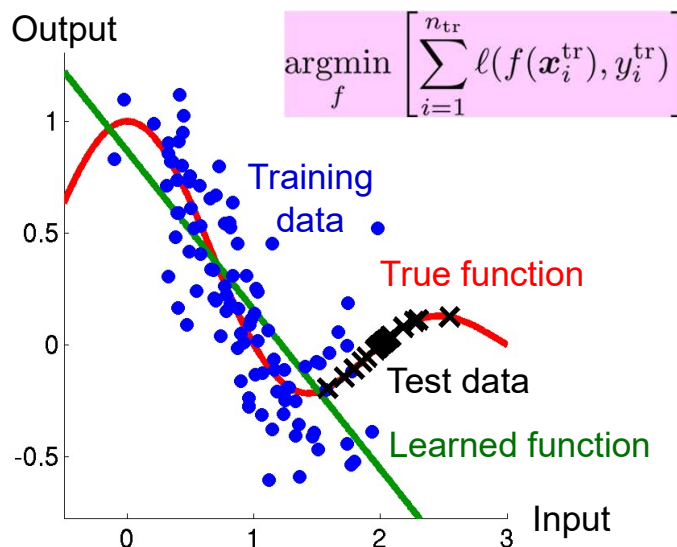
Reliable ML from Unreliable Data

Workshop @ NeurIPS 2025

■ Covariate shift adaptation by **importance-weighting**:

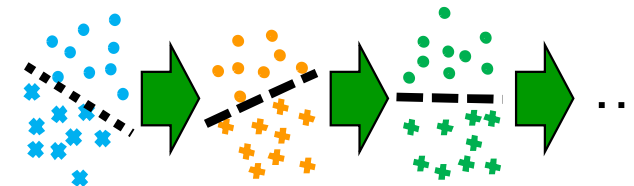
- Importance estimation followed by weighted training.

Sugiyama & Kawanabe
(MIT Press 2012)



■ Recent advances:

- Joint importance-predictor estimation.
- **Sequential** distribution shifts.
- **Joint** distribution shifts.
- **Out-of-distribution** extension.
- Applications in language model alignment and test-time adaptation.



Zhang+ (ACML2020 Best Paper Award, SNCS2021)

Bai+ (NeurIPS2022), Zhang+ (NeurIPS2023, ICML2025),
Qian+ (ICML2024)

Fang+ (NeurIPS2020 Spotlight)

Fang+ (NeurIPS2023 Spotlight)

Lodkaew+ (TMLR2025), Ackermann+ (COLM2025)



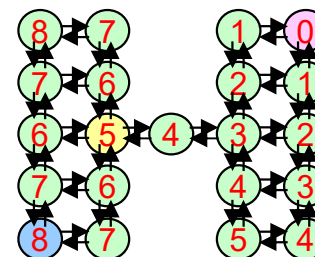
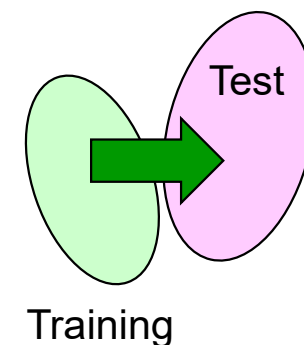
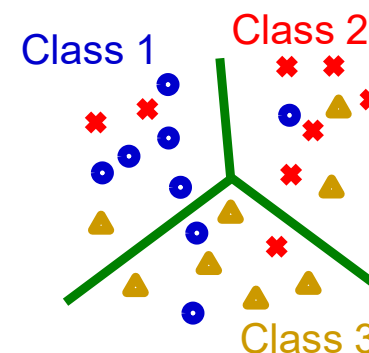
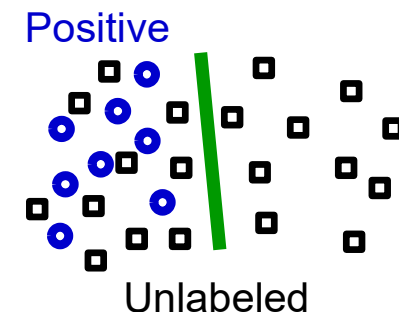
Contents

19

1. Background materials

2. Reward modeling in RL

- A) Distribution shift in language model alignment
- B) Recursive reward aggregation
- C) Further developments



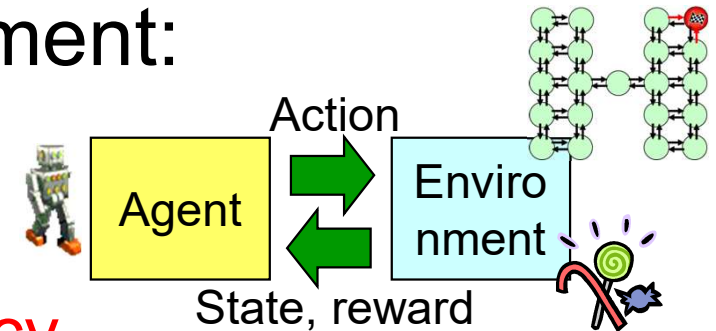
Reinforcement Learning (RL)

20

Sutton & Barto (MIT Press 2018)

■ An agent interacts with the environment:

- Take an action, make a transition, and receive a reward.
- Learn an optimal decision-making policy that maximizes aggregated rewards.



■ Many successful applications:

- Games, robots, language models,...

■ User Input: Rewards

- **Success:** Positive reward (“Praising”)
- **Failure:** Negative reward (“Punishing”)

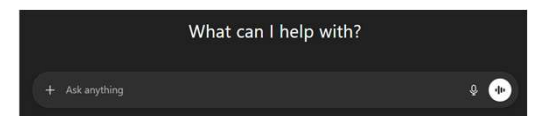
■ Designing (or learning) the rewards has been a long-standing challenge!



<https://www.itmedia.co.jp/news/articles/1705/23/news122.html>



Sugimoto+ (IEEE-RAM2016)



<https://chatgpt.com/>



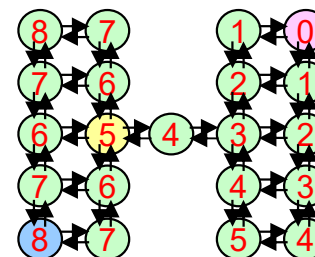
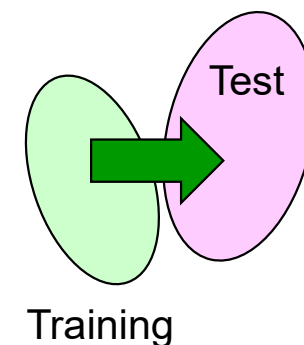
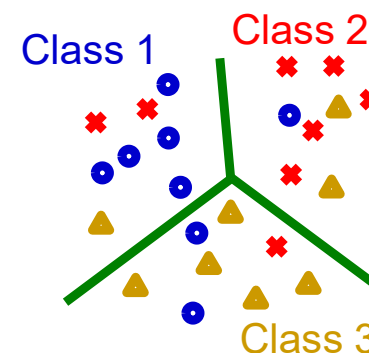
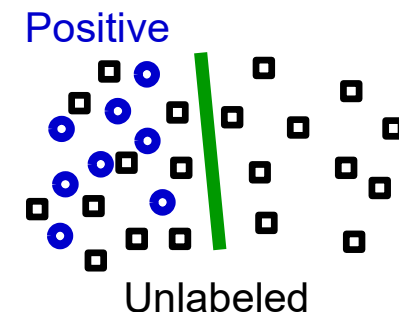
Contents

21

1. Background materials

2. Reward modeling in RL

- A) Distribution shift in language model alignment
- B) Recursive reward aggregation
- C) Further developments



Large Language Model Alignment 22

- **State s** : Question “What is the best pet?”
- **Action a** : Answer “Dog.”
- True reward $R(s, a)$: unknown.
 - $R(\text{“What is the best pet?”}, \text{“Dog.”}) = 1.0$
 - $R(\text{“What is the best pet?”}, \text{“Cat.”}) = 0.8$
- Reward is learned from **human feedback**.
 - Reward model $\pi^1(s)$ is used, which finetunes pre-trained model $\pi^0(s)$ with expert data $\{(s, a)\}$.

Reinforcement Learning from Human Feedback (RLHF)

■ **Preference data** $(s, a_w, a_l) \sim P_{\pi^1}(s, a_w, a_l)$:

Christiano+
(NeurIPS2017)

- For question s , obtain preferred and unpreferred responses a_w and a_l from P_{π^1} .
- Assume the **Bradley-Terry** model: Bradley+ (Biometrika1952)

$$P(a_w > a_l | s) = \sigma(R(s, a_w) - R(s, a_l))$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

■ **Train reward model R_θ** by maximum likelihood:

$$\hat{\theta} = \operatorname{argmax}_{\theta} \mathbb{E}_{(s, a_w, a_l) \sim P_{\pi^1}} [\log \sigma(R_\theta(s, a_w) - R_\theta(s, a_l))]$$

■ Use learned reward $R_{\hat{\theta}}$ to train policy $\pi_\phi(a|s)$ by **KL-penalized policy gradient**:

$$\epsilon, \beta > 0$$

$$\phi \leftarrow \phi + \epsilon \nabla_{\phi} \left\{ \mathbb{E}_{a \sim \pi_{\phi}(s), s \sim P(s)} [R_{\hat{\theta}}(s, a)] - \beta D_{\text{KL}}(\pi_{\phi} || \pi^1) \right\}$$

Distribution Shift in Reward Modeling²⁴

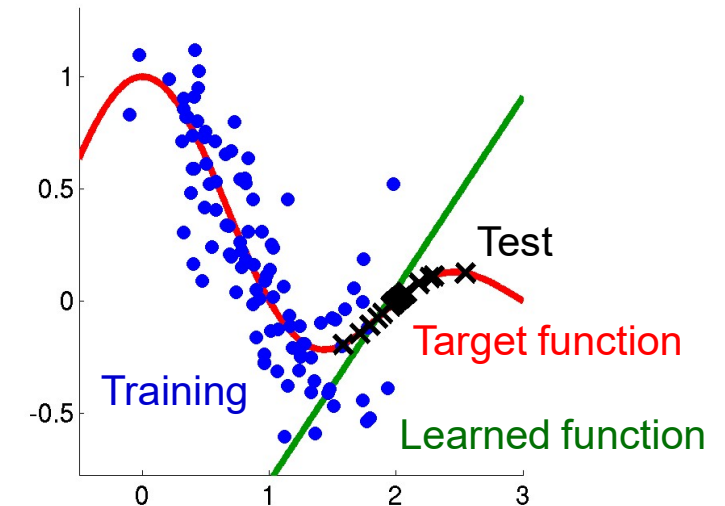
Ackermann+ (COLM2025)

■ Training and evaluation:

- Use data from $\pi^1(s)$ for training.
- At the i -th iteration, use data from $\pi^i(s)$ for evaluation.

■ Distribution shift occurs!

■ Importance-weighting:



$$\max_{\theta} \hat{\mathbb{E}}_{(s, a_w, a_l) \sim P_{\pi^1}} [w(s, a_w, a_l) \log \sigma(R_{\theta}(s, a_w) - R_{\theta}(s, a_l))]$$

$$w(s, a_w, a_l) = \frac{P(s) \pi^i(a_w|s) \pi^i(a_l|s) P(a_w > a_l|s)}{P(s) \pi^1(a_w|s) \pi^1(a_l|s) P(a_w > a_l|s)} = \frac{\pi^i(a_w|s) \pi^i(a_l|s)}{\pi^1(a_w|s) \pi^1(a_l|s)}$$

- Reward model is updated every k iterations.

Direct Policy Optimization (DPO)²⁶

Rafailov+ (NeurIPS2023)

■ RLHF:

1. Train a reward model from human feedback.

$$\hat{\theta} = \operatorname{argmax}_{\theta} \hat{\mathbb{E}}_{(s, a_w, a_l) \sim P_{\pi^1}} [\log \sigma(R_{\theta}(s, a_w) - R_{\theta}(s, a_l))]$$

2. Use the learned reward to train a policy.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\hat{\phi} = \operatorname{argmax}_{\phi} \hat{\mathbb{E}}_{a \sim \pi_{\phi}(s), s \sim P(s)} [R_{\hat{\theta}}(s, a)] - \beta D_{\text{KL}}(\pi_{\phi} || \pi^1)$$

$$\beta > 0$$

■ DPO:

- Directly train a policy with the analytic optimal reward:

$$r_{\phi}(s, a) = \beta \log \frac{\pi_{\phi}(a | s)}{\pi^1(a | s)}$$

$$\hat{\phi} = \operatorname{argmax}_{\phi} \hat{\mathbb{E}}_{(s, a_w, a_l) \sim P_{\pi^1}} [\log \sigma(r_{\phi}(s, a_w) - r_{\phi}(s, a_l))]$$

Distribution Shift in LLM Deployment²⁷

Lodkaew+ (TMLR2025)

■ Collect preference data $(s, a_1, a_2, b) \sim P_{\pi^1}(s, a_1, a_2, b)$:

- For question s , prepare responses a_1, a_2 such that $b = 1$ if a_1 is more preferred; otherwise $b = -1$.
- Assume the Bradley-Terry model:

$$P(b, |a_1, a_2, s) = \sigma(b\{R(s, a_1) - R(s, a_2)\})$$

■ Distribution shift in model deployment $P_{\text{test}}(s, a_1, a_2, b)$:

- Prompt shift $P_{\text{test}}(s)$
- Response shift $P_{\text{test}}(a_1, a_2 | s)$
- Preference shift $P_{\text{test}}(b, |a_1, a_2, s)$

Type of shift	Factor		
	1	2	3
a No shift			
b Full shift	✓	✓	✓
c Prompt shift	✓		
d Response shift		✓	
e Preference label shift			✓
f Prompt + response shift	✓	✓	
g Prompt + preference label shift	✓		✓
h Response + preference label shift		✓	✓

■ Importance-weighted training:

$$\max_{\phi} \hat{\mathbb{E}}_{(s, a_1, a_2, b) \sim P_{\pi^1}} w(s, a_1, a_2, b) \ell_{\text{DPO}}(s, a_1, a_2, b)$$

$$\ell_{\text{DPO}}(s, a_1, a_2, b) = \log \sigma \left(b(r_{\phi}(s, a_1) - r_{\phi}(s, a_2)) \right)$$

$$r_{\phi}(s, a) = \beta \log \frac{\pi_{\phi}(a | s)}{\pi^1(a | s)}$$

$$w(s, a_1, a_2, b) = \frac{P_{\text{test}}(s, a_1, a_2, b)}{P_{\pi^1}(s, a_1, a_2, b)} = \frac{P_{\text{test}}(s) P_{\text{test}}(a_1, a_2 | s) P_{\text{test}}(b | a_1, a_2, s)}{P_{\pi^1}(s) P_{\pi^1}(a_1, a_2 | s) P_{\pi^1}(b | a_1, a_2, s)}$$

■ Practical implementation

- **IW-DPO-L**: loss-value-based importance Fang+ (NeurIPS2020)

$$w(s, a_1, a_2, b) \approx \frac{P_{\text{test}}(\ell_{\text{DPO}}(s, a_1, a_2, b))}{P_{\pi^1}(\ell_{\text{DPO}}(s, a_1, a_2, b))}$$

- **IW-DPO-R**: reward-based importance

$$w(s, a_1, a_2, b) \approx \frac{P_{\text{test}}(r_{\phi}(s, a_1), r_{\phi}(s, a_2))}{P_{\pi^1}(r_{\phi}(s, a_1), r_{\phi}(s, a_2))}$$



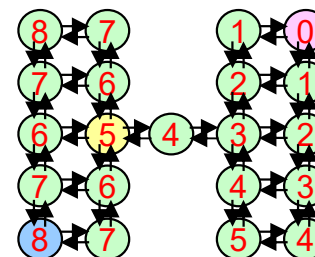
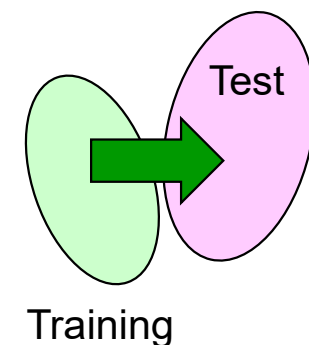
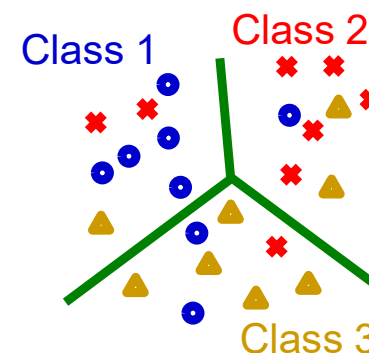
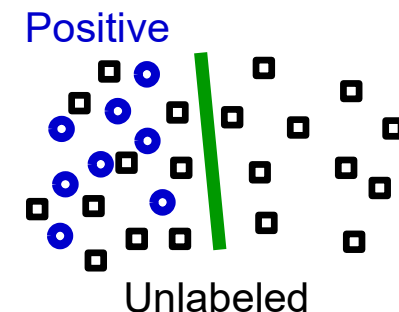
Contents

30

1. Background materials

2. Reward modeling in RL

- A) Distribution shift in language model alignment
- B) Recursive reward aggregation
- C) Further developments



Generalizing Reward Aggregation 31

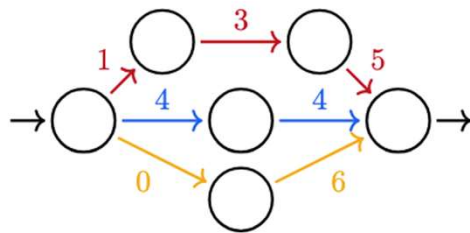
- Standard RL aggregates rewards by **a discounted sum**.

$$\sum_t \gamma^t R(s_t, a_t, s_{t+1})$$

- But other aggregation methods are often useful:

- Drug discovery:** Maximum reward in an interval.

- Portfolio optimization:** Sharpe ratio (mean/std).



	sum [↑]	mean [↑]	max [↑]	min [↑]	top-2 [↑]	range [↓]
π_1	9	3	5	1	3	4
π_2	8	4	4	4	4	0
π_3	6	3	6	0	0	6

- We want to handle various aggregation methods in a **unified manner**!

Recursion in Reward Aggregation 32

- Discounted sum can be **recursively** expressed:

$$\begin{aligned}\text{sum}_\gamma[r_1, r_2, r_3, \dots] &\equiv r_1 + \gamma r_2 + \gamma^2 r_3 + \dots \\ &= r_1 + \gamma(r_2 + \gamma r_3 + \dots) \\ &= r_1 + \gamma \text{sum}_\gamma[r_2, r_3, \dots]\end{aligned}$$

- RL utilizes this recursive form effectively:

- **Ex.:** Bellman equation for state-value functions.

$$V^\pi(s) = \mathbb{E} \left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \middle| s_0 = s \right) = \mathbb{E} [R(s, a, s') + \gamma V^\pi(s')]$$

- **Generalized reward aggregations:** Tang+ (RLC2025)

- Initialization: $\tau_1 \leftarrow \text{init}$;
- Update: $\tau_t \leftarrow \text{update}(r_t, \tau_{t-1})$;
- Post-process: $\tau_t \leftarrow \text{post}(\tau_t)$;

Discounted sum:

- $\text{init} = 0$;
- $\text{update}(r, \tau) = r + \gamma\tau$;
- $\text{post}(\tau) = \tau$;

Various Examples of Recursive Reward Expressions

Tang+ (RLC2025)

	definition $\text{post} \circ \text{agg}_{\text{init}, \triangleright} : [R] \rightarrow R$	initial value of statistic(s) $\text{init} \in T$	update function $\triangleright : R \times T \rightarrow T$	post-processing $\text{post} : T \rightarrow R$
discounted sum	$r_1 + \gamma r_2 + \dots + \gamma^{t-1} r_t$	discounted sum s : $0 \in \mathbb{R}$	$+_\gamma := [r, s \mapsto r + \gamma \cdot s]$	$\text{id}_{\mathbb{R}}$
discounted min	$\min\{r_1, \gamma r_2, \dots, \gamma^{t-1} r_t\}$	discounted min n : $\infty \in \overline{\mathbb{R}}$	$\min_\gamma := [r, n \mapsto \min(r, \gamma \cdot n)]$	$\text{id}_{\overline{\mathbb{R}}}$
discounted max	$\max\{r_1, \gamma r_2, \dots, \gamma^{t-1} r_t\}$	discounted max m : $-\infty \in \overline{\mathbb{R}}$	$\max_\gamma := [r, m \mapsto \max(r, \gamma \cdot m)]$	$\text{id}_{\overline{\mathbb{R}}}$
log-sum-exp	$\log(e^{r_1} + e^{r_2} + \dots + e^{r_t})$	log-sum-exp m : $-\infty \in \overline{\mathbb{R}}$	$[r, m \mapsto \log(e^r + e^m)]$	$\text{id}_{\overline{\mathbb{R}}}$
range	$\max(r_{1:t}) - \min(r_{1:t})$	max m $\begin{bmatrix} -\infty \\ \infty \end{bmatrix} \in \overline{\mathbb{R}}^2$ min n	$\left[r, \begin{bmatrix} m \\ n \end{bmatrix} \mapsto \begin{bmatrix} \max(r, m) \\ \min(r, n) \end{bmatrix} \right]$	$\left[\begin{bmatrix} m \\ n \end{bmatrix} \mapsto m - n \right]$
mean	$\bar{r} := \frac{1}{t} \sum_{i=1}^t r_i$	length n $\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} \mathbb{N} \\ \mathbb{R} \end{bmatrix}$ sum s	$\left[r, \begin{bmatrix} n \\ s \end{bmatrix} \mapsto \begin{bmatrix} n+1 \\ s+r \end{bmatrix} \right]$	$\left[\begin{bmatrix} n \\ s \end{bmatrix} \mapsto \frac{s}{n} \right]$
		length n $\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} \mathbb{N} \\ \mathbb{R} \end{bmatrix}$ mean m	$\left[r, \begin{bmatrix} n \\ m \end{bmatrix} \mapsto \begin{bmatrix} n+1 \\ \frac{n \cdot m + r}{n+1} \end{bmatrix} \right]$	$\left[\begin{bmatrix} n \\ m \end{bmatrix} \mapsto m \right]$
variance	$\frac{1}{t} \sum_{i=1}^t (r_i - \bar{r})^2 = \overline{r^2} - \bar{r}^2$	length n $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} \mathbb{N} \\ \mathbb{R} \\ \mathbb{R}_{\geq 0} \end{bmatrix}$ sum s sum square q	$\left[r, \begin{bmatrix} n \\ s \\ q \end{bmatrix} \mapsto \begin{bmatrix} n+1 \\ s+r \\ q+r^2 \end{bmatrix} \right]$	$\left[\begin{bmatrix} n \\ s \\ q \end{bmatrix} \mapsto \frac{q}{n} - \left(\frac{s}{n}\right)^2 \right]$
		length n $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} \mathbb{N} \\ \mathbb{R} \\ \mathbb{R}_{\geq 0} \end{bmatrix}$ mean m variance v	$\left[r, \begin{bmatrix} n \\ m \\ v \end{bmatrix} \mapsto \begin{bmatrix} n+1 \\ \frac{n \cdot m + r}{n+1} \\ v + \frac{n(r-m)^2 - (n+1)v}{(n+1)^2} \end{bmatrix} \right]$	$\left[\begin{bmatrix} n \\ m \\ v \end{bmatrix} \mapsto v \right]$
top- k	k -th largest in $r_{1:t}$	top-1 $\begin{bmatrix} -\infty \\ -\infty \\ \vdots \end{bmatrix} \in \overline{\mathbb{R}}^k$ top-2 buffer $\vdots$	$\left[r, b \mapsto \begin{cases} \text{insert}(r, b) & r > \min b \\ b & r \leq \min b \end{cases} \right]$	$[b \mapsto \min b]$

■ Q-learning: Watkins & Dayan (MLJ1992)

Algorithm 2 Q-learning (Watkins and Dayan, 1992) with recursive reward aggregation

```
1: Input: transition function  $p : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ , reward function  $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ , terminal  
   condition  $\omega$ , recursive reward aggregation function  $\text{post} \circ \text{agg}_{\text{init}, \triangleright} : [\mathcal{R}] \rightarrow \mathcal{R}$   
2: Parameters: learning rate  $\alpha \in (0, 1]$ , exploration parameter  $\epsilon \in (0, 1)$   
3: Initialize state-action statistic function  $\tau : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{T}$  with initial value  $\text{init} \in \mathcal{T}$   
4: for each episode do  
5:   Initialize state  $s$   
6:   while  $s$  is not terminal do  
7:     Compute state-action value function  $q(s, a) = \text{post}(\tau(s, a))$  for state  $s$  and all  
       actions  $a$   
8:     Select action  $a$  using  $\epsilon$ -greedy policy based on value function  $q(s, a)$   
9:     Execute action  $a$ , observe next state  $s'$  and reward  $r$   
10:    Update state-action statistic function  $\tau$ :  
11:     $\tau(s, a) \leftarrow \tau(s, a) + \alpha \left( \max_{a' \in \mathcal{A}} (r \triangleright \tau(s', a')) - \tau(s, a) \right)$ ,  
12:    where  $\max_{a' \in \mathcal{A}} (r \triangleright \tau(s', a')) = r \triangleright \tau(s', a^*)$  and  $a^* = \arg \max_{a' \in \mathcal{A}} \text{post}(r \triangleright \tau(s', a'))$   
13:    Update state  $s \leftarrow s'$   
14:  end while  
15: end for  
16: Output: estimated optimal statistic function  $\tau$  and optimal policy  $\pi(s) =$   
        $\arg \max_{a \in \mathcal{A}} q(s, a)$ , where  $q(s, a) = \text{post}(\tau(s, a))$ 
```

Convert state-action statistics
to state-action values

Bellman equation for
state-action statistics

● Similarly, we can extend

■ Proximal policy optimization (PPO);

■ Twin delayed deep deterministic policy gradient (TD3).

Schulman+ (arXiv2017)

Fujimoto+ (ICML2018)



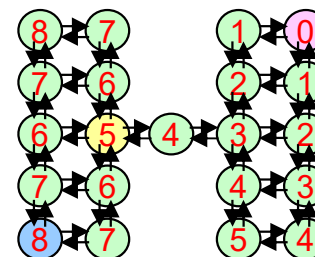
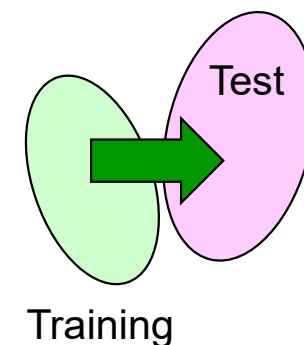
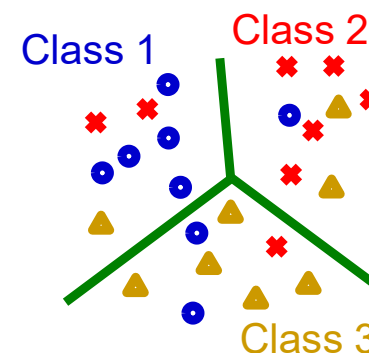
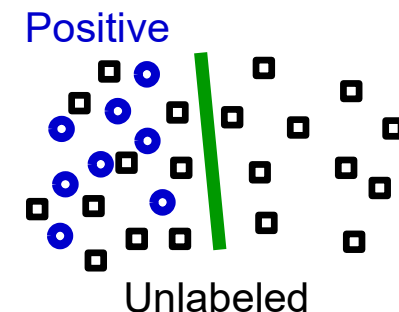
Contents

35

1. Background materials

2. Reward modeling in RL

- A) Distribution shift in language model alignment
- B) Recursive reward aggregation
- C) Further developments



Imitation Learning from Weak and Noisy Supervision

- Learning reward from demonstrations by **inverse RL** and **imitation learning**.

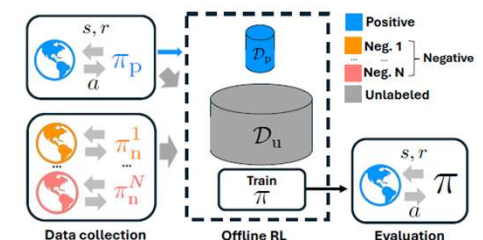
Ng & Russell (ICML2000), Ho & Ermon (NeurIPS2016)

- Extensions to **weak and noisy demonstrations**:

Wu+ (ICML2019), Tangkaratt+ (ICML2000, AISTATS2021), Cai+ (ICLR2023, NeurIPS2023)

- Multi-domain RL with **positive-unlabeled domain labels**.

Nishimori+ (RLC2025)

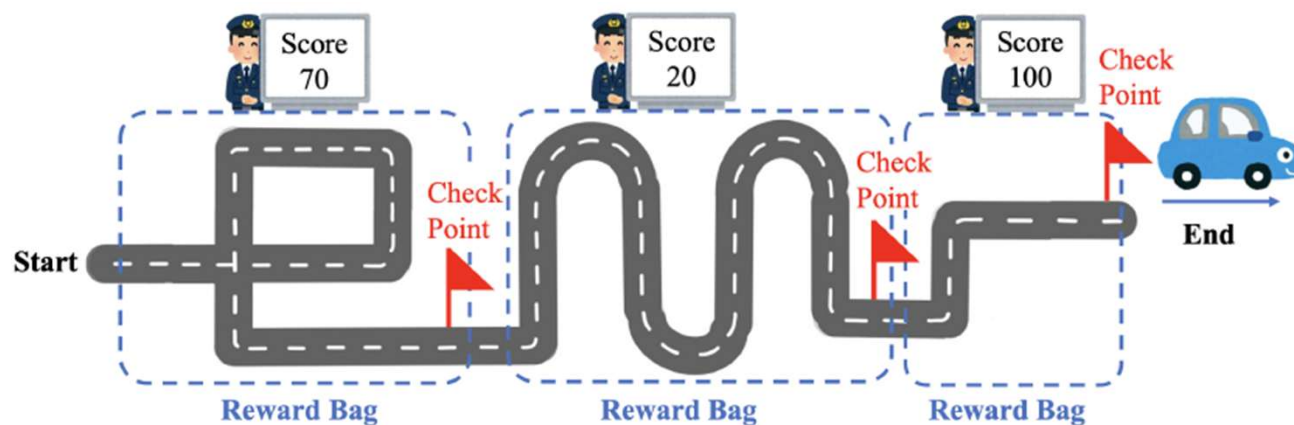


- Noise-robust RL with **symmetric losses**.

$$\ell(z) + \ell(-z) = K \quad \text{Nishimori+ (TMLR2026)}$$

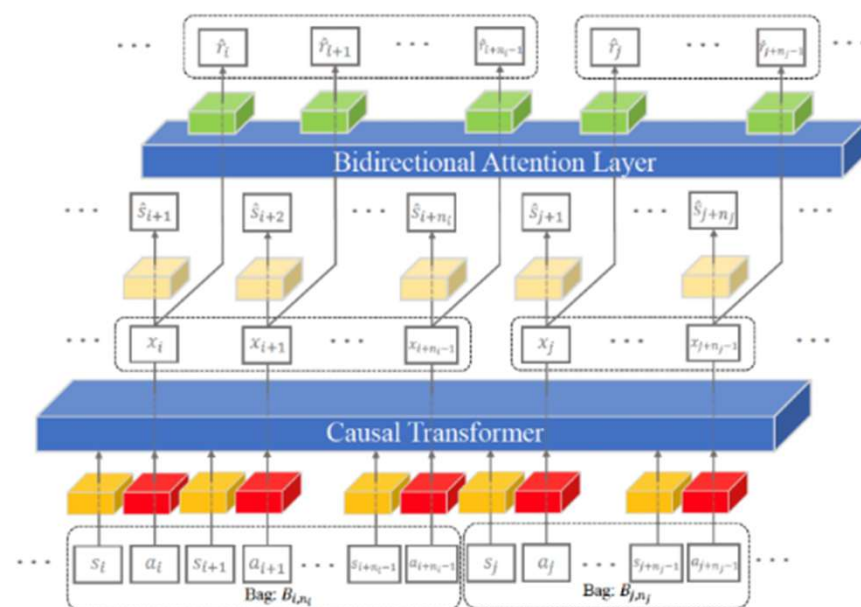
■ Rewards are not necessarily given immediately at each time step:

- A bagged-reward is given at the end of each interval.



■ Reward is redistributed by reward-bag transformer.

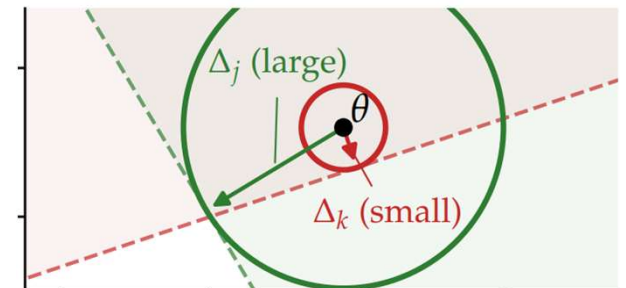
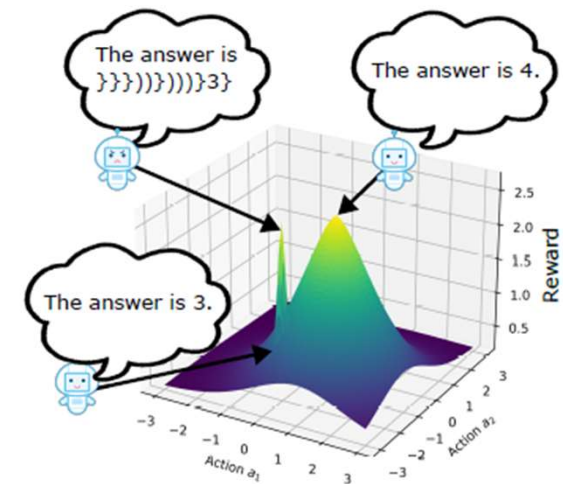
Tang+ (TMLR2025)



- Amodei+ (arXiv2016)

- ## Ackermann+ (ICML2026)

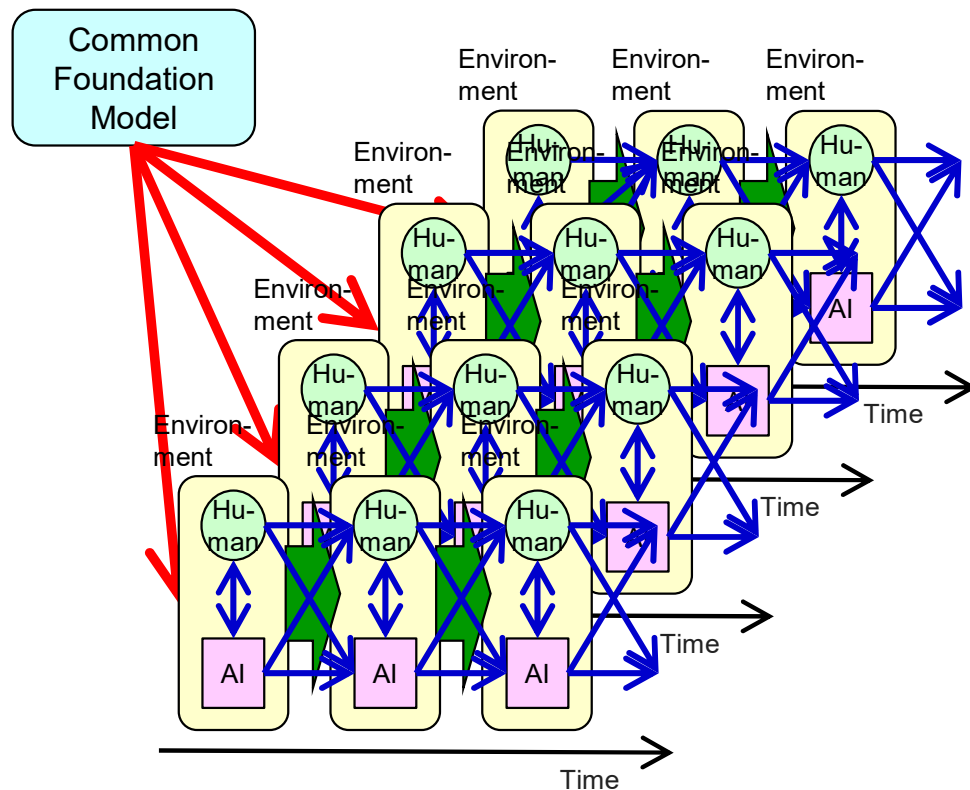
- ## Ono+ (ICML2026workshop)



- **Reward modeling** is the central issue in RL.
- **Various ML methods** can be utilized:
 - Weakly supervised learning,
 - Noise-robust learning,
 - Distribution shift adaptation.
- **Further Challenges:**
 - Generalizing RL to handle (non-scalar) **structured rewards**.
 - Coping with **reward hacking**.

Future Challenges

- Common foundation models are **personalized** over **multiple tasks** and **modalities**.
- Need **a unified theory in dynamic and diverse environments!**



Learning in dynamic environments:

- Online/continual learning
- Imperfect/structured learning
- Sequential decision making
- Probabilistic inference

Learning in diverse environments:

- Multi-agent learning/interaction
- Multi-modal/multi-task learning
- Privacy-preserving learning
- Adversarial learning
- Fairness-aware learning

Computational/energy efficiency:

- Large model/big data learning
- Distributed optimization
- Compressed data learning
- New computing paradigms