Recursive Reward Aggregation

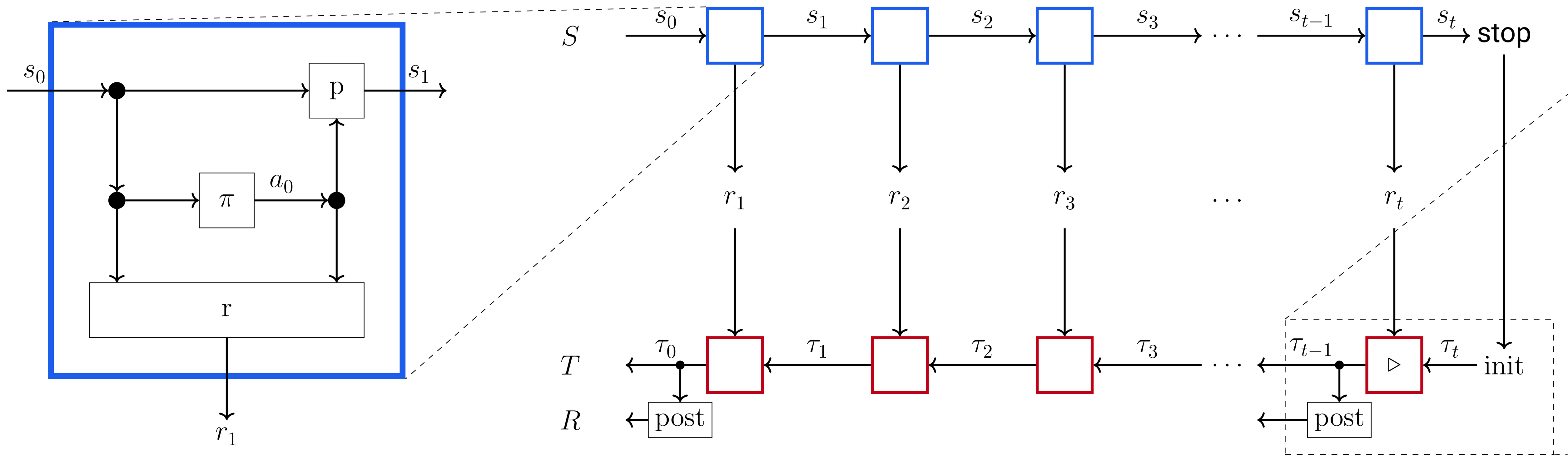
Yuting Tang[1,2] Yivan Zhang[1,2] Johannes Ackermann[1,2]
Yu-Jie Zhang[2] Soichiro Nishimori[1,2] Masashi Sugiyama[2,1]
[1]The University of Tokyo [2]RIKEN AIP

arXiv:2507.08537
https://yivan.xyz/rra

Recursive generation and recursive aggregation lead to generalized Bellman equations.

- states $s \in S$
- actions $a \in A$
- rewards $r \in R$
- statistics $\tau \in T$

- transition function $\mathrm{p} : S \times A \to S$
- policy $\pi : S \to A$
- reward function $\mathrm{r} : S \times A \to R$
- copy $\bullet : x \mapsto (x, x)$



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

Summary

- An **algebra fusion** perspective on the recursive structure of Markov decision processes (MDPs)
- Generalized **Bellman equations** for alternative reward aggregations (e.g., max, mean, variance)
- Integration into *value-based* (e.g., Q-learning) and *actor-critic* (e.g., PPO, TD3) algorithms
- Direct optimization of the *Sharpe ratio* ($\frac{\text{mean}}{\text{std}}$) in portfolio management tasks in finance

Why do we optimize the discounted sum?

- Standard? Convenient? Mathematically elegant?
- Practical benefits? Theoretical guarantees?
- *Reward hypothesis*? If true, all preferences can be represented in this way.

But... does it always align with our requirements?

- **Max**: peak performance in drug discovery
- **Min**: bottleneck objective in network routing
- **Variance**-based criteria: risk-aversion in finance, robotics, and control

We want to optimize other reward aggregations, *directly*, *efficiently*, and *effectively*.

Discounted sum is a *recursive function*

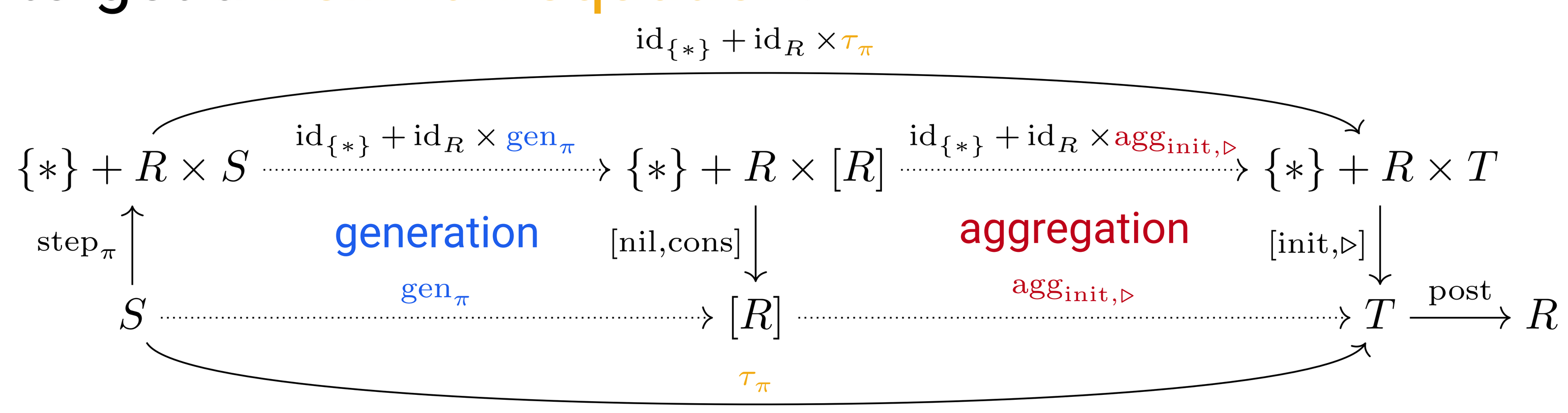
... and we can use other recursive aggregations.

$$\begin{aligned}
\mathrm{sum}_\gamma[r_1, r_2, r_3, \dots] &= r_1 + \gamma r_2 + \gamma^2 r_3 + \dots \\
&= r_1 + \gamma(r_2 + \gamma r_3 + \dots) \\
&= r_1 + \gamma\, \mathrm{sum}_\gamma[r_2, r_3, \dots] \\
\big\downarrow \\
\mathrm{agg}_{\mathrm{init},\triangleright}[r_1, r_2, r_3, \dots] &:= r_1 \triangleright \mathrm{agg}_{\mathrm{init},\triangleright}[r_2, r_3, \dots]
\end{aligned}$$

- **initial value** $\mathrm{init} \in T$ of statistics ($\mathrm{agg}_{\mathrm{init},\triangleright}[\,] = \mathrm{init}$)
- **update function** $\triangleright : R \times T \to T$
- **statistic aggregation function** $\mathrm{agg}_{\mathrm{init},\triangleright} : [R] \to T$

Reward generation is also a *recursive function*

... so we can "*fuse*" it with a recursive aggregation to get a Bellman equation.

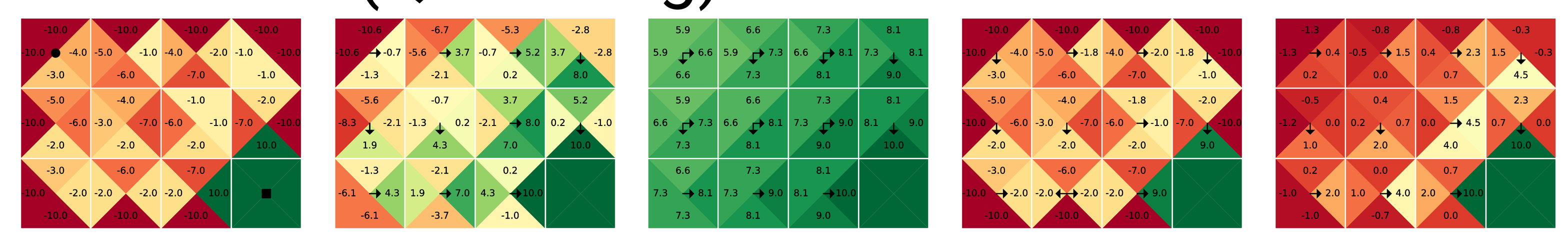


Theorem (Bellman equation for statistic function)

$$\tau_\pi(s_t) = \begin{cases} \mathrm{init} & s_t \text{ is terminal} \\ r_{t+1} \triangleright \tau_\pi(s_{t+1}) & s_t \text{ is non-terminal} \end{cases}$$

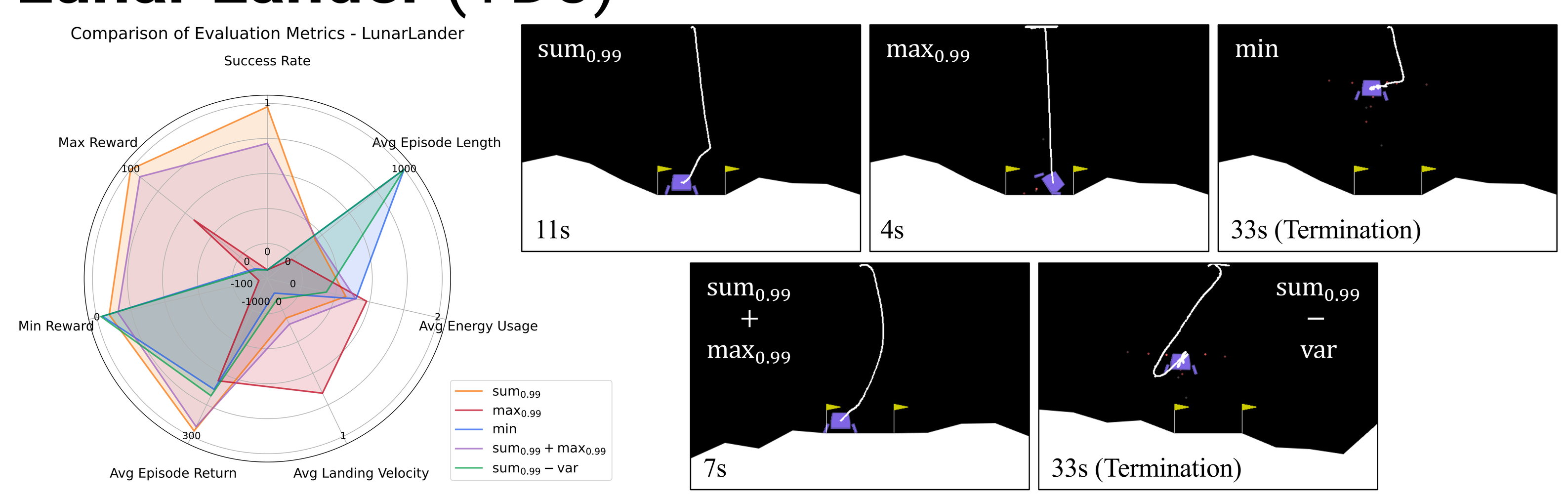
Experiments

Grid-world (Q-learning)



Grid $\mathrm{sum}_{0.9}$ $\max_{0.9}$ $\min_{0.9}$ mean

Lunar Lander (TD3)



Future work

- Multi-dimensional or non-numerical feedback?
- Agent states? **Automata as aggregators**?
- List $\to$ **tree**, list function $\to$ **tree traversal**?
- Logic, reasoning, safety, and alignment?

References

Wei Cui and Wei Yu. Reinforcement learning with non-cumulative objective. *Machine Learning in Communications and Networking*, 2023.
Ralf Hinze, Thomas Harper, and Daniel W. H. James. Theory and practice of fusion. In *Symposium on Implementation and Application of Functional Languages*, 2010.
William F. Sharpe. Mutual fund performance. *The Journal of Business*, 1966.