Reinforcement Learning
of Marked Temporal Point Processes

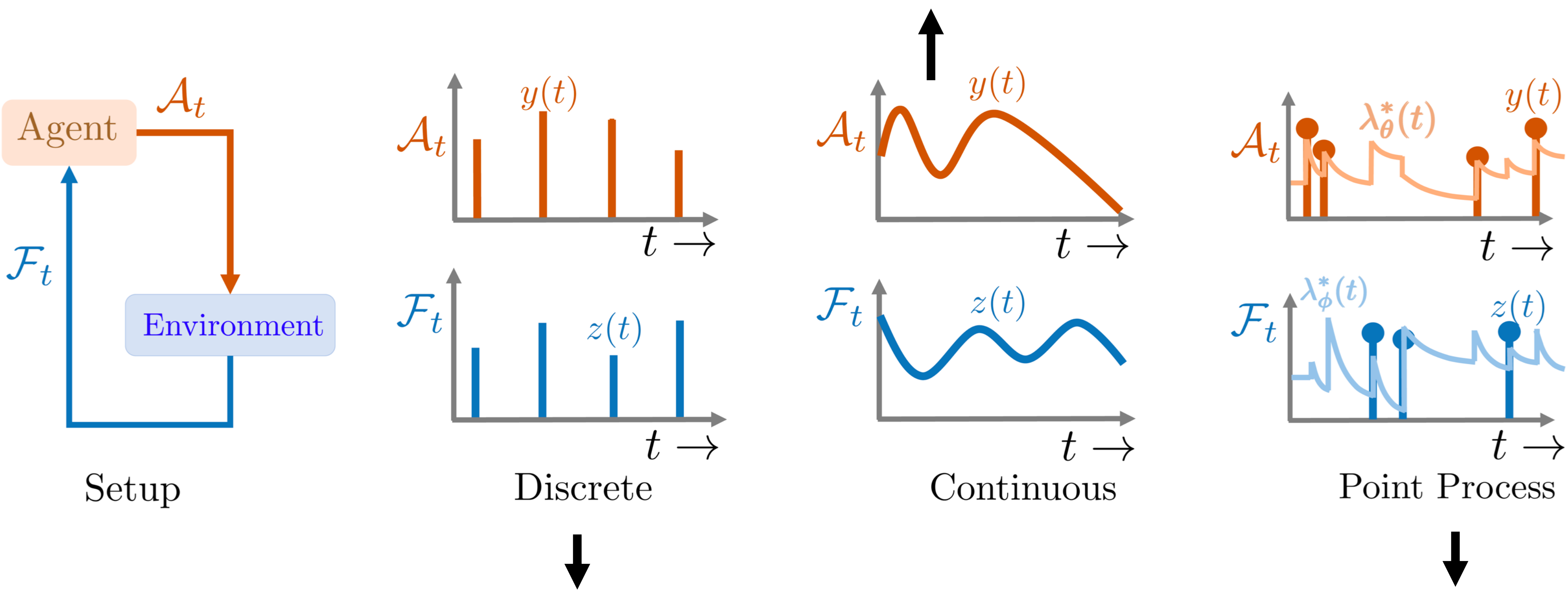
Human-centered Machine Learning

http://courses.mpi-sws.org/hcml-ws18/

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Reinforcement learning on different settings

Actions and feedback are real-valued functions in continuous time



Actions and feedback occur in discrete time

Actions and feedback are asynchronous events localized in continuous time

Reinforcement learning of marked TPP

If the problem dynamics cannot be expressed using SDEs with jumps or the objective is intractable:

➡ **Reinforcement learning of marked temporal point processes**

→ **Policy gradient** [Upadhyay, 2018]

→ **Policy iteration** [Farajtabar et al., 2017]

Similarly as with optimal control:

Policy is characterized by an intensity function!

Reinforcement learning of marked TPP

If the problem dynamics cannot be expressed using SDEs with jumps or the objective is intractable:

poral

Next, details on the approach based on policy gradient

Policy iteration [Farajtabar et al., 2017]

Similarly as with optimal control:

Policy is characterized by an intensity function!

Viral marketing

Agent

Social media user

Environment

Followers' Feed

Forbes

For Brands And PR: When Is The Best Time To Post On Social Media?

THE HUFFINGTON POST

The Best Times to Post on Social Media

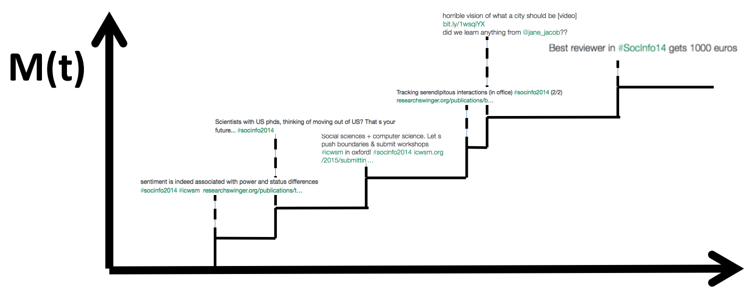
When to post to maximize views or likes?

$\mu_i(t) = u(t) \longrightarrow N_i(t)$

Design (optimal) posting intensity

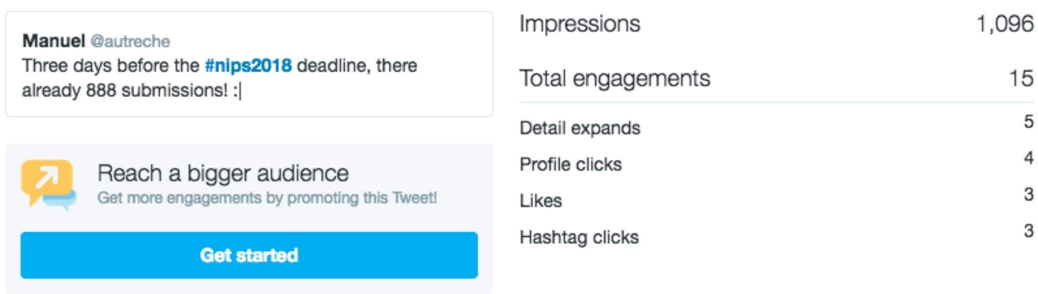
Marks (feedback) given by environment

Visibility dynamics are unknown



Proprietary algorithmic ranking

However, one may have access to quality metrics

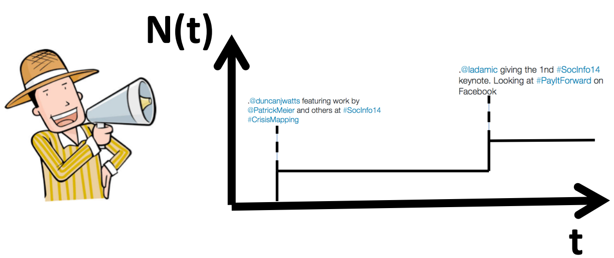


Key idea:

Think of these metrics as rewards in a reinforcement learning setting!

[Upadhyay et al., 2018]

Broadcasters and feedback



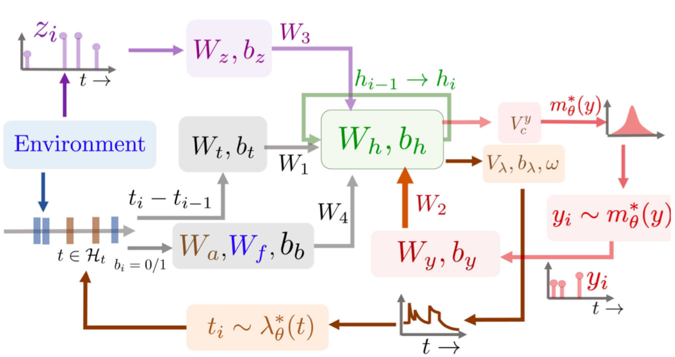
$$p^*_{\mathcal{A};\theta} = (\lambda^*_\theta, m^*_\theta)$$

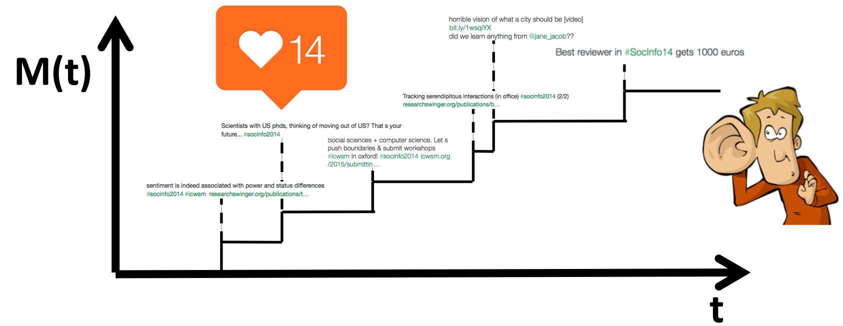
Policy

Intensity

Mark distribution

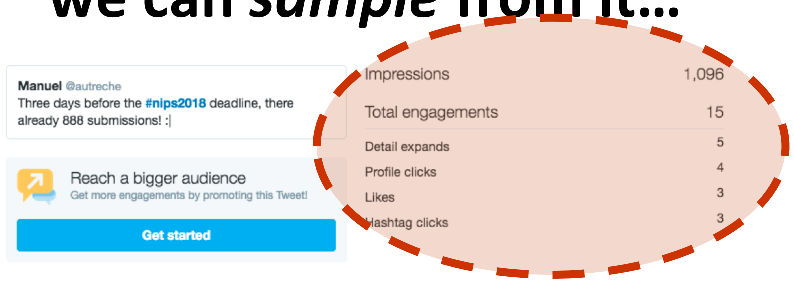
Parametrized using RNNs





$$p^*_{\mathcal{F};\phi} = (\lambda^*_\phi, m^*_\phi)$$

We do not know the *feedback* distribution but we can *sample* from it…



…and measure quality metrics (rewards)

[Upadhyay et al., 2018]

What is the goal in reinforcement learning?

We aim to maximize the average reward in a time window [0, T]:

$$\underset{p^*_{\mathcal{A};\theta}(\cdot)}{\text{maximize}} \quad \overbrace{\mathbb{E}_{\underbrace{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)}_{\text{Actions and environment are asynchronous!}}} \underbrace{[R^*(T)]}_{\text{Reward (Cumulative)}}}^{J(\theta)}$$

Connection to optimal control:

$$J(r(t), \lambda(t), t) = \min_{u(t,t_f]} \mathbb{E}_{(N,M)(t,t_f]} \left[\phi(r(t_f)) + \int_t^{t_f} \ell(r(\tau), u(\tau)) \, d\tau \right]$$

[Upadhyay et al., 2018]

Policy gradient

We use gradient descent to improve the policy, i.e., the intensity, over time:

$$\theta_{l+1} = \theta_l + \alpha_l \nabla_\theta J(\theta)|_{\theta=\theta_l}$$

We need to compute the gradient of an average. But the average depends on the parameters!

$$\nabla_\theta J(\theta) = \nabla_\theta \; \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} [R^*(T)]$$

Parameters!

[Upadhyay et al., 2018]

Reinforce trick to compute gradient

The reinforce trick allows us to overcome this implicit dependence:

$$\nabla_\theta J(\theta) = \nabla_\theta \; \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} \left[R^*(T) \right]$$

Appendix A in Upadhyay et al., 2018

$$\nabla_\theta J(\theta) = \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} \left[R^*(T) \nabla_\theta \log \mathbb{P}_\theta(\mathcal{A}_T) \right]$$

Parameters!

[Upadhyay et al., 2018]

Likelihood of action events

$$\nabla_\theta J(\theta) = \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} \left[R^*(T) \nabla_\theta \log \mathbb{P}_\theta(\mathcal{A}_T) \right]$$

Likelihood of posts by our broadcaster!

$$\mathbb{P}(\mathcal{A}_T) := \left(\prod_{e_i \in \mathcal{A}_T} \lambda^*_\theta(t_i) \right) \exp\left(-\int_0^T \lambda^*_\theta(s)\, ds \right)$$

The key remaining question is how to parametrize the intensity $\lambda^*_\theta(t)$

Parameters & functional form!

[Upadhyay et al., 2018]

Policy parametrization

Parameters

Output layer:

Last time the broadcaster posted

$$\lambda^*_\theta(t) = \exp\left(b_\lambda + w_t(t - t') + \boldsymbol{V}_\lambda \boldsymbol{h}_i\right)$$

Hidden layer:

$$\boldsymbol{h}_i = \tanh(\boldsymbol{W}_h \boldsymbol{h}_{i-1} + \boldsymbol{W}_1 \boldsymbol{\tau}_i + \boldsymbol{W}_4 \mathbf{b}_i + \boldsymbol{b}_h)$$

Input layer:

$e_i = 0$ action

$e_i = 1$ feedback

$$\boldsymbol{\tau}_i = \boldsymbol{W}_t(t_i - t_{i-1}) + \boldsymbol{b}_t \qquad \mathbf{b}_i = \boldsymbol{W}_a(1 - e_i) + \boldsymbol{W}_f e_i + \boldsymbol{b}_b$$

i-th event (i-1)-th event

[Upadhyay et al., 2018]

Sampling from the policy

$$\lambda^*_\theta(t) = \exp\left(b_\lambda + w_t(t - t') + \boldsymbol{V}_\lambda \boldsymbol{h}_i\right)$$

The intensity can increase or decrease every time an event by the other broadcasters take place:

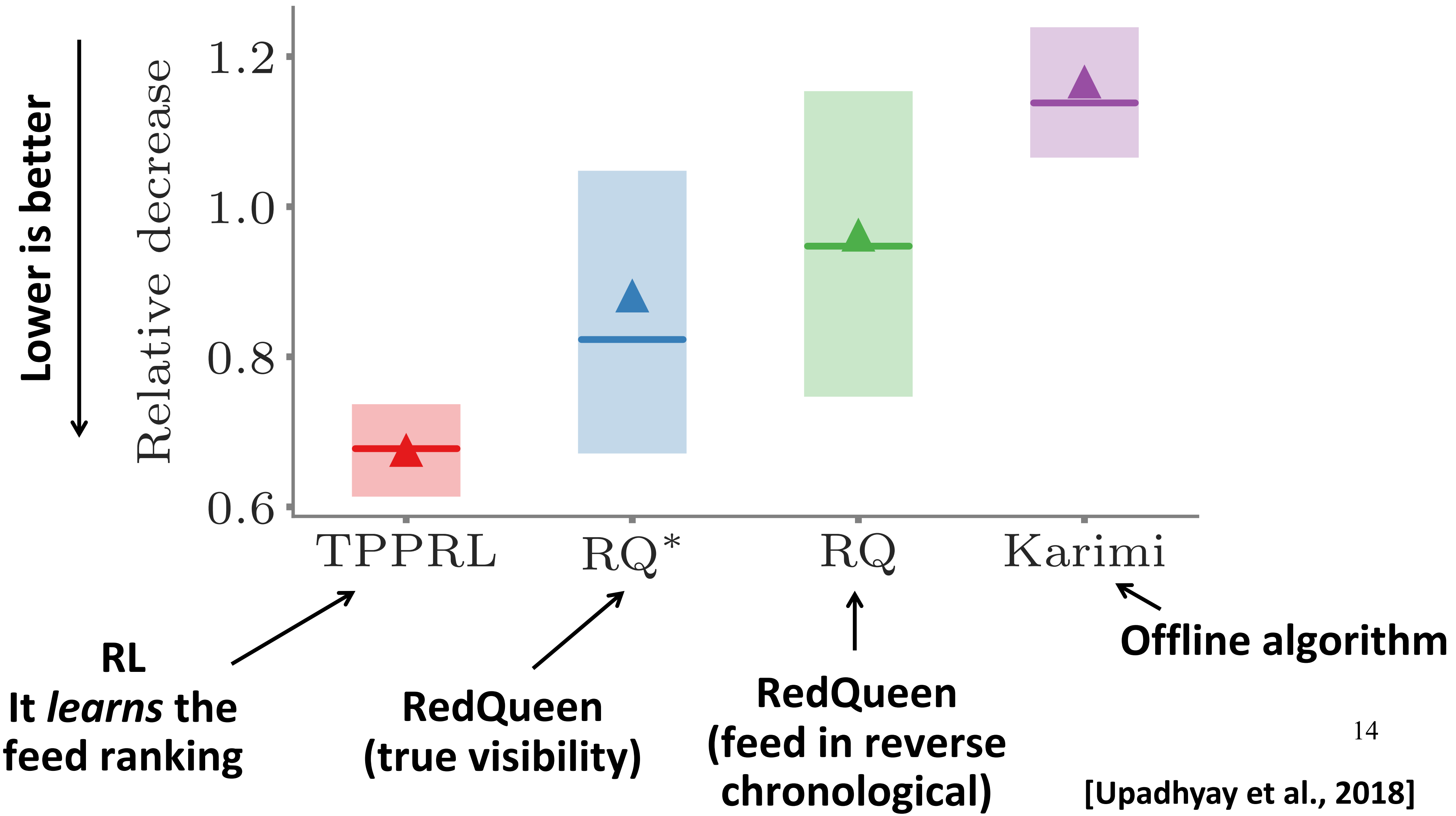
- → **We cannot apply just superposition**
- → **We can use inversion sampling: The CDF is a function by parts, where each part is defined once an event by the other broadcasters happens**

Appendix C in Upadhyay et al., 2018

[Upadhyay et al., 2018]

Average Rank

$$R^*(T) = \int_0^T r(t)dt$$

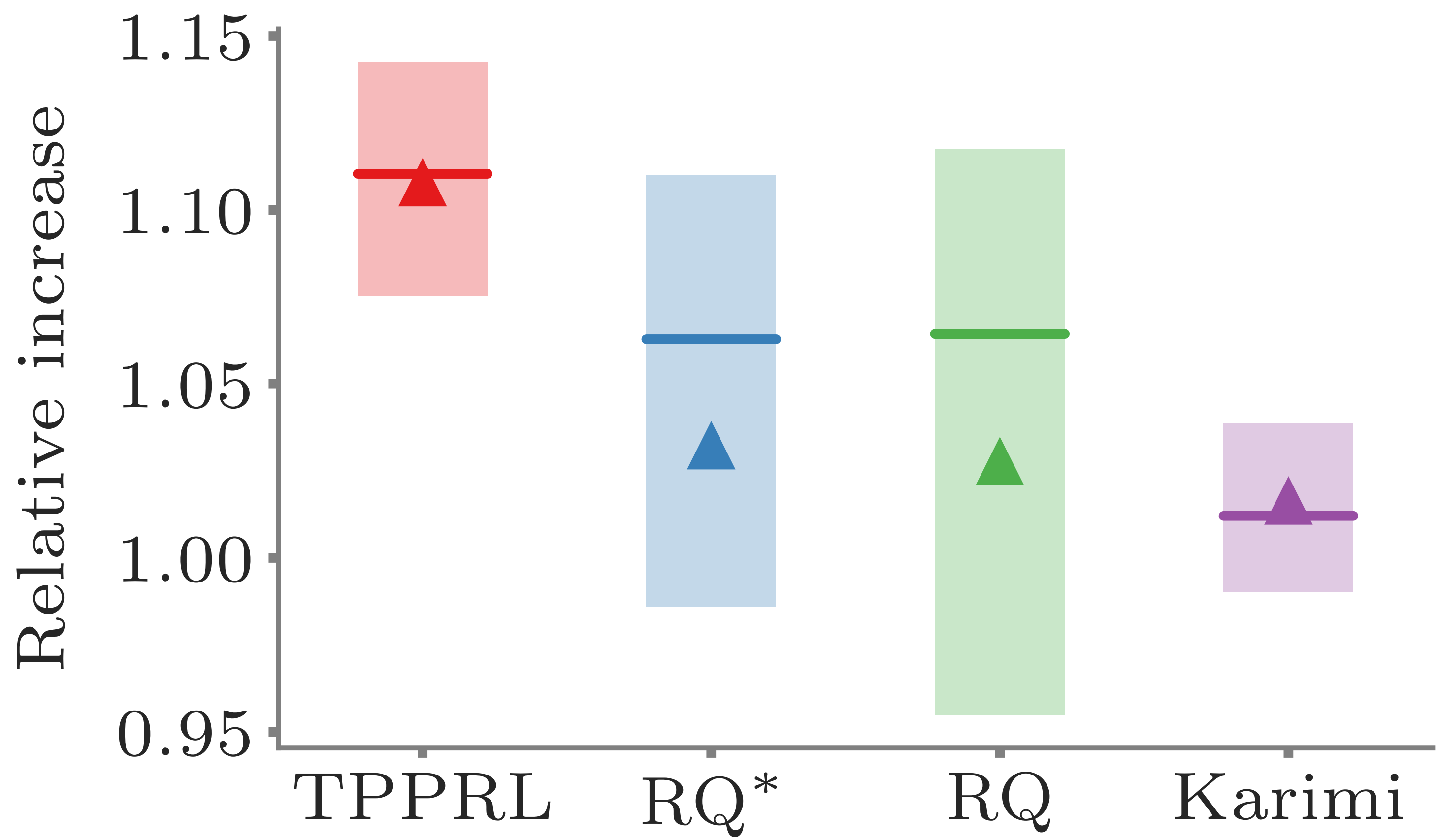


[Upadhyay et al., 2018]

Time at the top

$$R^*(T) = \int_0^T \mathbb{I}(r(t) < 1)dt$$

Higher is better



RL
It *learns* the feed ranking

RedQueen (true visibility)

RedQueen (feed in reverse chronological)

Offline algorithm

[Upadhyay et al., 2018]