

# **Thompson Sampling in Adversarial Environments**

# Multi-Armed Bandit Problem

## With Full Information

For each timestep  $t = 1, 2, \dots, T$

1. Forecaster chooses action  $a_t$
2. Observe loss  $\ell_{i,t}$  for all actions  $i \in \{1, \dots, N\}$
3. Forecaster suffers loss  $\ell_{a_t,t}$

# Multi-Armed Bandit Problem

**With Full Information**

For each timestep  $t = 1, 2, \dots, T$

1. Forecaster chooses action  $a_t$
2. Observe loss  $\ell_{i,t}$  for all actions  $i \in \{1, \dots, N\}$
3. Forecaster suffers loss  $\ell_{a_t,t}$

**application: A/B Testing**

# Multi-Armed Bandit Problem

**With Full Information**

For each timestep  $t = 1, 2, \dots, T$

1. Forecaster chooses action  $a_t$
2. Observe loss  $\ell_{i,t}$  for all actions  $i \in \{1, \dots, N\}$
3. Forecaster suffers loss  $\ell_{a_t,t}$

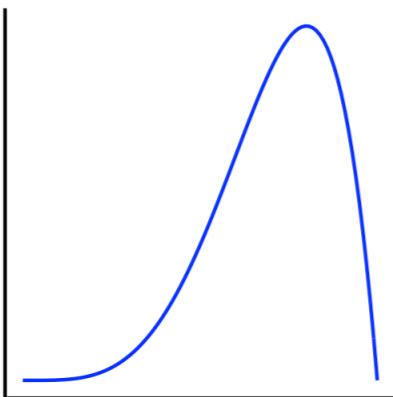
**application: A/B Testing**

**goal: minimize regret**

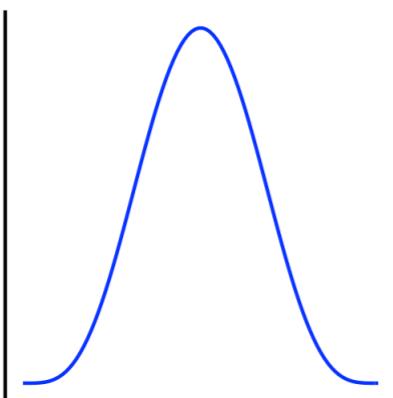
$$R_T = \hat{L}_T - L_T^*$$

# Action

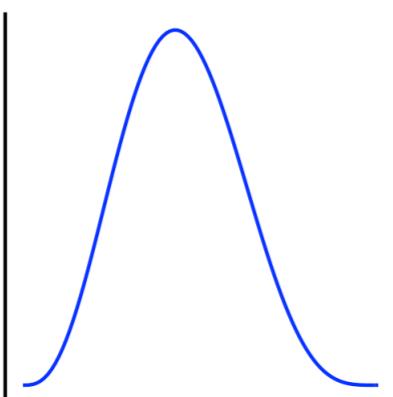
**A**



**B**

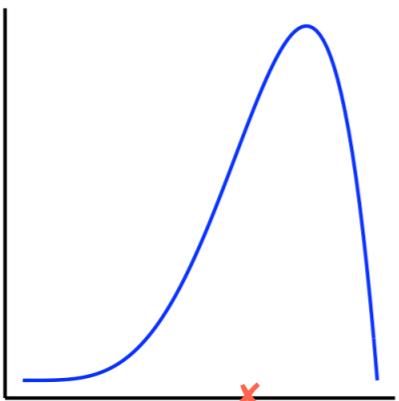


**C**

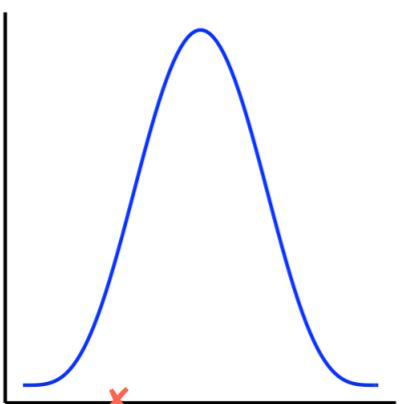


# Action

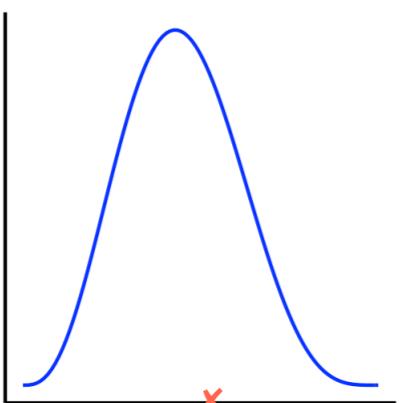
**A**



**B**

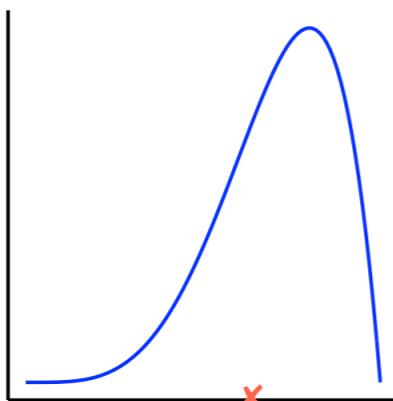


**C**

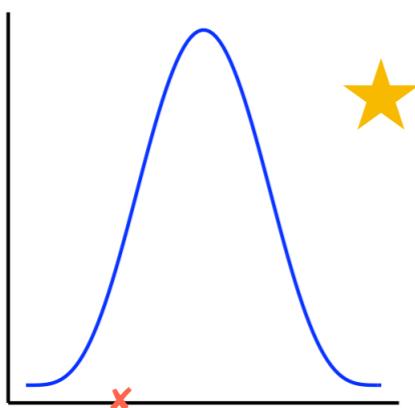


# Action

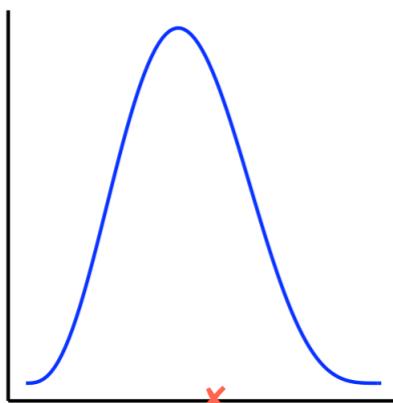
A



B



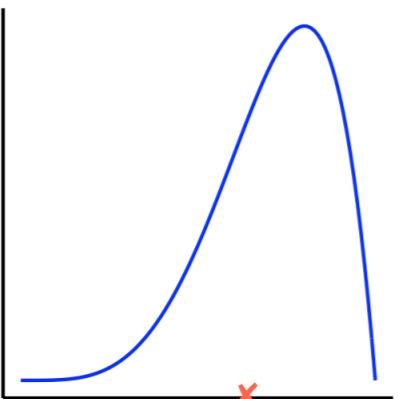
C



**Action**

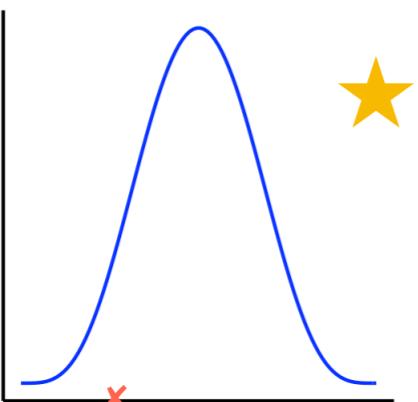
**Loss**

**A**



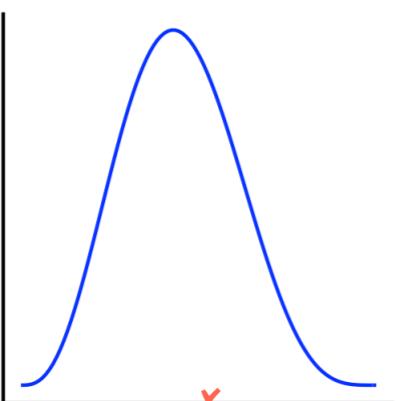
**0.7**

**B**



**0.9**

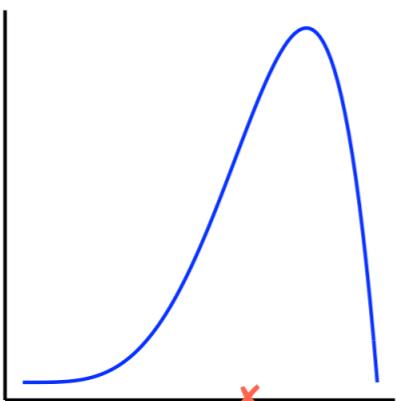
**C**



**0.2**

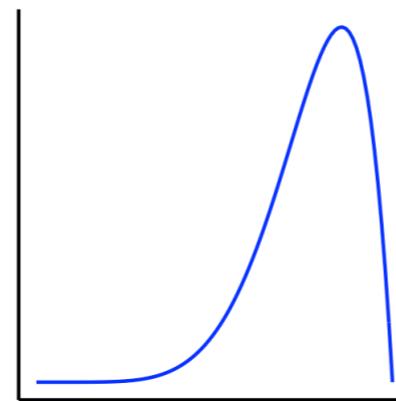
## Action

A

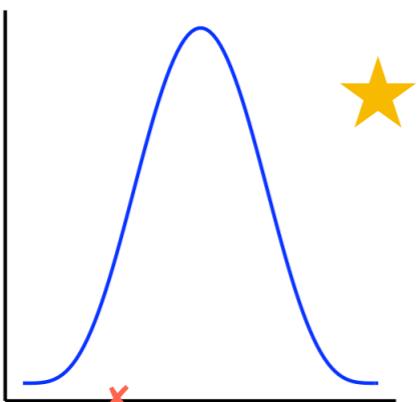


## Loss

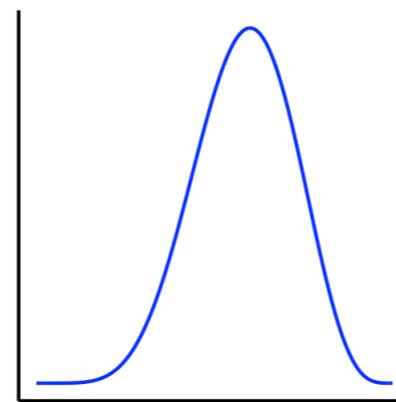
0.7



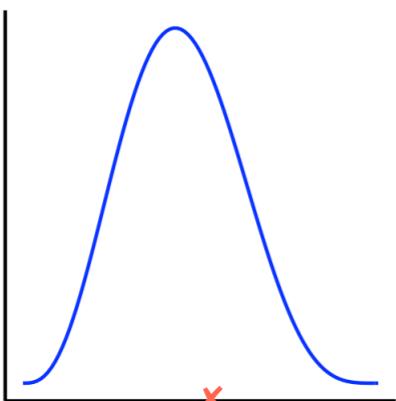
B



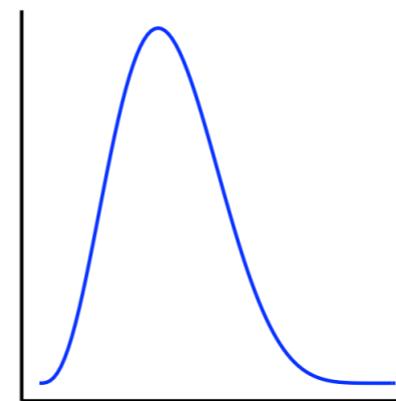
0.9



C

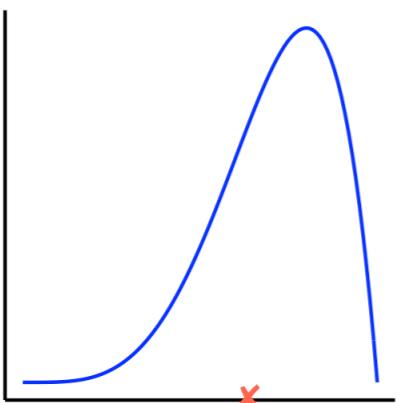


0.2



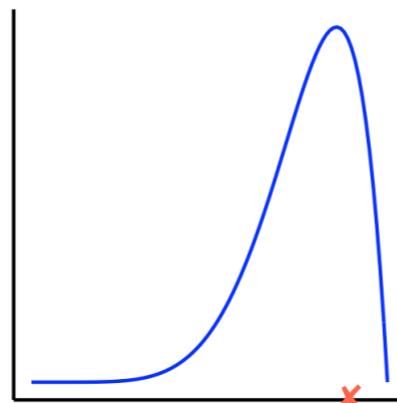
## Action

A

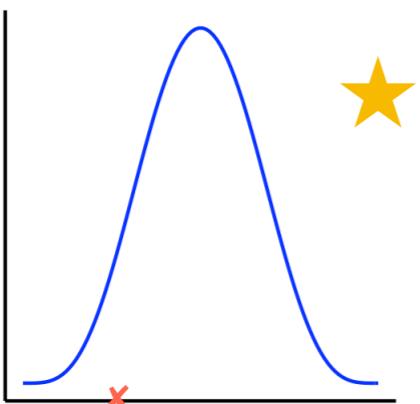


## Loss

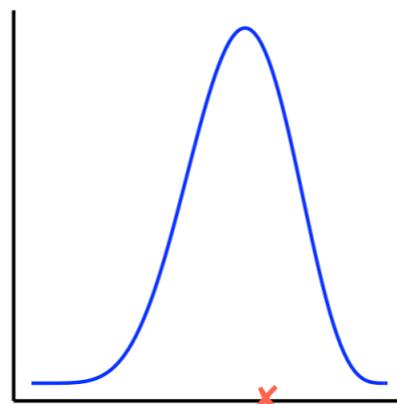
0.7



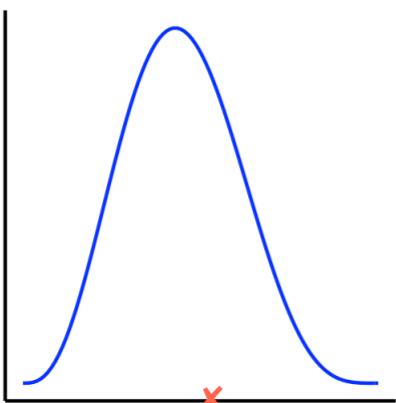
B



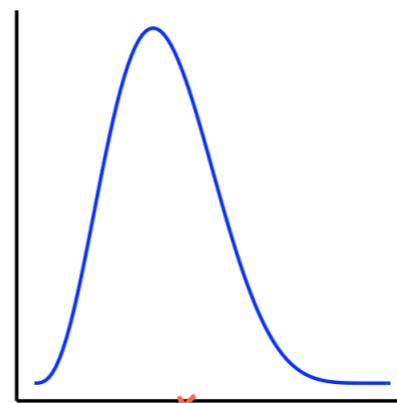
0.9



C

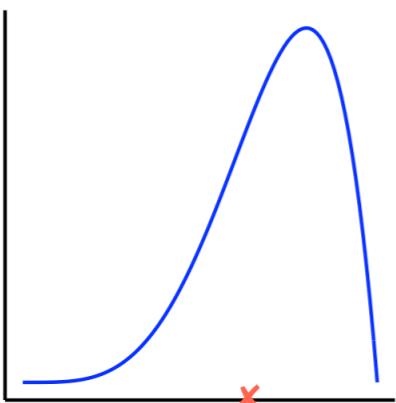


0.2



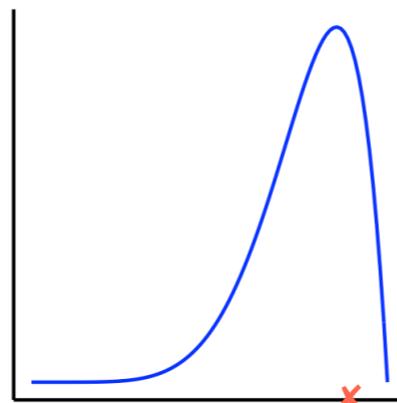
## Action

A

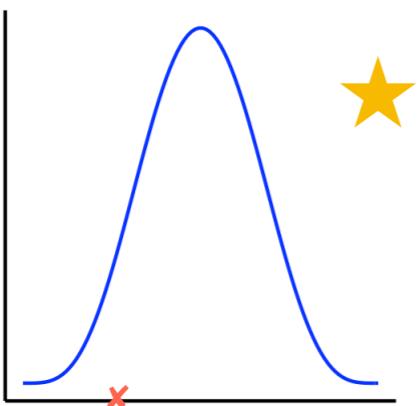


## Loss

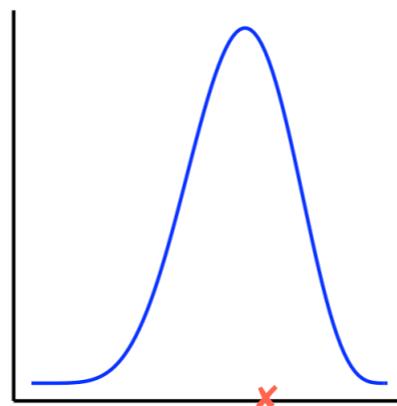
0.7



B

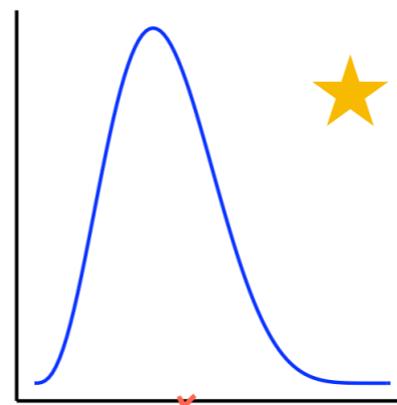
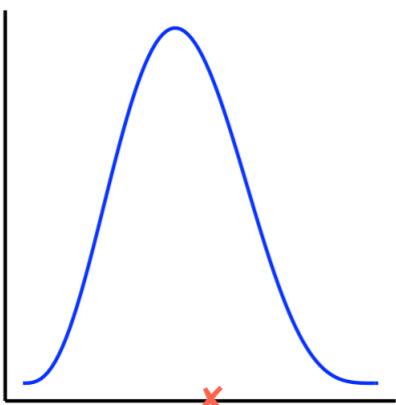


0.9



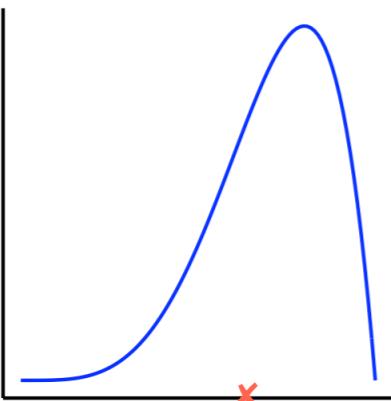
C

0.2



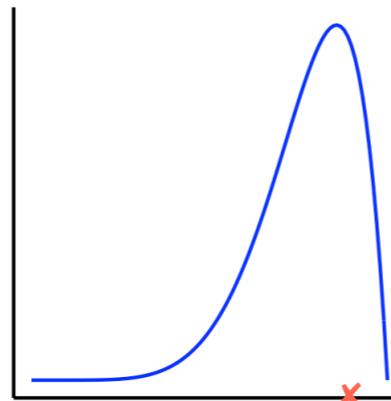
## Action

A



## Loss

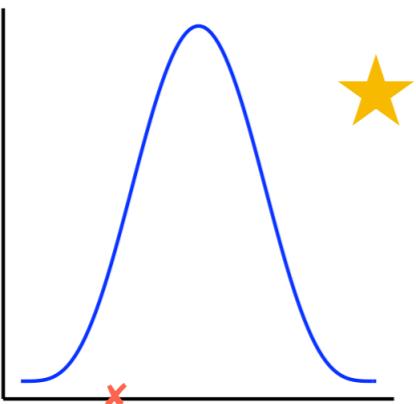
0.7



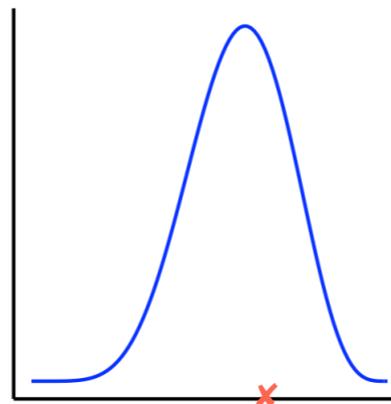
## Loss

0.8

B

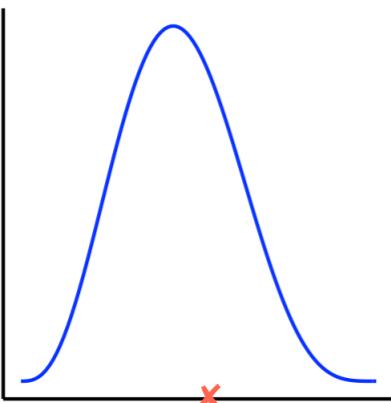


0.9

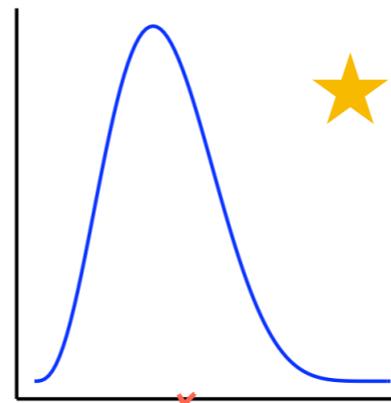


0.7

C



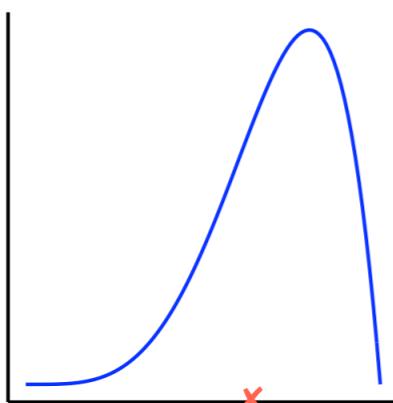
0.2



0.3

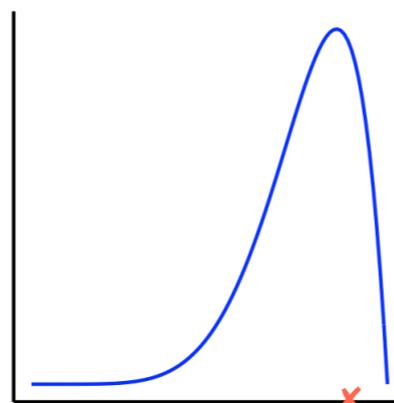
## Action

A



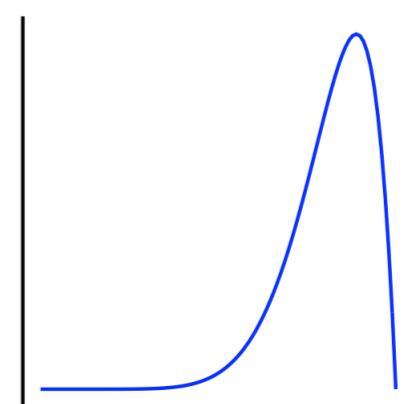
## Loss

0.7

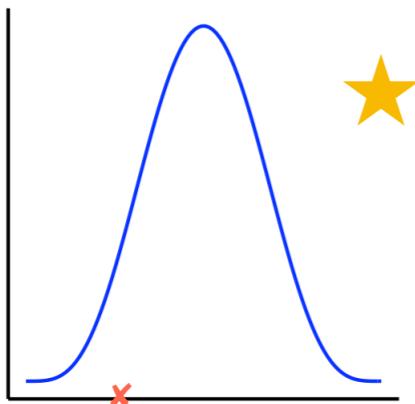


## Loss

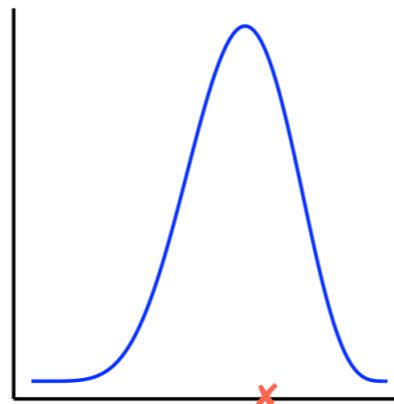
0.8



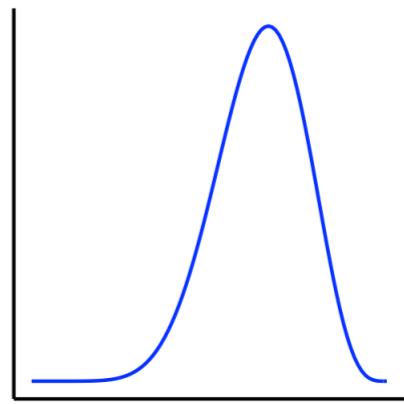
B



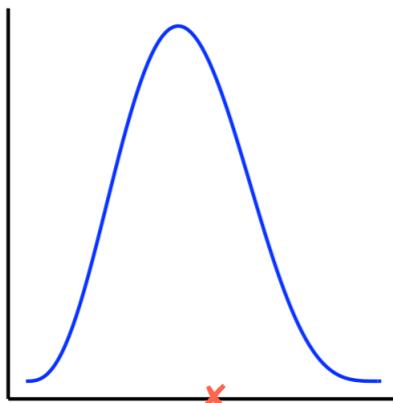
0.9



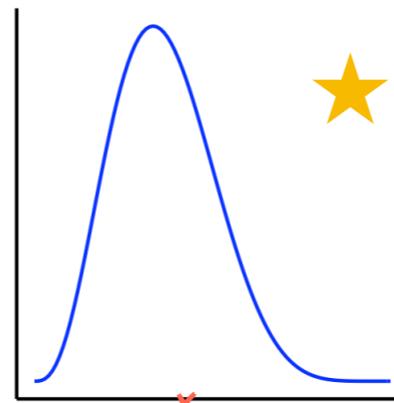
0.7



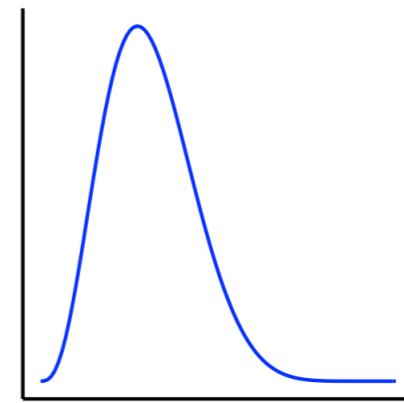
C



0.2

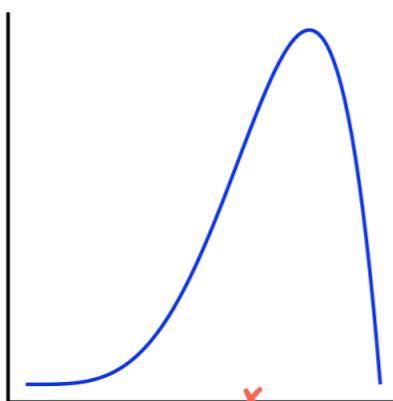


0.3



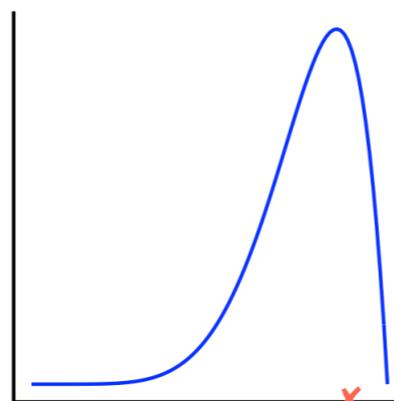
## Action

A



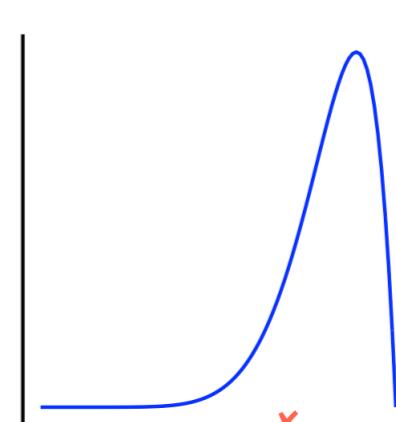
## Loss

0.7

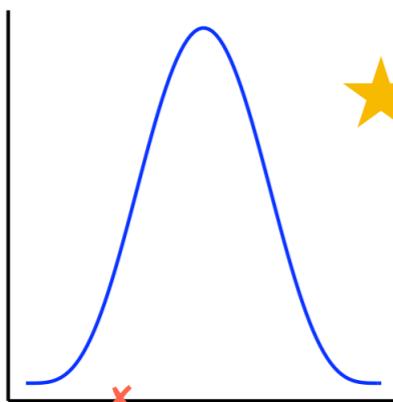


## Loss

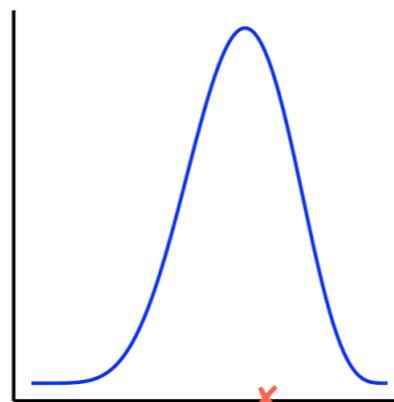
0.8



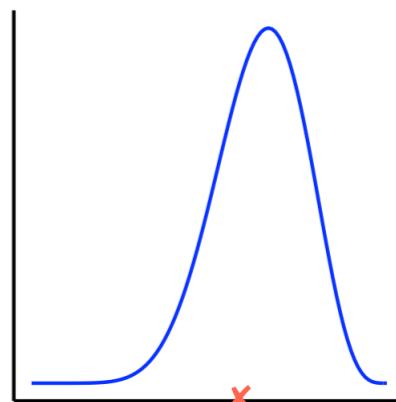
B



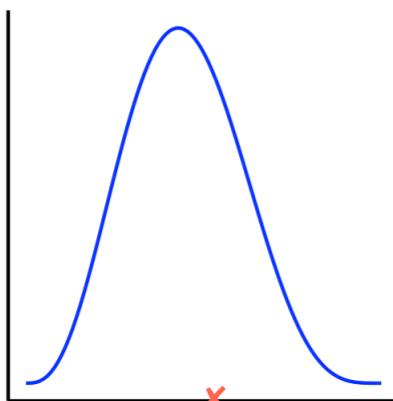
0.9



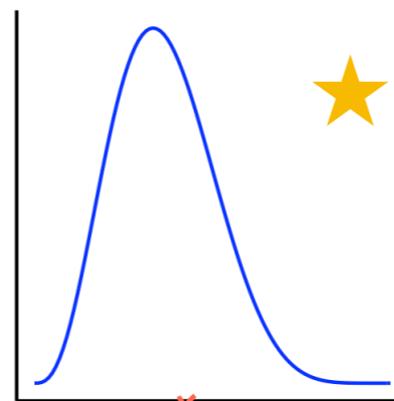
0.7



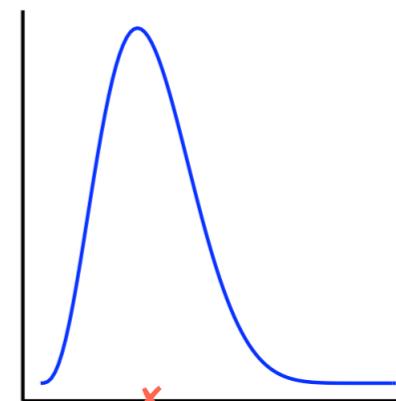
C



0.2

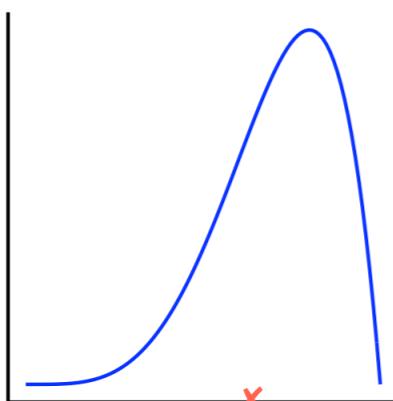


0.3



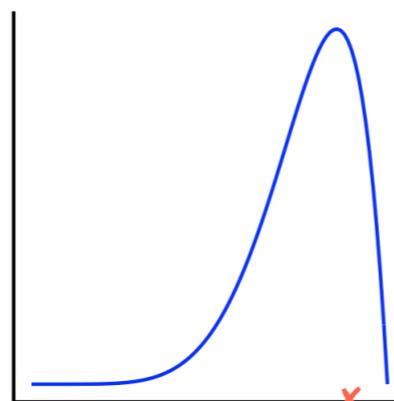
## Action

A



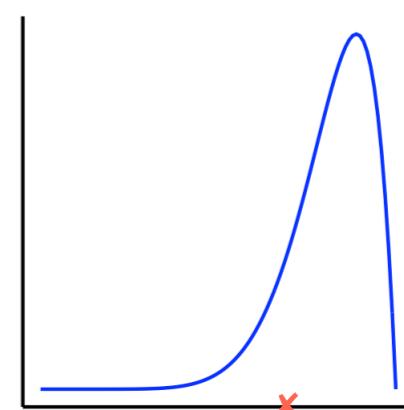
## Loss

0.7

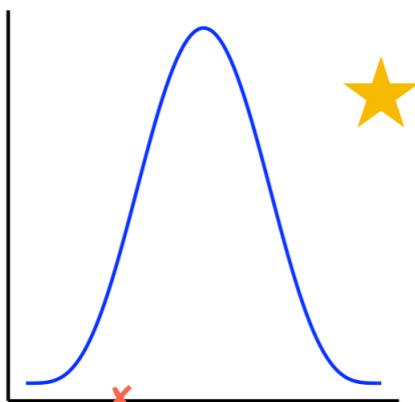


## Loss

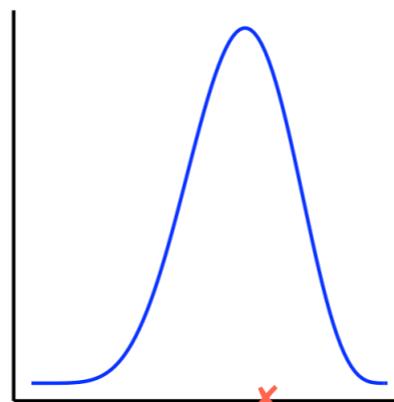
0.8



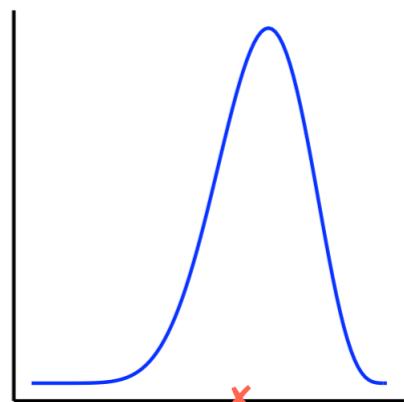
B



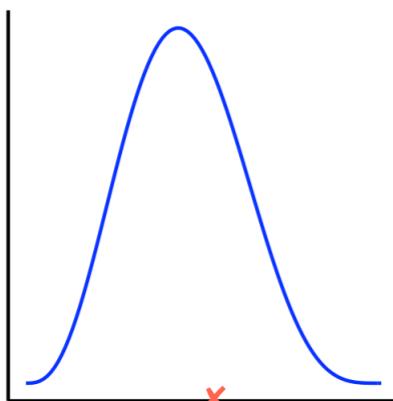
0.9



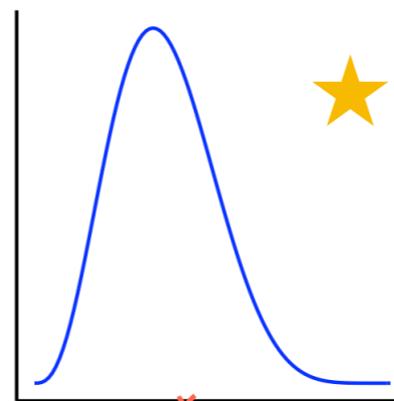
0.7



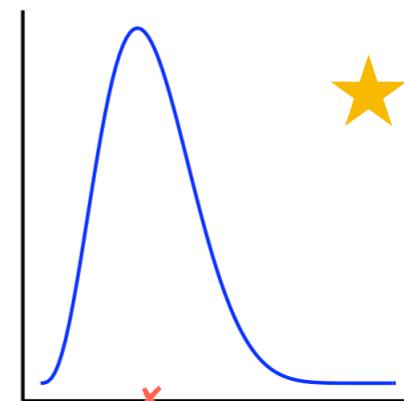
C



0.2

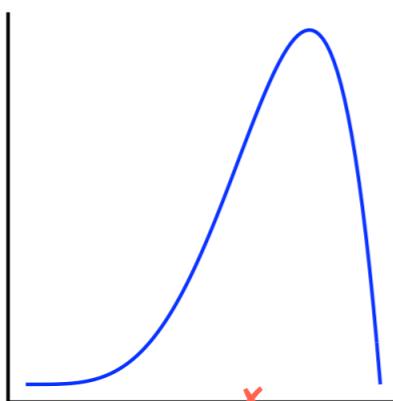


0.3



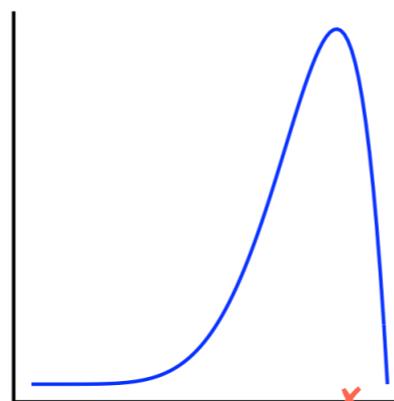
**Action**

**A**



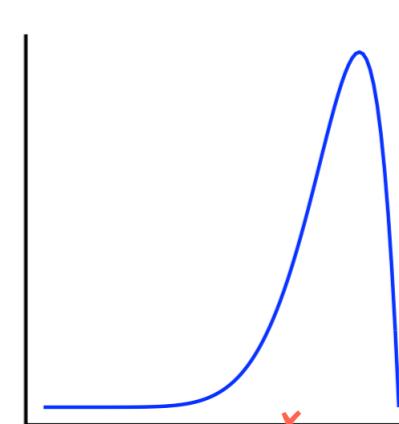
**Loss**

**0.7**



**Loss**

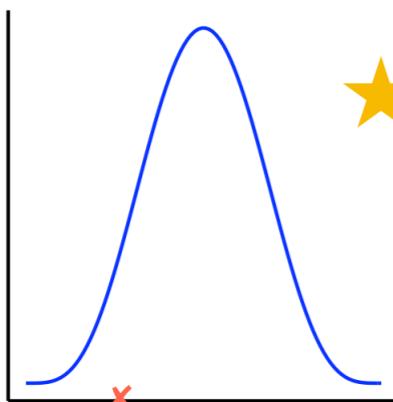
**0.8**



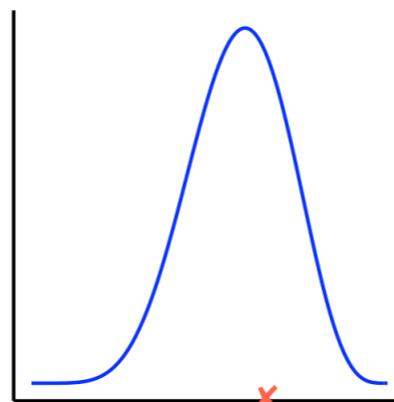
**Loss**

**0.7**

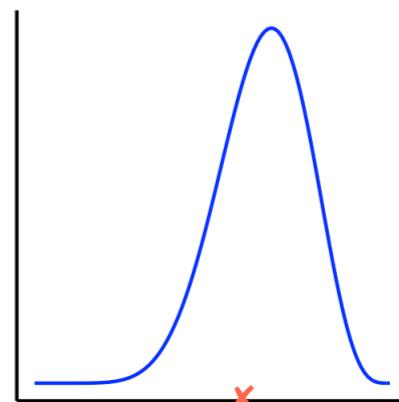
**B**



**0.9**

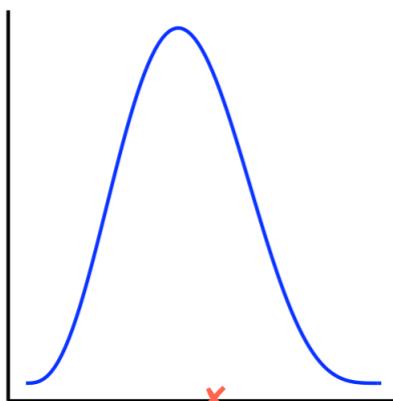


**0.7**

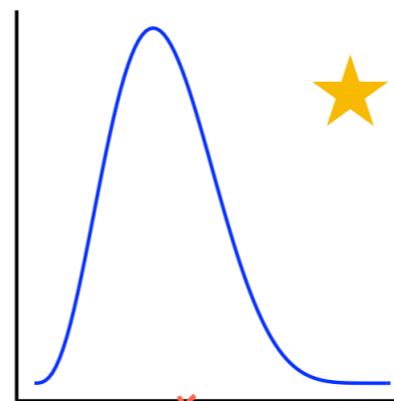


**0.2**

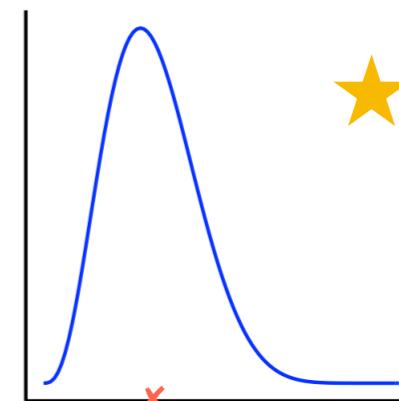
**C**



**0.2**



**0.3**



**0.4**

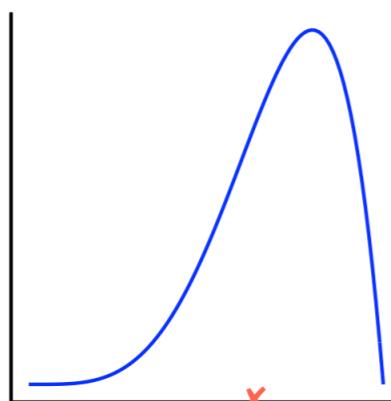
**★**

**★**

**★**

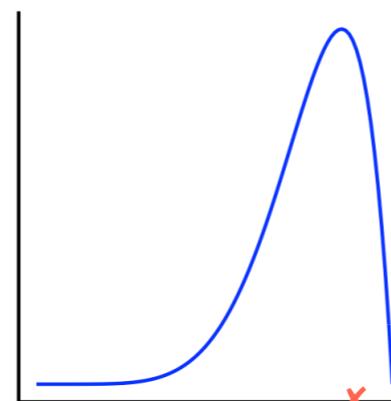
**Action**

**A**



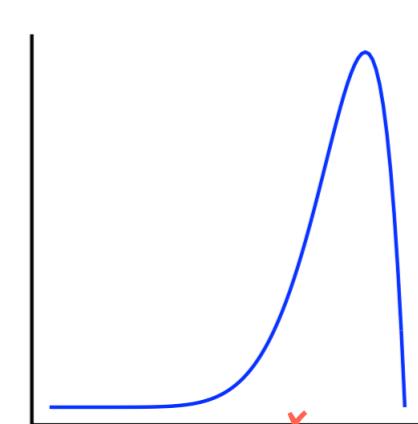
**Loss**

**0.7**



**Loss**

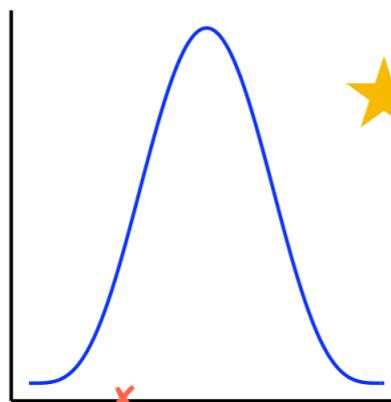
**0.8**



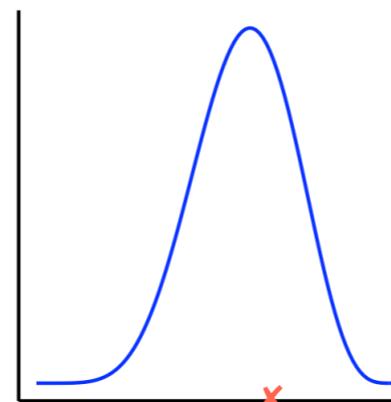
**Loss**

**0.7**

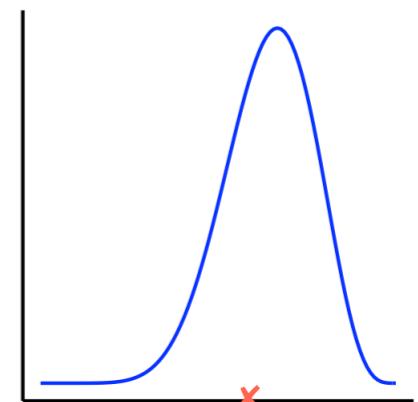
**B**



**0.9**

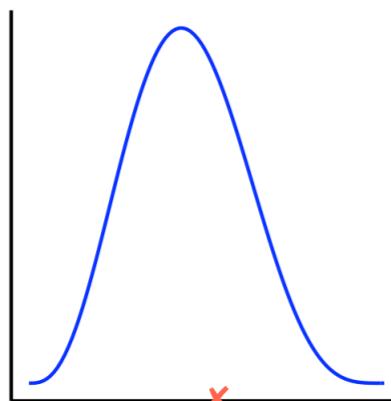


**0.7**

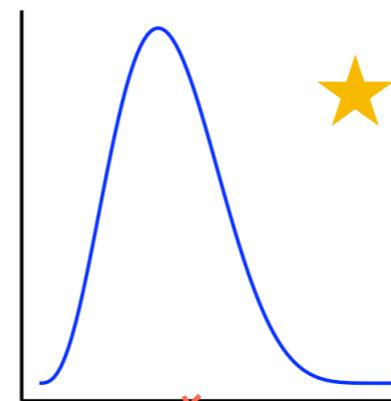


**0.2**

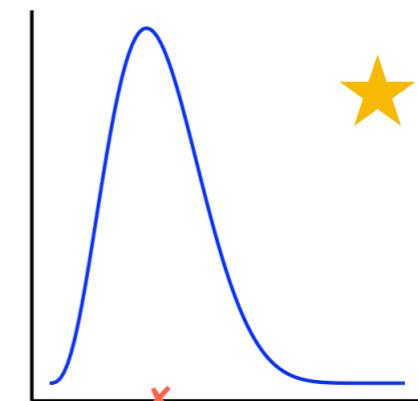
**C**



**0.2**



**0.3**

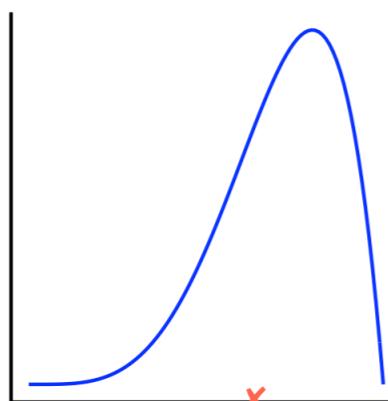


**0.4**

$$\text{Total Loss} = 0.9 + 0.3 + 0.4 = 1.6$$

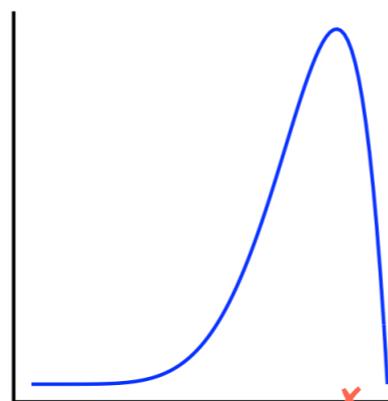
Action

A



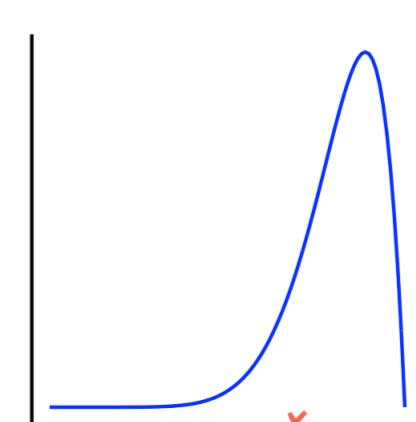
Loss

0.7



Loss

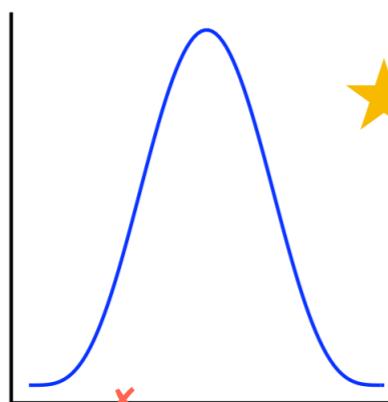
0.8



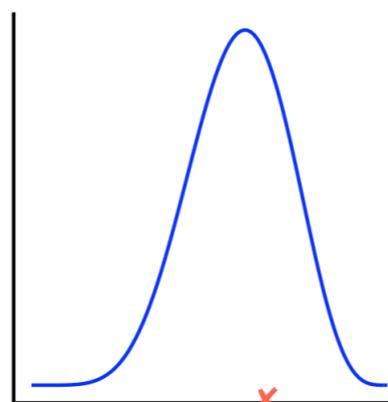
Loss

0.7

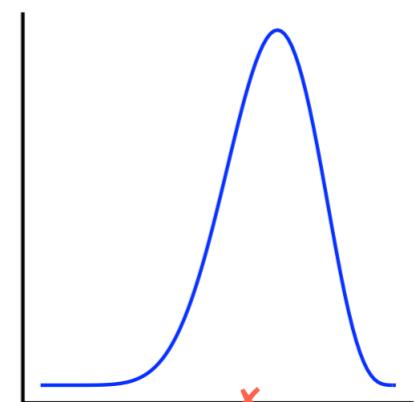
B



0.9

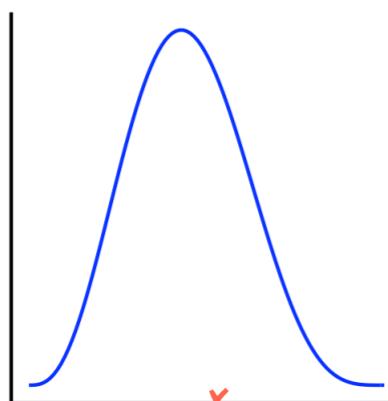


0.7

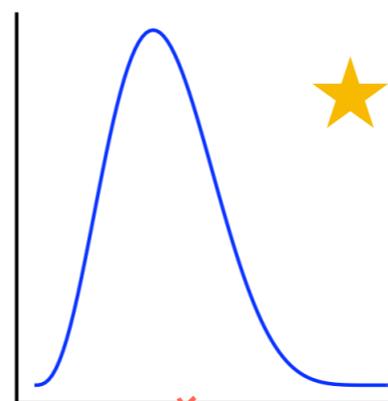


0.2

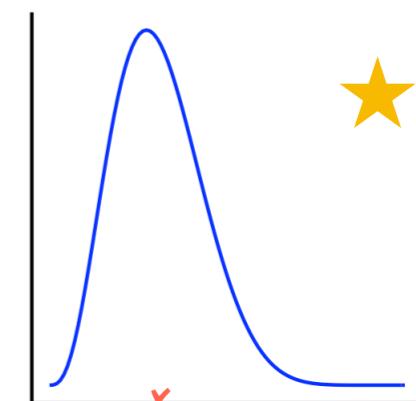
C



0.2



0.3



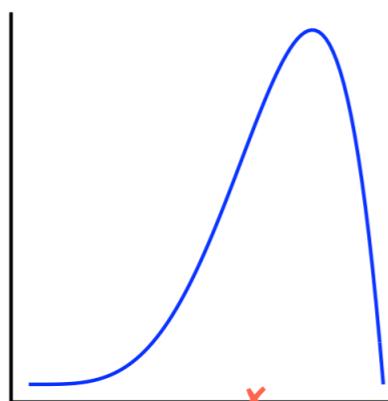
0.4

$$\text{Total Loss} = 0.9 + 0.3 + 0.4 = 1.6$$

$$\text{Loss of Best Fixed Action} = 0.2 + 0.3 + 0.4 = 0.9$$

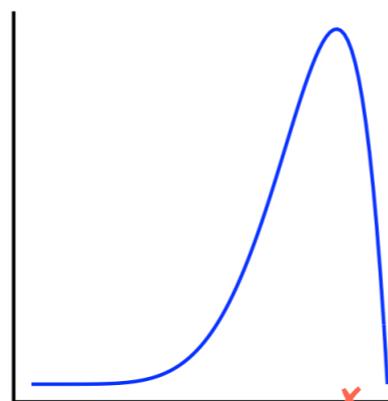
Action

A



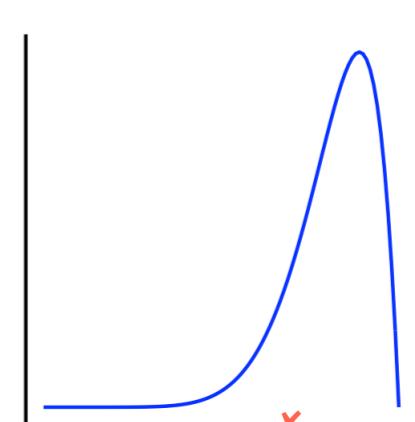
Loss

0.7



Loss

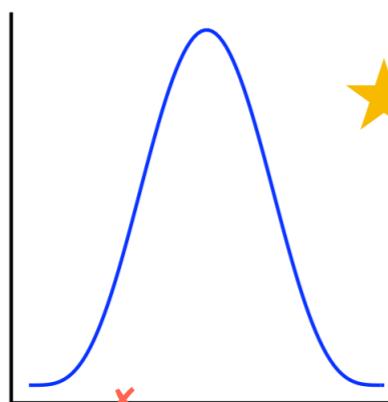
0.8



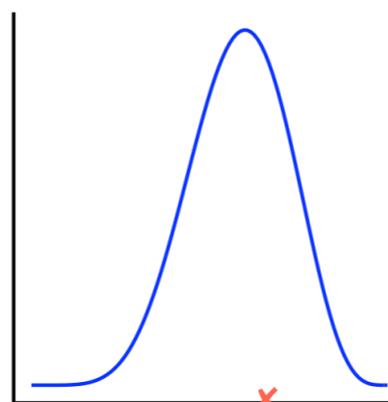
Loss

0.7

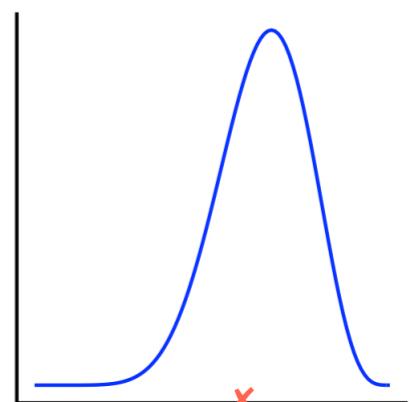
B



0.9

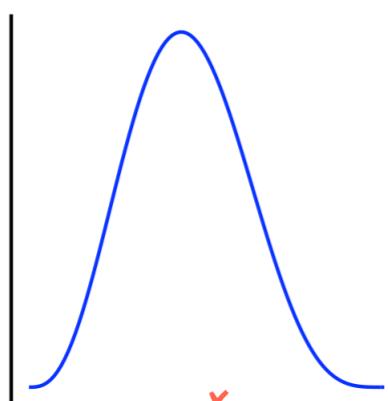


0.7

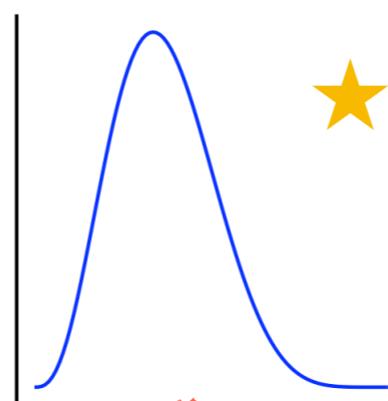


0.2

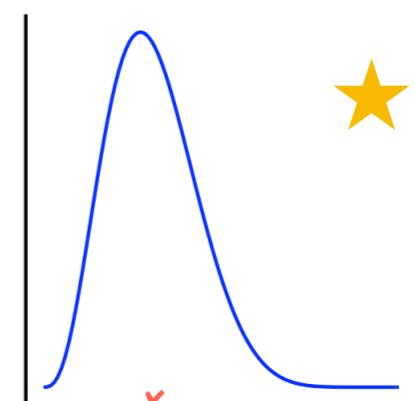
C



0.2



0.3



0.4

$$\text{Total Loss} = 0.9 + 0.3 + 0.4 = 1.6$$

$$\text{Loss of Best Fixed Action} = 0.2 + 0.3 + 0.4 = 0.9$$

$$\text{Regret} = 1.6 - 0.9 = 0.7$$

# Thompson Sampling

## Beta Bernoulli

$$P(\theta_{i,t} | \ell_{i,t}) = \frac{\text{posterior}}{\text{beta}} \frac{\text{liklihood}}{\text{bernonulli}} \frac{\text{prior}}{\text{beta}}$$
$$P(\theta_{i,t} | \ell_{i,t}) = P(\ell_{i,t} | \theta_{i,t}) P(\theta_{i,t})$$

# Thompson Sampling

## Beta Bernoulli

$$P(\theta_{i,t} | \ell_{i,t}) = \frac{\text{posterior}}{\text{beta}} \frac{\text{liklihood}}{\text{bernonulli}} \frac{\text{prior}}{\text{beta}}$$
$$P(\theta_{i,t} | \ell_{i,t}) = P(\ell_{i,t} | \theta_{i,t}) P(\theta_{i,t})$$

### Algorithm:

Set parameters for each action  $\alpha_i = 1, \beta_i = 1$

For each timestep  $t = 1, 2, \dots, T$

1. Sample from Beta( $\alpha_i, \beta_i$ ) for every action  $i \in \{1, \dots, N\}$  and chose the minimum
2. Observe losses  $\ell_{i,t}$  for every action
3. Perform Bernoulli trial  $\tilde{\ell}_{i,t} \sim \text{Bernoulli}(\ell_{i,t})$
4. Update parameters:

$$\alpha_i = \alpha_i + \tilde{\ell}_{i,t}$$

$$\beta_i = \beta_i + 1 - \tilde{\ell}_{i,t}$$

# Thompson Sampling

## Beta Bernoulli

$$P(\theta_{i,t} | \ell_{i,t}) = \frac{\text{posterior}}{\text{beta}} P(\ell_{i,t} | \theta_{i,t}) \frac{\text{liklihood}}{\text{beroulli}} P(\theta_{i,t}) \frac{\text{prior}}{\text{beta}}$$

### Algorithm:

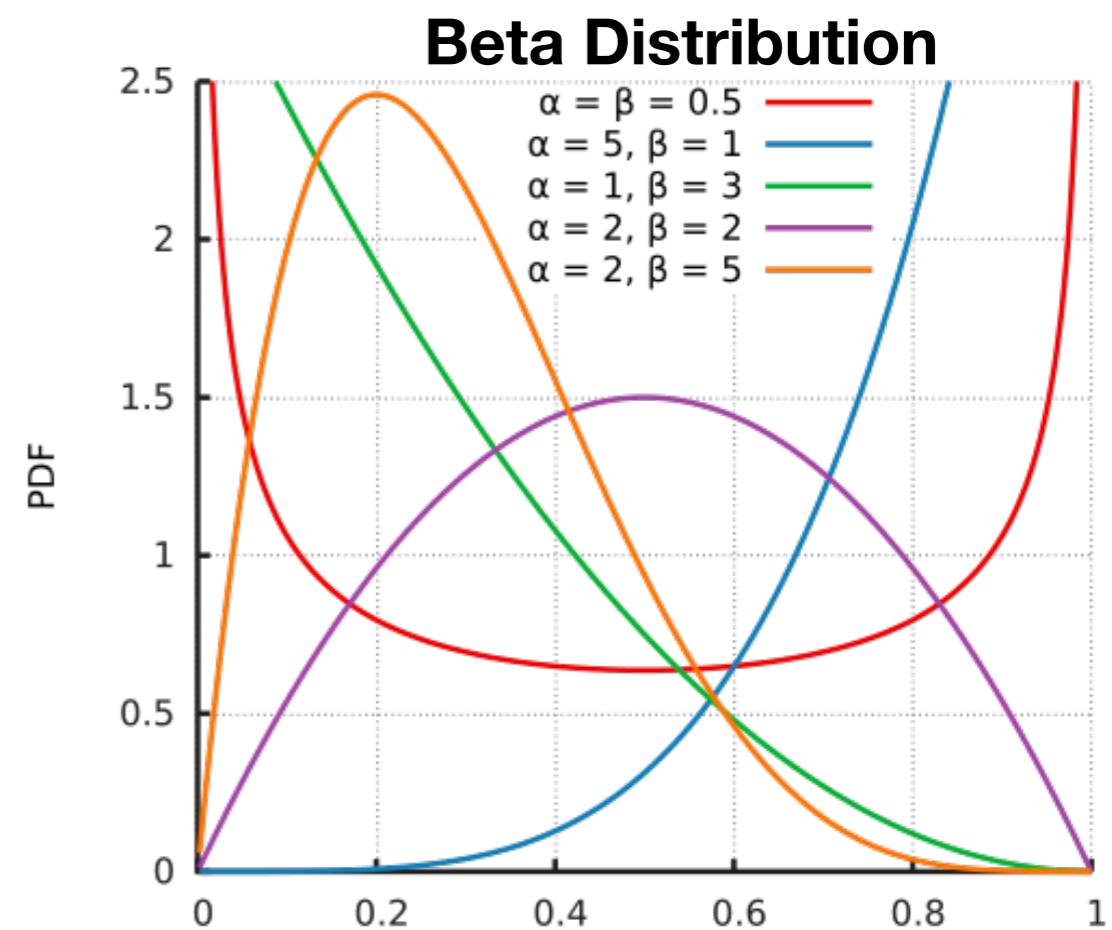
Set parameters for each action  $\alpha_i = 1, \beta_i = 1$

For each timestep  $t = 1, 2, \dots, T$

1. Sample from  $\text{Beta}(\alpha_i, \beta_i)$  for every action  $i \in \{1, \dots, N\}$  and chose the minimum
2. Observe losses  $\ell_{i,t}$  for every action
3. Perform Bernoulli trial  $\tilde{\ell}_{i,t} \sim \text{Bernoulli}(\ell_{i,t})$
4. Update parameters:

$$\alpha_i = \alpha_i + \tilde{\ell}_{i,t}$$

$$\beta_i = \beta_i + 1 - \tilde{\ell}_{i,t}$$



# Other Algorithms

## Exponential Weighted Average

$$P(a_t = i) = \frac{e^{-\eta_t L_{i,t-1}}}{\sum_{j=1}^N e^{-\eta_t L_{j,t-1}}}$$

Algorithm	Type	Adversarial Regret Bound
Exponential Weighted Average	$\eta_t = \sqrt{8(\ln N)/t}$	$\mathcal{O}(\sqrt{T \log N} + \log N)$
	$\eta = \sqrt{8(\ln N)/T}$	$\mathcal{O}(\sqrt{T \log N})$
	$\eta_t = \min\{1, C\sqrt{(\ln N)/\text{Var}(\hat{L}_t)}\}$	$\mathcal{O}(\sqrt{\text{Var}(\hat{L}_T) \log N} + \log N)$
	AdaHedge	$\mathcal{O}(\sqrt{L^* \log N} + \log N)$

# Other Algorithms

## Follow the Perturbed Leader

$$a_t = \operatorname{argmin}_i L_{i,t-1} + Z_{i,t}$$

Algorithm	Type	Adversarial Regret Bound
Follow the Perturbed Leader	Uniform	$\mathcal{O}(\sqrt{TN})$
	Random Walk	$\mathcal{O}(\sqrt{T \log N} + \log T)$
	Exponential	$\mathcal{O}(\sqrt{L^* \log N} + \log N)$
	Dropout	$\mathcal{O}(\sqrt{L^* \log N} + \log N)$

# How does Thompson Sampling compare?

If losses of each arm are independent and identically distributed between 0 and 1 the regret of Beta-Binomial Thompson Sampling is bounded in:  $\mathcal{O}(\ln T)$

But what about adversarial losses?

**goal:**  $\max_{\ell} E[R_T]$

**where:**  $\ell \in [0, 1]^{N \times T}$

# Constant Sum Games

**Payoff Matrix:**  $P \in [0, 1]^{N \times N}$

## Row Player v.s. Column Player

**For every round:**

Row player chooses  $i$

Column player chooses  $j$

Row player incurs loss  $P_{i,j}$

Column player incurs loss  $1 - P_{i,j}$

# Constant Sum Games

**Payoff Matrix:**  $P \in [0, 1]^{N \times N}$

## Row Player v.s. Column Player

**For every round:**

Row player chooses  $i$

Column player chooses  $j$

Row player incurs loss  $P_{i,j}$

Column player incurs loss  $1 - P_{i,j}$

identity	rock paper scissors	uniform
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0 & 1 \\ 1 & 0.5 & 0 \\ 0 & 1 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.9 & 0.8 & 0.8 \\ 0.5 & 0.6 & 0.4 \\ 0.3 & 0.4 & 0.6 \end{bmatrix}$

# Constant Sum Games

**Payoff Matrix:**  $P \in [0, 1]^{N \times N}$

## Row Player v.s. Column Player

For every round:

Row player chooses  $i$

Column player chooses  $j$

Row player incurs loss  $P_{i,j}$

Column player incurs loss  $1 - P_{i,j}$

*Let's duel!*

**identity**

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

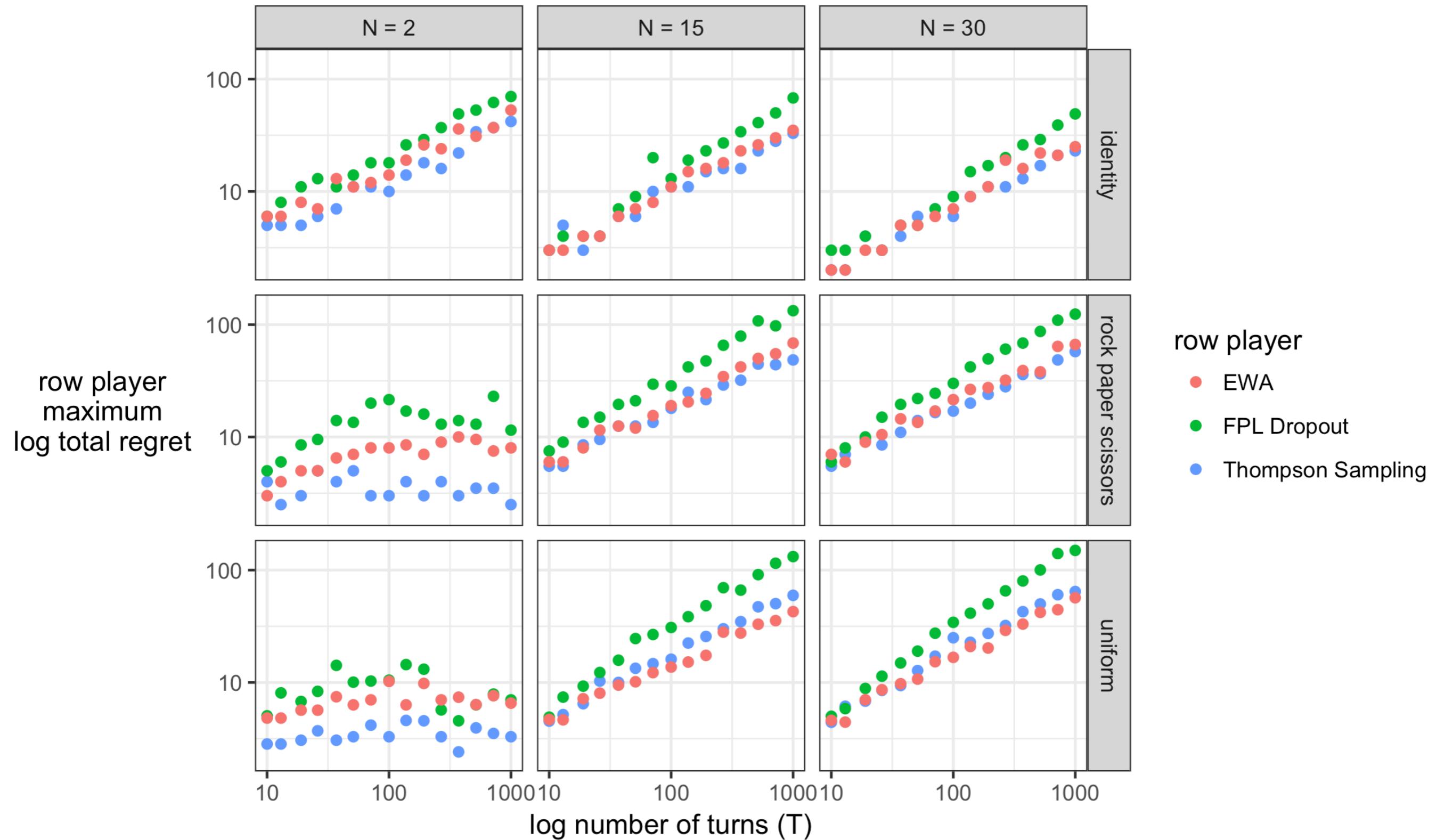
**rock paper scissors**

$$\begin{bmatrix} 0.5 & 0 & 1 \\ 1 & 0.5 & 0 \\ 0 & 1 & 0.5 \end{bmatrix}$$

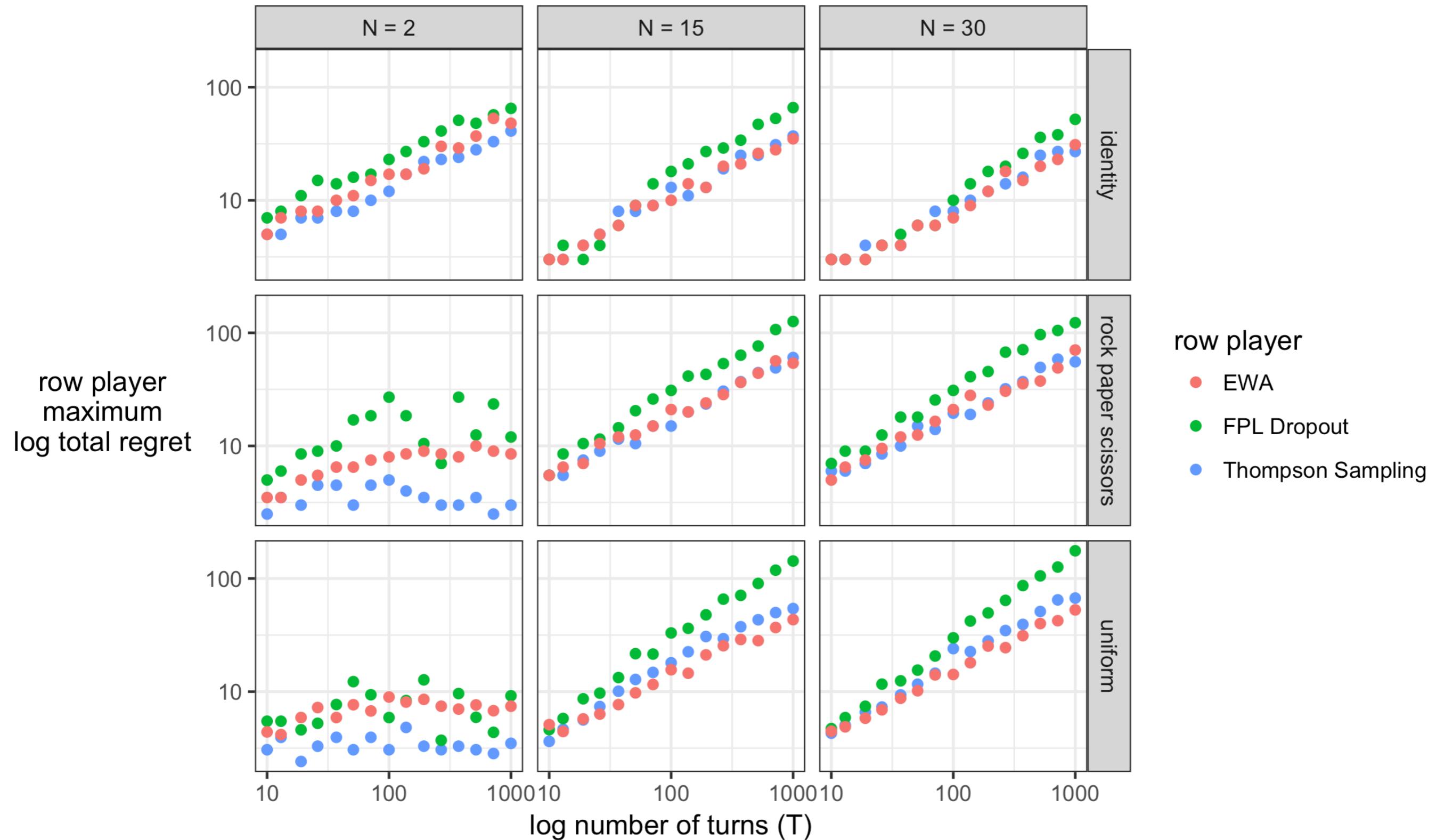
**uniform**

$$\begin{bmatrix} 0.9 & 0.8 & 0.8 \\ 0.5 & 0.6 & 0.4 \\ 0.3 & 0.4 & 0.6 \end{bmatrix}$$

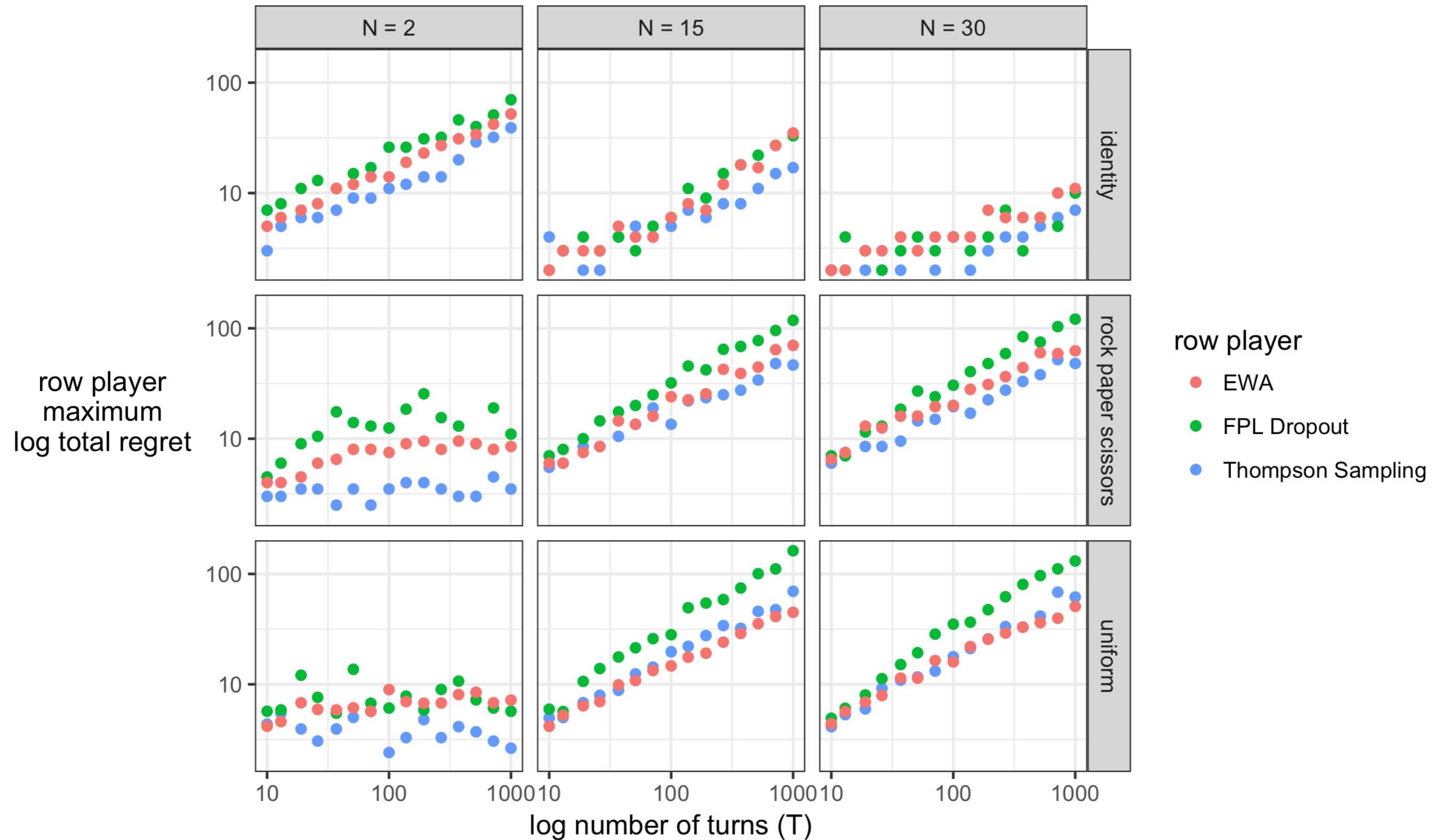
## column player: Thompson Sampling



## column player: EWA



## column player: FPL Dropout



# Evolutionary Methods

**goal:**  $\max_{\ell} E[R_T]$

**where:**  $\ell \in [0, 1]^{N \times T}$

## Vocabulary:

‘individual’ = loss matrix:  $\ell \in [0, 1]^{N \times T}$

‘population’ = group of 100 individuals

## Algorithm:

Initialize population uniformly at random

For each generation:

    Estimate regret of each individual

    Remove two thirds individuals with lowest regret

    Remaining individuals have two children each

    Repeat

**Note: we maintain separate populations for each algorithm and values of N and T**

## Evolutionary Method Results

