Statistical Verification

of Probabilistic Termination Proofs

W. Bailkoski | M. Giacobbe | TA. Henzinger | K. Kueffner | L. Stella

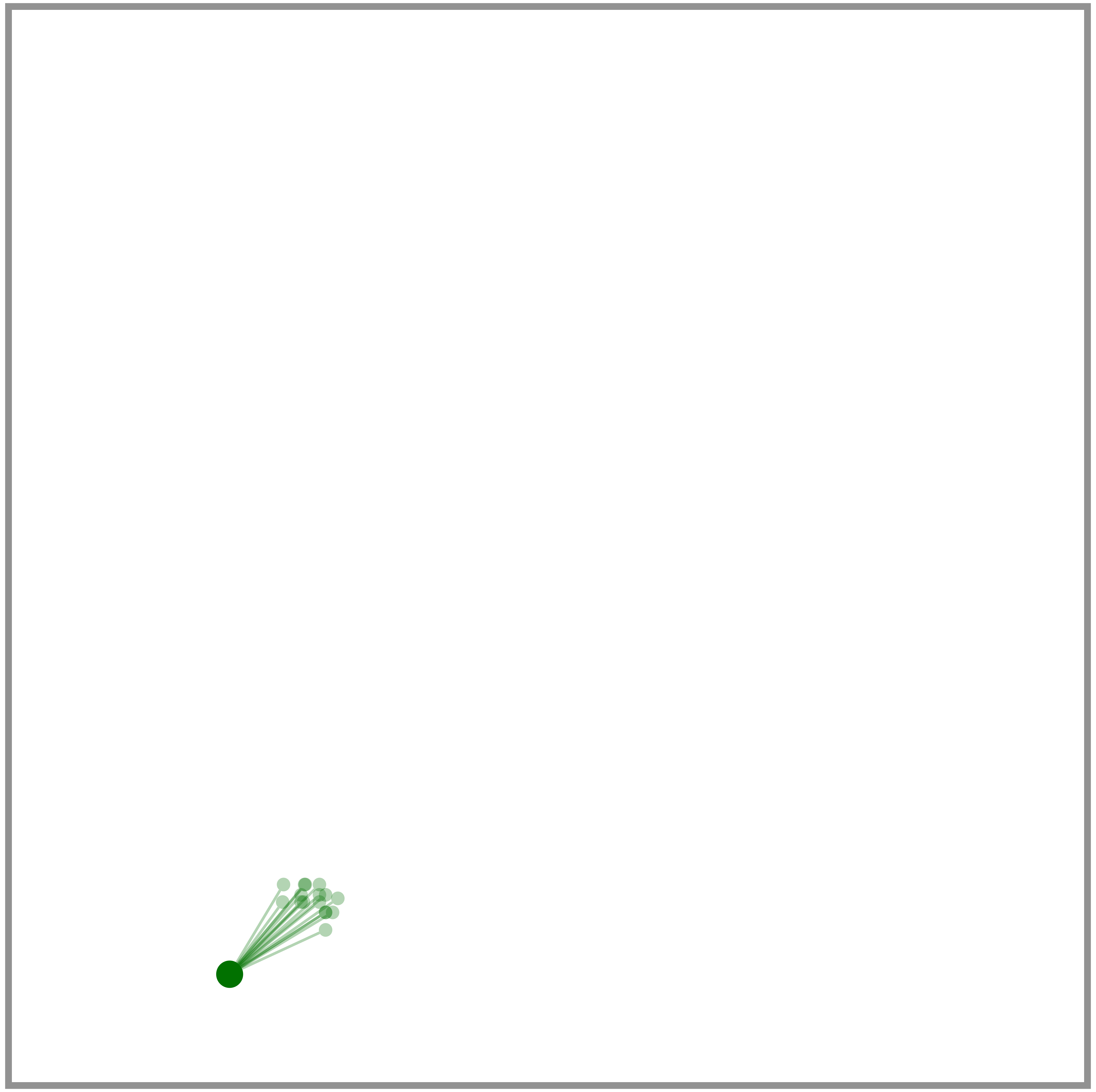
Institute of Science and Technology Austria

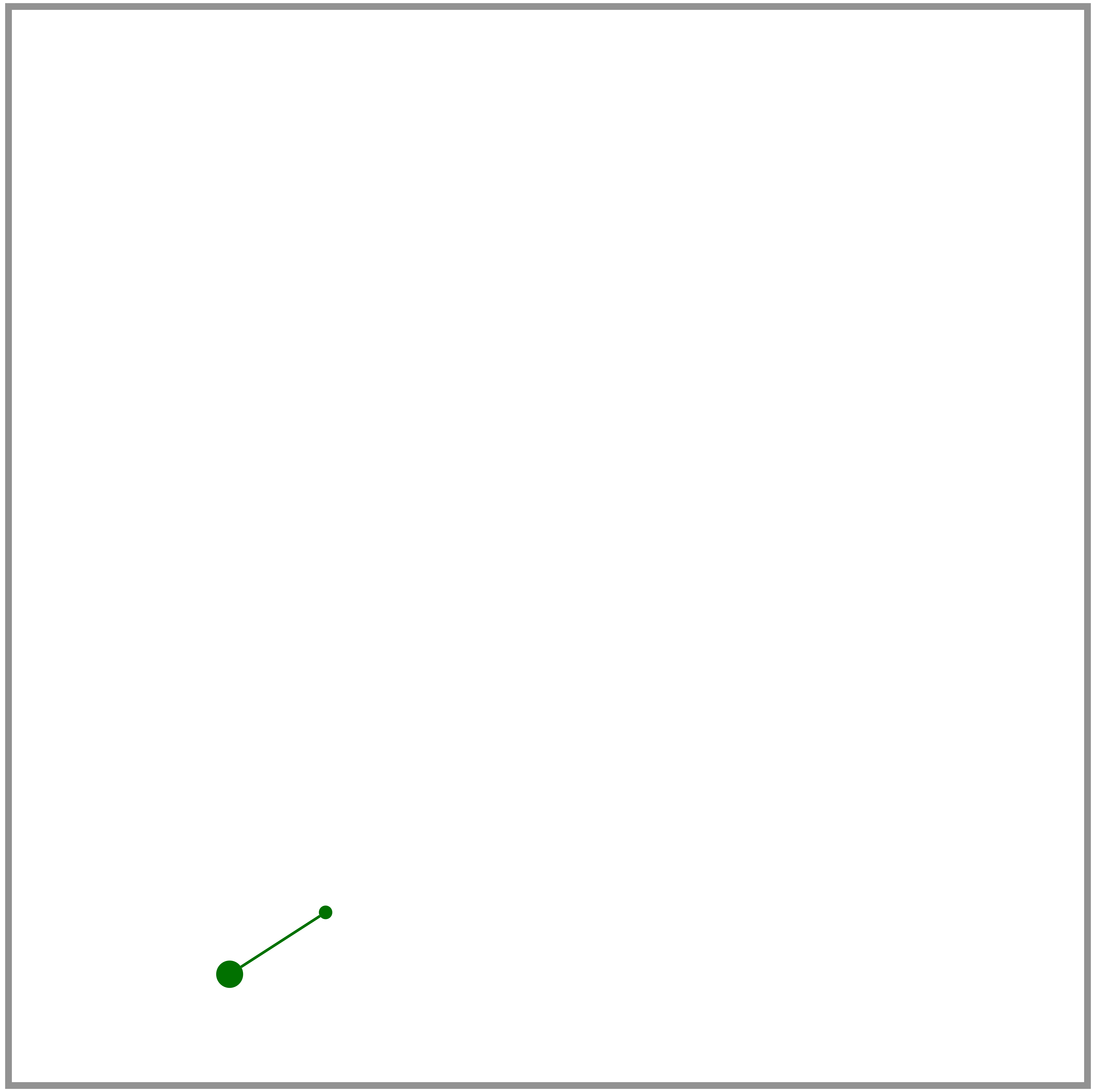
Motivation.

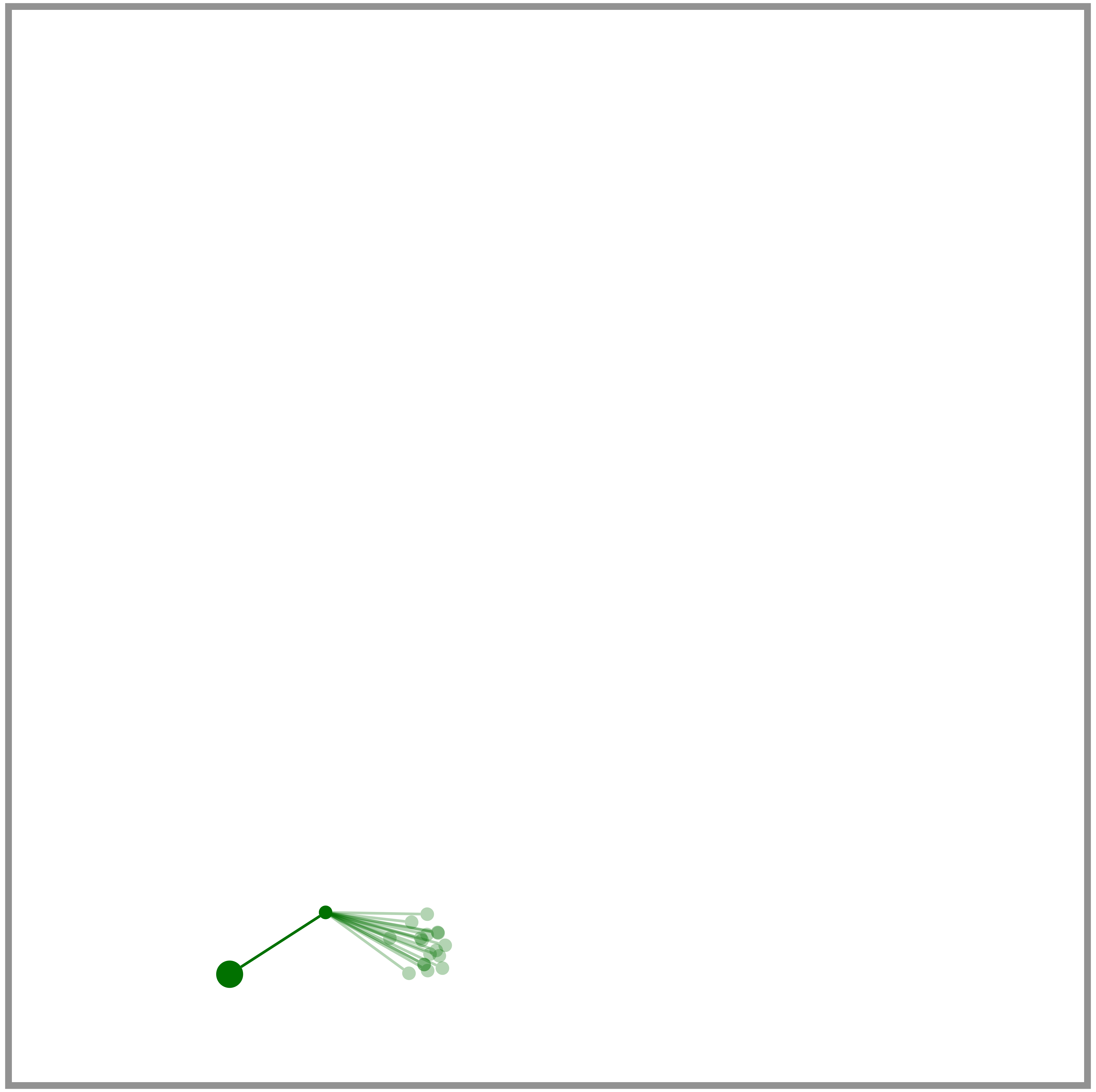
Termination of a stochastic process.

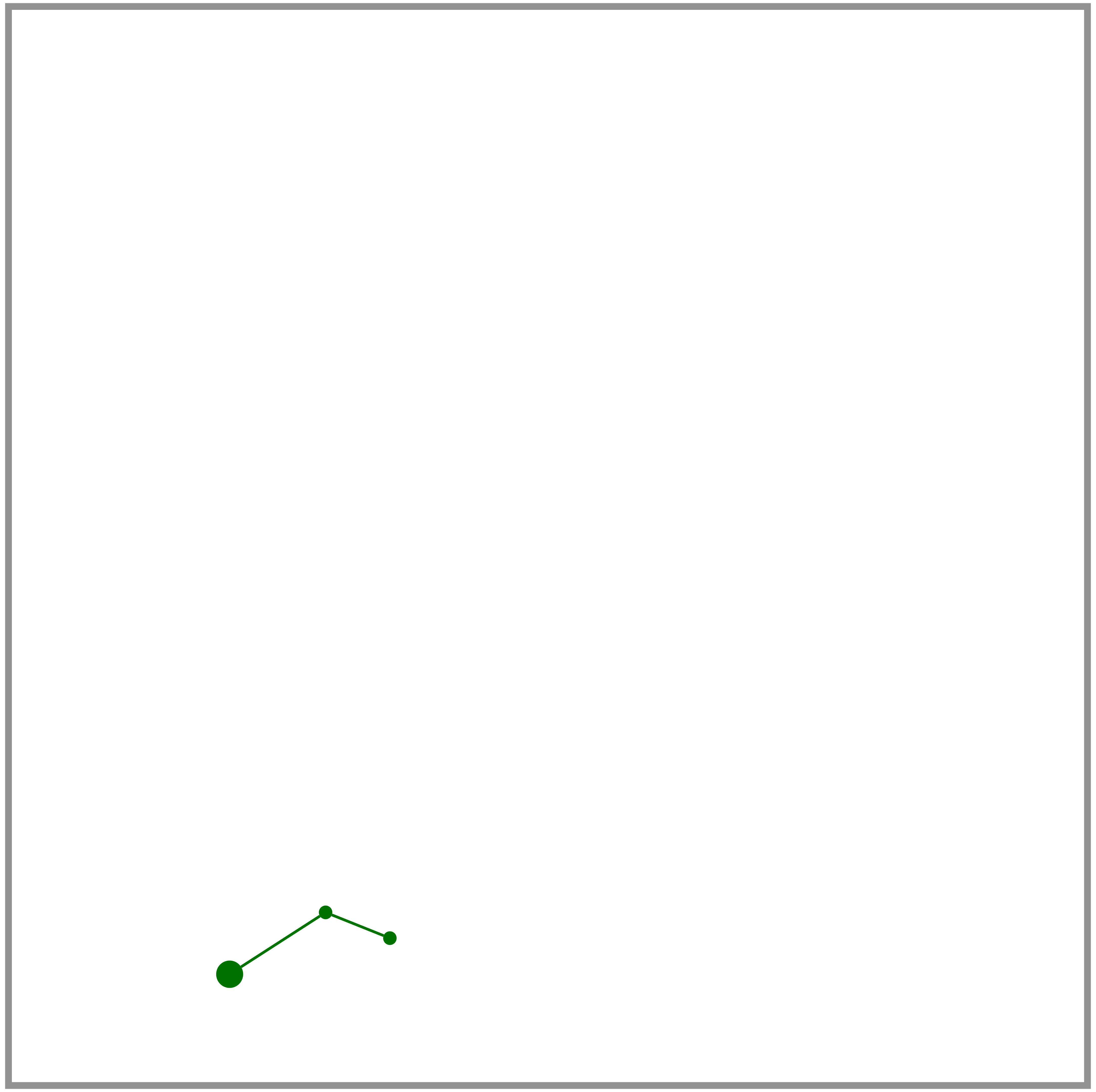
$$\mathbb{X} := (\mathcal{X}, d)$$

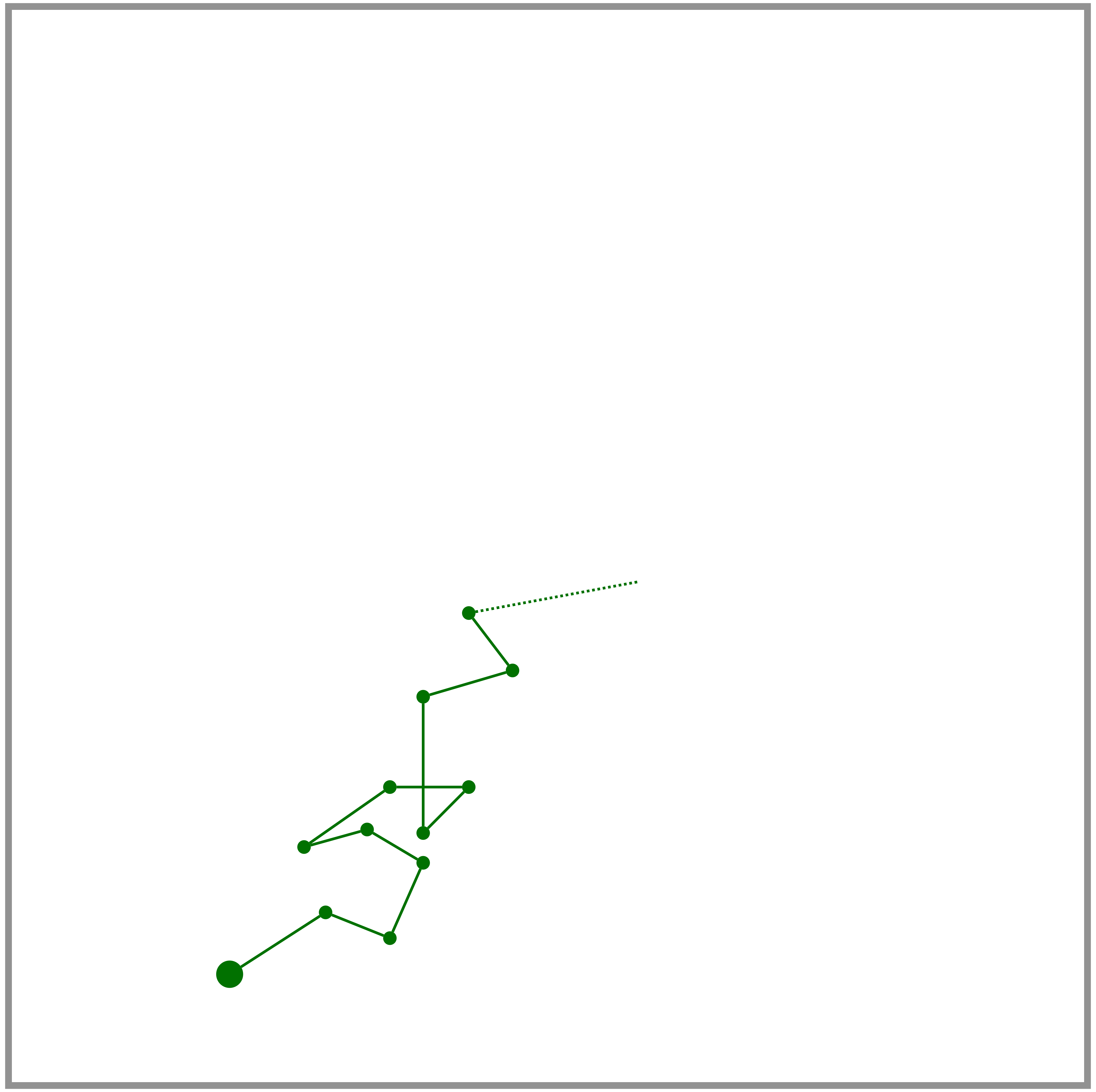
$$\mathcal{X} \subset \mathbb{R}^2$$











$$\overrightarrow{X} := (X_t)_{t \in \mathbb{N}} \sim (P, x_0)$$

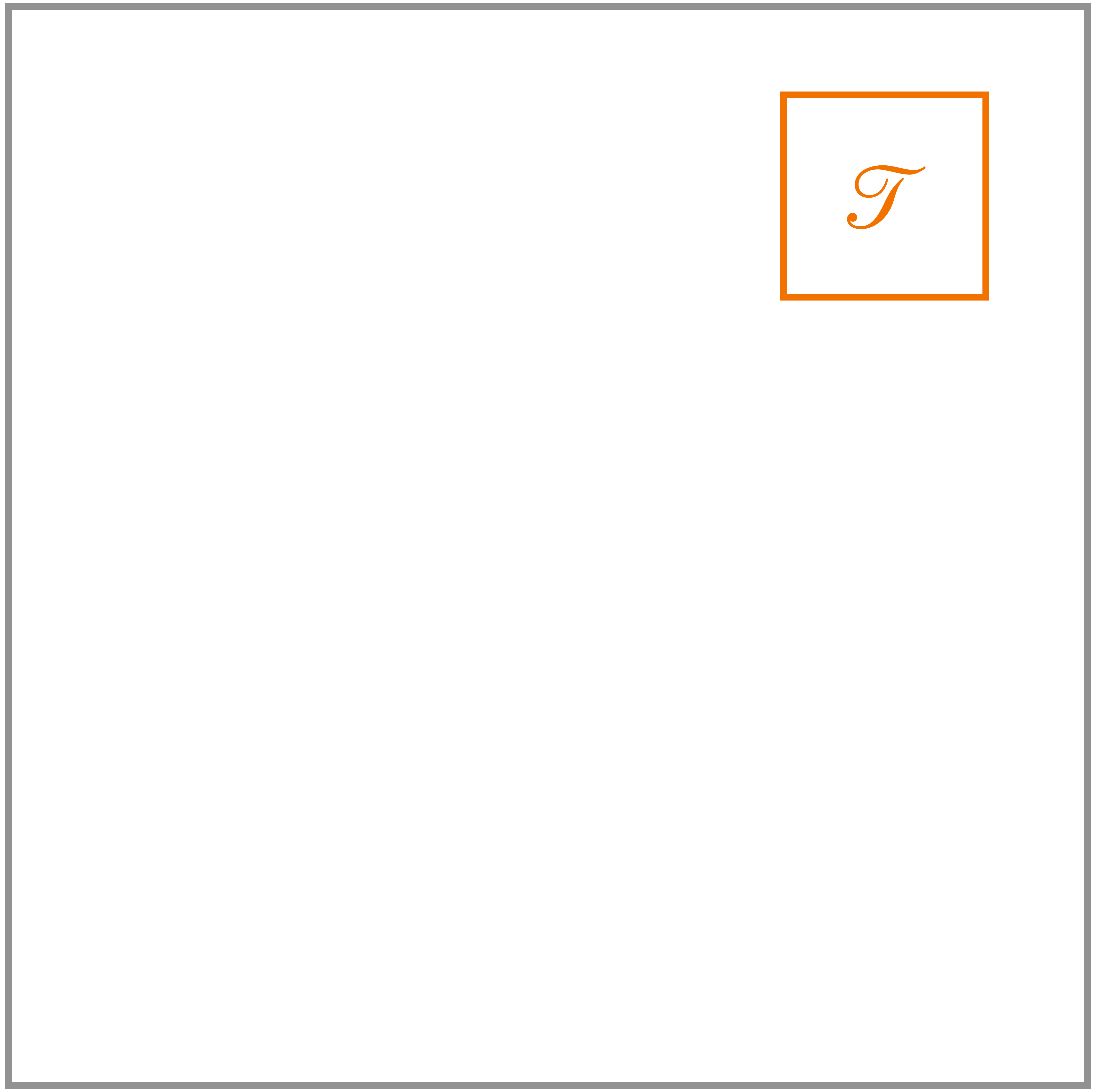
Markov process

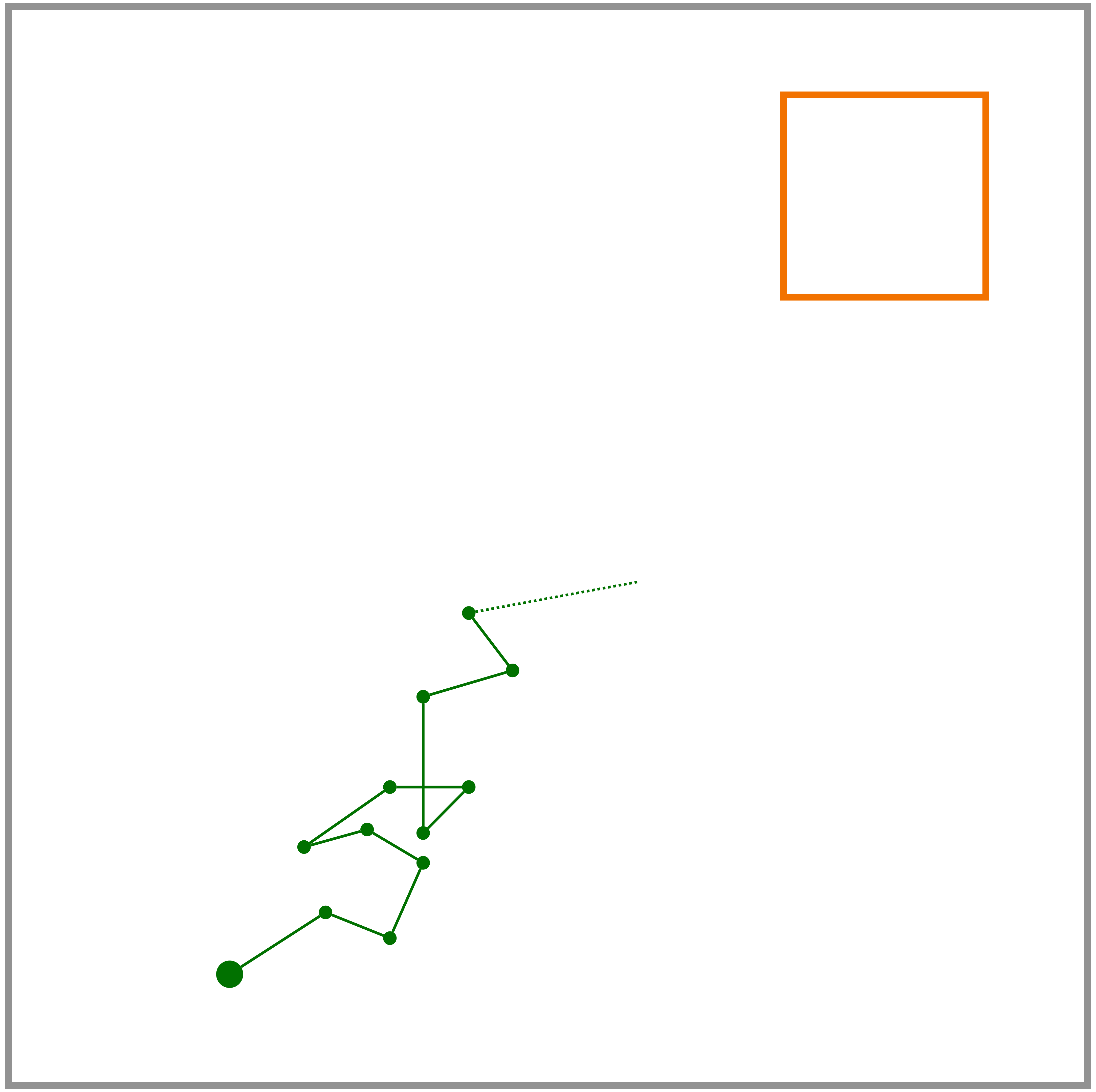
$$\forall x \in \mathscr{X} : Y \sim xP$$

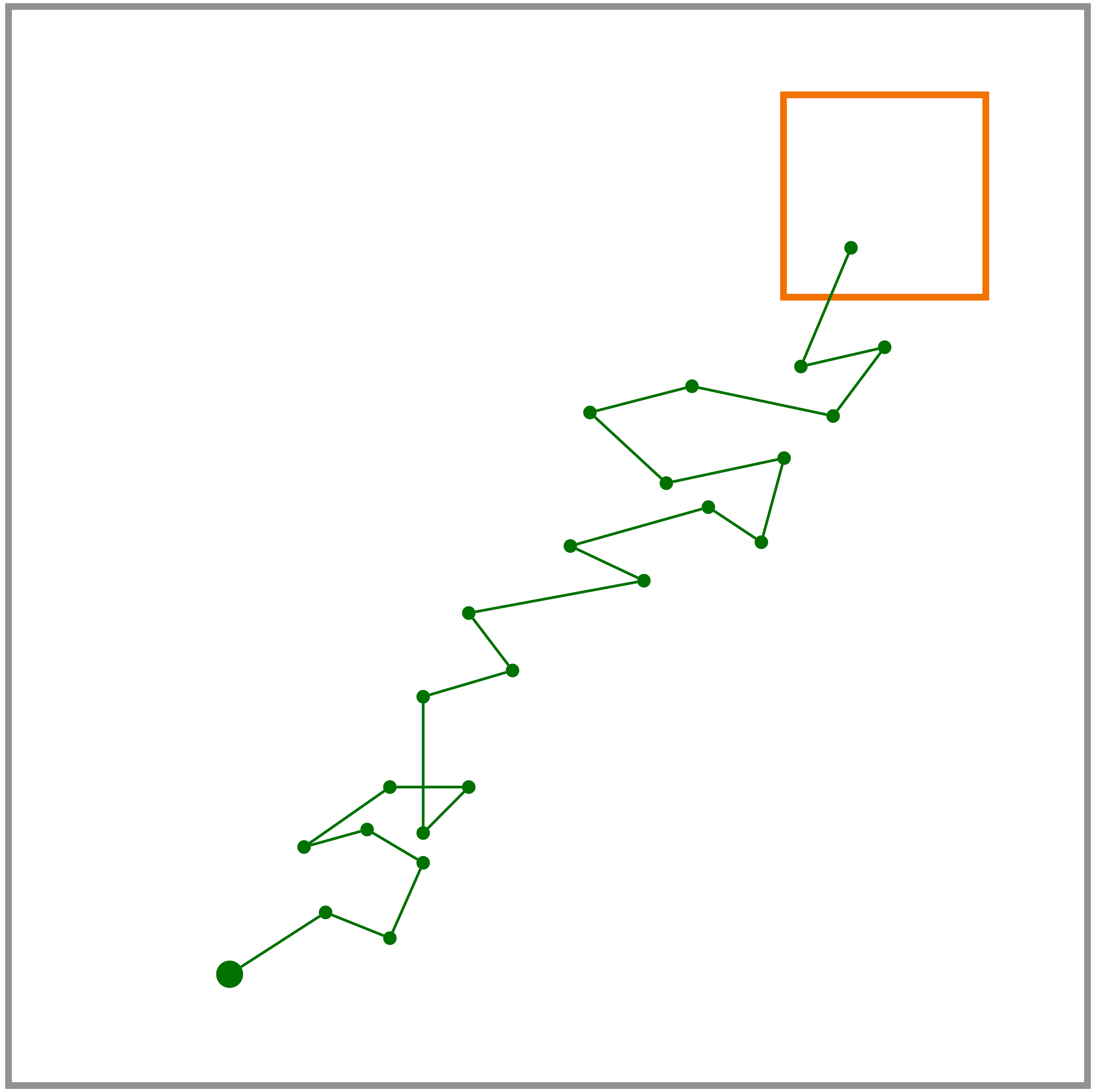
Markov transition kernel

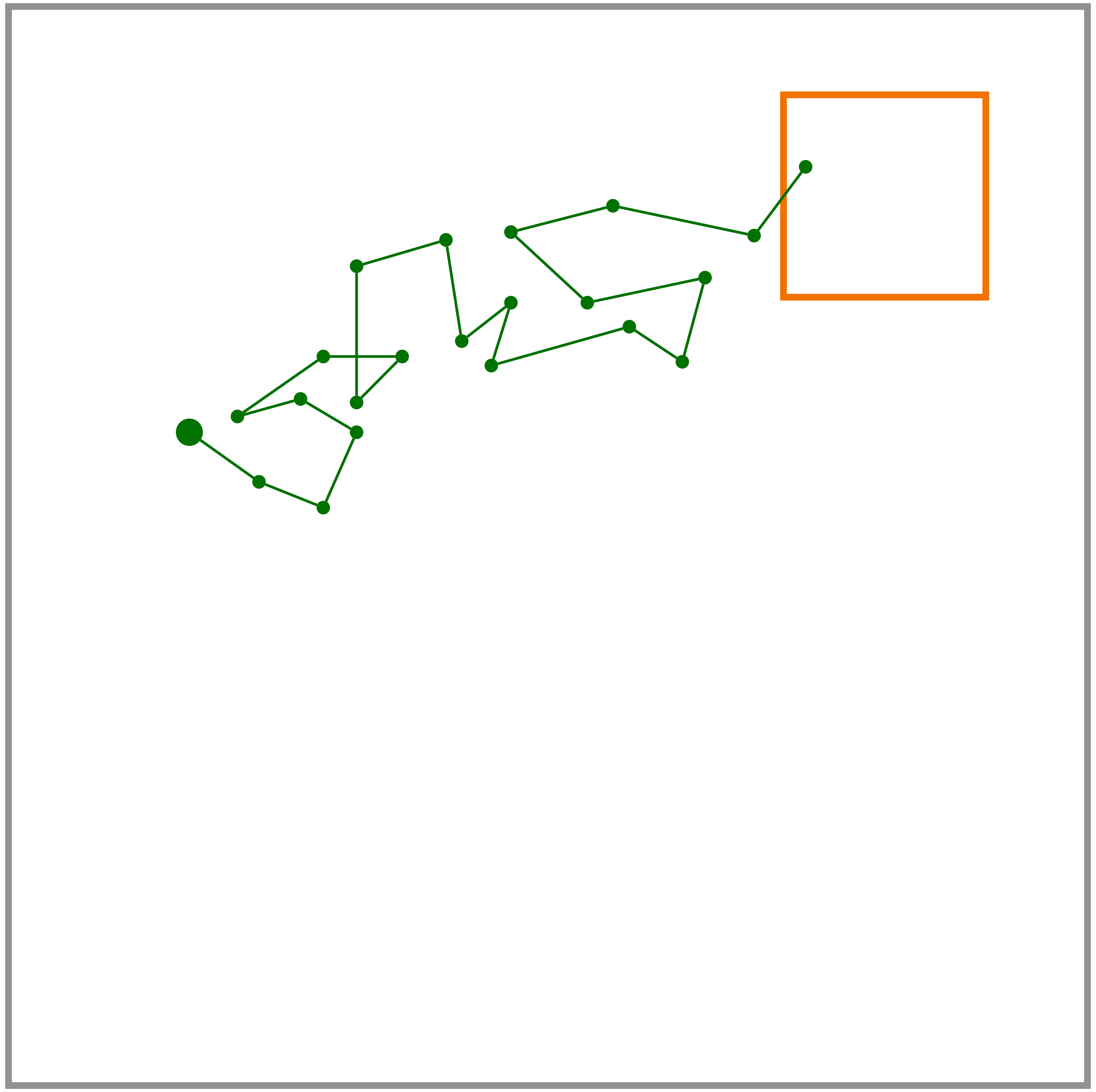
Question:

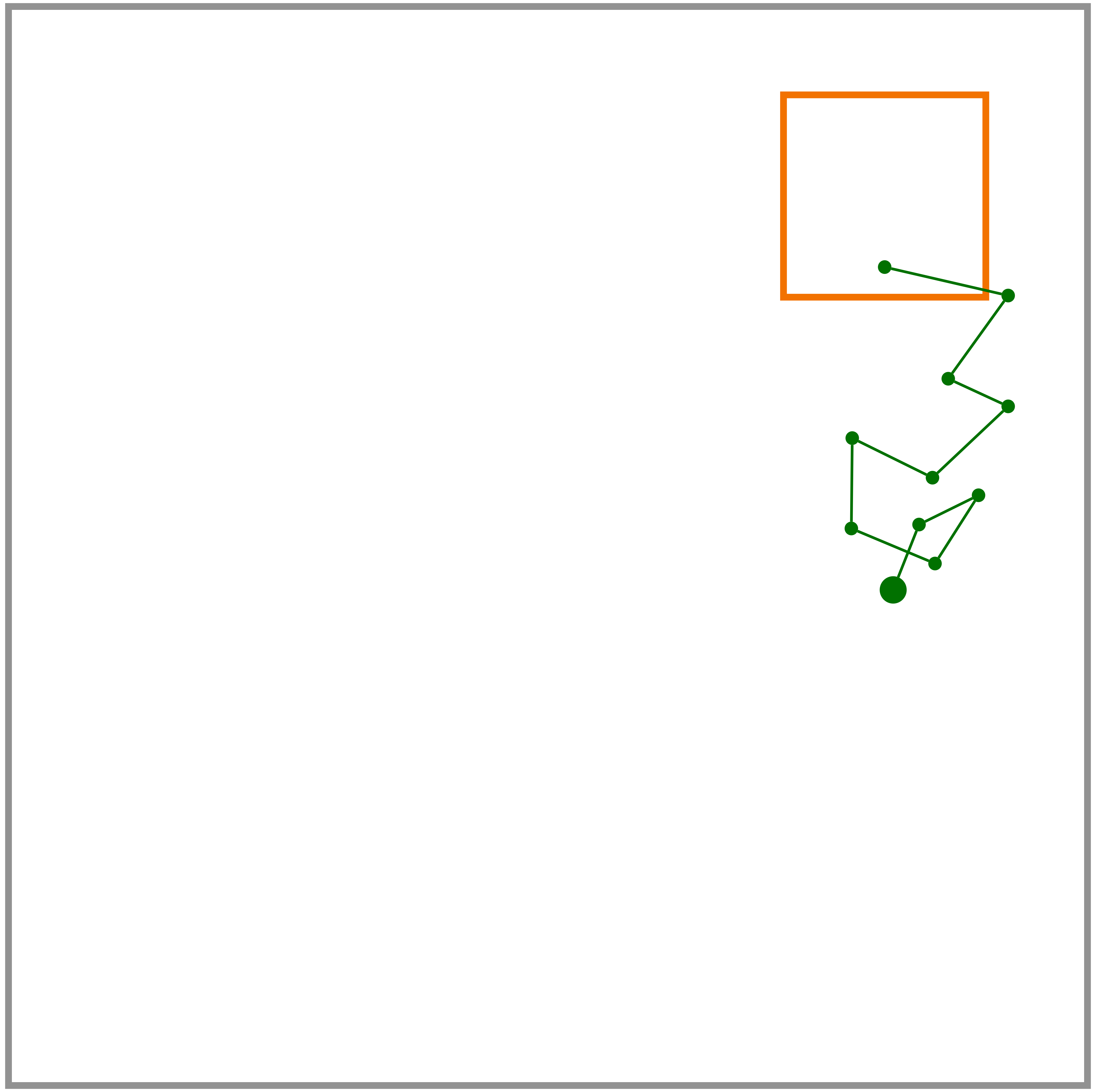
Does it reach a target region from every initial state?


𝒯









How to prove this?

Ranking super-martingales!

$$\overrightarrow{M} := (M_t)_{t \in \mathbb{N}}$$

Stochastic process

$$\forall t \in \mathbb{N}:$$

$$M_t \geq K \in \mathbb{R}$$

Lower bounded

$$\forall t \in \mathbb{N}:$$

$$\mathbb{E}(M_{t+1} \mid \overrightarrow{M_t}) \leq M_t - \varepsilon$$

Decreases in expectation

Goal:

Find a function $f: \mathcal{X} \to \mathbb{R}$ *such that*

$$f(\overrightarrow{X}) := (f(X_t))_{t \in \mathbb{N}}$$

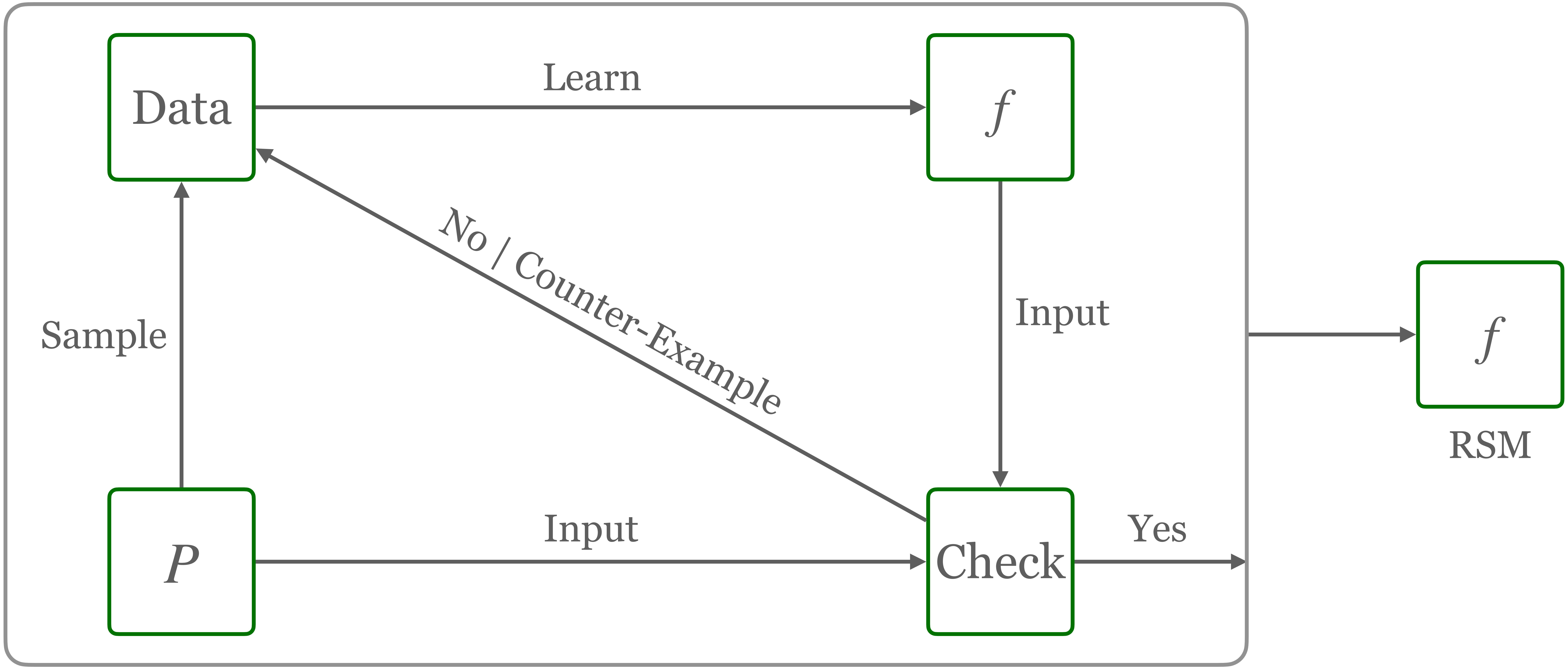
is a ranking super-martingale on $\mathcal{X} \setminus \mathcal{T}$.

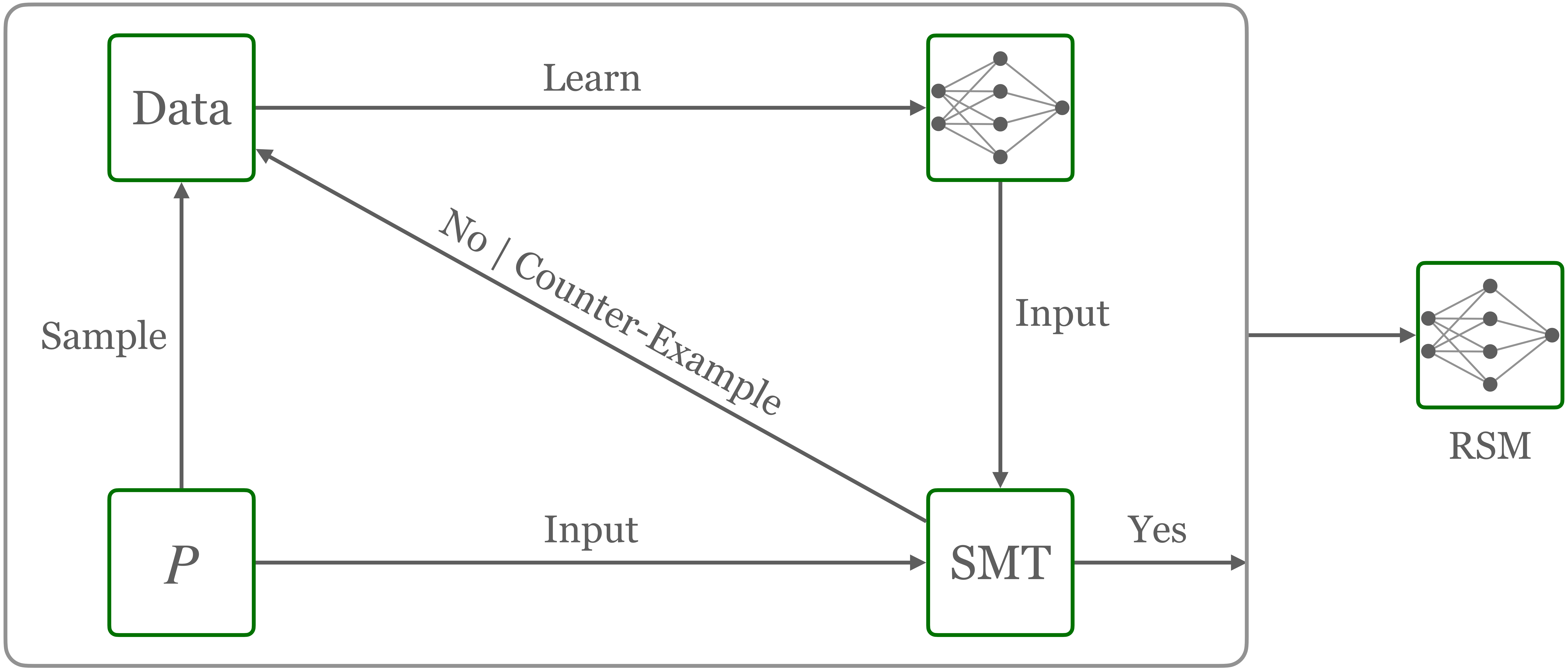
Intuition:

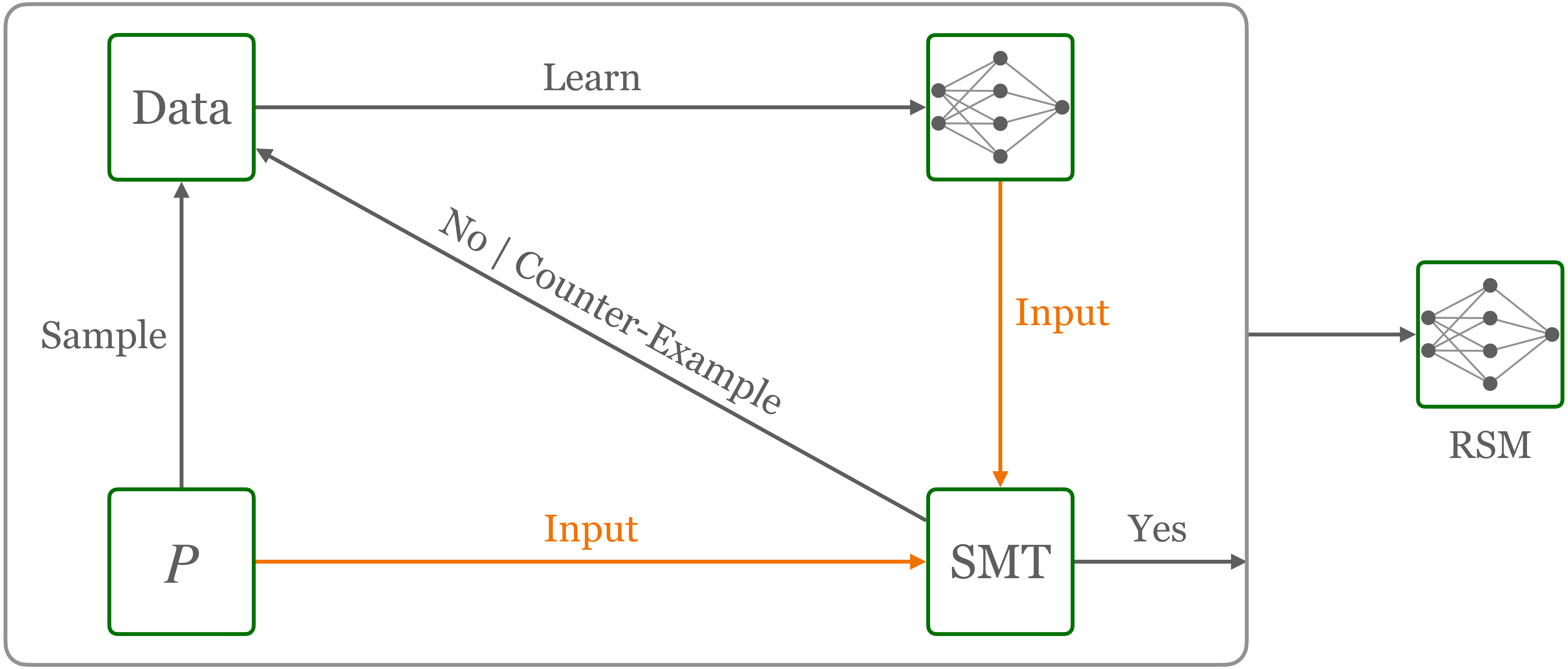
With every step
the process gets closer to the target
in expectation.

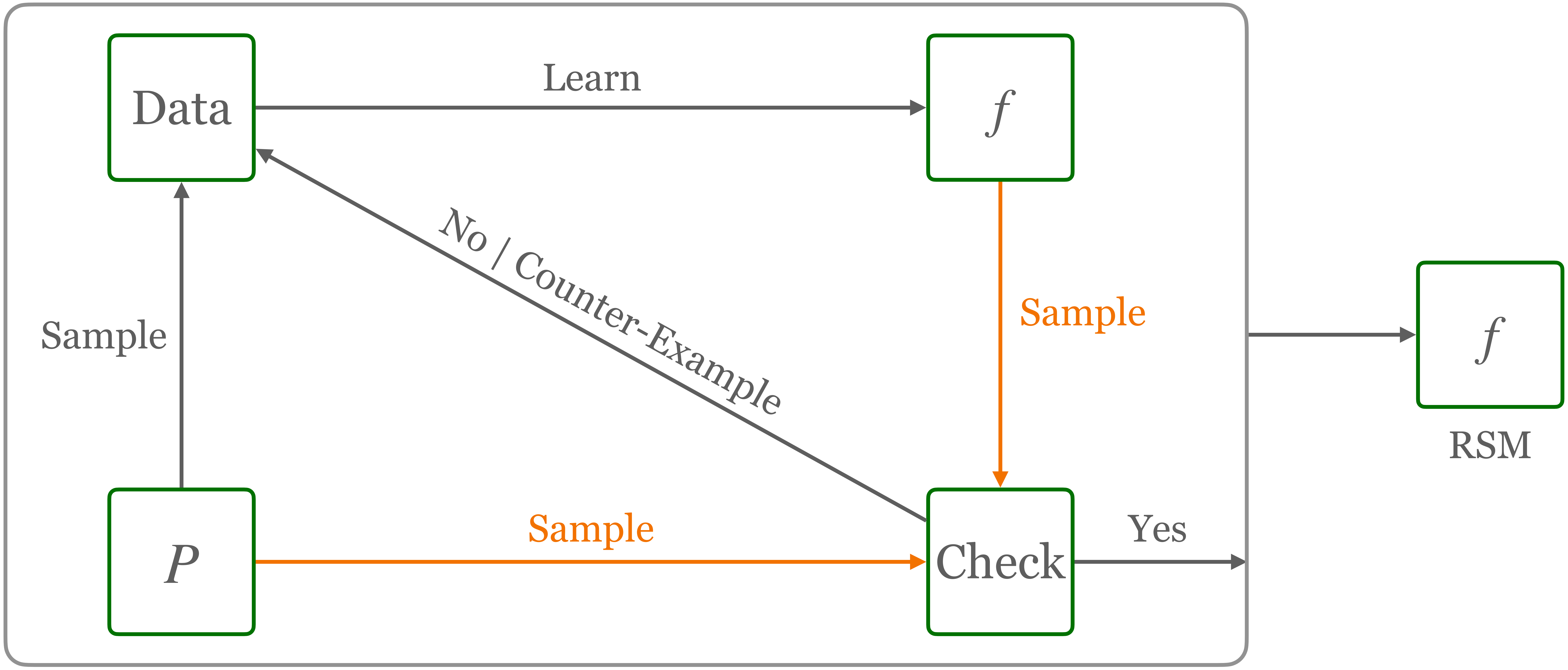
One approach:

Counterexample-guided inductive synthesis.


Data
Learn
f
Sample
No / Counter-Example
Input
f
RSM
P
Input
Check
Yes


Data
Learn
Sample
No / Counter-Example
Input
P
Input
SMT
Yes
RSM


Data
Learn
Sample
No / Counter-Example
Input
P
Input
SMT
Yes
RSM


Data
Learn
f
No / Counter-Example
Sample
Sample
f
RSM
P
Sample
Check
Yes

Statistical Verification.

What do we gain?
What do we loose?

f	:	Known Neural Network	$<$	Known Lipschitz Constant + Bounded + Sample Access
P	:	Known Dynamics	$<$	Known Lipschitz Constant + Sample Access
Yes/No	:	Sure Guarantee	$>$	Probabilistic Guarantee

Statistical Verification.

Checking the super-martingale condition.

Problem Instance:

A problem instance is the tuple

$$\mathcal{I} := (P, f, \gamma_P, \gamma_f, c_f, \delta)$$

Problem Instance:

A problem instance is the tuple

$$\mathcal{I} := (\underbrace{P, f}_{\text{System}}, \gamma_P, \gamma_f, c_f, \delta)$$

System

Problem Instance:

A problem instance is the tuple

$$\mathscr{I} := (P, f, \gamma_P, \gamma_f, c_f, \delta)$$

Certificate

Problem Instance:

A problem instance is the tuple

$$\mathscr{I} := (P, f, \gamma_P, \gamma_f, c_f, \delta)$$

System Lipschitz constant

Problem Instance:

A problem instance is the tuple

$$\mathscr{I} := (P, f, \gamma_P, \gamma_f, c_f, \delta)$$

Certificate Lipschitz constant

Problem Instance:

A problem instance is the tuple

$$\mathcal{I} := (P, f, \gamma_P, \gamma_f, c_f, \delta)$$

Certificate range

$$c_f := \sup f - \inf f$$

Problem Instance:

A problem instance is the tuple

$$\mathcal{I} := (P, f, \gamma_P, \gamma_f, c_f, \delta)$$

Confidence

$\delta \in (0,1)$

Problem Statement:

Given a problem instance $\mathcal{I}$ find an algorithm $\mathcal{A}$
with knowledge of $(\gamma_P, \gamma_f, c_f, \delta)$ and sample access of (P, f), s.t.

$$\mathcal{A}(\delta) \iff \forall x \in \mathcal{X} \backslash \mathcal{T} : \mathbb{E}_{Y \sim_x P}(f(Y)) \leq f(x) - \varepsilon$$

with probability $1 - \delta$ upon termination.

Problem Reduction:

How to approach this?

$$\forall x \in \mathscr{X} \setminus \mathscr{T} : \mathbb{E}_x(f(Y)) \leq f(x) - \varepsilon$$

$$\forall x \in \mathcal{X} \setminus \mathcal{T} : \mathbb{E}_x(f(Y)) \leq f(x) - \varepsilon$$

$$\iff \forall x \in \mathcal{X} \setminus \mathcal{T} : \underbrace{\mathbb{E}_x(f(Y)) - f(x) + \varepsilon}_{R_x} \leq 0$$

$$\forall x \in \mathcal{X} \backslash \mathcal{T} : \mathbb{E}_x(f(Y)) \leq f(x) - \varepsilon$$

$$\iff \forall x \in \mathcal{X} \backslash \mathcal{T} : \mathbb{E}_x(f(Y)) - f(x) + \varepsilon \leq 0$$

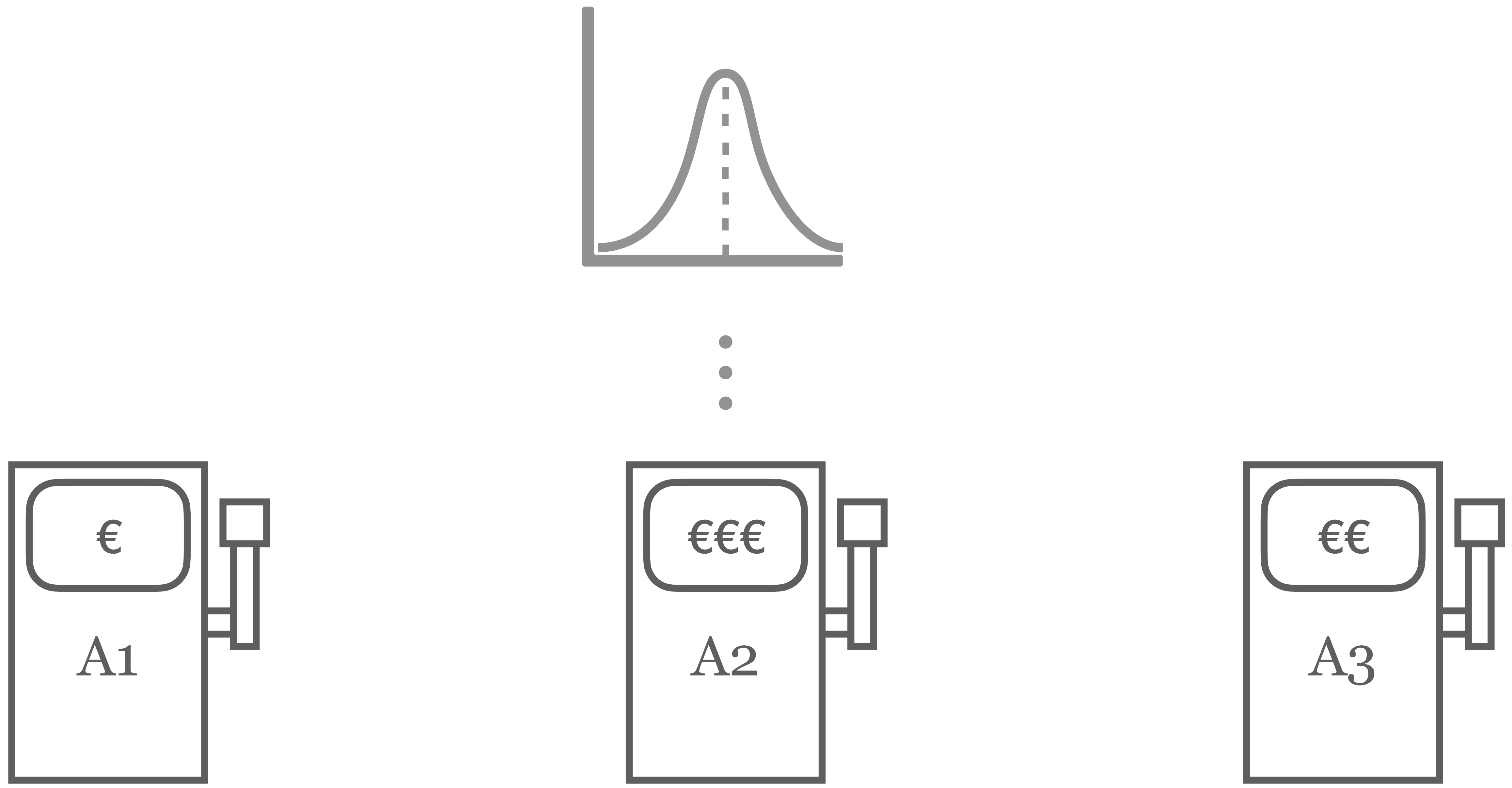
$$\iff \sup_{x \in \mathcal{X} \backslash \mathcal{T}} R_x \leq 0$$

Actual Problem.

Find point with highest expected reward.

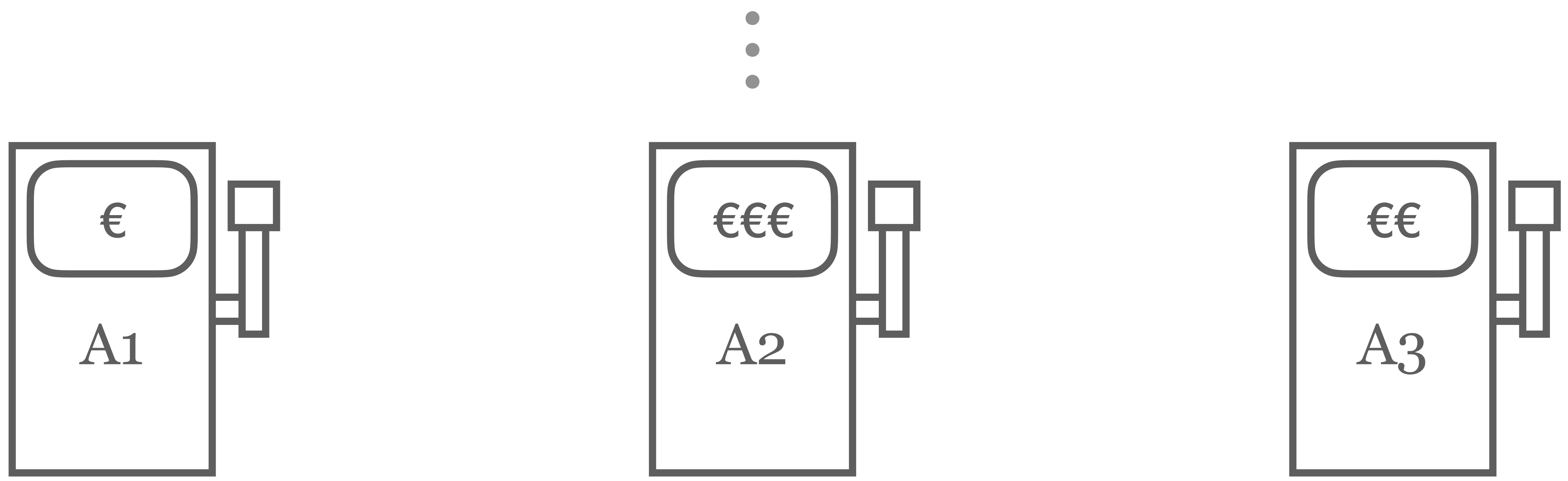
Multi-Armed Bandits.

A small detour.

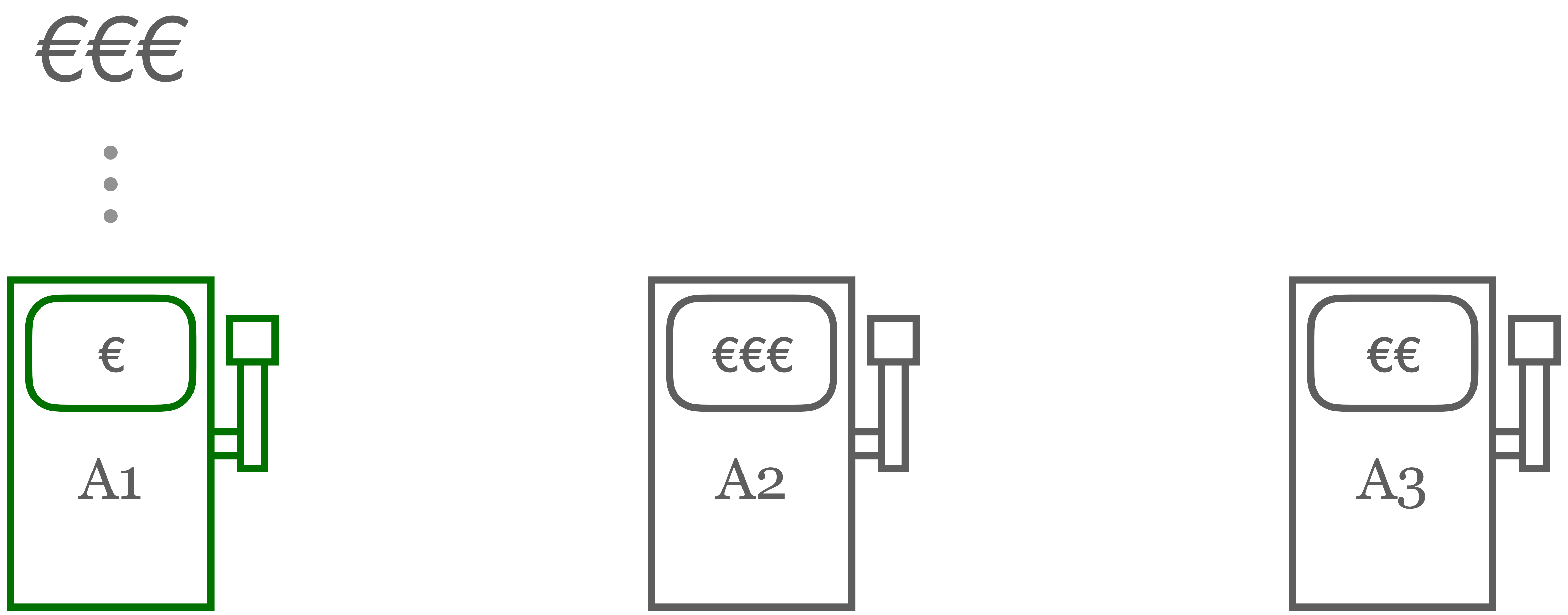


Reward distributions

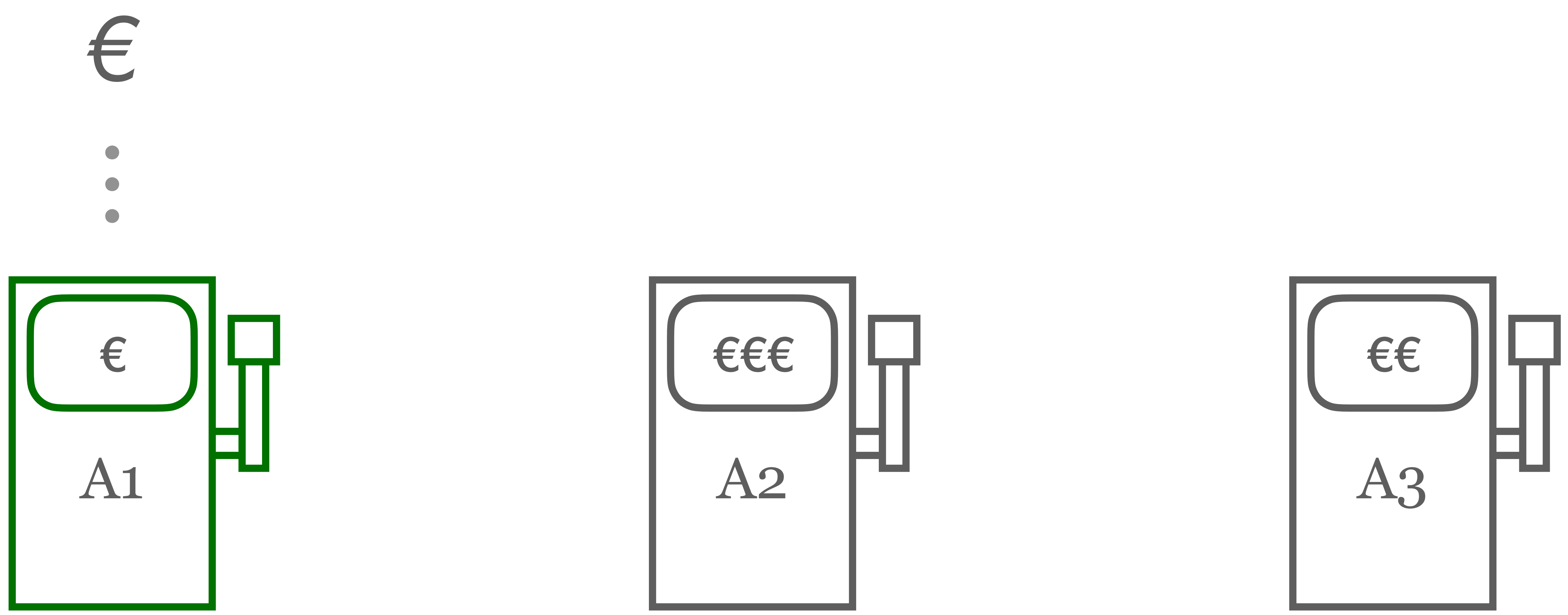
(R_2, σ_2^2)



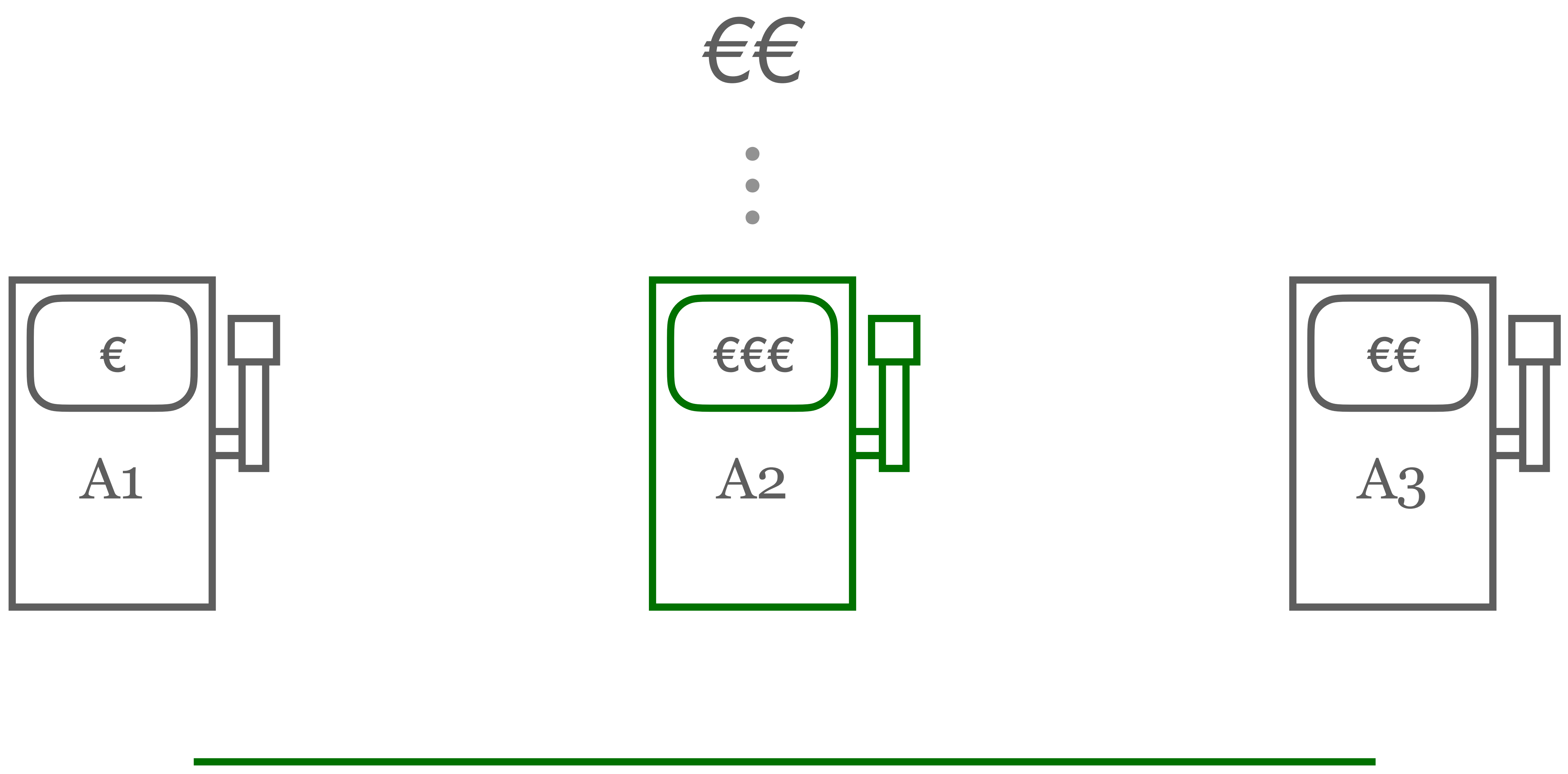
Sub-Gaussian



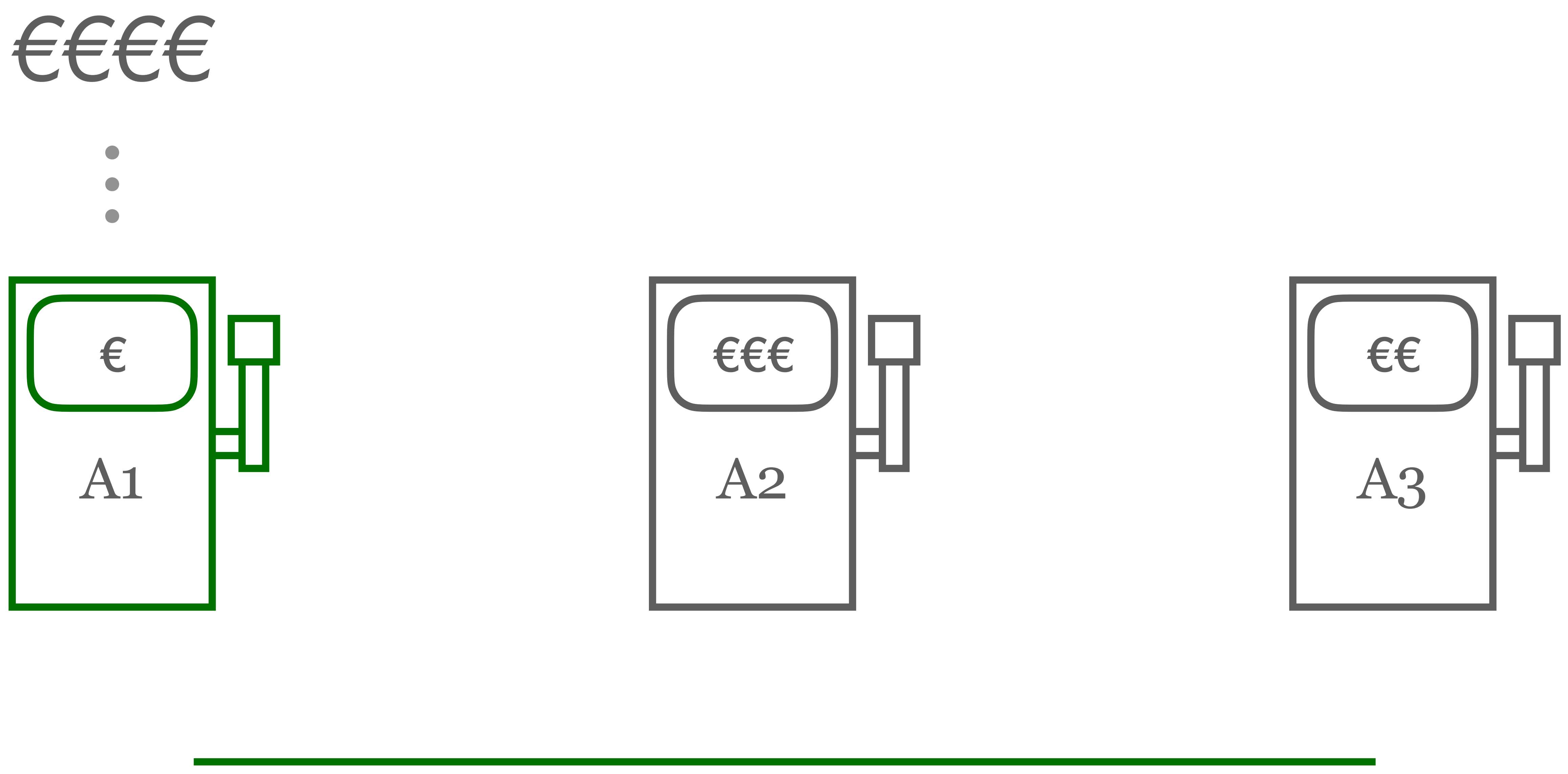
Sample



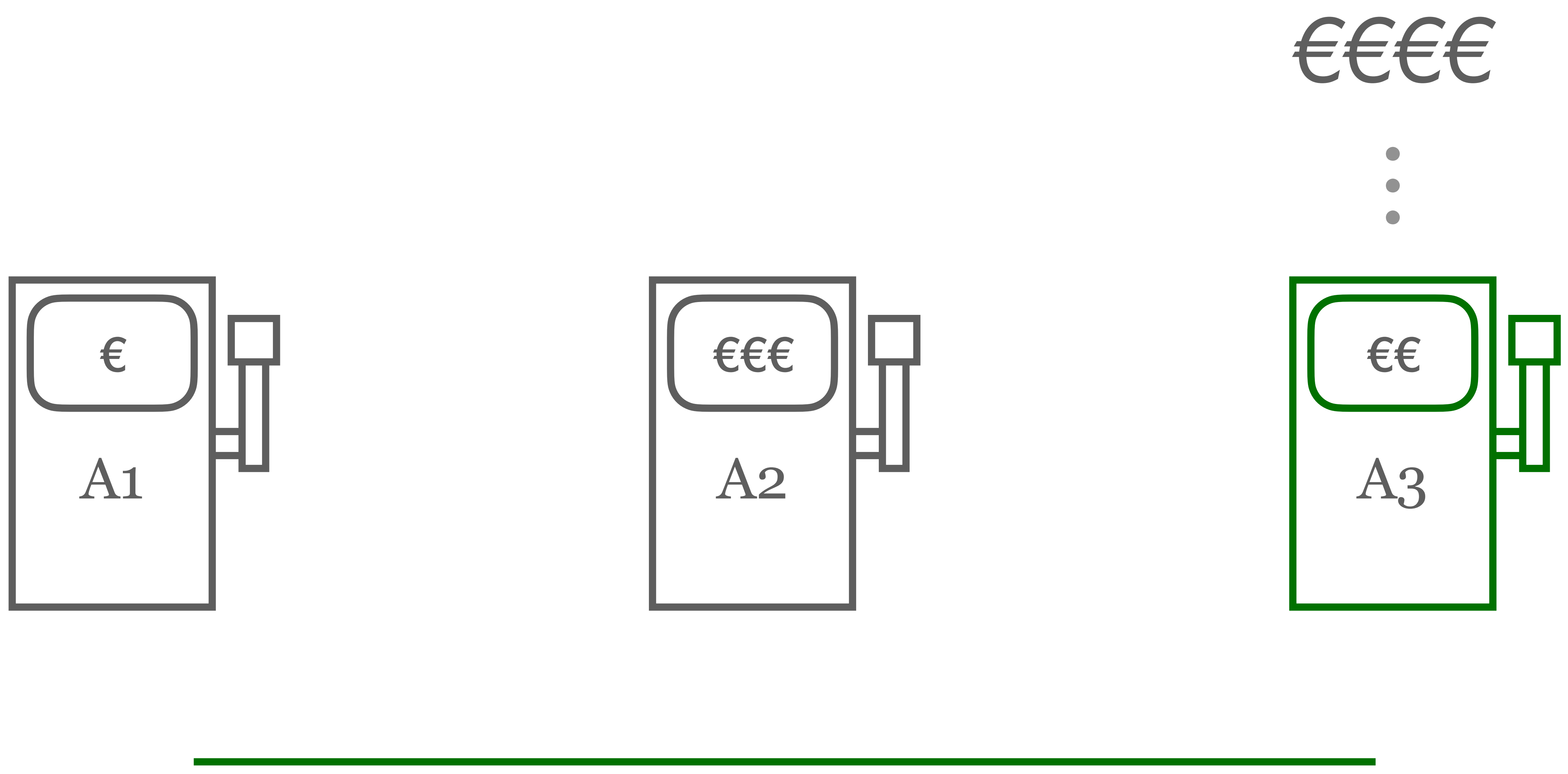
Sample



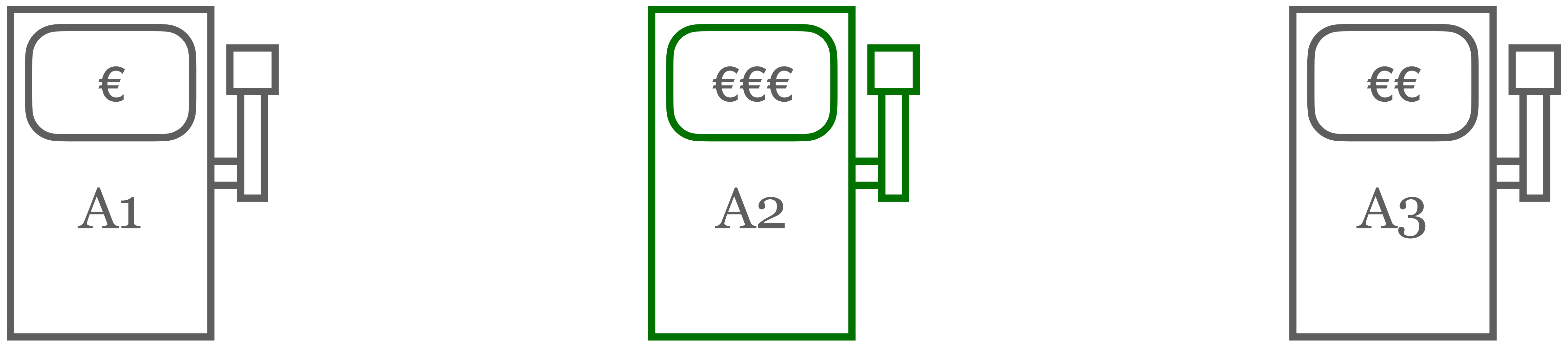
Sample



Sample



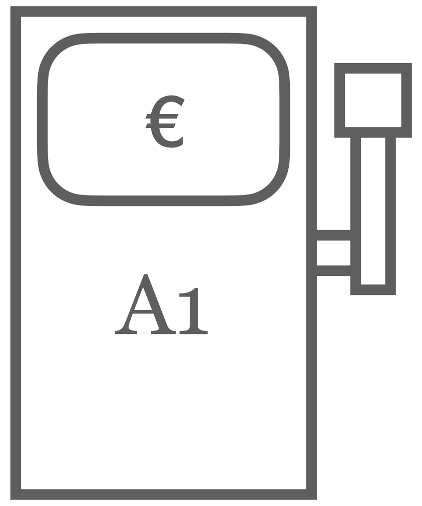
Sample



Find highest expected reward

Upper Confidence Bound.

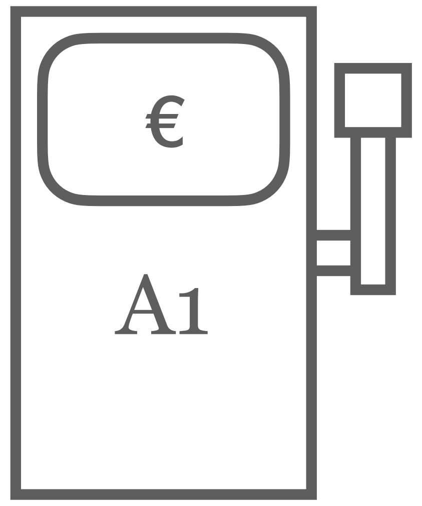
Balance Exploration and Exploitation.



$$\dots \; X_1^1, X_2^1, X_3^1, \; \dots \; X_{N_t^1}^1$$

Sample history

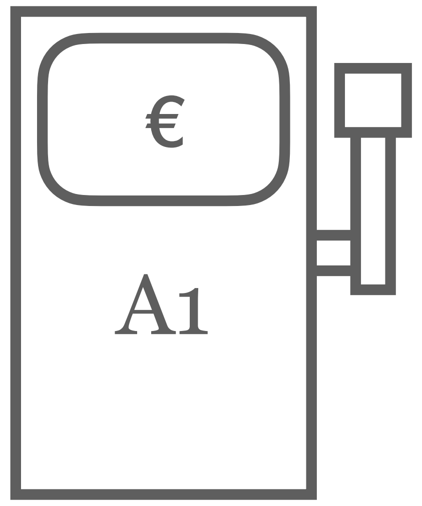
$$\frac{1}{N_t^1}\sum_{s=1}^{N_t^1} X_s^1$$



$$\cdots \; X_1^1, X_2^1, X_3^1, \; \dots \; X_{N_t^1}^1$$

Compute average

$$\frac{1}{N_t^1}\sum_{s=1}^{N_t^1} X_s^1 \pm \sqrt{\frac{C(\sigma_1^2)\log(4t/\delta)}{N_t^1}}$$



$$\ldots\ X_1^1, X_2^1, X_3^1, \ \ldots\ X_{N_t^1}^1$$

Account for uncertainty

$\hat{R}_t^i$ CS_t^i

$$\frac{1}{N_t^i}\sum_{s=1}^{N_t^i} X_s^i \pm \sqrt{\frac{C(\sigma_i^2)\log(4t/\delta)}{N_t^i}}$$

Upper and lower confidence bound

$$\forall i \in [N]:$$

$$\mathbb{P}(\forall t \in \mathbb{N}: R_i \in \hat{R}_t^i \pm \mathrm{CS}_t^i) \geq 1 - \delta$$

Probability bound

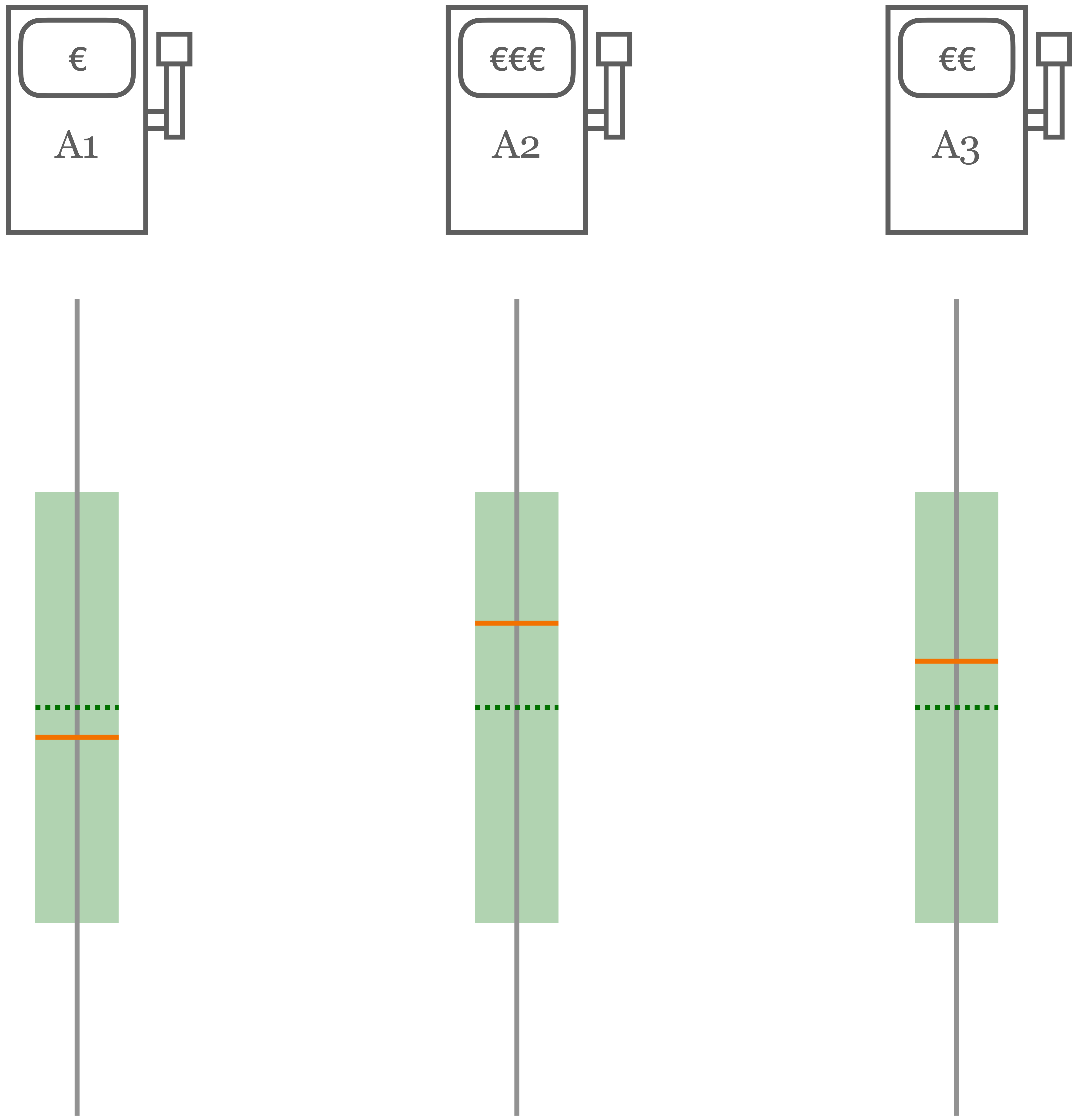
Algorithm.

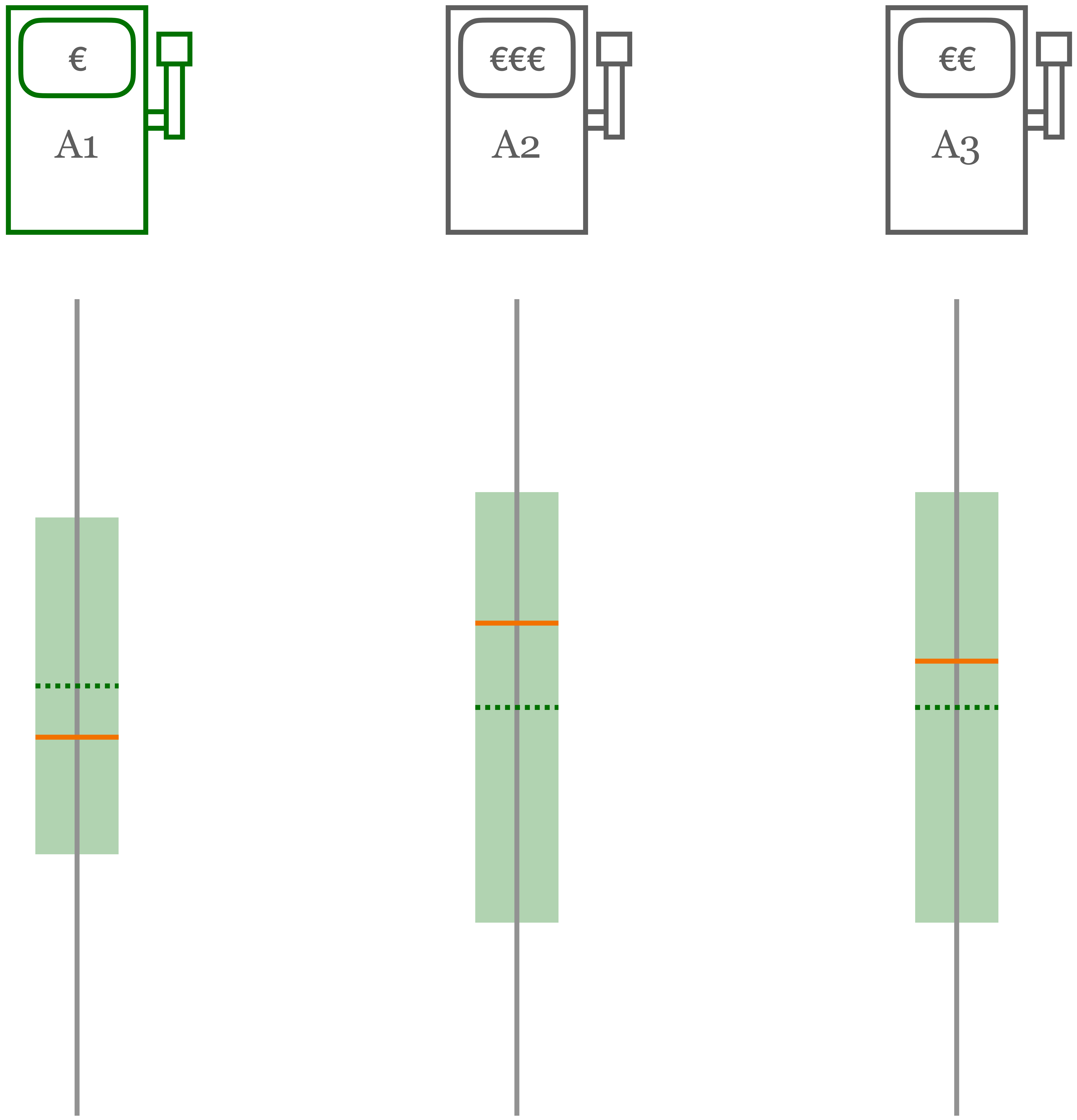
Always choose the arm with the highest UCB.

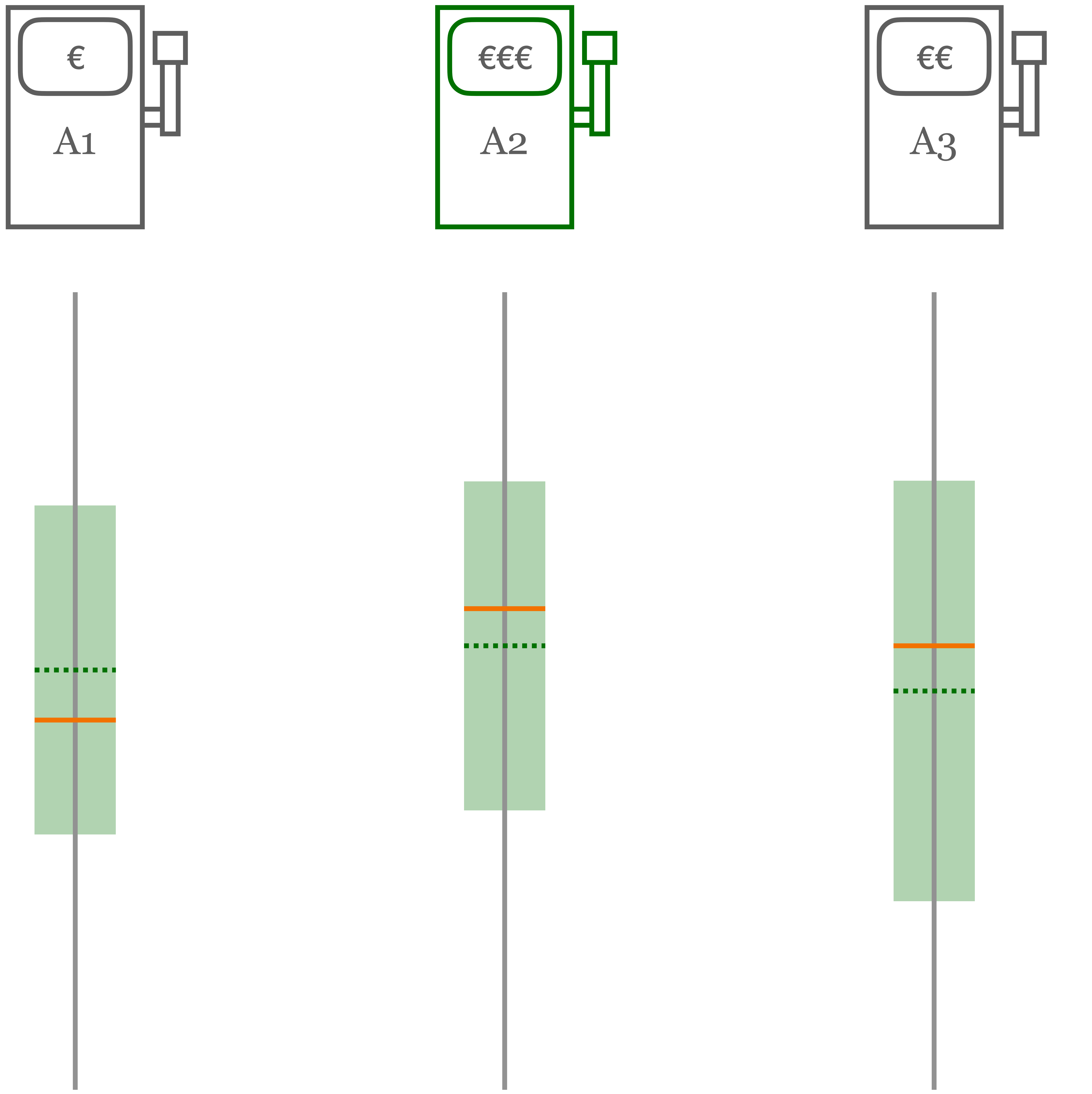
While True :

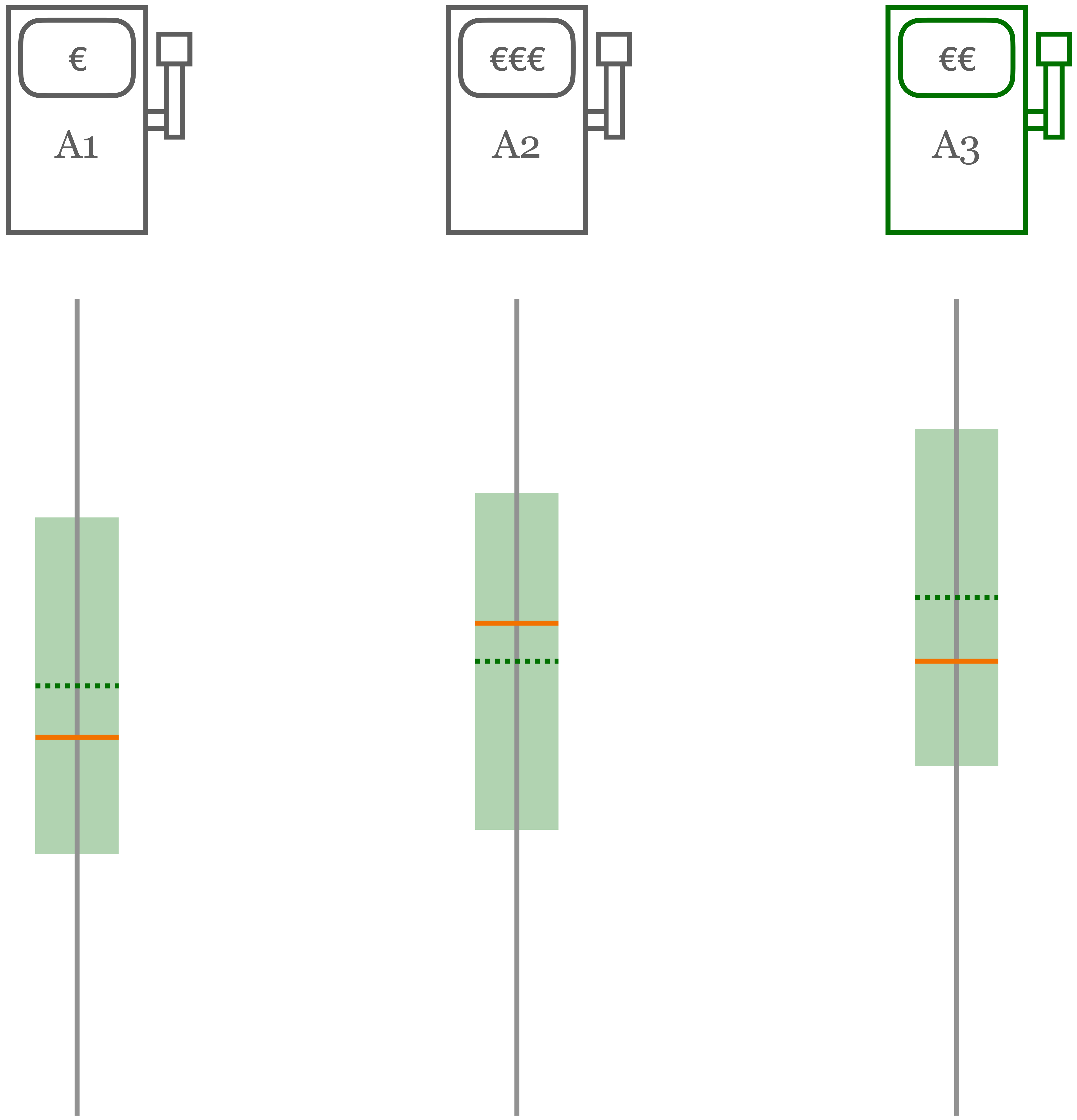
$$I_{t+1} \leftarrow \arg\max_{i \in [N]} (\hat{R}_t^i + \mathrm{CS}_t^i)$$

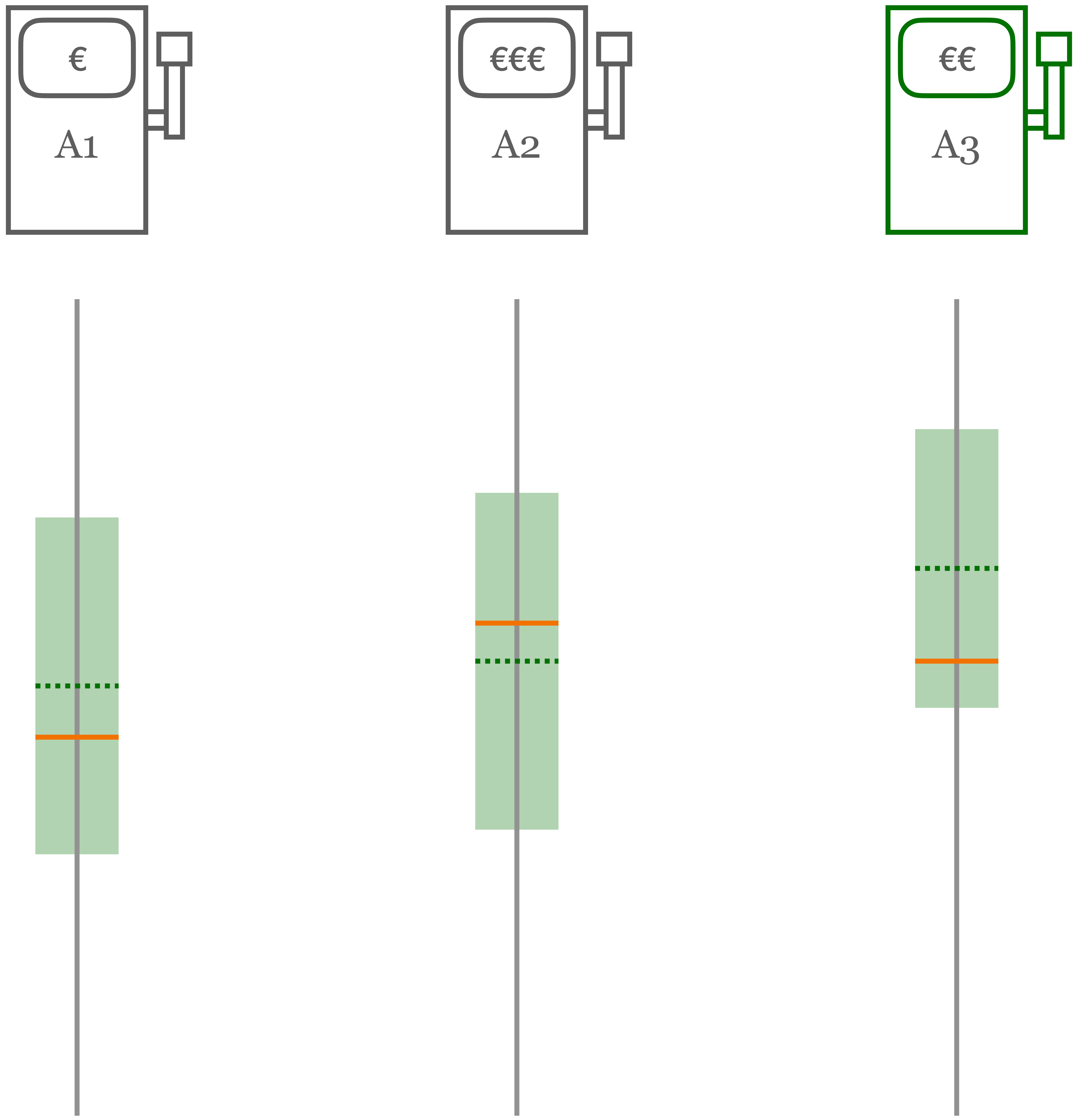
$$X_{t+1}^{I_{t+1}} \sim A_{I_{t+1}}$$

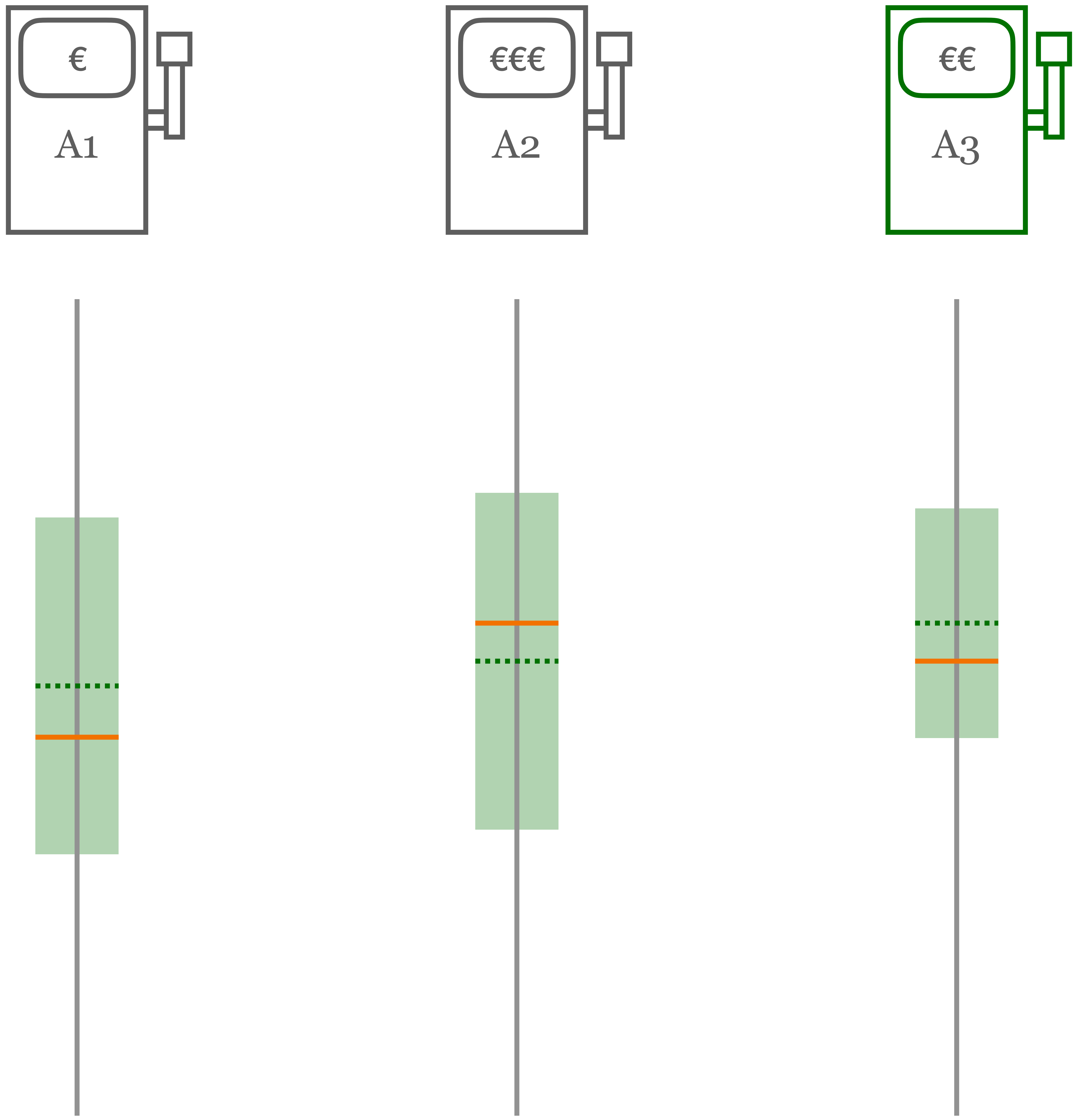

€
A1
€€€
A2
€€
A3

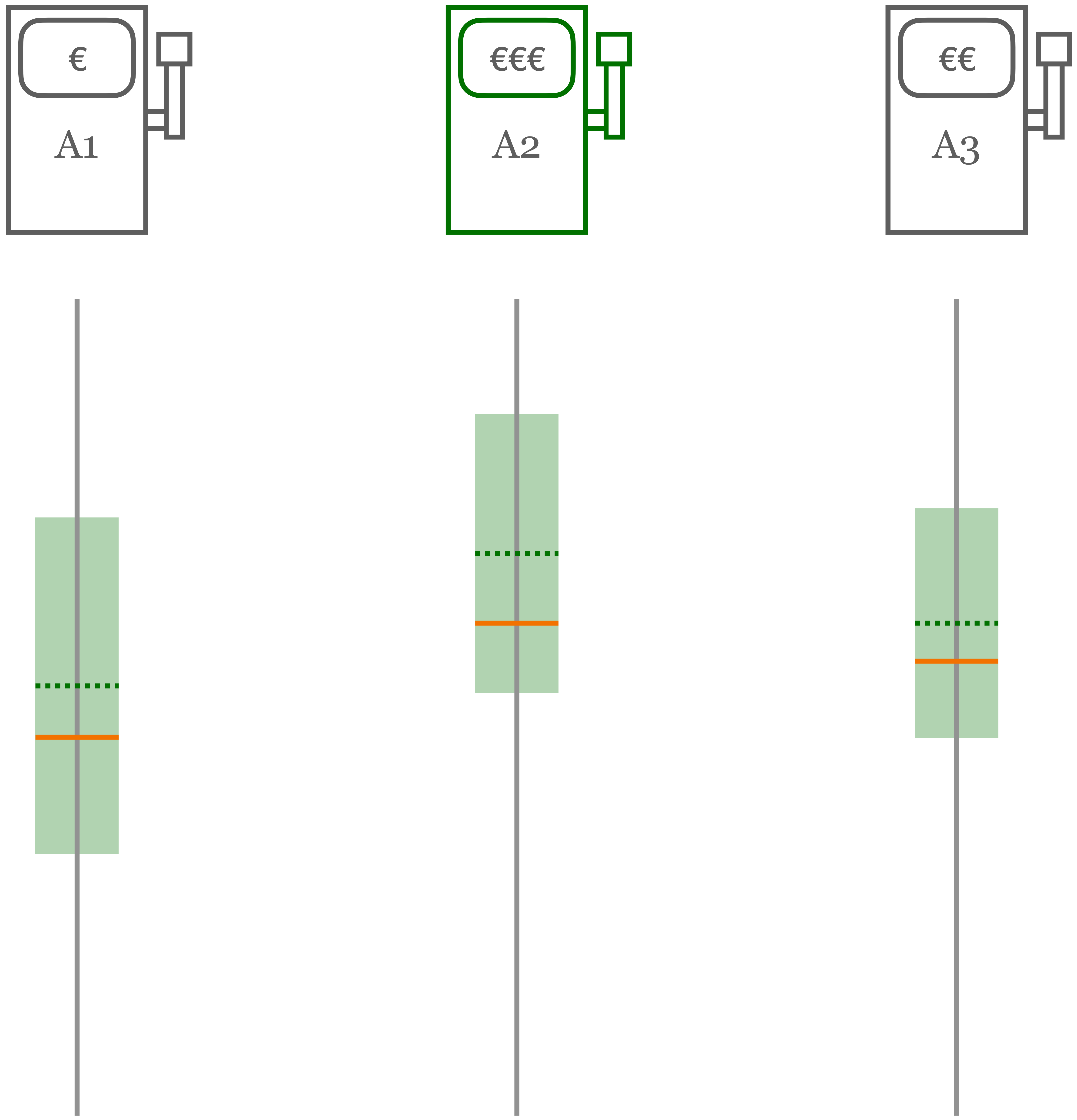

€
A1
€€€
A2
€€
A3


€
A1
€€€
A2
€€
A3


€
A1
€€€
A2
€€
A3


€
A1
€€€
A2
€€
A3


€
A1
€€€
A2
€€
A3


€
A1
€€€
A2
€€
A3

Search Algorithm.

$\textit{Check if: } \forall i \in [N] : R_i \leq 0$

While True :

If $0 < \hat{R}_t^{I_t} - \mathrm{CS}_t^{I_t}$:

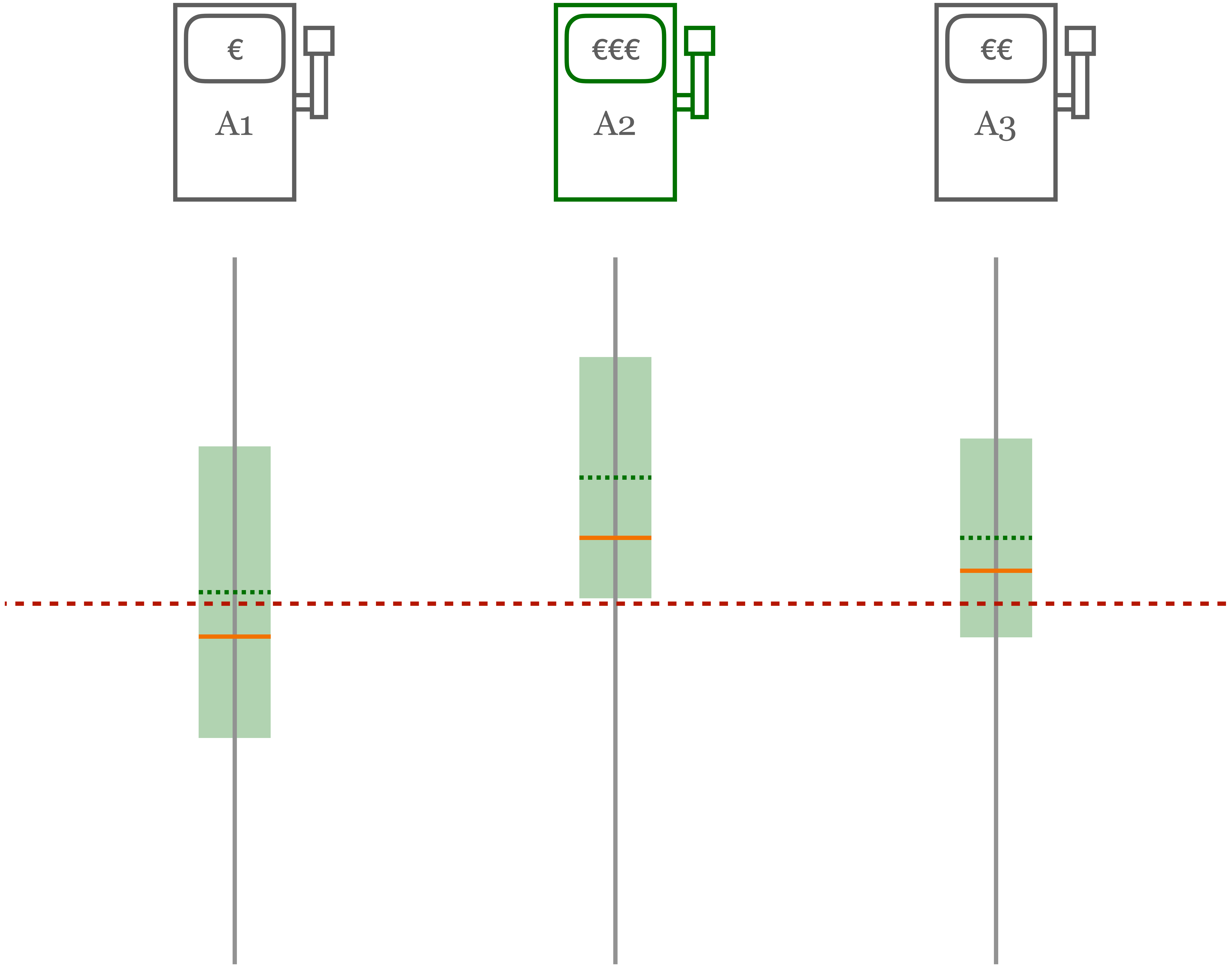
Return False

If $0 \geq \hat{R}_t^{I_t} + \mathrm{CS}_t^{I_t}$:

Return True

$$I_{t+1} \leftarrow \arg\max_{i \in [N]} (\hat{R}_t^i + \mathrm{CS}_t^i)$$

$$X_{t+1}^{I_{t+1}} \sim A_{I_{t+1}}$$


€
A1
€€€
A2
€€
A3

Why?

Because with probability $1 - \delta$*:*

$$R_{i*} \leq \hat{R}_t^{i*} + \mathrm{CS}_t^{i*} \leq \hat{R}_t^{I_t} + \mathrm{CS}_t^{I_t}$$

and

$$\hat{R}_t^{I_t} - \mathrm{CS}_t^{I_t} \leq R_{I_t}$$

Lipschitz-Bandits

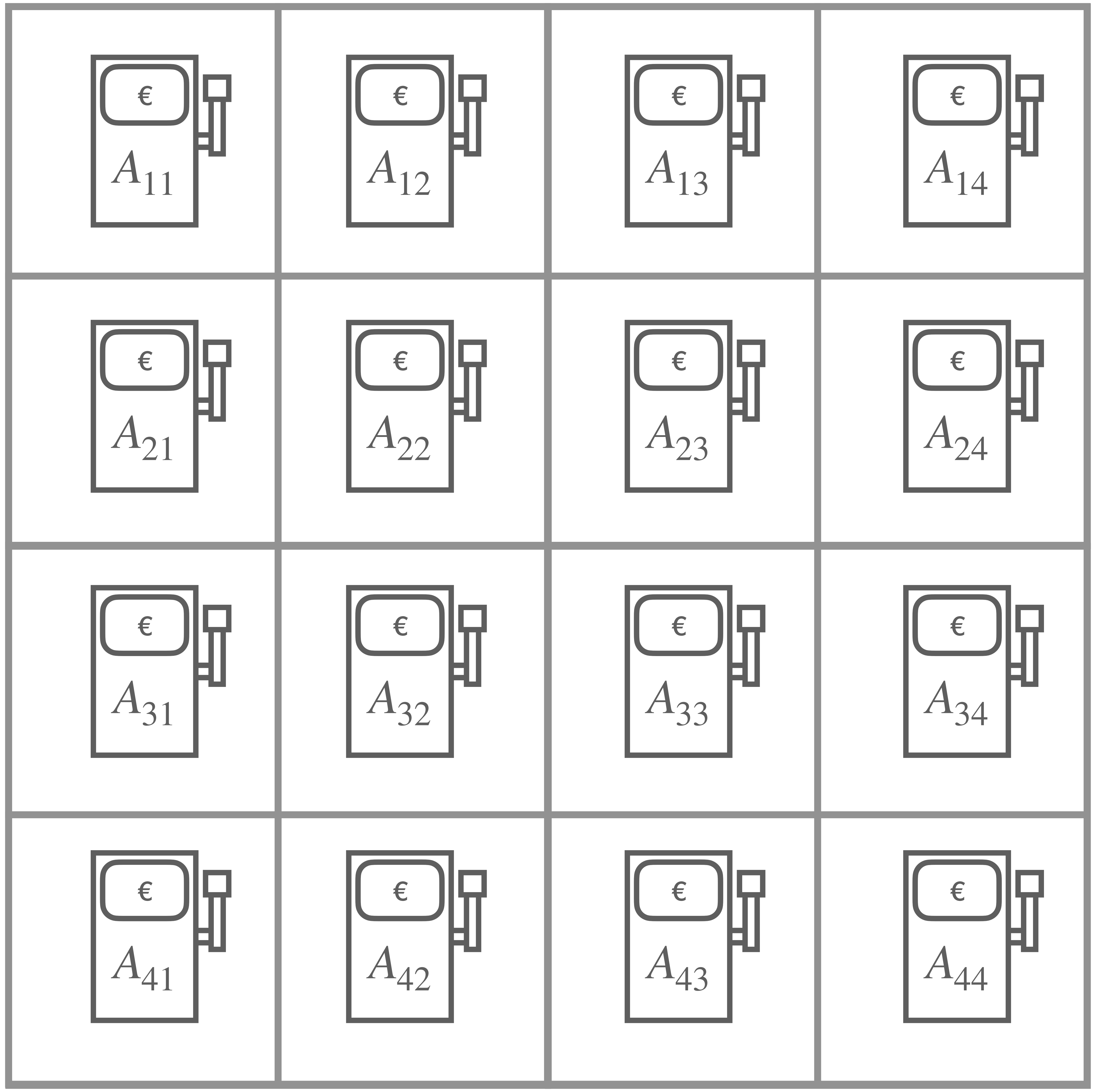
With adaptive Gridding.

Problem Statement:

Given a problem instance $\mathcal{I}$ find an ***algorithm*** $\mathscr{A}$
with knowledge of $(\gamma_P, \gamma_f, c_f, \delta)$ *and sample access of* (P, f), *s.t.*

$$\mathscr{A}(\delta) \iff \sup_{x \in \mathscr{X} \setminus \mathscr{T}} R_x \leq 0$$

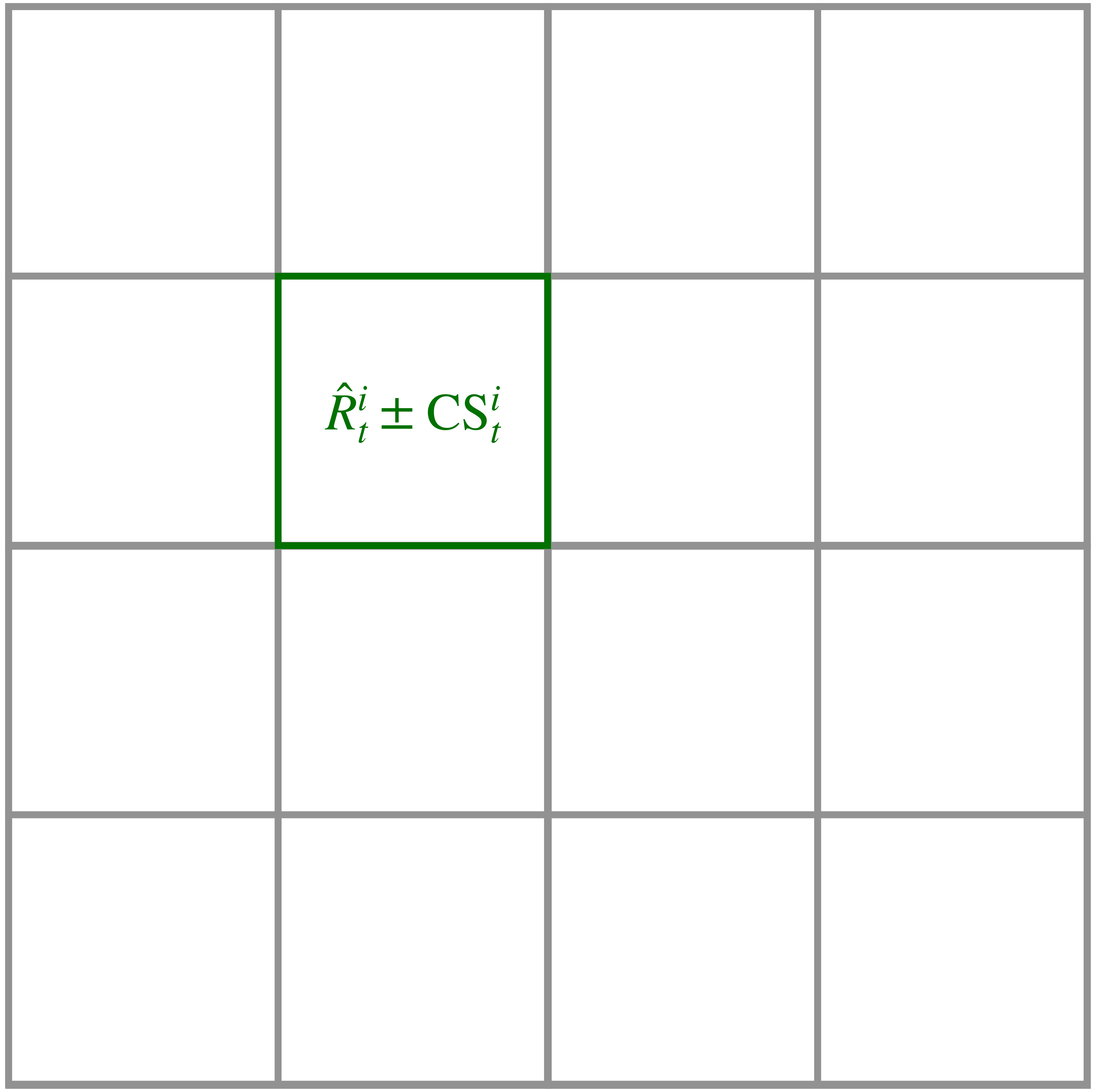
with ***probability*** $1 - \delta$ *upon termination.*

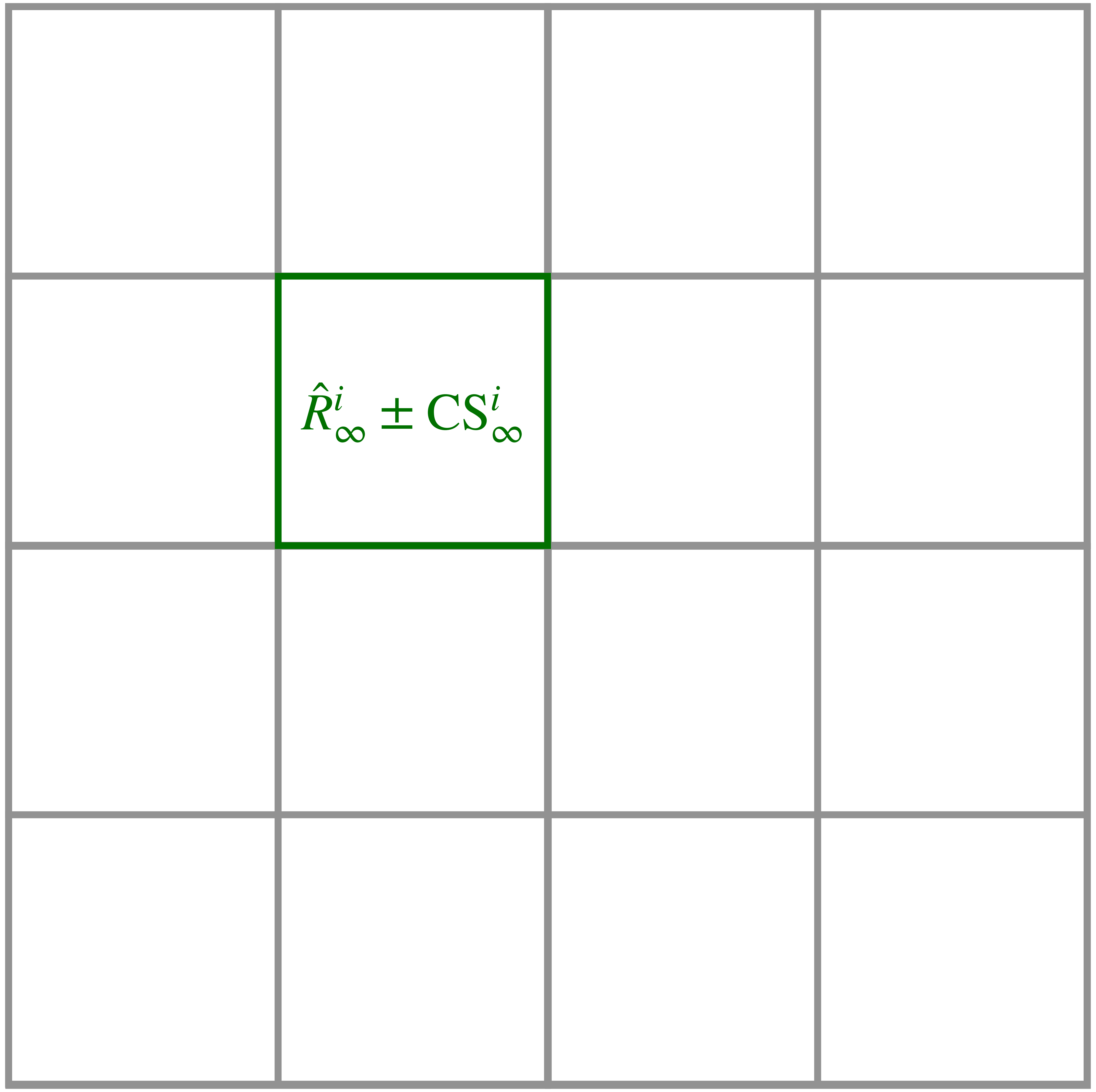

€
A_{11}
€
A_{12}
€
A_{13}
€
A_{14}
€
A_{21}
€
A_{22}
€
A_{23}
€
A_{24}
€
A_{31}
€
A_{32}
€
A_{33}
€
A_{34}
€
A_{41}
€
A_{42}
€
A_{43}
€
A_{44}

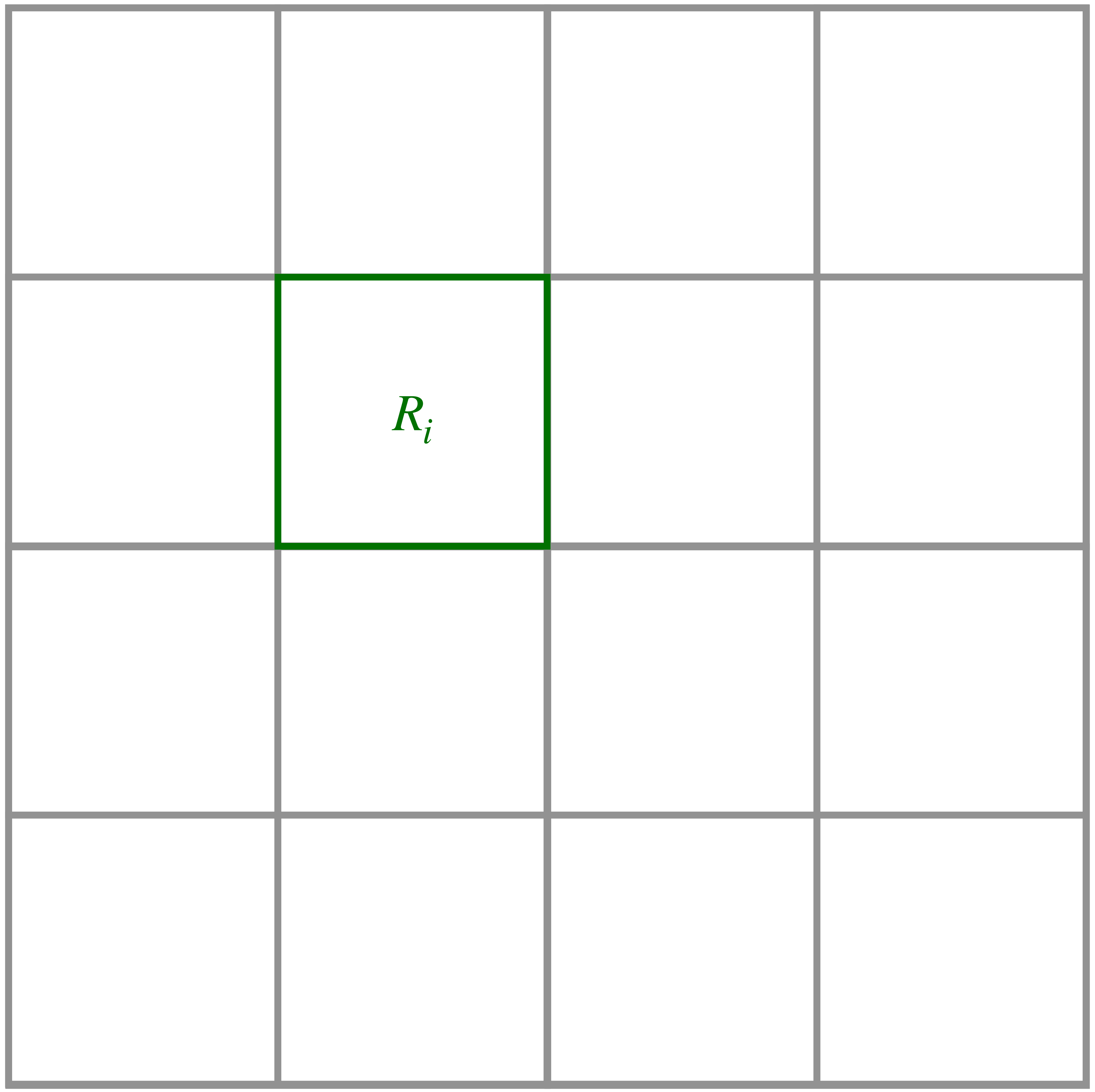
$$\forall t \in \mathbb{N} \forall i \in \mathcal{I}_t : Y \sim X_t^i P :$$

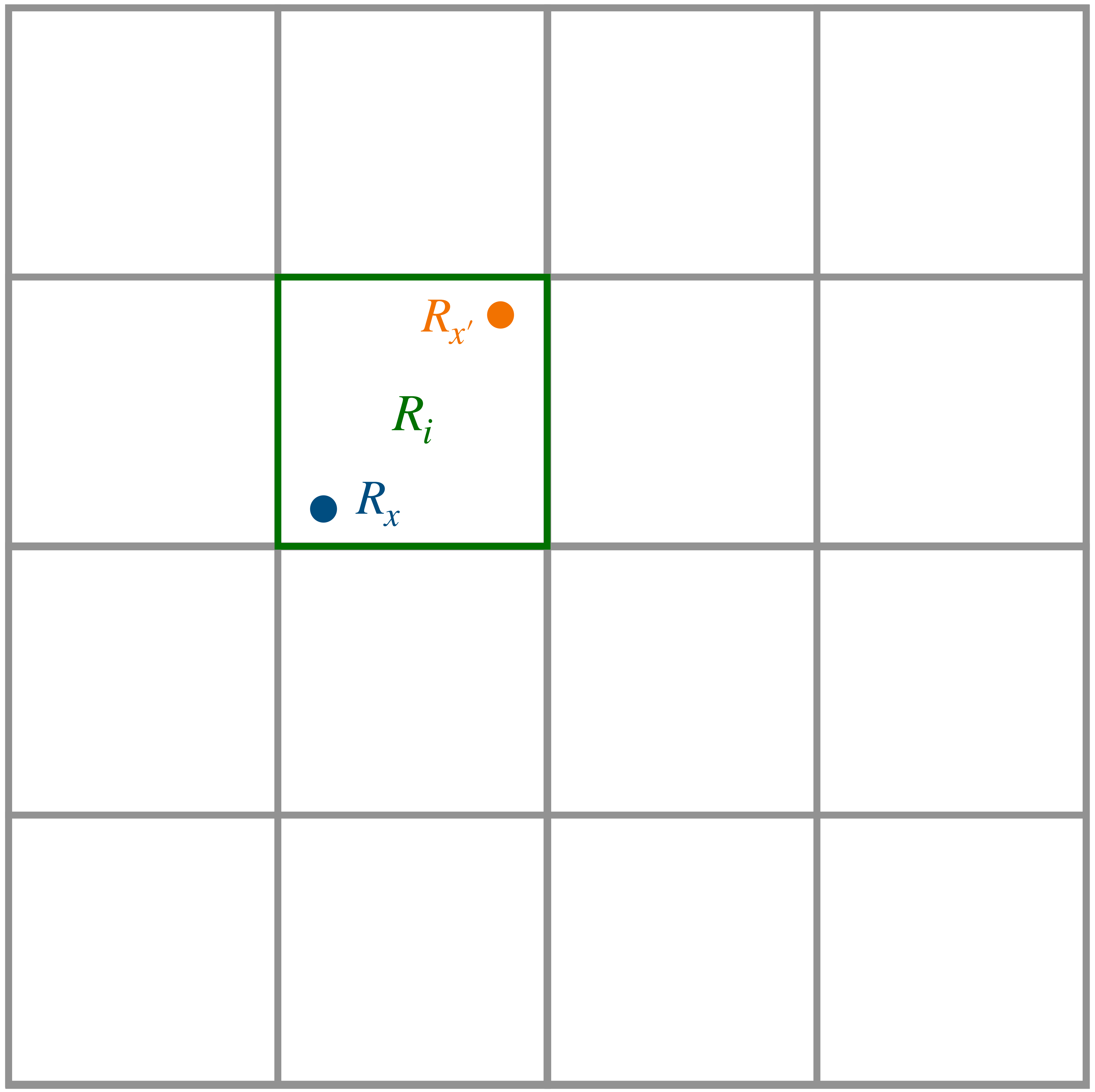
$$f(Y_t^i) - f(X_t^i) + \varepsilon$$

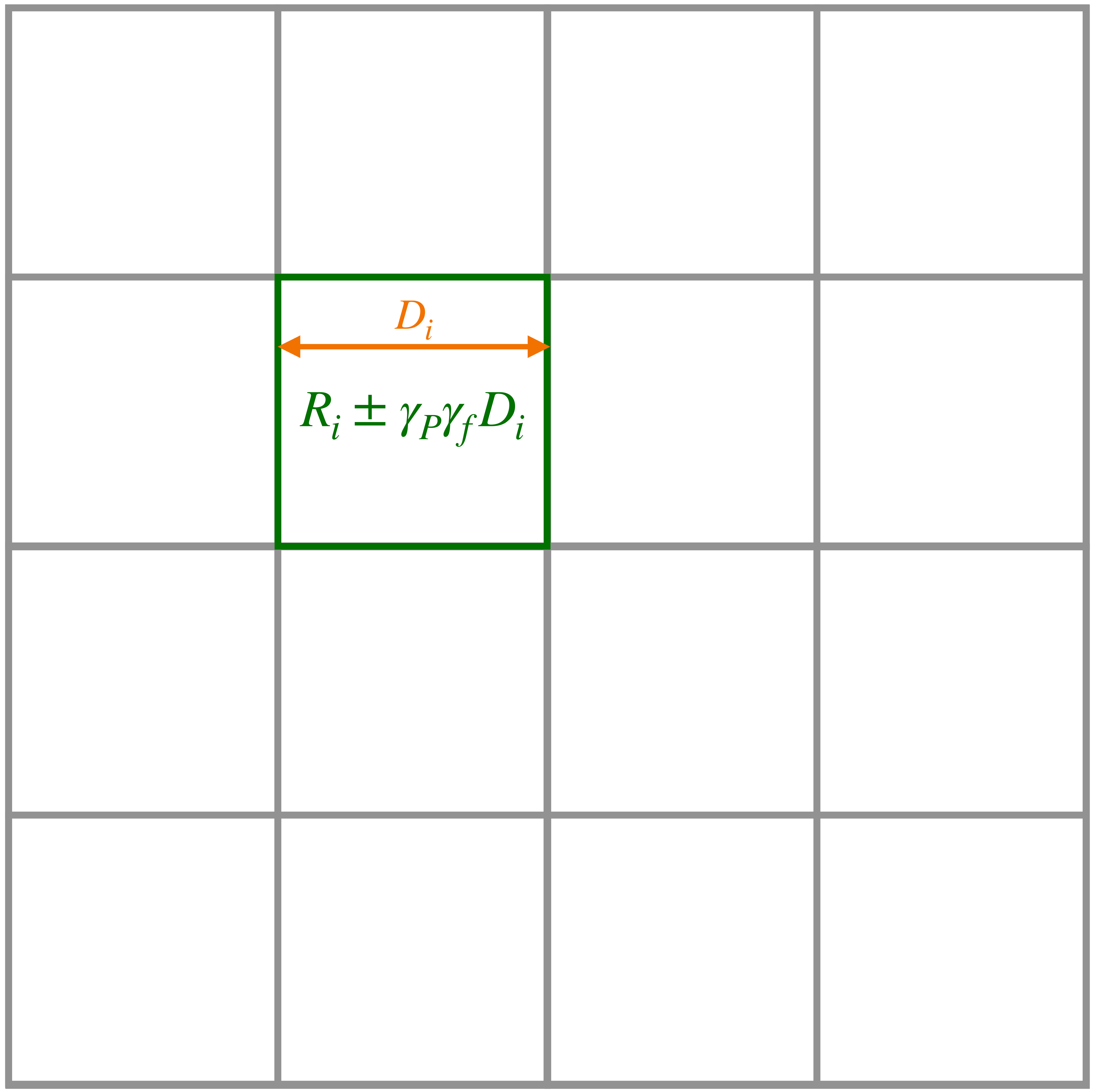
Observed Reward


$\hat{R}_t^i \pm \mathrm{CS}_t^i$


$\hat{R}^i_\infty \pm \mathrm{CS}^i_\infty$


R_i


$R_{x'}$
R_i
R_x


D_i
$R_i \pm \gamma_P \gamma_f D_i$

$$\forall i \in \mathcal{I}\ \forall t \in \mathbb{N}:$$

$$\hat{R}_t^i \pm \mathrm{CS}_t^i + \gamma_P \gamma_f D_i$$

Lipschitz LCB & UCB

$\sigma_i := c_f$ DE_t^i

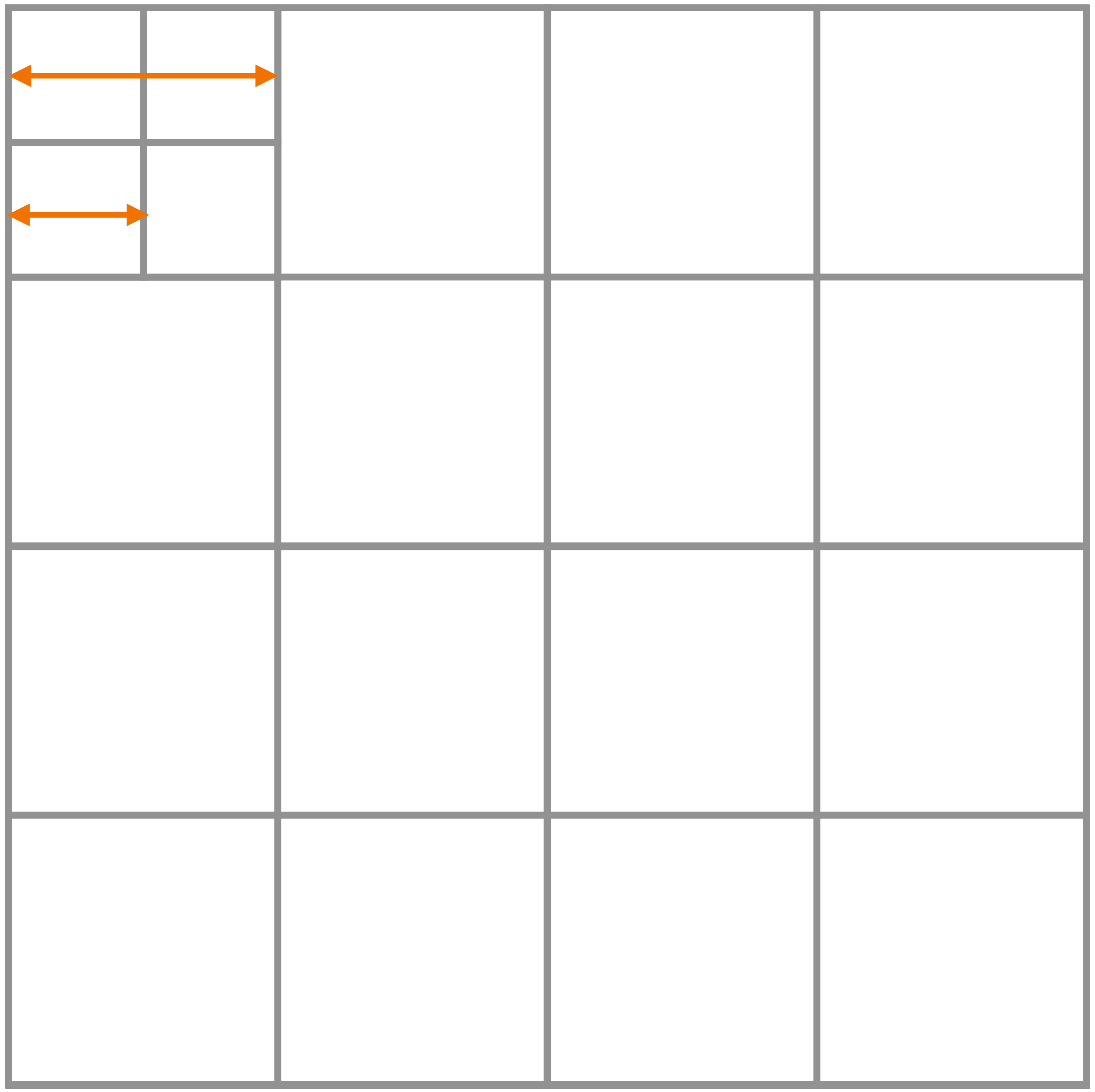
$$\hat{R}_t^i \pm \mathrm{CS}_t^i + \gamma_P \gamma_f D_i$$

Lipschitz LCB & UCB

Zoom into…

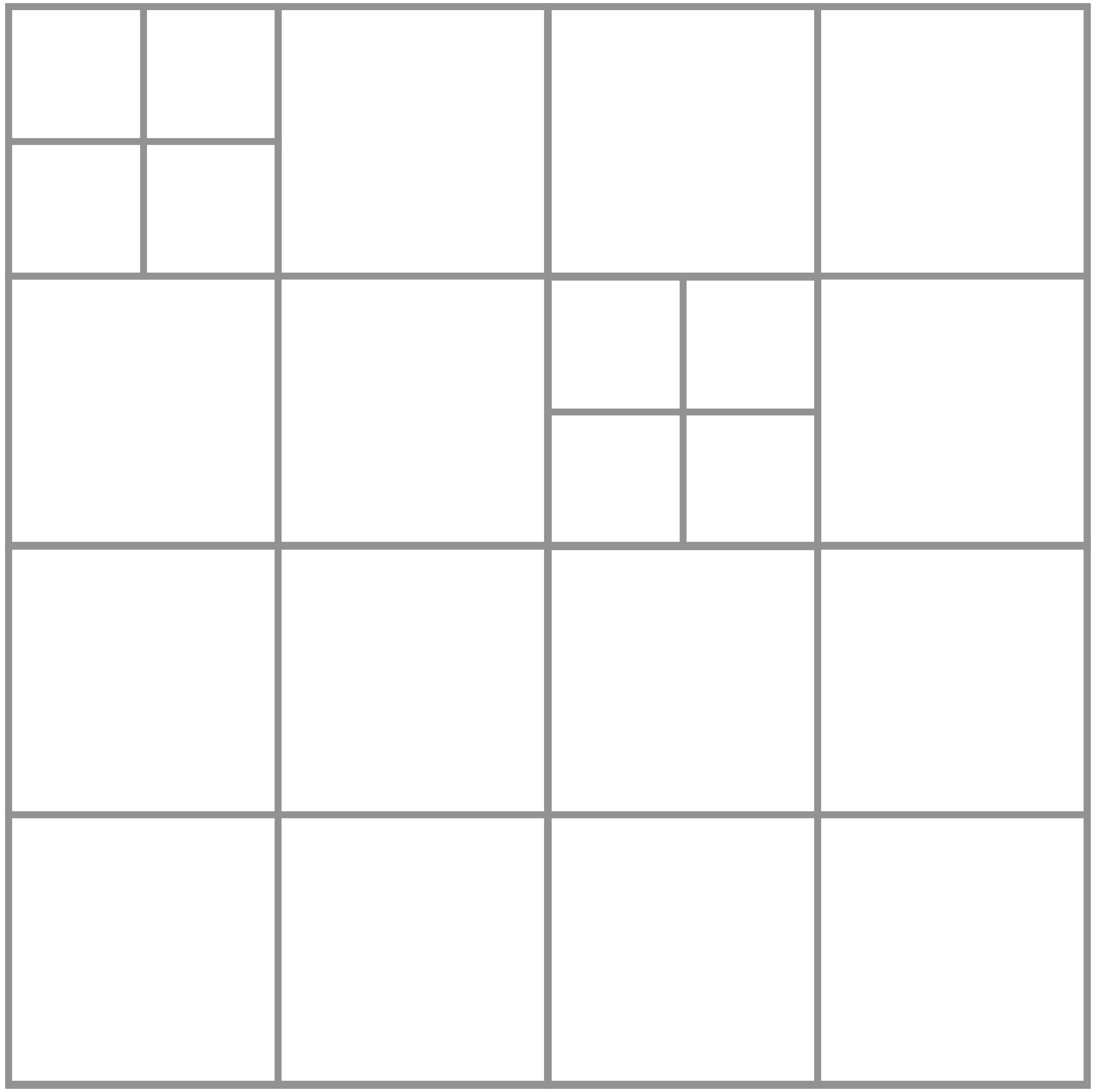
…questionable areas.

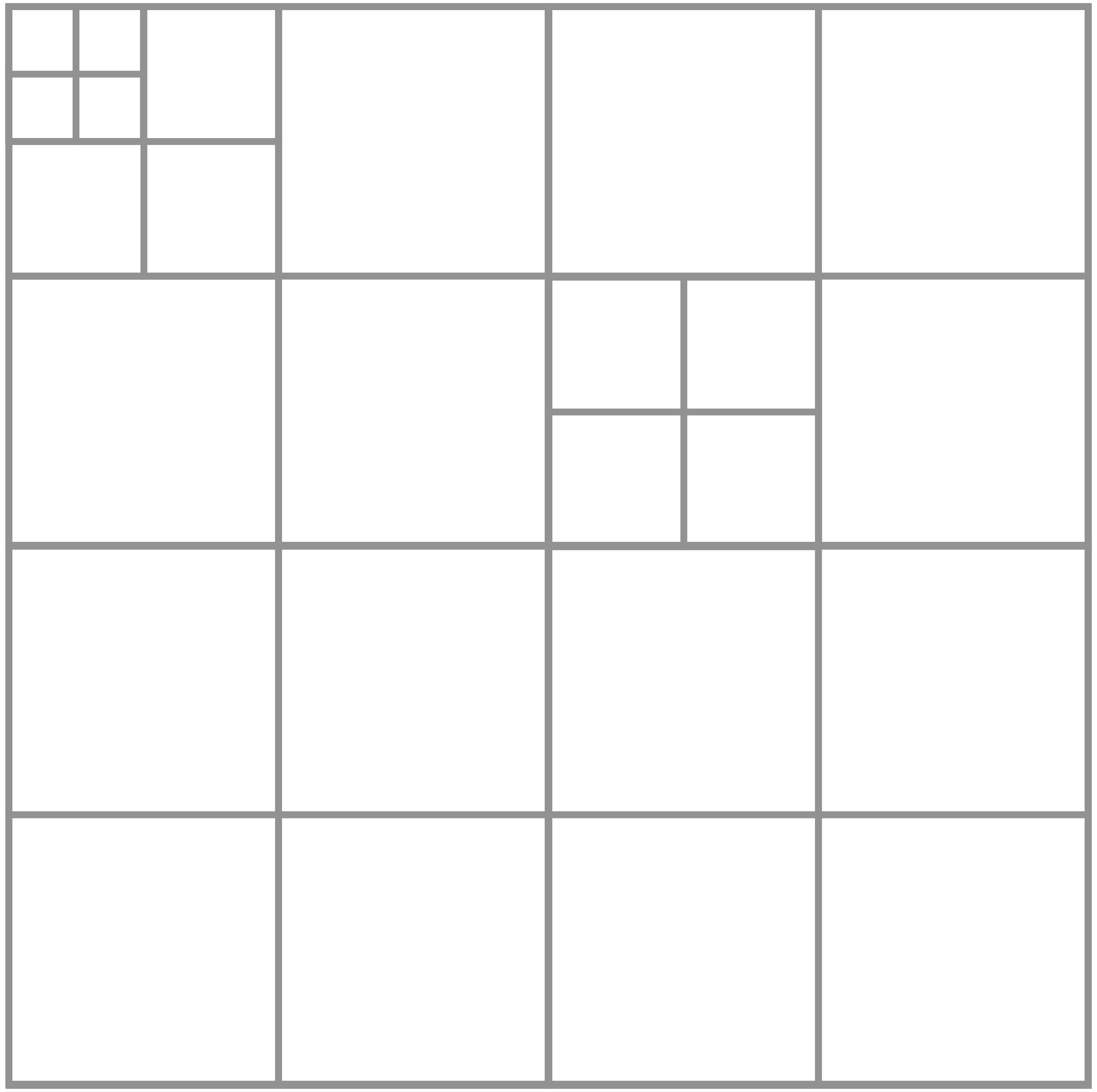
≈ 0			
		≈ 0	
$\ll 0$			

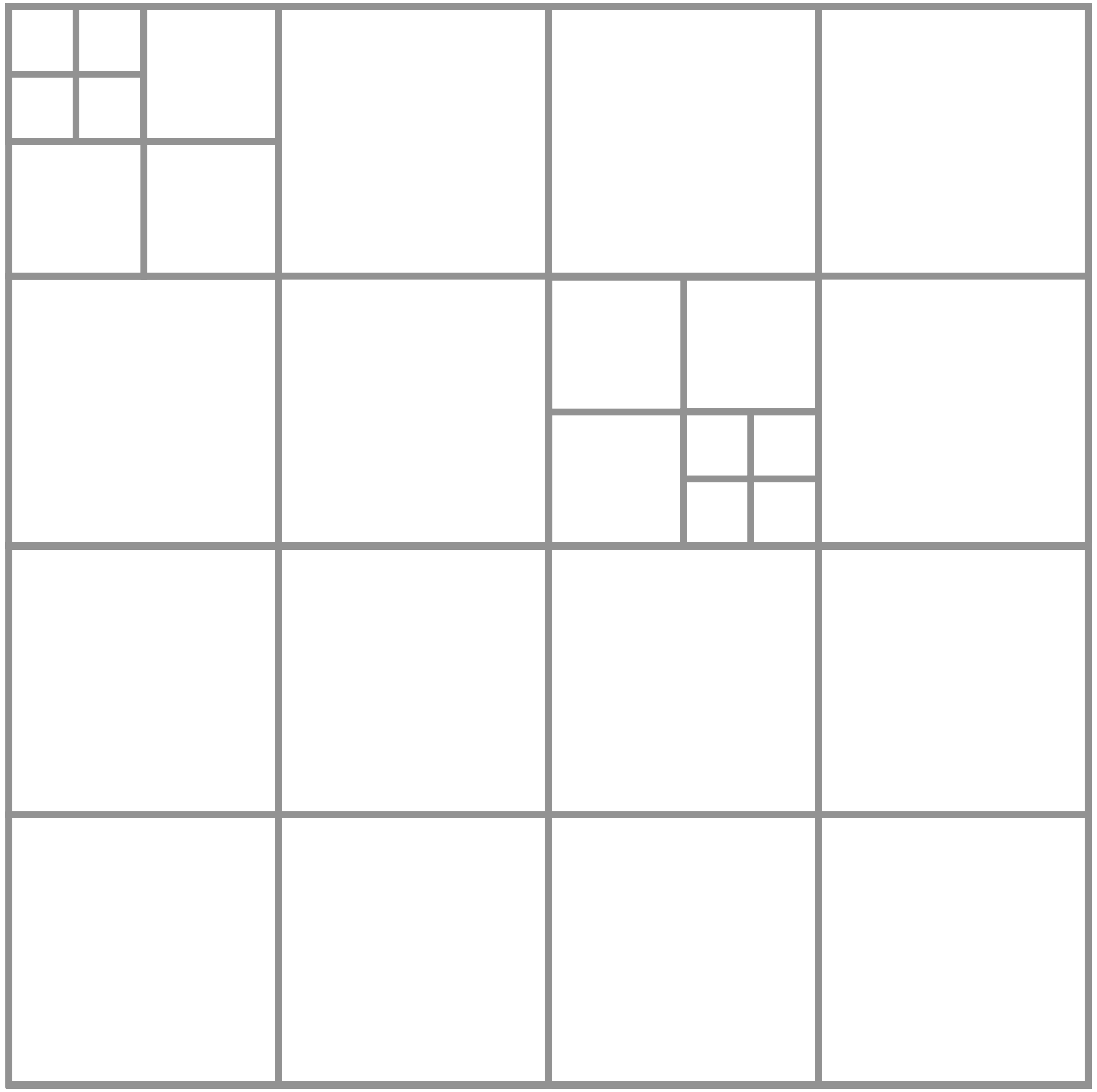


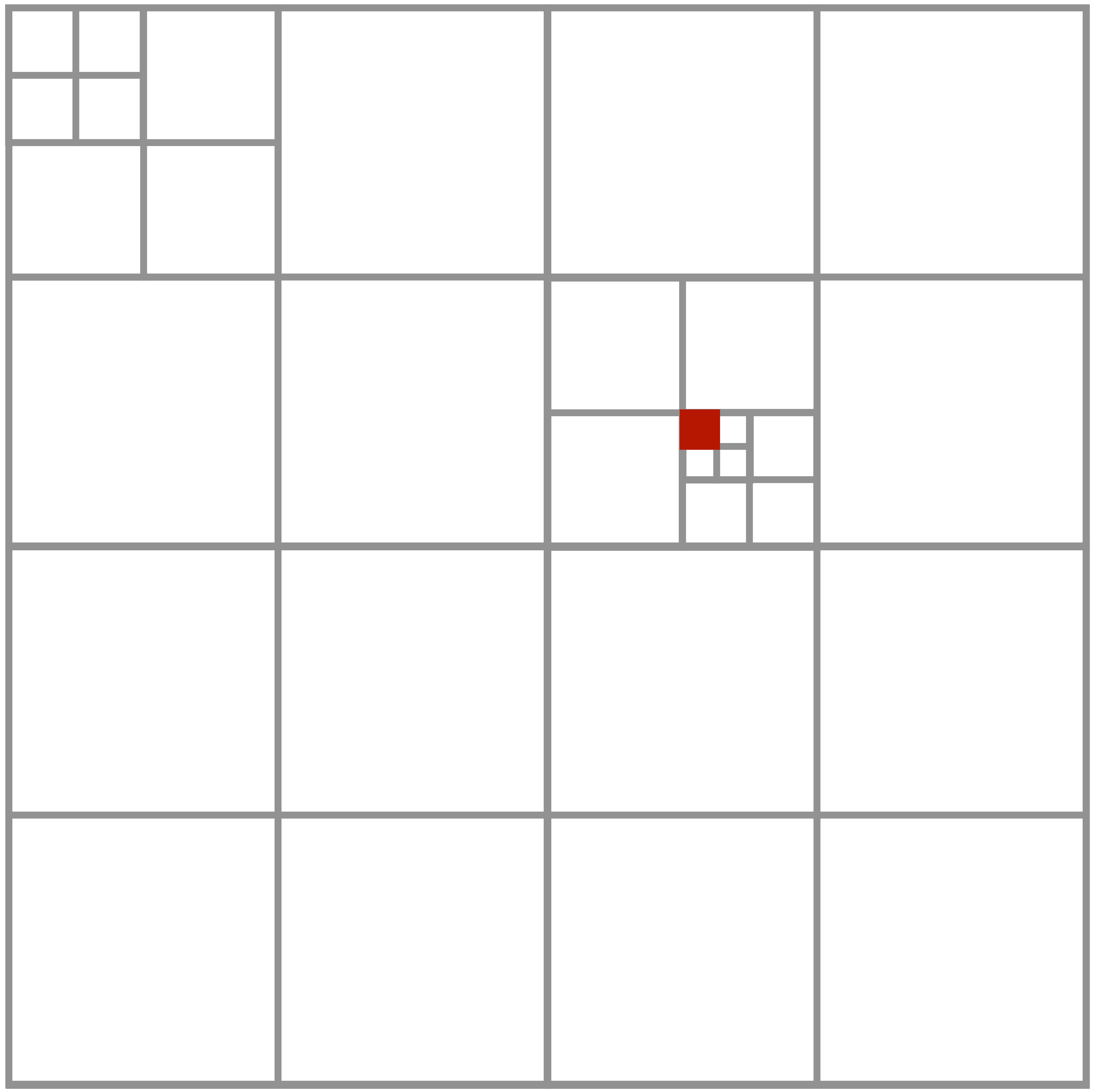
$$\hat{R}_t^i \pm \mathrm{CS}_t^i \uparrow + \ \mathrm{DE}_t^i \downarrow$$

Splitting









Balance statistical and discretisation exploration.

Hopefully, reduce the number of grid cells while executing the MAB algorithm.

Statistical Checker…

…for probabilistic termination proofs.

Discussion.

Contributions, limitations, and improvements.

Certificate Verification.

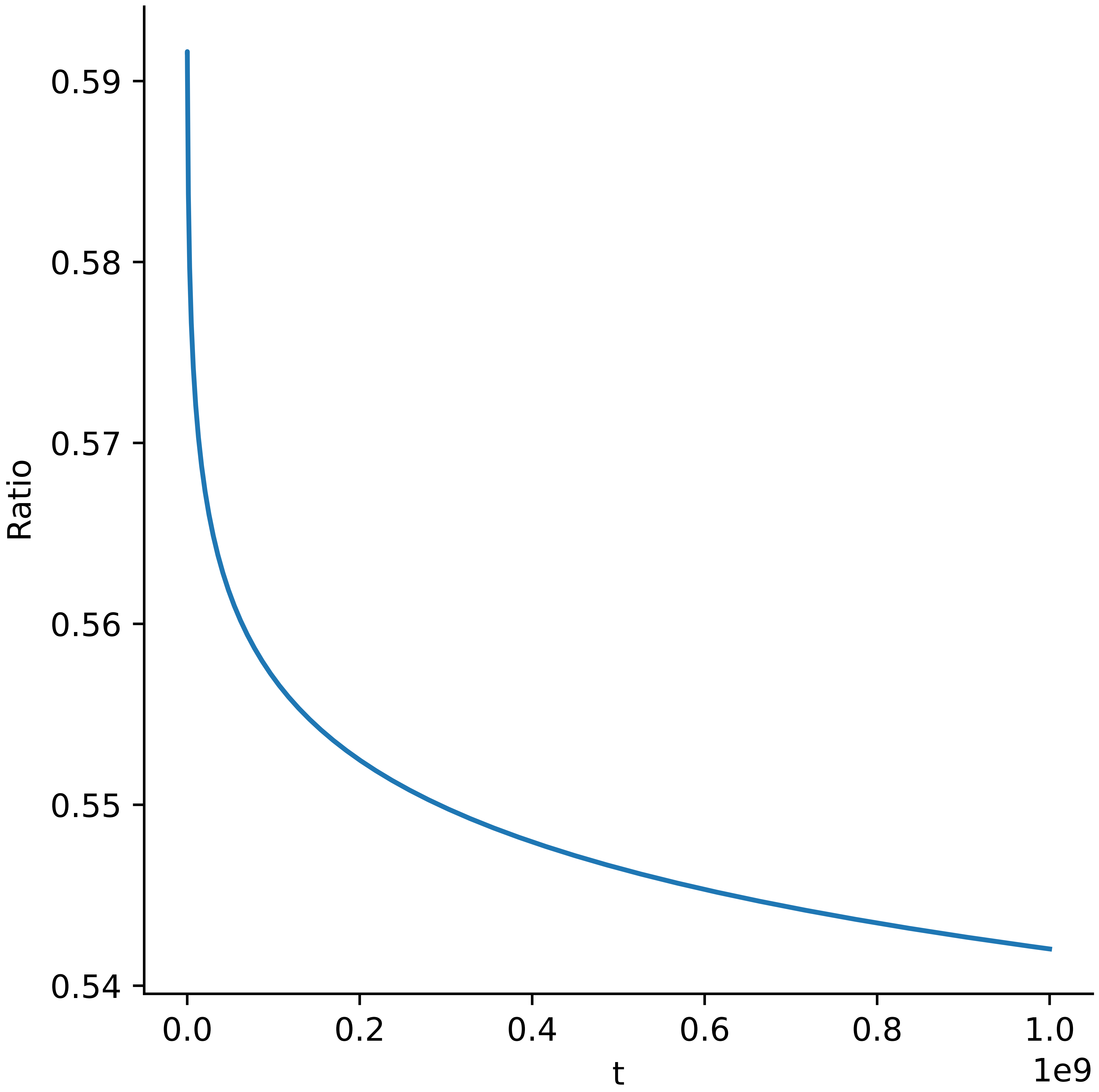
We noticed that this can be done using MAB.
In the process we reduced the required assumptions.

f	:	Known Neural Network	$<$	Known Lipschitz Constant + Bounded + Sample Access
P	:	Known Dynamics	$<$	Known Lipschitz Constant + Sample Access
Yes/ No	:	Sure Guarantee	$>$	Probabilistic Guarantee

Lipschitz MAB.

Improved on existing works.
(Tentative)

CS	:	$\sqrt{\frac{4\sigma_i^2 \log(4t/\delta)}{N_t^i}}$	$<$	$\sqrt{\frac{3.3\sigma_i^2(2\log\log N_t^i + \log(2/\delta))}{N_t^i}}$
CS Comp.	:	Linear w.r.t. the grid cells.	$<$	Constant.
Loop	:	Linear w.r.t. the grid cells.	$<$	Logarithmic w.r.t. the grid cells.


0.59
0.58
0.57
0.56
0.55
0.54
Ratio
0.0
0.2
0.4
0.6
0.8
1.0
t
1e9

Limitations.

Requires relatively tight Lipschitz constants.

Runtime increases if reward is close to 0.

Improvements.

Parallelisation: Search sub-spaces independently.
Soft-gridding: Use information from neighbouring grid cells.
Adaptive Bounds: Use empirical variance.
Local Lipschitz Constant: Compute or estimate.

Related Works.

A bit of context.

MAB Algorithm.

Kleinberg et.al. (2008): Zooming algorithm;
Wang et.al. (2019): Adaptive gridding + bound;
Jamieson et.al. (2014): Close Gap, $\log t \to \log\log t$;
Howard et.al. (2021): Predictable process, $t \to N_t$.

Summary:

We developed a MAB based verification procedure to validate probabilistic termination proofs with high-probability and improved on existing MAB algorithms.