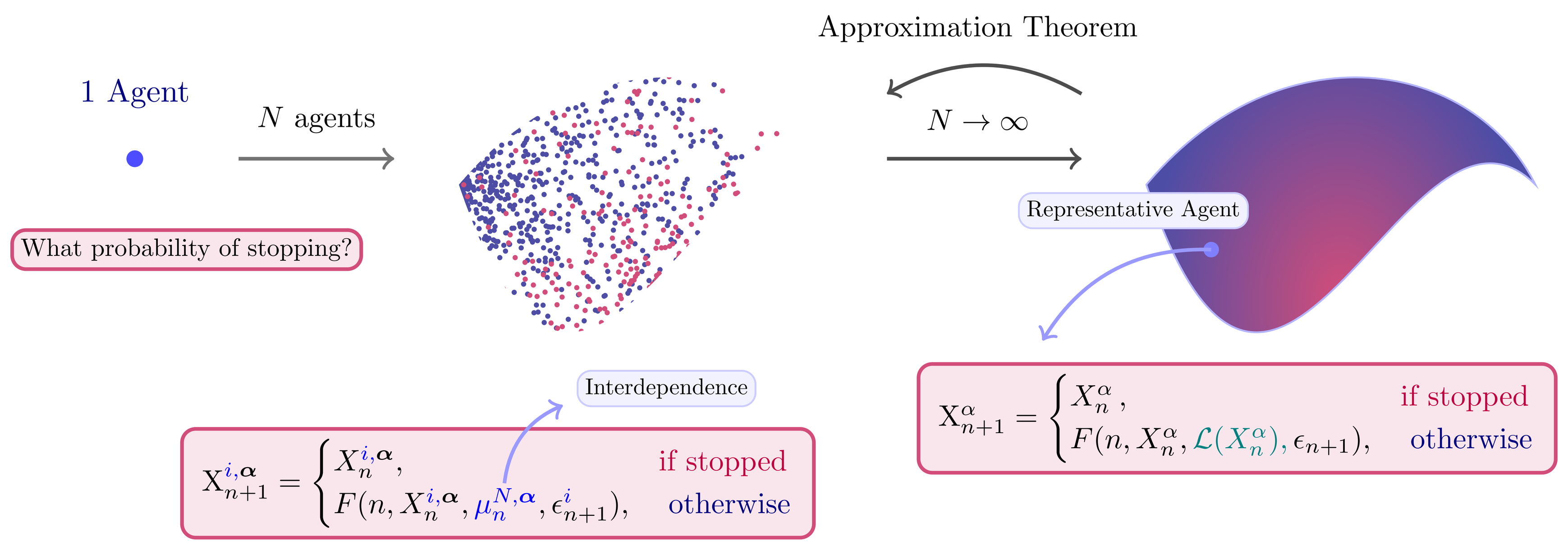
上海纽约大学
NYU SHANGHAI
Georgia Tech Machine Learning Center

Learning to Stop: Deep Learning for Mean Field Optimal Stopping

Lorenzo Magnino*,[1] Yuchen Zhu*,[2] Mathieu Laurière[1]

[1]NYU Shanghai, [2]Georgia Institute of Technology

Paper



A Quick Summary

Target Problem:

- Find practical algorithms for solving Optimal Stopping of Mean-field dynamics of discrete-time and finite state space

Importance:

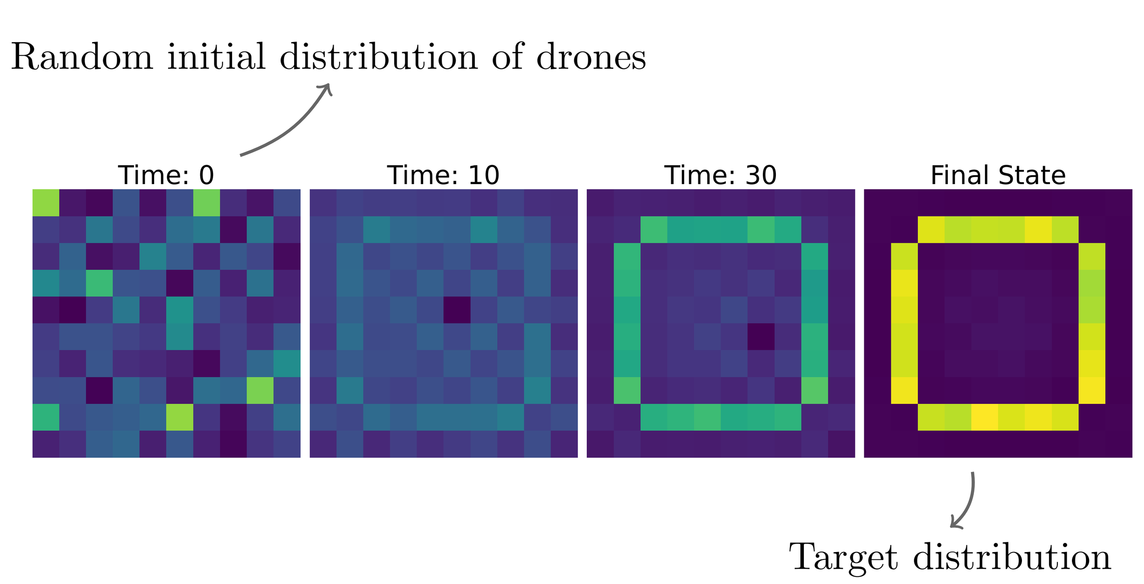
- A proxy for solving Multi-agent OS due to law of representative agent
- Applications: option pricing, swarm robotics, etc

Approach:

- Introduce extra extended state to signal the decision status of each agent
- Relate Mean-field Optimal Stopping to Mean-field Control to build theory

Results:

- Establish $\mathcal{O}(1/\sqrt{N})$ approximation error between N-agent MAOS and MFOS
- Propose two algos: Direct Approach and Dynamic Programming Principle
- Consider two type of policies: synchronous and asynchronous
  - synchronous: agent decides based on population distribution (suboptimal in many cases)
  - asynchronous agent decides based on **current state**, **time** and population



Mean-field Optimal Stopping

We are concerned with finding control α for the cost:

Multi-agent: $$J^N(\alpha^1, \dots, \alpha^N) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \Phi(X^{i,\alpha}_{\tau^i}, \mu^{N,\alpha}_{\tau^i})\right]$$

Mean-field: $$J(\alpha) = \mathbb{E}\Big[\Phi(X^{\alpha}_{\tau}, \mathcal{L}(X^{\alpha}_{\tau}))\Big].$$

Agent dynamic with extended state:

$$\begin{cases} X^{\alpha}_0 \sim \mu_0, \qquad A^{\alpha}_0 = 1 \\ \alpha_n \sim \pi_n(\cdot|X^{\alpha}_n) = \mathrm{Ber}(p_n(X^{\alpha}_n)); \;\; A^{\alpha}_{n+1} = A^{\alpha}_n(1-\alpha_n) \\ X^{\alpha}_{n+1} = \begin{cases} F(n, X^{\alpha}_n, \mathcal{L}(X^{\alpha}_n), \epsilon_{n+1}), \text{if } A^{\alpha}_n(1-\alpha_n) = 1 \\ X^{\alpha}_n, \qquad \text{otherwise.} \end{cases} \end{cases}$$

Mean-field dynamic evolution:

$$\begin{cases} \nu^p_0(x,0) = 0, \quad \nu^p_0(x,1) = \mu_0(x), \qquad \nu^p_{n+1} = \bar{F}(\nu^p_n, p_n), \\ (\bar{F}(\nu,h))(x,a) = \Big(\nu(x,0) + \nu(x,1)h(x)\Big)(1-a) + \Big(\sum_{z\in\mathcal{X}} \nu(z,1)\Big(q^{\nu}_{z,x}(1-h(z))\Big)\Big)a, \end{cases}$$

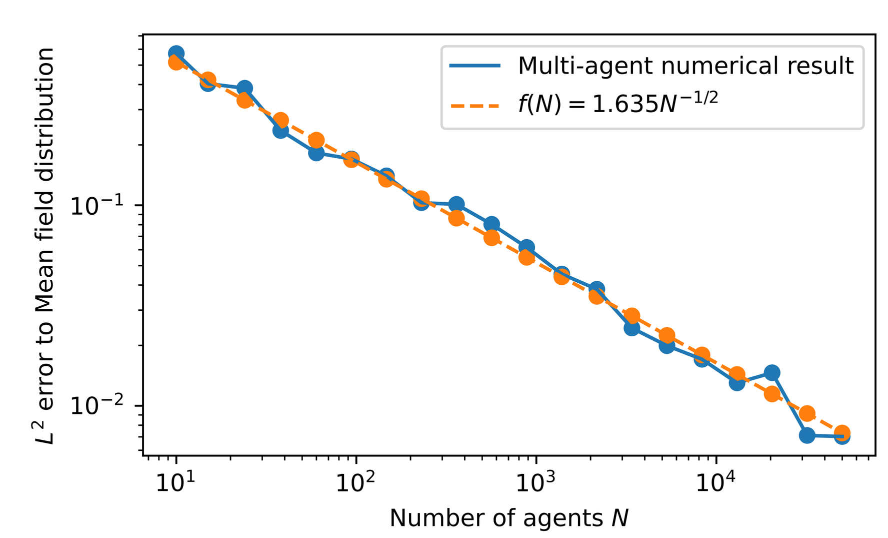
Dynamic Programming Principle:

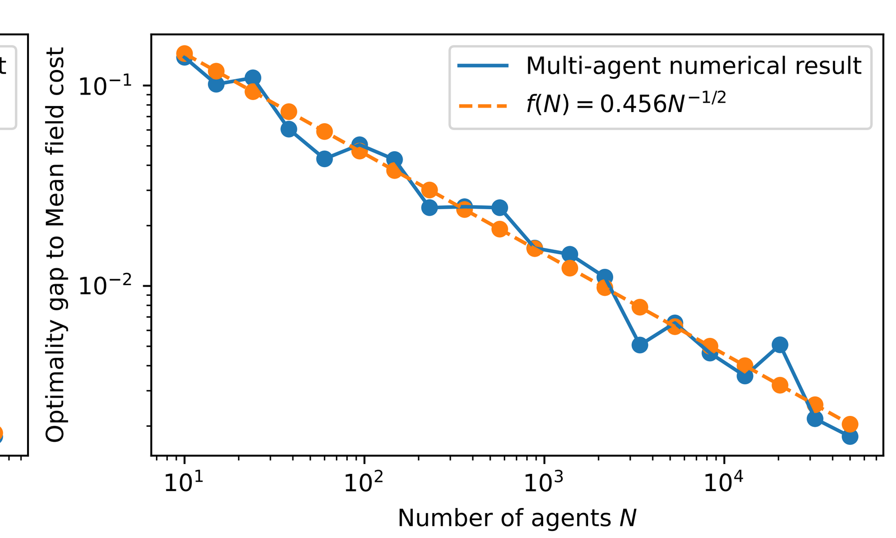
$$V_n(\nu) := \inf_{p\in\mathcal{P}_{n,T}} J(p(x),\nu) = \inf_{p\in\mathcal{P}_{n,T}} \sum_{m=n}^{T} \sum_{(x,a)\in\mathcal{S}} \nu^{p,\nu,n}_m(x,a)\Phi(x,\mu^{p,\nu,n}_m)ap_m(x),$$

$$\begin{cases} V_T(\nu) = \sum_{(x,a)\in\mathcal{S}} \nu(x,a)\Phi(x,\nu_X)a, \\ V_n(\nu) = \inf_{h\in\mathcal{H}} \sum_{(x,a)\in\mathcal{S}} \nu(x,a)\Phi(x,\nu_X)ah(x) + V_{n+1}(\bar{F}(\nu,h)), \qquad n < T, \end{cases}$$

ε-Approximation Error: p^* optimal for MFOS, $\hat{p}$ optimal for MAOS (each agent use same policy), $\mathcal{O}(1/\sqrt{N})$ error

$$J^N(p^*,\dots,p^*) - J^N(\hat{p},\dots,\hat{p})$$
$$\leq 2TL_{\Psi}(1+L_p)\left[\frac{|S|}{4\sqrt{N}}\left(\frac{1-(L_{\bar{F}}(1+L_p))^T}{1-(L_{\bar{F}}(1+L_p))}\right) + (L_{\bar{F}}(1+L_p))^T\frac{\sqrt{|S|-1}}{2\sqrt{N}}\right]$$

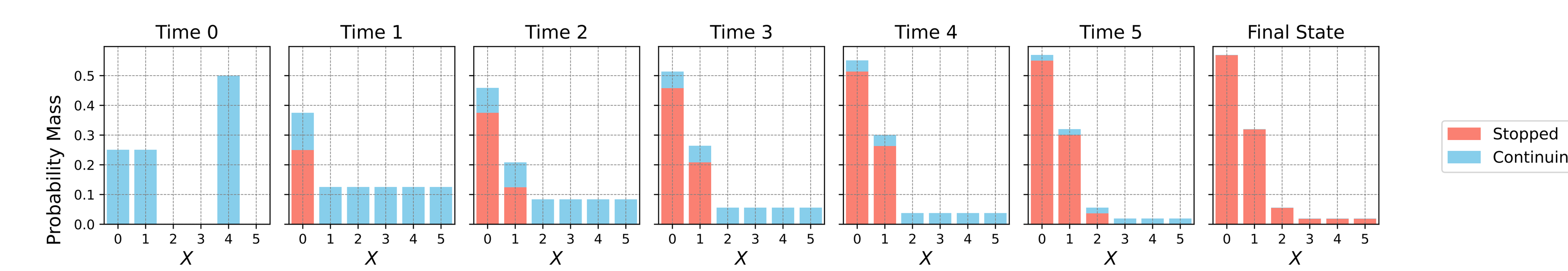




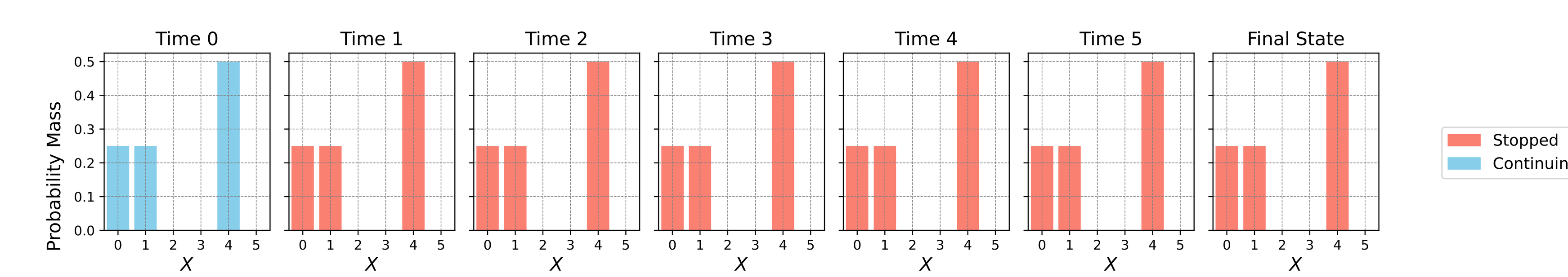
Rolling a Die (for 4 times!)

Setting: Each agent rolls a die and decides whether to stop. If one stops, he pays the cost of the **current number** he had. If one is not satisfied, he can reroll the die (up to 4 time). 25% starts with 1, 25% starts with 2, 50% starts with 5.

Asynchronous: $V^* = 1.6525$, stop when landing on smaller numbers



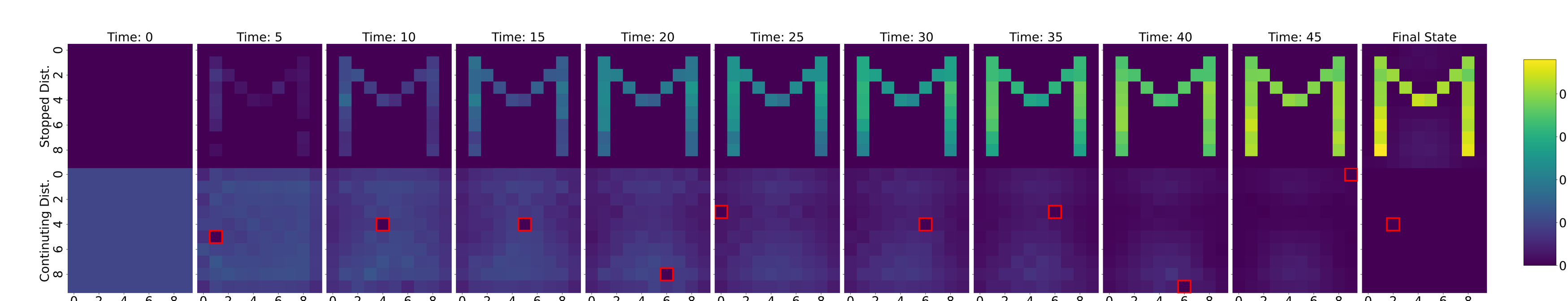
Synchronous: $V^* = 3.25$, stop at beginning



Matching the Letter (on 10 by 10 grid!)

Setting: Each agent is a drone starting randomly somewhere on the grid. For 50 steps, the drone diffuses uniformly to **accessible neighboring positions**. The goal is to finally end up with a distribution of drones that **matches a given letter (like a show!)**. An obstacle shows up in a random position to block the way each time.

Trajectory Snapshots:



Robustness to Initial distributions: