ADAGE-Diff: Two-level adaptive agent based modelling for differentiable policy design

Benjamin Patrick Evans[*]
JP Morgan AI Research
London
United Kingdom

Sihan Zeng
JP Morgan AI Research
Palo Alto
USA

Sumitra Ganesh
JP Morgan AI Research
New York
USA

Leo Ardon
JP Morgan AI Research
London
United Kingdom

Abstract

Agent-based models (ABMs) are a valuable tool for simulating complex systems. However, ABMs have limitations such as manual rule specification, lack of adaptation, intractability, and computational cost, limiting wide scale adoption. Recently, ADAGE was introduced to address the first two issues with a bi-level optimisation framework. However, this framework exacerbates the latter two issues. To help remedy these concerns, in this work, the bi-level framework is integrated with a differentiable simulator, resulting in tractable parameter updates and improved computational efficiency. The applicability of the framework is demonstrated for automated policy design, showing how taxation policies can be learnt to maximise fairness in a canonical multi-agent market entrance game with adaptive agents.

1 Introduction

While successful in a variety of domains (1; 2), agent-based models (ABMs) have been criticised due to the manual definition of behavioural rules (3; 4), lack of adaptation (the Lucas critique) (5), their intractability (6) and the computational cost of the models. The first two limitations can be addressed by integrating machine learning techniques such as reinforcement learning for the behavioural rules (7) combined with a bi-level optimisation for adaptation (8; 9). However, in existing implementations (such as (8)), the nested-level (inner layer) still features a general agent-based simulator, exacerbating criticisms of intractability and computational costs, particularly when learning is involved, limiting the scalability of such approaches. Utilising a differentiable inner layer promises to be a worthy direction to address this, allowing large-scale simulations (10) for real-world applications,

Differentiable agent-based models (dABM) have been a growing area in recent years (11; 12; 13; 14), driven by the availability of various new tools (15; 16). For example, during COVID, differentiable epidemiology (17; 18) became a key focus behind the modelling efforts for various countries due to the scalability of execution, calibration (6) and validation (19) of these models.

However, thus far, these dABMs have yet to be well integrated with learning-based frameworks, e.g., for work on automated policy design with adaptive agent behaviour. The closest works on policy design with dABMs are (17; 19). However, (17) only considers policy *evaluation*. (19) looks at epidemiological policy design but directly modifies the agent behavioural rules, rather than letting behaviour adapt through optimisation. In contrast, (8) adapts behavioural rules automatically but is costly to compute as the underlying simulator is not differentiable.

This leaves open an important gap combining differentiable simulators with adaptive learning frameworks for true automated policy design, which we look to address in this work. Specifically we,

1) Introduce a novel differentiable market entrance environment

2) Integrate this environment with a bi-level framework for gradient-based behavioural learning

3) Demonstrate how the framework can be used for automated policy design

[*] `benjamin.x.evans@jpmorgan.com`

D3S3: Data-driven and Differentiable Simulations, Surrogates, and Solvers @ NeurIPS 2024.

2 Bi-level Framework

ADAGE (9) is a generic two-layer framework for adaptive agent-based simulation (Fig. 1). The framework formalises adaptive ABM as a Stackleberg game $\mathcal{M}$ in an environment Ω between an outer layer L (leader) configuring the environment, and an inner agent-based simulation layer with follower agents $\mathbf{F}$ learning behavioural rules. Thus far, Ω has been assumed to be a general ABM; however, in this work, we exploit the case when Ω is differentiable to demonstrate efficient automated policy design with a differentiable adaptive simulator.

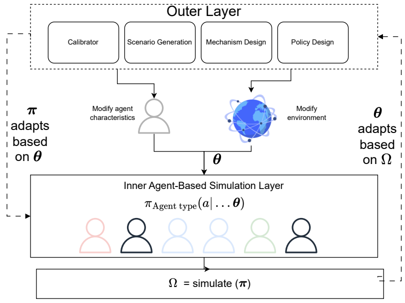

Outer Layer
Calibrator
Scenario Generation
Mechanism Design
Policy Design
Modify agent characteristics
Modify environment
$\boldsymbol{\pi}$ adapts based on $\boldsymbol{\theta}$
$\boldsymbol{\theta}$ adapts based on Ω
$\boldsymbol{\theta}$
Inner Agent-Based Simulation Layer
$\pi_{\text{Agent type}}(a| \ldots \boldsymbol{\theta})$
Ω = simulate ($\boldsymbol{\pi}$)

Figure 1: ADAGE bi-level framework.

2.1 Optimisation

$\mathcal{M}$ is a Partially Observable Markov Game with $N + 1$ agents (20), $N = |\mathbf{F}|$ of these agents form the simulation layer, and one agent is reserved as the leader L. The game can be represented as a tuple: (S, A, T, r, O, γ) where S is the state space, $A = (A_0, A_1, \ldots, A_N)$ the action space, $T : S \times A \to S$ the transition function, $r : S \times A \to \mathbb{R}^N$ the reward functions, $O = (O_0, O_1, \ldots, O_N)$ the observation spaces, and γ the discount factor. The outer layer learns a strategy π_L, and the inner layer agents learn individual strategies $\pi_i \in \boldsymbol{\pi}_{\mathbf{F}}$, to take reward maximising actions $a \sim \pi$.

L operates first, and $\mathbf{F}$ follow. At time step t each active agent i takes an action $a_{i,t} \sim \pi_i(O_{i,t})$ based on their private observation $O_{i,t}$, receiving utility $U_{i,t}$ from the environment Ω. Each agent in the game is attempting to find a strategy π_i to maximise their expected return in Ω:

$$R_i = \mathbb{E}\Bigg[\sum_t \gamma^t U_i(s_t, a_{i,t}, a_{-i,t})\Bigg]. \tag{1}$$

where $a_{-i,t}$ is the actions of the other agents. The task is to find a Stackelberg equilibrium, i.e. a solution $(\pi_L^*, \boldsymbol{\pi}_{\mathbf{F}}^*)$ at which no agent i can further improve R_i holding π_{-i} fixed. Due to the "gradient domination" condition satisfied by Eq. (1), every stationary point is globally optimal (21), therefore to find $(\pi_L^*, \boldsymbol{\pi}_{\mathbf{F}}^*)$ it suffices to solve the following coupled system of $N + 1$ non-linear equations. Each equation i states that the policy π_i is a first-order stationary point for agent i given that every other agent $j \neq i$ follows policy π_j:

$$\begin{cases} \nabla_{\pi_i^*} R_i = 0, \quad \forall i \in \mathbf{F}, \\ \nabla_{\pi_L^*} R_L = 0, \text{for the outer layer.} \end{cases} \tag{2}$$

The approach we take in this work for solving these coupled equations is alternating gradient descent (A-GD), i.e., maintain $\pi_{t,i}$ as an estimate of π_i^* and iteratively update $\pi_{t,i}$ for every agent i in the direction of $\nabla_{\pi_{t,i}} R_i$, where $\nabla_{\pi_{t,i}} R_i$ is computed with π_{-i} fixed to their latest iterates:

$$\begin{aligned} \pi_{t+1,L} &= \pi_{t,L} + \alpha_{t,L} \cdot \nabla_{\pi_{t,L}} R_L \ \ \text{given} \{\pi_{t,j}\}_{j \in \mathbf{F}}. \\ \pi_{t+1,i} &= \pi_{t,i} + \alpha_{t,i} \cdot \nabla_{\pi_{t,i}} R_i \ \ \text{given} \{\pi_{t,j}\}_{j \neq i}, \ \forall i \in \mathbf{F}. \end{aligned} \tag{3}$$

When R_i exhibits strong structure such as strong convexity, convergence of A-GD to $(\pi_L^*, \boldsymbol{\pi}_{\mathbf{F}}^*)$ is guaranteed under proper choices of learning rates $\alpha_{t,L}$, $\boldsymbol{\alpha}_{t,\mathbf{F}}$ (22; 23). In general, we need to choose $\alpha_{t,L} \gg \boldsymbol{\alpha}_{t,\mathbf{F}}$ to approximate a nested-loop algorithm which runs multiple $\boldsymbol{\pi}_{\mathbf{F}}$ updates per π_L update.

3 Environment: Market Entrance

For demonstration of agent adaptation and automated policy learning, we focus on a well established canonical agent-based environment, the market entrance game (24; 25). The entrance game represents various real-world problems, such as resource allocation, traffic management, and market profitability.

In the entrance game Ω, agents $i \in \mathbf{F}$ decide $a_i \in \{0, 1\}$ whether to enter a market at each time t. The utility depends on the entrance decisions of the other agents and a market capacity $1 \leq C \leq N$ [2]:

[2] let $c = \frac{C}{N}$.

$$U_i = \begin{cases} v, & a_i = 0. \\ v + 2 \cdot (C - m) & a_i = 1. \end{cases} \tag{4}$$

where a_i is the individual attendance decision, and $m = \sum_j^N a_j$ is the total attendance. The payoff for staying out is fixed to $v \geq 0$. Agents are rewarded (penalised) for entering an underpopulated (overpopulated) market. There are many equilibria for this game, however, the unique symmetric mixed strategy equilibrium is all agents attending with probability $p_i = p^* = \frac{C-1}{N-1}$ (26).

3.1 Differentiable

There are various challenges converting ABMs into continuous differentiable versions (27). In this section, we detail the changes required to convert Ω to a differentiable version Ω'. The three main challenges here are the discrete A, the piece-wise utility (Eq. (4)), and the discrete attendance m.

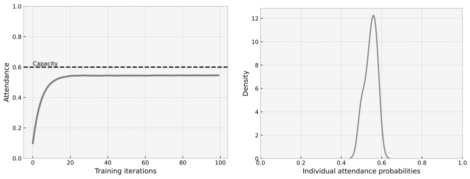
Rather than agents deciding binarily whether to attend, agents in Ω' instead learn the probability of attendance $0 \leq a_i \leq 1$ (so $a_i = p_i$), converting a_i to a continuous action rather than a discrete one. This permits rewriting U_i in terms of the expected payoff based on this attendance probability as:

$$U_i = \big[a_i \cdot (v + 2 \cdot (C - m))\big] + \big[(1 - a_i) \cdot v\big]. \tag{5}$$

which recovers Eq. (4) in the case of $a_i = 0$ or $a_i = 1$, but is importantly no longer piece-wise. The updated attendance calculation $m = \sum_j a_j$ is then the (continuous) expected attendance likelihood.

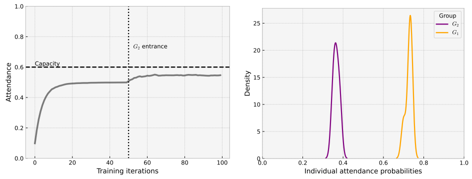
4 Experiments

4.1 Simulation Layer



(a) Convergence (b) Attendance

Figure 2: Learnt behaviour of $\mathbf{F}$ with $c = 0.6$ (Extended c in Figs. 7 and 8).



(a) Convergence (b) Attendance

Figure 3: Learnt behaviour with distinct groups G_1, G_2 with $c = 0.6$.

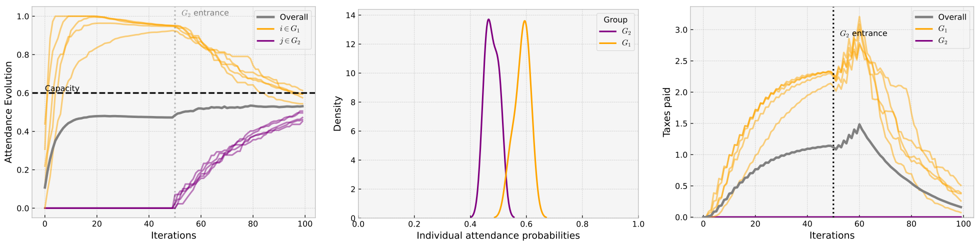
To validate the learnt behaviour of the inner layer $\boldsymbol{\pi}_{\mathbf{F}}$, experiments are run on Ω' in isolation (i.e., without any outer layer). Agents learn strategies resulting in equilibrium attendance rates p^* (Fig. 2a). Each agent independently converges to the approximate mixed-strategy equilibrium (of $a_i \approx p^*, \forall i$) maximising their individual R_i, as displayed by the distribution of a_i in Fig. 2b (across c in Fig. 8).

4.1.1 Fairness

Section 4.1 assumes $\forall i \in \mathbf{F}$ begin participating in the market at iteration 0, enabling convergence to fair outcomes as the agents are simultaneously updating their behaviour in response to the other agents. However, interesting short-run dynamics arise on the path to a new equilibrium if we assume two groups $G_1 \subset \mathbf{F}, G_2 \subset \mathbf{F}, G_1 \cap G_2 = \emptyset$ of agents: G_1 = Early participants, and G_2 = Late participants. For example, G_1 could represent first movers, and G_2 those who only participate later.

We split $\mathbf{F}$ into these two subgroups: $G_1 = \{0, 1, \ldots \frac{N}{2}\}, G_2 = \{\frac{N}{2} + 1, \frac{N}{2} + 2, \ldots N\}$, and assign different optimisation entries for these groups ι_{G_1}, ι_{G_2}, with $\iota_{G_2} \gg \iota_{G_1}$ to represent late participation, and rerun the optimisation process (Appendix A.2). The results are shown in Fig. 3 for $c = 0.6$. While m converges towards an equilibrium (confirmed by the long-range dynamics in Fig. 13), Ω' remains in an unfair state (28) in the shorter run dynamics (on the path to convergence), where $i \in G_1$ continue to attend with much higher frequency than $j \in G_2$, essentially blocking G_2 out of the market temporarily. This separation can be seen by the two distinct distributions in Fig. 3b, with $\boldsymbol{p}_{G_1} \gg \boldsymbol{p}_{G_2}$ (which holds across initialisations and thresholds Fig. 10). In the following section, we consider remedies to speed up the convergence towards the fair equilibrium ($a_i = p^*$ for all $i \in \mathbf{F}$), demonstrating efficient policy design with the bi-level framework.

4.2 Policy Design



(a) Convergence

(b) Attendance probabilities

(c) Paid Taxes

Figure 4: Learnt behaviour with L. The outer layer learns to increase τ after G_2 start participating in the market (a.). The taxes are suitably directed at G_1 (c.) As the agents begin to converge, the taxes paid reduce (a.), resulting in more equal attendance distributions (b.)

An outer layer L is used to design taxation policies to maximise fairness in Ω':

$$U_L = -0.5 \cdot \sum_{i \in \mathbf{F}} \sum_{j \in \mathbf{F}} \frac{|R_i - R_j|}{\bar{R}}. \tag{6}$$

representing the negative Gini coefficient of agent returns (shifted by min R_i), where $\bar{R} = \sum_{i \in \mathbf{F}} R_i / N$ is the mean (shifted) return across agents. Specifically, this achieved through the introduction of penalties $\tau \geq 0$ to Eq. (4) based on the overuse of resources:

$$U_i^{\tau} = U_i - (a_i \cdot \tau \cdot \max(a_i - m, 0)). \tag{7}$$

where τ serves as a tax or entrance fee for entering the market, penalising agents for over-attendance (compared to m), i.e., higher tax for more usage, causing agents to adapt their behaviour based on τ (Fig. 6c). The action a_L for L is to choose a taxation rate $0 \leq \tau \leq 0.1$ to maximise Eq. (6).

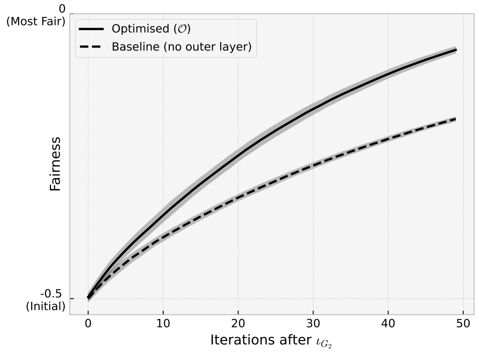


Figure 5: Resulting fairness curves (mean ± std) across 100 runs.

The attendances $\boldsymbol{a}$ and resulting fairness when introducing the outer layer are presented in Figs. 4b and 5 respectively. The outer layer successfully learns appropriate taxation policies (visualised in Fig. 4c) that minimise the attendance difference between the two groups (Fig. 4b vs Fig. 3b). This fairness is further verified by looking at the resulting curve in Fig. 5, where the outer layer learns appropriate taxation rates τ to penalise "greedy" utilisation, resulting in significantly fairer outcomes and speeding up convergence towards this fairer equilibrium, demonstrating efficient policy design.

5 Discussions and Conclusion

This work has shown how a differentiable agent-based simulator can be used as part of a bi-level optimisation process for automated policy design in a system with adaptive agents, demonstrated through taxation of over-utilisation of a shared resource. While previous work has considered adaptive behaviour (29), or differentiable design (19), this work integrates both adaptation and differentiability to overcome limitations of existing approaches for automated policy design.

The continued progress in differentiable agent-based simulators, paired with modern optimisation techniques (30), promises to expand the applications of agent-based modelling, helping to successfully address persisting questions surrounding tractability, cost, and adaptation.

Disclaimer

This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan”) and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

[1] R. L. Axtell and J. D. Farmer, “Agent-based modeling in economics and finance: Past, present, and future,” *Journal of Economic Literature*, pp. 1–101, 2022.

[2] L. Tesfatsion, “Agent-based computational economics: Overview and brief history,” *Artificial intelligence, learning and computation in economics and finance*, pp. 41–58, 2023.

[3] A. Turrell, “Agent-based models: understanding the economy from the bottom up,” *Bank of England Quarterly Bulletin*, p. Q4, 2016.

[4] O. A. Osoba, R. Vardavas, J. Grana, R. Zutshi, and A. Jaycocks, “Modeling agent behaviors for policy analysis via reinforcement learning,” in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 213–219, IEEE, 2020.

[5] M. Napoletano, “A short walk on the wild side: Agent-based models and their implications for macroeconomic analysis,” *Revue de l’OFCE*, no. 3, pp. 257–281, 2018.

[6] A. Quera-Bofarull, A. Chopra, A. Calinescu, M. Wooldridge, and J. Dyer, “Bayesian calibration of differentiable agent-based models,” *ICLR Workshop AI for Agent-Based Modelling*, 2023.

[7] L. Ardon, J. Vann, D. Garg, T. Spooner, and S. Ganesh, “Phantom - a RL-driven multi-agent framework to model complex systems,” in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’23, (Richland, SC), p. 2742–2744, International Foundation for Autonomous Agents and Multiagent Systems, 2023.

[8] S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher, “The AI economist: Taxation policy design via two-level deep multiagent reinforcement learning,” *Science Advances*, vol. 8, no. 18, p. eabk2607, 2022.

[9] B. Patrick Evans, S. Zeng, S. Ganesh, and L. Ardon, “Adage: A generic two-layer framework for adaptive agent based modelling,” *Conference on Complex Systems*, 2024.

[10] A. Chopra, R. Raskar, E. S. Gel, J. Subramanian, B. Krishnamurthy, S. Romero-Brufau, K. S. Pasupathy, and T. C. Kingsley, “DeepABM: scalable and efficient agent-based simulations via geometric learning frameworks - a case study for covid-19 spread and interventions,” in *Proceedings of the Winter Simulation Conference*, WSC ’21, IEEE Press, 2022.

[11] J. Dyer, A. Quera-Bofarull, A. Chopra, J. D. Farmer, A. Calinescu, and M. Wooldridge, “Gradient-assisted calibration for financial agent-based models,” in *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 288–296, 2023.

[12] P. Andelfinger, “Towards differentiable agent-based simulation,” *ACM Transactions on Modeling and Computer Simulation*, vol. 32, no. 4, pp. 1–26, 2023.

[13] P. Andelfinger, “Differentiable agent-based simulation for gradient-guided simulation-based optimization,” in *Proceedings of the 2021 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pp. 27–38, 2021.

[14] A. Chopra, J. Subramanian, B. Krishnamurthy, and R. Raskar, “Agenttorch: Agent-based modeling with automatic differentiation,” in *Second Agent Learning in Open-Endedness Workshop*, 2023.

[15] A. Chopra, J. Subramanian, B. Krishnamurthy, and R. Raskar, “flame: A framework for learning in agent-based models,” in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pp. 391–399, 2024.

[16] A. Quera-Bofarull, J. Dyer, A. Calinescu, J. D. Farmer, and M. Wooldridge, “Blackbirds: Black-box inference for differentiable simulators,” *Journal of Open Source Software*, vol. 8, no. 89, p. 5776, 2023.

[17] A. Chopra, A. Rodríguez, J. Subramanian, A. Quera-Bofarull, B. Krishnamurthy, B. A. Prakash, and R. Raskar, “Differentiable agent-based epidemiology,” in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’23, (Richland, SC), p. 1848–1857, International Foundation for Autonomous Agents and Multiagent Systems, 2023.

[18] A. Chopra, A. Rodríguez, J. Subramanian, B. Krishnamurthy, B. A. Prakash, and R. Raskar, “Differentiable agent-based epidemiological modeling for end-to-end learning,” in *ICML 2022 Workshop AI for Agent-Based Modelling*, 2022.

[19] A. Quera-Bofarull, A. Chopra, J. Aylett-Bullock, C. Cuesta-Lazaro, A. Calinescu, R. Raskar, and M. Wooldridge, “Don’t simulate twice: One-shot sensitivity analyses via automatic differentiation,” in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’23, (Richland, SC), p. 1867–1876, International Foundation for Autonomous Agents and Multiagent Systems, 2023.

[20] M. Gerstgrasser and D. C. Parkes, “Oracles & followers: Stackelberg equilibria in deep multi-agent reinforcement learning,” in *International Conference on Machine Learning*, pp. 11213–11236, PMLR, 2023.

[21] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, “On the theory of policy gradient methods: Optimality, approximation, and distribution shift,” *Journal of Machine Learning Research*, vol. 22, no. 98, pp. 1–76, 2021.

[22] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, “A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic,” *SIAM Journal on Optimization*, vol. 33, no. 1, pp. 147–180, 2023.

[23] S. Zeng, T. T. Doan, and J. Romberg, “A two-time-scale stochastic optimization framework with applications in control and reinforcement learning,” *SIAM Journal on Optimization*, vol. 34, no. 1, pp. 946–976, 2024.

[24] I. Erev and A. Rapoport, “Coordination,“magic,” and reinforcement learning in a market entry game,” *Games and economic behavior*, vol. 23, no. 2, pp. 146–175, 1998.

[25] C. F. Camerer, *Behavioral game theory: Experiments in strategic interaction*. Princeton university press, 2011.

[26] J. Duffy and E. Hopkins, “Learning, information, and sorting in market entry games: theory and evidence,” *Games and Economic Behavior*, vol. 51, no. 1, pp. 31–62, 2005.

[27] A. Quera-Bofarull, J. Dyer, A. Calinescu, and M. Wooldridge, “Some challenges of calibrating differentiable agent-based models,” *ICML Differentiable Almost Everything Workshop*, 2023.

[28] S.-H. Chen and U. Gostoli, “Coordination in the El Farol bar problem: The role of social preferences and social networks,” *Journal of Economic Interaction and Coordination*, vol. 12, pp. 59–93, 2017.

[29] Q. Mi, S. Xia, Y. Song, H. Zhang, S. Zhu, and J. Wang, “TaxAI: A dynamic economic simulator and benchmark for multi-agent reinforcement learning,” in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’24, (Richland, SC), p. 1390–1399, International Foundation for Autonomous Agents and Multiagent Systems, 2024.

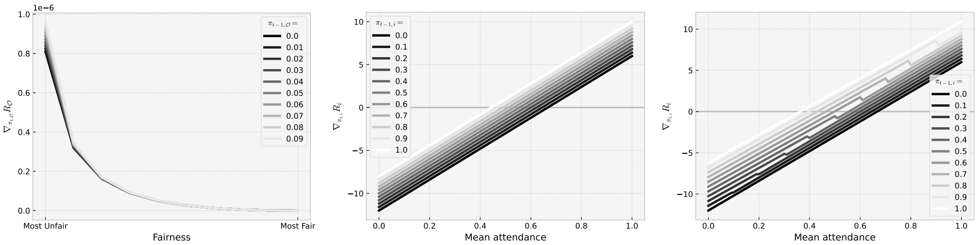
[30] S. Zeng and T. Doan, "Fast two-time-scale stochastic gradient method with applications in reinforcement learning," in *The Thirty Seventh Annual Conference on Learning Theory*, pp. 5166–5212, PMLR, 2024.

[31] W. B. Arthur, "Foundations of complexity economics," *Nature Reviews Physics*, vol. 3, no. 2, pp. 136–145, 2021.

A Optimisation details

Experiments are performed with $I = N \cdot 10$ iterations of A-GD (Eq. (3)). Each iteration, $i \in \mathbf{F}$ performs one update step. Every other iteration, L also performs an update step. We use $\boldsymbol{\alpha}_{t,\mathbf{F}} = 0.001$, and $\alpha_{t,L} = 0.04$. Sensitivity analysis is performed on these parameters in Appendix C.

A.1 Gradient updates



(a) Outer gradient update $\pi_{0,L}$ based on varying inner strategies $\boldsymbol{\pi}_{\mathbf{F}}$ (x-axis)

(b) Inner gradient update $\pi_{t,0}$ with $\pi_L = 0.$ and varying π_i for $i > 0$ (x-axis)

(c) Inner gradient update (across $\pi_{t,0}$, shades) with $\pi_L = 0.1$ and varying π_i for $i > 0$ (x-axis)

Figure 6: Example gradient updates across different starting strategies (shades, per legend).

Example gradient update steps are visualised in Fig. 6. The gradient update for the outer layer is non-linear in the Gini coefficient. The update step for the inner layer is linear based on the mean attendance rate of the other participants (Fig. 6b), and a non-linearity is introduced when taxation penalties are present (e.g. as shown in Fig. 6c).

A.2 Late participants

To model late participants, we set $\iota_{G_1} = 0$, and $\iota_{G_2} = \frac{I}{2}$. For iterations $0, \ldots \iota_{G_2}$, only G_1 is learning ($\alpha_{t,j} = 0$ for all $j \in G_2$). At iteration ι_{G_2}, G_2 enter the market and begin learning as well, so from iterations $\iota_{G_2}, \ldots, I$ all agents learn simultaneously, i.e., $\alpha_{t,j} > 0$ for all $i \in \mathbf{F}$.

B Penalised utilities

The penalised utility function is:

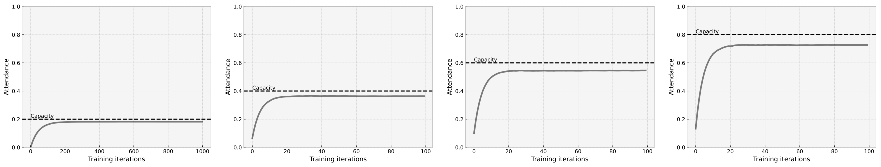
$$U_i^{\tau} = \big[a_i \cdot (v + 2 \cdot (C - m))\big] + \big[(1 - a_i) \cdot v\big] - \big[a_i \cdot \tau \cdot \max(a_i - m, 0)\big]. \tag{8}$$

Eq. (8) is not differentiable everywhere, specifically at or outside the boundaries of max (with 0 gradient). However, in practice, this still works well with Ω', as the cases we care about here are the over-attendees, where the derivative is defined. Agents with $p_i < m$ will not be penalised and, thus, do not require this gradient update.

C Sensitivity analysis

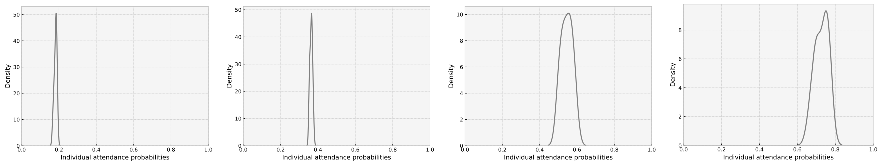
C.1 Capacity c

Following convention, we focus primarily on the $c = 0.6$ case (31), but provide results across c in Figs. 7 and 8. In each case, we see convergence to the mixed strategy equilibrium, confirming the results are independent of the exact c used.



(a) $c = 0.2$ (b) $c = 0.4$ (c) $c = 0.6$ (d) $c = 0.8$

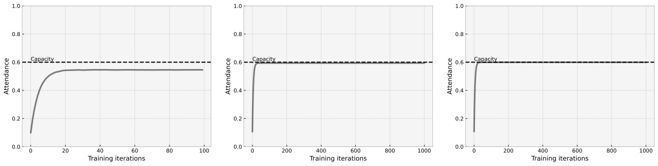
Figure 7: Convergence paths across capacities c



(a) $c = 0.2$ (b) $c = 0.4$ (c) $c = 0.6$ (d) $c = 0.8$

Figure 8: Attendance probability distributions across capacacties c.

C.2 Participants N



(a) $N = 10$ (b) $N = 100$ (c) $N = 1000$

Figure 9: Convergence paths across N

We use $N = 10$ for analysis, but show $N \in \{10, 100, 1000\}$ (for $c = 0.6$) in Fig. 9. Note that since the mixed strategy equilibrium is $p^* = \frac{C-1}{N-1}$, in each case, the mixed strategy equilibrium is learnt, and as N increases this more closely approximates $m = c$. The consistency of results across N confirms the conclusions are independent of specific N.

C.3 Strategy Initialisation

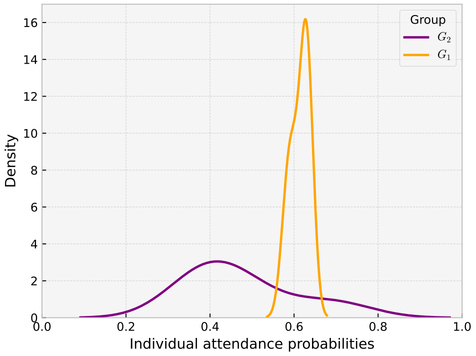


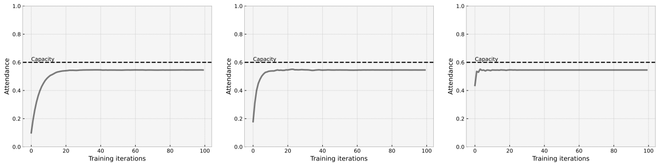
Figure 10: Resulting attendance distributions under uniform strategy initialisations $c = 0.6, N = 10$.

To confirm (un)fairness is not trivially a result of the initial strategy initialisation $\boldsymbol{\pi}_{\mathbf{F}} = 0$, we analyse across uniform random initialisation for $\pi_{t+1,i} \sim U(0,1), \forall i \in \mathbf{F}$ Fig. 10. We see the resulting attendance distributions still significantly differ, indicating the unfairness remains even after (50) learning iterations have taken place.

C.4 Optimisation parameters

In this section we check the sensitivity to the learning rates of the two layers.

C.4.1 Inner layer learning rate α

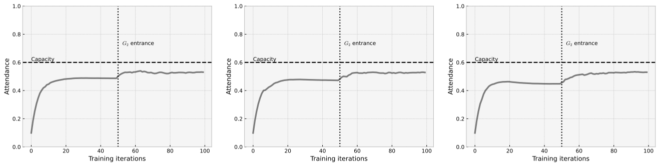


(a) $\boldsymbol{\alpha}_{t,\mathbf{F}} = 0.001$ (b) $\boldsymbol{\alpha}_{t,\mathbf{F}} = 0.002$ (c) $\boldsymbol{\alpha}_{t,\mathbf{F}} = 0.01$

Figure 11: Convergence paths across $\boldsymbol{\alpha}_{t,\mathbf{F}}$ with no tax.

Altering the learning rate of the inner layer alters the convergence speed, but in each case still converges to the equilibrium (Fig. 11). The default case is $\boldsymbol{\alpha}_{t,\mathbf{F}} = 0.001$.

C.4.2 Outer layer learning rate $\alpha_{t,L}$



(a) $\boldsymbol{\alpha}_{t,L} = 0.02$ (b) $\boldsymbol{\alpha}_{t,L} = 0.04$ (c) $\boldsymbol{\alpha}_{t,L} = 0.08$

Figure 12: Convergence paths across $\alpha_{t,L}$ with late participants (and $\boldsymbol{\alpha}_{t,\mathbf{F}} = 0.01$).

Adjusting the outer layer learning rate has relatively little effect on the resulting convergence Fig. 12, indicating stability around exact learning rates. The default case is $\alpha_{t,L} = 0.04$.

D Long range dynamics

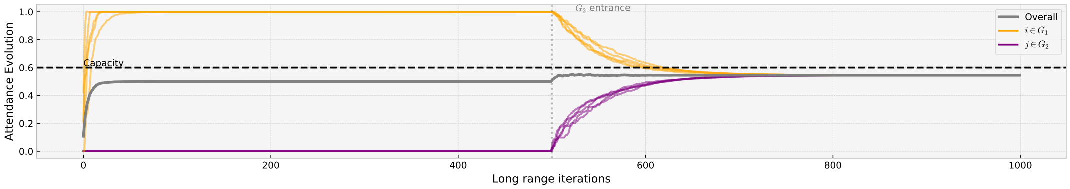


Figure 13: Long range convergence dynamics with $c = 0.6$, $N = 10$ (over extended iterations).

To show convergence given a long enough timescale, we present the long range dynamics in Fig. 13. We see the two groups eventually converge after a period of learning.