

Environment archetypes

Concept. To define and develop an archetype simulation environment for simple state transitions. In our classification scheme, this archetype is defined by a trivial state partition graph topology and would make sense for simulations of sequential design problems, sports matches and other simple gameplay domains. We will also discuss the typical ways in which the state of the system may only partially be observed in realistic examples, and analyse how best to deal with each situation. For the mathematically-inclined, this chapter will define the mapping of our formalism to simple state transitions. For the programmers, the software which is designed and described in this chapter can be found in the public Git repository here: <https://github.com/worldsoop/worldsoop>.

4.1 Simple state transitions

The simple state transition archetype refers to simulation environments where there is no obvious computational benefit to partitioning the state into concurrently updating or acting on separate components. There may even be performance benefits from keeping state information all within the same common data structure in memory, but this can depend on the specific problem of study.

In the interest of completeness with respect to the chapters which follow on from this one, we have illustrated the trivial state partition graph topology for this archetype in Fig. 4.1.

In order to understand what sorts of data might be collected about this archetype in realistic scenarios, let's begin by considering the types of real-world problem domain which have leveraged simulation environments to train control algorithms in the literature. The subset of these which may best suit the simple state transition environment archetype are:

- Event-based simulations of sports matches, e.g., football [1], rugby [2], tennis [3], etc., and other forms of game — all of which typically define a relatively simple global match/gameplay context as their ‘state’.
- Sequential design simulations to change the configuration or data collection strategies of, e.g., astronomical telescopes [4, 5], biological experiments [6] and user interfaces [7], which typically define a relatively simple and finite set of possible actions that can be taken.



Figure 4.1: State partition graph topology for simple state transition archetypes.

Based on the examples given above, what kinds of data may be available to infer the state and parameters of a simulation environment using this archetype?

In the case of team sports matches, a team manager could have access to a large body of data which assesses the capabilities of their players before a game, but they must rely on more subjective or limited data to assess the performance of their team (and the opposition) in the middle of a match. The ‘state’ of the whole system in this case can be not only the state of play, but also information about, e.g., fatigue or on-the-day performance of each player for both teams and sets of substitutes.

Talk about sequential design data here...

In a general offline learning setting, realistic historical data collected about the state values for this archetype will typically provide an important mechanism for model determination. Users of this archetype in the real world also typically have access to independent datasets that can be used offline to better determine some parameters of the simulation environment which are not observed directly. In the online learning setting, one typically directly observes some portion of the state values, and may only indirectly observe the other state values, if at all.

Within this topological archetype there will also be other categories to think are applicable:

- *Types of observation that can be made about the state.*
- *Types of action that can be taken by an agent.*
- *Types of agent? How frequently do these actions get taken? On what part of the state?*

Types of actor in this archetype:

- *sports team manager and other game player 1. substitute players 2. changes to tactics*

4.2 Dynamic spatial fields

The dynamic spatial field archetype refers to simulation environments which have highly-structured, bidirectional communication between state partitions. The graph topology of this archetype is totally connected, but some connections matter more than others. As a helpful analogy, you can think of these partitions as being structured topologically in a kind of ‘lattice’ configuration, where connections to other partitions over different distances in the lattice can contribute different importance weights in affecting each local state partition. This lattice structure can be best intuited from a visual discription, so we have illustrated an example graph topology for the dynamic spatial field archetype in Fig. 4.2.

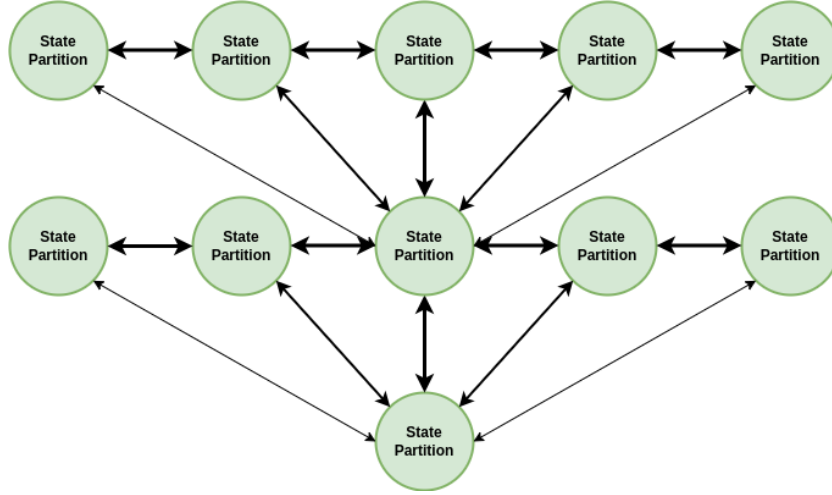


Figure 4.2: State partition graph topology for dynamic spatial field archetypes. The graph should be totally connected but we only show some of the possible connections for simplicity.

Depending on the spatial dimensionality of the field, we might need to visualise the lattice in Fig. 4.2 as existing in more spatial dimensions than the 2-dimensional example we have illustrated. However, as it turns out, there are many important real-world examples of 2-dimensional spatial systems to control anyway.

Before we move onto a discussion of data, we can study spatial systems which follow this archetype a little more mathematically by adapting the probabilistic formalism we developed in the first part of this book.

Need to give more recall to the reader here about the probabilistic formalism...

Let's return to this probabilistic formalism that we introduced earlier and note that the covariance matrix estimate with elements $C_{t+1}^{ij}(z)$ represents a matrix that could get very large, depending on the problem. For example; if we encoded the state of a 2-dimensional spatial field of values into the elements X_t^i , the number of elements in the covariance matrix $C_{t+1}^{ij}(z)$ would scale as $4N^2$ — where N here is the number of spatial points we wanted to encode.

One solution to this scaling problem is to exploit the fact that, in many spatial processes, the proximity of points can strongly determine how correlated they are. Hence, for pairwise distances further than some threshold, the covariance matrix elements should tend towards 0. If we were to place points along the diagonal of $C_{t+1}^{ij}(z)$ in order of how close they are to each other, this threshold would then be represented as a *banded matrix*. We have illustrated such a matrix in Fig. 4.3 in which the 'bandwidth' is defined as the number of diagonals one needs to traverse from the main diagonal before encountering a diagonal of 0s.

- At some point it might be sensible to move into the Fourier domain here — at least for derivations and calculations. Probably more intuitive for the reader to keep it mostly in real space though if possible.
- The extra detail that's also needed here is to consider how we encode a 2-dimensional spatial

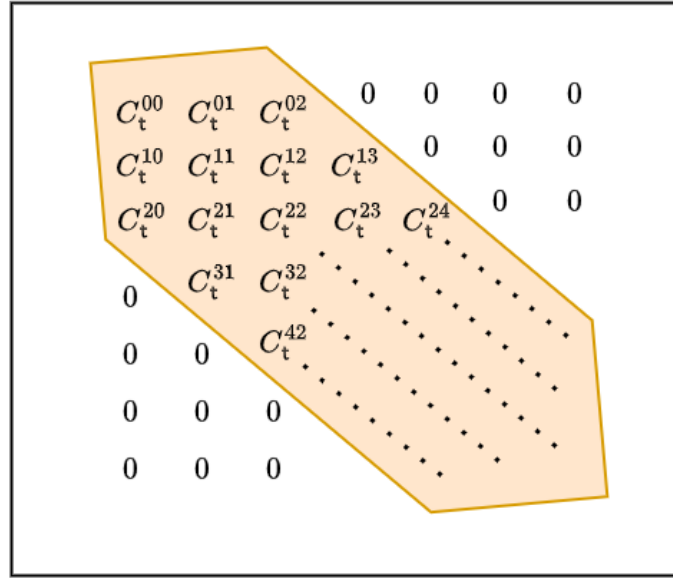


Figure 4.3: An illustration of a banded covariance matrix with a bandwidth of 2.

process into our state vector, and how the elements of the resulting state vector might be correlated to one another depending on their spatial proximity. If we start with a Markovian Gaussian random field, we can derive the Matérn kernel over these spatial coordinates in order to correlate the state vectors in such a way.

- Also look into the Radial Basis Function (RBF) and higher-order derived kernels based on DALI expansion [8] in order to try and capture non-Gaussianity.

To begin our discussion of data, let’s start by considering the subset of real-world problem domains which fit well within the dynamic spatial field archetype. These would be:

- Spatial simulations of population disease spread and control in the context of global disease outbreaks [9] or endemic, spatially-clustered infections like malaria [10].
- Spatial ecosystem management environments to infer forest wildfire dynamics [11] or improve conservation decision-making [12].
- Weather system simulations to improve decision-making for agricultural yields [13] or enhance stormwater flood mitigations [14].

Within this topological archetype there will also be other categories to think are applicable:

- Types of observation that can be made about the state.
- Types of action that can be taken by an agent.

- Types of agent? How frequently do these actions get taken? On what part of the state?

Types of actor in this archetype:

- public health authority and wildlife/national park control authority and livestock/crop farmer
 1. spatially detect disease or damage
 2. change state of a subset of the population

4.3 Distributed state networks

The distributed state network archetype refers to simulation environments whose bidirectional state partition graph topology is completely arbitrary, requiring no particular connectivity structure at all. Each state partition in this archetype typically refers to the same type of real-world node, object or sub-model. Due to the flexibility in topological structure, this archetype is well-suited to ‘network’ models of realistic phenomena. We have illustrated the state partition graph which fits into this category in Fig. 4.4.

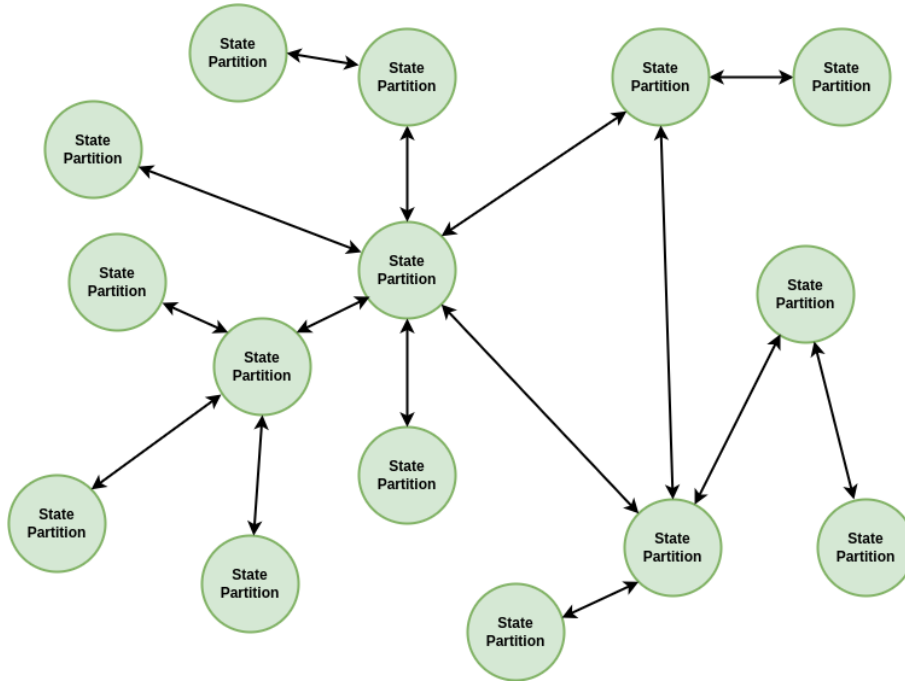


Figure 4.4: State partition graph topology for distributed state network archetypes.

Before discussing what kinds of data are typically available to infer states and parameters of environments in this archetype, it will be informative to consider the various real-world problem domains which apply. The particular subset of these domains which seem to fit well within this classification include:

- Computational models of human brain conditions, e.g., Parkinson’s disease [15], epilepsy [16], Alzheimer’s [17], etc., for deep brain stimulation control and other forms of treatment.

- Simulations of complex urban infrastructure networks to target various kinds of optimisation, e.g., traffic signal control [18], power dispatch [19] and water pipe maintenance [20].

Within this topological archetype there will also be other categories to think are applicable:

- Types of observation that can be made about the state.
- Types of action that can be taken by an agent.
- Types of agent? How frequently do these actions get taken? On what part of the state?

Types of actor in this archetype:

- brain doctor and traffic light controller and city infrastructure maintainer 1. change the state of a subset of nodes in the network

4.4 Multi-stage pipelines

The multi-stage pipeline archetype refers to simulation environments with a directional state partition graph with arbitrary connection topology. In this case, each state partition corresponds to a separate stage in some pipeline model of the real-world phenomenon. Directionality in the connection structure is the key distinction between this archetype and the others in our classification scheme. We've provided one illustrated example of a multi-stage pipeline state partition graph in Fig. 4.5.

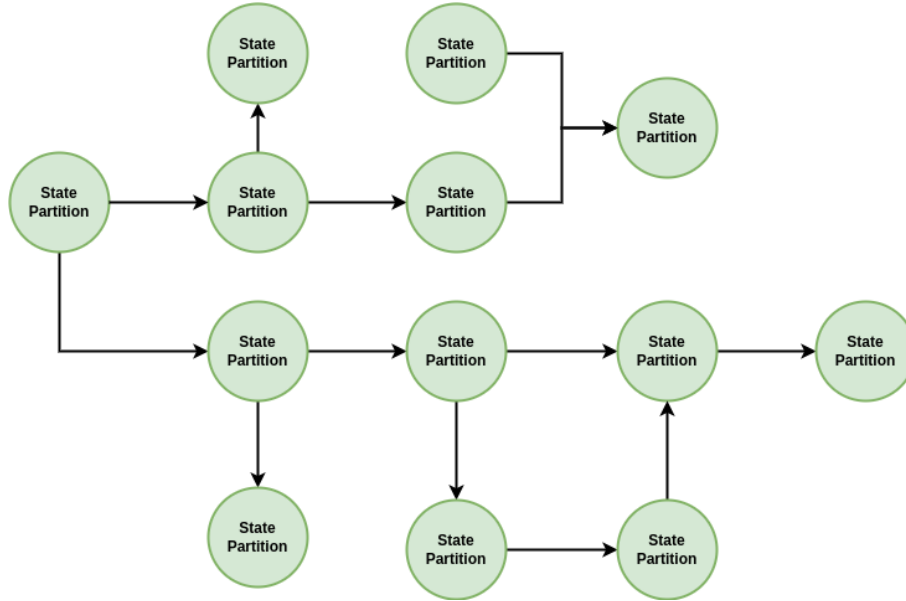


Figure 4.5: State partition graph topology for multi-stage pipeline archetypes.

We're now ready to discuss how multi-stage pipelines are typically observed with data from realistic scenarios. To do this, it will help to first consider which real-world problem domains are best-suited to this archetype structure. From the literature, we can identify these as:

- Simulations of human logistics pipelines, e.g., organised supply chains [21], humanitarian aid distribution pipelines [22], hospital capacity planning [23], etc.
- Environments which emulate data pipeline optimisation problems [24].

Within this topological archetype there will also be other categories to think are applicable:

- Types of observation that can be made about the state.
- Types of action that can be taken by an agent.
- Types of agent? How frequently do these actions get taken? On what part of the state?

Types of actor in this archetype:

- supply/relief chain controller and hospital logistics manager and data pipeline controller 1. modify the relative flows between different pipeline stages

4.5 Centralised exchanges

The centralised exchange archetype refers to simulation environments with a very specific bidirectional state partition graph topology. The graph connection structure is a star configuration where every state partition is connected to the same, centralised state partition which itself has a unique function in the model. In case this description isn't that clear, we've added an illustration of the graph for this archetype in Fig. 4.6.

Before moving on to discuss how data is typically collected for centralised exchanges, it will be informative to list the examples of phenomena where this archetype can be found in the real-world. These problem domains are:

- Financial [25, 26] and sports betting [27] market simulations for developing algo-trading strategies and portfolio optimisation [28], as well as housing market simulations [29, 30] to evaluate government policies.
- Simulations of other forms of resource exchange through centralised mediation, such as in prosumer energy markets [31].

Within this topological archetype there will also be other categories to think are applicable:

- Types of observation that can be made about the state.
- Types of action that can be taken by an agent.
- Types of agent? How frequently do these actions get taken? On what part of the state?

Types of actor in this archetype:

- financial/betting/other market trader and market exchange mediator 1. interact with the market using an agent or collection of agents

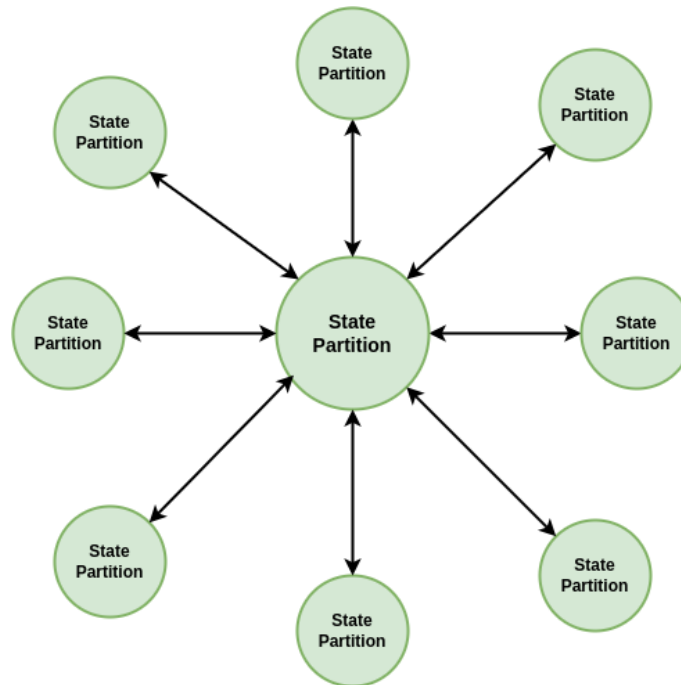


Figure 4.6: State partition graph topology for centralised exchange archetypes.

Bibliography

- [1] M. Pulis and J. Bajada, “Reinforcement learning for football player decision making analysis,” in *StatsBomb Conference*, 2022.
- [2] T. Sawczuk, A. Palczewska, and B. Jones, “Markov decision processes with contextual nodes as a method of assessing attacking player performance in rugby league,” in *Advances in Computational Intelligence Systems: Contributions Presented at the 20th UK Workshop on Computational Intelligence, September 8-10, 2021, Aberystwyth, Wales, UK 20*. Springer, 2022, pp. 251–263.
- [3] N. Ding, K. Takeda, and K. Fujii, “Deep reinforcement learning in a racket sport for player evaluation with technical and tactical contexts,” *IEEE Access*, vol. 10, pp. 54 764–54 772, 2022.
- [4] P. Jia, Q. Jia, T. Jiang, and J. Liu, “Observation strategy optimization for distributed telescope arrays with deep reinforcement learning,” *The Astronomical Journal*, vol. 165, no. 6, p. 233, 2023.
- [5] S. Yatawatta and I. M. Avruch, “Deep reinforcement learning for smart calibration of radio telescopes,” *Monthly Notices of the Royal Astronomical Society*, vol. 505, no. 2, pp. 2141–2150, 2021.
- [6] N. J. Treloar, N. Braniff, B. Ingalls, and C. P. Barnes, “Deep reinforcement learning for optimal experimental design in biology,” *PLOS Computational Biology*, vol. 18, no. 11, p. e1010695, 2022.
- [7] J. D. Lomas, J. Forlizzi, N. Poonwala, N. Patel, S. Shodhan, K. Patel, K. Koedinger, and E. Brunskill, “Interface design optimization as a multi-armed bandit problem,” in *Proceedings of the 2016 CHI conference on human factors in computing systems*, 2016, pp. 4142–4153.
- [8] E. Sellentin, M. Quartin, and L. Amendola, “Breaking the spell of gaussianity: forecasting with higher order fisher matrices,” *Monthly Notices of the Royal Astronomical Society*, vol. 441, no. 2, pp. 1831–1840, 2014.
- [9] A. Q. Ohi, M. Mridha, M. M. Monowar, and M. A. Hamid, “Exploring optimal control of epidemic spread using reinforcement learning,” *Scientific reports*, vol. 10, no. 1, p. 22106, 2020.
- [10] R. Carter, K. N. Mendis, and D. Roberts, “Spatial targeting of interventions against malaria,” *Bulletin of the World Health Organization*, vol. 78, pp. 1401–1411, 2000.

- [11] S. Ganapathi Subramanian and M. Crowley, “Using spatial reinforcement learning to build forest wildfire dynamics models from satellite images,” *Frontiers in ICT*, vol. 5, p. 6, 2018.
- [12] M. Lapeyrolerie, M. S. Chapman, K. E. Norman, and C. Boettiger, “Deep reinforcement learning for conservation decisions,” *Methods in Ecology and Evolution*, vol. 13, no. 11, pp. 2649–2662, 2022.
- [13] M. Chen, Y. Cui, X. Wang, H. Xie, F. Liu, T. Luo, S. Zheng, and Y. Luo, “A reinforcement learning approach to irrigation decision-making for rice using weather forecasts,” *Agricultural Water Management*, vol. 250, p. 106838, 2021.
- [14] S. M. Saliba, B. D. Bowes, S. Adams, P. A. Beling, and J. L. Goodall, “Deep reinforcement learning with uncertain data for real-time stormwater system control and flood mitigation,” *Water*, vol. 12, no. 11, p. 3222, 2020.
- [15] M. Lu, X. Wei, Y. Che, J. Wang, and K. A. Loparo, “Application of reinforcement learning to deep brain stimulation in a computational model of parkinson’s disease,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 339–349, 2019.
- [16] J. Pineau, A. Guez, R. Vincent, G. Panuccio, and M. Avoli, “Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach,” *International journal of neural systems*, vol. 19, no. 04, pp. 227–240, 2009.
- [17] K. Saboo, A. Choudhary, Y. Cao, G. Worrell, D. Jones, and R. Iyer, “Reinforcement learning based disease progression model for alzheimer’s disease,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 903–20 915, 2021.
- [18] K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, “A survey on reinforcement learning models and algorithms for traffic signal control,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–38, 2017.
- [19] Q. Li, X. Zhang, J. Guo, X. Shan, Z. Wang, Z. Li, and K. T. Chi, “Integrating reinforcement learning and optimal power dispatch to enhance power grid resilience,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1402–1406, 2021.
- [20] Z. A. Bukhsh, H. Molegraaf, and N. Jansen, “A maintenance planning framework using online and offline deep reinforcement learning,” *Neural Computing and Applications*, pp. 1–12, 2023.
- [21] Y. Yan, A. H. Chow, C. P. Ho, Y.-H. Kuo, Q. Wu, and C. Ying, “Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 162, p. 102712, 2022.
- [22] L. Yu, C. Zhang, J. Jiang, H. Yang, and H. Shang, “Reinforcement learning approach for resource allocation in humanitarian logistics,” *Expert Systems with Applications*, vol. 173, p. 114663, 2021.
- [23] S. S. Shuvo, M. R. Ahmed, H. Symum, and Y. Yilmaz, “Deep reinforcement learning based cost-benefit analysis for hospital capacity planning,” in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–7.

- [24] K. Nagrecha, L. Liu, P. Delgado, and P. Padmanabhan, “Intune: Reinforcement learning-based data pipeline optimization for deep recommendation models,” in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 430–442.
- [25] T. G. Fischer, “Reinforcement learning in financial markets-a survey,” FAU Discussion Papers in Economics, Tech. Rep., 2018.
- [26] T. L. Meng and M. Khushi, “Reinforcement learning in financial markets,” *Data*, vol. 4, no. 3, p. 110, 2019.
- [27] D. Cliff, “Bbe: Simulating the microstructural dynamics of an in-play betting exchange via agent-based modelling,” *arXiv preprint arXiv:2105.08310*, 2021.
- [28] A. Dangi, “Financial portfolio optimization: computationally guided agents to investigate, analyse and invest!?” *arXiv preprint arXiv:1301.4194*, 2013.
- [29] B. Yilmaz and A. Selcuk-Kestel, “A stochastic approach to model housing markets: The us housing market case,” *Numerical Algebra Control and Optimization*, vol. 8, no. 4, 2018.
- [30] A. Carro, M. Hinterschweiger, A. Uluc, and J. D. Farmer, “Heterogeneous effects and spillovers of macroprudential policy in an agent-based model of the uk housing market,” *Industrial and Corporate Change*, vol. 32, no. 2, pp. 386–432, 2023.
- [31] R. May and P. Huang, “A multi-agent reinforcement learning approach for investigating and optimising peer-to-peer prosumer energy markets,” *Applied Energy*, vol. 334, p. 120705, 2023.