UNIVERSAL MECHANISM DESIGN FOR AI POLICY-MAKING

Edwin Zhang, Sadie Zhao, Tonghan Wang, Safwain Hossain
Department of Computer Science
Harvard University
Cambridge, MA 02134, USA
{ezhang,sadie_zhao,tonghan_wang,shossain}@g.harvard.edu

Stephan Zheng
Asari AI
San Francisco, CA 94016, USA
stephan@asari.ai

David C. Parkes, Milind Tambe & Yiling Chen
Department of Computer Science
Harvard University
Cambridge, MA 02134, USA
{parkes,tambe,chen50}@g.harvard.edu

ABSTRACT

Artificial Intelligence (AI) holds promise as a technology that can be used to improve government and economic policy-making. This paper introduces the **Universal Mechanism Design Game**, a general framework for the use of AI for automated policy-making. The framework extends mechanism design to capture a fully general economic environment, including voting on policy objectives, and gives a direction for the systematic analysis of government and economic policy through AI simulation. We highlight key open problems for future research that we hope can enable social-welfare objectives in AI-based policy, thereby promoting more ethical and responsible decision making.

1 INTRODUCTION

Macroeconomic policy formulation is a domain fraught with complexity, where traditional economic models provide limited foresight into the outcomes of policy decisions. Policy makers must not only understand the immediate implications of individual policies but also their aggregate and long-term effects. In addition, human policy-makers may not be incentive-aligned with the general public, and may prioritize lobbyist interests or reelection (de Figueiredo & Richter, 2014). In this light, AI-based approaches to policy design, that can simulate economies and take-up different objectives, hold potential for improved policy understanding and formulation .

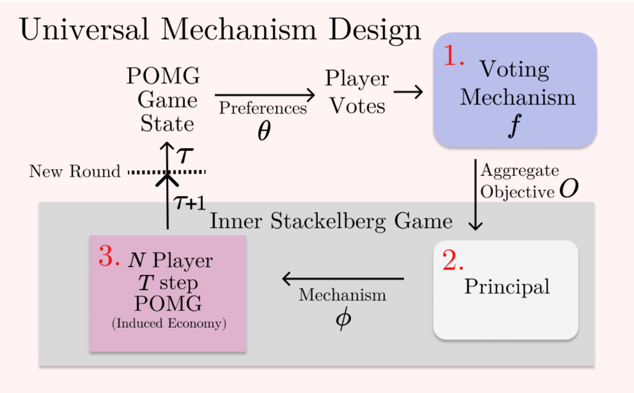

Figure 1: *The proposed game.* The process begins with voting, where preferences are reported to a voting mechanism (1). This defines an objective for the Principal who designs a mechanism (2) that parameterizes an N-player Partially Observable Markov Game (POMG). This POMG unfolds over several timesteps T (3). Following the POMG, game state information is extracted to initiate a new round of voting, thereby creating a feedback loop for adaptation and optimization of policy-making decisions, and with the last POMG game state used as the first game state of the new round. This whole process is repeated for τ timesteps.

In our model, we suggest to address the concern in regard to a misaligned policy-maker with "Voting on Values," this coupled with a Principal who seeks to achieve the suggested policy goals. . We capture the complexity of a general economic environment by modeling the economy as a Partially Observable Markov Game (POMG).

Ideally, a theoretical framework tailored for AI-led policymaking in complex economic systems should balance **expressivity** with **computational tractability**, while bringing **theoretical clarity**. In this paper, we propose a new universal mechanism design framework which makes progress towards these desiderata. We also discuss open problems within this framework, along with prospective solution concepts and algorithmic approaches, taking forward a dialogue on AI's application to macroeconomic policy design. By introducing this framework, we seek to foster a methodological evolution in the approach to economic modeling and mechanism design for policy goals, leveraging AI to elevate the capacity of policymakers and enhance economic resilience and governance effectiveness.

2 The Universal Mechanism Design Game

At a high level, we frame the economic design problem as a Stackelberg game between the policy designer and economic participants. The economic participants first vote for a given objective, or values to optimize for. Subsequently, the economic designer (leader) attempts to maximize this objective by designing the rules of an economic system, which induce an economic environment for the participants. We model this environment as a Partially Observable Markov Game (POMG) with the economic participants as the agents. We refer to this as the **Universal Mechanism Design Game** because it generalizes mechanism design in a number of ways; e.g., it involves voting on goals, and it involves the design of an economic policy for an economic environment in which agents take actions and report types (thus, this involves questions of moral hazard as well as adverse selection). We defer related work to Appendix A, preliminary definitions to Appendix B, and see Appendix C for a more detailed treatment and formalization of our model. We give a proof in Appendix D.

3 Challenges, Open Problems and Conclusion

Based on the AI-led economic policy-making framework presented, the following key open problems of our framework are proposed for further exploration: The **computational complexity** of the proposed game is high, as even the Dec-POMDP, a specific case of POMG, is known to be NEXP-Complete (Bernstein et al., 2000; Pynadath & Tambe, 2002). An important open challenge will be finding new theoretical insights within this problem setting, and exploiting them to make the proposed game tractable. Separately, **preference aggregation and democratic representation** in voting mechanisms is a complex challenge that requires advanced algorithms to reflect collective preferences while respecting minority views. The system must adapt to different cultural and decision-making paradigms, possibly by leveraging the capabilities of Large Language Models (LLMs) to imitate human-like decision-making (Hämäläinen et al., 2023; Park et al., 2023). To ensure responsible **AI governance and accountability**, oversight mechanisms must be established, incorporating both AI and human collaboration, with experts overseeing AI decisions. Furthermore, exploring the socioeconomic interactions within these systems is critical, especially in understanding and deriving the conditions for **convergence to desired equilibria** and the impact of dynamic shocks, ensuring systems can adapt and maintain stability amidst fluctuations. In addition, we leave the problem of deciding what **alternatives are presented for voting** open, and whether the alternatives should contain policy proposals in addition to just social welfare objectives. If policy proposals, this also raises questions regarding human ability to **evaluate alternate policy**—if all people have ever known is one society, it may be unreasonable to assume that they can judge alternative societies well. To some degree, this is captured in our model as the outer voting loop allows the participants to change their mind as their type evolves within the inner game. However, further serious consideration must be given to this issue. For a more detailed discussion of these and several other issues, please refer to Appendix E. In an era where AI is gaining increasing importance in helping shape governmental and economic policy decisions (Engstrom et al., 2020; House, 2023), understanding its influence through a rigorous framework has never been more critical. In this paper, we present a theoretical framework for simulation that merges mechanism design with AI to potentially help better inform economic policy-making. It tackles issues such as preference aggregation and equilibria in complex economic systems. Significant challenges, including democratic representation and accountability in AI-driven systems, are highlighted. We hope to engage interdisciplinary expertise and foster collaborative innovation, and aspire to help create AI systems that not only enhance economic resilience and governance effectiveness but also uphold democratic ideals and ethical standards.

References

Kenneth J. Arrow. *Social Choice and Individual Values*. Yale University Press, 2012. ISBN 978-0-300-17931-6. URL `https://www.jstor.org/stable/j.ctt1nqb90`.

Susan Athey and Ilya Segal. An efficient dynamic mechanism. *Econometrica*, 81(6):2463–2485, 2013.

Andrew Atkeson and Robert E Lucas Jr. On efficient distribution with private information. *The Review of Economic Studies*, 59(3):427–453, 1992.

Didier Aussel, Luce Brotcorne, Sébastien Lepaul, and Léonard von Niederhäusern. A trilevel model for best response in energy demand-side management. *European Journal of Operational Research*, 281(2):299–315, 2020.

David P Baron and David Besanko. Regulation and information in a continuing relationship. *Information Economics and policy*, 1(3):267–302, 1984.

Nicola Basilico, Stefano Coniglio, and Nicola Gatti. Methods for finding leader–follower equilibria with multiple followers. *arXiv preprint arXiv:1707.02174*, 2017.

Dirk Bergemann and Juuso Välimäki. The dynamic pivot mechanism. *Econometrica*, 78(2):771–789, 2010.

Dirk Bergemann and Juuso Välimäki. Dynamic mechanism design: An introduction. *Journal of Economic Literature*, 57(2):235–74, June 2019. doi: 10.1257/jel.20180892. URL `https://www.aeaweb.org/articles?id=10.1257/jel.20180892`.

Daniel S Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of markov decision processes, 2000.

Simon Board. Selling options. *Journal of Economic Theory*, 136(1):324–340, 2007.

Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

Herminia I Calvete and Carmen Galé. Linear bilevel multi-follower programming with independent followers. *Journal of Global Optimization*, 39(3):409–417, 2007.

Chi Cheng, Zhangqing Zhu, Bo Xin, and Chunlin Chen. A multi-agent reinforcement learning algorithm based on stackelberg game. In *2017 6th Data Driven Control and Learning Systems (DDCLS)*, pp. 727–732. IEEE, 2017.

Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 82–90, 2006.

Vincent Conitzer and Toby Walsh. *Barriers to Manipulation in Voting*, pp. 127–145. Cambridge University Press, 2016. doi: 10.1017/CBO9781107446984.007.

Pascal Courty and Li Hao. Sequential screening. *The Review of Economic Studies*, 67(4):697–717, 2000.

Michael J Curry, Uro Lyi, Tom Goldstein, and John P Dickerson. Learning revenue-maximizing auctions with differentiable matching. In *International Conference on Artificial Intelligence and Statistics*, pp. 6062–6073. PMLR, 2022.

John M. de Figueiredo and Brian Kelleher Richter. Advancing the Empirical Research on Lobbying. *Annual Review of Political Science*, 17(1):163–185, 2014. doi: 10.1146/annurev-polisci-100711-135308. URL `https://doi.org/10.1146/annurev-polisci-100711-135308`. _eprint: https://doi.org/10.1146/annurev-polisci-100711-135308.

Zhijian Duan, Haoran Sun, Yurong Chen, and Xiaotie Deng. A scalable neural network for dsic affine maximizer auction design. *arXiv preprint arXiv:2305.12162*, 2023.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning: Advances in differentiable economics. *Journal of the ACM*, Forthcoming 2023. First version, ICML 2019, pages 1706–1715. PMLR, 2019.

Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. Good Rationalizations of Voting Rules. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, September 2010.

David Freeman Engstrom, Daniel E. Ho, Catherine M. Sharkey, and Mariano-Florentino Cuéllar. Government by Algorithm: Artificial Intelligence in Federal Administrative Agencies, February 2020. URL `https://papers.ssrn.com/abstract=3551505`.

Peter Eso and Balazs Szentes. Dynamic contracting with adverse selection: an irrelevance result. Technical report, Citeseer, 2013.

Piotr Faliszewski and Jörg Rothe. *Control and Bribery in Voting*, pp. 146–168. Cambridge University Press, 2016. doi: 10.1017/CBO9781107446984.008.

Tanner Fiez, Benjamin Chasnov, and Lillian Ratliff. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning*, pp. 3133–3144. PMLR, 2020.

Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. ISSN 00029890, 19300972. URL `http://www.jstor.org/stable/2312726`.

Jiarui Gan, Edith Elkind, Sarit Kraus, and Michael Wooldridge. Mechanism design for defense coordination in security games. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 402–410, 2020.

Alex Gershkov and Benny Moldovanu. Dynamic revenue maximization with heterogeneous objects: A mechanism design approach. *American economic Journal: microeconomics*, 1(2):168–198, 2009.

Matthias Gerstgrasser and David C Parkes. Oracles & followers: Stackelberg equilibria in deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 11213–11236. PMLR, 2023.

Edward J Green. Lending and the smoothing of uninsurable income. *Contractual arrangements for intertemporal trade*, 1:3–25, 1987.

Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored mdps. In *Advances in neural information processing systems*, pp. 1523–1530, 2002a.

Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *ICML*, volume 2, pp. 227–234. Citeseer, 2002b.

John C. Harsanyi. Games with incomplete information played by ”bayesian” players, i-iii. part iii. the basic probability distribution of the game. *Management Science*, 14(7):486–502, 1968. ISSN 00251909, 15265501. URL `http://www.jstor.org/stable/2628894`.

The White House. Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence, October 2023. URL `https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artifici`

Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. Evaluating Large Language Models in Generating Synthetic HCI Research Data: a Case Study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pp. 1–19, New York, NY, USA, April 2023. Association for Computing Machinery. ISBN 978-1-4503-9421-5. doi: 10.1145/3544548.3580688. URL https://dl.acm.org/doi/10.1145/3544548.3580688.

Dmitry Ivanov, Iskander Safiulin, Igor Filippov, and Ksenia Balabaeva. Optimal-er auctions through attention. *Advances in Neural Information Processing Systems*, 35:34734–34747, 2022.

Albert Xin Jiang, Ariel D Procaccia, Yundi Qian, Nisarg Shah, and Milind Tambe. Defender (mis) coordination in security games. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*, pp. 4880–4889. PMLR, 2020.

Sham M Kakade, Ilan Lobel, and Hamid Nazerzadeh. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Operations Research*, 61(4):837–854, 2013.

Marek Kapicka. Efficient allocations in dynamic private information economies with persistent shocks: A first-order approach. *Review of Economic Studies*, 80(3):1027–1054, 2013.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.

Hangyu Mao, Wulong Liu, Jianye Hao, Jun Luo, Dong Li, Zhengchao Zhang, Jun Wang, and Zhen Xiao. Neighborhood cognition consistent multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7219–7226, 2020.

Parinaz Naghizadeh and Mingyan Liu. Voluntary participation in cyber-insurance markets. In *Workshop on the Economics of Information Security (WEIS)*, 2014.

Noam Nisan and Amir Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35(1):166–196, April 2001. ISSN 0899-8256. doi: 10.1006/game.1999.0790. URL https://www.sciencedirect.com/science/article/pii/S089982569990790X.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior, August 2023. URL http://arxiv.org/abs/2304.03442. arXiv:2304.03442 [cs].

Ariel D. Procaccia. Can approximation circumvent gibbard-satterthwaite? In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pp. 836–841. AAAI Press, 2010.

Ariel D. Procaccia. Cake cutting algorithms. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D.Editors Procaccia (eds.), *Handbook of Computational Social Choice*, pp. 311–330. Cambridge University Press, 2016. doi: 10.1017/CBO9781107446984.014.

David V. Pynadath and Milind Tambe. Multiagent teamwork: Analyzing the optimality and complexity of key theories and models. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2*, AAMAS '02, pp. 873–880, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581134800. doi: 10.1145/544862.544946. URL https://doi.org/10.1145/544862.544946.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4292–4301, 2018.

Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Tuomas Sandholm. Automated Mechanism Design: A New Application Area for Search Algorithms. In Francesca Rossi (ed.), *Principles and Practice of Constraint Programming – CP 2003*, Lecture Notes in Computer Science, pp. 19–36, Berlin, Heidelberg, 2003. Springer. ISBN 978-3-540-45193-8. doi: 10.1007/978-3-540-45193-8_2.

Weiran Shen, Pingzhong Tang, and Song Zuo. Automated Mechanism Design via Neural Networks, May 2021. URL `http://arxiv.org/abs/1805.03382`. arXiv:1805.03382 [cs].

Zhenyu Shi, Runsheng Yu, Xinrun Wang, Rundong Wang, Youzhi Zhang, Hanjiang Lai, and Bo An. Learning expensive coordination: An event-based deep rl approach. In *International Conference on Learning Representations*, 2019.

Tianmin Shu and Yuandong Tian. M^{3}RL: Mind-aware multi-agent management reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896, 2019.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Kurian Tharakunnel and Siddhartha Bhattacharyya. Leader-follower semi-markov decision problems: theoretical framework and approximate solution. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 111–118. IEEE, 2007.

William Thomson. *Introduction to the Theory of Fair Allocation*, pp. 261–283. Cambridge University Press, 2016. doi: 10.1017/CBO9781107446984.012.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020a.

Kai Wang, Lily Xu, Andrew Perrault, Michael K Reiter, and Milind Tambe. Coordinating followers to reach better equilibria: End-to-end gradient descent for stackelberg games. *arXiv preprint arXiv:2106.03278*, 2021a.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning*, 2020b.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.

Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. Dop: Off-policy multi-agent decomposed policy gradients. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021c.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Noah Williams. Persistent private information. *Econometrica*, 79(4):1233–1275, 2011.

Huaqing Zhang, Yong Xiao, Lin X Cai, Dusit Niyato, Lingyang Song, and Zhu Han. A multi-leader multi-follower stackelberg game for resource management in lte unlicensed. *IEEE Transactions on Wireless Communications*, 16(1):348–361, 2016.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*, 2020.

Appendix

A Related Work

The proposed framework resides at the convergence of various strands of research, including but not limited to Stackelberg game learning, multi-agent reinforcement learning, dynamic mechanism design, and computational social choice. In this section, we delve into a comprehensive exploration of its connections with prior research.

A.1 Stackelberg Game

From the perspective of the principal, it plays a Stackelberg game with agents of different types. Stackelberg games model many real-world problems that exhibit a hierarchical order of play by different players, including taxation Zheng et al. (2020), security games Jiang et al. (2013); Gan et al. (2020), and commercial decision-making Naghizadeh & Liu (2014); Zhang et al. (2016); Aussel et al. (2020). In the simplest case, a Stackelberg game contains one leader and one follower. For these games with discrete action spaces, Conitzer & Sandholm (2006) show that linear programming approaches can obtain Stackelberg equilibria in polynomial time in terms of the pure strategy space of the leader and follower. To find Stackelberg equilibria in continuous action spaces, Jin et al. (2020); Fiez et al. (2020) propose the notion of local Stackelberg equilibria and characterize them using first- and second-order conditions. Moreover, Jin et al. (2020) show that common gradient descent-ascent approaches can converge to local Stackelberg equilibria (except for some degenerate points) if the learning rate of the leader is much smaller than that of the follower. Fiez et al. (2020) give update rules with convergence guarantees. Different from these works, in this paper, we consider Stackelberg games with multiple followers.

More sophisticated than its single-follower counterpart, unless the followers are independent Calvete & Galé (2007), computing Stackelberg equilibria with multiple followers becomes NP-hard even when assuming equilibria with a special structure for the followers Basilico et al. (2017). Recently, Wang et al. (2021a) propose to deal with an arbitrary equilibrium which can be reached by the follower via differentiating though it. Gerstgrasser & Parkes (2023) proposes a meta-learning framework among different policies of followers to enable fast adaption of the principal.

Multi-agent reinforcement learning holds the promise to extend Stackelberg learning to more general and realistic problems. Tharakunnel & Bhattacharyya (2007) propose Leader-Follower Semi-Markov Decision Process to model sequential Stackelberg learning problem. Cheng et al. (2017) propose Stackelberg Q-learning but without any convergence guarantee. Shu & Tian (2019); Shi et al. (2019) study leader-follower problems from an empirical perspective, where the leader learn deep models to predict the followers' behavior. To the best of our knowledge, a multi-agent reinforcement learning method with a desired convergence property, even when using linear function approximation, is still absent from the literature.

A.2 Multi-Agent Reinforcement Learning

Another important component of the proposed framework is the followers' behavior learning. Deep multi-agent reinforcement learning algorithms have witnessed significant advances in recent years. COMA Foerster et al. (2018), MADDPG Lowe et al. (2017), PR2 Wen et al. (2019), and DOP Wang et al. (2021c) study the problem of policy-based multi-agent reinforcement learning. They use a (decomposed) centralized critic to calculate gradients for decentralized actors. Value-based algorithms decompose the joint value function into individual utility functions in order to enable efficient optimization and decentralized execution. VDN Sunehag et al. (2018), QMIX Rashid et al. (2018), and QTRAN Son et al. (2019) progressively expand the representation capabilities of the mixing network. QPLEX Wang et al. (2020a) implements the full IGM class Son et al. (2019) by encoding the IGM principle into a duplex dueling network architecture. Weighted QMIX Rashid et al. (2020) proposes weighted projection to decompose any joint action-value functions. There are other works that investigate into MARL from the perspective of coordination graphs Guestrin et al. (2002b;a); Böhmer et al. (2020) and communication Singh et al. (2019); Mao et al. (2020).

Among these works, role-based learning frameworks Wang et al. (2020b; 2021b) are the most related to our work. They learn the types (or the roles) of different agents from scratch based on the shared reward signal. The objective is to automatically decompose the task and reduce the learning complexity by learning sub-task specific polices. However, these works are majorly studied in the setting of the Decentralized Partially Observable Markov Decision Process (Dec-POMDP), and are thus different from our work by two points: (1) The reward is shared among agents; and (2) The dynamics, including reward and transition dynamics, are static in these models.

A.3 Dynamic Mechanism Design

The study of optimal dynamic mechanism design has its roots in the pioneering work of Baron & Besanko (1984), who employed a first-order approach in a two-period single-agent scenario to determine the best mechanism for regulating a natural monopoly. They described the ideal distortions using an "informativeness measure," akin to an impulse response function in a two-period context. Courty & Hao (2000) explored a similar model to examine optimal advanced ticket sales, also establishing certain conditions for the workability of a dynamic allocation rule. Expanding on this, Eso & Szentes (2013) extended their analysis to multiple agents, focusing on the optimal revelation of information in auctions.

The exploration of optimal mechanisms in a multi-agent setup with an infinite time horizon was initially undertaken by Board (2007). He built upon the work of Eso & Szentes (2013) in a scenario where the timing of allocation is endogenous, involving the sale of options by the principal. Subsequently, Kakade et al. (2013) addressed a category of allocation problems encompassing Board's model and our bandit auctions. They demonstrated that the optimal mechanism resembles a virtual version of the dynamic pivot mechanism proposed by Bergemann & Välimäki (2010).

The focus of existing research on optimal dynamic mechanisms in specific scenarios is due to the complexity of formulating a manageable optimization problem with implementable allocation rules. Typically in the realm of designing efficient mechanisms aimed at maximizing expected surplus, the allocation rules are predetermined. Developments in this area include dynamic adaptations of the static Vickrey-Clarke-Groves and expected-externality mechanisms for broad, quasilinear private-value settings, as introduced by Bergemann & Välimäki (2010) and Athey & Segal (2013). There is also an expanding body of work examining both efficient and profit-maximizing dynamic mechanisms in situations where each participant only acquires a single piece of private information, but the participants or objects are introduced randomly over time, as seen in studies like Gershkov & Moldovanu (2009). Although the incentive compatibility in these models is analyzed statically, dynamic complexities arise from the timing challenges that the designer faces. For further insights, the Bergemann & Välimäki (2019) survey offers an in-depth overview of these topics.

An expanding body of research is examining dynamic mechanisms that are both efficient and aimed at maximizing profits in scenarios where each participant has access to a single piece of private information, with the arrival of agents or items occurring randomly over time, as illustrated in the work of Gershkov & Moldovanu (2009). While the analysis of incentive compatibility in these models remains static, dynamic complexities arise due to the challenges related to optimal timing that the designer encounters. Different from these works, in this paper, we study multi-agent reinforcement learning under dynamic mechanism in this work.

Dynamic insurance and optimal taxation are related areas of study. The initial research in this field, inspired by the works of Green (1987) and Atkeson & Lucas Jr (1992), operated under the assumption of independently and identically distributed types. More recent studies have shifted focus to the concept of persistent private information, with notable contributions from Kapicka (2013). For more information, one can refer to Williams (2011).

A.4 Automated and Algorithmic Mechanism Design

Automated Mechanism Design has a rich history, and was first introduced by Sandholm (2003), where search algorithms are used to computationally create specific rule sets (mechanisms) for games that lead to desirable outcomes even when participants act in self-interest. This approach hypothesized that automated approaches would someday outperform traditional manual designs, be applicable to a broader range of problems, and circumvent economic impossibility results, by transferring the burden of design from humans to machines. More recently, the work of automated mechainsm design has been advanced through deep learning, in the framework known as *differen-*

tiable economics. Dütting et al. (Forthcoming 2023) use deep neural network to learn the allocation and payment rules of auctions. Since then, a line of follow-up work has been introduced, extending the framework to make the architecture more powerful and general (Shen et al., 2021; Ivanov et al., 2022; Duan et al., 2023; Curry et al., 2022).

Algorithmic mechanism design was later introduced by Nisan & Ronen (2001). It focuses on distributed computational systems where the participating entities, termed as agents, act out of self-interest, potentially manipulating the system to their advantage. Consequently, algorithms in such settings must be designed with the strategic behavior of agents in mind. More specifically, the field of algorithmic mechanism design explores how to devise algorithms or protocols that align the incentives of self-interested agents with the intended outcome of the algorithm or system designer. This is pursued by incorporating payments or other incentives within the system to encourage agents to act in a manner that leads to socially desirable outcomes. For example, in the context of task scheduling problems, where multiple agents may possess different private information regarding task execution times, an algorithmic mechanism could be designed to motivate truthful reporting of this private information. Such mechanisms have broad applications ranging from network routing, where data packets need to be sent through various routers, to auction systems, where resources are allocated among bidders with private valuations.

A.5 Computational Social Choice

Computational social choice is an interdisciplinary field combining computer science and social choice theory, focusing on the application of computational techniques to social choice mechanisms (such as voting rules or fair allocation procedures) and the theoretical analysis of these mechanisms with computational tools (Brandt et al., 2016). The field draws on methods from complexity theory, artificial intelligence, and algorithm design, aiming to understand the computational difficulty associated with various social choice problems, and to develop algorithms for solving them (Conitzer & Walsh, 2016; Faliszewski & Rothe, 2016). A fundamental aspect is the study of manipulative behavior in elections and other collective decision-making processes, as well as the design of systems resistant to manipulation (Elkind et al., 2010; Procaccia, 2010). Additionally, computational social choice encompasses the fair distribution of resources, often involving complex allocation problems (Thomson, 2016; Procaccia, 2016). With its roots in the seminal work of Gale & Shapley (1962) and others, the field has grown significantly in the 21st century, finding applications in areas such as multi-agent systems and recommendation systems Brandt et al. (2016).

B Preliminaries

Here we give some preliminaries on several foundational games and solution concepts that we build upon.

Definition 1. *A $(n+1)$-player* **Stackelberg-Nash Game** $\mathcal{S} = (n, m, \mathcal{X}, \mathcal{Y}, \boldsymbol{u})$ *comprises one player called the* **leader** *and $n \in \mathbb{N}\backslash\{0\}$ players called* **followers**. *In a Stackelberg-Nash game, the leader first commits to an action $\boldsymbol{x} \in \mathcal{X}$ from action space $\mathcal{X} \subset \mathbb{R}^m$. Then, having observed the leader's action, each follower $i \in [n]$, responds with an action y_i in their action space $\mathcal{Y}_i \subset \mathbb{R}^m$. We define the followers' joint action space $\mathcal{Y} = \bigtimes_{i\in[n]} \mathcal{Y}_i$. We refer to a collection of actions $\boldsymbol{y} = (y_1, \cdots, y_n) \in \mathcal{Y}$ as a followers' action profile, and to a collection $(x, y) \in \mathcal{X} \times \mathcal{Y}$ comprising an action for the leader and a follower action profile as simply an action profile.*

After all players choose an action, the leader receives payoff $u_o(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}$, while each follower $i \in [n]$ receives payoff $u_i(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}$. Each player $i \in [n]$ aims to maximize her payoff, and the leader aims to maximize her payoff assuming the followers will best respond.

*For all followers $i \in [n]$, we define the δ-***best-response correspondence** $\mathcal{BR}_i^\delta(\boldsymbol{x}, y_{-i}) = \{y_i \in \mathcal{Y}_i \mid u_i(\boldsymbol{x}, \boldsymbol{y}) \geqslant \max_{y_i \in \mathbf{Y}_i} u_i(\boldsymbol{x}, (y_i, y_{-i})) - \delta\}$ *and the* **joint δ-best-response correspondence** $\mathcal{BR}^\delta(\boldsymbol{x}, \boldsymbol{y}) = \bigtimes_{i\in[n]} \mathcal{BR}_i^\delta(\boldsymbol{x}, y_{-i})$.

Fixing the leader's action $\boldsymbol{x} \in \mathcal{X}$, a Stackelberg Nash game $\mathcal{S}$ induces a **lower-level Nash game** $\mathcal{G}^\mathcal{S} = (n, m, \mathcal{Y}, \boldsymbol{u}_{-0}(\boldsymbol{x}, \cdot))$ *among the followers.*

Definition 2. *A* (ε, δ)**-strong Stackelberg-Nash equilibrium (SSNE)** *in a Stackelberg-Nash game $\mathcal{S} = (n, m, \mathcal{X}, \mathcal{Y}, \boldsymbol{u})$ is an action profile $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathcal{X} \times \mathcal{Y}$ such that $u_0(\boldsymbol{x}^*, \boldsymbol{y}^*) \geqslant \max_{x\in\mathcal{X}} \max_{\mathcal{BR}^\delta} u_0(\boldsymbol{x}, \boldsymbol{y}) - \varepsilon$ and $u_i(\boldsymbol{x}^*, \boldsymbol{y}^*) \geqslant \max_{y_i \in \mathcal{Y}_i} u_i(\boldsymbol{x}^*, (y_i, y^*_{-i})) - \delta$, for all $i \in [n]$.*

Definition 3. *A* ***Partially Observable Markov Game*** *(POMG) $\mathcal{M}$ with n agents is a tuple $(S, A, T, r, \Omega, O, \gamma, \mu_0)$. Here,*

- *S is a shared state space for all agents;*
- *$A = \bigtimes_{i\in[n]} A_i$ is the joint action space;*
- *$T : S \times S \times A \to [0, 1]$ is a stochastic transition function;*
- *$r : S \times A \to \mathbb{R}^n$ is the reward function with $r = (r_1, \cdots, r_n)$;*
- *$\Omega = \bigtimes_{i\in[n]} \Omega_i$ is the joint observation space;*
- *$O : \Omega \times S \times A \to [0, 1]$ is the stochastic observation function;*
- *$\gamma \in [0, 1)$ is a discount factor;*
- *$\mu_0 \in \Delta(S)$ is the initial state distribution.*

An agent's behavior in this game is characterized by its policy $\pi_i : \Omega \to A$, which maps observations to actions.

Definition 4. *A $(n + 1)$-player* **Stackelberg-Markov-Nash Game** $\mathcal{S} = (n, m, \Phi, \Pi, \boldsymbol{u})$ *comprises one player called the* **leader** *and $n \in \mathbb{N}\backslash\{0\}$ players called* **followers**. *In a Stackelberg-Markov-Nash game, the leader first commits to an action $\phi \in \Phi$ from action space $\Phi \subset \mathbb{R}^m$ which induces a n-player* **low-level (Partially Observable) Markov Game** $\mathcal{M}^\phi = (S, A, T^\phi, r^\phi, \Omega, O^\phi, \gamma^\phi, \mu_0^\phi)$. *Then, having observed the leader's action, each follower $i \in [n]$, responds with an policy $\pi_i : \Omega \to A_i$ in their policy space Π_i. We define the followers' joint action space $\Pi = \bigtimes_{i\in[n]} \Pi_i$. We refer to a collection of policies $\pi = (\pi_1, \cdots, \pi_n) \in \Pi$ as a followers' policy profile.*

After all players choose an action/policy, the leader receives payoff $u_o(\phi, \pi) \in \mathbb{R}$, while each follower $i \in [n]$ receives payoff $u_i(\phi, \pi) = \mathbb{E}^{\mathcal{M}^\phi, \pi}[\sum_{t=0}^{\infty} (\gamma^\phi)^t r^\phi(s^t, a^t)] \in \mathbb{R}$. Each player $i \in [n]$ aims to maximize her payoff, and the leader aims to maximize her payoff assuming the followers will best respond.

For all followers $i \in [n]$, we define the **δ-best-response correspondence** $\mathcal{BR}_i^\delta(\phi, \pi_{-i}) = \{\pi_i \in \Pi_i \mid u_i(\phi, \pi) \geqslant \max_{\pi_i \in \Pi_i} u_i(\phi, (\pi_i, \pi_{-i})) - \delta\}$ *and the* **joint δ-best-response correspondence** $\mathcal{BR}^\delta(\phi, \pi) = \bigtimes_{i \in [n]} \mathcal{BR}_i^\delta(\phi, \pi_{-i})$.

Definition 5. *A* **(ε, δ)-strong Stackelberg-Markov-Nash equilibrium (SSMNE)** *in a Stackelberg-Markov-Nash game $\mathcal{S} = (n, m, \Phi, \Pi, \boldsymbol{u})$ is an action profile $(\phi^*, \pi^*) \in \Phi \times \Pi$ such that $u_0(\phi^*, \pi^*) \geqslant \max_{\phi \in \Phi} \max_{\mathcal{BR}^\delta} u_0(\phi, \pi) - \varepsilon$ and $u_i(\phi^*, \pi^*) \geqslant \max_{\pi_i \in \Pi_i} u_i(\phi^*, (\pi_i, \pi_{-i}^*)) - \delta$, for all $i \in [n]$.*

Definition 6. *A* **(One-Shot) Optimal Mechanism Design** *problem $\mathcal{P} = (n, m, \mathcal{T}, \mathcal{O}, \boldsymbol{t}, \boldsymbol{u}, u_0)$ comprises of n agents, each $i \in [n]$ owns a private type $\boldsymbol{t}_i \in \mathcal{T}_i$ from a set of possible types $\mathcal{T}_i \subset \mathbb{R}^m$. An agent's preferences over outcomes $\boldsymbol{o} \in \mathcal{O}$, for a set $\mathcal{O}$ of outcomes, can be expressed in terms of a utility function that is parameterized by the type. Let $u_i(\boldsymbol{o}, \boldsymbol{t}_i)$ denote the utility of agent i for outcome $\boldsymbol{o} \in \mathcal{O}$ given type $\boldsymbol{t}_i$. A strategy (policy in RL) $s_i : \mathcal{T}_i \to \mathcal{A}_i$ is a complete decision rule, that defines the action an agent will select in every distinguishable state of the world. Let $a_i = s_i(\boldsymbol{t}_i) \in \mathcal{A}_i$ denote the action of agent i given type $\boldsymbol{t}_i$, where $\mathcal{A}_i$ is the set of all possible actions available to agent i.*

A **mechanism** *$M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ defines the set of actions $\mathcal{A}_i$ available to each agent i, and an outcome rule $g : \bigtimes_{i \in [n]} \mathcal{A}_i \to \mathcal{O}$, such that $g(\boldsymbol{a})$ is the outcome implemented by the mechanism for action profile $\boldsymbol{a} = (a_1, \cdots, a_n)$.*

A mechanism $M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ is a **direct-revelation mechanism** *if $\mathcal{A}_i = \mathcal{T}_i$ for all $i \in [n]$, and we can denote it by $M = (g)$. Moreover, a* **mechanism** *$M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ will induce a game, in which each agent i chooses strategy a_i to maximize their utility $u_i(g(a_i, a_{-i}), \boldsymbol{t}_i)$. A direct-revelation mechanism M is Nash (resp. Bayesian-Nash, dominant-strategy Nash)* **incentive-compatible** *if the true type profile $\boldsymbol{t}$ is a is Nash (resp. Bayesian-Nash, dominant-strategy Nash) equilibrium of its induced game. For a full introduction to the Bayesian-Nash solution concept, please refer to Harsanyi (1968).*

$u_0 : \mathcal{M} \times \bigtimes_{i \in [n]} \mathcal{A}_i \to \mathbb{R}$ is a principal objective function, where $u_0(M, \boldsymbol{a})$ represents the expected utility/revenue of the when mechanism designer chooses mechanism M and agents choose action profile $\boldsymbol{a}$.

The goal of the mechanism designer is to design a mechanism $M = (\{\mathcal{A}_i\}_{i \in [n]}, g) \in \mathcal{M}$ that maximizes $u_0(M, \boldsymbol{a}^*)$, where strategy profile $\boldsymbol{a}^* = (a_1^*, \cdots, a_n^*)$ is an (Nash, Bayesian-Nash, dominant-strategy) equilibrium to the game induced by M.

Definition 7. *A* **(One-Shot) Optimal Mechanism Design with Stochastic Agents** *problem $\mathcal{P} = (n, m, \mathcal{T}, \mathcal{O}, \boldsymbol{u}, p, u_0)$ comprises of n agents, each $i \in [n]$ owns a private type from a set of possible types $\mathcal{T}_i$ that may change over a infinite time horizon. In essence, at each time step t, agent i owns a private type $\boldsymbol{t}_i^{(t)} \in \mathcal{T}_i$. An agent's preferences over outcomes $\boldsymbol{o} \in \mathcal{O}$, for a set $\mathcal{O}$ of outcomes, can be expressed in terms of a utility function that is parameterized by the type. Let $u_i(\boldsymbol{o}, \boldsymbol{t}_i)$ denote the utility of agent i for outcome $\boldsymbol{o} \in \mathcal{O}$ given type $\boldsymbol{t}_i$. The types of all agent evolve based on current types and outcome stochastically, according to a* **joint type transition** *probability function $p : \mathcal{T} \times \mathcal{T} \times \mathcal{O} \to [0, 1]$. A strategy $s_i : \mathcal{T}_i \to \mathcal{A}_i$ is a complete decision rule that defines the action an agent for a given state. Let $a_i = s_i(\boldsymbol{t}_i) \in \mathcal{A}_i$ denote the action of agent i given type $\boldsymbol{t}_i$, where $\mathcal{A}_i$ is the set of all possible actions available to agent i.*

A **mechanism** *$M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ defines the set of actions $\mathcal{A}_i$ available to each agent i, and an outcome rule $g : \bigtimes_{i \in [n]} \mathcal{A}_i \to \mathcal{O}$, such that $g(\boldsymbol{s})$ is the outcome implemented by the mechanism for action profile $\boldsymbol{a} = (a_1, \cdots, a_n)$.*

A mechanism $M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ is a **direct-revelation mechanism** *if $\mathcal{A}_i = \mathcal{T}_i$ for all $i \in [n]$, and we can denote it by $M = (g)$. Moreover, a mechanism $M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ will induce a POMG, in which each agent aims to maximize their discounted cumulative utility over time.*

$u_0 : \mathcal{M} \times \bigtimes_{i \in [n]} \mathcal{S}_i \to \mathbb{R}$ is a principal objective function, where $u_0(M, \boldsymbol{s})$ represents the expected utility/revenue of the when mechanism designer chooses mechanism M and agents choose policy profile $\boldsymbol{s}$.

The goal of the mechanism designer is to design a mechanism $M = (\{\mathcal{A}_i\}_{i \in [n]}, g) \in \mathcal{M}$ that maximizes $u_0(M, \boldsymbol{s}^*)$, where strategy profile $\boldsymbol{s}^* = (s_1^*, \cdots, s_n^*)$ is an Nash equilibrium to the POMG induced by M.

Definition 8. *A* **Dynamic Mechanism Design (DMD)** *problem $\mathcal{P} = (n, m, \mathcal{T}, \mathcal{O}, \boldsymbol{u}, p, u_0, \mathcal{H})$ comprises of n agents, each $i \in [n]$ owns a private type from a set of possible types $\mathcal{T}_i$ that may change over a infinite time horizon. In essence, at each time step t, agent i owns a private type $\boldsymbol{t}_i^{(t)} \in \mathcal{T}_i$. An agent's preferences over outcomes $\boldsymbol{o} \in \mathcal{O}$, for a set $\mathcal{O}$ of outcomes, can be expressed in terms of a utility function that is parameterized by the type. Let $u_i(\boldsymbol{o}, \boldsymbol{t}_i)$ denote the utility of agent i for outcome $\boldsymbol{o} \in \mathcal{O}$ given type $\boldsymbol{t}_i$. The types of all agent evolve based on current types and outcome stochastically, according to a* **joint type transition** *probability function $p : \mathcal{T} \times \mathcal{T} \times \mathcal{O} \to [0, 1]$. Moreover, we can define the public history space at time step t as $\mathcal{H}^{(t)} = \{(\boldsymbol{a}^{(s)}, \boldsymbol{o}^{(s)})_{s<t} \mid \boldsymbol{a}^{(s)} \in \mathcal{A}, \boldsymbol{o}^{(s)} \in \mathcal{O}\}$ and the complete history space as $\bigcup_{t \in \mathbb{N}} \mathcal{H}^{(t)}$.*

A strategy $s_i : \mathcal{H} \times \mathcal{T}_i \to \mathcal{A}_i$ is a a function that defines the action an agent will select given a history. Let $s_i(\boldsymbol{h}, \boldsymbol{t}_i) \in \mathcal{A}_i$ denote the action of agent i given type $\boldsymbol{t}_i$ and history $\boldsymbol{h}$, where $\mathcal{A}_i$ is the set of all possible actions available to agent i.

A **mechanism** *$M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ defines the set of actions $\mathcal{A}_i$ available to each agent i, and an outcome rule $g : \mathcal{H} \times \bigtimes_{i \in [n]} \mathcal{A}_i \to \mathcal{O}$, such that $g(\boldsymbol{a})$ is the outcome implemented by the mechanism for action profile $\boldsymbol{a} = (a_1, \cdots, a_n)$ given the history observed by the mechanism. A mechanism $M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ is a* **direct-revelation mechanism** *if $\mathcal{S}_i = \mathcal{T}_i$ for all $i \in [n]$, and we can denote it by $M = (g)$.*

$u_0 : \mathcal{M} \times \bigtimes_{i \in [n]} \mathcal{S}_i \to \mathbb{R}$ is a principal objective function, where $u_0(M, \boldsymbol{s})$ represents the expected utility/revenue of the when mechanism designer chooses mechanism M and agents choose policy profile $\boldsymbol{s}$.

The goal of the mechanism designer is to design a mechanism $M = (\{\mathcal{A}_i\}_{i \in [n]}, g) \in \mathcal{M}$ that maximizes $u_0(M, \boldsymbol{s}^*)$, where $\boldsymbol{s}^* = (s_1^*, \cdots, s_n^*)$ is the action profile chosen by all agents to maximize their cumulative utility over time.

C Details of Universal Mechanism Design Game

Here we give the formal definition of our game, as well as a more detailed analysis and breakdown of our proposed game.

Definition 9. *A* **Universal Mechanism Design Game** $\mathcal{S} = (\Phi, P, \phi_0, D, \delta, \Theta, \mathcal{O}, f)$ *is a one-leader-n-follower online Stackelberg-Markov-Nash Game, where*

- $\Phi \subseteq \mathbb{R}^k$ is the economic designer action space;
- $P : \Phi \mapsto \mathcal{M}^\phi$ is a policy implementation map that maps from an economic designer action $\phi \in \Phi$ to a parameterized POMG $\mathcal{M}^\phi = (S, A^\phi, T^\phi, r^\phi, \Omega^\phi, O^\phi, \gamma^\phi, \mu_0^\phi)$;
- $\phi_0 \in \Phi$ is some initial action;
- $D : \Phi \times \Phi \mapsto \mathbb{R}_{\geqslant 0}$ is a divergence measure on the leader action space;
- $\delta > 0$ is the divergence constraint;
- $\Theta \subseteq \mathbb{R}^{(n+1)\times m}$ is the type space. Note that since we define the first row of a specific type instantiation $\theta \in \Theta$ to be the type of the principal, the Θ has $(n+1)$ rows. We thus refer to Θ_1 to be the type space of the principal and Θ_{-1} to be the type space of all participants;
- $\mathcal{O} = \{\mathcal{O}_i\}_{i\in[n]}$ is some finite set of predefined social welfare functions, where each $\mathcal{O}$ maps $\Phi \times \Pi \mapsto \mathbb{R}$. We give examples of several possible choices of objectives below in Appendix C. Π here refers to the set of all possible policy profiles in the parameterized POMG.
- $f : \Theta \mapsto \mathcal{O}$ is a social choice function representing the voting mechanism.

Both the infinite-horizon and finite-horizon version of the Universal Mechanism Design Game can be considered. Note that we do not need to introduce a discount factor for the infinite-horizon version, as our model's objective changes at each step, and thus we consider the Principal only maximizing for the objective at the current timestep. In the finite horizon case, we add an additional time horizon $\mathcal{T}$.

The Universal Mechanism Design Game can be cleanly divided into a outer **Voting** model and **Inner** game, which is played with the Principal's objective determined by the Voting model.

Definition 10. *The* **Voting** *model is defined as* $\mathcal{S} = (\mathcal{O}, f, \Theta)$.

We use the standard axiomatic model (Arrow, 2012), where $\mathcal{O}$ is the set of alternatives, f is the social choice function, and Θ is the set of all preference profiles. Intuitively, as a specific agent i's type θ_i for row i in $\theta \in \Theta$, can be thought of as some latent vector which represents the agent's values. This type contains all information necessary for recovering a partial ordering over alternatives, a more specific way of defining preferences. The goal of the Voting model is then to define a objective for the Principal to optimize, given these types. To do so, the outer model defines a Voting Mechanism f and asks the players for a preference report $\theta_{-1} \in \Theta_{-1}$, which does not necessarily have to be truthful. The Voting mechanism then computes the objective $\mathcal{O} = f(\theta_1, \theta_{-1})$ as a result of the vote. Here, we set $\theta_1 \in \Theta_1$ to be the preferences of the Principal. Importantly, the objective function includes the full θ, which allows expressing preferences of the principal if one wished to encoded a form of "moral objectivity", or other biases. We also make this modeling choice for generality, as it allows our model to express mechanisms such as auctions where the objective of the principal may be entirely selfish and not depend at all on the participant's types.

Social Welfare Examples. Examples of social welfare functions that could be included in the voting set are the Utilitarian objective

$$\mathcal{O}(\phi, \pi) = \sum_i J(\pi_i^\phi) \tag{1}$$

where J is the expected discounted return $J = \sum_t (\gamma^\phi)^t r_i^\phi(s_t, a_{i,t}, a_{-i,t})$, π is the tuple of all agents $\pi = (\pi_i)_{i\in[n]}$, and π_i are singular agents that map $\Omega_i^\phi \rightarrow A_i^\phi$. Other possible choices include the

Nash Welfare objective:

$$\mathcal{O} = \left(\Pi_i J(\pi_i^\phi)\right)^{1/n} \tag{2}$$

As well as the Egalitarian objective:

$$\mathcal{O} = \min_i J(\pi_i^\phi) \tag{3}$$

These are perhaps the most commonly discussed objectives, but bespoke or custom welfare functions could also be considered and added to the set of alternatives.

Definition 11. *The* **Inner** *game $I = (\Phi, P, D, \delta, \phi_0)$ is a Stackelberg-Markov-Nash Game.*

The Inner game is played subsequently after the Voting model, and can be thought of as a single timestep of the full game. The economic designer (leader) will choose action $\phi \in \Phi$ which induces a parameterized **Induced Economy** $\mathcal{M}^\phi = (S, A^\phi, T^\phi, r^\phi, \Omega^\phi, O^\phi, \gamma^\phi, \mu_0^\phi)$ through the policy implementation map $P : \phi \mapsto \mathcal{M}^\phi$.

Thus, the objective of the leader in the Inner game is then to design a POMG, given the objective $\mathcal{O}$ decided prior in the Voting model.

$$\begin{aligned} &\max_\phi \mathcal{O}(\phi, \pi; \theta) \\ \text{s.t.} \quad &D(\phi_0, \phi) \leqslant \delta \\ &\mu_0^{P(\phi)} = \Delta(s_T) \end{aligned} \tag{4}$$

Again, π here is the tuple of all agents $\pi = (\pi_i)_{i \in [n]}$, and π_i individual agents that map $\Omega_i^\phi \to A_i^\phi$. Our notation $\mu_0^{P(\phi)}$ denotes the μ_0 of the tuple $P(\phi)$, and $\Delta(s_t)$ refers a Delta Dirac distribution centered on s_T. Therefore, the second constraint $\mu_0^{P(\phi)} = \Delta(s_T)$ forces the ϕ to choose a POMG that has the same initial state as the terminal state of the last round, so that contuinity is kept between rounds.

It is also important to note adding a direct reporting of preferences to the model does not increase the expressivity of our game, as in principle one could encode this report within the voting mechanism as well.

Lastly, we remark that this constrained optimization can also be transformed into an unconstrained problem by using an additional reparameterization $f : \xi \mapsto \hat{\Phi}$, where $\xi \in \Xi := \mathbb{R}^n$ and $\hat{\Phi} := \{\phi \mid D(\phi_0, \phi) \leqslant \delta\}$. The optimization can then proceed in $\mathbb{R}^n$ with no constraints. In this case, the Inner subgame would reduce to $I = (\Xi, P)$.

Definition 12. *The* **Induced Economy** *is a Partially Observable Markov Game $\mathcal{M}^\phi = (S, A^\phi, T^\phi, r^\phi, \Omega^\phi, O^\phi, \gamma^\phi, \mu_0^\phi)$.*

Finally, the **Induced Economy** is defined as the POMG produced as the output of the Inner Game. Agents within the POMG interact with one another and attempt to maximize their utility according to their true preferences. The n economic participants (followers) will play strategically in the parameterized POMG $\mathcal{M}^\phi$. At each step t of the game, every follower i chooses an action $a_{i,t}$ from their action space A_i, the game state evolves according to the joint action $\boldsymbol{a}_t = (a_{1,t}, \cdots, a_{n,t})$ and the transition function T, and agents receive observations and reward according to O and r. An agent's behavior in the game is characterized by its policy $\pi_i : \Omega_i^\phi \to A_i^\phi$, which maps observations to actions. Each follower in the POMG $\mathcal{M}^\phi$ individually seeks to maximize its own (discounted) total return $\sum_t (\gamma^\phi)^t r_i^\phi(s_t, a_{i,t}, a_{-i,t})$.

D Generalization of Dynamic Mechanism Design

Notably, we generalize dynamic mechanism design in the following two ways:

1) By introduction of the outer voting model, we generalize the objective of the mechanism designer. The objective is no longer a fixed function, but rather a choice from a discrete set of options. Therefore, a static mechanism is not enough to solve our problem - instead, one must solve for a mechanism designer.

2) In the original mechanism design problem, the mechanism has control over only an allocation and transfer function, which in reinforcement learning terms can be thought of as the next state transition function and the reward function, respectively. However, in true policymaking this is oftentimes unrealistic, as the Principal additionally has control over defining the state and action spaces as well. Therefore, our framework generalizes the dynamic mechanism parameterization of the induced game as well, by allowing changes in the observation space.

We show here that Dynamic Mechanism Design (DMD) Problem by Athey & Segal (2013) is strictly generalized by the Universal Mechanism Design Game. Therefore, all results that hold for the Universal Mechanism Design Game will also hold for DMD.

Proposition 1. *The set of all Dynamic Mechanism Design problem instances is a strict subset of the set of all Universal Mechanism Design Game problem instances.*

Proof.

Every Dynamic Mechanism Design game is a Universal Mechanism Design Game

First, we show that **every instance of Dynamic Mechanism Design problem can be represented as a Universal Mechanism Design Game.**

To prove this, we first show every (One-Shot) Mechanism Design with Stochastic Agent problem can be represented as a Stackelberg Markov Nash Game, which is a single timestep of the Universal Mechanism Design Game. Please refer to Appendix B for game definitions.

Let $\mathcal{P} = (n, m, \mathcal{T}, \mathcal{O}, \boldsymbol{u}, p, u_0)$ be a (One-Shot) Mechanism Design with Stochastic Agent problem. Say $M^\phi = (g^\phi)$ is a parametric direct-revelation mechanism, we claim that M^ϕ induce a POMG $\mathcal{M}^\phi = (S, A, T^\phi, r^\phi, \Omega, O^\phi, \gamma, \mu_0)$ whose transition function, reward function, and observation function is parameterized by ϕ. Note that at each time step t, the state is characterized by the joint type profile: $\boldsymbol{s}^{(t)} = \boldsymbol{t}^{(t)}$, and each agent $i \in [n]$ can only observe their own type $\omega_i^{(t)} = \boldsymbol{t}_i^{(t)}$. After selecting the action $a_i^{(t)} = s_i(\boldsymbol{t}_i^{(t)})$, each agent receives reward defined by $r_i^\phi(\boldsymbol{s}^{(t)}, a_i^{(t)}, a_{-i}^{(t)}) = u_i(g^\phi(a_i^{(t)}, a_{-i}^{(t)}), \boldsymbol{t}_i^{(t)})$. Then, all agent will transit to next state $\boldsymbol{s}^{(t+1)}$ according to the transition function defined by $T^\phi(\boldsymbol{s}^{(t+1)}, \boldsymbol{s}^{(t)}, ((a_i^{(t)}, a_{-i}^{(t)}))) = p(\boldsymbol{t}^{(t+1)}, \boldsymbol{s}_t, g^\phi(a_i^{(t)}, a_{-i}^{(t)}))$. Moreover, each agent will only observe their own type, so we can define the observation function as $O^\phi(\boldsymbol{t}_i^{(t+1)}, \boldsymbol{s}^{(t)}, (a_i^{(t)}, a_{-i}^{(t)})) = \sum_{\boldsymbol{t} \in \mathcal{T}: \boldsymbol{t}_i = \boldsymbol{t}_i^{(t+1)}} p(\boldsymbol{t}, \boldsymbol{s}^{(t)}, g^\phi(a_i^{(t)}, a_{-i}^{(t)}))$.

Note that in the induced POMG, each agent aims to maximize their expected cumulative utility defined by $u_i(\phi, \pi) = \mathbb{E}^{\mathcal{M}^\phi, \pi}[\sum_{t=0}^{\infty} r^\phi(\boldsymbol{s}^{(t)}, \boldsymbol{a}^{(t)})]$ (Here the payoff is not discounted, but it would be the same with discounted payoff.)

Therefore, we can represent $\mathcal{P}$ as a $(n+1)$-player Stackelberg-Markov-Nash Game where, the mechanism designer is a leader who first commits an action ϕ which induces a n-player low-level (Partially Observable) Markov Game $\mathcal{M}^\phi = (S, A, T^\phi, r^\phi, \Omega, O^\phi, \gamma^\phi, s_0^\phi)$. Then, the agents are the followers who best-respond by playing a Nash equilibrium in the induced POMG.

Next, we show that every instance of the Dynamic Mechanism Design problem can be reduced to a (One-Shot) Mechanism Design with Stochastic Agent problem following the Markovization argument by Athey & Segal (2013).

Say $\mathcal{P} = (n, m, \mathcal{T}, \mathcal{O}, \boldsymbol{u}, p, u_0, \mathcal{H})$ is an instance of Dynamic Mechanism Design problem. Our goal is to “Markovize” the DMD problem.

First, we define the extended type space $\mathcal{T}' = \mathcal{H} \times \mathcal{T}$ and extended action space $\mathcal{A}' = \mathcal{H} \times \mathcal{A}$. Thus, we can define the new joint type transition function $p' : \mathcal{T}' \times \mathcal{T}' \times \mathcal{O} \to [0, 1]$ by $p'((\boldsymbol{h}^{(t+1)}, \boldsymbol{t}^{(t+1)}), (\boldsymbol{h}^{(t)}, \boldsymbol{t}^{(t)}), \boldsymbol{o}^{(t)}) = \mathbb{1}(\boldsymbol{h}^{(t+1)} = \boldsymbol{h}^{(t)} \cup (\boldsymbol{a}^{(t)}, \boldsymbol{o}^{(t)})) \cdot p(\boldsymbol{t}^{(t+1)}, \boldsymbol{t}^{(t)}, \boldsymbol{o}^{(t)})$. Moreover, each agent chooses action $a_i' = (\boldsymbol{h}, a_i) = s_i(\boldsymbol{t}_i') = s_i((\boldsymbol{h}, \boldsymbol{t}_i))$ based on strategy $s_i : \mathcal{T}_i' \to \mathcal{A}_i'$. The new utility function $u_i' : \mathcal{O} \times \mathcal{T}_i' \to \mathbb{R}$ is then defined by $u_i'(\boldsymbol{o}, \boldsymbol{t}_i') = u_i'(\boldsymbol{o}, (\boldsymbol{h}, \boldsymbol{t}_i)) = \boldsymbol{u}(\boldsymbol{o}, \boldsymbol{t}_i)$.

For the mechanism, the outcome rule is now $g' : \mathcal{A}' \to \mathcal{O}$ defined by $g'(\boldsymbol{a}') = g(\boldsymbol{h}, \boldsymbol{a})$, while the principal objective function is the same.

Therefore, we can conclude that the Dynamic Mechanism Design problem $\mathcal{P}$ is equivalent to the (One-Shot) Mechanism Design with Stochastic Agent problem $\mathcal{P}' = (n, m, \mathcal{T}', \mathcal{O}, \boldsymbol{u}', p', u_0)$.

NOT EVERY UNIVERSAL MECHANISM DESIGN GAME IS A DYNAMIC MECHANISM DESIGN GAME

Then, we show that **there exists an instance of Universal Mechanism Design Game that cannot be captured by Dynamic Mechanism Design problem**.

We can consider a slight extension of the Dynamic Mechanism Design problem which can still be captured in our model but falls outside the scope of Dynamic Mechanism Design.

Let $\mathcal{P} = (n, m, \mathcal{T}, \mathcal{O}, \boldsymbol{u}, p, u_0, \mathcal{H})$ be an instance of a Dynamic Mechanism Design problem, and let $M = (\{\mathcal{A}_i\}_{i \in [n]}, g)$ be a mechanism. We can extend the action space for each agent, such that at each time step t, besides the action $a_i^{(t)} \in \mathcal{A}_i$ that is sent to the mechanism, each agent also commits some effort $e_i{}^{(t)} \in \mathcal{E}_i$ which will not be sent to the mechanism and may directly influence the utility of other agents. Thus, each agent's utility function $u_i : \boldsymbol{o} \times \mathcal{E} \times \mathcal{T}_i$ under mechanism M when $\boldsymbol{a}, \boldsymbol{e}$ are chosen is given by $u_i(g(i), \boldsymbol{e}, \boldsymbol{t}_i)$. This is not captured by Dynamic Mechanism Design problem, but we can simply represent it as a $(n+1)$-player Stackelberg-Markov-Nash Game following a similar argument above. The key difference in modeling here is that DMD does not allow direct manipulation of other agents through an additional action, but a POMG is flexible enough to incorporate such a change. Thus, since Universal Mechanism Design Game parameterizes a general POMG our model is able to capture an additional direct change in the action space.

□

This results demonstrates the generality of our approach, and it suggests that by effectively solving the Universal Mechanism Design Game there is potential for unifying theories and methodologies that were previously considered distinct, but all fall under the the Universal Mechanism Design model.

E Detailed Discussion of Open Problems

Preference Aggregation and Democratic Representation.

Aggregation algorithms within the Voting Mechanism: The development of sophisticated algorithms that can effectively aggregate disparate and potentially conflicting preferences of diverse agent populations is a significant challenge. These algorithms must ensure that the outcomes represent collective preferences without overwhelming the minority views.

Mechanism resilience to manipulative behavior: Designing voting mechanisms that can withstand strategic reporting and manipulation attempts by agents is imperative. It is necessary to research methods to detect and penalize dishonest behavior while encouraging truthful revelation of preferences.

Incorporating diverse decision-making models: The framework must be flexible enough to respect various cultural, ethical, and socio-economic decision-making paradigms that different groups of agents might exhibit. In addition, such agents should imitate humans well. A potential path forward here is through leveraging the newfound capabilities of Large Language Models (LLMs) for imitating humans at surprisingly high degrees of realism.

AI Governance and Accountability.

Transparent decision-making processes: AI systems involved in policy-making need to have their decision-making processes fully transparent. The creation of interpretable AI models that can provide explanations for suggested policies is essential for trust and accountability.

Legal and ethical frameworks for AI decisions: There is an urgent need to establish legal and ethical frameworks that delineate the responsibilities and liabilities associated with AI-driven decision-making. These frameworks should set guidelines for what constitutes fair and lawful AI behavior in an economic context.

Oversight and human-AI collaboration: Establishing effective oversight mechanisms that involve both AI and human collaboration is critical. The role of human experts in supervising and guiding AI decisions and their ability to intervene when AI-driven policies deviate from desired outcomes is still to be determined.

Convergence to Desired Equilibria.

Existence and characterization of equilibria: Fundamental work is required to characterize the conditions under which a stable equilibrium might exist in such complex socio-economic interactions. The uniqueness or multiplicity of equilibria and the conditions leading to each scenario need in-depth exploration.

Algorithmic stability and multi-agent coordination: Researching advanced multi-agent reinforcement learning algorithms that can demonstrably converge to equilibrium points. The simultaneous coordination of learning processes among all agents, with varying objectives and possibly divergent strategies, remains an open problem.

Influence of dynamic changes on equilibria: The complex dynamics of economic systems call for a deep understanding of the sensitivity of equilibria to shocks and changes in the environment and agent behavior.

Below we include an extended list of further open problems that may be worth consideration and future research.

1. **Preference Aggregation and Democratic Representation:**
    - How should the framework account for the diversity of agents' preferences?
    - What are effective voting mechanisms that ensure fair representation of all agents in the Outer Voting Game?
    - Can we design mechanisms that are robust to strategic manipulation by agents during the preference reporting stage?
2. **Algorithmic Fairness and Ethical Considerations:**

- How can we ensure that policies designed by AI do not introduce or perpetuate biases against certain groups?
- What ethical guidelines should be integrated into the AI decision-making framework to avoid harmful outcomes?

3. **Convergence to Desired Equilibria:**
   - Under what conditions do solutions to the Universal Mechanism Design Game converge to socially desirable equilibria?
   - What learning algorithms can ensure convergence in the presence of multiple agents with different objectives?
4. **Scalability and Complexity:**
   - How can the Universal Mechanism Design Game be scaled to accommodate a large number of agents without compromising computational tractability?
   - What approaches can be used to simplify the model without losing the representation of real-world complexities?
5. **Incentive Compatibility and Mechanism Robustness:**
   - What are effective incentive structures that align individual agent goals with collective outcomes?
   - How can we design mechanisms that are robust to changing economic conditions and agent behaviors?
6. **Policy Implementation and Adaptability:**
   - What are effective strategies for implementing economic policy changes in dynamic environments?
   - How can the framework adapt to unexpected changes in the economy or agent behavior?
7. **Bounded Rationality and Behavioral Modelling:**
   - How should bounded rationality and imperfect information be incorporated into agent models to make them more human-like?
   - What are effective ways to model and simulate agents' behavioral patterns?
8. **Integration of Heterogeneous Agent Types:**
   - How can the framework effectively integrate both AI and human agents with potentially different types of rationality?
   - What techniques can manage interfaces between the AI policy-making system and human stakeholders?
9. **AI Governance and Accountability:**
   - Who is held accountable for decisions made by the AI system?
   - How can transparency and interpretability be achieved in AI-driven policy-making processes?
10. **Long-Term Policy Evolution and Learning:**
    - How can we design AI systems that take into account long-term policy evolution and socio-economic changes?
    - What learning mechanisms can enable AI systems to iteratively improve policy-making in conjunction with human oversight?