HCMD-*zero*: LEARNING VALUE ALIGNED MECHANISMS FROM DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Artificial learning agents are mediating a larger and larger number of interactions among humans, firms, and organizations, and the intersection between mechanism design and machine learning has been heavily investigated in recent years. However, mechanism design methods make strong assumptions on how participants behave (e.g. rationality), or on the kind of knowledge designers have access to a priori (e.g. access to strong baseline mechanisms). Here we introduce HCMD-*zero*, a general purpose method to construct mechanism agents. HCMD-*zero* learns by mediating interactions among participants, while remaining engaged in an electoral contest with copies of itself, thereby accessing direct feedback from participants. Our results on the Public Investment Game, a stylized resource allocation game that highlights the tension between productivity, equality and the temptation to free-ride, show that HCMD-*zero* produces competitive mechanism agents that are consistently preferred by human participants over baseline alternatives, and does so automatically, without requiring human knowledge, and by using human data sparingly and effectively Our detailed analysis shows HCMD-*zero* elicits consistent improvements over the course of training, and that it results in a mechanism with an interpretable and intuitive policy.

1 INTRODUCTIO

Artificial learning agents are beginning to play a central role in our institutions. From social networks, to investment management, and traffic routing, an ever growing number of interactions among humans, firms and organizations are mediated by adaptive systems.

While the intersection between mechanism design and machine learning has been heavily investigated in recent years, most methods make strong assumptions on either the behavior and preferences of participants (e.g. rationality), or on the kinds of knowledge, baseline mechanisms, or data we have access to before constructing a new mechanism agent for a given economic interaction.

Here we address these restrictive assumptions and present a general method to design a mechanism agent that is able to mediate complex economics interactions among human participants, while requiring no access to alternative mechanisms, human knowledge of the underlying interaction dynamics, and making very few assumptions on the nature of participants' preferences and strategies.

Our method builds on the pipeline outlined in Koster et al. (2022), but extends it to scenarios with *zero* human knowledge. It works by letting an adaptive agent repeatedly mediate interactions among participants, while remaining engaged in an electoral competition against a copy of itself. Letting the agent compete for participants' votes, rather than chasing proxy measures such as welfare or equality, ensures that the mechanism remains aligned to the preferences of its constituents, even when these preferences are hard to specify.

More specifically, we set out to construct a mechanism agent that is able to mediate economic interactions among human participants, and that is preferred by humans over alternative mechanisms.

We test our method on a stylized investment game where human participants could earn real money and which is known to stress the tensions between welfare, equality and the temptation to free ride. Our results show that the method presented here was able to construct a complex mechanism policy based on a simple expression of preference; and that this policy is favored by novel participants over a

baseline that was previously established as strong in this task. Moreover, our detailed analysis shows how the election-against-self curriculum pushes our agent towards interpretable mediation schemes with more and more pronounced punish / reward regions.

The impact of AI on our institutions is growing rapidly; and as such the intersection of mechanism design and machine learning is receiving considerable attention. Here we show that merging the most basic democratic principle of "one person, one vote" with modern machine learning and game theory insights leads to a general method for designing mechanisms that are aligned with the preferences of their constituents, while requiring *zero* human knowledge.

2 Related Work

Value alignment and AI safety have been intensely investigated in recent years both from a normative perspective (Gabriel, 2020), and from a technical one (Dafoe et al., 2020); and there is a growing support for building participatory systems for AI ethics and governance (Rahwan, 2017; Lee et al., 2019).

Mechanism design is a sub-field of economics that studies how to design the rules and incentives of multi-agent interactions, so that self-interested participants will prefer certain strategies, often trading off their own welfare for that of the group. The field has a long history to which it is near impossible to do justice, see (Maskin, 2008) for a review. More recently mechanism design has been studied from an algorithmic point of view (Conitzer & Sandholm, 2002; Nisan & Ronen, 2001), as well as a machine learning one (Dütting et al., 2017; Manisha et al., 2018; Tacchetti et al., 2019; Koster et al., 2022). Finally, researchers have recently turned their attention to the role that mechanism design can play in our pursuit of social good (Abebe & Goldner, 2018).

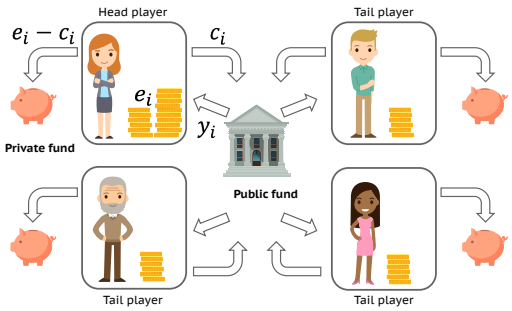

Figure 1: Public Investment Game (PIG): 4 players play over 10 identical rounds. Each round players receive an endowment (which remains the same for each player across all rounds). On each round, players can decide how much to keep private (directly contributing to their monetary payout after the game) or contribute to the public fund. The public fund sums all contributions and multiplies them by 1.6. The public fund is redistributed to the players on each round according to the policy of the mechanism. The returns from the fund then are added to each players monetary payout. Thus, the policy of the mechanism can influence whether the game is a social dilemma and how much initial inequality is redressed by the payouts of the fund.

Agent based models (ABMs), where a computer simulation predicts how autonomous agents will adapt to certain environment interventions, has been a tool used by policy makers to design new mechanisms since its inception. ABMs have received renewed attention after the 2008 Economic Crisis (Farmer & Foley, 2009; Hamill & Gilbert, 2015).

The problem of building artificial agents that coordinate with human participants starting from "zero knowledge" had been investigated both in the computer game setting (Strouse et al., 2021), and in a simulated economy environment (Zheng et al., 2020). Similarly, self-play and no-human-knowledge methods have been successfully applied to challenging constant-sum two-player games in recent past (Silver et al., 2016; 2017; Vinyals et al., 2019).

The method we present here builds heavily on the pipeline outlined in Koster et al. (2022): Human Centered Mechanism Design (HCMD), which uses a similar approach to train a mechanism agent on the same stylized economic game we consider, and with the same goal of producing mechanisms that are preferred by human participants over baseline alternatives. Here we provide a detailed comparison and highlight the substantial differences between HCMD and our contributions: First, HCMD does not rely on self-play, and rather trains mechanism agents using knowledge of the baseline mechanisms it needs to be preferred over, which need to be known a priori. Second, HCMD is not an iterative method, and it assumes that participants behavior and strategies do not depend on the mechanism they interact with, and can thus be modeled once and for all. The method presented here, on the other hand, works iteratively and thus takes into account how human participants react to changes in the

mechanism agent. Finally, HCMD uses ad-hoc knowledge about the underlying interaction dynamics to construct predictive models of participants' voting behavior; our method does not. In summary, HCMD is not a *zero*-knowledge method: it requires access to alternative mechanism policies for the specific interaction it sets out to mediate, it relies on knowledge of the voting dynamics to construct voting models, and finally it assumes that humans strategies are unaffected by changes in the mechanism policy, which is at best something that needs to be verified empirically for each interaction.

3 METHODS

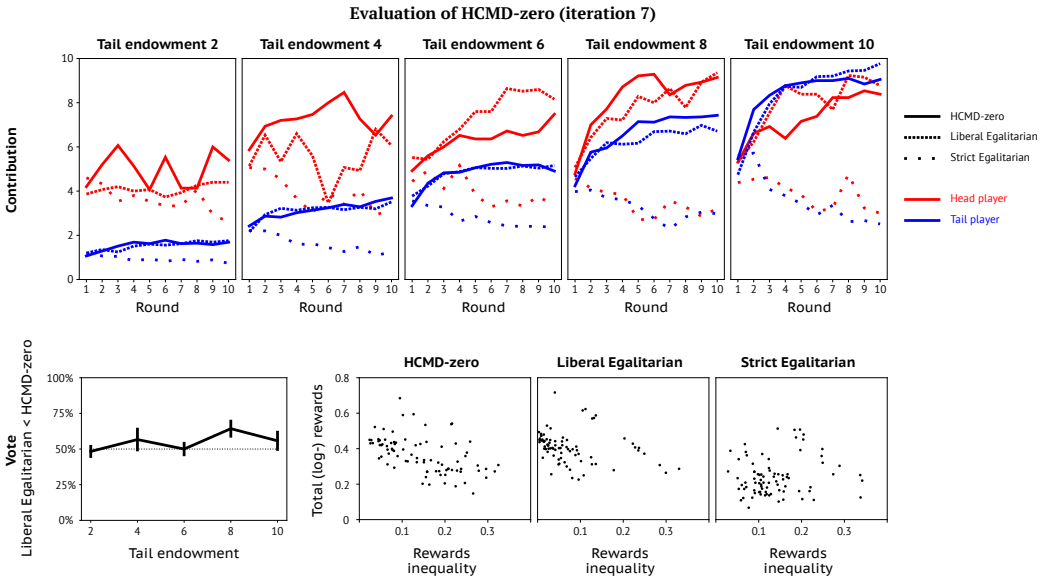

Figure 2: Evaluation of HCMD-*zero* after convergence against Liberal Egalitarian. Top panel: contribution of head and tail players across rounds, as a function of endowment, for each mechanism. Bottom left panel: votes in favour of HCMD-*zero* against Liberal Egalitarian, as a function of tail endowment. Bottom-right panel: Scatter plot of total reward (sum of log-rewards) against reward inequality (gini coefficient), for each mechanism. Each dot corresponds to one group, aggregated across all endowments.

We begin this section describing the Public Investment Game (PIG) we used for our experiments, and then proceed to introduce our methods in detail. We highlight here that while we report results on the PIG, and we use it to ground our exposition, our methods are applicable in very general settings.

3.1 PUBLIC INVESTMENT GAME FOR PARTICIPANTS AND MECHANISMS

From the point of view of the participants, the PIG is a general-sum 4 player game that unfolds over 10 identical rounds. At the beginning of each round, each participant receives an endowment of $e_{i,t}$ "coins", with $i = 1, \ldots, 4$, and $t = 1, \ldots, 10$, and decides what fraction of coins $\rho_{i,t}$ they would like to invest in a public fund that grows with a fixed multiplier of 1.6. A mechanism agent then observes $e_{i,t}$ and $\rho_{i,t}$, and determines 4 redistribution weights $w_{i,t} \geq 0$, with $\sum_i w_{i,t} = 1$, according to which the fund is returned in its entirety: each participant receives $w_{i,t}(1.6 \times \sum_i \rho_{i,t} e_{i,t})$, and the game moves on to the next round. In our experiments with human participants, at the end of the 10 rounds, each participant collected a monetary reward proportional to the funds they received from the public investment fund, and the endowments they decided not to contribute $R_i = \sum_t r_{i,t} = \sum_t w_{i,t}(1.6 \times \sum_i \rho_{i,t} e_{i,t}) + (1 - \rho_{i,t}) e_{i,t}$. From the point of view of the participants, the redistribution decisions of the mechanism agent are folded in the game transition kernel.

In our human experiments, we let the same cohort of players face two mechanism agents in independent instances of the PIG played in sequence. We then asked each participant to cast a vote on

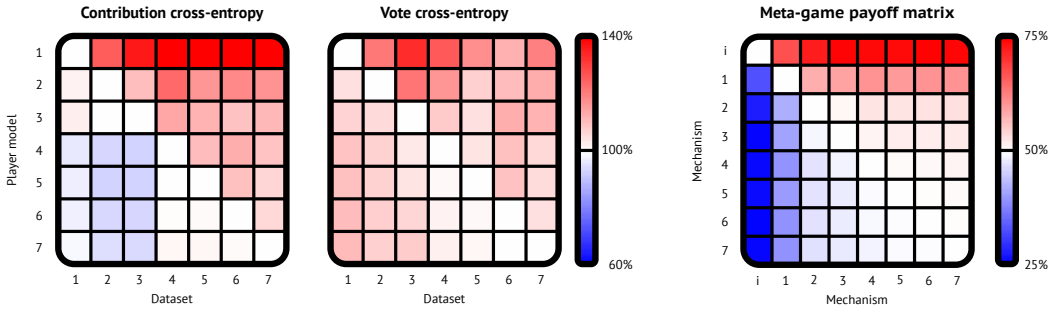

Figure 3: Monitoring performance of player model and mechanism across training iterations. Left panel: contribution and vote cross-entropy (ratio normalized by the diagonal) of player models p_s across datasets D_s. Player models made better predictions on datasets acquired in earlier iterations. Right panel: payoff matrix from the meta-game, where pairs of mechanisms compete for votes in simulation. Later iterations obtain monotonically increasing votes against earlier versions, with convergence after iteration 7. The initial iteration (random mechanism) is denoted with i.

which mechanism they would like to re-experience in a third follow up game in which the mechanism would be decided by majority vote. It is worth noting that participants collected monetary rewards in all three games (two initial, and one follow up), and thus had "skin in the game" when reporting which mechanism they preferred.

From the point of view of the mechanism agent, the PIG is a 2 player constant-sum game. First, the two candidate mechanisms face independent sequential decision making problems with states coinciding with endowments, contributions, and redistribution histories, and actions coinciding with redistribution weights. Second, the two mechanisms collect a reward based on the number of votes cast in their favor. Similarly to what happens with participants, the transition kernel implements the PIG game dynamics, as well as the participants contributing and voting behaviors.

3.2 Self-play loop overview

Iteration	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$	$s=6$	$s=7$
Groups	73	45	51	101	53	49	42
Contrib. Linear size	8	8	8	32	32	32	32
Contrib. LSTM size	4	4	4	8	8	8	8

Table 1: Amount of data collected and modeling hyper-parameters for each iteration.

Here we introduce HCMD-*zero*, a method to construct a mechanism agent that is able to mediate economic interactions among human participants, and that is preferred by humans over alternative mechanisms.

We build on the recent successes of self-play to train competitive agents in challenging 2-player constant sum games (Silver et al., 2016; 2017; Vinyals et al., 2019), and let our mechanism agent update its redistribution policy so as to win against a copy of itself in the PIG election game we outlined above (i.e. the two mechanism agents faced by participants in the two initial games were in fact copies of one another, this fact was not disclosed to participants).

Notably, we did not train our mechanism agent directly on experience acquired while interacting with human participants (this would require access to a prohibitive amount of data). Instead we constructed "virtual participants" by training neural network models to closely imitate human contributing and voting behavior in the PIG, and let our mechanism agents interact with these instead.

The method we outline is thus composed of 3 *phases*, which are repeated in order over several *iterations*: 1) *Acquire*: collect behavioral data (contributions and voting) by letting participants interact with a mechanism agent in two sequential instances of the PIG, 2) *Model*: construct accurate models of participants' voting and contributing behaviors, 3) *Optimize*: construct a simulated election

game environment using "virtual participants" and optimize the mechanism agent's redistribution policy in pursuit of participants preferences. As stated above, the 3 phases *Acquire*, *Model*, and *Optimize* are repeated in order within each *iteration*. In particular, in each *Acquire* phase, we collect behavioral trajectories of human participants interacting with the mechanism agent produced by the most recent *Optimize* phase (we use a random mechanism on the first iteration). The *Model* phase trains participants' models using data from all previous iterations, and finally the *Optimize* phase relies on the most recent virtual participants models to construct the mechanism agent's training environment.

3.3 *Acquire* STEP: DATA ACQUISITION

We used a crowd-sourcing platform to acquire contributing and voting behavior data from human participants ($n = 1656$). All participants gave informed consent to participate in the experiment. In particular, during the *Acquire* step of each iteration s, groups of 4 human participants completed two episodes of the PIG game interacting with a mechanism agent endowed with the most recent parameters (see Sec. 3.5 for details on the mechanism agent), and voted for the episode they preferred. We denote this data-set as D_s. Since the mechanism and the conditions (e.g. the endowment) are identical in both episodes, the only difference between them is driven by the randomness in human behavior.

3.4 *Model* STEP: MODEL PARTICIPANTS

During the *Model* step for iteration s, we trained independent models to predict human contributions and votes (we jointly refer to these models as "virtual participants") from all data-sets $D_1, \ldots, D_s$ collected thus far.

The contribution model is a neural network similar to that in (Koster et al., 2022), which takes as input each players normalized endowments and contributions: $e_{i,t}/10$ and $c_{i,t}/10$, as well as each players fractional contribution $\rho_{i,t} = c_{i,t}/e_{i,t}$, and outputs the log-likelihood of contributing $0, 1, \ldots, 10$ coins (10 coins being the maximum endowment). The network was applied independently for each participant and composed of an input linear layer, a LSTM and an output linear layer. The contribution model was trained to minimize group-wise cross-entropy between predicted and actual contributions.

The votes model is a simple linear layer, which we apply independently for each participant, that takes in the flattened observations from a single episode (10 rounds $\times$ 3 endowment/contribution/payout $\times$ 4 participants) and produces a single output, which can be interpreted as the log-likelihood of voting for the current episode. The same linear layer is applied to both episodes, and a softmax normalization produces the final probabilities. We train this network to minimize group-wise cross-entropy between predicted and actual votes, with an additional l^2 regularization loss of the linear layer parameters.

Since the amount of data available increases with every iteration, hyper-parameters must be adjusted each time. In our experiments, we tuned the l^2 regularization and network size using cross validation with a random 70%-30% train/eval split. We reconstituted the original data-set for training (see Tab. 1 for details).

3.5 *Optimize* STEP: TRAIN MECHANISM AGENT

Similarly to HCMD, we parameterize the mechanism policy with a Graph Network (Battaglia et al., 2018) policy function π, with no memory and parameters θ, which takes as input the current endowment and contribution from each player (as nodes of a fully connected graph) and outputs deterministic redistribution weights of each player $w_{i,t}$.

During the *Optimize* step, we update the redistribution policy parameters $\theta_{s-1} \rightarrow \theta_s$ by letting our mechanism agent interact with the most current "virtual participant" models p_s.

Specifically, we trained the mechanism by approximating the mechanism agent's policy gradient through a bespoke low variance estimator based on Stochastic Computation Graphs (Schulman et al., 2016) that exploits the differentiable structure of the PIG while accounting for the stochastic nature of the player model's contributions (similar to (Koster et al., 2022)). We note that while this choice is suitable for our set up, the learning rule can be replaced by any Reinforcement Learning technique that fits the problem at hand. During training, we used batches of 1000 games equally split

among the endowment condition we considered: $[10, 2, 2, 2]$, $[10, 4, 4, 4]$, $[10, 6, 6, 6]$, $[10, 8, 8, 8]$ and $[10, 10, 10, 10]$. The mechanism's policy was trained using an ADAM optimizer with learning rate $4e - 5$. Finally, we fixed the number of gradient updates to 2000 for intermediate iterations and 10000 for the final one. This choice warrants a brief discussion: there is a trade off between how aggressively we require our participants model p_s to extrapolate beyond its training distribution (recall that training data was collected using mechanism parameters $\theta_0 \dots \theta_{s-1}$), and how many total iterations (and thus data collection steps) we prescribe.

Related to the choice of training updates within an iteration's *Optimize* step, our method requires determining how many *iterations*, i.e. repetitions of our *Acquire*, *Model* and *Optimize* pipeline, we should complete. Our proposed approach is to construct a meta-game, that is, recording the results of a round-robin election tournament among the mechanism agents produced at each iteration, constructed using the most recent "virtual participants" models (see Fig. 3). More precisely we constructed a meta-game as a two-player game defined by a payoff matrix of size $s \times s$ with entry i, j corresponding to the proportion of votes collected by mechanisms playing with parameters θ_i and θ_j over 100 games. Once the actions corresponding to later checkpoints no longer constitute a dominant strategy in the meta-game, or when their advantage becomes negligible, we can conclude that HCMD-*zero* has converged, since it no longer produces meaningful improvements[1].

4 RESULTS

In this section we show the results of applying our method in the Public Investment Game. We first show the performance against baselines after 7 iterations of training with HCMD-*zero*. Then, we explore in more detail the learning dynamics of the model of human participants, as well as the convergence of the mechanism. Finally, we provide an analysis of the mechanism's behavior throughout training.

4.1 PERFORMANCE OF HCMD-*zero* AT EVALUATION

In order to validate our approach, we trained a mechanism in the Public Investment Game (PIG). Similar to Koster et al. (2022), we divided participants into one "Head" player that always received an endowment $e_{head,t} = 10$ and three "tail" players that received a "tail" endowment $e_{tail,t} \in \{2, 4, 6, 8, 10\}$. The tail endowment was consistent within a group, across tail payers and for all mechanisms they interacted with. We evaluated the mechanism θ_7 by collecting new data specifically for this purpose. Humans interacted with the trained mechanism and with a baseline alternative in two subsequent games (in counterbalanced order). Our choice of baseline was Liberal Egalitarian mechanism, a redistribution scheme that disburses the public fund according to the proportion of endowment contributed by each participant, as baseline. Koster et al. (2022) show that Liberal Egalitarian is a strong baseline that is preferred by humans over the Strict Egalitarian, which divides the fund in equal parts.

Results are shown in Fig. 2. HCMD-*zero* was voted more often than Liberal Egalitarian, achieving an average of 54.3% of the votes ($p < 0.06$ with a non-parametric analysis that corrects for in-group correlations). More specifically, HCMD-*zero* achieved at least half of the votes against Liberal Egalitarian (see bottom-left panel), whilst matching the contributions from players (top panel). At the group level, HCMD-*zero* matched the performance of Liberal Egalitarian in trading off the productivity of the group (incentivizing the head player to contribute more) and the inequality of the group (redistributing to the tail players; bottom-right panel).

4.2 PARTICIPANT MODELS DISPLAY EQUILIBRIUM EFFECTS

We turned to look at the predictive power of the "virtual participants" model p_s across iterations. Our iterative method addresses the fact that human contribution and voting behavior depends on the mechanism. Fig. 3 (left two panels) shows that this effect is observed in practice. We construct a

[1]This assumes that there are no Condorcet cycles among the mechanisms, an assumption which is true in practice (see Fig. 3), and already "baked in" the choice of self-play (see (Balduzzi et al., 2019)). Should such cycles become apparent when constructing the meta-game, one could turn to standard methods to address them (e.g. Fictitious Play or Double Oracle).

contribution and vote cross-validation matrix by reporting in entry i, j the cross-entropy loss achieved by each model p_i (rows) on each data-set D_j (columns); recall that model p_i is trained using data-sets $D_1, \ldots, D_i$ (matrix entries are normalized per-column by the corresponding diagonal entry). The figure clearly shows that the predictive performance of each model degrades progressively for each subsequent data-set indicating that participants contributing and voting behavior has changed.

4.3 Mechanism improvement and convergence

On every iteration s we constructed a meta-game as described in the methods above, where each row and column corresponds to the mechanisms $\theta_0, \ldots, \theta_s$ and each cell corresponds to the number of votes obtained in simulation with the "virtual participants" model p_s. This can be found for iteration 7 on the right panel in Fig. 3. With HCMD-*zero*, the optimization of the mechanism showed consistent improvements on every iteration, with diminishing returns until convergence on iteration 7.

4.4 Analysis of mechanism behavior

We analyze the learnt mechanism policy across iterations in Fig. 4. For each tail endowment (columns) and across iterations (rows), we illustrate the mechanism's policy on a grid containing the contributions of the head (y-axis) and tail players (x-axis, averaged across the 3 tail players). Then, for each possible contribution pair, we computed the average redistribution weight across episodes and players. These are plotted with yellow favouring redistribution to the head player (high endowment) and blue to the tail players (low endowment). Our mechanism player effectively learns a policy similar to the Liberal Egalitarian mechanism (see additional row at the bottom): straight lines fanning out from left to right and bottom to top; but which punishes harshly Head players that do not contribute enough: pinching at the bottom left.

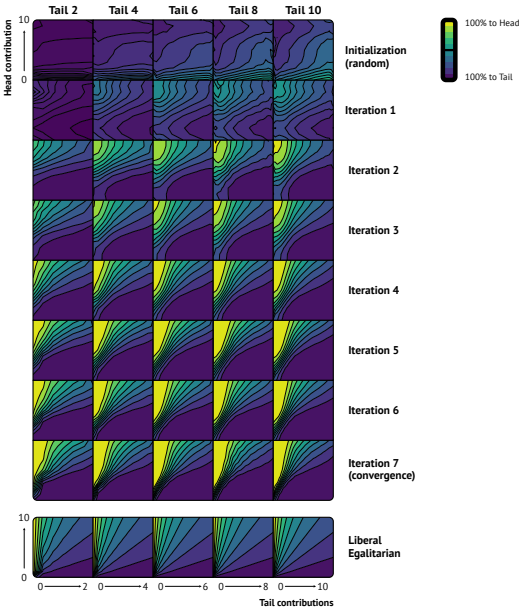

Figure 4: Illustration of the learnt mechanism policy across training iterations. Each heat-map illustrates a mechanism (rows) under a given tail endowment (columns). The rows correspond to mechanism θ_0 followed by iterations θ_s. The last row corresponds to the competing mechanism used in evaluation. Heat-maps illustrate the fraction of payout distributed to the head (yellow) or tail players (blue) as a function of the contributions provided by the head (y-axis, bottom to top) and tail players (x-axis, left to right).

5 Discussion

We have introduced HCMD-*zero*, a general purpose method to construct mechanisms that are preferred by human participants. Our methods require no baseline or alternative mechanisms, and no knowledge of the environment dynamics. Our methods use participant modeling and self-play to minimize the amount of data that is required to train a mechanism, and they iteratively address the challenges posed by equilibrium effects, where the participants behavior changes in response to updates in the mechanism policy. Our results show that HCMD-*zero* produces a competent mechanism agent in the challenging Public Investment Game. Our detailed analysis shows that our mechanism policy is consistently improved across iterations, and provides an interpretation of its final policy.

Artificial learning agents are becoming a centerpiece of our institutions, and as such methods to ensure that mechanisms are aligned to the values of their constituents are being heavily investigated. The ideas and results presented here indicate that integrating the most basic democratic principle of one person one vote, with modern machine learning techniques is a viable and fruitful path forward.

REFERENCES

Rediet Abebe and Kira Goldner. Mechanism design for social good. *AI Matters*, 4(3):27–34, 2018.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M. Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games, 2019.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 103–110. Morgan Kaufmann Publishers Inc., 2002.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, and David C Parkes. Optimal auctions through deep learning. *arXiv preprint arXiv:1706.03459*, 2017.

J Doyne Farmer and Duncan Foley. The economy needs agent-based modelling. *Nature*, 460(7256): 685–686, 2009.

Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3):411–437, 2020.

Lynne Hamill and Nigel Gilbert. *Agent-based modelling in economics*. John Wiley & Sons, 2015.

Raphael Koster, Jan Balaguer, Andrea Tacchetti, Ari Weinstein, Tina Zhu, Oliver Hauser, Duncan Williams, Lucy Campbell-Gillingham, Phoebe Thacker, Matthew Botvinick, and Christopher Summerfield. Human-centered mechanism design with Democratic AI. *arXiv preprint arXiv:2201.4137011*, 2022.

Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. Webuildai: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW): 1–35, 2019.

Padala Manisha, CV Jawahar, and Sujit Gujar. Learning optimal redistribution mechanisms through neural networks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 345–353. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Eric S Maskin. Mechanism design: How to implement social goals. *American Economic Review*, 98 (3):567–76, 2008.

Noam Nisan and Amir Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35 (1-2):166–196, 2001.

Iyad Rahwan. Society-in-the-loop: programming the algorithmic social contract. *Ethics and Information Technology*, 20(1):5–14, Aug 2017. ISSN 1572-8439. doi: 10.1007/s10676-017-9430-8. URL `http://dx.doi.org/10.1007/s10676-017-9430-8`.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs, 2016.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34, 2021.

Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. A neural architecture for designing truthful and efficient auctions. *arXiv preprint arXiv:1907.05181*, 2019.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

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, 2020.