Human adaptation to adaptive machines converges to game-theoretic equilibria

Benjamin J. Chasnov[1], **Lillian J. Ratliff**[1], **and Samuel A. Burden**[1,*]

[1]Department of Electrical & Computer Engineering, University of Washington, Seattle, WA 98195, USA
[*]Corresponding author: sburden@uw.edu

ABSTRACT

Here we test three learning algorithms for machines playing general-sum games with human subjects. The algorithms enable the machine to select the outcome of the co-adaptive interaction from a constellation of game-theoretic equilibria in action and policy spaces. Importantly, the machine learning algorithms work directly from observations of human actions without solving an inverse problem to estimate the human's utility function as in prior work. Surprisingly, one algorithm can steer the human-machine interaction to the machine's optimum, effectively controlling the human's actions even while the human responds optimally to their perceived cost landscape. Our results show that game theory can be used to predict and design outcomes of co-adaptive interactions between intelligent humans and machines.

Introduction

Adaptive machines have the potential to assist *or* interfere with human behavior in a range of contexts, from cognitive decision-making[1,2] to physical device assistance[3–5]. Therefore it is critical to understand how machine learning algorithms can influence human actions, particularly in situations where machine goals are misaligned with those of people[6]. Since humans continually adapt to their environment using a combination of explicit and implicit strategies[7,8], when the environment contains an adaptive machine, the human and machine play a *game*[9,10]. Game theory is an established framework for modeling interactions between two or more decision-makers that has been applied extensively in economic markets[11] and machine algorithms[12]. However, existing approaches make assumptions about, rather than empirically test, how adaptation by individual humans is affected by interaction with an adaptive machine[13,14]. Here we tested learning algorithms for machines playing general-sum games with human subjects. Our algorithms enable the machine to select the outcome of the co-adaptive interaction from a constellation of game-theoretic equilibria in action and policy spaces. Importantly, the machine learning algorithms work directly from observations of human actions without solving an inverse problem to estimate the human's utility function as in prior work[15,16]. These results show that game theory can be used to predict and design outcomes of co-adaptive interactions between intelligent humans and machines.

We studied games played between humans H and machines M. The games were defined by quadratic functions that mapped scalar actions of each human h and machine m to costs $c_H(h,m)$ and $c_M(h,m)$. Games were played continuously in time over a sequence of trials, and the machine adapted within or between trials. Human actions h were determined from a manual input device (mouse or touchscreen) as in Figure 1A, while machine actions m were determined algorithmically from the machine's cost function c_M and the human's action h as in Figure 1B. The human's cost $c_H(h,m)$ was continuously shown to the human subjects via the height of a rectangle on a computer display as in Figure 1A, which the subject was instructed to "make as small as possible", while the machine's actions were hidden.

Game-theoretic equilibria in continuous games

The experiments reported here were based on a game that is *continuous*, meaning that players choose their actions from a continuous set, and *general-sum*, meaning that the cost functions prescribed to the human and machine were neither aligned nor opposed. Unlike pure optimization problems, game players cannot control all variables that determine their cost. Each player seeks its own preferred outcome, but the game outcome will generally represent a compromise between players' conflicting goals. We considered *Nash*[17], *Stackelberg*[18], *consistent conjectural variations*[19], and *reverse Stackelberg*[20] equilibria of the game (Definitions 4.1, 4.6, 4.9, 7.1 in[10] respectively), in addition to each player's *global optimum*, as possible outcomes in the experiments. Formal definitions of these game-theoretic concepts are provided in the references, but we provide plain-language descriptions in the next paragraph. Table 1 contains expressions for the cost functions that defined the game considered here as well as numerical values of the resulting game-theoretic equilibria.

Nash equilibria[17] arise in games with simultaneous play, and constitute points in the joint action space from which neither player is incentivized to deviate (see Section 4.2 in[10]). In games with ordered play where one player (the *leader*) chooses its

action assuming the other (the *follower*) will play using its best response, a Stackelberg equilibrium[18] may arise instead. The leader in this case employs a *conjecture* about the follower's policy, i.e. a function from the leader's actions to the follower's actions, and this conjecture is consistent with how the follower plays the game (Section 4.5 in[10]); the leader's conjecture can be regarded as an *internal model*[13,21,22] for the follower. Shifting from Nash to Stackelberg equilibria in our quadratic setting is generally in favor of the leader whose cost decreases. Of course, the follower may then form a conjecture of its own about the leader's play, and the players may iteratively update their policies and conjectures in response to their opponent's play. In the games we consider, this iteration converges to a *consistent* conjectural variations equilibrium[19] defined in terms of actions *and* conjectures: each player's conjecture is equal to their opponent's policy, and each player's policy is optimal with respect to its conjecture about the opponent (Section 7.1 in[10]). Finally, if one player realizes how their choice of policy influences the other, they can design an *incentive* to steer the game to their preferred outcome, termed a *reverse* Stackelberg equilibrium[20] (Section 7.4.4 in[10]).

Results

We conducted three experiments with different populations of human subjects recruited from a crowd-sourcing research platform[23]. The participants engaged in tasks defined by a pair of cost functions c_H, c_M illustrated in Figure 1A,B that were quadratic in the scalar variables $h, m \in \mathbb{R}$. The experiments were designed to yield distinct game-theoretic equilibria in both action and policy spaces. These analytically-determined equilibria were compared with the empirical distributions of actions, policies, and costs reached by humans and machines over a sequence of trials in each experiment. In all three experiments, we found that the observed behavior converges to the predicted game-theoretic values depicted in Figure 4A. Illustrations of the algorithms employed in the three experiments are shown in Figure 4B,C,D and sourcecode implementing a numerical simulation of the algorithms are provided in the supplement.

Experiment 1: gradient descent in action space

In our first experiment (Figure 1), the machine adapted its action within trials using what is arguably the simplest optimization scheme: gradient descent[24,25] (see Methods and Figure 4B). We tested seven adaptation rates $\alpha \geq 0$ for the gradient descent algorithm as illustrated in Figure 1C,D,E for each human subject, with two repetitions for each rate and the sequence of rates occurring in random order. We found that distributions of median action vectors for the population of $n = 20$ human subjects in this experiment shifted from the *Nash equilibrium* (NE) at the slowest adaptation rate to the *human-led Stackelberg equilibrium* (SE) at the fastest adaptation rate (Figure 1C). Importantly, this result would not have obtained if the human was also adapting its action using gradient descent, as merely changing adaptation rates in simultaneous gradient play does not change stationary points[25]. The shift we observed from Nash to Stackelberg, which was in favor of the human (Figure 1E), was statistically significant in that the distribution of actions was distinct from the SE action but not the NE action at the slowest adaptation rate and vice-versa for the fastest rate (Figure 1D; $P \leq 0.05$ comparing to SE at the slowest adaptation rate and to NE at the fastest adaptation rate, and $P > 0.05$ comparing to NE at the slowest adaptation rate and to SE at the fastest adaptation rate). Discovering that the human's empirical play is consistent with the theoretically-predicted best-response function for its prescribed cost is important, as this insight motivated us in subsequent experiments to elevate the machine's play beyond the action space to reason over its space of *policies*, that is, functions from human actions to machine actions.

Experiment 2: conjectural variation in policy space

In our second experiment (Figure 2), the machine played affine policies (i.e. m was determined as an affine function of h) and adapted its policies by observing the human's response (see Methods, Figure 2F,G, and Figure 4C). Trials came in pairs, with the machine's policy in each pair differing only in the constant term. After each pair of trials, the machine used the median action vectors from the pair to estimate a *conjecture*[19,26] (or *internal model*[13,21,22]) about the human's policy, and the machine's policy was updated to be optimal with respect to this conjecture. Unsurprisingly, the human adapted its own policy in response. Iterating this process shifted the distribution of median action vectors for a population of $n = 20$ human subjects (distinct from the population in the first experiment) from the *human-led Stackelberg equilibrium* (SE) toward a *consistent conjectural variations equilibrium* (CCVE) in action and policy spaces (Figure 2A). The shift we observed away from the SE cost toward the CCVE cost from the first to last iteration, which was in favor of the human at the machine's expense, was statistically significant in that the distribution of actions was distinct from the SE but not the CCVE at the final iteration (Figure 2B; $P \leq 0.05$ comparing to SE and $P > 0.05$ comparing to CCVE at iteration $k = 9$). The machines' empirical conjectures overlapped with theoretical predictions of human policies at all conjectural variation iterations (Figure 2G), suggesting that both humans and machines estimated consistent conjectures of their opponent.

Experiment 3: gradient descent in policy space

In our third experiment (Figure 3), the machine adapted its affine policy using a *policy gradient* strategy[25] (see Methods, Figure 3F,G, and Figure 4D). Trials again came in pairs, with the machine's policy in each pair differing this time only in the linear term. After a pair of trials, the median costs of the trials were used to estimate the gradient of the machine's cost with respect to the linear term in its policy, and the linear term was adjusted in the direction opposing the gradient to decrease the cost. Iterating this process shifted the distribution of median action vectors for a population of $n = 20$ human subjects (distinct from the populations in the first two experiments) from the *human-led Stackelberg equilibrium* (SE) toward the machine's *global optimum* (Figure 3A), which can also be regarded as a *reverse Stackelberg equilibrium* (RSE)[20], this time optimizing the machine's cost at the human's expense (Figure 3D). The shift we observed away from the SE cost toward the RSE cost from the first to last iterations was statistically significant in that the distribution of actions was distinct from the SE but not the RSE at the final iteration (Figure 3B; $P \leq 0.05$ comparing to SE and $P > 0.05$ comparing to RSE at iteration $k = 9$) and the machines' empirical policy gradients overlapped with theoretically-predicted values (Figure 3G), suggesting that the machine can accurately estimate its policy gradient and minimize its cost. In essence, the machine elevated its play by reasoning in the space of policies to steer the game outcome in this experiment to the point it desires in the joint action space. We report results from variations of this experiment with different policy initializations and machine minima in the Supplement.

Discussion

Our experiments feature continuous human-machine interactions, introducing a dynamic perspective not present in prior work on interactions between intelligent human and machine agents[27]. By mapping a constellation of game-theoretic equilibria, we obtained predictions for the outcomes of these dynamic interactions. Our experimental results agree remarkably well with the predictions, providing empirical support for a game theory approach to analysis and synthesis of these systems.

When the machine played any policy in our experiments (i.e. when the machine's action m was determined as a function of the human's action h), it effectively imposed a constraint on the human's optimization problem. The policy could arise indirectly, as in the first experiment where the machine descended the gradient of its cost at a fast rate, or be employed directly, as in the second and third experiments. In all three experiments, the empirical distributions of human actions or policies were consistent with the analytical solution of the human's constrained optimization problem for each machine policy (Figure 1D; Figure 2B,C; Figure 3B,C). This finding is significant because it shows that optimality of human behavior was robust with respect to the cost we prescribed and the constraints the machine imposed, indicating our results may generalize to other settings where people (approximately) optimize their own utility function. We report results from variations of all three experiments with non-quadratic cost functions in the Supplement.

There is an exciting prospect for adaptive machines to assist humans in work and activities of daily living as tele- or co-robots[13], interfaces between computers and the brain or body[28,29], and devices like exoskeletons or prosthetics[3–5]. But designing adaptive algorithms that play well with humans – who are constantly learning from and adapting to their world – remains an open problem in robotics, neuroengineering, and machine learning[13,28,30]. We validated game-theoretic methods for machines to provide assistance by shaping outcomes during co-adaptive interactions with human partners. Importantly, our methods do not entail solving an inverse optimization problem[15,16]; rather than estimating the human's cost function, our machines learn directly from human actions. This feature may be valuable in the context of the emerging *body-/human-in-the-loop optimization* paradigm for assistive devices[3–5], where the machine's cost is deliberately chosen with deference to the human's metabolic energy consumption[31] or other preferences[32].

Our results demonstrate the power of machines in co-adaptive interactions played with human opponents. Although humans responded rationally at one level by choosing optimal actions in each experiment, the machine was able to "outsmart" its opponents over the course of the three experiments by playing higher-level games in the space of policies. This machine advantage could be mitigated if the human rises to the same level of reasoning, but the machine could then go higher still, theoretically leading to a well-known infinite regress[33]. We did not observe this regress in practice, possibly due to bounds on the computational resources available to our human subjects as well as our machines[34]. One important limitation of the present work was the duration of our experiments, which was short: approximately 10 to 14 minutes. If this duration was extended, it is likely subjects would get bored and the outcome may shift away from the equilibria reported here. Future work will need to consider the effect of attention in applications of interest[35].

As machine algorithms permeate more aspects of daily life, it is important to understand the influence they can exert on humans to prevent undesirable behavior, ensure accountability, and maximize benefit to individuals and society[6,36]. Although the capabilities of humans and machines alike are constrained by the resources available to them, there are well-known limits on human rationality[37] whereas machines benefit from sustained increases in computational resources, training data, and algorithmic innovation[38,39]. Here we showed that machines can unilaterally change their learning strategy to select from a wide range of theoretically-predicted outcomes in co-adaptation games played with human subjects. Thus machine learning algorithms may have the power to aid human partners, for instance by supporting decision-making or providing assistance

when someone's movement is impaired. But when machine goals are misaligned with those of individuals or society, it may be necessary to impose limitations on algorithms to ensure the safety, autonomy, and well-being of people.

Materials and Methods

Experimental Design

Human subjects were recruited using the online crowd-sourcing research platform *Prolific*[23]. Experiments were conducted using procedures approved by the University of Washington Institutional Review Board (UW IRB STUDY00013524). All research was performed in accordance with relevant guidelines and regulations. Informed consent was obtained from all participants. Participant data were collected on a secure web server. Each experiment consisted of a sequence of trials: 14 trials in the first experiment, 20 trials in the second and third experiments. During each trial, participants used a web browser to view a graphical interface and provide manual input from a mouse or touchscreen to continually determine the value of a scalar action $h \in \mathbb{R}$. This cursor input was scaled to the width of the participant's web browser window such that $h = -1$ corresponded to the left edge and $h = +1$ corresponded to the right edge. Data were collected at 60 samples per second for a duration of 40 seconds per trial in the first experiment and 20 seconds per trial in the second and third experiments. Human subjects were selected from the "standard sample" study distribution from all countries available on Prolific. Each subject participated in only one of the three experiments. No other screening criteria were applied.

At the beginning of each experiment, an introduction screen was presented to participants with the task description and user instructions. At the beginning of each trial, participants were instructed to move the cursor to a randomly-determined position. This procedure was used to introduce randomness in the experiment initialization and to assess participant attention. Throughout each trial, a rectangle's height displayed the current value of the human's cost $c_H(h,m)$ and participant was instructed to "keep this [rectangle] as *small* as possible" by choosing an action $h \in \mathbb{R}$ while the machine updated its action $m \in \mathbb{R}$. A square root function was applied to cost values to make it easier for participants to perceive small differences in low cost values. After a fixed duration, one trial ended and the next trial began. Participants were offered the opportunity to take a rest break for half a minute between every three trials. The experiment ended after a fixed number of trials. Each experiment lasted approximately 10–14 minutes and the participants received a fixed compensation of $2 USD (all data was collected in 2020). The user interface presented to human subjects was identical in all experiments. However, the machine adapted its action and policy throughout each experiment, and the adaptation algorithm differed in each experiment.

Cost functions

In Experiments 1, 2, and 3, participants were prescribed the quadratic cost function

$$c_H(h,m) = \tfrac{1}{2}h^2 + \tfrac{7}{30}m^2 - \tfrac{1}{3}hm + \tfrac{2}{15}h - \tfrac{22}{75}m; \tag{1}$$

the machine optimized the quadratic cost function

$$c_M(h,m) = \tfrac{1}{2}m^2 + h^2 - hm. \tag{2}$$

These costs were designed such that the players' optima and the constellation of relevant game-theoretic equilibria were in distinct positions as listed in Table 1 and illustrated in Figure 4A. In particular: the coefficients on h^2 in c_H and m^2 in c_M were set to $1/2$ and the constant term in both costs was set to zero without loss of generality. Seven of the remaining eight coefficients in the two quadratic cost functions were uniquely determined by arranging Nash and Stackelberg symmetrically about the machine's global optimum and choosing the human's global optimum to be distinct from these three points. The final free parameter was chosen to position the consistent conjectural variations equilibrium to be distinct from the other equilibria and to be stable under the conjectural iteration algorithm. During each trial of an experiment, the time series of actions from the trials were recorded as human actions $h_0, \ldots, h_t, \ldots, h_T$ and machine actions $m_0, \ldots, m_t, \ldots, m_T$, for a fixed number of samples T. At time t, the players experienced costs $c_H(h_t, m_t)$ and $c_M(h_t, m_t)$.

Experiment 1: gradient descent in action space

In the first experiment, the machine adapted its action using gradient descent,

$$m^+ = m - \alpha\, \partial_m c_M(h,m), \tag{3}$$

with one of seven different choices of adaptation rate $\alpha \in \{0, 0.003, 0.01, 0.03, 0.1, 0.3, 1\}$. At the slowest adaptation rate $\alpha = 0$, the machine implemented the constant policy $m = -0.2$, which is the machine's component of the game's Nash equilibrium. At the fastest adaptation rate $\alpha = 1$, the gradient descent iterations in (3) are such that the machine implements the linear policy $m = h$. Each condition was experienced twice by each human subject, once per symmetry (described in the next paragraph), in randomized order. Numerical simulation sourcecode for this algorithm is provided in Supplementary Information.

To help prevent human subjects from memorizing the location of game equilibria, at the beginning of each trial a variable s was chosen uniformly at random from $\{-1,+1\}$ and the map $h \mapsto sh$ was applied to the human subject's manual input for the duration of the trial. When the variable's value was $s=-1$, this had the effect of applying a "mirror" symmetry to the input. The joint action was initialized uniformly at random in the square $[-0.4,+0.4]\times[-0.4,+0.4]\subset\mathbb{R}^2$. Each trial lasted 40 seconds.

Experiment 2: conjectural variation in policy space

In the second experiment, the machine adapted its policy by estimating a *conjecture* about the human's *policy*. To collect the data that was used to form its estimate, the machine played an affine policy in two consecutive trials that differed solely in the constant term,

$$\text{nominal policy } m = L_M h, \tag{4a}$$
$$\text{perturbed policy } m' = L_M h' + \delta. \tag{4b}$$

This algorithm is illustrated in Figure 2F,G and Figure 4C. The machine used the median action vectors $(\widetilde{h},\widetilde{m})$, $(\widetilde{h}',\widetilde{m}')$ from the pair of trials to estimate a conjecture about the human's policy using a ratio of differences,

$$\widetilde{L}_H = \frac{\widetilde{h}'-\widetilde{h}}{\widetilde{m}'-\widetilde{m}}. \tag{5}$$

The machine used this estimate of the human's policy to update its policy as

$$L_M^+ = \frac{1-2\widetilde{L}_H}{1-\widetilde{L}_H}. \tag{6a}$$

In the next pair of trials, the machine employed $m = L_M^+ h$ as its policy. This conjectural variation process was iterated 10 times starting from the initial conjecture $\widetilde{L}_H = 0$, which yields the initial best-response policy $m = h$. Numerical simulation sourcecode for this algorithm is provided in Supplementary Information.

In this experiment, the machine's policy slopes $L_{M,0}, L_{M,1}, \ldots, L_{M,k}, \ldots, L_{M,K-1}$ and the machine's conjectures about the human's policy slopes $\widetilde{L}_{H,0}, \widetilde{L}_{H,1}, \ldots, \widetilde{L}_{H,k}, \ldots, \widetilde{L}_{H,K-1}$ were recorded for each conjectural variation iteration $k \in \{0,\ldots,K-1\}$ where $K = 10$ iterations. In addition, the time series of actions within each trial as in the first experiment, with each trial now lasting only 20 seconds, yielding $T = 1200$ samples used to compute the median action vectors used in (5).

Experiment 3: gradient descent in policy space

In the third experiment, the machine adapted its policy using a policy gradient strategy by playing an affine policy in two consecutive trials that differed only in the linear term,

$$\text{nominal policy } m = L_M h, \tag{7a}$$
$$\text{perturbed policy } m' = (L_M + \Delta) h'. \tag{7b}$$

This algorithm is illustrated in Figure 3F,G and Figure 4D. The machine used the median action vectors $(\widetilde{h},\widetilde{m})$, $(\widetilde{h}',\widetilde{m}')$ from the pair of trials to estimate the gradient of the machine's cost with respect to the linear term in its policy, and this linear term was adjusted to decrease the cost. Specifically, an auxiliary cost was defined as

$$\widetilde{c_M}(L_M) := c_M\left(h, L_M(h-h_M^*)+m_M^*\right), \tag{8}$$

and the pair of trials were used to obtain a finite-difference estimate of the gradient of the machine's cost with respect to the slope of the machine's policy,

$$\partial_{L_M}\widetilde{c_M}(L_M) \approx \frac{1}{\Delta}\left(\widetilde{c_M}(L_M+\Delta) - \widetilde{c_M}(L_M)\right). \tag{9}$$

The machine used this derivative estimate to update the linear term in its policy by descending its cost gradient,

$$L_M^+ = L_M - \gamma\,\partial_{L_M}\widetilde{c_M}(L_M) \tag{10}$$

where γ is the policy gradient adaptation rate parameter ($\gamma = 2$ in this Experiment). Numerical simulation sourcecode for this algorithm is provided in Supplementary Information.

Cost functions and game-theoretic equilibria

H's cost function	M's cost function
$c_H(h,m) = \frac{1}{2}h^2 + \frac{7}{30}m^2 - \frac{1}{3}hm + \frac{2}{15}h - \frac{22}{75}m$	$c_M(h,m) = \frac{1}{2}m^2 + h^2 - hm$

game-theoretic equilibria	H**'s and** M**'s actions**	H**'s and** M**'s policy slopes**
H's optimum	$(h_H^*, m_H^*) = (+0.1, +0.7)$	
M's optimum	$(h_M^*, m_M^*) = (0,0)$	
Nash equilibrium	$(h^{\mathrm{NE}}, m^{\mathrm{NE}}) = (-0.2, -0.2)$	
human-led Stackelberg equilibrium	$(h^{\mathrm{SE}}, m^{\mathrm{SE}}) = (+0.2, +0.2)$	$L_H^{\mathrm{SE}} = -0.2, \quad L_M^{\mathrm{SE}} = 1$
consistent conjectural variations equilibrium	$(h^{\mathrm{CCVE}}, m^{\mathrm{CCVE}}) \approx (0.276, 0.373)$	$L_H^{\mathrm{CCVE}} \approx -0.54, \; L_M^{\mathrm{CCVE}} \approx +1.35$
machine-led reverse Stackelberg equilibrium (equal to M's optimum)	$(h^{\mathrm{RSE}}, m^{\mathrm{RSE}}) = (0,0)$	$L_H^{\mathrm{RSE}} = 1/7, \quad L_M^{\mathrm{RSE}} = 5/11$

Table 1. Cost functions and game-theoretic equilibria of the game studied in Experiments 1, 2, and 3.

Statistical Analysis

To determine the statistical significance of our results, we used Student's t-test and Hotelling's T^2-test with significance threshold $P \leq 0.05$ applied to one- and two-dimensional distributions of median action data from independent populations of $n = 20$ subjects per experiment; both tests are described in Chapter 5 of[40]. Student's t-test was applied to distributions of human actions h at the slowest and fastest learning rates in experiment 1 (corresponding to $\alpha = 0$ and $\alpha = 1$ in Figure 1) where data are constrained to a 1-dimensional subspace of the joint action space ($m = 0$ when $\alpha = 0$ and $m = h$ when $\alpha = 1$). Hotelling's T^2-test was applied in experiments 2 and 3 to distributions of 2-dimensional joint action vectors (h,m) from the final iteration $k = 9$. In all cases we test the null hypothesis that the sample mean is not significantly different from a specified game theory equilibrium (different in each experiment). If the test fails to reject the null hypothesis ($P > 0.05$), we interpret this outcome as evidence that the means are not significantly different, consistent with the guidance in Chapter 5 of[40].

References

1. Sutton, R. T. *et al.* An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine* **3**, 1–10 (2020).
2. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K. & Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv. (CSUR)* **54**, 1–35 (2021).
3. Felt, W., Selinger, J. C., Donelan, J. M. & Remy, C. D. "Body-In-The-Loop": Optimizing device parameters using measures of instantaneous energetic cost. *PLoS ONE* **10**, e0135342, DOI: 10.1371/journal.pone.0135342 (2015).
4. Zhang, J. *et al.* Human-in-the-loop optimization of exoskeleton assistance during walking. *Science* **356**, 1280–1284, DOI: 10.1126/science.aal5054 (2017).
5. Slade, P., Kochenderfer, M. J., Delp, S. L. & Collins, S. H. Personalizing exoskeleton assistance while walking in the real world. *Nature* **610**, 277–282, DOI: 10.1038/s41586-022-05191-1 (2022).
6. Thomas, P. S. *et al.* Preventing undesirable behavior of intelligent machines. *Science* **366**, 999–1004, DOI: 10.1126/science.aag3311 (2019).
7. Taylor, J. A., Krakauer, J. W. & Ivry, R. B. Explicit and implicit contributions to learning in a sensorimotor adaptation task. *J. Neurosci.* **34**, 3023–3032 (2014).

8. Heald, J. B., Lengyel, M. & Wolpert, D. M. Contextual inference underlies the learning of sensorimotor repertoires. *Nature* **600**, 489–493 (2021).
9. Von Neumann, J. & Morgenstern, O. *Theory of games and economic behavior* (Princeton University Press, 1947).
10. Başar, T. & Olsder, G. J. *Dynamic noncooperative game theory* (SIAM, 1998).
11. Varian, H. R. *Microeconomic analysis* (Norton & Company, 1992).
12. Goodfellow, I. *et al.* Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. & Weinberger, K. Q. (eds.) *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, 2672–2680 (Curran Associates, Inc., 2014).
13. Nikolaidis, S., Nath, S., Procaccia, A. D. & Srinivasa, S. Game-theoretic modeling of human adaptation in human-robot collaboration. In *ACM/IEEE Conference on Human-Robot Interaction (HRI)*, 323–331 (2017).
14. Madduri, M. M., Burden, S. A. & Orsborn, A. L. A game-theoretic model for co-adaptive brain-machine interfaces. In *IEEE/EMBS Conference on Neural Engineering (NER)*, 327–330 (IEEE, 2021).
15. Li, Y., Carboni, G., Gonzalez, F., Campolo, D. & Burdet, E. Differential game theory for versatile physical human-robot interaction. *Nat. Mach. Intell.* **1**, 36–43 (2019).
16. Ng, A. Y. & Russell, S. J. Algorithms for inverse reinforcement learning. In *ACM International Conference on Machine Learning (ICML)*, 663–670 (2000).
17. Nash, J. F. Equilibrium points in *N*-Person games. *Proc. Natl. Acad. Sci. United States Am.* **36**, 48–49, DOI: 10.1073/pnas.36.1.48 (1950).
18. von Stackelberg, H. *Marktform und Gleichgewicht* (Springer, 1934).
19. Bowley, A. L. *The mathematical groundwork of economics: an introductory treatise* (Clarendon Press, 1924).
20. Ho, Y.-C., Luh, P. B. & Olsder, G. J. A control-theoretic view on incentives. *Automatica* **18**, 167–179, DOI: 10.1016/0005-1098(82)90106-6 (1982).
21. Huang, J. *et al.* Internal models in control, biology and neuroscience. In *IEEE Conference on Decision and Control (CDC)*, 5370–5390, DOI: 10.1109/CDC.2018.8619624 (2018).
22. Wolpert, D. M., Ghahramani, Z. & Jordan, M. I. An internal model for sensorimotor integration. *Science* **269**, 1880–1882, DOI: 10.1126/science.7569931 (1995).
23. Palan, S. & Schitter, C. Prolific.com—a subject pool for online experiments. *J. Behav. Exp. Finance* **17**, 22–27, DOI: 10.1016/j.jbef.2017.12.004 (2018).
24. Ma, Y.-A., Chen, Y., Jin, C., Flammarion, N. & Jordan, M. I. Sampling can be faster than optimization. *Proc. Natl. Acad. Sci. United States Am.* **116**, 20881–20885, DOI: 10.1073/pnas.1820003116 (2019).
25. Chasnov, B., Ratliff, L., Mazumdar, E. & Burden, S. A. Convergence analysis of gradient-based learning in continuous games. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, vol. 115 of *Proceedings of Machine Learning Research*, 935–944 (2020).
26. Figuières, C., Jean-Marie, A., Quérou, N. & Tidball, M. *Theory of Conjectural Variations* (World Scientific, 2004).
27. March, C. Strategic interactions between humans and artificial intelligence: Lessons from experiments with computer players. *J. Econ. Psychol.* **87**, 102426 (2021).
28. Perdikis, S. & d. R. Millán, J. Brain-Machine interfaces: A tale of two learners. *IEEE Syst. Man, Cybern. Mag.* **6**, 12–19, DOI: 10.1109/MSMC.2019.2958200 (2020).
29. De Santis, D. A framework for optimizing co-adaptation in body-machine interfaces. *Front. Neurorobotics* **15**, 40, DOI: 10.3389/fnbot.2021.662181 (2021).
30. Recht, B. A tour of reinforcement learning: The view from continuous control. *Annu. Rev. Control. Robotics, Auton. Syst. (ARCRAS)* **2**, 253–279, DOI: 10.1146/annurev-control-053018-023825 (2019).
31. Abram, S. J. *et al.* General variability leads to specific adaptation toward optimal movement policies. *Curr. biology: CB* **32**, 2222–2232.e5, DOI: 10.1016/j.cub.2022.04.015 (2022).
32. Ingraham, K. A., Remy, C. D. & Rouse, E. J. The role of user preference in the customized control of robotic exoskeletons. *Sci. Robotics* **7**, eabj3487, DOI: 10.1126/scirobotics.abj3487 (2022).
33. Harsanyi, J. C. Games with incomplete information played by "Bayesian" players, I-III. *Manag. science* **8**, 159–182, 320–334, 486–502 (1967).

34. Gershman, S. J., Horvitz, E. J. & Tenenbaum, J. B. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science* **349**, 273–278 (2015).

35. Petersen, S. E. & Posner, M. I. The attention system of the human brain: 20 years after. *Annu. review neuroscience* **35**, 73–89, DOI: 10.1146/annurev-neuro-062111-150525 (2012).

36. Cooper, A. F., Moss, E., Laufer, B. & Nissenbaum, H. Accountability in an algorithmic society: Relationality, responsibility, and robustness in machine learning. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 864–876, DOI: 10.1145/3531146.3533150 (2022).

37. Tversky, A. & Kahneman, D. Judgment under uncertainty: Heuristics and biases. *Science* **185**, 1124–1131, DOI: 10.1126/science.185.4157.1124 (1974).

38. Hilbert, M. & López, P. The world's technological capacity to store, communicate, and compute information. *Science* **332**, 60–65, DOI: 10.1126/science.1200970 (2011).

39. Jordan, M. I. & Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science* **349**, 255–260, DOI: 10.1126/science.aaa8415 (2015).

40. Rencher, A. C. & Christensen, W. F. *Methods of Multivariate Analysis* (John Wiley & Sons, Inc, 2012).

Acknowledgements

Funding

U.S. National Institutes of Health 5T90DA032436-09 MPI (BJC)
U.S. National Science Foundation Award #1836819 (LJR)
U.S. National Science Foundation Award #2045014 (SAB)

Author contributions

Conceptualization: BJC, LJR, SAB
Methodology: BJC, LJR, SAB
Investigation: BJC
Visualization: BJC
Supervision: LJR, SAB
Writing—original draft: BJC, LJR, SAB
Writing—review & editing: BJC, LJR, SAB

Data and materials availability

The data and analysis scripts needed to reproduce all figures and statistical results reported in both the main paper and supplement are publicly available in a Code Ocean capsule, `codeocean.com/ capsule/6746258`. The sourcecode used to conduct experiments on the Prolific platform are publicly available on GitHub, `github.com/dynams/web`.

Additional information

Competing interests

The authors filed the following patent application pertaining to the results presented in this paper: *WO2023003979 - OPTIMAL DATA-DRIVEN DECISION-MAKING IN MULTI-AGENT SYSTEMS*. The co-inventors on this patent are Benjamin Chasnov, Lillian Ratliff, Samuel Burden, Amy Orsborn, Tanner Fiez, Maneeshika Madduri, Joseph Sullivan, and Momona Yamagami. This patent specifically pertains to the algorithm studied in Experiment 2.

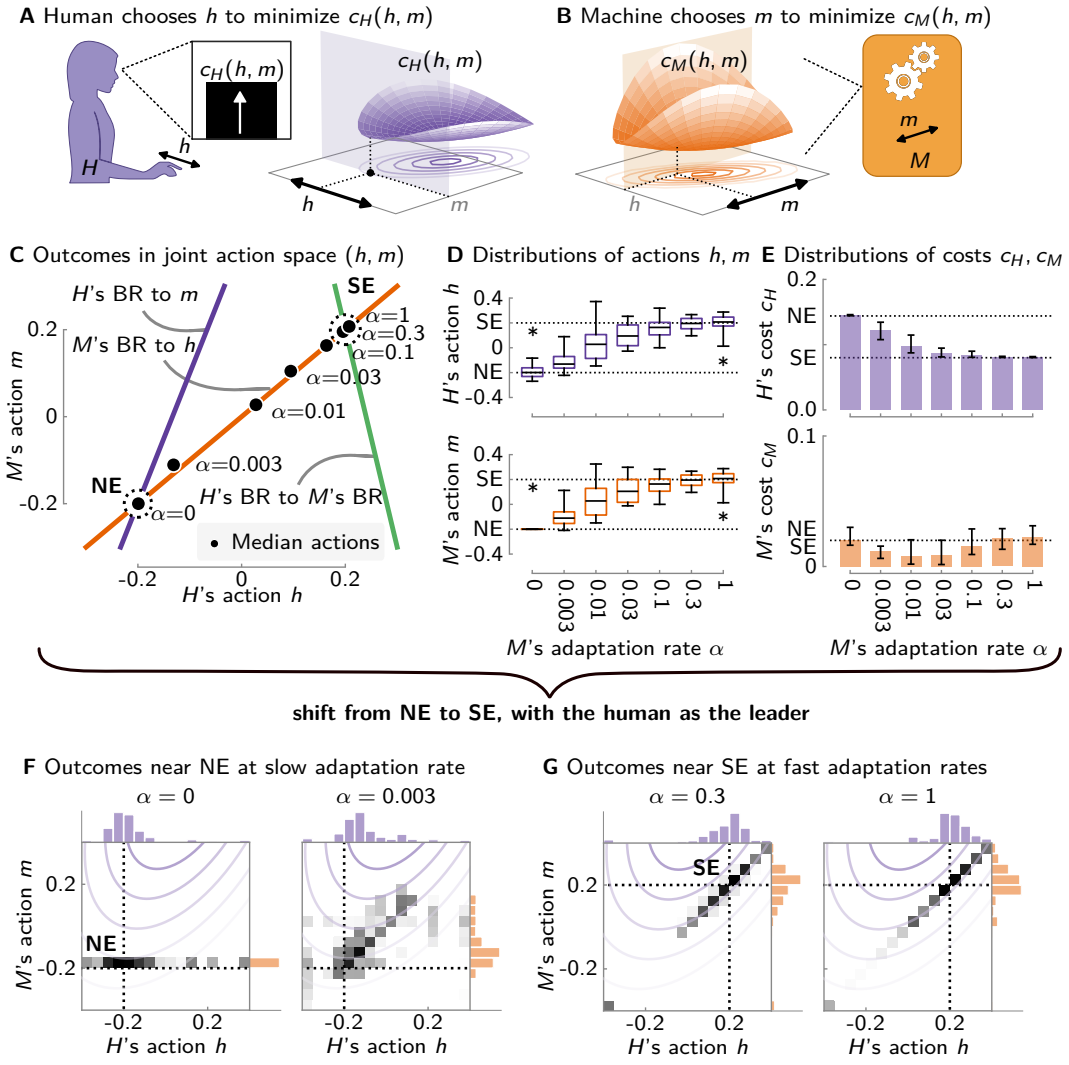

Figure 1. Gradient descent in action space (Experiment 1, $n = 20$). (**A**) Each human subject H is instructed to provide manual input h to make a black bar on a computer display as small as possible. The bar's height represents the value of a prescribed cost c_H. (**B**) The machine M has its own cost c_M chosen to yield game-theoretic equilibria that are distinct from each other and from each player's global optima. The machine knows its cost and observes human actions h. In this experiment, the machine updates its action by gradient descent on its cost $\frac{1}{2}m^2 - hm + h^2$ with adaptation rate α. (**C**) Median joint actions for each α overlaid on game-theoretic equilibria and best-response (BR) curves that define the Nash equilibrium (NE) and Stackelberg equilibrium (SE), respectively). (**D**) Action distributions for each machine adaptation rate displayed by box-and-whiskers plots showing 5th, 25th, 50th, 75th, and 95th percentiles. Statistical significance ($*$) determined by comparing initial and final distributions to NE and SE actions using Student's t-test ($P < 0.001$ comparing to SE and $P = 0.2$ comparing to NE at $\alpha = 0$; $P < 0.001$ comparing to NE and $P = 0.5$ comparing to SE at $\alpha = 1$). (**E**) Cost distributions for each machine adaptation rate displayed using box plots with error bars showing 25th, 50th, and 75th percentiles. (**F,G**) One- and two-dimensional histograms of actions for different adaptation rates ($\alpha \in \{0,0.003\}$ in (F), $\alpha \in \{0.3, 1\}$ in (G)) with game-theoretic equilibria overlaid (NE in (F), SE in (G)).

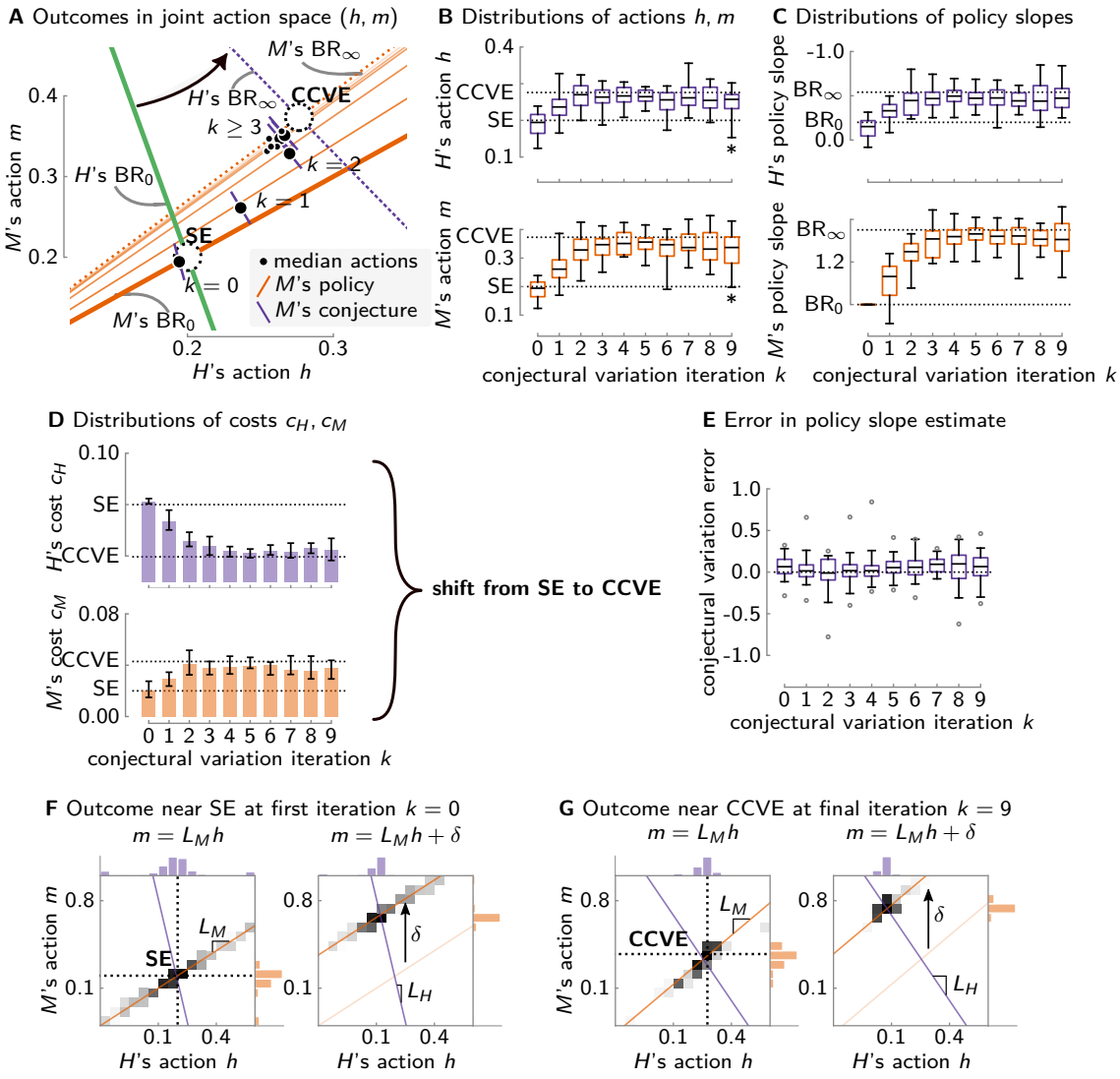

Figure 2. Conjectural variation in policy space (Experiment 2, $n = 20$). Experimental setup and costs are the same as Figure 1A,B except that the machine uses a different adaptation algorithm: in this experiment M iteratively implements policies $m = L_M h$, $m = L_M + \delta$ to measure and best-respond to conjectures of the human's policy and updates the policy slope L_M. (**A**) Median actions, conjectures, and policies for each conjectural variation iteration k overlaid on game-theoretic equilibria corresponding to best-responses (BR) at initial and limiting iterations (BR_0 and BR_∞, respectively) predicted from Stackelberg equilibrium (SE) and consistent conjectural variations equilibrium (CCVE) of the game, respectively. (**B**) Action distributions for each iteration displayed by box-and-whiskers plots as in Figure 1D. (**C**) Policy slope distributions for each iteration displayed with the same conventions as B; note that the sign of the top y-axis is reversed for consistency with other plots. Statistical significance ($*$) determined by comparing action distribution at iteration $k = 9$ to SE and CCVE using Hotelling's T^2 test ($P < 0.001$ for SE and $P = 0.06$ for CCVE). (**D**) Cost distributions for each iteration displayed using box-and-whiskers plots as in Figure 1E. (**E**) Error between measured and theoretically-predicted machine conjectures about human policies at each iteration displayed as box-and-whiskers plots as in B,C. (**F,G**) One- and two-dimensional histograms of actions for different iterations ($k = 0$ in F, $k = 9$ in G) with policies and game-theoretic equilibria overlaid (SE and BR_0 in F, CCVE and BR_∞ in G).

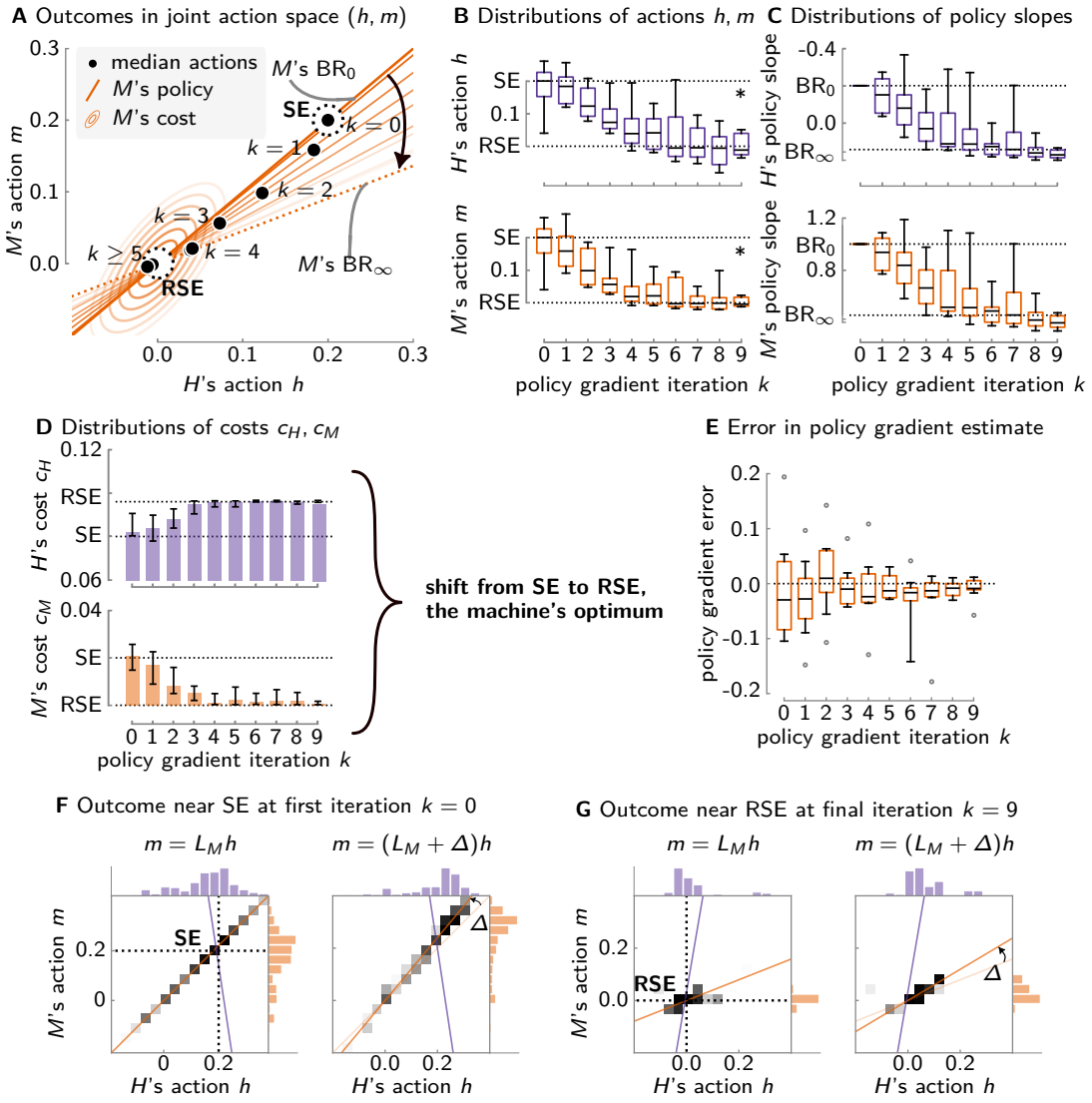

Figure 3. Gradient descent in policy space (Experiment 3, $n = 20$). Experimental setup and costs are the same as Figure 1A,B except that the machine uses a different adaptation algorithm: in this experiment, M iteratively implements linear policies $m = L_M h$, $m = (L_M + \Delta)h$ to measure the gradient of its cost with respect to its policy slope parameter L_M and updates this parameter to descend its cost landscape. (**A**) Median actions and policies for each policy gradient iteration k overlaid on game-theoretic equilibria corresponding to machine best-responses (BR) at initial and limiting iterations (BR_0 and BR_∞, respectively) predicted from the Stackelberg equilibrium (SE) and the machine's global optimum (RSE), respectively. (**B**) Action distributions for each iteration displayed by box-and-whiskers plots as in Figure 1D. (**C**) Policy slope distributions for each iteration displayed with the same conventions as B; note that the sign of the top subplot's y-axis is reversed for consistency with other plots. (**D**) Cost distributions for each iteration displayed using box-and-whiskers plots as in Figures 1E and 2D. Statistical significance ($*$) determined by comparing action distribution at iteration $k = 9$ to SE and RSE using Hotelling's T^2 test ($P < 0.001$ comparing to SE and $P = 0.11$ comparing to RSE). (**E**) Error between measured and theoretically-predicted policy slopes at each iteration displayed as box-and-whiskers plots as in B,C. (**F,G**) One- and two-dimensional histograms of actions for different iterations ($k = 0$ in F, $k = 9$ in G) with policies and game-theoretic equilibria overlaid (SE in F, RSE in G).

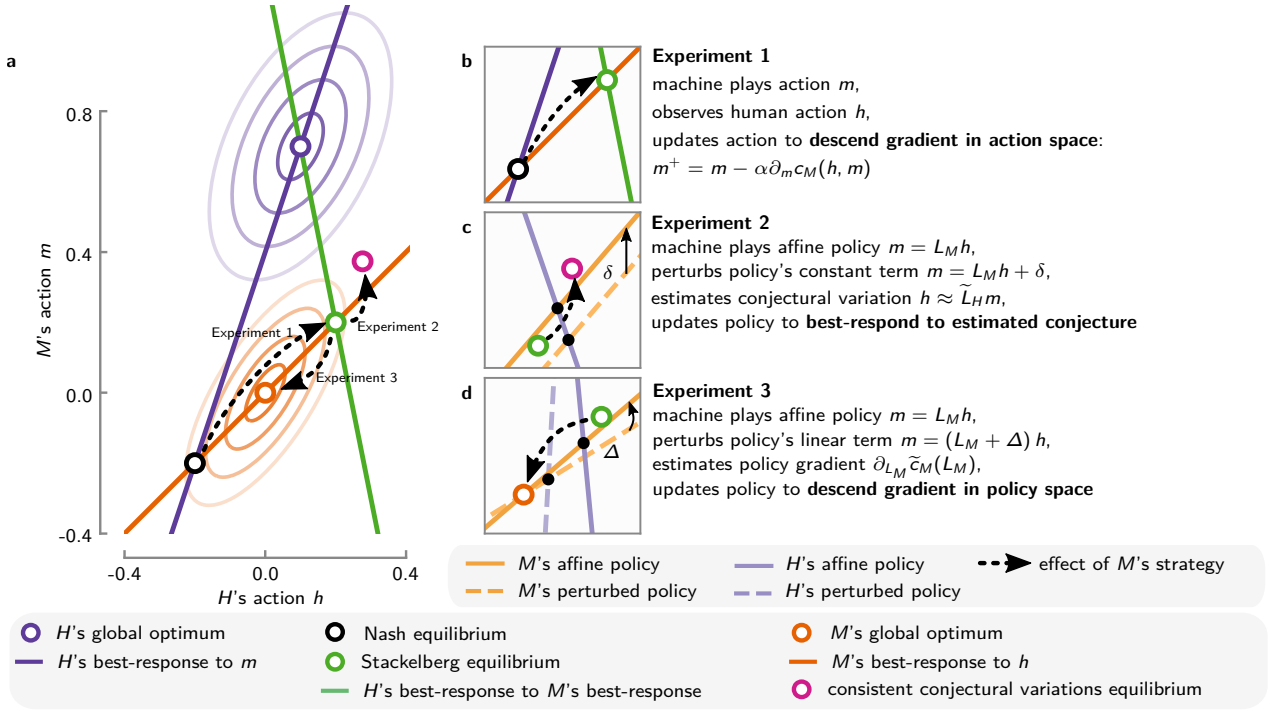

Figure 4. Overview of co-adaptation experiment between human and machine. Human subject H is instructed to provide manual input h to make a black bar on a computer display as small as possible. The machine M has its own prescribed cost c_M chosen to yield game-theoretic equilibria that are distinct from each other and from each player's global optima. (**a**) Joint action space illustrating game-theoretic equilibria and response functions determined from the costs prescribed to human and machine: *global optima* defined by minimizing with respect to both variables; *best-response* functions defined by fixing one variable and minimizing with respect to the other. Machine plays different strategies in three experiments: (**b**) gradient descent in *Experiment 1*; (**c**) conjectural variation in *Experiment 2*; (**d**) policy gradient descent in *Experiment 3*.

Supplementary Information for: Human adaptation to adaptive machines converges to game-theoretic equilibria

Benjamin J. Chasnov[1], **Lillian J. Ratliff**[1], **and Samuel A. Burden**[1,*]

[1]Department of Electrical & Computer Engineering, University of Washington, Seattle, WA 98195, USA
[*]Corresponding author: sburden@uw.edu

ABSTRACT

Here we test three learning algorithms for machines playing general-sum games with human subjects. The algorithms enable the machine to select the outcome of the co-adaptive interaction from a constellation of game-theoretic equilibria in action and policy spaces. Importantly, the machine learning algorithms work directly from observations of human actions without solving an inverse problem to estimate the human's utility function as in prior work. Surprisingly, one algorithm can steer the human-machine interaction to the machine's optimum, effectively controlling the human's actions even while the human responds optimally to their perceived cost landscape. Our results show that game theory can be used to predict and design outcomes of co-adaptive interactions between intelligent humans and machines.

Supplementary Information

Supplementary Methods

Additional experiments, whose results are reported in this Supplement but not the main paper, were conducted with different quadratic and non-quadratic costs to demonstrate the generality of the experiment and theory. First, Experiment 3 was repeated with a different initialization of the machine's policy: instead of initializing the machine's policy to $m = h$, it was initialized to $m = 0$. Next, Experiment 3 was repeated 9 times with different global optima for the machine: the machine's quadratic cost re-parameterized as

$$c_M(h,m) = \frac{1}{2}(m-m_M^*)^2 - (m-m_M^*)(h-h_M^*) + (h-h_H^*)^2$$

with $h_M^* \in \{-0.1, 0, +0.1\}$ and $m_M^* \in \{-0.1, 0, +0.1\}$ to test whether the machine can drive the behavior to any one of a finite set of points in the joint action space, and to test whether the reverse-Stackelberg equilibrium $(h^{\mathrm{RSE}}, m^{\mathrm{RSE}}) = (h_M^*, m_M^*)$ is a stable equilibrium of policy gradient. Subsequently, Experiments 1, 2, and 3 were repeated with non-quadratic cost functions in the *Cobb-Douglas* form:

$$c_H(h,m) = 1 - 2(1-h)^{0.175}(h+1.1m)^{0.5} \tag{1}$$

was used in replicates of Experiments 1, 2, and 3;

$$c_M(h,m) = 1 - 2(1-m)^{0.2}(m+1.1h)^{0.5} \tag{2}$$

was used in replicates of Experiments 1 and 2, and

$$c_M(h,m) = (m-m_M^*)^2 + (h-h_M^*)^2 \text{ with } (m_M^*, h_M^*) = (0.5, 0.5) \tag{3}$$

was used in replicates of Experiment 3. Pairing c_H from (1) with c_M from (2) yields the following game-theoretic equilibria in the replicates of Experiments 1 and 2:

$$\begin{aligned}(h^{\mathrm{NE}}, m^{\mathrm{NE}}) &\approx (0.590, 0.529),\\ (h^{\mathrm{SE}}, m^{\mathrm{SE}}) &\approx (0.429, 0.579),\\ (h^{\mathrm{CCVE}}, m^{\mathrm{CCVE}}) &\approx (0.392, 0.336).\end{aligned}$$

Pairing c_H from (1) with c_M from (3) yields the following equilibrium in the replicates of Experiment 3:

$$(h^{\mathrm{RSE}}, m^{\mathrm{RSE}}) = (0.5, 0.5).$$

The human's actions were constrained to $[0.2, 0.8]$ in these replicates of the experiments and the manual input was accordingly normalized to this range. The machine's actions were constrained to $[0, 1]$. Experiment-specific changes to protocol designs are described in subsequent subsections.

Experiment 1: gradient descent in action space

The preceding methods were modified as follows for the experiments with non-quadratic costs: five adaptation rates were tested: $\alpha \in \{0, 0.003, 0.03, 0.3, \infty\}$ where $\alpha = 0$ corresponds to the machine player the Nash equilibrium, and $\alpha = \infty$ corresponds to the machine implementing its best-response, $m = -\frac{77}{270}h + \frac{20}{27}$; the joint action was initialized uniformly at random in the square $[0.3, 0.7] \times [0.3, 0.7] \subset \mathbb{R}^2$.

Experiment 2: conjectural variation in policy space

The preceding methods were modified as follows for the experiments with non-quadratic costs: given machine's non-quadratic cost in Cobb-Douglas form

$$c_M(h, m) = 1 - 2(1 - m)^{a_M}(m + d_M h)^{b_M} \tag{4}$$

where $a_M, b_M > 0$ and $d_M \geq 1$, the machine's conjectural variation iteration is

$$L_{M,k+1} = -\frac{a_M d_M}{a_M + b_M + b_M d_M L_H}, \tag{5a}$$

$$\ell_{M,k+1} = \frac{b_M + b_M d_M L_H}{a_M + b_M + b_M d_M L_H}, \tag{5b}$$

given the estimated human policy slope L_H. Additionally, the policy constant perturbation value was set to $\delta = 0.15$, which was chosen to within the appropriate scale of the Cobb-Douglas costs.

Experiment 3: gradient descent in policy space

The policy gradient approach required no modifications. The same algorithm with policy slope perturbation $\Delta = 0.2$ and learning rate $\gamma = 2$ was used.

Supplementary Results

Additional experiments were conducted with different quadratic and non-quadratic costs to demonstrate the generality of the experimental and theoretical results. Figures displaying these supplementary results (Figures 1–4) use violin plots to better visualize the full distribution of outcomes while still showing quartile ranges using dotted lines within each violin.

Machine initialization (Experiment 3)

To demonstrate that the outcome of the machine's policy gradient adaptation algorithm does not depend on the initialization of the machine's policy, we repeated Experiment 3 with initial policy slope to $L_M = 0$. Iterating policy gradient shifted the distribution of median action vectors for a population of human subjects to the machine's global optimum (Figure 1).

Machine optimum (Experiment 3)

To demonstrate that the machine can drive the human action to any point in the action space so long as the joint action profile is stable, the three experiments were conducted with differing machine minima. A grid of machine minima were tested $h_M^* \in \{-0.1, 0, +0.1\}$ and $m_M^* \in \{-0.1, 0, +0.1\}$. Iterating policy gradient descent shifted the distribution of median action vectors for a population of human subjects to the machine's global optimum (Figure 2).

Non-quadratic costs (modified Experiments 1, 2, and 3)

To demonstrate the generality of the experiments and theory, we conducted modified Experiments 1, 2 and 3 using non-quadratic costs. In Experiment 1, the distributions of median action vectors for a population of human subjects shifted from the Nash equilibrium at the slowest rate to the human-led Stackelberg equilibrium at the fastest adaptation rate (Figure 3A). In Experiment 2, iterating the process of estimating conjectural variations shifted the distribution of median action vectors for a population of human subjects from the human-led Stackelberg equilibrium to a consistent conjectural variations equilibrium (Figure 3B). In Experiment 3, iterating policy gradient descent shifted the distribution of median action vectors for a population of human subjects to the machine's global minimum (Figure 3C).

Numerical simulations

To provide simple descriptive models for the outcomes observed in each of the three Experiments, numerical simulations were implemented using the Python programming language version 3.8. The shared parameter, cost and gradient definitions are included in Sourcecode S1. The results from the simulation are overlaid on top of the violin data plots from the main paper (Figure 4). In Experiment 1, the simulation captures the transition from the Nash equilibrium at the slowest rate to the human-led Stackelberg equilibrium at the fastest rate (Figure 4A). In Experiment 2, the simulation captures the transition from the human-led Stackelberg equilibrium to the consistent conjectural variations equilibrium (Figure 4B). In Experiment 3, the simulation captures the transition from the human-led Stackelberg equilibrium to the machine's global optimum (Figure 4C). These simulations help bridge the gap between theoretical equilibrium analysis and experimental observations, offering insight into the transient dynamics that lead to the observed equilibria.

The Python implementation of our simulations is structured around common cost functions and gradient calculations across all experiments. Sourcecode S1 contains these shared definitions, including the quadratic cost parameters, cost functions, and gradient calculations that form the mathematical foundation for all three experimental simulations. The subsequent experiment-specific code (Sourcecodes S2-S4) builds upon these definitions.

Experiment 1 To predict what happens in the range of adaptation rates between the two limiting cases (i.e. for $0 < \alpha < \infty$), a simulation of the human's behavior was implemented based on approximate gradient descent. The model of the human uses finite differences to estimate the derivative of its cost (c_H) with respect to its action (h) and then adapts its action to descend this cost gradient. Importantly, it is assumed that the human performs these derivative estimation and gradient descent procedures slower than the machine, i.e. the human takes one gradient step for every K machine steps. Since the machine's steps occur at a rate of 60 samples per second, this timescale difference corresponds to the human taking steps at a rate of $60/K$ samples per second.

In Sourcecode S2, we simulate a range of machine adaptation rates and observe how the equilibrium shifts. The simulation alternates between machine adaptation phases and human gradient adaptation phases. It reproduces the transition from Nash equilibrium at slow rates to Stackelberg equilibrium at fast rates as observed in the experimental data.

Experiment 2 To predict what happens when the machine perturbs the constant term of its policy and uses the outcome to estimate of the human's policy slope, a simulation of the conjectural variations iteration was implemented. The machine best responds to the human's policy. The model of the human uses the derivative of its cost (c_H) assuming that the machine's action (m) is related to its own action (h) by conjectural variation ($L_{M,k}$) and then adapts its action to descent this cost gradient. It is assumed that the machine observes the human and machine's actions to compute the estimate of the human's policy slope ($\widetilde{L}_{H,k}$).

In Sourcecode S3, we simulate ten iterations of conjectural variation. For each iteration, the machine runs paired trials with and without a constant perturbation to its policy. After each pair of trials, the machine updates its policy slope to the mathematically optimal response given its estimate of the human's policy. The simulation demonstrates how this iterated process converges to the Consistent Conjectural Variations Equilibrium (CCVE).

Experiment 3 To predict what happens when the machine perturbs the linear term of its policy, a simulation was implemented based on policy gradient. The model of the human is the same as the previous simulation of Experiment 2. The machine uses the gradient estimate of the observed cost, and does not require observe the human's action or policy as was required in the previous experiment.

In Sourcecode S4, we simulate ten iterations of policy gradient descent. For each policy gradient iteration, the machine runs paired trials with perturbed and unperturbed policy slopes, measures the resulting costs, then updates its policy in the direction that reduces its cost. This approach demonstrates how policy gradient can guide the joint actions toward the machine's optimal point without requiring explicit modeling of the human's policy.

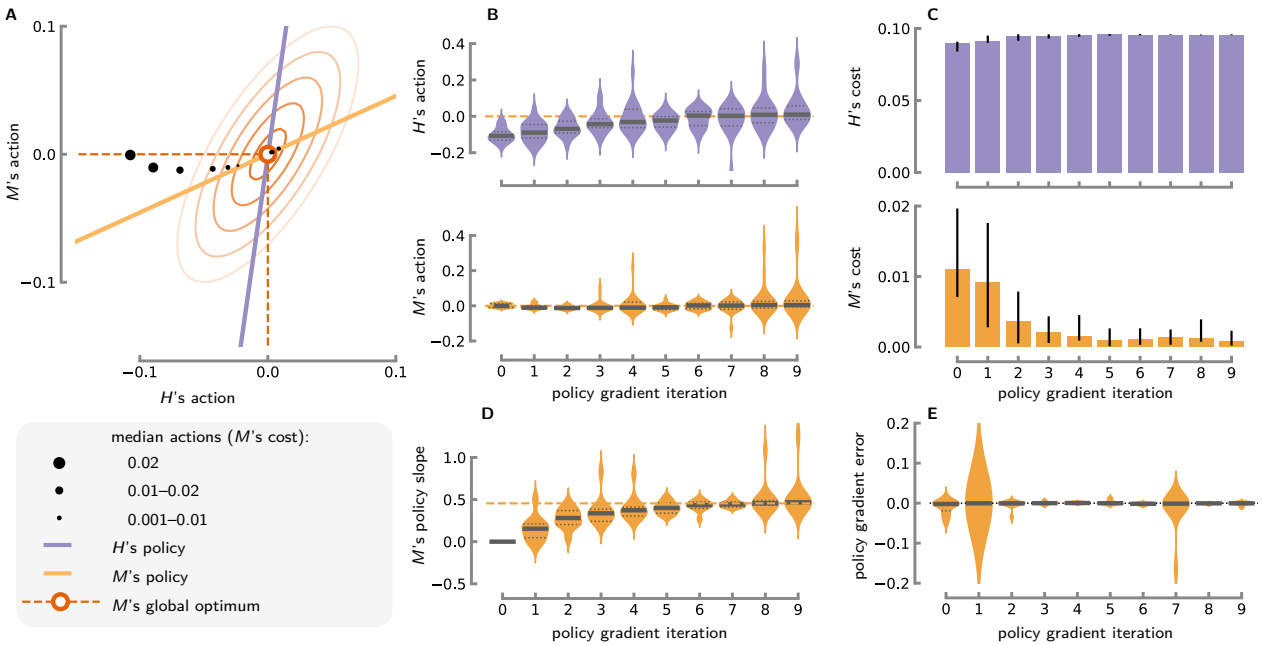

Figure 1. **Experiment 3 with different initial policy** ($n = 20$): gradient descent in policy space for a different initial machine policy. (**A**) Game-theoretic equilibria and best-response functions. (**B**) Decision vector distributions. (**C**) Cost distributions. (**D**) Machine policy slopes. (**E**) Estimation error of machine policy gradients. Action interquartile range (IQR) in (B) contains the machine's minimum at each iteration 4 to 9. Machine's policy slope distribution IQR in (D) reaches the theoretically-predicted slope that would yield the machine's minimum as the game outcome. The machine's policy gradient IQR in (E) contains the theoretical gradient at every policy gradient iteration.

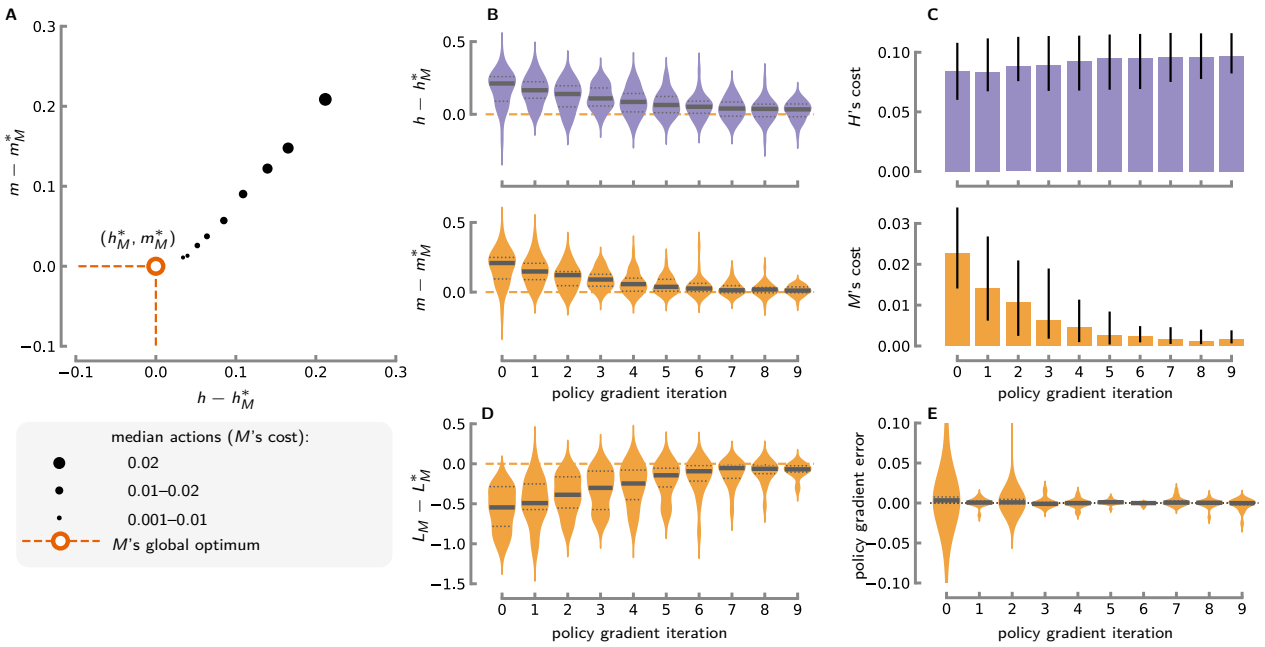

Figure 2. **Experiment 3 with different machine optima** ($n = 18$): gradient descent in policy space for differing machine optima. (**A**) Game-theoretic equilibria and best-response functions. (**B**) Decision vector distributions. (**C**) Cost distributions. (**D**) Machine policy slopes. (**E**) Estimation error of machine policy gradients. Action IQR in (B) contains the machine's minimum at each iteration 7 to 9. Machine's policy slope distribution IQR in (D) approaches the theoretically-predicted slope that would yield the machine's minimum as the game outcome.

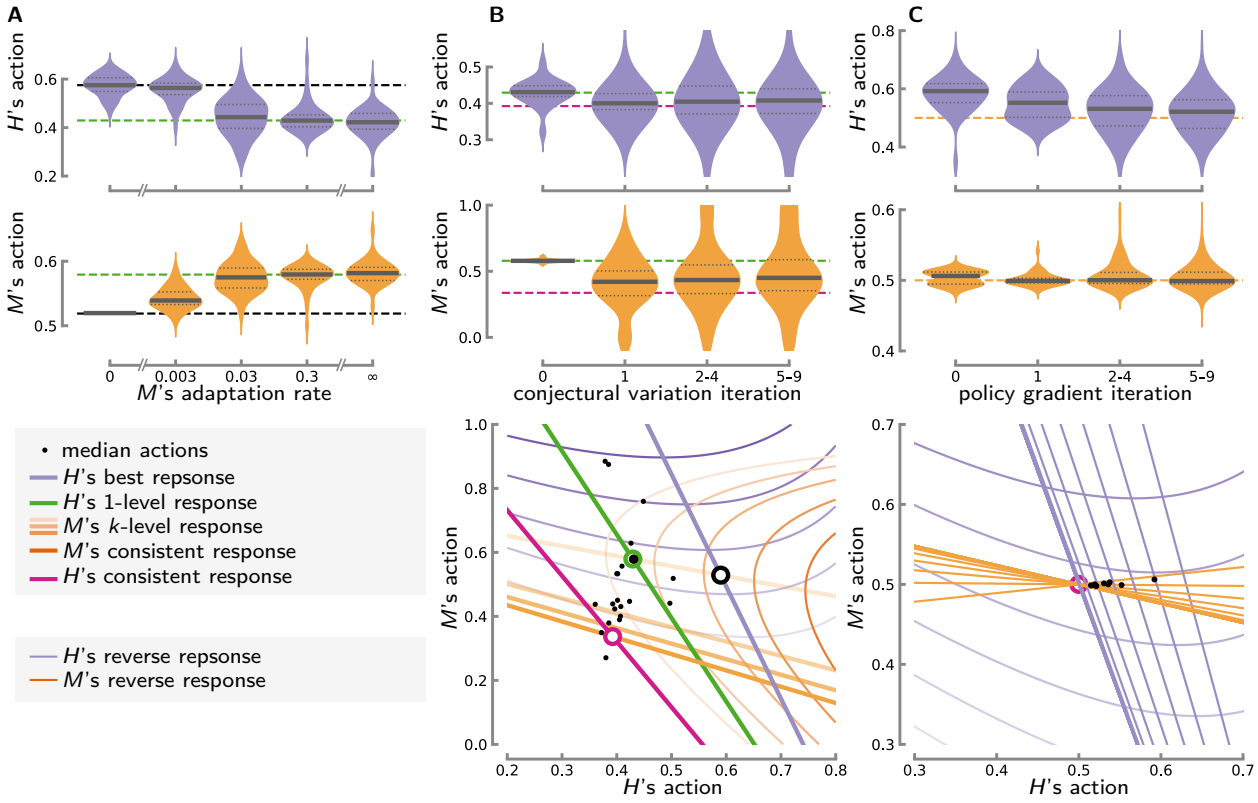

Figure 3. **Modified Experiments 1, 2 and 3 with Cobb-Douglas costs** ($n = 20 \times 3$): (**A**) Gradient descent in action space; decision vector distributions. (**B**) Conjectural variation in policy space; decision vector distributions, game theoretic equilibria and best-response functions. (**C**) Gradient descent in policy space; decision vector distributions and policy gradient iterations.

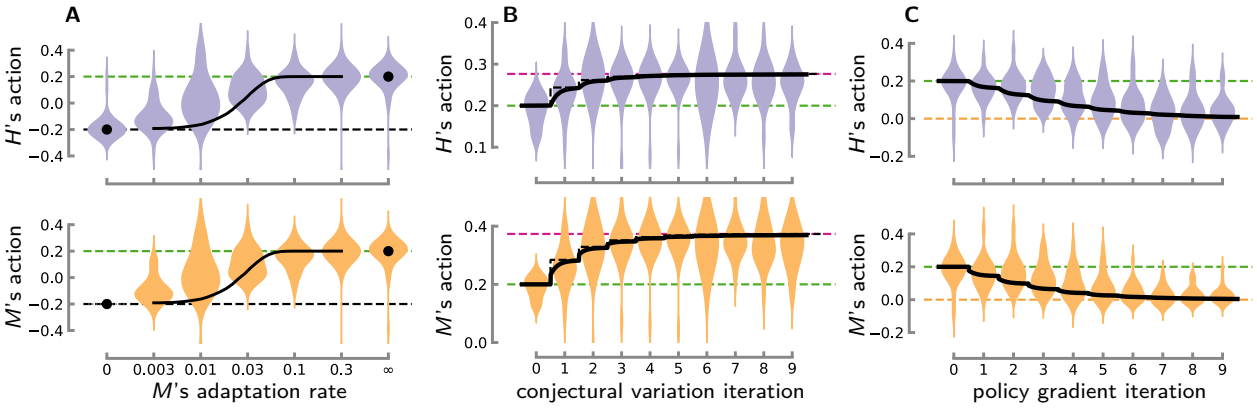

Figure 4. **Simulations of Experiments 1, 2 and 3**: Solid lines and dots are the simulation data, overlaid on violin plots of the distribution of actions. Dashed lines are analytical predictions. (**A**) Gradient descent in action space. (**B**) Conjectural variation in policy space. (**C**) Gradient descent in policy space.

```
T = 10000   # Number of time samples for simulation

# These parameters create the cost landscape described in the paper
# They are carefully chosen to create distinct equilibria (Nash, Stackelberg, CCVE)
AH, BH, DH, hH, mH = 1, -1/3, 7/15, 1/10, 7/10  # hH,mH is human's optimum
AM, BM, DM, hM, mM = 1, -1, 2, 0, 0             # hM,mM is machine's optimum

# Human's cost function (quadratic with interaction term)
def cost_H(h, m):
    return AH*(h-hH)**2/2 + (h-hH)*BH*(m-mH) + DH*(m-mH)**2/2

# Machine's cost function (quadratic with interaction term)
def cost_M(h, m):
    return AM*(m-mM)**2/2 + (h-hM)*BM*(m-mM) + DM*(h-hM)**2/2

# Human's gradient considering machine's policy slope LM
def grad_H(h, m, LM):
    return AH*(h-hH) + BH*(m-mH) + LM*(BH*(h-hH) + DH*(m-mH))

# Machine's gradient considering human's policy slope LH
def grad_M(h, m, LH):
    return AM*(m-mM) + BM*(h-hM) + LH*(BH*(h-hH) + DH*(m-mH))
```

Sourcecode S1: Definitions of parameters, cost functions and gradients of the two players.

```
# Machine's adaptation rates - logarithmically spaced
alphas = [3*10**(i/10) for i in range(-29,-9)]

# Human's adaptation rate (estimated from experimental data)
beta = 0.003 # human's adaptation rate (assumed)

# Small perturbation for estimating human's cost gradient
delta = 1e-5

results = []
for alpha in alphas:
    # Scale iterations to maintain relative adaptation rates
    K = ceil(alpha/beta)          # Number of machine steps per human step
    N = ceil(T/K)*K+1             # Total iterations
    h,m = [0]*N, [0]*N            # Initialize action trajectories

    for t in range(0, T, K):      # Main simulation loop
        c_H = []                  # Track human's observed costs

        for d in [delta, 0]:      # Perturb human action to estimate gradient
            for k in range(t, t+K):
                # Human maintains perturbed action while machine adapts
                h[k] = h[t] + d   # Apply small perturbation to estimate gradient
                # Machine performs gradient descent on its cost
                m[k+1] = m[k] - alpha*grad_M(h[k], m[k], 0)
            c_H.append(cost_H(h[k], m[k]))  # Record final cost for perturbation

        # Estimate human's cost gradient using finite difference
        gradH = (c_H[0]-c_H[1])/delta

        # Human update: perform gradient descent scaled by relative rates
        h[t+K] = h[t] - K*beta*gradH
        m[t+K] = m[k+1]  # Keep machine's final action
```

Sourcecode S2: Numerical simulation of Experiment 1. This simulation models how machine adaptation rate affects equilibrium outcomes. We test a range of adaptation rates from very slow (approximately Nash) to very fast (approximately Stackelberg).

```
K = 10       # Number of conjectural variation iterations (matching experiment)
delta = 1e-1 # Perturbation size for policy constant term

# Initialize action and policy trajectories
h,m = [0]*(K*T+1), [0]*(K*T+1)  # Actions for all iterations and time steps
LH,LM = [0]*(K+1), [0]*(K+1)    # Policy slopes for each iteration
LM[0] = -BM/AM                  # Initial machine policy slope (Nash best response)

# Conjectural variation loop - iteratively estimate and respond to human policy
for k in range(K):
    h_, m_ = [], []             # Store steady-state actions for estimation

    for d in [delta, 0]:        # Run trials with and without perturbation
        for t in range(k*T, k*T + T):
            # Human updates assuming machine follows policy m = LM*h
            h[t+1] = h[t] - beta*grad_H(h[t], m[t], LM[k])

            # Machine implements linear policy with possible constant perturbation
            m[t+1] = LM[k]*(h[t]-hM) + mM + d

        # Store final steady-state actions for this trial
        h_.append(h[t+1])
        m_.append(m[t+1])

    # Estimate human's policy slope from response to perturbation
    # Calculated as change in h divided by change in m
    LH[k+1] = (h_[1] - h_[0])/(m_[1] - m_[0])

    # Update machine policy to best-response given estimated human policy
    # This is derived from the optimality condition for quadratic costs
    LM[k+1] = -(BM + LH[k+1]*DM)/(AM + LH[k+1]*BM)
```

Sourcecode S3: Numerical simulation of Experiment 2. This simulation models how machines can estimate and respond to human policies. The machine perturbs its policy constant term to estimate human policy slope.

```
K = 10       # Number of policy gradient iterations (matching experiment)
Delta = 1e-1 # Perturbation size for policy slope
beta = 3e-3  # Human's learning rate (estimated from experimental data)
gamma = 2    # Policy gradient step size (learning rate for policy updates)

# Initialize action and policy trajectories
h,m = [0]*(K*T+1), [0]*(K*T+1)  # Actions for all iterations and time steps
LH,LM = [0]*(K+1), [0]*(K+1)    # Policy slopes for each iteration
LM[0] = -BM/AM                  # Initial machine policy (Nash best response)

# Main policy gradient loop - estimate gradient and update policy
for k in range(K):
    c_M = []                    # Track machine's observed costs

    for D in [Delta, 0]:        # Run trials with and without slope perturbation
        for t in range(k*T, k*T+T):
            # Human updates assuming machine follows perturbed policy
            h[t+1] = h[t] - beta*grad_H(h[t], m[t], LM[k] + D)

            # Machine implements perturbed linear policy
            # The form is: m = (LM+D)*(h-hM) + mM
            m[t+1] = (LM[k] + D)*(h[t] - hM) + mM

        # Store machine's final cost for gradient estimation
        c_M.append(cost_H(h[t], m[t]))

    # Estimate policy gradient using finite difference method
    gradM = (c_M[0] - c_M[1])/Delta

    # Update machine's policy slope using gradient descent
    LM[k+1] = LM[k] - gamma*gradM
```

Sourcecode S4: Numerical simulation of Experiment 3. This simulation models how machines can optimize their policies directly using gradient descent in policy space to minimize their cost.