Cooperating with Machines

Jacob W. Crandall*[1], Mayada Oudah[1], Tennom[1], Fatimah Ishowo-Oloko[1], Sherief Abdallah[2,3], Jean-François Bonnefon[4], Manuel Cebrian[5], Azim Shariff[6], Michael A. Goodrich[7], and Iyad Rahwan*[8]

[1]*Masdar Institute of Science and Technology, Abu Dhabi, UAE*
[2]*British University in Dubai, Dubai, UAE*
[3]*School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK*
[4]*Toulouse School of Economics (Center for Research in Management), Toulouse, France*
[5]*NICTA, Melbourne, Australia*
[6]*Department of Psychology, University of Oregon, Eugene, OR 97403, USA*
[7]*Computer Science Department, Brigham Young University, Provo, UT 84602, USA*
[8]*The Media Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

August 24, 2015

Abstract

Since Alan Turing envisioned Artificial Intelligence (AI) [1], a major driving force behind technical progress has been competition with human cognition. Historical milestones have been frequently associated with computers matching or outperforming humans in difficult cognitive tasks (e.g. face recognition [2], personality classification [3], driving cars [4], or playing video games [5]), or defeating humans in strategic zero-sum encounters (e.g. Chess [6], Checkers [7], Jeopardy! [8], or Poker [9]). In contrast, less attention has been given to developing autonomous machines that establish mutually cooperative relationships with humans even when the self-regarding preferences of humans and machines are in conflict, but are not fully opposed. A main challenge has been that human cooperation does not require sheer computational power, but rather relies on intuition [10], cultural norms [11], emotions and signals [13], and pre-evolved dispositions toward cooperation [12], common-sense mechanisms that are difficult to encode in machines. Here, we combine a state-of-the-art machine-learning algorithm with novel mechanisms for generating and acting on signals to produce a new learning algorithm that cooperates with people and other machines at levels that rival human cooperation in two-player repeated games. This is the first general-purpose algorithm that is capable, given a description of a previously unseen game environment, of learning human-level cooperation within short timescales. It does so without pre-programming of well-known, game-specific strategies (e.g. tit-for-tat), thus enabling human-AI cooperation in scenarios previously unanticipated by algorithm designers.

*Correspondence should be addressed to `jcrandall@masdar.ac.ae` and `irahwan@mit.edu`

Cooperation is among the fundamental drivers of human civilization and culture [14]. Though cooperation often requires people to refrain from self-serving behaviors, it can still emerge in interactions between self-interested individuals [15, 16, 17]. As autonomous machines increasingly permeate our lives, a new human-machine society is emerging in which algorithms make increasingly autonomous decisions that maximize the machine's objectives. While there have been many depictions of artificial intelligence (AI) subjecting humanity in such future societies, there is reason to hope and design for a more favorable outcome in which machines with self-regarding preferences choose to cooperate, rather than compete, with people. To facilitate cooperation in human-machine societies, self-serving machines must employ algorithms that, like humans, learn to cooperate even when defection promises to produce higher short-term payoffs.

Unfortunately, algorithms capable of establishing cooperative relationships with people in arbitrary scenarios are not easy to come by. A successful algorithm must satisfy several conditions. First, it must not be domain-specific – it must have superior performance in a wide variety of scenarios. Second, the algorithm must learn to establish effective relationships with people it has never met before. To do this, it must be able to deter potentially exploitative behavior from its partner and, when beneficial, determine how to elicit cooperation from a (potentially distrustful) partner who might be disinclined to cooperate. Third, when associating with people, the algorithm must learn effective behavior within very short timescales – i.e., within only a few rounds of experience. These requirements create many technical challenges for machines (see SI: Appendix A), the sum of which often causes AI algorithms to defect even when cooperation would be beneficial in the long run.

In addition to algorithmic challenges related to computing strategies that elicit cooperation, human-AI cooperation is difficult due to differences in the way that humans and machines reason. Human cooperation does not appear to require sheer computational power, but rather relies on intuition [10], cultural norms [11], emotions and signals [13], and pre-evolved dispositions toward cooperation [12]. On the other hand, AI relies on computation coupled with random exploration [18] to investigate the potential benefits of various strategies. However, random exploration is likely to breed distrust in a human partner, thus leading to dysfunctional, non-cooperative, relationships.

Evaluating the state-of-the-art. Despite these challenges, our goal is to develop AI algorithms that learn to cooperate with people as well as people do in arbitrary repeated games. As a first step, we evaluated existing AI algorithms in hopes of identifying algorithms that may potentially be able to learn to cooperate with people. Pioneered by the work of Littman [19], research on online-learning algorithms for repeated games has produced increasingly sophisticated and powerful algorithms. We conducted Axelrod-style tournaments [20] in which we paired 19 representative, state-of-the-art AI algorithms (plus RANDOM) together across 390 different games. Representative results are given in Figure 1 (see SI for more details). The algorithms S++ [21] and BULLY [22] (an algorithm that produces zero-determinant strategies [23]) emerged as the top-two algorithms in round-robin tournaments (Figure 1). Notably, *generalized tit-for-tat* was one of the less successful algorithms, which highlights the competitiveness of modern AI learning algorithms.

While BULLY is able to exploit some algorithms, it performs poorly against others (including itself) due to its inability to consistently cooperate (Figure 1a). This limits its evolutionary robustness [24]. S++, on the other hand, has high performance against each algorithm. It gained the highest population share when we investigated evolutionary dynamics [25, 26] independent of the initial population and interaction length (Figure 1b). More importantly, unlike many other machine-learning algorithms, S++ tends to learn effective behavior within relatively few rounds (Figure 1c), fast enough to support interactions with people.

S++ (Figure 2; see SI for more details) combines and builds on decades of research in computer science, economics, and the diverse array of behavioral and social sciences. It uses the description of the game environment to compute a diverse set of experts. These experts include customized behavioral rules and learning algorithms, each of which uses distinct mathematics to produce a strategy over the entire space of the game. It then uses a meta-level control strategy based on aspiration learning [27, 28, 29] to dynamically select which expert to follow at any given time. Formally, let E denote the set of experts computed by S++. In each epoch (beginning in round t), S++ computes the potential $\rho_j(t)$ of each expert $e_j \in E$, and compares this potential with its aspiration level $\alpha(t)$ to form a reduced set $E(t)$ of experts:

$$E(t) = \{e_j \in E : \rho_j(t) \geq \alpha(t)\}. \tag{1}$$

This reduced set consists of the experts that S++ believes could potentially produce satisfactory payoffs. It then selects one expert $e_{\text{sel}}(t) \in E(t)$ using a satisficing decision rule [28, 29]. Over the next m rounds, it follows the strategy prescribed by $e_{\text{sel}}(t)$, after which it updates its aspiration level as follows:

$$\alpha(t+m) \leftarrow \lambda^m \alpha(t) + (1 - \lambda^m) R, \tag{2}$$

where $\lambda \in (0, 1)$ is the learning rate and R is the average payoff obtained by S++ in the last m rounds. It also updates each expert $e_j \in E$ based on its peculiar reasoning mechanism, and then begins a new epoch.

Because S++ uses a set of experts, each of which can be arbitrarily sophisticated, it can produce rich and complex strategies within relatively short timescales by selecting the ‘right’ expert for the job. In this way, it can learn widely different strategies depending on its partner’s behavior (Extended Data: Figure 6). This mechanism also allows it to be used in sophisticated game structures, such as repeated stochastic games [30].

Pairing algorithms with people. To evaluate the ability of S++ to establish cooperative relationships with people, we conducted two user studies in which we paired people with S++ in four different repeated normal-form games and two different stochastic games (Extended Data: Figure 7), respectively. As points of comparison, we also paired people with MBRL-1 (a model-based reinforcement-learning algorithm) in normal form games, and CFR (the algorithm used in championship Computer Poker algorithms [9]) in stochastic games. In both studies, S++ paired with S++ was the only partnership that established high levels of mutual cooperation (Figure 3a-b), thus demonstrating the ability of S++ to adapt to diverse and complex scenarios. Nevertheless, despite the willingness of S++ to cooperate, it was unable to consistently establish cooperative relationships with people. S++ and people cooperated with each other less than 30% of the time, statistically equivalent to what was achieved in human-human pairings. This inability to cooperate resulted in substantially lower payoffs to both players than if they had both always cooperated.

One reason for this relative lack of cooperation may be that players could not communicate throughout the game. We found support for this idea in a second condition of the second user study, in which players were allowed to communicate face-to-face throughout the game. This opportunity to communicate resulted in substantial increases in cooperation between humans (Figure 3b).

An algorithm that generates and responds to signals. If full and consistent cooperation with people requires communication, machines must be endowed with the capacity to convey and act on signals about their intentions and expectations. While signaling comes naturally to humans, the same cannot be said of machines. Most machine-learning algorithms have internal representations that are difficult for us to understand. As such, it is not obvious how these algorithms can be made to act on and convey the kinds of signals that would be effective in interactions with people.

However, unlike typical machine-learning algorithms, the internal structure of S++ provides a clear, high-level representation of the algorithm’s dynamic strategy, which can be described in terms of the dynamics of the underlying experts. Since many of the experts encode high-level ideas (e.g., leader strategies [22]), we can use S++ to generate speech acts that describe its intentionality and use speech acts from its partner to augment its expert-selection mechanism. The resulting new algorithm, dubbed S# (pronounced ‘S sharp’) is depicted in Figure 4 (see SI for details).

S# differs from S++ in two ways. First, the partner's proposed plans, signaled via speech acts, are used to further reduce the set of experts that S# considers selecting (Figure 4-top). Formally, let $E_{\text{cong}}(t)$ denote the set of experts in round t that are *congruent* with the last joint plan proposed by S#'s partner (see SI for how congruence is computed). Then, S# considers selecting experts from the following set:

$$E(t) = \{e_j \in E_{\text{cong}}(t) : \rho_j(t) \geq \alpha(t)\}. \tag{3}$$

If this set is empty (i.e., no desirable options are congruent with the partner's proposal), $E(t)$ is calculated as in Eq. (1). Second, S# also extends S++ by generating speech acts that convey the "stream of conscience" of the algorithm. Specifically, a finite-state machine with output is generated automatically for each expert (Figure 4-bottom). Given the current state of the expert and the game outcomes, the state machine produces speech derived from a pre-determined set of phrases. With the exception of action labels provided in the game description, these speech acts are game-generic, though some adaptations must be made for general stochastic games (see SI).

To evaluate the ability of S# to consistently cooperate with people, we conducted a third, 66-participant, user study in which S# and people interacted in three normal-form games (Prisoner's Dilemma, Chicken, and the Alternator Game). The participants were divided into two categories: those that could and could not communicate. When communication was permitted, players could send messages (limited to the pre-determined speech acts available to S#) to their partner at the beginning of each round. The results of the study are summarized in Figure 5. When communication was not allowed, the results were similar to previous studies. However, when communication was possible, mutual cooperation in human-human and human-S# pairings doubled. Overall, pairings of two S# players produced the highest levels of mutual cooperation. Together, these results demonstrate the effectiveness of S#'s combined learning and speech systems. S# consistently formed cooperative relationships with people and other machines that followed S#.

Potential implications. Our results open the opportunity to study human-machine cooperation in a way that builds on the rich literature on human cooperation in behavioral economics and evolutionary biology [31]. In particular, as machines become increasingly autonomous, we see human-machine cooperation as a phenomenon that emerges from the interaction among self-interested parties [32], rather than by issuing explicit design constraints that are programmed into domain-specific computational systems [33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]. Our results demonstrate that emergent cooperation between humans and autonomous, self-regarding machines is now feasible.

While AI algorithms are gaining the ability to consistently learn to cooperate among themselves even without communication, human cooperation appears to depend on signaling. In light of these findings, greater effort must be placed on developing autonomous machines that can effectively communicate with people, and vice versa, in order to create shared representations [44, 45, 46, 40] and coordinate quickly on desirable equilibria. Our work emphasizes the importance of integrating signaling into decision-making algorithms, and presents case studies that others can work from to continue to make progress [45].

Since Alan Turing argued that machines could potentially demonstrate intelligence, AI has been regularly portrayed as a threat to humanity, paving the way to severe disruption of labor markets [47], or machine-dominated dystopias. The fear of enslavement by machines was articulated in 1922 by British physician Havelock Ellis: "The greatest task before civilization at present is to make machines what they ought to be, the slaves, instead of the masters of men" [48]. Most attempts to curb this threat have followed the path of hardcoding legal or moral principles into computer code, such as Asimov's Three Laws of Robotics [49], itself an unsolved problem [50, 51]. Our research demonstrates that a new path is possible: machines designed to selfishly maximize their payoffs can, and will, make an autonomous choice to cooperate with humans.

References

[1] A. M. Turing. Computing machinery and intelligence. *Mind*, pages 433–460, 1950.

[2] A. J. Toole, P. J. Phillips, F. Jiang, J. Ayyad, N. Penard, and H. Abdi. Face recognition algorithms surpass humans matching faces over changes in illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1642–1646, 2007.

[3] W. Youyou, M. Kosinski, and D. Stillwell. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, 112(4):1036–1040, 2015.

[4] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The Stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[6] M. Campbell, A. J. Hoane, and F. Hsu. Deep blue. *Artificial intelligence*, 134(1):57–83, 2002.

[7] J. Schaeffer, N. Burch, Y. Björnsson, A. Kishimoto, M. Müller, R. Lake, P. Lu, and S. Sutphen. Checkers is solved. *Science*, 317(5844):1518–1522, 2007.

[8] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79, 2010.

[9] M. Bowling, N. Burch, M. Johanson, and O. Tammelin. Heads-up limit holdem poker is solved. *Science*, 347(6218):145–149, 2015.

[10] D. G. Rand, A. Peysakhovich, G. T. Kraft-Todd, G. E. Newman, O. Wurzbacher, M. A. Nowak, and J. D. Greene. Social heuristics shape intuitive cooperation. *Nature Communications*, 5, 2014.

[11] R. Boyd and P. J. Richerson. Culture and the evolution of human cooperation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1533):3281–3288, 2009.

[12] A. Peysakhovich, M. A Nowak, and D. G. Rand. Humans display a cooperative phenotype that is domain general and temporally stable. *Nature Communications*, 5, 2014.

[13] R. H. Frank. *Passions Within Reason: The Strategic Role of the Emotions.* W. W. Norton & Company, 1988.

[14] M. Tomasello. *A Natural History of Human Thinking*. Harvard University Press, 2014.

[15] B. Skyrms. *The Stag Hunt and the Evolution of Social Structure*. Cambridge Press, 2003.

[16] M. A. Nowak. Five rules for the evolution of cooperation. *Science*, 314(5805):1560–1563, 2006.

[17] S. Bowles and H. Gintis. *A cooperative species: Human reciprocity and its evolution*. Princeton University Press, 2011.

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–277, 1996.

[19] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163, 1994.

[20] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[21] J. W. Crandall. Towards minimizing disappointment in repeated games. *Journal of Artificial Intelligence Research*, 49:111–142, 2014.

[22] M. L. Littman and P. Stone. Leading best-response strategies in repeated games. In *IJCAI workshop on Economic Agents, Models, and Mechanisms*, Seattle, WA, 2001.

[23] W. H. Press and F. J. Dyson. Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413, 2012.

[24] C. Adami and A. Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature Communications*, 4, 2013.

[25] P. D. Taylor and L. Jonker. Evolutionarily stable strategies and game dynamics. *Mathematical Biosciences*, 40:145–156, 1978.

[26] M. A. Nowak. *Evolutionary dynamics*. Harvard University Press, 2006.

[27] H. A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[28] R. Karandikar, D. Mookherjee, D. R., and F. Vega-Redondo. Evolving aspirations and cooperation. *Journal of Economic Theory*, 80:292–331, 1998.

[29] J. R. Stimpson, M. A. Goodrich, and L. C. Walters. Satisficing and learning cooperation in the prisoner's dilemma. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 535–544, 2001.

[30] J. W. Crandall. Robust learning in repeated stochastic games using meta-gaming. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2015. To appear.

[31] D. G. Rand and M. A. Nowak. Human cooperation. *Trends in Cognitive Sciences*, 17(8):413–425, 2013.

[32] R. Lin and S. Kraus. Can automated agents proficiently negotiate with humans? *Communications of the ACM*, 53(1):78–88, 2010.

[33] N. R. Jennings, L. Moreau, D. Nicholson, S. Ramchurn, S. Roberts, T. Rodden, and A. Rogers. Human-agent collectives. *Communications of the ACM*, 57(12):80–88, 2014.

[34] I. R. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion. Human-robot teaming for search and rescue. *Pervasive Computing, IEEE*, 4(1):72–79, 2005.

[35] A. Azaria, Y. Gal, S. Kraus, and C. V. Goldman. Strategic advice provision in repeated human-agent interactions. *Autonomous Agents and Multi-Agent Systems*, pages 1–26, 2015.

[36] A. Azaria, Z. Rabinovich, C. V. Goldman, and S. Kraus. Strategic information disclosure to people with multiple alternatives. *ACM Transactions on Intelligent Systems and Technology*, 5(4):64, 2014.

[37] P. Scerri, D. V. Pynadath, and M. Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17(1):171–228, 2002.

[38] N. B. Sarter and D. D. Woods. Team play with a powerful and independent agent: A full-mission simulation study. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 42(3):390–402, 2000.

[39] E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 467–474, 2012.

[40] E. Kamar, Y. Gal, and B. J. Grosz. Modeling information exchange opportunities for effective human–computer teamwork. *Artificial Intelligence*, 195:528–550, 2013.

[41] Y. Gal, B. J. Grosz, S. Kraus, A. Pfeffer, and S. Shieber. Agent decision-making in open mixed networks. *Artificial Intelligence*, 174(18):1460–1480, 2010.

[42] T. W. Bickmore and R. W. Picard. Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction*, 12(2):293–327, 2005.

[43] C. Sidner, C. Lee, C. D. Kidd, N. Lesh, and C. Rich. Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1):140–164, 2005.

[44] G. Klein, P. J. Feltovich, J. M. Bradshaw, and D. D. Woods. Common ground and coordination in joint activity. *Organizational simulation*, 53, 2005.

[45] K. Dautenhahn. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):679–704, 2007.

[46] C. Breazeal. Toward sociable robots. *Robotics and autonomous systems*, 42(3):167–175, 2003.

[47] E. Brynjolfsson and A. McAfee. *The second machine age: work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company, 2014.

[48] H. Ellis. *Little essays of love and virtue*. Black, 1922.

[49] I. Asimov. *I, robot*, volume 1. Spectra, 2004.

[50] W. Wallach and C. Allen. *Moral machines: Teaching robots right from wrong*. Oxford University Press, 2008.

[51] P. Lin, K. Abney, and G. A. Bekey. *Robot ethics: the ethical and social implications of robotics*. MIT press, 2011.

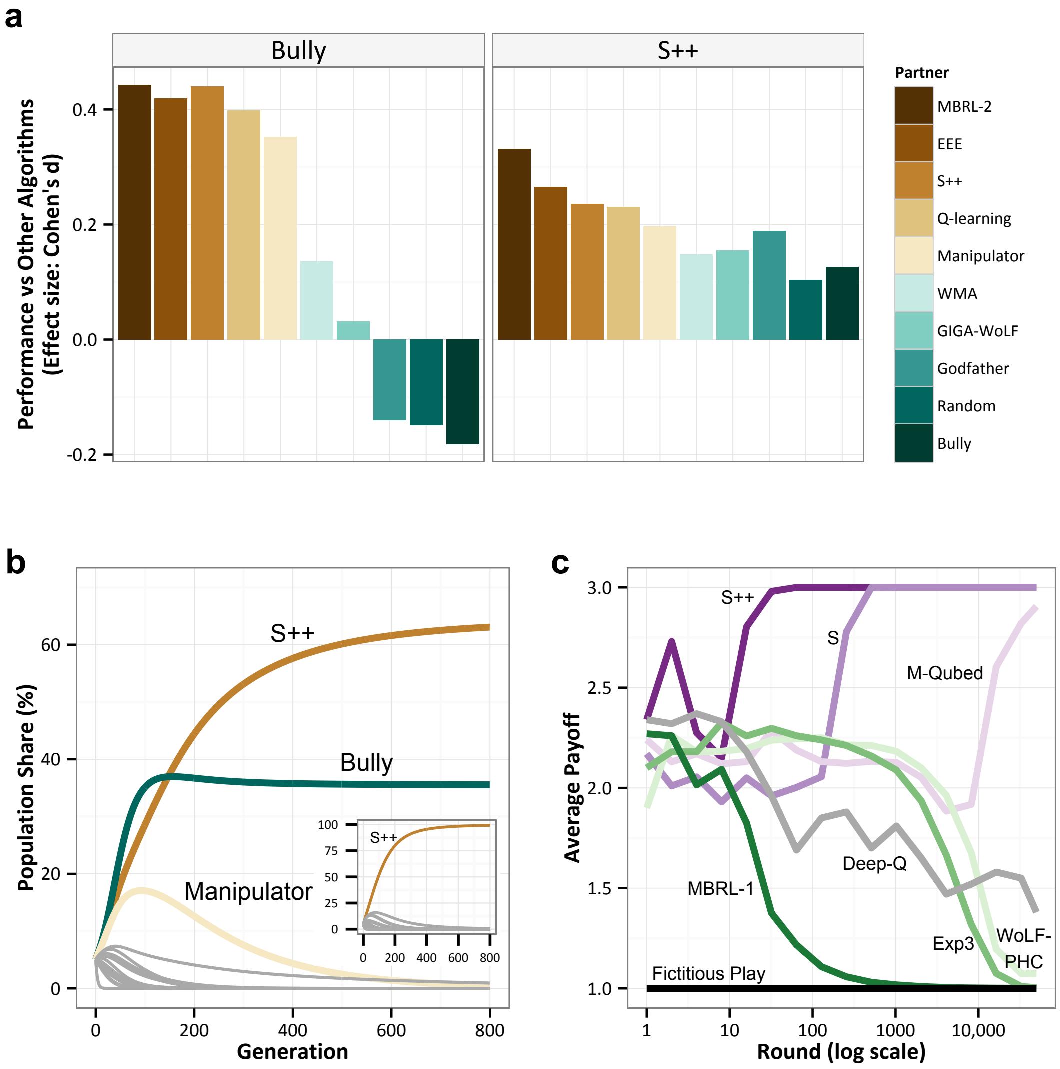


Figure 1: Selected results from an empirical evaluation of algorithms, in which 19 representative algorithms, plus Random, were paired in 390 different two-player, two-action repeated games. S++ [21] and Bully [22] (an algorithm that produces zero-determinant strategies [23]) were the top-performing algorithms in the round-robin tournaments. **(a)** Shows the relative payoffs of S++ and Bully compared with the average payoffs of other algorithms (in terms of effect size: Cohen's *d*) against a variety of algorithms in 50,000-round games. While Bully is very effective against many algorithms, it performs poorly against others. S++ substantially outperforms the average in all pairings. **(b)** Population share over time in 50,000-round games (inset shows 100-round games). S++ gained the highest population share regardless of initial population or interaction length. Bully is unable to gain a majority due to its inability to cooperate with itself and like-minded associates. **(c)** Average payoffs over time of eight different machine-learning algorithms in self-play in a standard (0-1-3-5)-prisoner's dilemma. Results are an average of 50 trials each. Some of the algorithms learn to cooperate (the purple-shaded curves), while others learn to defect (green/grey-shaded curves). Only S++ learns to cooperate within timescales that support interactions with people.

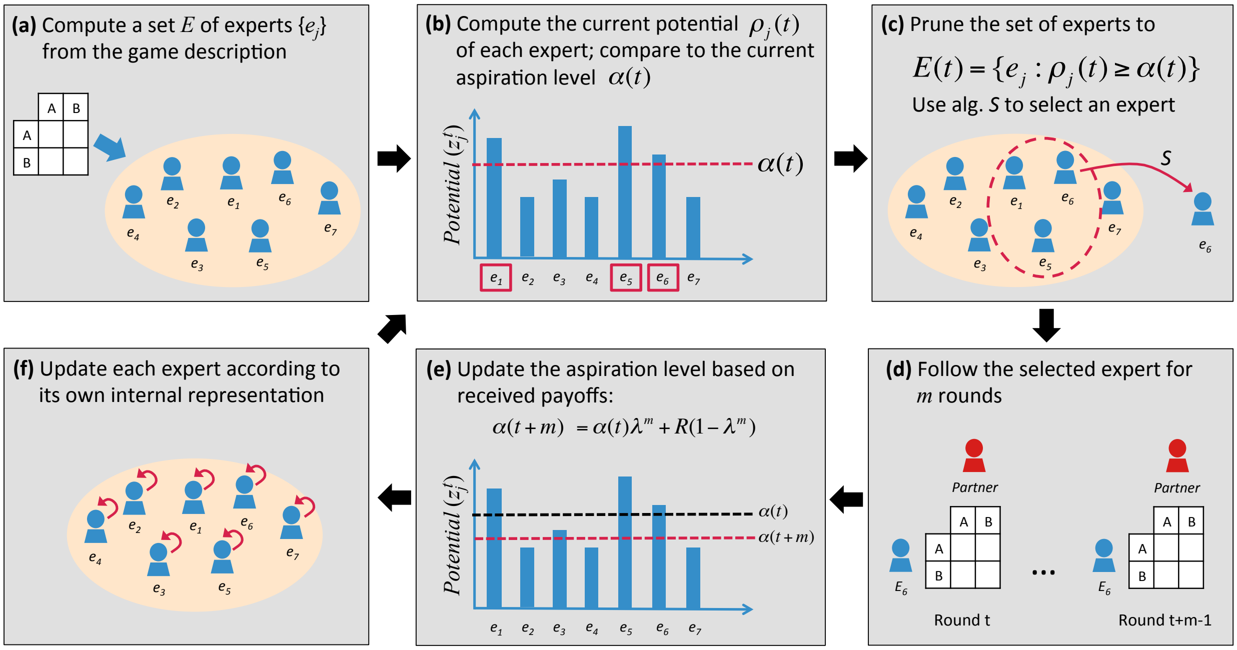

(a) Compute a set E of experts {e_j} from the game description
A
B
e_1 e_2 e_3 e_4 e_5 e_6 e_7
(b) Compute the current potential ρ_j(t) of each expert; compare to the current aspiration level α(t)
Potential (z_j^t)
α(t)
e_1 e_2 e_3 e_4 e_5 e_6 e_7
(c) Prune the set of experts to
E(t) = {e_j : ρ_j(t) ≥ α(t)}
Use alg. S to select an expert
S
e_1 e_2 e_3 e_4 e_5 e_6 e_7
(d) Follow the selected expert for m rounds
Partner
E_6
Round t
...
Round t+m-1
(e) Update the aspiration level based on received payoffs:
α(t+m) = α(t)λ^m + R(1 − λ^m)
Potential (z_j^t)
α(t)
α(t+m)
e_1 e_2 e_3 e_4 e_5 e_6 e_7
(f) Update each expert according to its own internal representation
e_1 e_2 e_3 e_4 e_5 e_6 e_7

Figure 2: An overview of S++. **(a)** Prior to beginning the game, S++ uses the description of the game to compute a set E of expert strategies. Each expert encodes a strategy or learning algorithm defining a behavior over all game states. **(b)** After computing the experts, S++ computes the potential, or highest expected utility, of each expert. The potentials are then compared to an aspiration level $\alpha(t)$, which encodes the average per-round payoff that the algorithm believes is achievable. **(c)** The aspiration level and potentials are used to eliminate experts that are unlikely to produce satisfactory payoffs. S++ then selects an expert from among the remaining experts using algorithm S [28, 29]. **(d)** The machine follows the strategy dictated by the selected expert for m rounds of the repeated game. **(e)** The machine updates its aspiration level based on the average reward R it has received over the last m rounds of the game. **(f)** The experts are each updated according to their own internal representations. The algorithm then returns to step (b) and repeats the process for the duration of the game.

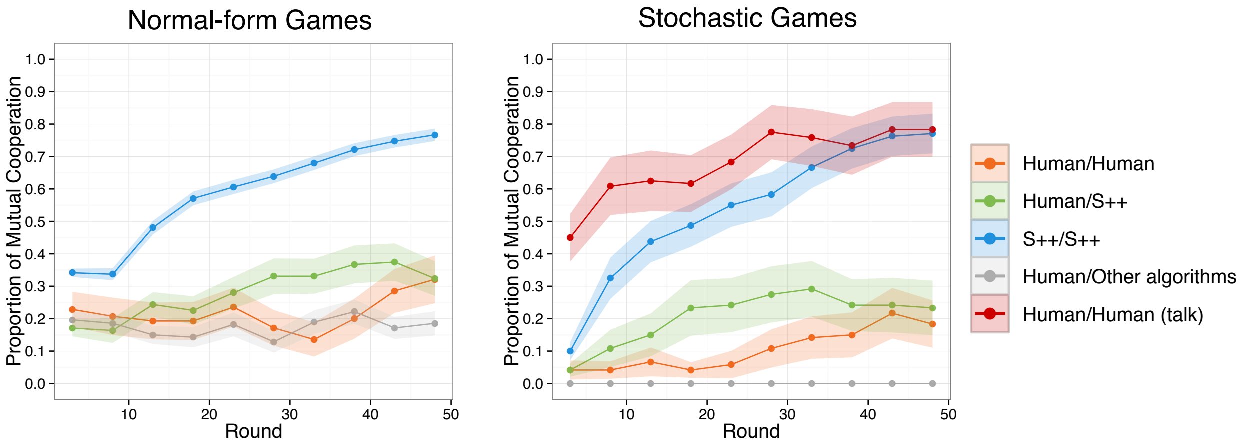
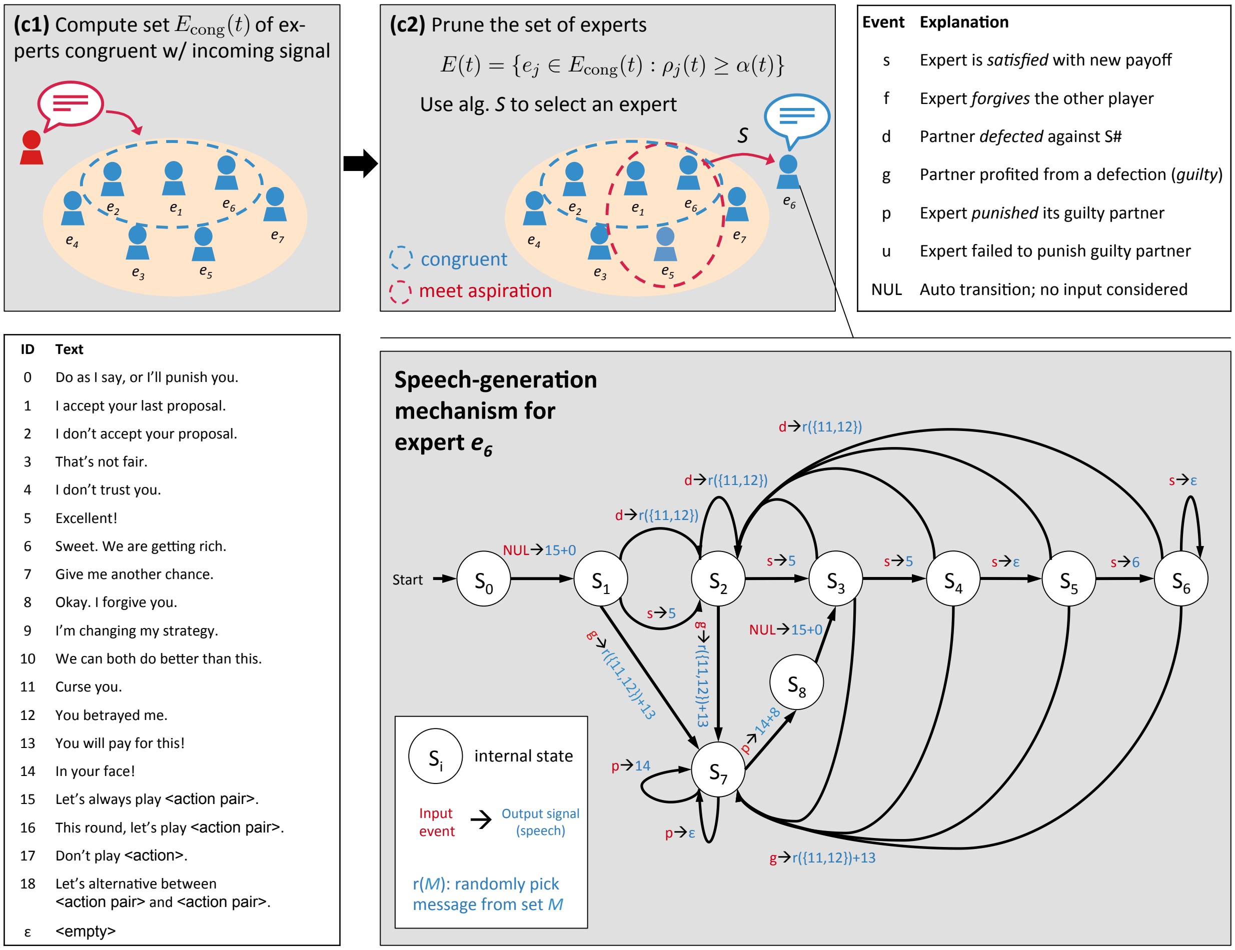


Figure 3: Proportion of mutual cooperation over time **(a)** in four normal-form games (Prisoner's Dilemma, Chicken, Chaos, and Shapley's Game) as observed in a 58-participant user study, and **(b)** in two stochastic games (SGPD and The Block Game; see Extended Data: Figure 7) as observed in a 96-participant user study. Error bands show the standard error on the mean. S++ paired with S++ was the only partnership that consistently learned to cooperate in both sets of games in the absence of the ability to communicate. However, like the other computer algorithms that we tested (MBRL-1 in normal-form games and CFR in stochastic games), S++ was unable to consistently elicit cooperative behavior from people. People also failed to consistently cooperate with each other when they could not communicate. However, in a second condition of the second study, people were allowed to talk freely (face-to-face) throughout the games, which resulted in substantially higher levels of cooperation.


(c1) Compute set $E_{\text{cong}}(t)$ of experts congruent w/ incoming signal
e_1
e_2
e_3
e_4
e_5
e_6
e_7
(c2) Prune the set of experts
$E(t) = \{e_j \in E_{\text{cong}}(t) : \rho_j(t) \geq \alpha(t)\}$
Use alg. *S* to select an expert
S
congruent
meet aspiration
Speech-generation mechanism for expert e_6
Start
S_0
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
NUL→15+0
d→r({11,12})
d→r({11,12})
d→r({11,12})
s→5
s→5
s→5
s→ε
s→6
s→ε
g→r({11,12})+13
g→r({11,12})+13
NUL→15+0
p→14+8
p→14
p→ε
g→r({11,12})+13
S_i internal state
Input event → Output signal (speech)
r(*M*): randomly pick message from set *M*

Event	Explanation
s	Expert is *satisfied* with new payoff
f	Expert *forgives* the other player
d	Partner *defected* against S#
g	Partner profited from a defection (*guilty*)
p	Expert *punished* its guilty partner
u	Expert failed to punish guilty partner
NUL	Auto transition; no input considered

ID	Text
0	Do as I say, or I'll punish you.
1	I accept your last proposal.
2	I don't accept your proposal.
3	That's not fair.
4	I don't trust you.
5	Excellent!
6	Sweet. We are getting rich.
7	Give me another chance.
8	Okay. I forgive you.
9	I'm changing my strategy.
10	We can both do better than this.
11	Curse you.
12	You betrayed me.
13	You will pay for this!
14	In your face!
15	Let's always play <action pair>.
16	This round, let's play <action pair>.
17	Don't play <action>.
18	Let's alternative between <action pair> and <action pair>.
ε	<empty>

Figure 4: An overview of S#, an algorithm that interweaves signaling capabilities into S++. **(Top)** S# uses its partner's speech acts to further prune the set of experts it considers following, which is done by expanding step c into parts c1 and c2. First, S# determines which experts carry out plans congruent with its partners last proposed plan. Next, S# selects an expert from among those experts that are both congruent with its plan and meet its aspiration level. If this set is empty, S# selects its expert identically to S++. **(Bottom)** The currently selected expert generates signals based on its game-generic state machine. Given the current state of the expert and game events (top-right), the expert produces speech from a pre-determined list of speech acts (bottom-left).

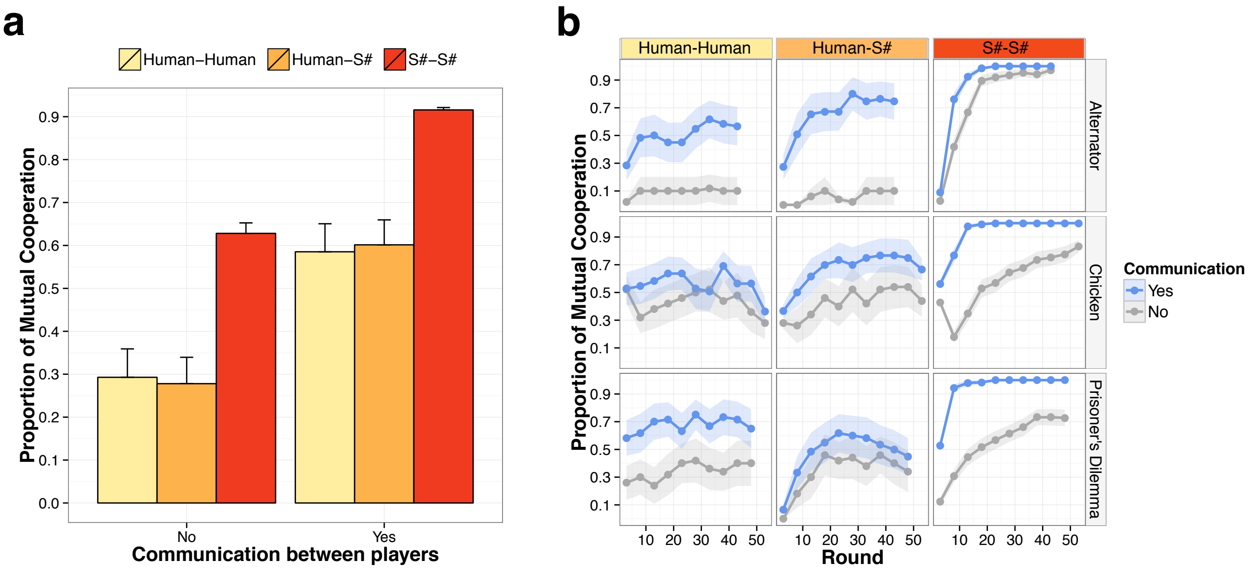


Figure 5: Results of a third, 66-participant user study in which people were paired with each other and S#. Note that S# is identical to S++ when communication is not possible. **(a)** The average proportion of mutual cooperation across all three games in each pairing under conditions in which players could and could not communicate with each other. Error bars show the standard error on the mean. **(b)** The average proportion of mutual cooperation over time in each game in each pairing and condition. S#-S# pairings produced (by far) the highest levels of mutual cooperation under both conditions, with communication providing substantial increases in cooperation. People did not consistently cooperate with each other in the absence of communication. However, the ability to communicate doubled the amount of mutual cooperation in both human-human and human-S# pairings.

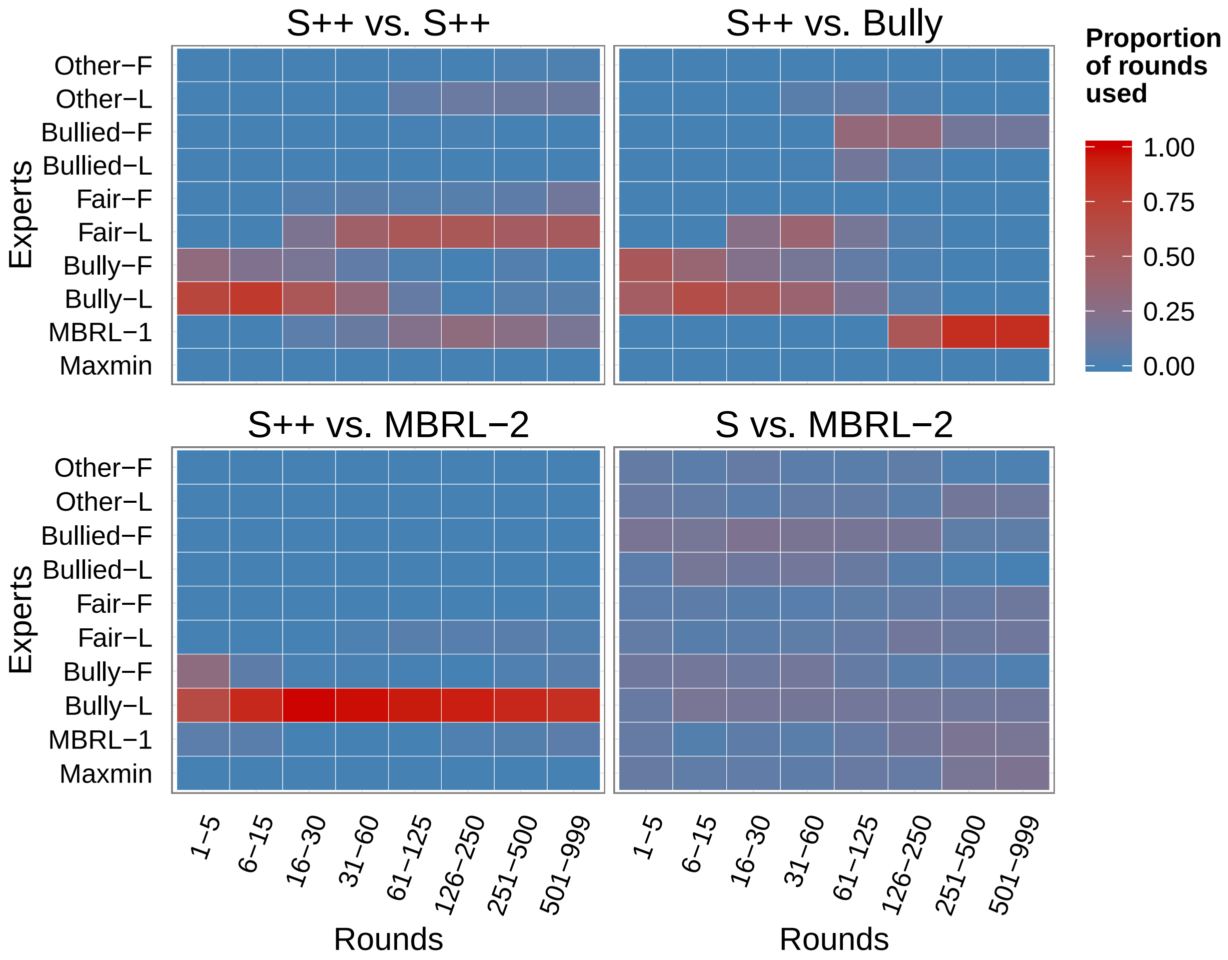
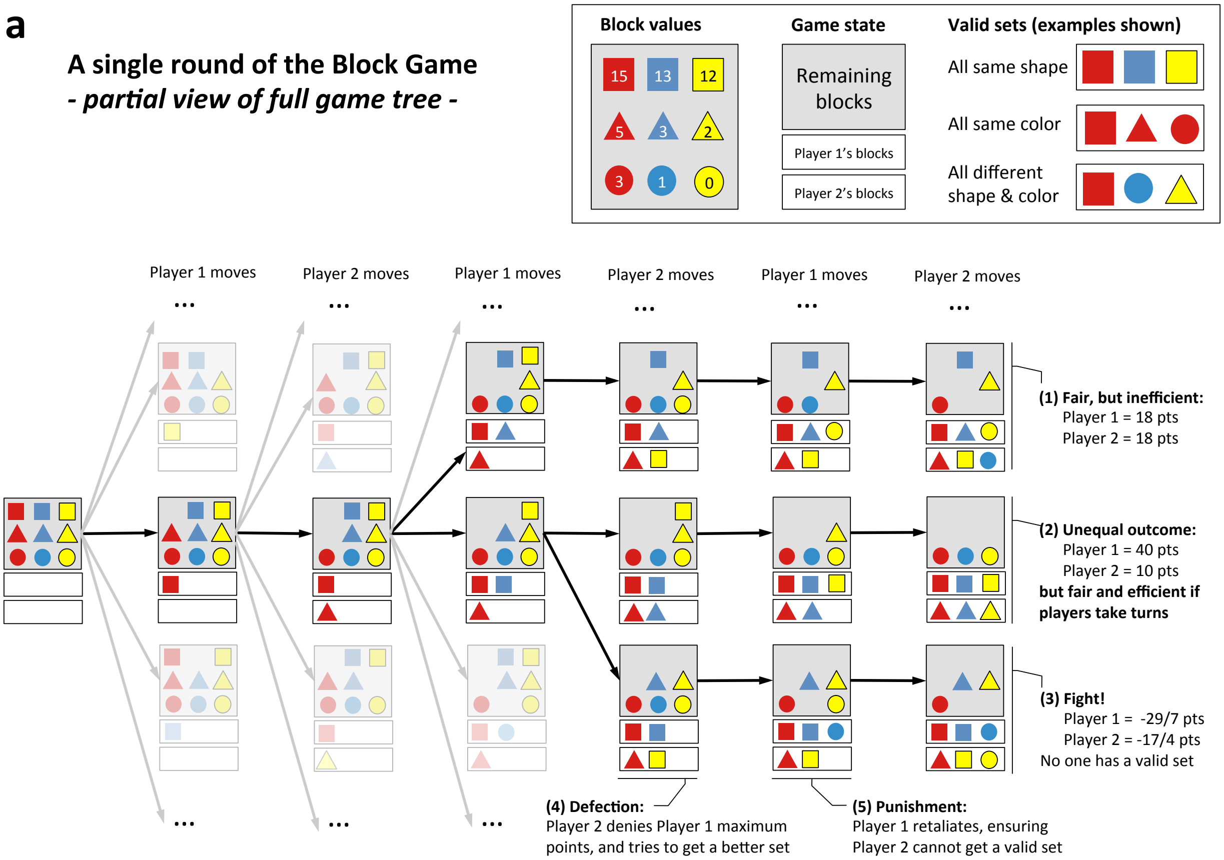
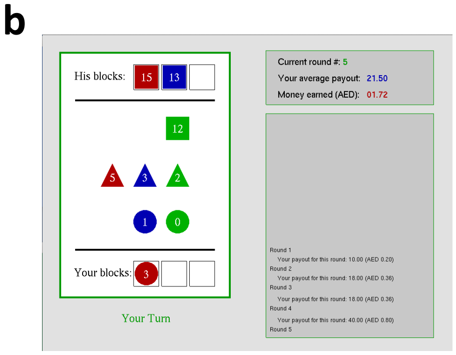


Figure 6: (Extended Data Figure) An illustration of the learning dynamics of S++ in Chicken. For ease of understanding, experts are categorized into groups (see SI). **(Top-left)** The proportion of time that S++ selects each group of experts over time when paired with a copy of itself. S++ initially seeks to bully its associate, but then switches to fair, cooperative experts when attempts to exploit are unsuccessful. **(Top-right)** When paired with BULLY, S++ learns the best response, which is to be bullied, achieved by playing MBRL-1, Bully-L, or Bully-F. **(Bottom-left)** S++ quickly learns to play experts that bully MBRL-2. **(Bottom-right)** On the other hand, algorithm S does not learn to consistently bully MBRL-2. The primary difference between S and S++ is that S++ uses the expert-pruning mechanism illustrated in steps 2 and 3 of Figure 2; S does not. This pruning rule allows S++ to focus on teaching MBRL-2 to accept being bullied, thus producing high payoffs for S++.





His blocks:

Your blocks:

Your Turn

Current round #: 5
Your average payout: 21.50
Money earned (AED): 01.72

Round 1
Your payout for this round: 10.00 (AED 0.20)
Round 2
Your payout for this round: 18.00 (AED 0.36)
Round 3
Your payout for this round: 18.00 (AED 0.36)
Round 4
Your payout for this round: 40.00 (AED 0.80)
Round 5

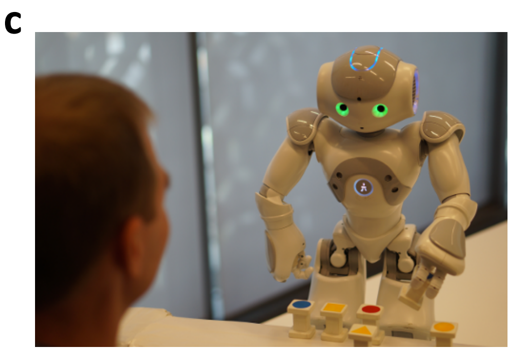


Figure 7: (Extended Data Figure) **(a)** An extensive-form game in which two players share a nine-piece block set. The two players take turns selecting blocks from the set until each has three blocks. The goal of each player is to get a *valid* set of blocks with the highest value possible, where the value of a set is determined by the sum of the numbers on the blocks. Invalid sets receive negative points. (1) A fair, but inefficient outcome in which both players receive 18 points. (2) An unequal outcome in which one player receives 40 points, while the other player receives just 10 points. However, when the players take turns getting the higher payoff (selecting all the squares), this is the Nash bargaining solution of the game, producing an average payoff of 25 to both players. (3) An outcome in which neither player obtains a valid set, and hence both players lose points. (4) This particular negative outcome is brought about when player 2 defects against player 1 by taking the block that player 1 needs to complete its (most-valuable) set. (5) Player 1 then retaliates to ensure that player 2 does not get a valid set either. **(b)** In a second user study, participants played the Block Game with other people and with computer algorithms via a graphical user interface. **(c)** We also implemented S++ on a Nao robot to play the Block Game with people.