Properties of Winning Iterated Prisoner's Dilemma Strategies.

Nikoleta E. Glynatsi[1, 2], Vincent Knight[2, †], and Marc Harper[3, †]

[1]Max Planck Institute for Evolutionary Biology, Research Group Dynamics of Social Behavior, Germany
[2]Cardiff University, School of Mathematics, UK
[3]Google Inc., Mountain View, CA, USA
[†]V.K. and M.H. contributed equally to this work.

Abstract

Researchers have explored the performance of Iterated Prisoner's Dilemma strategies for decades, from the celebrated performance of Tit for Tat to the introduction of the zero-determinant strategies and the use of sophisticated learning structures such as neural networks. Many new strategies have been introduced and tested in a variety of tournaments and population dynamics. Typical results in the literature, however, rely on performance against a small number of somewhat arbitrarily selected strategies in a small number of tournaments, casting doubt on the generalizability of conclusions. In this work, we analyze a large collection of 195 strategies in thousands of computer tournaments, present the top performing strategies across multiple tournament types, and distill their salient features. The results show that there is not yet a single strategy that performs well in diverse Iterated Prisoner's Dilemma scenarios, nevertheless there are several properties that heavily influence the best performing strategies. This refines the properties described by Axelrod in light of recent and more diverse opponent populations to: be nice, be provocable and generous, be a little envious, be clever, and adapt to the environment. More precisely, we find that strategies perform best when their probability of cooperation matches the total tournament population's aggregate cooperation probabilities. The features of high performing strategies help cast some light on why strategies such as Tit For Tat performed historically well in tournaments and why zero-determinant strategies typically do not fare well in tournament settings. Furthermore, our findings have implications for the future training of autonomous agents, as understanding the crucial features for incorporation into these agents becomes essential.

1 Background

The Iterated Prisoner's Dilemma (`IPD`) is a repeated two-player game that models behavioral interactions, specifically interactions where self-interest clashes with collective interest. In each turn of the game, both players simultaneously and independently decide between cooperation (C) and defection (D). This decision is made with the memory of all prior interactions. The payoffs for each player at each turn are influenced by their own choice and the choice of the other player. To this end, the payoffs of the game are defined by

	Cooperate (C)	Defect (D)
Cooperate (C)	R, R	S, T
Defect (D)	T, S	P, P

where typically $T > R > P > S$ and $2R > T + S$. The most common values used in the literature [1] are $R = 3$, $P = 1$, $T = 5$, $S = 0$, and these are the values also used in this work.

Conceptualizing strategies and understanding the best way to play the game have been of interest to the scientific community since the formulation of the game in 1950 [2]. Following Axelrod's computer tournaments in the 1980s [3, 4], round-robin computer tournaments became a common evaluation technique for newly designed strategies. The winner of both of Axelrod's tournaments [3, 4] was the simple strategy Tit For Tat (`TFT`). `TFT` cooperates on the first turn and thereafter copies the previous action of its opponent, retaliating against defections with a defection and forgiving a defection if followed by cooperation. Axelrod concluded that the strategy's robustness was due to four properties, which he adapted into four suggestions for success in an `IPD` tournament:

(i) Do not be envious by striving for a payoff larger than the opponent's payoff.

(ii) Be "nice" by not being the first to defect.

(iii) Reciprocate both cooperation and defection; Be provocable to retaliation and forgiveness.

(iv) Do not be too clever by scheming to exploit the opponent.

Forgiveness, in this context, is a strategy's ability to cooperate after a DC outcome to achieve mutual cooperation again. In environments without noise, `TFT` would end up in DC only if it had received a defection and then retaliated. Subsequently, `TFT` would forgive an opponent that apologizes (in a DC round) by returning to cooperation, as mutual cooperation is deemed better than mutual defection.

Due to the strategy's strong performance in both tournaments and a series of evolutionary experiments [1], `TFT` was often claimed to be a highly robust (and sometimes the most robust) strategy for the `IPD`. There are strategies that have built upon `TFT` and the reciprocity-based approach. In [5], a strategy called Gradual was introduced, constructed to have the same qualities as those of `TFT` with one addition. Gradual has a memory of the previous rounds of play in the game, recording the number of defections by the opponent and punishing them with a growing number of defections. It then enters a calming state in which it cooperates for two rounds. A strategy with the same intuition as Gradual is Adaptive Tit for Tat [6]. Adaptive Tit for Tat maintains a continually updated estimate of the opponent's behavior and uses this estimate to condition its future actions. Other research has built upon the limitations of `TFT`. For example, in [7, 8, 9, 10], it was shown that `TFT` suffered in environments with noise. This was mainly due to the strategy being too provocable and its lack of generosity and contrition. Since `TFT` immediately punishes a defection, in a noisy environment, it can get stuck in a repeated cycle of defections and cooperations. Some new strategies, more robust in tournaments with noise, were soon introduced, including Nice and Forgiving [7], Generous Tit For Tat [11], and Pavlov (aka Win Stay Lose Shift) [12], as well as later variants such as OmegaTFT [13].

Finally, others introduced strategies deviating completely from the originally suggested properties of success. For example, a set of "envious" Iterated Prisoner's Dilemma (`IPD`) strategies were introduced, called zero-determinant strategies (ZDs), in [14]. These strategies attempt to force a linear relationship between stationary payoffs against other memory-one opponents, potentially ensuring that they receive a higher average payout. While ZDs were introduced with a small tournament in which some were reportedly successful [15], this result has not generally held in future work [16]. Furthermore, in [17], a series of "clever" strategies trained using reinforcement learning were introduced. These strategies were trained using lookup tables [18], hidden Markov models [17], and finite-state automata [19], on a set of 170 strategies.

One thing that has remained the same is that the introduction of a new strategy is often accompanied by a claim that the new strategy is the best performing strategy for the `IPD`, often without extensive testing against a broad spectrum of opponents or representative classes of opponents. The lack of testing against formally defined strategies and tournament winners is understandable given the effort required to implement the hundreds of published `IPD` strategies. Implementing prior strategies faithfully is often extremely difficult or impossible due to insufficient descriptions and lack of published implementations or code. Despite these challenges, the absence of thorough testing raises concerns about claims regarding the superiority or robustness of newly introduced strategies.

In this paper, we evaluate the performance of a significant number of `IPD` strategies across a diverse array of tournaments. Many of the strategies used in our analysis are drawn from well-known and named strategies in `IPD` literature, including previous tournament winners. This contrasts with other work that may have randomly generated essentially arbitrary strategies, often constrained to specific classes such as memory-one strategies or those of a certain structural form like finite state machines or deterministic memory-two strategies. Furthermore, our tournaments encompass variations, including standard tournaments resembling Axelrod's original ones, tournaments with noise, probabilistic match length, and both noise and probabilistic match length. This diversity in strategies and tournament types provides new insights and tests earlier claims in alternative settings against known powerful strategies. More specifically, we show that the previous tournament winners are lacking against large enough opponent pools; they do not appear among the top-performing strategies anymore. This could be due to likely suffering from a lack of diversity in the strategies they were trained/tested against, finding it hard to adapt to the new strategies.

It is important to note that we do not assert the existence of a single best-performing strategy across all tournaments or tournament types. On the contrary, our work demonstrates that such a strategy does not exist (notwithstanding a few strategies with broadly high performance). The primary objective of this paper, presented in the latter parts of the paper, is to continue the discussion on the properties of successful strategies, a conversation started by Axelrod. The results of our analysis conclude that the properties of a successful strategy in the Iterated Prisoner's Dilemma (`IPD`) are:

(i) Be a little bit envious

(ii) Be "nice" in non-noisy environments or when game lengths are longer

(iii) Reciprocate both cooperation and defection appropriately; Be provocable in tournaments with short matches, and generous in tournaments with noise

(iv) It's ok to be clever

(v) Adapt to the environment; Adjust to the mean population cooperation

We believe that the discussion on the properties of winning strategies holds significant importance. It aims to provide guidance to researchers designing new strategies and those training strategies. Specifically, much like the recognized value of diversity in training datasets, such as variations in image perspective, skin color, etc., are critical in training accurate and generalizable machine learning models, we show that diversity in the population of opponent strategies is of paramount importance in the construction and evaluation of game theory strategies. As AI agents start to interact as scale, they will benefit from exposure to a wide variety of alternative agents. Moreover, conducting a similar analysis can shed light on already trained strategies, aiding in understanding the key features they have autonomously developed during their training processes.

The rest of the paper is organized as follows. In section 2, we describe the data collection process. In subsection 3.1, we present the best performing strategies for each type of tournament. Subsection 3.2 explores the traits that contribute to good performance, and in subsection 3.3, we focus on the features of the winners of the tournaments. Finally, the results are summarized in section 4. This manuscript introduces several parameters, which are discussed in the following sections; the full set of parameters and their definitions are also provided in the Supplementary Material.

2 Data collection

The data collection of various types of tournaments and the use of different strategies are made possible due to an open-source library called `Axelrod-Python` [20] (version 3.0.0). `Axelrod-Python` enables the simulation of `IPD` tournaments and contains an extensive list of strategies. Most of these strategies are described in the literature, with a few exceptions contributed specifically to the package. In this paper, we use a total of 195 strategies, which can be found in the Supplementary Material. The package supports several tournament types, and this work considers standard, noisy, probabilistic ending, and noisy probabilistic ending tournaments.

Standard tournaments are similar to Axelrod's well-known tournaments [3]. In these tournaments, there are N strategies, and each strategy plays an iterated game with n turns against all other strategies, not including self-interactions. *Noisy tournaments* also involve N strategies and n turns, but in each turn, there is a probability p_n that a player's action is flipped. Compared to these two tournaments, in *probabilistic ending tournaments* the number of turns is not fixed. Instead, a match between strategies ends with a given probability p_e. Finally, *noisy probabilistic ending tournaments* incorporate both a noise probability p_n and an ending probability p_e. For smoother results, each tournament is repeated k times, and this repetition factor was allowed to vary to assess the impact of smoothing. The winner of each tournament is determined based on the average score achieved by a strategy from the entire set of repetitions, not by the number of wins.

The process of collecting tournament results is outlined in Algorithm 1. For each trial, a random size N is selected, and a random list of N strategies from the 195 available. Subsequently, one standard, one noisy, one probabilistic ending, and one noisy probabilistic ending tournament are conducted for the selected list of strategies. The parameters for the tournaments, as well as the number of repetitions, are chosen once for each trial. We have run a total of 11400 trials of Algorithm 1. For each trial, we collect the results for four different tournaments, resulting in a total of 45606 (11400×4) tournament results. Each tournament outputs a result summary in the form of Table 1.

Algorithm 1: Tournament Data Collection Algorithm

for *seed* $\in [0, 11420]$ **do**
- $N \leftarrow$ randomly select integer $\in [3, 195]$;
- players $\leftarrow$ randomly select N players;
- $k \leftarrow$ randomly select integer $\in [10, 100]$;
- $n \leftarrow$ randomly select integer $\in [1, 200]$;
- $p_n \leftarrow$ randomly select float $\in [0, 1]$;
- $p_e \leftarrow$ randomly select float $\in [0, 1]$;
- result standard $\leftarrow$ Axelrod.tournament(players, n, k);
- result noisy $\leftarrow$ Axelrod.tournament(players, n, p_n, k);
- result probabilistic ending $\leftarrow$ Axelrod.tournament(players, p_e, k);
- result noisy probabilistic ending $\leftarrow$ Axelrod.tournament(players, p_n, p_e, k);

return *result standard, result noisy, result probabilistic ending, result noisy probabilistic ending*;

During the data collection process, the probabilities of noise (p_n) and tournament ending (p_e) were allowed to take values between 0 and 1. However, commonly used values for these probabilities are $p_n \leq 0.1$ and

$p_e \leq 0.1$. This is to make the results more interpretable. For example, consider a strategy competing in an environment with $p_n > 0.1$. In cases with a high value of noise, most of the actions the strategy takes are the complete opposite of what the strategy is designed to do. Therefore, we will focus on the tournaments for which $p_n \leq 0.1$ and $p_e \leq 0.1$. Thus, the results presented here pertain to subsets of the noisy and probabilistic ending tournaments. Specifically, the results rely on 1150 tournaments with noise, 1134 tournaments with a probabilistic ending, and 117 tournaments with both noise and a probabilistic ending. We also provide an analysis of the paper considering the entire datasets, and these results are presented in the Supplementary Material. The general results of the analysis are not affected by the restriction of the noise and probabilistic ending probabilities.

3 Results

3.1 Top ranked strategies across tournaments

A strategy has participated in multiple tournaments of each type, and to evaluate its overall performance, we introduce a measure called the *normalized rank*. In each tournament, the strategies receive a rank (R), where 0 denotes that the strategy was the winner, and $N-1$ indicates that the strategy came last in the tournament. The normalized rank, denoted as r, is calculated as $r = \frac{R}{N-1}$. Thus, the rank a strategy achieved over the number of players in the tournament. The performance of the strategies is assessed based on the *median of the normalized rank*, denoted as $\bar{r}$.

For example, let's consider the well-known strategies `TFT` and Gradual. Each strategy participated in several tournaments of each type. In Figure 1 we show the distribution of the normalised ranks of these strategies in each of the four tournaments. We can observe that `TFT` looks to be normally distributed normalized rank. In comparison, Gradual's performance has longer tails, indicating that there were tournaments where the strategy performed very well or very poorly. Overall, Gradual achieves a lower median rank, signifying that it performs better than `TFT` except in the case of noisy and probabilistic ending tournaments (lower rank is better).

The top 15 strategies for each tournament type, based on $\bar{r}$, are presented in Table 2, while the r distributions for the top-ranked strategies can be found in Figure 2.

In standard tournaments dominating strategies were those trained using reinforcement learning techniques. 10 out of the 15 top strategies were introduced in [17]. These strategies are based on finite state automata (FSM), hidden Markov models (HMM), artificial neural networks (ANN), lookup tables (LookerUp), and stochastic

						Rates							
Rank	Name	Median score	Cooperation rating (C_r)	Win	Initial C	CC	CD	DC	DD	CC to C	CD to C	DC to C	DD to C
0	EvolvedLookerUp2 2 2	2.97	0.705	28.0	1.0	0.639	0.066	0.189	0.106	0.836	0.481	0.568	0.8
1	Evolved FSM 16 Noise 05	2.875	0.697	21.0	1.0	0.676	0.020	0.135	0.168	0.985	0.571	0.392	0.07
2	PSO Gambler 1 1 1	2.874	0.684	23.0	1.0	0.651	0.034	0.152	0.164	1.000	0.283	0.000	0.136
3	PSO Gambler Mem1	2.861	0.706	23.0	1.0	0.663	0.042	0.145	0.150	1.000	0.510	0.000	0.122
4	Winner12	2.835	0.682	20.0	1.0	0.651	0.031	0.141	0.177	1.000	0.441	0.000	0.462
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Table 1: **Result Summary Example of a Tournament.** A result summary consists of N rows, with each row containing information for each strategy that participated in the tournament. This information includes the strategy's rank (R), median score, the cooperation rate (C_r), the number of match wins, and the probability that the strategy cooperated in the opening move. Additionally, it provides the probabilities of a strategy being in any of the four states (CC, CD, DC, DD) and the cooperation rate after each state.

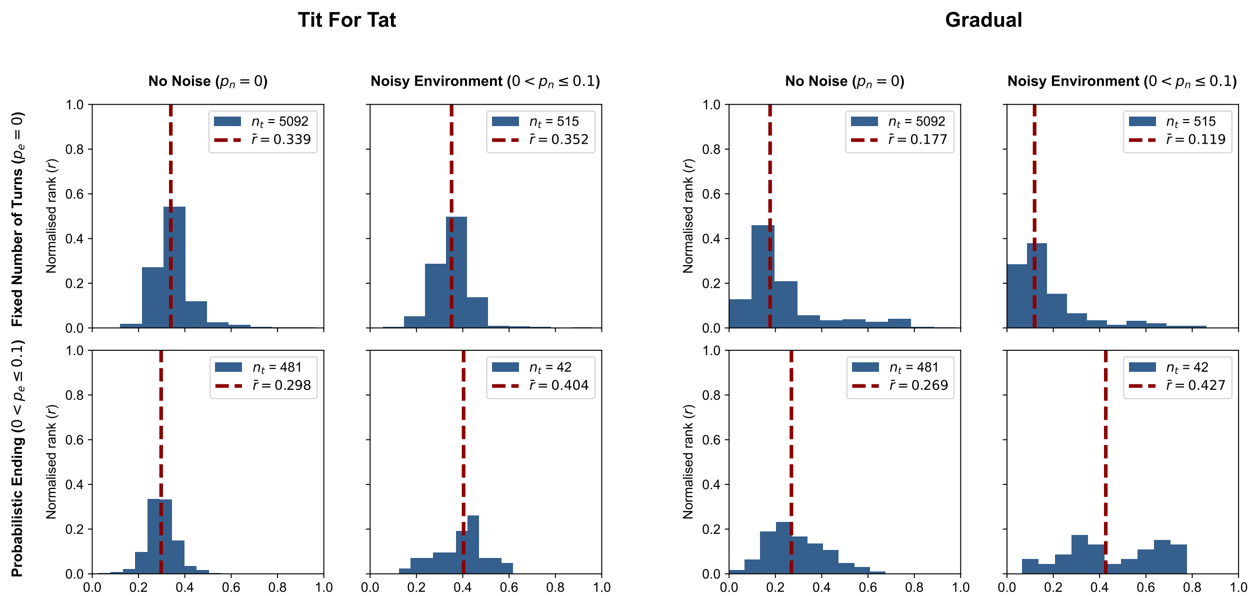


Figure 1: **Examples of normalized rank distributions for two strategies, `TFT` and Gradual.** We plot the distributions of r for the two strategies in the four tournament types. As a reminder, lower values of r correspond to better performances. The top left quadrant of each plot shows the distribution for standard tournaments (fixed number of turns and no noise). The top right quadrant shows the distribution for noisy tournaments (fixed number of turns and noise). The bottom left quadrant shows the distribution for probabilistic ending tournaments (no noise and probabilistic ending). Finally, the bottom right quadrant shows the distribution for noisy probabilistic ending tournaments (noise and probabilistic ending). In each quadrant, we also show the number of data points. Both strategies participated in a similar number of tournaments. Based on the median rank, which we use in this work to define overall performance, `TFT` performs best in probabilistic ending tournaments, whereas Gradual was in standard tournaments.

	Standard		Noisy ($p_n \leq 0.1$)		Probabilistic ending ($p_e \leq 0.1$)		Noisy probabilistic ending	
	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$
0	Evolved HMM 5	0.007	DBS	0.0	Evolved FSM 16	0.0	Raider	0.022
1	Evolved FSM 16	0.01	Evolved FSM 16 Noise 05	0.008	Evolved FSM 16 Noise 05	0.013	MEM2	0.037
2	EvolvedLookerUp2_2_2	0.011	Evolved ANN 5 Noise 05	0.013	MEM2	0.027	Prober 3	0.039
3	Evolved FSM 16 Noise 05	0.017	BackStabber	0.024	Evolved HMM 5	0.043	Evolved FSM 16 Noise 05	0.048
4	PSO Gambler 2_2_2	0.022	DoubleCrosser	0.025	EvolvedLookerUp2_2_2	0.049	Hard Prober	0.072
5	Evolved ANN	0.029	Evolved ANN 5	0.028	Spiteful Tit For Tat	0.059	Spiteful Tit For Tat	0.078
6	Evolved ANN 5	0.034	Evolved ANN	0.038	Nice Meta Winner	0.069	Better and Better	0.089
7	PSO Gambler 1_1_1	0.037	Spiteful Tit For Tat	0.051	NMWE Finite Memory	0.069	Grudger	0.091
8	Evolved FSM 4	0.049	Evolved HMM 5	0.051	NMWE Deterministic	0.07	Fortress4	0.096
9	PSO Gambler Mem1	0.05	Level Punisher	0.052	Grudger	0.07	Meta Winner Memory One	0.099
10	Winner12	0.06	Omega TFT	0.059	NMWE Long Memory	0.074	NMWE Long Memory	0.099
11	Fool Me Once	0.061	Fool Me Once	0.059	Nice Meta Winner Ensemble	0.076	Nice Meta Winner	0.104
12	DBS	0.071	PSO Gambler 2_2_2 Noise 05	0.067	EvolvedLookerUp1_1_1	0.077	NMWE Deterministic	0.109
13	DoubleCrosser	0.072	Evolved FSM 16	0.078	NMWE Memory One	0.08	NMWE Memory One	0.112
14	BackStabber	0.075	EugineNier	0.08	NMWE Stochastic	0.085	Nice Meta Winner Ensemble	0.115

Table 2: Top performances for each tournament type based on $\bar{r}$. The results of each type are based on 11420 unique tournaments. The results for noisy tournaments with $p_n < 0.1$ are based on 1151 tournaments, and for probabilistic ending tournaments with $p_e < 0.1$ on 1139. The top ranks indicate that trained strategies perform well in a variety of environments, but so do simple deterministic strategies. For noisy tournaments DBS is the top ranked strategy with $\bar{r} = 0$, thus DBS won every tournament it participated in. The same for Evolved FSM 16 Noise 05 in probabilistic ending.

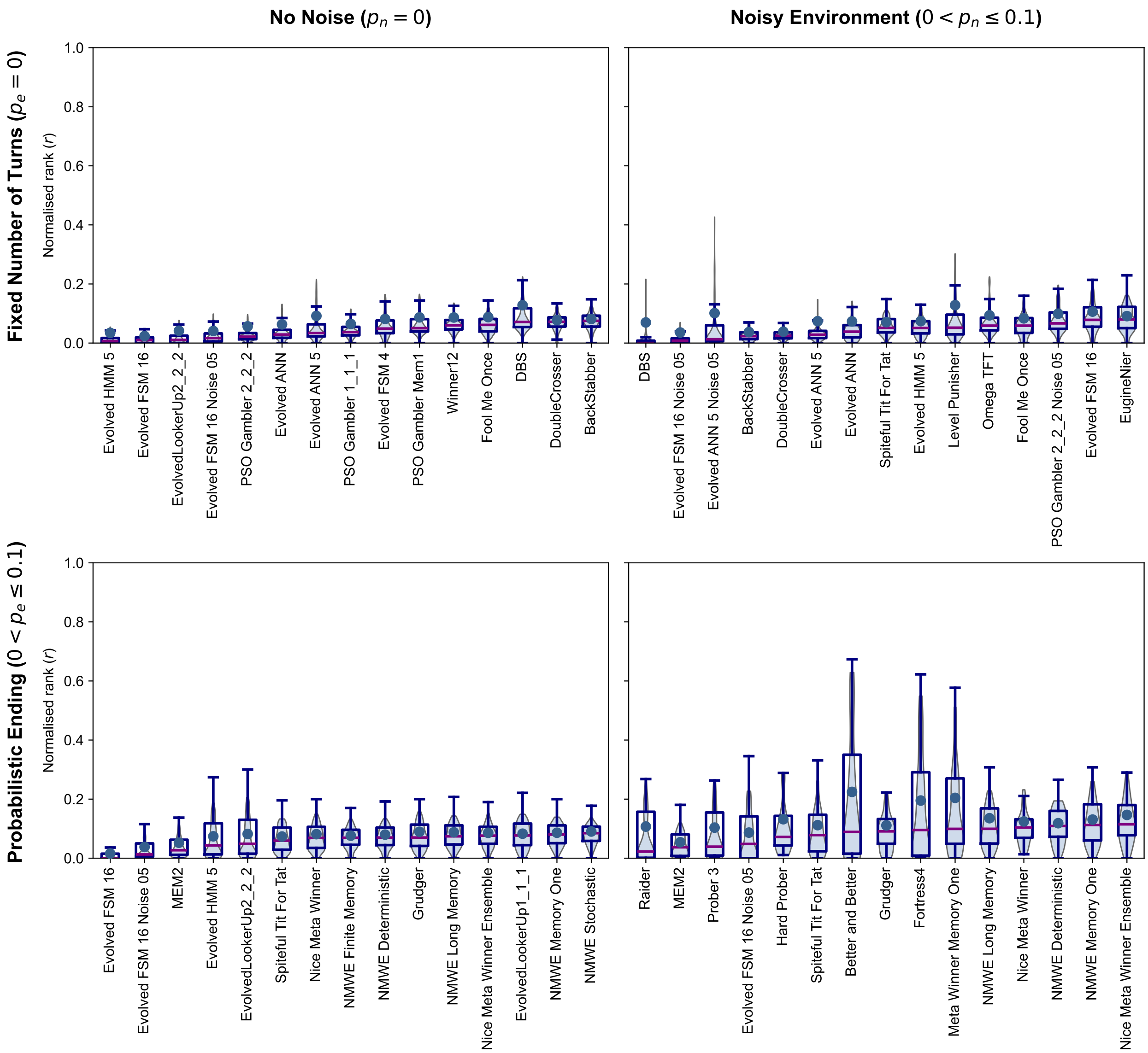


Figure 2: r **distributions of the top 15 strategies in different environments.** A lower value of $\bar{r}$ corresponds to a more successful performance. A strategy's r distribution skewed towards zero indicates that the strategy ranked highly in most tournaments it participated in. Most distributions are skewed towards zero.

lookup tables (Gambler). They have been trained using reinforcement learning algorithms (evolutionary and particle swarm algorithms) to perform well against a subset of the strategies in `Axelrod-Python` in a standard tournament. Thus, their performance in the specific setting was anticipated, although still noteworthy given the random sampling of tournament participants. DoubleCrosser and BackStabber, both from the `Axelrod-Python`, use the number of turns and are set to defect in the last two rounds. These strategies can be characterized as "cheaters" because their source code allows them to know the number of turns (unless the match has a probabilistic ending). These strategies were expected to not perform as well in tournaments where the number of turns is not specified. Finally, Winner 12 [16] and DBS [21] are both from the literature. DBS is a strategy specifically designed for noisy environments; however, it ranks highly in standard tournaments as well. Similarly, the fourth-ranked player, Evolved FSM 16 Noise 05, was trained for noisy tournaments yet performs well in standard tournaments.

In the case of noisy tournaments, the top-performing strategies include strategies specifically designed for noisy tournaments. These are DBS, Evolved FSM 16 Noise 05, Evolved ANN 5 Noise 05, PSO Gambler 2 2 2 Noise 05, and Omega Tit For Tat [13]. Omega TFT, a strategy designed to break the deadlocking cycles of CD and DC that TFT can fall into in noisy environments, places 10th. The rest of the top ranks are occupied by strategies that performed well in standard tournaments and deterministic strategies such as Spiteful Tit For Tat [22], Level Punisher [23], Eugine Nier [24].

Furthermore, in tournaments with probabilistic endings, the highly ranked strategies leaned towards defecting strategies and trained finite state automata, as demonstrated by the works of Ashlock et al.[25, 26]. The most effective strategies in probabilistic ending tournaments are also a series of ensemble Meta strategies, trained strategies that performed well in standard tournaments, and Grudger[20] and Spiteful Tit for Tat [22]. The Meta strategies [20] utilize a team of strategies and aggregate the potential actions of the team members into a single action in various ways.

While no single strategy consistently outperforms all others in any of the distinct tournament types or across various tournament types, certain types of strategies consistently achieve top rankings. These include strategies that have undergone training, those that retaliate, and those that adapt their behavior based on preassigned rules to optimize outcomes. These findings challenge some of Axelrod's suggestions, particularly the advice to "not be clever" and "not be envious".

3.2 The effect of strategy features on performance

For each strategy, we have a variety of features as described in Table 3. These features capture measures related to a strategy's behavior in the tournaments it competed in, as well as intrinsic properties, such as whether a strategy is deterministic or stochastic. The correlation coefficients between the features for performance evaluation, the median score and the median normalised rank are given by Table 4. The correlation coefficients between all features have also been calculated and a graphical representation can be found in the Supplementary Material.

In standard tournaments, the features CC to C, C_r, $C_r/C_{\max}$, and the cooperating ratio compared to C_{median} and C_{mean} have a moderately negative effect on the normalized rank (a smaller rank is better) and a moderate positive effect on the median score. The SSE error and the DD to C rate have the opposite effects. Thus, in standard tournaments, behaving cooperatively corresponds to a more successful performance. Even though being nice generally pays off, that does not hold against defective strategies. Being more cooperative after a mutual defection, that is not retaliating, is associated with lesser overall success in terms of normalized rank. Compared to standard tournaments, in both noisy and noisy probabilistic ending tournaments, the higher the rates of cooperation, the lower a strategy's success and median score. A strategy would not want to cooperate more than both the mean and median cooperator in such settings. In probabilistic ending tournaments, the cooperation rate of the winners and its relative comparison to the cooperation rates of the tournament have

feature	feature explanation	source	value type	min value	max value
stochastic	If a strategy is stochastic	strategy classifier from APL	boolean	Na	Na
makes use of game	If a strategy makes used of the game information	strategy classifier from APL	boolean	Na	Na
makes use of length	If a strategy makes used of the number of turns	strategy classifier from APL	boolean	Na	Na
memory usage	The memory size of a strategy divided by the number of turns	memory size from APL	float	0	1
SSE	A measure of how far a strategy is from ZD behaviour	method described in [27]	float	0	1
max cooperating rate (C_{max})	The biggest cooperating rate in a given tournament	result summary	float	0	1
min cooperating rate (C_{min})	The smallest cooperating rate in a given tournament	result summary	float	0	1
median cooperating rate (C_{median})	The median cooperating rate in a given tournament	result summary	float	0	1
mean cooperating rate (C_{mean})	The mean cooperating rate in a given tournament	result summary	float	0	1
C_r / C_{max}	A strategy's cooperating rate divided by the maximum	result summary	float	0	1
C_{min} / C_r	A strategy's cooperating rate divided by the minimum	result summary	float	0	1
C_r / C_{median}	A strategy's cooperating rate divided by the median	result summary	float	0	1
C_r / C_{mean}	A strategy's cooperating rate divided by the mean	result summary	float	0	1
C_r	The cooperating ratio of a strategy	result summary	float	0	1
CC to C rate	The probability a strategy will cooperate after a mutual cooperation	result summary	float	0	1
CD to C rate	The probability a strategy will cooperate after being betrayed by the opponent	result summary	float	0	1
DC to C rate	The probability a strategy will cooperate after betraying the opponent	result summary	float	0	1
DD to C rate	The probability a strategy will cooperate after a mutual defection	result summary	float	0	1
p_n	The probability of a player's action being flip at each interaction	trial summary	float	0	1
n	The number of turns	trial summary	integer	1	200
p_e	The probability of a match ending in the next turn	trial summary	float	0	1
N	The number of strategies in the tournament	trial summary	integer	3	195
k	The number of repetitions of a given tournament	trial summary	integer	10	100

Table 3: **Included features for performance evaluation analysis.** Stochastic, makes use of length and makes use of game are APL classifiers that determine whether a strategy is stochastic or deterministic, whether it makes use of the number of turns or the game's payoffs. The memory usage is calculated as the number of turns the strategy considers to make an action (which is specified in the APL) divided by the number of turns. The SSE (introduced in [27]) shows how close a strategy is to behaving as a ZDs, and subsequently, in an extortionate way. The method identifies the ZDs closest to a given strategy and calculates the algebraic distance between them as the sum of squared error (SSE). A SSE value of 1 indicates no extortionate behaviour at all whereas a value of 0 indicates that a strategy is behaving as a ZDs. The memory usage of strategies is the number of rounds of play used by the strategy when deciding on an action, divided by the number of turns in each match. For example, Winner12 uses the previous two rounds of play, and if participating in a match with 100 turns its memory usage would be 2/100. For strategies with an infinite memory size, for example Evolved FSM 16 Noise 05, memory usage is equal to 1. Note that for tournaments with a probabilistic ending the number of turns was not collected, so the memory usage feature is not used for probabilistic ending tournaments. The rest of the features considered are the CC to C, CD to C, DC to C, and DD to C rates as well as cooperating ratio of a strategy, the minimum (C_{min}), maximum (C_{max}), mean (C_{mean}) and median (C_{median}) cooperating ratios of each tournament.

	Standard		Noisy $p_n \leq 0.1$		Probabilistic ending $p_e \leq 0.1$		Noisy probabilistic ending	
	r	median score	r	median score	r	median score	r	median score
CC to C rate	-0.501	0.501	-0.210	0.194	-0.336	0.348	0.087	0.015
CD to C rate	0.226	-0.199	0.337	-0.235	0.458	-0.352	0.609	-0.372
DC to C rate	0.127	-0.100	0.227	-0.111	0.164	-0.105	0.410	-0.203
DD to C rate	0.412	-0.396	0.549	-0.391	0.433	-0.378	0.615	-0.407
C_r	-0.323	0.383	0.298	-0.051	-0.060	0.160	0.595	-0.213
C_{max}	0.000	0.050	-0.000	0.244	-0.000	0.079	-0.000	0.296
C_{min}	0.000	0.085	0.000	-0.070	0.000	0.128	0.000	0.000
C_{median}	0.000	0.209	0.000	0.572	-0.000	0.324	0.000	0.667
C_{mean}	0.000	0.229	-0.000	0.583	-0.000	0.354	-0.000	0.689
C_r / C_{max}	-0.323	0.381	0.307	-0.076	-0.060	0.156	0.608	-0.246
C_{min} / C_r	0.109	-0.080	-0.141	-0.011	0.024	0.029	-0.335	0.092
C_r / C_{median}	-0.330	0.353	0.326	-0.258	-0.065	0.111	0.614	-0.464
C_r / C_{mean}	-0.331	0.357	0.325	-0.228	-0.066	0.114	0.617	-0.431
N	-0.000	-0.009	-0.000	-0.017	-0.000	0.011	0.000	0.139
k	-0.000	-0.002	-0.000	-0.003	-0.000	0.010	-0.000	0.035
n	-0.000	-0.125	-0.000	-0.392	-	-	-	-
p_n	-	-	0.000	-0.244	-	-	0.000	-0.272
p_e	-	-	-	-	0.000	0.257	0.000	0.568
Make use of game	-0.003	-0.022	-0.047	0.014	-0.046	0.022	-0.110	0.057
Make use of length	-0.158	0.124	-0.224	0.139	-0.173	0.128	-0.206	0.115
SSE	0.473	-0.452	0.589	-0.412	0.458	-0.418	0.571	-0.383
stochastic	0.006	-0.024	0.010	-0.007	-0.001	0.001	-0.001	0.002
memory usage	-0.098	0.108	-0.080	0.114	-	-	-	-

Table 4: **Correlations between the features of Table 3 and the normalised rank and the median score.** The correlation coefficients are calculated using the Spearman's rank correlation coefficient.

no effect. The only features that have an effect are the CD to C rate, which is the tendency of a strategy to forgive, and the SSE rate, which has a positive effect on the normalized rank.

A multivariate linear regression has been fitted to model the relationship between the features and the normalized rank. Based on the graphical representation of the correlation matrices given in the Supplementary Information, several features are highly correlated and have been removed before fitting the linear regression model. The features included are given in Table 5 alongside their corresponding p values in distinct tournaments and their regression coefficients. The CD to C rate has a positively statistically significant effect on the normalized rank across all tournament types. This suggests that being generous tends to lower one's performance. In the case of probabilistic ending tournaments, the coefficient of the CD to C rate is the highest, indicating that one should be more provocative in this setting. Similarly, the SEE error rate has a positive effect on the normalized rank, suggesting that being extortionate pays off, especially in noisy tournaments. The measures of cooperation, C_r and $C_r/C_{\max}$, also exhibit a significant effect. In noisy probabilistic ending tournaments, this effect is positive; however, the coefficient is very close to zero. In other tournament types, the effect is negative, indicating that one should aim to be less cooperative than the mean cooperator of the tournament. However, we cannot interpret the result as suggesting that a strategy should be as uncooperative as possible.

The results presented here suggest that generosity/provocation and a strategy's cooperation rate, particularly in comparison to the tournament averages, are significant features. The analysis suggests that strategies should be more generous in noisy tournaments and less generous in probabilistic ending tournaments. Moreover, strategies should aim to not cooperate more than the mean cooperator in their tournaments. We note the analysis is limited as we only consider a linear relationship between these parameters and the rank. To further investigate the effects of the parameters discussed in this section, we have conducted a more detailed analysis in the next section, focusing on the performances of the winners of the tournaments.

	Standard		Noisy $p_n \leq 0.1$		Probabilistic ending $p_e \leq 0.1$		Noisy probabilistic ending	
	R adjusted: 0.541		R adjusted: 0.373		R adjusted: 0.457		R adjusted: 0.537	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
constant	0.695	0.000	0.560	0.000	0.627	0.000	0.345	0.005
CC to C rate	-0.042	0.000	-0.163	0.000	-0.046	0.000	0.032	0.029
CD to C rate	0.297	0.000	0.064	0.000	0.412	0.000	0.292	0.000
DC to C rate	0.198	0.000	0.142	0.000	0.193	0.000	0.193	0.000
SSE	0.258	0.000	0.328	0.000	0.190	0.000	0.228	0.000
C_{max}	-0.068	0.000	-0.048	0.214	-0.040	0.347	-0.011	0.936
C_{min}	-0.161	0.000	-0.029	0.367	-0.049	0.017	0.008	0.912
C_{mean}	0.117	0.000	-0.133	0.000	-0.159	0.000	-0.468	0.000
C_{min} / C_r	0.057	0.000	-0.006	0.322	0.054	0.000	0.034	0.099
C_r / C_{mean}	-0.468	0.000	-0.073	0.000	-0.150	0.000	0.094	0.000
k	0.000	0.325	0.000	0.965	0.000	0.079	0.000	0.065
n	0.000	0.000	-	-	-	-	-	-
memory usage	-0.010	0.000	-0.008	0.000	-	-	-	-
C_r / C_{median}	-	-	0.069	0.001	-0.142	0.000	-	-
p_n	-	-	-0.131	0.010	-	-	-0.278	0.048
p_e	-	-	-	-	-0.071	0.016	0.320	0.024

Table 5: **Results of multivariate linear regressions with r as the dependent variable.** R squared is reported for each model. The R scores of the fitted models indicate their capability to explain some of the variation in the median rank. Most of the features have a statistically significant effect on the normalized rank. A multivariate linear regression has also be fitted on the median score. The coefficients and p values of the features can be found in Supplementary Information. Both approaches lead to similar conclusions.

3.3 Features of top performing strategies

In Figure 3, we present the distributions of the cooperation ratio and C_r/C_{mean} for the winners of tournaments. A value of $C_r/C_{\text{mean}} = 1$ implies that the cooperation ratio of the winner was the same as the mean cooperating ratio of the tournament, and we observe that this occurs for most tournament types, apart from the case of noisy and probabilistically ending tournaments. In the case of probabilistic ending tournaments, there are several winners that cooperated much less than that, confirming the results of the previous section that defecting strategies can be winners in probabilistic ending tournaments. The distribution of the cooperation rates showcases a high cooperation rate in standard tournaments and probabilistic ending tournaments. In tournaments with noise, we observe a much less cooperative behavior, which could result from strategies being cautious of potential flip actions by the co-player or strategies not suited for noise holding grudges against defections.

Analyzing the SSE distributions across different tournament types (Figure 4) suggests that successful strategies exhibit some extortionate behavior, though not consistently. ZDs are a set of strategies that are often envious, as they attempt to exploit their opponents. The winners of the tournaments considered in this work demonstrate envious behavior, but not to the extent observed in many ZDs. While the exact interactions between matches are not recorded here, the work of [17], which introduced the trained strategies appearing in the top-ranked strategies of Section 3.1, did record such interactions. In [17], it was shown that clever strategies managed to achieve mutual cooperation with stronger strategies while exploiting weaker ones. This could explain the clever winners in our analysis and the observed SSE distributions.

This might also be the reason why ZDs fail to appear in the top ranks—they attempt to exploit all opponents and cannot actively adapt back to mutual cooperation against stronger strategies, which requires a deeper memory. It's worth noting that ZDs tend to perform poorly in population games for a similar reason: they aim to exploit other players using ZDs, failing to form a cooperative subpopulation [28]. This makes them effective invaders but poor at resisting invasion.

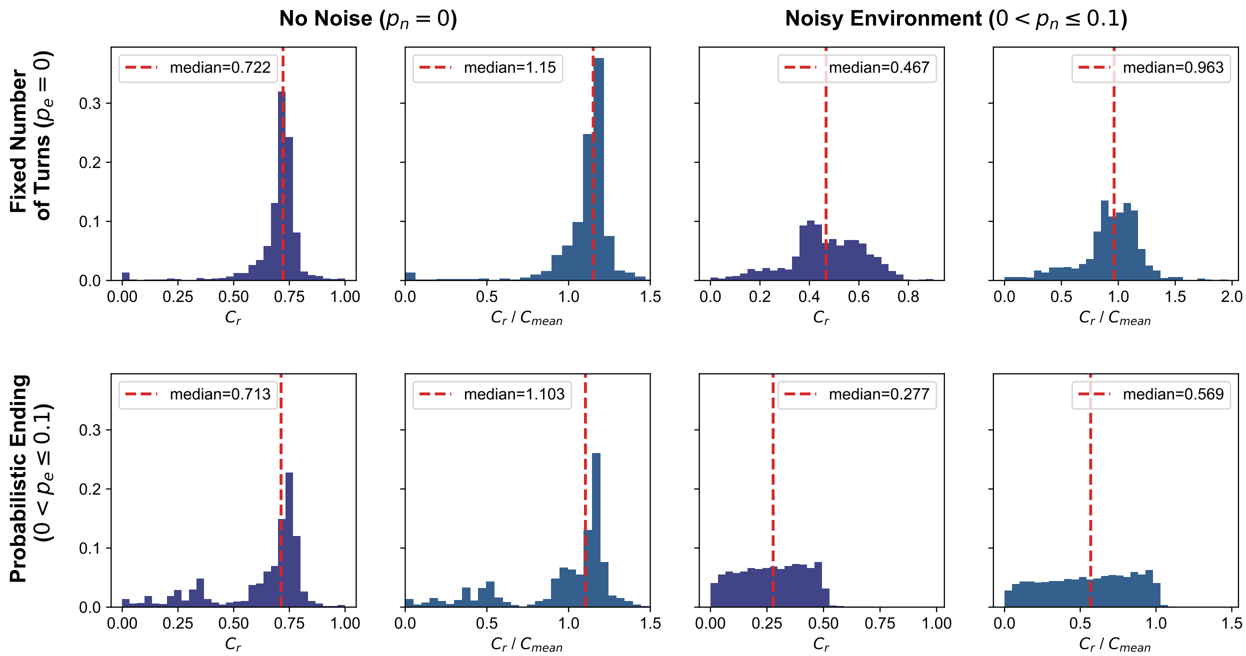


Figure 3: **Distributions of C_r and C_r/C_{mean} for the winners of tournaments.** A value of $C_r/C_{\text{mean}} = 1$ imply that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament.

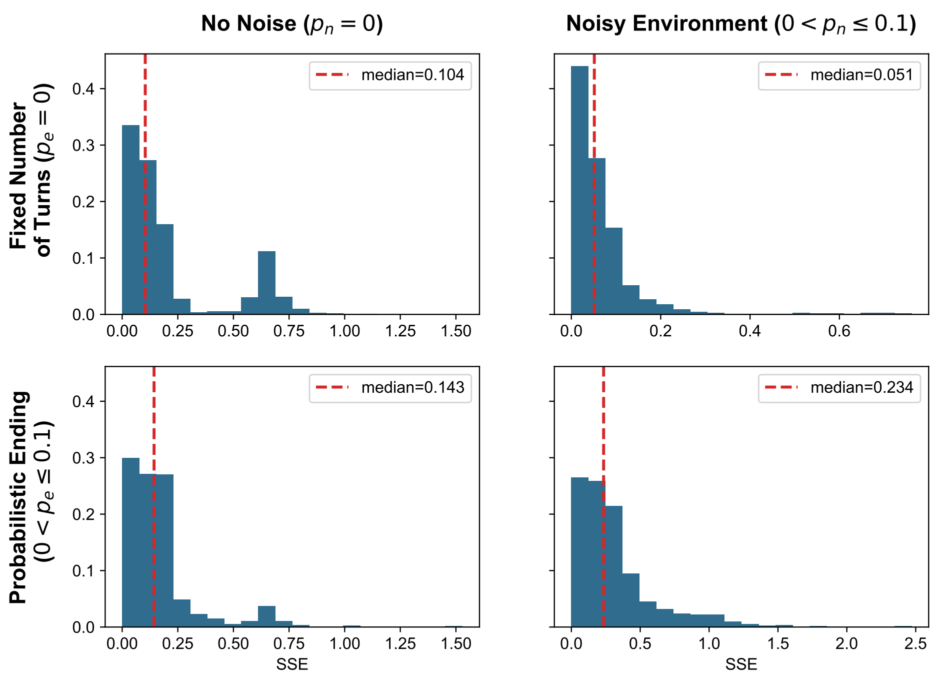


Figure 4: **Distributions of SSE error for the winners of tournaments.** As a reminder, the SSE error shows how close a strategy is to behaving as a ZDs, and subsequently, in an extortionate way. A SSE value of 1 indicates no extortionate behaviour at all whereas a value of 0 indicates that a strategy is behaving as a ZDs.

Finally, we examine the distributions of the cooperation rates after the outcomes CC, CD, DC, and DD, as shown in Figure 5. In the case of cooperating after mutual cooperation, the results align with expectations; the distributions skew towards 1, indicating that the winners of the tournaments are more likely to cooperate after mutual cooperation. Regarding the CD outcome and the likelihood to cooperate after such a result, capturing generosity, the distributions skew towards 1/2, not 1, suggesting that strategies need to reduce their readiness to forgive. This aligns with the known result that Generous Tit For Tat generally outperforms `TFT` in most settings. In probabilistic ending tournaments, there is a peak at 0, suggesting that strategies should not be too generous in tournaments with short matches. Such a peak also appears in standard tournaments; however, not in tournaments with noise, where a strategy should be more generous.

Part of a strategy's envious behavior can be captured by the rate of DC to C. In noisy tournaments, winners are not too envious, but in tournaments without noise, we can see that winners behave in two ways. Some are a bit envious, whereas others are very envious. In the DD to D, we can observe that, expectedly, the results are skewed towards 0. However, there are winners that attempt to recover from a DD outcome. The remaining results are as expected, skewed towards 0.

4 Discussion

This manuscript explores the performance of 195 strategies in the `IPD` in thousands of computer tournaments. The collection of computer tournaments presented here is the largest and most diverse in the literature. The 195 strategies are drawn from `Axelrod-Python` library and include strategies from the `IPD` literature. The computer tournaments encompass four different types. So, what is the best way to play the `IPD`? And is there a single dominant strategy for the `IPD`? There was not a single strategy within the collection of 195 strategies that managed to perform well in all the tournament variations it competed in. A strategy ranking highly in a specific environment did not guarantee its success over different tournament types, with a few exceptions - strategies that generalize better. Already well-known in the AI/ML literature, adding noise to training data leads to more robust models. We see that clearly here, where the strategies trained for noise (or designed for noise) tend to be better generalists. There were instances where a few strategies trained in narrow conditions outperformed more generalist strategies, as they tend to overfit. However, the strategies trained with noise perform well in general, whilst the strategies trained specifically on no noise or small subpopulations do not.

We also examined the best-performing strategies across various tournament types and analyzed their salient features. This demonstrated that there are properties associated with the success of strategies that contradict the originally suggested properties of Axelrod [1]. We showed that complex or **clever** strategies can be effective, whether trained against a corpus of possible opponents or purposely designed to mitigate the impact of noise such as the DBS strategy. Moreover, we found some strategies designed or trained for noisy environments were also highly ranked in noise-free tournaments which reinforces the idea that strategies' complexity/cleverness is not necessarily a liability, rather it can confer adaptability to a more diverse set of environments. We also showed that while the type of exploitation attempted by ZDs is not typically effective in standard tournaments, **envious** strategies capable of both exploiting and not their opponents can be highly successful. Based on the results of [17] this could be because they are selectively exploiting weaker opponents while mutually cooperating with stronger opponents. Highly noisy or tournaments with short matches also favoured envious strategies. These environments mitigated the value of being nice. Uncertainty enables exploitation, reducing the ability of maintaining or enforcing mutual cooperation, while triggering grudging strategies to switch from typically cooperating to typically defecting.

The features analysis of the best performing strategies demonstrated that a strategy should reciprocate, as suggested by Axelrod, but it should relax its readiness to do so and be more **generous**. For noisy environments this is inline with the results of [7, 8, 9, 10], however, we also showed that generosity pays off even in standard settings, and that in fact the only setting a strategy would want to be too provocable is

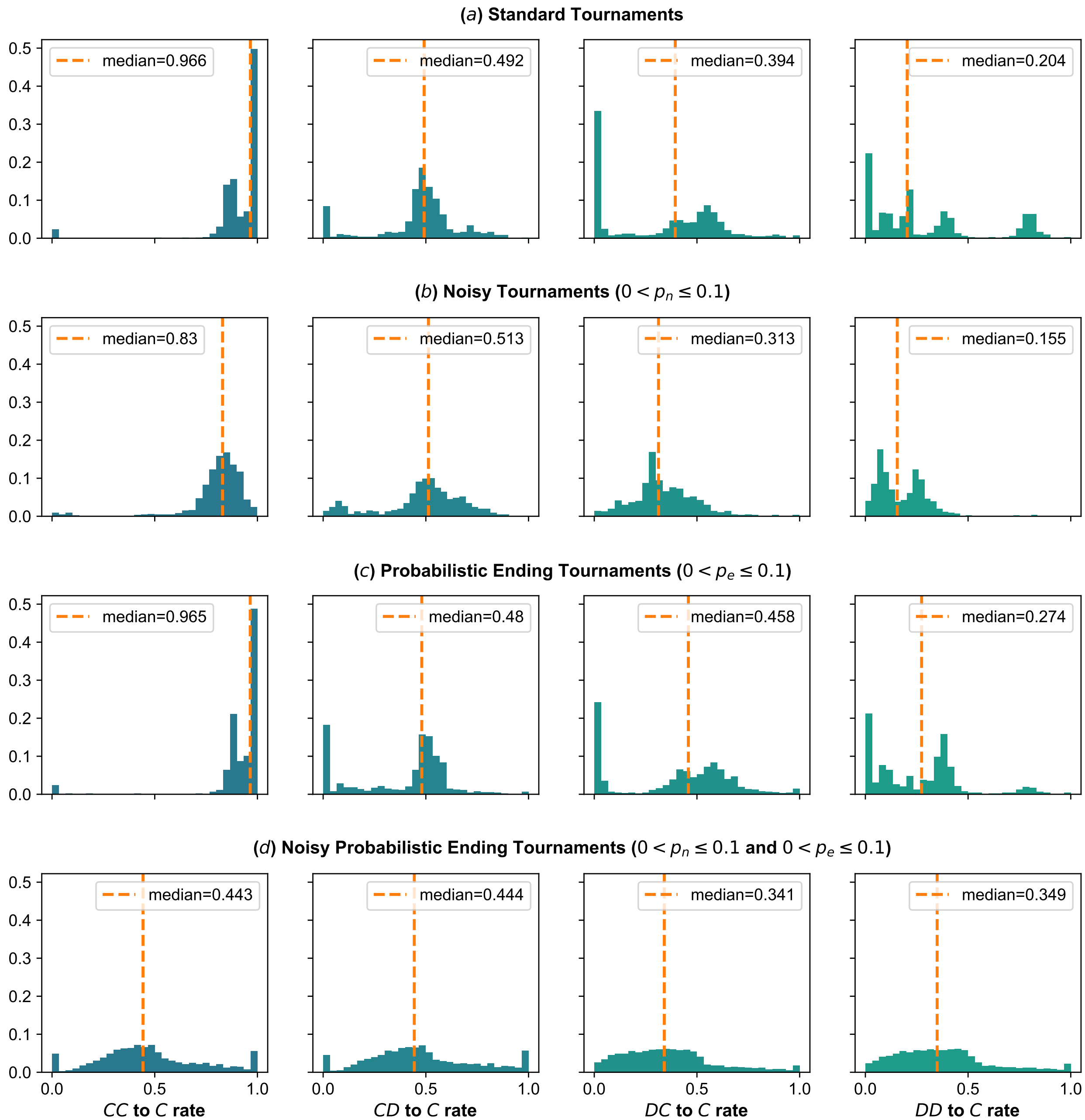


Figure 5: **Distributions of rates CC to C, CD to C, DC to C, and DD to C for the winners of tournaments.**

when the matches are not long. Forgiveness as defined by Axerlod was not explored here. This was mainly because the two round states were not recorded during the data collection. This could be a topic of future work that examines the impact of considering more rounds of history. The features analysis also concluded that there is a significant importance in **adapting to the environment**, and more specifically, to the mean cooperator. In most tournament types, the winner of the tournament was also the average cooperator. Even in tournaments with short matches where defecting behavior could secure a win, a large number of winners were average cooperators.

This could potentially explain the early success of `TFT`. `TFT` naturally achieves a cooperation rate near C_{mean} by virtue of copying its opponent's last move while also minimizing instances where it is exploited by an opponent (cooperating while the opponent defects), at least in non-noisy tournaments. It could also explain why Tit For N Tats does not fare well for $N > 1$ – it fails to achieve the proper cooperation ratio by tolerating too many defections.

Similarly, our results could suggest an explanation regarding the intuitively unexpected effectiveness of memory-one strategies historically. Given that among the important features associated with success are the relative cooperation rate to the population average and the four memory-one probabilities of cooperating conditional on the previous round of play, these features can be optimized by a memory-one strategy such as `TFT`. Usage of more history becomes valuable when there are exploitable opponent patterns. This is indicated by the importance of SSE as a feature, showing that the first-approximation provided by a memory-one strategy is no longer sufficient. These results highlight a central idea in evolutionary game theory in this context: the fitness landscape is a function of the population (where fitness in this case is tournament performance). While that may seem obvious now, it shows why historical tournament results on small or arbitrary populations of strategies have so often failed to produce generalizable results.

Overall, the five properties successful strategies need to have in a `IPD` competition based on the analysis that has been presented in this manuscript are:

(i) Be "nice" in non-noisy environments or when game lengths are longer

(ii) Be provocable in tournaments with short matches, and generous in tournaments with noise

(iii) Be a little bit envious

(iv) Be clever

(v) Adapt to the environment (including the population of strategies).

The results presented here were based only on a subset of the whole data we have collected. The analysis of the full dataset is discussed in the Supplementary Information. However, we can see that the general results of our work remain the same. In the Supplementary Information, we also evaluate the importance of features using a random forest classifier and a clustering approach. The results of these analyses are also in line with the results presented here.

The data set described in this work contains the largest number of `IPD` tournaments, to the authors knowledge. The raw data set is available at [29] and the processed data at [30]. Further data mining could be applied and provide new insights in the field.

5 Data availability statement

The raw and processed datasets have been made publicly available [30, 29] and can be used for further analysis and insights.

6 Acknowledgements

N.G. acknowledges the generous support of the European Research Council Starting Grant 850529: E-DIRECT and the Max Planck Society. A variety of open-source software have been used in this work. The authors would like to express their gratitude to the open-source software community, whose invaluable contributions significantly enhanced the development and execution of this research. Namely, the authors would like to thank the developers of the following software packages: Axelrod-Python library for IPD simulations [20], the Matplotlib library for visualisation [31], The Numpy library for data manipulation [32], and finally the scikit-learn library for data analysis [33].

References

[1] Axelrod, R. & Hamilton, W. D. The evolution of cooperation. *science* **211**, 1390–1396 (1981).

[2] Flood, M. M. Some experimental games. *Management Science* **5**, 5–26 (1958). URL `https://doi.org/10.1287/mnsc.5.1.5`. `https://doi.org/10.1287/mnsc.5.1.5`.

[3] Axelrod, R. Effective choice in the prisoner's dilemma. *Journal of Conflict Resolution* **24**, 3–25 (1980). URL `https://doi.org/10.1177/002200278002400101`. `https://doi.org/10.1177/002200278002400101`.

[4] Axelrod, R. More effective choice in the prisoner's dilemma. *Journal of Conflict Resolution* **24**, 379–403 (1980).

[5] Beaufils, B., Delahaye, J. P. & Mathieu, P. Our meeting with gradual, a good strategy for the iterated prisoner's dilemma. In *Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, 202–209 (1997).

[6] Tzafestas, E. Toward adaptive cooperative behavior **2**, 334–340 (2000).

[7] Bendor, J., Kramer, R. M. & Stout, S. When in doubt... cooperation in a noisy prisoner's dilemma. *The Journal of Conflict Resolution* **35**, 691–719 (1991). URL `http://www.jstor.org/stable/174072`.

[8] Donninger, C. *Is it Always Efficient to be Nice? A Computer Simulation of Axelrod's Computer Tournament* (Physica-Verlag HD, Heidelberg, 1986). URL `https://doi.org/10.1007/978-3-642-95874-8_9`.

[9] Molander, P. The optimal level of generosity in a selfish, uncertain environment. *The Journal of Conflict Resolution* **29**, 611–618 (1985). URL `http://www.jstor.org/stable/174244`.

[10] Selten, R. & Hammerstein, P. Gaps in harley's argument on evolutionarily stable learning rules and in the logic of "tit for tat". *Behavioral and Brain Sciences* **7**, 115–116 (1984).

[11] Nowak, M. A. & Sigmund, K. Tit for tat in heterogeneous populations. *Nature* **355**, 250 (1992).

[12] Nowak, M. & Sigmund, K. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature* **364**, 56 (1993).

[13] Kendall, G., Yao, X. & Chong, S. Y. *The iterated prisoners' dilemma: 20 years on*, vol. 4 (World Scientific, 2007).

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

[15] Stewart, A. J. & Plotkin, J. B. Extortion and cooperation in the prisoner's dilemma. *Proceedings of the National Academy of Sciences* **109**, 10134–10135 (2012).

[16] Mathieu, P. & Delahaye, J. P. New winning strategies for the iterated prisoner's dilemma. *Journal of Artificial Societies and Social Simulation* **20**, 12 (2017). URL `http://jasss.soc.surrey.ac.uk/20/4/12.html`.

[17] Harper, M. *et al.* Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PloS one* **12**, e0188046 (2017).

[18] Axelrod, R. The evolution of strategies in the iterated prisoner's dilemma. *Genetic Algorithms and Simulated Annealing* 32–41 (1987). URL `http://ci.nii.ac.jp/naid/10000082922/en/`.

[19] Miller, J. H. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization* **29**, 87 – 112 (1996). URL `http://www.sciencedirect.com/science/article/pii/0167268195000526`.

[20] project developers, T. A. Axelrod: 3.0.0. `http://dx.doi.org/10.5281/zenodo.807699` (2016).

[21] Au, T. C. & Nau, D. Accident or intention: that is the question (in the noisy iterated prisoner's dilemma). In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 561–568 (ACM, 2006).

[22] Lifl (1998) prison. `http://www.lifl.fr/IPD/ipd.frame.html`. Accessed: 2017-10-23.

[23] A., E. Coopsim v0.9.9 beta 6. `https://github.com/jecki/CoopSim/` (2015).

[24] prase. Prisoner's dilemma tournament results. `https://www.lesswrong.com/posts/hamma4XgeNrsvAJv5/prisoner-s-dilemma-tournament-results` (2011).

[25] Ashlock, W. & Ashlock, D. Changes in prisoner's dilemma strategies over evolutionary time with different population sizes. In *2006 IEEE International Conference on Evolutionary Computation*, 297–304 (IEEE, 2006).

[26] Ashlock, W., Tsang, J. & Ashlock, D. The evolution of exploitation. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, 135–142 (IEEE, 2014).

[27] Knight, V. A., Harper, M., Glynatsi, N. E. & Gillard, J. Recognising and evaluating the effectiveness of extortion in the iterated prisoner's dilemma. *CoRR* **abs/1904.00973** (2019). URL `http://arxiv.org/abs/1904.00973`. `1904.00973`.

[28] Knight, V., Harper, M., Glynatsi, N. E. & Campbell, O. Evolution reinforces cooperation with the emergence of self-recognition mechanisms: an empirical study of the moran process for the iterated prisoner's dilemma. *arXiv preprint arXiv:1707.06920* (2017).

[29] Glynatsi, N. E. A data set of 45686 Iterated Prisoner's Dilemma tournaments' results [RAW DATA] (2023). URL `https://doi.org/10.5281/zenodo.10246248`.

[30] E., N. A data set of 45686 Iterated Prisoner's Dilemma tournaments' results (2023). URL `https://doi.org/10.5281/zenodo.10246247`.

[31] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* **9**, 90–95 (2007).

[32] Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering* **13**, 22–30 (2011).

[33] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

Electronic supplementary material

Properties of Winning Iterated Prisoner's Dilemma Strategies

Nikoleta E. Glynatsi, Vincent A. Knight, Marc Harper

This document provides additional details and analysis based on the complete datasets for noisy and probabilistic ending tournaments. In Section 1, a summary of all parameters and notation used in the manuscript is presented. Section 2 offers further details on the correlation analysis and includes additional information. In Section 4, we present the results of the correlation analysis when all features of the strategies are considered. In Section 5, the results of the random forest and clustering evaluations with the inclusion of all features are discussed. Section 6 presents the results of the regression analysis on the median score of the strategies. Finally, in Section 7, a comprehensive list of all strategies considered in this work is provided.

1 Parameters Summary

All the parameters used in this manuscript alongside their explanation are given by Table 1.

Feature	Explanation
SSE	A measure of how far a strategy is from extortionate behaviour defined in [26].
$C_{\max}$	The biggest cooperating rate in the tournament.
$C_{\min}$	The smallest cooperating rate in the tournament.
C_{median}	The median cooperating rate in the tournament.
C_{mean}	The mean cooperating rate in the tournament.
C_r / $C_{\max}$	A strategy's cooperating rate divided by the maximum cooperating rate in the tournament.
$C_{\min}$ / C_r	The minimum in the tournament divided by a strategy's cooperating rate.
C_r / C_{median}	A strategy's cooperating rate divided by the median cooperating rate in the tournament.
C_r / C_{mean}	A strategy's cooperating rate divided by the mean cooperating rate in the tournament.
C_r	The cooperating rate of a strategy.
CC to C rate	The probability a strategy will cooperate after a mutual cooperation.
CD to C rate	The probability a strategy will cooperate after being betrayed by the opponent.
DC to C rate	The probability a strategy will cooperate after betraying the opponent.
DD to C rate	The probability a strategy will cooperate after a mutual defection.
p_n	The probability of a player's action being flipped at each interaction.
n	The number of turns in a match.
p_e	The probability of a match ending in the next turn.
N	The number of strategies in the tournament.
k	The number that a given tournament is repeated.

Table 1: The features which are included in the performance evaluation analysis.

2 Top ranked strategies and features analysis for the entire data set

In this section we carry out a similar analysis as the one presented in main manuscript, but this time we use the entire data set for noisy and probabilistic ending tournaments.

2.1 Top ranked strategies across tournaments

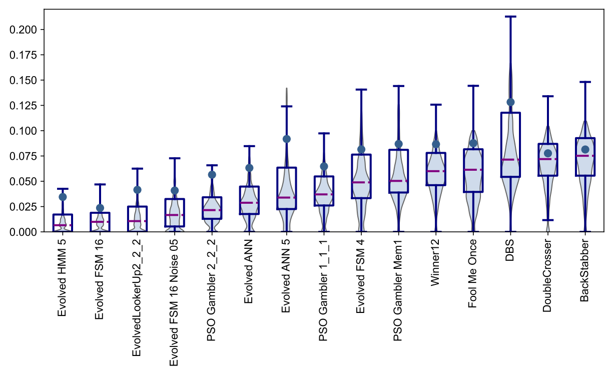
The top 15 strategies for each tournament type based on $\bar{r}$ are given in Table 2. The data collection process was designed such that the probabilities of noise and ending of the match varied between 0 and 1. However, commonly used values for these probabilities are values less than 0.1. Thus, Table 2 also includes the top 15 strategies in noisy tournaments with $p_n < 0.1$ and probabilistic ending tournaments with $p_e < 0.1$.

	Standard		Noisy		Noisy ($p_n < 0.1$)		Probabilistic ending		Probabilistic ending ($p_e < 0.1$)		Noisy probabilistic ending	
	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$
0	Evolved HMM 5	0.007	Grumpy	0.14	DBS	0.0	Fortress4	0.013	Evolved FSM 16	0.0	Alternator	0.304
1	Evolved FSM 16	0.01	\$e\$	0.19	Evolved FSM 16 Noise 05	0.008	Defector	0.014	Evolved FSM 16 Noise 05	0.013	\$\phi\$	0.31
2	EvolvedLookerUp2_2_2	0.011	Tit For 2 Tats	0.206	Evolved ANN 5 Noise 05	0.013	Better and Better	0.016	MEM2	0.027	\$e\$	0.312
3	Evolved FSM 16 Noise 05	0.017	Slow Tit For Two Tats	0.21	BackStabber	0.024	Tricky Defector	0.019	Evolved HMM 5	0.043	\$\pi\$	0.317
4	PSO Gambler 2_2_2	0.022	Cycle Hunter	0.215	DoubleCrosser	0.025	Fortress3	0.022	EvolvedLookerUp2_2_2	0.049	Limited Retaliate	0.353
5	Evolved ANN	0.029	Risky QLearner	0.222	Evolved ANN 5	0.028	Gradual Killer	0.025	Spiteful Tit For Tat	0.059	Anti Tit For Tat	0.354
6	Evolved ANN 5	0.034	Cycler CCCCCD	0.229	Evolved ANN	0.038	Aggravater	0.028	Nice Meta Winner	0.069	Limited Retaliate 3	0.356
7	PSO Gambler 1_1_1	0.037	Retaliate 3	0.23	Spiteful Tit For Tat	0.051	Raider	0.031	NMWE Finite Memory	0.069	Retaliate 3	0.356
8	Evolved FSM 4	0.049	Retaliate 2	0.239	Evolved HMM 5	0.051	Cycler DDC	0.045	NMWE Deterministic	0.07	Retaliate	0.357
9	PSO Gambler Mem1	0.05	Defector Hunter	0.24	Level Punisher	0.052	Hard Prober	0.051	Grudger	0.07	Retaliate 2	0.358
10	Winner12	0.06	Retaliate	0.242	Omega TFT	0.059	SolutionB1	0.06	NMWE Long Memory	0.074	Limited Retaliate 2	0.361
11	Fool Me Once	0.061	Hard Tit For 2 Tats	0.25	Fool Me Once	0.059	Meta Minority	0.061	Nice Meta Winner Ensemble	0.076	Hopeless	0.368
12	DBS	0.071	Arrogant QLearner	0.25	PSO Gambler 2_2_2 Noise 05	0.067	Bully	0.061	EvolvedLookerUp1_1_1	0.077	Arrogant QLearner	0.406
13	DoubleCrosser	0.072	Limited Retaliate 3	0.253	Evolved FSM 16	0.078	EasyGo	0.071	NMWE Memory One	0.08	Cautious QLearner	0.409
14	BackStabber	0.075	ShortMem	0.253	EugineNier	0.08	Fool Me Forever	0.071	NMWE Stochastic	0.085	Fool Me Forever	0.418

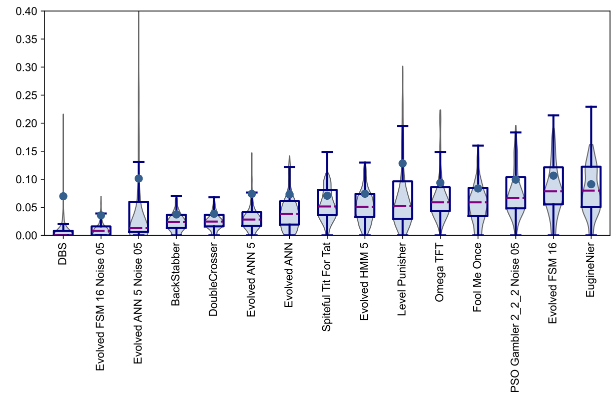
Table 2: Top performances for each tournament type based on $\bar{r}$. The results of each type are based on 11420 unique tournaments. The results for noisy tournaments with $p_n < 0.1$ are based on 1151 tournaments, and for probabilistic ending tournaments with $p_e < 0.1$ on 1139. The top ranks indicate that trained strategies perform well in a variety of environments, but so do simple deterministic strategies. The normalised medians are close to 0 for most environments, except environments with noise not restricted to 0.1 regardless of the number of turns. Noisy and noisy probabilistic ending tournaments have the highest medians.

The r distributions for the top ranked strategies of Table 2 are given by Figure 1.

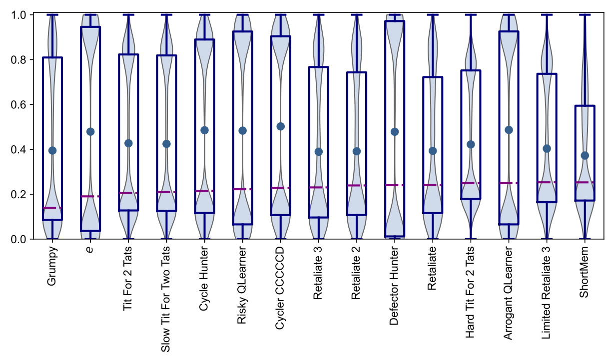
In standard tournaments 10 out of the 15 top strategies were introduced in [22]. These are strategies based on finite state automata (FSM), hidden Markov models (HMM), artificial neural networks (ANN), lookup tables (LookerUp) and stochastic lookup tables (Gambler) that have been trained using reinforcement learning algorithms (evolutionary and particle swarm algorithms). They have been trained to perform well against a subset of the strategies in APL in a standard tournament, thus their performance in the specific setting was anticipated although still noteworthy given the random sampling of tournament participants. DoubleCrosser, BackStabber and Fool Me Once, are strategies not from the literature but from the APL. DoubleCrosser is an extension of BackStabber and both strategies make use of the number of turns because they are set to defect on the last two rounds. It should be noted that these strategies can be characterised as “cheaters” because the source code of the strategies allows them to know the number of turns in a match (unless the match has a probabilistic ending). These strategies were expected to not perform as well in tournaments where the number of turns is not specified. Finally, Winner 12 [32] and DBS [10] are both from the literature. DBS is a strategy specifically designed for noisy environments, however, it ranks highly in standard tournaments as well. Similarly the fourth ranked player, Evolved FSM 16 Noise 05, was trained for noisy tournaments yet performs well in standard tournaments. Figure 1a shows that these strategies typically perform well in any standard tournament in which they participate.



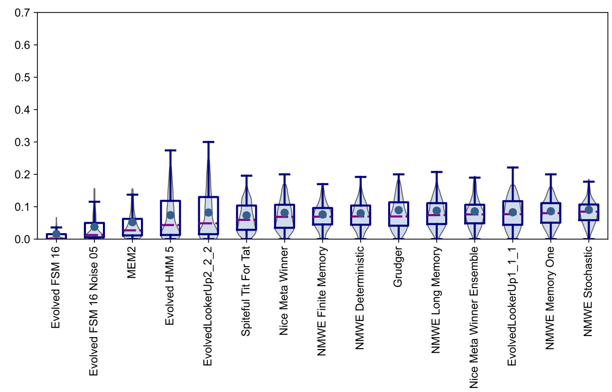
(a) r distributions of top 15 strategies in standard tournaments.



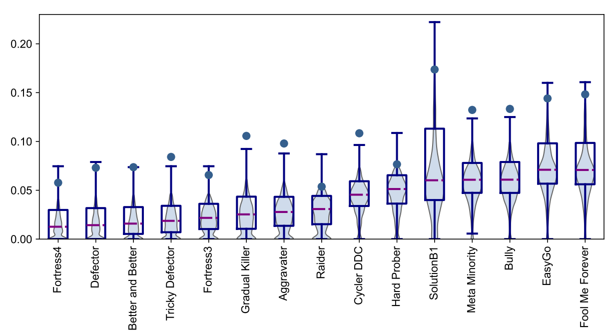
(b) r distributions of top 15 strategies in noisy tournaments with $p_n < 0.1$.



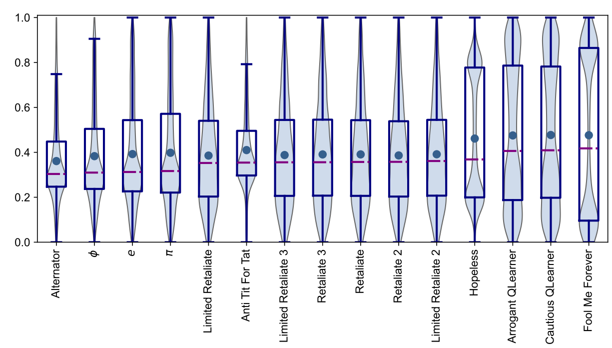
(c) r distributions of top 15 strategies in noisy tournaments.



(d) r distributions of top 15 strategies in 1139 probabilistic ending tournaments with $p_e < 0.1$.



(e) r distributions of top 15 strategies in probabilistic ending tournaments.



(f) r distributions of top 15 strategies in noisy probabilistic ending tournaments.

Figure 1: r distributions of the top 15 strategies in different environments. A lower value of $\bar{r}$ corresponds to a more successful performance. A strategy's r distribution skewed towards zero indicates that the strategy ranked highly in most tournaments it participated in. Most distributions are skewed towards zero except the distributions with unrestricted noise, supporting the conclusions from Table 2.

In the case of noisy tournaments with smaller noise $p_n < 0.1$ the top performing strategies include strategies specifically designed for noisy tournaments. These are DBS, Evolved FSM 16 Noise 05, Evolved ANN 5 Noise 05, PSO Gambler 2 2 2 Noise 05 and Omega Tit For Tat [25]. Omega TFT, a strategy designed to break the deadlocking cycles of CD and DC that TFT can fall into in noisy environments, places 10th. The rest of the top ranks are occupied by strategies which performed well in standard tournaments and deterministic strategies such as Spiteful Tit For Tat [1], Level Punisher [3], Eugine Nier [36].

In contrast, the performance of the top ranked strategies in noisy environments when $p_n \in [0, 1]$ is bimodal. The top strategies include strategies which decide their actions based on the cooperation to defection ratio, such as ShortMem [18], Grumpy [38] and e [38], and the Retaliate strategies which are designed to defect if the opponent has tricked them more often than a given percentage of the times that they have done the same. The bimodality of the r distributions is explained by Figure 2 which demonstrates that the top 6 strategies were highly ranked due to the their performance in tournaments with $p_n > 0.5$, and that in tournaments with $p_n < 0.5$ they performed poorly. At a noisy level of 0.5 or greater, mostly cooperative strategies become mostly defectors and vice versa.

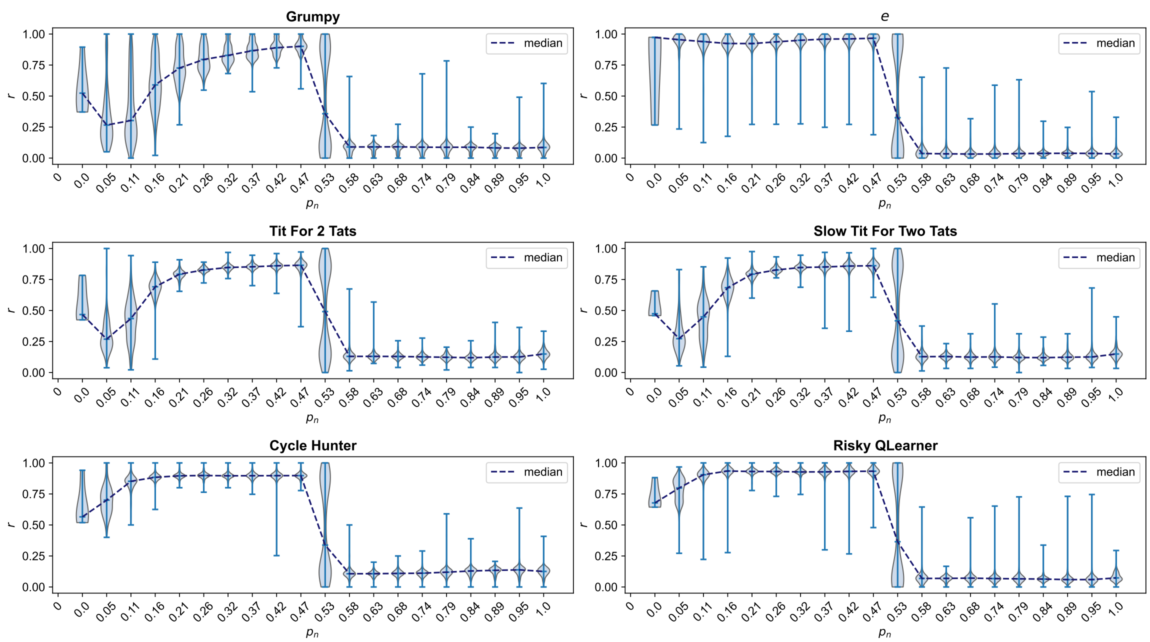


Figure 2: Normalised rank r distributions for top 6 strategies in noisy tournaments over the probability of noisy (p_n).

The most effective strategies in probabilistic ending tournaments with $p_e < 0.1$ are a series of ensemble Meta strategies, trained strategies which performed well in standard tournaments, and Grudger [38] and Spiteful Tit for Tat [1]. The Meta strategies [38] utilize a team of strategies and aggregate the potential actions of the team members into a single action in various ways. Figure 1d indicates that these strategies performed well in any probabilistic ending tournament.

In probabilistic ending tournaments with $p_e \in [0, 1]$ the top ranks are mostly occupied by defecting strategies such as Better and Better, Gradual Killer, Hard Prober (all from [38]), Bully (Reverse Tit For Tat) [34] and Defector, and a series of strategies based on finite state automata introduced by Daniel Ashlock and Wendy Ashlock: Fortress 3, Fortress 4 (both introduced in [8]), Raider [9] and Solution B1 [9]. The success of defecting strategies in probabilistic ending tournaments is due to larger values of p_e which lead to shorter matches (the expected number of rounds is $1/p_e$), so the impact of the PD being iterated is subdued. This is captured by the Folk Theorem [20] as defecting strategies do better when the likelihood of the game ending in the next turn increases. This is demonstrated by Figure 3, which gives the distributions of r for the top 6 strategies in probabilistic ending tournaments over p_e.

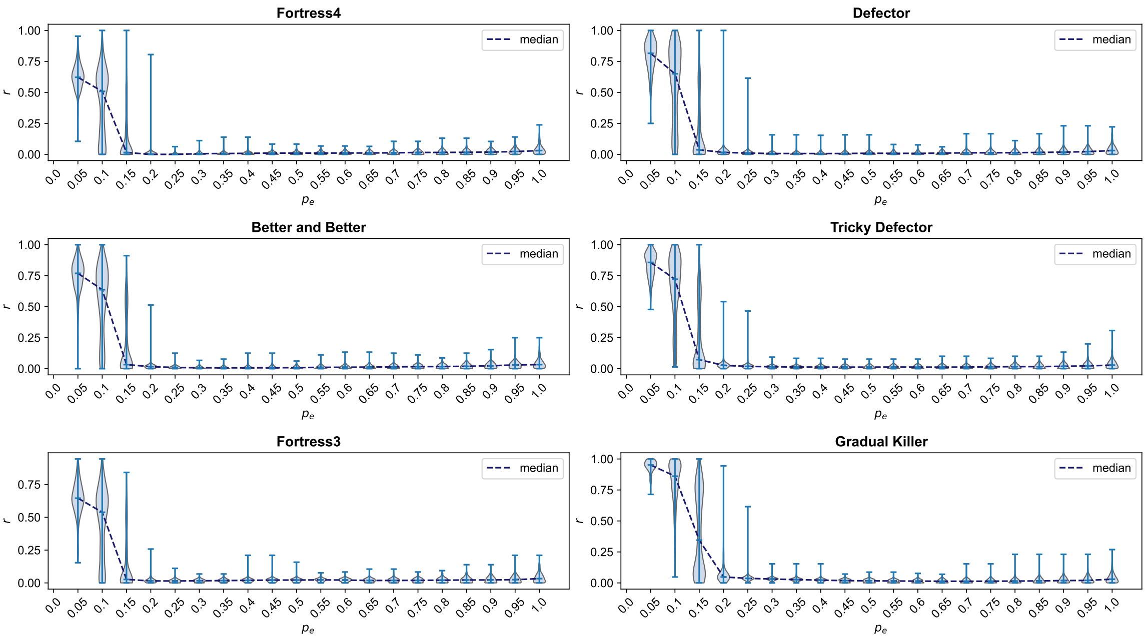


Figure 3: Normalised rank r distributions for top 6 strategies in probabilistic ending tournaments over p_e. The 6 strategies start of with a high median rank, however, their ranked decreased as the the probability of the game ending increased and at the point of $p_e = 0.1$.

The top performances in tournaments with both noise and a probabilistic ending have the largest median values compared to the top rank strategies of the other tournament types. The $\bar{r}$ for the top strategy is approximately at 0.3, indicating that the most successful strategy can on average just place in the top 30% of the competition.

On the whole, the analysis has shown that:

- In standard tournaments the dominating strategies were strategies that had been trained using reinforcement learning techniques.
- In noisy environments where the noise probability strictly less than 0.1 was considered, the successful strategies were strategies specifically designed or trained for noisy environments.
- In probabilistic ending tournaments most of the highly ranked strategies were defecting strategies and trained finite state automata, all by the authors of [8, 9]. These strategies ranked high due to their performance in tournaments where the probability of the game ending after each turn was bigger than 0.1.
- In probabilistic tournaments with p_e less than 0.1 the highly ranked strategies were strategies based on the behaviour of others.
- From the collection of strategies considered here, no strategy can be consistently successful in noisy environments, except if the value of noise is constrained to less than a 0.1.

3 The effect of strategy features on performance

The correlation coefficients between the strategies' features and the median score and the median normalised rank for the full dataset of tournaments are given by Table 3. The correlation coefficients between all features have been calculated and a graphical representation can be found in the Section 4.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending	
	r	median score	r	median score	r	median score	r	median score
CC to C rate	-0.501	0.501	0.413	-0.504	0.408	-0.323	0.260	0.023
CD to C rate	0.226	-0.199	0.457	-0.331	0.320	-0.017	0.205	-0.220
DC to C rate	0.127	-0.100	0.509	-0.504	-0.018	0.033	0.341	-0.016
DD to C rate	0.412	-0.396	0.533	-0.436	-0.103	0.176	0.378	-0.263
C_r	-0.323	0.383	0.711	-0.678	0.714	-0.832	0.579	-0.136
C_{max}	0.000	0.050	0.000	0.023	-0.000	0.046	0.000	-0.004
C_{min}	0.000	0.085	-0.000	-0.017	-0.000	0.007	-0.000	0.041
C_{median}	0.000	0.209	-0.000	0.240	0.000	0.187	0.000	0.673
C_{mean}	0.000	0.229	-0.000	0.271	0.000	0.200	-0.000	0.690
C_r / C_{max}	-0.323	0.381	0.616	-0.551	0.715	-0.833	0.536	-0.117
C_{min} / C_r	0.109	-0.080	-0.358	0.250	-0.134	0.151	-0.368	0.113
C_r / C_{median}	-0.330	0.353	0.652	-0.669	0.713	-0.852	0.330	-0.466
C_r / C_{mean}	-0.331	0.357	0.731	-0.740	0.721	-0.861	0.650	-0.621
N	-0.000	-0.009	0.000	0.002	0.000	0.003	0.000	0.001
k	-0.000	-0.002	0.000	0.002	0.000	0.001	0.000	-0.008
n	-0.000	-0.125	-0.000	-0.024	-	-	-	-
p_n	-	-	0.000	0.207	-	-	0.000	-0.650
p_e	-	-	-	-	0.000	0.165	0.000	-0.058
Make use of game	-0.003	-0.022	0.025	-0.082	-0.053	-0.108	0.013	-0.016
Make use of length	-0.158	0.124	0.005	-0.123	-0.025	-0.090	0.014	-0.016
SSE	0.473	-0.452	0.463	-0.337	-0.157	0.224	0.305	-0.259
stochastic	0.006	-0.024	0.022	-0.026	0.002	-0.130	0.021	-0.013
memory usage	-0.098	0.108	-0.009	-0.017	-	-	-	-

Table 3: Correlations between the strategies' features and the normalised rank and the median score.

In standard tournaments the features CC to C, C_r, C_r/C_{max} and the cooperating ratio compared to C_{median} and C_{mean} have a moderately negative effect on the normalised rank (smaller rank is better), and a moderate positive on the median score. The SSE error and the DD to C rate have the opposite effects. Thus, in standard tournaments behaving cooperatively corresponds to a more successful performance. Even though being nice generally pays off that does not hold against defective strategies. Being more cooperative after a mutual defection, that is not retaliating, is associated to lesser overall success in terms of normalised rank. Figure 4 confirms that the winners of standard tournaments always cooperate after a mutual cooperation and almost always defect after a mutual defection.

Compared to standard tournaments, in both noisy and in probabilistic ending tournaments the higher the rates of cooperation the lower a strategy's success and median score. A strategy would want to cooperate less than both the mean and median cooperator in such settings. In probabilistic ending tournaments the correlation coefficients have larger values, indicating a stronger effect. Thus a strategy will be punished more by its cooperative behaviour in probabilistic ending environments, supporting the results of the previous subsection as well. The distributions of the C_r of the winners in both tournaments are given by Figure 5. It confirms that the winners in noisy tournaments cooperated less than 35% of the time and in probabilistic ending tournaments less than 10%. In noisy probabilistic ending tournaments and over all the tournaments' results, the only features that had a moderate effect are $C_r/C_{\text{mean}}, C_r/C_{\text{max}}$ and C_r. In such environments cooperative behaviour appears to be punished less than in noisy and probabilistic ending tournaments.

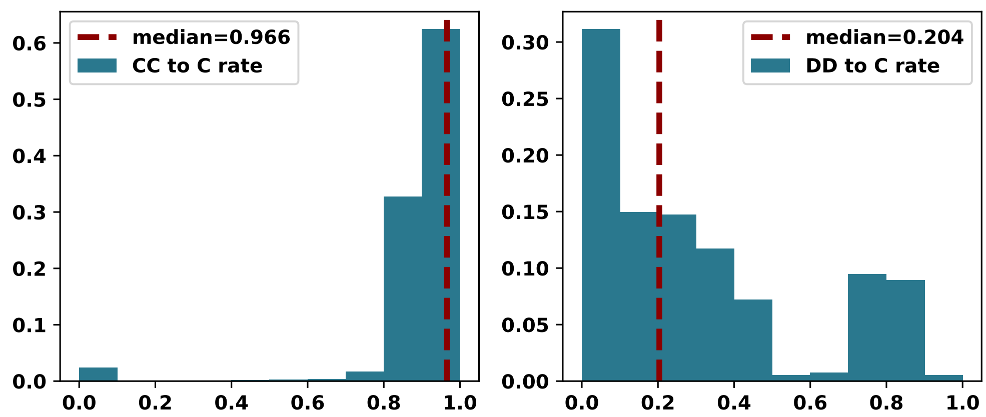


Figure 4: Distributions of CC to C and DD to C for the winners in standard tournaments.

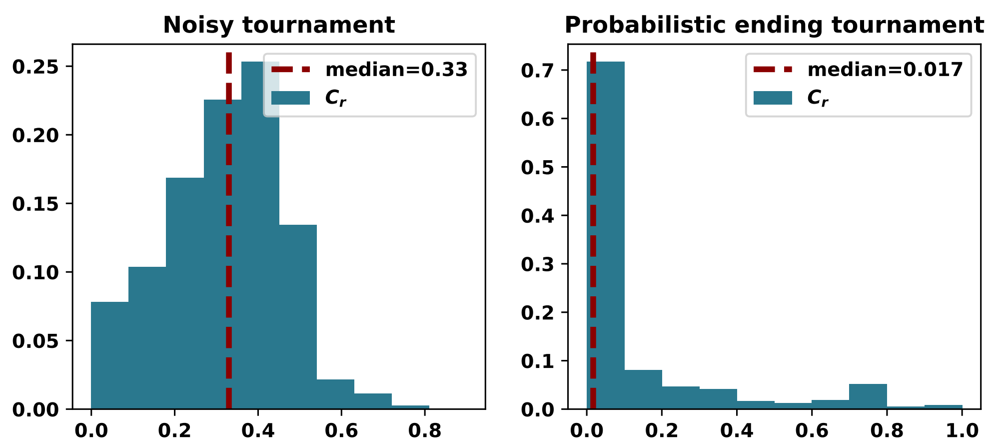


Figure 5: C_r distributions of the winners in noisy and in probabilistic ending tournaments.

A multivariate linear regression has been fitted to model the relationship between the features and the normalised rank. Based on the graphical representation of the correlation matrices given in Section 4 several of the features are highly correlated and have been removed before fitting the linear regression model. The features included are given by Table 4 alongside their corresponding p values in the distinct tournaments and their regression coefficients.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending	
	R adjusted: 0.541		R adjusted: 0.588		R adjusted: 0.587		R adjusted: 0.471	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
constant	0.695	0.000	0.443	0.0	-0.057	0.018	0.004	0.031
CC to C rate	-0.042	0.000	0.150	0.0	0.017	0.000	0.197	0.000
CD to C rate	0.297	0.000	-0.034	0.0	0.182	0.000	0.022	0.000
DC to C rate	0.198	0.000	0.064	0.0	-0.030	0.000	0.090	0.000
SSE	0.258	0.000	0.237	0.0	-0.041	0.000	0.144	0.000
C_{max}	-0.068	0.000	-	-	-0.021	0.403	-0.090	0.000
C_{min}	-0.161	0.000	1.068	0.0	-0.170	0.000	-	-
C_{mean}	0.117	0.000	-0.722	0.0	-0.024	0.000	-0.112	0.000
C_{min} / C_r	0.057	0.000	-0.544	0.0	0.125	0.000	-	-
C_r / C_{mean}	-0.468	0.000	0.272	0.0	0.525	0.000	0.403	0.000
k	0.000	0.325	0.000	0.1	0.000	0.002	0.001	0.000
n	0.000	0.000	-	-	-	-	-	-
memory usage	-0.010	0.000	0.002	0.0	-	-	-	-
p_n	-	-	-0.039	0.0	-	-	-	-
p_e	-	-	-	-	0.000	0.757	-0.149	0.000

Table 4: Results of multivariate linear regressions with r as the dependent variable. R squared is reported for each model.

A multivariate linear regression has also be fitted on the median score. The coefficients and p values of the features can be found in Section 6. This approach leads to similar conclusions.

The feature C_r/C_{mean} has a statistically significant effect across all models and a high regression coefficient. It has both a positive and negative impact on the normalised rank depending on the environment. For standard tournaments, Figure 6 gives the distributions of several features for the winners of standard tournaments. The C_r/C_{mean} distribution of the winner is also given in Figure 6. A value of $C_r/C_{\text{mean}} = 1$ implies that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament, and in standard tournaments, the median is 1. Therefore, an effective strategy in standard tournaments was the mean cooperator of its respective tournament.

The distributions of SSE and CD to C rate for the winners of standard tournaments are also given in Figure 6. The SSE distributions for the winners indicate that the strategy behaved in a ZD way in several tournaments, however, not constantly. The winners participated in matches where they did not try to extortionate their opponents. Furthermore, the CD to C distribution indicates that if a strategy were to defect against the winners the winners would reciprocate on average with a probability of 0.5.

Similarly for the rest of the different tournaments types, and the entire data set the distributions of C_r/C_{mean}, SSE and CD to C ratio are given by Figures 7, 9, 10 and 11.

Based on the C_r/C_{mean} distributions the successful strategies have adapted differently to the mean cooperator depending on the tournament type. In noisy tournaments where the median of the distribution is at 0.67,

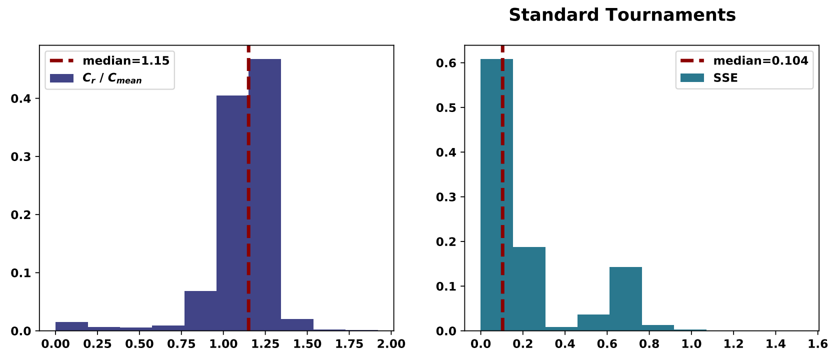


Figure 6: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of standard tournaments. A value of $C_r/C_{\text{mean}} = 1$ imply that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament. An SSE distribution skewed towards 0 indicates a extortionate behaviour by the strategy.

and thereupon the winners cooperated 67% of the time the mean cooperator did. In tournaments with noise and a probabilistic ending the winners cooperated 60%, whereas in settings that the type of the tournament can vary between all the types the winners cooperated 67% of the time the mean cooperator did. Lastly, in probabilistic ending tournaments above more defecting strategies prevail (Section **??**), and this result is reflected here.

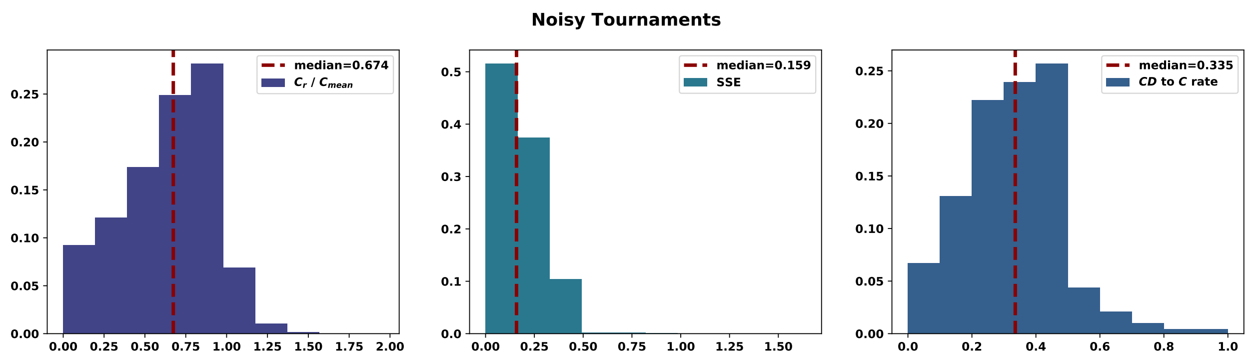
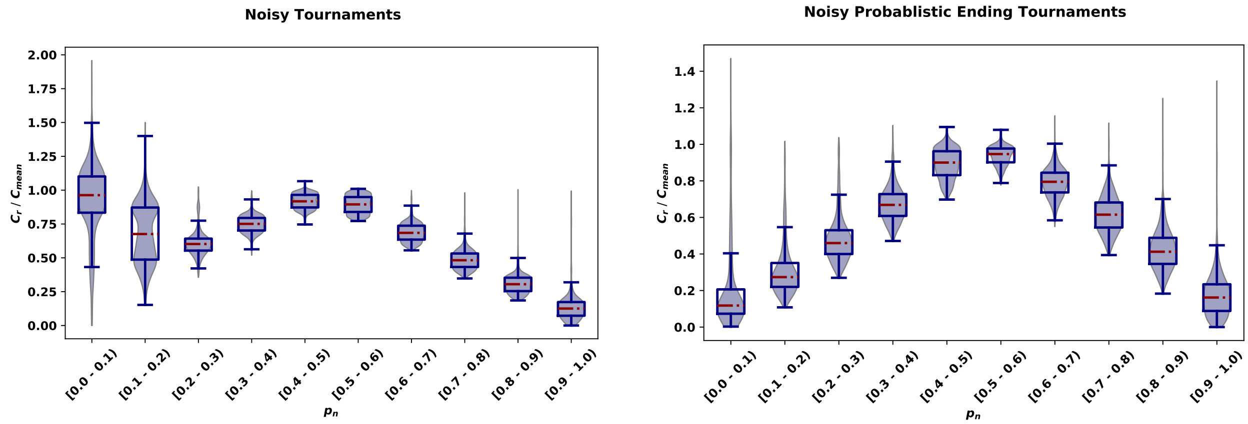


Figure 7: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of noisy tournaments.

The probability of noise has been observed to substantially affect optimal behaviour. Figure 8 gives the ratio C_r/C_{mean} for the winners in tournaments with noise, over the probability of noise. From Figure 8a it is clear that the cooperating only 67% of the time the mean cooperator did is optimal only when $p_n \in [0.2, 0.4)$ and $p_n \in [0.6, 0.7]$. In environments with $p_n < 0.1$ the winners want to be close to the mean cooperator, similarly to standard tournaments, and as the probability of noise is exceeding 0.5 (where the game is effectively inverted) strategies should aim to be less and less cooperative.

Figure 8 gives C_r/C_{mean} for the winners over p_n in tournaments with noise and a probabilistic ending. The optimal proportions of cooperations are different now that the number of turns is not fixed, successful strategies want to be more defecting that the mean cooperator, that only changes when p_n approaches 0.5. Figure 8 demonstrates how the adjustments to C_r/C_{mean} change over the noise in the to the environment, and thus supports how important adapting to the environment is for a strategy to be successful.

The distributions of the SSE across the tournament types suggest that successful strategies exhibit some



(a) C_r/C_{mean} distribution for winners in noisy tournaments over p_n.

(b) C_r/C_{mean} distribution for winners in noisy probabilistic ending tournaments over p_n.

Figure 8: C_r/C_{mean} distributions over intervals of p_n. These distributions model the optimal proportion of cooperation compared to C_{mean} as a function of (p_n).

extortionate behaviour, but not constantly. ZDs are a set of strategies that are often envious as they try to exploit their opponents. The winners of the tournaments considered in this work are envious, but not as much as many ZDs.

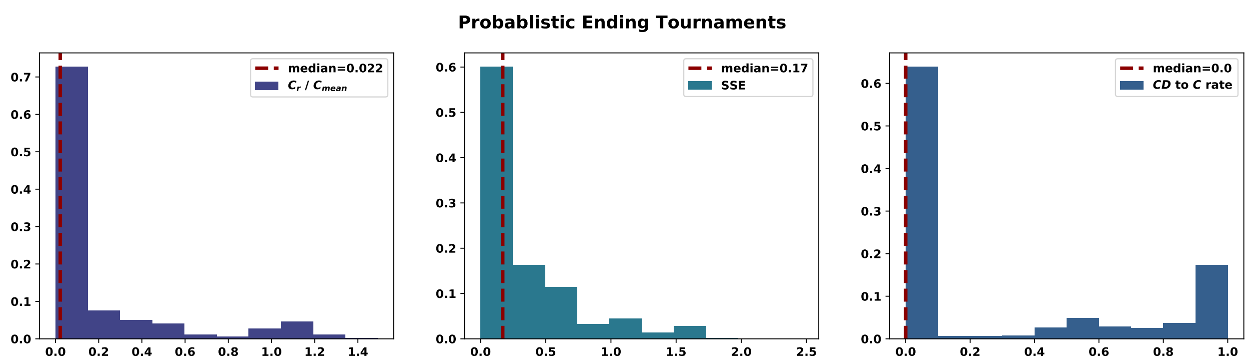


Figure 9: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of probabilistic ending tournaments.

The distributions of the CD to C rate evaluate the behaviour of a successful strategy after its opponent has defected against it. In standard tournaments it was observed that a successful strategy reciprocates with a probability of 0.5, and in a setting that the type of the tournament can vary between all the examined types a winning strategy would reciprocate on average with a probability of 0.58. In tournaments with noise a strategy is less likely to cooperate following a defection compared to standard tournaments, and in probabilistic ending tournaments a strategy will reciprocate a defection. This leads to adjusting the recommendation of being provocable to defections made by Axerlod. A strategy should be provocable in tournaments with short matches, but in the rest of the settings a strategy should be more generous.

Further statistically significant features with strong effects include C_r/C_{min}, C_r/C_{max}, C_{min} and C_{max}. These add more emphasis on how important it is for a a strategy to adapt to its environment. Finally, the features number of turns, repetitions and the probabilities of noise and the game ending had no significant effects based on the multivariate regression models.

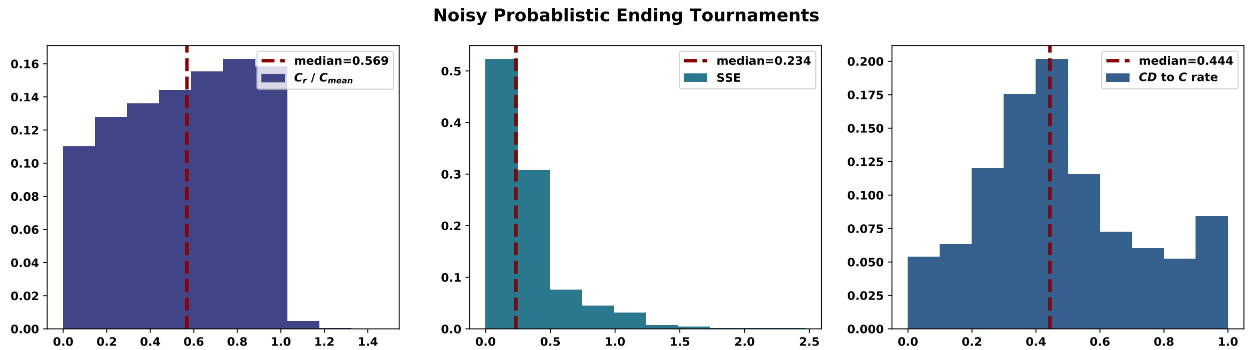


Figure 10: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of noisy probabilistic ending tournaments.

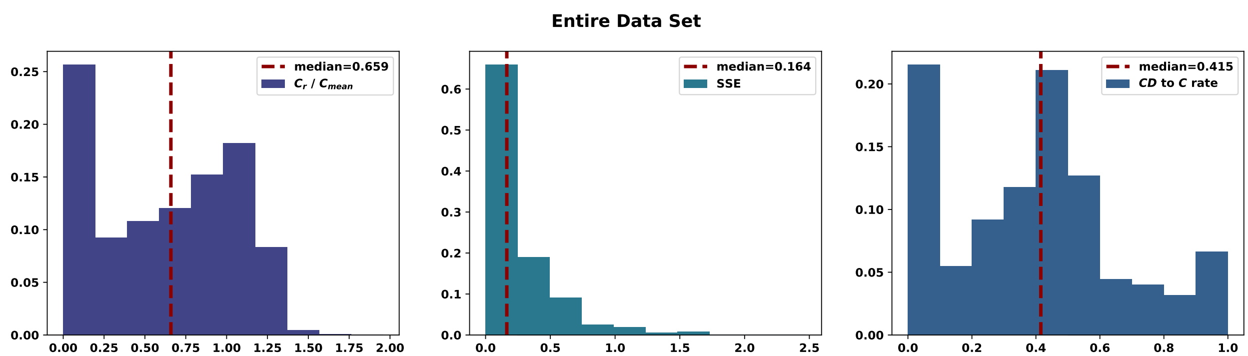


Figure 11: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners over the tournaments of the entire data set.

A third method that evaluates the importance of features using clustering and random forests can be found in the Section 5. The results uphold the outcomes of the correlation and multivariate regression. It also evaluates the effects of the whether or not a strategy is stochastic, makes use of the knowledge of the utility values, or makes use of match length. These were not evaluated by the methods above because there are binary variables. The results showed that they have no significant effect on a strategy's performance.

4 Correlation coefficients

In this section we present the correlation coefficients for the features presented in Table 3 of the main manuscript. The correlation coefficients are calculated using the Spearman's rank correlation coefficient. The results are presented as a graphical representation.

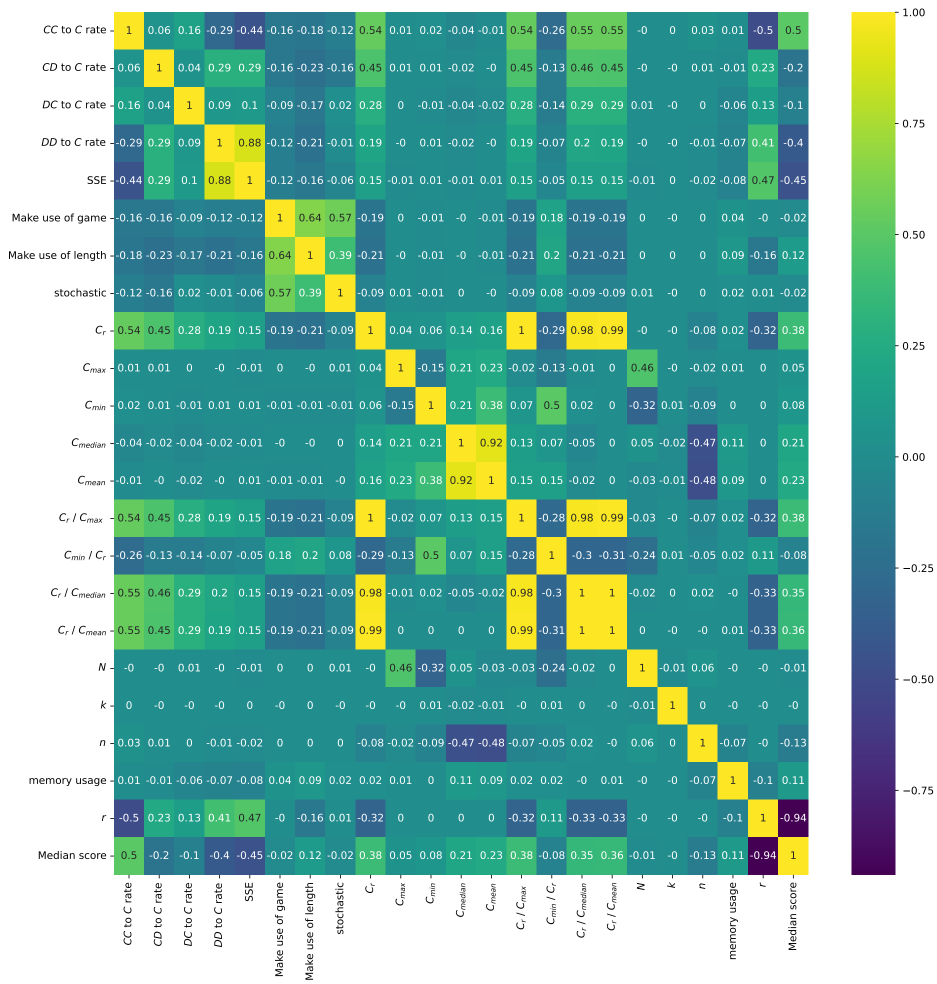


Figure 12: Correlation coefficients of strategies' features for standard tournaments.

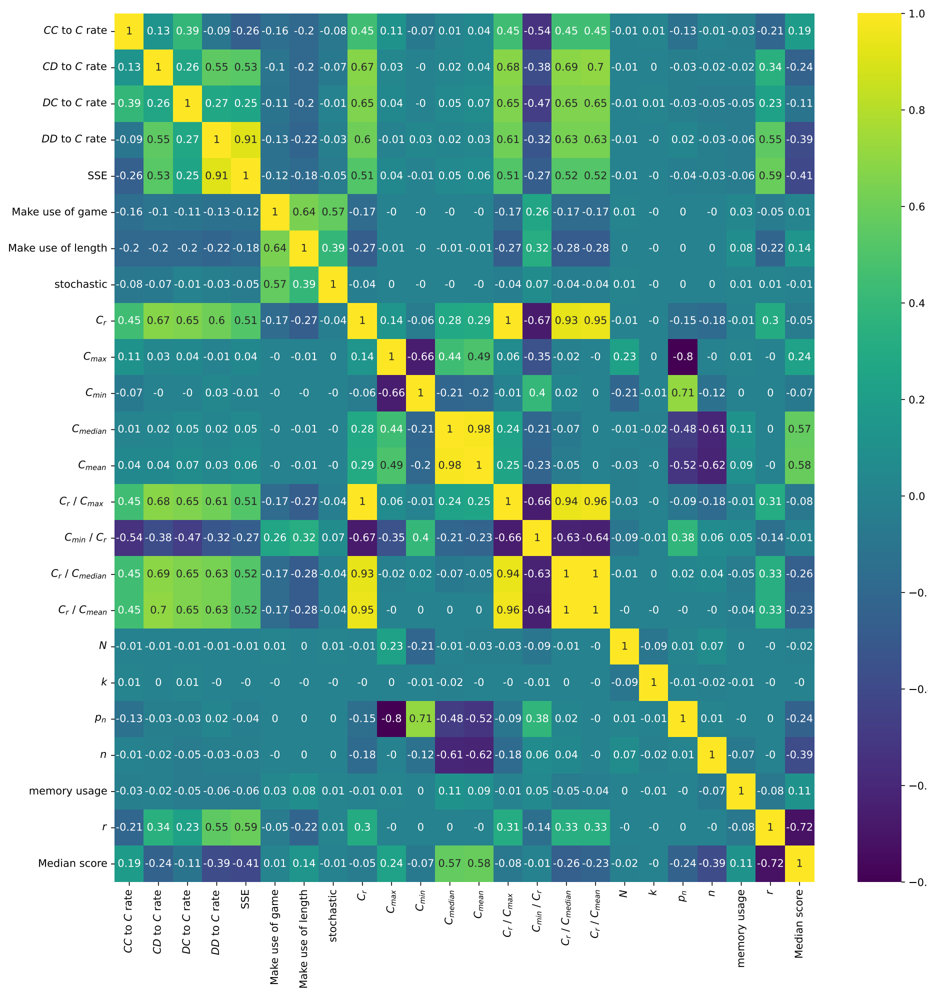


Figure 13: Correlation coefficients of strategies' features for noisy tournaments with $p_n \leq 0.1$.

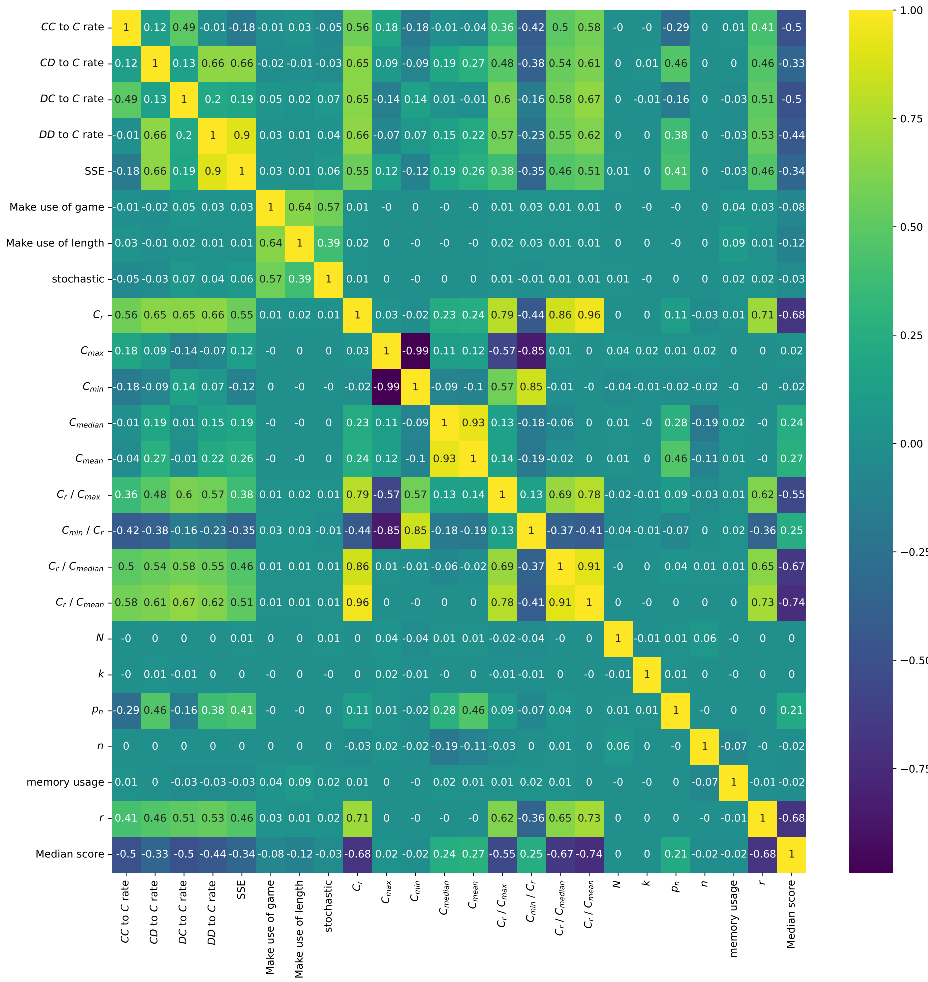


Figure 14: Correlation coefficients of strategies' features for noisy tournaments.

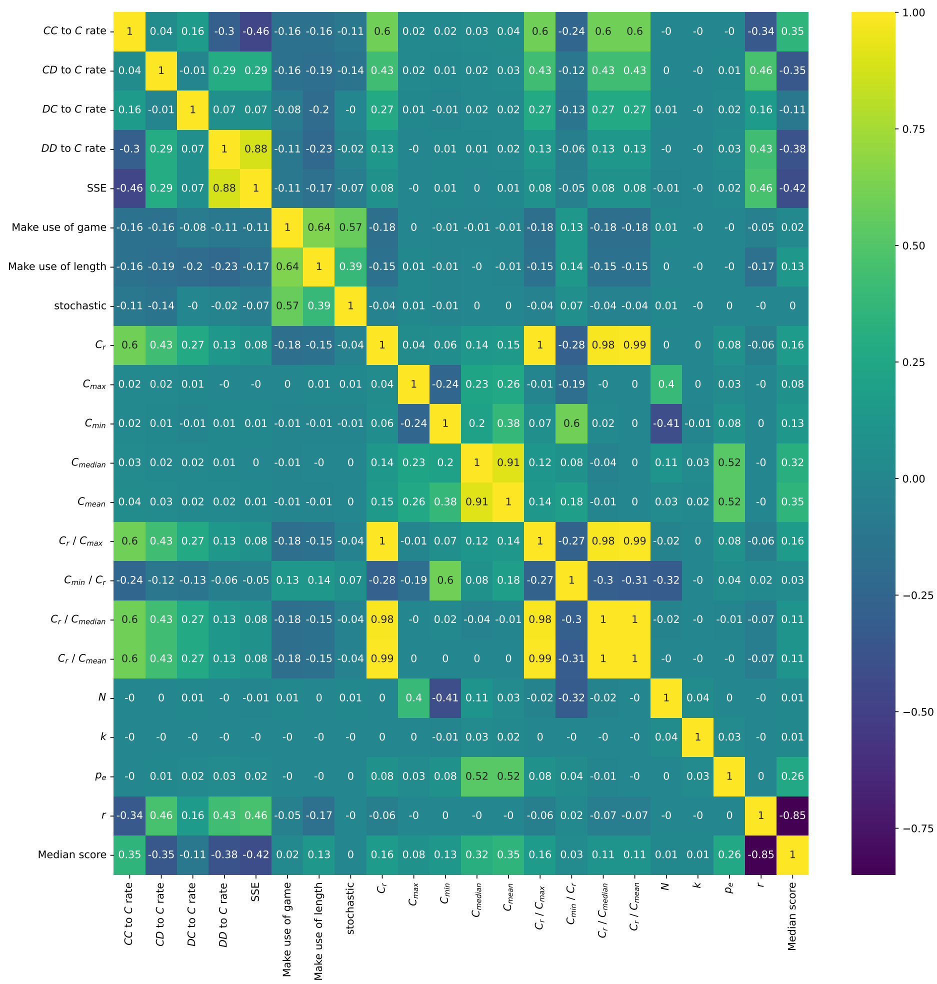


Figure 15: Correlation coefficients of strategies' features for probabilistic ending tournaments with $p_e \leq 0.1$.

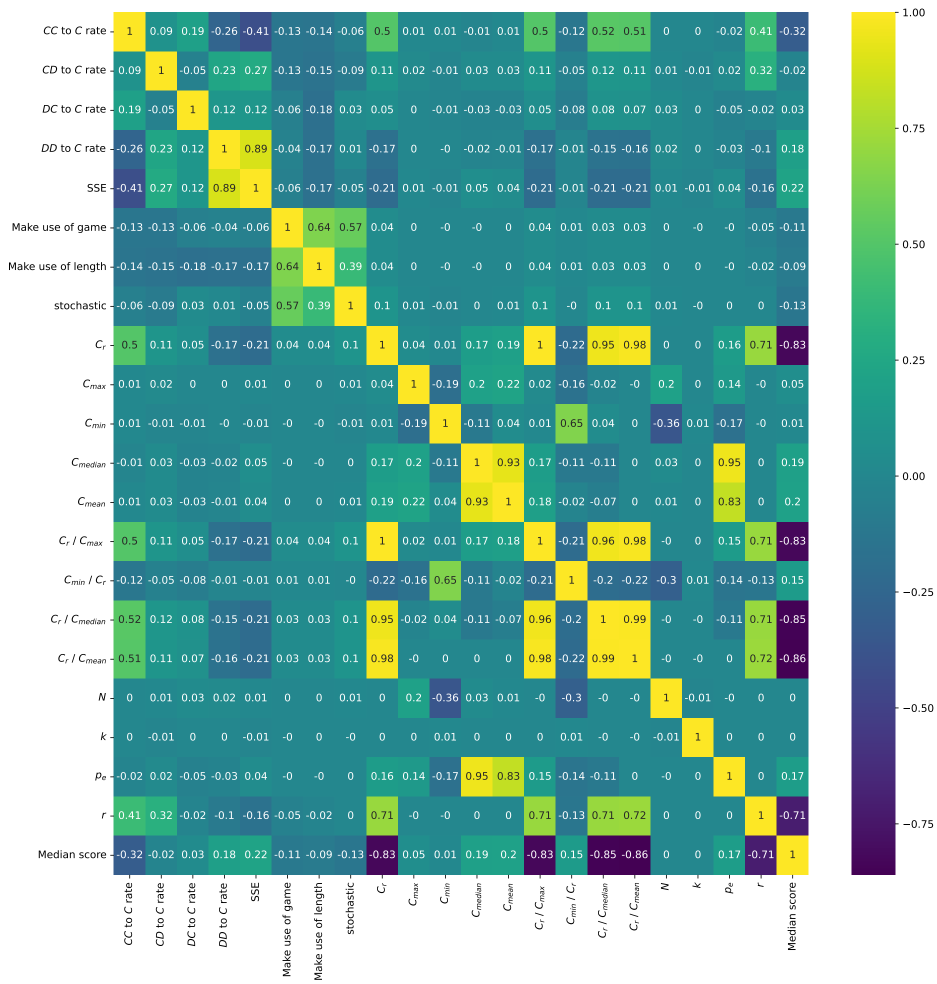


Figure 16: Correlation coefficients of strategies' features for probabilistic ending tournaments.

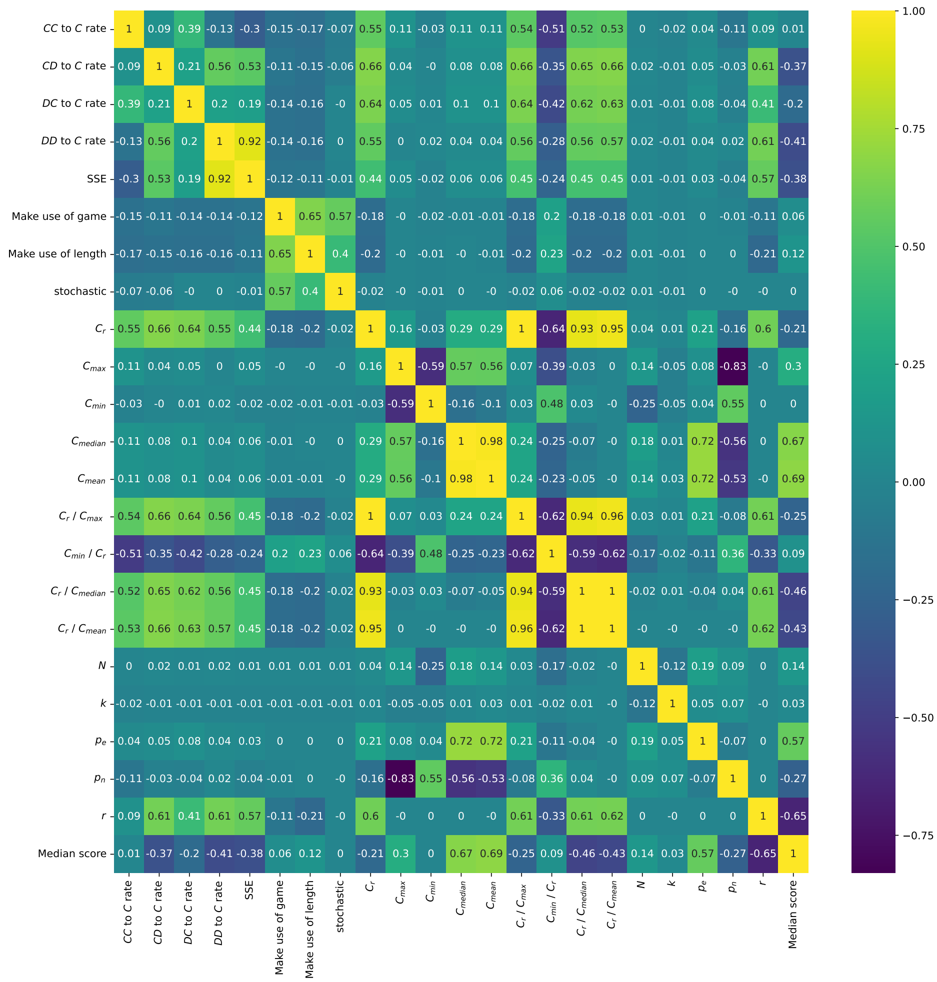


Figure 17: Correlation coefficients of strategies' features for noisy probabilistic ending tournaments with $p_n \leq 0.1$ and $p_e \leq 0.1$.

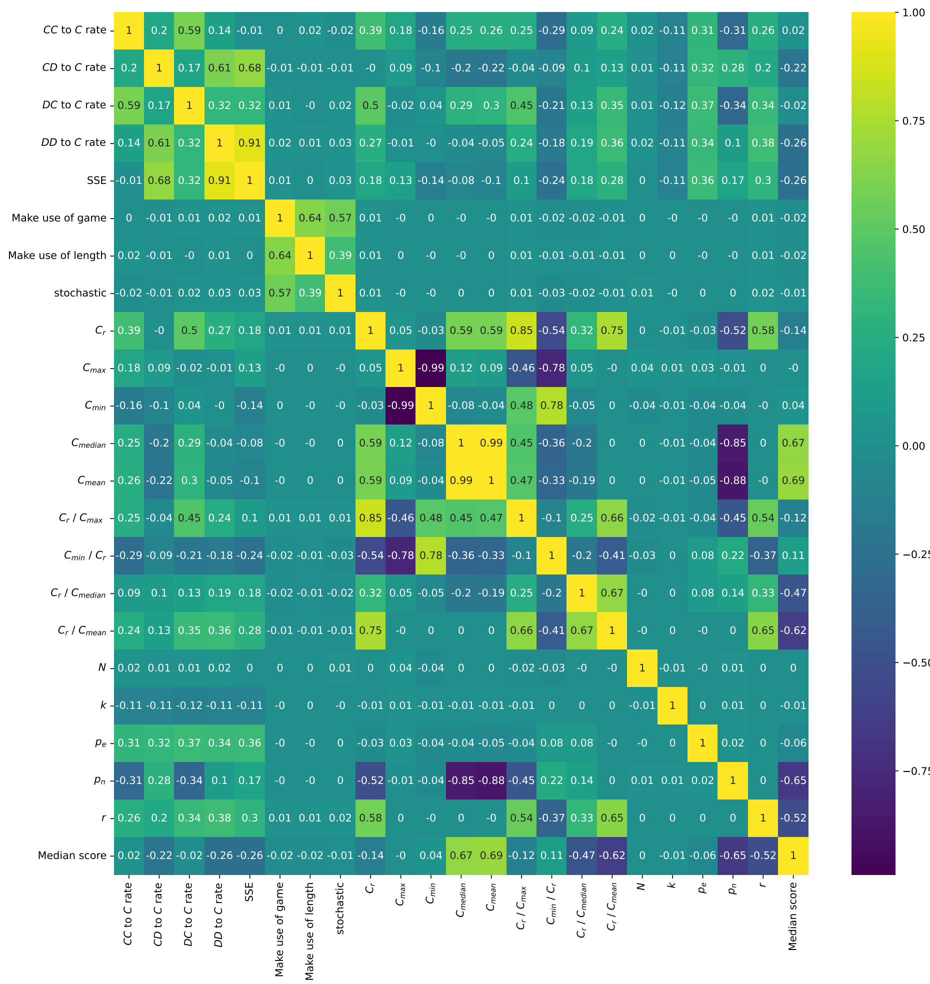


Figure 18: Correlation coefficients of strategies' features for noisy probabilistic ending tournaments.

5 Evaluation based on clustering and random forest.

The final method to evaluate the features importance in a strategy's success is a combination of a clustering task and a random forest algorithm. Initially the performances are clustered into different clusters based on them being successful or not. The performances are clustered into successful and unsuccessful clusters based on 4 different approaches. More specifically:

- **Approach 1:** The performances are divided into two clusters based on whether their performance was in the top 5% of their respective tournaments. Thus, whether r was smaller or larger than 0.05.
- **Approach 2:** The performances are divided into two clusters based on whether their performance was in the top 25% of their respective tournaments. Thus, whether r was smaller or larger than 0.25.
- **Approach 3:** The performances are divided into two clusters based on whether their performance was in the top 50% of their respective tournaments. Thus, whether r was smaller or larger than 0.50.
- **Approach 4:** The performances are clustered based on their normalised rank and their median score by a $k-$means algorithm [5]. The number of clusters is not deterministically chosen but it is based on the silhouette coefficients [40].

Once the performances have been assigned to a cluster for each approach a random forest algorithm [16] is applied. The problem is a supervised problem where the random forest algorithm predicts the cluster to which a performance has been assigned to using the features of Table **??**. The random forest models are trained on a training set of 70% of the tournaments results. The accuracy of each model based on R^2 and the number of clusters for each tournament type (because in the case of Approach 4 it is not deterministically chosen) are given by Table 5. The out of the bag error (OOB) [23] has also been calculated. The models fit well, and a high value of both the accuracy measures on the test data and the OOB error indicate that the model is not over fitting.

The importance that the features of Table **??** had on each random forest model are given by Figures 19, 20, 21, 22 and 23. These show that the classifiers stochastic, make use of game and make use of length have no significant effect, and several of the features that are highlighted by the importance are inline with the correlation results. Moreover, the smoothing parameter k appears to no have a significant effect either. The most important features based on the random forest analysis were C_r/C_{median} and C_r/C_{mean}.

6 Multivariate linear regression on median score

A multivariate linear regression has also been fitted to model the relationship between the features and the median score. The features included are given in Table 6 alongside their corresponding p values in distinct tournaments and their regression coefficients. Table 6 provides the results when considering tournaments with $p_n \leq 0.1$ and $p_e \leq 0.1$. The results when the probabilities are not constrained are given in Table 7.

7 List of strategies

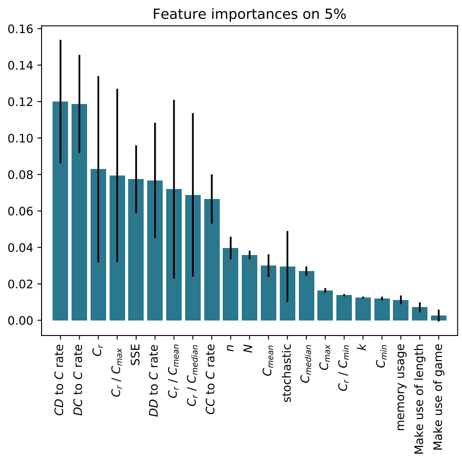
The strategies used in this study which are from Axelrod-Python library version 3.0.0.

Tournament type	Clustering Approach	Number of clusters	R^2 training data	R^2 test data	R^2 OOB score
standard	Approach 1	2	0.998831	0.987041	0.983708
	Approach 2	2	0.998643	0.978626	0.969202
	Approach 3	2	0.998417	0.985217	0.976538
	Approach 4	2	0.998794	0.990677	0.982959
noisy	Approach 1	2	0.997786	0.972229	0.968332
	Approach 2	2	0.997442	0.963254	0.955219
	Approach 3	2	0.997152	0.953164	0.940528
	Approach 4	3	0.996923	0.950728	0.935444
probabilistic ending	Approach 1	2	0.997909	0.981490	0.978120
	Approach 2	2	0.997883	0.973492	0.967150
	Approach 3	2	0.990448	0.890068	0.875822
	Approach 4	2	0.999636	0.995183	0.992809
noisy probabilistic ending	Approach 1	2	0.995347	0.957846	0.952353
	Approach 2	2	0.992813	0.909346	0.898613
	Approach 3	2	0.990579	0.824794	0.806540
	Approach 4	4	0.989465	0.841652	0.824052
over 45606 tournaments	Approach 1	2	0.997271	0.972914	0.969198
	Approach 2	2	0.996323	0.951194	0.940563
	Approach 3	2	0.993707	0.906941	0.891532
	Approach 4	3	0.993556	0.913335	0.898453

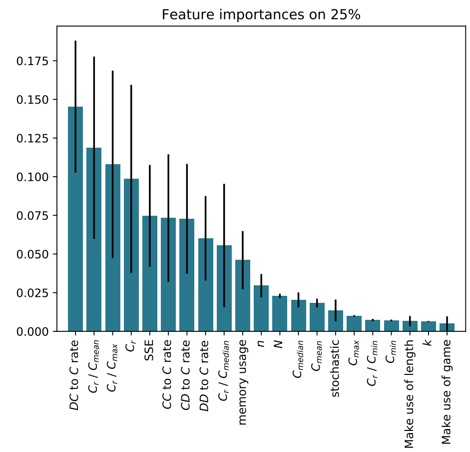
Table 5: Accuracy metrics for random forest models.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending	
	R adjusted: 0.576		R adjusted: 0.561		R adjusted: 0.488		R adjusted: 0.762	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
constant	0.928	0.000	1.082	0.000	1.259	0.000	1.642	0.000
CC to C rate	0.043	0.000	0.104	0.000	0.024	0.000	-0.032	0.000
CD to C rate	-0.325	0.000	-0.052	0.000	-0.229	0.000	-0.110	0.000
DC to C rate	-0.204	0.000	-0.076	0.000	-0.102	0.000	-0.070	0.000
SSE	-0.294	0.000	-0.186	0.000	-0.131	0.000	-0.109	0.000
C_{max}	0.056	0.000	-0.060	0.011	-0.005	0.849	-0.240	0.000
C_{min}	0.156	0.000	-0.159	0.000	0.012	0.385	0.083	0.005
C_{mean}	1.838	0.000	2.247	0.000	1.822	0.000	2.030	0.000
C_{min} / C_r	-0.049	0.000	0.040	0.000	-0.018	0.000	-0.065	0.000
C_r / C_{mean}	0.552	0.000	-0.227	0.000	0.027	0.125	-0.043	0.000
k	-0.000	0.856	-0.000	0.245	-0.000	0.610	-0.000	0.572
n	-0.000	0.000	-	-	-	-	-	-
memory usage	0.010	0.000	0.004	0.000	-	-	-	-
C_r / C_{median}	-	-	0.252	0.000	0.181	0.000	-	-
p_n	-	-	0.770	0.000	-	-	0.494	0.000
p_e	-	-	-	-	0.763	0.000	0.285	0.000

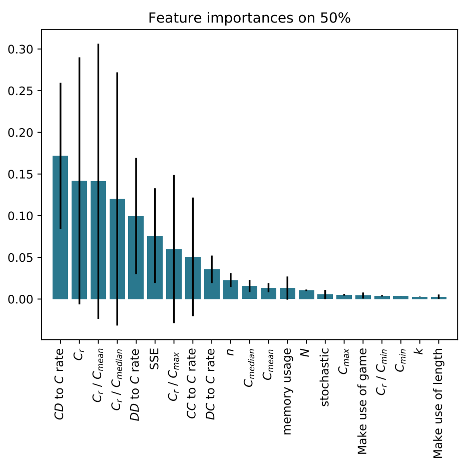
Table 6: Results of multivariate linear regressions with the median score as the dependent variable. R squared is reported for each model. For noisy tournaments $p_n \leq 0.1$ and for probabilistic ending tournaments $p_e \leq 0.1$.



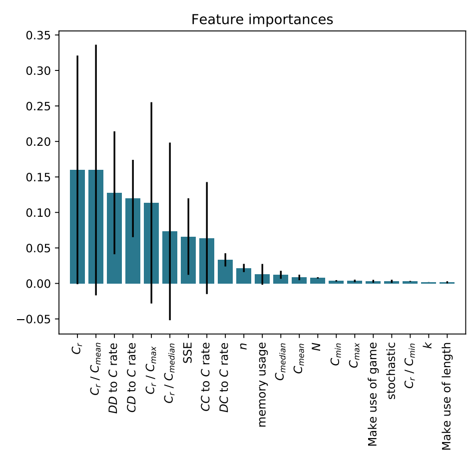
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

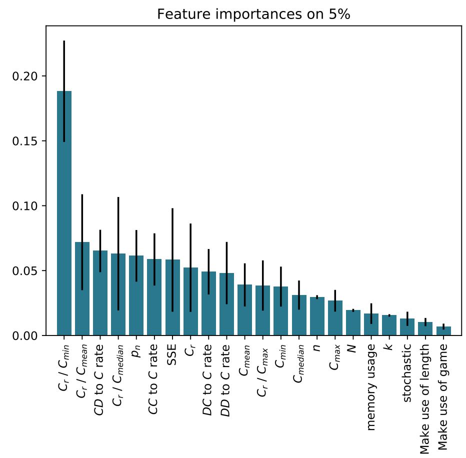


(c) Importance of features for clusters on 50% performance.

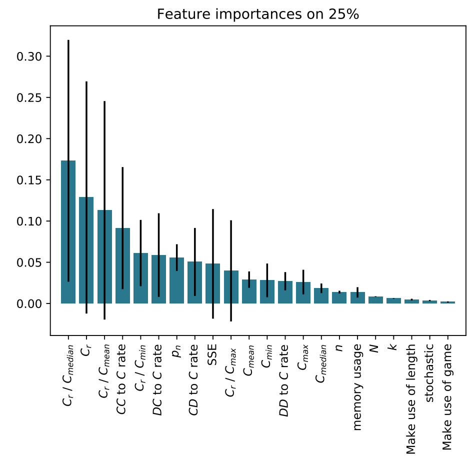


(d) Importance of features for clusters based on $k$means algorithm.

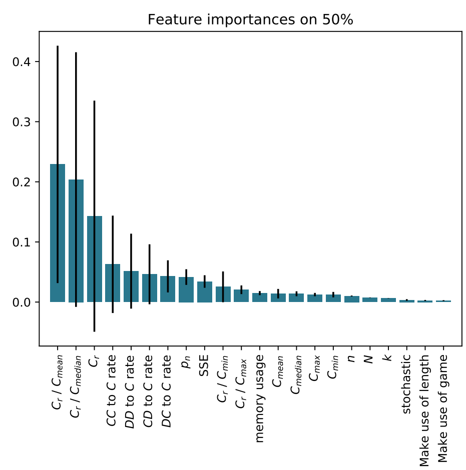
Figure 19: Importance of features in standard tournaments for different clustering methods.



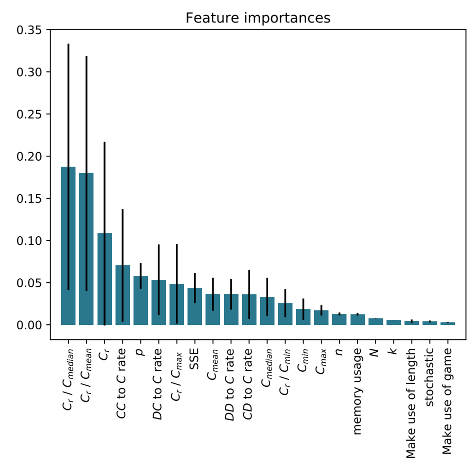
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

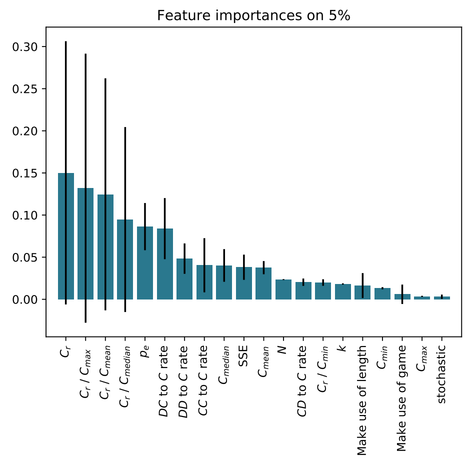


(c) Importance of features for clusters on 50% performance.

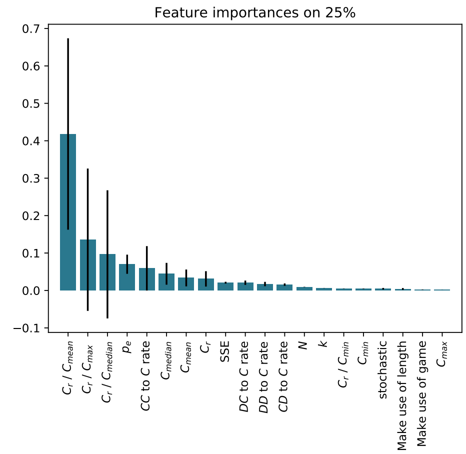


(d) Importance of features for clusters based on $k$means algorithm.

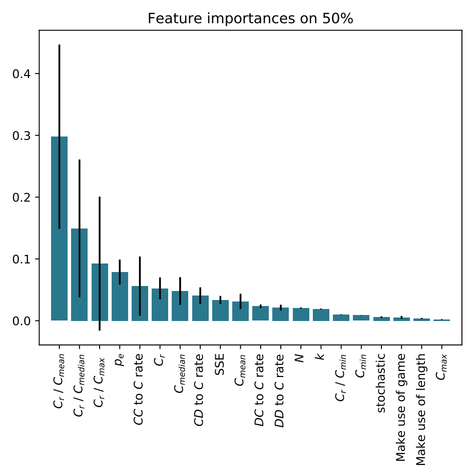
Figure 20: Importance of features in noisy tournaments for different clustering methods.



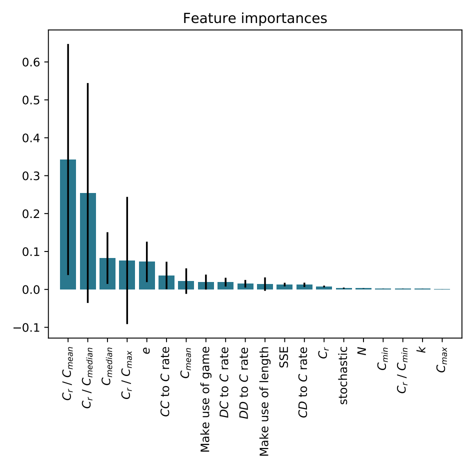
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

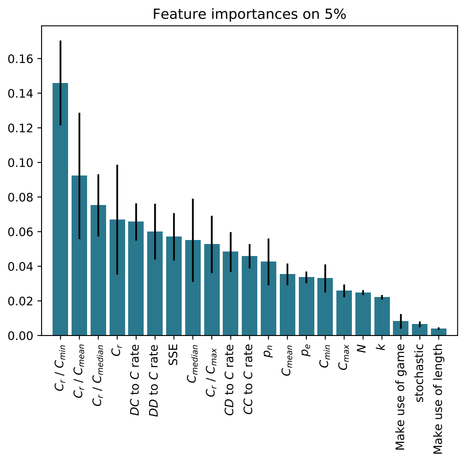


(c) Importance of features for clusters on 50% performance.

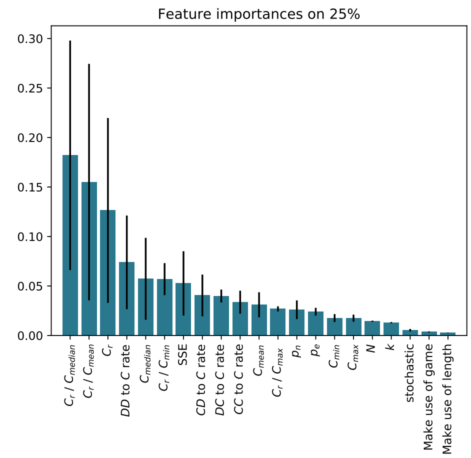


(d) Importance of features for clusters based on $k$means algorithm.

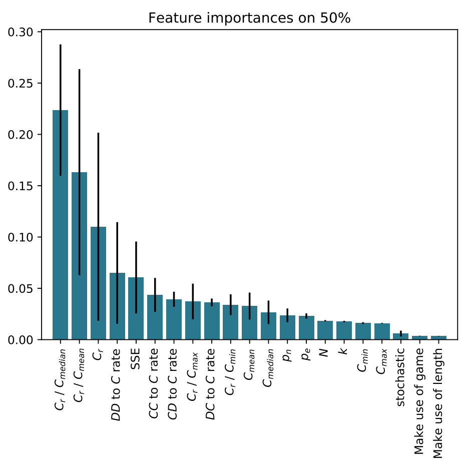
Figure 21: Importance of features in probabilistic ending tournaments for different clustering methods.



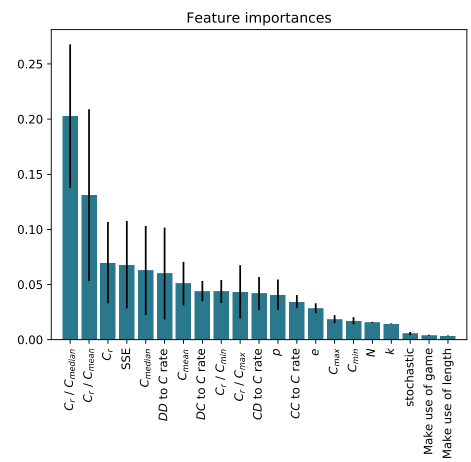
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

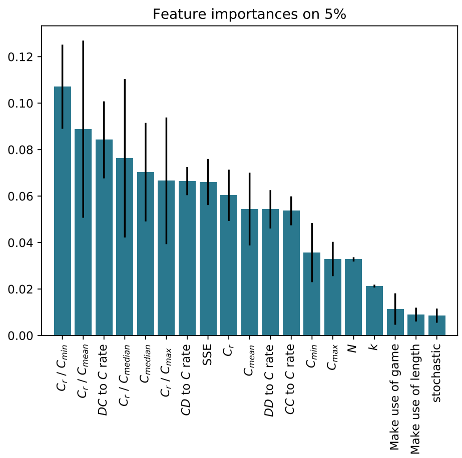


(c) Importance of features for clusters on 50% performance.

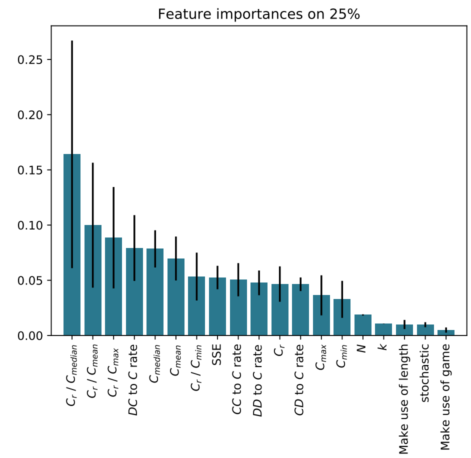


(d) Importance of features for clusters based on $k$means algorithm.

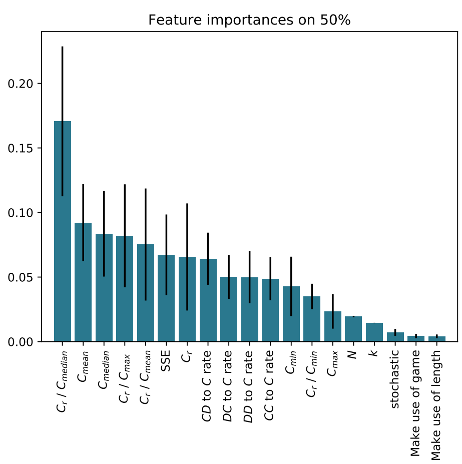
Figure 22: Importance of features in noisy probabilistic ending tournaments for different clustering methods.



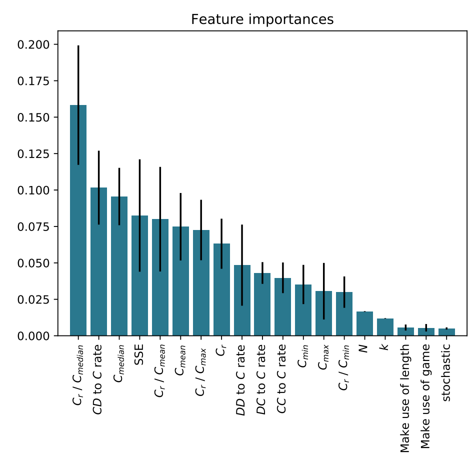
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.



(c) Importance of features for clusters on 50% performance.



(d) Importance of features for clusters based on $k$means algorithm.

Figure 23: Importance of features over all the tournaments for different clustering methods.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending	
	R adjusted: 0.575		R adjusted: 0.666		R adjusted: 0.816		R adjusted: 0.869	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
constant	0.928	0.000	2.143	0.000	2.466	0.000	1.824	0.000
CC to C rate	0.043	0.000	-0.468	0.000	0.223	0.000	-0.008	0.000
CD to C rate	-0.325	0.000	0.105	0.000	0.060	0.000	0.074	0.000
DC to C rate	-0.204	0.000	0.060	0.000	0.066	0.000	-0.002	0.011
SSE	-0.294	0.000	-0.365	0.000	0.055	0.000	-0.035	0.000
C_{max}	0.056	0.000	-	-	-0.045	0.081	-0.181	0.000
C_{min}	0.156	0.000	0.264	0.000	0.311	0.000	-	-
C_{mean}	1.838	0.000	2.046	0.000	1.506	0.000	2.273	0.000
C_{min} / C_r	-0.049	0.000	-0.252	0.000	-0.204	0.000	-	-
C_r / C_{mean}	0.552	0.000	-0.579	0.000	-1.137	0.000	-0.610	0.000
k	-0.000	0.856	0.000	0.788	0.000	0.009	0.000	0.005
n	-0.000	0.000	-	-	-	-	-	-
memory usage	0.010	0.000	-0.006	0.000	-	-	-	-
p_n	-	-	0.119	0.000	-	-	-	-
p_e	-	-	-	-	0.025	0.000	-0.028	0.000

Table 7: Results of multivariate linear regressions with the median score as the dependent variable. R squared is reported for each model.

1. ϕ [38]
2. π [38]
3. e [38]
4. ALLCorALLD [38]
5. Adaptive [29]
6. Adaptive Pavlov 2006 [25]
7. Adaptive Pavlov 2011 [29]
8. Adaptive Tit For Tat: 0.5 [42]
9. Aggravater [38]
10. Alexei [36]
11. Alternator [13, 33]
12. Alternator Hunter [38]
13. Anti Tit For Tat [24]
14. AntiCycler [38]
15. Appeaser [38]
16. Arrogant QLearner [38]
17. Average Copier [38]
18. Backstabber [38]
19. Better and Better [1]
20. Bully [34]
21. Calculator [1]
22. Cautious QLearner [38]
23. Champion [12]
24. CollectiveStrategy [30]
25. Contrite Tit For Tat [44]
26. Cooperator [13, 33, 37]
27. Cooperator Hunter [38]
28. Cycle Hunter [38]
29. Cycler CCCCCD [38]
30. Cycler CCCD [38]
31. Cycler CCCDCD [38]
32. Cycler CCD [33]
33. Cycler DC [38]
34. Cycler DDC [33]
35. DBS [10]
36. Davis [11]
37. Defector [13, 33, 37]
38. Defector Hunter [38]
39. Double Crosser [38]
40. Desperate [43]
41. DoubleResurrection [3]
42. Doubler [1]
43. Dynamic Two Tits For Tat [38]
44. EasyGo [29, 1]
45. Eatherley [12]
46. Eventual Cycle Hunter [38]
47. Evolved ANN [38]
48. Evolved ANN 5 [38]
49. Evolved ANN 5 Noise 05 [38]
50. Evolved FSM 16 [38]
51. Evolved FSM 16 Noise 05 [38]
52. Evolved FSM 4 [38]
53. Evolved HMM 5 [38]
54. EvolvedLookerUp1 1 1 [38]
55. EvolvedLookerUp2 2 2 [38]

56. Eugine Nier [36]
57. Feld [11]
58. Firm But Fair [19]
59. Fool Me Forever [38]
60. Fool Me Once [38]
61. Forgetful Fool Me Once [38]
62. Forgetful Grudger [38]
63. Forgiver [38]
64. Forgiving Tit For Tat [38]
65. Fortress3 [8]
66. Fortress4 [8]
67. GTFT [21, 35]
68. General Soft Grudger [38]
69. Gradual [15]
70. Gradual Killer [1]
71. Grofman[11]
72. Grudger [11, 14, 15, 43, 29]
73. GrudgerAlternator [1]
74. Grumpy [38]
75. Handshake [39]
76. Hard Go By Majority [33]
77. Hard Go By Majority: 10 [38]
78. Hard Go By Majority: 20 [38]
79. Hard Go By Majority: 40 [38]
80. Hard Go By Majority: 5 [38]
81. Hard Prober [1]
82. Hard Tit For 2 Tats [41]
83. Hard Tit For Tat [2]
84. Hesitant QLearner[38]
85. Hopeless [43]
86. Inverse [38]
87. Inverse Punisher [38]
88. Joss [11, 41]
89. Knowledgeable Worse and Worse [38]
90. Level Punisher [3]
91. Limited Retaliate 2 [38]
92. Limited Retaliate 3 [38]
93. Limited Retaliate [38]
94. MEM2 [31]
95. Math Constant Hunter [38]
96. Meta Hunter Aggressive [38]
97. Meta Hunter [38]
98. Meta Majority [38]
99. Meta Majority Finite Memory [38]
100. Meta Majority Long Memory [38]
101. Meta Majority Memory One [38]
102. Meta Minority [38]
103. Meta Mixer [38]
104. Meta Winner [38]
105. Meta Winner Deterministic [38]
106. Meta Winner Ensemble [38]
107. Meta Winner Finite Memory [38]
108. Meta Winner Long Memory [38]
109. Meta Winner Memory One [38]
110. Meta Winner Stochastic [38]
111. NMWE Deterministic [38]
112. NMWE Finite Memory [38]
113. NMWE Long Memory [38]
114. NMWE Memory One [38]
115. NMWE Stochastic [38]
116. Naive Prober [29]
117. Negation [2]
118. Nice Average Copier [38]
119. Nice Meta Winner [38]
120. Nice Meta Winner Ensemble [38]
121. Nydegger [11]
122. Omega TFT [25]
123. Once Bitten [38]
124. Opposite Grudger [38]
125. PSO Gambler 1 1 1 [38]
126. PSO Gambler 2 2 2 [38]
127. PSO Gambler 2 2 2 Noise 05 [38]
128. PSO Gambler Mem1 [38]
129. Predator [8]
130. Prober [29]
131. Prober 2 [1]
132. Prober 3 [1]
133. Prober 4 [1]
134. Pun1 [8]
135. Punisher [38]
136. Raider [9]
137. Random Hunter [38]
138. Random: 0.5 [11, 42]
139. Remorseful Prober [29]
140. Resurrection [3]
141. Retaliate 2 [38]
142. Retaliate 3 [38]
143. Retaliate [38]
144. Revised Downing [11]

145. Ripoff [7]
146. Risky QLearner [38]
147. SelfSteem [18]
148. ShortMem [18]
149. Shubik [11]
150. Slow Tit For Two Tats [38]
151. Slow Tit For Two Tats 2 [1]
152. Sneaky Tit For Tat [38]
153. Soft Go By Majority [13, 33]
154. Soft Go By Majority 10 [38]
155. Soft Go By Majority 20 [38]
156. Soft Go By Majority 40 [38]
157. Soft Go By Majority 5 [38]
158. Soft Grudger [29]
159. Soft Joss [1]
160. SolutionB1 [6]
161. SolutionB5 [6]
162. Spiteful Tit For Tat [1]
163. Stalker [17]
164. Stein and Rapoport [11]
165. Stochastic Cooperator [4]
166. Stochastic WSLS [38]
167. Suspicious Tit For Tat [15, 24]
168. TF1 [38]
169. TF2 [38]
170. TF3 [38]
171. Tester [12]
172. ThueMorse [38]
173. ThueMorseInverse [38]
174. Thumper [7]
175. Tit For 2 Tats (**Tf2T**) [13]
176. Tit For Tat (**TFT**) [11]
177. Tricky Cooperator [38]
178. Tricky Defector [38]
179. Tullock [11]
180. Two Tits For Tat (**2TFT**) [13]
181. VeryBad [18]
182. Willing [43]
183. Win-Shift Lose-Stay (**WShLSt**) [29]
184. Win-Stay Lose-Shift (**WSLS**) [27, 35, 41]
185. Winner12 [32]
186. Winner21 [32]
187. Worse and Worse[1]
188. Worse and Worse 2[1]
189. Worse and Worse 3[1]
190. ZD-Extort-2 v2 [28]
191. ZD-Extort-2 [41]
192. ZD-Extort-4 [38]
193. ZD-GEN-2 [28]
194. ZD-GTFT-2 [41]
195. ZD-SET-2 [28]

References

[1] Lifl (1998) prison. `http://www.lifl.fr/IPD/ipd.frame.html`. Accessed: 2017-10-23.

[2] The prisoner's dilemma. http://www.prisoners-dilemma.com/, 2017.

[3] Eckhart A. Coopsim v0.9.9 beta 6. `https://github.com/jecki/CoopSim/`, 2015.

[4] C. Adami and A. Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature communications*, 4(1):2193, 2013.

[5] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[6] D. Ashlock, J. A. Brown, and P. Hingston. Multiple opponent optimization of prisoner's dilemma playing agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1):53–65, 2015.

[7] D. Ashlock and E. Y. Kim. Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies. *IEEE Transactions on Evolutionary Computation*, 12(5):647–659, 2008.

[8] W. Ashlock and D. Ashlock. Changes in prisoner's dilemma strategies over evolutionary time with different population sizes. In *2006 IEEE International Conference on Evolutionary Computation*, pages 297–304. IEEE, 2006.

[9] W. Ashlock, J. Tsang, and D. Ashlock. The evolution of exploitation. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 135–142. IEEE, 2014.

[10] T. C. Au and D. Nau. Accident or intention: that is the question (in the noisy iterated prisoner's dilemma). In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 561–568. ACM, 2006.

[11] R. Axelrod. Effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, 24(1):3–25, 1980.

[12] R. Axelrod. More effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, 24(3):379–403, 1980.

[13] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.

[14] J. S. Banks and R. K. Sundaram. Repeated games, finite automata, and complexity. *Games and Economic Behavior*, 2(2):97–117, 1990.

[15] B. Beaufils, J. P. Delahaye, and P. Mathieu. Our meeting with gradual, a good strategy for the iterated prisoner's dilemma. In *Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 202–209, 1997.

[16] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[17] A. Carvalho, H. Rocha, F. Amaral, and F. Guimaraes. Iterated prisoner's dilemma-an extended analysis. *Iterated Prisoner's Dilemma-An extended analysis*, 2013.

[18] A. Carvalho, H. P. Rocha, F. T. Amaral, and F. G. Guimaraes. Iterated prisoner's dilemma-an extended analysis. 2013.

[19] M. R. Frean. The prisoner's dilemma without synchrony. *Proceedings of the Royal Society of London B: Biological Sciences*, 257(1348):75–79, 1994.

[20] D. Fudenberg and E. Maskin. The folk theorem in repeated games with discounting or with incomplete information. In *A Long-Run Collaboration On Long-Run Games*, pages 209–230. World Scientific, 2009.

[21] M. Gaudesi, E. Piccolo, G. Squillero, and A. Tonda. Exploiting evolutionary modeling to prevail in iterated prisoner's dilemma tournaments. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):288–300, 2016.

[22] M. Harper, V. Knight, M. Jones, G. Koutsovoulos, N. E. Glynatsi, and O. Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PloS one*, 12(12):e0188046, 2017.

[23] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[24] C. Hilbe, M. A. Nowak, and A. Traulsen. Adaptive dynamics of extortion and compliance. *PloS one*, 8(11):e77886, 2013.

[25] G. Kendall, X. Yao, and S. Y. Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

[26] V. A. Knight, M. Harper, N. E. Glynatsi, and J. Gillard. Recognising and evaluating the effectiveness of extortion in the iterated prisoner's dilemma. *CoRR*, abs/1904.00973, 2019.

[27] D. Kraines and V. Kraines. Pavlov and the prisoner's dilemma. *Theory and decision*, 26(1):47–79, 1989.

[28] S. Kuhn. Prisoner's dilemma. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2017 edition, 2017.

[29] J. Li, P. Hingston, S. Member, and G. Kendall. Engineering Design of Strategies for Winning Iterated Prisoner ' s Dilemma Competitions. 3(4):348–360, 2011.

[30] J. Li and G. Kendall. A strategy with novel evolutionary features for the iterated prisoner's dilemma. *Evolutionary Computation*, 17(2):257–274, 2009.

[31] J. Li, G. Kendall, and S. Member. The effect of memory size on the evolutionary stability of strategies in iterated prisoner ' s dilemma. X(X):1–8, 2014.

[32] P. Mathieu and J. P. Delahaye. New winning strategies for the iterated prisoner's dilemma. *Journal of Artificial Societies and Social Simulation*, 20(4):12, 2017.

[33] S. Mittal and K. Deb. Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 13(3):554–565, 2009.

[34] J. H. Nachbar. Evolution in the finitely repeated prisoner's dilemma. *Journal of Economic Behavior & Organization*, 19(3):307–326, 1992.

[35] M. Nowak and K. Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56, 1993.

[36] prase. Prisoner's dilemma tournament results. `https://www.lesswrong.com/posts/hamma4XgeNrsvAJv5/prisoner-s-dilemma-tournament-results`, 2011.

[37] 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.

[38] The Axelrod project developers. Axelrod: 3.0.0. `http://dx.doi.org/10.5281/zenodo.807699`, April 2016.

[39] A. J. Robson. Efficiency in evolutionary games: Darwin, nash and the secret handshake. *Journal of theoretical Biology*, 144(3):379–396, 1990.

[40] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[41] A. J. Stewart and J. B. Plotkin. Extortion and cooperation in the prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(26):10134–10135, 2012.

[42] E. Tzafestas. Toward adaptive cooperative behavior. *From Animals to animals: Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior (SAB-2000)*, 2:334–340, 2000.

[43] P. Van den Berg and F. J. Weissing. The importance of mechanisms for the evolution of cooperation. *Proceedings of the Royal Society B: Biological Sciences*, 282(1813):20151382, 2015.

[44] J. Wu and R. Axelrod. How to cope with noise in the iterated prisoner's dilemma. *Journal of Conflict resolution*, 39(1):183–189, 1995.