

Improving Generalization Performance in Co-evolutionary Learning

Siang Yew Chong, Peter Tiño, Day Chyi Ku[†], and Xin Yao

The Centre of Excellence for Research in
Computational Intelligence and Applications (CERCIA)

School of Computer Science

The University of Birmingham, UK.

E-mails: S.Y.Chong@cs.bham.ac.uk, P.Tino@cs.bham.ac.uk, X.Yao@cs.bham.ac.uk

[†]Faculty of Information Technology

Multimedia University, Malaysia.

E-mail: dcku@mmu.edu.my

Abstract

Recently, the generalization framework in co-evolutionary learning has been theoretically formulated and demonstrated in the context of game-playing. Generalization performance of a strategy (solution) is estimated using a collection of random test strategies (test cases) by taking the average game outcomes, with confidence bounds provided by Chebyshev's Theorem. Chebyshev's bounds have the advantage that they hold for any distribution of game outcomes. However, such a distribution-free framework leads to unnecessarily loose confidence bounds. In this contribution, we take advantage of the near-Gaussian nature of average game outcomes and provide tighter bounds based on parametric testing. This enables us to use small samples of test strategies to guide and improve the co-evolutionary search. We demonstrate our approach in a series of empirical studies involving the iterated prisoner's dilemma and the more complex Othello game in a competitive co-evolutionary learning setting. The new approach is shown to improve on the classical co-evolutionary learning in that we obtain increasingly higher generalization performance using relatively small samples of test strategies. This is achieved without large performance fluctuations typical of the classical approach. The new approach also leads to faster co-evolutionary search where we can strictly control the condition (sample sizes) under which the speed-up is achieved (not at the cost of weakening precision in the estimates).

Keywords: Co-evolutionary learning, generalization, evolutionary computation, machine learning, iterated prisoner's dilemma, Othello.

1 Introduction

Co-evolutionary learning refers to a broad class of population-based, stochastic search algorithms that involves the simultaneous evolution of competing solutions with coupled fitness [1]. The co-evolutionary search process is characterized by the adaptation of solutions in some form of representation involving repeated applications of variation and selection [2]. Co-evolutionary learning offers an attractive alternative approach for problem solving in cases where obtaining an absolute quality measurement to guide the search for solutions is difficult or not possible. One such problem is game-playing [3, 4, 5, 6, 7, 8]. Unlike classical machine learning that requires an absolute quality measurement, the search process in co-evolutionary learning can be guided by strategic interactions between competing solutions (learners). Early studies [9, 10, 11] have further argued that the co-evolutionary search may benefit from these strategic interactions from one generation to the next that results in an *arms race* of increasingly innovative solutions.

Generalization is one of the main research issues in co-evolutionary learning. Recently, we have formulated a theoretical framework for generalization in co-evolutionary learning [12]. Other past studies such as [13] has investigated an approach to analyze performance in co-evolving populations through non-local adaptation. A general framework for statistical comparison of performance of evolutionary algorithms (EAs) has been recently formulated [14]. In line with these studies, the generalization framework offers a rigorous approach to performance analysis of co-evolutionary learning, whether for individual co-evolved solutions, or for the population of co-evolved solutions in any generation.

We have demonstrated the generalization framework in the context of game-playing. Generalization performance of a strategy (solution) is estimated using a collection of random test strategies (test cases) by taking the average game outcomes, with confidence bounds provided by Chebyshev’s Theorem [15]. Chebyshev’s bounds have the advantage that they hold for any distribution of game outcomes. However, such a distribution-free framework leads to unnecessarily loose confidence bounds. In this paper, we take advantage of the near-Gaussian nature of average game outcomes through the Central Limit Theorem [16] and provide tighter bounds based on parametric testing. Furthermore, we can strictly control the condition (i.e., sample size under a given precision) under which the distribution of average game outcomes (generalization performance estimates) converges to a Gaussian through the Berry-Esseen Theorem [17].

These improvements to the generalization framework now provide the means with which we can develop a general and principled approach to improve generalization performance in co-evolutionary learning that can be implemented efficiently. Ideally, if we compute the true generalization performance of any co-evolving solution and directly use it as the fitness measure, co-evolutionary learning would lead to the search for solutions with higher generalization performance. However, direct estimation of the true generalization performance using the distribution-free framework can be computationally expensive [12]. Our new theoretical contributions that exploit the near-Gaussian nature of generalization estimates allow us: (1) to find out in a principled manner the required number of test cases for robust estimations (given a controlled level of precision) of generalization performance, and (2) to subsequently use small sample of random test cases sufficient for robust estimations to compute generalization estimates of solutions directly as the fitness measure to guide and improve co-evolutionary learning.

Early studies [18, 19, 20] have shown that the classical co-evolutionary learning approach that uses relative fitness (i.e., fitness evaluation that depends on other competing solutions in the population) does not necessarily lead to solutions with increasingly higher generalization performance. Others have investigated various approaches to improve performance in co-evolutionary learning, e.g., by exploiting diversity in the population [21, 22, 23], using other notions for fitness measure such as pareto dominance [24], and using archives of test cases [25, 26, 27] among others. In [28], a study has been made to investigate how performance can be improved in a cooperative co-evolutionary learning framework (a population member only represents part of a complete solution) as compared to most other studies that have considered the competitive co-evolutionary learning framework (a population member represents a complete solution).

Unlike these past studies, we demonstrate an approach to improve generalization performance in a principled manner that can be implemented as an efficient algorithm (e.g., using small samples of test cases) that is verified in a principled manner as well. Our approach directly uses generalization estimates as the fitness measure in a competitive co-evolutionary learning setting. A series of empirical studies involving the iterated prisoner’s dilemma (IPD) and the more complex Othello game is used to demonstrate how the new approach improves on the classical approach in that evolved strategies with increasingly higher generalization performance are obtained using relatively small samples of test strategies. This is achieved without large performance fluctuations typical of the classical approach. The new approach also leads to faster co-evolutionary search where we can strictly control the condition (sample sizes) under which the speed-up is achieved (not at the cost of weakening precision in the estimates). It is faster than using the distribution-free framework (it requires an order of magnitude smaller number of test strategies) to achieve similarly high generalization performance.

More importantly, our approach does not depend on the complexity of the game. For some games that

are more complex (under some measures of game complexity), more test strategies may be required to estimate the generalization performance of a strategy for a given level of precision. However, this will come out automatically and in a principled manner from our analysis. The necessary sample size for robust estimations can then be set and subsequently, generalization estimates can be computed and directly used as the fitness measure to guide co-evolutionary search of strategies with higher generalization performance.

We note that this study is a first step towards understanding and developing theoretically motivated frameworks of co-evolutionary learning that can lead to improvements in the generalization performance of solutions. Although our generalization framework makes no assumption on the underlying distribution of test cases, we demonstrate one application where the underlying distribution in the generalization measure is fixed and known *a-priori*. Generalization estimates are directly used as the fitness measure to improve generalization performance in co-evolutionary learning. This has the effect reformulating co-evolutionary learning to that of an evolutionary learning approach, but with the advantage of a principled and efficient methodology that has the potential of outperforming classical co-evolutionary approach on difficult learning problems such as games. Further studies may involve extending the generalization framework in formulating co-evolutionary learning systems where the population acting as test samples can adapt to approximate a particular distribution that solutions should generalize to.

The rest of this paper is organized as follows. Section 2 presents the theoretical framework for statistical estimation of generalization performance, and improvements made to provide tighter bounds through the Central Limit and Berry-Esseen Theorems. We mention two kinds of parametric testing: (a) making statistical claims on the hypothesized performance of a strategy and (b) comparing performance differences of a pair of strategies. Section 3 demonstrates how one can find out and set the required number of test strategies for robust estimation of generalization performance in a principled manner, using the IPD game for illustration. It is shown that a small number of test strategies is sufficient to estimate generalization performance with good accuracy. Section 4 demonstrates how generalization estimates can be used directly as the fitness measure to improve co-evolutionary learning. We first illustrate the new co-evolutionary approach using the IPD game and later consider the more complex Othello game. Finally, Section 5 concludes the paper with some remarks for future studies.

2 Statistical Estimation of Generalization Performance in Co-evolutionary Learning

2.1 Games

In co-evolutionary learning, the quality of a solution is determined relative to other competing solutions in the population through interactions. This can be framed in the context of game-playing, i.e., an interaction is a game played between two strategies (solutions) [12]. We assume that there is a potentially vast but finite set of possible strategies that can be involved in playing the game. At each time step, a strategy can select a move from a finite set of possible moves to play the game. Endowing strategies with memory of their own and opponents' moves results in an exponential explosion in the number of such strategies.

Consider a game and a set \mathcal{S} of M distinct strategies, $\mathcal{S} = \{1, 2, \dots, M\}$. Denote the game outcome of strategy i playing against the opponent strategy j by $G_i(j)$. Different definitions of $G_i(j)$ (different notions of game outcomes) for the generalization performance indicate different measures of quality [12]. The win-lose function for $G_i(j)$ is given by:

$$G_W(i, j) = \begin{cases} C_{\text{WIN}}, & \text{if } g(i, j) > g(j, i), \\ C_{\text{LOSE}}, & \text{otherwise,} \end{cases} \quad (1)$$

where $g(i, j)$ and $g(j, i)$ are payoffs for strategies i and j at the end of the game, respectively, and

C_{WIN} and C_{LOSE} are arbitrary constants with $C_{\text{WIN}} > C_{\text{LOSE}}$. We use $C_{\text{WIN}} = 1$ and $C_{\text{LOSE}} = 0$ and arbitrarily choose a stricter form of $G_W(i, j)$ (a loss is awarded to both sides in the case of a tie). As an example that we present later, our choice of $G_W(i, j)$ is to simplify analysis. In this case, the “all defect” strategy that plays full defection is known to be the only one with the maximum generalization performance when $G_W(i, j)$ is used irrespective of how test strategies are distributed in \mathcal{S} for the IPD game. This is not necessarily true for other definitions, such as the average-payoff function.

2.2 Estimating Generalization Performance

A-priori some strategies may be favored over the others, or all strategies can be considered with equal probability. Let the selection of individual test strategies from \mathcal{S} be represented by a random variable J taking on values $j \in \mathcal{S}$ with probability $P_S(j)$. G_i is the true generalization performance of a strategy i and is defined as the mean performance (game outcome) against all possible test strategies j :

$$G_i = \sum_{j=1}^M P_S(j) G_i(j). \quad (2)$$

In other words, G_i is the mean of the random variable $G_i(J)$,

$$G_i = E_{P_S}[G_i(J)].$$

In particular, when all strategies are equally likely to be selected as test strategies, i.e., when P_S is uniform, we have

$$G_i = \frac{1}{M} \sum_{j=1}^M G_i(j). \quad (3)$$

The size M of the strategy space \mathcal{S} can be vary large, making direct estimation of G_i , $i \in \mathcal{S}$, through Equation 2 infeasible. In practice, one can estimate G_i through a random sample S_N of N test strategies drawn i.i.d. from \mathcal{S} with probability P_S . The estimated generalization performance of strategy i is denoted by $\hat{G}_i(S_N)$ and given as follows:

$$\hat{G}_i(S_N) = \frac{1}{N} \sum_{j \in S_N} G_i(j). \quad (4)$$

If the game outcome $G_i(J)$ varies within a finite interval $[G_{\text{MIN}}, G_{\text{MAX}}]$ of size R , the variance of $G_i(J)$ is upper-bounded by $\sigma_{\text{MAX}}^2 = (G_{\text{MAX}} - G_{\text{MIN}})^2/4 = R^2/4$. Using Chebyshev’s Theorem [15], we obtain

$$P(|\hat{G}_i - G_i| \geq \epsilon) \leq \frac{R^2}{4N \cdot \epsilon^2} \quad (5)$$

for any positive number $\epsilon > 0$. Note that Chebyshev’s bounds (Equation 5) are distribution-free, i.e., no particular form of distribution of $G_i(J)$ is assumed. One can make statistical claims of how confident one is on the accuracy of an estimate given a random test sample of a known size N using Chebyshev’s bounds [12].

2.3 Error Estimations for Gaussian-distributed Generalization Estimates

Selection of the sample \mathcal{S}_N of test strategies can be formalized through a random variable \mathcal{S}_N on \mathcal{S}^N endowed with the product measure induced by P_S . Estimates of the generalization performance of strategy i can be viewed as realizations of the random variable $\hat{G}_i(\mathcal{S}_N)$. Since game outcomes $G_i(J)$ have finite mean and variance, by the Central Limit Theorem, for large enough N , $\hat{G}_i(\mathcal{S}_N)$ is Gaussian-distributed. Claims regarding the “speed of convergence” of the (cumulative) distribution of $\hat{G}_i(\mathcal{S}_N)$ to the (cumulative) distribution of a Gaussian can be made quantitative using the Berry-Esseen Theorem [17].

First, normalize $G_i(J)$ to zero mean:

$$X_i(J) = G_i(J) - G_i. \quad (6)$$

Denote the variance of $G_i(J)$ (and hence the variance of $X_i(J)$) by σ_i^2 . Since $G_i(J)$ can take on values in a finite domain, the third absolute moment

$$\rho_i = E_{P_S} [|X_i(J)|^3]$$

of X_i is finite.

Second, normalize $\hat{G}_i(\mathcal{S}_N)$ to zero mean:

$$Y_i(\mathcal{S}_N) = \hat{G}_i(\mathcal{S}_N) - G_i = \frac{1}{N} \sum_{j \in \mathcal{S}_N} X_i(j). \quad (7)$$

Third, normalize $\hat{G}_i(\mathcal{S}_N)$ to unit standard deviation:

$$Z_i(\mathcal{S}_N) = \frac{Y_i(\mathcal{S}_N)}{\frac{\sigma_i}{\sqrt{N}}}. \quad (8)$$

The Berry-Esseen Theorem states that the cumulative distribution function (cdf) F_i of $Z_i(\mathcal{S}_N)$ converges (pointwise) to the cdf Φ of the standard normal distribution $N(0, 1)$: For any $x \in \mathbb{R}$,

$$|F_i(x) - \Phi(x)| \leq \frac{0.7975}{\sqrt{N}} \frac{\rho_i}{\sigma_i^3}. \quad (9)$$

It is noted that only information on σ_i and ρ_i are required to make an estimate on the pointwise difference between cdf's of $Z_i(\mathcal{S}_N)$ and $N(0, 1)$. In practice, since the (theoretical) moments σ_i and ρ_i are unknown, we use their empirical estimates.

To ensure that the cdf's of $Z_i(\mathcal{S}_N)$ and $N(0, 1)$ do not differ pointwise by more than $\epsilon > 0$, we need at least

$$N_{\text{cdf}}(\epsilon) = \frac{0.7975^2}{\epsilon^2} \frac{\rho_i^2}{(\sigma_i^2)^3} \quad (10)$$

test strategies.

Let us now assume that the generalization estimates are Gaussian-distributed (e.g., using the analysis above, we gather enough test points to make the means *almost* Gaussian distributed). Denote by $z_{\alpha/2}$

the upper $\alpha/2$ point of $N(0, 1)$, i.e., the area under the standard normal density for $(z_{\alpha/2}, \infty)$ is $\alpha/2$, and for $[-z_{\alpha/2}, z_{\alpha/2}]$ it is $(1 - \alpha)$. For large strategy samples S_N , the estimated generalization performance $\hat{G}_i(S_N)$ of strategy $i \in \mathcal{S}$ has standard error of σ_i/\sqrt{N} . Since σ_i is generally unknown, the standard error can be estimated as

$$\frac{\hat{\sigma}_i(S_N)}{\sqrt{N}} = \sqrt{\frac{\sum_{j \in S_N} (G_i(j) - \hat{G}_i(S_N))^2}{N(N-1)}} \quad (11)$$

and the $100(1 - \alpha)\%$ error margin of $\hat{G}_i(S_N)$ is $z_{\alpha/2}\sigma_i/\sqrt{N}$, or, if σ_i is unknown,

$$\Upsilon_i(\alpha, N) = z_{\alpha/2} \sqrt{\frac{\sum_{j \in S_N} (G_i(j) - \hat{G}_i(S_N))^2}{N(N-1)}}. \quad (12)$$

Requiring that the error margin be at most $\delta > 0$ leads to samples of at least

$$N_{\text{em}}(\delta) = \frac{z_{\alpha/2}^2 \sigma_i^2}{\delta^2} \quad (13)$$

test strategies.

In other words, to be $100(1 - \alpha)\%$ sure that the estimation error $|\hat{G}_i(S_N) - G_i|$ will not exceed δ , we need $N_{\text{em}}(\delta)$ test strategies. Stated in terms of confidence interval, a $100(1 - \alpha)\%$ confidence interval for the true generalization performance of strategy i is

$$(\hat{G}_i(S_N) - \Upsilon_i(\alpha, N), \hat{G}_i(S_N) + \Upsilon_i(\alpha, N)). \quad (14)$$

2.4 Statistical Testing for Comparison of Strategies

One can also make statistical claims regarding hypothesized performance of the studied strategies. For example, one may be only interested in strategies with true generalization performance greater than some threshold \tilde{G} . In this case, we can test whether i is a “bad” strategy by testing for $G_i < \tilde{G}$. The hypothesis H_1 that $G_i < \tilde{G}$ is substantiated at significance level of $\alpha\%$ (against the null hypothesis H_0 that $G_i = \tilde{G}$) if the test statistic

$$Z'_i(S_N, \tilde{G}) = \frac{\hat{G}_i(S_N) - \tilde{G}}{\sqrt{\frac{\sum_{j \in S_N} (G_i(j) - \hat{G}_i(S_N))^2}{N(N-1)}}} \quad (15)$$

falls below $-z_\alpha$, i.e., if $Z'_i(S_N, \tilde{G}) \leq -z_\alpha$. Alternatively, the hypothesis that strategy i is an acceptable strategy, i.e., $G_i > \tilde{G}$, is accepted (against H_0) at significance level of $\alpha\%$, if $Z'_i(S_N, \tilde{G}) \geq z_\alpha$. We can also simply test for $G_i \neq \tilde{G}$, in which case we require $|Z'_i(S_N, \tilde{G})| \geq z_{\alpha/2}$.

Crucially, we can compare two strategies $i, j \in \mathcal{S}$ for their relative performance. This can be important in the evolutionary or co-evolutionary learning setting when constructing a new generation of strategies. Assume that both strategies i and j play against the same set of N test strategies $S_N = \{t_1, t_2, \dots, t_N\}$. Statistical tests regarding the relation between the true generalization performances of i and j can be made using paired tests. One computes a series of performance differences on S_N ,

$$D(n) = G_i(t_n) - G_j(t_n), \quad n = 1, 2, \dots, N.$$

The performance differences are then analyzed as a single sample. At significance level of $\alpha\%$, strategy i appears to be better than strategy j by more than a margin \tilde{D} (against the null hypothesis that i beats j exactly by the margin \tilde{D}), if $Z_i''(S_N, \tilde{D}) \geq z_\alpha$, where

$$Z_i''(S_N, \tilde{D}) = \frac{\hat{D}(S_N) - \tilde{D}}{\sqrt{\frac{\sum_{n=1}^N (D(n) - \hat{D}(S_N))^2}{N(N-1)}}} \quad (16)$$

and

$$\hat{D}(S_N) = \frac{1}{N} \sum_{n=1}^N D(n). \quad (17)$$

For simply testing whether strategy i outperforms strategy j we set the margin to 0, i.e., $\tilde{D} = 0$. Analogously, strategy i appears to be worse than strategy j at significance level of $\alpha\%$, provided $Z_i''(S_N, 0) \leq -z_\alpha$.

Finally, strategies i and j appear to be different at significance level of $\alpha\%$, if $|Z_i''(S_N, 0)| \geq z_{\alpha/2}$. We stress that the comparison of strategies $i, j \in \mathcal{S}$ is done through a set of test strategies in S_N and not through a game of strategy i playing against strategy j . Although one may want to compare one strategy with another directly by having them competing against each other, it should be noted that specific properties in games such as intransitivity may lead to misleading results.

For small samples S_N of test strategies, we would need to use the t -statistic instead of the normally distributed Z -statistic employed here. Distribution of the t -statistic is the Student's t -distribution with $N - 1$ degrees of freedom. However, for sample sizes $N \geq 50$ used in this paper, the Student's t -distribution can be conveniently replaced by the standard normal distribution $N(0, 1)$.

2.5 Properties of Gaussian-distributed Generalization Estimates

It is common to use instead of the true standard deviation σ_i of game outcomes for strategy i its sample estimate (see Eq. (11))

$$\hat{\sigma}_i(S_N) = \sqrt{\frac{\sum_{j \in S_N} (G_i(j) - \hat{G}_i(S_N))^2}{N - 1}}. \quad (18)$$

If we generate n i.i.d. test strategy samples S_N^r , $r = 1, 2, \dots, n$, each of size N , then the generalization performance estimates

$$\hat{G}_i(S_N^r) = \frac{1}{N} \sum_{j \in S_N^r} G_i(j)$$

are close to being Gaussian-distributed with mean G_i and standard deviation σ_i/\sqrt{N} (for large enough N). Such generalization estimates can be used to estimate the confidence interval for σ_i as follows:

The sample variance of the estimates $\hat{G}_i(S_N^r)$ is

$$V_n^2 = \frac{\sum_{r=1}^n (\hat{G}_i(S_N^r) - \Gamma_i)^2}{n - 1}, \quad (19)$$

where

$$\Gamma_i = \frac{1}{n} \sum_{r=1}^n \hat{G}_i(S_N^r). \quad (20)$$

The normalized sample variance of a Gaussian-distributed $\hat{G}_i(S_N^r)$,

$$U_n^2 = \frac{(n-1) V_n^2}{\frac{\sigma_i^2}{N}}, \quad (21)$$

is known to be χ^2 -distributed with $n-1$ degrees of freedom.¹

The $100(1-\alpha)\%$ confidence interval for σ_i/\sqrt{N} is

$$\left(V_n \sqrt{\frac{n-1}{\chi_{\alpha/2}^2}}, V_n \sqrt{\frac{n-1}{\chi_{1-\alpha/2}^2}} \right), \quad (22)$$

where χ_{β}^2 is the value such that the area to the right of χ_{β}^2 under the χ^2 distribution with $N-1$ degrees of freedom is β . It follows that the $100(1-\alpha)\%$ confidence interval for σ_i is

$$\left(V_n \sqrt{\frac{N(n-1)}{\chi_{\alpha/2}^2}}, V_n \sqrt{\frac{N(n-1)}{\chi_{1-\alpha/2}^2}} \right), \quad (23)$$

which can be rewritten as

$$\left(\sqrt{\frac{N \cdot \sum_{r=1}^n (\hat{G}_i(S_N^r) - \Gamma_i)^2}{\chi_{\alpha/2}^2}}, \sqrt{\frac{N \cdot \sum_{r=1}^n (\hat{G}_i(S_N^r) - \Gamma_i)^2}{\chi_{1-\alpha/2}^2}} \right). \quad (24)$$

2.6 Ramifications of Statistical Estimation of Generalization Performance in Co-evolutionary Learning

This framework provides a computationally feasible approach to estimate generalization performance in co-evolutionary learning. A small sample of test strategies may be sufficient to estimate the generalization performance of strategies, even though the strategy space is huge. Furthermore, the framework has the potential application for developing efficient algorithms to improve co-evolutionary search. Our theoretical framework allows us to develop a methodology to find the number of test strategies required for the robust estimation of generalization performance. Subsequently, generalization estimates obtained using a small sample of test strategies (compared to the case of direct estimation of the true generalization) can lead to the co-evolutionary search of strategies with increasingly higher generalization performance since the selection of evolved strategies are based on their estimated generalization performances.

¹There is a trivial inequality for the third moment (page 210 of [29]) that however leads to rather broad bounds.

3 Examples of Statistical Estimation of Generalization Performance in Co-evolutionary Learning

We first illustrate several examples of statistical estimation of generalization performance in co-evolutionary learning based on our theoretical framework in Section 2. We consider the *three*-choice IPD game with deterministic and reactive, memory-one strategies since we can compute the true generalization performance (for simplicity, we assume that test strategies are randomly sampled from \mathcal{S} with a uniform distribution). We demonstrate how one can find and set the required number of random test strategies for robust estimation (given a controlled level of precision) of generalization performance for subsequent use of generalization estimates directly as the fitness measure in co-evolutionary learning. Our results would show that a smaller number of test strategies than predicted previously in [12] is sufficient for robust estimation of generalization performance.

3.1 The Iterated Prisoner's Dilemma Game

In the classical, two-player IPD game, each player is given two choices to play, cooperate or defect [30]. The game is formulated with the predefined payoff matrix specifying the payoff a player receives given the joint move it made with the opponent. Both players receive R (*reward*) units of payoff if both cooperate. They both receive P (*punishment*) units of payoff if they both defect. However, when one player cooperates while the other defects, the cooperator receives S (*sucker*) units of payoff while the defector receives T (*temptation*) units of payoff. The values R , S , T , and P must satisfy the constraints: $T > R > P > S$ and $R > (S + T)/2$. Any set of values can be used as long as they satisfy the IPD constraints (we use $T = 5$, $R = 4$, $P = 1$, and $S = 0$). The game is played when both players choose between the two alternative choices over a series of moves (repeated interactions).

The classical IPD game has been extended to more complex versions, e.g., the IPD with multiple, discrete levels of cooperation [31, 32, 33, 34, 35]. The n -choice IPD game can be formulated using payoffs obtained through the following linear interpolation:

$$p_A = 2.5 - 0.5c_A + 2c_B, \quad -1 \leq c_A, c_B \leq 1, \quad (25)$$

where p_A is the payoff to player A, given that c_A and c_B are the cooperation levels of the choices that players A and B make, respectively. The payoff matrix for the *three*-choice IPD game is given by Figure 1 [12].

		PLAYER B		
		+1	0	-1
PLAYER A	+1	4	2	0
	0	$4\frac{1}{2}$	$2\frac{1}{2}$	$\frac{1}{2}$
	-1	5	3	1

Figure 1: The payoff matrix for the two-player *three*-choice IPD game [12]. Each element of the matrix gives the payoff for Player A.

The payoff matrix for any n -choice IPD game must satisfy the following conditions [32]:

1. For $c_A < c'_A$ and constant c_B : $p_A(c_A, c_B) > p_A(c'_A, c_B)$,
2. For $c_A \leq c'_A$ and $c_B < c'_B$: $p_A(c_A, c_B) < p_A(c'_A, c'_B)$, and
3. For $c_A < c'_A$ and $c_B < c'_B$: $p_A(c'_A, c'_B) > (1/2)(p_A(c_A, c'_B) + p_A(c'_A, c_B))$.

These conditions are analogous to those for the classical IPD's: (1) defection always pays more, (2) mutual cooperation has a higher payoff than mutual defection, and (3) alternating between cooperation and defection pays less in comparison to just playing cooperation.

3.2 What is the Required Number of Test Strategies?

We would like to find out and set the required number N of random test strategies drawn i.i.d. from \mathcal{S} for robust estimation of generalization performance for a game. Instead of making some assumptions about the complexity of a game and the impact on the required number of random test strategies, we demonstrate a principled approach based on our theoretical framework in Section 2. Our approach exploits the near-Gaussian nature of generalization estimates and finds out the rate at which the distribution of generalization estimates converges to a Gaussian as the number of random test strategies to compute generalization estimates grows.

We illustrate our approach for the *three-choice* IPD game. We first collect a sample of 50 base strategies i , which we obtain by randomly sampling from \mathcal{S} with uniform distribution. We also collect 1000 independent samples S_N to compute 1000 estimates $\hat{G}_i(S_N)$. Each random sample S_N consists of N test strategies drawn i.i.d. from \mathcal{S} with uniform distribution.

For each base strategy i , we directly estimate the true generalization performance G_i from Equation 2 and normalize $G_i(J)$ by taking $X_i(J) = G_i(J) - G_i$. We can then compute estimates of the variance and the third absolute moment of $X_i(J)$ with respect to S_N , i.e., for each strategy i , we have a 1000-sample estimate of $\hat{\sigma}_i^2$ and another 1000-sample estimate of $\hat{\rho}_i$.

From the Berry-Esseen Theorem (Equation 10) [17], we can compute for each base strategy i the deviation from the Gaussian:

$$\epsilon = \frac{0.7975 \cdot \hat{\rho}_i}{\sqrt{N} \cdot \hat{\sigma}_i^3} \quad (26)$$

given different sample sizes of S_N . By systematically computing the error ϵ for S_N with $N = \{50, 100, 200, 300, 400, 500, 1000, 2000, 3000, 4000, 5000, 10000, 50000\}$, we can observe how fast the distribution of generalization estimates is converging to a Gaussian.

Since we do not know the true value of ϵ , we take a pessimistic estimate of ϵ . Both 1000-sample estimates of $\hat{\sigma}_i^2$ and $\hat{\rho}_i$ are first rank-ordered in an ascending order. A pessimistic estimate of ϵ would be to take a smaller value (2.5%-tile) of $\hat{\sigma}_i^2$ and a larger value (97.5%-tile) of $\hat{\rho}_i$.

Although we can directly compute the quantile intervals for $\hat{\sigma}_i^2$ from the χ^2 -distribution, we have loose bounds for $\hat{\rho}_i$ (based on the inequality from page 210 of [29]), which would result in unnecessarily larger values in our pessimistic estimate of ϵ . Our comparison of quantiles from a 1000-sample estimate of $\hat{\sigma}_i^2$ between empirical estimates and estimates obtained directly from the χ^2 -distribution (Equation 24) indicate an absolute difference around 0.03 when $N = 50$ (which is the smallest sample size we consider) and is smaller for larger values of N on average. Given the small absolute difference and that we are already computing pessimistic estimates of ϵ , we will use quantiles for $\hat{\sigma}_i^2$ and $\hat{\rho}_i$ obtained empirically for subsequent experiments.²

²The non-parametric quantile estimation is performed in the usual manner on ordered samples. Uniform approximation of the true distribution function by an empirical distribution function based on sample values is guaranteed, e.g., by the Glivenko-Cantelli Theorem [29, 36].

Figure 2 plots the results for the 50 strategies i showing ϵ against N . Table 1 lists out ϵ for different N s for ten strategies i . Naturally, increasing the sample size N leads to decreasing values of ϵ . However, there is a trade-off between more robust estimation of generalization performance and increasing computational cost. Figure 2 shows that ϵ decreases rapidly when N increases from 50 to 1000, but starts to level off from around $N = 1000$ onwards. Table 1 suggests that at $N = 2000$, S_N would provide a sufficiently robust estimate of generalization performance for a reasonable computational cost since one would need a five-fold increase of N to 10000 to reduce ϵ by half.³ Furthermore, since for non-pathological strategies⁴, $\hat{\sigma}_i^2$ and $\hat{\rho}_i$ in Equation 26 are finite moments bounded away from 0, for larger N , ϵ is dominated by the term $N^{-1/2}$. In our experiments, this implies that the trade-off between more robust estimations of generalization and computational cost is roughly the same for most of the strategies.

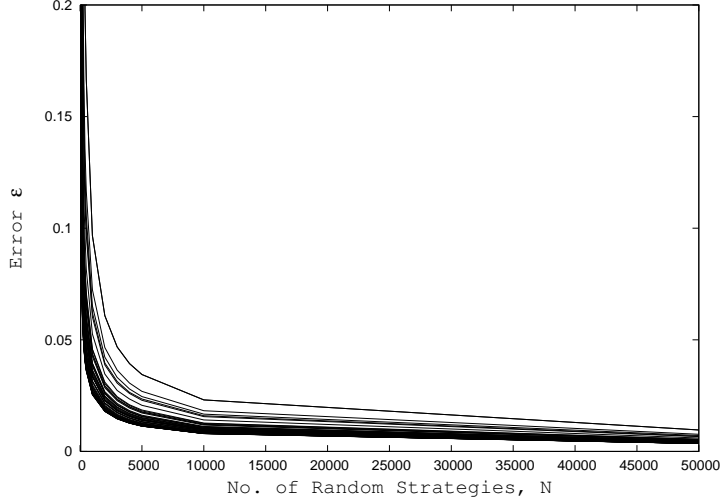


Figure 2: Pessimistic estimate of ϵ as a function of sample size N of test strategies for 50 random base strategies i .

Leaving the previous analysis aside for a moment and assuming that estimates $\hat{G}_i(\mathcal{S}_N)$ for a base strategy i are Gaussian-distributed, from Equation 13, we obtain the error δ :

$$\delta = \frac{z_{\alpha/2} \hat{\sigma}_i}{\sqrt{N}}. \quad (27)$$

We compute the pessimistic estimate of δ , taking 97.5%-tile of $\hat{\sigma}_i^2$ from the rank-ordered 1000-sample estimates. Our results for δ also indicate a trade-off between more robust estimation and increasing computational cost, and suggest that S_N at $N = 2000$ would provide a sufficiently robust estimate of generalization performance for a reasonable computational expense. The results in Table 2 show that the absolute difference between ϵ and δ becomes smaller for larger sample sizes (e.g., at $N > 1000$ the absolute difference is less than 0.01).

We also illustrate how two strategies can be compared with respect to their generalization performances through a test of statistical significance based on the normally distributed Z-statistic. For example, Table 3 shows numerical results for the p -values obtained from Z-tests directly using Equation 16 to find out whether one strategy outperforms another with respect to a sample of random test strategies of size N . In this case, since the two strategies actually differ substantially in performance, a small sample size of test strategies would be sufficient to test for its statistical significance (at around $N = 400$, the p -values are smaller than the significance level of 0.05). Our experiments with other pairs of strategies

³Note that because we take a pessimistic estimate of ϵ , it is possible that the computed value of ϵ is greater than the real one, especially for small sample sizes (e.g., strategy # 7 at $N = 50$).

⁴By pathological strategies we mean strategies with very little variation of game outcomes when playing against a wide variety of opponent test strategies.

Table 1: Pessimistic estimates of ϵ from Equation 26 for ten strategies i . The results show a trade-off between more robust estimation of generalization performance and increasing computational cost. Increasing the sample size N leads to decreasing error ϵ but soon levels off around $N = 1000$ onwards.

N	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
50	0.153908	0.12059	0.277445	0.791636	0.125334	0.40297	1.095542	0.132867	0.11612	0.233136
100	0.09876	0.083803	0.155298	0.357576	0.08611	0.227821	0.452723	0.090216	0.081356	0.138341
200	0.066866	0.058456	0.097296	0.200709	0.059497	0.136423	0.240982	0.06164	0.057204	0.086697
300	0.053277	0.047474	0.073737	0.141732	0.048203	0.099911	0.170278	0.049558	0.046587	0.067476
400	0.045463	0.041004	0.063622	0.114299	0.041509	0.082325	0.135654	0.042775	0.040304	0.057132
500	0.040469	0.036562	0.054709	0.100387	0.036949	0.071225	0.11657	0.037963	0.036012	0.049744
1000	0.027898	0.025662	0.036199	0.06219	0.025904	0.045869	0.072467	0.026473	0.025397	0.033765
2000	0.019337	0.018072	0.024639	0.039612	0.018206	0.031049	0.046544	0.018536	0.017922	0.023025
3000	0.015663	0.014728	0.019851	0.031357	0.014826	0.024533	0.036414	0.015082	0.014621	0.018436
4000	0.013501	0.012742	0.016939	0.02645	0.012814	0.020883	0.030649	0.013013	0.012655	0.015841
5000	0.012039	0.011384	0.015053	0.023489	0.011444	0.018533	0.026889	0.011622	0.011315	0.014013
10000	0.008441	0.008033	0.010422	0.015988	0.008074	0.012733	0.018249	0.008178	0.007994	0.009747
50000	0.003740	0.003583	0.004536	0.006792	0.003597	0.005481	0.007667	0.003634	0.003571	0.004266

Table 2: Statistics of $\{|\epsilon - \delta|_i\}$ for 50 strategies i .

N	Mean	Std. Dev.
50	0.199785	0.359618
100	0.093783	0.168228
200	0.043777	0.068606
300	0.030180	0.046173
400	0.024052	0.036315
500	0.020268	0.030288
1000	0.011935	0.016976
2000	0.007458	0.010326
3000	0.005749	0.007831
4000	0.004822	0.006521
5000	0.004219	0.005665
10000	0.002825	0.003728
50000	0.001173	0.001510

with smaller performance differences indicate the need for a larger sample size of test strategies to test for statistical significance.

We have illustrated examples of statistical estimation of generalization performance in co-evolutionary learning. Our studies have indicated that the number of test strategies that is required for robust estimation (given a controlled level of precision) of generalization performance is smaller than predicted earlier using the distribution-free framework (Chebyshev’s) [12]. This has an obvious impact in the use of generalization estimates as a fitness measure in co-evolutionary learning since estimations have to be repeated throughout the evolutionary process. Although we use the IPD game as an example, our theoretical framework presented in Section 2 can be applied to other more complex problems or scenarios. The information we need to find and set the required number of test strategies for robust estimation only involves the second (variance) and third order moments, which can be estimated as well. As an example that we present in the next section, we illustrate how the framework can be applied in the co-evolutionary learning of the more complex Othello game.

4 Using the Notion of Statistical Estimation of Generalization Performance as Fitness Measure in Co-evolutionary Learning

We will investigate the notion of directly using generalization estimates as a form of fitness measure in co-evolutionary learning. Ideally, we would like to make a direct estimation on the true generalization performance of the evolved strategy. In this case, co-evolutionary learning would lead to the search of strategies with increasingly higher generalization performance since the selection of evolved strategies are based on their generalization performances⁵. However, such direct estimations can be computationally expensive. Instead, we investigate the use of relatively small samples of test strategies to guide and improve the co-evolutionary search following our earlier studies on the number of test strategies required for robust estimation. We first study this new approach of directly using generalization estimates as the fitness measure for the co-evolutionary learning of the IPD game before applying it to the more complex game of Othello.

⁵In the same way, the selection process in the pareto co-evolution [24] is based on the pareto-dominance relationship although establishing such a relationship requires competing solutions to interact (solve) a sample of test cases.

Table 3: Computed p -values of Z -tests to determine whether a strategy i outperforms a strategy j . The table lists the results for 10 independent samples of test strategies of size N . Note that the true generalization performances of strategies i and j are known and are 0.4198 and 0.3066, respectively.

N	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
50	1.37×10^{-01}	9.68×10^{-02}	4.08×10^{-03}	1.63×10^{-02}	4.09×10^{-02}	9.68×10^{-02}	6.64×10^{-01}	8.63×10^{-01}	5.82×10^{-01}	4.17×10^{-02}
100	1.31×10^{-01}	9.60×10^{-02}	4.55×10^{-03}	5.62×10^{-03}	1.02×10^{-02}	1.36×10^{-01}	3.62×10^{-01}	5.67×10^{-01}	2.69×10^{-01}	2.99×10^{-02}
200	2.89×10^{-02}	9.55×10^{-02}	5.68×10^{-05}	2.93×10^{-04}	7.72×10^{-03}	1.57×10^{-04}	3.93×10^{-02}	1.39×10^{-01}	4.77×10^{-03}	1.31×10^{-04}
300	5.41×10^{-03}	3.39×10^{-02}	5.05×10^{-05}	3.52×10^{-05}	2.24×10^{-03}	1.45×10^{-06}	6.05×10^{-03}	6.74×10^{-02}	4.40×10^{-04}	6.43×10^{-06}
400	1.05×10^{-03}	3.60×10^{-03}	2.93×10^{-08}	3.50×10^{-06}	8.18×10^{-06}	8.22×10^{-08}	1.26×10^{-03}	7.10×10^{-03}	2.12×10^{-05}	1.69×10^{-06}
500	5.51×10^{-04}	5.05×10^{-04}	5.01×10^{-08}	3.19×10^{-10}	8.59×10^{-07}	1.61×10^{-07}	4.48×10^{-04}	3.42×10^{-03}	1.14×10^{-05}	8.14×10^{-07}
1000	9.94×10^{-09}	2.72×10^{-08}	1.89×10^{-15}	2.13×10^{-12}	1.11×10^{-13}	4.59×10^{-10}	3.79×10^{-09}	4.77×10^{-09}	7.07×10^{-09}	3.37×10^{-11}
2000	0	2.22×10^{-16}	0	0	1.11×10^{-16}	0	0	0	0	0

4.1 Co-evolutionary Learning of IPD

4.1.1 Strategy Representation

Various strategy representations for the co-evolutionary learning of IPD have been studied in the past, e.g., the look-up table with bit-string encoding [3], finite state machines [37, 38], and neural networks [32, 35, 39, 40]. The study in [41] has further investigated other forms of representations such as cellular representation for finite state machines and Markov chains among others, and their impact on the evolution of cooperation. We use the direct look-up table strategy representation [32] that directly represents IPD strategy behaviors through a one-to-one mapping between the genotype space (strategy representation) and the phenotype space (behaviors). The main advantage of using this representation is that the search space given by the strategy representation and the strategy space is the same (assuming a uniform strategy distribution in \mathcal{S}), which simplifies and allows direct investigation on the co-evolutionary search for strategies with higher generalization performance [12].

For a deterministic and reactive, memory-one n -choice IPD strategy, the direct look-up table representation takes the form of $m_{ij}, i, j = 1, 2, \dots, n$ table elements that specify the choice to be made given the inputs of i (player's own previous choice) and j (opponent's previous choice). The first move m_{fm} is specified independently rather than using pre-game inputs (two for memory-one strategies). m_{ij} and m_{fm} can take any of the n values (choices) used to produce the payoffs in the payoff matrix through a linear interpolation. Figure 3 illustrates the direct look-up table representation for the *three*-choice IPD strategy [32] where each table element can take +1, 0, or -1.

		Opponent's Previous Move		
		+1	0	-1
Player's	+1	m_{11}	m_{12}	m_{13}
Previous	0	m_{21}	m_{22}	m_{23}
Move	-1	m_{31}	m_{32}	m_{33}

Figure 3: Direct look-up table representation for the deterministic and reactive memory-one IPD strategy that considers three choices (also includes m_{fm} for the first move, which is not shown in the figure).

Mutation is used to generate an offspring from a parent strategy when using the direct look-up table for strategy representation [32]. Mutation replaces the original choice of an element in the direct look-up table with one of the remaining $n - 1$ possible choices with an equal probability of $1/(n - 1)$. Each element (m_{ij} and m_{fm}) has a fixed probability p_m of being replaced. The mutation can provide sufficient variations on strategy behaviors directly with the use of the direct look-up table representation (even for the more complex IPD game with more choices) [32].

4.1.2 Co-evolutionary Learning Procedure

The following describes the classical co-evolutionary learning procedure [32, 12]:

1. Generation step, $t = 1$:
Initialize $|\text{POP}|/2$ parent strategies $i = 1, 2, \dots, |\text{POP}|/2$ randomly.
2. Generate $|\text{POP}|/2$ offspring strategies $i = |\text{POP}|/2 + 1, |\text{POP}|/2 + 2, \dots, |\text{POP}|$ from $|\text{POP}|/2$ parent strategies through a mutation operator with $p_m = 0.05$.
3. All pairs of strategies in the population POP compete, including the pair where a strategy plays itself (round-robin tournament). For $|\text{POP}|$ strategies, every strategy competes a total of $|\text{POP}|$ games. The fitness of a strategy i is $\frac{1}{|\text{POP}|} \sum_{j \in \text{POP}} G_i(j)$.

4. Select the best $|\text{POP}|/2$ strategies based on fitness. Increment generation step, $t \leftarrow t + 1$.
5. Step 2 to 4 are repeated until termination criterion (a fixed number of generation) is met.

We consider $|\text{POP}| = 50$. All IPD games involve a fixed game length of 150 iterations. A fixed and sufficiently long duration for the evolutionary process ($t = 300$) is used. As in [12], we observe how the generalization performance of co-evolutionary learning (we measure the generalization performance of the top performing evolved strategy) changes during the evolutionary process. All experiments are repeated in 30 independent runs to allow for statistical analysis.

The classical co-evolutionary learning (which we refer as CCL) is used as a baseline for comparison with the improved co-evolutionary learning (which we refer as ICL) that directly uses generalization performance estimates as the fitness measure. We first study a simple implementation of this co-evolutionary learning approach where the estimate $\hat{G}_i(S_N)$ is directly used as the fitness of the evolved strategy i . The procedure for this new approach is similar to the baseline with the exception of Step (3). This is to allow a more direct comparison. We investigate the approach with different sample sizes $N = \{50, 500, 1000, 2000, 10000, 50000\}$. The sample S_N is generated anew every generation. We use a sample size of $N = 50000$ to provide a “ground truth” estimate close to the true generalization performance (based on the distribution-free framework) with which to compare results for cases where much smaller samples are used.

4.2 Results and Discussion

Figure 4 shows results for our experiments. Each graph plots the true generalization performance G_i of the top performing strategy of the population throughout the evolutionary process for all 30 independent runs. In particular, Figure 4(a) shows that when co-evolutionary learning uses a fitness measure based on relative performance between competing strategies in the population, the search process can exhibit large fluctuations in the generalization performance of strategies throughout co-evolution. This is consistent with observations from previous studies such as [21, 22, 32] where it has been shown that fluctuations in the generalization performance during co-evolution are due to overspecialization of the population to a specific strategy that is replaced by other strategies that can exploit it. Results from our baseline experiment show that the use of relative fitness measure does not necessarily lead to the co-evolutionary learning of strategies with increasingly higher generalization performance.

However, for all ICL experiments where estimates $\hat{G}_i(S_N)$ are directly used as the fitness measure, no evolutionary run is observed to exhibit large fluctuations in generalization performance (Fig. 4). This is in contrast to the case of CCL where runs exhibit large fluctuations during co-evolution (Fig. 4(a)). Starting with the case of a small sample of size 50, the search process of ICL-N50 (S_N , $N = 50$) exhibits only small fluctuations in the generalization performance during co-evolution. These fluctuations are a result of sampling errors from using a small sample of test strategies to estimate $\hat{G}_i(S_N)$, which can affect the ranking of strategies i in the co-evolving population for selection.

Results from Figure 4 suggest that when generalization estimates are directly used as the fitness measure, co-evolutionary learning converges to higher generalization performance. For example, when a sample of 500 test strategies is used to estimate $\hat{G}_i(S_N)$, more evolutionary runs converge to higher generalization performance without fluctuations compared to the case when 50 test strategies are used. However, we do not observe significant differences at the end of the evolutionary runs for ICLs when the sample size is increased further, i.e., between ICL-N2000 and ICL-N50000 (Fig. 4). Closer inspection on evolved strategies reveals that they play nearly “all defect” or are actually “all defect” strategies. This observation is expected since “all defect” strategy has the maximum generalization performance for the game outcome defined by Equation 1.

We have also collected various statistics on G_i measurements of CCL and ICLs using different sample sizes in Table 4. The table shows that starting from a small sample of 50 test strategies, the increase in the generalization performance of ICL is statistically significant in comparison to the case of CCL.

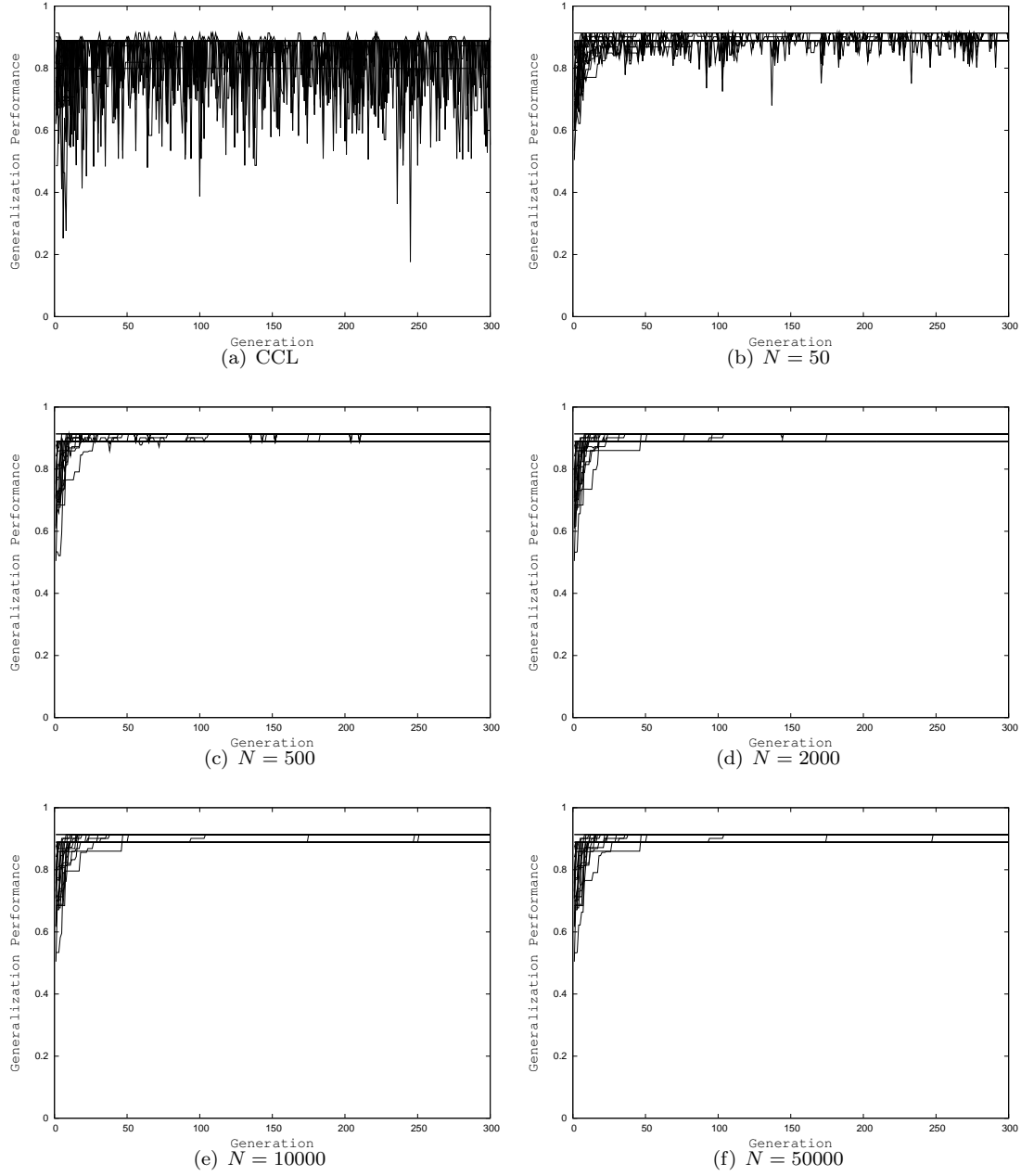


Figure 4: Comparison of CCL and different ICLs for the *three-choice* IPD game: (a) CCL, (b) ICL-N50, (c) ICL-N500, (d) ICL-N2000, (e) ICL-N10000, and (f) ICL-N50000. Shown are plots of the true generalization performance G_i of the top performing strategy of the population throughout the evolutionary process for all 30 independent runs.

The generalization performance of ICL appears to have settled with no significant increase when sample size N is increased from 500 to 50000 (which is the sample size based on the distribution-free framework and close in number to all possible strategies for the *three*-choice IPD). The estimates $\hat{G}_i(S_N)$ appear to be robust at small sample sizes of S_N to guide and improve co-evolutionary search to obtain strategies with high generalization performance. The co-evolutionary learning is also much faster since significantly smaller sample sizes (around an order of magnitude smaller in number of test strategies) are sufficient to achieve similarly high generalization performance.

Table 4: Summary of results for different co-evolutionary learning approaches for the *three*-choice IPD taken at the final generation. For each approach, the mean and standard error at 95% confidence interval over 30 runs are calculated. “Max” and “Min” give the maximum and minimum G_i over 30 runs. The t -tests compare CCL with ICLs.

Experiment	Mean \pm Std Err	Max	Min	t -test
CCL	0.873 ± 0.023	0.901	0.553	
ICL-N50	0.900 ± 0.005	0.914	0.889	-2.382^\dagger
ICL-N500	0.909 ± 0.003	0.914	0.889	-3.057^\dagger
ICL-N1000	0.908 ± 0.004	0.914	0.889	-2.892^\dagger
ICL-N2000	0.910 ± 0.003	0.914	0.889	-3.141^\dagger
ICL-N10000	0.909 ± 0.004	0.914	0.889	-2.974^\dagger
ICL-N50000	0.909 ± 0.004	0.914	0.889	-2.974^\dagger

† The t value 29 degrees of freedom is statistically significant at a 0.05 level of significance using a two-tailed test.

At this point, we have compared only the generalization performances of the co-evolutionary learning that directly uses $\hat{G}_i(S_N)$ with the classical co-evolutionary learning that uses relative fitness measure. However, it is of interest to investigate the co-evolutionary learning that uses a fitness measure consisting of a mixture of the two fitness values to determine the impact on generalization performance. We consider the simple implementation of a weighted sum of fitness measures:

$$\text{fitness}_i = (\eta) \cdot \left(\frac{1}{N} \sum_{j \in S_N} G_i(j) \right) + (1 - \eta) \cdot \left(\frac{1}{|\text{POP}|} \sum_{k \in \text{POP}} G_i(k) \right), \quad (28)$$

where higher η values give more weight to the contribution of estimates $\hat{G}_i(S_N)$ for the selection of evolved strategies. We investigate this approach where the estimate $\hat{G}_i(S_N)$ is computed with $N = 10000$ test strategies (to ensure a reasonable tradeoff between accuracy and computational expense) and η at 0.25 (MCL25-N10000), 0.50 (MCL50-N10000), and 0.75 (MCL75-N10000).

Results show that co-evolutionary learning is able to search for strategies with high generalization performance (Fig. 5). However, the inclusion of relative fitness leads to fluctuations in the generalization performance of co-evolutionary learning. The fluctuations are smaller and localized around a high generalization performance when the contribution of relative fitness is reduced (Fig. 5(b)). Our results suggest that the co-evolutionary search of strategies with high generalization performance is due to the estimate $\hat{G}_i(S_N)$ that contributes to the fitness measure. There is no positive impact to the generalization performance of co-evolutionary learning by including relative fitness.

4.3 Co-evolutionary Learning of Othello

In this section, we demonstrate our new approach to more complex problems. As an example, we will show that ICL also improves on co-evolutionary learning for the more complex game of Othello. We can achieve similarly high generalization performance using estimates requiring an order of magnitude smaller number of test strategies than the case of a distribution-free framework. We do not necessarily

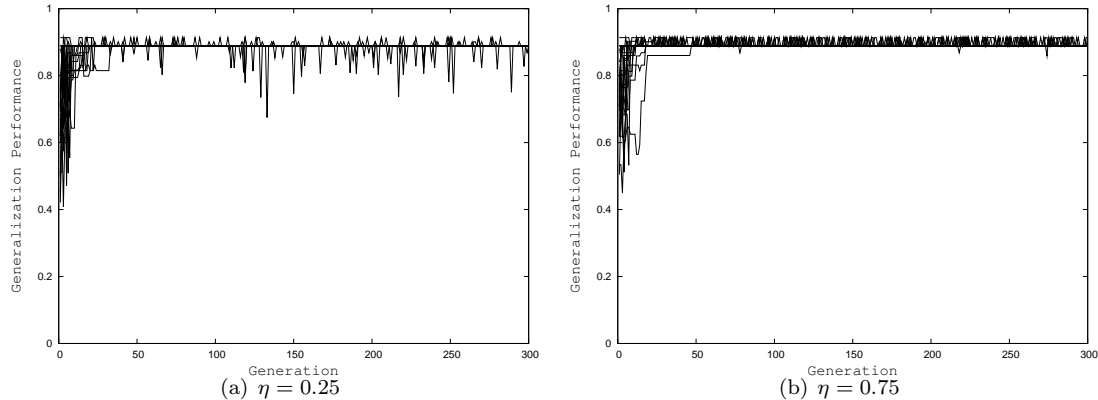


Figure 5: Different MCL-N10000s for the *three*-choice IPD game: (a) MCL25-N10000 and (b) MCL75-N10000. Shown are plots of the true generalization performance G_i of the top performing strategy of the population throughout the evolutionary process for all 30 independent runs.

need larger samples of test strategies when applying the new approach to more complex games. Instead, we can find out in advance the required number of test strategies for robust estimation before applying ICL to the game of Othello.

4.3.1 Othello

Othello is a deterministic, perfect information, zero-sum, board game played by two players (black and white) that alternatively place their (same colored) pieces on an eight-by-eight board. The game starts with each player having two pieces already on the board as shown by Figure 6. In Othello, the black player starts the game by making the first move. A legal move is one where the new piece is placed adjacent horizontally, vertically, or diagonally to an opponent’s existing piece (e.g., Figure 6 (top-right)) such that at least one of opponent’s pieces lies between the player’s new piece and existing pieces (e.g., Figure 6 (bottom-left)). The move is completed when the opponent’s surrounded pieces are flipped over to become the player’s pieces (e.g., Figure 6 (bottom-right)). A player that could not make a legal move forfeits and passes the move to the opponent. The game ends when all the squares of the board are filled with pieces, or when neither player is able to make a legal move [7].

4.3.2 Strategy Representation

Among the strategy representations that have been studied for the co-evolutionary learning of Othello strategies (in the form of a board evaluation function) are weighted piece counters [42] and neural networks [7, 43]. We consider the simple strategy representation of a weighted piece counter in the following empirical study. This is to allow a more direct investigation of the impact of fitness evaluation in the co-evolutionary search of Othello strategies with higher generalization performance.

A weighted piece counter (WPC) representing the board evaluation function of an Othello game strategy can take the form of a vector of 64 weights, indexed as $w_{rc}, r = 1, \dots, 8, c = 1, \dots, 8$, where r and c represent the position indexes for rows and columns of an eight-by-eight Othello board, respectively. Let the Othello board state be the vector of 64 pieces, indexed as $x_{rc}, r = 1, \dots, 8, c = 1, \dots, 8$, where r and c represent the position indexes for rows and columns. x_{rc} takes the value of +1, -1, or 0 for black piece, white piece, and empty piece, respectively [42].

The WPC would take the Othello board state as input, and output a value that gives the worth of the board state. This value is computed as:

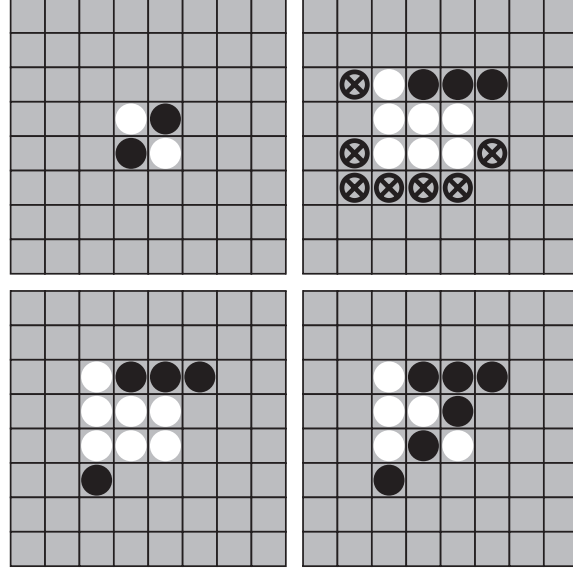


Figure 6: Figure illustrates basic Othello moves. (Top left) The positions of respective players' pieces at the start of the game. (Top right) The possible legal moves (which are indicated by black, crossed circles) at a later point of the game. (Bottom left) The black player selecting a legal move. (Bottom right) The black move is completed where surrounded white pieces are flipped over to become black pieces [7].

$$\text{WPC}(\mathbf{x}) = \sum_{r=1}^8 \sum_{c=1}^8 w_{rc} \cdot x_{rc}, \quad (29)$$

where the more positive value of $\text{WPC}(\mathbf{x})$ would indicate WPC's interpretation that the board state \mathbf{x} is more favorable if WPC is a black player. The more negative value of $\text{WPC}(\mathbf{x})$ would indicate WPC's interpretation that the board state \mathbf{x} is more favorable if WPC is a white player [42].

We consider a simple mutation operator, where the WPC weight of the offspring w'_{rc} can be obtained by adding a small random value to the corresponding WPC weight of the parent w_{rc} :

$$w'_{rc} = w_{rc} + k * F_{rc}, \quad r = 1, \dots, 8, c = 1, \dots, 8 \quad (30)$$

where k is a scaling constant ($k = 0.1$) and F_{rc} is a real number randomly drawn from $[-1, 1]$ with a uniform distribution and resampled for every combination of r and c (total of 64 weights). For the

experiments, we consider the space of Othello strategies given by the WPC representation with $w_{rc} \in [-10, 10]$. The simple mutation-operator can provide sufficient variation to the Othello game strategy represented in the form of a WPC evaluation function. We note that choices for various parameters are not optimized. The main emphasis of our study is to investigate the impact of generalization performance estimates used as fitness measure in improving the generalization performance of co-evolutionary learning.

4.3.3 Measuring Generalization Performance on Othello Strategies

Unlike the IPD game that is symmetric, the Othello game is not necessarily symmetric, i.e., the black and white players may not have the same sets of available strategies [23]. In this case, we consider two estimates of generalization performance. We estimate the generalization performance of a black WPC through Othello game-plays against a random test sample of white WPCs. Conversely, we estimate the generalization performance of a white WPC through Othello game-plays against a random test sample of black WPCs. A random test sample of Othello WPC is obtained through random sampling of w_{rc} from $[-10, 10]$ having a uniform distribution and resampled for every combination of r and c . We use a random sample of 50000 test WPCs (opposite color) to directly estimate the generalization performance of evolved WPC since we cannot compute the true generalization performance.

4.3.4 Co-evolutionary Learning Procedure

Given the approach we use to measure the generalization performance of evolved Othello WPC, we repeat all experiment settings twice: one for black WPC and one for white WPC. For example, the CCL of black WPC is described as follows:

1. Generation step, $t = 1$:
Initialize $|\text{POP}|/2$ parent strategies $i = 1, 2, \dots, |\text{POP}|/2$ randomly. For a WPC_i , w_{rc}^i is real number randomly sampled from $[-0.2, 0.2]$ having a uniform distribution and resampled for every combination of r and c .
2. Generate $|\text{POP}|/2$ offspring strategies $i = |\text{POP}|/2 + 1, |\text{POP}|/2 + 2, \dots, |\text{POP}|$ from $|\text{POP}|/2$ parent strategies through a mutation operator given by Equation 30.
3. All pairs of strategies in the population POP compete, including the pair where a strategy plays itself (round-robin tournament). For $|\text{POP}|$ strategies, every strategy competes a total of $|\text{POP}|$ games. The fitness of a black WPC strategy i is $\frac{1}{|\text{POP}|} \sum_{j \in \text{POP}} G_i(j)$, where $G_i(j)$ is the game outcome to i for an Othello game played by i (black) and j (white).
4. Select the best $|\text{POP}|/2$ strategies based on fitness. Increment generation step, $t \leftarrow t + 1$.
5. Step 2 to 4 are repeated until termination criterion (i.e., a fixed number of generation) is met.

For the co-evolutionary learning of Othello, we consider the same population of 50 evolving strategies and a shorter evolutionary duration of 200 generations compared to the co-evolutionary learning of IPD. This is due to the increase in computational expense in a single Othello game compared to a single IPD game. All experiments are repeated in 30 independent runs to allow for statistical analysis.

As in the previous experiment, CCL is used as a baseline for comparison, and ICLs with different sample sizes of $N = \{50, 500, 1000, 5000, 10000, 50000\}$ to estimate $\hat{G}_i(S_N)$ are considered. The sample size of $N = 50000$ is to provide a “ground truth” estimate close to the true generalization performance with which to compare results for cases where much smaller samples are used. Note that different samples of S_N are used to estimate $\hat{G}_i(S_N)$ as the fitness measure in ICL and to estimate the generalization performance of ICL for analysis.

4.3.5 Results and Discussion

Figure 7 shows results for our experiments. Each graph plots the estimated generalization performance $\hat{G}_i(S_N)$ ($N = 50000$) of the top performing strategy of the population throughout the evolutionary process for all 30 independent runs. As with the CCL of the simpler IPD game (Fig. 4(a)), results for the CCL of black and white WPCs indicate a search process with large fluctuations in the generalization performance of strategies throughout co-evolution (Figs. 7(a) and 7(b)). Our results suggest that co-evolutionary learning does not necessarily lead to Othello WPC strategies with increasingly higher generalization performance when a relative fitness measure is used.

When estimates $\hat{G}_i(S_N)$ are directly used as the fitness measure in co-evolutionary learning, fluctuations in the generalization performance are reduced and that the co-evolutionary search converges to higher generalization performance compared to the case of CCL for the Othello game (Fig. 7). We observe that ICL can search WPCs with higher generalization performance although small fluctuations can be seen during co-evolution when estimates $\hat{G}_i(S_N)$ are computed using a small sample of 50 test strategies. Further increase in sample size leads to further improvements in generalization performance of ICL, e.g., when $N = 500$ (Figs. 7(c) and 7(d)).

There is a point where we observe that a significant increase in the sample size does not bring about a significant increase in generalization performance of ICL. For example, results for ICL-N5000 is similar to that of ICL-N50000 (Figs. 7(e) and 7(f)). This is consistent with results from experiments to find the required number of test strategies for robust estimation of generalization performance (Fig. 8). The figure suggests a trade-off at $N = 5000$ for S_N to provide sufficiently robust estimation for a reasonable computation cost since substantially increasing N to 50000 would not lead to a significant decrease in the error ϵ for the Othello game.

Tables 5 and 6 compare the generalization performance of CCL with ICLs at the end of the generational runs for black and white WPCs, respectively. They show that there is a positive and significant impact on the generalization performance in co-evolutionary learning when estimates $\hat{G}_i(S_N)$ are directly used as the fitness measure. The means over 30 runs are higher while the standard errors at 95% confidence interval are lower when comparing results between ICLs and CCL. In addition, results of controlled experiments for co-evolutionary learning with the fitness measure being a mixture of the estimate $\hat{G}_i(S_N)$ and relative fitness (Equation 28) indicate that when the contribution of the relative fitness is reduced while that of the estimate $\hat{G}_i(S_N)$ is increased, higher generalization performance can be obtained with smaller fluctuations throughout co-evolution. These results further support our previous observation from Figure 7 that co-evolution converges to higher generalization performance without large fluctuations as a result of directly using the generalization estimate $\hat{G}_i(S_N)$ as the fitness measure.

Table 5: Summary of results for different co-evolutionary learning approaches for black Othello WPC taken at the final generation. For each approaches, the mean and standard error at 95% confidence interval over 30 runs are calculated. “Max” and “Min” give the maximum and minimum G_i over 30 runs. The t -tests compare CCL with ICLs.

Experiment	Mean \pm Std Err	Max	Min	t -test
CCL	0.745 \pm 0.018	0.836	0.643	
ICL-N50	0.822 \pm 0.008	0.896	0.828	−7.827 [†]
ICL-N500	0.875 \pm 0.005	0.913	0.880	−13.773 [†]
ICL-N1000	0.890 \pm 0.004	0.914	0.868	−16.352 [†]
ICL-N5000	0.899 \pm 0.003	0.913	0.878	−16.065 [†]
ICL-N10000	0.900 \pm 0.004	0.915	0.867	−15.550 [†]
ICL-N50000	0.906 \pm 0.003	0.920	0.887	−17.816 [†]

[†] The t value 29 degrees of freedom is statistically significant at a 0.05 level of significance using a two-tailed test.

Our empirical studies indicate that the use of generalization estimates directly as the fitness measure can have a positive and significant impact on the generalization performance of co-evolutionary learning

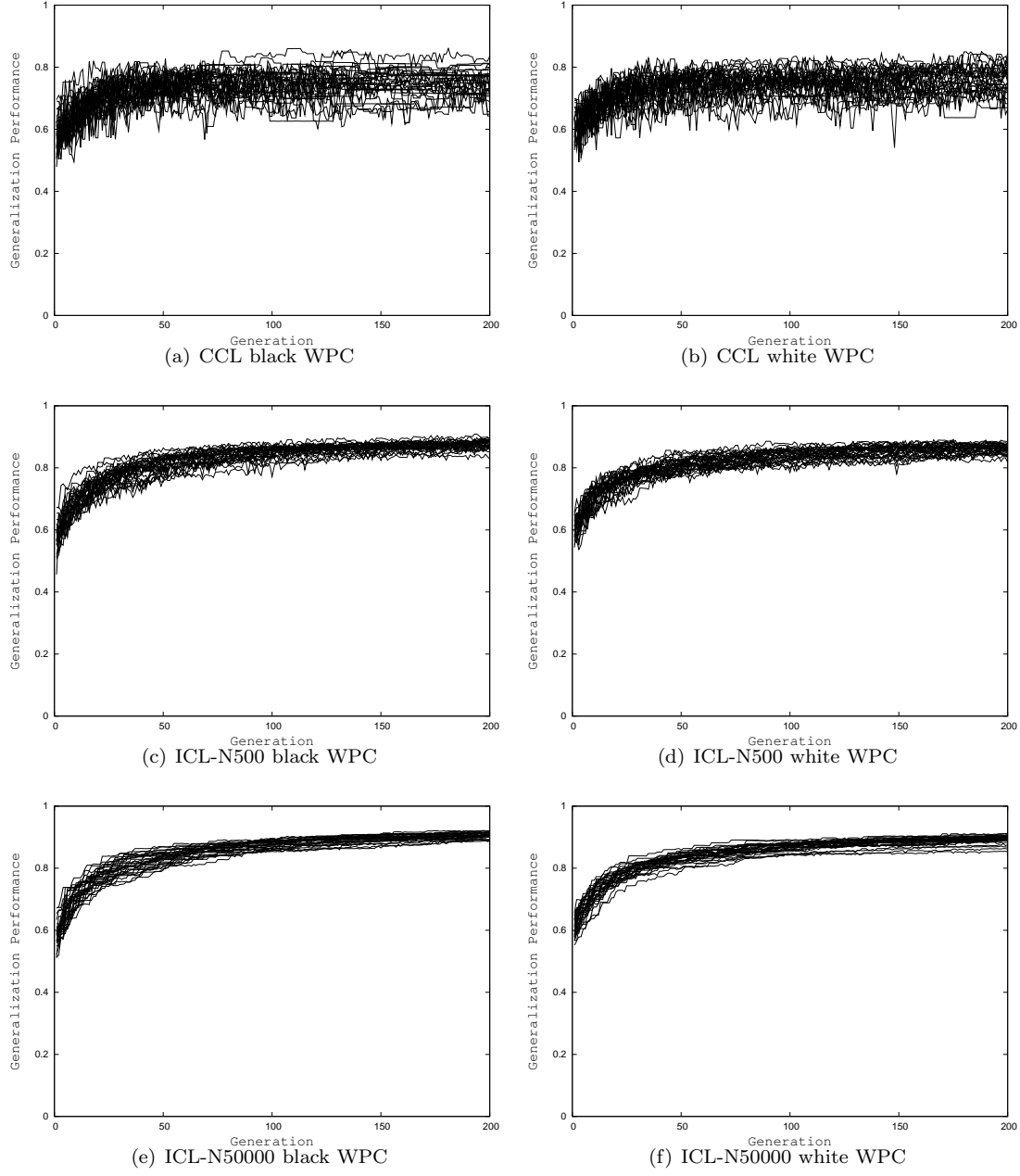


Figure 7: Comparison of CCL and different ICLs for the Othello game: (a) CCL black WPC, (b) CCL white WPC, (c) ICL-N500 black WPC, (d) ICL-N500 white WPC, (e) ICL-N50000 black WPC, and (f) ICL-N50000 white WPC. Shown are plots of the estimated generalization performance $\hat{G}_i(S_N)$ (with $N = 50000$) of the top performing strategy of the population throughout the evolutionary process for all 30 independent runs.

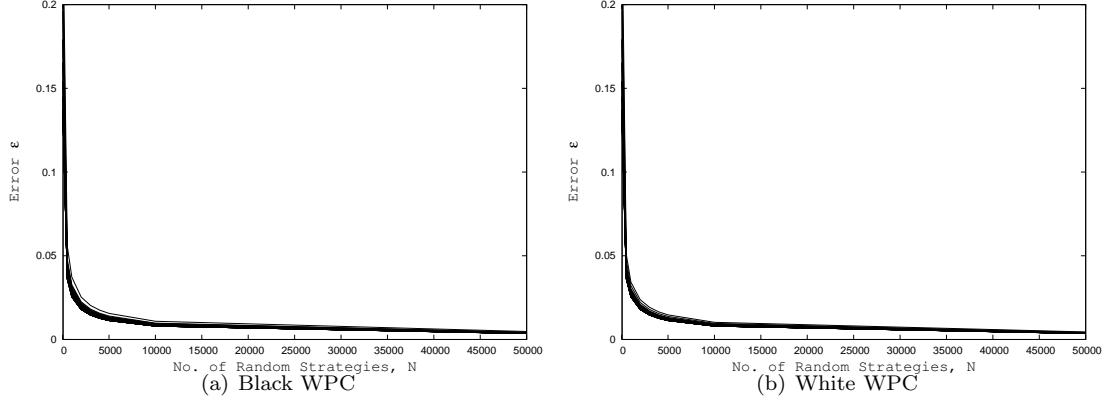


Figure 8: Pessimistic estimate of ϵ as a function of sample size N of test strategies for 50 random base strategies i for the Othello game.

Table 6: Summary of results for different co-evolutionary learning approaches for white Othello WPC taken at the final generation. For each approaches, the mean and standard error at 95% confidence interval over 30 runs are calculated. “Max” and “Min” give the maximum and minimum G_i over 30 runs. The t -tests compare CCL with ICLs.

Experiment	Mean \pm Std Err	Max	Min	t -test
CCL	0.758 ± 0.017	0.848	0.664	
ICL-N50	0.818 ± 0.007	0.851	0.771	-6.620^\dagger
ICL-N500	0.860 ± 0.006	0.883	0.818	-10.302^\dagger
ICL-N1000	0.873 ± 0.005	0.901	0.833	-12.944^\dagger
ICL-N5000	0.886 ± 0.006	0.906	0.842	-12.655^\dagger
ICL-N10000	0.890 ± 0.004	0.909	0.868	-15.112^\dagger
ICL-N50000	0.893 ± 0.005	0.911	0.853	-13.987^\dagger

† The t value 29 degrees of freedom is statistically significant at a 0.05 level of significance using a two-tailed test.

for both the IPD and Othello games. The new approach (ICL) can obtain strategies with higher generalization performance without large performance fluctuations and is faster compared to the case when a distribution-free framework is used, requiring an order of magnitude smaller number of test strategies to achieve similarly high generalization performance. More importantly, it is not necessary to use larger samples of test strategies when applying ICL to more complex games. One can observe the similarity in the rate at which the error ϵ decreases for increasing sample size N for both the IPD and Othello games (Figs. 2 and 8), and subsequently the similarity of the impact of using $\hat{G}_i(S_N)$ on the generalization performance of ICL (Figs. 4 and 7). We stress that one can use our approach to find and set the required number of test strategies for robust estimation in a principled manner before applying ICL to a new game.

We do note that there are many issues related to the design of co-evolutionary learning systems for high performance. For example, design issues can be problem-specific and involve representation, variation and selection operators [4, 7] as well as more sophisticated development of systems involving incorporation of domain knowledge [44] that have the potential to provide superior solutions compared to other learning approaches [8]. We only address the issue of selection (generalization estimates used to guide co-evolutionary search) in a principled manner that also can be implemented practically. Although fine-tuning parameters such as mutation rate and population size can have an impact on our numerical results, our general observations would hold. In addition, various selection and variation approaches can have different impact on generalization performance in co-evolutionary learning for different real-world problems (and games in particular). Here, it is of interests to use common tools for rigorous quantitative

analysis such as generalization measures we have formulated in [12]. As an example, we have previously studied both generalization estimates using unbiased sample of random test strategies (obtained through uniform sampling of \mathcal{S}) and biased sample of random test strategies that are superior in game-play and more likely to be encountered in a competitive setting (obtained through a multiple partial enumerate search). We have also recently started a preliminary investigation on the impact of diversity on the generalization performance of co-evolutionary learning [45].

5 Conclusion

We have addressed the issue of loose confidence bounds associated with the distribution-free (Chebyshev’s) framework we have formulated earlier for the estimation of generalization performance in co-evolutionary learning and demonstrated in the context of game-playing. Although Chebyshev’s bounds hold for any distribution of game outcomes, they have high computational requirements, i.e., a large sample of random test strategies is needed to estimate the generalization performance of a strategy as average game outcomes against test strategies. In this paper, we take advantage of the near-Gaussian nature of average game outcomes (generalization performance estimates) through the Central Limit Theorem and provide tighter bounds based on parametric testing. Furthermore, we can strictly control the condition (sample size under a given precision) under which the distribution of average game outcomes converges to a Gaussian through the Berry-Esseen Theorem.

These improvements to our generalization framework provide the means with which we develop a general and principled approach to improve generalization performance in co-evolutionary learning that can be implemented as an efficient algorithm. Ideally, co-evolutionary learning using the true generalization performance directly as the fitness measure would be able search for solutions with higher generalization performance. However, direct estimation of the true generalization performance using the distribution-free framework can be computationally expensive. Our new theoretical contributions that exploit the near-Gaussian nature of generalization estimates provide the means with which we can now: (1) find out in a principled manner the required number of test cases for robust estimations of generalization performance, and (2) subsequently use the small sample of random test cases to compute generalization estimates of solutions directly as the fitness measure to guide and improve co-evolutionary learning.

We have demonstrated our approach on the co-evolutionary learning of the IPD and the more complex Othello game. Our new approach is shown to improve on the classical approach in that we can obtain increasingly higher generalization performance using relatively small samples of test strategies and without large performance fluctuations typical of the classical approach. Our new approach also leads to faster co-evolutionary search where we can strictly control the condition (sample sizes) under which the speed-up is achieved (not at the cost of weakening precision in the estimates). It is much faster compared to the distribution-free framework approach as it requires an order of magnitude smaller number of test strategies to achieve similarly high generalization performance for both the IPD and Othello game. Note that our approach does not depend on the complexity of the game. That is, no assumption needs to be made about the complexity of the game and how it may have an impact on the required number of test strategies for robust estimations of generalization performance.

This study is a first step towards understanding and developing theoretically motivated frameworks of co-evolutionary learning that can lead to improvements in the generalization performance of solutions. There are other research issues relating to generalization performance in co-evolutionary learning that need to be addressed. Although our generalization framework makes no assumption on the underlying distribution of test cases (P_S), we have demonstrated one application where P_S in the generalization measure is fixed and known *a-priori*. Generalization estimates are directly used as the fitness measure to improve generalization performance of co-evolutionary learning (in effect, reformulating the approach as evolutionary learning) in this paper. There are problems where such an assumption has to be relaxed and it is of interests to us for future studies to formulate naturally and precisely co-evolutionary learning systems where the population acting as test samples can adapt to approximate a particular distribution that solutions should generalize to.

6 Acknowledgement

The authors would like to thank Profs. Simon Lucas and Thomas Runarsson for providing access to their Othello game engine that was used for the experiments of this paper. The authors would also like to thank the anonymous associate editor and reviewers for their comments that have helped to improve this paper. This work was partially supported by an EPSRC grant (No. GR/T10671/01) on “Market Based Control of Complex Computational Systems.”

References

- [1] X. Yao. Introduction. *Informatica (Special Issue on Evolutionary Computation)*, 18:375–376, 1994.
- [2] T. Bäck, U. Hammel, and H. -P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, 1997.
- [3] R. Axelrod. The evolution of strategies in the iterated prisoner’s dilemma. In L. D. Davis, editor, *Genetic Algorithms and Simulated Annealing*, chapter 3, pages 32–41. Morgan Kaufmann, New York, 1987.
- [4] K. Chellapilla and D. B. Fogel. Evolution, neural networks, games, and intelligence. *Proceedings of the IEEE*, 87(9):1471–1496, September 1999.
- [5] J. Noble. Finding robust texas hold’em poker strategies using pareto coevolution and deterministic crowding. In *Proc. of the 2002 International Conference on Machine Learning and Applications (ICMLA’02)*, pages 233–239, Nevada, USA, 2002.
- [6] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.
- [7] S. Y. Chong, M. K. Tan, and J. D. White. Observing the evolution of neural networks learning to play the game of othello. *IEEE Transactions on Evolutionary Computation*, 9(3):240–251, 2005.
- [8] T. P. Runarsson and S. M. Lucas. Coevolution versus self-play temporal difference learning for acquiring position evaluation in small-board go. *IEEE Transactions on Evolutionary Computation*, 9(6):540–551, 2005.
- [9] S. Nolfi and D. Floreano. Co-evolving predator and prey robots: Do “arms races” arise in artificial evolution? *Artificial Life*, 4(4):311–335, 1998.
- [10] S. G. Ficici and J. B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In C. Adami, R. K. Belew, H. Kitano, and C. Taylor, editors, *Artificial Life VI: Proc. of the Sixth Int. Conf. on Artificial Life*, pages 238–247, Cambridge, MA, 1998. The MIT Press.
- [11] R. A. Watson and J. B. Pollack. Coevolutionary dynamics in a minimal substrate. In *Proc. of the 2001 Genetic and Evolutionary Computation Conference (GECCO’01)*, pages 702–709, California, USA, 2001.
- [12] S. Y. Chong, P. Tiño, and X. Yao. Measuring generalization performance in co-evolutionary learning. *IEEE Transactions on Evolutionary Computation*, 12(4):479–505, 2008.
- [13] D. Ashlock and B. Powers. The effect of tag recognition on non-local adaptation. In *Proc. of the 2004 Congress on Evolutionary Computation (CEC’04)*, pages 2045–2051, 2004.
- [14] D. Shilane, J. Martikainen, S. Dudoit, and S. J. Ovaska. A general framework for statistical performance comparison of evolutionary computation algorithms. *Information Sciences*, 178(14):2870–2879, 2008.

- [15] B. V. Gnedenko and G. V. Gnedenko. *Theory of Probability*. Taylor & Francis, 1998.
- [16] W. Feller. *An Introduction to Probability Theory and Its Application, Volume 2*. Wiley, New York, 1971.
- [17] E. B. Manoukian. *Modern Concepts and Theorems of Mathematical Statistics*. Springer-Verlag, New York, 1986.
- [18] P. J. Darwen and X. Yao. On evolving robust strategies for iterated prisoner’s dilemma. In *Progress in Evolutionary Computation*, volume 956 of *Lecture Notes in Artificial Intelligence*, pages 276–292. Springer, 1995.
- [19] X. Yao, Y. Liu, and P. J. Darwen. How to make best use of evolutionary learning. In R. Stocker, H. Jelinck, B. Burnota, and T. Bossomaier, editors, *Complex Systems - From Local Interactions to Global Phenomena*, pages 229–242. IOS Press, Amsterdam, 1996.
- [20] J. L. Shapiro. Does data-model co-evolution improve generalization performance of evolving learners? In *The 5th International Conference on Parallel Problem Solving from Nature PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 540–549. Springer, Amsterdam, The Netherlands, September 27–30, 1998.
- [21] P. J. Darwen. *Co-evolutionary Learning by Automatic Modularization with Speciation*. PhD thesis, University of New South Wales, Sydney, Australia, November 1996.
- [22] P. J. Darwen and X. Yao. Speciation as automatic categorical modularization. *IEEE Transactions on Evolutionary Computation*, 1(2):101–108, 1997.
- [23] C. D. Rosin and R. K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.
- [24] E. D. de Jong and J. B. Pollack. Ideal evaluation from coevolution. *Evolutionary Computation*, 12(2):159–192, 2004.
- [25] S. G. Ficici and J. B. Pollack. A game-theoretic memory mechanism for coevolution. In *Proc. of the 2003 Conference on Genetic and Evolutionary Computation (GECCO’03)*, volume 2723 of *Lecture Notes in Computer Science*, pages 286–297. Springer-Verlag, 2003.
- [26] E. D de Jong. The incremental pareto-coevolution archive. In *Proc. of the 2004 Conference on Genetic and Evolutionary Computation (GECCO’04)*, pages 525–536, 2004.
- [27] E. D de Jong. The maxsolve algorithm for coevolution. In *Proc. of the 2005 Conference on Genetic and Evolutionary Computation (GECCO’05)*, pages 483–489, New York, USA, 2005.
- [28] R. P. Wiegand and M. A. Potter. Robustness in cooperative coevolution. In *Proc. of the 2006 Conference on Genetic and Evolutionary Computation (GECCO’06)*, pages 369–376, Seattle, USA, 2006.
- [29] K. L. Chung. *A Course in Probability Theory*. Academic Press, San Diego, third edition, 2001.
- [30] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.
- [31] P. J. Darwen and X. Yao. Co-evolution in iterated prisoner’s dilemma with intermediate levels of cooperation: Application to missile defense. *International Journal of Computational Intelligence and Applications*, 2(1):83–107, 2002.
- [32] S. Y. Chong and X. Yao. Behavioral diversity, choices, and noise in the iterated prisoner’s dilemma. *IEEE Transactions on Evolutionary Computation*, 9(6):540–551, 2005.
- [33] P. J. Darwen and X. Yao. Why more choices cause less cooperation in iterated prisoner’s dilemma. In *Proc. IEEE 2001 Congress on Evolutionary Computation (CEC’01)*, pages 987–994, Seoul, Korea, May 27–30, 2002.

- [34] X. Yao and P. J. Darwen. How important is your reputation in a multi-agent environment. In *Proc. IEEE 1999 Conference on Systems, Man, and Cybernetics (SMC'99)*, pages 575–580, Tokyo, Japan, October 12–15, 1999.
- [35] P. J. Darwen and X. Yao. Does extra genetic diversity maintain escalation in a co-evolutionary arms race. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 4(3):191–200, July 2000.
- [36] S. I. Resnick. *A Probability Path*. Birkhäuser, Boston, 1998.
- [37] D. B. Fogel. The evolution of intelligent decision making in gaming. *Cybernetics and Systems: An International Journal*, 22:223–236, 1991.
- [38] 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.
- [39] P. G. Harrald and D. B. Fogel. Evolving continuous behaviors in the iterated prisoner’s dilemma. *BioSystems: Special Issue on the Prisoner’s Dilemma*, 37:135–145, 1996.
- [40] N. Franken and A. P. Engelbrecht. Particle swarm optimization approaches to coevolve strategies for the iterated prisoner’s dilemma. *IEEE Transactions on Evolutionary Computation*, 9(6):562–579, 2005.
- [41] D. Ashlock, E.-Y. Kim, and N. Leahy. Understanding representational sensitivity in the iterated prisoner’s dilemma with fingerprints. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 36(4):464–475, 2006.
- [42] S. Lucas and T. P. Runarsson. Temporal difference learning versus co-evolution for acquiring othello position evaluation. In *Proceedings of the 2006 IEEE Symposium on Computational Intelligence and Games (CIG 2006)*, pages 52–59. IEEE Press, 2006.
- [43] K. -J. Kim, H. Choi, and S. -B. Cho. Hybrid of evolution and reinforcement learning for othello players. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games (CIG 2007)*, pages 203–209. IEEE Press, 2007.
- [44] K.-J. Kim and S.-B. Cho. Knowledge into evolutionary speciated checkers players. *IEEE Transactions on Evolutionary Computation*, 9(6):615–627, 2005.
- [45] S. Y. Chong, P. Tiño, and X. Yao. Relationship between generalization and diversity in co-evolutionary learning. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(3):214–232, 2009.