DEEP LEARNING TO PLAY GAMES

DANIELE CONDORELLI AND MASSIMILIANO FURLAN

Abstract. We train two independent neural networks to play normal-form games. At each iteration, we input a random bimatrix of payoffs into separate row and column networks, which output mixed strategies over individual actions. The parameters of each network are updated online via stochastic gradient descent to minimise their own instantaneous regret given the opponent's strategy. Our computational experiments demonstrate that the networks converge to approximate Nash equilibria across all games. For all 2×2 and in 80% of 3×3 games with multiple equilibria, they converge to the risk-dominant one. Our results show how Nash play can emerge from learning across heterogeneous games.

1. Introduction

Nash equilibrium is the most widely used solution concept in applications of game theory. However, concerns remain about its conceptual foundations. Even if players are mutually certain of their ability to best respond, what guarantees correct beliefs are formed about their opponents's behaviour? One way to resolve this issue is to view Nash play as the limit outcome of a learning or evolutionary process. For example, if best responding agents interact repeatedly in a given strategic game and form beliefs about their opponents' behaviour based on empirical observations of outcomes, their play converges to a Nash equilibrium, if it converges at all.

The learning (or evolutionary) explanation is canonical, but not fully satisfactory. Convergence may require many periods, yet individuals rarely engage in repeated play of the same game. Moreover, Nash equilibrium retains predictive power even when human subjects encounter a game for the first time (e.g., see Goeree and Holt (2001)). To address these criticisms, it is common to informally hypothesise that humans learn by drawing analogies from playing different games. As Fudenberg and Levine (1998) note, "our presumption that players do extrapolate across games [...] is an important reason to think that learning models have some relevance in real-world situations." [1]

In this paper, we propose a formal theory of learning across games and provide evidence that supports Nash play as a limit result, without postulating an exogenous notion of similarity among games or making agents play any one game repeatedly. Computationally, we demonstrate that the behaviour of two independent neural networks, trained to minimise instantaneous regret (i.e., the payoff loss from not best responding to the opponent's strategy) while playing a sequence of

---

September 6, 2024 — Preliminary draft. Daniele Condorelli: d.condorelli@warwick.ac.uk; Massimiliano Furlan: massimiliano.furlan@warwick.ac.uk. We thank Francoise Forge, Alkis Georgiadis-Harris, Marco Li Calzi, Francesco Nava, Balazs Szentes and Bernhard von Stengel for inspiring discussions. We are additionally grateful to Bernhard for sharing his code to perform the Harsanyi-Selten linear tracing procedure. We also thank seminar audiences at the University of Warwick, HKU, University of Bergamo and Royal Holloway Theory Conference. The code is available at https://github.com/massimilianofurlan/nn_bimatrix_gamess.

[1] Similarly, Kreps (1990) writes: "If we rely solely on the story of directly relevant experience to justify attention to Nash equilibria, and if by directly relevant experience we mean only experience with precisely the same game in precisely the same situation, then this story will take us very little distance outside the laboratory. It seems unlikely that a given situation will recur precisely in all its aspects, and so this story should be thought of as applying to a sequence of 'similar' situations."

randomly generated normal-form games, appears to converge towards a specific selection from the (approximate) Nash correspondence.

We define a neural network player as a differentiable function $f_n^i(\cdot; w)$ parameterised by a vector of weights w, which maps two-player normal-form $n \times n$ games (i.e., an ordered list of $2n^2$ payoffs) into a mixed strategy for player $i = \{1, 2\}$. The row (player 1) and column (player 2) networks are trained by a dynamic adversarial process that updates their arbitrarily initialised weights through a very large number of iterations. In each period a game is randomly drawn from a set that encompasses virtually all possible preferences over lotteries on the outcomes, and both networks output their mixed strategy plays. If a network does not best respond to the opponent's strategy, its weights are slightly adjusted via stochastic gradient descent in the direction that reduces regret, with the adjustment of each weight proportional to the size of the derivative.[2]

Once training of our fine-tuned models is complete, we evaluate their performance on a new set of randomly generated games. Our main finding is that the networks learn to closely approximate Nash equilibrium in nearly every game. An ϵ-Nash equilibrium is achieved if the maximum regret experienced by either player is below ϵ. We show that the average maximum regret (MaxReg) in the test set, which comprises randomly generated games including a substantial fraction with a unique mixed equilibrium, is nearly zero in the 2×2 test games and less than 0.03 in 3×3 games, after training that lasts for over one trillion games.[3] For comparison, when the networks play each action with equal probability, the average maximum regret is 0.66 in 2×2 games and 0.68 in 3×3. To confirm our finding, we also examine the maximum total variation distance across players between network play and the closest Nash equilibrium (MaxDistNash), where total variation distance measures the maximum difference in the probabilities assigned to any one action by two distributions. We find that both networks map games into mixed strategies with maximum total variation from the closest Nash equilibrium smaller than 0.1 (0.05) in 98% (97%) of 2×2 games and in 85% (75%) of 3×3 games. Both metrics show slight improvement when we consider dominance-solvable games but deteriorate when restricted to games with a single mixed equilibrium. The key results is presented in Table 1.

2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.12	0.75
Mean MaxReg	0.00	0.02	0.00
	(0.01)	(0.02)	(0.00)
Mean MaxDistNash	0.01	0.03	0.01
	(0.05)	(0.05)	(0.05)

3×3 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.21	0.58
Mean MaxReg	0.03	0.09	0.01
	(0.08)	(0.10)	(0.06)
Mean MaxDistNash	0.06	0.12	0.04
	(0.14)	(0.14)	(0.14)

TABLE 1. Key results: Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

[2]The neural network functional form f is flexible enough to arbitrarily approximate any function arbitrarily closely by selecting a sufficiently large parameter vector (e.g., see Hornik et al. (1989)). Since the network player has access to the entire payoff matrix in making its play recommendation, the learning dynamics are not uncoupled in the sense of Hart and Mas-Colell (2003). Thus, the learnability of Nash equilibrium is not precluded.

[3]There are roughly 50 trillion distinct 3×3 games, just considering all possible combinations of weak ordinal preferences over outcomes for two players. Taking into account cardinal payoffs and machine precision in generating pseudo-random numbers, the probability of encountering the same game twice in our training samples is essentially zero.

Through joint experience each player learns to approximately predict the opponent's behaviour based on the strategic environment and best respond to it. During training, the players' regret decreases, first exponentially and then following a power law, supporting the theoretical hypothesis that two neural networks playing adversarially converge toward computing their respective parts of a certain selection from the Nash equilibrium in almost all games as both the number of games played and the network sizes increase. We do not prove our conjecture formally, but we hope the computational experiment we performed will motivate further theoretical work. Remarkably, there are, to our knowledge, no existing results guaranteeing convergence for adversarial neural networks playing specific non-zero sum games, let alone for networks attempting to coordinate on a solution concept.

Albeit at a slower pace, the networks learn to play mixed Nash equilibria in games with a unique mixed equilibrium. The slower pace can be attributed to at least two well-understood reasons. First, mixed equilibria are unstable. As long as the opponent's strategy is even slightly different from the equilibrium one, mixing entails positive regret. Second, computing mixed Nash equilibria is, in formal sense, more complex than identifying pure equilibria. Even though learning cannot stop unless both networks are playing their part of a mixed Nash equilibrium in games with only one mixed equilibrium, the reasons why converge appears to be successful are harder to pin down. Intuitively, dynamics with the flavour of those hypothesised by Fudenberg and Kreps (1993) might be at play, although in a larger space. In their work, convergence is made possible in any given game by players best responding in a non-deterministic way. This is the case here in the following sense. Because neural networks are optimally chosen continuous functions on the space of all games, as we move from one game to another there is a smooth transition from best responding with one action to best responding with the other.

The observation that thenetworks learn to play Nash equilibria raises the question of which equilibria are being selected when multiple exist. Our results show that in almost all 2×2 games and in 80% of 3×3 games with multiple equilibria, the networks select the risk-dominant equilibrium, as defined by Harsanyi and Selten (1988). Moreover, consistently with the risk dominant selection criterion, mixed equilibria are only played in games with a single mixed equilibrium and in few other exceptional cases. These findings reinforce arguments coming from evolutionary models (e.g., Kandori et al. (1993) and Young (1993)) in favour of selecting risk-dominant equilibria in 2×2 games and suggest some of these results may extend to populations of players jointly evolving a solution concept. On the other hand, they motivate caution toward viewing the linear-tracing refinement as the result of evolutionary forces in 3×3 games or larger.

We also tested the selection criterion implemented by the networks against various natural axioms for single-valued solution concepts. We found that the same equilibrium is selected when the role of players is swapped, suggesting that the networks learn approximately identical behaviour despite not having an identical experience. Additionally, the played equilibrium remains the same when the order of actions is permuted or when payoffs are transformed in ways that do not alter best reply correspondences or that only raise them at the equilibrium point. Lastly, we demonstrate that the same equilibrium is selected in 2×2 games and 3×3 ones built from those 2×2 by adding one strictly dominated action for each player. This consistency between models of different sizes further demonstrates robustness of the learned behaviour.

After discussing the relevant literature in Section 2, the rest of the paper is organised as follows. In Section 3, we provide a more technical description of our model. In Section 4 we present the results of our baseline specification, including its nearly perfect conformity to the precepts of rationalizability. In Section 5, we explore the dynamics of learning, highlighting the speed of

convergence to Nash and the different dynamics involved in learning to play pure Nash equilibria and mixed ones. Section 6 supports the convergence hypothesis by elaborating on the robustness of our results. We show that they remain largely unaffected by training models for larger games, employing alternative model specifications, such as different network architectures, departing from the assumption that the entire mixed strategies of the opponent are observed, and sampling on smaller sections of the space of games. Finally, Section 7 concludes with additional thoughts and potential directions for future research.

2. Literature Review

This paper has precursors. To our knowledge, the idea of using a regret-based approach to train a neural network is first discussed in Marchiori and Warglien (2008) for a set of fixed games. But closest to ours is the work of Spiliopoulos (2011, 2012). He used an adversarial approach to jointly train a population of neural networks to minimise the distance from best responding using randomly generated 2×2 and 3×3 games. While we share a similar setup, he focuses on pure equilibria and his results do not support the general hypothesis of convergence to Nash. For 3×3 games with a unique pure Nash equilibrium, he finds a 62% frequency of (ex-post) Nash play, against 96% in our case using his benchmark. Likely due to the limited availability of computational power at that time, the author concludes that neural networks are learning behavioural heuristics different from Nash.[4]

In their pioneering work, Sgroi and Zizzo (2009) trained a single neural network to identify Nash equilibrium in 3×3 games. In contrast to us, the network was trained via supervised learning on random games with only one Nash equilibrium, with the aim to minimise the output's distance from the Nash equilibrium strategy. Sgroi and Zizzo (2009) are not preoccupied with the result of an adversarial learning process, but focus on the ability of the network to find the Nash equilibrium. While they concluded that neural networks would be unlikely to be able to learn Nash, recent engineering research on the learnability of equilibria, e.g, Liu et al. (2024) and Duan et al. (2023), reaches conclusion in line with our finding.

A handful of papers have explored theoretical models where learning takes place through a sequence of randomly generated games. In a seminal contribution, LiCalzi (1995) studied fictitious-play dynamics. In his model, agents best respond upon observing which game they are playing, but beliefs over an opponent's behaviour are formed by pooling their actions in all past games.[5] Steiner and Stewart (2008) model the play of games have never been seen before by equipping the space from which games are drawn with a similarity measure. Players best respond to their learned belief about the behaviour of opponents, which is determined by a weighted average of behaviour on past play based on the measure of closeness between games. Mengel (2012) studies players engaging in adversarial reinforcement learning over both how the best partition a finite space of games, subject to a convex cost of holding finer partitions, and which behaviour to adopt in each element of the partition. Her approach with its endogenous partitioning of games is close in spirit to ours. However, her assumption that the set of possible games is finite allows learning to take place game-by-game when partitioning costs are small, which is in contrast to both Steiner and Stewart (2008) and us.

[4]A different machine learning angle is pursued in some recent work where neural networks, Hartford et al. (2016), or related algorithms, Fudenberg and Liang (2019), are trained on existing experimental data with the aim of predicting human behaviour.

[5]Relatedly, a substantial literature has also studied players who form a coarse view (or have a misspecified model) of the behaviour of the opponents. See Jehiel (2005) for a recent survey.

Our neural network provides a unique play recommendation for any possible game as a result of a competitive learning process. An analogous approach has been followed by others relying on different methodologies. In Selten et al. (2003), competing groups of students were asked to write programs able to offer a play recommendation for any game. The programs were then competitively tested in a set of randomly generated games and feedback was provided to the students. Programs were updated by the students and tested again, and the process was repeated for for five iterations. The software produced by the students in this fashion in the course of a semester ended up failing to compute Nash equilibrium in games with only one mixed strategy but achieved 98% success in choosing a pure Nash in dominant solvable games. When faced with multiple pure equilibria, the utilitarian one was favoured. Recently, Lensberg and Schenk-Hoppé (2021) have pursued a related idea computationally, but using genetic algorithms rather than students. A randomly mutating population of rules to play games compete at each iteration and more successful rules reproduced more. While Lensberg and Schenk-Hoppé (2021) agree with us regarding the selection of the risk-dominant equilibrium in 2×2 games, both they and Selten et al. (2003) conclude, in contrast to us, that the identified average heuristic at convergence is not a refinement of Nash.

Other authors have also studied the evolution of heuristics to play games. Samuelson (2001) characterises the choice, subject to complexity costs, of a finite automaton for playing random games chosen from three classes: bargaining, ultimatum and tournament. Stahl (1999) proposes a theory where a set of exogenous behaviour rules is reinforced based on their success. Relatedly, Germano (2007) considers evolutionary dynamics for a given set of exogenous heuristics, with selection based on the success over a distribution of random games. Among other features, our learning differentiates from Samuelson (2001) as the set of games we deal with is large, and from Stahl (1999) and Germano (2007) because the set of possible rules of play is unrestricted.

Finally, our work complements (and is supported by) the sparse experimental evidence that empirically demonstrates the ability of human subjects to extrapolate across games. Among others, the experiments of Cooper and Kagel (2003), Grimm and Mengel (2009), Devetag (2005), Marchiori and Warglien (2008) and Marchiori et al. (2021) all show strategic abilities can be learned and transferred to games that were not faced in the learning stage.

3. Training neural networks

We begin with some basic game theoretic definitions, which for ease of subsequent exposition we present with non-standard notation. Then we define the neural network architecture we employ and finally, we explain how training takes place.

Basic Definitions. We define $n \times n$ two-player strategic game G as a pair of real-valued $n \times n$ matrixes (G^1, G^2), where each element (j,k) of G^i indicating the payoff of player $i = 1, 2$ (row and column) when i plays action $j \in \{1, \ldots, n\}$ and the opponent chooses action $k \in \{1, \ldots, n\}$. We restrict attention to games where the vectorised payoff matrix of each player is a point in the n-radius sphere embdedded in the (n^2-1)-dimensional subspace orthogonal to $\mathbf{1} = (1, \ldots, 1)^\top \in \mathbb{R}^{n^2}$.[6] This restriction bounds payoffs between $-\sqrt{n^2-1}$ and $\sqrt{n^2-1}$ and reflects the assumption that preferences, rather than utilities, are the primitive of the game theoretic decision. In fact, there exists a bijection between Von-Neumann and Morgenstern preferences over lotteries on the $n \times n$ pure strategy profiles and a utility representation in the specified set. We denote by $\mathcal{G}_n$ the space of all such $n \times n$ two-player games.

[6]Formally, the space of payoffs for each player is $\{x \in \mathbb{R}^{n^2} : ||x||_2 = n \text{ and } \mathbf{1}^\top x = 0\}$.

Let $\sigma^i \in \Delta^{n-1}$ denote a mixed strategy for player $i = 1, 2$, represented as an $n \times 1$ vector over the n actions of that player. As usual, let σ^{-i} denotes a mixed strategy for the opponent. If the played profile of strategies is (σ^1, σ^2), then the payoff of player i is given by $(\sigma^i)^\top G^i \sigma^{-i}$. Define the *instantaneous regret* of player i in game G for profile (σ^1, σ^2), denoted $R^i(G, \sigma^1, \sigma^2)$, as the difference between the highest payoff delivered in game G by any pure strategy and that obtained by σ^i for given σ^{-i}. In matrix notation

$$R^i(G, \sigma^1, \sigma^2) = \max_{j \in \{1,2,\ldots,n\}} [G^i \sigma^{-i}]_j - (\sigma^i)^\top G^i \sigma^{-i}. \tag{1}$$

A profile (σ^1, σ^2) is a Nash equilibrium of game G if for $i = 1, 2$ we have $R^i(G, \sigma^1, \sigma^2) = 0$. Since our approach is computational and we are concerned with convergence to equilibrium, it is useful to introduce the notion of an approximate Nash equilibrium. We will say that a profile (σ^1, σ^2) is an ϵ-Nash equilibrium (or simply an ϵ-equilibrium) of game G if $R^i(G, \sigma^1, \sigma^2) \leq \epsilon$ for $i = 1, 2$.

Neural Networks. A n-action game-playing neural network for player i is a function

$$f_n^i(\cdot; w) : \mathcal{G}_n \to \Delta^{n-1},$$

where w is a vector of trainable parameters that defines the state of the network. The functional form f^i is determined by the network architecture. We use a multi-layer feed-forward network composed of an input layer, L (≥ 1) hidden layers of dimension d ($> 2n^2$), and an output layer, all fully connected. Additional details are provided next but can be skipped by the reader familiar with the underlying mathematics.

The *input layer* receives a bimatrix game, vectorises it into a $2n^2$-dimensional vector, denoted $\text{vec}(G)$, and linearly transforms it into another vector of dimension $d > 2n^2$. Then, a so-called ReLU (rectified linear) activation function that operates elementwise is applied. More formally, the input layer returns

$$h^{(0)}(G) = \max\left\{0,\, W^{(0)}\text{vec}(G) + b^{(0)}\right\},$$

where max operaters elementwise and $W^{(0)} \in \mathbb{R}^{d \times 2n^2}$ and $b^{(0)} \in \mathbb{R}^d$ are the trainable parameters of the input layer.

Each of the *hidden layers* $l \in \{1, \ldots, L\}$ receives the d-dimensional output vector of the preceding layer $h^{(l-1)}$ as input and applies to it a dimension-preserving linear transformation followed by a ReLU activation. That is,

$$h^{(l)}(h^{(l-1)}) = \max\left\{0, W^{(l)} h^{(l-1)} + b^{(l)}\right\},$$

where $W^{(l)} \in \mathbb{R}^{d \times d}$ and $b^{(l)} \in \mathbb{R}^d$ are the parameters of the l-th hidden layer.

Finally, the *output layer* tranforms the d-dimensional output of the last hidden layer $h^{(L)}$ to a vector in n dimensions. Formally, the output layer returns

$$y(h^{(L)}) = W^{(L+1)} h^{(L)} + b^{(L+1)},$$

where $W^{(L+1)} \in \mathbb{R}^{d \times n}$ and $b^{(L+1)} \in \mathbb{R}^n$ are the parameters of the layer.

To obtain a probability distribution over actions, the output layer is then mapped into the $n-1$ dimensional simplex through a softmax activation function. Formally, we transform the output vector y using

$$\text{softmax}(y) = \frac{e^y}{\sum_{i=1}^n e^{y_i}},$$

with $\mathbf{e}^y = (e^{y_1}, \ldots, e^{y_n})$.

Putting all together, the neural network $f_n^i(G, w)$ with L hidden layers of dimension d is

$$\text{softmax}\left(W^{(L+1)}\max\left\{0, W^{(L)}\max\left\{0, \cdots \max\left\{0,\ W^{(0)}\text{flat}\,(G) + b^{(0)}\right\}\cdots\right\} + b^{(L)}\right\} + b^{(L+1)}\right).$$

By stacking the parameters together in $w = \text{vec}(W^0, b^0, ..., W^{L+1}, b^{L+1})$ and counting, we see that the network $f_n^i(\cdot, w)$ has a total of $Ld^2 + (2n^2 + n + L + 1)d + n$ weights. Being the composition of continuous and almost always differentiable functions, it is continuous and almost always differentiable.

The working of a neural network with one hidden layer for 2×2 games is illustrated in Figure 1.

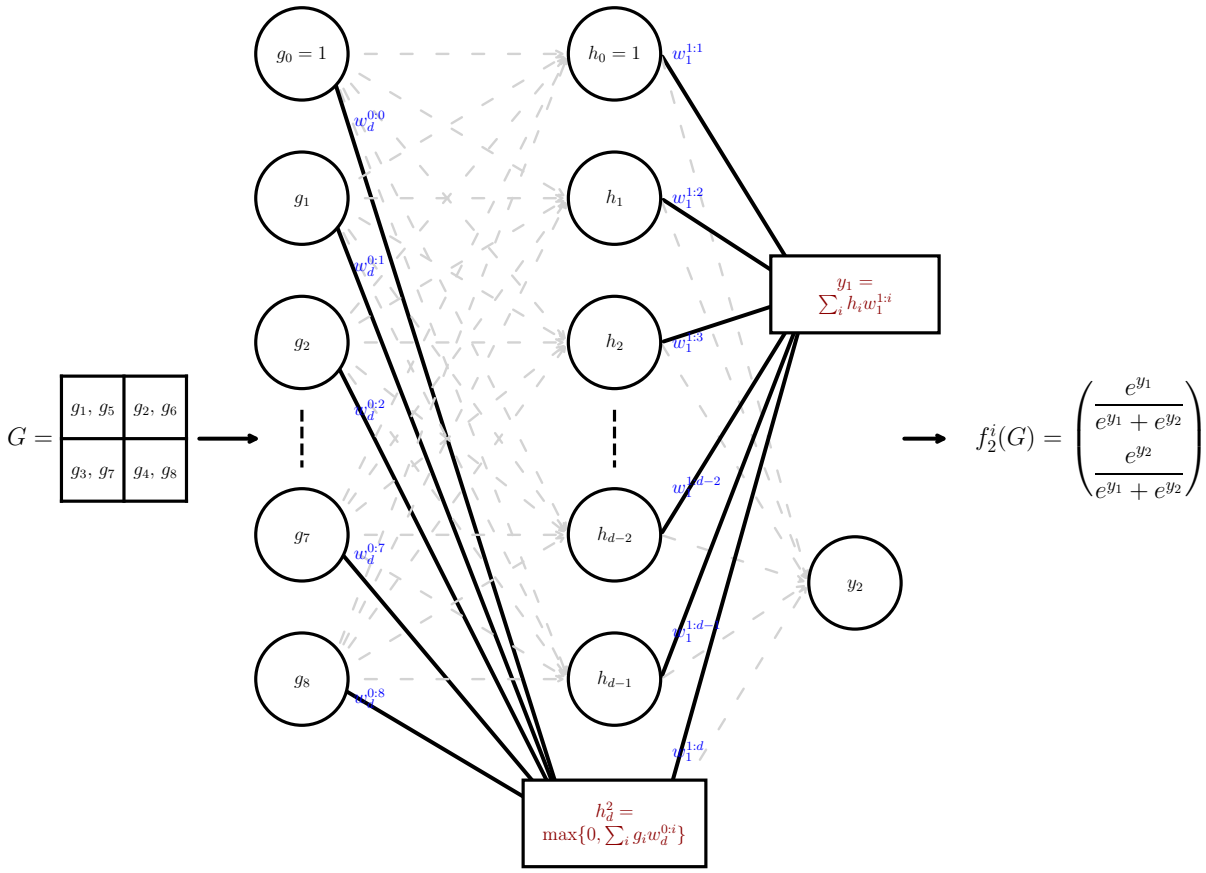

FIGURE 1. Working of a game playing neural network with one hidden layer. Input nodes $g_1, \ldots, g_n$ receive the eight payoffs of game G; nodes in the hidden layer, $h_1, \ldots, h_d$, and nodes in the output layer, y_1, y_2 perform computations. Softmax turns the output into a probability distribution $f_2^i(G; w)$ over actions. Emphasis is on two rectangular nodes, the bottom of the second layer, h_d^2, and the top of the output one, y_1. In red the computations performed by the node and in blue the trainable weights that are directly involved in the computations $w_d^{0:1}, \ldots, w_d^{0:8}$ and $w_1^{1:1}, \ldots, w^{1:d}$. The other connections between nodes are represented by shaded lines.

Training. At the outset, two independent networks to play $n \times n$ games are initialised with arbitrary parameters w_0^1 and w_0^2. Then, weights are updated dynamically as follows. In each period $t = 1, ..., T$ a game G_t is sampled from $\mathcal{G}_n$ according to some fixed distribution and, given their parameters at time t, denoted w_t^1 and w_t^2, the two networks generate a play recommendation

in G_t for both the row player, $f_n^1(G; w_t^1)$, and the column one, $f_n^2(G; w_t^2)$. Then, the weights for each network i are updated via stochastic gradient descent on loss function $\mathscr{L}^i\left(G, \sigma^i, \sigma^{-i}\right)$ (to be specialised later) evaluated at the network's i output strategy $\sigma^i = f^i(G_t; w_t^i)$ and at the period's data, that is the payoff matrix $G = G_t$ and the play of the opponent $\sigma^{-i} = f^{-i}(G_t; w_t^{-i})$).

Formally, for networks $f_n^1(G; w_0^1), f_n^2(G; w_0^1)$ and initial w_0^1 and w_0^2 updating of weights takes place according to

$$w_{t+1}^i = w_t^i - \eta_t \nabla_{w_t^i} \mathscr{L}^i(G_t, f_n^i(G_t; w_t^i), f_n^{-i}(G_t; w_t^{-i})) \quad i = 1, 2, \quad t = 0, \ldots, T.$$

where $\nabla_{w_t^i}$ denotes the gradient with respect to the weights of network i and η_t is the shared learning rate, which we assume declines exponentially during training according to $\eta_t = \alpha^t \eta_0$.[7] Intuitively, the learning rate establishes how important is the current experience vis-a-vis the past and may decline as more experience is accumulated.

Instead of updating the networks' weights using the gradient of the loss for a single game, weights can be updated differentiating average losses over batches of $B(\geq 1)$ games. Batching has a natural interpretation in terms of accumulating experience before adjusting behaviour, and is common practice in machine learning to optimise learning efficiency and stability. When $B > 1$, a group B of games is sampled in each period and a total number of $K = T \times B$ is played.

This weights updating process is repeated until all games up to T are exhausted. We call f_n^i the network i trained to play $n \times n$ games, omitting reference to the trained weights and the hyperparameters.

4. Results in the baseline model

In this section, we report the results we obtained from the training of our baseline model. Before, we discuss the model we use and testing.

Baseline Model. The feedforward network architecture we choose is the simplest in the practice of deep learning. For 2×2 games, we use a fully connected neural network with 8 hidden layers of 1024 neurons each, and for 3×3 games, a network with 8 hidden layers of 2048 neurons each. The total number of parameters in the networks is 8,408,066 for 2×2 games and 33,615,875 for 3×3 games. The initial learning rate, η_0, is set to 1 and decay exponentially with decay rate, α, equal to 0.999999 for 2×2 games and 0.9999995 for 3×3 games. The choice of hyperparameters, including the number of layers, neurons, learning rates, and decay rates, was determined through a process of trial and error to balance complexity and performance. While the speed of learning and ultimate performance are affected by the choice of the network architecture and its hyperparameters, in section 5 we show that the main insights we obtain are robust to alternative architectural choices.

We trained separate pairs of players for 2×2 and 3×3 games. From $\mathcal{G}_2$ and $\mathcal{G}_3$ we sampled, respectively, 2^{30} ($\approx$ 1 trillion) and 2^{31} ($\approx$ 2 trillions) games uniformly at random. The mean individual payoff in both classes is zero and the variance is $n^2/(n^2 - 1)$. Batching took place in groups of 128 games for 2×2 players and 256 for 3×3. Albeit substantial learning takes place with much fewer games, we opted to present results for the minimal training scale that delivered convincing evidence on the convergence of the algorithms. Uniform sampling was chosen for simplicity and guarantees that there are no degenerate games in our training set. As we show in the robustness section, the sampling method is not crucial.

[7] The network composite functional form f makes the calculation of the gradient parallelizable and not computationally expensive via an algorithm known as backpropagation. For example, in the network from Figure 1, $\frac{\partial y_1}{\partial w_1^{1:i}} = h_i$ and $\frac{\partial y_1}{\partial w_d^{0:i}} = w_1^{1:d} g_i$ if $h_d^i > 0$ and 0 otherwise.

In the baseline scenario, we employ a loss function that is half of the square of instantaneous regret (i.e. forgone payoff from not best responding to the mixed strategy played by the opponent). That is $\mathcal{L}^i(G, \sigma^i, \sigma^{-i}) = -R(G^i, \sigma^i, \sigma^{-i})^2/2$ for $i = 1, 2$. Because it is necessary to fix the strategy of the opponent to evaluate the counterfactual (i.e., how a change of action would lead to a different loss) the choice of a monotone transformation of instantaneous regret is natural. Squaring makes the gradient proportional to the experienced regret. Moreover, we assume the entire mixed strategy of the opponents is observed when the loss is evaluated. Both this and the previous squaring assumption speed up learning of mixed strategies in a nontrivial way. The former is because convexifying regret penalises pure strategies that are prone to produce larger regret on average when playing against a mixed strategy of the opponent. The latter is because it implies the learner can rely on more accurate data on how each game is played by the opponent. Nonetheless, in the robustness section, we show that learning to play Nash equilibria does not depend on these assumptions, by training the networks based on linear regret and on a loss defined based on a single pure action drawn from $f^{-i}(G)$.

The details of the neural network architecture, data generation, and the training process are summarised in the Table 2 below for both the 2×2 and 3×3 models.

2×2 Games

Network	Data	Optimization
$L = 8$	$G^i \sim \text{Uniform}(\mathcal{G}_2)$	$\mathcal{L}^i(G, \sigma^i, \sigma^{-i}) = -R(G^i, \sigma^i, \sigma^{-i})^2/2$
$d = 512$	$K = 1\,073\,741\,824$	$\text{T} = 8\,388\,608$
$\#w = 2\,106\,882$	$\text{B} = 128$	$\eta_0 = 1, \alpha = 0.999999$

3×3 Games

Network	Data	Optimization
$L = 8$	$G^i \sim \text{Uniform}(\mathcal{G}_3)$	$\mathcal{L}^i(G, \sigma^i, \sigma^{-i}) = -R(G^i, \sigma^i, \sigma^{-i})^2/2$
$d = 2048$	$K = 2\,147\,483\,648$	$\text{T} = 8\,388\,608$
$\#w = 33\,615\,875$	$\text{B} = 256$	$\eta_0 = 1, \alpha = 0.9999995$

TABLE 2. Summary of baseline models. L: number of layers; d: number of neurons per layer; $\#w$: number of weigths; T: number of optimization steps; B: batch size; K: total numer of games, $\mathcal{L}^i$: loss function, η_0: initial learning rate; α: learning rate decay.

Testing. To test our models and benchmark them against game-theoretic solutions, we generated random test sets of 2×2 and 3×3 games, each containing 2^{17} (131,072) games. We verified that in both cases empirical mean and variance of individual payoffs and the distribution of the numbers of Nash equilibria are close to those in the training set. For each game in these sets, we computed strictly dominated strategies for both players, pure strategy profiles surviving iterated elimination of strictly dominated strategies (i.e., the set of rationalisable profiles), and the set of Nash equilibria. For the case where there are multiple equilibria, we also computed the risk-dominant equilibrium selected by the Harsanyi and Selten (1988) linear tracing procedure and the utilitarian equilibrium.[8]

For all games in the test sets and the zero-sum test sets we generated the predictions of the trained row and column players and compared these to the above benchmarks. To do so, we make use of the notion of regret and maximum regret across players (MaxReg) in a game. The latter

[8]We used the pygambit python repository for computing Nash equilibria https://github.com/gambitproject/gambit. We used software provided by Bernhard Von Stengel to implement the linear tracing procedure that finds risk dominant equilibria.

indicates the maximal payoff-distance between networks'output strategies and best responses to the play of the opponent in the game. MaxReg in game G is the minimal ϵ such that the networks are playing an ϵ-equilibrium in G. Moreover, we evaluate distance in strategy space using the following definitions based on total variation. We say two strategies σ^i and $\tilde{\sigma}^i$ are δ-distant (or close) if $\sup_j |\sigma^i_j - \tilde{\sigma}^i_j| = \delta$. In words, strategies σ^i and $\tilde{\sigma}^i$ are δ-distant, or, equivalently, have total variation equal to δ, if the largest absolute difference in probability mass assigned to the same pure strategy is equal to δ. A strategy profile (σ^1, σ^2) is δ-distant from another profile $(\tilde{\sigma}^1, \tilde{\sigma}^2)$ if the the maximum distance between any two individual strategies in the profiles is δ. We define the maximal distance from Nash across players (MaxDistNash) in game G as the smallest δ such that the profile of strategies played by the networks in G is δ-distant to a Nash of G.[9]

Nash play. In Table 3 below we report our key statistics on the behaviour of the networks trained on 2×2 and 3×3 games, for all games in the test sets and for games grouped by the number of pure equilibria. Figure 2 displays the distribution of MaxReg achieved over the test set for all games and for games with only mixed Nash equilibria, for 2×2 and 3×3 games.

2×2 Games

	All Games	**# Pure Nash**		
		0	**1**	**≥ 1**
Fraction of 2^{17} Games	1.000	0.124	0.750	0.876
Mean MaxReg	0.003	0.019	0.000	0.001
	(0.010)	(0.017)	(0.001)	(0.012)
Mean MaxDistNash	0.009	0.032	0.006	0.002
	(0.051)	(0.053)	(0.053)	(0.020)

3×3 Games

	All Games	**# Pure Nash**		
		0	**1**	**≥ 1**
Fraction of 2^{17} Games	1.000	0.215	0.579	0.785
Mean MaxReg	0.029	0.085	0.013	0.015
	(0.080)	(0.097)	(0.061)	(0.081)
Mean MaxDistNash	0.056	0.122	0.042	0.029
	(0.137)	(0.141)	(0.138)	(0.106)

TABLE 3. Key results: Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

In 2×2 games, we obtain compelling results supporting convergence to Nash. The networks obtain an average maximal regret below 0.003 in the test set, compared to an average maximal regret from random play of 0.66. By looking at the distribution of the maximum regret obtained across the two players, we see the networks are playing an ϵ-Nash equilibrium with $\epsilon \leq 0.041$ in 99% of games. While ϵ-Nash equilibrium is the commonly used notion of an approximate equilibrium, it does not necessarily imply players are playing close to a Nash in strategy space. We confirm

[9]For instance, MaxDistNash is $0.05\overline{3}$ in rock-scissor-paper if player 1 is using strategy $(0.28, 0.33, 0.35)$ and player 2 is plays $(0.33, 0.33, 0.34)$

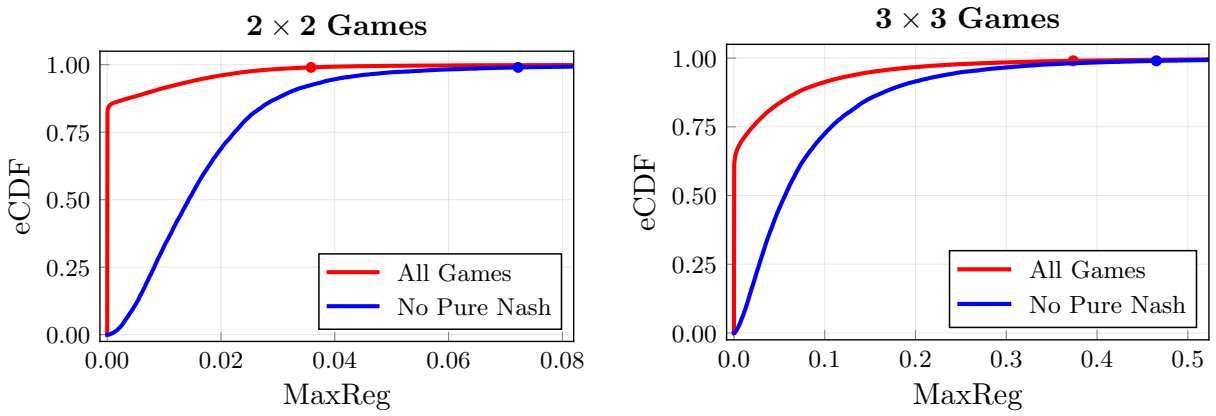

FIGURE 2. Empirical cumulative density of max regret across players in 2×2 and 3×3 games. Dots are placed in correspondence of the 99 percentile.

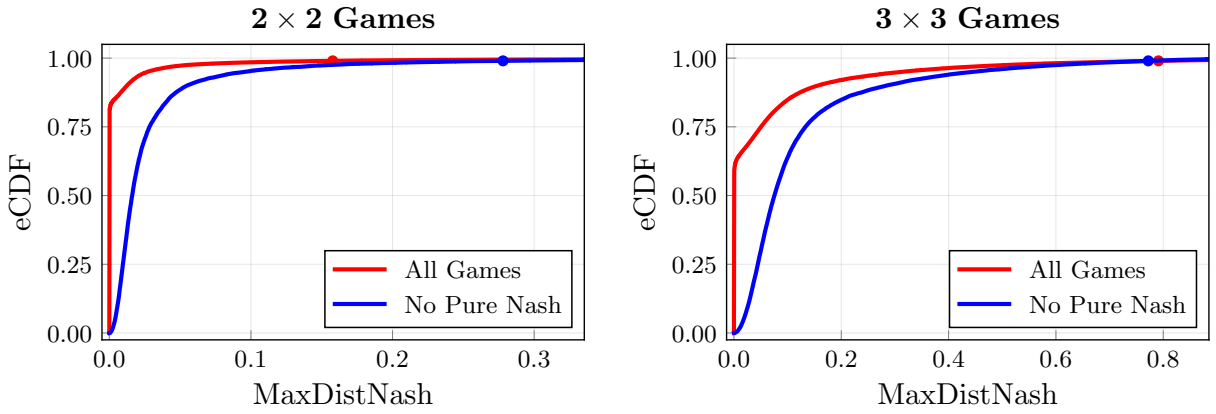

FIGURE 3. Empirical cumulative density of max distance from Nash across players in 2×2 and 3×3 games. Dots are placed in correspondence of the 99 percentile.

this is the case by looking at how δ-distant are players to equilibria. In our test set the average maximal distance across players from Nash is around 0.009. A better performance is achieved in games with only one pure Nash equilibrium (i.e., dominance solvable games) where the average max regret is 0.000 and the average δ-distance from Nash is 0.006. As a counterpoint, our metrics are slightly worse in games with a unique equilibrium which is in mixed strategies, where average regret is 0.019 and the maximal average distance is 0.032. As we mentioned in the introduction, learning to play mixed Nash is harder because of the well known instability of mixed equilibria. As long as the opponent is playing an epsilon away from equilibrium, responding with a mixed strategy is suboptimal.

The trained networks obtain comparable results in 3×3 games, although we observe an order of magnitude increase in average max regret and distance from Nash play. This is likely the case because the size of the neural network and the amount of training required to achieve worst case targets on mixed strategy Nash play grow exponentially in the number of available actions.[10] In 3×3 games players achieve an average max regret of 0.029, compared to a 0.68 benchmark from random play, and are playing an ϵ-Nash equilibrium with $\epsilon < 0.37$ (0.15) in 99% (95%) of games.

[10]Existing results on the computational complexity of Nash equilibrium (see Daskalakis et al. (2009) and Chen et al. (2007)) indicate that as the dimensionality of the game increases, the required network and training would have to increase non-polynomially.

In 3×3 games, the networks play with average maximal distance from Nash of 0.056. Maximal regret and distance from Nash reduce in dominance solvable games to 0.013 and 0.042, respectively. As was the case for 2×2 games, but to a greater extent, the networks have difficulties accurately learning to play mixed strategy equilibria. In the case of games with only one mixed equilibrium, the average max regret reaches 0.085. A significant increase is also observed in the average closeness of strategies to a Nash, which goes up to 0.122.

Based on our results above, we conjecture that two sufficiently flexible neural networks adversarially trained on a sequence of randomly generated finite normal-form games to minimise instantaneous regret converge to playing (an approximate) Nash equilibrium in every game. More precisely, we conjecture the following theoretical result. For any n and for any ϵ and γ arbitrarily small, there exist networks (f_n^1, f_n^2) with hyperparameters d, L, α and amount of training T both sufficiently large such that the adversarially trained netorks play an ϵ-equilibrium in all but a fraction γ of games in $\mathcal{G}_n$.

Because this conjecture concerns the dynamics of the learning process, it is empirically explored further in the next section. However, we are not able to provide a formal proof of the statement and little is known about convergence of adversarially trained networks except in zero sum situations, which have been studied in the context of generative adversarial networks known as GANs (see Daskalakis et al., 2017).

Equilibrium selection. Our simulations provide evidence that two neural network engaged in a competitive process learn to each play their part of some commonly identified Nash equilibrium. In practice, the two networks are jointly operating a selection from the Nash equilibrium correspondence. This raises the question of which equilibrium they are computing when a game admits more than one.

To provide an answer, we next look at games in our test set with more than one Nash equilibrium. To reduce noise, we focus on the subset of games where the networks played strategies at most 0.1-distant from the closest Nash equilibrium. For all games and for games with a unique Pareto optimal equilibrium, we computed the distribution of closest equilibrium play in our test set over the intersection of two categorical variables: whether or not the equilibrium is the risk dominant one and whether or not it is utilitarian. Finally, for both classes of games, we compute the same probabilities for games where the risk dominant equilibrium is different from the utilitarian one.[11] The results are in Table 4.

Remarkably, in more than 99% of 2×2 games with multiple equilibria the networks select the Harsanyi and Selten (1988) risk-dominant equilibrium. Non degenerate 2×2 games have either a unique equilibrium or two pure equilibria and a mixed one. In such games with multiple equilibria, the risk dominant equilibrium is the pure equilibrium that minimises the product of the losses from individual deviations from it (or the mixed equilibrium if the product of losses are identical).[12] Intuitively, it is an equilibrium which is robust to prediction errors about the opponent's strategy. One immediate implication of this finding is that mixed strategy are almost never played with multiple equilibria, since the set of games where the product of losses are equal has measure zero in the space of all games. Another is that the network will play the risk dominant equilibrium even if there exists a payoff dominant one (i.e., one that is Pareto optimal among the three equilibria). This runs contrary to what Harsanyi and Selten (1988) advocated as their leading

[11]The uniquely Pareto optimal equilibrium generically is, when it exists, the unique utilitarian equilibrium.

[12]In a game $\begin{pmatrix} A,a & B,b \\ C,c & D,d \end{pmatrix}$ with equilibria (A,a) and (D,d) the respective products of losses are $(A-C)(a-b)$ and $(D-B)(d-c)$

2 × 2 Games

	All games (16 474)		with PD equilibrium (8 305)		
	Utilitarian	**Not Utilitarian**	**Payoff-Dominant**	**Not Payoff-Dominant**	**Marginal**
Risk Dominant	0.667 {0.993}	0.325 [0.991]	0.710 {0.994}	0.284 [0.991]	0.992
Not Risk Dominant	0.003 [0.008]	0.005 {0.007}	0.002 [0.009]	0.004 {0.006}	0.008
Marginal	0.669	0.331	0.712	0.288	

3 × 3 Games

	All games (32 646)		with PD equilibrium (16 963)		
	Utilitarian	**Not Utilitarian**	**Payoff-Dominant**	**Not Payoff-Dominant**	**Marginal**
Risk Dominant	0.539 {0.879}	0.272 [0.705]	0.593 {0.900}	0.227 [0.663]	0.812
Not Risk Dominant	0.110 [0.285]	0.078 {0.121}	0.113 [0.331]	0.068 {0.100}	0.188
Marginal	0.649	0.351	0.706	0.294	

TABLE 4. Selection on 2×2 and 3×3 games. Statistics are limited to games in the test set with multiple equilibria where the networks are playing no more than 0.1-distant from the closest Nash. The tables on the left contain all games, while the tables on the right contain only games where payoff dominant equilibrium exists. Each subtable on the left (right) contains the sample frequencies of the network play for the four possible outcomes: utilitarian (payoff dominant) and risk dominant, risk dominant but not utilitarian (not payoff dominant), not risk dominant but utilitarian (payoff dominant), not risk dominant and not utilitarian (not payoff dominant). Numbers in square brackets report frequencies conditional on the two equilibria being different, while on curly brackets report frequencies when the two equilibria are identical.

solution concept, but is consistent with refinement based on stochastic stability from evolutionary game theory (e.g., Kandori et al. (1993) and Young (1993)).

Behaviour is more nuanced in 3×3 games. The risk dominant equilibrium computed according to the Harsanyi and Selten (1988) linear-tracing procedure is selected only 81% of times. Even when the risk dominant is not selected, the utilitarian equilibrium and the payoff dominant one are only selected about 60% of times. Hence, the networks appear to often play Pareto dominated equilibria. In particular, when the payoff dominant and the risk dominant equilibrium differ, the dominated risk-dominant one is chosen about 2/3 of times. The conclusion that mixed strategies are rarely played with multiple equilibria carries on to 3×3 games.

To enhance the behavioural characterization of the trained networks we tested adherence to a number of natural axioms, some of which were previously proposed in the literature on the axiomatisation of Nash equilibrium. We found that the selection operated by the networks satisfies: symmetry, independence from strategically irrelevant actions, equivariance, monotonicity, and invariance to payoff transformations that preserve the best reply structure. We now define and discuss these axioms in turn. In order to do so, we treat the output of our network as a family of single-valued solution concepts $f = \{f_n\}_{1<n\leq 3} = \{(f_n^1, f_n^2)\}_{1<n\leq 3}$.

We say that f is *symmetric* if it selects the same profile of strategies whenever the role of players is swapped. That is, $f_n^1(G^1, G^2) = f_n^2(G^2, G^1)$ for all $(G^1, G^2) \in \mathcal{G}_n$. To test this property we computed the δ-distance between $f_n^1(G^1, G^2)$ and $f_n^2(G^2, G^1)$ for all 2^{17} games in the test set, for $n = 2, 3$. We found an average distance (standard deviation) of 0.007 (0.053) in 2×2 games and 0.045 (0.129) in 3×3 games. The axiom implies that symmetric equilibria are played in symmetric games (i.e., games where $G_1 = G_2$). Since the ex-post experience of playing in the two roles is heterogeneous, symmetry of f further confirms that the learning is robust the realisation of games in the training set.

We say that f_n satisfies *equivariance* if and only if it selects the same profile of strategies whenever two games differ only because the order of actions has been reshuffled. More formally, f satisfies equivariance if $f_n^i(G^1, G^2) = f_n^i(\tilde{G}^1, \tilde{G}^2)$ for $i = 1, 2$ whenever $(G^1, G^2) \in \mathcal{G}_n$ and $(\tilde{G}^1, \tilde{G}^2) = (PG^1Q, (P(G^2)^\top Q)^\top)$ for some row and column permutation matrixes P and Q. Building on symmetry, we verified equivariance approximately holds by evaluating for each game in the test set the output of one of the two networks across the $(n!)^2$ permutations of the game. For each game, we measured the average distance from the centroid strategy across permutations. Then, averaging across games we obtained an overall average (standard deviation) of 0.004 (0.025) in 2×2 games and 0.034 (0.075) in 3×3 ones. This finding has a bite in our setting because the order of actions determines the placement of playoffs in the input layer. Therefore, the network could in principle coordinate to play a different equilibrium based on the observed order of actions.

We say that f_n satisfies *invariance to payoff transformations that preserve the best reply structure* if it selects the same profile of strategies following payoff transformations that do not alter the best reply correspondence. Formally, $f_n^i(G^1, G^2) = f_n^i(\tilde{G}^1, \tilde{G}^2)$ for $i = 1, 2$ if $(G^1, G^2) \in \mathcal{G}_n$ and $(\tilde{G}^1, \tilde{G}^2) = (a^1 G^1 + C_1, a^2 G^2 + C_2)$ for (a^1, a^2) positive scalars and (C^1, C^2) column-constant $n \times n$ matrices. For each game in the test set, we generated 64 random tranformation from distributions $a^i \sim \text{Uniform}(1, n)$ and $C^i_{1,j} = \cdots = C^i_{n,j} \sim \text{Uniform}(1, n)$, $i = 1, 2$, $j = 1, .., n$.[13] For each game, we computed the average across the transformed games of the δ-distance of the network prediction and the centroid strategy. Averaging over all games we obtained a mean value (standard deviation) of 0.013 (0.054) in 2×2 games and of 0.058 (0.102) in 3×3 ones. Notably, since f satisfies invariance to payoff transformations that preserve the best reply structure, it satisfies invariance to affine transformations of payoffs a fortiori.

A solution concept f satisfies *monotonicity* if, for all games where it selects a pure equilibrium, the pure equilibrium is still selected if we raise the payoff of players at the equilibrium point. Formally, for all (G^1, G^2) such that $f_n^1(G^1, G^2), f_n^2(G^1, G^2)$ identifes a pure equilibrium at action profile k, z, we must have $f_n^i(G^1, G^2) = f_n^i(\tilde{G}^1, \tilde{G}^2)$ if $\tilde{G}^1_{k,z} = \tilde{G}^1_{k,z} + h_1$, $\tilde{G}^2_{z,k} = \tilde{G}^2_{z,k} + h_2$ and $(G^1, G^2) = (\tilde{G}^1, \tilde{G}^2)$ otherwise, with $h_1, h_2 \geq 0$. We tested the property by taking all games in the test set where the networks are playing 0.05-distant from a Nash equilibrium, generating a couple of random increments uniformly from $[0, 1]$, adding them to the payoffs at the equilibrium, and evaluating the difference in behaviour in each such generated game from the centroid computed across all transformed games. Averaging over all games we obtained a mean value (standard deviation) of 0.000 (0.000) in 2×2 games and of 0.000 (0.002) in 3×3 ones. In light of the axiomatization by Harsanyi and Selten (1988) (see Theorem 3.9.1), monotonicity and the other three properties above are implied, for 2×2 games, by the networks selecting the risk-dominant equilibrium in every game.

[13] We restrict $a^i \geq 1$ to avoid generating games with small payoffs where a player is nearly indifferent among its strategies.

We say that a solution concept satisfies *independence from strategically irrelevant actions* if it selects the same equilibrium in any two games where one is obtained by adding strictly dominated actions to the other. To formalise this axiom, let $[(G_1, G_2)]_k$ indicate a game restricted to the first $k \leq n$ actions for both players and $[f_n(\tilde{G}^1, \tilde{G}^2)]_k$ indicate the analogously resticted output of the two networks. Independence from strategically irrelevant actions is satisfied if $f_n(G_1, G_2) = [f_{n+1}(\tilde{G}^1, \tilde{G}^2)]_n$ whenever $(\tilde{G}^1, \tilde{G}^2) = [(G_1, G_2)]_n$ and the $n+1$ actions in $(\tilde{G}^1, \tilde{G}^2)$ for both players are strictly dominated. We tested this axiom by extracting from the test set all 3×3 games where there was a single strictly dominated action for each player. Relying on symmetry, we then compared the output of one network in each 3×3 game restricted to the undominated strategies with the output of the same network in the 2×2 games obtained by eliminating the identified dominated strategies.[14] We found an average δ-distance (standard deviation) between f_2^1 and f_3^1 equal to 0.034 (0.124) in our sample. This property, which is tested using both the models trained for 2×2 and 3×3 games, is of independent interest. It establishes coherence between models trained on games of different sizes. The equilibrium selected by the learning process is not affected by the presence of strategically irrelevant actions and would continue to be selected by larger models in games that appear equivalent after dominated strategies are iteratively eliminated.

A summary of the results for the axioms we tested is presented in the table below.

	2×2 Games			**3×3 Games**		
	Obs.	**Avg. Dist. (std)**	**Quantiles 0.90 0.99**	**Obs.**	**Avg. Dist. (std)**	**Quantiles 0.90 0.99**
Symmetry	131072	0.007 (0.053)	0.062 0.097	131072	0.045 (0.129)	0.510 0.759
Equivariance	131072 × 4	0.004 (0.025)	0.038 0.060	131072 × 36	0.034 (0.075)	0.291 0.371
Best reply inv.	131072 × 64	0.013 (0.054)	0.200 0.330	131072 × 64	0.058 (0.102)	0.379 0.437
Monotonicity	112949 × 64	0.000 (0.000)	0.000 0.000	88557 × 64	0.000 (0.002)	0.000 0.000
Indep. irr. actions	36483	0.034 (0.124)	0.067 0.839		— —	— —

TABLE 5. Statistics for behavioural axioms. Independence of irrelevant actions is tested using both models so 3×3 statics are not available.

Iterated dominance. Playing Nash requires correctly predicting the opponent's behaviour, which is beyond what is implied by the assumptions of rationality and mutual, or even common, belief in rationality in the sense of Aumann and Brandenburger (1995). Nonetheless, we found it worthwhile to confirm that the trained networks conform to an even greater extent to the implications of rationality and common belief in rationality alone. To do so, we evaluate how much mass the networks place on strictly dominated actions and on actions that are not rationalizable (i.e., that do not survive iterated elimination of strictly dominated strategies). Results are in Table 6 and Figure 4.

[14] We preferred to use 3×3 games with dominated strategies as opposed to augmenting 2×2 games in order to maintain conditionally uniform sampling of such games in $\mathcal{G}_\ni$. Note that due to the possibility that a small mass is placed on the dominated actions, the resulting restricted strategy is not a distribution. Nonetheless, this has no material effect on computed total variation.

In 2×2 games, each network learns almost perfectly to avoid strictly dominated actions. Looking at dominance solvable games only, where at least one strictly dominated strategy exists for a player, the average mass placed on strictly dominated actions is 0.002. Remarkably, the average mass on strictly dominated strategies goes down to a number smaller than 0.0001 if we restrict attention to games where the payoff loss from playing the dominated action is above 0.1. We interpret this finding as the network allocating limited computation capacity to both predicting the action of the opponent and best responding to it, being less concerned about minor errors in the latter when the payoff consequences are smaller. The average mass placed on undominated strategies increases to 0.005 in 3×3 dominance solvable games but remains an order of magnitude lower than the average distance from Nash play.

Having established the above, we ask whether during training the networks also learn that their opponent is rational and adjust their behaviour accordingly. Looking at 2×2 games which are dominant solvable we observe that the mean mass placed on strategies that are not rationalizable is equal to 0.003, which is comparable to the performance obtained in avoiding strictly dominated strategies. Since a 2×2 game is dominance solvable when at least one of the player has dominated strategy, the figure shows that players almost perfectly predict the behaviour of an opponent with a dominant strategy and best respond to it. As for the case of dominated strategies, play of non rationalizable strategies in 2×2 games disappears once we look at games where making such mistakes has a non negligible cost. Finally, looking at 3×3 dominance solvable games we observe that the average mass placed on non rationalisable strategies is 0.019, suggesting players develop a higher-order belief in rationality and act accordingly, but such higher-order reasoning is not lossless.

2×2 Games

	All Games	**Partially Solvable**	**Dominance Solvable**
Fraction of Games	1.000	0.750	0.750
Mean Mass on undominated strategies	0.001 (0.027)	0.002 (0.031)	0.002 (0.031)
Mean Mass on non rationalizable strategies	0.002 (0.032)	0.003 (0.037)	0.003 (0.037)

3×3 Games

	All Games	**Partially Solvable**	**Dominance Solvable**
Fraction of Games	1.000	0.869	0.495
Mean Mass on undominated strategies	0.004 (0.042)	0.004 (0.045)	0.005 (0.050)
Mean Mass on non rationalizable strategies	0.011 (0.079)	0.013 (0.084)	0.019 (0.103)

TABLE 6. Average mass placed on dominated and non rationalizable strategies by class of games. Standard deviation in parenthesis. In partially solvable games a dominant strategy exists for at least one player.

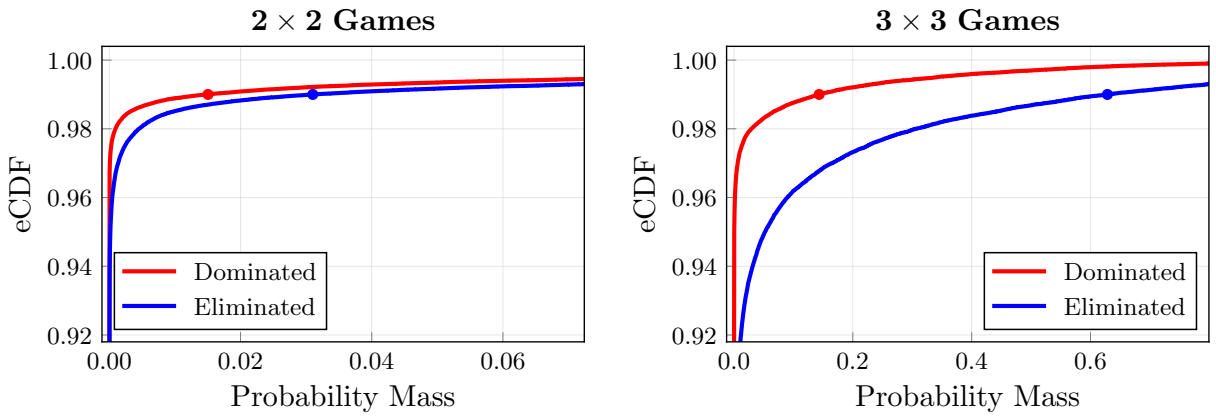

FIGURE 4. Empirical cumulative density of the mass placed on strictly dominated and non razionalizable strategies in 2×2 and 3×3 games dominance solvable games.

5. Learning dynamics

In this section, we briefly discuss the learning dynamics for our baseline scenario. First, we display the learning curves for 2×2 and 3×3 games. Specifically, we plot the maximum regret achieved on average over a batch of games by the two networks on the y-axis, and the optimization periods, from 1 to T, on the x-axis in a base-10 logarithmic scale.

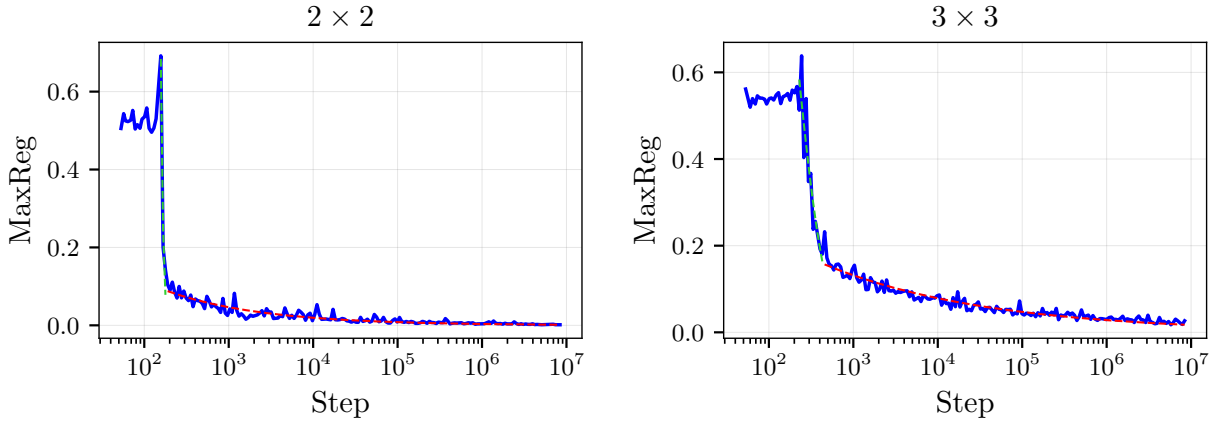

FIGURE 5. Learning curves for 2×2 and 3×3 models. The y-axis shows the maximum regret across periods, averaged over a batch of games with moving averages over 100 optimization periods. The x-axis uses a base-10 logarithmic scale.

Inspection of the learning curves suggests that learning proceeds monotonically, supporting our convergence conjecture. It takes place in three distinct phases. First, there is an initial and very short period of little or no learning, lasting around 140 (220) periods. Second, the networks experience a short phase of exponential learning, lasting around 30 (230) optimization periods. In this phase, most of the learning takes place, bringing the average regret down from the random benchmark to roughly 0.11 (0.13). The best-fit exponential curve, depicted in green in both graphs, has an exponential decay rate of roughly 0.1 for the 2×2 models and of 0.006 for 3×3. Finally, learning in the last and longest phase follows a power law. The best-fit power curve, depicted in red in the graphs, has a power-law exponent of roughly -0.38 for 2×2 models and of -0.22 for 3×3. Extrapolation from the power law indicates that 380 trillion further periods of training may

be required for the 3×3 models to bring down the average maximum regret the performance of 0.0016 achieved in 2×2 games.

We emphasize that, as is the case typically in the training of adversarial neural networks, such a well-behaved learning curve was not warranted a priori. In fact, the first-order approach we adopt is guaranteed to converge only under the assumption of a convex loss function and a stationary distribution of the opponent's behaviour, or for GANs playing fixed zero-sum games. However, none of these conditions hold in our case.

As we argued elsewhere, learning to play a mixed Nash equilibrium is generally more challenging than learning to play a pure strategy Nash equilibrium. This observation is reinforced by an analysis of learning restricted to two categories of games: games with only one mixed equilibrium, and games with either one pure Nash equilibrium or multiple equilibria, where we know the networks select a pure equilibrium. In particular, in the Figure 6 below, we report the outcome of evaluating the networks' performance at different stages of the learning process. For a sample of 128 optimization periods distributed logarithmically from 100 to T, we saved the models and evaluated the average maximum regret across players on the two classes of games within our test sets.

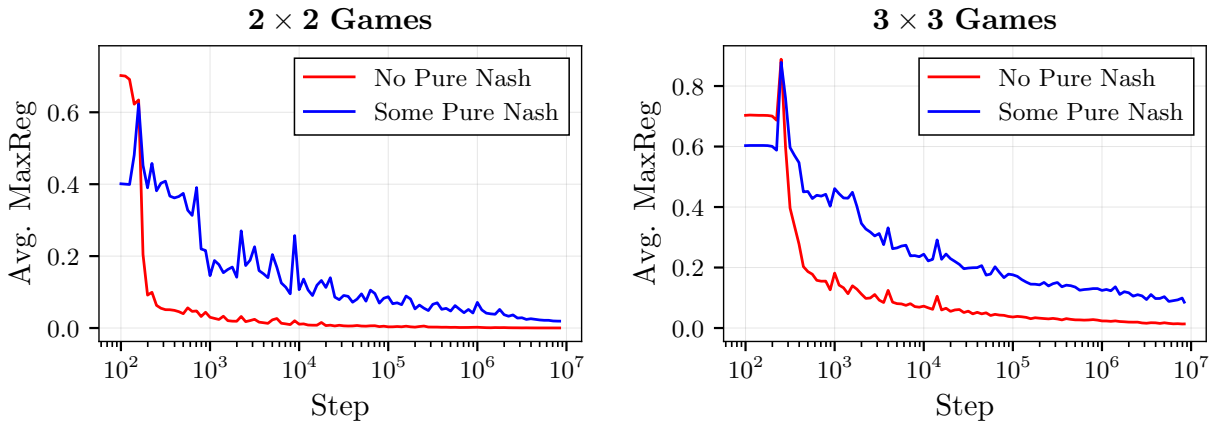

FIGURE 6. Evaluation of average maximum regret across players in 2×2 and 3×3 games at logarithmically sampled periods during learning, for games with only mixed equilibria and games with at least one pure equilibrium. The x-axis uses a base-10 logarithmic scale.

As the two pictures in Figure 6 illustrate, the learning of the mixed Nash equilibrium does not experience the typical burst of rapid improvement seen when games with pure equilibria are considered. Instead, learning begins and proceeds at a consistent power law rate. In contrast, when at least one pure equilibrium is present, performance accelerates quickly and reaches high levels, especially in 2×2 games. After this initial acceleration, as previously mentioned, learning continues with a decrease that follows a power law.

6. Robustness

We now discuss several variations on the baseline model. The aim is to highlight features of the model that are essential to obtain our results and those that are not. Conclusions are subject to change as the models in this section have been run with only minimal fine-tuning.

Training to play larger games. Our baseline analysis was conducted on 2×2 and 3×3 games. These games are the simplest and best-understood class of normal-form games and the computational demand to train game-playing networks grows nonlinearly in the number of actions. But to confirm that learning takes place in larger models, we report our results from training models for

4×4 and 5×5 games. The neural network used coincides with the neural network we used for 3×3 games. Training took place on 4 294 967 296 games (twice as much) and the batch size used was 1024 (four times as much).

We evaluated the models using a test set of 2^{17} randomly generated 4×4 and 5×5 games. The average maximal regret across players and the average maximal distance from the closest Nash equilibrium are reported in Table 7.

4×4 Games			
	All Games	**# Pure Nash**	
		0	**1**
Relative Share (of 2^{17})	1.00	0.26	0.51
Mean MaxReg	0.07	0.14	0.05
	(0.15)	(0.15)	(0.13)
Mean MaxDistNash	0.12	0.20	0.09
	(0.18)	(0.17)	(0.19)

5×5 Games			
	All Games	**# Pure Nash**	
		0	**1**
Relative Share (of 2^{17})	1.00	0.28	0.47
Mean MaxReg	0.11	0.18	0.09
	(0.19)	(0.18)	(0.18)
Mean MaxDistNash	0.16	0.23	0.14
	(0.21)	(0.18)	(0.21)

TABLE 7. 4×4 and 5×5 games:Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

While these models are less performing than our baseline ones, these results indicate that learning is taking place. To investigate whether the output of models trained on larger games agrees with the output of of our baseline models trained on smaller games, we perform a test for invariance to strategic irrelevant actions using the entire family of models (see Section 4 for details). Table 8 contains the results of the analysis.

Test of Independence			
	3×3	4×4	5×5
2×2	0.032	0.039	0.038
	(0.12)	(0.134)	(0.119)
3×3		0.092	0.088
		(0.186)	(0.192)
4×4			0.140
			(0.217)

TABLE 8. Test of independence from irrelevant action performed on pairs of models of different dimensions. See Section 4 for details.

The test scores provide reassurance that larger models play games with n actions when k actions for each player are strictly dominated as the smaller models play the same reduced games with $n - k$ actions.

Changing the training environment. In this subsection, we evaluate a number of changes to the training environment. First, we train our networks on a proper subset of the space of preferences. Second, we consider two changes to the loss functions. Third, we study the effect of batching. Finally, we perform training without imposing a decay rate on the learning rate.

To evaluate if models can generalise out of distribution their learning behaviour, we trained our baseline networks on a quarter of $\mathcal{G}_2$. We considered two different scenarios according to which 1/4 of the space was extracted.[15] Evaluations of the models performed on the complement of the training set relative to $\mathcal{G}_2$ are shown in Table 9.

Subspace (a) in 2×2 Games			
	All Games	**# Pure Nash**	
		0	**1**
Relative Share (of $3{\cdot}2^{15}$)	1.00	0.12	0.75
Mean MaxReg	0.01	0.03	0.00
	(0.04)	(0.03)	(0.01)
Mean MaxDistNash	0.02	0.06	0.02
	(0.10)	(0.10)	(0.11)

Subspace (b) in 2×2 Games			
	All Games	**# Pure Nash**	
		0	**1**
Relative Share (of $3{\cdot}2^{15}$)	1.00	0.12	0.75
Mean MaxReg	0.09	0.15	0.08
	(0.23)	(0.18)	(0.23)
Mean MaxDistNash	0.14	0.25	0.14
	(0.31)	(0.30)	(0.33)

TABLE 9. Training on subspaces: Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

We find the networks perform remarkably well in playing all games even when training on a subset of the space. However, while in the first scenario the performance is essentially equivalent to the baseline, we observe a loss of out sample performance in the second. This suggests that learning can take place even within a small subset of games, but that the choice of subset of games on which training takes place is important. Further work would be required to investigate the effect of sampling on training for performance over all games.

We trained our baseline models using a regret-based loss function with two main additional assumptions. First, that the loss is equal to the square of the regret. Second, that regret is computed by assuming observation of the entire mixed strategy of the opponent. Below, we dispense with each of these two assumptions in turn, limiting our attention to 2×2 games. The only difference to the baseline model is the initial learning rate which is set to $\eta_0 = 0.01$.

Table 10 reports the summary results for the trained models evaluated on our test sets.

The analysis indicate that learning takes place also when relaxing these two assumption and the two alternative models both deliver play close to that of the baseline. However, both models struggle to identify and play mixed equilibria. Because of the very limited fine-tuning we performed, we believe further analysis is required to determine whether both assumptions are indeed essential or not to learn mixed equilibria.

Next, we look at the effect of batching on learning, focusing on 2×2 games. We considered two scenarios: online training with no batching and doubling the batch size. Except for the number of games played, which is adjusted to keep the same number of optimization steps, and for a lower learning rate of 0.01 chosen for the case of online learning, all other details of the models remain

[15] Let $\mathbf{v} = (-1, -1, \ldots, 1, 1)^\top \in \mathbb{R}^{n^2}$ be the vector with the first n entries equal to -1 and the remaining n entries equal to 1, and let $\mathbf{u} = (-1, 1, -1, \ldots, 1)^\top \in \mathbb{R}^{n^2}$ be the vector with alternating entries -1 and 1. Recall that $\mathcal{G}_n$ denotes the space of games where the players' payoff vectors are in $S_n = \{x \in \mathbb{R}^{n^2} : ||x||_2 = n \text{ and } \mathbf{1}^\top x = 0\}$.. We consider training on games uniformly sampled from two subspaces of $\mathcal{G}_n$: one where the payoff vectors lie in $S_n \cap \{x \in \mathbb{R}^{n^2} : \mathbf{v}^\top x > 0\}$, and another where they lie in $S_n \cap \{x \in \mathbb{R}^{n^2} : \text{sgn}(\mathbf{v}^\top x) = \text{sgn}(\mathbf{u}^\top x)\}$. We evaluate networks on the complement (relative to $\mathcal{G}_n$) of the subspace where they are trained.

Linear Loss in 2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.125	0.75
Mean MaxReg	0.12	0.97	0.00
	(0.41)	(0.71)	(0.04)
Mean MaxDistNash	0.06	0.42	0.01
	(0.17)	(0.19)	(0.07)

Ex-post regret in 2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.125	0.75
Mean MaxReg	0.02	0.15	0.00
	(0.06)	(0.07)	(0.01)
Mean MaxDistNash	0.04	0.29	0.01
	(0.12)	(0.15)	(0.06)

TABLE 10. Linear loss and ex-post regret: Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

identical to the baseline ones. Results in Table 11 show that increasing or decreasing the batch size has no major effect on learning.

Online training in 2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.125	0.75
Mean MaxReg	0.10	0.06	0.00
	(0.03)	(0.05)	(0.01)
Mean MaxDistNash	0.03	0.07	0.02
	(0.10)	(0.08)	(0.10)

Double batch size in 2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.125	0.75
Mean MaxReg	0.00	0.02	0.00
	(0.01)	(0.01)	(0.00)
Mean MaxDistNash	0.01	0.03	0.01
	(0.05)	(0.05)	(0.05)

TABLE 11. Online and double batching: Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

Finally, we eliminated decay in the training learning rate. That is, $\alpha = 1$. All other details remained the same as in the baseline scenarios. Table 12 presents the results. Also in this case, the broad conclusion is that results are robust to changes, even drastic ones, to the learning rate.

No learning decay in 2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.125	0.75
Mean MaxReg	0.01	0.03	0.00
	(0.02)	(0.03)	(0.00)
Mean MaxDistNash	0.02	0.05	0.01
	(0.08)	(0.07)	(0.08)

TABLE 12. No learning decay: Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

Modifying the network architecture. To verify that our results are robust to changes to the networks' architecture, we trained two sets of networks for 2×2 games with a different architecture. In one set, we doubled the number of neurons per layer, while in the other we halved them. All other details of the models remained the same. The results in Table 13 confirm that even relatively large changes to the network architecture do not quailitatively alter the results we obtained in the baseline models.

Halved neurons 2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.125	0.75
Mean MaxReg	0.00 (0.01)	0.02 (0.02)	0.00 (0.00)
Mean MaxDistNash	0.01 (0.05)	0.03 (0.05)	0.01 (0.05)

Doubled neurons 2×2 Games

	All Games	# Pure Nash	
		0	1
Relative Share (of 2^{17})	1.00	0.125	0.75
Mean MaxReg	0.00 (0.01)	0.02 (0.01)	0.00 (0.00)
Mean MaxDistNash	0.01 (0.05)	0.03 (0.05)	0.01 (0.05)

TABLE 13. Architectural changes: Mean and standard deviation (in parenthesis) across the test set and by number of pure Nash equilibria of: the maximum regret among players; and the maximum total variation distance between the strategy played and the closest Nash among players. (Numbers are rounded.)

7. CONCLUSIONS

We train two independent neural networks to minimise their instantaneous regret by having them play adversarially in a sequence of random 2×2 and 3×3 games. We show that learning is effective in reducing regret. The average regret in the test set decreases from around 0.6 to less than 0.01 in 2×2 games and 0.03 in 3×3 ones. These results imply that the networks learn to play their part of an approximate (ϵ-)Nash equilibrium in every game. We interpret these findings as supporting the learning and evolutionary foundations of Nash equilibrium. Learning is possible even when players never play the exact same game twice. Additionally, we show that in all 2×2 games and in about 80% of 3×3 games, the two networks consistently select the risk-dominant equilibrium. This finding reinforces the game-by-game predictions made by the evolutionary and learning literature in 2×2 games.

We obtain these results after optimizing our models to balance complexity and performance. Then, we demonstrate that these results are robust to various modifications. Learning is unaffected by the sampling method and Nash play generalises well to games entirely outside the support of the sampling distribution. Nor is learning affected by the symmetry of the networks or details of their setup, such as the exact learning rate, its decay rate, or the network architecture. We also confirm that our results are robust to a key assumption in our baseline models: the full observation of the opponent's mixed strategy. While we observe a significant drop in performance when playing mixed equilibria, average regret still reaches 0.013 in 2×2 games.

We also train our models on 4×4 and 5×5 games and show that learning occurs in these larger games. Moreover, the larger models exhibit the same equilibrium selection behaviour in smaller games as our smaller baseline models. Therefore, we expect our main results to be broadly robust to the dimensionality of the game. However, new insights into equilibrium selection may emerge from full training of larger models. For instance, larger games are necessary to test concepts applicable to the normal form representations of extensive games, such as subgame perfection and Kohlberg and

Mertens (1986)'s strategic stability. Unfortunately, the complexity of computing Nash equilibria makes such training of larger models computationally expensive. Hence, the analysis of equilibrium selection in larger games is left for future work.

We present a learning model that we believe captures salient features of strategic behaviour in humans and animals. Similar to other cases where neural networks have been used to replicate intelligent human behaviour, our theory appears to better fit the learning of intuitive strategic decision-making rather than deliberate strategic thinking. Therefore, we would find interesting to calibrate our models with smaller training and assess their ability to explain experimental results. Crucially, we believe that such experiments should expose agents to naturally encountered strategic situations, rather than matrices of payoffs, whose processing is more likely to involve deliberate strategic reasoning. In this vein, several field experiments have already demonstrated the ability of experienced individuals to employ minmax strategies in sporting situations modelled as zero-sum games (see Chiappori et al., 2002; Palacios-Huerta, 2003; Walker and Wooders, 2001). However, since professional players likely come closest to satisfying the classic learning theory condition of replaying the same game multiple times, it would be interesting, consistent with our main observation, to test whether the strategic intuitions developed in one sport extend to others.

Another natural testing ground for a theory of intuitive strategic behaviour would be driving. In this context, one could argue that the payoffs for everyone are relatively clear, and many distinct strategic games are played, including non-zero-sum, not just slight variations of the same game. However, implemented in self-driving cars, for example, our networks would learn to best respond to current traffic behaviour. This contrasts with the exercise in this paper, where the two networks learn together to coordinate with each other. Nonetheless, we believe it is not far-fetched to think that our model could provide insights into how existing traffic rules may have evolved over time, starting with horse-riding and progressing to modern road systems. Similarly, our model could help explain how strategic intuitions initially develop in children during play.

References

Aumann, R. and Brandenburger, A. (1995). Epistemic conditions for nash equilibrium. *Econometrica*, 63(5):1161.

Chen, X., Deng, X., and Teng, S.-H. (2007). Settling the complexity of computing two-player nash equilibria.

Chiappori, P.-A., Levitt, S., and Groseclose, T. (2002). Testing mixed-strategy equilibria when players are heterogeneous: The case of penalty kicks in soccer. *American Economic Review*, 92(4):1138–1151.

Cooper, D. J. and Kagel, J. H. (2003). Lessons learned: Generalizing learning across games. *The American Economic Review*, 93(2):202–207.

Daskalakis, C., Goldberg, P. W., and Papadimitriou, C. H. (2009). The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. (2017). Training gans with optimism.

Devetag, G. (2005). Precedent transfer in coordination games: An experiment. *Economics Letters*, 89(2):227–232.

Duan, Z., Huang, W., Zhang, D., Du, Y., Wang, J., Yang, Y., and Deng, X. (2023). Is nash equilibrium approximator learnable?

Fudenberg, D. and Kreps, D. M. (1993). Learning mixed equilibria. *Games and Economic Behavior*, 5(3):320–367.

Fudenberg, D. and Levine, D. K. (1998). *The Theory of Learning in Games*, volume 1 of *MIT Press Books*. The MIT Press.

Fudenberg, D. and Liang, A. (2019). Predicting and understanding initial play. *American Economic Review*, 109(12):4112–4141.

Germano, F. (2007). Stochastic evolution of rules for playing finite normal form games. *Theory and Decision*, 62(4):311–333.

Goeree, J. K. and Holt, C. A. (2001). Ten little treasures of game theory and ten intuitive contradictions. *American Economic Review*, 91(5):1402–1422.

Grimm, V. and Mengel, F. (2009). Cooperation in viscous populations—experimental evidence. *Games and Economic Behavior*, 66(1):202–220.

Harsanyi, J. C. and Selten, R. (1988). *A General Theory of Equilibrium Selection in Games*, volume 1 of *MIT Press Books*. The MIT Press.

Hart, S. and Mas-Colell, A. (2003). Uncoupled dynamics do not lead to nash equilibrium. *American Economic Review*, 93(5):1830–1836.

Hartford, J. S., Wright, J. R., and Leyton-Brown, K. (2016). Deep learning for predicting human strategic behavior. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Jehiel, P. (2005). Analogy-based expectation equilibrium. *Journal of Economic Theory*, 123(2):81–104.

Kandori, M., Mailath, G. J., and Rob, R. (1993). Learning, mutation, and long run equilibria in games. *Econometrica*, 61(1):29–56.

Kohlberg, E. and Mertens, J.-F. (1986). On the strategic stability of equilibria. *Econometrica*, 54(5):1003.

Kreps, D. M. (1990). *A Course in Microeconomic Theory*. Princeton University Press.

Lensberg, T. and Schenk-Hoppé, K. (2021). Cold play: Learning across bimatrix games. *Journal of Economic Behavior & Organization*, 185(C):419–441.

LiCalzi, M. (1995). Fictitious play by cases. *Games and Economic Behavior*, 11(1):64–89.

Liu, S., Marris, L., Piliouras, G., Gemp, I., and Heess, N. (2024). Nfgtransformer: Equivariant representation learning for normal-form games. In *The Twelfth International Conference on Learning Representations*.

Marchiori, D., Di Guida, S., and Polonio, L. (2021). Plasticity of strategic sophistication in interactive decision-making. *Journal of Economic Theory*, 196:105291.

Marchiori, D. and Warglien, M. (2008). Predicting human interactive learning by regret-driven neural networks. *Science*, 319(5866):1111–1113.

Mengel, F. (2012). Learning across games. *Games and Economic Behavior*, 74(2):601–619.

Palacios-Huerta, I. (2003). Professionals play minimax. *Review of Economic Studies*, 70(2):395–415.

Samuelson, L. (2001). Analogies, adaptation, and anomalies. *Journal of Economic Theory*, 97(2):320–366.

Selten, R., Abbink, K., Buchta, J., and Sadrieh, A. (2003). How to play (3×3)-games.: A strategy method experiment. *Games and Economic Behavior*, 45(1):19–37. First World Congress of the Game Theory Society.

Sgroi, D. and Zizzo, D. J. (2009). Learning to play games: Neural networks as bounded-rational players. *Journal of Economic Behavior & Organization*, 69(1):27–38.

Spiliopoulos, L. (2011). Neural networks as a unifying learning model for random normal form games. *Adaptive Behavior*, 19(6):383–408.

Spiliopoulos, L. (2012). Interactive learning in 2×2 normal form games by neural network agents. *Physica A: Statistical Mechanics and its Applications*, 391(22):5557–5562.

Stahl, D. O. (1999). Evidence based rules and learning in symmetric normal-form games. *International Journal of Game Theory*, 28(1):111–130.

Steiner, J. and Stewart, C. (2008). Contagion through learning. *Theoretical Economics*, 3(4):431–458.

Walker, M. and Wooders, J. (2001). Minimax play at wimbledon. *American Economic Review*, 91(5):1521–1538.

Young, H. P. (1993). The evolution of conventions. *Econometrica*, 61(1):57–84.