

Comparing Neural Policy Learning and Neurosymbolic Search for Optimal Move Selection in Connect-4

Mete Baklaci
mtbaklaci@gmail.com

ABSTRACT

Neural networks can learn complex patterns from data and make accurate judgments that explicit rules may not be capable of making. However, in heavily rule-based contexts such as board games, symbolic methods based on search can provide more systematic reasoning over future states compared to neural networks. Investigating the complementary capabilities of both paradigms, the present study comparatively evaluates both purely neural and neural network-guided search approaches in Connect-4 game-playing. This is done by training policy and value convolutional neural networks, where the former is used to directly generate moves whereas the latter is integrated in the minimax search algorithm as the evaluation function, forming a neurosymbolic agent. Both agents have been evaluated against symbolic game-playing benchmarks. In head-to-head evaluation, the depth-5 neurosymbolic agent defeated the policy neural network in 82.5% of games. Against the symbolic benchmark, the neurosymbolic agent achieved an 88.5% win rate compared with 69.5% for the policy neural network. The results demonstrate that neural networks can learn significant strategic information in Connect-4, while combining value learning with symbolic search can substantially improve playing performance.

INTRODUCTION

Neural networks [1] possess a strong ability to learn complex patterns from data and make predictions in situations where manually defining relevant decision rules is difficult [2]. Rather than requiring all relationships between inputs and outputs to be specified in advance, neural networks can infer these relationships from examples, allowing them to exploit nuanced patterns. However, this flexibility bears with it a downside: neural networks may not be able to reproduce the systematic reasoning required by tasks within heavily rule-based systems [3]. In such environments, correct decisions must encapsulate recognizing patterns in the current state, and also reasoning over the consequences of possible actions and the constraints imposed by the environment [4].

This situation poses an important problem for the application of neural methods to rule-governed decision-making tasks. In this context, a purely neural system is a decision framework in which decision-making is performed entirely by an artificial neural network that learns patterns and relationships from data, rather than relying on an explicit representation of the rules governing the task [5]. Such systems can learn complex relationships between inputs and outputs from examples without requiring

these relationships to be explicitly specified. Consequently, a purely neural system may learn patterns associated with successful decisions without explicitly representing the rules or mechanisms that produce those decisions. This may allow it to acquire significant strategically useful information while providing no explicit mechanism for systematically reasoning over the consequences of alternative actions [6].

Symbolic methods [7], in contrast, rely on the explicit representation of knowledge, including rules governing the environment, and the algorithmic manipulation of these representations to achieve a goal [5]. They can, therefore, use methods such as search to examine a possible sequence of actions, but their effectiveness can depend on how well they can judge the quality of the emergent states. In addition to this, symbolic methods bear the additional benefit of interpretability due to explicit reasoning chains, while it is difficult to interpret the actions of neural networks, notorious for being “black boxes” [8].

Neurosymbolic approaches seek to combine these complementary properties by synthesizing neural approaches with symbolic reasoning structures, thereby complementing the pattern-recognition and learning capabilities of neural networks with explicit reasoning capabilities of symbolic systems [9] [10] [11]. Investigating this combination is therefore relevant to determining the extent to which neural networks can learn successful behaviour in rule-based environments on their own and also whether explicit reasoning mechanisms can make more effective use of this information learned by neural models.

A controlled exploration of this question demands an environment in which the underlying rules are well-defined and the quality of decisions can be precisely determined. Board games hence provide an ideal environment since their state transitions and action spaces are described by well defined rules and the game outcomes provide an objective measure of decision quality [12]. They, therefore, provide a useful setting for examining the extent to which neural networks can acquire strategically relevant information from examples. Among such games, Connect-4 is particularly suitable for this purpose [13]. Figure 1 illustrates a Connect-4 position and a sample move played in the given board configuration. Connect-4 is a deterministic, two-player, perfect-information game with a relatively small action space. Furthermore, Connect-4 has been mathematically solved, meaning that the perfect-play outcomes of all possible positions are computationally determined [14]. This makes it possible to obtain objective ground-truth information about both the quality of individual moves and the eventual outcome of positions.

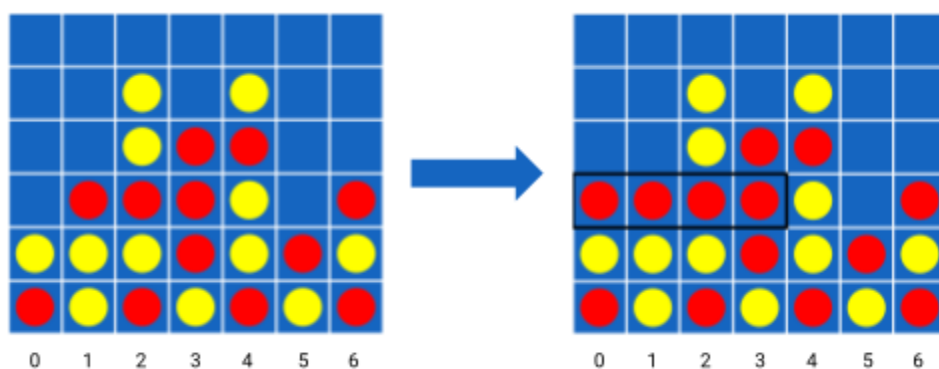


Figure 1: Sample Connect-4 board. Player 1 is represented with the red pieces and Player 2 is represented with the yellow pieces. A move played by Player 1 in Column 0 is depicted. The move has resulted in 4 of Player 1's pieces aligning horizontally, leading to a win for Player 1. For more details about the Connect-4 game rules, see the Dataset subsection under Materials and Methods section.

To this end, the present study utilizes Connect-4 to investigate the capabilities of purely neural and neurosymbolic approaches to game playing. Specifically, the study examines the extent to which convolutional neural networks can learn meaningful information directly from ground-truth, and whether a neural representation incorporated into explicit search can produce stronger game-playing performance than direct policy prediction. A policy CNN is trained to predict the optimal move directly from a board position, representing a purely neural approach in which decision-making is performed directly without explicit search. On the other hand, a second CNN with the same architecture is trained to estimate the perfect-play game outcome of a position instead and this value network is then incorporated into Minimax search, allowing learned evaluation to be combined with explicit reasoning over future game states. An overall summary of the results is provided in Main Findings.

Our primary contributions in this study are threefold:

1. We contribute an exact solver analyzed Connect-4 position dataset encompassing 446,302 Connect-4 positions from 21,000 simulated games.
2. We perform a set of ablation studies that investigate the effects of network architecture size, various input encoding formats, dataset size and symmetry-based data augmentation on the performance of policy and value networks.
3. We perform comparative analyses of the effectiveness of a purely neural policy network and a neurosymbolic agent that combines a neural value network with Alpha-Beta-pruned Minimax search in Connect-4 game playing by evaluation against benchmarks and face-to-face comparison.

RELATED WORKS

Symbolic Search-Based Approaches

Previous research on Connect-4 has extensively investigated search algorithms to maximize playing strength while minimizing computational cost. These methods use the explicit rules of the game to explore possible future states, and determine advantageous moves. Analysing these methods provides a baseline for the playing strength that symbolic methods can achieve.

Several studies have investigated optimizations to Minimax search. For instance, Nasa et al. [15] demonstrated that Alpha-Beta pruning substantially reduces the number of nodes evaluated compared with standard Minimax, enabling deeper searches at lower computational cost. Similarly, Tommy et al. [16] found that the MTD(f) algorithm achieved comparable playing strength to Alpha-Beta pruning while reducing move execution time and the number of leaf nodes explored. Together, these studies highlight the importance of search efficiency in improving the practical performance of Connect-4 agents.

Other studies have compared different search strategies. Sheoran et al. [17] investigated optimized Minimax with Alpha-Beta pruning and Monte Carlo Tree Search (MCTS), examining the effectiveness of these approaches under computational constraints. Qiu et al. [18] compared Greedy, Alpha-Beta pruning, Principal Variation Search (PVS), and MCTS, with MCTS providing the strongest game-playing performance compared to the other approaches. These studies illustrate that playing strength is affected considerably by the search algorithm being used.

Despite these improvements, search-based approaches must still evaluate non-terminal states when the search reaches its depth limit. The quality of these evaluations can therefore influence the moves selected by the search algorithm, and consequently its playing strength. The present study investigates whether this evaluation can be learned from perfect-play data using a neural value network, rather than relying solely on manually designed heuristics.

Learning-Based Approaches

Other studies in the literature have examined both neural and non-neural learning approaches, with reinforcement learning methods being more common than supervised learning methods.

Learning-based approaches have investigated whether game-playing strategies can be acquired by supervised or reinforcement learning, with the latter being more common. Schneider and Rosa [19] trained a multilayer perceptron to predict moves from Connect-4 board positions using supervised learning and data generated by symbolic agents. Their results demonstrated that a neural network could learn effective playing strategies from example games. However, their approach used relatively small network architectures restricted by the availability of computational power at the time, and their training labels were derived from other agents rather than a perfect solver.

Reinforcement learning has also been applied to Connect-4. Dabas et al. [20] evaluated a Double Deep Q-Learning (DDQN) agent against Minimax and MCTS, reporting competitive performance under certain computational conditions. Taylor and Stella [21] compared symbolic Q-learning with Minimax and MCTS, finding that their purely symbolic Q-table agent performed poorly, achieving a 0% winrate against the other agent classes. Wäldchen et al. [22] trained a Proximal Policy Optimization (PPO) agent using self-play and a convolutional neural network, which demonstrated the applicability of CNNs to Connect-4 policy and value learning. Taken together, these studies illustrate the capabilities of neural networks to learn Connect-4.

Existing research has also explored combining learned evaluations with search. Scheiermann and Konen [23] trained an n-tuple network to estimate game outcomes through temporal-difference learning and wrapped it with an AlphaZero-inspired Monte Carlo Tree Search procedure during testing. Their wrapped agent substantially outperformed the base network in their experiments, illustrating how search can make use of a learned representation to improve playing strength. Since the paper is predominantly interested in increasing the training efficiency of AlphaZero, n-tuple networks were used, however the present study aims at investigating the effects of compounding symbolic methods with neural networks instead.

Taken together, existing studies demonstrate that both symbolic search and neural learning can provide effective approaches to Connect-4 game playing, while several studies have also reported strong performance from systems that combine learned representations with search. However, relatively little work directly compares a neural decision-making approach with a neural model embedded within symbolic search under otherwise comparable conditions. The extent to which a supervised neural network can learn strategically useful information from perfect-play data, and whether a learned value representation combined with explicit search produces stronger game-playing performance than direct policy prediction, remains underexplored. The present study addresses this gap by training policy and value CNNs with the same architecture and perfect solver-derived training data. Comparing these agents against one another and symbolic game-playing benchmarks allows the study to investigate the difference in playing performance in the policy-target neural approach and value-target neurosymbolic approach.

MATERIALS AND METHODS

Dataset

A dataset of Connect-4 game positions was generated specifically for this work. Games were initialized from the empty Connect-4 board and progressed by selecting legal moves randomly for both players.

Connect-4 is played on a 7-column by 6-row board by two players who alternate turns (see Figure 1). On each turn, a player selects a non-full column and places one of their pieces into it. The played piece occupies the lowest available position in the selected column. A move is therefore legal if and only if the selected column contains at least one empty cell. When a player forms a sequence of four of their own pieces consecutively in the horizontal, vertical, or diagonal direction (i.e. 4-in-a-row), the game ends

immediately with a win for the player who formed the 4-in-a-row. If the board becomes completely occupied without either player forming a 4-in-a-row, the game ends in a draw. Unlike other games in the domain, Connect-4 does not use a cumulative score to determine the game outcome. The terminal state is evaluated based on deeming the first player to establish a 4-in-a-row the winner, or a draw if neither player succeeds in this.

After each resulting board position, the position was analyzed using the BitBully Connect-4 solver [24] to obtain two ground-truth labels: the best move according to the perfect solver and the perfect-play outcome of the position. For the former; in the case that multiple moves are winning, or all moves are losing, the solver prefers the move that leads to the quickest win or the most delayed loss. The latter represents whether the position leads to a win for Player 1, a draw, or a win for Player 2, annotated as +1, 0, or -1 respectively. The position is also annotated with a label representing the player to move.

Random moves were played by each player while sampling legal game positions. This method of position generation prevents introducing the strategic preferences of a particular game-playing agent into the dataset. The move-generation process therefore serves only as a mechanism for sampling the Connect-4 state space, while the strategic labels are independently determined by the perfect solver. This is particularly important for the present comparison, as using a strategic agent to generate the positions could bias the dataset toward states encountered under that agent's playing policy. Random legal play also ensures that the generated positions correspond to valid sequences of game actions rather than arbitrary board configurations, while providing positions from different stages and configurations of games.

BitBully was selected as the labeling mechanism because it provides exact analysis of Connect-4 positions and allows the required information to be queried conveniently through integration as a Python library. Figure 2 visualises position annotation with BitBully.

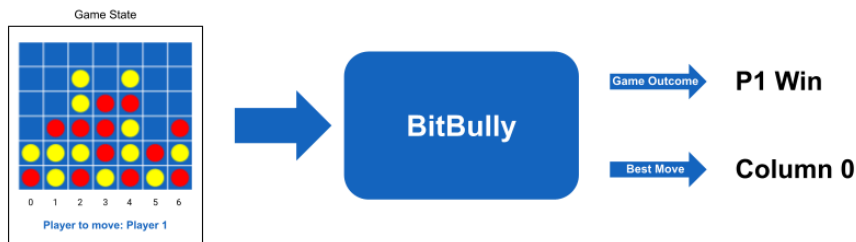


Figure 2: Annotation of game states using BitBully. The figure illustrates the position being input to the software, which provides the game outcome under perfect play as -1, 0 or 1; and the best move as a column from 0 to 6.

The training and testing datasets were generated independently, with 20,000 games sampled for the training dataset and 1,000 for the testing dataset, amounting to a total number of 425,347 positions for the

training dataset and 20,955 positions for the testing dataset. The best move, perfect-play outcome and game length distributions are shown in Figures 3-5.

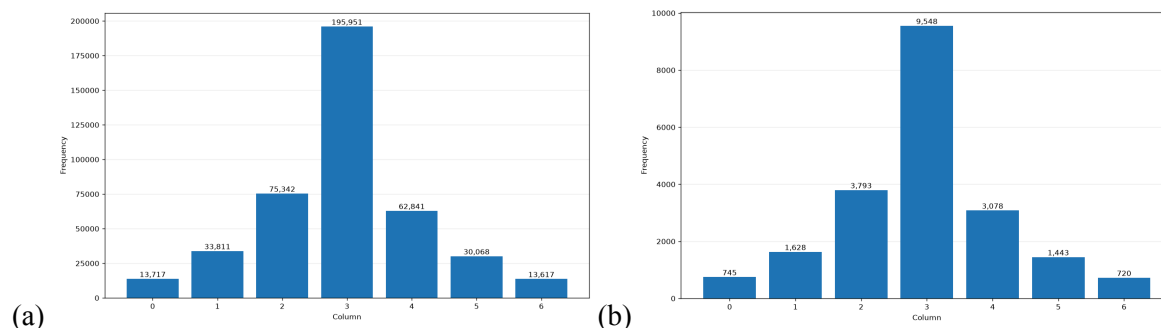


Figure 3: Best move distribution in the (a) training dataset and (b) testing dataset. The center column has the highest frequency whereas the edge columns have the lowest in both sets.

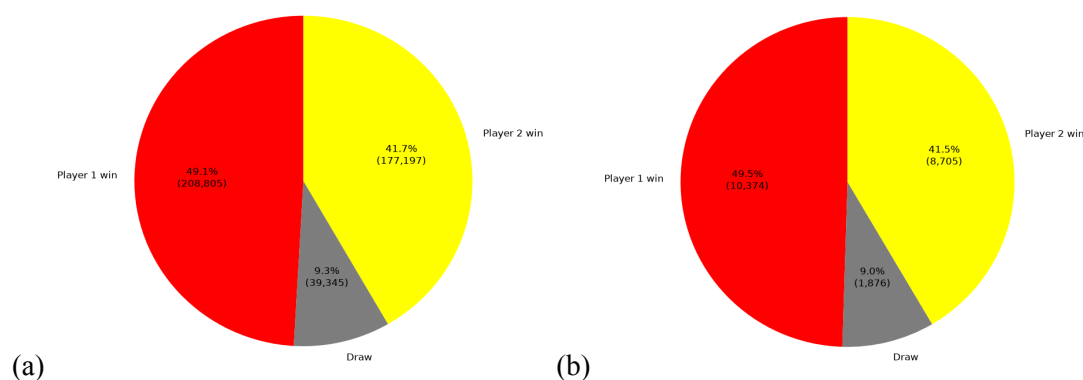


Figure 4: Perfect-play outcome distribution in the (a) training dataset and (b) testing dataset. The distributions are similar, with Player 1 wins contributing to a larger portion than Player 2 wins due to the inherent bias in Connect-4 that favors the first player to move.

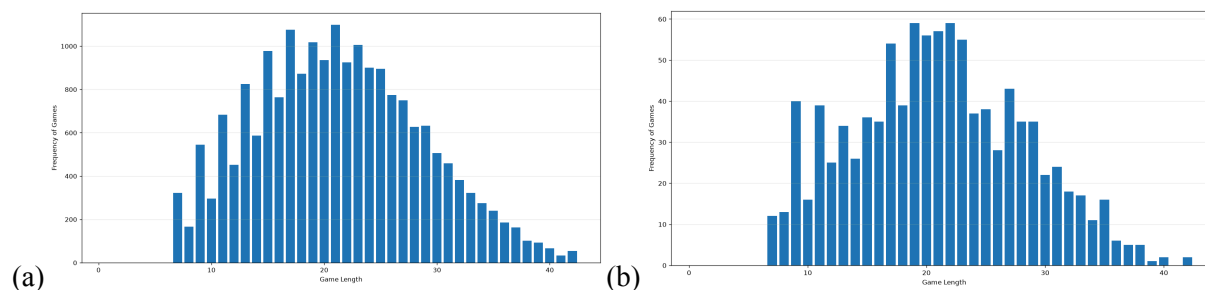


Figure 5: Game length distribution in the (a) training dataset and (b) testing dataset. The distributions are similar, with greater noise in (b) due to a smaller sample size.

The dataset is made available at Hugging Face¹.

Purely Neural Approach

The purely neural approach used in the present study is a policy network partially inspired by the networks in [19] and [22]. Instead of using the MLP architecture like [19], we employ a CNN architecture similar to [22], due to the superior 2-dimensional feature extraction capabilities of CNN architectures compared to MLP architectures. Besides, multiple studies have shown that CNNs have achieved high success for both policy and value representations [25] [26] [27]. The input encoding uses a $3 \times 6 \times 7$ input volume, which isolates player 1 and player 2 pieces in separate channels. As also pointed out in [22], the additional third channel used to encode empty spaces is redundant in perfect information scenarios, and hence has not been used. An additional channel has been used alongside the other two channels, which encodes the player to move. This is not strictly necessary, as this information can be inferred from the number of pieces on the board. However, empirical studies were made in this study that demonstrated that the addition of this extra channel achieves considerable improvements in training stability and speed (see Figures 12, 13). The network outputs 7 scalar values, activated with softmax, representing the network's confidence that each of the 7 possible columns that can be played in a given Connect-4 position are the best move. The chosen move of the network is considered to be the move with the highest confidence value. If the highest confidence move is illegal, the remaining moves are considered in descending order of confidence until a legal move is found.

The overall network consists of 5 convolutional layers, followed by 4 fully connected layers, as shown in Figure 6.

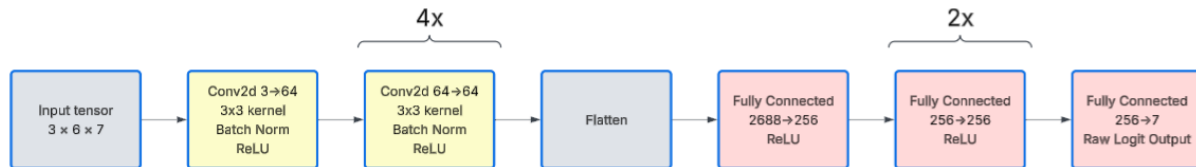


Figure 6: Schematic model of the policy network.

Neurosymbolic Approach

Minimax

Minimax search forms the symbolic component of the neurosymbolic agent designed in the present study. It is a two-player adversarial search algorithm that selects actions by recursively evaluating the possible future states resulting from each legal move. Both players are assumed to act optimally. Hence, one player attempts to maximize the expected utility of the game while the opposing player attempts to minimize it. Starting from the current board position, Minimax constructs a game tree by considering the legal moves available in the existing game state and recursively exploring future game states up to some depth d . The

¹ https://huggingface.co/datasets/MBaklaci/Connect4_RandomPlay_PerfectLabels

value of each position is then propagated back through the tree according to the player whose turn it is. This search can be used to generate a policy, where among all possible moves, the move that leads to the state with the largest minimax value can be chosen.

The algorithm for the minimax value of a game state s at depth d can be defined as:

MINIMAX(s, d):

 if $d = 0$ or s is a terminal state:

 return *UTILITY*(s)

 if *PLAYER*(s) = *MAX*:

 value $\leftarrow -\infty$

 for each move $\in$ *MOVES*(s):

$s' = \text{PLAY}(s, \text{move})$

 value $\leftarrow \max(\text{value}, \text{MINIMAX}(s', d - 1))$

 return value

 else:

 value $\leftarrow +\infty$

 for each move $\in$ *MOVES*(s):

$s' = \text{PLAY}(s, \text{move})$

 value $\leftarrow \min(\text{value}, \text{MINIMAX}(s', d - 1))$

 return value

where *UTILITY*(s) is an estimate of the utility of state s for the maximising player.

In the present study, this function is provided by the value CNN described in the following subsection. Rather than evaluating a non-terminal position using manually designed heuristics, the CNN estimates its perfect-play game outcome from the board configuration.

Alpha-Beta pruning is additionally applied to the Minimax search to reduce the number of game-tree nodes that must be evaluated. Because Alpha-Beta pruning only eliminates branches that cannot affect the final Minimax decision, it preserves the result of the underlying Minimax algorithm while reducing the computational cost [28] [15] [16].

Neural Utility Function

The neural utility function solves the regression problem that maps board states to perfect-play game outcomes within $[-1, 1]$, where positive values represent high confidence in player 1 victory and negative values represent high confidence in player 2 victory. The network architecture used for this network is

kept largely identical with the network described in the purely neural approach, with the major difference being the use of a single output, activated using tanh, representing the expected utility.

Similarly to the purely neural approach, this network consists of 5 convolutional layers, followed by 4 fully connected layers, as illustrated in Figure 7.

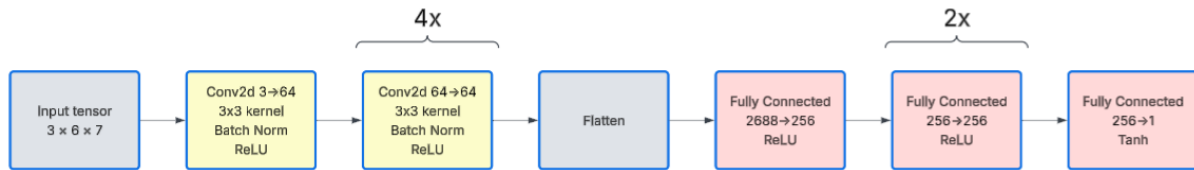


Figure 7: Schematic model of the value network

EXPERIMENTS AND RESULTS

Training and Implementation Details

For all models, a batch size of 64 was used, and training was facilitated over 50 epochs on the NVIDIA Tesla T4 GPU. The models were implemented and trained in Google Colaboratory [29].

A wide ensemble of ablation studies and experimentations were conducted by varying the network size, input encoding and dataset size. The experiment hyperparameters were adjusted using grid search, which revealed minor sensitivity to hyperparameter changes in our experiments. A learning rate of 0.01, weight decay of $1e-4$ and training momentum of 0.9 was used. Stochastic gradient descent was chosen as the optimiser. Mean squared error loss was used for value networks, and cross-entropy loss was used for policy networks.

Dataset Augmentation

Prior to training, left-right reflection across the vertical centre line was applied to each position in the training dataset to facilitate data augmentation. Connect-4 is symmetric about its vertical centre line, meaning that horizontally reflecting a valid board position produces an equivalent position that can be used as an additional training sample. Each original position was therefore reflected across the central vertical axis to generate an additional training example. The move label was reflected together with the board position. The perfect-play outcome label was unchanged by the transformation, as horizontal reflection does not alter whether a position is winning, drawing, or losing. The player-to-move information was likewise unchanged.

This augmentation has effectively doubled the number of training positions available in the training dataset. The testing dataset was left unaugmented, since the number of testing samples available are sufficient for the purposes of this experiment.

Main Findings

The training experiments have demonstrated that both the policy and value networks were able to learn generalizable knowledge from the solver-derived training dataset. The policy network achieved a minimum testing loss of 0.2843 and a maximum testing accuracy of 90.21%, while the value network achieved a minimum testing loss of 0.1149. The architectural ablation experiments on both policy and value networks showed that network size had minor influence on both testing loss or policy classification accuracy, suggesting that the selected architecture had sufficient capacity for the task. The five-convolutional-layer and four-fully-connected-layer architecture was therefore selected for the main experiments based on its slightly better policy performance. Experiments with various input encodings showed that explicitly representing the player-to-move information in the model input resulted in substantially greater training stability. Additionally, combining the board state representation of both players into a single signed channel produced largely similar performance to the separate-channel representation. However, experiments varying the training dataset configuration demonstrated a substantial dependence on training data. Removing the dataset augmentation significantly reduced testing loss and policy accuracy, and reducing the size of the dataset to 25% of its original size caused higher loss and lower accuracy still. The unaugmented and reduced datasets also resulted in considerable overfitting. Overall, these results conclude that training dataset size and symmetry-based augmentation had a substantially greater effect on network performance than the architectural and board-encoding variations investigated. Detailed analyses and interpretations of the training process and ablation results are provided in the Performance Analysis and Ablation Studies subsections.

The comparative experiments ran between agents demonstrated that both the purely neural agent and the neurosymbolic agent substantially outperformed random play, with the policy network achieving a 100.0% win rate and the neurosymbolic agents achieving 99.5–100.0% win rates across search depths one through five. Against a stronger greedy benchmark adapted from the literature, the policy network achieved a 69.5% win rate, whereas the neurosymbolic agents at various depths from 1-5 achieved win rates between 83.5% and 88.5%, with the depth-5 agent producing the strongest result and depth-1 the weakest. However, the depth-1 neurosymbolic agent already outperformed the policy network despite using only the shallowest search configuration. Direct head-to-head matches were also ran between the neural agent and neurosymbolic agents at various depths. The neurosymbolic agents won 56.5%, 69.5%, 78.0%, 77.0%, and 82.5% of games at depths one through five, respectively, demonstrating an overall improvement trend in playing strength with increased search depth despite the slight performance reduction at depth 4. The depth-5 neurosymbolic agent therefore achieved the strongest performance among all agents considered in the experiment, with an 88.5% win rate against the greedy benchmark and an 82.5% win rate against the policy network. These results show that the neurosymbolic agent consisting of the value network and search delivered better performance than the purely neural policy network under the experimental conditions of this study. However, because the policy and value networks were trained using different targets where one learns to predict the optimal move while the other learns to predict the game outcome, the experiments fall short of fully isolating the independent contribution of the search procedure; the observed advantage should therefore be interpreted as the combined effect of value learning and symbolic search. The generally stronger performance at greater search depths nevertheless

provides evidence that search is a strongly influential contributor of the playing strength of the neurosymbolic agent. The results of the comparative experiments are analysed at greater detail in the Comparative Analysis subsection.

Performance Analysis

The performance of the purely neural approach was initially evaluated using the testing loss and classification accuracy of the policy network. Testing loss measures the discrepancy between the probability distribution predicted by the network and the solver-derived move labels, while classification accuracy measures the proportion of board positions in the testing dataset for which the network predicts the solver-selected move in each epoch. These metrics provide an indication of the extent to which the network is able to learn the strategic information contained in the perfect-play training data.

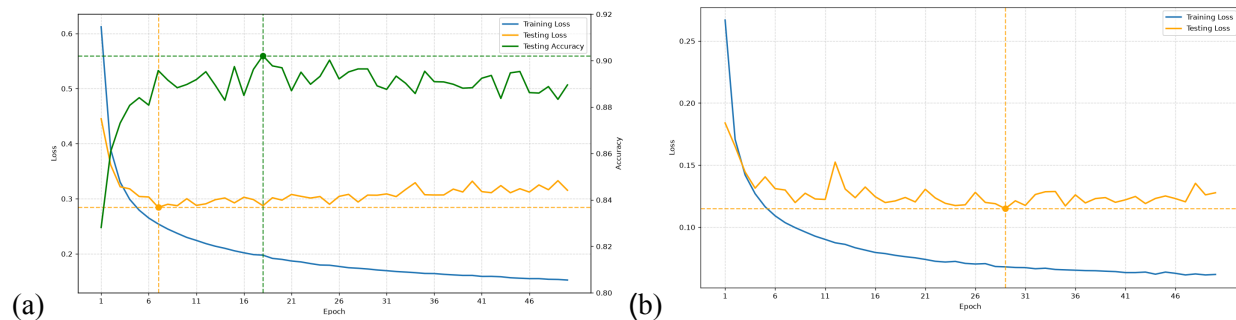


Figure 8: (a) The training loss, testing loss and accuracy values for the policy network. The maximum testing accuracy of 90.21% is observed in epoch 18. (b) The training loss and testing loss for the value network. The lowest loss of 0.1149 is observed in epoch 29.

The policy network demonstrated a minimum testing loss of 0.2843 and a maximum testing accuracy of 90.21% over the 50 training epochs. The values of these metrics throughout training is shown in Figure 8(a). The reduction in testing loss and the relatively high classification accuracy demonstrate that the network was able to generalize information from training and learn meaningful relationships between board configurations and solver-derived optimal moves. This suggests that neural networks are capable of learning strategically relevant information directly from Connect-4 game positions without explicitly performing game-tree search.

The classification accuracy does not represent complete game-playing performance, since an incorrectly predicted move may still be strategically viable even when it differs from the specific move selected by the solver, such as cases where multiple winning moves are present. Consequently, the playing strength of the policy network is evaluated separately through game-playing experiments presented later.

The network from the epoch with the highest testing accuracy was retained for future experiments.

The neural component of the neurosymbolic approach was evaluated using the testing loss of the value network. The testing loss measures the discrepancy between the outcome predicted by the network and

the solver-derived game outcome. This provides an indication of how accurately the network is able to learn the evaluation of game states from the perfect-play training data.

The value network achieved a minimum testing loss of 0.1149 at epoch 29 over the 50 training epochs. The training and testing loss throughout the training process are shown in Figure 8(b). The reduction in testing loss demonstrates that the network is capable of learning meaningful relationships between board configurations and their perfect-play outcomes. The effect of incorporating this learned evaluation into symbolic search is to be examined separately.

Ablation Studies

Multiple network architecture sizes were experimented with in order to identify the sensitivity of the model performance to architecture size. Figures 9 and 10 present the minimum testing loss values for both networks and the maximum accuracy values obtained by the policy network.

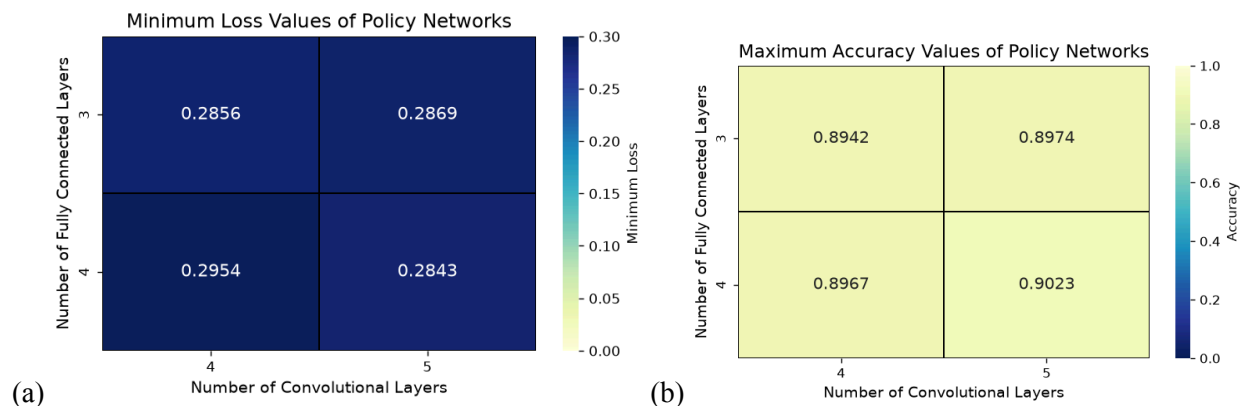


Figure 9: (a) Minimum testing loss and (b) maximum testing accuracy values obtained by policy networks of various sizes. The loss and accuracy is minimally affected by architecture size.

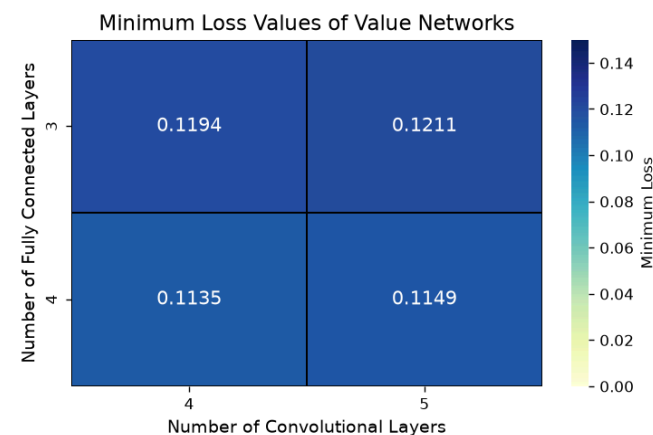


Figure 10: Minimum testing loss values obtained by value networks of various sizes. Similarly, the loss is minimally affected by architecture size.

Empirical results demonstrate that loss values and classification accuracy are very slightly, if at all, affected by network size for both value and policy networks. This provides evidence that the network architecture size used has sufficient capacity to learn relevant strategic information. Due to the slightly lower loss values and better accuracy in the policy network, the architecture with 5 convolutional layers and 4 fully connected layers was chosen for the main experiment.

Further experimentation was conducted with alternative input encoding formats. In addition to the three-channel encoding described previously, in which separate channels represent the positions of Player 1 and Player 2 pieces and the player to move, a two-channel encoding was evaluated. In this configuration, the player-to-move channel was removed, leaving only the two channels representing the pieces of each player. The player-to-move information was therefore not provided explicitly to the network, leaving the network to infer this information from the board configuration implicitly. This experiment was conducted to assess whether explicitly encoding the player to move provides a measurable benefit to network performance. Figures 11 and 12 present the results obtained with this input encoding.

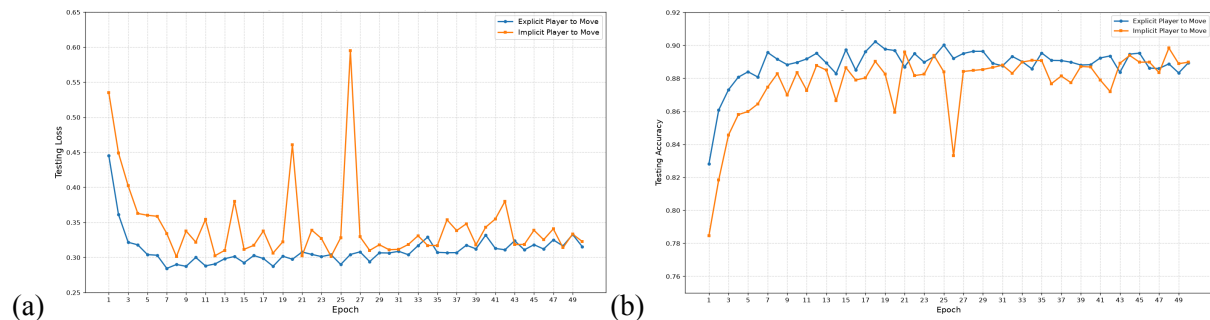


Figure 11: (a) Testing loss values and (b) testing accuracy of policy networks with implicit and explicit player-to-move information over 50 epochs. The implicit player-to-move demonstrates substantial training volatility.

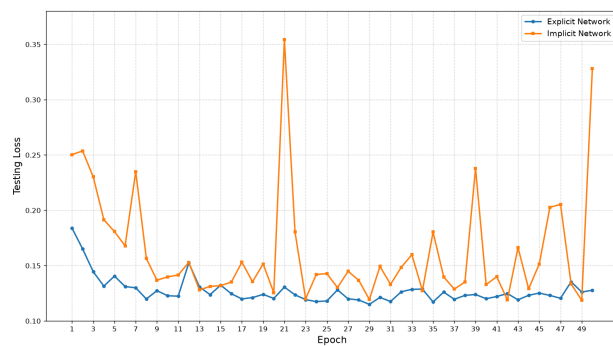


Figure 12: Testing loss values of value networks with implicit and explicit player-to-move information. Similarly, implicit player-to-move representation gives rise to training volatility.

The results indicate that explicitly encoding the player-to-move information in a separate channel provides a slight improvement in both loss and classification accuracy for the policy network. However, more notably, the networks using the explicit encoding demonstrate substantially greater training stability across epochs. In contrast, the loss and accuracy of the networks using the implicit encoding for both the policy and value networks fluctuate dramatically throughout training. This greater variability may suggest that requiring the networks to infer the player to move from the board configuration makes the learning process less stable.

A different input encoding was also evaluated in which the positions of both players were represented within a single channel. In this encoding, cells occupied by Player 1's pieces were assigned a value of +1, while cells occupied by Player 2's pieces were assigned a value of -1, with empty cells represented by 0. This therefore preserves the information of the game-state, reducing the board representation from two piece channels to a single channel. The player-to-move information was retained as a separate channel, resulting in a two-channel input representation overall. This experiment was conducted to determine whether unifying the representations of both players into a single signed channel affects the networks' learning capability. Figures 13 and 14 present the results obtained using this encoding.

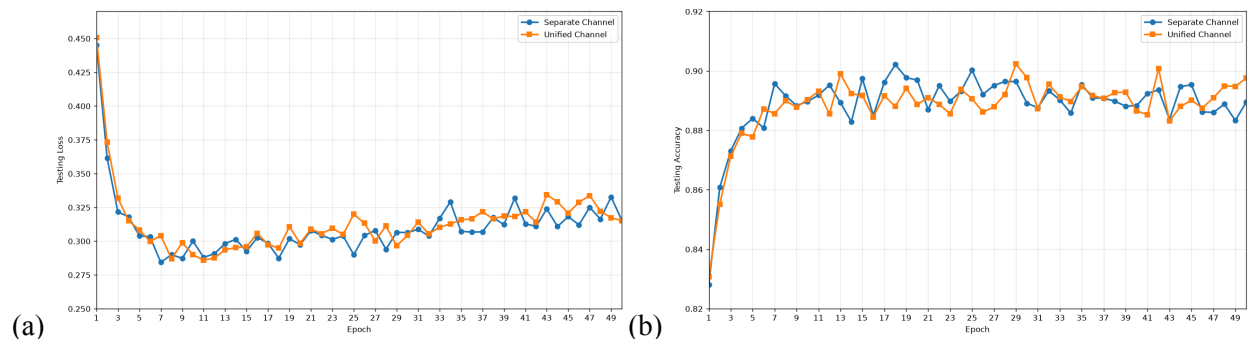


Figure 13: (a) Testing loss and (b) testing accuracy values of policy networks with separate and unified piece representation channels. The performance is comparable both in accuracy and loss.

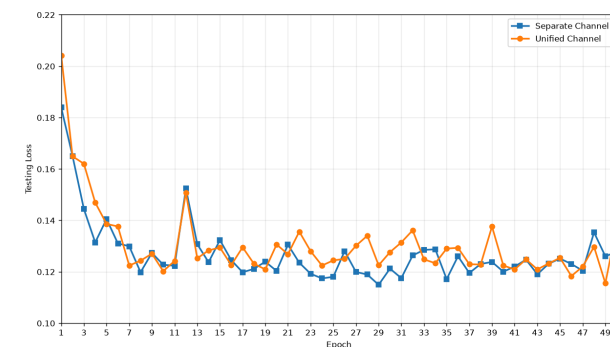


Figure 14: Testing loss values of value networks with separate and unified piece representation channels. The loss values are similar for both representations.

The separate-channel and unified signed-channel representations demonstrated very similar loss and classification accuracy values for both networks. Although the separate-channel representation achieves marginally better results in terms of loss, the observed difference is too small to provide meaningful evidence of a performance advantage. Furthermore, both encoding schemes exhibit comparable training stability, with no substantial differences in the fluctuations of their loss or accuracy values. These results suggest that combining the representations of both players into a single signed channel does not impair the networks' ability to learn strategic information.

Further experiments were conducted to investigate the effects of training dataset size and data augmentation on network performance. Three dataset configurations were evaluated. The first used the full training dataset with horizontal-reflection augmentation, corresponding to the configuration used in the main experiments. The second used the full training dataset without augmentation, allowing the effect of augmentation to be assessed independently of dataset size. The third used 25% of the original training dataset without augmentation. These configurations examine the contribution of the dataset augmentation and investigate the data requirements of the neural networks. The results of these experiments are presented in Figures 15 and 16.

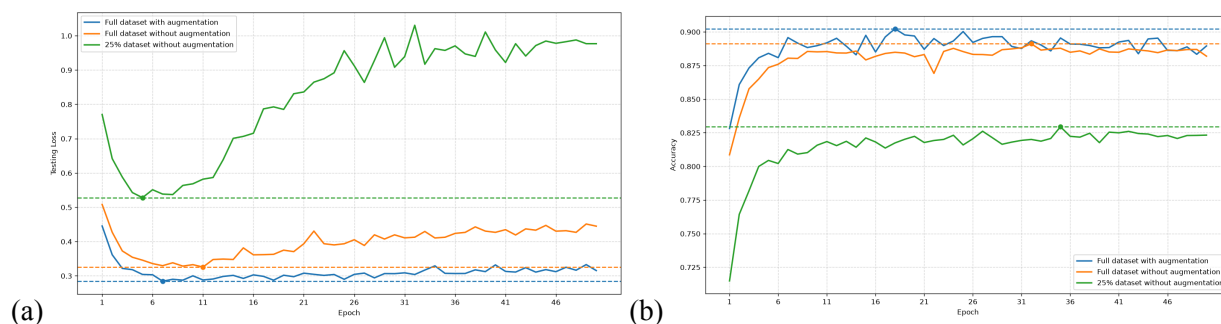


Figure 15: (a) Testing loss values and (b) testing accuracy of policy networks with various dataset configurations. Increasing the size of the dataset and performing augmentation both considerably benefit model accuracy and help mitigate overfitting.

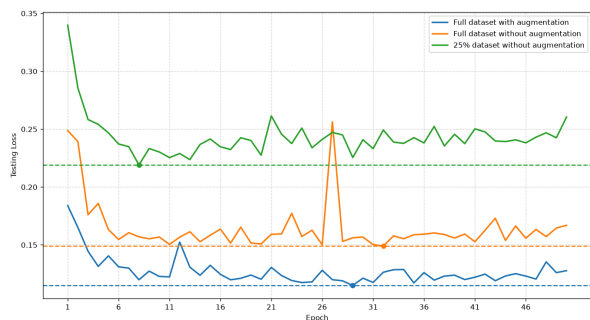


Figure 16: Testing loss values of value networks with various dataset configurations. Increasing the size of the dataset and augmentation has helped reduce loss considerably.

The results demonstrate that network performance is substantially dependent on the amount of training data available. The use of data augmentation produced a noteworthy reduction in testing loss compared with training on the full dataset without augmentation, indicating that the reflected positions provide useful information during training. Reducing the dataset size to 25% had a considerably greater effect with substantially higher testing losses and lower classification accuracy than the full-dataset configurations. Furthermore, the policy network trained on 25% of the dataset exhibited considerable overfitting. These results indicate that training dataset size is an important factor in the performance of both networks, with both dataset size and the use of symmetry-based augmentation having a substantial effect.

Comparative Analysis

The game-playing performance of the policy and neurosymbolic agents was evaluated through matches against two benchmark agents, a random agent and a greedy heuristic agent, followed by direct head-to-head matches between the policy network and the neurosymbolic agents. These experiments are purposed to evaluate the practical playing strength of the agents.

The random agent selects uniformly from the legal moves available in the current position. This provides a relatively weak baseline against which the performance of the learned agents can be evaluated.

The greedy agent evaluates the immediate game states resulting from each legal move using an objective heuristic evaluation function. The heuristic used by the greedy agent was adapted from the payoff function described by [21]. The heuristic evaluates the resulting board by considering every possible horizontal, vertical and diagonal four-cell segment on the board. For each four-cell segment, the number of pieces belonging to each player is examined. If a segment contains n pieces belonging to one player and no pieces belonging to the opponent, the corresponding n -in-a-row score is added to the overall heuristic value. Segments containing pieces from both players therefore do not contribute to the score. The score is positive when the segment contains the first player's pieces and negative when it contains the second player's pieces. Table 1 illustrates the n -in-a-row scores adapted from the paper.

Table 1: n -in-a-row scores used for the greedy heuristic

n	n -in-a-row score
4	∞
3	6

2	2
1	1

In addition to these four-cell pattern evaluations, additional scores are assigned according to the column in which the move is played. Once again, the scores are negated for the second player. Table 2 illustrates the move column bonuses.

Table 2: Move column bonuses used for the greedy heuristic

Move column	Bonus score
0	0
1	1
2	2
3	4
4	2
5	1
6	0

The total score from the four-cell pattern evaluations and the move column bonus are summed to obtain a final evaluation score which determines the choice of the greedy agent. The greedy agent chooses the move that maximises this score when it plays as the first player, and the move that minimises it when it plays as the second player.

For each faceoff comparison between two agents, 100 eight-ply openings were sampled randomly from the exhaustive Connect-4 8-ply positions dataset by Tromp [30]. Each opening was used to initialise two games between the agents being compared, where the players switched positions as the first player and second player. This is necessary due to Connect-4's inherent bias that puts the first player at an advantage. Consequently, each pair of agents played a total of 200 games, with both agents given an equal opportunity to play first. All faceoffs were made on the same 100 sampled openings.

The purely neural agent was evaluated against both the random and greedy agents. Each neurosymbolic agent, using Alpha-Beta-pruned Minimax search at depths one through five, was likewise evaluated independently against the benchmarks. Finally, the purely neural agent was evaluated directly against each neurosymbolic agent depth using the same aforementioned comparison protocol.

Performance Against Benchmarks

The purely neural agent demonstrated dominating performance against the random benchmark, winning 100% of the games played. This indicates that the policy network had learned strategically useful information, which has allowed it to find a strategy that is better than random. However, performance against the stronger greedy benchmark was lower, with the policy network winning 69.5% of games, losing 29.0%, and drawing 1.5%. The agent's win rate exceeding 50% indicates that the learned policy produces stronger game-playing performance than the symbolic heuristic under the evaluation protocol used.

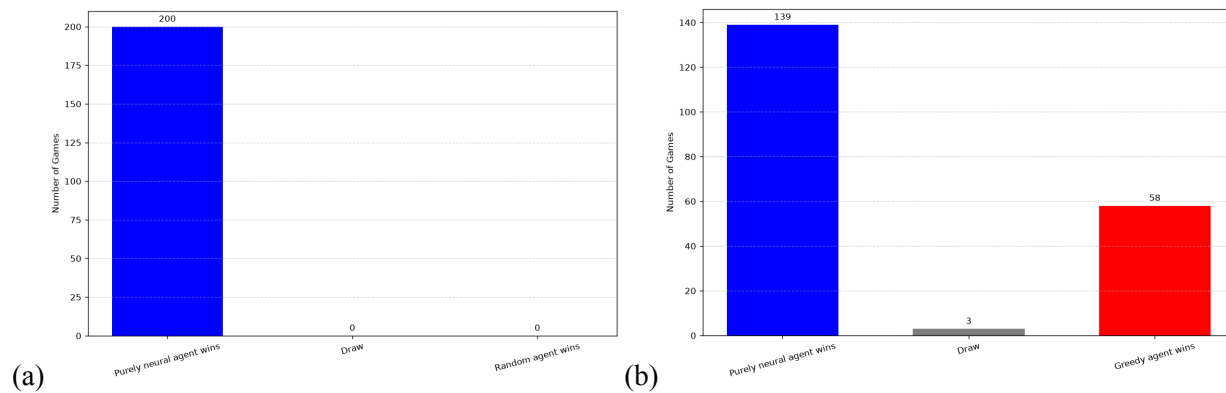


Figure 17: Results of 200 games played between the purely neural agent and (a) the random benchmark and (b) the greedy benchmark. The neural agent has won all games against the random agent and most games against the greedy agent.

The neurosymbolic agents at various depths exhibited similarly strong performance against the random benchmark. The depth-1, depth-2, and depth-4 agents each won 100.0% of their games, while the depth-3 and depth-5 agents each won 99.5%, with the remaining 0.5% of games being losses. The negligible variation across search depths indicates that the neurosymbolic configuration was similarly effective against random play.

More substantial differences were observed against the greedy benchmark. The depth-1 neurosymbolic agent achieved an 83.5% win rate, already exceeding the 69.5% achieved by the policy network against the same opponent. Increasing the search depth generally resulted in further improvements, with the depth-2 and depth-4 agents each achieving an 87.0% win rate, while the depth-5 agent achieved the highest performance at 88.5%. The depth-3 agent achieved a slightly lower win rate of 85.0%. Thus, although the relationship between search depth and win rate is not strictly monotonic, the deeper neurosymbolic agents generally demonstrated stronger performance against the greedy benchmark.

The substantially higher win rate of the depth-1 neurosymbolic agent compared with the policy network also indicates that the advantage cannot be attributed solely to deeper search. One possible explanation for this result is the difference in learning targets. Even at depth 1, the neurosymbolic agent provides superior performance to the policy network. At the same time, the depth-1 neurosymbolic agent explicitly evaluates the successor state resulting from each candidate move before selecting an action, unlike the

policy network which only considers the existing state, providing a small degree of symbolic reasoning. This suggests that both the difference in learning target and the use of symbolic search may contribute to the observed performance advantage, however further research is necessary to arrive at more conclusive results, which we leave as a future direction for studies.

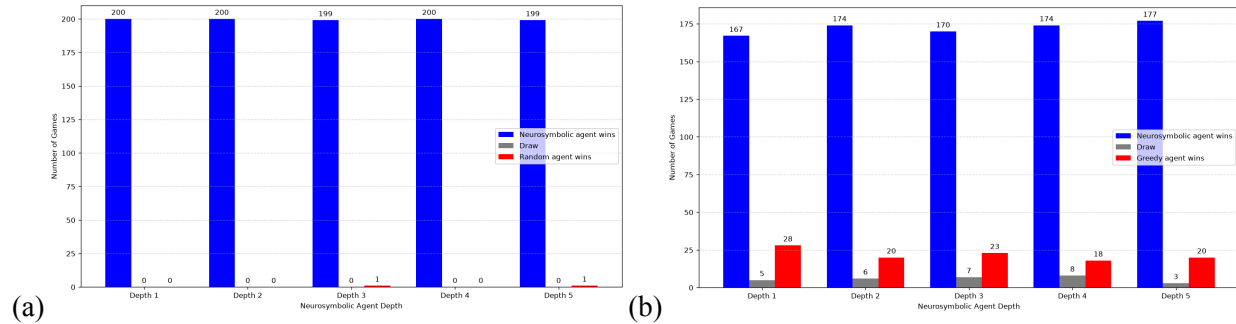


Figure 18: Results of 5 sets of 200 games played between neurosymbolic agents at various depths and (a) the random benchmark and (b) the greedy benchmark. The neurosymbolic agent won almost all games against random and the majority of games against greedy. All depths of the neurosymbolic agent have shown superior performance against greedy compared to the purely neural agent.

Head-to-Head Comparison Between Neural and Neurosymbolic Agents

The direct comparison between the policy network and the neurosymbolic agents provides a direct evaluation of the game-playing performance of the two approaches. The comparison is informative about the contribution of search since both neural networks use the same architecture and are trained from the same solver-derived dataset, however it cannot be conclusively said that the performance gain is caused only by the addition of search as the effects of learning target variation are not fully isolated. Regardless, results demonstrate that increasing search depth gives rise to stronger agents, strongly suggesting that search provides an influential contribution to the playing strength of the neurosymbolic agent.

Against the depth-1 neurosymbolic agent, the policy network won 35.0% of games, while the neurosymbolic agent won 56.5% and the remaining 8.5% were draws. Thus, in line with the results against the benchmarks, even the shallowest neurosymbolic configuration achieved a higher win rate than the policy network.

Increasing the search depth generally strengthened the neurosymbolic agent further. The depth-2 agent won 69.5% of games against the policy network, while the depth-3 agent won 78.0%. The depth-4 agent achieved a slightly lower win rate of 77.0%, indicating that additional search depth does not guarantee an improvement in every individual configuration. Nevertheless, the depth-5 agent achieved the strongest result, winning 82.5% of games against the policy network. The corresponding loss rate decreased from 35.0% for the depth-1 agent to 14.5% for the depth-5 agent.

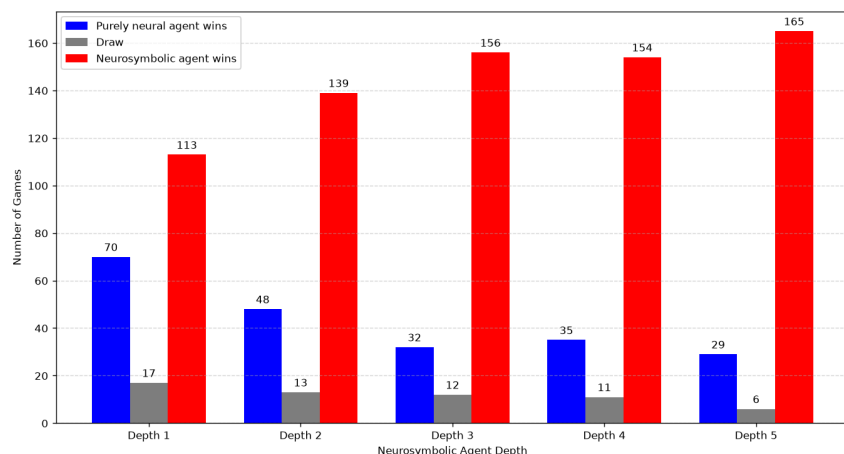


Figure 19: Results of 5 sets of 200 games played between the purely neural agent and neurosymbolic agents at various depths. The neurosymbolic agent beats the neural agent at all depths investigated.

The progression from a 56.5% win rate at depth 1 to 82.5% at depth 5 provides evidence that deeper search visibly improves the playing strength of the neurosymbolic agent. The depth-1 neurosymbolic agent outperforming the purely neural agent demonstrates that the value-learning approach, when combined with evaluation of immediate successor states, produces stronger game-playing performance compared to the policy-learning approach under the present study's experimental conditions.

Overall, the comparative experiments demonstrate that the policy network has successfully learned strategic information, evidenced by its 100.0% win rate against random play and 69.5% win rate against the greedy benchmark. However, the search-integrated value network agent consistently produced stronger game-playing performance. The depth-1 neurosymbolic agent already exceeded the policy network against both benchmarks and in head-to-head play with the purely neural agent. Moreover, deeper search has usually brought further improvements to the playing strength of the neurosymbolic agent. The depth-5 neurosymbolic agent, which was the strongest configuration, achieved an 88.5% win rate against the greedy benchmark and an 82.5% win rate against the policy network. These results therefore support the conclusion that, under the experimental conditions of this study, combining a neural value model with symbolic search can produce substantially stronger game-playing performance than direct neural policy prediction. However, because the two neural models were trained using different targets, the experiments do not establish conclusively the extent to which neurosymbolic playing advantage is caused due to the search procedure or the value-learning objective. The generally improved performance observed with increasing search depth however provides evidence that search is an important factor in the strength of the neurosymbolic agent.

DISCUSSION AND CONCLUSION

This study investigated the relative effectiveness of a purely neural policy network and a neurosymbolic agent that combines a neural value network with Alpha-Beta-pruned Minimax search for Connect-4 game

playing. In addition to this, a dataset of randomly generated Connect-4 positions annotated using perfect-play solver evaluations was contributed, and the effects of network architecture, input encoding, dataset size, and symmetry-based data augmentation were examined through a series of ablation experiments. The results showed that a purely neural approach can learn successful strategies, evidenced by its maximum testing accuracy of 90.21% and its 69.5% win rate against the greedy benchmark. However, the neurosymbolic agents consistently achieved stronger game-playing performance. Even the depth-1 neurosymbolic agent exceeded the policy network against the greedy benchmark, achieving an 83.5% win rate while the depth-5 agent achieved an 88.5% win rate. In direct head-to-head evaluation, the neurosymbolic agent demonstrated an overall positive trend in playing strength with depth, where the depth-5 neurosymbolic agent defeated the policy network in 82.5% of games. These findings indicated that, under the experimental conditions considered, a neural value model combined with symbolic search can outperform the purely neural policy model even at shallow search depths. The ablation studies further showed that the selected network architecture had sufficient capacity for the task and the performance was not significantly affected by architecture size. It has also been shown that explicitly encoding the player to move in the input improved training stability and produced a slight performance advantage. Dataset size and augmentation had more substantial effects, as reducing the size of the training dataset led to higher testing losses, lower accuracy in the policy network, and increased overfitting, whereas horizontal-reflection augmentation improved performance by providing additional valid data samples.

These results provide evidence that neural networks can learn to act strategically in rule-governed environments such as Connect-4 without the need for performing game-tree search. Nevertheless, the comparison demonstrates that a neural value model combined with symbolic reasoning can achieve stronger game-playing performance than the purely neural policy approach under the experimental conditions considered. The findings are consistent with the broader view that neural learning and symbolic reasoning can be complementary rather than competing approaches, particularly in environments with explicit rules and well-defined state transitions.

However, several limitations should be considered in interpreting the results of the present study. An important limitation concerns the experimental comparison between the purely neural and neurosymbolic agents. Although the two networks use the same architecture and are trained from the same solver-derived dataset, they are trained toward different targets, where the policy network learns to predict solver-selected moves whereas the value network predicts the perfect-play outcome of a position. Hence, the performance improvement of the neurosymbolic agent is not completely attributable to search alone. The training dataset was also likely a limiting factor in performance, evidenced by the positive trend in performance with data size in both networks. The experiments also considered search depths only up to five plies, so the point at which the performance benefit from search plateaus is unknown. The evaluation also used a fixed set of benchmark types and a fixed experimental protocol, hence, comparisons against stronger search-based agents, different heuristic agents, or other learned agents could provide a better assessment. Furthermore, the performance of the greedy benchmark is dependent on the particular scoring values used. The study also focused on a single game and a single network architecture family, limiting

the extent to which the findings can be generalized to other rule-based environments and other neural architectures.

A further limitation concerns the use of the testing dataset during model selection. Although test accuracy in this study remained roughly constant across multiple epochs, this does not eliminate the possibility of selection bias. Future work and deployment should utilize a separate validation set for model and checkpoint selection while reserving an independent test set for final evaluation.

An additional limitation arises due to the uncertainty of the reported win rates. Due to limited computational resources, the comparative experiments were run only once. Hence, it is not possible to report confidence intervals or uncertainties for the obtained results. Future work could conduct multiple evaluations using disjoint sets of openings and report confidence intervals or other appropriate measures of uncertainty to establish the reliability of the arrived conclusions in agent performances.

Future work could also investigate larger and more diverse datasets, alternative network architectures and training procedures, deeper or adaptive search, more varied baseline agents, and evaluations across multiple board games. It would also be useful to examine the computational cost of the two approaches and the quality of their decisions under time constraints for real-world deployment. Furthermore, an experimental procedure that controls the learning targets between the two approaches would be beneficial in isolating the effect of search on agent performance.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my mentor, Dr. Pramit Saha, for their guidance, feedback, and support throughout this research. His insights and willingness to share his expertise have been invaluable to the development of this work.

REFERENCES

- [1] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, no. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, ‘Deep Learning’, *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: <https://doi.org/10.1038/nature14539>.
- [3] B. Liang, Y. Wang, and C. Tong, ‘AI Reasoning in Deep Learning Era: From Symbolic AI to Neural–Symbolic AI’, *Mathematics*, vol. 13, no. 11, p. 1707, May 2025, doi: 10.3390/math13111707.

- [4] E. Giunchiglia, A. Tatomir, S. M. Cătălina, and T. Lukasiewicz, 'CCN+: A Neuro-Symbolic Framework for Deep Learning with Requirements', *Synergies between Machine Learning and Reasoning*, vol. 171, Aug. 2024, doi: 10.1016/j.ijar.2024.109124.
- [5] P. Hitzler, A. Eberhart, M. Ebrahimi, M. K. Sarker, and L. Zhou, 'Neuro-Symbolic Approaches in Artificial Intelligence', *National Science Review*, vol. 9, no. 6, Mar. 2022, doi: 10.1093/nsr/nwac035.
- [6] H. Shindo, Q. Delfosse, and D. S. Dhimi, 'BlendRL: A Framework for Merging Symbolic and Neural Policy Learning', in *International Conference on Learning Representations*, K. Kersting, Ed., Apr. 2025.
- [7] R. Hoehndorf and N. Queralt-Rosinach, 'Data Science and Symbolic AI: Synergies, Challenges and Opportunities', *Data Science*, vol. 1, no. 1–2, pp. 27–38, Oct. 2017, doi: 10.3233/ds-170004.
- [8] M. Garnelo and M. Shanahan, 'Reconciling Deep Learning with Symbolic Artificial Intelligence: Representing Objects and Relations', *Current Opinion in Behavioral Sciences*, vol. 29, pp. 17–23, Oct. 2019, doi: 10.1016/j.cobeha.2018.12.010.
- [9] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, 'Neuro-Symbolic Artificial Intelligence: Current Trends', *The European Journal on Artificial Intelligence*, vol. 34, no. 3, pp. 197–209, Mar. 2022, doi: 10.3233/aic-210084.
- [10] T. R. Besold et al., 'Neural-Symbolic Learning and Reasoning: A Survey and Interpretation', Nov. 10, 2017. Accessed: Sep. 17, 2026. [Online]. Available: <https://arxiv.org/abs/1711.03902v1>
- [11] W. Wang, Y. Yang, and F. Wu, 'Towards Data-And Knowledge-Driven AI: A Survey on Neuro-Symbolic Computing', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 2, pp. 878–899, Feb. 2025, doi: 10.1109/tpami.2024.3483273.
- [12] G. Jadhav et al., 'A Hybrid Search–Learning Framework for Artificial Intelligence in Board Games', *Journal of Information and Communications Technology: Algorithms, Systems and Applications*, vol. 1, no. 3, Dec. 2025, doi: 10.64189/ict.25315.
- [13] M. A. Ala'anzy, A. Madiyarova, A. Aigeldiyev, R. Zhanuzak, and O. Alnaseri, 'Connect-4 AI: A Comprehensive Taxonomy and Critical Review of Methods and Metrics', *Symmetry*, vol. 18, no. 2, p. 293, Feb. 2026, doi: 10.3390/sym18020293.
- [14] V. Allis, 'A Knowledge-Based Approach of Connect-Four', *ICGA Journal*, vol. 11, no. 4, pp. 165–165, Dec. 1988, doi: 10.3233/icg-1988-11410.
- [15] R. Nasa, R. Didwania, S. Maji, and V. Kumar, 'Alpha-Beta Pruning in Mini-Max Algorithm - an Optimized Approach for a Connect-4 Game', Apr. 2018.

- [16] L. Tommy, M. Hardjianto, and N. Agani, 'The Analysis of Alpha Beta Pruning and MTD(f) Algorithm to Determine the Best Algorithm to Be Implemented at Connect Four Prototype', in IOP Conference Series: Materials Science and Engineering, 2017.
- [17] K. Sheoran, G. Dhand, M. Dabas, N. Dahiya, and P. Pushparaj, 'Solving Connect 4 Using Optimized Minimax and Monte Carlo Tree Search', *Advances and Applications in Mathematical Sciences*, vol. 21, no. 6, Apr. 2022.
- [18] Y. Qiu, Z. Wang, and D. Xu, 'Comparison of Four AI Algorithms in Connect Four', in 2nd International Conference on Mechanical Engineering, Intelligent Manufacturing and Automation Technology, 2022.
- [19] M. O. Schneider and J. L. Garcia Rosa, 'Neural Connect 4 - a Connectionist Approach to the Game', in VII Brazilian Symposium on Neural Networks, Nov. 2002.
- [20] M. Dabas, N. Dahiya, and P. Pushparaj, 'Solving Connect 4 Using Artificial Intelligence', in International Conference on Innovative Computing and Communications, Aug. 2021.
- [21] H. Taylor and L. Stella, 'An Evolutionary Framework for Connect-4 as Test-Bed for Comparison of Advanced Minimax, Q-Learning and MCTS', May 26, 2024. Accessed: Sep. 17, 2026. [Online]. Available: <https://arxiv.org/abs/2405.16595v1>
- [22] S. Wäldchen, S. Pokutta, and F. Huber, 'Training Characteristic Functions with Reinforcement Learning: XAI-methods Play Connect Four', in Proceedings of the 39th International Conference on Machine Learning, 2022.
- [23] J. Scheiermann and W. Konen, 'AlphaZero-Inspired Game Learning: Faster Training by Using MCTS Only at Test Time', *IEEE Transactions on Games*, vol. 15, no. 4, pp. 637–647, Dec. 2023, doi: 10.1109/tg.2022.3206733.
- [24] M. Thill, BitBully. (Jun. 17, 2026). Accessed: Sep. 17, 2026. [PyPI]. Available: <https://pypi.org/project/bitbully/>
- [25] D. J. N. J. Soemers, V. Mella, E. Piette, M. Stephenson, C. Browne, and T. Olivier, 'Transfer of Fully Convolutional Policy-Value Networks Between Games and Game Variants', Feb. 24, 2021. Accessed: Sep. 17, 2026. [Online]. Available: <https://arxiv.org/abs/2102.12375v1>
- [26] V. Mnih et al., 'Human-Level Control through Deep Reinforcement Learning', *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[27] D. Silver et al., ‘Mastering the Game of Go With Deep Neural Networks and Tree Search’, *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.

[28] D. E. Knuth and R. W. Moore, ‘An Analysis of Alpha-Beta Pruning’, *Artificial Intelligence*, vol. 6, no. 4, pp. 293–326, 1975, doi: 10.1016/0004-3702(75)90019-3.

[29] Google Colab. <https://colab.research.google.com/>

[30] J. Tromp, ‘Connect-4’. 1995. doi: 10.24432/C59P43.