Size-Generalizable Reinforcement Learning for m,n,k-Games Using Fully Convolutional Networks

Chang Chia-Wei *

Posted Date: 15 April 2026

doi: 10.20944/preprints202604.1103.v1

Keywords: deep reinforcement learning; zero-shot generalization; fully convolutional network; DQN; MNK game

 doi:10.20944/preprints202604.1103.v1

Article

Size-Generalizable Reinforcement Learning for m,n,k-Games Using Fully Convolutional Networks

Chang Chia-Wei

Chienkuo Technology University; kaojj222@gmail.com

Abstract

This study addresses the problem of zero-shot generalization (ZSG) in deep reinforcement learning by proposing an MNK game strategy learning method based on a Fully Convolutional Deep Q-Network (FCN-DQN). Research in deep reinforcement learning aims to develop algorithms that can generalize well to unseen environments at deployment time, thereby avoiding overfitting to the training environment. Solving this problem is crucial for real-world applications, where environments are diverse, dynamic, and inherently unpredictable. By constructing a fully convolutional reinforcement learning policy network composed entirely of convolutional layers with padding to preserve feature map dimensions, the proposed model is able to handle input boards of varying spatial sizes. The model effectively learns local pattern-based strategies and approximations of the k-in-a-row evaluation function rather than performing global search. Furthermore, due to parameter sharing, the network has a relatively small number of parameters and is able to share policy representations across different board scales, thereby improving both sample efficiency and inference efficiency. Experimental results demonstrate that, after being trained on a 3×3 board, the proposed model is able to achieve a certain degree of zero-shot generalization performance in larger, unseen board environments.

Keywords: deep reinforcement learning; zero-shot generalization; fully convolutional network; DQN; MNK game

1. Introduction

Board game problems have long served as important benchmark environments in artificial intelligence and reinforcement learning research. According to the literature [1][2], the 3×3 tic-tac-toe game contains 362,880 possible board configurations. After removing invalid states, there are 255,168 valid games in total. Among these, the first player wins in 131,184 configurations, the second player wins in 77,904 configurations, and the remaining 46,080 games end in a draw.

Tic-tac-toe is known to be a first-player advantage game, meaning that the first player has a higher probability of winning, while the opponent can only lose or draw. However, under optimal play from both players, the game is solved, and players will aim for a win if possible, otherwise a draw, but never a loss [3].

Early studies applied neural networks based on Hamming distance classifiers to solve optimal strategies for 3×3 tic-tac-toe, with a computational complexity of $O(n^3)$, ensuring that the game always ends in either a win or a draw [4]. In recent years, deep reinforcement learning (Deep RL) methods have been widely applied to board game strategy learning. Approaches such as AlphaGo and AlphaZero combine deep neural networks with Monte Carlo Tree Search and self-play, achieving superhuman performance in various board games [5][6].

However, traditional methods often rely on fixed board sizes or global search mechanisms, making it difficult for them to generalize across different board dimensions or varying winning conditions [7][8]. This study propose a policy learning method based on a Fully Convolutional Deep Q-Network (FCN-DQN). By leveraging local pattern learning and parameter sharing, the proposed

 doi:10.20944/preprints202604.1103.v1

method enables cross-scale generalization over different board sizes, allowing effective action-value estimation and strategy selection even in unseen game environments.

2. Theoretical Framework

Traditional reinforcement learning methods for board game problems often rely on global state representations, and their generalization performance tends to degrade when the environment size changes. This issue is particularly evident in MNK board game tasks, where policy learning must handle varying board sizes as well as different k-in-a-row winning conditions. To improve zero-shot generalization capability, this study constructs a policy model based on the assumption of local pattern learning.

Assume that the board state can be represented as a three-channel tensor $S \in R^{3xHxW}$, where the three channels correspond to the own's pieces, the opponent's pieces, and empty positions, respectively. This study adopts a Fully Convolutional Deep Q-Network (FCN-DQN) as the action-value function approximator, whose functional form is defined as:

$$Q(a|S;\theta) = f_{cnn}(S;\theta), Q(S;\theta) = R^{HxW}$$

Where θ denotes the set of parameters of the convolutional neural network, and f_{cnn} represents the mapping composed of multiple convolutional layers. For any action $a = (i, j)$, its corresponding action value can be expressed as:

$$Q(a|S;\theta) = Q(S;\theta)_{i,j}$$

By using zero-padding, the spatial dimensions of the feature maps are preserved, enabling the model to accept board inputs of arbitrary sizes.

This study assumes that the optimal strategy in the MNK game can be approximately decomposed into a local spatial pattern recognition problem, where the model primarily learns to evaluate the value of local k-consecutive structures, rather than relying on global board search. Due to the parameter-sharing property of convolutional layers, the model can reuse learned policy representations across different board scales, thereby improving cross-scale zero-shot generalization performance.

3. Methodology

This chapter provides a detailed description of the proposed cross-scale MNK board game decision-making framework. The overall method formulates the MNK game as a reinforcement learning problem and employs a Fully Convolutional Deep Q-Network (FCN-DQN) as an approximation model of the action-value function. Unlike traditional deep reinforcement learning approaches that rely on fixed input dimensions, the proposed architecture is based on a fully convolutional design, enabling the model to process board inputs of arbitrary sizes [9], thereby improving cross-scale zero-shot generalization capability.

First, the MNK game is formalized as a Markov Decision Process (MDP), with explicit definitions of the state representation, action space, and reward mechanism. Next, the proposed FCN-DQN architecture is introduced, including the convolutional layer design and feature mapping strategy. Subsequently, the training procedure of the model is described, covering the temporal difference learning mechanism and exploration strategy design. Finally, the chapter presents the zero-shot generalization design philosophy, explaining how local pattern learning and parameter sharing enable the model to maintain stable performance across different board scales.

3.1. Problem Formulation

In this study, the MNK game strategy learning problem is formulated as a Markov Decision Process (MDP). The MNK game is a typical turn-based zero-sum game, whose objective is to learn an optimal policy function that can operate across different board scales and maximize long-term expected rewards.

Formally, the MNK game environment can be defined as a five-tuple:

 doi:10.20944/preprints202604.1103.v1

$$\mathbf{MDP} = (\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R}, \boldsymbol{\gamma})$$

Where:

- The state space $S \in R^{3xHxW}$ represents the board state tensor. The three channels correspond to the own's pieces, the opponent's pieces, and empty cells, respectively;
- The action space A is defined as the set of all legal move positions on the board;
- The state transition function $P(s'|s,a)$ describes the dynamics of the game state induced by actions;
- The immediate reward function $R(s,a)$ guides policy learning. A positive reward is given when a winning state is achieved, a negative reward is assigned upon loss, and zero reward is given otherwise;
- The discount factor $\gamma \in [0,1]$ is used to balance short-term and long-term returns. In this study, a fixed discount factor is adopted to stabilize the policy learning process.

In the MNK game, the optimal policy can be approximated by the action-value function as follows:

$$Q^*(s,a) = \max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a, \pi\right]$$

This study further assumes that the optimal policy in the MNK game can be approximately decomposed into a local spatial structure evaluation problem, where policy learning primarily depends on value estimation of local connection patterns on the board rather than global search. To improve generalization across different board scales, this work adopts a Fully Convolutional Deep Q-Network (FCN-DQN) as the policy approximation function. The policy mapping can be formulated as:

$$Q(a|S;\theta) = f_{cnn}(S;\theta)$$

Where:

- θ denotes the set of neural network parameters;
- $f_{cnn}(\cdot)$ represents the value estimation function composed of multiple convolutional layers;
- A zero-padding strategy is applied to enable the model to handle board inputs of varying sizes;
- The parameter-sharing mechanism in convolutional layers allows the model to learn scale-invariant local policy representations, thereby supporting zero-shot generalization.

By leveraging a fully convolutional architecture, the MNK game strategy learning problem is transformed into a local spatial pattern value estimation problem, enabling the policy function to maintain structural consistency and generalization capability across different board scales. The following sections further describe the proposed FCN-DQN architecture and training mechanism.

3.2. Fully Convolutional DQN Architecture

To achieve scale-invariant policy learning, this study designs a Fully Convolutional Deep Q-Network (FCN-DQN). The input consists of a three-channel board tensor, and the output is a Q-value map with the same spatial dimensions as the board. The model does not include any fully connected layers, ensuring that it can adapt to board sizes of arbitrary $H \times W$.

The input tensor is defined as:

$$S \in R^{3\times H\times W}$$

where the three channels respectively represent:

- the distribution of the own's pieces;
- the distribution of the opponent's pieces;

 doi:10.20944/preprints202604.1103.v1

- the empty cell information.
- The model output is defined as:

$$Q(S;\theta) \in R^{H\times W}$$

Each spatial location corresponds to a Q-value estimate for the action a_{ij}.

$$Q(s, a_{ij};\theta)$$

This output formulation avoids the need for action re-indexing, allowing actions to be directly mapped to board positions.

This study employs a four-layer stack of 3×3 convolutional kernels for local feature extraction:

$$F^l = ReLU(W^l * F^{l-1} + b^l)$$

The channel configuration of each layer is as follows:

Layer	Input Channels	Output Channels	Kernel	Padding
Conv1	3	64	3×3	1
Conv2	64	128	3×3	1
Conv3	128	128	3×3	1
Conv4	128	64	3×3	1

Padding is set to 1 to preserve spatial consistency between input and output dimensions. The stacking of multiple 3×3 convolutional layers gradually expands the receptive field, enabling the model to learn higher-level strategic representations from local connection patterns.

To improve gradient flow stability and preserve original board information, a residual connection mechanism is introduced. First, a 1×1 convolution is applied to project the input into a 64-channel feature space:

$$I = W_r * S$$

Where W_r denotes the 1×1 convolution kernel. The residual addition is then performed after the first convolutional layer:

$$F^1 = \mathrm{ReLU}(W^1 * S + b^1) + I$$

This design can be viewed as a fusion of shallow features and the original board representation, which helps alleviate the vanishing gradient problem, enhances the preservation of low-level spatial information, and improves training stability and convergence.

The final output layer is a 1×1 convolution:

$$Q(S;\theta) = W_o * F^4 + b_o$$

where the number of output channels is set to 1. This layer performs a linear value mapping at each spatial location without altering the spatial structure. The final output dimension is given by:

$$(\mathrm{batch}, H, W)$$

The action selection rule is defined as:

$$a^* = arg\max_{i,j} Q(s, a_{i,j};\theta)$$

For illegal moves, this study employs an action masking mechanism during both training and decision-making stages to exclude invalid actions.

Since the model contains no fully connected layers and consists solely of convolutional operations with fixed kernel sizes and weight sharing, its number of parameters is independent of the board size. This property enables the model to naturally generalize to different H×W board configurations and achieve zero-shot generalization.

3.3. Training Strategy

To improve the stability of policy network learning and sample efficiency, this study adopts a self-play reinforcement learning training mechanism. During training, the agent iteratively improves its decision-making capability by competing against itself, allowing policy updates to be performed without relying on external labeled data.

 doi:10.20944/preprints202604.1103.v1

For state exploration, an ε-greedy strategy is employed to balance exploration and exploitation. A relatively high exploration probability is set at the early stage of training to increase coverage of the state space. As training progresses, the exploration rate is gradually reduced through exponential decay, while a minimum exploration threshold is maintained to prevent premature convergence of the policy.

To improve sample efficiency and reduce the impact of temporal correlations on the learning process, this study introduces an Experience Replay mechanism. The experience replay mechanism reduces sequential correlations and stabilizes gradient updates by randomly sampling past transition experiences. The training samples consist of the game state, selected action, reward signal, next state, and terminal flag. Random sampling is used to construct training batches, thereby improving training stability.

To enhance the model's robustness to rotational and reflectional invariance on the board, this study adopts geometric data augmentation strategies. Training samples are transformed through rotation and flipping operations to generate multiple equivalent strategy samples, thereby improving the policy network's ability to learn symmetric game structures.

In terms of value function updates, this study adopts a Double Deep Q-Network (Double DQN) for policy evaluation. The policy network is responsible for action selection, while the target network provides a stable estimate of the target value, thereby mitigating the overestimation bias commonly observed in conventional Deep Q-learning. The target update is defined as:

$$\mathrm{Y} = \mathrm{r} + (1 - \mathrm{d})\gamma \mathrm{Q}(\mathrm{s}', \arg\max_{a} Q(s', a; \theta), \theta^{-})$$

The loss function adopts the Huber loss to enhance robustness against outlier reward samples. Let the action value predicted by the policy network be $\mathrm{Q}_{\mathrm{pred}}$, and the value estimated by the target network be $\mathrm{Q}_{\mathrm{target}}$, The prediction error is then defined as:

$$\delta = \mathrm{Q}_{\mathrm{pred}} - Q_{target}$$

Huber loss defined as:

$$\mathrm{L}(\delta) = \begin{cases} \frac{1}{2}\delta^2, & |\delta| \le 1 \\ |\delta| - \frac{1}{2}, & |\delta| > 1 \end{cases}$$

The parameters of the policy network are optimized by minimizing the Huber loss:

$$\mathrm{L}(\theta) = E[Huber(Q(s, a; \theta), Y)]$$

When the prediction error is small, a quadratic loss is used to accelerate convergence; when the error is large, it switches to a linear loss to reduce the influence of anomalous samples on gradient updates, thereby improving the stability of the reinforcement learning training process.

The core objective of the training strategy in this study is to enable the model to learn local spatial structural decision-making patterns in the MNK game, rather than relying on a global board search. Through self-play learning, an exploration decay strategy, an experience replay mechanism, and geometric data augmentation methods, the stability and generalization ability of the policy network are improved.

3.4. Zero-shot Generalization Design Summary

Traditional deep reinforcement learning methods typically rely on a fixed environment scale for training. When there is a discrepancy in board size between the training and testing environments, the performance of the learned policy often degrades.

This study is based on the assumption of local spatial pattern learning, which posits that the optimal strategy in the MNK game primarily depends on evaluating the value of local connection patterns on the board, rather than performing a global search over the entire board state. To this end, a fully convolutional policy network is adopted, enabling the policy function to make decisions based solely on local receptive fields.

Due to the parameter-sharing property of convolutional neural networks, the model is able to learn scale-invariant representations of local strategies. By employing zero-padding techniques, the

 doi:10.20944/preprints202604.1103.v1

policy network can accept board states of varying sizes as input and generate action-value distributions with spatial dimensions consistent with the board, thereby enabling cross-scale policy transferability.

During the training process, this study combines self-play learning with geometric data augmentation strategies. By applying rotation and reflection transformations to board states, the space of equivalent strategy samples is expanded, allowing the model to learn symmetry properties of the game. This design helps improve the generalization performance of the policy model on unseen board configurations and reduces the risk of overfitting.

4. Performance Evaluation

This section evaluates the effectiveness and generalization capability of the proposed fully convolutional network Deep Q-Network (FCN-DQN).

4.1. Simulation Environment

This study constructs an experimental environment for MNK game strategy learning and zero-shot generalization performance evaluation. To verify the generalization capability of the proposed fully convolutional Deep Q-Network policy model, a self-play reinforcement learning training mechanism is adopted in the experiments.

During the training phase, policy learning is conducted exclusively in a fixed 3×3×3 board game environment. An ε-greedy exploration strategy is employed to balance exploration and exploitation. During training, the exploration rate is gradually reduced using an exponential decay schedule to prevent premature convergence of the policy.

The policy model is trained using an experience replay mechanism to reduce correlations between sequential samples. Each training sample consists of the board state, action, immediate reward, next state, and a terminal flag. Model updates are performed using a Double Deep Q-Network (Double DQN) architecture, where the policy network is responsible for action selection, while the target network provides stable value estimation.

To enhance the model's ability to learn geometric structures of the board, rotation and flipping data augmentation strategies are introduced to increase the diversity of policy samples.

After training, zero-shot generalization testing is conducted. In the testing phase, the trained policy model is directly applied to unseen board sizes and win-condition settings, including MNK game environments with different board dimensions, varying win conditions k, and evaluations of cross-scale policy reasoning capability.

4.2. Simulation Parameters

The experimental environment in this study is primarily designed for the MNK game task. To evaluate the generalization capability of the proposed fully convolutional Deep Q-Network policy model across different board sizes, the experimental settings are defined as follows.

During the training phase, the model is trained exclusively through self-play in a 3×3 board environment, with the number of training episodes set to 50,000.

Policy exploration is conducted using an ε-greedy mechanism. The initial exploration rate is set to 1.0 and is updated using an exponential decay schedule, with a minimum exploration rate of 0.01.

For the reinforcement learning update strategy, the discount factor γ is set to 0.99 to balance short-term and long-term returns. The batch size for training is set to 128, and the experience replay buffer capacity is set to 50,000 to reduce the temporal correlation among training samples.

The neural network model adopts a fully convolutional Deep Q-Network architecture, with a learning rate set to 0.001, and the Adam optimizer is used for parameter updates. To stabilize target value estimation, a Double DQN framework is employed, together with a target network whose parameters are synchronized every 500 steps.

 doi:10.20944/preprints202604.1103.v1

Regarding data augmentation, when the board has a square structure, rotation and reflection geometric transformations are applied to increase the diversity of policy samples and improve the model's ability to learn symmetric structures of the board.

4.3. Metrics

To evaluate the policy learning capability and zero-shot generalization performance of the proposed fully convolutional Deep Q-Network model, this study designs three main evaluation metrics, including win rate, generalization retention rate, and policy advantage retention rate.

First, the win rate is used to measure the model's game performance under different board configurations. The win rate is defined as the ratio of the number of games won by the model to the total number of test games:

$$\mathrm{Win}_{\mathrm{rate}} = \frac{N_{win}}{N_{total}} \times 100\%$$

Where $\mathrm{N}_{\mathrm{win}}$ denotes the number of games won by the model, and $\mathrm{N}_{\mathrm{total}}$ denotes the total number of test games.

Second, to quantify the model's ability to retain its policy in unseen board environments, this study introduces the Relative Retention Rate. This metric measures the extent to which the model's performance in test environments is preserved relative to its performance in the training environment, and is defined as:

$$\mathrm{RelativeRetention} = \left(\frac{\mathrm{R}_{\mathrm{test}}}{R_{train}}\right) \times 100\%$$

Where $\mathrm{R}_{\mathrm{test}}$ is the win rate in the test board environment, and $\mathrm{R}_{\mathrm{train}}$ is the baseline win rate in the training environment.

Furthermore, to evaluate the model's learning advantage relative to a random policy, this study introduces the Advantage Retention Rate, defined as:

$$\mathrm{AdvantageRetention} = \frac{\mathrm{R}_{\mathrm{test}} - R_{random}}{R_{train} - R_{random}} \times 100\%$$

Where $\mathrm{R}_{\mathrm{random}}$ denotes the baseline performance of the random policy.

In an ideal random gameplay setting where no strategic information or skill is considered, the distribution of the three possible outcomes (win, loss, and draw) is expected to approach uniformity. Therefore, the win rate can be approximately regarded as 33% as a baseline for a random policy, which is used to measure the performance difference between the learned strategy and the random baseline.

4.4. Simulation Results

(A) Overall policy performance

To evaluate the effectiveness of the proposed fully convolutional deep reinforcement learning strategy, large-scale simulation experiments are conducted across multiple MNK game environments. The model is trained on a 3×3×3 board environment and is directly tested on other board scales without any additional fine-tuning, thereby verifying its zero-shot generalization capability.

The experimental results show that the model achieves a 94.7% win rate in the training environment, indicating that the policy network can effectively learn the fundamental decision-making patterns in the MNK game. Table 1 illustrates the win rate under the setting of seed = 656.

Table 1.

Board Size	Win Rate (%)	Relative Retention (%)	Absolute Drop (%)	Advantage Retention (%)
3×3×3	94.7	100.0	0.0	100.0 (baseline)
4×4×4	78.4	82.8	-16.3	73.6
4×4×3	96.9	102.3	+2.2	103.6

 doi:10.20944/preprints202604.1103.v1

Board Size	Win Rate (%)	Relative Retention (%)	Absolute Drop (%)	Advantage Retention (%)
5×5×5	71.8	75.8	-22.9	62.9
5×5×4	95.8	101.2	+1.1	101.8
5×5×3	95.3	100.6	+0.6	101.0
6×6×6	64.2	67.8	-30.5	50.6
6×6×5	94.3	99.6	-0.4	99.4
6×6×4	97.4	102.9	+2.7	104.4
6×6×3	97.6	103.1	+2.9	104.7

B）Zero-shot generalization capability and graceful degradation behavior.

As the board size increases, the model performance exhibits a graceful degradation trend. Specifically, in the 4 × 4 × 4, 5 × 5 × 5, and 6 × 6 × 6 environments, the model achieves win rates of 78.4%, 71.8%, and 64.2%, respectively. These results indicate that the proposed fully convolutional network architecture can maintain consistent cross-scale policy representations and demonstrates strong capability in learning spatially invariant features.

（C）Impact of the win-condition parameter (line length) on policy performance

A further analysis is conducted on the impact of the win-condition parameter k on the model's generalization ability. When k=3 is fixed, the model maintains high policy performance across different board scales, achieving an average win rate of approximately 96.1%, indicating that it successfully learns local spatial structural decision rules. In contrast, when k=m, the decision of game termination depends on global connection structures, making policy learning more challenging. As a result, the model performance decreases moderately, with an average win rate of approximately 77.3%. This phenomenon suggests that the local pattern learning assumption is more suitable for approximate modeling of MNK game strategies.

（D）Policy stability and pattern learning capability of actions

The distributional statistics indicate that the model exhibits a strong preference for the central region of the board during the initial move phase. This behavior is consistent with classical game strategy theory, where central positions typically provide higher strategic expansion value. In addition, no significant fluctuations in policy performance are observed, suggesting that the proposed model demonstrates good training stability and policy consistency.

（E）Summary of Experimental Results

Overall, the proposed fully convolutional deep reinforcement learning model demonstrates strong policy learning capability and zero-shot generalization performance in the MNK game task. Several notable characteristics are observed: when the win condition k is fixed, larger board sizes tend to be easier for the model; changes in k represent the primary challenge for generalization; and the convolutional network successfully learns the abstract concept of "connecting k consecutive pieces."

Through local spatial pattern approximation and parameter-sharing mechanisms, the model effectively enhances cross-scale game reasoning ability and achieves a graceful degradation of performance.

4.5. Scheme Evaluation and Comparison

To further evaluate the performance of the proposed fully convolutional Deep Q-Network policy model under different board configurations, this section provides a detailed analysis across all test boards. The evaluation compares metrics including win rate, relative retention rate, absolute performance drop, and advantage retention rate.

（A）Overview of Board Win Rates and Performance Metrics

The experimental results are shown in Table 1, where the 3×3×3 board is used as the training baseline environment, and the win rate of the random strategy is set to 33.0%.

As can be seen from the table, the model achieves the highest win rate on the training board (3×3×3), and still maintains strong performance on several unseen board configurations (e.g., 4×4×3,

 doi:10.20944/preprints202604.1103.v1

5×5×3, and 6×6×3). Both the relative retention rate and advantage retention rate indicate that the proposed model exhibits strong zero-shot generalization capability.

Figure 1 illustrates the model's generalization performance across different board sizes, the effect of the win-condition parameter k, the distribution of game lengths, as well as heatmaps of move distributions for the 3×3×3, 4×4×4, 4×4×3, and 5×5×5 board configurations.

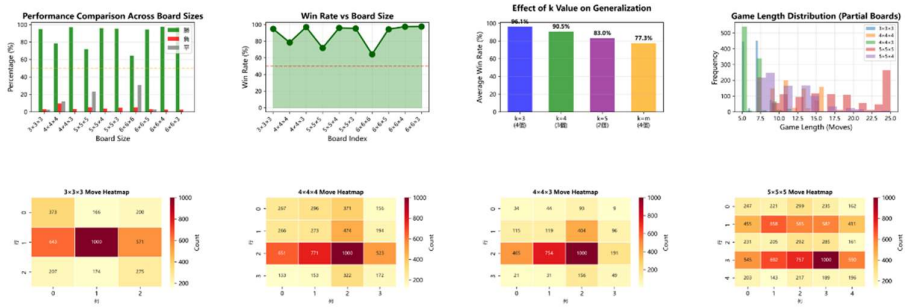

Figure 1. n

Figure 2 shows the heatmaps of move distributions for the 5×5×3, 5×5×4, and 6×6×6 board configurations.

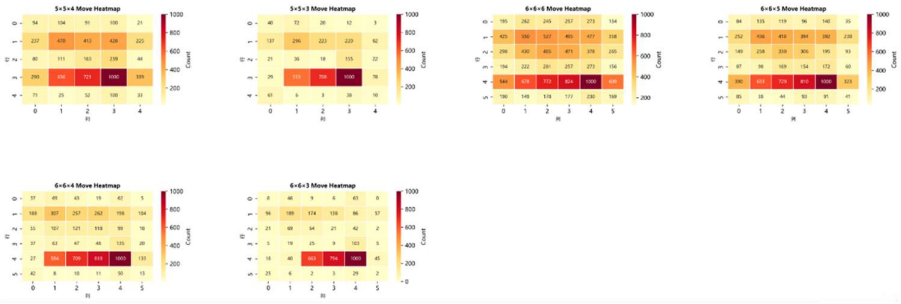

Figure 2

（**B**）Policy Stability and Pattern Learning Capability

Analysis of the initial move distribution reveals that the model prefers central positions on the board during the opening phase, which is consistent with classical game strategy theory. Failure case analysis further shows that short-term failures are mostly associated with errors in local pattern recognition, while long-term failures typically occur under larger board sizes or higher values of k. Overall, the model demonstrates stable performance and strong policy consistency, without significant fluctuations.

（C）Analysis by Win-Condition Parameter

The results are grouped and analyzed based on different win-condition parameters k, as follows:

k Value	Sample Size	Average Win Rate **(%)**	Advantage vs Random **(%)**	Relative Retention **(%)**
k=3	4	96.1	+63.1	101.5
k=4	3	90.5	+57.5	95.6
k=5	2	83.1	+50.1	87.8
k=m	4	77.3	+44.3	81.7

The results show that: When k is fixed, the model achieves the best generalization performance.

When k=m, the average performance drops by approximately 18.8 percentage points, reflecting the effectiveness of the local pattern learning assumption in approximating MNK game strategies.

Figure 3 shows the histogram distribution of absolute performance drop.

 doi:10.20944/preprints202604.1103.v1

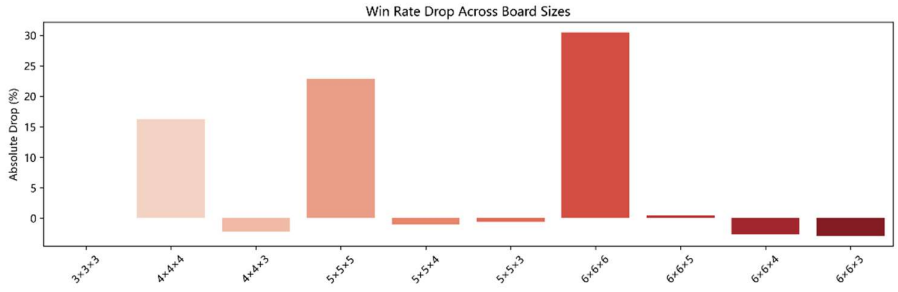

Figure 3

Figure 4 presents the line plot of advantage retention rate.

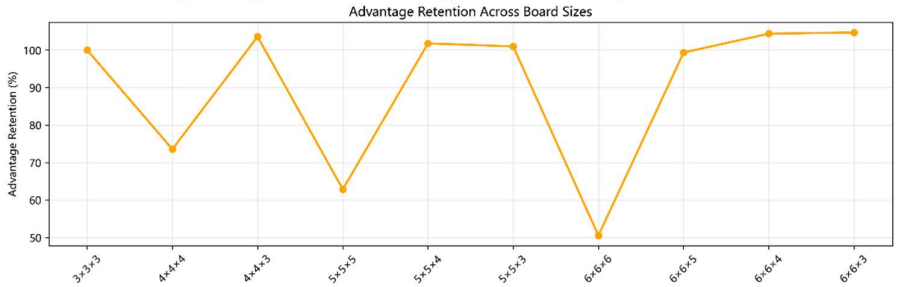

Figure 4

Figure 5 illustrates the heatmap distribution of relative retention rate.

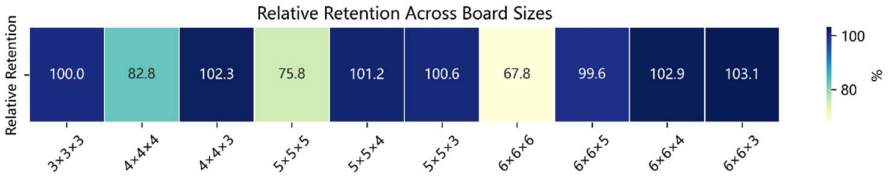

Figure 5

Figure 6 shows the radar chart for the 3×3×3 to 4×4×4 configurations, including advantage retention rate, relative retention rate, win rate, and absolute performance drop.

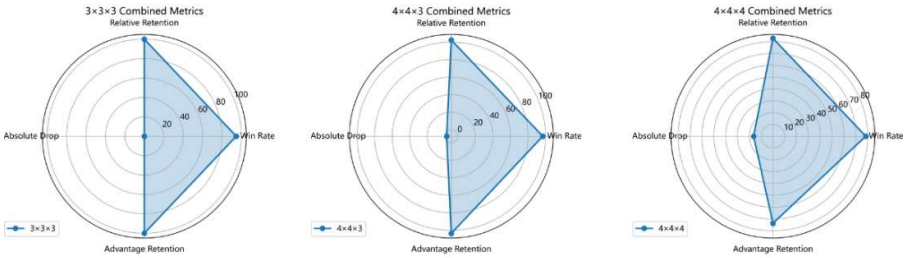

Figure 6

Figure 7 shows the radar chart for the 5×5×3 to 5×5×5 configurations, including advantage retention rate, relative retention rate, win rate, and absolute performance drop.

 doi:10.20944/preprints202604.1103.v1

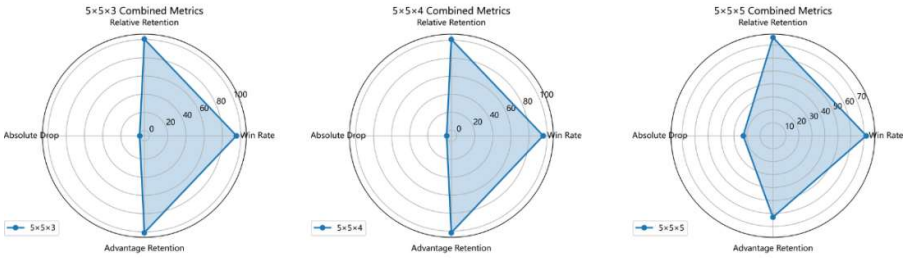

Figure 7

Figure 8 shows the radar chart for the 6×6×3 to 6×6×6 configurations, including advantage retention rate, relative retention rate, win rate, and absolute performance drop.

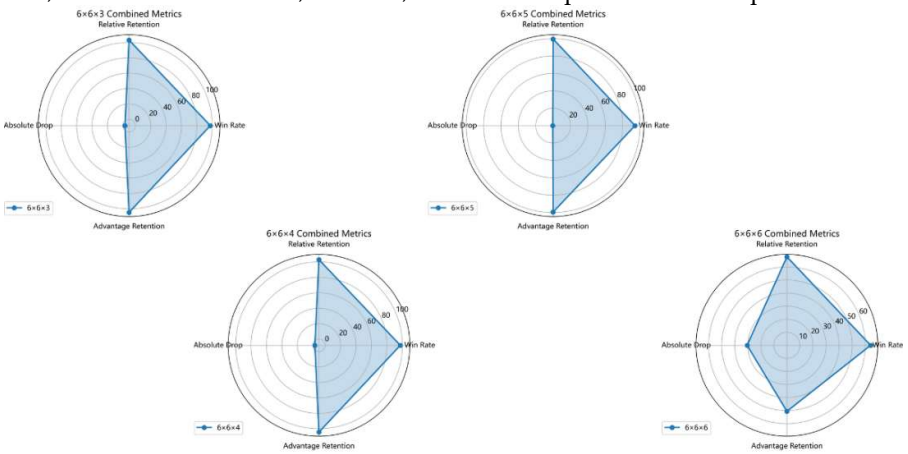

Figure 8

（D）Key Observations and Conclusions

Easier learning with fixed k: When k is fixed, larger board sizes lead to higher win rates. For example, in the k=3 group, the average win rate reaches 96.1%, indicating that the model successfully learns local patterns of connecting k consecutive pieces.

Variation in k is the primary challenge for generalization: when k changes, the win rate drops by approximately 18–20 percentage points, indicating a greater reliance on global structural reasoning.

The convolutional network successfully learns abstract patterns: the fully convolutional network effectively captures the concept of "connecting k consecutive pieces," and the combination of local pattern approximation and parameter sharing enhances generalization across different board sizes.

Good policy stability: the model exhibits a preference for central positions in the opening phase, with overall strategies that are consistent and stable.

In summary, the proposed fully convolutional Deep Q-Network model demonstrates excellent policy learning capability and zero-shot generalization performance in the MNK game, while exhibiting clear graceful degradation characteristics and strong cross-scale generalization ability.

5. CONCLUSIONS

This paper proposes a fully convolutional Network - Deep Q-Network (FCN-DQN) for the MNK game. By leveraging local pattern extraction and parameter sharing, the model is able to approximate optimal strategies across different board sizes and win conditions. The network relies solely on

 doi:10.20944/preprints202604.1103.v1

convolutional layers, enabling it to generalize directly from small-scale boards (e.g., 3×3×3) to larger boards, thereby achieving zero-shot generalization.

The experimental results show that the FCN-DQN achieves high win rates on both training and unseen boards. Analysis across different k values indicates that when the game is dominated by local patterns, the model performance remains stable, whereas in the case of k=m, which requires global reasoning, the average win rate decreases by approximately 19 percentage points. Metrics such as relative retention rate and advantage retention rate further demonstrate that the model successfully transfers local strategy knowledge to larger or more complex boards. Move heatmaps reveal a preference for central positions in the opening phase, indicating stable and consistent policy behavior.

In summary, the proposed FCN-DQN model demonstrates effective local pattern learning capability, strong cross-scale generalization performance, and a smooth performance degradation behavior under challenging conditions in the MNK game. These results highlight the potential of convolutional architectures for zero-shot policy learning.

References

1. How many tic-tac-toe (noughts and crosses) games are possible?, http: //www.se16.info/hgb/tictactoe.htm, accessed: 2021-1-15.
2. C.-H. Chou, Using tic-tac-toe for learning data mining classifications and evaluations, International Journal of Information and Education Technology 3 (4) (2013) 437.
3. Singh, K. Deep, A. Nagar, A" never-loose" strategy to play the game of tic-tac-toe, in: 2014 International Conference on Soft Computing and Machine Intelligence, Vol. 1, IEEE, 2014, pp. 1–5
4. N. F. Rajani, G. Dar, R. Biswas, C. K. Ramesha, Solution to the tic-tactoe problem using hamming distance approach in a neural network, in: 2011 Second International Conference on Intelligent Systems, Modelling and Simulation, Vol. 1, 2011, pp. 3–6. doi:10.1109/ISMS.2011.70.
5. D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, … & D. Hassabis, "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm", arXiv:1712.01815, 2017.
6. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, … & D. Hassabis, "Mastering the game of Go without human knowledge", *Nature*, vol. 550, pp. 354–359, 2017.
7. J. Tan Chong Min and M. Motani, "Brick Tic-Tac-Toe: Exploring the Generalizability of AlphaZero to Novel Test Environments", arXiv:2207.05991, 2022.
8. E. Korkmaz, "A survey analyzing generalization in deep reinforcement learning," *arXiv preprint arXiv:2401.02349*, Jan. 2024, doi: 10.48550/arXiv.2401.02349.
9. J. Long, E. Shelhamer, and T. Darrell, ''Fully convolutional networks for semantic segmentation,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, MA, USA, Jun. 2015, pp. 3431–3440.