GomokuNet: A Novel UNet-style Network for Gomoku Zero Learning via Exploiting Positional Information and Multiscale Features

Yifan Gao
Northeastern University
Shenyang, China
yifangao@stumail.neu.edu.cn

Lezhou Wu
Northeastern University
Shenyang, China
20184005@stu.neu.edu.cn

Haoyue Li
Northeastern University
Shenyang, China
20185541@stu.neu.edu.cn

***Abstract*—Since the tremendous success of the AlphaGo family (AlphaGo, AlphaGo Zero, and Alpha Zero), zero learning has become the baseline method for many board games, such as Gomoku and Chess. Recent works on zero learning have demonstrated that improving the performance of neural networks used in zero learning programs remains nontrivial and challenging. Considering both positional information and multiscale features, this paper presents a novel positional attention-based UNet-style model (GomokuNet) for Gomoku AI. An encoder-decoder architecture is adopted as the backbone network to guarantee the fusion of multiscale features. Positional information modules are incorporated into our model in order to further capture the location information of the board. Quantitative results obtained by ablation analysis indicate that our GomokuNet outperforms previous state-of-the-art zero learning networks on the RenjuNet dataset. Our method shows the potential to improve zero learning efficiency and AI engine performance.**

***Index Terms*—Convolution Neural Network, Zero Learning, Gomoku Game, UNet, Attention**

I. Introduction

Gomoku is also called five in a row. Originating from ancient Japan, Gomoku has been popular all around the world because of its relatively simple rules and moderate difficulty. Gomoku is played with black and white stones on a board with a size of 15. The black plays first, and two players take turns placing one stone of their own color on an empty place. The player who gets a row of five stones horizontally, vertically, or diagonally first can win the game.

Zero learning, mainly based on reinforcement learning, has become a benchmarking method of many board games and been widely used in training other board game AI like ELF OpenGo [1], Polygames [2], and KataGo [3], since the birth of the AlphaGo family (AlphaGo [4], AlphaGo Zero [5], and Alpha Zero [6]). Moreover, convolution neural network (CNN), as one of the most important structures of deep reinforcement learning, is turned out to have a significant impact on the final performance of the AI engine.

There are few pieces of research on Gomoku AI based on zero learning. Shao's work [7] first put forward an effective way to predict the move of Gomoku with CNN. Other works [8]–[10] look into the neural network structures of Gomoku trained by the zero learning method in the absence of expert knowledge. However, the neural network structures in the former researches are relatively simple and cannot be used to train a powerful Gomoku AI model.

Many improvements on CNN have been employed in the game of Go. Residual networks [11] used by AlphaGo contributed the first significant performance improvement. Compared with the typical CNN, AlphaGo Zero's performance increased by 600 Elo with the help of the residual networks. Nevertheless, further research [2] shows that in board games, the lack of location information dramatically reduces the final performance of neural networks because the fully connected layer used in AlphaGo Zero breaks the location relations. As an improvement of AlphaGo Zero, Polygames uses fully convolutional networks (FCN) to maintain the location information and achieves optimal performance in many kinds of board games. The authors of Polygames claims that learning strategies in board games tend to be similar to pixel-intensive image segmentation instead of classical image classification because moves are naturally reflected on the board. KataGo, as an open-source implementation of AlphaGo Zero, uses many innovative approaches to improve the performance of the CNN, like pre-activated residual networks, global pooling structure, and fixup technology. In particular, the global pooling structure makes the convolution layer base on the global state and further strengthens the model's ability to capture global information.

Although former models enhance the information capture process, it is still challenging for them to fit pixel-level intensive prediction because single-scale CNNs only have limited convolutional perception radius, leading to the lack of global strategies. Previous studies on board games proved this point, for example, the "ladder mistake" in the Go AI that led to the loss of the game [1]. . Therefore, we are inspired to explore CNN structures based on multiscale feature fusion. As a classical design in computer vision, multiscale feature fusion extracts low-level and high-level features through different receptive fields. It has been widely used in most computer

Yifan Gao and Lezhou Wu are co-first authors. (Corresponding authors: Yifan Gao.)

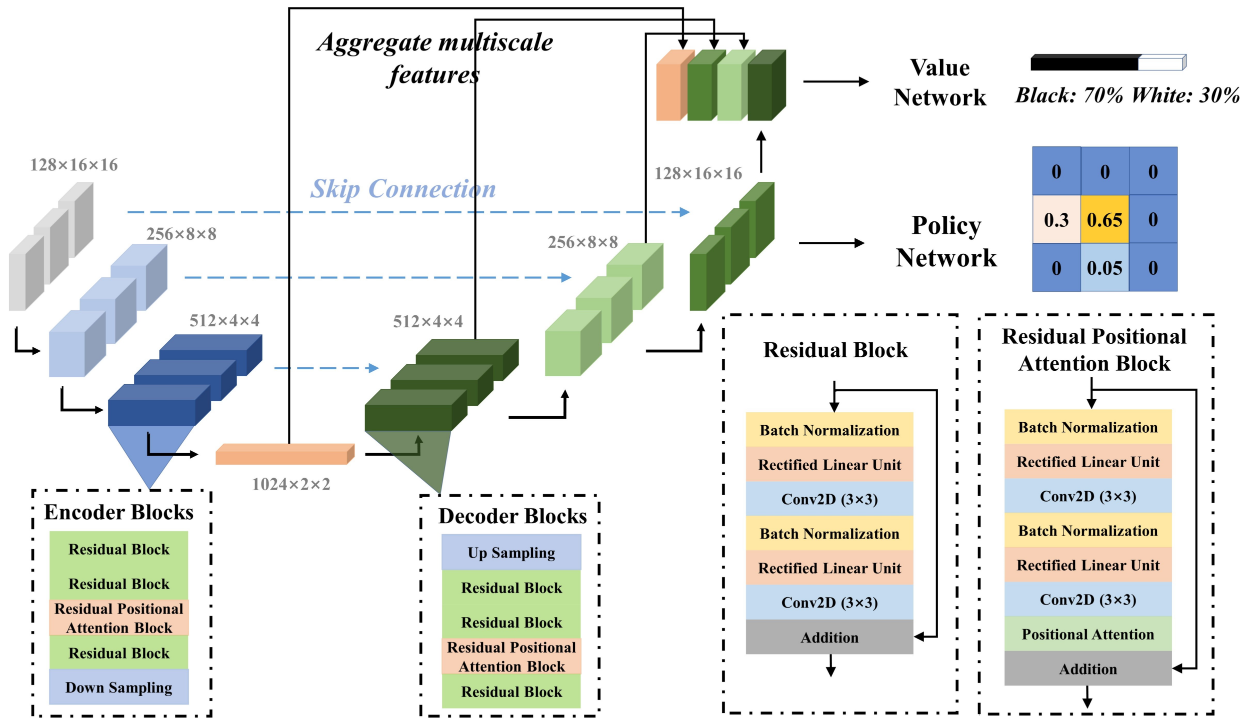

Fig. 1. An illustration of our proposed GomokuNet.

vision tasks and achieved advanced performance.

In this paper, we propose a novel CNN structure for the zero learning process of Gomoku game AI. Different from the previous structures based on classification models, our method is based on U-Net [12]. U-Net is a powerful CNN based on the encoder-decoder model and widely used in medical image segmentation. U-Net includes several times down-sampling and up-sampling. Thus, it can integrate low-level features and high-level features with different scales. Notably, to further capture the location information, we add the positional attention module (PAM) to the network according to the victory conditions of Gomoku. Compared with the widespread attention mechanisms, our PAM explicitly models the location information along with both horizontal and vertical directions. To be specific, the PAM integrates features along the horizontal direction and vertical direction, respectively, and encodes the feature map into the attention map sensitive to the location to improve the network's ability to capture the information of the location.

We carry out the experiment on a human Gomoku dataset called RenjuNet to verify the effectiveness of the proposed network. The result shows that our method achieves the optimal performance compared with the previous neural network models used in zero learning.

Our contributions are concluded as follows:

- We design a novel CNN structure based on U-Net. To the best of our knowledge, it the first time that the multiscale fusion structure has been used in zero learning.
- According to the rules of Gomoku, we put forward the PAM, which will integrate features along with horizontal and vertical directions. The location attention explicitly models the positional information of Gomoku stones and improves the performance of the model.

II. Method

A. Model Overview

In this paper, we propose a new network called GomokuNet to improve the learning efficiency of Gomoku AI. As shown in Figure 1, the basic structure of our model is a UNet-style network, including three down-sampling and up-sampling. Since the board size of Gomoku is 15, we need to use nearest-neighbor interpolation to sample it to 16x16 at the very beginning of the input and sample it back to 15x15 at the output of the policy head of the network.

B. UNet-style Architecture

The GomokuNet architecture is a UNet-style architecture that uses the strength of the residual networks and U-Net. The proposed GomokuNet takes advantage of the pre-activated residual block and the PAM. The residual block propagates information on the layer, allowing the establishment of a deeper neural network, which can solve the degradation problem of each encoder. This improves inter-channel dependency while reducing the computational cost. The proposed GomokuNet architecture contains three encoder blocks and three decoder blocks. Figure 1 shows a schematic diagram of the GomokuNet. The schematic diagram shows that each encoder block combines a residual block, a positional attention residual block, and a down-sampling layer. Each residual block

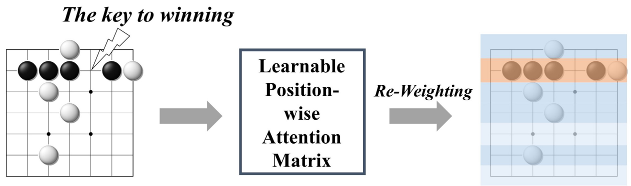

Fig. 2. According to the winning conditions of Gomoku, an intuitive idea of enhancing positional information.

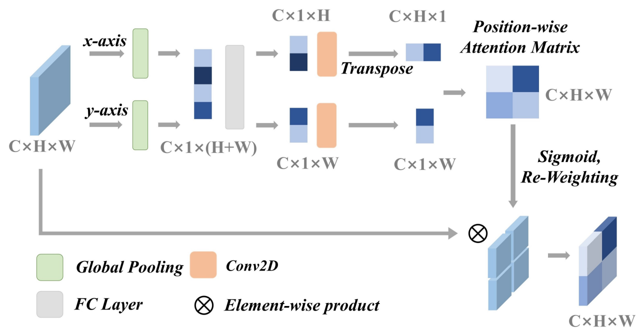

Fig. 3. Overview of the proposed PAM.

contains two 3×3 convolutional blocks and an identity mapping. Each convolution block includes a batch normalization layer, a rectified linear unit (ReLU) activation layer, and a convolution layer. The positional attention residual block adds a PAM before the identity mapping. At the end of the encoder block, a down-sampling layer is applied to reduce the spatial dimension of the feature map by half. The down-sampling layer is a convolutional block with a stride of 2.

The structure of the decoder block is similar to that of the encoder block, including a residual block, a positional attention residual block, and an up-sampling layer. To maintain the symmetry of the network, the up-sampling layer is placed at the forefront of the decoder block. The up-sampling layer uses nearest-neighbor sampling to double the spatial dimension of the feature map.

C. Positional Attention Module

The attention mechanism is an effective technology, which can help the model pay more attention to important information. KataGo uses the global pool structure in zero learning for the first time, which is a channel-based attention mechanism and improves the training speed by about 1.6 times. In this paper, we propose PAM to enhance network perception of the potential winning location of Gomoku. The horizontal or vertical direction of the winning key position of Gomoku usually contains important information (as shown in Figure 2, the horizontal axis of the winning key position of the black player contains four stones). Capturing such position information is very helpful to improve the prediction ability of the model. Given feature map $X \in R^{C\times H\times W}$, the structure of the PAM is as follows: (Figure 3)

- A average pooling layer along the x-axis and y-axis, output the statistics of the x-axis S_x (output shape $C\times1\times H$) and y-axis S_y (output shape $C \times 1 \times W$), respectively.
- Concatenate S_x and S_y along the last dimension to get S_{xy} (output shape $C \times 1 \times (H + W)$).
- A fully connected layer to S_{xy} outputs (output shape $C \times 1 \times (H + W)$), and divide S_{xy} into S_x (output shape $C \times 1 \times H$) and S_y (output shape $C \times 1 \times W$).
- A convolutional layer with a kernel size of 1 applied to S_x and S_y, then transpose S_x to $S_{x'}$ (output shape $C \times H \times 1$).
- Multiply $S_{x'}$ and S_y to get location-wise attention matrix X_a , and sigmoid activation applied to X_a (output shape $C \times H \times W$).
- Element-wise product using X and X_a, get the re-weighted feature map X (output shape $C \times H \times W$).

III. Experiments and Discussion

A. Dataset

In this article, the dataset used to train the deep learning model comes from RenjuNet[1]. RenjuNet is an online database of Gomoku containing as many as 96,000 games. All game records and related information are stored in XML format. RenjuNet includes 25 game rules, including different opening modes. Therefore, we eliminated the first six moves of each game and finally got a dataset containing 2301460 samples. We divide the dataset into a training set and a test set. The test set includes 15000 games randomly sampled and contains 355087 samples.

B. Baseline Model

In this work, we compare GomokuNet with several state-of-the-art previous works, including ConvNet, MobileNet, AlphaNet, PolyNet, KataNet.

- ConvNet: [7] designs a CNN to predict the moves of Gomoku, and it has achieved good performance in the RenjuNet database. This article uses a classic CNN, so we call it ConvNet.
- MobileNet: [13] applies MobileNet and residual networks to the game of Go and achieves a balance between speed and performance.
- AlphaNet: We call the CNN architecture used in AlphaGo Zero as AlphaNet.
- PolyNet: We call the neural network used in Polygames as PolyNet. In the experiment, the only difference between PolyNet and AlphaNet is using a fully convolutional network.
- KataNet: We call the neural network used in KataGo as KataNet. Our implementation of KataNet does not include domain-specific improvements (auxiliary ownership and scoring goals) because such information does not exist in Gomoku.

To ensure the fairness of the experiment, we set the different networks to roughly the same depth and width, that is, 20 blocks and 256 channels.

[1]http://www.renju.net/downloads/downloads.php

TABLE I
RESULTS OF OUR PROPOSED MODEL. FLOPS REPRESENT THE COMPLEXITY OF THE NETWORK.

Model	FLOPs (10^9)	Top-1	Top-5	Top-10	MSE Loss
ConvNet	2.6	43.02	77.00	87.83	0.950
AlphaNet	5.3	45.70	80.22	89.77	0.937
PolyNet	5.3	48.25	82.88	91.29	0.734
MobileNet	**1.2**	49.57	83.51	92.68	0.721
KataNet	4.9	51.33	84.08	92.94	0.708
GomokuNet	2.7	**52.40**	**85.02**	**93.56**	**0.664**

TABLE II
ABLATION ANALYSIS OF OUR PROPOSED MODEL. UA MEANS UNET ARCHITECTURE, RB MEANS RESIDUAL BLOCKS, MFA MEANS MULTISCALE FEATURE AGGREGATION.

UA	RB	PAM	MFA	Top-1	Top-5	Top-10	MSE Loss
✓				44.92	78.96	89.30	0.841
✓	✓			50.89	83.95	92.13	0.725
✓	✓	✓		52.28	84.17	**93.60**	0.748
✓	✓	✓	✓	**52.40**	**85.02**	93.56	**0.664**

C. Training

All models are trained for ten epochs using the SGD optimizer with a momentum of 0.9, and the learning rate is 0.005. Our experiments were carried out on the RTX 2080Ti graphic card. The implementation of the deep learning model uses PyTorch. During the training process, the data enhancement we used included eight positions of reflection and rotation.

The evaluation criteria of each model are the Top-1, Top-5, and Top-10 accuracy of the policy network and the MSE loss of the value network. The definition of MSE loss is as follows:

$$Loss = \frac{1}{N}\sum_{n=1}^{N}(predict_n - label_n)^2 \qquad (1)$$

D. Results

The comparison results are shown in Table I. It is observed that the model of capturing positional information demonstrates clear superiority. According to the experimental results, our model performs as the best on the RenjuNet dataset, outperforming the second performer (KataNet) with 1.07%, 0.94%, 0.62%, 0.044, respectively, and the model complexity is much lower (2.7 vs. 4.9). The lower complexity means that the zero learning AI engine can execute more Monte Carlo tree search (MCTS) per unit time.

E. Ablation Analysis

In this section, we performed ablation analysis on crucial modules, i.e., residual blocks, PAM, and multiscale feature aggregation method, to validate the effectiveness of our proposed model. The quantitative results of the ablation study on validation of the proposed GomokuNet are listed in Table II. It can be observed from the table that by gradually introducing residual blocks, PAM, and multiscale feature aggregation into the model, the performance of the model can be improved. In particular, multiscale feature aggregation mainly reduces the MSE loss of the value network.

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel CNN architecture for the Gomoku AI, which we refer to as GomokuNet. It introduces a UNet-style encoder-decoder network and the PAM to improve the ability of neural networks to extract features. Our network explicitly models the position information and builds the global strategy of the model through multiscale feature fusion, which is different from the existing zero learning networks. Under general settings, it achieved state-of-the-art results on the RenjuNet dataset. Ablation experiments also validate the effectiveness of the proposed method.

Although these preliminary results are encouraging, there is still much work to be done. Our network is still not trained under a complete zero learning process. The future work is to continue optimizing the network and applying it to the zero learning of Gomoku.

REFERENCES

[1] Y. Tian et al., "ELF OpenGo: an analysis and open reimplementation of AlphaZero," in International Conference on Machine Learning, May 2019, pp. 6244–6253.

[2] T. Cazenave et al., "Polygames: Improved zero learning," ICGA Journal, vol. 42, no. 4, pp. 244–256, Jan. 2020, doi: 10.3233/ICG-200157.

[3] D. J. Wu, "Accelerating Self-Play Learning in Go," arXiv:1902.10565.

[4] 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.

[5] D. Silver et al., "Mastering the game of Go without human knowledge," Nature, vol. 550, no. 7676, Art. no. 7676, Oct. 2017, doi: 10.1038/nature24270.

[6] D. Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," Science, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018, doi: 10.1126/science.aar6404.

[7] K. Shao, D. Zhao, Z. Tang, and Y. Zhu, "Move prediction in Gomoku using deep learning," in 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nov. 2016, pp. 292–297. doi: 10.1109/YAC.2016.7804906.

[8] Z. Xie, X. Fu, and J. Yu, "AlphaGomoku: An AlphaGo-based Gomoku Artificial Intelligence using Curriculum Learning," arXiv:1809.10595.

[9] P. Yan and Y. Feng, "A Hybrid Gomoku Deep Learning Artificial Intelligence," in Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference, Tokyo, Japan, 2018, pp. 48–52.

[10] X. Li, S. He, L. Wu, D. Chen, and Y. Zhao, "A Game Model for Gomoku Based on Deep Learning and Monte Carlo Tree Search," in Proceedings of 2019 Chinese Intelligent Automation Conference, Singapore, 2020, pp. 88–97.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[12] Ronneberger *et al.,* "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234–241.

[13] T. Cazenave, "Mobile Networks for Computer Go," *IEEE Transactions on Games*, pp. 1–8, 2020. doi: 10.1109/TG.2020.3041375.