

Bridging Local and Global Knowledge via Transformer in Board Games*

Yan-Ru Ju, Tai-Lin Wu, Chung-Chin Shih, Ti-Rong Wu[†]

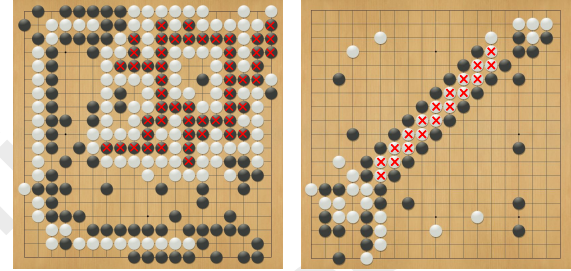
Institute of Information Science, Academia Sinica, Taiwan
{yanru, qwertgb1234, rockmanray, tirongwu}@iis.sinica.edu.tw

Abstract

Although AlphaZero has achieved superhuman performance in board games, recent studies reveal its limitations in handling scenarios requiring a comprehensive understanding of the entire board, such as recognizing long-sequence patterns in Go. To address this challenge, we propose *ResTNet*, a network that interleaves residual and Transformer blocks to bridge local and global knowledge. ResTNet improves playing strength across multiple board games, increasing win rate from 54.6% to 60.8% in 9x9 Go, 53.6% to 60.9% in 19x19 Go, and 50.4% to 58.0% in 19x19 Hex. In addition, ResTNet effectively processes global information and tackles two long-sequence patterns in 19x19 Go, including *circular pattern* and *ladder pattern*. It reduces the mean square error for circular pattern recognition from 2.58 to 1.07 and lowers the attack probability against an adversary program from 70.44% to 23.91%. ResTNet also improves ladder pattern recognition accuracy from 59.15% to 80.01%. By visualizing attention maps, we demonstrate that ResTNet captures critical game concepts in both Go and Hex, offering insights into AlphaZero’s decision-making process. Overall, ResTNet shows a promising approach to integrating local and global knowledge, paving the way for more effective AlphaZero-based algorithms in board games. Our code is available at <https://rlg.iis.sinica.edu.tw/papers/restnet>.

1 Introduction

AlphaZero [Silver *et al.*, 2018] has demonstrated superhuman performance in various board games, such as Go, chess, and Shogi. Following the advent of AlphaZero, researchers have successfully reproduced and extended the algorithm, achieving superhuman performance across a broader range of board games, such as Hex [Gao *et al.*, 2018] and Gomoku [Xie *et al.*, 2018; Liang *et al.*, 2023]. However, despite these advancements, several studies [Wang *et al.*, 2023;



(a) Circular pattern (b) Ladder pattern

Figure 1: Two challenging long-sequence patterns that require an understanding of global knowledge in 19x19 Go.

Tian *et al.*, 2019] have reported that AlphaZero remains vulnerable in scenarios requiring a more comprehensive understanding of the entire board, particularly in games with larger board sizes, such as 19x19 Go.

For example, recent research [Wang *et al.*, 2023] introduces a specialized program, named *cyclic-adversary*, designed specifically to find the weaknesses in KataGo [Wu, 2020], the current strongest open-sourced Go program. During the game, the cyclic-adversary subtly induces KataGo to form a circular pattern, as illustrated by the marked black stones in Figure 1a, while simultaneously encircling and capturing these stones. These circular patterns usually cover a wide range of the Go board, requiring a global understanding to handle these patterns effectively. Consequently, the cyclic-adversary successfully attacks KataGo with a success rate exceeding 90%. Another challenge requiring global understanding is the recognition of the *ladder pattern* in Go. In ladder, the player needs to simulate a long sequence of moves in zig-zag patterns across the entire board to capture or escape a group of stones, as shown by the marked white stones in Figure 1b. While this concept is relatively simple for human players, a previous study [Tian *et al.*, 2019] has shown that AlphaZero struggles to handle ladder patterns perfectly.

These examples highlight that handling global information remains a significant challenge in AlphaZero algorithms for board games, primarily due to their reliance on convolutional neural networks (CNNs), which are inherently designed to capture local patterns. To address this limitation, Transformers with self-attention mechanisms [Sutskever *et al.*, 2014;

*The full version is at <https://arxiv.org/abs/2410.05347>.

[†]Corresponding author.

Vaswani *et al.*, 2017] present a promising alternative, offering strong capabilities in capturing the global context. For example, similar to board games, computer vision – particularly image classification – primarily relies on extracting features in local patterns using CNNs. Recently, several network architectures, such as Vision Transformers (ViT) [Dosovitskiy *et al.*, 2020], Convolutional vision Transformers (CvT) [Wu *et al.*, 2021], and CoAtNet [Dai *et al.*, 2021], have been proposed to leverage Transformers to better capture global patterns in computer vision. However, directly applying these Transformers from computer vision to games does not yield better results [Czech *et al.*, 2023; Tseng *et al.*, 2024] due to the distinct objectives of the two domains: while the former focuses on achieving higher accuracy in image classification, the latter aims to maximize win rates in gameplay. This raises an important research question: *How can Transformers be effectively integrated into AlphaZero algorithms to improve global information processing while preserving the ability to recognize local patterns for board games?*

To answer this question, this paper introduces *ResTNet*, a novel network architecture specifically designed for AlphaZero algorithms which interleaves the use of residual and Transformer blocks to effectively balance between local and global information processing. The contribution of this paper can be summarized as follows:

- For the first time, the integration of Transformers and residual networks within AlphaZero for board games is thoroughly investigated. Experiments show ResTNet improves playing strength, with win rates increasing from 54.6% to 60.8% in 9x9 Go, 53.6% to 60.9% in 19x19 Go (using four KataGo models [Wu, 2020] as baselines), and 50.4% to 58.0% in 19x19 Hex (using MoHex [Arneson *et al.*, 2010] as the baseline), indicating a promising approach for training AlphaZero in board games.
- ResTNet demonstrates the ability to process global information by: (a) reducing the mean square error for recognizing circular patterns from 2.58 to 1.07 and lowering the attack probability from 70.44% to 23.91% against the cyclic-adversary among 24 games provided by Wang *et al.* [2023], and (b) improving ladder recognition accuracy from 59.15% to 80.01% in a human game collection.
- ResTNet offers insights into the interpretation of AlphaZero models and enhances explainability by visualizing attention maps from Transformers, capturing key concepts in both Go and Hex.

2 Background

2.1 AlphaZero

AlphaZero [Silver *et al.*, 2018] is a reinforcement learning-based algorithm that combines deep neural networks with search algorithms to master board games without the need for any human knowledge. The network architecture is composed of several residual blocks with convolutional layers, followed by two head outputs: a policy head for predicting the probability distribution of the next move, and a value head for estimating the win rate. The training process consists of

self-play and optimization phases. In the self-play phase, AlphaZero executes a Monte Carlo Tree Search (MCTS) [Coulom, 2007; Browne *et al.*, 2012] to generate self-play games and store them in a replay buffer. In the optimization phase, self-play games are uniformly sampled from the replay buffer and used to optimize the neural network model. The policy network aims to learn the move distribution that reflects the search results of the MCTS, while the value network aims to predict the game outcomes. The newly optimized network models are then used to generate new self-play games. Repeating this process allows AlphaZero to progressively enhance its performance and achieve superhuman performance.

2.2 Transformers in Computer Vision

In recent years, Transformers have achieved remarkable success in natural language processing [Vaswani *et al.*, 2017; Devlin *et al.*, 2019], and have also been successfully extended to computer vision. Unlike textual data, images contain rich positional and local information. To accommodate this, the Vision Transformer (ViT) [Dosovitskiy *et al.*, 2020] divides an image into patches, treating each patch as a token analogous to a word in a sentence. This allows the network to apply the Transformer’s sequential data processing capabilities to visual data.

Recent trends in computer vision show that integrating convolutional operations within the Transformer significantly enhances the performance [Wu *et al.*, 2021; Wang *et al.*, 2021; Guo *et al.*, 2022; Dai *et al.*, 2021]. For example, CoAtNet [Dai *et al.*, 2021] proposes a network architecture that begins with several convolutional blocks, followed by a series of Transformer blocks with relative attention [Shaw *et al.*, 2018]. CoAtNet has demonstrated state-of-the-art performance on image classification tasks, even under conditions of low data availability.

2.3 Transformers in Board Games

Transformers have recently been applied to board games, particularly in two directions: (a) using text representations of board games to frame game-playing AI as a natural language processing task, and (b) using visual representations of board games to apply transformers for processing these representations, similar to image classification tasks. For text representations, Ciolino *et al.* [2020] fine-tuned GPT-2 on Go games to predict the next move based on the SGF text format. Similarly, Feng *et al.* [2024] proposed ChessGPT, which finetunes a pre-trained Large Language Model (LLM) on chess datasets, including games, books, and videos, to predict moves or evaluate positions. Ruoss *et al.* [2024] trained a chess policy network through supervised learning on high-quality games annotated by Stockfish 16 [Romstad *et al.*, 2023], demonstrating that search-based programs can be distilled into large-scale transformers. Schultz *et al.* [2024] introduced the Multi-Action-Value (MAV) model, which integrates Transformers with internal and external planning and supports multiple board games within a unified framework.

On the other hand, for visual representation, researchers utilize Transformers to process board states directly. Sagri *et al.* [2023] proposed replacing the residual network with EfficientFormer [Li *et al.*, 2022b] and trained the policy and value

network from KataGo’s self-play games. Their experiments show that using EfficientFormer can achieve a higher win rate in scenarios where only CPU is available. Similarly, Monroe and Chalmers [2024] applied ViT architecture to train policy and value networks by supervised learning from a self-play game collection in the chess environment, achieving higher win rates when using only the policy or value network. In addition, Czech *et al.* [2023] proposed combining Mobile Convolutional Blocks [Sandler *et al.*, 2018] and Next Transformer Blocks [Li *et al.*, 2022a] in the chess environment. They improved the Transformer performance by identifying chess-specific input features.

While these studies have made progress in applying Transformers to visual board representations, they primarily adapt Transformers from computer vision without thoroughly investigating configurations tailored for board games. Furthermore, most of these works focus on supervised learning on game collections rather than training within the reinforcement learning framework of the AlphaZero algorithm. In contrast, our work focuses on visual representation by leveraging Transformers to design novel network configurations specifically for board games, bridging local and global knowledge within the AlphaZero training algorithm.

3 ResTNet

3.1 Network Design

This paper proposes *ResTNet*, a novel architecture that combines residual and Transformer blocks, as shown in Figure 2. It consists of several blocks, with each block being either a residual block or a Transformer block. For any given board, the network outputs a policy distribution and a value, as depicted in Figure 2a, where the architectures for the residual block and Transformer block are described in more detail in Figure 2b and 2c respectively. The residual block follows the same architecture as AlphaZero [Silver *et al.*, 2018], comprising two convolutional layers. The Transformer block is based on the standard Transformer [Vaswani *et al.*, 2017], augmented with relative position encoding [Shaw *et al.*, 2018], and uses four attention heads per block.

For simplicity, we use “R” to denote the residual block, and “T” to represent Transformer block in the network configuration for the rest of the paper. For instance, RRTRRT represents a network configuration that begins with two residual blocks, followed by a Transformer block, then two additional residual blocks, and finally another Transformer block. In addition to this notation, we employ a numerical naming method as an alternative way. For example, RRTRRT corresponds to the same configuration as 2R1T2R1T, while 6R1T represents a network configuration starting with six consecutive residual blocks followed by one Transformer block.

ResTNet offers a general version that covers a variety of network architectures. For example, in a 3-block configuration of ResTNet, if all blocks are residual blocks, represented as RRR or 3R, this configuration matches the architecture of the AlphaZero network. Conversely, if all blocks are Transformer blocks, denoted as TTT or 3T, the network aligns with ViT. Additionally, a configuration that starts with residual blocks and transitions to Transformer blocks towards

the end, such as RRT or RTT, corresponds to the CoAtNet architecture. Note that there are still some configurations, such as TRR or RTR, that are not covered in the above architectures but are included in ResTNet.

3.2 Feature Conversion

In ResTNet, the residual and Transformer blocks process different input features. The residual blocks utilize convolutional neural networks to process *2D feature maps*, whereas the Transformer blocks, designed for sequences handling, use *1D tokens* as input. In board games, precise and accurate positional representation is crucial, as each position on the board is unique and can differ significantly from neighboring positions. Therefore, we present a feature conversion method for transferring features between residual and Transformer blocks in ResTNet. Figure 2d illustrates feature conversion from residual to Transformer blocks. Note that the feature conversion from Transformer to residual blocks simply mirror Figure 2d left-right. The method converts the 2D feature maps into 1D tokens through a one-to-one mapping to preserve positional information, as described as follows.

The input/output representations of the residual block are consistent with the board representation, which is defined by dimensions $C \times H \times W$, where C represents the number of channels, and both H and W denote the height and width respectively. Hence, the initial board representation can be viewed as a specification of the residual block to the first conversion. Conversely, for Transformer blocks, the representation needs to be converted into 1D tokens, ordered according to the board positions, from the top-left to the bottom-right in either row-major or column-major manner. The size of 1D tokens is $H \times W$, with each token encoding a board position in a dimension of C . For a conversion from Transformer to residual block, these 1D tokens can further be reorganized into 2D feature maps, ensuring that each token is accurately rearranged to match the original board layout.

4 Experiments

This section presents an in-depth analysis of ResTNet in Go and Hex, selected for their reliance on global knowledge to understand gameplay. Subsection 4.1 evaluates the playing performance. Subsection 4.2 examines ResTNet’s ability to process global information, focusing on two well-known challenges in Go: recognizing *circular patterns* and *ladder patterns*. Finally, Subsection 4.3 explores the attention maps in ResTNet, offering insights into its explainability.

4.1 Playing Performance of ResTNet

Given that the proposed ResTNet allows for various combinations, we first evaluate different network architectures on 9x9 Go, which allows multiple evaluations under limited computational resources. Then, we apply the best-performing architecture discovered to 19x19 Go and 19x19 Hex.

Architecture Exploration on 9x9 Go

We use 6-block in 9x9 Go. Each residual block consists of 256 filters, whereas each Transformer block comprises 81 (9×9) tokens with an embedding size of 256. Table 1 lists the 6-block networks we evaluated, along with

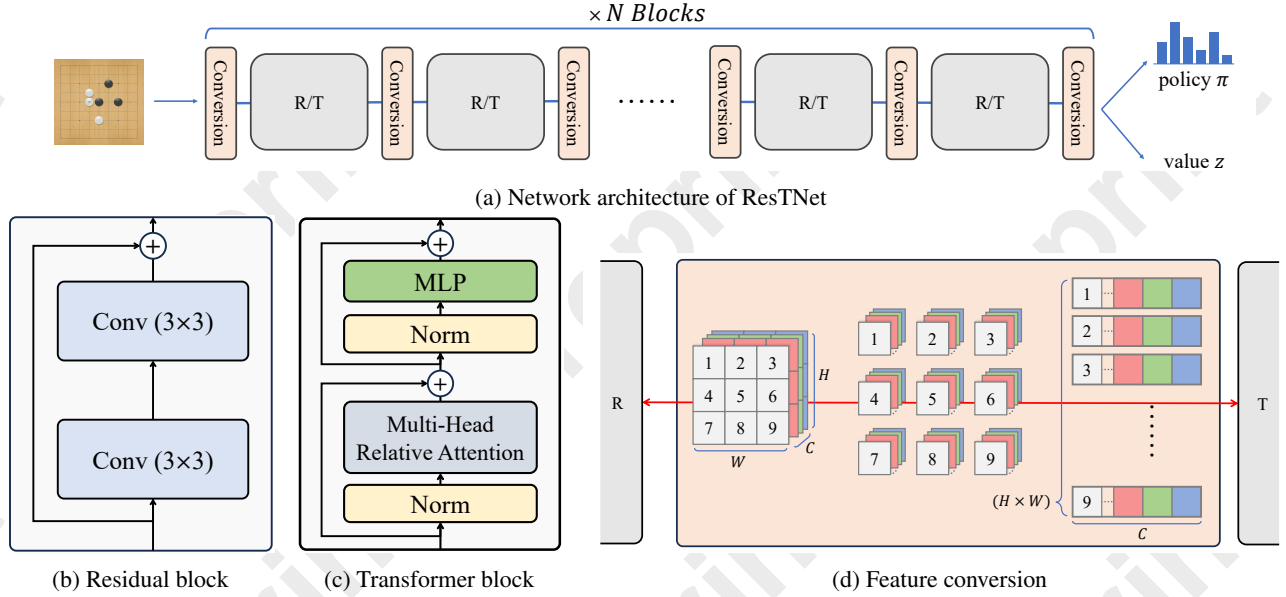


Figure 2: ResTNet consists of a sequence of blocks, each of which is either a residual block (R) or a Transformer block (T). The feature conversion transfers the features between R and T. Subfigure (d) illustrates conversion from R to T. The one from T to R simply left-right mirrors the network. The ones from R to R and T to T use an identical mapping, i.e., no conversion.

their inference time and the number of parameters. These networks can be further categorized into four types. First, the AlphaZero-like networks, comprising solely of convolutional neural networks, i.e., 6R. Second, the ViT-like networks, consisting purely Transformer-based networks, i.e., 6T. Third, the CoAtNet-like networks begin with a series of residual blocks followed by Transformer blocks, such as 5R1T, 4R2T, etc. Finally, the networks with interleavings of residual and Transformer blocks, such as RTRRRT, RRTRRT, etc. Considering the inference time, we only explore permutations that include two Transformer blocks and four residual blocks, with one Transformer block at the end to facilitate global information extraction¹. Note that these four categories are all encompassed within our proposed ResTNet.

Unlike in computer vision, where accuracy is commonly used as a benchmark to measure the performance of networks, our evaluation focuses on the playing strength. Specifically, we train each network using the Gumbel AlphaZero algorithm [Danihelka *et al.*, 2022] with 64 simulations based on an open-sourced AlphaZero framework [Wu *et al.*, 2025]. Each training generates a total of 1 million self-play games and includes 100,000 network optimization steps, requiring approximately 200 1080Ti GPU hours. For the evaluation, we compare our models against four KataGo models² as reference opponents across a total of 2,000 games.

From Table 1, models with an equal or greater number of Transformer blocks than residual blocks, i.e., 6T, 1R5T,

¹We also try placing R at the end, but it leads to worse results, similar to the finding in computer vision.

²These models are katal-b6c96-s115648256, katal-b10c128-s41138688, katal-b10c128-s108710656, and katal-b15c192-s798345984.

	Models	Time	Parameters	Win Rate
Conv only	6R	3.067	7.146	54.60% \pm 2.15%
ViT-like	6T	8.130	3.358	39.85% \pm 2.18%
CoAtNet-like	5R1T	3.650	6.619	56.00% \pm 2.18%
	4R2T	4.237	5.967	51.75% \pm 2.19%
	3R3T	5.000	5.314	47.85% \pm 2.19%
	2R4T	5.882	4.662	37.40% \pm 2.12%
	1R5T	6.329	4.001	31.15% \pm 2.03%
	TRRRRT	4.566	5.967	43.90% \pm 2.18%
	RTRRRRT	4.566	5.967	54.35% \pm 2.18%
	RRTRRRRT	4.566	5.967	60.80% \pm 2.14%
	RRRTTRRT	4.566	5.967	49.10% \pm 2.19%

Table 1: Various 6-block ResTNet model configurations, including inference time (in milliseconds), the number of parameters (in millions), and the win rate against KataGo under 2 seconds thinking time, along with the 95% confidence interval.

2R4T, and 3R3T, obtain a win rate of under 50% against KataGo. This is likely due to the lack of convolutional layers which are crucial for capturing local patterns in board games. For CoAtNet-like models, as the number of consecutive Transformer blocks increases, we observe a corresponding decrease in strength. This result suggests that using a long sequence of Transformer blocks does not necessarily improve performance. Surprisingly, this contradicts findings from CoAtNet in computer vision, where using consecutive Transformer blocks is suggested to yield better results. For networks that interleave residual and Transformer blocks, starting with Transformer blocks, i.e., TRRRRT performs the worst. This indicates the importance of initially extracting local patterns before processing global patterns.

	19x19 Go	19x19 Hex
10R	53.60% \pm 2.19%	50.40% \pm 4.39%
R3 (RRT)	60.90% \pm 2.14%	58.00% \pm 4.33%

Table 2: The playing performance of 10R and R3 (RRT) in 19x19 Go and 19x19 Hex respectively. The results include 95% confidence intervals.

On the other hand, RRT achieves the highest win rate of 60.80% against KataGo among all models. We conjecture that this repeating pattern of RRT effectively balances the extraction of global information while simultaneously preserving local patterns.

Performance on 19x19 Go and 19x19 Hex

We further evaluate the playing performance on 19x19 Go and 19x19 Hex based on the experiments from the previous subsection. We select two 10-block network models: 10R and R3 (RRT). 10R serves as an AlphaZero-like baseline model, while R3 (RRT), also denoted as RRRTRRT, begins with R and follows a repeating sequence of three RRT.

For 19x19 Go, to reduce the computational costs of training models from scratch, we trained these models using supervised learning on a human game collection. The collection contains a total of 1 million games played by 7 dan to 9 dan human Go players on Tygem [Cho and Corporation, 2001], a popular online Go platform. After training both models to a total of 150,000 optimization steps, we evaluate their performance against four KataGo models³ under the same thinking time of 5 seconds per move across a total of 2,000 games. For 19x19 Hex, we train two 10-block models directly using the Gumbel AlphaZero algorithm with 32 simulations. Each model trains with 500,000 self-play games and 100,000 optimization steps. Then, we evaluate the playing performance of the models by playing 1,000 games against MoHex [Arneson *et al.*, 2010], a well-known Hex program that won championships in computer Olympiads.

The results in Table 2 show that R3 (RRT) outperforms 10R in both 19x19 Go and 19x19 Hex. In summary, these experiments demonstrate that the discovered architecture, repeating RRT, efficiently integrates Transformers with residual blocks and improves playing performance in board games.

4.2 Global Information Ability of ResTNet

This subsection investigates the ability of the two trained 19x19 Go models, 10R and R3 (RRT), as described in the previous subsection, to process global information by evaluating their performance on two well-known challenges in Go: recognizing *circular patterns* and *ladder patterns*.

Recognition of Circular Patterns

Circular patterns refer to long sequences of blocks that form a circular enclosure around the opponent’s stones, usually spanning a wide area of the Go board, as shown in the top-right corner of Figure 1a. These patterns present a specific

³These models are kata1-b10c128-s41138688, kata1-b10c128-s108710656, kata1-b10c128-s46989824, and kata1-b10c128-s56992512.

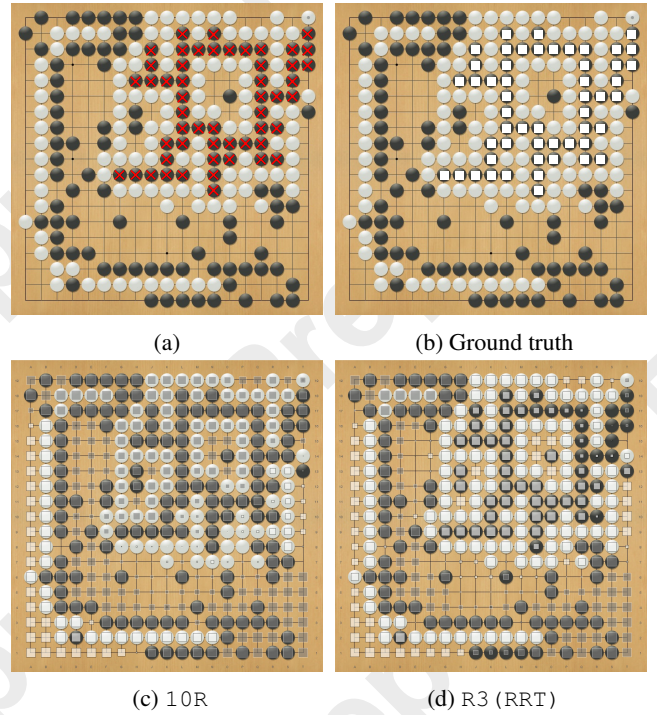


Figure 3: The circular pattern, ground truth results, and board evaluation results for 10R and R3 (RRT). The square in (c) and (d) indicates the ownership predicted by each model, with larger sizes representing higher confidence in ownership by black or white.

challenge for Go programs that require effective global information processing. Wang *et al.* [2023] developed an adversarial Go program called *cyclic-adversary*, which effectively induces circular patterns, causing most Go programs to misjudge their life-and-death status due to an insufficient understanding of global information. In addition, Wang *et al.* [2023] provided a game collection containing 24 games, each featuring circular patterns, played by cyclic-adversary against KataGo.

We first evaluate whether ResTNet can correctly recognize the life-and-death status of circular patterns in the game collection⁴ provided by Wang *et al.* [2023]. Specifically, we incorporate a board evaluation head [Gilmer, 2016; Wu *et al.*, 2018] into ResTNet. The board evaluation head, commonly used in current Go programs to assess a model’s understanding of life-and-death status, is designed to predict the ownership of each position on the board at the endgame. The output values are bounded within the range of $[-1, 1]$, where 1 represents black ownership and -1 represents white ownership. Figure 3a shows an example of circular pattern games from the game collection. In this game, the cyclic-adversary, playing as White, successfully establishes a circular pattern and deceives KataGo, resulting in the capture of the marked black stones. The ownership of these marked stones should be white, as shown in 3b. Interestingly, significant differences are observed between the evaluations of 10R

⁴These games are available at <https://goattack.far.ai/adversarial-policy-katago>.

	Circular Patterns		Ladder Patterns
	MSE	Being Attacked Rate	Accuracy
10R	2.58 ± 0.59	$70.44\% \pm 3.34\%$	$59.15\% \pm 0.24\%$
R3 (RRT)	1.07 ± 0.37	$23.91\% \pm 3.13\%$	$80.01\% \pm 0.19\%$

Table 3: Experiments for ResTNet’s global information ability on long-sequence patterns. For circular patterns, the MSE measures the error in recognizing the life-and-death status of circular patterns, while the being attacked rate represents the probability of being attacked by the cyclic-adversary. Lower values are better for both metrics. For ladder patterns, the accuracy indicates the model’s ability to correctly predict ladder pattern outcomes. All results include 95% confidence intervals.

and R3 (RRT), as depicted in Figure 3c and 3d. The 10R incorrectly predicts the marked stones as belonging to the black player. In contrast, the ownership predicted by R3 (RRT) accurately aligns with the ground truth, demonstrating robustness in recognizing circular patterns. We evaluate all games and quantify model performance using the mean square error (MSE) between the ground truth and board evaluation output, as summarized in Table 3. The results show that R3 (RRT) achieves a significantly lower MSE compared to 10R, indicating a more accurate understanding of the life-and-death status for circular patterns.

Moreover, since R3 (RRT) demonstrates a better understanding of circular patterns, it is worth investigating whether R3 (RRT) can defend against the cyclic-adversary. Specifically, both R3 (RRT) and 10R play against the cyclic-adversary using the same openings from Wang *et al.* [2023]’s game collection, with each opening beginning with a circular pattern. The model must recognize circular patterns and play a correct sequence of moves to avoid being captured and win the game; otherwise, it will lose. Given the inherent randomness of the cyclic-adversary, each opening is played by 30 games, resulting in a total of 720 evaluation games. Table 3 shows the probability of each model being attacked by the cyclic-adversary. Compared to 10R, R3 (RRT) significantly reduces the being attacked rate from 70.44% to only 23.91%, a reduction by a factor of 2.95. Overall, the results demonstrate that using ResTNet enhances the ability to leverage global knowledge for processing long sequences, addressing challenging problems that most Go programs struggle to handle.

Recognition of Ladder Patterns

We investigate whether ResTNet can effectively address another long sequence challenge – recognizing the *ladder pattern* in 19x19 Go. Unlike circular patterns, which can be generated using cyclic-adversary, there are no established strategies to induce ladder patterns. Therefore, we collected a total of 1,655,000 ladder patterns from the Tygem game collection. Figure 4a and 4c show two examples of ladder patterns. The stones marked with triangles indicate the player attempting to escape, serving as the defender, while the opponent is the attacker, trying to capture the marked stone. Note that the ladder dataset includes an equal number of escape successes and escape failures to ensure fairness.

Next, we use model probing [Alain and Bengio, 2017] to

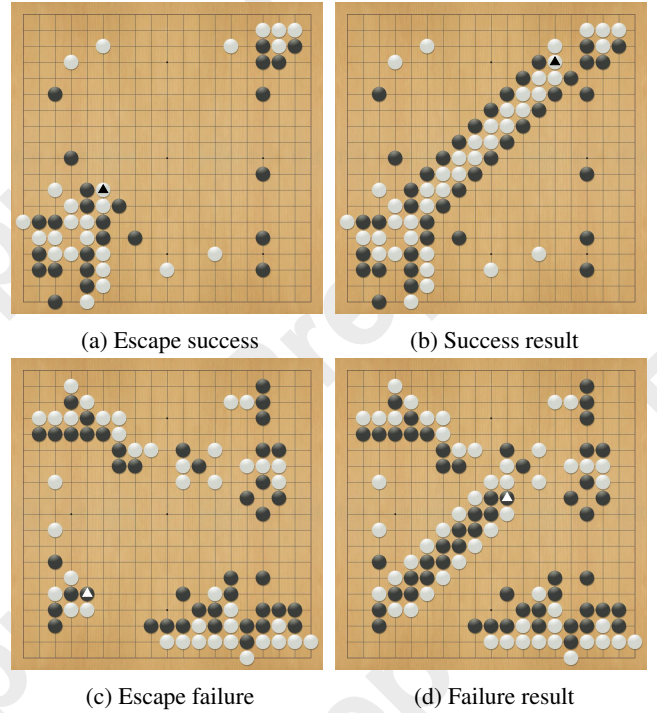


Figure 4: Two ladder patterns from the ladder dataset. (a) and (c) are the original boards, while (b) and (d) show the respective solutions.

examine whether the trained models capture the ladder patterns. Specifically, we incorporate a *ladder head* into the network output, freezing the backbone parameters while allowing only the ladder head to be trained. The ladder head aims to predict whether the defender can escape: 1 for success and -1 for failure, as shown in Figure 4b and 4d. This provides a straightforward approach to determine whether the backbone has learned ladder-related information, as the ladder head cannot recognize ladder patterns without this information. During the evaluation, for each ladder pattern, the model predicts escape success if the output of the ladder head is greater than 0.5, and escape failure if the output is less than -0.5. Values within the range of $(-0.5, 0.5)$ are classified as unknown and are considered incorrect predictions.

We train the ladder head on both 10R and R3 (RRT). After training, we evaluate these models on a separate testing ladder dataset consisting of 166,500 ladder patterns. Table 3 shows the accuracy of 10R and R3 (RRT) in identifying ladder patterns, where R3 (RRT) demonstrates a significant improvement, increasing accuracy from 59.15% to 80.01%. This aligns with the findings from ELF OpenGo [Tian *et al.*, 2019], which indicate that AlphaZero-like Go programs struggle to fully recognize ladder patterns. In contrast, the inclusion of Transformer blocks significantly enhances the model’s ability to process long sequences and recognize ladder patterns.

4.3 Visualization of ResTNet

We visualize ResTNet by constructing attention maps using the attention values from the Transformer block, where each token corresponds to an attention map, and the values rep-

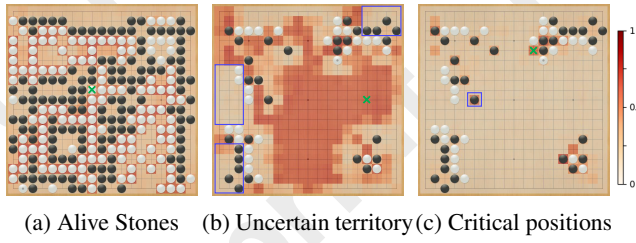


Figure 5: Since different blocks and heads may capture different types of knowledge, the attention values shown corresponding to the position marked in green, taken from the fourth head of the first T block for (a), the fourth head of the second T block for (b), and the third head of the second T block for (c). All positions are Black’s turn to move. We normalize the values to $[0, 1]$ for better visualization, with redder colors indicating higher values.

represent the relative importance of other tokens. By analyzing these attention maps, we can explore patterns or strategies utilized by ResTNet, offering insights into its behavior and decision-making process. This analysis is conducted on both 19x19 Go and 19x19 Hex using R3 (RRT), the same models used in subsection 4.1.

Attention Maps in 19x19 Go

We select two 19x19 Go games, including a circular pattern and a ladder pattern. Figure 5 illustrates the attention maps for the position marked in green, with redder colors indicating higher levels of relative importance. Interestingly, these attention maps correspond closely to Go knowledge concepts. First, the attention map in Figure 5a focuses exclusively on the white stones that remain alive until the end game, aligning with the life-and-death concept in Go. Notably, these stones are spread across the entire board, making them extremely challenging for convolutional networks to recognize effectively. Second, the attention map in Figure 5b highlights uncertain territory. The center area, not surrounded by any player, shows high attention values, while the three areas marked by blue rectangles, surrounded by one player, show low attention values. Finally, the attention map in Figure 5c highlights important positions. The three black stones can escape by following a sequence of moves starting with the green-marked position and connecting to the middle left black stone marked by the blue square. Interestingly, the attention map focuses on three black stones as well as the middle left black stone, indicating that it captures the concept of the ladder pattern.

Attention Maps in 19x19 Hex

Next, we examine the attention maps in 19x19 Hex, as shown in Figure 6. Similar to 19x19 Go, the attention maps in Hex show essential game concepts, particularly the fundamental strategy of connection. The attention map in Figure 6a highlights the concept of basic connection, focusing only on the black stones necessary to secure a win for Black, while irrelevant black stones, marked by blue hexagons, are excluded. Identifying connection stones across the entire board from left to right also demonstrates that ResTNet effectively captures global information. Moreover, in Figure 6b, the attention map shows a critical and advanced strategy in Hex,

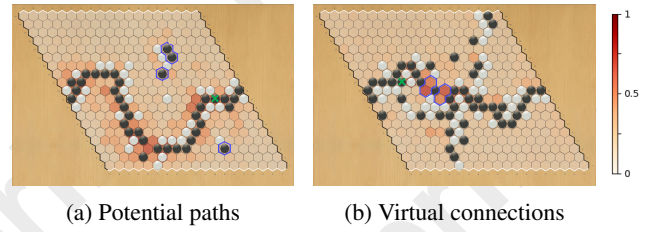


Figure 6: The attention maps for 19x19 Hex from the third head of the second and the third T block for (a) and (b), respectively. The next moves of (a) and (b) are for White and Black, respectively.

known as *virtual connections* [Hayward *et al.*, 2004], which refer to sets of positions that enable one player to connect two specified sets of stones, even if the opponent plays first. In conclusion, the attention maps in both Go and Hex demonstrate that ResTNet effectively learns global knowledge concepts while also offering opportunities for interpretability and explainability in board games.

5 Discussion

This paper proposes *ResTNet*, a network architecture specifically designed for AlphaZero algorithms in board games, integrating residual and Transformer blocks to bridge local and global knowledge. Our experiments show that repeating RRT blocks achieves the best performance, effectively balancing inference time and model parameters. ResTNet not only improves playing strength in both Go and Hex games, but also significantly addresses two challenging long-sequence patterns in 19x19 Go. For circular patterns, ResTNet reduces the mean square error from 2.58 to 1.07 and lowers the average probability of being attacked against cyclic-adversary from 70.44% to 23.91%, significantly reducing a factor of 2.95. For ladder patterns, ResTNet improves the accuracy of correctly predicting the outcome of ladder patterns from 59.15% to 80.01%. Furthermore, visualizing attention maps reveals that ResTNet captures game-specific strategies, further contributing to the interpretability of AlphaZero.

Due to computational constraints, this paper mainly explores ResTNet with repeating RRT in the main experiments. However, even without a large-scale network architecture search, RRT still achieves superior performance. Future work can build on this direction using larger-scale architecture search to further optimize ResTNet. Moreover, as ResTNet is game-independent, it can be seamlessly applied to other board games, such as Chess or Shogi, without requiring modifications. Future research could explore its robustness by designing other adversarial strategies to further improve its capabilities. While current board game programs have generally achieved superhuman performance, understanding their decision-making processes remains a critical challenge. ResTNet’s attention maps open promising research directions in explainable AI (XAI) for board games. In conclusion, we believe the proposed ResTNet and our empirical findings make a valuable contribution to the board game community, offering a strong foundation for future research in this field.

Acknowledgments

This research is partially supported by the National Science and Technology Council (NSTC) of the Republic of China (Taiwan) under Grant Number NSTC 113-2221-E-001-009-MY3, NSTC 113-2634-F-A49-004, and NSTC 113-2221-E-A49-127. The authors would also like to thank anonymous reviewers for their valuable comments.

References

- [Alain and Bengio, 2017] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations*, February 2017.
- [Arneson et al., 2010] Broderick Arneson, Ryan B. Hayward, and Philip Henderson. Monte Carlo Tree Search in Hex. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(4):251–258, December 2010.
- [Browne et al., 2012] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavenor, Diego Perez, Spyridon Samothrakis, and Simon Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, March 2012.
- [Cho and Corporation, 2001] Hun-hyun Cho and TYGEM Corporation. TYGEMGO. <http://www.tygemgo.com/>, 2001. Accessed: 2025-01.
- [Ciolino et al., 2020] Matthew Ciolino, Josh Kalin, and David Noever. The Go Transformer: Natural Language Modeling for Game Play. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, pages 23–26, September 2020.
- [Coulom, 2007] Rémi Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In *Computers and Games, Lecture Notes in Computer Science*, pages 72–83, Berlin, Heidelberg, 2007. Springer.
- [Czech et al., 2023] Johannes Czech, Jannis Blüml, and Kristian Kersting. Representation Matters: The Game of Chess Poses a Challenge to Vision Transformers, April 2023.
- [Dai et al., 2021] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. CoAtNet: Marrying Convolution and Attention for All Data Sizes. In *Advances in Neural Information Processing Systems*, volume 34, pages 3965–3977. Curran Associates, Inc., 2021.
- [Danihelka et al., 2022] Ivo Danihelka, Arthur Guez, Julian Schrittwieser, and David Silver. Policy improvement by planning with Gumbel. In *International Conference on Learning Representations*, April 2022.
- [Devlin et al., 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [Dosovitskiy et al., 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, S. Gelly, Jakob Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, October 2020.
- [Feng et al., 2024] Xidong Feng, Yicheng Luo, Ziyang Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. ChessGPT: Bridging Policy Learning and Language Modeling. *Advances in Neural Information Processing Systems*, 36, February 2024.
- [Gao et al., 2018] Chao Gao, Martin Müller, and Ryan Hayward. Three-Head Neural Network Architecture for Monte Carlo Tree Search. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 3762–3768, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization.
- [Gilmer, 2016] Justin Gilmer. Jmgilmer/GoCNN, 2016.
- [Guo et al., 2022] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. CMT: Convolutional Neural Networks Meet Vision Transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12165–12175, June 2022.
- [Hayward et al., 2004] R. Hayward, Y. Björnsson, M. Johansson, M. Kan, N. Po, and J. van Rijswijk. Solving 7x7 Hex: Virtual Connections and Game-State Reduction. In *Advances in Computer Games: Many Games, Many Challenges*, IFIP — The International Federation for Information Processing, pages 261–278, Boston, MA, 2004. Springer US.
- [Li et al., 2022a] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-ViT: Next Generation Vision Transformer for Efficient Deployment in Realistic Industrial Scenarios, August 2022.
- [Li et al., 2022b] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. EfficientFormer: Vision Transformers at MobileNet Speed. *36th Conference on Neural Information Processing Systems*, 2022.
- [Liang et al., 2023] Wen Liang, Chao Yu, Brian Whiteaker, Inyoung Huh, Hua Shao, and Youzhi Liang. AlphaZero Gomoku, September 2023.
- [Monroe and Chalmers, 2024] Daniel Monroe and Philip A. Chalmers. Mastering Chess with a Transformer Model, October 2024.

- [Romstad *et al.*, 2023] Tord Romstad, Marco Costalba, Joona Kiiski, Gary Linscott, and Stockfish Contributors. Stockfish, 2023.
- [Ruoss *et al.*, 2024] Anian Ruoss, Gregoire Deletang, Sourabh Medapati, Jordi Grau-Moya, Li Kevin Wenliang, Elliot Catt, John Reid, Cannada A. Lewis, Joel Veness, and Tim Genewein. Amortized Planning with Large-Scale Transformers: A Case Study on Chess. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.
- [Sagri *et al.*, 2023] Amani Sagri, Tristan Cazenave, Jérôme Arjonilla, and Abdallah Saffidine. Vision Transformers for Computer Go, September 2023.
- [Sandler *et al.*, 2018] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, June 2018.
- [Schultz *et al.*, 2024] John Schultz, Jakub Adamek, Matej Jusup, Marc Lanctot, Michael Kaisers, Sarah Perrin, Daniel Hennes, Jeremy Shar, Cannada Lewis, Anian Ruoss, Tom Zahavy, Petar Veličković, Laurel Prince, Satinder Singh, Eric Malmi, and Nenad Tomašev. Mastering Board Games by External and Internal Planning with Language Models, December 2024.
- [Shaw *et al.*, 2018] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [Silver *et al.*, 2018] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018.
- [Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc Le. Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*, 4, September 2014.
- [Tian *et al.*, 2019] Yuandong Tian, Jerry Ma, Qucheng Gong, Shubho Sengupta, Zhuoyuan Chen, James Pinkerton, and Larry Zitnick. ELF OpenGo: An analysis and open reimplementation of AlphaZero. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6244–6253. PMLR, May 2019.
- [Tseng *et al.*, 2024] Tom Tseng, Euan McLean, Kellin Pelrine, Tony Tong Wang, and Adam Gleave. Can Go AIs be adversarially robust? In *ICML 2024 Next Generation of AI Safety Workshop*, July 2024.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, pages 6000–6010, Red Hook, NY, USA, December 2017. Curran Associates Inc.
- [Wang *et al.*, 2021] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–558, October 2021.
- [Wang *et al.*, 2023] Tony Tong Wang, Adam Gleave, Tom Tseng, Kellin Pelrine, Nora Belrose, Joseph Miller, Michael D. Dennis, Yawen Duan, Viktor Pogrebnjak, Sergey Levine, and Stuart Russell. Adversarial Policies Beat Superhuman Go AIs. In *Proceedings of the 40th International Conference on Machine Learning*, pages 35655–35739. PMLR, July 2023.
- [Wu *et al.*, 2018] Ti-Rong Wu, I-Chen Wu, Guan-Wun Chen, Ting-Han Wei, Hung-Chun Wu, Tung-Yi Lai, and Li-Cheng Lan. Multilabeled Value Networks for Computer Go. *IEEE Transactions on Games*, 10(4):378–389, December 2018.
- [Wu *et al.*, 2021] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing Convolutions to Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22–31, October 2021.
- [Wu *et al.*, 2025] Ti-Rong Wu, Hung Guei, Pei-Chiun Peng, Po-Wei Huang, Ting Han Wei, Chung-Chin Shih, and Yun-Jui Tsai. Minizero: Comparative analysis of alphazero and muzero on go, othello, and atari games. *IEEE Transactions on Games*, 17(1):125–137, 2025.
- [Wu, 2020] David J. Wu. Accelerating Self-Play Learning in Go. In *Proceedings of the AAAI Workshop on Reinforcement Learning in Games*, November 2020.
- [Xie *et al.*, 2018] Zheng Xie, XingYu Fu, and JinYuan Yu. AlphaGomoku: An AlphaGo-based Gomoku Artificial Intelligence using Curriculum Learning, September 2018.