# AI-FML Agent with Patch Learning Machanism for Robotic Game of Go Application

Yi-Lin Tsai
Dept. of Computer Science and Information Engineering
National University of Tainan
Tainan, Tawain
m10759011@gm2.nutn.edu.tw

Chang-Shing Lee
Dept. of Computer Science and Information Engineering
National University of Tainan
Tainan, Tawain
leecs@mail.nutn.edu.tw

***Abstract*—In this paper, we propose an AI-FML agent with patch learning mechanism for robotic game of Go applications. The proposed AI-FML agent contains three kinds of intelligence, including a perceptual intelligence, a cognitive intelligence, and a computational intelligence, for the robotic application. Additionally, we embed the patch learning mechanism into the AI-FML agent. The method for running patch learning involves three steps. First, trains an initial global model, then trains a patch model for each identified patch, and finally updates the global model using the training data that do not fall into any patch. This paper adopts the Google DeepMind Master 60 games to be the training data and testing data. The experimental results show that the AI-FML agent with the patch learning mechanism can improve the performance of regression for robotic game of Go applications.**

***Keywords—AI-FML, Agent, Patch Learning, Fuzzy Machine Learning, Game of Go***

## I. Introduction

Nowadays, machine learning has been used in multiple real-world fields and industries, such as medical diagnosis, image processing, regression etc. However, training a high performance machine learning model is usually an iterative process which relies on experience and trial-and-error. Sometimes, we need to take some remedies to enhance its performance when it is dissatisfied [1]. Some strategies are implemented to enhance the performance on machine learning mechanisms, such as using a single deeper model, using a single broader model, connecting multiple simple base learners in parallel, or connecting multiple simple weak learners in series [1, 2]. In this paper, we propose an AI-FML agent with the patch learning (PL) mechanism to improve the performance of the machine learning [1, 2]. The idea of PL introduced by Mendel is as follows: *Consider a sculptor who is sculpting a human figure, after his first pass at this, the sculptor examines the entire figure and notices that improvements need to be made to certain parts of the figure. He zooms into the certain parts that need more work, after which he blends in the refined portions of the figure with the rest of the figure. He continues such iterative refinements until he is satisfied with the entire figure. Each patch in PL is analogous to a part in the figure that needs more work* [1]. According to [1], PL consists of three steps described as follows: (1) Train an initial global model using all training data first. (2) Identify patches from the initial global model and train a patch model for each patch. (3) Update the global model using training data [1]. Wu and Jerry [1] defined a patch as a connected polyhedron in the input domain. For example, a patch in a 1D input domain is an interval, and a patch in a 2D input domain can be a rectangle, an ellipse, and so on. However, generally identifying the patch locations is a very challenging task, and not every problem can be easily visualized. The PL connects multiple simple models both in parallel and in series to improve the learning performance. Mendel [2] introduced how PL can be performed using rule-based fuzzy systems, because it is easy to initialize patch candidates in a fuzzy system.

Regression, ensemble methods, and deep learning are important machine learning methods for data scientists [9]. An adaptive network-based fuzzy inference system (ANFIS) was proposed by Jang [12] in 1993. The ANFIS integrates both machine learning model and fuzzy logic principles and it has a potential to capture the benefits of both in real-world applications. Its inference system corresponds to a set of fuzzy rules that have learning capability to approximate nonlinear functions and is considered to be a universal estimator [12]. In our previous paper [3], we used AlphaGo Master sixty games as the experimental dataset to make the win rates predicted by the Darkforest AI bot closer to those predicted by the ELF Open Go AI bot based on FML-based genetic learning (GFML), XGBoost learning, and a seven-layered deep fuzzy neural network (DFNN) learning. In this paper, we further combine FML, PL, and adaptive network-based fuzzy inference system (ANFIS) with the deep learning to train a model to predict the win rates of the AlphaGo Master sixty games. The experimental results show that introducing PL mechanism has a better performance than the methods presented in our previous paper [3].

The remainder of this paper is as follows: Section II presents the structure of ANFIS-based patching learning mechanism. Section III describes the patch learning mechanism for game of Go data set. Section IV proposes the PL-based AI-FML agent for robotic application on predicting the win rate of Go game. Section V shows the experimental results. Finally, section VI draws the conclusions.

## II. Structure of ANFIS-Based Patch Learning Mechanism

### A. Patch Learning Mechanism

在傳統 ANFIS 的設計階段，使用訓練資料來優化輸入的隸屬函數與輸入參數，並使用所有訓練資料來優化評估指標[1、11]。基於文獻[1]中補丁學習的概念，本論文提

出了一種具有補丁學習機制的 AI-FML 代理人，從全域的模糊規則系統設計開始，然後找到評估指標最大的區域作為補丁。圖 1 顯示了用於預測 Master 60 盤棋局勝率的補丁學習架構，其描述如下：(1)我們使用所有訓練資料來訓練全域模型。(2)我們找出第 1 手(M1)到 P1 手($M_{P1}$)的輸入區域會引起較大的損失，因此我們使用屬於該區間的訓練資料來訓練一個補丁模型($PM_1$)以減少整體學習損失。接著我們採用第 P1+1 ($M_{P1+1}$)手到第 P2 ($M_{P2}$)手, …,第 N ($M_N$)手到第 PN ($M_{PN}$)手落於區間內的資料來訓練補丁模型 1 ($PM_1$), 補丁模型 2 ($PM_2$), …, 補丁模型 L ($PM_L$) (3)最後再使用補丁 1, 2, …, L 未使用的剩餘訓練資料更新全域的模糊規則系統。圖 1 以第 2 盤棋局為例，假設 L = 3，$MP_1$ = 20，$MP_2$ = 40 和 $MP_3$ = 90。

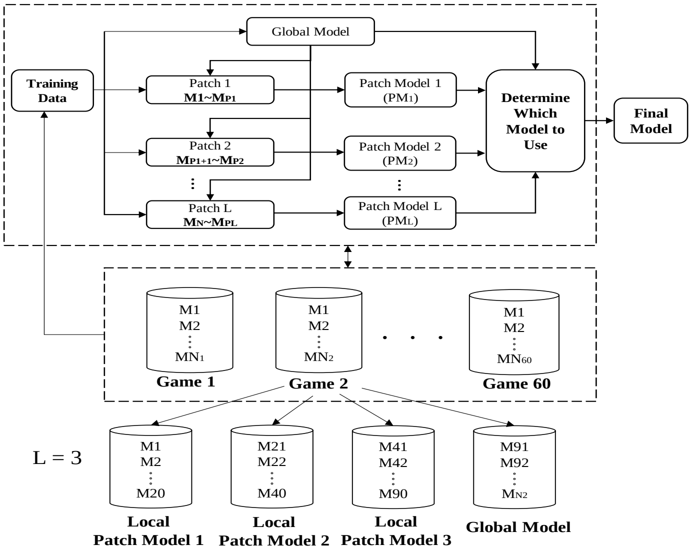


Fig. 1. Structure of patch learning for predicting the gane of Go Data set.

### B. *Adaptive Network-based Fuzzy Inference System (ANFIS) Mechanism*

基於 ANFIS 的補丁學習機制已嵌入到 AI-FML 代理人中，我們將其應用於預測圍棋資料的勝率。另外採用 ANFIS 算法訓練補丁模型以提高性能指標[1，11]。ANFIS 是基於 Takagi–Sugeno 模糊推論系統的人工神經網絡模型之一，並於 1990 年初被提出[10，11]。在本論文中，我們使用 PyTorch 來實作基於 ANFIS 的補丁學習模型框架。ANFIS 的架構包括前鑑部和後鑑部兩部分，其架構由五層網路所組成，如圖 2[11]。其描述如下: (1)第一層/模糊化層：我們使用隸屬函數來計算輸入值所對應的歸屬值。 (2)第二層/規則層: 負責通過乘積為輸入信號來生成規則的觸發強度。(3)第三層/正規化層：正規化觸發強度，將每個值除以所有的觸發強度總和。(4)第四層/推論層: 將第三層的正規化值與後鑑部參數相乘，並將結果發送到第五層。 (5)第五層/解模糊化層: 將所有輸出信號的總和計算，並生成最終輸出。

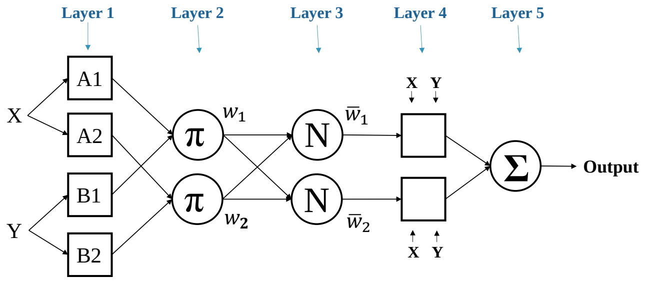


Fig. 2. *Structure of ANFIS with five layers[11].*

## III. ANFIS-BASED PATCH LEARNING FOR GO GAME DATA SET

### A. *Introduction to Data Set from IEEE WCCI 2020*

在本論文中，我們採用 2016 年 12 月和 1 月的 AlphaGo Master 60 盤棋局作為實驗數據，這些資料被指定為 IEEE CIS 旗艦會議上所舉行的競賽數據[3]。參賽者可以從 60 場比賽中選擇 40 場比賽作為訓練資料，將其餘 20 場比賽作為測試資料。並建構模糊推論系統的知識庫和規則庫。表 1 為第 1 盤棋局的第 1 步到第 20 步的資料。每個棋局都包含 Darkforest AI Bot 和 EFL OpenGo AI Bot 預測的數據。MoveNo 是手數編號，但 MoveNo 僅列出「奇數」編號(即 1、3、5 ...)，因為每一行對應一對黑棋與白棋。因此 MoveNo 1 的行對應於黑色的第 1 手(即 B1)和白色的第 1 手(即 W2)。MoveNo 145 對應於黑色第 145 手(即 B145)和白色第 146 手(即 W146)。如果最後的 MoveNo 為「奇數」，則最後一行的白色資訊將為空。訓練資料從第 1 局到第 40 局共有 3758 筆數據。測試資料從第 41 局到第 60 局共有 1880 筆數據。

TABLE I. MOVE 1 TO MOVE 20 OF GAME 1.

| Move No | DBSN | DWSN | DBWR | DWWR | DBTMR | DWTMR | EBWR | EWWR |
|---|---|---|---|---|---|---|---|---|
| 1 | 3863 | 2274 | 0.52 | 0.48 | 0 | 1 | 0.49 | 0.50 |
| 3 | 9283 | 7866 | 0.51 | 0.48 | 0.5 | 1 | 0.45 | 0.54 |
| 5 | 11395 | 6798 | 0.51 | 0.47 | 0.66 | 1 | 0.45 | 0.57 |
| 7 | 4499 | 10703 | 0.51 | 0.46 | 0.75 | 1 | 0.49 | 0.52 |
| 9 | 7388 | 20017 | 0.52 | 0.46 | 0.8 | 1 | 0.47 | 0.53 |
| 11 | 20098 | 9693 | 0.53 | 0.46 | 0.83 | 1 | 0.47 | 0.52 |
| 13 | 20017 | 14595 | 0.53 | 0.47 | 0.85 | 1 | 0.48 | 0.49 |
| 15 | 6786 | 4892 | 0.52 | 0.45 | 0.87 | 0.875 | 0.48 | 0.53 |
| 17 | 20017 | 6432 | 0.53 | 0.46 | 0.88 | 0.7778 | 0.48 | 0.53 |
| 19 | 5267 | 13267 | 0.51 | 0.48 | 0.9 | 0.8 | 0.48 | 0.50 |

### B. *Pre-processing Partition Function for Game Data Set*

Jang [11]在 1993 年論文中提到：「模糊條件語句是 IF A THEN B 形式的表達式，其中 A 和 B 是具有適當的隸屬函數的模糊集的標籤」和「fuzzy if-then 規則通常用於記錄了不精確的推論模式，這些模式對人類在不確定和不精確的環境中做出決策扮演著重要的角色」。

有多種用於劃分輸入域的分區函數，例如，明確分區函數或 Type-1 分區函數[2]。實值變數的明確分區由多個不重疊的相鄰區域所組成，這些區域是實數的間隔，其中每個區域中的隸屬程度為 1，在該區域之外為 0[2]。此外

實值變數的一階不確定性分區由多個重疊區間組成，其中我們可以確定重疊的開始和結束位置，因此每個重疊中的歸屬程度是介於 0 到 1 之間。一階不確定性分區被視為與其最近鄰域重疊的非矩形型 Type-1 分區函數。如圖 3 本實驗採用 64 個高斯函數的 Type-1 分區，用於圍棋數據預處理。

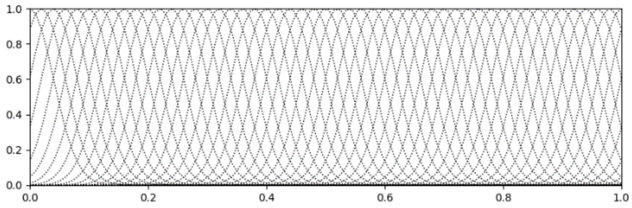
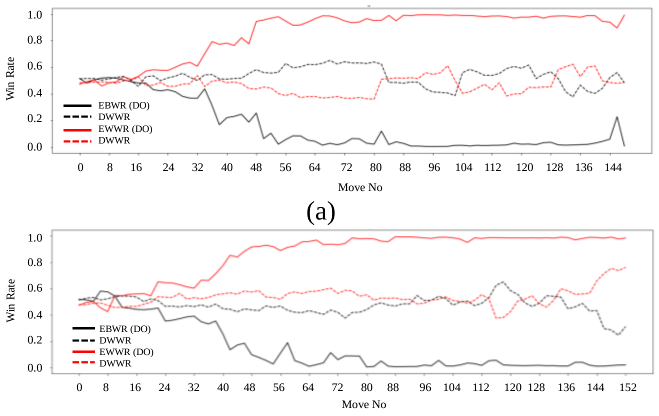


Fig. 3. Type-1 partition functions with 64 Gaussian functions for Go.

此外實驗中有兩個用於資料預處理的標準。第一個標準是「如果 Darkforest 的最終預測結果是錯誤的，則該局的資料集將不會用於訓練集或測試集。」例如我們在訓練集中將不採用第 15 盤、第 25 盤和第 29 盤的資料集。圖 4 顯示了兩盤 Darkforest 與 ElF OpenGo 的預測結果。

第二個標準是「如果 ELF OpenGo 和 Darkforest 之間的白色或黑色預測勝負差異超過 40%，則該局的資料將不會在訓練集或測試集中採用。」假設一盤對局中有 100 手，但 ELF OpenGo 和 Darkforest 預測有 40 個手數不同，則該局將從資料集中刪除。基於此標準，我們將第 8 盤、第 32 盤和第 36 盤從訓練數據集中刪除。在實驗測試集中我們刪除了第 44 盤、第 47 盤、第 56 盤和第 57 盤。 此外，我們還檢查資料集中的缺失值。如果有一局的第 1 手與最後一手都是相同的黑棋或白棋，我們會將該局的最後一筆資料移除。經過檢查後，我們從訓練集中刪除了 19 筆資料，從測試集中刪除了 6 筆資料。最後我們在此實驗中保留了 3143 筆訓練資料，測試集中保留 1239 筆資料。



(a)

(b)

Fig. 4. Predicted win rate curves for Games (a) 15 and (b) 32.

## IV. PL-based AI-FML agent for Robotic Game of Go Application

### A. ANFIS for AI-FML Agent on Game

基於補丁學習的 AI-FML 代理人中的模糊規則核心系統是 ANFIS，它由以下功能所組成：(1)含有數據的知識庫。(2)規則庫。(3)規則推論的運算子。(4)模糊化過程。(5)去模糊化過程[11]。圖 5 顯示了機器人應用於 AI-FML 代理人的 ANFIS 架構，該架構透過以下步驟建立：(1)在前鑑部使用隸屬函數檢索模糊變數的輸入值，以計算每個語意的隸屬程度。(2)結合前鑑部的隸屬程度以獲取每個規則的權重。(3)根據權重產生每個規則的後鑑部。(4)匯總結果以產生最終輸出[ 11]。人類輸入資料，AI-FML 代理人將解模糊後的值輸出給機器人，例如 Kebbi Air，Palro 或 Zenbo，以實現人類和機器人在教室中學習的目標[3]。

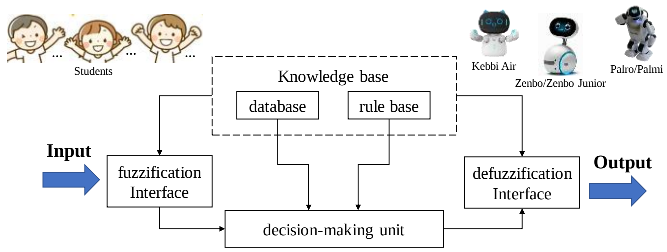


Fig. 5. Structure of ANFIS for AI-FML agent.

在本研究中，我們使用 64 個 Type-1 高斯分區函數對六個輸入模糊變數進行分區，包括 DBSN、DWSN、DBWR、DWWR、DBTMR 和 DWTMR。舉例來說，如果 $x$ 是 AI-FML 代理人補丁學習的輸入值 DBSN，有 64 條規則，如下所示：

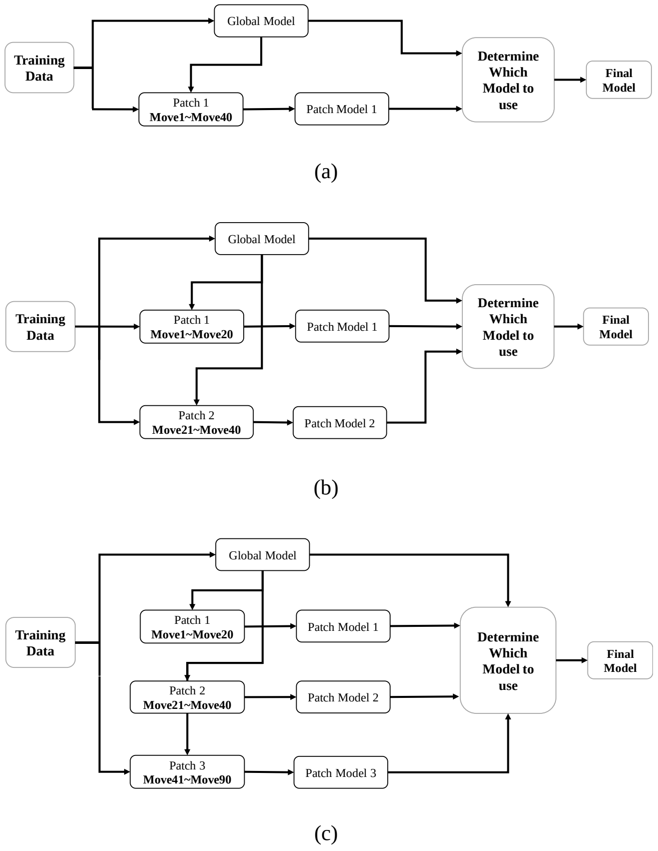
規則 1：IF $x$ 為分區 1，Then $y = y_1(x)$
規則 2：IF $x$ 為分區 2，Then $y = y_2(x)$
⋮
規則 64：IF $x$ 為分區 64，Then $y = y_{64}(x)$

其中 $y_1(x)$，$y_2(x)$，…，$y_{64}(x)$是 $x$ 的不同函數。在分區 P(1|$x$) 中，假設只有規則 1 被觸發，因此模糊系統輸出為 y = y1($x$)。另外在分區 P(2|$x$) 中，如果規則 1 和規則 2 均被觸發，因此模糊系統的輸出是 $y_1(x)$ 和 $y_2(x)$ 在分區 1 和 2 的隸屬程度加權平均值。

### B. PL-based AI-FML Agent for Game of Go Dataset

基於補丁學習的 AI-FML 代理人可以將 1-Patch、2-Patch 和 3-Patch 應用於電腦圍棋資料集。圖 6(a)是 1-Patch 的學習模型，此模型僅有一個 Patch 專門來預測第 1 手到第 40 手。其餘的手數交由全域模型進行預測。圖 6(b)是 2-Patch 的學習模型，此模型經由兩個補丁進行訓練。在此模型中我們將前 40 手切成兩等份，分別為第 1 手到第 20 手，以及第 21 手到第 40 手。其餘的手數交由全域模型進行預測。圖 6(c)是 3-Patch 的學習模型，補丁 1 與 2 分別

為第 1 手到第 20 手，以及第 21 手到第 40 手。補丁 3 為第 41 手到第 90 手。其餘的手數交由全域模型進行預測。



(a)

(b)

(c)

Fig. 6. Structures of Patch Learning. (a) 1-Patch; (b) 2-Patch; (c) 3-patch.

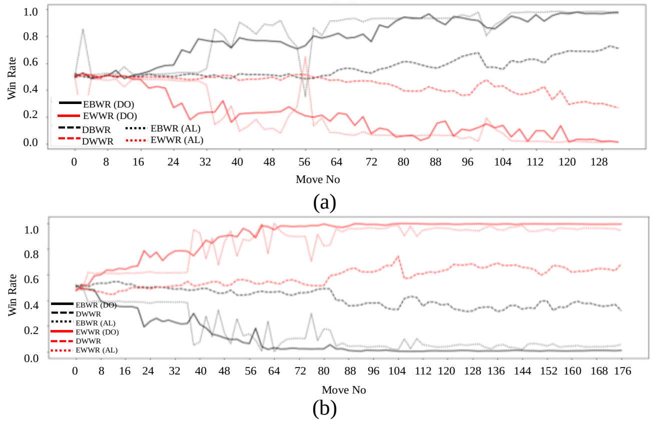
## V. Experimental results

### A. Experiment 1: ANFIS Global Learning for AI-FML Agent

在實驗 1 中，我們採用具有 Adam 優化器的 ANFIS 和學習速率為 0.001 的梯度下降法來更新神經網絡中的所有參數。表 2 說明了在訓練 2000 個迭代後使用平均絕對誤差(MAE)、均方誤差(MSE)和均方根誤差(RMSE)的評估指標數值。這種方法的性能比我們先前論文[3]中使用 Adam 優化器的 DNN 機器學習的方法還來的更好。此外實驗 1 減少了學習參數量和學習時間。圖 7 顯示了實驗 1 中第 39 局和第 58 局的回歸分析。虛線標記的曲線為 EBWR(AL)/EWWR(AL)是實驗 1 的實驗結果。圖中的虛線和實線分別是 Darkforest 和 ELF OpenGo 的預測勝率，黑色和紅色曲線分別代表黑棋和白棋。如圖 7(a)所示，虛線曲線雖然有趨近於 ELF OpenGo 所預測的勝率，但前 5 手和第 56 附近的手數仍具有較大的方差。從圖 7(b)可以發現，除了前 40 手以外，虛線曲線的預測趨勢與實線類似。由於上述這些情況，我們在實驗 2 中實現了補丁學習的概念。

TABLE II. Loss Evaluation based on MAE, MSE, and RMSE

| | Training set | Validate set | Testing set |
|---|---|---|---|
| MAE | 0.0683 | 0.0876 | 0.1197 |
| MSE | 0.0158 | 0.0286 | 0.0438 |
| RMSE | 0.1258 | 0.1692 | 0.2093 |



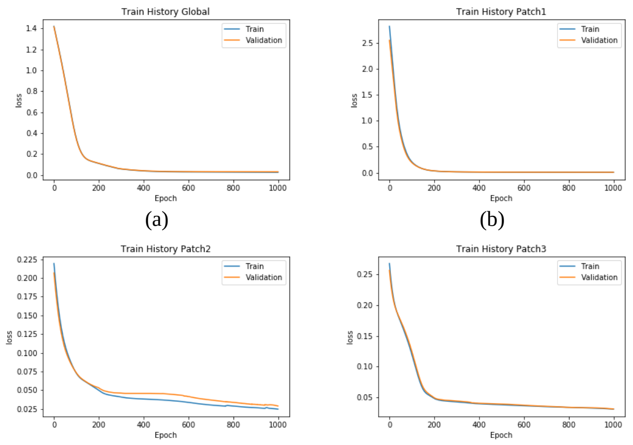
(a)

(b)

Fig. 7. Regression analysis for EXP. 1 in Games (a) 39 and (b) 58.

### B. Experiment 2: PL-based AI-FML Agent with 3-Patches

在實驗 2 中，我們採用具有三個補丁機制的補丁學習 AI-FML 代理人。我們按照以下方式劃分三個補丁：補丁 1 為第 1 手到第 20 手，補丁 2 為第 21 手到第 40 手，以及補丁 3 為第 41 手到第 90 手。首先我們先學習 1000 代的全域模型(IGM)，然後再學習 100 代的補丁 1 (LPM 3.1)、 補丁 2 (LPM 3.2) 和補丁 3 (LPM 3.3)。圖 8 (a)、(b)、(c) 和 (d) 分別顯示了 IGP、LPM3.1、LPM3.2 和 LPM3.3 在訓練期間的訓練集和驗證集的歷史學習曲線。我們也可以觀察到圖 8 (a)、(b)和(d)比圖 8(c)有更好的擬合曲線。也許這是因為第 20 手之後，棋局處於中間階段，處於不確定的情況下，這使得在學習模型時更難適應實際情況。在實驗 2 中，我們設計了兩個子實驗，描述如下：(1)實驗 2.1：我們僅使用經過訓練 1000 代的 IGP 來預測訓練集。 (2)實驗 2.2：我們使用訓練後的 IGM、LPM3.1、LPM3.2 或 LPM3.3 來預測訓練集，並根據每一手所相對應的預測模型。圖 9 顯示了每一手的平均損失曲線。橙色實線和紫色虛線是實驗 2.1 和實驗 2.2 的結果。損失值與實驗 2.1 相比，實驗 2.2 著重在補丁 1、補丁 2 和補丁 3 有下降的趨勢。



(a)

(b)

(c) (d)

Fig. 8. Learning curves for (a) IGM, (b) LPM3.1, (c) LPM3.2, and (d) LPM3.3.

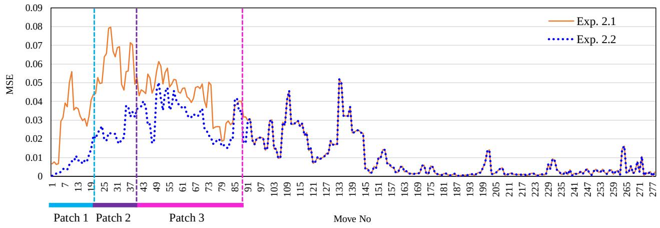


Fig. 9. MSE curves for each move in Exp. 2.

## VI. Conclusions

在本論文中我們展示了具有補丁學習機制的 AI-FML 代理人應用於圍棋機器人。驗實驗中執行了各種補丁學習模型，包括全域模型、一個補丁、兩個補丁與三個補丁的模型。此外補丁學習可以在機器人應用 AI-FML 代理人中被實現。我們採用 Google DeepMind Master 60 盤棋局作為訓練集和測試集。從實驗結果可以發現具有補丁學習機制的 AI-FML 代理人可以提高圍棋機器人在回歸模型上的性能。

## References

[1] D. Wu and J. M. Mendel, “Patch learning,” *IEEE Transactions on Fuzzy Systems*, Jul. 2019. (DOI: 10.1109/TFUZZ.2019.2930022)

[2] J. M. Mendel, “Adaptive variable-structure basis function expansions: Candidates for machine learning,” *Information Science*, vol. 496, pp. 124-149, 2019.

[3] C. S. Lee, Y. L. Tsai, M. H. Wang, W. K. Kuan, Z. H. Ciou, and N. Kubota, “AI-FML agent for robotic game of Go and AIoT real-world co-learning applications,” *2020 World Congress on Computational Intelligence* (*IEEE WCCI 2020*), Glasgow, Scotland, UK, Jul. 19-24, 2020.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[5] C. S. Lee, M. H. Wang, L. W. Ko, Y. Hsiu Lee, H. Ohashi, N. Kubota, Y. Nojima, and S. F. Su, “Human intelligence meets smart machine: a special event at the IEEE International Conference on Systems, Man, and Cybernetics 2018,” *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 1, pp. 23-31, Jan. 2020.

[6] C. S. Lee, M. H. Wang, Y. L. Tsai, L. W. Ko, B. Y. Tsai, P. H. Hung, L. A. Lin, and N. Kubota, “Intelligent agent for real-world applications on robotic edutainment and humanized co-learning,” *Journal of Ambient Intelligence and Humanized Computing*, 2019. (DOI: 10.1007/s12652-019-01454-4).

[7] C. S. Lee, M. H. Wang, L. W. Ko, N. Kubota, L. A. Lin, S. Kitaoka, Y. T Wang, and S. F. Su, “Human and smart machine co-learning: brain-computer interaction at the 2017 IEEE International Conference on Systems, Man, and Cybernetics,” *IEEE Systems, Man, and Cybernetics Magazine*, vol. 4, no. 2, pp. 6-13, Apr. 2018.

[8] C. S. Lee, M. H. Wang, S. C. Yang, P. H. Hung, S. W. Lin, N. Shuo, N. Kubota, C. H. Chou, P. C. Chou, and C. H. Kao, “FML-based dynamic assessment agent for human-machine cooperative system on game of Go,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 5, pp. 677-705, 2017.

[9] J. Castanon, “10 machine learning methods that every data scientist should know,” May 2019, [Online] Available: https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9.

[10] Wikipedia, “Adaptive neuro fuzzy inference system,” 2020 [Online] Available: https://en.wikipedia.org/wiki/Adaptive_neuro_fuzzy_inference_system.

[11] J. S. R. Jang, “ANFIS: Adaptive-Network-based Fuzzy Inference Systems,” *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 23, pp. 665-685, May 1993.

[12] Y. Tian and Y. Zhu, “Better computer Go player with neural network and long-term prediction,” Feb. 2016, [Online] Available: https://arxiv.org/abs/1511.06410.

[13] Y. Tian, J. Ma, Q. Gong, S. Sengupta, Z. Chen, J. Pinkerton, and C. L. Zitnick, “ELF Open Go: An analysis and open reimplementation of AlphaZero,” May 2019, [Online] Available: https://arxiv.org/abs/1902.04522.

[14] DeepMind, AlphaGo Master series: 60 online games, Jan. 2020, [Online] Available: https://deepmind.com/alphago-master-series.