International Journal of Human–Computer Interaction

ISSN: 1044-7318 (Print) 1532-7590 (Online) Journal homepage: www.tandfonline.com/journals/hihc20

EK-Chess: Chess Learning System Based on Top-Level Chess Expert Knowledge Graph

Mingyu Zhang, Qiao Jin, Qian Dong, Danli Wang & Jun Xie

To cite this article: Mingyu Zhang, Qiao Jin, Qian Dong, Danli Wang & Jun Xie (2025) EK-Chess: Chess Learning System Based on Top-Level Chess Expert Knowledge Graph, International Journal of Human–Computer Interaction, 41:7, 4174-4188, DOI: 10.1080/10447318.2024.2348229

To link to this article: https://doi.org/10.1080/10447318.2024.2348229

Published online: 29 May 2024.

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
2025, VOL. 41, NO. 7, 4174–4188
https://doi.org/10.1080/10447318.2024.2348229

Taylor & Francis
Taylor & Francis Group

Check for updates

EK-Chess: Chess Learning System Based on Top-Level Chess Expert Knowledge Graph

Mingyu Zhang[a,b], Qiao Jin[c], Qian Dong[d], Danli Wang[a,b], and Jun Xie[e]

[a]The State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China; [b]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China; [c]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA; [d]School of Physical Education, Shanghai University of Sport, Shanghai, China; [e]Capital University of Physical Education and Sports, Beijing, China

ABSTRACT

Chess, as a form of intellectual sport, has garnered significant attention from researchers, driving continuous research into computer-assisted player training. However, contemporary teaching or training models frequently confine learners to passive observation of computer-generated results. Beginners may find it challenging to comprehend the cognitive processes underlying decision-making. To address this issue, this article proposes EK-Chess, a knowledge graph-based chess teaching system that encompasses a series of endgame teaching scenarios. This system assists chess beginners in learning the positional evolution in pawn endgames, helping users comprehend offensive and defensive strategies in endgames. User studies validate the effectiveness, and support of the system in endgame learning.

KEYWORDS

Learning system; chess beginner; chess education; expert knowledge; knowledge graph; user study

1. Introduction

Chess is the most iconic intellectual sport globally. Board games have long been considered the litmus test for top-tier human intelligence and artificial intelligence. On one hand, chess provides researchers with ideal experimental conditions to test artificial intelligence algorithms. On the other hand, there is a continuous exploration of ways to integrate computers into the training of human chess players.

In 1997, IBM's Deep Blue (Campbell et al., 2002) achieved a significant technological breakthrough by defeating the world's top-ranked chess player, Garry Kasparov, with a score of 3.5–2.5. This marked a pivotal moment in the field of artificial intelligence. In 2017, AlphaZero (Silver et al., 2018) based solely on the rules of chess, defeated the traditional top-tier chess engine Stockfish (Stockfish, 2023), marking a new era in the development if artificial intelligence in chess. The advancements of artificial intelligence algorithms in the realm of chess have also sparked a trend of progress in the human chess player community. People are increasingly focusing on ways to utilize artificial intelligence technology to enhance human chess-playing abilities.

Currently, computer-assisted technology is widely used in chess teaching and training. Initially, computers could only provide chess databases, allowing chess players to review other players' game records publicly available on various platforms, like Gameknot (Gameknot, n.d.). They could study their opponents' playing styles and improve their chess skills. As computer technology advanced, researchers developed various chess products, including web-based platforms and mobile apps, to facilitate chess education and training.

In the early 21st century, computer engines began to serve as advisors, providing chess players with recommended best moves. Simultaneously, chess players could engage in self-training by playing against computer engines, further enhancing the capabilities of computer-assisted training (Wikipedia 2023a, 2023b). In 2020, a collaboration between the University of Toronto, Cornell University, and Microsoft resulted in Maia, which can assess a player's skill level and engage them at an appropriate level, thereby facilitating assisted training (McIlroy-Young et al., 2020). However, in teaching and training models primarily reliant on computers, learners often only see the presented outcomes of computer decisions without insight into the basis for these decisions. Especially for beginners, the lack of decision-making criteria will make it difficult for beginners to understand and grasp essential knowledge elements, patterns, and logical thinking skills in the development of the chessboard. This is not conducive to training participants' logical thinking abilities.

To address the lack of interpretability in existing educational systems, it is necessary to embed chess knowledge into the chess system. Knowledge graphs, through the use of visualization techniques, describe knowledge resources and their carriers. They can be employed for the construction, analysis, mining, and display of knowledge and the relationships between them. Knowledge graphs find extensive

CONTACT Danli Wang danli.wang@ia.ac.cn The State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China; Jun Xie chessbeijing@126.com Capital University of Physical Education and Sports, Beijing, China

applications in areas such as knowledge retrieval, intelligent question answering, and more (Ji et al., 2021).

Given this background, we processed the knowledge of chess experts. Leveraging the chess insights of top-level chess experts, we analyzed the moves presented by chess AI software. Based on this analysis, we constructed a knowledge graph for chess endgames and designed a chess teaching system grounded in expert knowledge. In the user study, our research involved two learning methods: EK-Chess-based (a system enhanced with top-level chess expert knowledge graph) and traditional methods (based on books and chessboards).

Our research questions are as follow:

- RQ1: How does EK-Chess facilitate chess knowledge acquisition in children compared to traditional book-based learning method?
- RQ2: What is the learning effectiveness of EK-Chess and book-based learning method for easy level (basic endgame strategy), middle level (tactical adaptation), and hard level (strategic mastery) chess endgame scenarios? (In Section 4.5.1, we explained how to categorize scenario types.)

To answer these questions, we conducted a controlled experiment with 20 students aged 8–10 comparing chess learning through EK-Chess software to traditional methods. Each group, consisting of 10 students, was assessed knowledge acquisition by chess tests before and after the learning session. Our quantitative analysis revealed that EK-Chess is user-friendly for children, offering simple operation and interactive learning. The software enhances learning by providing diverse move suggestions from a knowledge graph. Our findings also assessed the system's effectiveness in various chess scenarios.

2. Related work

2.1. Knowledge graph

A knowledge graph is a structured representation of facts, consisting of entities, relationships, and semantic descriptions. Entities can be real-world objects and abstract concepts, relationships represent the relation between entities, and semantic descriptions of entities, and their relationships contain types and properties with a well-defined meaning (Ji et al., 2021.). It has been confirmed that one of the research directions in human-like artificial intelligence is knowledge representation and reasoning. This can enable intelligent systems to represent knowledge and thereby acquire the ability to solve complex tasks (Newell et al., 1959; Shortliffe, 2012).

In recent years, knowledge graphs, as a structured form of human knowledge, have received significant attention from both academia and industry (Dong et al., 2014; Nickel et al., 2016; Wang et al., 2017; Hogan et al., 2020). Indeed, there are numerous large-scale knowledge graphs in the world, including: (1) Cyc: The Common Sense Knowledge Graph created by Douglas Lenat in 1984 (Matuszek et al., 2006). (2) WordNet: A lexical knowledge graph for English, released in 1995 by George A. Miller (Miller, 1995). (3) Google Knowledge Graph: Established by Google in 2012. Knowledge graphs have found practical applications in various domains (Heist et al., 2020). For example, Apple's Siri provides intelligent chat services, and IBM's i2 offers risk analysis services, among others. These knowledge graphs play a crucial role in powering intelligent systems and applications.

Chess is a complex game with many rules, strategies, and tactics. It can be empowered by knowledge graph in many ways. A knowledge graph can provide a structured and organized way for beginners to learn and understand these concepts. It can break down the game into smaller, more digestible pieces, making it easier for beginners to grasp the fundamentals. In addition, Knowledge graphs can be used to track a learner's progress. Beginners can see where they are in their chess learning journey and what topics they have mastered or need to focus on. Notably, it can also help motivate and guide their learning efforts. A knowledge graph can recommend specific resources, exercises, or lessons based on a learner's current knowledge and skill level. Inspired by these benefits, we chose to utilized knowledge graph in our system to customize the learning process and make it more effective and engaging.

2.2. Chess engines

Chess engines are computer programs that can analyze the value of chess pieces in different positions and generate one or more recommended moves. They play a crucial role in chess player training in the world of chess. These engines are typically command-line backends without graphical interfaces or windows, requiring graphical user interfaces (GUIs) as front-ends to complement their usage (Wikipedia, 2023a, 2023b). This separation of the chess engine and chess software allows them to evolve independently. To effectively support chess software, chess engines adhere to standardized interface protocols.

Commonly used chess engines include Leela Chess Zero, ConvChess, and Stockfish, among others (Wikipedia, 2023a, 2023b). Leela Chess Zero (Leela Chess Engine, 2023), introduced in 2018, is a modern computer engine based on reinforcement learning, while the others are based on the long-standing Alpha-Beta pruning algorithm. It assigns scores to moves based on the position and recommend the best moves. ConvChess (Barak & Nishith, 2015) is a convolutional neural network (CNN) engine by Barak Oshri and Nishith Khandwala. It was the first engine using CNN. It is implemented in Python using its machine learning libraries such as NumPy. Stockfish, first released in 2008, is known for its strong pruning and endgame reduction capabilities, allowing for deeper searches. It has won several chess engine championships and is considered the most powerful CPU engine among chess engines based on the Alpha-Beta pruning algorithm (Knuth & Moore, 1975).

Although many chess engines have powerful playing capabilities, and some online platforms support students' chess play and give the right move, their primary focus is not on chess teaching and learning. They lack emphasis on guiding the reason why to play the right move, which may trigger students' transition path dependence on engines and do harm to users in learning chess knowledge.

2.3. Computer-assisted chess teaching tools

Computer-assisted chess teaching tools with GUIs that allow integration with chess engines as backends are a popular category of tools. These tools often share common features. Many of them can integrate multiple chess engines, offering users the ability to configure and use their preferred chess engines. Stockfish is often set as the default engine for many chess teaching tools.

Regarding move recommendations, the software Arena (Martin, 2019) provides move scores for board positions but may lack detailed explanations or recommendations for moves. Online chess analysis tool Chess Compass (Chess Compass, n.d.) displays move scores in a visually appealing manner, making it easy for players to view and reference them but may not provide the basis for move recommendations. Commercial software Shredder (Shredder, n.d.) enhances move recommendations in endgames by showing the minimum number of optimal moves instead of numerical scores, making it more understandable for players. DecodeChess (DecodeChess, 2022) is the first and, currently, the only chess teaching product that can provide natural language explanations for recommended moves that are suitable for human understanding. However, its explanatory capabilities may weaken significantly when the system reaches endgame positions, providing only subsequent moves without explaining specific strategic ideas and significance.

The advancement of AI in chess has paralleled the emergence of modern computing technologies, some computer-assisted chess teaching tools have integrate AI techniques. For instance, explorations have been made for teaching purposes (Kerner, 1995), including tree structure explanations and saliency-based methods (i.e., highlighting pieces crucial for executing the selected move) to understand the behavior of trained models (Gupta et al., 2020). And another way to generating comments using natural language utilizing social media data (Jhamtani et al., 2018), creating a chess question-answering dataset (Cirik et al., 2015), or learning evaluation functions reflecting sentiments in chess discussions (Kamlish et al., 2019).

Contemporary teaching or training models frequently confine learners to passive observation of computer-generated results. Moreover, existing chess teaching tools fail to explain the specific strategic ideas and significance behind chess moves.

2.4. Summary

The work we have described showcases remarkable achievements in the domain and teaching scenarios of chess. However, current systems face several challenges. First, there's a limited focus on endgame teaching content in chess education. Second, there's a lack of chess teaching systems based on knowledge graphs. Third, the current artificial intelligence-related chess teaching processes lack standardized evaluation of participants' learning outcomes.

To address the issues of limited endgame teaching content and insufficient educational materials in chess instruction, our study constructed a knowledge graph for chess endgames based on expert knowledge. This knowledge graph encodes chess positions and moves using the Forsyth-Edwards Notation (FEN; Edwards, 1994) and Long Algebraic Notation (LAN; Hooper & Whyld, 1996) establishes boards and moves nodes, and connects nodes in a directed graph format. Our research introduced a new chess teaching system called EK-Chess. On one hand, it allows users to play chess within the system, offering real-time feedback. On the other hand, the system leverages expert knowledge from the constructed chess endgame knowledge graph to provide move information and explanations for specific scenarios. We further assessed the effectiveness and convenience of this system, with the results demonstrating that the system enhances the learning process for chess players, achieving better learning outcomes compared to traditional book-based learning methods.

3. System description

3.1. Overview

Inspired by prior work on training tools for chess learning and potential of knowledge graph, we created EK-Chess, a chess teaching system for beginners, by integrating a knowledge graph and GUI into an application with an intuitive interface, customizable learning paths, and expert knowledge. The system comprises a chess endgame knowledge graph (Section 3.2) and a chess application (Section 3.3), with a learning record mechanism (Section 3.4) connected them together. The application provides basic chess playing functions, chess endgame knowledge graph provides information on chessboard positions, moves, and expert knowledge for teaching, aiming to deliver a professional and personalized learning experience for users.

3.2. Chess endgame knowledge graph (CEKG)

A knowledge graph is a structured representation of facts, encompassing entities, relationships, and semantic descriptions, providing a meaningful framework for abstract chess concepts. The Chess Endgame Knowledge Graph (CEKG) was constructed using a top-down, manual approach (Qiao et al., 2016). This method involves human experts directly creating and organizing the graph's content. Since chess endgame knowledge has relatively low cost and complexity, the manual construction approach is a cost-effective choice, providing a straightforward and highly accurate means of building the knowledge graph.

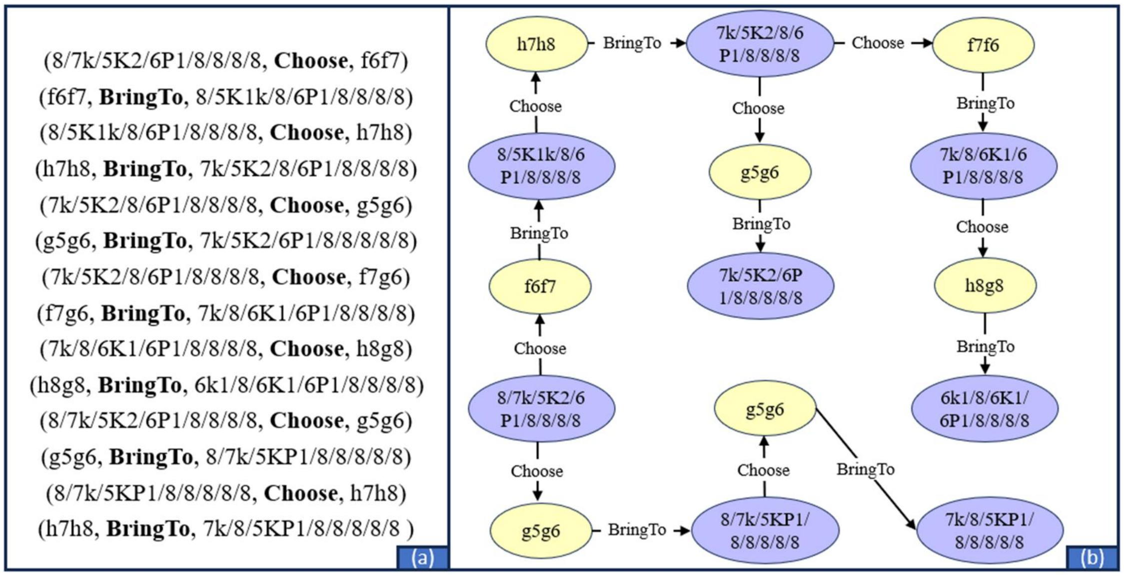

Figure 1. An example of triples and CEKG. (a) is triples of knowledge; (b) is entities and relations in CEKG.

In the design of the CEKG, the basic unit of the knowledge graph is the triple consisting of "Entity-Relationship-Entity." Under the guidance of chess experts, we have categorized all the entities into two classes: "Board" and "Move." And relationships have been categorized into two classes: "Choose" and "BringTo."

The "Board" class is used to describe chess board content. It has several attributes, including: (1) Id: This attribute serves as a unique identifier for each chess board. (2) FEN: The FEN string content that represents the layout of the corresponding chess board. (3) Label: This is a string description specific to the chess board. For the initial position (opening board), it may indicate the category of the endgame. For the final board, it might provide a summary of the endgame category. For intermediate boards, this field may be empty. These attributes help to uniquely identify and describe chess board configurations within the knowledge graph.

The "Move" class is used to describe chess move content. It includes the following attributes: (1) Id: This attribute serves as a unique identifier for each chess move. (2) LAN: It represents the specific chess move made, describing the move itself. (3) Promotion: This attribute is used to indicate pawn promotion situations. It is typically presented as a triplet {LAN, promotion, statement} where "LAN" represents the move, "promotion" describes the promotion situation (e.g., "Q" for Queen promotion), and "statement" provides a string explanation for the move. The promotion" attribute is typically used in conjunction with the "move" attribute to represent pawn promotions. For instance, "A8-Q" indicates a move from A8 with a pawn promotion to a Queen (Q). If there is no promotion involved in the move, this attribute is set to None. This structure allows for detailed descriptions of chess moves, including pawn promotions, in the knowledge graph.

Choose Relationship: This relationship connects "Board" to "Move." "Board" uses the "Choose" relationship to point to "Move," indicating which move is chosen on that particular board.

BringTo Relationship: This relationship connects "Move" to "Board." "Move" uses the "BringTo" relationship to point back to "Board," showing which board the move brings to.

In CEKG, we incorporate the expertise and recommendations of top chess experts and process the relevant content accordingly. The knowledge can be expressed in a triple in the form of (Board, Choose, and Move) and (Move, BringTo, and Board), examples of triples are illustrated in Figure 1(a). CEKG can also be represented as a directed graph with nodes as entities and edges as relations, examples of entities and relations are illustrated in Figure 1(b).

We are using Neo4j Community version 4.2.3 as the default database. With a total of 438 nodes and 426 relationships.

3.3. EK-Chess application

The EK-Chess application was developed using the PyCharm platform and is implemented in Python. It facilitates queries and retrieval from the CEKG. Additionally, the system is integrated with the Stockfish engine, enabling a knowledge graph-based chess endgame teaching system. The application consists of two phases: the initialization phase and the runtime phase.

During the initialization phase, the user selects a chess position to begin their learning journey. Once they enter a specific scenario, the system retrieves data from the CEKG, extracts the FEN data from the "Board" class, and generates a teaching scenario with chess piece information. As shown in Figure 2, after the user's selection, they enter the educational scenario for that chess position.

In the execution phase, when the user selects a scenario for learning, they enter the chessboard. Taking the King and Pawn endgame scenario 4 as an example, the system retrieves the "Board" node corresponding to the king and pawn endgame 4 in CEKG, extracting relevant expert knowledge, and first parses the FEN information in the "Board" node. The system will parse the types, quantities, and

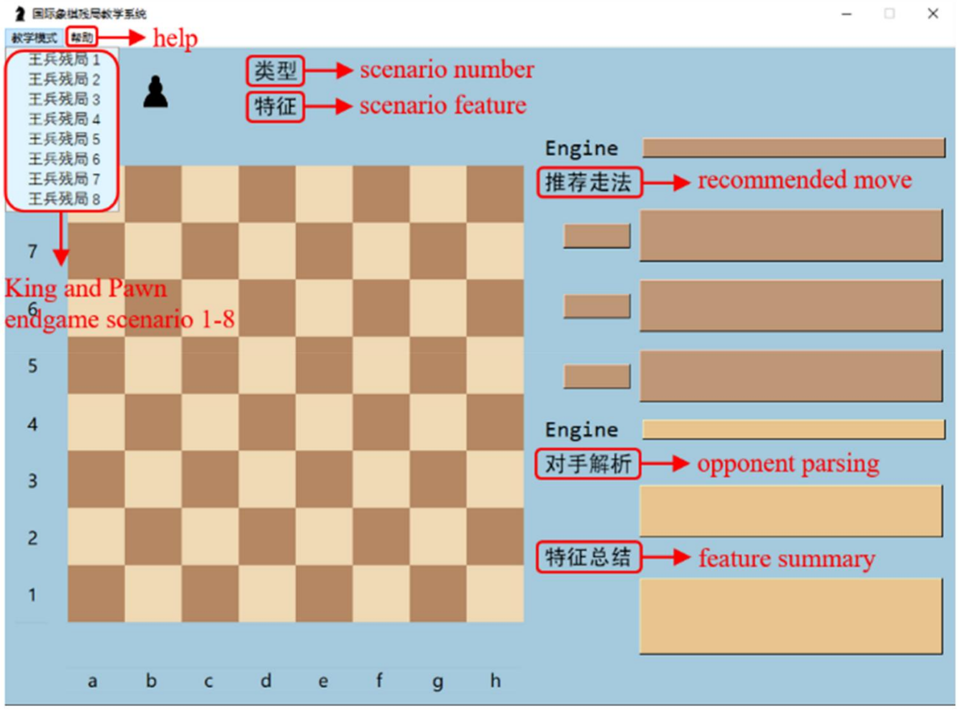

Figure 2. User interface of initialization phase.

positions of pieces in King and Pawn Endgame Scenario 4, and display this information on the electronic chessboard available for user. Additionally, the corresponding learning scenario information will be shown in the number information and feature information sections. As shown in Figure 3(a), users will receive the opening information of the learning scenario along with the corresponding number and feature information. Here, "4" indicates that this learning scenario is King and Pawn Endgame Scenario 4, and "王兵残局, 关键格" is "King and Pawn Endgame, critical square" in English. And the first step for the user is choose their side (note that not all endgame scenarios have this step; some scenarios have fixed sides).

Taking the example of a student choosing the white side, at this point, the system using CEKG to retrieves the "Choose" relationships for the current "Board" node, extracts move recommendations and explanations from the corresponding "Move" nodes, and presents it in the user's interface, as shown in Figure 3(b). For the recommended move, we offer users two options for their next move, each consisting of two components. The first component specifies the piece to be moved and its destination (for example, "Kd1" signifies moving the white king from c2 to d1). The second component provides an explanation for the move (for example, "Kd1" followed by the explanation "退王," which translates to "King retreat" in English). After the user completes their move as per the instructional guidance, the system retrieves and executes the "BringTo" relationship for the current "Move" node, progressing to the next "Board" node and retrieving its "Choose" relationships, continuing this process until the outcome of the scenario is determined. Once the user completes all the moves in this chess position (in Section 3.4, we have explained how to complete all the moves.), the system will provide a summary of the teaching in the feature summary box, the summary is an overall explanation of the scenario by top-level chess experts, elucidating the strategic approach users should adopt to secure victory, as shown in Figure 3(c).

The application based on Pycharm platform developed, where users can complete the learning process using the mouse and computer screen. We worked with chess experts and chess teachers to design 32 King and Pawn endgame scenarios. We create our scenario in application following the Chess Endgame Strategy (Xie & Lin, 2010). (In our experiment, we selected eight representative King and Pawn endgame scenarios as learning materials). Users can click on different scenario numbers to enter different chess positions for learning.

3.4. Learning record mechanism

As shown in Figure 3, users can choose to play as either the white or black side in some chess positions, and multiple recommended moves are provided within a single position. Therefore, we designed a learning record mechanism to help users keep track of the knowledge they have already learned.

The principle of this mechanism is that when users select an endgame scenario for learning, system extracts all the associated "Board" and "Move" elements related to the scenario. These elements are combined as a pair following the direction of "Board Choose Move", we defined pair as (B, M), B represents the "Board" class node, and M represents the "Move" class node. We add a status flag to convert the pair into a triplet: (B, M, S). So, at the beginning of learning each endgame scenario, we treat the CEKG as a tree structure and create a temporary array using

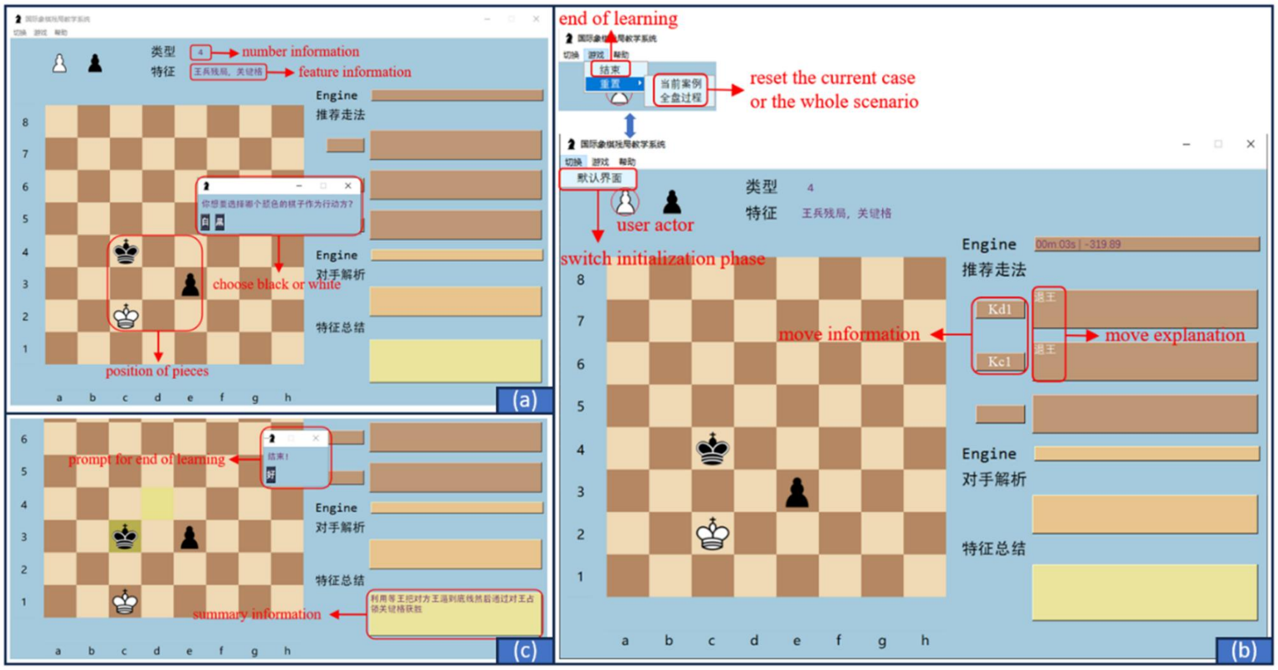

Figure 3. User interface of execution phase. (a) Information about the endgame scenario and prompts for users to choose either the black or white side; (b) the recommended moves, move explanation, and user-executable buttons; (c) learning completion prompts and learning summary information.

a breadth-first algorithm (Bundy & Wallen, 1984), like this: $T = ((B_1, M_{1,1}, S_{1,1}), \ldots, (B_n, M_{n,m}, S_{n,m}))$. The initial state of S is 0. Based on this, the function is formalized as following Eq. (1):

$$R = \prod_{i=1}^{n} \prod_{j=1}^{m} \left(S_{i,j}(B_i, M_{i,j})\right). \tag{1}$$

R represents the result of computation, B_i is the i-th node retrieved from the "Board" class, $M_{i,j}$ is the j-th node in the "Move" class that is pointed to by B_i through the "Choose" relationship. Only when the user completes the $M_{i,j}$ operation in the B_i node does $S_{i,j}$ change from 0 to 1. As a result, the value of R can be either 0 or 1. 0 indicates that the user has not completed all the operations for the selected endgame scenario, while 1 indicates that the user has completed all the operations for the selected endgame scenario.

Using King and Pawn Endgame 4 as an example, when a user has not completed the learning process for this scenario, the system prompts the user to continue and finish the remaining content, as shown in Figure 4(a). If the user selects "yes," the chess position will be reset, and the moves that the user has already selected will be color-coded to indicate their choices, as depicted in Figure 4(b). Following this approach, once the user completes all the moves in this chess position, the system will no longer prompt them to continue learning. Instead, it will provide a summary of the teaching in the feature summary box, as shown in Figure 3(c). However, if the user chooses "no" when asked whether to continue learning, the system will directly return the summary content and prematurely end the learning for this chess position.

4. User study

4.1. Study overview

In this experiment, 20 students were invited to participate, they have mastered the moves and rules of the pieces, can independently complete a game, but are still at the beginner stage without specific chess knowledge, skills, and strategies. To study the learning effectiveness of this system, we conducted a between-subject design, wherein participants were divided into two groups, each consisting of 10 children. Participants in the first condition used EK-Chess for learning, while another condition used traditional books for learning. Both conditions used the same chess scenarios, with the primary difference being one used electronic device for operation, and the other used physical chessboards. After mastering the system and book usage, participants were required to independently complete a pre-test, and after learning, they completed a post-test to assess their chess knowledge and skills. Besides the questionnaire, we also collected timing date from the participants, including the time spent on learning and answering questionnaire. The goal of the study is aiming to answer the following research questions (RQ):

- RQ1: How does EK-Chess facilitate chess knowledge acquisition in children compared to traditional book-based learning method?
- RQ2: What is the learning effectiveness of EK-Chess and book-based learning method for easy level (basic endgame strategy), middle level (tactical adaptation), and hard level (strategic mastery) chess endgame scenarios? (In Section 4.5.1, we explained how to categorize scenario types.)

Figure 4. King and pawn endgame 4 learning example. (a) Prompt for incomplete learning and choice of whether to continue; (b) color-coded feedback.

4.2. Experimental environment

The experiment was conducted on May 18th and 19th, 2023, in the Chess Classroom of Chengbei Center Sixth Street Primary School in Changping District, Beijing. In addition to the necessary tables, chairs, the room was also equipped with two cameras, five laptops running the EK-Chess system, five chessboards, several physical books, and several sets of pre-test and post-test questions, as shown in Figure 5. Two researchers and one school teacher were present during the study. One researcher served as the experimenter, responsible for program guidance, experiment documentation, and other experimental procedures. The other researcher assisted in preparing the experimental equipment and operating video recording devices. School teacher helped maintain order in the experimental room. During the experiment phase, the cameras recorded the entire experimental process.

Figure 5. Experiment environment.

4.3. Participants

A total of 20 students took part in this experiment, consisting of 10 boys and 10 girls, with an average age of 8.75 (SD = 0.94, age range: 6–10). We recruited students for our study by first having their teachers identify potential participants. The teachers then contacted the parents to inform them about the study. Following this, we either directly communicated with the parents to obtain their consent through completion of consent forms or sent the forms home with the students for parental approval. The participants were divided into two groups using random allocation: one group used EK-Chess for learning (G1), and the other group used books for learning (G2). Under each condition, ten children were divided into two groups, each with five members under different conditions. On May 18th, ten participants were, with five using EK-Chess for learning and five using books. The same participant ratio was used on May 19th in a similar way. Considering that chess is a highly specialized sport, and the learning scenarios designed in this system require participants to have a proficient understanding of basic chess knowledge but not yet be familiar with more advanced chess techniques, all participants have less than half a year of exposure to chess. Their previous chess learning experiences were limited to group classroom learning at school, and their chess skills are generally at the beginner level.

4.4. Procedure and task design

The experimental process took approximately 55 min, and the specific procedure as shown in Figure 6:

1. Experiment instruction (7 min): researchers provided instructions to the five participants on how to use EK-Chess and the other five participants on how to use books. They explained how to use the system, the learning tasks that needed to be completed, and the process for filling out the pre-test and post-test questions.
2. Practice tasks (8 min): researchers guided G1 and G2 separately to complete a practice learning task. During this time, participants were allowed to ask any questions to ensure a thorough understanding of how to use EK-Chess and books.
3. Pre-test (10 min): researchers distributed the pre-test questionnaires (the questionnaire assesses the

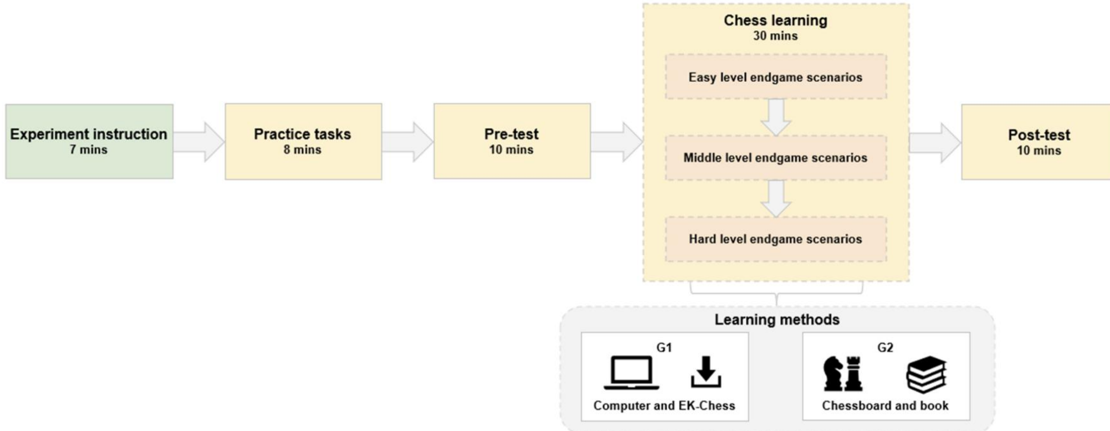

Figure 6. Overview of the experiment process.

participants' prior knowledge and refer Section 4.5.1) to the participants and guided them on how to fill out the questionnaires. Participants worked independently on the questionnaires, and after they completed the questionnaires, the researchers recorded the time taken and collected the questionnaires. They were not provided with correct answers or additional instruction after the test, and immediately proceeded to the formal experiment for learning.

4. Chess Learning (30 min): participants in G1 completed eight King and Pawn endgame scenarios learning tasks using the EK-Chess program on the computer. Participants in G2 completed the same eight learning tasks from the books using chessboards. Each participant worked independently on their tasks, and the process was recorded using cameras. The researchers also recorded the learning time for each participant.
5. Post-Test (10 min): researchers distributed post-test questionnaires (the questionnaire assesses the participants' posterior knowledge and refer Section 4.5.1) to the participants, participants fill out the questionnaires. Participants worked independently on the questionnaires, and after they completed the questionnaires, the researchers recorded the time taken and collected the questionnaires.

4.5. Measures and data collection

4.5.1. Questionnaires

We evaluated the participants' pre- and post-test chess knowledge level using questionnaires. To ensure the consistency in the difficulty level of the pre-test and post-test questionnaires, the questionnaires were evaluated and formulated by chess experts and teachers. Both the pre-test and post-test questionnaires consisted of 10 multiple-choice questions. Examples are shown in Figure 7.

Additionally, to investigate the learning effects of EK-Chess and books on different types of King and Pawn endgame scenarios. Guided by the advice of top-level chess experts, we designed and categorized the questions into three levels, each defined by their level of difficulty and complexity.

The criteria for difficulty classification are as follows: (1) Number of moves from start to finish; (2) Number of reasonable moves; (3) Whether similar content has been covered in the system or book.

Easy level (basic endgame strategy): The necessary conditions for the promotion of a pawn in the king and pawn endgame, as well as the understanding of the stalemate rules for the defending side. The questions in the questionnaire related to this level.

Middle level (tactical adaptation): Understanding the mutual constraints between the king of the advantaged side and the king of the disadvantaged side, adapting to the interference factors of changing piece colors, and making correct choices among multiple chess moves. The questions in the questionnaire related to this level.

Hard (strategic mastery): Comprehensive application of learned strategies in complex situations, assessing the integration of learners' mastery of chess knowledge. The questions in the questionnaire related to this level.

This categorization enables us to perform a comprehensive analysis of the participants' question accuracy, both globally and by specific types, during the subsequent data analysis.

4.5.2. Data analysis

We divided the participants into two conditions, collected 20 sets of pre-test questionnaire data and 20 sets of post-test questionnaire data. We conducted a mixed-design analysis (Huck & McLean, 1975) using the factors of pre-test, post-test scores, and learning conditions (G1 or G2).

In order to answer the research questions, we analyzed the quantitative results from the pre-test, post-test, time taken in learning, and time taken in answering questionnaires. For the quantitative data, we conducted one-way ANOVA for those two methods (EK-Chess and Book). We ensured that the statistical analyses were performed properly and fitted the requirements and assumptions. For the ANOVA, for example, we applied between-subject statistics, which are used to compare observations of an outcome for the different person being measured at different time points (methods). In order to examine the significant effect of the

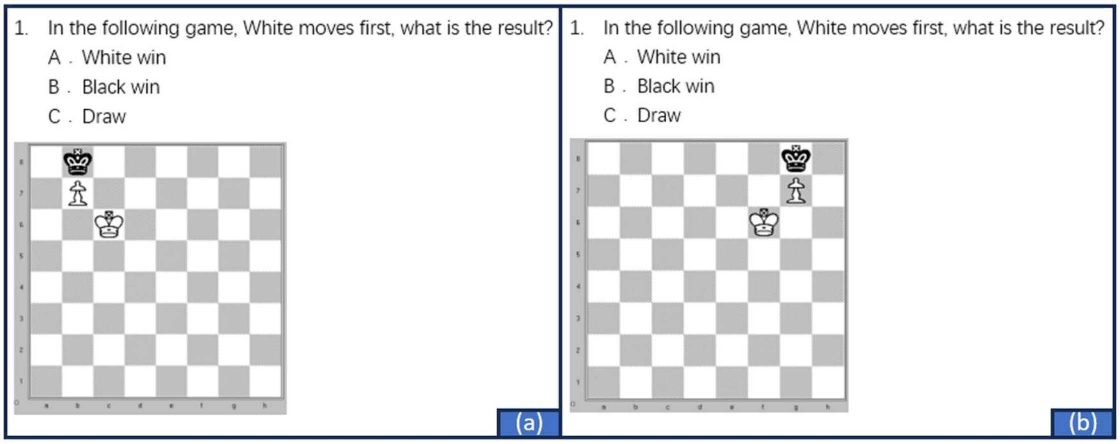

Figure 7. Examples from the questionnaires. (a) pre-test; (b) post-test.

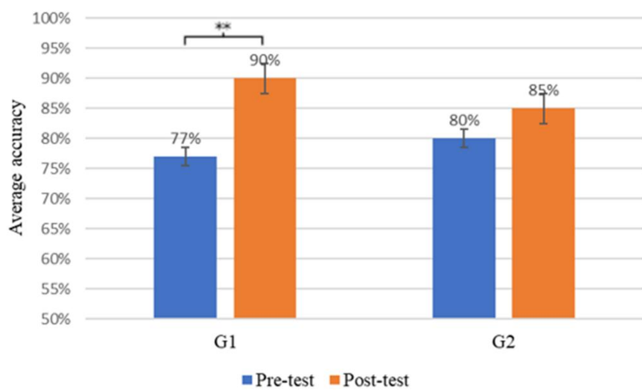

Figure 8. Chess proficiency test.

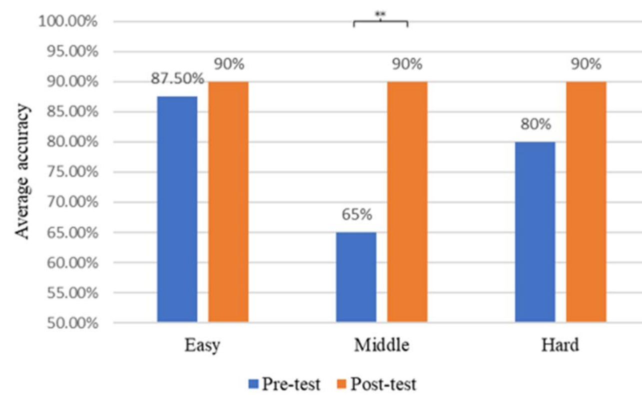

Figure 9. Results for different question types in G1.

ANOVA results, we conducted specific condition comparisons using a T-test at the 0.05 significance level.

5. Results

5.1. How does EK-Chess facilitate chess knowledge acquisition in children compared to traditional book-based learning method?

After completing the instructional content and practice exercises, the participants took the first ability test (pre-learning). After completing all the formal experiment tasks, the participants took the second ability test (post-learning). The data for the chess proficiency test before and after the experiment is shown in Figure 8 below.

G1 participants had an average accuracy rate of 77% in pre-test and 90% in post-test, there was a significant difference ($p = 0.030$) in pre-test and post-test average accuracy for G1 participant. G2 participants had an average accuracy rate of 80% in pre-test and 85% in post-test, however, for G2 participants, there was no significant difference ($p = 0.376$) between pre-test and post-test average accuracy. We found that participants using both learning methods improved their average after learning. Additionally, G1 participants showed a higher improvement.

Combining the results of significant differences, this suggests that the chess teaching effect of EK-Chess is superior to book-based teaching, and participants who used EK-Chess for learning experienced a significant improvement in chess proficiency before and after learning.

5.2. What is the learning effectiveness of EK-Chess and book-based learning method for easy level (basic endgame strategy), middle level (tactical adaptation), and hard level (strategic mastery) chess endgame scenarios?

As mentioned in Section 4.5.1, we divided the questions into three types, and the distribution of question types is the same for both pre-test questionnaires and post-test questionnaires. They are distributed as follows: easy level questions (4 questions), middle level questions (4 questions), and hard level questions (2 questions).

As shown in Figure 9, for G1 participants, in pre-test: (1) Average accuracy for easy level questions: 87.5%. (2) Average accuracy for middle level questions: 65%. (3) Average accuracy for hard level questions: 80%. In post-test: (1) Average accuracy for easy level questions: 90%. (2) Average accuracy for middle level questions: 90%. (3) Average accuracy for hard level questions: 90%.

As shown in Figure 10. For G2 participants, in pre-test: (1) Average accuracy for easy level questions: 87.5%. (2) Average accuracy for middle level questions: 70%. (3) Average accuracy for hard level questions: 85%. In post-test: (1) Average accuracy for easy level questions: 80%. (2)

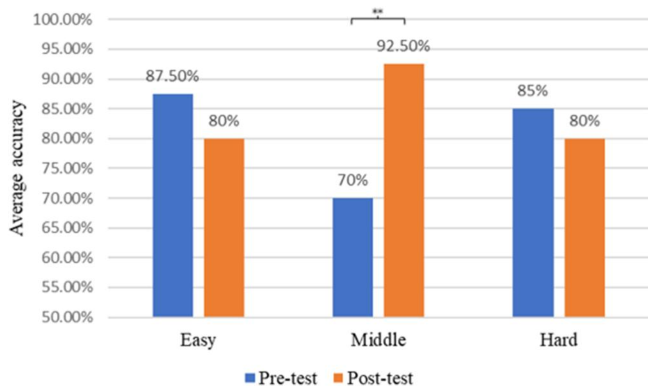

Figure 10. Results for different question types in G2.

Average accuracy for middle level questions: 92.5%. (3) Average accuracy for hard level questions: 80%.

It appears that G1 participants did not experience a decrease in average scores in any question type after the experiment, while G2 participants experienced a decrease in average scores in easy and hard level questions.

To further analyze our experimental results using a mixed-design analysis, we considered the factors of pre-test, post-test, question type, and learning methods. For G1 participants: (1) In easy level questions, there was no significant difference between the pre-test and post-test ($p = 0.618$). (2) In middle level questions, there was a significant difference ($p = 0.019$), with improved performance after using EK-Chess. (3) In hard level questions, there was no significant difference ($p = 0.418$). For G2 participants: (1) In easy level questions, there was no significant difference between the pre-test and post-test ($p = 0.146$). (2) In middle level questions, there was a significant difference ($p = 0.032$), with improved performance after using books. (3) In hard level questions, there was no significant difference ($p = 0.684$).

We found that both learning methods had no significant differences in easy and hard level questions, but there was a significant difference in middle, with G1 performing better in this type. This suggests that EK-Chess may be more effective for enhancing performance in specific types of questions, particularly those related to middle.

Based on Figure 10, it's evident that G2 experienced a decrease in accuracy in easy and hard level questions. We delve into the specific analysis of G2's performance in easy and hard level questions, as shown in Figure 11. For easy level questions: (1) four instances showed a decrease in the number of correct answers. (2) Five instances had no change in the number of correct answers. (3) One instance showed an increase in the number of correct answers. For hard level questions: (1) one instance showed a decrease in the number of correct answers. (2) Nine instances had no change in the number of correct answers. (3) There were no instances with an increase in the number of correct answers.

Among the four instances of decreasing correct answers in easy, three times involved a decrease from four correct answers to three correct answers, with only one question answered incorrectly. The other instance involved a decrease from three correct answers to two correct answers. Considering Figures 8 and 9, we observe that the pre-test average number of correct answers for easy level questions is the highest and very close to having all correct answers. Furthermore, based on the analysis provided, book-based learning approach had no significant learning effect on easy level questions. We believe that the difficulty of easy level questions is too low, and the majority of users had already mastered the chess knowledge associated with easy level questions scenarios. Users did not gain additional knowledge about easy level question scenarios after the learning process. For hard level questions (a total of two questions), seven participants answered all correctly, two participants had a correct count of 1, and only one participant decreased the number of correct answers from 1 to 0.

As shown in Figure 12, G1 experienced a single instance of a decrease in the number of correct answers in easy level questions, four instances of maintaining the same number of correct answers, and five instances of an increase in the number of correct answers. In hard level questions, G1 encountered a single instance of a decrease in the number of correct answers, six instances of maintaining the same number of correct answers, and three instances of an increase in the number of correct answers.

Consistent with the results analyzed in the previous section, we believe that easy level questions are too easy, with the majority of users having already mastered the chess knowledge associated with easy level question scenarios. Users did not gain additional knowledge about easy level question scenarios after completing the learning process. Similarly, for hard level questions, considering that book-based learning had no significant learning effect on this question type and that there were seven instances of all correct answers in pre-test, we conclude that the learning process did not provide participants with additional knowledge about hard level question scenarios. Therefore, while G1 showed improvements in accuracy for easy and hard level questions, it is reasonable to observe that these improvements did not exhibit significant differences.

The number of participants with changes in the number of correct answers in easy and hard level questions for both G1 and G2 participants is summarized in Figure 13. In easy level questions, G1 had a significantly larger number of participants with an increase in the number of correct answers compared to G2. Conversely, G1 had a lower number of participants with a decrease in the number of correct answers compared to G2.

Considering the analysis presented earlier, we conclude that EK-Chess exhibits the better effective teaching outcomes in middle level questions, and it also performs better than the textbook-based learning approach in easy and hard level questions.

6. Discussions

In this article, we presented EK-Chess, a chess teaching tool. With the assistance of chess experts and chess educators, we constructed a knowledge graph of chess endgames to

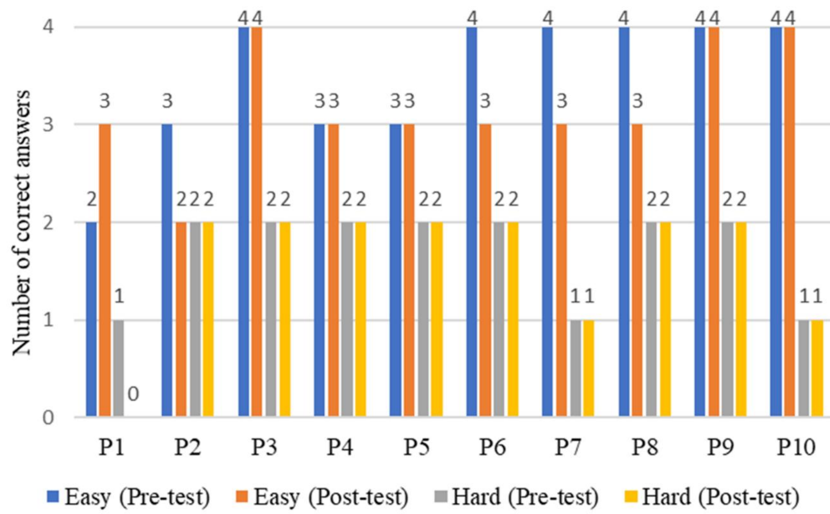

Figure 11. Results for number distribution for G2 in easy and hard level questions, P means participant.

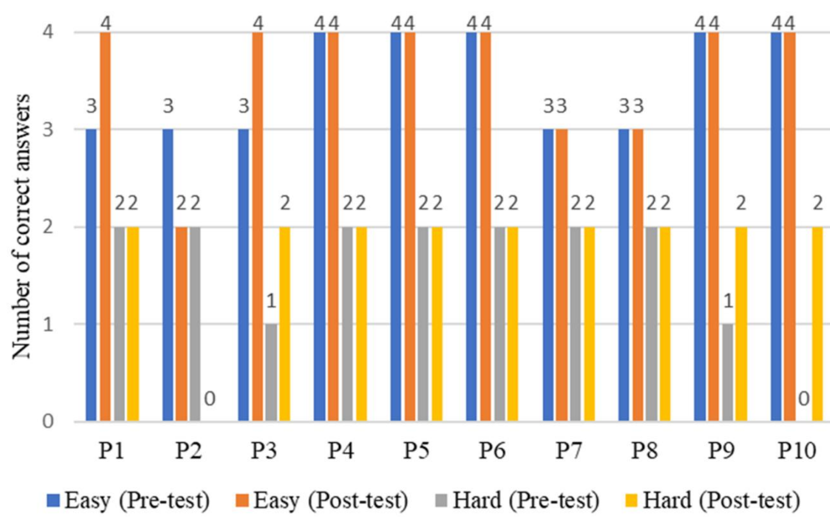

Figure 12. Results for number distribution for G1 in easy and hard level questions, P means participant.

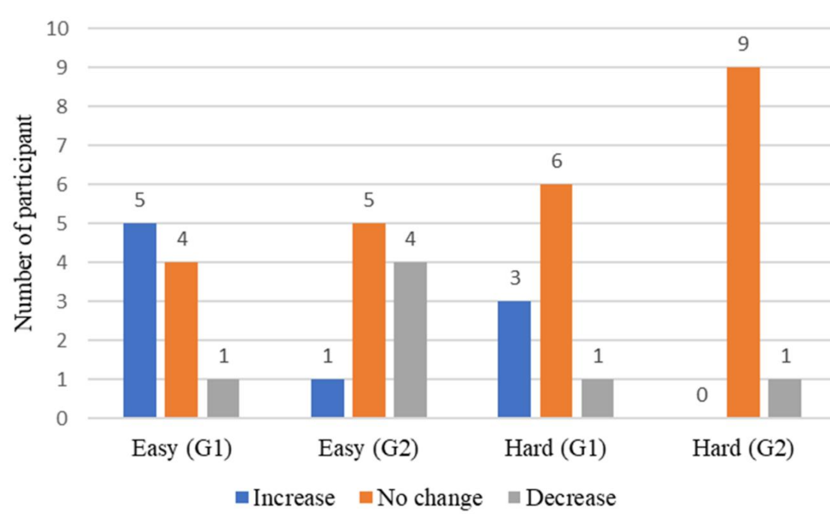

Figure 13. The participant changes in easy and hard level questions for both G1 and G2.

provide users with professional knowledge when they use the system for learning. In the following sections, we begin by summarizing our findings in line with the research questions, then provide the implications of these results for further research. Following this, we discuss the limitations in our study.

6.1. Principal results

In our user study, RQ1 centered on comparing the effectiveness of EK-Chess and traditional book-based methods in facilitating chess knowledge acquisition. Our quantitative results showed that both learning methods can help users acquire chess endgame knowledge, but the EK-Chess-based learning approach yields better learning outcomes than the book-based approach. Due to the anticipated differences in learning outcomes between the two methods, we further subdivided the teaching scenarios and pre-test and post-test questionnaires to provide more detailed insights, and the results align with our prediction.

RQ2 examined the learning outcomes of EK-Chess and book-based learning methods in different types (easy, middle, and hard level questions) of king and pawn endgames. Quantitative results indicate that, in middle level questions, participants from both groups (G1 and G2) showed significant improvements, with G1 participants exhibiting a better improvement.

Surprisingly, G1 participants experienced a decrease in the average accuracy of responses in Easy and Hard level questions after the learning phase. One possible reason is that our participants were primary school students with an average age of 8.75, and they showed a preference for electronic devices over traditional books. Moreover, participants in G1 did not exhibit a decrease in average correctness rates for easy and hard level questions. Based on our results (Section 5.2), for participants in G2, continuous searching in books during the learning process increased their external cognitive load (Cierniak et al., 2009). Existing literature has shown that cognitive load can affect users' emotions (DeFraine, 2016), and personal decisions can be influenced when users are in a negative emotional state (Zhao Y et al., 2022; Wang et al., 2024). Therefore, one possible explanation for our results is that G2 participants experienced increased cognitive load after learning, leading to negative emotions and subsequently influencing their decisions, resulting in a decrease in the average correctness of responses to the questions. Which leads to our research implications for record mechanism (Section 6.2).

6.2. Research implications on learning record mechanism

In our designed chess endgame scenarios, a fixed scenario can have multiple move paths. Our goal is to enable users to learn various strategies for specific scenarios, allowing them to comprehensively understand the chess knowledge taught in a scenario from different perspectives. Therefore, our system needs a mechanism to track user learning progress. Hence, we designed a learning recording mechanism.

Based on our results of RQ1 and RQ2, we found that some participants experienced a decrease in accuracy on some questions after learning. Combining with the discussion in Section 6.1, The key difference between the two methods is that book-based learning method lacks learning record mechanism. As mentioned in Section 3.4, our designed learning recording mechanism allows users to intuitively identify steps that have not been completed. In contrast, G2 participants may need to spend more time thinking to make such judgments and proceed with their learning.

Therefore, we believe that the learning recording mechanism is effective in teaching scenarios where there are multiple solution paths for a question. For example, the situation of multiple solutions in mathematical problems and the chess teaching addressed in this article. By assisting users through the learning recording mechanism, we aim to reduce cognitive load during the learning process, enhancing overall learning effectiveness. On the other hand, the learning record mechanism can also serve as a tool for recording user learning process data. In conjunction with the field of recommendation systems (Ko et al., 2022), modeling user learning preferences and knowledge mastery based on data from the user's learning process can be used to achieve personalized teaching recommendations, thereby improving teaching effectiveness.

6.3. Research implications on knowledge representation

Our results from RQ1 and RQ2 echoed prior work that knowledge graph is a powerful tool to assist students in chess knowledge learning. So, we suggest that future work can also consider similar approach to support chess learning, or using other deep learning models. We believe that employing a knowledge graph as the source of knowledge for a chess education system is highly suitable. First, knowledge graphs connect chess positions and moves in the form of a directed graph, offering excellent extensibility. Second, as structured knowledge, knowledge graphs can serve as a data source for training deep learning models. Existing deep learning techniques such as CNNs (Gu et al., 2018) and graph neural networks (Wu et al., 2020) can extract data features from knowledge graphs, facilitating further research.

Large language models (LLMs; Chiang et al., 2023) have achieved substantial impact across a variety of natural language processing (NLP) tasks. Their remarkable capabilities stem from the ever-increasing number of parameters and training data, which expands in correlation with their massive knowledge and emergent abilities like chain-of-thought reasoning (Kojima et al., 2022) and in-context learning (Brown et al., 2020).

However, LLMs still struggle with factual knowledge considering issues like completeness, timeliness, faithfulness, and adaptability (Kandpal et al., 2023). First, LLMs demonstrate limitations in terms of timely updates and expertise in specific domains and LLM can hardly incorporate new knowledge *via* continued training, which hampers the ability to tailor these models to accommodate specific knowledge demands. Therefore, these knowledge demands encourage comprehensive research efforts toward integrating LLMs with external sources of knowledge.

The existing research has demonstrated that enhancing LLMs with external knowledge can improve their performance in specific domains, like Chatlaw (Cui et al., 2023) and KnowledGPT (Wang et al., 2023). Therefore, the knowledge graph CEKG that we designed is meaningful. Our CEKG can be applied as a knowledge supplement for large language models in the field of chess, using structured top-level expert knowledge to enhance the model's performance. In the future, we also plan to integrate generative language models with knowledge graphs, incorporating the knowledge graph with expert insights into the generative language model to generate explanations for current moves and summaries of chess positions.

6.4. Limitations and future work

This study presents several limitations within its system design and evaluation. First, the user interface is characterized by a fundamental simplicity, which might challenge the user-friendliness, particularly for younger demographics. Second, there is a discernible need to augment the knowledge graph. Presently, it is predominantly oriented toward King and Pawn endgames, suggesting a narrowed focus in the content spectrum. Third, the methodological rigor in experimental design warrants enhancement, especially in the formulation of test questions. It is crucial to ensure a distinct separation between the test questions and the learning scenarios to maintain validity. Fourth, while the system is designed for beginner-level users of diverse ages, the evaluation phase was limited solely to children, which may not fully represent the intended user base. Lastly, the difficulty of the questionnaire we designed relied too much on certain quantitative criteria without considering students' subjective perceptions, and we did not conduct a pilot test resulting in higher accuracy rates for high-difficulty questions than for medium-difficulty questions.

In the future, we plan to make several improvements to the EK-Chess system. We will focus on the following main areas: interface design, knowledge graph design, user study, and generative language model design. In terms of interface design, we aim to optimize the interaction mechanisms, enhance system stability, and create endgame scenarios. In user study, we will pilot test the difficulty classification in the questionnaire and fully consider users' subjective perceptions of difficulty. We aim to develop more referenceable quantitative criteria and more accurate questionnaires accordingly. For knowledge graph design, we plan to expand the knowledge graph by adding more nodes to deepen and broaden the scope of expert knowledge included in the graph. We will also further evaluate our system with a broader demographic or design more specific interaction/interface for children. Additionally, in future experiments, we hope to establish standards or frameworks for evaluating the learning effectiveness of the system. In data analysis work, we will adopt the method of panel data study due to its unique advantages in understanding temporal dynamics, individual behavior, and causal relationships.

7. Conclusion

We propose a chess teaching system, EK-Chess, based on a knowledge graph, was designed using expertise from professional chess experts, emphasizing the explanation of expert knowledge while guiding users in their moves. Through user experiments and data analysis of ability assessments, learning time, and response times, we found that the EK-Chess system is effective in helping users learn chess knowledge by incorporating expert knowledge into the teaching process through the knowledge graph. Knowledge graphs as a form of structured human knowledge, combined with our designed learning record mechanism, the system achieves educational goals for different moves within the same chess scenario, allowing users to understand the evolution of chess endgame positions from multiple perspectives.

Our work has made several contributions:

- CEKG: We constructed a knowledge graph for chess endgames based on expert knowledge. This knowledge graph was created by extracting chess game information from professional books, combining game, and move information. It was stored using the graph database Neo4j (Miller, 2013).
- Development of EK-Chess: EK-Chess, a chess teaching system, was developed to read and utilize the constructed chess knowledge graph. It also integrated the chess engine Stockfish, combining the chess engine with a knowledge graph-based teaching system to provide high-quality instructional feedback.
- Comprehensive User Evaluation: We conducted a quantitative analysis of user data, providing a comprehensive assessment. EK-Chess was found to be user-friendly for children due to its simple operation and convenient interaction. The system deduced various move suggestions from the knowledge graph, offering users a more comprehensive learning experience. Our research also explored the learning effectiveness of the system in different chess scenarios.

Acknowledgments

We would like to thank our participants for generously contributing their time and valuable insight. We also appreciate the constructive feedback and suggestions provided by top-level chess experts and chess teachers, which helped improve this work.

Disclosure statement

There is no conflict of interest among the authors.

Funding

This work is supported by the [Natural Science Foundation of Beijing & Key project of Science and Technology Plan of Beijing Municipal Education Commission] under Grant [No. 21JD0044]; [National Natural Science Foundation of China] under Grant [No. 61872363]; the Beijing Natural Science Foundation under Grant No. L232098; and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000.

ORCID

Mingyu Zhang http://orcid.org/0009-0001-5224-0482
Qiao Jin http://orcid.org/0000-0001-5493-1343
Danli Wang http://orcid.org/0000-0002-8661-5588

References

Barak, O., & Nishith, K. (2015). *ConvChess*. github.com. https://github.com/BarakOshri/ConvChess

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., … Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Bundy, A., & Wallen, L. (1984). Breadth-first search. *Catalogue of Artificial Intelligence Tools*, 13–13.

Campbell, M., Hoane, A. J., & Hsu, F. (2002). Deep blue. *Artificial Intelligence*, *134*(1–2), 57–83. https://doi.org/10.1016/S0004-3702(01)00129-1

Chess Compass. (n.d.). *Chess Compass*. https://www.chesscompass.com/

Chiang, W. L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., Xing, E. P. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://vicuna.lmsys.org (accessed 14 April 2023), *2*(3), 6.

Cierniak, G., Scheiter, K., & Gerjets, P. (2009). Explaining the split-attention effect: Is the reduction of extraneous cognitive load accompanied by an increase in germane cognitive load? *Computers in Human Behavior*, *25*(2), 315–324. https://doi.org/10.1016/j.chb.2008.12.020

Cirik, V., Morency, L. P., Hovy, E. (2015). Chess Q&A: Question answering on chess games. *Reasoning, attention, memory (RAM) workshop* (Vol. 7, 7.6) Neural Information Processing Systems.

Cui, J., Li, Z., Yan, Y., et al. (2023). Chatlaw: Open-source legal large language model with integrated external knowledge bases. arXiv preprint arXiv:2306.16092.

DecodeChess. (2022). *Chess Analysis, Powered by AI*. decodechess.com. https://decodechess.com/

DeFraine, W. C. (2016). Differential effects of cognitive load on emotion: Emotion maintenance versus passive experience. *Emotion*, *16*(4), 459–467. https://doi.org/10.1037/emo0000140

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., & Zhang, W. (2014). *Knowledge vault: A web-scale approach to probabilistic knowledge fusion in SIGKDD* (pp. 601–610). ACM.

Edwards, S. J. (1994). *Standard portable game notation specification and implementation guide*. Dosegljivo. http://www.saremba.de/chessgml/standards/pgn/pgncomplete.htm

Gameknot. (n.d.). *WELCOME TO GAMEKNOT! THE PREMIER ONLINE CHESS BATTLEGROUND*. gameknot. https://gameknot.com/

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018, May). Recent advances in convolutional neural networks. *Pattern Recognition*, *66*, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013

Gupta, P., Puri, N., Verma, S., Kayastha, D., Deshmukh, S., Krishnamurthy, B., & Singh, S. (2020, April). Explain your move: Understanding agent actions using specific and relevant feature attribution. In *International Conference on Learning Representations (ICLR)*.

Heist, N., Hertling, S., Ringler, D., & Paulheim, H. (2020). *Knowledge graphs on the web–an overview* (pp. 3–22). IOS Press.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngonga Ngomo, A. C., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2020). Knowledge graphs. *ACM Computing Surveys (Csur)*, *54*(4), 1–37.

Hooper, D., & Whyld, K. (1996). [First pub. 1992]. Standard notation. *The Oxford companion to chess* (2nd ed., p. 389). Oxford University Press. ISBN 0-19-280049-3.

Huck, S. W., & McLean, R. A. (1975). Using a repeated measures ANOVA to analyze the data from a pretest-posttest design: A potentially confusing task. *Psychological Bulletin*, *82*(4), 511–518. https://doi.org/10.1037/h0076767

Jhamtani, H., Gangal, V., Hovy, E., Neubig, G., & Berg-Kirkpatrick, T. (2018, July). Learning to generate move-by-move commentary for chess games from large-scale social forum data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1661–1671). https://doi.org/10.18653/v1/P18-1154

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(2), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

Kamlish, I., Chocron, I. B., & McCarthy, N. (2019). Sentimate: Learning to play chess through natural language processing. arXiv preprint arXiv:1907.08321.

Kandpal, N., Deng, H., Roberts, A., Wallace, E., & Raffel, C. (2023, July). Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning* (pp. 15696–15707). PMLR.

Kerner, Y. (1995). Case-based evaluation in computer chess. In J. P. Haton, M. Keane, & M. Manago (Eds.), *Advances in case-based reasoning* (pp. 240–254). Springer.

Knuth, D. E., & Moore, R. W. (1975). An analysis of alpha-beta pruning. *Artificial Intelligence*, *6*(4), 293–326. https://doi.org/10.1016/0004-3702(75)90019-3

Ko, H., Lee, S., Park, Y., & Choi, A. (2022). A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics*, *11*(1), 141. https://doi.org/10.3390/electronics11010141

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, *35*, 22199–22213.

Leela Chess Engine. (2023). *Leela Chess Zero*. chessprogramming.org. https://www.chessprogramming.org/Leela_Chess_Zero

Liu, Q., Li, Y., Duan H., Liu, Y., & Qin, Z. (2016). Knowledge graph construction techniques. *Journal of Computer Research and Development*, *53*(3), 582–600. https://doi.org/10.7544/issn1000-1239.2016.20148228

Martin, B. (2019). *Arena Chess GUI*. https://www.playwitharena.de/

Matuszek, C., Witbrock, M., Cabral, J., & DeOliveira, J. (2006). An introduction to the syntax and content of Cyc. *AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, Stanford, CA, March 2006 (pp. 1–6).

McIlroy-Young, R., Sen, S., Kleinberg, J., & Anderson, A. (2020, August). Aligning superhuman ai with human behavior: Chess as a model system. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1677–1687).

Miller, G. A. (1995). WordNet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41. https://doi.org/10.1145/219717.219748

Miller, J. J. (2013, March). Graph database applications and concepts with Neo4j. In *Proceedings of the Southern Association for Information Systems Conference*, Atlanta, GA, USA (Vol. 2324, No. 36, pp. 141–147).

Newell, A., Shaw, J. C., & Simon, H. A. (1959, June). Report on a general problem solving program. In *IFIP Congress* (Vol. 256, p. 64).

Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, *104*(1), 11–33. https://doi.org/10.1109/JPROC.2015.2483592

Shortliffe, E. (2012). *Computer-based medical consultations: MYCIN* (Vol. 2). Elsevier.

Shredder. (n.d.). *Shredder*. shredderchess.com. https://www.shredderchess.com/

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140–1144. https://doi.org/10.1126/science.aar6404

Stockfish. (2023). *Stockfish*. stockfishchess.org. https://stockfishchess.org/

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499

Wang, X., Wang, D., Gao, X., Zhao, Y., & Chiu, S. C. (2024). Enhancing EEG-based decision-making performance prediction by maximizing mutual information between emotion and decision-relevant features. *IEEE Transactions on Affective Computing*, 1–13. https://doi.org/10.1109/TAFFC.2023.3329526

Wang, X., Yang, Q., Qiu, Y., Liang, J., He, Q., Gu, Z., Xiao, Y., & Wang, W. (2023). KnowledGPT: Enhancing large language models with retrieval and storage access on knowledge bases. arXiv preprint arXiv:2308.11761.

Wikipedia. (2023a). *Chess engine [DB/OL]*. https://en.wikipedia.org/wiki/Chess_engine.

Wikipedia. (2023b). *Computer chess [DB/OL]*. https://en.wikipedia.org/wiki/Computer_chess.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

Xie, J., & Lin, F. (2010). *Chess endgame strategy*. Jindun Press.

Zhao, Y., Wang, D., Wang, X., & Chiu, S. C. (2022). Brain mechanisms underlying the influence of emotions on spatial decision-making: An EEG study. *Frontiers in Neuroscience*, *16*, 989988. https://doi.org/10.3389/fnins.2022.989988

About the authors

Mingyu Zhang is a master student in artificial intelligence at University of Chinese Academy of Science, China. His research interests include human–computer interaction, natural language processing and machine learning.

Qiao Jin is a PhD candidate from Grouplens Research Center at the University of Minnesota – Twin Cities. Her research focuses on using AR/VR/MR to support remote education, collaboration, and social connection.

Dong Qian is a lecturer at School of Physical Education, Shanghai University of Sport. She received her Ph.D. degree in Peking University. She is national first-class athletes in chess. Her research interest includes educational technology, instructional design in Physical education, and chess education.

Danli Wang is a professor at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. She received her Ph.D. degree in Beihang University. Her research interests include human–computer interaction, user experience, artificial intelligence, physiological computation, and group recommendation.

Jun Xie serving as vice president of Capital University of Physical Education and Sports currently. She received her Ph.D. degree in Beijing Normal University. She has won four times world champion of women's chess. Her research interests include cognitive psychology, Olympic movement, leisure sports, and intelligence movement.