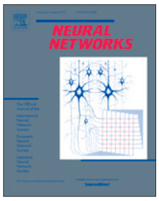
Neural Networks 155 (2022) 95–118

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Explaining *Aha!* moments in artificial agents through IKE-XAI: Implicit Knowledge Extraction for eXplainable AI

Ikram Chraibi Kaadoud [a], Adrien Bennetot [b,c,d], Barbara Mawhin [e], Vicky Charisi [f], Natalia Díaz-Rodríguez [g,*]

[a] IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
[b] U2IS Dept., ENSTA, Institut Polytechnique Paris, Inria Flowers Team, 828, Boulevard des Maréchaux 91762 Palaiseau Cedex, France
[c] Segula Technologies, Parc d'activité de Pissaloup, Trappes, France
[d] Institut des Systèmes Intelligents et de Robotique, Sorbonne Université, Paris, France
[e] Human Factors Department, EBT-Salient Aero Foundation, Spain
[f] European Commission, Joint Research Center (JRC), Seville, Spain
[g] Department of Computer Science and Artificial Intelligence, DaSCI Andalusian Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 13 December 2021
Received in revised form 21 July 2022
Accepted 1 August 2022
Available online 6 August 2022

Keywords:
Explainable AI
Reinforcement learning
Cognitive modeling
Developmental robotics
Post-hoc rule extraction
Knowledge extraction

ABSTRACT

During the learning process, a child develops a mental representation of the task he or she is learning. A Machine Learning algorithm develops also a latent representation of the task it learns. We investigate the development of the knowledge construction of an artificial agent through the analysis of its behavior, i.e., its sequences of moves while learning to perform the Tower of Hanoï (TOH) task. The TOH is a well-known task in experimental contexts to study the problem-solving processes and one of the fundamental processes of children's knowledge construction about their world. We position ourselves in the field of explainable reinforcement learning for developmental robotics, at the crossroads of cognitive modeling and explainable AI. Our main contribution proposes a 3-step methodology named Implicit Knowledge Extraction with eXplainable Artificial Intelligence (IKE-XAI) to extract the implicit knowledge, in form of an automaton, encoded by an artificial agent during its learning. We showcase this technique to solve and explain the TOH task when researchers have only access to moves that represent observational behavior as in human–machine interaction. Therefore, to extract the agent acquired knowledge at different stages of its training, our approach combines: first, a Q-learning agent that learns to perform the TOH task; second, a trained recurrent neural network that encodes an implicit representation of the TOH task; and third, an XAI process using a post-hoc implicit rule extraction algorithm to extract finite state automata. We propose using graph representations as visual and explicit explanations of the behavior of the Q-learning agent. Our experiments show that the IKE-XAI approach helps understanding the development of the Q-learning agent behavior by providing a global explanation of its knowledge evolution during learning. IKE-XAI also allows researchers to identify the agent's *Aha!* moment by determining from what moment the knowledge representation stabilizes and the agent no longer learns.

1. Introduction: a multi-disciplinary context for cognitive modeling

Knowledge development and its evolution in humans and artificial agents is a question that can be addressed from (i) the cognitive science perspective and (ii) the Artificial Intelligence (AI) perspective, more specifically, in open-ended learning (Doncieux et al., 2020, 2018) and the developmental robotics field (Lungarella, Metta, Pfeifer, & Sandini, 2003). In this introduction, we will present the multidisciplinary background of this work that aims to explain knowledge development within artificial agents by taking inspiration from the following areas: (1) Knowledge development in humans for problem-solving, (2) Reinforcement Learning (RL) in developmental robotics, and (3) eXplainable AI, the field that is rapidly evolving at the intersection of AI and human AI interaction (see Fig. 1).

[*] Corresponding author.
E-mail addresses: ikram.chraibi-kaadoud@imt-atlantique.fr (I. Chraibi Kaadoud), nataliadiaz@ugr.es (N. Díaz-Rodríguez).

https://doi.org/10.1016/j.neunet.2022.08.002
0893-6080/

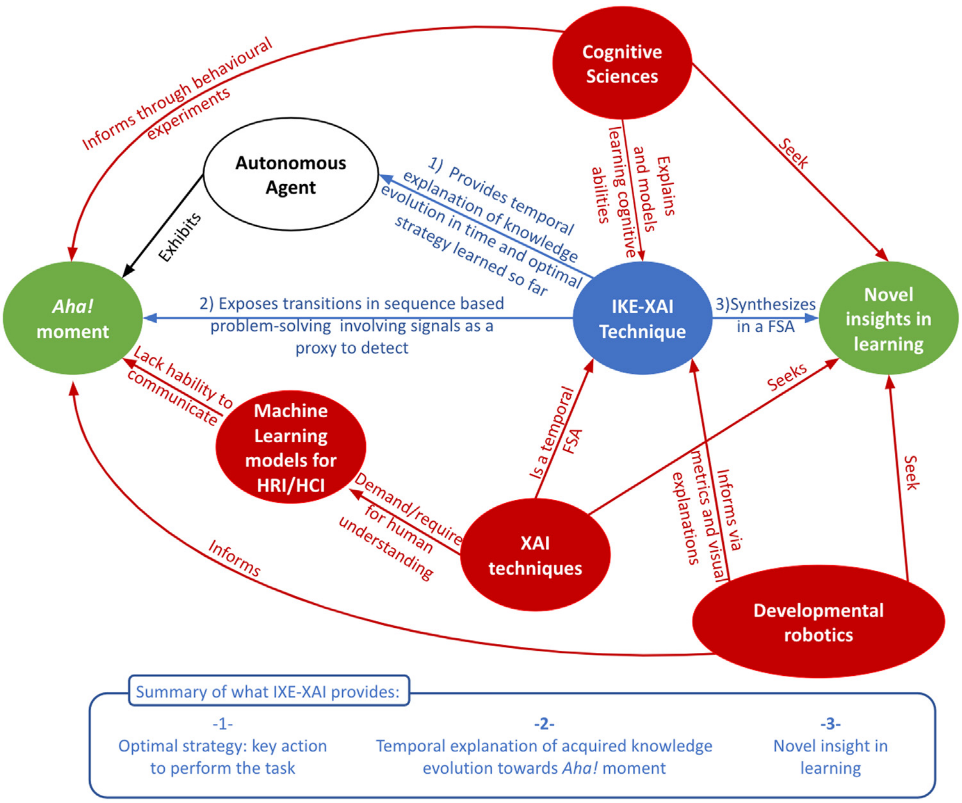

Cognitive Sciences
Informs through behavioural experiments
Autonomous Agent
Exhibits
1) Provides temporal explanation of knowledge evolution in time and optimal strategy learned so far
Explains and models learning cognitive abilities
Seek
Aha! moment
2) Exposes transitions in sequence based problem-solving involving signals as a proxy to detect
IKE-XAI Technique
3)Synthesizes in a FSA
Novel insights in learning
Lack hability to communicate
Machine Learning models for HRI/HCI
Is a temporal FSA
Seeks
Informs via metrics and visual explanations
Seek
Demand/require for human understanding
XAI techniques
Informs
Developmental robotics
Summary of what IXE-XAI provides:
-1- Optimal strategy: key action to perform the task
-2- Temporal explanation of acquired knowledge evolution towards Aha! moment
-3- Novel insight in learning

Fig. 1. Conceptual summary of current works and related fields: Knowledge graph of all contributed elements (in green) to explain sequential problem-solving and domain disciplines (in red) that inform IKE-XAI method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1.1. *Problem-solving task modeling: Knowledge development in humans and the acquisition of expertise*

The development of knowledge in humans has been extensively studied in the field of cognitive science as one of the phenomena that includes the construction of human mental representation of the world through their interaction with it. Knowledge can be acquired through explicit learning (the acquisition of explicit knowledge – easier to access and verbalize – using attentional resources (Hélie, Proulx, & Lefebvre, 2011)) and implicit learning (the acquisition of expertise through experience, in an incidental manner, without awareness of what has been learnt) (Ettlinger, Margulis, & Wong, 2011). Individuals who are repeatedly subjected to observed temporal regularities (i.e., variants or rules), eventually, extract, learn and include them immediately in their reasoning and their interactions with the environment without being aware of it. Therefore, implicit learning mechanisms producing improved performance during rehearsals (Cleeremans, Servan-Schreiber, & McClelland, 1989; Reber, 1967) lead to consider behavioral properties as an indicator of the problem-solving process and, in particular, of the reconstruction of the problem representation, namely the *Aha!* moment or problem insight (Charisi, Díaz-Rodríguez, Mawhin, & Merino, 2022).

For children, the knowledge development of a mental representation during learning to perform a task was studied by Piaget with his theory of cognitive development (Lefa, 2014) who assumes that during knowledge acquisition, learners gradually construct schemata about the world to help them understand and interpret it. A schema is defined as "*different sensory motor maps that the learner constructs about the world in their knowledge development*" (Lefa, 2014) and it is constructed by involving mental and physical actions. A schema includes both, the category of knowledge and the process to obtain that knowledge. Piaget describes the learner as actively engaged towards cognitive equilibrium, i.e., an ongoing process towards a state of balance between their mental schemata and their environment which refines and transforms mental structures. This adaptation relies on two dynamic processes continuously interacting: assimilation, i.e., the addition of a piece of information to existing cognitive structures and accommodation, i.e., the update of those structures in case of contradictions or conflicts between them and the information so that the learner can deal with new knowledge. There is thus a continuous update of a child's schema; the evolution of a child's behavior reflects these changes and the emergence various psychological structures or patterns of thinking that influence how children interpret information (Charisi, Liem, & Gomez, 2018; Lefa, 2014; Poissant, Poëllhuber, & Falardeau, 1994). As a result, one of the techniques cognitive scientists use to study children's knowledge development is with the use of problem-solving tasks that stimulate the above-mentioned processes.

A problem-solving task can be defined by three characteristics (Mayer, 1977; Poissant et al., 1994): (a) an initial state where the problem begins with a starting situation that is deemed unsatisfactory; (b) an objective state, where the desired situation is different from the initial situation and where a reflection is necessary to transform the initial state; (c) several obstacles that make the way to get from the initial state to the objective state not obvious and previously unknown. A problem can therefore be conceptualized as a difference between a current situation and the desired situation. Among the categories of problem-solving tasks, in the current work we will address the transformation problems category. We refer the reader to Poissant et al. (1994) for further details about problem-solving tasks.

Transformation problems have initial and final states both clearly defined and the relationships between elements of the problem are known in the initial and final states. The difficulty lies in the way to go from one state to the other. Generally, the more the players try to solve a problem, the more they understand the required strategies. This is the principle of transformation of knowledge and development of expertise from beginner to expert through practice (Hoffman, 2014; Kim, Ritter, & Koubek, 2013). At the cognitive level, the acquisition of expertise is due to the evolution from explicit to implicit knowledge (Reber, 1967). However, in some cases, the players can also fail the task several times, and try again until they suddenly have an *Aha!* moment and find a solution which is later further optimized. The *Aha!* moment is identified as a solution that appears suddenly thanks to intuition (Chronicle, MacGregor, & Ormerod, 2004; Kounios & Beeman, 2014) and transforms the processing of the problem: when a solution arises, the information that was difficult to process can be processed more fluidly (Topolinski & Reber, 2010). This experience is also called "insight problem-solving" or "insight problem" (Smith, 1995) and can be conceptualized as a two-phase process (Chronicle et al., 2004): (i) the fruitless search/blocking phase in the face of a situation or questioning, (ii) the sudden emergence of a solution after a mental pause or re-evaluation of the problem. Therefore, the *Aha!* moment often appears to be part of the transition from tacit knowledge (appearing in the exploration phase) to explicit knowledge (appearing in the exploitation phase) (Charisi et al., 2022).

Insight problem-solving is studied with the use of tasks that tend to evoke this type of solution experience (Chu & MacGregor, 2011) such as Tower of Hanoï (TOH) which is a prototype task in the category of transformation problems. The TOH is widely used in cognitive science to assess high-executive functions which include the following processes: (1) the player needs to experiment and try many strategies (sequences of moves) before discovering how to perform the task efficiently, and (2) the player needs to anticipate the future state of the task to allow him or her to consider the next moves. As a well-defined problem, easily administered, and brief, TOH represents an experimental context to evaluate the expertise acquisition in humans and artificial agents (Bennetot, Charisi, & Díaz-Rodríguez, 2020; Charisi, Gomez, Mier, Merino, & Gomez, 2020; Edwards, Downs, & Davidson, 2018).

During the task performance, implicit heuristic cognitive processes in the form of exploration function as a means for the child to identify that the use of an auxiliary movement (inhibition[1]) leads to the optimal solution of the task. Typically, the use of the auxiliary movement happens initially randomly, during spontaneous and exploratory actions, as a result of local heuristics to move the disk to an unoccupied peg (Klahr & Robinson, 1981; Welsh, 1991). After several attempts, the child realizes that it is the use of the strategy of inhibition that leads to the optimal solution. From that moment onward, a more consistent use of the inhibition strategy is observed which means that the learning has occurred. The behavioral onset of the sudden realization of the strategy of inhibition in the TOH task has been identified as the moment of insight or the *Aha!* moment, Awareness of this kind of representational change, though abrupt, takes place after a period of unconscious processing (Van Steenburgh, Fleck, Beeman, & Kounios, 2012). Children's behavioral patterns of solving the TOH task with a consistent use of inhibition after the *Aha!* moment at a faster pace, and with the probable exhibition of affective engagement due to the positive reward of the process, help us identify the moment of insight. However, since young children lack the meta-cognitive skills to explain their behavior, further research and understanding are needed on this phenomenon. A detailed description of the TOH, as a discussion about its characteristics for studying insight in problem-solving tasks with children, can be found in Appendix B.

It is important to highlight the modeling of the *Aha!* moment in artificial agents is an open scientific question. To our knowledge, there has been relatively little work considering insight from an artificial agent perspective (Colin & Belpaeme, 2019) and even less from a developmental robotics perspective. The work presented in this paper contributes to bridge this gap.

1.2. Reinforcement Learning (RL) for cognitive modeling: Developmental robotics, the design of autonomous cognitive capabilities for artificial agents inspired by children

Cognitive science examines the nature and function of human cognition and as such, it keeps inspiring AI approaches. In this work, we are interested in computational methods that widely study the knowledge acquisition and development in humans in a similar spirit to different Machine Learning (ML) approaches (Chraibi Kaadoud, Rougier, & Alexandre, 2022; Donnarumma, Maisto, & Pezzulo, 2016; Heuillet, Couthouis, & Díaz-Rodríguez, 2020; Lake, Ullman, Tenenbaum, & Gershman, 2017; Lesort, Rodríguez, Goudou, & Filliat, 2018; Madumal, Miller, Sonenberg, & Vetere, 2020; Puiutta & Veith, 2020; Servan-Schreiber, Cleermans, & McClelland, 1988; Yuan, Xiang, Crandall, & Smith, 2020), and robotics ones (Alexandre, Hinaut, Rougier, & Viéville, 2021; Cangelosi & Schlesinger, 2018; Colas, Karch, Sigaud, & Oudeyer, 2021; Lungarella et al., 2003).

Developmental robotics (Cangelosi & Schlesinger, 2018) – and synonyms *Cognitive developmental robotics*, *Autonomous mental development* as well as *Epigenetic robotics* – studies the autonomous design of behavioral and cognitive capabilities in artificial agents (AAs) with, among others, RL approaches. RL is a field of ML that focuses on how artificial agents undertake actions in an environment through the search for a balance between exploration (e.g., of uncharted territory) and exploitation (e.g., of current knowledge of sources of reward). This field addresses the question of designing autonomous agents that can evolve through experience and interaction (Sutton, Barto, et al., 1998). Developmental robotics is thus a field intrinsically interdisciplinary that directly draws inspiration from developmental principles and mechanisms observed in children's natural cognitive abilities to design AAs that learn to explore and interact with the world (Cangelosi & Schlesinger, 2018; Lungarella et al., 2003).

In Bennetot et al. (2020) the work of Charisi et al. (2020) is extended to test whether robotic learning and AA learning processes can be inspired by child development. They worked in the context of a child–robot interaction setting, with child-initiated robotic interventions. Due to its characteristics, the TOH task was used to study the initiation of a request for voluntary help. The authors focus on the analysis of "when" and "why" to ask for help, to study the impact of these questions on the resolution of collaborative tasks in inhibitory processes. Among the contributions of this work, we focus on one aspect of the study: the average number of moves of an AA. The authors used the variation in the number of moves to solve the task during the experiments as an explicit metric to assess changes in the behavior. While training the model, if the average number of moves decreases, the agent becomes more efficient in fulfilling the task since it needs fewer moves. The phenomenon was studied in different experimental contexts, which leads us to consider that there is indeed a trace of the hidden knowledge, i.e., latent representation, of the agent in its sequences of moves. This work is of particular interest because a child might not be able to verbalize the purposes of an action (depending on their age, vocabulary, and cognitive maturity level)

[1] Inhibition movement because moving a disk to a peg where it should not lie in the final state is an illusionary *apparent detour* of what seems to be the most straight forward way to arrive at the final state.

and thus, it seems even more complex to make it explicit for an AA to learn. This rationale motivates the need for explainability strategies that overcome this constraint and bring solutions, on the one hand, to better understand the behaviors of virtual agents and, on the other hand, to compare them and get inspiration from those of children.

1.3. Explainable AI: knowledge extraction from RL models

When it comes to cognitive modeling, one recurrent question concerns the knowledge representation of abstract models that tend to mimic human cognitive functions: "What knowledge do models acquire that can explain the logic and rationale of their behavior in a specific context?". It is important to highlight that the questions of AI models with explanation goals have been questioned and investigated a long time ago, even if the terms eXplainable Artificial Intelligence (XAI) and Interpretable ML have been recently proposed (Chraibi Kaadoud, Fahed, & Lenca, 2021). For example, efforts to explain Recurrent Neural Networks (RNN) and connections models with an internal representation of time (Durand, 1995) that model artificially the cognitive function of implicit learning using sequences, started in the eighties and continues to this day. Many works investigated the extraction of the encoded regularities, also called rules, using rule extraction algorithms (Jacobsson, 2005; Towell & Shavlik, 1993) in the form of a Finite State Automaton (FSA), i.e., a directed graph composed of nodes and transitions (Bhargava & Purohit, 2011; Elman, 1990; Lawrence, Giles, & Fong, 2000; Murdoch & Szlam, 2017; Omlin & Giles, 1996, 2000; Smith, 2003; Strobelt, Gehrmann, Huber, Pfister, & Rush, 2016; Wang, Zhang, II, Xing, Liu, & Giles, 2017; Weiss, Goldberg, & Yahav, 2017; Zeng, Goodman, & Smyth, 1993). Let us note that the majority of these works extract rules in the form of FSA. The domains of rule extraction and FSA generation are thus strongly linked. Among existing related works, we highlight the work of Towell and Shavlik (1993) at the crossroads of symbolic and connectionist approaches in artificial intelligence. This work lays the foundations of the field of knowledge extraction from neural networks by exploiting the advantages of both approaches. First, connectionist models can encode neural representations that take into account the importance of the past and the context. Second, symbolic representations are much more accessible and intelligible by humans, which facilitates knowledge transfer and "human inspection". The authors described a three-step process necessary for extracting rules from neural networks (rules initially in the form of symbolic knowledge): First, the knowledge must be acquired by a neural network through the learning phase. Second, the network must be refined, i.e., the learning must be tuned so that the acquired knowledge is of acceptable quality. Third, the knowledge can then be extracted from the network in the form of symbolic rules. This step is the most difficult, notably due to the complexity of extracting rules that need to be both (i) precise and faithful to the network's behavior, and (ii) "humanly intelligible". For this purpose, they introduce the concepts of Knowledge Based Artificial Neural network and Knowledge based neural network, the latter produced from the former, and propose the MoFN method for efficiently refining symbolic knowledge through neural networks. Their method extracted rules that generalize better to new examples (not seen during the training) than rules produced by "all-symbolic" rule refinement algorithms.

Recently, due to the social and legal context, the performance and accuracy of models are no longer the only criteria for evaluating algorithms. Their transparency has become an additional criterion that is now essential, especially for end-users (Weitz, Schiller, Schlagowski, Huber, & André, 2021), government and legal authorities (Truby, Brown, Ibrahim, & Parellada, 2022). Therefore, many advances are made in these fields of interpretable ML and XAI (Gilpin, Bau, Yuan, Bajwa, Specter, & Kagal, 2018; Guidotti, Monreale, Ruggieri, Turini, Giannotti, & Pedreschi, 2018). *Interpretability* is the ability to break down all inner mechanisms of a black box (without necessarily understanding them) (Doshi-Velez & Kim, 2017); whereas an explainable Artificial Intelligence (XAI) system produces details or reasons to clarify its functioning and ease its understanding, for a given particular audience (Arrieta et al., 2020). Explainability is strongly linked to the concept of "explanation" as an interface between the data, the AI model, and the targeted audience (Arrieta et al., 2020; Chaput, Cordier, & Mille, 2021; Guidotti et al., 2018). Defined as "*an information in a semantically complete format, which is self-sufficient and chosen according to the target audience regarding its knowledge, its expectations, and the context*" (Chraibi Kaadoud et al., 2021), an explanation is oriented to a target-audience. Its goal is to clarify the context, causes, and consequences of facts through a set of statements or additional information (Van Fraassen, 1988). The evaluation of an explanation is inherently contextual, often subjective, and linked to the target audience and its level of expertise about the issue studied.

1.4. Positioning and main contributions

Many works have shown that within humans, the study of exhibited behavior (performance evolution for example) can lead to understanding the implicit knowledge behind that behavior (Mix, Bower, Hancock, Yuan, & Smith, 2022; Schulz, 2012; Siegel, Magid, Pelz, Tenenbaum, & Schulz, 2021; Smith & Yu, 2008). Other works (Breazeal & Scassellati, 2002; Hussein, Gaber, Elyan, & Jayne, 2017; Schillaci, Hafner, & Lara, 2016) have focused on mimicking the behavior of natural agents using artificial ones (Colin, 2020; Oudeyer, Kaplan, & Hafner, 2007). Interpretable and explainable techniques were used to get hints about the reason behind a specific behavior. Our work is in line with all these studies and fits in the context of neural-symbolic learning methodology (X-NeSyL) (Díaz-Rodríguez et al., 2022), as it has two out of the three key components (neural and symbolic processing modules). This methodology blends neural and symbolic (in our case a FSA) components to provide more interpretable model outcomes.

To focus our study on problem-solving tasks of transformation problems and the associated implicit knowledge development within an AA, we define knowledge here as a set of facts, information, and skills acquired through experience by the AA that contribute to gaining a theoretical or practical understanding of a subject or the world. To do so, we get inspired from works made in Developmental robotics (Bennetot et al., 2020) and cognitive modeling (Chraibi Kaadoud et al., 2022) fields. The selected use case is the task of the TOH. Our research questions (RQs) are associated with the following hypotheses:

1. *H1*: The sequences of movements of an AA reflect and contain an implicit trace of the knowledge acquired by it as it learns the TOH task. This trace contains the best moves to reach the final state and the knowledge of the possible moves to reach the desired final state.
2. *H2*: The sequences of moves recorded at different stages of learning show the AA knowledge acquisition and development and, by extension, the transformation of its expertise in solving the task.
3. *H3*: Each sequence of movements is governed by implicit rules that can be extracted in the form of a Finite State Automaton.

Therefore, our objective is twofold, i.e., being able to identify the *Aha!* moment on both sides: for an AA to know when it stabilizes its knowledge, and for the model developer to identify

Table 1
Acronyms used in this paper.

Acronym	Meaning	Acronym	Meaning
XAI	eXplainable Artificial Intelligence	AA	Artificial Agent
IKE-XAI	Implicit Knowledge Extraction with XAI	RNN	Recurrent Neural Network
ML	Machine Learning	TOH	The Tower of Hanoï
SRL	State Representation Learning	TOH states	States of the TOH task (e.g., "111" or "321")
NN	Neural Network		
LSTM	Long Short Term Memory	LSTM states	Implicit activation states of the LSTM RNN
RL	Reinforcement learning	FSA	Finite State Automaton

and understand the optimized learning stage of the AA. Our main contribution consists of proposing a new methodology for Implicit Knowledge Extraction with XAI (IKE-XAI) to extract the implicit knowledge, in form of an automaton, encoded by an AA during training to perform a task. This methodology is applicable when researchers have only access to moves that represent observational behavior as in human–machine interaction. Specifically, we propose: (1) the IKE-XAI methodology to explain the learning agent training process and (2) an explanation interface of the evolution of the AA knowledge (i.e., latent representation) construction for better identifying their *Aha!* moment. By using the TOH task as the case study to showcase and validate the IKE-XAI methodology, we propose a novel approach that can be extended to transformation problems. Finally, as an explainable process needs to always be adapted for a target audience (Arrieta et al., 2020), we specify that IKE-XAI proposes explanations directed towards researchers in the XAI field interested in the study of explaining the learning of AAs, as well as towards the Developmental robotics community (Cangelosi & Schlesinger, 2018; Lungarella et al., 2003). Let us specify that to interpret the extracted FSA and their evolution, the target audience does not need any specific technical knowledge (e.g., LSTM computational principles), only the context of the task performed by the AA.

The ultimate goal of this work aims at having the potential to inform studies that explore children's problem-solving processes and the emergence of knowledge in tasks beyond the use case of the TOH task. In addition, identifying the process of implicit learning in AA and the transformation of implicit into explicit knowledge might give us insights on how to effectively support this phenomenon in young children. This is particularly relevant from a methodological perspective, given that often in early childhood the required cognitive tools and verbal capacities for children's meta-cognitive actions are still developing. Lastly, from a child-AI interaction perspective, this research might contribute towards the development of child-friendly explainable systems. At the same time, promoting transparency in the interaction can also effectively support children's decision-making processes.

We organize the current paper as follows: Section 2 presents related works in the field of explainable RL. In Section 3, we introduce the IKE-XAI methodology as a 3-step XAI process to extract, visualize and explain the acquired knowledge of an AA and how to evaluate its evolution. In Section 4, we present our results and finally discuss the outcomes of this study in Section 5, before concluding with prospective work in Section 6. Fig. 1 presents a conceptual summary of current works and related fields in the shape of a knowledge graph of all contributed elements to explain sequential problem-solving and domain disciplines that inform IKE-XAI method, and Table 1 presents all the acronyms used in this work and their meanings.

2. Related work: Explainable RL

Trust in human–robot interaction is mainly based on the intelligibility of the reasons behind a robot's action and the involved factors (Wells & Bednarz, 2021). XAI research in the context of ML and Deep Learning aims to explain the inner mechanisms of black-box models and provide information explaining why a model behaves the way it does (Arrieta et al., 2020; Heuillet et al., 2020). When it comes to provide explainability to the specific field of RL, we find several challenges (Wells & Bednarz, 2021): (1) Including time as part of the explanation: Compared to standard ML techniques where decisions can happen in isolation or are unrelated to each other and static in time, RL explanations can involve a large number of decisions made over time, and often aim at predicting next action to take in real-time. Explanations for AA will generally need to encompass actions or a set of actions that are related in some way, (2) Requirement to perform continual learning (Lesort, Lomonaco, Stoian, Maltoni, Filliat, & Díaz-Rodríguez, 2020) or not knowing the continuous stream data to learn from in advance: Except when the human-replay data is used (Vinyals, Ewalds, Bartunov, Georgiev, Vezhnevets, Yeo, et al., 2017), RL agents learn from a continuous streams of data that may not be able to face again in the future, data that get discarded, and the model should consolidate knowledge to better deal with similar data if it is presented in the future. Generating human-readable explanations is thus challenging since learning rather happens from the execution of actions in an environment supported by a feedback loop, based on observations not known in advance.

The majority of papers in XAI applied to explain RL models focused on application domains of video games (Wells & Bednarz, 2021) (e.g., Atari games and Pac-Man), grid-world environments (Heuillet, Couthouis, & Díaz-Rodríguez, 2022) with navigation tasks and robotics (Wallkötter, Tulli, Castellano, Paiva, & Chetouani, 2021). They also put in light, which is confirmed by the reviewed literature for this work, that in the field of RL, there are many ways to explain the learning of AAs (Heuillet et al., 2020). Among them, we can indicate various categories of works: (1) focusing on explaining AA policies (Verma, Murali, Singh, Kohli, & Chaudhuri, 2018), strategies (Amitai & Amir, 2021; Lapuschkin, Wäldchen, Binder, Montavon, Samek, & Müller, 2019), and reward functions (Huang, Held, Abbeel, & Dragan, 2019; Lage, Lifschitz, Doshi-Velez, & Amir, 2019), (2) designing explainable agents that can provide on request the reasons for its behavior (Langley, Meadows, Sridharan, & Choi, 2017), (3) focusing on the causal model behind the AA behavior through the knowledge of how its actions impact the environment (Madumal et al., 2020), (4) using a query-based approach to extract explanations from the AA (Hayes & Shah, 2017; Kazak, Barrett, Katz, & Schapira, 2019; Sridharan & Meadows, 2019) and (5) focusing on explaining the AA behavior such as Amir and Amir (2018), Engelhardt, Lange, Wiskott, , and Konen (2021), Sun and Sessions (2000), Wang, Gou, Shen, and Yang (2018a). The last one is the approach we follow in this paper.

One particular paradigm of XAI that studies how to extract useful information from RL models is State Representation Learning (SRL), as a particular case of feature learning in which the features to learn are *low dimensional, evolve through time, and are influenced by actions or interactions* (Lesort et al., 2018). This field is thus particularly suited for the RL field, where an AA learns to perform a task through interactions with its environment. SRL algorithms learn representations that capture the variation in the

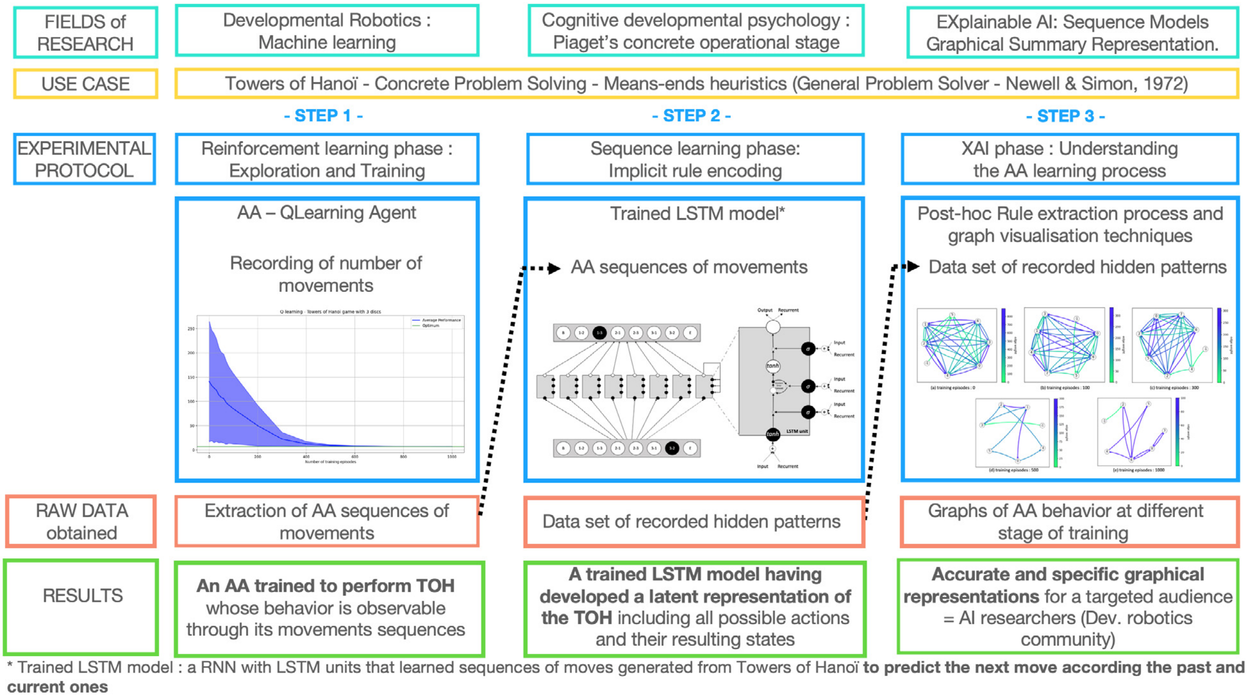

FIELDS of RESEARCH
Developmental Robotics : Machine learning
Cognitive developmental psychology : Piaget's concrete operational stage
EXplainable AI: Sequence Models Graphical Summary Representation.
USE CASE
Towers of Hanoï - Concrete Problem Solving - Means-ends heuristics (General Problem Solver - Newell & Simon, 1972)
- STEP 1 -
- STEP 2 -
- STEP 3 -
EXPERIMENTAL PROTOCOL
Reinforcement learning phase : Exploration and Training
Sequence learning phase: Implicit rule encoding
XAI phase : Understanding the AA learning process
AA – QLearning Agent
Recording of number of movements
Trained LSTM model*
AA sequences of movements
Post-hoc Rule extraction process and graph visualisation techniques
Data set of recorded hidden patterns
RAW DATA obtained
Extraction of AA sequences of movements
Data set of recorded hidden patterns
Graphs of AA behavior at different stage of training
RESULTS
An AA trained to perform TOH whose behavior is observable through its movements sequences
A trained LSTM model having developed a latent representation of the TOH including all possible actions and their resulting states
Accurate and specific graphical representations for a targeted audience = AI researchers (Dev. robotics community)
* Trained LSTM model : a RNN with LSTM units that learned sequences of moves generated from Towers of Hanoï to predict the next move according the past and current ones

Fig. 2. Experimental design of the IKE-XAI methodology to make explicit the process of the autonomous agent (AA) knowledge construction in three steps:
STEP (1) RL Phase: a Q-learning agent learns to perform the TOH task. At several stages of the learning process, the training process is suspended to make a recording of the AA's move sequences while it plays after learning. This step obtains: (a) sequences of moves and (b) an AA trained to perform TOH whose behavior is observable through its sequences of moves, to inform the solution chosen by the AA to reach the solution state (i.e., sequences of moves).
STEP (2) Moves Sequence Learning Phase: the recorded sequences of moves of the AA are fed to train an LSTM to predict the AA's next move at time t based on the current and past ones. This step returns a dataset of recordings of hidden patterns (i.e., the activity vectors of the hidden layer generated by the network at each input). The trained LSTM model had encoded an implicit representation of the TOH rules due to the learned sequences. Let us note that the trained LSTM model is trained on sequences generated from the TOH abstract representation (Fig. B.13).
STEP (3) XAI Phase: a post-hoc implicit rule extraction algorithm and a graph visualization technique are applied to the dataset of recorded hidden patterns to extract graphs of AA behavior at different stages of training.

environment generated by the agent's action. This approach is particularly suitable for robotics (Lesort et al., 2018) and can be used to explain the behavior of the agent. Among the reviewed literature, some works share characteristics with the current one and are discussed in Appendix A.

3. Implicit knowledge extraction for eXplainable AI (IKE-XAI) methodology: extracting automata to synthesize and explain the evolution of learning

Our work focuses on studying the behavioral changes of an AA, and to do so, the IKE-XAI methodology combines: (1) a RL approach where an AA learns to perform a task (in this use case through a Q-learning algorithm), (2) a ML approach for sequence learning, and (3) an XAI approach, as an interface, whose role is to make explicit the knowledge learned by the AA, in the form of an automaton to explain its evolution.

In this section, we present these three approaches as the three steps of the proposed method and protocol. We will explain with particular attention the concept of "state" and the difference between states of an LSTM and those of the TOH task. Fig. 2 presents the experimental design of the IKE-XAI 3-step approach to explain and make explicit the AA behavior. For further technical details, Appendix C describes parameters and algorithms at each step.

3.1. Step 1: RL with Q-learning algorithm

As in Bennetot et al. (2020), we are investigating the behavior changes of an AA, and to do it we utilize the use case of the TOH task using Q-learning (Watkins & Dayan, 1992) as a standard RL algorithm. It involves an artificial learning agent, a set of states (in our use case, the different positions in which the TOH can be found), and a set of actions that can be performed given a state (moving a disk from one rod to another). The agent can transition from one state to another by performing an action, which earns it a reward (a numerical score). The agent's goal is to maximize its total reward, and thus to choose in each state the action providing it the largest possible reward. The Q-learning algorithm uses a reward matrix that represents the rewards associated with each action when performed in each state. This matrix is initialized to an arbitrary fixed value and is updated using the reward obtained each time the agent acts. It is important to highlight that the Q-learning algorithm requires no initial model of the environment and it cannot memorize action sequences. A detailed description of the Q-learning Algorithm can be found in Jang, Kim, Harerimana, and Kim (2019).

Initially, as in the original work, we design a task with $N = 3$ disks to reduce the number of possible states to 27. We define an action as the movement realized by the AA at time t according to the state of the TOH task. An action is dependent on the task (i.e., the number of disks) and its state (e.g., for TOH with $N = 3$, it can be 111, 311). An action induces a change in the state of the game. For example, the action "1-3" means moving the top disk from rod 1 to rod 3 and changes the TOH state from 111 to 311. TOH states describe the task status through the number of disks on the rods at a specific time step t. For example, state "123", presented in Fig. B.12. (6), means that the small disk is on rod 1, the middle disk is on rod 2 in the middle, and the biggest disk is on rod 3. They correspond to a node in the graph representation of Fig. B.13 that represents all the possible moves for the TOH

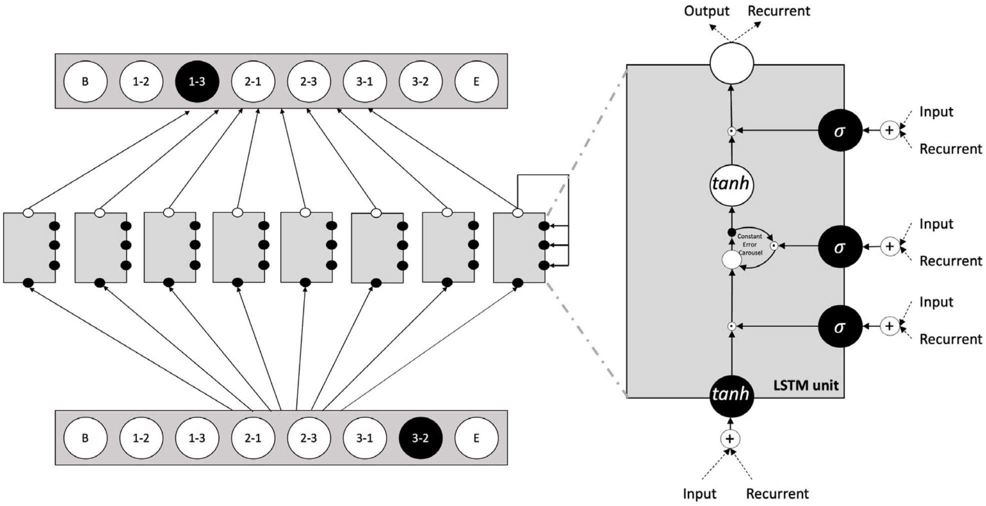


Fig. 3. Architecture of the LSTM model used in step 2 of IKE-XAI methodology. RNN with three layers: an input, a hidden, and an output layer. Input and output layers are both composed of artificial neurons. The hidden layer is composed of LSTM units (Gers, Schmidhuber, & Cummins, 1999) that provide a real-valued vector of size 8. Each LSTM unit is composed of: an input gate, a forget gate, an output gate and a CEC, i.e., a computational unit that remembers values over arbitrary time intervals. σ is the sigmoid activation function and *tanh* the hyperbolic tangent activation function. All white dots outside LSTM units are linked to all black dots. Skip connections connect input and output units. Figure adapted from Chraibi Kaadoud et al. (2022) and inspired from Lapalme (2006). .

with $N = 3$ disks. According to this figure, an action is associated with the number of edges present for each node. We thus identify that there are 3 possible actions according to Fig. B.13, except for nodes 111, 222, and 333, which have only two edges, and thus, two possible actions. Regardless of the value of N disks, if the number of rods remains invariant, then the list of possible movements remains the same. For the initial basic experiment of TOH with $N = 3$ rods, the list of possible movements is: "1-2", "1-3", "2-1", "2-3", "3-1", "3-2".[2] For example, it is possible to reach the TOH state "321" following the actions "1-2", "2-3", or "1-3" that can be codified as follows: (i) initial_state(311) + action(1-2) = new_state(321), (ii) initial_state(221) + action(2-3) = new_state(321), and (iii) initial_state(121) + action(1-3) = new_state(321).

Since we deal with a close-ended task, each sequence of moves has a beginning state and an ending state (easily identified in our case encoded with symbols B and E). The assignment of rewards proceeds as follows: illegal moves get a reward of $-\infty$, those leading to the goal state get a reward of 100, and other moves get a reward of 0. Appendix C.1 presents the algorithm of the Step 1.

There are important points to explicit about the task: (i) rewards put aside, the AA cannot lose the task, and (ii) the AA does not have any knowledge of all the possible moves that can be played. This is relevant to our work since we are trying to mimic the behavior of a child who learns a task without having any prior knowledge of it. The expertise (i.e., the acquisition of experience) is made precisely as the interaction with the game progresses and he arrives at the final states. The whole sequences carried out by an agent are considered valid by definition since they result from a succession of actions in a determined and delimited environment in terms of possibility (as it is not possible to carry out an action other than those authorized by the task). An agent cannot therefore lose but he can carry out very long or even infinite sequences of moves and not reach the final state. Therefore, all sequences of movements carry thus information, and we insist on the fact that we are not in a context of classification of a good or bad behavior (i.e., sequences). The consequence of these features is that the LSTM model will build its implicit representation only based on valid sequences without any prior knowledge of the grammar (set of rules) at the beginning of training. We will detail the importance of these features in the following section.

3.2. Step 2: Sequence learning using an RNN with LSTM units

Long short-term memories (LSTMs) are a type of artificial RNN architecture with feedback connections (Hochreiter & Schmidhuber, 1997) and a complex architecture that performs simple calculations. A common LSTM unit is composed of an LSTM block with one cell called a Constant Error Carousel (CEC), an input gate, an output gate, and a forget gate. The CEC is a computational unit that remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. This allows LSTM units to do essentially two things. On the one hand, they explicitly account for time thanks to their architecture. On the other hand, thanks to the "Constant Error Carousel", a computational unit, they can maintain the error on several time steps without it vanishing or being altered. This last feature allows LSTMs to handle long sequences without suffering from gradient fading or any other unfolding problem. It allows them to process not only single data points (such as images) but also entire sequences of data (such as a video). LSTM units are well-suited for classifying, processing, and making predictions based on time series data since there can be lags of unknown duration between important events in a time series. A detailed description of an LSTM and its variations can be found in Greff, Srivastava, Koutník, Steunebrink, and Schmidhuber (2017), and some explainability approaches to deal with these in Rojat, Puget, Filliat, Del Ser, Gelin, and Díaz-Rodríguez (2021).

For our investigation purpose, we reproduce the implementation presented by Chraibi Kaadoud et al. (2022). We use an RNN of three layers: an input, a hidden, and an output layer. Input and output layers are both composed of artificial neurons whereas the hidden layer is composed of LSTM units as described by Gers et al. (1999): with forget gates and skip connections but no peephole connections. Fig. 3 presents the implemented RNN architecture with LSTM units and a detailed representation of an LSTM unit (i.e., one block with one cell). The number of artificial neurons in the input and output layers is equal to the finite set of 8:6 actions

[2] Whereas for TOH with 4 rods, the list of possible movements increases to 12 moves: "1-2", "1-3", "1-4", "2-1", "2-3", "2-4", "3-1", "3-2", "3-4", "4-1", "4-2", "4-3".

that are possible in a TOH with 3 rods and 3 disks, i.e., "1-2", "1-3", "2-3", "3-2", "2-1", "3-1", and the beginning and ending symbols, i.e., B and E. In the remaining of the text, we will refer to the RNN with LSTM units as LSTM model. The purpose of the LSTM model is to learn valid sequences of moves (i.e., positive examples of sequences that respect the TOH rules displayed in Fig. B.13 and Table 2(b)[3]) leading to states, considered here as an artificial grammar, to encode an implicit representation of the task. More precisely, the LSTM model learns to predict the next move according to the current move and the past ones. Since LSTM units present fewer weaknesses than other RNNs when it comes to long sequences, these models are particularly suited for TOH sequences that can become very long when an agent learns through trial and error in RL. Appendix C.2 presents the algorithm of the Step 2.

Hence, once the learning phase of the LSTM model is made, the network can be used for two purposes: (1) to predict new sequences of moves when solving the task, (2) to check if a sequence of moves is valid according to the TOH rules implicitly encoded by the trained LSTM in its latent space. This is the reason why the trained LSTM model will be used to study the behavior of the AA according to the moves it adopts (see Step 3): while processing inputs (i.e., receiving stimuli) from its environment the network goes through different latent states. We will refer to them as LSTM states that can be understood as implicit activation states of the LSTM model at each prediction.

LSTM states are associated with hidden patterns that are n-dimensional coordinate vectors (n being the size of the hidden layer) in which the network is after processing each input at each time step t. We define a hidden pattern h as a numerical vector of size n that represents the outputs of the LSTM units at each time step. The relation between LSTM states and hidden patterns can be represented as follows: At time t, the network receives information i and internally produces the hidden pattern h, which corresponds to new coordinates in latent space. At the timestep $t + 1$, according to the received information i_{t+1}, the network produces a new hidden pattern h_{t+1} which corresponds to new coordinates of the latent space, at which the network will find itself. With each different input to the network, the LSTM hidden layer will produce a new hidden pattern, i.e., new coordinates in the latent space. Clustering the hidden patterns allows to quantify the latent space into k clusters according to a distance measurement. A cluster (e.g., cluster 2 in Fig. 8(a)) can thus be considered as a set of coordinates designating a "location" in the latent space that allows the model to have a particular state.

3.3. Step 3: eXplainable AI for post-hoc rule extraction

Explainable AI (XAI) techniques using rule extraction allow technical audiences such as data scientists, researchers, and domain scientists to validate and debug the model. One way to approach the problem is extracting the rules implicitly encoded by a recurrent network that learned sequences respecting those rules, in the form of a FSA (Elman, 1990; Omlin & Giles, 1996; Servan-Schreiber et al., 1988; Wang, Zhang, Ororbia, Alexander, Xing, Liu, et al., 2018b). Two cases can occur: (i) the grammar containing the rules is known (ii) the grammar is unknown. In the first case, the extracted rules in the form of a FSA can be compared to the original grammar as a baseline to evaluate the performance of the proposed method. In the second case, the extracted FSA may require a qualitative evaluation by experts of the studied domain. Our work is in the first case: we propose a methodology tested on data from a known grammar (TOH).

To make explicit this hidden knowledge acquired by the recurrent network, we chose to use the implicit rule extraction algorithm of Omlin and Giles (1996) adapted by Chraibi Kaadoud et al. (2022) and in line with Tiňo and Šajda (1995). These three works focused on studying RNNs implicit representation while they learn non-binary sequences. These can be of various sizes with symbols that can be present multiple times (within a sequence) but with different contexts. All sequences are generated from artificial grammars. More concretely, in these works, the objective is extracting structured knowledge acquired by implicit learning when the grammar is unknown and only valid sequences are available for learning (Schellhammer, Diederich, Towsey, & Brugman, 1998; Tiňo & Šajda, 1995), which is our case when training an AA.

The implicit rule extraction algorithm, presented as the third step of Fig. 2 can be described as follows:

(1) Process the sequences of moves by the LSTM model to retrieve a list of hidden patterns: Each one represents the hidden layer activity at each time step.

(2) Cluster hidden patterns using the k-means clustering algorithm (Zeng et al., 1993): To determine the best value of k, we compute the silhouette score s, a quality metric of a dataset clustering.[4] This step results in the association of each hidden pattern h to a cluster C and the grouping of many hidden patterns into a cluster C.

(3) Generation of a FSA for each value of k: Cluster C associated with each hidden pattern h represents the nodes of the automaton graph. By default, all graphs discussed in this work are directed graphs unless stated otherwise. The nodes of the graph represent the LSTM states (Appendix C.2) and each transition between two nodes, or a node on itself in the case of a recurrent loop, represents the change of the network from one state to another when it receives an input. The number of each node in each FSA depends on the value of k chosen for the k-means algorithm. Since each input is a move, each transition of the extracted graph can also be associated with a move as a label. It is important to highlight that each hidden pattern is related to a specific time step, and also a specific input. As proposed in the algorithm in Chraibi Kaadoud et al. (2022), we designate each edge of the FSA with a label that represents the input associated with the edge. The labeling of FSA transitions by the input data allows to explain "Which move realized by the AA led to a particular state?" A pseudo-code of the implicit rule extraction algorithm is provided in Appendix C.3.

We propose that each edge in the FSA has a label and weight that is incremented by a value of 1 each time the AA passes through the states linked by such transition. In other words, a network that passes through the same state x times, due to repeated sequential inputs, would have associated transition weight equal to x (note the presence of node -1, which designates the neutral state of the LSTM model after it is trained and before it receives any input).

[3] An example of non-valid sequence of moves i.e., would be {"1-3", "1-2", "2-3", "1-3", "1-2", "2-3", "1-2"} that does not respect the rules of the TOH task as described in Appendix B.

[4] To obtain the best value of k in the clustering phase, we compute the silhouette score s, a quality metric of clustering of a dataset in automatic classification for all samples for each value of k. More precisely, for a particular sample, the purpose of this metric is to study the distance between resulting clusters on a range of $[-1, 1]$. If the s value is close to -1 the sample is in the wrong cluster, if close to 0, then the sample is at the limit of two clusters and if close to 1, the clustering is considered of good quality, since the sample is assigned to one cluster and far from the others. Thus, the average silhouette score for all samples for each cluster is an objective metric that allows evaluating the quality of clustering for a k value (Rousseeuw, 1987)

Algorithm 1: IKE-XAI methodology: full 3-steps algorithm .

```
Require:
  $N = 3$
  training_sessions = [0, 100, 300, 500, 1000]
  learn_time= play_time=100

  Function IKE_XAI (N, training_sessions, learn_time, play_time) :
  # STEP 1 of IKE-XAI methodology  with RL: AA's Exploration and Training
  R = generate_reward_matrix(N)
  q_learning_sequences_of_moves = AA_exploration_and_training(R, training_sessions, learn_time, play_time)

  # STEP 2 of IKE-XAI methodology  with sequence learning: Implicit rule encoding
  toh_grammar= generate_grammar_from_reward_matrix(R)
  toh_dataset_of_sequences = generate_sequences(toh_grammar, 1000)
  trained_LSTM_model = LSTM_model_training(toh_dataset_of_sequences, number_units, number_LSTM_cells)

  # STEP 3 of IKE-XAI methodology  with XAI: understanding the AA learning phase
  for all session from training_sessions do
    AA_sequences= get_sequences_of_moves_per_training_session(q_learning_sequences_of_moves, session)
    hidden_patterns = trained_LSTM.test(AA_sequences)
    XAI_generation_of_FSA(hidden_patterns, AA_sequences, k)
  end for
  End function
```

We choose setting aside (i) the step of minimization[5] of the generated FSA (generally used in rule extraction algorithms) to focus on the visual information provided by the extracted latent representation through time, and (ii) the validation process of the extracted FSA with sequences to validate the implicit encoding power of the LSTM model since it is done and discussed for the same LSTM model architecture in Chraibi Kaadoud et al. (2022) to which we refer the reader.

3.4. Experimental protocol to produce global explanations with IKE-XAI

We propose an experimental design in three steps to make explicit the knowledge construction of a learning AA, in this case, while performing the TOH task. In the current section, we will detail how the algorithmic blocks presented in previous sections fit all together. As presented in Fig. 2, each step provides data for next one: Step 1 allows researchers and model developers to extract sequences of movements of the AA at different stages of learning. Step 2 uses this dataset as input to test the trained LSTM model. During the test phase, a dataset of recorded hidden patterns is extracted. Finally, Step 3 uses this dataset to perform a post-hoc rule extraction process for XAI purposes. Algorithm 1 presents the IKE-XAI global algorithm using the pseudo-code of the three steps. Appendix C presents the algorithms of the three steps.

4. Results

In this section, we will analyze and detail the results obtained by the proposed IKE-XAI methodology. We particularly focus on explaining the link between the behavior of the AA and the changes observed due to the XAI process rendered through the FSA. First, we describe the dataset and the characteristics of the sequences. Second, we analyze the behavior of the AA during the execution of the TOH task in three different contexts: TOH with $N = 3$, $N = 4$ and $N = 6$. Then, we present for each context the results of the IKE-XAI methodology by detailing the extracted automata obtained using the post-hoc implicit rule extraction algorithm performed on the LSTM model. Particular attention is paid to the classical context of TOH with $N = 3$ disks, and then test scalability on larger N. Finally, we discuss results meaning for TOH states and LSTM states at the explainability level.

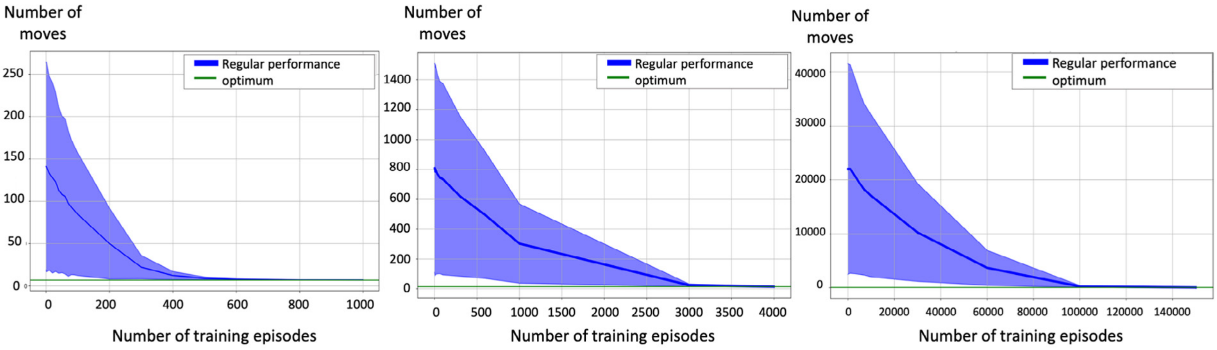
4.1. Description of the dataset: Sequences of autonomous agent moves

A play can be traced as a sequence of the task's states (Table 2(a)) or a sequence of moves made by the player (Table 2(b)). If states have each a unique index, transition labels are not unique. Indeed a move can be played many times but in different contexts. For example, the move "1-3" is made 3 times in the sequence presented in Table 2(b).

While the trained agent is playing, it is thus possible to record sequences that have the same beginning and ending states but varying lengths and repeated moves in different contexts. All sequences of the dataset have the symbol B to mark the beginning of the sequence and symbol E for the end of the sequence as shown in Table 2(c) This process is called encapsulation: it corresponds to the addition of a start and end symbol to a sequence. This technical implementation detail aids the neural network learning and explainability phases by making explicit the beginning and end of a sequence (Table 2(b)) (Servan-Schreiber et al., 1988). On the explainability level, encapsulation facilitates the identification of the movements associated with the transitions during the FSA generation phase (Fig. 2, Step 3). Thus, it becomes easier to interpret behaviors in terms of the extracted FSA of sequential problem-solving actions (Chraibi Kaadoud et al., 2022).

Each sequence of e moves contains $e - 1$ samples that correspond to pairs of "current move-next move". For example, in

[5] The minimization process consists of the transformation of a given FSA into a deterministic finite automaton having a minimal number of states and recognizing the same rational language (Hopcroft, Motwani, & Ullman, 2006).



(a) TOH with N=3 disks

(b) TOH with N=4 disks

(c) TOH with N=6 disks

Fig. 4. Evolution of the average number of moves of the AA while playing: (a) the TOH with $N = 3$ disks after different sessions of training with different numbers of training episodes: 0, 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. Each training session was reproduced 100 times to provide robust results; (b) the TOH with $N = 4$ disks after different sessions of training with different numbers of training episodes: 0, 1, 3, 6, 10, 30, 60, 100, 300, 600, 1000, 3000, 4000; (c) the TOH with $N = 6$ disks after different sessions of training with different numbers of training episodes: 0, 1000, 5000, 7000, 10000, 30000, 60000, 100000, 150000. In all contexts, each training session was reproduced 100 times.

Table 2
The TOH with $N = 3$ disks: Examples (a) and (b) show two runs of the game that express the same sequence of moves taken to solve the task (according to the graph of states in Fig. B.13). Sequence (c) represents the encapsulated version of (b) as learned by the action prediction LSTM model. The encapsulation process of sequences aids the LSTM model learning and FSA explanation extraction phases by making explicit the beginning and end of a sequence.

(a) Sequence of visited states	111, 311, 321, 221, 223, 123, 133, 333
(b) Sequence of moves	1-3 , 1-2, 3-2, 1-3, 2-1, 2-3, 1-3
(c) Sequence of moves encapsulated	B, 1-3, 1-2, 3-2, 1-3, 2-1, 2-3, 1-3, E

Table 2(c), the sequence has 9 moves, i.e., 8 pairs of moves: {(B, 1-3), (1-3, 1-2), (1-2, 3-2), (3-2, 1-3), (1-3, 2-1), (2-1, 2-3), (2-3, 1-3), (1-3, E)}.

At this point, it is important to clarify the difference between a move and a state. Each TOH state can result from either 2 or 3 different moves. For example, it is possible to reach the TOH state "321", following the moves "1-2", "2-3", or "1-3" (see Fig. B.13). We focus on the analysis of sequences of moves since we want to observe the evolution of the agent's behavior at successive time steps. To do so, the LSTM model learns to predict at time t the next move according to the current one and the agent's learning. This is why we decompose each sequence as pairs of moves "current move-next move".

Finally, let us clarify that, by construction, the LSTM model never receives the last move of a sequence as input. For example, we consider the sequence S_1: {B, 1-3, 1-2, 3-2, 1-3, 2-1, 2-3, 1-3, E}, followed by sequence S_2 that also starts with B. The pairs of moves associated with S_1 are: {(B, 1-3), (1-3, 1-2), (1-2, 3-2), (3-2, 1-3), (1-3, 2-1), (2-1, 2-3), (2-3, 1-3), (1-3, E)}. The LSTM model will receive a succession of the following inputs: {B, 1-3, 1-2, 3-2, 1-3, 2-1, 2-3 and 1-3}, for which it will have to predict respectively the following outputs: {1-3, 1-2, 3-2, 1-3, 2-1, 2-3, 1-3, E}. At the end of the input sequence from sequence S_1, the model again receives B as the input from sequence S_2. Consequently, the symbol E is never provided as input to the LSTM model. Fig. 11 illustrates this example through the recurrent loop bearing the label "B" between states 333 and 111.

4.2. Visualizing the evolution of the learning agent behavior: Preliminary analysis with different number of N disks

We recorded the average number of moves of an AA while performing the TOH task at different stages of its training. The AA plays after training at each stage. Preliminary analysis with different numbers of N disks showed that after a given number of episodes, the average number of moves tends to converge to a minimum. We focus thus on the analysis of the behavior during the first training episodes (until stabilization) with the hypothesis that the most important changes in the learning behavior can be observed then. Figs. 4(a), 4(b), and 4(c) present the evolution of the AA average number of moves to solve the task during training, for TOH with $N = 3$ disks (for the first 1000 training episodes), TOH with $N = 4$ disks (for the first 4000 training episodes), and TOH with $N = 6$ disks (for the first 150000 training episodes), respectively.

From the analysis of these three figures, we can observe that the AA learns alone to solve the task. For example, in the context of TOH with $N = 3$ disks, the average number of moves decreases during the first 500 training episodes before stabilizing at an average value of 9 moves to solve the task (7 is the optimal number of moves). Despite the problem' simplicity, the model never learns to solve the problem optimally using the minimal number of movements. Regardless, we observe that the evolution of the average number of movements during training reaches a plateau after 500 training episodes. This evolution of the length of the movement sequences does indeed contain an implicit trace of the knowledge acquired by the agent as it learns. Next section investigates how XAI explains the *Aha!* moment that leads the AA to converge, i.e., to realize how to solve the task in the context of TOH with $N = 3$ disks.

4.3. TOH with $N = 3$ disks: Finding the Aha! moment through extracted automaton analysis

To illustrate the methodology, we present the analysis of the extracted knowledge representations obtained for AA learning to perform TOH with $N = 3$ disks, during the first 1000 learning episodes. We chose to particularly analyze the AA behavior at certain stages of learning: at the beginning of the learning, after 100, 300, 500, and after reaching 1000 training episodes. For each training stage, we recorded during the training phase (for further details see Appendix C.1) a dataset of 100 sequences of moves. Since we consider 5 training stages, we have recorded 5 datasets of 100 sequences each, and 5 FSA have been extracted as a result

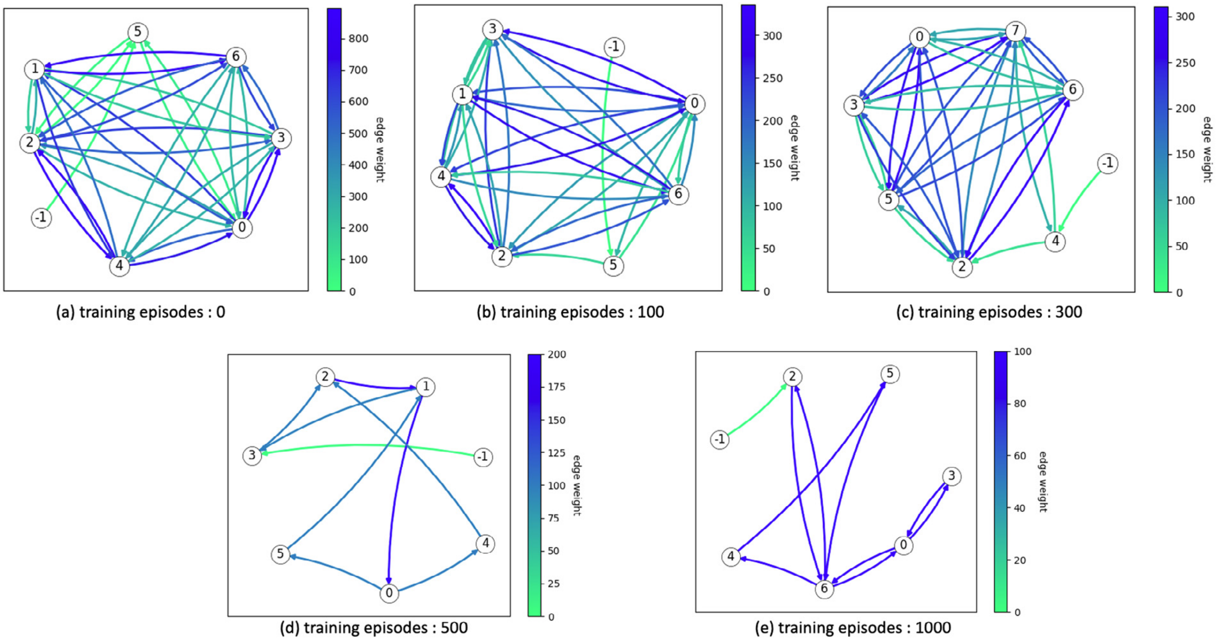

(a) training episodes : 0
(b) training episodes : 100
(c) training episodes : 300
(d) training episodes : 500
(e) training episodes : 1000
edge weight

Fig. 5. Extracted automata with color edges displaying their weights at different stages of the AA training while learning the TOH task with $N = 3$ disks for [0, 1000] training episodes. The darkness of the edge color represents the importance of the weight. Extraction made for: (a) $k = 7$, (b) $k = 7$, (c) $k = 8$, (d) $k = 7$ and (e) $k = 7$, k being the number of clusters for the k-means algorithm that has the best average silhouette score value. The IDs in the nodes represent the clusters' numbers. Each node is an LSTM state that corresponds to one or many TOH states. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of step 3 of the IKE-XAI methodology (Fig. 2, Section 3.3), on the first 1000 hidden patterns.

We analyze each FSA to compare its features (number of edges, edge labels, edge weights) to explicit the behavioral changes. Table 3 provides the features of the dataset, Fig. 5 presents the extracted automaton with colored edges at different stages of the selected training stages, and Figs. 6, 7, 8 represent the automata extracted and the associated edges table at the beginning of the training (Fig. 5(a)), after 300 training episodes (Fig. 5(c)) and after 1000 training episodes (Fig. 5(e)). Each represented FSA has weighted edges color-coded using a heatmap. The darker the color of the edge compared to its weight, the more the AA goes through the state to solve the task.

The sum of all weights of a FSA equals the total number of pairs of moves, i.e., samples of the dataset (i.e., the total number of pairs "current move-next move") analyzed for the explanation. In Figs. 5, 6(a), 7(a) and 8(a), edge weights in the y axis are normalized between 0 and 1: each edge weight is divided by the sum of the edges' weights for a FSA where each weight is obtained from the times that state transition is predicted by the LSTM model in one task solving sequence.

The analysis of the feature of datasets in the experiment (Table 3), the extracted automata and their associate table of edges feature allows us to observe that:

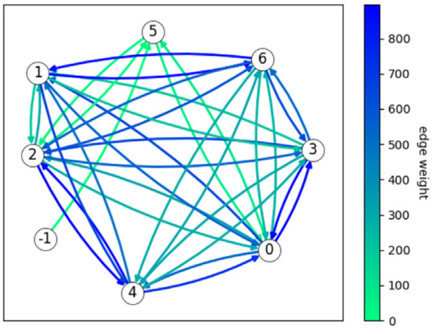
(1) According to Table 3 during the experiment the average number of moves per sequence goes from 159.49, at the beginning of the training, to 56.02 after 300 training episodes to an average of 9 moves after 1000 training episodes. We interpreted this result as an implicit modification of the AA behavior as it has learned the movements that allow it to perform the task efficiently. However, it should be noted that the best k value for the k-means algorithm does not evolve. Its value stays at $k = 7$ mainly. This implies that regardless of the size of the sequences and the stage of training, the LSTM model encodes an implicit representation composed of 7 main states according to the silhouette score calculated for the k-means algorithm at each snapshot of the training process selected to extract and display a FSA.

(2) According to Table 3 we observe that after 1000 training episodes, the average number of moves over 100 sequences of training is 9.0 and that the agent performed 900 moves in total. Figs. 6(b) and 6(c) showed that all transitions evolve to converge to the same weight value of 100 (except the first one), which implies that each move is repeated 100 times. Since the AA does not learn anymore after 1000 training episodes (Fig. 4(a)), we can emphasize its behavior to solve the task becomes repetitive, which implies a stabilization of its knowledge consolidation.

(3) According to Figs. 6(b), 6(c), 7(b) and 8(b), from 0 (the beginning of training) to 300 training episodes, the edges are numerous and of different weights. We counted 37 state transitions (i.e., edges) for the FSA at the beginning of the training, 34 for the extracted FSA after 300 learning episodes, and only 10 edges for the FSA after 1000 training episodes. These changes in the extracted automaton can be interpreted as follows: at the beginning, the agent tries different strategies before it ends up adopting the same sequence of moves, once it has found the optimal sequence of moves (or what it seems to be for it). Then, this translates into the reinforcement of the transition weights that correspond to the optimal sequence of moves, and the weakening of the rest.

(4) According to Fig. 5, edge weights of the extracted FSA, independently of the stage of learning, are higher for some edges than others, which implies that some moves were more repeated than others by the AA when playing.

(5) According to Figs. 6(a), 7(a), and 8(a), we observed that at the beginning of training and also after 300 training episodes, the extracted graph edges remain with different weights; while after 1000 training episodes, except one edge, all others adopt approximately the same weight values (i.e., dark blue). The analysis of Fig. 4(a) and Table 3 show that the average number of moves performed by the AA reduces drastically, about 83%, during training between episodes 300 and 500, at the same time as the performance greatly improves. Fig. 5 presents the



(a)

FSA Edge (source node - target node)	Associated action (move)	Computed weight
"-1" - 5	B	1
5 - 0	"1-3"	58
5 - 2	"1-2"	42
0 - 3	"3-1"	882
0 - 2	"1-2"	335
0 - 6	"2-3"	261
0 - 4	"2-1"	538
0 - 1	"3-2"	658
0 - 5	B	82
0 - 0	"1-3"	20
3 - 2	"1-2"	650

(b)

FSA Edge (source node - target node	Associated action (move)	Computed weight
3 - 0	"1-3"	873
3 - 6	"2-3"	519
3 - 4	"2-1"	303
3 - 1	"3-2"	207
2 - 4	"2-1"	897
2 - 1	"3-2"	292
2 - 6	"2-3"	616
2 - 0	"1-3"	279
2 - 3	"3-1"	542
2 - 5	B	17
2 - 2	"1-2"	3
4 - 0	"1-3"	729
4 - 2	"1-2"	864
4 - 1	"3-2"	503
4 - 6	"2-3"	316
4 - 4	"2-1"	102
4 - 3	"3-1"	283
1 - 0	"1-3"	557
1 - 4	"2-1"	671
1 - 6	"2-3"	867
1 - 3	"3-1"	215
1 - 2	"1-2"	188
6 - 3	"3-1"	630
6 - 1	"3-2"	838
6 - 4	"2-1"	286
6 - 2	"1-2"	564
6 - 0	"1-3"	261

(c)

Fig. 6. FSA extracted at the beginning of the training of the TOH task with $N = 3$ disks, with $k = 7$ for the k-means algorithm: (a) Extracted FSA with colored edges displaying their weights: the darker the edge, the more important the weight. (b, c) Associated actions and computed weights. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

dominant sequence of moves repeated by the AA after 1000 training episodes. These elements lead us to consider that the *Aha!* moment, i.e., the moment when the learning AA finds a solution, is around 500 episodes since after that the average number of movements barely evolves (from 10 to 9 average moves). In this case, it is not a particular time step, but a time range over which it modifies its behavior in a fast way to adopt what seems to be the optimal solution (even if computationally, the adopted solution is not the optimal one).

As per the previous observations and analysis, we can make two conclusions: First, the more the AA is trained to perform the TOH task, the more the extracted FSA tends to converge to a stable form with transitions of approximately equal weight. In other words, once the AA's number of average moves stabilizes, i.e., the behavior gets stable, the extracted FSA does not change anymore (and neither do the values of the associated edge weights). An AA that does not learn anymore, uses the already discovered path (i.e., optimized sequence of moves) to perform the task. Second, the *Aha!* moment during training can be detected using the IKE-XAI method by allowing researchers to identify the moment where the AA finds a solution through the adoption of the AA of stable behavior, not only in terms of convergence of the learning curve but also in terms of a graphical simplified representation of the same, in form of a FSA.

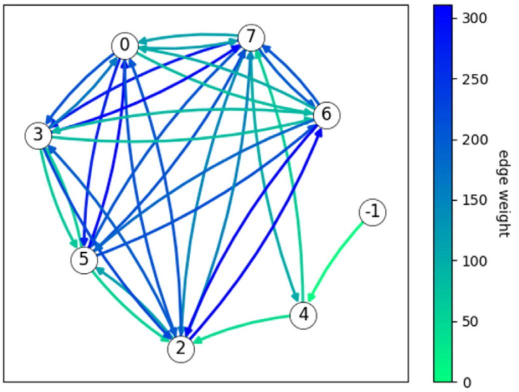
4.4. TOH with $N > 3$ disks

To evaluate the contribution of our approach on a more complex task, we chose to test the IKE-XAI methodology on $N = 4$ and $N = 6$. In this section, we present the results in these two contexts.

4.4.1. TOH with $N = 4$ disks

For $N = 4$ disks, the optimal number of moves is $2^4 - 1 = 15$. The graph of all possible moves contains 81 nodes and 120 edges. We recorded the average number of movements performed by the AA (i) every 10 training episodes from 0 to 100, (ii) every 100 training episodes from 100 to 1000, and (iii) every 500 training episodes, until reaching 8000 training episodes. The average was calculated on 100 simulations. We observed that for $N = 4$, the average number of sequences decreases between 0 and 4000 training episodes before stabilizing around the optimal number of movements. For 0 training episodes, the average number of movements recorded is 800. After 4000 training episodes, this number is reduced to 15. Therefore, we chose to extract the automata on 1000 hidden patterns at the following sessions: 0 (before training), 1000, 2000, 3000, and 4000 training episodes. Fig. 4(b) shows the evolution of the number of moves for $N = 4$.

Fig. 9 presents the automata extracted at different stages of training for an AA who performs a TOH with $N = 4$ disks on 1000 hidden patterns, i.e., 1000 time steps. We notice an evolution of the number of edges between the beginning of the training and after 4000 training episodes: from 37 to 11. We also observe an evolution of the weights of these edges: At the beginning of the training the maximum weight is 45 while after 4000 training episodes, the maximum weight is 983, but it concerns only one transition (the recurrent loop on node 0). Moreover, although the task has become more complex (the number of nodes and transitions in the graph of all possible moves of the TOH task with $N = 4$ disks is more important than for $N = 3$ disks) we observe that the automata extracted for 1000 patterns all have 7 nodes. This makes them intelligible visual representations with a small number of nodes. As the automata represent the construction and the evolution of the implicit knowledge of the AA, we emphasize that these representations allow observing the structuring of its knowledge through the evolution of its behavior. Indeed, as the training progresses, we observe the reinforcement of certain edges, and the forgetting of others, which represents an evolution. We can also see that the representations in Fig. 9(d) and Fig. 9(e), both allow us to identify the state in which the

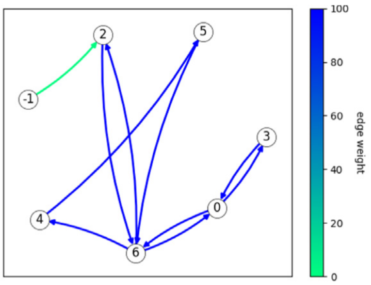


(a)

FSA Edge (source node - target node)	Associated action (move)	Computed weight
“-1” - 4	B	1
4 - 7	“1-3”	59
4 - 2	“1-2”	41
7 - 5	“3-2”	206
7 - 0	“2-3”	111
7 - 4	B	99
7 - 6	“2-1”	196
7 - 3	“3-1”	284
7 - 7	“1-3”	50
7 - 2	“1-2”	144
5 - 0	“2-3”	305
5 - 6	“2-1”	215
5 - 7	“1-3”	211
5 - 3	“3-1”	58
5 - 2	“1-2”	52
0 - 2	“1-2”	207
0 - 3	“3-1”	209
0 - 7	“1-3”	114
0 - 5	“3-2”	273
0 - 6	“2-1”	80
2 - 3	“3-1”	197
2 - 6	“2-1”	311
2 - 7	“1-3”	147
2 - 0	“2-3”	206
2 - 5	“3-2”	112
3 - 2	“1-2”	224
3 - 0	“2-3”	162
3 - 7	“1-3”	304
3 - 5	“3-2”	64
3 - 6	“2-1”	79
6 - 2	“1-2”	305
6 - 7	“1-3”	206
6 - 3	“3-1”	85
6 - 5	“3-2”	186
6 - 0	“2-3”	99

(b)

Fig. 7. FSA extracted after 300 training episodes of the TOH task with $N = 3$ disks, with $k = 8$ for the k-means algorithm: (a) Extracted FSA with colored edges displaying their weights: the darker the edge, the more important the weight. (b) Associated actions and computed weights. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)

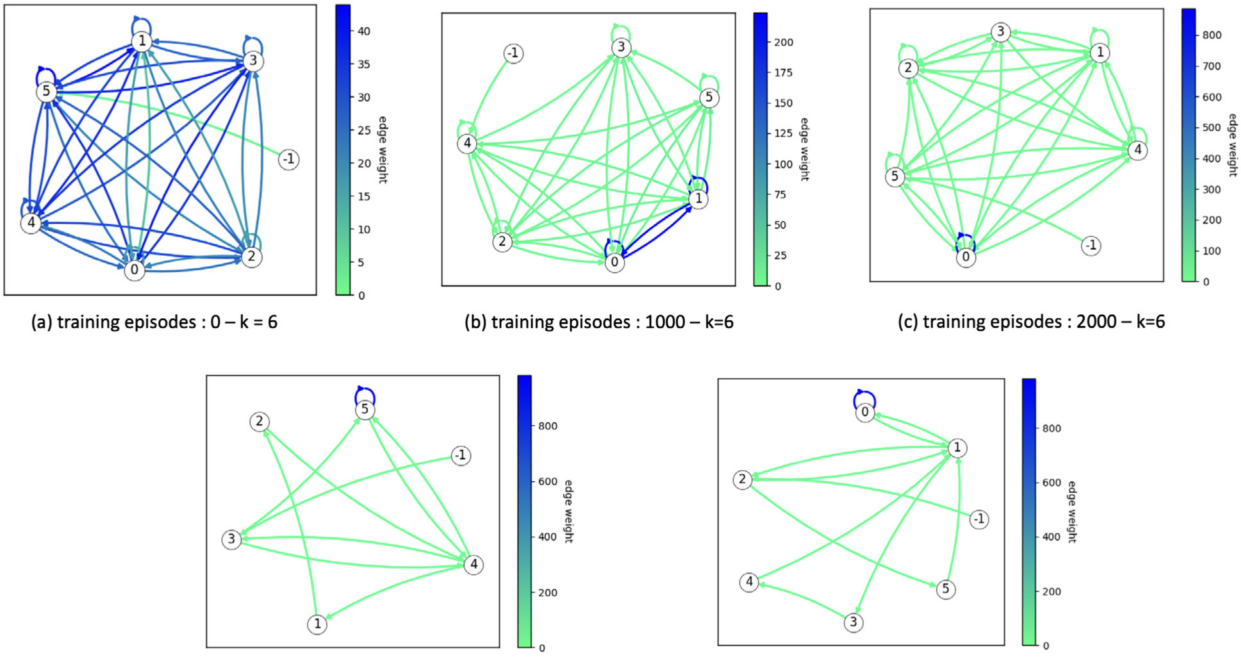
FSA Edge (source node - target node)	Associated action (move)	Computed weight
“-1” - 111	B	1
2 - 6	“1-3”	100
6 - 4	“1-2”	100
4 - 5	“3-2”	100
5 - 6	“1-3”	100
6 - 0	“2-3”	100
0 - 3	“3-1”	100
3 - 0	“2-3”	100
0 - 6	“1-3”	100
6 - 2	B	99

(b)

Fig. 8. FSA extracted after 1000 training episodes of the TOH task with $N = 3$ disks, with $k = 7$ for the k-means algorithm: (a) Extracted FSA with colored edges displaying their weights: the darker the edge, the more important the weight. (b) Associated actions and computed weights. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Analysis of the dataset composed of recorded sequences of moves of an AA trained to solve the TOH task with $N = 3$ disks. Each dataset is composed of 100 sequences of moves of various lengths and is analyzed at different stages of training: At the beginning, after 100, 300, 500, and 1000 learning episodes. Each dataset is clustered using the k-means algorithm for $k \in [5; 100]$. The table presents: (i) the best value of k selected for each dataset, (ii) the associated computed average number of moves per sequence (obtained as the total number of actions/total number of sequences), and (iii) the total number of samples (i.e., pairs of moves) of each dataset.

Number training episodes (i.e., sequences)	Best k value	Average number of moves per sequence	Total number of samples
0	7	159.49	15949
100	7	57.31	5731
300	8	56.02	5602
500	7	10	1000
1000	7	9	900



(a) training episodes : 0 – k = 6

(b) training episodes : 1000 – k=6

(c) training episodes : 2000 – k=6

(d) training episodes : 3000 –k = 6

(e) training episodes : 4000 – k=6

Fig. 9. Extracted automata with color edges displaying their weights at different stages of the AA training while learning the TOH task with $N = 4$ disks for [0, 4000] training episodes. The darkness of the edge color represents the importance of the weight. Extraction made for $k = 6$ for all stages of training, k being the number of clusters for the k-means algorithm that has the best average silhouette score value. The IDs in the nodes represent clusters' numbers. Each node is an LSTM state that corresponds to one or many TOH states. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

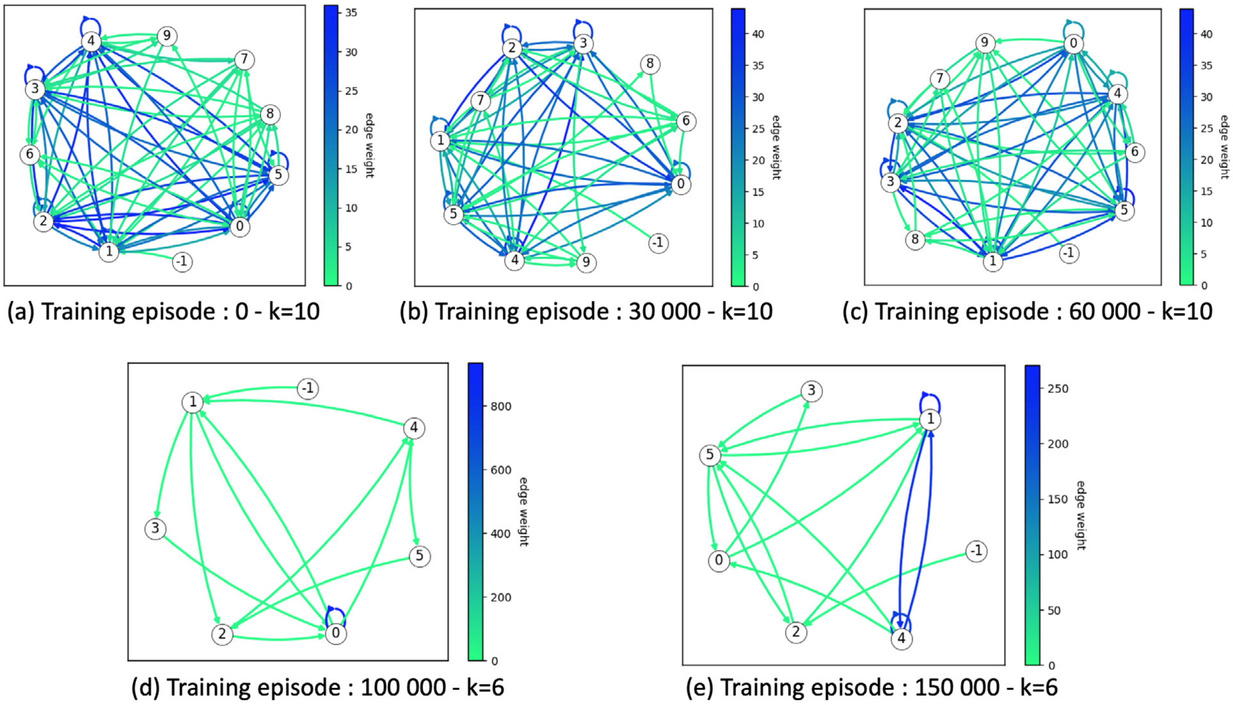
agent remains the longest: namely the 0 state. An analysis of the associated actions and computed weights allowed us to identify that the state 0 is linked to the following movements “3-1”, “2-1”, “3-2”, “1-3”, “1-2” that are played more than 200 times.

4.4.2. TOH with $N = 6$ disks

For $N = 6$ disks, the optimal number of moves is $2^6 - 1 = 63$. The graph of all possible moves contains 729 nodes and 1092 edges. During preliminary experiments, we recorded the AA average number of movements from 0 to 500000 training episodes in steps of 50000 training episodes. We observed that for $N = 6$, the average number of sequences decreases linearly between 0 and 150000 training sessions before stabilizing around 70 moves. We, therefore, chose to extract the automata at the following sessions: 0 (before training), 30000, 60000, 100000, and 150000 training sessions. Fig. 4(c) shows the evolution of the number of moves for $N = 6$. As for TOH with $N = 4$, we set a threshold number of movements so that the AA during playing does not go beyond 10000 moves per playing trial (i.e., to avoid infinite loops).

Fig. 10 presents the automata extracted at different stages of training for an AA who performs a TOH with $N = 6$ disks on the first 1000 hidden patterns, i.e., 1000 time steps. The analysis of the characteristics of the FSA shows an evolution of the number of nodes and the weights of the transitions. Indeed, between 0 and 60000 training episodes (Fig. 10(a), (b), (c)), the automata have 11 nodes and the maximum weight on all transitions is 36 at the beginning of the training. The maximum weight value increases to 44 for 30000 training episodes and remains at 44 after 60000 training episodes.

When the agent trains 100000 times (Fig. 10(d)), the extracted automaton associated with its behavior has 7 nodes. The maximum weight of the transitions is 936 but only concerns the recurrent loop on node 0. The other transitions have a weight that varies between 2 and 16 (except the first transition associated with the node -1) After 150000 training episodes, we observe that the maximum weight of the transitions is 271 (Fig. 10(e)). More precisely, 4 transitions have an important weight: 208 (transition (4, 4)), 227 (transition (4, 1)), 233 (transition (1, 4) and 271 (transition (1, 1)). The other transitions have weights that vary between 4 and 8 (except for the first transition involving node -1). These variations of weights between transitions allow us to establish that there is indeed a change of behavior in the movements performed by the agent. Over 1000 patterns, i.e., time steps, the agent explores several strategies through several different sequences of movements before reducing its choices to determined actions that are repeated after that. We also notice that although the number of nodes and possible transitions for TOH with $N = 6$ disks is much higher (729 nodes, 1092 transitions) than for $N = 4$ (81 nodes, 120 edges) and $N = 3$ (27 nodes and 39 edges), the extracted FSAs for the same number of patterns (1000 in the 3 cases) are still quite small (number of nodes between 6 and 10), which makes them visually accessible. Let us recall here that the nodes of the extracted automata correspond to the LSTM states. Each node can group one or more TOH states. Whether it is for $N = 4$ or $N = 6$, the number of nodes and transitions being very important makes the follow-up of the agent's behavior complex for a human observer. However, our automata allow synthetic representations of the TOH task with a reduced number of nodes, as perceived by the AA, by grouping together the TOH states that result from the same rule, i.e., series of moves. We recall here that a rule is defined as a sub-part of a sequence of 2 to 3 successive moves in time. For example, the study of the associated movements linked to the nodes and transitions of the automaton extracted after 150 000 training episodes (Fig. 10(e)) shows that the automaton often repeats the rule “3-1, 2-1, 3-2” in succession, resulting in the clustering of TOH states “213222”, “113222” and “112222” at node 4 of the automaton. Another example is the rule “1-2, 3-2, 3-1”, repeated several times in time at node 1 which induces the grouping of TOH states “312332”, “322332” and “222332”.



(a) Training episode : 0 - k=10

(b) Training episode : 30 000 - k=10

(c) Training episode : 60 000 - k=10

(d) Training episode : 100 000 - k=6

(e) Training episode : 150 000 - k=6

Fig. 10. Extracted automata with color edges displaying their weights at different stages of the AA training while learning the TOH task with $N = 6$ disks for [0, 150000] training episodes. The darkness of the edge color represents the importance of the weight. Extraction made for: (a) $k = 10$, (b) $k = 10$, (c) $k = 10$, (d) $k = 6$ and (e) $k = 6$, k being the number of clusters for the k-means algorithm that has the best average silhouette score value. The IDs in the nodes represent clusters' numbers. Each node is an LSTM state that corresponds to one or many TOH states. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

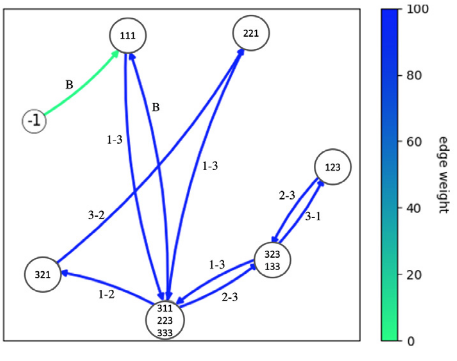
Table 4
Summary of the results concerning the number of average training episodes that result in an *Aha!* moment and the evolution of the average length of sequences at the beginning of training and after the *Aha!* moment in the different experimental contexts tested: TOH task with N in [3, 4, 6].

	TOH with $N = 3$	TOH with $N = 4$	TOH with $N = 6$
Aha! moment (average number of training episodes)	500	3000	100000
Average length of sequences at the beginning of training	6	800	21500
Average length of sequences after the *Aha!* moment	9	15	63

The results provided in the three different contexts (TOH task with N in [3, 4, 6]) show that the IKE-XAI methodology works in complex contexts. Our methodology offers a way to observe the phenomenon of implicit knowledge building and evolution present in the sequences of movements of the artificial agents. Regarding the *Aha!* moment, in the 3 contexts, from the analyses performed we concluded that the *Aha!* moment for an AA occurs when it changes its behavior noticeably, which translates into a significant change in the extracted FSAs and a stabilization of them. In the first context, TOH with $N = 3$ disks, the *Aha!* moment occurs around 500 training episodes (Figs. 5, 7 and 4(a)). In the second context, TOH with $N = 4$ disks, the *Aha!* moment occurs around 3000 training episodes (Figs. 9 and 4(b)). Finally in the third context, TOH with $N = 6$ disks, the *Aha!* moment occurs around 100000 training sessions (Figs. 10 and 4(c)). Table 4 summarizes these results and displays also the change in the average length of sequences between the beginning of the training and after the *Aha!* moment. In each experimental context, after each detected *Aha!* moment, extracted FSA stabilize (Figs. 5(d), 5(e), 9(d), 9(e) and 10(d) and 10(e)). Therefore, we consider that for an artificial agent, the *Aha!* moment corresponds to the moment when it adopts a monotonic strategy that it repeats thereafter. Let us underline the fact that the *Aha!* moment does not occur at the same time in the three contexts because of the complexity of the tasks, as when complexity increases, more exploration is required.

4.5. Discussion and investigation of the link between TOH states and LSTM states

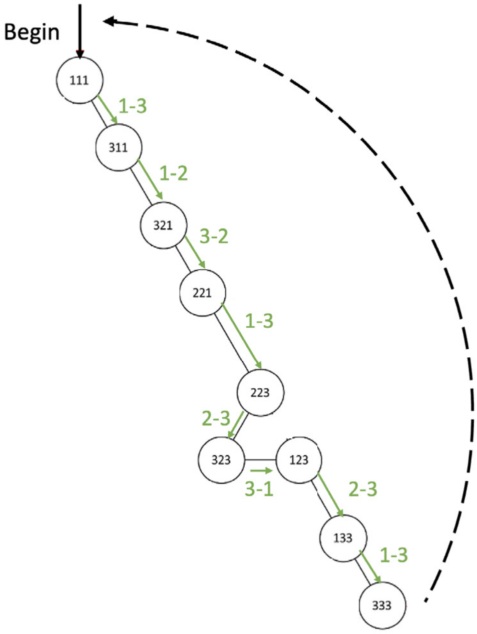
In this section, we discuss the link between the TOH states and LSTM states in the extracted automata. Fig. 11 presents the extracted FSA obtained (TOH with $N = 3$ disks) after 1000 training episodes, with TOH states mapped on the LSTM states (nodes). The FSA edge labels indicate the associated move played to go from one state of the task to another. As per Fig. 11, we observed that many TOH states can correspond to one LSTM state. The TOH states “311”, “223” and “333” are grouped at the level of an LSTM state or node “6” (Fig. 8(a)), and TOH states “323” and “133” are grouped at the level of an LSTM state or node “0” (Fig. 8(a)). More precisely, we observed that the states grouped are states resulting from the same move. For the game to be in states “311”, “223”, and “333”, the previous move is “1-3”, i.e., moving the top disk from rod 1 to rod 3. For the game to be in states “323” and “133” which are grouped, the previous move is “2-3”,i.e., moving the top disk from rod 2 to rod 3. Thus the



(a)

FSA Edge (source node - target node)	Associated action (move)	Computed weight
“-1” - 111	B	1
111 - 311	“1-3”	100
311 - 321	“1-2”	100
321 - 221	“3-2”	100
221 - 223	“1-3”	100
223 - 323	“2-3”	100
323 - 123	“3-1”	100
123 - 133	“2-3”	100
133 - 333	“1-3”	100
333 - 111	B	99

(b)



(c)

Fig. 11. Analysis of FSA extracted for TOH with $N = 3$ disks: (a) Extracted FSA after 1000 training episodes with $k = 7$ for the k-means algorithm. (1) colored edges represent their importance weight (the darker the edge, the more important the weight); (2) TOH states identify each FSA node (instead of LSTM states as in Fig. 8(a)); (3) Edge labels explain the move to make according to the trained LSTM, to go from one TOH state to another. (b) Associated actions and computed weights in the TOH task. (c) Associated path in the FSA graph representation (Fig. B.13): Green arrows represent the AA followed path, i.e., a sequence of moves that correspond to the task strategy displayed in (a) and (b) The explanation about the arrow from the final state “333” to the starting one “111” with label B is explained by the fact that the transitions are labeled at each time step by the data provided as input to the LSTM model (More details in Section 4.1). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

grouping of states of the game seems to imply that they are the result of the same move.

The order of the TOH states in a node, from top to bottom, represents the order of the TOH states in which the agent finds itself when acting according to the progression of the sequence of moves. The fact that some TOH states are grouped into the same node in the FSA, means that the LSTM model perceives these states as close in its learned latent space. Since an LSTM state is characterized by the node at which the input is currently being processed, different TOH states can be grouped within the same LSTM state.

Apart from the fact that their corresponding hidden states are similar in the vector space, the grouping of TOH states together in the same corresponding LSTM state allows us to understand the representation of the game for the agent as it interacts. This tends to be supported by the fact that the TOH states grouped resulted from the same action. The FSA extracted seems to carry the “vision of the world” that the AA has built through the interaction with its environment (here the TOH task with N disks), and that is hidden in the sequences. Moreover, if the action leading to a node of the automaton is repeated several times (high transition weight) and the latter gathers several TOH states, this can be interpreted as a strategy of movements for the agent to lead the game to a particular state that it considers part of the solution.

In summary, there is a clear link between interpreting LSTM states and the task (here TOH) states: the former can be seen or rendered as clusters of the latter, as the learning AA encoded them nearby in the embedding, and recalls them over time during its training experience. While this ability to cluster is interesting from the perspective of the question of analyzing the evolution of knowledge within an AA (hypothesis H2, Section 1.3), it is also an important limitation of our approach. Indeed, we recognize that the representation of time in the automaton in a simple and accessible way for a non-expert user of our methodology can be a limit for the adoption of our approach. When TOH states are grouped in an LSTM state, it is difficult to determine which TOH state was reached first, without going through the generation history of the automaton. This is also true for two transitions that connect two nodes in opposite directions. In summary, while the extraction of several automata allows us to visually observe the evolution of the agent’s knowledge, the complex access to the history of the choices made by the agent over time remains a major obstacle to the adoption of our methodology by a non-expert audience.

5. Discussion

In this paper, we tackle the problem of explaining the acquisition and evolution of the latent representation of a task acquired by an AA while learning. We focus our study on the TOH task, a well-known transformation problem in the field of problem-solving. The main contribution of our work is to propose the IKE-XAI end-to-end methodology for knowledge extraction from AA observational behavior. On the taxonomy of XAI techniques, IKE-XAI can be labeled as a post-hoc explainable methodology that provides a visual model-agnostic explanation (it can explain any AA behavior). At the experimental level, we were able to demonstrate that it is possible to extract the vision of the AA of a task (simple and complex one) using a sequence learning model (LSTM) and to extract knowledge, in the form of FSA that represents AA’s problem-solving strategies, for their explainability. Since our work is a multidisciplinary one at the crossroad of different domains, we will discuss our contributions for each domain: explainable RL and developmental psychology (*Aha!* moment).

First, let us discuss our work's contribution to the **explainable RL field**: this work offers the IKE-XAI methodology for implicit knowledge extraction in the shape of FSA when only sequences of moves are available (only positive examples). It is important to highlight that no precise methodology to extract AA's acquired representation of the task/world was available based on observational behavior such as in human–machine interaction. Our work thus allows filling this gap by combining RL, implicit rule learning from sequences, and a post-hoc rule extraction explainable algorithm. Regarding the contribution of explainability on the evolution of a learning agent behavior, we confirmed hypothesis *H2* (Section 1.3) by making explicit the behavioral changes of an AA due to the analysis of the edge weights of the extracted automata. From the AA behavioral analysis point of view, for TOH with N in [3, 4, 6] disks, the extracted automata at different stages of the AA training (1) provide information about the behavior change and knowledge acquisition, (2) evolve into a consolidated, simpler FSA that is easier to convey to debug and explain the model. This last characteristic allows to confirmed hypothesis *H3* (Section 1.3).

Note that our approach also allows explaining behaviors that are not optimal. For TOH with $N = 3$ disks, when after 1000 training episodes, the agent performs the task with 9 movements instead of 7 (that is the optimal number of moves), our method allows us to determine it. We considered that it was interesting to see the contribution of our method when the observed behavior is not the expected one. Fig. 11 illustrates this difference. In the present work, we did not investigate the reasons for this behavior. We wanted to stay focused on the explanation of the behavior by the construction and evolution of knowledge, which are intrinsic to the agent. Nevertheless, we would like to extend our work to this type of problem in the future. It is important to underline that the IKE-XAI method makes explicit the global behavior of the agent when it plays by putting forward only the part of the learned knowledge used to play even on a particular sequence of moves (independently of its size). Once the LSTM model has been trained on TOH sequences (step 2 of the IKE-XAI methodology), the FSA extraction algorithm allows to extract an automaton without any constraint on the size of the sequence. This permits a targeted explainability of a particular behavior i.e., a specific sequence of moves. The association of labels to transitions and time steps allows characterizing temporally an action that led to a state and describing it contextually (past actions, current action, and the resulting state in terms of action and perception). On top of that, the use of LSTM in IKE-XAI helps the method to encode long-term dependencies (long sequences with distant context) that can be hidden in the AA sequence of moves. In the case of the TOH task, only recent context (move and state at $t - 1$) matter for the prediction of the next movement and state. However Chraibi Kaadoud et al. (2022) showed that LSTM can encode and predict sequences of 100000 symbols length with long term dependencies. IKE-XAI can thus theoretically be used to analyze sequences of moves of an important length where first moves impact the prediction of moves that happen later in the sequence. Finally, the consolidation of the extracted automata makes it possible to establish the moment when the AA seeks no longer to try strategies to solve the task, by using only the most effective found paths it learned. Comparing the rules of TOH to streets in a city, during step 1 of IKE-XAI, the AA explores and learns the streets. During step 2 of IKE-XAI, the RNN learns all the itineraries that could be followed by the AA in the city to encode an implicit map of this city. Thus, step 3 takes care of explaining which part of the map the AA mainly used to solve the task, and how it finds a way to do it.

Next, at **developmental psychology (*Aha!* moment) level**, it is important to recall that there is only a small amount of work on the issue of insight modeling in artificial agents (Colin, 2020; Colin & Belpaeme, 2019). Our work contributes to investigating this scientific question. In this field, the *Aha!* moment is defined as a sudden and spontaneous change in an individual's behavior. At the cognitive level, it is the emergence in the conscious sphere of an individual of an idea that allows him to solve the problem he is facing. If this is apparently spontaneous (the famous *Aha!* moment), studies show that insight is not a sudden flash that comes from nowhere, but is the result of the implicit cognition[6] piecing together loosely connected bits of information stemming from prior knowledge and experiences to form novel associations among them (Carpenter, 2019). Moreover, the lack of other means of communication (and provided that he/she can express it) and without specific techniques (e.g., fMRI), make it difficult for us to hypothesize with confidence the occurrence of the *Aha!* moment. In the case of children, this was apparent especially through verbal reflection or/and non-verbal behaviors. Therefore it is difficult to demonstrate exactly when insight occurred in a scientific and applied technical manner. In children, in principle, the TOH requires the use of inhibitory control at certain instances, which means that for certain cases the solver does not perform the most obvious move. It was also observed that the transition from the exploratory phase to the phase of understanding requires that moment of cognitive transformation (the so-called *Aha!* moment). This moment can be observed (i) by considering the task performance and (ii) by taking into account the relevant verbal and non-verbal manifestations. In the case of the AA, we lack the second input. However we can underline some resemblances between the AA behavior and children regarding the *Aha!* moment: **(1)** the fruitless search/impasse phase from the resolution moment (convergence of the model) (Figs. 5, 9, 10), **(2)** the steepness of the performance curve (Fig. 4) that shows the expertise transformation in the AA, **(3)** the use of previously acquired learning of the task to make the "insight" possible, **(4)** The processing of the problem that becomes more fluidly after the "insight", and **(5)** the shift in the agent's behavior from exploration to exploitation i.e., the *Eureka!* moment for the agent and the *Aha!* moment for the researcher when he/she understands when it happens (an increase in the speed of resolution of the task). On top of that, if we consider that the concept of "moment" is not linked to a particular instant t but rather to a specific time range, this conceptualization of insight supports the interest of our study since it allows us to detect the range of time over which an agent finds and adopts a strategy (even if it is not optimal) over the long term. We are aware that this represents a limitation of our approach and we hope to work on this subject in our future work. Nevertheless, we believe that the current work offers an interesting way of investigating the subject of *Aha!* moment for autonomous agents and beyond that, it leads to a reflection on the question of the definition of insight for an artificial autonomous agent. The convergence of models is thus interesting for the study of this phenomenon in autonomous artificial agents, and more globally for the question of explainability and should be studied further.

Finally, regarding **the originality of our contribution concerning the original works** of Bennetot et al. (2020) and Chraibi Kaadoud et al. (2022), we differ from the first work by considering sequences of movements as instantiations of the agent's behavior (and not only their average sequence length) to evaluate the evolution of behavior and learning. Indeed in the current work, we hypothesized (Hypothesis *H1*, Section 1.3) – and confirmed – that each sequence of movements is governed by implicit rules of behavior that can be extracted. Our analysis

[6] That refers to *"unconscious influences reflecting perception, memory, and learning, without subjective phenomenal awareness"*. (Reingold & Ray, 2002)

of the data thus went further than the original work (originally with TOH with $N = 3$ disks) to explicit the behavior. Let us also highlight that we extended it to new experimental contexts ($N = 4$ and $N = 6$). Concerning the second work, originally in the field of interpretability, by positioning ourselves in the field of explainability applied to developmental robotics. We put forward that an implicit rule extraction algorithm from an RNN that learned sequences could also bring explanations on the behavior of an AA, using visualization techniques, at time t, but also during time at different stages of the training acquisition, which was not treated in the original paper. Our work is thus an original work that focuses on the problem of the study of the evolution of the construction of the latent representation of an AA through the study of its observational behavior.

To conclude, this technique is presented as a tentative model for insight problem-solving in artificial agents by mimicking children, and as a promising technique for improving the problem-solving abilities of artificial agents. Regarding the reproducibility or scalability of the IKE-XAI methodology, the selected use case on TOH, the type of dataset, and the algorithms provided in Appendix C allow it to be easily extended to other domains where the data to learn from is in the form of sequences of actions, decisions, or movements. In addition, the complete code is available on the GitHub plateform at https://github.com/ichraibi/Extracting_Aha_moment_from_Qlearning_agent_through_IKE-XAI_method.ip for further development or re-use by third parties.

6. Conclusion and future work

The proposed IKE-XAI methodology contributes to the field of explainability of learning agent behavior using a cognitive modeling approach. We considered in this work the field of explainable RL as an alliance of cognitive modeling and XAI. We used the TOH as a guiding showcase of the proposed IKE-XAI methodology. We showed that, as for humans during a problem-solving task, the AA builds a global knowledge (i.e., a latent representation) of the task performed during interaction with the environment. That knowledge evolves along training, i.e., and so does the acquisition of experience. The IKE-XAI methodology made it possible to identify the *Aha!* moment. The AA got a Eureka moment optimizing its moves for the TOH and we humans have a Eureka moment figuring out when the artificial agent is capable of doing it. As a conclusion of our results, we showed that IKE-XAI makes it possible to elucidate the evolution of knowledge acquisition of a learning AA through the study of its behavior over time in terms of an extracted, synthesizing FSA. This allows us to convey, in a symbolic manner, a more explainable vision of black box sequence learning models. As main directions for future work, an important perspective of this work would be to obtain iterative feedback from the developmental robotics community on how useful XAI approaches can be to understand the knowledge development of AI systems. It can have two-fold benefits and, at the same time, shed some light on new ideas and debates to improve the state-of-the-art in developmental psychology and cognitive science. It can also help us transpose the IKE-XAI methodology in different contexts such as studying the construction of a complex AA behavior (Lieto, Bhatt, Oltramari, & Vernon, 2018; Vanderelst & Winfield, 2018) or a child's knowledge during the autonomous learning process of a transformation problem. Regarding the latter, we would be particularly interested in the formal operational stage when the reasoning capacities are set up. This is the stage where the child's cognitive structures shape to be close to those of an adult and include conceptual reasoning (Donald, Lazarus, & Lolwana, 2006). The child can thus perform a variety of tasks involving the use of hypotheses to solve problems. Therefore, solving transformation problems such as TOH could bring interesting elements for the implementation of cognitive strategies in education and could be generalized to different cognitive profiles (Lefa, 2014). Additionally, it could contribute to (i) works that explore the type of possible targeted interventions and scaffolding that a child needs for the optimization of the problem-solving process in a specific task such as the TOH (Charisi et al., 2020; Charisi, Merino, Escobar, Caballero, Gomez, & Gómez, 2021), (ii) works that design child–robot interaction settings with child-initiated robotic interventions.

To conclude, we invite AI and XAI future research works and, more globally, the scientific community to adopt trans-disciplinary approaches, especially in the human-centered AI domain. Works at the crossroads of multiple fields may participate to improve knowledge about artificial cognition, through the study and modeling of human learning and, above all, to improve the trust in ML algorithms by making the reasons behind their behavior more accessible and transparent to everyone.

CRediT authorship contribution statement

Ikram Chraibi Kaadoud: Conceptualization, Data analysis and interpretation, Investigation, Design, Methodology, Software - all project parts, Writing – original draft. **Adrien Bennetot:** Software - Reinforcement Learning part, Writing–review & editing. **Barbara Mawhin:** Conceptualization, Methodology, Writing - review & editing. **Vicky Charisi:** Conceptualization, Writing– review & editing. **Natalia Díaz-Rodríguez:** Supervision, Conceptualization, Methodology, Writing - review & editing, Validation, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

Ikram Chraibi Kaadoud is currently supported by *Conseil régional de Bretagne* and the *European Union* via the *FEDER* program. Vicky Charisi is currently supported by the HUMAINT project of the JRC. Natalia Díaz-Rodríguez is currently supported by the Spanish Government Juan de la Cierva Incorporación grant IJC2019-039152-I funded by MCIN/AEI /10.13039/501100011033 by "ESF Investing in your future" and the Google Research Scholar Grant.

Appendix A. Explainable RL: Related work discussion

We discuss here some works in the field of eXplainable RL that share characteristics with the current work.

Sun and Sessions (2000) proposed a hybrid model combining RL algorithms which they created as a generator of reactive plans, and probabilistic planning algorithms to extract explicit, explainable plans for RL algorithms. In a specific problem scenario, plans can be generated in-place using a beam-seeking strategy that chains actions with optimal Q values at each step. The novelty of the work lies in a two-step process where the algorithm in first place performs a closed-loop policy acquisition and at a second step, an open-loop policy acquisition or semi-open loop policies. A beam search algorithm using an A* type heuristic performs

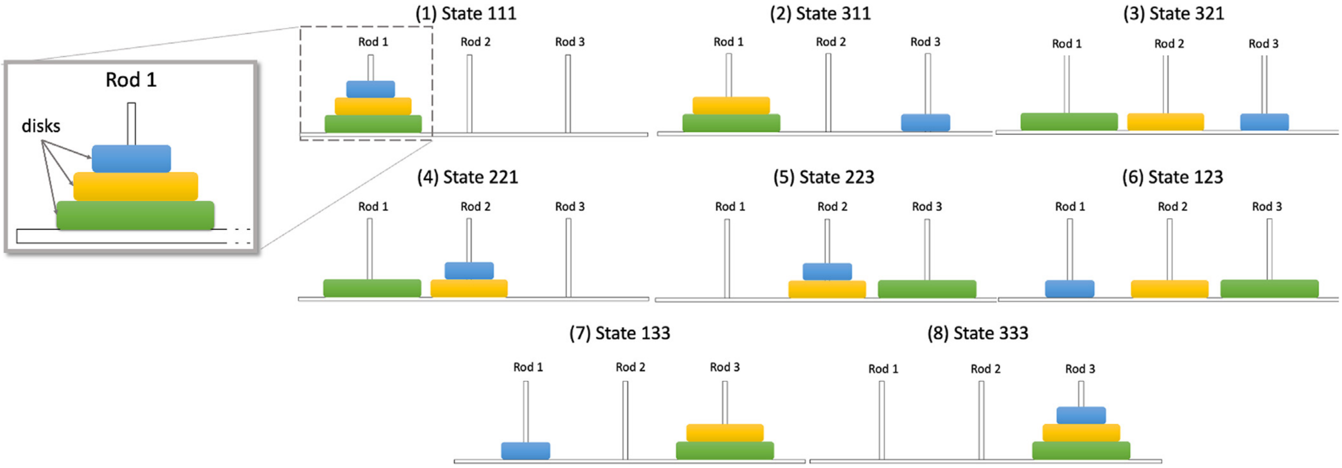


Fig. B.12. Schematic representation of the TOH with $N = 3$ disks and a specific sequence of moves {"1-3", "1-2", "3-2", "1-3", "2-1", "2-3", "1-3"} presented in Table 2(b) and with green arrows in Fig. B.13. The task's objective is to move the stack from first left-most rod (Rod 1) to the final right-most rod attending to the game rules (Rod 3). The minimum number of moves required to solve a TOH with N disks puzzle is $2^N - 1$. Thus for $N = 3$, the puzzle can be solved in 7 moves and goes through 8 different states including the starting state (111) and a final state (333). The encoding is such that each state has a unique identifier of 3 digits. Each digit encodes the position of a disk on a rod. The first digit encodes the position of the blue (smallest) disk, the second digit the yellow (medium size) disk, and the last digit encodes the position of the green (largest) disk. Each digit can take 3 possible values: "1" when the associated disk is in Rod 1, the value "2" when it is on Rod 2, and the value "3" when it is on Rod 3. For example, state "111" indicates that the three disks are all stacked on Rod 1 and state "321" encodes that the small blue disk is on Rod 3, the yellow mid-size disk is on Rod 2 in the middle, and the largest, the green disk is on the first rod. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the plan extraction process. Although this work is a precursor in explainable RL and aims at explaining how a sequential decision-making model can learn to plan explicitly without much a priori domain knowledge, it produces plans that can be explained as finite sequences of action specifications that lead to a goal. In our work, however, the sequences are not the explanation but the information to be explained, i.e., implicit knowledge. While this work focuses on explaining the RL algorithm for planning, our objective is to understand the evolution of knowledge acquisition of agents, that can be deduced from the action sequences.

Wang et al. (2018a) proposed DQNViz, an application-specific visualization tool for Deep-Q Reinforcement Learning Networks. Its purpose is to explore the AA's space of action to identify and extract AA's patterns of action or movement or reward. Through the alliance of visualization techniques and algorithmic tools (regular expressions, principal component analysis, dynamic time warping, and hierarchical clustering), the system allows the identification of AA behaviors and their investigation at particular moments. The tool can provide explanations about "what the agent did at what time?" and "what are the most activated layers of the neural network used at that moment?". Applied to certain AA behaviors, DQNViz allowed researchers to explore what caused bad behaviors for AA (such as hesitation or repetition). The work of Wang et al. (2018a) and the current one tend both to explore the AA space of action to explain the AA behaviors and provide visual explanations. However, DQNViz was made for Deep Learning architecture and Deep Learning experts (the usability for non-experts remains to be seen (Wells & Bednarz, 2021)), whereas we propose a multi-disciplinary approach using ML to extract a representation of the learning behavior of AA in the form of automata. Although we emphasize the importance of such work, we consider our work and motivations different from those of Wang et al. (2018a).

In Lapuschkin et al. (2019), the authors investigate the problem-solving behavior of AA to explicit its strategy. The authors proposed a semi-automated Spectral Relevance Analysis, referred to as SpRAy, to study the AA behavior in the context of arcade games and various tasks from the computer vision field. Based on heatmaps, SpRAy allows researchers to investigate the learned decision behavior of learning machines to detect the unexpected or undesirable ones or to label it "*from naive and short-sighted to well-informed and strategic*". Through this visual analysis of AA strategies, the authors question the efficiency of standard performance evaluation metrics, since they noticed that these can be oblivious in distinguishing diverse problem-solving behaviors. This work uses visual explanations to explore AA strategies, a line of work that we are aligned with. If the purpose of the authors is to characterize the strategies, ours is to study the building of the knowledge behind the strategy and its evolution. This work underlines the importance of visual explanation to investigate complex behaviors.

While previous works essentially complement ours in the sense of aiming at understanding AA problem-solving behavior, our work differs in one essential point: it focuses on analyzing the knowledge development and evolution in a learning AA.

Appendix B. A problem-solving task: The use case of the TOH

In this appendix, we describe first the TOH task and the data it generates in form of actions, i.e., sequences of moves, and second its interesting characteristics from a cognitive science perspective.

B.1. Description of TOH task

The TOH task involves, in its basic setting, three rods and a number of disks N of different incremental sizes. There are a set of specific rules to move the disks. Fig. B.12 displays a schematic representation of the TOH with three disks and rods. The task starts with all N disks stacked onto the left rod, in ascending order of size (the smallest at the top). As illustrated in the scenario in Fig. B.13, the objective of the task is to move the stack of disks from the left-most rod to the right-most rod according to the following rules: (1) Only one disk can be moved at a time; (2) Only the top disk in a rod can be moved in each action; (3) Each rod must preserve the order among its disks: no disk can hold a

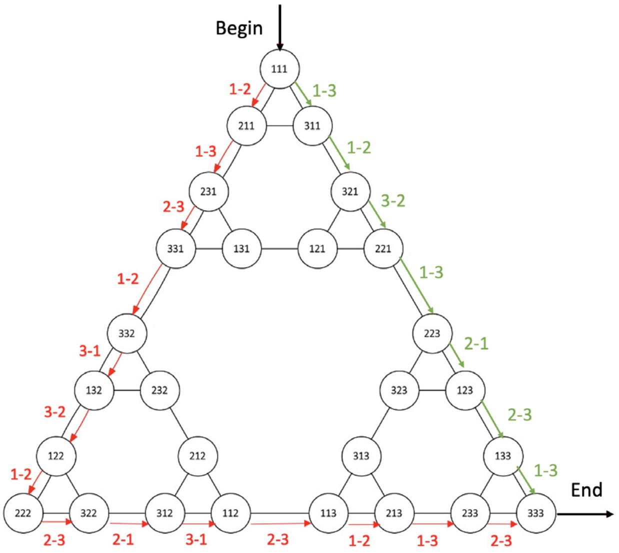


Fig. B.13. The TOH with $N = 3$ disks: graph representation of all the possible moves with 27 nodes and 39 edges. Each node represents a state of the task (i.e., the position of each disk on the rods). Each transition represents a possible action that leads to a new state. The represented task has starting (111) and final (333) states. Arrows display sequences of moves that allow succeeding in the task. Green arrows represent the optimized path, i.e., the minimum number of moves that correspond to the task strategy displayed in Fig. B.12, while it is also possible to deviate from the green optimal path (red arrows) to solve the task. For example, after the state "223", the agent might (faulty) move towards "323"-"313"-"213"-"233"-"333". (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

larger size one on top. Any disk can be moved on any rod as long as the moved disk is placed on top of a larger surface disk, or no other disk at all.

The graph in Fig. B.13 represents all potential moves that a player can make in a TOH task with $N = 3$ disks. Each node is named by a unique three digits identifier that indicates the position of the disks on the rods for the smallest, medium and large size disks, respectively. For example, state "111" indicates that the three disks are all stacked on rod number 1. Each transition of the graph represents an action, i.e., moving a disk from one rod to another and has a label that encodes the "starting rod - final rod" move; e.g., transition "1-2" between *begin* state "111" and "211" is read as: the first disk is moved from rod number 1 to number 2. With $N = 3$ disks, the TOH task has 27 states and can be solved in 7 moves (represented with green arrows in Fig. B.13). The minimal and thus, optimal number of moves required to solve the TOH task is $2^N - 1$. The task presents a finite set of 6 possible moves: "1-2", "1-3", "2-1", "2-3", "3-1" and "3-2". All sequences of moves start and finish with the same states (states "111" and "333" respectively in Fig. B.13), independently of the number of intermediate actions.

B.2. The TOH task to assess high-executive cognitive functions in problem-solving

The TOH is a prototype task of the category of transformation problems. It is widely used in cognitive science to assess high-executive functions in problem-solving. Among those, the TOH allows to track the activation patterns and networks in the brain using an fMRI as it involves complex stages of cognition and motor actions (Anderson, Albert, & Fincham, 2005; Dunbar, 1998). In neuropsychology, the TOH is one of the basics to evaluate executive functions such as problem-solving, cognitive flexibility, and response inhibition. As it relies on nonverbal planning, it is also used for cognitive remediation (Voelbel, Wu, Tortarolo, & Bates, 2016).

In cognitive modeling approaches, such as the current work, the TOH task is selected based on the fact that the optimal solution of this task involves the distinct cognitive strategy of inhibition. In novice agents, such as young children, it has been observed that other mechanisms emerge first starting from an unstable and misleading problem representation (Humphrey, 1951) such as in the form of exploration. Exploration is involved in the initial phase of the TOH task in children while the mobilization of the strategy of inhibition requires a representational restructuring (Ohlsson, 1992). While previous research such as in Gilhooly and Murphy (2005) has classified the TOH task as a non-insight task, the same authors discuss that in some cases the cognitive processes involved in tasks of insight and non-insight are blurred. Furthermore, we note that the current task classification made by Gilhooly and Murphy (2005) into insight and non-insight tasks is mainly based on experiments and perceptions of adult subjects. Their examination with young children and children of different age groups exhibit different cognitive processes in the solution of the TOH (Welsh, 1991). Lastly, the incremental nature of the TOH task allowed iterations with gradually more complex settings of the child or the artificial agent.

Appendix C. IKE-XAI framework implementation parameter setup and algorithms

In this appendix section, we provide the parameter setup for implementing the IKE-XAI framework. For each step, a pseudo-code is provided.

C.1. Step 1: Parameter setup and algorithm for the Q-learning algorithm

The AA here is a Q-learning algorithm (Watkins & Dayan, 1992) with a learning rate $\alpha = 1$, and a discount factor $\gamma = 0.8$. The exploration parameter is $\epsilon = 0$. The evaluation metric considered here is the average number of moves required to solve the task after. This metric is computed at different stages of training. Each stage is a two-step process: **(1)** the learning step of the AA that updates the Q-table according to the number of learning episodes, and **(2)** the playing step during which 100 sequences of moves are recorded. Sequences of moves and their length are recorded during the playing step of each training session at different stages of the training. We recorded the sequences (i) every 10 training episodes from 0 to 100, (ii) every 100 training episodes, between runs 100 and 1000, and (iii) every 500 training episodes, until reaching 8000 training episodes. Note that when the number of training episodes is 0, the AA plays from scratch without any training performed (i.e., with a non-updated Q-table). Finally, each recorded sequence of moves is processed as detailed in Section 4.1. The experiment was repeated 100 times to produce reproducible and robust results. Algorithm 2 presents the pseudo-code of step 1.

C.2. Step 2: Parameter setup and algorithm for the action prediction LSTM model

In our work, we implemented an RNN with three layers as described in Chraibi Kaadoud et al. (2022): an input and an output layer of 8 artificial neurons with sigmoïd activation function and a hidden layer of 8 LSTM cells. The learning phase was performed with the following parameters: a learning rate of 0.01, Adam

Algorithm 2: STEP 1 of IKE-XAI methodology: Algorithm for training the Q-learning agent to perform TOH with *N* disks .

```
Require:
  # Definition of game variables and rules according the number of disks N
  N = 3
  R = generate_reward_matrix(N)
  training_sessions = [0,100, 300, 500, 1000]
  learn_time= 100
  play_time=100

  Function AA_exploration_and_training((R, training_sessions, learn_time, play_time) :
  sequences={}
  for all n from learn_time do
    sequences_of_training_session={}
    for all i, Nb_episodes from enumerate(training_sessions) do
      Q= learn_Q(R, Nb_episodes)
      policy = get_policy(Q,R)
      sequence_states=[ ]
      for all j from play_time do
        start_state=get_starting_state(R)
        end_state = get_ending_state(R)
        state = start_state
        one_sequence=[ ]
        while state != end_state do
          state = random.choice(policy[state])
          one_sequence.append(state)
        end while
        sequence_states.append(one_sequence)
      end for
      sequences_of_training_session[i]= sequence_states
    end for
    sequences[n]= sequences_of_training_session
  end for

  return  sequences

  End Function
```

Algorithm 3: STEP 2 of IKE-XAI methodology: Training the LSTM model to learn sequences of TOH with *N* disks .

```
Require:
  # Definition of LSTM model parameters according the number of disks N
  toh_grammar= generate_grammar_from_reward_matrix(R)
  toh_dataset_of_sequences = generate_sequences(toh_grammar, 1000)

  Function LSTM_model_training(toh_dataset_of_sequences, number_units, number_LSTM_cells) :

  # Splitting data
  train_dataset = get_part_of_dataset(toh_dataset_of_sequences, 0.60)
  evaluation_dataset =get_part_of_dataset(toh_dataset_of_sequences, 0.20)
  test_dataset= get_part_of_dataset(toh_dataset_of_sequences, 0.20)

  # Building, training, validation and test of the LSTM model—–
  number_input_units = number_output_units= number_units
  LSTM_model.build_model(number_input_units, number_LSTM_cells, number_output_units)
  LSTM_model.learning(train_dataset)
  LSTM_model.evaluation(evaluation_dataset)
  LSTM_model.test(test_dataset)
  return  LSTM_model

  End Function
```

optimizer, and a number of epochs fixed at 10. The mean square error was used as the output evaluation metric. As a dataset, 1000 sequences of moves of various sizes were generated from the TOH abstract representation presented in Fig. B.13. 60% of the dataset

Algorithm 4: STEP 3 of IKE-XAI methodology: Algorithm for extracting rules in the form of a FSA using the hidden patterns of the LSTM model. Algorithm adapted from (Chraibi Kaadoud et al., 2022) .

```
Function XAI_generation_of_FSA (hidden_patterns, sequences, k) :
# Clustering ——————————————————-
clusters_list = k_means(k, hidden_patterns)
# Generation of automaton FSA——————————————————-
FSA = {} # Dictionnary
current_node= -1
FSA['nodes'].add(current_node)
FSA['edges'] = [ ] # list of dictionnaries
for all pattern $h$ of index $i$ from hidden_patterns do
  associated_cluster = clusters_list[$i$]
  if associated_cluster $\notin$ FSA['nodes'] then
    FSA['nodes'].add(associated_cluster)
  end if

  edge= {} # Dictionnary
  edge['id'] = (current_node, associated_cluster)
  if  edge $\notin$ FSA['edges']  then
    new_edge = edge
    new_edge['weight'] = 1
    new_edge['label'] = moves_list[$i$]
    FSA['edges'].add(new_edge)
  else
    edge['weight'] = edge['weight'] +1
    edge['label'] = edge['label']+ moves_list[$i$]
    FSA['edges'].update(edge)
  end if
  # Update of the current node
  current_node= associated_cluster
end for

plot_and_save(FSA)
return  FSA

End Function
```

was used for the training phase, 20% for the validation phase and 20% for the test phase. Sequences were randomly sampled. One-hot encoding was used to encode each move, providing a binary vector with a single non-zero unit. Algorithm 3 presents the pseudo-code of step 2.

C.3. Step 3: Parameter setup and algorithm for implicit rule extraction algorithm

This step represents the junction point where both RL (step 1 Fig. 2) and sequence learning (step 2 Fig. 2) meet. The sequences of moves recorded during the different training sessions of the AA represent its behavior. Each sequence is decomposed as a sample of "current move-next move". Each sample is input to the LSTM model, that should predict the next move (i.e., output) according to the current move (i.e., the input) and the past ones (the memory of the network). The dataset of hidden patterns obtained at the end of step 2 represents the starting point of the implicit rule extraction algorithm. We apply on the dataset a k-means clustering algorithm where k in [5, 100]. For each value of k, we computed the average silhouette score s. We choose the first value of k for which the value of s is the maximum. This allows extracting an intelligible FSA with the smallest possible number of nodes for which the clustering is good. We then perform, for that k value, the FSA generation algorithm of the FSA algorithm as described in Chraibi Kaadoud et al. (2022). The result of this step is the generation of a FSA that represents the behavior of the AA with weighted and labeled edges. These last ones can be interpreted as the actions made that induce the LSTM states transitions. Algorithm 4 presents the pseudo-code of step 3.

References

Alexandre, F., Hinaut, X., Rougier, N. P., & Viéville, T. (2021). Higher cognitive functions in bio-inspired artificial. *ERCIM News*, *125*, URL https://hal.inria.fr/hal-03189215.

Amir, D., & Amir, O. (2018). HIGHLIGHTS: summarizing agent behavior to people. In E. André, S. Koenig, M. Dastani, & G. Sukthankar (Eds.), *Proceedings of the 17th international conference on autonomous agents and multiagent systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018* (pp. 1168–1176). International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, URL http://dl.acm.org/citation.cfm?id=3237869.

Amitai, Y., & Amir, O. (2021). "I don't think so": Disagreement-based policy summaries for comparing agents. CoRR arXiv:2102.03064, URL https://arxiv.org/abs/2102.03064.

Anderson, J. R., Albert, M. V., & Fincham, J. M. (2005). Tracing problem solving in real time: fMRI analysis of the subject-paced Tower of Hanoï. *Journal of Cognitive Neuroscience*, *17*(8), 1261–1274.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115.

Bennetot, A., Charisi, V., & Díaz-Rodríguez, N. (2020). Should artificial agents ask for help in human-robot collaborative problem-solving? arXiv preprint arXiv:2006.00882.

Bhargava, S., & Purohit, G. (2011). Parsing with neural and finite automata networks: A graph grammar approach. *International Journal of Computer Applications*, *23*(4).

Breazeal, C., & Scassellati, B. (2002). Robots that imitate humans. *Trends Cognitive Siences*, *6*(11), 481–487.

Cangelosi, A., & Schlesinger, M. (2018). From babies to robots: the contribution of developmental robotics to developmental psychology. *Child Development Perspectives*, *12*(3), 183–188.

Carpenter, W. (2019). The *Aha!* moment: The science behind creative insights. In *Toward super-creativity-improving creativity in humans, machines, and human-machine collaborations*. IntechOpen.

Chaput, R., Cordier, A., & Mille, A. (2021). Explanation for humans, for machines, for human-machine interactions? In *Proceedings of the explainable agency in artificial intelligence WS, AAAI, February, 2021, virtual conference, United States* (pp. 4762–4764). AAAI Press.

Charisi, V., Díaz-Rodríguez, N., Mawhin, B., & Merino, L. (2022). On children's exploration, *Aha!* moments and explanations in model building for self-regulated problem-solving. In *IJCAI-ECAI workshop on AI evaluation beyond metrics*.

Charisi, V., Gomez, E., Mier, G., Merino, L., & Gomez, R. (2020). Child-robot collaborative problem-solving and the importance of child's voluntary interaction: A developmental perspective. *Frontiers in Robotics and AI*, *7*, 15.

Charisi, V., Liem, C. C., & Gomez, E. (2018). Novelty-based cognitive processes in unstructured music-making settings in early childhood. In *2018 joint IEEE 8th international conference on development and learning and epigenetic robotics (ICDL-EpiRob)* (pp. 218–223). IEEE.

Charisi, V., Merino, L., Escobar, M., Caballero, F., Gomez, R., & Gómez, E. (2021). The effects of robot cognitive reliability and social positioning on child-robot team dynamics. In *2021 IEEE international conference on robotics and automation (ICRA)* (pp. 9439–9445). IEEE.

Chraibi Kaadoud, I., Fahed, L., & Lenca, P. (2021). Explainable AI: a narrative review at the crossroad of knowledge discovery, knowledge representation and representation learning. In *Twelfth international workshop MRC-HCCS held at IJCAI-21* (pp. 6–18).

Chraibi Kaadoud, I., Rougier, N., & Alexandre, F. (2022). Knowledge extraction from the learning of sequences in a long short term memory (LSTM) architecture. *Knowledge-Based Systems*, *235*, Article 107657. http://dx.doi.org/10.1016/j.knosys.2021.107657, URL https://www.sciencedirect.com/science/article/pii/S0950705121009199.

Chronicle, E. P., MacGregor, J. N., & Ormerod, T. C. (2004). What makes an insight problem? The roles of heuristics, goal conception, and solution recoding in knowledge-lean problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(1), 14.

Chu, Y., & MacGregor, J. N. (2011). Human performance on insight problem solving: A review. *J. Probl. Solving*, *3*(2), http://dx.doi.org/10.7771/1932-6246.1094.

Cleeremans, A., Servan-Schreiber, D., & McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation*, *1*(3), 372–381.

Colas, C., Karch, T., Sigaud, O., & Oudeyer, P.-Y. (2021). Intrinsically motivated goal-conditioned reinforcement learning: a short survey. arXiv:2012.09830.

Colin, T. R. (2020). Intentions and creative insights: a reinforcement learning study of creative exploration in problem-solving. (Ph.D. thesis), University of Plymouth.

Colin, T. R., & Belpaeme, T. (2019). Reinforcement learning and insight in the artificial pigeon. In *41st annual meeting of the cognitive science society (CogSci 2019)* (pp. 1533–1539). Cognitive Science Society.

Díaz-Rodríguez, N., Lamas, A., Sanchez, J., Franchi, G., Donadello, I., Tabik, S., et al. (2022). Explainable neural-symbolic learning (X-Nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monuMAI cultural heritage use case. *Information Fusion*, *79*, 58–83.

Donald, D. R., Lazarus, S., & Lolwana, P. (2006). *Educational psychology in social context*. Oxford University Press.

Doncieux, S., Bredeche, N., Goff, L. L., Girard, B., Coninx, A., Sigaud, O., et al. (2020). DREAM architecture: a developmental approach to open-ended learning in robotics. arXiv preprint arXiv:2005.06223.

Doncieux, S., Filliat, D., Díaz-Rodríguez, N., Hospedales, T., Duro, R., Coninx, A., et al. (2018). Open-ended learning: a conceptual framework based on representational redescription. *Frontiers in Neurorobotics*, 59.

Donnarumma, F., Maisto, D., & Pezzulo, G. (2016). Problem solving as probabilistic inference with subgoaling: explaining human successes and pitfalls in the tower of hanoï. *PLoS Computational Biology*, *12*(4), Article e1004864.

Doshi-Velez, F., & Kim, B. (2017). A roadmap for a rigorous science of interpretability. arXiv arXiv:1702.08608.

Dunbar, K. (1998). Problem solving. *A Companion To Cognitive Science*, *14*, 289–298.

Durand, S. (1995). TOM, une architecture connexionniste de traitement de séquences: application à la reconnaissance de la parole. (Ph.D. thesis), Nancy 1.

Edwards, A. D., Downs, L., & Davidson, J. C. (2018). Forward-backward reinforcement learning. arXiv preprint arXiv:1803.10227.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Engelhardt, R., Lange, M., Wiskott, L., & Konen, W. (2021). Shedding light into the black box of reinforcement learning. In *Proceedings of the workshop "trustworthy AI in the wild", kl2021 held at 44th german conference on artificial intelligence* (pp. 1–2).

Ettlinger, M., Margulis, E. H., & Wong, P. (2011). Implicit memory in music and language. *Frontiers in Psychology*, *2*, 211.

Gers, F., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. *Neural Computation*.

Gilhooly, K. J., & Murphy, P. (2005). Differentiating insight from non-insight problems. *Thinking & Reasoning*, *11*(3), 279–302.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An approach to evaluating interpretability of machine learning. CoRR arXiv:1806.00069, URL http://arxiv.org/abs/1806.00069.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: a search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. http://dx.doi.org/10.1109/TNNLS.2016.2582924.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, *51*(5), 93.

Hayes, B., & Shah, J. A. (2017). Improving robot controller transparency through autonomous policy explanation. In *2017 12th ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 303–312). IEEE.

Hélie, S., Proulx, R., & Lefebvre, B. (2011). Bottom-up learning of explicit knowledge using a Bayesian algorithm and a new hebbian learning rule. *Neural Networks*, *24*(3), 219–232. http://dx.doi.org/10.1016/j.neunet.2010.12.002.

Heuillet, A., Couthouis, F., & Díaz-Rodríguez, N. (2020). Explainability in deep reinforcement learning. *Knowledge-Based Systems*, Article 106685.

Heuillet, A., Couthouis, F., & Díaz-Rodríguez, N. (2022). Collective explainable AI: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with Shapley values. *IEEE Computational Intelligence Magazine*, *17*(1), 59–71.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hoffman, R. R. (2014). *The psychology of expertise: cognitive research and empirical AI*. Psychology Press.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Automata theory, languages, and computation. *International Edition*, *24*(2).

Huang, S. H., Held, D., Abbeel, P., & Dragan, A. D. (2019). Enabling robots to communicate their objectives. *Autonomous Robots*, *43*(2), 309–326.

Humphrey, G. (1951). *Thinking: An introduction to its experimental psychology*. Wiley.

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, *50*(2), 1–35.

Jacobsson, H. (2005). Rule extraction from recurrent neural networks: Ataxonomy and review. *Neural Computation*, *17*(6), 1223–1263.

Jang, B., Kim, M., Harerimana, G., & Kim, J. W. (2019). Q-learning algorithms: A comprehensive classification and applications. *IEEE Access*, *7*, 133653–133667.

Kazak, Y., Barrett, C., Katz, G., & Schapira, M. (2019). Verifying deep-RL-driven systems. In *Proceedings of the 2019 workshop on network meets AI & ML* (pp. 83–89).

Kim, J. W., Ritter, F. E., & Koubek, R. J. (2013). An integrated theory for improved skill acquisition and retention in the three stages of learning. *Theoretical Issues in Ergonomics Science*, *14*(1), 22–37.

Klahr, D., & Robinson, M. (1981). Formal assessment of problem-solving and planning processes in preschool children. *Cognitive Psychology*, *13*(1), 113–148.

Kounios, J., & Beeman, M. (2014). The cognitive neuroscience of insight. *Annual Review of Psychology*, *65*, 71–93.

Lage, I., Lifschitz, D., Doshi-Velez, F., & Amir, O. (2019). Exploring computational user models for agent policy summarization. In S. Kraus (Ed.), *Twenty-eighth international joint conference on artificial intelligence, IJCAI 2019, Macao, China, August 10-16, 2019* (pp. 1401–1407). ijcai.org, http://dx.doi.org/10.24963/ijcai.2019/194.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*.

Langley, P., Meadows, B., Sridharan, M., & Choi, D. (2017). Explainable agency for intelligent autonomous systems. In S. P. Singh, & S. Markovitch (Eds.), *Proceedings of the thirty-first AAAI conference on artificial intelligence, February 4-9, 2017, San Francisco, California, USA* (pp. 4762–4764). AAAI Press, URL http://aaai.org/ocs/index.php/IAAI/IAAI17/paper/view/15046.

Lapalme, J. (2006). Composition automatique de musique à l'aide de réseaux de neurones récurrents et de la structure métrique. (Ph.D. thesis), Université de Montréal.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., & Müller, K. (2019). Unmasking clever hans predictors and assessing what machines really learn. CoRR arXiv:1902.10178, URL http://arxiv.org/abs/1902.10178.

Lawrence, S., Giles, C. L., & Fong, S. (2000). Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, *12*(1), 126–140.

Lefa, B. (2014). The piaget theory of cognitive development: an educational implications. *Educational Psychology*, *1*(9), 1–8.

Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., & Díaz-Rodríguez, N. (2020). Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, *58*, 52–68.

Lesort, T., Rodríguez, N. D., Goudou, J., & Filliat, D. (2018). State representation learning for control: An overview. CoRR arXiv:1802.04181, URL http://arxiv.org/abs/1802.04181.

Lieto, A., Bhatt, M., Oltramari, A., & Vernon, D. (2018). *The role of cognitive architectures in general artificial intelligence*. Elsevier.

Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, *15*(4), 151–190.

Madumal, P., Miller, T., Sonenberg, L., & Vetere, F. (2020). Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34 (03)* (2493–2500).

Mayer, R. E. (1977). *Thinking and problem solving: an introduction to human cognition and learning*. Scott, Foresman.

Mix, K. S., Bower, C. A., Hancock, G. R., Yuan, L., & Smith, L. B. (2022). The development of place value concepts: Approximation before principles. *Child Development*.

Murdoch, W. J., & Szlam, A. (2017). Automatic rule extraction from long short term memory networks. arXiv preprint arXiv:1702.02540.

Ohlsson, S. (1992). Information-processing explanations of insight and related phenomena. *Advances in the Psychology of Thinking*, *1*, 1–44.

Omlin, C., & Giles, C. (1996). Extraction of rules from discrete-time recurrent neural networks. *Neural Networks*, *9*(1), 41–52.

Omlin, C. W., & Giles, C. L. (2000). Symbolic knowledge representation in recurrent neural networks: Insights from theoretical models of computation. *Knowledge Based Neurocomputing*, 63–115.

Oudeyer, P.-Y., Kaplan, F., & Hafner, V. V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, *11*(2), 265–286.

Poissant, H., Poëllhuber, B., & Falardeau, M. (1994). Résolution de problèmes, autorégulation et apprentissage. *Canadian Journal of Education/Revue Canadienne de L'éducation*, 30–44.

Puiutta, E., & Veith, E. M. (2020). Explainable reinforcement learning: A survey. arXiv:2005.06247.

Reber, A. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior*, *6*(6), 855–863.

Reingold, E. M., & Ray, C. A. (2002). Implicit cognition.

Rojat, T., Puget, R., Filliat, D., Del Ser, J., Gelin, R., & Díaz-Rodríguez, N. (2021). Explainable artificial intelligence (XAI) on timeseries data: A survey. arXiv preprint arXiv:2104.00950.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Schellhammer, I., Diederich, J., Towsey, M., & Brugman, C. (1998). Knowledge extraction and recurrent neural networks: An analysis of an elman network trained on a natural language learning task. In *Proceedings of the joint conferences on new methods in language processing and computational natural language learning* (pp. 73–78). Association for Computational Linguistics.

Schillaci, G., Hafner, V. V., & Lara, B. (2016). Exploration behaviors, body representations, and simulation processes for the development of cognition in artificial agents. *Frontiers in Robotics and AI*, *3*, 39.

Schulz, L. (2012). The origins of inquiry: Inductive inference and exploration in early childhood. *Trends in Cognitive Sciences*, *16*(7), 382–389.

Servan-Schreiber, D., Cleermans, A., & McClelland, J. (1988). *Encoding sequential structure in simple recurrent networks*. Pittsburgh, PA, CMU-CS-88-183: School of Computer Ssience, Carnegie-Mellon UNiversity.

Siegel, M. H., Magid, R. W., Pelz, M., Tenenbaum, J. B., & Schulz, L. E. (2021). Children's exploratory play tracks the discriminability of hypotheses. *Nature Communications*, *12*(1), 1–9.

Smith, S. M. (1995). Fixation, incubation, and insight in memory and creative thinking. *The Creative Cognition Approach*, *135*, 156.

Smith, A. (2003). *Grammar inference using recurrent neural networks*. California: Department of Computer Science, UNiversity of San Diego, www.cse.ucsd.edu/ atsmith.

Smith, L., & Yu, C. (2008). Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, *106*(3), 1558–1568.

Sridharan, M., & Meadows, B. (2019). Towards a theory of explanations for human–robot collaboration. *KI-Künstliche Intelligenz*, *33*(4), 331–342.

Strobelt, H., Gehrmann, S., Huber, B., Pfister, H., & Rush, A. M. (2016). Visual analysis of hidden state dynamics in recurrent neural networks. arXiv preprint arXiv:1606.07461.

Sun, R., & Sessions, C. (2000). Learning plans without a priori knowledge. *Adaptive Behavior*, *8*(3–4), 225–253. http://dx.doi.org/10.1177/105971230000800302.

Sutton, R. S., Barto, A. G., et al. (1998). *Vol. 135*, *Introduction to reinforcement learning*. MIT press Cambridge.

Tiňo, P., & Šajda, J. (1995). Learning and extracting initial mealy automata with a modular neural network model. *Neural Computation*, *7*(4), 822–844.

Topolinski, S., & Reber, R. (2010). Gaining insight into the "Aha" experience. *Current Directions in Psychological Science*, *19*(6), 402–405.

Towell, G. G., & Shavlik, J. W. (1993). Extracting refined rules from knowledge-based neural networks. *Machine Learning*, *13*, 71–101. http://dx.doi.org/10.1007/BF00993103.

Truby, J., Brown, R. D., Ibrahim, I. A., & Parellada, O. C. (2022). A sandbox approach to regulating high-risk artificial intelligence applications. *European Journal of Risk Regulation*, *13*(2), 270–294. http://dx.doi.org/10.1017/err.2021.52.

Van Fraassen, B. (1988). The pragmatic theory of explanation. *Theories of Explanation*, *8*, 135–155.

Van Steenburgh, J., Fleck, J., Beeman, M., & Kounios, J. (2012). In W. K. Holyoak, & R. Morrison (Eds.), *Insight*. Oxford Handbook of Thinking and Reasoning.

Vanderelst, D., & Winfield, A. (2018). An architecture for ethical robots inspired by the simulation theory of cognition. *Cognitive Systems Research*, *48*, 56–66.

Verma, A., Murali, V., Singh, R., Kohli, P., & Chaudhuri, S. (2018). Programmatically interpretable reinforcement learning. In J. G. Dy, & A. Krause (Eds.), *Proceedings of machine learning research*: *Vol. 80*, *Proceedings of the 35th international conference on machine learning, ICML 2018, StockholmsmäSsan, Stockholm, Sweden, July 10-15, 2018* (pp. 5052–5061). PMLR, URL http://proceedings.mlr.press/v80/verma18a.html.

Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., et al. (2017). Starcraft ii: A new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782.

Voelbel, G., Wu, Z., Tortarolo, C., & Bates, M. (2016). Executive dysfunction and processing speed predict nonverbal problem solving deficits in a substance use disorder population. *International Journal of Brain Disorders and Treatment*, *2*(011).

Wallkötter, S., Tulli, S., Castellano, G., Paiva, A., & Chetouani, M. (2021). Explainable embodied agents through social cues: a review. *ACM Transactions on Human-Robot Interaction (THRI)*, *10*(3), 1–24.

Wang, J., Gou, L., Shen, H.-W., & Yang, H. (2018a). Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE Transactions on Visualization and Computer Graphics*, *25*(1), 288–298.

Wang, Q., Zhang, K., II, A. G. O., Xing, X., Liu, X., & Giles, C. L. (2017). An empirical evaluation of rule extraction from recurrent neural networks. arXiv preprint arXiv:1709.10380.

Wang, Q., Zhang, K., Ororbia, I., Alexander, G., Xing, X., Liu, X., et al. (2018b). A comparison of rule extraction for different recurrent neural network models and grammatical complexity. arXiv preprint arXiv:1801.05420.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3–4), 279–292.

Weiss, G., Goldberg, Y., & Yahav, E. (2017). Extracting automata from recurrent neural networks using queries and counterexamples. arXiv preprint arXiv:1711.09576.

Weitz, K., Schiller, D., Schlagowski, R., Huber, T., & André, E. (2021). "Let me explain!": exploring the potential of virtual agents in explainable AI interaction design. *Journal Multimodal User Interfaces*, *15*(2), 87–98. http://dx.doi.org/10.1007/s12193-020-00332-0.

Wells, L., & Bednarz, T. (2021). Explainable AI and reinforcement learning—A systematic review of current approaches and trends. *Frontiers in Artificial Intelligence*, *4*, 48.

Welsh, M. C. (1991). Rule-guided behavior and self-monitoring on the Tower of Hanoï disk-transfer task. *Cognitive Development*, *6*(1), 59–76.

Yuan, L., Xiang, V., Crandall, D., & Smith, L. (2020). Learning the generative principles of a symbol system from limited examples. *Cognition*, *200*, Article 104243. http://dx.doi.org/10.1016/j.cognition.2020.104243, URL https://www.sciencedirect.com/science/article/pii/S0010027720300627.

Zeng, Z., Goodman, R. M., & Smyth, P. (1993). Learning finite state machines with self-clustering recurrent networks. *Learning*, *5*(6).