Front. Digit. Educ., 2024, 1(2): 198–212
https://doi.org/10.1007/s44366-024-0008-9

RESEARCH ARTICLE

Future of Education with Neuro-Symbolic AI Agents in Self-Improving Adaptive Instructional Systems

Richard Jiarui Tong[a], **Xiangen Hu**[b]

[a] Macao University of Science and Technology, Macao 999078, China

[b] The Hong Kong Polytechnic University, Hong Kong 100872, China

Abstract This paper proposes a novel approach to use artificial intelligence (AI), particularly large language models (LLMs) and other foundation models (FMs) in an educational environment. It emphasizes the integration of teams of teachable and self-learning LLMs agents that use neuro-symbolic cognitive architecture (NSCA) to provide dynamic personalized support to learners and educators within self-improving adaptive instructional systems (SIAIS). These systems host these agents and support dynamic sessions of engagement workflow. We have developed the never ending open learning adaptive framework (NEOLAF), an LLM-based neuro-symbolic architecture for self-learning AI agents, and the open learning adaptive framework (OLAF), the underlying platform to host the agents, manage agent sessions, and support agent workflows and integration. The NEOLAF and OLAF serve as concrete examples to illustrate the advanced AI implementation approach. We also discuss our proof of concept testing of the NEOLAF agent to develop math problem-solving capabilities and the evaluation test for deployed interactive agent in the learning environment.

Keywords large language models (LLMs), neuro-symbolic cognitive architecture (NSCA), adaptive instructional systems (AIS), open learning adaptive framework (OLAF), never ending open learning adaptive framework (NEOLAF), artificial intelligence in education (AIED), intelligent tutoring system (ITS), LLM agent

1 Introduction

The emergence of ChatGPT has fueled research efforts in large language models (LLMs), transcending the traditional boundaries of artificial intelligence (AI). While AI researchers strive to enhance the usability and output quality of LLMs across diverse domains, scholars in sectors like education are exploring the transformative impacts of LLMs to address sector-specific issues (Abd-Alrazaq et al., 2023; Leiker et al., 2023; Tu et al., 2023; Yan et al., 2023). In education field, this exploration spans content generation, such as assessment and instructional items; interactive interfaces, such as chat and dynamic hints; and AI reasoning, like assignment grading and lesson planning.

This paper focuses on the nexus of these research vectors. The subsequent sections are organized as follows:

Initially, the current utilization of LLMs in education is scrutinized. We analyze the essential property of LLMs that makes it suitable for education purposes. Furthermore, we investigate the limitations of employing vanilla LLMs that hinder their applicability in certain advanced AI educational applications. To address these challenges, we propose the adoption of agent frameworks to unleash the power of LLMs for a new breed of AI applications.

Following the agent trajectory, the research introduces never ending open learning adaptive framework (NEOLAF), an integrated neuro-symbolic cognitive architecture (NSCA) devised for modeling and constructing self-improving agents. The discourse highlights how agents, based on such architecture, can learn from experience, symbolizing a significant step towards leveraging LLMs to provide cognitive functions in next-generation agents. Delving into the operational mechanics of NEOLAF agent the knowledge–situation–task–action–result (KSTAR) representation and process and introduce the concept of task-specific domain specific language (DSL) to agent to *think* and *learn* to perform goal-oriented cognitive actions.

The discussion then pivots to delineating how

Received January 10, 2024; accepted May 7, 2024

Richard Jiarui Tong (✉)
E-mail: richard.tong@ieee.org

NEOLAF agents can be integrated into the open learning adaptive framework (OLAF), a general adaptive instructional systems (AIS) and intelligent transport systems (ITS) architecture for developing scalable next-generation AI education solutions. This integration significantly amplifies the efficiency of ITS development—a currently laborious task demanding substantial resource allocation—without compromising quality.

In conclusion, we discuss our proof of concept testing of the NEOLAF agent to develop math problem-solving capabilities and the evaluation and feedback for deployed interactive agent in the real world learning environment.

2 LLMs for Contemporary Education

2.1 | Exciting AI Frontier for Education

LLMs, especially generative pre-trained transformer (GPT), can offer exciting opportunities in education at three progressive levels: generation, interface (both chat and other interactive interfaces), and planning and reasoning (Yadav, 2023). Before we dive into the agent-based architecture, we provide an overview of the background about the LLMs and the potential of LLMs in education. Then, we highlight how the limitations of the LLMs impact their usefulness in advanced educational applications.

2.2 | Essence of LLMs and Their Cognitive Functionalities

The unfolding narrative of LLMs, such as the GPT series, elucidates a pivotal juncture in the trajectory of AI (Brown et al., 2020; Joublin et al., 2023; Luitse & Denkena, 2021; Zhao et al., 2023). These behemoth models have showcased unparalleled prowess in understanding and generating text that closely mimics human language, rendering them indispensable in modern AI architectures (Schramowski et al., 2021). This subsection delves into the essence of LLMs, elucidating how they are poised to provide three quintessential levels of cognitive functionalities: generation, interface, and planning and reasoning, which are imperative for advancing human-centric AI applications.

2.2.1 Comparison Between LLMs and Compilers

At the crux, LLMs operate as compilers, transmuting text input into text output, anchored on ontology mapping and compression mechanisms. There are four commonalities between LLMs and compilers, including input processing, output generation, rules and patterns, and transformation of information.

Input processing. Both LLMs and compilers take in textual input. For LLMs, this is natural language text, while for compilers, it is a source code in a specific programming language (Aho et al., 2006).

Output generation. Both systems produce output based on their input. LLMs generate human-readable text, while compilers produce machine-executable code (Brown et al., 2020).

Rules and patterns. Both systems rely on predefined rules and patterns. Compilers use formal grammar rules of programming languages, while LLMs use statistical patterns learned from training data (Jurafsky et al., 2009).

Transformation of information. Both systems transform input information into different forms. Compilers convert high-level code to low-level machine instructions, while LLMs transform input prompts into contextually relevant responses (Vaswani et al., 2017).

However, there are eight distinct differences between LLMs and compilers, including purpose, processing approach, output determinism, error handling, learning and adaptation, intermediate representations, scope of knowledge, and execution model.

Purpose. Compilers translate human-readable source code into machine-executable instructions (Aho et al., 2006), while LLMs generate human-like text based on input prompts and learned patterns (Brown et al., 2020).

Processing approach. Compilers use deterministic algorithms and follow strict, predefined rules (Aho et al., 2006), while LLMs employ probabilistic methods, generating output based on learned patterns and statistical likelihoods (Bengio et al., 2003).

Output determinism. Compilers produce consistent, deterministic output for a given input (barring optimization variations) (Aho et al., 2006), while LLMs produce varied outputs for the same input due to their probabilistic nature (Radford et al., 2019).

Error handling. Compilers detect and report syntax errors and some logical errors in the source code (Aho et al., 2006), while LLMs do not have a concept of errors in the same sense. They aim for coherence and relevance but can produce incorrect or nonsensical outputs (Bender et al., 2021).

Learning and adaptation. Compilers are generally static, with their rules being hardcoded and only subject to change through manual updates (Aho et al., 2006). In contrast, LLMs are fine-tuned or adapted to new tasks with additional training data (Devlin et al., 2018).

Intermediate representations. Compilers often use multiple intermediate representations during the compilation process, such as abstract syntax trees and intermediate code (Aho et al., 2006). In contrast, LLMs operate on continuous vector representations of text,

without discrete intermediate stages (Mikolov et al., 2013).

Scope of knowledge. Compilers are limited to the specific programming language and target architecture they are designed for (Aho et al., 2006). In contrast, LLMs encompass a broad range of topics and generate content on various subjects (Brown et al., 2020).

Execution model. Compilers produce code meant for later execution by a computer (Aho et al., 2006), while LLMs generate output in real-time as part of their execution (Vaswani et al., 2017).

2.2.2 Unique Characteristics of LLMs

LLMs possess unique cognitive characteristics that stem from their pre-training and learning processes. The pre-training protocol like GPT, infused with reinforcement learning from human feedback (RLHF), bequeaths a robust foundational understanding prior to fine-tuning for specific tasks (Wei et al., 2021). The cognitive characteristics of LLMs burgeons with the enormity of the training datasets sourced from a diverse array of ontologies. These ontologies encompass disparate languages, programming paradigms, and a myriad of knowledge representations, underpinning the emergence of a range of wide-but-shallow general purpose cognitive characteristics, that can be super powerful but also unreliable, unpredictable and sometimes resulting in hallucination—a phenomenon where the model generates plausible but fictitious or unverifiable information.

2.2.3 Capabilities of LLMs

The capabilities of LLMs are categorized into: generation, interface, and planning and reasoning, as illustrated in Figure 1.

Generation. LLMs act as a universal translator or compiler, capable of transmuting any symbolic representation to and from text. This feature is instrumental for a plethora of applications encompassing natural language processing, code generation, and language translation among others.

Interface. The interface capability of LLMs is about engendering a seamless interaction between humans and machines. The human-in-the-loop (HITL) AI paradigm exemplifies this, where LLMs facilitate a symbiotic interaction. The advent of ChatGPT plugins heralds a new era, akin to the emergence of application stores, fostering a multi-modal sensing environment. The GPT-enabled application programming interfaces (APIs) become the next integration hub, extending the interface capabilities of LLMs, and providing a platform for the development and deployment of novel AI applications.

Planning and reasoning. The discourse on whether ChatGPT can think and reason resembles a contemporary rendition of the Turing test. It is incontrovertible that any generation task performed by ChatGPT is inherently a function of reasoning to some extent. By leveraging the knowledge enshrined in the LLMs parameters and employing mechanisms like pattern recognition and few-shot learning, ChatGPT exhibits a form of reasoning. This capability extends beyond rote pattern matching to a more nuanced understanding and inference-making, showcasing the potential of LLMs in performing cognitive tasks, and laying the groundwork for more advanced forms of reasoning in future AI systems.

2.3 | Overview of LLMs Capabilities in Contemporary Education

LLMs have marked a significant advancement in the application of AI in education. They have the potential to greatly enhance content generation, interactive learning interfaces, and analytical insights, which can be invaluable in personalizing education and improving learning outcomes (Figure 2).

LLMs and generative AI applications in adaptive instructional systems are illustrated in Figure 3. The diagram illustrates how LLMs and generative AI can be applied within the four key models of an AIS framework: the domain model, learner model, pedagogy model, and interaction model.

In the *domain model*, LLMs can generate assessment items and rubrics, instructional items, interactive items, lesson plans and learning goals, and

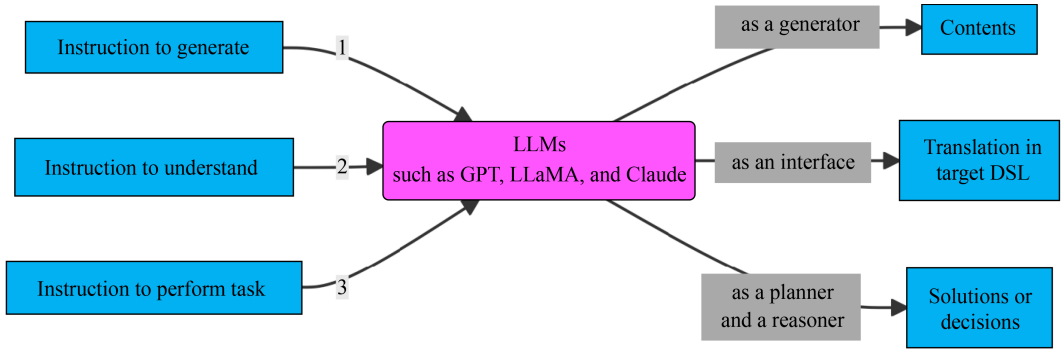

Figure 1 Capabilities of LLMs.

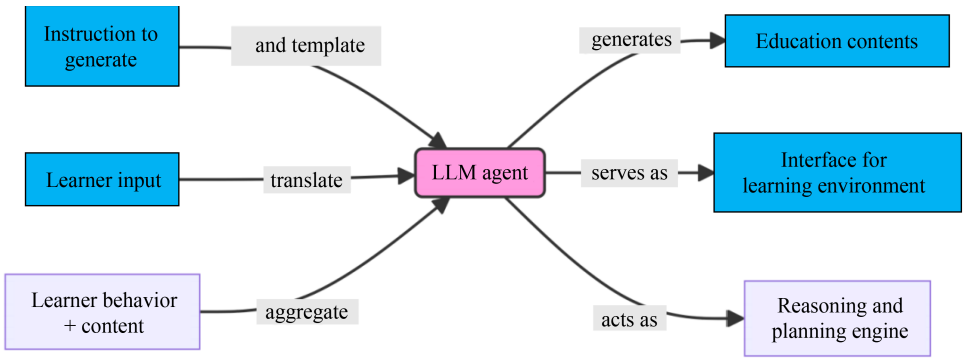

Figure 2 Overall capabilities of LLMs in the education field.

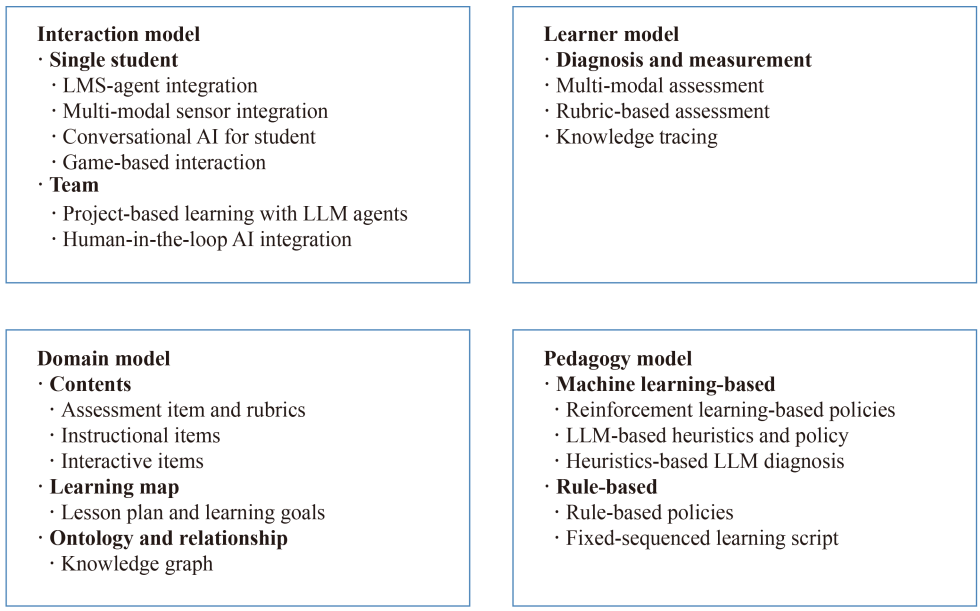

Figure 3 LLMs and generative AI applications in adaptive instructional systems.

construct knowledge graphs to represent the subject matter.

The *learner model* leverages AI for multi-modal and rubric-based diagnosis and assessment of learner performance, as well as knowledge tracing to model learner mastery over time.

In the *pedagogy model*, AI enables adaptive instructional strategies through reinforcement learning-based policies, LLM-based heuristics and policies, heuristics-based LLMs diagnosis, rule-based policies, and fixed-sequenced learning scripts.

Finally, the *interaction model* utilizes AI and LLMs to facilitate natural interactions between the learner and the ITS through LMS integration, multi-modal sensing, conversational AI, game-based interfaces, project-based learning with LLM agents, and human-in-the-loop AI support.

In summary, the integration of LLMs and generative AI across these four ITS models allows for highly personalized, engaging, and effective learning experiences that adapt to each learner's unique needs and characteristics.

2.4 Advantages and Disadvantages of LLMs

The core allure of LLMs like GPT, lies in their abilities to process and generate human-like text, drawing from a massive dataset to offer insightful responses or information. In the context of adaptive education, LLMs have significant potential. However, it is crucial to acknowledge their limitations, particularly in complex educational settings. Three limitations of LLMs include unpredictability, hallucination, and inability.

Firstly, the *unpredictability* of LLMs poses a serious issue. Despite extensive training data, LLMs generate output based on probabilistic methods, and they are unable to provide reasoning or justification for their responses. This unpredictability becomes glaringly evident in complex or nuanced scenarios, where it is crucial to align the output with established facts or well-founded logic.

Secondly, the *hallucination* of LLMs means that the incorrect or nonsensical information is produced.

This not only misleads students but also poses a risk in educational contexts, where accurate information is critical. Students relying on such erroneous output could develop a flawed understanding of the subject matter.

Thirdly, the *inability* to discern truth from falsehood is a substantial drawback of LLMs. In the absence of a axiom foundation framework, these models lack the capability for mathematical or logical reasoning. Despite providing answers that are likely correct based on their training data, LLMs do not actually know whether the information is accurate. This limitation is particularly problematic in disciplines like mathematics, physics, and philosophy, where axiomatic or logical reasoning is paramount.

Integrating LLMs with symbolic mechanisms can provide a counterbalance to these limitations. In such an architecture, the symbolic layer can act as a regulatory mechanism, vetting the probabilistic outputs of the language model against established axioms or rules. This integration ensures that the generated content is correct and grounded in logical reasoning. The hallucination issue can be mitigated by using the symbolic layer to cross-verify facts before they are presented. In essence, the symbolic integration imposes a structured and rule-based framework on the fluid and probabilistic nature of LLMs, making them more reliable and effective, particularly in complex educational scenarios.

The integration of agent architectures is critical in achieving a more harmonious blend of data-driven insights and rule-based reasoning. By combining LLMs with symbolic mechanisms, a more balanced and effective system can be created. In the NSCA, the language model serves as the intuitive and pattern-recognizing component, while the symbolic system takes care of structured reasoning and adaptation.

For instance, in a learning scenario, the agent could leverage the language model's capabilities to engage students and identify areas that need focus. The symbolic mechanism could then scaffold learning materials accordingly, even employing decision trees or rule-based algorithms to tailor future interactions. This integration provides robust framework for managing the complexity and individual variability inherent in educational settings.

3 Never Ending Open Learning Adaptive Framework

3.1 | Introduction of the NEOLAF

The NEOLAF is an agentic AI architecture based on LLMs as cognitive foundations. This framework reflects a step-change in education context, to develop intelligent, self-improving, and problem-solving agents (Tong & Lee, 2023). There are three key objectives of the NEOLAF. Firstly, it fosters the rapid development of explainable AI in education. Secondly, it facilitates the evolution of self-improving information processing systems. Thirdly, the NEOLAF introduces an agent capacity to learn from both humans and other machines after birth, reflecting a life-long learning paradigm. Fourthly, it forms the explicitly explainable learning processes. The NEOLAF draws inspiration from never-ending language learning (NELL) by Mitchell et al. (2018) and other cognitive architectures such as Soar and adaptive control of thought–rational (ACT–R).

3.2 | Conceptual Foundations

3.2.1 Neuro-Symbolic Architecture

The NEOLAF is designed as a neuro-symbolic architecture that combines the strengths of data-driven AI systems. This approach is inspired by Daniel Kahneman's distinction between System-1 heuristic-driven fast thinking and System-2 logical attention-driven slow thinking in human cognition (Kahneman, 2011).

Booch et al. (2020) extend this concept to AI systems, proposing that data-driven machine learning reflects human System-1 thinking, while symbolic approaches in AI reflect System-2 thinking. Within the NEOLAF, System-1 AI is represented by data-driven LLMs, and System-2 AI is implemented through KSTAR, a symbolic experience framing approach.

3.2.2 Comparison with Other Architectures

The NEOLAF differs from traditional cognitive architectures such as Soar and ACT–R in several ways. See Table 1 for details.

4 OLAF Solution Architecture for AI-Powered Education Software

Having established the development of intelligent, self-improving NEOLAF agents centered around the KSTAR process, we now delve deeper into its real-world applications within the open learning adaptive framework (OLAF). The NEOLAF's groundbreaking capabilities serve as a cornerstone of the OLAF architecture (Figure 4), bringing about a renewed perspective on AI-powered education software.

4.1 | Reasons for Agent-Based Architecture

Education is a dynamic and intricate domain, presenting a myriad of challenges that require personalized

Table 1 Cognitive architecture comparison

Cognitive architecture	Soar	ACT–R	NEOLAF
Model of cognition	Be based on symbolic reasoning and planning.	Be based on neural and cognitive processes.	Use both LLMs and symbolic approaches (System-1 and System-2 thinking).
Key focus	Be based on planning and problem-solving.	Be based on perception and attention.	Experience is equal to learning; use oracle for most of the self-learning.
Production	Be organized into a hierarchical network of sub-goals.	Be organized into a flat network.	Use *action* generated by LLMs based on *knowledge* of KSTAR for production.
Operators	Specify actions that can be taken to achieve a goal.	Specify the conditions that must be met for production rule to be executed.	Specify which sub-agent to execute the *action* steps encoded in task DSL.
Resulting states	Be produced by the actions specified by the operators.	Be represented by chunks, which store information about the current state.	The KSTAR is in working memory to guide the agent experience and all novel KSTAR experiences are memorized.

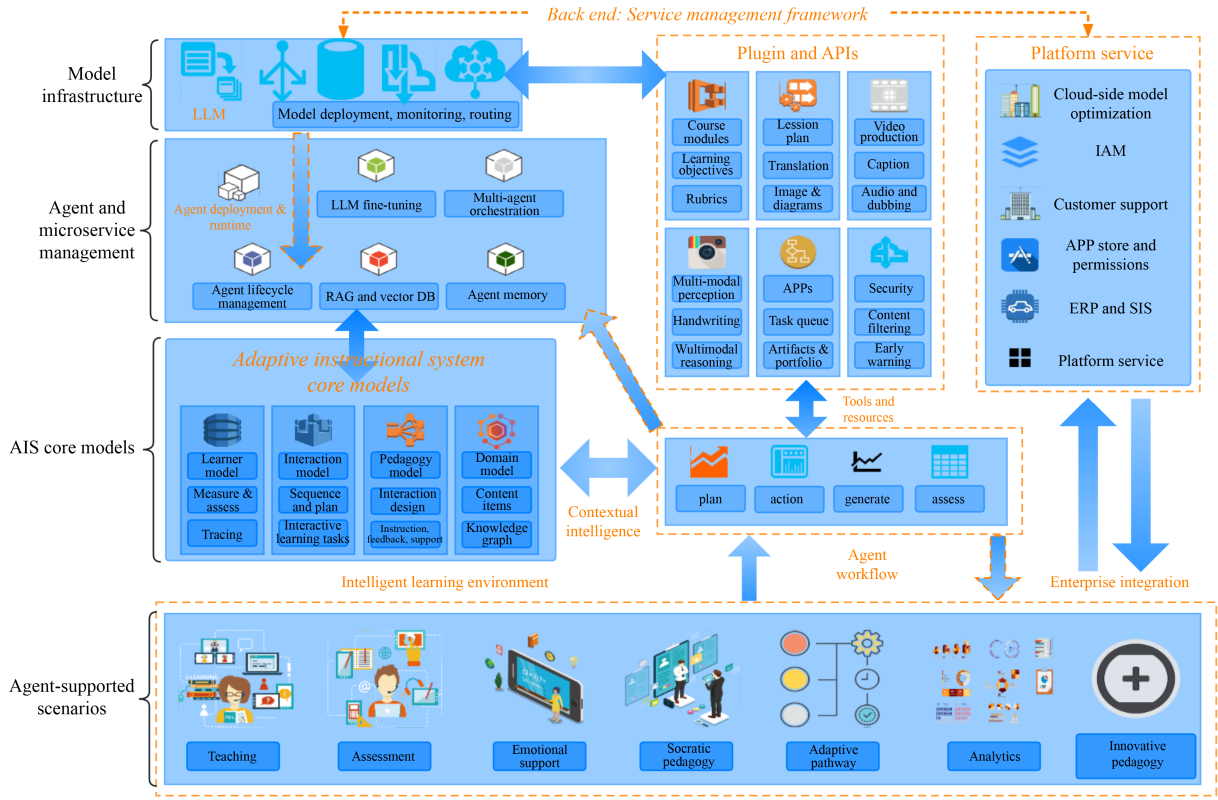

Figure 4 OLAF architecture illustration. LLM = large language model, RAG = retrieval-augmented generation, DB = database, AIS = automatic identification system, API = application programming interface, APP = application, IAM = identity and access management, ERP = enterprise resource planning, SIS = student information system.

and adaptive solutions. Traditional learning management systems (LMS), despite their widespread use, often fail to address these complexities. The LMS are built around fixed structures and deterministic logic, resulting in a limited capacity to handle diverse inputs, adapt to evolving contexts, and navigate ambiguous requirements.

However, the landscape of AI in education is undergoing a transformative shift, driven by the integration of agent-based workflows and LLMs. This combination is not only promising but essential for the future of educational technology, for several reasons: superior handling of complex inputs, adaptive contextual responses, and graceful navigation of ambiguity.

4.1.1 Superior Handling of Complex Inputs

LLMs, trained on extensive and varied datasets, excel in processing a broad spectrum of inputs. From handling structured academic assignments to open-ended student reflections, LLMs-powered agents seamlessly interpret and respond to a diverse ranges of content.

4.1.2 Adaptive Contextual Responses

The educational journey of every student is unique,

laden with specific challenges, aspirations, and learning curves. LLMs agents possess an innate ability to discern intricate nuances and contexts, ensuring that learners receive timely, relevant, and personalized feedback and guidance.

4.1.3 Graceful Navigation of Ambiguity

Education often treads in the realm of ambiguity. Whether it is interpretative questions, creative projects, or exploratory research, clear-cut answers are not always available. Traditional systems falter in these contexts, but LLMs agents thrive. Their capacity to navigate the gray areas, combined with a deep and semantic understanding derived from the NSCA, allows for a more nuanced and richer learning experience.

Integrating these LLMs agents within an architectural framework like the OLAF, especially when they are built upon principles of the NEOLAF, signifies a paradigm shift. Such a system, equipped with the strengths of neural processing, symbolic reasoning, and the capabilities of LLMs, offers a fluid, responsive, and deeply personalized educational journey, setting a new gold standard in AI-powered education.

4.2 | OLAF Architecture Overview

The holistic design of the OLAF agent architecture aims to seamlessly fuse AI capabilities into educational software. At the heart of this framework is the NEOLAF agent, a component tailored to leverage the LLMs to revolutionize learning experience. Its scalable and adaptable design, underpinned by the NEOLAF's principles, offers learners and educators personalized and intelligent experiences for optimized learning outcomes.

The OLAF represents a visionary approach to revolutionizing AI-powered education software. This architecture leverages the capabilities of LLMs to usher the next generation of personalized and adaptive learning experiences. In this section, we delve into the integration of the NEOLAF agent within the holistic system.

4.3 | Enhanced Information Architecture in the OLAF with NEOLAF Agents

The information architecture within the OLAF is a testament to its commitment to delivering a cohesive and enriched learning experience. This architecture goes beyond mere data structuring and organizing; it harnesses the power of AIS to provide a tailored learning journey for each individual. With the integration of the NEOLAF agents and their LLMs capabilities, the OLAF information architecture takes on a multi-dimensional, dynamic form, and far-surpassing traditional systems (Figure 5).

4.3.1 Learner Model

While traditional systems trace knowledge and generate analytics, the NEOLAF agents, powered by LLMs, take it a step further. They can dynamically model each learner's cognitive patterns, motivations, and learning styles, adapting in real-time to offer a truly personalized experience.

4.3.2 Domain Model

With the infusion of the NEOLAF's capabilities, the domain model transcends the mere cataloging of concepts and practices. It is about understanding them in depth, linking related concepts, identifying gaps in understanding, and providing resources to bridge those gaps. The synergy of the LLMs and neuro-symbolic architecture makes this possible, ensuring that learners have a holistic grasp of their subject matter.

4.3.3 Pedagogy Model

Pedagogy methods, activities, and rubrics are no longer static. The NEOLAF agents, through their goal-oriented cognitive actions, can dynamically adapt teaching methodologies based on the evolving needs of learners. Whether changing the teaching approach for a struggling student or introducing advanced topics for a fast learner, the model ensures optimal learning outcomes.

4.3.4 Interaction Model

Interaction model gets a significant boost from the NEOLAF's capabilities. Traditional feedback mechanisms are replaced with real-time, meaningful interactions. From guiding students through complex topics to facilitating peer interactions and collaborations, the LLMs-driven NEOLAF agents ensure that learning is not just about absorbing information but engaging deeply with it.

4.3.5 Information Architecture Summary

Incorporating NEOLAF agents and their LLMs capabilities into the OLAF information architecture is not merely an upgrade but a transformation. By fusing data-driven insights with deep semantic understanding, the OLAF ensures a learning environment that is adaptive, responsive, and deeply personal.

4.4 | NEOLAF in the Two-Stage Process of the OLAF

In the OLAF architecture, the construction

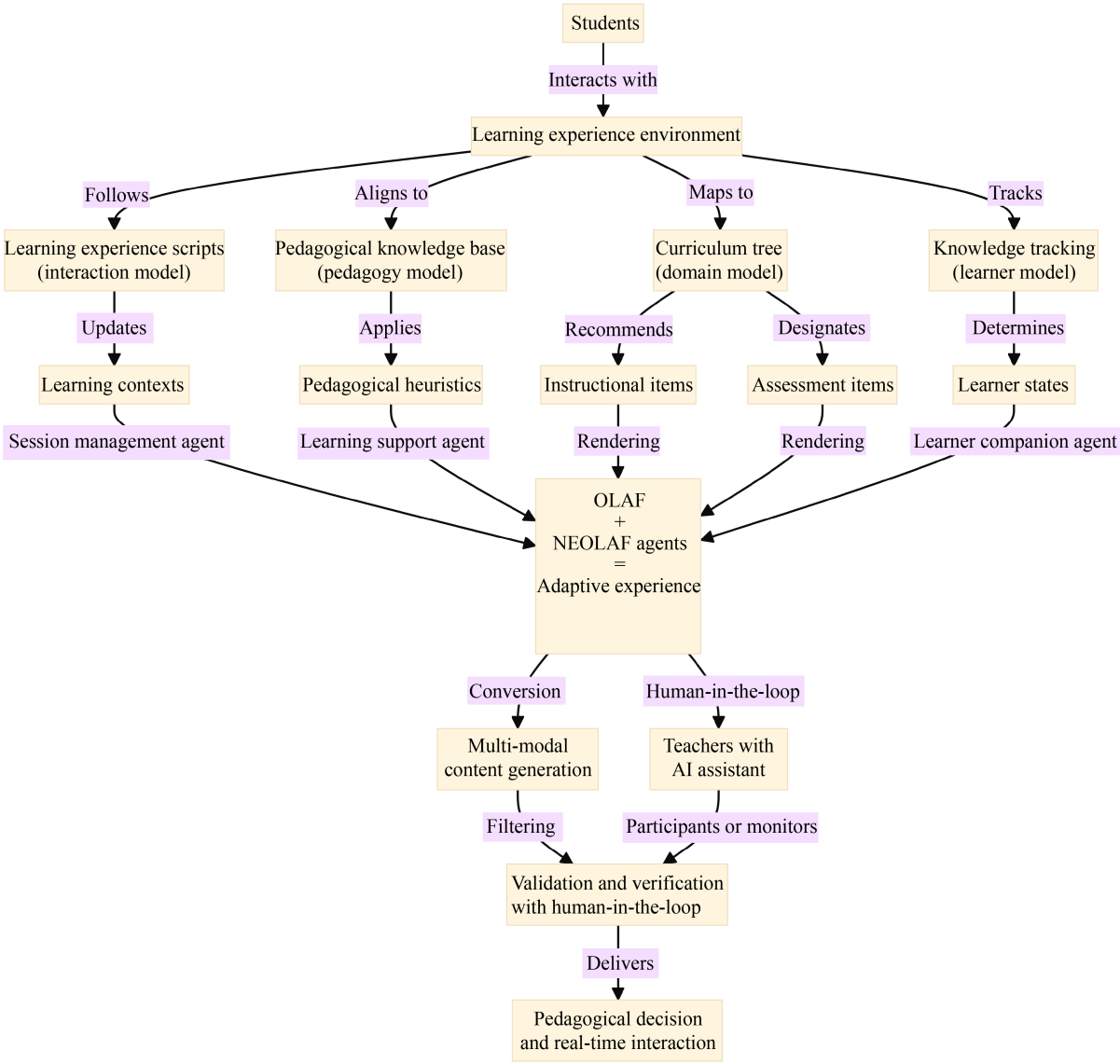

Figure 5 OLAF information architecture enhanced with the NEOLAF.

stage and operation stage are interconnected and continuously improved in a feedback loop as illustrated in Figure 6.

4.4.1 Integration of the NEOLAF in the OLAF's Construction Stage

In the construction stage of the OLAF (Figure 6), the focus is on harnessing the NEOLAF's self-learning capabilities. This stage is characterized by teachable agent framework, content generation, and software generation. The first trait of the construction stage is that OLAF acts as a *teachable agent* wherein humans teach agents the art of teaching, enabling them to learn and adapt teaching methodologies. The second trait is that OLAF facilitates *content generation* wherein agents use self-learning to create relevant and adaptable educational materials. The third trait of the construction stage is to catalyse *software generation*. The self-learning and teachable act process extends to the development of educational software, enhancing its functionality and user experience.

4.4.2 Role of NEOLAF in the OLAF's Operation Stage

In the operation stage, NEOLAF agents, trained in the construction stage, apply their skills in practice (Figure 7). There are two roles of NEOLAF in this stage, including dynamic student guidance and adaptive learning environment. The NEOLAF agents provide *dynamic student guidance* by leveraging their proficiency in teaching methodologies to lead students through their learning journey. In addition, the NEOLAF establishes *adaptive learning environment*. A collaborative team of agents creates a dynamic and responsive educational setting, tailored to the students' evolving needs and progress.

An example workflow as illustrated in Figure 8,

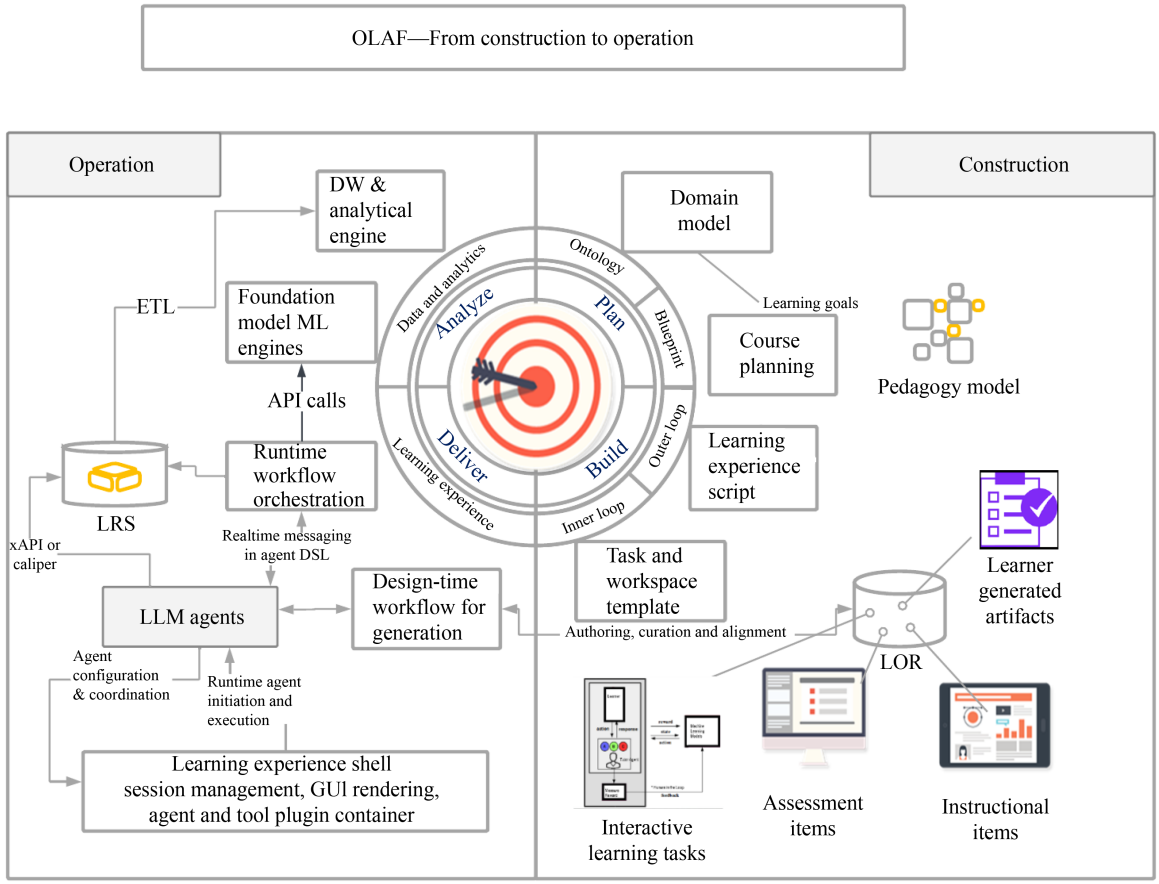

Figure 6 The OLAF construction and operation cycle. DW = Data Warehouse, ML = machine learning, ETL = extract, transform, and load, API = application programming interface, LRS = locally redundant storage, xAPI = experience application programming interface, DSL = Digital Subscriber Line, GUI = Graphical User Interface, LOR = learning object repository.

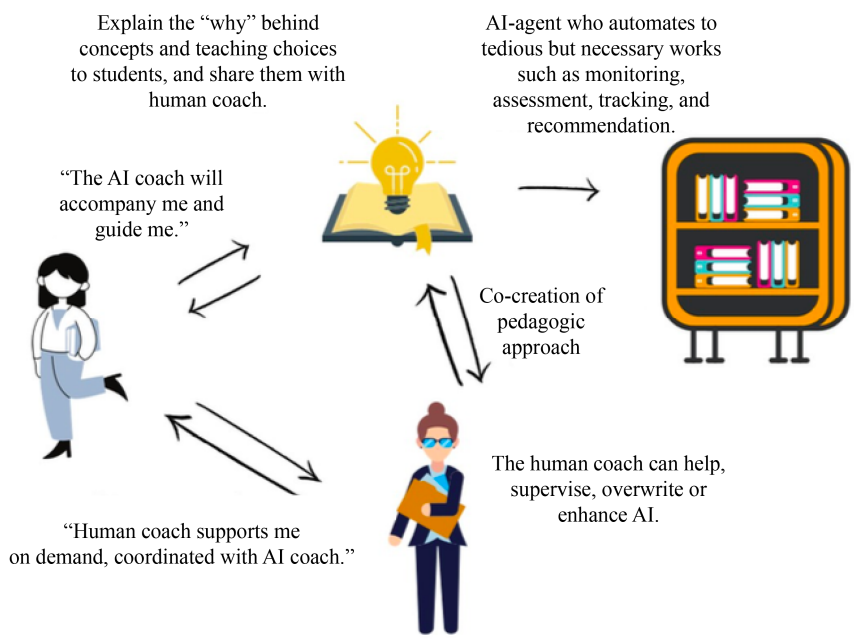

Figure 7 Agents in action in the OLAF to facilitate adaptive and human-in-the-loop in education.

demonstrates how a LLM agent is utilized in the operation stage. The process begins with the student taking a placement test, followed by the AI adaptive assistant agent grading and updating the diagnosis. The agent then generates a learning plan and recommends instructional items based on the diagnosis. If the agent reaches a confident diagnosis of the student's work, it proceeds; otherwise, it generates the next question and

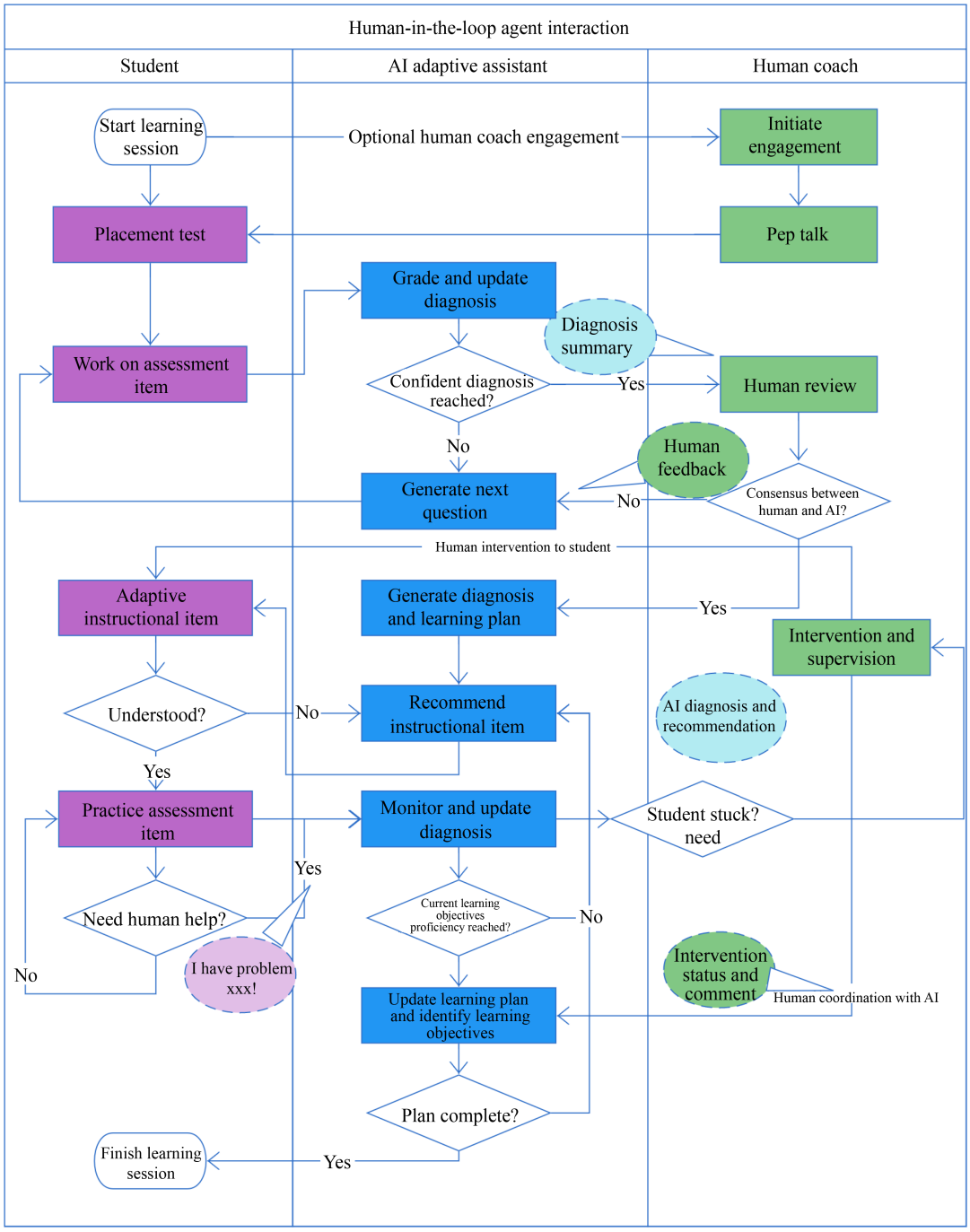

Figure 8 OLAF agent teaching and learning at the same time with human-in-the-loop.

repeat this process basing the pre-configured pedagogical policy. Throughout the learning session, human coaches can engage, provide feedback, and review AI recommendations, ensuring alignment between human and AI assessments. AI continuously monitors the student's progress, updating the learning plan until the learning objectives are met. If the student encounters difficulties or requests help, human intervention integrates AI recommendations with expert guidance, ensuring personalized support and effective learning outcomes.

5 Case Studies: NEOLAF Applications in Educational Settings

The NEOLAF represents a significant advancement in AI-powered educational technology. To demonstrate its versatility and effectiveness, we present two case studies that showcase NEOLAF's applications in diverse

educational contexts. These studies illustrate how NEOLAF tackles complex problem-solving tasks and provides nuanced error analysis, potentially revolutionizing both teaching methodologies and learning experiences.

5.1 | Case Study 1: NEOLAF in Advanced Math Problem-Solving Tasks

The first case study focuses on NEOLAF's capability to solve complex mathematical problems, a challenge for even the most advanced AI systems. This study aims to demonstrate how NEOLAF's unique architecture outperforms traditional AI approaches in tackling intricate and multi-step problems that require both extensive knowledge and precise reasoning.

5.1.1 Setup and Methodology

We develop a proof of concept (POC) to test NEOLAF agents in solving complex mathematical problems. The POC, conducted within a Python notebook, uses the problems and solutions from the American Invitational Mathematics Examination (AIME) as the source of challenging math problems. This choice is deliberate, as AIME problems are known for their difficulty and often require creative problem-solving approaches.

5.1.2 Integrated Tools and Utilities

The "problem solver bot" utilizes a sophisticated combination of four tools, including LLMs, mathematical tools, agent framework, and workflow framework. The first tool is *LLMs*, such as OpenAI's GPT (3.5 and 4.0), facilitating natural language understanding and generation. The second tool is *mathematical tools*, such as Wolfram Alpha and Python (Sympy), dealing with complex calculations and symbolic mathematics. The third tool is *agent framework*, such as Autogen, used for managing the operational dynamics of our agent. The fourth tool is *workflow framework*, such as OpenAI Assistant and Dify for optimizing the problem-solving workflow. These combinations allow us to leverage the strengths of each component, creating a system capable of understanding, planning, and executing complex mathematical operations.

5.1.3 Implementation Approach

The implementation follows a "divide and conquer" technique, distributing workload among multiple agents. This approach mirrors the collaborative approach of human experts solving complex problems and involves four steps. The first step is to retrieve prior knowledge from the KSTAR database, simulating how human experts draw on past experiences. The second step is to generate action plans using LLMs, mimicking the strategic planning phase of problem-solving. The third step is to switch to symbolic tools for complex calculations, similar to how a human might use a calculator for difficult computations. The fourth step is translating between natural language and tool-specific DSLs (domain-specific languages), representing the crucial skill of translating abstract mathematical concepts into concrete and executable steps.

5.1.4 Test Conditions and Results

We tested the system under three conditions to evaluate its performance. Firstly, the NEOLAF result before Oracle shows the system's base performance. Secondly, the NEOLAF result with Oracle to simulate the system's ability to learn from expert guidance. Thirdly, GPT-4 result serves as a baseline to compare against a state-of-the-art general-purpose AI.

Table 2 illustrates striking results. Before incorporating Oracle, the NEOLAF achieves 0% accuracy rate. Post-Oracle incorporation, the NEOLAF agent attains an 80% accuracy rate, a stark contrast to the 7% accuracy baseline established by GPT-4. These results demonstrate the power of NEOLAF's learning capabilities. While the initial performance is poor, the dramatic improvement after Oracle guidance showcases the system's ability to rapidly learn and apply new problem-solving strategies. This mimics the way human students might struggle with a new type of problem initially, but quickly improve with expert tutoring.

5.1.5 Key Insights

The POC reveals four critical insights, including KSTAR process, Oracle learning, tool utilization, and language translation. The centrality of the *KSTAR process* in navigating complex problems proves crucial. This structured approach to problem-solving allows the system to break down complex tasks into manageable steps. The effectiveness of *Oracle learning* mirrors the importance of expert guidance in human learning. The importance of optimal *tool utilization* becomes clear. Like a skilled human problem-solver choosing the right tool for each task, the system's ability to switch between different tools is key to its success. Language translation plays the critical role of accurate translation from natural language to tool-specific DSLs. This reflects the importance of precise communication in problem-solving, whether between humans or in human–computer interaction.

5.2 | Case Study 2: NEOLAF in Student Error Analysis

Building on the insights from the math problem-solving study, we next explore NEOLAF's potential in a more nuanced educational task—analyzing and categorizing student errors. This application aims to provide

Table 2 Comparison of AI solutions on math problems

Problem number	NEOLAF before Oracle	NEOLAF after Oracle	GPT-4	Category
1	0	1	0	Number theory
2	0	1	0	Geometry
3	0	1	1	Algebra
4	0	0	0	Geometry
5	0	1	0	Combinatorics
6	0	0	0	Algebra
7	0	1	0	Number theory
8	0	1	0	Algebra
9	0	1	0	Algebra
10	0	1	0	Number theory
11	0	1	0	Number theory
12	0	1	0	Algebra
13	0	1	0	Algebra
14	0	1	0	Geometry
15	0	0	0	Combinatorics

educators with deeper insights into student misconceptions, potentially allowing for more targeted and effective instruction.

5.2.1 Implementation and Methodology

We have implemented the NEOLAF framework as part an AI-powered error analysis system, designed to interact with students and diagnose the reasons behind their incorrect answers. The system is deployed in a controlled environment by Squirrel AI in China, involving three service centers and focusing on synchronized and normalized-process courses. This real-world testing environment allows us to evaluate NEOLAF's performance in actual educational settings.

5.2.2 Data Collection and Error Categories

Over a two-week period from May 5th to May 12th, 2024, the system analyzed 143 instances of student errors. While this sample size is relatively small, it provides valuable initial insights into both the operational efficacy of the system itself and common types of student error. The AI system categorized errors unrelated to knowledge component into several types, with the top three classifications being an incapacity to correctly diagnose the problem, a lack of attentiveness evidenced by insufficient question reading, and computational errors stemming from careless calculations.

Other categories include confidence in problem-solving capacity, feedback on misleading questions, and idle chat. This categorization provides a more nuanced understanding of student errors compared to simple right or wrong assessments.

5.2.3 System Performance and User Feedback

Initial testing reveals a diagnostic accuracy rate of approximately 70%, with the system accurately categorizing errors in 14 out of 20 cases. While there is room for improvement, this performance is promising for an initial deployment.

Importantly, two teachers provide positive feedback on AI's error diagnosis capabilities. They note that the system's efficacy in identifying common error types and providing useful insights into student misconceptions. This teacher feedback is crucial, as it validates the system's potential to provide actionable insights in real educational settings.

5.2.4 Challenges and Observations

Four challenges emerged during the case study, notably the intricacy in distinguishing between "misleading questions" and instances of "idle chat," a tendency for AI to overly persistent in seeking clarifications, premature termination of dialogues owing to constraints on conversational turns, and sporadic technical difficulties such as incomplete message transmissions and glitches in LaTeX formatting. These challenges highlight areas for future improvement and underscore the complexity of creating AI systems that can engage in nuanced, context-aware interactions with students.

5.2.5 Improvements and Future Work

Based on the case study results, several areas for improvement are identified: Firstly, the necessity to refine prompts is highlighted, with the objective of

reducing unnecessary repetition and improving the system's ability to pinpoint vital information swiftly. Secondly, expansion of the system's ability to accurately discern and classify complex scenarios, such as "misleading questions" and "idle chat." Thirdly, to enhance the validity and generalizability of the system's performance, increasing the sample size by expanding deployment across a broader spectrum of service centers and course types is recommended. Fourthly, to more accurately assess the AI's diagnostic accuracy and recall capabilities, manual labeling of 50 to 100 data points is proposed as a pivotal step. Fifthly, addressing technical issues, related to message transmission and LaTeX rendering, is crucial to ensure seamless operation. These improvements aim to enhance the system's accuracy and efficiency, and user engagement, making it more effective and user-friendly in real-world educational settings.

5.3 | Conclusions

These case studies demonstrate the versatility and potential of the NEOLAF framework in real-world educational settings. From solving complex mathematical problems to providing nuanced analysis of student errors, the NEOLAF shows promise in enhancing both the learning and teaching processes.

The math problem-solving case study showcases NEOLAF's ability to tackle high-complexity tasks, outperforming even advanced AI systems when provided with expert guidance. This suggests potential applications in advanced tutoring systems or as an aid for educators in developing challenging coursework.

The error analysis case study, while highlighting improvement, demonstrates NEOLAF's potential to provide educators with deeper insights into students' cognitive processes and misconceptions. This leads to more targeted instruction and personalized learning experiences.

Together, these studies depict an AI framework that significantly impacts education at multiple levels, from individual student support to curriculum development and teacher assistance. While challenges remain, the significant performance improvements observed highlight NEOLAF's potential to revolutionize AI-assisted education.

6 Future Outlook

The landscape of AI-powered education is rapidly evolving, with innovations like NEOLAF and OLAF showing remarkable promise. These architectures represent a significant leap forward in our approach to AI-assisted learning and problem-solving. By integrating System-1 LLMs capabilities with System-2 explicit reasoning and external services, and leveraging the KSTAR representation for both problem-solving and memory encoding, the NEOLAF and OLAF offer a more holistic and adaptable approach to AI in education.

6.1 | Key Advantages and Potential Impact

The combination of elements in NEOLAF, inspired by human cognition and merging symbolic and connectionist AI approaches, addresses many of the challenges associated with each method individually. It is envisaged that NEOLAF agents, powered by local LLMs, will exhibit competitive performance across a range of tasks while continually learning from experience. This trajectory could lead to the development of lightweight and ever-improving AI models, potentially offering a more efficient and economically alternative to current leading LLMs that are resource-intensive to train and maintain.

The potential impact of these architectures extends far beyond just being new AI-powered educational tools. They represent a vision for the future of education where learning becomes more intuitive, adaptive, and personalized. The integration of advanced architectures like NEOLAF and OLAF holds the potential to effect a fundamental shift in educational paradigm, making it more responsive to individual learners' needs and learning styles.

6.2 | Dawn of the Personal AI Era

Looking broader, neuro-symbolic agents like those embodied in the NEOLAF and OLAF are poised to become the foundation for the next generation in software and hardware development. These self-learning agents are spearheading the transition into what we call the personal AI (PAI) era. In this new paradigm, we envision a transformation of cloud-based LLMs and foundation models (FMs) into millions of highly personalized agents, each with embedded local-specific LLMs and FMs, operating either locally on devices or at the edge of networks.

This paradigm shift towards personalized AI agents has the potential to revolutionize not just education, but across numerous aspects of our daily lives. It promises a future where AI assistance is not just augmented in capacity, but also more tailored to individual needs, more privacy-conscious, and more readily available.

6.3 | Challenges and Limitations

While the outlook for NEOLAF and similar AI systems is promising, it is crucial to acknowledge the challenges

and limitations, encompassing LLMs hallucination, deployment costs, privacy and data security, ethical considerations, adaptability across domains, and technical robustness.

The first issue persists regarding the phenomenon of the *LLMs hallucination*, characterized by their tendency to produce false or nonsensical information, a matter of a significant concern, especially in educational contexts where accuracy is paramount. The second challenge is *deployment costs*. Despite the potential for more efficient models, the initial deployment of these advanced systems may still be resource-intensive and costly. The third critical issue is *privacy and data security*. As AI becomes more personalized, ensuring the privacy and security of individual data becomes increasingly important. The fourth issue is *ethical considerations*. The increasing role of AI in education raises important ethical questions about the balance between technological assistance and human instruction. The fifth issue is *adaptability across domains*. While promising in specific areas, ensuring these systems can adapt effectively across diverse educational domains and learning styles remain a challenge. The sixth issue is *technical robustness*. Addressing technical issues such as inconsistent performance, error handling, and integration with existing educational technologies will be vital for widespread adoption.

6.4 | Path Forward

Despite the aforementioned challenges, the potential benefits of NEOLAF, OLAF, and similar neuro-symbolic architectures in education are immense. The ongoing refinement of these systems necessitates concentrated efforts to address the identified limitations and challenges. This endeavor requires ongoing research, interdisciplinary collaboration, and a commitment to the principles of ethical and responsible AI development.

The future of AI in education is not just about smarter systems, but about creating more empathetic, adaptable, and personalized learning experiences. As we stand on the brink of the era of AI, architectures like NEOLAF and OLAF are paving the way towards a future where AI becomes a truly personalized learning companion, adapting and growing with each learner.

In conclusion, while challenges remain, the promise of these advanced AI architectures in revolutionizing education is clear. By continuing to innovate, address limitations, and focus on ethical implementation, we can work towards a future where AI-assisted education enhances human potential, fosters creativity, and makes lifelong learning a more accessible and enjoyable reality for all.

Acknowledgments We thank Squirrel AI for providing the deployment and testing environment for the interactive agents and their collaboration support. The process of writing this paper was greatly aided by generative AI models, including (but not exclusively) Microsoft's Bing, Anthropic's Claude, Mistral AI's Mistral, Google's Bard, Meta's LLaMA, and OpenAI' s GPT models.

Conflict of Interest The authors declare that they have no conflict of interest.

References

Abd-Alrazaq, A., AlSaad, R., Alhuwail, D., Ahmed, A., Healy, P. M., Latifi, S., Aziz, S., Damseh, R., Alrazak, S. A., & Sheikh, J. (2023). Large language models in medical education: Opportunities, challenges, and future directions. *JMIR Medical Education*, *9*, e48291.

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, techniques, & tools*. Boston: Addison-Wesley.

Bender, E. M., Gebru, T., McMillian-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In: *Proceedings of 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM, 610–623.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137–1155.

Booch, G., Fabiano, F., Horesh, L., Kate, K., Lenchner, J., Linck, N., Loreggia, A., Murugesan, K., Mattei, N., Rossi, F., & Srivastava, B. (2020). Thinking fast and slow in AI. *arXiv Preprint*, arXiv:2010.06002.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodel, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv Preprint*, arXiv:1810.04805.

Joublin, F., Ceravola, A., Deigmoeller, J., Gienger, M., Franzius, M., & Eggert, J. (2023). A glimpse in ChatGPT capabilities and its impact for AI research. *arXiv Preprint*, arXiv:2305.06087.

Jurafsky, D., Martin, J. H., Kehler, A., Linden, K. V., & Ward, N. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. New Jersey: Prentice Hall.

Kahneman, D. (2011). *Thinking, fast and slow*. New York: Farrar, Straus and Giroux.

Leiker, D., Finnigan, S., Gyllen, A. R., & Cukurova, M. (2023). Prototyping the use of large language models (LLMs) for adult learning content creation at scale. *arXiv PrePrint*, arXiv:2306.01815.

Luitse, D., & Denkena, W. (2021). The great transformer: Examining the role of large language models in the political economy of AI. *Big Data & Society*, *8*(2).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint*, arXiv:1301.3781.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., & Welling, J. (2018). Never-ending learning. *Communication of the ACM*, *61*(5), 103–115.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI.

Schramowski, P., Turan, C., Andersen, N., Rothkopf, C., & Kersting, K. (2021). Language models have a moral dimension. *arXiv Preprint*, arXiv:2103.11790.

Tong, R. J., & Lee, T. X. (2023). Trustworthy AI that engages humans as partners in teaching and learning. *Computer*, *56*(5), 62–73.

Tu, X., Zou, J., Su, W., & Zhang, L. (2023). What should data science education do with large language models? *arXiv Preprint*, arXiv:2307.02792.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv Preprint*, arXiv:1706.03762.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., Le, Q. V. (2021). Finetuned language models are zero-shot learners. *arXiv Preprint*, arXiv:2109.01652.

Yadav, G. (2023). Scaling evidence-based instructional design expertise through large language models. *arXiv Preprint*, arXiv:2306.01006.

Yan, L., Sha, L., Zhao, L., Li, Y., Martínez-Maldonado, R., Chen, G., Li, X., Jin, Y., & Gašević, D. (2023). Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, *55*, 90–112.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J., & Wen, J. (2023). A survey of large language models. *arXiv Preprint*, arXiv:2303.18223.