Exploring a Practice Teaching Mode for Electrical and Electronic Engineering Based on DeepSeek Agents

Qien Hou*
Huaibei Vocational and Technical College, Huaibei 235000, China
*Corresponding email: houqien@163.com

Abstract

The practical teaching of electrical and electronic engineering (EEE) is essential for cultivating students' engineering competence. However, it is persistently constrained by large class sizes, limited laboratory resources, stringent safety requirements, and the difficulty of providing timely, individualized guidance. Recent advances in large language models (LLMs), particularly open-source reasoning models such as DeepSeek-R1, coupled with the emergence of tool-using autonomous agents, offer a promising avenue for addressing these challenges. This paper proposes a practical teaching mode for EEE based on DeepSeek agents. This study designs a multi-agent architecture comprising a concept-explanation agent, an experiment-planning agent, a simulation and tool-use agent, a Socratic tutoring agent, and an assessment agent. These agents are orchestrated around concrete electrical and electronic tasks, such as circuit analysis, analog amplification, and digital logic design. A case study implemented in two undergraduate courses demonstrates the workflow. A mixed-methods evaluation indicates that the proposed mode improves learning engagement, shortens feedback latency, and enhances error-diagnosis ability. It also reveals risks, such as model hallucination and student over-reliance, which must be managed through human-in-the-loop safeguards. The findings provide a transferable blueprint for integrating open-source reasoning agents into hands-on engineering education.

Keywords

DeepSeek agent, Large language model, Electrical and electronic engineering, Practice teaching, Intelligent tutoring, Engineering education

Introduction

Electrical and electronic engineering (EEE) is a foundational discipline for modern engineering education. Its practical teaching - spanning circuit analysis, analog electronics, digital logic, and electrical machine control - serves as a bridge between abstract theory and authentic engineering practice. Through hands-on experiments, students learn to assemble circuits, operate instruments, interpret waveforms, localize faults, and cultivate the safety awareness and engineering judgement that classroom lectures alone cannot deliver. The quality of practice teaching therefore has a decisive influence on the formation of engineering competence.

However, practice teaching in EEE faces a set of long-standing and mutually reinforcing difficulties. First, growing enrollment has produced large classes in which a single instructor or laboratory assistant must supervise dozens of students simultaneously, making individualized, timely guidance difficult. Second, laboratory resources - benches, oscilloscopes, function generators, and prototype boards - are expensive to purchase and maintain, so effective hands-on time per student is limited. Third, safety constraints, particularly in power and motor experiments involving high voltage or high current, impose strict supervision ratios and restrict exploratory trial-and-error. Fourth, students are highly heterogeneous in prior preparation, so a uniform teaching pace leaves weaker students behind while stronger students are insufficiently challenged. These pressures have historically motivated the introduction of virtual and remote laboratories, which decouple some experiments from physical facilities, yet they still lack the adaptive, conversational, and explanatory guidance that a human tutor provides [1].

The rapid development of large language models (LLMs) has opened a new possibility. Foundation models have demonstrated broad competence in language understanding and generation [2]. Crucially, open-source reasoning models - most notably DeepSeek-R1, which was trained to produce long reasoning chains via reinforcement learning; DeepSeek-V3, an efficient mixture-of-experts backbone, and DeepSeek Math, specialized for mathematical reasoning - now deliver robust quantitative and logical capabilities at a fraction of the cost of proprietary models. Beyond mere text generation, the paradigm of LLM-based autonomous agents – in which a model interleaves reasoning with external actions such as invoking a calculator, simulator, or knowledge base - enables the construction of assistants. These assistants can not only answer questions but also plan experiments, execute simulations, and verify results. Despite this potential, most existing work on LLMs in engineering education focuses on general tutoring, assessment integrity, and prompt-based teaching strategies. Few systematic studies combine reasoning-and-acting agents with the practical teaching workflow for electrical and electronic engineering. This paper addresses that gap. Its contributions are threefold: (1) This paper proposes a practice teaching mode for EEE built on DeepSeek agents, articulated as a reusable multi-agent architecture. (2) This paper implements the mode in two undergraduate courses and describes representative cases spanning circuit analysis, analog amplification, and digital logic design. (3) This paper reports a mixed-methods evaluation of learning outcomes, engagement, and feedback latency, and this paper discusses the associated risks and safeguards.

Related work

Large language models and reasoning

Large language models are general-purpose foundation models trained at scale to predict text, and they have been shown to acquire a wide range of capabilities that transfer to downstream tasks. For engineering education, the most relevant capability is quantitative reasoning. DeepSeek-V3 is an open-source mixture-of-experts model whose efficiency makes large-scale deployment on campus infrastructure economically feasible [3]. DeepSeekMath demonstrated that an open model can approach the mathematical reasoning performance of much larger proprietary systems, which matters for circuit computation, Boolean algebra, and phasor analysis [4]. DeepSeek-R1 further showed that reinforcement learning can elicit long chain-of-thought reasoning and self-verification from a base model, markedly improving performance on mathematics and logical tasks. Because the reasoning traces of such models are readable, they can be exposed to students as worked examples, which aligns well with the step-by-step culture of EEE problem solving.

LLM-based autonomous agents

A LLM agent extends a language model with the ability to act. ReAct introduced the reason-and-act paradigm, in which a model alternates between generating a reasoning step and invoking an external action, thereby grounding its reasoning in observable feedback [5]. Toolformer showed that a model can teach itself to call external tools such as a calculator or a search engine without human annotation of tool-use trajectories [6]. A comprehensive survey systematized the field of LLM-based autonomous agents, identifying planning, memory, tool use, and multi-agent collaboration as the core components, and surveying applications from software engineering to scientific discovery [7]. These results imply that an educational agent for EEE can be equipped with a circuit simulator, a numerical solver, and a plotting tool and can iterate between hypothesis, simulation, and correction in a manner that mirrors real engineering debugging.

AI and LLMs in engineering education

Substantial literature has examined the role of LLMs in education. Kasneci et al. surveyed the opportunities and risks of LLMs for education, highlighting personalization, feedback generation, and the need for human oversight [8]. In the engineering domain specifically, Nikolic et al. benchmarked Chat Generative Pre-trained Transformer (ChatGPT) against engineering assessments across multiple institutions. They found that it could solve many examination problems, raising fundamental questions about assessment integrity and how such tools should be incorporated into the curriculum [9]. Heradio et al. provided a bibliometric analysis of virtual and remote laboratories, documenting decades of effort to extend laboratory access beyond physical facilities. More recently, generative intelligent tutoring systems built on Generative Pre-trained Transformer 4 (GPT-4) have been proposed with modular designs for future learning platforms [10]. What remains comparatively unexplored is the use of an open-source reasoning agent that can simultaneously plan experiments, employ simulation tools, and provide

Socratic feedback within the specific pedagogical workflow of EEE practice teaching - the central focus of this work.

Design of the DeepSeek-agent-based practice teaching mode

Overall pedagogical framework

The proposed mode follows a blended, student-centered structure organized into three phases. In the before-class phase, the agent helps students review the underlying theory, preview the experiment objectives, and prepare a design plan so that scarce laboratory time is spent on genuine manipulation rather than on first-time exposure to concepts. In the during-class phase, the agent acts as a just-in-time teaching assistant: It answers procedural questions, guides troubleshooting through Socratic questioning, and cross-checks student measurements against expected values computed by its tools. In the after-class phase, the agent supports reflection and report writing, providing rubric-based feedback on error analysis and helping students connect the experiment back to theory. Throughout all three phases, the instructor remains the authority and final arbiter: The agent is configured as an assistive, human-in-the-loop component rather than a replacement for the teacher, in line with the recommendations of prior work on LLM-assisted education.

Multi-agent architecture

The core of the model is a multi-agent architecture in which several specialized agents share a common memory, a tool registry, and a retrieval-augmented knowledge base built from the course syllabus, laboratory manuals, and safety regulations. The architecture is inspired by the planning - memory - tool use decomposition described in the agent survey and by the reason-and-act loop of ReAct. Five specialist agents cooperate under a lightweight orchestrator.

(1) Concept agent

This agent explains theoretical concepts and their engineering significance, derives key formulas (e.g., Kirchhoff’s laws, Thévenin’s theorem, the small-signal model, and Boolean simplification), and grounds its explanations in the course knowledge base through retrieval. This process reduces hallucination and maintains terminology consistency with the textbook.

(2) Experiment-planning agent

This agent decomposes a laboratory task into a sequence of concrete steps, proposes circuit topologies, estimates component values (e.g., biasing resistors for a given operating point), and flags safety constraints such as voltage limits and current ratings. It uses the chain-of-thought reasoning capability of DeepSeek-R1 to make its planning traceable and inspectable by the student.

(3) Simulation and tool-use agent

This agent wraps external tools - a Python numerical engine, a simulation program with integrated circuit emphasis-style (SPICE-style) circuit simulator, a scientific calculator, and a plotting library - following the tool-use paradigm of Toolformer. It executes simulations requested by the student or the planning agent and returns numerical values and waveforms, enabling theoretical expectations to be verified before any physical measurement.

(4) Socratic tutoring agent

This agent adopts a questioning style rather than providing direct answers. When a student reports a fault, it asks about the symptom, the last change made, and the expected versus measured values, guiding the student to isolate the fault progressively. This approach mirrors evidence-based tutoring strategies and is intended to prevent passive cognitive offloading.

(5) Assessment agent

This agent scores laboratory reports against a rubric, comments on error analysis and units, and produces individualized improvement suggestions. It also aggregates diagnostic information (e.g., recurring misconceptions) for the instructor, supporting data-informed teaching.

The orchestrator routes each student query to the appropriate specialist, maintains the conversation state in shared memory, and enforces a safety gate that blocks or defers requests involving hazardous operations until the instructor confirms them. This separation of concerns keeps individual prompts short and specialized, which improves reliability relative to a single monolithic prompt.

Mapping the architecture to EEE content

The architecture is organized around three representative EEE (Electrical and Electronics Engineering) content modules. The circuit analysis module covers direct current/alternating current (DC/AC) steady-state analysis, network theorems, and transient response. Within this module, the planning and simulation agents are central, as most tasks involve setting up and solving circuit equations. The analog electronics module covers diode and transistor circuits, as well as operational amplifiers. Here, the concept agent plays a significant role in explaining device behavior, while the simulation agent verifies bias points and gains. The digital logic and motor-control module covers combinational and sequential logic, counters, and basic drive circuits, with the tool-use agent performing logic simplification and timing simulation. The model also bridges the virtual and the physical: Students first complete an agent-guided simulation and design phase, then perform the actual experiment, and finally reconcile the simulated and measured results during the reflection phase - a practice long advocated in the virtual-laboratory literature [11].

Prompting, grounding, and safety

Several engineering measures enhance the system's reliability. System prompts constrain each agent's role, tone, and boundaries. Retrieval-augmented generation grounds factual and procedural answers in authoritative course documents, thereby mitigating hallucination - a recognized risk of using large language models (LLMs) in education. Wherever possible, numerical results are routed through verifiable tools instead of being generated as free text, and the Socratic agent is instructed to flag any uncertain statements. Safety guardrails prevent the direct execution of hazardous instructions and require instructor confirmation for experiments involving high voltage or rotating machinery. Finally, to uphold academic integrity, the assessment emphasizes process and reasoning over final answers. All agent-student interactions are logged for instructor review, addressing the integrity concerns associated with using generative AI in engineering assessment [12].

Implementation and case study

Deployment environment

The system was deployed in a university electrical and electronic engineering (EEE) laboratory setting. DeepSeek-V3 and DeepSeek-R1 were hosted on a local inference cluster, leveraging their open-source licenses and relatively low deployment costs. The tool layer consisted of a Python numerical engine (NumPy/SciPy), a SPICE-style circuit simulator, a scientific calculator, and a plotting library, all accessible to the agents via a tool registry. The knowledge base indexed course syllabi, laboratory manuals, component datasheets, and safety regulations and was queried through a retrieval module. A lightweight web interface enabled students to interact with the orchestrator, upload screenshots of waveforms, and receive simulation outputs inline. The system was integrated into two courses - Circuit Analysis (86 students) and Digital Electronics (72 students) - over one semester, running concurrently with conventional laboratory sessions.

A typical interaction workflow

A representative session proceeds as follows. A student states a task or uploads a question; the orchestrator classifies it and activates the appropriate specialist. For a design task, the planning agent produces a stepwise plan and estimates component values; the simulation agent then executes the corresponding circuit and returns waveforms or truth tables. The student builds the circuit physically and, if the measurement deviates, reports the symptom to the Socratic tutor, which asks diagnostic questions and suggests systematic checks. Finally, the assessment agent reviews the student's report against a rubric and delivers individualized comments. The instructor can review any thread at any time, and the safety gate intercepts hazardous requests before they reach the tool layer.

Case 1: DC circuit troubleshooting

In the Circuit Analysis Course, students were asked to assemble a two-loop resistive network and verify Kirchhoff's laws. When a student measured a branch current inconsistent with the predicted value, the Socratic tutor inquired whether the discrepancy was in sign or magnitude and prompted a check of the assumed current direction. It then used the simulation agent to demonstrate the effect of reversing one source polarity, helping the student localize the error to a mislabeled measurement reference. This interaction replaced waiting for the instructor and transformed a measurement error into a conceptual lesson about sign conventions.

Case 2: Common-emitter amplifier design

Students designed a common-emitter amplifier to meet a target voltage gain and output swing. The planning agent, using DeepSeek-R1's chain-of-thought reasoning, proposed a bias network and computed candidate resistor values for a specified quiescent point, the simulation agent returned the DC operating point and the small-signal gain. When the measured gain fell short, the tutor guided the student to examine the bypass capacitor and the load effect, and the simulation agent demonstrated how the gain varied with the emitter degeneration resistor. The agent's ability to interleave symbolic reasoning with quantitative tool calls enabled students to explore design trade-offs without monopolizing the instructor's time.

Case 3: Digital logic design and simulation

In the Digital Electronics Course, students implemented a modulo-6 counter. The concept agent reviewed state transitions and Boolean simplification; the tool-use agent performed Karnaugh-map simplification and produced a timing diagram, which students compared with the physical counter's output on the logic analyzer. Discrepancies involving a glitch were diagnosed through targeted questions about asynchronous versus synchronous inputs, demonstrating how an agent can support the debugging of time-dependent behavior that is otherwise difficult to explain in a large class.

Evaluation and discussion

Evaluation method

A mixed-methods evaluation was conducted. Quantitative instruments included a pre or post concept test on core topics, rubric scores for laboratory reports, and a Likert-scale questionnaire measuring engagement, perceived usefulness, and perceived feedback latency. Qualitative data were collected through semi-structured interviews with a purposive sample of students and through an analysis of agent-student interaction logs. For comparison, the same instruments were administered to a control cohort that completed the same laboratory exercises in the conventional manner without the agent. The comparison is illustrative rather than strictly causal, given the single-institution, single-semester scope.

Results

Students in the agent-supported cohort reported substantially shorter perceived feedback latency; most troubleshooting questions were answered within one to two minutes rather than after waiting in a queue for the instructor. Questionnaire responses indicated high engagement: A large majority agreed that the agent helped them understand errors rather than merely correcting them and that they were more willing to explore alternative designs because trial-and-error could first be carried out safely in simulation. Concept-test gains were comparable to or slightly better than those of the control cohort, with the largest difference observed in error-diagnosis items, where students had to identify the most likely fault given a set of symptoms. Interview data echoed this: students valued the Socratic style because it forced them to reason, and instructors reported that routine questions were deflected, freeing them to attend to higher-order problems. At the same time, a minority of students expressed concern that they might become dependent on the agent, and a small number of generated explanations required instructor correction, underscoring the residual hallucination risk.

Discussion

The results support three main benefits.

(1) Personalization: The agent adapts explanations to a student's specific error rather than repeating a fixed lecture.

(2) Scalability: Because DeepSeek models are open source and inexpensive to serve, a single deployment can support large cohorts at a low marginal cost.

(3) Timeliness: Just-in-time guidance shortens the loop between error and correction, which is central to effective practice-based learning. The architecture's separation of concerns also improves robustness, as each specialist prompt is narrow and easier to supervise than a single general prompt.

The risks must nevertheless be managed. Hallucination can produce plausible but incorrect circuit advice; this study mitigated this with retrieval grounding and tool-based verification of numerical claims. Over-reliance and cognitive offloading can undermine the very competence the laboratory is meant to build; the Socratic design and process-based assessment are intended to counteract this. Academic integrity and data privacy require logged, auditable interactions and policies on the acceptable use of generative assistance [13]. Finally, the evaluation is limited by its single-institution setting, modest sample size, and one-semester duration, so the results should be interpreted as preliminary evidence rather than generalizable proof.

Conclusion

This paper proposes a practical teaching model for electrical and electronic engineering based on DeepSeek AI agents, demonstrating its implementation through a multi-agent architecture. This architecture integrates concept explanation, experiment planning, tool-based simulation, Socratic tutoring, and rubric-based assessment. A case study conducted across two undergraduate courses demonstrated improvements in student engagement, feedback latency, and error-diagnosis capabilities. It also highlighted associated risks - such as AI hallucination, student over-reliance, and academic integrity concerns - that necessitate human-in-the-loop safeguards. The open-source nature of the underlying DeepSeek models (e.g., DeepSeek-R1, DeepSeek-V3) enhances the economic accessibility of this approach for a wide range of educational institutions. Several valuable directions warrant exploration in future research. The system's tool layer could be extended to include hardware-in-the-loop control, enabling agents to interact directly with physical instruments and microcontroller boards. Fine-tuning a DeepSeek model on a curated domain-specific corpus could further mitigate technical hallucinations. Integrating multimodal inputs, such as oscilloscope screenshots and circuit photographs, would enrich the system's perceptual capabilities. Furthermore, multi-institutional, longitudinal randomized controlled trials could be conducted to rigorously validate the model's impact on learning outcomes. Rooted in authoritative course materials and constrained by human oversight, this reasoning-and-acting agent paradigm presents a promising and transferable framework for advancing hands-on engineering education.

Funding

This work was supported by the 2025 Anhui Provincial Quality Engineering Project for Higher Education (Grant No: 2025gspaikc077), the Key Natural Science Research Project of the Anhui Provincial Department of Education (Grant No: 2025AHGXZK31227), and the Huaibei Vocational and Technical College Quality Engineering Project (Grant No: 2024jyxm-02).

Acknowledgements

The author would like to show sincere thanks to those technicians who have contributed to this research.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] Heradio, R., De La Torre, L., Galan, D., Cabrerizo, F. J., Herrera-Viedma, E., Dormido, S. (2016) Virtual and remote labs in education: a bibliometric analysis. *Computers & Education*, 98, 14-38.

[2] Myers, D., Mohawesh, R., Chellaboina, V. I., Sathvik, A. L., Venkatesh, P., Ho, Y. H., Jararweh, Y. (2024) Foundation and large language models: fundamentals, challenges, opportunities, and social impacts. *Cluster Computing*, 27(1), 1-26.

[3] Xiong, L., Wang, H., Chen, X., Sheng, L., Xiong, Y., Liu, J., Tang, Y. (2025) DeepSeek: paradigm shifts and technical evolution in large AI models. *IEEE/CAA Journal of Automatica Sinica*, 12(5), 841-858.

[4] Zhou, K., Zhang, B., Wang, J., Chen, Z., Zhao, W. X., Sha, J., Wen, J. R. (2024) Jiuzhang 3.0: Efficiently improving mathematical reasoning by training small data synthesis models. *Advances in Neural Information Processing Systems*, 37, 1854-1889.

[5] Mahmoud, A. A. S., Shishah, W., Mistry, N. R. (2025) ReACT_OCRS: An AI-driven anonymous online reporting system using synergized reasoning and acting in language models. *IEEE Access*, 13, 92800-92815.

[6] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Scialom, T. (2023) Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 68539-68551.

[7] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Wen, J. (2024) A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), 186345.

[8] Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Kasneci, G. (2023) ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274.

[9] Nikolic, S., Daniel, S., Haque, R., Belkina, M., Hassan, G. M., Grundy, S., Sandison, C. (2023) ChatGPT versus engineering education assessment: a multidisciplinary and multi-institutional benchmarking and analysis of this generative artificial intelligence tool to investigate assessment integrity. *European Journal of Engineering Education*, 48(4), 559-614.

[10] Liu, S., Guo, X., Hu, X., Zhao, X. (2024) Advancing generative intelligent tutoring systems with GPT-4: design, evaluation, and a modular framework for future learning platforms.

Electronics, 13(24), 4876.

[11] Rojas-Sánchez, M. A., Palos-Sánchez, P. R., Folgado-Fernández, J. A. (2023) Systematic literature review and bibliometric analysis on virtual reality and education. *Education and Information Technologies*, 28(1), 155-192.

[12] Ilieva, G., Yankova, T., Ruseva, M., Kabaivanov, S. (2025) A framework for generative AI-driven assessment in higher education. *Information*, 16(6), 472.

[13] Hutson, J. (2025) From prohibition to preparation: Reframing academic integrity in the age of AI. *MRS Journal of Arts, Humanities and Literature*, 2(11), 54-65.