A Test and Training Platform Utilizing a Raspberry Pi 5: AI-Retrieval, Tools and a Compact LLM

Siheng Ye, Xiaotong Li, Chang Qi, Linfei Yin, Yaokun Zhang and Ian Pratt

University of Nottingham Ningbo China

September 2026 camera-ready author manuscript

The following ten pages preserve the author-created manuscript associated with the 19 September 2026 camera-ready submission. This notice adds version and sharing information; it does not revise the manuscript.

The manuscript was accepted for presentation at IHCI 2026. It is not the publisher's Version of Record and does not include subsequent publisher production or corrections. Proceedings publication, a publisher DOI and indexing have not been verified as of 4 November 2026.

The manuscript title follows the submitted source. The conference programme uses an earlier title for the same Paper 106.

Manuscript terms

Shared on the author's personally maintained website under the proceedings author-manuscript route. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use. No Creative Commons or other open licence is applied to this manuscript.

Springer Nature Accepted Manuscript terms

Project page and current research record

A Test and Training Platform Utilizing a Raspberry Pi 5: AI-Retrieval, Tools and a Compact LLM

Siheng Ye, Xiaotong Li, Chang Qi, Linfei Yin, Yaokun Zhang, and Ian Pratt

University of Nottingham Ningbo China, Ningbo, China
{biysy32,ssyxl28,ssycq1,ssyly13,ssyyz62,ian.pratt}@nottingham.edu.cn

Abstract. Generic language-model assistants do not naturally align with a particular course. We present a Raspberry Pi 5-targeted platform combining a compact Qwen3-1.7B model, course-grounded retrieval, deterministic calculation, and a Web portal that exposes answers, evidence, and tool execution. It is configurable for individual course requirements; we evaluate system quality and edge deployment, not learning outcomes. GPU-accelerated evaluation compares seven configurations over 96 course questions: closed-book generation, retrieval-augmented generation (RAG), Calculator execution, and two supervised fine-tuning (SFT) recipes. A separate 24-question set compares Base and Base+RAG approaches. Base + Tool-Augmented RAG achieved the highest mean rubric score of 0.777 on a 0–1 scale. Retrieval improved mean source-sensitive quality; deterministic Calculator execution gave the clearest gain on 24 preclassified supported items. Neither SFT recipe demonstrated improvement over its matched Base configuration. A paired 96-question Base comparison did not detect a directional GPU–Pi score difference; the Pi ran the same quantized artifact more slowly.

Keywords: Human–AI interaction · Edge AI · Engineering education · Retrieval-augmented generation · Tool-augmented language models · Fine-tuning

1 Introduction

Large Language Models offer a tremendous opportunity to educate and train humans. Most are very generic and not particularly well-suited to the needs of the individual nor to the esoteric nature of many disciplines. In this paper we investigate teaching and knowledge assessment in the context of the Electronic and Electrical Engineering discipline. Our goals are to tailor the LLM to the detailed elements within a discipline, as well as assessing the topic weaknesses of a particular individual; learning and adapting to their needs with the aim of improving their subject knowledge. This tailored approach drives us towards local AI and edge deployment.

The Raspberry Pi 5 target motivates a compact 1.7-billion-parameter large language model (LLM) and four-bit deployment within its processor and memory

budgets. Compact models remain imperfect [1]: RAG supplies evidence [2], a Calculator executes supported tasks, and SFT changes weights. Students checking answers must distinguish evidence availability, execution, and correct synthesis.

We make three contributions: (1) a Web prototype that separates answers, sources, and calculation traces; (2) a controlled comparison of retrieval, preclassified Calculator execution, two SFT recipes, and a separate question-and-subject router audit; and (3) paired closed-book Base quality and model-request measurements using the same quantized artifact on Pi and GPU. "Web" means the application, not Internet search.

We ask three research questions. **RQ1**: How do retrieval and preclassified Calculator execution affect offline quality, and can the frozen router identify eligible requests within this benchmark? **RQ2**: Does either SFT recipe improve its matched Base? **RQ3**: Do paired closed-book scores show a directional Pi–GPU difference, and how does model-request performance compare?

2 Related Work

Low-rank adaptation (LoRA) trains small adapters while freezing pretrained weights [3]; efficient-model work also studies architecture and size [4]. RAG supplies external information [2], while tool use supplies computation [5]. Course-grounded tutors [6] and retrieval assistants [7] motivate our fixed-model comparison of retrieval, calculation, SFT, and device behavior. Citation support [8], capability communication [9], and overreliance risks [10] motivate inspectable evidence and tool traces.

3 System Architecture

Figure 1 summarizes the user-facing design. Ask routes supported numerical tasks to a deterministic Calculator and other questions to subject-scoped retrieval and model generation; Quiz is a separate path. The interface exposes answers, sources, and tool traces. Table 1 separates model weights from supporting components. To suit the Pi's resource budget, all conditions use Qwen3-1.7B with Q4_K_M four-bit quantization in GGUF, a single-file weights-and-metadata format. Closed-book conditions omit course context. Offline RAG searches 2,902 excerpts from 82 subject-isolated course-note files. BM25 (saturation 1.2; length normalization 0.75) retrieves 20 candidates; adjacent-word and source-locator bonuses rerank them to five with deterministic ties. The Web instead uses embedding retrieval. The offline Calculator supports 17 numerical families. A frozen question-and-subject router invokes a family only after identification, typed parsing, and execution; otherwise it returns a fallback decision intended for RAG. Separately, 24 items are preclassified to isolate execution benefit. Tools conditions share retrieval and tool inputs, differing only in weights.

Traceability exposes the source and computation path associated with an answer (Fig. 2). Ask separates generated answers, sources, and calculation steps;

Table 1. Seven evaluated systems. "Tools" denotes Tool-Augmented RAG: Calculator execution for supported items, otherwise the shared RAG path.

System	Context path	RAG	Calc.
Base	Closed-book	No	No
Base+RAG	Subject BM25, reranked Top-5	Yes	No
Base+Tools	Tool-Augmented RAG	Yes	Yes
Initial SFT (801 records)	Closed-book	No	No
Initial SFT+Tools	Tool-Augmented RAG	Yes	Yes
Expanded SFT (2,400 records)	Closed-book	No	No
Expanded SFT+Tools	Tool-Augmented RAG	Yes	Yes

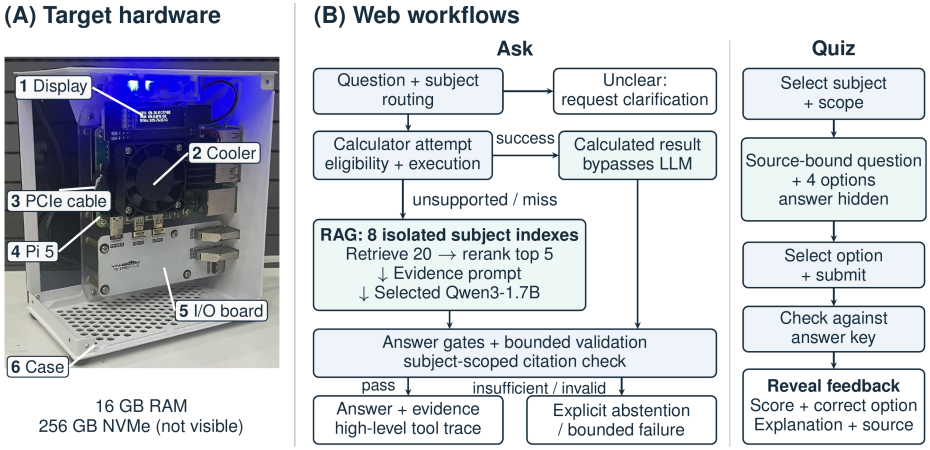


Fig. 1. Target hardware (A) and Web workflows (B). Ask routes requests to deterministic calculation or retrieval-grounded generation; Quiz hides its answer key until submission and then reveals feedback.

Quiz reveals feedback and sources after submission. A displayed source or selected model name does not by itself verify the answer or its producer.

4 Evaluation

4.1 Quality Sets and Controls

The main benchmark contains 12 public questions from each of eight electrical and electronic engineering subjects: Design and Implementation of Engineering Software, Electrical Energy Conditioning and Control, Engineering Mathematics, Electronic Processing and Communications, Information and Systems, Modelling: Methods and Tools, Power and Energy, and Introduction to Software

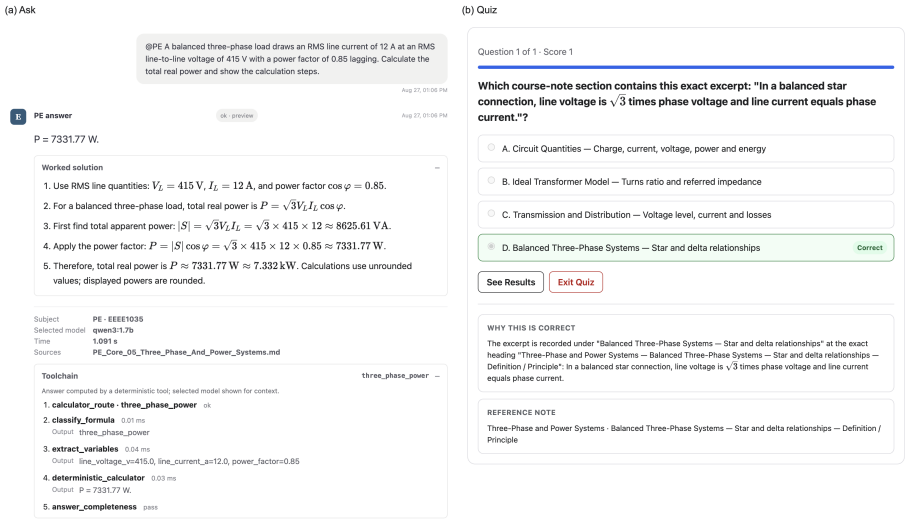


Fig. 2. Web examples. (a) Calculator output with worked steps and a tool trace. (b) Post-submission Quiz feedback with explanation and source. Cropping changes presentation only; questions, answers, and scores are unchanged.

Engineering and Programming. Its 96 items comprise 28 concept, 28 retrieval-integration, 24 Calculator-supported, eight Calculator-unsupported, and eight insufficient-evidence questions: seven systems yield 672 paired responses. A separate, unpooled 24-question retrieval-sensitive set compares only Base and Base+RAG. It was authored from course notes without system answers, scores, or retrieval outputs and checked for exact question duplicates against the main set. Protected test content was not accessed.

Both SFT variants share a pinned parent checkpoint, PyTorch 2.8.0, Transformers 4.56.2, and PEFT 0.17.1. LoRA updates query/value projections (rank 8, $\alpha = 16$, no dropout). Initial SFT uses 801 records, three epochs, AdamW at 2×10^{-4}, and four records/update; Expanded SFT uses 2,400 of 4,000 candidates, one epoch, 10^{-4}, and batch 4 with four-step accumulation. Both use 2,048-token sequences; this compares recipes, not data quantity alone.

Post hoc normalized lexical screens found no exact training-question match. No eligible pair among 95/96 main and 24/24 retrieval questions reached Jaccard 0.8; 734/744 recoverable Initial-SFT inputs met the eight-token screen.

Question text, corpus, model files, and rubric were fixed. Generation used llama-cpp-python, the Qwen3 non-thinking template, temperature 0 (greedy), 256 output tokens, 2,048 context tokens, seed 20260826, and end-of-message/end-of-text stop markers. The seven-system comparison used eight threads and all-layer GPU offload; its scores alone do not measure Pi quality. The matched Base analysis below provides separate device evidence. Closed-book prompts omit evidence; matched RAG conditions share questions, passages, and prompts.

Objective items used exact-match or numeric-tolerance checks. ChatGPT, an LLM judge rather than a human, provided the initial semantic scores (including insufficient-evidence cases) under opaque condition labels. The rubric measures central correctness (*core*, 0–4), required-content coverage (0–3), reasoning/evidence consistency (0–2), and relevance (0–1). The sum is capped at 4 for core ≤ 1 or 6 for core 2, then divided by 10; higher is better. Semantic "fully correct" requires core 4, coverage 3, and consistency 2, not the relevance point; objective fully correct requires binary score 1. Mean normalized score is primary; fully-correct counts are secondary.

The main benchmark used 280 first-pass semantic judgments, 86 second reviews, and 59 adjudications; the retrieval set used 48, 13, and 7. Mid-range scores, uncertainty, or caps triggered review, and component disagreement triggered adjudication. Each of 52 format fallbacks received two content reviews; one was adjudicated. The first author checked every GPU-study score against the rubric, which is not independent blinded validation. Selected reviewer pairs agreed on all components for 27/86 main and 6/13 retrieval cases; this selected subset does not estimate overall reliability.

Paired question-level percentile-bootstrap confidence intervals (CIs) describe mean-score differences. Exact two-sided McNemar tests instead compare binary fully-correct outcomes. The CIs are unadjusted; Holm correction applies only to McNemar p values for four primary contrasts: retrieval, Calculator, and Expanded SFT with and without tools. Initial SFT contrasts are secondary and unadjusted. The 64-item model-comparison subset comprises 28 concept, 28 retrieval-integration, and eight unsupported-Calculator items; it excludes the 24 supported and eight insufficient-evidence items. Subject comparisons are descriptive, not subject-level significance tests.

A post hoc audit replayed the frozen 17-family router over all 96 main questions using only the question and subject; gold fields were withheld until routing completed. It checked selection, family match, typed parsing, and execution, with no model or network calls.

4.2 Pi Quality and Device Performance

A separate CPU-only Pi run evaluated Base on all 96 main-benchmark questions using the same GGUF, closed-book prompts, and decoding controls. It used Python 3.13.5, llama-cpp-python 0.3.26, eight threads, and zero GPU layers, without retrieval, tools, or Web processing. Each question was attempted once. Scoring retained the original rubric: 46 responses used the original automatic scorer, while 48 were newly AI-scored with the same rubric under the original second-review and adjudication rules. Two grades were carried forward only when the question was identical, the final answer was byte-identical, and the rubric was unchanged; these were not new reviews. Thus, Pi quality covers closed-book Base; retrieval and Calculator effects come from the GPU study. The 96 GPU and Pi Base records were paired by item, with matching prompt and GGUF hashes. We report a 100,000-resample paired bootstrap CI (seed 20260914) and

Table 2. Main-benchmark quality. Mean score is normalized to 0–1. The fully-correct column reports the number of answers meeting the criterion in Sect. 4.1.

System	Mean score	Fully correct
Base (closed-book)	0.503	28/96
Base+RAG	0.555	34/96
Base+Tools	**0.777**	**54/96**
Initial SFT (closed-book)	0.438	25/96
Initial SFT+Tools	0.750	52/96
Expanded SFT (closed-book)	0.408	21/96
Expanded SFT+Tools	0.584	39/96

exact McNemar test; this post hoc analysis is outside Table 3's multiplicity family.

For performance, the Pi and RTX 6000D ran the checksum-matched Base Q4_K_M GGUF on eight questions, one per subject. The Pi used four Cortex-A76 cores, 16 GB memory, 256 GB NVMe storage (no SD card), active cooling, and a 5.1 V/5 A supply. One cold and two warm runs per question/device yielded 48 records. Pre-rendered prompts and zero routing time exclude retrieval/tools; cold means a fresh process, not a flushed OS cache, and warm runs may use caches. Time to first token (TTFT) is the initial wait; resident set size (RSS) is process memory. This study is separate from the 96-question quality run.

5 Results

5.1 Overall Quality and Matched Effects

Base+Tools has the highest observed main-benchmark score, followed by Initial SFT+Tools (Table 2). Neither SFT system exceeds its matched Base.

Table 3 separates retrieval, Calculator, and adaptation effects. On retrieval-sensitive items, Base+RAG improved the mean by 0.454, although the Holm-adjusted McNemar p value exceeded 0.05. Calculator-supported items gained +1.000 uniformly. Neither SFT recipe improved its matched Base; Expanded SFT decreased most within the full stack.

5.2 Failure Modes

Calculator-supported item counts vary by subject, so between-subject scores do not measure subject competence. Table 4 separates retrieval from synthesis: shared retrieval had 0/28 expected-source file misses, although file hits do not guarantee evidence-span hits. Misses and synthesis failures are mutually exclusive; central-correctness failure means core 0–2, and retrieved-but-incomplete means core 3–4 with coverage below 3.

The post hoc, in-set router audit selected all 24 supported items and declined the other 72; family match, typed parsing, and execution succeeded for all 24 selections. This checks the specified public questions, not unseen requests.

Table 3. Paired candidate-minus-comparator contrasts: n items and mean change Δ. Mean-score CIs are unadjusted. McNemar p values concern fully-correct outcomes; primary p values are Holm-adjusted, † secondary values unadjusted. All SFT contrasts use the 64-item model subset defined in Sect. 4.1. The Calculator contrast estimates preclassified execution benefit; routing is audited separately over all 96 items.

Contrast (subset)	n	Δ mean	95% paired CI	Fully-correct McNemar p
Base+RAG − Base (retrieval set)	24	+0.454	[0.350, 0.558]	Holm 0.0625
Base+Tools − Base+RAG (Calc.)	24	+1.000	[1.000, 1.000]	Holm $< 10^{-6}$
Initial SFT − Base	64	−0.112	[−0.234, 0.009]	$0.4807^{\dagger}$
Initial SFT+Tools − Base+Tools	64	−0.044	[−0.137, 0.050]	$1.000^{\dagger}$
Expanded SFT − Base	64	−0.134	[−0.244, −0.022]	Holm 0.1796
Expanded SFT+Tools − Base+Tools	64	−0.252	[−0.359, −0.147]	Holm 0.0222

5.3 GPU–Pi Performance

Table 5 reports medians. The GPU is substantially faster; the Pi nevertheless executes the same Base Q4_K_M artifact locally. Only 2/24 paired performance-run outputs were byte-identical; this differs from the 96-item quality pairing below. GPU video memory was not sampled; RSS is host memory on both devices.

Paired GPU–Pi quality. All 96 Pi requests completed. GPU and Pi means were 0.503 and 0.498, with 28/96 and 29/96 fully correct. The Pi-minus-GPU difference was −0.005 (95% paired CI [−0.041, 0.032]); Pi was better/tied/worse on 14/65/17 items. Fully-correct discordance was three Pi-only versus two GPU-only (exact McNemar $p = 1.000$). The analysis did not detect a directional difference, but did not establish equivalence: each device generated once per item and only 5/96 answers were byte-identical.

6 Discussion

What retrieval and tools contribute. The 24-item retrieval-sensitive set gave a Base+RAG mean-score gain of 0.454; on the main benchmark's 28 retrieval-integration items, Base+RAG scored 0.686 versus Base's 0.650, a descriptive difference of 0.036. These sets have different, deliberately chosen question mixes and are not interchangeable estimates. Expected-source files appeared in Top-5 for all 28 main-set items, yet Base+RAG had nine central-correctness failures (core 0–2) and 15 retrieved-but-incomplete answers (core 3–4, coverage below 3). A file hit neither locates the exact supporting passage nor verifies the synthesis. The uniform Calculator gain in Table 3 concerns 24 preclassified supported items;

Table 4. Diagnostic counts; dashes mark out-of-scope conditions. The separate router audit declined all 8/8 unsupported numerical items; the last two rows concern fallback-answer quality, not Calculator errors.

	Base			Initial SFT		Expanded SFT	
Diagnostic	Closed	RAG	Tools	Closed	Tools	Closed	Tools
Central-correctness failure	–	9/28	9/28	–	11/28	–	18/28
Retrieved but incomplete	–	15/28	16/28	–	13/28	–	9/28
Unsupported fallback: not fully correct	–	–	5/8	–	7/8	–	8/8
Insufficient evidence: not fully correct	3/8	3/8	4/8	2/8	5/8	4/8	8/8

Table 5. Median model-request measurements. Lower is better for time and memory; higher is better for generation throughput.

Measurement	RTX 6000D	Raspberry Pi 5	Unit
Model load time	0.430	3.447	s
Time to first token, cold	0.039	16.676	s
Time to first token, warm	0.005	0.118	s
Total request time, cold	0.344	28.239	s
Total request time, warm	0.311	9.607	s
Generation throughput, cold	253.66	3.85	tokens/s
Generation throughput, warm	317.81	8.56	tokens/s
Process resident set size	1.829	2.364	GiB

the separate post hoc router audit found correct decisions on the 96 public questions, but does not measure unseen-family or live-traffic routing. Thus the system-level gain cannot be assigned to model weights.

Interpreting the SFT results. Neither fine-tuning recipe improved its matched Base configuration; the Expanded SFT decline was most pronounced with the same tool-augmented context. Its authored training pool passed structural checks, and same-distribution held-out training diagnostics improved, but those diagnostics did not predict independent answer quality. In an earlier 64-question development comparison, the Expanded model's mean score was 0.369 versus Base's 0.615 in BF16, before the four-bit device comparison. This makes quantization alone an insufficient explanation for the negative result. The training examples were not exhaustively independently audited for semantic quality; answer and evidence overlap within the authored pool also limits what its held-out loss can establish. Data composition, optimization schedule, and prompt distribution could each matter, but these recipes change more than one factor, so we cannot identify a single cause or conclude that the questions were simply poor.

The shared retrieved inputs and the failures after source-file retrieval point to answer synthesis and completeness as useful targets for future controlled tests.

Interface and device implications. Ask displays a generated answer, cited course material, or a deterministic Calculator trace according to the route; Quiz reveals its answer and explanation after submission (Fig. 2). These affordances help users inspect the apparent source of a response, not its correctness. The offline BM25 comparison does not validate the deployed Web embedding-retrieval path. The paired Base GPU–Pi analysis did not detect a directional score difference, but only 5/96 answers were byte-identical and there was one generation per item/device; neither equivalence nor repeatability follows. No user study measured learning, trust, or usability.

7 Scope, Limitations, and Ethics

This study provides paired seven-system evidence, an all-item router audit, direct Pi execution, and controlled device measurements. The public benchmark is not independently held out; subject task mixes differ, and file hits do not verify evidence spans. The preclassified Calculator contrast and post hoc public 17-family audit do not establish unseen-family or deployed-traffic generalization; SFT conclusions concern two recipes.

Semantic scoring used a blinded LLM judge, triggered review/adjudication, and a first-author rubric check rather than independent non-author validation. The single-generation device comparison specifies no equivalence margin and does not evaluate Pi-side RAG, Calculator, or end-to-end Web quality. Learning and usability were not evaluated. Course materials require permission; the system supports rather than replaces professional judgment.

Future work. Future work will address external evaluation and usability, including closed-loop learning in which the system learns the weaknesses of a test subject and adjusts its questions to address those weaknesses.

System performance has been internally evaluated. External assessment, including demonstration of the system adapting to the learned weaknesses of a particular student, is left as an extended goal of this body of work.

8 Conclusion

The Pi target motivates our compact, four-bit model. Base+Tools was the highest-scoring offline GPU configuration; a separate audit confirmed routing within the public benchmark. Neither SFT recipe improved its matched Base. A paired Base comparison did not detect a directional GPU–Pi score difference, while the Pi was slower; neither result establishes cross-device equivalence or Pi-side Web quality.

References

1. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv:2505.09388 (2025). https://doi.org/10.48550/arXiv.2505.09388
2. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems 33 (2020), pp. 9459–9474. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html
3. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=nZeVKeeFYf9
4. Liu, Z., et al.: MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 32431–32454. PMLR (2024). https://proceedings.mlr.press/v235/liu24ce.html
5. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. In: Advances in Neural Information Processing Systems 36 (2023), pp. 68539–68551. https://doi.org/10.52202/075280-2997
6. Ma, I., Krone-Martins, A., Videira Lopes, C.: Integrating AI tutors in a programming course. In: Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V.1, pp. 130–136 (2024). https://doi.org/10.1145/3649165.3690094
7. Marciniak, J., Kubis, M., Gulczyński, M., Szpilkowski, A., Wieczarek, A., Szczepański, M.: Improving AI assistants embedded in short e-learning courses with limited textual content. In: Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 794–804. Association for Computational Linguistics (2025). https://doi.org/10.18653/v1/2025.bea-1.57
8. Gao, T., Yen, H., Yu, J., Chen, D.: Enabling large language models to generate text with citations. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 6465–6488. Association for Computational Linguistics (2023). https://doi.org/10.18653/v1/2023.emnlp-main.398
9. Amershi, S., et al.: Guidelines for human-AI interaction. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2019). https://doi.org/10.1145/3290605.3300233
10. Kasneci, E., et al.: ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences 103, 102274 (2023). https://doi.org/10.1016/j.lindif.2023.102274