Communication

From Automation to Certification: Benchmarking AI Chatbots in Software Testing

Niklas Retzlaff *

Posted Date: 10 January 2025

doi: 10.20944/preprints202501.0770.v1

Keywords: Artificial Intelligence; Chatbots; Large Language Models; Software Testing; Certification Exams; GPT-4o; Gemini; A4Q-SDET; ISTQB

Communication

From Automation to Certification: Benchmarking AI Chatbots in Software Testing

Niklas Retzlaff

Triagon Academy Malta (Doctoral Student), Villa Violette, Triq San Bernard Marsa MRS1331 Malta; niklas.retzlaff.dba@edu.triagon-academy.com

Abstract: Artificial intelligence (AI) chatbots, powered by large language models (LLMs), are reshaping software testing by automating critical tasks and enhancing productivity. This study evaluates the performance of two state-of-the-art LLMs, GPT-4o and Gemini 2.0 Flash Experimental, on practice exams for four industry-recognized software testing certifications: A4Q-SDET, ISTQB Certified Tester Foundation Level (CTFL), Advanced Level Test Manager (CTAL-TM), and Expert Level Test Manager (CTEL-TM). Both models demonstrated substantial competency, achieving passing scores across all exams. GPT-4o excelled in foundational and advanced managerial tasks, while Gemini outperformed in technical and practical test scenarios. An analysis of their performance across cognitive levels (K1–K4) reveals complementary strengths, with GPT-4o showing superior analytical capabilities (K4) and Gemini maintaining consistent performance across all levels. These findings highlight the potential of LLMs as tools for bridging knowledge gaps and enhancing software testing processes. Future research should explore real-world testing applications and the integration of LLMs into software testing workflows.

Keywords: artificial intelligence; chatbots; large language models; software testing; certification exams; GPT-4o; Gemini; A4Q-SDET; ISTQB

1. Introduction

Artificial Intelligence (AI) chatbots, powered by large language models (LLMs), are transforming numerous fields, including software testing. These advanced systems have demonstrated significant potential in automating critical tasks, such as generating and optimizing test cases, analyzing graphical user interfaces (GUIs), and predicting software defects [1–4]. Over one hundred studies have explored the application of AI in software testing, showcasing its ability to enhance productivity and accuracy in the software development lifecycle [2].

AI techniques such as machine learning and deep learning leverage large datasets to improve test case design and optimization [5,6]. Advances in computer vision have enabled automated GUI testing [7], and AI-driven tools are increasingly proficient at generating unit tests for diverse programming languages and frameworks [8]. Furthermore, AI can predict software defects by analyzing extensive historical data, providing valuable insights for preemptive error management [9–11]. These developments have shown AI's potential to perform tasks previously carried out manually with greater accuracy and efficiency [12,13].

While the focus of AI applications in software testing has primarily been on task automation, there is a growing interest in assessing whether these models can also demonstrate a comprehensive understanding of software testing principles, as measured by industry-recognized certification exams. This question is particularly relevant for software developers, who often assume testing responsibilities without formal training [14]. To ensure the quality of software products, sound knowledge of software testing is essential. However, studies indicate that many developers prioritize coding over testing and would benefit from structured training in effective testing practices [14]. This lack of formal training leads to a knowledge gap that can affect the quality and efficiency of testing. Certifications, such as

 doi:10.20944/preprints202501.0770.v1

the ISTQB Foundation Level or those offered by the Alliance for Qualification (A4Q), evaluate the understanding of key concepts and best practices and can serve as a benchmark for assessing the knowledge of software testers.

This study, therefore, explores the potential of AI chatbots not only to assist in practical testing tasks but also to demonstrate a solid understanding of software testing concepts as assessed by certification exams. Specifically, it evaluates the performance of two state-of-the-art LLMs, GPT-4o and Gemini 2.0 Flash Experimental, on practice exams for four industry-recognized software testing certifications, which vary in complexity and focus.

By analyzing the accuracy, reasoning capabilities, and consistency of these models in handling complex multiple-choice questions, this research aims to provide valuable insights into their applicability as tools for bridging knowledge gaps in software testing. This study contributes to understanding the capabilities of AI chatbots and determining whether they can achieve certification-level performance and serve as reliable resources for democratizing expertise in this field, thereby fostering a more efficient and effective software testing ecosystem

2. Materials and Methods

To evaluate the performance of two language models on software testing tasks and knowledge, we selected four practice exams published by the German Testing Board (GTB) [15].

Three of the four exams were developed and published by the International Software Testing Qualifications Board (ISTQB®) [16] while the fourth exam was developed and published by the Alliance for Qualification (A4Q) [17].

Four specific practice exams were chosen, covering a range of software testing competencies:

- **Software Development Engineer in Test (A4Q-SDET):** This exam consists of 40 multiple-choice questions, each with exactly one correct answer, and each question worth one point. The exam focuses on evaluating the technical and theoretical competencies of software development engineers in testing, emphasizing the integration of effective test strategies and techniques into the software development lifecycle.
- **ISTQB® Certified Tester Foundation Level (CTFL):** This exam includes 40 multiple-choice questions spanning cognitive levels K1 through K3. The questions evaluate foundational knowledge of software testing principles, processes, and tools, emphasizing understanding core concepts and applying them in basic scenarios.
- **ISTQB® Certified Tester Advanced Level Test Manager (CTAL-TM):** This exam contains 50 multiple-choice questions. Each question has either one or two correct answers. Questions are weighted between one and three points. The exam assesses advanced knowledge of test management, including the formulation of test strategies, leadership of testing teams, and the coordination of test-related communication with stakeholders.
- **ISTQB® Certified Tester Expert Level Test Manager (CTEL-TM):** This exam includes two components: 20 multiple-choice questions addressing cognitive levels K2 through K4, assessing advanced analytical and managerial skills. An essay question component (K5 through K6) designed to evaluate the ability to analyze complex scenarios and create sophisticated solutions. The essay portion was excluded from this study.

The selection of these exams aimed to cover a broad range of software testing competencies, from foundational knowledge (CTFL) to advanced managerial (CTAL-TM) and expert-level skills (CTEL-TM). This selection provides a comprehensive framework for evaluating the models' performance across diverse testing domains.

The ISTQB® categorizes exam questions into six cognitive levels [18]:

- K1 – Remember: Recognizing or recalling a term or concept.
- K2 – Understand: Explaining or interpreting a statement related to the question topic.
- K3 – Apply: Applying a concept or technique in a given context.

 doi:10.20944/preprints202501.0770.v1

- K4 – Analyze: Breaking down information related to a procedure or technique and distinguishing between facts and inferences.
- K5 – Evaluate: Making judgments based on criteria, detecting inconsistencies or fallacies, and assessing the effectiveness of a process or product.
- K6 – Create: Combining elements to form a coherent or functional whole, inventing a product, or devising a procedure.

In this study, only the multiple-choice components (K1 – K4) of the exams were evaluated. The essay questions in the CTEL-TM exam (K5 – K6) were excluded to maintain a consistent evaluation method across all exams.

Two large language models were evaluated: GPT-4o [**?**] and Gemini 2.0 Flash Experimental [**?**]. GPT-4o is designed for tasks requiring complex reasoning. It is trained on extensive textual data, enabling it to generate coherent responses and justifications across various domains. Its selection was based on its established ability to handle complex queries and produce logically consistent arguments. Gemini 2.0 Flash Experimental focuses on rapid inference and enhanced reasoning capabilities. It manages multifaceted queries, including multi-modal inputs, while maintaining efficiency. It is included to enable comparison with a more established model under test-specific conditions.

The models received the original practice exam questions in textual form, including images where applicable. Each question prompted the models to explain or justify their chosen answer. The first response from each model was recorded without any follow-up questions or clarifications. No specialized system instructions were provided to either model, and no default configuration settings were modified.

In total, 150 multiple-choice questions were administered, distributed as follows:

- 40 questions from A4Q-SDET,
- 40 questions from CTFL,
- 50 questions from CTAL-TM, and
- 20 questions from CTEL-TM (excluding essay questions).

By incorporating exams with varying cognitive demands and topical focuses, this study aims to provide a nuanced understanding of each model's capacity to address both technical and managerial aspects of software testing.

3. Results

Table 1 summarizes the performance of ChatGPT and Gemini on the two selected practice exams. Both models attained passing scores on each exam, although their relative performance varied by exam type.

Table 1. Comparison of ChatGPT and Gemini on software testing practice exams.

Exam	Model	Correct / Total	Percentage	Outcome
A4Q-SDET	GPT-4o	33 / 40	82.50%	Passed
A4Q-SDET	Gemini 2.0 Flash Experimental	37 / 40	92.50%	Passed
CTFL	GPT-4o	38 / 40	95.00%	Passed
CTFL	Gemini 2.0 Flash Experimental	34 / 40	85.00%	Passed
CTAL-TM	GPT-4o	76 / 88	86,36%	Passed
CTAL-TM	Gemini 2.0 Flash Experimental	70 / 88	79.55%	Passed
CTEL-TM	GPT-4o	16 / 20	80.00%	Passed
CTEL-TM	Gemini 2.0 Flash Experimental	16 / 20	80.00%	Passed

Gemini scored 37 out of 40 (92.50%), surpassing ChatGPT, which achieved 33 out of 40 (82.50%). Both scores exceeded the passing threshold, indicating that each model demonstrated substantial competency in the technical and theoretical testing concepts evaluated by the A4Q-SDET exam.

 doi:10.20944/preprints202501.0770.v1

On the more advanced test-management oriented CTAL-TM exam, ChatGPT achieved slightly higher accuracy, with 74 out of 88 (84.09%), compared to Gemini's 70 out of 88 (79.55%). Both models again met the passing criteria, reflecting a satisfactory command of test-management knowledge and practices.

The results indicate that Gemini 2.0 Flash Experimental outperformed GPT-4o in the A4Q-SDET exam, scoring 37 out of 40 (92.50%) compared to GPT-4o's 33 out of 40 (82.50%). Both scores exceeded the passing threshold, demonstrating substantial competency in technical and theoretical testing concepts evaluated by the A4Q-SDET exam.

In the foundational CTFL exam, GPT-4o achieved a higher score, with 38 out of 40 (95.00%), while Gemini scored 34 out of 40 (85.00%). Both models passed the exam, reflecting a strong grasp of foundational software testing principles and processes.

On the advanced CTAL-TM exam, which focuses on test management, GPT-4o again demonstrated higher accuracy, scoring 76 out of 88 (86.36%), compared to Gemini's 70 out of 88 (79.55%). Both models met the passing criteria, indicating proficiency in advanced test management knowledge and practices.

For the CTEL-TM exam, which evaluates expert-level testing skills, both GPT-4o and Gemini scored equally, with 16 out of 20 (80.00%), meeting the passing requirement. This parity suggests that both models are capable of handling complex test scenarios at an expert level.

Overall, the results suggest that while both models possess adequate capabilities to handle a range of software testing scenarios, each model exhibits relative strengths depending on the specific focus and complexity of the exam.

To further explore the performance of the two models across different cognitive levels (K-levels), their relative success rates were compared and visualized in Figure 1.

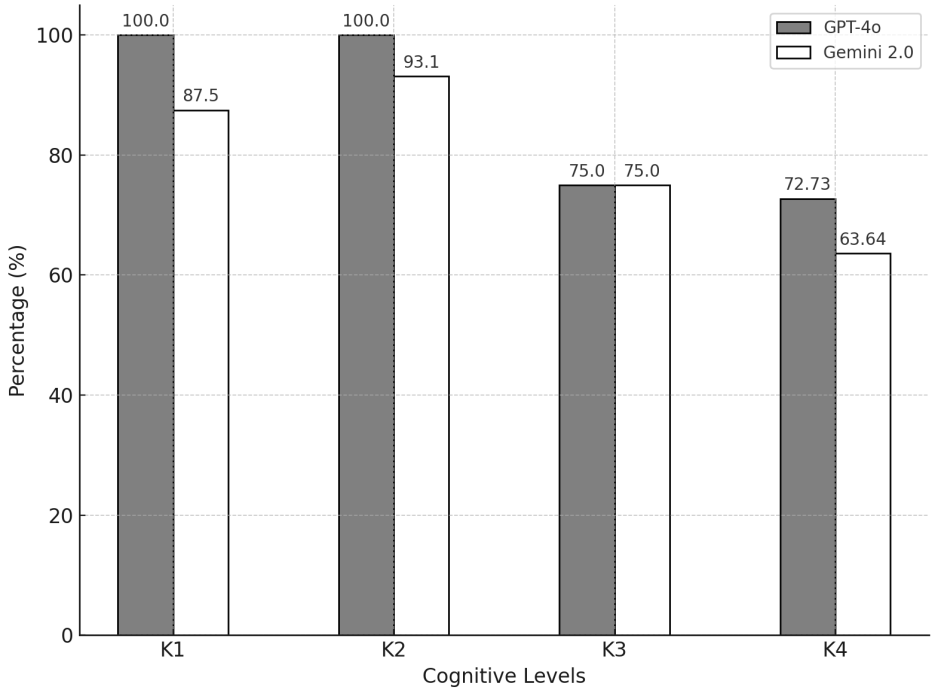

Figure 1. Comparison of GPT-4o and Gemini 2.0 Flash Experimental on K-levels in percentage.

Figure 1 reveals that GPT-4o achieved 100% accuracy for K1 and K2 levels, demonstrating its strength in foundational and comprehension-based questions. In contrast, Gemini achieved slightly lower scores for these levels, with 87.50% and 93.10%, respectively. For K3 (application-level questions), both models performed equally well with 75.00% accuracy. However, a notable difference was observed for K4 (analysis-level questions), where GPT-4o achieved 72.73% accuracy compared to Gemini's 63.64%.

 doi:10.20944/preprints202501.0770.v1

These results highlight GPT-4o's relative strength in handling higher cognitive-level tasks, such as analysis (K4), which require deeper reasoning and contextual understanding. Conversely, while Gemini performed slightly below GPT-4o for most levels, its consistent results suggest robust general capabilities across all cognitive levels. Together, these insights emphasize the complementary strengths of both models and their potential applications in diverse software testing scenarios.

4. Discussion

The findings of this study illustrate the potential of large language models (LLMs), specifically GPT-4o and Gemini 2.0 Flash Experimental, as valuable tools for supporting knowledge acquisition and practical applications in software testing. Both models demonstrated the ability to pass industry-recognized certification practice exams, underscoring their competence in handling complex software testing concepts. However, their relative performance differences reveal nuanced strengths and weaknesses that merit further examination.

The results indicate that both GPT-4o and Gemini possess substantial capabilities across varying cognitive levels and exam complexities. Notably, GPT-4o consistently outperformed Gemini in foundational (CTFL) and advanced managerial (CTAL-TM) exams, suggesting a stronger grasp of general testing principles and complex reasoning tasks. Conversely, Gemini excelled in the A4Q-SDET exam, which emphasizes technical competencies and practical integration of testing within the software development lifecycle. This finding suggests that Gemini may be better suited for contexts requiring rapid problem-solving and technical focus, whereas GPT-4o demonstrates a broader understanding and deeper analytical abilities.

The performance analysis across cognitive levels (K1–K4) further highlights the complementary strengths of these models. GPT-4o's superior performance in higher-order cognitive tasks (K4) reflects its proficiency in analysis and contextual reasoning, critical for tackling complex testing scenarios. In contrast, Gemini's consistent performance across all levels, particularly in K3-level (application) questions, suggests it may be more adaptable for practical, hands-on testing tasks.

The parity observed in K3 performance is especially notable, as this level demands the application of theoretical concepts in practical situations. It suggests that both models possess a foundational robustness in translating knowledge into action, a crucial attribute for software testers in real-world scenarios.

Participants in previous studies noted that LLMs like ChatGPT-4o face challenges due to outdated training data, resulting in inaccuracies when handling newer libraries, frameworks, and methodologies [19]. This limitation is particularly critical in software testing, where tools and standards evolve rapidly. Without frequent updates to training datasets, LLMs risk providing outdated or incorrect guidance, which can mislead users, especially in professional and certification contexts.

An additional risk lies in the reliance on old syllabi or earlier versions of certification guidelines for training LLMs. As certification standards and industry practices evolve, models trained on outdated information may fail to align with current requirements, reducing their effectiveness in preparing users for exams or real-world tasks. Ensuring that LLMs are regularly updated with the latest syllabi and advancements in software testing frameworks is essential to maintaining their relevance and reliability.

There are also quality-related challenges associated with LLM-generated outputs, including hallucination, where the model generates plausible but incorrect information, and security risks when integrating AI-generated content into sensitive workflows [19]. Concerns about data privacy are also prominent, as using LLMs may inadvertently expose sensitive or proprietary information. These challenges necessitate robust governance frameworks and best practices for the responsible use of LLMs in professional environments.

Recent studies suggest that LLMs like ChatGPT-4o perform well on multiple-choice questions [20], but they excel in open-ended question formats [21], where their ability to provide detailed reasoning and explanations becomes more apparent. A limitation of this study was the evaluation of the models using a single response per question, which may have introduced bias. Research by Zhu et al. [21] demonstrates that generating multiple responses for each question significantly improves

 doi:10.20944/preprints202501.0770.v1

accuracy, suggesting that LLMs benefit from iterative response generation when tackling complex scenarios. Incorporating this approach in future studies could provide a more robust assessment of LLM capabilities.

The ability of LLMs to achieve passing scores on professional certification exams demonstrates their potential as tools for democratizing access to software testing expertise. Developers, particularly those without formal training, could benefit from leveraging these models to bridge knowledge gaps, enhance their understanding of testing principles, and improve test case design and execution.

The observed variability in model performance highlights the importance of task-specific model selection. For foundational knowledge transfer or complex test management, GPT-4o may be the preferred choice due to its analytical strengths. Conversely, for technical problem-solving or integration within agile development workflows, Gemini may offer greater utility.

This study is limited by its reliance on practice exams, which may not fully capture the nuanced challenges of real-world testing scenarios. Moreover, the exclusion of essay components in the CTEL-TM exam precludes an evaluation of the models' abilities in generating detailed, context-rich test strategies or managerial solutions. Future research should explore:

- **Dynamic Real-World Testing:** Evaluating model performance in live environments, including GUI testing, defect prediction, and agile workflows.
- **Open-Ended Responses:** Assessing the ability of LLMs to tackle essay-style questions and generate comprehensive test strategies.
- **Multiple Responses:** Investigating the impact of generating multiple answers per question on accuracy and consistency.
- **Data and Framework Updates:** Studying the effect of regularly updating LLM training data to reflect new syllabi, frameworks, and tools.
- **Longitudinal Integration:** Analyzing the long-term impact of LLMs on productivity and quality in software testing teams.

The results of this study underscore the transformative potential of LLMs in software testing. Both GPT-4o and Gemini exhibit distinct strengths, making them valuable assets for different aspects of testing, from foundational knowledge reinforcement to practical application and management. By strategically integrating these models into the software development lifecycle, organizations can enhance testing efficiency and quality, bridge critical knowledge gaps, and foster a more robust testing ecosystem.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data is publicly accessible at https://doi.org/10.5281/zenodo.14618310.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ding, Y. Artificial Intelligence in Software Testing for Emerging Fields: A Review of Technical Applications and Developments. *Applied and Computational Engineering* **2024**, *112*, 161–175. https://doi.org/10.54254/2755-2721/2025.18116.
2. Wang, J.; Huang, Y.; Chen, C.; Liu, Z.; Wang, S.; Wang, Q. Software Testing With Large Language Models: Survey, Landscape, and Vision. *IEEE Transactions on Software Engineering* **2024**, *50*, 911–936. https://doi.org/10.1109/TSE.2024.3368208.
3. Hourani, H.; Hammad, A.; Lafi, M. The Impact of Artificial Intelligence on Software Testing. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2019; pp. 565–570. https://doi.org/10.1109/JEEIT.2019.8717439.
4. Khaliq, Z.; Farooq, S.U.; Khan, D.A. Artificial Intelligence in Software Testing : Impact, Problems, Challenges and Prospect, 2022. arXiv:2201.05371, https://doi.org/10.48550/arXiv.2201.05371.

5. Manojkumar, V.; Mahalakshmi, R. Test Case Optimization Technique for Web Applications. In Proceedings of the 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 2024; pp. 1–7. https://doi.org/10.1109/ICDCECE60827.2024.10548325.
6. Retzlaff, N. AI Integrated ST Modern System for Designing Automated Standard Affirmation System. In Proceedings of the 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2024; pp. 1386–1391. https://doi.org/10.1109/ICACITE60783.2024.10616416.
7. Amalfitano, D.; Coppola, R.; Distante, D.; Ricca, F. AI in GUI-Based Software Testing: Insights from a Survey with Industrial Practitioners. In *Quality of Information and Communications Technology*; Bertolino, A.; Pascoal Faria, J.; Lago, P.; Semini, L., Eds.; Springer Nature Switzerland: Cham, 2024; Vol. 2178, pp. 328–343. https://doi.org/10.1007/978-3-031-70245-7_23.
8. Guilherme, V.; Vincenzi, A. An initial investigation of ChatGPT unit test generation capability. In Proceedings of the 8th Brazilian Symposium on Systematic and Automated Software Testing, Campo Grande, MS Brazil, 2023; pp. 15–24. https://doi.org/10.1145/3624032.3624035.
9. Schütz, M.; Plösch, R. A Practical Failure Prediction Model based on Code Smells and Software Development Metrics. In Proceedings of the Proceedings of the 4th European Symposium on Software Engineering, Napoli Italy, 2023; pp. 14–22. https://doi.org/10.1145/3651640.3651644.
10. Stradowski, S.; Madeyski, L. Can we Knapsack Software Defect Prediction? Nokia 5G Case. In Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Melbourne, Australia, 2023; pp. 365–369. https://doi.org/10.1109/ICSE-Companion58688.2023.00104.
11. Pandit, M.; Gupta, D.; Anand, D.; Goyal, N.; Aljahdali, H.M.; Mansilla, A.O.; Kadry, S.; Kumar, A. Towards Design and Feasibility Analysis of DePaaS: AI Based Global Unified Software Defect Prediction Framework. *Applied Sciences* **2022**, *12*, 493. https://doi.org/10.3390/app12010493.
12. Yi, G.; Chen, Z.; Chen, Z.; Wong, W.E.; Chau, N. Exploring the Capability of ChatGPT in Test Generation. In Proceedings of the 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C), Chiang Mai, Thailand, 2023; pp. 72–80. https://doi.org/10.1109/QRS-C60940.2023.00013.
13. Olsthoorn, M. More effective test case generation with multiple tribes of AI. In Proceedings of the Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, Pittsburgh Pennsylvania, 2022; pp. 286–290. https://doi.org/10.1145/3510454.3517066.
14. Straubinger, P.; Fraser, G. A Survey on What Developers Think About Testing. In Proceedings of the 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), Florence, Italy, 2023; pp. 80–90. https://doi.org/10.1109/ISSRE59848.2023.00075.
15. German Testing Board. Probeprüfungen, 2025.
16. International Software Testing Qualifications Board. About Us, 2025.
17. Alliance for Qualification. About A4Q, 2025.
18. American Software Testing Qualifications Board. What Are the Levels of ISTQB Exam Questions?, 2025.
19. Vaillant, T.S.; Almeida, F.D.d.; Neto, P.A.M.S.; Gao, C.; Bosch, J.; Almeida, E.S.d. Developers' Perceptions on the Impact of ChatGPT in Software Development: A Survey, 2024. arXiv:2405.12195, https://doi.org/10.48550/arXiv.2405.12195.
20. Newton, P.; Xiromeriti, M. ChatGPT performance on multiple choice question examinations in higher education. A pragmatic scoping review. *Assessment & Evaluation in Higher Education* **2024**, *49*, 781–798. https://doi.org/10.1080/02602938.2023.2299059.
21. Zhu, L.; Mou, W.; Yang, T.; Chen, R. ChatGPT can pass the AHA exams: Open-ended questions outperform multiple-choice format. *Resuscitation* **2023**, *188*, 109783. https://doi.org/10.1016/j.resuscitation.2023.109783.