Smarter, not Bigger: Fine-Tuned RAG-Enhanced LLMs for Automotive Hardware-in-the-Loop Testing

Chao Feng[1], Zihan Liu[1], Siddhant Gupta[2], Jan von der Assen[1]
[1]Communication Systems Group, Department of Informatics, University of Zurich,
[2]Volvo Car Corporation
[cfeng, vonderassen]@ifi.uzh.ch, zihan.liu@uzh.ch, siddhant.gupta@volvocars.com

Abstract

Hardware-in-the-Loop (HIL) testing is essential for automotive validation but suffers from fragmented and underutilized test artifacts. This paper presents HIL-GPT, an industry-deployed retrieval-augmented generation (RAG) system that integrates semantic retrieval with domain-adapted large language models to support test engineers in real-world HIL workflows. The system combines domain-specific embeddings to enable traceable retrieval of test cases and requirements under industrial latency and cost constraints. Through empirical evaluation, we show that compact, domain-adapted models can achieve a favorable trade-off among accuracy, latency, and cost compared to larger general-purpose models, challenging the assumption that larger models are always preferable in industrial NLP systems. An A/B user study further confirms that HIL-GPT improves perceived helpfulness, truthfulness, and satisfaction over general-purpose LLMs.

1 Introduction

Hardware and software testing are critical stages in the automotive development lifecycle, where components must satisfy functional, reliability, and safety requirements before deployment (Garikapati et al., 2024). As vehicle platforms become more complex, manual testing workflows face increasing pressure from cost, time, and operational risk (Gaspar et al., 2024). Hardware-in-the-Loop (HIL) testing is a core validation technique in the automotive industry, where physical control units interact with simulated environments under controlled and repeatable conditions (Cheng et al., 2024). This setting supports regression testing, fault injection, and boundary-case validation before road deployment, at lower risk and lower cost than fully physical testing (Jooriah et al., 2024).

In production HIL environments, large volumes of engineering artifacts accumulate over long development cycles, including requirements, test cases, execution logs, and auxiliary documentation produced by different teams (Ali and Ali, 2024). These artifacts contain substantial validation knowledge, yet they are difficult to reuse efficiently in daily engineering practice. HIL workflows rely on proprietary toolchains and domain-specific engineering representations, including Controller Area Network (CAN) signals, module identifiers, and internal requirement–test links. These properties make retrieval difficult in practice, because relevant information is scattered across closed systems and heterogeneous artifacts, while public training data for such settings remains limited (Cheng et al., 2024).

Recent advances in large language models (LLMs) and retrieval-augmented generation (RAG) have created new opportunities for natural-language access to technical knowledge (Jin et al., 2024). In industrial HIL settings, however, direct use of general-purpose LLMs introduces practical constraints. Answers must remain traceable to engineering artifacts, retrieval must operate over closed and terminology-heavy data sources, and deployment must satisfy latency, cost, and maintainability requirements. These constraints make it insufficient to treat HIL support as a standard enterprise RAG or to assume that general-purpose LLMs will transfer well out of the box (Ji et al., 2023).

This paper presents HIL-GPT, an industry-oriented RAG pipeline for automotive HIL testing that combines semantic retrieval with domain-adapted language models to support natural-language queries over tests and requirements. The focus is on retrieval adaptation under realistic industrial constraints, including heterogeneous artifacts, limited labeled data, traceability requirements, and strict cost and latency budgets. HIL-GPT is designed for integration into existing HIL workflows rather than as a standalone experimental prototype.

HIL-GPT is evaluated through empirical experiments and a user study with automotive test en-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 5–12

gineers. The results show that compact, domain-adapted retrieval models outperform larger generic alternatives when retrieval quality, latency, and deployment cost are considered jointly. These findings indicate that effective industrial HIL assistance depends less on scaling model size alone and more on matching the retrieval pipeline to the artifact space and deployment setting. The paper also reports practical lessons for deploying reliable and efficient RAG-based assistants in industrial validation environments.

2 Background and Related Work

This section provides background on automotive HIL testing and situates our work within prior efforts on applying LLMs in domain-specific engineering settings.

2.1 HIL Testing and CAN-Based Workflows

HIL testing is a standard validation technique in the automotive industry, enabling real hardware components to interact with simulated environments under controlled and repeatable conditions. It is widely used to validate electronic control units and sensors before on-road deployment, supporting regression, fault injection, and boundary testing at lower cost and risk than physical testing (Garikapati et al., 2024; Gaspar et al., 2024).

A typical HIL setup includes a real-time simulation platform, a device under test, signal conditioning hardware, and orchestration software enforcing deterministic closed-loop execution (Howick et al., 2024). Communication commonly relies on the CAN with standardized message identifiers and vendor-specific encodings, supported by tools such as CANoe and CAPL for simulation, event injection, and test automation (Bosch GmbH, 1991; GmbH, 2022).

Despite their maturity, these toolchains produce large volumes of heterogeneous artifacts, including test scripts, signal definitions, logs, and requirements, spread across disconnected tools and formats. Consequently, test knowledge is difficult to retrieve and reuse, and current workflows offer limited support for natural-language access or cross-artifact reasoning, hindering the integration of NLP-based assistance into HIL environments.

2.2 LLMs in Domain-Specific Engineering Applications

Recent work has explored the use of LLMs in technical and engineering domains, typically through model fine-tuning or RAG. Fine-tuning adapts model representations to domain terminology and conventions, while RAG grounds model outputs in external knowledge sources at inference time (Elhambakhsh et al., 2025; Hu and Lu, 2024). These approaches have shown promise in tasks such as mechanical design analysis, aerospace diagnostics, and embedded systems support.

Across these domains, prior studies consistently report two practical challenges: the scarcity of labeled domain data and the high cost of deploying large general-purpose models in production environments. To address these issues, recent work has emphasized embedding-level adaptation, weak supervision, and contrastive learning as cost-effective means of improving retrieval quality under limited data budgets (Gao et al., 2021; Yang, 2024; Nguyen et al., 2024). Empirical evidence suggests that improving retrieval representations often yields larger gains than directly fine-tuning LLMs, particularly in safety-critical or terminology-heavy settings.

However, existing studies do not address the combination of constraints that characterizes automotive HIL environments: CAN-centric artifact spaces, closed industrial toolchains, traceability requirements, and strict deployment limits on latency, cost, and maintainability. The motivation of this paper is therefore not a new RAG architecture in isolation, but a practical retrieval-centered pipeline for this setting. It builds on prior work in domain-adapted LLMs and RAG, while focusing on how semantic retrieval can be adapted and deployed effectively within real automotive HIL workflows.

3 Architecture of HIL-GPT

HIL-GPT consists of an offline knowledge-preparation pipeline and an online retrieval-augmented interaction pipeline. The architecture is designed for industrial HIL environments with heterogeneous artifacts, limited labeled data, traceability requirements, and strict constraints on latency, cost, and maintainability, as illustrated in Figure 1.

HIL-GPT is designed around two principles: (i) grounding all model outputs in traceable engineering artifacts, and (ii) minimizing operational overhead while maintaining acceptable response quality. Rather than relying on a single large language model, the system combines lightweight semantic retrieval with domain-adapted language models to balance accuracy, latency, and cost in industrial deployment.

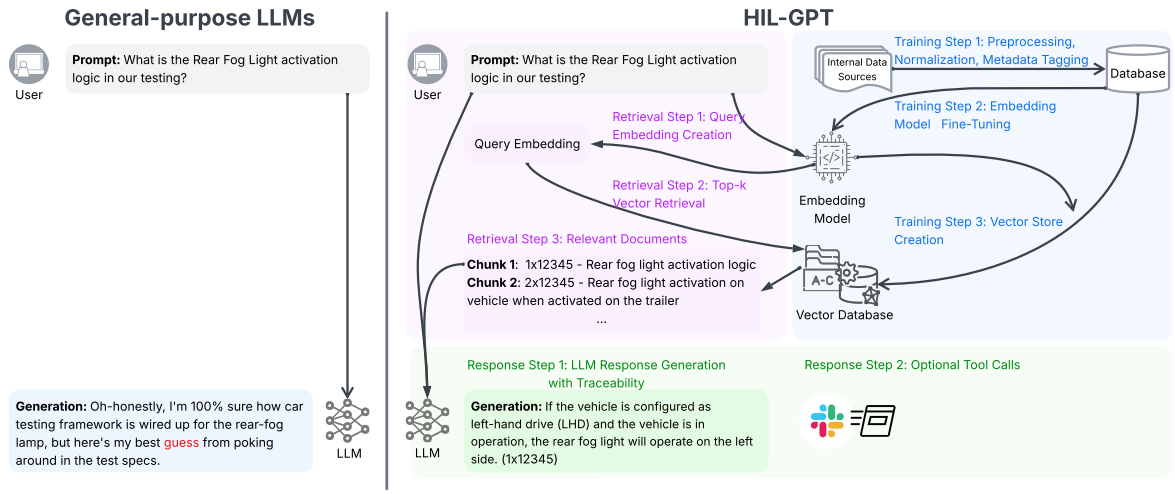

Figure 1: System architecture of HIL-GPT.

3.1 Offline Knowledge Preparation

The offline pipeline transforms heterogeneous HIL artifacts into a retrieval-ready knowledge base. Engineering data, including requirements, test cases, signal metadata, and related documentation, are extracted from existing toolchains and normalized into structured blocks. Each block preserves identifiers and engineering context, such as test case IDs, titles, requirement text, descriptive fields, and test-sequence information, so that retrieved content remains directly traceable to the artifact.

Because manually labeled retrieval data is limited in this setting, HIL-GPT uses a semi-automated weak-supervision pipeline to construct training data for embedding adaptation. Each training instance is organized as an anchor-positive-negative triplet.

Anchors are requirement-like natural-language queries derived from real engineering tasks. They are constructed from requirement text, test case descriptions, and related engineering documentation, and further expanded with LLM-generated paraphrases to increase linguistic coverage while preserving the original testing intent.

Positive examples are retrieved artifacts that correspond to the target functionality of the anchor. They are identified from existing requirement–test associations available in the engineering workflow, including links induced by shared CAN signals, module identifiers, and structural metadata.

Negative examples are domain-plausible but functionally incorrect alternatives. They are selected from artifacts that remain close to the anchor in terminology or subsystem context, but do not match the intended requirement–test relation. To make retrieval discrimination harder, the negative set is further enriched with challenging examples synthesized by an LLM under the same domain.

Embedding adaptation is performed with LoRA using TripletLoss, which improves terminology alignment and retrieval discrimination without the overhead of fine-tuning a large generator. The adapted embedding model is trained with an 80/20 train–validation split, using one negative example per positive instance in each triplet. This retrieval-centered design reflects the practical requirements of industrial environments, where models must remain replaceable, cost-aware, and maintainable under frequent workflow updates.

The resulting artifact collection is indexed with dense vector representations and lightweight metadata filters to support efficient retrieval over the normalized knowledge base.

3.2 Online Retrieval and Interaction

In online operation, engineers interact with HIL-GPT through a chatbot interface. Incoming queries are embedded and matched against the indexed knowledge base to retrieve requirements, test procedures, and context. Retrieved artifacts are assembled into a prompt for the generator, ensuring responses remain grounded in verifiable sources.

The online layer is designed to remain modular. Retrieval, context assembly, generation, and tool execution are handled as separate components, which allows model replacements without changing the overall workflow. This is important in indus-

trial deployment, where model choices may change over time due to cost, latency, or infrastructure constraints.

To support queries that require current system state, HIL-GPT also integrates external tools that expose live HIL information, CAN-related metadata, and predefined backend actions. Tool requests produced by the model are validated before execution, and the returned outputs are incorporated into the final response.

3.3 Traceability

Traceability is a core requirement in automotive validation. HIL-GPT therefore maintains an audit trail for each interaction, including the retrieved artifacts, assembled prompts, tool invocations, and generated responses. These records support inspection, debugging, and process auditing in engineering workflows.

The system is deployed within the enterprise infrastructure and integrated with the existing HIL workflow. In the current deployment, model access is provided through an enterprise Azure OpenAI subscription rather than a public API endpoint, so that proprietary artifacts, prompts, embeddings, and fine-tuning data remain within the organizational deployment boundary.

4 Evaluation

This section evaluates HIL-GPT from an engineering and deployment perspective, with the goal of determining whether the main design choices lead to reliable retrieval, traceable responses, and useful interaction in real automotive HIL workflows. The evaluation is organized around three experimental research questions (ERQs):

- ***ERQ 1:*** Under industrial constraints, how should embedding strategies and data-construction choices be combined to achieve reliable retrieval performance with acceptable cost and latency in HIL workflows?
- ***ERQ 2:*** Does domain-adapted retrieval improve source attribution and reduce unsupported responses in HIL testing tasks?
- ***ERQ 3:*** How do engineers perceive the usefulness and trustworthiness of a RAG-based assistant compared with a general-purpose LLM?

4.1 Evaluation Setup

The evaluation combines offline retrieval analysis, end-to-end system assessment, and exploratory user evaluation. Because explicit relevance labels are generally unavailable in industrial HIL environments, retrieval performance is evaluated using requirement–test associations derived from existing engineering artifacts.

The benchmark contains 2,172 query–target pairs constructed from the internal tests-and-requirements corpus. Each query is a natural-language question grounded in an existing requirement or test procedure, and each pair is associated with an `expected_block_id` that denotes the relevant test artifact. Retrieval correctness is evaluated as block-level top-k accuracy, that is, whether the `expected_block_id` appears in the retrieved `source_ids`. Unless otherwise noted, $k = 5$.

To support controlled comparison, each query is evaluated against the same artifact collection extended with domain-relevant distractors. Indexed blocks contain the test case ID, title, requirements section, description section, and test-sequence information, including steps, actions, and expected results. All experiments use a unified inference pipeline with consistent preprocessing and similarity computation across configurations.

4.2 Embedding Strategy Selection under Industrial Constraints

This experiment addresses *ERQ 1* by examining how different embedding strategies perform under the practical constraints of automotive HIL testing.

Embedding Model Comparison. A range of open-source and proprietary embedding models is first compared to establish baseline retrieval behavior before domain adaptation. Retrieval correctness is measured using top-1 block-level accuracy, which reflects whether the most relevant artifact is ranked first.

As shown in Table 1, larger embedding models do not consistently outperform more compact alternatives. On the full benchmark of 2,172 queries, the best-performing open-source model, `gtr-t5-large`, reached 58.2%. Notably, the compact `bge-base-en-v1.5`, despite having only 110M parameters, achieved a baseline accuracy of 50.7%, making it a viable candidate for cost-sensitive adaptation.

Table 1: Retrieval accuracy on full benchmark (2,172 queries).

Model	Params (M)	Acc. (%)
`text-embedding-ada-002` (2022)	—	58.89
`text-embedding-3-small` (2024c)	—	58.70
`gtr-t5-large` (a)	334.94	58.24
`gtr-t5-xl` (b)	1240.91	57.32
`bge-base-en-v1.5` (BAAI)	110.00	50.69

Impact of Domain Adaptation. The second step evaluates whether lightweight domain adaptation improves retrieval robustness. Selected open-source models are adapted using automatically constructed anchor–positive–negative triplets derived from existing engineering artifacts. This setup reflects realistic industrial conditions, where explicit relevance labels and manually curated negatives are limited.

Figure 2 shows that domain adaptation yields the largest gains for smaller models. In particular, adapting `bge-base-en-v1.5` increases top-1 accuracy from 50.7% to 60.7%, an absolute gain of 10.0 percentage points. By contrast, larger models such as `gtr-t5-large` show gains below 2 percentage points, which indicates weaker returns from adaptation under limited domain supervision.

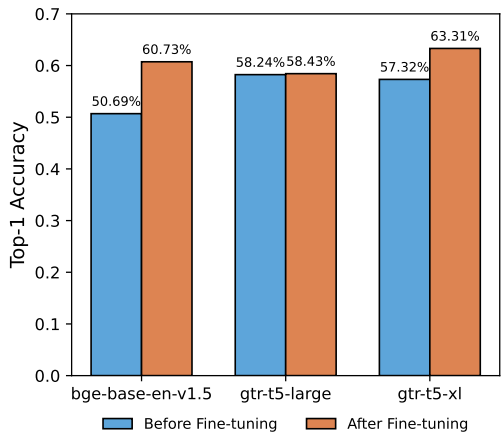

Figure 2: Top-1 retrieval accuracy before and after domain adaptation.

Effect of Data Construction Choices. The third step studies how data-construction choices affect retrieval under limited supervision. Table 2 reports results for three regimes: no adaptation, adaptation without negatives, and adaptation with full triplets. Using only positive associations increases accuracy from 50.7% to 55.8%. Adding automatically generated negative examples further increases accuracy to 60.7%, which shows that even limited contrastive information sharpens retrieval boundaries.

Table 2: Top-1 retrieval accuracy of `bge-base-en-v1.5` under different training regimes.

Base	Positive only	Positive + Negative
50.69%	55.76%	60.73%

Cost and Latency Considerations. Cost and latency are central constraints in industrial deployment. Table 3 summarizes the estimated adaptation cost and average embedding inference latency of the three evaluated models. Larger models provide only marginal retrieval gains at substantially higher cost and latency, which reduces their practical value in interactive HIL workflows. Adapting `bge-base-en-v1.5` required approximately USD 50 and yielded an average embedding inference latency of 15 ms per query, compared with approximately USD 150 and 45 ms for `gtr-t5-large`, and approximately USD 600 and 120 ms for `gtr-t5-xl`. These results indicate that the compact model provides the strongest retrieval–cost–latency trade-off in this setting.

Table 3: Estimated fine-tuning cost and average embedding inference latency.

Model	Cost (USD)	Latency (ms)
`bge-base-en-v1.5`	∼50	∼15
`gtr-t5-large`	∼150	∼45
`gtr-t5-xl`	∼600	∼120

Overall, these results indicate that effective embedding strategies in industrial NLP systems depend more on domain alignment and data-construction quality than on raw model capacity alone.

4.3 Impact of Domain-Adapted Retrieval on Source Attribution

This experiment addresses *ERQ 2*: whether domain-adapted retrieval improves source attribution in HIL testing tasks under a fixed RAG setup.

In industrial HIL workflows, the main requirement is not only fluent generation, but whether the response remains traceable to the underlying engineering artifact. The metric used here therefore evaluates source attribution rather than full factual correctness or sentence-level citation faithfulness.

Two LLMs, `GPT-4o` (OpenAI, 2024a) and `GPT-4o-mini` (OpenAI, 2024b), are evaluated on the same set of domain-specific queries. For each

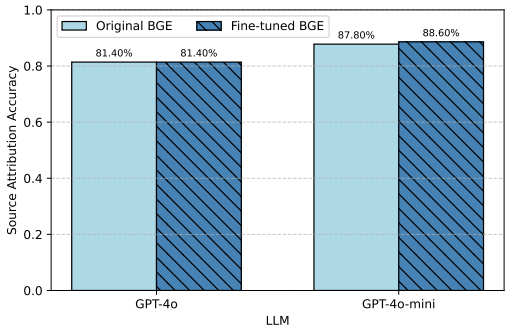

Figure 3: Source-attribution accuracy of `GPT-4o` and `GPT-4o-mini` using original and adapted `bge-base-en-v1.5` embeddings.

query, the top-5 artifacts are retrieved using either the original `bge-base-en-v1.5` embeddings or the domain-adapted variant selected in ERQ 1. The retrieved artifacts are injected into the same RAG prompt template. The model is instructed to answer the query and return the identifier of the artifact it relied on. A response is counted as correctly attributed if the returned identifier matches the known relevant artifact within the provided context.

As shown in Figure 3, domain-adapted retrieval improves source-attribution accuracy under the same RAG configuration. For `GPT-4o-mini`, accuracy increases to 88.6% when switching from the original to the adapted embeddings. For `GPT-4o`, the increase is minimal, even though both models receive the same retrieval structure and prompt template. These results indicate that retrieval quality remains a primary factor in response attribution, especially for smaller and more cost-efficient generators.

4.4 Exploratory User Evaluation of Practical Utility

This experiment addresses *ERQ 3*: how engineers perceive the usefulness and trustworthiness of a RAG-based assistant compared with a general-purpose LLM in realistic HIL testing tasks.

Ten professional engineers from two functional domains participated in the study. The evaluated tasks mainly focused on testcase generation and related test-engineering queries used by the test department and the CI/DevOps team. Participants interacted with two chatbot variants: HIL-GPT with RAG support and a general-purpose LLM (`GPT-4o-mini`) without retrieval support. For each task, participants were shown anonymized answer pairs generated by the two systems for the same query, without knowing which system produced which answer.

Across the evaluated tasks, the RAG-based assistant was preferred as more helpful in the majority of cases. Participants also rated it higher in completeness, trustworthiness, and overall satisfaction. Qualitative feedback suggests that responses in familiar terminology and identifiable engineering artifacts improved confidence and made verification easier.

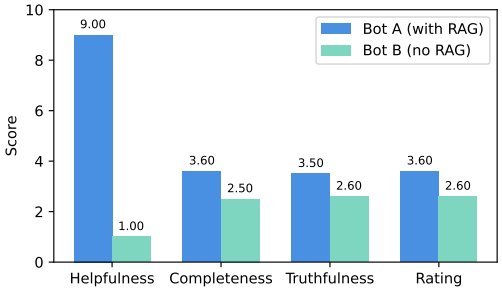

Figure 4: User evaluation ratings: Bot A (with RAG) vs. Bot B (no RAG).

The study also revealed a latency trade-off. The RAG-based system produced more detailed and context-aware responses, but with higher end-to-end latency than the general-purpose baseline. This suggests that different usage scenarios may favor different response profiles, and motivates adaptive configurations that balance response time and informational depth.

5 Lessons Learned

Several observations emerge from the evaluation. First, lightweight domain adaptation is most effective for smaller retrieval models: `bge-base-en-v1.5` improved from 50.7% to 60.7%, whereas larger models gained less than 2 percentage points under the same data constraints. Second, data-construction choices substantially influence retrieval quality. Using only positive associations increased accuracy to 55.8%, while adding a small set of automatically generated negative examples further raised performance to 60.7%. Third, deployment feasibility is shaped by cost and latency rather than retrieval accuracy alone. Adapting `bge-base-en-v1.5` required approximately USD 50 with 15 ms embedding inference latency, compared with approximately USD 150/45 ms and USD 600/120 ms for the larger

alternatives. Fourth, retrieval quality directly affects downstream source-attribution performance and perceived utility. For `GPT-4o-mini`, source-attribution accuracy increased to 88.6% when using adapted embeddings, and the RAG-based assistant was preferred in 9 out of 10 sessions in the exploratory user study. At the same time, several users reported that the RAG-enabled system showed higher end-to-end latency, with average response time increasing from 2.4 s to 17.4 s. This result indicates a practical trade-off between response traceability and responsiveness in interactive HIL workflows.

6 Conclusion and Future Work

This paper presented HIL-GPT, an industry-oriented RAG system for supporting requirement interpretation and test-knowledge access in automotive Hardware-in-the-Loop testing [1]. By focusing on retrieval adaptation under realistic industrial constraints, the evaluation showed that domain adaptation improves retrieval reliability and downstream source-attribution performance, with particularly strong gains for compact embedding models and smaller language models. The exploratory user study further indicated that responses grounded in domain-specific artifacts improve perceived usefulness and trustworthiness, which highlights the importance of retrieval quality and deployment-aware model selection over raw model scale in industrial systems.

The effectiveness of HIL-GPT depends on the quality and coverage of the underlying HIL artifacts. Fragmented or outdated requirements and test sequences reduce retrieval quality and source-attribution performance. In addition, the evaluation is confined to a specific automotive HIL environment and a limited set of workflows, so the observed gains do not directly transfer to other domains without additional adaptation. The current system also prioritizes response traceability over latency, which is not always suitable for time-critical use cases.

Future work will investigate adaptive strategies that balance response depth and latency based on task urgency and user intent, as well as broader validation across heterogeneous HIL configurations, domains, and artifact types.

[1] A previous version of this work appeared as a preprint on arXiv:2511.22584 .

Acknowledgements

This work has been supported by the University of Zürich UZH and by Volvo Cars, which funded and supported the project. Volvo Cars provided AI training data, AI training infrastructure, and expertise in automotive HIL testing.

Disclaimer

The views and opinions expressed are those of the authors and do not necessarily reflect the official policy or position of Volvo Cars.

References

Zeina Ali and Qutaiba I Ali. 2024. An efficient design of a basic autonomous vehicle based on can bus. *International Transactions on Electrical Engineering and Computer Science*, 3(1):41–56.

Bosch GmbH. 1991. *CAN Specification Version 2.0*. Classical CAN protocol standard (Part A/B).

Jingjun Cheng, Zhen Wang, Xiangmo Zhao, Zhigang Xu, Ming Ding, and Kazuya Takeda. 2024. A survey on testbench-based vehicle-in-the-loop simulation testing for autonomous vehicles: Architecture, principle, and equipment. *Advanced Intelligent Systems*, 6(6):2300778.

Fatemeh Elhambakhsh, Daniele Grandi, and Hyunwoong Ko. 2025. A domain adaptation of large language models for classifying mechanical assembly components. *arXiv preprint arXiv:2505.01627*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Divya Garikapati, Yiting Liu, Matt Brown, Tristan Littlehale, Hirofumi Yamamoto, and Chen Bao. 2024. Dual-cockpit human and hardware-in-the-loop test bench for autonomous vehicle development. *IEEE Transactions on Intelligent Vehicles*.

José F Gaspar, Rafael F Pinheiro, Mário JGC Mendes, Mojtaba Kamarlouei, and C Guedes Soares. 2024. Review on hardware-in-the-loop simulation of wave energy converters and power take-offs. *Renewable and Sustainable Energy Reviews*, 191:114144.

Vector Informatik GmbH. 2022. Capl scripting. Webpage. "CAPL is an acronym for Communication Access Programming Language, which is a programming language used in Vector testing tools chain.".

Susan Howick, Itamar Megiddo, and 1 others. 2024. A framework for conceptualising hybrid system dynamics and agent-based simulation models. *European Journal of Operational Research*, 315(3):1153–1166.

Yucheng Hu and Yuxing Lu. 2024. Rag and rau: A survey on retrieval-augmented language model in natural language processing. *arXiv preprint arXiv:2404.19543*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Haibo Jin, Leyang Hu, Xinuo Li, Peiyan Zhang, Chonghan Chen, Jun Zhuang, and Haohan Wang. 2024. Jailbreakzoo: Survey, landscapes, and horizons in jailbreaking large language and vision-language models. *arXiv preprint arXiv:2407.01599*.

Mohannad Jooriah, Daryna Datsenko, João Almeida, Ana Sousa, João Silva, and Joaquim Ferreira. 2024. A co-simulation platform for v2x-based cooperative driving automation systems. In *2024 IEEE Vehicular Networking Conference (VNC)*, pages 227–230. IEEE.

Zooey Nguyen, Anthony Annunziata, Vinh Luong, Sang Dinh, Quynh Le, Anh Hai Ha, Chanh Le, Hong An Phan, Shruti Raghavan, and Christopher Nguyen. 2024. Enhancing q&a with domain-specific fine-tuning and iterative reasoning: A comparative study. *arXiv preprint arXiv:2404.11792*.

Beijing Academy of Artificial Intelligence (BAAI). 2023. Baai general embedding (bge) models: Bge-base-en-v1.5. https://huggingface.co/BAAI/bge-base-en-v1.5. Last Accessed: 2026-01-07.

OpenAI. 2022. text-embedding-ada-002 model card. https://platform.openai.com/docs/guides/embeddings/what-are-embeddings. Last Accessed: 2026-01-07.

OpenAI. 2024a. Gpt-4o. https://platform.openai.com/docs/models/gpt-4o. Last Accessed: 2026-01-07.

OpenAI. 2024b. Gpt-4o-mini: Lightweight variant of gpt-4o. https://platform.openai.com/docs/models/gpt-4o. Last Accessed: 2026-01-07.

OpenAI. 2024c. text-embedding-3-small model card. https://platform.openai.com/docs/guides/embeddings/what-are-embeddings. Last Accessed: 2026-01-07.

Google Research. a. gtr-t5-large: Google's text retrieval transformer. https://huggingface.co/google/gtr-t5-large. Last Accessed: 2026-01-07.

Google Research. b. gtr-t5-xl: Google's text retrieval transformer. https://huggingface.co/google/gtr-t5-xl. Last Accessed: 2026-01-07.

Jinbiao Yang. 2024. Rethinking tokenization: Crafting better tokenizers for large language models. *International Journal of Chinese Linguistics*, 11(1):94–109.