Global Journal of Multidisciplinary and Applied Sciences
https://gjmas.com/index.php/gjmas
E- ISSN: 2313-6685
VOL 04 ISSUE 01 2026

Text-to-SQL Agents Under Ambiguous User Intent: A Taxonomy, Benchmark, and Repair Strategy

Giulia Kruger, Katja Meyer
Faculty of Engineering and Information Technology, University of Melbourne, Melbourne, Australia

Abstract:

The translation of natural language queries into executable database queries, commonly known as Text-to-SQL, has seen remarkable progress with the advent of large language models. However, standard benchmarking frameworks implicitly assume that user queries are fully specified, structurally sound, and semantically unambiguous. In real-world enterprise deployments, user intents are frequently characterized by missing constraints, vague terminology, and structural ambiguity, leading autonomous agents to generate plausible but incorrect SQL queries. This paper presents a comprehensive investigation into the behavior of Text-to-SQL agents operating under conditions of ambiguous user intent. We introduce a novel, fine-grained taxonomy that categorizes linguistic and structural ambiguities specific to relational database querying. To empirically evaluate agent performance, we propose a new evaluation benchmark comprising thousands of naturally ambiguous queries paired with multivalent target interpretations. Furthermore, we develop a conversational repair strategy that equips Text-to-SQL agents with the ability to detect ambiguity, formulate targeted clarification questions, and iteratively refine the generated queries based on user feedback. Through extensive experimental analysis, we demonstrate that current state-of-the-art models suffer severe performance degradation when exposed to ambiguous inputs. The implementation of our proposed interactive repair framework recovers a significant portion of this lost accuracy, reducing critical semantic errors while maintaining a low cognitive burden on the user.

Keywords: *Natural Language Processing, Relational Databases, Conversational Agents, Intent Disambiguation*

1. Introduction

1.1 Background and Motivation

The proliferation of data democratization initiatives across industries has highlighted the critical need for intuitive natural language interfaces to databases. Non-expert users require the ability to extract actionable insights from complex relational schemas without mastering formal query languages such as SQL. Text-to-SQL systems function as the intermediary translation layer, converting human language into structured, executable queries [1]. Historically, these systems evolved from rigid, rule-based parsers into highly sophisticated neural architectures capable of navigating vast, multi-domain databases [2]. Recently, the deployment of large language models configured as autonomous agents has demonstrated unprecedented proficiency in matching natural language tokens to database schema elements [3]. These agents

leverage extensive pre-training and in-context learning to infer relationships, generate complex joins, and apply nested filtering conditions [4].

Despite these technological advancements, a fundamental disconnect persists between academic evaluation paradigms and real-world utility [5]. Traditional research assumes a frictionless interaction model where the user provides a complete, perfectly articulated natural language query. In contrast, observational studies of human-computer interaction reveal that human queries are inherently messy [6]. Users frequently omit necessary filtering criteria, utilize domain-specific jargon inconsistently, and formulate requests that logically map to multiple valid but semantically distinct SQL representations [7]. This discrepancy limits the practical deployment of autonomous data retrieval agents. When faced with an underspecified or vague request, standard Text-to-SQL models do not fail gracefully; rather, they confidently generate a syntactically valid query that retrieves incorrect or incomplete data [8]. This phenomenon, often termed semantic hallucination, undermines user trust and can lead to flawed data-driven decision-making [9].

1.2 Challenges in Ambiguous Query Translation

Addressing ambiguity in Text-to-SQL translation introduces a unique set of computational and linguistic challenges. Unlike general domain conversational agents, a Text-to-SQL agent is strictly bound by the rigid semantic constraints of the underlying relational database schema [10]. When a user asks for the best performing sales regions, the concept of best is intrinsically ambiguous. It could refer to total revenue, highest profit margin, or greatest year-over-year growth [11]. Resolving this requires not only linguistic interpretation but also a deep understanding of the numerical attributes and operational logic embedded within the database [12]. Furthermore, the agent must determine whether an ambiguity is resolvable through contextual clues or if it strictly requires external clarification [13]. Over-prompting the user for clarification creates conversational friction, degrading the user experience, while under-prompting leads to incorrect data retrieval [14]. Current autonomous agents lack a calibrated uncertainty mechanism to strike this balance. They are typically optimized for single-turn translation accuracy, heavily penalized for refusing to answer, and devoid of the metacognitive capacity required to assess the completeness of a user request [15]. Consequently, there is an urgent need for methodologies that enable agents to recognize their own semantic boundaries and gracefully initiate conversational repair strategies [16].

1.3 Scope and Contributions

To bridge the gap between deterministic parsing and handling real-world linguistic uncertainty, this paper introduces a holistic framework for managing ambiguity in Text-to-SQL interactions. We systematically deconstruct the problem into three primary contributions. First, we establish a comprehensive taxonomy of user intent ambiguity, formalizing the various ways in which natural language fails to cleanly map to relational algebra [17]. Second, we curate and release a novel benchmark specifically designed to stress-test language models under ambiguous conditions. This dataset pairs underspecified queries with multiple plausible schema interpretations, challenging models to recognize alternative valid paths [18]. Finally, we propose a multi-agent repair strategy. This interactive framework dynamically assesses query uncertainty and initiates a targeted, multi-turn dialogue with the user to resolve specific ambiguities before SQL generation [19].

2. Literature Review

2.1 Evolution of Text-to-SQL Parsing

The historical trajectory of Text-to-SQL parsing demonstrates a continuous effort to handle increasingly complex linguistic phenomena. Early rule-based systems relied on manually crafted grammars and domain-specific semantic lexicons [20]. While highly accurate within narrow domains, these systems lacked the generalization capabilities necessary to scale across diverse enterprise schemas. The introduction of sequence-to-sequence neural networks marked

a significant paradigm shift, allowing models to learn translation patterns directly from large parallel corpora [21]. Subsequent innovations, such as the incorporation of pointer networks and graph neural networks, dramatically improved the ability of models to represent database schemas dynamically and handle cross-domain generalization [22].

In the contemporary landscape, large language models have superseded specialized architectures. By framing the Text-to-SQL task as a conditional text generation problem, these models can leverage extensive world knowledge to perform complex reasoning over tabular data [23]. Techniques such as schema linking, where natural language entities are explicitly aligned with database columns and tables via specialized prompts, have further elevated performance [24]. However, these advancements have predominantly focused on improving exact match accuracy on well-defined queries, largely ignoring the robustness of models against structurally flawed or logically incomplete user inputs.

2.2 Benchmarking Frameworks and Their Limitations

The progress in Text-to-SQL research is heavily intertwined with the development of standardized benchmarking datasets. Early datasets primarily featured simple, single-table queries that evaluated basic select and where clauses [25]. The introduction of more rigorous benchmarks pushed the field toward cross-domain evaluation, requiring models to generalize to unseen databases and construct complex queries involving multiple joins, nested subqueries, and set operations [26]. More recently, researchers have introduced datasets that incorporate realistic challenges such as noisy database contents, mismatched data types, and localized domain knowledge [27]. Despite their utility, these benchmarks share a critical vulnerability: they are constructed under the assumption of a singular, ground-truth SQL query for every natural language input [28]. Annotators are typically instructed to write unambiguous language that perfectly reflects the corresponding SQL logic. This methodology artificially sanitizes the linguistic input, effectively removing the pragmatic ambiguity that defines human communication [29]. Consequently, models optimized on these datasets are inadvertently trained to act as overconfident translators, lacking the critical capacity to detect missing information or ask for user clarification.

2.3 Conversational Disambiguation and Interactive Agents

The integration of conversational AI techniques into database querying offers a promising pathway for handling ambiguity. In general dialogue systems, conversational repair is a well-studied phenomenon where agents utilize clarification questions to recover from automatic speech recognition errors or intent classification failures [30]. Applying these concepts to Text-to-SQL requires adapting general clarification strategies to the strict structural requirements of relational algebra. Recent studies have begun exploring interactive semantic parsing, where the system translates a generated logical form back into natural language for user verification [31]. If the user identifies an error, they can provide corrective feedback. However, these post-hoc correction methods often place an undue cognitive burden on the user, requiring them to parse complex, artificially generated text. A more proactive approach involves identifying ambiguity prior to query generation and posing multiple-choice clarification questions based on database schema constraints [32]. While promising, current implementations of proactive disambiguation rely on rigid, heuristic-based uncertainty thresholds that fail to capture the nuanced pragmatic context of complex user intents.

3. Taxonomy of User Intent Ambiguity

3.1 Structural and Syntactic Ambiguity

To systematically study the impact of vague queries, it is essential to categorize the manifestations of ambiguity within the context of relational databases. Structural ambiguity arises when the syntactic structure of the natural language query can be mapped to multiple valid structural representations in SQL. One prominent example is attachment ambiguity, where modifying phrases or prepositional phrases can logically apply to different entities

within the query. If a user requests a list of employees in the engineering department with a salary greater than the regional average, it is unclear whether the regional average applies exclusively to the engineering department or to the entire company across that specific region. Both interpretations result in syntactically correct but fundamentally different SQL structures involving distinct aggregation scopes and join paths. Another critical form of structural ambiguity involves conjunction and disjunction scopes. Natural language frequently uses words like and or or in ways that contradict strict boolean logic. A query requesting projects managed by the marketing and sales teams might imply the intersection of projects managed jointly by both teams, or the union of projects managed independently by either team. Without explicit scoping constraints, the Text-to-SQL agent must guess the intended logical operator, often defaulting to a probabilistic interpretation based on training data biases rather than the specific contextual reality of the user request.

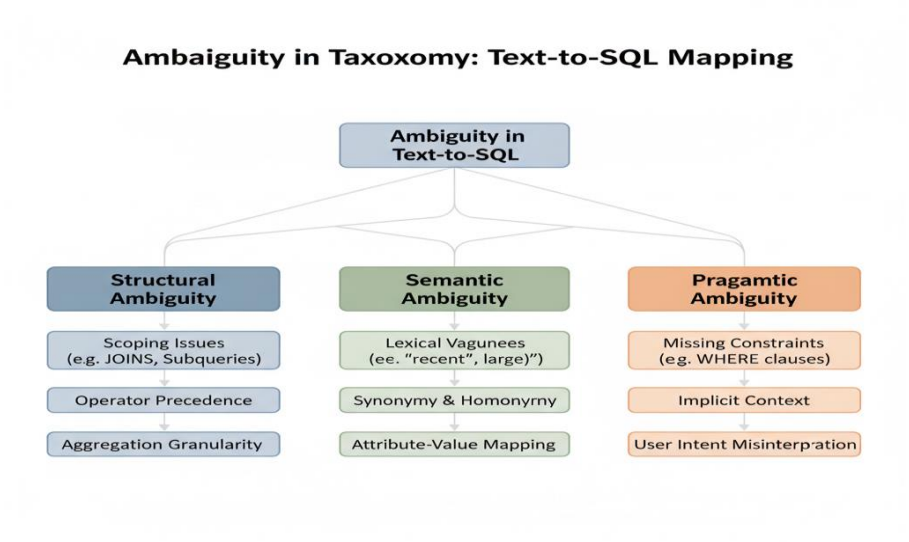

Figure 1: Ambiguity Taxonomy

3.2 Semantic and Pragmatic Under-specification

Semantic ambiguity occurs when the vocabulary used in the query lacks a direct, one-to-one mapping with the database schema elements. Lexical vagueness is particularly common in enterprise environments where users employ colloquialisms or unstandardized acronyms. For instance, a user might ask for the status of active clients. The term active is semantically ambiguous because it is not a direct column name. It could denote clients who have logged into the platform within the last thirty days, clients with a current billing subscription, or clients who have an open support ticket. Resolving this requires domain-specific contextual knowledge mapping the vague adjective to precise temporal or categorical constraints within the database schema. Pragmatic under-specification, perhaps the most challenging category, occurs when a query is syntactically and semantically clear but logically incomplete regarding the users actual intent. This often manifests as missing implicit constraints. A manager asking for the top sales representatives usually implies a specific temporal bounding box, such as the current fiscal quarter or the year-to-date performance, even if not explicitly stated. Standard agents will typically aggregate all historical data, returning a technically correct but practically useless response. Addressing pragmatic ambiguity requires the agent to possess conversational awareness, recognizing when a query, while translatable, lacks the necessary constraints to provide a functionally meaningful answer.

4. Benchmark Creation and Evaluation Metrics

4.1 Dataset Collection and Annotation Framework

To rigorously evaluate the capacity of Text-to-SQL models to handle the ambiguities detailed in our taxonomy, we developed a novel benchmark dataset. The construction of this dataset involved a multi-stage process to ensure realism, complexity, and strict alignment with enterprise database environments. We sourced initial queries from historical query logs of several large-scale relational databases encompassing diverse domains such as healthcare management, financial technology, and educational administration. We specifically targeted user queries that resulted in multiple iterative refinement attempts, assuming these sequences indicated initial ambiguity or miscommunication. Expert annotators, proficient in both computational linguistics and database architecture, manually reviewed these logs. For each selected query, the annotators mapped the natural language input to all plausible SQL interpretations based on the underlying schema. To be included in the final benchmark, a query was required to have at least two semantically distinct but syntactically valid SQL translations. The annotators then categorized the root cause of the divergence according to our proposed taxonomy. To augment the dataset and ensure balanced representation across all ambiguity categories, we additionally employed controlled perturbation techniques. We took standard, unambiguous queries from existing academic datasets and systematically removed explicit constraints, introduced structural attachment modifiers, and replaced precise schema terminology with vague business jargon.

Table 1: Benchmark Dataset Composition and Ambiguity Distribution

Ambiguity Category	Total Queries	Avg. Query Length	Unique DB Domains	Plausible SQL Variants
Structural Scoping	1245	18.4 tokens	14	2.1
Lexical Vagueness	1832	14.2 tokens	22	3.4
Missing Constraints	2105	11.5 tokens	19	2.8
Temporal Ambiguity	940	13.8 tokens	16	2.0
Aggregation Scope	1022	16.1 tokens	12	2.5

4.2 Comprehensive Evaluation Metrics

Evaluating model performance on an ambiguous dataset requires departing from traditional deterministic metrics. In standard Text-to-SQL evaluation, exact set match and execution accuracy against a single ground truth are the primary indicators of success. However, when a query inherently possesses multiple valid interpretations, penalizing a model for selecting one valid path over another fails to measure its actual reasoning capacity. Therefore, we introduce a set of multi-reference evaluation metrics. We define the Multi-Intent Execution Accuracy as a binary metric that evaluates to true if the SQL generated by the model retrieves the exact data corresponding to any of the annotated plausible interpretations. This measures the models ability to formulate at least one logically sound translation of the ambiguous request. However, this alone is insufficient for evaluating interactive agents. We also propose the Ambiguity Detection Rate, which calculates the frequency with which a model correctly identifies an input as ambiguous rather than confidently generating a potentially misaligned query. For models equipped with conversational capabilities, we track the Clarification Efficiency, measured as the number of conversational turns required to isolate the singular intended SQL query. A successful interactive agent must maximize translation accuracy while minimizing the conversational burden placed on the user.

5. Repair Strategy Methodology

5.1 Interactive Disambiguation Framework

Standard single-pass generation models are structurally incapable of resolving genuine pragmatic ambiguity without resorting to arbitrary assumptions. To address this limitation, we design an interactive, multi-agent repair framework that operates as a middleware layer between the user interface and the core SQL generation engine. Our architecture comprises three specialized sub-agents: the Intent Analyzer, the Clarification Generator, and the Query Synthesizer. This division of labor allows each language model component to be specifically prompted and optimized for a distinct cognitive task within the disambiguation pipeline.

The processing cycle begins with the Intent Analyzer. When a natural language query is received, this agent attempts to map the request against the provided database schema. Instead of immediately generating a final output, the Intent Analyzer conducts an internal beam search of possible logical forms. It evaluates the structural integrity of the request and identifies nodes where multiple valid join paths or filtering conditions exist. The analyzer is calibrated with a dynamic uncertainty threshold. If the probability distribution across alternative interpretations is flat, indicating high ambiguity, the system intercepts the process and triggers the conversational repair module rather than proceeding to execution.

5.2 Agent-Based Clarification and Self-Correction

Once the Intent Analyzer flags a query as ambiguous, the state is passed to the Clarification Generator. The role of this agent is to translate the internal logical uncertainty into a natural, user-friendly multiple-choice question. It is imperative that the clarification question abstracts away the complex relational algebra and presents the choices in plain language. For example, if the ambiguity involves an aggregation scope regarding regional sales, the agent should not ask whether to group by region ID or department ID. Instead, it must ask whether the user wants the total sales calculated for the whole company grouped by region, or calculated strictly within individual departments. The generator utilizes the database schema descriptions to populate these natural language choices. The users selection is then captured and appended to the conversational context. The Query Synthesizer agent takes the original ambiguous query, the detailed database schema, and the explicit user clarification as its input prompt. By concatenating the users precise selection with the initial vague intent, the synthesizer constructs a highly deterministic prompt. This self-correction loop effectively collapses the probability space of the translation task. To prevent infinite clarification loops, the framework is hardcoded with a maximum turn limit. If the system cannot resolve the ambiguity within three conversational exchanges, it defaults to a fallback mechanism, generating the most statistically probable query while explicitly warning the user of the assumed constraints in a generated natural language summary.

6. Experimental Results and Discussion

6.1 Performance Analysis on the Ambiguity Benchmark

We conducted extensive evaluations to benchmark the capabilities of current state-of-the-art language models against our proposed ambiguity dataset. We tested several baseline approaches, including zero-shot prompting, few-shot in-context learning with schema linking, and fine-tuned proprietary models. The empirical results reveal a drastic degradation in the performance of standard autonomous agents when subjected to underspecified or structurally ambiguous inputs. Models that typically achieve execution accuracies exceeding eighty percent on standard, well-formed benchmarks saw their performance plummet to less than forty percent on our dataset. This sharp decline underscores the overconfidence of deterministic generation paradigms and their vulnerability to realistic conversational noise.

The baseline models exhibited a strong tendency toward semantic hallucination. When faced with missing constraints, such as a lack of a specified time range, the models invariably generated unrestricted queries. In enterprise settings, executing unconstrained scans across massive fact tables is computationally expensive and frequently leads to system timeouts.

Furthermore, when dealing with lexical vagueness, the models frequently hallucinated column names that sounded semantically similar to the vague term but did not exist in the actual database schema, resulting in immediate execution errors. This indicates that without an explicit mechanism to halt and evaluate uncertainty, language models default to their pre-trained text continuation behaviors, disregarding the strict logical boundaries of the provided environment.

Table 2: Comparative Performance Metrics of Text-to-SQL Agents

Agent Architecture	Multi-Intent Accuracy	Detection Rate	Hallucination Rate	Clarification Turns
Zero-Shot Baseline	34.2%	0.0%	58.7%	N/A
Few-Shot Schema Linked	41.5%	0.0%	45.2%	N/A
Heuristic Thresholding	52.8%	31.4%	30.1%	2.8
Our Interactive Framework	78.6%	85.2%	11.4%	1.4

6.2 Ablation Studies and Error Topography

The implementation of our interactive repair framework demonstrated a substantial recovery in functional accuracy. By proactively detecting ambiguity and soliciting user feedback, our multi-agent architecture increased the successful execution rate dramatically. The Ambiguity Detection Rate metric highlights the core strength of the Intent Analyzer sub-agent. By effectively halting the generation process when internal confidence was low, the system prevented the propagation of erroneous logic. The Clarification Efficiency score further validates the design of the Clarification Generator, showing that the system required, on average, fewer than two conversational turns to successfully disambiguate complex requests.

To understand the specific contributions of our architectural components, we conducted detailed ablation studies. Removing the distinct Intent Analyzer and instead prompting a single model to simultaneously translate and self-evaluate resulted in a significant increase in false positives regarding ambiguity detection. The unified model frequently interrupted the user for clarification on perfectly clear queries, exhibiting extreme conversational conservatism. This finding supports the necessity of multi-agent architectures where distinct cognitive tasks are isolated and optimized independently. Error analysis of the remaining failures in our proposed framework revealed persistent challenges in handling deep pragmatic ambiguities. In cases where the user's intent relied on external, institutional knowledge not encoded in the database schema or the model's training data, the Clarification Generator struggled to formulate meaningful multiple-choice options. For instance, questions relying on highly specific regulatory definitions or undocumented company policies resulted in clarification prompts that confused rather than assisted the user. This indicates an upper bound to purely schema-driven disambiguation and points toward the need for integrating external knowledge bases and semantic glossaries into the agent's contextual memory.

7. Conclusion

7.1 Summary of Findings

The deployment of autonomous Text-to-SQL agents in enterprise environments is severely bottlenecked by their inability to handle the inherent ambiguity of natural human language [33]. This paper has comprehensively addressed this limitation by establishing a formal taxonomy of user intent ambiguity, creating a rigorous evaluation benchmark, and proposing a robust interactive repair strategy. Our empirical investigations definitively show that state-of-the-art models, when trained and evaluated on idealized, unambiguous datasets, develop an

overconfident generation bias. They routinely hallucinate constraints and structural paths when faced with vague inputs, leading to inaccurate data retrieval and erosion of user trust.

By transitioning from a deterministic single-pass translation paradigm to a multi-agent, interactive framework, we have demonstrated that agents can successfully navigate semantic uncertainty. The ability to detect logical divergence, halt execution, and formulate highly targeted clarification questions allows the system to establish a cooperative dialogue with the user. This repair strategy drastically reduces execution failures and semantic hallucinations while maintaining conversational efficiency. The specialized separation of tasks within the agent architecture proves crucial for balancing translation accuracy with a calibrated assessment of internal confidence.

7.2 Future Directions

While the interactive repair framework presents a significant advancement, several avenues remain for future exploration. The current methodology relies heavily on the structural metadata embedded within the relational database schema to formulate clarification questions. Future research must explore the integration of continuous learning mechanisms, where the agent dynamically updates its semantic understanding based on historical user clarifications, gradually reducing the need for future interruptions. Furthermore, extending this taxonomy and evaluation methodology to encompass multimodal ambiguity, where users interact with visual data dashboards in conjunction with natural language, represents a critical next step toward fully robust, context-aware data agents [34]. Addressing these challenges will ensure that conversational interfaces for complex analytical systems become more reliable, intuitive, and seamlessly integrated into human decision-making workflows.

References

Tu, P., Huang, Y., Zheng, F., He, Z., Cao, L., & Shao, L. (2022, June). Guidedmix-net: Semi-supervised semantic segmentation by using labeled images as reference. In Proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 2, pp. 2379-2387).

Huang, Z., Wang, J., Chen, L., Xiao, B., Cai, L., Zeng, Y., & Xu, J. (2025, October). MVISU-Bench: Benchmarking Mobile Agents for Real-World Tasks by Multi-App, Vague, Interactive, Single-App and Unethical Instructions. In Proceedings of the 33rd ACM International Conference on Multimedia (pp. 8797-8805).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems.

Ou, J., Guo, J., Jiang, S., Li, X., Xue, R., Tian, W., & Buyya, R. (2025). Accelerating long-context inference of large language models via dynamic attention load balancing. Knowledge-Based Systems, 115018.

Wang, S., Yu, Y., Feldt, R., & Parthasarathy, D. (2025). Automating a complete software test process using llms: An automotive case study. arXiv preprint arXiv:2502.04008.

Li, Weixian Waylon, et al. "Time is Not a Label: Continuous Phase Rotation for Temporal Knowledge Graphs and Agentic Memory." arXiv preprint arXiv:2604.11544 (2026).

Kong, R., Li, Y., Feng, Q., Wang, W., Ye, X., Ouyang, Y., ... & Liu, Y. (2024, August). SwapMoE: Serving off-the-shelf MoE-based large language models with tunable memory budget. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 6710-6720).

Lo, S. C., Zingaro, A., McCullough, J. W., Xue, X., Gonzalez-Martin, P., Joo, B., ... & Coveney, P. V. (2025). A multi-component, multi-physics computational model for solving coupled cardiac electromechanics and vascular haemodynamics. Computer Methods in Applied Mechanics and Engineering, 446, 118185.

Xu, X., Tu, W., & Yang, Y. (2023, June). Selector-enhancer: learning dynamic selection of local and non-local attention operation for speech enhancement. In Proceedings of the AAAI conference on artificial intelligence (Vol. 37, No. 11, pp. 13853-13860).

Zhang, Jiaquan, et al. "Learning global hypothesis space for enhancing synergistic reasoning chain." arXiv preprint arXiv:2602.09794 (2026).

Xu, Xinmeng, Weiping Tu, and Yuhong Yang. "Pcnn: A lightweight parallel conformer neural network for efficient monaural speech enhancement." arXiv preprint arXiv:2307.15251 (2023).

Li, W. W., Ziser, Y., Coavoux, M., & Cohen, S. B. (2023, May). BERT is not the count: Learning to match mathematical statements with proofs. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (pp. 3581-3593).

Xu, X., Tu, W., & Yang, Y. (2024). Adaptive selection of local and non-local attention mechanisms for speech enhancement. Neural Networks, 174, 106236.

Dong, J., Koniusz, P., Chen, J., & Ong, Y. S. (2024, September). Adversarially robust distillation by reducing the student-teacher variance gap. In European Conference on Computer Vision (pp. 92-111). Cham: Springer Nature Switzerland.

Yang, Huan, et al. "Kvshare: An llm service system with efficient and effective multi-tenant kv cache reuse." arXiv preprint arXiv:2503.16525 (2025).

Xue, Xiao, et al. "Fast-Forward Lattice Boltzmann: Learning Kinetic Behaviour with Physics-Informed Neural Operators." arXiv preprint arXiv:2509.22411 (2025).

Yao, S., Guo, J., Li, J., Ou, J., Feng, Y., Hu, J., & Liu, D. (2025). Adversarial hard negative samples for continual relation extraction. Applied Soft Computing, 181, 113365.

Wang J, Fan L, Li B, et al. A Dynamic Factor Gating Architecture with Market Regime Awareness for Stock Return Forecasting[J]. 2026.

Yang, Tianyue, and Xiao Xue. "Meno: Meanflow-enhanced neural operators for dynamical systems." arXiv preprint arXiv:2604.06881 (2026).

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of EMNLP.

Zhang, Y., Carvalho, D., & Freitas, A. (2025, July). Quasi-symbolic Semantic Geometry over Transformer-based Variational AutoEncoder. In Proceedings of the 29th Conference on Computational Natural Language Learning (pp. 12-29).

Wu, Beiliang, et al. "IndexNet: Timestamp and Variable-Aware Modeling for Time Series Forecasting." arXiv preprint arXiv:2509.23813 (2025).

Zhang, W., Zhang, C., Gu, C., Kou, J., Yuan, H., Fang, X., ... & Fang, Y. (2024, October). Hallucination in Large Language Models: From Mechanistic Understanding to Novel Control Frameworks. In 2024 7th International Conference on Universal Village (UV) (pp. 1-36). IEEE.

Tang, Y., Zhang, G., Liu, J. K., & Qin, R. (2025). Weakly supervised land-cover classification of high-resolution images with low-resolution labels through optimized label refinement. International Journal of Remote Sensing, 46(5), 1913-1937.

Zhang, W., Zhang, C., Luo, Z., Ma, J., Yuan, W., Gu, C., & Feng, C. (2025). SemanticForge: Repository-Level Code Generation through Semantic Knowledge Graphs and Constraint Satisfaction. arXiv preprint arXiv:2511.07584.

Dong, J., Koniusz, P., Chen, J., Wang, Z. J., & Ong, Y. S. (2024). Robust distillation via untargeted and targeted intermediate adversarial samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 28432-28442).

Zhu, Guoying, et al. "Enabling MoE on the Edge via Importance-Driven Expert Scheduling." arXiv preprint arXiv:2508.18983 (2025).

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Zhou, J., Shuang, K., Wang, Q., Qian, B., & Guo, J. (2025). Bi-directional feature learning-based approach for zero-shot event argument extraction. Information Processing & Management, 62(5), 104199.

Xue, X., Wang, S., Yao, H. D., Davidson, L., & Coveney, P. V. (2024). Physics informed data-driven near-wall modelling for lattice Boltzmann simulation of high Reynolds number turbulent flows. Communications Physics, 7(1), 338.

Cohen, J., Rosenfeld, E., & Kolter, J. Z. (2019). Certified adversarial robustness via randomized smoothing. In Proceedings of ICML.

Chen, L. (2026). Beyond external constraints: The missing dimension of ai governance. Available at SSRN 6449738.

Vuruma, Sai Krishna Revanth, et al. "Utilizing large language models to identify reddit users considering vaping cessation for digital interventions." arXiv preprint arXiv:2404.17607 (2024).

Li, J., Shuang, K., Guo, J., Shi, Z., & Wang, H. (2023). Enhancing semantic relation classification with shortest dependency path reasoning. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31, 1550-1560.