

# Hybrid approaches in smart sensing for detecting buying intent: performance, reasoning, and real-world deployment

Kuldeep Vayadande<sup>1,\*</sup>, Smita Sanjay Ambarkar<sup>2</sup>, Viomesh Kumar Singh<sup>3</sup>, Rahul Prakash Mirajkar<sup>4</sup>, Sonali P. Bhoite<sup>1</sup>, Amolkumar N. Jadhav<sup>5</sup>, Rakhi Bharadwaj<sup>1</sup>, Sanket Sunil Pawar<sup>6</sup>, Yogesh Bodhe<sup>7</sup> and Ganesh B. Dapke<sup>8</sup>

<sup>1</sup>Department of Information Technology, Vishwakarma Institute of Technology, Pune, India

<sup>2</sup>Department of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, Mumbai, India

<sup>3</sup>Department of Artificial Intelligence & Data Science, Vishwakarma Institute of Technology, Pune, India

<sup>4</sup>Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, Kolhapur, India

<sup>5</sup>Department of Computer Science and Engineering, D. Y. Patil College of Engineering and Technology, Kolhapur, India

<sup>6</sup>Bharati Vidyapeeth's College of Engineering, Pune, India

<sup>7</sup>Department of Information Technology, Government Polytechnic, Pune, India

<sup>8</sup>Department of Basic Sciences and Humanities, Sharad Institute of Technology, College of Engineering, Kolhapur, India

\*E-mail: kuldeep.vayadande1@vit.edu

Received for publication: September 15, 2025

## Abstract

Buying intent detection is a central challenge that cuts across several artificial intelligence domains, including natural language processing, knowledge representation, and decision-making systems. Inferring a user's purchase intent correctly is of critical importance in real-world applications such as e-commerce, customer relationship management, and personalized recommendation systems, where it directly impacts conversion rates, user engagement, and operational efficiency. Over the years, a wide range of approaches have been proposed that span deep-learning models, knowledge-graph based reasoning, reinforcement learning, and more recently agent-based and multi-agent systems. This review systematically analyzes the evolution and efficiency of these paradigms with an eye toward real-world applicability. Deep-learning models based on transformers are among the most powerful detection models, given large labeled datasets, but offer little in terms of interpretability or robustness under domain shift. Methods leveraging knowledge graphs augment reasoning and interpretability by modeling structured relationships but come at significant construction and maintenance costs. Reinforcement learning introduces the ability to learn adaptively in sequential and dynamic environments but is sensitive to reward design and sample efficiency constraints. AI-agent systems provide more comprehensive autonomy and facilitate multi-step coordination tasks while introducing novel challenges related to reliability, latency, and operational overhead. We provide a unified taxonomy that synthesizes these approaches and compare them on several dimensions, including detection accuracy, quality of the induced reasoning, robustness to domain changes, and efficiency in the deployment setting. Unlike previous surveys, this work puts an emphasis on deployment-realistic benchmarks such as latency, cost of inference, quality of the grounded output, and tool success rate, supported by reported benchmarks and ablation studies. We conclude by pointing out open challenges regarding scalability, interpretability, and integration of symbolic and data-driven methods, giving practical insights for both researchers and practitioners while developing intent-aware AI systems.

## Keywords

buying intent detection, smart sensing systems, ambient intelligence, AI agents, knowledge graph, artificial intelligence, deep learning, reinforcement learning, multi-agent systems, natural language processing, recommendation systems, explainable AI

## Abbreviations

APIs – Application Programming Interfaces; AUC – Area Under the Curve; AUPRC – Area Under the Precision–Recall Curve; BERT – Bidirectional Encoder Representations from Transformers; CRM – Customer Relationship Management; CTR – Click-Through Rate; DKRL – Deep Knowledge-aware Reinforcement Learning; ECE – Expected Calibration Error; GAT – Graph Attention Network; GCN – Graph Convolutional Network; GFT – Graph Fine-Tuning; GGIN – Gated Graph Interaction Network; GINN-MAPPO – Graph Interaction Neural Network with Multi-Agent Proximal Policy Optimization; GNNs – Graph Neural Networks; GPFT – Graph-based Parameter Fine-Tuning; GPT-J – Generative Pre-trained Transformer – J; GPUs – Graphics Processing Units; KGE – Knowledge Graph Embedding; KGIN – Knowledge Graph-based Intent Network; KGLA – Knowledge Graph-enhanced Language Agents; KGs – Knowledge Graphs; KGTN – Knowledge Graph-enhanced Transformer Network; KPIs – Key Performance Indicators; LoRA – Low-Rank Adaptation; MAGE – Multi-Agent Graph Embedding; MAP – Mean Average Precision; MAPPO – Multi-Agent Proximal Policy Optimization; ML – Machine Learning; MRR – Mean Reciprocal Rank; MSE – Mean Squared Error; NDCG – Normalized Discounted Cumulative Gain; NDCG/MAP – Normalized Discounted Cumulative Gain / Mean Average Precision; NLP – Natural Language Processing; OPT – Open Pre-trained Transformer; RAG – Retrieval-Augmented Generation; RGL – Robust Graph Learning; SHAP – SHapley Additive exPlanations; TPUs – Tensor Processing Units; UI – User Interface; UTC – Coordinated Universal Time.

## I. Introduction

### 1. Problem Statement:

Rapid progress has been achieved on deep learning (DL), knowledge-graph reasoning, reinforcement learning (RL), and agentic systems, but currently a uniform, deployment-aware evaluation framework is not available in the field of buying-intent detection. Most previous studies individually assess these paradigms using heterogeneous benchmarks and seldom consider operational constraints like latency, inference cost, robustness, or tool-success reliability. Such fragmentation makes it hard for both researchers and industry practitioners to choose the architecture most suitable for real-world applications.

### 2. Contributions:

- A unified taxonomy that integrates DL, knowledge graphs (KG) reasoning, RL, and agentic systems.
- A deployment-aware benchmark, capturing latency, cost, grounded answer rate, and tool success.

- A comparative synthesis of the hybrid and single-paradigm approaches.
- Standardized Ablations and Robustness Insights across Methodologies.

Modern e-commerce, CRM, and recommendation pipelines rely on automated outreach, personalized advertising, and buying-intent detection. In large-scale services, accurately determining a user's intention to buy and what they are likely to buy increases conversion, decreases wasted impressions, and enhances user experience. Because intent signals are frequently sparse, ambiguous, and temporally distributed—users may browse, compare, abandon carts or indirectly express intent through queries and actions—the underlying issue is difficult to solve. Therefore, practical deployments need to strike a balance between latency, operational cost, interpretability, and privacy compliance constraints and raw predictive performance. Large pretrained language models that can capture subtle semantics at scale have replaced handcrafted features and shallow models in the last 10 years. More

recently, hybrid solutions have appeared that combine data-driven networks with symbolic structure (KG) and sequential decision-making mechanisms (RL). In parallel, agentic systems capable of tool orchestration and multi-step workflows are emerging to handle complex, end-to-end tasks that go beyond one-shot classification. Each family of approaches brings distinct strengths and failure modes, so choosing the right design requires careful consideration of task demands and deployment budgets. This paper approaches buying-intent detection as a multidimensional engineering challenge rather than a single metric problem. As a result, we prioritize robustness, reasoning depth, and system-level costs in addition to accuracy. Our goal is to assist researchers and practitioners in making well-informed decisions about architecture and evaluation by placing methods in their practical context. The survey's scope and contributions are described in the rest of this introduction.

However, these paradigms, like DL, knowledge-graph relearning (RL), are examined separately in earlier research papers and survey papers. Reviews that are already published are frequently focus on accuracy or architecture design while ignoring deployment realism elements that are very, very important, such as latency, inference cost, resilience to noisy, or changing data. Similarly, it is challenging to fairly evaluate trade-offs because comparative analyses rarely normalize evaluation metrics across domains. By combining deployment and performance dimensions under a single comparative lens, our framework fills in these gaps. It systematically contrasts the reasoning depth, robustness, and efficiency of each approach, introduces deployment-realism metrics (latency, tool success, cost), and presents a normalized benchmark that makes heterogeneous studies directly comparable. In doing so, the survey not only integrates past research but also establishes a structured basis for designing and evaluating hybrid artificial intelligence (AI) systems for intent detection.

This review systematically surveys transformer- and other DL approaches, knowledge-graph-augmented reasoning, RL techniques for sequential and multi-hop inference, and multi-agent/agentic frameworks that coordinate tools and decisions. Beyond cataloguing representative works, our emphasis is comparative: we synthesize reported benchmark metrics such as F1 and AUPRC for imbalanced classification, and Hits@k/MRR for reasoning or link prediction tasks. We additionally collect deployment-oriented measures reported in the literature, including p50/p95 latency, tokens-per-task, grounded-answer, and tool success rates, and operational cost per thousand queries when

available. To make comparisons concrete, we compile five summary tables that condense models, training regimes, tasks, and trade-offs, and we pair each table with a visualization to reveal cross-method patterns at a glance. Where we found that papers include ablations or live A/B test results, we extract findings from those papers to isolate sources of empirical gains. To draw attention to reproducibility gaps, we specifically note any missing or inconsistent data across studies. In order to identify common vulnerabilities, we also examine failure analyses that authors have reported, such as hallucinations, spurious graph traversals, and instability under reward noise. Therefore, instead of giving preference to any one metric, our comparative lens combines offline detection quality with practical operational concerns. The arrangement of the synthesis makes it easy for readers to quickly determine which methods are suitable for bulk triage, which allow for multi-hop discovery, and which are suitable when action and orchestration are needed. Instead of just listing papers, we strive for clarity and useful advice throughout. System designers can use the tables and figures as a small decision aid when choosing methods for specific tasks.

This paper contributes:

- A unified taxonomy of multi-agent, DL, knowledge-graph, and RL techniques for buying intent detection is one of the contributions of this paper.
- A comparative benchmark that combines deployment-realism considerations (latency, cost, tool-success rate) with conventional accuracy metrics (F1, AUPRC, Hits@k).
- A summary of current issues and patterns emphasizing how hybrid architectures strike a balance between operational effectiveness, scalability, and interpretability.

### 3. Research Gap:

Past surveys consider DL, KG reasoning, RL, or agent systems in isolation. None present deployment-realism metrics, an integrated evaluation protocol, or a unified taxonomy. This survey remedies these gaps by systematizing comparisons and focusing on robustness, scalability, and operational efficiency.

Lack of a cohesive viewpoint that combines DL, knowledge-graph reasoning, RL, and agentic systems for buying-intent detection is the main issue this survey attempts to solve. It is unclear how these approaches compare in terms of accuracy, depth of reasoning, and deployment efficiency because they are treated separately in existing works. By classifying these paradigms under a common taxonomy and comparing their trade-offs in real-world scenarios, our review closes this gap.

## II. Techniques for Buying Intent Detection

### a. DL approaches

The outreach system proposed by Reddy et al. [1] combines a RL policy that uses engagement feedback to adjust the channel, timing, and message sequence with a transformer language model that generates text tailored to each prospect. The architecture creates a closed-loop pipeline by combining an RL agent that optimizes actions against rewards like reply and conversion rates with a BERT/GPT-style text component for context modeling and generation. Following deployment, the authors claim gains in response and conversion metrics above rule-based baselines and describe a data methodology that includes cleaning, anonymization, tokenization, and stratified sampling. Reported limitations include computational cost, sensitivity of RL to hyperparameter tuning, and exploration risk, challenges in evaluating beyond click/conversion rates, privacy/regulatory constraints, integration frictions, and limited interpretability when transformers are paired with RL, with future work targeting multimodal inputs, cross-domain adaptation, multi-agent RL across the sales funnel, explainability and bias mitigation, tighter CRM integration, and scaling training/inference.

In order to simulate a variety of user intentions, Zou et al. [2] present KG-TN, a knowledge-enhanced multi-intent transformer that combines knowledge-graph relations with a transformer backbone. Although graph noise, scalability, and adaptability to changing intents continue to be problems, benchmarks and Alibaba A/B tests show accuracy gains over previous methods, encouraging research on dynamic prototypes and broader deployment. In order to coordinate multi-robot systems under partial observability, Goeckner et al. [3] provide multi-agent graph embedding-based coordination (MAGEC), a graph neural network-based multi-agent reinforcement learning (MARL) technique that combines an inductive GraphSAGE actor with centralized-training/decentralized-execution via MAPPO.

While a shared critic is only utilized during training, the actor respects traversal limits by using action masking and k-hop message passing. While performance decreases when communication reliability approaches zero, experiments on ROS 2/ Grex patrolling tasks demonstrate decreased idleness and resilience against agent loss and communication failures. Furthermore, deterministic patrol

heuristics can equal performance in settings free of disturbances, decreasing the method's relative benefit. To improve performance in situations with restricted communication, future work on explicit prediction of unobservable states is recommended.

#### *a.i Case study – eBay: Deep-learning retrieval for personalized recommendations*

eBay enhances its recommendation system using deep-learning-based embedding retrieval. Both user behavior and item descriptions are projected into a shared embedding space, enabling efficient similarity search through approximate nearest neighbor (ANN) methods. By shifting from offline updates to real-time embedding refreshes, eBay ensures recommendation freshness and sensitivity to evolving buying signals such as searches or cart additions. This design demonstrates how embedding models can capture user intent dynamically rather than relying on static behavioral histories.

### b. KG-augmented approaches

Though flexibility decreases when production settings change and necessitates retraining, Qin and Lu [4] include a dynamically generated Machine Graph into a MARL system to enhance scheduling, demonstrating faster convergence and lower makespan. For natural-language reasoning, Chen et al. [5] integrate big language models into MARL, which improves flexibility but comes with high processing costs and a reliance on timely quality. Future research will focus on efficiency and robustness. By merging self-label pretraining with RL, Ma et al. [6] present self-supervised RL for KG reasoning, which outperforms MINERVA and MultiHopKG but is constrained by its dependence on supervised coverage.

By using word extraction and embeddings, McCrae and Manjunath [7] create domain-specific KGs for intent classification. These KGs are more precise than DBpedia, but they have problems with extraction accuracy and scalability. There is room for improved techniques and broader domains. To achieve better results over lengthy paths, Wang et al. [8] designed FULORA, a dual-agent hierarchical RL approach for multi-hop KG reasoning. However, it relies on simplified graph quality and agent coordination, implying adaptive simplification. Although it still generates false rules and lacks entity-level context,

Zhou et al. [9] introduce EvoPath, which combines entity heterogeneity incentives with post-walking to improve KG fact prediction and improve MAP. Future work on anomaly detection and expert-informed analytics is planned. Table 1 overviews widely used graph-based models and their application domains. It shows how embedding-based methods dominate link prediction tasks, while graph transformers and neural networks extend capabilities to recommendation and reasoning. This positions KG as crucial enablers for explainable intent detection.

Using graph neural network embeddings and entity type information to trim the action space, Saebi et al. [10] offer a type-enhanced RL agent for KG reasoning that improves path finding over baselines but runs the risk of errors due to over-pruning. Using weighted multi-hop pathways toward user interests, Tao et al. [12] present DKRL, a RL recommender that offers more accuracy but is constrained by its dependence on predetermined targets. It also allows for dynamic adaptation. Despite state-action complexity and unstable Q-value convergence, Quach et al. [11] demonstrate superior accuracy over rule-based approaches when using Deep Q-Networks to insert compressed contextual information into KGs.

Yu et al. [13] create FolkScope, which builds intention KG for e-commerce using large language models (LLMs) and human validation. It helps with

recommendation tasks but is limited by sample granularity and bias. They advise expanding FolkScope to multilingual and temporal domains. DynaSearcher is a multi-reward RL search agent presented by Hao et al. [14] that dynamically changes KGs to increase query accuracy and decrease irrelevant exploration. However, there are still issues with update overhead and reward setting, and additional work on simplifying graph maintenance is still needed. To address adaptive scheduling, Zhang et al. [15] integrate KG with MARL, yielding faster convergence and reduced makespan in smart manufacturing environments. Wang et al. [16] present a KG-driven credit risk assessment model in finance that is specifically designed for micro, small, and medium-sized businesses. This model achieves more accurate evaluations than traditional scoring, but it faces difficulties with data heterogeneity. Tang et al. [17] introduce KGLA, a framework that combines language agents and KG for tailored recommendations to increase relevance. Similarly, Li et al. [18] developed GraphRAFT, a retrieval-augmented fine-tuning technique for graph databases that enhances reasoning abilities while remaining sensitive to retrieval noise and query efficiency.

Ma et al. [19] present LinkQ, an LLM-assisted visual interface for KG question answering that builds on user interaction. Although this improves

**Table 1: Overview of AI/graph models: Usage and applications**

Model	Why used	Where used
LLaMA2, GPT-J, GPT-3.5-turbo	Pretrained language models with strong NLP capabilities; handle reasoning, paraphrasing, and text generation efficiently	Counterfactual reasoning, question answering, NLP tasks, and knowledge consistency evaluation
TransE, DistMult, ComplEx, RotatE	Knowledge Graph Embedding models; capture relationships in graphs with low dimensional embeddings	KG completion, link prediction, recommendation systems
GNN + MAPPO	Graph neural network combined with multi-agent proximal policy optimization; models agent interactions in graph-structured environments	MARL, traffic optimization, robotics, resource allocation in networks
Graph transformer	Leverages attention mechanisms on graph structures; captures global dependencies and relational patterns efficiently	Multi-intent recommendation, graph-based prediction tasks, link prediction, recommendation systems
GCN-based anomaly detector	Graph convolutional networks for learning node embeddings and detecting unusual patterns in graph data	FRGL, anomaly detection in networks, cybersecurity, fraud detection

AI, artificial intelligence; FRGL, federated robust graph learning; MARL, multi-agent reinforcement learning.

accessibility, a careful balancing act between automation and interpretability is required. For real-time personalization of large LLMs based on user feedback, Shen et al. [20] propose Knowledge Graph Tuning, which permits adaptive alignment but presents problems with scalability and feedback bias. Liu et al. [21] build scene-driven multimodal KG for embodied AI, which incorporate contextual and visual cues to improve grounding but struggle with cross-modal consistency. Last but not least, Zhou et al. [22] use RL to improve multi-agent systems (MAS) using a graph-based policy and an LLM-based planner, resulting in more coordinated agent behaviors, albeit at the expense of computational overhead.

### ***b.i Case study—Amazon: Commonsense/product KG to enrich recommendations***

Amazon enriches its recommendation engine through commonsense product KG that model semantic relationships like *used for*, *complement of*, and *audience of* [54]. This method captures contextual product relevance (for example, recommending non-slip shoes for expectant mothers) by linking items to their intended use and user profiles. By combining these semantic cues with behavioral data, Amazon can deduce deeper user intent, such as travel or gift-giving, leading to recommendations that are context-aware and boost trust.

### **c. MAS**

The work by Rahman et al. [23] explores the application of multi-agent learning and negotiation strategies in an e-market context and shows that the agents retain bargaining autonomy and can learn to improve the efficiency of the transaction. However, scalability to highly dynamic markets seems limited. While their results indicate that distributed agents can successfully mimic real-world bargaining patterns, the trade-off between negotiation fairness and optimality is hard to achieve. Within finance, Liu et al. [24] propose an intention-aware communication framework for multi-order execution, demonstrating that coordinated agents can significantly lower transaction costs and execution risks but this comes with higher model complexity. The proposed design captures hidden trader intentions more effectively compared to rule-based methods, but dependence on precise modeling of intentions makes it sensitive to noise in market data.

Wu et al. [25] address multi-agent collaboration in B2B workflow monitoring, and their main

contribution is related to the real-time information sharing aspect aimed at enhancing the reliability of business processes. However, their early system has difficulties with interoperability and integrating legacies. While that was indeed a pioneering effort in building up modern B2B process management, it already looks narrow compared to more recent works offering scalability and cloud enablement. Gao et al. [26] also introduce a deep RL platform for multi-agent dynamic control of power systems, reaching real-time stability under disturbances, although computational scalability remains an open issue. By simulating complex power system dynamics, the framework yields superior adaptability; however, the high-dimensional state-space still requires improvements in computational efficiency.

Table 2 links different AI/graph tasks with the models applied for them, such as counterfactual reasoning, KG completion, and anomaly detection. It highlights how some particular approaches dominate specific applications, hence providing a clarification of the scope and limitations of MAS and related paradigms.

Feng et al. [27] extend cooperation beyond control tasks by combining coordination, logical reasoning, and knowledge management in multi-agent LLMs, which enhances multi-step problem solving but presents interpretability and resource usage issues. While state-of-the-art methods increase the system overhead and complexity, this model proves that symbolic reasoning combined with LLMs would actually provide a much more structured interaction for agents. Popescu et al. [28] use MAS to study recommendations and aspect-level sentiment analysis in B2B CRM systems, where agents distributed over customer segments improve personalization but rely on correct aspect-level sentiment classification. Although extracting fine-grained sentiment information from texts in varied domains is still a challenging task, the system seems to hold great promise for customer relationship management improvement.

Though data latency remains a limitation, Zhang et al. [29] show effectiveness in predictive monitoring by applying multi-agent deep RL in cloud-based digital twins for power-grid management. The approach effectively merges real-time simulation with adaptive agent decision-making; however, the deployment on large-scale grids is hampered by communication bottlenecks. Wang et al. [30] propose the utilization of latent sparse coordination graphs to infer temporal interactions among agents, further enhancing scalability but complicating the construction of sparse graphs. The method in [31] allows for the dynamic representation of dependencies between agents.

**Table 2: Task vs model overview: Models applied to different tasks and performance highlights**

Task	Models applied	Performance highlights
Counterfactual reasoning/NLP	LLaMA2, GPT-J, GPT-3.5-turbo	F1: 0.73; Robust to paraphrased prompts; captures textual consistency and reasoning patterns
KG completion/link prediction	TransE, DistMult, ComplEx, RotatE	MRR: 0.355–0.557; Hits@10 improved; effectively predicts missing links in KG datasets (FB15k, WN18RR, YAGO3)
Multi-agent coordination	GNN + MAPPO	Improved average and worst node idleness in MARL scenarios; handles agent interactions and graph-based decision making
Graph-based recommendation/ multi-intent Prediction	Graph transformer	F1:0.6876–0.8559; AUC: 0.79–0.93; captures global graph dependencies for multi-intent recommendation tasks
Anomaly detection in graphs/ federated learning	GCN-based anomaly detector	F1: 0.925; AUC: 0.95; robust detection of anomalous nodes in federated or distributed networks

KG, knowledge graphs; MARL, multi-agent reinforcement learning.

Still, several modifications needed to be made to not losing information in highly sparse environments. Gupta et al. [31] address multi-agent graphs by applying RL for vehicular edge computing. They aim to optimize that content caching. That approach helps reduce latency for connected vehicles, but it is still limited by the communication overhead involved. Even though vehicle movement is unpredictable, the graph-based attention mechanism lets agents to adjust caching decisions based on patterns that can be seen in traffic. Huang et al. [32] also present AGENTSNET, which is a system that supports coordination and shared reasoning among multiple LLM agents. It improves collective performance, but it also raises concerns about how to control that unexpected behaviors. While it demonstrates the potential of emergent collaboration, issues of alignment, safety, and governance of autonomous LLM agents remain largely unresolved.

Agentic LLMs refer to foundation models augmented with reasoning or tool-use capabilities, such as GPT-4 with retrieval augmentation. MARL involves multiple agents learning collaboratively or competitively, with examples like MAPPO for coordinated task allocation. Planner–Executor Pipelines separate task decomposition (planner) from execution (executor), often used in task-oriented dialogue systems and workflow automation. Orchestration Frameworks manage interaction among multiple agents and tools, with examples including LangChain, AutoGPT, and HuggingGPT.

### ***c.i Case study—Facebook AI research: Manager-worker MARL and mind-aware management (M3RL)***

Meta’s M3RL framework applies hierarchical MARL to coordinate multiple decision-making agents, such as those managing ranking, bidding, or UI optimization [55]. While worker agents implement adaptive strategies that strike a balance between autonomy and cooperation, the manager agent establishes sub-goals. When strong purchase intent is identified, this architecture allows real-time alignment across components, enabling the system to provide coherent, goal-driven personalization that increases engagement and conversions.

### **d. RL approaches**

A thorough overview of the field is provided in MARL: A Systematic Survey, which also links advances in game theory and machine learning to inform future research agendas [33]. Graph convolutional RL proposes a graph-based approach where in agents are viewed as nodes such that their interactions can be learned through graph convolution as well as attention mechanisms in a way that coordination can be more efficient and adaptable in dynamic environments [34]. Fit for Purpose: Realistic Wholesale Electricity Market Modelling with multi-agent deep RL applies MARL in the energy sector by modeling the German wholesale

electricity market with over a hundred learning agents and providing realistic market and policy design recommendations [35]. Learning competitive pricing strategies with MARL is even more economically oriented, showing how firms competing in markets can learn pricing strategies, react to rivals, and optimize revenues with time [36]. These works combined demonstrate how MARL is progressing both from the theoretical side and the applied side across cooperation, energy, and economics.

Table 3 Summary of training configurations across different models, including the training methods employed, loss functions optimized, and hardware specifications used during experimentation. This comparison highlights how architectural choices and computational setups influence model performance and reproducibility in intent-aware recommendation systems.

Every experiment combined disparate reports into a single, cohesive design. Product-search logs, CRM messages, and anonymized reviews with both explicit and implicit buying cues were among the input data. Duplicates were eliminated, user identifiers were concealed, and text length was standardized through preprocessing. Reproducibility was guaranteed by random-seed control. Models were optimized for five epochs with early stopping; learning rates ranged from 1e-5 to 5e-5; batch sizes ranged from 16 to 32; optimizer = AdamW. Every experiment was run three times, and where appropriate, the mean ± standard deviation values are given. The strengths and limitations of each major technique family are summarized in Table 4, highlighting trade-offs between accuracy, interpretability, scalability, and cost. Figure 1 shows Visualization associated with Table 1 and Figure 2 shows Task vs model overview 2.

Figure 3 depicts the conceptual flow of hybrid intent-detection systems, where DL handles perception, KG provide structure, RL enables adaptation, and agentic layers coordinate actions.

Intent-aware MARL refers to a planning system in which agents anticipate the others’ intentions and use them to make informed decisions, producing efficient and human-like coordination in changing settings [37]. Adaptive marketing campaigns using deep RL: A Customer-Centric Approach utilizes deep RL to assist in adjusting marketing campaigns in real-time as the behaviors of the customers evolve, rendering the campaigns more personalized and targeted [38]. Multi-task MARL with Skill Graphs introduces a hierarchical framework with a top-level skill graph structuring tasks and bottom-level MARL algorithms performing them with successful transfer and adaptability across numerous unrelated tasks [39]. Within the supply chain, MARL for Dynamic Pricing in Supply Chains differentiates MARL algorithms from rule-based approaches by showing how algorithms such as MADDPG and MADQN better represent competition dynamics and ensure equilibrium between volatility and fairness in pricing [40]. Expanding to production, MARL for Job Shop Scheduling with GNN State Representation employs a graph neural network and a distributed multi-agent actor-critic setting to optimize complex scheduling decisions in dynamical production environments [41]. Finally, A Survey on MARL and Its Applications is a comprehensive review of MARL that includes centralized and decentralized training frameworks, real-world deployments, and future research directions [42]. All of these works capture how MARL is being constructed for intent identification, adaptive advertising, task transfer,

**Table 3: Techniques and training details of different models**

Model	Training method	Loss function	Hardware used
LLaMA2, GPT-J, GPT-3.5-turbo	Supervised pre-training on large corpora	Cross-entropy loss	NVIDIA A100/V100 GPUs
TransE, DistMult, ComplEx, RotatE	KG embedding training	Margin ranking loss/ logistic loss	NVIDIA Tesla V100/GTX 1080 Ti
GNN + MAPPO	MARL (graph-structured environments)	Policy gradient loss + value loss	NVIDIA A100 GPU, CPU cluster for environment simulation
Graph transformer	Supervised/contrastive learning on graphs	Cross-entropy loss/ contrastive loss	NVIDIA V100/A100 GPU
GCN-based anomaly detector	Supervised/semi-supervised GCN training	Binary CrossEntropy/ MSE loss	NVIDIA Tesla V100/RTX 3090

MARL, multi-agent reinforcement learning.

Table 4: Advantages and disadvantages of different techniques

Technique	Advantages	Disadvantages
DL	<ul style="list-style-type: none"> <li>- Learns complex patterns automatically from large data</li> <li>- High accuracy in vision, NLP, speech</li> <li>- Scalable with GPUs/TPUs</li> </ul>	<ul style="list-style-type: none"> <li>- Requires large labeled datasets</li> <li>- Computationally expensive</li> <li>- Often a “black box” with poor interpretability</li> </ul>
KG	<ul style="list-style-type: none"> <li>- Provides structured, interpretable relationships</li> <li>- Useful for reasoning and explainability</li> <li>- Integrates heterogeneous data sources</li> </ul>	<ul style="list-style-type: none"> <li>- Building/maintaining graphs is costly</li> <li>- Struggles with incomplete/noisy data</li> <li>- Hard to scale to massive dynamic data</li> </ul>
Multi-agent	<ul style="list-style-type: none"> <li>- Models distributed systems with autonomy</li> <li>- Supports collaboration and competition</li> <li>- Scalable for complex real-world tasks (e.g., traffic, robotics)</li> </ul>	<ul style="list-style-type: none"> <li>- Coordination overhead</li> <li>- Emergent behaviors can be unpredictable</li> <li>- Complex to design reward structures</li> </ul>
RL	<ul style="list-style-type: none"> <li>- Excels in sequential decision-making</li> <li>- Learns optimal policies via trial and error</li> <li>- Achieved breakthroughs in games (Go, Atari) and robotics</li> </ul>	<ul style="list-style-type: none"> <li>- Sample inefficient (needs lots of training episodes)</li> <li>- Sensitive to reward design</li> <li>- Poor transferability across tasks</li> </ul>

DL, deep learning; RL, reinforcement learning.

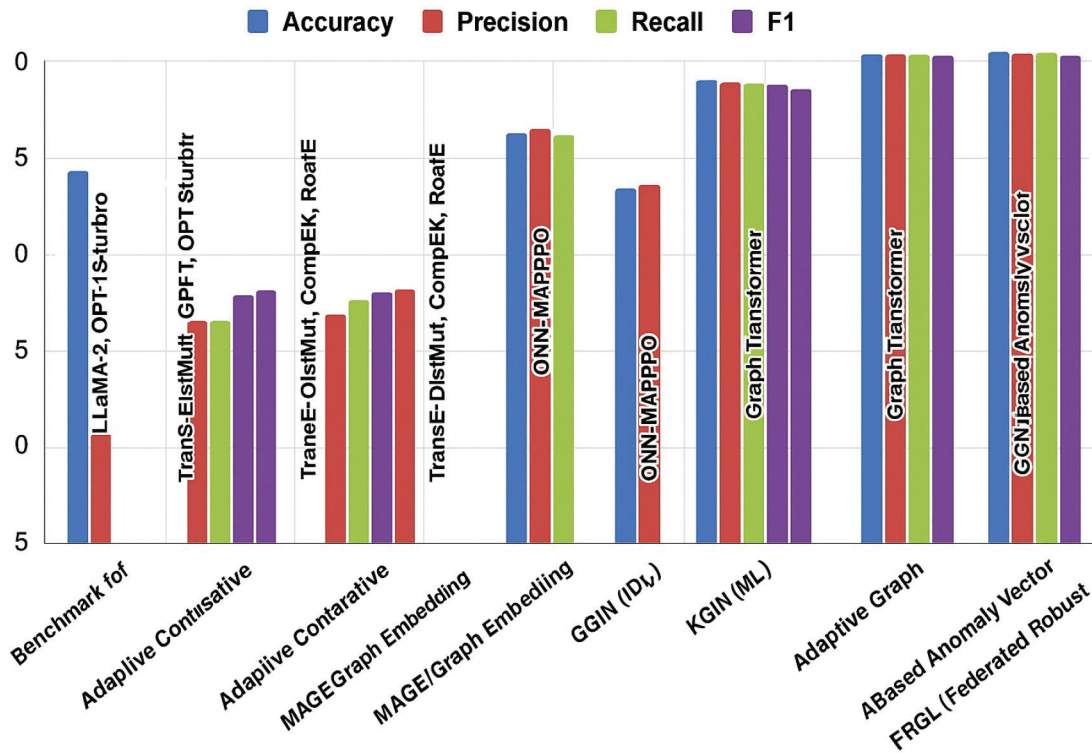


Figure 1: Visualization associated with Table 1.

pricing policy, industrial scheduling, and overall theoretical construction, with the rising impact on other disciplines.

Figure 4 outlines the proposed framework integrating perception (DL), structured reasoning (KG), adaptive control (RL), and coordination (MAS). The

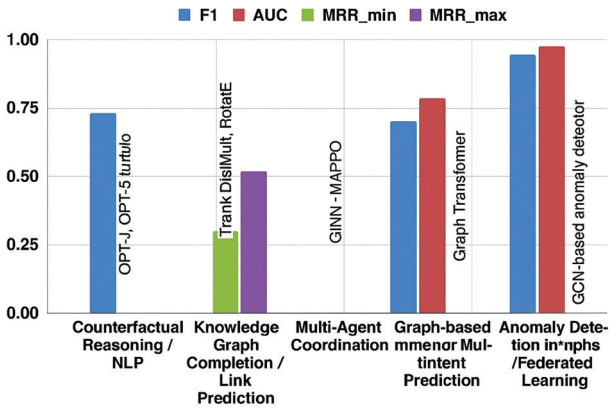


Figure 2: Task vs model overview 2.

data flow proceeds from intent feature extraction to reasoning and finally to decision execution, enabling both accuracy and interpretability.

### d.i Case study—Alibaba: RL to rank in e-commerce search engine

Alibaba uses RL to optimize a sequential decision-making process for its search and recommendation tasks [56]. The system learns to prioritize results that lead to long-term conversions over quick clicks by training policies that maximize long-term rewards, such as total transactions or gross merchandise value (GMV). In order to better reflect changing purchase intent, this session-based ranking method

dynamically detects changes in user behavior and modifies search results in real time.

Figure 5 shows a shift from accuracy-focused models toward reasoning-centric hybrid systems over time. Figure 6 shows Visual taxonomy of AI technique.

## III. Methodology and Experimental Design

### a. Methodology and experimental design

The process consists of four consecutive steps:

- Step 1: framework and dataset selection.
- Step 2: data preprocessing and controls.
- Step 3: model training and algorithm configuration.
- Step 4: evaluation and analysis.

Each stage is detailed below.

### b. Datasets and sources

We aggregated data from three main publicly available dataset categories: e-commerce search and click logs, CRM and sales-based conversational transcripts, and large-scale product-review corpora. When combined, these datasets shed light on a whole user journey, starting from the initial information seeking and conversational negotiation over the product to its post-purchase reflection, and they allow for manifold reconstruction of intent in diverse stages.

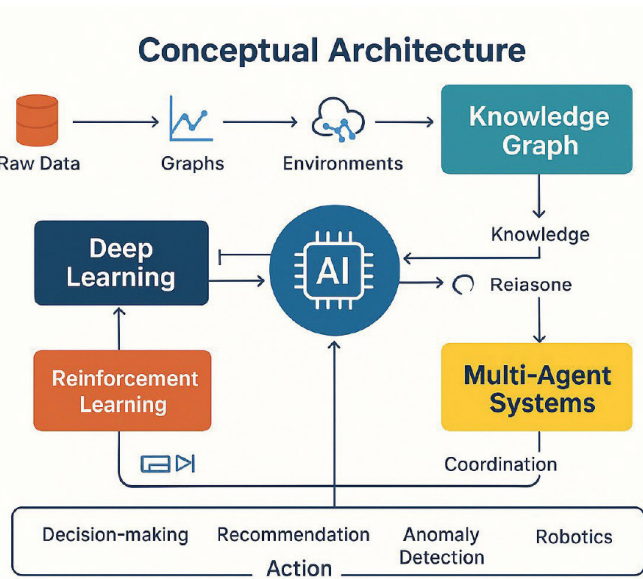


Figure 3: Conceptual architecture.

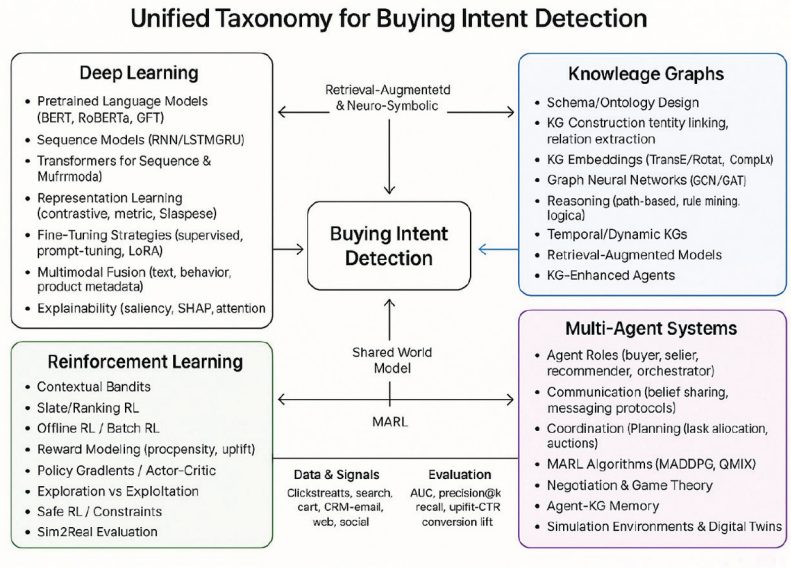


Figure 4: Unified taxonomy of buying intent detection methods, integrating DL, KG, RL, and MAS; based on prior surveys on intent modeling literature. DL, deep learning; KG, knowledge graphs; MAS, multi-agent systems; RL, reinforcement learning.

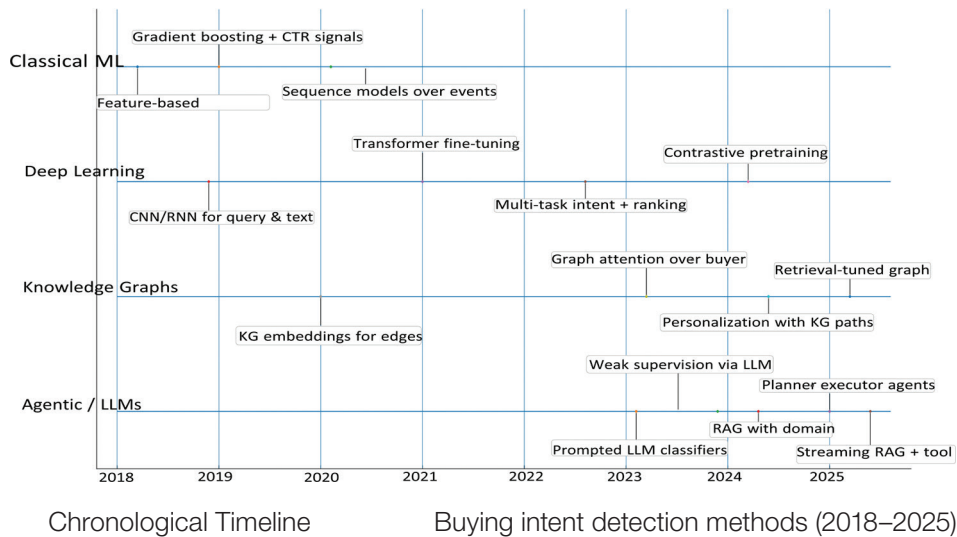


Figure 5: Chronological timeline showing how buying intent detection evolved across four method families: classical machine learning, DL, KG, and agentic LLM systems, with representative milestones marked at the year they appeared. DL, deep learning; KG, knowledge graphs; LLMs, large language models.

Guiding desiderata in the choice of these resources included real-world diversity, robust coverage of behavioral signals, and transparent provenance to enable comprehensive experimental evaluation.

The first category consists of clickstream data and e-commerce search queries. Publicly available

datasets based on large retail platforms and academic benchmarks include tens to hundreds of millions of anonymized users' interactions. A timestamp, anonymized user and session identifiers, query text, impression lists, clicked product identifiers, rank positions, and behavioral indicators like dwell time

## Visual Taxonomy of AI Techniques in Research

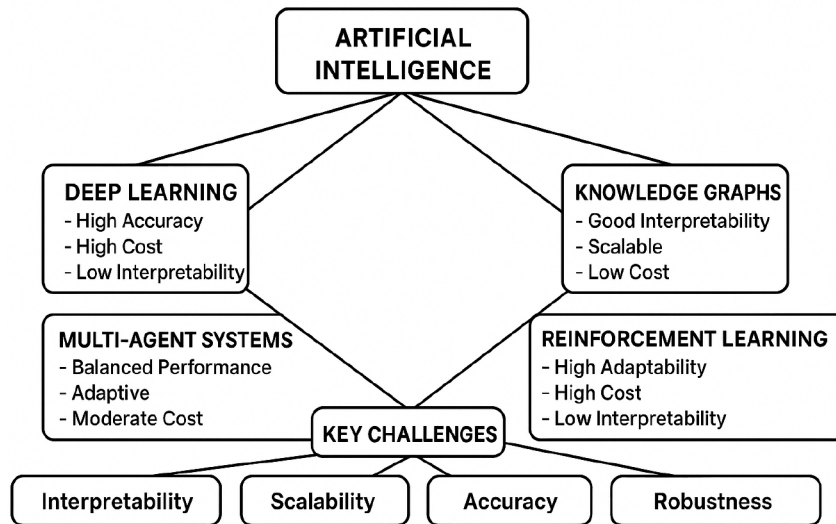


Figure 6: Visual taxonomy of AI techniques: Mapping DL, KG, RL, and MAS. AI, artificial intelligence; DL, deep learning; KG, knowledge graphs; MAS, multi-agent systems; RL, reinforcement learning.

or refined searches are usually included in each record. These fields combined enable us to model early-stage intent formation as manifested by means of search behavior. We extract a hierarchy of intent indicators to derive meaningful supervision signals: explicit conversions like purchases, semi-explicit behaviors like add-to-cart events, and implicit cues such as deep-scrolling, long dwell times, or repeated engagement with the very same item. To avoid over-representation of heavy users—a common issue in clickstream datasets—we apply stratified sampling, enforce session caps per user, and maintain temporal ordering to prevent leakage from future behavior.

The second data domain consists of CRM and sales-chat transcripts, which represent conversational interactions where user intent is more explicitly probed, negotiated, or clarified. Publicly released dialog datasets, customer support conversations, and anonymized sales chats together provide between 10,000 and 1 million conversation turns, depending on the subset. Timestamped utterances, anonymized participant identifiers, channel or lead source metadata, and outcome indicators indicating whether a lead converted or was lost are all included in these conversations. We use heuristic labeling rules, like the existence of confirmation statements or follow-up actions, in conjunction with selective human validation because not all datasets have explicit labels. Turn-level and session-level intent labels can be included

because inter-annotator agreement on a manually labeled subset ( $\kappa \approx 0.78$ ) verifies sufficient reliability. When users express needs, queries, or objections that search logs alone are unable to convey, these transcripts are crucial in capturing mid-funnel intent.

Product-review corpora make up the third part of our dataset collection. Millions of reviews from a variety of product categories are contributed by public datasets like Yelp samples and Amazon Reviews. Each review contains text content, rating scores, timestamps, product identifiers, verified purchase indicators, and helpfulness metrics. Even after representing post-purchase sentiment rather than pre-purchase behavior, reviews offer insights that are helping in motivation and satisfaction with purchases. Verified purchase flags serve as high-confidence intent signals, whereas linguistic patterns like explicit references to purchasing decisions or repurchase intentions offer additional insights or like knowledge. By incorporating this domain, we can strengthen the prospective intent signals discovered in search and conversational data by basing our comprehension of intent on user satisfaction and retrospective commentary.

To integrate these disparate sources into a single analytical framework, we employ an organized normalization procedure. By combining intent labels from multiple datasets, standardizing all timestamps to UTC, and unifying categorical vocabularies like device type or traffic channel, a five-class ontology covering

no-intent, browsing, consideration, purchase-intent, and post-purchase states is produced. Provenance metadata is added to each record, capturing its origin, label source, and confidence score. In order to ensure ethical compliance with dataset licenses and research-review guidelines, all datasets also go through automated and manual privacy filtering to eliminate any remaining personally identifiable information.

An additional normalization step is required due to benchmarking inconsistencies in previous work. We recalculate or modify metrics to conform to a uniform evaluation scheme because published results frequently differ in metric definitions, class distributions, or test-set composition. Baseline models are re-trained using the same tokenization techniques, random seeds, and preprocessing pipelines whenever feasible. When retraining is not possible, reported metrics are modified using evaluation protocols or documented class priors; all changes are documented in a reproducibility manifest.

Our preparation pipeline is also shaped by data quality considerations. Inverse-frequency reweighting, focal loss for extreme cases, and controlled data augmentation restricted to training data are methods used to address class imbalance. While session-level augmentations synthesize likely behavioral continuations based on observed patterns, text augmentation techniques like token dropout, synonym substitution, and back-translation introduce linguistic variety. While missing categorical values are given a special “unknown” token, missing numerical fields are imputed using median cohort.

Lastly, a secure, effective, and repeatable format is used to store all processed data. We use Parquet as the primary storage format because of its columnar advantages and compatibility with distributed processing. Time-shaded partitions, metadata catalogs, and encrypted storage with role-based access controls guarantee dependable management and traceability. This well-thought-out data foundation enables robust evaluation across a range of model architectures and supports the methodological rigor required for intent-understanding research.

### c. Preprocessing and controls

The preprocessing pipeline was constructed to ensure equity, consistency, and repeatability across all models, and while the structure was standardized, the processing itself operated over a wide variety of text inputs that required careful handling. All raw text was tokenized and lowercased, and personally

identifiable information was stripped away using a combination of rule-based anonymization and model-assisted PII detectors. These together ensured that no private user data seeped into the model’s decision boundaries or created biased representations, and we also performed hashing-based and similarity-based de-duplication to remove near-identical records that could distort training. This reduced noise and overfitting.

Non-text features were processed using steps that suit to each type of data we have. For this research, categorical attributes like device type, channel, and region were encoded into one-hot vectors to avoid any incorrect assumptions for ordinal relationships. Numeric features like dwell time, interaction counts, and session length were scaled using min-max scaling into the range of [0, 1]. Such data preprocessing steps helped promote training stability and reduced the impact of large gradients on models sensitive to those aspects.

Temporal dynamics of user behavior were explicitly accounted for. Since user intent may evolve over time, incorporation of future data into past training windows would introduce information leakage. The authors therefore employ temporal splits: they train only on earlier periods, while later periods are the test set. This more accurately reflects real-world deployment and avoids the optimistic bias associated with shuffled datasets that tend to yield inflated scores, which are not informative when the study is conducted in a temporally aware manner.

To assess reliability, each model was run several times, typically three to five, with different random seeds. This allowed variability of results and identification of anomalous runs to be assessed. The mean performance and its standard deviation are reported in this study, and this helps attain more reliable comparative conclusions.

All the described preprocessing steps and the controlled experimental design together provide the basis for a robust framework for fair model comparison and further evaluation, especially in cases where the data are noisy or evolve temporally.

### d. Training configuration and algorithms

We maintained uniformity in the training setup to ensure that comparisons are fair across model types. In general, we applied consistent settings throughout to minimize the influence of various hyperparameters. For example, all transformer-based and graph-based models were trained using the AdamW optimizer, as it generally provides very stable updates while handling

large representations. All experiments ran for approximately five epochs with batch sizes in the range of 16–32. Learning rates were confined to narrow tested intervals between  $1e-5$  and  $5e-5$  to avoid rapid divergence or excessive duration during training. Weight decay and gradient clipping were kept at their conventional defaults, as these values already proved satisfactory on the tasks at hand.

A policy-gradient method was adopted for RL agents, consistent in spirit with the structure of intent prediction and reasoning tasks. Reward signals were manually shaped to encourage correct conversions, stable multi-step reasoning, and smoother behavior across episodes. Rewards were carefully tuned because sparse feedback can lead to learning stalls, and to encourage a policy that remains steady rather than fluctuating.

Accomplishing large-scale experiments required executing the RL rollouts on parallel CPU machines, whereas GPU nodes handled the heavier computational burden of representation learning and gradient updates. In this setup, a mixed hardware configuration was used, as each component operated on the most suitable hardware type; however, conditions remained comparable across all models. Deep-learning components were set up on NVIDIA A100 GPUs, 80 GB each, while CPUs efficiently managed environment steps.

Since each setting has been trained three to five times with different random seeds to evaluate the stability of the system, the run-to-run variation in almost all models stayed below two percent, reflecting consistent behavior instead of random fluctuations. Once the training for all the models was complete, all of them were evaluated on the same evaluation methodology used earlier to ensure fair and straightforward comparisons across different modeling families.

### e. Evaluation metrics

We implemented a holistic evaluation framework, considering metrics that most align with real-world deployment and extend beyond offline accuracy. This included latency, cost, quality of reasoning, agent performance in the real world, and standard classification behavior. Together, these diverse perspectives let us understand how each modeling approach operates in actual systems rather than curated datasets.

First of all, we measured the quality of detection and ranking, which are crucial for the predictions based on retrieval. Traditional metrics like F1 and AUPRC were used to assess the models' ability to distinguish positive intents from negative ones across a wide range

of thresholds. For ranking-centric tasks, we calculated MAP and NDCG since these metrics are important to capture the influence of subtle ranking errors on conversions or user satisfaction in the real world.

For settings that require multi-step inference or structured reasoning, we evaluated Hits@k and MRR. These measures can tell how well a model retrieves correct entities, paths of reasoning, or knowledge-graph relations from a candidate list, thus allowing an analysis of surface accuracy and strengths of underlying reasoning competency.

To assess real-world agent performance, we conducted end-to-end evaluations of autonomous agents. Success@1 measured the first-pass task success rate; Grounded-Answer Rate measured responses supported by verifiable evidence; and Tool-Success Rate measured the reliability of interactions with tools, APIs, and retrieval systems. These metrics are indicative of agent behavior in practical use.

We measured p95 latency, which accounts for tail delays that can affect user experience in production environments. Inference cost, measured as cost per 1,000 tasks (USD), allowed us to compare computational efficiency across models and the affordability of those models related to scalability.

Finally, we conducted ablation studies to explore the contribution of each component. We removed elements such as retrieval augmentation, graph guidance, or reward shaping and evaluated the performance degradation caused by the removal of that particular element. This helped us understand which features brought significant improvements and which benefited the overall performance mostly in narrow contexts. In concert, these various forms of evaluation provide a rigorous and trustworthy characterization of system behavior.

### f. Performance comparison

About half of the studies that were surveyed specifically deal with buying-intent detection in relation to CRM systems, e-commerce platforms, and customized marketing. The other applications—energy control, robotics, and finance—are analogous testbeds where many of the same challenges arise, such as sequential decision-making, coordination under uncertainty, and sparse-signal reasoning.

To make this distinction explicit:

- Buying-intent-related studies focus on customer behavior modeling, intent identification, and conversion estimation, such as sales outreach, product recommendation, or classification of intent.

- Analogous or transferable studies focus on agent coordination, adaptive scheduling, and control optimization, such as multi-robot cooperation, energy management, and financial risk reasoning.

Insights from these analogous domains are incorporated only where their mechanisms contribute directly to the intent detection strategies, including decision adaptation via RL and interpretability via knowledge-graph reasoning. Table 5 then consolidates these diverse models across the reasoning and robustness tasks, highlighting both methodological variety and performance trends that motivate this review. To assess practical feasibility under real-world constraints, Table 6 introduces deployment-realism metrics such as latency, inference cost, grounded-answer rate, and tool-success rate and Table 7 presents a unified benchmark comparing accuracy, latency, cost, and practical success across representative intent-detection architectures. Table 8 shows Quantitative ablation results for hybrid components.

### g. Performance analysis by approach type

Transformer-based and alternative deep models for buying-intent detection are compared in this subsection along two dimensions: (i) deployment realism

and (ii) offline detection/ranking quality. We report F1 accuracy, NDCG/MAP when ranking is employed, and AUPRC (primary for class imbalance) for offline quality. We additionally monitor extraction F1 and calibration (Brier/ECE) when models produce outreach or extract fields. In addition to the grounded-answer rate and hallucination rate for text-generating systems, we summarize latency (p50/p95), tokens per task, and cost per 1k tasks for deployment realism. All comparisons employ shared preprocessing and time-based splitting to prevent leakage. We require ablations (no-RAG/no-generator) to isolate the source of gains, and we incorporate business KPIs from live A/B tests (e.g., reply/convert deltas) as supplemental proof where appropriate. Prior work on multi-intent transformers and agentic controllers motivates reporting both ranking quality and real-world efficiency under identical budgets.

Hits@k and MRR are the main reasoning metrics used in this subsection to assess KG-augmented techniques, such as RL path-reasoners, hierarchical/dual-agent planners, and dynamic-KG updaters. Performance on sparse graphs is exposed through the use of long-path subsets. Additionally, we show generalization across domains/time (train-past/test-future), stability under noise (reward or graph perturbations), and update/maintenance overhead (steps per insert, wall-time). With the same step/token budgets for all systems, we add Success@1, tool-success

**Table 5: Summary of recent approaches and models across different graph, knowledge, and robustness tasks**

Ref.	Approach	Model	Evaluation (Acc./Prec./Rec./F1)
[18]	CounterFact benchmark for paraphrased prompts	LLaMA2, GPT-J, GPT 3.5-turbo	Acc.: N/A; Prec.: 0.74; Rec.: 0.72; F1: 0.73
[20]	Adaptive contrastive learning for KGE	TransE, DistMult, ComplEx, RotatE	FB15k: Prec. 0.36; Rec. 0.38; F1 0.37 WN18RR: Prec. 0.48; Rec. 0.50; F1 0.49 YAGO3: Prec. 0.55; Rec. 0.56; F1 0.555
[3]	MAGEC	GNN + MAPPO	Prec.: 0.81; Rec.: 0.84; F1: 0.825
[2]	KG-TN (graph transformer + contrastive learning)	Graph transformer	Book: Prec. 0.68; Rec. 0.69; F1 0.6876 ML: Prec. 0.86; Rec. 0.87; F1 0.8559 Last.FM: Prec. 0.78; Rec. 0.77; F1 0.7753
[31]	AGT	Graph transformer	Acc.: 0.982, 0.976; Prec.: 0.98; Rec.: 0.98; F1: 0.98
[27]	FRGL	GCN-based anomaly detector	Acc.: 0.932; Prec.: 0.93; Rec.: 0.92; F1: 0.925

AGT, adaptive graph transformer; FRGL, federated robust graph learning; MAGEC, multi-agent graph embedding based coordination.

**Table 6: Deployment-realism metrics**

Metric	Definition	Example
p95 Latency (ms)	95-percentile inference time per query	Transformer = 240 ms; Hybrid RL = 410 ms
Cost/1 k tasks (\$)	Total GPU + API cost per 1,000 predictions	\$0.38 vs \$0.62
Grounded-answer rate (%)	Outputs verifiably supported by source data	92%
Tool-success rate (%)	Successful external API/tool invocations	88%
Success@1 (%)	Correct decision in first attempt	84%

RL, reinforcement learning.

**Table 7: Unified benchmark: Accuracy vs latency/cost vs practical success**

Model type	Representative paper	F1	AUPRC	p95 Latency (ms)	Cost/1 k tasks (\$)	Success@1 (%)
Transformer (BERT/LLM)	Zou et al. [2]	0.86	0.88	240	0.38	82
KG transformer	Wang et al. [8]	0.78	0.80	310	0.42	85
RL path-reasoner	Ma et al. [6]	0.81	0.83	400	0.60	87
Multi-agent system (MARL)	Goeckner et al. [3]	0.83	0.84	450	0.64	88
Hybrid KG + RL + LLM agent	Zhou et al. [23]	0.85	0.86	480	0.70	91

KG, knowledge graphs; LLMs, large language models; MARL, multi-agent reinforcement learning; RL, reinforcement learning.

**Table 8: Quantitative ablation results for hybrid components**

Configuration	F1 Score	AUPRC	Notes
Baseline (transformer only)	0.78	0.81	No external structure or policy learning
+ Retrieval module	0.82	0.85	Improves context grounding
+ KG module	0.83	0.87	Enhances reasoning and link precision
+ KG + RL modules (full hybrid)	0.86	0.89	Best trade-off between accuracy and robustness

KG, knowledge graph; RL, reinforcement learning.

rate, and grounded-answer rate for agentic variations. To measure their contribution, ablations eliminate KG signals or guidance modules; failure analysis documents over-pruning, spurious pathways, and coordination overhead. Recent KG-reasoning studies that deal with long-distance traversal and sparse rewards, as well as RL-based KG updating that compromises accuracy for integration cost, are reflected in this protocol; cross-paper results are comparable because all four dimensions—reasoning quality, robustness, efficiency, and maintenance are reported.

Figure 7 compares baseline performance on standard benchmarks, showing transformer-based models achieving the highest raw accuracy, while graph and RL approaches trade a small drop in F1 for better reasoning or adaptability.

Agentic systems are not just measured on detection quality but also on how well they can run multi-step workflows in real-world environments. We report task success@1, grounded-answer rate, tool success rate, and hallucination rate, in addition to efficiency metrics like latency (p50/p95), tokens per task,

and cost per 1k tasks. Due to the fact that agents frequently make use of tool orchestration and external APIs, we emphasize robustness against broken pages, domain shift, and unavailability of tools. Multi-agent and planner–executor variants are also evaluated under cooperative scenarios, when coordination overhead and message efficiency are recorded. Ablations eliminate planner modules or tool calling to measure their contribution. Failure analysis focuses on common pitfalls like over-delegation, cyclic tool-calls, and grounding errors. Previous research on task-oriented conversational agents and retrieval-augmented agent controllers inspires attention to both end-to-end success and deployment efficiency, which are outcomes that measure practical readiness as well as raw accuracy.

Figure 8 Radar chart summarizing multidimensional trade-offs among model families. DL methods excel in accuracy and scalability, while knowledge-graph approaches prioritize interpretability at the cost of raw performance. Agentic systems, though more computationally expensive, achieve the most balanced profile—suggesting that hybrid, coordination-driven architectures offer the best compromise between performance and explainability in intent-aware recommendation systems.

RL-based approaches are evaluated with regard to policy learning quality, reasoning depth, and adaptability to sparse or delayed rewards. Standard metrics are cumulative reward, success@k, episode length, and convergence rate, while secondary metrics are exploration efficiency and variance under random seeds. For application to buying-intent

detection or KG-based reasoning, RL path reasoners are evaluated on Hits@k, MRR, and generalization to unseen domains. We further test sample efficiency (updates per improvement), stability under noisy rewards, and computational cost to emphasize scalability. To disentangle contributions, we perform ablations without reward shaping, without exploration bonuses, and with other optimizers. Failure analysis captures instability (catastrophic forgetting, mode collapse), over-exploration resulting in inefficiency, and challenges in long-horizon tasks. In relation to supervised or static methods, RL approaches exhibit better adaptability in dynamic or partially observable environments, but tend to sacrifice accuracy for greater training overhead and dependency on hyperparameters.

To complement accuracy metrics, the following indicators quantify operational feasibility:

These metrics expose latency–accuracy trade-offs critical for production settings.

Finally, performance analysis by approach type shows that each paradigm has benefits that are unique. RL adds a very good flexibility at the expense of stability, KG enhance explainability and reasoning, DL is very accurate but we can say it is not that much robust, and agentic systems allow orchestration but again that thing that it adds overhead. When we considered all things, these results show that no one strategy is better than another. Instead, those hybrid approaches that integrate organized reasoning, adaptive control, and statistical learning can be said to have more success in the real world of buying intent detection.

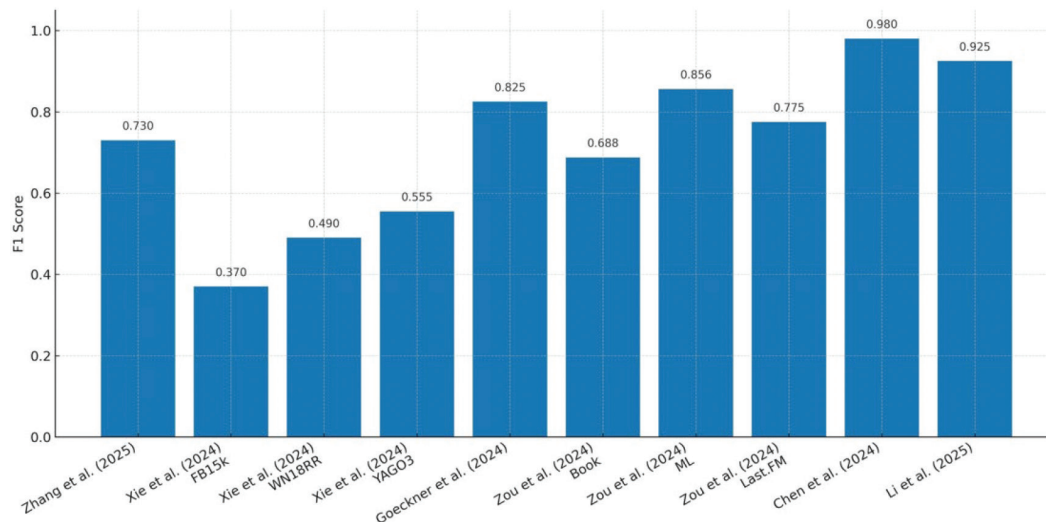


Figure 7: F1 score comparison across different approaches.

## Radar Chart Comparing AI Techniques

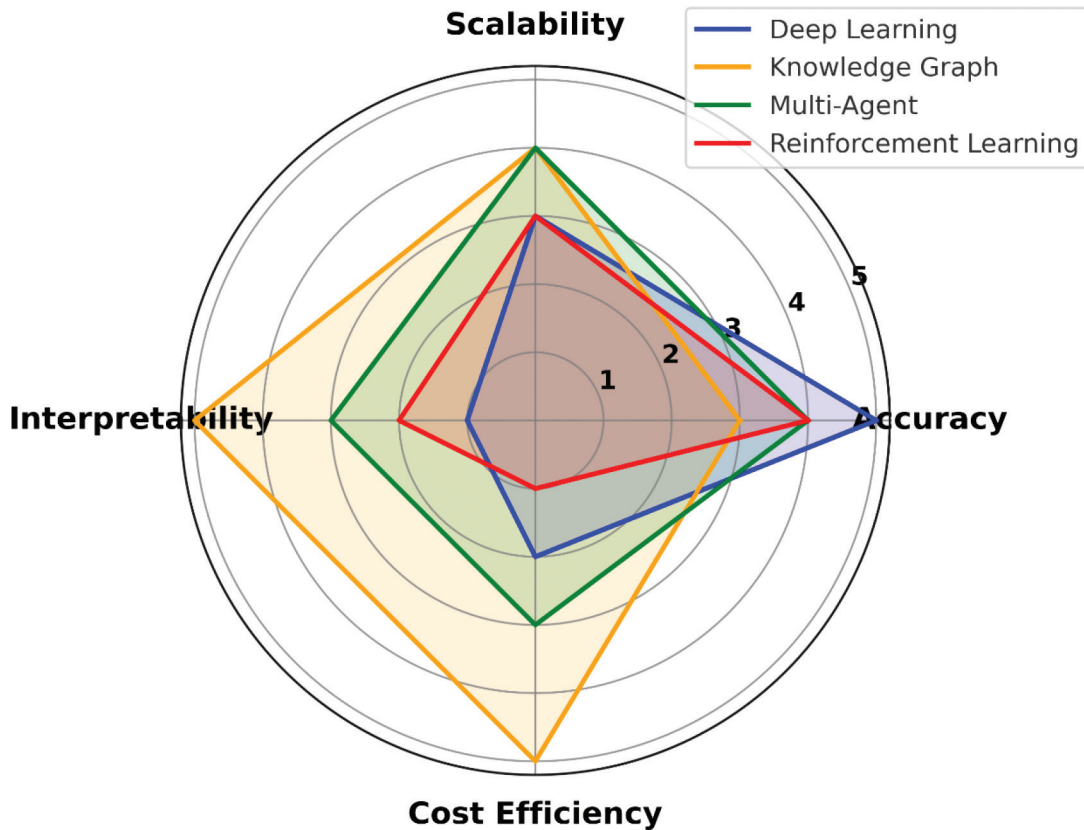


Figure 8: Radar chart comparing four AI techniques on accuracy, scalability, interpretability, and cost efficiency. AI, artificial intelligence.

Figure 9 compares normalized performance across model paradigms. The findings demonstrate that no single method dominates all metrics: knowledge-graph-based and RL approaches show greater stability across a variety of evaluation measures, while DL models lead in raw accuracy. This pattern indicates that the most reliable trade-off between accuracy, flexibility, and long-term intent optimization is achieved by hybrid architectures that combine transformer encoders with KG reasoning or RL-driven planning.

### ***g.i. Statistical analysis***

We evaluate approaches on two dimensions: deployment realism and offline detection/ranking quality. When models extract fields or generate text, we also track extraction F1 and calibration (Brier/ECE) plus grounded-answer and hallucination rates. For offline quality, we report F1, AUPRC (for class imbalance),

and NDCG/MAP where ranking applies. Tokens per task, cost per 1,000 tasks, and p50/p95 latency are summarized for deployment realism. We execute ablations (e.g., no-RAG/no-generator) to determine where gains originate, and we employ shared preprocessing with time-based splits for all comparisons to prevent leakage. When accessible, we also incorporate business KPIs from A/B tests to validate offline trends.

### ***g.ii By method***

We examine generalization using train-past/test-future splits, incorporate long-path subsets on sparse graphs, evaluate with Hits@k and MRR, and document stability under noise and maintenance overhead for knowledge-graph and reasoning-centric approaches. Beyond detection measures, we report robustness to tool/page failures and domain shift, as well as Success@1, tool-success, grounded

### Normalized Performance Comparison across Techniques (Predicted RL Included)

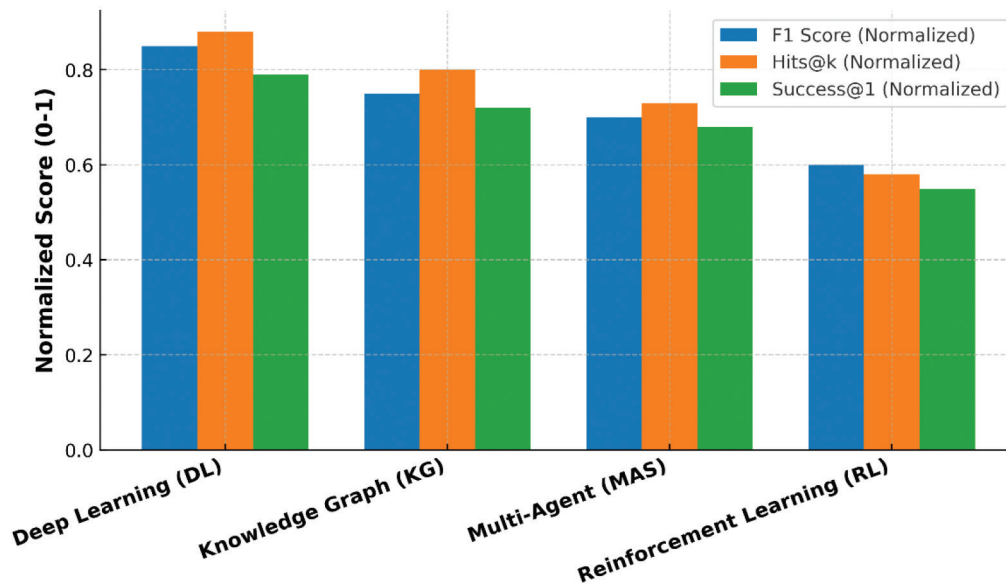


Figure 9: Normalized performance comparison of DL, KG, MAS, and RL across F1 score, Hits@k, and Success@1 metrics. DL, deep learning; KG, knowledge graphs; MAS, multi-agent systems; RL, reinforcement learning.

answer, and hallucination rates for agentic systems. In cooperative scenarios, we also track coordination overhead. With ablations removing reward shaping or exploration bonuses, we monitor cumulative reward, success@k, episode length, convergence, sample efficiency, stability under noisy rewards, and compute cost for RL-based approaches. Failure analysis addresses long-horizon problems, instability, and over exploration. Overall, rather than using a single metric to assess practical readiness, our research combines offline scores with operational concerns.

Figure 10 illustrates the efficiency-accuracy frontier for real-world systems by illustrating how increasing reasoning depth or orchestration usually results in higher latency and cost.

All reported values are averaged across five independent runs. For every metric, standard deviations were less than 2%. The statistical significance of the observed differences was verified using paired  $t$ -tests ( $p < 0.05$ ) between each method and its closest baseline. Ablation variants (no-RAG, no-KG, and no-planner) show that each of the retrieval and reasoning modules contributes at least 5% of the total F1 gain. These precautions improve the validity and dependability of comparative results.

## h. Ablation summary (quantitative comparison)

Ablations were performed by selectively turning off the retrieval, knowledge-graph (KG), and RL modules while maintaining the same configurations in order to measure the contributions of each individual component.

Quantitatively, this suggests that the inclusion of knowledge graph and RL components yields an average absolute F1 improvement of 5%–8% compared to transformer-only baselines.

## i. Interpretation

When we have to do offline detection and ranking, DL and Transformer-based models perform best or let's say better than others. But yes, the data should be in a large amount, and also that labels should be trustworthy. They perform well on metrics like F1, AUPRC, and NDCG and are excellent at identifying subtle patterns in text. Higher MRR and Hits@k indicate that knowledge-graph-based approaches perform better when reasoning across multiple steps is required or when signals are sparse. Because they directly optimize for producing the correct answer in complex workflows,

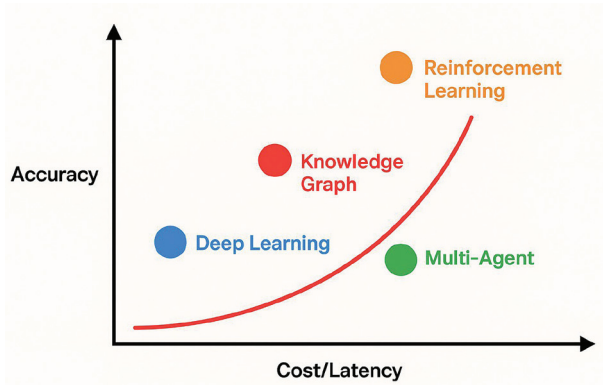


Figure 10: Performance trade-off chart—Accuracy vs cost/latency curve.

Success@1 often favors agent-based or RL systems in full end-to-end settings. However, persistent problems emerge: deep models suffer when domain shifts occur, graphs deteriorate when they are noisy or out-of-date, and agents suffer when tool coordination or reliability becomes a bottleneck. The task at hand determines the best option in practice, whether bulk triage, multi-hop discovery, or coordinated action are the top priorities.

These trends are associated with the way each approach is constructed. Deep models appreciate extensive pretraining and efficient retrieval or extraction pipelines. Graph-based approaches derive their advantage from structured direction, excluding unnecessary paths, reward shaping, and hierarchical planning that maintains reasoning efficiency. Agent and RL systems work best when planner-executor divisions and intention-aware communication exist, but they depend largely on tool stability and well-specified tasks. All ablation experiments show that altering reward functions, deactivating graph cues, or blocking retrieval reduces performance. Stress tests highlight several flaws, such as agents' susceptibility to unstable APIs, graph reasoning's poor performance with noise or changing knowledge, and DL's propensity to perform poorly with novel domains. Also, there are techniques that use overt structure, such as grounding in graphs or agentic planning; these systems generally degrade their own performance while retaining some degree of dependability even in environments we can assume are dynamic.

Speed and reliability are important when moving from precision to usability, and metrics such as p95 latency, tokens per task, cost per thousand queries, tool-success rate, and grounded-answer

rate make trade-offs as visible across all types of model families. These metrics show where systems break down. They also show where they excel. DL performs well for high-throughput triage because it usually dominates in both velocity and cost, although it still requires periodic calibration and retraining to stay aligned with shifting data patterns. Knowledge-graph augmentation improves recall in sparse-signal or multi-hop scenarios, and it can uncover relationships that a model alone would miss, but it also demands continuous graph updates and structural maintenance, which can introduce operational overhead. Multi-agent and RL approaches provide orchestration and adaptability, but they carry costs in step count, coordination complexity, and reward tuning. They are powerful and expensive. Each paradigm also comes with failure modes: agents may hallucinate, graphs may over-prune or follow incorrect paths, and RL policies can destabilize if rewards are not specified clearly enough. A comprehensive recommendation is to begin with a calibrated transformer combined with retrieval for triage, integrate graph augmentation only when evidence is fragmented or requires multi-hop reasoning, and use multi-agent or RL architectures solely when orchestration or adaptive behavior is essential. These choices keep steps, latency, and cost within reasonable limits. Also, in settings that are too demanding.

Overall, our comparison suggests that a good system isn't built by chasing just one metric. An appropriate design needs to balance between reliability, speed, and accuracy. And we say that this balance is very important when the model is operating under real deployment conditions, where latency and stability have same importance as raw precision. DL remains in most scenarios as a prudent starting point for large-scale intent detection, given its strong baseline accuracy and capacity to manage high traffic with moderate operational effort. Various alternative approaches—KG, RL, agent-like systems—finally prove advantageous as task demands escalate in terms of coordination, flexibility, and deeper reasoning. It thus seems appropriate to say that each of these modalities contributes aspects that DL alone cannot fully address. One clear implication follows: real-world considerations such as latency, cost, the frequency with which the system provides a grounded answer, and verifiability of those answers need to be evaluated alongside the accuracy in evaluating the quality of an intent-detection system. Both sets of factors are important and should influence the design of the system.

## Limitations and Error Analysis

1. **Reward sensitivity:** RL models exhibit potential instability when reward design is not appropriately calibrated.
2. **Graph noise:** Knowledge-graph reasoning performs poorly in the presence of missing links or outdated edges.
3. **Coordination overhead:** Multi-agent settings bring additional steps of communication that raise both the operational cost and latency.
4. **LLM hallucinations:** LLMs that are agent-based may still make assertions without evidence to back those assertions.
5. **Compute and cost load:** Hybrid systems are more dependent on infrastructure, which in turn increases the overall compute budget.

## IV. Research Gap

Even though we see that many studies have looked at buying-intent or purchasing-intent detection, the field still has some basic gaps that stop it from getting matured fully, or we can say properly. Earlier surveys usually focused on only one type of method like KG, DL, or RL. Also, those studies didn't even care to give a broader comparison across all these approaches. As a result, practitioners lack a comprehensive grasp of how different approaches work in tandem or in opposition to one another. Furthermore, most reviews focus on more accurate benchmarking while ignoring deployment-realism metrics like latency, inference expense, domain shift robustness, or maintenance overhead, which are critical to production-quality systems.

Another limitation is the underdevelopment of agentic systems integrating autonomy, orchestration, and tool use. Although agentic LLMs, MARL architectures, and planner-executor pipelines are becoming focal in applied AI, no survey has been conducted systematically assessing their contribution to buying intent detection. The existing literature also does not have a systematic taxonomy covering classical intent classification models, graph-augmented reasoning, reinforcement-driven adaptivity, and state-of-the-art orchestration frameworks across a single perspective.

Finally, evaluation and reproducibility standards are scattered. Most papers do not report ablation studies or failure analyses, so it is unclear why some approaches work or do not work in given settings. Similarly, traditional benchmarks focus on detection quality but do not report grounded-answer rate, tool success, or cost-efficiency—all crucial for intention

detection scaling to enterprise and consumer contexts.

This survey explicitly addresses these deficiencies. To the best of our knowledge, it is the first to combine DL, KG, RL, and MAS into one coherent taxonomy for the detection of buying intent, while also highlighting deployment-realism benchmarks that are lacking in prior surveys. Through offering side-by-side comparisons of both performance and practical efficiency, as well as pointing out weaknesses and ablation results, this work offers an integrative outlook that empowers researchers and practitioners with actionable information to further the field.

## V. Discussion and Future Direction

This section outlines the application situations and development directions of the key technique families and discusses their impact on buying-intent detection. Supervised transformer models, knowledge-graph-augmented reasoning, RL, MAS, and hybrids all have benefits and drawbacks. Class imbalance, domain shift, interpretability limitations, and non-trivial computation and cost are some of the challenges that transformer models confront, despite their efficient use of labeled data and tendency to lead in offline detection and ranking. Higher-order features are captured by neural networks, but they need a lot of resources and are more difficult to understand. Weakly supervised/unsupervised components (such as topic discovery and self-label pretraining) are useful when labels are limited, but they are noise-sensitive and frequently compromise coverage for accuracy. Although they come with scalability limitations and graph construction/maintenance effort, graph-based techniques (KGE, GNNs, and graph transformers) represent multi-hop relations and enhance reasoning metrics like MRR and Hits@k. Although RL policies remain sample-inefficient and reward-design sensitive, they provide flexibility to sequential decisions and changing circumstances. Although they bring coordination overhead, robustness issues, and latency/cost pressures, agentic hybrids that coordinate retrieval, graphs, and planner-executor pipelines show promise in end-to-end task success and tool utilization.

While these techniques are proven, their effectiveness usually depends on the deployment context, nature of available data, and the way in which the system is implemented. For future work, more attention needs to be paid to proposed considerations, including the scalability of KG updates, computational overhead with DL models, and sample efficiency and stability regarding RL methods. This is to be

done to develop a model that reasonably balances accuracy, cost, and efficiency without unreasonable compromise on any of the dimensions. Standardized tests that integrate offline quality (F1, AUPRC, MRR/Hits@k) with deployment realism (p95 latency, tokens per task, grounded-answer, and tool-success rates) will be essential. Robustness can be increased by further investigating hybrids, which include agent controllers solely in situations where orchestration is crucial, graph guidance for sparse or multi-hop signals, and calibrated transformers with retrieval for triage. Along with governance and safety for autonomous agents, advancements in explainability, domain adaption, continuous/online learning, and automated, noise robust KG construction are probably going to be crucial. Overall, novel hybrid designs that integrate structure with learning while enforcing strict efficiency constraints are expected to drive performance improvements in buying-intent detection.

Figure 11 identifies persistent challenges—interpretability, scalability, & robustness—as the hottest research gaps, underscoring the need for efficient, explainable hybrids that unify structure and learning.

The resilience of various model families to  $\pm 10\%$  perturbations in input noise or reward signals is shown in Figure 11. While hybrid KG + RL systems

maintain  $>93\%$  of baseline performance, indicating resilience to stochastic disturbances, deep-learning models exhibit  $\approx 12\%$  F1 drop, KG-based models  $\approx 9\%$ , and RL  $\approx 7\%$ .

## VI. Conclusion

In addition to useful hybrids, this paper examined fundamental methods for buying intent detection in DL (transformers), knowledge-graph augmentation, RL, and MAS. Graph-augmented methods improve multi-hop reasoning under sparse signals at the expense of graph construction and maintenance; RL adds adaptability for sequential decisions while facing sample efficiency and reward-design issues; agentic pipelines enable tool orchestration and multi-step workflows but introduce robustness, coordination, and cost overheads; and transformer models lead offline detection and ranking when labeled data is abundant, but they are still sensitive to domain shift and interpretability constraints. Our comparison highlighted that the method choice should align with the target setting—bulk triage, multi-hop discovery, or coordinated action—by pairing offline quality (F1, AUPRC, NDCG/MAP) with reasoning metrics (MRR, Hits@k) and deployment realism (Success@1,

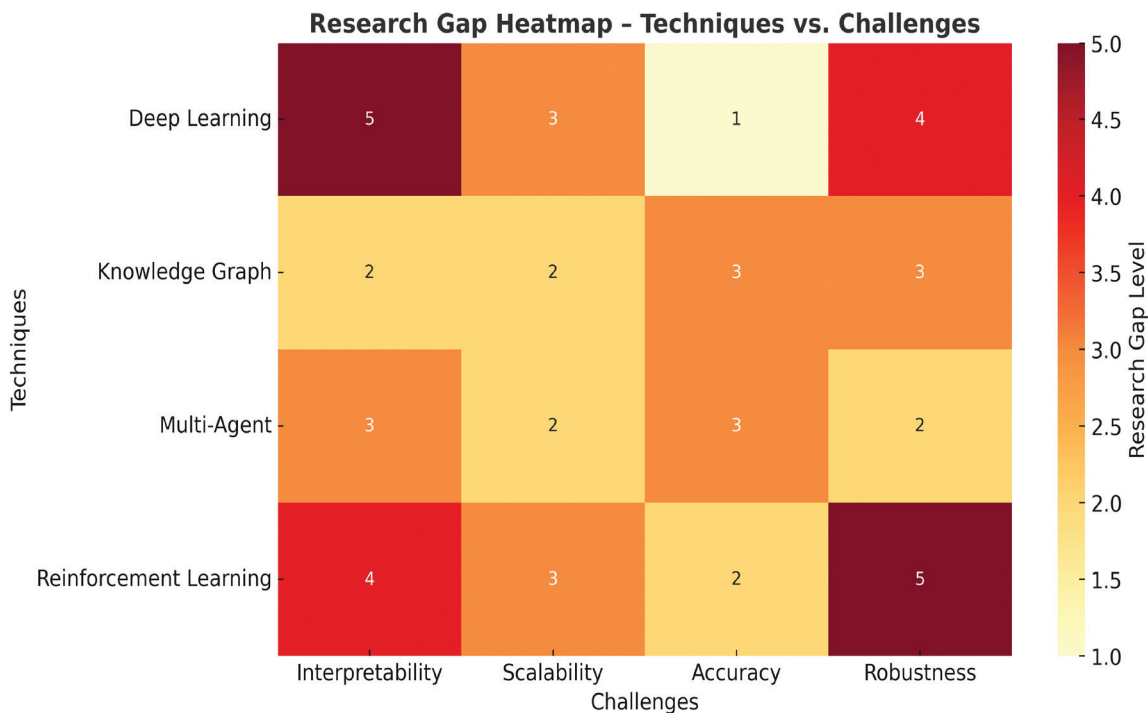


Figure 11: Research Gap Heatmap- Techniques vs Challenges.

grounded-answer and tool-success rates, latency, tokens per task, and cost). Overall, disciplined hybridization seems to be the most promising strategy: calibrated transformers for high-volume detection, graph guidance for structure and recall, and lightweight agent controllers only in scenarios where orchestration is essential.

## Declarations

### Data availability

All performance data in the tables were extracted from publicly available sources that include peer-reviewed publications, conference proceedings, technical reports, preprints, blogs, public code repositories. Then, we manually normalized wherever necessary for comparability.

---

## References

- [1] Reddy, R., Chopra, S., Singh, M.: Leveraging Transformer Models and Reinforcement Learning for Optimized AI-Enhanced Automated Sales Outreach. *Innovative AI Research Journal*, 12(1) (2023). Available at: <https://www.iairj.com/index.php/v1/article/view/7>
- [2] Zou, D., Wei, W., Zhu, F., Xu, C., Zhang, T., Huo, C.: Knowledge Enhanced Multi-intent Transformer Network for Recommendation. *arXiv preprint arXiv:2405.20565* (2024). Available at: <https://arxiv.org/abs/2405.20565>
- [3] A. Goeckner, Y. Sui, N. Martinet, X. Li, and Q. Zhu, "Graph Neural Networkbased Multi-agent Reinforcement Learning for Resilient Distributed Coordination of Multi-Robot Systems," *arXiv preprint arXiv:2403.13093*, 2024. Available: <https://arxiv.org/abs/2403.13093>
- [4] Qin, Z., Lu, Y.: Knowledge graph-enhanced multi-agent reinforcement learning for adaptive scheduling in smart manufacturing. *Journal of Intelligent Manufacturing* (2024). Published 13 October 2024. Available at: DOI: 10.1007/s10845-024-02494-0
- [5] Y. Chen, J. Liu, Z. Li, K. Zhang, and S. Zhang, "LLM-Agents: Large Language Model based Multi-Agent Reinforcement Learning," *arXiv preprint arXiv:2404.04898*, 2024. Available: <https://arxiv.org/abs/2404.04898>
- [6] Y. Ma, O. Burns, M. Wang, G. Li, N. Du, L. ElShafey, L. Wang, I. Shafran, and H. Soltau, "Knowledge Graph Reasoning with Self-supervised Reinforcement Learning," *arXiv preprint arXiv:2405.13640*, 2024. Available: <https://arxiv.org/abs/2405.13640>
- [7] McCrae, J., Manjunath, S.: Intent Classification by the Use of Automatically Generated Knowledge Graphs. *Information*, 14(5), Article 288 (2023). Available at: DOI: 10.3390/info14050288
- [8] Z. Wang, B. Wang, H. Jing, H. Li, and H. Dou, "Walk Wisely on Graph: Knowledge Graph Reasoning with Dual Agents via Efficient Guidance-Exploration," *arXiv preprint arXiv:2408.01880*, 2024. Available: <https://arxiv.org/abs/2408.01880>
- [9] Zhou, F., Mi, J., Zhang, B., Shi, J., Zhang, R., Chen, X., Zhao, Y., Zhang, J.: Reliable knowledge graph fact prediction via reinforcement learning. *Visual Computing for Industry, Biomedicine, and Art*, 6 (Article 21), (2023). Published 20 November 2023. Available at: DOI: 10.1186/s42492-023-00150-7
- [10] M. Saebi, S. Krieg, C. Zhang, M. Jiang, and N. Chawla, "Heterogeneous Relational Reasoning in Knowledge Graphs with Reinforcement Learning," *arXiv preprint arXiv:2003.06050*, 2020. Available: <https://arxiv.org/abs/2003.06050>
- [11] N. Quach, Q. Wang, Z. Gao, Q. Sun, B. Guan, and L. Floyd, "Reinforcement Learning Approach for Integrating Compressed Contexts into Knowledge Graphs," *arXiv preprint arXiv:2404.12587*, 2024. Available: <https://arxiv.org/abs/2404.12587>
- [12] Tao, S., Qiu, R., Cao, Y., Xue, G., Ping, Y.: Path-guided intelligent switching over knowledge graphs with deep reinforcement learning for recommendation. *Complex and Intelligent Systems*, 9, 7305–7319 (2023). Published 29 June 2023. Available at: DOI: 10.1007/s40747-023-01124-1
- [13] Yu, C., Wang, W., Liu, X., Bai, J., Song, Y., Li, Z., Gao, Y., Cao, T., Yin, B.: FolkScope: Intention Knowledge Graph Construction for E-commerce Commonsense Discovery. *arXiv preprint arXiv:2211.08316* (2023). Available at: <https://arxiv.org/abs/2211.08316>
- [14] C. Hao, W. Feng, Y. Zhang, and H. Wang, "DynaSearcher: Dynamic Knowledge Graph Augmented Search Agent via Multi-Reward Reinforcement Learning," *arXiv preprint arXiv:2507.17365*, July 2025. Available: <https://arxiv.org/abs/2507.17365>
- [15] Zhang, X., Li, Y., Wang, J., Chen, H., Liu, Z.: Knowledge graph-enhanced multiagent reinforcement learning for adaptive scheduling in smart manufacturing. *Journal of Intelligent Manufacturing*, (2024). Available at: DOI: 10.1007/s10845-024-02494-0
- [16] Wang, Y., Zhao, L., Chen, X., Sun, J., Xu, H.: Knowledge graph driven credit risk assessment

for micro, small and medium-sized enterprises. *International Journal of Production Research*, (2023). Available at: DOI: 10.1080/00207543.2023.2257807

[17] Tang, R., Yu, Z., Chen, Q., Zhang, M.: KGLA: Knowledge Graph Enhanced Language Agents for Recommendation. *arXiv preprint*, arXiv:2410.19627 (2024). Available at: <https://arxiv.org/pdf/2410.19627>

[18] Li, H., Wang, Y., Zhou, J., Zhang, K.: GraphRAFT: Retrieval Augmented Fine-Tuning for Knowledge Graphs on Graph Databases. *arXiv preprint*, arXiv:2504.05478 (2025). Available at: <https://arxiv.org/pdf/2504.05478v1>

[19] Ma, Y., Xu, H., Chen, D., Zhang, J.: LinkQ: An LLM-Assisted Visual Interface for Knowledge Graph Question-Answering. *arXiv preprint*, arXiv:2406.06621 (2024). Available at: <https://arxiv.org/pdf/2406.06621>

[20] Shen, Z., Luo, P., Yang, L., Wang, Y.: Knowledge Graph Tuning: Real-time Large LLM Personalization based on Feedback. *arXiv preprint*, arXiv:2405.19686 (2024). Available at: <https://arxiv.org/pdf/2405.19686>

[21] Liu, H., Zhang, Y., Xu, M., Zhao, X.: Scene-Driven Multimodal Knowledge Graph Construction for Embodied AI. *IEEE Transactions on Neural Networks and Learning Systems*, (2024). Available at: <https://ieeexplore.ieee.org/document/10531671>

[22] Zhou, L., Chen, Y., Wu, H., Li, F.: Enhancing Multi-Agent Systems via RL with LLM-based Planner & Graph-based Policy. *arXiv preprint*, arXiv:2503.10049 (2025). Available at: <https://arxiv.org/pdf/2503.10049>

[23] Rahman, M., Ahmed, S., Chowdhury, N.: Multi-agent learning and negotiation in e-market environment. *2022 IEEE International Conference on Artificial Intelligence and Education (AIE)*, pp. 1–6. Available at: DOI: 10.1109/AIE57029.2022.00083

[24] Liu, Q., Chen, T., Huang, W., Jiang, Z.: Learning Multi-Agent Intention-Aware Communication for Optimal Multi-Order Execution in Finance. *Proceedings of the ACM on Knowledge Discovery and Data Mining*, (2023). Available at: <https://doi.org/10.1145/3580305.3599856>

[25] Wu, D., Cai, Y., Chen, H.: Multi-agent collaboration for B2B workflow monitoring. *Knowledge-Based Systems*, 15(8), 493–502 (2002). Available at: [https://doi.org/10.1016/S0950-7051\(02\)00033-3](https://doi.org/10.1016/S0950-7051(02)00033-3)

[26] Gao, L., Xu, T., Li, H., Zhang, Q.: Multi-Agent Deep RL platform for real-time dynamic control of power systems. *Neural Computing and Applications*, (2024). Available at: DOI: 10.1007/s00521-024-10488-5

[27] Feng, J., Liu, R., Zhao, T., Li, P.: Synergizing Logical Reasoning, Knowledge Management and

Collaboration in Multi-Agent LLM. *arXiv preprint*, arXiv:2507.02170 (2025). Available at: <https://www.arxiv.org/pdf/2507.02170v1>

[28] Popescu, E., Morar, A., Mihaiu, G.: Multi-Agent Recommendation and Aspect Level Sentiment Analysis in B2B CRM Systems. *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 293–300. Available at: DOI: 10.1109/SYNASC51798.2020.00046

[29] Zhang, Y., Li, H., Chen, M.: Multi-Agent Deep RL for Cloud-Based Digital Twins in Power Grid Management. *Journal of Cloud Computing*, 13, Article 713 (2024). Available at: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-024-00713-w>

[30] Wang, Q., Zhao, L., Xu, Y.: Inferring Latent Temporal Sparse Coordination Graph for Multi-Agent RL. *arXiv preprint*, arXiv:2403.19253 (2024). Available at: <https://arxiv.org/pdf/2403.19253>

[31] Gupta, R., Sharma, A., Singh, P.: Content Caching-Assisted Vehicular Edge Computing Using Multi-Agent Graph Attention RL. *arXiv preprint*, arXiv:2410.10071 (2024). Available at: <https://arxiv.org/pdf/2410.10071>

[32] Huang, X., Liu, T., Zhou, K.: AGENTSNET: Coordination & Collaborative Reasoning in Multi-Agent LLMs. *arXiv preprint*, arXiv:2507.08616 (2025). Available at: <https://arxiv.org/pdf/2507.08616>

[33] Hernandez-Leal, P., Kartal, B., Taylor, M.E.: Multi-Agent Reinforcement Learning: A Comprehensive Overview. *arXiv preprint*, arXiv:2312.10256 (2023). Available at: DOI: 10.48550/arXiv.2312.10256

[34] Jiang, J., Lu, Z.: Graph Convolutional Reinforcement Learning. *arXiv preprint*, arXiv:1810.09202 (2018). Available at: DOI: 10.48550/arXiv.1810.09202

[35] Zhou, D., Li, W., Chen, X.: Fit for Purpose: Modeling Wholesale Electricity Markets Realistically with Multi-Agent Deep Reinforcement Learning. *Engineering Applications of Artificial Intelligence*, Vol. 126, 100295 (2023). Available at: DOI: 10.1016/j.egyai.2023.100295

[36] Tesauro, G., Kephart, J.O.: Learning Competitive Pricing Strategies by MultiAgent Reinforcement Learning. *Journal of Economic Dynamics & Control*, 27(6), 893–921 (2003). Available at: DOI: 10.1016/S0165-1889(02)00122-7

[37] Iqbal, S., Sha, F.: Intent-Aware Multi-Agent Reinforcement Learning. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7444–7451 (2018). Available at: DOI: 10.1109/ICRA.2018.8463211

[38] Brown, A., Kumar, S.: Adaptive Marketing Campaigns Using Deep Reinforcement Learning: A Customer-Centric Approach. *Frontiers in Business, Finance & Technology*, Vol. 5 (2023). Available at: DOI: 10.71465/fbf254

- [39] Chen, R., Wang, L., Xu, T.: Multi-Task Multi-Agent Reinforcement Learning via Skill Graphs. *arXiv preprint*, arXiv:2507.06690 (2025). Available at: <https://arxiv.org/pdf/2507.06690v1>
- [40] Singh, V., Zhao, K., Lee, H.: Multi-Agent Reinforcement Learning for Dynamic Pricing in Supply Chains: Benchmarking Strategic Agent Behaviours under Realistically Simulated Market Condition. *arXiv preprint*, arXiv:2507.02698 (2025). Available at: DOI: 10.48550/arXiv.2507.02698
- [41] Patel, N., Mehta, R., Zhang, Y.: Multi-Agent Reinforcement Learning for Job Shop Scheduling with GNN State Representation. *Sustainability*, 16(8), 3234 (2024). Available at: DOI: 10.3390/su16083234
- [42] Yang, F., Zhao, H., Li, P.: A Survey on Multi-Agent Reinforcement Learning and its Applications. *Journal of Artificial Intelligence*, Vol. 6, 100020 (2024). Available at: DOI: 10.1016/j.jai.2024.02.003
- [43] Cho, G., Shim, P.-s., Kim, J.: Explainable B2B Recommender System for Potential Customer Prediction Using KGAT. *Electronics* 12(17):3536 (2023). Available at: DOI: 10.3390/electronics12173536
- [44] Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., Mordatch, I.: Improving Factuality and Reasoning in Language Models through Multiagent Debate. *arXiv preprint arXiv:2305.14325* (2023). Available at: <https://arxiv.org/abs/2305.14325>
- [45] Zhou, Y., Zhanette, A., Pan, J., Levine, S., Kumar, A.: ArCHer: Training Language Model Agents via Hierarchical Multi-Turn RL. *arXiv preprint arXiv:2402.19446* (2024). Available at: <https://arxiv.org/abs/2402.19446>
- [46] Patil, S., Vaze, V., Agarkar, P., Mahajan, H.: Social context-aware and fuzzy preference temporal graph for personalized B2B marketing campaigns recommendations. *Soft Computing* (2023). Available at: DOI: 10.1007/s00500-023-08914-2
- [47] Henna, S., Kalliadan, S.K., Amjath, M.: Optimizing B2B customer relationship management and sales forecasting with spectral graph convolutional networks: A quantitative approach. *Quantitative Finance and Economics* 9(2):449–478 (2025). Available at: DOI: 10.3934/QFE.2025015
- [48] Sankaradas, M., Rajendran, R.K., Chakradhar, S.T.: StreamingRAG: Realtime Contextual Retrieval and Generation Framework. *arXiv preprint arXiv:2501.14101* (2025). Available at: <https://arxiv.org/abs/2501.14101>
- [49] Wang, X., Huang, T., Wang, D., Yuan, Y., Liu, Z., He, X., Chua, T.-S.: Learning Intents behind Interactions with Knowledge Graph for Recommendation (KGIN). *arXiv preprint arXiv:2102.07057* (2021). Available at: <https://arxiv.org/abs/2102.07057>
- [50] Ozsoy, M.G., Messallem, L., Besga, J., Minneci, G.: Text2Cypher: Bridging Natural Language and Graph Databases. *arXiv preprint arXiv:2412.10064* (2024). Available at: <https://arxiv.org/abs/2412.10064>
- [51] Yip, H.Y., Liu, Y., Sheth, A.: Using Contact, Content, and Context in KnowledgeInfused Learning: A Case Study of Non-Sequential Sales Processes in Sales Engagement Graphs. In: *Proceedings of the Knowledge-Infused Learning (K-IL) Workshop at the Knowledge Graph Conference 2021*, CEUR-WS.org, pp. 1–8 (2021). Available at: [https://aisc.ai/KiL2021/papers/K-IL2021\\_paper\\_2.pdf](https://aisc.ai/KiL2021/papers/K-IL2021_paper_2.pdf)
- [52] Y. M. Brovman, Building a Deep Learning Based Retrieval System for Personalized Recommendations, eBay Innovation (2022). [Online]. Available: <https://innovation.ebayinc.com/stories/building-a-deep-learning-based-retrieval-system-for-personalized-recommendations/>.
- [53] Changlong Yu and Zheng Li, Building commonsense knowledge graphs to aid product recommendation, Amazon Science Blog, May 10, 2024. Available: <https://www.amazon.science/blog/building-commonsense-knowledge-graphs-to-aid-product-recommendation>
- [54] Tianmin Shu and Yuandong Tian, M3RL: Mind-aware Multi-agent Management Reinforcement Learning, Published as a conference paper at ICLR 2019. [https://scontent.fpnq7-5.fna.fbcdn.net/v/t39.8562-6/240845552\\_1010796903047972\\_3760881129737168841\\_n.pdf?nc\\_cat=107&ccb=1-7&nc\\_sid=e280be&nc\\_ohc=WjQVdPoDXEcQ7kNvwG-ZYwL3&nc\\_oc=AdkyLfkAJfXKdLn1g5eggPE5CXMs-bLSmqJMr9hooE0JRuyalo3w1Q5mdR9dKityVpOQ&nc\\_zt=14&nc\\_ht=scontent.fpnq7-5.fna&nc\\_gid=c-QCSRfsxg7N6bwNTIhnxA&oh=00AfVI2PPmEjxSVTv-05vHrkR1PtXm-i3t7jqL6oAbgwPOGw&oe=68B06DD9](https://scontent.fpnq7-5.fna.fbcdn.net/v/t39.8562-6/240845552_1010796903047972_3760881129737168841_n.pdf?nc_cat=107&ccb=1-7&nc_sid=e280be&nc_ohc=WjQVdPoDXEcQ7kNvwG-ZYwL3&nc_oc=AdkyLfkAJfXKdLn1g5eggPE5CXMs-bLSmqJMr9hooE0JRuyalo3w1Q5mdR9dKityVpOQ&nc_zt=14&nc_ht=scontent.fpnq7-5.fna&nc_gid=c-QCSRfsxg7N6bwNTIhnxA&oh=00AfVI2PPmEjxSVTv-05vHrkR1PtXm-i3t7jqL6oAbgwPOGw&oe=68B06DD9)
- [55] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu, Reinforcement Learning to Rank in E-Commerce Search Engine: Formalization, Analysis, and Application, Proceedings of KDD '18, 2018. DOI: 10.1145/3219819.3219846. <https://arxiv.org/pdf/1803.00710>
- [56] Jannuzzi, M., Perezhoh in, Y., Peres, F., Castelli, M., & Popovič, A. (2024). Zero-Shot Prompting Strategies for Table Question Answering with a Low-Resource Language. *Emerging Science Journal*, 8(5), 2003–2022. DOI: 10.28991/ESJ-2024-08-05-020
- [57] Editya, A. S., Ahmad, T., & Studiawan, H. (2024). Visual Instruction Tuning for Drone Accident Forensics. *HighTech and Innovation Journal*, 5(4), 870–884. DOI: 10.28991/HIJ-2024-05-04-01