IoT

MDPI

Review

From Machine Learning-Based to LLM-Enhanced: An Application-Focused Analysis of How Social IoT Benefits from LLMs

Lijie Yang [1,†] and Runbo Su [2,*,†]

[1] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA; lijiey@andrew.cmu.edu
[2] Inria, Loria, CNRS, University of Lorraine, F-54000 Nancy, France
* Correspondence: runbo.su@inria.fr
† These authors contributed equally to this work.

Abstract: Recent advancements in large language models (LLMs) have added a transformative dimension to the social Internet of Things (SIoT), which is the combination of social networks and IoT. With LLMs' natural language understanding and data synthesis capabilities, LLMs are regarded as strong tools to enhance SIoT applications such as recommendation, search, and data management. This application-focused review synthesizes the latest related research by identifying both the synergies and the current research gaps at the intersection of LLMs and SIoT, as well as the evolutionary road from machine learning-based solutions to LLM-enhanced ones.

Keywords: social IoT; large language model; application; recommendation; search; data management; review

1. Introduction

Early IoT was regarded as a domain-specific network of sensors and devices collecting and processing data in a relatively local manner. As research progressed, the IoT vision expanded to link more and more heterogeneous devices, enabling them to share information and interoperate to support diverse services.

Despite these improvements, IoT remained largely a client-server paradigm in which people consumed data from things rather than interacting with devices directly as peers. Figure 1 visually illustrates the conceptual transition from pre-Internet to SIoT. It shows how interactions evolve from basic device communication (thing-to-thing) toward human-device social interactions, emphasizing how pervasive connectivity and social relationships among devices and users create richer ecosystems. It sets the stage for understanding why social aspects must be integrated into IoT environments. As shown in this figure, recognizing that social networks drive much of modern human behavior, researchers in [1] took the first steps toward "socializing" IoT by leveraging platforms such as Twitter to discover and share the capabilities of smart objects among friends. On the other hand, this work did not fully align with IoT's core vision of ubiquitous, object-centric service provision. Shortly thereafter, the term "Social IoT" was originally introduced to embed social network principles explicitly within IoT systems [2]. An important contribution defined a set of relationship management policies specifying various fundamental ties that govern how devices form social connections, collaborate, and discover services in a manner analogous to human relationships [3]. The authors in [4] proposed a reference SIoT architecture alongside a roadmap of enabling technologies and research directions.

check for updates

Academic Editor: Amiya Nayak

Received: 30 March 2025
Revised: 25 April 2025
Accepted: 29 April 2025
Published: 30 April 2025

Citation: Yang, L.; Su, R. From Machine Learning-Based to LLM-Enhanced: An Application-Focused Analysis of How Social IoT Benefits from LLMs. *IoT* **2025**, *6*, 26. https://doi.org/10.3390/iot6020026

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

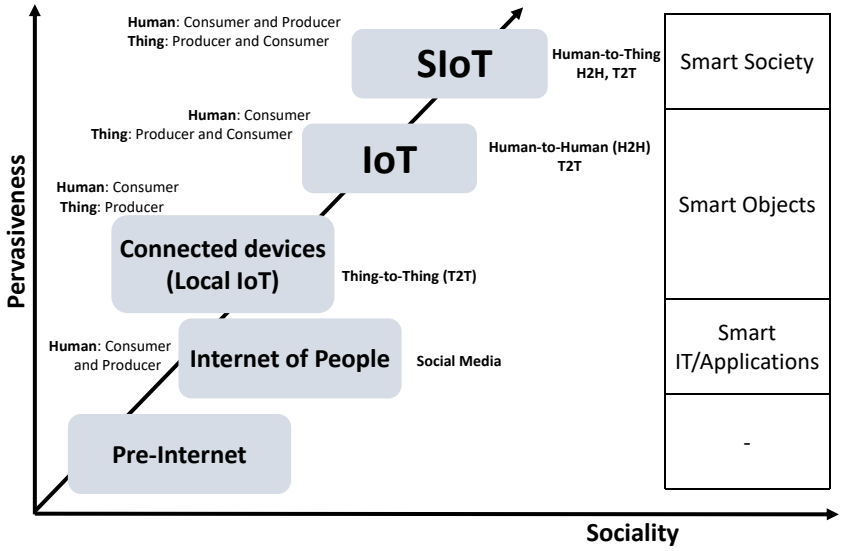

Figure 1. Evolution from pre-Internet to SIoT.

Building on these foundations, current SIoT trends emphasize semantic middleware for interoperable SIoT-based human-to-human communication, social recommendation, engines for dynamic and generative information retrieval, and edge-centric analytics to perform data management [5–7]. These advances underpin emerging SIoT applications across smart cities, vehicular networks, healthcare, and industrial IoT, realizing socially empowered devices that collaborate, learn, and self-organize to meet user needs. The SIoT architecture proposed in [4] considers a dynamic ecosystem where humans and things can actively publish and access data to participate in diverse applications. An intelligent system orchestrates interactions, managing data, recommendations, and searching functions, while a unified interface facilitates users' input and output. The Internet acts as the backbone, connecting devices and users for seamless and real-time communication. Taking these SIoT applications into consideration, Figure 2 illustrates two different representations of SIoT architecture.

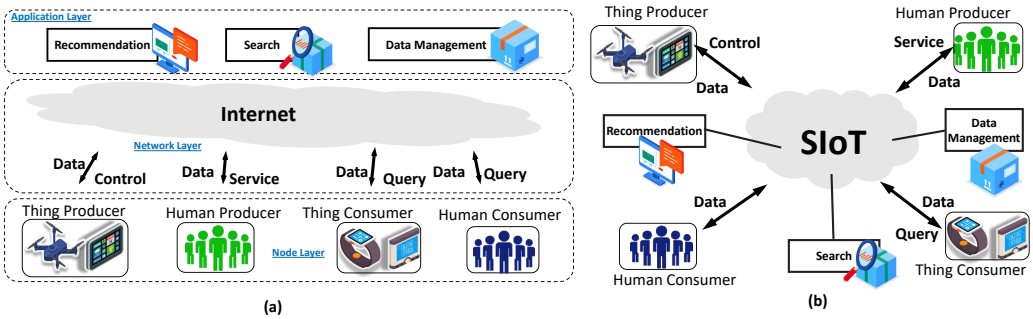

Figure 2. Two representations of SIoT architecture: (**a**) Layer-based architecture including Node, Network, and Application layers; (**b**) Role-based architecture, including Actors, Applications, and Interactions.

Recent advancements in LLMs have added a new dimension to SIoT. With robust natural language understanding and data synthesis capabilities, LLMs enable devices in SIoT to interpret complex commands, generate actionable insights, and enhance applications such as recommendation, search, and data management, as shown in Figure 3. Within this context, this survey aims to provide an application-focused analysis of applying LLMs within SIoT. By synthesizing the latest research on LLM-enhanced recommendation, search, and data management, we identify the synergies and gaps that exist at the intersection of these two emerging fields. In doing so, we not only highlight the potential of LLM-enhanced

SIoT systems but also outline the key research challenges and future directions necessary to realize their full capabilities.

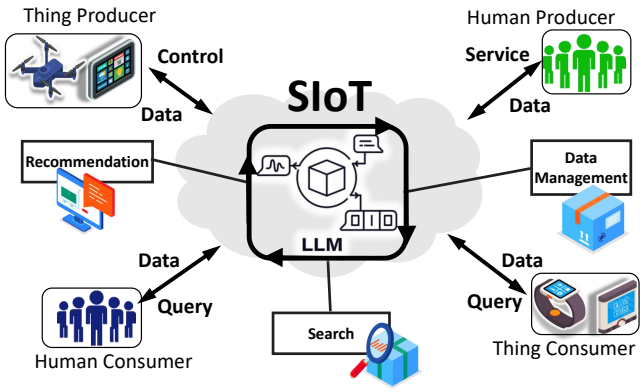

Figure 3. LLM-enhanced SIoT.

Based on Figures 2 and 3, we can notice the transition of SIoT architectures from traditional rule-based systems to LLM-enhanced intelligent SIoT systems. In Figure 2, conventional SIoT relies on static models and predefined workflows, limiting adaptability and natural language interaction. In contrast, Figure 3 shows how integrating LLMs enables devices to understand user intents, perform semantic searches, generate personalized recommendations, and manage and process data.

The remainder of this paper is organized as follows. Section 2 reviews LLMs and related technologies, providing an overview of LLM-enhanced SIoT applications. Sections 3–5 focus on detailing the technological advancements of LLM-enhanced recommendation, search, and data management applications. Section 6 provides a summary from the perspective of a keyword timeline and discusses the vulnerabilities of LLMs. Finally, Section 7 draws the conclusion.

2. Background and Motivation

2.1. Large Language Model (LLM)

LLMs have rapidly become essential in today's AI landscape, powering a wide spectrum of applications from advanced language processing [8] to strategic decision support [9]. LLMs like GPT, BERT, and their next-generation variants are valued for their capacity to perform human-like-accuracy linguistic tasks [8,10–12].

When applied to SIoT, as illustrated in Figure 3, LLMs' capabilities are powerful as they embed social intelligence directly into device networks; by translating natural-language instructions into device configurations, LLMs can significantly simplify the control of heterogeneous device communities and reduce technical barriers for users. They enrich raw sensor data with semantic context, i.e., automatically annotating activities and improving recognition accuracy, thereby enabling personalized, adaptive services across different environments [13]. Moreover, LLMs power automated task planning and workflow orchestration, optimizing resource usage and fostering collaboration among socially linked ends [14]. In essence, LLMs accelerate the transition from isolated "things" to socially empowered device ecosystems, unlocking the potential of SIoT for scalable, intelligent, and user-centered applications.

The deployment of LLMs in SIoT also relies on key supporting technologies like fine-tuning, retrieval-augmented generation, prompt engineering, and large multimodal models. These tools enhance LLM performance, adaptability, and efficiency in complex real-world SIoT scenarios.

- **Fine-tuning** tailors LLMs to specific domains by adjusting their weights on compact, task-focused datasets, boosting both relevance and accuracy without full model retraining [8]. Techniques such as adapter tuning and low-rank adaptation (LoRA) embed lightweight trainable layers into the existing network, preserving the original model's strengths while refining it for new applications [15,16].
- **Prompt engineering** serves as the cornerstone for optimizing LLM performance, particularly when tasks require intricate, multi-step reasoning. Techniques such as the chain of thought (CoT) prompting guide models through sequential logical steps, substantially enhancing their capacity for human-like reasoning [17]. Building on this foundation, frameworks like the tree of thought (ToT) [18] and graph of thought (GoT) [19] further structure problem-solving by organizing potential solution paths into hierarchical trees or interconnected graphs.
- Initially, **Retrieval-augmented generation** (RAG) was designed to fill gaps in a model's pretrained knowledge base, enabling the generation of up-to-date and deeply relevant responses [20]. In SIoT applications, RAG-based solutions fuse real-world social data into the inference process, significantly boosting the LLMs' accuracy and responsiveness [21].
- **Large multimodal models** (LMMs) extend LLMs by incorporating additional modalities such as images, audio, and video, enabling richer understanding and reasoning across multiple types of input [22]. In terms of processing data from socialized devices, LMMs can significantly enhance system intelligence by allowing devices to perceive and interpret multimodal data from their environment, e.g., recognizing visual cues, processing voice commands, or analyzing sensor patterns [23]. Thus, such capability supports more human-like interaction and contextual awareness in SIoT.

To better understand the shift in Social IoT system design, here we first provide a lightweight comparison between models using traditional Machine Learning (ML)/Deep Learning (DL) and LLMs, as shown in Figure 4.

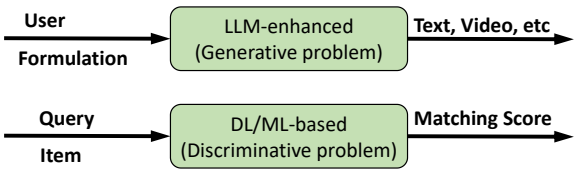

Figure 4. Comparison of ML-/DL-based and LLM-enhanced paradigms.

2.2. ML and LLMs

Traditional ML/DL-based approaches treat tasks as discriminative problems, primarily centered on computing similarity and matching scores between queries and candidate items. In contrast, LLM-enhanced approaches redefine these tasks as generation problems, aiming to produce relevant content directly conditioned on user input. This paradigm shift introduces greater flexibility and adaptability, enabling systems to handle open-ended queries and generate context-aware outputs. The detailed and in-depth comparison of these paradigms within SIoT applications is presented in Sections 3–5.

2.3. LLM-Enhanced SIoT Applications

As illustrated in Figure 3, several SIoT applications can be enhanced by LLMs.

In SIoT, recommendation, search, and data management play crucial roles in enabling intelligent, autonomous collaboration among socialized devices and users. Recommendation systems help devices identify reliable peers or services [24]. These systems evaluate past behavior, context (e.g., location or capabilities), and social relationships, which is essential in large, dynamic scenarios like smart society. As explained in [25], accurate

recommendations improve collaboration and security. LLMs could enhance these systems by providing deeper contextual reasoning, interpreting unstructured logs, and generating explainable trust decisions. Search applications allow SIoT devices and users to effectively discover services, data, or other devices. IoT search engines (IoT-SEs), described in [26,27], tackle challenges like scalability, metadata indexing, and heterogeneous data sources. An efficient search improves resource utilization and service discovery in dense IoT environments. LLMs can strengthen search systems by enabling natural language querying, enhancing semantic search, and generating context-aware responses, making complex searches more accessible to users and devices [28]. Finally, data management ensures the acquisition, processing, integration, and distribution of vast data streams across devices. As discussed in [6,29], robust data management is vital for maintaining data quality, interoperability, and processing efficiency in real-time SIoT applications. In this regard, LLMs offer the potential to automate tasks like data generation, summarization, and semantic integration, improving the usability of SIoT data management.

2.4. Related Work

The comparison between related work is summarized in Table 1. Ref. [30] focuses on LLM applications in wearable sensor-based health monitoring and activity recognition, offering limited insights into SIoT environments. Ref. [31] investigates LLM integration with general IoT systems and architectures like edge, fog, and cloud but does not address social dynamics or application-specific concerns of SIoT. Ref. [6] provides a conceptual review of the SIoT paradigm, emphasizing architecture and social relationships, yet without considering LLMs or generative models. Similarly, ref. [5] surveys SIoT components and challenges such as trust and service management, but omits recent advances in AI, particularly LLMs. As can be noticed, while these prior works have reviewed either SIoT or the application of LLMs in SIoT settings, most of them either insufficiently address the role of LLMs or do not focus specifically on the unique characteristics and challenges of SIoT. In this regard, our work offers a uniquely application-driven and SIoT-centric perspective that captures the evolutionary transition from traditional machine learning-based approaches to LLM-enhanced SIoT systems. Furthermore, our work extensively discusses vulnerabilities associated with LLM integration in SIoT, including risks such as prompt injection, model hallucination, data poisoning, and privacy leakage, which are often overlooked in existing surveys.

Table 1. Summary of related work.

Ref.	Focus Area	LLM Integration	Application	Vulnerability
[30]	Wearable Sensor-based	✓	Wearable healthcare use cases	∼
[31]	General IoT	✓	Industrial and edge IoT examples	∼
[6]	SIoT	-	SIoT architecture, relationships	✓
[5]	SIoT	-	General SIoT features and models	-
Our work	SIoT	✓	Recommendation, search, data management	✓

✓ = addressed; ∼ = partially addressed.

2.5. Scope and Methodology

This survey focuses on the use of LLMs to enhance three core application areas within SIoT: recommendation, search, and data management. The article selection and analysis process was conducted using a structured, multi-phase methodology inspired by the PRISMA review protocol. This ensured a rigorous, systematic exploration of how LLMs are integrated into SIoT contexts, reflecting both recent advances and ongoing challenges in this interdisciplinary space. In the first stage, we conducted a comprehensive literature

search across four major academic databases: IEEE Xplore, ACM Digital Library, Web of Science, and Google Scholar. The initial search was conducted in September 2024 and updated in February 2025 to capture recent advancements. Keywords were used in various combinations, as summarized in Table 2. To expand the coverage, we also examined articles citing or cited by the initially identified works. Duplicate versions were consolidated to avoid redundancy. The second stage involved abstract-level screening of the identified articles to assess their relevance to LLM applications in SIoT. We evaluated each article based on (i) the technical depth of the LLM architecture or usage, (ii) its focus on SIoT-specific functionalities, and (iii) its applicability to practical SIoT domains including social networks. Articles that mentioned LLMs or SIoT only in passing or focused solely on theoretical discussion without implementation relevance were excluded. In the final stage, we performed a full-text review of the selected articles to extract detailed concepts, challenges, and solution patterns relevant to LLM integration in SIoT. This included identifying recurring themes such as privacy-aware reasoning, human–device collaboration, or socially contextualized decision-making. An inductive analysis was used to organize findings without imposing rigid categories, enabling a richer understanding of the diverse ways LLMs are being applied in socially aware IoT environments. Some articles, while not explicitly targeting SIoT, were retained due to their strong relevance in adjacent domains like social networks and feedback-aware systems. Ultimately, the curated body of literature is categorized into three main application areas, i.e., recommendation, search, and data management, which structure the analysis and discussion in this review.

Table 2. Keyword groups and search terms used for LLM_SIoT article selection.

No.	Group	Keywords and Phrases
1	SIoT Core	"Social IoT" OR "SIoT" OR "Social Internet of Things"
2	LLM	"Large Language Models" OR "LLMs" OR "Large Models"
3	ML/DL	"Machine Learning" OR "Deep Learning" OR "Neural Networks"
4	Recommendation	"Recommendation system" OR "Recommender system"
5	Search	"Semantic search" OR "Information retrieval" OR "Natural language search" OR "question answering"
6	Data	"Data processing" OR "Data management" OR "Database"
7	Related Domains	"Social networks" OR "Social Media" OR "Social computing"

3. Recommendation

As illustrated in Figure 5, LLM-enhanced recommendation is a new paradigm where LLMs generate personalized item suggestions as natural language outputs rather than ranking existing items. By converting user profiles, histories, and contexts into text prompts, these models can directly produce item identifiers or descriptions tailored to individual needs. In SIoT environments, smart devices interact autonomously, and generative recommendation enables natural, context-aware, and conversational interfaces, allowing devices to adapt dynamically to user preferences and situations. Its zero-shot and multimodal capabilities make it especially suited for diverse, evolving IoT scenarios, fostering smarter and more collaborative device ecosystems. In this section, we first take a look at ML- and DL-based works and then focus on LLM-enhanced ones. All works of recommendations are summarized in Table 3, and a more comprehensive comparative analysis of these works is performed in Table 4 in terms of cost, efficiency, QoS, and accuracy.

Table 3. Evolution of methods from ML-/DL-based to LLM-enhanced recommendation applications.

Year	Ref.	Method/Model	Feature
2009	[32]	Matrix Factorization	Latent factor models for rating prediction and recommendation
2003	[33]	Collaborative Filtering	Scalable real-time recommendation system at Amazon
2001	[34]	Collaborative Filtering	Item similarity-based recommendation for scalability and quality
2019	[35]	BERT	Sequential recommendation using bidirectional transformer representations
2018	[36]	CNN	Top-N sequential recommendation via convolutional sequence embedding
2019	[37]	GRU+CNN+Attention	Explainable recommendations via user modeling and attentive review integration
2018	[38]	Transformer-based	SASRec: Self-attentive sequential recommendation with adaptive focus
2024	[39]	ChatGPT	Agent4Rec simulates user behavior using generative agents for recommendation testing
2024	[40]	LLaMA	LLMCRS coordinates sub-task resolution and dialogue generation in CRS
2023	[41]	LLaMA	GenRec uses LLMs to directly generate item recommendations from user context
2024	[42]	Flan-T5-XL	InstructRec frames recommendation as instruction following with instruction tuning
2024	[43]	GPT-4V, LLaVA-7B/13B	Rec-GPT4V employs large vision–language models for multimodal recommendation
2023	[44]	GPT-4/ChatGPT	LLMs evaluated as conversational recommenders on real Reddit data
2024	[45]	GPT-4/ChatGPT	LLMRG generates reasoning graphs to enhance interpretability in recommendations
2024	[46]	GPT-4	InteRecAgent integrates LLMs with traditional tools for interactive recommendation
2024	[47]	LLaMA	Llama4Rec integrates conventional models via mutual augmentation

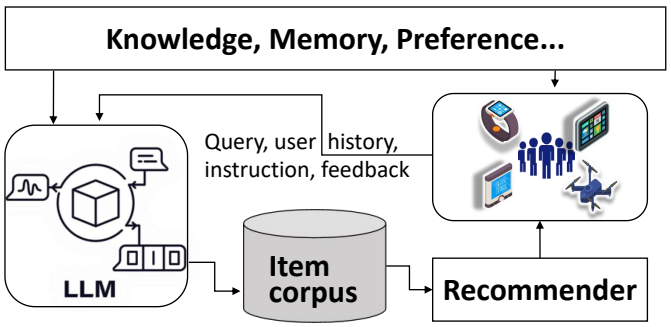

Figure 5. LLM-enhanced recommendation.

3.1. ML- and DL-Based Recommendation

An early study [34], introduced item-based collaborative filtering to improve scalability and efficiency over user-based methods. However, it struggled with data sparsity and lacked adaptability to evolving user behavior. Building on similar principles, ref. [33] proposed Amazon's item-to-item collaborative filtering, which enabled scalable, real-time recommendations but did not account for temporal dynamics in user preferences. A significant advancement came with [32], which popularized matrix factorization as a latent factor model for collaborative filtering. This method allowed for highly accurate predictions, particularly showcased in the Netflix Prize, yet still suffered from cold-start issues and lacked interpretability. As the limitations of traditional models became more evident, researchers began incorporating deep learning techniques to better capture user behavior over time. This shift was exemplified by [36], which introduced Caser, a convolutional sequence embedding model that modeled union-level and skip behaviors in user sequences. While effective at capturing short-term patterns, Caser was limited in modeling long-range dependencies. To address this, ref. [38] presented SASRec, a self-attention-based model inspired by the transformer architecture. SASRec balanced long- and short-term dependencies and adapted well across sparse and dense datasets, though it risked overfitting on extremely sparse data. One year later, ref. [37] proposed DER, a dynamic explainable recommender system that integrated time-aware gated recurrent units with an attention mechanism to provide personalized, state-aware explanations. On the other hand, its complexity and reliance on review data may affect its general applicability. Lastly, ref. [35] developed BERT4Rec, which applies a bidirectional Transformer and the Cloze task to model user behavior sequences. This approach reaches high-level performance by leveraging context from both past and future interactions but comes with high computational costs and challenges in aligning training and inference objectives.

Table 4. Detailed Comparison of ML-/DL-, and LLM-based recommendation applications.

Ref.	Cost	Efficiency	QoS	Accuracy
[32]	Moderate (SVM training and feature engineering)	Good (optimized kernel and margin selection)	Moderate (sensitive to feature noise)	Competitive (effective for simple queries)
[33]	Moderate (manual feature curation costs)	Good (efficient hand-crafted ranking pipelines)	Moderate (depends on feature quality)	Higher (with well-tuned features)
[34]	Moderate (ranking optimization overhead)	Good (fast scoring methods for retrieval)	Moderate (balanced for ad-hoc queries)	Competitive (optimized listwise training)
[35]	Moderate (lightweight compression improves cost)	Good (sparse attention helps efficiency)	Good (consistent performance under load)	Good (tuned through lightweight retraining)
[36]	High (large CNN with dense parameters)	Low (sequential bottlenecks in CNN layers)	High (excellent matching quality)	Good (effective at local matching)
[37]	Moderate (offloading strategies help balance cost)	Good (dynamic task scheduling)	Good (stable across varying loads)	Good (maintained under dynamic load)
[38]	High (full model inference on edge devices)	Good (token filtering reduces computation)	High (designed for scalable inference)	Maintained (even with early exit methods)
[39]	Moderate (memory-intensive processing)	Moderate (model pruning needed for speed)	Moderate (query complexity affects output)	Moderate (affected by over-pruning)
[40]	Moderate (attention-based retrieval modules)	Good (attention mechanisms tuned for speed)	Good (balanced generalization and specificity)	Good (balances general and specific queries)
[41]	Moderate (structured interaction models)	Good (stable in controlled environments)	Moderate (effective for narrow tasks)	Good (semantic match at structure level)
[42]	High (large-scale multi-pass ranking)	Moderate (bottlenecked by query expansion)	Moderate (fragile under noisy queries)	Moderate (good for shallow retrieval)
[43]	High (complex neural model with fine-tuning)	Moderate (trade-off between precision and latency)	High (high relevance in dense retrieval)	High (top-tier retrieval scores)
[44]	Moderate (efficient CNN designs)	Good (streamlined convolution operations)	Moderate (solid for local matching tasks)	Moderate (effective in local semantic capture)
[45]	Moderate (cost due to feature generation)	Moderate (relies on external pre-processing)	Moderate (less robust to domain shift)	Moderate (moderate semantic fidelity)
[46]	Moderate (modular lightweight components)	Good (task modularization improves execution)	Good (adapts across multiple scenarios)	Good (high precision with multi-tasking)
[47]	High (deep retrieval and re-ranking cost)	Moderate (depth increases retrieval latency)	Moderate (overfitting risk on small datasets)	High (deep architectures improve precision)

3.2. LLM-Enhanced Recommendation

Traditional ML- and DL-based recommendation systems have made significant progress but still face limitations in adaptability and reasoning. Recently, the emergence of LLMs has opened new opportunities for building more flexible, interactive, and intelligent recommender systems, i.e., LLM-driven innovations. Ref. [44] introduced a foundational idea by showing that LLMs, even in a zero-shot setting, can outperform fine-tuned conversational recommenders, highlighting their generalization and prompting capabilities but also revealing evaluation biases in existing datasets. Following this, ref. [42] proposed InstructRec, casting recommendation as an instruction-following task, where user needs are verbalized in natural language and processed by instruction-tuned LLMs, an approach that improves user interaction but depends heavily on instruction quality and task alignment. In this regard, ref. [41] also developed GenRec, a generative recommendation model that utilizes LLMs to directly generate item recommendations based on text prompts, offering enhanced flexibility but limited scalability in massive item spaces. Meanwhile, ref. [40] tackled the complexity of multi-step interactions in conversational recommendation via LLMCRS, orchestrating sub-task management and execution through LLMs, requiring reinforcement tuning for optimal performance.

Building on hybrid reasoning, ref. [46] introduced InteRecAgent, a modular agent framework combining LLMs with traditional recommenders, bridging general language understanding with domain-specific knowledge using tools and memory mechanisms. Similarly, ref. [47] proposed Llama4Rec, a mutual augmentation framework that synergizes LLMs and conventional models through data and prompt augmentation, enhancing robustness but requiring careful aggregation strategies. Extending beyond text, ref. [43] presented a multimodal recommendation system, Rec-GPT4V, leveraging vision–language models to interpret item images and summaries, although facing challenges in handling noisy visual data. A more logic-driven approach was proposed by the authors of [45], where LLMs were used to construct personalized reasoning graphs that encode user interests through causal chains, offering interpretability but still introducing computational complexity. Lastly, ref. [39] envisioned a generative agent ecosystem with Agent4Rec, where thousands of LLM-based agents simulate human behavior in recommendation environments, an innovative simulation tool for recommendation testing.

3.3. Observation

The evolution of intelligent recommendation from ML-/DL-based to LLM-enhanced is illustrated in Figure 6. Early recommendation systems were primarily based on classical machine learning techniques such as collaborative filtering and matrix factorization, as seen in [32,33]. These methods enabled scalable user–item matching but could not model contextual factors or adapt to dynamic user behavior. The field progressed with the adoption of deep learning approaches [36–38], introducing CNNs and RNNs to better capture sequential patterns in user interactions. Models like GRU4Rec and Caser improved recommendation quality but still faced limitations in generalization and interpretability, especially in multimodal environments.

The recent emergence of LLM-enhanced recommendation models marks a significant leap in capability and design. As highlighted in [48], models like InstructRec and GenRec reframe recommendation tasks as language modeling problems, enabling systems to process natural language queries, generate recommendations directly, and reason over user preferences in zero-shot or few-shot settings. This transition reflects a broader shift from fixed, feature-engineered pipelines to flexible, generative systems that can better adapt to diverse recommendation scenarios, especially within dynamic, socially-aware environments like SIoT.

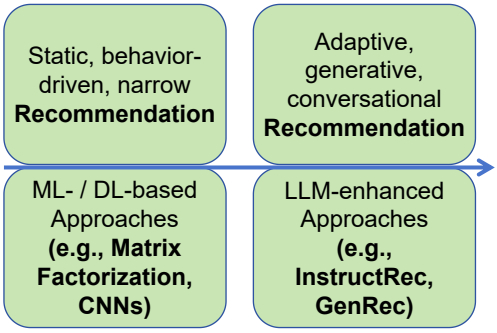

Figure 6. Recommendation applications from ML-/DL-based to LLM-enhanced.

3.4. Challenges and Future Directions

Despite the above advances in applying LLM in recommendation, a number of challenges still exist. Here, we focus on discussing several future directions for LLM-enhanced recommendation.

One major concern in the current LLM-enhanced recommendation application is hallucination [49,50], where LLMs produce plausible but incorrect outputs, which are especially risky in high-stakes domains like healthcare. To mitigate this, future recommendation systems should integrate factual knowledge sources (e.g., knowledge graphs) during training or inference and adopt output verification mechanisms to ensure actuality. Another pressing direction is building trustworthy LLMs [51], focusing on safety and privacy. LLMs are prone to adversarial inputs, calling for robust training and prompt sanitization. LLMs' black-box nature limits transparency, so enhancing explainability through interpretability tools is essential. Privacy preservation must also be prioritized through federated learning or privacy-aware tuning, given LLMs' reliance on sensitive data [52]. Additionally, domain-specific LLMs tailored for fields like healthcare or finance (e-commerce) offer better performance by aligning with domain knowledge, e.g., applying RAG [53], making recommendations more accurate and relevant.

4. Search

Search, also referred to as information retrieval (IR), is a core field that efficiently finds relevant information from large repositories in response to user queries. Traditionally embodied by web search engines like Google, IR systems match and rank content, such as text, images, or even documents, based on their relevance to a query. Over time, IR has evolved from basic keyword-based methods like Boolean logic and vector space models to advanced neural models that capture contextual and semantic nuances. With the rise of LLMs, as shown in Figure 7, these systems have gained enhanced capabilities in understanding, generating, and synthesizing information, allowing for more intelligent and conversational search experiences. Unlike recommendations, searches use short queries and sparse interactions, emphasizing semantic matching, while recommendation involves rich user formulations and dense interactions, prioritizing collaborative patterns over content semantics. In this section, we also review ML- and DL-based works before looking at LLM-enhanced ones. They are all summarized in Table 5, and similar to the review of recommendation applications, a more comprehensive comparative analysis of search-related works is performed in Table 6.

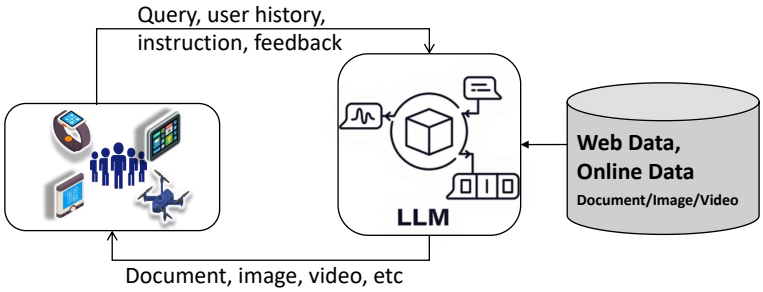

Figure 7. LLM-enhanced search.

Table 5. Evolution of methods from ML-/DL-based to LLM-enhanced in search applications.

Year	Ref.	Method/Model	Feature
2004	[54]	SVM, MaxEnt	Discriminative models for relevance classification in information retrieval
2009	[55]	Point/Pair/Listwise	Categorization of learning to rank methods in information retrieval
2016	[56]	LSTM-RNN	Deep sentence embedding for semantic similarity and document retrieval
2015	[57]	CNN	CNN architectures for sentence-level semantic matching
2014	[58]	CNN	Convolutional Latent Semantic Model for contextual web document ranking
2024	[59]	ChatGLM	Web navigation agent that surpasses GPT-4 using curriculum training and RLHF
2024	[60]	GPT-4V	Multimodal web agent executing end-to-end tasks via visual and text input
2024	[61]	GPT-4, GPT-3.5	Internet of Agents framework for collaborative multi-agent systems
2024	[62]	GPT-4o	MindSearch mimics human web search for multi-agent information integration
2024	[63]	BART-type	LTRGR enables generative retrieval to directly learn to rank
2024	[64]	BART-large	DGR enhances generative retrieval via knowledge distillation
2023	[65]	BART-large	Multi-view identifiers boost retrieval performance in generative systems
2024	[66]	ChatGPT	CoSearchAgent supports collaborative searching in messaging platforms
2023	[67]	Chinese PLMs	Interactive LFQA for Chinese using web search and synthesis models
2024	[68]	GPT-4	TRAD improves decision-making via step-wise thought retrieval

4.1. ML- and DL-Based Search

Similarly, the development of search applications has progressively advanced from machine learning-based to deep-learning-based. In 2004, the authors of [54] explored the use of discriminative models, particularly support vector machine searches, showing that SVMs could rival generative models by effectively learning from arbitrary features, though at the cost of requiring substantial feature engineering. Building on this, ref. [55] provided a foundational overview of learning-to-rank techniques, categorizing them into pointwise, pairwise, and listwise approaches. While reputable and influential, this work relied heavily on hand-crafted features and traditional feature extraction pipelines. Transitioning into the deep learning era, ref. [58] introduced the convolutional latent semantic model (CLSM), using a convolution-pooling architecture to capture rich contextual semantics in query–document pairs. Although showing effectiveness, CLSM's local n-gram focus limited its ability to model long-range dependencies. Furthermore, ref. [57] proposed CNN architectures for sentence matching, offering strong performance across matching tasks but similarly restricted by fixed receptive fields. The study presented in [56] developed a deep LSTM-RNN sentence embedding model trained on click-through data, capable of capturing sequential dependencies and automatically focusing on salient words. Despite this advance, the model required weak supervision and lacked scalability to longer documents.

Table 6. Detailed comparison of ML-/DL- and LLM-based search applications.

Ref.	Cost	Efficiency	QoS	Accuracy
[54]	Moderate (multi-view embeddings increase compute slightly)	Good (sparse retrieval efficient)	Good (multi-view robust search)	High (boosts dense retrieval models)
[55]	Moderate (knowledge distillation needed)	Good (multi-teacher setup enhances retrieval)	Good (knowledge transfer helps robustness)	Good (higher MRR scores)
[56]	High (generative ranking model training)	Moderate (two-phase training slows process)	High (direct ranking objective alignment)	High (outperforms dense retrievers)
[57]	High (multimodal input handling)	Moderate (multimodal fusion overhead)	Good (robust web navigation)	Good (strong benchmark results)
[58]	High (curriculum + reinforcement learning)	Moderate (open-domain variability affects speed)	Good (adaptive policy for different tasks)	High (state-of-the-art web benchmarks)
[59]	Moderate (collaborative retrieval adds complexity)	Moderate (multi-agent message passing)	Good (team-based query refinement)	Good (robust collective retrieval)
[60]	High (search across 100s of pages needs scaling)	Moderate (long-context needs tuning)	Good (parallel searching improves planning)	High (cognitive simulation enhances results)
[61]	Moderate (noise-tolerant agent navigation)	Good (aligned demonstrations boost stability)	Good (decision-making under noise)	Good (benchmark improvements)
[62]	High (open-ended multi-agent Internet search)	Moderate (complex messaging system)	Good (scalable teaming architecture)	Good (prototype shows promise)
[63]	Moderate (Slack platform dependency)	Good (lightweight agent for collaboration)	Good (real-time collaborative QA)	Moderate (depends on platform reliability)
[64]	Moderate (large web screenshot processing)	Moderate (HTML + image multimodal fusion)	Good (real-world task completion)	Good (benchmarks on real websites)
[63]	Moderate (suboptimal page navigation overhead)	Moderate (curriculum policy needed)	Moderate (versatility struggles)	Good (good closed benchmark results)
[66]	Moderate (lightweight retrieval agents)	Good (flexible multi-agent collaboration)	Good (dynamic agent teaming)	Moderate (prototype-level so far)
[67]	Moderate (scalable messaging)	Good (low-overhead team communication)	Good (prototype shows scalability)	Moderate (real-world implementation pending)
[68]	Moderate (batch processing efficiencies)	Good (processing optimization effective)	Good (batch search scalability)	Good (strong page-level recall)

4.2. LLM-Enhanced Search

While traditional ML- and DL-based methods have advanced search systems by improving semantic matching and ranking, they still face limitations in context understanding and adaptability. Motivated by these gaps, LLM-enhanced searches have been progressing rapidly in recent years, with each new system tackling distinct challenges across web navigation and generative and collaborative information retrieval.

Ref. [63] proposed that LTRGR should directly align generative training with ranking objectives, but its two-phase training pipeline remains a hurdle for end-to-end optimization. Building on this work, ref. [65] proposed multi-view identifiers for generative retrieval, combining synthetic, title, and substring identifiers to represent passages from multiple perspectives. While effective in improving ranking robustness, it still inherits the limitations of intermediate identifier generation. In [64], also following the work in [63], the authors introduced DGR, using knowledge distillation from cross-encoders to improve passage ranking in generative information retrieval. This approach bridges learning gaps but depends on high-quality teacher models.

As for the web navigation field, ref. [60] presented WebVoyager, a large multimodal model-based agent capable of end-to-end task completion on real websites using both screenshots and HTML. It enhances realism in evaluation but faces the complexity of multimodal understanding. Similarly, ref. [59] introduced AutoWebGLM, an open LLM-based web agent trained in curriculum learning and reinforcement learning. It performs strongly in benchmark tasks yet struggles with the versatility and ambiguity of open-domain web actions. Meanwhile, ref. [68] developed TRAD, a framework for step-wise in-context retrieval in LLM agents, improving decision quality through aligned and noise-reduced trajectory selection; its limitation lies in managing imperfectly retrieved demonstrations. In addition, ref. [62] proposed MindSearch, a multi-agent search framework that mimics human cognitive planning by decomposing queries and coordinating parallel searches across hundreds of pages. Although powerful in scale and performance, it requires complex orchestration and long-context handling.

Furthermore, interactive and collaborative searches have also gained attention. Ref. [67] introduced WebCPM, the first Chinese interactive long-form QA dataset and framework, offering human-like search simulations but limited to Chinese-language applications. Another work [66], proposed CoSearchAgent, a lightweight collaborative LLM agent embedded in Slack, enabling multi-user real-time search conversations, though it remains constrained by platform integration and agent coordination. Finally, ref. [61] conceptualized the Internet of Agents (IoA), a flexible framework for multi-agent collaboration inspired by the Internet, introducing a messaging-based architecture for scalable and dynamic teaming but remaining at the prototype stage for real-world implementation.

4.3. Observation

Figure 8 shows the evolution from ML-/DL-based searches to LLM-enhanced searches. Early search applications were largely grounded in machine learning techniques like support vector machines (SVM) and learning-to-rank frameworks, which required substantial feature engineering and domain expertise, as shown in [54,55]). These systems enabled basic query–document matching but lacked semantic depth and adaptability. The introduction of deep learning shifted this paradigm, incorporating CNNs and RNNs for sentence-level semantic matching [56–58]. Models such as LSTM-RNNs and the convolutional latent semantic model (CLSM) improved context awareness but still struggled with long-range dependencies and generalization. This foundation paved the way for the rise of LLMs in searches. As shown in [60,67], LLM-enhanced search systems such as ChatGLM, GPT-4V, and CoSearchAgent leverage generative reasoning and multimodal

understanding to support natural language querying, semantic retrieval, and collaborative web navigation. These systems are capable of interpreting queries in open-ended, zero-shot settings and synthesizing answers from large, dynamic corpora, which are capabilities that traditional ML/DL approaches could not support. This transition from structured matching to generative reasoning marks a pivotal advancement, especially for complex and personalized searches in SIoT environments.

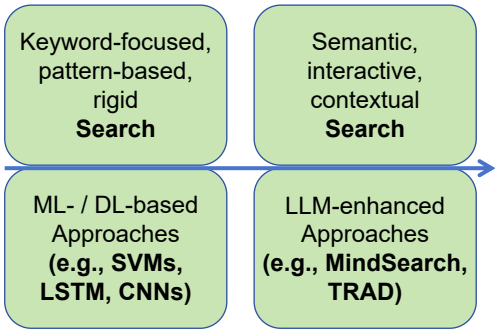

Figure 8. Search applications from ML-/DL-based to LLM-enhanced.

4.4. Challenges and Future Directions

In order to address the current limitations and unlock the full potential of generative capabilities, one main direction is developing unified search agents [48] that integrate query rewriting, retrieval, reranking, and response generation into end-to-end, conversational systems that mimic human search behaviors. Another challenge lies in designing robust, semantically meaningful document identifiers, such as multi-view or codebook-based identifiers [69,70]. Additionally, enabling generative search models to handle trade-offs between cost/efficiency and performance remains a major open problem [71].

5. Data Management

Data management involves collecting, sharing, and integrating heterogeneous data from numerous socialized devices and users. In this regard, as can be seen in Figure 9, LLMs can enhance such applications by enabling intelligent data transformation, integration, and interpretation through natural language understanding and reasoning, making SIoT systems more adaptive and context-aware. Similar to the previous two sections, we review ML- and DL-based works before LLM-enhanced ones, and we also summarize related works in Table 7. Similarly, we compared these works in Table 8.

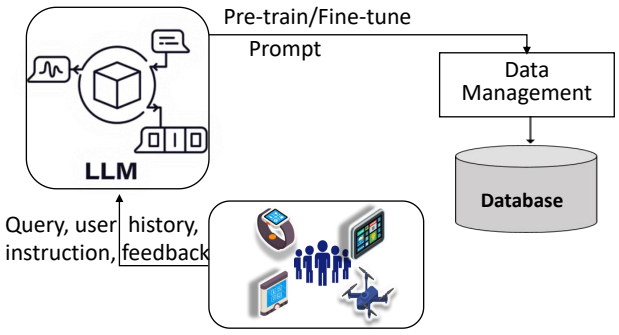

Figure 9. LLM-enhanced data management.

Table 7. Evolution of methods from ML-/DL-based to LLM-enhanced in data management applications.

Year	Ref.	Method/Model	Feature
2007	[72]	MLR + Bayesian Networks	Forecasting queries over time-series data
2010	[73]	Belief Propagation	Statistical data cleaning in relational databases
2013	[74]	Factorization Machines	Relational predictive modeling with high-cardinality data
2016	[75]	Delta Encoding	Lifecycle management for deep learning models
2016	[76]	DL System Optimization	System-level optimization for deep learning training
2017	[77]	A* Algorithm	Programming-by-example for data transformation (Foofah)
2020	[78]	BERT+table pretraining	Pretrained model for joint understanding of text and tabular data
2021	[79]	BERT+GPT hybrid	Relational pre-trained Transformer for data preparation automation
2023	[80]	No specified	Multi-step transformation synthesis to relationalize tables (Auto-Tables)
2023	[81]	GPT-3	Hybrid SQL querying over LLMs and databases
2023	[82]	GPT-3/3.5+FlanT5	Vision for hybrid LLM-DBMS systems for data access and integration
2023	[18]	GPT-4, PaLM	Tree of Thoughts: deliberate planning framework for reasoning with LLMs
2023	[83]	No specified	Lessons from data integration systems to inform augmented LLM research
2024	[84]	GPT-3.5	LLM framework for query rewrite, index tuning, and database optimization
2024	[85]	GPT-4	Automated database diagnosis for root cause analysis and optimization

5.1. ML- and DL-Based Data Management

In an early study [72], the Fa system was introduced with declarative forecasting queries combining data transformation, statistical modeling, and adaptive inference. This work was limited by its reliance on manual model selection and lacked support for unstructured or complex data while effective for structured time-series forecasting. Advancing this direction, ref. [73] presented ERACER, a statistical inference and data-cleaning framework based on belief propagation within relational databases. It enabled automated imputation and correction using relational dependency networks but remained computationally intensive and somehow constrained by approximate inference techniques. Later, the authors of [33] addressed the scalability challenge of learning over relational data by proposing algorithms for efficient training factorization machines with high-cardinality features. Although this model retained the traditional feature-based modeling paradigm, it showed how to extend ML methods to more complex data settings. The development of deep learning and related techniques, as seen in [75], introduced ModelHub, a system for managing the deep learning model lifecycle, including model versioning, parameter archiving, and exploration. On the one hand, it addressed model reproducibility and storage challenges. On the other hand, it did not extend to full training integration or dynamic data updates. Similarly, ref. [76] bridged databases and deep learning-based methods by identifying opportunities for using database principles (e.g., memory management and operation scheduling) to optimize DNN training systems and exploring how deep learning could enhance probabilistic tasks like knowledge fusion in databases. However, this work lacked concrete implementations, focusing more on outlining cross-disciplinary potential than delivering system-level tools.

Table 8. Detailed comparison of ML-/DL- and LLM-based search data management applications.

Ref.	Cost	Efficiency	QoS	Accuracy
[72]	Moderate (manual modeling effort)	Good (structured forecasting)	Good (structured, time-series focused)	Moderate (manual tuning needed)
[73]	Moderate (automated data cleaning)	Good (automated statistical inference)	Good (relational inference)	Good (belief propagation aids correction)
[74]	Moderate (early feature engineering models)	Good (feature-based scalability)	Moderate (general prediction tasks)	Good (feature-rich models)
[75]	High (training Transformer models)	Moderate (task-specific finetuning needed)	Good (table-semantics parsing)	High (transformer-based learning)
[76]	High (pretraining and tuning requirements)	Moderate (large models slow reasoning)	Good (query optimization and rewriting)	High (full LLM with database tuning)
[77]	Moderate (program synthesis overhead)	Good (example-driven automation)	Good (multi-step transformations)	Good (program synthesis achieves generalization)
[78]	Moderate (complex operator sets)	Good (auto-synthesis, minimal human intervention)	Good (handling non-relational tables)	Good (no example dependency boosts transferability)
[79]	High (joint modeling text and tables)	Moderate (scale limitations on large tables)	Good (semantic understanding boost)	Good (contextual retrieval precision)
[80]	Moderate (tuple-to-X task finetuning)	Good (pretrained generalizability)	Good (tuple matching and error resilience)	High (structured task performance)
[81]	Moderate (SQL-query based LLM interaction)	Moderate (complexity of SQL query generation)	Good (structured querying support)	Moderate (dependent on SQL formulation quality)
[82]	High (database-optimized LLM system)	Good (adaptability across tasks)	Good (adaptive database operations)	High (optimized for data management tasks)
[18]	Moderate (error diagnosis processing)	Moderate (reasoning efficiency based on task complexity)	Moderate (diagnosis system robustness)	Good (error localization accurate)
[83]	Moderate (deliberate search overhead)	Good (tree-search reduces random exploration)	Good (structured and open text reasoning)	Good (planning improves context coherence)
[84]	Moderate (integration with structured data)	Moderate (schema mapping overhead)	Moderate (inconsistent schema adaptation)	Good (provenance tracking helps reliability)
[85]	High (multi-modal cross-domain reasoning)	Moderate (scalability needs optimization)	Good (dynamic multimodal integration)	Moderate (cross-modal noise handling challenges)

5.2. LLM-Enhanced Data Management

While ML- and DL-based methods have advanced data management by improving model lifecycle handling and structured data processing, they still face limitations in automation, scalability, and reasoning. Recent work has turned to LLMs to address these challenges. Early efforts in LLM-enhanced data management, such as [77], introduced a program synthesis approach for data transformation by example, lowering the technical barrier for non-programmers but remaining limited by heuristic searches and its reliance on explicit examples. To further automate data processing, ref. [80] proposed synthesizing multi-step table transformations without user examples, successfully handling non-relational tables at scale but still constrained by a predefined operator set. Another work [78], extended BERT-style pretraining to jointly model text and tables for semantic parsing, advancing the understanding of structured and unstructured data, though facing challenges in scaling to large tables and generalizing across domains.

In terms of the transformer-based method, ref. [79] presented a unified transformer-based pretraining model for tuple-to-X tasks such as cleaning and schema matching. However, it required fine-tuning for each specific task and lacked dynamic reasoning capabilities. This gap was addressed in a later study [82], which envisioned using SQL as a precise interface to query LLMs directly, thereby bridging structured querying and unstructured knowledge, but raised concerns about factual consistency and query complexity. Expanding on this vision, ref. [84] proposed a database-optimized LLM framework capable of tasks like query rewriting and index tuning, offering strong adaptability. In addition, the authors of [85] introduced an LLM-based diagnosis system for database anomalies using reasoning strategies and collaborative mechanisms to outperform traditional tools.

Simultaneously, novel architectures like [18] were designed to enhance LLM reasoning by enabling deliberate searches over coherent text units, significantly improving planning and inference. On the integration side, ref. [83] reflected on how classical data integration systems share conceptual ground with augmented LLMs, advocating for schema reasoning, source mapping, and provenance tracking. Lastly, ref. [81] focused on handling querying over multi-modal data lakes using natural language, avoiding explicit integration via cross-modal representation learning.

5.3. Observation

Figure 10 visualizes the evolution of data management applications from ML-/DL-based searches to LLM-enhanced searches. Early works in data management applied classical machine learning techniques such as multiple linear regression and Bayesian networks to forecast structured time-series data (e.g., [73]) and belief propagation for statistical data cleaning in relational databases [72]. These methods were foundational but limited by manual feature engineering, rigid schema reliance, and lack of adaptability to unstructured or dynamic data sources. As IoT data volumes grew, deep learning approaches emerged, improving scalability and contextual understanding. For example, ref. [75,76] explored system-level optimization and lifecycle management for deep learning models, enabling better model versioning and integration with database backends. However, these approaches still required handcrafted training pipelines and lacked semantic flexibility.

The shift to LLM-enhanced data management represents a significant leap in capability. Papers like [18,84] illustrate the use of LLMs (e.g., GPT-3 and GPT-4) for SQL querying, semantic table understanding, and automated data integration. These systems leverage natural language interfaces to enable users to perform complex data queries, schema matching, and transformation tasks without the need for formal programming knowledge. Moreover, techniques such as program synthesis [77] and pretraining on text-table combinations [85] have dramatically expanded the scope of tasks LLMs can handle, progressing from entity

resolution and data cleaning to multi-step reasoning and optimization. This evolution from deterministic, model-specific methods to flexible, generative LLM frameworks underscores a paradigm shift where data management becomes more intuitive, adaptive, and capable of operating in heterogeneous SIoT environments.

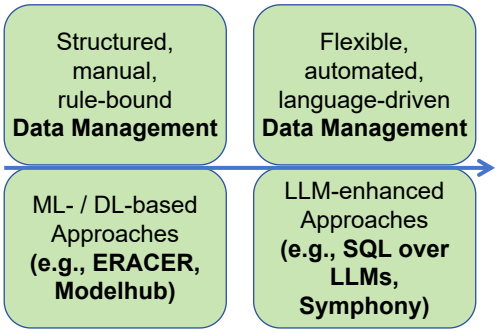

Figure 10. Data management applications from ML-/DL-based to LLM-enhanced.

5.4. Challenges and Future Directions

As LLMs increasingly interact with diverse data types, such as text and images, multimodal data management must ensure coherent integration, alignment, and retrieval across formats, yet current systems often lack standardized schema or robust fusion mechanisms [86]. Meanwhile, conflict data separation, i.e., distinguishing contradictory or noisy entries, remains difficult in large-scale datasets, where redundancy, misinformation, or inconsistent labeling can degrade model reliability if not effectively filtered or resolved [87]. Additionally, social bias in training data remains insufficiently addressed. This is because biased data or historical inequalities can lead to unfair or discriminatory LLM behavior [88].

6. Discussion

6.1. Timeline View of Collected Papers

The timeline in Figure 11 shows that early research focused on foundational machine learning techniques such as learning-to-rank and factorization machines. These methods laid the groundwork for personalized recommendation systems and efficient data organization, which are critical in SIoT, where devices continuously generate and exchange data. As IoT data grew in volume and complexity at later times, the emphasis shifted toward deep learning and database optimization. Deep learning techniques improved semantic understanding and pattern recognition in unstructured data, while database optimization ensured that vast amounts of sensor data could be stored, retrieved, and managed effectively, which is essential for real-time search and recommendation applications.

From 2020 onward, the utilization of LLMs marked a significant advancement. LLMs enable more interactive and context-aware SIoT, allowing natural language interfaces to interpret user queries and coordinate device interactions. Generative retrieval and multi-view identifiers further enhance search capabilities by allowing conversational queries and robust data association across various modalities in IoT environments.

Overall, this evolution culminates in LLM-enhanced solutions that are well-suited for SIoT applications. Novel recommendation systems benefit from the combined strengths of early factorization methods and recent generative techniques, while advanced search functions leverage natural language processing to improve information discovery. Moreover, data management strategies now integrate contextual and multimodal data, ensuring that SIoT performs updated, personalized, and efficient services for users.

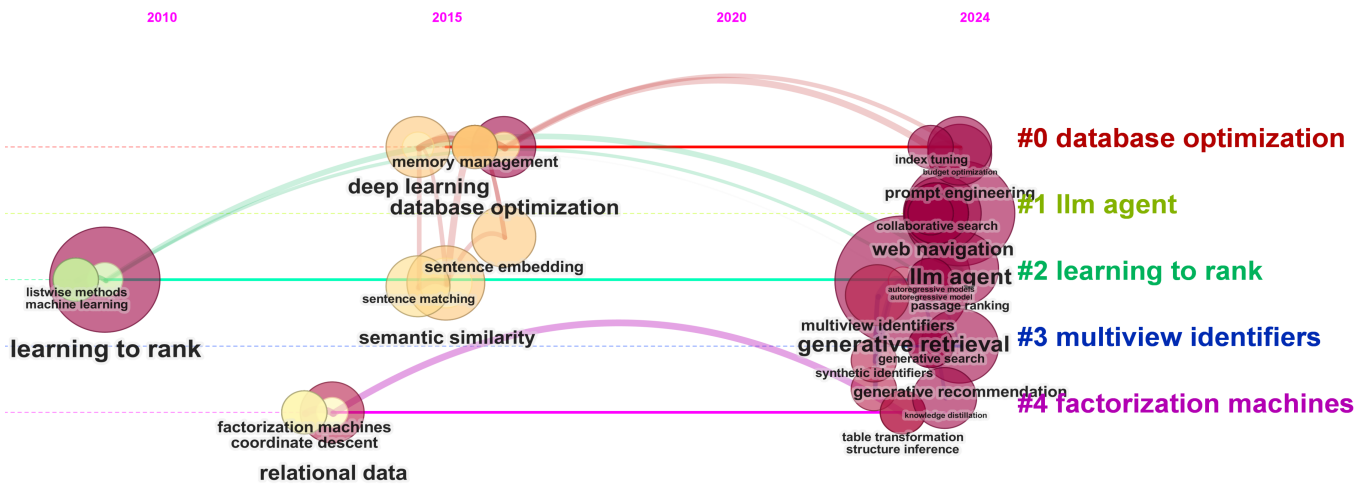

Figure 11. Timeline view of keyword clusters.

6.2. Overall Challenges and Future Directions

LLM-enhanced SIoT systems offer the potential to create more intelligent, adaptive, and user-centric services but also introduce new challenges in several aspects. Understanding these practical implications is essential for designing robust architectures, setting responsible deployment standards, and fostering sustainable innovation across socially interconnected environments.

6.2.1. Security and Privacy

LLMs face several critical vulnerabilities that can undermine their reliability and security. One major category is model poisoning and manipulation, where attackers inject a small number of malicious examples into the training data to alter the model's behavior. One study demonstrated that even as few as 100 poisoned samples can cause an LLM to consistently produce flawed outputs, and such attacks can also implant hidden Trojan triggers that remain dormant until activated [89,90]. Such manipulations not only compromise the dataset but may also affect the model's parameters' weights and internal activation layers, thereby threatening the overall integrity of the system. In addition to hallucination, model poisoning attacks pose unique threats to SIoT networks. Compromised models could manipulate recommendations, search results, or data management decisions across interconnected smart devices, as seen in the broader vulnerabilities discussed in [91]. This can be especially dangerous in critical domains like healthcare SIoT or smart city infrastructures, where incorrect outputs could cascade into large-scale failures. After the discussion of model poisoning, another critical one is malicious prompting, where adversaries manipulate inputs with carefully designed modifications, using techniques such as red teaming, special characters, or context ignoring to bypass the LLM's safety constraints. This can lead the model to generate harmful or unintended outputs, potentially triggering dangerous system behaviors in real-world SIoT applications [92]. For example, an attacker might subtly alter a prompt to manipulate the model into issuing hazardous instructions or leaking sensitive operational details. Additionally, LLMs are prone to information extraction and leakage attacks. In this kind of threat, adversaries exploit the model to recover sensitive training data, which can extract personal or confidential information, even when standard anonymization techniques are applied, posing severe privacy risks [93].

To address vulnerabilities like hallucination and data privacy risks in LLM-enhanced SIoT, combining LLMs with external knowledge sources such as knowledge graphs and adopting federated learning are promising strategies. Integrating knowledge graphs improve factual consistency, as shown by [94], reducing hallucinated outputs in device

control. Meanwhile, federated learning [95] enables decentralized model updates without exposing raw user data, preserving privacy in sensitive SIoT environments.

6.2.2. Computational Costs and Real-Time Constraints

Deploying LLMs in SIoT systems introduces significant computational and real-time challenges, especially at the edge. As highlighted in [96], LLMs like LLaMA-2 or GPT-4 demand substantial memory (e.g., over 8 GB even with FP16 compression) and computational power, exceeding the capabilities of typical SIoT nodes such as smart sensors and mobile hubs. The quadratic complexity of the self-attention mechanism further exacerbates throughput bottlenecks, making real-time inference difficult without aggressive optimization. Moreover, SIoT environments are highly heterogeneous, comprising devices with varying processing capabilities, which complicates the deployment of standard LLMs [5]. Many SIoT applications require low-latency responses (e.g., real-time social recommendations or emergency searches), yet LLM inference often introduces delays that violate real-time guarantees. Additionally, the energy and resource demands of LLMs conflict with the strict size, battery, and sustainability constraints typical of IoT devices. Without compression techniques [97], hardware acceleration [98], or intelligent edge-cloud collaboration [99], the integration of LLMs into SIoT remains computationally challenging, highlighting the need for future research on lightweight models and adaptive deployment strategies.

6.2.3. Energy Inefficiency and Sustainability

LLMs incur significant energy consumption, which is particularly problematic in decentralized IoT environments. Devices like smart sensors and wearables operate under tight energy budgets, and continuous interaction with cloud-based LLMs risks battery drain and carbon footprint issues [100,101]. Sustainability becomes a critical limitation for long-term SIoT deployment.

6.2.4. Ethical and Societal Concerns

Ethical issues such as bias propagation, lack of transparency, and erosion of user autonomy are amplified when deploying LLMs in socially intelligent settings. LLM-based recommendations or searches could inadvertently reinforce societal biases or manipulate behavior without user consent [102]. The risks of discrimination, misinformation, and fairness gaps become pronounced in heterogeneous SIoT communities.

7. Conclusions

This review provides a comprehensive and application-driven examination of the evolution from traditional machine learning and deep learning approaches to LLM-enhanced solutions within SIoT systems. By focusing on three core SIoT tasks, including recommendation, search, and data management, we illustrate how the adoption of LLMs enables more flexible, context-aware, and human-centric interactions compared to earlier static and feature-dependent methods. Our review not only summarizes the key technical advancements but also critically analyzes the challenges associated with deploying LLMs in decentralized, resource-constrained environments, such as computational overhead, latency sensitivity, and privacy concerns. Furthermore, this review highlights emerging vulnerabilities such as hallucination and model poisoning, which are particularly critical in socially connected IoT settings. We connect these vulnerabilities to SIoT-specific risks and propose actionable strategies such as knowledge graph integration, federated learning, and lightweight model adaptation to mitigate their impact. Beyond the technical discussion, this review emphasizes the broader practical implications for stakeholders involved in designing, regulating, and deploying LLM-enhanced SIoT systems. Responsible deploy-

ment must balance innovation with considerations of ethical AI, data governance, and system resilience, especially as SIoT networks scale across sensitive domains like healthcare, transportation, and smart cities. Looking ahead, future research directions should prioritize efficient and robust LLM adaptations tailored to the dynamic and decentralized nature of SIoT. Advancements in model compression, adaptive reasoning, context-aware optimization, and explainable AI will be crucial for realizing the full potential of LLMs in socialized intelligent IoT ecosystems.

Author Contributions: Methodology and validation, L.Y. and R.S.; Writing—original draft preparation, L.Y. and R.S.; Writing—review and editing, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No specific data are required for this paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
SIoT	Social Internet of Things
ML	Machine Learning
DL	Deep Learning
RAG	Retrieval-Augmented Generation
LMM	Large Multimodal Model
CoT	Chain of Thought
ToT	Tree of Thought
GoT	Graph of Thought
T2T	Thing-to-Thing
H2H	Human-to-Human

References

1. Kranz, M.; Roalter, L.; Michahelles, F. Things that twitter: Social networks and the internet of things. In Proceedings of the What Can the Internet of Things do for the Citizen (CIoT) Workshop at the Eighth International Conference on Pervasive Computing (Pervasive 2010), Helsinki, Finland, 17 May 2010; pp. 1–10.
2. Atzori, L.; Iera, A.; Morabito, G. Siot: Giving a social structure to the internet of things. *IEEE Commun. Lett.* **2011**, *15*, 1193–1195. [CrossRef]
3. Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Comput. Netw.* **2012**, *56*, 3594–3608. [CrossRef]
4. Ortiz, A.M.; Hussein, D.; Park, S.; Han, S.N.; Crespi, N. The cluster between internet of things and social networks: Review and research challenges. *IEEE Internet Things J.* **2014**, *1*, 206–215. [CrossRef]
5. Shahab, S.; Agarwal, P.; Mufti, T.; Obaid, A.J. SIoT (social internet of things): A review. In *ICT Analysis and Applications*; Springer: Singapore, 2022; pp. 289–297.
6. Malekshahi Rad, M.; Rahmani, A.M.; Sahafi, A.; Nasih Qader, N. Social Internet of Things: Vision, challenges, and trends. *Hum.-Centric Comput. Inf. Sci.* **2020**, *10*, 52. [CrossRef]
7. Dhelim, S.; Ning, H.; Farha, F.; Chen, L.; Atzori, L.; Daneshmand, M. IoT-enabled social relationships meet artificial social intelligence. *IEEE Internet Things J.* **2021**, *8*, 17817–17828. [CrossRef]
8. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K.N. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019.
9. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.

10. OpenAI. Introducing GPT-4o: Our Fastest and Most Affordable Flagship Model. 2024. Available online: https://platform.openai.com/docs/models (accessed on 26 February 2025).
11. Meta-AI. Llama 3.1: Advanced Open-Source Language Model. 2024. Available online: https://ai.meta.com/blog/meta-llama-3-1/ (accessed on 26 February 2025).
12. Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; Mian, A. A comprehensive overview of large language models. *arXiv* **2023**, arXiv:2307.06435.
13. Matwin, S.; Milios, A.; Prałat, P.; Soares, A.; Théberge, F. Survey of generative methods for social media analysis. *arXiv* **2021**, arXiv:2112.07041.
14. Sarhaddi, F.; Nguyen, N.T.; Zuniga, A.; Hui, P.; Tarkoma, S.; Flores, H.; Nurmi, P. LLMs and IoT: A Comprehensive Survey on Large Language Models and the Internet of Things. *TechRxiv* **2025**. [CrossRef]
15. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR: Cambridge, MA, USA, 2019; pp. 2790–2799.
16. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *ICLR* **2022**, *1*, 3.
17. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
18. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 11809–11822.
19. Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 17682–17690. [CrossRef]
20. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
21. Zeng, J.; Huang, R.; Malik, W.; Yin, L.; Babic, B.; Shacham, D.; Yan, X.; Yang, J.; He, Q. Large language models for social networks: Applications, challenges, and solutions. *arXiv* **2024**, arXiv:2401.02575.
22. Huang, D.; Yan, C.; Li, Q.; Peng, X. From large language models to large multimodal models: A literature review. *Appl. Sci.* **2024**, *14*, 5068. [CrossRef]
23. Du, N.; Wang, H.; Faloutsos, C. Analysis of large multi-modal social networks: Patterns and a generator. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, 20–24 September 2010; Proceedings, Part I 21; Springer: Berlin/Heidelberg, Germany, 2010; pp. 393–408.
24. Farhadi, B.; Asghari, P.; Mahdipour, E.; Javadi, H.H.S. A novel recommendation-based framework for reconnecting and selecting the efficient friendship path in the heterogeneous social IoT network. *Comput. Netw.* **2025**, *258*, 111016. [CrossRef]
25. Becherer, M.; Hussain, O.K.; Zhang, Y.; den Hartog, F.; Chang, E. On trust recommendations in the social internet of things—A survey. *ACM Comput. Surv.* **2024**, *56*, 1–35. [CrossRef]
26. Hatcher, W.G.; Qian, C.; Liang, F.; Liao, W.; Blasch, E.P.; Yu, W. Secure IoT search engine: Survey, challenges issues, case study, and future research direction. *IEEE Internet Things J.* **2022**, *9*, 16807–16823. [CrossRef]
27. Hatcher, W.G.; Qian, C.; Gao, W.; Liang, F.; Hua, K.; Yu, W. Towards efficient and intelligent internet of things search engine. *IEEE Access* **2021**, *9*, 15778–15795. [CrossRef]
28. Amin, F.; Majeed, A.; Mateen, A.; Abbasi, R.; Hwang, S.O. A systematic survey on the recent advancements in the Social Internet of Things. *IEEE Access* **2022**, *10*, 63867–63884. [CrossRef]
29. Gharaibeh, A.; Salahuddin, M.A.; Hussini, S.J.; Khreishah, A.; Khalil, I.; Guizani, M.; Al-Fuqaha, A. Smart cities: A survey on data management, security, and enabling technologies. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2456–2501. [CrossRef]
30. Ferrara, E. Large language models for wearable sensor-based human activity recognition, health monitoring, and behavioral modeling: A survey of early trends, datasets, and challenges. *Sensors* **2024**, *24*, 5045. [CrossRef] [PubMed]
31. Kök, İ.; Demirci, O.; Özdemir, S. When IoT Meet LLMs: Applications and Challenges. In Proceedings of the 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 15–18 December 2024; IEEE: New York, NY, USA, 2024; pp. 7075–7084.
32. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
33. Linden, G.; Smith, B.; York, J. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* **2003**, *7*, 76–80. [CrossRef]
34. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.

35. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1441–1450.
36. Tang, J.; Wang, K. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Los Angeles, CA, USA, 5–9 February 2018; pp. 565–573.
37. Chen, X.; Zhang, Y.; Qin, Z. Dynamic explainable recommendation based on neural attentive models. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 53–60.
38. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; IEEE: New York, NY, USA, 2018; pp. 197–206.
39. Zhang, A.; Chen, Y.; Sheng, L.; Wang, X.; Chua, T.S. On generative agents in recommendation. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024; pp. 1807–1817.
40. Feng, Y.; Liu, S.; Xue, Z.; Cai, Q.; Hu, L.; Jiang, P.; Gai, K.; Sun, F. A large language model enhanced conversational recommender system. *arXiv* **2023**, arXiv:2308.06212.
41. Ji, J.; Li, Z.; Xu, S.; Hua, W.; Ge, Y.; Tan, J.; Zhang, Y. Genrec: Large language model for generative recommendation. In Proceedings of the European Conference on Information Retrieval, Glasgow, UK, 24–28 March 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 494–502.
42. Zhang, J.; Xie, R.; Hou, Y.; Zhao, X.; Lin, L.; Wen, J.R. Recommendation as instruction following: A large language model empowered recommendation approach. In *ACM Transactions on Information Systems*; Association for Computing Machinery: New York, NY, USA, 2023.
43. Liu, Y.; Wang, Y.; Sun, L.; Yu, P.S. Rec-gpt4v: Multimodal recommendation with large vision-language models. *arXiv* **2024**, arXiv:2402.08670.
44. He, Z.; Xie, Z.; Jha, R.; Steck, H.; Liang, D.; Feng, Y.; Majumder, B.P.; Kallus, N.; McAuley, J. Large language models as zero-shot conversational recommenders. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; pp. 720–730.
45. Wang, Y.; Chu, Z.; Ouyang, X.; Wang, S.; Hao, H.; Shen, Y.; Gu, J.; Xue, S.; Zhang, J.Y.; Cui, Q.; et al. Enhancing recommender systems with large language model reasoning graphs. *arXiv* **2023**, arXiv:2308.10835.
46. Huang, X.; Lian, J.; Lei, Y.; Yao, J.; Lian, D.; Xie, X. Recommender ai agent: Integrating large language models for interactive recommendations. *arXiv* **2023**, arXiv:2308.16505. [CrossRef]
47. Luo, S.; Yao, Y.; He, B.; Huang, Y.; Zhou, A.; Zhang, X.; Xiao, Y.; Zhan, M.; Song, L. Integrating large language models into recommendation via mutual augmentation and adaptive aggregation. *arXiv* **2024**, arXiv:2401.13870.
48. Li, Y.; Lin, X.; Wang, W.; Feng, F.; Pang, L.; Li, W.; Nie, L.; He, X.; Chua, T.S. A survey of generative search and recommendation in the era of large language models. *arXiv* **2024**, arXiv:2404.16924.
49. Xue, Y.; Greenewald, K.; Mroueh, Y.; Mirzasoleiman, B. Verify when Uncertain: Beyond Self-Consistency in Black Box Hallucination Detection. *arXiv* **2025**, arXiv:2502.15845.
50. McKenna, N.; Li, T.; Cheng, L.; Hosseini, M.J.; Johnson, M.; Steedman, M. Sources of hallucination by large language models on inference tasks. *arXiv* **2023**, arXiv:2305.14552.
51. Liu, H.; Wang, Y.; Fan, W.; Liu, X.; Li, Y.; Jain, S.; Liu, Y.; Jain, A.; Tang, J. Trustworthy ai: A computational perspective. *ACM Trans. Intell. Syst. Technol.* **2022**, *14*, 1–59. [CrossRef]
52. Li, Y.; Tan, Z.; Liu, Y. Privacy-preserving prompt tuning for large language model services. *arXiv* **2023**, arXiv:2305.06212.
53. Zhou, H.; Lee, K.H.; Zhan, Z.; Chen, Y.; Li, Z.; Wang, Z.; Haddadi, H.; Yilmaz, E. Trustrag: Enhancing robustness and trustworthiness in rag. *arXiv* **2025**, arXiv:2501.00879.
54. Nallapati, R. Discriminative models for information retrieval. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 25–29 July 2004; pp. 64–71.
55. Liu, T.Y. Learning to rank for information retrieval. *Found. Trends® Inf. Retr.* **2009**, *3*, 225–331. [CrossRef]
56. Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; Ward, R. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 694–707. [CrossRef]
57. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional neural network architectures for matching natural language sentences. *Adv. Neural Inf. Process. Syst.* **2014**, *27* , 2042–2050.
58. Shen, Y.; He, X.; Gao, J.; Deng, L.; Mesnil, G. A latent semantic model with convolutional-pooling structure for information retrieval. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 101–110.

59. Lai, H.; Liu, X.; Iong, I.L.; Yao, S.; Chen, Y.; Shen, P.; Yu, H.; Zhang, H.; Zhang, X.; Dong, Y.; et al. AutoWebGLM: A Large Language Model-based Web Navigating Agent. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; pp. 5295–5306.
60. He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; Yu, D. WebVoyager: Building an end-to-end web agent with large multimodal models. *arXiv* **2024**, arXiv:2401.13919.
61. Chen, W.; You, Z.; Li, R.; Guan, Y.; Qian, C.; Zhao, C.; Yang, C.; Xie, R.; Liu, Z.; Sun, M. Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence. *arXiv* **2024**, arXiv:2407.07061.
62. Chen, Z.; Liu, K.; Wang, Q.; Liu, J.; Zhang, W.; Chen, K.; Zhao, F. Mindsearch: Mimicking human minds elicits deep ai searcher. *arXiv* **2024**, arXiv:2407.20183.
63. Li, Y.; Yang, N.; Wang, L.; Wei, F.; Li, W. Learning to rank in generative retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, USA, 20–27 February 2024; Volume 38, pp. 8716–8723.
64. Li, Y.; Yang, N.; Wang, L.; Wei, F.; Li, W. Multiview identifiers enhanced generative retrieval. *arXiv* **2023**, arXiv:2305.16675.
65. Li, Y.; Zhang, Z.; Wang, W.; Nie, L.; Li, W.; Chua, T.S. Distillation enhanced generative retrieval. *arXiv* **2024**, arXiv:2402.10769.
66. Gong, P.; Li, J.; Mao, J. Cosearchagent: A lightweight collaborative search agent with large language models. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024; pp. 2729–2733.
67. Qin, Y.; Cai, Z.; Jin, D.; Yan, L.; Liang, S.; Zhu, K.; Lin, Y.; Han, X.; Ding, N.; Wang, H.; et al. Webcpm: Interactive web search for chinese long-form question answering. *arXiv* **2023**, arXiv:2305.06849.
68. Zhou, R.; Yang, Y.; Wen, M.; Wen, Y.; Wang, W.; Xi, C.; Xu, G.; Yu, Y.; Zhang, W. TRAD: Enhancing LLM Agents with Step-Wise Thought Retrieval and Aligned Decision. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024; pp. 3–13.
69. Li, Y.; Wang, W.; Qu, L.; Nie, L.; Li, W.; Chua, T.S. Generative cross-modal retrieval: Memorizing images in multimodal language models for retrieval and beyond. *arXiv* **2024**, arXiv:2402.10805.
70. Long, X.; Ma, Z.; Hua, E.; Zhang, K.; Qi, B.; Zhou, B. Retrieval-Augmented Visual Question Answering via Built-in Autoregressive Search Engines. *arXiv* **2025**, arXiv:2502.16641. [CrossRef]
71. Rashid, M.S.; Meem, J.A.; Dong, Y.; Hristidis, V. EcoRank: Budget-Constrained Text Re-ranking Using Large Language Models. *arXiv* **2024**, arXiv:2402.10866.
72. Duan, S.; Babu, S. Processing forecasting queries. In Proceedings of the 33rd International Conference on Very Large Data Bases, Vienna, Austria, 23–27 September 2007; pp. 711–722.
73. Mayfield, C.; Neville, J.; Prabhakar, S. ERACER: A database approach for statistical inference and data cleaning. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, Indianapolis, IN, USA, 6–10 June 2010; pp. 75–86.
74. Rendle, S. Scaling factorization machines to relational data. *Proc. VLDB Endow.* **2013**, *6*, 337–348. [CrossRef]
75. Miao, H.; Li, A.; Davis, L.S.; Deshpande, A. Modelhub: Towards unified data and lifecycle management for deep learning. *arXiv* **2016**, arXiv:1611.06224.
76. Wang, W.; Zhang, M.; Chen, G.; Jagadish, H.; Ooi, B.C.; Tan, K.L. Database meets deep learning: Challenges and opportunities. *ACM Sigmod Rec.* **2016**, *45*, 17–22. [CrossRef]
77. Jin, Z.; Anderson, M.R.; Cafarella, M.; Jagadish, H. Foofah: Transforming data by example. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2017; pp. 683–698.
78. Yin, P.; Neubig, G.; Yih, W.t.; Riedel, S. TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv* **2020**, arXiv:2005.08314.
79. Tang, N.; Fan, J.; Li, F.; Tu, J.; Du, X.; Li, G.; Madden, S.; Ouzzani, M. RPT: Relational pre-trained transformer is almost all you need towards democratizing data preparation. *arXiv* **2020**, arXiv:2012.02469. [CrossRef]
80. Li, P.; He, Y.; Yan, C.; Wang, Y.; Chaudhuri, S. Auto-tables: Synthesizing multi-step transformations to relationalize tables without using examples. *arXiv* **2023**, arXiv:2307.14565. [CrossRef]
81. Chen, Z.; Gu, Z.; Cao, L.; Fan, J.; Madden, S.; Tang, N. Symphony: Towards Natural Language Query Answering over Multi-modal Data Lakes. In Proceedings of the CIDR, Amsterdam, The Netherlands, 8–11 January 2023; pp. 1–7.
82. Saeed, M.; De Cao, N.; Papotti, P. Querying large language models with SQL. *arXiv* **2023**, arXiv:2304.00472.
83. Halevy, A.; Dwivedi-Yu, J. Learnings from data integration for augmented language models. *arXiv* **2023**, arXiv:2304.04576.
84. Zhou, X.; Sun, Z.; Li, G. Db-gpt: Large language model meets database. *Data Sci. Eng.* **2024**, *9*, 102–111. [CrossRef]
85. Zhou, X.; Li, G.; Sun, Z.; Liu, Z.; Chen, W.; Wu, J.; Liu, J.; Feng, R.; Zeng, G. D-bot: Database diagnosis system using large language models. *arXiv* **2023**, arXiv:2312.01454. [CrossRef]
86. Yang, H.; Wu, R.; Xu, W. TransCompressor: LLM-Powered Multimodal Data Compression for Smart Transportation. In Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, Washington, DC, USA, 18–22 November 2024; pp. 2335–2340.
87. Xu, R.; Qi, Z.; Guo, Z.; Wang, C.; Wang, H.; Zhang, Y.; Xu, W. Knowledge conflicts for llms: A survey. *arXiv* **2024**, arXiv:2403.08319.

88. Ling, L.; Rabbi, F.; Wang, S.; Yang, J. Bias Unveiled: Investigating Social Bias in LLM-Generated Code. *arXiv* **2024**, arXiv:2411.10351. [CrossRef]
89. Zhang, X.; Zhang, Z.; Ji, S.; Wang, T. Trojaning language models for fun and profit. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; IEEE: New York, NY, USA, 2021; pp. 179–197.
90. Wan, A.; Wallace, E.; Shen, S.; Klein, D. Poisoning language models during instruction tuning. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; PMLR: Cambridge, MA, USA, 2023; pp. 35413–35425.
91. Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [CrossRef]
92. Harang, R. Securing Llm Systems Against Prompt Injection. Available online: https://developer.nvidia.com/blog/securing-llm-systems-against-prompt-injection/ (accessed on 15 October 2024).
93. Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; et al. Extracting training data from large language models. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Vancouver, BC, USA, 11–13 August 2021; pp. 2633–2650.
94. Zhang, K.; Liu, J. Review on the application of knowledge graph in cyber security assessment. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 768, p. 052103.
95. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3347–3366. [CrossRef]
96. Zheng, Y.; Chen, Y.; Qian, B.; Shi, X.; Shu, Y.; Chen, J. A review on edge large language models: Design, execution, and applications. *ACM Comput. Surv.* **2025**, *57*, 1–35. [CrossRef]
97. Ji, Y.; Xiang, Y.; Li, J.; Chen, W.; Liu, Z.; Chen, K.; Zhang, M. Feature-based Low-Rank Compression of Large Language Models via Bayesian Optimization. *arXiv* **2024**, arXiv:2405.10616.
98. Zadeh, A.H.; Mahmoud, M.; Abdelhadi, A.; Moshovos, A. Mokey: Enabling narrow fixed-point inference for out-of-the-box floating-point transformer models. In Proceedings of the 49th Annual International Symposium on Computer Architecture, New York, NY, USA, 18–12 June 2022; pp. 888–901.
99. He, Y.; Fang, J.; Yu, F.R.; Leung, V.C. Large language models (LLMs) inference offloading and resource allocation in cloud-edge computing: An active inference approach. *IEEE Trans. Mob. Comput.* **2024**, *23*, 11253–11264. [CrossRef]
100. Nguyen, S.; Zhou, B.; Ding, Y.; Liu, S. Towards sustainable large language model serving. *ACM SIGENERGY Energy Inform. Rev.* **2024**, *4*, 134–140. [CrossRef]
101. Argerich, M.F.; Patiño-Martínez, M. Measuring and improving the energy efficiency of large language models inference. *IEEE Access* **2024**, *12*, 80194–80207. [CrossRef]
102. Hazra, S. Review on Social and Ethical Concerns of Generative AI and IoT. In *Generative AI: Current Trends and Applications*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 257–285.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.