ChatSync: Large Language Model Enabled Spatial-Temporal Knowledge Reasoning for Production Logistics Synchronization

Jinpeng Li, Zhiheng Zhao, *Senior Member, IEEE*, Chen Yang, *Member, IEEE*, Sihan Huang, *Member, IEEE*, Lik-Hang Lee, *Senior Member, IEEE*, George Q. Huang, *Fellow, IEEE*

***Abstract*—With increasing pressure from customized demands, discrete manufacturing systems face challenges due to fluctuating resource requirements. These challenges hinder the synchronization of production logistics (PL), which is essential for coordinating resources and ensuring smooth production. Poor synchronization will result in resources waiting on each other, leading to delays and idle time. Accordingly, this paper proposes ChatSync, a framework leveraging large language model (LLM) and spatial-temporal knowledge reasoning to optimize resource allocation, delivery, and monitoring in industrial applications, particularly within the Industrial Internet of Things (IIoT) environment. First, the resource spatial-temporal graph (RSTG) is constructed by integrating real-time IIoT data and expert operational experience, enhancing the knowledge base of LLM through cross-domain knowledge fusion. Second, graph-based reasoning optimization is presented, incorporating spatial-temporal, contextual, and relational reasoning mechanisms, enabling LLM to achieve credible and responsible analysis and decision-making. Third, the PL-oriented ChatSync framework with knowledge and reasoning engines is proposed, supporting chat-based interactions for resilient resource allocation, personalized suggestion, and precise traceability. A case study in air conditioning manufacturing demonstrates that ChatSync outperforms existing benchmark methods in various PL phases, achieving a delivery punctuality rate of 91.2%.**

***Index Terms*—Production logistics, reasoning optimization, resource allocation, Industrial Internet of Things (IIoT), large language model (LLM), responsible AI.**

I. Introduction

PRODUCTION logistics (PL) focuses on in-plant logistics and distribution activities, aiming to address untimely and inaccurate deliveries. As PL connects multiple production processes by transporting materials and work-in-progress items, it accounts for approximately 95% of the total execution time in manufacturing [1]. Effective PL synchronization ensures that operators, tools, and materials are available within a specified time window and position, enabling production, assembly, or transportation operations to commence and conclude as planned [2]. However, production systems are rapidly transitioning from a paradigm characterized by fewer varieties and large batches to one that supports multiple varieties in smaller batches in Industry 4.0, with an anticipated future shift toward customization and individualization in Industry 5.0 [3]. Although many efforts have been made to address the increased complexity of PL due to this transition, such as Kanban [4], which offers a straightforward and visual interface for managing logistics [5], real-world implementation often falls short of expectations.

Recently, the large language model (LLM) has emerged as a transformative solution in industrial manufacturing processes [6], [7]. In contrast to traditional methods, LLM allows users to interact with systems through natural language [8]. Our research team's practical experience assisting many enterprises in upgrading their production systems shows that no matter how meticulously designed and sophisticated the system is, its effectiveness is often significantly diminished during actual implementation. Due to workers' varying educational backgrounds and levels of computer literacy, many users struggle with retrieving necessary data or understanding the system's output [9]. This gap between system complexity and user comprehension often leads to production inefficiencies. Given this issue, conversation, which has been regarded since the dawn of human society as the most efficient interaction mode, offers a promising alternative [10]. Especially in the current 21st century, with the widespread use of smartphones, nearly everyone can communicate through messaging apps like WhatsApp and WeChat. Therefore, a chat-based system powered by LLM holds substantial potential for end users in the PL chain. It can streamline operations by providing access to critical information through a conversational interface, recommending resource allocation plans, and offering handling suggestions to minimize personal injury caused by improper operation.

However, despite significant advancements in recent research on PL synchronization, the integration of LLM technology still presents several challenges:

1) How can real-time sensor data and multi-source knowl-

This work was supported by the National Natural Science Foundation of China (No. 52305557), Hong Kong Research Grants Council (No. 15203025, T32-707/22-N, C7076-22GF, R7036-22), Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515011930), Research Institute for Advanced Manufacturing (RIAM) of The Hong Kong Polytechnic University (No. CDLU, CDLM, CDJX). *(Corresponding authors: Zhiheng Zhao and Chen Yang.)*

Jinpeng Li, Zhiheng Zhao, Lik-Hang Lee and George Q. Huang are with the Department of Industrial and Systems Engineering and the Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: jin-peng.li@connect.polyu.hk; zhiheng.zhao@polyu.edu.hk; lik-hang.lee@polyu.edu.hk; gq.huang@polyu.edu.hk).

Chen Yang is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China(e-mail: yangchen666@bit.edu.cn).

Sihan Huang is with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China, and also with the Key Laboratory of Industry Knowledge and Data Fusion Technology and Application, Ministry of Industry and Information Technology, Beijing Institute of Technology, Beijing, China (e-mail: hsh@bit.edu.cn).

edge be fully fused into a unified graph that integrates data and knowledge domains? Data generated from Industrial Internet of Things (IIoT) devices and knowledge extracted from standards document and expert experience are heterogeneous and multi-scale. The absence of universal expressions that integrate information and knowledge domains obstructs collaboration between foundation models and LLM.

2) How can graph reasoning mechanisms be conducted on the graph, offering precise insights into allocation recommendation, operational suggestion, and resource traceability? While LLM offers excellent logical inference based on natural language processing (NLP), it may generate inaccurate insights and unreasonable suggestions due to hallucinations, resulting in production delays and resource misallocation.

3) How can LLM effectively assist different roles within the industrial park, from workers to managers, by providing tailored and interactive assistance through chat? Integrating LLM into PL synchronization workflows requires careful attention to ensuring ease of use, accuracy, and role-specific adaptability.

This research aims to improve PL synchronization through the integration of LLM technology and reasoning mechanisms. The main contributions are threefold: (1) We propose a novel approach for constructing resource status and knowledge graphs by applying ontology modeling and semantic analysis within LLM. These graphs, incorporating diverse perspectives, are fused into a unified Resource Spatial-Temporal Graph (RSTG), ensuring that ongoing activities in the physical space are accurately reflected in the digital space. (2) Three graph reasoning mechanisms built on the RSTG are proposed, including spatial-temporal reasoning for dynamic decision optimization, contextual reasoning for operational suggestion refinement, and relational reasoning for resource traceability improvement. These reasoning mechanisms assist the LLM in achieving precise delivery within PL while minimizing errors caused by hallucinations. (3) We present the LLM-powered ChatSync framework, which provides role-specific, interactive assistance. Chat-based interactions facilitate resource optimization for planners, operational guidance for material handlers, and resource traceability for logistics coordinators, thus enhancing overall efficiency and coordination within the industrial park.

The paper is organized as follows: Section II reviews related work. Section III introduces the overall framework of ChatSync. Section IV and Section V focus on the knowledge engine and reasoning engine included in ChatSync, respectively. The performances of proposed framework are presented in Section VI, and conclusions are summarized in Section VII.

II. Related Work

A. Production logistics synchronization

The term "synchronization" originates from the Greek concept of events "happening at the same time" or "agreeing in time" [11]. Within the manufacturing domain, synchronization refers to ensuring that the right components are delivered to subsequent production stages precisely when required [12], [13]. A prominent concept aligned with this principle is Just-In-Time (JIT) production, which prioritizes the synchronization of processes to ensure that "all processes produce the necessary parts at the necessary time, maintaining only the minimum stock required to link processes together" [14]. However, as customer demands become increasingly individualized and dynamic, JIT, which was primarily proposed in Industry 3.0 with a focus on automation, falls short in addressing the requirements of on-demand services, customized production, and resilient manufacturing in the eras of Industry 4.0 and Industry 5.0 [15].

Consequently, research efforts have shifted toward synchronized PL to enable more flexible, resilient, and sustainable production management [16], [17]. Guo et al. [18] proposed a synchronization mechanism combining mixed-integer linear programming and constraint programming to solve PL tasks, addressing precedence, spatial, and temporal constraints for small to medium problems. Pereira et al. [19] introduced a hybrid scheduling framework using greedy methods, local search, simulated annealing, and variable neighborhood search to select optimal algorithms for various scenarios. Zhang et al. [20] developed a nine-layer intelligent digital twin framework with an artificial bee colony algorithm to improve logistics timeliness and equipment utilization in flexible production planning. Zhao et al. [21] introduced a graduation-inspired system that uses smart tickets to synchronize PL workflow. Delivery tickets manage shipment timing, resource tickets coordinate tools and personnel, and execution tickets oversee task performance. The architecture employs the genetic algorithm for optimizing resource scheduling, allowing the system to adapt to variability.

Despite these advancements, most existing approaches focus predominantly on static synchronization strategies, overlooking the dynamic nature of real-world manufacturing environments. These approaches often fail to account for real-time disruptions such as demand fluctuations, machine breakdowns, and unforeseen delays. This gap underscores the urgent need for adaptive, resilient synchronization that dynamically responds to real-time changes, enabling more agile and responsive production systems in Industry 4.0 and 5.0. To bridge this gap, we leverage IoT data to construct spatial-temporal knowledge of PL resource, which provides precise tracking of evolving production states. By integrating this timely information into the synchronization process, our approach facilitates more responsive and coordinated utilization of resources.

B. Dynamic resource allocation focused on the execution stage

The manufacturing stages can be broadly categorized into planning, scheduling, and execution [22]. Planning focuses on the strategic, long-term organization of overarching project activities [23]. Scheduling entails breaking down major activities into detailed work packages and defining their specific content and timelines [24]. Execution emphasizes the flexible adjustment of task implementation and allocating operations to specific workers within designated time windows [25]. Among these stages, the execution stage is particularly critical for

improving PL synchronization, as it ensures both adaptability and stability in the face of uncertainties, such as fluctuations in production demand, frequent changes in resource availability and variability of order arrivals and delivery arrangement [26], [27].

To effectively address these challenges, extensive research has been devoted to exploring dynamic resource allocation and resilient decision-making strategies during this stage. Some studies advocate for data-driven approaches to prioritize resource allocation. Zhang et al. [28] proposed an analytical target cascading model, where manufacturing resources are linked at various levels to minimize cost, time, and energy deviations. Zhao et al. [29] introduced a resource allocation method using deep learning and graph-inspired algorithms to optimize a real-time resource graph based on time windows and capacity limits. With advancements in cloud computing and AI, new technologies are accelerating innovative solutions [30]. Pan et al. [31] proposed a multi-level digital twin system using edge, fog, and cloud computing to optimize efficiency and cost under varying dynamics. Aron et al. [32] developed a material handling system integrating cloud computing and machine learning, enabling demand-responsive adaptation and reconfigurable services in line with the Physical Internet.

Although previous studies have proposed different resource allocation methods in manufacturing, these approaches use greedy-based or distance-based ideas to handle the allocation problem according to the current environment without considering future resource demands at various sites. This lack of foresight often results in resources being dispatched from distant locations when new demands arise, rather than enabling proactive planning and optimized distribution. To address this limitation, we introduce reasoning mechanisms that proactively consider potential resource constraints and dynamic task requirements across PL stages, thereby enabling more efficient and forward-looking resource allocation.

C. Manufacturing-oriented large language models

Recently, LLM, represented by ChatGPT [33], has revolutionized NLP by enabling machines to comprehend and generate human-like text [34]. These models support natural interaction, advanced reasoning, and context-aware responses, offering significant potential to enhance operational efficiency and address complex challenges through their assistance in perception, decision-making, and execution processes [35]. By enabling seamless integration with existing manufacturing systems and providing natural language interfaces, LLM facilitate human-machine collaboration, forming more intuitive and efficient workflows.

Recent research has focused on adapting LLM for manufacturing tasks. Zhou et al. [36] proposed CausalKGPT, an LLM enhanced with domain-specific knowledge graphs to diagnose aerospace manufacturing defects, providing actionable insights for defect mitigation. Wang et al. [37] used an LLM to improve cobot navigation by interpreting natural language commands and generating executable code for tasks based on a 3D point cloud model. Colabianchi et al. [38] developed a digital assistant powered by an LLM to enhance assembly processes, using retrieval-based mechanisms and optimized prompts to improve response reliability and reduce hallucinations. Fan et al. [39] proposed an embodied intelligence framework for autonomous industrial robots, in which a GPT-4-powered LLM functions as an agent responsible for decision-making within a simulation environment, demonstrating the potential of LLMs in manufacturing automation. Tsushima et al. [40] eveloped an LLM-driven robotic assistance system designed to support human workers in the automobile manufacturing industry. By enabling flexible task planning through natural language dialogue, the system allows robots to interactively generate and execute work plans in response to human input. Yu et al. [41] presented a knowledge graph-enhanced LLM approach for dynamic rescheduling in human-robot collaborative disassembly tasks, where the LLM provides language-based task reordering suggestions, and the knowledge graph ensures constraint validation and resource dependency management.

Despite the growing interest in applying LLM and knowledge graphs in manufacturing, their application in complex decision-making scenarios remains limited. On the one hand, the application of knowledge graphs has predominantly focused on knowledge dissemination tasks such as troubleshooting guidance and instructional support, while their role in dynamic decision-making across multi-phase PL workflows remains largely unexplored. Additionally, due to the low interpretability of LLM and their susceptibility to hallucinations, significant challenges persist in ensuring the credibility of the recommendations generated by these models. To address these gaps, this research proposes the LLM-driven ChatSync framework, which integrates a knowledge engine to facilitate context-aware suggestion generation in specialized domains. Furthermore, it incorporates a reasoning engine designed to support responsible decision-making across different phases, thereby enhancing synchronization within PL workflows.

III. Overall Framework

The ChatSync framework, as illustrated in Fig. 1, adopts a multi-layered structure to address the PL multifaceted challenges by integrating domain-specific knowledge, real-time data analysis, and graph reasoning. It comprises the component layer, which serves as the knowledge and computational foundation; the reasoning layer, which provides targeted analytical capabilities for dynamic decision-making; and the application layer, which bridges intelligent insights with practical user interactions. Each layer is intricately connected, enabling the system to transform raw data into actionable recommendations, optimize resource allocation, enhance traceability, and ensure operational resilience across varying phases of the logistics workflow.

1) Component layer: The component layer serves as the foundation of the overall framework, offering a domain-specific vertical knowledge base and essential reasoning tools. The layer is structured around two primary modules: the knowledge engine and the reasoning engine. The knowledge engine integrates general, professional, and expert knowledge, progressively organizing this information into a resource knowledge graph. Concurrently, real-time data from

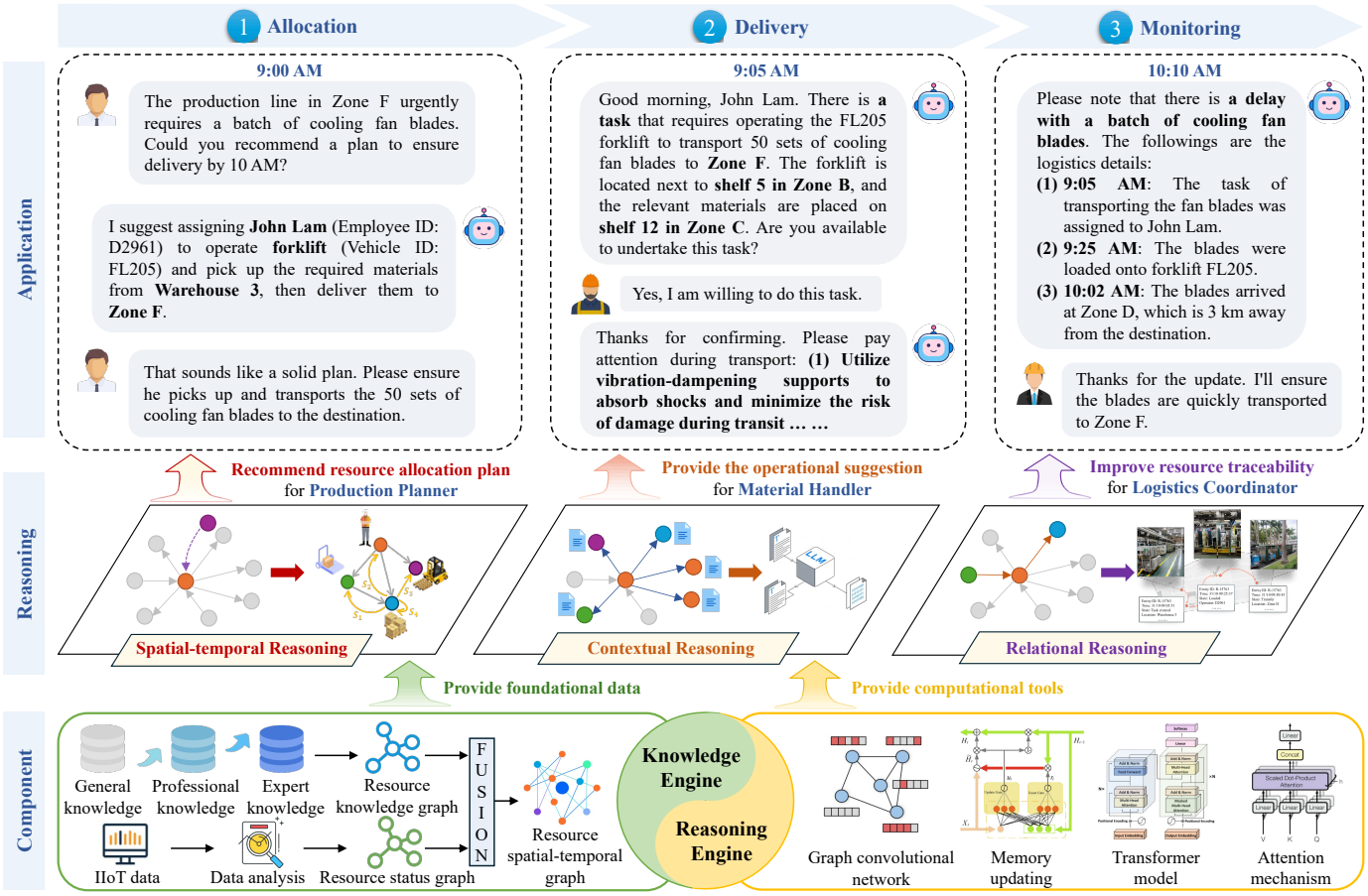

Fig. 1. The overall framework of the ChatSync.

IIoT devices is analyzed to construct a resource status graph, which captures the dynamic states of resources. These two graphs are fused into an RSTG, effectively encoding spatial and temporal relationships in the data, enabling a comprehensive representation of resource interactions. The reasoning engine equips the framework with powerful computational reasoning capabilities. It incorporates the graph convolutional network (GCN) to process heterogeneous nodes of knowledge graph, extracting structural and relational insights. Memory updating mechanisms enhance the LLM's ability to adapt dynamically to changing information, while the transformer model, powered by attention mechanisms, excels at modeling sequential dependencies and context awareness, enhancing dynamic resource management and resilient decision-making.

2) Reasoning layer: The reasoning layer underpins the analysis and decision-making of the LLM, utilizing multiple reasoning mechanisms on RSTG to address critical PL challenges. It encompasses three modules, spatial-temporal, contextual, and relational reasoning, each designed to provide targeted solutions to distinct operational needs. Spatial-temporal reasoning analyzes spatial dependencies and temporal dynamics to recommend resilient resource allocation plans for production planners. This ensures efficient resource deployment under dynamic and uncertain conditions, enabling the system to respond to evolving production demands with agility and minimal disruption. Contextual reasoning integrates task-specific information, such as material attributes and operational constraints, to provide precise and actionable suggestions for material handlers, reducing damage to materials and minimizing personal risk caused by improper operation during transportation. Relational reasoning aims to analyze the historical interaction data and entity relationships between personnel and resources to improve resource traceability, helping logistics coordinators monitor and manage the flow of resources with clarity and precision.

3) Application layer: The application layer empowers the ability to interact with users by providing targeted feedback tailored to different phases. During the allocation phase, the system recommends actionable and resilient resource allocation plans for production planners. Specifically, it suggests the assignment of specific personnel, vehicles, and materials to meet urgent production demands, ensuring that resources are optimally deployed to maintain operational efficiency under time constraints. In the delivery phase, the system directly engages with material handlers, offering context-aware operational suggestions to enhance transportation reliability and safety. For example, it may recommend using vibration-dampening measures to protect designated fragile goods, mitigating risks of damage caused by external shocks or improper handling. Lastly, during the monitoring phase, the system can improve resource traceability by generating comprehensive and real-time updates for logistics coordinators. The updates

include critical details such as task assignments, timestamps, current locations, and progress logs, allowing coordinators to address delays efficiently and ensuring accurate delivery to the intended destination.

IV. Spatial-Temporal Knowledge Representation and Fusion

In this section, we focus on the knowledge engine of ChatSync, which integrates sensor data from the information domain and operational guidance from the knowledge domain. By analyzing and fusing the resource status graph and the resource knowledge graph, we construct the RSTG. This fused representation serves as the foundation for subsequent reasoning, enabling the system to capture both real-time dynamics and domain expertise in a unified structure.

A. Resource status graph in information domain

In manufacturing enterprises, numerous uncertain factors, including personnel, equipment, materials, and environmental conditions, often lead to issues such as degraded delivery performance of PL [42], [43]. To achieve accurate delivery and provide actionable insights for more informed dispatching decisions, we systematically collected, processed, and integrated data in the information domain and conducted the resource status graph, which includes the physical topology of resources, inter-resource relationships, and manufacturing events. It can reflect the real-time location, movement trends, and historical job situations of various mobile resources, forming a digital twin of the operational environment.

Generating and updating a resource status graph can be structured across three levels, as shown in Fig. 2. The physical layer aims to acquire positional data for various resources. Specifically, Specifically, RFID tags are affixed to materials, and readers are used to scan these tags to obtain detailed information about the items. Bluetooth low energy (BLE) tags are attached to vehicles, and the spatial coordinates can be determined by analyzing the received signal strength indication [44]. Wi-Fi technology is employed for workers to estimate their location through the signal link with mobile phones, eliminating the need for bulky positioning devices [45].

The data collected from the physical layer is subsequently transmitted to the Cyber-Physical Internet (CPI) gateway, a specialized component designed to analyze spatial-temporal relationships and trends among PL resources [46]. Through core functions such as definition, filtration, interaction, and aggregation, the CPI gateway can continuously monitor resource status and proactively detect the occurrence and execution of transportation events. For instance, it can identify when an operator begins driving a vehicle toward a destination or when designated materials have arrived at the required resource point. Meanwhile, it ensures that the resource status remains dynamically updated, providing a comprehensive foundation for optimizing PL synchronization.

At the digital level, a centralized cloud database utilizes structured data from CPI gateway to construct the resource status graph, a dynamic and heterogeneous representation of spatial and temporal relationships among system-wide

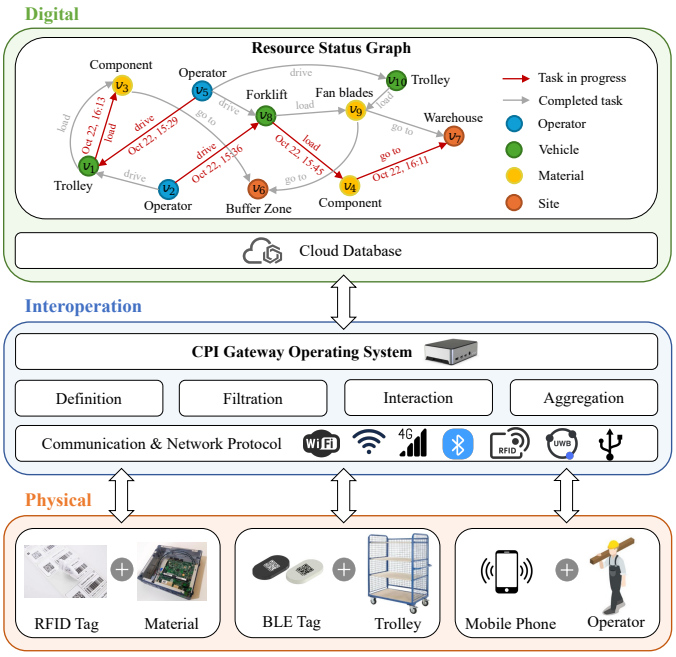

Fig. 2. Construction and update of resource status graph.

resources. The graph consists of diverse entities as nodes, including operator, vehicle, material, and site, each annotated with essential attributes such as type, entity ID, quantity, and position. Directed edges between nodes signify tasks or interactions, enriched with detailed metadata such as execution statuses (e.g., in progress or completed) and associated timestamps. As a high-level abstraction that translates raw data into actionable insights, the resource status graph provides a framework that enhances situational awareness by offering a clear overview of resource states, interdependencies, and execution progressions, enabling more efficient task coordination and resource optimization.

B. Resource knowledge graph in knowledge domain

Although PL guideline documents and safety management standards are extensive and detailed, their integration into daily operational processes remains insufficient [47]. While many companies have adopted different strategies and algorithms to store operational data and established rule-based retrieval systems to issue reminders, these solutions often produce rigid and redundant reminders. Recently, the integration of artificial intelligence, particularly LLM, has been proposed to improve workflow. However, despite the potential for LLM to comprehend contextual information and deliver domain-specific knowledge through fine-tuning, they remain susceptible to hallucinations and the generation of incorrect information [48].

Therefore, we developed a knowledge graph-based retrieval system that enables efficient semantic search and provides a more reliable tool for suggestion generation. As shown in Fig. 3, to construct a comprehensive and high-quality knowledge graph, we integrated general, professional, and expert knowledge from multiple sources, each contributing to different aspects of the knowledge representation. General knowledge

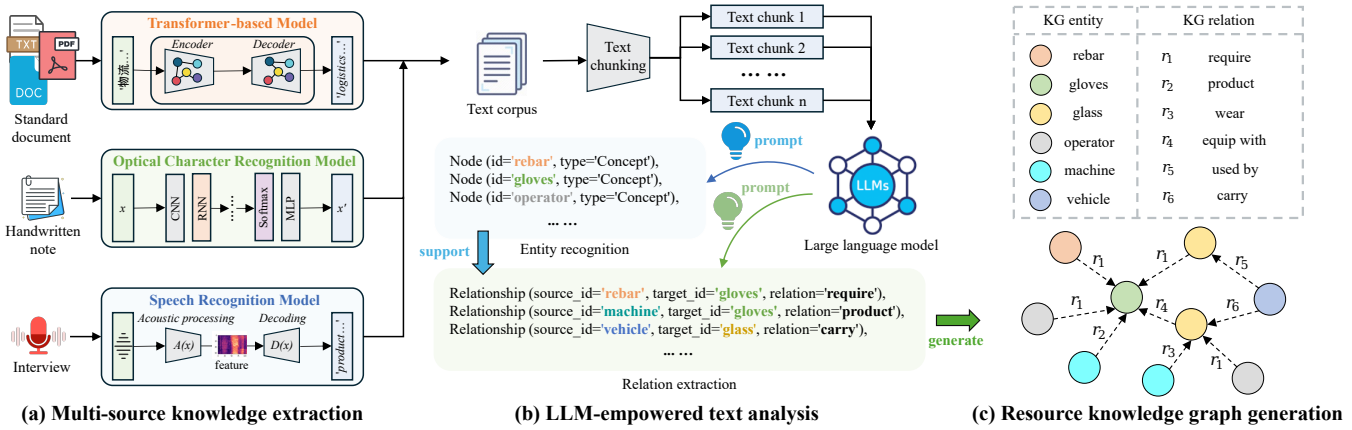

Fig. 3. Resource knowledge graph generation.

includes open-domain data that cover basic information, which creating a solid linguistic foundation for understanding knowledge graphs. Professional knowledge pertains to domain-specific files, including industry standards and internal process documents. Such documents offer detailed operation steps, enabling users to receive reminders that align with industry norms and safe management requirements. Expert knowledge is derived from the experiences of industry practitioners and logistics professionals. This type of knowledge is more informal and typically consists of feedback collected through interviews, field reports, and observational data. It is often less structured and can take various forms, from handwritten notes to interview transcripts. While challenging to formalize, expert knowledge adds significant value by providing experience-based suggestions to avoid personal injury from day-to-day operations.

Building upon this foundation, the processing of multi-source knowledge follows distinct procedures tailored to each data type. For standard documents, Transformer-based models such as BERT [49] are employed to extract and preprocess textual content. Specifically, the text is first normalized by removing non-printable characters, unifying Unicode encodings, and standardizing punctuation and whitespace. Subsequently, all non-English paragraphs are translated into English to ensure consistency in downstream processing. For handwritten notes, the process begins with scanning or photographing the content, followed by optical character recognition models [50], which are trained to recognize various handwriting styles and convert complex images into structured text. Likewise, audio data, like interview recordings, undergoes preprocessing via speech recognition models [51] to convert spoken language into accurate written text. Then, a unified text corpus is generated and will be divided into smaller, semantically meaningful units, such as sentences or paragraphs, to enhance processing efficiency and improve the granularity of information extraction for downstream tasks.

Given the enormous volume of text chunks, a bottom-up approach is employed to construct the resource knowledge graph. In this process, a native LLM is utilized for entity recognition and relation extraction. By incorporating a carefully designed prompt, the native LLM can effectively identify context-specific entities, such as material names, product specifications, and process steps, while also discerning complex interrelationships between them. The native LLM generates structured outputs in the form of entity-relation triplets, which serve as the foundation for constructing the resource knowledge graph. Entities are represented as nodes, and the relationships between them are captured as edges, enabling a scalable and efficient representation of complex knowledge structures.

C. Resource spatial-temporal graph based on cross-domain knowledge fusion

To provide accurate, real-time data alongside meaningful contextual knowledge, diverse perspectives from the information and knowledge domains are integrated into a unified graph structure, forming the RSTG. The cross-domain knowledge fusion process is illustrated in Fig. 4. Initially, the critical attributes, which encompass the unique characteristics of resources and will direct effort on operational suggestions, are identified for nodes within the information domain. For example, the product name and specifications of the material nodes, the type and load capacity of the vehicle nodes, and the qualifications and historical task performance of the operator nodes are considered.

Subsequently, the nodes in the knowledge domain, which represent concepts in the resource knowledge graph, are used to enrich the contextual information. For each previously selected node, the system searches for similar nodes in the resource knowledge graph that describe the same object from different perspectives. Once such nodes are identified, a directed edge is established between them, labeled as “related,” signifying the discovery of a potential new pattern. As illustrated in Fig. 4, when the material node is labeled “fan blade,” it serves as a digital representation of a specific batch of fan blades in the physical space. This allows the material node to be connected to the “fan blade” node in the knowledge domain via the “related” edge, enabling the RSTG to access both the information domain and knowledge domain attributes of the same object simultaneously.

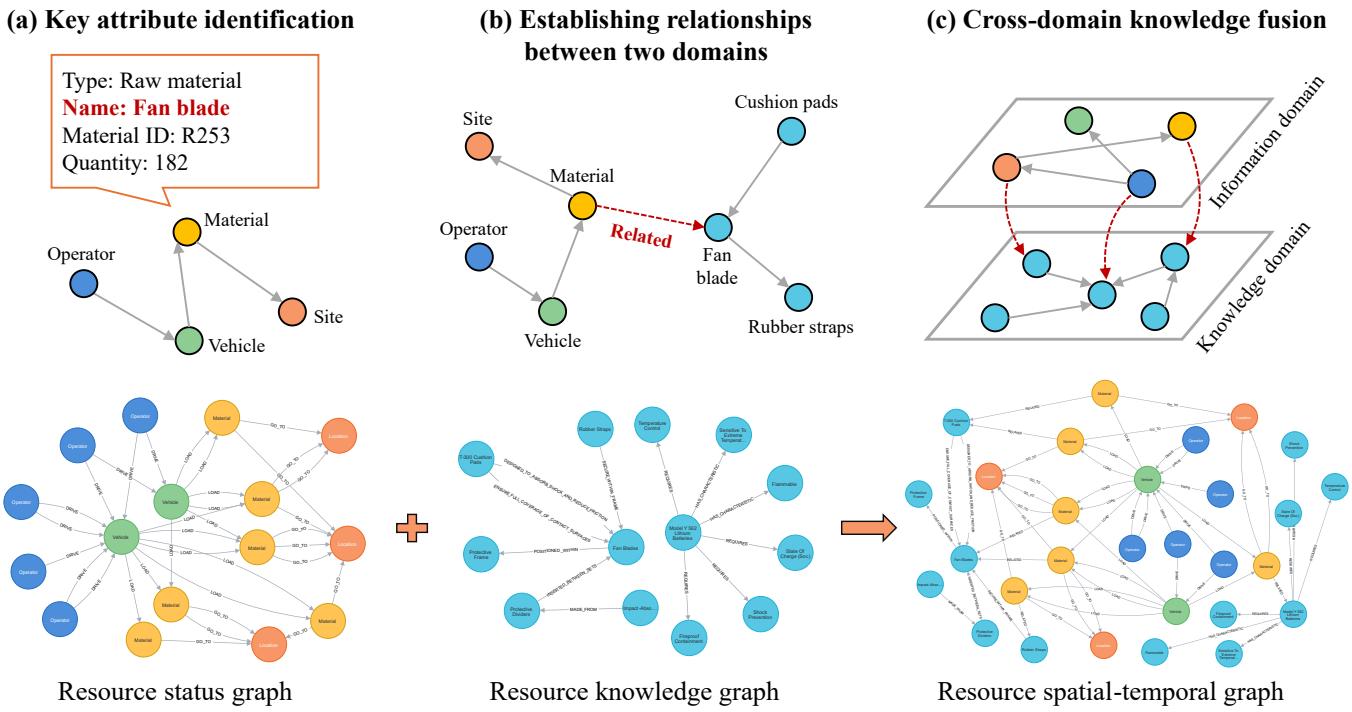

Fig. 4. Cross-domain knowledge fusion process.

After all patterns have been recognized, the RSTG, with spatial interconnectivity capturing the physical relationships between resources and temporal alignment tracking their status over time, will be constructed. Structurally, as depicted in the visualization in the lower part of the figure, the RSTG maintains the original form of both the resource status graph and the resource knowledge graph while enhancing them with additional, meaningful edges. Functionally, the RSTG bridges the gap between the real-time status of resources and a more abstract, knowledge-driven understanding of these resources, offering a robust foundation for dynamic analysis and decision-making.

V. Spatial-Temporal Knowledge Reasoning Optimization

The reasoning engine is the core of ChatSync, driving dynamic decision-making and suggestion generation in the execution stage. Unlike conventional knowledge reasoning, which typically focuses on the extraction and application of static domain knowledge, the knowledge reasoning optimization in PL emphasizes the dynamic, real-time adaptation of decisions and actions to changing operational conditions. In this section, we introduce the three pivotal reasoning optimization mechanisms in the reasoning layer: spatial-temporal reasoning for optimizing resource allocation, contextual reasoning for delivering personalized suggestions, and relational reasoning for improving resource traceability.

A. Spatial-temporal reasoning for resource allocation recommendation

Spatial-temporal reasoning focuses on analyzing spatial dependencies and temporal dynamics of nodes, which aims to generate optimized resource allocation plans by evaluating the suitability of interactions between resources over time and space. To facilitate such analysis, the RSTG is mathematically represented as a directed graph, denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Here, $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a set of nodes, each representing a resource node in information domain or a concept node in knowledge domain, and $\mathcal{E} \subseteq \{(v_i, v_j) | v_i, v_j \in \mathcal{V} \text{ and } i \neq j\}$ contains edges between nodes. For a given node v_i, it has two basic state vectors, namely message state $m_i(t)$ and memory state $s_i(t)$. Whenever interactions occur between resources in the real world, such as loading specified material node i onto a vehicle node j at time t, an event $e_{ij}(t)$ will occur. At this point, the state of the newest message regarding this event for both nodes can be expressed as

$$m_i(t) = Encoder\left(s_i\left(t^-\right), s_j\left(t^-\right), \Delta t, e_{ij}(t)\right), \quad (1)$$

$$m_j(t) = Encoder\left(s_j\left(t^-\right), s_i\left(t^-\right), \Delta t, e_{ij}(t)\right), \quad (2)$$

where $s_i(t^-)$ and $s_j(t^-)$ represent the memory state before the current time, Δt indicates the time interval since last resource interaction, and $Encoder(\cdot)$ is an embedding function which transforms resource information to high-level representations. To be simple, we set $Encoder(\cdot)$ as a concatenate operator, i.e., it concatenates the inputs to formulate the graph node representations.

As time progresses, there will be an increasing number of directed edges representing interaction events, and each node

in the RSTG may become connected to a large number of other nodes. For more effective feature propagation and learning, an aggregation operation $AGG(\cdot)$ is introduced to integrate a node's historical message states, formulated as

$$\bar{m}_i(t) = AGG\left(\{m_i(t-p) \,|\, 0 < p \leq t\}\right). \tag{3}$$

Here, p indexes the lag between the current time t and past time steps, and $m_i(t-p)$ denotes the message state of node i at a previous interaction. As an aggregation operator, $AGG(\cdot)$ has several variants: the most recent aggregator retains only the latest value, the maximum aggregator preserves the highest value, and the average aggregator computes the mean of all values [52]. To emphasize trends in resource changes over a recent period, the mean aggregator is ultimately adopted.

For compressed representation of each node's history and to capture long-term dependencies, the previous memory states of each node $s_i(t^-)$ and the aggregated message states $m_i(t)$ from past times are used to infer the updated memory states. The updating process is expressed as

$$z_i(t) = \sigma\left(W_i^Z \cdot \left[s_i\left(t^-\right) \parallel \bar{m}_i(t)\right] + b_i^Z\right), \tag{4}$$

$$r_i(t) = \sigma\left(W_i^R \cdot \left[s_i\left(t^-\right) \parallel \bar{m}_i(t)\right] + b_i^R\right), \tag{5}$$

where W_i^Z and W_i^R are learnable weight matrices, while b_i^Z and b_i^R are learnable biases. $\parallel$ represents concatenation operation which has a lower precedence than addition and multiplication operations, and $\sigma(\cdot)$ denotes the sigmoid function. From the perspective of function, (4) primarily determines whether the input at the current time step should be updated into the memory state, flexibly controlling the impact of current input on the memory state, thereby avoiding the vanishing gradient problem during training. (5) decides whether to ignore previous memory and reset the memory cell at the current time step.

To ensure that the memory state of nodes reflects short-term resource flow characteristics rather than persisting outdated states indefinitely, we employ the computational method depicted in (6) to allow nodes to forget some non-critical features. Subsequently, in conjunction with the values of the update module, each node utilizes (7) to adaptively achieve the self-updating of the memory state based on both historical and real-time states.

$$\widetilde{s}_i(t) = \sigma\left(W_i^S \cdot \left[r_i(t) \odot s_i\left(t^-\right) \parallel \bar{m}_i(t)\right] + b_i^S\right), \tag{6}$$

$$s_i(t) = (1 - z_i(t)) \odot s_i\left(t^-\right) + z_i(t) \odot \widetilde{s}_i(t), \tag{7}$$

where W_i^s and b_i^s are weight matrix and bias of node i, respectively. $\odot$ denotes element-wise dot product.

When the memory state $s_i(t)$ is obtained, the final spatial-temporal feature value is computed by L graph attention layers. The output of l-th layer can be given by

$$h_i^{(l)}(t) = GraphAttn\left(h_i^{(l-1)}(t), \mathcal{N}_i(t)\right), \tag{8}$$

where $h_i^{(l-1)}(t)$ is the output of the previous layer, and $\mathcal{N}_i(t)$ represents the set of neighboring nodes of node i. $GraphAttn(\cdot)$ denotes a graph attention layer. Notably, the initial input representation for each node is given by $h_i^{(0)}(t) = s_i(t)$. For each graph attention layer, it is defined as

$$GraphAttn\left(h_i^{(l-1)}(t), \mathcal{N}_i(t)\right) = \sigma\left(h_i^{(l-1)}(t) \parallel W_A \cdot (\text{head}_1 \parallel \cdots \parallel \text{head}_H)\right), \tag{9}$$

where H represents the total number of attention heads, and W_A is the transformation matrix. $head_p$ is the p-th output of the scaled dot-product attention, which is transformed using three matrices representing the query, key, and value, and it is obtained by

$$head_p = softmax\left(\frac{Q^{(l)}(t)W_p^Q {W_p^K}^T {K^{(l)}(t)}^T}{\sqrt{len\left(Q^{(l)}(t)W_p^Q\right)}}\right) \cdot V^{(l)}(t)W_p^V, \tag{10}$$

where W_p^Q, W_p^K, and W_p^V are the query, key, and value transformation matrices of the p-th head, respectively. T denotes the transpose operation of a matrix.

Since each head focuses on different aspects, the transformation matrices are different from each other. The query $Q^{(l)}(t)$ represents the content the model focuses on, the key $K^{(l)}(t)$ is a vector representation used to compare the similarity between the query and each position in the input sequence, and the value $V^{(l)}(t)$ contains the information to be attended. They can be expressed by

$$Q^{(l)}(t) = h_i^{(l-1)}(t) \parallel \phi(0), \tag{11}$$

$$K^{(l)}(t) = V^{(l)}(t) = h_j^{(l-1)}(t) \parallel e_{ij}\left(t_j\right) \parallel \phi\left(t - t_j\right), \tag{12}$$

where $j \in \mathcal{N}_i(t)$ and $t_j \leq t$. $e_{ij}(t_j)$ is the latest event information associated with neighbor node j, and t_j is the occurrence time of the corresponding event. $\phi(\cdot)$ is a time encoding method presented in Time2Vec [53], which can convert the time points in the time series into continuous vector representations, thus better capturing the periodicity, trends, and correlations of time. Ultimately, the output of multiple graph attention layers will be stored as spatial-temporal feature vectors of the corresponding node, i.e., $stf_i(t) = h_i^{(L)}(t)$.

When a new resource request arises, the system leverages spatial-temporal reasoning to identify the most appropriate node to fulfill the current demand, thereby generating an optimal allocation plan. The process begins with the selection of candidate nodes, expressed as

$$\mathcal{V}_f = Filter\left(\mathcal{V}, \delta_{req}, t\right), \tag{13}$$

where δ_{req} denotes the resource demand details, including the type, specification, quantity, and location of the required resources. $Filter(\cdot)$ is the filtering function designed to extract a subset of nodes $\mathcal{V}_f$ from the complete set $\mathcal{V}$ in the graph, ensuring that these candidates satisfy the requirements specified by δ_{req} and are available at time t.

Subsequently, the spatial-temporal suitability score for all candidate nodes in $\mathcal{V}_f$ is computed as

$$\varphi_{i,j} = W_E \cdot \Big(ReLU\big(\Phi_f * stf_i(t)\big) \odot ReLU\big(\Phi_g * stf_j(t)\big)\Big) + b_E. \tag{14}$$

where $*$ represents convolution operator, Φ_f and Φ_g are two learnable convolution filter, W_E and b_E denote learnable parameters, respectively, and $ReLU(\cdot)$ is a nonlinear activation function. This computation comprehensively captures the temporal and spatial dependencies between candidate and target nodes. A higher $\varphi_{i,j}$ value indicates a more favorable allocation, reflecting greater benefits achieved at a lower cost when assigning resource node i to meet the requirements of node j.

Finally, we use the $TopRank$ function, which will return the values and indices of the selected node with highest spatial-temporal suitability score, to derive the recommended resource allocation plan $\mathcal{S}_{rec}$, which can be expressed as

$$\mathcal{S}_{rec} = TopRank\left(\{\varphi_{i,j}, i \in \mathcal{V}_f, j \in \delta_{req}\}\right). \tag{15}$$

B. Contextual reasoning for personalized operation suggestion

Contextual reasoning is designed to provide personalized operational suggestions by identifying relevant knowledge nodes and constructing focused subgraphs. It aims to mitigate risks such as material damage and personal injury caused by improper handling during transportation, thereby enhancing operational safety and efficiency. To achieve this, contextual reasoning begins by representing the textual attributes of nodes and edges. Let x_v and x_e denote the sequential text associated with a node $v \in \mathcal{V}$ or an edge $e \in \mathcal{E}$. The feature vectors of knowledge nodes are initialized using pre-trained Embedding Models (EM), such as BGE [54], with the embeddings stored in a nearest-neighbor data structure. Specifically, given x_v as the textual attribute of node v, its representation z_v can be expressed as (16). The similar preprocessing method is also applied to edges.

$$z_v = EM(x_v). \tag{16}$$

When a material node v_m becomes a multi-hop neighbor of an operator node in the RSTG, it indicates that an operator is performing a loading or unloading task associated with the material node. To generate operational suggestions for the task, relevant nodes and edges containing the necessary knowledge are identified as

$$\mathcal{V}^* = TopN\left(\{Sim(z_m, z_v), v \in \mathcal{V}\}, \Theta\right), \tag{17}$$

where z_m represents the embedding of node v_m, and Sim measures the similarity between the text embeddings. The $TopN$ operation sorts the nodes in descending order of similarity and retrieves the top Θ elements.

Next, a subgraph $\mathcal{G}^*$ is constructed, encompassing the selected relevant nodes $\mathcal{V}^*$ and the edges $\mathcal{E}^*$ between these retrieved nodes. This subgraph provides two key benefits. Firstly, it filters out irrelevant nodes and edges that are unrelated to the ongoing task, which helps prevent irrelevant information from overshadowing useful data and ensures that the subsequent processing by the LLM remains focused. Secondly, it enhances computational efficiency by maintaining a manageable graph size, enabling the graph to be translated into natural language and input into the LLM for analysis.

$$\mathcal{G}^* = \{\mathcal{V}^*, \mathcal{E}^* \subseteq \{(\hat{v}_i, \hat{v}_j) | \hat{v}_i, \hat{v}_j \in \mathcal{V}^*\}\}. \tag{18}$$

To leverage the text analysis capabilities of LLM, the subgraph $\mathcal{G}^*$ is transformed into a textual format by flattening the textual attributes of its nodes and edges. Additionally, the text embedding z_m of the material node is concatenated to represent the query context. Let the textual flattening operation be denoted as $Textualize(\cdot)$, and the process can be expressed as

$$\mathcal{X}_{req} = CON(Textualize(\mathcal{G}^*), z_m). \tag{19}$$

Simultaneously, to address the challenges of fatigue and information overload stemming from repetitive reminders, contextual reasoning integrates dynamic prompts to deliver personalized suggestions tailored to diverse work scenarios. By leveraging historical operational data, the content of prompts is dynamically adjusted, ensuring the LLM based on contextual reasoning can provide practical and context-sensitive guidance aligned with actual work requirements. To enhance the relevance and efficacy of the prompts, we classify them into four predefined categories, each aligned with specific worker behaviors and performance patterns:

- **Apprentice Prompt:** For operators with minimal or no prior interaction with the current material type, this prompt offers detailed, step-by-step instructions to facilitate learning and reduce the likelihood of errors.
- **Master Prompt:** Designed for operators with a proven track record of accurate and efficient task execution for the same material, this prompt prioritizes brevity by focusing on critical operational steps, thereby minimizing cognitive load and streamlining information delivery.
- **Misoperation Prompt:** Aimed at operators who have previously selected incorrect vehicles, this prompt provides a concise reminder to verify vehicle selection, reducing the risk of future misallocations and ensuring proper asset utilization.
- **Deviation Prompt:** Targeted at operators with a history of reaching incorrect destinations, this prompt aims to reinforce task-specific details, particularly destination-related instructions, to mitigate delivery delays and enhance operational accuracy.

The personalize prompt selection mechanism can be formalized as

$$\mathcal{P}^* = \arg\max_{\eta} Evaluate\left(\mathcal{N}_o(t), \mathcal{P}_\eta, v_m\right), \tag{20}$$

where $\mathcal{N}_o(t)$ represents the adjacent nodes of the operator node v_o in the RSTG. These nodes correspond to objects involved in the task, such as vehicles, materials, and sites, while the edges encode task execution details, including timestamps, destinations, execution statuses, etc. $\mathcal{P}_\eta \in \{\mathcal{P}_{app}, \mathcal{P}_{mas}, \mathcal{P}_{mis}, \mathcal{P}_{dev}\}$ is the prompt set. $Evaluate$ serves as a scoring model to assess the relevance of each prompt category, with the highest-scoring

Algorithm 1: Target Identification

Input: Conditions set $\mathcal{C}$ and complete set of nodes $\mathcal{V}$ in RSTG
Output: Detailed information $\mathcal{F}$ of target node v^*

initialize an empty priority queue $\mathcal{Q}$;
initialize an empty visited set Λ;
initialize $v^* \leftarrow \emptyset$;
add a random subset of $\mathcal{V}$ of size k into $\mathcal{Q}$;
while $\mathcal{Q}$ *is not empty* **do**
  extract the first node from $\mathcal{Q}$ and assign it to v_{cur};
  if v_{cur} *satisfies all conditions in* $\mathcal{C}$ **then**
    set v^* to v_{cur} as the target node;
    break;
  end
  if $v_{cur} \notin \Lambda$ **then**
    add v_{cur} to Λ to avoid repeat visit; **foreach** $v_{next} \in \mathcal{N}_{cur}(t)$ **do**
      if $v_{next} \notin \Lambda$ **then**
        add v_{next} to the priority queue $\mathcal{Q}$ for further exploration;
      end
    end
  end
end
if $v^* = \emptyset$ **then**
  return None;
end
else
  do Algorithm 2 with the parameter v^*;
end

prompt selected as $\mathcal{P}^*$, which is subsequently input into the LLM.

Finally, the concatenated knowledge tokens $\mathcal{X}_{\text{req}}$ and the dynamically selected prompt $\mathcal{P}^*$ are fed into a pretrained LLM with frozen parameters θ. The model generates a probability distribution over the output sequence conditioned on the input as follows:

$$p_\theta(Y|[\mathcal{X}_{req} \parallel \mathcal{P}^*]) = \prod_{i=1}^{\mathcal{T}} p_\theta(y_i|y_{<i}, [\mathcal{X}_{req} \parallel \mathcal{P}^*]), \qquad (21)$$

where $\mathcal{T}$ denotes the total number of tokens in the output sequence. The LLM takes as input a sequence of tokens and models the conditional probability of each subsequent token given the preceding context, formulated as $p(y_\mathcal{T}|y_1, \ldots, y_{\mathcal{T}-1})$. The generated output sequence is denoted by $Y = (y_1, y_2, \ldots, y_\mathcal{T})$, which is interpreted as a set of operational suggestions delivered to the operator.

C. Relational reasoning for resource traceability improvement

Relational reasoning aims to enhance the ability to trace resources, enabling logistics coordinators to monitor logistics effectively and address potential delivery delays. Within the ChatSync framework, relational reasoning is activated under two scenarios. In the first scenario, when a logistics coordinator provides specific resource description information and inquires about its logistics progress, the relational reasoning utilizes the target identification algorithm to locate the corresponding resource node within the RSTG. Once the target node is identified, the connectivity analysis algorithm is employed to perform a retrospective analysis, delivering a comprehensive account of the node's interaction history and identifying potential causes of delays. The second scenario is triggered when IIoT-based data of a certain node updates reveal that the current time has exceeded the anticipated delivery deadline and the current location does not align with destination. In such cases, the system directly invokes the connectivity analysis algorithm to generate a detailed traceability report. Unlike the former scenario, the target identification algorithm is skipped because the position and structural context of the problematic node are already identified during its data update process, which eliminates the need to search for target node.

Algorithm 2: Connectivity Analysis

Input: Target node v^*
Output: Interaction report of the target node

initialize historical interaction information $\mathcal{F} \leftarrow \emptyset$;
initialize visited edges set $\Xi \leftarrow \emptyset$;
Function Tracing*(v)*:
  initialize pending edge list $\Omega \leftarrow \emptyset$;
  foreach *edge* e *in* v.links **do**
    if $e \notin \Xi$ **then**
      add e to Ξ to avoid repeat visit;
      record the event information of e into $\mathcal{F}$;
      add e to Ω for further exploration;
    end
  end
  while Ω *is not empty* **do**
    find the edge e_{next} with the earliest timestamp from Ω;
    remove e_{next} from Ω;
    call Tracing(e_{next});
  end
call Tracing(v^*);
generate interaction report using $\mathcal{F}$;

Next, we will provide a detailed description of the core algorithms used in relational reasoning. The target identification algorithm, outlined in Algorithm 1, leverages a priority queue $\mathcal{Q}$ to traverse the RSTG and find the target node v^* that satisfies the specified resource description $\mathcal{C}$. Initially, k nodes are randomly selected from the graph to populate $\mathcal{Q}$ as the starting points for the search process. These nodes are prioritized within $\mathcal{Q}$ according to their categories, with descending priority assigned to materials, operators, vehicles, and locations, thereby ensuring an orderly and efficient traversal process. During the entire search process, redundant visits are prevented through the maintenance of a visited set Λ. If no node fulfills the specified description $\mathcal{C}$, the algorithm terminates and returns None, indicating the absence of a suitable match. Conversely, upon successful identification of v^*, the connectivity analysis algorithm is subsequently invoked to perform resource tracing.

Fig. 5. Production environment of a workshop in the company.

In the connectivity analysis algorithm, as described in Algorithm 2, the interaction history $\mathcal{F}$ of the target node v^* is systematically reconstructed using a recursive procedure, TRACING. To optimize computational efficiency, visited edges set Ξ is maintained as a global variable, effectively preventing redundant exploration. The TRACING procedure processes the next unvisited edge with the earliest timestamp from the pending edge list Ω, which ensures that edges are examined in chronological order, thereby preserving temporal accuracy and coherence in the reconstructed interaction history. After visiting all edges in Ω, the Algorithm 2 will use the powerful text generation capability of LLM to output a comprehensive interaction report for logistics coordinators, enabling a deeper understanding of the resource flow and facilitating the identification of potential causes of delays.

VI. EXPERIMENT AND EVALUATION

To validate and assess the effectiveness of the proposed ChatSync, a series of experiments are conducted in this section. First, the experimental dataset and model training details are presented. Furthermore, a comparative analysis is conducted to demonstrate the superiority of ChatSync over benchmark approaches across various aspects of the PL workflow.

A. Setup of experiments

1) Data Collection: A case study was conducted on an air conditioning manufacturer located in the Greater Bay Area of China. The manufacturer's PL system is characterized by frequent resource allocation and transportation activities. Given the dynamic nature of decision-making and operational complexity, we deployed IIoT devices to collect the necessary data, as shown in Fig. 5. To collect real-time position data and resource interaction information, the CPI gateways were installed on the wall, which receives signals from IIoT devices distributed within the workstation, as shown in Fig. 5(a). Before transporting materials, such as the fan blades shown in Fig. 5(b), to their designated locations, operators must identify and operate the specific vehicle assigned by the system, as illustrated in Fig. 5(c). Each vehicle is equipped with a BLE tag to facilitate tracking and ensure efficient resource allocation and timely delivery, as shown in Fig. 5(d).

We collected two types of data in this company, namely production and knowledge data. Production data is primarily designed to train the spatial-temporal reasoning model of ChatSync, which generates resource allocation plans to synchronize the PL operations. To construct a high-quality training dataset, we collected resource requirements and positional coordinates data for one month. Besides, a team of logistics experts was invited to analyze the data and formulate optimal decisions, which were used as ground truth during model training. The resulting dataset contains approximately 50,000 decision records, encompassing resource allocation, vehicle matching, and operator assignment tasks, involving roughly 3,400 materials, vehicles, and operators. Knowledge data is utilized to train the contextual reasoning model, which enables the generation of operational guidance and suggestions for the operator during the execution of the PL tasks. As detailed in Section IV-B, ChatSync requires general, professional, and expert knowledge. General knowledge data is employed to enhance the basic semantic analysis capabilities of LLM. To minimize the time required for fine-tuning, we utilized a pre-trained open-source LLM as the foundational model. Professional knowledge was sourced from operation standards established by relevant national departments, as

well as company-specific procedural documents. Furthermore, expert knowledge was gathered through targeted interviews with experienced and senior operators, focusing on best practices for handling various components during transportation. After the transcribing and information process, knowledge data amounted to approximately 2,000 pages of structured data.

2) Parameters Settings: All computational experiments were conducted on a server equipped with an NVIDIA GeForce RTX 4090D GPU with 24 GB of memory, an Intel Xeon(R) Gold 6430 CPU with 32 cores, running Ubuntu 20.04, Python 3.8.10, and PyTorch 2.0.1 to implement the proposed model. For the production dataset, 70% of the data were used for training, 15% for validation, and the remaining 15% for testing. For the knowledge dataset, all documents were utilized for training. Following analysis of the relevant documents by human experts, approximately 50 true/false questions and 30 fill-in-the-blank questions were generated for logical judgment, in addition to 20 question-and-answer items designed for generative response testing.

In the reasoning engine of the proposed ChatSync model, the number of graph attention layers was set to 2, with a multi-head attention module using 4 heads per layer. The number of neighbors to sample was set to 10. Dropout was applied with a probability of 0.1. The Adam optimizer was employed with a learning rate of 0.0001, and the batch size was set to 200. DeepSeek V2 [55], a pre-trained open-source LLM in the general domain, was selected as the base model for text analysis. The node and time embeddings were set to a dimensionality of 100, while the memory module embeddings were set to 172. Each experiment was repeated five times. Parameters for the other baseline models were derived from their respective original proposals.

B. Phase 1: performance of resource allocation

1) Baselines: To verify the superiority of the proposed ChatSync with spatial-temporal reasoning, we compare it with several state-of-the-art resource allocation methods as follows:

- **Jodie** [56]: This framework is designed for resource allocation decisions by analyzing sequential interaction events. It employs the recurrent neural network (RNN) to capture the dynamic nature of graph structures and model their temporal evolution, facilitating resource distribution to the most relevant node pairs.
- **DyRep** [57]: This method focuses on learning node representations and computing node similarity for resource recommendation. It dynamically updates node embeddings based on historical interactions and edge types, making it well-suited for multi-relational graphs where nodes engage in diverse types of interactions.
- **TGAT** [58]: This approach utilizes the graph attention mechanism to aggregate feature information from neighboring nodes, adjusting attention weights based on past events. It generates real-time resource recommendations by evaluating the similarity between nodes' interaction histories.
- **TGN** [59]: This memory-based framework for deep learning on dynamic graphs models both past and current node interactions through a memory module. It prioritizes temporal factors, enabling the allocation of resources to nodes whose past interactions are most indicative of future resource needs.

The summary of the main differences between various methods is shown in Table I.

2) Metrics: To evaluate the model's performance at the task level and the system level, the following metrics are used:

- **Accuracy**: This metric represents the proportion of correct decisions made by the model. It is computed by dividing the number of allocation records that align with expert decisions by the total number of records within each task category.
- **Punctuality rate**: This metric assesses the timeliness of material deliveries, reflecting the sequential decision-making performance at the system level, which is needed to consider allocating resources, vehicles, and operators comprehensively. It is the ratio of resources delivered before the anticipated delivery deadline to the total number of resources transported.

3) Analysis of Training Loss: Fig. 6 presents the training loss trajectories for the proposed ChatSync alongside baseline models, with the shaded areas representing the standard deviation. The results demonstrate that our method achieves rapid convergence and sustains the lowest loss throughout the training process. While TGN exhibits pronounced fluctuations during the initial training stages, it ultimately stabilizes at a low loss value. Conversely, TGAT shows a clear advantage during the early epochs but converges to a performance level comparable to DyRep and Jodie in the later stages.

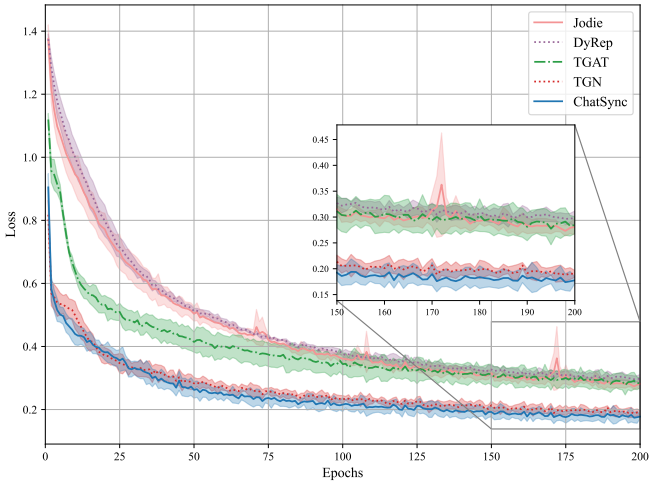

Fig. 6. Comparison of the training loss.

4) Approach Efficiency Study: Table II presents the average task-level accuracy of different approaches. This metric reflects how precisely each model performs in various task types: assigning appropriate resources based on production needs in resource allocation tasks, selecting suitable vehicles according to constraints such as capacity and availability in vehicle matching tasks, and deploying personnel effectively in operator assignment tasks.

TABLE I
COMPARATIVE ANALYSIS OF METHODS FOR RESOURCE ALLOCATION AND RECOMMENDATION

Method	**Temporal Reasoning Capability**	**Spatial Reasoning Capability**	**Strengths**	**Weaknesses**
ChatSync	★★★	★★★	Integrates graph convolutional layers with memory modules to capture complex spatial-temporal dependencies, excelling in dynamic resource allocation and continuously evolving recommendation scenarios.	More computational complexity and memory requirements, potentially limiting scalability and real-time responsiveness on large graphs.
TGN	★★★	★★☆	Uses temporal graph networks with memory to model evolving interactions over time, well suited for sequence-based recommendation.	Spatial reasoning is limited; may underperform when fine-grained spatial relationships are crucial.
TGAT	★★☆	★★☆	Employs self-attention over edges to highlight key events in long sequences, improving prediction accuracy in time-sensitive recommendation tasks.	Lacks deep spatial integration, reducing effectiveness in spatially heterogeneous resource allocation.
DyRep	★★☆	★☆☆	Designed for dynamic graphs to model the evolution of node relationships and interaction intensities over time.	Quite limited spatial modeling capability, which limits its applicability when spatial context influences allocation decisions.
Jodie	★☆☆	☆☆☆	Focuses on continuous-time modeling, beneficial for next-interaction forecasting.	No spatial component; unsuitable for scenarios where location or network structure matters.

TABLE II
TASK ACCURACY (%) ON DIFFERENT METHODS

Method	Resource allocation	Vehicle matching	Operator assignment
Jodie	77.28 ± 0.89	67.14 ± 0.31	79.63 ± 0.37
DyRep	86.32 ± 1.35	88.33 ± 1.52	90.48 ± 1.39
TGAT	87.88 ± 0.69	90.42 ± 0.44	89.77 ± 0.44
TGN	93.90 ± 0.57	96.32 ± 0.25	94.47 ± 0.28
ChatSync	**96.19 ± 0.14**	**96.34 ± 0.38**	**95.71 ± 0.33**

Fig. 7 illustrates the system-level punctuality rate. In a standardized PL workflow, the production planner must first allocate materials according to resource requirements, then assign operators to specific vehicles to transport the materials to their destinations. As a result, any sub-optimal decision in any of the task categories within the same order can cause delays in final material delivery. Consequently, the system-level punctuality rate is consistently lower than the corresponding task-level accuracy across all methods. Additional observations based on the experimental results are summarized as follows:

1) ChatSync demonstrates the best performance across all categories tasks, maintaining an accuracy of around 96%. Although system-level performance is more challenging due to task interdependencies, ChatSync still achieves a delivery punctuality rate of 91.24%. This can be attributed to the effective integration of GCN and node memory mechanism. The graph convolutional layer captures spatial dependencies by aggregating information from neighboring nodes, while the memory mechanism incorporates past allocation trends to predict future resource requirments in an implicit way. This dual module enables the model to comprehensively account for both temporal and spatial dynamics, making it highly effective in handling complex allocation tasks.
2) TGN demonstrates strong and consistent performance across all tasks, with approximately 95% task-level accuracy and 88% system-level punctuality. Although it can achieve similar accuracy in individual task, it lags behind our method by about three percentage points in punctuality rate. This is due to the model's only focus on learning temporal patterns, ignoring extracting feature from embeddings with spatial-temporal information, which limit its ability to handle sequential decision problem.
3) TGAT performs not well in Resource Allocation is not as strong as other models. This is because TGAT focuses heavily on temporal dependencies but does not fully incorporate spatial relationships. In PL, the spatial allocation of resources is as important as their temporal scheduling, and the lack of spatial reasoning in TGAT leads to its lower accuracy in tasks that require careful allocation of resources across the production network.
4) DyRep exhibits competitive performance in operator assignment and vehicle matching, but it struggles with spatial allocation because it lacks a mechanism for efficiently incorporating past allocation patterns, which can lead to suboptimal decisions.
5) Jodie consistently shows the lowest accuracy across all tasks. Although RNN are typically adept at handling temporal dependencies, but they lack effective memory forgetting mechanisms, leading to struggle with large and complex graph network.

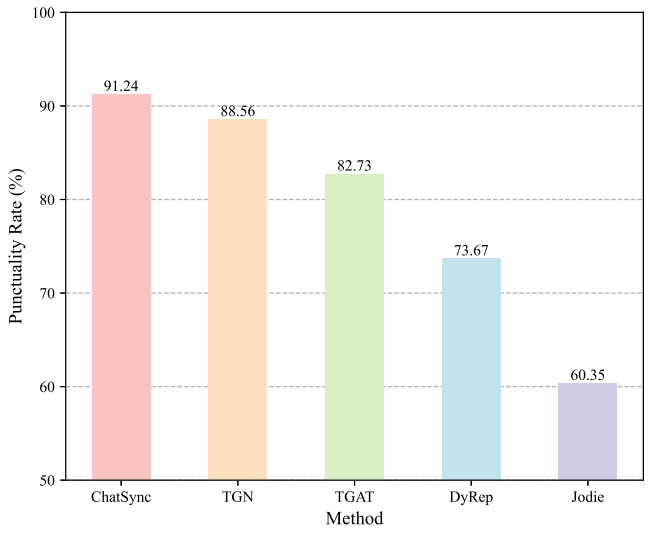

Fig. 7. Punctuality rate for different methods.

C. Phase 2: performance of suggestion generation

1) Baselines and Metrics: In order to assess the performance of ChatSync with contextual reasoning, we introduce the following baseline models for comparative analysis:

- **Base LLM**: An advanced language model comprising 236B parameters. It has been trained from scratch on a vast dataset of 2 trillion tokens in both English and Chinese.
- **LLM + Fine-tuning**: An enhanced version of the Base LLM, fine-tuned on a specialized dataset to improve performance in specific domains or tasks.
- **LLM + Vector Search**: Integrates the LLM with vector search capabilities, allowing for efficient retrieval of semantically similar texts by matching embeddings, thus improving contextual relevance in text generation.

To assess the performance of the models, we use a set of evaluation metrics that quantify both the accuracy and the quality of the generated response:

- **Accuracy**: This is a metric used in cognitive inference to evaluate the logical performance of the LLM. It is calculated by dividing the number of correct answers by the total number of questions, reflecting the model's understanding within a specific domain.
- **ROUGE-1/2/L**: They are metrics used to evaluate the quality of generated text by comparing n-gram overlap and sequence similarity between the model output and reference texts in suggestion generation. ROUGE-1 measures the overlap of unigrams, while ROUGE-2 evaluates bigram overlap. ROUGE-L focuses on the longest common subsequence between the texts, assessing the fluency and coherence of the generated output [60].

2) Model Performance Analysis: Table III presents the performance of different models on the knowledge dataset. In the cognitive inference category, which includes true/false and fill-in-the-blank questions, the accuracy rate is determined by the proportion of correctly answered questions. The base LLM exhibited the lowest accuracy and the primary reason lies in its training on general datasets, which lack relevant knowledge in specific domains, leading to suboptimal performance. In contrast, the other three approaches are trained using professional and expert knowledge, resulting in improved accuracy. The ChatSync with contextual reasoning achieves the highest accuracy by leveraging a knowledge graph to model relevant information, making it easier to extract meaningful insights through multi-hop search. In comparison, fine-tuning and vector search primarily rely on textual similarity for answer matching, which lacks an effective thinking process for complex problems.

In suggestion generation, which includes question-and-answer tasks, all models use expert-provided responses as ground truth to calculate ROUGE-1/2/L scores. Higher values of these metrics indicate closer alignment between the model's output and the expert-generated text. ChatSync equipped with contextual reasoning achieves the best performance across all metrics. It can also be found that the ROUGE-L score often falls between ROUGE-1 and ROUGE-2. This is because ROUGE-1 only considers the matching of individual words between the generated answers and the ground truth, while ROUGE-2 evaluates the overlap of consecutive word pairs, resulting in scores typically lower than ROUGE-1. On the other hand, ROUGE-L incorporates sentence-level matching by combining word overlap and sequence alignment, capturing more extended matching patterns. Consequently, the ROUGE-L score usually falls between the ROUGE-1 and ROUGE-2 scores.

3) Text Generation Analysis: Taking the XA-7 series cooling fan blades produced by the company as an example, Table IV presents the handling suggestion generated by each LLM for this product. The table includes the answers of various models and expert comments on those responses. The base LLM, lacking training in specialized domains, can only provide general suggestions and fails to give precise answers tailored to this product, so it is not shown in the table. The responses based on fine-tuning techniques are notably lengthy and not optimized for guiding frontline workers. Additionally, fine-tuning faces challenges with knowledge updates. If the operation guidance changes due to technological advancements in the company's products, the knowledge embedded in the LLM becomes outdated. In this case, Updating the knowledge base of LLM requires retraining or incremental learning, both of which increase training costs and may lead to catastrophic forgetting. LLM that integrate vector search and contextual reasoning are more suitable for long-term knowledge base maintenance as they improve the knowledge retrieval process rather than embed knowledge directly into the model's parameters, as fine-tuning does. However, vector search primarily matches based on textual similarity without analyzing relationships between text objects. As a result, the generated text may be excessively verbose and fail to deliver concise information.

4) Effect of Personalized Generation: In addition to providing more precise operational guidance, ChatSync incorporates dynamic prompts to support personalized suggestions regarding various scenarios. Table V displays the responses to various prompts, with bold text indicating the parts that change

TABLE III
PERFORMANCE IN TEXT ANALYSIS AND SUGGESTION GENERATION

Method	Cognitive Inference	Suggestion Generation		
	Accuracy (%)	ROUGE-1	ROUGE-2	ROUGE-L
Base LLM	76.05	0.1674	0.0586	0.1091
LLM + Fine-tuning	81.69	0.2382	0.0925	0.1509
LLM + Vector Search	84.51	0.2706	0.1004	0.1773
ChatSync	**90.14**	**0.3597**	**0.1276**	**0.2350**

based on the operator's real-time work situation. For example, when operators are handling materials for the first time, ChatSync will use an apprentice prompt to provide detailed but concise guidance. In contrast, the master prompt is used to reduce the volume of information for operators who have transported the same material multiple times, retaining only key steps to minimize redundancy and reduce worker fatigue. Similarly, the misoperation prompt and deviation prompt are designed to address specific past errors, such as operating an unauthorized vehicle or delivering materials to the wrong destination, by providing targeted reminders to prevent recurrence. In summary, the use of dynamic prompts in ChatSync enables the system to adapt operational guidance to operators' individual needs and past behaviours, enhancing efficiency while minimizing errors in diverse working environments.

D. Phase 3: performance of resource traceability

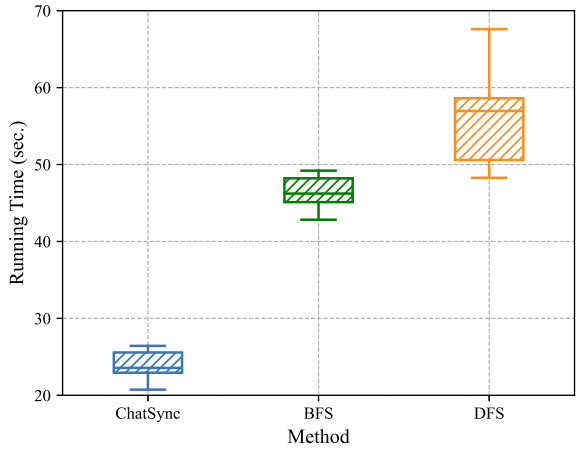

Fig. 8. Running time for different methods.

1) Baselines and Metrics: In resource traceability, ChatSync leverages relational reasoning to achieve rapid resource identification, providing logistics coordinators with a complete and clear overview of resource flow. To evaluate the search performance of ChatSync within RSTG, we compared it with two classic graph search algorithms:

- **Breadth-First Search (BFS)** [61]: This approach can systematically explore all nodes at the current depth level before progressing to nodes at the next depth level. BFS utilizes a queue data structure to manage and track vertices pending exploration, making it particularly effective for a layer-by-layer traversal of the graph.
- **Depth-First Search (DFS)** [62]: This algorithm delves as deeply as possible along each branch of the graph before backtracking to explore alternative paths. It is especially advantageous for tracing resource dependencies within hierarchical structures or long chains, enabling uninterrupted exploration of a resource path from its origin to its termination.

2) Result: The time required by different algorithms to trace the same resource is presented in Figure 8. Our proposed method, ChatSync with relational reasoning, achieves the shortest average time, approximately 23.8 seconds, compared to 46.3 seconds and 56.4 seconds for the other two algorithms. From the boxplots, it can be observed that the standard deviations of BFS and our method are nearly identical, indicating relatively stable performance. In contrast, DFS exhibits significant variability, as evidenced by the larger interquartile range. This variability stems from the depth-first nature of DFS, which prioritizes exploring an entire branch of the graph before backtracking to search other directions. If the target node is not within the current branch, the algorithm still requires exhaustively exploring all nodes in that branch, making its search time heavily dependent on the choice of branch and resulting in more significant fluctuations. BFS, on the other hand, employs a layered search strategy, avoiding deep traversal into irrelevant branches. This leads to a reduction in time consumption compared to DFS. ChatSync, with relational reasoning, further optimizes the process by prioritizing branches based on the relationships between nodes, actively targeting branches with a higher likelihood of containing the desired resource. Consequently, it achieves the lowest time consumption among all algorithms.

E. Managerial implications

Based on the experimental analyses, we derive the following managerial implications for optimizing PL workflow:

The chat-based interface is particularly effective for frontline workers, as it removes the need for specialized computer skills. This allows workers to access necessary task information and resource availability with ease, significantly reducing cognitive load and improving overall task execution. By simplifying interactions and focusing on the essentials, workers are able to perform their duties more efficiently without the burden of complex systems or technical training.

TABLE IV
SUGGESTIONS GENERATED BY DIFFERENT METHODS

Method	Handling Suggestion	Expert Comments
LLM + Fine-tuning	1. Selection and Installation of Cushion Pads Objective: To provide adequate protection to fan blades during handling and transport by ensuring proper cushioning. Procedure: • Material Selection: Use T-300 cushion pads exclusively. • Coverage: Ensure full coverage of contact surfaces. • Installation: Place pads on contact points and secure them within support brackets. • Vibration and Friction Prevention: Double-check securement to prevent vibrations or friction. 2. Securing and Stabilizing Fan Blades Objective: To ensure fan blades remain stable and secure within the protective frame during transport. Procedure: • Fixing Within Protective Frame: Position blades correctly within the frame. • Use of Rubber Straps: Utilize rubber straps to tie down blades securely. • Stability Check: Perform a stability check by gently shaking the frame. 3. Spacing and Use of Protective Dividers Objective: To prevent overlap or direct contact between fan blades, especially for fragile models. Procedure: • Spacing Requirement: Maintain a minimum spacing of 10 cm between each set of fan blades. • Insertion of Protective Dividers: Insert dividers between blades made of impact-absorbing materials. • Special Considerations for Fragile Models: Use additional protective layers or robust materials for fragile blades. 4. Inspection and Confirmation Objective: To verify that all securing and cushioning measures are in place before proceeding with transport. Procedure: • Post-Loading Inspection: Conduct a thorough inspection after loading the blades. • Post-Unloading Inspection: Perform another inspection upon unloading. • Confirmation: Inspect document results and correct any actions taken. **(227 words)**	From a content perspective, this is a typical GPT-style answer. It includes some information that is irrelevant for the operational phase, such as the "Objective." While this detailed response is suitable for new employee training, it adds unnecessary information overload for employees who handle numerous transport tasks daily.
LLM + Vector Search	1. Selection and Installation of Cushion Pad • Use T-300 cushion pads exclusively, as they are designed to absorb shock and reduce friction. • Ensure full coverage of the fan blades' contact surfaces, including leading and trailing edges, tips, and bases. • Place cushion pads directly on contact points where the blades will rest. • Double-check pad security to prevent vibrations. 2. Securing and Stabilizing Fan Blades • Position each blade within its dedicated protective frame that matches its dimensions and weight. • Use rubber straps in a crisscross pattern to secure the blades within the frame, ensuring no movement during transport. • Perform a stability check by gently shaking the frame to confirm that the blades do not move or detach. 3. Spacing and Use of Protective Dividers • Maintain a minimum spacing of 10 cm between each set of fan blades to prevent overlap or direct contact. • Insert protective dividers made from impact-absorbing materials like foam or rubber between each set of blades. • Use additional protective layers or robust materials for fragile blades. 4. Inspection and Confirmation • Conduct post-loading inspections to verify securing measures before transport. • Check for movement, and re-tighten straps or adjust blade positioning if detected. • Upon unloading, inspect again for any signs of movement or damage during transport. **(205 words)**	The answer includes the necessary operational steps but is somewhat too detailed and lengthy. In practice, operators receive tasks and related suggestions via their mobile phones, so recommendations should not be too long, as they may not fit on a single screen.
Chat-Sync	1. Selection and Installation of Cushion Pads • Use T-300 cushion pads exclusively. • Ensure pads fully cover blade contact surfaces. • Place pads directly on contact points and secure them with fasteners. • Double-check pad security to prevent vibrations. 2. Securing and Stabilizing Fan Blades • Position blades correctly in protective frames. • Use rubber straps in a crisscross pattern to secure blades. • Perform a stability check by gently shaking the frame. 3. Spacing and Use of Protective Dividers • Maintain 10 cm spacing between blades. • Insert protective dividers (foam or rubber) between blades. • For fragile models, use additional protective layers. 4. Inspection and Confirmation • Conduct thorough inspections post-loading and post-unloading. • Check cushion pads, straps, and dividers. • Inspect documents and correct actions. **(116 words)**	This response has significantly reduced word count compared to other versions, and each step is concise, enabling operators to quickly grasp the key points. By eliminating unnecessary details, it enhances clarity and makes it easier for operators to follow, leading to more efficient execution of tasks.

TABLE V
RESULT OF PERSONALIZED SUGGESTION GENERATION

Method	Response
ChatSync with Apprentice Prompt	Your task is to operate forklift CK953 to transport 10 sets of XA-7 fan blades from Shelf 8 in Zone A to Shelf 4 in Zone C. The forklift is located in Zone B near the entrance. Once in Zone C, place the fan blades in the designated area, ensuring they are properly aligned and organized according to the layout of the zone. **Before transporting the cooling fan blades, follow these handling guidelines:** **1. Selection and Installation of Cushion Pads** • **Use T-300 cushion pads exclusively.** • **Ensure pads fully cover blade contact surfaces.** • **Place pads directly on contact points and secure them with fasteners.** • **Double-check pad security to prevent vibrations.** **2. Securing and Stabilizing Fan Blades** • **Position blades correctly in protective frames.** • **Use rubber straps in a crisscross pattern to secure blades.** • **Perform a stability check by gently shaking the frame.** **3. Spacing and Use of Protective Dividers** • **Maintain 10 cm spacing between blades.** • **Insert protective dividers (foam or rubber) between blades.** • **For fragile models, use additional protective layers.** **4. Inspection and Confirmation** • **Conduct thorough inspections post-loading and post-unloading.** • **Check cushion pads, straps, and dividers.** • **Inspect documents and correct actions.**
ChatSync with Master Prompt	Your task is to operate forklift CK953 to transport 10 sets of XA-7 fan blades from Shelf 8 in Zone A to Shelf 4 in Zone C. The forklift is located in Zone B near the entrance. Once in Zone C, place the fan blades in the designated area, ensuring they are properly aligned and organized according to the layout of the zone. **Before transporting the cooling fan blades, follow these handling guidelines:** • **Use T-300 pads exclusively. Ensure full coverage on contact surfaces and secure firmly in brackets to prevent movement.** • **Position blades correctly in protective frames and secure with crisscross rubber straps, ensuring no movement.** • **Maintain 10 cm between blades and use impact-absorbing dividers. For fragile models, consider additional layers.** • **Post-loading and post-unloading, verify all securing measures are intact. Document any issues and corrective actions.**
ChatSync with Misoperation Prompt	Your task is to operate forklift CK953 to transport 10 sets of XA-7 fan blades from Shelf 8 in Zone A to Shelf 4 in Zone C. The forklift is located in Zone B near the entrance. Once in Zone C, place the fan blades in the designated area, ensuring they are properly aligned and organized according to the layout of the zone. **It seems that the system has noticed you've driven the wrong vehicle a few times recently. Remember, for this task, you should be using forklift CK953. To avoid any mix-ups in the future, please take a moment to double-check the vehicle ID before you start driving. This will help ensure you're using the correct equipment for the job.** Before transporting the cooling fan blades, follow these handling guidelines: ... ...
ChatSync with Deviation Prompt	Your task is to operate forklift CK953 to transport 10 sets of XA-7 fan blades from Shelf 8 in Zone A to Shelf 4 in Zone C. The forklift is located in Zone B near the entrance. Once in Zone C, place the fan blades in the designated area, ensuring they are properly aligned and organized according to the layout of the zone. **The system has detected that you have gone to the wrong destination in the past, which resulted in a delivery delay. Please confirm again that the destination of this task is Zone C. Please do not make any mistakes again.** Before transporting the cooling fan blades, follow these handling guidelines: ... ...

When allocating resources, considering the spatial and temporal relationships between different elements is crucial for optimizing operational efficiency. By leveraging real-time data, systems can ensure that workers, vehicles, and materials are deployed at the right time and location, preventing delays and improving overall workflow. Understanding these relationships allows for more dynamic decision-making, helping to avoid resource shortages or unnecessary idle time, which ultimately boosts productivity.

Conducting standardized checks before workers begin any task is essential to ensure both accuracy and consistency. However, the complexity of managing diverse materials in large-scale operations can make it challenging to consult reference materials or manuals constantly. To overcome this, offering context-specific suggestions tailored to the worker's past performance can streamline the process. By analyzing historical data, systems can provide personalized guidance, helping workers complete tasks more efficiently while reducing the risk of errors and minimizing product loss.

VII. CONCLUSION

This paper presents the ChatSync framework, designed to improve work efficiency through interactive communication with the system. The novelty of this work is threefold. First,

the proposed RSTG integrates information and knowledge domains, facilitating knowledge fusion for advanced analysis. Second, we propose spatial-temporal, contextual, and relational reasoning mechanisms to enhance decision-making in allocation recommendation, suggestion generation, and resource traceability. Third, ChatSync optimizes workflow in the PL chain by allowing various roles to access critical information via chat. It also provides personalized guidance for frontline workers, reducing cognitive load.

Despite the promising results, this work has several limitations. First, as the transfer records accumulate, the RSTG will gradually become a dense graph, which could complicate the process of identifying key information from an increasingly large set of adjacent nodes. Second, when the knowledge within the RSTG needs updating, such as when advancements in the fan manufacturing process lead to changes in the latest transportation standards, removing outdated knowledge nodes and establishing new connections can be a time-consuming task. This challenge of dynamic knowledge management in ChatSync could affect the system's overall responsiveness and efficiency in immediate or live scenarios.

Future research will focus on improving the scalability and adaptability of the RSTG within ChatSync to overcome these limitations. One key direction will be designing more efficient algorithms for extracting and prioritizing critical information from a dense graph, using sampling techniques or hierarchical graph structures to simplify the identification of relevant nodes. Additionally, methods for faster and more automated knowledge updates will be explored, such as creating indexes for each node in the knowledge domain to enable rapid identification of all affected nodes when the RSTG is updated.

REFERENCES

[1] Y. Zhu, J. Cheng, Z. Liu, Q. Cheng, X. Zou, H. Xu, Y. Wang, and F. Tao, "Production logistics digital twins: Research profiling, application, challenges and opportunities," *Robotics and Computer-Integrated Manufacturing*, vol. 84, p. 102592, 2023.

[2] A. Napoleone, E. Moretti, M. Macchi, and M. Melacini, "Synchronisation of material flows in mass-customised production systems: a literature-based classification framework and industrial application," *Production Planning & Control*, vol. 35, no. 14, pp. 1760–1778, 2024.

[3] D. Ivanov, "The industry 5.0 framework: viability-based integration of the resilience, sustainability, and human-centricity perspectives," *International Journal of Production Research*, vol. 61, no. 5, pp. 1683–1695, 2023.

[4] G. MacKerron, M. Kumar, V. Kumar, and A. Esain, "Supplier replenishment policy using e-kanban: A framework for successful implementation," *Production Planning & Control*, vol. 25, no. 2, pp. 161–175, 2014.

[5] C. Silva, L. M. Ferreira, M. Thürer, and M. Stevenson, "Improving the logistics of a constant order-cycle kanban system," *Production planning & control*, vol. 27, no. 7-8, pp. 650–659, 2016.

[6] P. Liu, L. Qian, X. Zhao, and B. Tao, "Joint knowledge graph and large language model for fault diagnosis and its application in aviation assembly," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 6, pp. 8160–8169, 2024.

[7] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[8] Z. Zhu, Y. Zhao, S. Qiu, K. Xu, Q. Yin, J. Huang, Z. Liu, and F.-Y. Wang, "Conversational crowdsensing in the age of industry 5.0: A parallel intelligence and large models powered novel sensing approach," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 6, pp. 8046–8063, 2024.

[9] W. Yan, B. Hu, Y. li Liu, C. Li, and C. Song, "Does usage scenario matter? investigating user perceptions, attitude and support for policies towards chatgpt," *Information Processing & Management*, vol. 61, no. 6, p. 103867, 2024.

[10] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q. L. Han, and Y. Tang, "A brief overview of chatgpt: The history, status quo and potential future development," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1122–1136, 2023.

[11] D. Guo, R. Y. Zhong, Y. Rong, and G. G. Q. Huang, "Synchronization of shop-floor logistics and manufacturing under iiot and digital twin-enabled graduation intelligent manufacturing system," *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 2005–2016, 2023.

[12] M.-T. H. Stanislav Chankov and J. Bendul, "Synchronization in manufacturing systems: quantification and relation to logistics performance," *International Journal of Production Research*, vol. 54, no. 20, pp. 6033–6051, 2016.

[13] H. Wu, M. Li, C. Yu, Z. Ouyang, K. hung Lai, Z. Zhao, S. Pan, S. Wang, R. Y. Zhong, Y.-H. Kuo, F. Zhang, W. Huang, Z.-J. M. Shen, E. Ballot, and G. Q. Huang, "Towards cyber-physical internet: A systematic review, fundamental model and future perspectives," *Transportation Research Part E: Logistics and Transportation Review*, vol. 197, p. 104051, 2025.

[14] Y. SUGIMORI, K. KUSUNOKI, F. CHO, and S. UCHIKAWA, "Toyota production system and kanban system materialization of just-in-time and respect-for-human system," *International Journal of Production Research*, vol. 15, no. 6, pp. 553–564, 1977.

[15] Y. Yin, K. E. Stecke, and D. Li, "The evolution of production systems from industry 2.0 through industry 4.0," *International Journal of Production Research*, vol. 56, no. 1-2, pp. 848–861, 2018.

[16] Y. Pan, R. Y. Zhong, T. Qu, L. Ding, and J. Zhang, "Multi-level digital twin-driven kitting-synchronized optimization for production logistics system," *International Journal of Production Economics*, vol. 271, p. 109176, 2024.

[17] Z. Zhang, T. Qu, K. Zhao, K. Zhang, Y. Zhang, W. Guo, L. Liu, and Z. Chen, "Enhancing trusted synchronization in open production logistics: A platform framework integrating blockchain and digital twin under social manufacturing," *Advanced Engineering Informatics*, vol. 61, p. 102404, 2024.

[18] D. Guo, "Fast scheduling of human-robot teams collaboration on synchronised production-logistics tasks in aircraft assembly," *Robotics and Computer-Integrated Manufacturing*, vol. 85, p. 102620, 2024.

[19] M. Tonizza Pereira and M. Seido Nagano, "Hybrid metaheuristics for the integrated and detailed scheduling of production and delivery operations in no-wait flow shop systems," *Computers and Industrial Engineering*, vol. 170, p. 108255, 2022.

[20] Z. Zhang, Z. Zhu, J. Zhang, and J. Wang, "Construction of intelligent integrated model framework for the workshop manufacturing system via digital twin," *The International Journal of Advanced Manufacturing Technology*, vol. 118, pp. 1–14, 02 2022.

[21] Z. Zhao, X. Xie, Q. Chen, P. Lin, W. Wu, M. Zhang, and G. Q. Huang, "Graduation-inspired manufacturing system and synchronization mechanism for hybrid assembly cell lines," *Computers and Industrial Engineering*, vol. 198, p. 110648, 2024.

[22] Y. Jiang, M. Li, B. J. Ma, R. Y. Zhong, and G. Q. Huang, "Data-driven out-of-order model for synchronized planning, scheduling, and execution in modular construction fit-out management," *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 16, pp. 2457–2480, 2024.

[23] B. Aas and S. W. Wallace, "Management of logistics planning," in *Information technologies, methods, and techniques of supply chain management*. IGI Global, 2012, pp. 255–271.

[24] A. B. Senouci and H. Adeli, "Resource scheduling using neural dynamics model of adeli and park," *Journal of Construction Engineering and Management*, vol. 127, no. 1, pp. 28–34, 2001.

[25] T. Qu, Y. Pan, X. Liu, K. Kang, C. Li, M. Thurer, and G. Q. Huang, "Internet of things-based real-time production logistics synchronization mechanism and method toward customer order dynamics," *Transactions of the Institute of Measurement and Control*, vol. 39, no. 4, pp. 429–445, 2017.

[26] S. Z. Shaohua Huang, Yu Guo and Y. Wang, "An internet-of-things-based production logistics optimisation method for discrete manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 1, pp. 13–26, 2019.

[27] Y. Liu, S. Pan, and E. Ballot, "Unveiling the potential of digital twins in logistics and supply chain management: Services, capabilities, and research opportunities," *Digital Engineering*, vol. 3, p. 100025, 2024.

[28] Y. Zhang, Z. Guo, J. Lv, and Y. Liu, "A framework for smart production-logistics systems based on cps and industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4019–4032, 2018.

[29] Z. Zhao, M. Zhang, J. Chen, T. Qu, and G. Q. Huang, "Digital twin-enabled dynamic spatial-temporal knowledge graph for production logistics resource allocation," *Computers and Industrial Engineering*, vol. 171, p. 108454, 2022.

[30] S. Iftikhar, S. S. Gill, C. Song, M. Xu, M. S. Aslanpour, A. N. Toosi, J. Du, H. Wu, S. Ghosh, D. Chowdhury, M. Golec, M. Kumar, A. M. Abdelmoniem, F. Cuadrado, B. Varghese, O. Rana, S. Dustdar, and S. Uhlig, "Ai-based fog and edge computing: A systematic review, taxonomy and future directions," *Internet of Things*, vol. 21, p. 100674, 2023.

[31] Y. Pan, T. Qu, N. Wu, M. Khalgui, and G. Huang, "Digital twin based real-time production logistics synchronization system in a multi-level computing architecture," *Journal of Manufacturing Systems*, vol. 58, pp. 246–260, 2021, digital Twin towards Smart Manufacturing and Industry 4.0.

[32] C. Aron, F. Sgarbossa, E. Ballot, and D. Ivanov, "Cloud material handling systems: a cyber-physical system to enable dynamic resource allocation and digital interoperability," *Journal of Intelligent Manufacturing*, vol. 35, no. 8, pp. 3815–3836, Dec 2024.

[33] OpenAI, "Gpt-4 technical report," 2024, available at https://arxiv.org/abs/2303.08774.

[34] H. Cui, Y. Du, Q. Yang, Y. Shao, and S. C. Liew, "Llmind: Orchestrating ai and iot with llm for complex task execution," *IEEE Communications Magazine*, pp. 1–7, 2024.

[35] C. Zhang, Q. Xu, Y. Yu, G. Zhou, K. Zeng, F. Chang, and K. Ding, "A survey on potentials, pathways and challenges of large language models in new-generation intelligent manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 92, p. 102883, 2025.

[36] B. Zhou, X. Li, T. Liu, K. Xu, W. Liu, and J. Bao, "Causalkgpt: Industrial structure causal knowledge-enhanced large language model for cause analysis of quality problems in aerospace product manufacturing," *Advanced Engineering Informatics*, vol. 59, p. 102333, 2024.

[37] T. Wang, J. Fan, and P. Zheng, "An llm-based vision and language cobot navigation approach for human-centric smart manufacturing," *Journal of Manufacturing Systems*, vol. 75, pp. 299–305, 2024.

[38] S. Colabianchi, F. Costantino, and N. Sabetta, "Assessment of a large language model based digital intelligent assistant in assembly manufacturing," *Computers in Industry*, vol. 162, p. 104129, 2024.

[39] H. Fan, X. Liu, J. Y. H. Fuh, W. F. Lu, and B. Li, "Embodied intelligence in manufacturing: leveraging large language models for autonomous industrial robotics," *Journal of Intelligent Manufacturing*, vol. 36, no. 2, pp. 1141–1157, 2025.

[40] Y. Tsushima, S. Yamamoto, A. A. Ravankar, J. V. S. Luces, and Y. Hirata, "Task planning for a factory robot using large language model," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2383–2390, 2025.

[41] W. Yu, J. Lv, W. Zhuang, X. Pan, S. Wen, J. Bao, and X. Li, "Rescheduling human-robot collaboration tasks under dynamic disassembly scenarios: An mllm-kg collaboratively enabled approach," *Journal of Manufacturing Systems*, vol. 80, pp. 20–37, 2025.

[42] C. Qian, W. Sun, and Y. Zhang, "Resourcenet: a collaboration network among decentralised manufacturing resources for autonomous exception-handling in smart manufacturing," *IET Collaborative Intelligent Manufacturing*, vol. 2, no. 3, pp. 109–114, 2020.

[43] S. Cheng, C. Chen, S. Pan, H. Huang, W. Zhang, and Y. Feng, "Citywide package deliveries via crowdshipping: minimizing the efforts from crowdsourcers," *Frontiers of Computer Science*, vol. 16, pp. 1–13, 2022.

[44] W. Wu, L. Shen, Z. Zhao, M. Li, and G. Q. Huang, "Industrial iot and long short-term memory network-enabled genetic indoor-tracking for factory logistics," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7537–7548, 2022.

[45] P. Li, W. Wu, Z. Zhao, and G. Q. Huang, "Indoor positioning systems in industry 4.0 applications: Current status, opportunities, and future trends," *Digital Engineering*, vol. 3, p. 100020, 2024.

[46] Z. Zhao, M. Zhang, W. Wu, G. Q. Huang, and L. Wang, "Spatial-temporal traceability for cyber-physical industry 4.0 systems," *Journal of Manufacturing Systems*, vol. 74, pp. 16–29, 2024.

[47] J. Lee, S. Ahn, D. Kim, and D. Kim, "Performance comparison of retrieval-augmented generation and fine-tuned large language models for construction safety management knowledge retrieval," *Automation in Construction*, vol. 168, p. 105846, 2024.

[48] L. Yang, H. Chen, Z. Li, X. Ding, and X. Wu, "Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3091–3110, 2024.

[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019.

[50] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing chinese characters with recurrent neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 849–862, 2018.

[51] S. Deena, M. Hasan, M. Doulaty, O. Saz, and T. Hain, "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition and alignment," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 572–582, 2019.

[52] M. Adjeisah, X. Zhu, H. Xu, and T. A. Ayall, "Towards data augmentation in graph neural network: An overview and evaluation," *Computer Science Review*, vol. 47, p. 100527, 2023.

[53] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker, "Time2vec: Learning a vector representation of time," 2019, available at https://arxiv.org/abs/1907.05321.

[54] P. Zhang, S. Xiao, Z. Liu, Z. Dou, and J.-Y. Nie, "Retrieve anything to augment large language models," 2023, available at https://arxiv.org/abs/2310.07554.

[55] DeepSeek-AI, "Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model," 2024, available at https://arxiv.org/abs/2405.04434.

[56] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1269–1278.

[57] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Dyrep: Learning representations over dynamic graphs," in *International conference on learning representations (ICLR)*, 2019.

[58] da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan, "Inductive representation learning on temporal graphs," in *International conference on learning representations (ICLR)*, 2020.

[59] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," in *International Conference on Machine Learning (ICML)*, 2020.

[60] M. Zhang, C. Li, M. Wan, X. Zhang, and Q. Zhao, "Rouge-sem: Better evaluation of summarization using rouge combined with semantics," *Expert Systems with Applications*, vol. 237, p. 121364, 2024.

[61] N. Banerjee, S. Chakraborty, V. Raman, and S. R. Satti, "Space efficient linear time algorithms for bfs, dfs and applications," *Theory of Computing Systems*, vol. 62, no. 8, pp. 1736–1762, 2018.

[62] S. Alpern and T. Lidbetter, "Search and delivery man problems: When are depth-first paths optimal?" *European Journal of Operational Research*, vol. 285, no. 3, pp. 965–976, 2020.