计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(10): 2715−2732, 2026

基于大语言模型的库存调度决策方法

林　丽[1,2]　李宗晟[3]　王　帅[1]

[1]（东南大学计算机科学与工程学院　南京　211189）

[2]（新一代人工智能技术与交叉应用教育部重点实验室（东南大学）　南京　211189）

[3]（北京京东远升科技有限公司　北京　100176）

（linli321@seu.edu.cn）

Inventory Scheduling Decision Method Based on Large Language Models

Lin Li[1,2], Li Zongsheng[3], and Wang Shuai[1]

[1]（*School of Computer Science and Engineering*, *Southeast University*, *Nanjing* 211189）

[2]（*Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications* (*Southeast University*), *Ministry of Education*, *Nanjing* 211189）

[3]（*Beijing JD Yuansheng Technology Co.*, *Ltd.*, *Beijing* 100176）

Abstract　Inventory scheduling plays a critical role in modern supply chains, with its primary objective being to balance production and demand through appropriate inventory allocation. As market demand becomes increasingly uncertain and product attributes more diverse, traditional experience-based inventory scheduling methods often struggle to effectively integrate product characteristics with demand fluctuations. Moreover, their decision outcomes lack interpretability, making it difficult to trace the rationale behind inventory schedules and reducing stakeholders' trust. The semantic understanding and knowledge-reasoning capabilities of pre-trained large language models, when combined with abundant real-world multi-source external data and multi-category demand data, offer opportunities to enhance existing LLM-based inventory scheduling methods. To capitalize on these opportunities, LLM-based inventory scheduling decision tasks have to face the following challenges: 1) how to effectively embed the inventory scheduling workflow into a multi-step reasoning process, yielding accurate and coherent reasoning trajectories; 2) how to balance the interpretability of allocation decisions with decision accuracy, avoiding a decline in interpretability caused by excessive focus on accuracy. To this end, we propose a novel multi-step reasoning framework for inventory scheduling based on large language models, termed InvLLM, with following primary contributions: the design of a self-calibrating multi-step reasoning method based on the inventory scheduling workflow and a multi-objective balanced optimization method for inventory scheduling decisions. Additionally, to address distortion in the reasoning fine-tuning phase, a "planning-token" supervised fine-tuning approach is proposed, combined with a generative reward model, achieving a balance between high accuracy and logical consistency. Experimental evaluations on a large-scale real-world dataset provided by JD.com demonstrate that, compared with existing inventory scheduling methods, the proposed InvLLM framework increases service rate by 3.24% and improves overall inventory evaluation scores by 1.55%, while reducing inventory rate by 1.33%, thereby validating its effectiveness and practical applicability.

收稿日期：2025-11-24；**修回日期**：2026-05-29

基金项目：国家自然科学基金青年科学基金项目（62402105）；江苏省自然科学基金项目（BK20230815）

This work was supported by the National Natural Science Foundation of China for Young Scientists (62402105) and the Natural Science Foundation of Jiangsu Province (BK20230815).

通信作者：王帅（shuaiwang@seu.edu.cn）

Key words inventory management; decision making; large language models; multi-step reasoning; reinforcement learning

摘 要 库存调度在现代供应链中扮演着至关重要的角色,其核心目标在于通过合理的库存调度平衡生产与需求之间的关系。然而,随着市场需求的不确定性和商品属性的多样化,基于经验的传统库存调度方法往往难以有效地将商品属性与需求波动结合考虑,并且其决策结果缺乏可解释性,使得其库存调度结果难以追溯原因且信任度较低。另一方面,大语言模型的出现为库存调度决策问题的可解释性提供了新的解决方案。然而,直接将库存调度流程中的补货策略引入推理过程会导致推理过程的误差积累,导致推理结果不稳定。针对上述问题,不仅要考虑推理过程的合理性,还要关注最终决策效果的优化。为了解决上述挑战,一种基于大语言模型的库存调度多步推理框架 InvLLM 被提出。主要贡献如下:1)基于库存调度工作流程设计了一种自校准的多步推理方法;2)提出了一种用于库存调度决策的多目标平衡优化方法。此外,为了解决推理微调阶段的偏差问题,提出了一种结合生成奖励模型的“词元规划”监督微调方法,实现了高准确性与逻辑一致性之间的平衡。通过在中国电子商务公司“京东”提供的大规模真实数据集上的实验评估,证明了所提出的 InvLLM 框架相较于现有的库存调度方法在服务率和总体库存评价分数上分别提高了 3.24% 和 1.55%,并降低了 1.33% 的库存率,验证了该方法的有效性和实用性。

关键词 库存管理;决策生成;大语言模型;多步推理;强化学习算法

中图法分类号 TP391.1; TP181

DOI: 10.7544/issn1000-1239.202550842 **CSTR:** 32373.14.issn1000-1239.202550842

库存管理作为企业供应链管理中的核心环节,已经逐渐成为推动企业高效运营和业务增长的关键组成部分[1]。其主要目标不仅仅是保障供应链的高效运作,还在于最小化交货延迟、最大化服务质量和客户满意度[2]。合理的库存调度决策,可以有效解决供给与需求之间的平衡问题[3]。在现实应用中,为了应对下游客户需求的高度不确定性,企业必须储备足够的库存以应对可能的需求波动。库存过多导致库存成本的上升,进一步影响到资金的流动性,从而可能对企业的财务状况产生负面影响;而库存过少则无法及时响应市场需求,进而导致潜在的销售损失以及客户满意度下降[4]。因此,如何制定合理的库存调度决策,在确保满足客户需求的同时,又能有效控制成本和风险,已成为优化供应链管理和提升企业竞争力的核心问题[5]。

在库存调度流程中,决策者的补货决策不仅需要考虑整体库存水平,还需要对库存中的每一个产品单元单独进行补货计划的制定。这一过程不仅要应对市场需求的随机波动,还需要在库存成本、持有风险与市场需求之间找到一个平衡点。并且不同的产品单元的属性特征在补货决策中起着至关重要的作用,对决策过程产生深远影响。具体而言,不同产品单元具有不同的周转速度、季节性变化等属性,这些因素直接影响着产品单元的补货周期和数量。因此,如何精确捕捉产品单元的个性化特征,并将其有效融入补货决策模型中,成为了现代库存管理中的一大挑战。

当前,大多数库存调度决策方法仍然依赖于基于经验的库存控制模型[6-7]。这些传统模型通常依赖于经验丰富的专家知识和高度复杂的数学公式,适用于处理规则性强且环境相对稳定的库存管理场景。然而,在需求高度不确定的市场环境中,这些模型往往无法有效应对复杂和动态的变化。特别是当考虑到多种产品单元属性信息时,这些传统模型未能有效地将产品单元特性与需求波动相结合。此外,现有库存管理系统缺乏足够的可解释性,难以帮助决策者理解背后的原因和逻辑,这使得决策结果的可追溯性和信任度较低,从而影响后续的决策和系统优化。

近年来,大语言模型(large language models, LLM)的迅速崛起为库存调度决策问题提供了一个全新的解决方案。现有研究表明,LLM 在自然语言理解方面展现了强大的能力,尤其是需求预测、库存管理和供应链优化等方面[8-9]。通过利用 LLM,企业可以从大量历史数据中提取有价值的信息,为库存调度决策提供数据支持,帮助发现可能的趋势和规律[10-11]。此外,LLM 强大的自然语言生成能力使其能够在决策过程中生成易于理解、透明且详细的决策解释,从

而提升了决策系统的可解释性和用户信任度[12-13]。通过这种方式，企业能够更加清晰地理解每次库存调度决策的合理性，并能为后续的优化与改进提供反馈信息，最终实现供应链管理的持续优化。

然而，使用 LLM 进行库存调度决策时，也面临以下 2 个显著的挑战：

1）在设计基于库存调度流程的多步推理方法时，任何单一推理环节的失误都可能累积误差，从而严重影响整体推理链的可靠性。库存调度流程文本往往蕴含大量复杂的业务规则、上下游关联信息以及专业术语，这就要求模型在每一步推断时正确理解并传递上一阶段的输出。具体而言，当模型在某一节点产生理解偏差或逻辑漏洞后，这种偏差不仅无法被后续步骤所纠正，反而会伴随推理链继续传递，并在随后的推断过程中逐步放大。尤其是在采用思维链技术引导多步推理时，模型需要在每一次输出后准确地将其作为新的输入，以维持推理链的完整性与一致性。但现实应用中，由于上下文长度限制、领域术语的歧义性以及模型自身的表达能力瓶颈，思维链常常会出现信息丢失、关键事实错置或逻辑跳跃等问题。即使模型已对库存管理的核心概念和操作流程有初步掌握，但对某一库存阈值条件的误判或对补货策略参数的错误引用都将随着推理层级的加深而被放大，最终导致生成的决策建议与实际业务需求出现显著偏差。

2）如果直接基于多步推理过程生成的补货量设计奖励函数，则模型可能会牺牲其生成的推理轨迹准确性。因此，设计的奖励函数需要考虑平衡多个目标，需在补货量决策的精度与推理解释的质量之间进行权衡，以确保模型既能做出准确的调度决策，又能提供可信的推理依据。首先，从补货决策数量角度看，补货数量的准确性直接决定了库存水平的合理性与供应链运作成本的高低；而从可解释性角度出发，多步推理决策过程的透明度与逻辑可追溯性则是管理者对模型结果产生信任的基础。二者看似密切相关，却往往难以在同一奖励函数中兼得：当以极高权重追求预测精度时，模型可能通过“黑箱”策略牺牲解释路径的合理性；而若将可解释性置于优先地位，则有可能因过度强调推理过程的完备性而导致补货数量偏离最优解。所以多目标奖励设计需确保库存决策精度的同时，增强决策的可解释性。

综上，本文将设计基于 LLM 的库存调度决策框架 InvLLM，其通过结合库存管理调度流程，实现在不确定需求下对产品单元进行多步推理补货决策过程。进一步地，本文将设计多目标平衡的强化学习方法以确保模型在优化性能的同时保持合理的解释性。本文的主要贡献如下：

1）设计基于库存调度流程的自校准多步推理方法，通过从库存调度流程文本中提取并整合关键信息，设计自校准的多步推理方法，实现对库存调度决策过程的精准推理，缓解了多步推理中误差累积导致的不稳定性问题。

2）设计基于多目标平衡的库存调度决策优化方法，通过对初始推理轨迹进行优化，设计基于多目标平衡的强化学习调度方法，确保库存决策精度的同时，增强决策的可解释性。

3）通过在中国电子商务公司“京东”提供的大规模真实数据集上的实验评估，证明了所提出的 InvLLM 相较于现有的库存调度方法在服务率和总体库存评价分数上分别提高了 3.24% 和 1.55%，并降低了 1.33% 的库存率，验证了该方法的有效性和实用性。

1 相关工作

近年来，已有大量研究致力于改进与库存管理补货相关的方法。其主要采用了基于深度学习模型的时间序列预测方法，用于预测未来商品的销售量，并将预测结果作为补货决策的基础。Wang 等人[14]提出 M-GAN-XGBOOST 混合模型，将长短期记忆网络（LSTM）[15]、生成式对抗网络（GAN）与决策树模型（XGBOOST）[16]有机融合，用于电商单品销售量的短期预测，并为精准营销与库存预警提供决策支持。Li 等人[17]针对服装销量问题，构建了复合 GRU–Prophet 模型。Chandriah 等人[18]针对汽车零部件需求场景，引入修改版 Adam 优化器对学习率自适应调整来用于循环神经网络，结果证明其提高了库存服务率，避免了缺货现象的产生。Joseph 等人[19]构建 CNN-BiLSTM 混合框架，并使用 Lazy-Adam 优化器，以零售门店商品需求挑战赛数据为载体，验证了卷积神经网络（CNN）[20]与时序依赖相结合的优势，并为库存补货决策设定目标。Liu 等人[21]针对不平衡的补货预测问题，提出随机欠采样和条件 Wasserstein 生成对抗网络的混合方法，其结果改进了对于不平衡的补货预测问题的性能。Shajalal 等人[22]针对补货预测领域的不平衡性，采用权重提升、随机过采样、合成少数类过采样技术以及相应组合，对深度全连接网络进行训练，实验证明其结果可以实现更优的预测性能并兼顾利润最大化。

尽管这些方法在预测精度方面有一定提升，但它们通常仅将未来的销售视为补货量，从而在需求波动或预测误差情况下产生过度库存或库存积压，缺乏对库存成本的全面考虑。

在过去的几年里，许多关于将强化学习(reinforcement learning, RL)技术整合到库存管理问题中的综合性研究得到了广泛关注。最早的研究可以追溯到 Giannoccaro 等人[23]的研究，其首创性地将供应链库存问题建模为多主体马尔可夫决策过程，采用仿真驱动的强化学习算法联合优化供应商、制造商与分销商的订货策略。该方法在大规模状态空间下仍能高效收敛，显著改善了整体供应链成本和服务率。随着研究的深入，学者们提出了不同的变体，特别是聚焦于具有 2 级供应链结构[24-25]、多级供应链结构[26-27]，以及多个零售商[28-30]的供应链系统。在所有这些研究中，强化学习方法的应用被用于制定供应链中接近最优的订购策略，目的是在满足不同目标的前提下优化库存管理，其目标包括最大化利润或最小化持有成本。在这些强化学习方法中，深度强化学习(DRL)被认为是一种通用技术，能够适应不同的应用场景，并且在仓库补货等复杂变化的随机过程中表现出强大的适应性。Boute 等人[31]系统性总结了 DRL 在库存控制中的主要算法设计选择，为研究者提供了全面的设计指南，但其未提出具体的库存调度决策方案。De 等人[32]针对易腐品库存调度的 DRL 训练不稳定与性能波动问题，引入基于启发式策略的潜在函数式奖励，并以深度 Q 网络(DQN)为例，验证了其方法在加速训练收敛、提高策略性能的效果。Kavuk 等人[33]将配送中心的库存调度决策建模为 DQN 问题，其结果在准时送达率和资源利用率上均优于企业现有的方法。Wang 等人[34]结合多智能体仿真与强化学习优化库存、重订货点和安全库存框架策略参数，在航空制造供应链案例中实现了动态补货策略，对多维绩效指标均有显著改善，且对需求/供应波动具有较强鲁棒性。Wang 等人[35]提出基于双重 DQN 的 DDLS 框架，改进状态空间表征与参数共享机制，实现了更智能的库存调度决策。Oroojlooyjadid 等人[36]将 DQN 引入供应链并建模为多智能体协作博弈，训练强化学习智能体在本地信息下近似全局最优订货策略，其结果显示所设计的方法可获得最优的库存成本。

虽然 DRL 能够学习到接近最优的策略，但其决策过程的可解释性较差，这对于需要透明决策的行业和领域来说其成为实施 DRL 的一个重大障碍。库存管理决策通常需要考虑到各种因素，如市场需求波动、库存水平等，而 DRL 在面对复杂的决策时，往往难以收敛进行求解，导致其在实际应用中有时未能达到预期效果。

随着库存管理场景下的补货决策日益变得动态和复杂，人工智能在提升决策能力方面的潜力引发了广泛的关注[37-38]。LLM 因其强大的自然语言理解与推理能力，逐渐被引入库存管理，旨在通过零样本学习或少样本微调，实现对历史规则文本和数据的高效解读与决策支持。Li 等人[39]针对供应链优化决策与利益相关方之间信任与可解释性不足的问题，提出了 OptiGuide 框架。该框架在保持最先进组合优化算法性能的同时，通过将优化结果与指令询问结合，让 LLM 以自然语言形式回答成本、物流方案等场景化问题，并且不将敏感数据暴露给外部模型。Song 等人[40]分析了 LLM 在供应链规划、客户服务、知识管理等各环节的应用现状与潜力。Jannelli 等人[41]设计多智能体代理网络，使不同公司或部门间的库存与交付时间协商可由 LLM 自动完成，并在库存管理案例中验证了框架的可行性。Li 等人[42]构建了一个由项目经理代理、销售预测代理和库存管理代理组成的多智能体框架，通过 LLM 之间的自主协同与自我纠错，实现对库存水平的精细化控制。其结果在提高库存周转率、降低运输与持有费用方面具有显著效果，并在零缺货率的前提下大幅削减总成本，为基于多 LLM 协作的库存调度决策优化提供新思路。

尽管一些工作从不同角度验证了 LLM 在库存补货决策中的应用价值，但它们更多聚焦于系统框架的构建与整体性能的提升，而对模型内部推理过程的可控性与准确性关注不足。库存补货决策本质上是一个典型的多步骤推理决策问题，要求模型不仅能整合历史规则和数据，不同情形下维持推理过程的连贯性与一致性。为此，需要在已有的 LLM 补货应用基础上，进一步研究如何确保模型在生成补货策略时，能够通过清晰、有序的链式思维实现精确推理。从而在不确定需求情境中既保证决策结果的可靠性，又提升决策过程的可解释性。

2 基于 LLM 的库存调度决策框架

图 1 展示了 InvLLM 的架构图，该方法结合了多源数据融合模块、库存调度步骤生成模块、自校准多步推理模块，帮助 LLM 在多步骤决策过程中提升

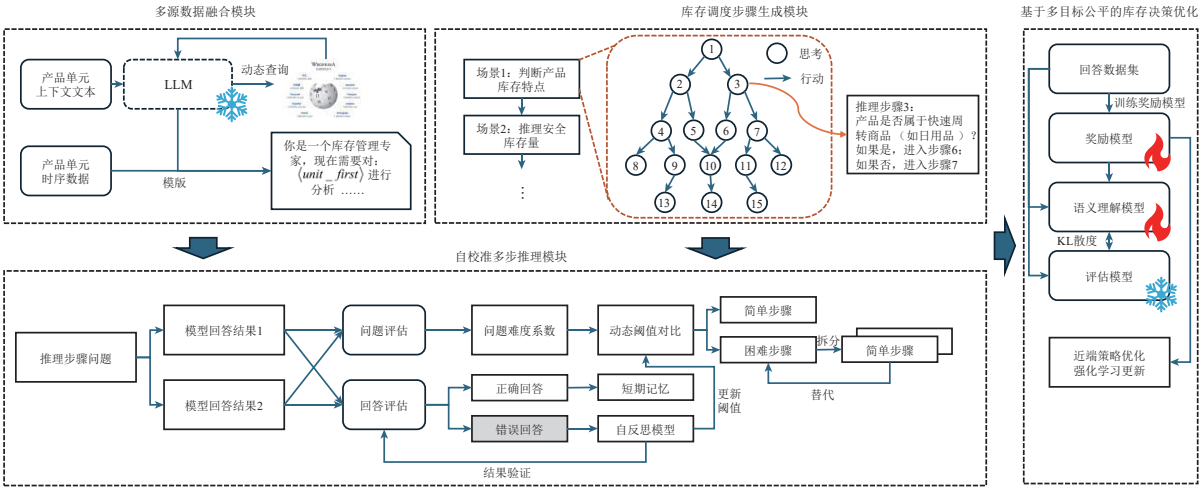


Fig. 1 Overall architecture of InvLLM

图 1 InvLLM 整体架构

可解释过程的准确性。

1)多源数据融合模块。该模块的主要功能是从维基百科的权威数据源实时查询最新的产品单元描述。通过对原始文本进行去噪和筛选，该模块能够提取出有效的关键信息，并最终生成简明且准确的事实信息摘要。这一过程确保了所获取的数据具有较高的可信度和相关性，为后续推理步骤提供了可靠的基础。

2)库存调度步骤生成模块。该模块负责根据现有的库存调度流程规则生成对应的问答流程图推理步骤。具体而言，该模块通过对复杂的库存调度决策过程进行系统化拆解，将整体决策过程分解为多个子场景。每个子场景对应着一系列特定的推理问题和相应的决策行动，以确保整个流程的逻辑严密性与执行可行性。模块首先根据预设的库存管理规则，对每一个子场景的决策进行明确划分。每个场景中的推理子步骤，不仅涵盖了需要解答的关键问题，还包括了在给定回答条件下应采取的具体行动决策。通过这一结构化的推理流程，模块实现了对库存调度决策的全面覆盖，确保每一项决策都能够依据实际情况和规则进行合理推导。

3)自校准多步推理模块。该模块负责为每一个补货决策步骤生成详细的解释。在此过程中，模块通过对比模型的回答，采用自校准机制对困难步骤进行分解和优化，确保推理过程的连贯性与准确性。通过这种方式，该模块不仅能够生成透明的决策解释，还能逐步改进推理策略，提升决策的可靠性。并且为了平衡推理过程中的难度，该模块通过自反思模型生成对应避免再次陷入错误的决策计划。自反思模型不仅帮助整体模型在推理过程中识别并避免重复错误，还能促进模型对问题的深刻理解，确保其能够准确地完成每一步骤的推理任务。

4)基于多目标平衡的库存决策优化。该模块旨在通过生成多目标平衡的奖励模型，并用强化学习方法联合优化微调 LLM，以进一步生成稳定、准确且具有可解释性的库存调度决策和解释。

3 多源数据融合

3.1 产品单元属性语义提取

鉴于维基百科能够提供更为详尽的产品属性背景知识，本节特选用该知识库作为主要外部信息源，以增强属性摘要的丰富性和准确性。

首先，大模型通过给定的产品单元u，构造对应产品单元的查询指令，其查询指令中包括该产品单元对应的 1 级、2 级、3 级属性文本，其中 3 级属性文本是和当前产品单元最相关的属性信息，而 1 级、2 级属性文本主要是构建上下文属性信息。本文根据构建好的通过维基百科数据库查询并返回与之最相关的文本信息C_u^k。该过程返回前k个与之最相关的结果。根据上一步内容中返回的前k个最相关非结构化产品单元属性文本C_u^k，提示大模型从产品单元属性文本中生成对应于库存管理流程的事实信息的摘要。该提示指令接受 2 个变量输入：指定的产品单元u和前k个最相关非结构化产品单元属性文本C_u^k。大语言模型M_{gen}(产品属性摘要的生成模型)生成一个可以从输入文本中学习的产品单元特征摘要X_u，X_u包含了产品单元的具体信息以及产品的其他相关

特性。部分查询到的产品单元特性可能包含更具体的信息,例如与巧克力相关的节日包括情人节、圣诞节、感恩节和复活节等。这一过程可以公式化为

$$X_u = M_{\text{gen}}(u, C_u^k)。\tag{1}$$

在生成的摘要中,产品属性对需求的影响可能在不同的时间节点上表现出显著差异。例如,在节假日或特殊促销期间,某些产品的需求量可能会显著增加。通过与外部维基百科数据库进行交互,模型能够从中提取出与产品相关的文本信息,并通过去噪处理生成与产品需求波动相关的事实摘要。这样,模型不仅能够捕捉到产品的静态属性,还能识别其可能蕴含的需求动态关系,进而为库存管理提供更为精准的决策支持。在本文中,返回前 $k(k=3)$ 个最相关非结构化产品单元属性文本。

3.2 产品单元异构数据语义融合

为解决如何让 LLM 感知如日常产品消耗需求、库存量等时序数据并捕捉不同时间段的变化问题,本文采用 Data2Text 模板生成技术,将时序数据转换为自然语言描述,将文本数据与时序序列相结合,从而使 LLM 能够处理并理解这些数据。Data2Text 模板生成技术通过定义具有可变占位符的文本模板,将时序数据与产品属性信息整合到一个统一的文本描述中。这些模板不仅包括固定的文本元素,还包含来自原始时序数据的具体信息,如每日消耗量、库存变化等。通过替换占位符,模板生成技术能够动态地为每个时间点生成描述性的文本 X_t,从而使得时序数据变得更加直观易懂。最后,将二者结合,生成具有时序性和环境信息的综合文本描述。形成一个全面的、包含时序性和环境信息的综合文本描述 $X_{(u,t)}$,如式(2)所示。

$$X_{(u,t)} = Data2Text(X_u, X_t)。\tag{2}$$

该文本描述不仅包括产品的静态属性,还包括与时间相关的动态变化信息,例如每日的需求量、库存量波动、节假日等相关信息。通过这种方式,时序数据与产品属性的信息能够被转化为易于理解的自然语言描述,使得后续的步骤中,LLM 能够更好地感知并分析库存需求的变化规律,为库存管理决策提供精准的支持。

4 库存调度步骤生成

库存调度步骤生成模块的主要目的是为库存调度决策生成清晰决策步骤。为了将复杂的库存管理决策过程拆分为易于理解和处理的部分,根据动态补货策略流程中的关键节点及逻辑转折点对整个流程进行划分,将整个库存调度流程划分为 4 个子场景,其分别为:判断产品库存特点、推理安全库存量、推理重订货点、决策最终补货量。其中,每个子场景代表一个较为独立的子任务,根据每个场景继续设计每个场景下具体的推理步骤,以便于完成该子场景下的推理任务,完整的推理流程如图 2 所示。

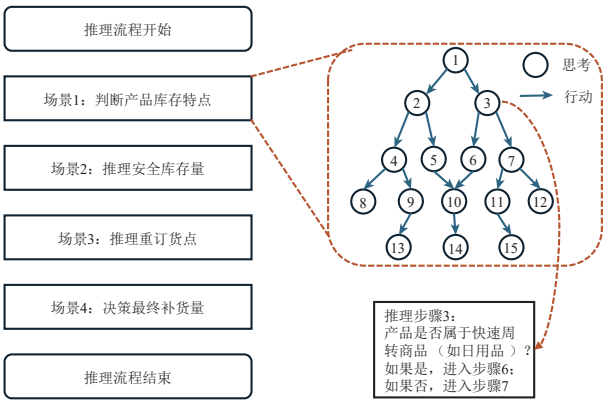


Fig. 2 The process of inventory scheduling

图 2 库存调度流程

4.1 库存调度流程子场景分割

首先将动态补货策略应用于本模块中,其中动态补货策略的通用公式为

$$\text{重订货点} = \text{平均日需求量} \times \text{补货间隔} + \text{动态安全库存}。\tag{3}$$

根据式(3),本文需要求解出对应的动态安全库存量和重订货点,并且依据重订货点确定最终的补货量。但在本文中,由于产品属性信息和时间、需求趋势紧密相关,所以需要额外判断产品的属性信息对于库存调度决策的影响,所以本文需要考虑产品单元属性特性、安全库存计算、重订货点计算、最终的补货量计算 4 个方面。所以依据这些方面,本节将整个库存调度流程切分为以下 4 个子场景:

1)推理产品单元的产品特性。在本子场景中,首先通过对产品单元的属性进行识别,判断产品是否属于快速周转商品、长尾商品,或者是否具有明显的季节性特征。此外,本文还需要考虑商品的需求波动性和高峰期情况。通过这些判断,可以为后续的库存管理决策提供依据。比如选择合适的服务率,对于需要额外补充的产品单元采用高服务率进行补货,从而为安全库存的计算提供基础。这一判断过程不仅有助于明确库存策略的定位,还能够确保库存能够满足不同市场条件下的需求变化。

2)确定安全库存量。本子场景的核心任务是根据历史需求数据，尤其是过去7天的平均需求量和需求的标准差，计算安全库存量。为了确保在需求波动和市场不确定性情况下库存能够保持充足，本节根据第1个子场景中的判断结果，选择合适的安全系数。此外，还需考虑市场因素，如节假日效应、促销活动等，对安全库存进行微调，从而提高库存策略的灵活性和响应能力。这一过程保证了在面对需求高峰时，库存可以及时补充，避免出现缺货或过剩的风险。

3)推理重订货点数量。在这一子场景中，本节通过计算平均交货周期需求，并结合已计算出的安全库存量，推理出重订货点的数量。重订货点的计算不仅仅是基于当前库存水平，还需要考虑产品单元的需求趋势和交货周期的变化。在此过程中，需动态监控当前库存与重订货点之间的关系，确保补货策略能够应对突发的市场需求变化，防止库存过剩或缺货的发生。每一次重订货点的推理过程需要被详细记录，为后续决策提供可靠依据，确保补货决策的透明性和可追溯性。

4)决策补货量。本阶段的任务是通过计算当前库存与重订货点的差值，最终确定补货量。通过将当前库存水平与重订货点进行比较，能够判断是否需要补货，以及需要补充的货量大小。该过程考虑了市场需求、库存特性和补货周期等因素，确保补货量既能满足需求，又避免了库存过剩。补货量的决策输出最终为库存调度系统提供具体的补货建议，指导实际的库存调整。

通过将库存调度流程划分为上述4个子场景，本节确保了每个环节都能够独立且精确地反映库存管理中的不同决策因素。通过逐步推理和精确计算，库存管理决策不仅能够有效应对动态需求波动，还能确保库存资源的合理配置，提升整体供应链的响应效率和稳定性。

根据前文切分的具体子场景，本节针对每一个子场景进一步提出了更为细粒度的推理流程，以提升库存调度决策的可解释性与准确性。具体而言，首先系统性地收集并整理了当前主流文献中与各子场景相关的库存调度流程规则[43-45]，确保推理基础的全面性与规范性。在此基础上，引入LLM对这些规则进行自动解析与建模，从而生成相应的规则流程图。随后，借助人工核验的方式对流程图的逻辑完整性与操作可行性进行审查与修正，确保其在实际业务中的应用价值。最终，得到每一个子场景包含多个需要进行推理的推理步骤内容，以及根据该推理步骤可以采取的下一步行动。

通过上述方法，在4个代表性子场景中系统性地收集并整理了对应的库存调度流程规则，涵盖从产品属性识别到补货决策的关键环节。在利用LLM生成各子场景规则流程图的基础上，进一步引入人工筛查机制，对流程图中可能存在的逻辑冲突或语义重复进行逐一审校与剔除，确保每一个子场景下的规则结构具备逻辑一致性与实际可行性，并且保证4个子场景相互衔接，从产品特性判断到安全库存计算，再到重订货点设定和最终补货量决策，依据动态补货策略构成了一套完整的库存管理与补货决策流程。

4.2 自校准多步推理

自校准多步推理模块的主要目的是为库存调度决策生成补货数量决策以及对应的解释。由于在多步推理的过程中，下一步的推理需要根据上一步推理得出的结果上下文进行分析，这就有可能导致随着推理的不断深入，提示指令会变得越来越长。为了缓解在多步推理过程中导致的上下文长度过长，本节设置了长短期记忆机制。对于每一次推理过程，本节将场景中的每一个步骤的推理结果上下文信息存储在短期记忆中，用以在场景中保证推理的进行。如果一个场景的流程结束，则将该场景的所有推理步骤保存到长期记忆中。

各个子场景间的短期记忆相互独立，一旦推理进行到下一个子场景，则将该场景的短期记忆存储的推理结果总结为场景要点传递到下一个子场景的开始，作为上下文信息保证在下一个场景中模型可以继续推理。根据库存调度步骤生成模块构建出的子场景，对于每个决策日D_i，每个子场景S_i中的每个推理步骤t_i，提示大语言模型M_{res}（用于生成响应文本）对每个步骤t_i生成相应的响应a_i。每个步骤的提示指令格式如图3所示。

该提示X包含：上一个模块生成的综合文本描述$X_{(u,t)}$、该场景下的短期记忆m_i、上一个场景的总结C_i、该场景下需要解决的推理步骤t_i和由自反思模型生成的推理回复R_i。给定这些输入，大语言模型M_{res}生成响应$a_{(S_i,t_i)}$，该响应包含该步骤下基于推理的行动和这一步决策的解释。本文将其形式化为式(4)。

$$a_{(S_i,t_i)} = M_{\text{res}}(X_{(u,t)}, m_i, t_i, C_i, R_i)。 \tag{4}$$

然而，针对当前模型的响应结果，亟需一种评价机制来判定输出的正确性；一旦模型产生错误响应，

输入

⟨Context⟩
你是一个库存管理专家，现在需要对：⟨*unit_first*⟩大类下的⟨*unit_second*⟩类下的⟨*unit_third*⟩产品进行补货决策，根据其库存性质，最相关的...
⟨/Context⟩

⟨Memory⟩
当前场景推理记忆如下：⟨*short_term_memory*⟩
历史场景总结如下：⟨*scenario_summaries*⟩
⟨/Memory⟩

⟨Instruction⟩
你当前处于场景⟨*current_scenario_information*⟩中，请严格按以下步骤执行：
① 分析场景中的问题：⟨*question_text*⟩
② 若涉及计算必须分步展示公式和数值结果
③ 必须从以下选项中选择下一步：⟨*valid_next_step*⟩
补充信息如下：⟨*additional_requirement*⟩
⟨/ Instruction⟩

⟨ Outputformat⟩
必须使用以下格式：
next_step: ⟨数字⟩
解释：
① 关键参数提取(如$\mu = XX$，$\sigma = XX$)
② 分步计算过程(如 $SS = 1.65 \times 242.6 \times sqrt(1) = 400$)
③ 结论陈述
⟨/ Outputformat⟩

输出

好的，我现在正处于场景...，我现在需要判断...

Fig. 3 Prompt instruction format in each inference step

图 3 各推理步骤的提示指令格式

误差便会在后续步骤中逐步累积。通过对这些错误响应的分析，其主要有以下 2 个方面原因：

1)推理步骤过于复杂。由于库存调度步骤生成模块所生成的推理步骤并没有考虑模型的实际表现，所以可能生成的推理步骤对于模型而言过于复杂，导致模型很难输出正确的响应。

2)模型响应失误。如果问题并不复杂，模型仍有一定概率判断失误导致响应错误。

为了解决这一问题，本节在原有的多步推理方法中引入自校准机制，如图 4 所示。

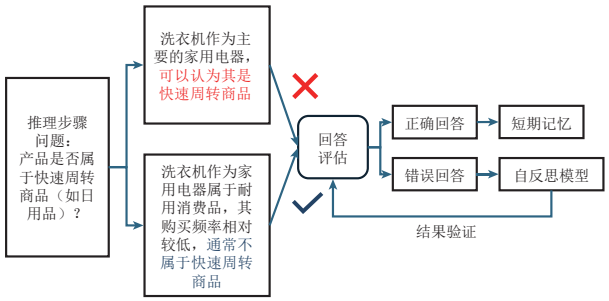


Fig. 4 Self-calibration framework of inference results

图 4 推理结果自校准框架

自校准机制从 2 个方面校准模型的响应：

1)通过计算推理步骤的难度系数，根据阈值区分该问题的难易程度，将困难步骤进一步自动拆分为简单步骤；

2)根据模型的回答，通过调整响应的推理步骤，对每一步生成的回答进行准确性校验。

为了解决推理步骤过于复杂的问题，本文通过提示 ChatGPT-4 同步生成该步骤下的响应，并且本文也将其作为 2 份响应的评估工具，它基于响应的相关性、准确性以及格式符合度进行综合评分。

为了避免偏差，回答为 2 次评估的平均值，并且 2 次评估调换 2 个模型的响应顺序以消除位置造成的影响。

获得每个步骤的分数后，本文使用式(5)来计算难度。

$$Q = (|p_1 - p_2| + 1)/\max(p_1 - p_2), \tag{5}$$

其中 p_1 和 p_2 代表 2 个响应的分数，该过程首先计算基座模型和评估模型分数差的绝对值，然后将其归一化到 0~1 的范围内，使分数差对不同分数水平的影响保持一致，最后得到可以反映问题难度的难度系数 Q。Q 分数越高，反应该步骤就越困难。基于此，本节设置了 1 个动态阈值，将步骤分为 2 类：Q 大于阈值时表示该步骤较为困难，反之，则可认为步骤较为容易。对于困难步骤，需将其进一步拆解为 2 个子步骤以保证模型的回答质量。

关于动态阈值的设定，本文需要结合模型回答的情况来分析，这也引出了另一个需要解决的问题，即如何判断模型响应是否正确，并在未来的推理中避免相同的错误。本文设计了“自反思”模型，如图 5 所示。其首先根据 GPT-4 对模型的回答进行判断，对于正确回答，将模型的回答保存在短期记忆中；对于错误的回答，进一步采用自反思模型生成相应的指导意见，并将这些指导意见作为额外的上下文信息传递给下一个推理步骤。

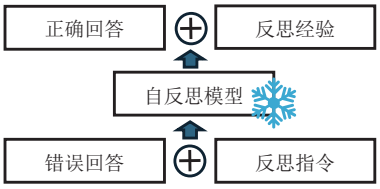


Fig. 5 The workflow of self-rethinking model

图 5 自反思模型工作流程

该方法旨在帮助 LLM 从过去的失败中进行学习，以提高其在未来任务中的表现。通过这种方式，LLM 能够在下一次执行推理步骤时参考之前的反思，改进决策过程。这种机制类似于人类在多次尝试后，通过总结先前失败的经验来不断优化自己的行动策略，从而在下一次尝试中表现得更加出色。其优势在于它提供了一种非常灵活和高效的学习方法，不需要对 LLM 进行复杂的微调，并且自反思模型允许更加细致和针对性的反馈。例如，它能够为 LLM 提供具体的行动提示，明确指出需要改进的地方，从而

帮助LLM在未来的推理步骤中做出更为准确的决策。此外，这种方法还为智能体提供了更加明确和可解释的情境记忆形式，使得智能体能够记住过去的经验，并在新的情境中加以应用，从而提高决策的稳定性和一致性。

对于不正确的响应，本文提示LLM为每次迭代生成言语反馈，反馈应该清楚地解释它在之前的推理中出错的地方，并为下一次迭代提出一个计划来缓解这种错误。

对于每一次迭代、每一次反思，LLM从失败中吸取的教训，这些教训被添加到其经验中。本文将其表示为一组反思。反思与原始输入一起再次输入响应生成大语言模型M_{res}，以生成下一次迭代的库存调度建议和解释。并且根据每个步骤的迭代次数，可以设置阈值来区分每个步骤的难易程度。阈值的计算为：

$$\tau = \frac{Q_{\max} - Q_{\min}}{1 + \mathrm{e}^{-a[(1-Q)-d_0]}} + Q_{\min} - d \times penalty(Q)。 \quad (6)$$

$penalty(Q)$为动态计算的惩罚项，其公式为

$$penalty(Q) = base + scale \times \frac{Q - Q_{\min}}{Q_{\max} - Q_{\min}}, \quad (7)$$

其中$Q_{\max}$和$Q_{\min}$分别代表所有问题难度系数的最大值和最小值，Q为难度系数，d_0和a为固定值，本文将a设置为10，d_0设置为0.3，d代表重试次数，惩罚项中$base$和$scale$为预先设定好的固定值，根据模型表现，本文将$base$设置为0.1，$scale$设置为0.2。式(6)(7)可以根据每个问题的难易程度结合重试次数自动设定动态阈值，可以更细粒度地判别推理步骤的难易程度，既保证了推理结果的准确和稳定性，又有效提升了整体推理效率和响应速度。

在生成最终的响应后，需要进一步将其保存至场景的短期记忆中，短期记忆中存储该场景下的所有推理轨迹。一旦推理进行到下一个场景时，需要进一步将上一个场景中的短期记忆总结为摘要，为下一个场景的推理提供推理上下文信息。

最后，还需要将所有场景中的短期记忆进一步存储在长期记忆中，以保存产品单元u的完整推理过程。

最终本文得到了模型经过库存调度流程推理后的推理总过程。对于同一个X，模型生成多个推理过程，并选取其中总反思次数最少的作为实际补货决策方案并以此更新库存，其余的进行保留，用作后续结果调优的数据集。本节所设计的基于库存调度流程的自校准多步推理方法缓解了在复杂场景下，LLM在推理过程中的误差累积，并且动态调整推理步骤并纠正错误，通过对决策过程的反馈进行优化，不断增强模型的决策能力与稳定性。

5 基于多目标平衡的库存决策优化方法

为了在基于库存调度流程的自校准多步推理方法中生成的补货调度推理轨迹的基础上实现对最终的补货结果进一步的优化，本节提出了一种基于多目标平衡的库存决策优化方法，旨在通过生成多目标平衡的奖励模型，用强化学习方法联合优化微调LLM，以进一步生成稳定、准确且具有可解释性的库存调度决策和解释。首先，本节介绍了方法的4个关键阶段：基于库存决策结果的监督微调、多目标平衡奖励模型设计及训练和近端策略优化(PPO)方法。通过这4个阶段的优化，模型能够有效提升决策精度，并在多目标环境下平衡不同的决策目标。

5.1 基于库存决策结果的监督微调

在监督微调阶段，使用补货决策轨迹集作为微调数据集，对预训练LLM进行有监督的参数更新。具体而言，本阶段的核心目标是“对齐”模型生成的多步推理补货决策轨迹，通过最小化生成分布与真实样例分布之间的差异，使模型能够在给定库存状态和需求信息时，输出高质量且连贯的决策文本。具体来说，首先将每个输入提示X与其对应的多步推理轨迹决策文本L拼接为模型的训练样本，其中对应多步推理决策序列，然后采用标准的交叉熵损失对模型输出进行约束。

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \ln P_\theta(L|X) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{t_i} \ln P_\theta(l_{i,t}|X, l_{i,1:t-1}), \quad (8)$$

其中$P_\theta(l_{i,t}|X, l_{i,1:t-1})$代表模型在给定提示$X$及前$t-1$个已生成词元的条件下，对第$t$个词元的预测概率，$\theta$为模型参数。通过该阶段的微调，模型不仅在语义和格式层面对回答文本进行了有效校准，而且在决策上实现了与多步推理轨迹文本高度对齐，形成了初始策略，为后续的奖励模型训练提供了可靠的参考分布，并为近端策略优化阶段奠定了坚实的策略参数基础。

5.2 多目标平衡奖励模型设计

对于奖励模型架构，本文使用预训练LLM，删除最后一个softmax层，并在最后添加1个额外的线性层。给定任何文本，奖励模型将为最后1个词元分配标量奖励值，奖励值越大，样本越好。

首先基于库存决策结果的偏好数据集构建,收集每个决策日期 *date* 中的每个产品单元的 X_{date} 对应的 k 个总推理过程 L_{date} 作为初始数据集 D。将 k 个回答中的每个回答分为推理部分 R_{date} 和补货数量部分 $Q_{u,date}$。对推理部分 R_{date},采用 GPT-4 从格式、一致性和连贯性 3 个维度进行打分,记为 $s_{u,\text{reason}}$ 并归一化至 [0, 1];对补货数量部分 $Q_{u,date}$,基于总体库存评价分数计算得分 $s_{u,\text{qty}}$,其结果范围为 [0, 1]。随后,将这 2 部分得分相加,得到综合得分:

$$s_u = s_{u,\text{reason}} + s_{u,\text{qty}} \text{。} \tag{9}$$

根据 s_u 对 k 条轨迹进行降序排序,选取得分最高的轨迹作为首选响应 $\tilde{y}$,得分最低的轨迹作为次选响应 $\hat{y}$。由此构成的偏好示例 $(X,\tilde{y},\hat{y})$ 共同组成了偏好数据集。这样的构建方式不仅侧重于提升模型在给定库存状态和需求信息下的决策精度,还兼顾了决策过程的可解释性。该偏好数据集为后续的奖励模型训练既提供了高质量决策示例,又保留了可解释性文本,有助于在保证库存补货决策精度的同时,增强模型生成的决策结果的可解释性。

5.3 多目标平衡奖励模型训练

在完成基于库存决策结果的偏好数据集构建后,需要利用该数据集对奖励模型进行有针对性的训练。通过偏好示例中首选响应与次选响应的对比结构,奖励模型将学会在给定相同输入提示 X_{D_i} 时,为不同候选输出分配合理的标量奖励,从而准确刻画不同推理轨迹的差异。对于输入 X_{D_i},首选和非首选样本的建模损失为

$$L(\psi) = \ln\sigma(r(X,\tilde{y}) - r(X,\hat{y}))\text{,} \tag{10}$$

σ 是 sigmoid 函数。r 表示具有参数 ψ 的奖励模型,$r(X,y)$ 是输入提示 X 和响应 y 的单个标量预测奖励,$\tilde{y}$ 为首选响应,$\hat{y}$ 为次选响应。

另一方面,在训练好奖励模型后,奖励函数不仅仅依赖单一的奖励模型输出的奖励值,本文在奖励函数中加入了 1 个额外的项,该项基于学习到的强化学习策略和初始监督模型之间的 KL(Kullback–Leibler)散度引入了惩罚。KL 散度项的引入主要是为了防止在训练过程中出现过度拟合现象,确保学习策略的多样性和鲁棒性。通过调整模型策略的分布,使其尽可能地接近初始监督模型的分布,这样既能够保持模型的预测稳定性,又能保证其在面对不同的环境时能够做出灵活的调整。

5.4 基于近端策略优化的强化学习方法

由于库存调度决策问题中涉及的状态-动作空间较为庞大,直接应用强化学习可能会面临诸多挑战。在本方法下,本节将库存调度流程的动态决策过程视为“环境”,并将 LLM 作为“智能体”。对于每个决策日 *date*,智能体从环境中接收一个状态,该状态由当前的库存信息、需求数据等构成。在该状态下,智能体根据其策略生成动作,即包含推理过程和补货数量的总补货决策过程。环境根据生成的动作返回一个奖励,该奖励根据多目标平衡奖励模型中所设计的奖励函数计算,其考虑到推理过程如一致性、连续性等特性的合理性和补货数量的精准性。在根据采取的补货决策动作后,智能体根据补货数量更新当前库存状态,并计算新的库存水平、需求情况及其他相关因素,进而转移到下一个状态,并重复进行该过程。

6 实验结果与分析

本节利用来自中国电子商务企业的京东公司的实验数据集对提出的模型进行了评估。该数据集分为 2 个部分:1)京东商城下单的订单需求数据,时间范围为 2022 年 1 月—2023 年 12 月共 2 年数据,涵盖 24 种品类的在京东商城下单的订单数量。2)京东仓库出库的订单需求数据,时间范围为 2023 年 12 月—2024 年 12 月共 1 年,涵盖 70 种品类的京东仓库的出库订单数量。对于 InvLLM 模型,本节验证了该模型中各算法构建的有效性。

6.1 数据集

实验数据集来自中国电子商务企业之一的京东公司。该数据集分为 2 个部分:1)京东商城下单的订单需求数据(简称下单订单数据集)。2)京东仓库出库的订单需求数据(简称出库订单数据集)。在数据划分上,本文将选择时间维度上前 80% 的历史订单数据作为训练集,其余的 20% 作为测试集。具体而言,针对商城下单的订单需求数据,本文使用 2022 年 1 月—2023 年 9 月的历史数据来训练 InvLLM 模型,训练完成后,本文使用该模型在 2023 年 10 月—12 月的历史数据预测未来日期的补货数量;对于出库订单需求数据,本文使用 2023 年 1 月—2024 年 9 月的历史数据来训练 InvLLM 模型,随后在 2024 年 10—12 月的历史数据中进行补货决策。对于初始库存量,本文采用基于需求数量的初始库存量设定方法。在实验中,对于不同品类的产品单元,本文计算其对应数据集的最初始 7 天的需求总数量,并将此需求总数量的 1.2 倍作为该产品单元的初始库存量,

旨在为模型提供一个合理的起始点，以模拟实际库存管理中的初始状态。每周的决策日定为周六，每次补货请求下达后，该批数量补货对应的到达时间为决策日的第 2 天，并根据实际需求消耗和到达的补货数量进一步更新库存。

6.2 验证方式与相关设定

为全面评估模型的性能，本文将计算模型决策的补货数量对应的总体指标，并通过这些指标对模型的实际效果进行评估。此外，本文还设计了对比实验和消融实验，通过这些实验进一步验证所提模型的优越性和适用性。所有实验均在 Python 3.9, cuda 12.0 环境下的 Pytorch 2.0.0 框架中完成，实验平台配置为 128 GB 内存和 2 块 32 GB 显存的 NVIDIA V100 GPU。为了确保模型的训练效果，本文采用了 Qwen2.5-7b 模型，并且对于同一输入产生 3 个推理结果。应用 AdamW 优化器进行训练，学习率设置为 5E−5，最大梯度范数为 1，批处理大小设定为 2。针对微调过程，本文采用了 LoRA 高效微调方法，LoRA 秩设置为 8，LoRA 缩放系数设置为 32，学习率倍数设置为 32，以提高训练效率并降低计算资源消耗。

6.3 评估指标

在本文中，本文借鉴了 3 类常用且具代表性的评价指标：服务率（service level, sla_i）[46]、库存率（inventory rate, inr_i）[47] 以及总体库存评价分数（overall inventory evaluation score, $OIES$）[48]。并基于实际数据对各指标公式进行了适当调整与校准，以更贴切地反映库存调度决策的好坏。具体而言：

服务率 sla_i 代表决策补货是否导致缺货，计算公式为

$$sla_i = 1 - \frac{\sum_{d\in D} stockout_{i,d}}{MSS(DemandSequence_i)}, \tag{11}$$

其中 D 为所需评估日期的集合，MSS 为最大子序列和，$stockout_{i,d}$ 为产品单元 i 在日期 d 当天的未被满足的需求量，$DemandSeuqence_i$ 为产品单元 i 的需求净增量时间序列。

库存率 inr_i 代表每次补货剩余库存量的成本，计算公式为

$$inr_i = \frac{\sum_{d\in D} qty_{inventory(i,d)}}{(qty_{inventory(i,d)} + qty_{using(i,d)}) \times T}, \tag{12}$$

其中 D 为所需评估日期的集合，$qty_{using(i,d)}$ 为产品单元 i 在日期 d 当天的资源使用量，$qty_{inventory(i,d)}$ 为产品单元 i 在日期 d 当天的资源库存量，T 为评估天数。

总体库存评价分数定义为

$$OIES = \frac{1}{2n}\sum_{i\in I}\left[f(sla_i) + (1 - inr_i)\right], \tag{13}$$

其中 I 为所有产品单元集合，n 为产品单元的数量，$f(\cdot)$ 用于将原始服务率映射至 (0, 1) 区间，以增强对中间水平的敏感度。

对于服务率 sla_i 而言，其值越大越好，最大值为 1；对于库存率 inr_i 而言，其值越小越好；对于总体库存评价分数 $OIES$ 而言，其值越大越好。上述 3 类指标在建模上既兼顾了决策服务保障与库存成本控制的平衡，又通过映射与归一化手段提升了数值稳定性与可比性，并且在达到一定服务率的情况下，使得库存率的影响更加显著。

7 实验评估与分析

7.1 比较方法

本文将 InvLLM 与 3 种类型的方法进行比较，包括基于预测模型的补货方法、基于 DRL 的补货方法、基于 LLM 的补货方法。1）基于预测模型的补货方法有 ARIMA, LSTM[49], Transformer[50], CNN-BiLSTM[23]。2）基于 DRL 的补货方法有 DDPG[51], SAC[52], TD3[53], DDLS[37]。3）基于 LLM 的补货方法有 Chatglm3-6B[54], Vicuna-7B-v1.5[55], Gemini-1.5[56], Claude-3.5, GPT-4[57]。

7.2 总体性能表现

本文将所提出的方法与所有基线方法进行了比较，并在表 1 和表 2 中展示了不同方法的性能比较。总体而言，本文方法在电子商务企业的真实数据集上的总体评价分数优于现有的所有基线方法。在下单订单数据集中，相较于现有的性能最先进的方法 DDLS, InvLLM 在服务率和总体库存评价分数上分别提高了 3.24% 和 1.55%，并降低了 1.33% 的库存率；在出库订单数据集中，InvLLM 在服务率和总体库存评价分数上分别提高了 3.82% 和 0.82%，并降低了 0.11% 的库存率。这一结果表明，InvLLM 不仅能够在库存管理中实现较高的服务率，而且在保持低库存率的同时，提升了整体决策效果。

具体而言，InvLLM 在服务率和总体评价分数上的表现都明显优于传统的基于预测模型和基于 DRL 的方法，尤其是在应对复杂的库存需求波动和补货决策时，能够更好地平衡服务率和库存率。

1）基于预测的补货方法的比较

基于预测的补货方法包括 ARIMA, LSTM, Transformer, CNN-BiLSTM 模型，它们通过预测库存需求

Table 1 Performance Comparison of Different Methods on Order Datasets

表 1 下单订单数据集不同方法的性能比较

补货方法	模型	服务率	库存率	总体分数
基于预测模型	ARIMA	0.810 5	0.791 9	0.582 6
	LSTM	0.827 8	0.795 7	0.584 0
	Transformer	0.836 7	0.782 9	0.591 9
	CNN-BiLSTM	0.833 8	0.760 6	0.602 6
基于 DRL	DDPG	0.839 5	0.781 8	0.592 9
	SAC	0.821 5	0.754 6	0.603 4
	TD3	0.842 1	0.775 8	0.596 3
	DDLS	0.816 5	0.741 2	0.609 1
基于 LLM	Chatglm3-6B	0.841 2	0.793 9	0.587 1
	Vicuna-7B-v1.5	0.838 8	0.796 3	0.585 5
	Gemini-1.5	0.814 1	0.752 6	0.603 0
	Claude-3.5	0.815 6	0.770 7	0.594 2
	GPT-4	0.809 0	0.742 5	0.607 0
	InvLLM（本文）	**0.843 8**	**0.731 5**	**0.618 7**

注：黑体数值表示最优值，下划线数值表示次优值。

Table 2 Performance Comparison of Different Methods on Outbound Order Datasets

表 2 出库订单数据集不同方法的性能比较

补货方法	模型	服务率	库存率	总体分数
基于预测模型	ARIMA	0.829 8	0.767 2	0.598 6
	LSTM	0.838 6	0.768 9	0.599 2
	Transformer	0.824 9	0.756 8	0.602 9
	CNN-BiLSTM	0.806 8	**0.726 7**	0.614 4
基于 DRL	DDPG	0.817 6	0.746 3	0.606 8
	SAC	0.832 8	0.729 4	0.618 0
	TD3	0.819 7	0.739 9	0.610 4
	DDLS	0.853 0	0.737 5	0.617 0
基于 LLM	Chatglm3-6B	0.849 1	0.762 5	0.604 0
	Vicuna-7B-v1.5	0.813 9	0.758 9	0.599 8
	Gemini-1.5	0.804 3	0.730 4	0.612 0
	Claude-3.5	0.818 7	0.742 9	0.608 7
	GPT-4	0.848 4	0.746 8	0.611 7
	InvLLM（本文）	**0.865 9**	0.728 6	**0.623 1**

注：黑体数值表示最优值，下划线数值表示次优值。

来决定补货策略。表 1 和表 2 中的实验结果显示，在下单订单数据集上，Transformer 相较于 ARIMA 和 LSTM 在服务率上具有明显优势，而 CNN-BiLSTM 则以 0.760 6 的最低库存率和 0.602 6 的最高总体分数优于其他预测模型。虽然其服务率(0.833 8)略低于 Transformer(0.836 7)，但更优的库存控制使其在整体表现上更好；在出库订单数据集上，Transformer 在总体分数方面领先于 ARIMA 和 LSTM，但 CNN-BiLSTM 依旧以 0.614 4 的总体评价分数优于其他预测模型。该模型不仅实现了四者最低的库存率(0.726 7)，也维持了相对稳定的服务率(0.806 8)，表明在兼顾库存成本与客户满意度方面具有较好平衡。综上所述，尽管基于预测的补货方法能够较好地捕捉库存需求波动，尤其是 CNN-BiLSTM 模型在 2 个数据集上的表现都较为优秀，但这些方法在建模时缺乏对库存周期、补货提前期、安全库存等库存调度流程规范的显式建模，导致预测结果直接用于补货数量计算时，无法动态调整策略以应对库存约束，难以从端到端的角度对“是否应该补、补多少”进行决策优化。虽然 CNN-BiLSTM 模型获得了较高的总体分数，但依然未能超越本文的 InvLLM 模型。

2)基于 DRL 的补货方法的比较

基于 DRL 的补货方法包括 DDPG, SAC, TD3, DDLS，旨在通过学习和优化库存调度策略，最大化长期回报。在下单订单数据集上，TD3 模型取得了与其他基线相比次优的服务率(0.842 1)，而 DDLS 模型则取得了次优的库存率(0.741 2)和总体分数(0.609 1)；在出库订单数据集上，SAC 取得了次优的总体分数(0.618 0)，而 DDLS 取得了次优的服务率(0.853 0)。综上所述，基于 DRL 的补货方法 DDPG, SAC, TD3, DDLS 在 2 个数据集上均实现了对传统基于预测模型方法的超越。这表明强化学习通过与环境交互不断优化策略，能够更好地适应需求波动并最大化长期回报，并能够在动态环境中不断调整策略，具有较强的适应性。但在 2 个数据集上，其服务率和总体分数均低于 InvLLM 模型，主要得益于 InvLLM 利用 LLM 在多步推理与自校准机制的支持下，能够生成更符合实际业务场景的补货决策。

3)基于 LLM 的补货方法的比较

基于 LLM 的补货方法，包括 Chatglm3-6B, Vicuna-7B-v1.5 等 LLM，是近年来在自然语言处理和生成领域取得显著进展的一类方法。这些模型通过大规模的预训练数据和强大的生成能力，能够进行复杂的推理和决策，从而为库存管理提供新的解决思路。在下单订单数据集上，虽然 Chatglm3-6B 和 Vicuna-7B-v1.5 取得了较高的服务率，但是其库存率也相对较高，导致其总体分数表现一般。而 Gemini-1.5, Claude-3.5, GPT-4 可以获得比 Chatglm3-6B 和 Vicuna-7B-v1.5 更优的总体分数，其中 GPT-4 的总体分数相较于其他 4 个模型最好(0.607 0)；在出库订单数据集上，

Gemini-1.5 与 Claude-3.5 二者在库存率上有所改进，显示出更好的平衡能力，GPT-4 则不仅实现了最高的服务率（0.848 4），还保持了较低的库存率（0.746 8）。

综上所述，基于 LLM 的补货方法虽然在文本理解和推理流程生成方面具有天然优势，但由于未对关键库存调度参数（如补货提前期、最小补货量、安全库存水平等）进行显式建模，其难以兼顾服务水平与库存成本。结果显示，GPT-4 在下单订单数据集上总体分数为 0.607 0，出库订单数据集为 0.611 7，均落后于专门设计的 InvLLM；而 InvLLM 则凭借规划词元、偏好训练及多目标平衡算法，在下单订单数据集取得 0.618 7（对比 GPT-4 领先 1.17 个百分点）、在出库订单数据集取得 0.623 1（领先 1.14 个百分点）的最佳表现，证明了将可解释推理与多目标强化学习深度融合，能够进一步提升大模型在库存补货策略上的决策质量和业务适用性。

7.3 消融实验

为了验证 InvLLM 模型中各模块的有效性，本节对多源数据融合模块、自校准多步推理模块、强化学习模块分别设计了消融实验来评估所设计模块的重要性。本文构建了以下 3 种模型变体：

1）InvLLM 不考虑多源数据融合模块（InvLLM w/o data to text module, w/o DT）。当不考虑外部信息文本摘要模块时，本文将删去从外部环境中获取的信息文本来验证多源数据融合模块对 InvLLM 的影响。

2）InvLLM 不考虑自校准多步推理模块（InvLLM w/o self calibration multistep reasoning, w/o SCMR）。本文跳过自校准的多步推理方法，使用原始的 prompt 引导模型得到补货数量，并根据模型的输出结果进一步设置强化学习的训练数据集，以此验证自校准多步推理方法对 InvLLM 模型的影响。

3）InvLLM 不考虑强化学习模块（InvLLM w/o reinforcement learning, w/o RL）。当不考虑强化学习进一步优化模型的推理可解释性及决策精度时，InvLLM 将退化成基于自校准多步推理方法的模型。其对应的消融实验结果如表 3、表 4 所示。

Table 3 Ablation Study Results on Order Datasets

表 3 下单订单数据集上的消融实验结果

模型	服务率	库存率	总体分数
w/o DT	0.830 4	<u>0.742 8</u>	<u>0.610 9</u>
w/o SCMR	<u>0.833 1</u>	0.780 7	0.592 4
w/o RL	0.829 5	0.755 0	0.604 6
InvLLM	**0.843 8**	**0.731 5**	**0.618 7**

注：黑体数值表示最优值，下划线数值表示次优值。

Table 4 Ablation Study Results on Outbound Order Datasets

表 4 出库订单数据集上的消融实验结果

模型	服务率	库存率	总体分数
w/o DT	0.854 8	<u>0.732 5</u>	<u>0.619 8</u>
w/o SCMR	<u>0.864 3</u>	0.759 0	0.607 7
w/o RL	0.825 6	0.736 5	0.613 2
InvLLM	**0.865 9**	**0.728 6**	**0.623 1**

注：黑体数值表示最优值，下划线数值表示次优值。

在去除多源数据融合模块（w/o DT）的设置中，模型在下单订单数据集和出库订单需求数据集上的库存率均为次优，虽然未能满足一些服务率的需求，但其总体分数为次优。由此可见，由于模型自身在预训练过程中已经存在一些简单的对于产品单元属性的感知，导致外部摘要信息并非提升服务率的关键，但它对于抑制库存率增长和维护更高的综合决策质量发挥了重要作用。

当跳过自校准多步推理模块（w/o SCMR）时，模型在下单订单数据集和出库订单数据集上的表现都比较差。这一结果表明，自校准多步推理模块不仅是保障高服务率的重要机制，更是抑制库存率过高、保持整体补货策略平衡的核心支撑。

在取消强化学习模块（w/o RL）的情况下，强化学习模块主要贡献在于细化“低库存”目标的实现，虽然略微牺牲了服务率，但能有效平衡二者关系，提升最终的综合表现。

7.4 参数敏感性分析

由于模型在基于库存调度流程的自校准多步推理方法中，对于同一输入，模型会产生k个推理结果，这些结果会用于进一步更新库存和对强化学习模块进行微调。为了进一步探索参数k对于模型效果的影响，本文分别探索了在 k=2, 3, 5, 7 的情况下模型的表现，具体表现如图 6 和图 7 所示。在本研究中，以 k = 3作为基准设置，并对 k=2, k=5, k=7 的情形进行了对比分析。

结果表明，当 k=2 时，由于仅生成 2 条推理样本，缺乏足够的多样性与数据支撑，模型性能较基准出现明显下滑。而当将 k 增至 5 或 7 时，虽然库存率在一定程度上有所降低、综合得分略有提升，但整体改进幅度有限，且在部分场景下甚至不及基准表现。其根本原因或在于，随着推理样本数量的增加，不确定性同步上升：尽管自校准多步推理机制对输出进行了筛选和优化，仍难免包含若干质量欠佳的推理

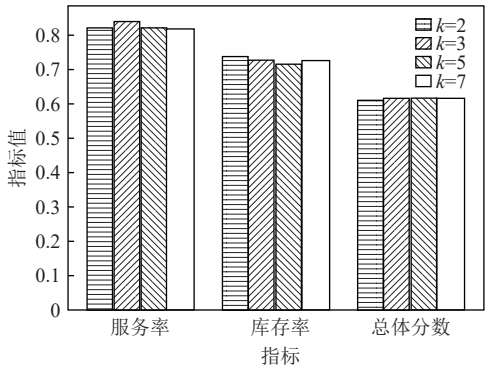


Fig. 6 Parameter sensitivity analysis on order datasets

图 6 下单订单数据集参数敏感性分析

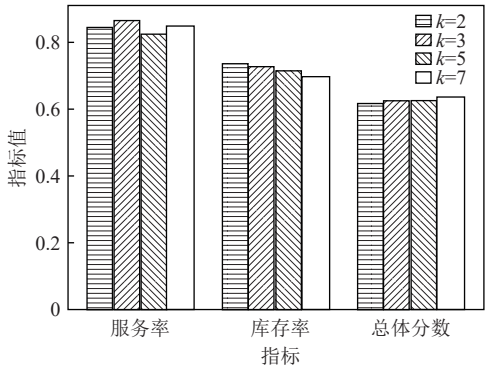


Fig. 7 Parameter sensitivity analysis on outbound order datasets

图 7 出库订单数据集参数敏感性分析

结果,这些参差不齐的样本反而可能在特定环境中削弱决策精度。

7.5 模型规模对决策结果的影响分析

为了研究不同的大模型规模对决策结果的影响,针对模型参数量的消融实验结果如表 5 所示。

所有模型均采用 Qwen2.5 架构,根据 Qwen 官方提供的不同参数规模,兼顾实验的可行性。选择 3 种不同的参数规模,分别是 3B(3×10^9),7B(7×10^9),14B(14×10^9)。7B 版本的模型为主要实验的基础版本,3B 为轻量化版本,14B 为语言能力更强的基线模型。实验结果表明,当模型的参数量增加时,2 个数据集上的库存决策效果均有所提升。但是在下单订单数据集上,7B 基础版本模型仍然获得了最高的服务率,这说明了在真实应用中,垂直领域的 LLM 应用效果并非与模型规模成正比,而依赖于微调机制、优化策略、数据规模等多方因素。

Table 5 Ablation Study Results of Models' Sizes on Two Datasets

表 5 2 个数据集上的模型规模消融实验结果

数据集	模型规模	服务率	库存率	总体分数
下单订单	3B	0.806 8	0.726 7	0.614 4
	7B	**0.843 8**	0.731 5	0.618 7
	14B	0.840 9	**0.728 8**	**0.619 6**
出库订单	3B	0.842 8	0.775 7	0.596 4
	7B	0.865 9	0.728 6	0.623 1
	14B	**0.881 6**	**0.724 4**	**0.625 6**

注:黑体数值表示最优值。

7.6 推理稳定性分析

为了验证所提出的基于多目标平衡的决策优化方法对于模型稳定性的影响,本文将该方法与多个在大模型决策优化中常见的强化学习方法进行对比。采用的数据集为第 3 节中生成的推理轨迹数据集,介绍如下:

1)RLHF(reinforcement learning from human feedback)[58]。该方法首先通过监督微调对预训练模型进行初步对齐,然后使用人工标注的偏好对数据训练奖励模型,最后以近端策略优化等强化学习算法在奖励信号下进一步优化语言模型策略。

2)DPO(direct preference optimization)[59]。该算法通过重新参数化奖励模型,推导出与 RLHF 目标等价的最优策略闭式解,进而以简单的分类损失直接优化语言模型,无需显式训练奖励模型或采样强化学习。

3)CPO(contrastive preference optimization)[60]。该方法在每次策略更新时通过信赖域方法确保近似满足预设约束,旨在训练既能最大化收益又能严格遵守安全或资源等行为限制的策略。

针对稳定性分析的实验结果如表 6 和表 7 所示。总体而言,在 2 个不同数据集上的实验结果均表明,InvLLM 所采用的策略优化方法在总体评分上优于常见的 RLHF, DPO, CPO 方法。原因主要在于其通过多步推理与多目标平衡策略对补货决策进行了更精细的调整,从而在适度牺牲少量服务水平的前提下显著降低库存占用,实现更优的决策输出。这体现了该优化方法在提升模型性能方面的稳定性。

在下单订单数据集中,DPO 在服务率(0.863 8)和库存率(0.744 9)的组合上已较 RLHF 和 CPO 更优,

Table 6 Effectiveness Analysis of Optimization Policies on Order Datasets

表 6 优化策略在下单订单数据集上的效果分析

模型	服务率	库存率	总体分数
RLHF	0.841 9	0.751 9	0.608 2
DPO	**0.863 8**	<u>0.744 9</u>	<u>0.614 7</u>
CPO	<u>0.854 9</u>	0.799 1	0.586 5
InvLLM	0.843 8	**0.731 5**	**0.618 7**

注:黑体数值表示最优值,下划线数值表示次优值。

Table 7 Effectiveness Analysis of Optimization Policies on the Outbound Order Datasets

表 7 优化策略在出库订单数据集上的效果分析

模型	服务率	库存率	总体分数
RLHF	0.875 6	0.742 0	0.617 6
DPO	0.878 5	0.751 0	0.613 4
CPO	**0.889 1**	0.779 1	0.600 4
InvLLM	0.865 9	**0.728 6**	**0.623 1**

注：黑体数值表示最优值，下划线数值表示次优值。

但其总体分数为 0.614 7，略低于 InvLLM 的 0.618 7。InvLLM 在服务率（0.843 8）上低于 DPO 约 0.02，但库存率（0.731 5）进一步下降，库存占用更小，因而在总体评分上领先。相较之下，RLHF 虽保持较高服务率（0.841 9），但库存率（0.751 9）未能进一步优化；CPO 虽然在满足某些约束时可提升服务率（0.854 9），但更高的库存率（0.799 1）严重拖累综合表现。由此可见，RLHF 和 CPO 更侧重保证服务或满足安全/策略约束，而未将库存成本控制纳入深度优化；DPO 侧重直接优化偏好，但其优化目标并未充分体现库存成本与服务水平的长周期权衡；而 InvLLM 通过多步推理结构和多目标奖励设计，能够在考虑补货逻辑连贯性的同时，对库存水平进行细粒度控制，从而在总体评估中取得更高得分。

而在下单订单数据集中，RLHF 和 DPO 的服务率（分别为 0.875 6 和 0.878 5）明显高于 InvLLM（0.865 9）；CPO 在服务率上达到 0.889 1，但库存率（0.779 1）偏高，导致总体分数仅 0.600 4；DPO 和 RLHF 的库存率分别为 0.751 0 和 0.742 0，总体分数分别为 0.613 4 和 0.617 6，也不及 InvLLM 的 0.623 1。InvLLM 虽然服务率略低，但库存率进一步优化至 0.728 6，总体得分显著领先。可见，RLHF 和 DPO 在追求更高服务率时，未充分约束库存成本增长；CPO 在满足约束的同时，往往采用较保守的补货策略，导致库存积压；而 InvLLM 的多步推理和多目标奖励设计使其能够在动态需求和环境信息下评估补货风险，通过对推理步骤和补货量的联合优化，在较低服务牺牲下获得更低库存占用，从而提升整体效果。

7.7 异常输出案例分析

为了对多步推理的完整性、逻辑连贯性和输出的可解释性进行对比分析，并与真实的业务规则进行一致性评估，选取了 Chatglm3-6B，GPT-4，InvLLM 进行比较。针对问题："我想为 2024 年 11 月 1—7 日的保暖内衣产品进行补货，当前库存量为 5 000，我应该在 2024 年 11 月 1 日补充多少货物才能满足未来 7 天的需求？"各模型的输出样例如表 8 所示。

从表 8 中可以看出，InvLLM 借助多源数据融合模块很好地感知了产品单元的相关属性的语义特征，并在后续决策中考虑了产品单元的相关属性；借助基于库存调度流程的自校准多步推理方法，其可以根据库存调度流程规则，进一步按照流程推理结果，并且可以考虑到更细粒度的推理；最终借助强化学习模块，进一步对模型输出的总体结果表现进行评估，优化最终的输出结果，使得其输出兼具可解释性和补货数量精度。

Table 8 Cases of Output from Three LLMs

表 8 3 个大模型的输出样例

模型	输出
Chatglm3-6B	补货量：4 000 解释：根据过去 7 天的需求数据，我们可以计算平均每日需求。…总需求为 (4 531+4 933+5 162+4 817+4 375+4 067+3 685=29 770)，平均每日需求为 29 770/7=4 253,…根据最近的趋势，补货量应该适度减少，…预计补货 4 000 个，…可以平衡接下来 7 天内的库存与需求，…
GPT-4	补货量：32 174 解释：也许我应该写出详细的解释。让我总结一下。首先库存水位为 0，需要补足足够的货物来满足未来 7 天的需求。过去 7 天的需求量为…考虑到波动性，我计算了安全库存，假设 lead time 为 1 天，置信水平为 95%，z 为安全值，安全库存 = 1.658 54 × $sqrt(1)$ ≈ 1 404件，总的补货量为 29 770+1 404=32 174 件…
InvLLM	补货量：27 947 解释：保暖内衣属于季节性商品，其购买时间较为集中，且 11 月是该商品的购买时间，但是不属于快速周转商品…过去 7 天需求量平均波动在 14.9% 左右，未超过 20%，且需求相对稳定，因此可继续使用正常的服务水平进行补货决策…安全库存 SS 的计算公式为 $SS = Z \times \sigma \times \sqrt{7}$ 在前面的步骤中得到了安全系数 Z=2.33，过去 7 天需求量标准差 σ = 515.4，因此安全库存量为 2.33×515.4× $\sqrt{7}$ = 3 177.2…考虑到未来 7 天的交货周期需求量 29 770…当前库存低于重订货点 29 770+3 177.2，应补足 7 天需求量…29 770+3 177.2−5 000=8 584.95…最终确定补货量为 27 947.2。

7.8 提示工程设计

提示工程需要不仅将静态的产品属性信息转化为易于理解的文本形式，还要将时序数据转化为自然语言描述，使得时序信息和产品属性之间的关系更加直观和清晰，进而提高 LLM 对库存需求的准确感知能力。为实现此目标，提示工程主要包含 2 部分：第 1 部分通过给定的产品单元，构造对应产品单元 u 对应的 1 级、2 级、3 级属性文本，其中 3 级属性文本是和当前产品单元最相关的细粒度属性信息，而 1 级、2 级属性文本则是粗粒度属性信息。根据网页查询的结果，返回前 k 个相关度最高的结果。第 2 部分关注的是如何让 LLM 感知如日常产品消耗需求、库存量等时序数据并捕捉不同时间段的变化。这一过程的挑战在于时序数据和文本数据的异构性。因

此采用 Data2Text 模板生成技术,将时序数据转换为自然语言描述,将文本数据与时序相结合,从而使 LLM 能够处理并理解这些数据。这些模板不仅包括固定的文本元素,还包含来自原始时序数据的具体信息,如每日消耗量、库存变化等。通过替换占位符,模板生成技术能够动态地为每个时间点生成描述性的文本,从而使得时序数据变得更加直观易懂。提示工程的构建过程如图 8 所示。

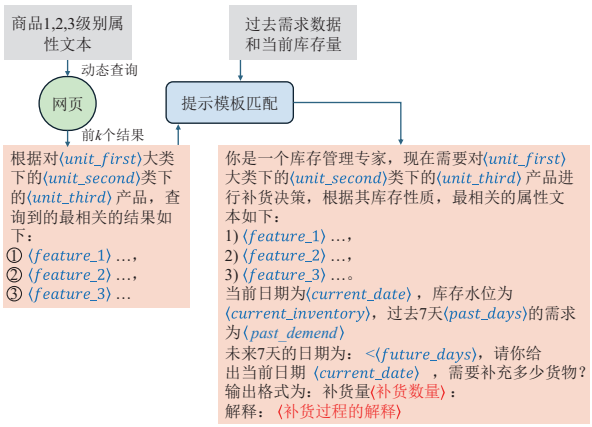


Fig. 8 The process of prompt construction

图 8 提示构建过程

8 结 论

本文针对已有库存调度决策研究工作的不足,在库存调度场景下,设计了 InvLLM 模型。本研究在理论层面提出并系统构建了一个集多源数据融合、库存调度步骤生成、自校准多部推理与多目标强化学习于一体的框架,从整体上解决了库存调度领域中数据异构性、推理误差累积、决策可解释性缺失以及多目标权衡的问题。具体而言,多源数据融合模块通过与维基百科知识库交互,结合时序模板匹配技术,将商品静态属性与需求波动转换成高质量的自然语言描述,为下游推理提供了丰富而精确的背景信息;基于库存调度流程的自校准机制能够在多步推理的过程中,动态识别并拆解“高难度”子任务,显著降低了因过度递归而导致的误差累积;自反思模型则模拟人类专家的反馈闭环,通过对先前决策的错误诊断以及生成修正意见,持续优化模型的推理策略与行动计划。

致谢:本研究工作得到东南大学至善青年学者项目(2242025RCB0040)和东南大学大数据计算中心共享服务平台与优势理科学科交叉平台的支持。

作者贡献声明:林丽、李宗晟负责方案整体设计、实验并撰写论文;王帅提出指导意见并修改论文。

参 考 文 献

[1] Wang Qinxin. Research on supply chain cost control of e-commerce enterprises in the context of big data: A case study of JD. com[J]. E-Commerce Letters, 2024, 13(4): 6735−6743 (in Chinese)
(王沁心. 大数据背景下电商企业供应链成本控制研究——以京东商城为例[J]. E-Commerce Letters, 2024, 13(4): 6735−6743)

[2] Bekiros S. Fuzzy adaptive decision-making for boundedly rational traders in speculative stock markets[J]. European Journal of Operational Research, 2010, 202(1): 285−293

[3] Zhang Xiaoyue, Dai Wenqiang, Cai Xiaoqiang. Inventory replenishment decisions with uncertain price and demand[J]. International Journal of Production Research, 2024, 62(7): 2663−2682

[4] Selukar M, Jain P, Kumar T. Inventory control of multiple perishable goods using deep reinforcement learning for sustainable environment[J]. Sustainable Energy Technologies and Assessments, 2022, 52: 102038

[5] Silvestre B, Gong Y, Bessant J, et al. From supply chain learning to the learning supply chain: Drivers, processes, complexity, trade-offs and challenges[J]. International Journal of Operations & Production Management, 2023, 43(8): 1177−1194

[6] Van C, Baesens B, Lemahieu W. Profarima: A profit-driven order identification algorithm for ARIMA models in sales forecasting[J]. Applied Soft Computing, 2017, 60: 775−785

[7] Fan Zhiping, Che Yujie, Chen Zhenyu. Product sales forecasting using online reviews and historical sales data: A method combining the bass model and sentiment analysis[J]. Journal of business research, 2017, 74: 90−100

[8] AghaeiR, Kiaei A, Boush M, et al. The potential of large language models in supply chain management: Advancing decision-making, efficiency, and innovation[J]. arXiv preprint, arXiv: 2501.15411, 2025

[9] Quan Yinzhu, Liu Zefang. Invagent: A large language model based multi-agent system for inventory management in supply chains[J]. arXiv preprint, arXiv: 2407.11384, 2024

[10] Bi Fenglin, Zhang Qiming, Zhang Jiarui, et al. A retrieval-augmented generation system based on sliding window strategies in large language models[J]. Journal of Computer Research and Development, 2025, 62(7): 1597−1610 (in Chinese)
(毕枫林, 张豈明, 张嘉睿, 等. 基于滑动窗口策略的大语言模型检索增强生成系统[J]. 计算机研究与发展, 2025, 62(7): 1597−1610)

[11] Chen Huimin, Liu Zhiyuan, Sun Maosong. The social opportunities and challenges in the era of large language models[J]. Journal of Computer Research and Development, 2024, 61(5): 1094−1103 (in Chinese)
(陈慧敏, 刘知远, 孙茂松. 大语言模型时代的社会机遇与挑战[J]. 计算机研究与发展, 2024, 61(5): 1094−1103)

[12] Liu Hao, Sferrazza C, Abbeel P. Chain of hindsight aligns language models with feedback[J]. arXiv preprint, arXiv: 2302.02676, 2023

[13] Wei J, Wang Xuezhi, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models[C]//Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2022, 35: 24824−24837

[14] Wang Song, Yang Yang. M-GAN-XGBOOST model for sales prediction and precision marketing strategy making of each product in online stores[J]. Data Technologies and Applications, 2021, 55(5): 749−770

[15] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735−1780

[16] Chen Tianqi, Guestrin C. XGBOOST: A scalable tree boosting system[C]//Proc of the 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2016: 785−794

[17] Li Yuanjiang, Yang Yi, Zhu Kai, et al. Clothing sale forecasting by a composite GRU–Prophet model with an attention mechanism[J]. IEEE Transactions on Industrial Informatics, 2021, 17(12): 8335−8344

[18] Chandriah K, Naraganahalli R. RNN/LSTM with modified adam optimizer in deep learning approach for automobile spare parts demand forecasting[J]. Multimedia Tools and Applications, 2021, 80(17): 26145−26159

[19] Joseph R, Mohanty A, Tyagi S, et al. A hybrid deep learning framework with CNN and bi-directional LSTM for store item demand forecasting[J]. Computers and Electrical Engineering, 2022, 103: 108358

[20] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks[C]//Proc of the 26th Annual Conf on Neural Information Processing Systems. Cambridge, MA: MIT, 2012: 1106−1114

[21] Liu Haoyue, Liu Qing, Liu Min. A novel hybrid sampling method based on cwGAN for extremely imbalanced backorder prediction[C]// Proc of the 2022 IEEE Int Conf on Systems, Man, and Cybernetics (SMC). Piscataway, NJ: IEEE, 2022: 768−773

[22] Shajalal M, Hajek P, Abedin M. Product backorder prediction using deep neural network on imbalanced data[J]. International Journal of Production Research, 2023, 61(1): 302−319

[23] Giannoccaro I, Pontrandolfo P. Inventory management in supply chains: A reinforcement learning approach[J]. International Journal of Production Economics, 2002, 78(2): 153−161

[24] Kim C, Kon I, Baek J. Asynchronous action-reward learning for nonstationary serial supply chain inventory control[J]. Applied Intelligence, 2008, 28: 1−16

[25] Kwon I, Kim C, Jun J, et al. Case-based myopic reinforcement learning for satisfying target service level in supply chain[J]. Expert Systems with Applications, 2008, 35(1/2): 389−397

[26] Chaharsooghi S, Heydari J, Zegordi S. A reinforcement learning model for supply chain ordering management: An application to the beer game[J]. Decision Support Systems, 2008, 45(4): 949−959

[27] Kwak C, Choi J, Kim C, et al. Situation reactive approach to vendor managed inventory problem[J]. Expert Systems with Applications, 2009, 36(5): 9039−9045

[28] Sun Ruoying, Zhao Gang. Analyses about efficiency of reinforcement learning to supply chain ordering management[C]//Proc of the 10th IEEE Int Conf on Industrial Informatics. Piscataway, NJ: IEEE, 2012: 124−127

[29] Dogan I, Guner A. A reinforcement learning approach to competitive ordering and pricing problem[J]. Expert Systems, 2015, 32(1): 39−48

[30] Jiang Chenzhi, Sheng Zhaohan. Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system[J]. Expert Systems with Applications, 2009, 36(3): 6520−6526

[31] Boute R, Gijsbrechts J, Van W, et al. Deep reinforcement learning for inventory control: A roadmap[J]. European Journal of Operational Research, 2022, 298(2): 401−412

[32] De M, Gijsbrechts J, Boute R. Reward shaping to improve the performance of deep reinforcement learning in perishable inventory management[J]. European Journal of Operational Research, 2022, 301(2): 535−545

[33] Kavuk EM, Tosun A, Cevik M, et al. Order dispatching for an ultra-fast delivery service via deep reinforcement learning[J]. Applied Intelligence, 2022, 52: 4274−4299

[34] Wang Hao, Tao Jiaqi, Peng Tao, et al. Dynamic inventory replenishment strategy for aerospace manufacturing supply chain: Combining reinforcement learning and multi-agent simulation[J]. International Journal of Production Research, 2022, 60(13): 4117−4136

[35] Wang Qinghao, Peng Yijie, Yang Yaodong. Solving inventory management problems through deep reinforcement learning[J]. Journal of Systems Science and Systems Engineering, 2022, 31(6): 677−689

[36] Oroojlooyjadid A, Nazari M, Snyder L, et al. A deep Q-network for the beer game: Deep reinforcement learning for inventory optimization[J]. Manufacturing & Service Operations Management, 2022, 24(1): 285−304

[37] Csaszar F, Ketkar H, Kim H. Artificial intelligence and strategic decision-making: Evidence from entrepreneurs and investors[J]. Strategy Science, 2024, 9(4): 322−345

[38] Wasserkrug S, Boussioux L, Sun W. Combining Large Language Models and OR/MS to Make Smarter Decisions[M/OL]//Tutorials in Operations Research: Smarter Decisions for a Better World. 2024[2026-02-19]. https://pubsonline.informs.org/doi/abs/10.1287/educ.2024.0275

[39] Li Beibin, Mellou K, Zhang Bo, et al. Large language models for supply chain optimization[J]. arXiv preprint, arXiv: 2307.03875, 2023

[40] Song Zhe, Xie Ying, Yang Lichao, et al. Large language models in supply chain management: A systematic literature review and application framework[J/OL]. International Journal of Production Research, 2026[2026-04-18]. https://www.tandfonline.com/doi/citedby/10.1080/00207543.2026.2641103?scroll=top&needAccess=true

[41] Jannelli V, Schoepf S, Bickel M, et al. Agentic LLMs in the supply chain: Towards autonomous multi-agent consensus-seeking[J]. arXiv preprint, arXiv: 2411.10184, 2024

[42] Li Zhihong, Ksibi A, Xu Xiaoying. Optimizing inventory management using a multi-agent LLM system[C]//Proc of the 39th Int Conf on Electronic Business. Berkeley, CA: Bepress, 2024: 308−318

[43] Setyadi H, Al A, Widodo P. Implementation economic order quantity

and reorder point methods in inventory management information systems[J]. Journal of Information Systems and Informatics, 2024, 6(1): 103−117

[44] Vania A, Yolina H. Analysis inventory cost jona shop with EOQ model[J]. Engineering, MAthematics and Computer Science Journal, 2021, 3(1): 21−25

[45] Nobil A, Sedigh A, Cardeans B. Reorder point for the EOQ inventory model with imperfect quality items[J]. Ain Shams Engineering Journal, 2020, 11(4): 1339−1343

[46] Chopra S, Meindl P. Strategy, planning, and operation[J]. Supply Chain Management, 2001, 15(5): 71−85

[47] Van P, Broekmeulen R, De K. Multi-echelon inventory management at sligro food group N. V.[D]. Eindhoven, Netherlands: School of Industrial Engineering, Eindhoven University of Technology, 2015

[48] Haque K, Hossain M, Hossain M, et al. Performance evaluation of Bangladeshi apparel and textile supply chain network, a case study[J]. Performance Evaluation, 2011, 1(1): 211−218

[49] Zheng Wendong, Hu Jun. Multivariate time series prediction based on temporal change information learning method[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(10): 7034−7048

[50] VaswaniA, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proc of the 31st Annual Conf on Neural Information Processing Systems. RedHook, NY: Curran Associate Inc, 2017: 6000−6010

[51] Tan Hao. Reinforcement learning with deep deterministic policy gradient[C]//Proc of the 2021 Int Conf on Artificial Intelligence, Big Data and Algorithms. Piscataway, NJ: IEEE, 2021: 82−85

[52] Haarnoja T, Zhou A, Hartikainen K, et al. Soft actor-critic algorithms and applications[J]. arXiv preprint, arXiv: 1812.05905, 2018

[53] Zhan Mengying, Chen Jinchao, Du Chegnlie, et al. Twin delayed multi-agent deep deterministic policy gradient[C]//Proc of the 2021 IEEE Int Conf on Progress in Informatics and Computing. Piscataway, NJ: IEEE, 2021: 48−52

[54] Zeng Aohan, Xu Bin, Wang Bowen, et al. Chatglm: A family of large language models from GLM−130b to GLM−4 all tools[J]. arXiv preprint, arXiv: 2406.12793, 2024

[55] Jones J, Van K, Mueller G. Vicuna2. 0: RISC-V embedded vector unit with half-precision floating-point support for TinyML[C]//Proc of the 2025 Austrochip Workshop on Microelectronics. Piscataway, NJ: IEEE, 2025: 69−72

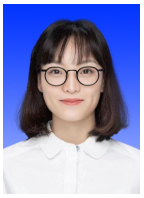
[56] Team G, Georgiev P, Lei V I, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context[J]. arXiv preprint, arXiv: 2403.05530, 2024

[57] Achiam J, Adler S, Agarwal S, et al. GPT-4 technical report[J]. arXiv preprint, arXiv: 2303.08774, 2023

[58] Ouyang Long, Wu J, Jiang Xu, et al. Training language models to follow instructions with human feedback[C]//Proc of the 35th Annual Conf on Advances in Neural Information Processing Systems. Cambridge, MA: MIT, 2022: 27730−27744

[59] Rafailov R, Sharma A, Mitchell E, et al. Direct preference optimization: Your language model is secretly a reward model[C]// Proc of the 36th Annual Conf on Advances in Neural Information Processing Systems. Cambridge, MA: MIT, 2023: 53728−53741

[60] Achiam J, Held D, Tamar A, et al. Constrained policy optimization[C]//Proc of the 34th Int Conf on Machine Learning. New York: PMLR, 2017: 22−31



Lin Li, born in 1993. PhD, assistant professor. Member of CCF. Her main research interests include sequence modeling and big data.

林　丽, 1993 年生。博士，助理教授。CCF 会员。主要研究方向为序列建模、大数据。



Li Zongsheng, born in 1999. Master. His main research interests include large language models and reinforcement learning.

李宗晟, 1999 年生。硕士。主要研究方向为大语言模型、强化学习。



Wang Shuai, born in 1987. PhD, professor. Member of CCF. His main research interests include Internet of things and data science.

王　帅, 1987 年生。博士，教授。CCF 会员。主要研究方向为物联网、数据科学。