供应链人工智能智能体系统：结构化决策提示与记忆检索

Konosuke Yoshizato 日本产业技术综合研究所 筑波，日本 名古屋工业大学 名古屋，日本
yoshizato.konosuke@otsukalab.nitech.ac.jp

Kazuma Shimizu 日本产业技术综合研究所 筑波，日本 NEC公司 东京，日本
smzkzm2019@nec.com

Ryota Higa 日本产业技术综合研究所 筑波，日本 NEC公司 东京，日本
r-higaryouta@nec.com

Takanobu Otsuka 日本产业技术综合研究所 筑波，日本 名古屋工业大学 名古屋，日本
otsuka.takanobu@nitech.ac.jp

arXiv:2602.05524v1 [cs.MA] 5 Feb 2026

摘要
本研究探讨基于大语言模型（LLM）的多智能体系统（MAS），将其视为库存管理的一种有前景的方法，而库存管理是供应链管理的重要组成部分。尽管这类系统因有望应对典型库存管理方法所面临的挑战而受到广泛关注，但其有效性仍存在一些关键的不确定性。具体而言，基于大语言模型的多智能体系统能否持续制定最优订货策略，并适应多样的供应链场景，尚不明确。为回答这些问题，我们研究了一种基于大语言模型的多智能体系统，其提示中包含固定订货策略，编码了问题设定的逐步流程，以及库存管理中常用的安全库存策略。实证结果表明，即使没有对提示进行详细调整，基于大语言模型的多智能体系统也能在受限场景中确定最优订货决策。为增强适应性，我们提出了一种名为 AIM-RM 的新型智能体，该智能体通过相似度匹配利用相似的历史经验。结果显示，AIM-RM 在各种供应链场景中的表现均优于基准方法，凸显了其稳健性和适应性。

关键词
供应链管理；库存管理；大语言模型；多智能体系统

1 引言

多智能体系统（多智能体系统）是能够自主感知环境、作出决策并执行动作的智能系统。这些系统已在许多实际应用中获得广泛关注。然而，要使这类系统有效运行，多个人工智能智能体必须独立且协同地开展行动。在此背景下，大语言模型（大语言模型）的最新进展

本文是被第25届自主智能体与多智能体系统国际会议（AAMAS 2026）接收的扩展摘要的完整版本。

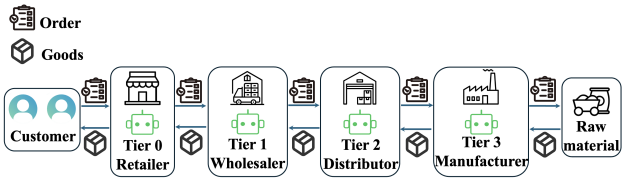

图1：供应链库存管理多智能体系统。在每个层级，智能体向上游智能体下单，并在交付提前期后接收货物。

极大增强了人工智能智能体的能力，使其非常适合多样的工业应用，包括制造业、金融和客户服务 [23, 24, 36, 44]。特别是，基于大语言模型的多智能体系统能够应对现实应用中涉及的复杂交互与谈判。

供应链管理（SCM）是应用基于大语言模型的多智能体系统的一个极具前景的领域。现代供应链面临着日益严峻的复杂性挑战，例如全球化 [4, 17] 和气候变化 [33]。这些挑战的核心是库存管理，它对供应链各环节的采购、生产和分销至关重要。在库存管理中应用多智能体系统，使各企业能够自主决策，同时促进供应链中各智能体之间的高效协调[10, 18, 20, 29]。这种方法既支持单个企业层面的优化，也能为所有利益相关者带来集体效益，这对于应对供应链管理问题至关重要 [16]。

现有库存管理研究主要采用两种主要方法之一。第一种方法使用启发式方法，例如基库存策略。尽管这些启发式方法应用广泛，但它们往往需要针对具体问题进行费力的定制，才能实现高性能 [11]。第二种方法采用强化学习，使系统能够自主形成灵活而有效的策略。然而，强化学习，尤其是

在多智能体和复杂供应链情境中，会产生巨大的计算成本，因而难以用于实际应用 [3, 46]。因此，将该方法应用于实际供应链运营仍然具有挑战性。

近期研究日益关注将基于大语言模型的人工智能体应用于供应链管理（SCM） [15, 19, 22, 35, 42]。其中许多研究通过检索增强生成（RAG） [21]，利用制造业中的知识共享和领域知识来解决库存管理问题。然而，大语言模型近期取得了长足进步，例如 GPT-5 [31] 和 o4-mini [32], 展现出强大的推理和数值计算能力。这些能力表明，最先进的模型无需依赖外部信息源，只需采用有效的决策策略，就能在供应链中高效地进行库存管理。因此，必须评估当前大语言模型基础模型独立处理库存管理任务的能力，因为这项评估是探索人工智能智能体在供应链优化领域潜力的关键第一步。

我们的首要目标是探究当前基于大语言模型的多智能体系统的局限性。了解这些限制对于实现第二个目标至关重要：设计能够适应多样供应链情境的基于大语言模型的多智能体系统。一项关键挑战在于，此类系统往往依赖情境特定提示词，这可能导致输出不稳定、性能不可靠 [30, 37, 40, 45]。此外，提示词调优过程往往需要投入大量精力 [9, 25], , 因此可能不切实际。为解决这些障碍，我们提出了一种整合采购策略与历史交易数据的方法。该方法旨在提升上下文感知能力和协同决策能力，使基于大语言模型的多智能体系统能够克服传统方法面临的挑战，并改善真实世界供应链管理中的优化效果。

1.1 我们的贡献

本研究有两个主要目标。首先，我们探讨了基于大语言模型的多智能体系统在库存管理领域的当前局限性。为实现这一目标，我们构建了模拟各种供应链情境的实验场景，并考察多智能体系统的表现。实验结果表明，在特定条件下，供应链中的基于大语言模型的多智能体系统能够成功识别并执行最优订货策略，无需针对场景调整提示。据我们所知，这是首项证明基于大语言模型的多智能体系统能够为多级库存管理问题找到最优解的研究。然而，结果也表明，即使基于大语言模型的多智能体系统在某一场景中找到了最优订货策略，也无法在不同的供应链场景中识别最优策略，这揭示了基于大语言模型的多智能体系统的一项局限性。

其次，我们旨在设计一种自适应的基于大语言模型的多智能体系统，使其在多样的供应链情境中稳健运行。为实现这一目标，我们提出一种方法，使智能体能够检索并参考与当前情境相似的过往经验。具体而言，通过将强化学习结果存储为过往经验，我们提出的多智能体系统能够隐式协调分布式智能体，并提供连贯的系统级行为，使基于大语言模型的多智能体系统能够灵活适应多样的供应链场景。因此，我们的研究既指出了基于大语言模型的多智能体系统在库存管理中的能力与局限，也提出了一种提升其适应性的方法。

2 文献综述

多层供应链中的库存管理问题被建模为多级库存管理问题，多年来已得到广泛研究（详情见 [7] ）。一些现有研究将这一问题建模为一种马尔可夫决策过程（MDP），并使用动态规划推导出最优策略 [38]。然而，由于问题复杂且数学模型难以处理，基库存策略等启发式方法在实践中较为常用 [11, 46]。

多级库存管理问题已在 OpenAI OR-gym 仿真库中实现 [14], , 从而便于应用多智能体强化学习方法。该领域的大量研究聚焦于集中训练与分散执行（CTDE）架构 [18], , 该架构介于完全集中式模型与完全分散式模型之间，使智能体在训练期间能够获取全局信息，同时以分散方式执行学习策略。尽管 CTDE 及相关强化学习方法 [8, 11, 26, 28] 往往表现出较强性能，但其实际应用仍受限于高样本复杂度，以及必须构建经真实数据校准的仿真器等挑战 [27]。

近年来，将基于大语言模型的多智能体系统应用于多级库存管理问题的研究显著加速。研究 [36] 提出了 InvAgent，并证明与启发式方法和强化学习方法相比，它能够使决策适应各种需求场景。结果表明，简单的基于大语言模型的多智能体系统即可实现高效的供应链管理。后续研究考察了智能体之间通过类人谈判与沟通进行有效决策的方式 [15, 19]。除仅利用基础大语言模型外，研究 [42] 还提出通过检索增强生成，借助外部知识源增强智能体能力，并使用相关资料，如教科书和制造业文献。尽管这些研究显著推进了基于大语言模型的智能体与供应链库存管理的融合，但简单的基于大语言模型的智能体能否实现真正的最优策略仍不明确。此外，这些智能体系统在复杂库存管理情境中的根本局限尚未得到充分阐明。

一项近期研究表明，提高大语言模型的推理力度并不总能带来更好的任务表现[12]。这种反直觉现象被称为过度思考，发生在推理令牌数量超过某一阈值时。推理模型的这一特性至关重要，因为它似乎会限制基于大语言模型的多智能体系统在执行多级库存管理问题等复杂任务时的有效性。

3 方法论

本节首先介绍本研究所考虑的供应链库存管理问题，然后给出我们的模型。

3.1 多层级库存管理

多层级供应链中的库存管理问题被建模为多级库存管理问题。在该模型中，供应链的每个层级都会存储物料，并将其用于生产。层级按升序编号，其中层级 0 代表零售商。在每个层级，生产一个单位的产品需要一个单位的物料，而每周期的最大产量受设施产能 c_m限制。成品不能作为库存储存，而是立即发往下游层级。在层级 m 生产的产品会发往层级 $m+1$，并在预先确定的运输提前期 L_m后到达。到达后，这些产品成为层级 $m+1$ 的库存，并用于后续生产。图 2 展示了供应链流程。

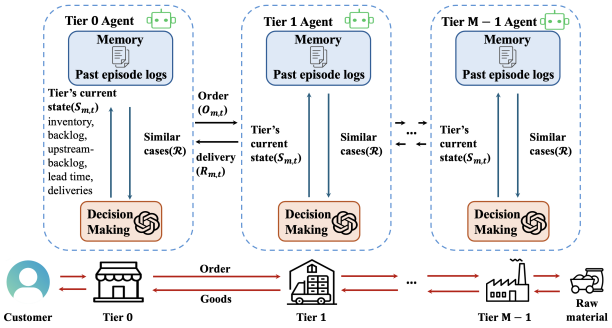

图 2：智能体与供应链环境交互的框架。从下游智能体开始，各层级的智能体在每一轮观测其状态，并向上游智能体下订单。我们提出的智能体通过相似度匹配检索历史经验数据，并在下订单时利用这些数据。

接下来，我们说明一个回合内的运行流程。每个回合包含T个周期。在每个周期$t \in [T] = \{1、\ldots\ldots、T\}$，各层级的智能体依次执行以下四个步骤：

（i）库存补货：交付 L_m 个周期前订购的物料，并补充库存水平。

（ii）订货与需求观测：各层级 $m \in M$向供应商订购$O_{m,t}$单位物料。同时，观测买方的需求，并尽可能利用现有库存满足需求。

(iii)产品发运：每个智能体确定其发货数量 $R_{m,t}$，该数量受生产能力或可用库存的限制，取两者中的较低值。发货后，剩余库存为 $I_{m,t}$。任何未满足的需求 $B_{m,t}$都会形成积压订单，并产生机会成本，且会优先在未来周期中得到满足。

(iv) 损益计算：每个智能体通过汇总产品销售收入、物料采购成本、积压惩罚和库存持有成本来计算其损益$P_{m,t}$。

上述数量可表示如下 [14]:

$$R_{m,t} = \min(B_{m+1,t-1} + O_{m,t}, c_{m+1}, I_{m+1,t-1} \tag{1}$$

$$+ R_{m+1,t-L_{m+1}}), m \in \mathcal{M}, \tag{2}$$

$$R_{M-1,t} = O_{M-1,t}, \tag{3}$$

$$S_{m,t} = R_{m-1,t}, \tag{4}$$

$$S_{0,t} = \min\left(B_{0,t-1} + D_t, c_0, I_{0,t-1} + R_{0,t-1}\right) \tag{5}$$

$$B_{m,t} = B_{m,t-1} + O_{m-1,t} - S_{m,t} \tag{6}$$

$$B_{0,t} = B_{0,t-1} + D_t - S_{0,t} \tag{7}$$

$$I_{m,t} = I_{m,t-1} + R_{m,t-L_m} - S_{m,t} \tag{8}$$

$$P_{m,t} = P_m S_m - r_m R_{m,t} - k_m B_{m,t} - h_m I_{m,t} \tag{9}$$

3.2 多级供应链管理中的智能体行为

在此问题设定中，每个层级的智能体都会选择一个动作$a_{m,t} = O_{m,t}$，该动作会影响环境并导致奖励、库存水平等变量发生变化。这一过程可以建模为 MDP。智能体与环境之间的交互通过用户代理实现。具体流程如下：

(i) 在第一轮开始时初始化环境。

(ii) 每个层级的智能体从环境中观察当前轮次的状态。

(iii) 观察到自身状态 $s_{m,t}$后，每个智能体决定其动作 $a_{m,t}$并将其发送给用户代理。

（iv）用户代理将每个智能体的动作传输至环境，环境随后返回下一状态$s_{m,t+1}$ 以及该轮次的奖励。

（v）用户代理将更新后的状态和奖励传递给每个智能体。当当前轮次尚未达到最终轮次 T时，流程返回步骤（ii）。状态$s_{m,t} \in \mathbb{R}^{4+2L_m}$定义为

$$s_{m,t} = [I_{m,t-1}, B_{m,t-1}, B_{m+1,t-1}, S_{m,t-L_m}, \ldots, S_{m,t-1}, R_{m,t-L_m}, \ldots, R_{m,t-1}]. \tag{10}$$

3.3 提出的智能体

在此，我们提出AIM-RM，这是一种能够利用过去类似情境中所得洞见的人工智能智能体。该智能体由两个主要模块组成：记忆模块和决策模块。下面将介绍各模块的具体细节。

该模块使用决策提示P_{DM}来确定订购动作。决策提示P_{DM}是用于决策的基础提示，其中包含必要信息，包括当前轮次 t、层级编号 m、当前状态$s_{m,t}$、历史数据H以及需求描述D。历史数据H仅在本节介绍的记忆模块可用时使用。需求描述D描述了所有轮次中的客户需求。作为示例，附录B提供了一个基于 [36]提出的提示所构建的决策提示示例。在接收到该提示后，决策模块不仅会生成动作，还会生成每个决策背后的推理，其形式化表示为

$$DE_{m,t} = \mathrm{LLM}_{\mathrm{DM}}(\mathcal{P}_{\mathrm{DM}}(s_{m,t}, \mathcal{H}, \mathcal{D})). \tag{11}$$

此处，决策输出 $DE_{m,t}$ 由订货量$O_{m,t}$和理由组成。

为便于大语言模型推导最优策略，我们对决策提示 P_{DM}进行了调整。然而，[36]中提出的初始版本并未明确包含多级库存管理流程中的关键信息，例如订单到达和订单时序。这一遗漏可能会降低大语言模型所做订货量决策的准确性。为解决这一局限，我们修订了 P_{DM}，加入了逐步说明 P_{SD}，清晰概述了第 3.1 节定义的流程。此项修改旨在加深大语言模型对库存管理流程的理解，并支持其做出更准确的决策。为将逐步说明整合进决策提示，P_{SD} iswas 被插入到 P_{DM}之后。作为参考，附录B中提供了逐步说明 P_{SD} 的示例

尽管逐步说明 P_{SD} 概述了环境，但并未具体说明订货策略。为此，我们在 P_{DM}中加入安全库存策略，通过同时考虑提前期和预测需求来确定订货量，并纳入安全库存（SS）。首先，我们将计划库存增量定义为 $\tilde{I}_{m,\,t} = I_{m,\,t-1} + \sum_{\tau=t-L_m}^{t-1} R_{m,\,\tau} - B_{m,\,t-1}$，表示未来一个提前期内库存水平的增加量。然后，我们分别将安全库存和未来消耗量定义为 $SS = z\hat{\sigma}\sqrt{L_m + 1}$ 和 $\tilde{C}_{t,\,m} = (L_m + 1)\hat{\mu} + SS$。其中，$\hat{\mu}$是预测需求，$\hat{\sigma}$是需求方差，$z$ 是由目标服务水平决定的安全系数。该安全库存公式是库存管理中使用的传统公式 [13]。随后，根据 $\tilde{D}_{t,\,m}$与 $\tilde{C}_{t,\,m}$之间的差额确定订货量。为将安全库存策略纳入决策过程，我们将 P_{SS} 插入 P_{SD}之后。该安全库存策略的提示示例见附录B。

为确保在多样场景中实现稳健性能，而无需进行大量提示调优，我们引入了用于存储历史的记忆 M 的外部实现，其中 M 是 $(s_{m,t}, O_m, P_m)$ 的集合。该记忆保存在向量数据库中，从而能够基于状态向量相似度检索相似的过往情形。为进一步提升对历史经验的有效利用，我们将一条说明如何使用该记忆的提示纳入决策提示 P_{DM}，具体做法是在 P_{SD} 之后插入记忆使用提示 P_{MU}。为便于说明，P_{MU} 的一个示例见附录B。

AIM-RM 不仅可以在其历史中存储历史回合，还可以存储多级库存管理问题的仿真结果，从而在分布式架构中实现间接协调，并产生连贯的全系统行为。例如，在强化学习仿真中，当每个智能体的最优动作已被指定时，存储每个智能体对应的那部分结果，可使它们通过所存储的结果选择与其他智能体的动作隐式协调的动作。

为了高效利用记忆函数，通过计算当前状态向量与历史经验的状态向量之间的欧几里得距离，使用向量表示执行相似度搜索。若计算得到的距离低于预定义阈值，则相应的过去的情节被视为相似并被选作参考。此外，智能体最多可检索并利用 K 个最接近的匹配结果。

算法 1： 一个周期内所有阶段的单轮决策过程
在一个周期内

输入： 当前周期 t，环境 Env，当前可观测状态 $s_{m,t}$，场景 场景
输出： 动作字典 action_dict
数据： 嵌入映射 ϕ: S $\to \mathbb{R}^d$; 分阶段记忆 $\{M_m\}$; 阈值 τ; 邻居数量 K; 决策提示 P_{DM}; 需求描述 D
设置动作_dict $\leftarrow$ {}
对于 $m \leftarrow 0$ **至** $M-1$ **执行**
  根据以下条件，从 M_m 中选择最近邻 N_K
  欧几里得距离 $d = \|\phi(s_{m,t}) - v\|_2$，
  按升序排列，其中 v 是状态嵌入向量 $\ell \in M$ 的 m
  应用阈值 τ筛选相似案例 R:
  $R = \{n \in N_K \mid d_i < \tau\}$。
  获取决策大语言模型的输出，输入为 P_{DM}、R、D 和 $s_{m,\,t}$:
  $(O_{m,\,t}$、理由$) \leftarrow$ 大语言模型$_{DM}(P_{DM}(s_{m,\,t}$, R、D$))$。
  将动作 $O_{m,t}$ 提交给环境，并观察下一状态和奖励:
  $(s_{m,\,t+1},\ P_{m,\,t}) \leftarrow$ 环境$(m,\ O_{m,\,t})$。
  记录此阶段的动作:
  动作_dict$[m] \leftarrow O_{m,\,t}$。
  使用当前记录更新记忆:
  $M_m \leftarrow M_m \cup \{(v_{m,\,t},\ O_{m,\,t},\ P_{m,\,t})\}$。
end
返回动作_dict

备注 1. 相似度匹配过程的计算成本估计为 $O(|M|d + X + K)$，其中 $|M|$ 是记忆 M 的基数，d是嵌入空间的维度，X是排序的计算成本，K是邻居数量。$O(|M|d)$ 项来自计算 M 中每个向量与当前状态之间的欧几里得距离。$O(K)$ 项来自识别相似度低于阈值的记忆。当使用某些快速排序算法时，计算成本 X 可为 $O(|M|\log(|M|))$，而在最坏情况下可为 $O(|M|^2)$。总体而言，只要 $M$and d不同时很大，相似度匹配过程所需的计算成本就较低。

4 实验

本节首先介绍本研究所考虑的多级库存管理问题的实验设置。接下来，我们介绍用于比较的基线方法。最后，我们展示数值实验的结果，包括对总奖励的分析。

4.1 实验设置

我们将同一智能体分配给多级库存管理问题中的每个层级，以构建多智能体系统。为确保每个智能体了解其各自的层级，我们提供如 [36], 所示的系统提示，该提示见附录B。

表1：实验场景的参数。列表中的数值依次从左到右对应阶段0至阶段3的取值。

场景	均匀	多样
阶段数	4	4
周期数	12	12
初始库存	[12,12,12,12]	[12,14,16,18]
提前期	[2,2,2,2]	[1,2,3,4]
产品产能	[20,20,20,20]	[20,22,24,26]
销售价格	[0,0,0,0]	[9,8,7,6]
订货成本	[0,0,0,0]	[8,7,6,5]
积压成本	[1,1,1,1]	[1,1,1,1]
持有成本	[1,1,1,1]	[1,1,1,1]

我们提供五种实验场景，其供应链参数设置和客户需求模式各不相同。对于供应链参数配置，我们考虑两种设置：统一设置，即所有层级共享相同参数，例如初始库存和提前期；以及多样设置，即这些参数在各层级之间有所不同。实验场景中使用的具体参数列于表1。对于客户需求模式，我们使用三种模式：恒定，即每轮次需求固定为4；递增，即需求随每轮次按 $D_{\text{inc}} = 2 + \lceil t/3 \rceil$ 增长；以及递减，即需求按 $D_{\text{dec}} = 2 + \lceil (12 - (t-1))/3 \rceil$ 下降。此处，$\lceil x \rceil$ 表示大于或等于 x 的最小整数。所有智能体均通过 $\mathrm{P_{DM}}$ 获知客户需求，其中明确了需求函数，并说明了每轮次出现的需求量。供参考，提示中的需求描述见附录B。

就实验配置而言，我们提供四种不同的智能体配置和两种大语言模型参数设置。前两种配置不使用记忆函数：一个智能体使用整合了分步描述 $\mathrm{P_{SD}}$ 的决策提示 $\mathrm{P_{DM}}$，标记为InvAgent（含步骤描述）；另一个智能体使用同时包含 $\mathrm{P_{SD}}$ 和安全库存策略 $\mathrm{P_{SS}}$ 的 $\mathrm{P_{DM}}$，记为InvAgent（含步骤和安全库存策略）。另外两种配置涉及记忆函数：一个模型不预加载任何先前的历史日志，但在回合期间存储历史日志，标记为AIM-RM（不含强化学习日志）；另一个模型使用来自IPPO评估数据的历史日志进行初始化，记为AIM-RM（含强化学习日志）。AIM-RM（含强化学习日志）检索 $K = 6$ 个最近邻，并以阈值 $\tau = 2$ 识别相似案例。在这两种基于记忆的配置中，智能体均使用带有分步描述 $\mathrm{P_{SD}}$ 的决策提示 $\mathrm{P_{DM}}$。特别是，对于AIM-RM（含强化学习日志），每个智能体的提示还包含记忆使用提示 $\mathrm{P_{MU}}$，用于说明如何使用检索到的相似案例。对于大语言模型参数配置，每种设置中的所有智能体均使用OpenAI o4-mini，推理参数设置为高 或中。为评估这些配置的性能，我们进行5个回合，以评估全部轮次和层级上的总奖励；该回合数量的确定旨在平衡大语言模型使用成本、时间消耗以及回合之间的变异性。

比较对象。 InvAgent是一种以o4-mini作为决策模块的智能体，并采用两种推理力度参数高 和中。InvAgent直接采用 [36]中提出的提示。按照该参考文献中描述的设置，引入了两种变体：一种纳入额外策略（InvAgent（含策略）），另一种不包含该策略（InvAgent（不含策略））。此外，为比较推理模型与非推理模型，我们提供了两种使用GPT-4.1而非o4-mini的InvAgent变体。每种条件均通过5次独立试验进行评估。

对于基于启发式方法的比较对象，我们纳入了两种具有代表性的启发式方法：基库存策略（Base-Stock）和需求跟踪策略（Tracking-Demand）。在基库存策略下，智能体每一步都会下达足够的订单，使库存达到预先设定的上限，具体订购 $(c_m - I_{m,\,t})$ 个单位。在需求跟踪策略下，智能体根据最长提前期周期 $\bar{S}$ 内的平均销售量，力求维持目标库存水平 $\bar{I}_{t,\,m}$。每一步，它都会订购 $\max\{\bar{I}_{m,t} - I_{t,m}, 0\}$ 个单位，其中 $\bar{S}_{m,\,t-1} = \sum_{u=1}^{L_{\max}} S_{t-u}/L_{\max}$ 和 $\bar{I}_{t,\,m} = \bar{S}_{m,\,t-1} L_m + B_{m,\,t-1}$。这里，$\bar{S}_{m,\,t-1}$ 表示最长提前期周期内的平均销售量，而 $\bar{I}_{t,\,m}$ 表示相应的目标库存。

作为 CTDE 框架下基于学习的基线方法，我们采用独立近端策略优化 IPPO [8]和多智能体近端策略优化 MAPPO [43]。IPPO 使用近端策略优化（PPO）算法 [39]，对每个智能体进行独立训练。该方法在所有智能体之间共享参数，使用单一共享策略来提高训练效率。MAPPO 是 PPO 的一种多智能体变体。该方法采用以所有智能体的全局观测为条件的集中式价值函数，从而减少并发多智能体学习所导致的非平稳性。为了确定这些模型在各个实验场景中的超参数，我们从候选集合中随机选取超参数组合，其中包括隐藏层大小、学习率和批量大小，并选择得分最高的组合。随后，我们在 100 次试验中评估平均奖励。超参数选择的详细信息见附录 A.3。

我们可以通过在条件 (1)–(8) 下求解收益最大化问题，确定所有层级的最优总奖励。最优策略奖励记为 Opt。在本实验中，我们使用相对最优差距 $\Delta = |(\mathrm{Opt} - r)/\mathrm{Opt}|$ 评估性能，其中 r 是策略的总奖励。所有实验场景的最优值见附录 A.1。

备注 2。我们与[36] 实验的一个关键区别在于，我们采用了确定性的客户需求设置，消除了随机性。这是因为大语言模型的使用成本较高，难以为涉及随机需求的场景 획보统计上充足的试验次数。我们认为，我们的实验设置仍足以评估智能体能否成功应对需求趋势。

注3。问题难度设置的复杂程度在不同实验场景中有所差异，这可以解释为何模型在某些场景中能够成功，而在其他场景中则不能。我们的动机之一

表2：本表给出了每个模型在各回合中的平均总奖励。加粗数字表示除强化学习模型（IPPO、MAPPO）之外，各模型中的最高得分。奖励以相对差距 Δ = |(最优值 − *r*)/最优值|表示，其中最优值 为最优值， *r* 为总奖励。第一行中的术语*const-uni* 表示常数均匀设置。同样，第一行中的其他名称是不同设置名称的缩写。前四个模型是我们提出的模型。

模型（推理力度：中）	Const-Uni	Dec-Div	Dec-Uni	Inc-Div	Inc-Uni	平均值
InvAgent（含步骤描述）	3.33	79.22	437.78	104.55	137.88	152.55
InvAgent（含步骤描述和 SS策略）	**0.00**	115.06	315.56	264.88	425.00	224.10
AIM-RM（不含强化学习日志）	69.17	66.27	219.11	162.64	175.77	138.59
AIM-RM（含强化学习日志）	60.00	**56.02**	**171.11**	**95.12**	**74.09**	**91.27**
InvAgent（GPT-4.1，不含策略）	120.00	244.58	991.11	474.79	262.88	418.67
InvAgent（GPT-4.1，含策略）	76.67	88.25	428.89	133.06	181.81	181.74
InvAgent（不含策略）	46.67	100.30	202.22	171.90	93.18	122.85
InvAgent（含策略）	183.33	118.67	200.00	179.33	155.30	167.33
基库存	146.67	140.36	340.00	162.81	112.12	180.39
需求跟踪	200.00	150.30	584.44	205.37	243.93	276.81
IPPO	0.00	25.30	137.78	38.01	12.88	42.79
MAPPO	30.00	16.27	60.00	44.21	19.70	34.04

表3：模型的总奖励结果如表所示，采用与表2相同的描述风格。为避免重复，省略了部分基准方法。

模型（推理力度：高）	Const-Uni	Dec-Div	Dec-Uni	Inc-Div	Inc-Uni	平均值
InvAgent（含步骤描述）	3.33	**48.80**	206.67	147.93	265.90	**134.53**
InvAgent（含步骤描述和 SS策略）	**0.00**	129.22	342.22	344.63	437.88	250.79
AIM-RM（不含强化学习日志）	36.67	80.48	185.33	130.90	384.55	163.59
AIM-RM（含强化学习日志）	19.33	50.90	188.89	**104.13**	390.45	150.74
InvAgent（不含策略）	46.67	66.87	**177.78**	412.81	800.00	300.83
InvAgent（含策略）	190.00	68.07	666.67	127.27	**151.51**	240.70

此场景配置旨在探究 AIM-RM 能否适应这些差异。例如，在递增和递减设置中，各阶段都需要销售 54 单位。因此，在均匀设置中，所有阶段的初始库存总量不足，制造商必须购买原材料以避免积压订单。然而，在多样设置中，制造商无需满足这一要求。

4.2 实验结果与讨论

本节介绍实验结果。首先，我们讨论总体结果，随后对每个场景的结果进行详细分析。最后，我们讨论推理力度的影响。补充数值结果，包括带有标准差的结果以及使用GPT-5得到的结果，见附录A.1。

4.2.1 总体结果。 表2和表3分别给出了基于o4-mini的多智能体系统在中等和高推理力度下的结果。这些表给出了平均总奖励的相对差距 Δ = |(Opt − *r*)/Opt|，其中Opt为最优值，*r* 为平均总奖励。

如这些表格所示，安全库存策略使基于 o4-mini 的多智能体系统在恒定需求场景中实现了最优值。然而，在涉及时变需求的场景中，安全库存策略并不充分，实际效果也较为有限。这一局限源于安全库存计算通常依赖历史需求数据进行预测，因此系统无法适应新出现的需求趋势变化。相比之下，采用中等和高推理力度的 AIM-RM（含强化学习日志）在所有模型中都获得了较高的平均奖励。值得注意的是，在采用中等推理力度时，它在所有实验场景中均排名第一，并且在平均奖励方面达到了与强化学习基准方法 IPPO 和 MAPPO 相当的性能。对 AIM-RM（含强化学习日志）和 AIM-RM（不含强化学习日志）的比较进一步表明，将强化学习日志纳入 AIM-RM 后能够提升性能，这说明 AIM-RM 对这些日志进行了恰当的利用。这些研究结果表明，AIM-RM（含强化学习日志）具备稳健的适应性和有效性，适用于各种供应链情境。

4.2.2 恒定-均匀。在恒定-均匀场景中，如表2和表3所示，步骤描述与安全库存策略相结合可获得最优奖励，而仅使用步骤描述也能取得良好表现。InvAgent（使用步骤描述和策略）的详细行为如图3所示。从图中可以看出，供应链中的每个智能体

都通过使到货与销售精确匹配，将库存和积压订单均维持在零，从而最大限度地降低持有成本。值得注意的是，所有智能体都成功理解了阶段结构，并始终在各自位置的最佳时机订购四个单位，从而执行最优策略。相比之下，包括MAPPO在内的其他模型未能始终确定最优订货量。

4.2.3 下降多样化场景。 如表2和表3所示，AIM-RM（使用强化学习日志）的表现优于其他基准方法，在中等推理力度下排名第一，在高推理力度下排名第二。其优异表现源于智能体能够回顾历史日志，从而同时依据日志和需求趋势下订单。此外，每个层级的智能体都能考虑各自的提前期并成功下订单，使所有层级的订货行为保持一致，整体表现也更加优越。表中显示，对于InvAgent（不使用策略），随着推理力度增加，其表现不断提升，最终达到与AIM-RM（使用强化学习日志）相当的水平。这是因为在高推理力度下，InvAgent（不使用策略）能够根据未来需求预测和提前期的具体数值，成功确定订货量。相比之下，在中等推理力度下，智能体不会依据具体数值下订单。

4.2.4 下降均匀场景。 在下降均匀场景中，如表2和表3所示，AIM-RM（含强化学习日志）总体上优于其他基准方法，在中等推理力度下排名第一，在高推理力度下排名第三。尤其是在高推理力度下，InvAgent（不含策略）的表现略优于AIM-RM（含强化学习日志）。

4.2.5 上升多样化场景。 在上升多样化场景中，如表2和表3所示，AIM-RM（含强化学习日志）在中等和高推理力度下均优于其他基准模型。这一优异表现源于与下降多样化场景中观察到的相同机制：使用强化学习日志使智能体能够根据历史数据和当前状况灵活地下单。相比之下，InvAgent（含步骤描述）的表现随着推理力度增加而变差。这种性能下降源于对供应链下游订货量的低估，该策略旨在最大限度地降低持有成本。这种低估与不断上升的需求趋势相结合，导致积压订单不断累积并引发订单量骤增，触发牛鞭效应，最终使奖励降低。

4.2.6 上升均匀场景。如表2所示，在上升均匀场景中，AIM-RM（带强化学习日志）在中等推理力度下取得了最佳表现。然而，表3显示，其表现随着推理力度的提高而下降。图4表明，高推理力度下的 AIM-RM 保持了较低的库存，相较于中等推理力度下的 AIM-RM，降低了更多持有成本，但也因此造成库存短缺，并导致上游层级出现突发的大额订单，尤其是在分销商和制造商处。这种行为会导致牛鞭效应。相比之下，InvAgent（带策略）在所有层级都下了大额订单，却没有任何数值依据，结果意外地与递增需求相吻合，积压订单也更少。因此，其总奖励优于 AIM-RM 的各个变体。

4.2.7 推理力度的影响。我们的结果表明，提高推理力度并不能始终改善性能。相反，其影响取决于提示的复杂程度。如表2和表3所示，提示较复杂的 AIM-RM（有强化学习日志）和 AIM-RM（无强化学习日志）在几乎所有场景中，性能都随着推理力度的提高而下降。相比之下，在提示相对简单的情况下，使用中等推理力度的 o4-mini 的 In- vAgent（有策略）和 InvAgent（无策略）均优于使用 GPT-4.1 的 InvAgent 变体。此外，在这些简单提示设置中，InvAgent 变体采用高推理力度时的表现通常优于采用中等推理力度时的表现。这些结果表明，提高推理力度有助于提升简单提示下的性能，但可能会损害复杂提示下的性能：例如，对于 InvAgent（有步骤和安全库存）和 AIM-RM（有强化学习日志）等模型，推理力度越大，性能反而越低。这些模式可能与过度思考 [12],的影响有关；过度思考是推理模型面临的关键问题，值得进一步研究。我们的实验为这一影响提供了新的佐证。

5 结论

本研究探讨了用于多级库存管理的基于大语言模型的多智能体系统，其中每个层级的智能体都试图通过精准订货来降低成本。本研究有两个主要目标。首先，我们考察了基于大语言模型的多智能体系统在库存管理领域的现有限制。实验结果表明，在常数均匀设置下，基于大语言模型的多智能体系统能够借助描述库存管理问题流程的提示和标准安全库存策略，识别并实施最优订货策略。然而，这类多智能体系统无法在不同的供应链场景中有效运行，凸显了基于大语言模型的多智能体系统的局限性。第二个目标是提升适应性。为此，我们提出了基于大语言模型的多智能体系统 AIM-RM，该系统通过相似度匹配利用相似的历史经验。数值实验表明，AIM-RM 具有稳健性和适应性，优于不同的基准方法，且性能与强化学习方法相当。

我们发现本研究有三项主要局限。首先，相似度匹配采用固定的相似度度量（欧几里得范数）和相似度阈值 $\tau = 2$。这一局限对性能的影响仍需进一步研究。其次，AIM-RM（含强化学习日志）有时似乎会在未考虑输入中的相似案例的情况下下单。然而，推理模型往往会直接输出推理结果，而不执行中间推理过程 [5]。因此，在我们的实验设置中，很难详细评估这一问题。关于推理力度对性能的影响，我们观察到，提高推理力度并不一定能提升性能。要探究这一影响，需要对 OpenAI 以外的推理模型开展大量研究，包括 Gemini 2.5 [6] 和 Claude Opus [2]。然而，这些研究超出了本研究的范围。

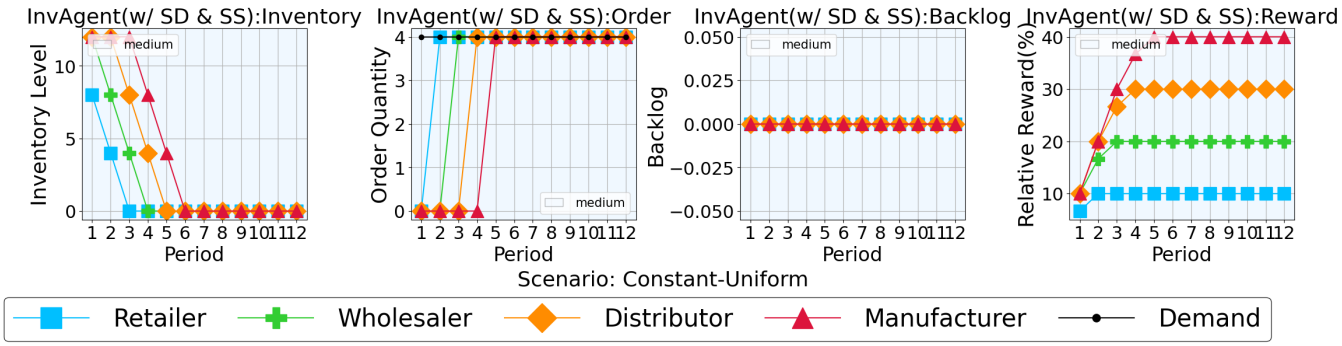

图3： InvAgent **（包含步骤描述和安全库存策略）在中等推理力度下的结果。这四张图从左到右分别展示了一个回合内的库存、积压订单、订货量和相对奖励** （r/最优值）。**“需求”图例对应客户需求，该需求绘制在订货量图中。**

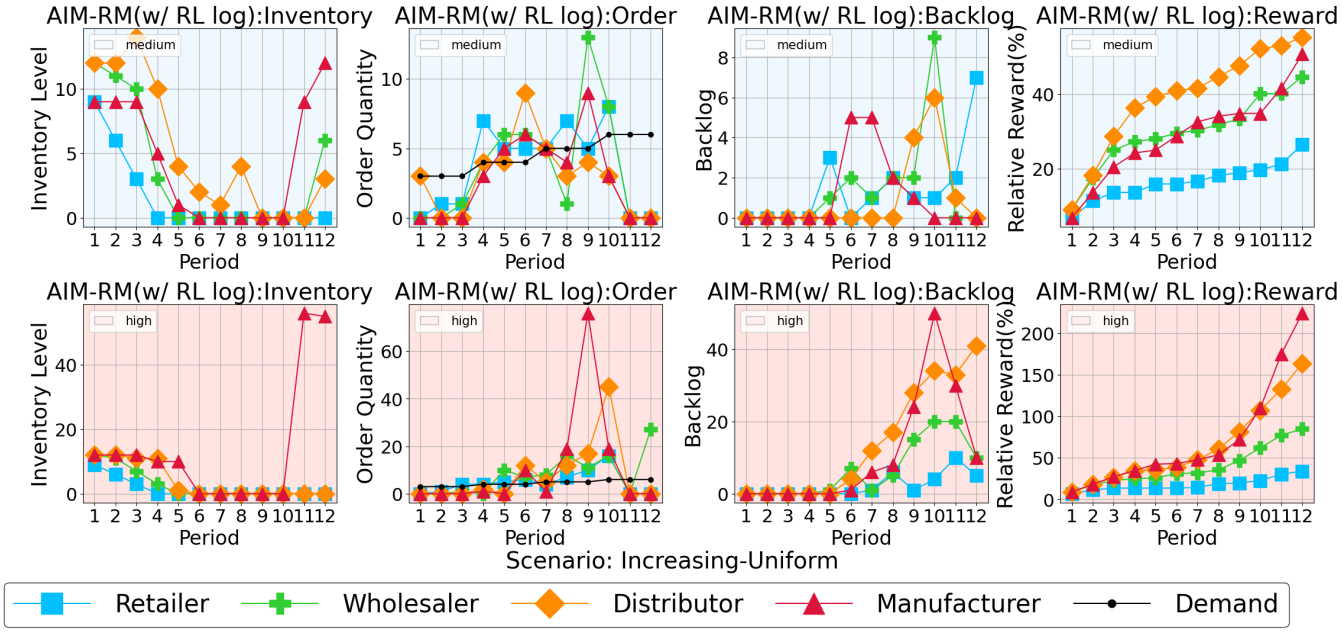

图4：一个回合内库存水平、积压订单、订货量和累积奖励的结果。第一行的图表示AIM-RM **（包含强化学习日志）在中等推理力度下的结果，第二行的图表示同一模型在高推理力度下的结果。**

我们发现未来研究有两个值得探索的方向。首先，可以通过整合 [41] 中提出的自我反思机制来增强 AIM-RM。该机制使大语言模型能够生成简短的反思，总结哪些做法奏效、哪些未能奏效及其原因，并为后续轮次制定可执行规则。尤其是，将以往的自我反思加入历史日志后，AIM-RM 可以在遇到相似案例时利用自我反思日志，从而提升性能。其次，我们可以扩展客户需求设定，将随机需求场景纳入其中。本研究采用确定性需求设定，是因为使用大语言模型进行足够数量的统计试验会产生较高的 API 成本。为避免这些高昂成本，我们现在可以使用本地高性能推理模型，例如 gpt-oss-120B[1], ，即使在随机条件下也能开展更广泛的实验。

致谢

作者感谢匿名审稿人提出的有益且富有建设性的意见。

参考文献

[1] Sandhini Agarwal、Lama Ahmad、Jason Ai、Sam Altman、Andy Applebaum、Edwin Arbus、Rahul K Arora、Yu Bai、Bowen Baker、Haiming Bao 等。2025。gpt-oss-120b 与 gpt-oss-20b 模型卡。arXiv 预印本 arXiv:2508.10925（2025）。[2] Anthropic。 [无日期]。Claude Opus 4.1。（[无日期]）。https://www.anthropic.com/claude/ opus [3] Robert N Boute、Joren Gijsbrechts、Willem Van Jaarsveld 和 Nathalie Van-vuchelen。2022。库存控制中的深度强化学习：路线图。《欧洲运筹学杂志》298，2（2022），401–412。[4] 可持续全球供应链研究网络。2023。地缘政治危机时期的可持续全球供应链：《2023年可持续全球供应链年度报告》。https://www.sustainablesupplychains.org/wp-content/uploads/2024/01/SustainableGlobalSupplyChains-Report2023.pdf[5] Yanda Chen、Joe Benton、Ansh Radhakrishnan、Jonathan Uesato、Carson Denison、John Schulman、Arushi Somani、Peter Hase、Misha Wagner、Fabien Roger 等。2025。推理模型并不总会说出自己的想法。arXiv 预印本 arXiv:2505.05410（2025）。[6] Gheorghe Comanici、Eric Bieber、Mike Schaekermann、Ice Pasupat、Noveen Sachdeva、Inderjit Dhillon、Marcel Blistein、Ori Ram、Dan Zhang、Evan Rosen 等。2025。Gemini 2.5：凭借先进推理、多模态、长上下文和新一代智能体能力，推动前沿发展。arXiv 预印本 arXiv:2507.06261（2025）。[7] Ton De Kok、Christopher Grob、Marco Laumanns、Stefan Minner、Jörg Rambau 和 Konrad Schade。2018。随机多级库存模型的类型学与文献综述。《欧洲运筹学杂志》269，3（2018），955–983。[8] Christian Schroeder De Witt、Tarun Gupta、Denys Makoviichuk、Viktor Makoviy-chuk、Philip HS Torr、Mingfei Sun 和 Shimon Whiteson。2020。在星际争霸多智能体挑战中，独立学习是否就足够了？arXiv 预印本 arXiv:2011.09533（2020）。[9] Ximing Dong、Shaowei Wang、Dayi Lin 和 Ahmed Hassan。2025。基于模型性能引导的评估数据选择，以实现有效的提示优化。载于《计算语言学协会：ACL 2025 论文集》。计算语言学协会。[10] Md Zahidur Rahman Farazi。2025。利用多智能体系统和机器学习增强供应链韧性：自适应决策框架。《美国工程与技术杂志》7，03（2025），6–20。[11] Kevin Geevers、Lotte Van Hezewijk 和 Martijn RK Mes。2024。使用深度强化学习进行多级库存优化。《中欧运筹学杂志》32，3（2024），653–683。[12] Soumya Suvra Ghosal、Souradip Chakraborty、Avinash Reddy、Yifu Lu、Mengdi Wang、Dinesh Manocha、Furong Huang、Mohammad Ghavamzadeh 和 Amrit Singh Bedi。2025。多思考总是有帮助吗？理解推理模型中的测试时扩展。arXiv 预印本 arXiv:2506.04210（2025）。[13] João NC Gonçalves、M Sameiro Carvalho 和 Paulo Cortez。2020。安全库存确定的运筹学模型与方法：综述。《运筹学视角》7（2020），100164。[14] Christian D. Hubbs、Hector D. Perez、Owais Sarwar、Nikolaos V. Sahinidis、Ignacio E. Grossmann 和 John M. Wassick。2020。OR-Gym：面向运筹学问题的强化学习库。arXiv:arXiv:2008 06319 。[15]

Valeria Jannelli、Stefan Schoepf、Matthias Bickel、Torbjørn Netland 和 Alexandra Brintrup。2024。供应链中的智能体式大语言模型：迈向自主多智能体共识寻求。arXiv 预印本 arXiv:2411.10184（2024）。[16] Fu Jia、Shiyuan Zhang、Xiao-Xue Zheng 和 Tsan-Ming Choi。2023。一种协调多智能体逆向供应链的新型机制：应对公平性问题。《国际生产经济学杂志》265（2023），108973。[17] Buggle Johannes、Cacault Paula 和 Danthine Jean-Pierre。2021。全球化的未来：构建有韧性且可持续的供应链。https://e4s. center/wp-content/uploads/2022/09/Supply_Chains_November2021.pdf [18] Toshiya Kaihara。2006。基于多智能体的供应链建模与动态环境。载于《基于多智能体的供应链管理》。Springer，203–215。[19] Samuel Kirshner、Yiwen Pan、Jason Xianghua Wu 和 Alex Gould。2024。洽谈条款：大语言模型供应链谈判中的智能体信息。Alex，《洽谈条款：大语言模型供应链谈判中的智能体信息》（2024年8月20日）（2024）。[20] Jundai Koketsu、Aya Ishigaki、Hiromasa Ijuin 和 Tetsuo Yamada。2022。基于多智能体系统的可持续供应链管理概念：采用线性物理规划进行谈判。载于《全球可持续制造大会》。Springer International Publishing Cham，119–126。[21] Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Heinrich Küttler、Mike Lewis、Wen-tau Yih、Tim Rock-täschel 等。2020。面向知识密集型自然语言处理任务的检索增强生成。《神经信息处理系统进展》33（2020），9459–9474。[22] Beibin Li、Konstantina Mellou、Bo Zhang、Jeevan Pathuri 和 Ishai Menache。2023。用于供应链优化的大语言模型。arXiv 预印本 arXiv:2307.03875（2023）。

[23] Yuan Li、Bingqiao Luo、Qian Wang、Nuo Chen、Xu Liu 和 Bingsheng He。2024。CryptoTrade：一种用于指导零样本加密货币交易的反思型基于大语言模型的智能体。载于《2024年自然语言处理实证方法会议论文集》。计算语言学协会，1094–1106。https://doi.org/10 18653 1 2024 63 . /v / .emnlp-main。[24] Guannan Liang 和 Qianqian Tong。2025。。大语言模型驱动的人工智能智能体系统及其在工业中的应用。arXiv 预印本 arXiv:2505.16120（2025）。[25] Pengfei Liu、Weizhe Yuan、Jinlan Fu、Zhengbao Jiang、Hiroaki Hayashi 和 Graham Neubig。2023。预训练、提示与预测：自然语言处理中提示方法的系统综述。《ACM计算调查》55卷第9期（2023），1–35。[26] Xiaotian Liu、Ming Hu、Yijie Peng 和 Yaodong Yang。2022。用于多级库存管理的多智能体深度强化学习。《生产与运营管理》（2022），10591478241305863。[27] Dhruv Madeka、Kari Torkkola、Carson Eisenach、Anna Luo、Dean P Foster 和 Sham M Kakade。2022。深度库存管理。arXiv 预印本 arXiv:2210.03137（2022）。[28] Marwan Mousa、Damien van de Berg、Niki Kotecha、Ehecatl Antonio del Rio Chanona 和 Max Mowbray。2024。去中心化库存控制系统中多智能体强化学习的分析。《计算机与化学工程》188（2024），108783。[29] Thierry Moyaux、Brahim Chaib-Draa 和 Sophie D'Amours。2003。基于令牌的多智能体协调：降低林业供应链中的牛鞭效应。载于《第二届自主智能体与多智能体系统国际联合会议论文集》，670–677。[30] Lilian Ngweta、Kiran Kate、Jason Tsay 和 Yara Rizk。2025。迈向对提示格式风格变化具有稳健性的大语言模型。载于《2025年美洲计算语言学协会分会会议：人类语言技术论文集》（第4卷：学生研究研讨会），529–537。[31] OpenAI。2025。GPT-5系统卡。https://cdn.openai.com/gpt-5-system-card.pdf [32] OpenAI。2025。OpenAI o3 和 o4-mini 系统卡。https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf [33] Sangmin Park 和 Hyunsoo Lee。2025。基于因果网络的多流深度学习的预测性供应链中断控制框架。（2025）。[34] Laurent Perron 和 Frédéric Didier。 [无日期]。CP-SAT。谷歌。https://developers.google.com/optimization/cp/cp_solver/ [35] Yinzhu Quan 和 Zefang Liu。2024。Econlogicqa：用于评估大语言模型经济序列推理能力的问答基准。arXiv 预印本 arXiv:2405.07938（2024）。[36] Yinzhu Quan 和 Zefang Liu。2024。Invagent：用于供应链库存管理的基于大语言模型的多智能体系统。arXiv 预印本 arXiv:2407.11384（2024）。[37] Abel Salinas 和 Fred Morstatter。2024。改变提示的蝴蝶效应：微小改动和越狱如何影响大语言模型的性能。载于《计算语言学协会 ACL 2024 论文集》，4629–4651。[38] Haralambos Sarimveis、Panagiotis Patrinos、Chris D Tarantilis 和 Chris T Kiranoudis。2008。供应链系统的动态建模与控制：综述。《计算机与运筹学》35卷第11期（2008），3530–3561。[39] John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。2017。近端策略优化算法。arXiv 预印本 arXiv:1707.06347（2017）。[40] Melanie Sclar、Yejin Choi、Yulia Tsvetkov 和 Alane Suhr。 [无日期]。量化语言模型对提示设计中虚假特征的敏感性，或：我是如何学会开始担忧提示格式的。载于《第十二届国际学习表征会议》。[41] Noah Shinn、Federico Cassano、Ashwin Gopinath、Karthik Narasimhan 和 Shunyu Yao。2023。Reflexion：通过语言进行强化学习的智能体。载于《第37届神经信息处理系统国际会议论文集》，8634–8652。[42] Haojie Wang、Jiuyun Jiang、L Jeff Hong 和 Guangxin Jiang。2025。用于供应链管理的大语言模型。arXiv 预印本 arXiv:2505.18597（2025）。[43] Chao Yu、Akash Velu、Eugene Vinitsky、Jiaxuan Gao、Yu Wang、Alexandre Bayen 和 Yi Wu。2022。PPO 在合作多智能体博弈中的惊人有效性。《神经信息处理系统进展》35（2022），24611–24624。[44] Wentao Zhang、Lingxuan Zhao、Haochong Xia、Shuo Sun、Jiaze Sun、Molei Qin、Xinyi Li、Yuqing Zhao、Yilei Zhao、Xinyu Cai 等。2024。用于金融交易的多模态基础智能体：工具增强、多样化且通用。载于《第30届 ACM SIGKDD 知识发现与数据挖掘会议论文集》，4314–4325。[45] Han Zhou、Xingchen Wan、Lev Proleev、Diana Mincu、Jilin Chen、Katherine A Heller 和 Subhrajit Roy。 [无日期]。批量校准：重新思考上下文学习和提示工程中的校准。载于《第十二届国际学习表征会议》。[46] Georg Ziegner、Michael Choi、Hung Mac Chan Le、Sahil Sakhuja 和 Arash Sarmadi。2025。迭代式多智能体强化学习：一种面向现实多级库存优化的新方法。arXiv 预印本

Konosuke Yoshizato、Kazuma Shimizu、Ryota Higa 和 Takanobu Otsuka

arXiv:2503.18201（2025）。

表 4：此表展示了实验场景中的最优奖励

模型	Const-Uni	Dec- Div	Dec- Uni Inc-Div		Inc-Uni
Opt	-120.00	332.00	-45.00	242.00	-132.00

A 数值实验细节

A.1 补充数值结果

最优值。所有层级的最优总收入是在约束条件（1）至（8）下通过最大化总收入得到的。只有存在集中式决策者时，求解该优化问题才可行，这比我们所考虑的分散式决策设定更为严格。因此，在分散式决策系统中实现最优策略极其困难。我们使用 Python 中 OR-Tools [34] 的 CP-SAT 求解器求解了该优化问题。各实验场景的最优值见表 4。

标准差。平均奖励及其标准差见表 5 和表 6。几乎所有实验的标准差均为 0。这些结果表明，大语言模型输出的随机行为对决策没有显著影响。

A.2 GPT-5 的结果

在正文中，我们使用 o4-mini 进行了实验。此外，我们还使用 GPT-5 分别以中等和高推理力度进行了相同的实验。表 7 和表 8分别展示了实验结果，结果表明，在表2、表3、表7和表8所呈现的所有结果中，采用中等推理力度的 o4-mini 配合 AIM-RM（含强化学习日志）取得了最佳平均表现。GPT-5 在中等和高推理力度下的平均表现与采用中等推理力度的 o4-mini 相当。这表明 GPT-5 的过度思考程度适中。InvAgent 的各个变体表现与 AIM-RM（含强化学习日志）相当。这些趋势表明，随着大语言模型不断进步，它们将能够独立做出复杂决策，无需复杂的提示调优和额外工具。

A.3 强化学习比较目标的超参数调优

我们采用与 [36] 中所述相同的方法确定超参数。我们使用表 9中列出的候选值探索超参数。从表中随机选取了30组超参数组合，包括隐藏层大小、学习率和批量大小，并选择得分最高的组合。各实验场景中 IPPO 和 MAPPO 所选的超参数分别列于表10和表11。使用各实验场景中选定的这些超参数，我们评估了奖励，如表2和表3所示。

B 提示示例

在数值实验中，我们使用了多种类型的提示，以下引用的图中分别展示了示例。图5展示了决策提示 P_{DM} 的示例，图6则展示了逐步描述提示 P_{SD}。图7给出了安全库存描述 P_{SS}，图8展示了记忆使用提示 P_{MU}。最后，图9展示了实验中使用的需求描述。此外，本节还补充了这些提示的详细信息。

决策提示。 此提示通过指出智能体的位置和当前状态来引导智能体，其中包括提前期、库存水平和当前欠交量。利用这些输入，智能体实际上通过此提示来决定动作。然而，此提示不包含有关问题设定或库存管理策略的任何信息。

逐步提示。 此提示概述了一个周期内执行的流程，如第3.2节所述，其中包括提前期计算示例。此流程对于计算所需库存水平至关重要，并能进一步帮助智能体结合提示精确选择动作。尤其是在智能体发出的订单到货或库存被消耗时，此流程尤为重要。

安全库存提示。 此提示向智能体传授一种标准的库存管理策略。该策略首先考虑提前期，计算库存位置，也就是提前期之后的未来库存。随后，策略计算目标库存水平，即预计在提前期结束前消耗的数量。订货量可根据前述两个数量之间的差额计算。具体而言，该策略明确指出了计算总到货需求所需的周期数，从而使智能体能够在 Const-Uni 设定下恰当地决定订货量。

记忆使用提示。此提示定义了“相似案例”一词，其中包含状态向量、动作、奖励和距离（相似度度量）。它还向智能体说明相似案例的含义及其定位方式。将这些相似案例代入此提示中，即可作为输入提供给智能体。

需求描述。需求描述概述了每个周期出现的客户需求。本研究假设需求是确定性的，即所描述的需求必定会出现。

表 5：本表展示了各模型在不同回合中的平均总奖励，括号内为标准差。描述风格与表2相同。省略了启发式和强化学习基准方法。

模型（推理力度：中）	Const-Uni	Dec- Div	Dec- Uni	Inc-Div	Inc-Uni	平均值
InvAgent（含步骤描述）	3.33(0.00)	79.22(0.00)	437.78(0.00)	104.55(0.00)	137.88(0.00)	152.55
InvAgent（含步骤描述和 SS策略）	**0.00(0.00)**	115.06(0.00)	315.56(0.00)	264.88(0.00)	425.00(0.00)	224.10
AIM-RM（不含强化学习_log）	69.33(18.31)	66.27(2.32)	219.11(13.38)	162.64(49.32)	176.36(57.38)	138.74
AIM-RM（搭配强化学习_日志）	60.00(11.74)	**56.02(0.00)**	**171.11(0.00)**	**95.12(31.03)**	**74.09(3.90)**	**91.27**
InvAgent（GPT-4.1，无策略）	120.00(0.00)	244.58(0.00)	991.11(0.00)	474.79(0.00)	262.88(0.00)	418.67
InvAgent（GPT-4.1，带策略）	76.67(0.00)	88.25(0.00)	428.89(0.00)	133.06(0.00)	181.81(0.00)	181.74
InvAgent（无策略）	46.67(0.00)	100.30(0.00)	202.22(0.00)	171.90(0.00)	93.18(0.00)	122.85
InvAgent（含策略）	183.33(0.00)	118.67(0.00)	200.00(0.00)	179.33(0.00)	155.30(0.00)	167.33

表 6：模型总奖励的结果。括号内的数字表示标准差。描述风格与表2中的相同。省略了启发式和强化学习基准。

模型（推理力度：高）	Const-Uni	Dec- Div	Dec- Uni	Inc-Div	Inc-Uni	平均值
InvAgent（含步骤描述）	3.33(0.00)	**48.80(0.00)**	206.67(0.00)	147.93(0.00)	265.90(0.00)	134.53
InvAgent（含步骤描述和SS策略）	**0.00(0.00)**	129.22 (0.00)	342.22 (0.00)	344.63 (0.00)	437.88 (0.00)	250.79
AIM-RM（不含强化学习日志）	36.67(5.58)	80.48(14.94)	185.33(6.22)	130.90(30.68)	384.55(18.63)	163.59
AIM-RM（含强化学习日志）	19.33(1.33)	50.90(0.00)	188.89(0.00)	**104.13(0.00)**	390.45(8.18)	**150.74**
InvAgent（无策略）	46.67(0.00)	66.87(0.00)	**177.78(0.00)**	412.81(0.00)	800.00(0.00)	300.83
InvAgent（含策略）	190.00(0.00)	68.07(0.00)	666.67(0.00)	127.27(0.00)	**151.51(0.00)**	240.70

表7：本表展示了GPT-5在各回合中使用每个模型时的平均总奖励。奖励表示为相对差距 $\Delta = |\mathrm{Opt} - r|/\mathrm{Opt}$，其中Opt 是最优值，$r$是总奖励。第一行中的*const-uni*代表常数均匀设置。第一行中的其他名称也是设置名称的缩写。

模型(推理力度:中等)	Const-Uni	Dec- Div	Dec- Uni	Inc-Div	Inc-Uni	平均值
InvAgent（含步骤描述）	23.33(0.00)	65.06(0.00)	202.22(0.00)	119.01(0.00)	170.68(0.00)	116.06
InvAgent（含步骤描述以及我们的策略）	**0.00(0.00)**	60.84 (0.00)	282.22 (0.00)	376.45 (0.00)	356.39 (0.00)	195.18
AIM-RM（不含强化学习日志）	17.33(1.33)	55.18(3.80)	175.11(8.00)	147.11(11.57)	129.55(5.25)	104.86
AIM-RM（含强化学习日志）	13.33(0.00)	51.81(0.00)	226.67(0.00)	118.60(0.00)	112.03(0.00)	104.49
InvAgent（无策略）	46.67(0.00)	**31.33(0.00)**	**173.33(0.00)**	**102.48(0.00)**	115.04(0.00)	**93.77**
InvAgent（含策略）	133.33(0.00)	131.33(0.00)	340.00(0.00)	108.68(0.00)	**81.20(0.00)**	158.91

表 8：使用 GPT-5 并采用高推理力度时的总奖励结果。描述方式与表2相同。省略了启发式方法和强化学习基准。

模型（推理力度：高）	Const-Uni	Dec- Div	Dec-Uni	Inc-Div	Inc-Uni	平均值
InvAgent（含步骤_描述）	76.67(0.00)	**40.06(0.00)**	**169.33(18.51)**	**95.04(0.00)**	191.67(0.00)	114.55
InvAgent（含步骤_说明以及我们的_策略）	**0.00(0.00)**	73.80(0.00)	202.02(0.00)	264.46(0.00)	444.70(0.00)	197.04
AIM-RM（不含强化学习_日志）	23.33(0.00)	42.11(1.69)	175.11(0.89)	114.38(5.62)	158.64(19.09)	102.71
AIM-RM（含强化学习_日志）	28.67(2.67)	44.88(0.00)	195.56(0.00)	102.07(0.00)	134.24(2.46)	101.08
InvAgent（无策略）	73.33(0.00)	51.20(0.00)	173.33(0.00)	111.98(0.00)	71.97(0.00)	**96.36**
InvAgent（含策略）	103.33(0.00)	93.37(0.00)	388.89(0.00)	170.25(0.00)	**64.39(0.00)**	164.05

表9：强化学习基准方法中的超参数候选值。

超参数	候选值
隐藏层数量	[[128, 128], [256, 256]]
激活函数	[ReLU]
学习率	[1e-4, 5e-4, 1e-3]
学习批次大小	[500, 1000, 2000]
SGD 小批量大小	[32, 64, 128]
随机梯度下降迭代次数	[5, 10, 20]
学习迭代次数	[500, 800, 1000, 1500]
γ	[1.0]

表1 10：每个场景中IPPO的超参数。

超参数	**Const-Uni**	**Dec-Div**	**Dec-Uni**	**Inc-Div**	**Inc-Uni**
隐藏层数量	[128, 128]	[128, 128]	[256, 256]	[128, 128]	[256, 256]
激活函数	ReLU	ReLU	ReLU	ReLU	ReLU
学习率	1e-4	1e-4,	1e-4	1e-4	1e-4
训练批次大小	1000	1000	1000	500	1000
SGD 小批量大小	128	128	128	32	128
SGD 试验次数	5	5	5	10	5
学习试验次数	1000	1000	1500	800	1500
γ	1.0	1.0	1.0	1.0	1.0

表11：MAPPO在各场景中的超参数。

超参数	**Const-Uni**	**Dec-Div**	**Dec-Uni**	**Inc-Div**	**Inc-Uni**
隐藏层数量	[256, 256]	[128, 128]	[128, 128]	[128, 128]	[256, 256]
激活函数	ReLU	ReLU	ReLU	ReLU	ReLU
学习率	1e-4	1e-4	1e-4	1e-4	1e-4
训练批次大小	1000	1000	1000	500	1000
SGD 小批量大小	128	64	32	128	64
SGD 试验次数	5	5	5	10	5
学习试验次数	1500	1000	1000	500	1000
γ	1.0	1.0	1.0	1.0	1.0

决策提示:
现在是第 round 轮，你处于供应链的第 {num_stages} 阶段。当前状态如下：- 提前期：lead_time 轮 - 库存水平：inventory 单位 - 当前欠交量（你欠下游的）：backlog 单位 - 上游积压订单（上游欠你的）：upstrea m_backlog 单位 - 先前销售额（最近 round 轮，从旧到新）：sales - 即将到货的交付（本轮及接下来的 round 轮，从近到远）：{deliveries}" {demand_description} 本轮来自第 {stage_idx} 阶段的下游订单为 {Action(stage_id-1)}。

图5：此提示向决策模块提供输入数据，包括当前状态和输出格式。

步骤描述:
##
#################
InventoryManagementEnv 中的单周期流程（按实际实现）###
##########################
1. 一个周期包含以下四个步骤：
a) 接收交付：接收先前订购的交付量，并更新库存水平。
b) 订货决策：决定向上游供应商下达的订货量。
c) 发货：将本周期订购的商品发给下游供应商。
d) 计算利润：根据销售、采购和成本计算利润。
提前期示例（L_m = 2） self.period：1 2 3 4 . . .
查看库存：I0 I1 I2* I3 . . .
决定订单：O1 O2 O3 .. .
↪ 发货 R1
↪ 到达 →后加入 I3
*在 t = 3 时做出决策期间，t = 1 时下达的批次尚未到达。

图6：$\mathcal{P}_{\mathrm{SD}}$ 向大语言模型讲授多级库存管理流程 t.

安全库存说明:
决策策略（提前期安全订货至目标策略）——严格遵循 1) 考虑提前期，计算你的库存位置（IP）：IP = 库存 + 未来 lead_time 个周期内的交付量总和 - 当前 _ 积压订单。2) 采用定期审查、先订货后发货的订货至目标策略。将基础目标设为：target_base ≈ (lead_time + 1) × μˆ(mu_hat)，并可选择增加安全库存：z × σˆ(sigma_hat) × √(lead_time + 1)。3) 根据 target_base 与当前 IP 之间的差额确定订货量。但请确保订货量不超过生产能力（prod_capacity）。4) 用 1–2 句话简要说明你如何得出订货量。

图7：此提示描述了制造业中的标准安全库存策略。

记忆使用说明:
你还会获得先前回合的检索结果，称为‘similar_cases‘。定义——‘similar_cases‘是指先前回合中数值状态与当前状态最相似的历史决策。- 状态向量字段（顺序必须完全如下）： [库存、积压订单、上游_积压订单、提前期、交付量] - 每个案例包含：‘state_vec‘、‘动作‘、‘奖励‘和‘距离‘（相似度度量）。- 距离值越低，表示与当前状况越相似。- 将其视为证据，而非规则。similar_cases：{similar_cases}

图8：此提示指示决策模块如何使用相似案例。

需求描述:
递增需求：零售商（阶段0）的每轮预期需求为 $(2+\lfloor (t-1)/3 \rfloor)$，其中轮次 $t = 1, \ldots, 12$。
相应地，需求每三轮递增一次：第1–3轮为2单位，第4–6轮为3单位，第7–9轮为4单位，第10–12轮为5单位。

递减需求：零售商（阶段0）的每轮预期需求为 $(2+\lceil (12-(t-1))/3 \rceil)$，其中轮次 $t = 1, \ldots, 12$。
相应地，需求每三轮递减一次：第1–3轮为6单位，第4–6轮为5单位，第7–9轮为4单位，第10–12轮为3单位。

图9：此提示描述表明，客户需求在所有轮次均会出现。