从假设到动作：将大语言模型推理转化为不确定性感知规划，赋能具身智能体

SeungWon Seo*、**SooBin Lim***、**Seongrae Noh***、**Haneul Kim**、**HyeongYeop Kang**†计算机科学与工程系，高丽大学{ssw03270,dnpcs,rhosunr99,adsky0309,siamizhkang}@korea.ac.kr -

arXiv:2602.04326v1 [cs.AI] 4 Feb 2026

摘要

在多智能体、部分可观测且去中心化的环境中运行的具身智能体，必须在对隐藏物体和协作者意图普遍存在不确定性的情况下进行规划与行动。近期将大语言模型（LLM）应用于具身智能体的研究进展，已经解决了许多长期存在的挑战，例如高层目标分解和在线适应。然而，不确定性目前仍主要通过频繁的智能体间通信来缓解。当涉及人类协作者时，这不仅会带来高昂的令牌和时间成本，还可能打乱既有的工作流程。我们提出 PCE，即规划器-组合器-评估器框架，将大语言模型推理轨迹中潜在的碎片化假设转换为结构化决策树。内部节点编码环境假设，叶节点映射到动作；随后，根据情景可能性、目标导向收益和执行成本对每条路径进行评分，从而在无需大量通信的情况下指导理性动作选择。在两个具有挑战性的多智能体基准测试（C-WAH 和 TDW-MAT）以及三种多样化的大语言模型骨干网络上，PCE 的成功率和任务效率始终优于以通信为中心的基线，同时令牌使用量相当。消融实验结果表明，即使应用 PCE，通过扩展模型容量或增加推理深度所获得的性能收益依然存在；与此同时，PCE 在不同模型容量和推理深度规模下始终提升基线，证实结构化不确定性处理能够与这两种规模扩展方式互补。进一步的用户研究表明，PCE 生成的通信模式被人类协作者认为更加高效且值得信任。综上，这些结果为将大语言模型潜在的假设转化为面向不确定性感知规划的可靠策略，确立了一条有原则的路径。

1 引言

设想两名家用智能体协作准备一餐：每个智能体只能感知厨房的一部分，但两者都必须协调行动，将食材收集起来并送到共享工作区。此类智能体是具身智能体的典型代表，它们感知周围环境、制定计划并执行动作，以在动态环境中实现指定目标（Fung 等人，2025）。当多个具身智能体必须在部分可观测的分散式环境中协作时，每个智能体有限的感知范围都会导致对未观测物体以及协作者意图和动作的普遍不确定性（Spaan 等人，2006；Amato 等人，2016；2015）。在每个规划步骤都对这些不确定性进行朴素探索，计算成本高得难以承受。此外，这种做法往往会产生次优或不一致的计划，因为可能隐藏状态的组合增长显著增加了评估可用动作相对价值和维持连贯信念的难度。

由大语言模型（LLM）驱动的智能体在此类场景中已取得显著进展。借助少样本和零样本推理，基于大语言模型的规划器能够将高层目标分解为可执行动作序列，根据环境变化调整计划，并可靠地推断协作者的意图，从而在复杂的长时域场景中实现稳健表现。然而，为了应对不确定性，大多数现有系统依赖与协作者反复进行自然语言

*本文作者贡献相同。
†通讯作者。

通信，以验证计划、交换信息并迭代完善联合策略。这种以通信为中心的范式代价高昂：频繁对话会消耗大量词元和时间；而当协作者是人类时，持续的提问和汇报可能会打乱既有工作流程。此外，单纯提升大语言模型能力或加深其推理链，并不能从根本上消除不确定性。若没有明确机制来识别和评估部分可观测环境中导致不确定性的假设，更大的模型仍难以权衡关于环境的各种相互竞争的假设，进而导致优先级错位、规划质量下降。

为解决这些局限，我们利用了两项关键的实证观察：第一，当被要求评估或选择动作时，大语言模型规划器会在零样本思维链推理轨迹中，针对环境中不确定的方面在内部生成隐含假设。例如，它们往往会假设存在一个不可见的物体，或推断协作者可能采取的动作。这些假设反过来可以用于识别缺失的信息。第二，我们发现，这类假设是在局部被调用并隐式引用的，而不会被明确汇总以支持全局决策。这种缺乏结构的处理方式使规划器无法系统地协调多个假设，因而难以检测逻辑冲突、一致地比较期望增益、权衡备选动作，并最终提升规划准确性。
因此，我们提出规划器-组合器-评估器（PCE），将这些内部生成的假设提取并汇总为一种以决策树形式呈现的统一结构化表示，从而能够对它们进行联合评估，以实现更可靠且具备不确定性感知能力的动作选择。在这棵树中，每个内部节点都指定一个关于环境的假设，每个叶节点则指定在沿路径累积的假设下被认为合适的动作。因此，每条从根节点到叶节点的路径都代表一种特定的假设组合，并最终导向一条动作轨迹。随后，我们引入一个评估器，根据每条路径的可能性、收益和成本为其评分，从而在无需大量通信的情况下指导理性动作选择。这一设计将大语言模型推理已生成的潜在知识转化为一种有原则的不确定性感知规划机制。

与以往主要通过迭代通信或联合计划协商来优化协作的多智能体规划器（Seo 等，2025；Liu 等，2025；Zu 等，2025）不同，PCE 作用于决策问题的另一个层面。PCE 不优化动作序列或通信策略，而是明确将环境假设视为首要决策变量，并在执行动作之前对其进行推理。这使规划范式从以通信为中心的协作，转向对嵌入智能体自身信念状态中的不确定性进行结构化推理。

在两个具有挑战性的多智能体基准测试 C-WAH（Zhang 等，2024b）和 TDW-MAT（Zhang 等，2024b）上的实验表明，PCE 在成功率、任务效率和令牌使用量方面始终优于以通信为中心的基线。消融研究显示，如果没有明确的不确定性处理，仅提升大语言模型能力或推理深度，相较于其计算成本只能带来有限的性能提升。用户研究进一步表明，PCE 提升了人类与智能体协作的可靠性。由于所提出的框架基于通用推理轨迹，而非特定模型的内部机制，因此可以轻松应用于各种大语言模型骨干网络和多智能体领域。我们通过在多种骨干模型上运行 PCE 验证了其通用性，包括 GPT-4o mini、GPT-OSS:20B 和 Gemma3:4B，并观察到它们均取得了稳定的改进。总之，这些结果将研究重点从依赖大量通信的缓解措施转向对大语言模型推理中已有假设的原则性利用，为具身系统中的不确定性感知规划开辟了新方向。

2 相关工作

具身多智能体协作。 具身智能体旨在充满多种物体的复杂环境中实现长期目标（Song 等，2023；Li 等，2024）。近期研究利用大语言模型作为规划器，将高层目标分解为用于导航和操作任务的可执行动作序列（Zhou 等，2024；Huang 等，2022；Song 等，2023；Wang 等，2023a;c），并通过零样本和少样本推理以及动态重规划来提升规划效果（Huang 等，2022；Song 等，2023）。除了单智能体设定之外，研究人员还针对超出单个智能体能力范围的任务，研究了多智能体环境中的协作规划（Zhang 等，2024c；Guo 等，2024）。

为了研究更现实的具身多智能体协作，研究人员关注这样的环境：分布式智能体必须基于各自的有限观测做出决策（Zhang 等，2024b；Seo 等，2025；Liu 等，2025；Zu 等，2025）。在这些设定中，智能体无法实时获取超出自身观测范围的信息。因此，它们必须通过物理移动来收集信息，或通过与协作者通信来共享信息，以减少移动时间（Oliehoek & Spaan，2012；Spaan 等人，2006；Foerster 等，2016）。

以往的工作通过密集通信来缓解不确定性。例如，ProAgent（Zhang等人，2024a年）会显式推理协作者的动作，并通过对话验证其推理；CoELA（Zhang等人，2024b年）和 REVECA（Seo 等人，2025）会交换状态和计划信息，以验证并调整策略；RoCo（Mandi 等人，2024年）和 CaPo（Liu 等，2025）引入了迭代式智能体间辩论；CoTS（Zu 等，2025）则利用基于通信的反馈来完善联合计划。尽管这些方法提升了协作效果，但它们会产生高昂的 token 和时间成本，而且在有人类参与时可能会打乱工作流程。

最近，LLaMAR（Nayak 等人，2024年）提出了一种长时域的基于大语言模型的规划框架，可在部分可观测环境中运行，而无需依赖繁重的通信；然而，其设定是在集中式多智能体环境中处理协作问题。

我们的工作采取了不同的路径：我们不依赖繁重的通信，而是提取并汇总大语言模型推理中的隐含假设，将其纳入一个决策树框架中进行联合评估，以实现理性动作选择。这使智能体能够自适应地平衡物理动作与通信动作，在降低通信开销的同时保持或提升规划性能。

面向具身智能体的大语言模型规模扩展。大语言模型的最新进展已使具身智能体能够进行自然语言推理和高层决策。扩展模型容量能够提升性能（Kaplan 等人，2020 年），但需要投入大量训练时间和数据资源。为弥补这些成本，研究者提出在训练阶段进行强化学习微调（Shao 等人，2024年；Wang 等人，2024年；Xu 等人，2025）；与此同时，思维链（Wei 等人，2022）、思维树（Yao 等人，2023）和自一致性（Wang 等人，2023b）等推理增强方法，则被引入推理时（Snell 等人，2025）。

然而，仅仅提升模型容量或推理深度，并不能解决模型根本上依赖存储在模型参数中的知识这一问题（Mirzadeh 等人，2024；Huang 等人，2023；Yin 等人，2024）。在需要外部知识的任务中，这一局限尤为明显（Lewis 等人，2020）：大语言模型会生成带有不确定性的输出，且质量有所下降（Li 等人，2025）；而在部分可观测的去中心化环境中，智能体的内部记忆很快就会过时，从而加剧不确定性并限制规划性能（Spaan 等人，2006；Amato 等人，2016；2015）。

据我们所知，此前尚无研究系统地探讨具身规划中的不确定性是否能仅通过扩展大语言模型来解决。我们的实验表明，除提升模型容量或推理深度所带来的收益外，PCE 在所有测试的骨干模型上均带来稳健的改进，表明其收益不仅可与规模扩展带来的收益叠加，而且与之有所不同。

树结构推理与搜索。基于树的框架已被广泛用于提升决策质量，但无论是在树所表示的内容，还是对通信的处理方式上，PCE 在概念上都不同于思维树（ToT）（Yao等人，2023）和 CoTS（Zu 等，2025）等现有方法。ToT 围绕推理步骤构建树以增强逻辑连贯性，但它主要运行于单个智能体的内部推理空间中。它隐含地假设环境完全可观测，并将树节点视为认知步骤，而非概率性的环境状态，因此限制了其在动态、部分可观测多智能体场景中的适用性。在多智能体设置中，CoTS 在联合推理与行动空间中执行树搜索，并通过迭代通信来探索和剪枝候选规划。这种以通信为中心的表述方式使对话成为搜索的先决条件，从而增加了延迟和令牌消耗，尤其是在不完美可观测的情况下。

相比之下，PCE 根据有关环境的不确定假设来构建决策树，而不只是依据推理步骤。关键在于，PCE 并不将通信视为搜索

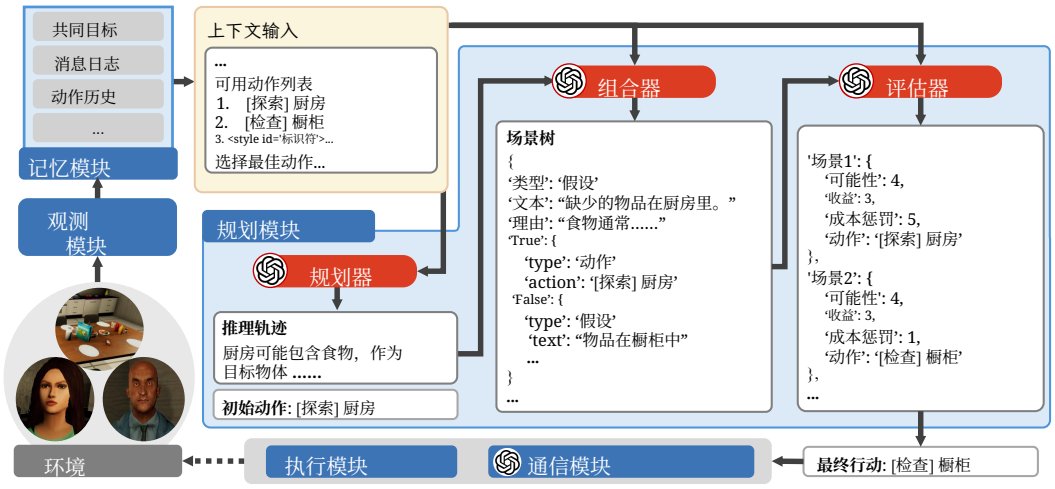

图1：PCE 采用由规划器、组合器和评估器构成的模块化规划流水线。

机制本身，而是将其视为搜索空间中的原子动作，并与物理动作进行评估比较。这样，只有当通信带来的期望效用高于独自行动时，智能体才会选择通信。

3 问题定义

我们将高成本通信条件下的协作具身智能体建模为去中心化部分可观测马尔可夫决策过程（DEC-POMDP）（Bernstein 等人，2002；Amato 等人，2013）。环境在有限时域 $t = 1$ 内通过状态 $s, \ldots, H$ 演化。在每个时间步，n智能体以转移概率 $P(s' \mid s, a^1, \ldots, a^n)$ 执行动作 $(a^1, \ldots, a^n)$。

每个智能体 i 接收一个观测 $o_t^i \in \mathcal{O}^i$，其中 $\mathcal{O}^i = \mathcal{O}^i_{\text{env}} \cup \mathcal{O}^i_{\text{com}}$ 结合了直接环境观测与基于通信的观测。观测历史按如下方式演变：$h_t^i = (h_{t-1}^i, o_t^i)$。

动作空间 $A^i = A^i_{phy} \cup A^i_{com}$ 包含物理动作和通信动作。本地策略 π^i 将当前历史映射为一个动作，$a_t^i = \pi^i(h_t^i)$。通信动作会发送根据 h_t^i 构造的消息 m_t^i，并广播给所有其他智能体 $j \neq i$。我们假设通信具有成本：每次 $a_t^i \in A_{com^i}$ 都会占用原本可用于 A^i_{phy}的时间，并且消息会延迟一步到达，即 m_t^i 会出现在 o_{t+1}^i中。

Our 目标是找到一组去中心化策略 $\pi = (\pi^1, \ldots, \pi^n)$，使智能体能够
在时域 H内共同实现由多个子目标组成的共同目标 G。

4 方法

PCE（见图1）采用具身智能体协作中常用的模块化设计（Zhang 等，2024b；Seo 等，2025；Liu 等，2025；Zu 等，2025）。它由观察、记忆、规划、通信和执行模块组成。其中，我们将规划模块重新设计为规划器–组合器–评估器流程，明确纳入不确定性处理，从而实现更可靠的动作选择。规划器、组合器和评估器的详细提示策略见附录A.12。

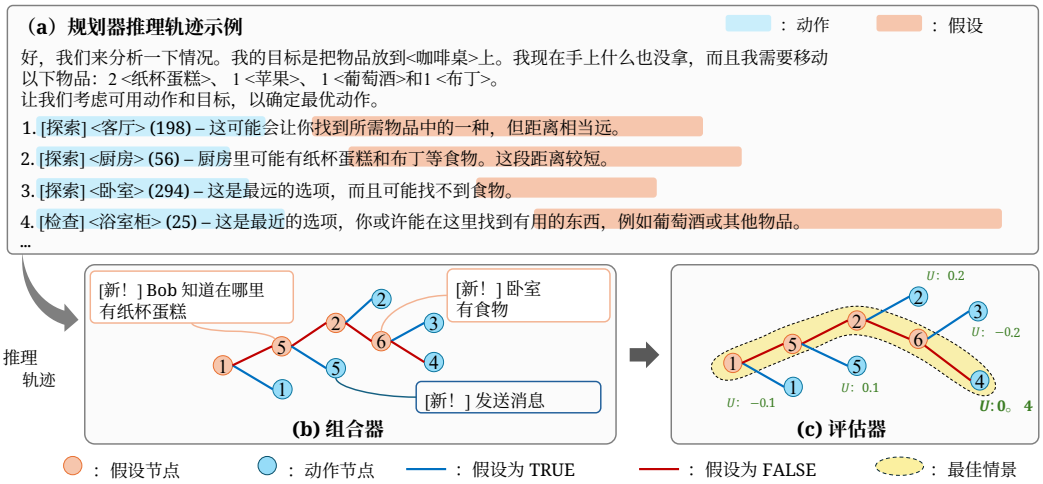

图 2：从推理轨迹到动作选择的流程。(a) 规划器生成推理轨迹。(b) 组合器从轨迹中提取假设，将其构建为决策树，并在需要时生成新的假设和通信动作，以拓展尚未探索的分支。(c) 评估器为每条路径评分；高亮路径表示其叶节点取得最高分的场景（U），由此确定智能体最终选择的动作。

4.1 观测和记忆模块

观察模块将原始环境信号和协作者信号转换为结构化的感知信息。每个智能体都会检测其感知范围内物体的名称/ID、位置、房间关联、可用交互以及物体状态，同时获取协作者手持的物体及其位置。封闭容器会遮挡其中的内容；例如，在执行打开等物理动作之前，橱柜内部始终未知，打开后，新显现的物体（如纸杯蛋糕）才会变得可观测。观测还包括协作者在上一步发送的消息。

记忆模块充当统一的信息库，整合静态信息与动态更新的数据。初始化时，记忆模块会填入共同目标 G 和低层动作技能库（用于执行动作的一组 API）。执行过程中，记忆会逐步更新推断出的任务进度、协作者信息、消息日志以及智能体自身的先前操作。为避免长时间模拟中内容无限增长，消息日志会截断为最近的 $K_{message}$ 条记录，动作历史则截断为最近的 K_{action} 条记录。详情请参阅附录 A.3。

4.2 规划器：面向动作选择的推理

规划器构成我们规划模块的第一阶段。它接收作为上下文输入 的共同目标 G、当前进度、消息日志、来自记忆的近期动作以及可用动作列表。据此，它推理每个可用动作为何有助于实现 G，并相应产生初始的动作选择。例如，如果智能体已经到达厨房并观察到一个橱柜，但尚未与其交互，规划器可能会提出诸如‘[检查]<厨房橱柜> (78)’这样的动作。借助大语言模型的推理能力，规划器不仅输出候选动作，还输出相应的推理轨迹。

从经验上看，这条轨迹中的每个候选动作往往都只基于对不确定环境的一个局部假设。例如，如图 2-(a) 所示，可用动作‘[gocheck] <bathroomcabinet>(25)’仅基于“这里或许能找到有用的东西”这一假设，而与其他备选动作相关的假设则仍未明说。因此，这些轨迹只揭示了彼此孤立的假设-动作关联，却未展示不同假设之间的关系，从而限制了规划器在不确定性下对动作进行排序的能力。

4.3 组合器：从推理轨迹到场景树

组合器会构建一棵显式的决策树。它首先对上下文输入和规划器的推理轨迹进行语义解读，以明确识别关键的不确定性（假设）。内部节点表示这些假设，并进行真/假划分，其分支条件沿路径继承。因此，从根节点到叶节点的路径定义了一个场景，即由一组假设构成并最终到达叶节点的组合。具体而言，每个叶节点都会分配一个旨在处理给定路径所涉及不确定性的动作——要么是用于直接交互的物理动作，要么是用于共享信息或向协作者发出指令的通信动作。

组合器并不枚举所有假设与动作之间的关系，而是自顶向下扩展树。在每个节点，局部排序策略会选择下一个要进行分支划分的假设，优先考虑最能降低不确定性、且对后续动作选择影响最大的假设。组合器不会计算真实概率，因为那相当于求解一个难以处理的部分可观测马尔可夫决策过程（POMDP）；而是利用大语言模型的常识推理来近似这些标准。它只考虑路径上尚未赋值且与当前前提一致的假设，从而避免生成不兼容的场景。

如果没有合适的假设可供选择，组合器会基于上下文中已有的实体（G、消息日志等）提出新的原子假设。这些候选项会按照相同的策略进行排序，若被选中，则会作为新的分支插入。扩展深度限制为 D，或者在进一步划分不会显著影响动作选择时提前停止；此时，生成的节点成为叶节点，并被分配最合适的动作。多个叶节点可能对应同一动作，体现该动作在不同场景下的适用性；而某些初始动作若始终不是最优选项，则可能被舍弃。
图2-(b)展示了这一过程。给定寻找食物的目标，根假设“客厅中包含它们”使True分支导向“[探索] <客厅> (198)”。在False分支上，前提发生变化，组合器生成一个新假设，即协作者（Bob）可能知道纸杯蛋糕的位置，从而将计划从物理探索转向信息收集动作，例如 [发送消息]。

4.4 评估器：情景可能性、条件增益与执行成本 T

评估器接收上下文输入 和决策树。随后，它对每个叶节点评分，以在不确定性下指导动作选择。对于每条从根到叶的路径，它估计：1）该场景 $\mathcal{S}$ 的可能性有多大；2）若该场景成立，该动作会在多大程度上推进目标；3）尝试该动作的代价有多高。所有分数均归一化到 $[0,1]$，以便比较。

情景可能性（$\mathcal{L}$）。$\mathcal{L}(\mathcal{S})$ 是指大语言模型根据智能体的观测和消息历史，对叶节点场景前提为真的概率所作的估计。例如，如果上次看到协作者时其身处厨房，那么“协作者仍在厨房”这一场景的得分会高于“厨房里空无一人”这一场景。

条件收益（$\mathcal{G}$）。$\mathcal{G}(a)$ 衡量在场景为真的条件下，执行动作 a 对实现目标能有多大推进作用。该值也由大语言模型估算。
利用 $\mathcal{L}$和 $\mathcal{G}$，我们将期望增益定义为

$$\mathbb{E}[\text{gain}] = \mathcal{L}(\mathcal{S}) \cdot \mathcal{G}(a),$$

其中，当场景为假时，$\mathcal{G}(a) = 0$。

执行成本（C）。$C(a)$ 量化尝试执行 a 所需的即时资源，无论其是否成功。我们使用示性函数将成本分解为移动和通信两部分：

$$C(a) \;=\; \alpha\, d(a)\, \mathbf{1}\{\text{move}(a)\} \;+\; \beta\, \ell(a)\, \mathbf{1}\{\text{comm}(a)\},$$

其中，$d(a)$ 是估计的遍历距离，$\ell(a)$ 是估计的消息长度，$\alpha,\ \beta > 0$是用于平衡移动和通信成本的缩放常数，$\mathbf{1}\{\cdot\}$是指示符。此设计体现了移动与通信互斥的特性：$\mathbf{1}\{\text{move}(a)\} + \mathbf{1}\{\text{comm}(a)\} = 1$。

最终得分（U）。对于每个叶节点动作，评估器都会计算一个最终得分：

$$U(\mathcal{S}, a) \;=\; \mathbb{E}[\text{gain}] \;-\; \lambda\, C(a),$$

其中 $\lambda > 0$ 控制成本敏感度。该评分综合考虑情景概率、目标导向效能和执行开销。按照 $U(\mathcal{S},\ a)$ 对叶节点进行排序，可在不确定性下做出合理的动作选择，如图 2-(c) 所示。实验部分将讨论 $\alpha,\ \beta,$ 和 λ 的实证设置。

4.5 通信与执行模块

这些模块将规划模块的决策转化为具体行为。当选择通信动作时，通信模块会向协作者发送消息或回复其消息。当选择物理动作（例如移动、抓取物体、搬运目标物体）时，执行模块会通过 A* 搜索计算路径，并调用动作技能库中相应的底层 Python API 来执行该动作。

5 实验

基准测试。在分散式控制的部分可观测性约束下，我们针对由多个房间组成的两项多目标家庭任务开展实验：C-WAH（Zhang 等，2024b年）和 TDW-MAT（Zhang 等，2024b 年）。C-WAH 包含 10 个回合，智能体需要在每个回合中完成 3–5 个子目标，时域 H 设为 250 步。TDW-MAT 包含 24 个回合，智能体的任务是运输 10 个目标物体以实现目标，时域 H 设为 3000 步。完整的环境详情见附录 A.1。

指标。 在 C-WAH 中，我们通过测量总步数 来评估智能体实现目标的速度。在 TDW-MAT 中，我们测量智能体运送出目标物体的比例（相对于目标物体总数），并分别统计食物、杂物及其平均值总计。

我们还使用使用次数 和通信来评估系统效率。使用次数 表示整个系统产生的令牌总消耗量。它不仅包括通信令牌，还明确涵盖框架内大语言模型模块产生的所有内部令牌，可作为总计算成本的替代指标。通信 衡量通信动作的数量。与其他指标不同，通信并没有内在的“越低越好”或“越高越好”之分。通信动作并非直接完成目标，而是依赖上下文的干预：有时，不必要的交流会拖慢进度；有时，则能避免错误计划并改善协作。因此，我们将通信视为用于诊断分析的描述性指标，而非成功指标；主要比较则侧重于任务表现和令牌使用量。

基线方法。 我们将 PCE 与四种具有代表性的基于大语言模型的协作智能体框架进行了比较：CoELA（Zhang 等，2024b年）、REVECA（Seo 等，2025）、CaPo（Liu 等，2025）和 CoTS （ Zu 等，2025 ）。CoELA 首先展示了由大语言模型驱动的具身智能体协作。REVECA 引入了记忆管理、基于相对邻近度的规划和计划验证。CaPo 通过迭代辩论设计了计划优化，而 CoTS 则将多计划探索与蒙特卡洛树搜索相结合。所有基线方法均在相同的环境和通信设置下运行。对于 PCE框架，我们使用默认超参数$D = 3,\ \alpha = 1,\ \beta = 1,\ \lambda = 1,\ K_{action} = 10,\ K_{message} = 3$。有关基线方法的更多细节，请参阅附录 A.2。

大语言模型。 为测试不同模型骨干网络的鲁棒性，我们在三个各具特色的大语言模型上运行每个框架：*gpt-4o-mini-2024-07-18* （GPT-4o mini）（Hurst 等人，2024年）、*google/gemma-3-4b-it*（Gemma3:4B）（Team 等人，2025）和*openai/gpt-oss-20b* （GPT-OSS:20B）（Agarwal 等人，2025）。GPT-4o mini 是商用大语言模型，而 Gemma3:4B 和 GPT-OSS:20B 是开源模型。GPT-OSS:20B 是一种大型推理模型，会在生成回复前进行内部推理；在对比中，我们使用其中等推理级别。GPT-4o mini 和 Gemma3:4B 不会借助显式推理模块生成输出，因此我们采用零样本思维链提示词，以促使其生成质量更高的计划。

表1：C-WAH 的实验结果。最佳结果以粗体标示；次佳结果加下划线。

		PCE	CoELA	REVECA	CaPo	CoTS
GPT-4o mini	总步数 ↓	**42.76**	60.40	46.80	60.82	64.00
	Comm	1.70	9.88	6.00	8.72	10.24
	使用次数 ↓	44353.56	55467.12	46312.16	**41702.00**	44628.12
GPT-OSS:20B	总步数 ↓	**49.60**	72.72	53.86	68.34	65.26
	Comm	2.11	9.22	6.49	8.20	8.32
	使用次数 ↓	**73535.24**	77727.20	74764.24	99810.44	95428.84
Gemma3:4B	总步数 ↓	**59.20**	77.20	62.56	75.88	72.32
	Comm	3.02	9.48	9.14	7.92	4.04
	使用次数 ↓	50984.7	49271.24	**44637.58**	64015.24	51966.64

表2：TDW-MAT 中的实验结果。最佳结果以粗体显示；次佳结果加下划线。

		PCE	CoELA	REVECA	CaPo	CoTS
GPT-4o mini	总计 ↑	**87.50**	62.50	81.25	73.33	75.00
	食物 ↑	**89.17**	65.83	80.83	82.50	84.17
	杂物 ↑	**85.83**	59.17	81.66	64.17	65.83
	Comm	3.58	13.33	43.76	70.79	108.92
	使用次数 ↓	197807.29	**113058.83**	185453.54	281404.71	411392.08
GPT-OSS:20B	总计 ↑	**81.25**	55.00	73.33	65.41	59.17
	食物 ↑	**85.00**	50.83	78.33	72.50	70.83
	杂物 ↑	**77.50**	59.17	68.33	58.33	47.50
	Comm	13.75	11.62	107.79	43.00	41.83
	使用次数 ↓	337225.12	**237498.88**	370737.17	348066.92	334912.67
Gemma3:4B	总计 ↑	**70.83**	45.84	52.09	67.50	63.33
	食物 ↑	**71.66**	50.00	56.67	70.83	64.17
	杂物 ↑	**70.00**	41.67	47.50	64.17	62.50
	Comm	9.42	27.42	108.00	57.88	55.96
	使用次数 ↓	184809.08	**98350.25**	308221.25	217626.50	212029.79

5.1 比较结果

表1和表2总结了双智能体在 C-WAH 和 TDW-MAT 上的协作结果。在三个大语言模型骨干网络上，我们的 PCE框架始终能在 C-WAH 中实现最快的目标完成速度，并在 TDW-MAT 中取得最高的成功率，在总计、食物和杂物 指标上均优于现有方法。

这些收益源于智能体能够在不确定性下以最少的通信采取行动。我们的方法通过明确地构建假设，而非依赖反复对话，减少了消息交换所耗费的时间，并将更多时间用于物理动作。这种行为有效抑制了不必要的通信驱动规划循环，从而降低了大语言模型推理成本。此外，尽管与 CoELA 等每步执行两次大语言模型推理的架构相比，PCE 的三模块大语言模型架构会带来更高的每步推理成本，但 PCE 大幅缩短了回合长度，抵消了这一开销。因此，PCE 在去中心化部分可观测性下实现了高性能，同时保持较低的使用次数 。

相比之下，CoELA、REVECA、CaPo 和 CoTS 通常高度依赖通信，这会增加步数并延迟目标达成。CoELA 缺乏系统的通信策略以及评估通信价值的机制，因此难以识别在长时域多房间环境中何时需要通过信息交换来消除不确定性。REVECA 可以请求验证目标导向计划所需的信息，但主动识别高度不确定因素并向其发起查询的能力有限。CaPo 和 CoTS 的优势在于能够生成多步计划，但在高不确定性的环境中，这些计划往往会失效，导致反复重新规划。

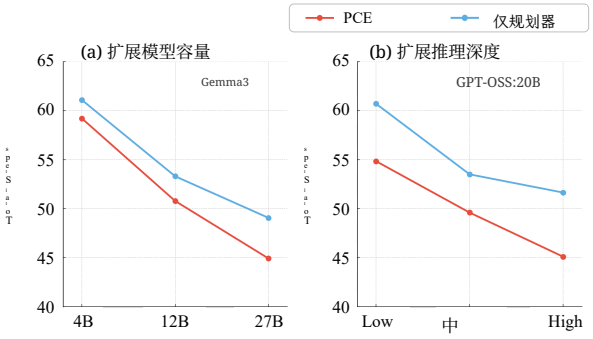

图3：C-WAH 环境中关于大语言模型扩展的消融实验结果。

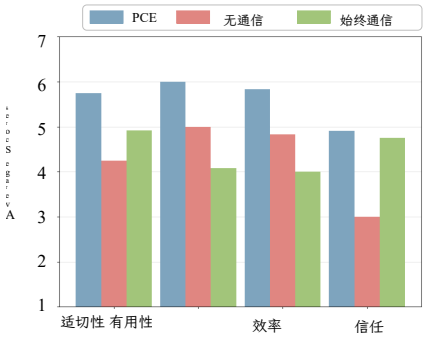

图4：C-WAH 环境中的用户研究结果。

表3：C-WAH 中的消融实验结果。最佳结果以粗体显示，次佳结果加下划线。

		PCE	无规划器	无组合器	无评估器
GPT-4o mini	总步数 ↓	**42.76**	56.46	<u>46.82</u>	47.34
	Comm 使用次数 ↓	1.70	9.52	0.26	1.2
		<u>44353.56</u>	139918.56	**33347.66**	44720.38

相比之下，PCE 通过将假设提取、场景构建和评估分离，在所有骨干模型和环境中都取得了稳定的改进。这使智能体能够识别何时进行信息交换确有价值，以有理有据的方式在物理动作与通信动作之间作出选择，并在去中心化部分可观测性下保持较高性能。

5.2 消融研究结果

大语言模型扩展。 我们首先考察，性能收益是否源于我们的结构设计，而非仅仅扩大大语言模型能力或加深推理深度。为此，我们将 PCE 与移除了组合器和评估器、因而不再进行显式不确定性处理的仅规划器 变体进行比较。如图3所示，无论是将骨干模型规模从 Gemma3:4B→12B→27B 提高，还是将 GPT-OSS:20B 的推理深度从低→中→高调高，仅规划器 都仅有小幅改进，而 PCE 始终能更快地完成目标。这表明，仅靠扩展规模，零散且隐式处理的假设依然存在，导致探索效率低下，并偶尔出现规划错误。相比之下，PCE 的合成器–评估器会显式组织并评估这些假设，表明我们的结构化不确定性处理能够与规模扩展协同作用，无论底层骨干模型如何，都能带来稳定的性能优势。

组件分析。我们还使用 GPT-4o mini 对 PCE 流程中的各个组件进行了消融实验。如表3所示，移除任何模块（无规划器、无组合器、无评估器）都会降低性能。没有规划器时，系统会直接根据上下文构建情景树，这会增加情景探索的难度，并且常常导致分支不连贯或冗余。没有组合器时，智能体在评估时只能依赖规划器的推理轨迹，因此无法考虑多个假设之间的冲突或互补关系，导致对备选情景的探索不完整。没有评估器时，系统在选择动作时不会进行定量可能性—收益—成本评估，从而降低决策质量。这些结果证实，每个模块都对不确定性感知规划至关重要。
我们进行了额外分析，以对 PCE 的结构基础进行压力测试，并严格评估其可靠性和可扩展性。

首先，我们构建了 PCE 的纯物理和纯通信变体，并进行了对比实验；此外，还分别将 PCE 与推理中心基线方法（思维链、思维树、自洽性）进行了比较。这些分析证实，PCE 的高

性能并非仅仅源于更复杂的推理轨迹，而是源于对不确定性的显式结构化与评估（附录 A.5）。

其次，我们通过一系列定量和定性评估来检验该框架的鲁棒性：(1) 随着智能体数量增加进行可扩展性测试，以验证协作效率（附录 A.9）；(2) 基于人类专家相关性研究，对组合器和评估器进行可靠性评估（附录 A.10、A.11）；(3) 进行超参数敏感性分析，以考察配置稳定性（附录 A.5）；(4) 与基于蒙特卡洛树搜索的规划器进行比较，以证明其相较于传统树搜索的效率优势（附录 A.8）。
最后，我们通过定性案例研究展示 PCE 如何借助结构化决策过程纠正不适定的计划，从而为观察到的定量提升提供可解释的依据（附录 A.7）。

5.3 用户研究结果

在人类实际协作中，通信是一把双刃剑：过多的信息交流会打乱工作流程，而缺少必要的交流则会使意图变得模糊，并降低协同表现。我们假设，PCE 能够结构化并评估假设，因此可以仅在确实有用时才触发通信，从而产生在人类看来协调一致、高效且值得信赖的行为。为验证这一点，我们在 C-WAH 环境中开展了一项用户研究，对比了 (1) PCE、(2) 无通信 （移除通信动作），以及 (3) 始终通信（每次物理动作前都强制通信）。12 名参与者（平均年龄 26.8 岁；8 名男性，4 名女性）获得了与智能体相同的观测和动作选择。

每完成一种方法后，参与者都会根据 7 点李克特量表（1：非常不同意，7：非常同意）回答四个问题：1）“智能体的动作是否符合您的意图？”（适切性）；2）“智能体在协作中是否有帮助？”（有用性）；3）“智能体的表现是否有助于高效实现目标？”（效率）；4）“您是否信任该智能体？”（信任）。随后，他们参加了简短的定性访谈。
如图4所示，PCE 在所有问题上的得分均为最高，证实了我们的假设：通过对不确定性进行推理并有选择地发起通信，PCE 实现了清晰度与效率之间的平衡，用户认为这种平衡体现出更强的协作性和可靠性。访谈反馈指出，始终通信会打断工作流程，而无通信则会使智能体的意图不明确，因而更难以获得信任。详细结果见附录 A.6。

6 结论

本文提出了 PCE，这是一种模块化的规划器-组合器-评估器框架，可从大语言模型推理轨迹中提取并构建其中隐含的假设，使具身智能体能够在部分可观测条件下以最少的通信进行规划。在两个多目标基准（C-WAH 和 TDW-MAT）以及三种不同的大语言模型骨干网络上，PCE 在成功率和任务效率方面始终优于以通信为中心的基线方法，同时令令牌用量保持相当水平。消融研究证实，每个模块都不可或缺，且显式处理不确定性能够带来仅靠模型规模扩展无法实现的性能提升。用户研究进一步表明，PCE 生成的通信模式在人类看来既高效又值得信赖。
虽然我们的实验聚焦于模拟多房间家庭任务，但所提出的机制并不局限于特定领域。未来的工作将探索把 PCE 扩展到更复杂的动态环境、更大且更多样化的智能体团队，以及实时自适应发现新假设。这些方向旨在检验该框架在更丰富的不确定性条件下的可扩展性和通用性，推动实现具身智能体有原则的去中心化协作这一更广泛的目标。

7 可复现性声明

为确保所提出方法和实验的可复现性，我们将代码作为匿名补充材料的一部分提供。这些材料包含复现本文所述方法所需的代码和说明。

参考文献

Sandhini Agarwal、Lama Ahmad、JasonAi、Sam Altman、Andy Applebaum、EdwinArbus、Rahul K Arora、Yu Bai、Bowen Baker、Haiming Bao 等。gpt-oss-120b 与 gpt-oss-20b 模型卡。*arXiv* 预印本 *arXiv:2508.10925*，2025。

¨Christopher Amato、Girish Chowdhary、Alborz Geramifard、N Kemal Ure 和 Mykel J Kochen-derfer。部分可观测马尔可夫决策过程的分散式控制。发表于第 *52* 届 *IEEE* 决策与控制会议，第 2398–2405 页。IEEE，2013 年。

Christopher Amato、George Konidaris、Gabriel Cruz、Christopher A Maynor、Jonathan P How 和 Leslie P Kaelbling。不确定性条件下多个机器人的分散式控制规划。发表于*2015* 年 *IEEE* 国际机器人与自动化会议（*ICRA*），第 1241–1248 页。IEEE，2015 年。

Christopher Amato、George Konidaris、Ariel Anders、Gabriel Cruz、Jonathan P How 和 Leslie P Kaelbling。在不确定性下进行多机器人协同的策略搜索。国际机器人研究杂志，35(14):1760–1778，2016。

Daniel S Bernstein、Robert Givan、Neil Immerman 和 Shlomo Zilberstein。分散式马尔可夫决策过程控制的复杂度。运筹学数学，27(4): 819–840，2002。

Jakob Foerster、Ioannis Alexandros Assael、Nando De Freitas 和 Shimon Whiteson。通过深度多智能体强化学习学习通信。神经信息处理系统进展，29，2016。

Pascale Fung、Yoram Bachrach、Asli Celikyilmaz、Kamalika Chaudhuri、Delong Chen、Willy Chung、Emmanuel Dupoux、Hongyu Gong、Herv´e J´egou、Alessandro Lazaric 等。具身人工智能智能体：世界建模。*arXiv* 预印本 *arXiv:2506.22355*，2025。

Chuang Gan、Jeremy Schwartz、Seth Alter、Damian Mrowca、Martin Schrimpf、James Traer、Ju- lian De Freitas、Jonas Kubilius、Abhishek Bhandwaldar、Nick Haber、Megumi Sano、Kuno Kim、Elias Wang、Michael Lingelbach、Aidan Curtis、Kevin Feigelis、Daniel M. Bear、Dan Gutfreund、David Cox、Antonio Torralba、James J. DiCarlo、Joshua B. Tenenbaum、Josh H. McDermott 和 Daniel L. K. Yamins。Threed world：一个用于交互式多模态物理仿真的平台，2021a。网址：https://arxiv.org/abs/2007.04954。

Chuang Gan、Siyuan Zhou、Jeremy Schwartz、Seth Alter、Abhishek Bhandwaldar、Dan Gutfreund、Daniel LK Yamins、James J DiCarlo、Josh McDermott、Antonio Torralba 等人。《Threedworld 运输挑战：一个面向物理真实具身人工智能的视觉引导任务与运动规划基准测试》。*arXiv* 预印本 *arXiv:2103.14025*，2021b。

Xudong Guo、KaixuanHuang、Jiale Liu、Wenhui Fan、NataliaV´elez、QingyunWu、HuazhengWang、Thomas L Griffiths 和 Mengdi Wang。《具身大语言模型智能体学会在有组织的团队中合作》。*arXiv* 预印本 *arXiv:2403.12482*，2024年。

Wenlong Huang、Pieter Abbeel、Deepak Pathak 和 Igor Mordatch。《语言模型作为零样本规划器：为具身智能体提取可执行知识》。载于国际机器学习会议，第 9118–9147 页。PMLR，2022。

Yuheng Huang、Jiayang Song、Zhijie Wang、Shengming Zhao、Huaming Chen、Felix Juefei-Xu 与 Lei Ma。三思而后行：大语言模型不确定性度量的探索性研究。*arXiv* 预印本 *arXiv:2307.10236*，2023。

AaronHurst、AdamLerer、AdamPGoucher、AdamPerelman、Aditya Ramesh、AidanClark、AJ Ostrow、Akila Welihinda、Alan Hayes、Alec Radford 等。GPT-4o 系统卡片。*arXiv* 预印本 *arXiv:2410.21276*，2024年。

Jared Kaplan、Sam McCandlish、Tom Henighan、Tom B Brown、Benjamin Chess、Rewon Child、Scott Gray、Alec Radford、Jeffrey Wu 和 Dario Amodei。《神经语言模型的缩放定律》。*arXiv* 预印本 *arXiv:2001.08361*，2020 年。

Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Heinrich K¨uttler、Mike Lewis、Wen-tau Yih、Tim Rockt¨aschel 等。《面向知识密集型 NLP 任务的检索增强生成》。神经信息处理系统进展，33：9459–9474，2020 年。

Manling Li、Shiyu Zhao、Qineng Wang、Kangrui Wang、Yu Zhou、Sanjana Srivastava、Cem Gokmen、Tony Lee、Erran Li Li、Ruohan Zhang 等。《具身智能体接口：具身决策大语言模型基准测试》。神经信息处理系统进展，37：100428–100534，2024年。

Xiaoxi Li、Guanting Dong、Jiajie Jin、Yuyao Zhang、Yujia Zhou、Yutao Zhu、Peitian Zhang 和 Zhicheng Dou。Search-o1：智能体搜索增强的大型推理模型。*arXiv* 预印本 *arXiv:2501.05366*，2025。

Jie Liu、Pan Zhou、Yingjun Du、Ah-Hwee Tan、Cees GM Snoek、Jan-Jakob Sonke 和 Efstratios Gavves。CaPo：面向高效具身多智能体协作的协作规划优化。载于第十三届国际学习表征会议，2025。

Zhao Mandi、Shreeya Jain 和 Shuran Song。RoCo：利用大语言模型进行辩证式多机器人协作。载于 *2024 IEEE* 国际机器人与自动化会议（*ICRA*），第 286–299 页。IEEE，2024。

Iman Mirzadeh、Keivan Alizadeh、Hooman Shahrokhi、Oncel Tuzel、Samy Bengio 和 Mehrdad Farajtabar。GSM-Symbolic：理解大语言模型数学推理的局限性。*arXiv* 预印本 *arXiv:2410.05229*，2024。

Sid Nayak、Adelmo Morrison Orozco、Marina Have、Jackson Zhang、Vittal Thirumalai、Darren Chen、Aditya Kapoor、Eric Robinson、Karthik Gopalakrishnan、James Harrison 等。面向部分可观测环境中多智能体机器人的长时域规划。神经信息处理系统进展，37：67929–67967，2024年。

Frans Oliehoek 和 Matthijs Spaan。面向通信延迟的多智能体部分可观测马尔可夫决策过程的基于树的求解方法。载于*AAAI*人工智能大会论文集，第26卷，第1415–1421页，2012年。Xavier Puig、Kevin Ra、Marko Boben、Jiaman Li、Tingwu Wang、Sanja Fidler 和 Antonio Torralba。Virtualhome：通过程序模拟家庭活动。载于*IEEE*计算机视觉与模式识别会议论文集，第8494–8502页，2018年。Xavier Puig、Tianmin Shu、Shuang Li、Zilin Wang、Yuan-Hong Liao、Joshua B Tenenbaum、Sanja Fidler 和 Antonio Torralba。观察与帮助：社会感知与人类与人工智能协作的挑战。*arXiv*预印本 *arXiv:2010.09890*，2020 年。

SeungWonSeo、SeongRaeNoh、Junhyeok Lee、SooBin Lim、WonHeeLee 和 HyeongYeop Kang。《Reveca：利用信息相关性和相对邻近度，在协作式语言智能体中实现自适应规划与基于轨迹的验证》。载于 *AAAI*人工智能大会论文集，第 39 卷，第 23295–23303 页，2025。

Zhihong Shao、Peiyi Wang、Qihao Zhu、Runxin Xu、Junxiao Song、Xiao Bi、Haowei Zhang、Mingchuan Zhang、YK Li、Yang Wu 等。《DeepSeekMath：推动开放语言模型数学推理能力的极限》。*arXiv* 预印本 *arXiv:2402.03300*，2024年。

Charlie Victor Snell、Jaehoon Lee、Kelvin Xu 和 Aviral Kumar。对大语言模型测试时计算进行最优规模扩展，在推理方面可能比规模扩展参数更有效。载于第十三届国际学习表征会议，2025。

Chan Hee Song、Jiaman Wu、Clayton Washington、Brian M Sadler、Wei-Lun Chao 和 Yu Su。Llm-planner：基于大语言模型的具身智能体少样本接地规划。载于*IEEE/CVF*国际计算机视觉大会论文集，第2998–3009页，2023。

Matthijs TJ Spaan、Geoffrey J Gordon 和 Nikos Vlassis。不确定性下通信智能体团队的去中心化规划。载于 第五届自主智能体与多智能体系统国际联合会议论文集，第 249–256 页，2006 年。

Gemma Team、Aishwarya Kamath、Johan Ferret、Shreya Pathak、Nino Vieillard、Ramona Merhej、Sarah Perrin、Tatiana Matejovicova、Alexandre Ram´e、MorganeRivi`ere 等。Gemma 3 技术报告。*arXiv* 预印本 *arXiv:2503.19786*，2025 年。

Guanzhi Wang、Yuqi Xie、Yunfan Jiang、Ajay Mandlekar、Chaowei Xiao、Yuke Zhu、Linxi Fan 和 Anima Anandkumar。Voyager：一种借助大语言模型的开放式具身智能体。*arXiv* 预印本 *arXiv:2305.16291*，2023a。

Shuhe Wang、Shengyu Zhang、Jie Zhang、Runyi Hu、Xiaoya Li、Tianwei Zhang、Jiwei Li、Fei Wu、Guoyin Wang和Eduard Hovy。强化学习增强的大语言模型：综述。*arXiv*预印本 *arXiv:2412.10400*，2024年。

Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc V Le、Ed H Chi、Sharan Narang、Aakanksha Chowdhery和Denny Zhou。自洽性提升语言模型中的思维链推理。载于第十一届国际学习表征会议，2023b。

Zihao Wang、Shaofei Cai、Guanzhou Chen、Anji Liu、Xiaojian Shawn Ma和Yitao Liang。描述、解释、规划与选择：基于大语言模型的交互式规划实现开放世界多任务智能体。神经信息处理系统进展，36:34153–34189，2023c。

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou 等。思维链提示能够激发大语言模型的推理能力。神经信息处理系统进展，35:24824–24837，2022。

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng 等。迈向大型推理模型：大语言模型强化推理综述。*arXiv* 预印本 *arXiv:2501.09686*，2025。

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao 和 Karthik Narasimhan。思维树：运用大语言模型进行审慎的问题求解。神经信息处理系统进展，36:11809–11822，2023。

Zhangyue Yin、Qiushi Sun、Qipeng Guo、Zhiyuan Zeng、Xiaonan Li、Junqi Dai、Qinyuan Cheng、Xuan-Jing Huang 和 Xipeng Qiu。动态推理：通过不确定性感知自适应引导增强大语言模型的推理能力。载于第*62*届计算语言学协会年会论文集（第*1*卷：长篇论文），第 2401–2416 页，2024年。

Ceyao Zhang、Kaijie Yang、Siyi Hu、Zihao Wang、Guanghe Li、Yihang Sun、Cheng Zhang、Zhaowei Zhang、AnjiLiu、Song-Chun Zhu 等。ProAgent：利用大语言模型构建主动协作智能体。载于*AAAI*人工智能大会论文集，第 38 卷，第 17591–17599 页，2024a年。

Hongxin Zhang、Weihua Du、Jiaming Shan、Qinhong Zhou、Yilun Du、Joshua B Tenenbaum、Tian-min Shu 和 Chuang Gan。利用大语言模型模块化构建协作具身智能体。载于第十二届国际学习表征会议，2024b年。

Hongxin Zhang、Zeyuan Wang、Qiushi Lyu、Zheyuan Zhang、Sunli Chen、Tianmin Shu、Behzad Dariush、Kwonjoon Lee、Yilun Du 和 Chuang Gan。Combo：用于具身多智能体协作的组合式世界模型。载于第十三届国际学习表征会议，2024c年。

Gengze Zhou、Yicong Hong 和 Qi Wu。Navgpt：利用大语言模型在视觉与语言导航中进行显式推理。发表于*AAAI*人工智能大会论文集，第 38 卷，第 7641–7649 页，2024年。

Lizheng Zu、Lin Lin、Song Fu、Na Zhao 和 Pan Zhou。通过协作树搜索增强具身多智能体协作。发表于计算机视觉与模式识别会议论文集，第 29513–29522 页，2025。

附录 A

A.1 环境设置细节

在我们的实验中，智能体的观测通过每个环境提供的 API 获取，并遵循先前的研究（Zhang 等，2024b）。我们在部分可观测环境中进行实验，包括 C-WAH 和 TDW-MAT。在下一节中，我们将介绍这些环境的细节。



（a）C-WAH　　(b) TDW-MAT

图 5：示例环境

A.1.1 通信式观察与帮助

通信式观察与帮助（C-WAH）环境在原始的观察与帮助挑战赛（Puig 等人，2020 年）基础上扩展，新增了智能体之间的通信功能。该环境基于面向多智能体仿真的平台 VirtualHome（Puig 等人，2018年）实现。我们在 C-WAH 中运行了 10 个回合，每个回合都会为智能体分配一个共同目标。如表 4 所示，这些共同目标分为五类，智能体必须从表 5 列出的八种可能动作中进行选择以达成目标。

在此设定中，智能体可以在执行指令时通过通信相互交换信息。进入房间后，智能体可以观察所有不在冰箱或微波炉等容器内的物体。若要查看容器内的物体，智能体必须明确执行一个额外动作将容器打开。为模拟现实世界中的通信限制，每个智能体每帧最多只能发送 500 个字符。

时域H 固定为 250 个仿真步，每项任务包含 3 到 5 个子目标（即物体）。如果未能在 250 步内达成共同目标，则该回合失败。图 5-(a) 展示了环境布局示意图。

表格 4：C-WAH 的目标设定

目标	**描述**
准备下午茶	放置 [纸杯蛋糕、布丁、苹果、果汁、葡萄酒] 在 咖啡桌上
清洗餐具	放置 [盘子，叉子]里面洗碗机
准备一餐	放置 [咖啡壶、纸杯蛋糕、煎饼、磅蛋糕、布丁、苹果、果汁，葡萄酒] 在 餐桌
放置食品杂货	放置 [纸杯蛋糕、煎饼、磅蛋糕、布丁、苹果、果汁、葡萄酒] 里面冰箱
布置餐桌	放置 [盘子，叉子]在 餐桌

表格 5：C-WAH 中的动作空间

动作	描述
走向	移动到与智能体位于同一房间的物体，或移动到某个房间
左转	向左转30度
右转	向右转30度
抓取	抓取一个物体
Open	打开一个关闭的容器
关闭	关闭一个打开的容器
Put	将持有物体放入一个打开的容器中或放置到表面上
发送消息	向其他智能体发送消息

A.1.2 THREEDWORLD 多智能体运输

ThreeDWorld 多智能体运输（TDW-MAT）环境是 ThreeDWorld 运输挑战（Gan 等人，2021b）的扩展版本，基于 TDW（Gan 等人，2021a）构建。该环境采用了更自然的物体摆放，并提供了种类更丰富的物体和容器，以支持物品运输。TDW-MAT 中的共同目标是运输两类物品：食物和杂物。每个回合包含 10 个目标物体和 2–5 个容器，这些容器经过策略性摆放，以便一次运输多件物品，详情见表6。

与 C-WAH 不同，TDW-MAT 中的智能体必须以 15度增量进行完整的 360度旋转，才能获取房间的完整信息。通信限制为每帧 500 个字符，时域 H 设为 3000 个仿真步。环境布局如图 5-(b) 所示，完整的动作空间见表7。

表6：TDW-MAT 环境中的目标物体和容器。

Task	Type	物体名称
Food	容器	碗、盘子、茶盘
	物体	面包、汉堡、一条面包、苹果、香蕉、橙子
杂物	容器	塑料篮、柳条篮、木篮
	物体	*iPhone*、钢笔、钥匙、*iPod*、打火机、钱包、计算器、笔筒、鼠标

表7：TDW-MAT 环境的动作空间。

动作	描述
向前移动	向前移动 0.5 米
左转	左转15度
右转	右转15度
抓取	抓取一个物体
放入	将目标放置到容器中
Drop	放下手中拿着的物体
发送消息	向其他智能体发送消息

A.2 基线方法

CoELA（Zhang 等，2024b）采用模块化的由大语言模型驱动的多智能体架构：每个智能体都包含感知、记忆、规划、通信和执行模块。智能体交换语言信息，并推理接下来该做什么。该架构是去中心化且灵活的，配合强大的大语言模型，能够在具身任务中实现良好的多智能体协作。

REVECA（Seo 等，2025）提出了一种基于大语言模型的协作式智能体架构，利用信息相关性和相对邻近度进行自适应规划，并采用基于轨迹的验证，防止协作者的动作导致错误规划，从而在部分可观测性下实现更高效的记忆使用和更强的协作。

CaPo（Liu 等，2025）引入了协作式元计划生成阶段，由智能体共同构建高层次任务分解，并在执行前分配子任务。在任务执行过程中，智能体进行进度自适应重规划，在观察到新信息或环境变化时动态更新元计划。这种先构建结构化计划、再进行自适应修订的两阶段流程增强了协作，减少了冗余，并提高了具身多智能体场景中的任务效率。

CoTS（Zu 等，2025）提出了一种受蒙特卡洛树搜索启发、用于多智能体规划的协作树搜索框架。智能体不再沿单一路径执行计划，而是探索多个候选分支，使用基于大语言模型的推理或启发式评分对其进行评估，然后共同确定一致的长期策略。专门的计划评估模块会监控执行进度，并在必要时选择性地触发计划更新，从而兼顾适应性与稳定性。这种设计能够提供更强的前瞻性，提高对规划错误的韧性，并减少协作过程中的不必要干扰。

A.3 记忆模块中的记忆示例 LES

本节介绍记忆模块中存储的信息，包括共同目标、物体信息、协作者信息、消息日志、先前操作和低层动作技能库，并对每项内容进行详细说明，同时提供示例。

共同目标。 共同目标 G 以自然语言描述智能体在给定回合中必须共同完成的家庭任务。以下是 C-WAH 和 TDW-MAT 中共同目标的示例。

- **C-WAH**：“找到目标物体并将其放到目标位置 <咖啡桌> (268)上：1 个布丁、1 瓶果汁、1 个苹果、2 个纸杯蛋糕。”
- **TDW-MAT**：“将 2 个面包、2 个汉堡、4 个苹果、1 条面包和 1 根香蕉运到床上。”

物体信息。 物体 ID 和名称、三维位置、房间 ID 和名称、可用动作以及物体状态（例如，物体是否被拿着、是否位于容器内，或是否可以抓取）。列表 1 展示了一个示例。

协作者信息。 协作者持有的物体及其三维位置。当协作者进入观测范围时，这些信息会更新，模块会存储最近一次观测到的值。

消息日志。 所有智能体发送的消息。仅存储最新的 $K_{message}$ 条消息，在我们的实现中，设定为 $K_{message} = 3$。

先前操作。 存储智能体从过去到现在执行的动作。仅保留最新的 K_{action} 个动作，在我们的实现中，设定为 $K_{action} = 10$。

低层动作技能库。低层动作技能库概述了与环境交互所需的 Python API。列表 2 展示了一个示例。

列表 1：物体信息示例

```
1# 一个只有一条记录的物体信息列表示例。2物体_信息_列表 = [{ 3 "物体_标识符
": 21, 4 "物体_名称": "苹果", 5 "位置": [13.22, 1.20, 5.41],6 "可用_动作": "gograb", 7 "
房间_名称": "客厅", 8 "房间_标识符": 198, 9 "状态": [15,"可抓取"]10 }]
```

列表 2：低层动作技能库示例

```
1# 低层动作技能库的简单示例 2def goexplore(self): 3 if current_room ==target_room: 4 #
根据规划前往指定房间。 5 6def gocheck(self): 7 if 'OPEN' in container['states']: 8 # 检查容器的
状态并尝试与其交互。 9 10def gograb(self): 11 if target in reachable_objects: 12 # 尝试抓取物
体，并确保其可用且符合条件。 13 14def goput(self): 15 if len(grabbed_objects) > 0: 16 # 将抓
取的物体放置在指定位置或容器中。
```

A.4 基于距离的规划

仅提供可用动作列表 时，大语言模型智能体在规划过程中往往难以考虑高效的移动路径或能够提升协作效率的分工。因此，遵循以往研究，我们也将物理距离作为一项关键认知因素。具体而言，记忆模块会根据每个可用动作对应的目标物体位置、最近一次观测到的协作者位置以及智能体自身的位置，计算智能体—物体距离和协作者—物体距离。这些距离标注会被纳入可用动作的文本描述，从而引导规划器在选择动作时兼顾空间效率和任务分配。最终，规划器将增强后的文本提示词作为输入，并从可用动作列表 中选择最适合实现共同目标的动作。

A.5 补充消融研究结果

不确定性处理。 我们分析物理动作与通信动作之间的成本—收益权衡，二者采用截然不同的不确定性处理策略。为此，我们比较四种基线方法：PCE、仅物理动作、仅通信动作和仅规划器。具体而言，仅物理动作 仅依靠组合器的物理动作来消解不确定性，而仅通信动作 只使用通信动作。对于所有基线方法，我们均使用GPT-4o mini作为骨干大语言模型。表8（不确定性处理）中的结果表明，PCE以比其他基线方法更少的步数 达成共同目标。特别是，仅物理动作 完全消除了通信成本，但必须通过物理探索加以补偿，从而导致过多的移动并延迟目标达成。在仅通信动作 条件下，智能体试图在执行任何物理动作之前，优先通过通信解决所有不确定性。具体而言，评估器首先检查决策树中所有可通过

表格 8：C-WAH 中的消融实验结果。最佳结果以粗体标示；次佳结果加下划线。

		PCE	仅通信动作	仅物理动作	仅规划器
不确定性处理	总步数 ↓	**42.76**	46.69	<u>45.38</u>	46.69
	Comm	1.70	4.78	0.0	2.39
	使用次数 ↓	<u>44353.56</u>	49051.36	45536.14	**26678.88**
		PCE	***CoT***	***ToT***	***SC***
推理增强	总步数 ↓	**42.76**	55.40	<u>50.96</u>	54.86
	Comm	1.70	6.56	1.22	1.94
	使用次数 ↓	**44353.56**	130633.3	149666.80	<u>129134.10</u>

表 9：超参数敏感性分析结果。

	变体*1*	总步数	变体*2*	总步数
树最大深度（D）	$D = 2$	44.6	$D = 4$	42.4
代价权重（α）	$\alpha = 0.5$	50.1	$\alpha = 1.5$	45.5
代价权重 (β)	$\beta = 0.5$	48.6	$\beta = 1.5$	44.6
全局惩罚 (λ)	$\lambda = 0.5$	58.7	$\lambda = 1.5$	45.4
记忆历史 (K_{action})	$K_{action} = 5$	44.4	$K_{action} = 15$	44.3
记忆历史 ($K_{message}$)	$K_{message} = 2$	49.5	$K_{message} = 4$	49.7

通信加以澄清的场景。每次通信动作之后，PCE框架会被再次调用，组合器重建决策树，以检查是否还能通过对话获得更多信息。这一迭代过程持续进行，直到组合器判定无法再从协作者处获取进一步有意义的信息。只有当决策树中没有任何场景能够通过通信带来额外收益时，智能体才会继续执行物理动作。因此，仅通信动作 能够快速获取协作者信息，但冗余通信会延迟目标完成并增加令牌成本。此外，仅规划器由于缺乏任何显式的不确定性处理，整体协作性能最低。综合来看，这些发现表明，偏向单一不确定性消解模式的方法在成本效益上存在固有的结构性局限。相比之下，PCE通过合成器–评估器以均衡方式利用两种策略，从而在最小化成本的同时提升不确定性处理与协作性能。

推理增强。 我们分析了不同推理增强策略在部分可观测条件下协作规划中的有效性。为此，我们比较了四种基线方法：PCE、思维链（*CoT*）、思维树（*ToT*）和自洽性（*SC*）。具体而言，*CoT* 通过单条线性思维链进行规划；*ToT* 利用思维树将推理扩展为多个分支，并通过树搜索探索候选方案；*SC* 则通过采样多条推理轨迹并以多数投票进行汇总来实现自洽性。对于所有基线方法，我们均使用 GPT-4o mini 作为底层大语言模型。表 8（推理增强）所示结果表明，与仅依赖推理的基线方法相比，PCE 始终能更高效、更可靠地实现共同目标。具体而言，*CoT* 能够提供稳定的推理，但缺乏多样性，因此在不确定性下往往会产生脆弱的计划。*ToT* 通过探索多个分支提升了多样性，但会产生显著的令牌成本；由于缺乏明确的不确定性评估，有时还会选择次优分支。*SC* 通过汇总多个采样计划提高了鲁棒性，但无法区分相互冲突的假设，导致决策不一致。综上，这些发现表明，仅通过扩大规模或增加采样来增强推理，并不足以应对不确定条件下的协作规划。此外，若不对不确定性进行明确的结构化处理和评估，随着推理深度增加，错误假设可能被放大，进而导致幻觉累积和错误传播。相比之下，PCE 通过合成器–评估器对假设进行明确的结构化处理和评估，从而实现有原则的不确定性感知规划，兼顾效率、鲁棒性和成本效益。

表10：用户研究参与者的代表性访谈回答。

	参与者反馈
无通信	“完全没有对话，这在某种程度上很方便，但它确实如此感觉不像是在进行协作。” “由于智能体没有说话，很难知道它在做什么，我无法信任它。” “完全没有任何协作。”
始终通信	“智能体说得太多了，我的工作流程不断被打断。” “频繁的通信有一些好处，但一直说话让人不舒-服。” “分工更容易了，但频繁交谈让人难以集-中注意力，而且令人沮丧。”
PCE	“没有不必要的通信，我也没有感到任何不便。” “通信的频率恰到好处，而且智能体在被询问时回答得很友善。” “我认为智能体工作得高效且有效。”

超参数敏感性分析。为考察各超参数对规划深度、沟通频率和执行效率的影响，我们在C-WAH 基准测试上进行了全面的敏感性分析。每次仅单独改变一个参数，同时保持其他参数不变。完整数值结果见表 9。在默认配置（$D = 3$，$\alpha = 1$，$\beta = 1$，$\lambda = 1$，$K_{action} = 10$，和 $K_{message} = 3$）下，模型达到 42.76 总步数。较浅的树（$D = 2$）限制了对替代假设情景的探索，导致性能下降。增加深度（$D = 4$）仅带来微小收益（42.4 步），却会在最坏情况下造成树规模指数级增长，从而提高幻觉风险和计算成本。这些结果表明，$D = 3$在表达能力与鲁棒性之间实现了有效平衡。系数 α 和 β 控制物理动作成本与沟通行动成本的相对影响。较低的 α 或较高的 β 会导致智能体低估物理动作，从而进行过度探索。反之，较高的 α 或较低的 β 会使决策偏向以沟通为主的策略，同样会降低性能。默认值（α=1，β=1）在各模态之间实现了均衡加权。全局惩罚 λ 调节预期效用与执行成本之间的权衡。较小的值会过度强调可能性和收益，往往促使智能体选择乐观但风险较高的分支。较大的值则会导致行为过于保守。经验趋势表明，λ=1 能够提供稳定的折中方案。改变记忆历史窗口（K_{action}，$K_{message}$）的结果显示，窗口过短或过长都会对性能产生负面影响。较短的历史会削弱长时程一致性，而较长的历史则会纳入无关上下文并增加输入规模。对于 $K_{message}$，敏感性尤为明显；偏离默认值会损害部分可观测条件下的协作。

A.6 补充用户研究结果

为进行比较分析，我们首先使用 Shapiro–Wilk 检验和 Kolmogorov–Smirnov检验，在5%显著性水平下检验数据的正态性。由于结果表明数据不服从正态分布，我们采用Wilcoxon符号秩检验进行成对比较。结果见图6。

分析结果显示，在无通信 条件下，PCE 在信任 和恰当性方面的表现显著优于该条件，表明在恰当时机进行通信能够提升用户信任，并使系统与用户意图保持一致。此外，PCE 在有用性和效率方面的得分显著高于始终通信 ，这表明过度通信会妨碍协作效率，而我们的方法仅在必要时提供信息，从而提升了协作效果。

此外，为了提供更丰富的说明，我们在表10中报告了用户研究中精选的访谈回答。参与者一致指出，在无通信 条件下，由于缺少di-

		总步数↓
GPT-4o mini	PCE	**72.42**
	无通信	<u>75.67</u>
	始终通信	114.25

表11：C-WAH 环境中的用户研究结果。最佳结果以粗体显示；次佳结果加下划线。

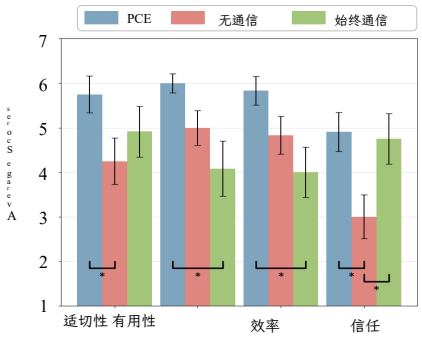

图6：使用 C-WAH 的用户研究结果。本图展示了针对四个研究问题的回答的平均得分及相关标准误。统计显著性以 * 表示，对应 $p < 0.05$。

alogue，他们很难感觉自己是在与智能体协作，这进而降低了他们的信任。在始终通信条件下，参与者表示，频繁的提问和信息请求扰乱了工作流程并延迟了目标完成。相比之下，在PCE中，参与者表示他们能够通过时机恰当的对话获得必要信息，从而加快目标达成，并且尽管与智能体交互，不适感也极小。

如表11所示，PCE所需的总步数少于两种基线方法。在无通信条件下，虽然没有对话意味着无需花费时间进行通信，但缺乏信息共享导致达成目标所需的步数大幅增加，从而降低了效率。相比之下，始终通信条件下，由于过度通信频繁打断工作流程，步数有所增加。PCE则通过及时通信获取必要信息并减少冗余移动，从而更高效地完成任务。

用户研究结果表明，通信动作不只是信息交换，还会影响人类用户与智能体交互时的体验和信任。虽然不必要的通信会降低协作效率，但符合上下文的适当通信能够加快目标达成，并增强对协作的信任。

A.7 定性案例研究

本节通过定性案例研究说明PCE框架在实践中的运作方式。具体而言，我们分析基线规划器因幻觉而提出次优或不合逻辑动作的场景，并展示PCE如何通过结构化假设生成和多准则评估加以纠正。这些案例展现了PCE在处理空间不确定性、社会推断和成本效益分析方面的灵活性。

案例1：纠正不合逻辑的物理搜索场景：智能体Alice已成功收集大部分物品，目前只剩下一件目标物品——一条面包。她目前位于客厅（标识符：4000）。卧室（标识符：8000）已经被彻底搜索过，确认其中只有一张床（没有目标物品）。与此同时，厨房（标识符：5000）和办公室（标识符：7000）仍完全未被探索。
规划器失效：依赖“剩余物品一定在某处”这一简单启发式，规划器臆想着先前访问过的房间可能仍有收益，并提议返回卧室再次搜索，忽略了负面观察历史。

PCE解决方案： 组合器识别出核心不确定性：“剩下的一条面包在哪里？”它根据相互竞争的假设构建场景树：（A）“面包在卧-

室里”和（B）“面包在厨房里”。接着，评估器为这些路径打分。包含假设 A 的路径得分极低，情景可能性 因为历史日志证实卧室已经检查过。相反，假设 B 被赋予较高的情景可能性 （因为该区域尚未探索）和较高的条件收益。因此，PCE 覆盖规划器的决策，选择 [探索] <厨房>，确保搜索策略在逻辑上保持一致。

案例 2：优化运输效率的场景：爱丽丝在客厅里，手持一件目标物品。最终放置地点（咖啡桌）就在她附近。厨房通常有许多物体，但距离很远。

规划器失效：受“厨房里有食物”这一常识性假设驱动，规划器建议前往遥远的厨房 收集更多物品后再返回。这种做法既贪心又低效，因为（1）爱丽丝只能携带有限数量的物品；（2）移动代价很高。

PCE解决方案：组合器通过引入规划器轨迹中未包含的关键假设来扩展推理轨迹：“协作者 Bob 可能已经清空了厨房。”评估器对各种场景进行权衡。前往厨房的路径会产生较高的执行成本 （行程时间），并且存在条件增益 为零的风险（如果 Bob 拿走了物品）。相比之下，前往附近放置地点的路径没有风险，并且当前物品的条件增益 有保障。因此，PCE 将动作转向 [goput],，优先确保当前收益这一全局最优行为，而非进行不确定的探索。

案例 3：战略资源管理场景： 智能体 Bob 已完成对卧室 （ID：5000）的搜索，目前位于客厅 （ID：4000）。他发现了两个容器（例如一个木篮），但尚未找到任何目标物品。其他房间仍未探索。容器距离很远（210 步），而客厅 中一处尚未探索的区域近得多（140 步）。
规划器失效： 规划器着眼于携带更多物品的长期效用，决定执行 [gograsp] <木篮>。然而，它未能考虑眼前的移动代价，也没有考虑到物品位置未知、容器是否真的有用尚不确定。

PCE解决方案： 组合器明确质疑该工具的效用，并将情境分支为“容器有助于收集”与“直接探索能更快找到物品”。评估器进行了成本效益分析，判断投入 210 步去获取效用不确定的容器，不如投入 140 步来消除附近未探索区域的高熵，更有效。因此，PCE 选择 [goexplore] <客厅> (3000)，优先考虑即时信息增益和较低代价，而非投机性的长期投资。

表 12：C-WAH 基准测试上的平均总步数比较。最佳结果以粗体标出。

	总步数 (↓)
MHP（基于蒙特卡洛树搜索）	64.90
PCE	**42.76**

A.8 补充基线方法：与基于蒙特卡洛树搜索的规划器的比较

为了对规划效率进行全面评估，我们将 PCE 与一种传统的树搜索基线进行比较：基于蒙特卡洛树搜索的分层规划器（MHP）。MHP 改编自 Watch-And-Help 挑战赛中的方法（Puig 等人，2020 年），利用蒙特卡洛树搜索来寻找最优动作轨迹。它通过最小化图遍历成本和步数来运作，主要基于环境的拓扑连通性和预定义的动作成本来优化计划。

分析。 如表 12 所示，PCE 展现出更高的效率，相较于 MHP，以显著更少的步数完成任务。尽管 MHP 能够根据拓扑结构有效地最小化遍历成本，但面对部分可观测环境中固有的高不确定性时却表现不佳。由于依赖拓扑连通性和动作成本，当目标位置在语义上存在歧义时，MHP 往往会执行低效的穷举搜索。相比之下，PCE

利用大语言模型的语义推理能力，根据上下文显式地建模可能性（例如，在未经实地验证的情况下推断苹果 很可能位于厨房 中），并依据期望效用而非仅仅依据接近程度来确定动作优先级。这些结果表明，在具身 DEC-POMDP 中，通过推理解决语义不确定性，是影响效率的更关键瓶颈，而非拓扑路径优化。

A.9 可扩展性分析：智能体数量的影响

多智能体规划面临的一项根本挑战是，随着智能体数量（N）增加，可能出现组合爆炸。本节从结构和实证两个角度分析 PCE 的可扩展性，展示该框架如何在智能体数量超过两个的团队中管理复杂度。

通过语义聚合实现结构效率。 直接从所有智能体的完整联合状态中提取不确定性，会导致决策树构建的组合复杂度激增。相比之下，我们的组合器旨在仅从规划器的推理轨迹、观测和记忆中提取不确定性。由于规划器本身会过滤无关细节，聚焦于显著的不确定性，因此组合器的搜索空间随不确定性本身的复杂度变化，而非随智能体数量线性增长。

我们观察到，大语言模型在对多个协作者进行推理时，会自然地采用语义聚合。规划器倾向于将其抽象为集体谓词（例如“是否有任何 协作者检查过厨房？”），而不是为每个智能体分别生成假设（例如“智能体 A 检查过吗？”与“智能体 B 检查过吗？”）。由于决策树基于这些聚合后的语义假设构建，即使智能体数量增加，最大树深（例如 $D = 3$）仍足以描述关键不确定性，而不会使问题变得难以处理。

实证验证（$N = 3,\ 4$）。为实证验证这一可扩展性，我们扩展了在 C-WAH 基准测试上的实验，纳入了包含 3 个和 4 个智能体的场景，同时保持树深度超参数不变（$D = 3$）。

表13：PCE 在 C-WAH 上随智能体数量增加的性能。随着智能体数量增加，总步数减少，表明协作有效，且没有增加规划开销。

智能体数量（N）	**总步数（↓）**
$N = 2$	42.76
$N = 3$	34.60
$N = 4$	28.50

如表13所示，完成任务所需的平均总步数随着智能体数量从 2 个增加到 4 个而单调递减。这一改进表明，PCE 能够成功协调更大团队中的劳动分工，而不会受到规划复杂度的拖累。该框架有效地将人力增加转化为任务效率提升。

局限性。虽然 PCE 能够稳健地扩展至中小规模团队（最多 $N = 4$），但我们承认，在涉及大量智能体或异构智能体之间存在严格顺序依赖关系的任务的场景中，简单的语义聚合可能会受到限制。此类极端规模扩展条件留待未来工作探索。

A.10 LLM生成的评分的准确性与稳定性

PCE 的一个关键组成部分是评估器为生成的场景赋予可靠评分的能力。为定量验证这些大语言模型生成的评分的稳定性，并回应对 LLM 作为评判者可靠性的担忧，我们开展了定量评估。

实验设置。 我们从 C-WAH 基准测试中的 10 个任务抽取了执行日志。四位熟悉该基准测试机制的领域专家担任标注员。这些专家

获得了与评估器相同的上下文输入和情景树，并被要求在 1 到 5 的离散量表上，手动为三个标准评分：情景可能性、条件增益和执行成本。

量化结果。 我们使用平均绝对误差（MAE）测量人类标注与评估器输出之间的一致性。

表14：在人类专家与大语言模型评估器之间，采用 5 分制计算的平均绝对误差（MAE）。MAE 值较低，表明模型的判断与专家直觉高度一致。

	MAE
情景可能性	0.91
条件增益	1.10
执行成本	0.88

如表14所示，评估器与人类判断高度一致。执行成本 指标的精确度最高（0.88），这可能是因为物理距离和时间都是客观的衡量标准。情景可能性 也表现出较高的一致性（0.91），因为它依据存储在智能体记忆中的任务进度来判断每种场景的合理性。尽管条件增益 的误差略高（1.10），但考虑到在部分可观测设定中估算未来效用本身具有主观性，这一误差仍在可接受范围内。

与模型规模扩展的相关性。 此外，正如我们的主要实验结果所示（见表1、表2），更强的骨干模型（例如 GPT-4o-mini）相较于较小的模型（例如 Gemma），始终能产生更优的规划结果。这一趋势证实，随着底层大语言模型推理能力的提升，评估器评分的精确度也会提高，从而直接带来更高的系统稳定性和更合理的决策。

A.11 组合器模块的可靠性分析

组合器将规划器的自由形式推理轨迹作为输入，将其重构为明确的假设，并利用这些假设构建决策树。因此，其可靠性至关重要：如果组合器抽取的信息有误或产生幻觉，后续的决策树就会失效。为严格评估其可靠性，我们进行了两项互补分析：1）假设层面的定量压力测试；2）决策树层面的专家评估。第一项实验关注的是，即使规划器的推理轨迹含糊不清，组合器识别关键假设的效果如何。第二项实验评估这些假设是否被组织成连贯的决策树结构。

A.11.1 假设层面的定量压力测试

为评估组合器能否可靠地识别关键假设（包括规划器的推理轨迹含糊或不完整的情况），我们针对单个假设节点设计了一项定量压力测试。由于抽取或生成的假设是否正确，本质上取决于语义和上下文，因此仅凭基于规则的指标或环境奖励无法可靠地进行验证。因此，人类专家是唯一可信的判断依据，可用于判定某项假设是否确实反映了规划器的意图，或是否是在歧义上下文中提出的合理假设。本实验的主要结果指标是经人类专家判定为有效的假设所占比例，我们将其称为组合器的命中率。

实验设置 本实验使用了从 10 个 C-WAH 任务中收集的执行日志。对于每条日志，我们修改了组合器的提示词，使每个假设节点都明确标注为以下类型之一：

- **抽取型**：直接基于规划器的推理轨迹得出的假设。

- **生成型**：根据环境、目标和记忆信息新生成的假设。

接下来，四位领域专家根据轨迹清晰度，将每个实例划分为简单、中等或困难。**简单**案例中的假设均有明确陈述，且干扰信息极少。**中等**案例涉及可推断但嵌入在含糊或繁杂推理中的假设。**困难**案例对应含糊不清的轨迹，其中关键假设并未直接提及，需要进行隐式推断或生成假设。

基于这一分类后的数据集，另外三位专家分别对每个假设进行了独立评估。只有当轨迹明确支持**抽取假设**时，才将其视为有效；而**生成的假设**若根据环境、目标、智能体记忆及协作者状态构成了符合上下文的合理假设，则视为有效。最终标签由三位评估者通过多数投票确定。按难度等级分别报告的命中率，量化了组合器的假设与人类判断的一致程度。

表15：组合器在不同难度等级下的假设有效性。

	Easy	中	Hard	总体
命中率（%）	84.3	77.8	76.7	80.6

组合器命中率结果。 表15总结了组合器在不同难度等级下的假设有效性。总体而言，组合器的有效率达到80.6%，即使在困难案例中也保持了76.7%的命中率。尽管从简单到困难，显性线索逐渐减少，隐式推断变得越来越必要，但性能仅逐渐下降，而非崩溃。这一模式表明，组合器并非仅仅生成听起来合理的文本，而是可靠地识别出与任务真正相关的不确定性。它还表明，即使规划器的推理轨迹包含噪声、遗漏或歧义，组合器仍能生成在上下文上适当且与智能体目标一致的假设。总之，本次压力测试的结果表明，即使输入推理轨迹的质量下降，组合器依然保持稳健。

系统级稳定性的补充评估。 此外，由于仅凭假设层面的准确率并不能充分解释系统级稳定性，我们使用同一数据集进行了补充分析，以考察评估器抑制错误假设的有效程度。我们将场景分为两类：专家判定所有假设均有效的场景，以及至少包含一个无效假设的场景。对于每个场景，我们比较了评估器给出的情景可能性 和最终得分。

表16：包含和不包含无效假设的场景的评估器得分。

	情景可能性	最终得分
仅包含有效假设	3.28	3.63
包含无效假设	2.85	2.51

表16报告了评估器为每组生成的平均情景可能性和最终得分。包含无效假设的场景在这两个指标上都出现了明显下降。情景可能性从3.28降至2.85，表明评估器能够检测到底层知识状态中的不一致或矛盾。最终得分则出现了更大幅度的下降，从3.63降至2.51，这反映出评估器在动作选择过程中主动抑制了这些有缺陷的场景。这些结果表明，PCE并不是一个依赖于组合器完美提取假设的脆弱系统。相反，评估器充当了结构安全层，防止非预期错误传播到最终动作决策中，从而维持整体性能的稳定。

表17：对由 Composer 生成的决策树进行专家评估（7 点李克特量表）。

指标	组合器	人类	评估重点
Q1：提取准确率	6.27	6.64	抽取出的假设是否确实存在于轨迹中?
Q2：生成能力	6.39	6.73	当轨迹含糊不清时，是否会恰当地生成新的假设?
Q3：排序逻辑	5.83	6.23	节点是否按不确定性降低和影响排序?
Q4：逻辑一致性	6.18	6.55	Are 是否存在 矛盾 in the 决策树?
Q5：行动适切性	5.98	6.15	叶节点行动是否符合场景路径?

A.11.2 决策树层级的专家评估

尽管假设层面的分析展示了组合器构建单个节点的准确程度，但这本身并不能保证这些节点在组装成完整决策树后能够形成连贯的策略。这一局限性在多智能体规划中尤为重要：即使每个假设在局部上都是正确的，如果假设的优先级设置不当、位于不一致的抽象层级，或与相应动作不匹配，智能体仍可能选择次优动作。为评估这一风险，我们将组合器生成的决策树与人类专家构建的决策树进行比较，以评估前者在结构和逻辑上的质量。

使用压力测试中采用的 10 个 C-WAH 任务的相同执行日志，四位领域专家根据相同的输入（环境状态、目标和规划器的推理轨迹）独立构建了决策树。随后，我们将这些人工生成的决策树汇总，形成 DecisionTreesDataset，并使用 7 点李克特量表，围绕五个评估维度对该数据集进行了盲法交叉评审。

结果见表17。尽管预计人类专家的总体得分会略高，但组合器在所有维度上的得分均达到5.83或以上，表明其质量水平总体上与专家表现相当。生成能力和逻辑一致性方面的优异结果表明，组合器不仅忠实地反映了规划器的推理，还能在必要时提出新的假设，同时与先前确立的假设保持一致。排序逻辑得分表明，组合器能够有效地优先处理关键不确定性，从而在决策过程中高效降低不确定性。行动适切性方面的结果进一步表明，组合器能够正确理解各场景路径的语义，并为相应的叶节点分配恰当的动作。

A.12 提示模板

通信

我是 $AGENT_NAME$。我的协作者 $OPPO_NAME$ 刚刚发来一条消息，我必须简洁、得体地回复
以便我们围绕共同目标保持协同。
目标：$GOAL$
我的进度：$PROGRESS$
对话历史：
$DIALOGUE _历史$
先前操作：$ACTION_HISTORY$
来自 $OPPO_NAME$ 的上一条收到的消息：
$LAST _消息$

首先，直接回答上一条消息中的任何明确问题。

然后，提供恰好一条简短的后续说明（状态、位置或确认），以最大程度降低与之最相关的不确定性
目标（例如，物品在哪里、谁去拿、某个房间是否已检查）。

若作出或确认承诺，请使用完整记法 <名称> (标识符) 指称特定物体，仅当该确切
标识符已从上下文中得知；否则只使用名称，不要编造标识符。

保持简短。礼貌、以动作为导向且不含歧义。

输出格式：严格按照以下格式返回一条动作字符串：
[发送_消息] <'...'>
请勿在前后添加任何额外文本。

回答：

图7：用于通信的提示词模板。

规划器

我是$AGENT_NAME$。我很着急，想和朋友$OPPO_NAME$一起完成家务。根据我们的
共同目标、对话历史、我的进度和先前操作，请帮我选择最佳的可用动作来
尽快实现目标。不要选择与已实现或
对手很可能已经实现的目标相关的动作。请注意，我一次可以拿两个物体。所有物体都表示为
<名称> (标识符)，例如 <表格> (712)。
目标：$GOAL$
进度：$PROGRESS$
对话历史：
$对话 _历史$
先前操作：$动作_历史$
可用动作：
$可用 _动作$
答案：

图8：用于规划器的提示词模板。

组合器

我是 $AGENT_NAME$。我正在行动前检查我的推理轨迹中的不确定性。推理轨迹 = 规划器在上一规划步骤中用于选择下一动作的内部思维链。目标：$GOAL$ 进度：$PROGRESS$ 对话历史：$DIALOGUE HISTORY$_先前操作：$ACTION_HISTORY$ 可用动作：$AVAILABLE ACTIONS$_推理轨迹：$ REASONING TRACE$_

你的任务是将推理轨迹转化为更深的思维树，枚举主要不确定性（包括多智能体情形），其中每个叶节点恰好对应一个下一动作（从可用动作中逐字复制）。评估器将对每个场景叶节点的（可能性、表现）与成本（距离/通信）进行评分，以选出最佳动作。

规则：。
1 识别推理轨迹中会实质影响动作选择的不确定陈述。- 不确定的= 假设、猜测、推测，或可能因协作者先前动作而失效的信息。- 示例：“苹果可能在厨房里。”或“Bob 可能已经检查过橱柜。”

2. 对每个不确定的陈述，创建一个二元分支，并将树的最大深度限制为 3。- “True”：假设正确。- “False”：假设不正确。

3. 按以下顺序扩展树：(i) 不确定性更高；(ii) 对目标达成、步数或令牌使用量的预期影响更大。

- 纳入多智能体情境中的第三种情况（例如 .g.，物体确实存在，但已被协作者拿走，而我的信息已过时).
- 对每个分支，继续处理下一个不确定因素；如果没有剩余的不确定因素，则附上且仅附上一个逐字复制的动作字符串 tim 来自可用动作。

4. 仅使用可用动作，为叶节点指定具体动作。- 不要编造新物体/标识符。只能使用推理轨迹、进度或可用动作中出现的实体。- 即使可用动作中没有列出，你也可以通过 [send_message] <'...'> 与协作者通信。所有其他动作都必须来自可用动作。- 如果两个不同的场景导向相同的动作，请保留独立的叶节点（评估器仍需要不同的场景来进行期望值计算）。

5. 仅输出表示场景树的嵌套 JSON 对象（不要添加其他文本）。- 键为 = 不确定陈述（或选定的动作）。- 值为 = 包含“True”和“False”分支的字典，或直接的动作字符串。- 每个动作都必须保留完整格式：<name> (id)，并逐字复制。

示例（与推理轨迹和可用动作配套）：

推理轨迹示例：
- 你在浴室里，手上什么也没拿。目标：将 3 把餐叉放入洗碗机。- 距离——厨房：7.38 米，客厅：13.89 米，卧室：8.39 米。- 检查浴室柜的代价最低（3.35 米），而且里面可能有一把叉子。

可用动作：
- [检查] <浴室柜> (190) - 我的代价：3.35 米 - [探索] <厨房> (11) - 我的代价：7.38 米 - [探索] <客厅> (267) - 我的代价：13.89 米 - [探索] <卧室> (172) - 我的代价：8.39 米 - [发送_消息]

```
示例输出（有效 JSON）：
{
 'type': '假设',
 'text': '<浴室柜> (190)内有一把<餐叉>',
 'reason': '附近的低成本容器；餐具可能存放在此处。',
 'True': {
  'type': '动作',
  'action': '[检查] <浴室柜> (190) - 我的成本：3.35 米'
 },
 'False': {
  'type': '假设',
  'text': 'Bob 已经从<浴室柜> (190)拿走了一把<餐叉> ',
  'reason': '缺乏关于协作者过去动作的观测；物体可能早些时候就已被取走。',
  'True': {
   '类型': '动作',
   '动作': '[发送_消息] <\'Bob，你已经从浴室柜里拿了一把叉子吗？\'>'
  },
  'False': {
   '类型': '假设',
   'text': '叉子在<厨房> (11)（比其他房间更有可能）',
   'reason': '餐具通常在厨房里；到厨房的距离（7.38米）比到客厅（13.89米）短。',
   'True': {
    '类型': '动作',
    '动作': '[探索] <厨房> (11) - 我的代价：7.38 米'
   },
   'False': {
    '类型': '假设',
    '文本': '下一个最适合探索的房间（按距离）是 <卧室> (172)，而不是 <客厅> (267)'
    'reason': '卧室成本 (8.39m) < 客厅成本 (13.89m)；两者均未探索。',
    'True': {
     'type': '动作',
     'action': '[探索] <卧室> (172) - 我的成本：8.39 米'
    },
    'False': {
     'type': '动作',
     'action': '[探索] <客厅> (267) - 我的成本：13.89 米'
    }
   }
  }
 }
}
```

图 9：组合器的提示词模板。

评估器

```
我是$AGENT_NAME$。我收到了检查器生成的场景树。每个叶节点都对应于特定假设集合（包括协作者已
拿走物体或信息过时等多智能体情形）下的一个具体下一步动作选择。


你可能会收到的树形格式：- 假设节点：{'type': 'hypothesis', 'text': '<假设>', 'reason': '<为何
如此认为>', 'True': {..}, 'False': {..}} - 动作节点（叶节点）：{'type': 'action', 'action': '[动作_
字符串]'}



你的任务：评估所有可执行的叶节点（每个包含具体动作的叶节点），并返回一个简洁的映射，供
选择器进行排序。对于每个场景（叶节点），请提供：

1. 可能性（1–5）：
  - 该分支（一组假设）为真的可能性有多大。
  - 1 = 极不可能，5 = 极有可能。
```

```
2. 收益（1–5）：
  - 在该分支下，此动作对推进/满足目标的程度。
  - 5 = 直接满足或最具影响的下一步；3 = 中等进度；1 = 边际。

3. 成本惩罚（1–5）：
  - 执行此动作的成本、风险或负担有多大（就精力、时间或通信开销而言）。
  - 考虑物理移动、通信和资源使用
  - 1 = 极低成本：通信（[发送_消息]），或移动距离小于 3 米
  - 3 = 中等成本：移动距离小于 10 米，或组合多个小动作。
  - 5 = 极高成本：跨越多个房间进行长距离移动（>15米）、大量消耗资源，或采取可能造成浪费的动作
耗费大量时间/精力。

4. 动作：
  - 从可用动作中准确选择一个与此场景分支相对应的动作。
  - 逐字复制完整的动作字符串（例如，'[gograb] <苹果> (36)'）。如果叶节点已包含具体动作
字符串，则使用该字
符串。

约束：
- 仅使用出现在可用动作中的动作（必须完全匹配；不得虚构物体/标识符）。- 你可以通过 [send_
message] <'...'> 与协作者通信，即使该操作未列在可用动作中。- 评估每个包含动作的叶节点。如果场景
树有 N 个可执行叶节点，则返回 N 个场景。- 按上述定义提供可能性、收益和成本惩罚。系统将计算性能。-
按固定顺序为场景编号（例如，从左到右进行深度优先遍历）：“ 场景 1”、“ 场景 2”、 .....

输入：
目标：$GOAL$
进度：$PROGRESS$
对话历史：
$对话          _历史$
先前操作：$动作_历史$
可用动作：
$可用          _动作$
场景树：
$SCENARIO  _TREE$

输出格式：
返回一个 JSON字典（不含额外文本），其中：
- 键 = 情景标识符（例如，“场景1”、“场景2”.....）。
- 值= 一个字典，包含：
 - '可能性'：整数（1–5）
 - '收益'：整数（1–5）
 - '成本惩罚'：整数（1–5）
 - '动作'：字符串

示例输出：
{
 '场景1': {
  '可能性': 4,
  '收益': 5,
  '成本惩罚': 3,
  '动作': '[gograb] <餐叉> (373)'
 },
 '场景2': {
  '可能性': 3,
  '收益': 4,
  '成本惩罚': 5,
  '动作': '[检查] <厨房橱柜> (75)'
 },
 '情景 3': {
  '可能性': 4,
  '收益': 2,
  '成本惩罚': 1,
  '动作': '[发送_消息] <"Bob，你检查过厨房了吗？">'
 }
}
```

图10：评估器的提示词模板。

A.13 大语言模型的使用

本文有限度地使用了大语言模型（LLM）作为辅助工具，以润色措辞并提升可读性。研究构思、方法设计、实验和分析等核心工作均由作者独立完成。作者对本文的所有内容负全部责任。