

# 基于 Casevo 框架的智能体决策能力优化研究

## Tree of Thought 在社会模拟中的应用与评估

王宇东（组长）、陈文远

人工智能与社会课程

2025 年 12 月

# 报告大纲

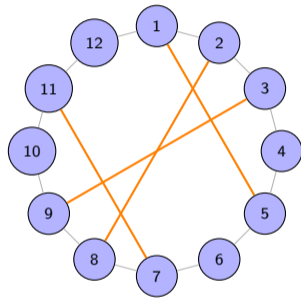
- ① 研究背景
- ② 研究方法
- ③ 实验结果
- ④ 综合分析
- ⑤ 成本效益分析
- ⑥ 结论与展望

## Casevo 框架

- 基于 LLM 的多智能体社会模拟框架
- 支持选举投票、资源分配、信息传播等场景
- 集成 CoT 推理、RAG 记忆、反思机制

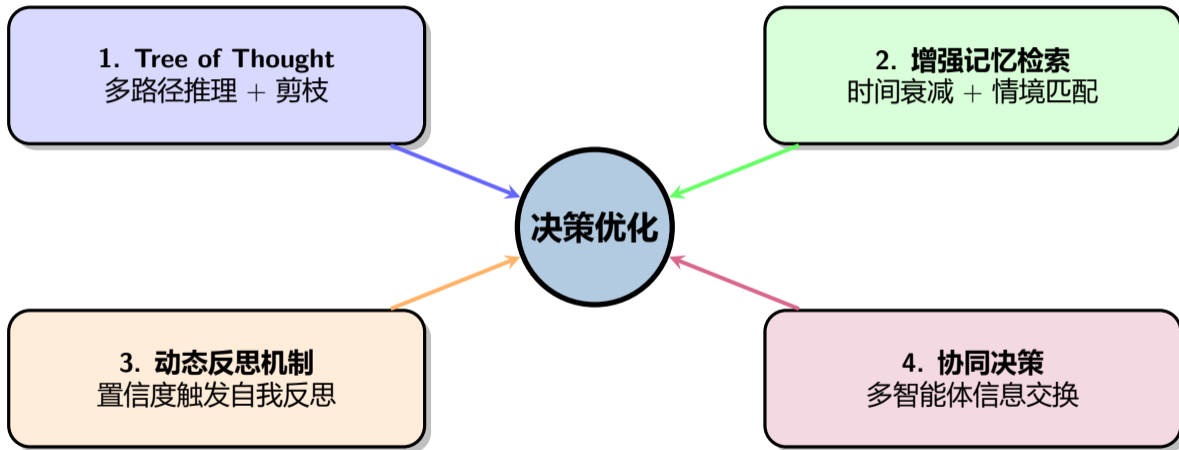
## 存在的问题

- 线性 CoT 在复杂决策中波动大
- 记忆检索效率和准确性有限
- 多智能体协同机制不完善



小世界网络拓扑

# 四项优化策略概览



# 策略 1: Tree of Thought 实现

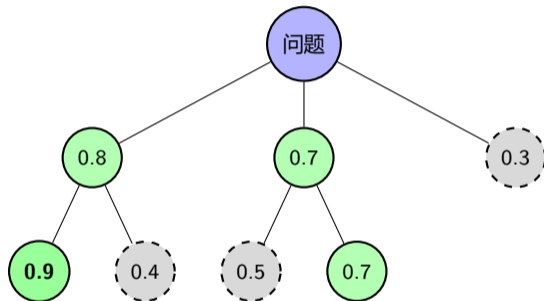
**核心思想：**将线性推理扩展为树状探索

**实现细节：**

- 1 **分支生成：**LLM 生成多个候选推理路径
- 2 **分支评估：**LLM 对每个路径打分 (0-1)
- 3 **Beam Search：**保留 Top-K 高分路径
- 4 **剪枝：**丢弃低于阈值  $\theta$  的路径
- 5 **迭代：**重复直到达到最大深度

**关键参数：**

- 最大深度  $D = 3$
- 束宽度  $K = 3$
- 剪枝阈值  $\theta = 0.6$



**每层：**生成 → 评估 → 保留 Top-K

# 策略 1: ToT 伪代码

## ToT 推理算法

**输入:** 问题  $P$ , 深度  $D$ , 束宽  $K$ , 阈值  $\theta$     **输出:**  $d^*$

1.  $\mathcal{B}_0 \leftarrow \{(\text{root}, 1.0)\}$
2. **for**  $d = 1$  **to**  $D$ :
3.     $\mathcal{C} \leftarrow \emptyset$
4.    **for**  $(n, s) \in \mathcal{B}_{d-1}$ :
5.      $B \leftarrow \text{LLM\_Gen}(n, P)$
6.     **for**  $b \in B$ :
7.        $s_b \leftarrow \text{LLM\_Eval}(b)$
8.       **if**  $s_b \geq \theta$ :  $\mathcal{C} \cup = \{(b, s_b)\}$
9.     $\mathcal{B}_d \leftarrow \text{TopK}(\mathcal{C}, K)$
10. **return**  $\arg \max \mathcal{B}_D$

## 关键点

- 每次 LLM 调用都有成本
- 深度  $D$  和分支数决定调用次数
- 阈值  $\theta$  控制探索广度

## LLM 调用次数

$$O(K \times B \times D)$$

$K$ = 束宽,  $B$ = 分支数,  $D$ = 深度

## 策略 2：增强记忆检索

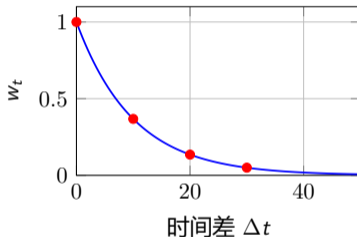
**问题：**原始 RAG 仅用语义相似度

**改进：**多因素加权检索

$$\text{score}(m) = \text{sim}(q, m) \cdot w_t \cdot w_c$$

- $\text{sim}(q, m)$ ：语义相似度
- $w_t = e^{-\lambda \cdot \Delta t}$ ：时间衰减
- $w_c$ ：情境匹配度

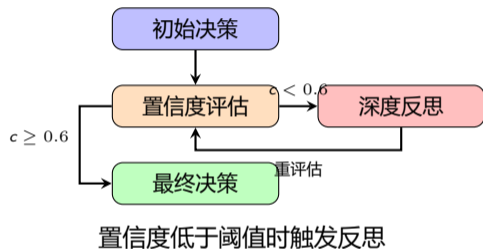
**实现步骤：**嵌入 → 相似度 → 时间衰减 ( $\lambda=0.1$ )  
→ 情境匹配 → Top-K



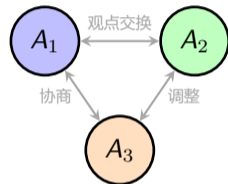
$w_t = e^{-0.1 \cdot \Delta t}$ ，近期记忆权重更高

## 策略 3-4：动态反思 & 协同决策

### 3. 动态反思机制



### 4. 协同决策



智能体间迭代协商达成共识

**注：**实验结果显示，动态反思和协同决策的效果有限，详见后续分析

场景	Agent 数	轮数	核心挑战
选举投票	30	6	社会影响下的态度演化
资源分配	20	$\leq 5$	快速收敛到公平分配
信息传播	50	10	阻止虚假信息扩散

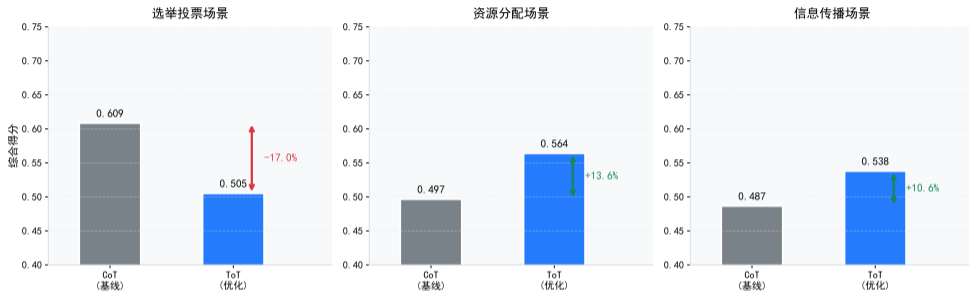
**实验规模:** 5 组配置  $\times$  3 次运行  $\times$  3 场景 = **45 次实验**

## 五组实验配置:

- baseline\_cot: 原始 CoT
- tot\_only: 仅启用 ToT
- tot+memory: ToT + 增强记忆
- tot+reflection: ToT + 动态反思
- full: 全部启用

# 核心发现：ToT 效果总览

ToT 多层次推理 vs CoT 线性推理 效果对比



✓ 资源分配  
+13.6%

✓ 信息传播  
+10.6%

✗ 选举投票  
-17.0%

## 关键发现

ToT 具有**场景敏感性**——多路径推理并非在所有场景都能带来收益！

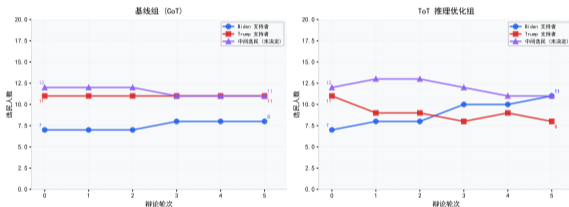
# 选举投票场景：推理能力 vs 综合得分

指标	CoT	ToT
推理能力得分	0.43	<b>0.95</b>
综合得分	<b>0.609</b>	0.505
推理深度	2	5
分支探索数	1	40
连贯性分数	0.41	0.82

## 矛盾现象

- 推理能力 +121%
- 综合得分 -17%

选举投票态度演化：三类选民变化对比



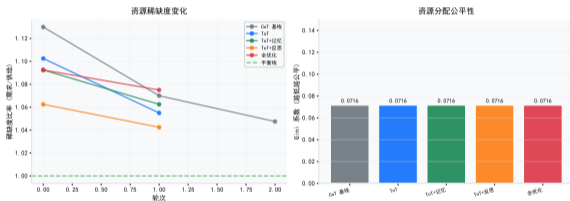
选民态度演化对比

# 资源分配场景：ToT + 记忆最优

配置	收敛轮数	标准差
baseline_cot	3.00	0.00
tot_only	2.33	0.58
<b>tot+memory</b>	<b>2.00</b>	<b>0.00</b>
tot+reflection	3.00	1.00
full	2.33	0.58

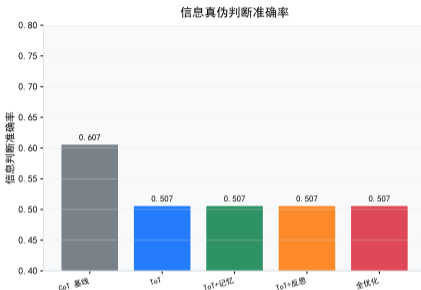
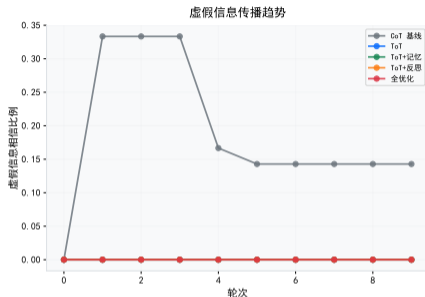
## 最优配置

**ToT + 增强记忆**：收敛最快（2 轮）、稳定性最高（ $\sigma = 0$ ）



## 资源分配收敛分析

# 信息传播场景：ToT 的过度保守行为



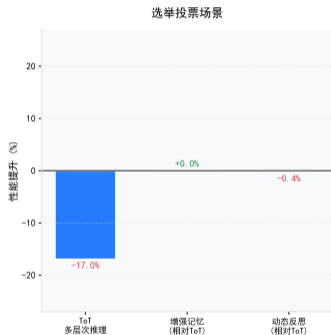
指标	CoT	ToT
虚假信任率	11.6%	0%
整体准确率	60.7%	50.7%
信息传播数	7	0

## 过度保守

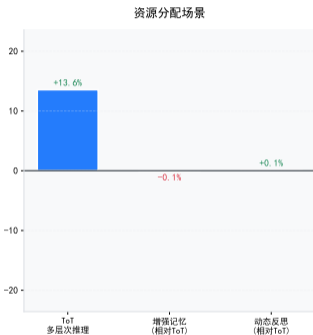
ToT 不仅阻止了虚假信息，  
也阻止了**所有真实信息**的传播！  
“宁可错杀一千，不可放过一个”

# 组件贡献度分析

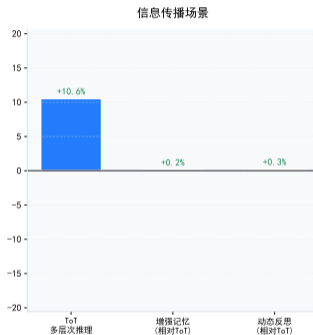
各优化组件的性能贡献分析



**ToT**  
核心贡献者

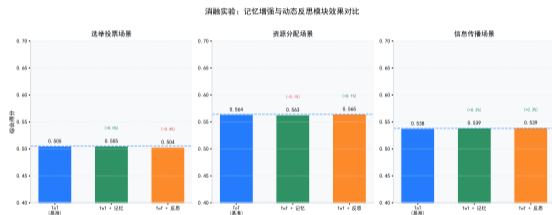


**增强记忆**  
场景依赖



**反思/协同**  
效果有限

# 消融实验：组件叠加效应



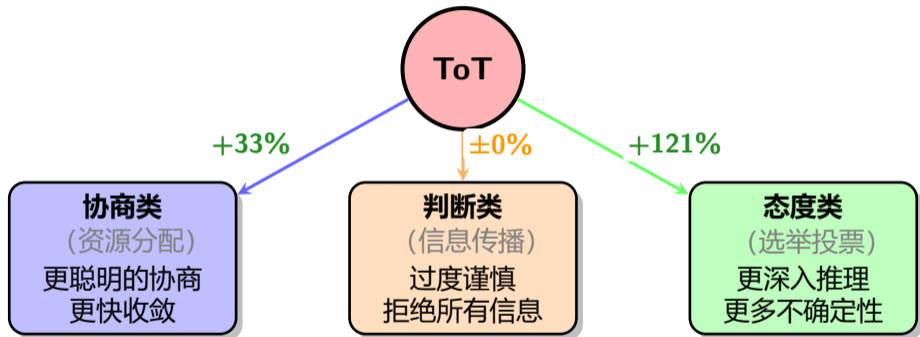
场景	tot_only	full
选举投票	0.505	0.504
资源分配	2.33 轮	2.33 轮
信息传播	50.7%	50.7%

意外发现

组件叠加无协同效应

$$1 + 1 \leq 2$$

# ToT 的场景行为模式



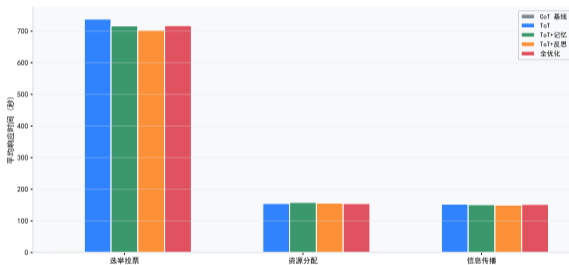
# 计算成本对比

场景	CoT	ToT
选举投票	3.3s	747s (227×)
资源分配	2.6s	155s (60×)
信息传播	1.9s	152s (80×)

## 成本来源

- ToT 需要多次 LLM 调用
- 分支生成 + 分支评估
- 深度越大，成本越高

各优化组件计算效率对比



各配置响应时间对比

# 配置推荐

场景	baseline	tot_only	tot+memory	full
选举投票	○	✓	✓✓	○
资源分配	○	✓	✓✓	○
信息传播	✓	△	△	△

✓✓: 强烈推荐    ✓: 推荐    ○: 可用    △: 谨慎使用

**推荐默认配置:** tot\_memory —效果最优、成本适中

## ✓ 验证成功

- ToT 提升推理深度 +150%
- ToT 提升推理连贯性 +100%
- 增强记忆加速协商收敛 +33%

## ✗ 未验证

- 动态反思提升决策质量
- 协同决策改善社会效应
- 组件叠加产生协同效应

## ⚠ 意外发现

- ToT 在判断类任务中**过度保守**
- 动态反思引入**不稳定性**
- ToT 具有**场景敏感性**

## 核心贡献

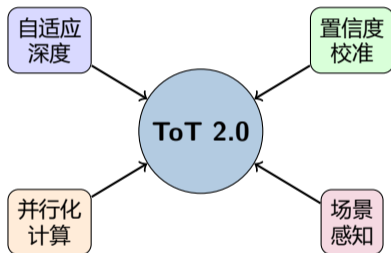
揭示了 ToT 的场景敏感性和过度保守行为，为后续研究提供实证依据

## 研究局限性

- 每组仅 3 次运行，统计显著性有限
- Agent 数量有限 (30/20/50)
- 依赖特定 LLM 模型版本
- 仅测试三种场景类型

## 未来工作

- 自适应 ToT: 动态调整深度
- 置信度校准: 避免过度保守
- 并行化优化: 降低计算成本
- 更多场景验证




## Tree of Thought 是最有效的单一优化策略

在资源分配和信息传播场景显著提升决策质量

但需注意其**场景敏感性**和**过度保守行为**

**推荐配置:** `tot_memory`

**谢谢！ 欢迎提问**

 代码仓库: [github.com/xxx/casevo-optimization](https://github.com/xxx/casevo-optimization)