Beyond Task Success: Evaluating Cooperation in LLM-Based Multi Agent Systems

Ashish Raj Shekhar*, **Saniya Mulla**, **Upasana Biswas**, **Priyanuj Bordoloi**, **Vivek Gupta***
{ashekha9, smulla1, ubiswas2, pbordolo, vgupt140}@asu.edu
Arizona State University
*Corresponding authors.

Abstract

Large language models are increasingly deployed as autonomous agents in multi-agent environments, but task completion alone may not indicate reliable strategic coordination. We study this issue in Collab-Overcooked, a cooperative benchmark where LLM agents coordinate through natural language under role asymmetry and verifier-guided execution. We model each rollout as the realized outcome of a finite-horizon cooperative Markov game and evaluate whether successful teams also exhibit efficient coordination: low redundancy, timely partner response, useful signaling, and robustness to partner identity. We extend Collab-Overcooked with non-required-cooperation layouts in which either agent can complete the task alone, allowing collaboration to be evaluated as an efficiency strategy rather than a feasibility requirement. We introduce a trace-grounded evaluation framework combining structured interdependence with Trace-Grounded Communication Outcome auditing. Across three LLM families and directed role pairings, we find that high task success often coexists with substantial coordination cost, including redundant actions, delayed or unfulfilled requests, verifier corrections, and strong role-pairing effects. Theory-of-Mind scaffolds improve some metrics but do not reliably eliminate these failures. Our results suggest that, in cooperative benchmarks of the Collab-Overcooked family, completion-based evaluation is insufficient and that trace-grounded coordination metrics are needed to measure strategic behavior; the metrics we introduce (TGCO, structured interdependence, and the NRC distinction) are defined to transfer beyond this benchmark.

1. Introduction

LLMs are increasingly used as agents that interact with tools, humans, and other agents. In zero-shot collaboration (ZSC), agents must coordinate with unfamiliar partners without joint task-specific training. This setting is central to human-AI teaming and multi-agent automation, where deployment quality depends not only on completing the task but also on how agents divide labor, respond to requests, repair mistakes, exploit resources, and adapt to partners; however, task success can mask collapsed coordination patterns in which agents stop adapting to their partner [4]. Benchmarks such as Hanabi-style belief-sensitive games [2, 11] and Overcooked-style cooperation [5, 15] show that LLM agents achieve high success rates which are often interpreted as evidence of planning or collaborative reasoning.

However, LLM agents are boundedly rational; their policies emerge from in-context inference and alignment patterns rather than utility maximization, and they cannot compute equilibria. This means task success does not imply strategic stability, and high reward does not rule out coordination waste. Worse, if all deployed agents share the same model, apparent coordination in self-play may

© A. R. Shekhar, S. Mulla, U. Biswas, P. Bordoloi & V. Gupta.

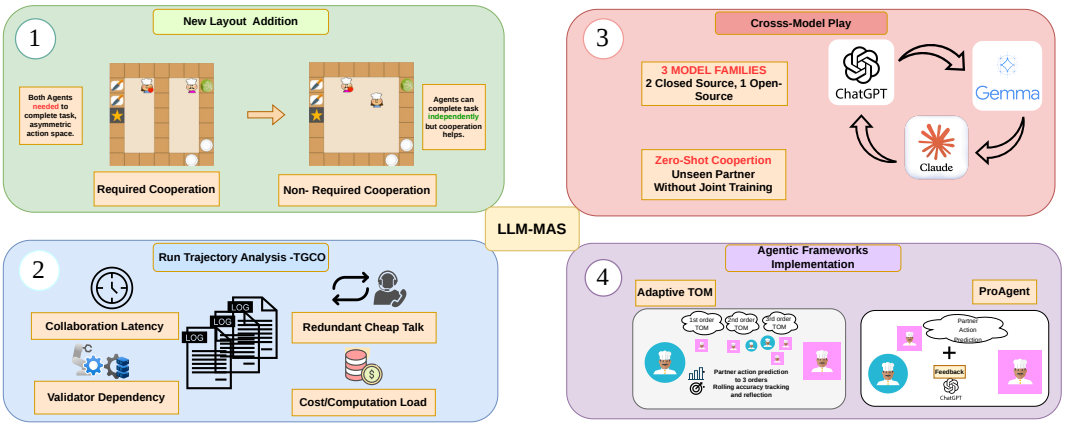

Figure 1: Overview of methodology and evaluation pipeline. 1) Addition of new Non-Required Cooperation layout 2) Cross-Model Family analysis for ZSC across open- and closed-source LLM 3) TGCO audit from run trajectory logs 4) Adaptive-ToM and ProAgent architectures.

reflect stylistic alignment rather than robust strategic reasoning, a form of algorithmic monoculture that breaks down when partners change [4, 9, 16]. Recent work also argues that temporally extended partner modeling remains under-measured in current LLM-based Theory-of-Mind frameworks and benchmarks [13].

We study this gap in Collab-Overcooked [15], where two LLM agents complete cooking tasks under full observability, asymmetric role capabilities, natural-language communication, and validator-guided action execution. The original benchmark usefully measures success, trajectory efficiency, intermediate-task efficiency, initiation capability, and response capability. We ask a complementary strategic question: when agents succeed, do their messages and actions form efficient partner-sensitive joint strategies?

We interpret a rollout as the realized outcome of a cooperative Markov game. A joint policy profile can be task-successful yet inefficient if another feasible profile achieves the goal with lower completion time, fewer verifier calls, fewer redundant actions, or lower communication cost. Communication is treated as a cheap-talk signal with strategic value only when it changes partner behavior in the trace, a condition TGCO audits directly.

Our contributions are concise:

- We introduce non-required-cooperation variants of Collab-Overcooked to separate cooperation required for feasibility from collaboration that emerges as an efficiency strategy.
- We propose trace-grounded metrics for strategic coordination: structured interdependence for directed enabling and TGCO for communication outcomes.
- We evaluate self-play and directed cross-model play across three LLM families, showing that role assignment and partner identity strongly affect coordination cost even when success remains high.

- We test whether explicit Theory-of-Mind scaffolds improve coordination and find mixed gains: belief tracking helps some metrics but often fails to produce timely partner-useful actions.

Our methodological novelty is concentrated in the trace-grounded metrics (TGCO, structured interdependence) and the NRC benchmark variant; the Pro-Agent and AToM comparison is an empirical reuse of existing partner-modeling scaffolds rather than a new method.

2. Related Work

LLM agents and multi-agent benchmarks. LLM agents have been evaluated in tool-use, code execution, API use, browsing, and structured environments [17, 18]. Benchmarks including LLM-Hanabi [11], and Collab-Overcooked [15] measure cooperative task performance. Our work focuses on the strategic structure hidden underneath task success: directed dependence, signal follow-through, and partner robustness.

Evaluating collaboration quality beyond task success. A growing line of work argues completion rate is an insufficient measure of multi-agent collaboration. MultiAgentBench [23] introduces milestone-based KPIs for collaboration quality, and MAST [6] builds a trace-annotated taxonomy of multi-agent failure modes. LLM-Coordination [1] and Tool-RoCo [21] probe coordination under tool use and role structure, and our monoculture concern connects directly to Kleinberg and Raghavan's analysis of algorithmic monoculture and social welfare [9]. We complement these with message-level, trace-grounded outcome auditing (TGCO) and the NRC distinction between feasibility-required and efficiency-motivated cooperation.

Interdependence and zero-shot cooperation. Zero-shot cooperation asks whether agents can collaborate with previously unseen partners [8, 14, 19, 22]. Constructive interdependence [4] evaluates who is helping whom in human-agent teams by mapping Markov games to symbolic traces and identifying useful dependencies. We add a communication audit and adapt this idea to LLM agents because the coordination in LLM agents occurs through natural language.

Theory of Mind in LLM agents. Theory of mind is the ability to reason about another agent's beliefs, goals, or likely actions [3]. LLMs perform well on some isolated ToM tests [7, 10], but static tests do not establish robust interactive coordination. We evaluate two ToM-oriented scaffolds, Pro-Agent [20] and Adaptive Theory of Mind (AToM) [12], in these temporally extended collaborative rollouts.

3. Method

3.1. Problem Formulation

We model Collab-Overcooked as a finite-horizon, two-player cooperative Markov game

$$\mathcal{G} = \langle \mathcal{S}, \mathcal{A}_1, \mathcal{A}_2, P, R, \gamma, T \rangle,$$

where $\mathcal{S}$ contains kitchen state, role state, active recipes, held objects, station contents, order progress, and verifier state. At each timestep, agent i observes a textual state description, may send a message m_i^t, chooses an action $a_i^t \in \mathcal{A}_i(s_t)$, and executes a joint transition. Role asymmetry induces $\mathcal{A}_1(s) \neq \mathcal{A}_2(s)$.

An LLM agent induces a history-dependent policy $\pi_i(a_i^t, m_i^t \mid h_i^t)$, so a pair (π_1, π_2) is a strategy profile. We evaluate profiles using a cost-sensitive cooperative objective

$$U(\pi_1, \pi_2) = \mathbb{E}\Big[\sum_{t=0}^{T} \gamma^t R_t - \lambda_1 C_{time} - \lambda_2 C_{tok} - \lambda_3 C_{ver} - \lambda_4 C_{red}\Big],$$

where C_{time} is completion time, C_{tok} token cost, C_{ver} verifier burden, and C_{red} redundant action cost. We use U as an *expository* objective that organizes the observable cost correlates measured in §3.2 ($C_{time}, C_{tok}, C_{ver}, C_{red}$) and makes the trade-off between task reward and coordination cost explicit; we do not instantiate the weights λ_i or optimize U, and we treat it as a scaffold for interpreting coordination cost rather than a quantity we estimate. The metrics in §3.2 measure observable correlates of each cost term.

Cooperative Game Settings Task structure determines whether collaboration is the only feasible solution (RC) or a chosen efficiency strategy (NRC). Let G denote task completion and $C(\pi)$ an execution cost. An RC task is one where no single-agent policy completes $\mathcal{G}$ with positive probability while some joint policy does: $\forall \pi_i,\ \mathbb{P}(\mathcal{G} \mid \pi_i) = 0$, and $\exists(\pi_1, \pi_2)$ with $\mathbb{P}(\mathcal{G} \mid \pi_1, \pi_2) > 0$. An NRC task admits a solo solution but joint execution can lower cost: $\exists \pi_i$ with $\mathbb{P}(\mathcal{G} \mid \pi_i) > 0$ and $\exists(\pi_1, \pi_2)$ with $C(\pi_1, \pi_2) < C(\pi_i, \pi_{\text{idle}})$. We extend Collab-Overcooked level-1 [15] to the NRC setting with symmetric action spaces but asymmetric task knowledge, and separate the agents to enable **cross-model** evaluation of partnership behavior.

3.2. Trace-Grounded Coordination Metrics

Each rollout is converted into a symbolic event trace with timestep, agent identity, action type a, object lineage o, message, verifier feedback, and recipe relevance.

Structured interdependence. Let N_{trig} be candidate trigger actions, N_{int} accepted dependencies, and N_{cons} constructive dependencies:

$$\text{ADR} = \frac{N_{\text{int}}}{N_{\text{trig}}}, \quad \text{IDensity} = \frac{N_{\text{cons}}}{N_{\text{int}}}, \quad \text{MOR} = 1 - \text{ADR}.$$

TGCO communication audit. For each message m_t^i, TGCO parses request units $u = (o, a, j)$ and checks target, relevance to R_t, redundancy, and partner execution in $[t, t + w_l]$. Each parsed, partner-directed request falls into exactly one of {EFFECTIVE, ASSISTED, REDUNDANT, INEFFECTIVE}, with counts $N_{\text{eff}}, N_{\text{assist}}, N_{\text{red}}, N_{\text{ineff}}$. Messages that parse to no partner-directed request unit are labeled UNSTRUCTURED and are excluded from all outcome counts and from the follow-rate denominator (consistent with Alg. 3 and Appendix C); they are tracked only via ParseRate. We report the outcome composition over auditable requests $N_{\text{aud}} = N_{\text{eff}} + N_{\text{assist}} + N_{\text{red}} + N_{\text{ineff}}$, and define follow-through only over requests that were *not* already satisfied before the message (consistent with the decision order in Alg. 3, where REDUNDANT returns before trace lookup, so redundant units never reach the effective/assisted decision):

$$\text{FollowRate} = \frac{N_{\text{eff}} + N_{\text{assist}}}{N_{\text{eff}} + N_{\text{assist}} + N_{\text{ineff}}}, \qquad \text{CommCost} = \frac{\text{tokens}}{N_{\text{int}}}.$$

Table 1: TGCO Outcome Categories

Outcome Category	Definition	TGCO Decision Rule
Effective	The partner completes the requested action within the analysis window, matching an accepted environment action without validator correction.	Parsed request $u = (o, a, j)$ passes target/relevance checks, is not redundant, and partner j performs (o, a) within $[t, t + w_l]$ with no prior correction.
Assisted	The partner completes the requested action, but only after validator feedback or correction. We use *delayed* as a synonym for this category.	Parsed request $u = (o, a, j)$ passes target/relevance checks, is not redundant, and partner j performs (o, a) within $[t, t + w_l]$ after a validator correction.
Redundant	The requested work was already done before the message, making the request unnecessary.	(o, a) is completed in trace T before time t; this is checked before trace lookup, so later repetition does not count as effective.
Ineffective	All unsuccessful or non-auditable cases, including malformed requests, wrong/ambiguous target, irrelevant asks, or missing/mismatched partner action.	No coherent request unit is parsed, the unit fails target/relevance checks, or no valid partner match for (o, a) occurs within $[t, t + w_l]$.

4. Experimental Setup

We evaluate GPT-4o, Claude Sonnet 4, and Gemma-31B in three same-model self-play pairings and six directed cross-model pairings. Directed roles matter: model A as chef with model B as assistant is not equivalent to the reverse. Each pairing is run across three task-complexity levels with 10 episodes per setting. GPT-4o and Claude are accessed through provider APIs; Gemma-31B runs on an H200 server. Full prompt templates appear in Appendix A; protocol details in Appendix C.

What each table reports. Table 2 covers the NRC layouts only. Table 3 macro-averages each model over all pairings in which it appears (self-play and cross-model, either role) and over all three task levels; Table 5 (Appendix B) breaks the *self-play* subset down by task level; the cross-model cost asymmetries in §5 use the directed cross-model pairings only, and Table 4 reports TGCO follow-through (uptake/waste) for the two cited scaffold configurations. For each (pairing, level) cell, Success and ADR are computed over the 10 rollouts; the aggregate tables report point estimates.

Explicit Partner Modeling Pro-Agent predicts the partner's likely next action and chooses anticipatory actions [20]. Adaptive-ToM maintains a rolling belief over partner subgoal, held object, blocked actions, recent latency, fulfillment, and communication style [12]. Both scaffolds add belief context to the decision prompt but do not modify the environment or verifier.

5. Results and Discussion

Across NRC, cross-model, ToM, and TGCO views, a single pattern holds: high success masks low-quality coordination. **NRC.** Both GPT-4o and Claude Sonnet 4 reach 100% solo completion with mean structured interdependence at zero, and team execution adds 25–33% to per-task completion time, so agents do not adopt collaboration as an efficiency strategy when feasibility does not demand

Table 2: RC/NRC performance. NRC tasks permit solo completion, allowing coordination to be evaluated as an efficiency strategy rather than a feasibility condition. **Aggregation:** NRC layouts only; mean $\pm$ std over 10 rollouts per cell.

Model	Setting	Success (%)	Avg Timestep	Solo feasible (%)
GPT-4o	NRC	95 ± 3	120 ± 18	100
Claude Sonnet 4	NRC	100 ± 0	100 ± 12	100
Gemma-31B	NRC	94 ± 4	160 ± 25	100

it. **Cross-model.** Partner identity, not the model alone, sets coordination cost: Claude→Claude consumes 5× more tokens than GPT-4o→Gemma, and follow rates span 0.55 to 0.82 across the same task. Verifier interventions track this noise (9 for GPT-4o self-play, 21 for Claude→GPT-4o and Gemma→GPT-4o; selected counts in Table 6), indicating the validator absorbs much of the apparent task competence and that small open-source models reach near-parity with frontier models largely through this scaffold. **ToM scaffolds.** Pro-Agent and A-ToM reduce redundant actions but do not consistently improve TGCO knowledge uptake: Pro-Agent on GPT-4o achieves the highest uptake (0.50) at the lowest waste ($\sim$0.15), while A-ToM on Sonnet sits at 0.75 waste with the same uptake (Table 4), and both raise per-episode latency and token cost. We also observe a striking failure mode: in GPT-4o self-play, the Pro-Agent partner-action-prediction step *degrades* coordination, with success falling from 0.93–1.00 to 0.30–0.70 and accepted-dependency ratio collapsing to zero at Levels 2–3 (Table 5). We attribute this to the anticipatory-subgoal step overriding otherwise-correct solo execution, which suggests partner-modeling scaffolds can be actively harmful when a capable model would coordinate adequately without them; because Pro-Agent and AToM differ in both mechanism and prompt length/structure, our "mixed gains" reading is a property of these two scaffolds as deployed rather than of ToM scaffolding in general. **TGCO.** Across all settings, the outcome composition over auditable requests is 44% REDUNDANT and 21% EFFECTIVE, with the remaining 35% split between ASSISTED (delayed but fulfilled) and INEFFECTIVE (Table 4); the four shares sum to 100%, and the FollowRate (redundant and unstructured units excluded, per Alg. 3) spans 0.55–0.82. As task complexity grows, agents communicate more but less usefully, with INEFFECTIVE units dominating Level 3 while EFFECTIVE and ASSISTED shrink. Together, success rate is a weak proxy for coordination quality, and trace-grounded metrics surface the redundancy, validator dependence, and signaling waste that completion numbers obscure. We scope these empirical conclusions to the Collab-Overcooked benchmark family—which has a small action space and fixed recipe structure—while the TGCO and NRC instruments themselves are general; classical cooperative solution concepts such as Pareto efficiency and strategy-proof signaling may fail to characterize these boundedly-rational joint policies, motivating new stability notions for agentic AI that we leave to future work.

Limitations. The model panel covers three LLM families and a single benchmark family; we therefore scope our empirical claims to Collab-Overcooked rather than LLM-based multi-agent systems broadly. The evaluated models may contain benchmark-adjacent Overcooked descriptions in pretraining; as a lightweight check (Appendix D) we prompted each model to describe Collab-Overcooked unprompted, but observed coordination may still partly reflect memorized priors rather than novel strategic adaptation. Verifier-guided retries mix interface robustness with collaborative

Table 3: ToM framework comparison across model families. **Aggregation:** macro-averaged over all pairings in which the model appears (1 self-play + 4 cross-model, as either chef or assistant) and over all 3 task levels.

Model	**Framework**	**Success Rate**	**ADR**	**Tokens / Episode**
GPT-4o	Pro-Agent	0.90	0.72	4612.50
GPT-4o	AToM	0.93	0.70	2464.50
Claude Sonnet 4	Pro-Agent	0.99	0.78	8200.24
Claude Sonnet 4	AToM	0.95	0.84	9850.00
Gemma-31B	Pro-Agent	0.87	0.59	4211.20
Gemma-31B	AToM	0.92	0.73	2102.30

Table 4: TGCO follow-through for the two scaffold configurations whose uptake/waste are cited in §5. **Uptake** is the knowledge-directed follow-through share $(N_{\text{eff}} + N_{\text{assist}})/(N_{\text{eff}} + N_{\text{assist}} + N_{\text{ineff}})$ (redundant units excluded); **Waste** is the redundant share $N_{\text{red}}/N_{\text{aud}}$ with $N_{\text{aud}} = N_{\text{eff}} + N_{\text{assist}} + N_{\text{red}} + N_{\text{ineff}}$. Uptake and Waste use different denominators, so they need not sum to 1. Pooled over all settings the outcome composition is EFFECTIVE 0.21, REDUNDANT 0.44, and ASSISTED+INEFFECTIVE 0.35.

Model	Framework	Uptake	Waste
GPT-4o	Pro-Agent	0.50	0.15
Claude Sonnet 4	AToM	0.50	0.75

reasoning, and we do not evaluate human-LLM teams. A controlled ToM study holding prompt length and information content fixed while varying only the partner-modeling mechanism, partner-adaptation tests, payoff perturbations, stronger joint-planning baselines, and verifier-stratified analysis of coordination quality remain future work.

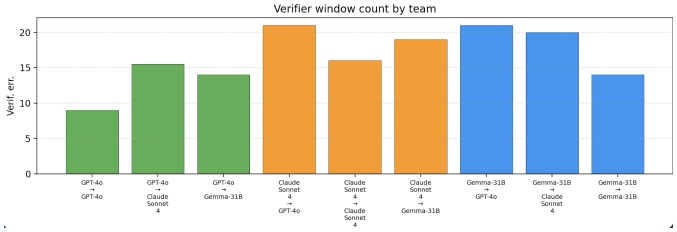

Figure 2: Verifier errors by directed role pairing. Same-model pairs (e.g., GPT-4o→GPT-4o) trigger fewer corrections than cross-model pairs (e.g., Claude→GPT-4o, Gemma→GPT-4o), where validator interventions reach 20–21 per episode. Numerical values are tabulated in Table 6.

References

[1] Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. arXiv:2310.03903.

[2] Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2019.103216. URL https://www.sciencedirect.com/science/article/pii/S0004370219300116.

[3] Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. Does the autistic child have a "theory of mind" ? *Cognition*, 21(1):37–46, 1985. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(85)90022-8. URL https://www.sciencedirect.com/science/article/pii/0010027785900228.

[4] Upasana Biswas, Vardhan Palod, Siddhant Bhambri, and Subbarao Kambhampati. Who is helping whom? analyzing inter-dependencies to evaluate cooperation in human-ai teaming. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(21):17347–17356, Mar. 2026. doi: 10.1609/aaai.v40i21.38787. URL https://ojs.aaai.org/index.php/AAAI/article/view/38787.

[5] Micah Carroll, Rohin Shah, Mark Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

[6] Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. arXiv:2503.13657.

[7] Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, and Minlie Huang. ToMBench: Benchmarking theory of mind in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15959–15983, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.847. URL https://aclanthology.org/2024.acl-long.847/.

[8] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other-play" for zero-shot coordination. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning*

Research, pages 4399–4410. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/hu20a.html.

[9] Jon Kleinberg and Manish Raghavan. Algorithmic monoculture and social welfare. *Proceedings of the National Academy of Sciences (PNAS)*, 118(22):e2018340118, 2021. doi: 10.1073/pnas.2018340118.

[10] Michal Kosinski. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45):e2405460121, 2024. doi: 10.1073/pnas.2405460121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2405460121.

[11] Fangzhou Liang, Tianshi Zheng, Chunkit Chan, Yauwai Yim, and Yangqiu Song. Llm-hanabi: Evaluating multi-agent gameplays with theory-of-mind and rationale inference in imperfect information collaboration game, 2025. URL https://arxiv.org/abs/2510.04980.

[12] Chunjiang Mu, Ya Zeng, Qiaosheng Zhang, Kun Shao, Chen Chu, Hao Guo, Danyang Jia, Zhen Wang, and Shuyue Hu. Adaptive theory of mind for llm-based multi-agent coordination. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(35):29608–29616, Mar. 2026. doi: 10.1609/aaai.v40i35.40204. URL https://ojs.aaai.org/index.php/AAAI/article/view/40204.

[13] Matthew Riemer, Zahra Ashktorab, Djallel Bouneffouf, Payel Das, Miao Liu, Justin D. Weisz, and Murray Campbell. Position: Theory of mind benchmarks are broken for large language models. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025. URL https://openreview.net/forum?id=BCP8UU2BcU.

[14] DJ Strouse, Kevin R. McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. ISBN 9781713845393.

[15] Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. Collab-overcooked: Benchmarking and evaluating large language models as collaborative agents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4922–4951, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.249. URL https://aclanthology.org/2025.emnlp-main.249/.

[16] Mingjie Sun. Hidden Properties of Large Language Models. 7 2025. doi: 10.1184/R1/29316830.v1. URL https://kilthub.cmu.edu/articles/thesis/Hidden_Properties_of_Large_Language_Models/29316830.

[17] Lei Wang, Chao Ma, Xuan Feng, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18:186345, 2024. doi: 10.1007/s11704-024-40231-1.

[18] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin,

Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey. 2023. URL https://arxiv.org/abs/2309.07864.

[19] Chao Yu, Jiaxuan Gao, Weilin Liu, Botian Xu, Hao Tang, Jiaqi Yang, Yu Wang, and Yi Wu. Learning zero-shot cooperation with humans, assuming humans are biased. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TrwE8l9aJzs.

[20] Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang, Feng Yin, Yitao Liang, and Yaodong Yang. Proagent: Building proactive cooperative agents with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17591–17599, Mar. 2024. doi: 10.1609/aaai.v38i16.29710. URL https://ojs.aaai.org/index.php/AAAI/article/view/29710.

[21] Ke Zhang, Xiaoning Zhao, Ce Zheng, Jiahong Ning, Dandan Zhu, Wenqi Zhang, Chen Sun, and Toshiharu Sugawara. Tool-roco: An agent-as-tool self-organization large language model benchmark in multi-robot cooperation, 2025. URL https://arxiv.org/abs/2511.21510.

[22] Kevin Zhao, Daniel Strouse, Haohan Ye, Michael Dennis, Eric Mazumdar, Anca Dragan, and Stuart Russell. Maximum entropy population-based training for zero-shot human-ai coordination. In *AAAI Conference on Artificial Intelligence*, 2023.

[23] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Aoran Wang, Mengdi Wang, and Jiaxuan You. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. arXiv:2503.01935.

Appendix A. ToM-Oriented Scaffold Implementations

This appendix provides pseudocode and prompt templates for the two ToM-oriented scaffolds. Both scaffolds wrap an underlying LLM call with structured belief context derived from the rollout trace; neither modifies the environment or the verifier.

A.1. Pro-Agent

Algorithm 1 Pro-Agent decision step for agent i at time t

Require: Environment state s_t, dialogue history H_t, valid action schema $\mathcal{A}_i(s_t)$, partner index j

1: $b_j^t \leftarrow \textsc{InferPartnerState}(s_t, H_t)$ {partner held object, subgoal, blocked actions}
2: $\hat{a}_j^{t+1} \leftarrow \textsc{PredictPartnerAction}(b_j^t, s_t)$
3: $g_i^t \leftarrow \textsc{SelectAnticipatorySubgoal}(s_t, b_j^t, \hat{a}_j^{t+1})$ {e.g., pre-stage object before request}
4: prompt $\leftarrow$ Compose$(s_t, H_t, b_j^t, \hat{a}_j^{t+1}, g_i^t, \mathcal{A}_i(s_t))$
5: $(a_i^t, m_i^t) \leftarrow$ LLM(prompt)
6: **return** (a_i^t, m_i^t)

The partner-state inference step extracts a structured tuple $\langle$role, held, station, last_request, likely_subgoal$\rangle$, and the predicted next action is generated as a constrained classification call against $\mathcal{A}_j(s_t)$.

Pro-Agent prompt sketch.

```
You are agent {role} in a two-cook kitchen.  Your partner's inferred
state is:  {partner_state}.  Your partner's most likely next action
is:  {predicted_action}.  Choose your action to anticipate or complement
this, avoiding duplicate work.  Respond as JSON with fields action,
target, message.
```

A.2. Adaptive Theory of Mind (AToM)

Algorithm 2 AToM decision step for agent i at time t

Require: Environment state s_t, dialogue history H_t, prior belief b_j^{t-1}, observed partner action a_j^{t-1}, observed partner message m_j^{t-1}, verifier feedback v^{t-1}

1: $\Delta \leftarrow$ COMPUTEFEEDBACKSIGNALS$(a_j^{t-1}, m_j^{t-1}, v^{t-1})$ {response latency, fulfillment, error}
2: $b_j^t \leftarrow$ UPDATEBELIEF(b_j^{t-1}, Δ, s_t)
3: prompt $\leftarrow$ Compose$(s_t, H_t, b_j^t, \mathcal{A}_i(s_t))$
4: $(a_i^t, m_i^t) \leftarrow$ LLM(prompt)
5: **return** (a_i^t, m_i^t)

Belief schema. b_j^t is encoded as a JSON-serialized dictionary with fields `subgoal`, `held_object`, `blocked_actions`, `recent_latency`, `recent_fulfillment`, and `communication_style`. The dictionary is included as a structured prefix in the agent's decision prompt.

Belief-update prompt sketch.

```
You are partner-modeling an agent.  Given the previous belief, the
partner's most recent action, message, and any verifier feedback,
return an updated belief as JSON. Update only fields with new evidence;
otherwise carry forward.
```

Appendix B. Additional Results

B.1. Per-Level Breakdown for ToM Scaffolds

B.2. TGCO Composition and Cost-Uptake Trade-off

B.3. Verifier Interventions by Directed Pairing

B.4. Fixed-Window ADR Robustness Check

Table 5: Success rate and ADR by task level under each scaffold. **Aggregation:** *self-play only* (each model paired with itself), reported per task level. The GPT-4o/Pro-Agent rows show the self-play coordination collapse discussed in §5 (success $1.00/0.97/0.93 \rightarrow 0.70/0.50/0.30$; ADR $\rightarrow 0$ at Levels 2–3).

Model	**Framework**	**Level 1**		**Level 2**		**Level 3**	
		Succ	ADR	Succ	ADR	Succ	ADR
GPT-4o	Baseline	1.00	0.30	0.97	0.25	0.93	0.18
GPT-4o	Pro-Agent	0.70	0.05	0.50	0.00	0.30	0.00
GPT-4o	AToM	0.90	0.55	0.80	0.50	0.70	0.45
Claude Sonnet 4	Baseline	0.85	0.12	0.69	0.08	0.55	0.05
Claude Sonnet 4	Pro-Agent	0.95	0.45	0.80	0.38	0.65	0.30
Claude Sonnet 4	AToM	0.85	0.55	0.70	0.50	0.55	0.45
Gemma-31B	Baseline	0.70	0.52	0.50	0.47	0.30	0.40
Gemma-31B	Pro-Agent	0.95	0.62	0.80	0.57	0.65	0.50
Gemma-31B	AToM	1.00	0.78	0.92	0.73	0.85	0.65

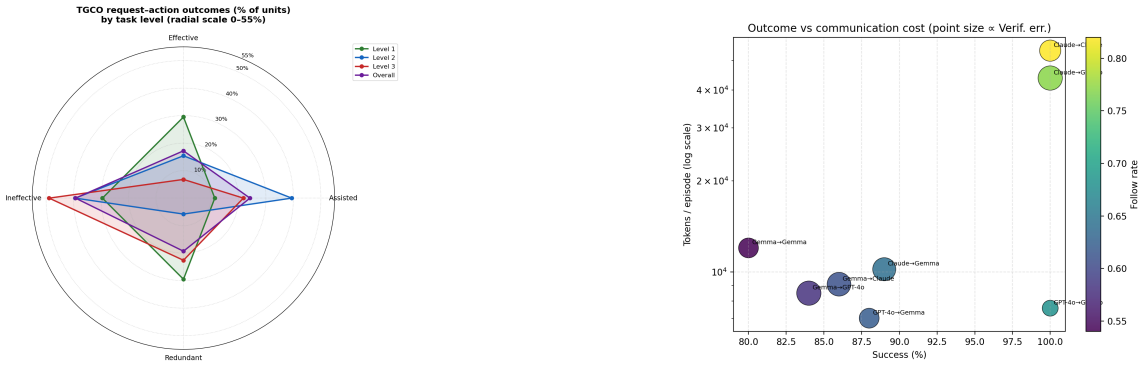

Figure 3: Left: TGCO outcome composition by task level. Right: Success vs. coordination waste vs. knowledge-directed uptake in cross-model experiments. The underlying numerical values are tabulated in Tables 4 and 6.

Appendix C. TGCO Audit

C.1. Setup additional details

Rollout count and scale. We evaluate 3 model families in 9 directed, role-sensitive pairings (3 self-play and 6 cross-model), across 3 task-complexity levels, with 10 rollouts per pairing-level cell unless otherwise noted. This yields $9 \times 3 \times 10 = 270$ rollouts per agent framework and 810 total rollouts when aggregating Baseline, Pro-Agent, and AToM. Experiments span 15 dishes in total (5 per level). All models use temperature 0, identical prompt templates, JSON output schemas, and validator-guided retry policies. Logs include messages, selected actions, verifier feedback, parser repairs, invalid attempts, token counts, and wall-clock latency.

Table 6: Verifier interventions per episode for selected directed role pairings (chef → assistant) cited in §5 and Fig. 2. Same-model pairs trigger fewer corrections than the selected cross-model pairs.

Pairing (chef → assistant)	Verifier interventions / episode
GPT-4o → GPT-4o	9
Claude → GPT-4o	21
Gemma → GPT-4o	20–21

Table 7: Comparison of level-normalized ADR ($\alpha = 0.15$) and fixed-window ADR ($w = 20$) per task level (macro-averaged across pairings). The level-3 ADR drop is preserved under both window choices, indicating that the degradation is not a window artifact.

Level	**Level-normalized ADR**	**Fixed-window ADR** ($w = 20$)
Level 1	0.71 ± 0.07	0.74 ± 0.07
Level 2	0.66 ± 0.08	0.66 ± 0.08
Level 3	0.58 ± 0.10	0.55 ± 0.11

C.2. Pseudo Code for TGCO

The decision order matters: Redundant is checked before TraceLookup because a prior completion means the request was unnecessary regardless of whether the partner acts again. The four return values map directly to the outcome table; UNSTRUCTURED messages are filtered out before any scoring, so they do not penalize ParseRate but are excluded from the FollowRate and DelayedRate denominators. This convention is the one stated in §3.2 and used throughout the paper.

Algorithm 3 TGCO audit for message m_t from agent i at time t

Require: Message m_t, execution trace $\mathcal{T}$, acceptance window w_l, recipe state R_t, partner index j

$u \leftarrow$ PARSE(m_t) {extract request unit (o, a, j): object, action, target}
if $u = \emptyset$ **then**
  return UNSTRUCTURED {no identifiable partner-directed request; excluded from all denominators}
end if
$v_{\text{target}} \leftarrow$ TARGETCHECK$(u,\ j)$ {intended actor j is identifiable in u}
$v_{\text{rel}} \leftarrow$ RELEVANCECHECK$(u,\ R_t)$ {(o, a) belongs to active recipe or task state}
if $\neg\, v_{\text{target}}$ **or** $\neg\, v_{\text{rel}}$ **then**
  return INEFFECTIVE
end if
prior $\leftarrow$ REDUNDANCYCHECK$(\mathcal{T},\ u,\ t)$ {(o, a) already completed before t}
if prior **then**
  return REDUNDANT
end if
match $\leftarrow$ TRACELOOKUP$(\mathcal{T},\ u,\ t,\ w_l)$ {j executes (o, a) within $[t,\ t + w_l]$}
if $\neg$ match **then**
  return INEFFECTIVE {no partner action, object mismatch, or self-executed}
end if
$e_{\text{val}} \leftarrow$ VALIDATORCHECK$(\mathcal{T},\ u,\ t,\ w_l)$ {validator correction occurs before match}
if e_{val} **then**
  return ASSISTED
else
  return EFFECTIVE
end if

Appendix D. Contamination Probe

To gauge benchmark familiarity, we prompted each model to describe Collab-Overcooked *unprompted*—providing no layout, recipe, or rule—and scored whether it could reproduce task-specific structure (NRC layouts, recipe graph, role asymmetry). In all three cases (GPT-4o, Claude Sonnet 4, and Gemma-31B), the models showed only generic familiarity with Overcooked-style collaborative cooking and did not reproduce the specific NRC layouts, recipe graph, or role-asymmetric task structure used in our experiments. This is a lightweight check rather than a full contamination audit: it can flag verbatim layout memorization but cannot rule out general Overcooked priors acquired during pretraining. Accordingly, we scope our empirical claims to the Collab-Overcooked family (see Limitations) and treat coordination behavior as potentially reflecting familiar patterns as well as novel strategic adaptation.

Camera-Ready Edits Summary

This page is provided for workshop administration and transparency, as requested in the camera-ready instructions; it summarizes the changes relative to the original submission (Submission 31) and is not part of the technical content of the paper. All revisions fit within the one additional page permitted; each item names the reviewer(s) addressed.

A. Metric definitions made internally consistent (Reviewer uCXp).

- *FollowRate denominator (Cons. 1):* §3.2 now excludes REDUNDANT and UNSTRUCTURED units, $\text{FollowRate} = (N_{\text{eff}}+N_{\text{assist}})/(N_{\text{eff}}+N_{\text{assist}}+N_{\text{ineff}})$, matching Alg. 3 (REDUNDANT returns before TRACELOOKUP); the full composition is reported separately.
- *"Delayed" defined (Cons. 1):* "delayed" is now an explicit synonym for ASSISTED (fulfilled after validator correction, a follow-through success); §5 reports a four-way composition summing to 100%.
- *Unstructured handling unified (Cons. 2):* §3.2, App. C, and Alg. 3 now agree—UNSTRUCTURED units are filtered out before scoring and excluded from every denominator (the prior §3.2 "folded into INEFFECTIVE" claim is corrected).

B. Table and reporting transparency (Reviewers bcVd, 52kr).

- *Aggregation scopes labeled (bcVd W1; 52kr):* captions of Tables 2, 3, 5 now state the exact subset (NRC-only; macro-average over all pairings+levels; self-play per level), and §5 adds a "what each table reports" pointer—resolving the Table 3 vs. Table 4 ADR discrepancy.
- *Pro-Agent/GPT-4o collapse acknowledged (bcVd W2):* §5 now reports the GPT-4o self-play collapse under Pro-Agent (success $1.00/0.97/0.93 \rightarrow 0.70/0.50/0.30$; ADR$\rightarrow 0$), citing Table 5.
- *Body TGCO table added (bcVd W3):* new Table 4 defines *uptake* and *waste* with their (distinct) denominators and reports them for the two configurations cited in §5, grounding the "uptake 0.50, waste 0.15" claim.
- *Figure numbers tabulated (bcVd W7):* verifier counts behind Fig. 2 are now in Table 6; Fig. 3 values are cross-referenced.
- *Per-cell variability (bcVd W4; 52kr):* §4 reports point estimates over the 10 rollouts per cell.

C. Scope, formalism, ToM, contamination (Reviewers 52kr, bcVd, AC).

- *Claims scoped (52kr; AC):* abstract, §5, and Limitations scope empirical conclusions to the Collab-Overcooked family while keeping TGCO/NRC as general instruments.
- *Formalism (bcVd W5; 52kr):* §3.1 now states the Markov game and utility U are expository (λ_i not instantiated, U not optimized); the Pareto/strategy-proof remark is softened and deferred.
- *ToM comparison caveated (52kr; bcVd W6):* §5/Limitations note Pro-Agent and AToM differ in mechanism *and* prompt structure, so "mixed gains" applies to these scaffolds as deployed; §1 credits the metrics+NRC as the novelty.
- *Contamination probe (52kr):* new App. D adds an unprompted-description probe and reports its outcome; verifier-stratified analysis is acknowledged as future work in the Limitations.

D. Related work and citations (Reviewers uCXp, bcVd).

- *Related work strengthened (uCXp Cons. 3):* a new §2 paragraph cites MultiAgentBench [23], MAST [6], LLM-Coordination [1], Tool-RoCo [21], and Kleinberg & Raghavan [9].
- *Broken references corrected (bcVd W8):* Xi et al. (arXiv:2308.04026 → "The Rise and Potential of Large Language Model Based Agents," arXiv:2309.07864) and the Sun 2025 thesis DOI (re-pointed to KiltHub) are fixed; the LLM-Hanabi ID (2510.04980) was verified valid and left unchanged.