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

author names withheld

Under Review for NExT-Game 2026

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 completion-based evaluation is insufficient for LLM-based multi-agent systems and that trace-grounded coordination metrics are needed to measure strategic behavior.

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 [3]. Benchmarks such as Hanabi-style belief-sensitive games [1, 8] and Overcooked-style cooperation [4, 12] 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 reflect stylistic alignment rather than robust strategic reasoning, a form of algorithmic monoculture that breaks down when partners change [3, 13]. Recent work also argues that temporally extended

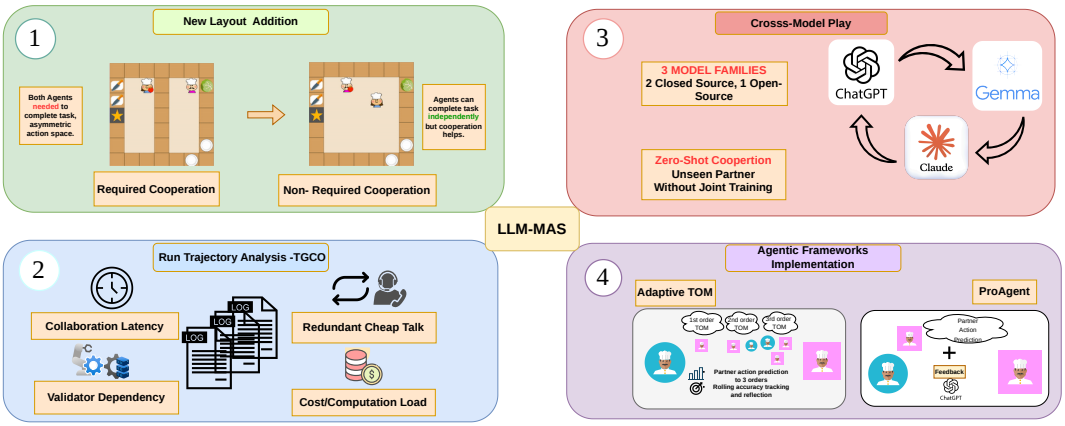

Figure 1: Overview of methodology and evaluation pipeline. 1) Addition of new Non-Required Cooperation layout 2) Cross-Model Family analysis for ZSC across open- and closed-source LLM 3) TGCO audit from run trajectory logs 4) Adaptive-ToM and ProAgent architectures.

partner modeling remains under-measured in current LLM-based Theory-of-Mind frameworks and benchmarks [10].

We study this gap in Collab-Overcooked [12], 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.

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 [14, 15]. Benchmarks including LLM-Hanabi [8], and Collab-Overcooked [12] measure cooperative task performance. Our work focuses on the strategic structure hidden underneath task success: directed dependence, signal follow-through, and partner robustness.

Interdependence and zero-shot cooperation. Zero-shot cooperation asks whether agents can collaborate with previously unseen partners [6, 11, 16, 18]. Constructive interdependence [3] 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 [2]. LLMs perform well on some isolated ToM tests [5, 7], but static tests do not establish robust interactive coordination. We evaluate two ToM-oriented scaffolds, ProAgent [17] and Adaptive Theory of Mind (AToM) [9], 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. U makes the trade-off between task reward and coordination cost explicit; 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 [12] to the NRC setting with symmetric action spaces but asymmetric task knowledge, and separate the agents to enable **cross-model** evaluation of partnership behavior.

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.	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 $[t, t + w_l]$.

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 $[t, t + w_l]$. Outcomes are EFFECTIVE, ASSISTED, REDUNDANT, and INEFFECTIVE; unstructured messages are included in INEFFECTIVE. Let $N_{\text{req}} = N_{\text{eff}} + N_{\text{assist}} + N_{\text{red}} + N_{\text{ineff}}$. We report

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

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.

Table 2: RC/NRC performance. NRC tasks permit solo completion, allowing coordination to be evaluated as an efficiency strategy rather than a feasibility condition.

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

Explicit Partner Modeling Pro-Agent predicts the partner's likely next action and chooses anticipatory actions [17]. Adaptive-ToM maintains a rolling belief over partner subgoal, held object, blocked actions, recent latency, fulfillment, and communication style [9]. 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 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), 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, and both raise per-episode latency and token cost. **TGCO.** Across all settings, 44% of audited request units are redundant and 35% are ineffective or delayed; 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. Classical cooperative solution concepts such as Pareto efficiency and strategy-proof signaling fail to characterize these joint policies, motivating new stability notions for boundedly-rational agentic AI.

Limitations. The model panel covers three LLM families and a single benchmark; verifier-guided retries mix interface robustness with collaborative reasoning, and we do not evaluate human-LLM teams. Partner adaptation tests, payoff perturbations, and stronger joint-planning baselines remain future work.

Table 3: ToM framework comparison across model families.

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

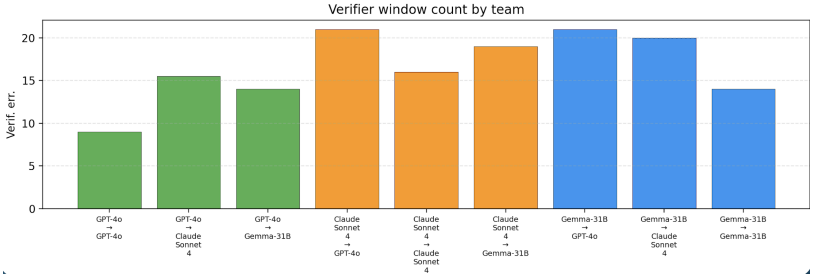

Figure 2: Verifier errors by directed role pairing. Same-model pairs (e.g., GPT-4o→GPT-4o) trigger fewer corrections than cross-model pairs (e.g., Claude→GPT-4o, Gemma→GPT-4o), where validator interventions reach 20–21 per episode.

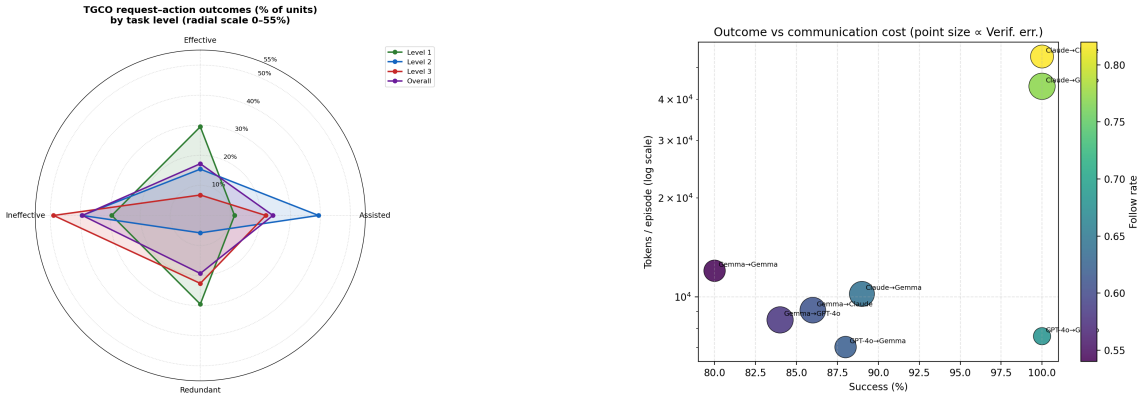

Figure 3: Left: TGCO outcome composition by task level. Right: Success vs. coordination waste vs. knowledge-directed uptake in cross-model experiments.

References

[1] 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 han-

abi 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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/.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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/.

[13] 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.

[14] 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.

[15] Zhiheng Xi, Weizhe Chen, Xiaoxuan Guo, Wei Wang, Yuhang Chen, Wen Zhang, Xinyu Gao, Weizhou Li, Yifei Ma, Xizhou Wang, et al. Large language model agents: A survey. *arXiv preprint arXiv:2308.04026*, 2023.

[16] 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.

[17] 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.

[18] 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.

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

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

$\Delta \leftarrow \textsc{ComputeFeedbackSignals}(a_j^{t-1}, m_j^{t-1}, v^{t-1})$ {response latency, fulfillment, error}
$b_j^t \leftarrow \textsc{UpdateBelief}(b_j^{t-1}, \Delta, s_t)$
prompt $\leftarrow$ Compose$(s_t, H_t, b_j^t, \mathcal{A}_i(s_t))$
$(a_i^t, m_i^t) \leftarrow$ LLM(prompt)
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

Table 4: Success rate and ADR by task level under each scaffold.

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

B.2. Fixed-Window ADR Robustness Check

Table 5: 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

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.

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 don't penalize ParseRate but are excluded from FollowRate and DelayedRate denominators.

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

$u \leftarrow \textsc{Parse}(m_t)$ {extract request unit (o, a, j): object, action, target}
if $u = \emptyset$ **then**
  return UNSTRUCTURED {no identifiable partner-directed request}
end if
$v_{\text{target}} \leftarrow \textsc{TargetCheck}(u,\ j)$ {intended actor j is identifiable in u}
$v_{\text{rel}} \leftarrow \textsc{RelevanceCheck}(u,\ R_t)$ {(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 $\leftarrow \textsc{RedundancyCheck}(\mathcal{T},\ u,\ t)$ {(o, a) already completed before t}
if prior **then**
  return REDUNDANT
end if
match $\leftarrow \textsc{TraceLookup}(\mathcal{T},\ u,\ t,\ w_l)$ {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
$e_{\text{val}} \leftarrow \textsc{ValidatorCheck}(\mathcal{T},\ u,\ t,\ w_l)$ {validator correction occurs before match}
if e_{val} **then**
  return ASSISTED
else
  return EFFECTIVE
end if