Collaborative Tasks with Heterogenous LLM Students

Anonymous ACL submission

Abstract

Advances in LLMs offer hope of corresponding advances in agent participation in teamwork, while also posing new challenges in designing multi-agent benchmarks for evaluating these agents and integrating them effectively into hybrid teams in real-world situations. While prior work has demonstrated that LLMs can operate in multi-agent settings, they often oversimplify the complexity of collaboration in critical dimensions, such as restricting evaluation to in-domain and single episode tasks amongst homogeneous LLM groups. To bridge this gap, we propose a new cooperative multi-agent task, Kitchen-Alien Rush, which includes both out-of-domain multi-episode evaluation, as well as evaluates the effectiveness of hybrid groups in collaboration. Our findings reveal that our evaluation exposes gaps in multi-agent collaboration, as LLM agents struggle to perform in the out-of-domain task and show inconsistent improvement over multiple episodes in hybrid teams. By identifying these gaps, we motivate the need for future work in addressing weaknesses of hybrid multi-agents systems for out-of-domain multi-episode tasks.

1 Introduction

Human-human collaboration is challenging and has been demonstrated to benefit from automated support, such as agent-based support (Adamson et al., 2014; Naik et al., 2024). The language capabilities of state-of-the-art Large Language Models (LLMs) offer hope for advances in this space (Brown et al., 2020; Driess et al., 2023), but also raise new questions about how to design such agents and eventually to introduce them effectively into hybrid teams in real world scenarios. Simulation studies offer a means to generate synthetic data and to narrow down the space of designs, strategies, and behavior practices, or troubleshoot novel agent capabilities in highly controlled environments prior to running more realistic but costly user studies with human participants (Gao et al., 2023). This paper contributes a simulation study paradigm involving LLM agents that can play the role of collaborators or supporters of collaboration.

In the same way that collaboration benefits humans, strong collaborative skills in multi-agent systems may enhance problem solving and decision making (Zhang et al., 2024c). Collaboration creates the opportunity for agents to learn from each other's output and increase the abilities of weaker models to complete tasks. Moreover, if we desire models to seamlessly assist humans, they must understand how to address challenges in collaboration, such as adapting support to variable levels of capabilities and improving coordination with other partners over time (Hu et al., 2020; Carroll et al., 2019).

In recent years, there has been increased interest in developing collaborative multi-agent systems using LLMs (Carroll et al., 2019; Gong et al., 2023; Zhang et al., 2024b; Light et al., 2023; Zhou et al., 2024). However, prior agent simulation studies are limited in their ability to inform design work for agents that participate in hybrid teams because of the narrow scope of task contexts. While these frameworks have demonstrated that LLMs can operate in multi-agent settings, they often oversimplify the complexity of collaboration in critical dimensions. Firstly, they restrict evaluation to in-domain and single episode tasks, which fails to measure a model's ability to adapt to changing circumstances, partner relationships, and feedback. Secondly, they do not evaluate heterogeneous LLM teams including models with varying strengths and capabilities. This motivates evaluation of multi-agent LLM systems with hetereogenous partners across domain and multiple episodes.

Our main contributions as a paradigm for advancing work in agent support for hybrid teams include:

- **Task** To bridge this gap of developing complex collaborative benchmarks, we propose a new cooperative multi-agent task, Kitchen-Alien Rush. In contrast to prior work, our task consists of multiple episodes and includes two domains (Table 1).

- **Teacher** We explore a simple teacher-feedback agent framework for multi-episode settings.
- **Findings** Our findings reveal that LLM agents struggle to perform in the out-of-domain task and show inconsistent improvement over multiple episodes. Moreover, when placed in hybrid teams, groups fail to adapt to their partner, with stronger partners contributing the majority of performance, which does not exhibit behavior that is expected to enable groups to succeed in scenarios that require group coordination.

2 Related Work

Collaboration Tasks for LLMs As LLMs have demonstrated remarkable capabilities in single-agent tasks (Brown et al., 2020; Qin et al., 2023; Driess et al., 2023; Wu et al., 2023), there has been increased interest in developing benchmarks for multi-agent scenarios. These benchmarks range across multiplayer games (Carroll et al., 2019; Bard et al., 2019; Light et al., 2023; Qi et al., 2024; Gong et al., 2023), virtual embodied household tasks (Zhang et al., 2024b; Guo et al., 2024), multi-robot collaboration (Mandi et al., 2023), and social scenarios (Zhou et al., 2024). Most involve pure collaboration tasks in which groups work toward a single, shared goal. Notable examples include coordination of cooking in Overcooked (Carroll et al., 2019), investigation of Theory of Mind abilities in Hanabi (Bard et al., 2019), and completion of tasks in virtual embodied environments (Zhang et al., 2024b; Mandi et al., 2023). Other multi-agent benchmarks have mixed objectives in which groups have to compete against other teams (e.g., Avalon (Light et al., 2023), Werewolf (Xu et al., 2024)) or potentially negotiate conflicting social goals and motivations (e.g., Sotopia (Zhou et al., 2024), Civilization (Qi et al., 2024))

Current benchmarks do not investigate an agent's ability to adapt to partner relationships and group coordination strategies over an extended time frame. Moreover, tasks are set in typical scenarios where world knowledge accessible to LLMs might bolster performance. Thus, existing benchmarks are not useful for evaluating the ability to adapt to new domains through abstraction. Consequently, prior results cannot separate aspects of performance that are indicative of abstraction and generalization, for example by leveraging strategic knowledge about collaboration. Our proposed task addresses these issues by evaluating across multiple episode and task domains.

	Domain		**Episodic**		**Group**	
Framework	In	Out	Single	Multi	Homo	Hetero
Zhang et al. (2024b)	✓	✗	✓	✗	✓	✗
Guo et al. (2024)	✓	✗	✓	✗	✓	✓
Shi et al. (2023)	✓	✗	✓	✗	✓	✓
Xu et al. (2024)	✓	✗	✓	✓	✓	✗
Zhang et al. (2024a)	✓	✗	✓	✗	✓	✗
Liu et al. (2023)	✓	✗	✓	✗	✓	✗
Agashe et al. (2024)	✓	✗	✓	✗	✓	✗
Mandi et al. (2023)	✓	✗	✓	✗	✓	✗
Zhou et al. (2024)	✓	✗	✓	✗	✓	✓
Ours	✓	✓	✓	✓	✓	✓

Table 1: Comparison of multi-agent tasks and frameworks with LLMs.

Multi-Agent LLM Systems Most multi-agent systems enhance performance with increased reasoning guidance (Wei et al., 2022) and reflection feedback strategies (Yao et al., 2022; Shinn et al., 2023). Other frameworks break down complex tasks by managing team organization (Guo et al., 2024; Zhao et al., 2024; Hong et al., 2023). In a similar vein, other works improve LLM task planning through hierarchical reasoning (Liu et al., 2023), and sub-task planning (Mandi et al., 2023). Our work extends the frontier by developing and evaluating across multiple episodes and task domains.

Computer Supported Collaborative Agents The field of computer-supported collaborative learning offers numerous technologies and design principles for supporting collaboration, for the purpose of mitigating these collaboration challenges (Cress et al., 2021). Some of that work features intelligent conversational agents (Naik et al., 2024; Adamson et al., 2014; Karyotaki et al., 2022) or teachable agents (Ma et al., 2024; Jin et al., 2024). Other applications of AI to support collaboration monitor the progression of a group's knowledge to identify learning and adjust instruction as needed (Weitekamp et al., 2020; Han et al., 2021). Our work bridges between past literature by offering a simulation-based paradigm with the long term goal of design of agents for supporting collaboration in hybrid teams with multiple humans and multiple

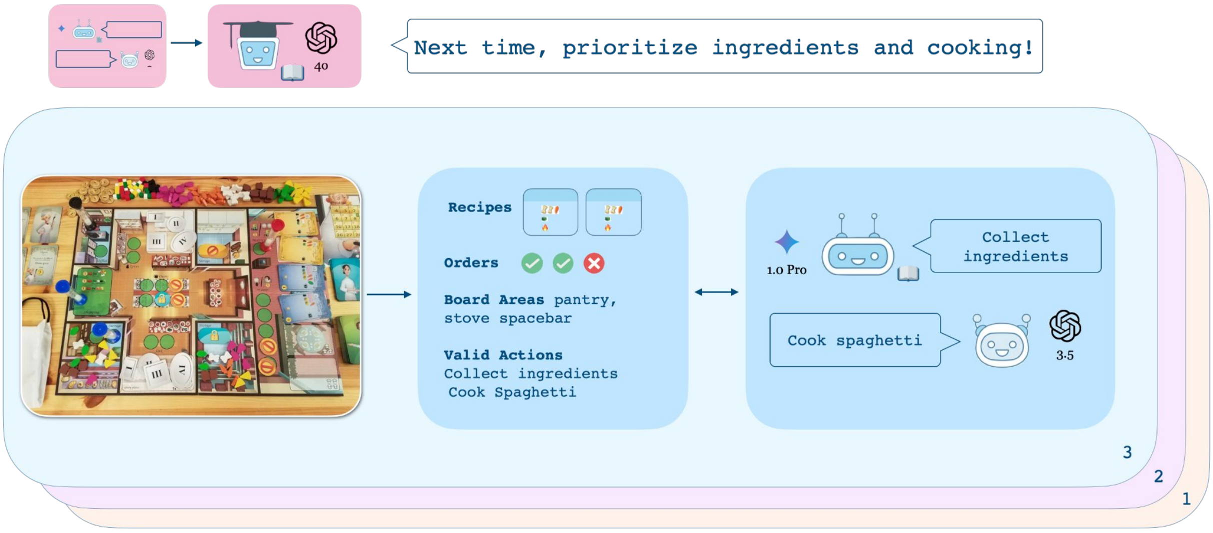

Next time, prioritize ingredients and cooking!
40
Recipes
Orders
Board Areas pantry, stove spacebar
Valid Actions Collect ingredients Cook Spaghetti
1.0 Pro
Collect ingredients
Cook spaghetti
3.5
3
2
1

Figure 1: **Kitchen Rush**: A new evaluation benchmark for LLM collaboration over three episodes. In each episode, the game state is initialized and encoded into a textual observation. From observations with manual or teacher scaffolding, agents choose actions and update the game state.

autonomous agents.

3 Task: Kitchen Rush & Alien Rush

To evaluate multi-agent groups in collaboration games within a realistic coordination environment, we require a benchmark analyzing performance in long-horizon interactions measured across multiple episodes. Consequently, we introduce a new text simulation environment and task for multi-agent coordination inspired by the board game Kitchen Rush (Turczi and Bagiartakis, 2017). Additionally, to evaluate group collaboration skills in an out-of-domain setting, we implement a parallel out-of-domain version of Kitchen Rush, which we refer to as Alien Rush.

3.1 In Domain Kitchen Rush

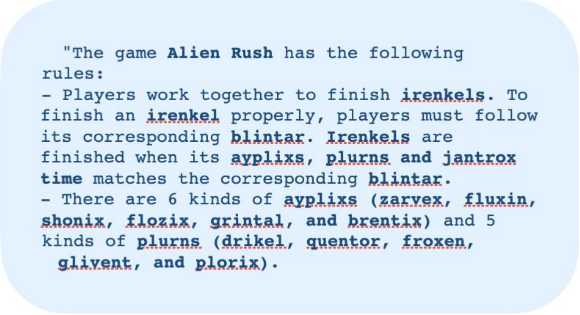
In Kitchen Rush, the goal of each episode is to complete all assigned recipes within the specified number of turns. Each turn, agents control a specific number of workers, each of which can be used to complete one high-level action per turn, such as taking ingredients, or cooking a meal on a stove. An episode of a scenario ends when all assigned recipes are either finished or ruined, or the group exceed the allotted number of turns for the episode.

Agents work together in teams to complete each recipe. Every recipe requires (1) **ingredients** (pasta, carrots, lettuce, meat, or bread), (2) **spices** (white, black, green, red, and yellow), and (3) **cook time** to complete. In Figure 3, Spaghetti Aglio e Olio requires collecting ingredients (2 pasta and 1 carrot), spices (1 green spice), and needs to be cooked once. If an agent is not careful or a group does not coordinate their actions properly, they may ruin an order (e.g., cooking a meal before collecting ingredients, exceeding the cook time of the order). Once ruined, orders cannot be completed.

In addition to coordinating actions to complete dishes, groups must also appropriately share resources. Every action must be carried out in a specific location of the kitchen (e.g., ingredients from the **pantry**, spices from the **spicebag**, and orders are cooked on the **stoves**). Locations have a limited number of action spaces, which restricts how many of the same action can be completed every turn. A kitchen with only two stoves means there can only be two cook actions every turn. In this case, if the recipes require long cook times, then it would be imperative that agents share stoves carefully and plan to cook earlier than later.

3.2 Out of Domain Alien Rush

To evaluate group collaboration performance in an out-of-domain setting, we evaluate on Alien Rush, a parallel version of Kitchen Rush which swaps all in-domain cooking entities to out-of-domain "alien" gibberish words. To do so, in each written prompt of Kitchen Rush, we manually identify all words and stems that refer to cooking. This includes cooking related terms such as ingredients, spices, recipe names, and stems of cooking-related verbs. Next, we generated gibberish words using ChatGPT for each cooking themed entity in Kitchen Rush. Us-



"The game **Alien Rush** has the following rules:
- Players work together to finish **irenkels**. To finish an **irenkel** properly, players must follow its corresponding **blintar**. **Irenkels** are finished when its **ayplixs**, **plurns** and **jantrox time** matches the corresponding **blintar**.
- There are 6 kinds of **ayplixs** (**zarvex**, **fluxin**, **shonix**, **flozix**, **grintal**, and **brentix**) and 5 kinds of **plurns** (**drikel**, **quentor**, **froxen**, **glivent**, and **plorix**).

Figure 2: Abbreviated sample Alien Rush manual.

ing a lexical mapping dictionary, we translate any text within Kitchen Rush to Alien Rush gibberish (Figure 2). This same one-to-one mapping is used across all scenarios and episodes of the game.

Consequently, Kitchen Rush and Alien Rush are functionally equivalent tasks; however, agents cannot benefit from their implicit world knowledge of cooking when performing in the Alien Rush Domain. Only high-level collaborative strategies should transfer to this novel setting.

3.3 Multi-Episode Scenarios

To allow for experimental design, a scenario's recipes and resources are easily modifiable, and rounds may be played with any number of agents. Each scenario consists of three episodes each with five assigned recipes, which must be completed within ten turns. Each episode within a scenario is strategically equivalent with the same initial state of resources and recipe composition but relevant entities are swapped. For instance, we alter the Spagetti Aglio Olio into a Mac and Cheese with two cheese and 1 pasta (instead of two pasta and one carrot), 1 black spice (instead of one green spice), and cook times remain the same. Initial resources are adjusted such that the same kinds of conflicts occur, such as ensuring one resource is severely limited. This prevents any success across episodes that could come from memorizing recipes but maintains the same difficulty and flavor of coordination challenges over a multi-episode evaluation.

Similar to work by Carroll et al. (2019), this flexibility of customization enables design of scenarios that can target evaluation of specific low-level game understanding versus high-level coordination strategies. When manipulating experimental variables, we require evaluation which can pinpoint why models are failing. For example, while transferring to an out-of-domain setting, we must identify whether

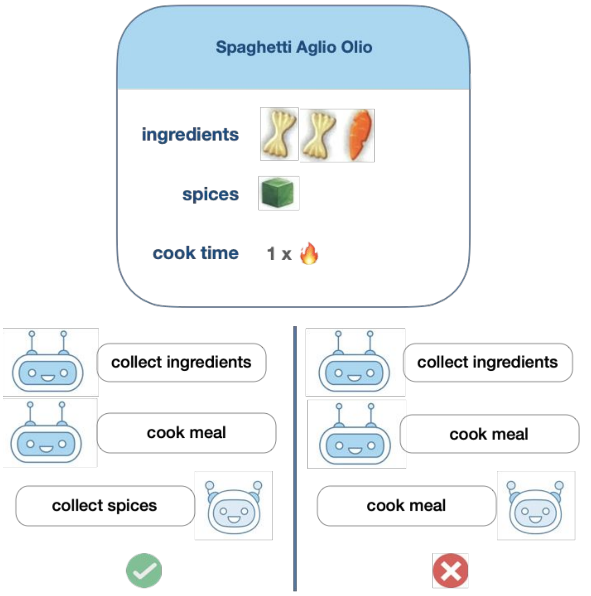


Figure 3: Example of a Kitchen Rush Recipe specifying ingredients, spices, and length of cook time. Agents must coordinate their actions in a valid order to complete the recipe (left). Otherwise, agents may ruin the recipe, such as cooking the meal too many times (right).

the domain shift results in an agent's inability to play the game due to lack of understanding of game rules in an alien setting or whether agents can understand the game but are unable to transfer higher-level coordination strategies across domain. Thus, we design an easy scenario and a hard scenario.

Easy Scenario serves as a baseline to evaluate an agent's understanding of the game. Each episode can be completed by one agent within the allotted turns and does not require coordination strategies. There are no resources conflicts, so agents have unlimited ingredients, spices, and actions spaces at each location. Groups that can complete all five recipes signal that they have mastery of the basic rules of the game. However, groups that ruin recipes in this scenario suggest they lack the ability to play the game at its simplest level.

Hard Scenario evaluates group ability to coordinate resources and strategies. The pantry and stoves are severely limited, such that all five recipes can only be completed by the final turn. Successful completion requires agents to divide tasks between collecting ingredients and cooking recipes. Without task division, groups cannot complete all five recipes within the allotted time. Thus, groups that complete less than five recipes, ruin recipes, or exceed the allotted time of the episode may indicate

Group Recipes/Turns		
Completed	(CR)	# recipes completed by the group
Ruined	(RR)	# ruined recipes completed by the group
Turns		# of turns in the episode
Individual Actions		
Positive	(A^+)	Agent contributed to completing recipes
Negative	(A^-)	Agent contributed to ruining recipes
Neutral	($A^=$)	Neutral action (e.g. no action) by an agent
Total		$A^+ + A^- + A^=$

Table 2: Evaluation Metrics

that they did not effectively apply group coordination strategies.

3.4 Metrics

We evaluate episodes on group and individual metrics (Table 2). Metrics are measured over all three episodes. By measuring over multiple episodes, we evaluate a group's ability to adapt to other partners and settings over an extended time frame. We use these metrics to measure performance and distinguish between greedily acting groups (individual success at the cost of coordination strategies) vs. balanced collaborative groups (high group success with equal individual participation).

4 Methods & Experimental Design

Success of human collaborations varies based on task characteristics, group composition including distribution of expertise and scaffolding. As we aim to define a paradigm in which questions related to supporting hybrid human-agent teams can be addressed going forward, we incorporate these three dimensions in our experimental design. To investigate the impact of one form of domain characteristics, we manipulate the extent to which tasks build on every-day knowledge with the Kitchen Rush and Alien Rush settings. To investigate expertise and group composition, we manipulate the family of models, GPT (OpenAI, 2023) and Gemini (Google, 2023), as well as the strength of model (GPT-3.5-Turbo, GPT-4o, Gemini 1.0 Pro and Gemini 1.5 Flash). We explore group composition in terms of homogeneity or heterogeneity along these two separate dimensions. Additionally, we explore two different types of support, namely a manual and a teacher (which is always a strong model, but which may be from either model family).

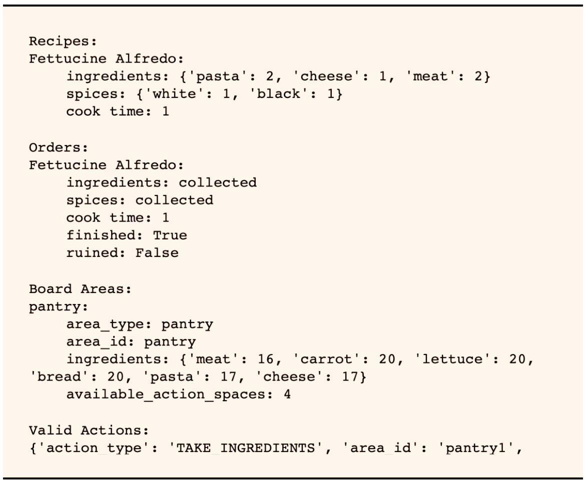
4.1 Methods

As simple baselines, we implement an LLM agent similar to those in prior LLM agent tasks (Yao et al., 2022; Agashe et al., 2024). In this method, LLMs are given an action generation prompt which contains relevant observations obtained from the environment and a list of candidate actions. Agents may also be given reasoning and additional knowledge of the task, which we refer to as scaffolding. With this textual information, agents must pick the best next action to update the environment from the candidate actions. We implement several ablations of scaffolding which we detail below. Full prompts for scaffolds may be found in Appendix A.

Observation Only (No Scaffolding) We provide agents with a text description of the current turn of the game with relevant details (e.g., recipes, order status, resource availability). Agents are prompted to choose the next best action within a list of valid candidate actions. An abbreviated sample of the observation is shown in Figure 4. In this condition, we examine the extent to which implicit world knowledge about the domain and the task structure helps model performance.

Observation and Manual Along with observations, we provide agents with a manual explaining the rules of the game. Agents may use provided explicit knowledge of the game to generate action from the observation. Thus, with a given manual, we investigate how well agents can reason from explicit low-level static information given about the task. For reference, we provide a sample of the manual in Figure 5.

Multi-Episode Teacher Feedback To design a framework for multi-agent collaborative groups that increase their performance over multi-episode interactions, we implement a teacher agent to provide relevant plans and strategies to agents from prior round history. Similar to the action generation prompt, in order to generate feedback, we give the teacher expert knowledge of the game via a board game manual as well as prior action history and results from the previous round. From this information, we prompt teachers to generate new strategies for the next round of the game or adjust prior strategies. We evaluate two teacher agents: **(1) a group strategy teacher** that gives the same feedback and strategies to every agent, and **(2) an individual roles and feedback teacher** that gives personalized roles and feedback to each



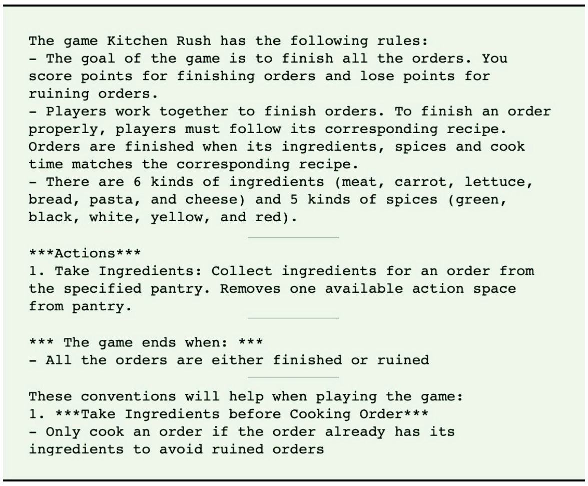
```
Recipes:
Fettucine Alfredo:
    ingredients: {'pasta': 2, 'cheese': 1, 'meat': 2}
    spices: {'white': 1, 'black': 1}
    cook time: 1

Orders:
Fettucine Alfredo:
    ingredients: collected
    spices: collected
    cook time: 1
    finished: True
    ruined: False

Board Areas:
pantry:
    area_type: pantry
    area_id: pantry
    ingredients: {'meat': 16, 'carrot': 20, 'lettuce': 20,
'bread': 20, 'pasta': 17, 'cheese': 17}
    available_action_spaces: 4

Valid Actions:
{'action_type': 'TAKE_INGREDIENTS', 'area_id': 'pantry1',
```

Figure 4: Abbreviated observation prompt



```
The game Kitchen Rush has the following rules:
- The goal of the game is to finish all the orders. You
score points for finishing orders and lose points for
ruining orders.
- Players work together to finish orders. To finish an order
properly, players must follow its corresponding recipe.
Orders are finished when its ingredients, spices and cook
time matches the corresponding recipe.
- There are 6 kinds of ingredients (meat, carrot, lettuce,
bread, pasta, and cheese) and 5 kinds of spices (green,
black, white, yellow, and red).

***Actions***
1. Take Ingredients: Collect ingredients for an order from
the specified pantry. Removes one available action space
from pantry.

*** The game ends when: ***
- All the orders are either finished or ruined

These conventions will help when playing the game:
1. ***Take Ingredients before Cooking Order***
- Only cook an order if the order already has its
ingredients to avoid ruined orders
```

Figure 5: Abbreviated manual prompt

player. In this condition, we examine how explicit high-level dynamic feedback may improve performance across multiple episode.

4.2 Experimental Design

The effectiveness of collaborations depends on the nature of the tasks, the makeup of the group (particularly the distribution of skill and expertise), and scaffolding. In our effort to create a paradigm for exploring how to best support hybrid human-agent teams, we incorporate these factors into our experimental approach. For each experimental dimension, we evaluate on both easy and hard scenarios to separate evaluation of low-level individual game coordination and high-level group collaborative skills.

Domain The level of implicit strategy knowledge of a large language model on a task varies by domain. We evaluate models across domains in order to identify whether the agents and teachers abstract generalizable collaboration strategies. As such, we evaluate pairs on both the in-domain Kitchen Rush task and the out-of-domain Alien Rush task. In our paradigm, we aim to separate aspects of performance contributed by the implicit world knowledge of in-domain tasks to the use high-level abstractions to transfer knowledge in out-of-domain tasks.

LLM Group Composition We experiment with homogeneous and heterogeneous pairs. Homogeneous pairs contain agent partners with the same LLM models. In contrast, heterogeneous pairs consist of partners with two different LLM models. We vary along four models across two LLM families: GPT-4o, GPT-3.5-Turbo, Gemini 1.0 Pro, and Gemini 1.5 Flash. We further frame our analysis across different LLM strengths in which we categorize larger LLM models with longer contexts as strong (GPT-4o and Gemini 1.5 Flash) and smaller models with shorter contexts as weak (GPT-3.5-Turbo, Gemini 1.0 Pro). This dimension aims to success the ability of LLMs to adapt to other partners.

Teacher Scaffolding We vary experiments by different scaffolding. In particular, we examine the effects of group strategy and personalized roles teachers in comparison to static scaffolding in manuals. We limit the teacher model to strong models. By measuring the effects of dynamic scaffolding, we aim to address whether LLMs can improve performance and learn over multi-episode interactions.

5 Results and Discussion

5.1 Domain Results

We investigate the effects of domain on different model groups. We focus our analysis on homogeneous groups. In doing so, we evaluate baseline performance across domains for each model.

In domain, strong models perform well. Weaker models struggle (Figure 6). Across both domains, we observe that strong models tend to perform better than weaker models. GPT-4o tends to perform the best while Gemini 1.5 Flash is a close equivalent when a manual is present. In particular, we note that strong groups can finish all five recipes in the easy difficulty while weaker groups struggle to complete more than two. The high performance by strong models in the easy scenario indicate that they have a stronger understanding of basic game rules while the lower performance suggest weaker models do not. This is also shown by the severe

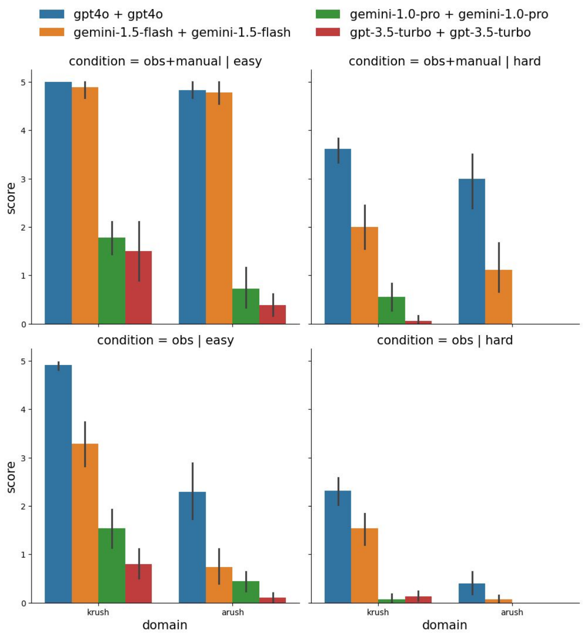


Figure 6: Comparison of average number of completed recipes and error margin across domain, conditions, and difficulty for homogeneous groups

lack of recipes completed by weaker models in the difficult collaborative scenario. This aligns with expectations of models given their size and context length.

Out-of-domain, strong models can perform with a manual, however performance drops on hard problems. Strong groups with a manual can understand basics of Alien Rush. In fact, in the easy scenario, both strong groups are able to achieve close to equivalent performance to Kitchen Rush. Thus, we conclude if strong models are given a manual, they have enough explicit knowledge to play the game in both domains despite having less implicit world knowledge in the out-of-domain Alien Rush environment.

However, even with a manual, strong groups in the harder scenario perform worse in Alien Rush. Unlike in-domain findings, they do not reach equivalent performance to the Kitchen Rush counterpart. This gap suggests that coordination skills required in the hard difficulty might not transfer in out-of-domain settings.

Performance drops in out-of-domain hard problems may be explained by lack of collaborative skill transfer and difficulty of the setting (Table 3). To investigate this performance gap, we looked more closely at the nature of the performance across domains for strong groups. In the hard scenario, all five recipes can only be completed by the final turn. Groups that fail fall under one of two cases: (1) groups complete less recipes but time out, which signals agents are competent but are not collaborating effectively or (2) groups ruin recipes and terminate episode early, which signals agents are less competent and struggle to apply rules properly in the out-of-domain setting.

Turns		< 10			= 10			
					CR			Ave
Model	Domain		1	2	3	4	5	RR
GPT-4o	Kitchen	0	0	22	78	0	0	0.11
GPT-4o	Alien	0	11	22	67	0	0	0.22
Gemini 1.5 Flash	Kitchen	11	33	22	0	0	34	2.66
Gemini 1.5 Flash	Alien	0	0	0	0	0	100	3.88

Table 3: Percentage of homogeneous strong groups with a manual that terminated early (<10) or reached turn limit (=10) in the hard difficulty, and average ruined recipes from runs.

When examining the performance of GPT-4o in Kitchen Rush, we observe that GPT-4o models tend to complete more recipes before timing out. This suggests that while the model's collaboration was not perfectly successful, they were still able to collaborate well enough to complete four recipes. In contrast, groups tend to complete less recipes and time out in Alien Rush. This suggests that in an out-of-domain setting, GPT-4o cannot as effectively transfer collaborative skills shown from Kitchen Rush to Alien Rush.

5.2 LLM Group Composition Results

With the established trends across domains for homogeneous groups, we investigate the differences between homogeneous and heterogeneous groups. Again, for this analysis we focus on experiments with manual as a baseline across model groups. By focusing on groups with manuals, we control that all models have explicit knowledge of both domains.

Homogeneous strong groups perform better than heterogeneous weak-strong groups Homogeneous strong groups perform the best, and homogeneous weak groups perform the worst. In the middle, heterogeneous strong-weak groups perform worse than homogeneous strong, but they perform better than homogeneous weak groups. Thus, weaker model benefit from stronger partners (see Appendix C.1).

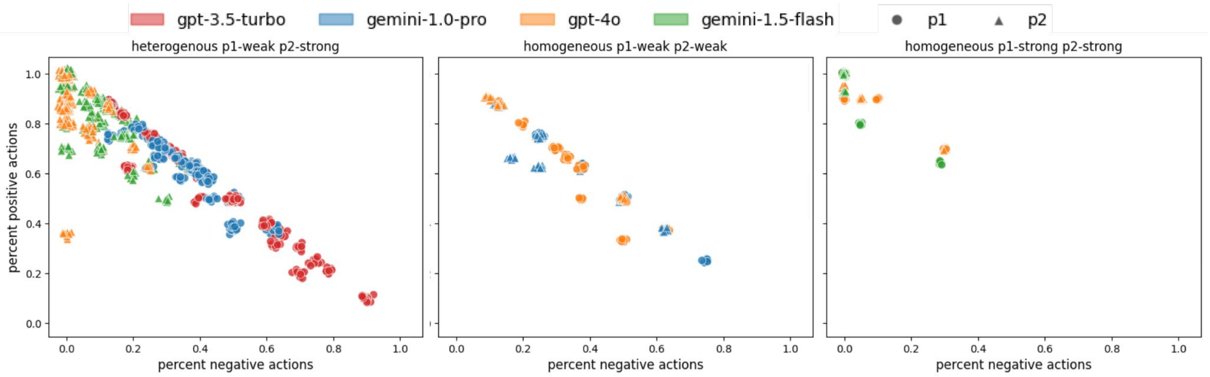


Figure 7: Plot of percent positive actions to percent negative actions across group compositions

Weaker models do not benefit from stronger models because models collaborate well together but through unbalanced participation. To examine why heterogeneous models see improvements over weaker models, we examine the distributions of positive and negative actions across players and models (Figure 7).

The distributions between weak-strong groups reveal that strong models tend to contribute to more positive actions and less negative actions. In contrast, weak models contribute to more negative actions and less positive actions. The graduation across the four models in weak-strong is consistent, showing performance is unbalanced proportional to the strength of the model. Consequently, while heterogeneous groups perform better than their weaker counterparts, it is not because strong partners increase collaborative strategies, but rather the strong model contributes to the performance.

Similarly, when comparing to homogeneous groups, we do not see as clear of a player split. Homogeneous pairs contribute more equally than heterogeneous. Thus, while models can may play well with their identical counterparts, they do not coordinate well with others.

5.3 Multi-Episode and Teacher Results

We finish our analysis by investigating the performance of the teacher framework. Initial exploration found that teachers help, except when models already have a manual. Consequently, we analyze the rest of the teacher findings only on groups without a manual, as well as report obs+manual with no teacher baseline to allow comparison between the two.

Having a teacher helps generally, but not more than a manual. Trends in both domains suggest having a teacher benefits groups, but not better than a manual (see Appendix C.2). Similar to manual findings, weak-weak groups models see marginal benefits. However, weak-strong and strong-strong groups see greater gains for teachers and manuals. The manual to teacher gap is much larger in out-of-domain Alien Rush. However, teachers give much more direct coordination strategies than the manual. This may suggest that groups cannot apply the collaboration strategies as well or that Alien Rush strategies are more difficult to generate and understand.

Trends of scaffolding across episodes can be inconsistent Across episodes, we see improvement from the teacher feedback. In the first episode, which has no feedback, the performance is consistently the lowest. Additionally, we see upward trends in performance in episode two, after the first round of feedback. However, trends can be inconsistent and are within standard deviation (see Appendix C.2). In fact, it is common for episode three to plateau or perform worse than prior episodes. This motivates future work to develop frameworks that improve performance in multi-episode tasks.

6 Conclusion

We present a new multi-agent coordination task, Kitchen-Alien Rush. By evaluating hybrid LLM teams within in-domain and out-of-domain across multiple episodes, our task addresses several critical dimensions of collaboration that prior work does not consider. We find that the proposed task environment evaluates key dimensions not addressed in prior multi-agent literature. By exposing these dimensions in our evaluation, we lay the foundation for future work to address gaps in performance of heterogeneous LLM groups in long horizon collaborative tasks.

7 Limitations

While this work contributes and expands on prior literature in collaborative multi-agent tasks for LLMs, it has several limitations which we detail below.

Limitations of Simulation We limit our current benchmark in several ways that reduce the scope of our evaluation. Firstly, we restrict our task to text-only simulation. In reality, collaboration is a complex, multimodal interaction that cannot be captured solely through written communication. Effective collaboration includes verbal and non-verbal cues, such as body language, tonality of speech, and joint attention, all of which ground communication and coordination. Secondly, our task is constrained to a fixed turn-based interaction, which cannot capture aspects of multi-agent tasks such as interjection and simultaneous coordination. And lastly, our simulation does not include human interaction, which limits our benchmark's ability to transfer findings to human collaboration applications. While we did not implement other modalities in the scope of this work, we choose to base our simulation on a real board game, Kitchen Rush, to allow for future work on this dimension.

Lack of Finetuning We do not evaluate the effect of fine-tuning in our task. While we do assess the foundational capability of LLMs which is necessary in evaluating the implicit knowledge of models in this task, this approach limits our understanding of an LLM's potential performance with training.

References

David Adamson, Gregory Dyke, Hyeju Jang, and Carolyn Penstein Rosé. 2014. Towards an agile approach to adapting dynamic collaboration support to student needs. *International Journal of Artificial Intelligence in Education*, 24:92–124.

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. 2024. Llm-coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. *Preprint*, arXiv:2310.03903.

Nolan Bard, Jakob N. Foerster, A. P. Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, H. Larochelle, Marc G. Bellemare, and Michael H. Bowling. 2019. The hanabi challenge: A new frontier for ai research. *ArXiv*, abs/1902.00506.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. 2019. *On the utility of learning about humans for human-AI coordination*. Curran Associates Inc., Red Hook, NY, USA.

Ulrike Cress, Carolyn Rosé, Alyssa Friend Wise, and Jun Oshima. 2021. *International handbook of computer-supported collaborative learning*, volume 19. Springer.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2023. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *ArXiv*, abs/2312.11970.

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, and Jianfeng Gao. 2023. Mindagent: Emergent gaming interaction. *Preprint*, arXiv:2309.09971.

Google. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang, Thomas L Griffiths, and Mengdi Wang. 2024. Embodied llm agents learn to cooperate in organized teams. *arXiv preprint arXiv:2403.12482*.

Jeongyun Han, Kwan Hoon Kim, Wonjong Rhee, and Young Hoan Cho. 2021. Learning analytics dashboards for adaptive support in face-to-face collaborative argumentation. *Computers & Education*, 163:104041.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. *Preprint*, arXiv:2308.00352.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. 2020. "Other-play" for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4399–4410. PMLR.

Hyoungwook Jin, Seonghee Lee, Hyungyu Shin, and Juho Kim. 2024. Teach ai how to code: Using large language models as teachable agents for programming education. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–28.

Maria Karyotaki, Athanasios Drigas, and Charalabos Skianis. 2022. Chatbots as cognitive, educational, advisory & coaching systems. *Technium Social Sciences Journal*, 30:109–126.

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. 2023. Avalonbench: Evaluating LLMs playing the game of avalon. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Jijia Liu, Chao Yu, Jiaxuan Gao, Yuqing Xie, Qingmin Liao, Yi Wu, and Yu Wang. 2023. Llm-powered hierarchical language agent for real-time human-ai coordination. *Preprint*, arXiv:2312.15224.

Qianou Ma, Hua Shen, Kenneth Koedinger, and Tongshuang Wu. 2024. How to teach programming in the ai era? using llms as a teachable agent for debugging. *Preprint*, arXiv:2310.05292.

Zhao Mandi, Shreeya Jain, and Shuran Song. 2023. Roco: Dialectic multi-robot collaboration with large language models. *Preprint*, arXiv:2307.04738.

Atharva Naik, Jessica Ruhan Yin, Anusha Kamath, Qianou Ma, Sherry Tongshuang Wu, Charles Murray, Christopher Bogart, Majd Sakr, and Carolyn P. Rose. 2024. Generating situated reflection triggers about alternative solution paths: A case study of generative ai for computer-supported collaborative learning. *Preprint*, arXiv:2404.18262.

OpenAI. 2023. Gpt-4 technical report.

Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, Bangcheng Yang, Pring Wong, Yifan Zhong, Xiaoyuan Zhang, Zhaowei Zhang, Nian Liu, Wei Wang, Yaodong Yang, and Song-Chun Zhu. 2024. Civrealm: A learning and reasoning odyssey in civilization for decision-making agents. *Preprint*, arXiv:2401.10568.

Yujia Qin, Shi Liang, Yining Ye, Kunlun Zhu, Lan Yan, Ya-Ting Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Marc H. Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv*, abs/2307.16789.

Zijing Shi, Meng Fang, Shunfeng Zheng, Shilong Deng, Ling Chen, and Yali Du. 2023. Cooperation on the fly: Exploring language agents for ad hoc teamwork in the avalon game. *Preprint*, arXiv:2312.17515.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Neural Information Processing Systems*.

David Turczi and Vangelis Bagiartakis. 2017. Kitchen rush. Board game. Illustrated by Gong Studios.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Daniel Weitekamp, Zihuiwen Ye, Napol Rachatasumrit, Erik Harpstead, and K. Koedinger. 2020. Investigating differential error types between human and simulated learners. *Artificial Intelligence in Education*, 12163:586 – 597.

Yue Wu, So Yeon Min, Shrimai Prabhumoye, Yonatan Bisk, Ruslan Salakhutdinov, Amos Azaria, Tom M. Mitchell, and Yuan-Fang Li. 2023. Spring: Studying the paper and reasoning to play games.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2024. Exploring large language models for communication games: An empirical study on werewolf. *Preprint*, arXiv:2309.04658.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

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. 2024a. Proagent: Building proactive cooperative agents with large language models. *Preprint*, arXiv:2308.11339.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. 2024b. Building cooperative embodied agents modularly with large language models. *ICLR*.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. 2024c. Exploring collaboration mechanisms for llm agents: A social psychology view. *Preprint*, arXiv:2310.02124.

Zhonghan Zhao, Kewei Chen, Dongxu Guo, Wenhao Chai, Tian Ye, Yanting Zhang, and Gaoang Wang. 2024. Hierarchical auto-organizing system for open-ended multi-agent navigation. *Preprint*, arXiv:2403.08282.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*.

A Example Prompts

We provide several full examples of Kitchen Rush (Figure 8 and 10) and Alien Rush Prompts (Figure 9 and 11)

B Details of Metrics

In this appendix, we elaborate on the details of our metrics in Table 2 and how they measure collaboration performance in our task.

Group Performance Our metrics capture overall group performance with the total number of completed recipes and ruined recipes. Throughout our evaluation, we refer to score as the number of completed recipes in an episode. This score serves as a quick reference to estimate success and failure in an episode. Moreover, these metrics also evaluate graded success on a spectrum in which groups that complete all five recipes and ruin none are the most successful whereas group complete zero recipes and ruin all five recipes are the least successful.

Efficiency We evaluate efficiency by tracking the turns to complete or ruin the recipes. Evaluating total turns taken in an episode may proxy measurement of coordination level between groups. Groups that complete more recipes at lower turn levels suggest they coordinated better whereas groups that terminate early due to ruining all recipes signal lack of understanding of the game. Moreover, groups that perform as single actors rather than in coordinated groups will have longer episodes.

Contribution Balance Lastly, we measure individual metrics of positive, negative, and neutral action counts to measure contribution balance of each agent. While certain groups may perform well in our scenarios, individual metrics allow us to capture how each agent contributes to the performance. With this metric, we can distinguish between groups with balanaced coordination versus groups with inbalanced participation in which one model greedily contributes to the majority of performance.

C Additional Analysis Figures

This section includes additional figures references in our analysis.

C.1 Average Completed Recipes Across Different Model Pairs (Figure 12)

C.2 Teacher Performance Across Episodes (Figure 13)

C.3 Teacher Performance across group strength, composition, and domain (Figure 14)

```
"The game Kitchen Rush has the following rules:
- The goal of the game is to finish all the orders. You score points for
finishing orders and lose points for ruining orders.
- Players work together to finish orders. To finish an order properly,
players must follow its corresponding recipe. Orders are finished when
its ingredients, spices and cook time matches the corresponding recipe.
- There are 6 kinds of ingredients (meat, carrot, lettuce, bread, pasta,
and cheese) and 5 kinds of spices (green, black, white, yellow, and
red).
- If an order is cooked before its ingredients are collected, it is
ruined. If an order is ruined, it cannot be finished.
- There are 3 types of board areas: pantries, spicebags, and stoves.
Each board area has a limited number of action spaces. If there are no
available action spaces, the board area cannot be used. Action spaces
reset to available after a turn is finished.
- Players have two workers which can carry out one action per turn.
- Players can only choose action for their workers from the Valid
Actions

***Actions***
1. Take Ingredients: Collect ingredients for an order from the specified
pantry. Removes one available action space from pantry.
2. Take Spices: Collect spices for an order from specified spicebag.
Removes one available action space from spicebag.
3. Cook Order: Cook a meal by one cooking time on specified stove.
Removes one available action space from stove.

*** The game ends when: ***
- All the orders are either finished or ruined OR
- You have played all 10 turns of the game.

These conventions will help when playing the game:
1. ***Take Ingredients before Cooking Order***
- Only cook an order if the order already has its ingredients to avoid
ruined orders
2. ***Do Not Work on Ruined or Finished Orders***
- If an order is ruined or finished, you do not need to work on it. Do
not do actions on ruined orders or finished orders
3. ***Avoid Ruining Orders**
- Do not cook an order before its ingredients are collected.
```

Figure 8: Full Kitchen Rush manual

```
The game Alien Rush has the following rules:
- The goal of the game is to finish all the irenkels. You score points
for finishing irenkels and lose points for ruining irenkels.
- Players work together to finish irenkels. To finish an irenkel
properly, players must follow its corresponding blintar. Irenkels are
finished when its ayplixs, plurns and jantrox time matches the
corresponding blintar.
- There are 6 kinds of ayplixs (zarvex, fluxin, shonix, flozix, grintal,
and brentix) and 5 kinds of plurns (drikel, quentor, froxen, glivent,
and plorix).
- If an irenkel is jantroxed before its ayplixs are collected, it is
ruined. If an irenkel is ruined, it cannot be finished.
- There are 3 types of board areas: drovus, quelixths, and yorvexs. Each
board area has a limited number of action spaces. If there are no
available action spaces, the board area cannot be used. Action spaces
reset to available after a turn is finished.
- Players have two workers which can carry out one action per turn.
- Players can only choose action for their workers from the Valid
Actions

***Actions***
1. Take Ayplixs: Collect ayplixs for an irenkel from the specified
drovus. Removes one available action space from drovus.
2. Take Plurns: Collect plurns for an irenkel from specified quelixth.
Removes one available action space from quelixth.
3. Jantrox Irenkel: Jantrox a pokem by one jantroxing time on specified
yorvex. Removes one available action space from yorvex.

*** The game ends when: ***
- All the irenkels are either finished or ruined
- You have played all 11 turns of the game.

These conventions will help when playing the game:
1. ***Take Ayplixs before Jantroxing Irenkel***
- Only jantrox an irenkel if the irenkel already has its ayplixs to
avoid ruined irenkels
2. ***Do Not Work on Ruined or Finished Irenkels***
- If an irenkel is ruined or finished, you do not need to work on it. Do
not do actions on ruined irenkels or finished irenkels
3. ***Avoid Ruining Irenkels**
- Do not jantrox an irenkel before its ayplixs are collected.
```

Figure 9: Full Alien Rush Manual

```
Recipes:
Fettucine Alfredo:
   ingredients: {'pasta': 2, 'cheese': 1, 'meat': 2}
   spices: {'white': 1, 'black': 1}
   cook time: 1

Orders:
Fettucine Alfredo:
   ingredients: not collected
   spices: not collected
   cook time: 0
   finished: False
   ruined: False

Board Areas:
pantry:
   area_type: pantry
   area_id: pantry
   ingredients: {'meat': 40, 'carrot': 40, 'lettuce': 40, 'bread': 40, 'pasta': 40, 'cheese': 40}
   available_action_spaces: 4

spicebag:
   area_type: spicebag
   area_id: spicebag
   spices: {'green': 12, 'black': 12, 'white': 12, 'red': 8, 'yellow': 8}
   available_action_spaces: 4

stoves:
   area_type: stoves
   area_id: stoves
   available_action_spaces: 4

Valid Actions:
   0.{'action_type': 'TAKE_INGREDIENTS', 'area_id': 'pantry', 'value': {'order': 'Fettucine Alfredo'}}
   1.{'action_type': 'TAKE_SPICES', 'area_id': 'spicebag', 'value': {'order': 'Fettucine Alfredo'}}
   2.{'action_type': 'COOK_MEAL', 'area_id': 'stoves', 'value': {'order': 'Fettucine Alfredo'}}
```

Figure 10: Sample Kitchen Rush Observation

```
Blintars:
Zarnorblat Alfrek:
   ayplixs: {'grintal': 2, 'brentix': 1, 'zarvex': 2}
   plurns: {'froxen': 1, 'quentor': 1}
   jantrox time: 1

Irenkels:
Zarnorblat Alfrek:
   ayplixs: not collected
   plurns: not collected
   jantrox time: 0
   finished: False
   ruined: False

Board Areas:
drovus:
   area_type: drovus
   area_id: drovus
   ayplixs: {'zarvex': 40, 'fluxin': 40, 'shonix': 40, 'flozix': 40, 'grintal': 40, 'brentix': 40}
   available_action_spaces: 4

quelixth:
   area_type: quelixth
   area_id: quelixth
   plurns: {'drikel': 12, 'quentor': 12, 'froxen': 12, 'plorix': 8, 'glivent': 8}
   available_action_spaces: 4

yorvexs:
   area_type: yorvexs
   area_id: yorvexs
   available_action_spaces: 4

Valid Actions:
   0.{'action_type': 'TAKE_AYPLIXS', 'area_id': 'drovus', 'value': {'irenkel': 'Zarnorblat Alfrek'}}
   1.{'action_type': 'TAKE_PLURNS', 'area_id': 'quelixth', 'value': {'irenkel': 'Zarnorblat Alfrek'}}
   2.{'action_type': 'JANTROX_POKEM', 'area_id': 'yorvexs', 'value': {'irenkel': 'Zarnorblat Alfrek'}}
```

Figure 11: Sample Alien Rush Observation

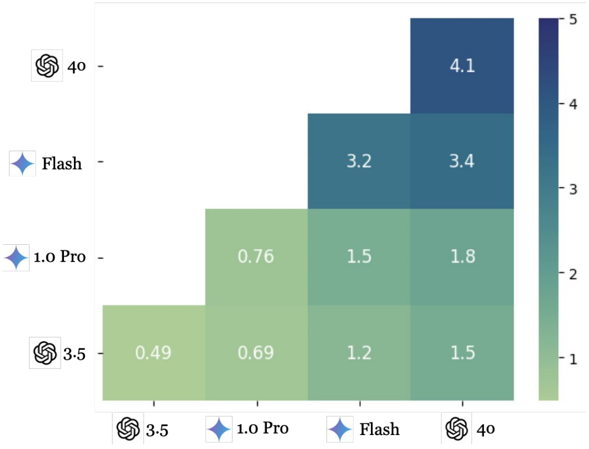


Figure 12: Average completed recipes between varying model group pairings

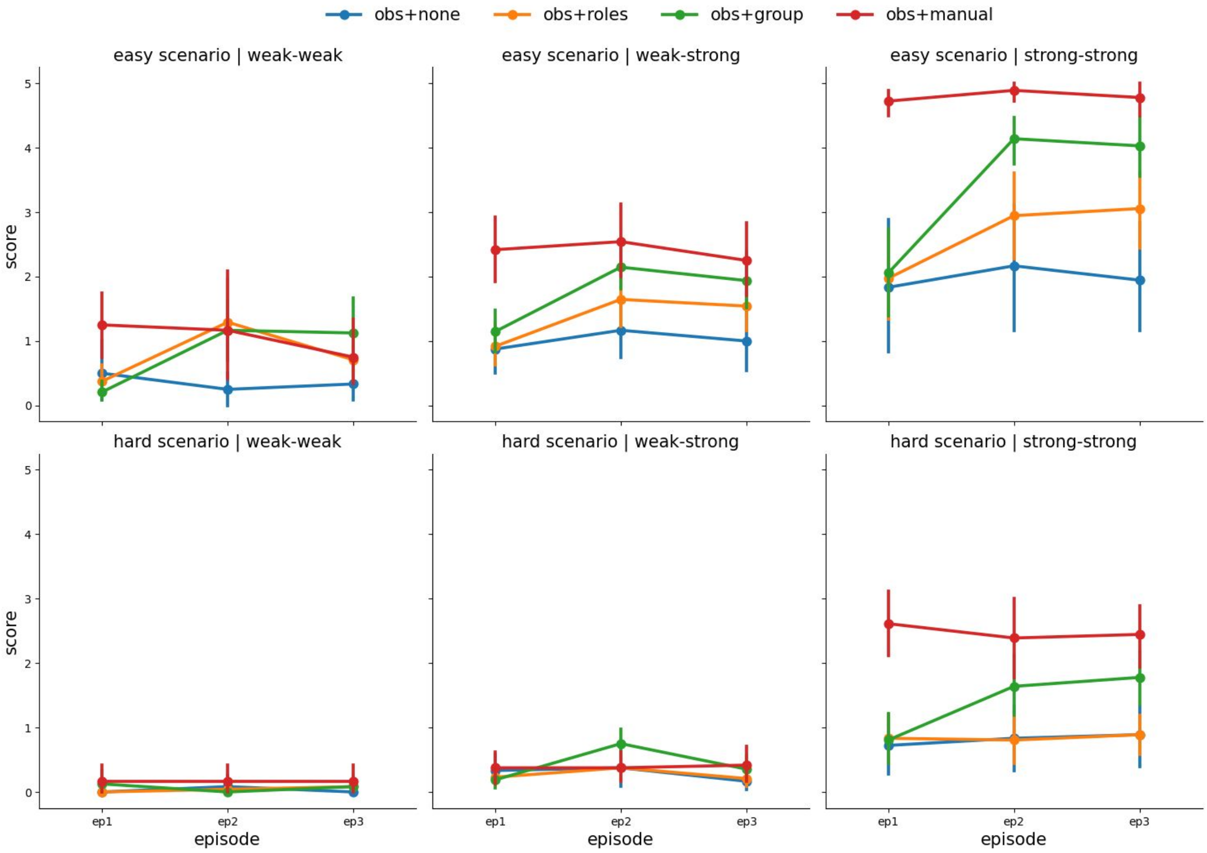


Figure 13: Average completed recipe performance across episodes for observation only agents with a teacher

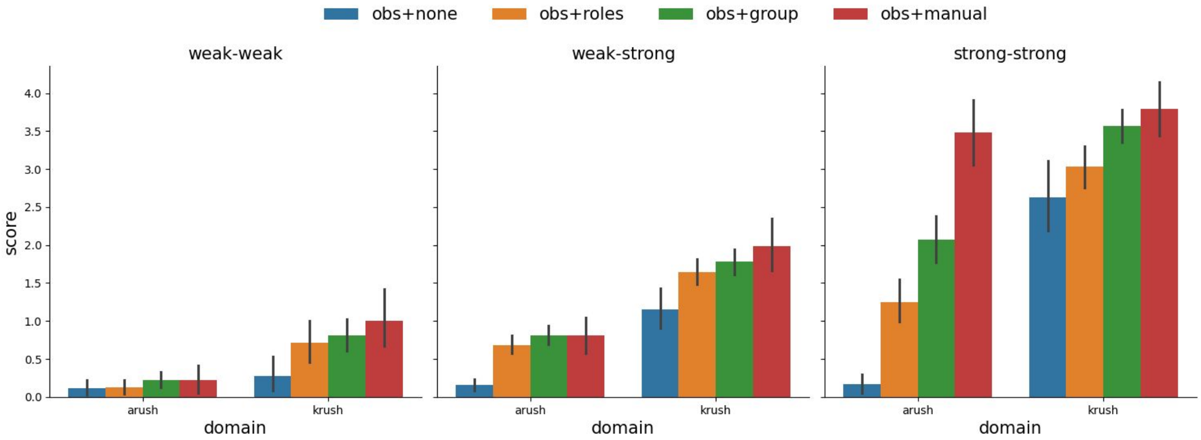


Figure 14: Average score performance for observation only agents with a teacher. Results are divided across group strength composition and domain.