RATFTOUILLE: EVALUATING EMOTIONAL AND GOAL-DIRECTED REASONING CAPABILITIES OF LANGUAGE MODELS USING COOKING RECIPES

Anonymous authors
Paper under double-blind review

ABSTRACT

While much of the current study of LLM reasoning is focused on goal-directed reasoning and emotional reasoning as two separate tasks, there are many real-world scenarios where LLMs are required to do them jointly. For example, if a user deletes a photo and then expresses strong sadness, an assistant should infer that the deletion was likely unintended and that the user wants the photo recovered. In cases like these, an LLM must infer a user's emotional state and use that inference to perform reasoning on their goals. In order to systematically evaluate LLMs abilities to perform emotionally contextualized goal-directed reasoning (ECGD), we propose the Recipe-Actor Transcript Framework (RATF), a framework that procedurally generates step-by-step transcripts of simulated actors following recipes to make food, where the recipes are sourced from the RECIPENLG dataset. We utilize these transcripts to test whether an LLM can correctly identify the intents and goals of a simulated actor, and predict their next action. More precisely, we apply RATF to study two LLM abilities: (1) goal prediction, where LLMs must infer the actor's goal given their past behavior, i.e., what are they trying to cook, and (2) next-action prediction, where we evaluate the accuracy of the LLM's predicted next-action distribution of the simulated actor when given in-context examples of actor behavior. We test several frontier LLMs and find that LLMs struggle with these tasks, and their performance is highly sensitive to surface-level variations in the way the prompt is presented. We find evidence suggesting that LLMs rely on heuristics to solve this task rather than performing true ECGD reasoning.[1]

1 INTRODUCTION

Emotions are not merely labels or static states to be recognized; they provide evidence about a person's underlying goals. A passive-aggressive response from a friend while making plans indicates they do not want to participate. An LLM coding assistant, after notifying the user that it is executing a (potentially harmful) script, needs to be able to infer from the user's negative emotional response that it must stop execution of the script (Mansoor, 2025). Children with language disorders often struggle to communicate with others, so caretakers in school settings are tasked with inferring their needs and goals using non-verbal emotional cues (Cross, 2011). Each of these scenarios requires not just recognizing the emotion displayed by the user, but utilizing it as information to infer the goal of that user.

As LLMs are increasingly deployed to interact with humans and other agents (Doherty et al., 2025; Berrezueta-Guzman et al., 2024; He et al., 2024; Yang et al., 2026; Guo et al., 2025; Bo et al., 2024), it is critical that we understand how well they can utilize inferred emotional states to reason about others' goals. This leads us to our high-level research question: *Are current frontier LLMs capable of inferring the emotions of users and utilizing that information to reason about their goals and behavior*? There are existing benchmarks that aim to measure LLM's goal-directed reasoning and emotional intelligence abilities (Sabour et al., 2024; Chen et al., 2024; Paech, 2023), but these very same benchmarks suffer from training data contamination (Chen et al., 2025), and further evaluations of both of these abilities *in tandem* as we describe have yet to exist.

[1] All code and data generation scripts are freely available at `[redacted]`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

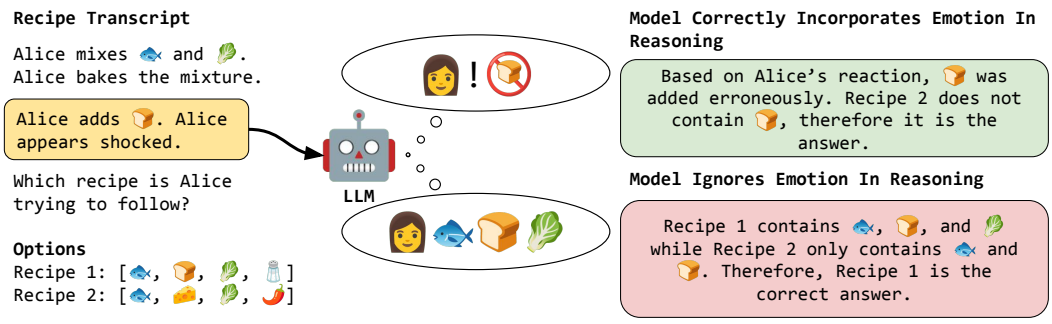

Figure 1: A diagram of the mistaken ingredient test. Given a transcript of an actor adding a mistaken ingredient in the intermediate steps of a recipe, the model must (a) identify the actor's mistake based on their emotional reaction, and (b) use this mistake reaction to determine the recipe the actor is attempting to make.

To this end, we present RATF, the **R**ecipe-**A**ctor **T**ranscript **F**ramework. In order to test LLMs on emotionally contextualized goal-directed reasoning (ECGD), RATF generates transcripts of simulated actors cooking food by following a recipe, and occasionally performing steps that evoke emotional reactions. Though the initial recipe data comes from the `RecipeNLG` dataset (Bień et al., 2020), each *transcript*, a step-by-step sequence of recipe actions by the actor, is newly generated after parsing and filtering the data, so the transcripts are novel and guaranteed to be free from training data contamination, enabling the evaluation of the ECGD capabilities of LLMs in a contamination-free fashion. RATF provides us with a high degree of control over the properties of the simulated actors and recipe transcripts, enabling systematic evaluation where we can carefully study the effect of one variable while tightly controlling for others. Using RATF, we seek to answer the following research questions:

RQ1 Given observations of a user's actions, can LLMs predict the user's emotions, and further, the user's goals?

RQ2 Given observations of a user's history of actions and emotional cues, can LLMs predict that user's next action?

In our experiments, we test LLMs on ECGD reasoning by asking them to infer which recipe a simulated human actor is trying to follow based on a transcript of their actions, where the actor makes a mistake on one of the steps and reacts emotionally. The only way the LLM can answer the question correctly is if it bases its reasoning off of the emotional reaction. Additionally, we give LLMs the task of predicting how simulated actors will react after making a mistake while following a recipe. In this task, the LLMs are given a transcript of the actor's actions in the current recipe so far, as well as a set of in-context examples containing transcripts of the actor following other recipes, so the LLM must synthesize information about the actor across the in-context examples in order to reason about how the actor will respond to the new mistake.

We find that LLMs' ability to determine which recipe an actor is trying to follow from a multiple-choice set of recipes is strongly dependent on the composition of the multiple-choice set of recipes we present. We present the LLMs with different numbers of recipe options to choose from, each of which have similar but not identical ingredients, and find that the LLM's accuracy strongly depends on the number of options with which they're presented. Additionally, we find that LLMs struggle to utilize information given in in-context examples to accurately predict the actions of a simulated actor. Our framework and its details are presented in Section 3, including a description of the experiments we run in Section 3.2. In Section 4, we analyze our results, perform ablation studies with model reasoning effort, and investigate model sensitivity with respect to variations in emotion verbalization and non-emotional reactions to mistakes.

2 Related Work

Emotional Intelligence and Reasoning in LLMs. Emotional capabilities in language models have traditionally been evaluated through tasks such as sentiment and emotion classification (Mohammad

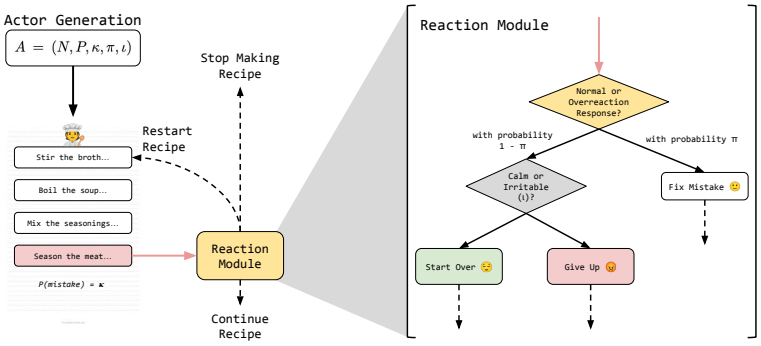

Figure 2: When simulating an actor's reaction to a mistake, an actor's personality values determine their reaction. An actor's "perfectionism" value π determines how often they choose to fix mistakes and continue with the recipe; π ranges from 0 (always continues) to 1 (never continues). An actor's irritability value ι is a binary parameter which determines if an overreaction is calm (restarting the recipe) or irritable (giving up on the recipe).

et al., 2018). More recent benchmarks test emotional intelligence by testing whether LLMs can correctly classify a person's emotions (Chen et al., 2024; Paech, 2023). In EMOBENCH, human annotators are asked to create scenarios that allow authors to evaluate LLM emotional intelligence via classification, and further ask LLMs to use the classified emotion to solve a social reasoning task (Sabour et al., 2024; Sreedar et al., 2026). A parallel line of work studies fine-grained emotion recognition in multimodal models using combinations of facial expressions, text, and speech (Rha et al., 2026; Lian et al., 2025; Cheng et al., 2024).

Emotional intelligence also appears in broader benchmarks for social and theory-of-mind reasoning, since in these settings, LLMs must reason through the belief states of agents in order to determine their emotions. Such studies include BIGTOM (Gandhi et al., 2023), SOCIAL-IQA (Sap et al., 2019), and DYNTOM (Xiao et al., 2025), where they present scenarios in which models must reason about agents' beliefs, intentions, actions, and, in some cases, emotional responses.

Previous emotional intelligence and reasoning benchmarks discussed above generally treat emotion recognition or social-state inference as an endpoint. Our benchmark requires LLMs to reason over actor's goals and emotions jointly, as they must use recognized emotions as evidence for inferring the simulated actor's goal or future behavior.

The prior work most similar to ours is EVENT2MIND (Rashkin et al., 2018), which predicts agents' intents and emotional reactions from short event descriptions. Although EVENT2MIND connects actions, intentions, and reactions, it differs from our setting in two important respects. First, intent and reaction are predicted as separate outputs rather than requiring the model to use an observed emotional reaction as evidence to infer an actor's goal. Second, EVENT2MIND is a fixed publicly available dataset that predates modern LLMs, and is therefore highly likely to be included in LLM training data. Further, its descriptions come from scraped blog posts, stories, and Wikipedia entries. Our framework generates novel simulations of actors, behavioral parameters, and action traces, which greatly mitigates the risk of examples having been seen in pre-training data. We explore the extent to which EVENT2MIND may have been leaked into pretraining data in Appendix B.

Goal-directed Reasoning in LLMs. Goal-directed reasoning is the process of reasoning about observed and potential actions in an environment, and reasoning about which actions lead to progress toward some goal. Existing work on LLMs' ability to reason about the goals of an observed agent often use *board games* as testbeds, since roles and objectives are clearly defined (Costarelli et al., 2024; Qiao et al., 2023). Additionally, existing work also evaluates whether LLMs can deduce the goals of users via multi-turn conversation (Coscia et al., 2025) or static human-written comments (Zhu et al., 2024). However, most of the existing work that evaluates the goal-directed reasoning abilities of LLMs focuses on whether the LLMs themselves can perform goal-directed reasoning under various environments (Arghal et al., 2026) or long contexts (Bidochko & Vyklyuk, 2025; Tan et al., 2024; Valmeekam et al., 2022). Unlike prior work, we evaluate whether LLMs can infer an

actor's goal from incidental behavioral and emotional evidence, both of which are signals that are not explicitly intended to convey either the goal itself or the agent's progress towards achieving it.

Cooking as a Testbed for Measuring AI Capabilities. Cooking provides a natural environment for studying goal-directed behavior since recipes specify well-defined recognizable goals and recipe steps provide partially ordered sequences of required actions (i.e., intermediate steps). Prior cooking datasets have represented recipes as structured programs or action graphs (Papadopoulos et al., 2022), aligned alternative textual and visual realizations of the same procedure (Lin et al., 2020), and evaluated extraction of structured actions from instructional videos (Xu et al., 2020). Shin et al. (2024) most closely resembles the experimental portion of our study where they train an LLM to deduce subgoals from cooking recipes, using training data generated by an LLM given a recipe. These resources primarily target procedural understanding, generation, retrieval, or perception rather than the inference of an actor's latent goal and behavioral characteristics. RATF instead uses structured recipe data as the basis of a controlled social-reasoning environment.

3 METHODS

Cooking data and recipes provide a natural source of detailed step-by-step instructions with a clear singular goal, rendering such data highly appropriate for studying goal-directed reasoning. Recipe data also provide an excellent case study for emotional reasoning, since people who follow recipes often react emotionally in the process of (possibly imperfectly) executing the recipe. There are substantial cooking and recipe data available to readily facilitate our exploration; RECIPENLG (Bień et al., 2020) alone contains over one million instances of structured recipe data sourced from across the internet. The cooking testbed also enables the generation of recipe transcripts, where simulated actors perform the steps in the recipe. The use of simulated actors ensures that LLMs have zero a-priori knowledge of examples beyond general understanding of cooking, thereby mitigating data contamination. The following experiments utilize recipe transcripts, which are generated procedurally from simulated actors and real-world recipes.

3.1 FRAMEWORK DESIGN

The RECIPENLG dataset contains recipe titles, ingredient lists, and natural-language instruction lists. To generate transcripts for our experiments, we first convert each recipe into a structured representation that explicitly records the inputs, action, and output of each step. We then simulate an actor following this structured recipe, eventually making and reacting to a mistake. For the next-step prediction experiments, we additionally define a behavioral model that determines the actor's reaction to that mistake.

We refer to each example from RECIPENLG as a *ground-truth recipe* $G = (I, S)$, where I is the list of base ingredients and $S = (S_1, \ldots, S_n)$ is the sequence of natural-language instructions. From G, we construct a structured recipe $R = (R_1, \ldots, R_n)$ sequentially.

We begin by initializing the set of available ingredients as $U_1 = I$. For each step $k = 1, \ldots, n$, we apply a parser f, implemented using `gpt-5.4-mini`, to the current available ingredients U_k and instruction S_k:

$$R_k = f(U_k, S_k) = (C_k, V_k, O_k).$$

Here, $C_k \subseteq U_k$ denotes the ingredients or intermediate products used in step k, V_k is the corresponding action expressed as a present-tense verb (e.g "bake"or "mix"), and O_k is a short description of the resulting intermediate product. For example, a step may map ingredients such as `crab` and `seasoning` to the action `mix` and the output `seasoned crab mixture`. The system prompt used by f is provided in Appendix Figure 9. After parsing each step, we update the set of available ingredients by removing the consumed inputs and adding the newly produced output:

$$U_{k+1} = (U_k \setminus C_k) \cup O_k.$$

This procedure is repeated for each step in S, producing the structured recipe R, which is then used to generate the recipe transcripts in our experiments.

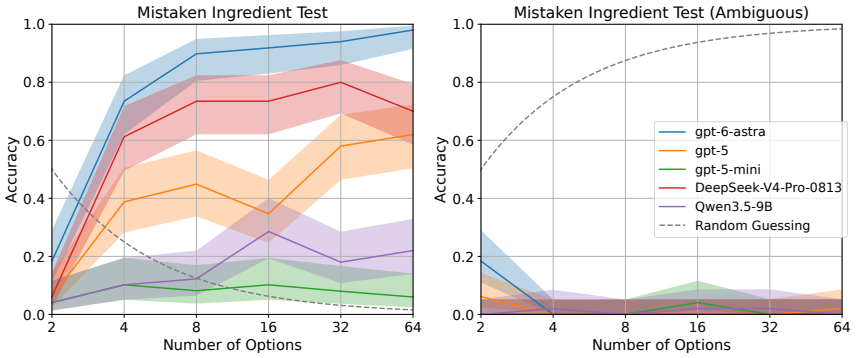

Figure 3: Each trial, LLMs are shown a transcript of an actor following a hidden recipe without mistake behavior enabled and are asked to predict the recipe from a multiple-choice list. We vary the number of given multiple-choice options and run 50 trials per option number setting. (**Left**) Recipe options are identical except for the last ingredient, but there is only one correct answer. (**Right**) There are multiple correct answers, and only one incorrect answer. Shading indicates a 95% confidence Wilson's interval.

Actors in our framework are defined as a tuple $A = (N, P, \kappa, \pi, \iota)$, where N and P are the actor's name and subject pronoun, respectively; κ, π, and ι are parameters that characterize the actor's "personality". During the transcript generation, the actor may perform *mistake* actions, where they perform an action that does not follow the recipe. An actor's *coordination* value $\kappa \in [0, 1]$ determines how likely an actor is to make and notice a mistake after each step of the recipe (e.g. for $\kappa = 0.8$, after each step, there is a 20% chance that the actor will make and notice a mistake). When an actor performing the recipe makes a mistake, the perfection π and irritability ι values determine how an actor reacts. As shown in Figure 2, perfectionism $\pi \in [0, 1]$ is the probability that the actor will fix the mistake and continue with the recipe. Given the actor has made a mistake, the binary irritability value $\iota \in \{+1, -1\}$ determines whether the actor's reaction is calm (restarting the recipe; when $\iota = +1$) or angry (giving up on making the recipe; when $\iota = -1$). One difficulty of modeling personalities and action distributions of actors is making sure transcripts are sufficiently long for the LLM to be able to infer the personality of the actor from their actions. If an actor's personality model is highly complex, with many parameters, it is unlikely that an LLM reading the recipe transcripts can accurately estimate the parameters of the actor's personality model, at least not without a substantial number of recipe transcripts. As such, we have chosen to represent actors and their personalities in this simplified manner to avoid having to create extremely long transcripts and confounding with the LLM's ability to estimate personality parameters from very long-context inputs.

When combined with the specifications of the actor A, each step k of an structured recipe can be converted into a natural language sentence. For example, "Alice mixes 2 flounder fillets, roughly 1 lb., 3 tsp. salt, and 5 tsp. pepper, producing some seasoned fillets.". The template is written more generally: "[N_A] [V_k] [C_k] producing [O_k]" where N_A is the name of the actor A making the recipe. Converting the full set of steps into natural language sentences produces a full natural language recipe transcript of an actor following a recipe step-by-step. For the following tests, we generate recipe transcripts using ground-truth recipes from a subset of the RecipeNLG dataset (we consider all recipes with 8-32 steps inclusive and 8-32 ingredients inclusive). RecipeNLG recipes with more than 32 steps or ingredients were often noisy or artifacts of preprocessing errors. Similarly, we required the selected recipes to have at least 8 steps or ingredients in order to avoid generating trivial examples.

In our experiments, the LLM is given a textual transcript of a simulated actor A following a recipe until they add an ingredient which does not belong, and outwardly reacts to their mistake. In each transcript, the actor only performs $k < n$ structured recipe steps before making a mistake. k is small enough that not all ingredients in I have been used. In our experiments, the recipe transcript is at most 4 steps long to avoid outliers in transcript length.

3.2 EXPERIMENTS

3.2.1 RECIPE IDENTIFICATION WITH MISTAKEN INGREDIENT

In this experiment, the model must identify the recipe that the actor intended to follow from a set of multiple-choice options. We generate each example procedurally from a ground-truth recipe $G = (I, S)$.

First, we select the mistaken ingredient m. We sample m uniformly from the base ingredients in I that have not yet appeared in the transcript. We then use `gpt-5.4-mini` to generate a set $E = \{e_1, \ldots, e_n\}$ of plausible replacement ingredients that do not occur in I. The corresponding prompt is provided in Appendix Figure 10.

Next, we construct the candidate recipe options. Each option is obtained by replacing exactly one ingredient in I with an ingredient $e_k \in E$. For option k, we sample an ingredient $i_k \in I$ and define

$$\Omega_k = (I \setminus \{i_k\}) \cup \{e_k\}, \qquad k = 1, \ldots, n.$$

An option is labeled correct when $i_k = m$, and incorrect when $i_k \neq m$. Thus, every candidate differs from the ground-truth ingredient list by exactly one ingredient, while the correct candidates are precisely those in which the mistaken ingredient has been removed. Replacing rather than simply deleting ingredients also allows us to generate many distinct candidate lists while keeping all options the same size and at the same edit distance from I.

For example, suppose the intended recipe is a sugar cookie recipe with ingredient list $I =$`[flour, butter, sugar, eggs]`, and the actor mistakenly adds $m =$ `almonds`. Candidate recipes might include `[flour, butter, sugar, eggs]` and `[flour, butter, sugar, almonds]`. The first is correct because it excludes the mistakenly added ingredient m, whereas the second is incorrect because it includes m.

Finally, we construct the transcript so that it ends with "[N_A] added some [i']. [N_A] [R]", where N_A is the actor's name, i' is the ingredient that was added, and R is a negative reaction indicating that a mistake occurred. Names and reactions are sampled from predefined banks (Appendix Figure 11) to reduce confounding. We also evaluate non-emotional reactions in Figure 5. The correct answer is the candidate recipe whose ingredient list excludes i'. Solving the task therefore requires the LLM to use the reaction R to infer that adding i' was a mistake and identify the intended recipe accordingly. Doing so requires sound ECGD ability. Results are shown in Figure 3 (left).

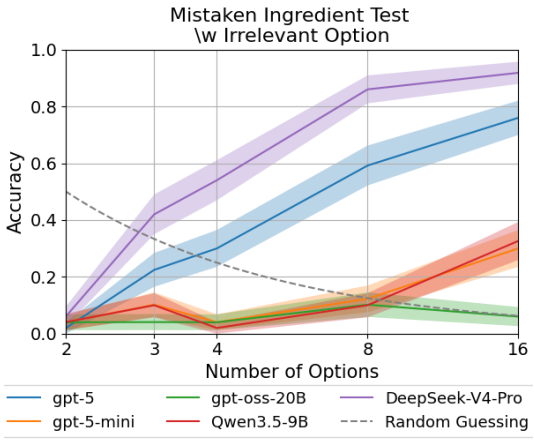

Figure 4: Mistaken Ingredient test with a fixed number of 16 total recipe options and varying proportions of ingredient-overlapping options. Accuracy generally improves with more relevant options, except when only one is relevant.

Similarly, we also study the effect of the number of relevant vs. irrelevant options (options with ingredient lists not sourced from the initial ground-truth recipe) by fixing the total number of options at 16 and varying the number of relevant options. In addition to testing the effect of multiple options, this experiment will provide evidence on whether LLMs rely on a heuristic where they focus on the ingredients mentioned in the transcript to produce its output, and give us insight into how sensitive model's ECGD ability is.

3.2.2 PERSONALITY-BASED NEXT STEP PREDICTION

In this experiment, we test frontier LLMs' capability to model the behavior of actors. This follows the same transcript construction as before, except the model's are asked to predict the actor's next action. We chose to focus on mistake reactions as they are most relevant to emotional reasoning, whereas other actions are more informed by the mechanics of cooking. All recipes chosen for in-context

and test examples are sampled uniformly at random from the subset of RECIPENLG described in Section 3. Since the LLM's task is to match the action distribution of the actor rather than finding the most likely next action, the LLM is queried multiple times with the same prompt using temperature sampling ($T = 1$) and an empirical distribution of the predicted next action is collected and compared to the empirical reaction distribution gathered from mistake reactions in the in-context examples.

We run three versions of this experiment, each time we vary the type of in-context examples provided to the LLM. In each, the LLM is provided with five in-context examples of recipe transcripts, each containing a narrative of an actor (possibly imperfectly) following a recipe. The LLM is subsequently given a test example of a recipe transcript of an actor A following a different recipe. The three settings provide: (1) five examples of A following recipes, (2) five examples involving distinct actors, and (3) five examples involving distinct actors who share A's personality profile. If the LLM reproduces A's distribution over next actions following a mistake, this suggests that it can use observed behavioral history to form a model of A and extrapolate their future behavior.

4 RESULTS

In this section, we detail the results of the experiments described in Section 3.2. In our experiments, we tested `GPT-6-Astra` (OpenAI, 2026), `GPT-5` (Singh et al., 2025), `GPT-5-mini`, `GPT-OSS` (Agarwal et al., 2025), `Qwen-3.5-9B` (Qwen Team, 2026), and `DeepSeek-V4-Pro` (Xu et al., 2026). Further details on the models we use, as well as examples of prompts given to models, are provided in Appendix A.

4.1 RECIPE IDENTIFICATION WITH MISTAKEN INGREDIENT

As shown in Figure 3 (left), `DeepSeek-V4-Pro`, `GPT-5`, and `GPT-6-Astra` perform the best on the Mistaken Ingredient test, while the other three models struggle in all settings. Although the accuracy of random guessing is highest at two options, we find that these three models along with `Qwen-3.5-9B`'s performance improves with more options. As the number of options is increased beyond 16, performance either stabilizes or even declines slightly for most models. Since the correct answer remains unchanged, adding other options around it should either leave performance unaffected or make the task more difficult by introducing additional irrelevant context. However, our results show that when the models are given the least amount of options to consider, they perform their worst, with 2 options being the lowest accuracy setting across every model, and worse 4 options than 8 or more. With this, it's clear that the number of options presented to the LLMs affects their ability to ECGD reasoning. Testing multiple reasoning levels for `gpt-5` and `DeepSeek-V4-Pro` shows that models performed better at higher reasoning and worse at lower/no reasoning.

One potential explanation of our observation that the LLM's ECGD reasoning performance is so affected by increasing the total number of options beyond 2 and 3 is due to the increasing number of *incorrect* options. In order to test this, we repeated the experiment discussed above with one key difference. The LLM is still given a recipe transcript where an actor makes a mistake while following a recipe, but all of the recipe options the multiple-choice list are correct *except one*. This means the given transcript could be applied to multiple recipes. In their prompts, the models are instructed that multiple answers can be correct and their task is to just pick any correct answer (see Appendix A). We call this the *ambiguous* setting, whereas the setting above where there is only one correct answer is called the *unambiguous* setting. The results of this experiment are shown in Figure 3 (right). Interestingly, as the number of ambiguous, correct options increases, all tested models perform worse. This suggests that LLMs utilize a heuristic in goal-directed reasoning where they have a tendency to predict a less ambiguous incorrect goal rather than an ambiguous correct goal.

We also investigate models' sensitivity to the different emotional and non-emotional reactions, shown in Figure 5. These tests involved the five emotional reactions discussed previously and the non-emotional reaction "[actor] has made a mistake" as a baseline. In the 2-option setting, most models perform similarly across all types of reactions. `gpt-6-astra` appears to be the most sensitive to specific reactions, performing particularly poorly on the "shakes their head slowly" reaction. However, `gpt-5` and `gpt-5-mini` see far more variance in performance across reactions in the 4-option setting. In both settings, all models see their best performance when they are explicitly told that the actor

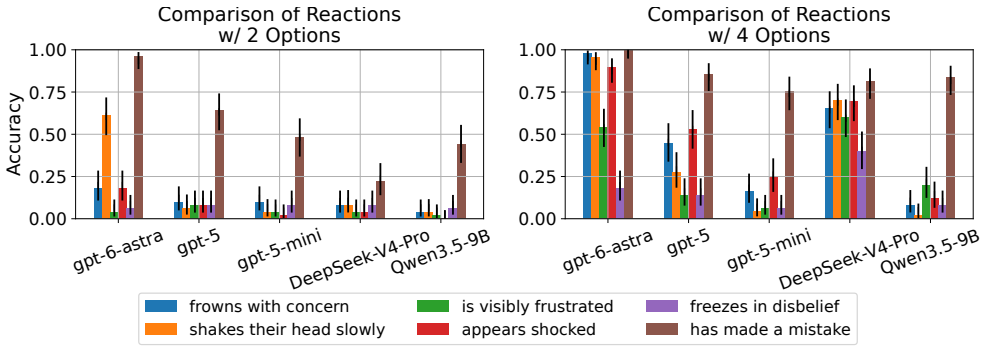

Figure 5: Testing models' sensitivity to specific reactions using the standard mistaken ingredient test with two and four given options. Error bars represent a 95% confidence Wilson's interval. Notably, the non-emotional reaction performs significantly better than all emotional reactions, particularly in the 2 option example.

made a mistake. This indicates that LLMs don't realize that the actor has made a mistake in adding the last ingredient based on their emotional reaction.

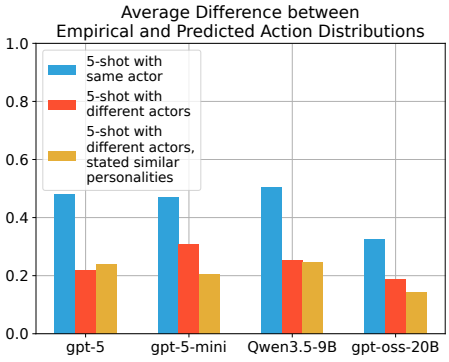

Figure 6: Average difference between the target actor's empirical and predicted action distributions under four 5-shot conditions: examples match the target's name and personality, differ in both, or use a different name with the same personality.

To further explore the relationship between the multiple-choice options that LLM's are given and their accuracy on RATF, we repeat the Mistaken Ingredient test, but fix the number of recipe options at 16, and vary the number of options that share common ingredients with the correct answer. An option with at least one common ingredient with the correct answer is categorized as a "relevant" option in our evaluation. We find that as we vary the number of relevant options out of 16 options given to LLMs (Figure 4), models perform similarly to how they would perform if that the number of relevant options was the total number of options available, as in in Figure 3 (left). This is true of all models aside from `gpt-5` and `DeepSeek-V4-Pro`, which perform the best in both settings. This provides further evidence that LLMs are using a heuristic based on considering what options share ingredients with other options, rather than doing genuine ECGD reasoning.

4.1.1 Error Analysis: How much reasoning is explained by ambiguity avoidance?

To gain insight into why models perform inversely to random guessing in the unambiguous and ambiguous tests, we performed error analysis on the reasoning traces produced by models when responding in both settings. As the OpenAI API does not provide the exact reasoning trace for responses, error analysis was performed on `DeepSeek-V4-Pro`. Analysis was performed on 10 sample responses randomly selected from the 2- and 4-option tests for both settings. As shown in Table 1, in the 2-option standard setting, `DeepSeek-V4-Pro` did not perform any reasoning on the actor's mistake reaction (beyond restating the transcript) in any samples, instead gravitating towards the heuristic of matching mentioned ingredients. At 4 options, the model struggles to select one of three wrong answers which often leads to it doubling back and considering the mistake reactions. The model occasionally fails to associate the mistake reactions with the actor mistakenly adding the associated ingredient, but succeeds often enough to greatly increase the task success rate. However, the reasoning traces generated from the 2- and 4-option tests in the ambiguous setting mirror those of the 2-option standard tests, in that models once again failed to take the actor's reaction into account in their reasoning. We take these findings to mean that while models do not necessarily make use of an "ambiguity avoidance" heuristic to select the right answer, when faced with an ambiguous set of

potentially correct answers, models tends to spend more time rethinking and exploring other lines of reasoning which may lead to the correct answer.

4.2 Personality-Based Next Action Prediction

The results of the personality-based next step prediction task are displayed in Figure 6. Across all models tested, we find that whether or not information on the target actor's personality is included makes no difference. On average, LLMs are actually *farther* from the true action distribution when given five in-context examples of the actor in question than in settings where they receive no information about the actor. LLMs perform the best when given in-context examples that actors that share the same personality parameters but have distinct names (their personalities are identical to that of the actor in the test example). This implies that the models do not have the innate ability to recognize that examples of the target actor's past actions provide insight into their emotional profile and goals. Only when they are told explicitly that their examples match the personality of the target actor do they start to utilize this information to better reason about the actor's displayed emotion and how it relates to their action.

Error Type	**2 Opt.**	**4 Opt.**
Ignores reaction	10	4
Recognizes Mistake, not ingredient	0	3
Recognizes Both, still incorrect	0	3

Table 1: Error types found in a random sample of 10 `Deepseek-V4-Pro` incorrect responses when given 2 and 4 options.

5 Conclusion

We present RATF, a framework used to procedurally generate step-by-step transcripts of simulated actors following recipes to make food. These transcripts enable us to investigate the ECGD reasoning capabilities of frontier models.

We observed that models struggle to predict simulated actor's goals given their emotions and actions. Further, changing the number of options LLMs must choose from in multiple-choice format leads to improved performance. Our experiments demonstrate that this is related to the number of relevant options the LLM thinks it must consider. Additionally, we observe that when given in-context examples demonstrating the way a simulated actor behaves, the LLM fails to utilize this to accurately predict that actor's next action after making a mistake, and instead performs similarly or even worse than if it had no information about the actor at all. This again indicates that rather than doing proper ECGD reasoning based on the inferred emotional state of the actor, LLMs rely on some other strategy to solve these problems.

While LLMs display some ECGD capabilities, we find in both of our main experiments that LLMs seem to rely on heuristics and other strategies to solve these problems. From this, we conclude that LLMs ECGD abilities are highly limited.

Limitations and Future Work

The use of `gpt-5.4-mini` in recipe transcript generation sometimes leads to sentences which are grammatically incorrect or do not flow smoothly. While this does not appear to have a major effect on results, we aim to improve the generation pipeline to support more apt output names and varied sentence structure to better model natural text.

As stated previously, the model for actor personalities used in our experiments is rather rudimentary. It is possible that given a more natural model of actor personality/behavior, LLMs will have an easier time making internal actor models to reason with. We hope to develop a more robust model of actor personalities and test an actor's full action distribution instead of just mistake reactions.

We are interested in the ambiguity avoidance heuristic exhibited by LLMs during the mistaken ingredient experiments. We intend to look into whether this phenomenon is exclusive to ECGD tasks or if it is more widespread.

GENERATIVE AI USE DISCLOSURE

In this work, we used generative AI tools for partially generating and touching up synthetic data sets. All other required disclosure tasks were either performed by hand or are not applicable to this work. Additionally, we used generative AI tools for assisting with navigating matplotlib/pyplot documentation. The generated dataset was scanned for artifacts and verified by 2 researchers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Raghu Arghal, Fade Chen, Niall Dalton, Evgenii Kortukov, Calum McNamara, Angelos Nalmpantis, Moksh Nirvaan, Gabriele Sarti, and Mario Giulianelli. A behavioural and representational evaluation of goal-directedness in language model agents. *arXiv preprint arXiv:2602.08964*, 2026.

Santiago Berrezueta-Guzman, Mohanad Kandil, María-Luisa Martín-Ruiz, Iván Pau de la Cruz, and Stephan Krusche. Future of adhd care: evaluating the efficacy of chatgpt in therapy enhancement. In *Healthcare*, volume 12, pp. 683. MDPI, 2024.

Andrii Bidochko and Yaroslav Vyklyuk. Thought management system for long-horizon, goal-driven llm agents. *Journal of Computational Science*, pp. 102740, 2025.

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wiśniewski, and Agnieszka Lawrynowicz. Recipenlg: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th international conference on natural language generation*, pp. 22–28, 2020.

Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. Reflective multi-agent collaboration based on large language models. *Advances in Neural Information Processing Systems*, 37:138595–138631, 2024.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, et al. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10091–10109, 2025.

Yuyan Chen, Songzhou Yan, Sijia Liu, Yueze Li, and Yanghua Xiao. Emotionqueen: A benchmark for evaluating empathy of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2149–2176, 2024.

Zebang Cheng, Zhi-Qi Cheng, Jun-Yan He, Jingdong Sun, Kai Wang, Yuxiang Lin, Zheng Lian, Xiaojiang Peng, and Alexander G Hauptmann. Emotion-llama: Multimodal emotion recognition and reasoning with instruction tuning. *Advances in Neural Information Processing Systems*, 37: 110805–110853, 2024.

Adam J Coscia, Shunan Guo, Eunyee Koh, and Alex Endert. Ongoal: Tracking and visualizing conversational goals in multi-turn dialogue with large language models. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–18, 2025.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. Gamebench: Evaluating strategic reasoning abilities of llm agents. *arXiv preprint arXiv:2406.06613*, 2024.

Melanie Cross. *Children with social, emotional and behavioural difficulties and communication problems: There is always a reason*. Jessica Kingsley Publishers, 2011.

Emily Doherty, E Margaret Perkoff, Sean von Bayern, Rui Zhang, Indrani Dey, Michal Bodzianowski, Sadhana Puntambekar, and Leanne Hirshfield. Piecing together teamwork: A responsible approach to an llm-based educational jigsaw agent. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2025.

Kanishk Gandhi, Jan-Philipp Franken, Tobias Gerstenberg, and Noah D. Goodman. Understanding social reasoning in language models with language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13518–13529, 2023.

Shuaishuai Guo, Yanhu Wang, Biqian Feng, and Chenyuan Feng. Large language model-assisted semantic communication systems. *Wireless Semantic Communications: Concepts, Principles and Challenges*, pp. 163–182, 2025.

Jessica He, Stephanie Houde, Gabriel E Gonzalez, Darío Andrés Silva Moran, Steven I Ross, Michael Muller, and Justin D Weisz. Ai and the future of collaborative work: Group ideation with an llm in a virtual canvas. In *Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work*, pp. 1–14, 2024.

Zheng Lian, Haoyu Chen, Lan Chen, Haiyang Sun, Licai Sun, Yong Ren, Zebang Cheng, Bin Liu, Rui Liu, Xiaojiang Peng, et al. Affectgpt: A new dataset, model, and benchmark for emotion understanding with multimodal large language models. *arXiv preprint arXiv:2501.16566*, 2025.

Angela S. Lin, Sudha Rao, Asli Celikyilmaz, Elnaz Nouri, Chris Brockett, Debadeepta Dey, and Bill Dolan. A recipe for creating multimodal aligned datasets for sequential tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4871–4884. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.440.

Sanya Mansoor. Claude-powered ai agent's confession after deleting a firm's entire database: 'i violated every principle i was given'. *The Guardian*, 2025. URL `https://www.theguardian.com/technology/2026/apr/29/claude-ai-deletes-firm-database`.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pp. 1–17, 2018.

OpenAI. GPT-6 Astra System Card. OpenAI Deployment Safety Hub, 2026. URL `https://deploymentsafety.openai.com/gpt-6-astra`.

Samuel J Paech. Eq-bench: An emotional intelligence benchmark for large language models. *arXiv preprint arXiv:2312.06281*, 2023.

Dim P. Papadopoulos, Enrique Mora, Nadiia Chepurko, Kuan Wei Huang, Ferda Ofli, and Antonio Torralba. Learning program representations for food images and cooking recipes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16559–16569, June 2022. doi: 10.1109/CVPR52688.2022.01606.

Dan Qiao, Chenfei Wu, Yaobo Liang, Juntao Li, and Nan Duan. Gameeval: Evaluating llms on conversational games. *arXiv preprint arXiv:2308.10032*, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. Event2mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 463–473, 2018.

Hyeongseop Rha, Jeong Hun Yeo, Yeonju Kim, and Yong Man Ro. Emotion-coherent reasoning for multimodal llms via emotional rationale verifier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2029–2037, 2026.

Sahand Sabour, Siyang Liu, Zheyuan Zhang, June Liu, Jinfeng Zhou, Alvionna Sunaryo, Tatia Lee, Rada Mihalcea, and Minlie Huang. Emobench: Evaluating the emotional intelligence of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5986–6004, 2024.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4463–4473. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1454.

Jungkyoo Shin, Jieun Han, SeungJun Kim, Yoonseon Oh, and Eunwoo Kim. Task planning for long-horizon cooking tasks based on large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13613–13619. IEEE, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Arjhun Sreedar, Rohan Pillay, and Laukik Patade. From emotion classification to emotional reasoning: Enhancing emotional intelligence in large language models. *arXiv preprint arXiv:2601.01407*, 2026.

Xiaoyu Tan, Yongxin Deng, Xihe Qiu, Weidi Xu, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. Thought-like-pro: Enhancing reasoning of large language models through self-driven prolog-based chain-of-thought. *arXiv preprint arXiv:2407.14562*, 2024.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

Yang Xiao, Jiashuo Wang, Qiancheng Xu, Changhe Song, Chunpu Xu, Yi Cheng, Wenjie Li, and Pengfei Liu. Towards dynamic theory of mind: Evaluating LLM adaptation to temporal evolution of human states. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24036–24057. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1171.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Frank F. Xu, Lei Ji, Botian Shi, Junyi Du, Graham Neubig, Yonatan Bisk, and Nan Duan. A benchmark for structured procedural knowledge extraction from cooking videos. In *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*, pp. 30–40, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlpbt-1.4.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *Advances in Neural Information Processing Systems*, 38:107309–107336, 2026.

Qihao Zhu, Leah Chong, Maria Yang, and Jianxi Luo. Reading users' minds from what they say: An investigation into llm-based empathic mental inference. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88407, pp. V006T06A018. American Society of Mechanical Engineers, 2024.

A Experiment Settings

In this section, we provide further details about the model setting we use as well as examples of specific prompts. Table 2 shows the temperature and reasoning effort used in our experiments. We

Model	Temperature	Reasoning Effort
`GPT-5` (Singh et al., 2025)	1.0	medium
`GPT-5-mini`	1.0	–
`GPT-OSS` (Agarwal et al., 2025)	1.0	–
`Qwen-3.5-9B` (Qwen Team, 2026)	1.0	medium
`DeepSeek-V4-Pro` (Xu et al., 2026)	1.0	medium

Table 2: Models and generation settings used in our experiments.

found that increasing the reasoning effort resulted in many more tokens being generated for a small performance benefit.

Additionally, Figure 8 contains examples of prompts used in our experiments.

B Data Contamination in Previous Benchmarks

```
Below are a set of transcripts describing Person A cooking various recipes,
separated by asterisks:  ***** Person A attempts to make the recipe "Summertime
Finger Salad".
Person A trim the 10 collard greens, stems removed, producing some prepared
collard greens.
Person A blends 1 cup almond butter, 1 tablespoon ginger, 14 cup lemon juice, 14
cup raw honey or 14 cup agave nectar, 2 tablespoons sesame oil, 1 pinch chili
powder, 14 cup soy sauce and 12 cup water, producing some sauce.
*****
Person A roll 2 cups carrots, 2 cups red cabbage, 2 cups cucumbers and collard
base, producing some roll.
Person A arrange the serving bowl, producing some arrangement.
Person A finishes making the recipe.
***** Person B attempts to make the recipe "Mom's Hearty Chicken and Rice
Soup"...
*****
Person C attempts to make the recipe "Tarte Flambee (Alsatian Pizza) "...
*****
Person E attempts to make the recipe "Cheesy Pizza Primavera"...
*****
All previous people share a similar emotional profile with Person Q.
Below is an incomplete transcript of Person Q making the recipe "Taco Salad":
Person Q add 1 head lettuce, 2 or 3 tomatoes, 1 onion, 1 can chili hot beans, 1
c.  Cheddar cheese, 1 1/2 c.  Catalina dressing, 1 (8 oz.)  jar taco sauce, 1
pkg.  taco seasoning, 1 lb.  hamburger and 1 pkg.  Doritos or Fritos, producing
some taco salad.
Person Q combines the taco salad, producing some taco salad mixture.
Person Q toss the taco salad mixture, producing some tossed taco salad mixture.
Person Q notices she made a mistake in step 3.
Given the following options, which option best represents how Person Q will
react?
Option 1:  fix the mistake
Option 2:  restart the recipe
Option 3:  give up
```

(a) Sample next action prediction prompt

EVENT2MIND is the closest prior work to ours. It consists of short scenarios containing a single event involving one or two characters, and models are tasked with predicting either a character's intent or emotional reaction using a fixed vocabulary of candidate phrases. An example is shown in Figure 7a.

```
Below are a list of recipes Alice may be attempting to make, followed by their
ingredients:

Recipe 1 ['1 block silken tofu', '1 lemon', '1/2 teaspoon chile flakes', '2
cloves garlic', '4 cups chopped cilantro', '1 package kombu ', '1 tablespoon
fish sauce', '5 tablespoons miso paste', '2 1/2 cups dried bonito flakes', '1/2
cup chopped scallions', '1-2 teaspoons kosher salt', '4 quarts water', '3 cups
chopped greens', '1 tablespoon sugar']

Recipe 2 ['1 tablespoon fish sauce', '5 tablespoons miso paste', '4 quarts
water', '1 package kombu ', '1 lemon', '2 1/2 cups dried bonito flakes', '3
cups chopped greens', '1/2 cup chopped scallions', '1/2 teaspoon chile flakes',
'1 tablespoon soy sauce', '4 cups chopped cilantro', '1/4 cup dried shiitake
mushrooms', '1 block silken tofu', '1-2 teaspoons kosher salt', '2 cloves
garlic']

Recipe 3 ['1/2 cup chopped scallions', '1 package kombu ', '1 block silken
tofu', '2 tablespoons grated fresh ginger', '1 tablespoon sugar', '4 quarts
water', '5 tablespoons miso paste', '1/4 cup dried shiitake mushrooms', '2
cloves garlic', '2 1/2 cups dried bonito flakes', '3 cups chopped greens', '4
cups chopped cilantro', '1 lemon', '1-2 teaspoons kosher salt', '1/2 teaspoon
chile flakes']

Recipe 4 ['4 cups chopped cilantro', '1 lemon', '2 cloves garlic', '4 quarts
water', '1 block silken tofu', '1/2 teaspoon chile flakes', '1 package kombu ',
'1/4 cup dried shiitake mushrooms', '5 tablespoons miso paste', '1/2 cup chopped
scallions', '1 tablespoon sugar', '1 tablespoon vegetable oil', '1 tablespoon
fish sauce', '1-2 teaspoons kosher salt', '2 1/2 cups dried bonito flakes']

Given the following transcript of actions performed by Alice, output only the
name of the recipe Alice is trying to cook.

Alice wipes the 1 package kombu , producing some wiped kombu.
Alice puts the 4 quarts water, producing some pot contents.
Alice drains wiped kombu and pot contents, producing some dashi.
Alice mixes in the 1/4 cup dried shiitake mushrooms.  Alice appears shocked.
```

(a) Sample mistaken ingredient test prompt

Figure 8: Example prompts from both experiments.

```
Given a list of available ingredients and a step from a cooking recipe, output a
python string list of ingredients used in the step pulled from the available
list, a simple present tense verb which best describes the action being
performed in the step (i.e.  mixes), and a name for the result of performing
the step.  Do not output any of your thought process.

GIVE YOUR OUTPUT IN THE FOLLOWING FORMAT AND OUTPUT NOTHING ELSE:
([list of used ingredients], "verb", "name")
```

Figure 9: System Prompt given to `gpt-5.4-mini` to parse a list of available ingredients (U_k) and a step from a cooking recipe (S_k) into the ([list of used ingredients], "verb", "name") format.

To assess whether this benchmark remains challenging for modern LLMs, we adapt it to more directly test E/GD reasoning. Specifically, we explicitly state the character's emotional reaction and ask the model to infer the character's underlying goal. Because the original benchmark was designed for supervised models with a fixed output vocabulary, we reformulate the task as multiple choice to accommodate unconstrained LLM generation. Each question includes the correct answer alongside distractors sampled from the answer sets of other examples, as illustrated in Figure 8a.

```
[previous list of ingredients...]

Given the previous list of ingredients used in a recipe, identify what recipe
might be being made and provide an ingredient (with measurements) that could
be added to the recipe which does not currently exist in the given ingredient
list.  Try to generate ingredients which are very different from the existing
ingredients while also belonging in the same recipe.  Do not generate an
existing ingredient with a different amount/preparation method (i.e.  if 1/2
cup diced fresh mango exists in the list, do not generate 1/2 cup grated fresh
mango, or if 1/4 cup apple cider vinegar exists in the list, do not generate 1
tablespoon apple cider vinegar).  Output only the ingredient (with measurements)
and nothing else.
```

Figure 10: System Prompt given to `gpt-5.4-mini` to generate E. Beginning with the initial list of ingredients $E = I$, e is taken from the output and appended to E which is then fed back into the prompt. Each repetition increases E by one ingredient until the desired length is reached.

```
name_list = [ 'Alice', 'Akash', 'Bob', 'Charlie', 'Dhruv', 'Eve', 'Fred',
'Gautam', 'Henry', 'Jose', 'Kevin', 'Larry', 'Myles', 'Rohan' ]

reaction_list = [ 'appears shocked', 'freezes in disbelief', 'is visibly
frustrated', 'frowns with concern', 'shakes their head slowly', ]
```

Figure 11: Actor name and reaction bank for missing ingredient tests.

We find that `gpt-5-mini` performs well on this task, achieving up to 92% accuracy. We attribute this result partly to the simplicity of the scenarios and partly to the fact that the original events were collected from text available online. By contrast, our benchmark uses procedurally generated scenarios with controlled actors and behaviors, reducing the likelihood that test examples appear verbatim in pretraining data and enabling more challenging evaluations of whether models use emotional information to infer goals.