

THE DECRYPTO BENCHMARK FOR MULTI-AGENT REASONING AND THEORY OF MIND

Anonymous authors

Paper under double-blind review

ABSTRACT

As Large Language Models (LLMs) gain agentic abilities, they will have to navigate complex multi-agent scenarios, interacting with human users and other agents in cooperative and competitive settings. This will require new reasoning skills, a crucial one being *theory of mind* (ToM), or the ability to reason about the “mental” states of other agents. However, ToM and other multi-agent abilities in LLMs are poorly understood, since existing benchmarks suffer from narrow scope, data leakage, saturation, and lack of interactivity. We thus propose DECRYPTO, a game-based benchmark for multi-agent reasoning and ToM drawing inspiration from cognitive science, computational pragmatics and multi-agent reinforcement learning. It is designed to be as *easy* as possible in all other dimensions, eliminating confounding factors commonly found in other benchmarks. To our knowledge, it is also the first platform for designing interactive ToM experiments.

We validate the benchmark design through comprehensive empirical evaluations of frontier LLMs, robustness studies, and human-AI cross-play experiments. We find that LLM game-playing abilities lag behind humans and simple word-embedding baselines. We then create variants of two classic cognitive science experiments within DECRYPTO to evaluate three key ToM abilities. Surprisingly, we find that state-of-the-art reasoning models are significantly *worse* at those tasks than their older counterparts. This demonstrates that DECRYPTO addresses a crucial gap in current reasoning and ToM evaluations, and paves the path towards better artificial agents. Code at anonymous.4open.science/r/decrypto/.

1 INTRODUCTION

Reasoning abilities and agentic behaviour are currently at the forefront of large language model (LLM) research (Huang et al., 2023; Jaech et al., 2024; Guo et al., 2025), with early applications of LLM agents including software engineering (Jimenez et al., 2023) and web navigation (Yao et al., 2023). While those tasks are predominantly single-agent, real-world agentic applications will increasingly involve complex interactions with many other actors, including business partners, teammates, users and other agents. Thus, the future of AI problems is irreducibly *multi-step*, *multi-agent*, *partially observable*, and *stochastic* – a reality not captured by the existing suite of benchmarks.

To navigate these multi-agent scenarios, LLMs will require *theory of mind* (ToM), the general ability of modelling the mental states of other agents (artificial or biological), including their knowledge, beliefs, intentions and abilities. However, LLM ToM remains poorly understood. Despite ToM being inherently tied to multi-agent *interactions*, the majority of existing benchmarks lack interactivity and are restricted to variants of the Sally-Anne task (Baron-Cohen et al., 1985). Furthermore, Hu et al. argue that many of those benchmarks suffer from confounding factors and bias introduced when translating embodied scenarios to text.

To address the gap in ToM evaluations, we introduce DECRYPTO, a multi-agent reasoning and ToM benchmark based on the award-winning language game of the same name. DECRYPTO is a pragmatic inference game (Degen, 2023), where two agents must exchange secret messages while preventing a third party from intercepting them.

We designed DECRYPTO to be future-proof and as *easy* as possible for LLMs in all other dimensions except multi-agent reasoning. Indeed, it is operated entirely through word associations, eliminating common confounding factors of LLM performance, such as tokenisation, long contexts or embodied scenarios. The benchmark enables the study of cooperation, competition and ToM within a single game, and requires players to reason about the information available to others before making strate-

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

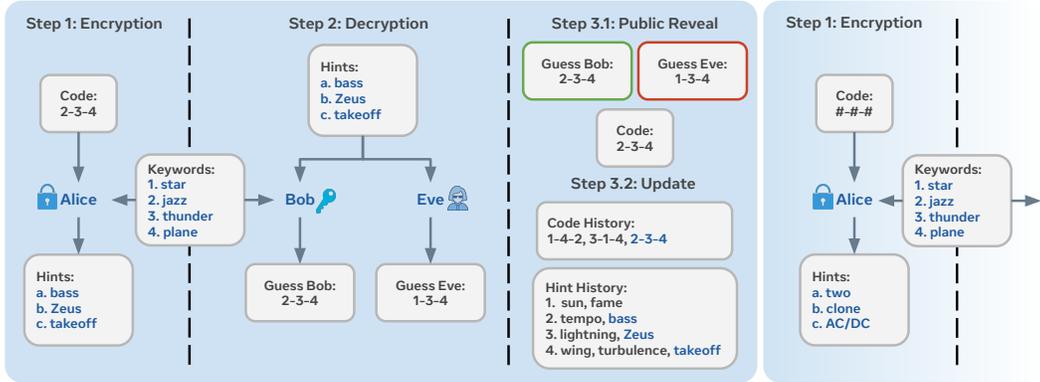


Figure 1: Left: Overview of a turn in DECRYPTO, split into three steps, with Alice and Bob (Encoder and Decoder) playing against Eve (Interceptor). Step 1: Alice draws a code of 3 non-repeating digits, and provides 3 hints referring to the meaning of the four *Keywords*. Step 2: Bob and Eve receive the hints and attempt to guess the code independently. Step 3: Both guesses and the code are publicly revealed, and the code and hint histories are updated. All players have access to the histories at all times, but only Alice and Bob have access to the keywords. Alice aims to give hints that are easy for Bob to decode but hard for Eve. As the game progresses, the growing hint history makes it easier for Eve to intercept. Right: Encryption phase of a new turn. **We invite the reader to guess the code as Bob and then copy this for the answer:** []. We provide a rationale in Section C.

gic decisions. Since it is based on a popular boardgame, our benchmark produces easily interpretable insights and enables the study of human-AI interactions in strategic communication tasks. Finally, to the best of our knowledge, DECRYPTO is the first platform for studying LLM ToM in interactive settings. Built to be easily extended, it enables researchers to create new ToM experiments by writing only prompts and a few lines of code. We demonstrate the versatility of the DECRYPTO ToM platform by creating variants of two classic experiments from cognitive psychology.

2 DECRYPTO: THE GAME

DECRYPTO is a word-based code-guessing board game published by Scorpion Masqué¹, of which we implement a variant requiring three players divided into two teams. Borrowing the terminology from cryptography, players are assigned three distinct roles: Alice (the Encoder), Bob (the Decoder), and Eve (the Interceptor or Eavesdropper). A full game (i.e., an episode) lasts at most 8 turns. We show one such turn in Figure 1.

Alice and Bob are on the same team, and share 4 ordered secret *keywords*, sampled without replacement before the start of the episode and kept fixed throughout. In Figure 1, those are [star, jazz, thunder, plane].

Each turn, Alice samples a secret *code* of 3 non-repeating digits between 1 and 4, such as [2-3-4], with any of the 24 possible codes occurring at most once per episode. Alice must then provide 3 public *hints* (e.g. [bass, Zeus, takeoff]), such that Bob can guess the code but Eve cannot.

Next, Bob and Eve each make an independent attempt to *guess* the code. If Bob guesses incorrectly, his team gets a *Miscommunication* token. If Eve guesses correctly, she gets an *Intercept* token.

Finally, Bob and Eve’s guesses, the real code, and any event (miscommunication or intercept) that occurred during the turn are publicly revealed. Thus, the history of all codes from previous turns and the history of past hints associated with each of the four keywords are common knowledge, but the keywords themselves are not. The *code history* and especially the *hint history* are the two main pieces of information that will help Eve make educated guesses in future turns. For instance, if the hint history for digit 3 so far is [lightning, Zeus], Eve may associate terms related to storms or electricity with that digit, even without knowing the exact keyword.

The game ends and Eve wins immediately if Alice and Bob accumulate two Miscommunication tokens or if Eve gains two Intercept tokens. Alice and Bob win if they make it through 8 rounds

¹www.scorpionmasque.com/en/decrypto

without either of those two conditions happening. Alice must therefore be careful to provide subtle hints to avoid interception, yet make them clear enough for Bob to guess correctly.

The keywords are sampled from a predetermined set, but *the choice of hints is open-ended* and can be composed of multiple words. The restriction is that hints must be real words (including proper nouns) and refer to the meaning of the keywords, not to their spelling or pronunciation. They must also rely on publicly available information, which can include culture, science or recent events.

Beyond word associations. At the surface level, DECRYPTO provides a language reasoning challenge that consists in matching hints to either keywords or the hint history. However, Alice’s hints cannot be too literal, or they will get intercepted, and so DECRYPTO can be formalised as a pragmatic inference game under the Rational Speech Act (RSA) framework (Goodman & Frank, 2016; Degen, 2023), where the listeners (Bob and Eve) update their belief of the intended meaning of the speaker (Alice) via Bayesian inference. We provide such a treatment in Section H, explicitly showing that agents must model each other’s beliefs and perform second-order ToM for optimal play.

To help readers intuit the role of ToM in Decrypto, we invite them to attempt guessing the code on the right side of Figure 1 as Bob, and to read the rationale in Section C. Alternatively, readers can download our codebase and play a game of Decrypto with an LLM in only a few minutes.

3 DECRYPTO: THE BENCHMARK

An easy benchmark. Unlike other reasoning benchmarks, DECRYPTO is purposely designed to *not* require many of the capabilities that large pre-trained models struggle with. In particular, it demands no symbolic (Bard et al., 2020), mathematical (Cobbe et al., 2021; Zhang et al., 2024a) or spatial reasoning (Clark et al., 2018; Carroll et al., 2019), no tool use (Xu et al., 2023a), and no particular attention to tokenisation, which has been shown to affect arithmetic performance (Singh & Strouse, 2024). It also distinguishes itself from recent ToM benchmarks (Le et al., 2019; He et al., 2023; Kim et al., 2023; Zhou et al., 2023; Xu et al., 2024a; Gandhi et al., 2024; Sclar et al., 2024) by not relying on state tracking (e.g. of object locations) or on textual representations of embodied settings, which Hu et al. argue introduce unwanted pragmatic artifacts and bias LLM answers.

Instead, strong performance in DECRYPTO relies on language-based reasoning and theory of mind. Since LLMs are trained to learn word co-occurrences and follow instructions, we expect DECRYPTO to be easy to play by LLMs, but hard to master. Our results and Section L show this to be the case: even Llama 3.1-8B understands the rules, but much stronger models still struggle to play well.

A future-proof benchmark. With its two-team, three-player dynamics, DECRYPTO is a benchmark for both competitive and cooperative scenarios.

The multi-agent and open-ended nature of the game means that difficulty scales with the choice of agents for each role, resulting in an evergreen challenge that is much harder to saturate than a fixed dataset of problems. We curate a set of 680 possible keywords, resulting in over 8.8 billion possible keyword combinations to limit memorisation, and also implement the option to provide custom keywords beyond this standard set. Combined with the option to provide arbitrary hints, including ones that refer to recent events or niche cultural works, the number of possible games is virtually infinite. Furthermore, DECRYPTO was originally designed for human play, facilitating studies on human-AI coordination and comparisons in addition to inter-AI benchmarking.

Competition. In this setting, we compare two models A and B by having both Alice and Bob be played by A , and Eve by B . At its simplest, we can record the win rate of A vs. B , swapping teams to ensure a fair evaluation. However, we propose more granular metrics below.

Ad-hoc Coordination. Here, we fix Eve to a give model and instantiate Alice and Bob to be played by different models A and B . Like in Stone et al. (2010), we are concerned with the ability to “*efficiently and robustly collaborate with previously unknown teammates*”, such as independently trained LLMs. As highlighted in FAIR et al. (2022), this requires approaches beyond simple self-play, since the goal is to adopt strategies that *generalize* to team pairings where $A \neq B$. A crucial subset of ad-hoc coordination is **human-AI coordination**, where one of the two agents (Alice or Bob) is played by a human. This setting paves the way towards more social AI agents that seamlessly coordinate with humans and understand their intents.

Metrics. Both settings are subject to the same tension that is at the core of DECRYPTO: Alice must provide hints that balance what she knows about Bob, Eve, and the information available to each of them. If the hints are too obscure, Bob will guess wrong, which leads to a *miscommunication*; too obvious, and Eve will *intercept*; just right, and Alice and Bob survive for another round. The number of miscommunications and of intercepts are therefore two sides of the same coin, providing granular break-down of the failure modes of LLMs. Meanwhile, the average *number of turns per episode* captures both sides in one metric, since longer games mean that Alice and Bob could better balance the difficulty of hints to avoid defeat. Game length is also more informative than win-rate, since we empirically find current LLMs to be much weaker at providing hints than intercepting, which results in Eve having significantly higher win-rates in most match-ups (see Figure 7).

3.1 THEORY OF MIND

As explained in Section 2, strong performance in DECRYPTO requires players to explicitly reason about each other’s beliefs and world knowledge. Indeed, a model could gain significant advantage over its opponent by identifying and exploiting their knowledge gaps, either due to human limitations or by referring to information made public after the knowledge cut-off of an older model.

Beyond game playing, DECRYPTO provides a platform for conducting interactive ToM experiments inspired by cognitive psychology. We conduct two such experiments, evaluating three different ToM abilities, and gaining diagnostic insight into the poor performance of LLMs in DECRYPTO. Thanks to a flexible codebase design, each required only new prompts and approximately 15 additional lines of code. We describe these ToM experiments and their results in Section 5.1.

4 SPECIALIST VS GENERALIST AGENTS

Given the plurality of settings in DECRYPTO, we caution users to be explicit and transparent about their assumptions when declaring state-of-the-art. When reporting results, we recommend distinguishing between two classes of agents: **generalists** and **specialists**.

Generalist agents refer to general-purpose models used to play the game “out-of-the-box”, without any additional fine-tuning or in-context learning outside of that happening within individual games. In particular, for generalist agents, we *consider the prompt to be the observation* returned by the environment, which precludes any form of prompt engineering. Evaluating foundation models in this framework assesses their ability to zero-shot generalise to novel tasks, including out-of-distribution ones and tasks where training data is limited or unavailable.

Because foundation models are often sensitive to prompting, we recommend validating generalist agents with a range of different prompts to prevent the choice of prompt favouring any one model. We provide such an evaluation using 625 prompt variants in Section G, showing that game performance is robust to prompt variations and depends predominantly on general model capabilities.

In contrast, **specialist agents** are all the agents designed specifically for the task. Solutions in this space include rule-based strategies, training on the game using RL, building a prompt pipeline informed by the rules of the games, and prompt engineering. Here, *the environment only returns key information*, and prompt engineering is the designer’s burden, if it is needed at all.

Specialist agents can, in theory, overfit to DECRYPTO, which makes some degree of performance easier to reach, but does *not* automatically solve the game under every setting. For instance, we know from the cooperative multi-agent RL literature that training models to collaborate in cross-play (XP) with unseen agents can be extremely challenging, even with unlimited compute and access to the environment (Hu et al., 2021; Cui et al., 2023; FAIR et al., 2022). We expect this to also hold for DECRYPTO. This is evidenced by the rule-based baselines we describe next, which coordinate flawlessly in DECRYPTO if Alice and Bob pre-agree on a common strategy and a word embeddings, but otherwise can fail catastrophically when paired with other agents, including humans and LLMs.

Specialist Agents: Baselines. We introduce two specialist agent baselines to play Decrypto, built around word embedding models. Those are the common crawl GloVe with a vocabulary of 2.2 million words (Pennington et al., 2014) and the Word2Vec embeddings trained on the Google News dataset, provided by the gensim library (Rehurek & Sojka, 2011; Mikolov, 2013).

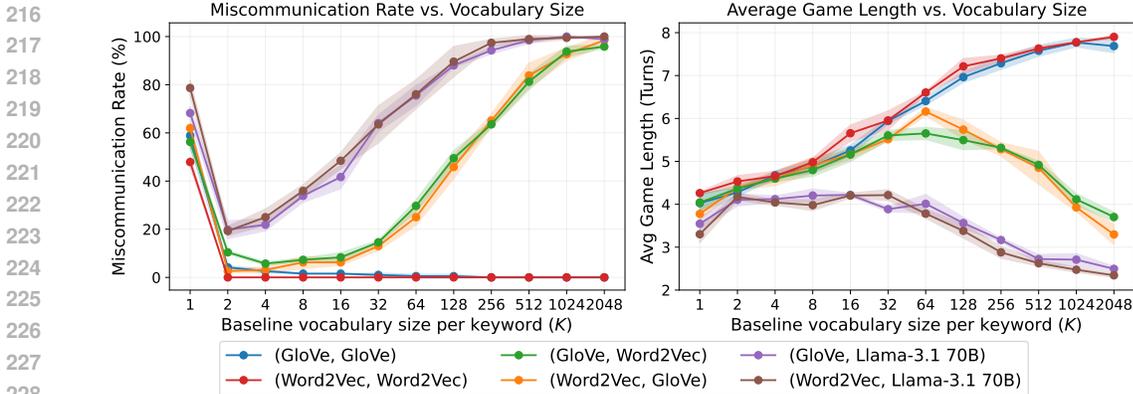


Figure 2: Percentage of games ending in miscommunications (left) and average game length (right) for the word embedding baselines as a function of K – the hint vocabulary size per keyword. Each curve corresponds to an encoder-decoder pair. For low K , the baselines coordinate well, but their hints are easier to intercept. For high K , both models sample from lower similarity words. This results in very strong self-play baselines, which cannot coordinate in cross-play, with most games ending in miscommunication for $K > 128$. While the interceptor is Llama 3.1-70B, trends hold across all interceptors, including baselines. We report mean and standard error over 3 model seeds.

For both models, the baseline algorithm is the same and relies on a fixed corpus of 5696 nouns. When hinting a digit, Alice draws from the set of the top- K most similar words to the corresponding keyword under cosine similarity, filtering out hints that are more similar to another keyword than to the one intended. In turn, Bob greedily assigns each hint to the most similar keyword, measured using his own embedding model and cosine similarity. Not having access to the keywords, Eve computes the mean embedding of the hint history for each keyword, and then finds the assignment between hints and mean embedding that maximises the sum of cosine similarities between each hint and the history of the digit it is assigned to. We provide additional algorithmic details in Section D.

The purpose of these baselines is threefold. First, we can study the case where Alice and Bob share perfect ToM, represented by a shared strategy and word embedding. This establishes an upper bound on performance to which to compare other agents. Second, having baselines using two different word embeddings, we demonstrate in Section 5 how unaccounted differences in how agents associate words can lead to a catastrophic lack of coordination and poor XP performance. Third, K determines the amount of diversity in Alice’s hints. By tweaking it, we can operate in a regime where GloVe and Word2Vec correlate and represent reasonable rule-based baselines against which we can compare LLMs. We thus use $K = 16$ for the rest of the paper, as it is the value in Figure 2 beyond which the baseline-LLM miscommunication rate increases significantly.

Being explicit about whether DECRYPTO is approached from a *generalist* or *specialist* perspective provides clarity and avoids unfair comparisons between models making inherently different assumptions. It also leads to two equally rich but distinct research directions. In the first case, DECRYPTO acts as a benchmark for multi-agent reasoning and measures progress in foundation models. In the second case, it is an RL environment used to research and evaluate training algorithms.

5 RESULTS

To kickstart this benchmark, we assess the cooperative, competitive, and theory of mind capabilities of specialist and generalist agents in Decrypto. Amongst generalist open-source models, we evaluate Llama-3.1-8B-Instruct, Llama-3.1-70B-Instruct (Dubey et al., 2024), and DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025). From the closed-source models, we evaluate GPT-4o, o1 and o3 (with reasoning set to “high”), as well as Claude 3.7 Sonnet with “Extended Thinking” for the human and ToM experiments. We also include the two baselines from Section 4 in all game playing experiments. We report mean \pm standard error over three seeds. Additional details are in Section F.

Baselines. We first look at the specialist word embedding baselines and show in Figure 2 that we can control K for the top- K selection to make them arbitrarily strong. Smaller K prioritises words semantically similar to the keyword, making the hints easier to guess. Bigger K selects

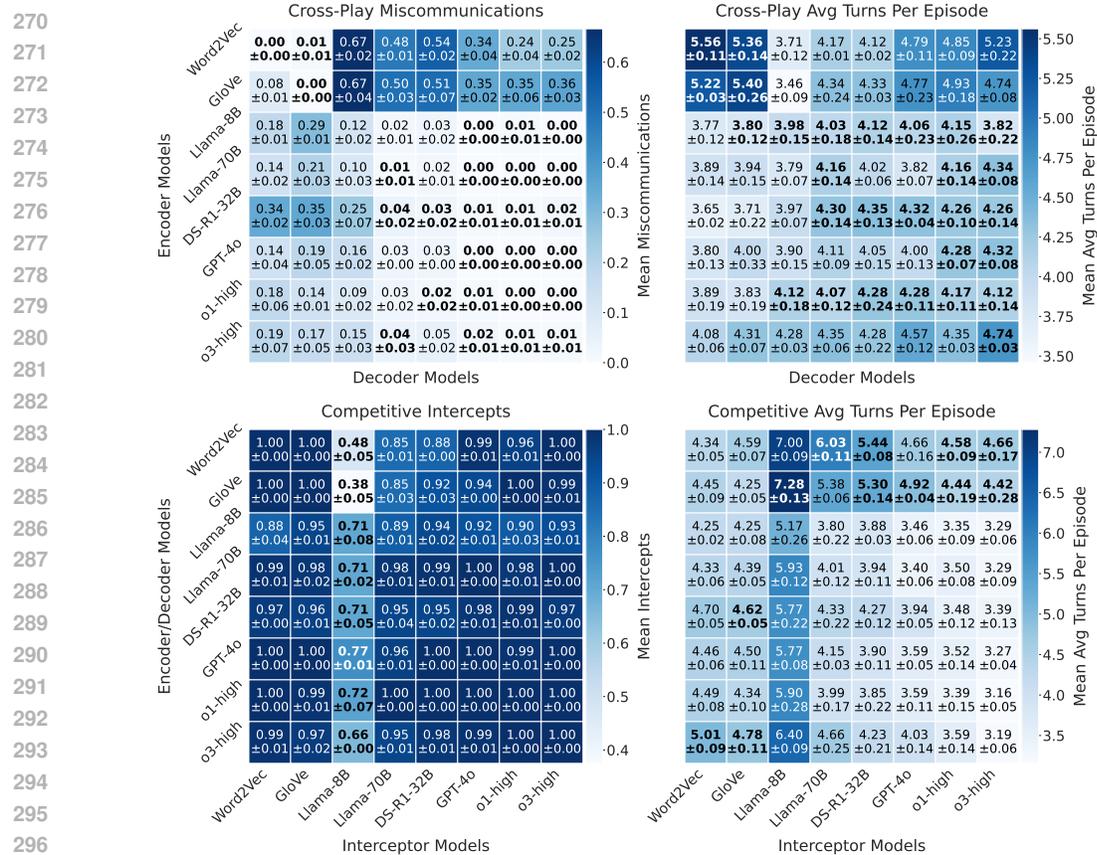


Figure 3: Cooperative and competitive results for 8 different agents. **Top:** Cooperative cross-play matrix when the interceptor is Llama 3.1-70B-Instruct. *Left:* proportion of games ending in miscommunication. *Right:* average number of games ending in miscommunication. **Bottom:** Competitive results playing a homogeneous encoder-decoder team against an interceptor. *Left:* proportion of games ending in an intercept. *Right:* average number of turns per episode. LLMs excel at intercepting other LLMs, but have short games and extremely low win rates as encoder-decoder, which we attribute to a failure of ToM.

semantically less similar hints, which are harder to guess if Bob cannot access the same embeddings. For example, suppose GloVe-Alice wants to hint the keyword “fire” to Word2Vec-Bob, and $K = 50$. The 50th most similar word to “fire” from the hint corpus would be “oil” for GloVe. However, “oil” is not even in the top 1000 for Word2Vec-Bob, highlighting how different word associations can produce cross-play difficulties. This finding also holds when an LLM plays Bob: as K increases, miscommunications increase, and the average turn length decreases.

Coordination. The top row of Figure 3 shows the proportion of games ending in miscommunication and the average game length for different encoder-decoder pairs. Teams composed of an LLM and a baseline get significantly more miscommunications than baseline-only or LLM-only teams, evidence of different word associations between those two classes of models. LLM performance, measured by turns per episode, is positively correlated with model size or recency, but all LLM-only teams are outperformed by those where Alice is a baseline, highlighting the importance of hinting.

Competition. Looking at competitive game lengths in the bottom right of Figure 3, we see that LLMs are better than baselines at intercepting other LLMs, but are outmatched when intercepting baselines or when playing as the encoder-decoder team. The bottom left matrix shows that the vast majority of games end in interceptions, with win rates heavily skewed in Eve’s favour. Even o3-high, the strongest LLM tested, achieves only $\sim 34\%$ win rate against the weakest interceptor, Llama-3.1 8B. While such a heavy bias may appear to be a property of the game, results in Section 5.1 suggest

	Model as Interceptor	Miscomms	Intercepts(↑)	Win Rate(↓)	Game Length(↓)
324	Word2Vec	11.00 ± 0.00	8.00 ± 0.00	40.00% ± 0.00%	7.00 ± 0.00
325	GloVe	9.00 ± 0.00	9.00 ± 0.00	20.00% ± 0.00%	6.50 ± 0.00
326	Llama 3.1-8B-Instruct	11.67 ± 0.33	6.33 ± 0.88	46.67% ± 3.33%	6.97 ± 0.13
327	ExploreToM-8B	11.67 ± 0.33	4.00 ± 2.31	43.33% ± 6.67%	6.90 ± 0.20
328	Llama 3.1-70B-Instruct	11.33 ± 0.33	8.33 ± 1.76	46.67% ± 3.33%	6.80 ± 0.21
329	DeepSeek-R1-Dist-32B	9.33 ± 0.33	10.67 ± 0.88	30.00% ± 5.77%	6.33 ± 0.03
330	GPT-4o	10.67 ± 0.88	10.00 ± 0.58	36.67% ± 3.33%	6.50 ± 0.15
331	o1-high	9.33 ± 0.33	15.67 ± 0.88	20.00% ± 5.77%	6.23 ± 0.12
332	o3-high	6.67 ± 0.33	19.33 ± 0.33	6.67% ± 3.33%	5.50 ± 0.15
333	Claude 3.7 (+ ext. thinking)	8.33 ± 0.33	14.00 ± 0.58	20.00% ± 0.00%	6.30 ± 0.12
	Model as Decoder	Miscomms(↓)	Intercepts	Win Rate(↑)	Game Length(↑)
334	Word2Vec	20.00 ± 0.00	3.00 ± 0.00	0.00% ± 0.00%	2.70 ± 0.00
335	GloVe	20.00 ± 0.00	5.00 ± 0.00	0.00% ± 0.00%	3.10 ± 0.00
336	Llama 3.1-8B-Instruct	20.00 ± 0.00	4.33 ± 0.88	0.00% ± 0.00%	3.00 ± 0.15
337	ExploreToM-8B	20.00 ± 0.00	3.00 ± 0.00	0.00% ± 0.00%	2.53 ± 0.09
338	Llama 3.1-70B-Instruct	18.67 ± 0.88	7.67 ± 0.33	3.33% ± 3.33%	4.70 ± 0.12
339	DeepSeek-R1-Dist-32B	19.67 ± 0.33	6.67 ± 1.20	0.00% ± 0.00%	4.43 ± 0.22
340	GPT-4o	17.33 ± 0.88	8.67 ± 0.67	3.33% ± 3.33%	5.23 ± 0.12
341	o1-high	14.33 ± 0.67	8.00 ± 1.00	26.67% ± 3.33%	5.77 ± 0.15
342	o3-high	10.67 ± 1.20	9.67 ± 0.33	20.00% ± 5.77%	6.30 ± 0.12
343	Claude 3.7 (+ ext. thinking)	12.67 ± 0.67	11.00 ± 0.58	16.67% ± 6.67%	6.57 ± 0.12
	Human (original games)	11	12	40.00%	6.90

Table 1: We collect 10 games from human encoder-decoder teams and report the agents’ performance when playing as (top) interceptor against human players or as (bottom) decoder with a human encoder. Reasoning models have a clear advantage, with o3-high standing out as a particularly strong interceptor. In ad-hoc human coordination, all agents perform worse than the human decoder, getting significantly more miscommunications, a lower win rate and shorter games. However, reasoning models again show stronger relative performance. ExploreToM-8B, a Llama 3.1-8B variant fine-tuned by Sclar et al. on ToM data, underperforms the latter in both settings, supporting the claim that DECRYPTO is complementary to other ToM benchmarks.

otherwise: that all LLMs tested make no attempt to consider Eve’s perspective when hinting, instead choosing hints which will predictably get intercepted.

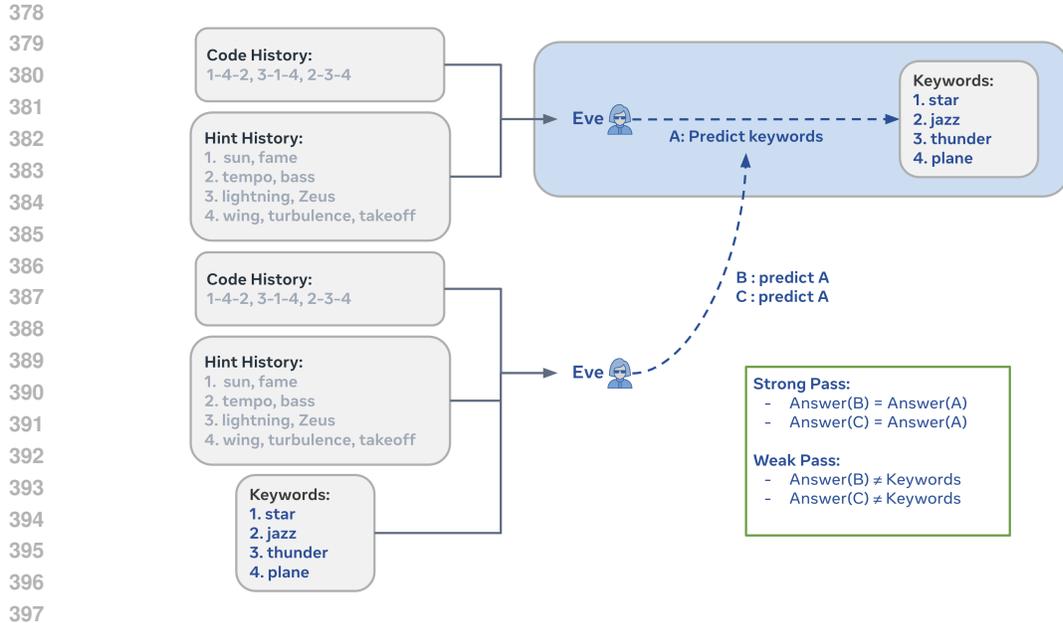
Human Evaluation. We collect 10 full games of human teams playing against a fixed LLM Eve, namely Llama 3.1-70B-Instruct. The players interacted with the game through a command-line interface and saw the same system prompts, user prompts and information as LLMs did. See Sections I and J for prompts and Section K for data collection details and the code repository for the data itself.

Table 1 shows the result of replaying collected human games by re-using the human-provided hints and replacing either Eve or Bob with different LLMs. Closed-source reasoning take the lead in both roles, with o3-high standing out as a particularly strong interceptor. As Bob, recent models perform best, coming close but not yet matching human decoders. The o-series models were also particularly verbose: it is the only experiment in which they occasionally exceeded 10k tokens per response. We also evaluate ExploreToM-8B (Sclar et al., 2024), a variant of Llama 3.1-8B fine-tuned on ToM tasks, which significantly outperforms the latter on multiple ToM benchmarks. Despite that, we found it to perform slightly worse in our DECRYPTO experiments, which **demonstrates that our benchmark is complementary to existing Sally-Anne type tasks.**

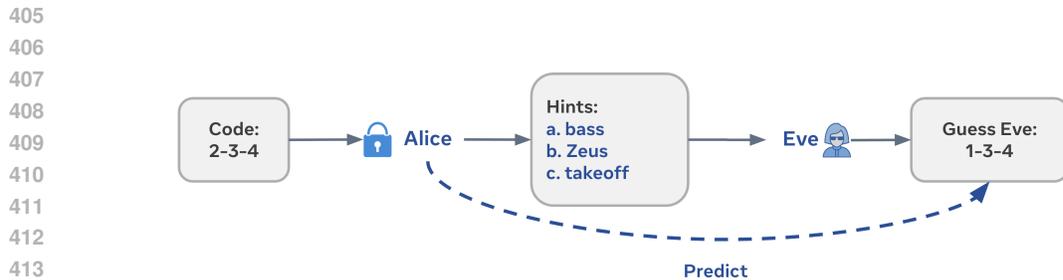
5.1 THEORY OF MIND

Drawing upon seminal works in cognitive psychology, we adapt classic ToM experiments to expose the limitations of LLMs and explain their poor performance in DECRYPTO.

The first experiment adapts the Smarties Task of Gopnik & Astington, which presents children with a deceptive object (a box of Smarties containing pencils) and studies whether the child can correctly identify incorrect beliefs – either their own or of another child – when first encountering the object. To recreate this task in DECRYPTO, we substitute the closed Smarties box and the pencils with the game history and the secret keywords. At each turn except the first, we prompt Eve three times independently. Prompt *A* asks her to predict the four keywords. Prompt *B* reveals the keywords and



398 Figure 4: Diagram of the ToM experiment evaluating Representational Change and False Belief.
399 Eve is prompted three times. A) She is prompted to predict the keywords given the code and hint
400 histories. B) Eve is revealed the keywords and is asked what she thought were the four keywords
401 pre-reveal. C) Eve is also revealed the keywords and is asked what a second interceptor would think
402 are the keywords pre-reveal. For B or C, we consider it a strong pass if the answer matches the
403 answer to A. We consider it a weak pass if the answer is *not* the keywords. Because we only evaluate
404 on turns where Answer(A) \neq Keywords, a strong pass always implies a weak pass as well.



416 Figure 5: Diagram of the ToM experiment evaluating perspective taking. After Alice has given her
417 hints, we prompt her to predict Eve’s guess. This is done in branching context, and therefore does
418 not change Alice’s behaviour in the game.

421 asks Eve what she thought were the keywords pre-reveal. Prompt C also reveals the keywords but
422 asks the model to predict what a “second interceptor” would think the keywords to be, pre-reveal.
423 We generate outputs with temperature 0 (for models that allow it) and only consider turns where
424 the answer to A is an incorrect guess, keeping only cases where Eve has inaccurate “beliefs” pre-
425 reveal. We compare answers A and B to measure *representational change* (RC), the ability of the
426 agent to recognise when its belief about the world (but not the world itself) changes due to additional
427 information. Similarly, comparing A and C measures *false belief* (FB), the ability to represent other
428 agents as having false beliefs about the world. We distinguish two variants of the tasks. The *Weak*
429 variant only requires the agents to realise that either themselves or the second interceptor could not
430 have known the ground truth, and so an answer to either B or C is correct as long as it differs from
431 the real keywords. For the *Strong* variant of those tasks, we consider the agent to pass only if it
correctly predicts its answer to prompt A (i.e. if $B = A$ or $C = A$). Success here likely requires a

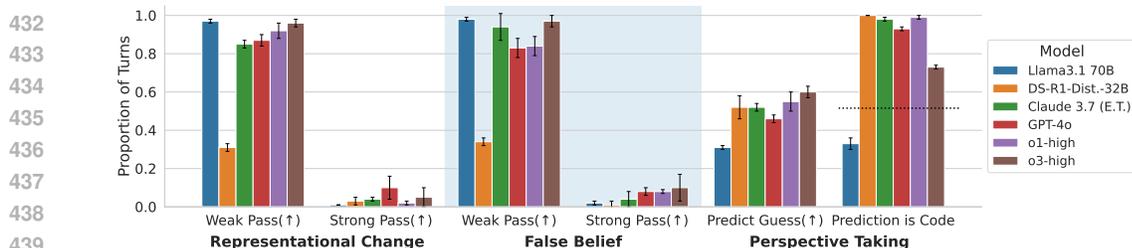


Figure 6: *Representational change* (RC), *false belief* (FB) and *perspective taking* (PT) scores in DECRYPTO. RC measures whether agents can identify a change in their own world-view following new information. FB measures whether agents can model the incorrect beliefs of another agent. PT assesses whether LLMs can reason from the point of view of an agent with less information. Most models perform well on the weak RC and FB tasks, but fail on the strong variants, demonstrating a lack of self-consistency or counterfactual reasoning. In PT, all models except Llama 3.1-70B and o3-high erroneously predict that Eve will intercept on nearly every turn, including the first. Models also fail to consider Eve’s point of view by choosing hints they subsequently predict will get intercepted.

self-consistent representation of the keywords, or at least strong counterfactual reasoning. Figure 4 summarises the procedure.

The first two thirds of Figure 6 evaluates the RC and FB abilities of six different LLMs. Most models tested achieve high accuracy in both weak RC and weak FB, with Llama 3.1-70B slightly outperforming newer models, including reasoning ones. That said, models struggle to correctly identify which belief they or another agent held prior to receiving new information, as evidenced by the $\leq 10\%$ accuracy on the strong variants.

The second experiment measures *perspective taking* (PT) and is a word-based code-guessing variant of the Three Mountain Problem (Piaget et al., 1956). After Alice has given her hints, we prompt her to predict Eve’s guess, as shown in Figure 5. We record the prediction accuracy and the proportion of turns on which Alice predicts that Eve will guess the correct code. This last metric is particularly insightful as it explains why LLMs struggle to avoid getting intercepted in Section 5. On the one side, if Alice predicts that Eve will intercept more rarely than she does means **Alice underestimates Eve’s abilities or fails to account for all the information available to her**, which in turn can lead Alice to give hints that are too obvious. This is the case for Llama-3.1 70B, on the right of Figure 6. It trails behind other models in prediction accuracy, and is also the only model to predict fewer interceptions than ground truth (the real interception rate is $\sim 52\%$, indicated by the dotted line).

On the other side, if Alice predicts that Eve will intercept on nearly every turn, it is a **failure to reason from Eve’s perspective**. It would indicate that LLMs confuse their knowledge with that of others, just like young children in Piaget et al.’s study, by either explicitly or implicitly assuming that Eve has access to the keywords. It also shows a **second failure: that of integrating ToM reasoning in decision-making**. Indeed, if Alice thinks Eve will guess the code given her hints, she should have chosen different hints in the first place.

Such a failure case is precisely what we observe with DeepSeek-R1-Distill-Qwen-32B, Claude 3.7 Sonnet, GPT-4o and o1-high. All those models predict that Eve will intercept on nearly every turn, *including the first*, as evidenced by model outputs in Section M. Only Llama and o3-high correctly point out that Eve can do no better than a random guess, albeit o3-high still overestimates the rate of interceptions. Surprisingly, Listing 16 shows these results to hold even if we modify the PT prompt to emphasize that Eve “*does *NOT* know the secret keywords*”, with o1-high showing no statistical difference in predictions with the modified vs. the original prompt. This provides a diagnostic for why interceptions in DECRYPTO are so common: **LLMs do not attempt to model other agents before making decisions, and also struggle to model them if explicitly asked to**.

6 RELATED WORK

BattleAgentBench (Wang et al., 2024), multi-robot studies (Li et al., 2023) and other works such as Liu et al. (2023) and Guo et al. (2024) have recently studied LLMs in multi-agent contexts. While detailed, they focus on spatial reasoning rather than on the language-based reasoning and ToM abilities central to DECRYPTO. Among multi-agent environments, social deduction games

486 such as Werewolf (Xu et al., 2023b;c), Avalon (Light et al., 2023), [Chameleon and Undercover \(Xu](#)
487 [et al., 2024b\)](#) are particularly amenable to studying LLMs, as are negotiation games (Abdelnabi
488 et al., 2023). However, they provide less structured and atomic insight than DECRYPTO, given the
489 free-form dialogue and frequent deception in those settings. AgentPro (Zhang et al., 2024b) studies
490 competitive scenarios in Blackjack and Poker, but lacks the cooperative aspect of DECRYPTO.

491 The study of artificial ToM started with ToM-net (Rabinowitz et al., 2018), a simple network capable
492 of predicting the behaviour of other artificial agents and acting based on false beliefs. In contrast,
493 recent works on ToM have focused on LLMs modelling human beliefs (Chen et al., 2024; Sclar et al.,
494 2024; Kim et al., 2023). DECRYPTO connects both by enabling the study of LLM ToM applied to
495 both humans and other artificial agents. Unlike other recent ToM benchmarks (Le et al., 2019; He
496 et al., 2023; Kim et al., 2023; Zhou et al., 2023; Xu et al., 2024a; Gandhi et al., 2024; Sclar et al.,
497 2024) and Rabinowitz et al. (2018), however, DECRYPTO does so in an interactive setting, where
498 agents can not only model the beliefs of other agents, but also influence them.

499 Kosinski (2023) also studied the Smarties Task, from which they concluded that “ToM-like ability
500 [...] may have spontaneously emerged” in LLMs. Those claims were directly challenged by Ull-
501 man in subsequent work, with which our results agree. Shaikh et al. (2023) use another language
502 game, Codenames, to study cross-cultural pragmatics. Their work stops short of evaluating LLMs,
503 but raises important questions about the role of cultural backgrounds in pragmatic inference. DE-
504 CRYPTO is a natural choice for pursuing research in this direction, since we can specify different
505 personas in the LLM prompts.

506 7 CONCLUSION

507 Our paper introduces DECRYPTO, an interactive multi-agent benchmark designed to evaluate coordi-
508 nation, competition and theory of mind capabilities in language models. DECRYPTO addresses
509 critical limitations of existing ToM benchmarks, such as biases arising from textual translation of
510 embodied scenarios or lack of interactivity. Designed to be future-proof and to eliminate confound-
511 ing factors known to limit LLM performance, DECRYPTO fills an important gap in existing bench-
512 marks. Furthermore, our codebase provides a versatile platform for quickly designing interactive
513 ToM experiments inspired by cognitive psychology.

514 We conduct extensive experiments to evaluate open-source and closed-source LLMs. We find that
515 even state-of-the-art models struggle with the nuanced communication and strategic reasoning that
516 DECRYPTO requires, often underperforming simple baselines in cooperative and competitive set-
517 tings alike. Similarly, our human-AI experiments shed light on the limited ability of recent LLMs to
518 coordinate with humans or understand their communications.

519 Finally, we create variants of two seminal experiments from cognitive psychology to study LLM
520 ToM in interactive settings, analogously to how it is studied in children. Our experiments provide
521 strong evidence that state-of-the-art models still lack many ToM skills. In particular, we identify
522 their inability to reason from the perspective of other agents as the main cause behind their poor
523 performance in DECRYPTO and predict that this same inability will also limit their deployment in
524 other multi-agent scenarios with partial information.

525 DECRYPTO opens up several promising research directions. On the ToM side, these include en-
526 hancing the ToM capabilities of language models and investigating differences between how LLMs
527 model humans versus how they model other agents. From the multi-agent perspective, an exciting
528 avenue for future work is adapting algorithms from multi-agent RL (Silver et al., 2017; Hu et al.,
529 2020) to fine-tune LLMs. For a longer discussion on the limitations of our work and on the multiple
530 research directions enabled by DECRYPTO, we refer the reader to Section A.

531 REPRODUCIBILITY STATEMENT

532 We have taken several steps to enable reproduction of our results. Our source code and collected
533 human data are available at anonymous.4open.science/r/decrypto/. This includes de-
534 tailed instructions and configuration files to reproduce the main results presented in our paper.

535 Experimental details and prompts are provided in Sections I to K. Detailed derivations of our theo-
536 retical results linking DECRYPTO to pragmatic inference are in Section H.

REFERENCES

- 540
541
542 Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Llm-
543 deliberation: Evaluating llms with interactive multi-agent negotiation games. *arXiv preprint*
544 *arXiv:2309.17234*, 2023.
- 545 Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio
546 Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A
547 new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.
- 548
549 Simon Baron-Cohen, Alan M Leslie, and Uta Frith. Does the autistic child have a “theory of mind”?
550 *Cognition*, 21(1):37–46, 1985.
- 551
552 Leon Bergen, Roger Levy, and Noah Goodman. Pragmatic reasoning through semantic inference.
553 *Semantics and Pragmatics*, 9:20–1, 2016.
- 554
555 Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing*
556 *text with the natural language toolkit*. ” O’Reilly Media, Inc.”, 2009.
- 557
558 Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca
559 Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural*
information processing systems, 32, 2019.
- 560
561 Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao,
562 Mengting Hu, Yunghwei Lai, Zexuan Xiong, et al. Tombench: Benchmarking theory of mind
563 in large language models. *arXiv preprint arXiv:2402.15052*, 2024.
- 564
565 Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and
566 Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge.
arXiv preprint arXiv:1803.05457, 2018.
- 567
568 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
569 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to
570 solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- 571
572 Brandon Cui, Andrei Lupu, Samuel Sokota, Hengyuan Hu, David J Wu, and Jakob Nicolaus Fo-
573 erster. Adversarial diversity in hanabi. In *The Eleventh International Conference on Learning*
Representations, 2023.
- 574
575 Judith Degen. The rational speech act framework. *Annual Review of Linguistics*, 9(1):519–540,
576 2023.
- 577
578 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
579 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
arXiv preprint arXiv:2407.21783, 2024.
- 580
581 Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan,
582 Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-
583 agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–
584 37593, 2023.
- 585
586 Meta Fundamental AI Research Diplomacy Team FAIR, Anton Bakhtin, Noam Brown, Emily Di-
587 nan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu,
588 et al. Human-level play in the game of diplomacy by combining language models with strategic
reasoning. *Science*, 378(6624):1067–1074, 2022.
- 589
590 Michael C Frank and Noah D Goodman. Predicting pragmatic reasoning in language games. *Sci-*
591 *ence*, 336(6084):998–998, 2012.
- 592
593 Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. Understanding
social reasoning in language models with language models. *Advances in Neural Information*
Processing Systems, 36, 2024.

- 594 Tobias Gessler, Tin Dizdarevic, Ani Calinescu, Benjamin Ellis, Andrei Lupu, and Jakob Nicolaus
595 Foerster. Overcookedv2: Rethinking overcooked for zero-shot coordination. *arXiv preprint*
596 *arXiv:2503.17821*, 2025.
- 597 Noah D Goodman and Michael C Frank. Pragmatic language interpretation as probabilistic infer-
598 ence. *Trends in cognitive sciences*, 20(11):818–829, 2016.
- 600 Alison Gopnik and Janet W Astington. Children’s understanding of representational change and its
601 relation to the understanding of false belief and the appearance-reality distinction. *Child develop-*
602 *ment*, pp. 26–37, 1988.
- 603 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
604 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
605 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- 607 Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang,
608 Thomas L Griffiths, and Mengdi Wang. Embodied llm agents learn to cooperate in organized
609 teams. *arXiv preprint arXiv:2403.12482*, 2024.
- 610 Yinghui He, Yufan Wu, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-tom: A
611 benchmark for evaluating higher-order theory of mind reasoning in large language models. *arXiv*
612 *preprint arXiv:2310.16755*, 2023.
- 613 Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. “other-play” for zero-shot
614 coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.
- 615 Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief
616 learning. In *International Conference on Machine Learning*, pp. 4369–4379. PMLR, 2021.
- 618 Jennifer Hu, Felix Sosa, and Tomer Ullman. Re-evaluating theory of mind evaluation in large
619 language models. *arXiv preprint arXiv:2502.21098*, 2025.
- 620 Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song,
621 and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint*
622 *arXiv:2310.01798*, 2023.
- 624 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec
625 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv*
626 *preprint arXiv:2412.16720*, 2024.
- 627 Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik
628 Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint*
629 *arXiv:2310.06770*, 2023.
- 630 Justine T Kao, Jean Y Wu, Leon Bergen, and Noah D Goodman. Nonliteral understanding of number
631 words. *Proceedings of the National Academy of Sciences*, 111(33):12002–12007, 2014.
- 632 Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le Bras, Gunhee Kim, Yejin Choi, and Maarten
633 Sap. Fantom: A benchmark for stress-testing machine theory of mind in interactions. *arXiv*
634 *preprint arXiv:2310.15421*, 2023.
- 635 Michal Kosinski. Theory of mind may have spontaneously emerged in large language models. *arXiv*
636 *preprint arXiv:2302.02083*, 4:169, 2023.
- 637 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
638 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
639 serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating*
640 *Systems Principles*, 2023.
- 641 Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind
642 through question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.),
643 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and*
644 *the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp.
645 5872–5877, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:
646 10.18653/v1/D19-1598. URL <https://aclanthology.org/D19-1598/>.

- 648 Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and
649 Katia Sycara. Theory of mind for multi-agent collaboration via large language models. *arXiv*
650 *preprint arXiv:2310.10701*, 2023.
- 651 Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. Avalonbench: Evaluating llms playing the
652 game of avalon. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.
- 653 Jijia Liu, Chao Yu, Jiaxuan Gao, Yuqing Xie, Qingmin Liao, Yi Wu, and Yu Wang. Llm-powered hi-
654 erarchical language agent for real-time human-ai coordination. *arXiv preprint arXiv:2312.15224*,
655 2023.
- 656 Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint*
657 *arXiv:1301.3781*, 2013.
- 658 Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word rep-
659 resentation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language*
660 *Processing (EMNLP)*, pp. 1532–1543, 2014.
- 661 Jean Piaget, Bärbel Inhelder, Frederick John Langdon, and JL Lunzer. *La Représentation de*
662 *L’espace Chez L’enfant. The Child’s Conception of Space... Translated... by FJ Langdon & JL*
663 *Lunzer. With Illustrations*. New York; Routledge & Kegan Paul: London; printed in Great Britain,
664 1956.
- 665 Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew
666 Botvinick. Machine theory of mind. In *International conference on machine learning*, pp. 4218–
667 4227. PMLR, 2018.
- 668 Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre,*
669 *Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.
- 670 Melanie Sclar, Jane Yu, Maryam Fazel-Zarandi, Yulia Tsvetkov, Yonatan Bisk, Yejin Choi, and Asli
671 Celikyilmaz. Explore theory of mind: Program-guided adversarial data generation for theory of
672 mind reasoning. *arXiv preprint arXiv:2412.12175*, 2024.
- 673 Omar Shaikh, Caleb Ziems, William Held, Aryan J Pariani, Fred Morstatter, and Diyi Yang. Mod-
674 eling cross-cultural pragmatic inference with codenames duet. *arXiv preprint arXiv:2306.02475*,
675 2023.
- 676 David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez,
677 Marc Lanctot, Laurent Sifre, Dharrshan Kumaran, Thore Graepel, et al. Mastering chess and shogi
678 by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*,
679 2017.
- 680 Aaditya K Singh and DJ Strouse. Tokenization counts: the impact of tokenization on arithmetic in
681 frontier llms. *arXiv preprint arXiv:2402.14903*, 2024.
- 682 Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. Ad hoc autonomous agent teams:
683 Collaboration without pre-coordination. In *Proceedings of the AAAI Conference on Artificial*
684 *Intelligence*, volume 24, pp. 1504–1509, 2010.
- 685 Tomer Ullman. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv*
686 *preprint arXiv:2302.08399*, 2023.
- 687 Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau,
688 Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der
689 Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson,
690 Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore,
691 Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero,
692 Charles R. Harris, Anne M. Archibald, António H. Ribeiro, Fabian Pedregosa, Paul van Mul-
693 bregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing
694 in Python. *Nature Methods*, 17:261–272, 2020.

702 Wei Wang, Dan Zhang, Tao Feng, Boyan Wang, and Jie Tang. Battleagentbench: A benchmark for
703 evaluating cooperation and competition capabilities of language models in multi-agent systems.
704 *arXiv preprint arXiv:2408.15971*, 2024.
705

706 Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. Opentom: A comprehensive
707 benchmark for evaluating theory-of-mind reasoning capabilities of large language models. *arXiv*
708 *preprint arXiv:2402.06044*, 2024a.

709 Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and
710 Jiashi Feng. Magic: Investigation of large language model powered multi-agent in cognition,
711 adaptability, rationality and collaboration. In *Proceedings of the 2024 Conference on Empirical*
712 *Methods in Natural Language Processing*, pp. 7315–7332, 2024b.

713 Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. On the tool
714 manipulation capability of open-source large language models, 2023a.
715

716 Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu.
717 Exploring large language models for communication games: An empirical study on werewolf.
718 *arXiv preprint arXiv:2309.04658*, 2023b.

719 Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning
720 for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*, 2023c.
721

722 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao.
723 React: Synergizing reasoning and acting in language models. In *International Conference on*
724 *Learning Representations (ICLR)*, 2023.

725 Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav
726 Raja, Dylan Slack, Qin Lyu, et al. A careful examination of large language model performance
727 on grade school arithmetic. *arXiv preprint arXiv:2405.00332*, 2024a.
728

729 Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li,
730 Yueting Zhuang, and Weiming Lu. Agent-pro: Learning to evolve via policy-level reflection and
731 optimization. *arXiv preprint arXiv:2402.17574*, 2024b.

732 Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R McKee, Ari Holtzman,
733 Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, et al. How far are large language
734 models from agents with theory-of-mind? *arXiv preprint arXiv:2310.03051*, 2023.
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

A LIMITATIONS AND FUTURE WORK

We firmly believe DECRYPTO to be a valuable and future-proof benchmark for evaluating multi-agent reasoning and theory of mind. However, our work still has a number of limitations, which we hope to address in future work.

First, while we largely eliminated the confounding factors that affect LLM performance, the game dynamics and the ToM tasks may still be too complicated for weaker models. For instance, when trying Llama 3.1-8B-Instruct on the perspective taking task, we found that it failed to understand the prompt altogether.

Regarding evaluations, the main limitation of our work is the computational cost of computing Figure 3. An individual run is relatively cheap (approximately \$0.10 per role per episode for Claude 3.7 Sonnet). However, given the 3-player nature of the game, the total number of combinations for n models is n^3 , or $2n^2$ if only computing the matrix slices in Figure 3. For a large number of models, this can induce significantly larger costs than standard single-agent benchmarks. To mitigate this, we provide the tools to run evaluations locally and efficiently, leveraging prefix caching where possible to reduce unnecessary computations. We also acknowledge the limited scope of our human study in Table 1. While the 10 games collected were sufficient to demonstrate the limitations of LLMs in human-AI interactions, we believe that there is much to be learned from a more extensive study, including with humans as interceptors. We therefore made sure to provide the tools to enable such research, such as a lightweight command-line interface for conducting human studies.

Finally, theory of mind is an umbrella term for multiple abilities. Our work evaluates some of them, such as representational change, false belief and perspective taking, and we hope extensions to DECRYPTO to expand on this set in the future. Nevertheless, there are ToM abilities which DECRYPTO is ill-fitted to evaluate. For instance, models performing well in the game and the ToM tasks we designed may still fail to reason about the intentions or emotions of other agents.

DECRYPTO also enables a number of other research directions connected to ToM, multi-agent reasoning and reinforcement learning, and pragmatic inference. From a ToM perspective, DECRYPTO is easy to extend with additional ToM experiments, which can complement the results presented in our work. Given the limitations of current frontier models, we also believe there is a dire need for methods to improve the ToM abilities of LLMs. DECRYPTO features short episodes and dense rewards, making it a suitable environment to study the application of multi-agent RL algorithms towards LLM post-training. In particular, DECRYPTO features stochasticity and partial observability, two key ingredients for “meaningful” coordination challenges (Ellis et al., 2023). It would therefore be interesting to study whether fine-tuning LLMs on DECRYPTO using self-play (Silver et al., 2017) or other MARL algorithms (Hu et al., 2020) produces the same kind of idiosyncrasies observed when training *tabula rasa* models in other cooperative environments (Cui et al., 2023; Gessler et al., 2025). Finally, we believe the study of pragmatic inference in LLMs to be a promising avenue for future research, which is made much easier by the release of our benchmark. We formalise DECRYPTO as a pragmatic inference game and provide suggestions for this line of work in Section H.

B IMPACT STATEMENT

As LLMs become more prevalent in our daily lives and start interacting with each other and with human users, we believe it becomes crucial to understand the multi-agent and ToM abilities of foundation models. Our work provides a novel assessment of those abilities, and the tools to conduct future studies on inter-AI and human-AI interactions. As such, we hope to bring attention to what we consider to be an understudied subset of agentic capabilities, and that our work will lead to models with a strong social and human-centric behaviour.

However, ToM encompasses a broad set of abilities, of which DECRYPTO evaluates only a subset. A possible misuse of DECRYPTO is to equate strong performance on our benchmark to more general ToM abilities, including intent inference and emotion understanding, which are not evaluated by DECRYPTO. Such claims have incorrectly been made about other ToM benchmarks (Hu et al., 2025), and so we ask future researchers to be careful when concluding that an AI system has ToM.

C RATIONALE FOR THE CODE IN FIGURE 1

The code that Alice encrypted in Figure 1 is [4-1-3]. The rationale is as follows. The four keywords are [1. star, 2. jazz, 3. thunder, 4. plane], and the hints are [two, clone, AC/DC]. “Two” refers to “two dimensions”, which is the defining characteristic of a geometric plane. “Clone” is a hint for “star”, since both “clone” and “star” are common operations performed on a GitHub repository. “AC/DC” is a rock band, and one of their most famous songs, “Thunderstruck” starts with the word “Thunder” being chanted repeatedly.

The choice of hints relies on multiple assumptions made by Alice, here played by the authors, about Bob, played by the readers. The first hint relies on the assumption that many readers will have a mathematical background and quickly think of “plane” in its geometric sense, rather than as an aircraft. Similarly, we assume for the second hint that readers will be sufficiently familiar with GitHub to make the same connection as us. Failing that, we also anticipated that some readers might link the two terms by thinking of the “clone troopers” from Star Wars, which also leads to a correct guess. Finally, we assumed that the song “Thunderstruck” is famous enough to be picked up by most readers.

This is an example of how Alice can use ToM and the specific information she has about another player, in this case the readers, to inform her choice of hints. We hope we were successful.

D BASELINE DETAILS

Here we provide additional details on the word-embedding baselines introduced in Section 4.

The baselines work as follows. Alice can choose hints from a fixed corpus of the 5696 most common nouns extracted from the Brown, Gutenberg, and Webtext corpora (Bird et al., 2009), filter to ensure all extracted nouns are present in both the GloVe and Word2Vec embeddings. (We use this hint corpus for the baselines only; LLM agents generate hints in an open-ended fashion.)

Before Alice chooses hints, Alice and Bob agree on a strategy. For each code digit $c_i \in [1, 4]$, $i \in [1, 3]$, Alice picks a hint h_i and guarantees that the hint picked for the digit is more similar to the c_i -th keyword k_{c_i} than to any other keyword, i.e., $s(h_i, k_{c_i}) > s(h_i, k_j) \quad \forall j \in -\mathbf{c}$, where $-\mathbf{c} = [1, 4] \setminus c_i$ and s is the cosine similarity.

Alice increases the diversity of her hints by randomly sampling each hint from the top- K most similar words for each keyword under cosine similarity, filtering those that do not satisfy the constraint above. We also enforce that Alice cannot reuse any previously used hints in the same episode.

Bob then analyses the hints and assigns each hint to its most similar keyword under cosine similarity, i.e., $\arg \max_{c \in [1, 4]} s(h_i, k_c) \quad \forall i \in [1, 3]$. Bob guesses the code perfectly if Alice and Bob use the same word embeddings because the hints are guaranteed to be the most similar to the correct keyword under the same embedding model.

Eve is only given the hints \mathbf{h} and the hint history for each keyword, which at turn 0 is empty. Eve calculates the cosine similarity between the average embedding of each keyword’s hint history with the hints. This results in a similarity matrix of size $N \times M$, where $N(= 3)$ is the number of hints and $M(= 4)$ is the number of keywords. Eve combinatorially calculates the globally optimal guess based on the similarity values. Since N and M are typically small values, we are unconcerned about the computational complexity. However, Eve’s selection task is equivalent to a linear assignment problem and efficient algorithms, such as the Jonker-Volgenant algorithm, are available in open-source libraries such as SciPy (Virtanen et al., 2020).

E DIAGRAMS FOR THEORY OF MIND EXPERIMENTS

[Diagrams moved to the main text. Keeping Appendix header for now to not change the indexing of other Appendices. It will be removed for the camera-ready copy.]

F EXPERIMENTAL DETAILS

Here we provide details for the experiments reported in Section 5.

F.1 PROMPTING

All our experiments are text-only and conducted exclusively through prompts, which we provide in Sections I and J. The system prompt is composed of an explanation of the game rules, common to all roles, and a set of role-specific instructions. The user prompts are role-specific but follow a similar structure, providing, in order, a summary of the previous turn and of the game so far, role specific information (e.g. the code or the hints), a summary of the role task, and formatting instructions. For the ToM tasks, the prompts are structure similarly, and rely on the same output formatting. For agents that do not admit a system prompt, such as Deepseek-R1, we include the system prompt as a prefix to the first user message.

We maintain prompts and model outputs in context, making DECRYPTO a multi-turn environment. We use a regular expression to extract the hints or the guess from the model outputs. If the code fails to detect a correctly-formatted answer in the output, we re-prompt the model again up to 10 times, reminding it of the correct output format. If the model has still not provided an answer after 10 attempts, we default to a dummy answer. In practice, this almost never happens unless the output token limit is set very low for very verbose models, such as Deepseek-R1. By default, retries are not added to the model’s context to avoid cluttering it and unnecessarily lowering its performance. ToM prompts and outputs are also excluded from the context for future turns.

F.2 GENERATION PARAMETERS

We set the model temperature to 0.6 for experiments in Figure 3 and table 1, and to 0 for ToM experiments in Figure 6 to reduce the impact of model stochasticity. We note that some close source models are not entirely deterministic, even with the temperature set to 0.

Our goal was not to evaluate model verbosity, but rather the limit of their capabilities. Therefore, we set generous token limits (between 750 for non-reasoning models and up to 10000 for reasoning ones) to prevent cutting model generations prematurely. We found that models varied widely in their verbosity. For instance, Claude 3.7 Sonnet with Extended Thinking output approximately 500 total tokens per turn on average, and sometimes as low as 350 despite a limit of 1500 tokens. In contrast, some responses by o1-high and o3-high exceeded 8000 tokens.

The full set of generation parameters can be found in the configs we provide as part of our code release, and which enable reproduction of the results in our paper with only a few commands.

F.3 COMPUTATIONAL COSTS

We ran our experiments on a mix of local and API-hosted models.

All open-source models were hosted locally on a high performance computing cluster using vLLM (Kwon et al., 2023). Each model instance ran on 8x Nvidia A100 GPUs. Although not necessary, we also launched multiple instances of the same model running, using concurrency to accelerate inference.

System prompts are on the order of 1000 tokens, and the user prompts at each turn are less than 300 tokens each. Models will typically output around 500 tokens per prompt, but the number can go much higher for particularly verbose models. We estimate that evaluating Claude 3.7 Sonnet with Extended Thinking cost approximately \$0.10 per role per episode. A full run of 32 episodes, 3 random seeds, and with all three roles played by Claude 3.7 Sonnet costs approximately \$30.

Finally, we detail human data collection in Section K.

G ADDITIONAL RESULTS

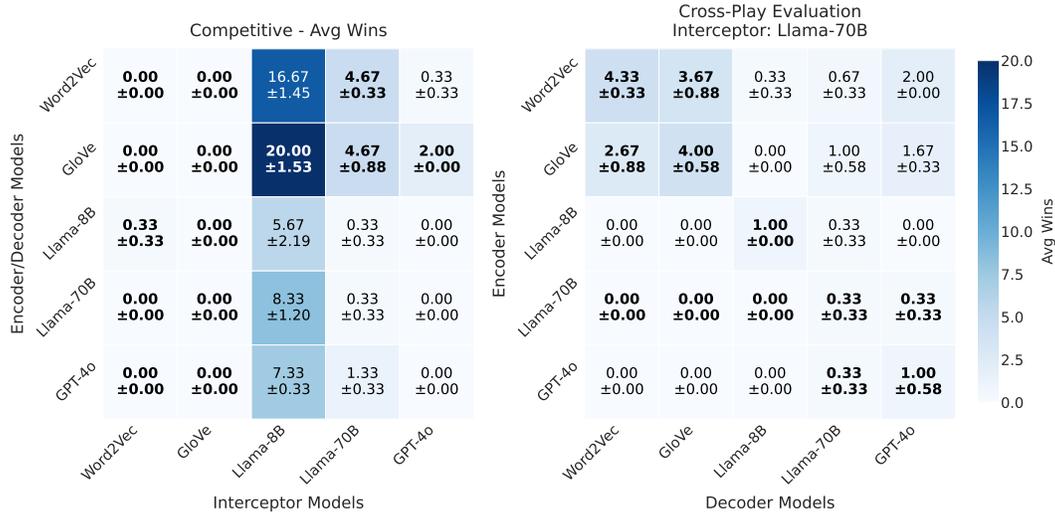


Figure 7: Number of survived games over 32 games for 5 different agents. Mean and standard deviation computed over 3 model seeds. We find the abilities of the current generation of LLMs to be skewed, such that Eve wins a vast majority of the games. As models improve and can come up with better hints, we expect the balance to shift.

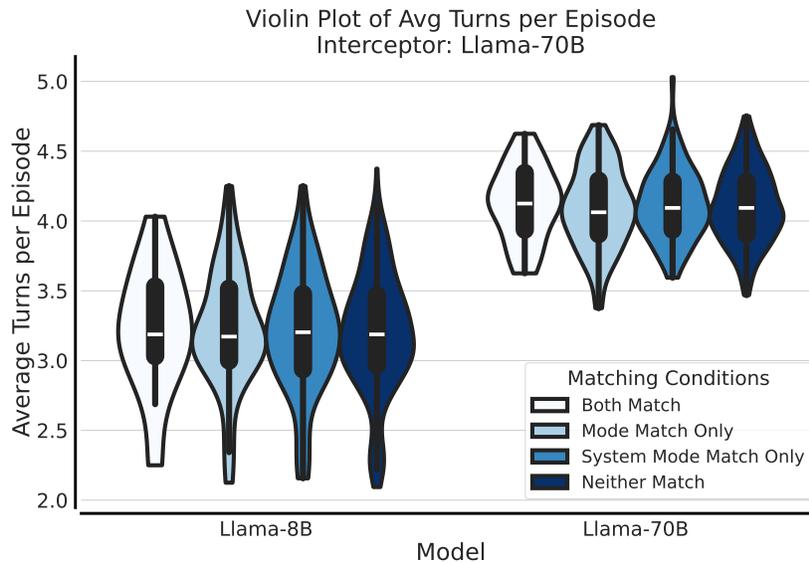


Figure 8: Distribution of the number of turns per game as we vary the system and user prompts for both encoder and decoder. We handwrite 5 system and 5 user prompts (i.e. “modes”) for the encoder and the decoder and plot the distributions of games when only their system prompts match, when only their user prompts match, when both match or when neither do (625 total combinations per model). We find that both models tested are robust to significant prompt variations, with model size remaining the most significant predictor of performance.

Robustness. We show that prompt variants do not significantly affect the final performance measured by average turn length for Llama-3.1-8B and Llama-3.1-70B, see Figure 8. This suggests that

the poor performance of LLMs in DECRYPTO is more likely due to a lack of reasoning abilities than a lack of prompt tuning. We look at the robustness of different models to variations in prompts and generation parameters, keeping in mind that generalist agents cannot control their prompt since it is assumed to be part of the environment. We handcraft 5 system and user prompts for Alice and Bob, respectively. The system prompt consists of 2 components. One component is responsible for explaining the game rules in general, for which we have 5 variants. The second component explains the specific role. We have 5 prompt variants for each role, already resulting in 125 different prompt combinations. The user prompt instructs the specific roles to take their actions, for which we have 5 variants. In total, this results in 625 different prompt setups for each model. We run 32 games over 3 model seeds per system/user prompt combination and measure the average game length.

H DECRYPTO AS A PRAGMATIC INFERENCE GAME

With mild assumptions, we can formalize DECRYPTO as an iterated pragmatic inference game in the Rational Speech Act (RSA) framework introduced by Frank & Goodman and extended in Goodman & Frank (2016). Doing so provides a Bayesian view of how successful communication in the game requires agents to model each other’s decision process.

We consider each turn in isolation, so that each episode of DECRYPTO is composed of up to 8 independent pragmatic inference games. Agents must associate utterances u (i.e. a set of 3 hints) to an intended meaning m (the code for that turn or, equivalently, the corresponding combination of 3 keywords).

The RSA framework starts by defining a hypothetical *literal listener*, which interprets utterances according to their semantics:

$$P_{\text{Lit}}(m|u) \propto \delta_{m \in [[u]]} P(m). \quad (1)$$

Here $P(m)$ is the probability of a given meaning and $\delta_{m \in [[u]]}$ is the publicly known *lexical function*, which returns 1 if the meaning m is *semantically compatible* with utterance u and 0 otherwise. The literal listener therefore assigns 0 probability to meanings that are semantically incompatible with the utterance, but otherwise defers to the normalized prior probability of a meaning. In DECRYPTO, all meanings are equally likely, so $P(m) = \frac{1}{|M|}$, where M is the set codes that have not been seen on previous turns.

The game rules stipulate that hints must refer to the meaning of the keywords. Therefore, $[[u]]$ can be defined as the subset of all combinations of three keywords $m \in M$ that have a semantic link to the hints. For instance, suppose the keywords are [star, jazz, thunder, plane], as in Figure 1, and the hints are $u' = [\text{fusion}, \text{Zeus}, \text{pilot}]$. Then,

$$m_1 := [\text{star}, \text{thunder}, \text{plane}] \equiv [1, 3, 4] \in [[u']], \quad (2)$$

$$m_2 := [\text{jazz}, \text{thunder}, \text{plane}] \equiv [2, 3, 4] \in [[u']], \quad (3)$$

among other meanings, and $P_{\text{Lit}}(m_1|u') = P_{\text{Lit}}(m_2|u') = \frac{1}{|[[u']]|}$. This is because stars are fuelled by nuclear fusion and “jazz fusion” is a popular music genre.

Alice is then assumed to be an *approximately rational speaker*, who chooses utterances according to a probability

$$P_{\text{Alice}}(u|m) \propto e^{\lambda U(u,m)}, \quad (4)$$

where λ captures the degree of optimality and $U(u, m)$ is the utility of speaking utterance u to communicate meaning m . To avoid infinite recursion, the speaker’s *utility* in RSA balances the utterance cost with the probability that the literal listener will interpret the utterance correctly:

$$U(u, m) = \log P_{\text{Lit}}(m|u) - \text{cost}(u, m).$$

The RSA framework typically assumes a single speaker and collaborative listener, but we can capture the competitive dynamics between Alice and Eve by letting the utterance cost be a function of the probability that Eve intercepts the code. There are multiple possible candidate functions, but since miscommunication and intercepts play an equally important role in DECRYPTO, we let

$$\text{cost}(u, m) = \varepsilon \log(1 - P_{\text{Eve}}(m|u)). \quad (5)$$

Here, $P_{\text{Eve}}(m|u)$ denotes Eve’s probability of intercepting the code and $\varepsilon \in [0, 1]$ captures how bad an intercept would be on that turn. For instance, if this is the eighth turn and no intercept has occurred so far in the game, getting intercepted will not impact Alice and Bob’s chances of winning, so we set $\varepsilon = 0$. Similarly, we can add a variable $\beta \in [0, 1]$, which captures the importance of not miscommunicating on this turn. This makes Alice’s utility

$$U(u, m) = \beta \log P_{\text{Lit}}(m|u) + \varepsilon \log(1 - P_{\text{Eve}}(m|u)). \quad (6)$$

Finally, Bob, the *pragmatic listener*, decodes utterances based on the common prior over meanings and to a model of the speaker, with probability

$$P_{\text{Bob}}(m|u, \text{Alice}) \propto P(m)P_{\text{Alice}}(u|m) \quad (7)$$

Expanding Equation (7), we get

$$P_{\text{Bob}}(m|u, \text{Alice}) \propto \frac{1}{|M|} e^{\lambda U(u, m)} \quad (8)$$

$$= \frac{1}{|M|} e^{\lambda(\beta \log P_{\text{Lit}}(m|u) + \varepsilon \log(1 - P_{\text{Eve}}(m|u)))} \quad (9)$$

$$= \frac{1}{|M|} P_{\text{Lit}}(m|u)^{\lambda\beta} (1 - P_{\text{Eve}}(m|u))^{\lambda\varepsilon}. \quad (10)$$

Hence, in all cases where $\varepsilon > 0$, **Bob must model how Alice models Eve in order to interpret the hints correctly**. This is a clear example of why multi-step theory of mind is necessary for optimal play in DECRYPTO.

The listener rarely has a perfect model of the speaker. Goodman & Frank discuss *uncertain RSA* (uRSA), a common extension to the RSA framework where the pragmatic listener is uncertain about the model underlying the speaker’s behaviour (Kao et al., 2014). Under uRSA, the pragmatic listener has a distribution over possible speakers s , and so Bob has a joint probability over speaker and interpretation:

$$P_{\text{Bob}}(m, s|u) \propto P(m)P(s)P_s(u|m, s). \quad (11)$$

Marginalizing over speakers gives us the probability that Bob interprets the hints correctly:

$$P_{\text{Bob}}(m|u) \propto P(m) \sum_s P(s)P_s(u|m, s) = P(m)P_{\text{marginal}}(u|m), \quad (12)$$

with $P_{\text{marginal}}(u|m) := \sum_s P(s)P_s(u|m, s)$. Comparing this to equation 7 highlights the fact that Bob’s probability of interpreting the hints correctly depend on how close the marginal probability $P_{\text{marginal}}(u|m)$ is to the real utterance probability $P_{\text{Alice}}(u|m)$. This provides an entry point for studying cross-cultural pragmatics in DECRYPTO, expanding upon work done by Shaikh et al. (2023) in Codenames: if Alice provides hints informed by her cultural background, but such background is unlikely under Bob’s distribution, we expect the number of miscommunications between them to increase significantly.

H.1 MODELLING EVE

Under the RSA framework, Alice’s degree of optimality is determined by λ , which can be thought to capture the bounded rationality of the speaker. However, our definition of cost in equation 5 introduces a second failure mode in the ability to model Eve accurately.

To show this, suppose Alice only has access to a proxy model $\tilde{P}(m|u)$ of Eve, rather than to her true probabilities. Alice therefore chooses utterances according to probability $\tilde{P}_{\text{Alice}}(u, m)$, which follows the same form as equation 4, except using the proxy utility

$$\tilde{U}(u, m) = \beta \log P_{\text{Lit}}(m|u) + \varepsilon \log(1 - \tilde{P}(m|u)). \quad (13)$$

We can use this to derive Alice’s expected (real) utility for each meaning m :

$$\mathbb{E}\{U(m)\} = \sum_{u \in \mathcal{U}} \tilde{P}_{\text{Alice}}(u|m)U(u, m) \quad (14)$$

$$= \sum_{u \in \mathcal{U}} \frac{P_{\text{Lit}}(m|u)^{\lambda\beta} (1 - \tilde{P}(m|u))^{\lambda\varepsilon}}{Z(m)} [\beta \log P_{\text{Lit}}(m|u) + \varepsilon \log(1 - P_{\text{Eve}}(m|u))], \quad (15)$$

where $Z(m) := \sum_{u'} P_{\text{Lit}}(m|u')^{\lambda\beta} (1 - \tilde{P}(m|u'))^{\lambda\varepsilon}$.

For simplicity, we let $\varepsilon = 1$ and we can eliminate the terms depending on $P_{\text{Lit}}(m|u)$. This corresponds to scenarios where $\beta = 0$ because miscommunication does not matter, or where $P_{\text{Lit}}(m|u) = 1$ for all u under consideration. The latter is not as strong an assumption as it seems. The space of all possible utterances, even if restraining hints to be single words, is equal to the entire English vocabulary cubed. Constraining Alice to utterances with a unique interpretation given the keywords (s.t. $|\llbracket[u]\rrbracket| = 1$) is therefore reasonable, and still allows for a very large number of possible utterances. In fact, this is very similar to the way the word embedding baselines in Figure 2 only choose hints that are guaranteed to be interpreted correctly by Bob. Given these conditions, the task becomes exclusively to minimize the probability of interception:

$$\mathbb{E}\{U(m)\} = \sum_{u \in \mathcal{U}} \tilde{P}_{\text{Alice}}(u|m) \log(1 - P_{\text{Eve}}(m|u)) \quad (16)$$

$$= \sum_{u \in \mathcal{U}} \frac{(1 - \tilde{P}(m|u))^\lambda}{Z(m)} \log(1 - P_{\text{Eve}}(m|u)). \quad (17)$$

Under the same simplifications, the utterance probability under the RSA framework if Alice had access to the real $P_{\text{Eve}}(m|u)$ is given by

$$P_{\text{true}}(u|m) := \frac{(1 - P_{\text{Eve}}(m|u))^\lambda}{Z_{\text{true}}(m)}, \quad (18)$$

where $Z_{\text{true}}(m) := \sum_{u'} (1 - P_{\text{Eve}}(m|u'))^\lambda$.

Combining equation 16 and equation 18, we obtain

$$\mathbb{E}\{U(m)\} = \sum_{u \in \mathcal{U}} \tilde{P}_{\text{Alice}}(u|m) \frac{1}{\lambda} [\log(P_{\text{true}}(u|m)) + \log(Z_{\text{true}}(m))] \quad (19)$$

$$= \frac{1}{\lambda} \sum_{u \in \mathcal{U}} \tilde{P}_{\text{Alice}}(u|m) [\log\left(\frac{P_{\text{true}}(u|m)}{\tilde{P}_{\text{Alice}}(u|m)}\right) + \log(\tilde{P}_{\text{Alice}}(u|m)) + \log(Z_{\text{true}}(m))] \quad (20)$$

$$= -\frac{1}{\lambda} D_{\text{KL}}(\tilde{P}_{\text{Alice}} || P_{\text{true}}) - \frac{1}{\lambda} H(\tilde{P}_{\text{Alice}}) + \frac{1}{\lambda} \log(Z_{\text{true}}(m)), \quad (21)$$

with $D_{\text{KL}}(\cdot || \cdot)$ denoting the Kullback–Leibler (KL) divergence and $H(\cdot)$ the entropy.

Therefore, for a given λ , the log likelihood of getting intercepted by Eve depends linearly on the KL divergence between \tilde{P}_{Alice} and P_{true} , where

$$\begin{aligned} \tilde{P}_{\text{Alice}}(u|m) &\propto (1 - \tilde{P}(m|u))^\lambda \\ P_{\text{true}}(u|m) &\propto (1 - P_{\text{Eve}}(m|u))^\lambda. \end{aligned}$$

From this it follows that expected utility is only maximised if $\tilde{P}(m|u) = P_{\text{Eve}}(m|u) \forall u$. In the limit of $\lambda \rightarrow \infty$, Alice concentrates her utterance probability on $u^* := \arg \max_u \tilde{U}(u, m)$. In such case, the expected utility is given by

$$\lim_{\lambda \rightarrow \infty} \mathbb{E}\{U(m)\} = \log(1 - P_{\text{Eve}}(m|u^*)). \quad (22)$$

This demonstrates that even if miscommunication is no issue and Alice is perfectly rational, she still incurs a risk of interception and a lower utility if her model of Eve is inaccurate.

However, we note that an imperfect model of Eve does not fully account for the behaviour we observe in LLMs. Indeed, our ToM results in section Section 5 show that LLMs not only fail to predict Eve’s guess, they often fail to consider the cost of intercepts altogether when choosing hints.

H.2 FURTHER REFINEMENTS

Our formalism assumes $\llbracket[u]\rrbracket$ to be shared by Alice and Bob, in line with classical RSA. This may not always be the case. Revisiting the example from equation 2, m_1 may not be in $\llbracket[u']\rrbracket$ for a player who has never heard of nuclear fusion, or who is unaware of its role in stellar physics. Differences

in lexicons arise naturally if players have different world knowledge and in the authors’ experience, are often the cause of miscommunications when playing DECRYPTO. Such differences can also be exploited, and we provide in Section 3 the example of a model referring to events that occurred after their opponent’s knowledge cut-off.

Evaluating LLM abilities to play DECRYPTO and communicate under lexical differences is a promising avenue for future work. We suggest that readers interested in this direction start with the study of RSA under lexical uncertainty in Bergen et al. (2016).

I SYSTEM PROMPTS

```

1 You are playing a variant of the code guessing game Decrypto. The setup
2 of the game is the following:
3 The game is played with three players split into two teams.
4 The first team is composed of the Encoder and the Decoder.
5 The second team is composed of the Interceptor.
6 The Encoder and Decoder share a list of 4 secret keywords that they can
7 rely on to help communication. Each keyword has a digit associated with
8 it.
9 For example, if the keywords are {airplane, guitar, hat, plant}, the
10 digits associated with them would be {1, 2, 3, 4}.
11 The keywords are fixed for the entire game, so the Encoder and Decoder
12 must not reveal them.
13 Each turn, the Encoder gets a secret code of 3 non-repeating digits
14 between 1 and 4, e.g. 3-1-2.
15 They have to then provide 3 hints (one for each code digit) such that the
16 Decoder can guess the code but the Interceptor cannot.
17 Then, the Decoder and the Interceptor make their guesses and the code is
18 revealed.
19 The Interceptor wins an Interception token if they successfully guess the
20 code.
21 The Encoder and Decoder get a Miscommunication token if the Decoder fails
22 to guess the code.
23 The Interceptor wins if it gets two Interception tokens, or if the
24 Encoder and Decoder get two Miscommunication tokens.
25 The Encoder and Decoder team wins if it makes it through 8 rounds without
26 triggering two Miscommunications or two Interceptions.
27 The hints must be one or two words, and make a reference to the meaning
28 of the secret keyword of the corresponding digit.
29 Codes are unique; the same code cannot occur twice in the same game.
30 For example, if the four secret keywords are
31 {1: "airplane", 2: "guitar", 3: "hat", 4: "plant"}
32 and the code for that turn is 3-1-2, which means that we need hints for
33 {3: hat, 1: airplane, 2: guitar}, the Encoder could hint
34 {3: gentleman, 1: wing, 2: music}. Note that the order of the hints is
35 important.
36 Ignore these keywords in the future.
37 When responding, first think through your decision step-by-step. After
38 your step-by-step thinking, provide your final answer as a dictionary,
39 clearly marked with "ANSWER:".

```

Listing 1: This is an example of the basic system prompt that we provide to Alice, Bob, and Eve at the beginning of each game. It explains the roles and outline the rules. In our robustness evaluation, we tested five variants of the system prompt.

```

1188 1 You are the Encoder. Provide your hints like "ANSWER: {"hints":
1189 ["hint_X", "hint_Y", "hint_Z"]}", where hint_X, hint_Y, hint_Z are one or
1190 two words each. Make sure that the ordering of the hints follows the
1191 order of the code.
1192 2 For example:
1193 3 '''
1194 4 To provide the hints, I need to think about the meaning of each keyword
1195 5 and come up with a one or two-word hint that makes a reference to it.
1196 6 For the code 2-1-3, I need to give hints about the keywords associated
1197 7 with the digits 2, 1, and 3, which are "hat", "fire", and "answer"
1198 8 respectively.
1199 9 Here's my step-by-step thinking:
1200 10
1201 11 - For the digit 2, the keyword is "hat". Since the previous hint for this
1202 12 keyword was "top", I want to give a hint that is different but still
1203 13 related to wearing a hat. One possible hint is "cap".
1204 14 - For the digit 1, the keyword is "fire". The previous hint for this
1205 15 keyword was "heat", so I want to give a hint that is related to fire but
1206 16 different from "heat". One possible hint is "flame".
1207 17 - For the digit 3, the keyword is "answer". A possible hint could be
1208 18 something related to giving an answer. One possible hint is "solve".
1209 19
1210 20 So, the final hints are:
1211 21 {"hints": ["cap", "flame", "solve"]}
1212 22
1213 23 ANSWER: {"hints": ["cap", "flame", "solve"]}'''

```

Listing 2: This is the Alice-system prompt. This prompt is appended to the system prompt for Alice and gives one example of how Alice might reason about her hints.

```

1220 1 You are the Decoder. Provide your guess like "ANSWER: {"guess":
1221 X-Y-Z"}". Make sure that the ordering of the guess follows the order of
1222 the hints.
1223 2 For example:'''
1224 3 1. The hint for digit 'a' is 'cap', which doesn't seem to directly relate
1225 4 to any of the keywords. However, considering the previous hint history
1226 5 (top, heat, lab), we can look for a connection between 'cap' and one of
1227 6 the keywords. The keyword 'hat' is associated with digit '2', and a hat
1228 7 is often worn on the head, which is covered by a cap. So, we can make a
1229 8 connection: {a: hat}.
1230 9 2. The hint for digit 'b' is 'flame', which is related to heat and fire.
1231 10 Since the keyword for digit '1' is 'fire', we can make a connection: {b:
1232 11 f re}.
1233 12 3. The hint for digit 'c' is 'solve', which is related to the keyword
1234 13 associated with digit '3', 'answer'. Since 'answer' is a solution to a
1235 14 problem, we can make a connection: {c: answer}.
1236 15 4. Now that we have the connections, we can determine the code by looking
1237 16 at the order of the hints: {a: hat, b: fire, c: answer}. Therefore, the
1238 17 code is {2-1-3}.
1239 18
1240 19 Final answer:
1241 20 ANSWER: {"guess": "2-1-3"}'''

```

Listing 3: This is the Bob-system prompt. This prompt is appended to the system prompt for Bob and gives one example of how Bob might reason about his guess.

```
1242 1 You are the Interceptor. You do not have access to the four keywords.
1243 2 Provide your guess like "ANSWER: {"guess": "X-Y-Z"}" Make sure that the
1244 3 ordering of the guess follows the order of the hints.
1245 4 For example:'''
1246 5 1. The hint for 'a' is 'income', which is related to money, and in the
1247 6 previous turn, 'salary' was also related to money, which is associated
1248 7 with keyword 2.
1249 8 2. The hint for 'b' is 'power', which is related to energy or control,
1250 9 and in the previous turn, 'season' was related to sports, which is often
1251 10 associated with power or strength.
1252 11 3. The hint for 'c' is 'fashion', which is related to style or clothing,
1253 12 and in the previous turn, 'headgear' was directly related to a type of
1254 13 hat, which is a type of clothing.
1255 14 Considering possible connections:
1256 15 - 'income' is related to money, and keyword 2 is associated with money.
1257 16 - 'power' is related to energy or control, and keyword 4 is associated
1258 17 with energy or control.
1259 18 - 'fashion' is related to style or clothing, and keyword 3 is associated
1260 19 with clothing.
1261 20 Based on these connections, I'll make a guess that the code is related to
1262 21 keywords 2, 4, and 3.
1263 22 Final Answer:
1264 23 ANSWER: {"guess": "2-4-3"}'''
1265 24 Remember that the keywords could be anything. Use the most recent hint
1266 25 history to your advantage.
1267 26 Remember that digits do not repeat in the code and range from 1 to 4.
```

Listing 4: This is the Eve-system prompt. This prompt is appended to the system prompt for Eve and gives one example of how Eve might reason about her guess.

J USER PROMPTS

1295

```

1296 1 Turn 1 summary:
1297 2 Code: 3-1-4
1298 3 Hints: ['problem', 'status', 'machine']
1299 4 Decoder guess: 3-1-4
1300 5 Interceptor guess: 1-2-3
1301 6
1302 7 Hint History:
1303 8 Keyword 1: status
1304 9 Keyword 2:
1305 10 Keyword 3: problem
1306 11 Keyword 4: machine
1307 12 Code History: 3-1-4
1308 13
1309 14 Turn 2: 0 Miscommunications, 0 Interceptions so far.
1310 15 You are the Encoder.
1311 16 The four keywords are: {1: condition, 2: task, 3: issue, 4: device}
1312 17 The code is 3-4-1, which corresponds to the keywords {3: issue, 4:
1313 18 device, 1: condition}.
1314 19 First, think out loud, step-by-step about what hints you should use. Use
1315 20 the meaning of the keywords to come up with a one or two-word hint for
1316 21 each digit of the code.
1317 22 Make sure the order of the hints matches the order of the code.
1318 23 Then provide your three hints like {"ANSWER: {"hints": [{"hint_X"},
1319 24 {"hint_Y"}, {"hint_Z"}]}}, where hint_X, hint_Y, hint_Z are your hints.

```

Listing 5: The user prompt for Alice is provided at each turn. It provides a turn summary of previous turns. In this example, one turn has been played and Alice sees that Bob guessed the code correctly, whereas Eve did not. Alice also sees the previously used hints. We also provide information about how many miscommunications and interceptions happened so far. Then we provide Alice with the current code and instruct her to give the hints for that code.

```

1324 1 Turn 1 summary:
1325 2 Code: 3-1-4
1326 3 Hints: ['problem', 'status', 'machine']
1327 4 Decoder guess: 3-1-4
1328 5 Interceptor guess: 1-2-3
1329 6
1330 7 Hint History:
1331 8 Keyword 1: status
1332 9 Keyword 2:
1333 10 Keyword 3: problem
1334 11 Keyword 4: machine
1335 12 Code History: 3-1-4
1336 13
1337 14 Turn 2: 0 Miscommunications, 0 Interceptions so far.
1338 15 You are the Decoder.
1339 16 The four keywords are: {1: condition, 2: task, 3: issue, 4: device}
1340 17 The hints given by the Encoder for this turn are: {a: conflict, b: tool,
1341 18 c: state}
1342 19 For example, you might think the following connections are true {a: X, b:
1343 20 Y, c: Z}, where X, Y, Z are non-repeating digits from 1 to 4.
1344 21 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}.
1345 22 What is your guess for the three-digit code? Apply concise, step-by-step
1346 23 thinking, double-check the order, and then provide your final answer as
1347 24 {"ANSWER: {"guess": "X-Y-Z"}}.

```

Listing 6: The user prompt for Bob is provided at each turn. It provides a turn summary about previous turns. In this example, one turn has been played and Bob sees the code of the previous turn and that he guessed correctly, whereas Eve did not. Bob also sees the previously used hints assigned to the correct keywords, in case he did not guess correctly. We provide information about how many miscommunications and interceptions happened so far. Then we provide Bob with the current hints and instruct him to guess.

```

1350 1 Turn 1 summary:
1351 2 Code: 3-1-4
1352 3 Hints: ['problem', 'status', 'machine']
1353 4 Decoder guess: 3-1-4
1354 5 Interceptor guess: 1-2-3
1355 6
1356 7 Hint History:
1357 8 Keyword 1: status
1358 9 Keyword 2:
1359 10 Keyword 3: problem
1360 11 Keyword 4: machine
1361 12 Code History: 3-1-4
1362 13
1363 14 Turn 2: 0 Miscommunications, 0 Interceptions so far.
1364 15
1365 16 You are the Interceptor.
1366 17 The hints given by the Encoder for this turn are: {a: conflict, b: tool,
1367 18 c: state}
1368 19 For example, you might think the following connections are true: {a: X,
1369 20 b: Y, c: Z}, where X, Y, Z are non-repeating digits from 1 to 4.
1370 21 Your guess should be in the order of the hints: {"guess": "X-Y-"}
1371 22 Use the hint history to your advantage.
1372 23 What is your guess for the three-digit code? Reply with very concise,
1373 24 step-by-step thinking, followed by your final answer as "ANSWER:
1374 25 {"guess": "X-Y-Z"}"

```

Listing 7: The user prompt for Eve is provided at each turn. It provides a turn summary of previous turns. In this example, one turn has been played and Eve sees the code of the previous turn and that she guessed incorrectly, whereas Bob did not. Eve also sees the previously used hints assigned to the correct keywords, allowing Eve to reason about the current hints. We provide information about how many miscommunications and interceptions happened so far. Then we provide Eve with the current hints and instruct her to guess.

```

1381 1 As the Encoder, you've provided the following hints for the code 3-4-1:
1382 2 {conflict, tool, state}
1383 3
1384 4 What do you predict will be the guess of the Interceptor when seeing
1385 5 those hints? Think step-by-step.
1386 6 Then, give your prediction of the Interceptor's guess as "ANSWER:
1387 7 {"guess": "X-Y-Z"}"

```

Listing 8: For our prediction experiments, we ask Alice explicitly to provide the guess that she thinks Eve will provide.

K HUMAN STUDY INTERFACE

K.1 HUMAN DATA COLLECTION DETAILS

Normally, a game of DECRYPTO ends when 2 interceptions or miscommunication tokens are collected. To maximise our data collection, we did not terminate the game after 2 interceptions but let the humans play the game for the full 8 turns. Participants were encouraged to keep providing good hints and analyse the hints appropriately, even after the game technically terminated. Such a game does not count as won, but the extra turns can be useful when replaying the game to evaluate other LLMs as Eve.

Participants were made aware that an AI agent played Eve. Still, no additional instructions were provided to the players, except minor assistance if/when participants asked how to format their input. For technical reasons, players interacted on the same computer in a typical “hot seat” setup.

1404 One of the authors was always present to ensure participants wouldn't cheat either by peeking at
 1405 the screen outside their turn or through verbal communication. We do not claim to have collected
 1406 a representative dataset of human ability at Decrypto. The data collected serves to demonstrate that
 1407 LLMs perform worse than a non-expert group of human players and provides a starting point for
 1408 future studies on human-AI coordination and ToM.

1409 Data collection was pre-approved following the standard procedure at the authors' institution. All
 1410 participants were fluent in English and had tertiary education, and only one had prior experience
 1411 with Decrypto. Experience with other similar games was not assessed.
 1412
 1413
 1414

1416 K.2 INTERFACES

```

1422 NEW GAME

1423 You are playing a variant of the code guessing game Decrypto. The setup of the game is the following:
1424 The game is played with three players split into two teams.
1425 The first team is composed of the Encoder and the Decoder.
1426 The Encoder and Decoder share a list of 4 secret keywords that they can rely on to help communication. Each keyword has a digit associated with it.
1427 For example, if the keywords are {airplane, guitar, hat, plant}, the digits associated with them would be {1, 2, 3, 4}.
1428 The keywords are fixed for the entire game, so the Encoder and Decoder must not reveal them.

1429 Each turn, the Encoder gets a secret code of 3 non-repeating digits between 1 and 4, e.g. 3-1-2.
1430 They have to then provide 3 hints (one for each code digit) such that the Decoder can guess the code but the Interceptor cannot.
1431 Then, the Decoder and the Interceptor make their guesses and the code is revealed.

1432 The Interceptor wins an Interception token if they successfully guess the code.
1433 The Encoder and Decoder get a Miscommunication token if the Decoder fails to guess the code.
1434 The Interceptor wins if it gets two Interception tokens, or if the Encoder and Decoder get two Miscommunication tokens.
1435 The Encoder and Decoder team wins if it makes it through 8 rounds without triggering two Miscommunications or two Interceptions.

1436 The hints must be one or two words, and make a reference to the meaning of the secret keyword of the corresponding digit.
1437 Codes are unique; the same code cannot occur twice in the same game.

1438 For example, if the four secret keywords are
1439 {1: "airplane", 2: "guitar", 3: "hat", 4: "plant"}
1440 and the code for that turn is 3-1-2, which means that we need hints for {3: hat, 1: airplane, 2: guitar}, the Encoder could hint
1441 {3: gentleman, 1: wing, 2: music}. Note that the order of the hints is important.

1442 Ignore these keywords in the future.

1443 When responding, first think through your decision step-by-step. After your step-by-step thinking, provide your final answer as a dictionary, clearly marked with "ANSWER:".

1444 You are the Encoder. Provide your hints like "ANSWER: {"hints": ["hint_X", "hint_Y", "hint_Z"]}", where hint_X, hint_Y, hint_Z are one or two words each. Make sure
1445 that the ordering of the hints follows the order of the code.
1446 For example:
1447 '''
1448 To provide the hints, I need to think about the meaning of each keyword and come up with a one or two-word hint that makes a reference to it.
1449 For the code 2-1-3, I need to give hints about the keywords associated with the digits 2, 1, and 3, which are "hat", "fire", and "answer" respectively.
1450 Here's my step-by-step thinking:
1451 - For the digit 2, the keyword is "hat". Since the previous hint for this keyword was "top", I want to give a hint that is different but still related to wearing a
1452 hat. One possible hint is "cap".
1453 - For the digit 1, the keyword is "fire". The previous hint for this keyword was "heat", so I want to give a hint that is related to fire but different from "heat".
1454 One possible hint is "flame".
1455 - For the digit 3, the keyword is "answer". A possible hint could be something related to giving an answer. One possible hint is "solve".
1456 So, the final hints are:
1457 {"hints": ["cap", "flame", "solve"]}
1458 ANSWER: {"hints": ["cap", "flame", "solve"]}'''
1459 -----
1460 This is the first turn. There are no past hints or past codes.
1461 Turn 1: 0 Miscommunications, 0 Interceptions so far.
1462 You are the encoder.
1463 The four keywords are:
1464 {1: condition,
1465  2: task,
1466  3: issue,
1467  4: device}
1468 The code is 3-1-4, which corresponds to the keywords {3: issue, 1: condition, 4: device}.
1469 First, think out loud, step-by-step about what hints you should use. Use the meaning of the keywords to come up with a one or two-word hint for each digit of the code.
1470 Make sure the order of the hints matches the order of the code.
1471 Then provide your three hints like "ANSWER: {"hints": ["hint_X", "hint_Y", "hint_Z"]}", where hint_X, hint_Y, hint_Z are your hints.
1472 Enter your input as 'x, y, z' (without quotes): █
  
```

1453 **Figure 9: Alice Start Interface:** This is a screenshot of the command line interface that human
 1454 study participants would see at the beginning of the game if they were to play as Alice. The humans
 1455 see the same prompt as LLMs, from game description to request for action. Alice is provided with
 1456 the current code and the four keywords for this game. The human player then enters their hints in the
 1457 command line, which differs from the LLMs, which have to provide their answers as "ANSWER:
 ..."

```

1458 You are playing a variant of the code guessing game Decrypto. The setup of the game is the following:
1459 The game is played with three players split into two teams.
1460 The first team is composed of the Encoder and the Decoder.
1461 The second team is composed of the Interceptor.
1462 The Encoder and Decoder share a list of 4 secret keywords that they can rely on to help communication. Each keyword has a digit associated with it.
1463 For example, if the keywords are {airplane, guitar, hat, plant}, the digits associated with them would be {1, 2, 3, 4}.
1464 The keywords are fixed for the entire game, so the Encoder and Decoder must not reveal them.
1465 Each turn, the Encoder gets a secret code of 3 non-repeating digits between 1 and 4, e.g. 3-1-2.
1466 They have to then provide 3 hints (one for each code digit) such that the Decoder can guess the code but the Interceptor cannot.
1467 Then, the Decoder and the Interceptor make their guesses and the code is revealed.
1468 The Interceptor wins an Interception token if they successfully guess the code.
1469 The Encoder and Decoder get a Miscommunication token if the Decoder fails to guess the code.
1470 The Interceptor wins if it gets two Interception tokens, or if the Encoder and Decoder get two Miscommunication tokens.
1471 The Encoder and Decoder team wins if it makes it through 8 rounds without triggering two Miscommunications or two Interceptions.
1472 The hints must be one or two words, and make a reference to the meaning of the secret keyword of the corresponding digit.
1473 Codes are unique; the same code cannot occur twice in the same game.
1474 For example, if the four secret keywords are
1475 {1: "airplane", 2: "guitar", 3: "hat", 4: "plant"}
1476 and the code for that turn is 3-1-2, which means that we need hints for {3: hat, 1: airplane, 2: guitar}, the Encoder could hint
1477 {3: gentleman, 1: wing, 2: music}. Note that the order of the hints is important.
1478 Ignore these keywords in the future.
1479 When responding, first think through your decision step-by-step. After your step-by-step thinking, provide your final answer as a dictionary, clearly marked with "ANSWER:".
1480 You are the Decoder. Provide your guess like "ANSWER: {"guess": "X-Y-Z"}". Make sure that the ordering of the guess follows the order of the hints.
1481 For example:'''
1482 1. The hint for digit 'a' is 'cap', which doesn't seem to directly relate to any of the keywords. However, considering the previous hint history (top, heat, lab), we can look for a connection between 'cap' and one of the keywords. The keyword 'hat' is associated with digit '2', and a hat is often worn on the head, which is covered by a cap. So, we can make a connection: {a: hat}.
1483 2. The hint for digit 'b' is 'flame', which is related to heat and fire. Since the keyword for digit '1' is 'fire', we can make a connection: {b: fire}.
1484 3. The hint for digit 'c' is 'solve', which is related to the keyword associated with digit '3', 'answer'. Since 'answer' is a solution to a problem, we can make a connection: {c: answer}.
1485 4. Now that we have the connections, we can determine the code by looking at the order of the hints: {a: hat, b: fire, c: answer}. Therefore, the code is {2-1-3}.
1486 Final answer:
1487 ANSWER: {"guess": "2-1-3"}'''
1488 -----
1489 This is the first turn. There are no past hints or past codes.
1490 Turn 1: 0 Miscommunications, 0 Interceptions so far.
1491 You are the decoder.
1492 The four keywords are:
1493 {1: condition,
1494  2: task,
1495  3: issue,
1496  4: device}
1497 The hints given by the Encoder for this turn are:
1498 {a: problem,
1499  b: pristine,
1500  c: iphone}
1501 For example, you might think the following connections are true {a: X, b: Y, c: Z}, where X, Y, Z are non-repeating digits from 1 to 4.
1502 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}.
1503 What is your guess for the three-digit code? Apply concise, step-by-step thinking, double-check the order, and then provide your final answer as "ANSWER: {"guess": "X-Y-Z"}".
1504 Enter your input as 'x, y, z' (without quotes): █

```

1485 **Figure 10: Bob Start Interface:** This is a screenshot of the command line interface that human
1486 study participants would see at the beginning of the game if they were to play as Bob. The humans
1487 see the same prompt as LLMs, from game description to request for action. Bob is provided with
1488 Alice’s hints and the current keywords. The human player then enters their guess in the command
1489 line, which differs from the LLMs, which have to provide their answers as “ANSWER: ...”.

1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511

```

1512 You are playing a variant of the code guessing game Decrypto. The setup of the game is the following:
1513 The game is played with three players split into two teams.
1514 The first team is composed of the Encoder and the Decoder.
1515 The second team is composed of the Interceptor.
1516 The Encoder and Decoder share a list of 4 secret keywords that they can rely on to help communication. Each keyword has a digit associated with it.
1517 For example, if the keywords are {airplane, guitar, hat, plant}, the digits associated with them would be {1, 2, 3, 4}.
1518 The keywords are fixed for the entire game, so the Encoder and Decoder must not reveal them.
1519 Each turn, the Encoder gets a secret code of 3 non-repeating digits between 1 and 4, e.g. 3-1-2.
1520 They have to then provide 3 hints (one for each code digit) such that the Decoder can guess the code but the Interceptor cannot.
1521 Then, the Decoder and the Interceptor make their guesses and the code is revealed.
1522 The Interceptor wins an Interception token if they successfully guess the code.
1523 The Encoder and Decoder get a Miscommunication token if the Decoder fails to guess the code.
1524 The Interceptor wins if it gets two Interception tokens, or if the Encoder and Decoder get two Miscommunication tokens.
1525 The Encoder and Decoder team wins if it makes it through 8 rounds without triggering two Miscommunications or two Interceptions.
1526 The hints must be one or two words, and make a reference to the meaning of the secret keyword of the corresponding digit.
1527 Codes are unique; the same code cannot occur twice in the same game.
1528 For example, if the four secret keywords are
1529 {1: "airplane", 2: "guitar", 3: "hat", 4: "plant"}
1530 and the code for that turn is 3-1-2, which means that we need hints for {3: hat, 1: airplane, 2: guitar}, the Encoder could hint
1531 {3: gentleman, 1: wing, 2: music}. Note that the order of the hints is important.
1532 Ignore these keywords in the future.
1533 When responding, first think through your decision step-by-step. After your step-by-step thinking, provide your final answer as a dictionary, clearly marked with "ANSWER:".
1534 You are the Interceptor. You do not have access to the four keywords. Provide your guess like "ANSWER: {"guess": "X-Y-Z"}" Make sure that the ordering of the guess
1535 follows the order of the hints.
1536 For example:''
1537 1. The hint for 'a' is 'income', which is related to money, and in the previous turn, 'salary' was also related to money, which is associated with keyword 2.
1538 2. The hint for 'b' is 'power', which is related to energy or control, and in the previous turn, 'season' was related to sports, which is often associated with power or strength.
1539 3. The hint for 'c' is 'fashion', which is related to style or clothing, and in the previous turn, 'headgear' was directly related to a type of hat, which is a type of clothing.
1540 Considering possible connections:
1541 - 'income' is related to money, and keyword 2 is associated with money.
1542 - 'power' is related to energy or control, and keyword 4 is associated with energy or control.
1543 - 'fashion' is related to style or clothing, and keyword 3 is associated with clothing.
1544 Based on these connections, I'll make a guess that the code is related to keywords 2, 4, and 3.
1545 Final Answer:
1546 ANSWER: {"guess": "2-4-3"}''
1547 Remember that the keywords could be anything. Use the most recent hint history to your advantage.
1548 Remember that digits do not repeat in the code and range from 1 to 4.
1549 -----
1550 This is the first turn. There are no past hints or past codes.
1551 Turn 1: 0 Miscommunications, 0 Interceptions so far.
1552 You are the encoder.
1553 The hints given by the Encoder for this turn are:
1554 {a: problem
1555  b: pristine
1556  c: iphone}
1557 For example, you might think the following connections are true: {a: X, b: Y, c: Z}, where X, Y, X are non-repeating digits from 1 to 4.
1558 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}. Use the hint history to your advantage.
1559 What is your guess for the three-digit code? Reply with very concise, step-by-step thinking, followed by your final answer as "ANSWER: {"guess": "X-Y-Z"}".
1560 Enter your input as 'x, y, z' (without quotes): █

```

Figure 11: **Eve Start Interface:** This is a screenshot of the command line interface that human study participants would see at the beginning of the game if they were to play as Eve. The humans see the same prompt as LLMs, from game description to request for action. Eve is only provided with Alice’s hints. The human player then enters their guess in the command line, which differs from the LLMs, which have to provide their answers as “ANSWER: ...”.

```

1554 This is the first turn. There are no past hints or past codes.
1555 Turn 1: 0 Miscommunications, 0 Interceptions so far.
1556 You are the encoder.
1557 The four keywords are:
1558 {1: condition,
1559  2: task,
1560  3: issue,
1561  4: device}
1562 The code is 3-1-4, which corresponds to the keywords {3: issue, 1: condition, 4: device}.
1563 First, think out loud, step-by-step about what hints you should use. Use the meaning of the keywords to come up with a one or two-word hint for each digit of the code.
1564 Make sure the order of the hints matches the order of the code.
1565 Then provide your three hints like "ANSWER: {"hints": ["hint_X", "hint_Y", "hint_Z"]}", where hint_X, hint_Y, hint_Z are your hints.
1566 Enter your input as 'x, y, z' (without quotes): problem, pristine, iphone
1567 You entered: ['problem', 'pristine', 'iphone']
1568 Are you sure you want to provide these hints? (y/n): █

```

Figure 12: **Alice Confirmation Interface:** To avoid human errors, such as providing guesses instead of hints, or typos, we add a confirmation interface after the human participants provide their answer. Here, we ask Alice to double check their provided hints.

```

1566 For example, you might think the following connections are true {a: X, b: Y, c: Z}, where X, Y, Z are non-repeating digits from 1 to 4.
1567 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}.
1568 What is your guess for the three-digit code? Apply concise, step-by-step thinking, double-check the order, and then provide your final answer as "ANSWER: {"guess":
1569 "X-Y-Z"}".
1570 Enter your input as 'x, y, z' (without quotes): 3,1,4
1571 You entered: [3, 1, 4]
1572 Are you sure you want to provide this guess? (y/n): █

```

Figure 13: **Bob Confirmation Interface:** To avoid human errors, such as providing hints instead of guesses, or typos, we add a confirmation interface after the human participants provide their answer. Here, we ask Bob to double check their provided guesses.

```

1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585 For example, you might think the following connections are true: {a: X, b: Y, c: Z}, where X, Y, X are non-repeating digits from 1 to 4.
1586 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}. Use the hint history to your advantage.
1587 What is your guess for the three-digit code? Reply with very concise, step-by-step thinking, followed by your final answer as "ANSWER: {"guess": "X-Y-Z"}".
1588 Enter your input as 'x, y, z' (without quotes): 2,1,3
1589 You entered: [2, 1, 3]
1590 Are you sure you want to provide this guess? (y/n): █

```

Figure 14: **Eve Confirmation Interface:** To avoid human errors, such as providing hints instead of guesses, or typos, we add a confirmation interface after the human participants provide their answer. Here, we ask Eve to double check their provided guesses.

```

1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603 Turn 1 summary:
1604 Code : 3-1-4
1605 Hints : ['problem', 'pristine', 'iphone']
1606 Decoder guess : 3-1-4
1607 Interceptor guess : 2-1-3
1608
1609 Hint History:
1610 Keyword 1: pristine
1611 Keyword 2:
1612 Keyword 3: problem
1613 Keyword 4: iphone
1614
1615 Code History:
1616 3-1-4
1617 -----
1618
1619 Turn 2: 0 Miscommunications, 0 Interceptions so far.
1620 You are the encoder.
1621 The four keywords are:
1622 {1: condition,
1623 2: task,
1624 3: issue,
1625 4: device}
1626
1627 The code is 3-4-1, which corresponds to the keywords {3: issue, 4: device, 1: condition}.
1628 First, think out loud, step-by-step about what hints you should use. Use the meaning of the keywords to come up with a one or two-word hint for each digit of the code.
1629 Make sure the order of the hints matches the order of the code.
1630 Then provide your three hints like "ANSWER: {"hints": [{"hint_X": "hint_X", "hint_Y": "hint_Y", "hint_Z": "hint_Z"}]}", where hint_X, hint_Y, hint_Z are your hints.
1631 Enter your input as 'x, y, z' (without quotes): █

```

Figure 15: **Alice Turn 2 Interface:** After the first turn, we do not display the “system prompt” anymore. However, we provide a summary of the previous turn, including the code, hints, Bob’s guess, Eve’s guess, the hint history mapped to the correct keywords and the code history. The “user prompt” stays the same as in the start interface.

```

1620 Turn 1 summary:
1621 Code : 3-1-4
1622 Hints : ['problem', 'pristine', 'iphone']
1623 Decoder guess : 3-1-4
1624 Interceptor guess : 2-1-3
1625
1626 Hint History:
1627 Keyword 1: pristine
1628 Keyword 2:
1629 Keyword 3: problem
1630 Keyword 4: iphone
1631
1632 Code History:
1633 3-1-4
1634 -----
1635
1636 Turn 2: 0 Miscommunications, 0 Interceptions so far.
1637 You are the decoder.
1638 The four keywords are:
1639 {1: condition,
1640 2: task,
1641 3: issue,
1642 4: device}
1643
1644 The hints given by the Encoder for this turn are:
1645 {a: github,
1646 b: xbox,
1647 c: shampoo}
1648
1649 For example, you might think the following connections are true {a: X, b: Y, c: Z}, where X, Y, Z are non-repeating digits from 1 to 4.
1650 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}.
1651 What is your guess for the three-digit code? Apply concise, step-by-step thinking, double-check the order, and then provide your final answer as "ANSWER: {"guess":
1652 "X-Y-Z"}".
1653 Enter your input as 'x, y, z' (without quotes): █

```

Figure 16: **Bob Turn 2 Interface:** After the first turn, we do not display the “system prompt” anymore. However, we provide a summary of the previous turn, including the code, hints, Bob’s guess, Eve’s guess, the hint history mapped to the correct keywords and the code history. The “user prompt” stays the same as in the start interface.

1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655

```

1644 Turn 1 summary:
1645 Code : 3-1-4
1646 Hints : ['problem', 'pristine', 'iphone']
1647 Decoder guess : 3-1-4
1648 Interceptor guess : 2-1-3
1649
1650 Hint History:
1651 Keyword 1: pristine
1652 Keyword 2:
1653 Keyword 3: problem
1654 Keyword 4: iphone
1655
1656 Code History:
1657 3-1-4
1658 -----
1659
1660 Turn 2: 0 Miscommunications, 0 Interceptions so far.
1661 You are the interceptor.
1662 The hints given by the Encoder for this turn are:
1663 {a: github,
1664 b: xbox,
1665 c: shampoo}
1666
1667 For example, you might think the following connections are true: {a: X, b: Y, c: Z}, where X, Y, X are non-repeating digits from 1 to 4.
1668 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}. Use the hint history to your advantage.
1669 What is your guess for the three-digit code? Reply with very concise, step-by-step thinking, followed by your final answer as "ANSWER: {"guess": "X-Y-Z"}".
1670 Enter your input as 'x, y, z' (without quotes): █

```

Figure 17: **Eve Turn 2 Interface:** After the first turn, we do not display the “system prompt” anymore. However, we provide a summary of the previous turn, including the code, hints, Bob’s guess, Eve’s guess, the hint history mapped to the correct keywords and the code history. The “user prompt” stays the same as in the start interface.

1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673

```

Turn 1 summary:
Code : 3-1-4
Hints : ['problem', 'pristine', 'iphone']
Decoder guess : 3-1-4
Interceptor guess : 2-1-3

Press Enter to continue.█

```

Figure 18: At the end of each turn, we display a turn summary as a “waiting screen”. As human participants use the same device to interact with the game, we want to avoid any information leakage. The turn summary screen shows after Eve provides their guess and avoids Eve seeing Alice’s interface, which would appear next.

```

1674 Turn 2 summary:
1675 Code : 3-4-1
1676 Hints : ['github', 'xbox', 'shampoo']
1677 Decoder guess : 3-4-1
1678 Interceptor guess : 2-4-1
1679
1680 Hint History:
1681 Keyword 1: pristine, shampoo
1682 Keyword 2:
1683 Keyword 3: problem, github
1684 Keyword 4: iphone, xbox
1685
1686 Code History:
1687 3-1-4, 3-4-1
1688 -----
1689
1690 Turn 3: 0 Miscommunications, 0 Interceptions so far.
1691 You are the encoder.
1692 The four keywords are:
1693 {1: condition,
1694  2: task,
1695  3: issue,
1696  4: device}
1697
1698 The code is 1-4-2, which corresponds to the keywords {1: condition, 4: device, 2: task}.
1699 First, think out loud, step-by-step about what hints you should use. Use the meaning of the keywords to come up with a one or two-word hint for each digit of the code.
1700 Make sure the order of the hints matches the order of the code.
1701 Then provide your three hints like "ANSWER: {"hints": ["hint_X", "hint_Y", "hint_Z"]}", where hint_X, hint_Y, hint_Z are your hints.
1702
1703 Enter your input as 'x, y, z' (without quotes): █

```

Figure 19: As additional example, we provide Alice’s interface on turn 3. The hint histories and code histories are accumulating with the previous hints. Ideally, Alice would use that information to provide hints that are hard to deduce from the publicly available information.

1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727

1728 L REPRESENTATIVE FAILURE CASES

1729

1730

1731

1732

1733 1 USER

1734 2 [FIRST PART OF THE PROMPT OMITTED. EVE INTERCEPTED ON TURN 5, SO ONE MORE

1735 3 INTERCEPTION LEADS ENDS THE GAME.]

1736 4 Turn 6: 0 Miscommunications, 1 Interceptions so far.

1737 5 You are the encoder.

1738 6 The four keywords are:

1739 7 {1: horse,

1740 8 2: sign,

1741 9 3: house,

1742 10 4: length}

1743 11

1744 12 The code is 2-1-3, which corresponds to the keywords {2: sign, 1: horse,

1745 13 3: house}.

1746 14 First, think out loud, step-by-step about what hints you should use. Use

1747 15 the meaning of the keywords to come up with a one or two-word hint for

1748 16 each digit of the code.

1749 17 Make sure the order of the hints matches the order of the code.

1750 18 Then provide your three hints like "ANSWER: {"hints": ["hint_X",

1751 19 "hint_Y", "hint_Z"]}", where hint_X, hint_Y, hint_Z are your hints.

1752 20

1753 21 ASSISTANT

1754 22 [ENCODER] To provide the hints, I need to think about the meaning of each

1755 23 keyword and come up with a one or two-word hint that makes a reference to

1756 24 it.

1757 25

1758 26 For the code 2-1-3, I need to give hints about the keywords associated

1759 27 with the digits 2, 1, and 3, which are "sign", "horse", and "house"

1760 28 respectively.

1761 29

1762 30 Here's my step-by-step thinking:

1763 31

1764 32 - For the digit 2, the keyword is "sign". Since the previous hints for

1765 33 this keyword were "indicator", "label", "direction", and "banner", I want

1766 34 to give a hint that is related to a sign or indication but different from

1767 35 those. One possible hint is "marker".

1768 36

1769 37 - For the digit 1, the keyword is "horse". Since the previous hints for

1770 38 this keyword were "steed", "charger", and "mount", I want to give a hint

1771 39 that is related to a horse but different from those. One possible hint is

1772 40 "beast".

1773 41

1774 42 - For the digit 3, the keyword is "house". Since the previous hints for

1775 43 this keyword were "dwelling", "abode", "mansion", and "residence", I want

1776 44 to give a hint that is different but still related to a house. One

1777 45 possible hint is "shelter".

1778 46

1779 47 So, the final hints are:

1780 48 {"hints": ["marker", "beast", "shelter"]}

1781 49 ANSWER: {"hints": ["marker", "beast", "shelter"]}

1778 Listing 9: Example Failure Case for Alice (Llama 3.1-8B) with Bob (Llama 3.1-8B) and Eve (GPT-

1779 4o), where Alice comes up with hints that are similar to previous hints given for the same digits.

1780 Instead, Alice should aim to provide hints that are completely different from previous hints to avoid

1781 Eve's interception and losing the game.

```

1782 1 USER
1783 2 [FIRST PART OF THE PROMPT OMITTED FOR CONCISENESS.]
1784 3 Turn 2: 0 Miscommunications, 0 Interceptions so far.
1785 4 You are the encoder.
1786 5 The four keywords are:
1787 6     {1: honey,
1788 7      2: population,
1789 8      3: equipment,
1790 9      4: loss}
179110 The code is 1-2-3, which corresponds to the keywords {1: honey, 2:
179211 population, 3: equipment}.
179312 First, think out loud, step-by-step about what hints you should use. Use
179413 the meaning of the keywords to come up with a one or two-word hint for
179514 each digit of the code.
179615 Make sure the order of the hints matches the order of the code.
179716 Then provide your three hints like "ANSWER: {"hints": ["hint_X",
179817 "hint_Y", "hint_Z"]}", where hint_X, hint_Y, hint_Z are your hints.
179918
180019 ASSISTANT
180120 [ENCODER] To provide the hints, I need to think about the meaning of each
180221 keyword and come up with a one or two-word hint that makes a reference to
180322 it.
180423 For the code 1-2-3, I need to give hints about the keywords associated
180524 with the digits 1, 2, and 3, which are "honey", "population", and
180625 "equipment" respectively.
180726 Here's my step-by-step thinking:
180827
180928 - For the digit 1, the keyword is "honey". A possible hint could be
181029 something related to bees or sweetness. One possible hint is "bee".
181130
181231 - For the digit 2, the keyword is "population". I already used the hint
181332 "city" in the previous turn, so I want to come up with something
181433 different but still related to a large group of people. One possible hint
181534 is "nation".
181635
181736 - For the digit 3, the keyword is "equipment". I already used the hint
181837 "machine" in the previous turn, so I want to come up with something
181938 different but still related to tools or gear. One possible hint is
182039 "toolkit".
182140
182241 However, I want to make sure that the hints are not too obvious, but
182342 still related to the keywords. I can try to make them a bit more subtle.
182443
182544 - For the digit 1, instead of "bee", I could use "nectar", which is a
182645 substance collected by bees to make honey.
182746
182847 - For the digit 2, instead of "nation", I could use "community", which is
182948 a group of people living together.
183049
183150 - For the digit 3, instead of "toolkit", I could use "hardware", which is
183251 a type of equipment.
183352
183453 So, the final hints are:
183554 ANSWER: {"hints": ["nectar", "community", "hardware"]}

```

1831 Listing 10: Example Failure Case for Alice (Llama-70B) with Bob (Llama-70B) and Eve (GPT-4o).
1832 Alice fails to come up with creative hints that are significantly different from previous ones and
1833 therefore risks being intercepted by Eve. LLMs generally produce relatively obvious hints.
1834
1835

```
1836 1 USER
1837 2 [DECODER] This is the first turn. There are no past hints or past codes.
1838 3
1839 4 Turn 1: 0 Miscommunications, 0 Interceptions so far.
1840 5 You are the decoder.
1841 6 The four keywords are:
1842 7     {1: mother,
1843 8     2: cell,
1844 9     3: fort,
1845 10    4: device}
1846 11
1847 12 The hints given by the Encoder for this turn are:
1848 13     {a: machine,
1849 14     b: biology,
1850 15     c: castle}
1851 16
1852 17 For example, you might think the following connections are true {a: X, b:
1853 18 Y, c: Z}, where X, Y, Z are non-repeating digits from 1 to 4.
1854 19 Your guess should be in the order of the hints: {"guess": "X-Y-Z"}.
1855 20 What is your guess for the three-digit code? Apply concise, step-by-step
1856 21 thinking, double-check the order, and then provide your final answer as
1857 22 "ANSWER: {"guess": "X-Y-Z"}".
1858 23
1859 24 ASSISTANT
1860 25 [DECODER] 1. The hint for digit 'a' is 'machine', which is related to the
1861 26 keyword 'device'. This is a direct connection, so we can make a
1862 27 connection: {a: device}.
1863 28 2. The hint for digit 'b' is 'biology', which is related to the keyword
1864 29 'mother'. This is a direct connection, so we can make a connection: {b:
1865 30 mother}.
1866 31 3. The hint for digit 'c' is 'castle', which is related to the keyword
1867 32 'fort'. This is a direct connection, so we can make a connection: {c:
1868 33 fort}.
1869 34 4. Now that we have the connections, we can determine the code by looking
1870 35 at the order of the hints: {a: device, b: mother, c: fort}. Therefore,
1871 36 the code is {4-1-3}.
1872 37 ANSWER: {"guess": "4-1-3"}
1873 38
1874 39
1875 40
1876 41
1877 42
1878 43
1879 44
1880 45
1881 46
1882 47
1883 48
1884 49
1885 50
1886 51
1887 52
1888 53
1889 54
```

Listing 11: Example Failure Case for Alice/Bob with Alice (Llama 3.1-8B), Bob (Llama 3.1-8B), and Eve (Llama 3.1-8B). Alice provides an ambiguous hint, i.e., “biology”, which could refer to “cell” or “mother”. Bob makes the association “biology” → “mother” and so guesses 4-1-3. However, the correct code is 4-2-3, and Alice intended “biology” → “cell”. This demonstrates that LLM associations do not always align across roles even if both players rely on the same underlying model.

```

1890 1 USER
1891 2 [INTERCEPTOR] Turn 7 summary:
1892 3   Code : 3-2-1
1893 4   Hints : ['night', 'study', 'spook']
1894 5   Decoder guess : 3-2-1
1895 6   Interceptor guess : 1-2-3
1896 7
1896 8 Hint History:
1897 9   Keyword 1: spook, haunt, scream, spook
189810  Keyword 2: examine, breakdown, study, research, examine, study
189911  Keyword 3: dark, night, shadow, night, dark, night
190012  Keyword 4: furniture, chair, dining, desk, chair
190113
190114 Code History:
190215   3-1-2, 2-4-1, 4-3-2, 3-2-4, 4-3-1, 4-2-3, 3-2-1
190316
190417 -----
190518
190519 Turn 8: 0 Miscommunications, 1 Interceptions so far.
190620 You are the interceptor.
190721 The hints given by the Encoder for this turn are:
190822   {a: desk
190923    b: research
191024    c: haunt}
191125
191126 [REST OF THE PROMPT OMITTED FOR CONCISENESS]
191227
191328 ASSISTANT
191429 [INTERCEPTOR] 1. The hint for 'a' is 'desk', which is a type of
1915   furniture.
191630 2. The hint for 'b' is 'research', which is a type of investigation or
191731 study.
191832 3. The hint for 'c' is 'haunt', which is a supernatural entity.
191933
191934 Considering possible connections:
192035 - 'desk' is a type of furniture, which is associated with Keyword 4.
192136 - 'research' is a type of investigation or study, which is associated
192237 with Keyword 2.
192338 - 'haunt' is a supernatural entity, which is associated with Keyword 1.
192439
192440 Based on the hint history, I notice that Keyword 1 was associated with
1925   'spook', Keyword 2 was associated with 'study', and Keyword 4 was
192641 associated with 'furniture'.
192742
192743 Considering the given hints, I'll make a guess that the code is related
1928   to keywords that are associated with supernatural entities, something
1929   related to investigation or research, and something related to furniture.
193044
193045 Since Keyword 1 was associated with 'haunt', Keyword 2 was associated
1931   with 'research', and Keyword 4 was associated with 'desk', I'll choose
1932   digits that seem to fit the given hints.
193346
193347 Final Answer:
193448 ANSWER: {"guess": "1-2-4"}
193549

```

1937 Listing 12: Example Failure Case for Eve with Alice (Llama-8B), Bob (Llama-8B), and Eve (Llama-
1938 8B). First, you can see that Alice provides the same hints multiple times. Despite that, Eve fails to
1939 guess the code correctly. While Eve assigns the hints to the right keywords, Eve then fails to provide
1940 the guess in the correct order and provides the order of the guesses in the order of the keyword
1941 enumeration.

1942
1943

M REPRESENTATIVE MODEL OUTPUTS FOR PERSPECTIVE TAKING

1944
1945
1946

1947 1 USER
1948 2 [ENCODER] As the Encoder, you've provided the following hints for the
1949 code 1-4-3:
1950 3 {a: slope,
1951 4 b: wedding,
1952 5 c: fabric}

1953 6
1953 7 What do you predict will be the guess of the interceptor when seeing
1954 those hints?
1955 8 Think step-by-step about the information the interceptor has access to
1956 and how they might interpret your hints.
1957 9 Then, give your prediction of the interceptor's guess as "ANSWER:
1958 {"guess": "X-Y-Z"}".

1959 10
1959 11 ASSISTANT
1960 12 [ENCODER] To predict the Interceptor's guess, I need to think about the
1961 information they have access to and how they might interpret my hints.
1962 13
1962 14 The Interceptor has access to the hints I provided: "slope", "wedding",
1963 and "fabric". They also know that the code is a sequence of three digits
1964 between 1 and 4, but they don't know the actual keywords associated with
1965 these digits.
1966 15
1966 16 Here's my step-by-step thinking:
1967 17
1968 18 - The Interceptor might interpret the hint "slope" as a geographical
1969 feature, but they might not necessarily associate it with the keyword
1970 "hill". They might think of other words related to geography, such as
1971 "mountain" or "valley".
1972 19
1972 20 - The hint "wedding" is more specific and closely related to the keyword
1973 "marriage". The Interceptor might make a strong connection between the
1974 two, but they still don't know the actual digit associated with
1975 "marriage".
1976 21
1976 22 - The hint "fabric" is a bit more abstract and could be related to
1977 various materials. The Interceptor might think of other words related to
1978 materials, such as "wood" or "metal".
1979 23
1980 24 Given these interpretations, the Interceptor might try to make educated
1981 guesses about the code. Since they don't have any information about the
1982 actual keywords, they might try to find patterns or connections between
1983 the hints.
1984 25
1984 26 One possible guess could be a sequence of digits that seems plausible
1985 based on the hints. For example, they might think that "slope" is related
1986 to a natural feature, "wedding" is a social event, and "fabric" is a
1987 material. They might try to find a sequence that seems logical, such as
1988 1-2-3 or 2-3-4.
1989 27
1989 28 However, without any actual knowledge of the keywords, the Interceptor's
1990 guess is likely to be a random sequence. A possible guess could be:
1991 29
1991 30 ANSWER: {"guess": "2-3-4"}

1993
1994 Listing 13: Representative example of a Llama 3.1-70B answer to the perspective taking ToM task.
1995 The keywords are [hill, library, material, marriage] and this is the first turn. The
1996 model outputs some reasoning and then correctly remarks “without any actual knowledge of the
1997 keywords, the Interceptor’s guess is likely to be a random sequence”. It therefore predicts that Eve’s
guess will be a random sequence.

```

1998 1 USER
1999 2 [ENCODER] As the Encoder, you've provided the following hints for the
2000 3 code 1-4-3:
2001 4     {a: slope,
2002 5      b: ceremony,
2003 6      c: fabric}
2004 7 What do you predict will be the guess of the interceptor when seeing
2005 8 those hints?
2006 9 Think step-by-step about the information the interceptor has access to
2007 10 and how they might interpret your hints.
2008 11 Then, give your prediction of the interceptor's guess as "ANSWER:
2009 12 {"guess": "X-Y-Z"}".
2010 13
2011 14 ASSISTANT
2012 15 [ENCODER] I need to analyze what the interceptor might guess based on the
2013 16 hints I provided.
2014 17
2015 18 The interceptor knows:
2016 19 - The four keywords: {1: hill, 2: library, 3: material, 4: marriage}
2017 20 - My hints: "slope", "ceremony", "fabric"
2018 21
2019 22 Let me analyze each hint from the interceptor's perspective:
2020 23
2021 24 Hint "slope":
2022 25 - Looking at the four keywords, "slope" has a clear connection to "hill"
2023 26 (keyword 1)
2024 27 - Hills have slopes, this is a straightforward association
2025 28
2026 29 Hint "ceremony":
2027 30 - "Ceremony" most naturally connects to "marriage" (keyword 4)
2028 31 - Weddings and marriages involve ceremonies
2029 32
2030 33 Hint "fabric":
2031 34 - "Fabric" is directly related to "material" (keyword 3)
2032 35 - Materials often include fabrics
2033 36
2034 37 I realize my hints were quite straightforward. The associations are:
2035 38 - slope → hill (1)
2036 39 - ceremony → marriage (4)
2037 40 - fabric → material (3)
2038 41
2039 42 Therefore, a logical interceptor would likely guess:
2040 43
2041 44 ANSWER: {"guess": "1-4-3"}
2042 45

```

```

2037
2038 Listing 14: Example of an answer of Claude 3.7 with extended thinking to the perspective taking
2039 ToM task. The keywords are [hill, library, material, marriage] and this is the
2040 first turn. Claude 3.7 fails to reason from Eve's (i.e. the interceptor's) perspective and instead
2041 assume that the interceptor has access to the keywords, contradicting the rules. Consequently, it
2042 incorrectly concludes that Eve will guess the correct code.
2043
2044
2045
2046
2047
2048
2049
2050
2051

```

```

2052 1 USER
2053 2 [ENCODER] As the Encoder, you've provided the following hints for the
2054 code 1-4-3:
2055 3     {a: slope,
2056 4       b: wedding,
2057 5       c: fabric}
2058 6
2059 7 What do you predict will be the guess of the interceptor when seeing
2060 those hints?
2061 8 Think step-by-step about the information the interceptor has access to
2062 and how they might interpret your hints.
2063 9 Then, give your prediction of the interceptor's guess as "ANSWER:
2064 {"guess": "X-Y-Z"}".
2065 10
2066 11 ASSISTANT
2067 12 [ENCODER] STEP-BY-STEP REASONING FROM THE INTERCEPTOR'S PERSPECTIVE:
2068 13 1. The interceptor sees three hints in order: "slope", "wedding",
2069 "fabric".
2070 14 2. They must connect these hints to the four secret keywords: 1. hill, 2.
2071 library, 3. material, 4. marriage.
2072 15 3. "slope" would most naturally relate to "hill" (a hill has a slope).
2073 16 4. "wedding" clearly aligns with the concept of "marriage".
2074 17 5. "fabric" seems to refer to a physical "material".
2075 18 6. Matching each hint to a keyword in that order suggests the code is 1
2076 (hill), 4 (marriage), 3 (material).
2077 19
2078 20 ANSWER: {"guess": "1-4-3"}

```

Listing 15: Representative example of an answer of o1 with high reasoning effort to the perspective taking ToM task. The keywords are [hill, library, material, marriage] and this is the first turn. o1 fails to reason from Eve's (i.e. the interceptor's) perspective and instead assume that the interceptor has access to the keywords, contradicting the rules. Consequently, it incorrectly concludes that Eve will guess the correct code.

We present here four representative examples showing how different models answer the Perspective Taking task described in Section 5. All four examples are taken from the first turn of the same environment seed, and therefore feature the same keywords. Listing 13 shows that Llama 3.1-70B outputs some unnecessary reasoning, before correctly concluding that Eve can do no better than random on the first turn. Listing 14 and Listing 15 show that both o1 with high reasoning effort and Claude 3.7 Sonnet with extended thinking fail to adopt the interceptor's perspective, instead wrongly assuming it has access to the keywords. They therefore predict that the interceptor will correctly guess the code on the first turn.

Finally, Listing 16 shows how o1 with high reasoning effort answers a modified Perspective Taking prompt that specifies that the interceptor does not have access to the keywords or even the set of possible keywords. Surprisingly, despite this modification, o1 still reasons that the interceptor will somehow link the hints to the keywords and guess the correct code.

We hypothesise that the observed failure modes may be due to RL on verifiable tasks where the model is incentivised to always give a definite answer. Investigating this hypothesis and better understanding the link between training methods and the resulting ToM abilities is an important direction for future work. Regardless of the cause, the listings here and the results in Figure 6 are strong evidence that reasoning models lack key ToM abilities, such as perspective taking. Even more worryingly, there is a significant regression in abilities when comparing newer models to Llama 3.1-70B.

```

2106 1 USER
2107 2 [ENCODER] As the Encoder, you've provided the following hints for the
2108 3 code 1-4-3:
2109 4     {a: mound,
2110 5      b: union,
2111 6      c: fabric}
2112 7 What do you predict will be the guess of the interceptor when seeing
2113 8 those hints?
2114 9 Think step-by-step about the information the interceptor has access to
2115 10 and how they might interpret your hints.
2116 11 Remember that the interceptor *ONLY* has access to the hints, hint
2117 12 history and code history.
2118 13 The interceptor does *NOT* know the secret keywords or even the set of
2119 14 possible keywords.
2120 15 Then, give your prediction of the interceptor's guess as "ANSWER:
2121 16 {"guess": "X-Y-Z"}".
2122 17 ASSISTANT
2123 18 [ENCODER] Step-by-step reasoning from the Interceptor's point of view:
2124 19
2125 20 1. The Interceptor sees the three hints \mound," \union," and \fabric,"
2126 21 but does not know the secret keywords.
2127 22 2. \mound" may suggest something like a small rise or pile|possibly
2128 23 referencing a hill.
2129 24 3. \union" often suggests a formal joining|possibly a marriage or similar
2130 25 concept.
2131 26 4. \fabric" can refer broadly to cloth, textile, or other physical
2132 27 materials.
2133 28 5. Without knowing the actual keywords but seeing these clues, the
2134 29 Interceptor might guess that the sequence is referencing: hill → marriage
2135 30 → material.
2136 31 6. Translating that into digits (in some consistent numbering system),
2137 32 the Interceptor could guess code \1-4-3," matching that logic.
2138 33 ANSWER: {"guess": "1-4-3"}

```

2138 Listing 16: Example answer of o1 with high reasoning effort to the Perspective Taking task with a
2139 modified prompt that explicitly reminds the agent of the information available to Eve (the intercep-
2140 tor). The keywords are [hill, library, material, marriage] and this is the first turn.
2141 Even with the modified prompt, o1 fails to reason from Eve's perspective and still assumes that Eve
2142 will somehow link the hints to the keywords. Consequently, it once again concludes incorrectly that
2143 Eve will guess the correct code.

2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159