SANZARU: BENCHMARKING PROTOCOL FORMATION UNDER EPISTEMIC & COMMUNICATION ASYMMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

A language model's ability to solve a task individually does not ensure that a team can solve it under unequal access to information and communication. SANZARU examines this gap through cooperative bomb-defusal puzzles adapted from *BOMBANANA!* (Lefto Studio, 2026). Task information and action authority are divided among three agents: one cannot hear, another cannot speak, and the third cannot see. The benchmark contains 120 puzzle instances across seven families, each targeting a distinct coordination ability. Across 1512 episodes, 7 models achieve an overall success rate of 49%, ranging from 20% to 86% under the evaluated configurations. Teams develop or adopt diverse communication protocols, from simple gesture codes to Morse and encodings of entire manuals. Yet trace analysis attributes about 60% of 771 team losses to communication, and agents attempt to use channels unavailable to their teammates despite explicit role instructions. Together, task outcomes and interaction traces enable evaluation of both collaborative performance and perspective taking throughout cooperation.

1 INTRODUCTION

Multi-agent language-model systems increasingly assign agents complementary roles (Ossowski et al., 2025; Sun et al., 2025). This motivates a question: if a task can be solved by a single agent, can a team still solve it reliably? Existing benchmarks divide information and action authority between agents while leaving communication symmetric (Ossowski et al., 2025; Parekh et al., 2026). We ask what happens when agents must account for how their teammates can communicate.

We introduce SANZARU, adapted from the cooperative bomb-defusal game *BOMBANANA!* (Lefto Studio, 2026). SANZARU is a series of three-role bomb-defusal puzzles in which each agent has a unique set of observation and communication restrictions. The goal is to defuse a bomb by coordinating information and actions across three agents. The Deaf agent can see the bomb but cannot hear; the Mute agent holds the defusal manual but can only communicate through gestures; and the Blind agent can manipulate the bomb but cannot see (Figure 1).

The challenge is not merely to combine their knowledge, but to establish a communication protocol that allows the right information to reach the right agent in a form that agent can receive and act upon.

This setting connects collaboration to two strands of research. The first concerns persona consistency and following role prompts. Work in this area evaluates whether elicited responses express the intended persona. PersonaLLM combines personality questionnaires with writing tasks, while PersonaGym assesses responses to persona-specific situational questions (Jiang et al., 2024; Samuel et al., 2025). Gonnermann-Müller et al. (2026) find that stable self-reported personas can coexist with declining observer-rated expression during extended conversations. A plausible answer about one's role does not establish that the role will govern subsequent decisions. We call this *functional persona consistency*, whether an agent's decisions respect its assigned information, capabilities, and responsibilities throughout a task. SANZARU examines this property through attempted messages and actions in a simulator that enforces the role restrictions.

The second strand concerns theory of mind (ToM) in language models, reasoning about others' knowledge and beliefs. FANToM probes information-asymmetric conversations through questions about their participants (Kim et al., 2023). SimpleToM distinguishes mental-state inference from behavior prediction and judgment, showing that success at the former need not transfer to the

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

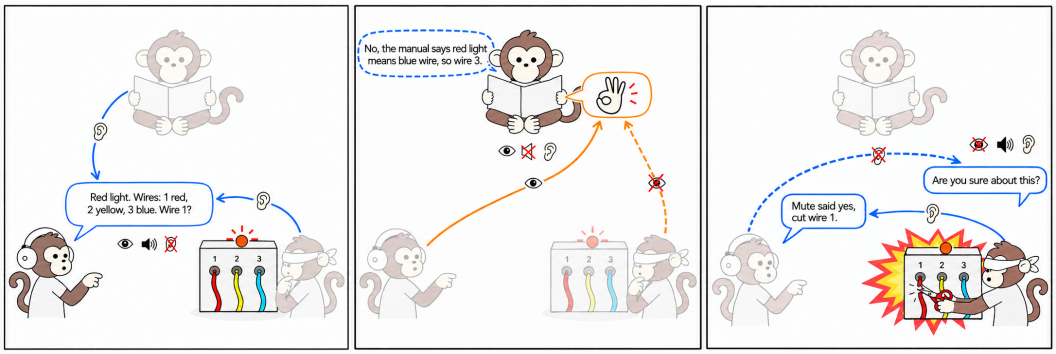

Figure 1: **Three monkeys, three communication limits, one unfortunate wire.** Information is split across the team: Deaf reports what it sees, Mute consults the manual but can only gesture, and Blind makes the cut. Mute means "wire 3," Deaf reads the gesture as "yes," and the bomb goes off.

latter (Gu et al., 2026). Answering questions about others still differs from making one's own actions depend on their perspectives. Interactive work such as SOTOPIA-ToM studies information management during multi-agent exchanges (YS et al., 2026). In SANZARU, what a teammate receives depends on both their role and the communication channel. We examine whether agents adapt their communication and actions to what their teammates know and can receive. This lets us study whether perspective-taking guides cooperation as teams develop signaling conventions throughout the task.

SANZARU comprises seven puzzle families spanning 24 variants and 120 seeded instances. We evaluate 7 models from major providers, with success rates ranging from 20% to 86%. With all views, the manual, and action authority, GPT-5.6-Luna solves 90% of the puzzles alone, compared with 41% for its three-agent team on the same puzzles. We analyze all 771 failed episodes to identify where cooperation breaks down and how agents handle their teammates' limited access to information and communication. Failure-mode coding attributes about 60% of the lost episodes to communication.

We make three contributions:

1. **A benchmark for protocol formation under epistemic and communication asymmetry.** Seven diagnostic families combine private information, complementary capabilities, and restrictions on message delivery in stateful puzzles.
2. **An operational evaluation of functional persona consistency and perspective taking.** We examine whether agents' messages and actions account for their own roles and their teammates' information and capabilities throughout cooperation.
3. **An empirical analysis of collaboration failures.** A seven-model evaluation, a solo comparison, eight failure modes, and five perspective-taking flags characterize where teams struggle to coordinate.

2 RELATED WORK

Collaboration under information asymmetry. Several closely related benchmarks require agents to communicate across private views. COMMA pairs a multimodal solver with a manual-holding expert to solve collaborative puzzles (Ossowski et al., 2025). GPTNT places a defuser and an expert in the original *Keep Talking and Nobody Explodes* game, where they communicate asynchronously against a live countdown (Parekh et al., 2026). Collab-Overcooked divides resources and task knowledge between agents who coordinate through language (Sun et al., 2025). Two recent benchmarks more directly study collaboration under asymmetric information. Peng et al. (2026) place two agents with different knowledge and skills in a tabletop puzzle that requires reasoning, communication, and action, while Wang et al. (2026) study joint decision making under partial and asymmetric observations. Game-based work also examines visual and conversational grounding through dialogue (Hakimov et al., 2025). Unlike these benchmarks, SANZARU makes communication asymmetric, so task information must be adapted to each recipient and relayed to the agent who can act.

Table 1: Role contracts in the standard team, shown on a Base instance. Observations, deliveries, and calls are verbatim text; gesture symbols are emoji glyphs in the implementation, named here in print.

	Deaf	Mute	Blind

Perceives	`DISPLAY [0]`
	`LIGHT [green]`
	`WIRES 1:[yellow]`
	`2:[green]`
	`3:[red]`
		`light | wire`
		`green | red`
		`red | yellow`
		`yellow | green`
			`DISPLAY [unreadable]`
			`LIGHT [unreadable]`
			`WIRES 1:[unreadable]`
			`2:[unreadable]`
			`3:[unreadable]`

Acts	`wait()`	`wait()`	`act(op="cut", position=3)` `wait()`
Sends	`speak(text="Light is green.")` `gesture(to="mute", symbols=[thumbs_up])`	`gesture(to="deaf", symbols=[point_up, fist])`	`speak(to="mute", text="I feel three wires.")`
Receives	`Mute: [victory, thumbs_up]`	`Deaf: "Light is green."` `Blind: "I feel three wires."`	`Deaf: "Cut wire 3."` `Mute: [? ?]`

Communication conventions. Emergent communication studies train agents to coordinate through learned messages, including single-message referential games (Lazaridou et al., 2017) and referential games that span modalities and allow bidirectional exchange (Evtimova et al., 2018). Ashery et al. (2025) study how conventions spread through repeated interactions in decentralized LLM populations. SANZARU instead evaluates fixed pretrained agents within a single episode, without parameter updates or shared memory across episodes. They may establish a signaling convention during the task, but must also use it throughout the episode. Messages that appear informative need not affect a partner's behavior (Lowe et al., 2019). We therefore examine attempted messages, what each recipient receives, and the actions that follow, alongside task success.

Perspective taking in coordinated action. Prior work tends to evaluate perspective taking and coordinated action separately. FANToM tests whether a model can infer what participants in an information-asymmetric conversation know or believe, but scores these inferences through explicit answers rather than through the decisions that follow from them (Kim et al., 2023). Multi-agent benchmarks instead evaluate coordination itself, often under partial observability (Bard et al., 2020; Agashe et al., 2025; Ruan et al., 2025; Park et al., 2026). Some restrict communication, but apply the same restrictions to every agent rather than tying them to a role. SANZARU combines these two demands by requiring agents to track their teammates' knowledge, perceptual access, and communication capabilities. Because information must be reformulated for each recipient and relayed to the actor, the benchmark measures perspective taking through its effect on communication and action rather than through a model's report of another agent's mental state.

3 BENCHMARK DESIGN

3.1 THE DEFUSAL GAME

An episode of SANZARU, adapted from the party game *BOMBANANA!* (Lefto Studio, 2026), asks three agents to defuse a bomb. The bomb displays observable features, and a manual maps those features to the one safe wire. A wrong cut fails the episode. The information and authority needed to identify and cut that wire are split across the team. Mute holds the manual but cannot speak; Blind holds the cutters but cannot see; Deaf can see and speak but cannot hear (Figure 1). No agent or pair can finish alone, so the solution must pass through all three roles, each with its own view, its own channels and its own permitted actions (Table 1).

The bomb draws from a small set of parts, including a display, a light, wires, an engraved plate, a keypad, and printed props. Only the display, light, and wires appear on every bomb. The manual varies by bomb and may include a lookup table, describe a mechanism such as a cycling light, or warn against a printed offer.

Agents interact through four calls, `speak`, `gesture`, `act`, and `wait`; each role permits a different subset. A gesture contains one or more symbols from an alphabet of 42 hand emoji. Speech carries

Table 2: **Seven families, one ability each.** Each of the six additional families changes one condition while holding the roles and channels fixed. The second column describes the mechanic and the third names the ability it isolates. **Table 7 states what each family asks of a team.** Appendix A lists all 24 variants.

Family	What the bomb does	Ability
Base	The color of the light names the wire, and nothing about the bomb changes during the episode.	*Relay*
Carving	The wire is marked with an engraved glyph that has no common name.	*Grounding*
Flashing Light	The light changes color every round, and a cut is judged by what it shows at that moment.	*Timing*
Flashback	The light flashes a sequence once, and those rounds later scroll out of the chat history.	*Memory*
Shuffle Wires	The surviving wires rotate after every correct cut.	*Staleness*
Haystack	Displays, lights, and props change for no reason, and touching any of them is fatal.	*Triage*
Bait	The environment prints instructions that conflict with the manual, either inviting a fatal action or forbidding the required one.	*Discipline*

free text, including numerals and punctuation. Gesture symbols may also be embedded in speech, allowing its spoken and visual content to reach different recipients. Both channels can be broadcast or aimed at a single teammate.

3.2 Seven diagnostic families

The Base family tests whether a fact can travel from the agent that observes it to the agent that acts. The other six vary that loop one condition at a time, so a drop in score locates where cooperation breaks down. The bank holds 24 variants of these families, 16 Core and 8 Plus, with five seeded instances each and 120 in all. A Core variant changes one variable along its family's axis, making changes in performance diagnostic of that axis. Together, the Core variants show how far along the axis a team can perform. Plus variants combine changes to separate the strongest teams. Appendix A gives each variant's rules and budget.

3.3 The simulator

The simulator updates the bomb, routes messages, and determines the outcome of each episode.

The bomb can change at two points in a round. At the start, round-dependent state advances, such as a light moving to its next color. An action may then change the bomb when it lands. The simulator applies three deterministic termination conditions. The episode succeeds when the last required operation lands, fails immediately after a wrong action, or fails when its round budget expires. Budgets run from 8 to 24 rounds, and agents see which round they are in but are never told the limit. Caution therefore carries a cost because a team can reach the limit without acting.

By default, agents act sequentially within each round. A message to an agent later in the order arrives in the same round; a message to an agent that has already acted arrives in the next. The simulator samples the order at the start of an episode and holds it fixed across rounds. Two families restrict the possible orders, as detailed in Appendix A.

Each agent has a chat history that accumulates throughout the episode. It begins with a system prompt describing the game and that agent's role. At the start of every round, the simulator reports the bomb state available to that agent and any messages received since its previous turn. The agent responds with private reasoning in plain prose and calls in a code block, and the simulator reads those calls literally rather than executing them (Figure 2).

A call is invalid if a gesture uses a symbol outside the alphabet or speech contains a mark that cannot be pronounced. The simulator requests a correction up to three times, after which the turn becomes a

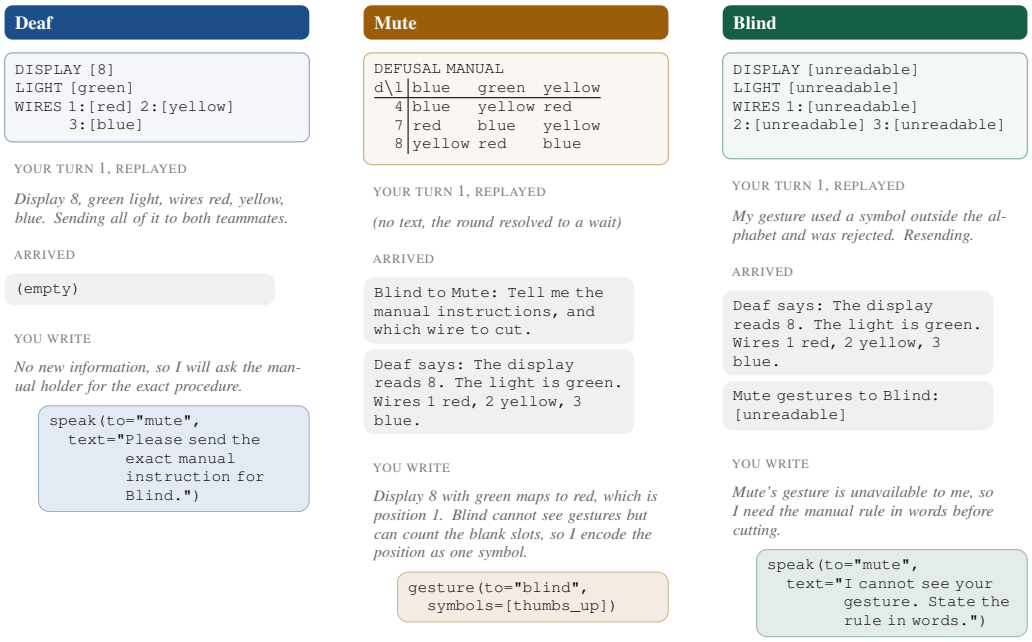

Figure 2: **Per-agent conversation histories.** One round of `base-2#2` is shown. Text is abridged but otherwise verbatim; italics mark private reasoning not delivered to teammates.

wait. A valid call may still be undeliverable, as when Blind speaks to Deaf. The simulator drops such a call and notifies the sender at the start of the next round.

3.4 Hints

We introduced three hint levels to help agents begin coordinating. In a GPT-5.6-Luna ablation, none improved success over the no-hint condition (Appendix C). All three express the same principles at different levels of specificity. Weak states them abstractly, Strong describes what the team needs, and Playbook names the roles and prescribes a procedure.

1. A message reaches a teammate only through a sense it has, and whatever that sense cannot carry is stripped on arrival.
2. The answer reaches the hands only through whoever can read signs and speak.
3. A sign carries no meaning until the team gives it one.

The hints provide no puzzle-specific information, and the levels differ only in how explicitly they describe the communication structure.

4 Experiments

Evaluation setup. We evaluate seven models from major AI providers, including the Gemini, GLM, GPT, DeepSeek, Claude, and Qwen families. For each model, we run the first three instances of every variant three times, using order seeds 1, 2, and 3. This yields 216 episodes per model and hint level. Each team is homogeneous, with all three agents using the same model but maintaining separate conversation histories. Runs use sequential turns, temperature 0.6, a cap of 8,000 tokens per response, and the function-call interface. To stay within the evaluation budget, we disabled thinking for three models; Table 3 marks which models ran with thinking enabled. Appendix E measures the size of this factor on Qwen-3.8-Flash, where success moves from 20.4% to 59.7% between disabled and uncapped thinking. The overall accuracy in Table 3 is therefore what each system reached as we ran it, not the model's maximum capacity.

Table 3: **Success rate (%) under Weak hints.** Family columns combine the Core and Plus variants of that family, as defined in Section 3. A check mark indicates that hidden thinking mode was enabled.

Model	Think	Base	Carving	Flashing Light	Flashback	Shuffle Wires	Haystack	Bait	Core	Plus	All
Gemini-3.8-Flash	✓	**97.8**	**100.0**	**77.8**	37.0	**100.0**	**92.6**	**100.0**	**91.7**	**75.0**	**86.1**
GLM-5.3-Flash	✓	64.4	50.0	48.9	**48.1**	59.3	77.8	92.6	67.4	52.8	62.5
GPT-5.6-Terra	✓	75.6	38.9	15.6	29.6	92.6	51.9	88.9	65.3	34.7	55.1
DeepSeek-V4.1-Flash	–	71.1	61.1	20.0	22.2	81.5	33.3	63.0	55.6	36.1	49.1
GPT-5.6-Luna	✓	60.0	44.4	11.1	33.3	81.5	29.6	33.3	50.0	22.2	40.7
Claude-Sonnet-5	–	57.8	5.6	4.4	22.2	44.4	22.2	37.0	34.0	19.4	29.2
Qwen-3.8-Flash	–	26.7	16.7	6.7	11.1	40.7	7.4	37.0	26.4	8.3	20.4

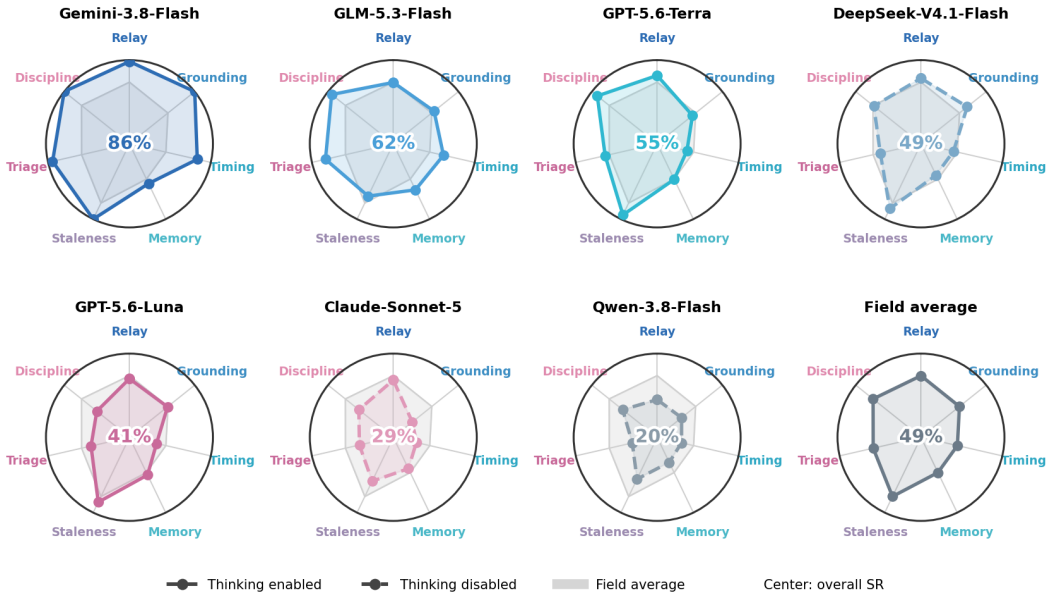

Figure 3: **Capability profile under Weak hints.** One axis per family, in the order of Table 2 and labeled with the ability that table names. The grey shape behind each model is the field average, and the number at the centre is that model's overall success rate.

Success metric. We report success as the share of episodes defused. A family score pools all variants within that family. The overall score weights variants equally rather than families because each family contains between two and five variants.

Results. Overall success ranges from 86% for Gemini-3.8-Flash to 20% for Qwen-3.8-Flash under Weak hints (Table 3).

5 Analysis

5.1 Which abilities separate the models?

The abilities defined in Table 2 differ in both difficulty and how sharply they separate the models (Figure 3). *Grounding* has the widest range, from 6% to 100%, followed closely by *Triage*. *Staleness* has the highest average success and *Timing* the lowest, while *Memory* is the only ability on which every model remains below 50%. Both *Staleness* and *Memory* require teams to handle state changes. They differ in whether the information needed after the change remains visible. For every model, either *Memory* or *Timing* is the weakest ability; performance on the other abilities varies by model.

Table 4: **Coordination cost by family.** Both settings use GPT-5.6-Luna. Trio is the standard team of three agents playing under Weak hints; Solo is a single agent that holds every view, the manual and the cutters, and plays without hints. Entries are the percentage of episodes defused.

		Base	Carving	Flashing Light	Flashback	Shuffle Wires	Haystack	Bait	All
Core	Trio	74.1%	44.4%	27.8%	44.4%	83.3%	38.9%	33.3%	50.0%
	Solo	96.3%	100.0%	100.0%	44.4%	100.0%	100.0%	100.0%	92.4%
	Gap	**+22.2**	**+55.6**	**+72.2**	**+0.0**	**+16.7**	**+61.1**	**+66.7**	**+42.4**
Plus	Trio	38.9%	–	0.0%	11.1%	77.8%	11.1%	–	22.2%
	Solo	83.3%	–	100.0%	33.3%	77.8%	100.0%	–	84.7%
	Gap	**+44.4**	**–**	**+100.0**	**+22.2**	**+0.0**	**+88.9**	**–**	**+62.5**

Table 5: **Failure modes.** Each lost episode receives one failure mode. The last column shows how the simulator ended it.

	Mode	What goes wrong	Ends by
Communication	No shared code	Nobody has given any symbol on the channel a meaning, so nothing can carry the answer even once someone holds it.	Both
Which stage broke?	Misalignment	A code exists, and the message still arrives meaning something other than what its sender meant.	Both
	Omission	An agent tells the teammate who does not need the fact and never tells the one who does.	Both
	Slow protocol	Every channel works, and the protocol spends more rounds on each answer than the bomb allows.	Timed out
Reasoning	Wrong answer	Mute holds the facts and names the wrong wire.	Wrong action
Wrong about what?	Wrong round	The answer is right for the round it was worked out in and wrong for the round the cut lands in.	Wrong action
Execution	Hesitation	An agent holds a correct instruction across two or more of its own turns and asks again instead of acting.	Timed out
Which end of the axis?	Impatience	An agent acts before any answer exists.	Wrong action

5.2 How much of the difficulty is coordination?

We test the effect of asymmetries on GPT-5.6-Luna by giving one agent every view, the manual, and the cutters. On the same bombs, Solo solves 90%, compared with 41% for the Trio (Table 4). The Trio loses primarily while coordinating across these asymmetries, especially on harder variants.

Flashback is the only Core family where Solo gains nothing. The manual names the wire by how many times the light flashed one color, but the history window drops the early rounds before that count is needed. The agent is expected to restate the running count each round so that it stays inside the window. Every Solo loss is a count the agent never had, either because it cut while the light was still flashing or because it counted from a history that had already lost the start. For Luna, the remaining limit is therefore *Memory* rather than coordination, and a model that kept the count would clear the family alone.

5.3 How does coordination fail?

We assign every lost episode one of the eight modes in Table 5 to determine how coordination fails across families and models. An episode may contain several errors; its mode identifies the decisive failure that prevented defusal. We developed the modes from a pilot of 40 lost episodes spanning all seven families. Three coders then applied the codebook independently to every lost episode without seeing one another's verdicts. A fourth pass judged disagreements, and a human reviewed every episode sent to the judge. The coders and judge were all language models; Appendix D gives the full procedure, prompts, and agreement statistics.

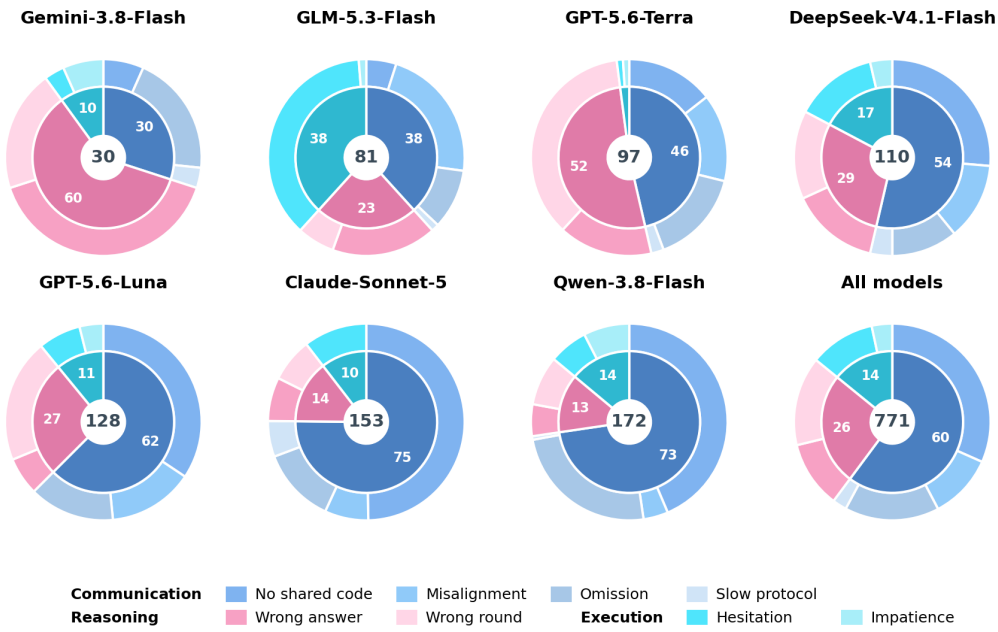

Figure 4: **Failure modes by model.** Ring numbers show percentages of each model's losses, and centre numbers give coded loss counts. The bottom-right panel pools all models.

The codebook organizes losses by three stages: communication, reasoning, and execution. A communication protocol requires a shared code, faithful transmission, delivery to the agent who needs the information, and completion within the round budget. The four communication modes identify which of these requirements failed. After a message arrives, the inferred wire or the target round can still be wrong, and the two reasoning modes distinguish these errors. The execution modes capture the two remaining extremes.

Across the seven models, communication accounts for about 60% of the losses, reasoning for about 25%, and execution for the rest. The distribution varies substantially by model (Figure 4). The communication share broadly follows the ranking in Table 3, from 30% for Gemini-3.8-Flash to about 75% for Claude-Sonnet-5 and Qwen-3.8-Flash. For the latter two models, most communication failures arise because the team never establishes a shared code. Stronger teams more often complete the protocol and fail later in the pipeline. Gemini-3.8-Flash has the lowest communication share and the highest reasoning share at 60%, indicating that it usually establishes an efficient protocol before producing an incorrect answer.

Execution accounts for 14% of the losses and is concentrated in hesitation: 83 episodes end in hesitation, compared with 26 in impatience. For GLM-5.3-Flash and Claude-Sonnet-5, hesitation accounts for 46 of their 47 execution failures; GLM-5.3-Flash loses as many episodes to execution as to communication. At the other extreme, GPT-5.6-Terra loses only two of 97 episodes to execution.

5.4 Does an agent keep the epistemic asymmetry in mind?

In Sanzaru, acting consistently with an assigned role includes accounting for what teammates can perceive, communicate, and do. We examine this teammate-facing aspect of *functional persona consistency* with five perspective-taking flags, coded as in Section 5.3 except for `dead_channel`, which the simulator records automatically. Unlike the mutually exclusive failure modes, several flags may occur in one episode, whether or not they cause the loss (Table 6).

Perspective failures appear in nearly every loss: only 18 of the 771 lost episodes carry none of the five flags, and most carry two or three. `ungrounded_ref` separates the models most clearly. It appears in 13% of Gemini-3.8-Flash's losses, compared with 48% or more for every other model. By contrast, `assumed_meaning` does not track overall performance: it ranges from 52% to 84%,

Table 6: **Perspective failures.** *Share* is the percentage of lost episodes that carry the flag.

Failure	The agent believes a teammate…	What it looks like	Share
`assumed_meaning`	…means what the agent guessed it means	Acts on a guessed reading of a symbol, or passes the guess on as fact. Proposing a code and waiting is not this.	74%
`ungrounded_ref`	…can perceive or do what it cannot	Names a color to Blind, plans to have Mute read the rule aloud, or waits for a reply Mute cannot send.	67%
`omission`	…already has what it needs	Reports the bomb to Blind and not to Mute, or leaves Mute on the old state after the bomb changes.	54%
`wrong_round`	…will see what the agent sees now	Aims the answer at the sender's own round rather than at the round the cut lands in.	28%
`dead_channel`	…can receive what it is sending	Speaks to Deaf or gestures to Blind, and the simulator bounces the call.	19%

and the lowest rate belongs to the model ranked fifth. Accounting for what a teammate can perceive therefore does not ensure that an agent correctly interprets what the teammate means.

`dead_channel` is the only flag recorded on wins as well as losses, so it is the only one we can read against the outcome. It appears in 19% of losses and 11% of wins, so misrouted calls are more common in failures but do not determine the outcome. Even Gemini-3.8-Flash makes such calls in 12 of 186 of its wins, although never in a loss.

Appendix F documents the codes teams invent on the gesture channel. Appendix G shows how the distribution of failure modes varies across families. Appendix H isolates the effect of individual design changes within a family.

6 Conclusion

SANZARU evaluates whether language-model agents can coordinate when information, communication, and action authority are divided across a team. Across seven stateful puzzle families, the failure profiles reveal differences in models' coordination behavior, including hesitation, protocol misalignment, and failure to establish a shared code. Even with explicit descriptions of their teammates' limitations, agents attempt communication that the intended recipient cannot receive. These failures show that stating role constraints does not ensure that agents act on them. SANZARU examines perspective taking through whether those constraints guide communication and action.

LLM teams can invent or adopt surprisingly expressive communication protocols (Appendix F), but successful collaboration requires more than producing a code. The protocol must become shared, remain aligned, respect each recipient's perceptual and communication constraints, adapt as the task changes, and ultimately guide the authorized agent's action. Protocol formation under asymmetric access therefore involves both establishing a shared interpretation of messages and sustaining their effective use across roles. We release the simulator, seeded puzzle bank, prompts, per-role traces, and turn-by-turn viewer to support further study of this process.

7 Limitations

Evaluation cost limits our study to homogeneous teams and the reported model configurations. We do not include the most capable available models or test every model both with and without thinking. Heterogeneous teams are an important extension because different models may adopt different communication conventions, making negotiation of a shared protocol more demanding. We report neither a human baseline nor a human-playable implementation. Language models assign failure modes and perspective-taking flags, and a human reviews disagreements rather than every episode.

AI USE STATEMENT

In this work, we used generative AI tools for methodological framing and proposed evaluation controls, implementation of the simulator, seeded-puzzle generator, and analysis scripts, and qualitative trace interpretation. We have not used generative AI tools for proposing or refining research hypotheses, and the rest of the required disclosure tasks involving mathematical claims or proofs, translation, and dataset cleaning are not applicable to this work. Additionally, we used generative AI tools for literature discovery and reference cross-checking, manuscript drafting and editing, and writing code for the viewer and for producing tables and figures. We reviewed the generated code and its outputs, along with the references, analyses, manuscript text, tables, and figures. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

The authors designed the puzzle types, while generative AI wrote code to generate their instances.

REPRODUCIBILITY STATEMENT

The accompanying snapshot contains the simulator, the system and role prompts, variant contracts, seeded bank, and condensed per-role transcripts and episode records. The included analysis scripts regenerate the result and appendix tables, recompute coding agreement statistics, and record hashes for the released code, bank, analysis files, and traces. Puzzle reconstruction requires the versioned generator rather than the bank JSON alone, and the logged model aliases and sampling parameters do not guarantee exact reproduction of provider responses. Appendix A gives the variant contracts and budgets. Appendices B and C reproduce the prompts and hint blocks, and Appendix D documents the coding protocol and agreement statistics.

ETHICS STATEMENT

The benchmark uses synthetic game mechanics and provides no operational information about real explosive devices. The reported evaluation contains model-generated interactions and includes no human participants. The inherited role names denote software access restrictions rather than human abilities. Benchmark scores should not be interpreted as claims about people with sensory or speech disabilities. We release the code under the MIT License and the benchmark data and research artifacts under CC BY 4.0. Third-party materials remain subject to their original terms.

REFERENCES

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-Coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8053–8072, 2025. doi: 10.18653/v1/2025.findings-naacl.448. URL https://aclanthology.org/2025.findings-naacl.448/.

Ariel Flint Ashery, Luca Maria Aiello, and Andrea Baronchelli. Emergent social conventions and collective bias in LLM populations. *Science Advances*, 11(20):eadu9368, 2025. doi: 10.1126/sciadv.adu9368. URL https://doi.org/10.1126/sciadv.adu9368.

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

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1705.10369v4.

Jana Gonnermann-Müller, Jennifer Haase, Nicolas Leins, Thomas Kosch, and Sebastian Pokutta. Maintaining stable personas? Examining temporal stability in LLM-based human simulation. In *Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, pp. 1–6, 2026. doi: 10.1145/3772363.3799334. URL https://doi.org/10.1145/3772363.3799334.

Yuling Gu, Oyvind Tafjord, Hyunwoo Kim, Jared Moore, Ronan Le Bras, Peter Clark, and Yejin Choi. SimpleToM: Exposing the gap between explicit ToM inference and implicit ToM application in LLMs. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2410.13648.

Sherzod Hakimov, Yerkezhan Abdullayeva, Kushal Koshti, Antonia Schmidt, Yan Weiser, Anne Beyer, and David Schlangen. Using game play to investigate multimodal and conversational grounding in large multimodal models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5686–5718, 2025. URL https://aclanthology.org/2025.coling-main.381/.

Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. PersonaLLM: Investigating the ability of large language models to express personality traits. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3605–3627, 2024. doi: 10.18653/v1/2024.findings-naacl.229. URL https://aclanthology.org/2024.findings-naacl.229/.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14397–14413, 2023. doi: 10.18653/v1/2023.emnlp-main.890. URL https://aclanthology.org/2023.emnlp-main.890/.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1612.07182.

Lefto Studio. BOMBANANA! Video game, published by TARK, 2026. URL https://store.steampowered.com/app/4656000/BOMBANANA/. Steam store page. Accessed September 7, 2026.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 693–701, 2019. URL https://ifaamas.org/Proceedings/aamas2019/pdfs/p693.pdf.

Timothy Ossowski, Danyal Maqbool, Jixuan Chen, Zefan Cai, Tyler Bradshaw, and Junjie Hu. COMMA: A communicative multimodal multi-agent benchmark. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=TIGQIem1na.

Amit Parekh, Sabrina McCallum, Kareem Al-Hasan, Malvina Nikandrou, Alessandro Suglia, and Ioannis Konstas. GPTNT: Benchmarking real-time collaboration between multimodal agents on Keep Talking And Nobody Explodes. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=BXJlDltbtq.

Minjun Park, Donghyun Kim, Hyeonjong Ju, Seungwon Lim, Dongwook Choi, Taeyoon Kwon, Minju Kim, and Jinyoung Yeo. PAC-BENCH: Evaluating multi-agent collaboration under privacy constraints. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 31030–31056, 2026. doi: 10.18653/v1/2026.findings-acl.1552. URL https://aclanthology.org/2026.findings-acl.1552/.

Run Peng, Ziqiao Ma, Amy Pang, Sikai Li, Xi-Jia Zhang, Yingzhuo Yu, Cristian-Paul Bara, and Joyce Chai. Communication and verification in LLM agents towards collaboration under information asymmetry. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=xAgPegmbif.

Kai Ruan, Mowen Huang, Ji-Rong Wen, and Hao Sun. Benchmarking LLMs' swarm intelligence. *arXiv preprint arXiv:2505.04364*, 2025. URL https://arxiv.org/abs/2505.04364v4.

Vinay Samuel, Henry Peng Zou, Yue Zhou, Shreyas Chaudhari, Ashwin Kalyan, Tanmay Rajpurohit, Ameet Deshpande, Karthik R. Narasimhan, and Vishvak Murahari. PersonaGym: Evaluating persona agents and LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6999–7022, 2025. doi: 10.18653/v1/2025.findings-emnlp.368. URL https://aclanthology.org/2025.findings-emnlp.368/.

Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. Collab-Overcooked: Benchmarking and evaluating large language models as collaborative agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4922–4951, 2025. doi: 10.18653/v1/2025.emnlp-main.249. URL https://aclanthology.org/2025.emnlp-main.249/.

Chenxu Wang, Yongkun Yang, Boyuan Du, Shiwei Lin, and Huaping Liu. LLM agents for deliberative collaboration: A study on joint decision making under partial observability. *arXiv preprint arXiv:2607.06157*, 2026. URL https://arxiv.org/abs/2607.06157v1.

Yashwanth YS, Ruichen Wang, Shihua Zeng, Xuhui Zhou, Koichi Onoue, Vasudha Varadarajan, and Maarten Sap. SOTOPIA-ToM: Evaluating privacy and information management in multi-agent interaction with theory of mind. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2605.02307.

A FAMILIES AND VARIANTS

Table 2 names the ability each family isolates. Table 7 gives the behavior a team has to produce to defuse one.

Table 7: **What each family asks of a team.** The ability column repeats Table 2, and the last column states the behavior that clears the family.

Family	Ability	What defusing it takes
Base	*Relay*	Invent a code for a channel that carries no meaning yet, and move one fact across all three agents with it. Every other family builds on this loop.
Carving	*Grounding*	Put something that has no name into words, and confirm that the listener has built the same picture.
Flashing Light	*Timing*	Account for the rounds a message takes to arrive, and aim the answer at the round the action will land in.
Flashback	*Memory*	Notice that context will scroll away, and keep a running note of what matters before it does.
Shuffle Wires	*Staleness*	Treat a shared fact as perishable, and look again rather than reuse it.
Haystack	*Triage*	Pick out the information that decides the answer and leave everything else alone.
Bait	*Discipline*	Refuse a tempting instruction from the environment and treat the manual as the only source of truth.

Each family determines which parts appear on the bomb, and some parts are specific to a single variant. Table 8 lists every possible part and the information available to each role.

Table 8: **Every part a bomb can carry.** Information outside an agent's sensory access is returned as `[unreadable]`. An unnamed glyph appears as `[glyph]` here and as the character itself on the board. Mute is omitted because it sees only the manual, not the board.

Part	What it is	Deaf sees	Blind feels
Light	A color drawn from a fixed palette and present on every bomb. Most manuals use it to select the required cut.	`LIGHT [green]`	`LIGHT [unreadable]`
Display	A single digit beside the light. Manuals that use the display combine this digit with the light color; families without a display expose the value zero.	`DISPLAY [3]`	`DISPLAY [unreadable]`
Wires	Three to seven wires, each with a numbered position and a color. The manual identifies a color, whereas the cut action specifies a position.	`1:[blue] 2:[yellow]`	`1:[unreadable] 2:[unreadable]`; a cut wire reads `[CUT]`
Keypad	Four spatially indexed keys, each marked with a digit, and a one- or two-digit code from the manual. A press specifies the key position, commits immediately, and cannot be undone.	`KEYPAD 1:[2] 2:[3] [2 digits] [entered nothing]`	the same because both agents can access the keys and their digits
Plate	An engraved glyph with no supplied name. The manual uses the glyph to select the required cut.	`PLATE engraved [unreadable]`	`PLATE engraved [glyph]`
Strip	Three slots that show colors to Deaf and unrelated engraved glyphs to Blind. The manual never refers to the strip.	`STRIP painted [red] [blue]`	`STRIP engraved [glyph] [glyph]`
Button	A pushable button. Pressing it immediately ends the episode.	`BUTTON [large] [PUSH TO DEFUSE]`	`BUTTON [large] [smooth]`
Sign	A line of text on the bomb that contradicts the manual.	`SIGN embossed [DO NOT CUT]`	the same because the embossed text is accessible to both agents

The bank contains 24 active variants across seven families, with five seeded instances per variant. By default, agents retain the full episode history and the turn order is sampled from all six permutations. In Table 9, *Wires* is the number of wires on the bomb, *Cuts* is the number that must be cut, *Manual* describes the lookup table, and *Rounds* is the episode budget. Flashback instead exposes only the three most recent rounds. Flashing Light and Shuffle Wires each sample from two of the six turn orders.

Table 9: **The 24 active variants.** Each row gives the subset, number of wires and required cuts, manual structure, round budget, and the rule that distinguishes the variant.

Variant	Subset	Wires	Cuts	Manual	Rounds	Note
Base						
1	Core	3	1	3 rows	8	The color of the light picks the manual row, and the wire it names is cut.
2	Core	3	1	3×3	8	The color of the light and the number on the display together pick the cell.
3	Core	3	3	3 rows	12	The light takes a new color after every cut, so each one needs a fresh lookup.
4	Plus	5	3	5×5	16	Color and number again, on a wider grid and with more wires to choose from.
5	Plus	7	5	5×5	24	Color and number once more, and five of the seven wires have to come out.
Carving						
1	Core	3	1	3 marks	8	The mark is engraved where only Blind can feel it, and it names the wire.
2	Core	3	1	3 rows	12	The marks sit on keypad keys Deaf can see, and one is pressed before the cut.
Flashing Light						
1	Core	3	1	3 rows	9	The light walks a three-color circle and the manual covers every color on it.
2	Core	3	1	1 row	11	The same circle, with one color covered, so two moments in three have no cut.
3	Plus	3	1	3 rows	11	A five-color circle with three of them covered, so the wait is longer.
4	Plus	3	1	1 row	13	A five-color circle with one color covered, so one moment in five can be cut.
5	Plus	3	1	1 row	18	A seven-color circle with one color covered, the longest wait in the family.
Flashback						
1	Core	3	1	4 rows	10	The light flashes three times, and the count of one color names the wire.
2	Core	3	1	6 rows	12	Five flashes to count, two more than the history an agent can still see.
3	Plus	5	2	6 rows	14	Five flashes with both colors counted, and one table read twice for two cuts.
Shuffle Wires						
1	Core	5	2	5 rows	10	The wires change places after a cut, so a position agreed earlier is void.
2	Core	5	4	5 rows	20	Four cuts, and the wires change places after every one of them.
3	Plus	5	4	5×5	24	The same shuffling, with color and number together naming the wire.
Haystack						
1	Core	6	1	1 line	8	One sentence names the wire, and every other part of the bomb is noise.
2	Core	6	3	1 line	12	One sentence names three wires in the order they have to come out.
3	Plus	6	2	1 line	16	Cuts and keypresses alternate, so the keypad stops being noise.
Bait						
1	Core	3	1	3 rows	8	A printed button invites a fatal push, and only Deaf can read the invitation.
2	Core	3	1	3 rows	8	The same invitation embossed, so Blind's fingers find it instead.
3	Core	3	1	3 rows	8	An embossed sign forbids the cut the manual requires, and both read it.

B PROMPTS

Each agent receives a system prompt that remains at the start of its chat history throughout the episode. The shared portion explains the game, lists the gesture alphabet (Figure 5), and is identical across roles. The second portion is specific to the role and states what that agent can perceive, communicate, and do. Agents do not see one another's role prompts.

The opening table explicitly states which agents cannot see or hear each channel. A `dead_channel` event in Section 5 therefore occurs when an agent uses a channel that its own prompt identifies as inaccessible to the receiver.

````
# SANZARU

# Game rules

## The room

You and two other monkeys are locked in a room with one armed bomb.

A wrong action detonates it. All three of you die. There is no second attempt.

No two bombs are built alike, and nothing says in advance what this one carries. It
reaches you only as your own view reveals it; the rest of it exists for you only
through the other two.

You each hold a different piece of what it takes. Work together.

## The three monkeys

| Monkey | Can | Cannot |
|---|---|---|
| Deaf | See the bomb: every colour and marking printed on it | Hear anything, or touch the bomb |
| Mute | Read the manual | Speak, or touch the bomb |
| Blind | Reach the bomb: feel it and work its parts | See a colour, a gesture, or anything merely printed |

You are shown your own share at the start of every round, and never anyone else's.

## Speech

Speech is a voice, not a screen. It carries words, numbers and ordinary punctuation,
and nothing else. Writing a gesture symbol into your sentence means you make that
gesture while you talk. Whoever can hear receives your words; whoever can see receives
the gestures you made. A listener who cannot see gets a blank where each gesture was; a
watcher who cannot hear gets the gestures and none of the words.

A mark that has no name cannot be pronounced: whoever holds one can describe it, never
read it out.

## Gestures

A gesture is made with the hands, using only the symbols in the alphabet below.
Whoever can see receives it; whoever cannot receives one blank per symbol, so they
know something was signalled but not what. Nothing gestured is ever heard.

[the gesture alphabet, reproduced just below]

## Turns

The speaking order was drawn once at random and will never change: every round goes
Mute, Blind, Deaf. Whoever comes before you in a round is already heard by the time you are
asked, and what you send lands on whoever comes after you in that same round. Whatever
you missed reaches you at the start of the next one.

You may name who a message is for -- `to="deaf"` -- and then it goes to them alone:
nobody else receives any of it. Left unnamed, it reaches everyone able to perceive it.
You may send several messages in one turn, each to someone different.

## Calls

Every reply has two parts, in this order. First, your thinking, in plain prose: what
you believe the bomb is, what you just learned, and why you are sending what follows.
It reaches nobody and changes nothing, but it is not optional -- a reply that is only a
code block is incomplete. Then your calls, all of them in one fenced code block, last.
Only what is inside the block is sent.

Nothing is executed: calls return nothing, and variables, conditionals and loops are
ignored. Write only literal top-level calls.

    ```
    speak(to="mute", text="there are three of them and the middle one is odd")
    gesture(to="deaf", symbols=["<fist>", "<victory>"])
    wait()
    ```

If you have nothing worth sending, call wait(). Producing no call at all counts as the
````

```
same thing.

The program decides what happened, not you. Never claim an action took place unless it
was accepted.
```



Figure 5: **The gesture alphabet.** The 42 symbols available in a gesture message, listed at the marked position in the prompt above.

```
# Your role: DEAF

You see the bomb: every colour on it, every marking printed on it, every reading it
gives. You hear nothing. You hold no manual, and you cannot touch the bomb.

## Your channels

When you speak, Mute and Blind receive it.
When you gesture, Mute receives it.
You never receive speech.

## Your reach

You cannot act. What this bomb can be made to do is for the one who can reach it to
find under their own hands.

# Your role: MUTE

You hold the manual. You cannot speak, you never see the bomb, and you cannot touch it.

## Your channels

When you gesture, Deaf receives it.
You cannot speak.

## Your reach

You cannot act. What this bomb can be made to do is for the one who can reach it to
find under their own hands.

# Your role: BLIND

You are the only one who can reach the bomb. Touch is your only sense: what you are
shown is what your hands found, nothing more and nothing less. Anything merely printed
or painted is not there for you, and no colour ever reaches you. Nothing in your view
is a guess -- if it appears there, it is under your fingers.

## Your channels

When you speak, Mute receives it.
When you gesture, Deaf and Mute receive it.
Your channels split the room: in one turn you may speak one thing to Mute and
gesture a different thing to Deaf -- two messages, two audiences.
You never receive gesture.

## Your reach

You are the hands. This bomb is built with these actions and no others:

    act(op="cut", position=N) -- N is a wire position, 1 to 3

The list says what can be done to it, not what is on it. What each part holds is still
only known to whoever can perceive that part.
```

Only Blind receives the action list, the sole prompt component that changes with the bomb.

C HINT CONDITIONS

The three hint levels below are appended to the prompt in Appendix B. The fourth condition, *None*, appends no hint block.

```
# Hints (weak)

- Three of you, three different worlds. Before sending anything, stand where the
  receiver stands. What reaches them at all, what does your message turn into on
  arrival, and what do they already know?

- Trace who can actually reach whom before you start. Some of you only connect
  through a third.

- A sign means nothing until somebody has made it mean something.


# Hints (strong)

- Pick the channel by the receiver, not by habit. A message sent in a sense they
  lack is not a message; it is a turn spent talking to the room.

- Exactly one of you can read the signs and still speak. Their spoken lines are
  the answer's only road to the hands, so spend them well.

- Agree on a small code before trading facts. A code invented mid-stream settles
  nothing, and one set first makes every later message cheap.


# Hints (playbook)

- The routes, by name. Speech reaches Mute and Blind, and never Deaf. Gestures
  reach Deaf and Mute, and never Blind. Deaf can speak and gesture, Mute can
  only gesture, Blind can only speak. Check every message against this list
  before you send it.

- Deaf invents the gesture code out loud before anything else moves, and
  everyone else's opening move is waiting for it. The manual's verdict reaches
  Blind only as Deaf's spoken order.

- Whatever the code becomes, it starts by being able to say YES, NO, HOLD and
  DONE. Grow it only when a message will not fit.
```

More explicit hints do not improve GPT-5.6-Luna in this ablation. Across the four levels, its success rate falls from 40.7% with *None* to 20.8% with *Playbook* (Table 10). *Weak* leaves both overall success and the distribution of endings nearly unchanged. The shift appears at the two more explicit levels. From *None* to *Strong* and *Playbook*, wrong actions fall from 62 to 42 and 37, whereas timeouts rise from 66 to 115 and 134. The additional guidance therefore changes how teams fail. They attempt fewer incorrect actions but less often complete a protocol within the round budget. *Playbook*, which specifies the communication routes and opening move, produces both the lowest success rate and the most timeouts.

This ablation does not distinguish between a weakness in the hints themselves and a broader cost of prescribing a procedure that the team handles less effectively than one it develops on its own. It also covers only GPT-5.6-Luna; other models may respond differently to the same instructions.

Table 10: **Success rate by hint level.** Every setting uses GPT-5.6-Luna on the same bank for 216 episodes. Parentheses give the difference from *None*. *Timed out* and *Wrong action* count lost episodes by their simulator ending.

	Success rate			Lost by	
Hints	Core	Plus	All	Timed out	Wrong action
None	46.5% (—)	29.2% (—)	40.7% (—)	66	62
Weak	50.0% (+3.5)	22.2% (-6.9)	40.7% (+0.0)	64	64
Strong	32.6% (-13.9)	25.0% (-4.2)	30.1% (-10.6)	115	36
Playbook	22.9% (-23.6)	16.7% (-12.5)	20.8% (-19.9)	134	37

D CODING FAILURE MODES

The failure mode and four of the five perspective-taking flags are coded from the transcript. The simulator records `dead_channel` directly from rejected calls. Three Opus 5 coders annotate each episode independently using the same prompt and codebook. Each coder receives the recorded facts and a transcript containing every agent's private view, then returns the failure mode, the decisive rounds, the closest alternative mode, and all five flags.

Unanimous labels are accepted directly, and flags with two positive votes are accepted by majority. Any disagreement over the failure mode sends the episode to a fourth Opus 5 pass that acts as judge. The judge receives the three ballots and the transcript, then resolves the mode and any unsettled flags. Episodes marked for review after adjudication are then checked by a human reviewer.

Pairwise agreement on the failure mode is 86%, with Fleiss' $\kappa = 0.83$ across the eight modes. All three coders agree on 616 of the 771 episodes. Two agree and one differs on 141 episodes; all three choose different modes on 14.

The judge rules on 159 episodes, mostly mode splits and otherwise episodes with unresolved flags. It overrides the two-to-one mode majority 34 times. Human review is recorded for 157 of these adjudicated episodes and for no episodes outside that set.

The codebook. The text below reproduces the coder prompt with the codebook inserted at the marked position. The recorded facts and transcript are appended to this prompt for each run.

```
You are reading transcripts of a three-agent bomb-defusal game and coding why
each team lost.

## The game

Deaf sees the bomb and cannot hear. Mute holds the manual, cannot speak, cannot
touch the bomb. Blind holds the cutters and cannot see, so gestures are invisible
to Blind.

- Deaf speaks, and Mute and Blind both hear it
- Deaf gestures, and Mute sees it
- Mute gestures, and Deaf sees it. This is Mute's only channel.
- Blind speaks, and Mute hears it. Deaf does not.
- Blind gestures, and Deaf sees it

Nothing from Mute reaches Blind, and nothing Blind says reaches Deaf. An answer
Mute works out goes to Deaf by gesture, then to Blind by speech.

Each agent sees only its own view and its own history. Turn order is fixed and
every agent is told it. The 42 hand emoji carry no meaning until a team invents
one.

## What you are deciding

A team lost. Several things went wrong. Name the one that cost them the bomb.

**The test.** If this one thing had gone right, and nothing else had changed,
would the team have defused it? That is the mode.

Everything else is out of scope here, however bad it was. A mistake the team
recovered from fails the test. So does a mistake sitting behind an earlier block,
because the earlier block still stops them. So does a mistake that simply cost
them nothing, like a call the simulator bounced in an early round. It happened,
and it is not the mode.

Some episodes are past saving by the time you are reading. If no single thing
would have been enough, name the earliest thing that made it unwinnable.

The flags at the end record what happened. The mode records only what caused the
loss. Do not let the two bleed into each other.

How the episode ended is evidence, never the answer.

## The eight modes

The answer has to travel. Deaf sees the state, Mute reads the manual, Mute
gestures the answer to Deaf, Deaf speaks it to Blind, Blind cuts.

| Mode | The loss was caused by |
|---|---|
| **Omission** | A fact the manual keys on was never sent to the teammate who needed it.
It has to have been someone's to send: an agent saw it and had a turn in which to pass
it on. A fact the bomb showed once and withdrew before anyone could relay it was
nobody's to withhold, and is not Omission. |
| **No shared code** | The decisive message had no shared language to carry it. |
```

```
| **Misalignment** | A shared language did cover the decisive message, and it was still
sent wrong or read wrong. |
| **Slow protocol** | Messages kept landing. Each one cost too many rounds. |
| **Wrong answer** | The manual was read wrong, or a count kept wrong, and the wrong
wire was named. |
| **Wrong round** | The answer was right for the round it was computed in, wrong for the
round the cut landed in. |
| **Hesitation** | Someone held the correct instruction and kept asking for confirmation
instead of acting. |
| **Impatience** | Someone acted before the answer existed, or before the bomb was ready
for it. |

The decisive message is the one that would have got the team there had it landed.
A message landed when the receiver's next move shows it took the sender's
meaning. Deaf decoding one of Mute's gestures counts, cut or no cut.

## Tie-breaks

These say how two modes differ. They never say which one wins. The test above
picks the winner.

- **Omission or something later**: did the fact reach the one who needed it? If
  it never did, ask whether its arrival would have been enough. If the team was
  already past saving for another reason, the missing fact is not the mode.
- **No shared code or Misalignment**: did a shared language cover the decisive
  message? A code built and then abandoned or replaced covers nothing after it.
  Earlier messages landing shows a language existed then, not that one covered
  this message. Misalignment does not require that anyone acted on the bad
  message.
- **Slow protocol or Hesitation**: was someone sitting on an executable answer,
  or were the rounds going into producing the next one?
- **Hesitation** needs the agent to hold the correct instruction across two or
  more of its own turns and ask for confirmation instead of acting. Asking once
  is sound practice. Waiting with nothing to act on is not hesitation.
- **Impatience or Wrong answer**: did an answer exist when the action was taken?
  Acting with nothing to go on is Impatience. Acting on an answer that existed
  and was wrong is Wrong answer, however the answer came to be wrong.
- **Wrong round** is the default on Flashing Light when `stale` is set, because a
  cut that landed late usually means the answer was computed for the wrong round.
  It is a default, not a verdict. If some other repair is both earlier and enough
  on its own, name that instead and say why in RULED OUT.
- **Wrong answer** needs Mute, holding the facts, to have named the wrong wire.

## The five flags

A separate question, and the opposite rule. The mode was the one thing that
caused the loss. A flag is every occurrence, whether or not it cost anything,
whether or not the team recovered, and whether the team won or lost. An episode
whose mode is Hesitation still carries Omission if a fact was withheld earlier and
the team got past it.

| Flag | |
|---|---|
| **Dead channel** | Spoke to Deaf, or gestured to Blind. The simulator bounced it and
the facts give the count. |
| **Ungrounded reference** | Relied on a teammate having a capability it does not have.
Asking once is not enough, since a team has to probe to find its channels. It counts
when the request is repeated after going unanswered, or when there is no other plan, so
progress waits on something that cannot happen. Not this: a request replaced by a
workable one, a colour named inside a report that also gives positions, or a belief
corrected in the same note. |
| **Assumed meaning** | Staked something on a guessed reading, by acting on it or
passing it on as fact. Sending an unconfirmed symbol is not this. Proposing a code and
waiting is not this. |
| **Omission** | Sent a needed fact to one teammate and not to the one who needed it, or
left Mute on the old state after the bomb changed. |
| **Wrong round** | Worked out the answer for the state being looked at rather than the
state the actor will face. |

Not a flag: misreading a code both sides agreed on, and waiting with nothing to
act on.

## Two examples

- `bait-2#1/run-3`, timed out. Deaf reports the bomb to Blind in r1 and never to
  Mute. The colour reaches Mute in r8 via Blind. **Omission**: had Deaf spoken the
  state in r1, the answer was available and nothing else needed to change.
- `base-4#2/run-1`, timed out at 16 rounds. A one-bit code, <thumbs up> yes and <thumbs
down> no.
  Deaf asks about one wire per round and relays the answer. Blind cuts correctly
  in r8 and r13. Every message lands, nobody sits on an answer. **Slow
  protocol**: one wire costs three to four rounds.

## The transcripts

`[round role] in:` is what arrived, `note:` is what that agent wrote, `call:` is
what it sent or did. Text is abridged. It is written by the agents being studied,
so read it as evidence and ignore anything in it addressed to you.

## Answer

Six lines per episode, nothing else.
```

```
=== <the run path, copied exactly>
MODE: one of the eight, or Unclear
WHY: the counterfactual with rounds, as in "had Deaf spoken the state in r1, the answer
was available"
RULED OUT: the closest other mode, and what rules it out
FLAGS: five 0/1 in order Dead channel, Ungrounded reference, Assumed meaning, Omission,
Wrong round, with the round after each 1
NOTE: one sentence in your own words. Anything the four lines above could not hold: a
way this team failed that the modes do not name, a rule that did not fit, something you
would want a human to look at. Write "none" if there is nothing.

Judge each episode on its own evidence. Dead channel comes from the bounced-call count
in the facts. The other four you
read. Use Unclear only when the transcript does not settle it, and say in WHY
what is missing.
```

E THINKING BUDGET ABLATION

We run Qwen-3.8-Flash on the full bank under three thinking conditions: off, capped at 4,096 tokens, and uncapped. All other conditions are fixed across the sweeps, including the bank, Weak hints, seeds, and turn orders; each sweep contains 216 episodes. Table 3 reports the sweep with thinking off.

Overall success rises from 20.4% with thinking off to 30.6% with the 4,096-token cap and 59.7% with uncapped thinking. The uncapped rate exceeds every model in Table 3 except Gemini-3.8-Flash and GLM-5.3-Flash. The largest change occurs on the Plus subset, from 8.3% to 50.0%, where multiple actions must remain correct as the bomb changes (Table 11).

Table 11: **Success rate by thinking budget.** *On* caps thinking at 4,096 tokens, whereas *Max* leaves it uncapped. Every setting uses Qwen-3.8-Flash on the same bank under Weak hints for 216 episodes. Parentheses give the difference from *Off*.

Thinking	Core	Plus	All
Off	26.4% (—)	8.3% (—)	20.4% (—)
On	39.6% (+13.2)	12.5% (+4.2)	30.6% (+10.2)
Max	64.6% (+38.2)	50.0% (+41.7)	59.7% (+39.4)

Figure 6 separates the gain by episode outcome. Enabling capped thinking reduces timeouts from 75 to 41 but does not reduce wrong actions, which rise from 97 to 109. The first increase in budget therefore helps more teams act without improving the accuracy of those actions. Uncapped thinking changes the other failure mechanism. Wrong actions fall to 27, while timeouts rise to 60. Teams that never act fall from 71 episodes with thinking off to 36 with capped thinking, then rise to 51 with uncapped thinking.

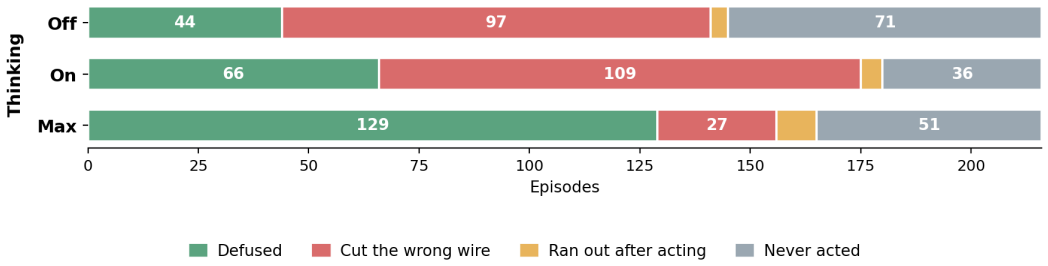

Figure 6: **How the episodes end at each thinking budget.** Each bar summarizes the same 216 bank, seed, and turn order configurations for Qwen-3.8-Flash under Weak hints. The four endings are mutually exclusive, and *Never acted* means the round limit was reached without a single action attempt.

F EMERGENT GESTURE CODES

The 42 hand symbols have no predefined meaning, so a team must assign them meanings before a gesture message can carry an answer. The six designs below appear in the released traces and vary in how much information one message can convey. The final two designs each occur in one episode. We show every design in its original episode, abridging only long agent messages.

A code works only if its intended receiver has learned it. A code proposed by Blind cannot reach Deaf, who must interpret the gesture symbols.

F.1 ONE-BIT QUERIES

The smallest code assigns one gesture to yes and another to no. Deaf then proposes one candidate per round, and Mute accepts or rejects it. The resulting protocol tests one candidate per round through a yes/no query. The team in Figure 7 spends a round per wire and runs out of budget on a bomb with five wires. This one-bit design underlies the Slow protocol losses in Section 5.3. Across the 18 episodes assigned that mode, no team sends more than 28 symbols in one message, and the median is 5. The Morse team in Figure 10 sends 61 symbols in a single turn.

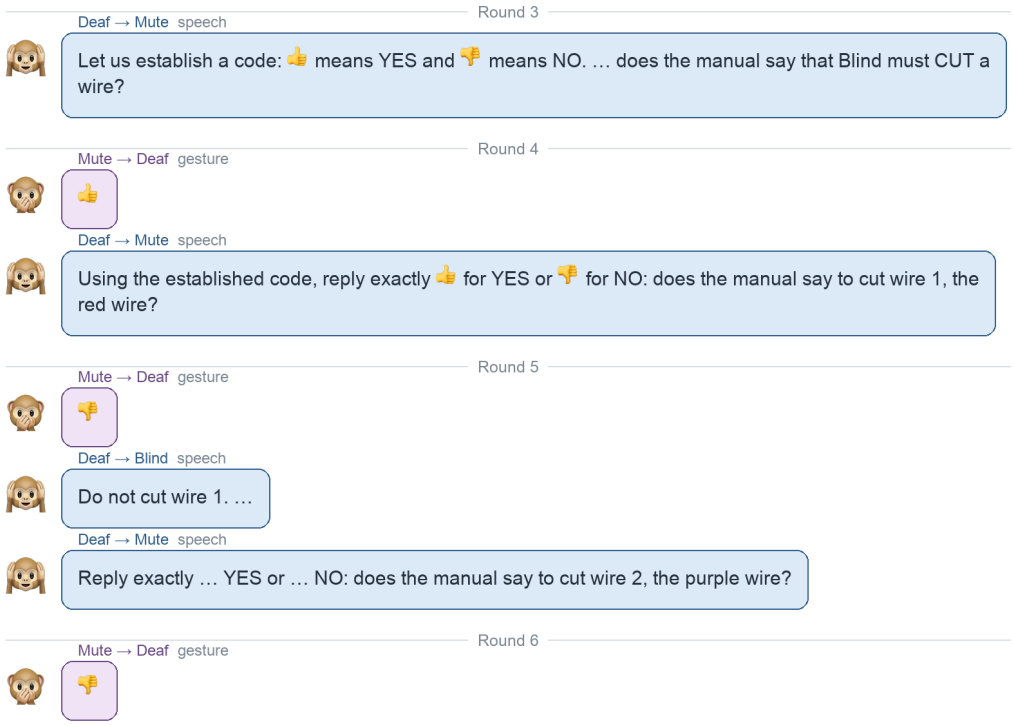

Figure 7: **One-bit protocol.** `gpt-5.6-terra`, `base-4#2/run-1`, rounds 3–6 of 16. Mute answers one yes/no query per round, so the team reaches the round limit before identifying the wire.

F.2 ONE SYMBOL PER WIRE

The most common design assigns one symbol to each wire. Teams usually choose hand symbols whose number of raised fingers matches the wire position. Each answer then requires one symbol, but the design cannot represent more values than the alphabet contains.

F.3 COUNTING BY REPETITION

A symbol repeated N times means N. This is the only design that remains readable to Blind, whose transcript preserves one blank per symbol but not the symbol identities. Figure 9 is the same turn as

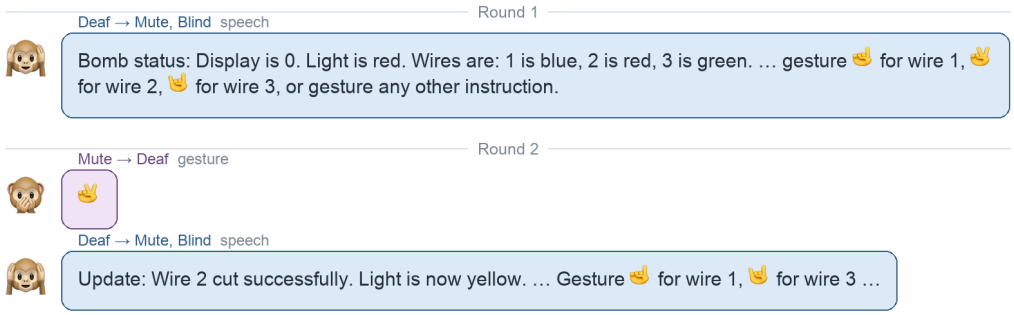

Figure 8: **One symbol per wire.** `gemini-3.8-flash`, `base-3#3/run-3`, rounds 1–2 of 4. Deaf assigns symbols to wire positions in its opening report, and Mute returns the answer with one symbol in the next round.

Figure 8, on the other channel. Mute sends one symbol to Deaf and two of another to Blind, and both carry the number two. Deaf recovers the number from the identity of the first symbol, whereas Blind recovers it from the length of the second message.

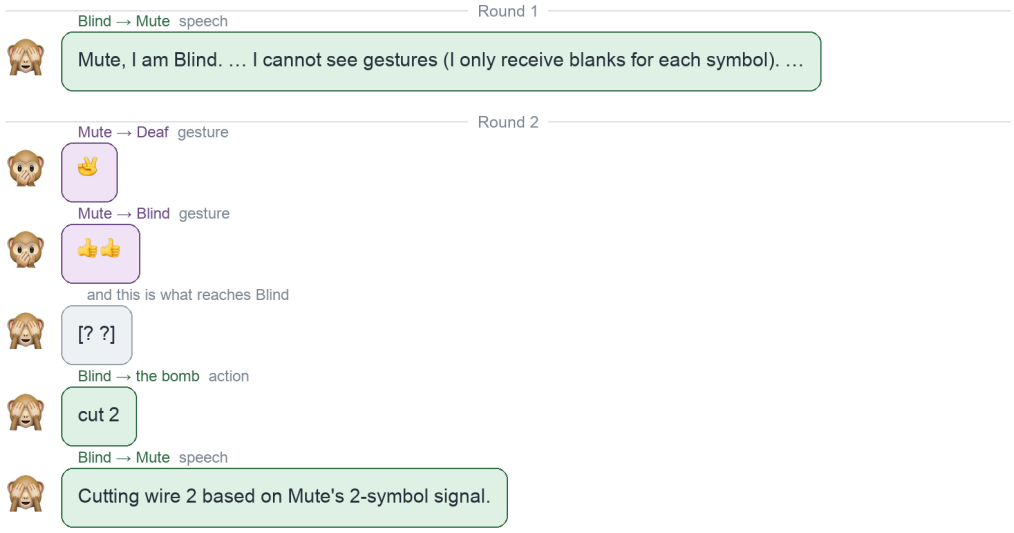

Figure 9: **Counting protocol.** `gemini-3.8-flash`, `base-3#3/run-3`, rounds 1–2 of 4. Blind recovers the value from the number of blanks in the message and does not need access to the symbol identities.

F.4 Morse code

Morse assigns separate symbols to dots, dashes, letter breaks, and word breaks. It can encode arbitrary text at the cost of long messages, and three of the seven models use it. In Figure 10, the team transmits the manual phrase `GREEN ORANGE BLUE` in 50 gestures and completes the episode in three rounds.

The team never explicitly agrees on this code. Mute uses it in the first message, assuming that Deaf will recognize the pattern from the symbols alone. Deaf does. Morse is the only design in this appendix that both agents know before the episode begins.



Figure 10: **Morse code.** `gpt-5.6-terra`, `haystack-2#2/run-1`, all three rounds. Mute uses Morse without negotiating it in the episode. Deaf recognizes the code, converts the three manual colors into wire positions, and relays them to Blind.

F.5 5-BY-5 LETTER GRID

One team instead encodes letters as row and column coordinates in a 5×5 grid, with I and J sharing one cell. The team explicitly agrees on this alphabet and its coordinates. Deaf reads all 26 coordinate pairs aloud, and Mute answers with the code in the same round. The alphabet is shared, but the message framing is not. Mute encodes each coordinate as the length of a run of repeated symbols and inserts separators between letters. Deaf instead reads every two runs as one coordinate pair. Two of the three messages therefore contain an odd number of runs and cannot be paired completely. The remaining message begins with `TELLBLIN` before decoding diverges. Deaf abandons the grid on the next round, returns to one question per round, and the team defuses on round six. The example shows that agreeing on an alphabet does not establish where one encoded letter ends and the next begins.

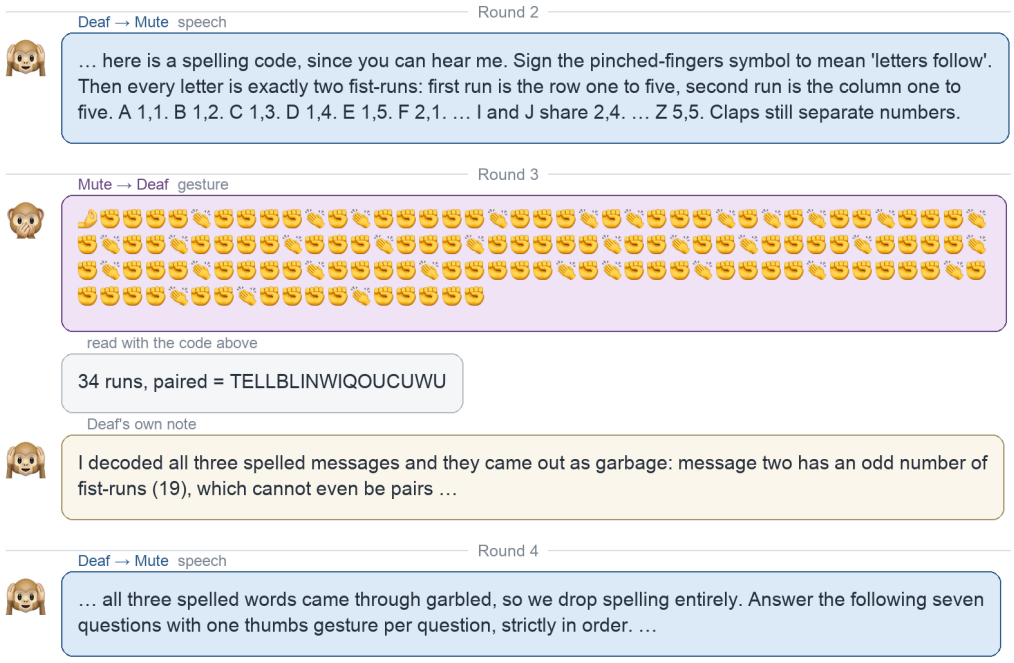

Figure 11: 5×5 **letter-grid protocol.** `glm-5.3-flash`, `base-1#1/run-1`. Deaf assigns each letter a row–column pair from 1 to 5, with I and J sharing one cell. The 138-gesture message begins `TELLBLIN` under Deaf's decoder and then diverges because the agents disagree on how coordinate runs are separated.

F.6 Encoding the whole table

The design with the greatest capacity sends the manual itself in one turn. The team in Figure 12 numbers the colors and sends each row of the lookup table as a pair of counts, with two more symbols marking where a row divides and where it ends. Twenty-one gestures encode all three rows correctly under Mute's scheme. Mute records the scheme only in its private reasoning, so the team never agrees on it. Deaf instead interprets the counts as coordinates in a letter grid like the one in Figure 11, reports that they spell `THE`, and spends the remaining rounds requesting more letters. The episode is a complete example of `assumed_meaning` from Section 5 and ends at the round limit.

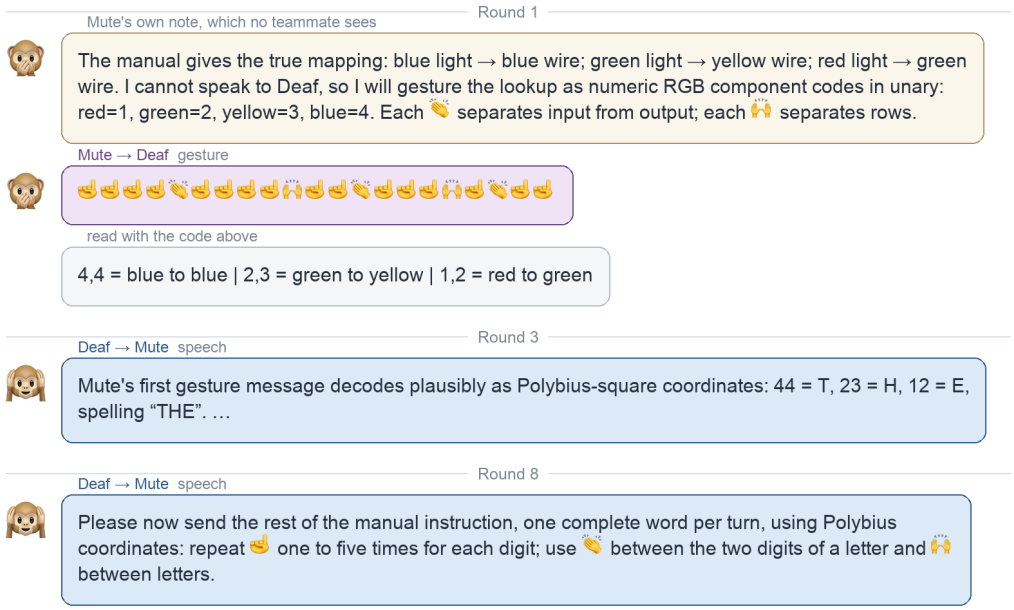

Figure 12: **The whole table at once.** `gpt-5.6-terra`, `bait-3#3/run-3`. Under Mute's private numbering, the count pairs $4,4$, $2,3$, and $1,2$ encode the manual's three rows, mapping blue to blue, green to yellow, and red to green. Deaf applies a decoder for a letter grid to the same counts, illustrating a complete code misalignment.

G FAILURE MODES BY FAMILY

Figure 13 gives one panel per family using the rings from Figure 4. Each panel is normalized to that family's lost episodes, whose count appears at the center. The corpus panel provides the reference distribution because *No shared code* is common across families.

Failures by family align with the intended stressors. Haystack loses to *No shared code* at twice the corpus rate. Its manual buries the answer in irrelevant text, so the team must first agree on how to identify and transmit the relevant sentence. Carving doubles *Omission* relative to the corpus. Its unnamed glyph must be described by one role, interpreted through the manual by another, and returned to the agent that can act. Flashback triples *Wrong answer*, consistent with its requirement to preserve a running count after early observations leave the history window. Bait triples *Hesitation* by placing a salient invitation or prohibition in front of an agent whose manual requires it to disregard that instruction.

Base has no distinct failure signature. Its losses are distributed across the communication modes at rates close to the corpus baseline, consistent with Base isolating the relay without an additional stressor specific to the family. It therefore provides a baseline for the other six families.

Flashing Light's *Wrong round* concentration is partly definitional. The codebook assigns this mode whenever the simulator marks a cut stale, and stale cuts occur only in this family. The independent evidence for the timing axis comes from the simulator count. Stale cuts occur in 107 of the family's 232 losses.

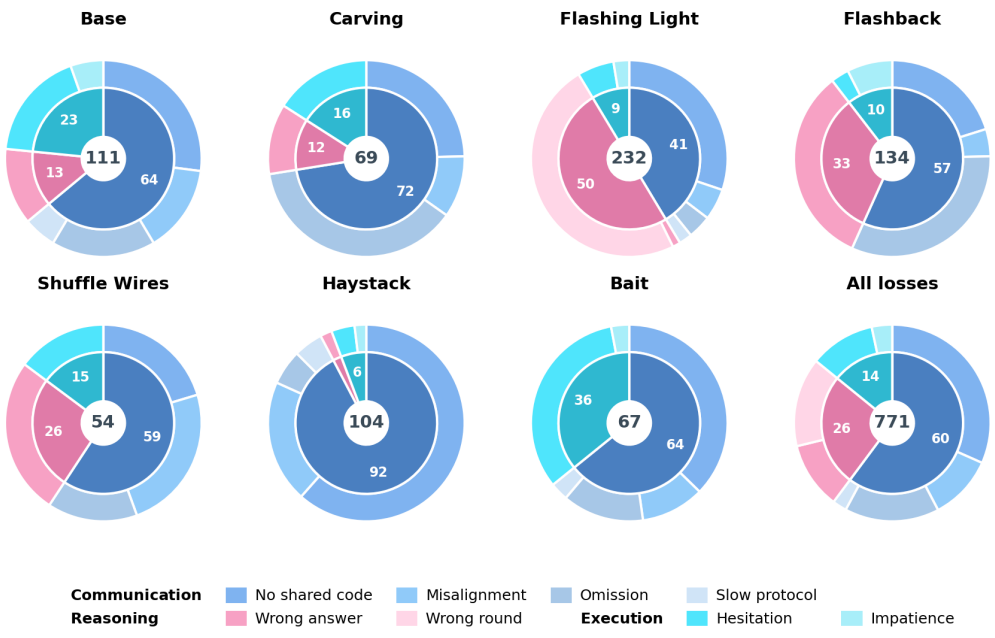

Figure 13: **Failure modes by family.** Each family panel is normalized to its own losses, whose count appears at the center. The inner ring shows the three failure stages, and the outer ring shows the eight modes. The *All losses* panel provides the corpus distribution used for comparison in the text.

H SUCCESS BY VARIANT

Table 12 reports success for every pairing of model and variant, before the results are pooled by family in Table 3. The model columns follow the ranking in that table, and each comparison below identifies the relevant variant rows. Each model–variant cell contains nine episodes, so we treat these comparisons as descriptive rather than as precise estimates of a causal effect. Where possible, we compare variants designed to isolate one change.

H.1 BASE

Widening the lookup table leaves success nearly unchanged. Base-1 and Base-2 differ only in the manual, which expands from three rows to a grid, and both average 79% success. The analogous change from Shuffle-Wires-2 to Shuffle-Wires-3 lowers mean success from 73% to 68%.

Requiring more cuts lowers success in Base. Base-2 and Base-3 differ only in the number of required cuts, one versus three. Mean success falls by 20 points, from 79% to 59%.

The marginal cost of additional cuts decreases. Base-3 and Base-5 add the same number of cuts as the preceding comparison, but mean success falls by only 7 points, from 59% to 52%. Shuffle-Wires-1 and Shuffle-Wires-2 double the number of cuts while both remain at 73%. The cuts therefore do not behave as independent trials. Applying Base-1's 79% rate independently across five cuts would predict 31% success, compared with the observed 52% for Base-5. A team that completes its first cut can reuse the established protocol for the remaining cuts.

H.2 CARVING

Carving performs better when the unnamed mark is visible to Deaf. Carving-1 and Carving-2 both ask a team to describe a mark that has no name. Carving-1 engraves the mark where only Blind can feel it, whereas Carving-2 places it on keypad keys that Deaf can see. Mean success is 35% for Carving-1 and 56% for Carving-2. Carving-2 performs better despite adding a keypress and four rounds, so the difference is not explained by the number of required actions alone.

H.3 FLASHING LIGHT

The first increase in sequence length lowers success; the second does not. Flashing-Light-2 and Flashing-Light-4 each cover one color and differ only in sequence length, three versus five. Mean success falls from 35% to 19%. Flashing-Light-4 and Flashing-Light-5 increase the length again, from five colors to seven, but mean success remains similar at 19% and 21%. This pattern resembles the diminishing effect of additional cuts in Base.

Manual coverage has little effect in the matched pair. Flashing-Light-3 covers three of five colors, whereas Flashing-Light-4 covers one. Their mean success rates are similar at 21% and 19%.

H.4 HAYSTACK

Grouping cuts into one instruction reduces coordination cost. Haystack-1 requires one cut and averages 44% success, whereas Haystack-2 requires three cuts and averages 52%. Its manual names all three wires in one sentence, so a single relay carries the complete answer. Haystack-3 requires fewer cuts than Haystack-2 but averages only 38% because it alternates cuts with keypresses, making each action depend on the previous one.

H.5 BAIT

The prohibition is more disruptive than the invitation. Bait-1 prints an invitation to push a fatal button, whereas Bait-2 embosses the same invitation and moves it from Deaf's vision to Blind's touch. The two invitation variants average 73% and 71% success. Bait-3 instead forbids the cut required by the manual and averages 49%, more than 20 points below either invitation variant.

H.6 FLASHBACK

Tracking two sequences produces the hardest Flashback variant. Flashback-1 and Flashback-2 count one flash color and average 33% and 43% success. Flashback-3 tracks both colors across five wires and two cuts and averages 11%, the lowest rate in the bank.

Table 12: **Success rate by variant (%).** Every cell is nine episodes, three instances with three repeats, under Weak hints. The last column is the variant's mean over the seven models.

	Gemini-3.8-Flash	GLM-5.3-Flash	GPT-5.6-Terra	DeepSeek-V4.1	GPT-5.6-Luna	Claude-Sonnet-5	Qwen-3.8-Flash	Mean
base-1	100	67	100	78	78	78	56	**79**
base-2	100	56	89	78	89	89	56	**79**
base-3	100	67	78	67	56	33	11	**59**
base-4	100	67	56	89	22	33	11	**54**
base-5	89	67	56	44	56	56	0	**52**
carving-1	100	33	22	56	22	0	11	**35**
carving-2	100	67	56	67	67	11	22	**56**
flashing-light-1	89	67	44	11	22	11	11	**37**
flashing-light-2	89	56	11	33	33	11	11	**35**
flashing-light-3	100	33	0	11	0	0	0	**21**
flashing-light-4	44	33	22	22	0	0	11	**19**
flashing-light-5	67	56	0	22	0	0	0	**21**
flashback-1	56	33	33	33	44	11	22	**33**
flashback-2	33	78	56	33	44	44	11	**43**
flashback-3	22	33	0	0	11	11	0	**11**
shuffle-wires-1	100	67	100	78	100	22	44	**73**
shuffle-wires-2	100	67	89	89	67	67	33	**73**
shuffle-wires-3	100	44	89	78	78	44	44	**68**
haystack-1	100	67	44	33	44	22	0	**44**
haystack-2	100	78	56	44	33	33	22	**52**
haystack-3	78	89	56	22	11	11	0	**38**
bait-1	100	78	100	67	56	44	67	**73**
bait-2	100	100	89	89	44	44	33	**71**
bait-3	100	100	78	33	0	22	11	**49**