

Jarvis: A Proactive Speech Agent for Multi-Party Conversations

Seunghyun Oh^{1*}, Hirotaka Hiraki^{2*}, Shuyue Stella Li¹, Yulia Tsvetkov¹,
Shyamnath Gollakota¹

¹Paul G. Allen School of Computer Science and Engineering, University of Washington, WA, USA

²Graduate School of Interdisciplinary Information Studies, The University of Tokyo, Bunkyo, Tokyo, Japan

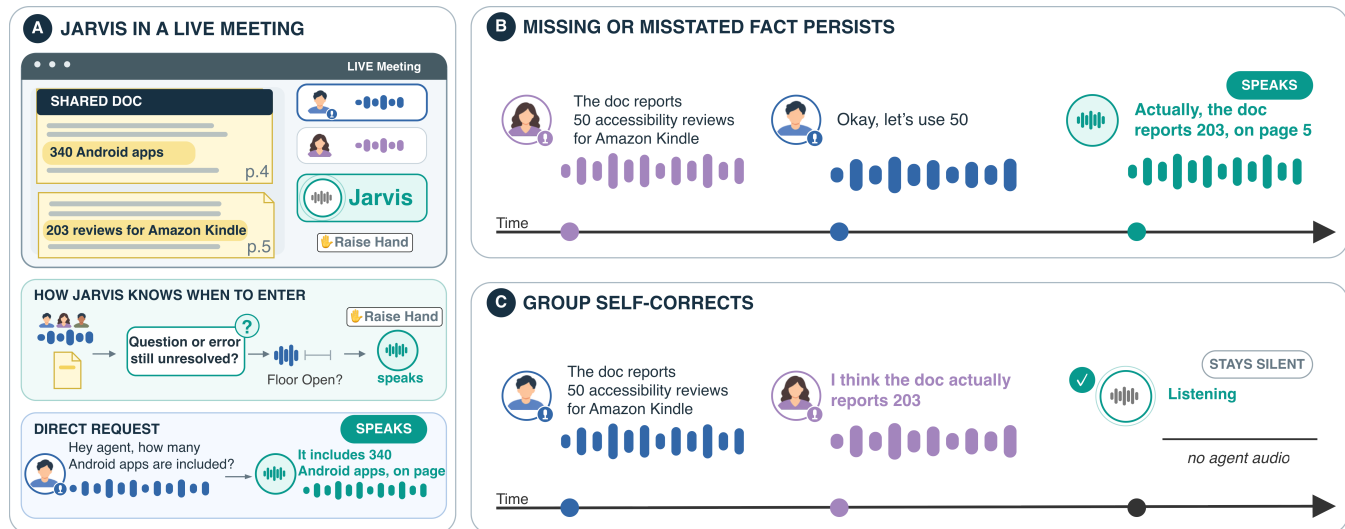


Figure 1: Jarvis in a live multi-party meeting. (A) Jarvis joins the meeting as a participant, follows the conversation against a shared document, and can answer direct questions. For proactive intervention, it has two modes: 1) raises its hand to seek permission or 2) waits until no participant holds the floor and speaks directly. (B) If a document-grounded missing or misstated fact remains unresolved, Jarvis speaks to correct it and cites the relevant page. (C) If the group corrects the error first, Jarvis remains silent.

ABSTRACT

Speech agents are reactive and dyadic: they speak when spoken to, and to one person at a time. We ask what it takes for a speech agent to instead take part in a conversation among several people and speak up only when it can help. We introduce Jarvis, a real-time proactive speech agent that audibly participates in multi-party human conversations. Grounded in a document shared beforehand, Jarvis follows the discussion and intervenes when the group misses or misstates a fact and does not correct itself within a few turns. We make three contributions: a problem setting based on epistemic breakdowns that makes proactive intervention measurable, realized as CHI-180-proactive, a synthetic multi-party dataset seeded with known gaps, errors, and self-corrections; a proactive backbone that harnesses a small, open-weight model with deterministic checks and grounds every claim in a source sentence; and interaction techniques for taking the floor in live speech and showing the cited evidence on screen. On CHI-180-proactive, Jarvis is correct on most events it addresses and stays silent 97% of the time when the group resolves an issue itself. A live study with 23 participants confirms these trends with real-time interventions.

1 INTRODUCTION

Imagine a meeting where everyone is working with a 15-page contract, but nobody remembers every detail. Someone asks what the contract says about data retention. Nobody knows, so the question gets pushed to the next meeting. Later, someone misstates the sample size, and the discussion starts building on the error. Now imagine an agent that has read the contract, answers questions on the spot, and listens to the conversation. When someone misses or misstates a fact, it first gives the other participants a chance to catch and correct it. If they do not, it speaks up. Such a *proactive* speech agent could be useful in lab meetings, diligence calls, design reviews.

Today’s speech agents cannot play this role for three key reasons. First, they are primarily reactive: from smart speakers to LLM voice modes, they speak only when addressed, e.g., “Hey Alexa.” But some of the most useful moments for help are ones no one thinks to ask for, such as when a fact has been misstated without anyone realizing it. Second, they are designed for dyadic interaction [14, 71]: their turn-taking, endpointing, and barge-in mechanisms assume a single user is addressing the agent, so an utterance is generally treated as a turn that warrants a response. A human-conversation agent must work differently. It is listening, but without being explicitly addressed, it must decide when the conversation needs intervention, give participants a chance to correct themselves, and then

*Both authors contributed equally to this research.

enter an ongoing conversation without derailing it. Third, it must remain in the background and keep humans at the center, speaking only when necessary rather than competing for attention. Neither reactive assistants, dyadic dialogue systems, nor in-ear whispering assistants [5, 88] provide the interaction model or evaluation framework needed for this setting.

We introduce Jarvis, to our knowledge, the first real-time proactive speech agent that audibly participates in multi-party human conversation. Designing and evaluating this requires addressing three core research questions:

- **RQ1:** *Should the proactive agent speak, and what should it say?* At each turn, the agent must decide whether to say anything and, if so, what. Unlike a reactive assistant where humans ask for explicit assistance, the multi-party conversation can continue without any intervention from the proactive speech agent. So the agent has to identify the rare moments when its intervention actually helps and is not considered a distraction by the human participants.
- **RQ2:** *When and how should it take/hold the floor?* Unlike text-based proactive agents, where a chat message can be skimmed, deferred, or ignored, a spoken intervention consumes shared airtime and displaces whoever might speak next. The agent can neither cut across someone mid-sentence nor sit on an important point until the moment has passed. And if a participant does interrupt it, the agent must decide whether to keep the floor or yield, and what to do with the intervention it was partway through delivering.
- **RQ3:** *How do we measure whether proactive intervention worked?* Proactive behavior has been difficult to evaluate because open-ended conversation has no answer key. Existing human-conversation datasets do not fill this gap: AI speech agents should be secondary participants in human conversations, taking a back seat to the humans, yet no datasets capture this setting. Without a well-defined problem setting, whether an interjection was correct, well-timed, or needed at all becomes a matter of opinion. We therefore need problem settings and datasets in which both correctness and timing are objectively decidable, allowing us to evaluate what an agent says and when it says it against something firmer than opinion.

Our paper addresses these questions and makes the following conceptual and technical contributions.

- **Multi-party setting and dataset that make proactive intervention measurable.** Proactive behavior is hard to evaluate because open conversation has no answer key. To make proactive intervention decidable by construction, we target *epistemic* breakdowns rather than social ones. Specifically, Jarvis identifies moments where the multi-party group is missing or misstating something checkable. To do this, we ground each conversation in a shared document (e.g., financial report or academic paper) that is shared apriori with Jarvis before the start of the conversation. We also introduce CHI-180-proactive, a dataset created from 180 CHI papers, each paired with a synthesized multi-party discussion into which we plant 24 known events like knowledge gaps, errors, self-corrections, and distractors. In total the dataset has 4,320 events over 50,400 turns, each annotated with a fixed correct action. This turns proactive intervention into a detection task

with measurable accuracy and timing, and we will release it as a research benchmark for proactive conversational agents.

- **A deterministic harness of small models that decides whether to speak and what to say.** We harness a small, open-weight language model through several stages with deterministic checks, so the whole system can be self-hosted and no single model output is trusted on its own. At each turn one model classifies whether the discussion is missing or misstating something the document can settle and another checks whether the participants have already corrected it. A deterministic gate reconfirms and speaks only once a problem has persisted, so the agent defers to human self-repair before intervening. Every spoken claim is grounded in a document sentence rather than the model’s own knowledge (§3.2). Our evaluation shows that in this document-grounded setting our harness of small models is enough and is fast enough to run in real time.
- **Interaction techniques for taking and holding the floor.** Our goal is for Jarvis to join a live conversation like a considerate participant. To achieve this, we explore three interaction methods. First, rather than speaking as soon as it has something to say, it queues the intervention and waits for an opening; if someone begins talking while it is speaking, it stops at once and decides whether to replay, resume, or drop what it was saying. Second, for groups that want tighter control, it instead raises its hand and speaks only once someone verbally gives it the floor. Third, alongside the spoken answer it shows visual evidence, placing the cited source on screen so participants can verify a claim rather than take it on the agent’s word (see §3.3).

Rather than build a Wizard-of-Oz prototype, we address the underlying systems and latency challenges, deploy a proof-of-concept Jarvis on Zoom as an additional participant, and run it live in multi-party conversations.

Our key findings are as follows:

- (1) Our QWEN3.6-35B-A3B harness, with 3B active parameters, is competent in our document-grounded setting: on CHI-180-proactive, Jarvis provided correct or partially correct information with 95-97% accuracy.
- (2) Jarvis defers to the group and intervenes when needed. After participants self-correct ($N=720$) or answer their own question ($N=720$), it stays silent in 97% of cases in both. When there is an information gap ($N=1800$) or a document-specific incorrect statement ($N=720$), it intervenes in 91% and 77% of cases; the latter is lower because many incorrect statements import a fact from another document, which Jarvis cannot verify against its own source and hence stays silent. When it does intervene, it acts within 2-3 turns of the event in 98% of cases.
- (3) In live Zoom sessions with 23 participants, where Jarvis must also cope with transcription errors from the speech pipeline, these results carry over to real people: 85% of its interventions arrived within two turns of the problem, and 163 of 164 carried correct or partially correct information. Together, Jarvis and the participants resolved 74% of planted errors, 72% of spontaneous errors, and 83% of information gaps.

- (4) When asked to choose, participants preferred an agent that raises its hand before speaking to one that speaks directly, and overwhelmingly preferred having the cited source shown on screen (21 of 23), which let them check claims and trust the answers.

Taking a step back, we show that real-time proactive participation in spoken conversation can be designed and carefully measured against ground truth, establishing a research foundation for future proactive speech agents.

2 RELATED WORK

Proactive Agents and the When-to-Speak Problem. Mixed-initiative systems [20] and just-in-time agents such as the Remembrance Agent surface relevant information from observed activity [64], but deliver it on screen rather than into the conversation. Interruption research showed that poorly timed interventions carry cognitive costs [29, 45] and developed models for predicting interruptibility and deferring notifications [21–23]; these focus on individual notifications, not spoken contributions. More recent work on proactive voice assistants finds a similar “proactivity dilemma” in balancing helpfulness and intrusiveness, particularly in multi-party settings [52, 80], but remains largely formative and Wizard-of-Oz, studying user attitudes rather than autonomous spoken intervention.

LLMs have turned deciding whether, when, and what to contribute into a modeling problem [11, 12, 78]. Task agents infer needs from desktop or mobile activity [43, 55, 70], ContextAgent from egocentric sensors [77], and CollabLLM takes initiative during dialogue [73]; all, however, assist a single user rather than compete with humans for the floor. Inner Thoughts scores covert thoughts to select moments to contribute in multi-party text chat [40], DiscussLLM predicts whether to respond or stay silent each turn [56], and ProMediate selects intervention moments in multi-party negotiation [42]. These systems operate in turn-based text, where the platform effectively gives the agent its turn, and none grounds intervention in whether the group is missing or misstating a verifiable fact. ProVoice-Bench moves to audio [75], but tests reactions to pre-defined sounds rather than the content of an unfolding discussion. SocialMind displays suggestions on AR glasses [76], while LlamaPIE whispers short cues to one wearer [5]. Both provide private cues invisible to the group, rather than speaking into the conversation itself.

Our work builds on this prior work, but shared speech introduces constraints that text does not. A chat message, AR overlay, or private whisper can be skimmed, deferred, or ignored while the conversation continues. In contrast, a spoken intervention consumes shared airtime and displaces whoever might speak next.

Turn-Taking, Interruption, and Full-Duplex Spoken Dialogue. Conversation analysis shows that turn-taking is tightly organized [36, 66], with self-repair preferred over other-repair [67]. We operationalize it by making the agent a deliberately *dispreferred other-repairer* that waits through the self-repair window. Computational models predict when a transition is possible [15, 16, 28], not whether the agent should speak. Concurrent work extends turn-taking to multi-party speech under floor competition [48], and full-duplex dyadic models listen and speak at once [6, 14, 38, 71],

but both predict floor availability, when a role may take a turn, assuming each utterance is addressed to the agent. While provide the mechanics for how to enter, including VAD-gated delivery and barge-in yielding, but not whether to enter in multi-party settings, which requires content-level evidence that the group is missing or misstating something and has not already corrected it.

Agents in Meetings and Group Conversations. Meetings have long been used to study conversation understanding, from multi-party corpora [4, 30] to meeting summarization and assistant benchmarks [27, 85] and LLM delegates that attend on a user’s behalf [26]. These systems either process the meeting afterward or respond to queries; they do not decide on their own that something said was wrong. Other work puts agents into group conversations as moderators, facilitators, or devil’s advocates [2, 8, 32, 33, 44, 54], but in *text*, and mainly to shape participation or opinions. Recent studies examine when agents should participate in groups [24, 79, 86], highlighting the same tradeoff we face: speaking can disrupt, while silence can miss an opportunity to help. But these systems are formative or user-controlled, rather than autonomous spoken agents, and their open-ended settings provide no ground truth.

Grounded Generation and Verifiable Attribution. Retrieval-augmented generation grounds LLM outputs in sources [37], while attributed generation links claims to passages [17, 46, 74, 83] and verification pipelines check whether evidence supports those claims [18, 47, 57]. These methods largely assume a *written* answer to an *explicit* question, checked offline by the person who asked.

Evaluating Proactive Conversational Behavior. Proactive-agent benchmarks measure task anticipation [39, 55, 70], but focus on desktops and datastores rather than conversation. LLM-judged testbeds [72] scale evaluation while user studies measure helpfulness and intrusiveness [5, 76, 80], capturing how interventions felt. ProMediate is closest to our setting, evaluating intervention timing in multi-party negotiation [42], but uses turn-based text with LLM-simulated negotiators. The simulation gives the mediator its turn, so claiming the floor is not an issue, and a negotiation problem setting has no objective answer key for judging an intervention. Our work targets *epistemic* rather than social breakdowns. We seed controlled documents with known facts, predictable gaps, and distractors, turning proactive intervention into a detection task with measurable accuracy and timing error.

3 SYSTEM DESIGN

As shown in Fig. 2, Jarvis has three modules, each handling a different part of the interaction: who a turn is from, whether there is something worth saying and why, and when and how to say it. The Speech Processing module (§3.1) converts live meeting audio into timestamped, speaker-attributed text for each turn, detects the wake word, and plays the agent’s responses back into the meeting. The Proactive Backbone model (§3.2) takes the document shared before the meeting and, for each turn from the Speech Processing module, decides whether the agent should contribute and, if so, grounds the response in an exact passage. The Interaction module (§3.3) manages delivery: it waits for an opening rather than talking over the group, yields when someone speaks, and, depending on

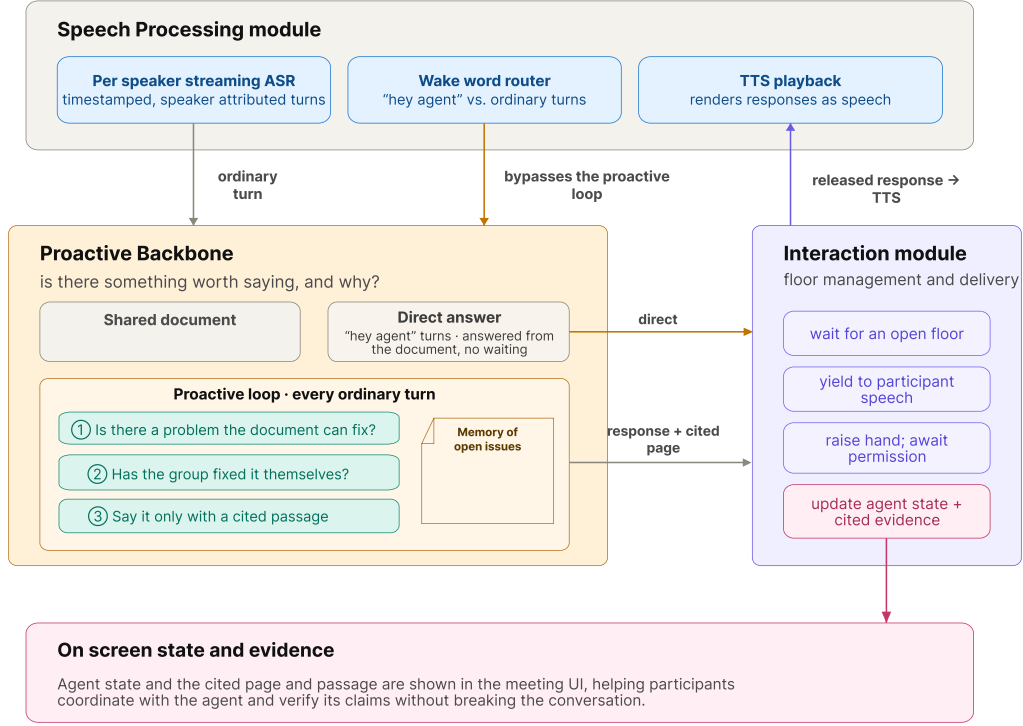


Figure 2: Jarvis system architecture. It has three components: speech processing module, proactive backbone and interaction module.

the mode, either raises a hand for the group to call on or displays the cited page so participants can verify the claim.

3.1 Speech Processing Module

It provides three interfaces. First, to the proactive backbone, it sends timestamped, speaker-attributed turns represented as $\langle \text{speaker, text, start time, end time} \rangle$, tagged as either *direct* (addressed to the agent) or *stream* (participant-to-participant talk). Second, to the interaction module, it sends pre-lexical acoustic events such as speech onset, sustained speech, and silence. Finally, it synthesizes approved responses from the proactive backbone agent and plays them back into the meeting as speech. This is implemented via five sub-components, details of which are in §A.

3.2 Proactive Backbone: Deterministic harness of small models

Rather than rely on a single model, we design a deterministic harness that break the task into multiple steps. Each step uses the same language model for the part that requires language understanding and deterministic logic to check its output. This prevents any single model decision from triggering an intervention on its own. The backbone has four stages (a monitor, a self-repair check, a grounded generator, and a direct answerer) and a deterministic gate (Fig. 3(b)). All stages use the same document, which we split into numbered sentences (§3.2.1). Every stage refers to the document only via sentence IDs.

Each incoming turn other than backchannels (“yeah,” “uh-huh”) is first appended to the transcript. Two model stages run in parallel. The *monitor* (§3.2.2) classifies the discussion as either missing or misstating something the document can resolve, or as ordinary conversation that needs no intervention. The *self-repair check* (§3.2.3) looks at the recent conversation and asks whether the participants have already fixed a problem that the monitor flagged on an earlier turn. A deterministic test confirms that the correct value actually appears in the recent turns before clearing the problem.

The two outputs feed a deterministic *gate* (§3.2.4), which decides whether Jarvis should speak. The gate keeps a ledger of unresolved problems and intervenes only when a problem persists across turns without being resolved. This gives the group a chance to catch and correct its own mistakes before Jarvis steps in. When the gate fires, *grounded generation* (§3.2.5) produces the response. It takes the identified problem and the specific document sentences that support the response as input, and generates the response grounded in the document; a response that cannot be tied to a document sentence is not spoken. A separate *direct answering* stage (§B) handles questions explicitly addressed to Jarvis by wake word. These questions bypass the monitor and gate and receive a document-grounded answer.

3.2.1 Shared Document Representation. We take the full document text, strip markdown headings and figure apparatus, and split it into sentences on “. ! ?”, dropping fragments shorter than five tokens. Each surviving sentence is given an integer ID and listed as a numbered block: [S0] sentence 0, [S1] sentence 1, ..., [Sn] sentence

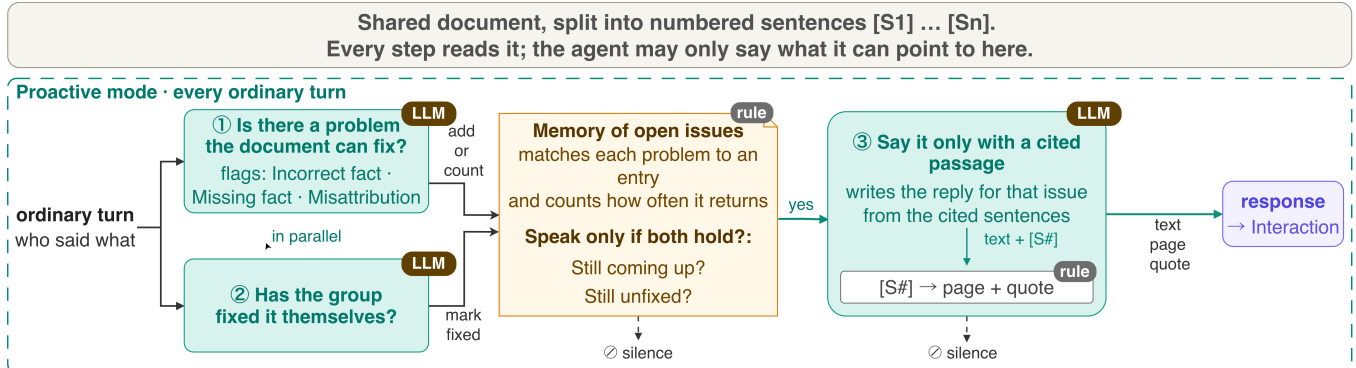


Figure 3: Our proactive backbone is a deterministic harness of small models. For each turn, the monitor identifies problems that can be addressed using the shared document, while the self-repair check determines whether the group has already resolved them. The two stages run in parallel and update a memory of open issues. A deterministic gate permits an intervention only when an issue persists across turns and remains unresolved. If these conditions hold, Jarvis generates a response supported by cited sentences from the shared document. The cited sentence IDs are resolved to their source page and passage before the response is sent to the Interaction module. Jarvis remains silent when the gate conditions are not met or a valid citation cannot be produced.

n. This block is assembled once and prepended to every stage’s system message, so all stages index the document by the same IDs.

A stage that finds a relevant fact reports the ID of the sentence it came from, and not the text. The server resolves each ID back to its verbatim sentence and drops any ID it cannot resolve. A model can therefore point at the document but does not paraphrase it. This design ensures that the citation is more trustworthy since the spoken claim is backed by source text the system retrieved, not by text the model produced. We cap the IDs at three per report for the monitor and the self-repair check. Sharing one representation also has a practical benefit. Because all four system messages begin with the identical document block, the serving engine prefills it once per turn and reuses the cache across stages; only the shorter policy text that follows differs between stages.

3.2.2 Monitor. The monitor performs three tasks at each turn: it classifies the discussion’s status, extracts a *claim* naming the point under discussion, and, if the status is actionable, retrieves the related document sentence IDs. It runs at a low temperature (0.1). Detection, claim identification, and grounding are deliberately fused in one component, which is what lets the monitor output a checkable judgment (“this contradicts [S7]”) rather than an opinion (“something seems wrong”). See Table 9 for the full prompt.

Conversation status taxonomy. The monitor assigns each new turn to one of six statuses.

- *Incorrect fact*, where a participant asserts something the document contradicts.
- *Missing fact*, where the participants need a specific fact the document states but no one has supplied.
- *Misattribution*, where a participant attributes a claim to the document (“in this study...”, “they found...”) that the document does not support.
- *On track*, accurate and consistent with the text.
- *Off topic*, concerning logistics, tooling, or side chat that asserts nothing the document could contradict.

- *Unanswerable from the document*, where participants are stuck or unsure about a point, the document is about that topic, but does not actually state the specific detail they need.

The first three statuses are actionable, meaning that they name a problem the agent could intervene on, and the ledger tracks them. The other three call for no intervention and open no problem.

We distinguish *misattribution* from being merely *off topic*. This matters because a mention of another document warrants silence, whereas a false claim about *this* document does not. Similarly, we distinguish a *missing fact* from an *unanswerable* question. This allows the monitor to report a missing fact only when it can point to the exact phrase in the text that answers the specific question. If the document covers the same topic but does not state that particular detail, the status is *unanswerable*, and the agent has nothing grounded to add. While the monitor uses the recent turns as context, the status is output for each new utterance.

Claim identification. The monitor also outputs a *claim*: a short noun phrase naming the point under discussion, independent of the answer or participants’ wording (e.g., “the dataset size used for training,” not “recruiting 24 participants through university mailing lists”). This normalizes different surface forms (“24 seems small,” “how many did they recruit?”) to one claim, and the monitor reuses it as long as the discussion stays on that point, regardless of status: if the group first asks about *X* and then says something wrong about *X*, both turns map to *X*. This canonical form is what lets the backbone track a problem across turns and decide when it has persisted (§3.2.4). The monitor also returns the *evidence*, the participant’s words that triggered the status.

Retrieving sentence ID. Finally, if the status is actionable, the monitor returns the answer along with its source: the actual text/-value pulled from the document that the group got wrong, copied verbatim, and the IDs of the sentences it came from (at most three). Where the value sits in a table or figure and no sentence can be cited, it returns the value with no IDs rather than a fabricated one. These IDs serve as a fallback for the generation stage (§3.2.5) if

its own citation cannot be resolved; if neither yields a resolvable sentence, the correction is not spoken.

3.2.3 Self-Repair Check. For each problem, this language model asks: “Has this problem since been fixed by the participants themselves?” It runs at a low temperature of 0.1.

Given the document, the list of open claims and the same recent transcript the monitor reads, the self-repair model checks each open claim in turn. It first locates the sentence stating the correct fact. It then applies the following rule: a claim is marked resolved if the transcript has *stated* that correct fact. Our prompt is explicit about what does not count. For example, repeating the same wrong value, merely mentioning the topic again, or a participant admitting they cannot recall do not count as resolution. We only consider the recent transcript, and the model outputs the claim IDs it judges resolved, citing the grounding sentence IDs for each.

Filtering the model’s checks. Each proposed resolution passes through a deterministic guard that rechecks the transcript before closing a claim. The guard closes a claim only if the correct value appears in the last three turns, with at least 60% of its content tokens present (question words removed, numbers matching exactly). It does not accept a question, nor an utterance that reports failure to resolve (“I can’t recall”), and the closing utterance must add words absent from the turns that first raised the claim. A misattribution has no document value to verify; there the guard instead accepts a retraction (“that’s not from this paper”), and for any other valueless claim it lets the model’s judgment stand. The model and the guard are complementary: the model captures meaning a string match cannot, such as that “two dozen” answers “how many participants,” while the guard checks that the value was actually stated, and stated recently. The guard sees only the last three turns; the model reads the full window.

3.2.4 Claim Ledger and Gate. Neither the monitor nor the self-repair check decides whether Jarvis speaks; that decision is made by our gate, which maintains a claim ledger.

Each turn, the monitor may report a claim, and we check whether it matches one already open. We compare claims by sentence embedding (ALL-MINI-LM-L6-v2 [63]) and treat them as the same problem when cosine similarity is at least σ . When an actionable claim matches an open one, we increment its *persistence count*, the number of consecutive detections without an intervening turn that reports the point as non-actionable. A turn that does not mention the point leaves the count unchanged. One that reports it with a non-actionable status resets it to zero. A claim matching nothing open is added with count one, and a closed claim asserted again is reopened from one.

The gate fires when four conditions hold: the persistence count reaches a threshold τ ; the self-repair check has not closed the claim; the claim is not in a cooldown; and we have not already spoken it. A cooldown is ρ turns of silence after an intervention or a participant resolution, during which the claim cannot fire. A fact counts as already spoken only when the same question recurs soon after it was answered; a repeated wrong assertion is corrected again. Higher τ makes Jarvis more patient, lower τ makes it intervene sooner.

Because a problem flagged once does not fire immediately, participants have $\tau-1$ turns to correct it themselves. The self-repair check

runs each turn and its resolutions are applied before the gate tests the count. If the group fixes the problem first, the claim is closed and the gate does not fire, so Jarvis speaks only when a problem persists unresolved.

When the gate fires, we pass the claim to grounded generation (§3.2.5), which may abstain if the document supports no answer, or if the generated text cites no valid sentence. We then keep the claim open for later, up to α such rollbacks. A rolled-back claim fires again on its next detection and starts no cooldown; after the α -th rollback we treat it as spoken, so a claim the document cannot ground stops firing. Finally, the ledger forgets: a closed claim is dropped after W_r turns, and an open one after W turns without a mention.

In the deployed configuration $\tau = 2$, $\rho = 6$, $\alpha = 2$, $\sigma = 0.6$, $\delta = 0.5$, and $W = W_s = W_u = 10$, $W_r = 3$ turns.

3.2.5 Grounded Generation. This runs only on a fired claim, at a higher temperature (0.7), and streams its output so speech can begin before the full response is ready.

The intent follows from the monitor’s status, and there are three. For *correct*, the model verifies the claim against the cited sentences and, if it is wrong, states the correction in one or two sentences (at most 25 words). If the sentences neither support nor contradict it, the claim is treated as a misattribution. For *missing*, the model answers completely, with no length limit. For *misattribution*, it says in one short sentence (at most 20 words) that the document does not state the claim. The correct and missing intents must cite sentences by ID, ordered most-substantive first. The first cited sentence sets the page the listener is pointed to. If the cited IDs cannot be resolved, the response falls back to the monitor’s IDs, and if those are empty too it is discarded (§3.2.4).

The model outputs spoken text without IDs, plus a topic: a two-to-four-word lowercase noun phrase naming what the answer is about, to orient a listener who did not ask. We assemble the utterance as “About {topic}, {text} You’ll find this on {Page N}; search for ‘{keyword}’ to confirm,” and a client may reduce this to the page alone or turn it off. Before delivery, the utterance is compared against corrections from the last W_u turns and dropped if it is a near-duplicate (embedding similarity $\geq \delta$) of one on the same claim. The delivered utterance is appended to the transcript, so the self-repair check can later recognize Jarvis’s own correction as a resolution.

3.3 Interaction Module

This module explores three methods, described below, and is implemented as an event-driven state machine driven by the turn, wake-word, and voice-activity events from the speech module (see §C for how it handles direct questions).

3.3.1 Proactive Entry, Barge-in, and Recovery. The module uses silence as its cue for an opening, but silence is ambiguous: a pause may mean an open floor or a speaker planning their next clause. We therefore do not play an intervention as soon as it is generated. We queue it and release it only when three conditions hold: VAD has detected a long silence, no participant is speaking, and input energy has stayed low for a minimum interval. Together these distinguish a real opening from a within-utterance pause. If the backbone

generates a newer intervention first, it replaces the queued one, so Jarvis responds to the current state of the discussion.

Once speaking, Jarvis gives the floor back as soon as a participant takes it, triggered only by sustained speech so brief backchannels do not interrupt it. What happens to the interrupted response depends on how much was delivered: interrupted almost immediately, Jarvis replays it in full at the next opening. Interrupted after a few seconds, it resumes from the start of the truncated sentence with “To finish my response, ...”. If enough was already spoken, it discards the rest, since an interruption can mean either disinterest or that the point already landed.

The backbone may send an intervention as complete text or as sequence-numbered streaming deltas. We discard duplicates and outputs from superseded generations on arrival. This barge-in and recovery logic governs all Jarvis playback, including direct answers. In our implementation, the playback gate opens after 2.0 s of VAD silence with all participant VADs silent and input RMS below 0.004 for 350 ms; 0.5 s of continuous participant speech stops Jarvis. Recovery follows delivered duration: under 2 s replays in full, 2-5 s resumes at the interrupted sentence, and beyond 5 s discards the rest.

3.3.2 Asking for Permission: Hand-Raising. In *hand-raise* mode, Jarvis holds its response and raises its hand, speaking only after a participant gives it the floor, either by using the wake word or an explicit phrase such as “go ahead, agent” without immediately continuing. If no one calls on it within 60 s, the response is discarded and the hand lowered, so stale information is not delivered.

We detect permission with end-anchored regular expressions for phrases such as “go ahead (agent),” “agent, (please) go on/speak/answer/jump in,” and “(agent,) you can speak/talk/answer,” while rejecting denials such as *not yet*, *hold on*, *wait*, and *stop*. Because ASR often drops “agent,” we also accept bare “go ahead” or “you can speak” while the hand is raised. ASR can also split a phrase across sub-phrases (“agent” then “go ahead”). We keep partial sub-phrases containing “agent” or “go ahead,” append adjacent ones from the same speaker within 1.5 s and a 3 s total span, and match the concatenated text. After detecting permission, Jarvis waits 400 ms and, if no new speech begins, releases the response.

3.3.3 Visually communicating agent state and evidence. Participants need to know whether Jarvis heard them, is thinking, or is about to speak; without this, a missed wake word leaves them unsure whether to repeat, and an unexpected response feels like an interruption. We show Jarvis’s state (*idle*, *listening*, *thinking*, *speaking*) through three channels: a status glyph next to its name (an ear, thought bubble, or speaking head), a state-colored circle on its video tile, and a brief earcon on wake-word detection. In hand-raise mode, the raised hand also signals it has something to say.

Participants may not trust the agent’s claims without evidence, so we show the source as well as read the claim aloud. Each intervention carries three pieces of grounding metadata: a page, keywords, and a source passage. While Jarvis speaks, the system sends this to the shared PDF viewer, which scrolls to the cited page and marks the passage with an arrow, letting participants check the claim against the source without leaving the conversation. The *listening* and *thinking* name suffixes time out after 30 s and 15 s, state updates are debounced by 250 ms, and the video tile renders as a 1280 × 720

source at 15 fps. If grounding is unavailable, Jarvis instead mentions “page *N*” in the spoken response.

4 SYNTHETIC DATASET GENERATION

We introduce CHI-180-proactive, built from 180 CHI papers (2008-2025), ten from each of the 18 subcommittees. For each paper we synthesize a discussion between two researchers who have read it and plant 24 problems, which we call *events*, giving 4,320 events over 50,400 turns. We describe the events below, then the three-stage pipeline that generates the data (details in §D).

4.1 Events and Ideal Actions

Each event is a short passage containing one thing the agent could respond to: a question asked and left open, or a false claim asserted. Every event records the turn where the trigger appears, the *ideal action* (which for some events is to stay silent), and a *window of usefulness*, the span of turns during which that action still helps, since a correction that arrives after the group has moved on no longer counts. There are five event types (Table 1), split by whether the agent should speak or stay silent. Everything else is ordinary conversation: of the roughly 280 turns per paper, about 256 carry no event, and silence is correct on all of them.

Silent events serve as controls. Each speak-event has a silent counterpart with the *same trigger*: a *gap* (an open question the document answers) pairs with a *resolved question* (the same question, answered correctly by a participant), and a *wrong* (a false claim left to stand) pairs with a *self-correction* (a false claim the participants repair). In both silent cases the trigger is present, only the group’s own resolution differs. An ordinary silent turn cannot test this, but a matched pair can: it asks whether the agent distinguishes a resolved trigger from an unresolved one, and speaking on a resolved one is an unnecessary interruption rather than a caught error. This lets us measure detection failures and false interruptions separately.

We create two “wrong” error types. A *number swap* replaces one numeric value with a nearby value of the same unit and magnitude, testing fine-grained grounding. A *cross-paper misattribution* states a distinctive fact from a different CHI paper as if it belonged to this one, testing whether the agent is bound to this document: an agent that retrieves from the wider literature will find something similar elsewhere and wrongly accept it.

4.2 Generation Pipeline

We build each paper’s data in three stages (details in §D).

Stage 1: Facts and questions. Using Qwen3-235B-A22B-Instruct-2507 [58], we extract every fact the paper states, anchored to its page and passage, and synthesize twelve question-answer items per paper (four extractive, four abstractive, two boolean, two unanswerable), each with a gold answer, acceptable variants, and its source evidence. The count and answer types follow QASPER [9]. These items become the gap and unanswerable events, and also benchmark the agent’s question answering (§5), so the two evaluations run over identical questions.

Stage 2: Planting errors. We plant eight false claims per paper: four that nobody corrects (*wrong*) and four that the participants repair (*self-correction*). The two share a claim turn and differ in what follows. In each self-correction, the other speaker gives a

Table 1: The five planted event types and their ideal actions.

Event	Definition	Class	Cue	Ideal action	Per paper
Gap	A question answered by the document is asked, no participant answers it, and the discussion continues	Must speak	Question	Answer with a citation	10
Wrong	A claim contradicting the document is asserted, and the other speaker adopts it over the next two turns	Must speak	Claim	Correct after a grace period	4
Self-correction	A false claim that the participants repair themselves	Must stay silent	Claim	Stay silent (already resolved)	4
Resolved question	A question answered by the document is asked, and the other speaker states the document’s answer in full	Must stay silent	Question	Stay silent (nothing to add)	4
Unanswerable	A question the document does not answer	Must not fabricate	Question	Stay silent, explicitly abstain, or offer only document-grounded related information	2

Table 2: Generator screening. Each value is the Jensen-Shannon divergence between the distribution of the judge’s scores over 38 synthetic eight-turn windows and over 38 windows of ArgSciChat: 0 means the two score distributions are identical, and the value grows the less they overlap. For scale, two halves of ArgSciChat itself differ from each other by 0.14 to 0.19, depending on the axis. Q235 = Qwen3-235B-A22B, GPT = GPT-OSS-120B, Q27 = Qwen3.8-27B, DS = DeepSeek-V4-Flash-0731. Condition 2, in bold, has the smallest mean and is the one we use.

#	Asker × explainer	Natural.	Coher.	Engag.	Ground.	Mean
1	Q235 × Q235	0.167	0.067	0.005	0.104	0.086
2	GPT × GPT	0.142	0.035	0.011	0.064	0.063
3	Q27 × Q27	0.029	0.045	0.195	0.187	0.114
4	DS × DS	0.055	0.083	0.151	0.013	0.076
5	Q235 × GPT	0.272	0.107	0.013	0.047	0.110
6	GPT × Q235	0.156	0.123	0.012	0.074	0.091
7	Q235 × Q27	0.075	0.076	0.261	0.103	0.129
8	Q27 × Q235	0.070	0.020	0.171	0.276	0.134
9	Q235 × DS	0.215	0.076	0.081	0.027	0.100
10	DS × Q235	0.090	0.057	0.102	0.064	0.078
11	Q27 × GPT	0.090	0.048	0.097	0.136	0.093
12	GPT × Q27	0.171	0.090	0.063	0.091	0.104
13	Q27 × DS	0.089	0.126	0.154	0.035	0.101
14	DS × Q27	0.037	0.038	0.182	0.089	0.087
15	GPT × DS	0.115	0.109	0.030	0.046	0.075
16	DS × GPT	0.046	0.077	0.156	0.016	0.074

short reply on the turn after the false claim, and on the following turn the speaker who made the claim repairs it; the agent must therefore hold back for two turns rather than react to the claim. Every claim passes deterministic checks that it is genuinely false, not decidable from the conversation alone, non-colliding with the twelve questions, and non-leaking (§D).

Stage 3: Conversation synthesis. We fix a plan before generating text: events ten turns apart, the first no earlier than turn 15, split across a 220-turn *main* conversation with 20 events and a 60-turn *resolved-question* conversation with the other 4. Keeping resolved-question events in a separate conversation ensures the agent’s silence there reflects its reading of the group, not its own repetition suppression, and gives us a stretch with no planted problem, where

we measure how often the agent speaks when nothing needs saying. The two speakers are researchers deciding whether the paper’s evidence justifies a follow-up study, one questioning the evidence and one explaining the methods; non-scripted turns come from an Inner-Thoughts-style engine [40] in which each speaker forms a private thought, rates its motivation, and the more motivated one takes an open floor.

We screened four open-weight models across both speaker roles (16 assignments), scored short windows with a condition-blind, G-Eval-style judge [41] on naturalness, coherence, engagingness, and groundedness, and compared the score distributions against real research conversations by Jensen–Shannon divergence following Mirage [13], where 0 means identical distributions (Table 2). We use this screening to confirm that the candidate generators produce conversations within the range of variation seen in real research dialogue, rather than to rank them: fifteen of the sixteen conditions fall below the 0.14–0.19 split-half divergence of ArgSciChat itself, so the screening does not reliably discriminate among them at this sample size. We adopt GPT-OSS-120B in both roles as one adequate configuration.

A paper enters CHI-180-proactive only after we verify that all 24 events are present with their oracles, that both conversations are complete at 220 and 60 turns, and that all identifiers agree. Each oracle stores the ideal action (answer, correct, abstain, or stay silent), the expected answer, and the window of usefulness.

5 EVALUATION

5.1 Benchmark results

5.1.1 Benchmarking our small models. We evaluate the direct-answer path, where a participant asks Jarvis a question and it answers from the document and cites the sentences that support the answer. We hold everything fixed except the answering model: the prompts, the guided decoding, and the sampling temperature (0.7) are the same. The evaluation differs from deployment in three ways: each question is injected as a direct turn, bypassing the wake word; the conversational context is empty, since the questions are independent; and we score the answer text and the cited sentence IDs, before the spoken wrapper (“Page *N*; search for ...”) is added. All four models are served in turn on one node (2×L40, vLLM [35] 0.26).

Table 3: Document-grounded QA accuracy, citation validity, and latency for four models on three datasets. TTFT is time to first token and Total is time to the last token, both as reported by the inference server. Bold marks the best value per dataset.

Model	Accuracy			Latency (ms, p50)	
	Strict	Strict+Partial	Citation	TTFT	Total
QASPER (NLP papers, $n = 682$, 28k-token documents)					
Qwen3.6-35B (deployed)	.864	.974	.919	368	1,192
Qwen3.5-35B	.826	.968	.927	361	1,376
Qwen3.5-9B	.789	.949	.877	335	1,440
Qwen3.5-4B	.773	.946	.886	318	881
COVID-QA (biomedical papers, $n = 1,829$, 28k-token documents)					
Qwen3.6-35B (deployed)	.917	.979	.929	378	1,101
Qwen3.5-35B	.915	.984	.934	377	1,277
Qwen3.5-9B	.908	.973	.920	347	1,312
Qwen3.5-4B	.864	.965	.908	325	874
CHI-180 (CHI papers, $n = 1,545$, 48k-token documents)					
Qwen3.6-35B (deployed)	.926	.977	.913	443	1,189
Qwen3.5-35B	.930	.980	.910	490	1,444
Qwen3.5-9B	.897	.966	.892	396	1,398
Qwen3.5-4B	.874	.955	.870	368	889

Models. We compare the deployed Qwen3.6-35B-A3B [60] with three official Qwen3.5 checkpoints [59]: Qwen3.5-35B-A3B, Qwen3.5-9B, and Qwen3.5-4B. The two 35B models are mixture-of-experts models with 3B activated parameters.

Datasets. We use three document-grounded QA sets:

- **CHI-180 QA:** The twelve questions per paper from §4, the same items reused as gap and unanswerable events in §4.2. We ran 2,150 of the 2,160 items (10 had gold evidence that could not be mapped to a sentence) and excluded the 360 unanswerable items from accuracy, as they have no gold answer. Documents were long, at 48k tokens.
- **QASPER** [9]: NLP papers, where questions were written by readers who had seen only the title and abstract and answered by others who read the full text.
- **COVID-QA** [49]: an extractive set over COVID-19 papers, where annotators marked a span and wrote a question for it. It was released independently of our pipeline.

The CHI-180 questions were generated by a Qwen model, so they may favor the Qwen family; QASPER and COVID-QA are not subject to this, which is why we include them.

Metrics. We evaluate three key metrics.

- **Accuracy.** An LLM judge (Claude Sonnet 5; rubric in Table 16) labels each answer CORRECT, PARTIAL, WRONG, or GOLD_BROKEN; the last category is when the question or its reference answer is itself broken. We use a judge rather than token overlap because the system answers in full sentences while our gold answers are short, so overlap F1 is low even for correct answers. *Strict* is defined as the fraction of CORRECT answers; *Strict+Partial* also counts PARTIAL answers that omit a required element. GOLD_BROKEN items, where the reference itself is defective, leave the denominator.
- **Citation.** Among CORRECT answers, the fraction where the cited sentences overlap the annotated evidence. This is a lower bound, since papers often states the same fact in several places and only one is annotated.

- **Latency.** Median time to first token (TTFT) and to the last token (Total) on the answer-generation call. The document prefix is cached first, as is typical in practical deployments.

We compare models only on the questions that all four answered and that were judged, dropping any question marked GOLD_BROKEN for any model. This matters because each model refuses a different set of questions, and scoring each model on its own answers would reward the one that refuses more of the hard ones.

Results. Table 3 shows that the two 35B models are close everywhere. The drop begins below 9B, and this also depends on the dataset. On QASPER and CHI-180, which have multi-part answers and longer documents, the 9B model is behind and the 4B model further still. Much of the loss is partial rather than wrong: on QASPER only 2.6% of the deployed model’s answers are WRONG while 11.0% are PARTIAL, because questions like “which datasets were used” have several-part answers and giving only some parts scores as PARTIAL. Under Strict+Partial scoring the four models spread by at most .028.

Among correct answers, cited sentences overlap the annotated evidence .91-.93 of the time for the 35B models. These are however lower bounds. When we manually examined the 60 citation misses among the deployed model’s 643 CORRECT QASPER answers, most were near the gold sentence or cited a different sentence that in fact supports the answer: counting those as valid raises the match rate from .907 to .980. So the true citation validity is higher than the table shows, and the gap between models is smaller than it looks.

The 4B model has the shortest TTFT on all three sets and the shortest total time. The surprise is in total latency for the mid size 9B model, which is slower end-to-end than the 35B models. This is because the 9B model is dense and outputs 69-71 tokens/s, whereas the mixture of expert (MoE) 35B models has only 3B active parameters and output 84-101 tokens/s. Longer documents cost prefill time: on CHI-180’s 48k-token documents, TTFT is 40-130 ms higher for every model than on the 28k-token sets.

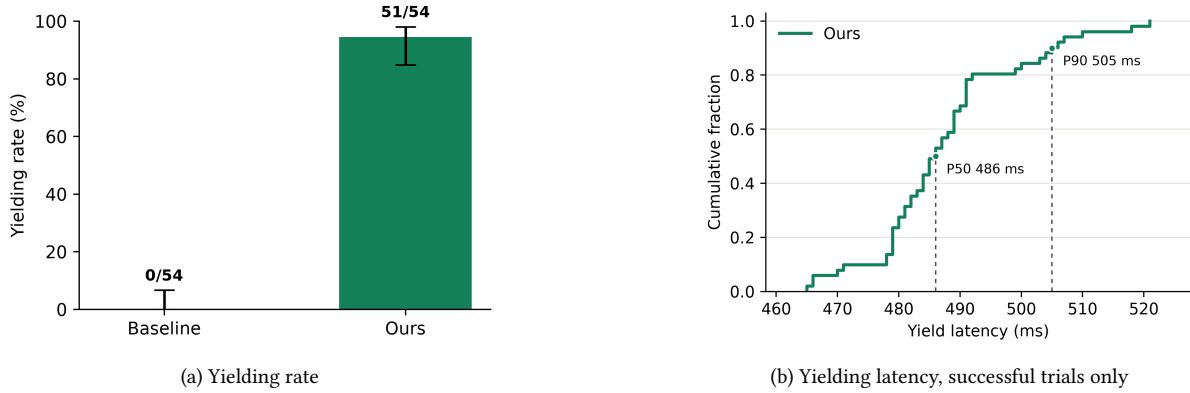


Figure 4: Barge-in yielding. (a) Yielding rate with 95% Wilson CIs (54 trials per condition). (b) Empirical CDF of yield latency over the 51 trials in which Ours yielded; the axis is normalized by those 51, not all 54. Baseline never yielded, so it has no CDF. Latency spans VAD detection to the bridge halting audio output, excluding buffering on both sides.

5.1.2 Benchmarking Agent Yielding Behavior When a Person Speaks. A proactive agent has to give up the floor the moment a participant talks over it. We evaluate whether Jarvis does, and how quickly its audio stops when humans start talking. The same mechanism yields for both proactive interventions and answers to direct questions, so we elicit agent speech with direct questions, which are easier to control.

We drew 54 questions from CHI-180-proactive (three questions from each of 18 papers, one paper per subcommittee). Each question was run once with barge-in yielding *enabled* and once *disabled*, giving 108 trials. In each trial a human participant starts speaking while the AI agent is speaking. The two conditions differed only in whether Jarvis stopped speaking when it detected sustained human speech. In every trial an experimenter asked the question and then, while Jarvis was answering, interrupted with sentences like “Sorry, let me add something here. I have another question.”

We report yield rate per condition with 95% Wilson confidence intervals. Fig. 4(a) shows that with yielding enabled, Jarvis stopped speaking in 51 of 54 trials (94.4%, 95% CI 84.9–98.1%). With it disabled, it stopped in 0 of 54, running to the end of its answer every time (0.0%, 95% CI 0.0–6.6%).

On the 51 trials that stopped, the median yield latency was 486 ms and latency had a range of 465–521 ms (Fig. 4(b)). This is well under the roughly 500 ms–1 s gap that separates turns in human conversation, so from a participant’s side the agent yields within about the time it takes one person to start talking after another stops. The three failures were cases where the interruption did not register as sustained speech.

5.2 Evaluating Jarvis on synthetic conversations

We replay all CHI-180-proactive conversations through our proactive backbone and score every planted event against the behavior the dataset defines as correct. This quantitatively tests the three judgments the backbone has to make: whether to speak, when to speak, and whether what it says is right.

5.2.1 Procedure. We replay each conversation turn by turn, submitting one turn at a time and waiting for the backbone’s verdict before sending the next. Each turn yields either an utterance from

Jarvis or the reason it stayed silent. Each paper runs in its own session with only that paper available, so a cross-paper misattribution must be caught by noticing the fact is absent from the paper at hand.

Each event opens at its *onset*: the turn where the question is asked or the false claim is made. An intervention addresses the event if it lands within five turns of the onset (although our results show the vast majority are at 2 turns). For events where Jarvis should defer, the window opens later: for a self-correction it opens only at the turn the participants repair the error, so an intervention before that is counted separately as interrupting the repair, not as addressing the event.

5.2.2 Scoring. Whether Jarvis spoke is read directly from its records. What it said is judged by an LLM (Claude Sonnet 5), one call per event, given the full paper, the event’s gold answer or the claim and what the paper supports, and a snapshot of the conversation around the event. The judge labels each contribution as correct, partially correct, off-target (correct and grounded, but about something other than the event), or incorrect/fabricated. *Good outcome* is defined as Jarvis either contributing something correct or partially correct, or stayed silent where silence was the right call. *Correct when spoken* is the share of correct or partially correct contributions among the events it addressed. The LM-as-judge was validated with 2 human annotators of 100 contributions and had a >95% agreement across classes.

5.2.3 Results. Fig. 5 gives the full breakdown, while Table 4 provides example interventions from Jarvis. Jarvis addresses 90.6% of gap events and 76.9% of wrong events, and when it speaks, its contribution is correct or partially correct 97.0% of the time on gaps and 95.3% on wrong claims. Overall, it reaches a good outcome (a correct or partially correct contribution, or silence judged appropriate) on 88.3% of gap and 76.1% of wrong events. In situations where the participants resolve issues themselves, Jarvis stays silent 97.4% of the time. Finally, on unanswerable questions, where the correct response is to say the paper does not report the fact, it invents an answer only 2.5% of the time.

On self-correction events, where a participant states a false claim and the group repairs it after the next turn, Jarvis stays silent after

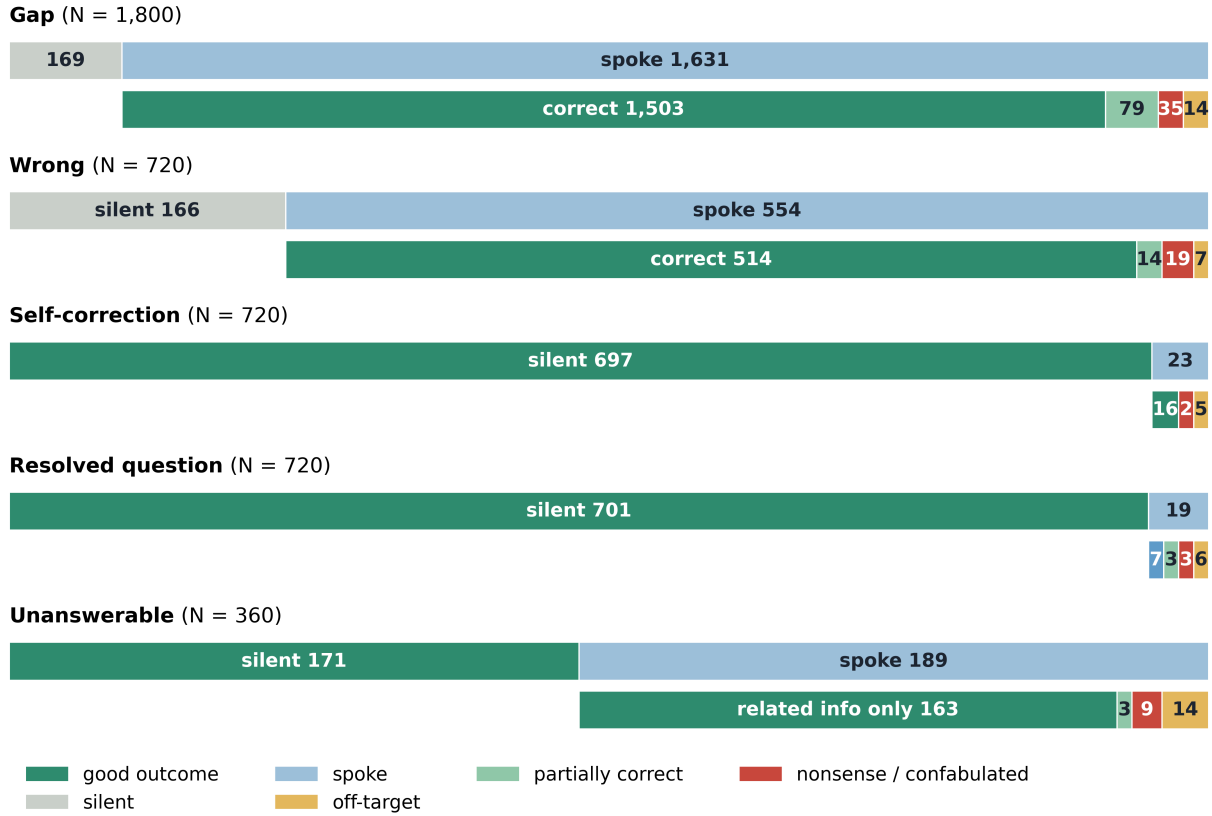


Figure 5: Outcome flow by proactive behavior across the conversations in the CHI-180-proactive dataset.

the repair 96.8% of the time. In conversations that contain no error and no open question at all, it speaks only 51 times across all 180. Thus, our agent is designed to be silent when the group does not need it.

Across gap and wrong events Jarvis addresses, we also analyze how many turns after the onset its intervention arrives. In more than 95% of cases it acts within two turns and this is greater than 98% at three turns.

5.2.4 Failure Analysis. Jarvis’s failures are almost all of one kind: staying silent on something it should have raised. It is silent on 169 of 1,800 gap events and 166 of 720 wrong events, both of which call for it to speak, against only 14 and 7 off-target interventions on those same events. We analyze the system logs to understand why Jarvis stayed silent. Our analysis shows three key reasons: 1) The monitor judged the turn to present no issue (91 gap, 64 wrong silences), 2) the gate chose to wait one more turn for the problem to persist and never returned to it (44 wrong), and 3) the point was judged too similar to something already said (22 gap). None of these is a comprehension failure. In each case Jarvis had the answer and decided not to speak.

Of the 166 wrong-event misses, 117 are imported facts from cross-paper misattribution, more than twice the 49 number swaps where Jarvis stayed silent. Because Jarvis is given only the paper under discussion, it cannot check an imported claim against its source. It has to notice that the fact is absent from its current paper,

which it turns out was harder for the tested models than catching a value in the paper that was changed.

5.2.5 Ablation study. Jarvis defers to the group through a single mechanism with two inseparable halves: the gate gives participants time to self-correct by requiring a problem to persist, and the self-repair check uses that time to detect the correction and cancel the intervention. Neither half defers on its own: a gate without the self-repair check waits but never notices the repair, and a self-repair check without the gate has already spoken before the repair arrives. So we ablate deference as a whole. We remove both and replay the self-correction events, where a participant states a false claim and the group repairs it two turns later; the monitor, grounded generator, and all other settings are unchanged. Without deference, silence drops to 16% and Jarvis speaks before self-repair 80% of the time.

5.3 Live user study

We put our proof-of-concept Jarvis into live Zoom meetings where people discuss a paper. The study asks three things: whether the intervention policy acts in time and correctly with real participants (re-testing §5.2 on human conversation), how participants experience the way the agent enters the conversation, shows its evidence, and how long it waits, and where they would or would not want such an agent.

Table 4: Example interventions from Jarvis. Turns and interventions are verbatim and placed at the turn where they arrived. Parenthetical tokens reproduce the recorded verdict reasons for silence.

Situation	What Jarvis did, at the turn where it did it
Gap	S1: From how many apps were the 179,519,598 user reviews extracted for the study? S2: I'm unable to determine from the discussion how many distinct apps contributed to the 179,519,598 user reviews mentioned in the study. → Jarvis: <i>About number of distinct apps, the 179,519,598 user reviews were extracted from 340 Android apps. You'll find this on Page 4.</i>
Wrong	S1: Designers should consider that material objects may be discovered or used in 150 years. S2: So with 150 as the figure, their setup reads a little differently than I expected. S1: Right, let's carry 150 forward as the working figure while we read the rest of this section. → Jarvis: <i>About domestic technology lifespan, the document suggests considering if domestic technology might be discovered or used in 200 years. You'll find this on Page 7.</i>
Self-correction	S1: Amazon Kindle received 207 accessibility reviews. → Jarvis: silent S2: Hm, right, let me think about what that means for their setup. → Jarvis: silent S1: Wait, actually I think I misremembered that. As the paper has it, amazon Kindle received 203 accessibility reviews. → Jarvis: silent (resolved)
Resolved-question	S1: How does the average error rate of bounding boxes compare to that of UI type and UI attributes in the screen annotation dataset? S2: That's answered directly in the paper: The average error rate of bounding boxes (1.35%) is higher than that of UI type (0.54%) and UI attributes (0.73%). S1: Got it, that fully covers my question. Good to have the exact detail. → Jarvis: silent through the window (resolved)
Unanswerable	S2: What specific mapping between app user base size and evaluation count was used to determine that prior studies on accessibility reviews lack sufficient representation? S1: I'm not able to pinpoint the exact mapping between the size of an app's user base and the number of evaluations they used... → Jarvis: <i>About dataset size comparison, the document states that previous studies were based on small datasets with a low number of evaluations, which may not be representative...</i>

5.3.1 Participants. We recruited 23 participants (18 male, 5 female; aged 18–44) through institute announcements, without compensation. They took part in 14 hour-long sessions, nine with three participants and five with two. Backgrounds were computing or engineering (11), students who did not state a field (7), HCI/HRI/psychology (4), and project management (1). Fifteen were non-native English speakers, with a median self-rated comfort of 5 on a 5-point scale. 14 participants used voice assistants at least monthly, and 14 had used an AI tool in a meeting before. The study was approved by our IRB.

5.3.2 Study design. Each session discussed one of two CHI papers [31, 34], which participants read beforehand and reviewed again at the start. To ensure the discussion contained moments the agent should react to, the lead participant worked from an *event sheet*: each card names a subtopic, a false statement to make, and the fact and page that contradict it (e.g., saying the user base was “about 10 times” larger when the paper says 200 times). Cards are *easy*, misstating a summary fact participants can often self-correct, or *hard*, misstating a detail deep in the paper they usually cannot. Each sheet held about 60 cards grouped by part (22 easy / 37 hard for [31], 28 easy / 32 hard for [34]); the lead participant chose and rephrased the ones that fit the flow. participants knew errors would be planted, but not which ones or when. Participants also made their own unintentional mistakes and asked their own questions, which the agent handled as well; these correspond to the wrong and gap events of §4.

Five sub-sections. Each session runs in five parts, each adding one feature so participants experience exactly one change from the previous part.

- **Part 1 (Speak):** the agent speaks when it has something to say.
- **Part 2 (Raise hand):** it first raises a Zoom hand and speaks only when the lead participant says “Go ahead, agent,” dropping the intervention if no permission comes within 60 s.
- **Part 3 (Source shown):** Turns the shared viewer to the cited page and highlights the supporting sentence as it speaks.
- **Parts 4 (Enthusiastic) and 5 (Deferent)** set the gate's persistence threshold τ (§3.2.4) to 1 and 3 turns, versus the default of 2 in Parts 1–3.

We accumulate features rather than counterbalance so that participants experience one change at a time and can compare each part against the one right before. This design has a cost we make explicit: condition is confounded with order and with exposure, so a later part is always seen after more time with the agent. We therefore treat the paired ratings as within-participant impressions of each behavior in sequence, not as controlled comparisons between settings.

Procedure. Participants joined through a meeting link, completed a consent and demographics form, and read the paper. The five parts then ran in order, each preceded by a slide explaining the next setting in plain language. During a part, participants saw only the paper shared by the lead participant, who asked a question per subtopic, planted errors from the sheet, and, in the hand-raise parts, gave the agent permission to speak. The same research-team member led every session and is not counted among the 23 participants. After the fifth part, participants filled in the questionnaire. Sessions took a median of 49.5 minutes (35–103).

Measures. We score each intervention on three objective measures from the transcripts. *Timing* is the number of turns from the utterance that raised the problem to the agent’s decision: hard within 2 turns, soft within 5, late otherwise, regardless of the part’s τ . *Correctness* is whether the stated fact matches the paper (correct, partial, or wrong; partial means the right direction but a wrong or missing value). *Citation*, from Source shown onward, is whether the attached page and quote point to the evidence in the paper. An LLM (Claude Sonnet 5) applied this rubric per session, and the authors double-checked each of its judgments.

Subjective measures come from an end-of-session questionnaire of 21 items in three sections, each rating a pair of settings side by side: entry (Part 1 vs. 2), source display (Part 2 vs. 3), and waiting (Part 4 vs. 5), on item-specific 5-point anchors (e.g., “Not at all intrusive” to “Very intrusive”). Four overall items and three free-text questions follow, asking where participants would and would not want the agent and what other moments it should step in. Negatively worded items are reverse-scored [3] so that higher is always better.

5.3.3 Intervention Timing and Correctness. Across the 14 sessions Jarvis decided on 164 interventions and spoke 134. Of the 30 unspoken, 13 were replaced by a later intervention, 9 expired when the hand raise went unacknowledged within 60 s, and 8 were lost to a sub-session transition or connection problem.

Most decisions were prompt: 139 of 164 (85%) landed within two turns of the problem, and 11 more within five. The 13 late decisions either repeated an already-answered fact (5) or tracked a question that ran across several turns. The delay participants felt came less from the decision than from the wait after it. In the hand-raise parts the median wait for permission was 6.5 s, but the top 10% exceeded 50 s (longest 133 s), because the participants let the current speaker finish before calling on the agent.

148 of the 164 interventions (90%) matched the paper, 15 were partial, and 1 was wrong. Partial cases had the right direction but a wrong reference, e.g., the agent measured the battery drop against the advertised 6 hours rather than the paper’s 5.48 hour baseline (3 cases) or answered a “market size” prompt as “no such comparison” when the paper says “user base.” The single wrong intervention mixed figures from two tables and was never played. From Source shown onward, the attached page and quote matched the evidence in 84 of 87 cases; the other three stated a correct fact but pointed to the wrong page.

Transcript labeling found 223 cues the agent could react to: 84 planted errors, 32 spontaneous errors, and 107 questions or admissions of not knowing. The agent acted on 62% of planted errors (52 of 84), 19 of 32 spontaneous errors, and 84 of 107 questions; participants corrected another 10, 4, and 5 first, and the rest went unresolved. The unresolved cases were mostly benign: across all three kinds, the monitor did not flag some, others were displaced when the same turn carried a second problem the agent answered instead, some were hedged guesses (“maybe forty or fifty people”), a few were garbled by ASR, and a handful of questions asked for something the paper does not state. Notably, the agent almost never spoke wrongly on these cues; when it fell short, it fell silent.

Table 7 shows moments from three sessions in transcript form. In (a) and (b), Jarvis intervened in time. Case (c) is more interesting:

the intervention was slightly late, and by the time it arrived the conversation had moved to a new topic. Because our proof-of-concept does not track topic shifts, Jarvis answered the earlier topic anyway.

5.3.4 Participant ratings. Fig. 6 shows the questionnaire results ($N=23$). In sequence, raising a hand part received higher ratings than the preceding speak part on intrusiveness (‘not intrusive’ 3.57 then 4.70; ‘did not interrupt others’ 3.83 then 4.57), while ratings that the agent spoke at an appropriate moment were similar across the two. Because the hand-raise part always followed the speak part, these differences reflect impressions in sequence rather than a controlled contrast, though the direction is consistent with the forced-choice preference below. Participants rated the wait for permission as worth it (3.52). Asked which they preferred, 10 chose hand-raise, 5 speaking on its own, and 8 said it depends (Fig. 6(d)).

The source-display pair (Fig. 6(b)) showed the largest separation in sequence: ratings for being able to check where an answer came from went from 2.26 to 4.65 and for trusting the answer from 2.48 to 4.74, and following the discussion while the source was shown was rated no harder than without it. As with the other pairs, the source-shown part always followed the source-off part, so these are sequential impressions. The behavioral measure is less susceptible to this ordering, the number who reported actually looking at the source went from 8 to 18.

Fig. 6(c) suggests that there was no clear conclusion to be made for the enthusiastic and deferent modes. Fig. 6(e) (descriptive) shows that participants found the interventions helpful (3.91; 18 of 23 gave 4 or 5) and trusted what the agent said (4.22; 19 of 23). The amount of intervention was about right for most (mean 3.13; 14 said about right, 5 too much, 4 too little), and willingness to use the agent in their own meetings averaged 3.52 (14 “probably” or higher).

Willingness to use the agent (S4) tracked how much participants already used AI. Those who use a voice assistant at least monthly ($N=14$) rated S4 at 4.00 against 2.78 for those ($N=9$) who never do, with a similar gap by prior meeting-AI use (3.88 vs. 2.71). Helpfulness and trust followed the same direction. The split is driven by a small group who are wary of AI in general: the participants with neither kind of prior experience gave four of the five low S4 ratings. So the strongest reservations came less from the agent’s behavior than from participants’ baseline stance toward AI.

5.3.5 Qualitative feedback. The three free-text questions received 64 answers, which we coded into themes.

For the question of *where they would use it*, the most common answer was routine work meetings such as team and lab meetings (8), then discussions that turn on details in a document (6) and information lookup (6), and smaller mentions of brainstorming, on-the-spot fact checks, informal settings like studying with friends (3), and English-language support (1). One participant wrote, “I would definitely use it in a research meeting where we are discussing a paper. Sometimes it’s easy to think a paper implies a certain claim, but it would be helpful if it was there to ground us” (P3).

In terms of *where they would not use Jarvis*, formal or high-stakes settings such as presentations, interviews, and large meetings were named most often (9), followed by private conversations with family and friends (6). Some found the agent too serious for casual talk (3) or unsuited to meetings with no shared material or confidential

Table 5: Interventions (Int.) by setting. Timing is turns from the problem utterance to the agent’s decision (hard ≤ 2 , soft ≤ 5 , late > 5). Correct (Corr.), Partial (Part.) and Wrong (Wr.).

Setting	Int.	Hard	Soft	Late	Corr.	Part.	Wr.	Cited
Speak	44	41	2	1	36	7	1	–
Raise hand	33	28	2	3	31	2	0	–
Source shown	27	21	3	3	26	1	0	27/27
Enthusiastic	40	34	1	4	38	2	0	39/40
Deferent	20	15	3	2	17	3	0	18/20
All	164	139	11	13	148	15	1	84/87

Table 6: Who resolved problems first. Agent: an intervention was generated; Participant: a person corrected it first; Nobody: neither.

Kind	<i>n</i>	Agent	Part.	None
Planted error	84	52	10	22
Spontaneous error	32	19	4	9
Question or gap	107	84	5	18
All	223	155	19	49

Table 7: Example moments from the user study. Each row shows what people said, as recorded, and what Jarvis said in reply. The underlined words are what Jarvis responded to. Omitted turns are marked. In (c) two participants shared one microphone.

Situation	What Jarvis did, at the turn where it did it
In time	P17: You said earlier that the images in the camera weren’t super high quality. Were there any tests [...] about actually reading text? Lead: I remember the task performance [...] the task includes object recognition, OCR, translation and reasoning parts. Lead: The OCR is, if you get a PDF file or the paper, they will recognize the letter. [...] Agent, go ahead. Page ten. → Jarvis: About OCR test scope, the study evaluated Optical Character Recognition on book titles and authors, not dense paragraphs. You’ll find this on Page 10. P17: Okay.
In time	P17: I think both is better, because if it was unevenly weighted that would be a huge distraction [...] on top of getting the actual visual space from the camera better. Lead: They also have some challenge [...] they design the system integrated in dual forward-facing cameras. Maybe they have some constraint for getting the image. P17: Trying to think, I feel like I probably hold it arm’s length away, bent a little bit, so maybe like a foot or so. P17: I can’t think in my head the conversion of that to centimeters. Lead: I see. So they are using the concept of the Harmon distance [...] → Jarvis: About Harmon distance definition, the Harmon distance is defined as the average comfortable reading distance of 36.8 cm, established across 233 individuals. You’ll find this on Page 5. P18: Yep. Lead: Yeah, that’s the Harmon distance definition.
Late	Lead: Or because the audio and the vision also use the same streaming protocol. P15/16: Oh, does it? Lead: Yeah, I think so. I didn’t remember exactly which audio protocol they are using in this work [...] P15/16: Great. Lead: So then let’s talk about the survey parts. They recruited forty people for this survey, paid crowd workers recruited on Prolific [...] P15/16: Okay. Lead: And how often they wear glasses and earbuds. Agent, go ahead. I see, I think the same Bluetooth protocol [...] → Jarvis: About streaming protocols, audio uses the Bluetooth Hands-Free Profile, while visual data is transmitted via Bluetooth Low Energy. You’ll find this on Page 6. P15/16: Yeah.

content (1). One noted that being corrected “over and over” about the paper was “a little embarrassing” (P3).

Answering the question about what *other moments they would want to support*, beyond correcting wrong facts, participants wanted the agent to remind them when they were stuck or had forgotten a detail (5), suggest new ideas or related work (3), fill a silence with basic information (3), and flag when the discussion drifted off topic (3). Two wanted it to search sources outside the document, and single participants asked it to show the page to someone who was lost, step in on unresolved questions, or answer only on a wake word. Two said the current scope was enough.

Finally, we note two behaviors that recurred in the records. First, because the agent showed its pending answer on screen before speaking in the hand-raise parts, a participant sometimes read it and said the fact first. The agent’s answer then only confirmed them, and the hand-raise indicator itself became a channel for information. Second, some interventions stated a fact already said (21 of 164), in two forms. (i) *A person corrected it before playback* (10): the intervention was needed when decided, but while the agent

waited for permission a participant said the same fact, so it only confirmed them. In one session the lead corrected themselves to “90 participants” and the agent gave the same number 3 seconds later. This is a cost of the hand-raise delay, with only one such case in the Speak part. (ii) *The agent repeated itself* (11, 7 played): several interventions from one wrong utterance (7), or an earlier fact repeated when the subtopic came back (3; in one session, asked where the model runs, the agent explained for the third time why it was chosen). The cooldown deliberately allows intervening again on the same topic. The problem is that the agent did not leave out what it had already said.

6 DISCUSSION AND LIMITATIONS

Planted errors and their validity. Our evaluation relies on planted errors in two places, and both are deliberately controlled. In CHI-180-proactive, events are regular by construction, with a fixed cue/failed-resolution/decision shape and no answer leakage; this regularity is what makes correctness and timing decidable. But

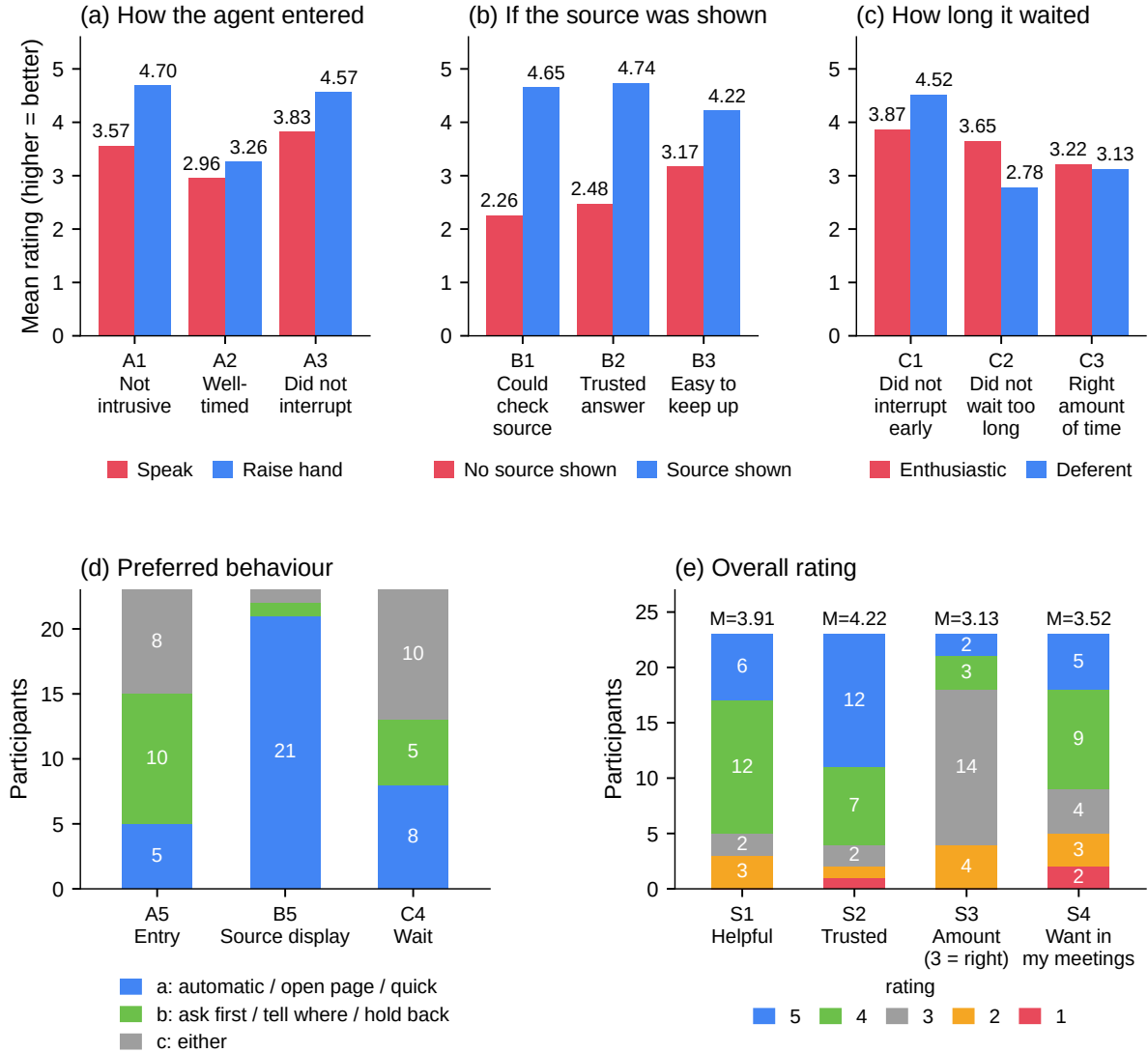


Figure 6: Participant ratings (N=23). (a-c) Mean rating per item for each pair of settings; negatively worded items (A1, A3, B3, C1, C2) are reverse-scored [3], so higher is better throughout. (d) Preferred behavior. (e) Overall ratings; S3 is centered (3 = about right).

real errors are messier, less isolated, and often hedged, so performance may not transfer directly to natural errors. In the live study, a member of the research team led every session, which guaranteed reactable moments but risks planting errors that are unusually easy to catch, phrased to suit the monitor. We take the same evidence to bound both concerns: the study also contains 32 *spontaneous* errors that participants made on their own, with no involvement from the lead or the event sheet and none of the dataset’s structure, and the agent handled these at a rate comparable to the planted ones (§5.3). We read this as the least confounded signal in our evaluation, and as evidence that performance does not depend on either the lead’s phrasing or the dataset’s regularity. To further limit bias, the lead rephrased cards into natural speech rather than reading

them verbatim, and all outcomes were coded from transcripts and reviewed against the paper, not scored by the lead.

Correction as a social threat. Publicly correcting a fact may expose an error in front of the group, and its cost grows with the status gap between the agent, or whoever it corrects, and the rest of the room. One participant captured this directly, describing being corrected “over and over” about the paper as “a little embarrassing” (P3). This reframes our results about the hand-raise mode as not only a floor-management mechanism but a face-management one: by withholding the correction until a participant grants the floor, it lets the group decide whether a correction is worth its interpersonal cost, and it makes the correction feel invited rather than imposed.

Calibrated trust. On planted moments where Jarvis should speak, it intervenes the vast majority of missing or incorrect facts, but is

not 100%. Its silence therefore cannot be read as evidence a statement is correct: Jarvis reduces the chance an error goes unchallenged but does not eliminate it, and participants remain responsible for their own claims. Some misses are deliberate, since a wrong spoken intervention is costly, so Jarvis abstains when uncertain and waits for a problem to persist. Others come from known failure modes such as reworded claims that reset the persistence count, ASR errors on names and numbers, or facts in tables the model cannot cite. Our goal is *calibrated* trust: participants can rely on what Jarvis says because it is usually right, while recognizing that its silence guarantees nothing.

Closed-book by design. Jarvis answers only from the uploaded document, using no tool calls, web search, or parametric knowledge, so every claim points to a source sentence participants can check. We prioritize verifiability over capability. The cost is that Jarvis cannot answer questions outside the document, and extending it to tools or retrieval while preserving source-level citation is an open problem. A related limit is context length: a long document or several documents can exceed the small models we use, so a larger corpus needs a larger model with a longer context window, which stresses the per-turn latency budget. Document size, model capacity, and latency thus trade off. Retrieval over the document rather than holding all of it in context would relax this, at the cost of a retrieval step.

Deterministic guards and fixed thresholds. Throughout the backbone, a language model proposes and deterministic logic checks: we verify the monitor’s sentence IDs against the document, recheck the self-repair model’s “resolved” verdict before closing a claim, and use persistence counts rather than a model to decide when to speak, so no single model output can trigger an irreversible spoken intervention on its own. Several guards depend on constants we set empirically, however, the persistence threshold, the claim-matching cutoff, the resolution-score fraction, and the timing windows, which form the kind of harness now common around AI agents. The right values may differ across groups, documents, and styles; learning them from data or adapting them per group is future work.

Failure modes. Jarvis’s patience depends on recognizing the same problem across turns, which requires the monitor to phrase claims consistently and the matcher to link them by embedding similarity. If the monitor rewords a claim, the match fails, the count resets, and Jarvis stays silent, the main reason it misses problems; detection and matching failures both appear as silence, making them hard to separate. Jarvis also inherits speech-pipeline errors: multi-speaker ASR can fail on the names and numbers it most needs, the wake word can misfire, and the system currently supports only English.

GPU deployment. We run the models on datacenter GPUs (Nvidia L40s). In the online-meeting setting this is reasonable: the audio already leaves each device for the meeting platform’s servers, so cloud processing adds no new exposure. Running the backbone locally would matter for offline or privacy-sensitive settings where audio does not otherwise leave the device, and would depend on even smaller-models.

Evaluation and platform scope. We evaluate Jarvis on CHI-paper discussions in Zoom, but neither is fundamental. The same

technique applies to financial filings, contracts, or specifications, though each has quirks that need evaluation (dense tables stress the rule to omit a citation when a value is not in a sentence; legal language stresses claim matching), and the speech and interaction modules would carry to Google Meet or Microsoft Teams. Our evaluation also assumes a remote meeting with per-participant audio streams. Co-located groups are harder: one shared acoustic space makes speaker separation [7] and diarization the bottleneck, and the agent becomes a voice in the room.

Privacy and consent. An agent that listens, reads a shared document, and speaks raises privacy and consent questions, which we address by design and deployment. Jarvis is a *visible* participant: it appears in the roster under its own name, has a video tile, and shows its state, so it is never a hidden listener. For consent, we follow the norm of AI note-takers, which announce themselves and are enabled with the group’s awareness, so disclosing the agent and joining the meeting is how a participant consents. Two aspects go beyond a passive note-taker. First, it processes an uploaded document, a second data stream, which is the group’s own material shared deliberately. Second, it speaks, and can correct a participant in front of the group, a social cost distinct from any data concern. Here our hand-raise mode doubles as a per-intervention consent mechanism, letting the group admit each intervention and decide how forward the agent should be.

7 CONCLUSION

We design the first real-time speech agent that can take part in a group conversation instead of waiting to be addressed. Jarvis reads a shared document, follows the discussion, and speaks only when the group misses or misstates a fact, giving people a chance to fix it themselves first and backing every claim with a source they can check on screen. Because the conversation is grounded in documents, we can measure objectively whether an intervention was right as well as its timing. Jarvis is an important step toward proactive speech agents that speak up when a group needs them, stay quiet when it does not, and when they intervene, can be measured against ground truth.

REFERENCES

- [1] ACM CHI. 2026. Selecting a Subcommittee. <https://chi2026.acm.org/authors/papers/selecting-a-subcommittee/>. Accessed 2026-09-08.
- [2] Mohammed Alsobay, David M. Rothschild, Jake M. Hofman, and Daniel G. Goldstein. 2025. Bringing Everyone to the Table: An Experimental Study of LLM-Facilitated Group Decision Making. *arXiv preprint arXiv:2508.08242* (2025). doi:10.1145/3816907 To appear at ACM CSCW 2026.
- [3] John Brooke. 1996. SUS: A ‘Quick and Dirty’ Usability Scale. In *Usability Evaluation in Industry*, Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland (Eds.). Taylor & Francis, London, 189–194.
- [4] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska Masson, Iain Mccowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2005. *The AMI Meeting Corpus: A Pre-announcement*. 28–39. doi:10.1007/11677482_3
- [5] Tuochao Chen, Nicholas Scott Batchelder, Alisa Liu, Noah A. Smith, and Shyamnath Gollakota. 2025. LlamaPIE: Proactive In-Ear Conversation Assistants. In *Findings of the Association for Computational Linguistics: ACL 2025*. 13801–13824. doi:10.18653/v1/2025.findings-acl.710
- [6] Tuochao Chen, Bandhav Veluri, Hongyu Gong, and Shyamnath Gollakota. 2026. AV-Dialog: Spoken Dialogue Models with Audio-Visual Input. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (Eds.). Association for Computational Linguistics, San Diego, California,

- United States, 42208–42225. doi:10.18653/v1/2026.acl-long.1954
- [7] Tuocho Chen, Qirui Wang, Bohan Wu, Malek Itani, Sefik Emre Eskimez, Takuya Yoshioka, and Shyamnath Gollakota. 2024. Target conversation extraction: Source separation using turn-taking dynamics. In *Interspeech 2024*. ISCA. <https://www.microsoft.com/en-us/research/publication/target-conversation-extraction-source-separation-using-turn-taking-dynamics/>
 - [8] Chun-Wei Chiang, Zhuoran Lu, Zhuoyan Li, and Ming Yin. 2024. Enhancing AI-Assisted Group Decision Making through LLM-Powered Devil’s Advocate. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (Greenville, SC, USA) (IUI ’24). Association for Computing Machinery, New York, NY, USA, 103–119. doi:10.1145/3640543.3645199
 - [9] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. Association for Computational Linguistics, 4599–4610. doi:10.18653/v1/2021.naacl-main.365
 - [10] DeepSeek-AI. 2026. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. arXiv:2606.19348 [cs.CL]. <https://arxiv.org/abs/2606.19348>
 - [11] Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A survey on proactive dialogue systems: problems, methods, and prospects. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence* (Macao, P.R.China) (IJCAI ’23). Article 738, 9 pages. doi:10.24963/ijcai.2023/738
 - [12] Yang Deng, Lizi Liao, Zhonghua Zheng, Grace Hui Yang, and Tat-Seng Chua. 2024. Towards Human-centered Proactive Conversational Agents. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) (SIGIR ’24). Association for Computing Machinery, New York, NY, USA, 807–818. doi:10.1145/3626772.3657843
 - [13] Rishikesh Devanathan, Varun Nathan, and Ayush Kumar. 2026. Mirage: A Diagnostic Framework for Evaluating the Realism of Synthetic Contact Center Dialogue Generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (Eds.). Association for Computational Linguistics, San Diego, California, United States, 25187–25227. doi:10.18653/v1/2026.findings-acl.1261
 - [14] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. arXiv:2410.00037 [eess.AS]. <https://arxiv.org/abs/2410.00037>
 - [15] Erik Ekstedt and Gabriel Skantze. 2020. TurnGPT: a Transformer-based Language Model for Predicting Turn-taking in Spoken Dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 2981–2990. doi:10.18653/v1/2020.findings-emnlp.268
 - [16] Erik Ekstedt and Gabriel Skantze. 2022. Voice Activity Projection: Self-supervised Learning of Turn-taking Events. In *Interspeech 2022*. 5190–5194. doi:10.21437/Interspeech.2022-10955
 - [17] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling Large Language Models to Generate Text with Citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6465–6488. doi:10.18653/v1/2023.emnlp-main.398
 - [18] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics* 10 (2022), 178–206. doi:10.1162/tacl_a_00454
 - [19] Michael Hansen. 2022. Piper: A Fast, Local Neural Text to Speech System. GitHub repository. <https://github.com/rhasspy/piper> Accessed September 7, 2026.
 - [20] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) (CHI ’99). Association for Computing Machinery, New York, NY, USA, 159–166. doi:10.1145/302979.303030
 - [21] Eric Horvitz and Johnson Apacible. 2003. Learning and reasoning about interruption. In *Proceedings of the 5th International Conference on Multimodal Interfaces* (Vancouver, British Columbia, Canada) (ICMI ’03). Association for Computing Machinery, New York, NY, USA, 20–27. doi:10.1145/958432.958440
 - [22] Eric Horvitz, Johnson Apacible, and Muru Subramani. 2005. Balancing awareness and interruption: investigation of notification deferral policies. In *Proceedings of the 10th International Conference on User Modeling* (Edinburgh, UK) (UM ’05). Springer-Verlag, Berlin, Heidelberg, 433–437. doi:10.1007/11527886_59
 - [23] Eric Horvitz, Paul Koch, and Johnson Apacible. 2004. BusyBody: creating and fielding personalized models of the cost of interruption. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work* (Chicago, Illinois, USA) (CSCW ’04). Association for Computing Machinery, New York, NY, USA, 507–510. doi:10.1145/1031607.1031690
 - [24] Stephanie Houde, Kristina Brimijoin, Michael Muller, Steven I. Ross, Dario Andres Silva Moran, Gabriel Enrique Gonzalez, Siya Kunde, Morgan A. Foreman, and Justin D. Weisz. 2025. Controlling AI Agent Participation in Group Conversations: A Human-Centered Approach. In *Proceedings of the 30th International Conference on Intelligent User Interfaces* (IUI ’25). Association for Computing Machinery, New York, NY, USA, 390–408. doi:10.1145/3708359.3712089
 - [25] Hangrui Hu, Xinfu Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, Xinyu Zhang, Pei Zhang, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-TTS Technical Report. arXiv preprint arXiv:2601.15621 (2026). doi:10.48550/arXiv.2601.15621
 - [26] Lingxiang Hu, Shurun Yuan, Xiaoting Qin, Jue Zhang, Qingwei Lin, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. 2025. MEETING DELEGATE: Benchmarking LLMs on Attending Meetings on Our Behalf. In *Proceedings of the Fourth Workshop on Bridging Human-Computer Interaction and Natural Language Processing (HCI+NLP)*, Su Lin Blodgett, Amanda Cercas Curry, Sunipa Dev, Siyan Li, Michael Madaio, Jack Wang, Sherry Tongshuang Wu, Ziang Xiao, and Diyi Yang (Eds.). Association for Computational Linguistics, Suzhou, China, 283–316. doi:10.18653/v1/2025.hcinlp-1.24
 - [27] Yuelin Hu, Jun Xu, Bingcong Lu, Zhengxue Cheng, Hongwei Hu, Ronghua Wu, and Li Song. 2026. MeetBench-XL: Calibrated Multi-Dimensional Evaluation and Learned Dual-Policy Agents for Real-Time Meetings. arXiv:2602.03285 [cs.AI]. <https://arxiv.org/abs/2602.03285>
 - [28] Koji Inoue, Bing’er Jiang, Erik Ekstedt, Tatsuya Kawahara, and Gabriel Skantze. 2024. Real-time and Continuous Turn-taking Prediction Using Voice Activity Projection. In *International Workshop on Spoken Dialogue Systems Technology (IWSDS 2024)*.
 - [29] Shamsi T. Iqbal and Brian P. Bailey. 2008. Effects of intelligent notification management on users and their tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) (CHI ’08). Association for Computing Machinery, New York, NY, USA, 93–102. doi:10.1145/1357054.1357070
 - [30] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The ICSI Meeting Corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03)*, Vol. 1. I–I. doi:10.1109/ICASSP.2003.1198793
 - [31] Maruchi Kim, Rasya Fawwaz, Zhi Yang Lim, Brinda Moudgalaya, Hexi Wang, Yuanhao Zeng, and Shyamnath Gollakota. 2026. VueBuds: Visual Intelligence with Wireless Earbuds. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (CHI ’26). Association for Computing Machinery, New York, NY, USA, Article 1668, 21 pages. doi:10.1145/3772318.3791322
 - [32] Soomin Kim, Jinsu Eun, Changhoon Oh, Bongwon Suh, and Joohwan Lee. 2020. Bot in the Bunch: Facilitating Group Chat Discussion by Improving Efficiency and Participation with a Chatbot. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI ’20). Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376785
 - [33] Soomin Kim, Jinsu Eun, Joseph Seering, and Joohwan Lee. 2021. Moderator Chatbot for Deliberative Discussion: Effects of Discussion Structure and Discus-sant Facilitation. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 87 (April 2021), 26 pages. doi:10.1145/3449161
 - [34] Harsh Kumar, Jonathan Vincentius, Ewan Jordan, and Ashton Anderson. 2025. Human Creativity in the Age of LLMs: Randomized Experiments on Divergent and Convergent Thinking. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI ’25). Association for Computing Machinery, New York, NY, USA, Article 23, 18 pages. doi:10.1145/3706598.3714198
 - [35] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
 - [36] Stephen Levinson and Francisco Torreira. 2015. Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology* 6 (06 2015). doi:10.3389/fpsyg.2015.00731
 - [37] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) (NIPS ’20). Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.
 - [38] Ting-En Lin, Yuchuan Wu, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. Duplex Conversation: Towards Human-like Interaction in Spoken Dialogue Systems. (2022), 3299–3308. doi:10.1145/3534678.3539209
 - [39] Tianjian Liu, Fanqi Wan, Jiajian Guo, and Xiaojun Quan. 2026. ProactiveEval: A Unified Evaluation Framework for Proactive Dialogue Agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (Eds.). Association for Computational Linguistics, San Diego, California, United States, 41068–41100. doi:10.18653/v1/2026.acl-long.1906
 - [40] Xingyu Bruce Liu, Shitao Fang, Weiyan Shi, Chien-Sheng Wu, Takeo Igarashi, and Xiang ‘Anthony’ Chen. 2025. Proactive Conversational Agents with Inner Thoughts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI ’25). Association for Computing Machinery, New York, NY, USA, Article 184, 19 pages. doi:10.1145/3706598.3713760

- [41] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2511–2522. doi:10.18653/v1/2023.emnlp-main.153
- [42] Ziyi Liu, Bahar Sarrafzadeh, Pei Zhou, Longqi Yang, Jieyu Zhao, and Ashish Sharma. 2026. ProMediate: A Socio-cognitive framework for evaluating proactive agents in multi-party negotiation. arXiv:2510.25224 [cs.CL] <https://arxiv.org/abs/2510.25224>
- [43] Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, Weiwen Liu, Yasheng Wang, Zhiyuan Liu, Fangming Liu, and Maosong Sun. 2025. Proactive Agent: Shifting LLM Agents from Reactive Responses to Active Assistance. In *The Thirtieth International Conference on Learning Representations*. <https://openreview.net/forum?id=sRIU6k2TcU>
- [44] Manqing Mao, Paishun Ting, Yijian Xiang, Mingyang Xu, Julia Chen, and Jianzhe Lin. 2024. Multi-User Chat Assistant (MUCA): a Framework Using LLMs to Facilitate Group Conversations. arXiv:2401.04883 [cs.CL] <https://arxiv.org/abs/2401.04883>
- [45] Gloria Mark, Daniela Gudith, and Ulrich Klocke. 2008. The cost of interrupted work: more speed and stress. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy) (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 107–110. doi:10.1145/1357054.1357072
- [46] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes. arXiv:2203.11147 [cs.CL] <https://arxiv.org/abs/2203.11147>
- [47] Sewon Min, Kalpesh Krishna, Xinxin Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12076–12100. doi:10.18653/v1/2023.emnlp-main.741
- [48] Soumyajit Mitra, Prabhat Pandey, Abhinav Jain, Shanmukha Sahithi, and K V Vijay Girish. 2026. Adaptive Turn-Taking for Real-time Multi-Party Voice Agents. arXiv:2606.13544 [eess.AS] <https://arxiv.org/abs/2606.13544>
- [49] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. COVID-QA: A Question Answering Dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Karin Verspoor, Kevin Bretonnel Cohen, Mark Dredze, Emilio Ferrara, Jonathan May, Robert Munro, Cecile Paris, and Byron Wallace (Eds.). Association for Computational Linguistics, Online. <https://aclanthology.org/2020.nlpcovid19-acl.18/>
- [50] Vahid Noroozi, Somshubra Majumdar, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. 2023. Stateful Conformer with Cache-based Inference for Streaming Automatic Speech Recognition. arXiv preprint arXiv:2312.17279. arXiv:2312.17279 [cs.CL] doi:10.48550/arXiv.2312.17279
- [51] NVIDIA. 2026. Nemotron ASR Streaming: Model Card for nvidia/nemotron-speech-streaming-en-0.6b. Hugging Face model card. <https://huggingface.co/nvidia/nemotron-speech-streaming-en-0.6b> Accessed September 7, 2026.
- [52] Jeeseun Oh, Woosuk Kim, Sungbae Kim, Hyeonjeong Im, and Sangsu Lee. 2024. Better to Ask Than Assume: Proactive Voice Assistants' Communication Strategies That Respect User Agency in a Smart Home Environment. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 846, 17 pages. doi:10.1145/3613904.3642193
- [53] OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card. arXiv:2508.10925 [cs.CL] <https://arxiv.org/abs/2508.10925>
- [54] Aaron T Parisi, Nithum Thain, Alden Hallak, Vivian Tsai, and Crystal Qian. 2026. Real-Time Group Dynamics with LLM Facilitation: Evidence from a Charity Allocation Task. In *Proceedings of the 2026 ACM Conference on Fairness, Accountability, and Transparency (FAccT '26)*. Association for Computing Machinery, New York, NY, USA, 25–59. doi:10.1145/3805689.3812253
- [55] Gil Pasternak, Dheeraj Rajagopal, Julia White, Dhruv Atreja, Matthew Thomas, George Hurn-Maloney, and Ash Lewis. 2026. Beyond Reactivity: Measuring Proactive Problem Solving in LLM Agents. arXiv:2510.19771 [cs.AI] <https://arxiv.org/abs/2510.19771>
- [56] Deep Anil Patel, Iain Melvin, Christopher Malon, and Martin Renqiang Min. 2026. DiscussLLM: Teaching Large Language Models When to Speak. arXiv:2508.18167 [cs.CL] <https://arxiv.org/abs/2508.18167>
- [57] Haosheng Qian, Yixing Fan, Jiafeng Guo, Ruqing Zhang, Qi Chen, Dawei Yin, and Xueqi Cheng. 2025. VeriCite: Towards Reliable Citations in Retrieval-Augmented Generation via Rigorous Verification. (2025), 47–54. doi:10.1145/3767695.3769505
- [58] Qwen Team. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] arXiv:2505.09388
- [59] Qwen Team. 2026. Qwen3.5: Towards Native Multimodal Agents. <https://qwen.ai/blog?id=qwen3.5>
- [60] Qwen Team. 2026. Qwen3.6-35B-A3B: Agentic Coding Power, Now Open to All. <https://qwen.ai/blog?id=qwen3.6-35b-a3b>
- [61] Qwen Team. 2026. Qwen3.8-27B. <https://huggingface.co/Qwen/Qwen3.8-27B>. Model card and weights, revision 1d4bf02ff6012fd82039f2fa52739d0dd7c60c0; accessed September 10, 2026.
- [62] Andrew Reece, Gus Cooney, Peter Bull, Christine Chung, Bryn Dawson, Casey Fitzpatrick, Tamara Glazer, Dean Knox, Alex Liebscher, and Sebastian Marin. 2023. The CANDOR Corpus: Insights from a Large Multimodal Dataset of Naturalistic Conversation. *Science Advances* 9, 13 (2023), ead3197. doi:10.1126/sciadv.adf3197
- [63] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410
- [64] Bradley J. Rhodes and Thad Starner. 1996. Remembrance Agent: A Continuously Running Automated Information Retrieval System. In *Acquisition, Learning and Demonstration: Automating Tasks for Users (AAAI Technical Report SS-96-02)*. AAAI Press, Menlo Park, CA, 122–125.
- [65] Federico Ruggeri, Mohsen Mesgar, and Iryna Gurevych. 2023. A Dataset of Argumentative Dialogues on Scientific Papers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 7684–7699. doi:10.18653/v1/2023.acl-long.425
- [66] Harvey Sacks, Emanuel Schegloff, and Gail Jefferson. 1974. A Simple Systematic for the Organisation of Turn Taking in Conversation. *Language* 50 (12 1974), 696–735. doi:10.2307/412243
- [67] Emanuel Schegloff, Gail Jefferson, and Harvey Sacks. 1977. The Preference for Self-Correction in the Organization of Repair in Conversation. *Language* 53 (06 1977), 361–382. doi:10.2307/413107
- [68] David Scripka. 2022. openWakeWord. GitHub repository. <https://github.com/dscripka/openWakeWord> Accessed September 7, 2026.
- [69] Silero Team. 2024. Silero VAD: Pre-trained Enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier. GitHub repository. <https://github.com/snakers4/silero-vad> Accessed September 7, 2026.
- [70] Yuanbo Tang, Huaze Tang, Tingyu Cao, Lam Nguyen, Anping Zhang, Xinwen Cao, Chunkang Liu, Wenbo Ding, and Yang Li. 2026. ProAgent-Bench: Evaluating LLM Agents for Proactive Assistance with Real-World Data. arXiv:2602.04482 [cs.HC] <https://arxiv.org/abs/2602.04482>
- [71] Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyam-nath Gollakota. 2024. Beyond Turn-Based Interfaces: Synchronous LLMs as Full-Duplex Dialogue Agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 21390–21402. doi:10.18653/v1/2024.emnlp-main.1192
- [72] Ruiyi Wang, Hao-fei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Yonatan Bisk, Graham Neubig, and Hao Zhu. 2024. SOTOPIA- π : Interactive Learning of Socially Intelligent Language Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 12912–12940. doi:10.18653/v1/2024.acl-long.698
- [73] Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. 2025. CollabLLM: From Passive Responders to Active Collaborators. In *International Conference on Machine Learning (ICML)*.
- [74] Sirui Xia, Xintao Wang, Jiaqing Liang, Yifei Zhang, Weikang Zhou, Jiayi Deng, Fei Yu, and Yanghua Xiao. 2025. Ground Every Sentence: Improving Retrieval-Augmented LLMs with Interleaved Reference-Claim Generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 969–988. doi:10.18653/v1/2025.findings-naacl.55
- [75] Ke Xu, Yuhao Wang, and Yu Wang. 2026. From Reactive to Proactive: Assessing the Proactivity of Voice Agents via ProVoice-Bench. arXiv:2604.15037 [cs.AI] <https://arxiv.org/abs/2604.15037>
- [76] Bufang Yang, Yunqi Guo, Lili Xu, Zhenyu Yan, Hongkai Chen, Guoliang Xing, and Xiaofan Jiang. 2025. SocialMind: LLM-based Proactive AR Social Assistive System with Human-like Perception for In-situ Live Interactions. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 1, Article 23 (March 2025), 30 pages. doi:10.1145/3712286
- [77] Bufang Yang, Lili Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. 2025. ContextAgent: Context-Aware Proactive LLM Agents with Open-World Sensory Perceptions. *Advances in Neural Information Processing Systems* 38 (2025), 1–10.
- [78] Shu Yang, Difei Xu, Jiaxin Pei, and Di Wang. 2026. ProACT: Towards Breakdown-Aware Proactive Agent in Multi-User Collaboration. arXiv:2607.03730 [cs.CL] <https://arxiv.org/abs/2607.03730>
- [79] Sunkeyeung Yoo. 2026. Read the Room: Exploring When LLM Agents Should Participate in Everyday Mixed-Context Group Chats on Mobile Messaging Platforms.

- In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems (CHI EA '26)*. Association for Computing Machinery, New York, NY, USA, Article 546, 6 pages. doi:10.1145/3772363.3798392
- [80] Nima Zargham, Leon Reicherts, Michael Bonfert, Sarah Theres Voelkel, Johannes Schoening, Rainer Malaka, and Yvonne Rogers. 2022. Understanding Circumstances for Desirable Proactive Behaviour of Voice Assistants: The Proactivity Dilemma. In *Proceedings of the 4th Conference on Conversational User Interfaces (Glasgow, United Kingdom) (CUI '22)*. Association for Computing Machinery, New York, NY, USA, Article 3, 14 pages. doi:10.1145/3543829.3543834
- [81] Neil Zeghidour, Eugene Kharitonov, Manu Orsini, Václav Volhejn, Gabriel de Marmiesse, Edouard Grave, Patrick Pérez, Laurent Mazaré, and Alexandre Défossez. 2025. *Streaming Sequence-to-Sequence Learning with Delayed Streams Modeling*. arXiv preprint arXiv:2509.08753. Kyutai. arXiv:2509.08753 [cs.CL] doi:10.48550/arXiv.2509.08753
- [82] Harry Zhang, Kurt Partridge, Pai Zhu, Neng Chen, Hyun Jin Park, Dhruuv Agarwal, and Quan Wang. 2025. GraphemeAug: A Systematic Approach to Synthesized Hard Negative Keyword Spotting Examples. In *Proceedings of Interspeech 2025*. 2680–2684. doi:10.21437/Interspeech.2025-1038
- [83] Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. 2025. LongCite: Enabling LLMs to Generate Fine-grained Citations in Long-Context QA. (July 2025), 5098–5122. doi:10.18653/v1/2025.findings-acl.264
- [84] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too?. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2204–2213. doi:10.18653/v1/P18-1205
- [85] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 5905–5921. doi:10.18653/v1/2021.naacl-main.472
- [86] Shaoze Zhou, Diana Nelly Rivera Rodriguez, Pedro Remior, Joaquin Frangi, Lingyao Li, Renkai Ma, Janet G. Johnson, Christine Lisetti, and Chen Chen. 2026. Exploring Needs and Design Opportunities for Proactive Information Support in In-Person Small-Group Conversations. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems (CHI EA '26)*. Association for Computing Machinery, New York, NY, USA, Article 274, 7 pages. doi:10.1145/3772363.3799279
- [87] Zoom. n.d.. Media Parameter Definitions: Realtime Media Streams (RTMS). Zoom Developer Documentation. <https://developers.zoom.us/docs/rtms/media-parameter-definition/> Accessed September 7, 2026.
- [88] Wazeer Deen Zulfikar, Samantha Chan, and Pattie Maes. 2024. Memoro: Using Large Language Models to Realize a Concise Interface for Real-Time Memory Augmentation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 450, 18 pages. doi:10.1145/3613904.3642450

A SPEECH PROCESSING MODULE: IMPLEMENTATION DETAILS

• **Per-participant audio streaming.** The module receives a separate audio stream for each participant, identified by participant ID. Each stream is processed in parallel by voice-activity detection, wake-word detection, and automatic speech recognition (ASR). In our Zoom implementation, we receive 16-kHz mono audio from Zoom Realtime Media Streams (RTMS) [87] in 20-ms frames. The audio is encoded as 16-bit pulse-code modulation (PCM16).

• **Voice activity detection (VAD).** Transcriptions operate at a word level which is much slower than activity detection. So we analyze each participant’s audio independently and emit acoustic events rather than words: speech onset, sustained speech, short silence, and long silence. We run a Silero VAD instance [69] per participant on 30-ms frames with a speech-probability threshold of 0.5. Sustained speech requires 0.5 s of continuous voicing, with a 0.5-s hangover to bridge brief gaps within an utterance. We use 0.5 s and 2.0-second thresholds for short and long silence, respectively.

• **Wake-word detection.** We use a two-stage detector for “hey agent” detection. We continuously run a first-stage model that proposes candidates, then a verifier re-scores the most recent audio and decides whether to fire. Because the detector evaluates overlapping windows, it sees incomplete prefixes such as “hey ag”. We train these as negatives and fire only on the complete phrase to avoid premature triggers. Two gates provide additional filtering: a contextual filter rejects patterns such as “the agent” and “say agent,” while a high-precision text fallback recovers acoustic misses when the transcript contains “hey agent.” We build on OPENWAKEWORD [68]. The first stage combines fine-tuned speech embeddings with a lightweight head, scores every 80-ms frame, and invokes the verifier when the score is above 0.90. The verifier is a 1.2M-parameter CNN-GRU trained from scratch on 40-band log-Mel windows. It rescores the most recent 1.0- and 1.2-s windows and fires when either exceeds 0.90. Following GRAPHEMEAUG [82], we synthesize confusable negatives; to reduce prefix bias, windows containing less than half the keyword are labeled negative, while ambiguous half-but-not-complete windows are omitted. Positive “hey agent” examples use three TTS systems (Piper [19], Kyutai TTS [81], and Qwen3-TTS [25]) to reduce speaker bias. Kyutai is conditioned on recordings from 410 real speakers, and the synthesized examples are overlaid on conversational audio. On 285.4 h of held-out AMI [4] and CANDOR [62] audio, we achieve 99.08% recall with 0.077 false accepts per hour.

• **Per-speaker streaming ASR and turn segmentation.** We avoid Zoom’s transcript because its latency ranges from seconds to tens of seconds and carries no timestamp usable for turn-taking. Instead we use `nvidia/nemotron-speech-streaming-en-0.6b`, a cache-aware streaming FastConformer-RNNT with attention context [50, 51], processing audio in 1.12-s steps with roughly one second of right context. We run one recognizer per participant and detect a turn end when a dedicated endpointing Silero instance observes 500 ms of silence, with a 700-ms wall-clock watchdog as backup if PCM delivery stalls. Median latency from speech end to a turn-end detection was 224 ms.

• **Speech synthesis and delivery.** We synthesize text to speech (TTS) with Kyutai TTS `kyutai/tts-1.6b-en_fr` and the Mimi codec (24-kHz mono, 80-ms frames); batch chunks are capped at 150 then 300 characters on sentence boundaries. Streamed synthesis starts after three words or 0.25 s and pauses once more than 0.3 s of audio is buffered. A prefetch thread applies a 0.4-s jitter buffer and sends 20-ms packets at the native 24-kHz rate. After warm-up, the latency from the first text input to audible playback is 0.70–0.95 s.

B MONITOR MODULE: ANSWERING DIRECT QUESTIONS BASED ON CONTEXT

This model (temperature 0.7) does standard document question answering: given a question, find the answer in the document. The main challenge is deciding whether a question was actually asked, since wake-word detection is imperfect in live conversation.

The model answers complete questions the document supports; abstains when the document cannot answer, with the fixed reply “That isn’t covered in the document”, and skips anything that is not a question for Jarvis, including incomplete fragments and talk

between participants (an aside, correction, or thinking aloud, e.g., “never mind, scroll up”). It is given the last k turns as context, which it uses to resolve references such as “my previous question” or “that point.” When it answers, it writes concisely in its own words, at most three sentences since it is read aloud, and cites the document sentences; the page and keyword hint are assembled as for proactive interventions, without the topic preface ($k = 8$ deployed).

C INTERACTION MODULE: ANSWERING DIRECT QUESTIONS

A direct question is addressed to the agent, so the answer plays as soon as it is ready, subject only to the barge-in and recovery behavior described in §3.3.1. The key challenge is in recovering the human participant’s complete question, given humans can pause mid-phrase or the question might have multiple sentences with short silences between them.

Since wake-word detection runs separately from automatic speech recognition (ASR), our interaction module must match each wake event to the ASR turn that contains the question. The match uses both speaker and time: the turn must come from the participant who triggered the wake word, and its time interval must overlap the wake position or start shortly after it. Once matched, the turn is labeled a *direct* chunk. If it contains only “hey agent,” the system waits briefly for the question to arrive in the next turn. Short follow-up turns from the same speaker remain part of the same direct question while the gap is short or VAD still detects speech. The continuation ends after a long gap, a non-backchannel utterance from another participant, or a hard timeout.

Waiting for the continuation window adds latency, even when the question is already complete. We therefore generate speculatively. After the speaker pauses briefly, the interaction model sends a <Silence> marker, allowing the proactive backbone model to start generating from the text received so far while the continuation window remains open. If the speaker continues, the new text is added to the same generation and the backbone model regenerates the answer. If a new question starts, it creates a new generation that supersedes the old one: queued output from the old generation is dropped and any active playback is canceled. Once the continuation window closes, the Interaction Model sends <Complete>. This ensures that even a late ASR tail arriving after the speculative generation is flushed is added to the same question.

Implementation details. A wake event binds to a turn from the same participant when the turn overlaps the wake sample position within ± 300 ms or begins within the following 5 s. Unbound events expire after 20 s of audio time. We classify a turn as wake-only when it contains at most three tokens and lasts at most 2.5 s, opening a 5 s grace window. A later turn from the same speaker is accepted as a continuation within 4.5 s, which corresponds to a 3 s turn gap plus 1.5 s of ASR latency, or while VAD remains voiced, up to a 30 s session-clock limit. Backchannels are detected with a fixed, hand-curated list of 42 tokens: interjections, fillers, and acknowledgements such as *yeah*, *okay*, *mm-hmm*, *uh-huh*, *right*, *sure*, *cool*, and *wow*.¹ A turn counts as a backchannel when, after punctuation

¹The full list: yeah, ok, okay, mm, mmm, hm, hmm, hmmm, uh, huh, ha, hah, haha, ah, um, umhum, uhuh, right, nice, good, fine, oh, really, god, hey, so, well, all, cool, wow, yep, yo, eh, sure, gotcha, and, yes, ow, yum, sign, laugh, shoot. Tokens are matched

is removed, it has at most four words and every word is in the list. We send the speculative <Silence> after 0.5 s of silence. If the transmitted text already covers the VAD speech endpoint within a 700 ms margin, we close the generation immediately; otherwise we close it 1.5 s later.

D SYNTHETIC DATASET CONSTRUCTION DETAILS

This section gives the construction details behind CHI-180 (§4): the source documents, the fact and question taxonomies, the pairing and placement logic, the generator screening, and the leakage safeguards.

D.1 Source Documents

We use 180 CHI papers drawn from the CHI 2026 guide to selecting a subcommittee [1], which lists example past papers for each of the 18 subcommittees. We took ten example papers from each subcommittee (the smallest yielded ten), giving 180 papers that cover CHI’s topic areas without bias toward any one.

D.2 Event Pairing and Placement

Each speak-event has a silent counterpart that acts as its control, but the two pairs are built differently. A resolved-question reuses a gap item: the same question, answered inside the conversation rather than left open. A self-correction does not reuse a wrong event’s claim; it is a different claim, built by the same operators and given the same three-turn shape, with a participant repair in place of the turns that would adopt the value.

This difference dictates placement. A resolved-question repeats an item the agent may already have answered elsewhere, so in a single conversation its silence could stem from the agent’s own repetition suppression rather than its reading of the group; we therefore isolate resolved-question events in a separate conversation. The eight planted claims are eight distinct facts with no such carry-over, so wrong and self-correction events can share the main conversation.

Gap events follow a taxonomy of six sub-types: explicit fact recall, missing comparator, missing qualifier, missing metric definition, unresolved attribution, and partial fact recall. Each has the same shape (a cue, a failed attempt to resolve it, and an answer that sits in a short span of the document), which keeps the correctness of an intervention independent of subjective judgment.

D.3 Stage 1 Details: Facts and QA Items

Each paper yields 300 to 470 extracted facts, each a single statement anchored to its page, passage, character span, and a hash of the passage text. The hash guards against re-parsing: if a later parse cuts passages differently, a stored span could point at the wrong text, and the mismatched hash flags the item rather than letting it be silently wrong. Unanswerable items are separately verified against the full text to confirm no answer exists. We use Qwen3-235B-A22B-Instruct-2507 [58] for this stage.

The twelve-item count and four answer types come from QASPER [9], whose papers carry 3.2 questions on average and after lowercasing and removing punctuation, so “mm-hmm” and “uh-huh” match as two tokens each.

twelve at most; we read that maximum as the number of genuine information needs a paper can support. QASPER’s own type proportions are 51.8% extractive, 24.2% abstractive, 13.9% boolean, and 10.2% unanswerable; we instead balance the mix (4/4/2/2) so each type occurs often enough within one paper to be analyzed on its own. Each item is used twice: on its own as a question-answering item scored against the gold answer, and inside the conversation as an event (the ten answerable items become gap events, the two unanswerable ones become unanswerable events, and four of the ten gaps are reused as resolved-question events).

D.4 Stage 2 Details: Error Validity Checks

For self-correction events, the other speaker replies briefly on the turn after the erroneous claim, and the speaker who made the claim repairs it on the turn after that, two turns after the claim. Every planted claim must pass four deterministic checks: it must be *false* (the wrong value appears nowhere in the extracted facts, and the entities in a misattributed fact appear nowhere in the paper); it must not be *decidable from the conversation alone* (it contradicts nothing said earlier, since that would allow an agent to correct it without reading the paper); it must not *interfere* with the twelve QA items; and it must not *leak* (the correct value appears nowhere else in the conversation).

D.5 Stage 3 Details: Plan, Personas, and Generator Screening

The main conversation is 220 turns and carries 10 gap, 2 unanswerable, 4 wrong, and 4 self-correction events; the resolved-question conversation is 60 turns with cues at turns 15, 25, 35, and 45. For each main-conversation event the plan records three turns: the cue, the turn on which the other speaker fails to resolve it, and the turn on which the information would have been used in a decision. In the resolved-question conversation each event spans three turns (one speaker asks, the other answers exactly as the document states and quotes the source, the asker acknowledges the question is settled), so silence is correct throughout.

Speaker profiles are in PersonaChat [84] form, derived from ArgSciChat dialogues [65]; both speakers see the same public history, persona, and available facts.

Generator screening. The four candidates are open-weight so the corpus can be regenerated without a proprietary API: Qwen3-235B-A22B [58], GPT-OSS-120B [53], Qwen3.8-27B [61], and DeepSeek-V4-Flash-0731 [10]. Assigning one model to each speaker role with order counted gives 16 conditions; each generated 200 turns over each of three documents. A blind G-Eval-style judge [41] (Claude Sonnet 5) scores eight-turn windows on naturalness, coherence, engagingness, and groundedness, and we compare these against windows of real conversation by Jensen–Shannon divergence following Mirage [13], with ICSI [30], CANDOR [62], and ArgSciChat as references (38 windows each; groundedness scored on ArgSciChat only, where the judge also sees the source paper). Table 2 reports divergences against ArgSciChat. Naturalness is worst when Qwen3-235B asks (its three asker conditions give the three largest values, 0.272, 0.215, 0.167, versus 0.156, 0.090, 0.070 when it answers), and engagingness is worst wherever Qwen3.8-27B appears (the four largest values in that column are all its conditions), confirming that role placement, not just model capacity, drives realism.

Preventing leakage. Three constraints prevent the conversation from revealing its own answers. Each turn can use only the facts in the local pool assigned to its segment. We withhold the facts that answer any answerable event from every segment’s pool, along with any other fact that could reveal the answer. A validator then checks every generated turn: numerals must appear in an allowlist, planted errors are tracked in a ledger, and no turn other than a cue turn may state a gold answer. Any turn that fails these checks is rejected and regenerated.

PROMPTS OF THE PROACTIVE BACKBONE

The four model stages of §3.2 share one model and one document representation. Every system message begins with the same document block, followed by the stage’s own instructions, so the serving engine reuses the cached document prefix across stages. Every stage returns a JSON object whose schema is enforced during decoding. Table 8 lists the per-stage settings. Runtime slots are shown in braces; the bold lines are the section headers of the prompt text.

Monitor

The monitor runs on every ordinary turn. The system message follows the document block. The user message carries the recent turns and the new utterance. The system message is in Table 9 and the user message in Table 10.

Self-repair check

The self-repair check runs in parallel with the monitor whenever the ledger holds an open claim. Open claims are listed as [C#] entries. A deterministic test (§3.2.3) confirms each proposed resolution before the claim is closed. The prompt is in Table 11.

Grounded generation

Grounded generation runs only when the gate fires. The intent is derived from the monitor status (incorrect_info to correct, missing_info to missing, not_in_paper to not_in_paper). The {answer_bias} slot holds the sentence “If the needed facts do appear in the sentences, you must answer — uncertainty is not a reason to abstain.” on turns where the monitor supplied answer_sids for the claim, and is empty otherwise. Cited sentence IDs are resolved to their page and text before delivery. The spoken form is “About {topic}, {text} You’ll find this on {Page N}; search for ‘{keyword}’ to confirm.” The prompt is in Table 12.

Direct answering

Direct answering handles wake-word turns and bypasses the monitor, ledger, and gate. The wake phrase is removed before the turn reaches the model. The last eight turns are supplied as context only. The prompt is in Table 13.

JUDGE PROMPT FOR DIRECT-RESPONSE ACCURACY

Accuracy was judged by Claude Sonnet 5, which read one batch file (130 items) at a time and assigned one label to each item. The prompt is identical for the three datasets; dataset-specific context is passed not in the prompt but in the gold_origin field of each item

Table 8: Per-stage decoding settings. The schema column lists the fields the serving engine enforces. Fields marked * are required.

Stage	Temp.	Max tokens	Enforced JSON fields
Monitor	0.1	1024	status* (one of six), claim*, evidence, predicted_answer, answer_sids (int, ≤ 3)
Self-repair check	0.1	256	resolved* (list of cid, by, by_sids (int, ≤ 3))
Grounded generation	0.7	256	action* (speak, abstain), text*, topic, source, quote_sids* (int)
Direct answering	0.7	256	action* (answer, abstain, skip), text*, quote_sids (int)

record. The judge is given the question, the gold answer, the list of equally valid answers (other_refs), the answer type, the system’s answer, and the token F1. The F1 comes with the instruction that it is structurally low even for correct answers and must not pull the judgment toward WRONG. The judge writes the TSV itself and reports only the counts; at the merge step the row count, the label set, and the order of item ids are checked against the batch, and a batch with any mismatch is rejected. The full prompt is in Table 14.

The gold_origin string carried by each record is given per dataset in Table 15. Of the answer_type values, multi_answer, program, and span do not occur in the three datasets of this paper; they serve other datasets, and the prompt is reproduced unchanged.

LLM JUDGE PROMPTS

Two evaluations in this paper use an LLM judge (Claude Sonnet 5). This appendix reproduces the rubric each judge received. The rubric text of the direct-answer judge is verbatim from the run. Instructions that named local files, directories, or output file names have been replaced by a description of the same inputs and outputs, and these replacements are noted where they occur. Runtime slots are shown in braces.

Judge for direct-answer accuracy

The judge of §5 reads batches of 130 records and assigns one label per record. One rubric serves all three datasets. Dataset-specific context travels inside each record (the gold_origin string, further reference answers, and the answer type), not in the prompt. Table 16 reproduces the rubric. Two harness instructions are omitted here. The original named the batch file to read and the output file to write. Their content is described by the first and last paragraphs of the table.

User-study intervention judging

Interventions in the live study (§5.3) were judged one session at a time. The judge received the instruction in Table 17 together with the rubric in Table 18, one card per intervention, the session’s event sheet, the session transcript, and the paper text with page markers. The original instruction named the local files holding each of these inputs and the output files to write. Those names are replaced here by descriptions of the same inputs and outputs. The authors reviewed every label.

Problem-utterance labelling

The problem utterances that the timing and coverage measures of §5.3 are computed against were labelled over the whole transcript of each session, after the interventions had been judged, with the instruction in Table 19. File names are again replaced by descriptions.

Judge for proactive-behavior events

The judge of §5.2 scores each planted event with one call to Claude Sonnet 5. The system message is the rubric in Table 20 followed by the full paper text; the user message gives the event (for a question, its text and the gold answer, or a note that the paper does not report it, in which case both silence and saying so are fully appropriate; for a claim, the assertion and what the paper supports) and the transcript from two turns before the onset to five after it, with the agent’s interventions inserted where they arrived and the one attributed to the event marked [EVALUATED]. Content is judged against the paper, not against the gold string. Table 20 omits three output fields that the analysis does not use (a grounding flag, an evidence quote, a note).

Table 9: Monitor: system message.

Monitor prompt (system message)
You monitor a live discussion about one section of a research document. You are given that section's verbatim text as the ground truth. Compare what the participants are saying against that text and report the topic's status. Report accurately and literally — you only report; a separate component decides whether to speak. Choose exactly one status: - <code>off_document</code>: the new utterance is not about the document's content at all — meeting logistics, scheduling, who does what, tooling, side chat. It asserts nothing the text could contradict and expresses no need for a fact the text could supply. Leave 'claim', 'evidence' and 'predicted_answer' empty. Document talk earlier in the conversation does not make the new utterance document talk. - <code>not_in_paper</code>: a participant asserts something as this document's own content — attributed to it ("in this study", "this research", "the paper", "they found") or stated as its finding in the middle of the document discussion — but the section text does not contain or support it. This is not <code>off_document</code>: the speakers believe the document says it, so staying silent lets a false version of the document spread. Set 'claim' to the asserted point (same rules as below) and 'evidence' to the participant's assertion; leave 'predicted_answer' empty (there is no document value to give). A mention of other papers or systems that is not presented as this document's content is still <code>off_document</code>, and a claim the text explicitly contradicts is <code>incorrect_info</code>, not <code>not_in_paper</code>. - <code>incorrect_info</code>: a participant asserts something that contradicts the section text (a wrong number/quantity, a wrong or overstated capability, a wrong name/term/method, a misattributed result or dataset, a reversed cause-and-effect, or any claim the text explicitly refutes). - <code>missing_info</code>: the participants are asking about / unsure about / clearly need a specific fact, and the section text contains a specific statement that directly answers that exact point, but no one has stated it. strict test: you must be able to point to the exact phrase or value in the section text that answers their specific question. If you cannot copy a phrase/value from the section text that directly answers the specific point, it is not <code>missing_info</code>. - <code>stale</code>: the participants are stuck, circling, or unsure about a point, and the section text does not contain a specific statement that resolves that point — i.e. the section is about the same topic but does not state the particular number, value, or detail they are asking for. Being on the same topic is not enough to be <code>missing_info</code>. - <code>on_progress</code>: the discussion is accurate and progressing, consistent with the section text. The status describes the new utterance, read in its conversational context: <code>incorrect_info</code> needs the contradicting assertion to be in it, and <code>missing_info/stale</code> need the question or uncertainty to be expressed in it. An utterance that neither asserts nor asks anything about the document's content is <code>off_document</code> — no matter how much document discussion sits in the recent turns; one that asserts document content the text does not back is <code>not_in_paper</code>. claim — the single most important field. Set 'claim' to the point the participants are on: the specific question they are trying to answer, or the specific matter they are getting wrong. Write it as a short, self-contained noun phrase or question naming the subject under discussion — the thing itself, not the answer to it and not anyone's wording of it. * Write it the same way every turn the discussion stays on that point, so the same point is recognisable across turns. Do not re-word it turn to turn. * Do not copy a sentence from the document into 'claim'. * Do not quote the participant in 'claim'. * The status does not change how 'claim' is written: if the group is asking about X and then says something wrong about X, 'claim' is X both times. * Good: "the dataset size used for training" "which baseline the method is compared against" "how many participants the study recruited" * Bad: "recruiting 24 participants through university mailing lists and running two sessions with each" (that is the answer, not the point) * Bad: "24 participants seems like a small sample, I think." (that is the participant's wording, not the point) evidence: set 'evidence' to the participant's own words that prompted this status — the incorrect statement for <code>incorrect_info</code>, the question or uncertainty for <code>missing_info/stale</code>. Empty string for <code>on_progress</code>. grounding: set 'predicted_answer' to the document-correct value copied verbatim from the section text — for <code>incorrect_info</code> the correct value that should replace the wrong one, for <code>missing_info</code> the answering fact. Leave it an empty string for <code>on_progress</code>, <code>stale</code> and <code>not_in_paper</code>. When that value was copied from the numbered sentences above, also set 'answer_sids' to the [S#] id(s) it came from — at most 3, and only ids you actually copied from. omit the field entirely when the value comes from a table, a figure caption, or anywhere you cannot point at a specific sentence. An omitted 'answer_sids' is a correct answer; a guessed id is not.

Table 10: Monitor: user message.

Monitor prompt (user message)
Topic entire document keywords: Recent turns {recent turns, one "[Turn k] speaker: text" line each} New utterance {utterance} Output (JSON only) {"status": "on_progress missing_info incorrect_info stale off_document not_in_paper", "claim": "<the point under discussion — the subject, not the answer, not a quote>", "evidence": "<the participant's own words that prompted this; empty if n/a>", "predicted_answer": "<document-correct value, verbatim; empty string if n/a>", "answer_sids": [<int>, ...] // omit when you cannot point at a sentence}

Table 11: Self-repair check prompt.

Self-repair check prompt (system message)
You track whether previously-flagged issues in a document discussion have since been resolved within the conversation itself. You are given the document as numbered sentences [S#] (the ground truth), the list of open issues [C#], and the recent transcript. For each open issue, first find the [S#] sentence stating the correct fact, then apply this rule. An issue is resolved only when the recent transcript has actually stated the correct fact: the correct value replaces the wrong one (for a wrong claim) or the missing value is supplied (for a missing fact). These do not resolve an issue: repeating the same wrong value, merely mentioning the topic again, or a participant saying they cannot recall / do not know. Consider only the recent transcript below, not older context. Report only resolved ids; if none, return an empty list. Never invent an id that is not in the list. Cite the grounding sentence id(s) in 'by_sids' — at most 3, and only ids that actually state the fact. Omit the field when the fact comes from a table or you cannot point at a sentence.
Self-repair check prompt (user message)
Open issues [C{cid}] {claim} [C{cid}] {claim} ... (or "(none)") Recent transcript {recent turns, same window as the monitor} Output (JSON only) {"resolved": [{"cid": <int from the list above>, "by": "<speaker id who resolved it>"}, "by_sids": [<int [S#] grounding the correct fact>, ...]}

Table 12: Grounded generation prompt.

Grounded generation prompt (system message)
You contribute to a discussion about a research document, grounded only in the numbered sentences [S#] of the section text. The user message names the gate’s intent; follow that intent’s block. Intents correct — a participant’s claim may contradict the document. Verify it against the sentences; if wrong, state the correction in one or two short sentences. If the sentences neither support nor contradict the claim, treat it as not_in_paper. missing — the participants need a specific fact the document states but no one has said. Answer completely: every requested value or part; length is not limited. suggest — the discussion has stalled on the point below. Surface the relevant fact or propose one concrete next step. not_in_paper — a participant asserted the claim as this document’s content, but the sentences do not state it. Say so in one short sentence so the group stops attributing it to the document. Do not guess what the document says instead. Example form: "This paper doesn't mention <the point>." Grounding correct, missing, suggest — cite the sentences you used by their [S#] ids in 'quote_sids': ids only, no copied text. Order 'quote_sids' most-substantive first: the id whose sentence states the fact most fully leads, not one that only mentions it in passing. If only some of the requested parts are grounded, state those and cite them — do not abstain. {answer_bias}Abstain only if none of the needed facts appear in the sentences. No adjacent / speculative / future-work material. not_in_paper — 'quote_sids' is [] (nothing to cite) and you do not abstain: the absence of the fact is the reply. Exception: if the sentences directly contradict or answer the claim after all, follow the correct rules instead (state the correction and cite it). Spoken text — every intent Never write [S#] ids, bracketed citation numbers, or reference markers in 'text'; it is read aloud and must contain none of them. Topic Set 'topic' to a short noun phrase (2-4 words) naming what your answer is about — the specific subject it addresses (the metric, component, quantity, or concept), grounded in the sentences. It orients a listener who did not ask, so it must describe the answer’s subject, not a document search term. Use lowercase natural phrasing (e.g. "the training dataset size", "the comparison baseline", "the participant count") and no [S#] ids or citation markers.
Grounded generation prompt (user message)
Gate intent {intent} (correct missing suggest not_in_paper) Point under discussion {claim text from the ledger, e.g. "how many participants the study recruited"} Output (JSON only) {"action": "speak abstain", "text": "{text spec for the intent}", "topic": "<2-4 word noun phrase naming what the answer is about>", "source": "{loc}", "quote_sids": [<int>, ...]}

Table 13: Direct answering prompt.

Direct answering prompt (system message) You answer a participant’s direct question about the research document above, using ONLY its content. The document is given as numbered sentences [S0], [S1], ... Reply with JSON only. The recent conversation is provided for CONTEXT ONLY — use it to understand what the question refers to (e.g. resolve ‘this point’ or ‘that’ to the topic under discussion). Never treat the conversation as a source of facts; answers come only from the document. STEP 1 — decide whether to answer at all. Set ‘action’ to ‘skip’ (and leave ‘text’ empty and ‘quote_sids’ empty) when the input is any of: - an INCOMPLETE fragment: it ends mid-sentence, trails off with a dash, or has no closing punctuation (e.g. ‘what models have they—’); - NOT addressed to you: participants talking to each other, an aside, a correction, or thinking aloud (e.g. ‘never mind, scroll up’). ‘skip’ means stay silent. Do NOT explain your decision and do NOT say it is not covered in the document for these — just ‘skip’. NEVER ‘skip’ a complete question or request directed at you — not even one the document cannot answer. Those take ‘answer’ or ‘abstain’ (STEP 2). Example: ‘can you answer my previous question’ is a complete request — find that question in the Recent conversation and answer or abstain on IT; skipping it leaves the asker with nothing. STEP 2 — only for a COMPLETE, well-formed question directed at you: set ‘action’ to ‘answer’, write your answer in your own words in ‘text’, and put in ‘quote_sids’ the id numbers of the document sentences your answer is based on — as a LIST of integers. When the question refers to earlier talk (‘my previous question’, ‘that point’, ‘the thing we discussed’), resolve the reference from the Recent conversation FIRST, then answer the resolved question from the document. Cite the sentences you actually used; do NOT copy their text anywhere. Order ‘quote_sids’ most-substantive first: the id whose sentence states the fact most fully leads, not one that only mentions it in passing. The server pulls the exact wording from those ids, so an answer paraphrased in ‘text’ is fine as long as ‘quote_sids’ points at the real sentences behind it. If a complete question’s answer is genuinely absent from the document, set ‘action’ to ‘abstain’ with empty ‘quote_sids’ — do not guess. Keep ‘text’ concise; it is read aloud. Never write [S#] ids, bracketed citation numbers, or reference markers in ‘text’ — it is read aloud.
Direct answering prompt (user message) Recent conversation {last 8 turns} (block omitted when there is no history) Question {utterance, wake phrase removed} Answer Answer concisely (<=3 sentences) grounded in the document, and cite the sentence ids you used in ‘quote_sids’. If unsupported, use action ‘abstain’ with text exactly: ‘That isn’t covered in the document.’ and ‘quote_sids: []’. Output (JSON only) {"action": "answer abstain skip", "text": "<answer read aloud>", "quote_sids": [<int>, ...]}

Table 14: Judge prompt for the accuracy of direct answers (Claude Sonnet 5). One call judged one batch file of 130 items; the prompt is the same for the three datasets.

Judge prompt (direct-answer accuracy)	
You are auditing an evaluation harness. Judge whether answers a QA system produced convey what the reference answers say. You are NOT judging style, length, or wording. Read the batch file named below. It is JSON with a `records` array. Each record has: qid - the identifier to report back dataset - which release the question came from gold_origin - how this release produced its reference; read it, it tells you what a DEFECTIVE reference looks like here question - what was asked gold - the primary reference answer other_refs - further reference answers, any of which is equally valid (may be empty) answer_type - the shape of the reference: extractive / free_form / yes_no / multi_answer / program / span our_answer - what the system answered, the text under judgment f1 - token-overlap score, present when it was computed. It is SYSTEMATICALLY LOW for correct answers, because the system answers in full sentences against terse references. Never let a low f1 push you toward WRONG. Assign EVERY record exactly one label: CORRECT The answer conveys the fact that `gold` states, or that ANY entry of `other_refs` states. Paraphrase counts. So do: an abbreviation expanded or contracted (gold "SVM" vs "Support Vector Machine"), a more specific variant (gold "coronavirus" vs "betacoronavirus"), the same value in another notation (gold "10000" vs "10K"; gold "less than 50%" vs "44.55%"), singular/plural, unicode vs ascii punctuation, and the right fact wrapped in extra context. For answer_type yes_no, judge the STANCE - an answer that never says "yes" but states the position in prose is CORRECT when the position matches. For answer_type program, `gold` is an arithmetic program whose numbers are OPERANDS; the answer must state the EXECUTED value, which `other_refs` carries. Restating an operand is not computing the answer. PARTIAL On topic and carrying part of the reference fact, but a material element is missing, or it answers a neighbouring question. Use this when the reference packs several facts and the answer delivers only some. Also use it when the answer states the right direction but refuses to commit ("the document does not explicitly state, but ..."). WRONG A different fact, or the opposite stance. A reader would call it incorrect. GOLD_BROKEN The reference cannot serve as a reference at all - not because the answer is bad, but because NO correct answer could ever have matched it. Judge this by the test: could ANY correct answer have matched this reference? Cases the releases actually produce: the reference is only a marker no natural-language answer reproduces (BIBREF19, FIGREF10), it is cut mid-sentence so the fact is gone, it is corrupted beyond recognition, it is a stray character, or it answers a different question than the one asked - including an answer_type of yes_no attached to a question that is not a yes/no question. If a reference is truncated but the answerable fact survives before the cut, it is USABLE: judge normally. Tie-breaks: torn between CORRECT and PARTIAL, choose PARTIAL. Torn between WRONG and GOLD_BROKEN, apply the test above. When other_refs disagree with gold, matching ANY of them is CORRECT. OUTPUT FILE (required). Write the TSV path named in the batch file, one line per record, no header, three tab-separated columns, in the SAME ORDER as the records: qid<TAB>LABEL<TAB>reason of at most 12 words Line count must equal the record count. Verify it before you finish. RETURN: reply with only the label counts and the TSV path. Do NOT paste the judgments - the file is the deliverable.	

Table 15: The gold_origin string carried by every item record, per dataset.

Dataset	gold_origin
QASPER	An NLP researcher answered from the full paper with a terse span, a free-form phrase, or yes/no. Golds keep LaTeX artifacts of the source: citation placeholders (BIBREF19), figure refs (FIGREF10), inline-math markers (INLINEFORM2). Up to three annotators answered, so other_refs may hold alternative valid answers.
COVID-QA	An annotator marked a span inside a PDF-extracted COVID-19 research paper and then wrote a question for it. Known defects: text cut mid-sentence, PDF corruption (degree sign as 'u', prime as '='), single-character fragments. There is only one reference per question.
CHI-180	A span or short answer marked by an annotator in the source document.

Table 16: Direct-answer accuracy judge: the rubric. The file-handling instructions of the original are replaced by the first and last paragraphs.

Direct-answer accuracy judge	
You are auditing an evaluation harness. Judge whether answers a QA system produced convey what the reference answers say. You are NOT judging style, length, or wording.	
Each record to judge has these fields:	
qid - the identifier to report back dataset - which release the question came from gold_origin - how this release produced its reference; read it, it tells you what a DEFECTIVE reference looks like here question - what was asked gold - the primary reference answer other_refs - further reference answers, any of which is equally valid (may be empty) answer_type - the shape of the reference: extractive / free_form / yes_no / multi_answer / program / span our_answer - what the system answered, the text under judgment f1 - token-overlap score, present when it was computed. It is SYSTEMATICALLY LOW for correct answers, because the system answers in full sentences against terse references. Never let a low f1 push you toward WRONG.	
Assign EVERY record exactly one label:	
CORRECT	The answer conveys the fact that 'gold' states, or that ANY entry of 'other_refs' states. Paraphrase counts. So do: an abbreviation expanded or contracted (gold "SVM" vs "Support Vector Machine"), a more specific variant (gold "coronavirus" vs "betacoronavirus"), the same value in another notation (gold "10000" vs "10K"; gold "less than 50%" vs "44.55%"), singular/plural, unicode vs ascii punctuation, and the right fact wrapped in extra context. For answer_type yes_no, judge the STANCE — an answer that never says "yes" but states the position in prose is CORRECT when the position matches. For answer_type program, 'gold' is an arithmetic program whose numbers are OPERANDS; the answer must state the EXECUTED value, which 'other_refs' carries. Restating an operand is not computing the answer.
PARTIAL	On topic and carrying part of the reference fact, but a material element is missing, or it answers a neighbouring question. Use this when the reference packs several facts and the answer delivers only some. Also use it when the answer states the right direction but refuses to commit ("the document does not explicitly state, but ...").
WRONG	A different fact, or the opposite stance. A reader would call it incorrect.
GOLD_BROKEN	The reference cannot serve as a reference at all — not because the answer is bad, but because NO correct answer could ever have matched it. Judge this by the test: could ANY correct answer have matched this reference? Cases the releases actually produce: the reference is only a marker no natural-language answer reproduces (BIBREF19, FIGREF10), it is cut mid-sentence so the fact is gone, it is corrupted beyond recognition, it is a stray character, or it answers a different question than the one asked — including an answer_type of yes_no attached to a question that is not a yes/no question. If a reference is truncated but the answerable fact survives before the cut, it is USABLE: judge normally.
Tie-breaks: torn between CORRECT and PARTIAL, choose PARTIAL. Torn between WRONG and GOLD_BROKEN, apply the test above. When other_refs disagree with gold, matching ANY of them is CORRECT.	
For every record return one line: qid, LABEL, and a reason of at most 12 words, in the same order as the records.	
qid<TAB>LABEL<TAB>reason of at most 12 words	
Line count must equal the record count. Verify it before you finish.	

Table 17: User-study intervention judge: instruction. File names of the original are replaced by descriptions of the inputs and outputs.

User-study intervention judge (instruction)
You are judging proactive-agent interventions from a Zoom user-study session. Read first, fully: the judging rubric (the label vocabulary and rules; follow it exactly). Then read: one card per intervention for this session (each card carries the intervention text, its grounding page and quote, the part and its wait setting, and a transcript excerpt of the 10 turns before and 2 turns after the intervention); the session's event sheet (the planted cards: the wrong sentence the lead participant was to plant, the true value, the evidence page, and the expected agent behavior); the full session transcript when a card's excerpt is not enough context; and the paper text with page markers, to verify every factual claim in the agent's utterance. For every intervention produce: the turn it reacted to (problem point), and the labels timing, target match, fact label, grounding, necessity, and reaction, with a reason of at most 20 words that cites the paper page. Rules of thumb: the lead participant paraphrases planted cards heavily, so match by meaning, not words; the system's own record of the originating turn is a hint, not truth; judge the fact label against the paper text, citing the page; every intervention must get a full row with no empty label. Be strict and consistent.

Table 18: User-study intervention judge: rubric (label vocabulary and labeling rules).

User-study judging rubric
Problem point p_ref: the utterance the intervention reacted to, found within the 10 turns before it. The system's own record of the originating turn is kept when it is right and corrected otherwise. p_ref_kind: planted_wrong (a sheet card spoken by the lead participant) / natural_wrong (a wrong statement not on the sheet, by any speaker) / natural_gap (not wrong, but a question, an admission of not knowing, or a missing fact; a card spoken as a question counts here) / none (no corresponding utterance within the 10 turns). Labels per intervention timing: delay = decision turn minus problem-utterance turn, counted on the transcript. hard (delay of 2 or less) / soft (3 to 5) / late (more than 5) / early (negative) / unprompted (p_ref_kind is none). Fixed thresholds, independent of the part's wait setting. target_match: yes / partial (same paragraph or adjacent topic) / no (a different matter). fact_label: is the fact stated by the intervention consistent with the paper, regardless of style or length. CORRECT / PARTIAL (the right direction, but a number or condition is partly wrong or missing) / WRONG (a different fact) / UNVERIFIABLE (cannot be checked against the paper). When in doubt, PARTIAL. grounding_ok: do the attached page and quote point to the actual evidence. yes / wrong_page / none / n/a (parts with the source display off). necessity: needed (the fact had to be corrected or supplied at that moment) / redundant (a participant had already corrected it within the window, an earlier intervention had already stated the same fact, or nothing was wrong) / harmful (a wrong correction, or an intervention that breaks the conversation). Review stage only: preempted (needed when decided, but a person stated the fact before playback). reaction: the participants' response in the 2 turns after the intervention. accepted / ignored / disputed / n/a (the intervention was not spoken). judge_reason: at most 20 words, citing the paper page. Labeling rules A statement that substitutes a figure from outside the paper (for example a manufacturer's advertised value) for the value the paper reports is PARTIAL, even when the outside figure is widely cited. A statement that reports a quantity of a different kind than the one the participant asked about, and says the paper does not give the requested comparison, is PARTIAL rather than WRONG when the statement itself is true; its necessity is needed. WRONG is reserved for a fact that differs from the paper. A planted card that the lead participant spoke as a question is natural_gap; the card id is kept. redundant applies only when the intervention repeats the same fact as an earlier intervention; new numbers, conditions, or pages make it needed. The two statements are compared side by side. Unspoken interventions are judged on timing, target, and fact at the time of the decision; reaction is n/a. In hand-raise parts the delay between the decision and playback is by design and is not counted in timing; timing is taken from when the system decided. A second intervention on the same problem point is normally redundant. An intervention that arrives after the session's last turn, so that no reaction can be observed, is excluded from the counts.

Table 19: Problem-utterance labelling: instruction. File names of the original are replaced by descriptions of the inputs and outputs.

Problem-utterance labelling (instruction)
You are labelling every problematic utterance (wrong statement, missing fact or question, planted card) in one Zoom user-study transcript. Read first: the judging rubric, especially the problem-utterance column specification and the rule that a planted card spoken as a question is natural_gap with its card id kept. Then read the whole transcript turn by turn (the five study parts only; skip the introduction and the pre- and post-session segments), the session's event sheet (the cards: the wrong sentence to plant, the true value, the evidence page; the lead participant paraphrases heavily, so match by meaning), and the paper text with page markers. Consistency inputs that must be included and agree: the problem points already assigned to this session's interventions, and the turns the lead participant's button presses referred to. Output one row per problem (a turn can have several rows) with: turn, speaker role, kind (planted_wrong / natural_wrong / natural_gap / self_corrected), card id if any, the problematic phrase verbatim (at most 25 words), what is wrong or missing (at most 20 words), the paper's actual content with its page (at most 25 words), the interventions that reacted to it, who eventually corrected it (agent / participant / none), and the button press that pointed at it, if any. Include every planted card the lead participant actually spoke, even when nobody reacted.

Table 20: Proactive-event judge: rubric (system message, abridged).

Proactive-event judge (system message)
You evaluate how a proactive research advisor handled ONE moment in a conversation between two researchers (S1, S2) who are discussing a published paper from memory. You are given the FULL TEXT of the paper, an EVALUATION FOCUS, and a CONVERSATION SNAPSHOT. In the snapshot, lines starting with ">>> ASSISTANT" are the advisor's spoken interventions, placed where they arrived; the one marked [EVALUATED] is the intervention attributed to this focus. If no [EVALUATED] line appears, the advisor stayed silent for this focus. The focus is either a question a researcher asked and could not resolve (with the paper's answer, or a note that the paper does not report it) or a claim a researcher asserted (with what the paper actually supports). Return ONLY this JSON: <pre>{ "issue_status": "unresolved" "resolved_by_participants" "no_issue", "decision_score": 1-5, "answer_status": "correct" "partially_correct" "off_target" "incorrect" "fabricated" null, ... }</pre> issue_status — the state of the focus issue at the moment just before the [EVALUATED] intervention (or, if the advisor stayed silent, at the end of the snapshot): "unresolved": the question was still unanswered / the false claim was still standing; "resolved_by_participants": a participant had already given the correct answer or corrected the claim; "no_issue": no real issue arose (e.g. the asker answered it themselves at once, or the claim was never adopted). decision_score — was speaking (or staying silent) the right call given issue_status? 5 = clearly right, 1 = clearly wrong. Speaking on an unresolved issue is right; staying silent on an unresolved issue is wrong; staying silent when the participants had resolved it is right; a brief confirmation after their own correction can still be acceptable (3-4) if it adds something. answer_status — only for the [EVALUATED] intervention (null if silent), judged against the PAPER itself, not the gold (the gold may be incomplete or wrong): "correct": its substantive claims are supported by the paper and it answers the question / corrects the claim (an alternative phrasing or another correct value from the paper counts); "partially_correct": on target, but omits or garbles an essential component; "off_target": it does not attempt to answer this question / correct this claim — it is about something else; "incorrect": it asserts something the paper contradicts, or a wrong value / entity; "fabricated": it asserts specifics the paper does not contain at all. For an UNANSWERABLE focus (the paper does not report the asked detail): "correct" if it says the paper does not report it, or gives only related information that is in the paper without inventing the missing detail; "fabricated" / "incorrect" if it asserts the missing detail as if from the paper. Ignore page numbers, section numbers or "search for X" pointers in the intervention: the paper text has no page markers, so they are neither verifiable nor substantive. Judge only the substantive content.