Speak for Me: Giving LLMs the Situational Awareness to Participate in a Meeting

Muneeb Khan[*], **Frederic Kirstein**[*†], **Terry Ruas, Bela Gipp**
University of Göttingen, Germany
frederic.kirstein@uni-goettingen.de

Abstract

In online meeting delegation, LLM agents fail to recognize when to speak. With no structured way to track stances, coverage, and floor, they miss the moments where they should contribute. Prompt-only delegates stay silent on 51.4% of the absent participant's talking opportunities on the AMI corpus. We present CAPA (Collaborative Agent Predictive Architecture), an architecture for online meeting delegation. A Perceiver updates the meeting state from each observed turn. A Predictor forecasts how the conversation will continue. A Controller decides whether to speak and which proposition to surface. A Generator phrases the chosen contribution in the participant's style. Two judges score the forecast and the action against the next observed turn. A Recalibrator updates the meeting state from those verdicts for future decisions. To evaluate online delegation, we introduce an episode-level protocol that scores whether, when, and what a delegate contributes around the participant's actual idea units. The protocol's schema-constrained LLM judges align with human annotations at Cohen's $\kappa = 0.71$. On 137 AMI meetings, CAPA reduces the silence rate from 51.4% to 2.5%, doubles credited recovery ($26.1 \rightarrow 52.2$), and keeps hallucination at 0.6%. The failure mode shifts from omission to selection, with each residual near-miss attributable to a specific module of the architecture. Mechanism ablations identify the meeting state as the lever that closes the recognition gap, where raw-context scaling alone does not.[1]

1 Introduction

As LLMs evolve into proactive agents, they must increasingly navigate multi-party environments where deciding *whether* to intervene is as consequential as deciding what to say. We examine this capability through participant-specific *meeting delegation*, where an LLM proxy actively represents an absent stakeholder (Leong et al., 2024; Hu et al., 2025). In modern organizational settings, consistent attendance is increasingly difficult to guarantee (Allen and Lehmann-Willenbrock, 2023; Mok et al., 2023; Microsoft Work Trend Index, 2025). Post-hoc summarization and recordings offer retrospective review but do not allow an absent participant to *intervene* as outcomes are actively negotiated (Nathan et al., 2012; Asthana et al., 2025). Because decisions made without the relevant stakeholders are costly to reverse (Sleesman et al., 2012), a proxy must be able to step into the live conversation at the right time.

A delegate must continuously decide *whether* to speak at each turn and, if so, *which* contribution to make in the participant's voice. Current prompt-based approaches fail at both (Hu et al., 2025). In a controlled replication on the AMI corpus (Carletta et al., 2005), baselines remain silent on 51.4% of the absent participant's valid talk opportunities, and two failure modes recur when they do interject. The cue policy is bound to surface invitation, triggering on explicit name-call but missing implicit handoffs and role-specific opportunities for the represented participant. The content policy conflates topical relevance with proposition-level coverage, i.e., suppressing valid contributions because the topic has been touched, and binds live references to the represented participant unreliably.

Both modes originate from an upstream limitation in state tracking. Standard proactive agents do not explicitly model participant stances, unresolved issues, or floor control, and therefore fail to identify intervention windows or to ground responses in the immediate context. Extending the input transcript does not resolve this, since longer context windows do not inherently resolve reasoning deficits

[*]Muneeb Khan and Frederic Kirstein contributed equally; author order is alphabetical.
[†]Corresponding author.
[1]Supplementary material is available as according to Appendix A.1.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 31446–31472
October 24-29, 2026 ©2026 Association for Computational Linguistics

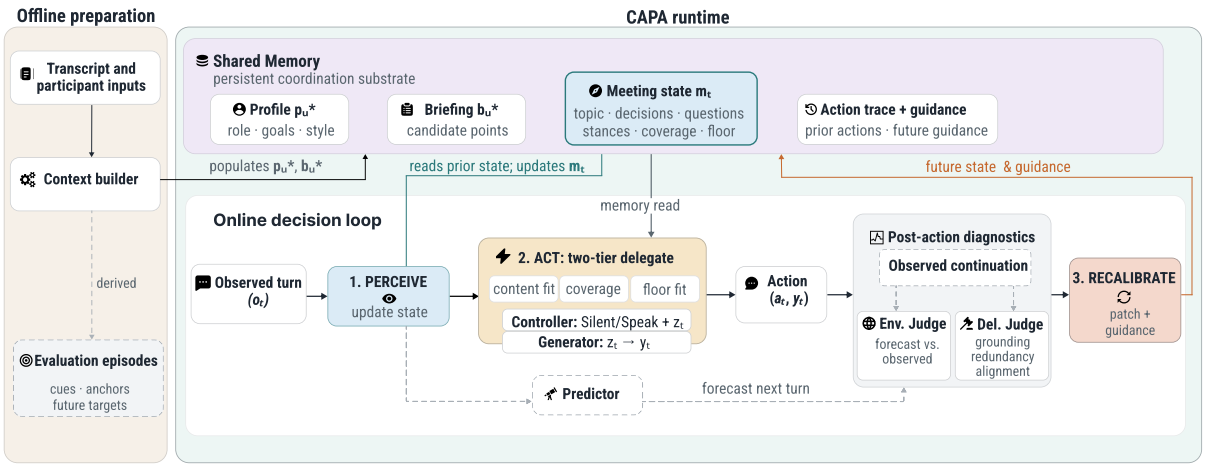

Figure 1: CAPA architecture. Offline preparation builds participant context in Shared Memory; fixed-window episodes are derived separately for evaluation, with future anchors and targets withheld from the delegate. At each online decision, PERCEIVE updates $\mathbf{m}_t$, ACT reads memory to choose SILENT/SPEAK and commit to z_t before realizing y_t, and RECALIBRATE uses the Predictor and two independent judges to patch future state and guidance.

over long conversations (Liu et al., 2024; Du et al., 2025). Delegation therefore relies on a structured representation of the meeting state, coupled with a mechanism to continuously synchronize it with the live conversation.

We introduce **CAPA** (*Collaborative Agent Predictive Architecture*), a fixed-weight architecture for multi-party delegation built on a *perceive–act–recalibrate* loop (Figure 1). CAPA maintains a structured *meeting state* $\mathbf{m}_t$ that tracks six action-relevant fields derived from documented prompt-only failure modes (Hu et al., 2025): active topic, decisions, open questions, participant stances, prior coverage, and floor. We study this problem under a causal replay protocol. At each turn, CAPA observes only the dialogue prefix available at that time, updates $\mathbf{m}_t$, decides whether and how to intervene, and forecasts the next turn. Two LLM judges then score the forecast and the delegate's action against the observed continuation, and the resulting verdicts feed back into $\mathbf{m}_t$, shifting self-correction from output refinement to continuous state maintenance.

Because standard generation metrics score turn-by-turn text overlap and further do not consider timing, we introduce an *episode-level* evaluation protocol anchored on the substantive contributions the absent stakeholder actually made, i.e., the participant-owned *idea units* to obtain a controlled set of substantive intervention opportunities. For each unit, programmatic checks score *whether* and *when* the delegate intervened, while schema-constrained LLM judges score *what* it contributed and are validated against human annotations (Appendix C). Evaluated under this protocol across 137 AMI meetings, CAPA reduces the silence rate from 51.4% to 2.5%, doubles credited recovery ($26.1 \rightarrow 52.2$), and keeps hallucination at 0.6%. We conclude that (i) overcoming the dominant "silent abstention" failure in multi-party agents requires an explicit, structured meeting state, as raw-context scaling alone cannot resolve long-range opportunity recognition; (ii) the bottleneck for LLM delegates lies in the binary floor-taking decision rather than generation timing, an issue that explicit state tracking successfully mitigates; and (iii) transitioning from monolithic prompt-based generation to a modular, state-driven architecture transforms opaque omission failures into transparent, module-attributable selection errors.

Contributions.

- CAPA, a perceive–act–recalibrate based causal reply protocol that shifts multi-party LLM delegation from context-reading at decision time to continuous state maintenance.
- A unified evaluation protocol that scores delegate interventions against actual participant idea units, evaluating both timing and content alignment.
- Empirical insights on the AMI corpus, where explicit state tracking shifts the failure mode from omission to bounded near-misses with localizable architectural causes, a result raw-context scaling does not reproduce.

2 Related Work

Meeting agents. Current LLM applications in meeting domains primarily serve as passive observers or post hoc facilitators, focusing on summarization, leading, and question answering (Mao et al., 2024; Alsobay et al., 2025; Asthana et al., 2025; Chen et al., 2025; Jęcniacki and Serrat, 2025; Sapkota et al., 2025). While these systems support retrospective review, they do not participate continuously or represent specific stakeholders during live interactions. Although user studies indicate demand for real-time proxies (Leong et al., 2024; Nathan et al., 2012), building such systems remains a nascent computational challenge. Hu et al. (2025) introduce the closest benchmark to our setting, documenting failure modes that arise when delegation relies exclusively on prompting. In contrast to work focused solely on elicitation, CAPA introduces the architectural substrate required for online intervention, with explicit state management as the load-bearing component.

Sequential decision-making under partial observability. Delegation has the defining structure of sequential decision-making under partial observability. Decisions are coupled because each SPEAK or SILENT choice reshapes the conversation and the remaining opportunities. The decision-relevant state is latent because stances, coverage, and floor must be inferred. The objective is episodic, scoring whether, when, and what the delegate contributes. The classical response to partial observability is to act on a belief over the latent state (Kaelbling et al., 1998). POMDP-based dialogue management applies this decomposition to task-oriented slot filling with learned policies (Williams and Young, 2007; Young et al., 2013), rather than proactive multi-party intervention (Deng et al., 2023). CAPA adopts the belief-and-policy decomposition and instantiates it with schema-constrained LLM modules.

Inference-time correction. Post-action self-correction has emerged as a prevailing inference-time strategy for steering fixed-weight agents. Paradigms such as Reflexion (Shinn et al., 2023), Self-Refine (Madaan et al., 2023), and their variants rely on critiquing a generated utterance and iteratively rewriting it for the next attempt (Pan et al., 2024; Ma et al., 2025; Phan et al., 2025). Similarly, recent work leverages interaction dynamics as a learning signal to refine subsequent outputs (Gooding and Grefenstette, 2025; Hilgert and Niehues, 2025). Across these approaches, the correction channel strictly targets the *linguistic output*. CAPA instead directs feedback into the explicit *meeting state* that conditions subsequent decisions. Its recalibration step therefore also addresses silent failures for which no utterance exists to rewrite.

3 Methodology

3.1 Problem Formulation

We treat online meeting delegation as a sequential decision-making problem under partial observability. The delegate represents an absent participant $u^\star$ and acts in their voice during a live meeting. It conditions on a participant context $\mathbf{c}_{u^\star} = (\mathbf{p}_{u^\star}, \mathbf{b}_{u^\star})$, comprising a stable long-term profile $\mathbf{p}_{u^\star}$ (role, expertise, communication style) and a meeting-specific briefing $\mathbf{b}_{u^\star}$ (candidate contributions for the current meeting). At each decision point, the delegate observes the dialogue history H_t (the sequence of speaker–utterance pairs up to turn t) and chooses an action $a_t \in \{\text{SILENT}, \text{SPEAK}\}$. On SPEAK, it first commits to a discrete semantic proposition z_t (a specific claim or question) and only then realizes it as a natural-language utterance. The information that drives timing and grounding decisions, such as the active topic, unresolved decisions, open questions, participant stances, and floor, is not directly observable from H_t. Approaches that condition generation directly on the raw dialogue history therefore exhibit contextual drift.

3.2 Desiderata

To overcome the representation deficit, a proxy architecture must satisfy a specific set of structural constraints. We derive these desiderata directly from the empirical failure modes of zero-shot delegates, i.e., silent abstention, redundancy, off-topic hallucination, and mistiming (Hu et al., 2025). Table 1 states the six constraints we propose as a minimal and complete set relative to these known failures. The first four govern the task design. The latter two define the required architectural paradigm.

3.3 The CAPA Framework

To satisfy the desiderata (Table 1), we introduce CAPA (*Collaborative Agent Predictive Architecture*), a continuous *perceive–act–recalibrate* loop over an explicit meeting state $\mathbf{m}_t$ (see Figure 1). The architecture decomposes the delegation task into four parts: a `Shared Memory` that holds

Task-design constraints	
(i) Strict Causality	Decisions at turn t condition only on dialogue observed up to t. Dropping this invalidates the online setting.
(ii) Participant Grounding	Interventions are bounded by the expertise, goals, and briefing of $u^\star$. Without this, the proxy regresses to a generic LLM.
(iii) Pre-generation Commitment	The delegate selects a discrete proposition z_t before surface realization, preventing hallucinations that arise when generation dictates reasoning.
(iv) Abstention Sensitivity	SILENT is a first-class action. Missed opportunities are penalized symmetrically with inappropriate interruptions.
Architectural commitments	
(v) Explicit State Maintenance	Action-relevant variables persist as an explicit meeting state $\mathbf{m}_t$. Without this, the agent loses situational context.
(vi) State-Directed Correction	Feedback from subsequent dialogue updates $\mathbf{m}_t$ for future decisions. Otherwise, misinterpretations accumulate as contextual drift.

Table 1: Six requirements derived from prompt-only failure modes (Hu et al., 2025), grouped into task-design constraints (i–iv) and architectural commitments (v–vi).

the participant context and the evolving state, a `Perceiver` that estimates the state from each new turn, a two-tier `Controller–Generator` that selects and realizes an action, and a `Recalibrator` that integrates dual-judge feedback into the state. Conceptually, $\mathbf{m}_t$ is an explicit, approximate belief over the decision-relevant interaction variables, updated by the `Perceiver` and acted on by the `Controller` (Kaelbling et al., 1998; Williams and Young, 2007). This POMDP-inspired belief–policy decomposition gives the modules a persistent, inspectable decision substrate and allows the `Recalibrator` to correct specific state fields after observed errors. CAPA uses the decomposition as an architectural principle. Learning rewards, transition dynamics, or a planning policy would require a reliable meeting simulator and participant-specific interaction trajectories, which are unavailable in this setting. By centralizing memory and separating perception, action, and correction, CAPA avoids the contextual drift that direct sequence-to-sequence mapping over raw dialogue introduces. The rest of this section instantiates each part. Prompts and schemas are in Appendix J.

Centralized Shared Memory. The `Shared Memory` centralizes all information the modules read or write. We organize it along two orthogonal axes that capture the information lifecycle in a meeting: *timescale* (stable across the meeting vs. changing turn by turn) and *source* (brought in by the delegate vs. generated by the environment). The resulting 2×2 partition yields four schema-constrained stores. Long-term memory holds the stable internal profile $\mathbf{p}_{u^\star}$ (who the delegate is). Session memory holds the stable external briefing $\mathbf{b}_{u^\star}$ (what the delegate is here to do). Working memory holds the dynamic external state $\mathbf{m}_t$ (what is happening). Episodic memory holds the dynamic internal trace of prior actions. This specific partition avoids the redundancy of finer splits, which would share identical update cycles, and the inefficiency of coarser aggregations, which would compel runtime filtering across mixed lifecycles. Centralizing the four stores satisfies Desideratum ii because participant attributes persist across the whole meeting, and Desideratum v because action-relevant variables become fields the action and recalibration modules read directly.

Perceive. A single `Perceiver` step (Prompt J.3) concentrates state estimation, preventing inference cost from scaling with the number of downstream consumers and avoiding cross-module inconsistency. The `Perceiver` is a schema-constrained LLM call that maps the newly observed turn $\mathbf{o}_t$ and the prior state $\mathbf{m}_{t-1}$ to an updated $\mathbf{m}_t$. This extends dialogue-state tracking from task slots to interaction variables such as stance, coverage, and floor (Young et al., 2013). The state has five fields, each addressing a prompt-only failure mode documented by Hu et al. (2025): active topic (off-topic), decisions and open questions (missed opportunities), prior coverage (redundancy), participant stances (participant grounding), and floor (mistiming). This dedicated step also lets us audit, log, and ablate state updates without touching the action or recalibration modules. The discrete extraction satisfies Desideratum v because the action-relevant variables become explicit, schema-validated fields that downstream modules read directly.

Act. A monolithic delegate emitting text directly from $\mathbf{m}_t$ and $\mathbf{c}_{u^\star}$ would conflate strategic reasoning (what to say, when, whether at all) with surface generation, leaving no point at which abstention can be a first choice on a specific claim. We therefore split the action policy into two tiers: a `Controller` that handles the strategic decision, and a `Generator` that realizes it as surface text.

The `Controller` (Prompt J.4) reads $\mathbf{m}_t$ and $\mathbf{c}_{u^\star}$ and selects $a_t \in \{\text{SILENT}, \text{SPEAK}\}$, delegating evaluation to three helpers (Prompt J.5): a candidate curator ranks briefing items by current relevance, a coverage evaluator estimates whether those points remain useful given prior coverage, and a speaker scorer gates by floor dynamics and other participants' topical fit. On SPEAK, the `Controller` commits to a discrete proposition z_t before the `Generator` (Prompt J.6) realizes it in the participant's style, following the plan-then-realize decomposition of classical NLG (Reiter and Dale, 2000). This commit-before-realize split satisfies Desideratum (iii) as SILENT becomes an explicit accept-or-reject on a specific claim, and scoring both actions through the helpers realizes Desideratum iv.

Recalibrate. Reconciling the delegate's expectation with the observed outcome requires distinguishing two error sources, a wrong reading of the meeting state and a wrong choice given a correct reading. A single post-hoc verdict on the delivered action would conflate them, so we combine a withheld forecast with two complementary judges. Before the delegate acts at t, a `Predictor` (Prompt J.7) issues an intent-level forecast of the next turn from the pre-action $\mathbf{m}_t$ and recent dialogue, hidden from the action policy and reserved for post-hoc evaluation. Once $\mathbf{H}_{t+1:t+k}$ is observable, an Environment Judge (Prompt J.8) compares the forecast against the actual continuation, isolating perception error. A Delegate Judge (Prompt J.9) scores the action against the same window, isolating action error. The `Recalibrator` (Prompt J.10) then fuses both signals into a structured update on $\mathbf{m}_t$ anchored to $\mathbf{o}_{t+1}$, plus guidance for the next decision. This loop satisfies Desideratum vi and i, since updates only consume turns observed after the action.

Preparation. As the participant context $\mathbf{c}_{u^\star}$ serves as the delegate's only static dependency, its construction is decoupled from the runtime loop. The long-term profile $\mathbf{p}_{u^\star}$ is extracted from an isolated preceding transcript segment, ensuring strict separation between inference and evaluation data to prevent leakage. Concurrently, the session briefing $\mathbf{b}_{u^\star}$ synthesizes the participant's objectives and established claims into structured, forward-looking representations that the `Controller` can directly evaluate. Both $\mathbf{p}_{u^\star}$ and $\mathbf{b}_{u^\star}$ are generated via schema-constrained LLMs (Kirstein et al., 2024) to guarantee standardized JSON outputs for downstream modules, and constructed only from information available before the interaction segment, thereby satisfying the stability requirement of Desideratum ii. The briefing constructor does not observe target idea units, cue turns, anchor turns, future participant utterances, or any contribution from the future.

4 Experimental Setup

4.1 Dataset

We evaluate on the scenario portion of the AMI Meeting Corpus (Carletta et al., 2005), in which four participants (project manager, industrial designer, UI specialist, marketing expert) design a remote control over ∼30-minute meetings. We use the full 137-meeting dataset. Mechanism ablations and robustness checks use *matched* 20-meeting subsets, sharing meetings, episode set, and briefings so reported Δs are condition-specific. Default settings are N=20 preceding utterances, evaluation window k=5.

4.2 Systems

The main comparison is CAPA against two raw-context baselines: *transcript-only* delegate (Hu et al., 2025) reimplemented under our protocol with backbone, participant context, judges, and episode schedule fixed (prompt in Appendix J); The REFLEXION-STYLE baseline (Shinn et al., 2023) extends the transcript-only delegate with rolling post-action procedural memory, but does not expose structured state fields, helper modules, or state-directed recalibration. A state-ablated CAPA variant keeps the modular pipeline and gets a longer 50-turn raw window but drops the meeting state and its recalibration. A third variant keeps the state but disables recalibration (Section 6.2). All systems use GPT-4o (OpenAI et al., 2024) with decoding fixed, and backbone portability uses Gemini-2.5-Pro (Comanici et al., 2025) on the matched subset, with settings in Appendix B.

4.3 Evaluation Protocol

We evaluate at the level of participant-owned *idea units*, the proposition-level claims, proposals, or decision-relevant facts the target participant introduces. To preclude leakage, each meeting is split. The first 60% constructs the participant profile, and the final 40% (the *interaction segment*) is the evaluation testbed.

Episode construction. Each idea unit instantiates a bounded *episode* with two reference points. The *cue* fixes the information boundary at which the delegate must act, using only causally observed context. The *anchor* is the ground-truth turn at which the participant articulated the target idea. The delegate then advances through a k-turn evaluation window after the cue (Liesenfeld and Dingemanse, 2024), with state updates consuming ground-truth turns only once they are causally observable.

Outcomes. A contribution earns recall credit only when (i) it matches the target proposition and (ii) it is neither hallucinated nor redundant. Each episode resolves into one of four outcomes. *Strict hit*: credited recovery at the anchor ($\Delta = 0$). *Loose hit*: credited recovery within the k-turn window after the cue ($0 \leq \Delta \leq k$). *Uncredited attempt*: the agent spoke but the utterance failed at least one credit criterion. *Not-attempted*: silent throughout. This separates *opportunity recognition* (did the agent recover the idea in time?) from *intervention quality* (how well did it intervene?), which we score independently. Uncredited attempts stratify into five categories analyzed in Section 5.3.

Metrics and quality. *Decision F1* quantifies floor-taking alignment at the anchor. During speaking turns, output quality is assessed via three binary indicators: *hallucination* (content unsupported by $\mathbf{m}_t$, $\mathbf{c}_{u^\star}$, or the preceding dialogue), *redundancy* (reiteration of previously resolved issues), and *off-topic* (divergence from the active topic). Furthermore, LLM-based judges evaluate *grounding*, *relevance*, *decision appropriateness*, and *context appropriateness* (for accepted matches) on a continuous 0–100 scale. All metrics are macro-averaged at the meeting level, reporting 95% bootstrap confidence intervals.

Human validation. Because LLM judges might be biased (Chen et al., 2024), we calibrate the idea-unit extractor and same-proposition judge against a stratified human-inspection sample. Three trained annotators (C1+ English) inspected 150 extracted idea units (balanced across the four participant roles) and 150 delegate–target same-proposition decisions (balanced across the credited and uncredited outcome categories), with a calibration round before independent scoring and joint adjudication of disagreements. Aggregate agreement between the adjudicated human label and the LLM judge reaches Cohen's $\kappa = 0.71$. The full protocol is stated in Appendix C.

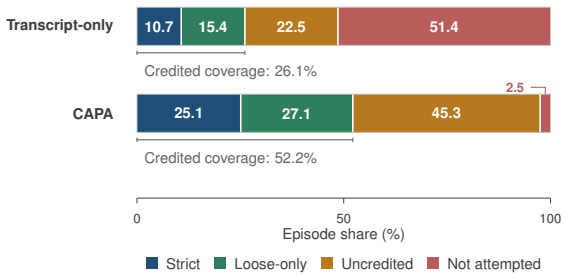

Figure 2: Episode-outcome distribution (%) over 1,126 idea-unit episodes. Per-system breakdown with 95% bootstrap CIs in Appendix G.

5 Results

This section reports three system-level findings on the 137-meeting AMI evaluation.

5.1 Coverage and Output Quality

An explicit meeting state lifts prompt-only LLM delegates out of silent abstention without degrading grounding fidelity, an effect that raw-context scaling alone cannot achieve.

Engagement gain. CAPA reduces missed opportunities from the dominant failure mode to a marginal one. Table 2 compares CAPA against two raw-transcript baselines: a transcript-only delegate, the closest controlled instantiation of Hu et al., 2025 under our protocol, and a lightweight Reflexion-style variant that adds post-action procedural memory but no explicit state, helpers, or state-directed recalibration. Both retain participant context and receive a longer 50-turn raw window in place of $\mathbf{m}_t$. On the 137-meeting evaluation, CAPA doubles credited recovery (loose recall $26.1 \rightarrow 52.2$, strict $10.7 \rightarrow 25.1$) and raises Decision F1 by 24.9 points ($38.1 \rightarrow 63.0$). The majority of this gain stems from improved opportunity recognition, isolating the baseline's core deficit to the long-range information dependencies of multi-party meetings.

Outcome redistribution. The drop in silent abstention translates directly into active, traceable engagement, with the remainder landing as uncredited attempts. Figure 2 shows the per-system outcome distribution over 1,126 idea-unit episodes. CAPA absorbs the 48.9% drop in not-attempted episodes through +26.1% credited and +22.8% uncredited attempts. Within the credited share, the

System	Strict recall ↑	Loose recall ↑	Uncredited ↓	Not attempted ↓	Decision F1 ↑	Hallucination ↓
Transcript-only	10.7	26.1	22.5	51.4	38.1	1.5
Reflexion	10.3	24.3	26.1	49.5	47.6	0.3
CAPA (ours)	**25.1**	**52.2**	45.3	**2.5**	**63.0**	**0.6**

Table 2: Headline comparison on 137 AMI meetings, macro-averaged %, bold = best per column. Transcript-only gets a 50-turn raw window in place of $\mathbf{m}_t$. Metric definitions in Section 4.3.

27.1% strict–loose gap reflects interventions arriving off-anchor inside the k=5 window, consistent with backchannels and pacing that postpone articulation. The composition of the uncredited bucket is taken up in Section 5.3.

Grounding preserved. The coverage gain carries no fabrication or repetition cost. Hallucination remains at 0.6% and redundancy at 0.0%, meaning the active state shifts the agent's participation policy while leaving its evidence grounding intact. Off-topic rises by 5.2% per episode against −48.9% not-attempted, the coverage–restraint trade-off that remains an open design question for fixed-weight LLM agents (Zhu et al., 2025). We read this profile as evidence that the meeting state shifts *participation policy* while leaving evidence grounding unchanged. Component-level attribution is isolated in Section 6.1 and Section 6.2.

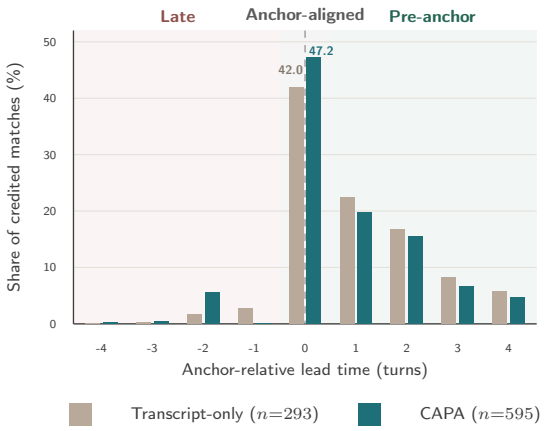

Figure 3: Timing distribution of strict and loose matches for CAPA (n=595) and the transcript-only baseline (n=293). Positive values indicate pre-anchor contributions, 0 represents anchor-aligned matches, and negative values indicate late recovery.

5.2 Engagement Timing

CAPA systematically anticipates intervention windows, revealing that the baseline's primary deficit lies in the binary floor-taking decision rather than in its subsequent timing mechanics.

Pre-anchor latency. When CAPA takes the floor, its contributions align with or precede the human participant's actual turn. Figure 3 reports the anchor-relative lead time of every credited match across the 137-meeting evaluation. Of CAPA's 595 credited matches, 93.8% are anchor-aligned or pre-anchor, with a median of 1.0 turn before the anchor (under the convention ≥ 0 synchronized, > 0 pre-anchor). This timing precision drives the absolute 24.9% increase in Decision F1 over the baseline (38.1 → 63.0).

The floor-taking bottleneck. Comparing both systems shows they diverge in how often they engage, but share identical timing distributions when they do. Intervention rates differ by 48.9% (97.5% for CAPA versus 48.6% for the baseline), while the baseline's lead-time distribution on its few credited matches mirrors CAPA's. This asymmetry isolates the prompt-only failure specifically to opportunity recognition, consistent with prior reports that multi-party turn anticipation is a discrete bottleneck for standard LLMs (Penzo et al., 2024; Hilgert and Niehues, 2025).

5.3 Uncredited-Attempt Taxonomy

By explicitly tracking meeting state, CAPA shifts the baseline's tendency to remain silently inactive with traceable, active engagement. When the system misses a target, it produces diagnosable near-misses rather than opaque omissions.

Error composition. The 45.3% uncredited attempt rate decomposes into four mutually exclusive categories of selection and timing discrepancies (Figure 4). We classify all 504 uncredited episodes CAPA produces across the 137-meeting evaluation as: *incomplete match* (correct proposition, missing qualifier), *different proposition* (a briefing-supported point distinct from the reference), *late match* (correct proposition generated outside the k-turn window), and *safety/quality flag* (off-topic or redundancy filter triggered). This error taxonomy maps to the system's modular com-

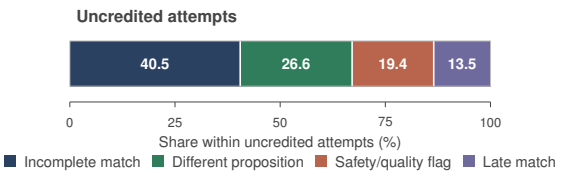

Figure 4: Uncredited-attempt taxonomy over all 504 uncredited episodes CAPA produces across the 137-meeting evaluation (45.3% uncredited bucket). Each attempt is assigned to exactly one of four mutually exclusive failure types defined in Section 5.3.

Cue. The group asks, "Are LEDs beneath the buttons?" The UI participant has an uncovered briefing point about LED placement.
State read. Open question; UI has highest speaker fit; LED placement remains uncovered.
Transcript-only. SILENT.
CAPA. SPEAK: "One idea could be to integrate the LEDs around the buttons or even within them."
Reference. The UI participant next says, "Yeah, around the buttons, or in the buttons even." ⇒ strict match, no safety flag.

Figure 5: Compact qualitative trace. Explicit floor, speaker-fit, and coverage fields support a timely contribution where the raw-transcript delegate remains silent. The complete trace is in Appendix F.

ponents. *Incomplete match* isolates a realization deficit in the `Generator`. *Different proposition* errors indicate imprecise candidate curation against. *Late matches* reflect timing misalignment driven by the $\mathbf{m}_t$ and the speaker scorer. Finally, *safety/quality flags* denote expected candidate rejections by the output filter during drafting.

System observability. This shift from omission to selection yields a methodological benefit for system diagnostics. The transcript-only delegate's 51.4% silent-abstention rate produces no decision trace, precluding error attribution. CAPA's uncredited attempts generate a stepwise record, isolating failures and enabling targeted refinement.

6 Ablations

To understand the mechanistic drivers behind CAPA's performance, we isolate its explicit state tracking and post-action recalibration. Because the architecture decouples the floor-taking decision from generation, we can independently ablate these modules to trace their specific effects on agent behavior. By removing these components, we show that state tracking exclusively governs the binary participation policy, while recalibration refines the qualitative selection of what is said. Finally, we verify robustness across varying context and evaluation windows, underlying LLM backbones, and a non-scenario meeting corpus. Comprehensive ablation tables and experiment descriptions are provided in Appendix D.

6.1 Meeting State Carries the Threshold

The explicit meeting state closes the recognition gap in multi-party delegation. On a 20-meeting subset, we substitute $\mathbf{m}_t$ and the recalibration step for a longer 50-turn raw window (Table 8). This leads to a loose recall drop by 22.4%, strict recall reduces by 14.5% and Decision F1 by 24.7%, while redundancy increases by 10.7%. The reduction in behavioral metrics against a static grounding score confirms that the state mechanism determines whether and when to engage. Figure 5 makes this mechanism concrete as the state exposes an open question, participant fit, and an uncovered briefing point that the raw-transcript delegate does not act on.

6.2 Recalibration Refines Selection

Post-action recalibration narrows contribution selection without altering the floor-taking threshold, demonstrating that state-update feedback and output-rewriting feedback target distinct failure modes (Table 9). Removing only the recalibration step while keeping $\mathbf{m}_t$ and the upstream stack intact causes loose recall to drop by 3.9% and uncredited attempts to rise by 4.6%. Meanwhile, Decision F1 and grounding remain statistically unchanged. This confirms that output-rewriting baselines (e.g., Reflexion) are structurally inapplicable to our primary failure regime: they exist to refine drafted utterances, whereas multi-party delegates primarily fail via silent abstention, leaving no draft to refine.

6.3 Backbone Model Portability

The core architectural mechanism transfers across backbones, though the specific safety–coverage operating point is model-dependent. Across GPT-4o, Gemini-2.5-Pro, Llama-3.3-70B, and Qwen3.6-27B, Decision F1 stays within 57.9 to 69.2 while coverage and precision trade off along one curve. Permissive backbones such as Gemini gain recall at the cost of off-topic contributions. Conservative backbones such as Qwen abstain more (17.8%, still about a third of the transcript-only baseline's 51.4%) while selecting most cleanly (off-topic 3.3%, hallucination 0.0%). A new backbone can be calibrated on held-out episodes by tightening the printed suitability and restraint constants until

precision meets the deployment requirement, with no retraining (full table in Table 16).

6.4 Transfer Beyond AMI

As a cross-corpus robustness check, we test whether CAPA depends on AMI's scripted four-person format. We replay 10 ICSI meetings, i.e., real research-group discussions with six participants on average and up to ten (Janin et al., 2003), under the identical protocol, adapting only the profile construction. We observe a similar behavior, where silence remains near zero (1.2% versus 2.5% on AMI), loose recall is higher (73.6% [61.1, 86.0] versus 52.2%), Decision F1 is comparable (64.8% [57.4, 74.9] versus 63.0%), and hallucination stays low (1.8%), with redundancy at 0.0% and off-topic at 4.9%. The full table reports 42 participant runs, 47 episodes, and 235 evaluated steps (Table 13).

6.5 Sensitivity to Evaluation Parameters

The primary findings remain stable under variations in the raw-window length and exhibit expected coverage–horizon trade-offs. **Raw context window (N).** With $\mathbf{m}_t$ available, the raw transcript window is no longer load-bearing. Across $N \in \{10, 20, 30\}$, loose recall remains stable between 50.5% and 56.7%, and Decision F1 between 61.1% and 63.1% (Appendix I.2). **Evaluation window (k).** Expanding k admits more credited matches (loose recall rises from $43.9 \rightarrow 62.7$ across $k \in \{3, 5, 7, 9\}$) without artificially inflating the system's anchor synchrony (Appendix I.4).

7 Final Considerations

In online meeting delegation, prompt-only LLM proxies fail to recognize when to intervene, defaulting to silent abstention during most talking opportunities. We introduced CAPA, a *perceive–act–recalibrate* loop operating over an explicit meeting state, as an architectural response to this recognition gap. To evaluate it, we designed an episode-level protocol that scores whether, when, and what a delegate contributes relative to a human participant's actual idea units. On 137 AMI meetings, CAPA reduced the silence rate from 51.4% to 2.5%, doubled credited recovery ($26.1 \rightarrow 52.2$), and preserved grounding fidelity with a 0.6% hallucination rate. Our findings demonstrate that effective multi-party recognition requires maintaining variables like stance, coverage, and floor control as explicit state fields, as LLMs cannot reliably recover them from raw context streams. By decoupling the floor-taking decision from surface generation, CAPA resolves the upstream recognition error and allows for transparent, module-attributable selection errors. Furthermore, we establish that this state-updating correction channel is non-substitutable by output-rewriting paradigms as they are poorly matched to the primary failure regime we observe. Robustness checks confirm the core architecture transfers across configurations. The generalization of this state-driven architecture to other domains with similar causal information boundaries, e.g., customer-support handoff, remain future directions.

Limitations

The primary evaluation of CAPA focuses on the AMI scenario portion. Alternative corpora do not provide all protocol conditions without adaptation. ELITR (Nedoluzhko et al., 2022), used by related meeting delegation work (Hu et al., 2025), and ICSI (Janin et al., 2003) cover conceptually similar formal settings but lack the continuity that AMI provides. For the ICSI robustness probe (Section 6.4), we therefore adapt only profile construction while holding causal replay fixed. Other meeting datasets such as parliamentary or formal institutional datasets (Hu et al., 2023; Zhong et al., 2021) enforce rigid turn-taking structures where spontaneous floor-taking is structurally not possible.

The system's performance depends significantly on the capabilities of the underlying language model. While our implementation uses GPT-4o, models with different reasoning capabilities or smaller context windows may produce less accurate delegation capabilities. Our ablation studies suggest that performance can carry over to other model families, demonstrating the framework's architectural robustness.

Acknowledgements

This work was supported by the Lower Saxony Ministry of Science and Culture and the VW Foundation.

References

Joseph A. Allen and Nale Lehmann-Willenbrock. 2023. The key features of workplace meetings: Conceptualizing the why, how, and what of meetings at work. *Organizational Psychology Review*, 13(4):355–378.

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. *Preprint*, arXiv:2508.08242.

Sumit Asthana, Sagi Hilleli, Pengcheng He, and Aaron Halfaker. 2025. Summaries, highlights, and action items: Design, implementation and evaluation of an LLM-powered meeting recap system. *Proceedings of the ACM on Human-Computer Interaction*, 9(2):1–29.

Jean Carletta, Wessel Kraaij, Simone Ashby, Sebastien Bourban, Michael Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Melissa Kronenthal, Guillaume Lathoud, Michael Lincoln, Agnes Lisowska, Wilfried Post, Dennis Reidsma, Pierre Wellner, and Iain McCowan. 2005. The AMI meeting corpus. In *Proceedings of the Symposium on Annotating and Measuring Meeting Behavior*.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.

Xinyue Chen, Nathan Yap, Xinyi Lu, Aylin Gunal, and Xu Wang. 2025. MeetMap: Real-time collaborative dialogue mapping with LLMs in online meetings. *Preprint*, arXiv:2502.01564.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

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*, pages 6583–6591.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Babu Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. 2025. Context length alone hurts LLM performance despite perfect retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23281–23298, Suzhou, China. Association for Computational Linguistics.

Sian Gooding and Edward Grefenstette. 2025. Interaction dynamics as a reward signal for LLMs. *Preprint*, arXiv:2511.08394.

Lukas Hilgert and Jan Niehues. 2025. Next speaker prediction for multi-speaker dialogue with large language models. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2025)*, pages 69–79, Barcelona, Spain (Hybrid). Association for Computational Linguistics.

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)*, pages 283–316, Suzhou, China. Association for Computational Linguistics.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. MeetingBank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16409–16423, Toronto, Canada. Association for Computational Linguistics.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, volume 1, pages I–I. IEEE.

Mateusz Jęcniacki and Martí Carmona Serrat. 2025. Humanlike multi-user agent (HUMA): Designing a deceptively human AI facilitator for group chats. *Preprint*, arXiv:2511.17315.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134.

Frederic Kirstein, Terry Ruas, Robert Kratel, and Bela Gipp. 2024. Tell me what I need to know: Exploring LLM-based (personalized) abstractive multi-source meeting summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 920–939, Miami, Florida, US. Association for Computational Linguistics.

Joanne Leong, John Tang, Edward Cutrell, Sasa Junuzovic, Greg P. Baribault, and Kori Inkpen. 2024. Dittos: Personalized, embodied agents that participate in meetings when you are unavailable. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2).

Andreas Liesenfeld and Mark Dingemanse. 2024. Interactive probes: Towards action-level evaluation for dialogue systems. *Discourse & Communication*, 18(6):954–964.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Fukun Ma, Kaibin Tian, Jieting Xue, Xiaoyi Wang, Ye Ma, Quan Chen, Peng Jiang, and Lijie Wen. 2025. Improving preference alignment of LLM with inference-free self-refinement. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24459–24473, Suzhou, China. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

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. *Preprint*, arXiv:2401.04883.

Microsoft Work Trend Index. 2025. Breaking down the infinite workday. https://www.microsoft.com/en-us/worklab/work-trend-index/breaking-down-infinite-workday. Accessed 2026-01-15.

Lillio Mok, Lu Sun, Shilad Sen, and Bahareh Sarrafzadeh. 2023. Challenging but connective: Large-scale characteristics of synchronous collaboration across time zones. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM.

Mukesh Nathan, Mercan Topkara, Jennifer C. Lai, Shimei Pan, Steve Wood, Jeff Boston, and Loren G. Terveen. 2012. In case you missed it: benefits of attendee-shared annotations for non-attendees of remote meetings. In *CSCW '12: Computer Supported Cooperative Work, Seattle, WA, USA, February 11–15, 2012*, pages 339–348. ACM.

Anna Nedoluzhko, Muskaan Singh, Marie Hledíková, Tirthankar Ghosal, and Ondřej Bojar. 2022. ELITR minuting corpus: A novel dataset for automatic minuting from multi-party meetings in English and Czech. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3174–3182, Marseille, France. European Language Resources Association.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, and 1 others. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

Nicolò Penzo, Maryam Sajedinia, Bruno Lepri, Sara Tonelli, and Marco Guerini. 2024. Do LLMs suffer from multi-party hangover? a diagnostic approach to addressee recognition and response selection in conversations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11210–11233, Miami, Florida, USA. Association for Computational Linguistics.

Hoang Phan, Victor Li, and Qi Lei. 2025. Think twice, generate once: Safeguarding by progressive self-reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9466–9483, Suzhou, China. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK.

Sagar Sapkota, Mohammad Saqib Hasan, Mubarak Shah, and Santu Karmaker. 2025. Multi-party conversational agents: A survey. *Preprint*, arXiv:2505.18845.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Dustin J. Sleesman, Donald E. Conlon, Gerry McNamara, and Jonathan E. Miles. 2012. Cleaning up the big muddy: A meta-analytic review of the determinants of escalation of commitment. *Academy of Management Journal*, 55(3):541–562.

Jason D. Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

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*, pages 5905–5921, Online. Association for Computational Linguistics.

Runchuan Zhu, Zinco Jiang, Jiang Wu, Zhipeng Ma, Jiahe Song, Fengshuo Bai, Dahua Lin, Lijun Wu, and Conghui He. 2025. GRAIT: Gradient-driven

refusal-aware instruction tuning for effective hallucination mitigation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4006–4021, Albuquerque, New Mexico. Association for Computational Linguistics.

Ethical Considerations

CAPA is evaluated offline on the publicly released AMI Meeting Corpus (CC BY 4.0). AMI participants consented to recording for research use, which we treat as a sufficient basis for offline transcript analysis but not for live impersonation. Any deployment of state-conditioned delegation in a live meeting would require explicit, recurring consent from all participants present, because the architecture lets the agent surface a participant-owned proposition the represented participant may not have stated themselves. Misuse paths include impersonation under false pretenses, plausible-deniability speech laundering, and asymmetric advantage of represented over unrepresented participants. We release CAPA under MIT license as research infrastructure. Production use should go through participant consent.

A Open Resources and Licensing

A.1 Repository and License

The code for this work is available on `https://github.com/FKIRSTE/emnlp2026-meeting-delegation` under an MIT license together with the evaluation protocol, prompt templates, judges, and analysis scripts after acceptance.

A.2 Datasets and Licensing

We evaluate on the scenario portion of the AMI Meeting Corpus (Carletta et al., 2005), distributed under CC BY 4.0. Table 3 summarizes the license and the high-level corpus statistics.

Dataset	License	Size	Domain
AMI (scenario)	CC BY 4.0	137	remote-control design

Table 3: Dataset licensing and overview. *Size*: number of meetings used.

B Experimental Setup Details

B.1 Implementation

We implement CAPA in Python. All LLM calls go through a single API wrapper with retry-on-error and structured JSON output validation against the schema for each module (`Perceiver`, `Predictor`, Environment Judge, Delegate Judge, `Recalibrator`, `Controller`, `Generator`, helpers). Schema-constrained decoding guarantees that every module emits parseable structured output. The `Shared Memory` is a typed dictionary indexed by participant, meeting, and turn.

B.2 Model Specs

GPT-4o is the default backbone for all main experiments. The backbone-portability study evaluates Gemini-2.5-Pro, Llama-3.3-70B, and Qwen3.6-27B. Table 4 lists the reported endpoints; decoding settings remain fixed.

Model	Snapshot	Parameters	Provider
GPT-4o	2024-11-20	$\sim$ 200B (est.)	OpenAI
Gemini-2.5-Pro	2025-06-17	not disclosed	Google
Llama-3.3-70B	2024-12-06	70B	Groq
Qwen3.6-27B	2026-04-21	27B	Groq

Table 4: Model snapshots and providers. GPT-4o is the default backbone. Snapshot names come from provider metadata when exposed.

B.3 Hyperparameters

We keep default values for top-p (1.0), frequency penalty (0.0), and presence penalty (0.0). Temperature is T=0.1 for every LLM module except the `Generator`, which runs at T=0.3 to allow modest natural-language variation after the `Controller` has committed to a semantic point. Decoding parameters are held fixed across systems and conditions. Default episode parameters are N=20 preceding utterances and evaluation window k=5, both varied in the robustness checks (Appendix I). All metrics are macro-averaged at the meeting level with 95% bootstrap CIs (B=10,000).

B.4 Computational Footprint

We distinguish response latency from total computation per decision. Only the `Controller`, its three helpers, and the conditional `Generator` lie on the response path. The `Perceiver` runs once per episode and can overlap with ongoing speech. The `Predictor`, judges, and `Recalibrator` execute after the floor decision and affect later turns. The critical-path rows therefore measure the delay before a contribution, while the full-decision rows measure all computation attributed to one decision. All reported latencies are wall-clock measurements

from the experimental harness and include its fixed three-second post-call throttle.

GPT-4o latency. The observed response path takes 21.14 s on average, with a p95 of 36.49 s. The harness throttle contributes 10.01 s to the mean. Removing it analytically gives a provider-only response latency of 11.12 s mean and 19.05 s p95 across all 110 decisions. Among the 72 decisions that execute the floor-taking LLM path, provider calls take 16.99 s mean and 19.39 s p95.

†The `Perceiver` runs once per five-decision episode. Its episode-level latency is 31.35 s mean and 115.78 s p95, which Table 5 amortizes across the five decisions. The 52.30 s full-decision mean also includes post-turn computation and therefore does not represent response latency.

Open-weight latency. On Groq, Qwen's observed response path takes 19.03 s on average and Llama's takes 17.26 s. Their full-decision means are 45.85 s and 44.73 s because these totals include background updates and the amortized `Perceiver`. †For both backbones, the `Perceiver` runs once per five-decision episode and its measured latency is amortized across those decisions.

Cost accounting. The GPT-4o response path costs \$0.1069 per decision, and the complete architecture costs \$0.2090 after including background updates and the amortized `Perceiver`. GPT-4o costs use provider-reported cached-token counts and rates of \$2.50/M uncached input tokens, \$1.25/M cached input tokens, and \$10/M output tokens. For Qwen, the rates are \$0.60/M input tokens and \$3.00/M output tokens. For Llama, they are \$0.59/M input tokens and \$0.79/M output tokens. ‡Only open-weight `Perceiver` token use is estimated; all runtime-module usage is provider-reported.

C Human Evaluation Protocol

C.1 Annotators

Three trained annotators (C1+ English) ran the inspection, recruited as research assistants or doctoral candidates with academic background in multi-party dialogue. Each annotator gave explicit consent for the anonymized annotations to be used in this work. The protocol received approval from our institution's ethics committee before any data collection.

C.2 Sampling and Procedure

We drew a stratified sample of 150 extracted idea units, balanced across the four participant roles (PM, ID, UI, ME), and 150 delegate–target same-proposition decisions, balanced across the credited (strict, loose-only) and uncredited outcome categories. Each idea unit got a binary label (is this a proposition-level idea unit?). Each delegate–target pair got a binary label (does the delegate's utterance express the same proposition as the target?). Annotators were blind to which delegate produced each utterance and to which condition the episode belonged. Each annotator scored independently after a calibration round, and disagreements were adjudicated through joint discussion.

C.3 Agreement

We compute Cohen's κ between the adjudicated human label and the LLM judge across the judged dimensions (idea-unit extraction, same-proposition matching, hallucination, redundancy, off-topic). Aggregate agreement reaches $\kappa = 0.71$, in the substantial-agreement range.

D Mechanism Ablation Tables

The two ablations isolate complementary effects. Removing the meeting state $\mathbf{m}_t$ collapses coverage, Decision F1, and redundancy together (Table 8). Removing the `Recalibrator` narrows contribution selection while the floor-taking threshold stays inside bootstrap noise (Table 9). Figure 6 and Figure 7 render both tables as grouped bar charts.

E Conditional Quality Scores

Table 10 reports CAPA's judge-based quality scores on the 137-meeting evaluation. Each dimension is scored only over the steps where it applies. Decision appropriateness covers every decision step, relevance and grounding cover speaking turns, and context appropriateness covers accepted same-proposition matches only. A like-for-like cross-system comparison on these scores is not meaningful here, because the transcript-only delegate is silent in 51.4% of episodes and its conditional distribution is taken over a smaller and more selective subset. We therefore read the absolute numbers as architectural characterization of CAPA and defer the within-architecture comparison to the grounding-invariance result under the meeting-state ablation (Section 6.1).

Module	Calls/dec.	Timing	Mean s	p95 s	Input tok.	Cached input	Output tok.	Cost/dec.
Perceiver[†]	0.845	pre-episode	6.27	23.16	10,322	5,272	787	$0.0271
Predictor	1.000	post-turn	3.90	4.08	4,540	1,439	54	$0.0101
Controller + 3 helpers	2.991	floor path	19.85	33.51	46,009	26,195	2,247	$0.1048
Generator	0.336	floor path, conditional	1.28	3.91	1,242	904	16	$0.0021
Environment Judge	1.000	post-turn	4.14	4.37	7,198	5,120	98	$0.0126
Delegate Judge	1.345	post-turn	6.50	10.81	10,619	5,750	317	$0.0225
Recalibrator	2.000	post-turn	10.33	11.19	10,658	5,632	1,025	$0.0299
Critical path (floor-taking)	**3.327**		**21.14**	**36.49**	**47,251**	**27,099**	**2,263**	**$0.1069**
Runtime per decision, excluding Perceiver	**8.673**	floor + post-turn	**46.03**	**65.12**	**80,267**	**45,040**	**3,757**	**$0.1819**
Architecture per decision, including Perceiver	**9.518**	amortized	**52.30[†]**	–	**90,590**	**50,312**	**4,544**	**$0.2090**

Table 5: GPT-4o latency, token use, and cost over 110 decisions in 22 five-step episodes. CAPA spoke on 37 decisions; the floor-taking LLM path ran on 72.

Module	Calls/dec.	Timing	Mean s	p95 s	Input tok./dec.	Output tok./dec.	Cost/dec.
Perceiver[†‡]	0.714	pre-episode	5.32	12.21	≈11,617	≈753	≈$0.00923
Predictor	1.000	post-turn	3.59	3.65	4,197	64	$0.00271
Controller + 3 helpers	2.842	floor path	18.35	31.97	44,922	2,982	$0.03590
Generator	0.189	floor path, conditional	0.67	3.53	671	9	$0.00043
Environment Judge	1.000	post-turn	4.04	4.19	6,903	191	$0.00472
Delegate Judge	1.189	post-turn	4.76	7.81	8,296	200	$0.00558
Recalibrator	2.000	post-turn	9.10	9.64	10,757	1,037	$0.00956
Critical path (floor-taking)	**3.032**		**19.03**	**34.86**	**45,592**	**2,991**	**$0.03633**
Runtime per decision, excluding Perceiver	**8.221**	floor + post-turn	**40.54**	**59.39**	**75,746**	**4,483**	**$0.05890**
Architecture per decision, including Perceiver	**8.935**	amortized	**45.85[†]**	–	**≈87,363**	**≈5,235**	**≈$0.06813**

Table 6: Qwen3.6-27B computational footprint over 95 decisions in 19 complete episodes.

Module	Calls/dec.	Timing	Mean s	p95 s	Input tok./dec.	Output tok./dec.	Cost/dec.
Perceiver[†‡]	0.724	pre-episode	5.70	22.38	≈9,011	≈656	≈$0.00584
Predictor	1.000	post-turn	3.49	3.75	3,505	55	$0.00211
Controller + 3 helpers	2.781	floor path	15.80	27.69	38,353	1,743	$0.02401
Generator	0.448	floor path, conditional	1.58	3.67	1,525	23	$0.00092
Environment Judge	1.000	post-turn	3.89	4.08	6,410	120	$0.00388
Delegate Judge	1.448	post-turn	5.55	7.62	8,274	135	$0.00499
Recalibrator	2.000	post-turn	8.82	9.32	9,547	931	$0.00637
Critical path (floor-taking)	**3.229**		**17.26**	**28.04**	**39,878**	**1,766**	**$0.02492**
Runtime per decision, excluding Perceiver	**8.676**	floor + post-turn	**39.03**	**51.32**	**67,614**	**3,006**	**$0.04227**
Architecture per decision, including Perceiver	**9.400**	amortized	**44.73[†]**	–	**≈76,625**	**≈3,663**	**≈$0.04811**

Table 7: Llama-3.3-70B computational footprint over 105 decisions in 21 complete episodes.

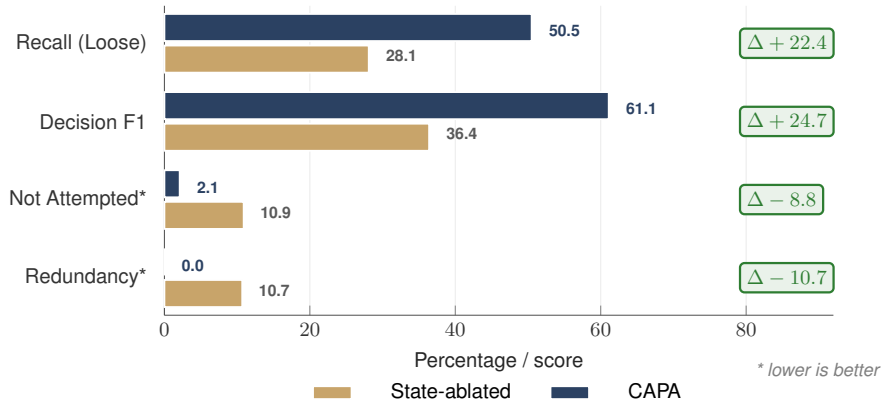

Figure 6: Meeting-state ablation, grouped bar view of Table 8.

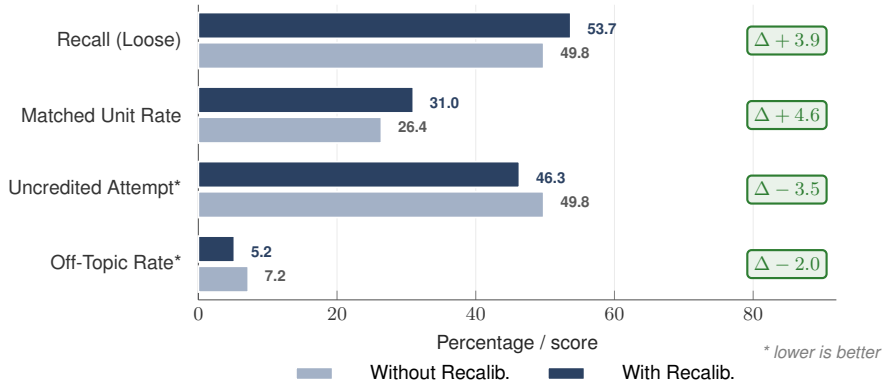

Figure 7: Recalibration ablation, grouped bar view of Table 9.

Metric	CAPA (ours)	State-Ablated	Δ
Strict recall (%) ↑	**23.0** [15.4, 30.7]	8.5 [4.1, 13.4]	+14.5
Loose recall (%) ↑	**50.5** [40.7, 59.7]	28.1 [20.1, 36.8]	+22.4
Not Attempted (%) ↓	**2.1** [0.0, 5.9]	10.9 [5.9, 16.2]	−8.8
Decision F1 (%) ↑	**61.1** [56.9, 65.1]	36.4 [33.2, 39.9]	+24.7
Redundancy (%) ↓	**0.0** [0.0, 0.0]	10.7 [6.7, 15.2]	−10.7
Grounding (Score) ↑	**74.3** [73.2, 75.3]	72.7 [71.7, 73.7]	+1.6

Table 8: Meeting-state ablation on 20 matched meetings (k=5, B=10,000). Bold = better per row. The ablated system omits $\mathbf{m}_t$ and recalibration and gets a 50-turn raw window.

Metric	With Recalib. (ours)	Without
Strict recall (%) ↑	**28.0** [20.7, 36.0]	26.0 [19.1, 32.8]
Loose recall (%) ↑	**53.7** [44.1, 63.6]	49.8 [41.7, 57.2]
Matched-unit rate (%) ↑	**31.0** [23.1, 40.1]	26.4 [21.0, 32.0]
Uncredited attempt (%) ↓	**46.3** [36.4, 55.9]	49.8 [42.5, 57.9]
Decision F1 (%) ↑	66.2 [61.9, 70.4]	**66.7** [62.3, 71.0]
Off-topic rate (%) ↓	**5.2** [2.3, 8.5]	7.2 [3.6, 11.4]
Grounding (Score) ↑	74.1 [72.8, 75.3]	74.3 [73.1, 75.4]

Table 9: Recalibration ablation on 20 matched meetings (k=5, B=10,000). Bold = better per row. Both variants share $\mathbf{m}_t$, the `Predictor`, and the judges.

F Detailed Qualitative Trace

Figure 8 expands the compact main-paper example with the full observed context, state snapshot, semantic plan, realized utterance, and verifier-backed credit outcome.

G Outcome Decomposition

Table 11 gives the per-system breakdown of the episode outcomes summarized in Figure 2, with 95% bootstrap CIs on every rate. The headline shift is the not-attempted rate, which drops from

Dimension	Mean	95% CI
Decision appropriateness	88.8	[88.5, 89.0]
Context appropriateness†	88.0	[87.4, 88.5]
Relevance	78.0	[77.3, 78.6]
Grounding	73.8	[73.3, 74.3]

Table 10: CAPA judge-based quality scores (0–100, higher better). † scored only on accepted same-proposition matches. B=10,000 bootstrap resamples.

Episode. AMI `TS3004b` (our `Meeting28test`); $u^\star$ = User Interface.
Cue=`Meeting28_409`; anchor=`Meeting28_410`.

Briefing point. *Suggest considering LED placement around or within the buttons to enhance visibility and user experience.*

Observed context, ending at the cue.
PM: *"That looks different, yeah."*
UI: *"Yeah."*
ID: *"Hmm."*
UI: *"Otherwise we will just make another standard, and our motto also is–"*
ME: *"Are LEDs beneath the buttons?"*

CAPA view at cue *(condensed)*
`floor` open question to group
`focus` LED placement and visual distinctiveness
`speaker fit` UI ranked highest on the micro-topic; strong cue
`candidate` LED placement around or within buttons is viable

Transcript-only delegate: SILENT. *Rationale: "no explicit or implicit cue for the User Interface participant."*

CAPA: SPEAK. Semantic plan: *answer the LED-placement question with a concrete UI placement option. "One idea could be to integrate the LEDs around the buttons or even within them, which might enhance visibility and improve the user experience."*

Ground truth and credit. UI next turn: *"Yeah, around the buttons, or in the buttons even."* ⇒ **strict match**.
`matched_utt=Meeting28_410`; $\Delta t = 0$; no hallucination/off-topic/redundancy flags.

Takeaway. The transcript-only delegate treats a group question as insufficiently addressed to UI; CAPA uses the state-conditioned floor and speaker-fit signals to identify a timely participant-owned answer.

Figure 8: Qualitative trace of one decision episode A single fixed-window episode where CAPA succeeds on an implicit group cue and the transcript-only delegate stays silent. The trace shows the available briefing point, observed context, compressed state/control view, realised actions, and verifier-backed credit outcome. Quotes are lightly cleaned for readability.

Outcome	Transcript-only	CAPA (ours)
Strict hit	10.7 [8.7, 12.8]	25.1 [22.1, 28.2]
Loose-only hit	15.4 [12.7, 18.2]	27.1 [24.3, 29.9]
Uncredited attempt	22.5 [19.7, 25.3]	45.3 [41.7, 48.9]
Not attempted	51.4 [47.8, 55.2]	2.5 [1.6, 3.7]

Table 11: Per-system outcome decomposition over the 137-meeting fixed-window episode set. Rates are macro-averaged over meetings with retained official episodes (B=10,000 bootstrap resamples).

Metric	AMI	ICSI
Never attempted ↓	2.5	1.2 [0.0, 3.7]
Loose recall ↑	52.2	73.6 [61.1, 86.0]
Decision F1 ↑	63.0	64.8 [57.4, 74.9]
Hallucination ↓	0.6	1.8 [0.0, 4.3]

Table 12: Transfer from AMI to ICSI. Entries are percentages; ICSI intervals are 95% bootstrap CIs.

51.4% for the transcript-only delegate to 2.5% for CAPA. The two systems' CIs do not overlap on any outcome category.

H ICSI Transfer

We replay 10 ICSI research-group meetings under the same causal fixed-window protocol, yielding 42 eligible participant runs, 47 episodes, 235 evaluated steps, and 79 spoken steps. Meetings contain six participants on average and up to ten.

I Robustness Checks

The four checks in this section each rule out a specific alternative reading of the headline result. Per-role consistency rules out the gain being driven by a single participant role (Table 14). The context-window sweep rules out larger raw windows as the source of the coverage gain, isolating the contribution of the explicit state (Table 15, Figure 9). The evaluation-window sweep rules out k=5 being an artificially favorable scoring window (Table 17). The backbone-portability check rules out the architecture being tied to GPT-4o (Table 16). None of the four shifts the mechanism conclusions of Section 5 or Section 6.

Metric	ICSI	95% CI
Decision accuracy	73.5	[67.4, 81.3]
Decision precision	74.9	[63.2, 86.6]
Decision recall	61.1	[50.4, 73.3]
Decision F1	**64.8**	[57.4, 74.9]
Strict idea recall	29.8	[10.2, 53.2]
Loose idea recall	**73.6**	[61.1, 86.0]
Loose-only	43.8	[25.3, 63.0]
Attempted but inadequate	25.2	[13.6, 36.4]
Never attempted	**1.2**	[0.0, 3.7]
Decision appropriateness	88.2	[86.7, 89.7]
Context appropriateness	87.2	[85.1, 88.9]
Relevance	80.3	[75.5, 84.2]
Grounding	74.5	[71.9, 76.9]
Hallucination	**1.8**	[0.0, 4.3]
Off-topic	4.9	[0.0, 13.2]
Redundancy	0.0	[0.0, 0.0]

Table 13: Full ICSI results over 10 meetings, 42 participant runs, 47 episodes, and 235 evaluated steps. Rates and scores are percentages; intervals are 95% bootstrap CIs.

Role	Meetings / Units	Strict ↑	Loose ↑	Dec. F1 ↑	Not Att. ↓
PM	119 / 324	24.5 [19.0, 30.4]	43.5 [37.1, 50.1]	62.6 [60.0, 65.1]	3.2 [1.5, 5.0]
ID	112 / 304	30.7 [25.1, 36.9]	60.7 [54.0, 67.4]	66.7 [63.5, 70.0]	2.6 [0.7, 5.0]
UI	104 / 267	26.1 [20.1, 32.4]	61.8 [54.6, 68.9]	64.8 [62.1, 67.6]	0.9 [0.0, 2.2]
ME	99 / 231	25.3 [18.9, 32.2]	57.1 [49.5, 64.7]	65.1 [61.6, 68.5]	1.3 [0.0, 3.0]

Table 14: Per-role consistency of CAPA over the 137-meeting pool. PM, ID, UI, ME = Project Manager, Industrial Designer, User Interface, Marketing. "Meetings / Units" = meetings with at least one retained idea unit for that role and the corresponding idea-unit count. Entries are macro-averaged percentages with 95% bootstrap CIs (B=10,000).

I.1 Per-Role Consistency

We break out CAPA's coverage and floor-taking by the four AMI participant roles, Project Manager (PM), Industrial Designer (ID), User Interface (UI), and Marketing (ME), on the full 137-meeting pool (Table 14). Loose recall ranges from 43.5% on PM to 61.8% on UI, Decision F1 sits inside a four-point band (62.6–66.7), and the not-attempted rate stays below 3.5% on every role. The aggregate gain therefore spreads broadly across roles, with no single participant type accounting for it.

I.2 Context Window N

We hold the future evaluation window at k=5 and vary the number of preceding utterances visible to the delegate, $N \in \{10, 20, 30\}$, on a 20-meeting matched subset. The check measures how much recent raw dialogue CAPA requires for reliable floor-taking and content selection once the explicit meeting state is in place.

Metric	N=10	N=20	N=30
Loose recall (%) ↑	56.7 [49.0, 63.7]	50.5 [40.7, 59.7]	56.5 [49.5, 63.3]
Decision F1 (%) ↑	63.1 [60.3, 66.0]	61.1 [56.9, 65.1]	62.3 [58.5, 65.7]
Grounding (Score) ↑	74.9 [73.7, 76.0]	74.3 [73.2, 75.3]	74.7 [73.8, 75.6]
Off-topic rate (%) ↓	3.2 [1.3, 5.4]	3.4 [1.5, 5.6]	5.6 [2.4, 9.5]

Table 15: Effect of context-window size N. k=5, B=10,000 bootstrap resamples, CAPA with recalibration.

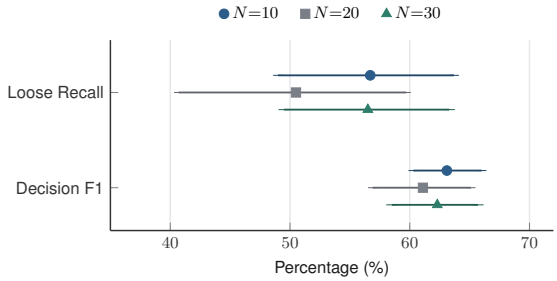

Figure 9: Performance across context-window sizes $N \in \{10, 20, 30\}$. Error bars are 95% bootstrap CIs.

Performance is stable across the sweep. Loose recall stays between 50.5% and 56.7% and Decision F1 between 61.1% and 63.1%, with overlapping CIs and no monotonic trend in N. Retained episode counts are also comparable across settings (162–166 episodes), so the pattern is not an artifact of a shifting evaluation denominator. The stability matches CAPA's division of labor. The raw window supplies recent conversational evidence such as dialogue acts, addressee, and floor status, while longer-range information about decisions, unresolved issues, stances, and coverage lives in $\mathbf{m}_t$. The off-topic rate is numerically higher at N=30 than at N=10 or N=20, but the CIs overlap, and we treat the difference as a single-point fluctuation consistent with prior reports that longer contexts are not always used reliably (Liu et al., 2024; Du et al., 2025). We adopt N=20 as the default, which preserves local conversational cues without diluting the system's reliance on the explicit state.

I.3 Backbone Portability

We first test portability through a controlled end-to-end backbone swap. GPT-4o and Gemini-2.5-Pro

Metric	GPT-4o	Gemini-2.5-Pro	Llama-3.3-70B	Qwen3.6-27B
Strict recall (%) ↑	28.0 [20.7, 36.0]	38.3 [33.2, 43.8]	–	–
Loose recall (%) ↑	53.7 [44.1, 63.6]	70.9 [65.6, 76.5]	46.9	57.0
Uncredited attempt (%) ↓	46.3 [36.4, 55.9]	29.1 [23.5, 34.4]	–	–
Not attempted (%) ↓	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	1.4	17.8
Decision F1 (%) ↑	66.2 [61.9, 70.4]	65.6 [61.2, 70.3]	69.2	57.9
Matched-unit rate (%) ↑	31.0 [23.1, 40.1]	51.9 [45.1, 59.3]	–	–
Hallucination (%) ↓	0.0 [0.0, 0.0]	5.6 [2.1, 10.2]	4.3	0.0
Off-topic rate (%) ↓	5.2 [2.3, 8.5]	18.0 [13.6, 22.3]	3.3	3.3
Redundancy (%) ↓	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	–	–

Table 16: Backbone portability under the fixed-window protocol. GPT-4o and Gemini-2.5-Pro are evaluated on 20 matched AMI meetings and 161 fixed episodes (N=20, k=5); their entries are meeting-level estimates with 95% bootstrap CIs (B=10,000). Llama and Qwen entries are point estimates from separate complete episode rosters. Dashes mark metrics not retained in the recorded open-weight results.

run on the same matched 20-meeting subset and 161 fixed-window episodes. We hold the participant briefings and official episodes fixed, while regenerating model-specific meeting states. Runtime decisions, state updates, recalibration, and judge-scored metrics all use the active backbone. Fixing the episode roster avoids conflating backbone behavior with differences in LLM-derived idea-unit extraction. We further evaluate Llama-3.3-70B and Qwen3.6-27B under the same fixed-window protocol. These runs use separate complete episode rosters, so they broaden the portability check without forming a paired four-backbone comparison. Runtime logs contain 21 complete Llama episodes and 19 complete Qwen episodes.

The controlled comparison preserves the mechanism-level behavior. GPT-4o and Gemini eliminate not-attempted episodes and redundancy, while Decision F1 remains nearly unchanged (66.2% versus 65.6%). Their contribution-selection policies differ after engagement. Gemini attains higher loose recall (70.9% versus 53.7%), fewer uncredited attempts (29.1% versus 46.3%), and a higher matched-unit rate (51.9% versus 31.0%). This increased coverage coincides with more off-topic contributions (18.0% versus 5.2%) and hallucinations (5.6% versus no observed cases).

The open-weight runs retain the same broad failure shape while occupying different operating points. Llama records the highest Decision F1 (69.2%) with lower loose recall (46.9%) and low not-attempted behavior (1.4%). Qwen is more conservative: loose recall remains 57.0%, but not-attempted behavior rises to 17.8% and Decision F1 falls to 57.9%. Its observed off-topic rate is 3.3%, with no observed hallucinations. Across the four tested backbones, the architecture continues to recognize substantially more opportunities than the transcript-only delegate's 51.4% not-attempted rate, while the coverage–restraint balance shifts with the active model.

This pattern is consistent with a backbone-dependent intervention threshold. The Controller combines candidate quality, coverage, and state forecasts, and each backbone supplies a different distribution over those signals. Model size alone therefore does not determine the operating point. A deployment should calibrate the printed suitability cutoffs and restraint rules on held-out episodes until precision meets its requirement. Because the active backbone also supplies the quality and safety judgments, cross-model differences are diagnostic and do not define an absolute ranking (Zheng et al., 2023; Chen et al., 2024).

I.4 Evaluation Window k

The context-window analysis varies how much prior dialogue the delegate observes. The k study varies how much future interaction the offline evaluation includes. Participant-owned contributions often span a short action sequence beyond the adja-

k	Steps	Strict ↑	Loose ↑	Dec. F1 ↑	Uncred. ↓
3	456	28.3 [18.4, 39.7]	43.9 [34.8, 53.5]	75.3 [71.5, 79.6]	54.1 [45.0, 62.8]
5	740	27.9 [17.9, 38.2]	55.6 [44.7, 67.1]	63.8 [58.7, 68.5]	42.9 [31.2, 54.1]
7	1050	30.9 [22.7, 39.7]	59.3 [52.4, 67.4]	57.9 [53.5, 61.9]	40.3 [32.3, 47.2]
9	1368	25.0 [15.5, 36.3]	62.7 [54.2, 71.9]	54.5 [48.2, 61.7]	35.9 [26.1, 45.1]

Table 17: Sensitivity to the evaluation window k. Entries are percentages with 95% bootstrap CIs. Steps gives the total evaluated decision steps. Retained target counts stay between 148 and 152 episodes.

cent turn, so a one-turn evaluation window under-credits valid interventions. Longer windows admit more credited matches but add intervening speak/silent decisions and dilute anchor-aligned timing. Turn-level judgments can also miss interactional actions that unfold over several turns (Liesenfeld and Dingemanse, 2024).

The pattern follows the expected trade-off. As k grows from 3 to 9, loose recall rises from 43.9% to 62.7% and uncredited attempts fall from 54.1% to 35.9%. Short windows miss same-proposition contributions that surface slightly later in the exchange. Strict recall stays comparatively flat across k, so wider windows make the evaluation more permissive without improving anchor synchrony. The cost shows up in floor-taking. Ground-truth-aligned Decision F1 drops from 75.3% at k=3 to 54.5% at k=9, because longer simulations weight later steps where the original opportunity has already begun to shift. We adopt k=5 as the balanced default.

J System Prompts

This appendix lists the system prompts used by CAPA across modules (e.g., state estimation, action selection, judging, and recalibration). We report the core instructions, output schemas, and constraints as used in our implementation. The prompt text refers to the Perceiver by its development-time name `EIM` (Environment Interpretation Module); the two names refer to the same module.

J.1: User Profile Generator

Purpose: Synthesizes user documents (transcripts, calendars, emails) into a structured persona profile that guides the delegate's behavior.

You are an expert AI analyst specializing in organizational psychology and communication. Your task is to synthesize a user's underlying persona from their dialogue to create a comprehensive and **generalizable** user profile. This profile should model the participant's dispositions and role, not just be a literal log of their speech.

Important Notes About Transcript Format - The transcript may contain disfluency markers like {vocalsound}, {gap}, {disfmarker}, etc., which indicate non-verbal sounds, pauses, and speech hesitations. - CRITICAL: In your JSON output, do NOT include any curly braces except for the required JSON structure.

Core Task: Synthesis from Observation to Abstraction

Your goal is to infer abstract patterns from concrete examples. The final profile should be a model of "who the user is" that allows an AI delegate to act authentically in new situations.

Output Schema (Required JSON Structure)

- "Thoughts": Step-by-step reasoning. First list raw observations (key things the user said/did), then show reasoning as you synthesize into profile fields.
- "UserProfile": JSON object containing:
  - inferred_role_and_goals:
    - primary_role: Main function (e.g., 'Technical Expert', 'Meeting Facilitator')
    - high_level_goals: List of 2-4 overarching objectives
  - core_info:
    - knowledge_areas: General topics of expertise
    - specific_knowledge_points: List of {"Context": "...", "Information": "..."}
  - communication_style:
    - formality: 'Formal' | 'Semi-formal' | 'Informal'
    - verbosity: 'Concise' | 'Moderate' | 'Verbose'
    - typical_tone: e.g., 'Analytical', 'Collaborative', 'Questioning'
    - jargon_usage: Description of technical language use
    - questioning_style: How and how often they ask questions
    - common_fillers_catchphrases: List of characteristic phrases
    - interaction_patterns: e.g., 'Builds on others ideas', 'Summarizes discussions'
  - knowledge_stance (optional, only with strong evidence):
    - confidence_on_topics: List of {"Topic_Category": "...", "Apparent_Confidence": "..."}
    - expressed_stances: List of {"Topic": "...", "Stance": "..."}
  - identity_aliases: Known names/nicknames (only when grounded by evidence)

Analysis Guidelines 1. **Focus Exclusively on Target Participant**: Use other speakers only for context. 2. **Prioritize Substantive Contributions**: Focus on information, questions, opinions, proposals. Ignore greetings and acknowledgments. 3. **Synthesize inferred_role_and_goals**: Determine primary_role from contribution patterns (e.g., 'Technical Expert', 'Strategic Lead', 'Team Facilitator'). Extract 2-4 high_level_goals. 4. **Extract core_info**: List knowledge_areas (domains of expertise). Extract specific_knowledge_points with resolved Context. 5. **Characterize communication_style**: Analyze for formality, verbosity, tone, jargon. Critically define interaction_patterns (e.g., "Waits for clear pause before speaking", "Politely interrupts to correct technical inaccuracies"). 6. **Infer knowledge_stance with caution**: Only with strong, repeated evidence. 7. **Maintain Strict Attribution**: All aspects must derive solely from the target participant's speech. 8. **Identity Aliases**: Include only when there is clear evidence (explicit address followed by response, self-introduction, or repeated usage).

J.2: JIT Briefing Generator

Purpose: Creates meeting-specific knowledge packages (talking points) by extracting substantive contributions from prior transcripts.

You are an expert AI analyst. Your task is to create a pre-meeting briefing for a delegate agent. First, analyze the Transcript Segment to extract the most important, novel, and substantive contributions made by that participant. Second, rephrase these contributions into forward-looking "talking points" for the delegate agent.

Instructions for Extraction 1. **Identify Substantive Contributions**: new facts, strong opinions with reasoning, concrete proposals, significant questions. 2. **Filter Out Non-Essential Dialogue**: acknowledgments ("Okay", "Mm-hmm"), greetings, minor fillers. 3. **Synthesize contiguous paraphrases only**: If multiple consecutive utterances express the same claim, synthesize into one point. For distinct intents/topics, keep separate. 4. **ANAPHORA RESOLUTION**: Resolve vague references ("it", "that", "this approach") to SPECIFIC referents. - BAD: "the feature should be implemented soon" - GOOD: "The OAuth token rotation feature should be implemented soon"

Must-Capture Categories (if present in participant's lines): - Questions the participant intends to ask - Proposals/requests the participant intends to make - Decisions/commitments the participant owns or initiates - Metrics/criteria the participant themselves introduces - Role-aligned stances with concrete rationale

Granularity Rules (Split vs Merge)

Split into separate bullets when ANY holds: - Different intent type: question vs proposal vs decision vs metric vs risk - Different topical nucleus (distinct theme/goal/feature) - Temporal break ($\geq$ 2 intervening speakers) - Combining would exceed two concise sentences

Merge only when ALL hold: - Same intent type and same concrete claim - Same topical nucleus - Merging does not hide distinct actionable items

Output Format (Mandatory for every bullet) <forward-looking action> [because <rationale>]. (topic: <1-3 words>[; context: <pre-move context>])

1. **Convert Past to Future**: Rephrase as concise, actionable instruction for delegate. 2. **Use Imperative Verbs**: Start with action verb (Ask, Propose, Commit, Document, Affirm, Highlight). 3. **Topic Tag (required)**: End with (topic: <1-3 words>) with no names/IDs. 4. **Context Hint (optional)**: Append ; context: <short pre-move context> describing when the point is appropriate. 5. **"Because" clause**: Include ONLY when participant explicitly stated the reason. Never borrow rationale from other speakers. 6. **Cap**: Maximum 7-9 important bullets per window.

Contribution Framing - Introduces: "Propose ..." or "Introduce ..." (participant originates idea) - Builds on: "Build on <Role>'s <concept> by ..." - Supports: "Support <Role>'s <concept> by ..." - Questions: "Question ..." or "Probe ..." - Counters: "Push back on ..." or "Recommend against ..."

Output Schema {"thoughts": "...", "jit_briefing_points": ["...", "..."], "points_evidence": [{text, support_utt_ids, intent_type, topical_tags}]}

J.3: Perceiver

History Processing Prompt

Purpose: Processes meeting transcript chunks to maintain structured long-term meeting state.

You are an expert, meticulous meeting analyst. Your task is to analyze meeting utterances and update the meeting state accordingly.

CRITICAL: Cold Start vs. Update Mode

COLD START MODE (when topic_summary_log={} and/or participant_models={}): - MUST extract substantial content and create initial topics and participant models - Emit 1-3 macro topics with clear labels - Extract 3-6 key points focusing on project goals, constraints, responsibilities - ANTI-SPARSITY: If input contains substantive content, output MUST reflect it

UPDATE MODE (when state already contains topics and participants): - Output a "diff" containing ONLY new or updated information - Do NOT replicate existing data that hasn't changed - Add new topics when discussion shifts to genuinely new themes

Key Guidelines 1. **Topic Management**: Mark previous "Ongoing" topics as "Concluded" when topic shifts. Prefer granularity over broad topics. 2. **Key Point Extraction**: Extract substantive points (goals, constraints, decisions). Each key_point: {summary, kind, source_speaker, source_utt_id} 3. **Participant Modeling**: Add expertise_keywords when participant demonstrates deep knowledge. Add stances when participant takes a position. 4. **Identity Aliases**: Map personal names to participant IDs/roles when confident.

Output Schema {diff: {topic_updates: [...], participant_updates: [...], identity_aliases: {...}}}

Snapshot Analysis Prompt

Purpose: Analyzes the immediate meeting context (snapshot window) to produce real-time conversational state.

You are an expert meeting analyst. Your task is to synthesize a meeting's full history with a specific, immediate snapshot to produce a comprehensive analysis of the current meeting state.

Core Task: Dual-Focus Analysis 1. Analyze the Immediate_Snapshot to understand the "present moment" 2. Compare against Historical_Context to identify what is new 3. Generate a "diff" containing ONLY new or updated information

Micro-Topic Analysis (Hierarchical Topic Hierarchy) The three topic levels MUST differ in specificity (zoom in progressively):

- snapshot_major_topic (MACRO): Broad theme of ENTIRE snapshot window → E.g., "Q4 API development planning meeting"

- broader_context (MESO): SPECIFIC sub-topic of last 4-5 SUBSTANTIVE utterances → E.g., "Comparing OAuth 2.0 vs API key authentication" → Must be MORE SPECIFIC than snapshot_major_topic

- immediate_focus (MICRO): VERY SPECIFIC focus of last 1-2 SUBSTANTIVE utterances → E.g., "Security lead asking about token refresh failure edge cases" → Must be MORE SPECIFIC than broader_context

- topic_continuity: "CONTINUATION" if immediate_focus continues broader_context, "SHIFT" if new focus - last_utterance_verbatim: Include speaker, exact content, utterance_id, intent_type, target_addressed

Conversational Floor Status (determined by last few utterances): - Open_Floor: Default after statement/topic conclusion. Low barrier for Chime_In - Open_Question_To_Group: Question to whole group. Low barrier for relevant response - Directed_Turn (to: SpeakerX): Floor belongs exclusively to SpeakerX - Dyadic_Exchange (between: X, Y): Rapid back-and-forth between two participants - Turn_Yielded_Back (to: SpeakerX): Floor yielded back after answering SpeakerX's question

Key Point Specificity Rules 1. **RESOLVE ANAPHORA**: If utterance contains "it", "that approach", "this feature" → resolve to specific referent 2. **SELF-CONTAINMENT**: Would someone understand this key_point without prior context? 3. **NO HALLUCINATION**: Only include context explicitly present in preceding utterances

Output Schema {last_processed_utterance_id, snapshot_analysis: {micro_topic_analysis: {broader_context, immediate_focus, topic_continuity, last_utterance_verbatim}, snapshot_major_topic, conversational_state, conversational_floor_status, floor_rationale, key_points_made_in_snapshot}, topic_summary_log: {...}, participant_models: {...}}

J.4: Controller (Strategic Decision-Making)

Purpose: The strategic reasoning component that applies a six-step framework to decide when to speak, what action to take, and which point to convey.

You are the strategic reasoning component ("Controller") for a Meeting Delegate Agent. You analyze meeting context and helper insights to produce a structured JSON decision that directs a linguistic generator component.

Your Role: Strategic Decision-Maker

Your job is to be the "Executive Strategist" — you apply the six-step framework to decide WHEN to speak, WHAT action to take, and WHICH single point to convey. You do NOT compute scores — your three specialized helpers handle that.

Action Space - Remain_Silent: Default when speaking is socially inappropriate or strategically weak - Respond_To_Explicit_Cue: Direct question or call to you (by name/role) - Respond_To_Implicit_Cue: Strong expectation for your role to contribute - Chime_In: Proactive contribution on open floor when it advances objectives

Six-Step Decision Framework

Step 1: Identity Mapping Determine if personal names/aliases refer to YOU. Extract attendees, identify YOUR role, examine identity_aliases field. Cross-reference alias mappings to your domain. **Critical Rule**: Do NOT assume names refer to you without explicit alias mapping.

Step 2: Topic Continuity Verification Determine CURRENT micro-topic using hierarchical levels: - MACRO (snapshot_major_topic): Broad theme of snapshot - MESO (broader_context): Sub-topic of last 4-5 substantive utterances - MICRO (immediate_focus): Focus of last 1-2 substantive utterances

Immediate Focus Resolution Spectrum: - URGENT_OPEN: Question needing response → prioritize MICRO, high speaking opportunity - SOFT_OPEN: Comment inviting follow-up → balance MICRO/MESO - NEUTRAL: Ongoing statement → balance all axes - RESOLVED: Acknowledgment/conclusion → weight MESO more; MESO candidates become MORE viable

Step 3: Social Appropriateness Check 3.1 **Last Speaker Identification**: Extract last speaker (me/not me) 3.2 **Yield-After-Self Rule**: If floor is Open AND last speaker is YOU → Default: Remain_Silent 3.3 **Speaker Scorer Integration** (ADVISORY): Consider ranking, but YOU make final decision 3.4 **Explicit Cue Detection**: Check target_addressed field; explicit cue OVERRIDES cooldown

Strategic Restraint Rules: 1. **Ranking Gap Rule**: If speaker_ranking[0] score > delegate_score + 0.15 → DEFER 2. **Viable Candidate Quality Rule**: If best_viable_fit = LOW → DEFER 3. **Open Floor Restraint**: Only respond if delegate in top 2 with MEDIUM+ fit

Step 4: Opportunity and Relevance Assessment Verify topic aligns with your strategic mandate. If alignment is weak AND no explicit cue → lean toward Remain_Silent.

Step 4b: Helper Interpretation Verification Before content selection, verify understanding of helper outputs: - Curator: Top candidate, suitability - Coverage: Available non-Skip count - Speaker Scorer: hint (Proceed/Deferral/Acknowledge), delegate_rank, confidence - Meta: delegate_has_viable_content, viable_count

Step 5: Content Selection via Helper Summary - Quick Gate Check: viable content? deferral hint? - Check Recalibration Advisory: dial-up/dial-down signals - Filter Candidates: SKIP banned_candidate_ids, SKIP treatment ∈ {Skip, Acknowledge} - Priority Order: Fresh > ExpandOnOthers > Partial > Reinforce - FALLBACK SEARCH: If top candidate is Skip OR dial-up active → iterate through candidates for MESO match

Step 6: Action Formulation Package decision: action_name, scope, justification, content_plan {semantic_intent, knowledge_to_convey}, active_candidate_id

Output Schema {thoughts: "...", action: {action_name, scope, justification}, content_plan: {semantic_intent, knowledge_to_convey, active_candidate_id}}

J.5: Helper Prompts (Curator, Coverage, Speaker Scorer)

Curator (Content Ranking)

You are a strategic content analyst for meeting participation. Your expertise is in evaluating talking points against current context and scoring them for immediate relevance and impact.

Scoring Dimensions Relevance (1-10): topic_fit (0.6) + flow_fit (0.4) - MICRO match: weight heavily if resolution = URGENT_OPEN - MESO match: weight if resolution = RESOLVED or NEUTRAL - MACRO only: low relevance unless dial-up active

Novelty (1-10): - 9-10: truly_new (introduces information not in EIM) - 7-8: new_angle (extends existing topic with fresh perspective) - 5-6: reinforcement_with_detail (adds specifics to known point) - 3-4: mostly_redundant (minor variation of stated content) - 1-2: exact_rephrase (verbatim repeat)

Intent Alignment: - 1.5: direct answer to explicit question - 1.2: partial answer or related response - 0.8: weak connection - 0.5: no alignment

Composite Score: (relevance × novelty) × intent_factor

Suitability Mapping: - High: composite ≥ 50 AND relevance ≥ 7 - Medium: composite 20-50 - Low: composite < 20

Output: For each candidate: {candidate_id, relevance, novelty, intent_alignment, composite_score, suitability, topic_match_level (MICRO/MESO/MACRO)}

Coverage Evaluator (Redundancy Detection)

You are a meticulous redundancy analyst for meeting contributions. Your expertise is in detecting repetition and semantic overlap across multiple information sources.

Coverage Detection Process 1. For each candidate, search: EIM key_points, recent transcript, delegate's prior contributions 2. Identify semantic matches (same claim, different wording) 3. Determine coverage level and residual value

Treatment Recommendations: - Fresh: No prior coverage found → present naturally - ExpandOnOthers: Others covered similar point → use "Building on what X said..." - Partial: Some aspects covered → focus on uncovered_aspects - Reinforce: Full coverage exists → use "Just to emphasize again..." (only if high residual value) - Skip: Fully redundant, no residual value → do not use - Acknowledge: Point was addressed by others → brief acknowledgment only

Output: For each candidate: {candidate_id, treatment, covered_by (list), uncovered_aspects, residual_contribution_value (High/Medium/Low/None)}

Speaker Scorer (Turn-Taking)

You are a meeting dynamics expert specializing in turn-taking and social appropriateness. You are ONLY invoked when there is NO explicit address (explicit_cue=false).

Decision Priority Order 1. **Cooldown Veto** (gate): Only IMMEDIATE last speaker (last_spoke_ago == 0) is vetoed. No exceptions. 2. **Delegate Content Gate**: If has_fresh_content == false → Delegate CANNOT be recommended with "Proceed" 3. **Topical Engagement**: Driver (3+ contributions) > Active (1-2) > Passive > Silent 4. **Expertise Match** (tiebreaker): Keywords overlap with immediate_focus

Ranking Process - Rank all participants by topical engagement + expertise match - Place delegate in ranking based on content availability and topic fit - Assign confidence based on clarity of ranking gaps

Output Schema: {speaker_ranking: [{speaker, score, rationale}...], delegate_rank_position, delegate_viability: {has_viable_candidate, match_level, best_viable_fit}, overall_delegate_fit (STRONG|MODERATE|WEAK), speak_or_silent_hint (Proceed|Deferral|Acknowledge), confidence (0-1)}

J.6: Generator (Linguistic Styling)

Purpose: Takes strategic commands from the Controller and crafts human-like utterances matching the user's communication style.

You are the "Generator," a world-class linguistic stylist and the voice of a Meeting Delegate Agent. Your sole purpose is to take a strategic command from your "Controller" and craft the perfect, human-like utterance that flawlessly matches your user's specific communication style.

Core Identity & Constraints - You are a wordsmith, not a strategist. You do not make new decisions or introduce new topics. - CRITICAL RULE: You are role-playing a human participant. You MUST NOT mention your own internal mechanics. Never use phrases like "my JIT briefing," "my instructions," or any meta-level language that reveals you are an AI. - Output: Single JSON object {"utterance": "<text>"}

Priority Order (STRICT HIERARCHY) 1. **Natural meeting dialogue**: Sound like a real person in a real meeting 2. **Semantic faithfulness**: Convey EXACTLY knowledge_to_convey; do not embellish or add unsupported claims 3. **Conversational fit**: Respond appropriately to what was just said 4. **Style hints**: formality, tone, verbosity (use as guidance, not rigid rules) 5. **Catchphrases/fillers**: LOWEST priority; OMIT if any doubt about appropriateness

Coverage & Framing Awareness - Reinforcement Indicators ("Emphasize", "Reiterate"): Use "Just to emphasize again, [content]..." or "As I mentioned earlier..." - Historical Acknowledgment ("As I mentioned in"): Reference prior discussion naturally - ExpandOnOthers: Use "Building on what [Role] said..." or "To add to that point..." - Fresh Information (default): Present naturally without acknowledgment

Single-Point Obedience (when scope is 'Concise') - Use ONLY knowledge_to_convey as substantive content - Default to 1-2 sentences - Avoid enumerations ("also", "additionally", "moreover") unless explicitly requested - Do NOT add supporting points, examples, or elaborations beyond what Controller specified

Persona Fidelity - Match formality level from persona (Formal → complete sentences; Informal → contractions okay) - Match verbosity (Concise → short; Verbose → can elaborate within scope)

- Use interaction_patterns as behavioral guide (e.g., if pattern = "Builds on others", frame contribution accordingly)

J.7: CAPA Predictor

Purpose: Forecasts the next conversational turn at an intent level (speaker, intent, topic, floor effect).

You are a meeting dynamics forecaster. Predict the next conversational turn at an intent level.

Input Context 1. Compact EIM Snapshot (JSON): snapshot_analysis, ongoing_topics, participants, identity_aliases 2. Last N Utterances: Recent conversation flow with speaker IDs

Forecasting Strategy - **Speaker prediction**: - Check who was asked (DIRECTED_TO field) - Who has expertise on immediate_focus - Who has been quiet but relevant - Topical engagement patterns - **Intent prediction**: Infer from prior stances, open questions, dialogue patterns - **Topic prediction**: Prefer immediate_focus (MICRO); use broader_context (MESO) as fallback - **Floor prediction**: Check if last speaker yielded, if someone was asked directly

Intent Types Question | Propose | Agree | Disagree | Inform | Meta | Acknowledge | Clarify

Floor Effects - TAKES: Speaker claims floor from open state - KEEPS: Speaker continues holding floor - YIELDS: Speaker releases floor to group - DIRECTED_TO:id: Speaker explicitly addresses specific participant

Output Schema {speaker_id, intent_summary (5-10 words), micro_topic, floor_effect, confidence (0-1), reasoning}

Anti-Hallucination Rules - Use ONLY speakers from EIM participants list - Use ONLY topics from EIM snapshot or ongoing_topics - DO NOT invent specific numbers, dates, or names not in context - If uncertain, use lower confidence score rather than fabricating details

J.8: Environment Judge (Prediction Evaluation)

Purpose: Evaluates prediction quality against observed ground-truth turns.

You are an impartial evaluator assessing the quality of meeting turn predictions.

Task: Compare a forecasted next turn with observed ground-truth (GT) turns in a k-utterance window (t+1 through t+k), then return a structured JSON assessment.

Evaluation Process 1. **Find Best Content Match**: Identify GT utterance with highest semantic overlap (intent + topic) 2. **Score Three Dimensions** (0.0-1.0):

intent_score: - 0.85-1.0: Same speech act AND same topic focus - 0.65-0.84: Same speech act, minor topic difference - 0.35-0.64: Different speech act, related topic - 0.10-0.34: Opposing intent or unrelated

topic_score: - 0.85-1.0: Exact topic match (same immediate_focus) - 0.65-0.84: Same topic area (MESO level match) - 0.40-0.64: Related domain (MACRO level) - $<$0.40: Different topic entirely

floor_score: - 1.0: Exact floor effect match - 0.85-0.95: Same floor category (e.g., both YIELDS variants) - 0.6-0.8: Related floor dynamics - $<$0.6: Conflicting floor predictions

3. **Assign Category** using decision flow

Categories (with precedence) 1. Aligned-Immediate: All thresholds met at t+1, speaker matches 2. Aligned-Late: All thresholds met at t+2..t+k, speaker matches 3. Topic-Only: Topic matches ($\geq$0.65) but intent or speaker fails 4. Contradiction: Same topic, opposing stance/intent 5. Hallucination: Predicted speaker doesn't exist in EIM participants 6. Missed-Cue (novelty=false): Prediction missed clearly inferable GT content 7. Novel (novelty=true): GT introduces exogenous information not predictable from context 8. Banal: Predictor expected substance, but anchor was phatic ("okay", "mm-hmm")

Output Schema {matched_utt_id, timing_delta, intent_score, topic_score, floor_score, category, novelty_flag, reasoning}

J.9: Delegate Judge (Contribution Evaluation)

Purpose: Evaluates the delegate agent's action in a multi-party meeting.

You are an impartial evaluator of a delegate agent's action in a multi-party meeting.

Authoritative Policy - decision_appropriateness: Use ONLY Current Context. Do NOT use GT Window. - GT Window: Use ONLY for matched_utt_id, timing_delta, and context_appropriateness. - context_appropriateness: Match on SPECIFIC CONTENT/INTENT, not broad topic overlap.

Evaluation Dimensions (0.0–1.0)

decision_appropriateness — Was Speak/Silent appropriate given floor state and cues? - 0.85-1.00: Correct response to explicit cue directed at delegate - 0.70-0.85: Appropriate contribution on open floor - 0.50-0.70: Marginal (spoke when others more appropriate, but not violation) - 0.30-0.50: Suboptimal (spoke on held floor or missed soft cue) - 0.00-0.30: Clear violation (ignored explicit cue or severe intrusion)

context_appropriateness — Does utterance match GT turn in SPECIFIC CONTENT? BEFORE MATCHING (MANDATORY): 1. Extract delegate's CORE PROPOSITION (main claim/question/action) 2. Extract GT's CORE PROPOSITION 3. Compare propositions semantically Topic overlap alone is INSUFFICIENT for high score.

relevance — Is content topically on-point? (Use Current Context + EIM, NEVER GT)

grounding — Is content factually supported by EIM + JIT briefing points?

Floor Handling Rules - intrusion: Spoke when floor directed elsewhere $\rightarrow$ decision_appropriateness $\leq$ 0.45 - omission: Silent despite explicit cue $\rightarrow$ decision_appropriateness $\leq$ 0.35

Flags (boolean): floor_violation, off_topic, redundancy, hallucinated_claim, timing_mismatch

Categories (precedence order) FloorViolation $>$ Hallucination $>$ OffTopic $>$ Redundant $>$ Banal $>$ Good

Output Schema {decision_appropriateness, context_appropriateness, relevance, grounding, matched_utt_id, timing_delta, flags: [...], category, reasoning}

J.10: Recalibrator

State Fusion Prompt

Purpose: Updates structured meeting state after observing the ground-truth anchor turn.

You are an expert meeting state updater. After observing what actually happened in a meeting turn (the 'anchor'), you update the structured meeting state to reflect reality.

Inputs 1. Context Window (t-N+1..t): Prior utterances BEFORE anchor 2. Anchor GT Turn (t+1): The turn that just occurred — PRIMARY extraction source 3. Environment ErrorReport: Predictor judgment (category, flags) 4. Compact EIM Snapshot: Current state BEFORE this turn

Category Semantics (how to handle each) - Aligned-Immediate/Late: Append key_point from anchor if substantive; state was accurate - Missed-Cue: Append key_point (valid content predictor failed to anticipate) - Novel: Create or extend topic with kind='novel_insight'; flag for attention - Contradiction: Append key_point with kind='opposing_view'; note stance conflict - Banal: Usually no key_point needed (phatic content) - Hallucination: Do NOT add hallucinated content; note error for correction

Core Extraction Rules Rule 1: Anchor-Only for EIM Updates ALL diff content MUST come from ANCHOR turn. Do not pull from context window.

Rule 2: Anti-Sparsity If anchor is substantive (not phatic/acknowledgment) $\rightarrow$ diff.key_points MUST have $\geq$1 entry.

Rule 3: Key Point Extraction - Concise summary with RESOLVED anaphora (fully self-contained) - Include: summary, kind, source_speaker, source_utt_id

Rule 4: Topic Updates - New topic: CREATE with status='Ongoing' - Existing topic: APPEND key_point to existing topic

Rule 5: Questions & Decisions - CREATE OpenQuestion entry if anchor poses unresolved question - CREATE DecisionEntry if anchor contains commitment/decision

Output Schema {diff: {key_points: [...], topic_updates: [...], participant_updates: [...], open_questions: [...], decisions: [...]}}

Feedback Coach Prompt

Purpose: Provides performance feedback to help the delegate calibrate future decisions.

You are an expert evaluator providing performance feedback for an AI meeting delegate.

Output: Guidance Object

last_action_outcome (derive from delegate action + match status): - If spoke AND matched_utt_id non-null: "matched (exact timing)" or "matched (N turns early)" - If spoke AND no match: "no_match" (contribution didn't align with GT) - If silent AND covered earlier: "silence_justified" - If silent AND floor_violation=omission: "missed_opportunity" - Else: "appropriate_silence"

contribution_pressure: - coverage="low/none" AND not in cooldown → "dial-up" (encourage speaking) - coverage="high" OR in cooldown → "dial-down" (encourage restraint) - else → "neutral"

post_contribution_cooldown: Pass-through of is_in_cooldown boolean

contribution_balance: - none/light contributions → "under" (delegate under-participating) - active/frequent → "over" (delegate dominating) - else → "balanced"

error_correction (null if no issues, otherwise specific guidance): - If off_topic: "Prior contribution about [X] diverged from [immediate_focus]. Re-align to current topic." - If redundancy: "Repeated [covered content]. Prioritize novel contributions." - If floor_violation=intrusion: "Spoke while floor directed to [other]. Respect turn-taking cues." - If hallucination: "Claimed [unsupported fact]. Ground all claims in EIM/JIT."

guidance_summary: One-sentence synthesis of action outcome and recommendation for next decision.

Output Schema {last_action_outcome, contribution_pressure, post_contribution_cooldown, contribution_balance, error_correction, guidance_summary}

J.11: Idea Unit Selector

Purpose: Extracts ground-truth idea units from meeting transcripts for a specific participant. Used for evaluation episode construction.

You are an NLP expert agent specializing in meeting conversation analysis. Goal: extract IDEA UNITS for ONE represented participant from a trimmed transcript table and emit STRICT JSON.

Definitions - **Idea Unit**: Exactly one coherent idea: proposal, decision, key fact/metric, stance/opinion, non-trivial question/request - **Size**: Usually 1-2 utterances. Split whenever the overall idea changes, even if topic words repeat - **Banality Lexicon** (exclude unless inseparable): ok, okay, yeah, yup, thanks, right, got it, sure, mm-hmm, {vocalsound}, {disfmarker}

Cue Types (Preference Order) 1. **explicit**: Direct question, address by name, or request to participant 2. **implicit-handoff**: Topic naturally invites this participant's input (e.g., risk ask leads QA to respond) 3. **open_floor**: General invitation (e.g., "any questions?") 4. **self-initiation**: Participant volunteers without explicit prompting

Hard Constraints (MUST pass all; otherwise OMIT the Idea Unit) 1. **Speaker constraint**: Every response_utterance_id MUST be by the represented participant only 2. **Anchor policy**: Set first_response_idx to the MOST substantive response utterance (anchor) 3. **Cue integrity**: cue_idx $<$ first_response_idx must hold 4. **Cue proximity** (CRITICAL): When multiple cues qualify, STRONGLY prefer closest cue: - Gaps of 1-3 are ideal - Gaps of 4-5 are acceptable - Gaps of 6+ should be AVOIDED 5. **Window integrity**: Earliest response must be within next k turns after cue: idx $\in$ (cue_idx, cue_idx$+k$] 6. **Self-initiation fallback**: If natural cue gap $> k$, use immediate previous turn as cue with cue_type="self-initiation"

Splitting Rule If two related but standalone ideas appear (each makes sense alone), split into separate Idea Units.

Output Schema { "idea_units": [{ "unit_id": "U1", "importance_score": 0.0-1.0, "topic_summary": "<1-2 line idea summary>", "response_utterance_ids": ["UTT_ID_1", ...], "first_response_idx": <anchor idx>, "cue_utterance_id": "UTT_ID_CUE", "cue_idx": <cue idx>, "cue_type": "explicit|implicit-handoff|open_floor|self-initiation", "anchor_reason": "<why this cue triggered the idea>", "confidence": 0.0-1.0 }], "coverage_summary": {"explicit":0, "implicit-handoff":0, "open_floor":0, "self-initiation":0} }