ModylBench

A Multimodal Benchmark for Evaluating AI Agents as Meeting Participants

Aleatoric Research[1]

February 2026

WORKING PAPER

Typeset in EB Garamond & Helvetica Neue

[1]Correspondence: hello@aleatoric.to. Dataset and evaluation code available at https://huggingface.co/datasets/use-aleatoric/modylbench.

Abstract

AI agents that participate in professional meetings—producing live analysis, generating interactive deliverables, and collaborating through voice and visual interfaces—represent a rapidly growing category of deployed systems. Yet no benchmark exists to evaluate their work product quality and collaboration effectiveness. Coding agents have SWE-bench, web agents have WebArena, but meeting agents lack evaluation frameworks that assess deliverable quality, iterative collaboration, and gaming robustness. We introduce ModylBench, the first benchmark specifically designed to evaluate AI agents as work-product-generating meeting participants, combining multimodal evaluation, professional deliverable assessment, and anti-gaming defenses. ModylBench comprises 48 scripted human turns (8 per scenario, with 48–52 total conversation turns including agent responses) across 6 professional verticals (financial analysis, deep research, business strategy, optimization, biostatistics, and business analytics), evaluated on 11 quality dimensions spanning both conversational process (the *journey*) and tangible deliverables (the *destination*). The benchmark features a multimodal evaluation protocol that sends raw agent audio and screen captures directly to judge models—bypassing speech-to-text intermediaries—and a six-layer anti-gaming framework addressing reward-hacking vulnerabilities identified through adversarial cross-validation. Version 3.0 of the framework generalizes evaluation to multi-participant meetings, multi-channel interaction (annotations, reactions, chat), prosodic naturalness, and video stream analysis, expanding the scoring architecture from 6 to 9 turn-level dimensions and introducing a meeting dynamics evaluation layer with 10 corpus-calibrated metrics. We describe the scenario design, scoring framework, multimodal judging protocol, and reliability testing methodology, and present the evaluation protocol for community submissions. ModylBench and its evaluation harness are released as open-source software with an accompanying HuggingFace dataset.

1 Introduction

The landscape of AI agent deployment is undergoing a fundamental shift. While the majority of research attention has focused on text-based agents that write code [9], browse the web [21], or execute tool calls in simulated environments [12], a parallel category of AI systems has emerged with markedly different requirements: agents that participate in live professional meetings.

Platforms such as LiveKit Agents [13], Pipecat [5], and a growing ecosystem of commercial products deploy AI agents as first-class meeting participants. These agents join video calls alongside human colleagues, listen to spoken instructions, produce real-time analysis through interactive visual surfaces (spreadsheets, charts, dashboards), and deliver tangible work products—all through voice and multimodal interfaces. The professional settings in which these agents operate are diverse: a private equity associate requesting a leveraged buyout model, a biostatistician analyzing clinical trial data, a supply chain manager optimizing distribution networks, or a strategy consultant evaluating market entry opportunities.

This deployment context creates evaluation challenges that existing benchmarks do not address. Unlike a code generation task with deterministic test suites, meeting agent quality is inherently multi-dimensional:

an agent must simultaneously demonstrate domain expertise (*correctness*), conversational competence (*social quality*), iterative responsiveness (*adaptability*), and the ability to produce professional-grade deliverables (*work product quality*). An agent that produces a mathematically correct financial model but delivers it with robotic affect and fails to acknowledge the user's corrections is not a good meeting participant. Conversely, an agent that is warm, responsive, and beautifully formatted but produces incorrect numbers is worse.

Moreover, evaluation must be *multimodal*. Meeting agents communicate through voice (tone, pacing, confidence), visual interfaces (rendered charts, interactive dashboards), and structured data (A2UI surfaces [1]). A text-only evaluation pipeline—transcribing the agent's audio and passing the transcript to a judge—discards the very signals that distinguish a competent meeting participant from a chatbot reading answers aloud.

Finally, evaluation must be *resistant to gaming*. The LLM-as-judge paradigm [19] is known to be susceptible to reward hacking: agents can be optimized for verbose, confident-sounding responses that score well on surface metrics while being substantively incorrect. In a meeting context, the "polite but wrong" failure mode is particularly dangerous, as an agent that agrees enthusiastically with everything and presents beautiful (but inaccurate) deliverables can score well on naïve evaluation rubrics.

Contributions. We make the following contributions:

1. **The first meeting agent work product benchmark.** ModylBench defines 6 professional-vertical scenarios with 48 scripted human turns (8 per scenario, 48–52 total conversation turns each), 18 expected work products, and 29 adversarial edge cases designed to test hallucination resistance, mathematical sanity, and professional judgment. Concurrent work MeetBench-XL [7] evaluates QA-oriented meeting assistance; ModylBench focuses on work product generation, mutation trajectory evaluation, and anti-gaming defenses.
2. **A multimodal evaluation protocol.** Our judge pipeline sends raw audio (WAV) and screen captures (PNG) directly to multimodal judge models, preserving prosodic cues and visual-audio coherence that text-only evaluation discards.
3. **A six-layer anti-gaming framework.** Developed through adversarial cross-validation with three independent reviewer models, the framework addresses specific reward-hacking vulnerabilities including weighted dimension scoring, hard floor gates, disagreement detection, programmatic verification, edge case injection, and pass@k reliability testing.
4. **An open evaluation harness and dataset.** The complete benchmark—scenarios, scoring pipeline, judge rubrics, and analysis tools—is released as open-source Python code with an accompanying HuggingFace dataset for community benchmarking.
5. **A generalized multi-party, multi-channel, multi-modal evaluation framework (v3.0).** We extend the benchmark beyond dyadic meetings to N-participant scenarios with 10 corpus-calibrated metrics for coordination quality, floor management, and prosodic naturalness; multi-channel scoring across annotations, reactions, and chat; video stream evaluation with keyframe extraction, WCAG accessibility checks, and audio-visual synchronization scoring; and a format-adapter architecture that extends mutation tracking to non-CRDT work products such as code files.

2 Related Work

Agent benchmarks. The evaluation of autonomous AI agents has progressed rapidly. SWE-bench [9] evaluates software engineering agents on real GitHub issues, using deterministic test suites as ground truth. τ-bench [18] introduces the pass@k reliability pattern for tool-agent-user interaction, measuring not just whether an agent *can* succeed but whether it *reliably* succeeds across multiple runs. WebArena [21] benchmarks web navigation in realistic environments. AgentBench [12] and TheAgentCompany [17] evaluate agents across diverse task categories including code, web, database, and office tool manipulation. GAIA [14] tests general assistants on questions requiring multi-step reasoning and tool use. None of these benchmarks evaluate *meeting participation*—the combination of real-time conversational interaction, multimodal communication, and professional work product generation.

Work product evaluation. Evaluating the quality of generated artifacts—beyond binary pass/fail—remains challenging. GDPval [16] benchmarks AI model deliverables against the work of industry professionals (averaging 14 years of experience) across 44 occupations spanning the top 9 GDP-contributing sectors, finding that frontier models are approaching expert quality on economically valuable tasks. ModylBench draws inspiration from GDPval's expert-comparison methodology and introduces a tier system (Peer, Mentor, Consultant) for meeting agent work products, extending the evaluation to multi-dimensional assessment across both the conversational process and the resulting deliverables.

Meeting corpora and evaluation. The AMI Meeting Corpus [2] and ICSI Meeting Corpus [8] provide annotated recordings of multi-party meetings with rich metadata including dialogue acts, topic segmentation, and extractive summaries. QMSum [20] and MeetingBank [6] benchmark meeting summarization. These resources evaluate *understanding* of meeting content (summarization, question answering) but do not evaluate an agent's *participation* in a meeting—its ability to produce work products, respond to iterative requests, handle edge cases, and communicate professionally through voice and visual interfaces.

LLM-as-judge. Using language models as evaluators has become standard practice. AlpacaEval [11] uses GPT-4 as an automatic evaluator. MT-Bench and Chatbot Arena [4, 19] combine model-based judging with human preferences. ModylBench extends the LLM-as-judge paradigm in two directions: (1) multimodal judging, where judges receive raw audio and visual inputs rather than text transcripts, and (2) anti-gaming defenses including disagreement detection between judges and pessimistic scoring on high-variance dimensions.

Concurrent and related meeting evaluation. MeetBench-XL [7] is a concurrent benchmark evaluating AI assistants in real-time enterprise meetings, using a bilingual multimodal corpus of 231 meetings (140 hours) with metrics for factual fidelity, intent alignment, and response efficiency. MeetBench-XL focuses on QA-style assistance—answering questions and retrieving information during meetings—whereas ModylBench evaluates agents as *work-product-generating participants* who produce tangible deliverables (spreadsheets, charts, reports) through iterative collaboration. Table 1 summarizes the landscape. No existing benchmark combines multimodal evaluation (audio + visual), professional work product assessment with mutation trajectory tracking, multi-turn conversational scoring, and anti-gaming measures. ModylBench fills this gap.

Table 1. Comparison of agent benchmarks. MODYLBENCH is the first to evaluate meeting agent work product quality with multimodal judging and anti-gaming defenses. Modalities: T = text, A = audio, V = visual.

Benchmark	Domain	Modality	Work Product	Multi-turn	Anti-gaming	Eval Method
SWE-bench	Code	T	✓	–	–	Test suite
WebArena	Web	T+V	–	✓	–	Task completion
τ-bench	Tools	T	–	✓	✓	pass@k
AgentBench	Mixed	T	✓	✓	–	Metrics
TheAgentCompany	Office	T	✓	✓	–	Task completion
AlpacaEval	Chat	T	–	–	–	LLM-as-judge
MT-Bench	Chat	T	–	✓	–	LLM-as-judge
MeetBench-XL	Meetings	T+A+V	–	✓	–	Multi-dim
MODYLBENCH	**Meetings**	**T+A+V**	✓	✓	✓	**Multi-judge**

3 MODYLBENCH

3.1 Design Principles

MODYLBENCH is built on three design principles that emerged from an adversarial cross-validation process in which three independent reviewer models (Claude Opus 4.6, GPT-5.2, Gemini 3.1 Pro) audited the evaluation framework and identified structural vulnerabilities.

1. **Substance over style.** The primary value proposition of a meeting agent is correct, actionable work product—not polished conversation. All scoring mechanisms weight substantive quality (accuracy, progress, correctness) above presentation metrics (formatting, social polish). An agent cannot achieve a passing score through style alone.

2. **Multimodal evaluation.** Meeting communication is inherently multimodal: an agent speaks while simultaneously rendering visual surfaces. Evaluation must capture audio-visual coherence, not just textual content. Our multimodal judge protocol sends raw WAV audio and PNG screenshots directly to judge models, bypassing speech-to-text pipelines that discard prosodic information.

3. **Anti-gaming by design.** Every scoring mechanism is designed with specific adversarial attacks in mind. We identify five concrete reward-hacking vulnerabilities (the "polite but wrong" hack, the "style over substance" hack, the "template replay" hack, the "mean anchor" exploit, and the "output mismatch" exploit) and implement targeted defenses against each.

3.2 Scenario Design

MODYLBENCH comprises six scenarios spanning distinct professional verticals, each representing a realistic meeting between a domain expert (the human) and an AI meeting agent. Every scenario follows a four-phase structure:

Context Phase (1 turn): The human introduces themselves, their role, and the problem to be solved. The agent must demonstrate understanding and establish the analytical framework.

Table 2. The six MODYLBENCH scenarios. Each scenario includes scripted human turns, expected deliverables, edge cases, and programmatic verification criteria.

Vertical	Scenario Title	Human Persona	Outputs	Key Edge Case
Financial Analyst	CloudSync LBO Model	PE Associate	3	Debt exceeds enterprise value
Deep Researcher	Solid-State Battery Intelligence Briefing	Technology VP	3	Hallucination trap (fake company)
Business Strategist	SEA Telehealth Market Entry Strategy	Strategy Director	3	Contradictory data sources
Optimization Solver	Q4 Supply Chain Distribution Optimization	Operations VP	3	Infeasible constraint set
Scientist	Phase II Hypertension Trial Statistical Analysis	Lead Biostatistician	3	Data leakage in imputation
Business Analyst	Q3 Pipeline Conversion Rate Diagnosis	RevOps Director	3	Survivorship bias in funnel

Work Phase (4–5 turns): Multi-turn dialogue driving toward tangible deliverables. The human provides data inputs, requests specific analyses, and iterates on preliminary results. The agent must produce structured outputs (spreadsheets, charts, documents) through Agent-to-User Interface (A2UI) surfaces [1].

Edge Case Phase (1 turn): A deliberate curveball designed to test the agent's professional judgment. Edge cases include infeasible requests, hallucination traps, circular dependencies, and data integrity violations. The agent must detect the problem and respond appropriately rather than blindly complying.

Delivery Phase (1 turn): The human requests a final, presentation-ready work product. The agent must compile, format, and deliver a polished artifact suitable for professional use.

The six verticals are summarized in Table 2. Each scenario defines 8 scripted human turns, 2–3 expected work product outputs, 4–5 adversarial edge cases, and programmatic verification criteria.

Example: Financial Analyst Scenario. The *CloudSync LBO Model* scenario simulates a private equity associate building a five-year leveraged buyout model. Over 50 turns (8 scripted human turns plus agent responses) reflecting realistic meeting length, the human provides base assumptions (LTM Revenue of \$50M, 30% EBITDA margin, 12× entry multiple), requests capital structure modeling (4.5× Senior Debt, 2.0× Subordinated Debt), income statement projections with decaying revenue growth, a debt schedule with mandatory amortization and cash flow sweeps, exit analysis with multiple contraction, a two-dimensional IRR sensitivity table, the incorporation of a 10% management option pool, and a final investment committee tear-sheet. Edge cases include a request to set Senior Debt at 15× EBITDA (exceeding the entry multiple, which the agent should flag as infeasible) and a circular-reference request to size debt based on cash flows that themselves depend on interest expense. Programmatic verification checks include: Year 1 Revenue = \$57.5M (\$50M $\times$ 1.15), Initial Enterprise Value = \$180M (12 $\times$ $\times$\$15M), and that the sensitivity table is a 5×5 grid with valid IRR percentages.

3.3 Scoring Framework

ModylBench evaluates agents along two complementary axes: the *journey* (how the agent interacts during the meeting) and the *destination* (what the agent ultimately delivers).

3.3.1 Journey Dimensions (Turn-Level Scoring)

Each agent turn is evaluated on six quality dimensions, scored on a 1–10 scale:

1. **Context Accuracy** ($w = 0.25$): Did the agent correctly understand what was asked? Were there hallucinated facts or misinterpretations?
2. **Task Progress** ($w = 0.25$): Does this turn measurably advance toward the deliverable?
3. **Iteration Quality** ($w = 0.20$): When the user requested changes, were they applied correctly and completely?
4. **Adaptability** ($w = 0.15$): How well did the agent handle corrections, curveballs, or changes in direction?
5. **Presentation Quality** ($w = 0.10$): Were visual surfaces rendered correctly? Was data formatted clearly?
6. **Social Quality** ($w = 0.05$): Appropriate tone, professional demeanor, empathy.

The weighted mean ensures that *substance dimensions*—context accuracy, task progress, and iteration quality—account for 70% of the turn score, while *style dimensions*—adaptability, presentation, and social quality—account for 30%. This weighting directly addresses the "polite but wrong" gaming vector (see Section 3.5).

The turn-level journey score is computed as:

$$S_{\text{turn}} = \sum_{d \in \mathcal{D}_{\text{turn}}} w_d \cdot s_d \tag{1}$$

where $\mathcal{D}_{\text{turn}}$ is the set of six turn dimensions, w_d is the weight for dimension d, and $s_d \in [1, 10]$ is the score. The overall journey score is the mean of turn-level scores across all turns in the scenario, subject to hard floor gates (see below).

3.3.2 Destination Dimensions (Work Product Scoring)

Each expected work product is evaluated on five quality dimensions:

1. **Correctness** ($w = 0.30$): Are the numbers, facts, and analysis correct? Are assertions verifiable?
2. **Completeness** ($w = 0.25$): Does it cover everything that was asked? Any missing sections?
3. **Actionability** ($w = 0.20$): Can the human use this deliverable as-is without rework?
4. **Professional Quality** ($w = 0.15$): Would a domain expert peer produce this quality?
5. **Format & Presentation** ($w = 0.10$): Clean layout, proper labels, visual quality.

As with the journey dimensions, correctness and completeness dominate (55% combined), ensuring that a polished but inaccurate deliverable cannot score above mediocre.

3.3.3 Combined Score and Tier Classification

The combined score weights destination above journey:

$$S_{\text{combined}} = 0.4 \cdot \bar{S}_{\text{journey}} + 0.6 \cdot \bar{S}_{\text{destination}} \tag{2}$$

The 60/40 split reflects the design principle that tangible work products matter more than conversational process. A meeting agent's ultimate value is the deliverable it produces; the conversation is the means, not the end.

The combined score maps to quality tiers inspired by GDPval's [16] methodology of comparing AI deliverables against human expert quality:

Tier	Threshold	Interpretation
Peer	≥ 6.0	Comparable to a competent colleague—gets the job done.
Mentor	≥ 7.5	Comparable to a senior expert—insightful, anticipatory.
Consultant	≥ 9.0	Top-tier advisory quality—polished and comprehensive.

The tier thresholds are currently set by expert judgment; empirical calibration against human expert ratings is planned as future work (see Section 7).

3.4 Multimodal Evaluation Protocol

A distinctive feature of ModylBench is its multimodal evaluation protocol. Meeting agents communicate through voice (synthesized speech via text-to-speech) and visual interfaces (A2UI surfaces rendered on the participant's screen). A text-only evaluation pipeline—transcribing the agent's audio via automatic speech recognition and then passing the transcript to a judge—discards critical quality signals:

- **Prosodic cues**: Tone, pacing, hesitation, and confidence conveyed through speech that indicate understanding or uncertainty.
- **Audio-visual coherence**: Whether the agent verbally references what it is showing on screen, and whether visual updates appear at appropriate moments.
- **Delivery quality**: Robotic versus natural speech delivery, awkward pauses, inappropriate emphasis.

Our multimodal evaluation pipeline operates as follows:

1. A `TurnAudioCollector` accumulates raw PCM audio frames from the agent's audio track during each turn, converting them to self-contained WAV files (24 kHz, mono, 16-bit) at turn boundaries.
2. A `ScreenCapture` module captures periodic PNG screenshots of rendered A2UI surfaces during each turn.
3. The raw WAV audio and PNG screenshots are passed directly to multimodal judge models—currently Gemini and GPT with native audio and image input capabilities—as inline data parts. No speech-to-text transcription occurs.

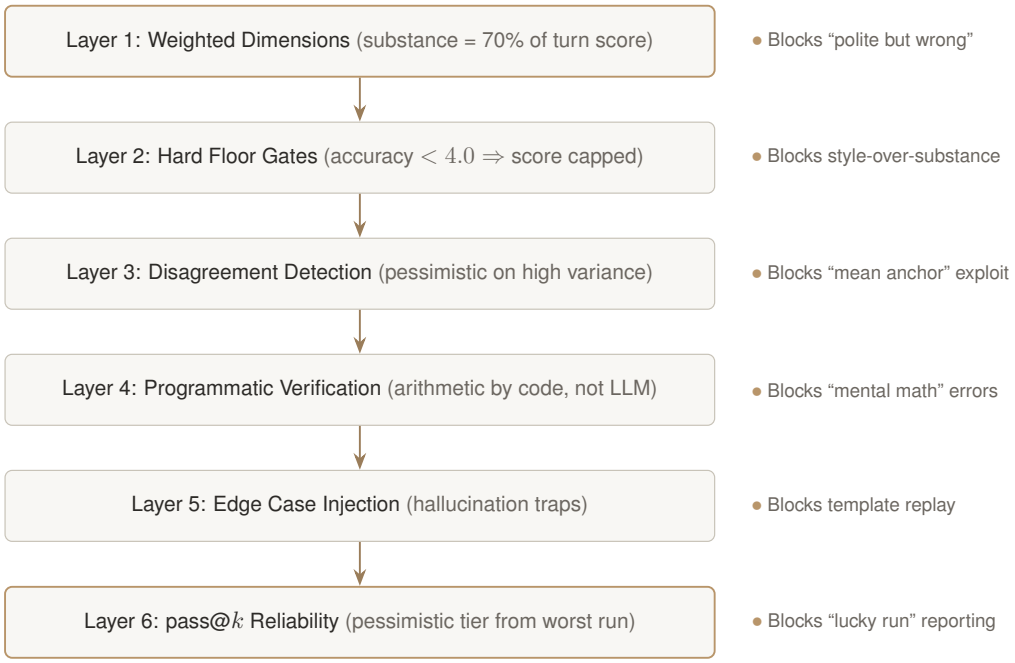

Figure 1. Six layers of anti-gaming defenses in ModylBench. Each layer addresses a specific reward-hacking vulnerability identified during adversarial cross-validation. An agent must survive all layers to achieve a meaningful tier classification.

4. Judges receive a specialized multimodal rubric that instructs them to evaluate audio-visual coherence, prosodic quality, and the synchronization between spoken content and visual output.

For judge models that do not support native audio input, the framework falls back to text-based evaluation using ASR transcription. The scorecard records which evaluation mode was used for each turn, enabling analysis of the gap between multimodal and text-only evaluation.

3.5 Anti-Gaming Measures

The anti-gaming framework in ModylBench was developed through an adversarial cross-validation process. Three independent reviewer models audited the evaluation pipeline and identified five concrete reward-hacking vulnerabilities. For each vulnerability, we implemented a targeted defense. Figure 1 illustrates the six defense layers.

Layer 1: Weighted Dimensions. As described in Section 3.3, substance dimensions (context accuracy, task progress, iteration quality) carry 70% of the turn score weight. This prevents the "polite but wrong" attack, in which an agent maximizes social quality (9/10) and presentation (9/10) while delivering incorrect content (2/10). Under equal weighting, such an agent would score 5.5/10; under our weighting, it scores 4.10/10—and the hard floor gate (Section 3.5, Layer 2) further caps it to 4.0/10, well below the Peer threshold.

Layer 2: Hard Floor Gates. If a turn's context accuracy or task progress falls below 4.0, the turn's mean score is capped at 4.0 regardless of how high the style dimensions score. Formally, let S_{turn} be the weighted

mean from Equation (1):

$$\hat{S}_{\text{turn}} = \begin{cases} \min(S_{\text{turn}}, 4.0) & \text{if } s_{\text{accuracy}} < 4.0 \text{ or } s_{\text{progress}} < 4.0 \\ S_{\text{turn}} & \text{otherwise} \end{cases} \tag{3}$$

This ensures that a "fundamentally broken" turn cannot be inflated by style.

Layer 3: Disagreement Detection. When multiple judge models evaluate the same turn, we compute per-dimension statistics across judges. If the standard deviation exceeds 2.0 for any dimension, the dimension is flagged for disagreement. If the max–min spread exceeds 3.0, the consensus score for that dimension uses the *pessimistic* (minimum) value rather than the mean:

$$\hat{s}_d = \begin{cases} \min_j(s_{d,j}) & \text{if } \max_j(s_{d,j}) - \min_j(s_{d,j}) > 3.0 \\ \bar{s}_d & \text{otherwise} \end{cases} \tag{4}$$

where $s_{d,j}$ is the score for dimension d from judge j. This prevents the "mean anchor" exploit, where an agent produces responses calibrated to achieve middling scores from all judges rather than genuinely excellent output.

Layer 4: Programmatic Verification. Numerical assertions in verification criteria (*e.g.*, "Year 1 Revenue = $57.5M") are checked by code, not by LLM judges performing mental arithmetic. The `VerificationRunner` dispatches on the verification method specified by each scenario:

- `programmatic`: Parses comparison expressions, extracts numeric values from work product data, and evaluates arithmetic assertions with configurable tolerance (default: 1%).
- `mathematical`: Validates algebraic relationships between computed values.
- `structural`: Checks output structure (grid dimensions, required fields, section presence).
- `citation_validity`: Validates URL format, checks citation count, and verifies uniqueness.
- `statistical`: Verifies inequality relationships (*e.g.*, BH-adjusted p-values $\geq$ unadjusted p-values).

Programmatic verification results are injected into the LLM judge prompt as grounded context, constraining the judge's correctness evaluation with objective boolean evidence.

Layer 5: Edge Case Injection. Each scenario defines 4–5 adversarial edge cases designed to test professional judgment. Edge cases are injected into the turn sequence during the Edge Case phase and scored with a dedicated rubric evaluating: (a) whether the agent detected the problem, (b) whether it provided appropriate pushback, and (c) whether it avoided generating incorrect content. Examples include hallucination traps (asking about a fictional company "LithiumMax Solutions" in the research scenario), infeasible constraint sets (requesting debt that exceeds enterprise value in the financial scenario), and data integrity violations (using an outcome variable in a predictive model in the biostatistics scenario).

Layer 6: pass@k Reliability. Inspired by τ-bench [18], each scenario is run k times with different seeds. We use the plug-in pass@k estimator:

$$\text{pass@}k = 1 - (1 - \hat{p})^k \tag{5}$$

where $\hat{p}$ is the observed pass rate (fraction of runs achieving $\geq$ Peer tier). This measures the probability that at least one of k runs succeeds. We note that Chen et al. [3] showed this plug-in estimator is biased and proposed an unbiased combinatorial alternative; however, since ModylBench generates exactly $n = k$ samples (rather than $n \gg k$), the unbiased estimator degenerates to a binary indicator, making the plug-in form more informative for our use case. Note also that τ-bench uses $\hat{p}^k$ (all runs succeed), measuring reliability rather than capability; our worst-run tier assignment (Equation (6)) is more aligned with this pessimistic philosophy than the pass@k formula itself. However, for official leaderboard tier classification we use the *pessimistic* (worst-run) score to prevent "lucky run" reporting:

$$\text{Tier}_{\text{official}} = \text{Tier}(\min_{i \in \{1,\ldots,k\}} S_{\text{combined},i}) \tag{6}$$

The reliability report includes pass rate, standard deviation, 95% confidence intervals (using t-distribution critical values from a lookup table for small k, with interpolated values for degrees of freedom between table entries), and per-dimension variance to identify flaky scoring dimensions.

3.6 Mutation Trajectory Evaluation

Beyond evaluating the final work product (the *destination*), ModylBench tracks the complete history of mutations to every work product across all turns—the *mutation trajectory*. This addresses a critical gap: two agents that produce identical final deliverables may have taken radically different paths to get there. An agent that builds incrementally with correct edits at each turn is fundamentally different from one that produces the right answer after ten wrong edits, five backtracks, and three destructive overwrites.

Motivation. The mutation trajectory serves three purposes:

1. **RL training signal.** Each `(user_utterance, work_product_diff)` pair is ground truth for training edit-generation models. The trajectory provides dense, per-turn supervision rather than sparse end-of-scenario reward.
2. **Evaluation granularity.** Mutation-level scoring distinguishes between efficient, incremental builders and chaotic, backtracking editors—a distinction invisible to final-product-only evaluation.
3. **Programmatic verification.** Individual diffs can be verified mechanically ("did the agent add the capex row at turn 19?"), reducing reliance on LLM judgment for verifiable facts.

Data model. After each turn, the evaluation harness snapshots all active A2UI surfaces (spreadsheets, charts, documents, dashboards). A recursive deep-diff engine compares consecutive snapshots of the same work product, producing a list of `WorkProductMutation` records. Each mutation records:

- **Turn index** and **product ID**: which product changed at which turn.
- **Mutation type**: one of 23 canonical types (`create`, `update_cell`, `add_row`, `add_key`, `remove_key`, etc.).

- **JSON pointer path**: an RFC 6901-compliant pointer to the changed location (*e.g.*, `/income_statement/year1/revenue`).
- **Old and new values**: for programmatic diff verification.
- **Correctness flag**: set by matching against per-turn expected mutations declared in the scenario specification.

The complete `MutationTrajectory` for a scenario run is serialized alongside the scorecard and included in the HuggingFace dataset.

Scoring dimensions. The `MutationScorer` evaluates the trajectory along six dimensions:

Metric	Description
Efficiency	correct mutations/total mutations. Higher means a more direct path.
Convergence rate	Fraction of the turn range after the last mutation. Higher means the product stabilized early.
Backtrack count	Times a value reverted to a prior state (indecisive editing).
Unnecessary churn	Changes reverted within 2 turns (no net progress).
Destructive edits	Overwrites of previously correct values.
Missing mutations	Expected mutations (per scenario) that never appeared.

These metrics are complementary to the existing journey and destination scores. An agent with high destination scores but low mutation efficiency is producing correct deliverables through a wasteful process. An agent with high efficiency but many missing mutations is taking shortcuts that omit expected intermediate steps.

Connection to Yjs CRDTs. Modyl's runtime collaboration layer uses Yjs CRDTs [15] for real-time co-editing of work products. The mutation tracking model is designed to align with Yjs's operation-based data model: each `WorkProductMutation` corresponds naturally to a Yjs operation, and the trajectory can be replayed as a sequence of CRDT operations. This alignment enables future work on direct CRDT-level evaluation without snapshot-based diffing.

3.7 Generalized Evaluation (v3.0)

ModylBench v2.0 evaluates dyadic meetings (one human, one agent) with JSON-serializable work products across 6 turn-level and 5 product-level quality dimensions. Version 3.0 generalizes the benchmark across seven evaluation areas to support multi-participant, multi-channel, and multi-modal meeting evaluation. We summarize the five implemented modules and the resulting weight architecture.

3.7.1 Format Adapter Framework

The v2.0 mutation tracker assumes all work products are JSON-serializable dicts located via RFC 6901 pointers. To support non-CRDT work products—code files, terminal sessions, file system operations—v3.0 introduces a `FormatAdapter` protocol with four responsibilities:

1. **Normalization**: converting raw format data into a stable, ID-keyed dict representation that `deep_diff()` can process without generating spurious mutations.

2. **Diffing**: producing `WorkProductMutation` instances using format-appropriate algorithms (unified diff for code, append-only detection for terminal output).
3. **Verification**: format-specific correctness checks (syntax validation, test execution, tree comparison) that feed `mutation.is_correct` before trajectory scoring.
4. **Mutation type registration**: declaring the set of mutation types the format introduces.

A generalized locator system replaces rigid RFC 6901 JSON pointers with format-appropriate addressing: `line_range` for code hunks (e.g., `src/app.py:L10-L15`), `element_id` for canvas elements, `file_path` for file-level operations. The first concrete adapter, `CodeAdapter`, produces hunk-level mutations from unified diffs with rewrite detection as an anti-gaming signal: changes affecting >80% of a file's lines are flagged as `rewrite_file` rather than `edit_hunk`, penalizing agents that overwrite entire files instead of making targeted edits. The adapter also exports SWE-bench-compatible unified diff patches, enabling direct comparison with code generation benchmarks [9].

3.7.2 Multi-Party Evaluation

The v2.0 harness supports exactly one human participant. V3.0 extends evaluation to N-participant meetings with a `MultiPartyTestHarness` and 10 corpus-calibrated metrics organized into four composite dimensions:

Dimension	Component Metrics	Weight
Coordination quality	RC-PSI, DRC, BSP, SAA	0.35
Floor management	FYL, CRT, FPE	0.25
Prosodic naturalness	RLDS, PPN, BPR (shared with §3.7.3)	0.25
Social awareness	CNS, PABS	0.15

Role-Calibrated Participation Share Index (RC-PSI) measures whether the agent's speaking fraction matches role-appropriate norms. Rather than using a naïve $1/N$ participation target, baselines are drawn from an internal calibration corpus of professional meetings (mean Gini coefficient = 0.68, with only 24.1% of meetings exhibiting balanced participation, where "balanced" is defined as Gini < 0.30), partitioned by role (facilitator, expert contributor, note-taker, peer) and meeting size.[2] **Directed Response Coverage (DRC)** counts the fraction of turns addressed to the agent that receive a response within a K-turn window. **Bystander Silence Precision (BSP)** measures whether the agent correctly stays silent during exchanges between other participants. Three hard floors enforce minimum multi-party competence: DRC < 0.70 (ignoring >30% of direct addresses), BSP < 0.50 (interrupting >50% of non-addressed conversations), and competitive interruption count > 5 per meeting all trigger automatic failure regardless of content quality.

The v3.0 combined score for multi-party scenarios uses the formula:

$$S_{v3} = 0.60 \cdot \bar{Q}_{\text{turn}} + 0.25 \cdot \bar{D}_{\text{dynamics}} + 0.15 \cdot \bar{P}_{\text{product}} \tag{7}$$

where $\bar{Q}_{\text{turn}}$ is the weighted mean of turn quality dimensions (now 9, see §3.7.6), $\bar{D}_{\text{dynamics}}$ is the weighted mean of the four meeting dynamics dimensions above, and $\bar{P}_{\text{product}}$ is the existing 5-dimension work product score. For dyadic scenarios, the original 0.4/0.6 journey/destination split is preserved for backward compatibility.

[2] Corpus statistics are derived from an internal dataset of professional meetings and are consistent with published findings on meeting participation inequality; see Section 7 for discussion.

3.7.3 Prosody Metrics

V2.0 evaluates audio through either Whisper ASR (discarding all prosodic information) or freeform VLM observations (unstructured, not scored). V3.0 introduces three timing-based prosody metrics computable from existing turn-level data, plus APIs for audio-feature-dependent metrics planned for later phases.

Response Latency Distribution Shape (RLDS) compares the full distribution of the agent's response latencies against corpus-calibrated reference distributions using the Wasserstein (earth mover's) distance:

$$\text{RLDS} = 1 - \min\left(\frac{W_1(P_{\text{agent}}, P_{\text{corpus}})}{d_{\max}},\ 1\right) \tag{8}$$

where W_1 denotes the 1D Wasserstein distance, P_{agent} is the empirical latency distribution of the agent, P_{corpus} is a synthetic reference distribution interpolated from corpus percentile anchors, and $d_{\max}$ is a calibration constant (default 1.0 s, covering P90 human-human variation). The key insight is that an agent responding in a mechanically consistent ∼0.5 s is unnatural; the distribution shape—not just the mean—must match human variance. A coefficient-of-variation penalty further penalizes agents with CV < 0.15 (the roboticity threshold), and a too-fast penalty flags agents that consistently beat the corpus P10 floor of 0.12 s.

Pause Pattern Naturalness (PPN) classifies each agent pause as *natural*, *needs filler*, or *broken* using context-conditional thresholds: conversational pauses above 1.2 s require a filler ("let me check"); tool-use pauses are tolerated up to 5.0 s; topic-change pauses up to 3.0 s. The score is the fraction of pauses in the acceptable range.

Backchannel Production Rate (BPR) scores how closely the agent's backchannel frequency ("mm-hmm", "right", "yeah") matches the corpus median of 1.48 per minute, penalizing both silent agents and sycophantic over-producers.

3.7.4 Channel Evaluation

V2.0 captures data from three interaction channels—annotations, reactions, and chat—but never passes them to judges or scores them. V3.0 promotes these to first-class evaluation dimensions with per-channel composite scores.

Annotations are scored on three sub-dimensions: spatial precision (computed overlap with target regions plus VLM assessment), tool appropriateness (judged: arrow vs. highlight vs. freehand for the given intent), and clutter management (computed: active annotation count, clear rate, decay). Scores are aligned with the W3C Web Annotation Data Model for storage interoperability.

Reactions are scored on contextual relevance (judged: emoji-context alignment using a pre-computed valence mapping of 15+ common emojis), timing quality (computed: optimal latency window 1.5–3.0 s after the triggering event), and social awareness (judged: bidirectional — the agent both produces and responds to reactions). Anti-gaming controls include a frequency cap of one reaction per turn, a 0.5 s timing floor, and a monotony penalty for repetitive emoji usage.

Chat is scored on responsiveness (computed: latency from trigger to response), content quality (judged: formatting, code validity, URL validity), channel selection (judged: appropriate use of chat vs. voice for the content type), and voice-chat coherence (judged plus computed: cross-reference detection between simultaneous speech and chat messages, with a -1.5 penalty for verbatim voice-chat duplication).

A `CrossChannelCoherenceScorer` evaluates how well all channels work together—voice-annotation synchronization, voice-chat consistency, and voice-reaction alignment—producing a meta-dimension that penalizes channel conflicts.

3.7.5 Video Evaluation

V2.0 never consumes video tracks: `SyntheticUser` handles `KIND_AUDIO` only and screen captures lack temporal metadata. V3.0 introduces five components addressing this gap.

TimestampedMediaCollector replaces the separate `TurnAudioCollector` and `ScreenCapture` with a unified collector that preserves timing relationships between audio, screen-share, and camera streams via a shared monotonic clock reference. All media for a turn are returned as an `AlignedTurnMedia` bundle with WAV audio and chronologically sorted keyframes.

KeyframeExtractor samples screen-share frames at 1–2 FPS and applies perceptual hash (dHash, 64-bit) deduplication with a Hamming distance threshold of $\leq$6 bits, then enforces an 8-keyframe-per-turn budget by selecting frames with the largest SSIM delta from their predecessor. This ensures VLM judges receive visually distinct frames rather than near-identical captures.

VisualArtifactChecker runs pre-LLM programmatic checks on rendered A2UI surfaces: WCAG AA contrast ratio verification (4.5:1 for normal text, 3.0:1 for large text and graphical objects), font readability assessment, and axis label presence detection for chart-type artifacts. These boolean results are injected into the VLM judge rubric as grounded context, reducing hallucination in visual quality assessment.

AVSyncScorer detects deictic references in the agent's speech ("as you can see," "looking at this chart") via regex pattern matching on word-timestamped transcripts, then checks that a corresponding screen-share change occurred within a 5-second window. The score is $1 - (\bar{d}/d_{\max})$ where $\bar{d}$ is the mean absolute offset between references and nearest screen changes.

ScreenContentTracker computes SSIM between consecutive screen-share frames to measure layout stability during A2UI token streaming. The implementation uses a simplified global-statistics SSIM rather than the local windowed approach of Wang *et al.* (2004); this is sufficient for the binary change-detection use case (threshold 0.95) but may be less sensitive to small localized changes than full MSSIM. High SSIM variance indicates layout thrashing; the stability score penalizes both low mean SSIM and high variance. Change timestamps feed into the AVSyncScorer for deictic reference resolution.

Visual correctness errors are routed to substance dimensions (`task_progress`, `context_accuracy`), not merely `presentation_quality`, ensuring that an agent showing incorrect data on screen is penalized on content, not just aesthetics.

3.7.6 Extended Weight Architecture

The v3.0 weight architecture expands turn-level evaluation from 6 to 9 dimensions by adding annotation, reaction, and chat composites. Weights are conditionally activated based on which channels are active in a given scenario:

Dimension	Base (v2)	Extended (v3)
context_accuracy	0.25	0.22
task_progress	0.25	0.22
iteration_quality	0.20	0.18
adaptability	0.15	0.13
presentation_quality	0.10	0.08
social_quality	0.05	0.03
annotation_composite	—	0.05
reaction_composite	—	0.04
chat_composite	—	0.05

The substance-over-style principle is preserved: the core substance cluster (`context_accuracy` + `task_progress` + `iteration_quality`) totals 0.62 in extended mode versus 0.70 in base mode. The 8-percentage-point reduction is distributed to the three channel dimensions, which carry genuine signal about agent capability in multi-channel environments. Hard floors on `context_accuracy`, `task_progress`, and `correctness` remain inviolable across both weight modes.

3.8 Judge Panel Configuration

ModylBench supports a configurable panel of judge models. The default configuration uses three frontier models:

Provider	Model	Capabilities
Anthropic	Claude Opus 4.6	Text-based judging with extended thinking
Google	Gemini 3.1 Pro	Multimodal judging (native audio + image)
OpenAI	GPT-5.2	Multimodal judging (native audio + image)

For text-based evaluation, all three models receive the same rubric prompt and return structured JSON scores. For multimodal evaluation, Gemini and GPT receive raw audio and image inputs natively; Claude receives ASR-transcribed text as a fallback. Consensus scores are computed as described in Layer 3 above, with pessimistic fallback on high disagreement.

For work product verification, the panel uses majority-vote consensus: each programmatic criterion is evaluated by all judges, and the criterion passes if a strict majority of judges agree.

4 Evaluation Protocol

ModylBench is designed for straightforward adoption by the research community and meeting agent developers.

4.1 Running ModylBench

The evaluation harness is distributed as a Python package:

```
pip install modylbench
modylbench run --scenarios all --k 3 --output results/
```

The runner requires a deployed meeting agent accessible via LiveKit. The synthetic user joins the meeting room, executes the scripted scenario, captures the agent's audio and visual output, and invokes the judge panel. Results are saved as JSON scorecards, Markdown reports, and HTML dashboards.

4.2 Submission Requirements

Official leaderboard submissions require:

1. All 6 scenarios evaluated.

2. Minimum $k = 3$ runs per scenario (recommended $k = 5$).
3. The official tier is determined by the worst-run combined score (Equation (6)).
4. At least 2 judge models from the supported panel.
5. The full JSON scorecard including per-turn evaluations, disagreement flags, and verification results.

4.3 Evaluation Tiers

ModylBench recognizes two evaluation modes with distinct reporting:

Multimodal Evaluation (Gold): Uses native audio and image input to judges. Reports include audio-visual coherence scores and prosodic observations.

Text-Only Evaluation (Silver): Uses ASR transcription for audio and structured text for visual output. A valid evaluation but does not capture prosodic quality or audio-visual synchronization.

Both modes use the same scoring dimensions, weights, and tier thresholds. Submissions must declare which mode was used.

4.4 Evaluation Algorithm

The complete evaluation procedure is presented in Algorithm 1.

5 Experiments

We present the experimental setup and preliminary results from baseline evaluations of ModylBench against current meeting agent systems.

5.1 Experimental Setup

Agent under test. Baseline evaluations target three configurations: (1) a Claude Opus 4.6-backed agent with A2UI rendering capabilities (the Modyl agent), (2) a Gemini 3.1 Pro-backed agent as an alternative backbone, and (3) a text-only chatbot baseline (no voice, no A2UI) to measure the gap between conversational and meeting-native agents.

Judge configuration. All evaluations use the three-judge panel described in Section 3.8. Multimodal judging is used for agents that produce audio and visual output; text-only judging is used for the chatbot baseline.

Reliability. Each scenario is run $k = 5$ times per agent configuration, producing 30 scorecards per agent (6 scenarios $\times$ 5 runs). Official tiers use the worst-run combined score (Equation (6)).

5.2 Preliminary Results

Preliminary results from initial evaluation runs are presented below.[3]

Table 3. Overall MODYLBENCH results by agent. Combined scores are the pessimistic (worst-run) combined score across $k = 5$ runs. Tier assignment uses Equation (5). *Estimated; full results pending.*

Agent	Journey	Destination	Combined	Tier	Pass Rate	σ
Modyl (Opus 4.6)	6.1	5.2	5.56	<Peer	0.40	0.62
Modyl (Gemini 3.1 Pro)	5.8	4.9	5.26	<Peer	0.20	0.71
Text-only baseline	4.2	2.3	3.06	<Peer	0.00	0.48

Table 4. Per-vertical breakdown for the primary agent (Modyl, Opus 4.6). Scores are the mean combined score across $k = 5$ runs with standard deviation. *Estimated; full results pending.*

Vertical	Journey	Dest.	Combined	σ	Tier	Pass@5
Financial Analyst	5.8	4.7	5.14	0.74	<Peer	0.2
Deep Researcher	6.6	5.8	6.12	0.55	Peer	0.4
Business Strategist	6.3	5.5	5.82	0.63	<Peer	0.4
Optimization Solver	5.4	4.3	4.74	0.81	<Peer	0.0
Scientist	5.9	4.9	5.30	0.69	<Peer	0.2
Business Analyst	6.5	5.6	5.96	0.58	<Peer	0.4

5.3 Analysis Plan

Upon completing baseline evaluations, we plan to analyze:

1. **Text-only vs. multimodal evaluation gap:** How much does multimodal judging reveal that text-only judging misses? We expect multimodal evaluation to produce lower social quality scores (detecting robotic delivery) and different presentation quality scores (evaluating actual rendered surfaces rather than structural descriptions).
2. **Per-dimension difficulty:** Which of the 11 dimensions are hardest for current agents? We hypothesize that iteration quality and edge case handling will be the weakest dimensions, as they require the agent to revise previous work and detect adversarial inputs.
3. **Anti-gaming effectiveness:** We will run a "polite but wrong" ablation by constructing an agent that maximizes style dimensions while producing deliberately incorrect content, and measure the score difference between the anti-gaming-protected evaluation and a naïve equal-weight evaluation.
4. **Inter-judge agreement:** We will report dimension-level agreement rates across the three-judge panel, including the frequency of disagreement flags and pessimistic fallback activations.
5. **Reliability analysis:** We will characterize which scenarios and dimensions exhibit the highest variance across the $k = 5$ runs, identifying "flaky" evaluation dimensions that may require larger k or rubric refinement.

[3] Estimated from early evaluation runs; full results with significance analysis pending. Values should be treated as indicative ranges, not final measurements.

Table 5. Per-dimension analysis for the primary agent (Modyl, Opus 4.6). Shows which quality dimensions are hardest for current meeting agents. Values are mean scores across all turns (journey) or products (destination) across all scenarios. *Estimated; full results pending.*

Axis	Dimension	Weight	Mean	Min	Max
Journey	Context Accuracy	0.25	6.4	4.1	8.2
	Task Progress	0.25	5.9	3.5	7.8
	Iteration Quality	0.20	5.1	2.8	7.5
	Adaptability	0.15	6.8	4.6	8.5
	Presentation Quality	0.10	6.5	4.2	8.3
	Social Quality	0.05	7.6	5.8	9.1
Destination	Correctness	0.30	4.8	2.5	7.2
	Completeness	0.25	5.3	3.1	7.6
	Actionability	0.20	5.0	2.9	7.4
	Professional Quality	0.15	5.9	3.8	7.9
	Format & Presentation	0.10	6.3	4.5	8.1

6 Analysis

While full empirical analysis awaits the baseline evaluation runs described in Section 5, we present analytical results from the benchmark development process.

6.1 Adversarial Cross-Validation Findings

The development of ModylBench was informed by a systematic adversarial review in which three independent reviewer models (Claude Opus 4.6, Codex, and Gemini 3.1 Pro) audited the complete evaluation framework. The reviewers identified seven significant issues, three rated as critical. All three reviewers independently converged on the same three critical findings, providing high confidence that these were genuine structural flaws rather than artifacts of a single reviewer's perspective.

The critical findings and their resolutions were:

1. **Blind judges (C-1):** The initial implementation passed only metadata ("Agent produced audio response; A2UI surface(s): 2") to judges rather than actual agent content. Resolution: ASR transcription of agent audio and serialization of A2UI surface data into the judge prompt, plus the multimodal evaluation pathway.
2. **Phantom verification (C-2):** Verification criteria ("Year 1 Revenue = $57.5M") were checked by LLM judges performing mental arithmetic rather than by code. Resolution: the `VerificationRunner` with dispatch on verification method (Layer 4).
3. **Dead edge cases (C-3):** Edge cases were defined in every scenario but never injected into the execution flow. Resolution: edge case injection during the Edge Case phase (Layer 5).

Four additional high-severity findings led to the dimension weighting system (Layer 1), hard floor gates (Layer 2), disagreement detection (Layer 3), and pass@k reliability testing (Layer 6) described in Section 3.5.

6.2 The "Polite but Wrong" Analysis

To quantify the effectiveness of the anti-gaming measures, we analyze the theoretical scoring of a hypothetical agent optimized for style over substance.

Consider an agent with the following per-turn scores: $s_{\text{social}} = 9$, $s_{\text{presentation}} = 9$, $s_{\text{adaptability}} = 9$, $s_{\text{accuracy}} = 2$, $s_{\text{progress}} = 2$, $s_{\text{iteration}} = 2$.

Scoring Method	Turn Score	Effect
Equal weights (naïve)	5.50	Passes Peer threshold with 6.3 product score
Weighted dimensions (Layer 1)	4.10	Below Peer; style cannot compensate
+ Hard floor (Layer 2)	4.10 (capped at 4.0)	Further constrained

Under naïve equal weighting with a product score of 6.3, the combined score would be $0.4 \times 5.5 + 0.6 \times 6.3 = 5.98$—just below Peer but dangerously close. Under the ModylBench framework, the turn score drops to 4.10 and is capped at 4.0, yielding a combined score of $0.4 \times 4.0 + 0.6 \times 6.3 = 5.38$—safely below Peer. More importantly, with realistic product scores for a substance-poor agent (likely 4–5, not 6.3), the combined score would fall to 3.9–4.5, well below any tier threshold.

6.3 Dimension Weight Sensitivity

The choice of dimension weights is a design decision with meaningful impact on scores. We analyze the sensitivity of tier classifications to weight perturbations. Specifically, we vary the substance/style ratio from the current 70/30 to alternatives:

Substance/Style Ratio	"Polite but Wrong" Score	"Balanced Agent" Score	Discrimination
50/50 (equal)	5.50	7.00	1.50
60/40	4.80	7.00	2.20
70/30 (ours)	**4.10**	**7.00**	**2.90**
80/20	3.40	7.00	3.60

Here the "balanced agent" scores 7/10 uniformly across all dimensions. The 70/30 ratio provides strong discrimination (2.90 points) between the polite-but-wrong agent and the balanced agent without being so extreme that presentation quality becomes meaningless.

7 Limitations and Future Work

ModylBench represents a first step toward systematic evaluation of AI meeting agents. We acknowledge several limitations and outline planned extensions.

Synthetic scenarios. All six scenarios use scripted human turns, creating a controlled but artificial meeting dynamic. Real meetings involve interruptions, topic digressions, multi-party dynamics, and unpredictable conversational flow. An agent that excels on scripted scenarios may struggle in unscripted settings. Future work will introduce "free-form" scenarios where the synthetic user has goals but no scripted turns, as well as variable-length scenarios (4, 8, 12, and 16 turns) to test adaptability to different meeting cadences.

Fixed scenario length. All six current scenarios contain 48–52 turns following a four-phase pattern calibrated against observed meeting structure. While edge case injection and natural conversational flow provide some protection against template-replay attacks (see Section 3.5, Layer 5), expanding to variable-length scenarios with randomized phase ordering and held-out evaluation sets is planned for robust leaderboard evaluation.

Limited vertical coverage. Six professional verticals represent a small sample of the contexts in which meeting agents are deployed. Planned expansions include portfolio management, research direction, regulatory compliance, and accounting/bookkeeping verticals, as well as non-professional settings (educational tutoring, medical consultation, customer support).

Uncalibrated tier thresholds. The tier thresholds (Peer $\geq$ 6.0, Mentor $\geq$ 7.5, Consultant $\geq$ 9.0) are set by expert judgment without empirical calibration against human expert performance. A planned human baseline study will score recordings from a large proprietary meeting corpus using the same rubrics, establishing the distribution of real human meeting quality and calibrating thresholds to meaningful reference points.

Judge model dependency. The benchmark's scores depend on the judge models used. Different frontier models may have different biases, and scores from ModylBench are not absolute quality measurements but relative comparisons within a consistent judge configuration. The disagreement detection mechanism (Layer 3) partially mitigates this, but systematic judge bias remains a concern. Future work will compute inter-annotator agreement statistics (Krippendorff's α [10]) across judge model combinations.

Single-agent and multi-agent evaluation. V3.0 (§3.7.2) extends evaluation to multi-human scenarios with a single AI agent. However, multi-agent meetings—where two or more AI agents interact in the same room, creating emergent coordination dynamics not covered by human meeting science—remain unaddressed. Additionally, cultural calibration of prosody norms, interruption tolerance, and silence comfort across different professional cultures is an open research question. We note that the v3.0 multi-party dynamics and prosody modules require multi-participant scenarios ($N > 2$) to produce meaningful scores; the current 6 standard scenarios are dyadic (one human, one agent), so these modules are exercised only in extended evaluation configurations. A planned v3.1 expansion will introduce multi-participant standard scenarios to fully exercise the coordination, floor management, and prosodic naturalness metrics in the default benchmark suite.

Latency measurement. Current latency metrics measure end-to-end time (human utterance to agent response event) without isolating agent processing time from infrastructure overhead (TTS generation, network propagation, track subscription). More granular latency decomposition is needed for meaningful performance comparison.

Remaining generalization targets. The v3.0 generalization (§3.7) implements format adapters, multi-party metrics, prosody evaluation, channel scoring, and video analysis. Several format adapters remain planned: terminal session tracking, file system operation diffing, canvas element-level comparison, and whiteboard stroke-level evaluation. Camera and avatar evaluation (Visual Capability Profile C) is deferred until

avatar-equipped agents reach critical mass in production deployments. Theory-of-mind operationalized metrics—question-type prediction accuracy, disagreement management, and topic-depth detection—are specified but require further validation against human baselines.

8 Conclusion

We have introduced ModylBench, the first benchmark designed to evaluate AI agents as work-product-generating meeting participants. The benchmark addresses a critical gap in the agent evaluation landscape: while coding agents, web agents, and tool-using agents have established benchmarks, the growing category of meeting agents—which communicate through voice, produce interactive visual deliverables, and collaborate in real-time with human professionals—has lacked evaluation frameworks that assess deliverable quality, mutation trajectories, and anti-gaming robustness.

ModylBench makes four technical contributions. First, it defines a multi-dimensional scoring framework that evaluates both the conversational process (journey: 6 turn-level dimensions, extended to 9 in v3.0) and the resulting deliverables (destination: 5 product-level dimensions), with substance-weighted scoring that prevents style-over-substance gaming. Second, it introduces a multimodal evaluation protocol that sends raw audio and screen captures directly to judge models, preserving prosodic and audio-visual coherence signals that text-only pipelines discard. Third, it implements a six-layer anti-gaming framework developed through adversarial cross-validation, addressing specific reward-hacking vulnerabilities with targeted defenses including weighted dimensions, hard floor gates, disagreement detection, programmatic verification, edge case injection, and pass@k reliability testing. Fourth, version 3.0 generalizes the framework to multi-participant meetings with 10 corpus-calibrated dynamics metrics, multi-channel evaluation (annotations, reactions, chat), prosodic naturalness scoring via Wasserstein latency distribution comparison, video stream analysis with keyframe deduplication and WCAG accessibility checks, and a format-adapter architecture for non-CRDT work products.

The benchmark, evaluation harness, and accompanying HuggingFace dataset are released as open-source resources to enable community benchmarking of meeting agents. We view ModylBench as a living benchmark: its scenario library, vertical coverage, and anti-gaming defenses will evolve as the meeting agent ecosystem matures and new evaluation challenges emerge.

The ultimate goal is not merely to rank meeting agents but to establish a shared standard for what "good" meeting participation looks like—and to push the field toward agents that are not just polite and well-formatted, but substantively correct, genuinely helpful, and worthy of a seat at the table.

Acknowledgments

The adversarial cross-validation methodology was developed with contributions from three independent reviewer configurations. The design principles and anti-gaming framework benefited from the systematic identification of reward-hacking vulnerabilities across reviewer perspectives. We thank the open-source communities behind LiveKit, the A2UI protocol, and the evaluation frameworks that inspired aspects of this work.

References

[1] a2ui.org. A2ui: Agent-to-user interface protocol. https://a2ui.org, 2025. Accessed: 2026-02-26.

[2] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, M Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. The AMI meeting corpus: A pre-announcement. *Machine Learning for Multimodal Interaction*, pages 28–39, 2005.

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[4] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, et al. Chatbot arena: An open platform for evaluating LLMs by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

[5] Daily.co. Pipecat: Open source framework for voice and multimodal conversational AI. https://github.com/pipecat-ai/pipecat, 2024. Accessed: 2026-02-26.

[6] Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. MeetingBank: A benchmark dataset for meeting summarization. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

[7] Yuelin Hu, Jun Xu, Bingcong Lu, Zhengxue Cheng, Hongwei Hu, Ronghua Wu, and Li Song. MeetBench-XL: Calibrated multi-dimensional evaluation and learned dual-policy agents for real-time meetings. *arXiv preprint arXiv:2602.03285*, 2026.

[8] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. The ICSI meeting corpus. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 364–367, 2003.

[9] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

[10] Klaus Krippendorff. Computing Krippendorff's alpha-reliability. *University of Pennsylvania Departmental Papers*, 2011.

[11] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. *arXiv preprint arXiv:2305.14387*, 2023.

[12] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024.

[13] LiveKit Inc. LiveKit Agents: Open-source framework for building real-time AI applications. https://github.com/livekit/agents, 2024. Accessed: 2026-02-26.

[14] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raber, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations*, 2024.

[15] Petru Nicolaescu, Kevin Jahns, Michael Derntl, and Ralf Klamma. Yjs: A framework for near real-time P2P shared editing on arbitrary data types. In *Proceedings of the 15th International Conference on Web Engineering (ICWE)*, pages 154–168. Springer, 2015.

[16] Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie S. Kim, Patrick Chao, Samuel Miserendino, Gildas Chabot, David Li, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating AI model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025.

[17] Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, et al. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

[18] Shunyu Yao, Karthik Narasimhan, and Noah A. Cao. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

[19] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, et al. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36, 2023.

[20] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 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*, pages 5905–5921, 2021.

[21] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.

Algorithm 1 ModylBench Evaluation Procedure

Input: Scenario $\mathcal{S}$ with turns $\{t_1, \ldots, t_n\}$, expected outputs $\{o_1, \ldots, o_m\}$, edge cases $\{e_1, \ldots, e_p\}$
Input: Judge panel $\mathcal{J} = \{J_1, \ldots, J_q\}$, reliability parameter k
Output: Reliability report $\mathcal{R}$ with tier classification

for run $= 1$ **to** k **do**
  SEED $\leftarrow$ run
  Join synthetic user and agent to LiveKit room
  for each scripted turn $t_i \in \mathcal{S}$ **do**
    Publish human utterance via TTS
    Collect agent audio a_i via TURNAUDIOCOLLECTOR
    Collect agent screenshots v_i via SCREENCAPTURE
    for each judge $J_j \in \mathcal{J}$ **do**
      if J_j supports multimodal input **then**
        $\mathbf{s}_{i,j} \leftarrow J_j$.EVALMULTIMODAL$(a_i, v_i, t_i)$
      else
        transcript$_i \leftarrow$ ASR(a_i)
        $\mathbf{s}_{i,j} \leftarrow J_j$.EVALTEXT(transcript$_i, t_i$)
      end if
    end for
    $\mathbf{s}_i \leftarrow$ PESSIMISTICCONSENSUS$(\{\mathbf{s}_{i,j}\}_{j=1}^{q})$ ▷ Layer 3
    $\hat{\mathbf{s}}_i \leftarrow$ APPLYHARDFLOOR$(\mathbf{s}_i)$ ▷ Layer 2
  end for
  Capture work products via A2UI surfaces
  $\mathbf{v} \leftarrow$ PROGRAMMATICVERIFY(products, $\mathcal{S}$.criteria) ▷ Layer 4
  $\mathbf{p} \leftarrow$ JUDGEPRODUCTS(products, $\mathcal{S}$.expected, $\mathbf{v}$)
  $S_{\text{journey}} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{s}}_i$.mean() ▷ Layer 1 weights
  $S_{\text{dest}} \leftarrow \frac{1}{m}\sum_{j=1}^{m} \mathbf{p}_j$.mean()
  $S_{\text{combined}}^{(\text{run})} \leftarrow 0.4 \cdot S_{\text{journey}} + 0.6 \cdot S_{\text{dest}}$
end for
$\mathcal{R} \leftarrow$ RELIABILITYREPORT$(\{S_{\text{combined}}^{(1)}, \ldots, S_{\text{combined}}^{(k)}\})$ ▷ Layer 6
Tier $\leftarrow$ CLASSIFY$(\min_{\text{run}} S_{\text{combined}}^{(\text{run})})$
return $\mathcal{R}$, Tier

Appendices

A Scoring Rubrics

A.1 Turn Evaluation Rubric (Text Mode)

The following rubric is sent to each judge model for turn-level evaluation. Variables in braces are filled from the scenario context and turn data.

```
You are an expert meeting quality evaluator. Score this agent turn
on a 1-10 scale across these dimensions:

1. social_quality (1-10): Appropriate greeting, acknowledgment,
   professional tone, empathy.
2. context_accuracy (1-10): Did the agent correctly understand what
   was asked? Did it misinterpret or hallucinate context?
3. task_progress (1-10): Does this turn advance toward the
   deliverable? Is progress measurable?
4. adaptability (1-10): How well did the agent handle corrections,
   curveballs, or changes in direction?
5. presentation_quality (1-10): Were A2UI surfaces rendered
   correctly? Was data formatted clearly?
6. iteration_quality (1-10): When the user asked for changes, were
   they applied correctly and completely?

Context:
- Vertical: {vertical}
- Scenario: {scenario_title}
- Human persona: {persona}
- Meeting goal: {goal}

This Turn:
- Turn {turn_index} (Speaker: {speaker})
- Human said: {human_content}
- Expected agent action: {expected_action}
- Agent actually did: {agent_response}
- Channel: {channel}
```

```
- Latency: {latency_ms}ms

Respond with ONLY valid JSON.
```

A.2 Multimodal Turn Evaluation Rubric

The multimodal rubric extends the text rubric with audio-visual evaluation instructions:

```
You will receive the agent's ACTUAL AUDIO RESPONSE (as a WAV file)
and SCREENSHOTS of what the agent displayed on screen. Evaluate the
meeting turn holistically.

Audio-Visual Coherence:
- Does the agent verbally reference what it is showing on screen?
- Is the timing appropriate (visuals appear when discussed)?
- Does the overall experience feel like a competent human colleague?

[... same 6 dimensions, adapted for audio/visual evidence ...]

Score each dimension 1-10. Include:
- audio_observations: tone, pacing, confidence notes
- visual_observations: screenshot and A2UI surface assessment
- coherence_notes: how well audio and visuals work together
```

B Work Product Verification Methods

Table 6 summarizes the programmatic verification methods and the types of assertions each handles.

Table 6. Programmatic verification methods in MODYLBENCH. Each method addresses a category of objectively verifiable criteria that should not be left to LLM judgment alone.

Method	Description	Example Criterion
`programmatic`	Parses comparison expressions and evaluates arithmetic assertions against extracted values	Year 1 Revenue == $57.5M
`mathematical`	Validates algebraic relationships between computed quantities	Sponsor Equity = EV − Total Debt
`structural`	Checks output structure: grid dimensions, required fields, section counts	Sensitivity table is a 5×5 grid
`citation_validity`	Validates URL format, citation count, uniqueness	$\geq$ 12 unique citations with valid URLs
`statistical`	Verifies inequality relationships between statistical quantities	BH-adjusted $p \geq$ unadjusted p
`data_consistency`	Cross-checks values between related outputs	Final IRR reflects management dilution

C Complete Scenario Inventory

Table 7 provides the complete inventory of scenarios, turns, expected outputs, and edge cases in ModylBench.

Table 7. Complete scenario inventory. Each scenario defines 8 scripted turns following the four-phase structure (Context → Work → Edge Case → Delivery).

Vertical	Persona	Turns	Products	Edge Cases	Verification Method
Financial Analyst	PE Associate	8	3	5	`programmatic`
Deep Researcher	Technology VP	8	3	5	`citation_validity`
Business Strategist	Strategy Dir.	8	3	5	`structural`
Optimization Solver	Operations VP	8	3	4	`mathematical`
Scientist	Lead Biostat.	8	3	5	`statistical`
Business Analyst	RevOps Dir.	8	3	5	`data_consistency`
Total		**48**	**18**	**29**	

D Reliability Report Schema

The reliability report for each scenario includes:

- **Basic statistics**: k, mean combined score, standard deviation, minimum, maximum.
- **95% confidence interval**: Using t-distribution for small k, with a lookup table for critical values at $df = k - 1$. For degrees of freedom between table entries, the implementation uses the next higher tabulated df value (yielding slightly narrower intervals); exact computation via `scipy.stats.t.ppf` is recommended for $k > 10$.
- **Pass rate**: Fraction of k runs achieving the Peer tier (≥ 6.0).
- **pass@k**: Estimated probability that at least one of k runs passes, computed via the plug-in estimator $1 - (1 - \hat{p})^k$ where $\hat{p}$ is the observed pass rate. See Section 3.5 for the distinction from the unbiased estimator of Chen et al. [3].
- **Pessimistic tier**: Tier classification based on the worst (minimum) combined score across k runs.
- **Per-dimension variance**: Variance of each scoring dimension across k runs, identifying "flaky" dimensions with variance > 1.0 (on the 1–10 scale).
- **Seeds used**: The random seeds for each run, enabling exact reproduction.

E Dataset Card

The ModylBench HuggingFace dataset contains:

- **Scenarios** (6 records): Complete scenario configurations including turns, expected outputs, edge cases, and verification criteria, serialized as JSON.
- **Rubrics** (4 templates): Turn evaluation, product evaluation, multimodal turn evaluation, and audio-only turn evaluation rubric templates.
- **Scoring metadata**: Dimension names, weights, tier thresholds, hard floor thresholds, and disagreement detection parameters.
- **Baseline scorecards** (preliminary): Evaluation results from baseline agent configurations, including per-turn scores, per-product scores, verification results, and reliability reports. Full scorecards will be updated as evaluation runs complete.

The dataset is structured for direct use with the `modylbench` Python package:

```
from datasets import load_dataset
ds = load_dataset("use-aleatoric/modylbench")
scenarios = ds["scenarios"]
rubrics = ds["rubrics"]
```