REPETITIONBENCH: HEALTHCARE CONVERSATIONAL BENCHMARK FOR MULTI-TURN AGENT REPETITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Repetition is a pervasive failure mode in conversational Large Language Model (LLM) agents, manifesting as re-asked questions, redundant confirmations, and unnecessary restatements. Such behavior disrupts the natural flow of the dialogue and erodes user trust. Yet repetition remains under-measured, and existing benchmarks do not stress-test this behavior in realistic, conversational settings. To address this gap, we introduce REPETITIONBENCH, a benchmark for assessing when repetition becomes inappropriate in extended, task-oriented dialogue. We study phone-based healthcare conversations between humans and AI agents. REPETITIONBENCH comprises (i) a fine-grained taxonomy spanning semantic and surface-level repetition; (ii) 13 corresponding verifiers (6 deterministic and 7 subjective); and (iii) an adversarial set of 6,892 contexts from conversations averaging over 20 agent turns, grouped into easy and hard tiers. A companion REPETITIONBENCH-GOLD contains 624 human-annotated turn samples; two candidate responses per sample yield 8,736 response items and 18,372 nurse judgments after adjudication. We compare 13 frontier models released over the past two years for failures across multiple repetition categories. We also show that post-training of LLMs using our repetition-verifiers leads to the lowest failure rates on all 14 evaluated models. To the best of our knowledge, REPETITIONBENCH is the first work to enable a systematic study of repetition in multi-turn conversational LLM agents, providing a basis for scalable evaluation prior to real-world deployment, while the verifiers in this work can be used to evaluate and analyze different modes of repetition.[1]

1 INTRODUCTION

Multi-turn conversational agents must sustain *natural*, *engaging*, multi-turn dialogue when interacting with users in real-world domains such as healthcare (Laranjo et al., 2018; Fitzpatrick et al., 2017; See et al., 2019). However, state-of-the-art LLMs do not yet achieve the same level of naturalness as humans, and a common failure mode they suffer is *repetition*. Rather than rephrasing or referring to prior context, agents built on these LLMs often ask the same question, repeat already confirmed details, or mirror the user's language verbatim. These behaviors are uncharacteristic of natural human-human dialogue when done unnecessarily (Li et al., 2016; Holtzman et al., 2020; Welleck et al., 2020; See et al., 2019). This repetitive behavior can erode users' trust in the agent and cause them to disengage before crucial tasks (e.g., clinical assessments) are complete (Bickmore et al., 2010; Miner et al., 2017). This impact is particularly acute in healthcare conversations: reviews of a text-based mental-health support app describe redundant responses as frustrating and unhelpful (Chaudhry & Debi, 2024) and agents may ask well-formed questions that patients have already answered in long medical conversations (Lim et al., 2025). This makes it important to evaluate a response in the context of the full conversation.[2]

On the other hand, not all repetition is undesirable (Laranjo et al., 2018; Miner et al., 2017): agents may empathetically acknowledge a patient's concern (Suchman et al., 1997), repeat an unanswered question, or confirm critical information (Talevski et al., 2020). The boundary between *appropriate* and *unnecessary* repetition is therefore subtle. The challenge is to distinguish these helpful restatements from redundant ones using the dialogue history. Simple similarity metrics such as fuzzy string match cannot reliably make this distinction (Deriu et al., 2021; Mehri & Eskenazi, 2020; Yeh et al., 2021). Existing work has largely sought to reduce repetition during training (See et al., 2019; Li et al.,

[1] We release the dataset at Kaggle.

[2] In our phone-call simulations with nurses role-playing patients, inappropriate repetition occurs in 13.1% of dialogue contexts.

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

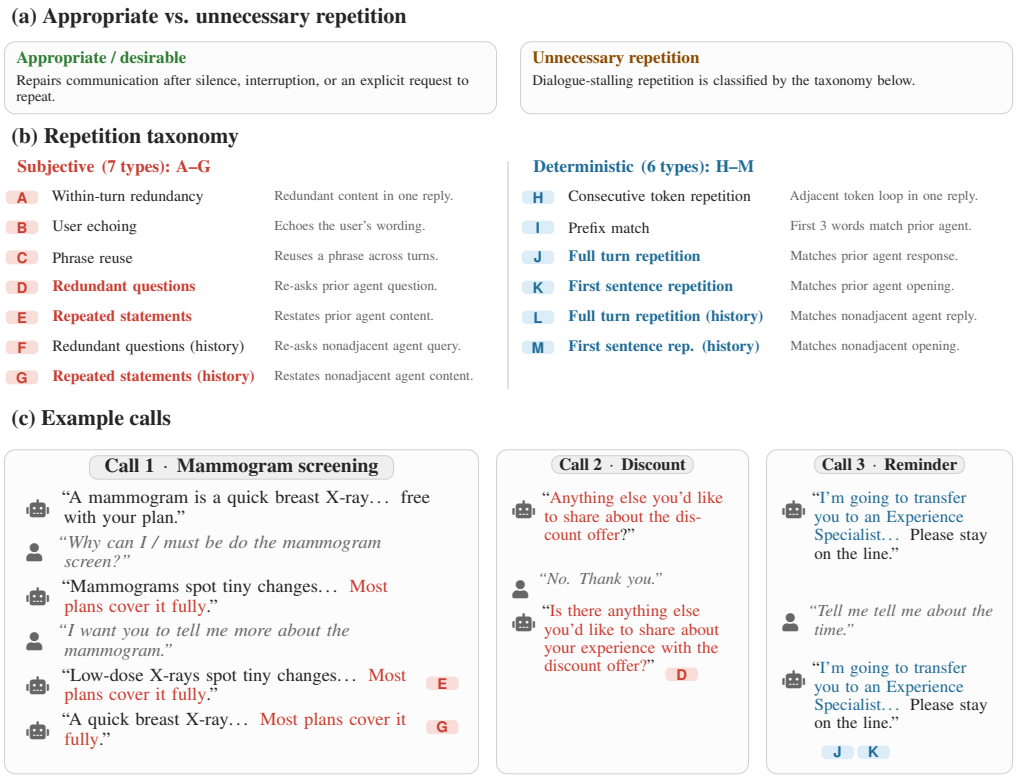


Figure 1: **Repetition types and annotated examples.** (a) Distinction between appropriate and unnecessary repetition; (b) taxonomy of 13 repetition types; (c) three clinical calls showing why both subjective and deterministic checks matter for assessing repetition.

2020; Shi et al., 2021) or treat repetition only as an auxiliary signal rather than an independent failure dimension (Zheng et al., 2023; Bai et al., 2024; Sirdeshmukh et al., 2025; Ma et al., 2024).

To address this gap, we present REPETITIONBENCH, the first adversarial benchmark built from healthcare conversations averaging over 20 turns. It combines a taxonomy of repetition (§ 2) and human-calibrated verifiers (§ 3.2). With these verifiers, we select challenging contexts to elicit repetition (§ 3) under realistic conditions. We present our pipeline in Figure 2. Benchmarking frontier models on REPETITIONBENCH shows that current LLMs have made progress on surface-level repetition, but still struggle with semantic repetition: they often vary their wording without meaningfully varying the content. This problem becomes especially apparent over longer conversations, where models must maintain diverse discourse across turns.

We summarize our contributions in four points:

1. **Taxonomy.** A 13-category taxonomy of conversational repetition across semantic, structural, and surface-level dimensions (§ 2). The taxonomy reflects service deployments in healthcare, where distinguishing helpful acknowledgment from redundancy is critical (Section 2).

2. **Verifiers.** Six deterministic and seven subjective verifiers for scalable repetition detection (§ 3.2). These LLM verifiers are designed to correlate strongly with human annotations.

3. **Adversarial benchmark.** REPETITIONBENCH, a 6,892-sample adversarial dataset, plus an evaluation of 13 frontier LLMs.

4. **Using REPETITIONBENCH as training signal.** Our verifier-based rewards reduce repetition during RL post-training (§ 4.2). This shows why evaluating repetition requires both dialogue history and determining when to repeat is appropriate.

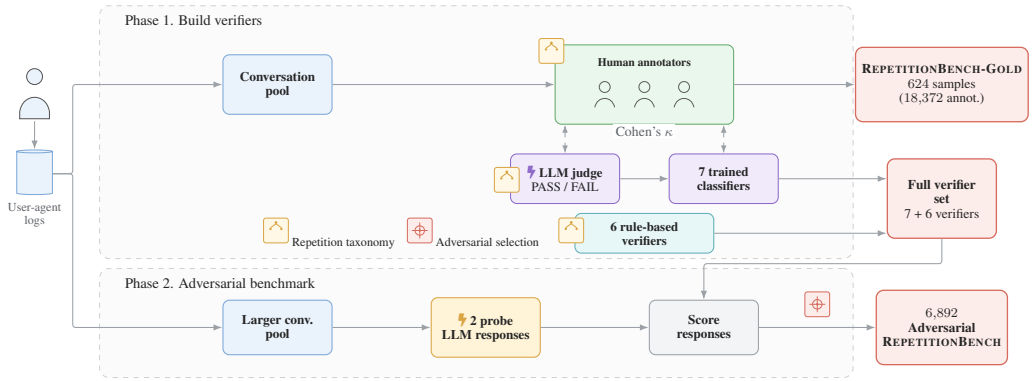


Figure 2: **Two-phase pipeline for constructing REPETITIONBENCH.** Phase 1 uses human annotations to develop and calibrate the repetition verifiers. Phase 2 applies the full verifier set to select adversarial conversation contexts for the final benchmark.

2 CONVERSATIONAL REPETITION TAXONOMY

Existing repetition literature provides n-gram overlap, semantic similarity, or lexical diversity as repetition measurement standard (Xi et al., 2021); computational taxonomies distinguish repetition within a response from repetition of earlier turns (Li et al., 2020); and linguistic work distinguishes repetition across linguistic levels and between self-repetition from repetition of another speaker's content (Tannen, 1987). These standards motivate the key parts of our taxonomy, but they do not fully capture the failures that arise in clinical, task-oriented, empathetic dialogue. In particular, existing measures often quantify repetition as a single score and cannot distinguish appropriate repetition, such as safety confirmation, from failures such as re-asking an already answered question.

We therefore extend prior works along two dimensions: the linguistic unit being repeated and the dialogue scope of repetition. The first captures word-, phrase-, semantic-, sentence-, and discourse-structure-level repetition; the second captures *who* repeats *whom* (Tannen, 1987). We also incorporate dialogue-based multi-turn system failures such as redundant questioning (Mai & Carson-Berndsen, 2023). Our taxonomy captures 13 diverse forms of repetitive behavior, as shown in Figure 1. Each type has explicit PASS/FAIL criteria, grouped into deterministic and subjective types (Figure 5).

2.1 DETERMINISTIC REPETITION TYPES

Deterministic repetition types use rule-based lexical matching over explicit patterns: consecutive token repetition (H), prefix match (I), full-turn repetition (J , L), and first-sentence repetition (K , M), as shown in Figure 5. Below, we describe each deterministic verifier by its scope and linguistic unit (Appendix E).

Templated openings: prefix match (I). Conversational agents typically develop habitual opening sentences that are reused across turns regardless of the context. This verifier extracts the first three words of the current response and checks whether they match the first three words of the immediately preceding agent turn, ignoring case. For example, if both the prior and current response open with *"I understand your. . . "*, the verifier returns FAIL. This verifier captures templated opening reuse without requiring semantic judgment.

Verbatim regurgitation across turns: full-turn repetition (L) and first-sentence repetition (M). A more severe failure mode is the agent reiterating a prior response verbatim. Verifier L compares the current response, after whitespace normalization, against earlier non-adjacent agent turns; the immediate prior turn is covered by J . Verifier M compares first sentences against those earlier turns and flags literal substring reuse after whitespace normalization; the immediate prior turn is covered by K . For example, if a transfer message such as *"I'll connect you to a nurse now."* appears in turn three and the agent re-emits the same message in turn nine, L returns FAIL.

Verifier M fails when an agent reuses an earlier opening sentence. These failures are strong signals of decoding collapse (Yao et al., 2025).

Other rule-based verifiers: consecutive token repetition (H). The remaining verifier targets loops within a single response, where the same token repeats consecutively. Verifier H returns FAIL when a single word or phrase (any sequence of 2–30 characters) appears five or more times consecutively, excluding cases such as digits, punctuation, whitespace, single letters, ordinals, or punctuation-only tokens (PASS). Although deterministic verifiers are fast and capture obvious repetitions, they are incomplete and must be combined with more advanced verifiers to penalize more subtle forms of repetition. We present these as subjective verifiers in the next section.

2.2 Subjective repetition types

Subjective repetition types capture patterns that depend on context rather than surface form: within-turn redundancy (A), user echoing (B), phrase reuse (C), redundant questions (D , F), and repeated statements (E , G), as shown in Figure 5. A key challenge is that repetition is not purely a surface-level phenomenon: similar phrasing may be appropriate (PASS) or problematic (FAIL) depending on whether the restatement supports empathy or confirmation. For example, phrases such as "I understand what you're going through," even when repeated, can be appropriate when they acknowledge the patient's emotional state.

For each subjective repetition type, we implement subjective verifiers that use LLM-as-a-judge (judge: GPT-5.5 (OpenAI, 2026)). To reflect the scope and linguistic units defined in the taxonomy, we structure each verifier prompt to take prior agent turn(s), the current response, and a binary PASS/FAIL decision criterion as inputs (Appendix O). For each repetition type, we specify exemptions (e.g., proper or compound nouns) that should be treated as acceptable (Appendix A).

Redundant questions (D , F). Rules can only detect lexically identical question repetition but cannot recognize semantic equivalence across paraphrases. Verifier D flags re-asking against the preceding agent turn. For example, given the prior turn *"What prescription medications are you currently taking?"* and a current response *"What are the medications you are actively taking?"*, the verifier identifies them as repeating queries with only trivial lexical variation and returns FAIL. An exact-match rule would not catch this case even though it is duplicative from the patient's perspective. The verifier F extends this comparison to all agent turns, capturing longer-range repetition. The verifier extracts questions from each relevant turn and assesses whether they are semantically equivalent.

Repeated statements (E , G). The agent restates sentences from prior turns without adding new information. Verifier E flags repetition against the immediately preceding agent turn, for example *"Thank you for confirming."* followed shortly by *"Thanks for your confirmation."*. The across-turn variant G verifier checks for semantic equivalence across earlier turns, returning FAIL when the current sentence restates prior content without advancing the conversation.

Phrase reuse (C). Verifier C extracts spans of four or more words shared between the current response and the preceding agent turn, then applies an explicit exemption list. For example, the span *"at four one, . . . , zero one two three"* recurring across turns is exempted as phone-number reuse, while a scaffolding phrase returns FAIL because it constitutes stylistic reuse that adds no new content (Figure 5). This exemption logic is the primary reason the verifier is LLM-judged rather than rule-based: character-level string equality would over-flag the exempt cases and penalize clinically appropriate behavior.

Other subjective types (A , B). The remaining subjective types target repetition inside a single response or against the user's turn. For example, for *"I can help. I can help you with that."*, the agent repeats itself in consecutive sentences which is not natural in human-human conversation (A). User echoing (B) occurs when the agent mirrors the patient's wording with minimal change (e.g., Agent: *"You feel dizzy."*): this is potentially linked to attention sinks (Yona et al., 2025) (Figure 5).

3 DATASET CONSTRUCTION

Building on the taxonomy of inappropriate repetition, we construct REPETITIONBENCH in three stages. We first create a REPETITIONBENCH-GOLD of nurse-annotated model responses to supervise and evaluate the subjective repetition verifiers. We then calibrate these verifiers against human judgments and combine them with the deterministic verifiers. Finally, we apply the full verifiers to fetch responses from two probe models and select contexts that elicit inappropriate repetition.

All contexts in our benchmark are from conversations generated by human nurses in the U.S., who role-play as real patients and interact with an AI agent over a phone call, across a diverse set of clinical scenarios, including colonoscopy preparation, medical appointment scheduling, post-discharge instruction sharing, medication adherence, care management and retention, device-support hotlines, etc. We sample distinct multi-turn *payloads* from patient–agent interaction logs. Each payload consists of: (i) a system prompt specifying the AI agent's role, the clinical task it needs to perform, the agent's communication guidelines, and the relevant patient context; (ii) a conversation history of $n \geq 20$ turns, containing the prior AI and patient turns; and (iii) the current patient turn to which the agent must respond. Throughout, we use *payload* and *context* interchangeably.

3.1 REPETITIONBENCH-GOLD FOR VERIFIER SUPERVISION

We construct a REPETITIONBENCH-GOLD to train and evaluate subjective verifiers against human judgment. We invite expert human annotators label each payload using guidelines based on our taxonomy (§ 2.2). U.S.-based registered nurses were recruited to label the dataset. For each payload, these *expert* annotators label two model responses across the seven subjective repetition types, by providing PASS/FAIL labels to each category (details in Appendix D). Each payload is paired with responses from two target models ($1{,}248$ responses total). The seven LLM-judged categories yield $8{,}736$ response–category items ($624 \times 2 \times 7$). Two primary annotators label each item independently, producing $17{,}472$ primary annotations. A more experienced third annotator resolves disagreements on 900 items (10.3% of all items), for a total of $18{,}372$ individual human annotations. We report inter-rater reliability (IRR) between the first two annotators using Cohen's κ (Cohen, 1960) and agreement rate (accuracy) (Artstein & Poesio, 2008) in Table 1,[3] yielding an overall κ=0.49 at 90.1% accuracy, moderate inter-annotator agreement that our best verifiers exceed (GPT-5.5 κ=0.64). Our REPETITIONBENCH-GOLD consists of 624 payloads drawn from 465 unique conversations spanning 16 healthcare call topics. The underlying conversations contain an average of 14.7 agent and 14.2 user turns ($p_{95} = 39$, max $= 118$ agent turns; Table 5).

3.2 HUMAN-CALIBRATED SUBJECTIVE REPETITION VERIFIERS

Finetuning subjective verifiers. We use the gold-labeled corpus to train and evaluate verifiers calibrated to human judgments. We provide two verifier families for the benchmark: an LLM-as-judge verifier and a fine-tuned Qwen3.5-27B Qwen Team (2026) checkpoint. The LLM-as-judge verifier uses GPT-5.5 with high reasoning effort, yielding a computationally expensive but simple-to-deploy verification method. The Qwen3.5-27B checkpoint is fine-tuned on the training split of the gold-labeled corpus with two gold-anchored synthetic-data recipes: (1) *SFT-CoT* uses CoTs generated by GPT-5.5 at high reasoning effort; their style and token distribution differ from Qwen's. (2) *SFT-StyleMatch* instead has Qwen rewrite each rationale conditioned on GPT-5.5's reasoning and the gold label, then trains on the Qwen-style traces. Both recipes balance PASS/FAIL labels across scenarios.[4]

Verifier–human agreement. Our best LLM-as-a-judge verifiers achieve high agreement with human judgment, as shown in Table 1: GPT-5.5's agreement with the gold labels exceeds measured human inter-rater reliability overall. The reported SFT-StyleMatch variant achieves high held-out agreement, indicating that both the GPT-5.5 zero-shot judge and the style-matched Qwen3.5-27B verifier are calibrated to human judgments.

[3]The auxiliary verifier to distinguish between appropriate and unnecessary repetition achieves Cohen's κ=0.64 against the golden labels (above the 0.61 high-agreement threshold).

[4]We compare full SFT with LoRA training (Hu et al., 2022) across multiple hyperparameter settings, selecting the best checkpoint based on validation accuracy. Training details, data balancing procedures, and hyperparameter sweeps are provided in Appendix F.

Table 1: Verifier agreement with human annotations on a balanced training subset of the REPETITIONBENCH-GOLD (left, N=1,461) and the held-out eval set (right, N=150). Here, N counts response–repetition type items, not distinct contexts. κ: Cohen's kappa; σ: standard deviation across 3 runs.

Scenario	Golden Training Set (N=1,461)				Golden Eval set (N=150)			
	GPT-5.5	Qwen 3.5-27B	Human IRR		GPT-5.5		Qwen 3.5 27B SFT	
	$\kappa \pm \sigma$	$\kappa \pm \sigma$	κ	Acc. (%)	$\kappa \pm \sigma$	Acc. (%)	$\kappa \pm \sigma$	Acc. (%)
A Within-turn redundancy	0.53 ± 0.02	$\mathbf{0.54 \pm 0.02}$	0.24	98.6	1.00 ± 0.00	100.0 ± 0.0	1.00 ± 0.00	100.0 ± 0.0
B User echoing	$\mathbf{0.56 \pm 0.01}$	0.47 ± 0.03	0.31	94.7	$\mathbf{0.67 \pm 0.00}$	83.3 ± 0.0	0.44 ± 0.19	72.2 ± 9.6
C Phrase reuse	$\mathbf{0.57 \pm 0.01}$	0.31 ± 0.04	0.23	80.8	0.26 ± 0.12	62.8 ± 5.9	$\mathbf{0.46 \pm 0.13}$	73.1 ± 6.7
D Redundant questions	$\mathbf{0.73 \pm 0.00}$	0.68 ± 0.00	0.64	91.6	0.73 ± 0.04	86.4 ± 2.1	$\mathbf{0.85 \pm 0.00}$	92.6 ± 0.0
E Repeated statements	0.61 ± 0.03	$\mathbf{0.63 \pm 0.02}$	0.58	92.3	1.00 ± 0.00	100.0 ± 0.0	1.00 ± 0.00	100.0 ± 0.0
F Redundant questions (history)	$\mathbf{0.65 \pm 0.01}$	0.64 ± 0.00	0.55	86.6	0.55 ± 0.00	78.4 ± 0.0	$\mathbf{0.84 \pm 0.05}$	91.9 ± 2.7
G Repeated statements (history)	$\mathbf{0.57 \pm 0.01}$	0.52 ± 0.03	0.46	86.3	0.68 ± 0.08	84.0 ± 4.0	$\mathbf{0.73 \pm 0.12}$	86.7 ± 6.1
OVERALL (sample-wise)	$\mathbf{0.64 \pm 0.01}$	0.57 ± 0.00	0.49	90.1	0.65 ± 0.01	82.4 ± 0.0	$\mathbf{0.77 \pm 0.04}$	88.7 ± 0.0

Verifier bias. Under the same prompts on adjudicated gold labels, the reported overall Cohen's κ was 0.64 for GPT-5.5, compared with 0.36 for GLM-5.2, 0.24 for Kimi-K2.6, and 0.21 for DeepSeek-V4-Pro. The human–human agreement was 0.49. GPT-5.5 was therefore the only tested zero-shot judge to exceed measured human agreement on this set; this comparison motivates its selection rather than establishing that it is unbiased. The GPT-5.5 result was stable across three runs (Table 1). On the separate held-out 150-item evaluation set, the fine-tuned Qwen3.5-27B verifier reached $\kappa = 0.77$, compared with 0.65 for GPT-5.5 under the same evaluation. These are verifier–human agreement results, distinct from the full-benchmark model rankings in § 4.1.

3.3 REPETITIONBENCH CONSTRUCTION

Hard and easy splits. Having calibrated the subjective verifiers against human judgments, we use them alongside the deterministic verifiers to construct REPETITIONBENCH. We used 2,723 phone-based healthcare conversations and generated responses with two probe models, GPT-4o (OpenAI, 2024) and DeepSeek V3.1 (DeepSeek-AI, 2025). If either response is flagged for repetition, we regarded them as adversarial and included those payloads in our benchmark.[5] As a result, REPETITIONBENCH contains 6,892 payloads spanning the same 16 healthcare topics as the REPETITIONBENCH-GOLD (Table 6). The conversations in these payloads average 22.2 agent and 21.8 user turns.[6] (Table 5). Among these, 2,203 (32.0%) are *hard* split where both probe models trigger at least one verifier, though not necessarily the same one and the remaining 4,689 (68.0%) are *easy* split where only one probe model triggers a verifier.

Repetition patterns by difficulty splits. Across both tiers, the most frequent flags are repeated statements (G) and first sentences (M) from earlier agent turns, followed by phrase reuse (C); redundant questions (D) and consecutive-token loops (H) are rarer (Figure 3a; Table 7). Most verifier pairs have Jaccard similarity below 0.21 (Jaccard, 1901), suggesting distinct failure modes are introduced in our taxonomy (Figure 3c; Appendix I). We also measure each payload's *breadth*,[7] the number of distinct verifiers that flag either model's response. Payloads that trigger more verifier types are more likely to flag both probe models (14% at breadth one $\rightarrow$ 80% at breadth six).

4 EXPERIMENTS

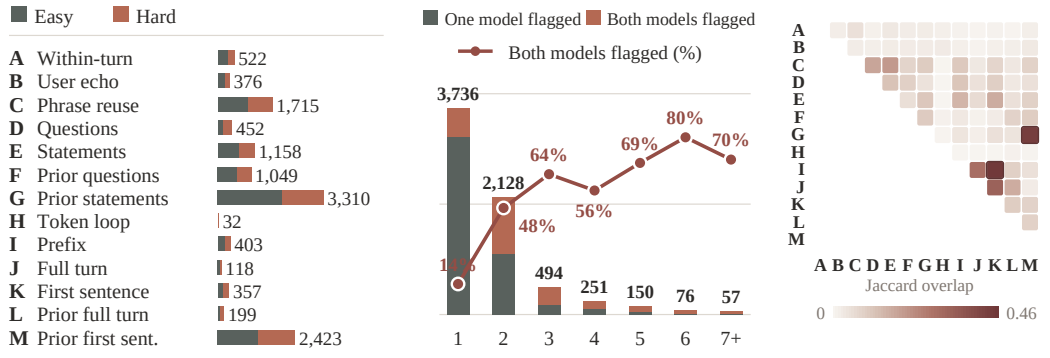
4.1 BENCHMARKING FRONTIER MODELS

We evaluate 13 frontier LLMs on REPETITIONBENCH: Claude Opus 4.7, Claude Opus 4.7 + Thinking, and Claude Haiku 4.5 Anthropic (2026); GPT-5.5, GPT-5.5 + Thinking, and GPT-5.4 mini OpenAI

[5]We use these earlier versions as probes so that none of the 13 frontier models evaluated later, including newer GPT and DeepSeek models, determines which payloads are included the benchmark.

[6]About 95% have at most 55 agent turns, while the longest has 132 agent turns.

[7]The measures can diverge: a payload can be hard at breadth one if the same verifier flags both models, or easy at breadth six if all six verifiers flag only one model.



(a) Verifier counts. (b) Breadth vs. both-model flags (c) Pairwise overlap.

Figure 3: **Verifier-level structure of REPETITIONBENCH.** (a) Number of payloads flagged by each verifier, split by easy and hard tier, and a context can trigger several verifiers; (b) Stacked payload counts by breadth (number of distinct verifiers triggered): the light segment is flagged on one target model and the dark segment on both. The line gives the percentage flagged on both. Payloads that trigger more verifier types are more likely to flag both probe models; (c) Upper-triangle Jaccard similarity of flagged-payload sets. Distinct failure modes are introduced in our taxonomy.

(2026); Gemini 3.1 Pro, Gemini 3.1 Pro + Thinking, and Gemini 2.5 Flash Google DeepMind (2026); GLM-5.1 and GLM-5.1 + Thinking GLM-5-Team (2026); Kimi K2.6 Moonshot AI (2026); and DeepSeek-V4-Pro DeepSeek-AI (2026). We apply the 13 verifiers (§3.2) to every model response. Our primary metric is the ANY-failure rate: the fraction of responses flagged by at least one verifier in a group. We report it for all 13 verifiers, the seven subjective verifiers, and the six deterministic verifiers, counting each response once per group.

Among the 13 frontier models, hard-tier ANY_{all} ranges from 44.61% for GPT 5.5 +T to 76.86% for DeepSeek-V4-Pro (Figure 4a). For every frontier model, subjective verifiers flag 1.7–$2.2\times$ as many hard-tier responses as deterministic verifiers (Figure 4b). Depending on the model, 21–35% of all hard-tier responses pass every deterministic verifier but trigger at least one subjective verifier. These *subjective verifier-triggering* responses account for 44–54% of hard-tier responses flagged by any verifier, calculated as $(\text{ANY}_{\text{all}} - \text{ANY}_{\text{det}})/\text{ANY}_{\text{all}}$. This suggests that deterministic checks alone miss a substantial share of potential repetition on hard payloads. Moreover, the per-verifier results also show that across-turn restatements are more common than within-response repetition (Table 11).

4.2 USING REPETITIONBENCH VERIFIERS FOR POST-TRAINING

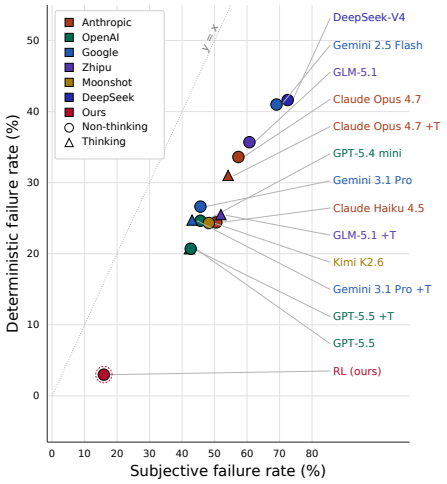
RL model performance We test whether REPETITIONBENCH verifiers provide a useful RL signal by post-training DeepSeek-V3.1 with Group Relative Policy Optimization (GRPO) (Shao et al., 2024) on healthcare dialogue prompts that were not selected to elicit repetition. The composite reward assigns a 15% weight to the verifier-derived repetition signal and 85% to task-level dialogue quality. A FAIL from any of three deterministic verifiers (e.g., Full turn repetition J, Full turn repetition (history) L, or Consecutive token repetition H) sets the sampled response's reward to zero.

Among 14 models evaluated on REPETITIONBENCH, our RL model has the lowest aggregate ANY-failure rates: 12.55% on easy and 17.75% on hard splits (Figure 4a). Subjective ANY-failure is 11.76% and 15.94% on the two tiers, compared with 1.28% and 2.99% for deterministic ANY-failure. To assess whether this lead depends on the particular mix of payloads, we recompute model rankings across $10{,}000$ paired bootstrap resamples. The RL model ranks first on both tiers in every resample. On each tier, its ANY_{all} is lower than every baseline's (Holm-adjusted McNemar $p < 0.001$ for each comparison. See Appendix K).

To examine where repetition failures occur, we fit an multivariable regression over $96{,}488$ responses (14 models on $6{,}892$ contexts each), including task type, difficulty, dialogue state, dialogue turn count, context length, model family, response length, and reasoning mode. Dialogue state, especially multiple user clarification requests or repeated discussion of the same topic, was the strongest

Model	Subjective		Deterministic		All	
	Easy	Hard	Easy	Hard	Easy	Hard
Claude Opus 4.7	35.28	57.34	15.16	33.61	37.76	61.41
Claude Opus 4.7 +T	32.66	54.17	14.74	30.93	35.11	58.47
Claude Haiku 4.5	30.64	50.54	9.27	24.41	31.64	53.35
GPT-5.5	28.39	42.66	9.31	20.70	29.41	45.11
GPT-5.5 +T	28.56	42.26	9.67	20.61	29.41	44.61
GPT-5.4 mini	28.96	45.65	11.10	24.64	30.28	48.10
Gemini 3.1 Pro	27.68	45.65	11.27	26.63	29.05	48.78
Gemini 3.1 Pro +T	26.37	43.07	10.82	24.64	27.73	45.92
Gemini 2.5 Flash	47.37	69.07	19.99	40.99	49.73	72.64
GLM-5.1	38.38	60.73	16.80	35.69	40.65	64.58
GLM-5.1 +T	33.98	51.95	11.72	25.41	35.70	54.76
Kimi K2.6	29.24	48.23	10.87	24.32	30.98	50.95
DeepSeek-V4-Pro	52.69	72.51	20.24	41.62	55.09	76.86
RL (ours)	**11.76**	**15.94**	**1.28**	**2.99**	**12.55**	**17.75**
Mean (14)	*32.28*	*49.98*	*12.30*	*26.94*	*33.93*	*53.09*

(a) Aggregate ANY-failure rates (%) by verifier group and difficulty. Lower is better.



(b) Hard-tier subjective versus deterministic ANY-failure rates for the 13 frontier models.

Figure 4: Verifier-group ANY-failure rates on REPETITIONBENCH.

reported factor. Dialogue-state features correlated with deterministic and subjective repetition flags (Spearman's $\rho = 0.41$ and 0.34, respectively; Appendix L). We then tested whether the RL model's lower failure rate persisted across decoding choices. We evaluated RL and DeepSeek-V4-Pro on 200 REPETITIONBENCH-GOLD contexts sampled evenly across payload lengths. We used four temperature/top-p combinations: $(0, 1)$, $(0.3, 1)$, $(0.7, 1)$, and $(0.7, 0.8)$. The three nonzero-temperature combinations were each run with two seeds, giving seven runs per model. At top-$p = 1$, DeepSeek-V4-Pro's ANY_{all} failure rate, averaged across the two seeds, fell from 49.5% at temperature 0.3 to 39.0% at 0.7. At temperature 0.7, it rose to 48.8% when top-p was lowered to 0.8. The RL model's rate ranged from 10.5% to 17.0% and was lower in all seven runs (Figure 7; Appendix M).

Reward-weight ablation. To assess how the repetition reward affects other capabilities of the model, we train separate checkpoints of our RL model with repetition-reward weights of 0%, 15%, and 30%. We evaluate unnecessary and appropriate repetition, empathy/dialogue, clinical correctness, task adherence, and instruction following on IFBench (Pyatkin et al., 2025). Increasing the reward roughly halved unnecessary repetition (59.7% → 30.7%) between the 0% and 30% checkpoints without reducing empathy (90.8%), clinical confirmation (42.4%), or task adherence (80.5%). Performance on the unrelated IFBench benchmark also remained stable (36.5% for the base model vs. all RL checkpoints). This suggests that that optimizing for repetition does not come at the cost of other general capabilities of the model.

Table 2: Reward-weight ablation.

Metric	Base	0%	15%	30%
Empathy/dialogue	87.0	90.8	86.3	90.8
Clinical correctness	48.0	40.0	37.2	42.4
Task adherence	81.0	80.0	78.8	80.5
Unnecessary rep. ↓	58.6	59.7	46.8	30.7
Appropriate rep. ↑	58.4	58.4	51.9	49.4
IFBench ↑	36.5	37.1	37.9	37.4
SODA-Eval ↓	2.36	2.88	2.29	2.07

Note: Base is DeepSeek-V3.1.

Cross-domain generalization. To test whether the reduction in repetition extends beyond our benchmark family, we compare the RL model with its DeepSeek-V3.1 base on the public SODA-Eval benchmark (Mendonça et al., 2024). SODA-Eval differs from our training setup in domain (open-domain social dialogue), repetition taxonomy, and judges. We select the 271 SODA-Eval contexts whose original benchmark response was labeled repetitive, then remove that response. Each model generates five new responses for every context. Repetition rate falls from 2.36% for the base to 2.07% for the RL model with a 30% repetition-reward weight (Appendix N). Mean judge-rated response quality rises by about 5%. Together, these results are consistent with transfer beyond our benchmark family and its reward signal (Appendix N).

Table 3: **Comparison with repetition and multi-turn benchmarks.** REPETITIONBENCH is the only clinical, multi-turn benchmark covering all listed properties.

Benchmark	Repetition focus	Multi-dimensional	Automatic scoring	Adversarial construction	Domain	Size	Turns
ACUTE-EVAL (Li et al., 2019)	✓	✗	✗	✗	Open-domain	—	—
CodeRepetEval (Dong et al., 2025)	✓	✓	✓	✗	Coding	2,048	1
Liu et al. (2025)	✓	✓	✓	✗	Coding	1,299	1
Multi-turn							
MT-Bench (Zheng et al., 2023)	✗	✗	✓	✗	General	80	1
MT-Bench-101 (Bai et al., 2024)	✗	✓	✓	✗	General	1,388*	∼3
MultiChallenge (Sirdeshmukh et al., 2025)	✗	✓	✓	✗	General	273	∼5
EvolIF(Jia et al., 2026)	✗	✓	✓	✓†	General	varies	varies
REPETITIONBENCH (ours)	✓	✓	✓	✓	Healthcare	6,892	23

5 RELATED WORK

Dialogue research has often treated repetition as one dimension of quality. PersonaChat and ConvAI2 provide multi-turn settings, and ACUTE-EVAL measures overall dialogue preference (Zhang et al., 2018; Dinan et al., 2019; Li et al., 2019); reducing repetition can improve engagement (See et al., 2019). More targeted work uses unlikelihood training and cross-turn RL rewards (Li et al., 2020; Shi et al., 2021), studies redundant questions after users have already supplied answers (Mai & Carson-Berndsen, 2023), or controls copying and generation loops through decoding (Xi et al., 2021; Holtzman et al., 2020; Welleck et al., 2020). Production-LLM repetition has also been categorized by pattern (Wang et al., 2025). Repeating prompts can change model answers unpredictably (Shaier et al., 2024), underscoring the need to distinguish user repetition from agent repetition. In medical pre-consultation, repetitive questions may acquire no new diagnostic information (Lim et al., 2025); users of a mental-health support app likewise described redundant replies as frustrating (Chaudhry & Debi, 2024). This motivates distinguishing inappropriate repetition from useful confirmation.

Prior works cover related capabilities without reporting repetition by failure type. MT-Bench, MT-Bench-101, MultiChallenge, and AgentBoard evaluate judging, instruction following, coherence, or task progress (Zheng et al., 2023; Bai et al., 2024; Sirdeshmukh et al., 2025; Ma et al., 2024); multi-turn performance remains challenging, and repetition has been identified as an understudied axis (Laban et al., 2025; Li et al., 2025). Our data construction follows model-in-the-loop adversarial filtering (Nie et al., 2020; Kiela et al., 2021; Bartolo et al., 2020), using verifier agreement across two target models as a difficulty signal.

Aggregate metrics cannot establish whether repetition is inappropriate in context. Distinct-n measures surface diversity (Li et al., 2016); MAUVE and BERTScore summarize distributional quality and semantic similarity (Pillutla et al., 2021; Zhang et al., 2020). Conversational repetition can support coherence (Tannen, 1987), while automatic dialogue metrics do not always align with human judgments (Li et al., 2021). REPETITIONBENCH therefore reports category-level failure rates from verifiers evaluated against human annotations.

6 CONCLUSION

In REPETITIONBENCH, we introduced an adversarial benchmark and evaluation framework for repetition in multi-turn conversational agents, grounded in a 13-category taxonomy, human-calibrated verifiers, and realistic healthcare interactions. Our results show that frontier LLMs increasingly avoid surface-level repetition, but still struggle with semantic repetition in longer conversations. This gap matters because a paraphrased response can appear fluent, yet, forcing the user back through an already resolved exchange. Our taxonomy and verifiers make that distinction measurable, separating unnecessary semantic repetition from useful restatement. As the first fine-grained, scalable framework for evaluating repetition in multi-turn agents, REPETITIONBENCH allows for pre-analysis before real-world deployment. This benchmark remains limited to turn-level responses from U.S. English, phone-based healthcare conversations and therefore cannot assess task completion, user satisfaction, clinical accuracy, recovery after feedback, or clinical safety. Future work will evaluate full conversations in broader settings, refine the reward with appropriate-repetition cases, and explore internal model interventions that reduce semantic repetition while preserving useful restatement.

AI USE STATEMENT

Language models were used to generate responses for benchmark construction and evaluation: GPT-4o and DeepSeek-V3.1 probed candidate contexts, and 13 frontier models supplied evaluated responses. GPT-5.5 judged the seven subjective repetition categories and generated training rationales; Qwen3.5-27B rewrote rationales for verifier training. DeepSeek-V3.1 was post-trained with verifier-guided rewards. OpenAI Codex helped revise plot and prose of the manuscript. The authors retain responsibility for checking AI-assisted content and for the final manuscript and results.

ETHICS STATEMENT

Synthetic conversations and contributor care. REPETITIONBENCH uses phone conversations in which U.S.-based registered nurses role-played patients interacting with an AI agent; no real patient encounters or protected health information were collected. Scenarios cover routine topics such as appointment scheduling, post-discharge follow-up, and medication adherence without referring to identifiable individuals. The released artifact contains conversation text, repetition labels, and verifier outputs, but no demographic, biometric, or contact information about the nurses. We screened incidentally produced names, phone numbers, and addresses and replaced them with synthetic placeholders. Annotators were registered nurses with clinical experience or experts trained on the repetition guidelines. They were compensated at rates consistent with their professional skill, reviewed no distressing material beyond routine clinical scenarios, and could decline tasks without penalty. The two-stage annotation and conflict-resolution protocol is documented in Appendix D.

Interpreting benchmark scores. The benchmark measures one failure mode: inappropriate repetition across linguistic dimensions. A low failure rate does not establish clinical accuracy, empathy, or safety for patient-facing deployment. Scores should complement clinical safety review, end-to-end task evaluation, and post-deployment monitoring, rather than stand alone as evidence of deployment readiness.

Scope and judge bias. The corpus is U.S. English, phone-based, and drawn from a fixed set of healthcare topics (Table 6). Model rankings and verifier behavior may not transfer to other languages, dialects, modalities, or domains. LLM-as-judge verifiers may inherit bias from the judge model; we calibrate them against human gold labels and also provide six rule-based verifiers.

Release and compute costs. The dataset is linked in the abstract. The annotation protocol, including PASS/FAIL rules and appropriate-repetition cases, appears in Appendix D; verifier prompts, exemptions, and rule-based pseudocode appear in Appendix O. We plan to release executable verifier code, standalone guidelines, and adjudication rules with the final artifact so flagged behavior can be inspected and reproduced. The current manuscript alone lacks the response-level data needed to recompute every bootstrap interval or exploratory analysis. Construction, training, and evaluation required inference across 13 LLMs and seven LLM-as-judge verifiers per sample. Estimated GPU-hours appear in Appendix H, F.4, and G.3, allowing users to consider partial-benchmark or rule-based-only alternatives.

REFERENCES

Anthropic. Introducing claude opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, 2026.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. URL https://doi.org/10.1162/coli.07-034-R2.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. MT-Bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7421–7454, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL https://arxiv.org/abs/2402.14762.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678, 2020. URL https://aclanthology.org/2020.tacl-1.43/.

Timothy Bickmore, Daniel Schulman, and Langxuan Yin. Maintaining engagement in long-term interventions with relational agents. *Applied Artificial Intelligence*, 24(6):648–666, 2010. doi: 10.1080/08839514.2010.492259. URL https://doi.org/10.1080/08839514.2010.492259.

Beenish Moalla Chaudhry and Happy Rani Debi. User perceptions and experiences of an AI-driven conversational agent for mental health support. *mHealth*, 10:22, 2024. doi: 10.21037/mhealth-23-55. URL https://mhealth.amegroups.org/article/view/126211/html.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2025. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro, 2026.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1):755–810, 2021. doi: 10.1007/s10462-020-09866-x. URL https://doi.org/10.1007/s10462-020-09866-x.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W. Black, Alexander Rudnicky, Jason Williams, Joelle Pineau, Mikhail Burtsev, and Jason Weston. The second conversational intelligence challenge (ConvAI2), 2019. URL https://arxiv.org/abs/1902.00098.

Yihong Dong, Yuchen Liu, Xue Jiang, Bin Gu, Zhi Jin, and Ge Li. Rethinking repetition problems of LLMs in code generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. URL https://aclanthology.org/2025.acl-long.48/.

Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): A randomized controlled trial. *JMIR Mental Health*, 4(2):e19, 2017. doi: 10.2196/mental.7785. URL https://mental.jmir.org/2017/2/e19/.

GLM-5-Team. Glm-5: from vibe coding to agentic engineering. https://arxiv.org/abs/2602.15763, 2026.

Google DeepMind. Gemini 3.1 pro. https://deepmind.google/models/gemini/pro/, 2026.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Paul Jaccard. étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

Qi Jia, Ye Shen, Xiujie Song, Kaiwei Zhang, Shibo Wang, Dun Pei, Xiangyang Zhu, and Guangtao Zhai. One battle after another: Probing llms' limits on multi-turn instruction following with a benchmark evolving framework, 2026. URL https://arxiv.org/abs/2511.03508.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 4110–4124. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.naacl-main.324/.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*, 2025. URL https://arxiv.org/abs/2505.06120. Accepted at ICLR 2026.

Liliana Laranjo, Adam G. Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica Chen, Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie Y. S. Lau, and Enrico Coiera. Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association*, 25(9):1248–1258, 2018. doi: 10.1093/jamia/ocy072. URL https://doi.org/10.1093/jamia/ocy072.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014/.

Margaret Li, Jason Weston, and Stephen Roller. ACUTE-EVAL: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*, 2019. URL https://arxiv.org/abs/1909.03087.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4715–4728, Online, 2020. Association for Computational Linguistics. URL https://arxiv.org/abs/1911.03860.

Xinmeng Li, Wansen Wu, Long Qin, and Quanjun Yin. How to evaluate your dialogue models: A review of approaches. *arXiv preprint arXiv:2108.01369*, 2021. URL https://arxiv.org/abs/2108.01369.

Yubo Li, Xiaobin Shen, Xinyu Yao, Xueying Ding, Yidi Miao, Ramayya Krishnan, and Rema Padman. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv preprint arXiv:2504.04717*, 2025. URL https://arxiv.org/abs/2504.04717.

Seungseop Lim, Gibaeg Kim, Wooseok Han, Jean Seo, Hyunkyung Lee, Jaehyo Yoo, and Eunho Yang. Format inertia: A failure mechanism of LLMs in medical pre-consultation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1437–1450, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-industry.101. URL https://aclanthology.org/2025.emnlp-industry.101/.

Mingwei Liu, Juntao Li, Ying Wang, Xueying Du, Zuoyu Ou, Qiuyuan Chen, Bingxu An, Zhao Wei, Yong Xu, Fangming Zou, Xin Peng, and Yiling Lou. Code copycat conundrum: Demystifying repetition in LLM-based code generation. *arXiv preprint arXiv:2504.12608*, 2025. URL https://arxiv.org/abs/2504.12608.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2401.13178.

Long Mai and Julie Carson-Berndsen. I already said that! degenerating redundant questions in open-domain dialogue systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 226–236, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-srw.33. URL https://aclanthology.org/2023.acl-srw.33/.

Shikib Mehri and Maxine Eskenazi. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 681–707, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.64. URL https://aclanthology.org/2020.acl-main.64/.

John Mendonça, Isabel Trancoso, and Alon Lavie. Soda-Eval: Open-domain dialogue evaluation in the age of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11687–11708, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.684. URL https://aclanthology.org/2024.findings-emnlp.684/.

Adam S. Miner, Arnold Milstein, and Jefferey T. Hancock. Talking to machines about personal mental health problems. *JAMA*, 318(13):1217–1218, 2017. doi: 10.1001/jama.2017.14151. URL https://doi.org/10.1001/jama.2017.14151.

Moonshot AI. Kimi k2.6. https://huggingface.co/moonshotai/Kimi-K2.6, 2026.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.acl-main.441/.

OpenAI. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. URL https://arxiv.org/abs/2410.21276.

OpenAI. Openai api documentation. https://developers.openai.com/api/docs/guides/latest-model, 2026.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021. URL https://arxiv.org/abs/2102.01454.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1645. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/46499a0622ecf568b72d17b61e45dbd5-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1702–1723, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1170. URL https://aclanthology.org/N19-1170/.

Sagi Shaier, Mario Sanz-Guerrero, and Katharina von der Wense. Asking again and again: Exploring LLM robustness to repeated questions. *arXiv preprint arXiv:2412.07923*, 2024. URL https://arxiv.org/abs/2412.07923.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Weiyan Shi, Yu Li, Saurav Sahay, and Zhou Yu. Refine and imitate: Reducing repetition and inconsistency in persuasion dialogues via reinforcement learning and human demonstration. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3478–3492, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. URL https://arxiv.org/abs/2012.15375.

Ved Sirdeshmukh, Kaustubh Deshpande, Johannes Mols, Lifeng Jin, Ed-Yeremai Cardona, Dean Lee, Jeremy Kritz, Willow Primack, Summer Yue, and Chen Xing. MultiChallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18632–18702, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://arxiv.org/abs/2501.17399.

Anthony L. Suchman, Kathryn Markakis, Howard B. Beckman, and Richard Frankel. A model of empathic communication in the medical interview. *JAMA*, 277(8):678–682, 1997. doi: 10.1001/jama.1997.03540320082047. URL https://doi.org/10.1001/jama.1997.03540320082047.

Jason Talevski, Anna Wong Shee, Bodil Rasmussen, Georgie Kemp, and Alison Beauchamp. Teach-back: A systematic review of implementation and impacts. *PLOS ONE*, 15(4):e0231350, 2020. doi: 10.1371/journal.pone.0231350. URL https://doi.org/10.1371/journal.pone.0231350.

Deborah Tannen. Repetition in conversation: Toward a poetics of talk. *Language*, 63(3):574–605, 1987. doi: 10.2307/415006. URL https://doi.org/10.2307/415006.

Weiwei Wang, Weijie Zou, and Jiyong Min. Solving LLM repetition problem in production: A comprehensive study of multiple solutions. *arXiv preprint arXiv:2512.04419*, 2025. URL https://arxiv.org/abs/2512.04419.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJeYe0NtvH.

Yadong Xi, Jiashu Pu, and Xiaoxi Mao. Taming repetition in dialogue generation, 2021. URL https://arxiv.org/abs/2112.08657.

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. Understanding the repeat curse in large language models from a feature perspective. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL https://arxiv.org/abs/2504.14218.

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. A comprehensive assessment of dialog evaluation metrics. In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pp. 15–33, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eancs-1.3. URL https://aclanthology.org/2021.eancs-1.3/.

Itay Yona, Ilia Shumailov, Jamie Hayes, Federico Barbero, and Yossi Gandelsman. Interpreting the repeated token phenomenon in large language models. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.08908.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2204–2213, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL https://aclanthology.org/P18-1205/.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

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. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.05685.

APPENDIX

A REPETITION TAXONOMY

Figure 5 gives the scope, definition, and example pattern for each of the 13 repetition categories.

B DATASET STATISTICS

Table 4 lists the samples, distinct conversations, and verifier scenarios in REPETITIONBENCH-GOLD and REPETITIONBENCH. Table 5 gives turn and token-count distributions; Table 6 gives the healthcare call topics. Both datasets come from nurse role-play calls with a clinical agent.

Table 4: Dataset overview. The 4,368 REPETITIONBENCH-GOLD category records come from 624 turn-level samples in 465 conversations; each record contains labels for two candidate model responses, yielding 8,736 response–category items. Appropriate-repetition controls are listed separately and are not part of the seven-category REPETITIONBENCH-GOLD count. Benchmark rows count turn-level contexts and are broken down by the two-target-model difficulty split. # conversations counts distinct `call_id` values within each group.

Group	# rows	# conversations	# scenarios
REPETITIONBENCH-GOLD			
total	4,368	465	7
Auxiliary appropriate-repetition controls			
total	78	73	1
REPETITIONBENCH			
easy	4,689	2,350	13
hard	2,203	1,187	13
total	6,892	2,723	13

Table 5: Per-utterance distribution of conversation turn counts and total payload token counts. REPETITIONBENCH-GOLD statistics cover the seven-category core; appropriate-repetition controls are shown separately. Token counts use the BPE encoding applied to the full payload.

Group	Quantity	mean	std	median	p95	p99	max
REPETITIONBENCH-GOLD							
core	Agent turns	14.7	14.0	12	39	67	118
	User turns	14.2	13.9	12	38	67	116
	Total tokens	2,939.7	1,237.1	2,762	5,084	7,524	9,885
Auxiliary appropriate-repetition controls							
controls	Agent turns	7.8	8.9	6	25	31	31
	User turns	7.5	8.5	4	24	30	30
	Total tokens	4,615.1	2,410.9	3,797	9,067	9,864	10,549
REPETITIONBENCH							
easy	Agent turns	21.2	15.5	17	51	91	132
	User turns	20.8	15.5	17	51	91	131
	Total tokens	2,293.1	753.0	2,150	3,706	5,113	9,553
hard	Agent turns	24.4	18.0	18	64	92	124
	User turns	24.0	18.0	18	63	92	124
	Total tokens	2,644.0	1,197.5	2,281	5,219	7,213	11,246
total	Agent turns	22.2	16.4	18	55	91	132
	User turns	21.8	16.4	17	55	91	131
	Total tokens	2,405.2	933.1	2,183	4,179	6,181	11,246

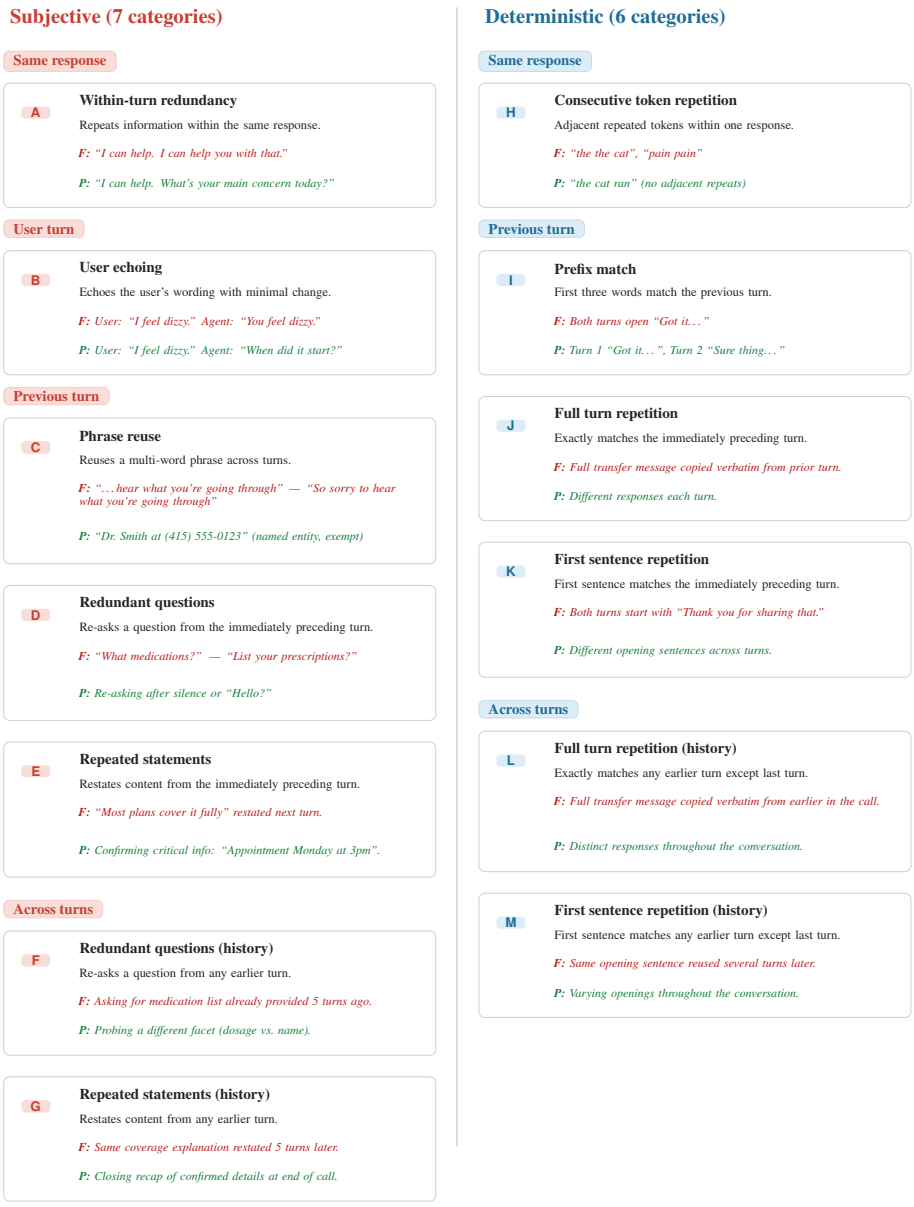


Figure 5: Granular taxonomy of inappropriate repetition categories. **Subjective** and **deterministic** types are both organized by scope of comparison. Each category shows a **F** example and a **P** example (**F**=FAIL, **P**=PASS), either because the response does not violate the FAIL criterion or because the repetition is contextually appropriate.

Table 6: Topic distribution per group. U = utterances, C = unique conversations. The REPETITIONBENCH-GOLD core and auxiliary appropriate-repetition controls are reported separately. We report the five largest topics individually and aggregate the remainder into *Other*; the full per-topic breakdown is withheld to protect proprietary deployment information.

Topic	**REPETITIONBENCH-GOLD**		**Auxiliary controls**	
	U	C	U	C
Pre-procedure preparation	64	48	7	6
Post-discharge follow-up	78	64	4	4
Medication adherence and refill	56	40	7	7
Care management and retention	70	50	7	7
Device support hotline	66	31	6	3
Other	290	232	47	46
Total	**624**	**465**	**78**	**73**

Topic	**REPETITIONBENCH**					
	easy		hard		total	
	U	C	U	C	U	C
Pre-procedure preparation	521	239	303	155	824	278
Post-discharge follow-up	584	327	212	139	796	378
Medication adherence and refill	354	212	170	101	524	243
Care management and retention	534	283	247	132	781	327
Device support hotline	306	89	215	71	521	108
Other	2,390	1,200	1,056	589	3,446	1,389
Total	**4,689**	**2,350**	**2,203**	**1,187**	**6,892**	**2,723**

Table 7: Verifier fires in REPETITIONBENCH by probe model and difficulty tier. GPT-only and DS-only mean that only GPT-4o or DeepSeek-V3.1, respectively, triggered the verifier; Both means both models did. A turn can contribute to several verifier rows.

Verifier		**Failure attribution**				**Difficulty tier**	
		GPT-only	Both	DS-only	Total	Hard	Easy
A	Within-turn redundancy	111	64	347	522	188	334
B	User echoing	160	46	170	376	138	238
C	Phrase reuse	700	388	627	1,715	757	958
D	Redundant questions	148	126	178	452	269	183
E	Repeated statements	478	208	472	1,158	477	681
F	Redundant questions (hist.)	429	192	428	1,049	446	603
G	Repeated statements (hist.)	1,087	849	1,374	3,310	1,305	2,005
H	Consecutive token repetition	4	12	16	32	15	17
I	Prefix match	151	74	178	403	186	217
J	Full turn repetition	40	20	58	118	53	65
K	First sentence repetition	123	63	171	357	171	186
L	Full turn repetition (hist.)	57	43	99	199	111	88
M	First sentence rep. (hist.)	632	721	1,070	2,423	1,140	1,283
Total verifier-fires		4,120	2,806	5,188	12,114	5,256	6,858
Unique adversarial turns			**6,892**			**2,203**	**4,689**
as % of corpus			100.0%			32.0%	68.0%

C BENCHMARK VERIFIER FLAGS

Volume. Repeated statements from earlier turns (G) account for 3,310 fires (27.3% of the total), followed by first-sentence repetition from earlier turns (M , 2,423) and phrase reuse (C , 1,715).

Table 8: Comparison modes used by the seven repetition dimensions.

Mode	Compared against	Unit
Across agent turns	Immediately preceding agent utterance	Statement, question, or span
Across agent & patient	The patient's last turn	Whole response
Single-turn	The same agent response	Sentence, word, or phrase
Across all previous turns	All prior agent utterances *except* the immediately preceding one	Statement or question

Together they account for 61.5% of the 12,114 fires. Consecutive-token repetition (H) is least frequent, with 32 fires.

Model attribution. DeepSeek-V3.1 produces more verifier-flagged responses than GPT-4o on 10 of the 13 verifiers, including all six deterministic types (H – M). The largest DS-only versus GPT-only gaps occur for first-sentence repetition from history (M , 1,070 vs. 632), repeated statements from history (G , 1,374 vs. 1,087), and within-turn redundancy (A , 347 vs. 111). GPT-4o has more model-specific phrase reuse failures (C , GPT-only 700 vs. DS-only 627); its small leads on E and F are six and one fire, respectively. Same-verifier cross-model agreement (the Both column) is highest on G (849) and M (721).

Tier composition per verifier. Hard-tier turns make up 32.0% of the benchmark but account for 43.4% of verifier fires (5,256/12,114). Of the 13 verifiers, only redundant questions (D) and full-turn repetition from history (L) fire more often on hard turns than on easy ones.

D HUMAN ANNOTATION PROTOCOL

The nurse annotators used the following rules to judge each candidate response.

D.1 STAGE 1: SHOULD THE AGENT REPEAT?

For each task, the annotator first reads the final patient turn (and the prior context if it is ambiguous) and sets the column `agent_SHOULD_repeat` to TRUE if any of the following hold:

- The patient does not respond.
- The agent checks for presence, e.g., "Are you still there?".
- The patient says only "yes".
- The latest patient turn explicitly asks the agent to repeat.
- The patient interrupts the agent mid-utterance.
- The patient implicitly requests repetition, e.g., "Hello?", indicating that they may not have heard the prior turn.

If any condition above applies, the annotator records the reason and skips Stage 2. When the latest patient turn explicitly asks for repetition, all seven dimension labels are PASS.

D.2 STAGE 2: SEVEN REPETITION DIMENSIONS

The seven dimensions are evaluated independently; multiple dimensions can trigger on the same response. Each dimension specifies a comparison target, a unit of analysis, and FAIL/PASS criteria. The four comparison modes are summarized in Table 8.

Units. A *sentence* ends with a period or an exclamation point; a *question* ends with a question mark; a *span* is a multi-word phrase or clause that is not a complete sentence and typically ends with "…". "That's correct." is a sentence; "I'll just confirm with the doctor…" is a span. When a response contains both a candidate sentence and a candidate span, the unit type chosen is the one carrying the actual repeated material.

D.2.1 D1. REPETITIVE SENTENCES

Comparison target. The immediately preceding real agent utterance.

Unit. Full statement sentences ending with a period. Questions and spans are excluded.

FAIL. A statement sentence in the final response that is identical to, or differs only trivially from, a statement sentence in the preceding agent turn. Trivial edits include tense changes, single-word synonym substitutions, small modifiers (e.g. "just", "really"), and determiner swaps. Two sentences that share the same SVO (Subject-Verb-Object) frame and add at most three new content words are considered trivially edited. Repeated acknowledgment or scaffolding statements (e.g., "Thanks for confirming.") also FAIL.

PASS. The sentence differs in any non-trivial linguistic way: a different main verb or predicate frame ("found" vs. "have", "scheduled" vs. "have"), a different framing or lead-in, a meaningfully different scope ("a few in-person options" vs. "a few options"), or substantive new detail. Reuse of the same topic, intent, or entity does not by itself constitute repetition.

D.2.2 D2. REPETITIVE SENTENCES (ALL PREVIOUS TURNS)

Identical PASS/FAIL criteria to D1, but the comparison target is the union of all prior real agent utterances *except* the immediately preceding one. The unit is unchanged.

D.2.3 D3. DUPLICATIVE QUESTIONS

Comparison target. The immediately preceding real agent utterance.

Unit. Full questions ending with a question mark.

FAIL. An identical or trivially reworded question ("Would you like" vs. "Do you want"), or a re-ask for information already requested in the previous agent turn *and* already provided by the patient.

PASS. A new question that serves a distinct purpose, advances the task, narrows scope, introduces new constraints or options, or is a logical follow-up to the patient's reply. Useful generic stems ("Can you please…", "Would you mind…") do not count toward repetition. Also PASS if the agent asks no question or no prior agent turn exists.

D.2.4 D4. DUPLICATIVE QUESTIONS (ALL PREVIOUS TURNS)

Identical PASS/FAIL criteria to D3, with the same expanded comparison target as D2.

D.2.5 D5. IDENTICAL SPANS

Comparison target. The immediately preceding real agent utterance.

Unit. Multi-word phrases or clauses that are not complete sentences. Single words do not qualify.

FAIL. Verbatim repetition of an earlier multi-word span; identical or trivially modified acknowledgment or scaffolding spans ("I am glad vs. "I am very glad"); template spans in which only factual fields change. Common failure templates include conversation-control phrases ("let me check that for you", "I just want to make sure"), state-assertion scaffolds ("you're currently taking", "day you were born"), checklist scaffolds, slot-selection templates ("would either of those times work", "would any of those times work"), social closings, and reused script openers.

PASS. The repetition is required for factual accuracy or clarity; the span belongs to one of the ignore-list categories in Table 9; or no baseline agent utterance exists.

Table 9: Ignore list for D5. Spans falling into these categories do not trigger a FAIL.

Category	Examples
Proper nouns / named entities	people, medications, products, organizations
Identifiers	phone numbers, record IDs, addresses, dates, times
Compound nouns	"date of birth", "available appointment slots"
Greetings, closings	"Hello", "Goodbye", "Hi there"
Single-word utterances	"Yes", "No", "Okay", "Sure"
Filler / affirmation	"I understand", "Of course", "Absolutely"
Dosage and timing	"2 tablets in the morning"
Scheduling / availability	"Available on", "Scheduled for", "Available tomorrow"
Politeness wrappers	"can you", "can you please", "can you please tell me"

D.2.6 D6. SINGLE-TURN REPETITION

Comparison target. The final agent response itself; no comparison to prior turns.

Unit. Consecutive or near-consecutive words, phrases, or sentences within the final response.

FAIL. The same information expressed twice without added value; a repeated acknowledgment or scaffolding phrase appearing more than once; an exact 3+ word phrase repeated verbatim two or more times outside a scheduling or confirmation workflow; the same date or time appearing three or more times in a scheduling response (twice is acceptable).

PASS. The repetition serves distinct functional purposes (e.g., unavailable slot, alternative offer, confirmation question); it is part of a scheduling or confirmation workflow; it conveys clinical content that requires repetition for clarity (systolic vs. diastolic readings, dosage); or it is empathetic intensification ("so, so sorry"), natural prepositional overlap ("check in in a month"), or spelled-out digits.

D.2.7 D7. PARROTING THE USER

Comparison target. The patient's last turn.

Unit. The agent's response as a whole, including its sentences and primary content.

FAIL. The agent restates the patient's last message identically or near-identically without adding new content; the agent echoes a single patient word as a standalone sentence; the agent echoes the full patient message phrased as a confirmation question. For example, patient: "1345 Toledo Drive, Suite 100"; agent: "1345 Toledo Drive, Suite 100. Is that correct?" is FAIL.

PASS. The agent paraphrases for clarification or confirmation *and* adds reconfirming or clarifying content; the agent confirms information from earlier turns rather than the last one; the agent offers a different time, date, or option than the patient requested; or the agent references only one or two patient words within a sentence whose primary content is new ("Are you still taking Metformin?" after the patient said "Metformin" is PASS, since the medication name is a necessary referent).

D.3 ANNOTATION EDGE CASES

- If the immediately preceding turn contains no real agent utterance, scan backward to the most recent one.
- If the agent takes two consecutive turns (no patient response in between), treat both as a single agent turn for comparison purposes.
- If the conversation ends with an agent turn, compare the candidate response against the preceding agent turn.
- If the agent's response is cut off, evaluate only the portion uttered before the cutoff.

- If patient turns are missing from the conversation column but appear in the payload, use the payload conversation.

D.4 Global PASS overrides

The following conditions override a FAIL and result in PASS, regardless of the dimension being evaluated:

- The latest patient turn explicitly asks the agent to repeat.
- The patient asks the agent to wait, and the agent repeats an utterance when resuming.

An earlier request to repeat does not exempt a later response. D2 still compares that response against earlier agent turns.

If no earlier agent turns exist beyond the immediately preceding one, D2 and D4 are PASS. The other dimensions are still evaluated.

D.5 Common annotation mistakes

- Do not flag a single repeated word as an identical span; spans must be multi-word.
- Do not flag compound nouns ("date of birth", "your health") as identical spans; they appear on the D5 ignore list.
- Do not confuse spans with sentences. A span may end with "…" and is partial; a sentence ends with a period and is complete.
- Do not automatically PASS D2 because the patient asked for repetition in an earlier turn; the earlier agent turns are still comparison targets.
- Do not include questions in D1 and D2; only full statement sentences count.
- For D2 and D4, the comparison set is the union of all prior agent turns except the immediately preceding one, not only the most recent prior turn.

E Per-verifier specifications

Table 10 gives the complete specification for the 13 verifiers. Rule-based verifiers operate via exact string equality at the granularity given by their target unit; subjective verifiers follow the prompt structure described in §O, with per-verifier differences in target unit, evidence scope, and exemptions.

F Supervised fine-tuning details for verifier training

F.1 Base model selection

We select Qwen3.5-27B Qwen Team (2026) as the base verifier model due to its strong instruction-following performance relative to its computational cost. On IFBench, Qwen3.5-27B achieved a score of 76.5, slightly outperforming the substantially larger Qwen3.5-122B-A10B (76.1) while being significantly cheaper to fine-tune and serve. We also evaluated Qwen3.6, but found that Qwen3.5-27B better preserved the instruction-following behaviors required for repetition verification.

F.2 Teacher style matching pipeline

For the *GPT-teacher* setting, we style match chain-of-thought rationales and PASS/FAIL judgments from GPT-5.5. To improve alignment with golden annotations, we employ a two-pass generation strategy. If the teacher's first-pass prediction disagrees with the golden label, we re-prompt using the ground-truth label as a hint and regenerate the reasoning trace. We additionally apply regex-based filtering to remove any leakage of the hint into the visible chain-of-thought.

In the $1{,}631$-item teacher-generation pool, GPT-5.5 agreed with the golden label on the first pass for $1{,}193$ items (73.1%), while the remaining 438 items (26.9%) required the hinted retry procedure. The balanced modeling subset described next contains $1{,}611$ of these items.

Table 10: Per-verifier specifications. All seven LLM verifiers return PASS when the latest patient turn explicitly asks for repetition.

Verifier	Target unit	Scope	Decision rule / exemptions
Deterministic — rule-based lexical matching			
`consecutive_token_repetition`	Word or short span	Within turn	FAIL for ≥ 5 consecutive repeats; exempt digits, ordinals, single letters, and punctuation
`prefix_match`	First 3 words	Adj. prior turn	FAIL on case-insensitive equality of the three-word prefixes
`first_sentence_repetition`	First sentence	Adj. prior turn	FAIL if the prior first sentence is a substring of the current first sentence after whitespace normalization
`first_sentence_repetition_history`	First sentence	Earlier turns	Same substring rule against earlier non-adjacent agent turns
`full_turn_repetition`	Full response	Adj. prior turn	FAIL on equality after whitespace normalization
`full_turn_repetition_history`	Full response	Earlier turns	Same equality rule against earlier non-adjacent agent turns
Subjective — LLM-as-judge (GPT-5.5)			
`within_turn_redundancy`	Acks, time refs, fillers, repeated phrases	Within turn	FAIL if redundancy is meaningful (not stylistic): repeated words, 3+ word phrases, duplicated date/time, repeated acknowledgments (e.g. "I hear you")
`redundant_questions`	Extracted questions	Adj. prior turn	FAIL on semantic match (incl. rephrasings)
`redundant_questions_history`	Extracted questions	All prior turns	Same logic, extended to full history
`phrase_reuse`	4+ word spans	Adj. prior turn	Exempt: proper nouns, named entities, phone numbers, URLs, patient-requested
`repeated_statements`	Statement sentences	Adj. prior turn	FAIL if current sentence semantically restates prior content without new information
`repeated_statements_history`	Statement sentences	All prior turns	PASS when the latest patient turn asks for repetition
`user_echoing`	User-originated phrases	Latest user turn	FAIL if reused content is the primary response, not confirmation or clinical reference

F.3 Data splits and balancing

We apply per-scenario balancing to ensure equal numbers of PASS and FAIL examples for each repetition category. A fixed-seed random split of the $1{,}611$-item modeling subset is used across all experiments, yielding $1{,}461$ training examples and 150 validation examples. The same split is reused for all model, teacher, and hyperparameter configurations.

F.4 Training configuration

Supervised fine-tuning uses the `ms-swift` framework on a single $8\times$H200 node for both full SFT and LoRA-based training. All SFT runs train for 3 epochs with a cosine learning-rate schedule, 5% warmup, and a batch size of 32. Full SFT runs take approximately 2 hours 40 minutes; LoRA runs take approximately 1 hour 40 minutes. We ran 12 SFT configurations in total across the LoRA-rank, learning-rate, and teacher-source sweep, totaling approximately 200 H200 GPU-hours.

For full SFT, we sweep learning rates in $\{5 \times 10^{-7}, 2.5 \times 10^{-6}, 5 \times 10^{-6}\}$. For LoRA, we evaluate ranks $r \in \{16, 32\}$ with learning rates in $\{5 \times 10^{-5}, 1 \times 10^{-4}\}$. LoRA α is set to $2r$ throughout using `all-linear` target modules.

G Reinforcement learning training details for verifier training

G.1 Algorithm and reward

We use Group Relative Policy Optimization (GRPO), initialized from the best SFT checkpoint. For each input, we sample $G{=}8$ rollouts and assign a binary $0/1$ reward by exact-matching the predicted PASS/FAIL verdict against the golden label; advantages are normalized within the rollout group. We use asymmetric PPO clipping ($\epsilon_{\text{low}}{=}0.2$, $\epsilon_{\text{high}}{=}0.28$) and a low-variance KL penalty to the SFT reference (coefficient $\beta{=}0.01$). Groups in which every rollout receives the same reward carry no learning signal and are filtered before the gradient step.

G.2 Optimizer and schedule

We optimize with a learning rate of 1×10^{-6}, a cosine schedule with 0.5 cycles and a minimum-LR ratio of 0.1, gradient clipping at 1.0, and entropy coefficient 0.01. Each training step samples 64 prompts $\times$ 8 rollouts (512 samples), with a PPO mini-batch of 16 prompts. The actor and reference policies use FSDP2 hybrid-shard parallelism (intra-node sharding, inter-node replication); rollouts use vLLM with tensor-parallel size 4.

G.3 Hardware and wall-clock

RL training is conducted using the `verl` framework on four $8{\times}$H200 nodes (32 H200 GPUs total). Each RL run takes approximately 14 hours of wall-clock time. We ran 2 RL configurations, totaling approximately 900 H200 GPU-hours.

G.4 Best checkpoint

We select checkpoints based on validation accuracy. The best-performing configuration uses full supervised fine-tuning with **SFT-StyleMatch** and a learning rate of 5×10^{-6}, trained for 3 epochs.

H Benchmark evaluation compute

Each of the $6{,}892$ benchmark samples is checked by six deterministic verifiers (H – M) and seven subjective verifiers (A – G). The deterministic checks require no LLM calls. A subjective check can return PASS without a call when there is no relevant comparison unit (Appendix O.1). If every subjective verifier is called for every sample, the maximum is $6{,}892 \times 7 = 48{,}244$ calls per evaluated model. The estimates below use that maximum.

Per-call token statistics. Across completed judge calls, the averages are $4{,}127$ input tokens, 408 reasoning tokens, and 359 response tokens (768 output tokens in total). Applied to the maximum call count, these averages give ≈ 199.1M input tokens and ≈ 37.1M output tokens.

GPT-5.5 verifier (LLM-as-judge). At $5 per million input tokens and $30 per million output tokens, the maximum-call estimate costs about $2,110 per model. A batch rate at half that price gives about $1,055.

Qwen 3.5-27B verifier (open-source alternative). The fine-tuned Qwen 3.5-27B verifier runs through vLLM on one H200. It completed 150 validation calls in 174 seconds (0.86 calls/sec). At that rate, $48{,}244$ calls take about 15.5 GPU-hours, or $54 at $3.50 per H200 GPU-hour. Bypassed calls reduce both estimates.

I Overlap of human labels and benchmark verifier fires

On the RepetitionBench-Gold, the seven subjective categories were annotated for $1{,}248$ model-response instances. Exactly one label was assigned to 28% of responses, at least two to 17%, and at least three to fewer than 4%. Pairwise Jaccard similarities between conceptually different

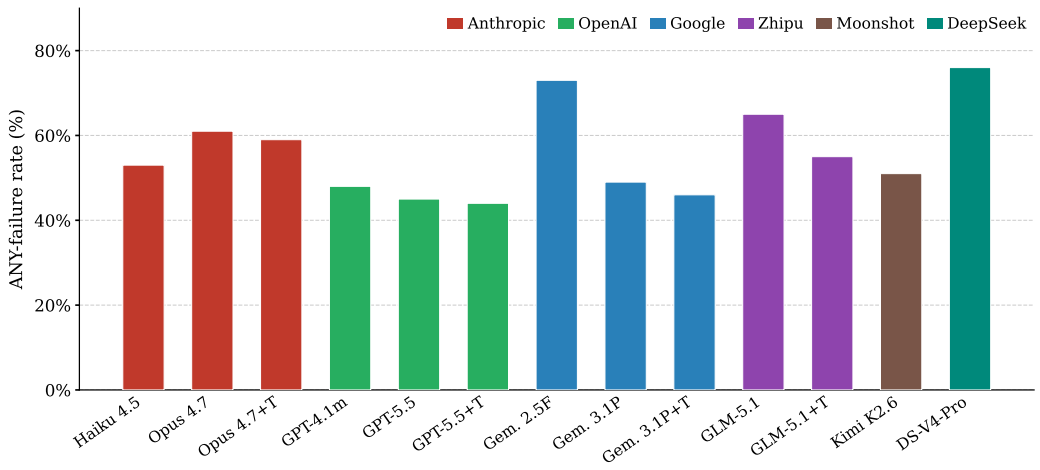


Figure 6: ANY-failure rate on the hard split (%) for the 13 frontier models, grouped by family. Lower is better.

categories were at most 0.14 (at most 0.10 for most pairs), with phi correlations near zero. The most closely related subjective categories concern questions at different dialogue scopes: D identifies an immediate redundant question, whereas F identifies one that recurs from an earlier non-adjacent turn.

On REPETITIONBENCH, the upper-triangular matrix in Figure 3c shows how often the 13 verifier flags co-occur. Pairwise Jaccard similarity is below 0.21 for most pairs. Two clusters have higher overlap. In the deterministic prefix/full-turn/first-sentence cluster (I , J , K), first-sentence and prefix repetition overlap only partly ($J = 0.45$): full-turn repetition would miss 67% of first-sentence cases and 71% of prefix cases. The history-scope pair G / M has $J = 0.44$; G captures semantic restatements from history, whereas M detects verbatim reuse of an earlier opening sentence. These categories therefore retain distinct flagged cases even where they co-occur. The primary ANY-failure metric is the union of all verifier flags, so overlap does not count a response more than once.

J BENCHMARK ANALYSIS

J.1 PER-VERIFIER FRONTIER-MODEL RESULTS

Figure 4(a) reports the percentage of responses flagged by at least one verifier. Among the 13 frontier models, the lowest rate is 27.73% on easy contexts (Gemini 3.1 Pro +Thinking) and 44.61% on hard contexts (GPT-5.5 +Thinking). Table 11 gives the per-verifier rates. Its Mean columns average rates within the subjective and deterministic groups; they are not ANY-failure rates.

J.2 REPETITION FAILURES ACROSS MODEL FAMILIES

Figure 6 groups the 13 frontier models by family on the hard split. GPT-5.5 +Thinking has the lowest ANY-failure rate (44.61%), and DeepSeek-V4-Pro the highest (76.86%). Adding Thinking reduces the rate from 48.78% to 45.92% for Gemini 3.1 Pro and from 64.58% to 54.76% for GLM-5.1; for GPT-5.5, it changes from 45.11% to 44.61%.

K UNCERTAINTY IN FULL-BENCHMARK MODEL COMPARISONS

We compare the ANY-failure rate for all $\binom{14}{2} = 91$ pairs of models separately on the easy and hard splits. A pairwise difference is the failure rate of Model A minus that of Model B, in percentage points; a positive value means that Model A fails more often. For each split, the reported 95% confidence intervals use 10,000 paired bootstrap resamples of benchmark rows, sampled with replacement. The

Table 11: Per-verifier failure rates (%) of evaluated models on REPETITIONBENCH. Lower is better. The top block lists the seven LLM-judged **Subjective** verifiers (A – G); the bottom block lists the six rule-based **Deterministic** verifiers (H – M). Each verifier column is split into **E** (easy) and **H** (hard) according to the benchmark's two-target-model construction rule. **Mean** is the arithmetic mean of the per-verifier rates within each group, not the fraction of samples on which any group verifier fires.

Model	Subjective														Mean	
	A		B		C		D		E		F		G			
	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H
Claude Opus 4.7	1.11	0.91	4.83	4.48	9.27	16.89	2.32	4.53	6.15	9.56	4.27	7.38	17.01	34.65	6.4	11.2
Claude Opus 4.7 +Thinking	0.85	0.72	4.21	4.66	8.70	15.53	2.00	4.30	5.68	8.65	3.87	7.70	16.31	31.79	5.9	10.5
GPT-5.5	1.47	1.45	4.87	5.39	7.10	12.36	1.57	3.03	5.25	6.75	2.57	3.99	11.72	22.24	4.9	7.9
GPT-5.5 +Thinking	1.40	1.18	4.87	5.30	6.76	10.24	1.51	2.63	5.51	6.84	2.45	4.76	12.38	22.60	5.0	7.7
GLM-5.1	1.42	0.77	4.13	5.03	11.99	20.20	2.15	5.12	5.44	9.06	4.89	9.24	20.03	36.78	7.2	12.3
GLM-5.1 +Thinking	1.68	1.86	5.06	5.66	10.74	16.17	2.28	4.62	5.32	7.52	3.64	6.75	16.88	30.39	6.5	10.4
Claude Haiku 4.5	1.28	1.00	5.66	5.75	7.02	13.68	1.38	2.99	4.40	6.75	2.76	6.79	14.65	29.53	5.3	9.5
GPT-5.4 mini	2.42	2.04	3.57	4.30	7.06	13.36	1.76	3.31	4.23	6.34	4.23	7.56	13.23	25.23	5.2	8.9
Gemini 3.1 Pro	1.25	0.77	3.47	3.40	6.85	12.55	1.55	2.72	3.83	5.39	2.93	6.07	14.57	29.03	4.9	8.6
Gemini 3.1 Pro +Thinking	1.00	1.04	3.21	3.99	6.80	13.31	1.23	2.49	3.76	5.12	2.83	4.98	13.68	27.31	4.6	7.9
Gemini 2.5 Flash	2.76	2.04	6.87	6.75	15.35	22.69	2.81	6.11	6.85	10.55	5.59	10.91	24.75	44.16	9.3	14.7
Kimi K2.6	1.32	0.82	3.51	4.12	8.34	14.45	1.66	3.44	4.87	7.56	3.74	7.11	13.50	27.04	5.3	9.2
DeepSeek-V4-Pro	14.48	12.27	5.80	6.57	14.78	23.91	2.87	5.66	9.42	13.72	8.21	12.91	24.35	41.26	11.4	16.6
RL-ours	2.42	2.85	3.47	4.39	2.13	3.62	0.15	0.27	2.40	2.22	0.62	1.31	2.25	4.48	1.9	2.7

Model	Deterministic												Mean	
	H		I		J		K		L		M			
	E	H	E	H	E	H	E	H	E	H	E	H	E	H
Claude Opus 4.7	0.09	0.41	1.23	2.54	0.26	0.77	1.62	3.49	0.26	1.59	14.18	31.61	2.9	6.7
Claude Opus 4.7 +Thinking	0.11	0.36	1.13	2.36	0.30	0.68	1.42	2.58	0.28	1.36	13.71	29.21	2.8	6.1
GPT-5.5	0.00	0.09	0.72	1.31	0.19	0.63	0.98	1.59	0.06	0.82	8.68	19.79	1.8	4.0
GPT-5.5 +Thinking	0.00	0.09	0.72	1.31	0.23	0.54	1.11	1.68	0.06	0.82	8.99	19.57	1.9	4.0
GLM-5.1	0.02	0.36	1.83	2.99	0.17	0.63	1.68	2.94	0.60	1.99	15.18	33.38	3.2	7.0
GLM-5.1 +Thinking	0.04	0.27	1.25	1.59	0.06	0.09	1.08	1.86	0.09	0.59	10.42	23.69	2.2	4.7
Claude Haiku 4.5	0.04	0.54	0.26	1.18	0.00	0.36	0.55	1.72	0.09	1.04	9.02	23.19	1.7	4.7
GPT-5.4 mini	0.43	0.91	0.83	2.58	0.23	0.86	1.34	2.90	0.43	1.63	9.89	22.46	2.2	5.2
Gemini 3.1 Pro	0.04	0.23	0.87	1.22	0.06	0.09	1.00	1.45	0.02	0.68	10.46	25.36	2.1	4.8
Gemini 3.1 Pro +Thinking	0.04	0.23	0.79	1.13	0.00	0.00	1.11	1.45	0.02	0.18	10.51	23.41	2.0	4.4
Gemini 2.5 Flash	0.04	0.45	2.68	4.26	0.70	1.09	2.66	4.39	0.66	2.49	17.82	37.27	4.1	8.3
Kimi K2.6	0.06	0.41	1.04	1.95	0.11	0.32	1.19	1.99	0.09	0.59	9.72	22.37	2.0	4.6
DeepSeek-V4-Pro	0.13	0.36	2.51	5.07	1.11	1.90	2.98	6.16	1.85	4.57	18.37	38.36	4.5	9.4
RL-ours	0.57	0.82	0.00	0.09	0.00	0.00	0.00	0.27	0.00	0.00	0.72	1.90	0.2	0.5

same resampled rows are used for all models, preserving the paired comparison. Pairwise significance is assessed with McNemar tests, with p-values Holm-adjusted across all 91 comparisons within each split. The two parts of the pairwise results appear in Tables 12 and 13. For every baseline, its ANY-failure rate exceeds that of RL (ours) on both splits, with all corresponding Holm-adjusted $p < 0.001$.

To assess rank stability, we rank models by ANY-failure rate within each of the 10,000 paired resamples, using the same sampled rows for every model. Table 14 reports point ranks, 95% bootstrap rank intervals, and mean ranks. RL (ours) ranks first in every resample on both splits (mean rank 1.00; rank interval $[1, 1]$). The held-out 150-item gold subset is used for verifier validation, not for these model rankings.

Table 12: Pairwise ANY-failure differences for all model pairs, part 1. Entries are Model A minus Model B in percentage points. Brackets give 95% confidence intervals from 10,000 paired-bootstrap resamples; the p columns give Holm-adjusted McNemar p-values for the easy and hard splits separately. Continued in Table 13.

		Easy		Hard	
Model A	Model B	$A - B$ [95% CI]	Holm p	$A - B$ [95% CI]	Holm p
Claude Opus 4.7	Claude Opus 4.7 +T	2.67 [1.41, 3.95]	0.001	2.91 [1.04, 4.77]	0.042
Claude Opus 4.7	Claude Haiku 4.5	6.12 [4.61, 7.63]	< 0.001	8.03 [5.76, 10.35]	< 0.001
Claude Opus 4.7	GPT-5.5	8.27 [6.72, 9.81]	< 0.001	16.25 [13.84, 18.61]	< 0.001
Claude Opus 4.7	GPT-5.5 +T	8.38 [6.80, 9.94]	< 0.001	16.80 [14.30, 19.25]	< 0.001
Claude Opus 4.7	GPT-5.4 mini	7.53 [5.91, 9.15]	< 0.001	13.25 [10.85, 15.66]	< 0.001
Claude Opus 4.7	Gemini 3.1 Pro	8.72 [7.14, 10.26]	< 0.001	12.53 [10.12, 14.93]	< 0.001
Claude Opus 4.7	Gemini 3.1 Pro + Thinking	9.98 [8.42, 11.52]	< 0.001	15.39 [13.07, 17.70]	< 0.001
Claude Opus 4.7	Gemini 2.5 Flash	−11.92 [−13.54, −10.34]	< 0.001	−11.30 [−13.57, −9.08]	< 0.001
Claude Opus 4.7	GLM-5.1	−2.90 [−4.48, −1.41]	0.007	−3.18 [−5.36, −1.00]	0.060
Claude Opus 4.7	GLM-5.1 +T	2.03 [0.45, 3.60]	0.206	6.58 [4.31, 8.94]	< 0.001
Claude Opus 4.7	Kimi K2.6	6.74 [5.16, 8.30]	< 0.001	10.49 [8.13, 12.94]	< 0.001
Claude Opus 4.7	DeepSeek-V4-Pro	−17.30 [−18.96, −15.65]	< 0.001	−15.52 [−17.84, −13.21]	< 0.001
Claude Opus 4.7	RL (ours)	25.23 [23.59, 26.79]	< 0.001	43.62 [41.13, 46.12]	< 0.001
Claude Opus 4.7 +T	Claude Haiku 4.5	3.45 [1.90, 5.01]	< 0.001	5.13 [2.91, 7.40]	< 0.001
Claude Opus 4.7 +T	GPT-5.5	5.61 [4.01, 7.19]	< 0.001	13.35 [10.99, 15.66]	< 0.001
Claude Opus 4.7 +T	GPT-5.5 +T	5.72 [4.14, 7.31]	< 0.001	13.89 [11.44, 16.34]	< 0.001
Claude Opus 4.7 +T	GPT-5.4 mini	4.86 [3.26, 6.46]	< 0.001	10.35 [7.94, 12.85]	< 0.001
Claude Opus 4.7 +T	Gemini 3.1 Pro	6.06 [4.48, 7.61]	< 0.001	9.62 [7.31, 11.98]	< 0.001
Claude Opus 4.7 +T	Gemini 3.1 Pro + Thinking	7.31 [5.71, 8.87]	< 0.001	12.48 [10.12, 14.89]	< 0.001
Claude Opus 4.7 +T	Gemini 2.5 Flash	−14.59 [−16.19, −12.99]	< 0.001	−14.21 [−16.48, −11.94]	< 0.001
Claude Opus 4.7 +T	GLM-5.1	−5.57 [−7.10, −4.05]	< 0.001	−6.08 [−8.26, −3.86]	< 0.001
Claude Opus 4.7 +T	GLM-5.1 +T	−0.64 [−2.24, 0.92]	1.000	3.68 [1.36, 5.99]	0.036
Claude Opus 4.7 +T	Kimi K2.6	4.07 [2.50, 5.67]	< 0.001	7.58 [5.17, 10.08]	< 0.001
Claude Opus 4.7 +T	DeepSeek-V4-Pro	−19.96 [−21.65, −18.30]	< 0.001	−18.43 [−20.79, −16.07]	< 0.001
Claude Opus 4.7 +T	RL (ours)	22.56 [20.92, 24.16]	< 0.001	40.72 [38.18, 43.21]	< 0.001
Claude Haiku 4.5	GPT-5.5	2.15 [0.58, 3.71]	0.132	8.22 [5.81, 10.71]	< 0.001
Claude Haiku 4.5	GPT-5.5 +T	2.26 [0.64, 3.84]	0.109	8.76 [6.31, 11.26]	< 0.001
Claude Haiku 4.5	GPT-5.4 mini	1.41 [−0.19, 3.01]	0.909	5.22 [2.77, 7.63]	< 0.001
Claude Haiku 4.5	Gemini 3.1 Pro	2.60 [1.09, 4.07]	0.017	4.49 [2.13, 6.95]	0.005
Claude Haiku 4.5	Gemini 3.1 Pro + Thinking	3.86 [2.30, 5.33]	< 0.001	7.35 [4.99, 9.71]	< 0.001
Claude Haiku 4.5	Gemini 2.5 Flash	−18.04 [−19.71, −16.40]	< 0.001	−19.34 [−21.65, −17.02]	< 0.001
Claude Haiku 4.5	GLM-5.1	−9.02 [−10.64, −7.46]	< 0.001	−11.21 [−13.48, −8.94]	< 0.001
Claude Haiku 4.5	GLM-5.1 +T	−4.09 [−5.69, −2.54]	< 0.001	−1.45 [−3.81, 0.95]	1.000
Claude Haiku 4.5	Kimi K2.6	0.62 [−0.98, 2.15]	1.000	2.45 [0.05, 4.95]	0.428
Claude Haiku 4.5	DeepSeek-V4-Pro	−23.42 [−25.10, −21.75]	< 0.001	−23.56 [−25.92, −21.15]	< 0.001
Claude Haiku 4.5	RL (ours)	19.11 [17.53, 20.56]	< 0.001	35.59 [33.09, 38.13]	< 0.001
GPT-5.5	GPT-5.5 +T	0.11 [−1.22, 1.41]	1.000	0.54 [−1.41, 2.59]	1.000
GPT-5.5	GPT-5.4 mini	−0.75 [−2.28, 0.81]	1.000	−3.00 [−5.40, −0.54]	0.167
GPT-5.5	Gemini 3.1 Pro	0.45 [−1.09, 1.96]	1.000	−3.72 [−6.13, −1.32]	0.042
GPT-5.5	Gemini 3.1 Pro + Thinking	1.71 [0.19, 3.22]	0.439	−0.86 [−3.27, 1.50]	1.000
GPT-5.5	Gemini 2.5 Flash	−20.20 [−21.88, −18.51]	< 0.001	−27.55 [−30.00, −25.10]	< 0.001
GPT-5.5	GLM-5.1	−11.18 [−12.82, −9.55]	< 0.001	−19.43 [−21.88, −17.02]	< 0.001
GPT-5.5	GLM-5.1 +T	−6.25 [−7.85, −4.67]	< 0.001	−9.67 [−12.07, −7.22]	< 0.001
GPT-5.5	Kimi K2.6	−1.54 [−3.07, 0.00]	0.599	−5.76 [−8.08, −3.40]	< 0.001
GPT-5.5	DeepSeek-V4-Pro	−25.57 [−27.30, −23.84]	< 0.001	−31.77 [−34.23, −29.32]	< 0.001
GPT-5.5	RL (ours)	16.95 [15.40, 18.51]	< 0.001	27.37 [24.83, 29.96]	< 0.001
GPT-5.5 +T	GPT-5.4 mini	−0.85 [−2.45, 0.75]	1.000	−3.54 [−6.08, −1.04]	0.060
GPT-5.5 +T	Gemini 3.1 Pro	0.34 [−1.19, 1.86]	1.000	−4.27 [−6.72, −1.91]	0.013
GPT-5.5 +T	Gemini 3.1 Pro + Thinking	1.60 [0.06, 3.11]	0.599	−1.41 [−3.81, 1.00]	1.000
GPT-5.5 +T	Gemini 2.5 Flash	−20.30 [−22.01, −18.58]	< 0.001	−28.10 [−30.64, −25.65]	< 0.001
GPT-5.5 +T	GLM-5.1	−11.28 [−12.92, −9.68]	< 0.001	−19.97 [−22.51, −17.43]	< 0.001
GPT-5.5 +T	GLM-5.1 +T	−6.36 [−7.93, −4.76]	< 0.001	−10.21 [−12.71, −7.63]	< 0.001
GPT-5.5 +T	Kimi K2.6	−1.64 [−3.22, −0.09]	0.599	−6.31 [−8.67, −3.86]	< 0.001
GPT-5.5 +T	DeepSeek-V4-Pro	−25.68 [−27.38, −23.97]	< 0.001	−32.32 [−34.86, −29.73]	< 0.001
GPT-5.5 +T	RL (ours)	16.85 [15.29, 18.36]	< 0.001	26.83 [24.38, 29.32]	< 0.001
GPT-5.4 mini	Gemini 3.1 Pro	1.19 [−0.43, 2.75]	1.000	−0.73 [−3.27, 1.82]	1.000

Table 13: Pairwise ANY-failure differences, part 2 (continued from Table 12). Brackets give 95% paired-bootstrap confidence intervals; p-values are from McNemar tests with Holm adjustment across the 91 pairs in each split. The left estimate/p pair is the easy split and the right pair is the hard split.

		Easy		Hard	
Model A	Model B	$A - B$ [95% CI]	Holm p	$A - B$ [95% CI]	Holm p
GPT-5.4 mini	Gemini 3.1 Pro +T	$2.45\ [0.85, 4.03]$	0.051	$2.13\ [-0.32, 4.63]$	0.704
GPT-5.4 mini	Gemini 2.5 Flash	$-19.45\ [-21.11, -17.79]$	< 0.001	$-24.56\ [-27.01, -22.11]$	< 0.001
GPT-5.4 mini	GLM-5.1	$-10.43\ [-12.05, -8.85]$	< 0.001	$-16.43\ [-18.84, -13.94]$	< 0.001
GPT-5.4 mini	GLM-5.1 +T	$-5.50\ [-7.10, -3.90]$	< 0.001	$-6.67\ [-9.21, -4.13]$	< 0.001
GPT-5.4 mini	Kimi K2.6	$-0.79\ [-2.41, 0.81]$	1.000	$-2.77\ [-5.22, -0.27]$	0.256
GPT-5.4 mini	DeepSeek-V4-Pro	$-24.82\ [-26.47, -23.12]$	< 0.001	$-28.78\ [-31.23, -26.28]$	< 0.001
GPT-5.4 mini	RL (ours)	$17.70\ [16.19, 19.19]$	< 0.001	$30.37\ [27.87, 32.86]$	< 0.001
Gemini 3.1 Pro	Gemini 3.1 Pro +T	$1.26\ [0.02, 2.50]$	0.599	$2.86\ [1.00, 4.77]$	0.042
Gemini 3.1 Pro	Gemini 2.5 Flash	$-20.64\ [-22.29, -19.02]$	< 0.001	$-23.83\ [-26.19, -21.47]$	< 0.001
Gemini 3.1 Pro	GLM-5.1	$-11.62\ [-13.24, -10.07]$	< 0.001	$-15.71\ [-18.11, -13.39]$	< 0.001
Gemini 3.1 Pro	GLM-5.1 +T	$-6.70\ [-8.25, -5.16]$	< 0.001	$-5.95\ [-8.35, -3.54]$	< 0.001
Gemini 3.1 Pro	Kimi K2.6	$-1.98\ [-3.50, -0.45]$	0.206	$-2.04\ [-4.45, 0.36]$	0.704
Gemini 3.1 Pro	DeepSeek-V4-Pro	$-26.02\ [-27.77, -24.25]$	< 0.001	$-28.05\ [-30.59, -25.56]$	< 0.001
Gemini 3.1 Pro	RL (ours)	$16.51\ [14.95, 18.06]$	< 0.001	$31.09\ [28.55, 33.55]$	< 0.001
Gemini 3.1 Pro +T	Gemini 2.5 Flash	$-21.90\ [-23.52, -20.24]$	< 0.001	$-26.69\ [-28.96, -24.38]$	< 0.001
Gemini 3.1 Pro +T	GLM-5.1	$-12.88\ [-14.48, -11.30]$	< 0.001	$-18.57\ [-20.84, -16.25]$	< 0.001
Gemini 3.1 Pro +T	GLM-5.1 +T	$-7.95\ [-9.51, -6.42]$	< 0.001	$-8.81\ [-11.21, -6.45]$	< 0.001
Gemini 3.1 Pro +T	Kimi K2.6	$-3.24\ [-4.76, -1.71]$	< 0.001	$-4.90\ [-7.31, -2.45]$	0.002
Gemini 3.1 Pro +T	DeepSeek-V4-Pro	$-27.28\ [-28.98, -25.49]$	< 0.001	$-30.91\ [-33.45, -28.37]$	< 0.001
Gemini 3.1 Pro +T	RL (ours)	$15.25\ [13.73, 16.76]$	< 0.001	$28.23\ [25.69, 30.73]$	< 0.001
Gemini 2.5 Flash	GLM-5.1	$9.02\ [7.34, 10.62]$	< 0.001	$8.13\ [5.95, 10.35]$	< 0.001
Gemini 2.5 Flash	GLM-5.1 +T	$13.95\ [12.28, 15.61]$	< 0.001	$17.88\ [15.57, 20.20]$	< 0.001
Gemini 2.5 Flash	Kimi K2.6	$18.66\ [17.00, 20.35]$	< 0.001	$21.79\ [19.47, 24.15]$	< 0.001
Gemini 2.5 Flash	DeepSeek-V4-Pro	$-5.37\ [-7.04, -3.65]$	< 0.001	$-4.22\ [-6.49, -1.95]$	0.006
Gemini 2.5 Flash	RL (ours)	$37.15\ [35.49, 38.81]$	< 0.001	$54.93\ [52.56, 57.24]$	< 0.001
GLM-5.1	GLM-5.1 +T	$4.93\ [3.43, 6.40]$	< 0.001	$9.76\ [7.63, 11.85]$	< 0.001
GLM-5.1	Kimi K2.6	$9.64\ [8.06, 11.22]$	< 0.001	$13.66\ [11.30, 16.07]$	< 0.001
GLM-5.1	DeepSeek-V4-Pro	$-14.40\ [-16.04, -12.71]$	< 0.001	$-12.35\ [-14.66, -10.03]$	< 0.001
GLM-5.1	RL (ours)	$28.13\ [26.49, 29.73]$	< 0.001	$46.80\ [44.35, 49.21]$	< 0.001
GLM-5.1 +T	Kimi K2.6	$4.71\ [3.16, 6.25]$	< 0.001	$3.90\ [1.54, 6.22]$	0.026
GLM-5.1 +T	DeepSeek-V4-Pro	$-19.32\ [-21.03, -17.59]$	< 0.001	$-22.11\ [-24.56, -19.66]$	< 0.001
GLM-5.1 +T	RL (ours)	$23.20\ [21.60, 24.78]$	< 0.001	$37.04\ [34.45, 39.58]$	< 0.001
Kimi K2.6	DeepSeek-V4-Pro	$-24.03\ [-25.78, -22.31]$	< 0.001	$-26.01\ [-28.55, -23.47]$	< 0.001
Kimi K2.6	RL (ours)	$18.49\ [16.93, 20.03]$	< 0.001	$33.14\ [30.59, 35.68]$	< 0.001
DeepSeek-V4-Pro	RL (ours)	$42.53\ [40.90, 44.10]$	< 0.001	$59.15\ [56.83, 61.42]$	< 0.001

Table 14: Model rank stability under 10,000 paired-bootstrap resamples. Rank 1 has the lowest ANY-failure rate. RI denotes the 95% bootstrap rank interval.

Easy			
Model	Rank	95% RI	Mean rank
RL (ours)	1	[1, 1]	1.00
GPT-5.5 +Thinking	4	[3, 6]	4.25
Gemini 3.1 Pro +Thinking	2	[2, 3]	2.06
GPT-5.5	5	[3, 6]	4.47
Gemini 3.1 Pro	3	[3, 6]	3.67
GPT-5.4 mini	6	[3.5, 7.5]	5.82
Kimi K2.6	7	[5, 8]	7.01
Claude Haiku 4.5	8	[6, 8]	7.73
Claude Opus 4.7 +Thinking	9	[9, 10]	9.22
GLM-5.1 +Thinking	10	[9, 10]	9.79
Claude Opus 4.7	11	[11, 11]	10.99
GLM-5.1	12	[12, 12]	12.00
Gemini 2.5 Flash	13	[13, 13]	13.00
DeepSeek-V4-Pro	14	[14, 14]	14.00

Hard			
Model	Rank	95% RI	Mean rank
RL (ours)	1	[1, 1]	1.00
GPT-5.5 +Thinking	2	[2, 4]	2.42
Gemini 3.1 Pro +Thinking	4	[2, 5]	3.67
GPT-5.5	3	[2, 4]	2.96
Gemini 3.1 Pro	6	[5, 7]	5.75
GPT-5.4 mini	5	[4, 6]	5.26
Kimi K2.6	7	[6, 7]	6.96
Claude Haiku 4.5	8	[8, 9]	8.10
Claude Opus 4.7 +Thinking	10	[10, 10]	10.00
GLM-5.1 +Thinking	9	[8, 9]	8.88
Claude Opus 4.7	11	[11, 11]	11.00
GLM-5.1	12	[12, 12]	12.00
Gemini 2.5 Flash	13	[13, 13]	13.00
DeepSeek-V4-Pro	14	[14, 14]	14.00

L CONTEXT CORRELATES OF REPETITION

We analyzed 96,488 responses: 13 frontier models and our RL model on the same 6,892 contexts. The regression included task type, difficulty, dialogue state, turn count, context length, model family, response length, and reasoning mode. Repeated clarification and continued discussion of the same topic were associated with more repetition flags. Dialogue-state features correlated with deterministic flags (Spearman's $\rho = 0.41$) and subjective flags ($\rho = 0.34$). These associations do not establish causation.

M DECODING SENSITIVITY

We sampled 200 distinct REPETITIONBENCH-GOLD contexts evenly across payload lengths and tested nine models on each. At top-$p = 1$, we used temperatures 0, 0.3, and 0.7. We also tested temperature 0.7 with top-$p = 0.8$. Each nonzero-temperature setting was run twice with different seeds, giving seven runs per model and 12,600 responses in total. We scored every response with all 13 verifiers.

Table 15: SODA-Eval results at temperature 1 on 271 contexts. Each model generated five responses per context (1,355 total). Flags count responses judged repetitive; quality is the mean score on a 1–5 scale. RL (30%) uses a 30% repetition-reward weight.

Model	Flags	Rate	Quality
DeepSeek-V3.1 base	32	2.36%	4.539
RL (30%)	28	2.07%	4.767

Figure 7 plots the RL model and DeepSeek-V4-Pro, which account for 2,800 of those responses. The RL model's ANY_{all} rate was lower in every run; the gap ranged from 24.0 to 36.5 percentage points. In the main text, the DeepSeek-V4-Pro rates at nonzero temperature average the two seeds.

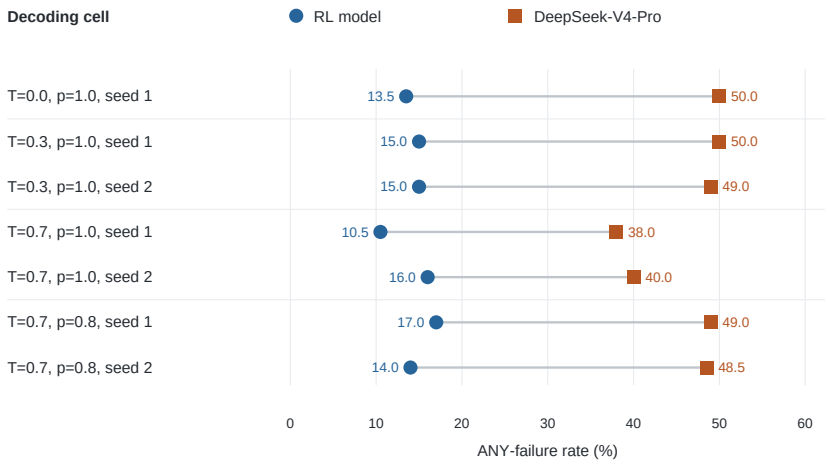


Figure 7: ANY-failure rates across decoding runs on the same 200 REPETITIONBENCH-GOLD contexts. Each point is the fraction of responses flagged by at least one of the 13 verifiers. Each model contributes one response per context and run. T denotes temperature and p denotes top-p; seed 1 and seed 2 are repeated runs.

Across all nine models, the paired seed-1 contrast from temperature 0.3 to 0.7 at top-$p = 1$ was -3.83 percentage points (95% paired bootstrap CI $[-5.72, -1.94]$). Lowering top-p from 1 to 0.8 at temperature 0.7 gave $+2.67$ points (CI $[0.72, 4.61]$). Each pooled contrast uses 1,800 context–model pairs; these estimates describe decoding changes across nine models, not the two-model gap above.

N EXTERNAL SODA-EVAL EVALUATION

We use the 271 SODA-Eval (Mendonça et al., 2024) test contexts whose released responses carry a REPETITION label. We remove each released response and give both models the same preceding dialogue. At temperature 1, the DeepSeek-V3.1 base and the RL checkpoint trained with a 30% repetition-reward weight each generate five responses per context, with a 128-token limit and no repetition-specific instruction or penalty. An identity-blind GPT-5.6-terra judge applies the SODA-Eval taxonomy and scores response quality from 1 to 5.

The response-level difference is 0.30 percentage points. Mean judge-rated quality is about 5% higher for RL (30%).

O VERIFIER PROMPTS AND PSEUDO-CODE

This section reproduces the artifacts that operationalize each of the 13 verifiers introduced in §3.2: the LLM-judge prompts used by the seven subjective verifiers (A–G) and the pseudo-code for the six deterministic verifiers (H–M). A shared comparison-unit

extraction routine (Appendix O.1) pre-computes the baseline agent utterance, the patient's last turn, and the set of all earlier agent turns excluding the immediately preceding one. These pre-extracted units are substituted into each subjective prompt as named template variables (e.g. `{immediately_preceding_agent_utterance}`, `{patient_last_turn}`, `{baseline_agent_utterances_set}`, `{ai_agent_response}`, `{conversation}`). `{grade_format_prompt}` is replaced by a binary PASS / FAIL output schema ("yes" = PASS, "no" = FAIL). For readability, Unicode dividers, curly quotes, ellipses, arrows, and inequality signs in the released prompts have been transliterated to ASCII; the substantive instructions are unchanged.

O.1 Comparison-unit Extraction

Each subjective verifier consumes a small set of pre-extracted comparison units rather than the raw transcript. The procedure below collapses consecutive agent turns (when the agent spoke multiple times in a row without an intervening patient turn) into a single baseline utterance, mirroring the annotation protocol's consecutive-agent-turns rule (Appendix D.3). Four boolean "has–" flags act as bypasses: when the baseline contains no candidate units of the relevant type (questions or statement sentences), the verifier short-circuits to PASS without an LLM call, eliminating both spurious failures and unnecessary inference cost.

```
function extract_comparison_units(conversation):
  turns = parse_speaker_text_pairs(conversation)
      # ordered list of (speaker in {agent, patient}, text)

  patient_last = last_text_with_speaker(turns, "patient") or "NONE"

  # Walk back from the most recent agent turn, accumulating consecutive
  # agent turns until a patient turn or the start is reached.
  i = last_index_with_speaker(turns, "agent")
  consecutive = []
  split_idx   = 0
  while i >= 0 and turns[i].speaker == "agent":
    consecutive.insert(0, turns[i].text)
    split_idx = i
    i        -= 1

  if consecutive is non-empty:
    baseline = "\n".join(consecutive)              # immediately preceding utterance
    previous = [t.text for t in turns[:split_idx]
                if t.speaker == "agent"]           # all earlier agent turns
  else:
    baseline = "NONE"
    previous = []

  baseline_set = format_as_turn_list(previous) if previous else "NONE"

  return {
    "immediately_preceding_agent_utterance": baseline,
    "patient_last_turn":                     patient_last,
    "baseline_agent_utterances_set":         baseline_set,
    "baseline_has_questions":      "YES" if "?" in baseline                    else "NO",
    "baseline_has_statements":     "YES" if has_statement_sentence(baseline)   else "NO",
    "baseline_set_has_questions":  "YES" if any("?" in t for t in previous)    else "NO",
    "baseline_set_has_statements": "YES" if any(has_statement_sentence(t)
                                            for t in previous)                 else "NO",
  }

# A statement sentence is detected by a word-final period followed by
# whitespace or end-of-string ("\w\.\s|\w\.$"), avoiding false positives
# from common abbreviations.
function has_statement_sentence(text):
  return regex_search(r"\w\.\s|\w\.$", text) is not None
```

O.2 Shared System Prompt for Subjective Verifiers

All seven subjective verifiers (A – G) use the system prompt below for unit definitions and global PASS/FAIL exceptions. It takes priority over their user prompts. An explicit request to repeat in the latest patient turn therefore yields PASS even for G, whose user prompt includes a conflicting repeat-handling instruction.

```
Your job is to evaluate repetition quality in the AI agent response.
```

```
UNIT DEFINITIONS (DO NOT MIX UNITS):
- Sentence / statement: ends with "."
- Question: ends with "?"
- Span: phrase/word/clause-level content; a sentence ending with "..." is a span.
- For spans, evaluate ONLY phrases/words/clauses within the response
  (not a full sentence or full question).
  Example: "That's correct." is a sentence, NOT a span.

GLOBAL EXCEPTIONS / OVERRIDES (HIGHEST PRIORITY):
- If the patient explicitly asks for repetition -> PASS.
- If the agent repeats because the patient asked to wait -> PASS.
- If the patient explicitly asked for repetition -> PASS.
     Count as PASS if the utterance contains a direct request such as:
     - "repeat that", "repeat what you said"
     - "say that again", "could you say that again"
     - "repeat", "say it again"

GENERAL REPETITION PRINCIPLES:
- Any sentence/question that is identical OR nearly identical
  (minor word changes) counts as repetition failure.
  One-word difference can still be a FAIL if the majority matches
  beyond that.
- Trivial word swaps do NOT count as paraphrasing.
- Pay special attention to repeated acknowledgement / scaffolding
  (e.g., "Thanks for confirming...").
```

O.3 Subjective Verifier User Prompts

Each subjective verifier composes the shared system prompt above with a dimension-specific user prompt. We reproduce the user prompts below in the order A – G.

O.3.1 Within-turn redundancy

(A, `unnecessary_single_turn`)

Evaluates the response in isolation; no comparison to prior turns.

```
Dimension: Unnecessary repetition within a single turn.

Evaluate the final agent response ALONE for redundant or unnecessary
repetition. Do NOT compare to previous turns.

YOU ARE GIVEN:
1) Final agent response (to grade): {ai_agent_response}

FAIL if ANY of the following are true:

1) Consecutive repeated word (e.g., "wine wine wine", "in in").
   EXCEPTIONS - these consecutive patterns are PASS:
   - Spelling out numbers/address digits
   - Empathetic intensifiers: "so, so sorry", "very, very" = PASS
   - Natural prepositional overlap: "check in in a month" = PASS

2) The EXACT same multi-word phrase (3+ words, verbatim) appears 2+ times
   AND is not part of a scheduling/confirmation workflow.

3) Same date/time appears 3+ times in a scheduling response.
   2 times is acceptable. Only 3 or more triggers FAIL.
   EXCEPTION: if the 3+ mentions serve DIFFERENT purposes (unavailable +
   alternative + confirmation), that is PASS even if 3+.

4) Repeated acknowledgement/scaffolding phrase appears multiple times
   (e.g., "Glad to hear that" said twice in one response).

IMPORTANT - ALL OF THESE ARE PASS (not single-turn repetition):

Scheduling & confirmation (ALWAYS PASS):
- Date/time mentioned 2x for confirmation = PASS
- Date mentioned 2-3x across unavailable + alternative + question = PASS
- "those times work for you... Would either of those times work" = PASS
- "Just to confirm, [date/time]... Is that correct?" = PASS
- "tomorrow" mentioned 2-3x in scheduling context = PASS
- Location name repeated in scheduling ("Flower Mound location") = PASS

Medical & clinical (ALWAYS PASS):
- Blood pressure numbers for explanation (systolic vs diastolic) = PASS
- "clear liquid diet" in medical instructions = PASS
```

```
- "diabetes self-monitoring" or service names in listings = PASS
- Dosage numbers referenced for confirmation = PASS

Empathetic speech (ALWAYS PASS):
- "so, so sorry" = PASS
- "Oh no..." = PASS

Conversational (ALWAYS PASS):
- "check in in a month" = PASS (phrasal verb + preposition)
- Same noun different modifiers ("replacement device... new device") = PASS
- Common pronouns or function words = PASS

Output strictly:
{grade_format_prompt}
```

O.3.2 USER ECHOING / PARROTING (B, PARROTING_USER)

Compares the agent's response only to the patient's most recent turn.

```
Dimension: Parroting the user (AGENT vs PATIENT LAST TURN).

YOU ARE GIVEN:
1) Final agent response (to grade): {ai_agent_response}
2) Full conversation transcript (for context): {conversation}
3) Pre-extracted patient's last turn:
{patient_last_turn}

NOTE: Item 3 has been pre-extracted from the conversation for you.
Use it as the primary source for comparison. Refer to the full
conversation only if you need additional context.

YOUR GOAL:
Determine if the agent parrots the patient's MOST RECENT turn.

CRITICAL: Only compare to the patient's LAST turn (most recent).

SCORING:
- "yes" = PASS = no parroting
- "no"  = FAIL = parroting detected

FAIL ("no") if:
The agent echoes back the patient's LAST turn as a WHOLE or as the
PRIMARY content of the response. This means the agent repeats the
patient's full statement, full address, full number, or full phrase
as the main thing they say, even if they append a question or "Got it."

FAIL examples:
- Patient: "1345 Toledo Drive, Suite 100"
  -> Agent: "1345 Toledo Drive, Suite 100. Is that correct?"
  = FAIL (full address echoed back)
- Patient: "150 milligrams" -> Agent: "150 milligrams. Got it." = FAIL
- Patient: "Schedule an appointment"
  -> Agent: "Did you say you want to schedule an appointment?"
  = FAIL (echoing back as question)
- Patient: "He has" -> Agent: "He has, right?" = FAIL
- Patient: "8 o'clock on Monday"
  -> Agent: "8 o'clock on Monday is noted."
  = FAIL (echo + minimal ack)
- Patient: "23 Oak Street, Apartment 7"
  -> Agent: "23 Oak Street, Apartment 7. Got that noted."
  = FAIL (full address echoed)

WHAT DOES NOT COUNT AS PARROTING:
- Shared dates, times, locations, medication names that naturally appear
  in scheduling or medical workflows
- Agent confirming info from EARLIER turns (not the last turn)
- Agent offering a DIFFERENT time/date/option than what patient requested
- Agent clarifying misheard content
- Single words like "yes", "no", "okay" appearing in both turns
- "Goodbye" or basic conversational closers

PASS ("yes") examples:
- Patient: "Yes" / "Yeah" / "No" / "That's correct"
  -> Agent confirms earlier info or moves forward
  = PASS (not echoing the last turn)
- Patient asks about 2:00 PM -> Agent offers 2:30 PM = PASS (different time)
- Patient: "May 29th"
  -> Agent: "May 29th at 2:30 PM at Flower Mound..."
  = PASS (shared date is natural in scheduling)
```

```
- Patient: "Metformin"
  -> Agent: "Are you still taking Metformin?"
  = PASS (medication name is necessary reference)
- Patient: "One fifty"
  -> Agent: "I heard 1150, doesn't seem right"
  = PASS (clarification of misheard content)
- Patient: "five mg"
  -> Agent: "5 mg. Was the dose administered?"
  = PASS (brief reference + new question about different topic)
- Patient: "145 over 92"
  -> Agent: "Both numbers are higher than normal..."
  = PASS (medical confirmation + explanation)
- Patient: "No" -> Agent: "Goodbye." = PASS

The KEY distinction:
- Echoing the WHOLE patient message or its PRIMARY content = FAIL
- Referencing 1-2 words while adding substantial new content = PASS

PASS EXCEPTIONS (override FAIL):
- <TASK> explicitly instructs repeating/reconfirming.
- Patient explicitly asks for repetition.
- No patient turn exists.

REMEMBER: "no" = parroting detected (FAIL),
          "yes" = no parroting (PASS).

Output strictly:
{grade_format_prompt}

REASONING must include IN THIS ORDER:
1) Patient's last turn (quote it exactly)
2) Agent's response (quote it exactly)
3) What content from the patient's last turn appears in the
   agent's response
4) Whether the echo is the whole/primary content or just a brief
   reference
5) FINAL SCORE: "yes" (no parroting) or "no" (parroting detected)
```

O.3.3 PHRASE REUSE / IDENTICAL SPANS (C , IDENTICAL_SPAN)

Detects multi-word spans (4+ words) reused verbatim from the immediately preceding agent utterance, gated by a long ignore list of clinically or operationally required templates.

```
Dimension: Identical spans (SPANS ONLY: phrases, clauses, word combinations).

YOU ARE GIVEN:
1) Final agent response (to grade): {ai_agent_response}
2) Full conversation transcript (for context): {conversation}
3) Pre-extracted baseline agent utterance (immediately preceding real
   agent utterance):
{immediately_preceding_agent_utterance}
4) Pre-extracted patient's last utterance (to detect repetition requests):
{patient_last_turn}

NOTE: Items 3 and 4 have been pre-extracted from the conversation for you.
Use them as the primary source for comparison. Refer to the full
conversation only if you need additional context.

YOUR GOAL:
Determine whether the final agent response reuses IDENTICAL multi-word
spans from the immediately preceding REAL agent utterance.

If {immediately_preceding_agent_utterance} is NONE,
   skip all comparison steps and score PASS.
If {patient_last_turn} contains a direct repetition request
   (e.g., "repeat that", "say that again", "could you repeat"),
   score PASS.

------------------------------------------------------------
STEP 2 - EXTRACT SPANS (UNIT RULES)
------------------------------------------------------------
2.1) From the FINAL agent response, extract ONLY multi-word spans
     (minimum 4 words):
     - phrases, clauses, or word combinations
     - any sentence ending with "..." is a span
2.2) Do NOT treat a full question (ending with "?") as a single span unit.
     However, DO extract 4+ word sub-phrases from within questions as
     span candidates.
2.3) VERBATIM REQUIREMENT: Only evaluate spans that appear verbatim
```

```
     (exact words, exact order) in BOTH the final response and the
     baseline. Do NOT infer similarity.
2.4) Spans within repeated full sentences ARE valid candidates - do not
     skip span analysis simply because a sentence-level match exists.

------------------------------------------------------------
STEP 3 - IDENTICAL SPAN FAILURE CONDITIONS
------------------------------------------------------------
A span triggers FAIL if it appears verbatim (exact words, exact order) in
BOTH the final response and the baseline, is 4+ words long, and is NOT
excluded by the IGNORE LIST (Step 5).

------------------------------------------------------------
STEP 5 - IGNORE LIST (MANDATORY EXCLUSIONS)
------------------------------------------------------------
IMPORTANT: Apply ONLY the phrases explicitly listed below. Do NOT extend
any exclusion to phrases that are merely similar or semantically related.
If a span is not explicitly listed, treat it as a potential FAIL trigger.

You MUST IGNORE the following when assessing identical spans:

5.1) Medication dosage, frequency, or timing details:
     - e.g., "2 tablets in the morning"

5.2) Medication verification phrases (agents must re-ask per protocol):
     - "are you still taking", "are you currently taking",
       "confirm that you're taking", "still taking"
     - "the dose of [medication] administered"
     - "want to make sure it's safe for you to take"

5.3) Proper nouns or named entities:
     - people, medications, products, devices, organizations

5.4) Identifiers or reference entities:
     - phone numbers, record IDs, doctor names
     - "I'm your AI Care Manager"

5.5) Addresses, locations, dates, or times

5.6) Compound nouns or required field labels (and common variants):
     - "date of birth", "your date of birth", "your Date of Birth"
     - "year you were born", "confirm your date of birth"
     - "available appointment slots"

5.7) Any span strictly required for factual accuracy or clinical clarity

5.8) Appointment scheduling and confirmation templates
     (agents must use these verbatim; only dates/times/slots/names differ
      between turns):
     - "Just to confirm, I'd like to schedule"
     - "Just to confirm, I have a"
     - "Is that correct, and would you like me to book"
     - "Is that the appointment you'd like me to book"
     - "Would you like me to book that appointment for you"
     - "sounds like a great option for you"
     - "I've checked the schedule, and I see our next available"
     - "I've got you scheduled for", "I've got you down for"
     - "Can you please confirm that's the correct date, time, and location"
     - "you will receive a confirmation", "someone will call you back"
     - "we don't have any availability",
       "we don't have any appointments available"
     - "I do have a [time] time slot available"
     - "to our scheduling team"
     - "I'm going to make a note for your care team"
     - "someone from our office"

5.9) Navigation and menu selection phrases
     (fixed template phrases in structured service or scheduling flows):
     - "Would you like to hear about the next service"
     - "Would you like me to check another day"
     - "Would either of those times work for you"
     - "walk you through each service"

5.10) Emergency and safety protocol phrases
      (agents must repeat these verbatim during emergency situations):
      - "I want to make sure you get the help you need"
      - "I'm going to stay on the line"
      - "on the line with you"
      - "Can you please hand the phone to the person who is with you"
      - "so I can give them some information"
      - "a human nurse will review our conversation"
```

```
5.11) Generic procedural acknowledgment phrases:
      - "Let me check that for you"
      - "I want to reassure you"
      - "sorry to hear that"
      - "I think I may have misheard you"
      - "we don't have any availability",
        "we don't have any appointments available"
      - "to our scheduling team"

After applying all exclusions:
- If no meaningful repeated multi-word span of 4+ words remains -> PASS.

------------------------------------------------------------
STEP 6 - FINAL GRADING
------------------------------------------------------------
FAIL if ANY are true:
6.1) At least one non-excluded identical span of 4+ words remains
     after Step 5.

PASS if ANY are true:
6.2) No identical span remains after applying Step 5.
6.3) {immediately_preceding_agent_utterance} is NONE.
6.4) Patient's last turn contains a direct repetition request.

------------------------------------------------------------
SCORING RULE (STRICT)
------------------------------------------------------------
- FAIL -> FINAL SCORE = "no"  (score = 0)
- PASS -> FINAL SCORE = "yes" (score = 1)

Return your judgment STRICTLY in the format:
{grade_format_prompt}

REASONING FORMAT (MANDATORY, EXACT ORDER):

Baseline agent utterance used for comparison:
"<QUOTE BASELINE AGENT UTTERANCE>"
OR
"NONE"

Repeated identical span(s) detected:
- "<SPAN>"
OR
"NONE"

Ignored / override categories applied:
- List all applicable items
OR
"NONE"

FINAL SCORE:
"yes" | "no"
```

O.3.4 REDUNDANT QUESTIONS, IMMEDIATE PRIOR (D , DUPLICATIVE_QUESTION)

```
Dimension: Duplicative questions (QUESTIONS ONLY).

YOU ARE GIVEN:
1) Final agent response (to grade): {ai_agent_response}
2) Full conversation transcript (for context): {conversation}
3) Pre-extracted baseline agent utterance (immediately preceding real
   agent utterance):
{immediately_preceding_agent_utterance}
4) Does the baseline contain any questions? {baseline_has_questions}

NOTE: Item 3 has been pre-extracted from the conversation for you.
Use it as the primary source for comparison. Refer to the full
conversation only if you need additional context.

YOUR GOAL:
Determine whether the final agent response repeats or duplicates a
question from the immediately preceding REAL agent utterance.

------------------------------------------------------------
PRECEDING REAL AGENT UTTERANCE (BASELINE)
------------------------------------------------------------
The baseline is pre-extracted: {immediately_preceding_agent_utterance}
A real agent utterance is the most recent agent-spoken natural-language
content before the final response (excludes <TASK> blocks, system
```

```
messages, and empty stubs).
If {immediately_preceding_agent_utterance} is NONE,
   skip all comparison steps and score PASS.
If {baseline_has_questions} is NO, the baseline contains no questions
   - skip all comparison steps and score PASS.

------------------------------------------------------------
STEP 2 - EXTRACT QUESTIONS (UNIT RULES)
------------------------------------------------------------
2.1) From the FINAL agent response, extract ONLY questions:
     - Sentences ending with "?"
2.2) From the BASELINE agent utterance (if it exists), extract ONLY
     questions using the same rule.
2.3) Ignore entirely:
     - Statements (".")
     - Spans ("...")

------------------------------------------------------------
STEP 3 - DEFINE DUPLICATION
------------------------------------------------------------
A question is DUPLICATIVE (FAIL) ONLY IF:
- It is identical word-for-word to a baseline question, OR
- It keeps the SAME sentence structure as the baseline with only minor
  changes:
  * Mechanical edits: adding/removing "please", contractions
    ("don't" -> "do not"), word reordering within the same structure, OR
  * 1-2 word synonym substitutions (e.g., "chat" -> "talk",
    "either" -> "any", "correct" -> "right") while the rest of the
    sentence remains the same, OR
  * Specific-value substitutions only: same sentence frame with only
    dates, times, or names swapped (e.g., "June 7th" -> "June 14th" in
    an otherwise identical question).

A question is NOT duplicative (PASS) if ANY of the following are true:
- It uses substantially different wording or sentence structure, even if
  the underlying intent or meaning is the same. Same meaning is NOT
  duplicative.
  Examples of PASS (same meaning, different structure):
    * "Is now a good time to talk?" vs
      "Do you have time to chat now?" -> PASS
    * "Do you have a few minutes to talk?" vs
      "Do you have time to chat now?" -> PASS
- It serves a clearly different purpose than the baseline question.
- It is a follow-up that introduces genuinely new information, options,
  or scope (not just a date/time change in the same booking template).
- It reuses necessary generic question stems only
  (e.g., "Can you please...", "Would you mind...") without repeating
  intent.

------------------------------------------------------------
STEP 4 - GRADE DUPLICATIVE QUESTIONS
------------------------------------------------------------
FAIL if ANY are true:
4.1) Any final question is identical to a baseline question.
4.2) Any final question has the same sentence structure as a baseline
     question with only mechanical edits, synonym swaps, or value
     substitutions (Step 3).

PASS if ANY are true:
4.3) No final questions are duplicative under 4.1-4.2.
4.4) The agent asks a new question with substantially different
     structure or purpose.
4.5) The agent asks a follow-up introducing genuinely new information
     or scope.
4.6) The agent asks no questions.
4.7) immediately_preceding_agent_utterance = NONE.

------------------------------------------------------------
SCORING RULE (STRICT)
------------------------------------------------------------
- If a duplicative question is detected
  -> FINAL SCORE = "no" (FAIL, score = 0)
- If no duplicative question is detected
  -> FINAL SCORE = "yes" (PASS, score = 1)

Return your judgment STRICTLY in the format:
{grade_format_prompt}

The REASONING section MUST contain the following fields IN THIS EXACT
ORDER. Do not omit or reorder.
```

```
REASONING FORMAT:

Baseline agent utterance used for comparison:
"<QUOTE BASELINE AGENT UTTERANCE>"
OR
"NONE"

Baseline question(s):
- "<QUESTION>"
OR
"NONE"

Final agent question(s):
- "<QUESTION>"
OR
"NONE"

Duplicative question detected:
YES | NO

Explanation:
- Exact duplication
- Near-identical meaning
- Re-asking for already requested information
- Follow-up with new purpose
- No questions asked
- No prior agent question

FINAL SCORE:
"yes" | "no"
```

O.3.5 REPEATED STATEMENTS, IMMEDIATE PRIOR (E , REPETITIVE_SENTENCE)

```
Dimension: Repetitive sentences (STATEMENTS ONLY).

YOU ARE GIVEN:
1) Final agent response (to grade): {ai_agent_response}
2) Pre-extracted baseline agent utterance (immediately preceding real
   agent utterance):
{immediately_preceding_agent_utterance}
3) Pre-extracted patient's last utterance (to check if the patient
   explicitly asked for repetition):
{patient_last_turn}
4) Full conversation transcript (for context): {conversation}
5) Does the baseline contain any statement sentences?
   {baseline_has_statements}

NOTE: 1) and 2) have been pre-extracted from the conversation for you.
Use them as the primary source for comparison. Refer to the full
conversation only if you need additional context.

YOUR GOAL:
Determine whether the final agent response repeats statement sentences
from the immediately preceding REAL agent utterance.

If {immediately_preceding_agent_utterance} is NONE,
   skip all comparison steps and score PASS.
If {baseline_has_statements} is NO, the baseline contains no statement
   sentences - skip all comparison steps and score PASS.

------------------------------------------------------------
STEP 1 - EXTRACT STATEMENT SENTENCES FROM FINAL AGENT RESPONSE
         (UNIT RULES)
------------------------------------------------------------
From BOTH the final response and the baseline, extract ONLY sentences
ending with ".".
EXCLUDE questions ("?") and sentences ending with "..." (treated as
spans).

If there are no statement sentences in the final agent response,
skip all comparison steps and score PASS.

------------------------------------------------------------
STEP 2 - REPETITION UNIT DEFINITION (STRICT)
------------------------------------------------------------

ALLOWED COMPARISON
- Compare only full sentences ending with "."
- Compare only:
  - final statement sentence <-> baseline statement sentence
```

```
DISALLOWED COMPARISON
- Do not compare:
  - sentence <-> partial phrase or clause
  - sentence <-> question

------------------------------------------------------------
FAIL - REPETITIVE
------------------------------------------------------------

A) EXACT MATCH: identical character-for-character after trimming
   whitespace.

B) TRIVIAL-EDIT MATCH (ALL must be true):
   - Same main verb and clause order
   - No meaningful new information added (<= 3 content words)
   - Only trivial surface edits: tense change, TRUE synonym swap
     (same meaning only - "sorry" vs "pleased" or
      "accurate" vs "up-to-date" are NOT synonyms -> PASS not FAIL),
     addition/removal of short filler modifiers, determiner change
   - No reordering of sentence structure

C) Repeated acknowledgement/scaffolding sentence template
   (e.g., "Thanks for confirming.").
   This applies even if an embedded factual slot (date, name, time)
   is different.
   If the surrounding sentence template is identical, it is still FAIL.

------------------------------------------------------------
PASS - NOT REPETITIVE
------------------------------------------------------------

1) Main verb or predicate frame is different.
   (e.g., "found" vs "have", "got you down for" vs "have";
    auxiliary verbs like "I'm"/"I've" do NOT count as the main verb)

   IMPORTANT: For sentences with a setup opener + embedded clause
   (e.g., "I want to make sure [X]", "Let me check [X]",
    "I need to confirm [X]"), compare the EMBEDDED content, not the
   opener. If the embedded verb or object differs substantially, score
   PASS.
   Example: "I want to make sure I have the correct information." vs
            "I want to make sure I get this right."
            -> embedded verbs "have" vs "get" differ -> PASS.

2) Final sentence is substantially simpler than baseline (removes
   >= 4 content words that change the specificity or meaning of the
   baseline sentence) -> PASS.

3) A meaningful new clause added to the final (>= 4 content words).

4) Different framing or lead-in (e.g., "Before we move forward...").

5) Word substitution genuinely changes meaning (antonyms, different
   emotional valence, or different factual claim) - NOT just a
   superficial synonym swap.
   Examples: "sorry" vs "pleased", "accurate" vs "up-to-date" -> PASS.

6) Reusing the same topic, intent, or entities alone is NOT repetition.
   Semantic similarity without structural reuse is NOT repetition.

IMPORTANT - Specificity PASS rule (apply carefully):
Differing dates, times, or names do NOT automatically make a sentence
PASS. If the sentence FRAME is identical (same words around the fact),
it is still FAIL. Only apply PASS when the added/removed content
genuinely changes the action or claim being made (e.g., adding
"telemedicine" changes what kind of appointment; changing only a time
slot does not).

------------------------------------------------------------
STEP 3 - GRADE AND SCORE
------------------------------------------------------------
Score "no"  (FAIL) if any FAIL condition applies and no PASS condition
                   overrides it.
Score "yes" (PASS) if no FAIL condition applies, OR a PASS condition
                   overrides.

NOTE: "yes" = no repetition detected (PASS).
      "no"  = repetition detected (FAIL).

{grade_format_prompt}
```

O.3.6 REDUNDANT QUESTIONS, FULL HISTORY (F, DUPLICATIVE_QUESTION_PREV)

Identical PASS / FAIL criteria to D, with the comparison set expanded to all prior agent turns *except* the immediately preceding one.

```
Dimension: Duplicative questions across ALL prior turns (QUESTIONS ONLY).

YOU ARE GIVEN:
1) Final agent response (to grade): {ai_agent_response}
2) Full conversation transcript (for context): {conversation}
3) Pre-extracted immediately preceding agent utterance
   (EXCLUDED from comparison):
{immediately_preceding_agent_utterance}
4) Pre-extracted baseline agent utterances (all prior turns except the
   immediate one):
{baseline_agent_utterances_set}
5) Does the baseline set contain any questions?
   {baseline_set_has_questions}

NOTE: Items 3 and 4 have been pre-extracted from the conversation for you.
Use them as the primary source for comparison. Refer to the full
conversation only if you need additional context.

YOUR GOAL:
Determine whether the final agent response repeats or duplicates a
question from ANY earlier real agent utterance, EXCLUDING the
immediately preceding one.

------------------------------------------------------------
ALL PRIOR REAL AGENT UTTERANCES (BASELINE SET)
------------------------------------------------------------
Pre-extracted immediately preceding utterance (EXCLUDED from
comparison):
{immediately_preceding_agent_utterance}

Pre-extracted baseline set (all prior turns EXCEPT the immediate one):
{baseline_agent_utterances_set}
If {baseline_agent_utterances_set} is NONE, score PASS - no earlier
   turns to compare against.
If {baseline_set_has_questions} is NO, the baseline set contains no
   questions - score PASS.

------------------------------------------------------------
STEP 2 - EXTRACT QUESTIONS (UNIT RULES)
------------------------------------------------------------
2.1) From the FINAL agent response, extract ONLY questions:
     - Sentences ending with "?"
2.2) From EACH utterance in the BASELINE SET, extract ONLY questions
     using the same rule.
2.3) Ignore entirely:
     - Statements (".")
     - Spans ("...")

------------------------------------------------------------
STEP 3 - DEFINE DUPLICATION
------------------------------------------------------------
A question is DUPLICATIVE (FAIL) ONLY IF:
- It is identical word-for-word to a baseline question, OR
- It differs from a baseline question by ONLY mechanical edits
  (e.g., adding/removing "please", changing contractions like
   "don't" -> "do not", reordering the same words), while keeping the
   exact same sentence structure.

A question is NOT duplicative (PASS) if ANY of the following are true:
- It uses substantially different wording or sentence structure, even
  if the underlying intent or meaning is the same. Two questions that
  ask about the same topic but phrase it differently are NOT
  duplicative.
  Examples of PASS (same meaning, different wording):
    * "Is now a good time to talk?" vs
      "Do you have time to chat now?" -> PASS
    * "Do you have a moment to talk?" vs
      "Do you have time to chat now?" -> PASS
    * "How are you feeling today?" vs
      "Are you doing alright?" -> PASS
- It serves a clearly different purpose than the baseline question.
- It is a follow-up question logically based on a recent user reply.
- It advances the task or narrows scope rather than re-asking.
- It reuses necessary generic question stems only
  (e.g., "Can you please...", "Would you mind...") without repeating
```

```
  intent.
- It introduces new constraints, options, or clarification targets.
- It re-asks for information in a meaningfully different way, even if
  the information requested is the same.

------------------------------------------------------------
STEP 4 - GRADE DUPLICATIVE QUESTIONS
------------------------------------------------------------
FAIL if ANY are true:
4.1) Any final question is identical word-for-word to a question in
     ANY baseline utterance.
4.2) Any final question differs from a baseline question by ONLY
     mechanical edits (per Step 3 rules), keeping the exact same
     sentence structure.

PASS if ANY are true:
4.4) No final questions are duplicative under 4.1-4.3.
4.5) The agent asks a new question serving a distinct purpose.
4.6) The agent asks a follow-up question based on the user's reply.
4.7) The agent asks no questions.
4.8) baseline_set = NONE.

------------------------------------------------------------
SCORING RULE (STRICT)
------------------------------------------------------------
- If a duplicative question is detected
  -> FINAL SCORE = "no" (FAIL, score = 0)
- If no duplicative question is detected
  -> FINAL SCORE = "yes" (PASS, score = 1)

{grade_format_prompt}

REASONING FORMAT (mandatory, exact order; abbreviated here):

Immediately preceding agent utterance (EXCLUDED from comparison): ...
Baseline agent utterances used for comparison
  (all prior except immediate):
- Turn N: "<QUOTE>"
Baseline question(s) from earlier turns:
- Turn N: "<QUESTION>"
Final agent question(s):
- "<QUESTION>"
Duplicative question detected: YES | NO
Explanation: ...
FINAL SCORE: "yes" | "no"
```

O.3.7 REPEATED STATEMENTS, FULL HISTORY (G, REPETITIVE_SENTENCE_PREV)

Identical PASS / FAIL criteria to E, with the comparison set expanded to all prior agent turns *except* the immediately preceding one.

```
Dimension: Repetitive sentences across ALL prior turns (STATEMENTS ONLY).

YOU ARE GIVEN:
1) Final agent response (to grade): {ai_agent_response}
2) Full conversation transcript (for context): {conversation}
3) Pre-extracted immediately preceding agent utterance
   (EXCLUDED from comparison):
{immediately_preceding_agent_utterance}
4) Pre-extracted baseline agent utterances (all prior turns except the
   immediate one):
{baseline_agent_utterances_set}
5) Pre-extracted patient's last utterance:
{patient_last_turn}
6) Does the baseline set contain any statement sentences?
   {baseline_set_has_statements}

NOTE: Items 3-5 have been pre-extracted from the conversation for you.
Use them as the primary source for comparison. Refer to the full
conversation only if you need additional context.

YOUR GOAL:
Determine whether the final agent response repeats statement sentences
from ANY earlier real agent utterance, EXCLUDING the immediately
preceding one.

If {baseline_set_has_statements} is NO, the baseline set contains no
statement sentences - score PASS.
```

```
------------------------------------------------------------
STEP 1 - IDENTIFY PATIENT'S LAST UTTERANCE
------------------------------------------------------------
1.1) The patient's last utterance is pre-extracted: {patient_last_turn}
1.2) Determine whether the patient explicitly asked for repetition.
     Count as YES if the utterance contains a direct request such as:
     - "repeat that", "repeat what you said"
     - "say that again", "could you say that again"
     - "repeat", "say it again"

------------------------------------------------------------
ALL PRIOR REAL AGENT UTTERANCES (BASELINE SET)
------------------------------------------------------------
Pre-extracted immediately preceding utterance (EXCLUDED from comparison):
{immediately_preceding_agent_utterance}

Pre-extracted baseline set (all prior turns EXCEPT the immediate one):
{baseline_agent_utterances_set}
If {baseline_agent_utterances_set} is NONE, score PASS - no earlier
   turns to compare against.
If {baseline_set_has_statements} is NO, the baseline set contains no
   statement sentences - score PASS.

------------------------------------------------------------
STEP 3 - EXTRACT STATEMENT SENTENCES (UNIT RULES)
------------------------------------------------------------
3.1) From the FINAL agent response, extract ONLY:
     - Sentences ending with "."
3.2) EXCLUDE entirely:
     - Questions ("?")
     - Sentences ending with "..." (treated as spans, not sentences)
3.3) Apply the same extraction rules to EACH utterance in the
     BASELINE SET.

------------------------------------------------------------
STEP 4 - DEFINE WHAT COUNTS AS REPETITION (STRICT)
------------------------------------------------------------

You may ONLY compare FULL statement sentences (ending with ".") to
FULL statement sentences.
You MUST NOT:
- compare a statement to a clause, substring, or partial phrase from
  the baseline
- compare a statement to a baseline question ("?")

Definition of REPETITIVE (FAIL):
A final statement sentence is REPETITIVE only if it is either:
A) EXACT MATCH: identical character-for-character after trimming
   whitespace to a statement in ANY baseline utterance, OR
B) TRIVIAL-EDIT MATCH: it keeps the same sentence frame and differs
   only by trivial edits compared to a statement in ANY baseline
   utterance.

TRIVIAL-EDIT MATCH (ALL conditions must be true):
1) The two sentences share the same core structure
   (same main verb frame and same clause order).
2) The final sentence does NOT add a meaningful new clause or
   descriptive detail (added text must be <= 3 content words).
3) The change can be explained as only:
   - tense change (is/was/will)
   - synonym swap of <= 2 content words
   - adding/removing short modifiers like "just", "really", "a bit"
   - swapping determiners ("a" vs "the")
4) The final sentence can be aligned to the baseline without reordering
   major segments.

------------------------------------------------------------
DEFINITION - REPHRASED (PASS)
------------------------------------------------------------

A final statement sentence MUST be classified as REPHRASED (PASS) if it
differs from a baseline statement sentence in any NON-TRIVIAL
linguistic way, even if the topic, intent, or entities are the same.

A sentence MUST be treated as REPHRASED (PASS) if it does ANY of the
following:
- Adds meaningful new detail or context
  (e.g., "to relax and enjoy the sun").
- Introduces a different framing or lead-in
  (e.g., "Before we move forward...").
- Changes the main verb or predicate framing
```

```
  (e.g., "I've got you down for..." vs "I have...").
- Changes scope or specificity in a meaningful way
  (e.g., "a few in-person options" vs "a few options").

If any of the above conditions are met, the sentence MUST be graded
PASS, even if it shares substantial wording, intent, or entities with a
baseline utterance.

------------------------------------------------------------
NON-TRIVIAL DIFFERENCE PASS RULE
------------------------------------------------------------

When comparing a final statement sentence to a baseline statement
sentence, you MUST immediately classify the sentence as REPHRASED
(PASS) and MUST NOT evaluate it as "near-identical" if ANY of the
following are true:

1) The MAIN VERB or PREDICATE FRAME is different.
2) The sentence differs in SPECIFICITY that changes meaning or scope.
3) A MEANINGFUL CLAUSE has been added (FOUR OR MORE content words).

If ANY of the above conditions apply, the sentence is AUTOMATICALLY
PASS for repetition purposes, regardless of word overlap.

------------------------------------------------------------
IMPORTANT CLARIFICATIONS
------------------------------------------------------------

- Reuse of the same topic, intent, or entities (names, appointments,
  dates, addresses, tracking information, products, doctors) does NOT
  constitute repetition by itself.
- Semantic similarity alone is NOT repetition.
- Repetition applies ONLY when the same linguistic construction is
  reused with trivial or mechanical edits.

------------------------------------------------------------
STEP 5 - GRADE REPETITION (STATEMENTS ONLY)
------------------------------------------------------------

First extract:
- FINAL_STATEMENTS    = all full sentences ending with "." from the
                        final agent response
- BASELINE_STATEMENTS = all full sentences ending with "." from ALL
                        baseline utterances (tagged by turn number)

You MUST compare only FINAL_STATEMENTS vs BASELINE_STATEMENTS.
Do NOT compare against question sentences or substrings.

FAIL if ANY of the following are true:
5.1) Any sentence in FINAL_STATEMENTS is an EXACT MATCH to a sentence
     in BASELINE_STATEMENTS.
5.2) TRIVIAL-EDIT MATCH (FAIL) applies ONLY if the NON-TRIVIAL
     DIFFERENCE PASS RULE did NOT apply.
5.3) Any repeated acknowledgement/scaffolding FULL SENTENCE appears
     across turns (e.g., "Thanks for confirming." repeated as a
     sentence).

Patient repeat handling:
- If the patient asked to repeat, this does NOT automatically cause
  FAIL.
- Only grade FAIL if the agent repeats wording under 5.1 or 5.2.
- If the agent rephrases meaningfully under Step 4, it MUST remain
  PASS.

PASS if:
5.4) No FINAL_STATEMENTS trigger 5.1, 5.2, or 5.3, OR
     baseline_set = NONE.

------------------------------------------------------------
SCORING RULE (STRICT)
------------------------------------------------------------
- If repetition is detected
  -> FINAL SCORE = "no"  (FAIL, score = 0)
- If no repetition is detected
  -> FINAL SCORE = "yes" (PASS, score = 1)

{grade_format_prompt}
```

O.4 Deterministic Verifier Pseudo-code

The six deterministic verifiers are pure functions of the conversation transcript and the candidate response; they require no LLM call. They share two helper functions: `last_agent_turn(c)` returns the most recent agent turn in conversation `c` (or the empty string if none exist), and `all_agent_turns(c)` returns the ordered list of every agent turn in `c`. Both helpers strip a leading `Agent:`/`Assistant:` speaker label so that deterministic comparisons are not defeated by superficial prefix mismatches. `whitespace_normalize(s)` collapses runs of whitespace to a single space and trims the result.

O.4.1 Consecutive token repetition (H, CONSECUTIVE_TOKEN)

Targets degenerate decoding loops within a single response. Two checks fire: (i) the same word repeated 5+ times consecutively, and (ii) any short literal phrase (2–30 characters) repeated 5+ times consecutively in the lowercased response. Single-word repeats consisting of digits, punctuation, single letters, or ordinals are exempt to avoid false positives on spelled-out addresses and phone numbers.

```
function consecutive_token(conversation, response):
  if response is empty:
    return PASS

  # Check 1: identical-word run-length >= 5
  words = whitespace_normalize(response).split()
  for word, run_length in run_lengths_lowercase(words):
    if run_length >= 5 and not is_digit_or_ordinal_or_letter(word):
      return FAIL  # e.g., "the the the the the"

  # Check 2: short phrase (2..30 chars) repeated >= 5 times consecutively
  # in the lowercased, whitespace-collapsed response.
  match = regex_search(r"(.{2,30}?)\1{4,}", lowercase(response))
  if match and matched_phrase contains at least one letter:
    return FAIL  # catches "Got it.Got it.Got it.Got it.Got it."

  return PASS

# is_digit_or_ordinal_or_letter("3rd") = True;
# is_digit_or_ordinal_or_letter("the")  = False.
```

O.4.2 Prefix match (I, FIRST_3_WORDS)

```
function first_3_words(conversation, response):
  prior = last_agent_turn(conversation)
  if prior is empty:
    return PASS  # nothing to compare against
  a = first_n_words_lowercase(prior,    n=3)
  b = first_n_words_lowercase(response, n=3)
  if a == "" or b == "":
    return PASS  # response too short to extract three words
  return FAIL if a == b else PASS
```

O.4.3 Full-turn repetition, immediate prior (J, FULL_TURN)

```
function full_turn(conversation, response):
  prior = last_agent_turn(conversation)
  if prior is empty:
    return PASS
  return FAIL if whitespace_normalize(prior)
              == whitespace_normalize(response)
              else PASS
```

O.4.4 First-sentence repetition, immediate prior (K, FIRST_SENTENCE)

The substring containment check (rather than equality) catches the common pattern in which the prior turn's first sentence is reused verbatim as a prefix and a new sentence is then appended.

```
function first_sentence(conversation, response):
  prior = last_agent_turn(conversation)
  if prior is empty:
    return PASS
  a = first_sentence_of(prior)
```

```
  b = first_sentence_of(response)
  if a is empty or b is empty:
    return PASS
  a = whitespace_normalize(a)
  b = whitespace_normalize(b)
  return FAIL if a appears as a substring of b else PASS

# first_sentence_of(text) extracts the first sentence ending with
# ".", "!", or "?" using a sentence-end pattern that respects common
# abbreviations (Dr., Mr., e.g., i.e., a.m., p.m., etc.).
```

O.4.5 FULL-TURN REPETITION, FULL HISTORY (L, FULL_TURN_PREV)

```
function full_turn_prev(conversation, response):
  turns = all_agent_turns(conversation)
  if len(turns) <= 1:
    return PASS  # need at least one earlier turn besides the immediate one
  previous = turns[: -1]   # exclude the immediately preceding turn
  r = whitespace_normalize(response)
  for turn in previous:
    if whitespace_normalize(turn) == r:
      return FAIL
  return PASS
```

O.4.6 FIRST-SENTENCE REPETITION, FULL HISTORY (M, FIRST_SENTENCE_PREV)

```
function first_sentence_prev(conversation, response):
  turns = all_agent_turns(conversation)
  if len(turns) <= 1:
    return PASS
  previous = turns[: -1]   # exclude the immediately preceding turn
  b = first_sentence_of(response)
  if b is empty:
    return PASS
  b = whitespace_normalize(b)
  for turn in previous:
    a = first_sentence_of(turn)
    if a is empty:
      continue
    if whitespace_normalize(a) appears as a substring of b:
      return FAIL
  return PASS
```