RESIMA: A RECURSIVE SELF-IMPROVING MULTI-AGENT LLM-AS-A-JUDGE FRAMEWORK FOR HALLUCINATION AND OMISSION DETECTION AND CORRECTION IN CLINICAL NOTES

Mehrshad Tavana[1], **Samridhi Vaid, Mohamed Abdalla**[1], **Randy Goebel**[1,2], **J. Ross Mitchell**[1,2]
[1]University of Alberta [2]Alberta Machine Intelligence Institute
{atavana, svaid, mabdall2, rgoebel, jmitche2}@ualberta.ca

ABSTRACT

LLM-as-a-Judge is almost always a single model call, but verifying a long generation against its source is not a single task. It requires two asymmetric searches: a precision check that every claim in the output is grounded in the source, and a recall check that every salient fact in the source survives into the output. We show these searches interfere when merged into one call, and that separating them matters more than iteration does. We present RESIMA, a recursive self-improving multi-agent judge. Three detection agents run in parallel, one per comparison direction, and a correction agent edits the generation from their merged findings. A meta-agent then rewrites the detection prompts between iterations using the pipeline's own error traces, closing a feedback loop over the judge's instructions rather than its outputs. We evaluate on reference-free clinical documentation, where errors are both verifiable and consequential, using injected-error transcripts, an out-of-domain transfer set, and expert-labeled production outputs. RESIMA recovers the large majority of injected errors and agrees with expert labels more than the experts agree with each other. An ablation isolating decomposition from iteration shows decomposition is the larger source of the gain, and the full pipeline substantially outperforms a single-pass judge, especially on weaker models. This failure mode is not specific to clinical text: it applies to any judge verifying a long output against a long source in a single pass.

1 INTRODUCTION

Physicians spend nearly two hours on electronic health record and desk work for every hour spent with a patient (Sinsky et al., 2016), and clerical load from the record is independently associated with a 40% higher risk of burnout (Shanafelt et al., 2016). Ambient AI scribes answer this directly: a large language model reads a transcript of the encounter and drafts the note (Tierney et al., 2024; Liu et al., 2025). The failure mode is equally direct. When Williams et al. (2025) asked GPT-4 to summarize 100 emergency department encounters, 42% of the resulting notes contained a hallucinated detail and 47% omitted something clinically relevant, embedded in prose fluent enough that a busy clinician has little reason to look twice.

Catching those errors is not a summarization-evaluation problem in the usual sense. Reference-based metrics compare a candidate against a human-written summary (Papineni et al., 2002; Lin, 2004; Zhang et al., 2020; Sellam et al., 2020), but production deployments have no such reference: the transcript is the only ground truth, and the note under review may contain errors no human writer would have made. Asgari et al. (2025) measured this gap directly, finding hallucination and omission rates across 18 clinical summarization configurations to be uncorrelated with BLEU, ROUGE, and related automatic scores. The natural alternative is to use a language model as the judge (Zheng et al.,

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.

2023; Gu et al., 2024; Li et al., 2024), and clinical work has followed that path (Brake & Schaaf, 2024; Kanithi et al., 2024). Almost all of it uses the same design: one model call, given the source, the candidate, and a list of error types, asked to return every error it can find.

We argue that this design has a structural blind spot, and that the blind spot is not where aggregate metrics suggest. Verifying a note against a transcript is not one comparison but three, running in different directions. Asking whether a note claim is supported by the transcript (note-to-transcript) starts from a span and searches for evidence. Asking whether a transcript fact survived into the note (transcript-to-note) starts from the transcript and searches the note, and, critically, its positive findings leave no trace on the note side. Asking whether shared content was altered in transit (paired comparison) requires holding two versions of the same fact in working memory at once. A single merged prompt must interleave all three, and the direction it quietly abandons first is the one with no note-side anchor. A hallucination a detector skips leaves the offending sentence sitting in the note for a later pass to find; an omission it never searches for leaves nothing behind at all.

This paper tests that argument rather than assuming it. We present **RESIMA**, a recursive self-improving multi-agent judge that assigns each comparison direction its own detector, feeds the merged findings to a correction agent under an explicit no-fabrication and structure-lock policy, and lets a prompt self-improvement agent rewrite the detection prompts between passes. We then take the pipeline apart with an ablation ladder that adds its three design commitments (decomposition, iteration, and prompt self-improvement) one at a time against a matched single-pass baseline that reads the concatenation of the same three prompts in one call, with the judge model, correction agent, decoding configuration, and span-matching threshold held fixed throughout.

Contributions.

- **Architecture.** RESIMA, a judge that decomposes detection by comparison direction rather than by note section, coupled to a correction agent and a gradient-free prompt self-improvement loop (Section 3).
- **Controlled ablation.** At equal prompt content, decomposition is worth $+8.8$ pp and $+9.9$ pp for a closed- and an open-weight judge, roughly twice the value of iteration and sixteen times its token efficiency, and the full pipeline gains $+19.1$ pp and $+21.5$ pp over the single-pass baseline (Section 5).
- **Mechanism.** The single-pass judge loses accuracy as error density rises (-12.8 pp across density bins, against -1.7 pp for the decomposed judge), misattributes categories three to four times more often, and inverts the recall-over-precision ordering a safety screen requires (Section 6).
- **Validation at deployment scale.** Across 8,350 labels, two judge models, and three corpora including 15 expert-annotated production notes, the decomposed judge clears expert-expert agreement ($\kappa = 0.87$ vs. 0.69) where the single-pass judge does not (0.67); we release all agent prompts (Section 7, Section K).

2 Related Work

Faithfulness in clinical text. Huang et al. (2024) separate intrinsic hallucinations, which contradict the source, from extrinsic ones, which add unverifiable content (Maynez et al., 2020). Benchmarks formalize parts of this: Med-HALT (Pal et al., 2023), HaluEval (Li et al., 2023), and MED-OMIT (Schumacher et al., 2023). Das et al. (2025) report that open-weight models capture admission reasons but systematically drop follow-up recommendations, and Suhas et al. (2025) find that general-domain hallucination detectors transfer poorly to clinical text. We treat omission and hallucination as separate *search directions* rather than as separate labels within one task.

LLM-as-a-Judge and its reliability. Judging with a model scales where human review does not (Zheng et al., 2023; Gu et al., 2024; Li et al., 2024). Clinical variants report encouraging agreement ($\kappa = 0.79$ on SOAP note consistency (Brake & Schaaf, 2024); a reference-free clinical framework (Kanithi et al., 2024)), but the reliability costs are well catalogued: expert agreement of only 64-68% on knowledge-intensive clinical tasks (Szymanski et al., 2024), at least twelve bias types (Ye et al., 2024), instability even at zero temperature (Schroeder & Wood-Doughty, 2024), a decoy effect in

medical misinformation (Liu & He, 2024), and hallucinations inside the judge's own output (Curran et al., 2024). These findings motivate structural remedies rather than better single prompts.

Decomposition, iteration, and prompt optimization. Three lines of work supply the commitments we disentangle. Multi-agent decomposition improves consistency over single-model baselines (Darwish et al., 2025; Du et al., 2024); iterative self-correction (Madaan et al., 2023; Shinn et al., 2023) and structured reasoning (Wei et al., 2022; Yao et al., 2023) improve outputs by re-running a model on its own results; and automatic prompt optimization searches instruction space directly (Zhou et al., 2023; Pryzant et al., 2023), with output formatting alone changing judge accuracy materially (Laskar et al., 2025). Agentic pipelines typically ship all three together, so published gains cannot be attributed to any one. The ablation in Section 5 separates them for a judge task at equal prompt content; its ordering (decomposition first, iteration last and most expensive) is the result we would most want a practitioner to carry away.

3 METHOD

3.1 PROBLEM SETTING

A clinical encounter is a transcript T, the clinician-patient exchange produced by automatic speech recognition, and a generated note N, a structured document with sections for history of presenting illness, physical examination, and impression and plan. In production N is written automatically from T with no human-authored reference available (Tierney et al., 2024; Liu et al., 2025). Given (T, N) we address two tasks: **detection**, identifying clinically significant discrepancies between N and T, and **correction**, producing a refined note N' that removes unsupported content, repairs errors, restores clinically significant omissions, and preserves the structure of N. Throughout, T is the sole source of truth.

A **hallucination** ($N \to T$) is a claim in N with no support anywhere in T: a fabricated medication, a fabricated pertinent negative, boilerplate discharge language. An **omission** ($T \to N$) is a clinically relevant fact present in T that does not appear in N in any form, including paraphrase. A **factual error** ($T \leftrightarrow N$) is content present in both but materially altered in transit, or a transcription error in T propagated into N. Omission is the only one of the three whose positive evidence is an absence on the note side, which is the property the rest of this paper turns on.

3.2 AGENTIC JUDGE PIPELINE

Figure 1(b) shows the pipeline. Each iteration dispatches three detection agents concurrently, one per comparison direction, and passes their merged findings to a correction agent; the loop runs for at most eight iterations and exits early when a pass returns no new finding. Every agent call is agentic rather than a single prompt-to-response mapping: it observes the inputs, plans which parts of the note warrant scrutiny, acts by cross-referencing content and drafting a structured finding, and re-checks flagged spans before returning. All detection agents emit JSON with fields `location`, `error`, `rationale`, and `exact_content`; requiring a verbatim span is what makes automated matching against ground truth possible.

Agent 1: hallucination detection ($N \to T$). The agent reads the note claim by claim and asks, for each, whether the transcript supports it anywhere in any wording. It reports only content that is absent from the transcript entirely; the moment a fact is present but stated wrongly, an explicit boundary rule hands the span to Agent 3.

Agent 2: omission detection ($T \to N$). The agent first extracts clinically relevant facts from the transcript (chief complaint details, pertinent negatives, review of systems, history, medications, allergies, social and family history, examination findings, investigations, plan elements), then searches the whole note for each, allowing paraphrase, and only then applies a clinical-significance filter. Extraction precedes search deliberately: an agent that reads the note first anchors on the note's structure and stops looking for categories the note does not mention. Pertinent negatives receive an explicit checklist, since they are the easiest content to drop silently.

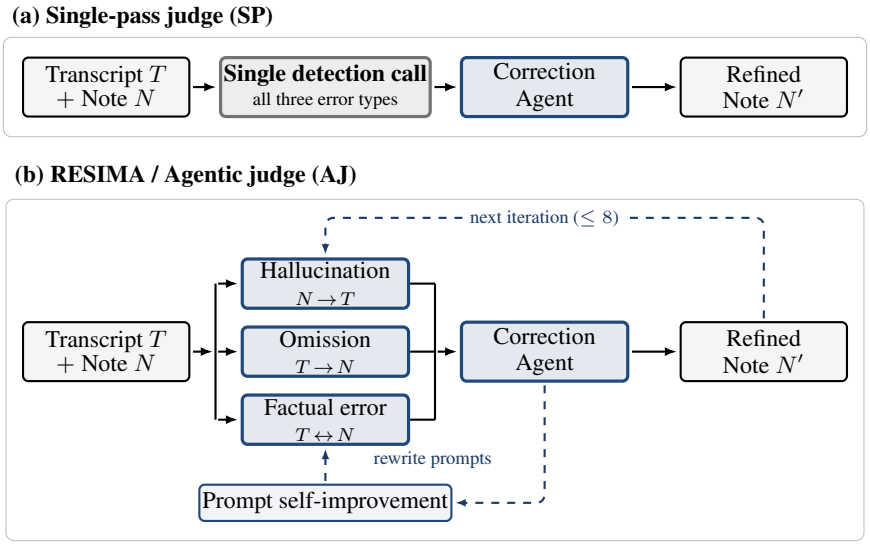

Figure 1: Architectures compared in the ablation. (a) One merged detection call. (b) RESIMA: one agent per comparison direction, then correction, with optional iteration and prompt rewrite. Correction, judge model, and span matching are identical in both.

Agent 3: factual error detection ($T \leftrightarrow N$). The agent enumerates every stated clinical value in the transcript (durations, doses, frequencies, severities, laterality, trajectories, named medications) before reading the note, so the comparison is value-driven rather than sentence-driven. It emits two labels: an error for a note-side distortion, and a transcript error for an apparent speech-recognition failure in T itself, the one case in the pipeline where the source of truth is under suspicion. Its boundary rule is the converse of Agent 1's, and every finding must name the transcript value it disagrees with, which makes a boundary violation visible rather than silent.

Correction agent. The correction agent receives the note and the merged findings and edits under five rules: (i) no fabrication, (ii) restore omissions using concise clinical language, (iii) structure lock on headings, order, and formatting, (iv) minimal edits, and (v) repair clear transcription errors propagated into the note when clinical intent is recoverable. It returns only the refined note, which re-enters the loop for the next detection pass.

Prompt self-improvement agent. After detection and correction complete, this agent inspects the iteration's own detections over a sampled batch and decides, independently for each of the three detection prompts, whether a systematic weakness is present. It is instructed to act on patterns rather than incidents, with a two-case recurrence bar, and it may revise one, two, three, or none of the prompts. Revisions must preserve the required template placeholders and output schema; a validity check discards any rewrite that drops a placeholder or renames a schema field, so a bad revision costs one iteration instead of breaking the pipeline. The agent operates on the *prompts* after a full cycle, whereas the correction agent operates on the *note* after each detection pass. All prompts are reproduced in Section K.

3.3 Single-Pass Baseline and Ablation Ladder

The pipeline makes three commitments at once, and any result reported on the assembled system confounds them. We therefore specify a matched single-pass judge (Figure 1a) and a ladder that adds the commitments one at a time.

The single-pass judge is the conventional configuration: one model call, no loop, no prompt adaptation. Its prompt is the concatenation of the three detection prompts: the same source-of-truth principle, the same label definitions and worked examples, the same hotspot checklists, the same verbatim-span requirement. Exactly three things change. The reciprocal boundary rules, which in the agentic design

hand a span to one of the other two agents, become a single instruction to assign each finding a `category` field; the three JSON schemas merge into one object carrying three arrays; and the agent is told to work through the note once, reporting all three error types as it encounters them. Everything downstream is untouched.

The ladder has four rungs. **SP-1** is the single merged detection call. **AJ-1** splits that call into three parallel agents at a fixed budget of one pass. **AJ-8** adds iteration (at most eight passes; early exit when a pass returns no new finding). **AJ-8+SI** adds prompt self-improvement and is the assembled pipeline. Each step adds exactly one design commitment.

Three controls make the ladder interpretable. First, prompt content is held fixed within each dataset's ladder: the first three rungs run the initial hand-written prompts, so AJ-8 is exactly the iteration-1 configuration of the self-improvement experiment and the final step is that experiment's converged gain. Second, the judge model, decoding configuration, correction agent, and matching threshold are identical across rungs, so within a ladder the only variable is the commitment being added. Third, every rung runs on all three datasets with both judge models.

Two limits are worth stating before any numbers. **Decomposition changes two things at once**: it gives each direction its own instruction context *and* its own generation budget. Section 6 reports measurements bearing on which dominates, but the ladder attributes the combined effect to decomposition. **The ladder is sequential, not factorial**: iteration and self-improvement have no single-pass counterpart, so each contrast is measured with the commitments below it already in place. Only SP-1 $\rightarrow$ AJ-1 is a strict single-variable comparison.

4 Experimental Setup

Datasets. We evaluate on three corpora totalling 8,350 scored labels (Section B). The **ED benchmark** contains 272 emergency department transcript-note pairs in which an injection agent (Gemini 2.5 Pro) corrupts an original note under a fixed budget of 24-31 errors per case while never adding true transcript information absent from the original (6,528 scored labels). The injected distribution is calibrated to clinical collaborators' account of production failure modes: roughly 47% omissions, 33% hallucinations, 20% factual errors. **PriMock57** is a public corpus of 57 mock primary-care consultations (Babylon Health, 2019), processed identically (1,386 labels). The **real subset** is 15 production notes from the Alberta Health Services Jenkins Scribe (built on GPT-4.5; 436 labels), annotated in Label Studio by three clinicians; five cases were independently double-annotated, and labels are merged by union. No synthetic errors are injected into the real subset.

Models. The closed-weight judge is Gemini 2.5 Pro, orchestrated through the Google Agent Development Kit (Google, 2025) with prompts stored in version-controlled text files. The open-weight judge is Llama 3.3 70B Instruct (Dubey et al., 2024) served through a vLLM OpenAI-compatible endpoint with byte-identical prompts, which makes the cloud-versus-on-premises comparison relevant to privacy-sensitive health deployments. Both run at temperature 0.

Detection metric. Ground-truth error records are aligned to judge outputs by cosine similarity between the ground-truth `exact_content` field and the reported span, at an 80% threshold. Matching is category-scoped: a span whose text clears the threshold but whose category label differs is scored as a miss, which is conservative, because a hallucination reported as a factual error sends the correction agent a different instruction than the error requires. Match rate is the fraction of ground-truth errors matched, and is recall on the ground-truth error set. To keep the cost of category-scoping visible rather than silent, we additionally report the **category misattribution rate**: the share of reported spans that clear the threshold against some ground-truth error but are filed under a different category.

Agreement against human annotation. On the real subset the reference point is pairwise Cohen's κ between two clinicians. For the judge the unit is the identified span and chance agreement is negligible, so $\kappa \approx p_o$, numerically the category-scoped match rate. The human coefficient is chance-corrected over three labels after span alignment and is the more conservative of the two; we treat the comparison as indicative rather than as a ceiling test (Section N).

Config.	Model	Halluc. ↑	Omission ↑	Factual ↑	Overall ↑	Δ vs. SP-1
SP-1	Gemini	82.5 ± 0.6	71.2 ± 0.8	76.2 ± 0.4	76.7 ± 0.5	-
	Llama	76.6 ± 0.7	62.0 ± 1.0	69.7 ± 0.5	69.4 ± 0.7	-
AJ-1	Gemini	88.3 ± 0.5	82.7 ± 0.7	85.4 ± 0.3	85.5 ± 0.4	$+8.8$
	Llama	83.5 ± 0.6	75.4 ± 0.8	78.8 ± 0.4	79.3 ± 0.5	$+9.9$
AJ-8	Gemini	91.7 ± 0.4	88.2 ± 0.6	89.7 ± 0.3	89.9 ± 0.6	$+13.2$
	Llama	87.6 ± 0.5	81.6 ± 0.7	84.1 ± 0.3	84.4 ± 0.7	$+15.0$
AJ-8+SI	Gemini	$\mathbf{96.9 \pm 0.3}$	$\mathbf{94.7 \pm 0.4}$	$\mathbf{95.7 \pm 0.2}$	$\mathbf{95.8 \pm 0.4}$	$\mathbf{+19.1}$
	Llama	$\mathbf{93.0 \pm 0.4}$	$\mathbf{89.0 \pm 0.5}$	$\mathbf{90.7 \pm 0.2}$	$\mathbf{90.9 \pm 0.5}$	$\mathbf{+21.5}$

Table 1: Detection match rate (%, mean ± SEM) for the four ablation rungs on the 272-case benchmark. Higher is better; best per model in bold. Prompt content, judge model, correction agent, and matching threshold are identical across rungs.

Correction metrics. Refined notes are compared to the original uncorrupted note (synthetic) or the expert-corrected reference (real). **Line-based word evaluation** aligns lines across the original, injected, and refined notes by TF-IDF cosine similarity and classifies words within aligned lines after stop-word filtering, yielding word-level precision, recall, F1, and accuracy. **N-gram overlap** reports ROUGE-1, ROUGE-L, and BLEU-1 (Papineni et al., 2002; Lin, 2004).

Cost accounting and statistics. For every configuration we log completed model calls, prompt and completion tokens, and wall-clock latency per case, measured on the same hardware and endpoints as the accuracy results; detection agents are dispatched concurrently, so tokens and latency scale differently with decomposition. Configurations are compared on paired case-level data. We report point estimates in percentage points with 95% percentile bootstrap confidence intervals over 1,000 case-level resamples and two-sided paired bootstrap p-values, treating $p < 0.05$ as significant. Given its size, we describe differences on the 15-case real subset as suggestive unless the interval excludes zero by a clear margin. Each configuration was run once, so we report differences between configurations rather than variability across seeds.

5 What Decomposition Buys

Table 1 reports the full ladder on the 272-case benchmark for both judge models. The single-pass baseline is not a weak system: at $76.7 \pm 0.5\%$ it recovers more than three of every four injected errors while running the same instruction content the agentic pipeline starts from, in two model calls. It is nevertheless 19.1 pp below the assembled pipeline, and 21.5 pp below for the open-weight judge.

Decomposition is the largest and the cheapest of the three commitments. Table 2 reports each sequential contrast with paired bootstrap intervals alongside what it costs. Splitting the merged call into three agents, with no extra passes and no prompt adaptation, recovers $+8.8$ pp and $+9.9$ pp, roughly twice what iteration contributes and about half again what self-improvement contributes, and every interval excludes zero by a wide margin. Expressed as percentage points per ten thousand additional tokens, decomposition returns 10.5 and self-improvement 10.9, against 0.7 for iteration: a factor of roughly sixteen. Decomposition and self-improvement are cheap because they change *what* the model is asked; iteration is expensive because it multiplies the whole pipeline. Latency behaves differently again, because the three agents run concurrently: AJ-1 completes in 52 seconds against SP-1's 38, so decomposition is nearly free in wall-clock terms even where it triples detection tokens, whereas the loop takes the pipeline from under a minute to about four.

The gain concentrates on the direction whose evidence is an absence. Decomposition alone gains $+11.5$ pp on omissions for Gemini and $+13.4$ pp for Llama, against $+5.8$ and $+6.9$ pp on hallucinations and $+9.2$ and $+9.1$ pp on factual errors (Table 1). Across the full ladder the ordering is unchanged, at $+23.5$ and $+27.0$ pp on omissions against $+14.4$ and $+16.4$ pp on hallucinations. The asymmetry has a structural explanation rather than a statistical one, and it is the argument of Section 1 made quantitative: a merged prompt must search in two directions at once, and the transcript-to-note direction is the one with no note-side anchor, so an omission the merged call never looks for produces

Effect	Contrast	Gemini 2.5 Pro		Llama 3.3 70B		Cost (Gemini)	
		Δ (pp)	95% CI	Δ (pp)	95% CI	+tok. (k)	pp/10k
Decomposition	SP-1 $\rightarrow$ AJ-1	$+\mathbf{8.8}$	$[7.9, 9.8]$	$+\mathbf{9.9}$	$[8.8, 11.1]$	8.4	$\mathbf{10.5}$
Iteration	AJ-1 $\rightarrow$ AJ-8	$+4.4$	$[3.7, 5.2]$	$+5.1$	$[4.3, 6.0]$	66.4	0.7
Self-improvement	AJ-8 $\rightarrow$ AJ-8+SI	$+5.9$	$[5.1, 6.8]$	$+6.5$	$[5.6, 7.5]$	5.4	$\mathbf{10.9}$
Architecture	SP-1 $\rightarrow$ AJ-8	$+13.2$	$[12.1, 14.4]$	$+15.0$	$[13.6, 16.4]$	74.8	1.8
Full pipeline	SP-1 $\rightarrow$ AJ-8+SI	$+19.1$	$[17.8, 20.5]$	$+21.5$	$[19.9, 23.1]$	80.2	2.4

Table 2: Sequential contrasts in overall match rate on the 272-case benchmark, with 95% percentile bootstrap confidence intervals over 1,000 case-level resamples. Every contrast is significant at $p < 0.001$ under the two-sided paired bootstrap test. Cost columns report additional tokens per case and the return per ten thousand additional tokens for the Gemini judge; measured per-case totals are 9.4k (SP-1), 17.8k (AJ-1), 84.2k (AJ-8), and 89.6k (AJ-8+SI) tokens.

no output at all, whereas a hallucination it skips at least leaves the offending sentence in the note for a later pass. The same asymmetry reappears in the iteration step, which recovers $+5.5$ pp on omissions but only $+3.4$ pp on hallucinations: extra passes help the harder direction somewhat, but nowhere near as much as giving it a dedicated agent does.

Decomposition also compresses the spread between categories. Under SP-1 the distance between Gemini's best and worst category is 11.3 pp; three agents narrow it to 5.6 pp at AJ-1, and the full pipeline reaches 2.2 pp. For a screen whose usefulness is set by its weakest check rather than its average one, that compression matters as much as the change in the mean. The weaker judge benefits more from every step, gaining 21.5 pp against Gemini's 19.1 pp, which suggests structure substitutes to some degree for model capability.

6 Why the Single-Pass Judge Misses

Two measurements narrow down the mechanism. Together they rule out the most benign explanation, that the merged call finds the same errors and simply labels them more noisily.

Category misattribution is real but small. Figure 2(right) reports the share of reported spans that clear the similarity threshold against a ground-truth error but are filed under the wrong category, and are therefore scored as misses. Misattribution is three to four times more common under the single pass in every condition. The confusions are not symmetric: on the 272-case benchmark, 54% of Gemini's misattributed spans are factual errors reported as hallucinations, 23% are omissions reported as factual errors, and 14% are hallucinations reported as factual errors. The dominant confusion is exactly the one the reciprocal boundary rules exist to prevent: content present in the transcript but stated wrongly looks, to a reader scanning for unsupported claims, very much like content absent from the transcript altogether. This shows the boundary rules doing work rather than merely restating the taxonomy, but misattribution accounts for well under a fifth of the single-pass shortfall, so relabelling does not explain the gap.

The remainder is under-reporting that worsens as the note gets busier. Figure 2(left) bins the 272 cases by injected-error count. The decomposed pipeline is essentially flat, losing 1.7 pp (Gemini) and 2.4 pp (Llama) from the sparsest to the densest bin; the single-pass judge falls 12.8 and 14.0 pp over the same range. A judge whose accuracy depends on how much is wrong with the note is a judge operating under a budget, and the notes it handles worst are the ones a reviewer most wants it to handle well. Three parallel calls relieve that budget in two ways at once, by giving each direction its own instruction context and its own output allocation, and the ablation as designed cannot say which matters more. A cleaner experiment would hold total output budget fixed across architectures; the two mechanisms imply different remedies, a context effect being addressed by shortening prompts and a budget effect by relaxing output limits.

The two architectures sit at different operating points. Every agentic configuration we report favours recall over precision, which is the appropriate bias for a screen running ahead of physician

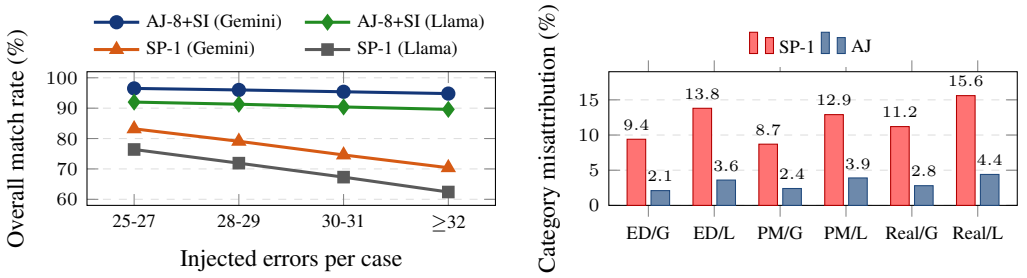

Figure 2: **Left:** match rate by injected-error density on the 272-case benchmark. The decomposed pipeline is nearly flat; the single-pass judge loses 12.8 pp (Gemini) and 14.0 pp (Llama) from the sparsest to the densest bin. **Right:** category misattribution rate, lower is better, across datasets (ED, PriMock57, Real) and judges (G = Gemini, L = Llama). The agentic bar is AJ-8+SI on synthetic data and AJ-8 on the real subset.

Dataset	Model	SP-1	AJ-1	AJ-8	AJ-8+SI	Arch. Δ
ED 272	Gemini	76.7 ± 0.5	85.5 ± 0.4	89.9 ± 0.6	$\mathbf{95.8 \pm 0.4}$	+13.2
	Llama	69.4 ± 0.7	79.3 ± 0.5	84.4 ± 0.7	$\mathbf{90.9 \pm 0.5}$	+15.0
PriMock57	Gemini	74.7 ± 2.6	83.4 ± 2.3	87.6 ± 2.1	$\mathbf{92.3 \pm 1.4}$	+12.9
	Llama	69.1 ± 2.6	78.8 ± 2.4	83.3 ± 2.1	$\mathbf{88.4 \pm 1.4}$	+14.2
Real 15	Gemini	67.0 ± 2.3	82.3 ± 1.8	$\mathbf{87.2 \pm 2.0}$	-	+20.2
	Llama	60.8 ± 2.3	76.4 ± 2.0	$\mathbf{82.1 \pm 2.2}$	-	+21.3

Table 3: Overall match rate (%, mean ± SEM) across the ladder. The final column is SP-1 → AJ-8 (decomposition plus iteration at fixed prompt content), the contrast defined identically everywhere. Self-improvement is held inactive on the real subset so that evaluation remains a generalization test.

sign-off. The single-pass judge reverses that ordering in all six model-dataset conditions: on the ED benchmark Gemini's SP-1 precision exceeds its recall by 7.1 pp (81.4 ± 0.5 vs. 74.3 ± 0.5), against 91.3 ± 0.4 precision and 94.2 ± 0.3 recall for AJ-8+SI (Section C). What the single-pass judge reports, it reports with reasonable accuracy; the problem is how much it never reports at all. That distinction is invisible in a match rate, it cannot be closed by threshold tuning because there is no threshold to move, and it is exactly what a judge silently truncating its own output looks like.

7 Generalization, Human Validation, and Correction

The architectural effect holds everywhere and widens on real data. Table 3 repeats the ladder on all three corpora. Comparing like with like on SP-1 → AJ-8 (the contrast defined identically on every dataset, since self-improvement is held inactive on the real subset). The advantage is 13.2 and 15.0 pp in-domain, 12.9 and 14.2 pp after the shift to primary care, and 20.2 and 21.3 pp on production notes. All six contrasts are significant under the paired bootstrap test ($p < 0.001$ synthetic, $p = 0.002$ real). The real-data magnitude rests on 15 cases and we would not quote it precisely, but the direction matters more for deployment than the synthetic numbers do: the gap between a single-pass judge and a decomposed one is narrowest on exactly the data where errors were injected to a known taxonomy, and widest on the notes a deployed system would actually see. Reporting this ablation only on the synthetic benchmark would have understated the case by roughly seven points.

Prompt self-improvement converges, and spends its effort on whichever category is weakest. On the synthetic benchmarks the loop gains +5.9 and +6.5 pp (ED) and +4.7 and +5.1 pp (PriMock57), with most of the movement complete by iteration 3 and every configuration flat from iteration 5 through 8 (Section G). Omissions are the largest mover in all four synthetic configurations. On real notes the picture inverts: +2.5 pp overall, but +10.4 pp of it lands on hallucinations against +1.9 and +1.4 pp elsewhere. The agent went after the weakest category in front of it. The entire real-data gain came from rewriting three prompt files. It is in-domain adaptation rather than

generalization: 15 annotated cases cannot be split into a held-out portion and still leave anything to measure.

On production notes the decomposed judge clears expert-expert agreement and the single-pass judge does not. Human-human agreement is substantial but imperfect ($\kappa = 0.69$; Table 29), so even experts do not always agree on what is reportable. The decomposed judge agrees with the union-merged expert reference at $\kappa = 0.87$ (Llama 0.82); the single-pass judge reaches only 0.67. We do not claim the judge is a more reliable annotator than a clinician: the coefficients are not chance-corrected on the same basis (Section 4), and agreeing with a reconciled reference is easier than two clinicians independently arriving at the same labels. The defensible claim is that residual judge-expert disagreement is no larger than expert-expert disagreement. Hallucination, the scarcest category, where the chance correction is smallest, is where the expert pair and the judge are closest to level.

Correction is bounded by detection. With the correction agent and its five rules held fixed, word-level F1 falls by 0.13-0.15 under the single pass in every condition (Section H): undetected omissions never reach the correction agent and therefore never return to the note. Case-level omission match rate and ROUGE-L are tightly coupled. ROUGE-L sits below ROUGE-1 and BLEU-1 by similar margins in both architectures, because a correct clinical edit still resequences the surrounding sentence; the two effects are separable, and only one responds to the architecture. Word-level F1 stays at or above 0.80 in every agentic condition.

8 Discussion

A design principle that is not specific to clinical notes. When a judge must search in two directions, the direction whose positive evidence is an *absence* is the one that needs its own agent. That direction produces no output when it is skipped, so its failures are invisible to aggregate metrics, to the judge's own self-critique, and to any downstream stage that consumes the findings. A single-pass judge is not uniformly weaker; it is weaker in a predictable place. We expect the same asymmetry wherever a model must verify a candidate against a source and simultaneously check the source for content the candidate dropped: claim verification against retrieved evidence, code review against a specification, translation adequacy.

The synthetic-to-real gap is category-specific. Moving from the ED benchmark to production notes costs Gemini 8.6 pp, which decomposes into -21.9 pp on hallucinations, -8.9 on omissions, and -6.0 on factual errors, enough to invert the category ordering. Two explanations compete: production hallucinations are intrinsically harder, or the prompts were tuned to find a kind of hallucination production notes largely do not contain. Injection produces explicit fabrications; manual review of 40 mismatched cases (Section I) found real hallucinations to be routine plan and discharge language the agent had learned to skip. The real-data self-improvement run discriminates: under the capability explanation it should recover little; under the taxonomy explanation it should produce a large hallucination-specific gain. The second is what happened ($+10.4$ pp on hallucinations). Not all of the gap is taxonomy (11.5 pp of the hallucination shortfall survives), but rewriting a prompt file is nearly free and closing a capability gap is not. Synthetic-to-real transfer for LLM judges should be reported per error category.

What a self-improvement loop can and cannot fix. No parameters change; the loop is a search over instructions guided by the agent's own outputs. It converged by iteration 5 and spent its effort on the weakest category. It can only fix failures that are both systematic and visible, so a blind spot at initialization survives, plausibly the residual 6.1 pp on real notes.

9 Conclusion

We showed that an automated reviewer assembled from general-purpose models can sit between the generated note and the physician who signs it, and that how that reviewer is decomposed is the dominant design decision. Splitting detection by comparison direction is worth roughly twice iterating the detectors and sixteen times as much per token.

ETHICS STATEMENT

This work studies automated review of clinical documentation and touches on human subjects data, dual use, and deployment risk. All emergency department transcripts and notes were accessed under institutional data use agreements and research ethics review at the host institution, and were de-identified before any model call. Expert annotation was performed by clinicians as part of that approved protocol. PriMock57 is a public dataset of mock consultations with consenting participants (Babylon Health, 2019), and we introduce no new public dataset containing patient data. Two risks deserve explicit statement. First, an automated judge that misses an error can create false assurance, and the category most likely to be missed, omission, is the one whose failures leave no visible trace in the note, so we position this pipeline as a first-pass screen *before* physician sign-off and not as a replacement for it. Second, the correction agent edits clinical text, and although it operates under an explicit no-fabrication policy, any residual fabrication in a refined note is a patient safety hazard; the pipeline should not be deployed without a human in the loop. The real-data evidence comes from a single scribe system at a single institution, all notes are in English and follow two clinical templates, and our results should not be read as evidence of safety in other languages, institutions, or documentation standards. Methodologically, the single-pass baseline uses concatenated detection prompts rather than a prompt authored from scratch for a merged call; each configuration was run once at temperature 0, so we report paired differences rather than seed-level variability; and decomposition confounds instruction context with output budget (Section 3.3). We used AI assistants for spelling, grammar, and phrasing, and for building the annotation review interface; all experimental design, implementation, analysis, and claims are the authors' own.

REPRODUCIBILITY STATEMENT

Every agent prompt used in this work is reproduced verbatim in the appendix: the three detection prompts in Section K, the correction prompt, the prompt self-improvement prompt, the single-pass baseline prompt, and the error injection prompt that generates the synthetic ground truth. Section 3.3 specifies exactly which three elements differ between the single-pass baseline and the agentic detectors. Section 4 gives the judge models and serving configuration, the decoding temperature, the 80% cosine threshold and category-scoping rule used for span matching, the κ definitions for both sides of the human comparison, the correction metric definitions, and the bootstrap procedure (1,000 case-level resamples, two-sided paired test) behind every confidence interval and p-value. Section B reports per-dataset descriptive statistics and the realized versus targeted error-injection composition. The ED benchmark and the production notes are governed by data use agreements and cannot be released; PriMock57 is public, and the full injection and judging pipeline can be reproduced on it from the released prompts. Each configuration was run once at temperature 0, and we report paired differences between configurations rather than variability across seeds, which is a limitation of the evidence rather than of the description.

REFERENCES

Elham Asgari, Nina Montaña-Brown, Magda Dubois, et al. A framework to assess clinical safety and hallucination rates of LLMs for medical text summarisation. *npj Digital Medicine*, 2025. doi: 10.1038/s41746-025-01670-7.

Babylon Health. PriMock57: A dataset of mock primary care consultations. `https://github.com/babylonhealth/primock57`, 2019.

Nicholas Brake and Torsten Schaaf. Comparing two model designs for clinical note generation; is an LLM a useful evaluator of consistency? *arXiv preprint arXiv:2404.06503*, 2024. doi: 10.48550/arxiv.2404.06503.

C. Curran, N. Neehal, and K. Murugesan. Examining trustworthiness of LLM-as-a-Judge systems in a clinical trial design benchmark. In *IEEE BigData*, 2024. doi: 10.1109/bigdata62323.2024.10825592.

Ahmed M. Darwish, Eman A. Rashed, and Ghada Khoriba. Mitigating LLM hallucinations using a multi-agent framework. *Information*, 2025. doi: 10.3390/info16070517.

A. B. Das, S. Ahmed, and S. K. Sakib. Hallucinations and key information extraction in medical texts: A comprehensive assessment of open-source large language models. *arXiv preprint arXiv:2504.19061*, 2025. doi: 10.48550/arxiv.2504.19061.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)*, 2024.

Abhimanyu Dubey et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arxiv.2407.21783.

Google. Agent development kit documentation. `https://google.github.io/adk-docs/`, 2025.

Jinlan Gu et al. A survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594*, 2024. doi: 10.48550/arxiv.2411.15594.

Lei Huang et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Computing Surveys*, 2024. doi: 10.1145/3703155.

Pradyumna K. Kanithi et al. MEDIC: Towards a comprehensive framework for evaluating LLMs in clinical applications. *arXiv preprint arXiv:2409.07314*, 2024. doi: 10.48550/arxiv.2409.07314.

Md Tahmid Rahman Laskar et al. Improving automatic evaluation of large language models (LLMs) in biomedical relation extraction via LLMs-as-the-Judge. *arXiv preprint arXiv:2506.00777*, 2025. doi: 10.48550/arxiv.2506.00777.

Haopeng Li et al. LLMs-as-Judges: A comprehensive survey on LLM-based evaluation methods. *arXiv preprint arXiv:2412.05579*, 2024. doi: 10.48550/arxiv.2412.05579.

Junyi Li et al. HaluEval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*, 2023. doi: 10.48550/arxiv.2305.11747.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.

Jiqun Liu and Jiangen He. The decoy dilemma in online medical information evaluation: A comparative study of credibility assessments by LLM and human judges. *arXiv preprint arXiv:2411.15396*, 2024. doi: 10.48550/arxiv.2411.15396.

T.-L. Liu et al. Mixed methods evaluation of ambient artificial intelligence scribe adoption in a large health system. *JAMA Network Open*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Ankit Pal, L. K. Umapathi, and M. Sankarasubbu. Med-HALT: Medical domain hallucination test for large language models. *arXiv preprint arXiv:2307.15343*, 2023. doi: 10.48550/arxiv.2307.15343.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002. doi: 10.3115/1073083.1073135.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Kevin Schroeder and Zachary Wood-Doughty. Can you trust LLM judgments? reliability of LLM-as-a-Judge. *arXiv preprint arXiv:2412.12509*, 2024. doi: 10.48550/arxiv.2412.12509.

Eric Schumacher et al. MED-OMIT: Extrinsically-focused evaluation metric for omissions in medical summarization. In *Machine Learning for Health (ML4H) at NeurIPS*, 2023.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2020.acl-main.704.

Tait D. Shanafelt, Lotte N. Dyrbye, Colin P. West, et al. Relationship between clerical burden and characteristics of the electronic environment with physician burnout and professional satisfaction. *Mayo Clinic Proceedings*, 2016. doi: 10.1016/j.mayocp.2016.05.007.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Christine Sinsky, Lacey Colligan, Ling Li, et al. Allocation of physician time in ambulatory practice: A time and motion study in 4 specialties. *Annals of Internal Medicine*, 2016. doi: 10.7326/M16-0961.

B. N. Suhas et al. Fact-controlled diagnosis of hallucinations in medical text summarization. *arXiv preprint arXiv:2506.00448*, 2025. doi: 10.48550/arxiv.2506.00448.

Agata Szymanski et al. Limitations of the LLM-as-a-Judge approach for evaluating LLM outputs in expert knowledge tasks. *arXiv preprint arXiv:2410.20266*, 2024. doi: 10.48550/arxiv.2410.20266.

Alyssa A. Tierney et al. Ambient artificial intelligence scribes to alleviate the burden of clinical documentation. *NEJM Catalyst Innovations in Care Delivery*, 2024. doi: 10.1056/CAT.23.0404.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Christopher Y. K. Williams, Jaskaran Bains, Tianyu Tang, et al. Evaluating large language models for drafting emergency department encounter summaries. *PLOS Digital Health*, 2025. doi: 10.1371/journal.pdig.0000899.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Jiawei Ye et al. Justice or prejudice? quantifying biases in LLM-as-a-Judge. *arXiv preprint arXiv:2410.02736*, 2024. doi: 10.48550/arxiv.2410.02736.

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.

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 (NeurIPS), Datasets and Benchmarks Track*, 2023.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *International Conference on Learning Representations (ICLR)*, 2023.

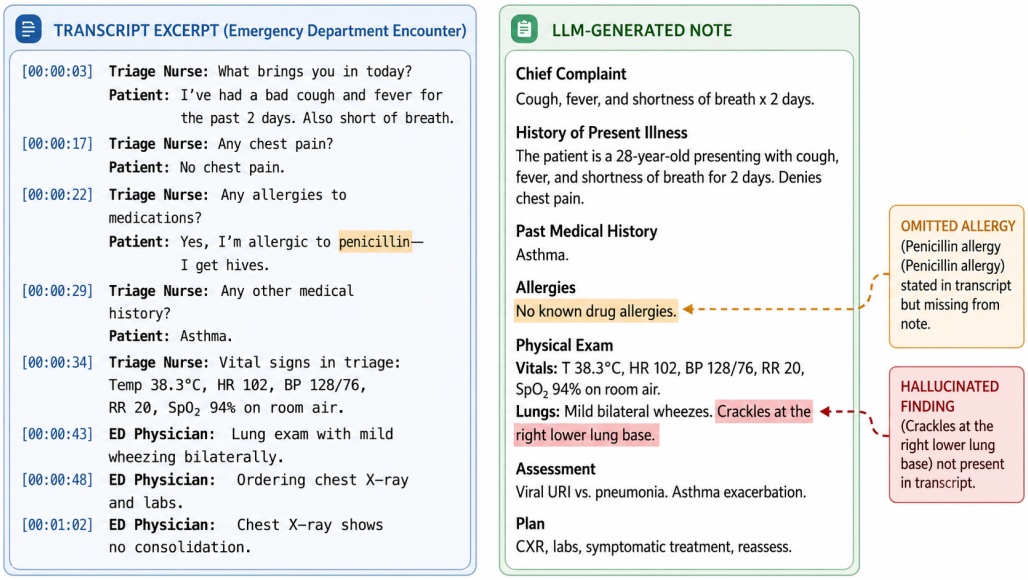

Figure 3: A synthetic emergency department transcript excerpt and the note generated from it. The note contains a fabricated lung finding and omits a stated allergy. Neither is visible without checking the transcript, and the omission leaves no trace anywhere in the note, which is the asymmetry the decomposition in Section 3 is built around.

A A Motivating Case

B Dataset Details

Table 4 reports descriptive statistics for the three evaluation corpora. Word counts are whitespace tokens on the plain-text transcript and on the note the judge receives, that is, the error-injected note for the two synthetic benchmarks and the unmodified production note for the real subset.

	ED benchmark	**PriMock57**	**Real subset**
Cases	272	57	15
Domain	Emergency department	Primary care	Emergency department
Note source	Research scribe	Research scribe	Production scribe (GPT-4.5)
Ground truth	Injected errors	Injected errors	Expert annotation
Transcript words	1,383 [1,179-1,625]	1,657 [1,321-1,929]	1,116 [1,007-1,131]
range	842-2,445	639-2,964	791-2,584
Corrected note, words	334 [291-373]	194 [178-235]	278 [260-294]
Hallucination labels	2,588	535	48
Omission labels	2,571	575	106
Factual-error labels	1,369	276	282
Total labels	**6,528**	**1,386**	**436**

Table 4: Descriptive statistics of the three evaluation corpora. Lengths are whitespace tokens, reported as median [IQR] with the full range underneath.

Two patterns are worth noting. PriMock57 transcripts are longer than the emergency department transcripts (median 1,657 versus 1,383 words) but carry roughly one fifth as many labels in absolute terms simply because the benchmark is smaller; per case, injection density is comparable. On the emergency department benchmark the corrected note is substantially longer than the note the judge received, which is the expected signature of a correction agent whose dominant action is restoring omitted transcript content rather than deleting fabricated content.

Table 5 gives the specialty mix of the 272-case emergency department benchmark. Respiratory and general emergency cases form the majority; musculoskeletal cases are the next largest group.

Specialty prefix	**Cases**
RES (respiratory and general emergency)	213
MSK (musculoskeletal)	46
CAR, GAS, DER, and other (cardiac, gastrointestinal, dermatology, general)	13

Table 5: Case counts by clinical specialty in the 272-case emergency department benchmark.

Table 6 lists the model used in each pipeline role. Gemini 2.5 Pro is the closed-weight judge and the error injector; Llama 3.3 70B is the open-weight judge, run with byte-identical prompts; production notes come from the Alberta Health Services Jenkins Scribe (GPT-4.5).

Role	**Model**
Error injection	Gemini 2.5 Pro
Agentic judge and correction (closed-weight)	Gemini 2.5 Pro
Agentic judge and correction (open-weight)	Llama 3.3 70B
Production notes (Jenkins Scribe)	GPT-4.5

Table 6: Models and their roles in the framework.

B.1 Realized Error Injection Composition

The injection agent is instructed to target 12-15 omissions, 8-10 hallucinations, and 4-6 modification errors per case. Table 7 compares those targets against what it produced across the 272 cases. The realized medians sit inside every target band and the interquartile ranges are narrow. A small number of cases overshoot the hallucination and modification bands; we did not filter them out, since removing them would bias the benchmark toward the notes the injector found easiest to corrupt.

Category	**Target**	**Median [IQR]**	**Range**
Omission	12-15	13 [13-14]	12-15
Hallucination	8-10	10 [9-10]	8-23
Modification	4-6	5 [5-6]	4-15
Total	24-31	29 [28-30]	25-47

Table 7: Target versus realized injected-error composition per case on the 272-case benchmark.

Not every injected record is scorable. Matching requires a non-empty `exact_content` field to align against, and a small fraction of records describe an error without quoting a span, most often when the injector removed an entire conversation block and had no note-side text to point at. Those records are excluded from the label totals, which is why the scored total (6,528) is lower than the raw injected total.

C Pooled Detection Metrics

Table 8 reports precision, recall, F1, and accuracy for the single-pass baseline against the best agentic configuration on every dataset and both judges. Recall is the expert-label or injected-label match rate; precision scores judge detections against the reference label set; F1 is their harmonic mean; accuracy is the overall match rate of Section 5. The ordering recall $\geq$ accuracy $\geq$ F1 $>$ precision holds in all six agentic conditions and inverts in all six single-pass conditions, which is the operating-point result discussed in Section 6.

Table 9 reports the same four metrics for the assembled pipeline only (AJ-8+SI on synthetic data, AJ-8 on the real subset), matching the thesis comparison of the two judge models at a fixed architecture.

Figure 4 plots the four metrics by dataset for each judge. Figure 5 shows the same six assembled-pipeline conditions as grouped bars.

Dataset	Model	Config.	Precision ↑	Recall ↑	F1 ↑	Accuracy ↑
ED 272	Gemini	SP-1	81.4 ± 0.5	74.3 ± 0.5	77.7 ± 0.4	76.7 ± 0.5
		AJ-8+SI	$\mathbf{91.3 \pm 0.4}$	$\mathbf{94.2 \pm 0.3}$	$\mathbf{92.4 \pm 0.3}$	$\mathbf{95.8 \pm 0.4}$
	Llama	SP-1	76.8 ± 0.6	67.1 ± 0.7	71.6 ± 0.6	69.4 ± 0.7
		AJ-8+SI	88.0 ± 0.5	89.3 ± 0.4	88.3 ± 0.3	90.9 ± 0.5
PriMock57	Gemini	SP-1	80.9 ± 1.3	72.9 ± 1.4	76.7 ± 1.6	74.7 ± 2.6
		AJ-8+SI	$\mathbf{90.5 \pm 1.1}$	$\mathbf{93.3 \pm 1.0}$	$\mathbf{91.8 \pm 1.5}$	$\mathbf{92.3 \pm 1.4}$
	Llama	SP-1	76.2 ± 1.5	67.6 ± 1.6	71.6 ± 1.8	69.1 ± 2.6
		AJ-8+SI	86.1 ± 1.2	89.4 ± 2.3	87.7 ± 1.7	88.4 ± 1.4
Real 15	Gemini	SP-1	78.2 ± 1.3	69.1 ± 1.5	73.4 ± 1.4	67.0 ± 2.3
		AJ-8	$\mathbf{84.8 \pm 1.1}$	$\mathbf{88.8 \pm 1.3}$	$\mathbf{86.4 \pm 1.1}$	$\mathbf{87.2 \pm 2.0}$
	Llama	SP-1	74.0 ± 2.1	63.4 ± 2.3	68.3 ± 1.9	60.8 ± 2.3
		AJ-8	80.1 ± 1.9	83.7 ± 2.0	81.5 ± 1.5	82.1 ± 2.2

Table 8: Pooled precision, recall, F1, and accuracy (%, mean ± SEM) for the single-pass baseline and the best agentic configuration across all three corpora. Best per dataset in bold.

Dataset	Model	Precision ↑	Recall ↑	F1 ↑	Accuracy ↑
ED 272	Gemini	91.3 ± 0.4	94.2 ± 0.3	92.4 ± 0.3	95.8 ± 0.4
	Llama	88.0 ± 0.5	89.3 ± 0.4	88.3 ± 0.3	90.9 ± 0.5
PriMock57	Gemini	90.5 ± 1.1	93.3 ± 1.0	91.8 ± 1.5	92.3 ± 1.4
	Llama	86.1 ± 1.2	89.4 ± 2.3	87.7 ± 1.7	88.4 ± 1.4
Real 15	Gemini	84.8 ± 1.1	88.8 ± 1.3	86.4 ± 1.1	87.2 ± 2.0
	Llama	80.1 ± 1.9	83.7 ± 2.0	81.5 ± 1.5	82.1 ± 2.2

Table 9: Pooled precision, recall, F1, and accuracy (%, mean ± SEM) for both judge models under the assembled pipeline (AJ-8+SI on synthetic benchmarks, AJ-8 on the real subset).

D Per-Category Detection Results

The main text reports overall match rates. Tables 10 to 12 give the per-category breakdown at the assembled-pipeline endpoint on each corpus. Table 13 consolidates those rates and the change relative to the in-domain emergency department benchmark.

Category	Labels	Gemini 2.5 Pro	Llama 3.3 70B
Hallucination	2,588	96.9 ± 0.2	93.0 ± 0.3
Omission	2,571	94.7 ± 0.3	89.0 ± 0.5
Factual error	1,369	95.7 ± 0.1	90.7 ± 0.2
Overall	**6,528**	$\mathbf{95.8 \pm 0.3}$	$\mathbf{90.9 \pm 0.4}$

Table 10: Detection match rate (%, mean ± SEM) by error category on the 272-case emergency department benchmark at AJ-8+SI.

Figure 6 shows the overall rates. Figure 7 breaks the same comparison down by error category.

E Ablation Ladder in Full

The main-text ablation is reproduced here so that every Chapter 4 result table appears in one place. Table 14 is the per-category ladder on the 272-case benchmark. Table 15 is the sequential contrast table. Table 16 repeats the overall ladder on all three corpora.

Figure 8 plots the 272-case ladder. Figure 9 splits it by error category. Figure 10 compares the single-pass baseline with the best agentic rung on every corpus.

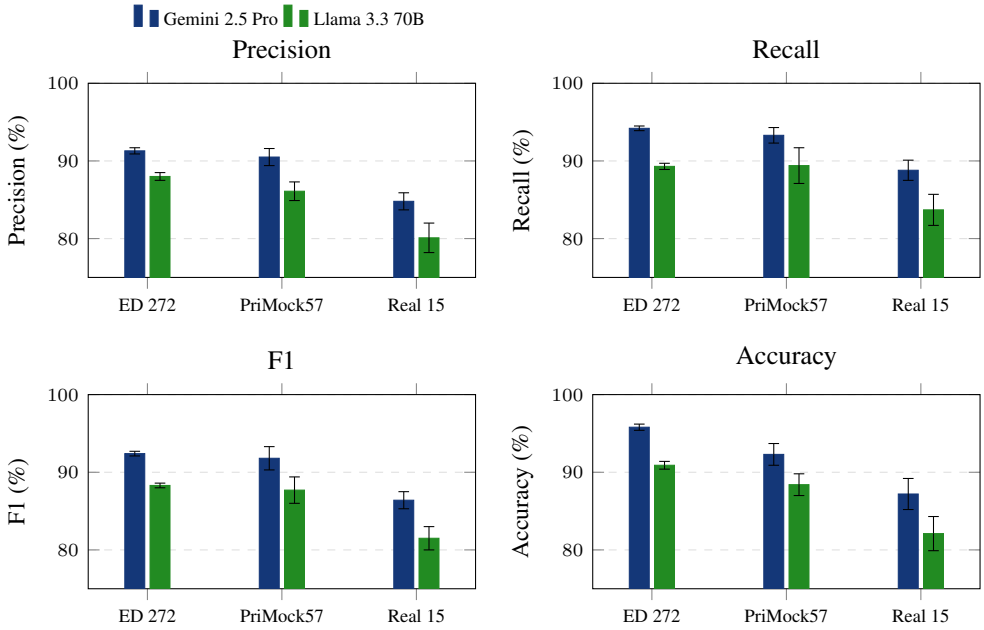

Figure 4: Precision, recall, F1, and accuracy (mean ± SEM) by dataset for both judges under the assembled pipeline.

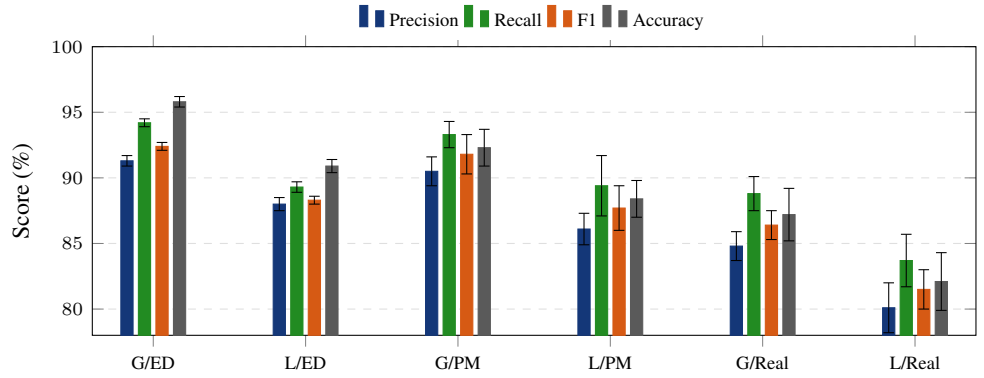

Figure 5: Precision, recall, F1, and accuracy (mean ± SEM) across all six assembled-pipeline conditions. G = Gemini, L = Llama, ED = ED 272, PM = PriMock57, Real = 15-case production subset.

F Ablation Diagnostics

Table 17 reports the category-misattribution rates plotted in Figure 2(right). Table 18 reports the density-bin rates plotted in Figure 2(left). Table 19 reports the measured per-case cost of each ablation rung for the Gemini judge.

Figure 11 restates the density result as a standalone plot. Figure 12 plots accuracy against token cost: the dashed line is the efficient frontier through SP-1, AJ-1, and AJ-8+SI. AJ-8 is the only rung below it.

Category	Labels	Gemini 2.5 Pro	Llama 3.3 70B
Hallucination	535	94.0 ± 0.9	90.7 ± 1.0
Omission	575	90.8 ± 1.4	86.4 ± 1.5
Factual error	276	92.0 ± 0.6	88.2 ± 0.6
Overall	**1,386**	$\mathbf{92.3 \pm 1.3}$	$\mathbf{88.4 \pm 1.3}$

Table 11: Detection match rate (%, mean ± SEM) by error category on the 57-case PriMock57 benchmark at AJ-8+SI.

Category	Labels	Gemini 2.5 Pro	Llama 3.3 70B
Hallucination	48	75.0 ± 6.2	66.7 ± 5.8
Omission	106	85.8 ± 3.3	83.0 ± 3.6
Factual error	282	89.7 ± 2.2	84.4 ± 2.1
Overall	**436**	$\mathbf{87.2 \pm 2.0}$	$\mathbf{82.1 \pm 2.2}$

Table 12: Detection match rate (%, mean ± SEM) against expert labels on the 15-case production subset at AJ-8 (self-improvement held inactive).

G Prompt Self-Improvement Trajectories

G.1 Synthetic Benchmarks

Table 20 reports per-category match rates for every synthetic configuration at iterations 1-5; iterations 6-8 are identical to iteration 5 in all four configurations and are collapsed. Figure 13 plots the overall trajectories across all eight iterations. All four configurations rise through iteration 5 and are exactly flat thereafter, which is what we read as convergence: the agent returns unchanged prompts because it can no longer point to a failure cause recurring in at least two cases.

Figure 14 plots the same synthetic runs one panel per configuration and one line per error category.

G.2 Real Production Notes

Table 21 reports the real-data run, with the self-improvement agent switched back on for eight iterations over the 15 production cases. Iteration 1 runs the frozen synthetic prompts and therefore reproduces the frozen-prompt result exactly. The total gain is $+2.5$ pp, less than half the $+5.9$ pp the same agent extracted on the 272-case benchmark, which is expected since the loop starts from converged prompts rather than a first draft. In absolute terms it recovered 11 of the 56 labels the frozen configuration had missed, taking the judge from 380 to 391 matched labels out of 436. Only the hallucination gain clears its own standard error, so it is the one category where we would describe the improvement as established rather than suggestive.

Table 22 places the converged real-data configuration next to both frozen-prompt judges. Against frozen Gemini the gain is $+2.5$ pp overall and $+10.4$ pp on hallucinations; only the hallucination gain clears its own standard error. The comparison against frozen Llama is larger but conflates the change of judge with the activation of the loop.

Table 23 reports the pooled detection metrics for the self-improved Gemini run, with the frozen real-data row and the two synthetic Gemini rows for context. Precision gains more than recall ($+3.2$ vs. $+2.5$ pp), because two of the four dominant failure modes in Section I are false positives and the revisions addressing them are effectively suppression rules. The synthetic-to-real accuracy gap narrows from 8.6 pp to 6.1 pp, so roughly 29% of the residual transfer gap is recoverable by letting the same loop run on the target distribution and the rest is not. This experiment has no held-out real-data split (15 expert-annotated cases cannot be divided and still leave anything to measure), so 89.7% describes what the pipeline reaches when allowed to adapt to a distribution, not what it would score on the next 15 production notes.

Figure 15 places the real-data overall trajectory next to the two synthetic Gemini curves. Figure 16 shows where the real-data gain came from. Figure 17 puts all three Gemini category trajectories on

Category	Model	Match rate			Change vs. ED 272	
		ED 272	PriMock57	Real 15	→ PM57	→ Real
Hallucination	Gemini	96.9 ± 0.2	94.0 ± 0.9	75.0 ± 6.2	-2.9	-21.9
	Llama	93.0 ± 0.3	90.7 ± 1.0	66.7 ± 5.8	-2.3	-26.3
Omission	Gemini	94.7 ± 0.3	90.8 ± 1.4	85.8 ± 3.3	-3.9	-8.9
	Llama	89.0 ± 0.5	86.4 ± 1.5	83.0 ± 3.6	-2.6	-6.0
Factual error	Gemini	95.7 ± 0.1	92.0 ± 0.6	89.7 ± 2.2	-3.7	-6.0
	Llama	90.7 ± 0.2	88.0 ± 0.6	84.4 ± 2.1	-2.7	-6.3
Overall	Gemini	95.8 ± 0.3	92.3 ± 1.3	87.2 ± 1.9	-3.5	-8.6
	Llama	90.9 ± 0.4	88.4 ± 1.3	82.1 ± 2.2	-2.5	-8.8

Table 13: Detection match rate (%, mean ± SEM) by error category and model across the three corpora, with the change in percentage points relative to the 272-case emergency department benchmark. Synthetic rows are AJ-8+SI; real rows are AJ-8.

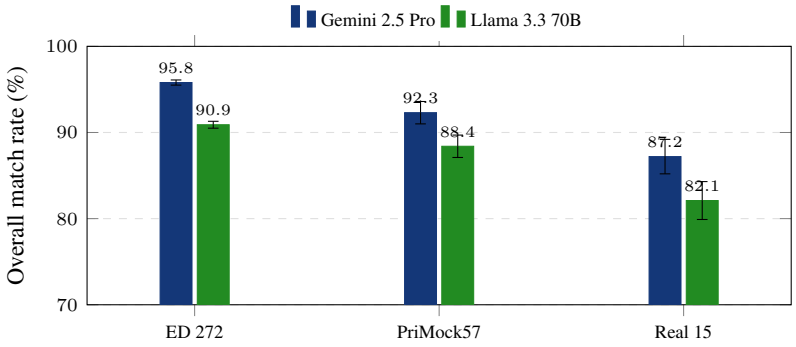

Figure 6: Overall detection match rate (mean ± SEM) by dataset for both judge models at the assembled-pipeline endpoint.

shared axes. Figure 18 compares frozen and self-improved Gemini on real notes by category and by pooled metric.

H Correction Results in Full

Table 24 reports word-level and n-gram correction quality for both judges across all three corpora under the assembled pipeline. Word recovery is uniformly high: line-based F1 stays at or above 0.78 in every condition and above 0.83 in every Gemini condition. Sequence-level overlap is consistently lower, with ROUGE-L roughly 0.04 below ROUGE-1 and 0.09 below BLEU-1 in every condition, which is the signature of the minimal-edit and structure-lock rules rather than of weak correction. Cross-setting degradation is modest and model-independent, at roughly 0.03 F1 moving to PriMock57 and 0.06 moving to production notes, and the Llama-to-Gemini gap of 0.04-0.05 F1 is stable rather than widening on harder data, which makes on-premises deployment viable at a well-characterized cost.

Figure 19 shows the four headline correction metrics across the six model-dataset conditions.

Table 25 holds the correction agent and its five editing rules fixed and varies only the detection stage upstream of it. Word-level F1 falls by 0.13 to 0.15 under the single pass in every condition: undetected omissions never reach the correction agent and therefore never return to the note. The structural signature of the correction agent survives the change. ROUGE-L sits below ROUGE-1 and BLEU-1 in both architectures by similar margins, which attributes that shortfall to the minimal-edit and structure-lock rules rather than to detection quality.

Table 26 reports the same seven metrics for the Gemini judge with the self-improvement agent switched on over the 15 production cases. Every metric moves in one direction, but no single gain

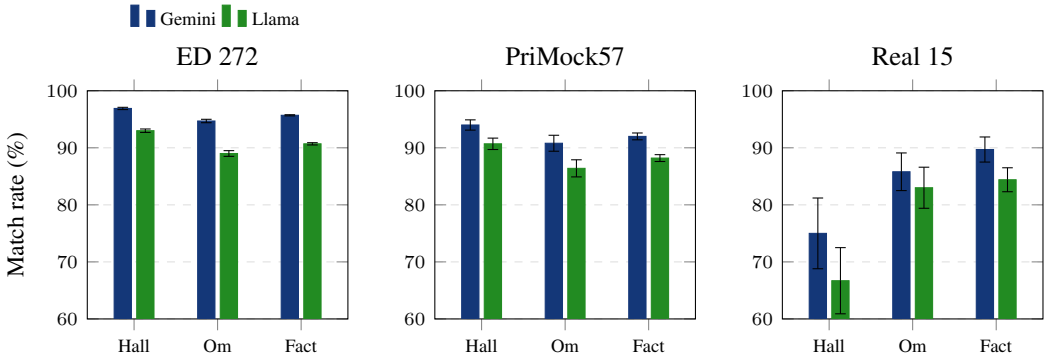

Figure 7: Per-category detection match rates (mean ± SEM) for both judge models on each corpus. Hall = hallucination, Om = omission, Fact = factual error.

Config.	Model	Halluc.	Omission	Factual	Overall	Δ vs. SP-1
SP-1	Gemini	82.5 ± 0.6	71.2 ± 0.8	76.2 ± 0.4	76.7 ± 0.5	-
	Llama	76.6 ± 0.7	62.0 ± 1.0	69.7 ± 0.5	69.4 ± 0.7	-
AJ-1	Gemini	88.3 ± 0.5	82.7 ± 0.7	85.4 ± 0.3	85.5 ± 0.4	$+8.8$
	Llama	83.5 ± 0.6	75.4 ± 0.8	78.8 ± 0.4	79.3 ± 0.5	$+9.9$
AJ-8	Gemini	91.7 ± 0.4	88.2 ± 0.6	89.7 ± 0.3	89.9 ± 0.6	$+13.2$
	Llama	87.6 ± 0.5	81.6 ± 0.7	84.1 ± 0.3	84.4 ± 0.7	$+15.0$
AJ-8+SI	Gemini	96.9 ± 0.3	94.7 ± 0.4	95.7 ± 0.2	95.8 ± 0.4	$+19.1$
	Llama	93.0 ± 0.4	89.0 ± 0.5	90.7 ± 0.2	90.9 ± 0.5	$+21.5$

Table 14: Detection match rate (%, mean ± SEM) for the four ablation rungs on the 272-case benchmark. AJ-8 reproduces the iteration-1 row of Table 20; AJ-8+SI reproduces the iteration 5-8 row.

approaches its own standard error on a 15-case sample, so the table supports that correction did not degrade rather than that any individual metric improved by a measurable amount. The correction agent itself was never rewritten; any movement is downstream of 11 newly detected labels across 15 notes.

Figure 20 shows the same seven metrics side by side for the frozen and self-improved Gemini runs on the real subset. Every metric moves in one direction, but no single gain approaches its own standard error.

Figure 21 shows the coupling that bounds the correction track. Because the correction agent only edits spans the detection agents surface, correction quality is bounded by detection completeness, and the binding constraint is omission detection: content that is never flagged is never restored. Closing the remaining correction gap therefore requires investment on the detection side rather than further tuning of the correction prompt.

I Qualitative Error Analysis

We manually reviewed 40 mismatched cases, 20 false positives and 20 false negatives, split evenly between the synthetic and real-data experiments. Most failures came down to how things were said rather than how hard they were clinically. Hallucination false positives were mostly paraphrased pertinent negatives: the patient said something in plain language, the note wrote it up in clinical language, and the agent did not connect the two. Hallucination false negatives appeared somewhere else entirely, in plan and discharge sections, where invented text resembles standard template wording. Omissions split the same way: false positives usually occurred when content was in the note but filed under a heading the agent did not check, and false negatives were mostly facts mentioned

		Gemini 2.5 Pro		Llama 3.3 70B		Cost (Gemini)	
Effect	**Contrast**	Δ (pp)	95% CI	Δ (pp)	95% CI	+tok. (k)	pp/10k
Decomposition	SP-1 $\rightarrow$ AJ-1	$+8.8$	$[7.9, 9.8]$	$+9.9$	$[8.8, 11.1]$	8.4	10.5
Iteration	AJ-1 $\rightarrow$ AJ-8	$+4.4$	$[3.7, 5.2]$	$+5.1$	$[4.3, 6.0]$	66.4	0.7
Self-improvement	AJ-8 $\rightarrow$ AJ-8+SI	$+5.9$	$[5.1, 6.8]$	$+6.5$	$[5.6, 7.5]$	5.4	10.9
Architecture	SP-1 $\rightarrow$ AJ-8	$+13.2$	$[12.1, 14.4]$	$+15.0$	$[13.6, 16.4]$	74.8	1.8
Full pipeline	SP-1 $\rightarrow$ AJ-8+SI	$+19.1$	$[17.8, 20.5]$	$+21.5$	$[19.9, 23.1]$	80.2	2.4

Table 15: Sequential contrasts in overall match rate on the 272-case benchmark, with 95% percentile bootstrap confidence intervals over 1,000 case-level resamples. Every contrast is significant at $p < 0.001$ under the two-sided paired bootstrap test. Cost columns report additional tokens per case and the return per ten thousand additional tokens for the Gemini judge.

Dataset	**Model**	**SP-1**	**AJ-1**	**AJ-8**	**AJ-8+SI**	**Arch. Δ**
ED 272	Gemini	76.7 ± 0.5	85.5 ± 0.4	89.9 ± 0.6	95.8 ± 0.4	$+13.2$
	Llama	69.4 ± 0.7	79.3 ± 0.5	84.4 ± 0.7	90.9 ± 0.5	$+15.0$
PriMock57	Gemini	74.7 ± 2.6	83.4 ± 2.3	87.6 ± 2.1	92.3 ± 1.4	$+12.9$
	Llama	69.1 ± 2.6	78.8 ± 2.4	83.3 ± 2.1	88.4 ± 1.4	$+14.2$
Real 15	Gemini	67.0 ± 2.3	82.3 ± 1.8	87.2 ± 2.0	-	$+20.2$
	Llama	60.8 ± 2.3	76.4 ± 2.0	82.1 ± 2.2	-	$+21.3$

Table 16: Overall match rate (%, mean $\pm$ SEM) across the ablation ladder on all three corpora. The final column is SP-1 $\rightarrow$ AJ-8. All six of those contrasts are significant under the paired bootstrap test ($p < 0.001$ synthetic, $p = 0.002$ real).

once and never repeated. Table 27 gives one typical example of each and the prompt revision the self-improvement agent made in response.

J Worked Example

Figure 22 shows one complete case end to end, rendered in the review interface used for annotation and error analysis. The case is a respiratory encounter from the 272-case benchmark, chosen because its transcript is the shortest in the set (842 words) and therefore fits on a page while still exercising all three error categories. The injector introduced 30 errors into this note: 11 hallucinations, 14 omissions, and 5 modification errors, of which 13 carry a quotable span and are therefore scored.

Three properties of the pipeline are visible in this single case. The hallucinated allergy line, "No known allergies," directly contradicts a transcript statement and is the most dangerous of the injected errors; it is caught by the note-to-transcript agent, not by the omission agent, because the note does contain an allergies section. The dropped Social History and Family History sections are the mirror image: nothing in the note points at them, so only a transcript-to-note pass can surface them. And the boilerplate plan lines are the failure mode the hallucination prompt calls out explicitly, because they read as routine clinical language and are easy to wave through. The correction agent's behaviour illustrates the structure-lock and no-fabrication rules: it restored the omitted sections under their template headings rather than folding them into the history of presenting illness, corrected a misspelled medication in place without touching the surrounding line, and replaced the fabricated impression with the clinician's spoken assessment rather than deleting the impression outright.

K Agent Prompts

This section reproduces the full instruction text of every agent, exactly as used. The three detection prompts appear in comparison-direction order. Each is rendered with its structure preserved: section headings, label definitions, checklists, and the JSON output schema the agent must emit. The tokens `[TRANSCRIPT]` and `[NOTE]`, shown as highlighted chips in the figures, are the two template

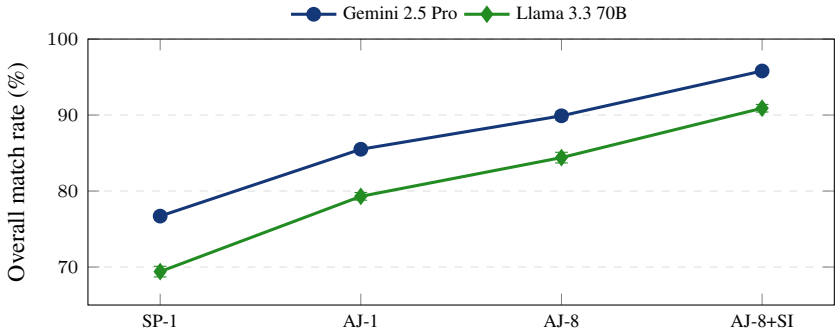

Figure 8: Overall detection match rate (mean ± SEM) on the 272-case benchmark across the ablation ladder. Each step adds one design commitment: decomposition, then iteration, then prompt self-improvement.

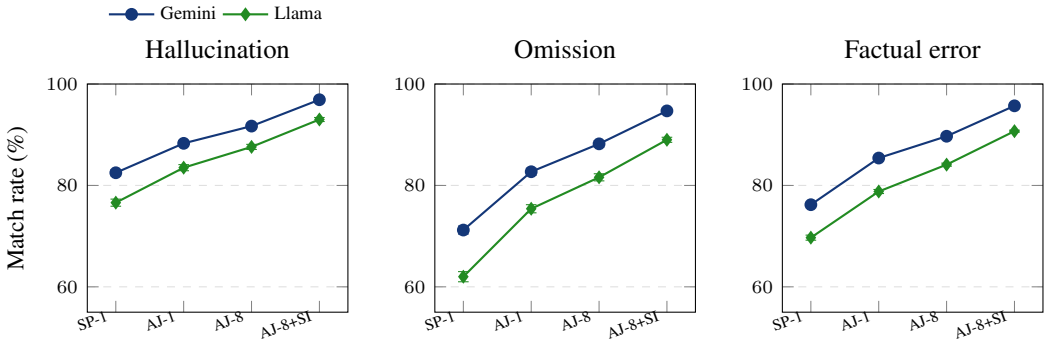

Figure 9: Detection match rate (mean ± SEM) by error category across the ablation ladder on the 272-case benchmark. Omission is the category most damaged by collapsing the three agents into one call and the category that gains most from decomposition.

placeholders filled in at run time. The correction agent additionally receives the detection findings from the current iteration.

Because hallucination detection and factual-error detection are separate agents, each prompt carries an explicit boundary rule naming the other. The hallucination prompt reports a span only when the underlying information is absent from the transcript entirely; the moment the fact is present but stated wrongly, the span belongs to the factual-error agent. The factual-error prompt states the converse and additionally requires every finding to name the transcript value it disagrees with, which makes a boundary violation visible rather than silent. Keeping the two scopes disjoint matters for scoring as well as detection, since matching is category-scoped and a span reported by the wrong agent is a miss. The single-pass baseline of Section 3.3 is the concatenation of Sections K.1 to K.3 with these boundary rules replaced by a single `category` field instruction and the three schemas merged into one object.

K.1 Agent 1: Hallucination Detection ($N \rightarrow T$)

This agent performs the note-to-transcript pass. It reads the note claim by claim and asks, for each, whether the transcript supports it anywhere, in any wording. It assigns two labels: hallucination for content absent from the transcript entirely, and irrelevant for non-clinical conversation that does not belong in a note. The irrelevant label sits with this agent rather than with the factual-error agent because it is, like hallucination, a judgment about whether note-side content should be there at all.

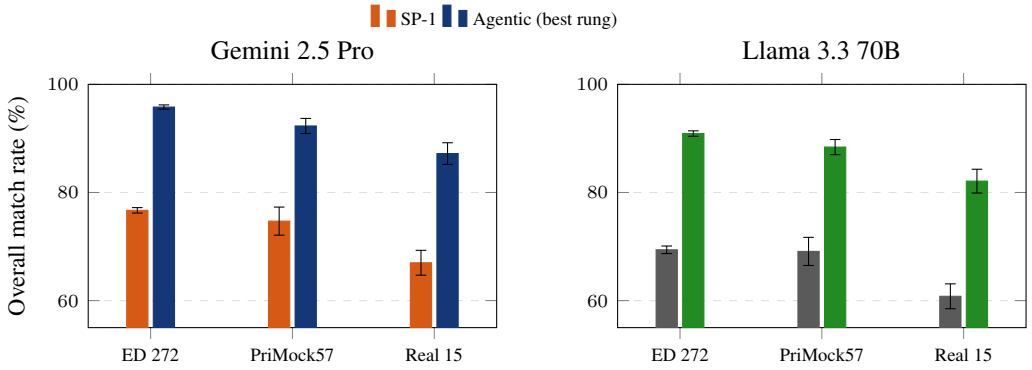

Figure 10: Single-pass baseline against the best agentic configuration available on each corpus (AJ-8+SI on the synthetic benchmarks, AJ-8 on the real subset).

Dataset	Model	SP-1	Agentic	Reduction
ED 272	Gemini	9.4%	2.1%	4.5×
	Llama	13.8%	3.6%	3.8×
PriMock57	Gemini	8.7%	2.4%	3.6×
	Llama	12.9%	3.9%	3.3×
Real 15	Gemini	11.2%	2.8%	4.0×
	Llama	15.6%	4.4%	3.5×

Table 17: Category misattribution rate: the share of reported spans whose text matches a ground-truth error but whose category label does not. The agentic column is AJ-8+SI on the synthetic benchmarks and AJ-8 on the real subset.

K.2 Agent 2: Omission Detection ($T \rightarrow N$)

This agent performs the transcript-to-note pass. It extracts clinically relevant facts from the transcript first, then searches the whole note for each one, and only then applies a clinical-significance filter. Running extraction before search is deliberate: an agent that reads the note first anchors on the note's structure and stops looking for categories the note does not mention.

K.3 Agent 3: Factual-Error Detection ($T \leftrightarrow N$)

This agent performs the paired comparison. Unlike the other two it never asks whether content should be present; it takes content appearing on both sides and asks only whether the note states it correctly. The prompt opens by instructing the agent to build a list of every stated clinical value in the transcript before it reads the note at all, so the comparison is value-driven rather than sentence-driven.

K.4 Note Correction

The correction agent consumes the merged output of all three detection agents and returns a refined note. Its rules encode the tension the pipeline has to manage: it may not fabricate, but restoring an omission is not fabrication, because the restored content is verified source-of-truth material.

K.5 Prompt Self-Improvement

The self-improvement agent runs once per loop iteration, after the detection and correction agents have processed a sample of cases. It does not review individual notes. It inspects the detections themselves, decides independently for each detection prompt whether a systematic weakness is present, and rewrites only prompts for which it can point to a failure cause recurring in at least two cases.

Injected errors per case	AJ Gemini	AJ Llama	SP Gemini	SP Llama
25-27	96.5	92.0	83.2	76.4
28-29	96.0	91.3	79.1	71.9
30-31	95.4	90.4	74.6	67.3
≥ 32	94.8	89.6	70.4	62.4
Δ (densest $-$ sparsest)	-1.7	-2.4	-12.8	-14.0

Table 18: Overall detection match rate (%) by injected-error density on the 272-case benchmark. Agentic columns are AJ-8+SI; single-pass columns are SP-1.

Config.	Detection calls	Total calls	Tokens (k)	Latency	Match rate
SP-1	1	2.0	9.4	38 s	76.7%
AJ-1	3	4.0	17.8	52 s	85.5%
AJ-8	≤ 24	19.4	84.2	3.9 min	89.9%
AJ-8+SI	≤ 24	20.6	89.6	4.2 min	95.8%

Table 19: Measured cost per case for the Gemini judge on the 272-case benchmark. Total calls, tokens, and latency are case-level means; calls fall below the ceiling because the loop exits early when a pass returns no new finding. Detection agents are dispatched concurrently.

Three design choices do most of the work. The two-case recurrence bar keeps the loop from chasing single-case noise, which in early development produced prompt churn with no match-rate benefit. The per-prompt failure taxonomy gives the agent concrete patterns to look for rather than an open-ended instruction to improve the prompt, and it includes the boundary-leakage pattern that separating hallucination from factual-error detection makes possible: a span reported by the wrong agent is attributed to whichever prompt should have owned it, so the two scopes do not drift into overlap across iterations. The validity check is what makes an automated rewrite safe to deploy: any revision that would drop a template placeholder or alter a field name in the output schema is discarded and the previous version kept, so a bad rewrite costs one iteration rather than breaking the pipeline.

L Error Injection Prompt

The synthetic benchmarks are built by a separate agent that corrupts an existing note under a fixed error budget while never adding true information from the transcript. Because the resulting labels are the ground truth against which every detection number is scored, we release this prompt in full alongside the judge prompts. Its omission section deliberately targets non-question-and-answer negations and whole conversation blocks rather than isolated facts, because pilot inspection showed that dropping a single word produces an omission no reader would call clinically significant. Its hallucination section includes a fabricated non-Canadian medication in every case, which tests whether a downstream system flags an unfamiliar drug name rather than silently accepting it. Its modification section prefers near-synonym substitutions such as *constant* for *persistent* over numeric swaps, which are easier to catch, and caps numeric changes at two per case.

M Note Template

Every note in this study, generated, corrupted, or corrected, follows the institutional full-visit template reproduced below. The correction agent's structure-lock rule is defined relative to this template: it may not create, merge, split, or reorder these sections. Sections marked omittable are dropped when the encounter contains no corresponding information, which is why a missing Social History heading is not by itself evidence of an omission and has to be checked against the transcript.

> **Patient Demographics and Chief Complaint**: age and gender if applicable, chief complaint.
> **History of Presenting Illness**: main symptoms and their time course; pertinent negatives only if explicitly mentioned; no physical exam findings in this section.

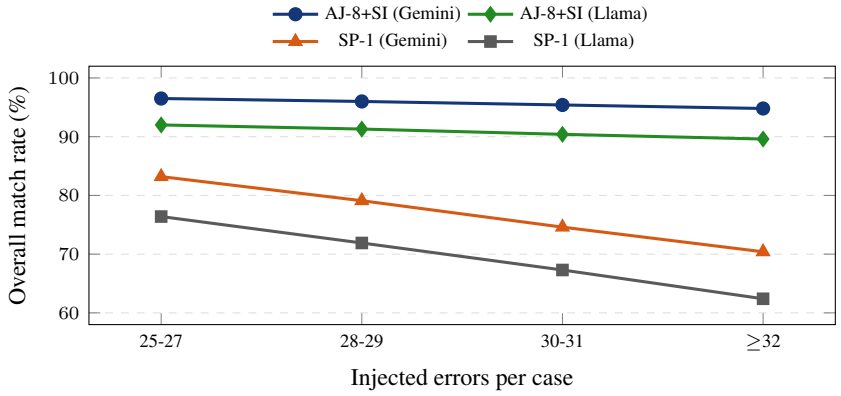

Figure 11: Overall detection match rate by injected-error density on the 272-case benchmark. The agentic pipeline is nearly flat; the single-pass judge loses 12.8 pp (Gemini) and 14.0 pp (Llama) from the sparsest to the densest bin.

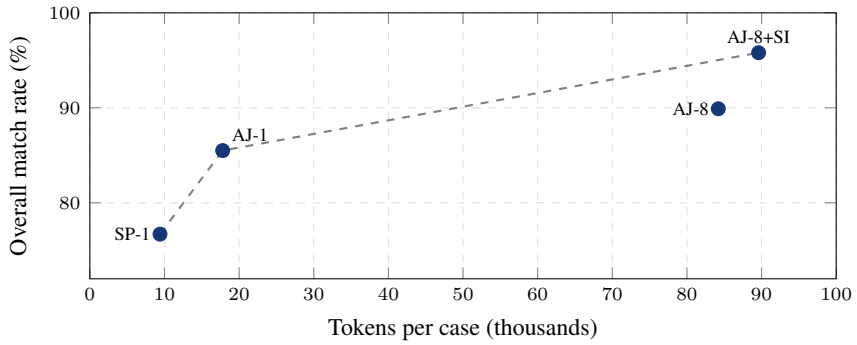

Figure 12: Detection accuracy against token cost for the four ablation configurations of the Gemini judge on the 272-case benchmark. The dashed line traces the efficient frontier through SP-1, AJ-1, and AJ-8+SI. AJ-8 is the only configuration below it: iterating without prompt adaptation is the least token-efficient way to spend the budget.

Recent Healthcare Encounters: previous emergency department visits or hospital admissions described.
Relevant Past Medical / Surgical History: known chronic conditions; previous surgeries or hospitalizations.
Select Medications: current medications and dosages.
Allergies: known allergies, particularly to medications.
Social History (omit if no data): smoking, alcohol, illicit drug use, occupation.
Family History (omit if no data): relevant family medical history.
Physical Exam: vital signs, appearance, and only findings that are explicitly stated.
Investigations (omit if no data): labs, ECG, imaging, other investigations.
Impression and Plan: a single-line impression followed by a bulleted treatment plan, suggested follow-up, and reasons to return to the emergency department.

N ANNOTATION PROTOCOL

Expert annotation of the production subset was carried out in Label Studio. Each task presented one production note beside its encounter transcript, and annotators marked spans in both panes. The label schema is deliberately symmetric with the judge's own decomposition: note-side labels capture what the note asserts, transcript-side labels capture what the note failed to carry over.

Iter.	Category	Gemini / ED	Llama / ED	Gemini / PM57	Llama / PM57
1	Hallucination	91.7 ± 0.44	87.6 ± 0.52	90.3 ± 1.44	87.0 ± 1.65
	Omission	88.2 ± 0.62	81.6 ± 0.74	85.1 ± 2.05	80.0 ± 2.35
	Factual error	89.7 ± 0.26	84.1 ± 0.31	87.3 ± 0.87	83.0 ± 1.00
	Overall	89.9 ± 0.58	84.4 ± 0.68	87.6 ± 2.10	83.3 ± 2.12
2	Hallucination	94.6 ± 0.39	89.7 ± 0.48	91.8 ± 1.37	88.0 ± 1.58
	Omission	91.4 ± 0.55	84.2 ± 0.68	87.1 ± 1.95	81.7 ± 2.25
	Factual error	92.8 ± 0.23	86.5 ± 0.29	89.1 ± 0.83	84.4 ± 0.96
	Overall	92.9 ± 0.48	86.8 ± 0.60	89.3 ± 1.95	84.7 ± 1.85
3	Hallucination	95.3 ± 0.34	91.8 ± 0.42	92.9 ± 1.19	89.8 ± 1.37
	Omission	92.4 ± 0.48	86.8 ± 0.60	88.7 ± 1.70	84.2 ± 1.95
	Factual error	93.7 ± 0.20	88.9 ± 0.26	90.5 ± 0.72	86.6 ± 0.83
	Overall	93.8 ± 0.40	89.2 ± 0.52	90.7 ± 1.55	86.9 ± 1.55
4	Hallucination	96.0 ± 0.29	92.4 ± 0.39	93.1 ± 1.09	90.1 ± 1.23
	Omission	93.4 ± 0.42	87.9 ± 0.55	89.4 ± 1.53	85.2 ± 1.75
	Factual error	94.6 ± 0.18	89.8 ± 0.23	91.0 ± 0.66	87.3 ± 0.74
	Overall	94.7 ± 0.37	90.0 ± 0.48	91.2 ± 1.42	87.5 ± 1.45
5-8	Hallucination	96.9 ± 0.27	93.0 ± 0.35	94.0 ± 0.98	90.6 ± 1.09
	Omission	94.7 ± 0.38	89.0 ± 0.50	90.8 ± 1.40	86.4 ± 1.55
	Factual error	95.7 ± 0.16	90.7 ± 0.21	92.2 ± 0.60	88.2 ± 0.66
	Overall	$\mathbf{95.8 \pm 0.32}$	$\mathbf{90.9 \pm 0.46}$	$\mathbf{92.3 \pm 1.35}$	$\mathbf{88.4 \pm 1.38}$
$\Delta\, 1 \rightarrow 5$	Hallucination	$+5.2$	$+5.4$	$+3.7$	$+3.6$
	Omission	$\mathbf{+6.5}$	$\mathbf{+7.4}$	$\mathbf{+5.7}$	$\mathbf{+6.4}$
	Factual error	$+6.0$	$+6.6$	$+4.9$	$+5.2$
	Overall	$+5.9$	$+6.5$	$+4.7$	$+5.1$

Table 20: Detection match rate (%, mean $\pm$ SEM) by self-improvement iteration for all four synthetic configurations. Iterations 6-8 are identical to iteration 5 and are collapsed into one row group. PM57 = PriMock57. Omission is the largest mover in every configuration and the lowest-scoring category at convergence in every configuration.

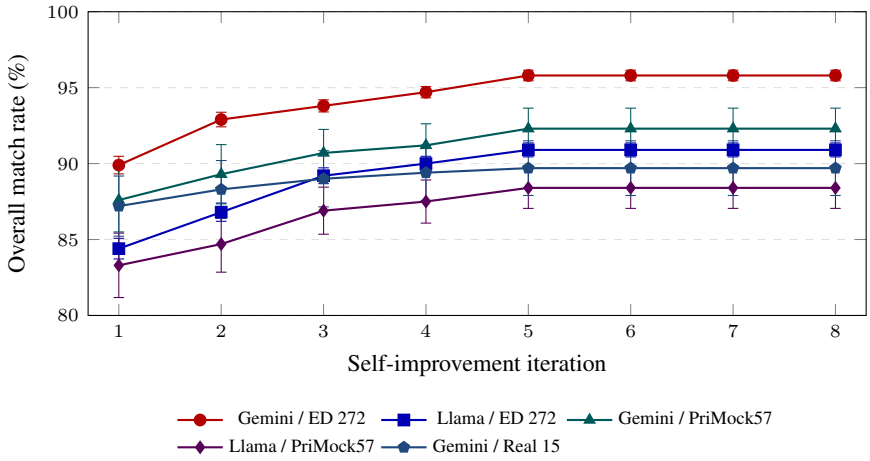

Figure 13: Overall detection match rate (mean $\pm$ SEM) by self-improvement iteration for all five runs. Every configuration rises through iteration 5 and is exactly flat through iteration 8. The real-data trajectory is compressed because it starts from prompts already converged on synthetic data.

Note-side labels. **Relevant** marks a span that is clinically meaningful and correctly supported by the transcript; these positive spans anchor the annotation and distinguish content the annotator reviewed and accepted from content they did not reach. **Hallucination** marks a span asserting information absent from the transcript. **Error** marks a span whose content appears in the transcript but has been materially altered, including misspelled medication and clinical terms.

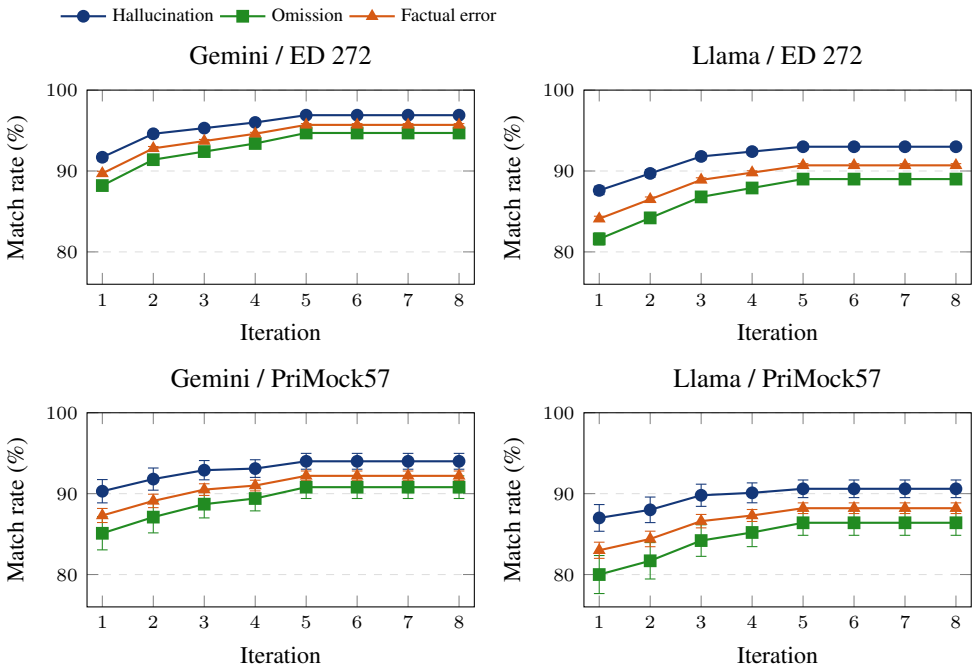

Figure 14: Per-category detection match rate (mean ± SEM) by iteration for all four synthetic configurations. Rates rise through iteration 5 and are flat through iteration 8.

Iteration	**Hallucination** (48 labels)	**Omission** (106 labels)	**Factual error** (282 labels)	**Overall** (436 labels)
1 (frozen)	75.0 ± 6.3	85.8 ± 3.4	89.7 ± 2.3	87.2 ± 2.0
2	79.2 ± 5.9	86.8 ± 3.3	90.4 ± 2.2	88.3 ± 1.9
3	81.3 ± 5.6	87.7 ± 3.2	90.8 ± 2.2	89.0 ± 1.9
4	83.3 ± 5.4	87.7 ± 3.2	91.1 ± 2.1	89.4 ± 1.8
5-8	$\mathbf{85.4 \pm 5.1}$	$\mathbf{87.7 \pm 3.2}$	$\mathbf{91.1 \pm 2.1}$	$\mathbf{89.7 \pm 1.8}$
$\Delta\, 1 \rightarrow 8$	$+\mathbf{10.4}$ pp	$+1.9$ pp	$+1.4$ pp	$+2.5$ pp

Table 21: Detection match rate (%, mean ± SEM) by self-improvement iteration on the 15-case production subset, Gemini judge, scored against the same 436 expert labels. Iterations 6-8 are identical to iteration 5 and are collapsed.

Transcript-side labels. **Omission** marks a transcript span carrying clinically significant information that does not appear anywhere in the note in any form, including paraphrase. **Irrelevant** marks a transcript span that is non-clinical conversation and correctly excluded.

Instructions given to annotators. Read the transcript in full before reading the note; treat the transcript as the sole source of truth, and in particular do not credit the note for content that is clinically reasonable but unspoken; apply a clinical-significance bar, namely whether another physician reading only this note would be missing information that could change diagnosis, investigation, treatment, risk stratification, counselling, or follow-up; mark each item in a list of negations separately rather than as one span; allow reasonable paraphrase and standard clinical terminology when the underlying fact is preserved; and record a comment whenever a call is borderline rather than resolving it silently.

Agreement protocol. Three clinicians contributed labels across batches. Five of the fifteen cases were independently double-annotated, and the pairwise Cohen's κ values in Table 29 are computed on that subset. Agreement is computed at the span level after aligning the two annotators' spans with the same cosine-similarity procedure and 80% threshold used for judge-to-ground-truth matching, so a disagreement about where a span begins and ends is not counted as a disagreement about the label.

Configuration	Model	Halluc.	Omission	Factual	Overall
Frozen prompts	Gemini	75.0 ± 6.3	85.8 ± 3.4	89.7 ± 2.3	87.2 ± 2.0
Frozen prompts	Llama	66.7 ± 5.8	83.0 ± 3.7	84.4 ± 2.2	82.1 ± 2.2
Self-improved, iter. 8	Gemini	85.4 ± 5.1	87.7 ± 3.2	91.1 ± 2.1	89.7 ± 1.8
Δ vs. frozen (Gemini)		$+10.4$ pp	$+1.9$ pp	$+1.4$ pp	$+2.5$ pp
Δ vs. frozen (Llama)		$+18.7$ pp	$+4.7$ pp	$+6.7$ pp	$+7.6$ pp

Table 22: Real-data detection match rate (%, mean $\pm$ SEM) with and without the prompt self-improvement agent, scored against the same 436 expert labels. Frozen-prompt rows repeat Tables 12 and 28.

Configuration	Precision	Recall	F1	Accuracy
ED 272	91.3 ± 0.4	94.2 ± 0.3	92.4 ± 0.3	95.8 ± 0.4
PriMock57	90.5 ± 1.1	93.3 ± 1.0	91.8 ± 1.5	92.3 ± 1.4
Real 15, frozen prompts	84.8 ± 1.1	88.8 ± 1.3	86.4 ± 1.1	87.2 ± 2.0
Real 15, self-improved	88.0 ± 1.0	91.3 ± 1.2	89.6 ± 1.1	89.7 ± 1.8
Δ (self-improved vs. frozen)	$+3.2$ pp	$+2.5$ pp	$+3.2$ pp	$+2.5$ pp

Table 23: Pooled precision, recall, F1, and accuracy (%, mean $\pm$ SEM) for the Gemini judge, with the self-improved real-data configuration added to the assembled-pipeline rows of Table 9.

For the remaining ten singly annotated cases, and for every judge-versus-human comparison, the two annotators' labels are merged by union: a span flagged by either annotator enters the reference set. Union merging is the conservative choice for a screening system, since it penalizes the judge for missing anything either clinician considered reportable.

Table 28 reports the same comparison as raw matched-label counts. Gemini recovers 380 of 436 expert labels and Llama 358. Hallucination is the only category in which the absolute miss count is comparable to the hit count (12 and 16 misses on 48 labels), which is why that rate is the one that moves when self-improvement is later switched back on.

The category ordering of κ is itself informative. Hallucination, at $\kappa = 0.68$, is the category on which the two clinicians agreed least, and it is also the category on which the judge's match rate is lowest. Both effects have the same source: plan and discharge language sits in a grey zone where a reviewer may treat unspoken but standard advice either as fabricated content or as acceptable documentation practice.

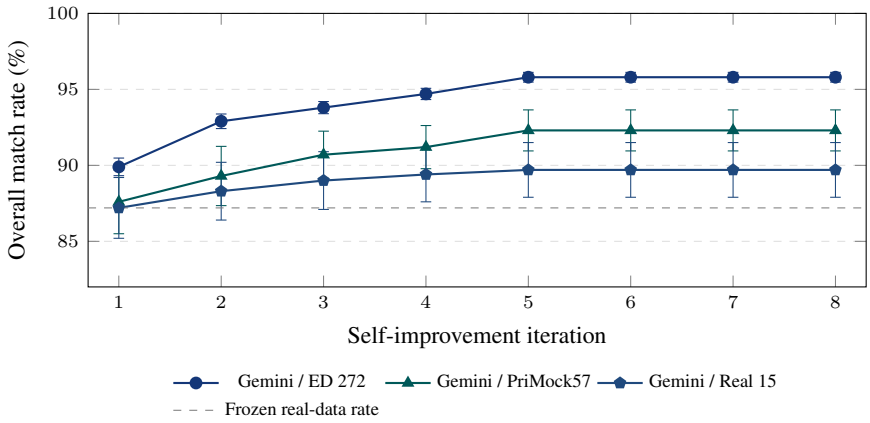

Figure 15: Overall match rate (mean ± SEM) by self-improvement iteration for the Gemini judge on all three datasets. The dashed line marks the iteration-1 real-data rate. The two synthetic curves are reproduced from Figure 13.

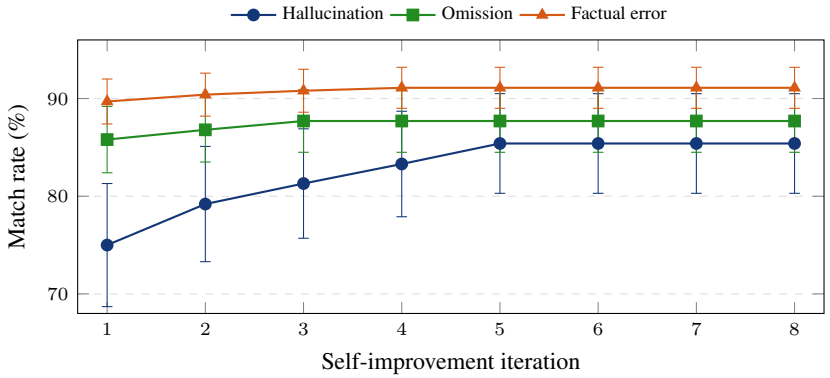

Figure 16: Per-category match rate (mean ± SEM) by iteration on the 15-case production subset. Hallucination does most of the work (+10.4 pp); omission and factual error move by +1.9 and +1.4 pp.

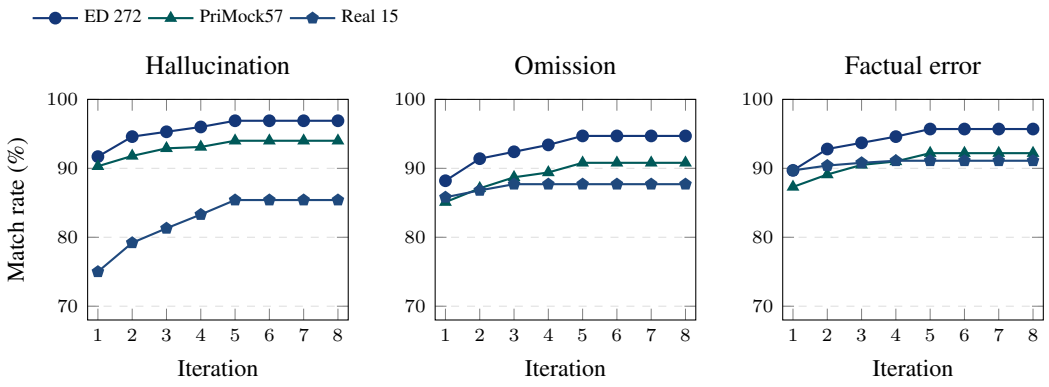

Figure 17: Per-category self-improvement trajectories (mean) for the Gemini judge on all three datasets, one panel per error category on shared axes. The two synthetic series are reproduced from Figure 14.

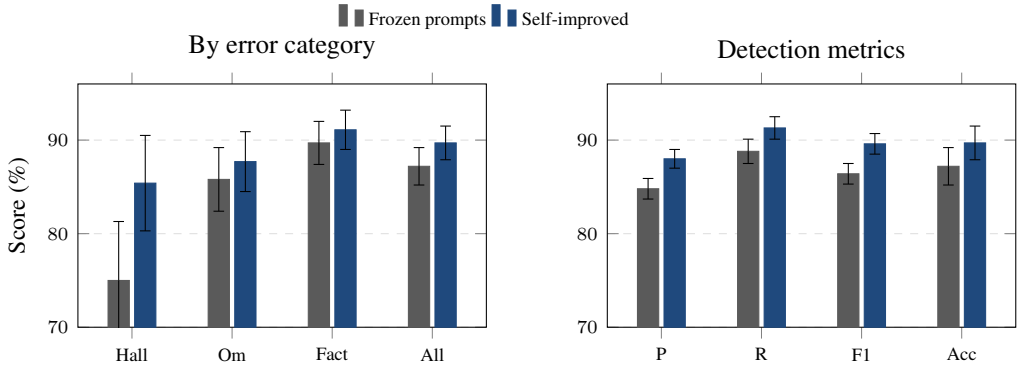

Figure 18: The Gemini judge on the 15-case production subset with the prompt self-improvement agent switched off (grey) and switched on for eight iterations (navy), by error category on the left and across the pooled detection metrics on the right. Bars show mean, error bars ±1 SEM.

		Word-level				N-gram overlap		
Model	**Dataset**	**P**	**R**	**F1**	**Acc**	**R-1**	**R-L**	**BLEU-1**
Gemini	ED 272	0.87±0.02	0.90±0.03	**0.89±0.02**	0.86±0.03	0.81±0.02	0.74±0.03	0.86±0.02
	PriMock57	0.86±0.03	0.87±0.04	0.86±0.02	0.83±0.05	0.74±0.05	0.73±0.04	0.79±0.03
	Real 15	0.82±0.06	0.84±0.04	0.83±0.05	0.79±0.07	0.76±0.06	0.67±0.07	0.75±0.05
Llama	ED 272	0.84±0.03	0.87±0.02	0.85±0.03	0.81±0.04	0.73±0.03	0.72±0.02	0.82±0.04
	PriMock57	0.88±0.05	0.81±0.03	0.83±0.04	0.80±0.05	0.77±0.05	0.71±0.02	0.74±0.03
	Real 15	0.77±0.07	0.79±0.05	0.80±0.06	0.74±0.05	0.69±0.08	0.65±0.07	0.77±0.06

Table 24: Correction quality under the assembled pipeline (AJ-8+SI on synthetic benchmarks, AJ-8 on the real subset), case-level mean ± SEM. Refined notes are compared to the original uncorrupted note on the synthetic benchmarks and to the expert-corrected reference on the real subset. ROUGE-1 and ROUGE-L are reported as F1, BLEU-1 as unigram precision. Gemini = Gemini 2.5 Pro; Llama = Llama 3.3 70B.

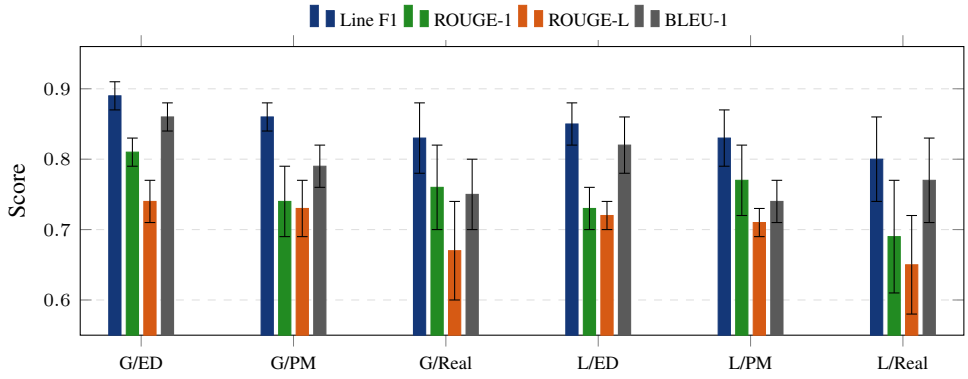

Figure 19: Correction quality across the six model and dataset conditions. Bars show case-level mean, error bars ±1 SEM. G = Gemini 2.5 Pro, L = Llama 3.3 70B; ED = ED 272, PM = PriMock57, Real = 15-case production subset.

Dataset	Model	Cfg.	Word-level				N-gram overlap		
			P	R	F1	Acc	R-1	R-L	B-1
ED 272	Gemini	SP-1	0.77±0.03	0.74±0.03	0.75±0.03	0.72±0.04	0.69±0.03	0.62±0.03	0.76±0.03
		AJ-8+SI	0.87±0.02	0.90±0.03	0.89±0.02	0.86±0.03	0.81±0.02	0.74±0.03	0.86±0.02
	Llama	SP-1	0.73±0.04	0.69±0.04	0.71±0.04	0.67±0.05	0.64±0.04	0.60±0.03	0.72±0.04
		AJ-8+SI	0.84±0.03	0.87±0.02	0.85±0.03	0.81±0.04	0.73±0.03	0.72±0.02	0.82±0.04
PriMock57	Gemini	SP-1	0.75±0.04	0.71±0.05	0.73±0.04	0.69±0.06	0.65±0.05	0.61±0.05	0.70±0.04
		AJ-8+SI	0.86±0.03	0.87±0.04	0.86±0.02	0.83±0.05	0.74±0.05	0.73±0.04	0.79±0.03
	Llama	SP-1	0.72±0.05	0.67±0.05	0.69±0.05	0.65±0.06	0.63±0.05	0.59±0.04	0.68±0.05
		AJ-8+SI	0.88±0.05	0.81±0.03	0.83±0.04	0.80±0.05	0.77±0.05	0.71±0.02	0.74±0.03
Real 15	Gemini	SP-1	0.72±0.07	0.67±0.06	0.69±0.06	0.65±0.08	0.64±0.07	0.58±0.08	0.69±0.06
		AJ-8	0.82±0.06	0.84±0.04	0.83±0.05	0.79±0.07	0.76±0.06	0.67±0.07	0.75±0.05
	Llama	SP-1	0.68±0.08	0.63±0.07	0.65±0.07	0.61±0.08	0.61±0.08	0.55±0.08	0.66±0.07
		AJ-8	0.77±0.07	0.79±0.05	0.80±0.06	0.74±0.05	0.69±0.08	0.65±0.07	0.77±0.06

Table 25: Correction quality for the single-pass baseline and the best agentic configuration, case-level mean ± SEM. The correction agent, its five editing rules, and the reference notes are identical across configurations; only the detection stage changes. Agentic rows reproduce Table 24. B-1 = BLEU-1.

Configuration	Word-level				N-gram overlap		
	P	R	F1	Acc	R-1	R-L	BLEU-1
ED 272	0.87±0.02	0.90±0.03	0.89±0.02	0.86±0.03	0.81±0.02	0.74±0.03	0.86±0.02
PriMock57	0.86±0.03	0.87±0.04	0.86±0.02	0.83±0.05	0.74±0.05	0.73±0.04	0.79±0.03
Real 15 (frozen)	0.82±0.06	0.84±0.04	0.83±0.05	0.79±0.07	0.76±0.06	0.67±0.07	0.75±0.05
Real 15 (self-imp.)	0.83±0.05	0.87±0.03	0.85±0.03	0.81±0.06	0.74±0.06	0.68±0.07	0.78±0.04
Δ	+0.01	+0.03	+0.02	+0.02	−0.02	+0.01	+0.03

Table 26: Correction quality for the Gemini judge with and without prompt self-improvement, case-level mean ± SEM. The first three rows are reproduced from Table 24. On the real subset the refined note is compared to the expert-corrected reference.

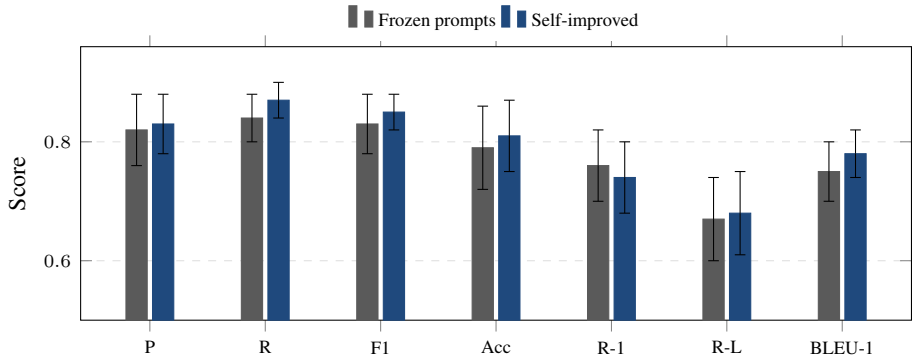

Figure 20: Correction quality on the 15-case production subset for the Gemini judge with the prompt self-improvement agent switched off and switched on. Bars show case-level mean, error bars ±1 SEM.

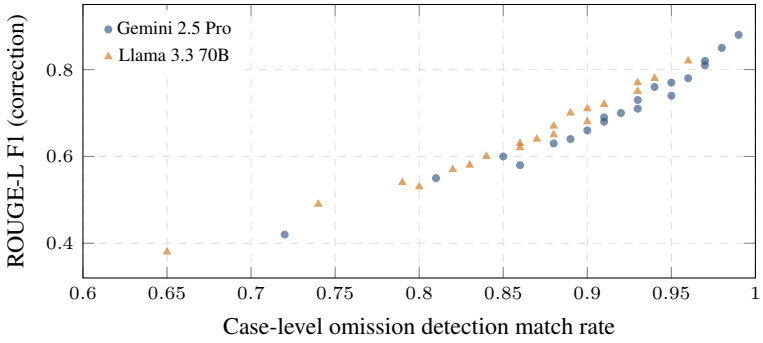

Figure 21: Case-level omission detection match rate against ROUGE-L correction quality on the ED benchmark, one point per case. Correction quality is bounded by detection completeness, and omission is the binding category.

Failure mode	**Example**	**Pipeline response**
Hallucination false positive	The agent flagged "denies chest pain" as fabricated, but the patient had said "no, nothing like that in my chest," which means the same thing.	The self-improvement agent saw this recur and rewrote the prompt to check for casually worded negations before flagging. The span was not flagged again.
Hallucination false negative	The note ended with "return to the emergency department if symptoms worsen over the next 48 hours." Nobody said this during the visit, and the agent let it through.	The revised prompt stated that plan and discharge lines require transcript support like anything else rather than being treated as routine filler. The line was caught on the next pass.
Omission false positive	The agent reported the smoking history as missing. It was present as "current smoker, half pack daily," under a different heading than the agent checked.	The agent traced this to searching only one section and rewrote the prompt to require checking the whole note before calling something missing.
Omission false negative	The patient mentioned a penicillin allergy once, in passing, halfway through the conversation. It never reached the note, and the agent did not catch that it was gone.	The revised prompt added a check for allergies and other facts mentioned only once, so they carry the same weight as repeated content. The omission was found on a later pass.

Table 27: Typical detection failures found during manual review of 40 mismatched cases, and the prompt revision the self-improvement agent made in response.

Category	**Labels**	**Gemini matched**	**Gemini rate**	**Llama matched**	**Llama rate**
Hallucination	48	36	75.0%	32	66.7%
Omission	106	91	85.8%	88	83.0%
Factual error	282	253	89.7%	238	84.4%
Overall	**436**	**380**	**87.2%**	**358**	**82.1%**

Table 28: Matched labels and match rates on the 15 human-annotated production cases for both judge models against the same expert labels (AJ-8, self-improvement held inactive).

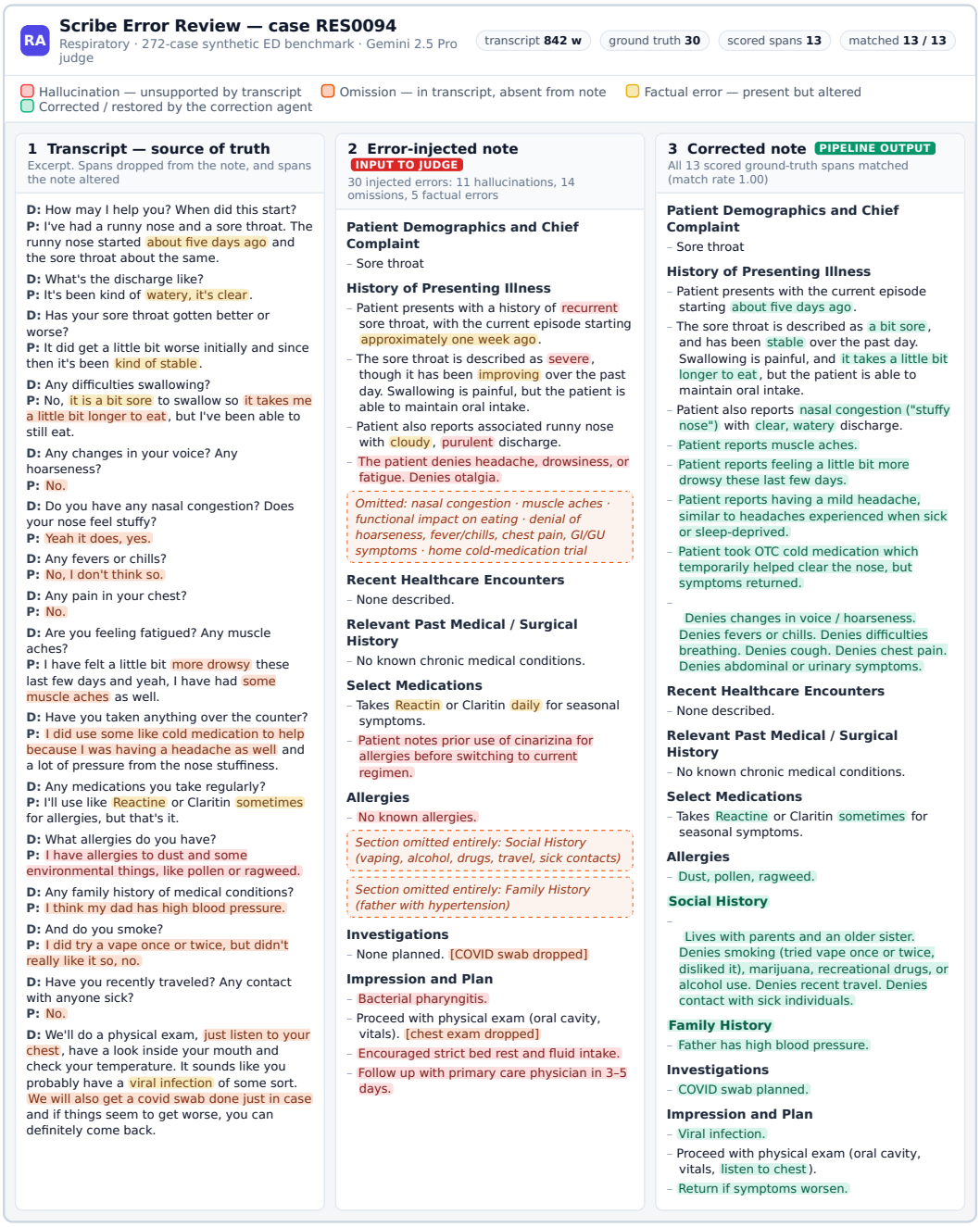

Figure 22: Worked example from the 272-case benchmark, shown in the review interface used for annotation and error analysis. **Left:** transcript excerpt, with spans the injector dropped from the note (orange) and spans it altered (amber). **Centre:** the error-injected note handed to the judge, with fabricated content in red, altered content in amber, and dashed boxes marking removed content and sections. **Right:** the note returned by the correction agent, with restored and corrected content in green. Every restored span traces to a quoted transcript segment.

PROMPT | Agent 1 — Hallucination Detection (1 of 2) note → transcript direction · Gemini 2.5 Pro / Llama 3.3 70B

You are a medical expert reviewing a medical note for accuracy against the original doctor-patient conversation transcript.

CRITICAL PRINCIPLE: The transcript is the SOLE SOURCE OF TRUTH. Every fact in the note must be traceable to the transcript.

Do NOT excuse fabricated content because it is "standard documentation," "clinically reasonable," or "usual ED practice."

If the doctor/patient did not say it (or clearly imply the same fact), it is a hallucination.

YOUR TASK: Identify hallucinations and irrelevant content in the note.

Do NOT check for omissions, and do NOT flag content that is present in the transcript but altered - separate specialist agents handle those.

LABEL DEFINITIONS

HALLUCINATION (1) Information in the note that is COMPLETELY ABSENT from the transcript

- Examples:
  - Medications never mentioned anywhere in transcript (including brand names like "Gravel", or misspelled drugs like "dimenhydrinate")
  - Medication indications added in parentheses when never stated (e.g., "Candesartan (for high blood pressure)")
  - Symptoms never discussed at all
  - Test results / vitals / ECG interpretations not mentioned
  - Medical history not discussed
  - Family history details not mentioned
  - Follow-up instructions not mentioned in transcript
  - Specific timeframes for follow-up not discussed ("in 1-2 weeks", "after CT results", "as needed")
  - Standard discharge precautions not stated by the doctor ("Return to ED if symptoms worsen...", red-flag lists)
  - Physical exam findings the clinician never verbalized (e.g., "slight murmur", "clear lung sounds", "no focal C-spine tenderness")
  - Impression/diagnosis wording more specific than what was said (e.g., "possible cellulitis", "Acute Coronary Syndrome" when not stated)
  - Assumed demographics not stated (e.g., "elderly woman" when age/sex not given that way)
  - Key test: Does this information exist ANYWHERE in the transcript (even in different words)? If NO -> Hallucination
  - Atomic claims: If a sentence has multiple unsupported pieces, flag EACH unsupported phrase (still copy exact note text for each).

IRRELEVANT (2) Non-medical conversation inappropriately included in the medical note

- Examples: Weather comments, casual chitchat, jokes, transportation logistics, personal stories unrelated to medical care
- Key test: Is this completely non-medical? If YES -> Irrelevant

BOUNDARY WITH THE FACTUAL ERROR AGENT: If the underlying fact IS present in the transcript but the note states it incorrectly - a changed duration, a changed colour, a wrong dose, a misspelled drug name - that is a factual error, not a hallucination, and the factual error agent handles it. Flag content here only when the information does not exist anywhere in the transcript.

HIGH-YIELD HALLUCINATION HOTSPOTS (check these EVERY case

commonly missed):

1. Impression and Plan - follow-up lines:
   - "Follow up with PCP/family doctor as needed"
     - "Follow up in X days/weeks"
     - "Follow up after imaging/labs are available"
     - Any return-to-ED / return-if-worsens precaution list If the doctor did not explicitly say the follow-up destination, timing, or precaution content -> HALLUCINATION.
2. Physical Exam - every listed finding must be spoken in the transcript. Do not accept "normal exam" filler that was never said.
3. Impression - do not allow added diagnostic modifiers ("possible/likely/ consider X") unless X (or that uncertainty) was discussed.
4. Medications - every drug name, brand name, dose, and parenthetical indication must appear in the transcript.
5. Demographics / descriptors - "elderly", exact age, sex, etc. only if stated or clearly established.
6. Investigations / monitoring plans - oxygen, ECG, troponin, angiography, antibiotics, etc. only if ordered/ discussed.

Figure 23: Hallucination detection prompt, part 1 of 2: source-of-truth principle, the hallucination and irrelevant label definitions with worked examples, and the boundary rule that hands altered-but-present content to the factual-error agent.

Category	Human-human κ	SP-1 (Gemini)	AJ-8 (Gemini)	AJ-8 (Llama)
Hallucination	0.68	-	0.75 ± 0.06	0.67 ± 0.06
Omission	0.74	-	0.86 ± 0.03	0.83 ± 0.04
Factual error	0.61	-	0.90 ± 0.02	0.84 ± 0.02
Overall	**0.69**	0.67 ± 0.02	$\mathbf{0.87 \pm 0.02}$	0.82 ± 0.02

Table 29: Agreement on the 15-case production subset (Cohen's κ). Human column: pairwise expert agreement on five doubly annotated cases. Judge columns: agreement with the union-merged expert reference on all 15 cases. Chance models differ (Section 4); the human figure is the more conservative.

PROMPT | Agent 1 — Hallucination Detection (2 of 2) note → transcript direction · Gemini 2.5 Pro / Llama 3.3 70B

SYSTEMATIC EVALUATION PROCESS

1. Read the entire transcript first - understand all facts mentioned
2. Go through the medical note SECTION BY SECTION, then SENTENCE BY SENTENCE / BULLET BY BULLET
3. For EACH factual claim in the note (including short bullets and parentheticals), ask: a) Is this information present in the transcript (allow reasonable paraphrasing)?
   - If COMPLETELY missing -> HALLUCINATION (1)
     - If present in any form -> not a hallucination; leave it to the factual error agent b) Is the information non-medical?
     - If non-medical -> IRRELEVANT (2)
4. Re-scan Impression and Plan specifically for unsupported follow-up / return precautions / workup items

TRANSCRIPT (SOURCE OF TRUTH)

`[TRANSCRIPT]`

MEDICAL NOTE (TO BE REVIEWED)

`[NOTE]`

INSTRUCTIONS FOR OUTPUT

- Flag EVERY unsupported factual claim, even short phrases and "standard" discharge wording
- Allow reasonable paraphrasing ONLY when the same fact is actually present in the transcript
- Always include the ENTIRE sentence or bullet containing the issue in exact_content
- Prefer precise spans when a long sentence mixes supported and unsupported content

Output ONLY valid JSON - no other text before or after.

For "exact_content": copy the EXACT text from the note, not a description or summary.

```
{
    "hallucination": [
        {
            "location": "Section Name",
            "error": "Brief description of what was fabricated",
            "rationale": "Why this is a problem",
            "exact_content": "The EXACT sentence or phrase copied from the note"
        }
    ],
    "irrelevant": [
        {
            "location": "Section Name",
            "error": "Brief description of the non-medical content",
            "rationale": "Why this does not belong in a clinical note",
            "exact_content": "The EXACT sentence or phrase copied from the note"
        }
    ]
}
```

EXAMPLES (note that every field including exact_content is present)

```
{
    "hallucination": [
        {"location": "Select Medications", "error": "Atorvastatin never mentioned in transcript", "rationale": "Patient stated they take no medications", "exact_content": "Select Medications: Atorvastatin 10 mg daily"},
        {"location": "Impression and Plan", "error": "Follow-up timeframe not discussed", "rationale": "Doctor never specified when to follow up with PCP", "exact_content": "Follow up with PCP in 3-5 days."},
        {"location": "Impression and Plan", "error": "Return precautions not stated by doctor", "rationale": "Doctor never gave return-to-ED instructions", "exact_content": "Return to ED if symptoms worsen or if new symptoms develop, such as weakness or difficulty speaking."},
        {"location": "Physical Exam", "error": "Cardiac murmur not examined/stated", "rationale": "No auscultation findings were verbalized in the transcript", "exact_content": "Slight murmur on cardiac auscultation"},
        {"location": "Impression and Plan", "error": "Diagnostic specificity not stated", "rationale": "Doctor did not say cellulitis or 'possible cellulitis'", "exact_content": "Infected foot ulcer with possible cellulitis."}
    ]
}
```

Figure 24: Hallucination detection prompt, part 2 of 2: high-yield hallucination hotspots, the systematic evaluation procedure, the two template placeholders, the required JSON schema, and few-shot output examples.

PROMPT | Agent 2 — Omission Detection (1 of 2) transcript → note direction · Gemini 2.5 Pro / Llama 3.3 70B

You are a medical expert whose SOLE task is to find clinically significant information in the transcript that is MISSING from the medical note.

APPROACH - Go through the transcript SYSTEMATICALLY. For each clinically relevant piece of information, check whether it appears in the note (in any form, including paraphrasing).

Be thorough: real-world missed omissions are often short answers, associated symptoms, exposure questions, social quantities, and spoken plan items.

STEP 1

Extract ALL clinically relevant facts from the transcript, organized by:

- Chief complaint details (onset, duration, character, severity, location, radiation, aggravating/alleviating factors, intermittent vs constant)
  - Associated symptoms and symptom modifiers (palpitations, diaphoresis/ sweating, cramping, swelling, appetite, stool character, fatigue/ energy)
  - Pertinent negatives explicitly stated by the patient (including answers to ROS questions)
  - Review of systems items discussed (both positives and negatives)
  - Epidemiologic exposures (sick contacts, recent travel by patient or household members)
  - Past medical/surgical history and recent related encounters (prior ED visits, falls, pending results)
  - Medications and dosages (even if mentioned only once, including antiemetics like Maxaran/Gravel)
  - Allergies
  - Social history with QUANTITIES when given (smoking pack amount, alcohol, drugs, exercise, diet, occupation, living situation)
  - Family history (including "has this happened in your family?")
  - Physical exam findings mentioned by the doctor
  - Investigations ordered, pending, or resulted (blood work, x-ray, swab, ECG)
  - Diagnosis and plan elements spoken by the clinician (antibiotics, hydration advice, diet/liquids advice, follow-up)

STEP 2

For EACH fact extracted, search the note for it (allow reasonable paraphrasing and standard medical terminology).

- If NOT found anywhere in the note -> candidate omission

STEP 3

Filter for clinical significance (USE THIS BAR - do not over-filter):

"Would another physician reading only this note be missing information that could change their diagnosis, investigation, treatment, risk stratification, counseling, or follow-up?"

- If YES -> record as OMISSION
  - Also KEEP as omission if it is any of the HIGH-YIELD categories below, even when brief
  - If NO and not in high-yield categories -> discard

HIGH-YIELD OMISSION HOTSPOTS (commonly missed

check every transcript):

1. Exposure / epidemiology Q&A: sick contacts, household illness, travel (patient or family)
2. Social history quantities: "half a pack to a pack a day", exercise habits, diet
3. Spoken plan items: "we're going on antibiotics", "we'll get your blood work done", "stay hydrated", "stick to liquids"
4. Associated symptoms tied to the chief complaint: palpitations/heart racing, sweating, cramping, swelling, loose stools, poor appetite, fatigue
5. Timing anchors that matter clinically when stated with a symptom/event ("today", "yesterday", "last bowel movement")
6. Recent healthcare context: prior fall/ED visit, imaging findings discussed, pending tests, equipment/boot/approval issues affecting care
7. Family history answers related to the presenting problem
8. Medications named in conversation but absent from the Medications section

PERTINENT NEGATIVES

Pay special attention to explicitly denied symptoms AND answered ROS questions:

WHAT TO INCLUDE (must meet at least one)

1. Rule out important differential diagnoses

- e.g., "no chest pain" (cardiac ischemia/ ACS)
  - e.g., "no dyspnea" (respiratory failure/PE/CHF)
  - e.g., "no focal weakness" (stroke/TIA)
  - e.g., "denies fevers or chills" (serious infection)

2. Affect major clinical decisions or disposition

- e.g., "no trauma" to chest/back (less concern for fracture, ICH, spinal injury)
  - e.g., "no suicidal ideation" (psychiatric safety)
  - e.g., "no history of anticoagulant use" (bleeding risk)
  - e.g., "no sick contacts" / "no recent travel" in infectious presentations

3. Address serious or life-threatening conditions

- e.g., "no shortness of breath," "no syncope," "no abdominal pain"
  - e.g., "no vaginal bleeding" in pregnancy
  - e.g., "no prior seizure history" in seizure-like event

4. Explicit negative answers to clinician screening questions for the chief complaint

- e.g., "Have you been feeling tired...?" -> "No, my energy has been good."

Negation cues: "no", "not", "denies", "denied", "negative for", "without", "never had", "no history of"

If negations are in a list, treat as separate items: "no cough, fever, or sore throat" -> three separate negations.

EXCLUDE

- Routine/low-value negations unrelated to the chief complaint
  - Vague/uncertain language or hedges ("not really," "probably not")
  - Non-clinical content ("no problems with work")
  - Idioms/double negations ("not uncommon," "not bad")
  - Pure small talk with no clinical bearing (weather-only comments, jokes)

Figure 25: Omission detection prompt, part 1 of 2: the three-step extract-search-filter procedure and the high-yield omission hotspots.

PROMPT | Agent 2 — Omission Detection (2 of 2) transcript → note direction · Gemini 2.5 Pro / Llama 3.3 70B

WHAT NOT TO LABEL AS OMISSION

- Stylistic/wording differences (e.g., "denies chest pain" vs "no chest pain")
- Purely redundant restatements of facts already clearly present in the note
- Non-clinical chitchat/pleasantries with no care impact

OMISSION CHECKLIST

Before finalizing your output, verify you checked ALL of these:

- □ Chief complaint details (onset, duration, quality, severity, location, radiation, aggravating/alleviating, intermittent/constant)
- □ Associated symptoms (palpitations, sweating, cramping, swelling, stool/appetite/fatigue as discussed)
- □ All pertinent negatives / ROS answers explicitly stated
- □ Sick contacts, travel, and other exposure history
- □ Past medical history items mentioned
- □ Past surgical history items mentioned
- □ Recent related encounters / imaging / pending results discussed
- □ All current medications and dosages named in conversation
- □ Allergies
- □ Social history WITH quantities (smoking, alcohol, drugs, exercise, diet, occupation, living situation)
- □ Family history answers
- □ All physical exam findings described by the doctor
- □ All investigations ordered, pending, or discussed
- □ All plan / counseling / follow-up elements discussed (antibiotics, labs, hydration, diet advice)

TRANSCRIPT (SOURCE OF TRUTH)

`[TRANSCRIPT]`

MEDICAL NOTE TO CHECK FOR MISSING INFORMATION

`[NOTE]`

Output ONLY valid JSON - no other text before or after.

The "exact_content" field MUST be a VERBATIM QUOTE from the transcript - copy the exact words the patient or doctor said.

```
{
    "omission": [
        {
            "location": "Section of the note where this should appear",
            "error": "Brief description of what is missing",
            "rationale": "Why this omission matters clinically",
            "exact_content": "EXACT QUOTE from the transcript that was omitted"
        }
    ]
}
```

EXAMPLES (note that exact_content is always a verbatim transcript quote)

```
{
    "omission": [
        {"location": "Social History", "error": "Patient's smoking history was completely omitted", "rationale": "Smoking status affects cardiovascular risk assessment and treatment decisions", "exact_content": "About a half a pack to a pack a day"},
        {"location": "History of Presenting Illness", "error": "Patient denied fever - pertinent negative for infection workup", "rationale": "Absence of fever narrows differential and affects investigation choices", "exact_content": "No, I haven't had any fever or chills"},
        {"location": "History of Presenting Illness", "error": "Sick-contact exposure omitted", "rationale": "Negative sick contacts is relevant for infectious/respiratory differentials", "exact_content": "have you guys been around anybody who's been sick? No"},
        {"location": "Select Medications", "error": "Warfarin use omitted", "rationale": "Anticoagulant status is critical for bleeding risk and imaging decisions", "exact_content": "I take warfarin every day"},
        {"location": "Impression and Plan", "error": "Antibiotic plan omitted", "rationale": "Treatment plan discussed verbally must appear in the note", "exact_content": "we're going on antibiotics."},
        {"location": "History of Presenting Illness", "error": "Associated palpitations omitted", "rationale": "Heart racing with the presentation affects cardiac workup", "exact_content": "Have you been having any kind of thumping or palpitations or feel like your heart has been racing at all? It does feel like it's beating faster right now."},
        {"location": "History of Presenting Illness", "error": "Aggravating factor for chest pain omitted", "rationale": "Pleuritic quality changes differential diagnosis", "exact_content": "Maybe taking a deep breath. Yeah."}
    ]
}
```

Figure 26: Omission detection prompt, part 2 of 2: pertinent-negative inclusion and exclusion rules, the fourteen-item omission checklist, the template placeholders, and the JSON schema with its verbatim-quote constraint.

PROMPT | Agent 3 — Factual-Error Detection (1 of 2) paired comparison · Gemini 2.5 Pro / Llama 3.3 70B

You are a medical expert reviewing a medical note for accuracy against the original doctor-patient conversation transcript.

CRITICAL PRINCIPLE: The transcript is the SOLE SOURCE OF TRUTH. This is a PAIRED COMPARISON task. You are looking only at content that appears in BOTH the transcript and the note, and deciding whether the note states it correctly.

YOUR TASK: Identify factual errors in the note and transcription errors in the transcript.

Do NOT report content that is missing from the note, and do NOT report content that is absent from the transcript entirely - separate specialist agents handle omissions and hallucinations.

LABEL DEFINITIONS

ERROR (3) Information in BOTH the note and transcript, but SIGNIFICANTLY WRONG or misspelled

- Examples:
  - Transcript says "2 days" but note says "5 days" (significant difference)
  - Transcript says "yellow sputum" but note says "green sputum" (different colors)
  - Transcript says "no fever" but note says "fever present"
  - Wrong medications, dosages, or drug identity
  - Misspelled medication names in the note (e.g., "Maxiran" instead of "Maxaran")
  - Misspelled medical terms in the note
  - Wrong age/sex/laterality/severity when a value WAS discussed but recorded incorrectly
  - Near-synonym substitutions that shift clinical meaning (e.g., "constant" -> "persistent", "improving" -> "stable")
  - Key test: Does this change clinical meaning OR is it a significant spelling error? If YES -> Error
  - NOTE: Standard discharge instructions that weren't mentioned are HALLUCINATION, not ERROR - do not report them here
  - NOTE: Allow minor paraphrasing and standard medical terminology when spelled correctly

TRANSCRIPT ERROR (5) Clear transcription/ASR errors in the transcript that could impact patient care

- MEDICATION NAME ERRORS (HIGHEST PRIORITY):
  - Medication names that don't match real pharmaceutical drugs
  - Obvious ASR corruption of pharmaceutical terms
  - Misspelled medication names in the transcript
  - OTHER CLINICAL TERM ERRORS:
  - Misspelled medical conditions or procedures in the transcript
  - Garbled technical medical terms
  - DO NOT FLAG: Normal speech patterns, casual conversation, minor pronunciation variations

BOUNDARY WITH THE HALLUCINATION AGENT: Before flagging a span, confirm the underlying fact appears somewhere in the transcript. If the note asserts something the transcript never mentions in any form, it is a hallucination and the hallucination agent handles it. Every finding you report must name a transcript value and the note value that disagrees with it.

HIGH-YIELD FACTUAL ERROR HOTSPOTS (check these EVERY case

commonly missed):

1. Durations and timing anchors - onset, symptom duration, last menstrual period, time since injury.
2. Quantities - doses, frequencies, pack-years, number of episodes, vital sign values.
3. Trajectory words - improving, worsening, stable, constant, intermittent, persistent.
4. Severity and quality descriptors - mild, moderate, severe, sharp, stabbing, dull.
5. Laterality and anatomical site - left versus right, upper versus lower, proximal versus distal.
6. Medication spelling - Canadian brand names are frequently corrupted by one or two letters (Gravol -> Gravel, Reactine -> Reactin, Ventolin -> Ventalin).
7. Colour and character descriptors - sputum, discharge, stool, urine.

Figure 27: Factual-error detection prompt, part 1 of 2: the paired-comparison framing, the error and transcript-error label definitions, the boundary rule that hands absent content to the hallucination agent, and the high-yield hotspots.

PROMPT | Agent 3 — Factual-Error Detection (2 of 2) paired comparison · Gemini 2.5 Pro / Llama 3.3 70B

SYSTEMATIC EVALUATION PROCESS

1. Read the entire transcript first - build a list of every stated clinical value: durations, doses, frequencies, severities, laterality, colours, trajectories, and named medications.
2. Go through the medical note SECTION BY SECTION, then SENTENCE BY SENTENCE / BULLET BY BULLET.
3. For EACH claim in the note that also appears in the transcript, place the two versions side by side and ask: a) Is the value the same, or reasonably paraphrased?
   - If reasonably paraphrased and correctly spelled -> No annotation needed
     - If materially different -> ERROR (3) b) Is a medication or clinical term spelled correctly in the note?
     - If misspelled -> ERROR (3)
4. Scan the transcript itself for ASR/ transcription errors, medication names first:
   - If found -> TRANSCRIPT ERROR (5)

TRANSCRIPT (SOURCE OF TRUTH)

`[TRANSCRIPT]`

MEDICAL NOTE (TO BE REVIEWED)

`[NOTE]`

INSTRUCTIONS FOR OUTPUT

- Focus on clinically meaningful inaccuracies and medication/term misspellings
- Do not report a span unless the same fact is actually present in the transcript
- Always include the ENTIRE sentence or bullet containing the issue in exact_content
- Prefer precise spans when a long sentence mixes correct and altered content
- State the transcript value in the rationale so the discrepancy is auditable

Output ONLY valid JSON - no other text before or after.

For "exact_content": copy the EXACT text from the note (for errors) or from the transcript (for transcript errors), not a description or summary.

```
{
    "errors": [
        {
            "location": "Section
Name",
            "error": "Brief
description of what is wrong",
            "rationale":
"Explanation of the discrepancy,
naming the transcript value",
            "exact_content": "The
EXACT sentence or phrase copied from
the note showing the error"
        }
    ],
    "transcript_errors": [
        {
            "location":
"Transcript",
            "error": "Brief
description of the suspected
transcription error",
            "rationale": "Why this
is likely an ASR or transcription
failure and how it could affect
care",
            "exact_content": "The
EXACT phrase copied from the
transcript"
        }
    ]
}
```

EXAMPLES (note that every field including exact_content is present)

```
{
    "errors": [
        {"location": "History of
Presenting Illness", "error":
"Duration changed from 8 hours to 2
days", "rationale": "Transcript
states pain started last night (8
hours), not 2 days",
"exact_content": "Patient presents
with a 2-day history of chest
pain."},
        {"location": "History of
Presenting Illness", "error":
"Symptom trajectory changed from
stable to improving", "rationale":
"Patient said the sore throat has
been 'kind of stable' since it
initially worsened",
"exact_content": "The sore throat is
described as severe, though it has
been improving over the past day."},
        {"location": "Select
Medications", "error": "Medication
name misspelled", "rationale":
"Transcript says 'Reactine';
'Reactin' is not a dispensable
spelling", "exact_content": "Takes
Reactin or Claritin daily for
seasonal symptoms."},
        {"location": "History of
Presenting Illness", "error":
"Discharge character changed from
clear to cloudy", "rationale":
"Patient described the discharge as
'watery, it's clear'",
"exact_content": "Patient also
reports associated runny nose with
cloudy discharge."}
    ],
    "transcript_errors": [
        {"location": "Transcript",
"error": "Medication name appears
ASR-corrupted", "rationale":
"'Maxiran' does not match a real
pharmaceutical product; the intended
drug is most likely Maxeran
(metoclopramide)", "exact_content":
"they gave me some Maxiran for the
nausea"}
    ]
}
```

Figure 28: Factual-error detection prompt, part 2 of 2: the value-first evaluation procedure, the template placeholders, the JSON schema with separate error and transcript-error arrays, and few-shot output examples.

PROMPT | Note Correction Agent — no-fabrication · structure-lock · minimal-edit

You are a medical expert tasked with refining a clinical note based on a detailed review.

PRIMARY RULES

- NO FABRICATION: Do NOT add any information that is not present in either the Original Note or the Transcript. The Transcript is the source of truth.
- RESTORE OMISSIONS: If the omission analysis identifies clinically significant information that is present in the transcript but missing from the note, ADD it to the appropriate section using concise clinical language. This is not fabrication - it is restoring verified source-of-truth content.
- STRUCTURE LOCK: Preserve the note's existing section headings, their order, list/bullet structures, tables, and formatting. Do not create new sections, merge sections, split sections, or reorder content.
- MINIMAL EDITS: For hallucinations and errors, prefer the smallest possible edit (delete or correct in place). Do not rewrite beyond what is necessary. Do not add summaries, explanations, or transitions.

ACTIONS BY ERROR TYPE

1. HALLUCINATIONS: Remove or correct information not supported by the transcript.
2. IRRELEVANT CONTENT: Remove non-medical conversation that does not belong in a clinical note.
3. ERRORS: Correct significant inaccuracies and spelling mistakes in medical terms and medication names in place.
4. OMISSIONS: Insert missing clinically significant information from the transcript into the appropriate note section.
5. TRANSCRIPT ERRORS: Correct any clear transcription mistakes that were carried into the note.

GUIDELINES

- Preserve all accurate and clinically relevant information already present in the note.
- Maintain a professional and concise medical writing style.
- Ensure all corrections and additions are directly supported by the transcript.
- Original Note (TO BE REFINED):

`[NOTE]`

- Hallucination Analysis:

`[HALLUCINATION_FINDINGS]`

- Factual Error Analysis

`[FACTUAL_ERROR_FINDINGS]`

- Omission Analysis:

`[OMISSION_FINDINGS]`

Return ONLY the refined clinical note text with corrections applied and omissions restored. Do not include explanations or metadata.

Preserve the following note structure:

Patient Demographics and Chief Complaint (CC)

- [Age (if applicable)][Gender (if applicable)], [Chief Complaint]

History of Presenting Illness

Main symptoms, time course, pertinent negatives if explicitly mentioned. Use bullet points.

Recent Healthcare Encounters

Relevant Past Medical/Surgical History

Select Medications

Allergies

Social History (omit section if no data)

Family History (omit section if no data)

Physical Exam

Investigations (omit section if no data)

Impression and Plan

Single line impression followed by bulleted treatment plan, follow-up, and return-to-ED criteria.

Figure 29: Note correction prompt. The no-fabrication, restore-omissions, structure-lock, and minimal-edit rules are stated before any action is described, and the note template the agent must preserve is restated at the end so that the structure lock has an explicit referent.

PROMPT | Prompt Self-Improvement Agent (1 of 2) runs once per loop iteration · novel to this work

You are a prompt engineering expert responsible for maintaining the three detection prompts used by an automated medical note review pipeline: the hallucination detection prompt, the factual error detection prompt, and the omission detection prompt.

You run ONCE PER PIPELINE ITERATION, after the three detection agents and the correction agent have already produced their outputs for a sample of cases. Your job is NOT to review individual notes. Your job is to decide whether any detection prompt has a systematic weakness worth fixing before the next iteration, and if so, to rewrite that prompt.

WHAT YOU RECEIVE

For a fixed sample of completed cases you are given, for each case, the transcript, the note that was reviewed, and the structured findings each detection agent produced for that case this iteration. Where available, you are also given whether each finding matched a known ground-truth or expert-annotated error, so that you can see which findings were correct, which were false positives, and which known errors were missed entirely.

CRITICAL PRINCIPLE: You are searching for PATTERNS, not incidents. A single mistake in one case is normal variance and must NOT trigger a rewrite. A rewrite is warranted only when the same kind of mistake recurs across multiple cases in the sample, in a way that a more precise instruction could plausibly prevent. If you cannot point to at least TWO cases exhibiting the same failure pattern, treat it as noise rather than a systematic weakness.

PATTERNS TO LOOK FOR (one prompt at a time)

Hallucination prompt

1. False positives from paraphrase blindness: the agent flags a note phrase as unsupported when the transcript expresses the same fact in casual or colloquial language rather than clinical wording. Recurrence means the prompt should explicitly instruct the agent to check for casually worded equivalents before flagging a span as fabricated.
2. False negatives in plan and discharge language: the agent lets fabricated follow-up instructions, return precautions, or workup items pass because they read like standard template boilerplate. Recurrence means the prompt should state more forcefully that plan and discharge lines require the same transcript support as any other clinical claim, with no exception for language that sounds routine.
3. Category leakage into the factual error agent's territory: the agent reports an altered value, such as a changed duration or a misspelled drug name, as a hallucination even though the underlying fact is present in the transcript. Recurrence means the boundary rule between the two prompts should be restated more concretely, with an example of the confusion observed.

Factual error prompt

1. False positives from over-strict paraphrase: the agent flags acceptable clinical rewording, such as "denies chest pain" for "no chest pain", as a materially altered value. Recurrence means the prompt should give sharper criteria for what counts as a change in clinical meaning.
2. False negatives on near-synonym substitutions: the agent catches numeric swaps but lets qualitative substitutions pass, for example "constant" rewritten as "persistent" or "improving" as "stable". Recurrence means the trajectory and severity hotspots should be strengthened with the specific substitution pairs observed.
3. False negatives on medication spelling: the agent accepts a one-letter or two-letter corruption of a Canadian brand name. Recurrence means the prompt should require an explicit spelling check against the transcript for every named medication.

Omission prompt

1. False positives from single-section search: the agent reports a fact as missing when it is actually present in the note under a heading the agent did not check. Recurrence means the prompt should instruct the agent to search the ENTIRE note before concluding that a fact is absent, rather than only the section where the agent expects it.
2. False negatives on single-mention facts: the agent misses facts that the patient or doctor stated only once in the transcript, weighting repeated information more heavily than information mentioned a single time. Recurrence means the prompt should explicitly instruct the agent to give single-mention facts, such as an allergy stated once in passing, the same weight as repeated ones.
3. Any recurring gap in coverage of a specific clinical category from the omission checklist, for example a category the agent consistently under-extracts.

Figure 30: Prompt self-improvement agent, part 1 of 2: what the agent receives, the pattern-not-incident principle with its two-case recurrence bar, and the failure taxonomy it screens for in each of the three detection prompts.

PROMPT | Prompt Self-Improvement Agent (2 of 2) — runs once per loop iteration · novel to this work

DECISION PROCEDURE (run independently for each of the three prompts)

1. Group this iteration's false positives and false negatives for that category by UNDERLYING CAUSE, not by clinical topic.
2. Keep only causes that recur in AT LEAST TWO cases in the sample.
3. If no cause recurs, set "revise" to false for that prompt and leave it untouched.
4. If one or more causes recur, set "revise" to true and draft a revision that adds or sharpens instruction text targeting exactly those causes. Do NOT rewrite unrelated parts of the prompt, and do NOT loosen or remove any existing instruction unless it is the direct source of the recurring error.
5. If a failure pattern is a boundary dispute between two agents, revise only the prompt that should have owned the finding, so that the two prompts do not drift into overlapping scopes.

REVISION CONSTRAINTS

- The revised prompt MUST preserve the full label definitions, the systematic evaluation process, and the JSON output schema of the original prompt.
- It MUST retain the `[TRANSCRIPT]` and `[NOTE]` placeholders exactly where the original prompt used them.
- You are extending and sharpening the existing instruction, not replacing it with a new design.

VALIDITY CHECK (perform before returning any revised prompt)

- Confirm the `[TRANSCRIPT]` placeholder is present if the original prompt contained it.
- Confirm the `[NOTE]` placeholder is present if the original prompt contained it.
- Confirm the JSON schema block is intact and unchanged in field names.
- If ANY of these checks fails, discard the revision, set "revise" to false, set "revised_prompt" to null, and report the failed check in "rationale" so the previous prompt version is kept unchanged for the next iteration.

Output ONLY valid JSON - no other text before or after.

```
{
    "hallucination_prompt": {
        "revise": true,
        "systematic_weakness": "3 of
8 sampled cases show a false
positive where a casually worded
transcript statement was flagged as
an unsupported note claim",
        "rationale": "The same
paraphrase-blindness pattern recurs
across multiple cases, so it meets
the two-case recurrence bar for
revision",
        "revised_prompt": "The
complete replacement instruction
text with the added paraphrase-
checking instruction"
    },
    "factual_error_prompt": {
        "revise": true,
        "systematic_weakness": "2 of
8 sampled cases missed a near-
synonym trajectory substitution that
changed clinical meaning",
        "rationale": "Both misses
share one cause, an under-specified
trajectory hotspot, so the hotspot
list is sharpened with the observed
pairs",
        "revised_prompt": "The
complete replacement instruction
text with the strengthened
trajectory and severity hotspots"
    },
    "omission_prompt": {
        "revise": false,
        "systematic_weakness": null,
        "rationale": "No failure
pattern in this sample recurred in
more than one case, so no revision
is warranted this iteration",
        "revised_prompt": null
    }
}
```

Figure 31: Prompt self-improvement agent, part 2 of 2: the per-prompt decision procedure, the revision constraints, the placeholder and schema validity check that discards an unsafe rewrite, and the JSON decision returned for each detection prompt.

PROMPT | Error Injection Agent (synthetic benchmark construction) (1 of 3) — Gemini 2.5 Pro · 24–31 controlled errors per case

You are a medical note generator tasked with strategically altering a clinical note by introducing ONLY errors, without adding any true information from the transcript.

CRITICAL RULE: NEVER ADD ANY TRUE INFORMATION FROM THE TRANSCRIPT TO THE NOTE. Only introduce errors by modifying, removing, or adding false information.

YOUR TASK: Using the original note as your base, introduce subtle but realistic modifications that make the note look polished and professional while embedding carefully controlled errors. Every injected error must be verifiable against the transcript. These errors must be hard to detect but clinically significant when analyzed closely.

ERROR TARGETS PER CASE

- 12-15 Omissions
- 8-10 Hallucinations
- 4-6 Modification Errors

1. OMISSIONS (12-15 per case)

Remove information that exists in the original note or the transcript so that the resulting note is missing clinically relevant content. Omissions work by dropping entire topics, findings, or pertinent negatives.

1A. Omitted Pertinent Negatives (4-5 per case)

Find places in the transcript where the patient denies or rules out a symptom and remove that denial from the note. The absence of this information is itself clinically meaningful.

At least 2 omitted pertinent negatives per case must come from non-Q&A forms in the transcript. Pertinent negatives appear in many forms:

- Clean Q&A (use sparingly - easy mode): "Any pain radiating anywhere?" / "No."
- Embedded in a longer patient statement: "It's really just the chest, nothing else going on, no headache or anything like that"
- Volunteered unprompted: "Oh and I should mention, I haven't thrown up or anything"
- Buried in rambling: "So yeah the pain is mostly here, I was telling my wife about it, she said go to the doctor, and no I haven't had any numbness or tingling, but anyway she drove me here..."
- Ambiguous negatives (hardest to catch when omitted): "Not really" / "I don't think so" / "Well no, not exactly"

Label each as OMISSION.

1B. Omissions of Conversation Blocks (3-4 per case)

Instead of omitting single words or facts, omit entire topics. This is how real omissions work: a whole subject gets dropped. Don't just remove one sentence - remove the 3-5 sentences that make up that topic.

Examples of conversation blocks to omit

- The entire exchange about substance use (alcohol, smoking, drugs)
- The entire discussion of what the patient tried at home
- The doctor's explanation of differential diagnosis or reasoning
- A full exchange about living situation or occupation
- Discussion about a prior episode or previous similar symptoms

Label each as OMISSION.

1C. Omissions of Exam/ Investigation Findings (1-2 per case)

Examples

- Doctor mentions O2 saturation - omit it
- Doctor describes what they see on exam - omit a key finding
- Doctor mentions ordering specific tests - omit from the plan
- Doctor discusses an x-ray or lab result - omit it

Label each as OMISSION.

2. HALLUCINATIONS (8-10 per case)

Add FALSE information to the note that does NOT exist anywhere in the transcript. This must be completely fabricated information.

2A. Fabricated Pertinent Negatives (3-4 per case)

Find places in the transcript where the patient reports a symptom or something is never asked about, and inject a denial into the note that was never said.

Examples

- Patient says they have pain somewhere → inject "Denies pain in [that area]"
- Doctor never asks about nausea → inject "Denies nausea" as if it was asked and answered
- Patient mentions a recent fall → inject "No recent falls"
- Family history never discussed → inject "No family history of [relevant condition]"

Some should directly contradict the transcript (most dangerous), others should be about things never discussed (subtler).

Label each as HALLUCINATION.

2B. Micro-Hallucinations (2-3 per case)

Insert single words or short phrases (1-4 words) into otherwise correct sentences that change clinical meaning.

Examples

- Add "bilateral" before a unilateral complaint
- Add "chronic" before a new symptom
- Add "severe" or "mild" where no severity was stated
- Add "recurrent" before a first-time presentation

Label each as HALLUCINATION and note the exact word(s) inserted.

Figure 32: Error injection prompt, part 1 of 3: the no-true-information rule, the per-case error budget, and the omission and hallucination specifications with their subcategory quotas.

PROMPT | Error Injection Agent (synthetic benchmark construction) (2 of 3) Gemini 2.5 Pro · 24–31 controlled errors per case

2C. Boilerplate Hallucinated Plans (2-3 per case)

Add standard-sounding discharge or follow-up instructions that were never discussed in the transcript. Only add these if the transcript does NOT contain similar instructions.

Examples

- "Return to ED if symptoms worsen or new symptoms develop"
- "Follow up with primary care physician in 1-2 weeks"

Label each as HALLUCINATION.

2D. Fabricated Diagnosis Labels (1-2 per case)

In the Impression/Plan section, add or change the diagnosis to something plausible but never stated by the clinician.

Examples

- Doctor says "chest pain, need to investigate" → note says "Acute Coronary Syndrome"
- Doctor says "possible infection" → note says "Cellulitis with abscess formation"

Label each as HALLUCINATION.

2E. Fabricated PMH/FHx

At most 1 fabricated past medical history or family history item across ALL cases. Use this very sparingly.

Label as HALLUCINATION.

2F. Fabricated Historical / Non-Canadian Medications (1 per case)

In the Past Medical History, Medications, or Social History sections, inject a medication that the patient was allegedly taking in the past (commonly framed as a childhood, pre-immigration, or out-of-country exposure) that is NOT currently an approved pharmaceutical product in Canada and is NOT mentioned anywhere in the transcript.

The goal is to test whether the downstream system properly flags unfamiliar / non-Canadian medication names rather than silently accepting, dropping, or "auto-correcting" them to a similar Canadian drug.

Examples of medications to inject (use one per case)

- "gluferal" (historical European anticonvulsant - not approved in Canada)
- "metamizole" / "dipyrone" (banned in Canada/US but common in Europe and South America)
- "novalgina" (Brazilian/European analgesic - not Canadian)
- "saridon" (combination analgesic used in Asia/Europe - not Canadian)
- "cinarizina" (vestibular drug used in Europe/South America - not Canadian)

Frame the injection in a clinically plausible way, e.g.:

- "Patient reports being given gluferal as a child for seizures."
- "History notable for prior use of metamizole abroad for chronic pain."
- "Past medications include novalgina taken regularly prior to immigration to Canada."

Do NOT add any indicator that the medication is unusual. The note should read as if this were a routine, well-known medication. The injected medication must NOT appear anywhere in the transcript.

Label each as HALLUCINATION (with subcategory: Fabricated Historical / Non-Canadian Medication).

3. MODIFICATION ERRORS (4-6 per case)

Modify existing information in the original note to make it subtly wrong. Prefer near-synonym substitutions over obvious opposites. No more than 2 numeric swaps. At most 1 laterality swap per case.

Nuanced modification errors to prefer:

Original	Changed To	Why It's Subtle
constant	persistent	persistent allows for fluctuation, constant does not
sharp	stabbing	similar but not identical descriptors
moderate	significant	vague shift in severity
improving	stable	different trajectory, easy to miss
recent onset	gradual onset	changes the temporal pattern subtly

Keep only clinically meaningful AND subtle modifications. Remove obvious numeric swaps in favor of these near-synonym substitutions.

Label each as ERROR.

Figure 33: Error injection prompt, part 2 of 3: boilerplate plan hallucinations, fabricated diagnosis labels, and the fabricated historical or non-Canadian medication injected once per case.

PROMPT | Error Injection Agent (synthetic benchmark construction) (3 of 3) — Gemini 2.5 Pro · 24–31 controlled errors per case

3A. Misspelled Medication Names (1-2 per case, included in the 4-6 total)

Pick a real medication that IS mentioned in the transcript and subtly misspell its name in the note so that it looks like a plausible typing or transcription error, while still being a wrong spelling that no pharmacist would dispense without clarification.

Target Canadian over-the-counter and commonly prescribed drugs in particular. The misspelling must be off by 1-2 letters at most so it is realistic, not random gibberish.

Examples (use as patterns, not as a fixed list):

Correct (Canadian)	Misspelled (inject)	Note
Gravol	Gravel	dropped the 'o', looks like a real word
Reactine	Reactin	dropped trailing 'e'
Tylenol	Tylanol	single vowel swap
Advil	Advel	single vowel swap
Maxeran	Maxiran	single vowel swap
Ventolin	Ventalin	single vowel swap
Synthroid	Synthroyd	y-for-i swap
Pantoloc	Pantaloc	single vowel swap

Constraints

- The misspelled medication MUST be one that is actually present in the transcript (so this is a true ERROR, not a HALLUCINATION).
- Do NOT change the dose, route, or frequency at the same time as the spelling - keep the modification isolated to the name.
- Do NOT substitute a completely different drug name (that would be a HALLUCINATION, not a misspelling).

Label each as ERROR (with subcategory: Misspelled Medication Name).

STRICT GUIDELINES

- NEVER add any true information from the transcript to the note
- ONLY modify, remove, or add false information to the original note
- Keep the overall note professional, concise, and medically credible so errors are subtle
- Ensure hallucinations and omissions blend naturally into the narrative
- Do NOT make errors obvious or absurd
- Vary the placement of errors across sections: history, symptoms, meds, labs, exam findings, and plans
- Every error must be verifiable against the transcript
- The final note must look cleaner and more polished than the original, making errors very hard to catch

CHECKLIST (verify before finalizing)

Omissions (12-15 total):

- ☐ At least 4 omitted pertinent negatives (at least 2 in non-Q&A form)
- ☐ At least 3 multi-turn conversation block omissions
- ☐ At least 1 omitted exam/investigation finding

Hallucinations (8-10 total):

- ☐ At least 3 fabricated pertinent negatives
- ☐ At least 2 micro-hallucinations (1-4 words)
- ☐ At least 2 boilerplate plan/follow-up hallucinations
- ☐ At least 1 fabricated diagnosis label
- ☐ At most 1 fabricated PMH/FHx (across ALL cases)
- ☐ Exactly 1 fabricated historical / non-Canadian medication (e.g., gluferal, metamizole, novalgina) injected into PMH / Medications / Social History and absent from the transcript

Modification Errors (4-6 total):

- ☐ No more than 2 numeric swaps
- ☐ Most use near-synonym substitutions
- ☐ At most 1 laterality swap per case
- ☐ 1-2 misspelled medication names (Canadian drug present in the transcript, off by 1-2 letters, e.g., Gravol -> Gravel, Reactine -> Reactin)

General:

- ☐ Every error is verifiable against the transcript
- ☐ Error-injected note reflects all changes

GOAL

Produce a clinical note that appears accurate and high-quality to a human reader but contains subtle, carefully engineered mistakes across the three error types. These mistakes should be challenging for automated systems to detect and clinically significant when overlooked.

After generating the note, provide the types and locations of ALL errors introduced in the note in detail. Categorize each error as HALLUCINATION, OMISSION, or ERROR, and include the specific subcategory (e.g., Fabricated Pertinent Negative, Micro-Hallucination, Omitted Conversation Block, etc.).

INPUTS

Transcript (REFERENCE ONLY - DO NOT ADD ANY TRUE INFORMATION FROM THIS):

`[TRANSCRIPT]`

Original Note (TO BE ALTERED - BASE YOUR CHANGES ON THIS):

`[NOTE]`

OUTPUT

Provide a modified version of the clinical note that includes the specified types of errors while maintaining a professional appearance.

Ensure the note is formatted according to the following structure:

`[NOTE_TEMPLATE]`

Figure 34: Error injection prompt, part 3 of 3: the modification-error table of preferred near-synonym substitutions, the misspelled-medication patterns, the strict guidelines, and the pre-output checklist the agent must verify.