

The Blocker Beasts and How to Tame Them: Schema-Driven Extraction in the Connective AI Layer of Network Operations

Madhava Gaikwad*

Independent

gaikwad.madhav@gmail.com

Mohan Maddula

Microsoft

momaddul@microsoft.com

Abstract

In a datacenter network, a technician repairs a link and an on-call engineer, on a different shift and a different continent, decides whether the repair holds. The only channel between them is the technician’s free-text note. When the note omits the measurements the engineer needs, the ticket stalls. The median blocked ticket took 7.0 days to resolve, of which roughly one hour was engineering work. The rest was waiting for an account of the work that the note did not carry.

We treat the note as the object to be repaired. A pipeline in production for ten months parses each note at write time into two lightweight schemas, STAR and ECO, using deterministic rules for measurements and a language model for prose. It reaches 0.913 F1 against a 2,141-note gold standard, above a stronger frontier model working alone, and across a staggered rollout of 450+ datacenters the deployed package cuts time-to-repair by 3.4 days. Resolution time falls steeply in a completeness score, at a rate that holds across five physically unrelated failure types, which is what we would expect if the constraint were in the handoff rather than in the physical work.

1 Introduction

Monitoring in a hyperscale datacenter network flags a degrading link; a datacenter technician replaces the marginal optical transceiver. Hours later and time zones away, an on-call engineer must confirm from the technician’s note that the fix held. On a fraction of tickets, they cannot. “Reseated optic, link looks fine” does not say whether the error counter returned to zero, whether the link has *flapped* since (gone down and come back up, the signature of an intermittent fault), or what the optical power reads. The engineer asks; the technician,

now off shift, answers the next day; the exchange repeats. We address that gap.

A few terms carry the paper (Appendix A.1 expands them). A *link* is an optical connection between two switches, and its health is read from hardware counters. The Cyclic Redundancy Check (CRC) counter increments once per corrupted frame; a rising count is the standard degradation signal, most often from an aging optical transceiver (the pluggable fiber endpoint). A repair is verifiable from telemetry. The counter stops, optical power returns to nominal, and the link stops *flapping*. The technician is on site and repairs; the on-call engineer is remote and decides closure from the note and the telemetry, one shift removed, through the asynchronous ticket. While a ticket is open the link is *derated*: no new customer workload lands on it until closure returns its capacity to the pool.

The tickets this paper targets are the ones where that loop stalls. We call them *blocked* tickets; Section 4 gives the measurement rule. About 3% of incident tickets blocked under the pre-deployment process, where the median blocked ticket took 7.0 days to resolve. Roughly one hour of that is engineering work. The rest is a note that failed to carry what the next reader needed, plus a shift boundary. The bottleneck is an asynchronous handoff under incomplete information, not the repair. At fleet scale, the cost is paid in customer-allocatable capacity held off-pool while tickets sit open.

This paper presents an extraction pipeline, deployed in the Microsoft Azure datacenter network, that parses the free-text technician note into structured fields at write time. The fields are drawn from two lightweight schemas. STAR (Situation, Task, Action, Result), adapted from structured-interview methodology (Development Dimensions International, n.d.), gives every note a slot for the four facts the engineer needs. ECO (Evidence, Claim, Opinion), adapted from argumentation theory (Toulmin, 1958), labels each statement as a measurement,

*This work was performed when Madhava Gaikwad was with Microsoft.

an assertion that depends on a measurement, or an unsupported judgment. The pipeline combines regex rules over technical fragments with a large language model over free prose. The parsed fields land in the same operational database that holds the device telemetry.

Rule-plus-LLM hybrids, structured output, and schema-driven extraction are individually familiar (Section 5); we make four contributions. (1) A hybrid rules-plus-LLM extraction pipeline for free-text technician notes (Section 3), evaluated against a 2,141-note gold standard and three operational outcomes. (2) Intrinsic evaluation: 91.3% F1 against the gold standard, at 1.2-second median latency across 15,247 production tickets, with the rule layer’s margin persisting across model generations and a stronger pure-LLM configuration still trailing the deployed hybrid (Section 4). (3) Extrinsic evaluation on the blocked-ticket subpopulation across three outcomes, and, because those are pre/post comparisons, group-time average treatment effects over the staggered rollout with not-yet-treated datacenters as controls (Section 4). (4) An empirical regularity within this deployment: resolution time falls nonlinearly in an extraction-completeness score σ , with exponential and power-law parameterizations fitting comparably; under the exponential fit, every per-category 95% interval for the decay parameter contains the pooled estimate across five failure types (Section 4). An information-theoretic reading motivates that parameterization as interpretation, not derivation (Appendix C.4).

We release the extraction prompt. The sanitized pattern classes are given in Table 4 and the matching protocol is specified in Appendix C.1 in enough detail to reimplement. The gold standard consists of production technician notes and cannot be released, and implementation code is not released.

2 Schemas as Extraction Targets

The pipeline parses each technician note into two structured records.

STAR. A STAR record has four text-valued fields: Situation, Task, Action, and Result. It asks four questions of every note — what was observed, what was the goal, what was done, what was the outcome — and a note that satisfies STAR fills all four.

ECO. An ECO record classifies each statement in the note as one of three types. Evidence is a mea-

surement, a counter value, an instrument reading, or a verifiable observation. Claim is an assertion that depends on a measurement. Opinion is an unsupported judgment. It produces three lists per note; an ECO record is well-formed when every Claim has at least one supporting Evidence item.

The schemas are deliberately simple. Both have decades of use outside computing, which makes reliable gold-standard inter-annotator agreement attainable, and they are small enough for an LLM to fill in one call. They are also expressive enough that the downstream task, an engineer closing a ticket, genuinely depends on whether the fields are filled. They are extraction targets; the schemas themselves do little of the engineering work. Section 3.1 works a concrete CRC-error ticket end-to-end through the deployed pipeline; Appendix A.2 gives further examples, including the same incident in terse and complete forms.

3 Extraction Pipeline

Figure 1 shows the pipeline.

Telemetry attachment. At ticket creation, an Evidence Collector queries the affected device for interface state, performance counters (CRC errors, frame errors, discards), 72-hour link-flap history, and optical diagnostics (TX/RX power, bias current, temperature), over gNMI/gRPC for modern devices and SNMP for legacy ones behind a normalization layer. The telemetry is written into the ticket as a structured blob keyed by interface, before any human writes anything, so the evidence the technician would otherwise have to quote is already present.

Rule-based extraction. As the technician writes notes, a regex layer matches structured fragments. The patterns target the kinds of fragments experienced technicians write in conventional forms, such as “CRC: 450,” “RX power -3.2 dBm,” “2 flaps in 30 min,” and device identifiers in the operator’s standard format. Rules are deterministic, contribute high-precision matches at no LLM cost, and catch approximately 60% of the tokens that end up in Evidence. The five pattern classes — counter transitions, flap counts with windows, optical-power readings, test artifacts, and timestamps — are tabulated with sanitized examples in Table 4 (Appendix B.1). Every class targets ECO-Evidence or STAR-Result and none targets Situation or Task, which is why rules alone are weak (Section 4). Raw pattern syntax is written against device log formats

and withheld as vendor-identifying; the classes and their targets make the layer’s contribution visible.

LLM extraction. The free prose between matched fragments goes to a small commercial LLM with a structured-output prompt (temperature 0, 1-second timeout) that specifies the STAR and ECO schemas, gives three few-shot examples from the gold standard, and asks for JSON (Li et al., 2024; Agarwal et al., 2025); the rule-matched fragments are provided as context so the model integrates rather than re-extracts them. Output is validated against a JSON schema before acceptance (full prompt in Appendix B.2).

Confidence flagging. Each extracted field carries a confidence flag for three failure modes seen in early deployment: *partial telemetry*, where a loaded device returns success with unavailable values; *ambiguous claims*, which the rules tag Result and the LLM tags Opinion; and *out-of-distribution prose*. Fields with low-confidence flags are routed to the human review path rather than feeding downstream automation directly.

Routing. A small classifier (logistic regression on schema-completeness features) routes each ticket to one of three states. `VALID` means the record is good enough to act on. The ticket goes to the on-call engineer with the structured record attached. `NEEDS_EVIDENCE` means something is missing, so a checklist goes back to the technician naming the empty or ungrounded fields. `AUTO_CLOSE` is the narrow third case, used when the note carries only an opinion or a deferral and the telemetry shows nothing an engineer needs to see. It closes the ticket without an engineer reviewing it first, and fires on just 0.3% of tickets; those closures are still reviewed and audited afterwards (Section 4).

Materialization. STAR fields, ECO lists, confidence flags, and the classification outcome are written as columns in the operational database, keyed by ticket ID, making the parsed output queryable alongside the telemetry (worked SQL in Appendix B.4).

3.1 A Worked Ticket

One production incident, end to end, values paraphrased for vendor anonymity (see Ethical Considerations). Technician note as written: “Reseated cable at 14:05; CRC 12,450 to 0 within 10m; no flaps for 6h.” The attached telemetry corroborates each fragment (JSON in Appendix A.2). The rule layer matches four fragments (the CRC transition,

the ten-minute window, the timestamp, the flap window). STAR extracted. *S*: link errors on a leaf-spine port; *T*: restore a clean link; *A*: reseated cable at 14:05, verified counters; *R*: CRC to 0, stable six hours. ECO extracted. *E*: CRC 12,450 to 0 in 10 min, flap count 0 over 6 h; *C*: reseal resolved the issue; *O*: keep in service and monitor. Every STAR field is filled and the Claim is grounded in two Evidence items, so schema completeness is 1.0, all three evidence indicators fire ($\sigma = 1.0$, Section 4), and the router returns `VALID`. The same incident written tersely — “reseated optic, link looks fine” — fills Action only, yields one ungrounded Claim and no Evidence, and routes to `NEEDS_EVIDENCE` with a checklist naming the missing fields. That contrast is the task in one glance (further examples in Appendix A.2).

4 Evaluation

The pipeline is evaluated intrinsically against a human-coded gold standard and extrinsically against three operational outcomes, all on production data over the same eight-month window.

Blocked-ticket population. Operationally, a ticket is blocked when recovery is waiting on a required technician action following review by a site reliability engineer (SRE). For the evaluation reported here, we identify the blocked-ticket subset retrospectively using a fixed three-day criterion measured from the recorded queue-entry transition. The same criterion is applied in all comparison periods. Membership is determined from workflow metadata only; note content, STAR or ECO fields, the completeness score σ , and routing decisions cannot affect whether a ticket enters the evaluation subset. The three-day criterion is an evaluation heuristic rather than the operational definition of a blocked ticket.

Technicians act on every incident ticket and 97% reach the SRE queue once the work is done; of all incident tickets, 3.0% sat past three days in that queue before deployment and 1.5% during it (Table 7). Parts, dispatch, and maintenance windows are datacenter-side and resolve before queue entry, so queue residence isolates the coordination gap rather than the repair. One consequence of deployment works against us. Because the pipeline prevents straightforward cases from becoming blocked, the surviving blocked set post-deployment should skew harder, biasing the pre/post comparisons below conservatively. The

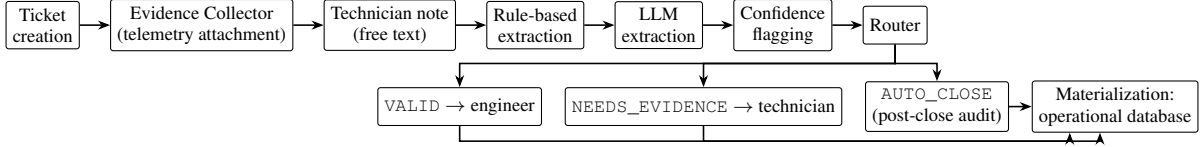


Figure 1: Extraction pipeline, with component names as used in the text. Telemetry attachment runs once at ticket creation; the rule-then-LLM extraction layers run on each note as it is written; the router’s three outcomes and the materialized record are described in Section 3.

Configuration	GPT-4-cl.	GPT-5.4-m.	GPT-5.5
Pure LLM	0.709	0.731	0.880
Hybrid	0.913	0.920	—

Table 1: Overall F1 by configuration and model. The hybrid was not run on GPT-5.5. The deployed pairing (hybrid, GPT-4-class, 0.913) is the configuration the extrinsic window was measured under.

blocked share itself falls over the window, from 3.0% to 1.5%, which is consistent with that reading.

Intrinsic: field-level extraction quality. We constructed a gold standard of 2,141 technician notes drawn from a stratified random sample over the eight-month deployment period. Two independent annotators with operations background labeled each note for STAR and ECO with Cohen’s $\kappa = 0.81$ overall (range $\kappa = 0.74$ for the ECO Claim/Opinion boundary to $\kappa = 0.89$ for STAR Action). Disagreements were resolved by a third annotator.

Overall F1 against the gold standard is 0.913 (precision 0.92, recall 0.89). Per-field F1 ranges from 0.86 on STAR-Task to 0.95 on STAR-Action. The Claim/Opinion boundary is the hardest for annotators, STAR Action the easiest. Appendices C.2 and C.1 give the per-field breakdown and the matching protocol (containment plus 70% token overlap, unit normalization, span-level Claim-grounding rules). Latency was 1.2 s median; uptime 99.97%. The 91.3% F1 reflects a GPT-4-class model; we keep it as the headline because the extrinsic results were measured under that configuration, with newer models in Table 1.

Ablation: models and configurations. Table 1 scores the configurations on the same gold standard under the Appendix C.1 protocol. Prompt, schemas, and few-shot examples are identical throughout; pure configurations omit only the rule-matched fragments.

Two readings follow. Holding the model fixed, the rule layer contributes 20.4 F1 points at GPT-

4-class and 18.9 at GPT-5.4-mini, so the design is not an artifact of a deprecated model; and upgrading moves the pure configuration by 2.2 points but the hybrid by only 0.7, as expected if most of a newer model’s headroom lies in fields the rules already fill. Even pure GPT-5.5 at 0.880 trails the deployed hybrid built on a weaker model. The pilot’s failure modes were invented STAR fields and paraphrased-away measurements, recognizable variants of known faithfulness failures (Maynez et al., 2020; Min et al., 2023; Tamber et al., 2025; Tam et al., 2024; Lim et al., 2025) and still present under pure GPT-5.5 (Appendix B.3). The rule layer suppresses them structurally. Fields filled from rule-matched fragments are copied from the source and cannot be hallucinated.

Rules only. A rules-only configuration scores 49% field-level F1, measured on a separate set of roughly 5,500 hand-labeled tickets from the six months preceding deployment, with no temporal overlap with the gold standard. That set is the corpus the patterns were developed against, so the figure measures fit to development data and is optimistic, not held out. The mechanism is visible from both sides. Rules populate Evidence and Result while Situation and Task stay empty, because fragments carry measurements, not diagnosis (Table 4); and the three fields the rule classes target rank first, second, and fourth of seven, while the three lowest-scoring fields are ones no rule class touches (Table 5). The two layers do not compete over the same fields.

Routing. A hybrid without confidence routing carries identical extraction F1 by construction, since flagging runs after extraction; its effect is on which records reach downstream automation. We therefore report it at the routing decision, as precision 0.669 and recall 0.653 of the low-confidence flag against gold field correctness, evaluated over all 14,987 extracted fields. Two thirds of flagged fields are genuinely incorrect and the flag catches two thirds of the errors present, which is a usable but not a strong operating point; it quantifies rather

than closes the calibration gap noted in our Limitations. We cannot isolate routing’s contribution to the extrinsic outcomes experimentally and do not imply otherwise.

Open-weight models. Qwen2.5 and Llama were assessed informally during development and neither reached deployable quality on this task, so we did not carry them forward and have no scored numbers to report.

Extrinsic: three operational outcomes. *Speed.* Median time-to-repair fell from 7.0 days pre-deployment to 3.5 days, a 50% reduction. Tickets that closed cleanly are outside this measurement and were not the target. *Quality of closure.* The 30-day reopen rate, a new incident against the same device or link within 30 days of closure, fell from 25% pre-deployment to approximately 5.5% over the last three months of the window. The joint speed-and-reopen reduction indicates tickets closing correctly on first attempt, not merely faster. *Coordination cost.* A round-trip is one cycle of engineer-asks-then-technician-responds on the ticket. The mean fell from 6.0 to 2.5 round-trips per ticket. Engineer review still happens on every ticket; structured records simply arrive complete enough to need fewer rounds. That is the same mechanism behind the reduction in time-to-repair (MTTR).

Causal identification: staggered-rollout analysis. The pre/post comparison above does not by itself separate the deployment effect from fleet-wide temporal changes, so we exploit the phased rollout across more than 450 datacenters, which proceeded in four waves with cumulative treated footprints of 5, 40, 80, and finally the full footprint (Appendix E). Until the final wave, datacenters that had not yet received the pipeline provide contemporaneous controls. At any point during the rollout we can compare sites that already have it against sites that do not, and both are living through the same fleet-wide conditions. We estimate group-time average treatment effects (ATTs) (Callaway and Sant’Anna, 2021) on that comparison group at two-week granularity, giving one estimate per rollout wave per period rather than a single number. We do this because a standard two-way fixed-effects estimator is biased under staggered adoption, comparing later-treated sites against already-treated ones (Goodman-Bacon, 2021). The comparison is only valid if treated and untreated sites were on similar trajectories beforehand. That is what the pre-deployment coefficients test. If they

Outcome	Group-time ATT	95% CI
MTTR / recovery time	−3.41 days	[−3.84, −2.98]
30-day reopen rate	−19.1 pp	[−21.8, −16.4]
Round-trips per ticket	−3.45 cycles	[−3.82, −3.08]

Table 2: Group-time average treatment effects from the staggered rollout. Not-yet-treated datacenters form the comparison group. Effects are absolute changes: days for MTTR, percentage points for the reopen rate, and round-trip cycles per ticket. Intervals are pointwise 95% confidence intervals.

sit near zero, the two groups were not already diverging before the pipeline arrived. The treatment is the deployed package, not the language model in isolation. Training, the site guide, and the rollout process changes shipped with the waves. Table 2 reports the aggregated estimates for the three outcomes used throughout this section, with time to recovery measured through return of the affected link or device to production; Appendix E gives full outcome definitions.

Appendix E reports the event-time coefficients individually, following Roth et al. (2023). The ten pre-deployment coefficients are centered near zero, none exceeds 0.22 days in absolute value against standard errors of 0.19 to 0.58, and every simultaneous band contains zero; they provide no evidence of a differential trend before rollout. The three aggregated estimates are close in magnitude to the pre/post differences reported above (−3.5 days, −19.5 percentage points, and −3.5 cycles), each of which falls inside the corresponding interval, so the staggered design corroborates rather than revises the direction and scale of the pre/post comparison. We do not read the agreement more finely than that. The estimator operates on datacenter-period means, whereas the pre/post MTTR figure is a global median over raw tickets, so the two are not computed on identical summaries. Post-deployment estimates vary across event time as the rollout meets different incident mixes and pre-existing ticket inventories, so Table 2 is the primary summary.

Composition and concurrent change. Category shares on the blocked subpopulation move by at most one percentage point between the two windows, and none moves in a direction that would make the deployment window easier (Table 7). Volume floods from datacenter-wide events and ticketing-system faults are filtered in both windows by a filter that predates the deployment. One

component of the package deserves naming. The materialized records also feed a service that suggests runbooks, the operator’s standard repair procedures, matching roughly one blocked note in ten. It ran throughout the window, is part of what the wave estimate measures, and is not ablated here. Two further threats, the seven prompt revisions and technician-side staffing, are stated in Limitations.

Operator context: customer-allocatable capacity. Every open blocked ticket holds a slice of capacity off the placement pool, so ticket-days held off-pool fall mechanically when MTTR falls. Customer-allocatable capacity did rise over the window, but many programs move it and we have not isolated this system’s share, so we do not report it as an outcome.

The extraction-completeness score σ . We define a completeness score $\sigma(T) \in [0, 1]$ per ticket. It combines how much of the schema is filled (the fraction of STAR fields present) with how good the evidence is (quantitative content, temporal coverage, and whether ECO Claims are grounded in Evidence). The two are weighted $w_s = 0.4$ and $w_e = 0.6$, set by a logistic regression predicting first-pass closure on a held-out 1,500-ticket set. Evidence outweighs schema because it predicts closure more strongly (odds ratios 8.2 against 3.1); a filled-in but unsupported record does not let the engineer act. The full definition is in Appendix A.3.

Empirical model and cross-category stability. Plotting ticket resolution time τ against σ over the eight-month deployment, we fit

$$\tau(\sigma) = \alpha e^{-\beta\sigma} + \gamma \quad (1)$$

with $\alpha = 4.3$ days, $\beta = 1.64$, $\gamma = 2.8$ days, and $R^2 = 0.94$. A ticket-level bootstrap gives a 95% interval for the pooled exponent of $[1.58, 1.72]$. The procedure and its one caveat, that non-converged replicates were replaced by further draws, are given in Appendix C.4. Figure 2 shows the pooled fit with its bootstrap band, alongside the five per-category fits.

The exponent admits a per-category test. If β is governed by the failure mode it should differ across physically dissimilar categories; if by the technician–engineer channel, it should be approximately constant. Table 3 reports Equation 1 refit on each of five categories.

Every per-category interval in Table 3 contains the pooled estimate, which is the evidence for stability we rely on. The intervals also differ in width

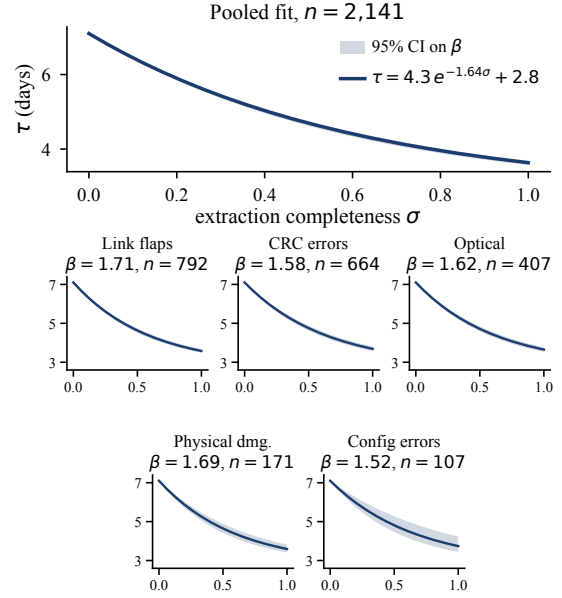


Figure 2: Fitted relationship between resolution time τ and extraction completeness σ : the pooled fit of Equation 1 with the bootstrap interval on β (top), and the five per-category refits with their own intervals (panels). Curves are drawn from the fitted parameters of Table 3; the underlying per-ticket scatter is not shown, as the ticket-level data are proprietary.

by a factor of three. Configuration errors span 0.80 at $n = 107$ against 0.25 for link flaps at $n = 792$. A coefficient of variation would treat the five point estimates as equally precise, so we do not report one.

$R^2 = 0.94$ does not by itself discriminate the exponential from linear or power-law alternatives, and the interpretation anchors on the exponential specifically. We fit the alternatives on the same data and compare by ΔAICc (Burnham and Anderson, 2002), a model-selection criterion that scores fit while penalizing extra parameters; a gap below about two counts as comparable support. The linear form is decisively worse ($\Delta\text{AICc} = 2443.2$), but the power law is not, with $\Delta\text{AICc} = 1.4$ and an Akaike weight of 0.33 against 0.67 (Table 6). The data therefore do not single out the exponential. We therefore treat Equation 1 as a descriptive parameterization rather than an empirically identified functional law. Because the comparison no longer singles out the exponential, the information-theoretic reading that motivates it is given in Appendix C.4 rather than developed here. The substantive finding — that completeness predicts resolution time consistently across physically distinct failure types

Incident type	β	95% CI	R^2	n
Link flaps	1.71	1.59–1.84	0.92	792
CRC errors	1.58	1.45–1.72	0.93	664
Optical issues	1.62	1.49–1.76	0.91	407
Physical damage	1.69	1.44–1.95	0.89	171
Configuration errors	1.52	1.10–1.90	0.87	107
Pooled	1.64	1.58–1.72	0.94	2,141

Table 3: Per-category fits of Equation 1 with bootstrapped 95% intervals. Every per-category interval contains the pooled estimate; interval overlap, not a coefficient of variation over point estimates of unequal precision, is the stability claim.

— is unaffected.

Auto-close audit. The `AUTO_CLOSE` path closes a ticket without pre-close engineer review; over the window it fired on approximately 0.3% of tickets, about 46 closures, under a deliberately low threshold. Two controls cover it after the fact. Every auto-closure is reported to the weekly operational review, where any case can be opened for discussion or escalation; across the window no case was identified as one that should have been closed differently. A 5% sample additionally receives end-to-end review with the datacenter technician, and none was overturned. Stated plainly, the scale is 46 closures, of which the sampled tier covers roughly two. This is a coverage statement and a record of what the reviews surfaced, not a measured error rate, and we report it as such. The pipeline is rated Sev3 on failure, meaning degraded internal tooling and no customer impact, and pre-existing runbooks still govern triage when it is unavailable (Appendix D). In month 10, after the reporting window, the model was upgraded and the threshold progressively raised on audit experience; all numbers here reflect the original regime.

5 Related Work

Log parsing (He et al., 2017; Du and Li, 2016) and outage-report summarization (Chen et al., 2019) extract structure from machine-generated logs or post-hoc documents. Real-time technician notes break those assumptions. They are short, terse, written by different people across shifts, and concurrent with the work.

Hybrid rules-plus-LLM pipelines are common in production NLP. A parallel literature aligns LLMs for extraction at training time (Qi et al., 2024; Sainz et al., 2024; Guo et al., 2024), improves structured output through constrained decoding (Li

et al., 2024; Chan et al., 2024; Agarwal et al., 2025), and builds schema-driven extraction for adjacent modalities (Bai et al., 2024). Our pipeline follows that recipe; what we add is deployment evidence at scale on technician notes.

STAR (Development Dimensions International, n.d.) and ECO (adapted from Toulmin, 1958) come from structured-interview methodology and argumentation theory respectively. ECO’s claim/evidence distinction connects to computational argument mining (Li et al., 2025), which targets debates and editorials; using these categories as runtime extraction targets for operational text is, to our knowledge, new. The nearest human-protocol analogue is the I-PASS handoff bundle in medicine (Starmer et al., 2014), whose staggered multi-hospital deployment reduced handoff errors. I-PASS assumes a person fills the schema; STAR and ECO are recovered from free text at scale.

Automated diagnosis systems (Kandula et al., 2009; Isard, 2007; Ferguson et al., 2021; Tan et al., 2019) reason about the technical system. Recent work points LLMs at incident management directly: root-cause recommendation (Ahmed et al., 2023), root-cause analysis (Chen et al., 2024; Roy et al., 2024), and benchmarking root-cause location (Xu et al., 2025). Our pipeline is complementary. It does not diagnose; it extracts what the human diagnostician wrote, for those systems and for the engineer.

6 Conclusion

The connective layer of the title is a brokered hand-off. The system moves the account of the work from the person who did it to the person who must act on it, and writes nothing on either person’s behalf. We measured one crossing, technician to engineer, in one organization. Tier one to tier two, shift to shift, and team to customer have the same structure, but we have not tested there the claim that would generalize this result — that the cost of a transfer is set by the sufficiency of the account.

Three questions follow. Whether β is a property of this workforce or of asynchronous handoff in general; deciding this requires a second organization, not a second dataset. Whether σ can be learned rather than specified, given that ours was fit to predict the outcome it is later used to explain. And whether the corpus the deployment produces as a by-product contains failure patterns that no single ticket exhibits (Dean and Barroso, 2013).

Limitations

The results are not independently verifiable.

The 2,141-note gold standard is production technician notes and cannot leave the operator; the scoring and matching implementations are not released either. Only the prompt skeleton of Appendix B.2 is. A reader therefore cannot recompute the extraction F1, the per-field scores, the flag precision and recall, or σ itself, and the extrinsic results were never recomputable from public material. What we can offer instead is specification. Appendix C.1 defines the matching protocol, Appendix A.3 gives σ in full, and Table 4 lists the rule classes and their targets, each in enough detail to reimplement. That is weaker than a release and we do not present it as equivalent.

Single organization. The empirical results are from one cloud provider and one technician/engineer workforce. The MTTR, reopen-rate, and round-trip results all reflect this single deployment. The exponent β in Equation 1 is stable within our deployment but may differ in other deployments. We are not in a position to claim cross-organization generality from a single-organization study.

Observational σ -versus- τ relationship. The relationship $\tau(\sigma) = \alpha e^{-\beta\sigma} + \gamma$ is observational. High- σ tickets may also have easier underlying problems, and σ 's weights were fit to predict first-pass closure, so the score is not constructed independently of the outcome it explains. The cross-category stability of β (Table 3) and the system-level reopen-rate and round-trip results partially mitigate but do not eliminate this confound. A controlled experiment would require running the deployed system on a held-back subset of tickets, which we have not done.

English-only gold standard. The gold standard is in English. F1 on non-English notes is approximately 87%, compared with 91.3% on English. Multilingual support is in progress.

Model and threshold change after the reporting window. The intrinsic F1, the MTTR, the reopen rate, the round-trip count, and the per-category β fits all reflect the GPT-4-class model and the 0.3%-auto-close threshold that were in production during the eight-month window the paper reports. In month 10, the model was upgraded and the auto-close threshold was raised. Rescoring of the gold standard under the replacement model is reported

in Table 1 (hybrid 0.920, pure 0.731); the extrinsic results remain paired with the 0.913 configuration under which they were measured. The production service is now migrating again, to GPT-5.6 Luna, under vendor-driven API deprecation. A pure-LLM F1 of approximately 0.88 on this task has been reported to us by a separate team; we have not rescored the gold standard under that model ourselves and cannot confirm the figure was produced under the matching protocol of Appendix C.1, so it is not comparable to the configurations in Table 1. That transition postdates the reporting window and no operational result in this paper is measured under it.

Information-theoretic interpretation is suggestive.

The information-theoretic argument in Appendix C.4 motivates the exponential form of Equation 1 under specific assumptions, principally linearity of I_0 in σ . The exponential fit on our data is consistent with these assumptions but does not prove them, and the subsection states this at its head. Because a power-law parameterization fits comparably under ΔAICc (Section 4), the exponential form is a descriptive choice rather than one the data single out. We do not present the argument as a derivation.

Gold-standard size. The 2,141-note gold standard is large by the standards of operational annotation but small by the standards of typical NLP benchmarks. The result that β is approximately category-independent is established on five categories of network incidents only. Other operational domains (storage, database operations, application-layer incidents) may behave differently.

Causal identification of extrinsic improvements.

The headline extrinsic results are pre/post comparisons on the same blocked-ticket subpopulation. Section 4 therefore estimates group-time average treatment effects over the staggered rollout, with not-yet-treated datacenters as controls. Three threats survive that design. A change landing on every region on the same date is not identified. The prompt's seven revisions partially confound prompt maturity with wave order. And technician-side staffing changes correlated with rollout order are outside our visibility. The estimand throughout is the deployed package, not the model in isolation.

Rules-only baseline is optimistic; non-LLM baselines are absent.

The 49% rules-only figure (Section 4) is measured on the corpus the

patterns were developed against and is an upper bound on that configuration; no notes overlap the gold standard. We report no classical information-extraction baseline (CRF taggers, constrained semantic parsers); a clean one is left to future work.

Confidence-flag calibration. The confidence flag is operationalized through the three-way routing decision, and we report it at that decision as precision/recall of the low-confidence flag against gold field correctness (Section 4). Per-field calibration (e.g., expected calibration error) is future work, and routing’s contribution to the extrinsic outcomes is not experimentally isolated.

Ethical Considerations

This study was conducted under operational safeguards approved by an internal review process. All ticket data were anonymized before analysis. Device identifiers were replaced with generic labels. Technician and engineer identifiers were hashed. Timestamps were shifted by random offsets. No personal or performance-related information was accessed or stored. All examples in this paper are paraphrased or synthetic.

Operational safety was prioritized throughout deployment. Automated actions were limited to evidence collection, classification, and the auto-close path of Section 4. No repair actions were executed by the system. Most closure paths require engineer review before closure; the narrowly scoped auto-close path (approximately 0.3% of tickets in the reporting window) may close a ticket without pre-close engineer approval when predefined evidence conditions are satisfied, subject to the post-close review and audit described in Section 4. All other closure decisions rested with human engineers. A phased rollout, opt-in participation, and continuous auditing ensured that no production incident resolution was compromised.

Technicians and engineers were informed participants in system deployment. Feedback was solicited regularly. Training ensured transparency of system capabilities and limitations. Stakeholder committees, including technician representatives, approved the design prior to global rollout. STAR/ECO outputs and the extraction-completeness score σ are not used for technician performance evaluation. This commitment was given in writing at the start of the deployment and has been maintained.

Large language models are used in two distinct

ways in this work. First, within the production pipeline, LLMs perform structured extraction of STAR and ECO components from technician notes. All LLM outputs are validated against rule-based checks, and all closure decisions other than the auto-close path above are reviewed by an engineer before closure. Final decisions rest with human engineers. Second, during manuscript preparation, LLM-assisted tools were used for grammar refinement and organization of text. All scientific claims, experimental design, implementation details, and results were developed and verified by the authors.

Acknowledgements

We thank the DRIs of the UDL team for their feedback throughout the deployment. We thank the late Percy Fitter for his suggestions and for asking the hard questions. We thank Sandeep Chaudhary for his help in productizing the code, and Abhishek Gandhi for the initial analysis of the incident-ticket corpus.

References

- Bhavik Agarwal, Ishan Joshi, and Viktoria Rojkova. 2025. Think inside the JSON: Reinforcement strategy for strict LLM schema adherence. arXiv preprint arXiv:2502.14905.
- Toufique Ahmed, Supriyo Ghosh, Chetan Bansal, Thomas Zimmermann, Xuchao Zhang, and Saravan Rajmohan. 2023. Recommending root-cause and mitigation steps for cloud incidents using large language models. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 1737–1749. IEEE.
- Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, Mark Dredze, and Alan Ritter. 2024. Schema-driven information extraction from heterogeneous tables. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10252–10273, Miami, Florida, USA. Association for Computational Linguistics.
- Kenneth P. Burnham and David R. Anderson. 2002. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd edition. Springer, New York.
- Brantly Callaway and Pedro H. C. Sant’Anna. 2021. Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2):200–230.
- Robin Chan, Katsiaryna Mirylenka, Thomas Gschwind, Christoph Miksovics, Paolo Scotton, Enrico Toniato, and Abdel Labbi. 2024. Adapting LLMs for structured natural language API integration. In *Proceedings of the 2024 Conference on Empirical Methods in*

- Natural Language Processing: Industry Track*, pages 991–1000, Miami, Florida, US. Association for Computational Linguistics.
- Yinfang Chen, Huaibing Xie, Minghua Ma, Yu Kang, Xin Gao, Liu Shi, Yunjie Cao, Xuedong Gao, Hao Fan, Ming Wen, and 1 others. 2024. Automatic root cause analysis via large language models for cloud incidents. In *Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys)*, pages 674–688.
- Yujun Chen, Xian Yang, Qingwei Lin, Hongyu Zhang, Feng Gao, Zhangwei Xu, Yingnong Dang, Dongmei Zhang, Hang Dong, Yong Xu, Hao Li, and Yu Kang. 2019. Outage prediction and diagnosis for cloud service systems. In *The World Wide Web Conference (WWW '19)*, pages 2659–2665. Association for Computing Machinery.
- Jeffrey Dean and Luiz André Barroso. 2013. The tail at scale. *Communications of the ACM*, 56(2):74–80.
- Development Dimensions International. n.d. STAR method for interviewing and feedback. <https://www.ddi.com/solutions/behavioral-interviewing/star-method>.
- Min Du and Feifei Li. 2016. Spell: Streaming parsing of system event logs. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 859–864. IEEE.
- Andrew D. Ferguson, Steve Gribble, Chi-Yao Hong, Charles Killian, Waqar Mohsin, Henrik Muehe, Joon Ong, Leon Poutievski, Arjun Singh, Lorenzo Vicisano, Richard Alimi, Shawn Shuoshuo Chen, Mike Conley, Subhasree Mandal, Karthik Nagaraj, Kondapa Naidu Bollineni, Amr Sabaa, Shidong Zhang, Min Zhu, and Amin Vahdat. 2021. Orion: Google’s Software-Defined networking control plane. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 83–98. USENIX Association.
- Andrew Goodman-Bacon. 2021. Difference-in-differences with variation in treatment timing. *Journal of Econometrics*, 225(2):254–277.
- Biyang Guo, He Wang, Wenyilin Xiao, Hong Chen, ZhuXin Lee, Songqiao Han, and Hailiang Huang. 2024. Sample design engineering: An empirical study on designing better fine-tuning samples for information extraction with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 573–594, Miami, Florida, US. Association for Computational Linguistics.
- Pinjia He, Jieming Zhu, Zibin Zheng, and Michael R. Lyu. 2017. Drain: An online log parsing approach with fixed depth tree. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 33–40. IEEE.
- Michael Isard. 2007. Autopilot: Automatic data center management. *ACM SIGOPS Operating Systems Review*, 41(2):60–67.
- Srikanth Kandula, Ratul Mahajan, Patrick Verkaik, Sharad Agarwal, Jitendra Padhye, and Paramvir Bahl. 2009. Detailed diagnosis in enterprise networks. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, pages 243–254. Association for Computing Machinery.
- Hao Li, Viktor Schlegel, Yizheng Sun, Riza Batista-Navarro, and Goran Nenadic. 2025. Large language models in argument mining: A survey. *arXiv preprint arXiv:2506.16383*.
- Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024. A simple but effective approach to improve structured language model output for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5133–5148, Miami, Florida, USA. Association for Computational Linguistics.
- Seungseop Lim, Gibaeg Kim, Wooseok Han, Jean Seo, Hyunkyung Lee, Jaehyo Yoo, and Eunho Yang. 2025. Format inertia: A failure mechanism of LLMs in medical pre-consultation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1437–1450, Suzhou, China. Association for Computational Linguistics.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Sewon Min, Kalpesh Krishna, Xinxin Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Yunjia Qi, Hao Peng, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2024. ADELIE: Aligning large language models on information extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Miami, Florida, USA. Association for Computational Linguistics.
- Jonathan Roth, Pedro H. C. Sant’Anna, Alyssa Bilinski, and John Poe. 2023. What’s trending in difference-in-differences? A synthesis of the recent econometrics literature. *Journal of Econometrics*, 235(2):2218–2244.

Devjeet Roy, Xuchao Zhang, Rashi Bhawe, Chetan Bansal, Pedro Las-Casas, Rodrigo Fonseca, and Saravan Rajmohan. 2024. Exploring LLM-based agents for root cause analysis. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, pages 208–219.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Amy J. Starmer, Nancy D. Spector, Rajendu Srivastava, Daniel C. West, Glenn Rosenbluth, April D. Allen, Elizabeth L. Noble, and 1 others. 2014. Changes in medical errors after implementation of a hand-off program. *New England Journal of Medicine*, 371(19):1803–1812.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? A study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.

Manveer Singh Tamber, Forrest Sheng Bao, Chenyu Xu, Ge Luo, Suleman Kazi, Minseok Bae, Miaoran Li, Ofer Mendelevitch, Renyi Qu, and Jimmy Lin. 2025. Benchmarking LLM faithfulness in RAG with evolving leaderboards. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 799–811, Suzhou, China. Association for Computational Linguistics.

Cheng Tan, Ze Jin, Chuanxiong Guo, Tianrong Zhang, Haitao Wu, Karl Deng, Dongming Bi, and Dong Xiang. 2019. NetBouncer: Active device and link failure localization in data center networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 599–614. USENIX Association.

Stephen E. Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Junjielong Xu, Qinan Zhang, Zhiqing Zhong, Shilin He, Chaoyun Zhang, Qingwei Lin, Dan Pei, Pinjia He, Dongmei Zhang, and Qi Zhang. 2025. OpenRCA: Can large language models locate the root cause of software failures? In *The Thirteenth International Conference on Learning Representations (ICLR)*.

A Background and Definitions

A.1 Background for Non-Networking Readers

This subsection introduces the physical and organizational setting for readers without a networking background.

The physical setting. A hyperscale operator runs a network with hundreds of thousands of switches connected by optical fiber. A common topology is a Clos fabric organized in layers. Top-of-rack switches aggregate the servers in one rack. Leaf switches aggregate top-of-rack switches. Spine switches aggregate leaves. When a physical link between two switches degrades, the operator wants to detect the problem, repair it, and confirm the repair quickly, because every degraded link is capacity removed from customer-facing service.

What “degraded” means. A network link degrades when its physical layer starts corrupting the frames it carries. The standard signal is the Cyclic Redundancy Check (CRC) error counter on the affected port. CRC errors are detected by hardware on the receiving side and incremented on every corrupted frame. Rising CRC errors are a leading indicator that the link’s physical layer needs intervention. The most common hardware cause is an aging or marginal optical transceiver, the small pluggable fiber-optic endpoint at each switch port. The standard fix is to replace the transceiver. Whether the fix held is verifiable from post-intervention telemetry. The CRC counter stops incrementing. Optical transmit and receive power return to nominal ranges (measured in dBm). The link does not flap, that is, it does not rapidly oscillate between up and down states.

Why customer capacity is at stake. Customer workloads in Azure are placed by an automated control plane onto the operator’s network and compute fleet. When an incident ticket is open against a link, the control plane treats that link’s capacity as derated. It stops placing fresh workload on that link. The link does not count toward the redundancy guarantees the operator sells to customers. Capacity is restored to the placement pool only when the ticket is closed. Customer-paid capacity that is derated against an open ticket is capacity the operator is holding off-pool for non-hardware reasons. This is the operational cost the system in this paper targets.

The human workflow. Two roles handle each incident. A technician is physically present at the datacenter and performs the repair. An on-call engineer, or SRE, is remote and decides whether the ticket can be closed based on the technician’s note and the post-intervention telemetry. The technician and the engineer are typically in different

locations and on different shifts. Their primary medium of coordination is the ticket itself, which is asynchronous. Each side reads and updates it on their own schedule. Synchronous coordination over phone or chat is rare and reserved for severe incidents (denoted Sev1 or Sev2 in the operator’s severity scale). Time-to-repair (MTTR) in this paper is therefore the wall-clock time from ticket open to ticket close, including the asynchronous back-and-forth between the technician and the engineer; we report medians unless stated otherwise. Closure returns the affected capacity to the placement pool, so for placement purposes it coincides with the link’s return to production. Data-collection protocols used by the system are gRPC Network Management Interface (gNMI) over gRPC for modern switches and Simple Network Management Protocol (SNMP) for older switches.

A.2 Worked Example: A Note Through STAR and ECO

The note “reseated optic, link looks fine” parses as follows. STAR has Action = “reseated optic” and the other three fields empty. ECO has one Claim (the link is stable) and no Evidence. The Claim is therefore ungrounded.

A complete note for the same incident parses as Situation = “CRC errors rising on leaf-04 port Et1/15,” Task = “determine source,” Action = “replaced optic; cleaned LC connector,” Result = “CRC held at 0 for 30 min post-swap; RX power −3.2 dBm to −1.8 dBm.” The corresponding ECO record has three Evidence items drawn from the Action and Result fields, one grounded Claim, and no Opinions.

Section 3.1 carries a full pipeline trace for a CRC incident. The telemetry attached at ticket creation for that ticket:

```
{ "interface": "sw-123/1/45",
  "port_state": { "admin": "up",
    "oper": "up", "speed_gbps": 100 },
  "crc_counters": { "before": 12450,
    "after": 0, "window_minutes": 10 },
  "flap_history": { "count": 0,
    "window_hours": 6 },
  "timestamp": "2025-08-18T14:15:00Z" }
```

A.3 Full Definition of the Extraction-Completeness Score σ

Section 4 introduces $\sigma(T) \in [0, 1]$ for each ticket T . The full definition is:

$$\sigma(T) = w_s \cdot \sigma_{\text{schema}}(T) + w_e \cdot \sigma_{\text{evidence}}(T) \quad (2)$$

with $w_s = 0.4$ and $w_e = 0.6$. The schema component is the fraction of STAR fields the pipeline filled:

$$\sigma_{\text{schema}}(T) = \frac{s_{\text{sit}} + s_{\text{task}} + s_{\text{act}} + s_{\text{res}}}{4} \quad (3)$$

where each s is 1 if the corresponding STAR field is non-empty and 0 otherwise. The evidence component is the fraction of three indicators satisfied:

$$\sigma_{\text{evidence}}(T) = \frac{e_{\text{quant}} + e_{\text{temp}} + e_{\text{ground}}}{3} \quad (4)$$

$e_{\text{quant}} = 1$ if the note contains a numeric value with a unit, $e_{\text{temp}} = 1$ if the note contains both pre-intervention and post-intervention observations, and $e_{\text{ground}} = 1$ if every ECO Claim has at least one supporting Evidence item.

B Pipeline Details

B.1 Rule Pattern Classes

Class	Sanitized example	Target(s)
Counter transition	CRC 12,450 → 0	ECO-E, STAR-R
Flap count + window	2 flaps in 30 min	ECO-E
Optical power	RX power −3.2 dBm	ECO-E
Test artifact	cable test pass	ECO-E
Timestamp	14:05	STAR-A, STAR-R

Table 4: Regex pattern classes, sanitized. E = Evidence, R = Result, A = Action. Every class targets Evidence or Result; none targets Situation or Task.

B.2 Extraction Prompts

The LLM extraction step uses a single prompt with the following structure.

1. An instruction block specifying the STAR and ECO schemas with one-sentence definitions of each field and constraints.
2. A JSON output schema with required and optional fields.
3. Three few-shot examples drawn from the gold standard, covering a CRC-error note, an optical-issue note, and a configuration-error note.
4. The input note plus any rule-matched fragments as context.

The deployed template runs to approximately 800 tokens before the input. What we release is the skeleton below, which carries the instruction structure without the few-shot examples or the field-level constraint text; those are drawn from and

describe production notes, so they are withheld for the same disclosure reason as the raw pattern syntax.

```
You are a strict evidence extractor for technician-  
to-engineer handoffs.  
  
Input note:  
<<<  
{technician_text}  
>>>  
  
For each note:  
1) Extract STAR (S, T, A, R)  
2) Extract ECO (E, C, O)  
3) Identify HARD_EVIDENCE:  
   - crc_counters: before, after, window  
   - flap_history: count, window  
   - test_result: name, pass|fail|unknown  
4) Classify: VALID | NEEDS_EVIDENCE | AUTO_CLOSE  
5) Assign confidence: CERTAIN | PROBABLE | AMBIGUOUS  
   | UNRELIABLE  
6) Give a one-line justification for the confidence  
   label  
7) Return a single valid JSON array
```

The four confidence labels are collapsed to the binary low-confidence flag of Section 3. AMBIGUOUS and UNRELIABLE raise the flag; the other two do not. The released skeleton specifies the task; it does not permit recomputation of the reported scores, which require the gold standard and the scoring implementation.

Prompt revisions are sourced from the 5% audit cycle. When the audit flags a class of errors, a few-shot example covering that class is added to the prompt, and the prompt is re-validated against the gold standard. Across the eight months of deployment, the prompt has undergone seven revisions. Each revision was tested for regression on the existing gold standard before promotion.

B.3 Pilot Study Motivating the Hybrid Design

Early in the project, we ran an informal pilot of approximately 500 production tickets through an LLM-only configuration of the pipeline. The configuration was a GPT-4-class model with a structured-output prompt close to the production version, but with the rule layer and the confidence flagging disabled. The pilot was qualitative and was conducted before the gold standard was built, so we do not report quantitative numbers.

The outputs were syntactically valid JSON with the requested STAR and ECO fields present. Two failure modes appeared often enough that we concluded the LLM-only configuration was not deployable as it stood.

First, given a terse note, the model confidently filled all four STAR fields with plausible content that was not in the note. A Situation, Task, Action, and Result were produced that read like a

competent ticket record but contained details the technician had not written.

Second, when the note contained specific measurements such as “CRC: 450 in 30 min”, the model paraphrased them into qualitative summaries like “CRC errors decreased after the intervention” in the output. The values an engineer would need to verify the fix from telemetry were lost.

Both failure modes are recognizable variants of behavior reported in the literature on abstractive summarization and faithfulness. Off-the-shelf LLMs tend to default to abstractive, plausible-completion behavior on terse technical input rather than to extraction. The pilot motivated the rule layer, the confidence flagging, and the schema validation step in the deployed pipeline. The pilot itself does not constitute a controlled baseline against the deployed system. The controlled comparison is in Section 4, where a pure-LLM configuration with GPT-5.5 on the full gold standard achieves 0.880 F1 against the deployed hybrid’s 0.913 and exhibits the same qualitative failure modes at lower rate.

B.4 SQL Examples

The materialized extraction makes analytical queries over notes implementable as SQL. Three examples follow.

Tickets where claims are not supported by post-intervention telemetry.

```
SELECT n.ticket_id, n.device_id,  
        n.eco_claim, t.crc_delta, t.  
        flap_count  
FROM    Notes n  
JOIN    Telemetry t ON t.ticket_id = n.  
        ticket_id  
WHERE   n.eco_claim IS NOT NULL  
AND    n.eco_evidence IS NULL  
AND    t.crc_delta_24h <= 0  
AND    n.classification = 'VALID';
```

Run weekly, this returned 147 tickets in which the technician asserted resolution but post-intervention CRC counts had not improved. Each was a candidate for engineer follow-up that would otherwise have closed silently.

Root-cause categories with the highest rate of missing Result fields.

```
SELECT root_cause, COUNT(*) AS n,  
        AVG(CASE WHEN star_result IS NULL  
                THEN 1 ELSE 0 END)  
        AS missing_result_rate  
FROM    Notes  
WHERE   star_action IS NOT NULL  
GROUP   BY root_cause  
ORDER   BY missing_result_rate DESC;
```

This identified link-flap tickets as having the highest rate of missing post-intervention flap counts, which fed back into the technician checklist for that root-cause category.

Per-region σ versus median MTTR.

```
SELECT region,
       AVG(sigma) AS mean_sigma,
       PERCENTILE_CONT(0.5) WITHIN GROUP
         (ORDER BY mttr_days) AS
         median_mttr
FROM   Tickets
WHERE  closed_at BETWEEN '2025-04-01'
              AND '2025-06-30'
GROUP BY region;
```

Regions with mean σ below 0.6 aligned with regions whose median MTTR was above the deployment-wide value. We used this output to target additional technician training.

C Evaluation Details

C.1 Field-Level Matching Protocol

The F1 numbers reported in Section 4 and Appendix C.2 are computed against a deterministic matching protocol. We document it here for reproducibility.

STAR text-valued fields. For each of the four STAR fields, both the gold standard and the pipeline produce either a span of text or an empty value. A pipeline span is counted as a match against the gold span if both are non-empty and either (i) the pipeline span lies within the gold span or vice versa (containment), or (ii) the two spans share at least 70% of their tokens after lower-casing, removing punctuation, and applying the unit normalization described below. Both empty counts as a match. Empty against non-empty (in either direction) counts as a miss.

Unit normalization. Numeric values are normalized before matching. Percentage forms (“15%”, “15 percent”, “approximately 15%”) collapse to a single canonical form; dBm values are rounded to one decimal place; CRC counts are matched as integers; time spans are converted to minutes (“30 min”, “half hour”, “0.5 hr” all map to 30). The normalization layer is the same one the deployed pipeline uses when populating the operational database, so the scoring is consistent with the downstream consumer.

ECO Evidence and Opinion lists. Each is a list of items. The match is computed at the item level, with two items counted as a match under the same

containment-or-70%-token-overlap rule used for STAR. Precision is items in the pipeline output that match a gold item; recall is items in the gold output that match a pipeline item; F1 is the harmonic mean.

ECO Claim grounding. A Claim is grounded if and only if at least one Evidence item in the same record satisfies a span-level linking rule: the Claim and the Evidence item share at least one normalized numeric value, or share a head-noun referring to the same telemetry signal (CRC, RX, flap count). The grounding flag is binary per Claim. The pipeline-vs-gold comparison at the Claim level reports both the Claim-extraction F1 (was the Claim captured at all) and the grounding-correctness rate (given that the Claim was captured, was its grounding flag set correctly). The 0.87 F1 reported for ECO-Claim in Table 5 is the Claim-extraction F1. Grounding accuracy on correctly-extracted Claims is 0.91, reported here rather than in the main table to avoid a metric mixed across two different decisions.

What this protocol does and does not cover. The protocol captures exact and near-exact matches with a 70% token-overlap tolerance for paraphrase. It does not use embedding similarity or learned matchers. The choice is deliberate. The deployed system materializes the extracted fields into a relational database that downstream consumers query with exact-match SQL, so the F1 protocol must match the level of agreement the downstream query consumer would observe. Inter-annotator agreement on the gold standard (Cohen’s $\kappa = 0.81$ overall) was computed using the same protocol.

C.2 Full Per-Field F1 Breakdown

Table 5 reports precision, recall, and F1 against the 2,141-note gold standard, broken down by field. Section 4 summarizes the overall and per-field range.

C.3 Extended Failure Modes on the Gold Standard

The most common failure modes of the pipeline on the gold standard fall into three classes. STAR-Result is confused with STAR-Action in cases where the technician omits a Result and the Action verb is past-tense (67 cases on the 2,141-note gold standard). ECO-Claim is confused with ECO-Opinion when the claim is hedged with terms such as “probably” or “looks like” (89 cases). Quantitative evidence is missed when the unit is non-

Field	Precision	Recall	F1
STAR-Situation	0.93	0.89	0.91
STAR-Task	0.88	0.84	0.86
STAR-Action	0.96	0.94	0.95
STAR-Result	0.92	0.88	0.90
ECO-Evidence	0.95	0.92	0.93
ECO-Claim	0.89	0.85	0.87
ECO-Opinion	0.91	0.83	0.87
Overall	0.92	0.89	0.91

Table 5: Field-level F1 on the 2,141-note gold standard. Overall F1 is 0.913 (reported as 91.3% throughout the paper).

standard, such as “approximately 15 percent” rather than “15%” (41 cases). Each failure mode is addressed by an additional regex pattern or a few-shot example in subsequent prompt revisions.

C.4 An Information-Theoretic Reading

Why should completeness and resolution time be related exponentially rather than in some other decreasing way? The following sketch offers one reason, by treating the handoff as a communication channel. The note carries some number of bits about the root cause, the engineer needs a certain number to act, and round-trips exist to close the gap. It explains why an exponential is a plausible shape; it is not evidence that the true relationship is exponential, and the comparison above shows the data do not settle that. A reader uninterested in the mechanism can skip to Section 5 without loss.

The argument is an interpretation, not a derivation. The entropy figures below come from a gold-standard subsample, linearity of the initial information I_0 in σ is an assumption, the exponential form follows under that assumption, and the fit above is consistent with the assumption without establishing it.

Treat the on-call engineer’s decision as a classification over root causes. Let θ be the true root cause, E the evidence in the ticket, and $H(\theta | E)$ the Shannon entropy of θ given E . On the gold-standard subsample this entropy drops approximately linearly as σ rises: $H(\theta | \sigma=0.0) \approx 1.47$ bits, $H(\theta | \sigma=0.5) \approx 0.91$, $H(\theta | \sigma=1.0) \approx 0.28$, an average drop of roughly 1.2 bits per unit of σ (the two halves of the range drop by 1.12 and 1.26, within 13%).

By Fano’s inequality, acting with error rate at most δ requires mutual information $I(\theta; O) \geq H(\theta) - h(\delta)$, with h the binary entropy; if the extracted record provides $I_0(\sigma)$ bits, the deficit

closed by further communication is $\Delta I(\sigma) = H(\theta) - I_0(\sigma) - h(\delta)$, and if each round-trip carries at most c bits, the expected round-trip count is bounded below by $\Delta I(\sigma)/c$. Taking I_0 approximately linear in σ , that count scales as $e^{-\beta\sigma}$, with β governed by c and the information-per-unit- σ ratio.

STAR fields are designed so each one contributes separately to root-cause classification, which makes linearity a natural assumption here; the functional-form comparison above is the check that can weaken it. Under this interpretation, β is the rate at which extraction completeness reduces the engineer’s residual decision uncertainty. It is a property of the technician–engineer channel rather than of the failure mode, which would instead make β vary by category. Table 3, with every per-category interval containing the pooled estimate, is the regularity the channel reading predicts; consistency, not proof, is what the fits provide.

Confound: σ versus ticket difficulty. The σ – τ relationship is observational. Complete notes may also mark problems that were easy to diagnose, and σ ’s weights were fit to predict first-pass closure (Section 4), so the score is not independent of the outcome it explains. A controlled experiment on a held-back subset has not been done; the Limitations section states what the observational result does and does not support.

Two pieces of evidence partially mitigate the difficulty confound but do not eliminate it. First, the cross-category stability of β (Table 3) is hard to explain through a difficulty confound alone, because the relationship between difficulty and note completeness should differ across incident categories with different physical complexity. Second, the reopen-rate and round-trip results in Section 4 are measured at the system level (deployment versus pre-deployment) rather than at the ticket level. A pure difficulty confound would predict that the system-level metrics depend on the difficulty distribution of incidents in the deployment window, which the composition stability in Table 7 does not support. We do not claim that $\beta \approx 1.64$ is universal across organizations; whether the exponent reproduces elsewhere is a question for future work.

Bootstrap procedure. Intervals on β resample individual tickets, stratified by failure category so the per-category counts are held fixed, refit α , β and γ jointly by nonlinear least squares on each replicate, and report bias-corrected and accelerated

(BCa) intervals over 10,000 replicates. Roughly 0.4% of replicates failed to converge and were replaced by further draws; conditioning on convergence in this way can narrow the interval slightly, and we report it rather than leave it implicit.

C.5 Functional-Form Comparison

Table 6 gives the comparison summarized in Section 4. All three forms are fitted to the same 2,141 gold-standard tickets by nonlinear least squares against a common total sum of squares.

Form	k	RSS	AICc	ΔAICc
Exponential	3	9,000.0	3080.3	0.0
Power law	3	9,005.9	3081.7	1.4
Linear	2	28,200.0	5523.5	2443.2

Table 6: Candidate functional forms for the relationship between σ and τ ($n = 2,141$, total sum of squares 150,000). k counts fitted parameters. The exponential and power-law forms have comparable support — Akaike weights 0.67 and 0.33, an evidence ratio near two — so the data do not identify the exponential uniquely; the linear form is decisively worse. Section 4 states the consequence for Contribution 4.

D Operational Lessons

Change management. The rollout was treated as a change-management exercise, not a software deployment. Technician representatives were involved from before the pilot. Training consisted of a two-hour course covering STAR and ECO with anonymized real tickets, a one-page reference guide at each site, and a three-month refresher. The rollout was phased one region at a time. Each region’s go/no-go was held until local technician feedback was collected. The phased approach added several months. It preserved the option to back out. Technician acceptance reached 94% after training.

System reliability. The system runs active-passive across regions with automatic failover within 30 seconds. It degrades gracefully when the LLM API is unavailable. Extraction falls back to rule-based parsing only. Classification continues with reduced accuracy. Affected tickets are flagged for human review. Operational visibility uses Prometheus metrics and Grafana dashboards. Incident response follows a 15-minute SLA. Post-incident reviews follow every Sev1 or Sev2.

Edge cases. Three classes required accommodations. Multi-component failures, where one ticket covers concurrent issues, are split at extraction time into linked sub-tickets. Intermittent issues required extending the telemetry history window from 24 to 72 hours. Non-English notes are handled by parallel-language prompts with translation of structured fields, with the accuracy gap quantified in the Limitations section. Proper multilingual support is in progress.

Privacy and security. Technician notes can contain device names, location identifiers, and rarely personal names. Only diagnostics needed for classification are collected. Raw notes are purged after 90 days. Structured fields are retained. Neither raw notes nor structured fields are used for technician performance evaluation. Access is role-based with full audit logging. Data is encrypted at rest and in transit. The system operates under SOC 2 Type II certification, and under GDPR for European regions.

Conservative auto-close. The auto-close rate sits at 0.3%. We auto-close only the clearest cases. Closure occurs without pre-close engineer approval. Every auto-closed ticket is reported to the weekly operational review and remains available there for discussion or escalation; separately, a 5% sample receives an end-to-end review with the datacenter technician (Section 4). Across the eight-month deployment, no auto-closure was overturned and no case was identified in review as one that should have been closed differently. Higher auto-close rates are achievable from this baseline. They were not achievable at the outset.

Detailed lessons. Five observations in roughly decreasing order of confidence.

The first is that evidence quality matters more than evidence quantity. The fits put 60% of the extraction-completeness weight on evidence indicators and 40% on schema completeness. This is consistent with operational experience. Tickets with one quantitative measurement plus a clear Result resolved faster than tickets with all four STAR fields filled but no numbers.

The second is that conservative automation builds the trust needed for aggressive automation. Auto-closing only 0.3% of tickets under this level of post-close scrutiny appears underambitious. It allowed the rest of the pipeline to be trusted enough to run unsupervised.

	Pre-deploy.	Deploy.
Link flaps	36.9%	35.9%
CRC errors	31.2%	31.5%
Optical issues	18.5%	19.1%
Physical damage	8.1%	8.3%
Configuration errors	5.3%	5.2%
Blocked share of all tickets	3.0%	1.5%
SRE headcount (window)	24	24

Table 7: Blocked-subpopulation composition by window (window averages, in percent of the blocked subpopulation). Link flaps and CRC errors hold steady period to period; the remaining categories vary within a few points with no directional drift.

The third is that the feedback loop with the people doing the work is the slowest-changing part of the system but the one with the highest leverage. Most per-region accuracy improvements came from technicians telling us when the checklist asked for something that did not apply, and us removing it.

The fourth is that cultural change lags technical deployment by a measurable margin. Technician acceptance reached 94% after training. Full adoption (structured notes by default, even unprompted) took roughly four months longer than the technical rollout.

The fifth is that concrete MTTR numbers drive organizational buy-in in a way that schema-design arguments do not. The 50% reduction is what got the second-year budget approved. The information-theoretic argument explained, after the fact, why the reduction held. Both mattered. They mattered in that order.

E Staggered-Rollout Treatment-Effect Analysis

The production rollout provides staggered treatment timing across the datacenter footprint. The four rollout waves were:

Wave	Cumulative DCs treated	Not yet treated
1	5	445+
2	40	410+
3	80	370+
4	full footprint	0

Waves 1–3 lasted approximately 2.5, 1, and 1 months respectively; wave 4 covers the remainder of the window.

For each datacenter d , let G_d denote its first treatment period. Outcomes are aggregated into two-week periods. For rollout cohort g and period t , the

group-time treatment effect is

$$\text{ATT}(g, t) = \mathbb{E}[Y_{d,t}(1) - Y_{d,t}(0) \mid G_d = g], \quad (5)$$

where the counterfactual outcome for treated datacenters is estimated using datacenters that have not yet received the rollout at time t (Callaway and Sant’Anna, 2021).

We estimate these effects separately for:

1. MTTR, measured through return of the affected link or device to production;
2. the fraction of closed tickets followed by an incident against the same device or physical link within 30 days; and
3. the mean number of technician–engineer round-trips per blocked ticket, where a round-trip is one engineer request followed by a technician response — a coordination measure, not a count of site visits.

Already-treated datacenters are not used as controls for later-treated datacenters (Goodman-Bacon, 2021). After the final rollout wave, no not-yet-treated comparison group remains, so identification of the staggered effect comes from periods in which rollout is incomplete.

The treatment is the complete deployed package. The extraction pipeline, technician training, one-page site guidance, rollout gates, and associated process changes were introduced with the rollout waves. The analysis does not attempt to identify a language-model-only effect.

Event-time estimates. Each datacenter is indexed relative to its own first treatment period. Event time $k = 0$ denotes the rollout period, negative values denote periods before rollout, and positive values periods after. Table 8 reports the coefficients individually rather than compressed into a single joint test, following the reporting guidance of Roth et al. (2023); Figure 3 plots the same estimates. The estimator output is given directly with no smoothing applied. Standard errors widen with $|k|$ because fewer rollout cohorts contribute at longer horizons. Three waves are estimable at $|k| \leq 3$, two out to $|k| \leq 7$, and one beyond.

We report the pre-treatment coefficients because they are the evidence bearing on the parallel-trends assumption the group-time estimates rest on. The post-treatment sequence rises over the first three periods and then flattens, which is consistent with

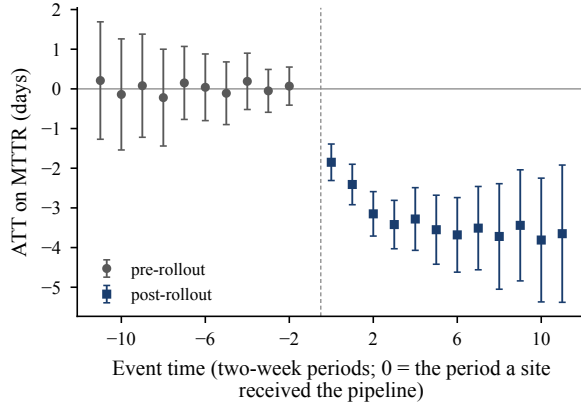


Figure 3: Event-time treatment effects on MTTR under the staggered rollout, in days, with simultaneous 95% confidence bands. Event time counts two-week outcome periods relative to the period in which a given datacenter received the pipeline, not the duration of the rollout itself. Circles are pre-treatment periods and squares post-treatment; the dashed line marks the rollout boundary and $k = -1$ is the reference period.

incomplete adoption immediately after rollout; it is not monotone thereafter, and we do not read the period-to-period movement, since datacenter incident arrivals are stochastic and early post-rollout periods can contain incidents that entered the workflow before rollout. The aggregated group-time ATT is the summary of the rollout effect.

Blocked-ticket evaluation rule. For the evaluation in this paper, blocked-ticket membership is constructed retrospectively from incident-management workflow metadata. A fixed three-day threshold from the recorded queue-entry transition is applied identically across comparison periods. No free-text judgment and no value produced by the extraction pipeline enters this selection rule. In particular, STAR fields, ECO fields, σ , and routing state cannot change evaluation membership.

Operationally, the broader blocked state means that recovery is waiting on a required technician action. The three-day threshold is used only to construct the fixed evaluation subset reported in this study.

Event time	ATT	SE	95% band	Cohorts
-11	0.21	0.58	$[-1.27, 1.69]$	1
-10	-0.14	0.55	$[-1.54, 1.26]$	1
-9	0.08	0.51	$[-1.22, 1.38]$	1
-8	-0.22	0.48	$[-1.44, 1.00]$	1
-7	0.15	0.36	$[-0.77, 1.07]$	2
-6	0.04	0.33	$[-0.80, 0.88]$	2
-5	-0.11	0.31	$[-0.90, 0.68]$	2
-4	0.19	0.28	$[-0.52, 0.90]$	2
-3	-0.05	0.21	$[-0.59, 0.49]$	3
-2	0.07	0.19	$[-0.41, 0.55]$	3
-1	reference period			
+0	-1.85	0.18	$[-2.31, -1.39]$	3
+1	-2.41	0.20	$[-2.92, -1.90]$	3
+2	-3.15	0.22	$[-3.71, -2.59]$	3
+3	-3.42	0.24	$[-4.03, -2.81]$	3
+4	-3.28	0.31	$[-4.07, -2.49]$	2
+5	-3.55	0.34	$[-4.42, -2.68]$	2
+6	-3.68	0.37	$[-4.62, -2.74]$	2
+7	-3.51	0.41	$[-4.56, -2.46]$	2
+8	-3.72	0.52	$[-5.05, -2.39]$	1
+9	-3.44	0.55	$[-4.84, -2.04]$	1
+10	-3.81	0.61	$[-5.37, -2.25]$	1
+11	-3.65	0.68	$[-5.38, -1.92]$	1

Table 8: Event-time coefficients for MTTR in days, reported individually rather than as a single joint pre-trend test. Intervals are simultaneous 95% confidence bands and are therefore wider than the pointwise intervals of Table 2. “Cohorts” is the number of rollout waves contributing at that event time; the final wave has no not-yet-treated comparison group and does not enter.