MEDSHARD: TASK-STRUCTURED INFERENCE IMPROVES CLINICAL QUALITY CONTROL ACROSS MODEL SCALES

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured clinical-document audits require a language model to track many rules, locate their evidence, and emit a variable number of findings. We introduce MedShard, a task-structured inference harness that compiles this workload into rule-specific execution paths. MedShard preserves rule-relevant fields, resolves eligible cases with deterministic prechecks, and combines focused model judgments with a deployed regex system through development-calibrated per-rule routing. On 250 Chinese admission notes evaluated against 19 audit rules, MedShard with a fixed 27B model achieves micro-F1 of 0.751 (95% CI [0.726, 0.776]), compared with 0.355 for direct prompting and 0.621 for the regex system. The complete LLM tier reaches 0.702, and calibrated routing improves on blanket union by 0.016 F1. Across 4B, 9B, 27B, and 122B-A10B configurations, MedShard improves F1 by 0.379–0.532 over each checkpoint's direct-prompt baseline. The routed 4B system exceeds direct 122B-A10B, while dense 27B achieves the highest routed score. On two mutually exclusive error-type benchmarks, decomposition increases detection precision while reducing type accuracy. These results connect rule-level inference design, evidence-source complementarity, and output structure to the performance of locally deployed clinical-document quality control.

1 INTRODUCTION

When a language model underperforms, the usual remedy is a bigger one: more parameters, more training, or a stronger hosted system. But a deployed system is never a bare model. Around the weights sit input transformations, task decomposition, deterministic tools, output verification, and routing policies, which together form an *inference harness*. Larger models are of course stronger on average, so the open question is *when engineering the computation around a fixed model can substitute for model scale*. We evaluate whether task-structured inference enables a smaller checkpoint to exceed the performance of direct prompting with a larger checkpoint on the same audit task.

Structured quality control (QC) of clinical notes makes the question concrete. One admission record has to be checked against explicit rules for facts that are missing, mutually inconsistent, or malformed, so the task pairs long-context reading with many fine-grained decisions. It also carries an unusually hard constraint on which model is admissible at all: regulation and hospital policy routinely forbid sending identifiable records to an external API, and a site that can host a dense 27B model on one accelerator cannot host a frontier-scale model at useful throughput. The rules are explicit to begin with: a hospital QC specification enumerates them, and the regex checker already deployed at our site covers the subset that pattern matching can reach. What stays open is the remainder: criteria that need semantic reading or comparison across fields, decided on records that cannot leave the hospital. Existing clinical NLP systems span symbolic extraction (Savova et al., 2010), neural coding (Hu et al., 2021), local privacy-preserving pipelines (Wiest et al., 2024; 2025), and LLM-based medical reasoning (Singhal et al., 2023). We study how rule-level inference structure changes the performance of fixed local language models on a shared clinical-document audit.

We study a Chinese admission-note benchmark with 19 non-exclusive QC rules. Monolithic prompting of Qwen3.5-27B obtains F1 0.355, below both a production regex system (0.621) and a hosted

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

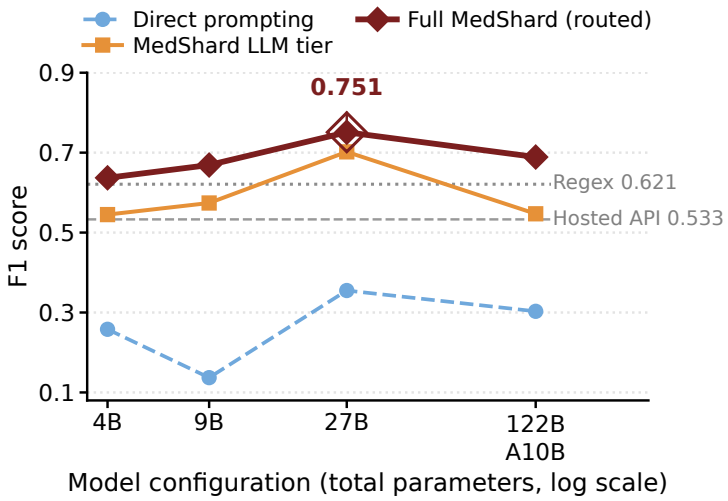


Figure 1: Within-checkpoint harness gains on the 250-record CHIP evaluation partition. Routing is re-calibrated per model on the development partition; the horizontal lines are model-independent references. 122B-A10B is a mixture-of-experts model with 10B active parameters.

proprietary-model reference (0.533). We introduce **MedShard**, which moves four burdens out of the monolithic prompt: it structures the note once, shards the audit into one decision per rule, applies rule-dependent prechecks and focused judgments, and calibrates whether each rule is served by the LLM tier, by regex, or by their union. A one-way verifier is available as a configuration flag, not a mandatory stage. The weights never change. Figure 1 previews the outcome: every tested checkpoint gains substantially over its own direct-prompt score, and the best routed system is the dense 27B configuration, not the largest one.

MedShard defines a rule-level execution interface that combines evidence-preserving field views, deterministic prechecks, focused judgments, and development-calibrated selection of LLM-tier and regex findings. This interface supports local execution, exact aggregation of rule-tagged outputs, and model-specific routing under a shared audit specification. Measuring within-checkpoint gains across dense 4B, 9B, and 27B models and a 122B-A10B mixture-of-experts configuration, we find that full MedShard reaches F1 0.751 with 27B and that every routed configuration exceeds the hosted direct-prompt reference. We examine how output structure interacts with decomposition by comparing a multi-rule audit with two mutually exclusive error-detection tasks. We also take the harness apart component by component, trace routing behavior rule by rule, attach paired-bootstrap uncertainty to every comparison, examine where the design fails, and document how to recalibrate it for a new rule set or site. An anonymous reproducibility package accompanies the paper.

2 Related Work

Inference-time structure. Prompt compression, decomposition, and test-time adaptation improve model use without changing base weights (Jiang et al., 2023; Prasad et al., 2024; Shi et al., 2024). Multi-agent systems similarly distribute work across roles (Wu et al., 2024; Hong et al., 2024), including medical decision making (Kim et al., 2024) and coding (Motzfeldt et al., 2025). MedShard's optional verifier is instead a one-way veto, not a debate role. Recent evidence shows that autoregressive LLMs can suppress all but one label during multi-label generation (Ma et al., 2025); decomposed and dichotomic prompting instead query output dimensions separately (Nie et al., 2025; Langner et al., 2025). Learned routers such as RouteLLM choose among models at request level (Ong et al., 2025). MedShard combines a rule output graph, evidence-preserving field views, logically safe prechecks, and per-rule LLM/regex/union routing.

Clinical QC and hybrid systems. Rule-based clinical NLP remains attractive for auditability but is brittle on semantic and cross-field criteria (Savova et al., 2010; Stanfill et al., 2010). Neuro-symbolic systems combine learned extraction with deterministic verification (Prenosil et al., 2025). Recent benchmarks frame medical error detection as a language-model task, notably MEDIQA-CORR and MEDEC (Ben Abacha et al., 2024; 2025). Chinese clinical NLP offers broad benchmarks (Cai et al., 2024), but admission-note QC differs from single-error correction because a record can violate many rules simultaneously. This distinction becomes the empirical scope condition studied below.

Small local medical models. Small open models can be competitive for EHR parsing (Lehman et al., 2023), and local processing reduces disclosure risk (Wiest et al., 2024; Liu et al., 2023). Prior work primarily asks which model or adaptation method to use. We instead hold training fixed and ask whether a structured inference program can recover more performance than increasing parameter count.

3 MEDSHARD

MedShard replaces one monolithic audit call with a program over record–rule pairs; Algorithm 1 in Appendix B states its control flow. The LLM tier is a hybrid execution path comprising deterministic field normalization and rule prechecks, followed by a focused model call for deferred cases. In the final 27B configuration, eight rules are resolved entirely by prechecks. An optional one-way verifier can veto a positive judgment, and the reported policy leaves it off. The regex tier reads the raw note along an independent path. A per-rule route, calibrated once on a held-out development partition, then decides whether a rule's findings come from the LLM tier, the regex tier, or their union.

3.1 TASK AND DESIGN PRINCIPLE

A record x contains seven free-text fields. The rule set $\mathcal{R} = \{r_1, \ldots, r_K\}$ has $K = 19$ and spans format normality, information completeness, cross-field consistency, and diagnosis quality. The scored target is a set

$$Y(x) = \{(r, f) : r \text{ is violated in field } f\}.$$

Unlike single-label classification, several rules can fire at once and one rule can produce several findings. Evaluation exactly matches (r, f) pairs and micro-aggregates true positives, false positives, and false negatives.

The key structural assumption is that rule outputs are *union-composable*: deciding one rule does not force another to be negative. MedShard exploits this structure by evaluating each (x, r) separately and taking the set union of the outputs. This trades one overloaded prompt for a program of small decisions whose inputs and tools are specialized by rule.

3.2 A STRUCTURAL SCOPE HYPOTHESIS

For rule r, let $\mathcal{F}_r$ be the set of all valid field-tagged findings (r, f) that the rule can emit, and let $Y_r(x) \subseteq \mathcal{F}_r$. The primary task admits any union $\bigcup_r Y_r$ with $Y_r \subseteq \mathcal{F}_r$, and the system predicts $\hat{Y}(x) = \bigcup_{r=1}^{K} \hat{Y}_r(x)$. Because the rule-specific finding sets are disjoint, TP, FP, and FN counts each sum across rule partitions, and micro-F1 is computed once from the aggregated totals. The output factorization permits independent rule execution and exact aggregation of rule-tagged findings. MedShard uses this factorization to localize evidence and specialize inference for each rule; the within-checkpoint comparisons quantify the resulting performance gains.

The same argument fails for a mutually exclusive label $c \in \{0, \ldots, K\}$. Separate binary judges produce scores that must compete or be reconciled, and multiple positives violate the output constraint. This predicts where harness gains should be largest: where (i) sub-checks are explicit, (ii) each sub-check is *separately addressable*, in the sense that its evidence can be localized to a stable view of the record and its output scored on its own, and (iii) outputs compose by union. Mutually exclusive tasks instead call for joint normalization or a constrained selector. We examine this prediction in the scope probes below.

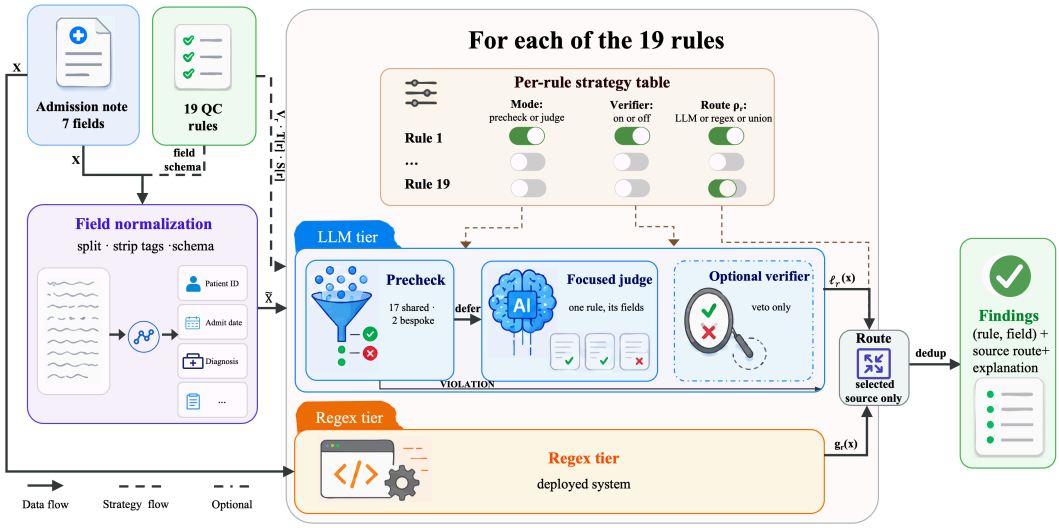


Figure 2: MedShard evaluates each record–rule pair through a task-structured LLM tier and an independent regex tier. Field normalization, the prechecks, and the regex tier are deterministic; only the focused judge and the optional verifier call the model. A precheck reads all normalized fields, emits a violation directly, and reaches the judge only when it defers.

3.3 The LLM Tier

Figure 2 shows how the two tiers and the route table fit together. The LLM tier is built from four stages.

Field normalization. A deterministic preprocessing step splits each admission note by its standard section headers, removes organizer-supplied inline tags, and converts diagnosis arrays into a stable textual schema. Each rule then receives only the fields named in its definition. This is selection, not semantic summarization: wording inside a field is preserved. Irrelevant context drops away without spending another model call to rewrite evidence. Write $\tilde{x}$ for the normalized record, $\mathcal{V}_r$ for the fields rule r names, and $\tilde{x}[\mathcal{V}_r]$ for the view the rule actually receives.

Per-rule sharding and prechecks. Each judge receives one rule definition, only its relevant fields, decision criteria, examples, and a constrained output schema. Every rule has a deterministic precheck that reads the normalized fields and returns VIOLATION with the offending field set, CLEAR, or DEFER, and it never converts uncertainty into a positive finding. Seventeen prechecks reuse a shared registry of field-presence and format tests, and two rules carry bespoke guards. A rule whose strategy is precheck-only has a total precheck that never defers. Eight rules are decided this way in the final 27B policy and issue no model call.

Rule templates and targeted corrections. Sixteen rules use a shared few-shot template populated with rule-specific content. Three rules—a present-illness severity requirement, the scope of a laboratory-format rule, and a diagnosis-completeness condition—kept failing under the shared template, so each received a rewritten rule-specific template. These three rewrites are the *targeted corrections* in the configuration trajectory below, and like every other design choice they were selected on development outputs alone.

Optional verification. After a positive focused judgment, the harness can issue a second, one-way call that confirms or vetoes each proposed finding. The verified set is a subset of the judged set by construction: the call cannot add a finding, change its field, or initiate dialogue. A per-rule flag controls the role, so a deployment chooses how much verification to pay for.

Table 1: Reference-finding distribution. "Full" is the 300-record corpus; "Evaluation" is the fixed 250-record partition.

Rule family	Rules	Full	Evaluation
Format normality (EN)	6	512	424
Information completeness (IC)	7	333	277
Cross-field consistency (CO)	4	68	54
Diagnosis quality (DQ)	2	11	10
Total	19	924	765

Let $\ell_r(x)$ denote the LLM-tier findings for rule r. A strategy table fixes the internal path of each rule: precheck only, precheck then a focused judge, or focused judge then the optional verifier. All calls use the same base model, with no fine-tuning and no inter-agent conversation.

3.4 Regex Tier and Calibrated Routing

The regex tier is the hospital's deployed matcher, the same system we also report as a standalone baseline. It reads the raw note independently of the LLM tier and returns $g_r(x)$. For each model configuration and rule, development outputs select a route

$$\rho_r \in \{\text{LLM}, \text{REGEX}, \text{UNION}\}$$

as $\rho_r = \arg\max_s \mathrm{F1}_{r,s}$, where $\mathrm{F1}_{r,s}$ scores only predictions of rule r from candidate source s on the 50 development records. When every candidate scores zero, a fixed tie rule keeps the LLM source. Per-rule F1 is a calibration proxy for this discrete choice; the reported system metric is micro-F1 after all rule counts are aggregated. The policy is 19 small discrete choices, not a learned router, and at a new site labeled calibration examples adapt the table to local rule definitions. The final output is

$$\hat{Y}(x) = \bigcup_{r \in \mathcal{R}} \begin{cases} \ell_r(x), & \rho_r = \text{LLM}, \\ g_r(x), & \rho_r = \text{REGEX}, \\ \ell_r(x) \cup g_r(x), & \rho_r = \text{UNION}. \end{cases}$$

Duplicate (r, f) pairs are removed. For the 27B system, 12 rules use the LLM tier and seven use the union, and none uses regex alone. The per-rule choice matters because the two tiers are complementary on some rules but redundant or precision-diluting on others.

Execution and observability. Calibration scores all three candidate sources for every rule on the development records, whereas deployment executes only the frozen route: a rule assigned to regex issues no model call. Beyond the eight precheck-only rules, a precheck also settles some judge-mode rules outright. Each output is parsed into a positive or no-finding decision, mapped back to the rule's target field, and stored with its source route, runtime, and free-text explanation, so a reviewer can audit any finding. No rule call depends on another.

4 Experiments and Results

4.1 Setup

Primary benchmark and partition. The primary corpus, *CHIP-300*, contains 300 Chinese admission notes from the medical-record QC track of the 11th China Health Information Processing Conference (CHIP 2025) (China Health Information Processing Conference, 2025). Organizers distributed de-identified records and hospital-QC labels to task participants under the shared-task terms. Each record follows the standard seven-field admission structure (Gao et al., 2019). The corpus contains 924 reference findings across 19 rules, with rule frequencies from 2 to 219; Table 1 gives the distribution. We fix a seed-42 partition of 50 development and 250 evaluation records. The evaluation partition contains 765 findings (3.06 per record), and all reported per-model routing choices use only development outputs. Format and information-completeness findings account for 701 of the 765 evaluation labels, defining the principal workload represented by the reported micro-F1.

Table 2: Finding-level performance on the 250-record evaluation partition. The routed system has F1 95% CI [0.726, 0.776].

System	P	R	F1
Qwen3.5-27B direct	0.276	0.497	0.355
Regex	0.760	0.525	0.621
Hosted API snapshot	0.725	0.421	0.533
MedShard-27B, LLM tier	**0.800**	0.626	0.702
Blanket LLM–regex union	0.717	**0.754**	0.735
MedShard-27B, routed	0.785	0.720	**0.751**

Models, baselines, and metrics. We use Qwen3.5-{4B, 9B, 27B, 122B-A10B}; 4B, 9B, and 27B are dense, while 122B-A10B is a mixture-of-experts model with 10B active parameters (Qwen Team, 2026). Models are served locally with vLLM 0.17.1 in non-thinking mode, temperature 0, top-$p = 1$, and no parameter updates. Judge calls allow at most 2,048 output tokens. The 27B model is the primary configuration. Baselines are: (1) *direct*, a single prompt containing the record and all 19 rule summaries; (2) the existing *regex* system; (3) a *recorded hosted-API reference from April 2026*; (4) the complete *LLM tier* without regex; (5) a *blanket union* that merges regex for every rule; and (6) coarser record- and rule-family-level hybrids, reported in Appendix C. The primary metric is micro F1 over exact (r, f) findings, with precision and recall. We score each system's final emitted findings under the released matching convention. MedShard deduplicates findings during aggregation; residual duplicate predictions in comparator outputs are retained by the scorer. The reported scores therefore evaluate the complete output pipelines. Confidence intervals quantify record-sampling variation for the frozen predictions, using 1,000 record-level bootstrap resamples (seed 42) and 95% percentile intervals; paired comparisons reuse the same indices and report one-sided empirical tail probabilities.

Scope probes. We evaluate the task-structure hypothesis on two benchmarks where a record has at most one error from five mutually exclusive types. *MEDEC-300* is a 300-record English sample from MEDEC (Ben Abacha et al., 2025). *Synth-MedErr-CN* is a Chinese counterpart with 350 records: 200 contain one injected error, balanced across five types, and 150 are error-free. It was constructed from a public clinical-record source independent of the primary corpus; injection and a separate QC pass use different prompts with Qwen3.5-27B to provide a controlled, balanced structural probe. For both probes, we compare an engineered monolithic prompt with matched per-type decomposition using the same model. Because these tasks lack the 19-rule regex tier, the comparison isolates decomposition and templates.

4.2 Harness Gains Across Model Configurations

Table 2 shows that the main gain is not a consequence of a stronger base model. With the 27B checkpoint fixed, the complete LLM-tier pipeline raises F1 from 0.355 to 0.702, and calibrated integration with regex raises it to 0.751. Table 6 in Appendix C attaches paired uncertainty to those comparisons: every interval excludes zero, and the routed system beats direct 27B by 0.396 ($p < 0.001$). Calibrated routing improves F1 by 0.049 over the complete LLM tier (95% CI [0.033, 0.067]) and by 0.016 over blanket union (95% CI [0.006, 0.026]). The corresponding point scores are 0.702, 0.735, and 0.751 for the LLM tier, blanket union, and calibrated routing, respectively. The coarser record- and rule-family-level hybrids fall .106 and .132 behind routed MedShard, confirming that the useful granularity is the individual rule.

Figure 1 makes the within-checkpoint comparison. Direct F1 for 4B, 9B, 27B, and 122B-A10B is respectively 0.258, 0.137, 0.355, and 0.303; their LLM-tier scores are 0.545, 0.574, 0.702, and 0.547, and their routed scores are 0.637, 0.669, 0.751, and 0.689. The harness raises every configuration by 0.379–0.532 absolute F1, and every routed configuration exceeds the hosted direct-prompt reference. This is substitution in the operational sense: a routed 4B configuration (0.637) more than doubles direct 122B-A10B (0.303).

Substitution has a ceiling, though. Dense 27B exceeds routed 122B-A10B by 0.062, so the best system is a co-designed pair, not the largest checkpoint. The route table acts as a capability adapter that

Table 3: Recorded Qwen3.5-27B configuration comparisons. The upper block reports prompt and verifier variants; the lower block reports the final-policy sequence with verification disabled.

Configuration	P	R	F1
Direct monolithic prompt	.276	.497	.355
Per-rule, raw rule text	.235	.863	.370
Per-rule, shared templates	.411	.542	.468
Candidate branch, verifier off	.363	.677	.473
Candidate branch, 7-rule verifier on	.493	.648	.560
Final LLM tier, no corrections	.772	.574	.658
Final LLM tier	.800	.626	.702
+ blanket regex union	.717	.754	.735
+ calibrated regex routing	.785	.720	**.751**

makes this concrete: 27B keeps an LLM contribution for all 19 rules (12 LLM, 7 union), whereas the 4B, 9B, and 122B-A10B tables each send two rules straight to regex, and the LLM-tier-to-routed increment ranges from $+0.049$ at 27B to $+0.142$ at 122B-A10B. The regex tier contributes most where the LLM tier is weakest, and the harness does not simply impose one favorable 27B policy on every model. Deployments can reuse the rule graph, field views, prechecks, and templates while recalibrating the discrete routes for whichever model fits their memory and throughput envelope; Appendix D reports each configuration's route counts.

4.3 Configuration Trajectory and Per-Rule Routing

Table 3 records two separate development branches rather than one factorial ablation. The upper block explores prompt interfaces on an earlier judge configuration and ends in a candidate that switches the one-way verifier on for seven rules. The lower block is the configuration line that produced the reported system, and it never enables a verifier, so rows are comparable inside a block but not across the horizontal rule that separates them. Within the upper block, moving from one monolithic output to raw per-rule prompts shifts the operating point toward recall, shared templates recover precision, and the seven-rule veto raises precision from 0.363 to 0.493 and F1 from 0.473 to 0.560. Within the lower block, the three targeted template corrections move F1 from 0.658 to 0.702, and the clean source-routing comparison is blanket union (0.735) against calibrated routing (0.751): per-rule selection keeps the complementary recall at higher precision. Together the two branches show that orchestration depth is a configuration choice, not a fixed cost: selective vetoing restores precision in one setting, while the best reported system stays single-pass.

Figure 3 exposes the same mechanism rule by rule. The six most frequent rules account for 602 of the 765 reference findings. Rule-level results show how the two sources combine: union improves the diagnosis-completeness rule from 0.822 to 0.889 and the history-completeness rule from 0.516 to 0.905. For a diagnosis-format rule, retaining the LLM tier preserves F1 0.864 versus 0.689 under union. A union route can land above both of its component scores, which is why per-rule calibration improves on blanket union. Because the router sees development labels only, the evaluation plot measures route transfer, not retroactive test-time selection. Full counts and routes are in Appendix B.

The residual errors concentrate where the design is weakest by construction. All four zero-F1 rules follow precheck-only paths in the final policy and together account for 18 of the 765 reference findings. Three have no positive development examples, leaving source selection at the default LLM-tier route. Their evaluation results therefore reflect the interaction between deterministic prechecks and development-calibrated source selection. A chief-complaint format rule shows the other side of the same mechanism: the LLM tier is equally silent on it, but four development instances sufficed to route it through regex and recover F1 0.667.

4.4 Scope Condition: When Does the Harness Help?

Table 4 maps the predicted structural boundary; one probe comes out negative and the other mixed, and we report both. On the primary union-composable task, the LLM tier improves F1 by 0.347 before regex is added. On MEDEC, sharding buys precision (0.736 to 0.796) at a real cost: recall

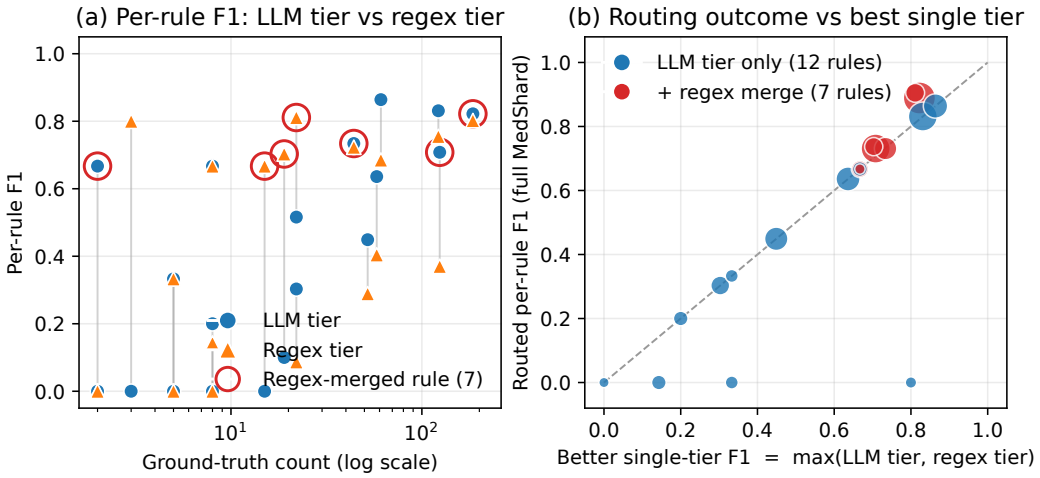


Figure 3: Per-rule mechanism for Qwen3.5-27B. Left: LLM-tier and regex F1 versus rule frequency; circled rules merge regex. Right: routed F1 versus the better single tier. Points above the diagonal show complementary unions; off-diagonal points show development-to-evaluation route variation. Marker area reflects evaluation frequency.

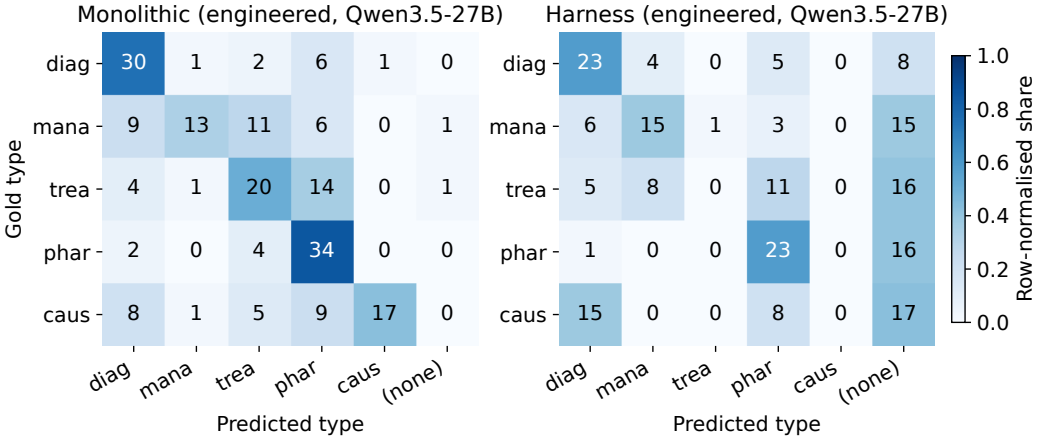


Figure 4: Error-type confusion matrices on the 200 positive Synth-MedErr-CN records. Independent category judges produce a distinct, higher-precision operating point.

falls from 0.701 to 0.539, accuracy from 0.693 to 0.667, and F1 from 0.718 to 0.643. On Synth-MedErr-CN, sharding raises detection accuracy from 0.574 to 0.731 and precision from 0.574 to 0.853, but detection F1 barely moves (0.727 to 0.731). The monolithic prompt calls almost every record faulty (recall 0.990), so its accuracy is close to the positive base rate of 0.571, and what sharding really buys is the ability to abstain. Type accuracy falls from 0.503 to 0.269 on MEDEC and from 0.570 to 0.305 on Synth-MedErr-CN (Figure 4). These results distinguish binary error detection from error-type assignment: the latter requires a decision among mutually exclusive labels, with abstentions scored as incorrect.

The pattern is what the union-composability criterion predicts, and it converts into a checklist for new tasks: does the output admit explicit sub-decisions, can each sub-decision's evidence be localized to a stable view, do positive outputs compose by union, and do deterministic detectors add complementary evidence? The first three favor sharding, the fourth favors routing, and a task that fails the third needs joint normalization or a constrained selector in place of a union. This is a design rule backed by one contrast, not a validated law.

Table 4: Scope-probe results with Qwen3.5-27B. "Type" is the share of positive records (167 for MEDEC, 200 for Synth-MedErr-CN) whose predicted error type matches the gold type, with an abstention counted as incorrect.

Data	System	Acc	P	R	F1	Type
MEDEC	Monolithic	.693	.736	.701	.718	.503
MEDEC	Harness	.667	.796	.539	.643	.269
Synth-CN	Monolithic	.574	.574	.990	.727	.570
Synth-CN	Harness	.731	.853	.640	.731	.305

4.5 Deployment Profile and Transfer

What the harness costs. Substitution here is substitution for *checkpoint selection*, not for inference compute, and the accounting should be explicit. The final 27B pipeline averages 9.9 focused model calls per record, with the remaining 9.1 rule checks resolved deterministically. Direct prompting issues one generation. Its recorded deployment uses one A100-80GB GPU and processes a record in 10.0 seconds; all verifier flags are disabled. Those calls are short. The pipeline emits about 100 output tokens per record in total, because each judge returns one verdict against a fixed schema instead of serializing an unknown list. The harness buys accuracy with more calls but far smaller ones on a fixed checkpoint.

Deployment profile. Table 11 in Appendix F reports each configuration together with its hardware: 27B runs on A100-80GB, while 9B and 4B run on L40S. The 27B configuration fits on one 80GB A100 at FP16 with a 16k context window. The smaller configurations are far more verbose, emitting roughly 690 and 920 output tokens per record against 100 for 27B. Records are processed independently in an offline workload, so instances replicate across GPUs without cross-device communication. The three points show single-GPU feasibility across memory tiers, not an interactive latency budget.

Deployment role. MedShard supports documentation quality control and makes no diagnostic claim about a patient. It surfaces rule-, field-, and source-linked findings beside the original note for human review, so a workflow can audit route-specific behavior, monitor distribution shift, and retain clinician authority over every action. Local inference keeps note processing inside the deploying environment.

Transfer protocol and artifact. For a new site, align each rule with local fields and safe prechecks, then use a labeled calibration set to select verifier flags and the LLM/regex/union source per rule, and freeze those choices before evaluation. Code, prompts, and aggregate results are provided in the supplementary material; clinical records and identifiers are excluded.

Limitations. Evaluation covers one 300-record Chinese corpus, a 50-record development partition, four model configurations, and two mutually exclusive error-type probes. The recorded configurations characterize system-level changes across this setting. Site transfer entails aligning local fields and rules, selecting prechecks and templates, and calibrating source routes on labeled local records. The deployment measurements describe offline document auditing.

5 Conclusion

MedShard compiles a 19-rule clinical-document audit into rule-specific inference paths with deterministic prechecks and calibrated evidence-source routing. With model weights fixed, the 27B system improves F1 from 0.355 to 0.751, and all four tested configurations gain at least 0.379. Calibrated routing combines complementary LLM-tier and regex findings, improving F1 by 0.016 over blanket union. The scope probes further show that detection precision and error-type assignment respond differently to decomposition. Together, these results make inference structure a concrete design variable for local clinical-document quality control.

AI USE STATEMENT

Generative AI tools are part of the method in two places described above: Qwen3.5-27B injected the errors in Synth-MedErr-CN and ran its separate QC pass, and a hosted proprietary model serves as a comparison system. In addition, we used generative AI assistants to help polish paper and debug the code. The authors reviewed all AI-assisted work: code was run and its outputs were checked against the stored result files from which every reported number is taken. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The primary corpus consists of de-identified admission notes with hospital-QC labels, distributed by the CHIP 2025 organizers to task participants under the shared-task terms. Synth-MedErr-CN was constructed from a public clinical-record source independent of the primary corpus. Clinical records and identifiers are excluded from the anonymous artifact. MedShard supports documentation quality control and makes no diagnostic claim about a patient; it surfaces findings beside the original note for human review, and local inference keeps note processing inside the deploying environment.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B describe the inference program and the per-rule policy; Section 4.1 and Appendix F give the data partition, model settings, and evaluation protocol. Code, prompts, and aggregate results are provided in the supplementary material; see its README for usage.

REFERENCES

Asma Ben Abacha, Wen-wai Yim, Yujuan Fu, Zhaoyi Sun, Fei Xia, and Meliha Yetisgen. Overview of the MEDIQA-CORR 2024 shared task on medical error detection and correction. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pp. 596–603, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.clinicalnlp-1.57. URL https://aclanthology.org/2024.clinicalnlp-1.57/.

Asma Ben Abacha, Wen-wai Yim, Yujuan Fu, Zhaoyi Sun, Meliha Yetisgen, Fei Xia, and Thomas Lin. MEDEC: A benchmark for medical error detection and correction in clinical notes. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22539–22550, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.1159. URL https://aclanthology.org/2025.findings-acl.1159/.

Yan Cai, Linlin Wang, Ye Wang, Gerard de Melo, Ya Zhang, Yanfeng Wang, and Liang He. MedBench: A large-scale Chinese benchmark for evaluating medical large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17709–17717, 2024.

China Health Information Processing Conference. The 11th China Health Information Processing Conference (CHIP 2025). http://www.cips-chip.org.cn/2025/, 2025. Medical-record quality-control shared-task track.

Yan Gao, Lei Gu, Yefeng Wang, Yandong Wang, and Feng Yang. Constructing a Chinese electronic medical record corpus for named entity recognition on resident admit notes. *BMC Medical Informatics and Decision Making*, 19(Suppl 2):56, 2019. doi: 10.1186/s12911-019-0759-2.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. Oral presentation.

Shuyuan Hu, Fei Teng, Lufei Huang, Jun Yan, and Haibo Zhang. An explainable CNN approach for medical codes prediction from clinical text. *BMC Medical Informatics and Decision Making*, 21 (Suppl 9):256, 2021. doi: 10.1186/s12911-021-01615-6.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. MDAgents: An adaptive collaboration of LLMs for medical decision-making. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Mikołaj Langner, Jan Eliasz, Ewa Rudnicka, and Jan Kocoń. Divide, cache, conquer: dichotomic prompting for efficient multi-label llm-based classification. In *2025 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 2084–2092. IEEE, 2025.

Eric Lehman, Evan Hernandez, Diwakar Mahajan, Jonas Wulff, Micah J. Smith, Zachary Ziegler, Daniel Nadler, Peter Szolovits, Alistair Johnson, and Emily Alsentzer. Do we still need clinical language models? In *Proceedings of the Conference on Health, Inference, and Learning (CHIL)*, volume 209 of *Proceedings of Machine Learning Research*, pp. 578–597, 2023. URL https://proceedings.mlr.press/v209/eric23a.html.

Zhengliang Liu, Yue Huang, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Yiwei Li, Peng Shu, Fang Zeng, Lichao Sun, Wei Liu, Dinggang Shen, Quanzheng Li, Tianming Liu, Dajiang Zhu, and Xiang Li. DeID-GPT: Zero-shot medical text de-identification by GPT-4, 2023. URL https://arxiv.org/abs/2303.11032.

Marcus Ma, Georgios Chochlakis, Niyantha Maruthu Pandiyan, Jesse Thomason, and Shrikanth Narayanan. Large language models do multi-label classification differently. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2472–2495. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.126. URL https://aclanthology.org/2025.emnlp-main.126/.

Andreas Geert Motzfeldt, Joakim Edin, Casper L. Christensen, Christian Hardmeier, Lars Maaløe, and Anna Rogers. Code like humans: A multi-agent solution for medical coding. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 22612–22627, 2025. URL https://aclanthology.org/2025.findings-emnlp.1231/.

Ercong Nie, Shuzhou Yuan, Bolei Ma, Helmut Schmid, Michael Färber, Frauke Kreuter, and Hinrich Schuetze. Decomposed prompting: Probing multilingual linguistic structure knowledge in large language models. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2025*, pp. 646–659. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-ijcnlp.38. URL https://aclanthology.org/2025.findings-ijcnlp.38/.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. ADaPT: As-needed decomposition and planning with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4226–4252, 2024. URL https://aclanthology.org/2024.findings-naacl.264/.

George A. Prenosil, Thilo K. Weitzel, et al. Neuro-symbolic AI for auditable cognitive information extraction from medical reports. *Communications Medicine*, 5, 2025. doi: 10.1038/s43856-025-01194-x.

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

Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C. Kipper-Schuler, and Christopher G. Chute. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010. doi: 10.1136/jamia.2009.001560.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Haotian Sun, Hang Wu, Carl Yang, and May Dongmei Wang. MedAdapter: Efficient test-time adaptation of large language models towards medical reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://aclanthology.org/2024.emnlp-main.1244/.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, et al. Large language models encode clinical knowledge. *Nature*, 620:172–180, 2023. doi: 10.1038/s41586-023-06291-2.

Mary H. Stanfill, Margaret Williams, Susan H. Fenton, Robert A. Jenders, and William R. Hersh. A systematic literature review of automated clinical coding and classification systems. *Journal of the American Medical Informatics Association*, 17(6):646–651, 2010. doi: 10.1136/jamia.2009.001024.

Isabella Catharina Wiest, Dyke Ferber, Jiefu Zhu, Marko van Treeck, Sonja K. Meyer, Radhika Juglan, Zunamys I. Carrero, Daniel Paech, Jens Kleesiek, Matthias P. Ebert, Daniel Truhn, and Jakob Nikolas Kather. Privacy-preserving large language models for structured medical information retrieval. *npj Digital Medicine*, 7(1):257, 2024. doi: 10.1038/s41746-024-01233-2.

Isabella Catharina Wiest, Fabian Wolf, Marie-Elisabeth Leßmann, Marko van Treeck, Dyke Ferber, Jiefu Zhu, Heiko Boehme, Keno K. Bressem, Hannes Ulrich, Matthias P. Ebert, and Jakob Nikolas Kather. A software pipeline for medical information extraction with large language models, open source and suitable for oncology. *npj Precision Oncology*, 9(1):313, 2025. doi: 10.1038/s41698-025-01103-4.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *Proceedings of the Conference on Language Modeling (COLM)*, 2024.

A Supplement and Artifact Scope

This appendix expands the strategy, per-rule results, statistical comparisons, scope probes, and deployment profile summarized in the main paper. All values come from the same frozen inference runs. The primary evaluation partition contains 250 records and 765 reference findings, the MEDEC probe contains 300 records, and Synth-MedErr-CN contains 350 records.

Code, prompts, and aggregate results are provided in the supplementary material, whose README describes their use. Primary records and record-level model outputs remain under their original data-access conditions and are not included.

B Final Per-Rule Configuration and Results

Table 5 gives the complete Qwen3.5-27B policy and evaluation F1 by rule. A judge route executes the deterministic precheck and calls the focused judge only when the precheck defers; superscript "p" marks the eight precheck-only rules. Superscript "c" marks the three corrected rule templates. "U" denotes the duplicate-free LLM–regex union, while "L" denotes the LLM tier. Routes are frozen by per-rule F1 on the 50-record development partition. The final system is single-pass: the optional verifier flag is off for every CHIP rule, and routed outputs are concatenated across rules.

Algorithm 1 MedShard inference under a frozen policy

Input: record x; rules $r_1, \ldots, r_K$; strategy table S; templates T; field views $\mathcal{V}_r$
Parameter: route $\rho_r \in \{\text{LLM}, \text{REGEX}, \text{UNION}\}$; mode $S[r].\text{mode} \in \{\text{PRECHECK}, \text{JUDGE}\}$; verifier flag $S[r].\text{ver}$
Output: scored finding set $\hat{Y}$ of (rule, field) pairs; the implementation also stores each finding's source and explanation
Note: All findings returned by Precheck, Judge, and Regex are tagged with the current rule; aggregation operates on (rule, field) pairs.

$\tilde{x} \leftarrow \text{Normalize}(x)$ {once per record}
$\hat{Y} \leftarrow \emptyset$
for $k = 1$ **to** K **do**
  $\ell \leftarrow \emptyset$; $g \leftarrow \emptyset$
  if $\rho_{r_k} \neq$ REGEX **then**
    $(v, F) \leftarrow \text{Precheck}(\tilde{x}, r_k)$ {all fields}
    if $v =$ VIOLATION **then**
      $\ell \leftarrow F$ {one finding per offending field}
    else if $v =$ DEFER **and** $S[r_k].\text{mode} =$ JUDGE **then**
      $\ell \leftarrow \text{Judge}(\tilde{x}[\mathcal{V}_{r_k}], r_k, T[r_k])$
      if $S[r_k].\text{ver}$ **then**
        $\ell \leftarrow \{f \in \ell : \text{Confirm}(f, \tilde{x}[\mathcal{V}_{r_k}], r_k)\}$
      end if
    end if{$v =$ CLEAR leaves $\ell = \emptyset$}
  end if
  if $\rho_{r_k} \neq$ LLM **then**
    $g \leftarrow \text{Regex}(x, r_k)$
  end if
  $\hat{Y} \leftarrow \hat{Y} \cup \ell \cup g$
end for
return $\hat{Y}$

Table 5: Complete per-rule strategy on the 250-record evaluation partition. GT is reference-finding frequency; L and R are the LLM and regex tiers. Detailed counts are provided in the supplementary material.

Rule	GT	L F1	R F1	Route	Final F1	Rule	GT	L F1	R F1	Route	Final F1
IC-RZCB-01	185	.822	.802	U	.889	EN-ZS-01^{p}	15	.000	.667	U	.667
EN-XB-01^{c}	124	.708	.370	U	.731	CO-XB-04^{p}	8	.000	.143	L	.000
EN-JW-01	122	.831	.755	L	.831	DQ-RZ-02	8	.667	.667	L	.667
EN-RZCB-01^{c}	61	.864	.685	L	.864	IC-JW-01	8	.200	.000	L	.200
EN-XB-02	58	.636	.404	L	.636	CO-XB-01^{p}	5	.333	.000	L	.333
IC-XB-01	52	.449	.289	L	.449	IC-ZK-01^{p}	5	.000	.333	L	.000
EN-FZ-01^{c}	44	.734	.723	U	.731	IC-ZS-01^{p}	3	.000	.800	L	.000
IC-XB-03	22	.516	.811	U	.905	IC-XB-02^{p}	2	.667	.000	U	.667
CO-XB-03	22	.303	.087	L	.303	DQ-RZ-01^{p}	2	.000	.000	L	.000
CO-XB-02^{p}	19	.100	.703	U	.737						

C Configuration and Statistical Detail

Table 3 records separate measured branches. The first three rows compare prompt interfaces, the two verifier rows quantify a seven-rule candidate, and the final four form the reported-policy sequence. "No corrections" precedes three rule-specific template changes, while both regex rows share the complete LLM tier.

The one-way verifier can confirm or veto a positive focused judgment, but cannot add a finding or start a dialogue. In the standalone candidate comparison, seven-rule verification changes P/R/F1 from .363/.677/.473 to .493/.648/.560. The separate final-policy development path records a verifier-flag count falling $3 \rightarrow 2 \rightarrow 0$ across development iterations and ends with every flag off; its final LLM and routed scores are .702 and .751. The candidate pair and final-policy path are separate, not cumulative.

Table 6: Paired-bootstrap difference between routed MedShard-27B and each comparator. The same 1,000 record-level bootstrap indices are used for both systems. Values are bootstrap means and may differ from rounded headline-score differences by .001.

Comparator	ΔF1 [95% CI]	p
Regex system	$+.130$ [.102, .157]	$< .001$
Record-level hybrid	$+.106$ [.081, .131]	$< .001$
Rule-family hybrid	$+.132$ [.105, .158]	$< .001$
Hosted API snapshot	$+.218$ [.163, .272]	$< .001$
Qwen3.5-122B-A10B direct	$+.447$ [.417, .479]	$< .001$
MedShard LLM tier	$+.049$ [.033, .067]	$< .001$
Blanket union	$+.016$ [.006, .026]	.003

D Cross-Configuration Detail

Table 7: Finding-level F1 and development-selected route counts across model configurations. L/U/R counts LLM, union, and regex-only rules. Routing is independently calibrated for each model.

Model	Direct	LLM	Routed	Gain	L/U/R
4B	.258	.545	.637	$+.379$	10/7/2
9B	.137	.574	.669	$+.532$	9/8/2
27B	.355	.702	**.751**	$+.396$	12/7/0
122B-A10B	.303	.547	.689	$+.386$	10/7/2

The 122B-A10B model is a mixture-of-experts system with 10B active parameters. Dense 27B is the best routed configuration. Model and harness are best selected jointly.

E SCOPE-CONDITION RESULTS

Table 8: MEDEC-300 error-flag results with Qwen3.5-27B. The official subtask-A-style error-flag accuracy is the primary metric. On the engineered rows, error-type accuracy over all 167 positive records—the same denominator used for Synth-MedErr-CN, with an abstention counted as incorrect—is .503 for the monolithic system and .269 for the harness.

MEDEC method	Acc	P	R	F1
Monolithic, minimal	.683	.665	.868	.753
Monolithic, engineered	.693	.736	.701	.718
Harness, minimal	**.703**	.710	.790	.748
Harness, engineered	.667	**.796**	.539	.643

Table 9: MEDEC engineered operating points across model configurations.

Model	Monolithic Acc	Harness Acc
4B	.550	.537
9B	.650	.627
27B	.693	.667
122B-A10B	.683	.650

Table 10: Synth-MedErr-CN on 350 records. Detection metrics use the binary has-error label; TypeAcc is error-type accuracy on the 200 positive records.

Synth-CN method	Acc	P	R	F1	TypeAcc
Monolithic	.574	.574	**.990**	.727	**.570**
Harness	**.731**	**.853**	.640	**.731**	.305

F EVALUATION AND EXECUTION PROTOCOL

Finding matching. The primary unit is an exact $(\text{rule}, \text{field})$ finding. True positives, false positives, and false negatives are micro-aggregated over the partition. Routed unions are deduplicated before scoring; residual duplicates from comparator outputs are retained as a multiset under the released scoring convention. Precision, recall, and F1 follow their standard definitions. Confidence intervals resample records, not individual findings, which preserves within-record correlation.

Uncertainty. Each reported model configuration corresponds to one frozen inference run. The reported 95% intervals are percentile intervals from 1,000 bootstrap resamples of the 250 evaluation records with replacement, using seed 42; bootstrap resampling does not rerun model inference. Paired comparisons reuse each resampled index vector for both systems. The one-sided empirical tail probability tests whether the paired difference is non-positive.

Serving configuration. All open models use vLLM 0.17.1, non-thinking mode, temperature 0, top-$p = 1$, and a maximum of 2,048 generated tokens per judge call. The 27B deployment uses FP16 and a 16k context limit. Both node types run Ubuntu 24.04.3 LTS with NVIDIA driver 595.58.03 and CUDA 13.2 on Intel Xeon Platinum 8358 hosts (128 logical cores, 881–1007 GB host memory). Each record is normalized once and all 19 rules enter the local precheck stage; 17 checks come from the shared registry and two bespoke guards complete the final policy. Only deferred rules on a judge path invoke the model. The harness can optionally verify a positive judgment, but the reported CHIP

policy enables no verifier flags. Runtime and parser states are logged with each call. Rule calls and records have no inter-call dependency and may be parallelized.

Table 11: Recorded full-pipeline deployment profile, one model instance per GPU. Hardware is included to contextualize each operating point.

Model	Hardware	Sec./rec.	Rec./min	Memory
27B	A100-80GB	10.0	6.0	75GB
9B	L40S-46GB	25.3	2.4	28GB
4B	L40S-46GB	36.1	1.7	12GB