

CLINICAL REPORT LABELLING

Structured labelling of radiology reports at scale

From graph extraction and free-form prompting to
schema-constrained, two-phase, batched inference

AUTHOR

Neurologic AI

PUBLISHED

July 2026

DOMAIN

CT chest reports

ABSTRACT

Radiology reports are the cheapest source of clinical ground truth in medical imaging and the hardest to use, because they are written for humans. This paper describes a production system that converts free-text CT chest reports into a fixed 25-finding structured label set at archive scale, and the path that produced it: rule-based extraction, RadGraph-style information extraction, and free-form LLM prompting. The current system replaces prose-described output formats with a versioned Pydantic schema enforced by constrained decoding, uses a flat, untiered label taxonomy in place of an earlier per-finding tiered design, splits

extraction into a cheap presence phase and a conditional modifier phase, and submits the whole workload through an asynchronous batch interface with prefix caching. On a radiologist-annotated held-out set of ~100 studies the system reaches 0.96 overall accuracy with zero schema-invalid responses. The two-phase split exploits the fact that only 2.3 of 25 findings are positive in a typical report: it removes ~32% of input tokens and ~39% of output tokens at equal or marginally better accuracy. Combined with batch pricing and a 10× cached-input discount, the estimated cost of labelling a ~600,000-study archive falls from roughly \$1,800 to roughly \$500 (about 3.6× cheaper end to end), while eliminating the parse-failure and output-truncation classes of error entirely.

0.96

Overall accuracy, ~100
held-out studies

0

Schema-invalid responses

3.6×

Cheaper end to end:
~\$1,800 to ~\$500 per
archive pass

01

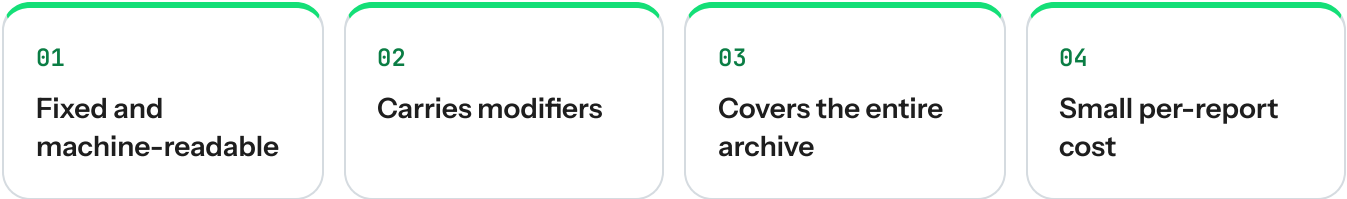
Introduction

Almost every imaging dataset used to train, validate, or audit a diagnostic model derives its labels, directly or indirectly, from the free-text report a radiologist dictated at read time. The report is authoritative, cheap to store, and already exists for every study ever performed. It is also unstructured, idiosyncratic, hedged, and written for another human rather than for a machine. Converting it into reliable structured labels is the rate-limiting step for training-set construction, cohort definition, registry reporting, production model auditing, and archive retrieval; the quality ceiling of every downstream model is set by this step.

The quality ceiling of every downstream model is **set by this step**.

The driver for this work was the need to produce study-level weak labels for a large CT chest corpus in order to train and validate vision-language models. That use case sets four requirements. The label set must be fixed and machine-readable, because a training

pipeline consumes it. It must carry modifiers (laterality, severity, size, lobe, chronicity), because a VLM evaluated only on presence/absence is not being evaluated on anything clinically interesting. It must cover the entire archive, because coverage determines training-set size. And its per-report cost must be small enough that relabelling the archive after a schema revision is an ordinary engineering decision rather than a budget request.



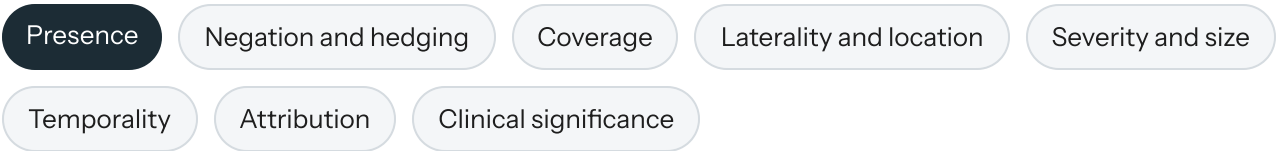
This paper describes why manual annotation, rule-based extractors, RadGraph-style information extraction, and free-form LLM prompting each fail the combined test of accuracy, cost, throughput, and operational reliability. It then presents our current approach (a schema-first task definition, a deliberately flat label taxonomy, constrained structured decoding, a two-phase presence-then-modifiers request pattern, and asynchronous batched inference) and reports its performance against those baselines.

02

The problem

2.1 What labelling a report actually requires

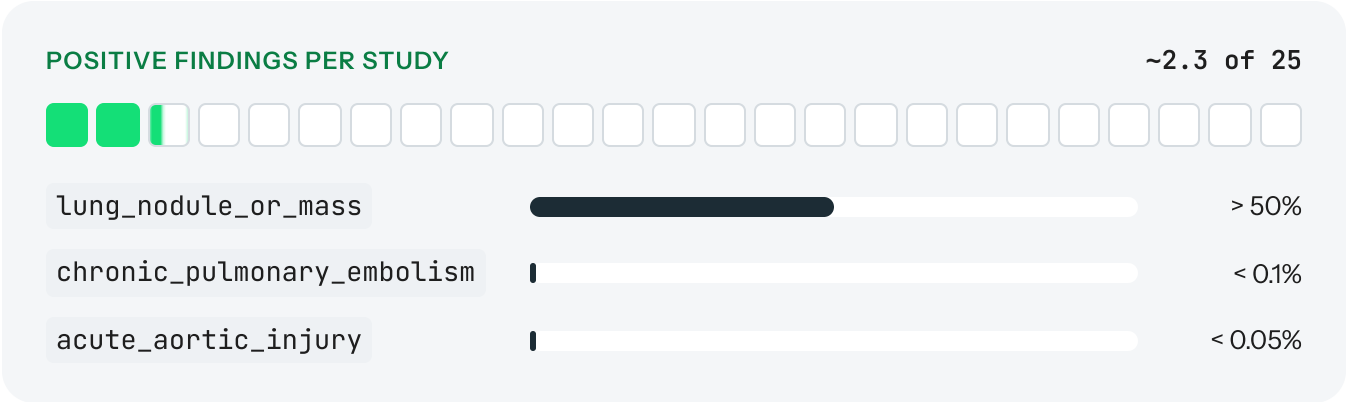
The task is simple to state and hard to specify. For each finding of interest the labeller must resolve **presence** (asserted, denied, or uncertain); **negation and hedging** (“no evidence of pneumothorax”, “cannot exclude”, “likely represents”); **coverage**, since silence is not negation; **laterality and location** down to lobe, segment or vertebral level; **severity and size**, including interval change in millimetres; **temporality**, distinguishing new from stable from a reference to a prior study; **attribution**, separating current imaging from clinical history and comparison exams; and **clinical significance**, including whether follow-up was recommended.



A correct label set must also respect the report’s internal logic. A finding described in the *Findings* section but omitted from the *Impression* carries different weight than one the radiologist chose to summarise. Speculative differentials must not be flattened into positive labels, and recommendations must not be read as observations.

2.2 The asymmetry that shapes the whole design

One property of radiology reports dominates the economics of the task: **reports are overwhelmingly negative**. The mean number of positive findings per study is approximately **2.3 out of 25**: roughly 10%. The distribution is extremely skewed: `lung_nodule_or_mass` is positive in more than 50% of studies, while `chronic_pulmonary_embolism` is positive in under 0.1% and `acute_aortic_injury` in under 0.05%.



Any design that pays the full cost of the modifier vocabulary for all 25 findings on every report is therefore paying roughly ten times over for the ~90% of finding slots that will be negative and carry no modifiers at all. Section 4.5 is the direct consequence of this observation.

03

How labelling is done today

3.1 Manual annotation by clinicians

Manual review is the reference standard precisely because it is expensive. Radiologist or trained-annotator review costs on the order of dollars and minutes per report, which puts corpora above a few thousand reports out of reach for most teams. It also does not remove the accuracy problem: inter-annotator agreement on hedged and incidental findings is consistently moderate rather than high, so multiple readers and adjudication are needed

for the very cases that matter most. Manual annotation does not scale, does not re-run when the schema changes, and creates a serial dependency on the scarcest resource in the organisation.

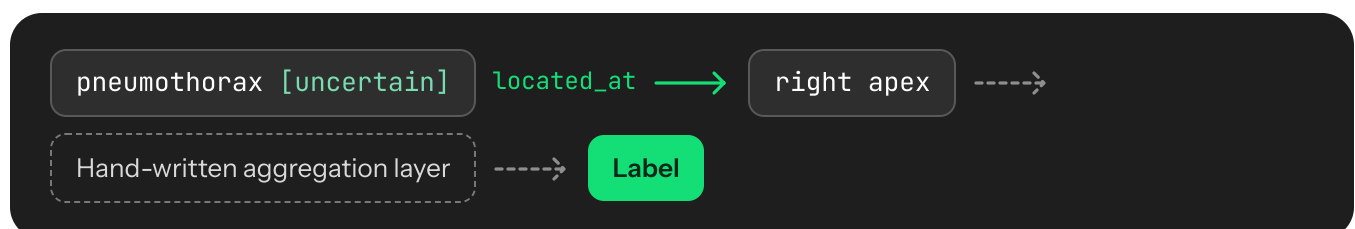
3.2 Rule-based extractors

The first generation of automated labellers (the CheXpert labeller, NegBio, and their descendants) combine a curated lexicon of finding mentions with dependency-parse rules for negation and uncertainty detection. They are fast, free to run, deterministic, and completely transparent: every label traces to the rule that fired. Their limitations are structural:

- **Fixed taxonomy.** They label a closed set of observations (classically 14 chest findings); adding one means authoring new lexicon entries and rules.
- **Brittleness to phrasing.** Dictation artefacts, abbreviations, institution-specific templates and translated reports all degrade recall silently.
- **Parse dependence.** Negation detection inherits the errors of a dependency parser trained on general-domain text.
- **No compositional reasoning.** Rules struggle with scope (“no focal consolidation, effusion, or pneumothorax”), comparison language, and differential lists.
- **Single-domain by construction.** Porting a rule set from chest radiographs to abdominal CT is effectively a rewrite.
- **No modifiers.** Rule systems return presence. Lobe, short-axis measurement and growth-versus-prior are outside what a lexicon-plus-negation design does.

3.3 RadGraph-based extraction

RadGraph reframes labelling as information extraction: entities (anatomy and observations, tagged definitely present, uncertain, or definitely absent) linked by relations (`located_at` , `modify` , `suggestive_of`). A model emits this graph for an unseen report, and a post-processing layer aggregates it into project-specific labels.



WHY IT IS ATTRACTIVE

The advantage is economics and control. Inference runs on local hardware with a comparatively small model, so marginal cost per report approaches the cost of electricity: no per-token charge, no rate limit, no vendor dependency. Throughput is bounded only by available GPUs, latency is predictable, and the pipeline can run inside a hospital firewall with no PHI leaving the network. The output is also intrinsically grounded: every entity is a span in the source text, which makes labels auditable by construction. For high-volume chest radiography under a tight budget it remains a defensible baseline, and it is why this approach was our starting point.

Where it falls short.

- **A graph is not a label.** Turning `pneumothorax [uncertain] located_at right apex` into the boolean or ordinal a downstream model consumes requires a hand-written aggregation layer. That layer accumulates heuristics, is where most end-to-end errors originate, and must be rebuilt for every new schema, reintroducing exactly the rule-authoring burden that motivated leaving rule-based systems.
- **Schema frozen at training time.** Severity, size, interval change, follow-up recommendation and clinical actionability are not first-class citizens, so they must be recovered imperfectly from `modify` relations, or not at all.
- **Domain shift.** Performance degrades on other modalities, body regions, institutional templates and report styles. Extending coverage requires new annotated data and retraining, not prompt changes.
- **Three-way certainty is too coarse.** Collapsing radiological hedging into present / uncertain / absent discards the gradation between “cannot be excluded” and “probably represents”.
- **No coverage signal.** RadGraph cannot distinguish “the radiologist called the lungs clear” from “the radiologist never mentioned the lungs”: the difference between a real negative and a missing value.
- **Weak on discourse.** Coreference, comparison to priors, wide-scope negation and section-aware weighting are outside what a span-and-relation model reliably captures.
- **Operational friction.** Credentialed data access, model hosting, GPU capacity and version pinning are real costs even though they are not per-token costs.

3.4 Free-form LLM prompting: the immediate predecessor

Our previous production system replaced the RadGraph aggregation layer with direct LLM prompting: a prompt describing the taxonomy and desired output format, concatenated with the report, sent to a frontier model, and the reply parsed into labels. This was a substantial accuracy improvement over both rule-based and graph-based extraction, particularly on negation, hedging and comparison language, and it made schema changes cheap. The first full pass over the corpus was produced this way. At production scale it exhibited three failure modes, plus a fourth observation that pointed to the fix.

3.4.1 Cost

Synchronous, per-report requests with long instruction preambles scale badly. The instruction block (taxonomy definitions, per-finding inclusion and exclusion criteria, the full modifier vocabulary for all 25 findings, edge-case rules, output format specification, few-shot examples) dominated a request averaging **~18,700 input tokens per report**, several times the length of a typical report. Most of every request was therefore redundant instruction, re-sent for every one of N reports, and interactive-tier pricing offers no discount for work with no latency requirement. A single full pass lands near **\$1,800**, and every schema revision costs that again.

3.4.2 Hallucination

With free-form generation the model is unconstrained in what it may assert. Observed failures include emitting findings not in the report (typically plausible co-occurrences, such as effusion alongside consolidation); inventing label values outside the permitted enumeration; fabricating measurements and laterality where the report is silent; over-confidently resolving hedged statements into positives; promoting a *Clinical History* mention into an imaging finding; and drifting into narrative explanation instead of labelling. These errors are pernicious because they are *plausible*: they pass casual review and surface only as unexplained label noise in downstream training.

3.4.3 Output parsing errors

Asking a model to “return JSON” is a request, not a guarantee. Roughly **5% of responses failed parsing on the first attempt**. A **label-normalisation pipeline of 20 repair rules** had to be maintained purely to fix output before it could be written to Parquet: it mapped `not_reported` and `unreported` to `not reported`; `unremarkable`, `within normal limits` and `not present` to `absent`; `unchanged` and `unchanged?` to a stability value; and the empty string to `uncertain`. Five further rules handled malformed *structure*, where the model

returned a JSON Schema fragment instead of a value (`{"value": "absent"} , {"const": "present"} , {"enum": "absent"}`) or lists and unresolvable structures that had to be defaulted to `uncertain` . Every one of those rules is a value the model was told not to emit and emitted anyway.

LABEL-NORMALISATION REPAIR RULES, PREVIOUS SYSTEM

<code>not_reported, unreported</code>	→ <code>not reported</code>
<code>unremarkable, within normal limits,</code>	
<code>not present</code>	→ <code>absent</code>
<code>unchanged, unchanged?</code>	→ <code>a stability value</code>
<code>"" (the empty string)</code>	→ <code>uncertain</code>
<code>{"value": "absent"} {"const": "present"}</code>	
<code>{"enum": "absent"}, lists, unresolvable</code>	→ <code>uncertain</code>

Beyond these, responses arrived wrapped in markdown fences, with trailing commentary, single quotes, unescaped characters in free-text fields, and inconsistent nesting between calls. Each required defensive parsing, retry logic and a quarantine path; and retries multiply cost while pushing a residue of silently dropped reports into the dataset. Because the parser must anticipate the model's stylistic variation, it is itself a source of bugs and version-to-version regressions.

3.4.4 The key observation: the schema *is* the prompt

DESIGN PREMISE

Much of the length and fragility of these prompts came from describing, in prose, what a machine-readable type definition expresses precisely: which fields exist, which are required, what their types are, and what values each enumerated field may take. Once the output format is declared as a schema the decoder must respect, the natural-language prompt no longer carries that burden. It shrinks to task framing and genuine clinical edge-case guidance, becomes far easier to review, and stops being the single artefact on which correctness of both content *and* format depends. This is the design premise of the new system.

3.5 Summary of the current state

Approach	Cost/report	Throughput	Modifiers	Schema flex.	Output reliability
Manual annotation	Very high	Very low	Full	High	High
Rule-based (CheXpert / NegBio)	Negligible	High	None	Very low	Deterministic
RadGraph + aggregation	Low	High	Partial	Low	Graph ≠ label
Free-form LLM prompting	High	Rate-limited	Unvalidated	High	Parsing failures
Schema-constrained, two-phase, batched	Low	High	Type-checked	High	Guaranteed

Table 1. Labelling approaches compared. Accuracy on hedged language is low-to-moderate for rule-based and RadGraph systems, and high for both LLM approaches.

04

The solution

We keep the component of the previous system that worked (the reasoning ability of an LLM over free clinical text) and replace the parts that did not: the shape of the taxonomy, the pricing tier, the request pattern, the request *count*, and the unconstrained generation of output format.

4.1
Flat taxonomy

4.2
Schema-first definition

4.3
Constrained decoding

4.4
Batching and prefix caching

4.5
Two-phase extraction

4.1 A flat, untiered label taxonomy

This is the change with the largest measured effect on accuracy, and it is a subtraction rather than an addition.

What the earlier design did. The first schema stratified the 25 findings into three reporting tiers according to how reliably radiologists document each one, and gave each tier its own rule for what an unmentioned finding meant. For reliably documented findings, silence meant `absent` . For anatomy-gated findings, silence meant `absent` *only if* the relevant organ had been described as normal, and `not_reported` otherwise, with a per-finding table of gate anatomy and gate phrases to decide which. For incidental findings, silence always meant `not_reported` . A fifth label value, for findings described only as stable or unchanged from prior, sat across all three tiers. The design was clinically well motivated: radiologists genuinely do document a pneumothorax more reliably than aortic atherosclerosis.

Why we removed it. Three measured problems, in increasing order of seriousness:

- 1. **The rule could not be stated in one sentence.** Assigning a label required knowing the finding’s tier, then, for one tier, consulting a gate-anatomy table and matching gate phrases. Annotators and the model applied that chain differently, and differently *for different findings*, so the disagreement was not a uniform bias correctable downstream.
- 2. **It was the bulk of the error budget.** The largest off-diagonal cells in the confusion matrix were `absent` ↔ `not_reported` confusions, concentrated on exactly the findings where the tier rule was most conditional. Genuine clinical errors were an order of magnitude rarer.
- 3. **It was unstable, not merely wrong.** In three-run repeat testing, the dominant disagreement was a finding flipping between `absent` and `not_reported` across identical runs, worst on anatomy-gated findings. The model was consistent about what it could see and inconsistent about how to apply a conditional rule to what it could not.

What replaced it. One rule, applied identically to all 25 findings:

If the report asserts the finding → `present` . If it negates the finding, or describes the relevant anatomy as normal → `absent` . If it hedges → `uncertain` . If neither the finding nor its anatomy is addressed → `not_reported` .

Asserts <code>present</code>	Negates, or anatomy normal <code>absent</code>	Hedges <code>uncertain</code>	Neither addressed <code>not_reported</code>
---------------------------------	---	----------------------------------	--

The `absent` / `not_reported` distinction is retained, because it separates a real negative from a missing value and downstream consumers need it. What is gone is the idea that the rule for drawing that line should depend on which finding is being labelled. There is no tier lookup, no gate-anatomy table, and no per-finding branch: in the schema, in the annotation guideline, or in post-processing. Measured on the same held-out set, the flat taxonomy raised overall accuracy from **0.89 to 0.96** and three-run agreement by about **5 points**.



4.2 Schema-first task definition

The taxonomy is defined once as an explicit, versioned Pydantic schema (field names, types, required/optional status, and closed enumerations for every categorical field) rather than described in prose inside a prompt. The schema is the contract between the labelling service and every downstream consumer: code-reviewable, diffable, testable. A taxonomy change is a change to a typed artefact with a version number, not an edit to a paragraph.

Crucially, the *clinical* definition of each finding also lives in the schema, as the field description, carrying inclusion and, more importantly, exclusion criteria in the same place as its type:

```
air_space_opacity: pulmonary ground-glass opacity, consolidation, infiltrate, airspace disease, air bronchograms, crazy-paving, halo/reversed halo sign, or tree-in-bud pattern. Do not count atelectasis, pulmonary nodule/mass, isolated septal thickening, fibrosis, honeycombing, emphysema, or scar.
```

This directly addresses Section 3.4.4. The system prompt drops from a multi-thousand-token taxonomy recitation to roughly twenty lines of task framing, evidence rules (“use only Findings and Impression; never Clinical History”) and anti-hallucination constraints, because the structural *and* definitional specification has moved into the schema where it belongs.

4.3 Structured output via constrained decoding

The schema is enforced at generation time: rather than asking the model to produce well-formed output and validating afterwards, the decoder is restricted to tokens that keep the output valid under the schema. In practice this is OpenAI structured outputs in strict mode, which requires `additionalProperties: false` on every nested object; our schema compiler injects that automatically so the Pydantic definition stays readable. The consequences:

1

Parsing errors are eliminated by construction, not reduced.

No fence-stripping, no regex recovery, no quarantine bucket, no retry-on-parse-failure loop, and no label-normalisation table (Section 3.4.3).

2

A large class of hallucination becomes unrepresentable.

A closed enumeration cannot take an invented value; a boolean cannot become a sentence; an unrequested finding has no field to occupy. The model can still be *wrong*, but no longer *out of range*, converting an open-ended error surface into a bounded one.

3

Downstream code simplifies.

Consumers program against a typed object; the defensive-parsing layer is deleted rather than maintained.

4

Modifiers become type-checked.

`max_size_mm` is a float or null; `lobe` is one of six enumerated values. An evaluation harness consumes these directly instead of regexing them out of prose.

5

Required fields force explicit decisions.

The model cannot silently omit a finding it is unsure about, which makes ambiguity visible and measurable instead of indistinguishable from absence.

4.4 Asynchronous batched inference with prefix caching

Report labelling is a bulk offline workload with no interactive latency requirement, and it is priced accordingly by submitting it through the asynchronous batch interface. Combined with caching of the invariant instruction prefix and right-sizing the model, this attacks cost from four directions:

Batch pricing

Applies a ~50% discount to work that tolerates a deferred completion window.

Prefix caching

Stops the shared instruction block from being billed at full rate once per report. Measured hit rates are **86% of input tokens single-call and 68% in the production two-phase configuration**, both at a **10× discount** (\$0.02 vs \$0.20 per million). The two-phase rate is lower because phase 2 assembles a per-report schema, so less of its prefix is invariant. Cache retention is set explicitly to 24 hours to match the batch window.

Model right-sizing

Made practical because the schema removes format-following from the model's burden (Section 4.6).

Higher effective throughput

Batch submission sidesteps the per-minute rate limits that bound synchronous loops. Measured synchronous throughput is ~12 s/report; a serial 600k pass at that rate is ~83 days.

4.5 Two-phase extraction: presence first, modifiers conditionally

The problem. The full schema is large because modifiers are large: 104 modifier fields across 25 findings, each with its own enumeration and description. Sending all of it on every request means paying for the entire modifier vocabulary on every report. But Section 2.2 established that only ~2 to 3 findings per report are positive, and **modifiers only exist for positive findings**. For roughly 90% of finding slots the modifier apparatus is dead weight in both directions: it inflates the input prompt and pads the output object with a long tail of explicit nulls.

The change. We split the single API call into two.

PHASE 1

Presence

The report is sent with a presence-only schema. Each of the 25 findings carries its clinical definition and exclusions and exactly one field, `label`. No modifiers are described and none may be emitted. The schema is sparse: a finding the report never addresses may be omitted entirely, and post-processing fills in `not_reported`.

PHASE 2

Modifiers

The phase-1 result is filtered to findings labelled `present` (or `abnormal`, for the regional findings). Only those findings' detail schemas (typically two or three of twenty-five) are assembled into a second request together with the report. The phase-2 prompt is explicitly forbidden from re-labelling presence.

Why it works. Phase 1 pays for 25 definitions but no modifiers; phase 2 pays for 2 to 3 modifier blocks and no definitions of the other 22. Neither call pays for the cross-product.

	Single call	Two-phase	Change
Input tokens / study	12,120	8,294	-32%
Output tokens / study	1,346	819	-39%
Cost / study (batch tier)	\$0.00111	\$0.00083	-25%
Overall accuracy	0.959	0.961	+0.2 pp

Table 2. Two-phase versus single-call extraction, same held-out set, same harness.

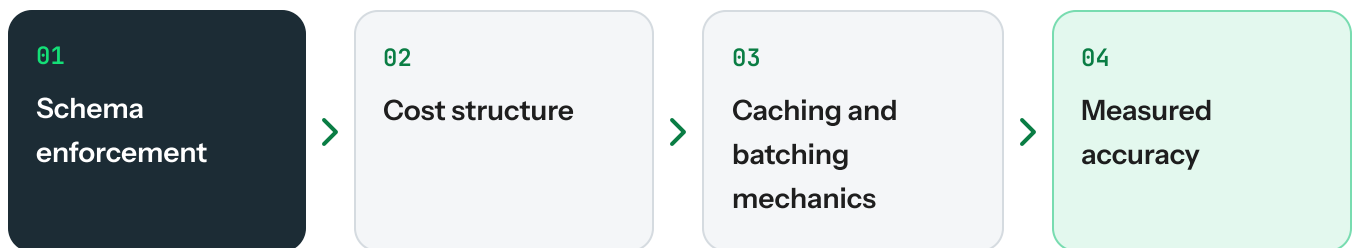
Accuracy does not degrade; it is marginally better, consistent with the phase-1 model having a much simpler job. Because output tokens dominate the bill on a small model with a heavily cached prompt (~75% of per-study cost in the single-call configuration), the 39% output reduction is where most of the saving comes from. On shorter reports and larger schemas the input reduction is greater; on early sample runs it reached 40 to 45%. A secondary benefit: the 4,000-token output truncation that silently dropped studies is no longer reachable: mean phase-1 output is ~460 tokens and phase-2 ~360.

The variant that did not work

We also tested chaining the two phases through the provider's conversation-continuation mechanism (`previous_response_id`), so phase 2 would not need the report re-sent. It is cheaper in raw tokens but worse on both axes that matter: accuracy fell to 0.948 and cost per study rose, because a continued response does not benefit from the same prefix-cache economics or batch discount as a fresh request against a cached instruction block. Re-sending the report against a cached prefix is cheaper than not re-sending it against an uncached conversation.

4.6 Model selection

The model was chosen on four criteria, in this order: **schema enforcement, cost structure, caching and batching mechanics, and measured accuracy**. Three of the four are properties of the platform rather than of the model's intelligence, which is the point. Once constrained decoding removes format-following from the model's job, this task does not need a frontier model, and the deciding factors become operational.



4.6.1 Schema enforcement: the disqualifying criterion

Our requirement is not “usually returns JSON”; it is “cannot return anything but a valid instance of this schema”, because that is what deletes the parsing layer, the normalisation table and the quarantine path.

OpenAI

STRICT

Provides this directly: a Pydantic model is passed as the response format and enforced during decoding in strict mode. Observed schema-invalid responses across evaluation and production runs: **zero**.

Gemini 2.5

HINT

Accepts a response schema, including through its OpenAI-compatible endpoint, but its JSON-schema support is a subset and our integration cannot set strict enforcement on that path: the schema is passed with `strict: false`, making it a strong hint rather than a decoder constraint. That reinstates post-hoc validation and retry, precisely the cost we are removing. Deeply nested optional objects, which our finding tree is made of, are the weakest part of that subset.

Claude

APPROXIMATED

Has no equivalent response-format parameter on the messages interface; our client explicitly warns that `response_format` is ignored for Anthropic models. Conformance must be approximated by expressing the schema as a tool definition or describing it in prose and validating afterwards. Both are workable engineering; neither gives the by-construction guarantee, and the prose route puts the schema back into the prompt.

4.6.2 Cost structure: output tokens dominate

With a heavily cached prompt, output tokens are 60 to 75% of the per-study bill (75% single-call, 61% two-phase), so output price per million is the number that decides the budget.

Model	Input	Cached	Output	Est. / 600k	Relative
gemini-2.5-flash-lite	\$0.10	\$0.01	\$0.40	~\$195	0.4×
gpt-5.4-nano (chosen)	\$0.20	\$0.02	\$1.25	~\$500	1.0×
gemini-2.5-flash	\$0.30	\$0.03	\$2.50	~\$900	1.8×
gpt-5.4-mini	\$0.75	\$0.075	\$4.50	~\$1,830	3.7×
claude-haiku-4.5 (Bedrock)	\$1.00	\$0.10	\$5.00	~\$2,190	4.4×
gemini-2.5-pro	\$1.25–2.50	\$0.13–0.25	\$10–15	~\$3,670	7.3×
claude-sonnet-4.5 (Bedrock)	\$3.00	\$0.30	\$15.00	~\$6,580	13.2×

Table 3. List prices per million tokens, and the estimated cost of one full pass over a ~600,000-study archive holding the *measured* two-phase token profile constant (8,294 input / 5,643 cached / 819 output per study) and applying each provider’s batch discount. The `gemini-2.5-pro` figure uses the low end of its published range; at the high end it is roughly \$4,900.

The only cheaper option is `gemini-2.5-flash-lite`, and it is cheaper by less than the value of the guarantee it cannot provide: at that tier we could not obtain strict schema conformance, so the saving would be spent on validation, retries and a residual drop rate. Every Claude tier is at least 4× the chosen configuration; the cheapest costs more than four full relabelling passes.

4.6.3 Caching and batching mechanics

These features have to *compose*, which is a stronger requirement than each existing. OpenAI’s prefix cache is automatic, applies a 10× discount, and, decisively, accepts an explicit 24-hour retention setting matching the batch window. Anthropic’s cache is explicit (`cache_control` breakpoints), carries a **1.25× write premium**, and defaults to a **5-minute TTL**, with a 1-hour TTL at additional cost; a workload spread across a multi-hour batch window gets very little from a 5-minute cache, and our measured hit rates depend on long retention. Gemini’s implicit caching has a minimum-token threshold and no comparable retention control. All three vendors offer ~50% batch discounts, but the differentiator is whether batch composes with *both* strict structured output and long-retention caching. On the OpenAI path it does; on the other two at least one leg is missing or degraded, and the arithmetic above assumes the best case for them. Batch submission also removes per-minute request ceilings from the critical path, which matters more than the discount for a 600k backfill, given the 83-day serial alternative.

4.6.4 Measured accuracy

At the nano tier with schema constraint and the flat taxonomy, overall accuracy is **0.96**, indistinguishable from what larger, more expensive configurations achieved in the same harness. A larger sibling model gained about a point of single-run accuracy at ~2.7× the cost per call: a poor trade at archive scale, and one the two-phase split partially recovers for free. We did not run a full head-to-head bake-off against Gemini and Claude at matched schema constraint, because a matched comparison is not constructible: the constraint is the thing they do not offer. The honest statement is that the comparison was decided on schema enforcement and cost structure, and accuracy confirmed the chosen configuration was good enough to stop looking.

4.6.5 What would change this decision

If strict, decoder-level schema enforcement became available on Gemini Flash-Lite with long-retention caching, the cost arithmetic would favour switching. If the schema grew until a nano-tier model’s accuracy degraded materially, the mini tier at $\sim 3.7\times$ would be next. The pipeline is model-agnostic behind a single `call_llm` interface with a per-model pricing table, so this is a configuration change, not a rewrite.

4.7 What this changes, end to end

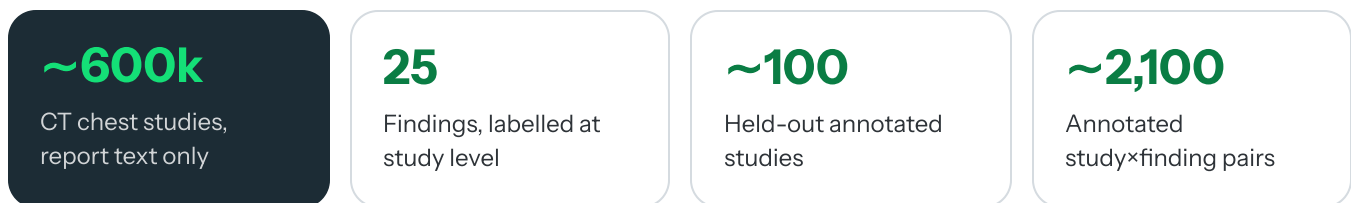
Failure mode of the previous system	Mechanism that resolves it
Dominant label confusion and run-to-run instability	Flat, untiered taxonomy; one rule for every finding (§4.1)
Not cost-effective	Batch pricing + prefix caching + model right-sizing (§4.4)
Paying for modifiers on negative findings	Two-phase presence-then-modifiers extraction (§4.5)
Hallucination	Closed enumerations and required fields enforced during decoding (§4.3)
Output parsing errors and normalisation tables	Constrained decoding guarantees schema-valid output (§4.3)
Output truncation dropping studies	Shorter per-call output; cap no longer reachable (§4.5)
Prompt fragility and length	Specification moved from prose into the schema (§4.2)
Rate-limited throughput	Asynchronous batch submission (§4.4)
Rigid taxonomy (RadGraph legacy)	Versioned schema, editable without retraining (§4.2)

Table 4. Each previous failure mode and the mechanism that removes it.

Method

5.1 Data

The reference deployment is a de-identified CT chest report corpus of ~600,000 studies, labelled at study level from report text only (no image input), across 25 findings. As established in Section 2.2, ~2.3 of 25 findings are positive per study on average, with a highly skewed distribution: the most common finding is positive in roughly half of studies, and six findings have positive prevalence below 1%. The held-out evaluation set is ~100 studies annotated by clinical reviewers, covering all 25 findings per study wherever the reviewer could reach a determination: ~2,100 annotated study×finding pairs. This set is small; it is the binding constraint on the confidence intervals in Section 7 and is discussed in Section 9.



5.2 Label schema design

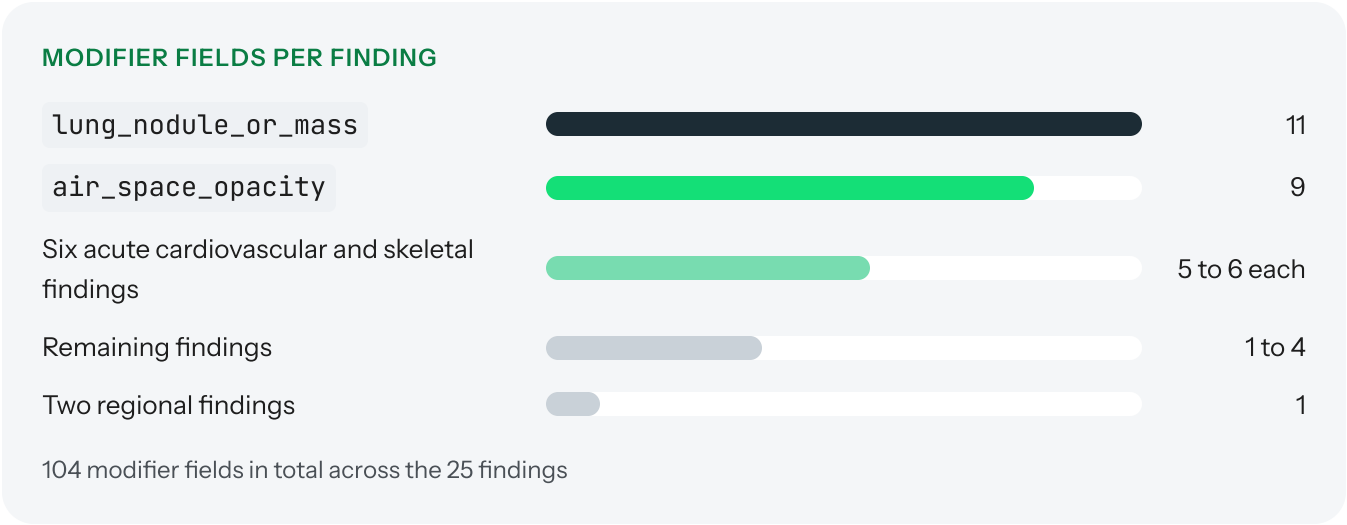
Every clinical finding uses the same four-value label set under the same rule (Section 4.1): `present` where the report clearly asserts the finding; `absent` where it explicitly negates the finding or describes the relevant anatomy as normal or unremarkable; `not_reported` where neither the finding nor its relevant anatomy is addressed; and `uncertain` for hedged language (“possible”, “suspicious for”, “cannot exclude”, “may represent”).

The two **regional findings**, `neck_base` and `upper_abdomen`, use a two-value set instead (`normal` and `abnormal`) because the question asked of them is not “is finding X present” but “is this region normal”. (`not_reported` remains available for a region the report never addresses; the decision axis when the region *is* addressed is strictly binary.)

The rule is uniform across all 25 findings. There are no reporting tiers, no gate-anatomy table, and no per-finding branch determining what silence means. There is also no fifth value for stable findings; interval change is a modifier on the positive finding, not a label.

Findings and modifiers. All 25 findings are retained, carrying **104 modifier fields** between them in the phase-2 detail schemas. The distribution is uneven and tracks clinical complexity: `lung_nodule_or_mass` is the heaviest at 11 fields (a per-lesion list covering type,

attenuation, lobe, size, slice, margin, pleural tag, volume, growth, malignancy suspicion and recommendation), followed by `air_space_opacity` at 9 (lobar distribution, laterality, opacity type and the crazy-paving / halo / tree-in-bud pattern flags). Six findings (the acute cardiovascular and skeletal ones, including `acute_pulmonary_embolism`, `acute_rib_fracture`, `pleural_effusion` and `pneumothorax`) carry 5 to 6 fields each, typically laterality, severity, anatomical level and an acuity or complication flag. The remaining findings carry 1 to 4, and the two regional findings carry a single free-text abnormality description. This total of 104 is the quantity the two-phase split is designed not to pay for on negative findings.



Exclusions are first-class. With the tier machinery gone, the largest source of systematic error is confusion between *adjacent findings*, so every definition carries explicit negative criteria alongside the positive ones. `air_space_opacity` excludes atelectasis, nodule/mass, isolated septal thickening, fibrosis, honeycombing, emphysema and scar; `pneumothorax` excludes pneumomediastinum, subcutaneous emphysema, pneumopericardium, bullae, blebs and cysts; `acute_rib_fracture` excludes healed fractures, callus and congenital variants; `bone_lesion` excludes fractures, degenerative change, Schmorl nodes and ordinary bone islands. This is now where the schema-authoring effort goes.

5.3 Prompt construction

Two system prompts, both short, because Section 4.2 moved the taxonomy into the schema.

The **phase-1 prompt** establishes the role (strict information-extraction system for CT chest reports); that schema field descriptions are the source of truth for definitions, exclusions and allowed labels; that no finding categories outside the schema may be created; that a

mention in *Clinical History* is not an imaging finding; that only the presence label may be extracted, with measurements, laterality, severity, lesion attributes, locations and recommendations explicitly out of scope; that missing findings must not be inferred; that one finding category must never be converted into another; and that hedged wording maps to `uncertain` rather than `present` .

The **phase-2 prompt** additionally establishes that only categories present in the supplied response schema may be returned; that presence must not be re-labelled; that details must not be extracted for absent, normal or not-reported findings; that only *Findings* and *Impression* are evidence; that unstated attributes are null rather than guessed; and a set of finding-specific anti-hallucination rules learned from observed failures: do not duplicate a lesion; do not convert pneumonia, consolidation, infiltrate or ground-glass opacity into a nodule or mass; do not convert lymphadenopathy, pleural, mediastinal or chest-wall disease into a pulmonary lesion; do not infer a mass from the words “suspicious”, “malignant” or “tumour” unless the report explicitly names a pulmonary mass or gives a focal pulmonary lesion >30 mm.

5.4 Structured output and constrained decoding

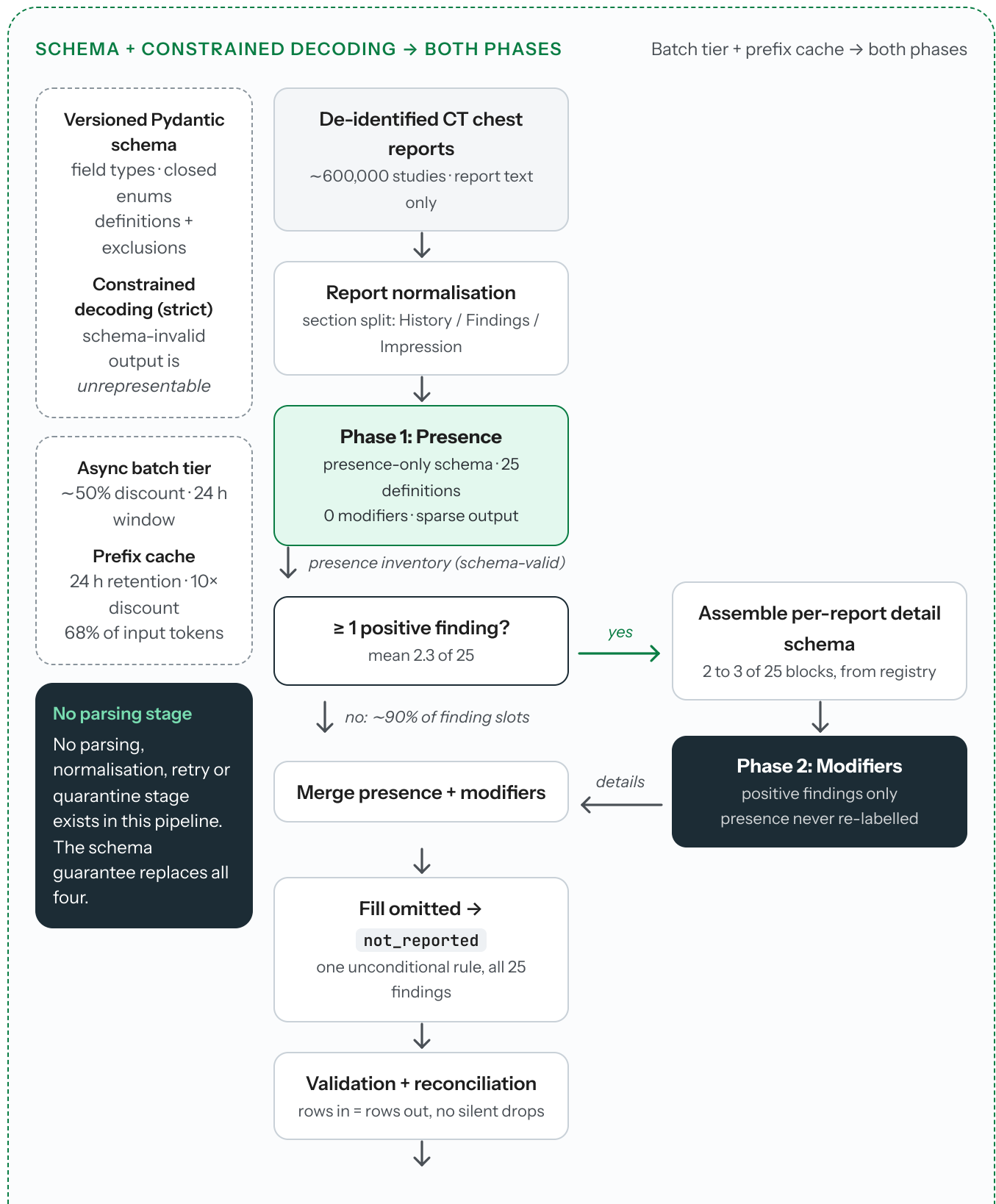
The Pydantic model is passed as the response format and enforced during decoding. Strict mode requires `additionalProperties: false` on every nested object; a schema-compilation step walks the generated JSON Schema and injects it, so the Pydantic source stays clean. Reasoning effort is set to `low` , measured to have minimal effect on accuracy or stability for this task while materially affecting output-token cost.

The phase-1 schema is deliberately **sparse**: each finding is `Optional` , and the model returns only findings that are positive, hedged, or explicitly negated. Omission is materially cheaper than an explicit object twenty-two times per report, and post-processing fills the gap with `not_reported` . Under the uniform rule this is a lossless transformation (omission has exactly one meaning for every finding), which was not true under the tiered design, and is a second reason the simplification paid for itself. The phase-2 schema is **assembled per report** from a registry of detail models, containing only the findings phase 1 marked positive.

5.5 Pipeline architecture

Figure 1 shows the production pipeline. Three properties are worth calling out. First, there is **no parsing stage**: the boxes that used to contain fence-stripping, regex recovery, value normalisation and quarantine are gone, replaced by the schema guarantee. Second, the

default-fill step is **one unconditional rule** rather than a per-finding branch, which is what the flat taxonomy bought at the code level. Third, **reconciliation is explicit**: the pipeline asserts that every input study produced an output row, so the silent-drop failure of Section 3.4.3 becomes a hard failure rather than a shortfall discovered later.



Labels (Parquet)

+ per-run cost/token manifest +
data card

Figure 1. The two-phase labelling pipeline. Phase 1 extracts presence for all 25 findings from a modifier-free schema; only studies with at least one positive finding proceed to phase 2, which receives a per-report schema containing just those findings' modifier blocks. Both phases run under strict constrained decoding on the asynchronous batch tier with a 24-hour prefix cache.

5.6 Quality control

Schema validation is structural and automatic (Section 5.4). Clinical validation uses the held-out annotated set, re-run against every schema version. For **stability sampling**, a sample is labelled three times independently and per-finding agreement measured; disagreement is triaged by pattern, since an `absent` / `not_reported` flip indicates a *rule* ambiguity to be fixed in the schema description whereas a `present` / `absent` flip indicates genuine textual ambiguity or model error. Every release ships a **data card** recording label taxonomy, per-finding class distribution, known biases and out-of-scope uses, including the explicit statement that these are **weak labels, not radiologist ground truth**, and must not be used for clinical decision-making without independent radiologist review.

06

Experiments

All experiments run against the same held-out set of ~100 annotated studies (~2,100 study×finding decisions) with the same harness. Every run records accuracy, per-finding accuracy, prompt tokens, cached tokens, completion tokens, wall-clock time and computed cost, so cost and accuracy always come from the same run. Unless stated otherwise the configuration is `gpt-5.4-nano`, reasoning effort `low`, structured output in strict mode, prefix caching with 24-hour retention, batch tier, flat taxonomy.

ID	Configuration	Description
A	Free-form prompting	Prose taxonomy in the prompt, JSON requested not enforced, synchronous, no caching. The previous production system.

ID	Configuration	Description
B	Single-call structured	Full schema (labels + all modifiers) in one constrained-decoding call, cached, batched.
C	Two-phase structured	Presence call, then conditional modifier call. Cached, batched. Production configuration.
D	Two-phase, chained	As C, but phase 2 continues the phase-1 response via <code>previous_response_id</code> rather than re-sending the report.

Table 5. Configurations compared. B and C were run at the batch tier; D was measured at the synchronous tier and is normalised to the batch tier wherever costs are compared.

Metrics. Overall accuracy across all annotated study×finding pairs is the headline number. Per-finding accuracy is reported because the aggregate hides a wide spread, and per-label precision, recall and F1 because the distribution is dominated by `absent` and aggregate accuracy is correspondingly generous. A confusion matrix over the label set separates clinically material errors from rule-boundary effects. **Three-run agreement** (the fraction of items on which three independent runs produce the same label) is a *stability* measure distinct from accuracy, and it is the one that predicts whether a relabelling pass will churn the training set. Schema validity rate, per-study cost and token profile, and wall-clock throughput complete the set.

Ablations. Flat versus tiered taxonomy is the primary accuracy result: replacing three per-finding reporting tiers with one uniform rule and dropping the fifth label value is worth **+7 points overall accuracy and +5 points three-run agreement**. Two-phase versus single call (C vs B) is the primary cost result, and chained versus re-sent phase 2 (D vs C) the counter-intuitive one; both are reported in Section 7.2. On **strict enforcement**, under the tiered taxonomy strict-on gained ~1 to 2 points of single-run accuracy and lost ~7 points of three-run agreement; under the uniform rule that stability penalty largely disappears, so strict is now on unconditionally. **Model tier:** nano versus the next tier up is ~1 point of accuracy at ~2.7× cost. **Reasoning effort:** none versus low has minimal effect, retained at low for output-token economy. **Prefix caching** has no accuracy effect by construction and serves 68% of input tokens from cache at a 10× discount in production.

Results

7.1 Labelling accuracy

Across ~100 studies in production configuration C, overall accuracy is **0.96**, with weighted precision, recall and F1 all 0.96, and **zero** schema-invalid responses. The taxonomy ablation is the single largest accuracy change in the project: the tiered five-value design scored **0.89** against **0.96** for the flat four-value rule. Three-run agreement improved by **5 points** over the same comparison. Agreement is a separate axis from accuracy (it measures whether three identical runs return the same label, not whether that label is correct), so the two gains are independent quantities rather than two estimates of one effect.

Label	Accuracy	Precision	Recall	F1
absent	0.96	0.99	0.96	0.97
not_reported	0.95	0.89	0.95	0.92
present	0.94	0.95	0.94	0.94
abnormal (regional)	0.99	0.94	0.99	0.96
normal (regional)	0.91	0.89	0.91	0.90
uncertain	0.76	0.71	0.76	0.72

Table 6. Performance by label value, production configuration.

uncertain remains the weakest category, and with only 15 annotated instances its support is far too small to draw a firm conclusion from. By finding type, the 23 clinical findings average 0.96 and the two regional findings 0.97.

Per-finding accuracy spans 0.82 to 1.00. The critical safety findings are at or near ceiling: **pneumothorax**, **pleural_effusion**, **pericardial_effusion** and **acute_rib_fracture** all reach 1.00, with **aortic_aneurysm** and **bone_lesion** at 0.99 and **copd_emphysema** and **neck_base** at 0.98. A broad middle band from 0.94 to 0.97 covers the bronchiectasis, lymphadenopathy, atelectasis, cardiomegaly, nodule, vertebral-fracture, air-space-opacity, valve-calcification and septal-thickening findings. Three findings sit below 0.90: **coronary_artery_calcifications** at 0.90, **acute_pulmonary_embolism** at 0.88,

`chronic_pulmonary_embolism` at 0.86 and `aortic_atherosclerosis` at 0.82. Of these, the pulmonary embolism figures rest on 9 to 25 annotated pairs each and are not stable estimates; the two cardiovascular findings are genuinely weak, and they are the findings radiologists document least consistently, where the `absent` / `not_reported` line is hardest to draw.

PER-FINDING ACCURACY, NAMED FINDINGS



7.2 Cost analysis

Config.	Input	Cached	Output	Cost/study	Acc.
A: free-form, sync, uncached	~18,700	0	~1,400	~\$0.0030	n/a
B: single-call structured	12,120	10,435 (86%)	1,346	\$0.00111	0.959
C: two-phase (production)	8,294	5,643 (68%)	819	\$0.00083	0.961
D: two-phase, chained	8,375	5,765	926	\$0.00090+	0.948

Table 7. Per-study token and cost profile, same held-out set, batch tier. Cached tokens are a subset of input tokens, billed at the 10× discount. † D was measured at the synchronous tier (\$0.0018/study) and is normalised to the batch tier for comparability.

Where the money goes. For configuration B, per study: uncached input \$0.00034, cached input \$0.00021, output \$0.00168; **output tokens are ~75% of the bill**. In configuration C, with fewer output tokens, output is ~61%. This is why the two-phase split, which cuts output by 39%, saves more than its 32% input reduction would suggest, and why output price per million is the deciding number in Section 4.6.2.

Projected over ~600,000 studies for one complete pass, configuration A costs ~\$1,800, B ~\$670 (2.7× cheaper) and C ~\$500 (**3.6× cheaper**). The practical significance is not the ~\$1,300 saved on one pass. It is that at ~\$500 a pass, **relabelling the entire archive after a schema revision is a routine operation**, which is what makes the versioned-schema design usable rather than aspirational, and what made the taxonomy change in Section 4.1 affordable to test on real data rather than argue about in a design review.

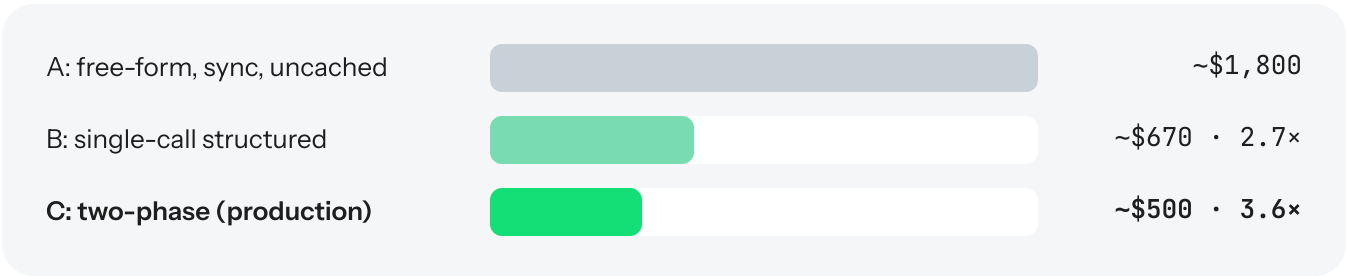


Figure 2. Projected cost of one complete pass over ~600,000 studies, by configuration, drawn to scale.

The chained-phase-2 result (D) is the counter-intuitive one. Re-using the conversation via `previous_response_id` avoids re-sending the report and does reduce raw phase-2 input tokens, but the continued turn does not benefit from the same prefix-cache economics, so at matched pricing tier it is *more* expensive than re-sending the report against a cached prefix, and it costs over a point of accuracy. Configuration C re-sends the report deliberately.

7.3 Throughput and output validity

Measured synchronous serial throughput is ~12 s/report, which projects to ~83 days for 600k studies; synchronous concurrent execution is rate-limit bound and lands somewhere between days and weeks depending on the limit. Batch submission completes a full pass **within the 24-hour window**. Latency per report is irrelevant to this workload and is not optimised; what matters is that a full archive pass completes inside a day, which makes schema iteration a same-week loop rather than a quarterly project.



Metric	Free-form (A)	Structured (B / C)
Responses failing format/parse on first attempt	~5%	0

Metric	Free-form (A)	Structured (B / C)
Value / structure repair rules required	20	0
Studies dropped to output truncation	~0.01%	0
Retry-on-parse-failure loop	Required	Removed
Quarantine path	Required	Removed

Table 8. Output validity and failure rates.

The drop rate under the free-form system is the number to keep in mind when the guarantee is described as a small win. It is small per study and it is silent, which is the worst combination for a training-set label source: studies vanished from the archive and the pipeline reported success.

7.4 Error analysis

Replacing the tier system with one uniform rule changed the shape of the error profile, not just its size. The `absent` / `not_reported` confusion that used to dominate is still present (the distinction is retained), but it shrank sharply once both annotator and model applied the same rule to every finding. Residual error, in order of size:

1. **`absent` vs `not_reported` on inconsistently documented findings.** Still the largest single category, now concentrated on a handful of incidental cardiovascular findings: `aortic_atherosclerosis` (0.82) and `coronary_artery_calcifications` (0.90) are the clearest cases. When a radiologist does not mention aortic plaque, the rule says `not_reported`, and an annotator reading the same report may treat a normal-aorta statement elsewhere as covering it.
2. **Adjacent-finding confusion.** The model assigns a real finding to the neighbouring class: air-space opacity versus atelectasis, consolidation versus pulmonary mass, hilar versus mediastinal nodal stations. This is a *definition* problem and it is addressable: every instance we triaged was fixed by tightening the exclusion list in the schema description rather than by changing the model or the prompt.

3. **Hedging-threshold mismatch.** The model's `present` / `uncertain` boundary does not match the annotators'. It over-calls `uncertain` on phrasing a reader treats as a soft positive ("likely represents", "compatible with"), showing up as `uncertain` precision of 0.71 against 15 annotated instances. With that support the honest conclusion is that the category is barely measurable, not that the model is bad at it.
4. **Genuinely ambiguous text.** A residual of positive/negative disagreements on reports where two readers would also disagree: findings mentioned only in a comparison sentence, or descriptions that stop just short of naming the finding. In three-run testing these are also the items the model flips on, the signature of real ambiguity rather than model error.
5. **Low-prevalence findings are not meaningfully evaluated.** `acute_aortic_injury`, `chronic_pulmonary_embolism` and `acute_pulmonary_embolism` have 9 to 25 annotated pairs each and near-zero positive prevalence; their reported accuracies move several points on a single item.

What is no longer in this list is worth stating explicitly: hallucinated findings, out-of-enumeration values, invented measurements, and unparseable output. Constrained decoding removed the first three as representable states and the fourth as a possibility.

FUTURE WORK: AN ORGAN-NORMALITY INVENTORY

Rather than asking the model to reason about anatomy gating (the conditional logic Section 4.1 removed), phase 1 can additionally return a flat inventory of organs and regions, each marked described-clear or not addressed. Post-processing then promotes `not_reported` to `absent` for every finding whose parent organ was described as clear. This keeps the model's job purely observational, moves the inferential step into deterministic code where it is auditable and cheap to revise, and should reduce both the residual `absent` / `not_reported` disagreement and the model's opportunity to hallucinate a coverage judgement.

Discussion

Simplify the labelling rule before optimising the extractor.

Seven points of accuracy came from making one rule apply to every finding, and no amount of prompt engineering, model upgrading or few-shot exempling would have recovered them; the model was not making a mistake, it was faithfully applying a conditional rule that annotators applied differently. The lesson is a specification one: a taxonomy whose labelling rule varies by category is not a richer label set, it is an unbudgeted disagreement generator, and it shows up as accuracy. Keep the distinctions consumers use; drop the conditionality in how you draw them.

The schema is the product.

The most durable result of this work is not a cost number; it is that the taxonomy became a versioned, typed, code-reviewed artefact rather than a paragraph inside a prompt. Everything else follows: the guarantee, the short prompt, the deleted parsing layer, the ability to diff two label versions, and the ability to hand a downstream team a type definition instead of a description. It is also what made the taxonomy change testable: flattening the rule was a schema diff and a \$500 re-run, not a project.

Structure buys more than accuracy: it buys a bounded error surface.

Constrained decoding did not make the model smarter. It made a whole class of failure unrepresentable. There is a meaningful engineering difference between “we reduced parse failures to 0.3%” and “parse failure is not a state this system can enter”, and the difference compounds: no retry logic, no quarantine, no normalisation table, no drift when the provider changes its default formatting.

Cost structure, not model quality, drove the platform decision.

With format-following offloaded to the decoder, this task runs on the cheapest tier of a frontier family. That inverts the usual selection criteria: the deciding factor became whether strict schema enforcement, long-retention prefix caching and batch discounting *compose on the same platform*. A 4 to 13× cost difference against Claude tiers, and 1.8 to 7× against Gemini tiers at comparable capability, is not a close call at 600,000 studies.

Exploit the prior.

The two-phase split is not a clever prompting trick; it is arithmetic applied to a property of the data. Reports are ~90% negative per finding slot and modifiers exist only for positives, so do not pay for modifiers until you know which findings are positive. The same reasoning generalises to any extraction task with a large conditional schema and a sparse positive rate, which is most clinical extraction.

Weak labels are a category, not a compromise.

These labels are for training and validating models, not for patient care. Labelled as such, at 0.96 accuracy and ~\$0.0008 per study, they are a good trade against radiologist time. Presented as ground truth, they would be a liability. The data card is part of the deliverable for that reason.

09

Limitations

01 Weak labels, not radiologist ground truth.

All archive labels are LLM-extracted from free text and are not clinically verified. They are suitable for model training, validation and research, and are explicitly out of scope for clinical decision-making without independent radiologist review and regulatory authorisation.

02 The uniform rule discards information about reporting reliability.

The same `absent / not_reported` boundary applies to every finding regardless of how consistently radiologists document it. That is what made the rule specifiable and stable, but the label set no longer encodes that a negative for a routinely reported finding is stronger evidence than a negative for an incidental one. Consumers who care must reintroduce it from finding-level priors.

03 `absent / not_reported` remains the largest residual disagreement,

concentrated on inconsistently documented incidental findings. Prevalence estimates for those findings should be treated as lower bounds.

04 Small evaluation set.

~100 studies and ~2,100 annotated pairs, with per-finding sample sizes from 9 to 96, so several per-finding figures carry wide confidence intervals and should be read as indicative. Expanding this set is the highest-value next investment.

05 Six findings have positive prevalence below 1%.

Extraction quality for these is essentially unmeasured, and downstream training on them will be constrained regardless of label quality.

06 `uncertain` is under-determined.

15 annotated instances. Do not build a decision rule on this label without further annotation.

07 Report-only.

Labels derive entirely from report text, never from images. Findings mentioned only in *Clinical History* are deliberately excluded, which can under-label studies where the radiologist references history without restating the finding.

08 Study-level independence.

Each study is labelled without its priors, so a stable chronic finding can appear as newly present across a patient's serial studies.

09 **Corpus prevalence is not population prevalence.**

Present/absent ratios do not mirror real-world CT chest prevalence, and deployment thresholds must be recalibrated.

10 **Single modality and region.**

Everything reported here is CT chest. The method is designed to port (the schema is the only domain-specific artefact), but the numbers do not.

11 **Vendor coupling.**

The by-construction validity guarantee depends on one provider's strict structured-output implementation. The pipeline abstracts the call site and the pricing table, but the guarantee itself does not port today (Section 4.6.5).

12 **Two-phase error propagation.**

A finding missed in phase 1 gets no phase-2 modifier extraction, so presence recall bounds modifier recall by construction. Measured accuracy did not degrade, but the dependency is structural and worth monitoring as the schema grows.

— 10

Conclusion

Structured labelling of radiology reports at archive scale is not primarily a modelling problem. Every generation of this system (rules, RadGraph, free-form prompting) was limited less by the extractor's reading ability than by what surrounded it: a hand-written aggregation layer, a frozen taxonomy, a prompt that had to specify both content and format, a parser that had to anticipate stylistic drift, and, most recently, a labelling rule that changed depending on which finding was being labelled.

Four changes removed those constraints. Flattening the taxonomy to one uniform rule over four label values deleted the specification ambiguity that accounted for the bulk of all errors, worth seven points of accuracy and five of stability. Moving the taxonomy into a versioned typed schema made the label set a reviewable artefact and shrank the prompt to clinical judgement. Enforcing that schema during decoding made invalid output unrepresentable,

deleting the parsing, normalisation, retry and quarantine layers along with the silent-drop rate. Splitting the request into a cheap presence phase and a conditional modifier phase exploited the fact that ~90% of finding slots are negative and carry no modifiers.

The measured result is 0.96 overall accuracy with zero schema-invalid responses, at roughly \$0.0008 per study on the batch tier: an estimated ~\$500 for a complete pass over a ~600,000-study archive, against ~\$1,800 for the free-form system it replaced. The residual error is dominated not by hallucination or misreading but by definitional edges (the `absent` / `not_reported` line on incidental findings, and confusion between clinically adjacent findings), both addressable by tightening the schema rather than changing the model.

The number that matters is not the total. It is the per-pass cost: at this price, relabelling the entire archive when the schema changes is a routine operation. That is what converts a label set from a fixed asset into a maintainable one, it is what allowed the taxonomy simplification to be settled by measurement rather than by argument, and it is the property everything else in this design was chosen to protect.

THE NUMBER THAT MATTERS

At this price, relabelling the entire archive when the schema changes is a routine operation.

REFERENCES

References

1. Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301-310, 2001.
2. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3462-3471, 2017.
3. Peng Y, Wang X, Lu L, Bagheri M, Summers RM, Lu Z. NegBio: A high-performance tool for negation and uncertainty detection in radiology reports. *AMIA Summits on Translational Science Proceedings*, 188-196, 2018. arXiv:1712.05898.

4. Irvin J, Rajpurkar P, Ko M, Yu Y, Ciurea-Illcus S, Chute C, et al. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):590–597, 2019.
5. Johnson AEW, Pollard TJ, Berkowitz SJ, Greenbaum NR, Lungren MP, Deng C, Mark RG, Horng S. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6:317, 2019.
6. Smit A, Jain S, Rajpurkar P, Pareek A, Ng AY, Lungren MP. CheXbert: Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. *Proceedings of EMNLP 2020*, 1500–1519. ACL Anthology 2020.emnlp-main.117.
7. Jain S, Agrawal A, Saporta A, Truong SQH, Duong DN, Bui T, et al. RadGraph: Extracting clinical entities and relations from radiology reports. *NeurIPS Datasets and Benchmarks Track*, 2021. arXiv:2106.14463.
8. Willard BT, Louf R. Efficient guided generation for large language models. arXiv:2307.09702, 2023.
9. Geng S, Josifoski M, Peyrard M, West R. Grammar-constrained decoding for structured NLP tasks without finetuning. *Proceedings of EMNLP 2023*. ACL Anthology 2023.emnlp-main.674.
10. OpenAI. Introducing Structured Outputs in the API, 2024. <https://openai.com/index/introducing-structured-outputs-in-the-api/>
11. OpenAI. Structured outputs, Batch API, and prompt caching: API documentation. <https://platform.openai.com/docs/guides/structured-outputs>, <https://platform.openai.com/docs/guides/batch>, <https://platform.openai.com/docs/guides/prompt-caching>
12. Anthropic. Prompt caching and the Message Batches API: Claude documentation. <https://docs.claude.com/en/docs/build-with-claude/prompt-caching>, <https://docs.claude.com/en/docs/build-with-claude/batch-processing>
13. Google. Gemini API: structured output, context caching, and batch mode. <https://ai.google.dev/gemini-api/docs/structured-output>, <https://ai.google.dev/gemini-api/docs/caching>
14. Pydantic. Data validation using Python type hints. <https://docs.pydantic.dev/>