

Decision Models Under Audit: A Comparative Study of Small Decision Models and Frontier LLMs for Structured Judgement

Jaimin Shah

Abstract

Small, purpose-built decision models are increasingly placed inside LLM pipelines to answer bounded questions: does a clause apply, which route should a case take, is this input safe. We compare four such deciders on two tasks. **Jev** is a hosted System One model. **Kev-0.8B** is a LoRA plus pointer head on Qwen3.5-0.8B. **Laya English** is a ModernBERT-large option scorer. **GPT-6 Luna** is a general-purpose LLM used as a decision model.

Experiment 1 is a controlled support-routing benchmark: 30 frozen messages, each scored under 3 option orders. Experiment 2 is an insurance-underwriting test harness: 8 synthetic applications, a 10-clause rulebook, PII and prompt-injection screens and a deterministic gate, with labels derived from the harness's own rules. It tests how deciders behave inside a workflow, not whether the workflow underwrites correctly.

Scored on their own decisions, before the harness's final agreement step, Jev's answers produced the rule-derived outcome on 8 of 8 applications, GPT-6 Luna's on 7, Laya's on 6 and Kev's on 4. On routing, Luna scored 30/30 and Jev 28/30. Jev selected the harness's clauses most accurately (F1 0.98 vs 0.90 for Luna). Kev matched Jev on clear and boundary routing (20/20), but its clause selection was close to chance (AUC 0.60), and on its own it approved an application with no claims history. Laya never chose the abstention route (0/90) and answered yes to every clause. At published prices, Jev's decision calls cost \$0.19 per 1,000 applications against \$0.42 for Luna.

We report observed failures separately from candidate explanations. The explanations involve architecture, pretraining, probability scaling, input design and an interface translation, and they remain hypotheses until ablations, repeated runs and larger samples test them. We close with a provisional use-case matrix and the cost of each decider at published API prices.

1. Introduction

A decision model returns a probability over a fixed set of answers instead of free text. Two answer shapes matter here: a **noul** (a yes-probability for one proposition) and a **choice** (a distribution over labelled options). This interface makes a model's output checkable by code, which is why it is attractive in regulated workflows such as insurance underwriting.

The interface hides very different machines. Some deciders are encoders that score how well each option matches the input. Some are small decoders with a pointer head over option markers. Some are frontier LLMs served through a decision endpoint. Published leaderboards rarely tell a practitioner which of these to put at which step of a pipeline, or why one fails where another succeeds.

This paper contributes:

1. A paired evaluation of four deciders on a frozen routing benchmark (Experiment 1) and on an insurance-underwriting test harness (Experiment 2), with gold labels fixed before inference.
2. Metrics beyond accuracy: option-order stability, calibration (ECE, Brier), score separation for rule selection (AUC), screening errors, per-call latency and cost.
3. A catalogue of observed failures, kept separate from candidate explanations, with the data each explanation needs before it can be accepted.
4. A provisional use-case matrix stating which decider fits which step, and what each costs per decision.

2. Models under study

The four deciders differ in how they reach an answer. Laya is an encoder that scores each option against the input. Kev is a small decoder with a pointer head that scores option markers. Jev and GPT-6 Luna are hosted models whose architectures are not disclosed.

Model	Family	Architecture (as published)	Size	Probability source	Serving in this study
Laya English	Encoder scorer	ModernBERT-large encoder → 2 extra transformer layers → option-scoring network	421.3 M params, 0.79 GiB	Native softmax, temperature 1.760	Local server, port 8010; english root checkpoint
Kev-0.8B	Decoder + pointer	Qwen3.5-0.8B-Base + LoRA adapter (43 MB) + pointer head over option markers (2.1 MB)	752.9 M params, 1.69 GiB	Native softmax, temperature 2.406	MLX server on Apple Silicon, port 8008
Jev (jev-latest)	Hosted decision model	Not disclosed	Not disclosed	Native System One noul and choice	TypeSafe API
GPT-6 Luna	General LLM	Not disclosed	Not disclosed	Native probabilities via the OpenAI Decisions API	OpenAI API

Model	Family	Architecture (as published)	Size	Probability source	Serving in this study
				(both experiments)	

Laya and Kev figures come from the pinned checkpoints and process measurements in the study repository. For Jev and GPT-6 Luna we treat the model as a black box. Any explanation of their behaviour (§7) therefore rests on their interface and outputs, not on their internals.

Two interface facts matter for the analysis. First, Laya and Kev score all options in one forward pass and generate no intermediate text. Second, the Decisions API has no separate criteria field for a yes/no question. The pipeline's adapter therefore folds each yes/no question and its yes and no criteria into one instruction string for GPT-6 Luna, while Jev, Kev and Laya receive the question and its criteria as separate fields (§6.2, F2).

3. Experimental setup

Both experiments froze inputs and gold labels before any model was run. In both, only the decision model changes between conditions; every other component is held fixed.

3.1 Experiment 1: support-routing benchmark

Thirty synthetic English customer messages were authored with an expected route: 10 *clear*, 10 *ambiguous* (missing evidence, off-topic, or two issues with no stated priority) and 10 *boundary* (negation, stated priority, irrelevant dates, an embedded instruction). The four routes are *billing*, *technical*, *sales* and *clarify*. *clarify* is an explicit abstention class, not a confidence threshold.

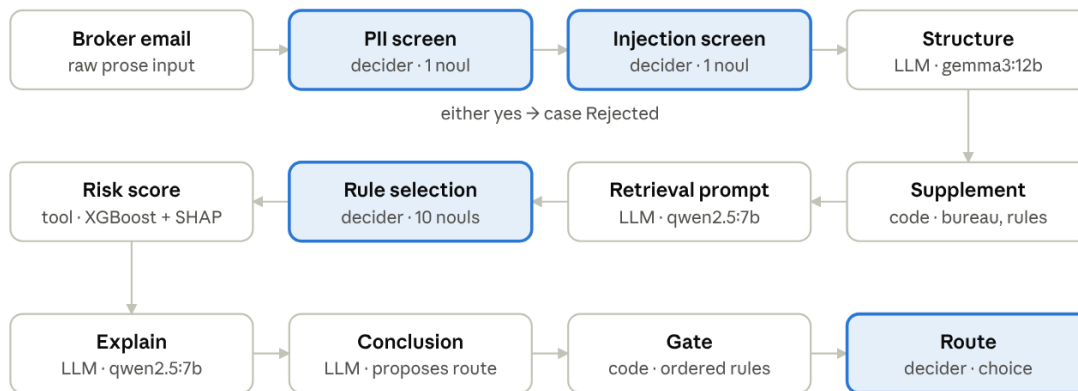
The routing policy is placed in the state ahead of each message, and each model answers one choice question. Every case is scored under three option orders, giving 90 decisions per model over 30 unique cases. The case file's SHA-256 is fixed (f8ae9a54...). We report the live four-model run.

3.2 Experiment 2: underwriting harness

A broker emails a medical-insurance application as prose. The pipeline extracts fields, attaches claims-bureau history and a 10-clause rulebook, predicts an annual medical charge with a fixed XGBoost model, and routes the file to APPROVE, REQUEST_INFO, REFER or DENY. Eight synthetic applications each target one gate path: a clean file, a high-charge smoker, cover above the cap, an incomplete file with three evidence gaps, a missing claims history, a scoring outage, a prompt injection and an email carrying an SSN, race and caste.

Decision models are called at four points; code owns the decision

☐ decision-model call (the component varied across runs)



Route runs only after the gate returns standard_rate or underwriter_review. If the route decider and the conclusion LLM agree, their route replaces the gate's; if they disagree, the case is referred to a person.

Underwriting pipeline, insurance-underwriting-agent @ 8b3a47b · 12 stages, 4 decision-model calls

The LLM stages (gemma3:12b for extraction, qwen2.5:7b elsewhere) were the same in every run. Only the decision model was swapped: Jev, Kev, Laya and GPT-6 Luna, with the latest full run reported for each. GPT-6 Luna was called through OpenAI's Decisions API, as in Experiment 1.

3.3 What the insurance example validates

Experiment 2 is a test harness for decision models, not an evaluation of underwriting. It shows how each decider's answers flow through a fixed, rule-based workflow. It says nothing about whether the workflow prices or accepts applicants correctly.

Experiment 2 tests	Experiment 2 does not test
Whether each decider's screening, clause and route answers match labels derived from the pipeline's own rulebook and gate	Whether those rules, thresholds or labels are sound underwriting
How a decider's answers propagate through a fixed gate to an outcome	Premium adequacy or pricing accuracy
Latency and token cost of each decider inside a multi-step workflow	Eligibility or acceptance decisions on real applicants
	Regulatory compliance or fairness
	Performance on real broker submissions

Three properties of the harness set these limits:

- **Labels are rule-derived.** The labels in `gold.json` are “code-owned”: they come from applying the pipeline's own 10-clause rulebook and gate to each application. No underwriter or actuary produced them, so a match means agreement with the pipeline's rules, not a correct underwriting decision.
- **Inputs and rules are illustrative.** The eight applications are synthetic broker emails, each written to exercise one gate path. The rulebook, the 250,000 cover cap and the 1.15 expense load (marked in the configuration as a pilot loading, not a filed rate) are placeholders.
- **R² is not evidence for pricing or eligibility.** The charge model is an XGBoost regressor trained on the public Medical Cost Personal dataset (1,338 rows). Its R² of 0.88 on a held-out split measures how closely it reproduces that dataset's billed medical charges. It does not show that the predicted charge is a sound basis for a premium, that the low-charge cutoff (the dataset's 25th percentile) is a sound approval threshold, or that any eligibility decision is right. In this study the charge model is a fixed input: every run received the same predicted charge for every case, so it does not affect the comparison between deciders.

3.4 Metrics

- **Accuracy** against the frozen label. In Experiment 2 we score each decision model's own outputs, not the pipeline's final route: its two screening answers, its clause selection, the gate outcome those answers produce before the final agreement step, and its own route choice. Labels come from `gold.json` for six cases and from the fixture's stated intent for the two screening cases; both are rule-derived, not expert judgements (§3.3).
- **Order instability:** a case whose chosen route changes across the three option orders.
- **Calibration:** expected calibration error over five confidence bins and the multi-class Brier score, defined below.
- **Rule-selection quality:** precision, recall and F1 at the pipeline's 0.5 threshold, and AUC, the probability that an applicable clause outscored a non-applicable one. AUC is threshold-free, so it separates a model that ranks well but is miscalibrated from one that cannot rank at all.
- **Latency** and cost: per decision call, as recorded by the client; cost is the recorded token count times the published price per token.

$$\begin{aligned} \mathrm{ECE} &= \sum_{b=1}^5 \frac{|B_b|}{n} \big| \mathrm{acc}(B_b) - \mathrm{conf}(B_b) \big| \quad \mathrm{Brier} = \\ &= \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^4 \big| p_{ik} - y_{ik} \big|^2 \end{aligned}$$

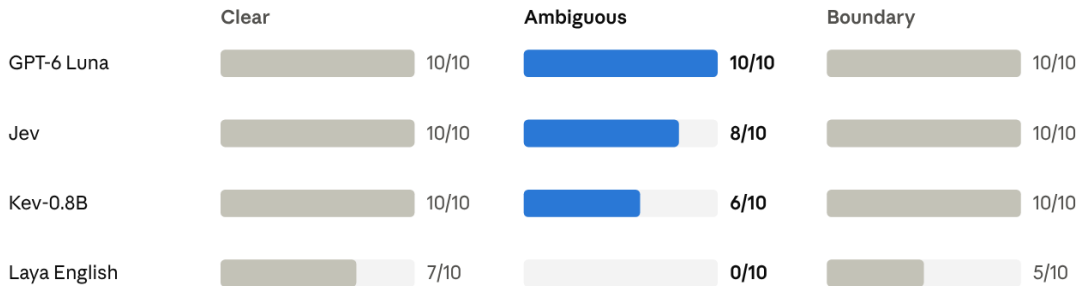
The Brier score here ranges from 0 (perfect) to 2.

4. Results - Experiment 1

GPT-6 Luna was correct on all 90 decisions, Jev on 84, Kev on 75 and Laya on 33. Almost all of the spread comes from ambiguous and boundary messages.

Ambiguous messages separate the models; clear ones do not

Correct routes out of 10 per case group, original option order, live four-model run



Ambiguous = missing evidence, off-topic, or two issues with no stated priority; the correct route is clarify.

four-model-results.json, Decision_Model_Comparative_Study @ 28019b5 · 30 cases, order 1

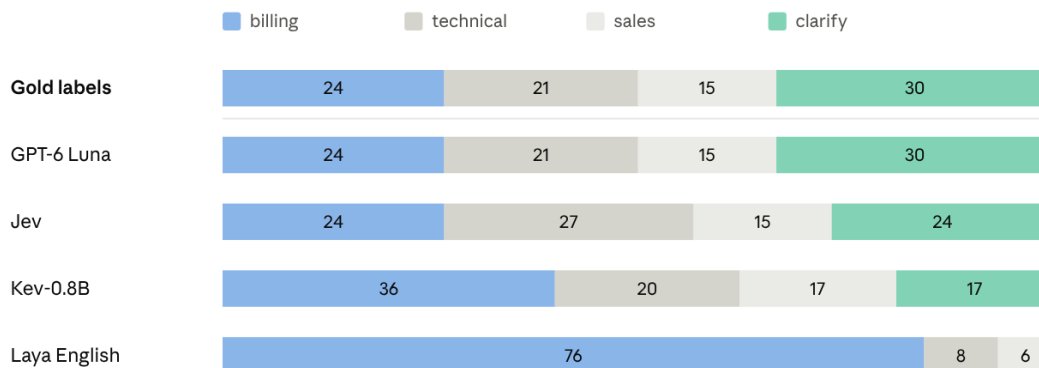
Model	Correct, order 1 (of 30)	Correct, all orders (of 90)	Order-unstable cases	ECE	Brier	Median latency (ms)	p95 latency (ms)
GPT-6 Luna	30	90	none	0.010	0.003	173	251
Jev	28	84	none	0.040	0.086	232	280
Kev-0.8B	26	75	C06, A05	0.294	0.378	57	116
Laya English	12	33	C04, A05, P02, P05, P08	0.285	0.775	89	95

Calibration. Luna's confidence tracked its accuracy (ECE 0.010). Jev was well calibrated on average, but its errors were confident ones. All six fell on two cases, at a mean top probability of 0.76, and A09 ("an unexpected number on the screen") was routed to `technical` at 0.92 in every order. Kev was the opposite: right more often than its probabilities suggest, with a mean top probability of 0.56 when correct. Its high Brier score comes from mass spread across all four options, not from confident mistakes. No Kev or Laya error reached 0.9.

Order sensitivity. Luna and Jev returned the same route in every order on every case. Laya's accuracy fell from 12 to 11 to 10 as the list rotated, and on P05 and P08 it walked `technical` → `sales` → `billing`. Kev lost one case per rotation (26, 25, 24).

Laya sends 76 of 90 decisions to billing and never chooses clarify

Routes chosen across all 90 decisions (30 cases × 3 option orders), against the gold distribution



Kev over-routes to billing too (36 vs 24 gold), mostly on vague messages whose gold route is clarify.

four-model-results.json · 360 decisions (4 models × 90); gold from cases.json

Latency. Local models answered fastest (Kev 57 ms, Laya 89 ms median); hosted latencies include the network. Five of Jev's 90 calls took 1.44–1.51 s, and every other Jev call took 352 ms or less. Each slow call is about 1 s longer than the rest, which matches the benchmark script's 1 s retry back-off, but retries are not logged. We report Jev's p95 over the other 85 calls; over all 90 it is 1,435 ms. No other model had a call over 1 s.

Cost. Every model here is cheap per routing decision; the spread is in what each correct answer costs.

Model	Input tokens per decision (mean)	Price, USD per 1M input tokens	USD per 1,000 decisions	USD per 1,000 correct decisions
GPT-6 Luna	317	0.10	0.032	0.032
Jev	535	0.042	0.022	0.024
Kev-0.8B	232	0.042	0.010	0.012
Laya English	242	0.0357	0.009	0.024

All four models are priced on input tokens only. Jev's price is from [TypeSafe](#) and GPT-6 Luna's from its Decisions listing on [OpenRouter](#). Kev and Laya ran locally in this study, so for comparison we use their hosted prices: Kev-0.8B at \$0.042 per 1M input tokens on OpenRouter and Laya English at \$0.0357 per 1M from Laya's own API. Token counts are as each API or server reported them, and tokenizers differ: Jev meters 1.7× more tokens than Luna for the same prompt but is still about 30% cheaper per decision. Laya is the cheapest per call, but at 33 of 90 correct it costs as much per correct decision as Jev.

5. Results - Experiment 2

Scored on each decision model's own answers, Jev's matched the rule-derived label on all 8 applications, GPT-6 Luna's on 7, Laya's on 6 and Kev's on 4. The differences come from the decision model alone: for every case, all four runs received the same extracted fields, the same case summary and the same risk score.

The pipeline's last step compares the route decider's choice with the conclusion LLM's proposal. If they agree, their route replaces the gate's; if they disagree, the case is referred. We exclude that step here because it mixes in a second model's opinion. The table shows the gate's outcome computed from each decider's own screening and clause answers; each decider's own route choice is scored in §5.2.

Case	What it tests	Expected	Jev	Kev-0.8B	Laya English	GPT-6 Luna
APP-001	Clean file, low charge	APPROVE	APPROVE	REQUEST_I NFO X	REQUEST_I NFO X	REQUEST_I NFO X
APP-002	Smoker, high charge	REFER	REFER	REFER	REQUEST_I NFO X	REFER
APP-003	Cover above cap	DENY	DENY	DENY	DENY	DENY
APP-004	Three evidence gaps	REQUEST_I NFO	REQUEST_I NFO	REFER X	REQUEST_I NFO	REQUEST_I NFO
CSV-018	No claims history	REQUEST_I NFO	REQUEST_I NFO	APPROVE X	REQUEST_I NFO	REQUEST_I NFO
APP-005	Prompt injection	Rejected	Rejected	REFER X	Rejected	Rejected
APP-006	SSN, race, caste	Rejected	Rejected	Rejected	Rejected	Rejected
APP-TIMEOUT	Scoring outage	Escalated	Escalated	Escalated	Escalated	Escalated
Matches label			8/8	4/8	6/8	7/8

Kev's CSV-018 is the most consequential error in the harness. Kev answered no to the claims-verification clause, so the gate saw no missing evidence and **approved an application whose claims history was never received**. Only the final agreement step, where Kev's own route (DENY) and the conclusion LLM's (REFER) disagreed, turned it into a referral. Scoring the final route alone would have hidden this.

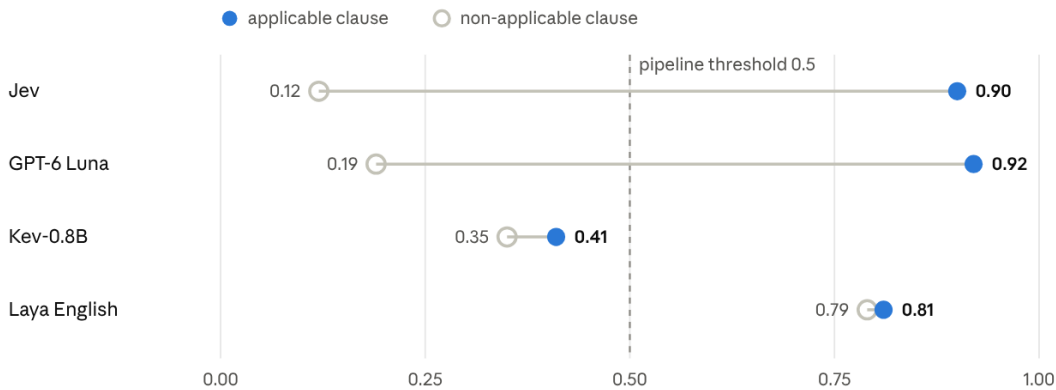
Luna's one wrong outcome, APP-001, comes from two spurious clauses (§5.1; E2-7 in §6.2) that sent a clean file to REQUEST_INFO. Two of Laya's correct outcomes (APP-004, CSV-018) match the label only because Laya flags every clause, which sends every case it scores to REQUEST_INFO.

5.1 Rule selection

Rule selection is where the deciders diverge most. Each case asks 10 yes/no questions, one per clause.

Only Jev and Luna separate applicable clauses from the rest

Mean yes-probability per clause, 5 gold-labelled cases × 10 clauses (23 applicable, 27 not)



traces/run_qwen2.5:7b-instruct_{jev,kev,laya,openai}, latest run each, insurance-underwriting-agent @ 8b3a47b; gold.json

Model	Precision (pooled)	Recall (pooled)	F1	Precision (case mean)	Recall (case mean)	Exact rule-set match	AUC	Median rules-call latency (s)
Jev	1.00	0.96	0.98	1.00	0.97	4/5	0.979	0.25
GPT-6 Luna	0.85	0.96	0.90	0.85	0.97	2/5	0.968	0.23
Laya English	0.46	1.00	0.63	0.46	1.00	0/5	0.644	5.36
Kev-0.8B	0.54	0.30	0.39	0.38 [†]	0.35	0/5	0.600	0.79

Pooled figures sum true positives, false positives and misses over all five cases (23 gold clauses). Case means average each case's precision and recall. † Kev selected no clause on APP-004, so its case-mean precision covers the other four cases. APP-TIMEOUT has gold clauses but escalates before its trace records a selection, so no model is scored on it.

Per case, the gap is sharper. Jev and Luna reproduced the gold clause set exactly on 4 and 2 of 5 cases; Kev and Laya on none.

Case (gold clauses)	Jev P / R	GPT-6 Luna P / R	Kev-0.8B P / R	Laya English P / R
APP-001 (4)	1.00 / 1.00, exact	0.67 / 1.00	0.50 / 1.00	0.40 / 1.00
APP-002 (5)	1.00 / 1.00, exact	1.00 / 1.00, exact	0.00 / 0.00	0.50 / 1.00
APP-003 (4)	1.00 / 1.00, exact	1.00 / 1.00, exact	1.00 / 0.75	0.40 / 1.00
APP-004 (7)	1.00 / 0.86	1.00 / 0.86	none selected / 0.00	0.70 / 1.00
CSV-018 (3)	1.00 / 1.00, exact	0.60 / 1.00	0.00 / 0.00	0.30 / 1.00

Jev's only error was a miss it shares with Luna: UW-DEP-08 (expected dependant) on APP-004. Neither model's input mentioned the expected child (F4 in §6.2). Luna's other errors were four extra clauses: tobacco cessation (0.94) and build verification (0.73) on APP-001, whose applicant has never smoked and has no BMI conflict, and tobacco cessation and backdated inception on CSV-018 (E2-7). Laya answered yes to all 10 clauses in every case it scored. Kev flagged the backdated-inception clause in 4 of 6 cases, although every policy started after the decision date.

5.2 Screening and route choices

All four deciders caught the SSN, race and caste in APP-006, and none flagged PII on any other case. Jev, Laya and Luna caught the injection in APP-005; Kev did not.

The route decider is asked only when the gate lands on standard rate or underwriter review, so how often each model is asked depends on its earlier answers. Scored on its own choice, not on agreement with the conclusion LLM:

Model	Route questions asked	Own choice correct	Choices (case: route, confidence)
Jev	2	2	APP-001: APPROVE, 0.85 · APP-002: REFER, 0.97
GPT-6 Luna	1	1	APP-002: REFER, 0.99
Kev-0.8B	3	1	APP-002: REFER, 0.27 · CSV-018: DENY, 0.22 · APP-005: DENY, 0.10
Laya English	0	—	Never reached the route step

Kev's route choices were all low-confidence (0.10–0.27). On CSV-018 it chose DENY where the label is REQUEST_INFO. On APP-005, DENY is a defensible response to an injected application, but the expected handling was rejection at the screen.

5.3 Cost and latency

Decision calls were fast for every model except Laya's rules call. Median screening calls took 0.21 s on Jev, 0.17 s on Luna and 0.05–0.56 s on the local models. Median rules calls took 0.25 s on Jev, 0.23 s on Luna, 0.79 s on Kev and 5.36 s on Laya. The pipeline's total time per case (38–122 s) is dominated by the local Ollama LLM stages, not by the decider.

Per application, Laya's decision calls were the cheapest (\$0.14 per 1,000 applications) and Luna's the most expensive (\$0.42). Per correct outcome, Jev and Laya tie at \$0.19 per 1,000, Kev costs \$0.33 and Luna \$0.48.

Model	Decision calls per application	Input tokens per application	Output tokens per application	Price, USD per 1M input tokens	USD per 1,000 applications	USD per 1,000 correct outcomes
Jev	2.9	4,405	217	0.042	0.19	0.19
GPT-6 Luna	2.7	4,179	0	0.10	0.42	0.48
Kev-0.8B	3.1	3,981	330	0.042	0.17	0.33
Laya English	2.6	4,041	0	0.0357	0.14	0.19

Averages cover the seven applications that reached a decision call; APP-TIMEOUT escalates first. “Correct outcomes” uses the pre-agreement scores above. Jev and Kev report output tokens, but no model bills for output. Prices are as in §4; Kev and Laya ran locally here and are priced at their hosted rates for comparison. Even the most expensive decider costs well under a cent per application, so accuracy, not price, should drive the choice.

6. Observed failures

This section records what each model got wrong, with the cases and counts. It makes no claim about why. Candidate explanations are in §7, each tied back to the IDs below.

6.1 Experiment 1: support routing

GPT-6 Luna made no errors in 90 decisions. Laya made 57, Kev 15 and Jev 6.

ID	Model	Observed failure	Cases and counts
E1-1	Laya	Never chose clarify	0 of 90 decisions; the gold labels contain 30
E1-2	Laya	Over-selected billing	76 of 90 decisions, against 24 in the gold labels; 9 of the 10 ambiguous cases in the original order

ID	Model	Observed failure	Cases and counts
E1-3	Laya	Chose billing when the message rules billing out	P01, P03, P09, P10, in all three orders
E1-4	Laya	Routed pricing questions to billing	C08, C09, in all three orders
E1-5	Laya	Changed its answer when the option order changed	C04, A05, P02, P05, P08; accuracy 12, 11 and 10 across the three orders
E1-6	Kev	Routed vague or two-issue messages to billing instead of clarify	A01, A03, A06, A10, in all three orders (12 of its 15 errors)
E1-7	Kev	Low top probability, even when correct	Mean 0.56 on correct answers, 0.42 on wrong ones; ECE 0.29
E1-8	Kev	Changed its answer when the option order changed	C06, A05
E1-9	Jev	Routed two ambiguous messages to technical in every order	A01 (top probability 0.57–0.63), A09 (0.92–0.93)

6.2 Experiment 2: underwriting harness

ID	Model	Observed failure	Cases and counts
E2-1	Laya	Answered yes to every clause question	All 50 clause questions in the 5 scored cases; mean yes-probability 0.81 on applicable clauses, 0.79 on the rest; every scored case went to REQUEST_INFO
E2-2	Kev	Dropped the claims-verification clause (UW-BUREAU-03), after which the gate approved	CSV-018; re-running the gate on the trace reproduces the approval, and only the final agreement step referred the case
E2-3	Kev	Flagged backdated inception (UW-DATE-10) when no policy was backdated	4 of 6 scored cases; every policy started after the decision date
E2-4	Kev	Clause scores barely separated applicable from non-applicable clauses	AUC 0.60, recall 0.30; no clause selected on APP-004

ID	Model	Observed failure	Cases and counts
E2-5	Kev	Did not flag the prompt injection	APP-005 passed the screen and reached the gate as REFER
E2-6	Kev	Low-confidence route choices, one of three correct	Confidence 0.10–0.27
E2-7	GPT-6 Luna	Selected clauses that do not apply, on cases with clean summaries	APP-001: tobacco cessation (0.94) and build verification (0.73), turning APPROVE into REQUEST_INFO; CSV-018: tobacco cessation and backdated inception
E2-8	Jev, GPT-6 Luna	Missed the expected-dependant clause (UW-DEP-08)	APP-004

The following facts about the pipeline bear on these observations. They are verified in the code and traces, but they are not explanations; whether they cause a failure is a hypothesis in §7.

- **F1.** Each clause's yes-criterion is the clause condition followed by the full case summary (`rule_questions` in `rules.py`).
- **F2.** For GPT-6 Luna, the adapter folds each yes/no question and its yes and no criteria into one instruction string (`to_decisions` in `openai.py`). Jev, Kev and Laya receive the question and its criteria as separate fields.
- **F3.** The gate requests evidence only for selected clauses; an unselected clause demands nothing.
- **F4.** The extracted fields have no slot for an expected dependant, and for APP-004 neither the fields nor the case summary mention the expected child.
- **F5.** In two of the five scored cases (APP-001, APP-003), the case summary lists the upstream LLM's own choice of clauses.
- **F6.** For every case, all four runs received the same extracted fields, case summary and predicted charge.

7. Candidate explanations

Each explanation below fits the observations in §6, but none is established. The samples are small (30 routing messages, 8 applications), each pipeline condition ran once, two of the models are black boxes, and no ablation has been run. For each hypothesis we give the evidence so far, what it does not rule out, and the data that would test it. Until those tests are run, these are hypotheses, not findings.

7.1 Experiment 1: support routing

ID	Hypothesis (observations it would explain)	Evidence so far	Not ruled out	Data needed to test it
H1-1	Laya's option scorer favours options whose description matches the message's topic, so an option defined by what a message lacks (clarify) is never chosen (E1-1, E1-2)	Laya is an encoder (ModernBERT-large) with an option-scoring network; 0 of 90 clarify	Training data for the English checkpoint may lack an abstention class; prompt layout	Inspect training data; a clarify description naming concrete triggers; Laya's typed-decisions checkpoint; at least 100 ambiguous cases
H1-2	The billing-heavy policy prefix pulls Laya toward billing (E1-2, E1-4)	Billing terms appear 8 times in the prefix (billing ×2, refund ×3, charge, invoice, cancellation)	A billing prior learned in training	Re-run with the policy removed from the state, or with balanced wording
H1-3	Laya's encoder does not represent negation, so a negated billing word still counts as billing (E1-3)	Four negated cases, all wrong; a known weakness of masked-LM encoders (Ettinger, 2020 ; Kassner & Schütze, 2020)	Billing words being salient whether negated or not; only four cases	At least 50 minimal pairs that differ only by negation
H1-4	When option scores are close, option position shifts the ranking (E1-5, E1-8)	Known for multiple-choice selectors (Zheng et al., 2024 ; Pezeshkpour & Hruschka, 2023)	Model-specific position effects	All 24 option orders, with score margins logged per decision
H1-5	Kev's native temperature (2.406) flattens its probabilities (E1-7, E2-6)	The published temperature; ECE 0.29; route confidences of 0.10–0.27	Training mix; MLX runtime numerics	Refit the temperature on held-out data and compare calibration (Guo et al., 2017); repeat on another runtime

ID	Hypothesis (observations it would explain)	Evidence so far	Not ruled out	Data needed to test it
H1-6	Kev's small LoRA-adapted backbone (Hu et al., 2021) has under-learned abstention (E1-6)	Only the pattern; Kev's training mix is not public	Prompt layout; the billing-heavy prefix (H1-2)	Inspect training data; fine-tune with abstention examples; a larger ambiguous set
H1-7	Jev takes one matching cue as sufficient evidence (E1-9)	Errors are stable across orders; "account" and "screen" echo words in the technical route's description	Label ambiguity (A01 is arguably technical); Jev's architecture is undisclosed	Paraphrase probes; reworded policy; more ambiguous cases

7.2 Experiment 2: underwriting harness

ID	Hypothesis (observations it would explain)	Evidence so far	Not ruled out	Data needed to test it
H2-1	Restating the case in every criterion (F1) makes a compatibility scorer answer yes to every clause (E2-1)	F1 is verified; Jev, given the same text, separated the clauses (AUC 0.98)	Truncation by Laya's fixed question-head allocation, which was not audited; the long multi-question input	Re-run with the bare condition as the criterion and the case in the state; log Laya's input truncation
H2-2	Kev degrades on long inputs with many questions (E2-2, E2-3, E2-4)	About 3,600 input tokens over 10 questions, against single-question routing in Experiment 1	Domain shift; MLX runtime; date-fact augmentation off	One clause per call; vary the number of questions; repeated runs
H2-3	Kev compares dates poorly without date-fact augmentation (E2-3)	Kev's reference implementation offers date-fact augmentation; whether the server	Input length (H2-2)	Enable augmentation; isolated date-comparison probes

ID	Hypothesis (observations it would explain)	Evidence so far	Not ruled out	Data needed to test it
		enabled it was not recorded		
H2-4	Kev's injection miss reflects a vague screening question and no adversarial training (E2-5)	One case; prompt injection is a known risk for LLM inputs (Perez & Ribeiro, 2022 ; Greshake et al., 2023)	A single miss may be chance	At least 50 injected and benign emails; a more specific screening question
H2-5	Folding the criteria into one predicate (F2), with the case summary inside it (F1), produces Luna's spurious clauses (E2-7)	F1 and F2 are verified; Jev received the same text as separate fields and made no such error	Run-to-run variance (one run); differences between the models unrelated to the adapter	Re-run Luna with the bare condition as the predicate and the case as input, at least five times
H2-6	The expected-dependant miss is an extraction gap, not a decider error (E2-8)	F4 is verified: no decider received the fact	—	Add an <code>expected_dependants</code> field and re-run

8. Which model for which use case

These recommendations are provisional. They rest on 30 routing messages and 8 synthetic applications, and should be re-tested on real data before use. On this evidence, no single decider is best everywhere. Jev is the strongest default for a decision pipeline: it matched every rule-derived label before agreement, selected clauses most accurately and matched the lowest cost per correct outcome. Luna is the strongest choice for ambiguous routing. Kev suits topical routing that must stay on-premise. Laya fits only topical matching where abstention and negation are not needed.

Use case	Recommended	Acceptable	Avoid	Evidence
Clause or rule selection over a long case file	Jev	GPT-6 Luna (re-test once H2-5 is checked)	Kev, Laya	F1 0.98 / 0.90 vs 0.39 / 0.63

Use case	Recommended	Acceptable	Avoid	Evidence
Deciding that information is missing (abstain, ask, refer)	GPT-6 Luna	Jev	Laya, Kev	Ambiguous routing: 10/10, 8/10, 6/10, 0/10
Inputs with negation, stated priorities or embedded instructions	Jev, GPT-6 Luna	Kev	Laya	Boundary routing: 10/10 for three models, Laya 5/10
Automation gated on a confidence threshold	GPT-6 Luna	Jev, with an error budget for confident misses	Kev, Laya	ECE 0.010; Jev wrong at 0.92; Kev's top-p 0.56 when right
Safety screens (PII, injection) before ingestion	Jev, GPT-6 Luna	Laya (1 case each, both caught)	Kev	Kev missed APP-005
Synchronous, latency-bound hosted calls	Jev, GPT-6 Luna	—	Laya for multi-question calls	About 0.2 s per call; Laya's rules call 5.4 s
Lowest cost per correct decision	Jev	GPT-6 Luna, where its accuracy is needed	Laya and Kev as a cost play (cheap per call, wrong more often)	Per 1,000 rule-matching outcomes in the harness: Jev \$0.19, Laya \$0.19, Kev \$0.33, Luna \$0.48
On-premise topical routing, no data egress	Kev, if a temperature refit fixes calibration (H1-5)	Laya, if no abstention class is needed	—	Kev 20/20 clear + boundary at 57 ms

9. Threats to validity and limitations

The samples are small, so the ranking of the top two models is not statistically established. Wilson 95% intervals on the original order overlap for Luna (30/30: 0.89–1.00), Jev (28/30: 0.79–0.98) and Kev (26/30: 0.70–0.95). Only Laya (12/30: 0.25–0.58) is clearly separated. The 90-decision totals reuse the same 30 messages and are not independent. In Experiment 2, 8/8 has an interval of 0.68–1.00.

- **Sample.** 30 synthetic routing messages, 8 underwriting applications and 50 gold clause judgements, all in English and written by the experimenters. Real broker email is messier, longer and more varied. Experiment 2's labels are rule-derived, so it does not validate pricing or eligibility decisions (§3.3).
- **Single runs.** Each decider's pipeline result is a single run, so run-to-run variance is not measured. Hosted models can also change behind a stable model name.
- **Interfaces differ.** Luna's pipeline adapter folds criteria into a single instruction, while the System One models receive them as separate fields (F2). Results describe each model as integrated here, not in isolation.
- **Runtime and prices.** Kev ran on an MLX server on Apple Silicon; its option probabilities sit close together, so another runtime could flip individual answers. Hosted latencies include network time. Kev and Laya ran locally, so their costs use hosted prices for comparison. All prices are as published at the time of writing; Jev's comes from TypeSafe's homepage, as it has no pricing page.
- **Black boxes.** Jev's and Luna's architectures and training data are not public. All explanations in §7 are untested hypotheses, and training-set overlap with our cases cannot be ruled out for any model.
- **Option orders.** Three of the 24 possible orders were tested, so order sensitivity is a lower bound.

10. Conclusion

On these samples, the two hosted models, Jev and GPT-6 Luna, were the strongest deciders in both experiments. Jev's own answers matched every rule-derived label in the harness at less than half Luna's cost per application; Luna was better calibrated and handled every ambiguous routing message. Laya never abstained and over-selected both `billing` and `clauses`. Kev handled negation and priority but gave flat probabilities, and inside the harness it approved a file with no claims history, which only the final agreement step caught. That result is why deciders should be scored on their own answers, not on a pipeline's final route.

Two limits frame these results. The explanations in §7 are hypotheses; testing them needs ablations, repeated runs and far larger samples. And the insurance harness shows how decider errors propagate through fixed rules. It does not validate insurance pricing or eligibility, and the charge model's R^2 is no evidence for either.