Are open-source decision models useful for enterprises?

A technical evaluation of Clef against hosted decision models and reasoning models

Mark J Sykes, AI One Labs | 2 November 2026
ContextOne sample classification task | 1,974 examples | 50 categories | Eight configurations

Executive summary

Yes. Stock Clef can classify enterprise requests locally with low latency. It trailed the best hosted configurations on accuracy in this test; its economic case depends on how much the hardware is utilized, and what value is placed on context sovereignty.

Clef Flash Q8_0 correctly classified **1,830 of 1,974 requests (92.71%)**, returned a valid result for every request, and delivered a median response in **221 ms**. It was close to Jev 1.13.0 (93.31%) and GPT-6 Luna Responses with low reasoning (93.21%). The test does not establish a reliable accuracy difference between these three configurations.

GPT-6 Luna Responses with high reasoning achieved the strongest observed top-1 result, **96.10%**, at a median **2.912 seconds**. GPT-6 Luna Decisions with descriptive labels and frozen shuffling reached **95.39%** at **286 ms**. Both Clef and native Luna Decisions were much faster than Responses with high reasoning.

For a stable test machine, assuming a $15,000 purchase price and a three-year write-down, allocated serving cost is about **$0.694 per 1,000 decisions at 10% utilisation** and **$0.144 at 50%**. The improved Luna Decisions run cost **$0.170 per 1,000** in reported usage. Data locality, availability requirements and the cost of a wrong route can outweigh these small API charges.

Query format changed the result

Changing choice values from numeric IDs to descriptive labels, together with a frozen shuffle, increased native Luna accuracy from 76.70% to 95.39%. Category descriptions were unchanged. The API accepted both formats, but their classification results differed sharply.

This paper reports measured classification results and explicit cost scenarios. It assesses workflow selection, not the safety or correctness of executing the selected workflow.

What an enterprise should take away

We would consider Clef for local workflow routing where the application can detect or tolerate occasional classification errors. The choice depends on the cost of a wrong route and whether uncertain requests can be reviewed.

Enterprise priority	What the evidence supports
Local control and fast routing	Clef returned 100% valid outputs, 92.71% top-1, 97.42% top-three and 306 ms p95 in the tested local configuration.
Highest observed top-1 quality	GPT-6 Luna | Responses API | High reasoning: 96.10%. Its advantage over Clef is 67 additional correct outputs, or 3.39 percentage points.
Hosted decisions at low latency	GPT-6 Luna | Decisions API | Descriptive labels | Shuffled: 95.39%, 541 ms p95, complete probabilities for valid answers.
Selective automation	Clef at top-label probability at least 0.80 accepts 77.61% of requests with 98.37% observed accuracy. The remaining requests require review or another classification method.
Cost discipline	Include infrastructure and staff costs, divided by the volume actually served. Low API prices make a dedicated rack difficult to justify for a small classification workload.

Open weights let an enterprise host the model within its own infrastructure, pin a version and inspect the serving software. The application still needs access controls, output validation and monitoring, with business rules governing any resulting action.

Clef is an Apache-2.0 open-weight model. The title uses the broader open-source question; throughout the measurements, “Clef” refers to the specific stock Q8_0 artifact and serving configuration tested. Availability of weights and serving code does not establish that all pretraining data or training processes are public. [P1]

Reading guide

Pages	Content
3-6	Test-set contents, task definition, APIs and statistical methods
7-15	Accuracy, paired comparisons, hard cases, calibration and latency
16-19	Hosted cost, workstation ownership and enterprise H200 rack economics
20	Clef serving requirements and batch-size configuration
21-24	Best Practice with OpenAI: evidence, hypotheses and query design
25	Enterprise implications and further research
26-29	Full taxonomy and per-category statistical tables
30	Category recall heatmap
31-32	Evidence register, reproducibility and external references

The 50-category classification task

The task is to identify the **single primary workflow** requested by a user, using a fixed ContextOne sample taxonomy of 50 enterprise workflows. Each example supplies a current prompt and a separate context field. A correct answer identifies the requested work, rather than simply recognising the business topic.

Property	Observed composition
Examples and classes	1,974 examples; all 50 classes represented; 34-40 examples per class
Baseline scenarios	1,488 examples: 496 scenario families, each represented at three intended difficulty levels
Boundary scenarios	486 distinct scenarios targeting close category boundaries
Resampling units	982 recorded scenario components: 496 groups of three and 486 singletons
Context availability	1,895 examples (96.00%) have nonempty context; 79 do not
Input length	Prompt plus context: median 75 whitespace-delimited words; p95 138; maximum 204
Class balance	45 categories have 40 examples; two have 35; two have 34; one has 36

The test set is a synthetic, scenario-based evaluation assembled to cover common enterprise activities and difficult boundaries between neighbouring workflows. The construction process is outside this paper. The contents, scoring policy and retained scenario relationships define the comparison reported here.

What the taxonomy covers

The taxonomy spans shared enterprise operations, document work, records and controls; customer service and account operations; sales and marketing; and software delivery and production operations. Categories describe actions such as approval routing, reconciliation, refund handling, account research, code review and runbook execution. The complete category names and definitions appear on pages 26-27.

The category mix is broad and nearly balanced, rather than weighted by any client's request frequencies. Accuracy and escalation rates will therefore differ in deployments dominated by straightforward support tickets or by ambiguous follow-ups.

Source: frozen ContextOne cohort and taxonomy, evidence E1. Counts and word lengths independently recomputed for this paper. Word counts describe public input only; model requests also contain policy, taxonomy and API structure.

Why short prompts can be difficult

This is primarily a test of semantic discrimination and context use, rather than a long-context stress test. A short follow-up can require more careful interpretation than a longer standalone request: the current action may be implicit, earlier work may already be complete, and a familiar keyword can point to the wrong workflow.

Boundary mechanism	n	Decision the model must make
Context follow-up	98	Recover the current requested action from prior context without classifying the earlier task.
Lexical decoy	97	Resist salient words that suggest a neighbouring class but do not determine the deliverable.
Deliverable versus topic	97	Distinguish the work requested from the subject matter being discussed.
Workflow stage	97	Identify the present stage: investigation, approval, execution, record update or recovery.
Scope change	97	Recognise that the user has narrowed, expanded or replaced the previous request.

Representative boundary cases

Visible request and context, abridged	Required distinction
“Who must approve before the account is created?” Context: the supplier data checks and required documents are complete.	Approval routing (03), rather than onboarding (26). Clef chose onboarding.
“And where does the EU 14-day rule override it?” Context: the preceding exchange concerned returns-policy wording.	Knowledge Q&A (05), rather than processing a refund (24). Clef chose refund handling.
“COI cleared. Take the next step.” Context: a procedure identifies the next step as creating an ERP vendor record.	The reference favours SOP execution (04); both Clef and GPT-6 Luna Responses high chose system-of-record update (13).

The reference labels follow the policy for choosing a primary workflow. In the final example, both the procedure being followed and the record update it requires are plausible categories. The policy must say which takes precedence.

The three baseline difficulty levels are intended construction strata, not a guaranteed ordering of measured difficulty. Clef scores 97.78%, 90.32% and 94.15% at levels 1, 2 and 3 respectively. We therefore report each stratum rather than assuming that the label “hard” defines an objective model-independent scale.

Eight configurations, three interfaces

We tested eight combinations of model, API, output requirements, reasoning setting and serving configuration. An identical model alias across two APIs does not establish that their inference paths are identical.

Condition	Interface and returned evidence	Control / run
Clef Flash Q8_0 SystemOne Decisions API	SystemOne choice; full 50-class probabilities. Stock Q8_0 on RTX 4090.	No exposed reasoning control. 1 Nov; serial.
Jev 1.13.0 SystemOne Decisions API	SystemOne choice; full 50-class probabilities. Hosted Jev 1.13.0.	No reasoning qualifier. Historical; concurrency 2.
GPT-5.6 Luna Responses API | Low reasoning	Responses; ranked top-three category object and self-reported probabilities.	Low reasoning. Historical; concurrency 2.
GPT-5.6 Luna Responses API | High reasoning	Responses; ranked top-three category object and self-reported probabilities.	High reasoning. Historical; concurrency 2.
GPT-6 Luna Responses API | Low reasoning	Responses; ranked top-three category object and self-reported probabilities.	Low reasoning. Historical; concurrency 2.
GPT-6 Luna Responses API | High reasoning	Responses; ranked top-three category object and self-reported probabilities.	High reasoning. Historical; concurrency 2.
GPT-6 Luna Decisions API | Numeric labels	Native Decisions; numeric choice values in canonical order; full probabilities.	No reasoning control. 1 Nov; serial.
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	Native Decisions; descriptive values and frozen per-example order; full probabilities.	No reasoning control. 2 Nov; pooled, concurrency 4.

OpenAI Responses generates a structured answer under the task prompt. Native Decisions separates input evidence from typed decision questions and returns a selected allowed value. Current documentation supports GPT-6 Luna and 2-255 unique string or boolean choice values; the request schema exposes no reasoning-effort setting. Jev and Clef use the SystemOne choice interface. [P2-P5]

The historical cloud measurements are from September 2026; fresh Clef and native Decisions measurements are from November. All conditions use the same public examples and category definitions. Request structures and output requirements differ by API. Reasoning-level comparisons are explicit within Responses; native Decisions must not be labelled “low”, “high” or “zero” reasoning.

Scoring and statistical inference

The primary denominator is all **1,974 attempted examples**. Every condition completed the full cohort. Invalid answers and native refusals count as unsuccessful outputs. A success therefore requires a valid, correct classification. Valid-only accuracy measures classification quality among answers that passed output validation.

Metric	Definition
Top-1 success	Correct valid first choice / 1,974. Invalid and refused outputs contribute zero.
Top-three hit rate	Reference class present in the valid returned top three / 1,974. A shortlist is not a completed single-choice decision.
Macro-F1	Mean of per-class F1 over all 50 classes. Invalid outputs count as a false negative for the reference class.
Paired difference	Accuracy of condition A minus B on the same examples. Report A-only wins, B-only wins and uncertainty together.
Calibration	Top-label ECE uses 10 equal-width probability bins on valid responses. Brier is the sum of squared class errors; log loss uses natural logarithms.
Timing and cost	Client-observed request latency; valid, failed and all-terminal timing distinguished. Cost uses saved usage and rates, including unsuccessful outputs.

Confidence intervals account for related scenarios

The paper uses 5,000 percentile bootstrap resamples of the 982 recorded scenario components, drawing whole components with replacement. All sibling difficulty variants move together. Accuracy remains weighted by rows in each resample; paired comparisons use the same selected components for both configurations. The paper-wide calculation uses NumPy PCG64, seed 20261102. These are 95% component-bootstrap intervals, not independent Bernoulli intervals.

This estimates variability across the recorded scenario structure conditional on the fixed reference labels. Per-class support is only 34-40, so fine-grained results identify diagnostic patterns rather than precise class-level rankings. Intervals are not adjusted for multiple comparisons. We use them to assess the strength of differences, not to declare statistical equivalence when an interval crosses zero.

Comparison controls

Public input, category definitions and classification policy were frozen. References and author metadata were withheld. No malformed probabilities were repaired, missing classes filled with zeros, labels shifted, or wrong predictions retried. Raw answers, mappings, usage and terminal states remain auditable. The descriptive/shuffled format was selected after diagnosis on this development-used cohort; its observed improvement is not an untouched production-validation estimate.

Historical top-three outputs have incomplete probability mass and are not renormalised into full distributions. Native full-distribution metrics use valid answers only; log-loss probabilities below 10^{-12} are clipped with counts disclosed. Sources: E1-E4.

Clef accuracy in the full comparison

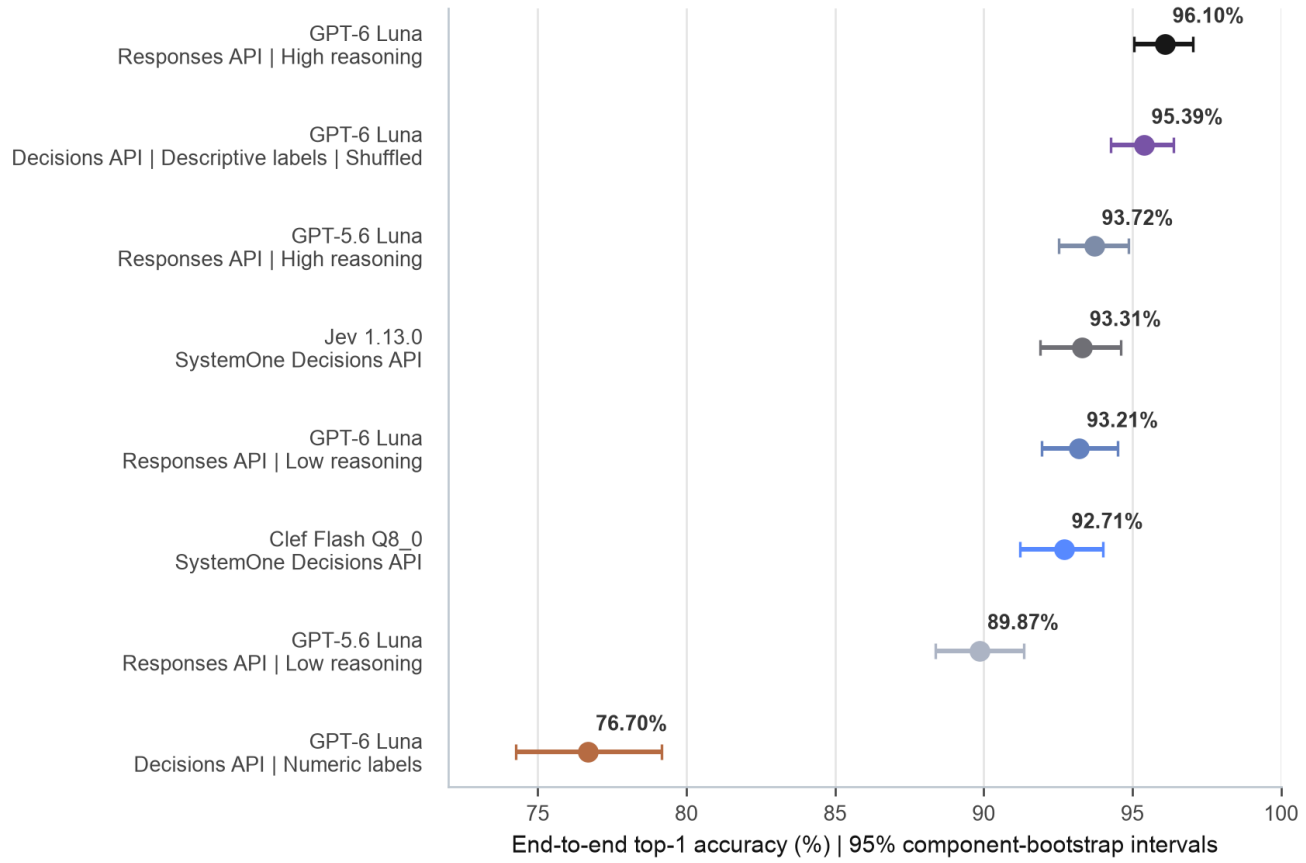


Figure 1. End-to-end top-1 accuracy across all 1,974 examples. Points show observed accuracy; bars show 95% component-bootstrap intervals. The horizontal scale is focused on 72-100% for comparison; these are point estimates, not magnitude bars. Source: E1 and independent analysis.

Clef achieves **92.71%** accuracy, with a 95% interval of **91.22-94.02%**. All 144 incorrect outputs passed validation: Clef selected the wrong category in each case. The best hosted configurations made fewer errors.

GPT-6 Luna Responses high returns 1,897 correct outputs, 67 more than Clef. The descriptive/shuffled Decisions condition returns 1,883, 53 more than Clef. By contrast, Jev and GPT-6 Luna Responses low are only 12 and 10 correct outputs ahead of Clef. The paired analysis on page 9 distinguishes these larger and smaller differences.

The original numeric-label Luna Decisions result remains visible at 76.70%. The API accepted this format, but its accuracy was substantially lower. The separate OpenAI section explains the diagnosis and subsequent full-cohort improvement.

Interpretation

Clef returned a valid answer to every request. Its 92.71% accuracy makes it a candidate where local deployment is required, provided the application can tolerate or catch the remaining classification errors.

Output validity changes the ranking

Configuration	Correct	Valid	Top-1 / all	Top-1 / valid	Top-3 / all	Macro-F1
GPT-6 Luna Responses API | High reasoning	1,897	1,953	96.10%	97.13%	97.57%	0.9653
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	1,883	1,971	95.39%	95.54%	98.63%	0.9544
GPT-5.6 Luna Responses API | High reasoning	1,850	1,938	93.72%	95.46%	96.66%	0.9450
Jev 1.13.0 SystemOne Decisions API	1,842	1,945	93.31%	94.70%	96.91%	0.9374
GPT-6 Luna Responses API | Low reasoning	1,840	1,914	93.21%	96.13%	95.49%	0.9447
Clef Flash Q8_0 SystemOne Decisions API	1,830	1,974	92.71%	92.71%	97.42%	0.9261
GPT-5.6 Luna Responses API | Low reasoning	1,774	1,871	89.87%	94.82%	92.91%	0.9207
GPT-6 Luna Decisions API | Numeric labels	1,514	1,970	76.70%	76.85%	96.56%	0.7527

Table 1. Each row has 1,974 attempts. “Valid” means the answer passed the required structure, allowed-label and probability checks for its configuration. Top-1 / valid is diagnostic and must not replace the all-attempt denominator. Top-three hit rate has the same all-attempt denominator.

Valid-only accuracy omits failed outputs

GPT-5.6 Luna Responses low is 94.82% accurate among its valid outputs but only 89.87% successful end to end, because 101 answers failed output validation and two were incomplete, leaving no usable classification for those requests. GPT-6 Luna Responses low moves from 96.13% valid-only accuracy to 93.21% end-to-end success. Clef has no such denominator gap in this run.

Configuration	Valid wrong	Invalid	Incomplete	Refused
Clef Flash Q8_0 SystemOne Decisions API	144	0	0	0
Jev 1.13.0 SystemOne Decisions API	103	29	0	0
GPT-5.6 Luna Responses API | Low reasoning	97	101	2	0
GPT-5.6 Luna Responses API | High reasoning	88	36	0	0
GPT-6 Luna Responses API | Low reasoning	74	60	0	0
GPT-6 Luna Responses API | High reasoning	56	21	0	0
GPT-6 Luna Decisions API | Numeric labels	456	0	0	4
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	88	0	0	3

An answer can name the correct category yet be unusable if its structure or probabilities fail validation. Historical mass tolerance (0.001) and native tolerance (0.000001) are retained without retrospective repair. Failed outputs are included in the all-attempt scores above.

Paired accuracy differences from Clef

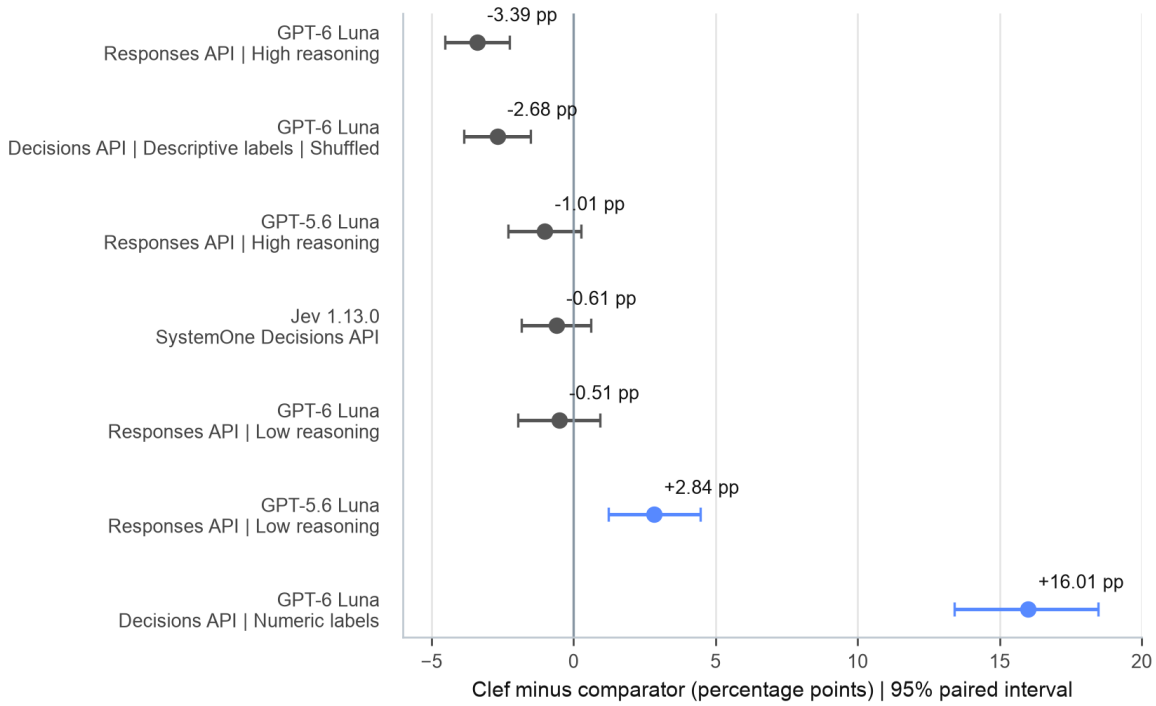


Figure 2. Clef minus each comparator in percentage points, using matched examples and 5,000 component resamples. Positive values favour Clef. All 1,974 examples are included in every pair. Source: E1.

Comparator	Clef-only correct	Other-only correct	Delta pp	95% interval pp
GPT-6 Luna Responses API | High reasoning	29	96	-3.39	-4.54 to -2.24
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	37	90	-2.68	-3.87 to -1.52
GPT-5.6 Luna Responses API | High reasoning	68	88	-1.01	-2.31 to +0.25
Jev 1.13.0 SystemOne Decisions API	54	66	-0.61	-1.84 to +0.61
GPT-6 Luna Responses API | Low reasoning	85	95	-0.51	-1.95 to +0.94
GPT-5.6 Luna Responses API | Low reasoning	142	86	+2.84	+1.23 to +4.45
GPT-6 Luna Decisions API | Numeric labels	381	65	+16.01	+13.40 to +18.47

Clef's differences from Jev, GPT-5.6 Luna Responses high and GPT-6 Luna Responses low have intervals spanning zero. The evidence does not resolve an advantage within those pairs. GPT-6 Luna Responses high and descriptive/shuffled Decisions show clear positive advantages over Clef on this cohort. Clef improves on GPT-5.6 Luna Responses low and the original numeric-label Decisions condition.

Clef gets 29 cases right that GPT-6 Luna Responses high misses. These disagreements could inform an ensemble, but selecting a model after seeing the reference answer would overstate the performance of any deployable system.

Accuracy on baseline and boundary cases

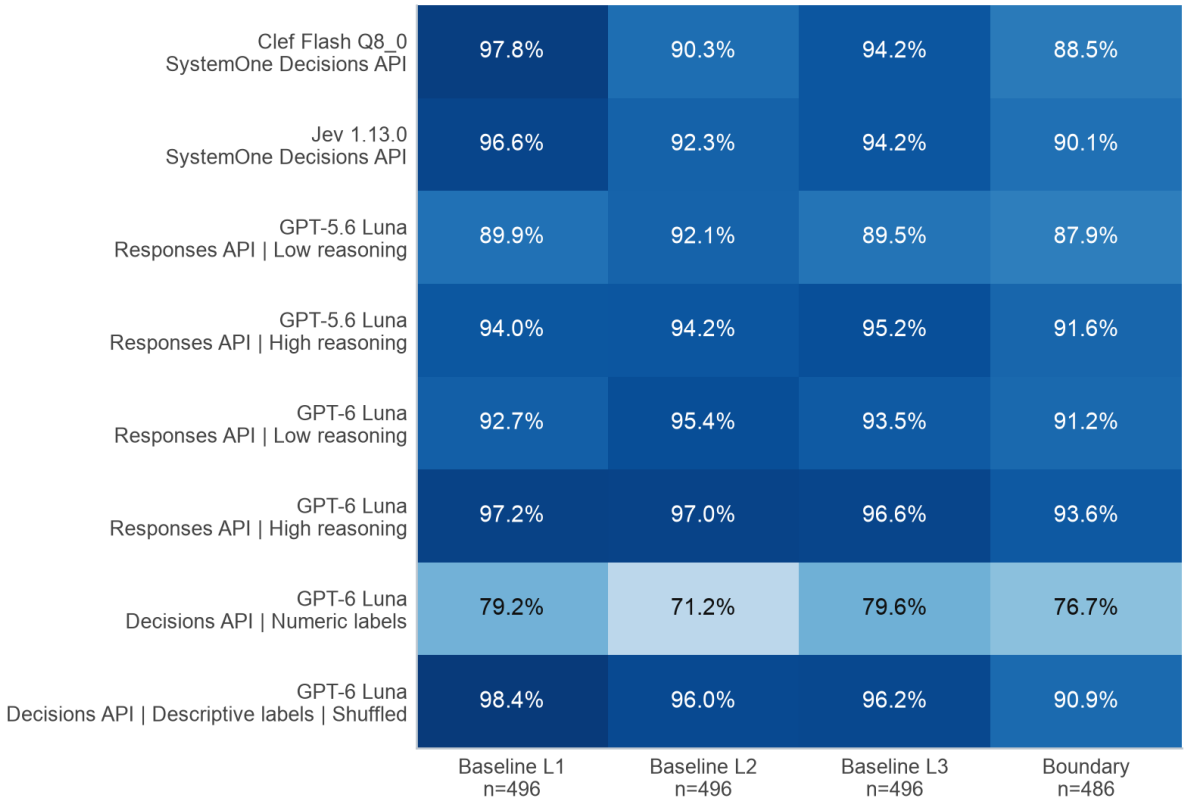


Figure 3. End-to-end accuracy by intended baseline difficulty and boundary cohort. All eight configurations use identical membership. The three baseline columns each contain 496 related-family variants; the boundary column contains 486 cases. Source: E1.

Configuration	Baseline correct	Baseline %	Boundary correct	Boundary %	Gap pp
Clef Flash Q8_0 SystemOne Decisions API	1400/1,488	94.09%	430/486	88.48%	5.61
Jev 1.13.0 SystemOne Decisions API	1404/1,488	94.35%	438/486	90.12%	4.23
GPT-5.6 Luna Responses API | Low reasoning	1347/1,488	90.52%	427/486	87.86%	2.66
GPT-5.6 Luna Responses API | High reasoning	1405/1,488	94.42%	445/486	91.56%	2.86
GPT-6 Luna Responses API | Low reasoning	1397/1,488	93.88%	443/486	91.15%	2.73
GPT-6 Luna Responses API | High reasoning	1442/1,488	96.91%	455/486	93.62%	3.29
GPT-6 Luna Decisions API | Numeric labels	1141/1,488	76.68%	373/486	76.75%	-0.07
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	1441/1,488	96.84%	442/486	90.95%	5.89

Clef falls from 94.09% on baseline examples to 88.48% on the boundary cohort, a 5.61-point gap. GPT-6 Luna Responses high reaches 93.62% on boundary cases; descriptive/shuffled Decisions reaches 90.95%. Applications handling terse follow-ups or requests that span several workflows should evaluate these cases separately.

Accuracy by boundary mechanism

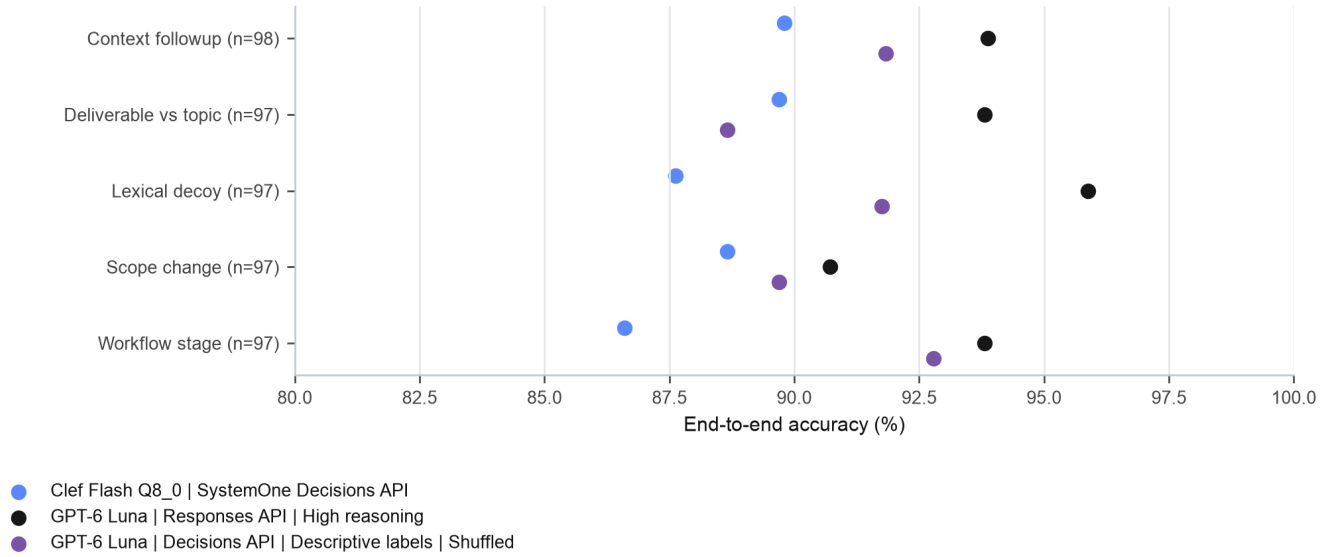


Figure 4. Boundary mechanisms for Clef and the two strongest aggregate configurations. Markers show all-attempt accuracy; these small strata contain 97-98 cases each. Source: E1.

Condition key: C = Clef Flash Q8_0, SystemOne Decisions API; J = Jev 1.13.0, SystemOne Decisions API; 5L/5H = GPT-5.6 Luna, Responses API, Low/High reasoning; 6L/6H = GPT-6 Luna, Responses API, Low/High reasoning; DN = GPT-6 Luna, Decisions API, Numeric labels; DS = GPT-6 Luna, Decisions API, Descriptive labels, Shuffled.

Mechanism	n	C	J	5L	5H	6L	6H	DN	DS
Context followup	98	88 (89.8%)	88 (89.8%)	89 (90.8%)	91 (92.9%)	92 (93.9%)	92 (93.9%)	78 (79.6%)	90 (91.8%)
Deliverable vs topic	97	87 (89.7%)	88 (90.7%)	81 (83.5%)	85 (87.6%)	84 (86.6%)	91 (93.8%)	73 (75.3%)	86 (88.7%)
Lexical decoy	97	85 (87.6%)	90 (92.8%)	82 (84.5%)	91 (93.8%)	87 (89.7%)	93 (95.9%)	74 (76.3%)	89 (91.8%)
Scope change	97	86 (88.7%)	85 (87.6%)	89 (91.8%)	88 (90.7%)	89 (91.8%)	88 (90.7%)	68 (70.1%)	87 (89.7%)
Workflow stage	97	84 (86.6%)	87 (89.7%)	86 (88.7%)	90 (92.8%)	91 (93.8%)	91 (93.8%)	80 (82.5%)	90 (92.8%)

Clef's lowest boundary-stratum score is workflow stage: 84/97 (86.60%). Its strongest is context follow-up: 88/98 (89.80%). The examples on page 4 illustrate how a step in a larger process can suggest two categories. These scores identify where Clef struggled; they do not reveal its internal reasoning.

GPT-6 Luna Responses with high reasoning correctly classified 93/97 lexical decoys (95.88%). The descriptive/shuffled native condition gains substantially over its numeric-order predecessor across every mechanism. The improvement appears across all five types of boundary case.

Client evaluations should include ambiguous transitions between workflows, especially where the selected category changes permissions or responsibility. An aggregate score can hide errors at these boundaries.

Where Clef makes mistakes

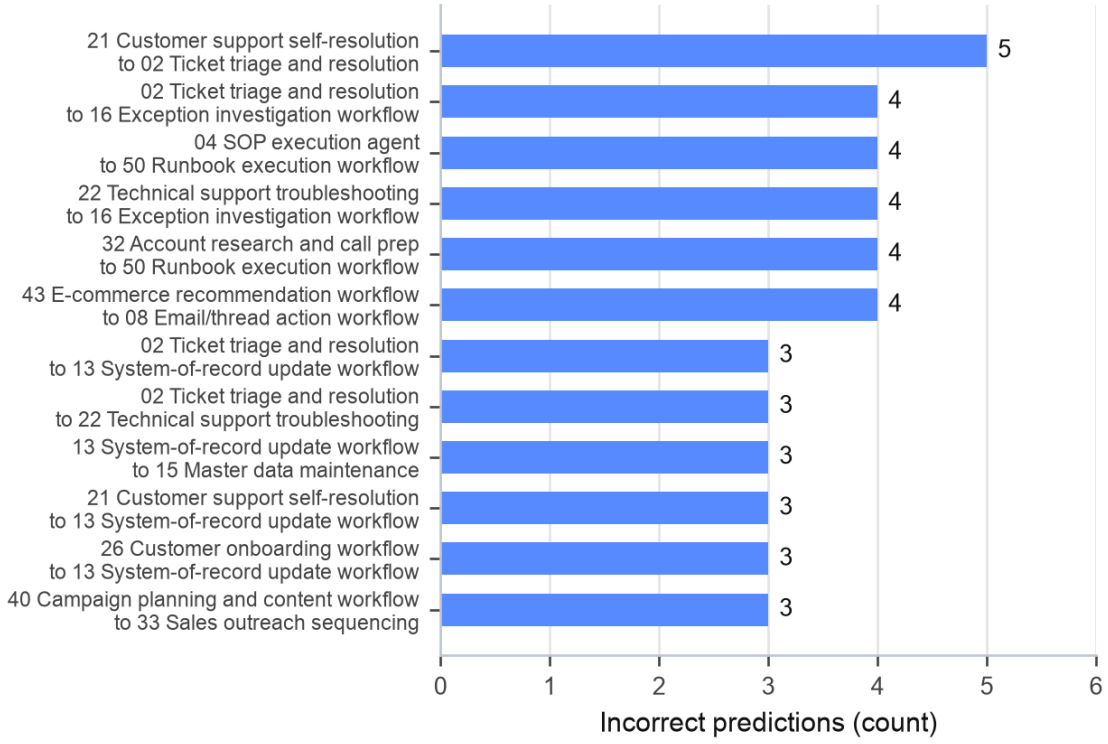


Figure 5. Twelve of the most frequent directed Clef confusions. Counts are actual examples, not percentages. Several additional pairs tie at three errors; the full confusion matrix is retained in the analysis. Source: E1.

Clef confused several pairs of neighbouring workflows; no single category absorbed most errors. Customer support self-resolution is confused with ticket triage five times; ticket triage, technical troubleshooting and several operational workflows were also mistaken for exception investigation or record updates.

Clef class / lowest recall	n	Correct	Precision	Recall	F1
02 Ticket triage and resolution	40	26	76.47%	65.00%	0.703
21 Customer support self-resolution	40	28	100.00%	70.00%	0.824
43 E-commerce recommendation workflow	40	30	100.00%	75.00%	0.857
26 Customer onboarding workflow	34	26	92.86%	76.47%	0.839
47 Test generation and maintenance	40	31	96.88%	77.50%	0.861
22 Technical support troubleshooting	40	32	86.49%	80.00%	0.831

Per-category precision and recall reveal which labels attract or lose assignments. System-of-record update has 92.50% recall but only 71.15% precision: it attracts incorrect assignments from other workflows. Ticket triage has 65.00% recall. These categories need different corrections: fewer false assignments to record updates, and fewer missed ticket-triage requests.

Single-category results have modest support. The appendix gives all 50 class counts and F1 values so readers can inspect the pattern without relying on a selectively chosen error list.

How well the probabilities predict accuracy

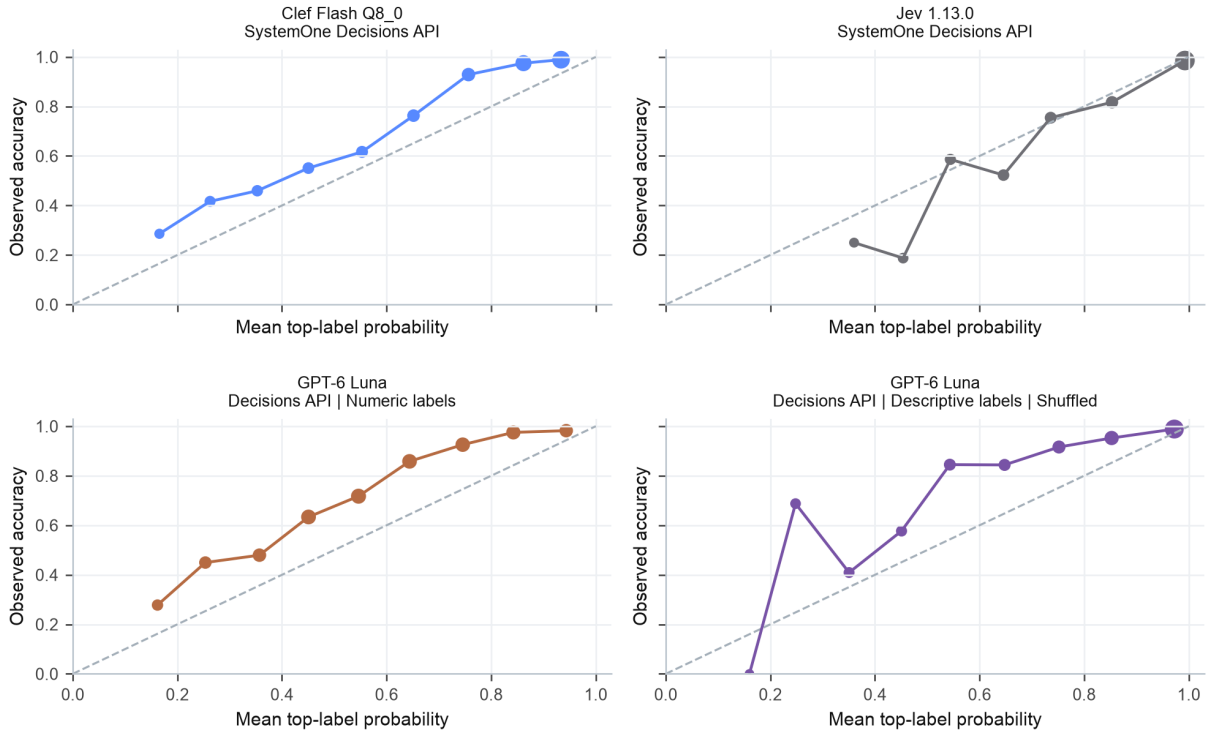


Figure 6. Reliability diagrams for the four native full-distribution configurations. Each point is a nonempty equal-width confidence bin; marker size increases with bin count. The diagonal is perfect calibration. The statistic is top-label probability, not a separate provider confidence field. Source: E1.

Configuration	Valid n	ECE	Brier	Log loss	Clipped
Clef Flash Q8_0 SystemOne Decisions API	1974	0.0880	0.1211	0.3703	0
Jev 1.13.0 SystemOne Decisions API	1945	0.0142	0.0790	0.4581	25
GPT-5.6 Luna Responses API | Low reasoning	1871	0.0830	-	-	-
GPT-5.6 Luna Responses API | High reasoning	1938	0.0875	-	-	-
GPT-6 Luna Responses API | Low reasoning	1914	0.0998	-	-	-
GPT-6 Luna Responses API | High reasoning	1953	0.0956	-	-	-
GPT-6 Luna Decisions API | Numeric labels	1970	0.1609	0.3526	0.8827	9
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	1971	0.0564	0.0835	0.3791	14

Lower values are better for all three metrics. Jev has the lowest ECE and Brier. Clef has slightly lower log loss than descriptive/shuffled Decisions despite weaker top-1 accuracy. Log loss is sensitive to rare near-zero reference probabilities: Jev has 25 clipped cases; Clef has none.

Clef is conservative in its top bin: mean probability 0.934 versus observed accuracy 0.989. Responses Brier and log loss are unavailable because its returned top-three probability mass is incomplete.

Accuracy at different confidence thresholds

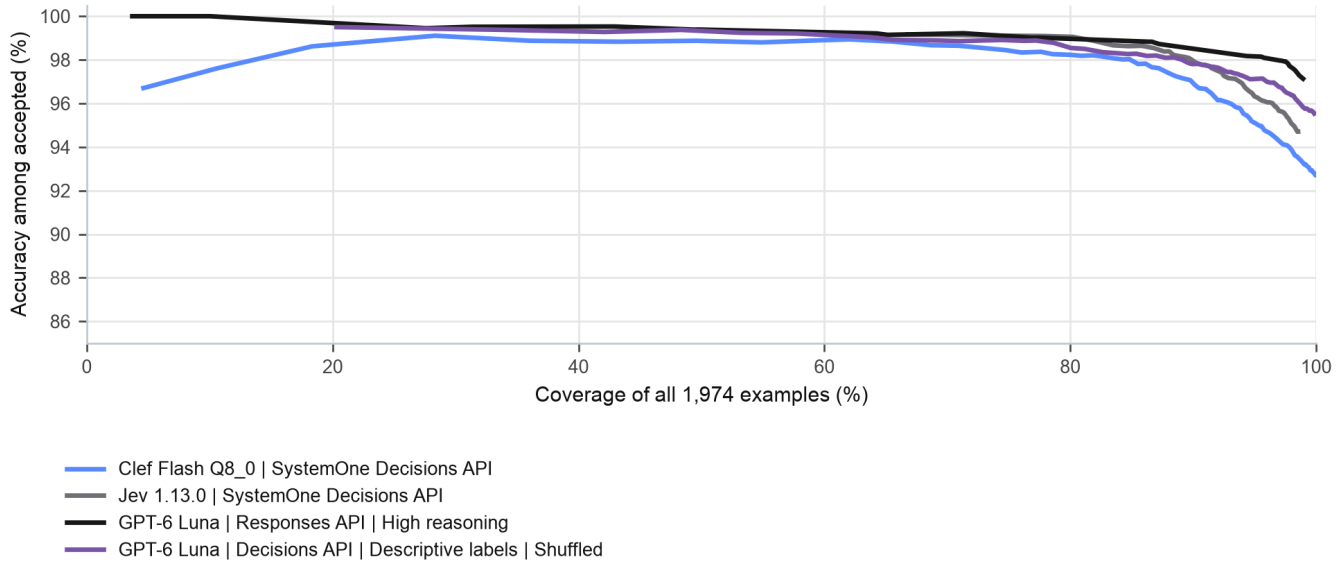


Figure 7. Accuracy among accepted outputs versus coverage of the full cohort as the top-label probability threshold changes. Curves omit thresholds retaining fewer than 50 cases; ties are accepted together. No post-hoc probability calibration is applied. Source: E1.

Clef threshold	Accepted	Coverage	Correct	Wrong	Accepted accuracy
0.50	1857	94.07%	1774	83	95.53%
0.80	1532	77.61%	1507	25	98.37%
0.90	977	49.49%	966	11	98.87%
0.95	209	10.59%	204	5	97.61%

At a threshold of 0.80, Clef returns 1,532 accepted classifications, of which 1,507 are correct and 25 are wrong. The other 442 examples would need review or another classification method. The 98.37% accepted accuracy applies only to the retained 77.61% of requests.

Raising the threshold did not always improve accuracy. The 0.95 cut accepts only 209 cases and yields 97.61% accuracy, below the 0.90 cut's 98.87%. Small selected subsets and imperfect calibration mean that increasingly high reported confidence need not produce monotonically better observed accuracy.

Choose the threshold by comparing the cost of accepted errors with the cost of escalation. Version it with the model, taxonomy and request format, then remeasure accuracy and coverage on the intended traffic before enabling automatic routing.

Operational implication

At the 0.80 threshold, Clef would route 1,532 requests and escalate 442, with 25 errors among those routed. We did not test whether a reviewer or second model would resolve the escalated cases correctly.

Latency favours native decision interfaces

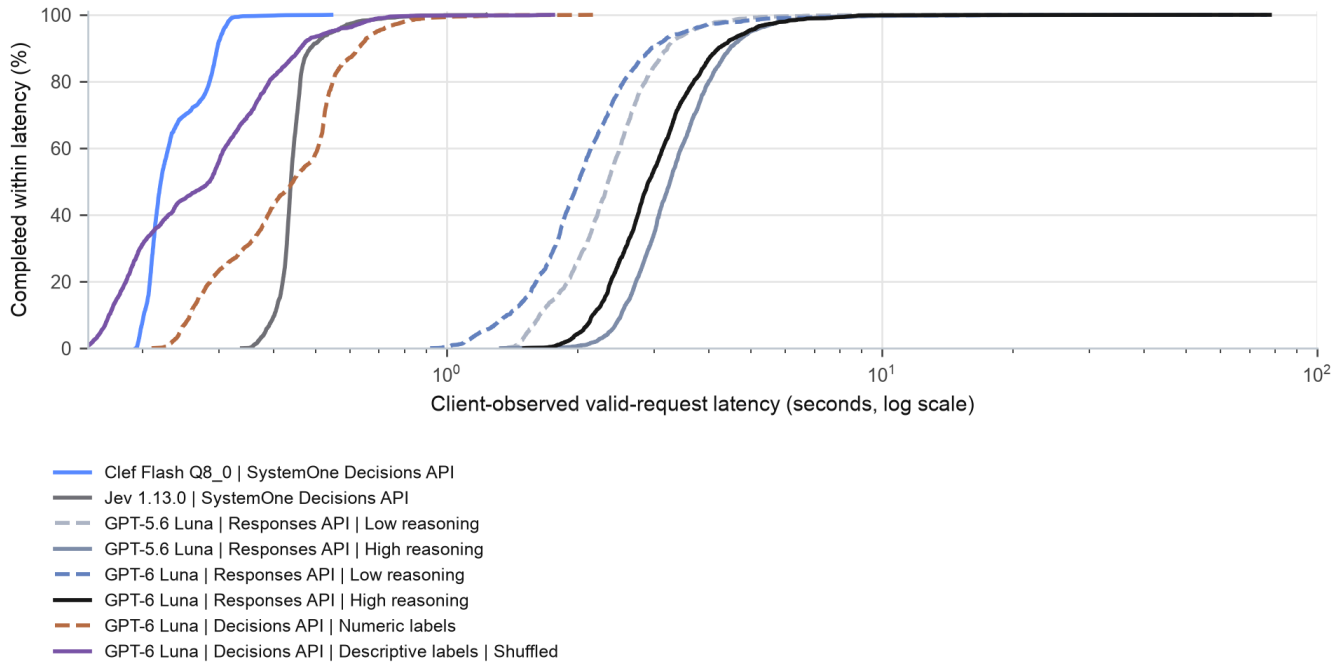


Figure 8. Empirical cumulative distribution of valid-request latency. A point at 95% is the p95 completion time. Logarithmic time axis; network and client overhead are included. Source: E1.

Configuration	Conc.	p50 s	p95 s	Mean s	Max s	Failed p50 s
Clef Flash Q8_0 SystemOne Decisions API	1	0.221	0.306	0.238	0.544	-
Jev 1.13.0 SystemOne Decisions API	2	0.439	0.550	0.449	1.237	0.440
GPT-5.6 Luna Responses API | Low reasoning	2	2.350	3.635	2.451	18.851	2.391
GPT-5.6 Luna Responses API | High reasoning	2	3.262	4.998	3.482	54.909	2.766
GPT-6 Luna Responses API | Low reasoning	2	2.010	3.542	2.232	76.766	2.122
GPT-6 Luna Responses API | High reasoning	2	2.912	4.926	3.177	77.886	2.730
GPT-6 Luna Decisions API | Numeric labels	1	0.444	0.691	0.447	2.184	0.525
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	4	0.286	0.541	0.301	1.758	0.229

Clef's median is 220.6 ms and p95 is 306.2 ms. In the recorded environments, GPT-6 Luna Responses high has a 13.2-times longer median and 16.1-times longer p95. Jev and original native Luna are around 0.44 seconds at the median; the descriptive/shuffled native run is 0.286 seconds.

The timing comparison includes differences in hardware, network and concurrency. Clef and original native Decisions were serial; historical conditions used concurrency two; the improved native run used pooling and concurrency four. Clef completed the cohort in 496.56 seconds (3.975 decisions/s). The descriptive/shuffled native run took 304.02 seconds including its inspection gap. Wall time and summed concurrent request time are different quantities.

Cost of the completed classification runs

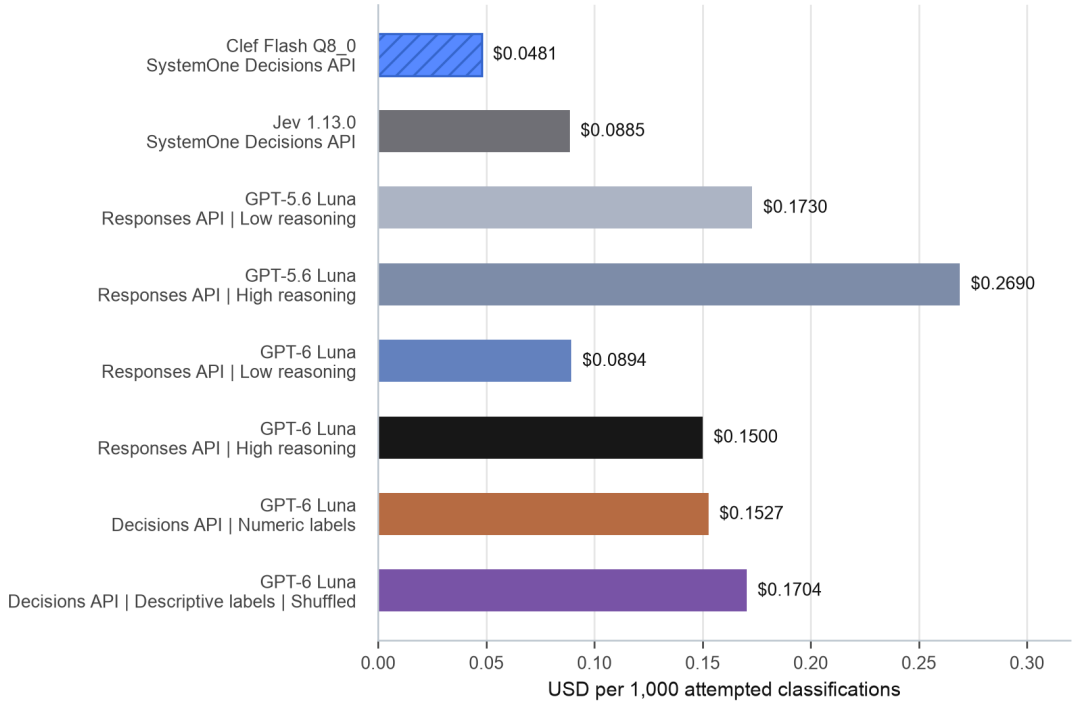


Figure 9. Cost per 1,000 attempts, including failed outputs. Hosted bars use saved usage and rates; the hatched Clef bar uses estimated electricity plus depreciation for its recorded run. See the next page for full local ownership costs. Sources: E1-E5.

Configuration	Full 1,974 run	Per 1,000 attempts	Per 1,000 correct
Clef Flash Q8_0 SystemOne Decisions API	$0.094867	$0.0481	$0.0518
Jev 1.13.0 SystemOne Decisions API	$0.174740	$0.0885	$0.0949
GPT-5.6 Luna Responses API | Low reasoning	$0.341465	$0.1730	$0.1925
GPT-5.6 Luna Responses API | High reasoning	$0.530967	$0.2690	$0.2870
GPT-6 Luna Responses API | Low reasoning	$0.176424	$0.0894	$0.0959
GPT-6 Luna Responses API | High reasoning	$0.296198	$0.1500	$0.1561
GPT-6 Luna Decisions API | Numeric labels	$0.301503	$0.1527	$0.1991
GPT-6 Luna Decisions API | Descriptive labels | Shuffled	$0.336334	$0.1704	$0.1786

Clef's run cost an estimated **$0.0949**: $0.0787 in depreciation and $0.0161 in electricity. The calculation uses its recorded 496.56 seconds, a $15,000 machine written down over three years, 650 W active power, $0.15/kWh and a 1.2 facility-energy multiplier.

Power and tariff are assumptions. Only the run's elapsed time is allocated here; idle capacity, maintenance and operator costs follow on page 17. Hosted costs use exact token-price products before accounting rounding. Cost per correct output divides by the observed correct count; business-error and shared application costs are excluded.

Full ownership cost of a $15,000 workstation

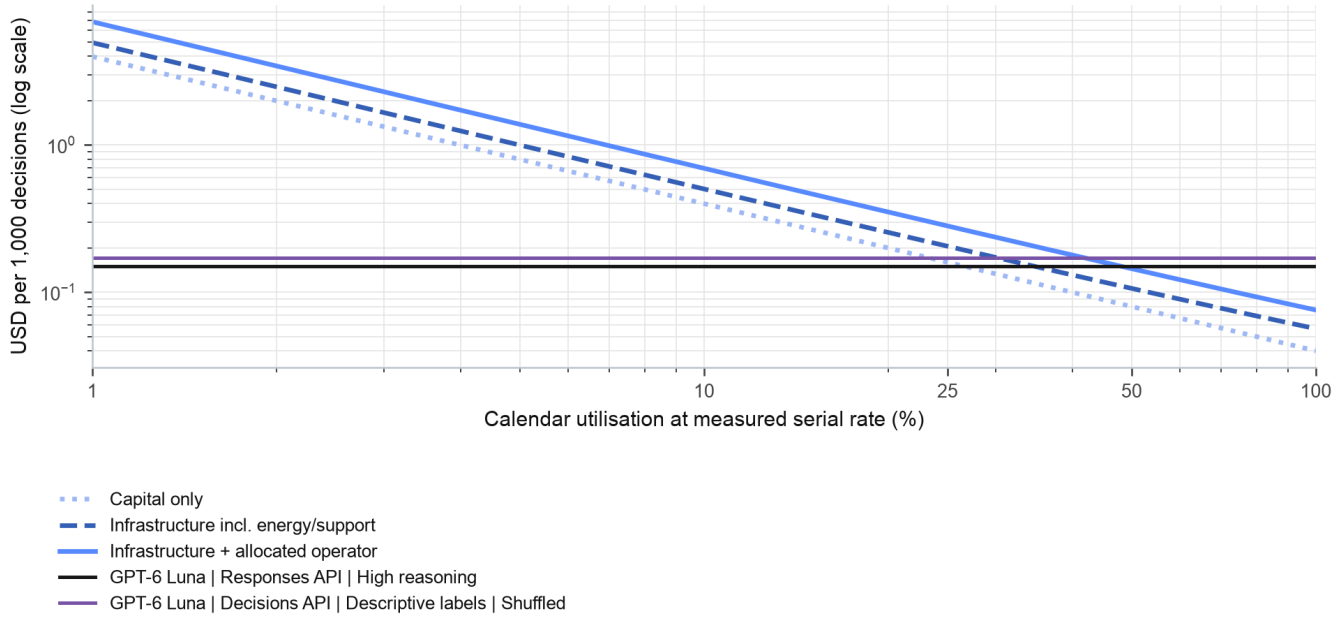


Figure 10. Modeled unit cost against calendar utilisation at the measured 3.975 decisions/s. Horizontal lines are observed hosted API costs. This is a capacity-based cost projection, not a year-long load test. Source: E5.

Utilisation	Annual volume	Capital / 1k	Infrastructure / 1k	Incl. operator / 1k
1%	1.25m	$3.988	$4.959	$6.873
10%	12.54m	$0.399	$0.502	$0.694
25%	31.34m	$0.160	$0.205	$0.282
50%	62.68m	$0.080	$0.106	$0.144
75%	94.03m	$0.053	$0.073	$0.098
100%	125.37m	$0.040	$0.056	$0.076

Capital: $15,000, straight-line over three years, zero resale value: $5,000/year. **Operating assumptions:** 100 W idle, 650 W active; $0.15/kWh; 1.2 facility-energy multiplier; maintenance at 5% of capital/year; $300/year space/network allocation; two operator hours/month at $100/hour. Power is a planning assumption, not measured wall draw. The electricity allowance is close to the recent US commercial average. [P9]

With all stated allocations, annual cost is $8,694 at 10% utilisation and $9,041 at 50%. The corresponding three-year costs are $26,083 and $27,124. “Infrastructure” includes depreciation, energy, maintenance and space/network; the final column also includes allocated operator time.

Cost-only parity with descriptive/shuffled Luna Decisions occurs at about **42.0% calendar utilisation**, or 52.7 million decisions/year. Parity with GPT-6 Luna Responses high occurs near **48.0%**, or 60.1 million/year. These crossings compare serving cost, not equal accuracy or business value. The 100% case assumes continuous service with no idle time or downtime.

An enterprise H200 rack: explicit assumptions

We model a rack with **four eight-GPU H200 nodes: 32 GPUs in total**. This scale could support replicas and concurrent workloads; a small decision model may need only one GPU. Low demand would leave much of a dedicated rack idle while its ownership costs continued.

Planning item	Central assumption / basis
Compute capital	4 nodes x $400,000 = $1,600,000. The anchor is Lenovo's published $397,801.60 eight-H200 configuration, priced 15 June 2026. [P6]
Ancillary capital	$50,000 for rack/network/storage/integration allowance; total capital $1,650,000.
Depreciation	Three years, zero resale value: $550,000/year.
Physical and power envelope	NVIDIA DGX H200 reference: 8U and up to 10.2 kW per eight-GPU node. Four nodes plus 2 kW ancillary allowance = 42.8 kW; reserve 45 kW. [P7]
Datacentre	45 kW x $350/kW/month = $189,000/year, all-in power, cooling and space planning allowance. No extra PUE energy charge is added. [P8]
Hardware support	12% of server capital/year ($192,000), plus 5% of ancillary capital ($2,500). These are planning allowances.
Operations and services	0.25 FTE at $150,000 fully loaded/year = $37,500; connectivity, monitoring and backup allowance $12,000/year.

Annual allocated cost

Capital allocation	Support	Datacentre	Operations / services	Total / year
$550,000	$194,500	$189,000	$49,500	$983,000

The central case totals **$2.949 million over three years**, or **$112.21 per calendar hour**. The price anchor and physical reference are from different vendor systems; this is not a single quoted SKU. A 45 kW rack needs a bespoke high-density facility design and commercial quote.

Public colocation advertisements provide an order-of-magnitude check, not a procurement commitment. The cited enterprise price page describes a January 2027 opening; it does not establish current capacity. Taxes, financing, insurance, duplicate disaster-recovery hardware and optional commercial software are additional if required. Shared application engineering is excluded from both hosted and local comparisons.

Rack cost depends on delivered throughput

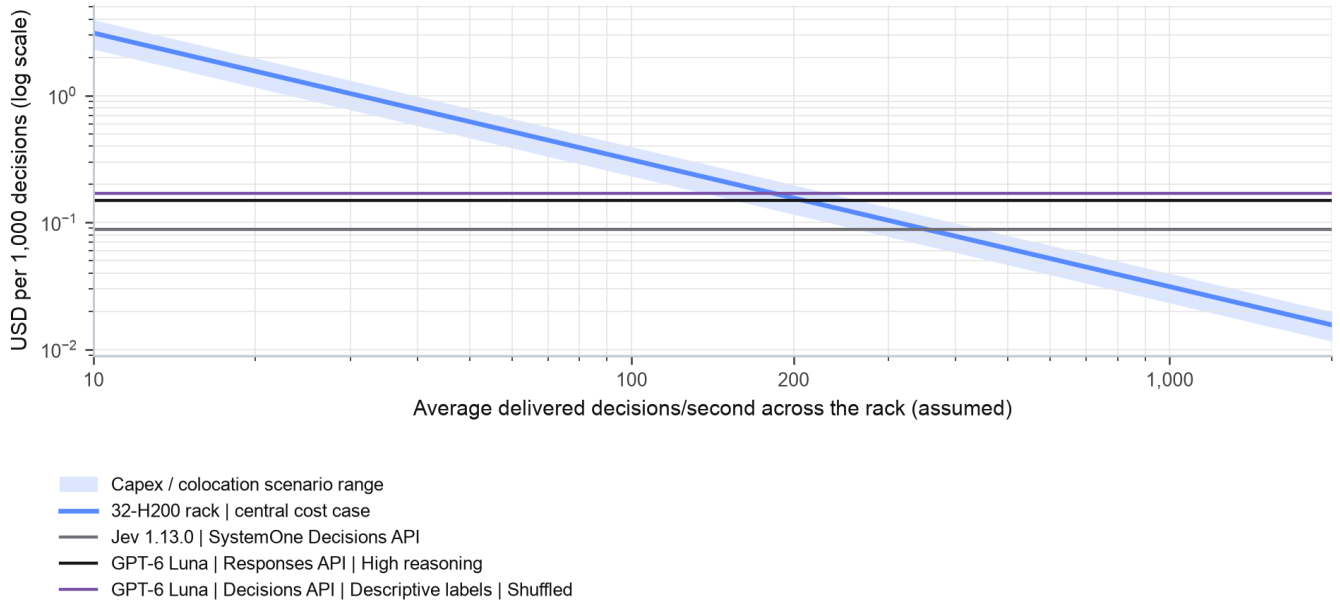


Figure 11. Conditional rack unit cost versus average delivered throughput. No H200 Clef throughput is measured in this study. Shading varies node capital from $300k-$500k and all-in colocation from $250-$450/kW/month; it is a scenario range, not a statistical interval. Source: E5.

Average delivered / s	Annual decisions	USD / 1,000	Busy rate at 50% use / s
10	0.315bn	$3.1171	20
50	1.577bn	$0.6234	100
100	3.154bn	$0.3117	200
200	6.307bn	$0.1559	400
500	15.768bn	$0.0623	1,000
1,000	31.536bn	$0.0312	2,000

For the central $983,000/year rack, cost-only parity requires approximately **183 average delivered decisions/s** against descriptive/shuffled Luna Decisions and **208/s** against GPT-6 Luna Responses high. At 50% useful utilisation, the serving stack would need roughly twice those busy-period rates. A load test would need to establish whether the serving stack can sustain those rates.

Higher-end hardware and a tuned serving stack can enable faster results and more concurrent requests. H200 offers 141 GB of GPU memory and 4.8 TB/s memory bandwidth per GPU, providing substantially more room for batching and replicas. Single-request latency does not scale inversely with GPU count, however, and this paper assigns no numerical H200 speedup to our cohort. [P7, P10]

For a shared rack, allocate capital and operations by a defensible capacity or usage share, and include reserved headroom. Calculate unit cost using delivered decisions under the required latency target. Across the stated capex and colocation sensitivity, three-year cost ranges from $2.243m to $3.655m; the deployment choice should be based on measured throughput under the required latency and availability targets.

Clef serving: the batch-size requirement

Clef could not process these decision requests with the default batch settings. Increasing the context window alone did not fix the problem: the physical batch also had to accommodate the complete rendered request, with enough capacity available to its prediction slot.

Setting	Initial configuration	Verified baseline
Context window	Not sufficient on its own	16,384 tokens
Physical batch	512 tokens	16,384 tokens
Evaluation batch	2,048 tokens	16,384 tokens
Prediction slots	4	1
GPU execution	Configuration-dependent	Full GPU offload; Flash Attention
Serving stack	LM Studio	0.4.28 build 2; runtime cuda12-avx2@2.67.0
Artifact	Stock Clef Flash	Clef-Flash-Q8_0.gguf; RTX 4090

Why the distinction is important

Clef uses non-causal decision evaluation: the complete rendered input must fit the effective physical batch used for the decision pass. A context limit expresses how much input the model can accept in principle; it does not guarantee that the selected batch and slot settings can evaluate that input in one pass. The default 512-token physical batch was below the roughly 2,000-token requests in this benchmark.

A smoke configuration with both batches set to 8,192 succeeded. The full baseline used 16,384 for physical batch, evaluation batch and context, with one prediction slot, to give conservative headroom against a byte-derived bound of 10,122 tokens. Actual reported input counts were 1,979-2,302, and every one of the 1,974 counts matched the independent reference tokenizer.

Batch size used in the baseline

The baseline used 16,384 as a capacity setting; each request actually contained 1,979-2,302 tokens. We did not establish the minimum required batch size or attempt a full benchmark at 8,192.

For another installation, size batches from the rendered schema and input, verify the effective configuration after loading, and test the largest intended requests without truncation. Re-evaluate concurrency after the serial path is correct. The measured artifact was the ggml-org Q8_0 GGUF release; the exact served bytes and embedded backend commit were not independently recovered, so no binary-identity claim is made beyond the recorded artifact and runtime. [E2, P1, P4]

Best Practice with OpenAI

GPT-6 Luna Decisions produced sharply different results for two representations of the same taxonomy. We preserved the original run, tested changes to label values and choice order, and checked the raw requests and responses for mapping errors.

GPT-6 Luna | Decisions API	Numeric labels, canonical order	Descriptive labels, shuffled
Correct / all	1,514 / 1,974	1,883 / 1,974
End-to-end accuracy	76.70%	95.39%
Top-three hit rate	96.56%	98.63%
Valid wrong / native refusals	456 / 4	88 / 3
Macro-F1	0.7527	0.9544
Exact usage-priced cost	$0.3015029	$0.3363345

The change produced **397 recoveries and 28 regressions**, a net gain of 369 correct answers: **+18.69 percentage points**, with a paired 95% component-bootstrap interval of +16.16 to +21.09 points. Unsuccessful outputs fell from 460 to 91, an 80.22% reduction. Both runs included every example in the cohort.

Exactly what changed

Choice values became unique lower-case category-name slugs instead of numeric strings, and all 50 value-description pairs were shuffled separately for each example. The prompt, context, classification policy and category descriptions remained unchanged. An independent audit checked all 98,700 category-description pairs in the frozen request plan.

The per-example permutation used a fixed seed and example identifier, never the reference category or a prior prediction. The plan was frozen before launch; no seed search was performed. The full cohort retained every example, including earlier failures. Complete probability distributions were kept, with no label-offset correction or probability repair.

The adjacent-ID error pattern

The original run contained 332 valid errors that selected exactly the next canonical category ID. The new format corrected 327 of those; only two next-ID errors remain among 88 valid wrong predictions. Raw selected values, mapped labels and probability maxima agreed in the original audit, so the evidence did not identify a local off-by-one scoring bug.

Both value encoding and order changed in the full-cohort intervention. Their separate contributions are not identified by that run. The following diagnostic tests the changes separately on a smaller set. Sources: E3-E4.

The diagnosis tested labels, order and count

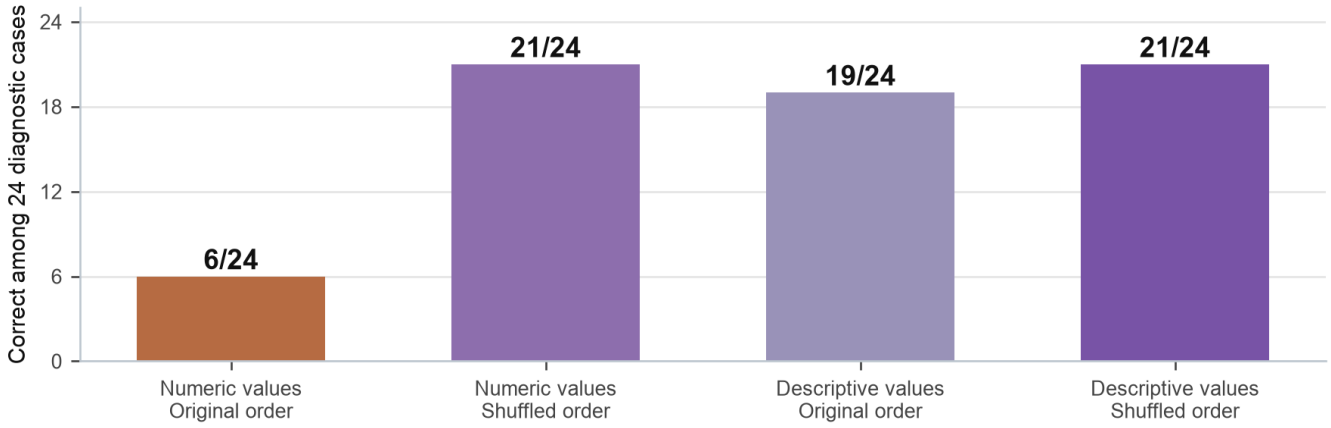


Figure 12. GPT-6 Luna, Decisions API: all four formats retain the full 50 choices. The diagnostic contains 24 deliberately selected cases: 12 adjacent-ID errors, six other errors and six correct controls. These are diagnostic counts, not benchmark accuracy estimates. Source: E3.

All 24 original-format predictions repeated exactly. Each modified full-50 condition fixed all 12 selected adjacent-ID errors without losing any of the six correct controls. Numeric-shuffled and descriptive-shuffled returned identical predictions and both scored 21/24; the diagnostic does not show that combining both changes adds to the benefit of shuffling alone.

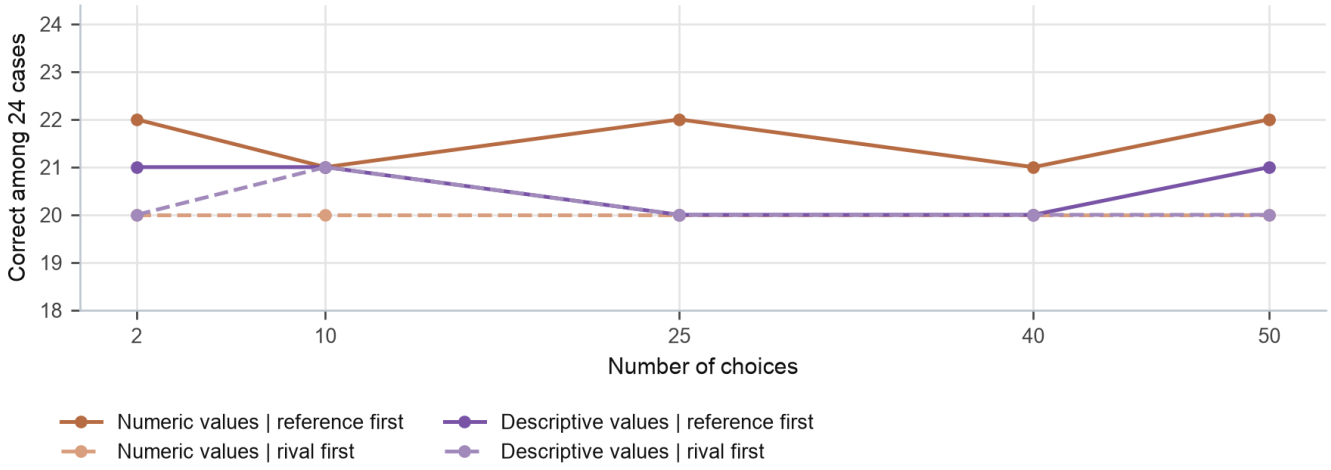


Figure 13. Correct diagnostic cases as the number of choices changes. Each set retains the reference and a rival in positions one and two, swapping their order in paired sweeps. The rival is the original wrong choice, or the highest-probability alternative for correct controls. All four sweeps use the same 24 cases. Source: E3.

The two-choice and fifty-choice endpoints have identical case-level correctness within each sweep. There is no monotonic deterioration as choices increase. This does not support a simple “50 choices is too many” explanation for the selected failures; changing order while retaining 50 choices already recovered most of them.

The reduced sets use the reference label to retain candidates, so they are not deployable benchmark conditions. Count also changes distractor content and input length; the two retained candidates occupy only the first two positions. The diagnostic narrows the explanation to representation sensitivity without ruling out every possible count-position interaction. All 576 diagnostic requests completed validly, with no retries or refusals.

Plausible explanations

The experiment establishes sensitivity to label representation and choice order, but it cannot establish which internal model, tokenisation or API mechanism caused it. The following hypotheses suggest tests that could distinguish the explanations.

Hypothesis	Why it fits the observation	What would distinguish it
Semantic binding	A descriptive output label names the task directly. A numeric ID requires the model to associate an arbitrary identifier with its description. Value changes alone improved 6/24 to 19/24.	Compare meaningful labels with opaque nonnumeric labels, matched for length, across independent tasks.
Order or position sensitivity	Shuffling numeric choices improved 6/24 to 21/24 with all 50 options present. Changing the order alone changed the predictions for these requests.	Repeat multiple frozen permutations; measure stability by class and position without selecting the best seed.
Representation interaction	The next-ID pattern is consistent with a binding, tokenisation or serialisation interaction. Local raw-to-label decoding agreed with the provider output.	Hold position fixed while varying ID encoding, label length and description placement; retain exact request/response traces.

Numeric strings are valid under the published API schema. The observed pattern warrants investigation, but does not establish a provider bug. Subtracting one from every predicted ID could damage other answers; a fixed shuffle also needs testing across independent examples and permutations. [P2-P3]

Reasoning controls and model aliases

The Responses API conditions explicitly requested low or high reasoning. Native Decisions exposes no reasoning-effort control in the current schema. Its zero reported reasoning tokens must not be interpreted as proof that it performs no internal computation. Similarly, the shared public alias gpt-6-luna does not prove that the two APIs follow the same computation path or that historical and current serving snapshots are identical.

The improved Decisions condition finishes 14 correct outputs behind Responses high: -0.71 points, with a paired interval spanning zero. The test therefore does not establish an accuracy difference or formal equivalence. Its lower latency warrants evaluation on client traffic using the same request format.

All 24 shared descriptive/shuffled diagnostic requests produced the same predictions in the full run. This small repeat check does not establish determinism across other requests.

Structuring and validating decision queries

Version the complete decision request, including choice values and order. This study changed those two fields while leaving category descriptions unchanged. Any later description changes should be tested separately.

Practice	Implementation consequence
Define the target precisely	State whether the question classifies the current action, intent, topic or outcome. Include precedence rules for overlapping workflows and completed earlier tasks.
Use meaningful stable values	Prefer unique semantic labels with an explicit mapping to internal IDs. Decode the returned choice value exactly; never infer an index from its array position.
Preserve pairs when permuting	Shuffle the complete value-description pair. Freeze an outcome-independent order or deterministic seed. Never use reference answers to choose a seed.
Test representation stability	Compare canonical and permuted formats, opaque and meaningful labels, and repeat payloads. Report recoveries and regressions, not only net improvement.
Separate evidence from instructions	Keep prompt/context as evidence; keep references and private metadata outside the model input. Specify how quoted instructions and prior completed work should be treated.
Validate the returned answer	Check model and question identities, allowed values, probability bounds and mass, refusal state and usage. Preserve invalid outputs instead of repairing scores silently.
Measure selective risk	Validate top-label probability against observed accuracy and coverage. Keep thresholds tied to the precise taxonomy, request format and deployment.
Optimise transport separately	Reuse HTTP connections and set concurrency intentionally. Measure request latency, queueing and whole-run throughput independently.
Apply caching to eligible input	Shared-input prefixes must contain at least 1,024 tokens; repeated taxonomy in question choices does not automatically qualify. A relocation of taxonomy text is a new prompt condition.

Example: mapping a label to an internal ID

For example, canonical category 29 maps to `multilingual_customer_workflow`. Its definition remains “translate, localise and preserve intent across customer service or sales interactions.” Keep that mapping independent of display order. The selected semantic value is mapped back to 29 only after the native answer passes validation.

The tested shuffle starts from sorted category IDs, hashes the fixed seed 20261102 plus the example identifier plus “shuffle”, uses the first 16 hexadecimal digits as a pseudorandom seed, and permutes all 50 pairs. The fixed seed makes the order reproducible; it was chosen independently of outcomes. Descriptions should still express category boundaries clearly; validate any later wording change as a separate condition. [E4, P2-P3]

Where Clef fits in an enterprise application

Clef is worth evaluating when request data must stay within the enterprise or routing requires low latency. It returned valid answers throughout this test and classified 92.71% correctly. Errors were more frequent at workflow boundaries. The two strongest hosted configurations correctly classified 53 and 67 more requests. A local deployment also needs enough demand to justify its ownership cost.

Account for the cost of routing errors

At a fixed serving volume, an extra few hundredths of a dollar per thousand API calls can be immaterial beside the cost of routing a request to the wrong authority or triggering a manual investigation. Conversely, a small accuracy advantage may not justify sending sensitive request content outside an approved boundary. The acceptable balance depends on the client's data restrictions and the consequences of a wrong route.

Application	How to use these results
Advisory routing	Compare top-three quality, acceptance rate and user correction cost; Clef's 97.42% top-three result may be useful where a person confirms the route.
Automatic low-consequence routing	Use end-to-end success plus confidence-gated coverage; define a safe handling path for ambiguous or invalid outcomes.
High-consequence workflow selection	Require stronger category-specific evidence, explicit authorisation rules and downstream validation. A correct class does not prove an action is permitted.
Local service procurement	Benchmark the intended serving stack and volume, then apply fully allocated cost. Purchase capacity for an observed requirement, not an assumed speedup.

Research questions worth pursuing

Independent operational samples would establish transfer from a balanced scenario set to real request frequencies. Repeated permutations and label encodings would quantify representation stability. Sustained load tests would measure concurrency, tail latency, power draw and availability. Hardware and precision comparisons would determine how the specific Q8_0 result transfers to other serving configurations. Category-specific review and calibrated escalation would help align model uncertainty with the cost of enterprise mistakes.

We would shortlist Clef for local routing and advisory classification. For automatic routing, the next evaluation should measure errors on the client's categories and traffic, including how uncertain cases are handled. The hosted results provide alternatives where higher accuracy matters more than local deployment.

AI One's conclusion

Choose the deployment using accuracy on client traffic, the cost of routing errors, latency and full ownership cost. Retain the tested query format and a defined handling path for uncertain or invalid answers.

The ContextOne sample taxonomy

The same 50 category names and definitions were supplied to every configuration. These definitions specify the sample benchmark task; they do not describe a production deployment. Category IDs are stable identifiers; they do not express an ordinal scale.

ID	Workflow	Definition
01	Enterprise request intake and routing	classify a request, determine owner, priority, SLA and next action.
02	Ticket triage and resolution	handle IT, HR, finance, legal or operations tickets end-to-end where possible.
03	Policy-aware approval routing	decide whether a request needs approval, who can approve it, and what evidence is required.
04	SOP execution agent	follow a documented operating procedure, collect required inputs, execute steps, and escalate exceptions.
05	Knowledge Q&A with citations	answer from internal documents, policies, tickets, contracts, code or wikis with source grounding.
06	Research brief generation	collect information from multiple sources, compare, synthesise and produce a structured briefing.
07	Meeting-to-actions workflow	transcribe, summarise, identify decisions, create tasks, update systems and chase owners.
08	Email/thread action workflow	read an email thread, infer the required action, draft replies, file records and update systems.
09	Document drafting workflow	produce first drafts of memos, reports, proposals, policies, board notes or client communications.
10	Document redlining and comparison	compare versions, identify material changes, classify risks and propose edits.
11	Document extraction to structured data	extract fields from PDFs, contracts, invoices, forms, emails, scans or images.
12	Form completion and submission	populate enterprise forms from available context and submit through workflow tools.
13	System-of-record update workflow	update CRM, ERP, HRIS, ITSM, ticketing, billing or compliance systems after validation.
14	Data reconciliation workflow	compare records across systems, identify mismatches, propose matches and resolve exceptions.
15	Master data maintenance	clean customers, suppliers, products, employees, assets or instruments across systems.
16	Exception investigation workflow	detect an exception, gather evidence, classify root cause and propose resolution.
17	Escalation and handoff workflow	decide when human judgement is needed, prepare the case file and route to the right person.
18	Status reporting workflow	convert live system state into project, operational, financial or executive status reports.
19	Audit trail generation	log decisions, evidence, tool calls, approvals, sources and rationale for later review.
20	Workflow monitoring and recovery	watch a process, detect stalled items, retry safe steps and alert owners.
21	Customer support self-resolution	answer customer requests, perform account actions and close cases where policy permits.
22	Technical support troubleshooting	diagnose product or service issues, guide fixes, collect logs and escalate if needed.
23	Customer case prioritisation	classify urgency, customer value, sentiment, regulatory sensitivity and SLA exposure.
24	Refund, return and credit handling	validate eligibility, calculate credit, initiate refund and document decision.
25	Complaint handling and de-escalation	classify complaints, draft responses, trigger remediation and flag regulatory risk.

Source: frozen ContextOne 50-category sample taxonomy, E1. The task asks for one primary workflow even where related actions appear in context.

The ContextOne sample taxonomy (continued)

The same 50 category names and definitions were supplied to every configuration. These definitions specify the sample benchmark task; they do not describe a production deployment. Category IDs are stable identifiers; they do not express an ordinal scale.

ID	Workflow	Definition
26	Customer onboarding workflow	collect documents, verify completeness, perform checks and open the account or service.
27	Customer health and churn intervention	monitor signals, identify risk, recommend or trigger outreach.
28	Proactive incident communication	detect service disruption, identify affected customers and send tailored updates.
29	Multilingual customer workflow	translate, localise and preserve intent across customer service or sales interactions.
30	Voice-of-customer mining	cluster complaints, reviews, calls and surveys into themes, root causes and action items.
31	Lead enrichment and scoring	research prospects, enrich CRM records and rank outreach priority.
32	Account research and call prep	produce briefings on company, people, history, risks, opportunities and recommended agenda.
33	Sales outreach sequencing	draft tailored emails, LinkedIn messages, follow-ups and call scripts.
34	RFP/RFI response workflow	read requirements, map capabilities, draft answers, collect evidence and flag gaps.
35	Proposal generation workflow	assemble pricing, scope, case studies, assumptions, risks and implementation plan.
36	CRM hygiene workflow	detect missing fields, duplicate accounts, stale opportunities and next-step inconsistencies.
37	Quote and CPQ support	configure products, check eligibility, apply rules and prepare quote packages.
38	Discount and pricing approval	assess margin, precedent, customer value and approval path.
39	Renewal and upsell trigger workflow	identify renewal risk, expansion signals and recommended commercial actions.
40	Campaign planning and content workflow	generate campaign plans, assets, variants, approvals and localisation.
41	Personalisation and segmentation workflow	assign customers to segments and generate tailored offers or content.
42	Competitive intelligence monitoring	track competitors, products, pricing, hiring, filings and messaging changes.
43	E-commerce recommendation workflow	act as a guided shopping or product-selection agent.
44	Marketing compliance review	check claims, disclosures, regulated language and brand standards before publication.
45	Code change implementation	inspect repo, plan changes, edit files, run tests and prepare a PR.
46	Code review workflow	review PRs for correctness, maintainability, tests, security and style.
47	Test generation and maintenance	infer missing tests, generate cases, update fixtures and run regressions.
48	Build and CI/CD failure diagnosis	inspect logs, identify failure cause, propose or apply fixes.
49	Production incident triage	ingest alerts, logs, traces and deployments, then propose root cause and mitigation.
50	Runbook execution workflow	execute operational recovery steps with checkpoints and human approval gates.

Source: frozen ContextOne 50-category sample taxonomy, E1. The task asks for one primary workflow even where related actions appear in context.

Per-category results

Condition key: C = Clef Flash Q8_0, SystemOne Decisions API; J = Jev 1.13.0, SystemOne Decisions API; 5L/5H = GPT-5.6 Luna, Responses API, Low/High reasoning; 6L/6H = GPT-6 Luna, Responses API, Low/High reasoning; DN = GPT-6 Luna, Decisions API, Numeric labels; DS = GPT-6 Luna, Decisions API, Descriptive labels, Shuffled.

Each cell reports **correct count / F1**. Support is the number of reference examples for the category. Recall is correct / support. F1 combines that recall with precision, including false positives arriving from other categories and false negatives from invalid outputs.

ID	n	C	J	5L	5H	6L	6H	DN	DS
01	40	40 / 0.94	39 / 0.99	40 / 1.00	39 / 0.99	38 / 0.97	40 / 1.00	40 / 0.94	39 / 0.99
02	40	26 / 0.70	13 / 0.49	27 / 0.79	27 / 0.79	28 / 0.82	31 / 0.86	36 / 0.81	32 / 0.85
03	40	38 / 0.96	40 / 0.99	37 / 0.93	39 / 0.97	40 / 0.98	39 / 0.97	40 / 0.95	40 / 0.99
04	40	35 / 0.88	34 / 0.91	34 / 0.86	34 / 0.87	35 / 0.92	35 / 0.91	37 / 0.84	35 / 0.92
05	40	37 / 0.95	39 / 0.97	39 / 0.95	40 / 0.95	40 / 0.99	40 / 0.99	40 / 0.98	40 / 0.96
06	40	39 / 0.95	38 / 0.96	38 / 0.95	40 / 0.99	40 / 0.99	40 / 0.99	40 / 0.89	38 / 0.96
07	40	35 / 0.92	35 / 0.92	21 / 0.69	28 / 0.82	22 / 0.71	33 / 0.90	35 / 0.93	33 / 0.90
08	35	35 / 0.92	35 / 0.96	30 / 0.90	34 / 0.94	29 / 0.87	34 / 0.96	35 / 0.95	35 / 0.96
09	40	36 / 0.92	38 / 0.94	40 / 0.95	38 / 0.92	39 / 0.96	38 / 0.95	40 / 0.78	38 / 0.88
10	40	40 / 1.00	40 / 0.99	38 / 0.96	39 / 0.99	40 / 0.99	40 / 0.99	40 / 0.99	40 / 1.00
11	40	40 / 0.96	39 / 0.96	37 / 0.93	40 / 0.96	40 / 0.96	40 / 0.99	39 / 0.99	38 / 0.97
12	34	30 / 0.92	31 / 0.94	34 / 0.97	34 / 0.99	33 / 0.97	34 / 0.99	30 / 0.92	33 / 0.97
13	40	37 / 0.80	34 / 0.73	35 / 0.81	36 / 0.80	35 / 0.83	36 / 0.88	28 / 0.79	31 / 0.81
14	40	40 / 0.92	40 / 0.85	36 / 0.89	38 / 0.88	39 / 0.91	40 / 0.94	40 / 0.95	38 / 0.94
15	40	39 / 0.94	33 / 0.85	38 / 0.87	37 / 0.87	40 / 0.89	37 / 0.91	38 / 0.86	39 / 0.89
16	40	38 / 0.84	38 / 0.95	38 / 0.87	37 / 0.91	39 / 0.91	38 / 0.90	39 / 0.86	39 / 0.90
17	40	39 / 0.99	37 / 0.95	35 / 0.93	40 / 1.00	36 / 0.95	40 / 1.00	39 / 0.99	40 / 0.99
18	40	40 / 0.96	40 / 0.96	39 / 0.97	40 / 1.00	40 / 0.99	40 / 0.99	39 / 0.97	40 / 0.94
19	40	38 / 0.97	40 / 1.00	35 / 0.93	36 / 0.95	35 / 0.93	40 / 1.00	39 / 0.99	39 / 0.99
20	40	40 / 0.95	40 / 0.96	38 / 0.97	38 / 0.97	38 / 0.97	39 / 0.99	40 / 0.96	40 / 0.98
21	40	28 / 0.82	28 / 0.81	35 / 0.92	32 / 0.88	35 / 0.92	39 / 0.97	33 / 0.89	40 / 0.95
22	40	32 / 0.83	39 / 0.88	36 / 0.90	38 / 0.92	40 / 0.96	38 / 0.93	32 / 0.81	36 / 0.91
23	40	38 / 0.95	39 / 0.99	35 / 0.93	39 / 0.99	37 / 0.96	38 / 0.97	38 / 0.97	38 / 0.97
24	40	33 / 0.86	36 / 0.94	32 / 0.86	38 / 0.95	34 / 0.91	39 / 0.97	37 / 0.96	39 / 0.96
25	40	40 / 0.98	38 / 0.97	38 / 0.96	39 / 0.99	38 / 0.97	38 / 0.97	31 / 0.85	38 / 0.97

The full taxonomy on pages 26-27 supplies names and definitions for every ID. A class with perfect recall can still have F1 below 1.00 when other examples are incorrectly assigned to it. With 34-40 examples per class, one additional correct answer moves recall by roughly 2.5-2.9 percentage points; interpret small class differences accordingly.

Per-category results (continued)

Condition key: C = Clef Flash Q8_0, SystemOne Decisions API; J = Jev 1.13.0, SystemOne Decisions API; 5L/5H = GPT-5.6 Luna, Responses API, Low/High reasoning; 6L/6H = GPT-6 Luna, Responses API, Low/High reasoning; DN = GPT-6 Luna, Decisions API, Numeric labels; DS = GPT-6 Luna, Decisions API, Descriptive labels, Shuffled.

Each cell reports **correct count / F1**. Support is the number of reference examples for the category. Recall is correct / support. F1 combines that recall with precision, including false positives arriving from other categories and false negatives from invalid outputs.

ID	n	C	J	5L	5H	6L	6H	DN	DS
26	34	26 / 0.84	30 / 0.88	26 / 0.84	26 / 0.87	30 / 0.94	28 / 0.90	23 / 0.78	30 / 0.94
27	40	38 / 0.96	39 / 0.97	36 / 0.91	38 / 0.95	36 / 0.92	39 / 0.96	36 / 0.87	39 / 0.97
28	40	40 / 0.98	40 / 1.00	37 / 0.96	39 / 0.99	40 / 1.00	39 / 0.99	38 / 0.93	39 / 0.97
29	40	37 / 0.95	35 / 0.92	37 / 0.95	38 / 0.96	39 / 0.97	39 / 0.97	12 / 0.45	37 / 0.95
30	40	40 / 1.00	40 / 1.00	39 / 0.99	37 / 0.96	39 / 0.99	40 / 1.00	35 / 0.73	40 / 1.00
31	40	40 / 1.00	38 / 0.97	33 / 0.90	38 / 0.97	38 / 0.97	40 / 1.00	14 / 0.48	40 / 1.00
32	40	34 / 0.89	34 / 0.91	33 / 0.90	34 / 0.91	35 / 0.92	35 / 0.93	1 / 0.03	34 / 0.92
33	40	39 / 0.94	40 / 0.96	39 / 0.96	39 / 0.99	40 / 1.00	40 / 1.00	31 / 0.60	39 / 0.96
34	40	40 / 0.99	40 / 1.00	30 / 0.86	39 / 0.99	38 / 0.97	40 / 1.00	35 / 0.83	40 / 1.00
35	35	34 / 0.94	34 / 0.96	35 / 0.93	33 / 0.90	34 / 0.94	35 / 0.97	16 / 0.56	35 / 1.00
36	40	34 / 0.92	34 / 0.92	28 / 0.82	32 / 0.89	28 / 0.82	31 / 0.87	28 / 0.65	31 / 0.87
37	40	37 / 0.94	39 / 0.97	35 / 0.93	37 / 0.96	38 / 0.97	40 / 1.00	10 / 0.40	40 / 0.99
38	40	39 / 0.99	39 / 0.99	31 / 0.87	39 / 0.99	37 / 0.96	39 / 0.99	22 / 0.49	39 / 0.99
39	40	40 / 0.98	40 / 0.99	35 / 0.93	39 / 0.99	36 / 0.95	39 / 0.99	12 / 0.36	40 / 0.99
40	40	36 / 0.94	38 / 0.95	34 / 0.92	37 / 0.95	40 / 1.00	40 / 0.99	7 / 0.18	39 / 0.97
41	40	39 / 0.99	40 / 1.00	38 / 0.97	40 / 1.00	39 / 0.99	40 / 1.00	32 / 0.63	40 / 0.98
42	40	40 / 0.99	39 / 0.97	40 / 1.00	40 / 1.00	39 / 0.99	40 / 1.00	8 / 0.29	39 / 0.97
43	40	30 / 0.86	36 / 0.95	39 / 0.99	39 / 0.99	38 / 0.97	39 / 0.99	9 / 0.23	37 / 0.96
44	40	35 / 0.90	35 / 0.93	33 / 0.89	33 / 0.90	33 / 0.90	35 / 0.93	33 / 0.65	35 / 0.92
45	40	36 / 0.94	40 / 0.98	38 / 0.95	39 / 0.99	39 / 0.97	40 / 1.00	31 / 0.85	40 / 0.99
46	40	39 / 0.97	39 / 0.99	39 / 0.99	39 / 0.99	40 / 1.00	40 / 1.00	40 / 0.89	40 / 1.00
47	40	31 / 0.86	36 / 0.94	38 / 0.96	40 / 0.99	39 / 0.97	40 / 0.99	39 / 0.97	39 / 0.97
48	40	40 / 0.92	40 / 0.99	40 / 0.99	40 / 0.99	40 / 0.99	39 / 0.97	37 / 0.95	39 / 0.97
49	40	38 / 0.92	39 / 0.99	40 / 1.00	40 / 1.00	40 / 1.00	40 / 1.00	10 / 0.38	40 / 0.99
50	36	35 / 0.88	35 / 0.89	36 / 0.88	35 / 0.85	35 / 0.86	34 / 0.88	30 / 0.62	34 / 0.87

The full taxonomy on pages 26-27 supplies names and definitions for every ID. A class with perfect recall can still have F1 below 1.00 when other examples are incorrectly assigned to it. With 34-40 examples per class, one additional correct answer moves recall by roughly 2.5-2.9 percentage points; interpret small class differences accordingly.

Recall across all 50 categories

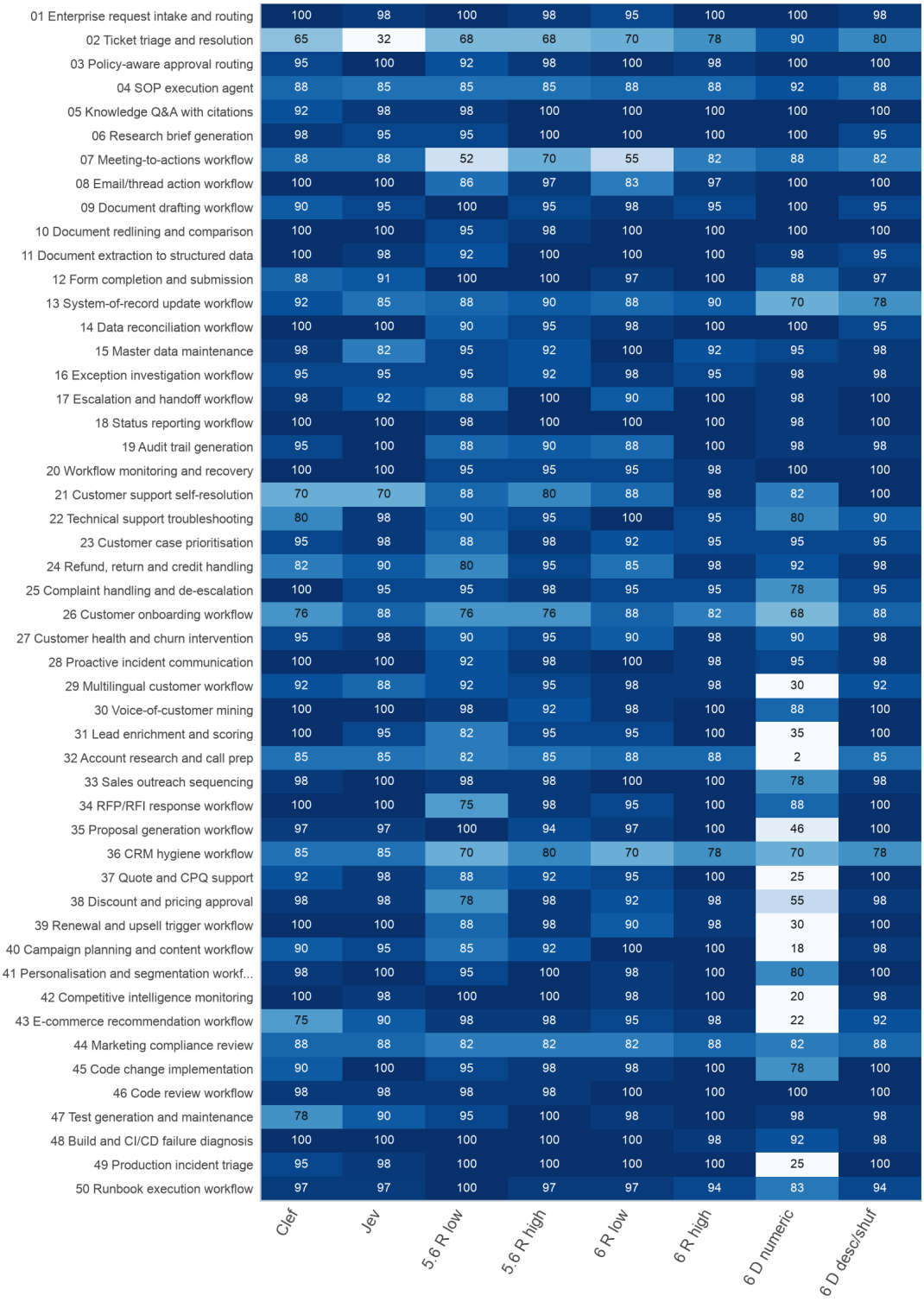


Figure 14. Category recall (%) across all eight configurations; invalid outputs count as misses. Each cell uses the class support on pages 28-29. Colour scale is 40-100%, with values below 40% saturated at the light end; printed values retain the exact rounded percentage. Source: E1.

Columns: Clef = Clef Flash Q8_0 / SystemOne Decisions API; Jev = Jev 1.13.0 / SystemOne Decisions API; 5.6 R low/high = GPT-5.6 Luna / Responses API / Low or High reasoning; 6 R low/high = GPT-6 Luna / Responses API / Low or High reasoning; 6 D numeric = GPT-6 Luna / Decisions API / Numeric labels; 6 D desc/shuf = GPT-6 Luna / Decisions API / Descriptive labels / Shuffled.

Evidence and reproducibility

All results were recomputed from frozen per-example run evidence. All eight condition scores, class statistics, latency quantiles and confidence-bin calculations were independently recomputed for this paper. The supporting offline analysis passed 2,480 assertions against the saved comparison evidence. No new benchmark inference or training was performed to produce this report.

ID	Evidence retained by AI One
E1	Eight-condition comparison: luna-decisions-shuffled-2026-11-02. Frozen cohort, comparison-metrics.json and comparison-drilldown.json; per-example predictions, status and timing for all 1,974 examples. Paper analysis: paper-statistics.json and reproducible build_statistics.py.
E2	Stock Clef baseline: clef-stock-2026-11-01. Raw request/response attempts, runtime checks, tokenizer verification and wall timing. Technical narrative: baseline-run-2026-11-01.md.
E3	Original native baseline: luna-decisions-2026-11-01-r2. Format diagnostic: luna-diagnostic-2026-11-02, including the frozen 576-call design and all raw outcomes.
E4	Descriptive/shuffled native baseline: luna-decisions-shuffled-2026-11-02. Frozen 1,974-request plan, explicit value mappings, raw distributions, usage, paired analysis and independent verification.
E5	Paper economics model: economics.json and build_economics.py. Assumed workstation capital and disclosed operating allowances; current vendor sources P6-P9; scenario arithmetic independently checked.

Original and descriptive/shuffled Decisions used 3,015,029 and 3,363,345 reported input tokens respectively, with no reported cache reads, writes, output or reasoning tokens. At $0.10 per million input tokens, exact costs are $0.3015029 and $0.3363345. Conservative per-request accounting totals were $0.302408 and $0.337206. [P2-P3]

Frozen identity and statistical recipe

Dataset SHA-256:
`7969453e947ef97af6f70e5a822b04845bfb8d54c1f14c6af11659bb8e8ee7ec`

Classification policy SHA-256:
`fcda2d2f24b226e344fad04280aed0dfc8cce44c4ef87fed76bbdbbef1e725d8`

Descriptive/shuffled request-plan SHA-256:
`ba320ba67d44c4fbd2601a228f8eb1874243490f6ee26865f5b2c2d072e5a90f`

Bootstrap: 5,000 draws; 982 whole scenario components; row-weighted scores; common seed 20261102; NumPy PCG64; percentile 2.5%/97.5% intervals. The original report's earlier Clef comparison intervals used seed 20261101. This paper consistently uses its independently recomputed set; small endpoint differences are Monte Carlo variation, not a changed cohort.

Accounting: exact native token-price products are used for economic comparisons; conservative per-request rounded totals remain disclosed. Local power, utilisation and operator allowances are scenarios. H200 throughput is unmeasured. The comparison is conditional on the fixed provisional references and the declared synthetic test mix; it is not a production error-rate guarantee.

External technical and cost references

Public sources were checked on 2 November 2026. Benchmark measurements are from September-November 2026. Vendor specifications and price anchors establish the stated assumptions; they do not validate this paper's workload throughput on untested hardware.

P1. Cloudflare, Clef-Flash model card
https://huggingface.co/Cloudflare/clef-flash/blob/main/README.md
9B open-weight model, Apache-2.0 licence, typed decision architecture and upstream serving information.

P2. OpenAI, Decisions guide
https://developers.openai.com/api/docs/guides/decisions
Supported model, typed questions, returned values, shared-input caching and native input pricing.

P3. OpenAI, Create a decision reference; API pricing
https://developers.openai.com/api/reference/resources/decisions/methods/create
Choice schema and request controls; no reasoning-effort parameter. Pricing supplement: developers.openai.com/api/docs/pricing.

P4. LM Studio, System One endpoint
https://lmstudio.ai/docs/developer/jev-compat/systemone
Native SystemOne state, questions and answers interface used by the local Clef adapter.

P5. TypeSafe, Choice primitive
https://docs.typesafe.ai/primitives/choice
Choice identifiers, descriptions, full probabilities and separate confidence field.

P6. Lenovo, Generative AI TCO, 2026 edition
https://lenovopress.lenovo.com/lp2368-on-premise-vs-cloud-generative-ai-total-cost-of-ownership-2026-edition
Updated 24 July 2026. Eight-H200 customer-price anchor $397,801.60 as of 15 June; 12% maintenance planning precedent. Vendor workload throughput and five-year lifecycle are not imported.

P7. NVIDIA, DGX H100/H200 User Guide
https://docs.nvidia.com/dgx/dgxh100-user-guide/introduction-to-dgxh100.html
Eight-GPU H200 node, 8U envelope and 10.2 kW maximum. GPU memory and bandwidth specifications are separately sourced in P10.

P8. JustColo, public colocation pricing
https://www.justcolo.net/pricing
Advertised enterprise all-in pricing provides a planning anchor; site describes January 2027 opening and standard terms. The 45 kW case requires a bespoke quote; availability is not established.

P9. US EIA, Electric Power Monthly, Table 5.3
https://www.eia.gov/electricity/monthly/epm_table_grapher.php?pStoreID=epp.%27%5B0%5D&t=epmt_5_03
23 October 2026 release: preliminary August commercial electricity average 14.47 cents/kWh. The model uses a rounded $0.15 US planning allowance, not a client tariff.

P10. NVIDIA, HGX AI Factory components
https://docs.nvidia.com/enterprise-reference-architectures/hgx-ai-factory-h100-h200-b200/latest/components.html
H200 GPU memory capacity and bandwidth specifications. Hardware capability is not a Clef throughput measurement.

Cost basis is USD, undiscounted straight-line allocation over three years with zero residual value. Public vendor advertisements are planning references; procurement prices, energy tariffs and service levels vary by location and contract.