MAHJ-Bench: A Multidimensional Benchmark of Language Model Reasoning Under Uncertainty

Airende Ojeomogha

September 2026

Abstract

A correct action does not imply a correct numerical explanation, and a missing response does not establish a reasoning error. We investigate these distinctions in MAHJ-Bench, an exact-reference evaluation of static decision problems motivated by imperfect-information reasoning. The audited campaign covers 57 tested model configurations in 11 execution groups, using 100 task IDs, 93 unique prompts, and the original 10 capability categories. A completion-adjusted analysis retains 4,813 terminal outcomes from 5,700 planned tested-model slots. Exact action selection succeeds on 62.8 % of planned slots, whereas strict correctness of the entire response contract reaches 44.0 %; 1,072 exact-action successes fail the complete contract. On 4,423 confidence-eligible predictions, mean stated confidence is 95.7 % against 80.9 % action accuracy. Completing missing outcomes also changes point-estimate leadership: the archived Astra configuration attains 100 strict successes versus Sol's 96 on their shared 100 task IDs, but the exact prompt-cluster paired test remains inconclusive ($p = 0.125$). We formalize scoring gates, selection policies, missingness bounds, probability and confidence diagnostics, cluster-aware comparisons, and a narrowly defined cost–quality frontier. Every empirical result derives from the same frozen evidence release. The findings support multidimensional, coverage-aware reporting rather than an unqualified model ranking. Integrated appendices provide the complete tested-model results, mathematical methods, configuration sensitivities, and the boundary of aggregate-level reproducibility.

Keywords: language-model evaluation; imperfect information; numerical reasoning; calibration; missing data; reproducibility.

1 Introduction

The reliability of a language-model answer is not a single event. A system must return a usable response, comply with its output contract, select an appropriate action, and compute the quantities needed to justify that action. It may succeed at one stage and fail at another. A response that names an optimal action but supplies an incorrect posterior or value is distinguishable from a fully correct answer; a request that never produces a retained answer is distinguishable from either.

MAHJ-Bench examines these distinctions with finite, exact-reference numerical tasks. Prompts specify hidden states, observations, opponent processes, policy changes, inspection plans, or counterfactual structures. The model returns a structured action, probability forecast, value, and confidence. Histories and sequential decisions are represented *within a single request*. The evidence therefore concerns static numerical microdecisions, not a full-game American Mahjong tournament, live multi-agent interaction, or online learning.

The campaign is an as-executed study. Endpoint identities, provider routes, output limits, and recovery histories differ across configurations. Subsequent gap completion supplies additional outcomes without changing previously retained grades. This design makes coverage and selection part of the scientific record rather than implementation details that can be discarded after collection.

We make three contributions. First, we report cross-family results from one internally audited, immutable release while preserving original-primary, historical terminal, and completion-adjusted views. Second, we formalize the distinction

between delivery, exact action, complete correctness, format-only extraction, calibration, and observed inference cost. Third, we expose the inferential consequences of missing outcomes and repeated prompts through matched comparisons, prompt-cluster tests, control sensitivities, and an explicit aggregate-data reproduction boundary. The resulting claim is deliberately scoped: the benchmark identifies failures in the executed numerical contract, not a universal ordering of intelligence.

2 Related work

HELM adopts a multidimensional evaluation perspective and emphasizes transparency about evaluation conditions [1]. The Benchmark Lottery demonstrates how task selection can change the apparent relative performance of methods [2]. MAHJ-Bench follows these methodological principles by reporting several endpoints and preserving task-composition sensitivities instead of treating one aggregate as a universal score.

The Hanabi Challenge studies cooperative decision making under imperfect information [3], while OpenSpiel supplies environments and algorithms for research in games [4]. These are conceptual neighbors, not equivalent execution settings. The present campaign consists of numerical prompts rather than interactive game trajectories. Similarly, τ-bench evaluates tool–agent–user interaction and consistency across repeated trials [5]; our fixed-state repeats and within-prompt policy descriptions are narrower.

Confidence calibration concerns the relationship between predicted confidence and correctness [6]. Squared probability scoring originates in forecast verification [7]. We distinguish confidence about an action from a probability vector over task outcomes, since the two answer different questions. Work on uncertainty in reinforcement-learning evaluation motivates reporting distributions and sensitivity rather than point estimates alone [8]. Dataset documentation also motivates stating composition, collection, intended interpretation, and restrictions together [9]. These connections inform the methods; external benchmark scores are not pooled with MAHJ results.

3 Benchmark and experimental design

3.1 Finite decision problems

Let i index a task, x_i its visible specification, Ω_i its finite latent-state space, and $\mathcal{A}_i$ its listed actions or policies. When the task is identified, its reference quantities include a probability vector q_i, a requested scalar $v_i^\star$, and an accepted action $a_i^\star$. For expected-utility selection, the common abstraction is

$$\mathbb{P}_i(\omega \mid x_i) = \frac{\mathbb{P}_i(x_i \mid \omega)\mathbb{P}_i(\omega)}{\sum_{\omega' \in \Omega_i} \mathbb{P}_i(x_i \mid \omega')\mathbb{P}_i(\omega')}, \qquad a_i^\star \in \arg\max_{a \in \mathcal{A}_i} \sum_{\omega \in \Omega_i} \mathbb{P}_i(\omega \mid x_i) u_i(a, \omega). \tag{1}$$

The notation describes the finite decision object; it does not assert that every task uses the same generator or that the model implements Bayesian inference. The requested scalar is the quantity named by that task's contract, not automatically the utility of the model's chosen action. Listed-policy evaluation is likewise not unrestricted policy search.

Intentional no-answer items test nonidentification rather than computational difficulty. Writing $\Theta(x_i)$ for structures consistent with the supplied information, a target g_i is nonidentified when

$$\exists\, \theta, \theta' \in \Theta(x_i) : \quad g_i(\theta) \neq g_i(\theta'). \tag{2}$$

The prescribed answer in such cases is `U`. A refusal, missing response, or invalid output is not converted into `U`. Conversely, the stratum named `oracle` contains answerable tasks and does not license abstention.

3.2 Population, task composition, and inference settings

The roster contains 58 planned entries, of which 57 were submitted. The unsubmitted Nova Premier entry contributes no tested outcomes and is excluded from tested-model comparisons. Every tested configuration has 100 planned task IDs, yielding 5,700 model–task slots. There are 93 distinct prompt strings: one reliability state appears eight times, and all other prompt strings occur once.

The original categories are Hidden-state reasoning; Uncertainty intelligence; Policy adaptation; Opponent modeling; Long-horizon planning; Reliability / consistency; Recovery intelligence; Information acquisition; Counterfactual reasoning; and Strategic plasticity. Their names, identifiers, and assignments remain unchanged. Appendix A gives the operational scope of each category. The designed strata contain ten easy, twenty medium, twenty-five hard, twenty

Table 1. Preserved analysis views on the same tested-model denominator ($N = 5,700$).

	Outcome counts				Strict success (%)	
View	T	S	E	F	Planned	Terminal
Original primary	4,553	2,379	3,394	3,598	41.7	52.3
Preserved terminal baseline	4,685	2,443	3,484	3,704	42.9	52.1
Completion-adjusted	4,813	2,507	3,579	3,803	44.0	52.1

T: terminal; S: strict complete; E: exact action; F: format-only semantic action. Terminal-conditioned accuracy changes its denominator; it does not improve a retained answer.

extra-hard, twenty oracle, and five no-answer task IDs. They are construction labels, not empirically calibrated hardness levels.

Models were evaluated through live API requests in execution groups. Requested identities refer to the archived endpoints, not a current model catalog or independently verified backend weights. The two mixed-developer groups are not single model families. Original requested output ceilings span 8,192–131,072 tokens; explicit JSON mode and seed controls were not uniform. Appendix C.2 reports these differences and the gap-completion sensitivities. Neither a common task suite nor a common requested ceiling establishes equal realized compute.

3.3 Frozen evidence and internal validation

The evidence release is `MAHJ-AsExecuted-v1.1-20260924`. The internal audit replayed 6,213 source score records across historical and supplemental roles, including explicit no-result records. That count is not the number of independent answered tasks. Exact-arithmetic checks covered 95 answerable oracles and five nonidentification witness sets; no score mismatches or source-grade changes were recorded. The audit supports consistency of the preserved implementation, finite reference calculations, and retained evidence. It is not external peer review, human certification, or proof of every possible natural-language interpretation.

4 Outcome selection and scoring

4.1 Three selection views

We distinguish an assigned *slot*, a collection *attempt*, and a *selected outcome*. Several attempts for the same model–task slot do not create additional independent task observations. The original-primary view retains original observations and missing original records. The preserved terminal baseline includes the recovery or replacement outcomes already selected under the historical policy. The completion-adjusted view adds the first qualifying terminal outcomes for predeclared gaps, without replacing baseline selections or choosing for correctness or formatting.

The preserved baseline contains 4,685 terminal selections. Gap completion adds 128 terminal outcomes and 64 strict successes, producing 4,813 terminal selections with 887 unfilled slots. Fifty-three lost original Claude observations remain missing in the original-primary view; replacement responses are not relabeled recovered originals. Table 1 preserves the three estimands explicitly.

4.2 Exact action and complete correctness

For model m and task i, let $t_{mi} \in \{0, 1\}$ indicate a selected terminal outcome. A terminal is a recorded final answer or qualifying native refusal; it need not be valid JSON or correct. The response contract is

$$\widehat{y}_{mi} = (\widehat{a}_{mi}, \widehat{p}_{mi}, \widehat{v}_{mi}, c_{mi}), \qquad \text{keys}(\widehat{y}_{mi}) = \{\texttt{choice}, \texttt{probabilities}, \texttt{value}, \texttt{confidence}\}. \tag{3}$$

Let g^{ex}_{mi} denote accepted exact parsing, transport, and identity gates; h_{mi} denote full-schema validity; P_{mi} denote probability-contract correctness; and V_{mi} denote requested-value correctness. The endpoint indicators are

$$e_{mi} = t_{mi} g^{\text{ex}}_{mi} \mathbb{1}\{\widehat{a}_{mi} = a^{\star}_{i}\}, \qquad s_{mi} = e_{mi} h_{mi} P_{mi} V_{mi}, \qquad 0 \le s_{mi} \le e_{mi} \le t_{mi}. \tag{4}$$

Strict completeness therefore requires more than the right action. Confidence must satisfy the schema, but strict correctness does not require it to equal an externally assigned confidence target. Numerical comparisons use the frozen absolute tolerance $\varepsilon = 10^{-5}$; null-valued contracts require explicit nulls. Appendix B gives the parser and numerical rules.

A separate format-only channel extracts a unique complete answer object under its declared wrapper rules and then applies the same numerical contract. Its action indicator is f_{mi}. It does not repair probabilities or values. Because extraction requires the full four-key object, f_{mi} is *not* a monotonic superset of e_{mi}: an exact correct action in an incomplete object can fail the semantic gate.

4.3 Denominators, decomposition, and missingness

For a fixed model or pooled analysis frame, write N for planned tested slots and (T, E, S, F) for sums of the corresponding indicators. Terminal coverage C, verified planned-slot strict success V, and terminal-conditioned strict accuracy A satisfy

$$C = \frac{T}{N}, \qquad V = \frac{S}{N}, \qquad A = \frac{S}{T}, \qquad V = CA \quad (T > 0). \tag{5}$$

The identity shows why a change in conditional accuracy is not the same as a change in end-to-end success. The observed partition

$$N = \underbrace{(N - T)}_{\text{unfilled}} + \underbrace{(T - E)}_{\text{terminal, no exact-action success}} + \underbrace{(E - S)}_{\text{exact action, incomplete contract}} + \underbrace{S}_{\text{strict complete}} \tag{6}$$

separates missing evidence from failure of a returned answer. The second and third terms still mix output-contract and reasoning effects; they are not pure cognitive-error counts.

For $M = N - T$ unfilled slots, the counterfactual all-slots correctness rate, holding observed grades fixed, lies within

$$\frac{S}{N} \;\le\; A_{\text{all slots}} \;\le\; \frac{S + M}{N}. \tag{7}$$

These are logical identification bounds, not confidence intervals. They allow every unfilled outcome to fail or succeed and make no missing-at-random assumption. They do not alter the observed end-to-end score S/N.

5 Cross-family results

5.1 Action choice and the full answer contract diverge

Across the completion-adjusted frame, exact action success is $3,579/5,700 = 62.8\%$ and strict complete success is $2,507/5,700 = 44.0\%$. The gap is 18.8 percentage points. In counts, Eq. (6) becomes

$$5,700 = 887 + 1,234 + 1,072 + 2,507. \tag{8}$$

Thus 30.0 % of exact-action successes fail the complete response contract. This is not evidence that all such answers contain a mathematical error: missing or extra schema fields can also prevent strict completeness. It is evidence that action-only scoring systematically omits requirements evaluated by the full contract.

Conditioning on the 4,813 terminal outcomes raises exact action accuracy to 74.4 % and strict accuracy to 52.1 %. No answer has improved; the denominator has changed. Overall coverage is 84.4 %, and the logical strict-correctness bounds over unfilled slots are [44.0 %, 59.5 %]. Figure 1 shows how the preserved views alter coverage and endpoints together.

5.2 Group composition and model-level interpretation

Table 2 includes all execution groups. Group totals combine different roster sizes, coverage patterns, and requested controls. They describe the tested mixture, not a randomly sampled developer population. In particular, the mixed groups cannot support general claims about a single developer or geographic model population.

Only 24 of 57 tested models have terminal outcomes for all planned tasks. Table 3 displays the largest observed strict counts while retaining coverage and action endpoints; Appendix D includes every configuration, including low-coverage and low-scoring entries. A configuration can have high accuracy on its retained answers while leaving many planned slots unfilled. Conversely, a fully delivered response can fail the numerical contract repeatedly. Neither should disappear into an answered-only leaderboard.

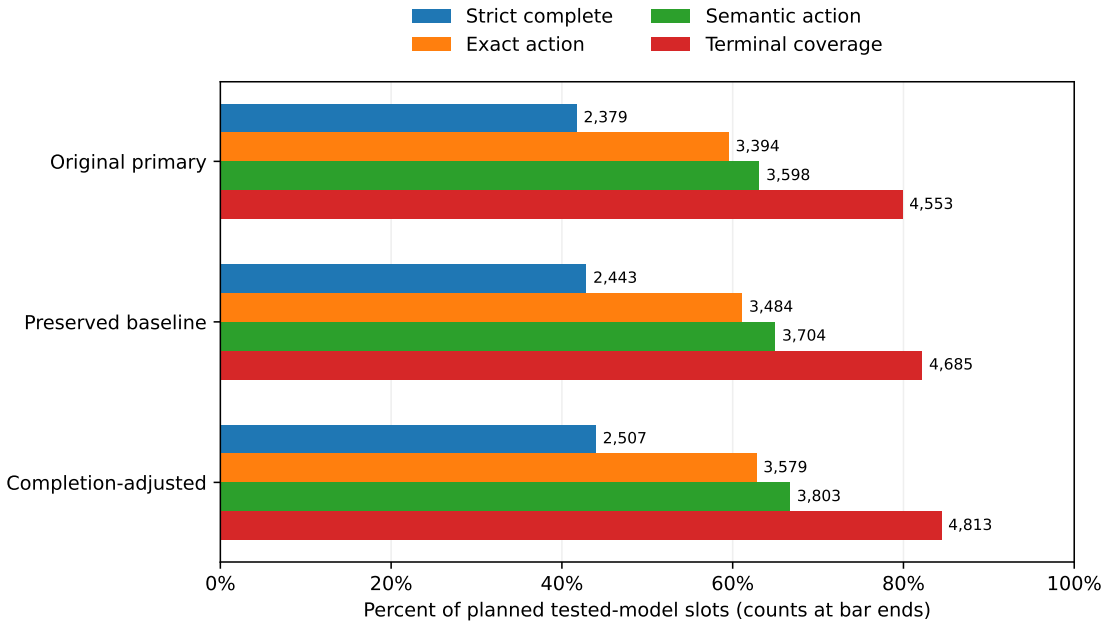

Figure 1. Outcome views on a fixed planned denominator. Bars show strict completeness, exact action, format-only semantic action, and terminal coverage. Counts are annotated. Historical replacements and new terminal additions remain separate selection changes rather than overwritten primary observations.

Table 2. Cross-group coverage and correctness in the completion-adjusted view.

Group	Execution label	Models	N	T	S	E	S/N (%)
G1	OpenAI–GPT	7	700	640	516	615	73.7
G2	Anthropic–Claude	5	500	486	300	341	60.0
G3	Google–Gemini	4	400	399	304	375	76.0
G4	Qwen	4	400	266	22	101	5.5
G5	DeepSeek	3	300	288	224	271	74.7
G6	Meta	3	300	296	148	233	49.3
G7	Mistral	2	200	120	40	99	20.0
G8	NVIDIA Nemotron	3	300	221	91	139	30.3
G9	Z.ai GLM	3	300	239	42	119	14.0
G10	Other developers	15	1,500	1,225	711	971	47.4
G11	Additional American models	8	800	633	109	315	13.6
Total		57	5,700	4,813	2,507	3,579	44.0

Execution labels identify the collection rosters, not randomly sampled developer populations. G10 and G11 pool multiple developers. Nova Premier was unsubmitted and is excluded.

5.3 Original capability slices and format sensitivity

Table 4 and Figure 2 retain the original categories. Reliability / consistency has the lowest planned-slot strict success in this mixture. Policy adaptation, Long-horizon planning, and Strategic plasticity also have low strict counts. These patterns combine numerical requirements, response compliance, missingness, and task construction; they do not establish that a broad cognitive faculty is intrinsically harder than another.

The operational distinction is especially important for ambiguous labels. Recovery intelligence evaluates damaged-hand repair plans, not API retry durability. Information acquisition evaluates listed inspection strategies under supplied likelihoods and fees, not actual external tool calls. Policy and opponent changes occur within prompts, not through persistent learning across interactions.

Format-only extraction rescues 226 exact-action failures and loses 2 exact-action successes because of its complete-object gate, a net difference of 224. It also rescues 110 complete answers. These effects isolate a limited change in interpretation of answer formatting; they do not constitute new inference or corrected numerical reasoning.

For the nonidentification diagnostic, 285 planned no-answer slots yield 273 terminal outcomes and 229 correct exact U

Table 3. Largest observed strict-completeness counts; all configurations have 100 planned slots.

Requested endpoint (short name)	Group	T	S	E	F	S/T (%)
gpt-6-astra	G1	100	100	100	100	100.0
gpt-5.5	G1	100	97	99	99	97.0
gpt-5.6-sol	G1	100	96	99	99	96.0
gemini-3.8-flash	G3	100	93	100	100	93.0
muse-spark-1.3	G6	96	91	95	95	94.8
gemini-3.1-pro-preview	G3	100	90	99	99	90.0
claude-opus-5	G2	99	87	95	95	87.9
deepseek-v4-pro-0813	G5	95	87	93	93	91.6
deepseek-v4.1-flash	G5	93	86	92	92	92.5
gemini-3.7-flash	G3	100	84	100	100	84.0
grok-4.5	G10	90	84	88	88	93.3
grok-4.6	G10	90	82	88	88	91.1
claude-fable-5.1	G2	100	81	91	97	81.0

The display is descriptive, not an uncertainty-adjusted ranking. Terminal coverage, retained controls and the shared task sample govern comparisons. Appendix D lists every tested endpoint.

Table 4. Original capability slices ($N = 570$ planned model–task slots per capability).

Original capability	T	S	E	S/N (%)	E/N (%)
Hidden-state reasoning	482	240	341	42.1	59.8
Uncertainty intelligence	480	289	380	50.7	66.7
Policy adaptation	529	183	356	32.1	62.5
Opponent modeling	494	240	396	42.1	69.5
Long-horizon planning	413	184	268	32.3	47.0
Reliability / consistency	421	167	353	29.3	61.9
Recovery intelligence	536	349	438	61.2	76.8
Information acquisition	503	326	379	57.2	66.5
Counterfactual reasoning	548	344	417	60.4	73.2
Strategic plasticity	407	185	251	32.5	44.0

Each category has ten task IDs. The reliability category contains only three unique prompts, one of which occurs eight times. Scopes and identifiers are specified in Appendix A.

answers. There are 28 false `U` answers among 5,415 planned answerable slots. This result concerns mathematically specified nonidentification, not safe abstention or truthfulness in open-ended deployment.

6 Probabilistic correctness and confidence

6.1 Probability forecasts and intrinsic uncertainty

A model's forecast vector and its confidence in an action are distinct objects. For an eligible forecast $\widehat{p}_i$ and oracle vector q_i, we report excess Brier and expected Brier:

$$B_i^{\text{excess}} = \sum_j (\widehat{p}_{ij} - q_{ij})^2, \qquad B_i^{\text{expected}} = B_i^{\text{excess}} + \sum_j q_{ij}(1 - q_{ij}). \tag{9}$$

To see the distinction, let Z_{ij} indicate whether forecast event j occurs, so $\mathbb{E}[Z_{ij} \mid x_i] = q_{ij}$. Expanding the squared error gives

$$\mathbb{E}\big[\|\widehat{p}_i - Z_i\|_2^2 \mid x_i\big] = \|\widehat{p}_i - q_i\|_2^2 + \sum_j \operatorname{Var}(Z_{ij} \mid x_i) = B_i^{\text{expected}}. \tag{10}$$

An exact forecast has zero excess Brier but generally positive expected Brier because the outcome itself remains uncertain. This follows the squared forecast-scoring principle [7]; it is not a claim to have observed future outcomes. Where the forecast events form a categorical partition, Z_i is one-hot. The formula also applies componentwise to declared indicator events.

There are 85 task IDs requesting numerical probability vectors. Contracts requesting null probabilities are not missing forecasts. On 3,610 eligible completion-view forecasts, pooled excess Brier is 0.04140 and expected Brier is 0.44554. Unavailable or invalid vectors are excluded from these denominators rather than replaced with invented probabilities.

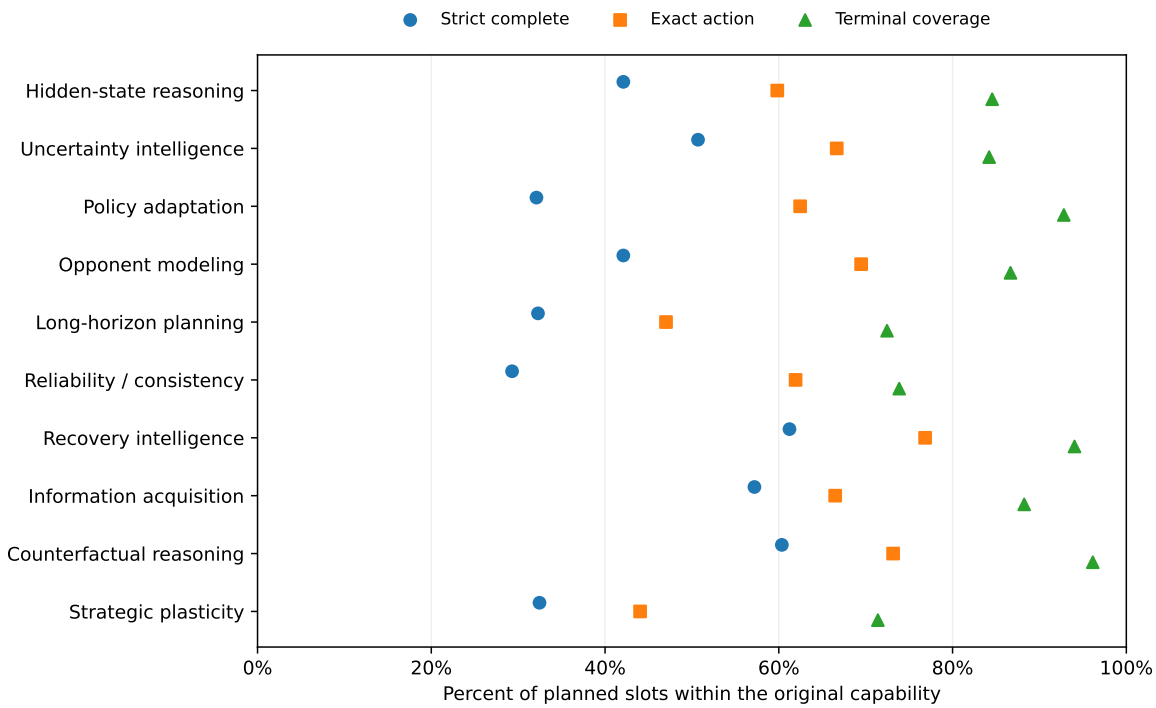

Figure 2. Capability profiles under the original ten-category registry. Each category contributes the same 570 planned model–task slots. Strict success, exact action, and terminal coverage remain separate endpoints. The reliability category includes a repeated state and should not be read as ten independent problems.

Table 5. Pooled confidence and probability-forecast diagnostics.

Diagnostic	Eligible rows	Estimate
Mean stated confidence	4,423	0.9570
Action accuracy on confidence-eligible rows	4,423	0.8090
Confidence Brier	4,423	0.17230
Fixed-bin ECE (five bins)	4,423	0.14991
Fixed-bin ECE (ten bins)	4,423	0.15049
Forecast excess Brier	3,610	0.04140
Forecast expected Brier	3,610	0.44554

Confidence predicts action correctness; the forecast vector predicts task outcomes. Brier quantities use the exact unnormalized squared-error sums specified in Eqs. (9)–(11). Eligibility is metric-specific.

6.2 Confidence calibration and selective risk

Let $\mathcal{I}_c$ be rows with an eligible confidence c_i and exact action outcome $y_i = e_i$. The binary confidence Brier and fixed-bin expected calibration error are

$$B^{\text{conf}} = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} (c_i - y_i)^2, \tag{11}$$

$$\text{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{|\mathcal{I}_c|} \left| \frac{\sum_{i \in \mathcal{B}_b} c_i}{|\mathcal{B}_b|} - \frac{\sum_{i \in \mathcal{B}_b} y_i}{|\mathcal{B}_b|} \right|. \tag{12}$$

Empty bins contribute zero. Confidence Brier combines aspects of calibration and discrimination; ECE depends on the bin partition [6]. Neither is interchangeable with the vector forecast error in Eq. (9).

On 4,423 eligible predictions, mean confidence is 95.7 % and action accuracy is 80.9 %, a signed gap of 14.8 percentage points. Confidence Brier is 0.17230. Among 845 wrong actions with usable confidence, 697 have confidence at least 0.90 and 222 have confidence exactly one. Mean confidence on wrong actions is 92.7 %. The five-bin and ten-bin ECE estimates are 0.14991 and 0.15049, respectively (Table 5).

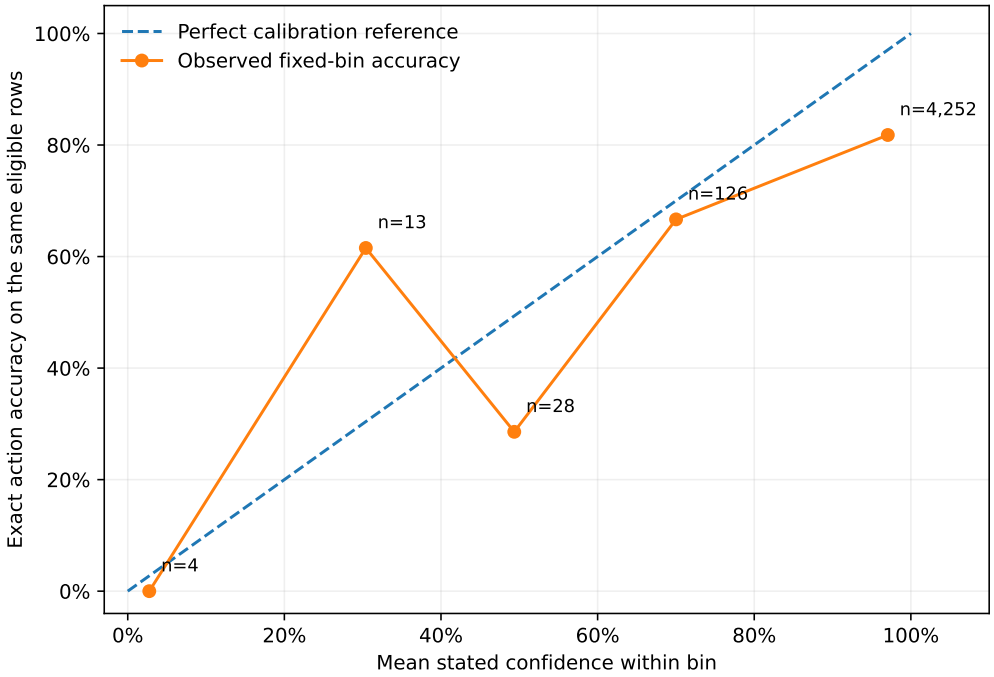

Figure 3. Fixed-bin confidence calibration. Each point compares mean stated confidence with exact action accuracy on the same eligible rows; bin counts are annotated. Sparse low-confidence bins are descriptive. The pooled curve does not establish deployment calibration or a uniform pattern for every model.

For threshold τ, retain the entire tie set $\mathcal{I}_\tau = \{i \in \mathcal{I}_c : c_i \geq \tau\}$. Selective action risk and the two relevant coverage measures are

$$R(\tau) = \frac{\sum_{i \in \mathcal{I}_\tau}(1 - y_i)}{|\mathcal{I}_\tau|}, \qquad C_c(\tau) = \frac{|\mathcal{I}_\tau|}{|\mathcal{I}_c|}, \qquad C_N(\tau) = \frac{|\mathcal{I}_\tau|}{N}. \tag{13}$$

At $\tau = 0.99$, 2,573 predictions remain, but 14.6 % still choose an incorrect action. The full threshold table appears in Appendix B. High reported confidence alone does not identify an error-free retained subset. These are pooled descriptive statistics, not calibrated guarantees for an external task population.

7 Paired inference and completion sensitivity

7.1 Matched comparisons and repeated prompts

For models a and b, let $\mathcal{I}_{ab} = \{i : t_{ai}t_{bi} = 1\}$ be mutually terminal task IDs. With endpoint $z \in \{s, e\}$, the paired difference is

$$\widehat{\Delta}_{ab} = \frac{1}{n_{ab}} \sum_{i \in \mathcal{I}_{ab}} (z_{ai} - z_{bi}), \qquad n_{ab} = |\mathcal{I}_{ab}|. \tag{14}$$

The matched sample must be shown: mutual terminal conditioning changes the task mixture for incomplete configurations. Only 2 task IDs are terminal for every tested model, so a universal complete-case comparison set discards nearly the entire suite.

Repeated calls on an identical prompt are grouped for uncertainty calculations. The analysis uses 10,000 deterministic prompt-cluster bootstrap resamples and an exact model-label swap test that moves every occurrence of the same prompt together. Across 1,596 model pairs and two endpoints, Holm correction [10] is applied to 3,192 tests within each declared view. The same-profile subset contains 164 pairs, but common requested controls do not prove equal effective compute. Appendix C specifies the bootstrap ratio, exact test, and multiplicity calculation.

7.2 A completion effect is not established superiority

Astra's original strict count is 78. Its 22 additional terminal outcomes are all strict successes and retain its own requested controls, producing 100/100. GLM-5.3, by contrast, adds 17 terminal outcomes without adding a strict success. Gap completion supplies evidence rather than mechanically improving conditional accuracy.

Table 6. Astra-minus-comparator paired strict-completeness results.

Comparator	n	K	b/c	Δ (pp)	Cluster interval (pp)	p	p_{Holm}
gpt-5.5	100	93	3/0	3.0	[-0.0, 7.0]	0.2500	1.00
gpt-5.6-sol	100	93	4/0	4.0	[0.9, 8.6]	0.1250	1.00
gemini-3.8-flash	100	93	7/0	7.0	[1.1, 13.2]	0.0625	1.00
muse-spark-1.3	96	89	5/0	5.2	[1.1, 10.3]	0.0625	1.00
gemini-3.1-pro-preview	100	93	10/0	10.0	[3.2, 17.8]	0.0156	1.00

n: mutually terminal task IDs; K: matched unique prompts; b/c: Astra-only / comparator-only strict successes. Intervals are empirical 95% prompt-cluster bootstrap intervals. Tests use exact prompt-cluster label swaps; Holm correction spans 3,192 tests per view. These selectively displayed comparisons are exploratory.

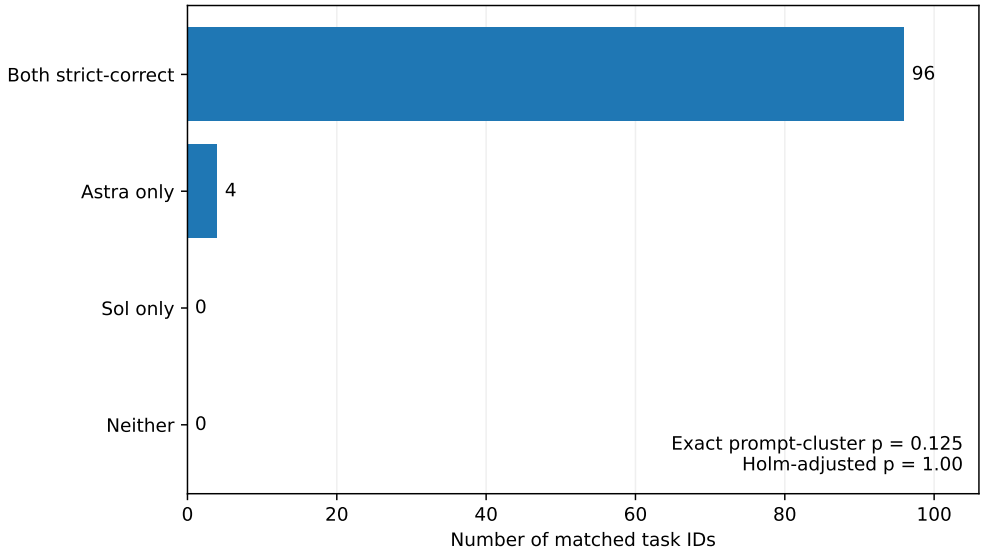

Figure 4. Astra–Sol paired contingency counts. Both configurations are terminal on the full suite. Four strict discordances favor Astra, but the exact prompt-cluster test is inconclusive. The figure describes an as-executed comparison, not an equal-compute or causal model-quality experiment.

Astra and Sol have 100 matched task IDs and 93 unique prompts. Both are strict-correct on 96 tasks; four favor Astra alone and none favor Sol alone. The observed difference is four percentage points, with exact prompt-cluster $p = 0.125$ and Holm-adjusted $p = 1.00$. Table 6 shows related top-count comparisons without suppressing sample sizes or adjusted evidence.

The empirical task-weighted cluster-bootstrap interval for Astra minus Sol is 0.9–8.6 percentage points. Its positive lower endpoint is not independent confirmation of superiority: an empirical percentile interval near a ceiling and an exact finite-discordance test need not support the same binary impression. The bootstrap represents resampling of this finite task composition, while the exact test assesses label-exchangeability evidence. Both are reported; the conclusion is not upgraded on the basis of the more favorable summary.

7.3 Declared-control sensitivities

Of the 128 new terminal outcomes, 94 retain the saved requested controls and 34 use higher output ceilings. Observable identity and reported token-bound checks retain 92 outcomes in the unchanged-control sensitivity and 126 across both declared strata. The two cap-exceeding outcomes remain in the full completion view and are excluded only in the corresponding control sensitivity. No outcome is excluded for its grade. These flags apply to the new selections; they do not certify every historical runtime setting. The numerical sensitivity table and original-control summary appear in Appendix C.2.

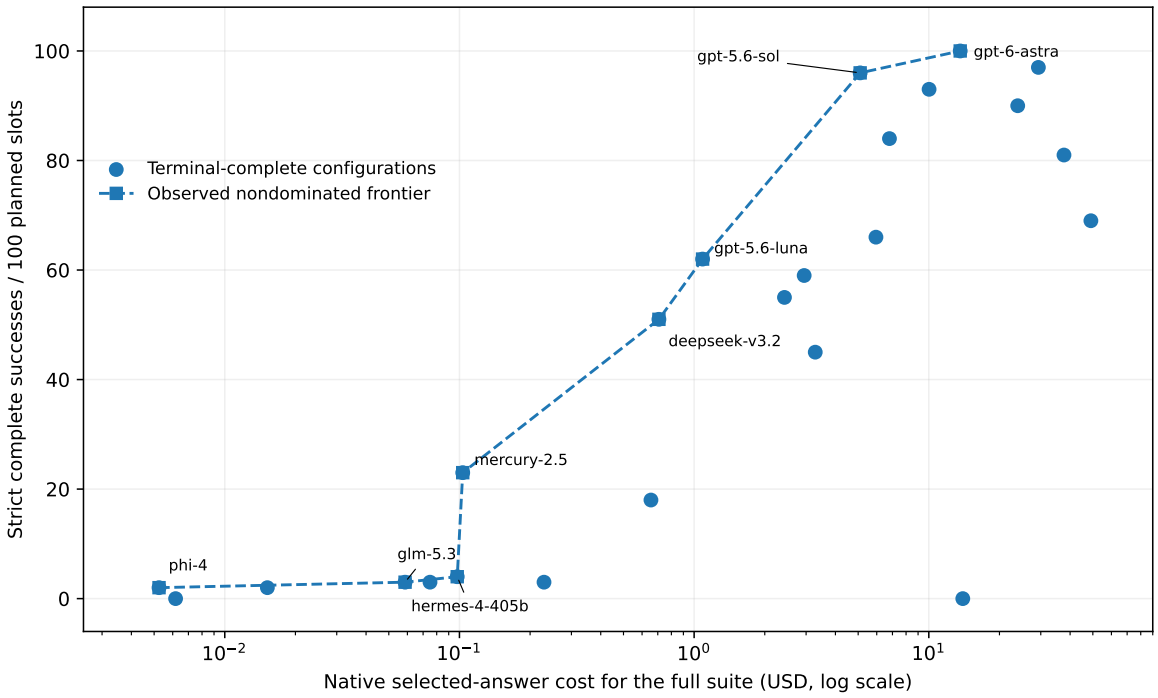

Figure 5. Selected-answer cost versus strict success among terminal-complete configurations. The horizontal axis is logarithmic. The connected frontier consists of observed nondominated points, not a fitted scaling law. Costs are historical native charges under heterogeneous recorded controls; unselected acquisition attempts are excluded.

8 Observed cost–quality trade-offs

We use a deliberately narrow accounting boundary. For selected terminal records with known native charges k_{mi}, define

$$K_m^{\mathrm{sel}} = \sum_{i:t_{mi}=1} k_{mi}, \qquad \kappa_m^{\mathrm{answer}} = \frac{K_m^{\mathrm{sel}}}{T_m}, \qquad \kappa_m^{\mathrm{strict}} = \frac{K_m^{\mathrm{sel}}}{S_m} \quad (S_m > 0). \tag{15}$$

These quantities exclude unselected failed acquisition attempts. Unknown charges are not free inference, and a maximum reservation is not confirmed spend. We do not interpret selected-answer cost as the total cost of executing or operating a benchmark service.

Within the 24 terminal-complete configurations with known selected-answer cost, m is nondominated when no other eligible m' satisfies

$$K_{m'}^{\mathrm{sel}} \le K_m^{\mathrm{sel}}, \qquad S_{m'} \ge S_m, \qquad \text{with at least one strict inequality.} \tag{16}$$

Figure 5 contains 8 such points. A low-cost, low-scoring configuration may be nondominated without being useful for an application. Frontier membership is a property of this cohort and accounting scope, not a purchase recommendation or universal efficiency ranking.

Astra's selected-answer cost is \$13.6095, compared with \$5.0986 for Sol, a ratio of 2.67. Their score difference must be interpreted jointly with the paired evidence and differing provider profiles. Realized completion tokens, requested token ceilings, and observed cost are distinct quantities; none identifies a randomized treatment of reasoning effort. The frontier table and accounting definitions are retained in Appendix F without account-level balances or budget information.

9 Limitations and reproducibility

Construct and population scope. This is a small synthetic suite of structured numerical problems with exact references. Capability names are source-preserved labels, not validated broad cognitive scales. A perfect finite-suite result does not establish perfect generalization. Full-game Mahjong strength, online learning, live multi-agent competence, real-world tool execution, safety, and deployment reliability are outside the observed design.

Selection and controls. Missingness is substantial and need not be random. Timeouts, serving behavior, output limits, and recovery policy affect which answers are retained. Lost original records cannot be restored by relabeling replacements. Different model routes and controls prevent causal attribution to architecture, model size, reasoning budget, or compute. Current endpoint availability and stable backend weights are not established by the archived names.

Dependence and retrospective inference. The paired analysis clusters identical prompts, but shared task grammars may produce residual dependence. Rank and weighting scenarios are deterministic sensitivities rather than confidence intervals. The analysis is retrospective, and the displayed pairwise comparisons are exploratory. Statistical nonsignificance is not equivalence. Unknown training exposure and contamination by related tasks are not resolved by cryptographic source custody.

Data and code boundary. The accompanying aggregate data contain the reported model, group, capability, difficulty, calibration, paired-comparison, and cost summaries, together with registry metadata and offline publication-reproduction code. Benchmark task text, per-item outcome matrices, oracle instances, production seeds, native responses, reasoning traces, credentials, and account records are not distributed. Public users can reconstruct aggregate tables and figures, but cannot independently replay raw scoring or rerun arbitrary task-level resampling from aggregate data alone. The complete methods and reproduction boundary are part of this paper, not a separate unpublished supplement.

Validation scope. The internal audit checks retained evidence and mathematical consistency under the recorded protocol. Publication checks reproduced core and calibration artifacts and verified the released aggregate lineage. Full financial and bootstrap workstreams were not newly rerun for this typeset manuscript; their estimates remain bound to the previously verified analysis release. AI tools assisted analysis, manuscript drafting, and typesetting. The results are derived from archived evidence rather than generated illustrative measurements.

10 Conclusion

MAHJ-Bench shows why response delivery, action choice, numerical completeness, formatting, and confidence must be reported separately. On a common planned denominator, action success exceeds full-contract success by 18.8 percentage points. High stated confidence coexists with action errors, and additional coverage can change a leading point estimate without establishing model superiority. Explicit selection views, finite-task mathematical contracts, prompt-aware paired inference, and declared accounting boundaries make these distinctions inspectable. The resulting research asset is a scoped cross-family diagnostic of the executed numerical tasks, with the full tested roster and supplementary methods integrated below.

References

[1] Percy Liang, Rishi Bommasani, Tony Lee, et al. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research*, 2023. arXiv:2211.09110.

[2] Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The Benchmark Lottery. 2021. arXiv:2107.07002.

[3] Nolan Bard, Jakob N. Foerster, Sarath Chandar, et al. The Hanabi Challenge: A New Frontier for AI Research. *Artificial Intelligence*, 280:103216, 2020. doi:10.1016/j.artint.2019.103216.

[4] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, et al. OpenSpiel: A Framework for Reinforcement Learning in Games. 2019. arXiv:1908.09453.

[5] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. 2024. arXiv:2406.12045.

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of ICML*, PMLR 70, pp. 1321–1330, 2017. PMLR 70:guo17a.

[7] Glenn W. Brier. Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review*, 78(1):1–3, 1950.

[8] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep Reinforcement Learning at the Edge of the Statistical Precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

[9] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for Datasets. *Communications of the ACM*, 64(12):86–92, 2021. arXiv:1803.09010.

[10] Sture Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[11] MAHJ Research. *MAHJ-AsExecuted-v1.1-20260924: Frozen Evidence and Aggregate Analysis*. 2026. Release identifiers and reproduction scope are specified in Appendix G.

Supplementary Methods and Results

Integrated appendices to MAHJ-Bench

A Original capability registry and task semantics

The names, identifiers, abbreviations, and task assignments follow the executed registry. The descriptions below are operational summaries of the audited probes, not replacement conceptual definitions. No category is remapped to a broader post hoc taxonomy. Each category contains ten task IDs; Reliability / consistency has three unique prompts and the other categories have ten each. Public summaries omit production prompt text and task-to-category mappings.

A.1 Hidden-state reasoning

Source identifier: `hidden_state`; *abbreviation:* HRI. Posterior inference over a finite hidden wall and shared noisy reporter; choose an expected-utility-optimal line. Hidden-state inference under a fully specified synthetic model; no full-game hidden-tile inference or general-domain transfer claim.

A.2 Uncertainty intelligence

Source identifier: `uncertainty`; *abbreviation:* UII. Posterior or joint probability forecasts and expected utility, including correlated draw/threat lotteries and a missing-dependence witness. Exact uncertainty calculation under supplied probabilities; not open-world uncertainty identification or generic truthfulness.

A.3 Policy adaptation

Source identifier: `policy_adaptation`; *abbreviation:* PAI. Reconstruct a listed multi-round commitment under announced rule/payoff/joker changes and switching costs. Scripted within-call reconstruction; no observed online policy updates, training, adaptation speed or live environment interaction.

A.4 Opponent modeling

Source identifier: `opponent_model`; *abbreviation:* OMI. Filter a specified latent opponent-style process and persistent transition regime, then predict its next action. Inference from a supplied history/model; no live multi-agent coordination, competitive tournament or learned opponent model.

A.5 Long-horizon planning

Source identifier: `long_horizon_planning`; *abbreviation:* LPI. Evaluate four supplied contingent stopping policies over a finite wall and choose the best listed policy. Finite policy evaluation over 4–8 draws; not unrestricted policy search or long-duration agent execution.

A.6 Reliability / consistency

Source identifier: `reliability`; *abbreviation:* Reliability. Solve three finite stopping-game states; make eight distinct calls on the same challenge state under the recorded settings. One challenge state supports within-state consistency; eight calls are not eight independent problems or a population reliability estimate.

A.7 Recovery intelligence

Source identifier: `recovery`; *abbreviation:* RIS. Evaluate four legal damaged-hand repair plans with a dead old joker and sunk costs; optimize future expected utility. This capability is damaged-hand decision reconstruction. Supplemental API retries, restart durability and transport recovery are separate operational evidence.

A.8 Information acquisition

Source identifier: `information_acquisition`; *abbreviation:* IAI. Choose among supplied inspection plans using sensor likelihoods, query fees and adaptive stopping, then select a terminal line. Numerical value of information under a stated model; no actual external tool use, API execution or learned information-gathering behavior.

A.9 Counterfactual reasoning

Source identifier: `counterfactual`; *abbreviation:* CRI. Compute structural counterfactuals with shared exogenous noise and factual evidence; identify an observationally nonidentified intervention. Reasoning within supplied causal structures; not real-world causal discovery or an external oracle service.

A.10 Strategic plasticity

Source identifier: `strategic_plasticity`; *abbreviation:* SPI. Compare four supplied cue/memory policies through scripted opponent/incentive shifts over 4–9 microhands, with switching costs. Listed-policy reconstruction in one call; no unrestricted control optimum, online learning or measured adaptation latency.

A.11 Policy values and counterfactual targets

Several task families compare a finite set of supplied policies. For a listed policy π and task horizon H_i, a compatible abstraction is

$$V_i(\pi \mid x_i) = \sum_{\omega \in \Omega_i} \mathbb{P}_i(\omega \mid x_i) \left[\sum_{t=1}^{H_i} r_{it}(\pi, h_t(\omega)) - \mathrm{cost}_i(\pi, \omega) \right], \qquad \pi_i^\star \in \arg\max_{\pi \in \Pi_i} V_i(\pi \mid x_i). \tag{A.1}$$

Here Π_i is the supplied policy set, and h_t is the within-task history. The equation formalizes a finite expectation; it does not expand the tested action space to all possible strategies. Query fees, switching costs, and stopping behavior enter only where specified by the task.

For structural counterfactual probes, factual evidence updates the shared exogenous state before an intervention changes the structural evaluation. With exogenous variable U, evidence e, and the intervened outcome function $Y_a(U)$,

$$\mathbb{P}(Y_a = y \mid e) = \sum_u \mathbb{1}\{Y_a(u) = y\} \mathbb{P}(u \mid e). \tag{A.2}$$

This differs from drawing fresh independent exogenous noise for the intervention. It summarizes the supplied-model counterfactual calculation; it does not constitute real-world causal discovery. Where the evidence admits incompatible target values, the no-answer condition in Eq. (2) applies.

B Exact scoring, eligibility, and auxiliary metrics

B.1 Parsing and schema contract

The exact parser requires the entire final text to decode as one JSON object. It does not strip code fences, search prefixes, coerce numeric strings, accept duplicate keys, or accept nonfinite JSON constants. The action must be a permitted string under the source contract. Recorded transport and identity gates remain part of scoring; unknown identity metadata is not treated as an observed contradiction.

Exact action correctness can be established before full-schema validity. Thus a correct permitted action in exact JSON can coexist with missing or extra keys and fail strict completeness. The strict schema requires exactly `choice`, `probabilities`, `value`, and `confidence`. Confidence must be finite and in its permitted interval; it is not required to equal the empirical frequency of correctness.

For an oracle probability target q_i, probability-contract correctness is summarized as

$$P_i = \begin{cases} \mathbb{1}\{\text{key present and } \widehat{p}_i = \mathsf{null}\}, & q_i = \mathsf{null}, \\ \mathbb{1}\{\widehat{p}_i \text{ valid}, \ \|\widehat{p}_i - q_i\|_\infty \le \varepsilon\}, & q_i \text{ numeric}. \end{cases} \tag{B.1}$$

Validity includes the requested length, finite entries, and the allowed probability range. When the task declares a distribution of length d_i, normalization is checked with the frozen length-scaled tolerance, $|\sum_j \widehat{p}_{ij} - 1| \le d_i \varepsilon$. This normalization requirement is not imposed indiscriminately on vectors of marginal event probabilities.

The requested-value check is

$$V_i = \begin{cases} \mathbb{1}\{\text{key present and } \widehat{v}_i = \mathsf{null}\}, & v_i^\star = \mathsf{null}, \\ \mathbb{1}\{\widehat{v}_i \text{ finite}, \ |\widehat{v}_i - v_i^\star| \le \varepsilon\}, & v_i^\star \text{ numeric}. \end{cases} \tag{B.2}$$

No judge-model inference or answer completion is used to repair these values. The published grades are the preserved source grades.

B.2 Format-only semantics and answerability

The semantic extractor accepts one unique complete answer object with exactly the required keys, either as the entire valid JSON text or as the sole complete object in an allowed wrapper. A valid top-level JSON value with an incorrect type or schema is not searched internally for an alternative answer. Multiple complete objects are ambiguous. Fields, duplicate keys, and numerical values are not repaired.

Consequently, gains and losses between the two action endpoints must be counted separately:

$$F - E = \sum_i \mathbb{1}\{f_i = 1, e_i = 0\} - \sum_i \mathbb{1}\{f_i = 0, e_i = 1\}. \tag{B.3}$$

For this release the terms are 226 and 2, respectively. Strict-to-semantic complete rescues are another quantity and should not be inferred from the net action difference.

The defined answer `U` applies only to intentional nonidentification. A null-valued response contract, a refusal, a malformed object, an unsubmitted slot, and an unavailable response are different states. None is substituted for another to increase coverage or accuracy.

B.3 Value error, regret, and logarithmic loss

Where a substantive chosen action has a defined utility, regret and normalized regret are

$$r_i = u_i(a_i^\star) - u_i(\widehat{a}_i), \qquad \widetilde{r}_i = \frac{r_i}{d_i^{\text{utility}}}, \qquad d_i^{\text{utility}} > 0, \tag{B.4}$$

with the declared positive scale or best-minus-worst utility range. Invalid actions, unfilled outcomes, and false `U` answers are not assigned invented utilities. Different task grammars have different utility scales, so raw pooled regret remains a diagnostic, not a universal capability measure.

Requested-value error uses the task's specified target:

$$\text{MAE} = \frac{1}{|\mathcal{I}_v|} \sum_{i \in \mathcal{I}_v} |\widehat{v}_i - v_i^\star|, \qquad \text{RMSE} = \sqrt{\frac{1}{|\mathcal{I}_v|} \sum_{i \in \mathcal{I}_v} (\widehat{v}_i - v_i^\star)^2}. \tag{B.5}$$

The target need not be the chosen action's return. Numeric values and eligibility must therefore be interpreted together. Raw-value errors are not silently winsorized or removed to improve the reported record.

Binary confidence logarithmic loss is $-y_i \log c_i - (1 - y_i) \log(1 - c_i)$ with the usual endpoint convention. A wrong prediction at confidence one gives infinite unclipped loss. A separately labeled clipped sensitivity cannot erase that endpoint failure. The public calibration summaries retain eligible counts, endpoint failures, and fixed-bin boundaries.

B.4 Threshold table and denominator checks

The thresholds in Table B.1 retain tied confidences together. An absent eligible denominator is undefined rather than zero. Confidence eligibility, valid probability-vector eligibility, and valid value-error eligibility are distinct sets; their metrics cannot share a denominator merely because they come from the same selected responses.

C Selection algorithm, controls, and paired inference

C.1 Preserving selections without best-answer search

Algorithm C.1 summarizes the frozen selection logic used to interpret the outcome views. The first qualifying terminal gap selection is determined before a control-sensitivity predicate is applied. A failed predicate makes the selection unavailable in that sensitivity; it does not permit a search for a later, better-scoring answer. Original-primary observations remain a separate view.

Table B.1. Confidence-threshold selective risk in the exact scoring channel.

τ	Retained	Eligible coverage (%)	Planned coverage (%)	Action error (%)	Strict error (%)
0.00	4,423	100.0	77.6	19.1	43.3
0.50	4,398	99.4	77.2	18.9	43.0
0.70	4,339	98.1	76.1	18.5	42.4
0.80	4,252	96.1	74.6	18.2	41.6
0.90	4,009	90.6	70.3	17.4	40.4
0.95	3,583	81.0	62.9	16.8	39.6
0.99	2,573	58.2	45.1	14.6	35.0

All confidence ties are retained at a threshold. Eligible coverage uses 4,423 predictions; planned coverage uses 5,700 slots. Strict error concerns the full answer contract, not only the chosen action.

Algorithm C.1 Terminal view construction without correctness-based selection

Require: Planned slots $\mathcal{D}$; preserved baseline B; ordered gap evidence G; view predicate v

for each model–task slot $(m, i) \in \mathcal{D}$ **do**
  if B_{mi} is terminal **then**
    $r \leftarrow B_{mi}$; eligible $\leftarrow$ true
  else
    $r \leftarrow$ first qualifying terminal record selected from G_{mi}
    eligible $\leftarrow (r \neq \varnothing) \wedge v(r)$
  end if
  if eligible **then**
    Emit the unchanged stored score and selection role of r
  else
    Emit an unfilled slot with its retained status distinctions
  end if
end for
Do not replace a baseline outcome, optimize over grades, or relabel replacements as originals

Table C.1. Requested-control heterogeneity by execution group.

Group	Models	Profiles	Output ceiling range	JSON mode	Seed set	Amended T
G1	7	3	65,536	7/7	7/7	1
G2	5	3	64,000–65,536	5/5	0/5	2
G3	4	1	65,536	4/4	4/4	0
G4	4	3	65,536	4/4	4/4	0
G5	3	2	65,536	3/3	3/3	3
G6	3	3	16,384–65,536	3/3	1/3	1
G7	2	1	65,536	2/2	2/2	5
G8	3	3	65,536	2/3	2/3	3
G9	3	2	32,768–65,536	3/3	3/3	0
G10	15	8	8,192–65,536	11/15	7/15	16
G11	8	8	8,192–131,072	4/8	3/8	3

JSON and seed columns count models with an explicit request, not confirmed effective runtime behavior. Profile counts are within-group distinct requested-control profiles. Ranges span the original requested maxima, not realized reasoning tokens.

C.2 Requested settings and control sensitivities

Model identity, provider route, requested reasoning controls, output-limit fields, JSON mode, temperature, and seed omission are retained as configuration metadata. A missing provider or returned identity is unknown, not proof of drift or proof of identity. Endpoint names and pinned routes cannot independently verify backend weight stability across collection dates.

Table C.1 summarizes original requested controls without exposing request bodies or production seeds. Its profile counts are within-group; some profiles can recur in multiple groups. Table C.2 applies the new-outcome control flags while keeping the entire historical baseline fixed. Both cap-exceeding new records remain in the full dataset. Configuration strata restrict interpretation but do not convert this retrospective campaign into a randomized equal-compute comparison.

Table C.2. Gap-completion sensitivity under preserved versus amended requested controls.

Selection sensitivity	Added T	Total T	S	E	S/N (%)
Preserved baseline	0	4,685	2,443	3,484	42.9
Unchanged requested additions	94	4,779	2,494	3,553	43.8
Unchanged + observable checks	92	4,777	2,494	3,552	43.8
All declared + observable checks	126	4,811	2,507	3,578	44.0
Full completion-adjusted	128	4,813	2,507	3,579	44.0

New-outcome flags do not certify historical runtime controls. The full view retains both observed cap-exceeding records; the corresponding observable-control sensitivity excludes them, without correctness-based selection.

C.3 Prompt-cluster bootstrap

Let $g = 1, \ldots, G$ index the $G = 93$ unique prompts in the full suite, including prompts with no mutually terminal observations for a given pair. Define the matched cluster denominator and difference

$$n_g = \sum_{i \in g} \mathbb{1}\{i \in \mathcal{I}_{ab}\}, \qquad D_g = \sum_{i \in g \cap \mathcal{I}_{ab}} (z_{ai} - z_{bi}). \tag{C.1}$$

For bootstrap replicate r, the released analysis draws $W^{(r)} \sim \text{Multinomial}(G; 1/G, \ldots, 1/G)$ and evaluates

$$\widehat{\Delta}_{ab}^{(r)} = \frac{\sum_g W_g^{(r)} D_g}{\sum_g W_g^{(r)} n_g}. \tag{C.2}$$

Zero-denominator replicates are excluded. Intervals use the 2.5th and 97.5th percentiles of 10,000 deterministic resamples. Paired outcomes and all eight copies of the repeated state move together within a prompt cluster.

For equal-unique-prompt weighting, the effect is instead

$$\widehat{\Delta}_{ab}^{\text{prompt}} = \frac{1}{\sum_g \mathbb{1}\{n_g > 0\}} \sum_{g: n_g > 0} \frac{D_g}{n_g}. \tag{C.3}$$

The bootstrap resamples cluster means and their nonempty-cluster denominator. Neither weighting defines a deployment population or removes dependence between related task grammars.

C.4 Exact label-swap test and multiplicity

Let $\mathcal{G}_+ = \{g : D_g \neq 0\}$ and $K_+ = |\mathcal{G}_+|$. Under the conditional prompt-level label-exchangeability assumption, the exact two-sided probability is

$$p_{ab} = 2^{-K_+} \sum_{\sigma \in \{-1,+1\}^{K_+}} \mathbb{1}\left\{ \left| \sum_{g \in \mathcal{G}_+} \sigma_g |D_g| \right| \geq \left| \sum_g D_g \right| \right\}. \tag{C.4}$$

The integer sign-sum distribution is evaluated exactly, without a Monte Carlo probability floor; all-zero differences give $p = 1$. Task-ID McNemar results remain secondary dependence-insensitive diagnostics.

For sorted $p_{(1)} \leq \cdots \leq p_{(J)}$ across the $J = 3,192$ tests within a view, Holm-adjusted probabilities are

$$\widetilde{p}_{(k)} = \min\left(1, \max_{j \leq k} (J - j + 1) p_{(j)}\right). \tag{C.5}$$

Holm's method controls family-wise error under valid constituent tests [10]; it cannot remedy a violated exchangeability assumption. Separate selection views are sensitivities, not independent replications.

C.5 Task weighting and rank sensitivity

Sensitivity scenarios weight task IDs, unique prompts, capabilities, or grammars equally, omit the repeated state, or omit one capability. With nonnegative weights w_i, verified and terminal-conditioned scores are

$$V_m(w) = \frac{\sum_i w_i s_{mi}}{\sum_i w_i}, \qquad A_m(w) = \frac{\sum_i w_i s_{mi}}{\sum_i w_i t_{mi}}. \tag{C.6}$$

Equal-capability and equal-task weights coincide in the balanced design. Unique-prompt weights are inverse multiplicities. Ties share rank; ranges across scenarios are deterministic sensitivities, not confidence intervals or probabilities of being best.

D Complete configuration-level results

Table D.1 preserves unchanged grades for every tested endpoint across the three selection views. Requested identifiers describe archived calls, not current availability or verified backend weights. Exact and semantic actions retain their distinct parsing gates.

Table D.1. Every tested configuration in execution-group order. All counts use 100 planned slots per endpoint.

Group	Archived requested endpoint	T	S_0	S_b	S	E	F	M
G1	openai/gpt-5.4-mini	86	50	50	51	80	80	14
G1	openai/gpt-5.5	100	94	94	97	99	99	0
G1	openai/gpt-5.6-luna	100	62	62	62	88	88	0
G1	openai/gpt-5.6-sol	100	96	96	96	99	99	0
G1	openai/gpt-5.6-terra	100	66	66	66	97	97	0
G1	openai/gpt-6-astra	100	78	78	100	100	100	0
G1	openai/gpt-oss-120b	54	44	44	44	52	52	46
G2	anthropic/claude-fable-5	100	60	69	69	73	73	0
G2	anthropic/claude-fable-5.1	100	70	80	81	91	97	0
G2	anthropic/claude-haiku-4.5	100	0	0	0	0	74	0
G2	anthropic/claude-opus-5	99	76	84	87	95	95	1
G2	anthropic/claude-sonnet-5	87	41	61	63	82	85	13
G3	google/gemini-3.1-pro-preview	100	90	90	90	99	99	0
G3	google/gemini-3.5-flash-lite	99	37	37	37	76	76	1
G3	google/gemini-3.7-flash	100	84	84	84	100	100	0
G3	google/gemini-3.8-flash	100	93	93	93	100	100	0
G4	qwen/qwen3.8-2.4t-a95b	66	6	6	6	28	28	34
G4	qwen/qwen3.8-27b	75	2	2	2	26	26	25
G4	qwen/qwen3.8-flash	63	10	10	10	24	24	37
G4	qwen/qwen3.8-max-0902	62	4	4	4	23	23	38
G5	deepseek/deepseek-v3.2	100	50	50	51	86	86	0
G5	deepseek/deepseek-v4-pro-0813	95	85	85	87	93	93	5
G5	deepseek/deepseek-v4.1-flash	93	85	85	86	92	92	7
G6	meta-llama/llama-4-maverick	100	2	2	2	46	46	0
G6	meta/muse-glimmer-30b	100	55	55	55	92	92	0
G6	meta/muse-spark-1.3	96	85	87	91	95	95	4
G7	mistralai/mistral-medium-3-5	99	36	36	36	85	85	1
G7	mistralai/mistral-small-2603	21	3	4	4	14	14	79
G8	nvidia/nemotron-3-super-120b-a12b	45	30	32	33	43	43	55
G8	nvidia/nemotron-3-ultra-550b-a55b	76	58	58	58	75	75	24
G8	nvidia/nemotron-3.5-lightning	100	0	0	0	21	21	0
G9	z-ai/glm-4.6v	100	18	18	18	57	58	0
G9	z-ai/glm-5.3	100	3	3	3	29	29	0
G9	z-ai/glm-5.3-flash	39	16	21	21	33	33	61
G10	amazon/nova-2-lite-v1	74	2	2	2	2	65	26
G10	bytedance-seed/seed-2-1-turbo	59	56	57	57	59	59	41
G10	bytedance-seed/seed-2.0-lite	100	45	45	45	89	89	0
G10	cohere/command-a	100	3	3	3	41	41	0
G10	minimax/minimax-m2.7	68	43	43	44	60	60	32
G10	minimax/minimax-m3	84	21	21	21	26	80	16
G10	moonshotai/kimi-k2.6	70	58	58	63	70	70	30
G10	moonshotai/kimi-k3	97	67	69	72	91	91	3
G10	thinkingmachines/inkling	84	55	55	56	81	82	16
G10	thinkingmachines/inkling-small	100	59	59	59	94	94	0
G10	upstage/solar-pro4	63	33	34	39	49	49	37
G10	x-ai/grok-4.5	90	84	84	84	88	88	10
G10	x-ai/grok-4.6	90	80	82	82	88	88	10
G10	xiaomi/mimo-v2.5	68	37	37	40	61	61	32
G10	xiaomi/mimo-v2.5-pro	78	43	43	44	72	72	22
G11	arcee-ai/trinity-large-thinking	62	34	35	38	57	58	38
G11	ibm-granite/granite-4.2-8b	53	21	21	22	42	42	47
G11	inception/mercury-2.5	100	23	23	23	67	68	0
G11	microsoft/phi-4	100	2	2	2	41	41	0
G11	nousresearch/hermes-4-405b	100	4	4	4	38	36	0
G11	poolside/laguna-s-2.1	53	17	17	17	19	41	47
G11	rekaai/reka-flash-3	65	0	0	0	0	0	35
G11	writer/palmyra-x5	100	3	3	3	51	51	0

T: terminal; S_0, S_b, S: primary, baseline, completion strict; E: exact action; F: semantic action; M: unfilled.

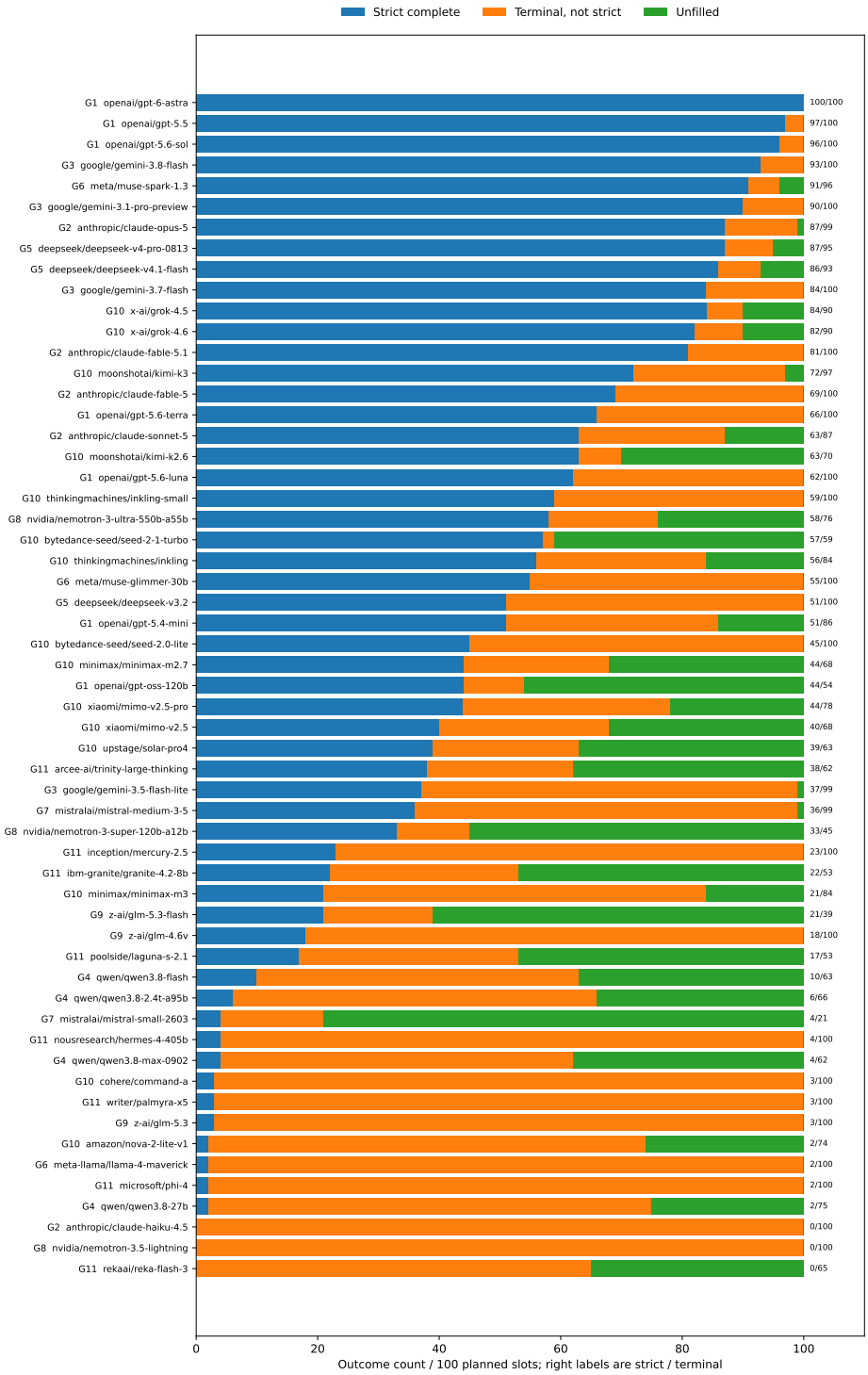

Figure D.1. Complete tested-model overview. Each bar partitions the same 100 planned slots into strict successes, other terminal outcomes, and unfilled slots. Ordering by observed strict count is descriptive. The full endpoint counts and original selection views are given in Table D.1.

E Difficulty, execution groups, and fixed-state repeatability

E.1 Designed difficulty

Table E.1 preserves the actual executed stratum counts rather than an earlier proposed mixture. The medium stratum contains twenty task IDs. The no-answer stratum measures intentional nonidentification, not a failed oracle. Differences across these designed labels combine grammar, output requirements, and coverage; they cannot by themselves calibrate a universal scale of difficulty.

Table E.1. Designed difficulty strata in the executed suite.

Designed label	Task IDs	N	T	S	E	S/N (%)
easy	10	570	535	334	411	58.6
medium	20	1,140	1,023	609	754	53.4
hard	25	1,425	1,208	591	884	41.5
extra_hard	20	1,140	886	345	677	30.3
oracle	20	1,140	888	399	624	35.0
no_answer	5	285	273	229	229	80.4

Labels describe the construction, not empirically calibrated hardness. The 20 oracle items are answerable. Only the five no_answer items intentionally have nonidentified targets.

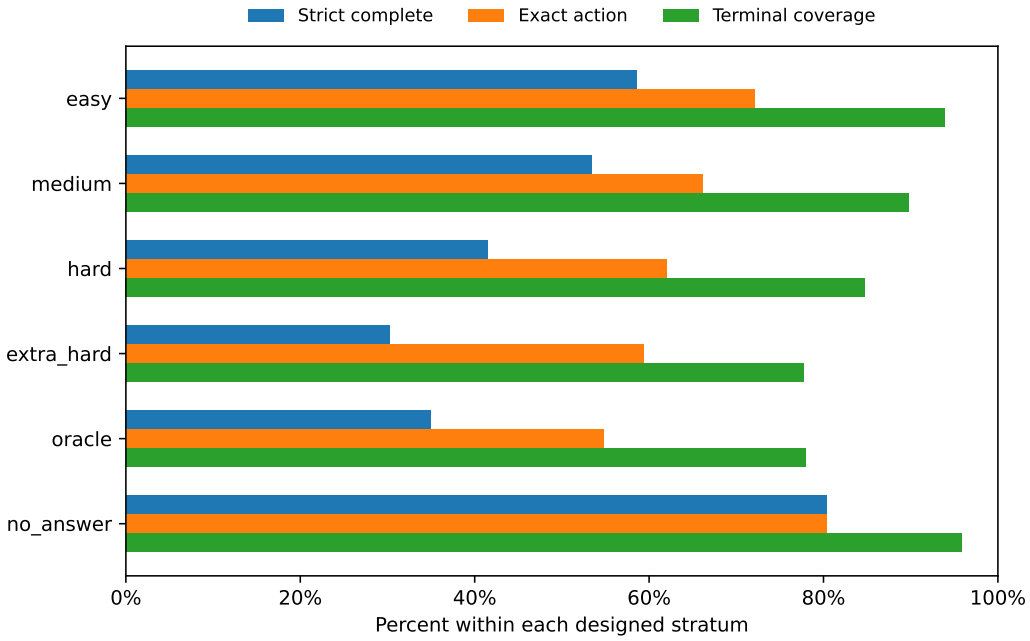

Figure E.1. Outcome rates by designed difficulty stratum. Each stratum uses its own planned-slot denominator. Oracle tasks are answerable, and the nonidentified no-answer items form a distinct contract. The ordering of labels does not impose an empirically validated monotonic hardness scale.

E.2 Coverage within execution groups

Figure E.2 provides the full coverage partition underlying Table 2. A low strict fraction can result from nonterminal outcomes, terminal answers without exact-action success, or incomplete numerical contracts. The graphic retains unfilled slots explicitly rather than converting them into inferred cognitive mistakes. Group rosters are heterogeneous, and the two mixed groups remain composition-weighted descriptions.

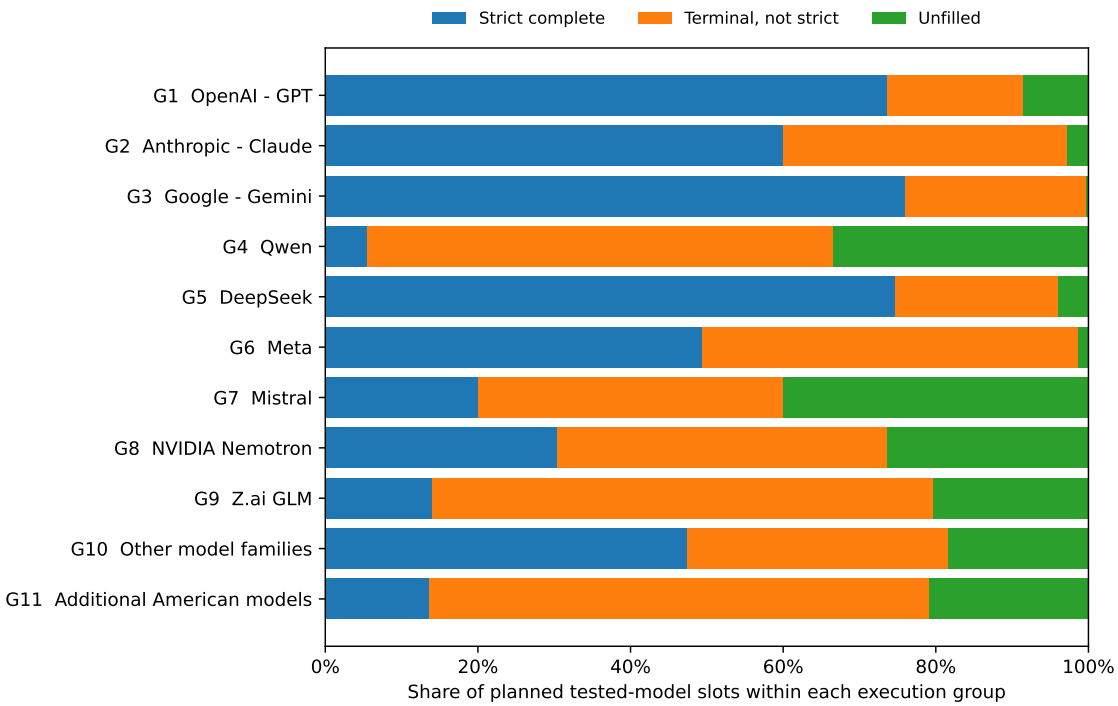

Figure E.2. Execution-group coverage partition. Strict successes, other terminal outcomes, and unfilled slots sum to the planned tested denominator in each group. This is a summary of the collected rosters, not a developer-population or geographic ranking.

E.3 All-success reliability is not best-of-k success

For the challenge state repeated eight times, the primary reliability diagnostic uses disjoint blocks of size $k \in \{1, 4, 8\}$. Let $\mathcal{B}_k$ be the planned blocks, and let s_r and t_r denote strict success and terminal status for repeat r. Planned-block and fully observed-block all-success rates are

$$\text{Pass}_k^{\text{planned}} = \frac{\sum_{B \in \mathcal{B}_k} \prod_{r \in B} s_r}{|\mathcal{B}_k|}, \qquad \text{Pass}_k^{\text{observed}} = \frac{\sum_{B \in \mathcal{B}_k} \prod_{r \in B} s_r}{\sum_{B \in \mathcal{B}_k} \prod_{r \in B} t_r}. \tag{E.1}$$

The observed rate is undefined when no complete block is available. A block containing a known failure cannot become all-success after filling missing observations. If U_k counts incomplete blocks with no observed failure, the all-success identification interval is

$$\left[\frac{S_k}{|\mathcal{B}_k|}, \frac{S_k + U_k}{|\mathcal{B}_k|} \right], \qquad S_k = \sum_{B \in \mathcal{B}_k} \prod_{r \in B} s_r. \tag{E.2}$$

These all-success quantities are not best-of-k pass@k. The latter asks whether *at least one* attempt succeeds, whereas Eq. (E.1) requires every member of a block to succeed.

A secondary combinatorial diagnostic considers all overlapping subsets of the eight repeats. If s repeats succeed and r are terminal, its planned and observed all-success fractions are $\binom{s}{k}/\binom{8}{k}$ and $\binom{s}{k}/\binom{r}{k}$, when the denominators are defined. These subsets overlap and are not independent experiments. Both diagnostics characterize one designed state, not a population-wide reliability guarantee.

Table F.1. Observed nondominated selected-answer cost–quality points among complete configurations.

Requested endpoint (short name)	$S/100$	Suite cost (USD)	Cost / terminal	Cost / strict
phi-4	2	0.0052	0.00005	0.00262
glm-5.3	3	0.0587	0.00059	0.01955
hermes-4-405b	4	0.0978	0.00098	0.02446
mercury-2.5	23	0.1033	0.00103	0.00449
deepseek-v3.2	51	0.7079	0.00708	0.01388
gpt-5.6-luna	62	1.0864	0.01086	0.01752
gpt-5.6-sol	96	5.0986	0.05099	0.05311
gpt-6-astra	100	13.6095	0.13610	0.13610

The cohort comprises 24 terminal-complete configurations. Costs cover selected terminal responses only, not all acquisition attempts or the total cost of running an evaluation. Historical charges are not current list prices.

F Cost scope and telemetry

Table F.1 reports the selected-answer frontier in numerical form. Native charges are summed using decimal arithmetic before display rounding. The denominator is the selected terminal or strict-success count as indicated; unknown costs are not imputed as zero. The cohort is restricted to terminal-complete configurations with known selected-answer costs, rather than pooling incomplete runs with different acquisition coverage.

Selected-answer costs are a subset of acquisition economics. Repeated failed requests, unselected outcomes, unresolved charge states, infrastructure, review, and support are outside this numerator. Therefore neither the frontier nor a cost-per-strict-answer ratio predicts the total price of a future evaluation service.

Reasoning-token counts are generally included within the retained completion-token accounting semantics and are not added again to produce a total. Native inconsistencies remain flagged rather than converted into valid fractions. Requested output ceilings are not realized reasoning amounts. Selected latency statistics are conditioned on terminal outcomes and may combine serving, queueing, generation, and client effects. Missing timings remain missing; timing proxies are not silently relabeled direct measurements. None of these observational quantities identifies a causal effect of a larger inference budget.

G Source identity and reproducibility boundary

The numerical source is `MAHJ-AsExecuted-v1.1-20260924`, with its associated `MAHJ-Stage3-v1.1-20260924` analysis [11]. Tables are regenerated from the associated aggregate CSV exports, figures retain the same aggregate evidence, and numeric text is linked to source fields or explicit arithmetic on those fields. Display rounding does not feed subsequent calculations. Original task definitions, selection roles, capability mappings, and source scores remain unchanged.

Frozen evidence archive (SHA-256)
`ca3bdd421ed06918cee1ed3bd940502024704 6d5b996bf3dea2264b23ae24aaf`

Frozen content manifest (SHA-256)
`6b7c9e0545f2969a0f0447c4554a6e3374db497f4db4a0a6fb119ef49772604d`

Analysis content manifest (SHA-256)
`7327098bf55d8b721e333656c3c25e00e102ca3e133018c865d0b8ebf467ed64`

These identifiers bind the evidence; they do not imply unrestricted access to task-level source material.

The aggregate companion contains model-, group-, capability-, and difficulty-level outcomes; calibration bins and threshold summaries; paired effects and uncertainty endpoints; selected-cost summaries; configuration metadata; a data dictionary; and publication-layer reproduction code. The retained capability registry omits production question mappings. Empty numeric cells preserve unknown or ineligible values rather than inventing zeros.

The offline numerical reconstruction supports the following commands from the aggregate package root:

```
python code/reproduce.py -verify-only
python code/reproduce.py -output reproduced
python code/reproduce.py -output reproduced -figures
```

The latter two commands are alternatives; each uses a new output directory. Figure rendering requires the companion's recorded plotting dependencies. Aggregate reconstruction does not call a model or overwrite source evidence. Typesetting the paper is separate from numerical verification.

Public reconstruction verifies the publication layer: it rebuilds reported scalar values, tables, and figures from the exported summaries. It cannot regenerate the withheld questions, replay raw-response scoring, inspect oracle instances, or independently recompute a prompt-level bootstrap from per-model aggregates alone. The paired intervals and configuration sensitivities are documented outputs of the bound analysis, not newly estimable quantities from the public summaries. This boundary is part of the method and limits the scope of independent replication.