LINALG-BENCH: WHEN LLMS STOP COMPUTING — STRUCTURED HALLUCINATION, SIGN DRIFT, AND DEPTH-GATED COLLAPSE IN MATRIX ARITHMETIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Frontier LLMs execute linear algebra flawlessly — until, at a measurable depth, they stop computing and start fabricating. We introduce LinAlg-Bench, a diagnostic benchmark of nearly 1,200 SymPy-verified problems across nine task types and 3×3–5×5 matrices, evaluated on ten frontier models at temperature zero (over 11,000 graded outputs, with results stable across three complete runs). Failure is not gradual but structural: models near ceiling on shallow tasks collapse to near zero on 5×5 eigenvalues — and the collapse is spectrum-conditional, with over 40% aggregate accuracy on integer spectra versus under 15% on irrational ones, implicating required numerical depth rather than matrix size. A three-stage forensic pipeline reveals that failures dissociate cleanly by task: eigenvalues fail by constraint-aware confabulation — fabricated values still match the matrix trace in nearly half of cases; determinants fail by sign accumulation, with a dimension-gated complete collapse that is absent at 4×4 yet dominant at 5×5. Forcing an $O(n^3)$ algorithm yields no recovery, chain-of-thought prompting rescues only the one model already strongest on the task, and chained shallow operations with total arithmetic matched to one eigenvalue computation remain near-ceiling — together implicating depth-of-state, not operation count, strategy, or prompting, as the bottleneck. LinAlg-Bench frames this working-memory limit as a falsifiable hypothesis and provides the instrument to test it.

1 INTRODUCTION

How deep can a frontier language model carry an exact computation before execution gives way — and what does it do when it can no longer continue? Answering this requires an instrument, not a leaderboard. LinAlg-Bench holds the algorithm constant and scales only matrix dimension, adds a factorial eigenvalue core that independently varies matrix symmetry and spectrum type, and attaches a forensic pipeline that classifies every failure by its earliest divergence from ground truth. Linear algebra is the ideal substrate: its operations are algorithmically precise, scale across dimension without changing the underlying task, and form a hierarchy in which each level adds exactly one computational demand absent from the level below — a model that fails a 5×5 determinant cannot blame unfamiliar mathematics, because the algorithm is identical to the 3×3 case it solved.

A demonstration study of ten frontier models shows what the instrument sees. At 3×3, nine of ten models are at or near perfect determinant accuracy. At 5×5 the population splits — four models retain ≥90.9% while four collapse to ≤6.3% — and eigenvalue accuracy falls to 11.0% overall, concentrated almost entirely where the answer has compact exact structure (integer spectra 19.7% vs. irrational 2.3%), motivating the working-memory account of Section 6. The mode of failure changes with it: at 3×3 models compute and slip; at 5×5 they increasingly abandon computation, fabricating answers that satisfy surface constraints — a shift documented across 11,940 exhaustively evaluated outputs. To locate the bottleneck behind these thresholds we triangulate with four designed probes — a forced-strategy ablation, an operation-count control, an integration-by-parts depth ladder outside linear algebra, and a five-level procedural scaffold — which together separate depth from volume, domain from mechanism, and universal limits from model-dependent ones.

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

Table 1: Five cognitive levels of the LinAlg-Bench taxonomy. Each level adds exactly one computational requirement absent in the level below; the critical boundary falls between Sequential and Recursive.

Level	Cognitive Demand	Tasks	Key Operation
Reading	Pure entry extraction	Trace, Transpose	Read and report entries
Arithmetic	Parallel independent computation	Mat-Vec, Multiplication, Power	Independent dot products
Sequential	State tracking through row ops	Rank, Nullity	Gaussian elimination
Recursive	Deep recursive sign management	Determinant	Cofactor expansion, ± 1 parity
Compositional	Determinant as sub-routine	Eigenvalues	Char. polynomial via determinant

LinAlg-Bench contributes, as an instrument: 1,194 SymPy-verified problems across 9 tasks and 3 dimensions with a 360-problem factorial eigenvalue core crossing symmetry with spectrum type; documented generator recipes enabling contamination-resistant regeneration of fresh, statistically matched problem sets; and a three-stage forensic judge pipeline whose error taxonomy transfers beyond linear algebra (Appendix O). As a demonstration study: the spectrum-type finding — answer structure, not matrix structure, predicts eigenvalue success; the 5×5 determinant bifurcation; the four-probe localisation of the bottleneck to sequential execution depth; and constraint-aware confabulation quantified at scale. All data, model outputs, error labels, generators, and the judge pipeline are publicly released.

2 Related Work

Mathematical-reasoning benchmarks have moved from arithmetic word problems (Cobbe et al., 2021) to competition mathematics (Hendrycks et al., 2021), multi-domain suites (Mishra et al., 2022), symbolic mathematics (Lample & Charton, 2020), and multi-level comprehension (Liu et al., 2024), but they measure final-answer accuracy on tasks where difficulty and novelty are confounded. LinAlg-Bench instead holds the algorithm fixed and scales only matrix dimension, isolating computational depth from mathematical knowledge — so failure mode, not merely failure rate, becomes measurable.

Transformers solve compositional tasks by shallow pattern-matching (Dziri et al., 2023), cannot reliably self-correct arithmetic (Huang et al., 2023), and confabulate through factual misgrounding or overconfident generation (Maynez et al., 2020; Kadavath et al., 2022); we document a structurally distinct mode — constraint-aware confabulation, fabrications that satisfy cheaply checkable invariants while violating expensive ones (§5.4). Process supervision improves multi-step reasoning (Lightman et al., 2023); Section 6 instead uses a five-level procedural scaffold as a measurement device, asking whether externalising intermediate state recovers execution. At the behavioural level, the forensic taxonomy hands the mechanistic-interpretability literature (Vig, 2019; Meng et al., 2022; Conmy et al., 2023) concrete, testable hypotheses.

3 Benchmark Design

LinAlg-Bench organises 9 linear algebra tasks into five cognitive levels based on the computational demands each operation places on a language model's processing pipeline. This taxonomy is not a difficulty ranking but a causal decomposition: each level adds exactly one computational requirement absent in the level below (Appendix B). The critical boundary falls between Sequential and Recursive: determinant computation requires maintaining hierarchical parity state across nested cofactor expansions simultaneously, while rank and nullity require only independent row operations (Gaussian elimination: the step-by-step row-reduction procedure in which each row update builds on the result of the previous step). Eigenvalue at 5×5 compounds this as a deliberate stress test — not a standard capability measurement — requiring a full 5×5 determinant followed by degree-5 polynomial root-finding.

The benchmark comprises 1,194 integer-entry problems spanning three dimensions — 317 at 3×3, 442 at 4×4, and 435 at 5×5 — each ground-truth verified against SymPy symbolic computation (Meurer et al., 2017). Determinant and eigenvalue receive the largest allocations (50/175/175 and 127/127/120 problems across the three dimensions, respectively) because they are the primary Recursive and Compositional tasks under investigation; the remaining seven subcategories use n=20

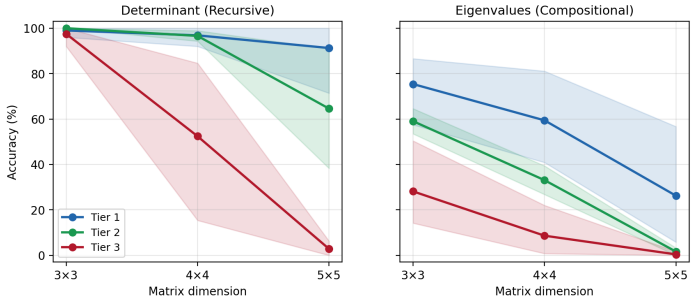

Figure 1: Accuracy across the dimensional gradient (1,194 problems × 10 models). Determinant falls 98.6% → 79.0% → 50.6% and eigenvalue 53.2% → 33.9% → 11.0% from 3×3 to 5×5; Reading- and Arithmetic-level tasks stay flat. Declared-pass scoring for OpenAI-o1 and GPT-5.2: Appendix C.

per dimension, sufficient to establish ceiling and floor effects at the Reading, Arithmetic, and Sequential levels.

The eigenvalue tier is a full factorial design with three factors: matrix symmetry (symmetric vs. non-symmetric with real spectrum), spectrum type (integer vs. irrational eigenvalues), and dimension (3×3, 4×4, 5×5). Each of the 12 cells contains 30 problems — 360 problems in total; 14 further problems with repeated eigenvalues form a separately labeled subset outside the factorial. The two factors separate two competing explanations of failure: spectrum type controls whether the answer has a compact exact form, while symmetry varies the structure of the matrix itself, independently of the answer format. Eigenvalue responses are graded as unordered sets of real values with per-value tolerance 0.011; because true eigenvalues are never closer than 0.33, deviations at this scale cannot confuse one spectrum for another. Integer-spectrum answers are accepted in integer or decimal form. Grading rules for all nine tasks and the automated judge pipeline are specified in Appendix B and Appendix E.

Ten frontier models spanning reasoning-optimised, mixture-of-experts, and standard instruction-tuned architectures were evaluated zero-shot at temperature 0 via API; full model details are in Appendix A. Code execution and external tools were disabled — this is a deliberate synthetic stress test of unaided recursive execution under increasing depth, not a practical evaluation of tool-use capability. For the two most expensive models, a declared-pass protocol was applied at 3×3 and 4×4 with direct measurement on all 5×5 problems; the protocol and its validation are documented in Appendix C, and every affected table is footnoted. Format sensitivity was evaluated at 4×4 and 5×5 across three input representations (LaTeX, Tabular, List) for nine models; format analysis targets non-reasoning architectures where parsing load interacts with computational depth (Appendix D).

4 Results

Figure 1 shows the central result: failure is not gradual but level-gated. Reading- and Arithmetic-level tasks stay flat at every dimension, Sequential tasks (rank, nullity) decline only gradually with Gaussian-elimination chain length, and the cliff concentrates entirely at the Recursive and Compositional levels. Full per-model and per-subcategory accuracy is in Appendix C (Tables C.1–C.3).

The decisive break is the determinant step at 5×5 — a 28.4-point drop from 4×4 that bifurcates rather than degrades: OpenAI-o1 (100%), Gemini-3.1-Pro (100%), Qwen3-235B (93.7%) and GPT-5.2 (90.9%) stay at or above 90.9%, while Claude-4.5-Sonnet (6.3%), Llama-3.3-70B (5.1%), Qwen2.5-72B (0%) and GPT-4o (0%) collapse. Even the top determinant models converge toward low eigenvalue accuracy by 5×5 (11.0% overall; best Gemini-3.1-Pro 56.7%, OpenAI-o1 32.5%), concentrated on spectra without compact exact form (§5.3) — two distinct failure thresholds, not a continuum, consistent with the compositionality gap (Press et al., 2022). Accuracies are stable across three complete re-runs of the 660-problem set (maximum deviation 2.6pp, mean 1.3pp, rankings unchanged; Appendix M), and format sensitivity is second-order: mid-tier models lose a few points on non-LaTeX renderings while fully collapsed models fail in every format — a computational limit, not a parsing artifact (Appendix D).

5 FORENSIC ERROR ANALYSIS

5.1 CLASSIFICATION METHODOLOGY

Forensic error classification was applied to all 3,933 incorrect responses (622 at 3×3, 1,277 at 4×4, 2,034 at 5×5), following the first-error principle: the earliest step at which computation diverges from ground truth is tagged, not the final observable symptom. The taxonomy comprises ten primary tags applicable across all nine subcategories — execution errors (sign_error, arithmetic, carry_down_error, memory_loss), structural errors (hallucination, method_fail), and artifacts (input_transcription, generation_truncation, formatting_mismatch, other_unmapped), defined in Table 2 — plus four eigenvalue-specific tags (generation_loop, algebraic_precedence, false_verification, variable_entanglement) that arise exclusively during characteristic polynomial expansion and root-finding, defined in Appendix E, Table E.2. All distributional analysis in Section 5 uses the ten primary tags; the eigenvalue-specific tags are reported in Appendix E.

Ten primary error tags of the LinAlg-Bench forensic taxonomy, grouped by family. Classification follows the first-error principle: the earliest step at which computation diverges from ground truth is tagged. Two prompt-level rules precede semantic judgment: the Magnitude Rule ($|\text{wrong}| = |\text{correct}| \rightarrow$ SIGN_ERROR) and the Truncation Precheck (no final answer $\rightarrow$ GENERATION_TRUNCATION).

Table 2: Ten primary error tags of the LinAlg-Bench forensic taxonomy, grouped by family. Classification follows the first-error principle: the earliest step at which computation diverges from ground truth is tagged. Two prompt-level rules precede semantic judgment: the Magnitude Rule (|wrong| = |correct| \rightarrow sign_error) and the Truncation Precheck (no final answer \rightarrow generation_truncation).

Tag	Family	Definition
SIGN_ERROR	Execution	Correct magnitude, wrong sign — parity or product error
ARITHMETIC	Execution	Wrong magnitude — numerical computation error
CARRY_DOWN_ERROR	Execution	Correct at step N, miscopied at step N+1
MEMORY_LOSS	Execution	Correctly stated at step N, recalled wrong at step M ($M \gg N$)
HALLUCINATION	Structural	Abandons or fabricates computation
METHOD_FAIL	Structural	Wrong algorithm applied from the first step
INPUT_TRANSCRIPTION	Artifact	Matrix misread at input before any arithmetic
GENERATION_TRUNCATION	Artifact	Response ends without a final answer
FORMATTING_MISMATCH	Artifact	All values correct, wrong output format
OTHER_UNMAPPED	Residual	Does not fit any above tag

Two rules are enforced before any semantic judgment: the Magnitude Rule ($|\text{wrong}| = |\text{correct}| \rightarrow$ SIGN_ERROR, never ARITHMETIC) and the Truncation Precheck (no final answer $\rightarrow$ GENERATION_TRUNCATION, never HALLUCINATION). Classification was performed by a three-stage pipeline following the LLM-as-a-Judge paradigm (Zheng et al., 2023): the Build Judge traces each response step by step to its first divergence from ground truth; a blind Validate Judge — given only the proposed tag and the original response — confirms or corrects it against tag-specific verification questions; and systematic disagreements were then resolved by human batch review, which iteratively refined the taxonomy. To validate pipeline reliability, the pipeline was checked against independent manual review of 2,520 responses: agreement is 91.2% overall (2,298 of 2,520), rising with matrix size to 93.0% at 5×5 (1,257 of 1,351) — highest exactly where most failures occur (Appendix E). All 11,940 outputs were evaluated at temperature 0. A single pass is sufficient: three complete runs of a 660-problem subset put the mean run-to-run deviation at 1.3 percentage points (maximum 2.6) with stable rankings (Appendix M). Because the benchmark is a complete enumeration of the defined problem set rather than a sample, standard errors and confidence intervals over model accuracy are not applicable.

5.2 FAILURE MODE SHIFT

Failure modes shift systematically with matrix dimension — and the subtypes shift inside them. At 3×3 (n=622), arithmetic leads (33.1%): models compute honestly and slip. The subtypes agree: sign

errors are dominated by multiplication slips (Product block, 68.0%; Appendix F), and hallucinations are still fabrications — 91.7% (143 of 156) are invented answers following a genuine failed attempt (Appendix G). At 4×4 (n=1,277), sign_error surges to 42.1%, and its dominant subtype flips to Parity (54.3%) — the plus/minus bookkeeping of cofactor expansion collapses, the largest single subtype shift in the dataset. Hallucination flips the same way: 81.0% (255 of 315) is now abandonment rather than fabrication — models begin to quit instead of inventing. At 5×5 (n=2,034), hallucination reaches 57.9% — 90.5% of it abandonment (1,066 of 1,178) — while sign_error retreats to 20.2% and arithmetic to 8.9%. The retreat is not recovery: the sign errors that remain are still dominated by parity bookkeeping (50.4%), and the raw counts stay level (Figure 2).

The shift unfolds in two stages. First, long-chain error types (generation_loop, memory_loss, variable_entanglement) appear once computation chains grow long, marking the onset of working-memory stress. Second, the regime flip completes at 5×5: hallucination overtakes sign_error, driven overwhelmingly by Complete_Collapse (Appendix H). The sequence — honest slips, collapsed bookkeeping, abandonment — is the behavioral signature of a working-memory limit; the persistence of sign errors across model tiers is consistent with findings that transformers systematically struggle with signed arithmetic (Lee et al., 2023).

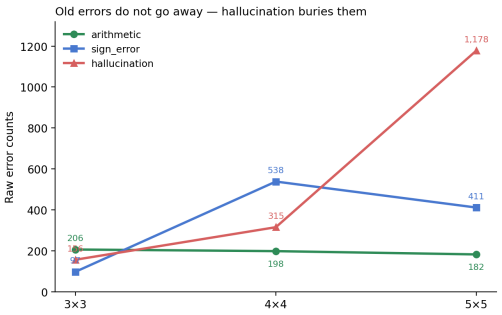

Figure 2: Old errors do not go away. Raw error counts by dimension: arithmetic stays flat (206, 198, 182) and sign errors persist (538 → 411); only hallucination grows (156 → 315 → 1,178), swelling the failure pool from 622 to 2,034 classified failures. The shares hand off accordingly — arithmetic leads at 3×3 (33.1%), sign_error at 4×4 (42.1%), hallucination at 5×5 (57.9%); full distribution in Appendix H.

5.3 Spectrum Type, Not Symmetry, Predicts Eigenvalue Success

The factorial eigenvalue core separates two candidate explanations of the 5×5 eigenvalue collapse: matrix structure (symmetric versus non-symmetric) and answer structure (integer versus irrational spectrum). Only answer structure matters. Aggregated over all 10 models and 360 problems, integer-spectrum accuracy is 50.7% against 12.1% for irrational spectra — a 4.2× gap that holds at every dimension (3×3: 79.7% vs 22.7%; 4×4: 52.7% vs 11.2%; 5×5: 19.7% vs 2.3%) — while symmetry moves aggregate accuracy by under three points (30.1% vs 32.6%) and never reverses an ordering (Figure 3). The answer's structure matters this much because it decides which route to a solution exists: an integer spectrum can be found by search — propose small-integer candidates, verify them against cheap invariants like the trace — whereas an irrational spectrum can only be computed, by carrying the characteristic polynomial through recursion and root-finding. Integer-spectrum accuracy therefore measures how searchable a problem is; irrational-spectrum accuracy is the benchmark's true measure of computational depth — and by that measure the best model at 5×5 scores 13.3%.

Transcript forensics confirm the two routes directly (Appendix I). Gemini-3.1-Pro's perfect 60/60 on 5×5 integer spectra is constraint-anchored search, not computation: 51 of the 60 transcripts verify a proposed spectrum against the trace and 50 against the determinant, while only a minority ever derive the characteristic polynomial. On irrational spectra the identical procedure becomes confabulation — integer candidate sets matching the trace still exist, so the model reports "verified" spectra that cannot be right; the checks pass either way. OpenAI-o1, the one model that computes rather than searches, obeys the same law from the other side: its integer-versus-irrational pass rates fall from 100%/72% at 3×3 to 60%/5% at 5×5 — the price of chaining hundreds of dependent arithmetic steps. The lone symmetry effect in the data reinforces the account: Gemini solves irrational spectra only on non-symmetric matrices (8/30), never on symmetric ones (0/30), where $\mathrm{Tr}(A^2) = \Sigma a_{ij}{}^2$ makes the verification shortcut cheapest — the cheap check does not make the mathematics harder; it makes fabrication more attractive. We treat this as a falsifiable claim: high integer-spectrum accuracy reflects shallow root-verification; irrational-spectrum accuracy is the better measure of genuine execution.

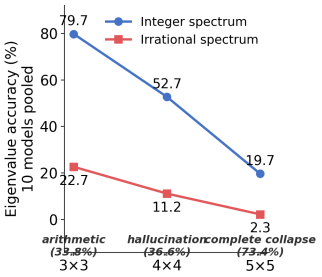

Figure 3: The spectrum gap holds at every dimension: pooled over ten models, integer-spectrum eigenvalue accuracy falls 79.7% → 52.7% → 19.7%, against 22.7% → 11.2% → 2.3% for irrational spectra.

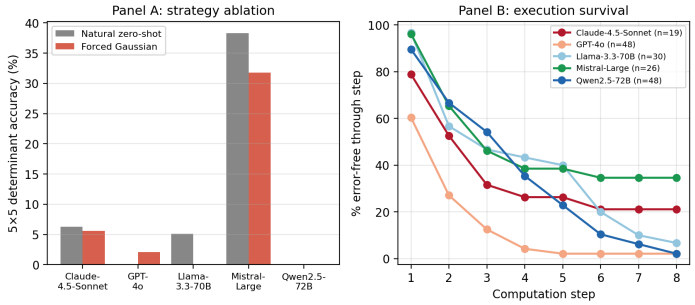

Figure 4: Forced-Gaussian ablation. Panel A: accuracy on a per-model subset of 5×5 determinant problems under enforced Gaussian elimination (Mistral-Large 31.8%, Claude-4.5-Sonnet 5.6%, GPT-4o 2.1%, Llama-3.3-70B and Qwen2.5-72B 0%); each model was evaluated on its own subset (selection rule in Appendix K). Panel B: per-step execution survival; by step 2 only 27–67% of traced runs remain error-free.

The failure mode itself changes kind as depth grows. At 3×3, hallucinations are fabrications by models still doing the work: 68.6% are ungrounded guesses — an invented answer following a failed computation. By 4×4 the dominant mode is already Complete Collapse (71.3%), and at 5×5 it reaches 85.4%: models quit on sight of the matrix without computing at all. Each model exits in a recognizable tier style — from simulated tool runs to immediately boxed fabricated answers (Appendix I, Table I.4). What they emit instead is not random guessing but confabulation, structured by whichever check remains cheap — quantified in Section 5.4; genuine computation survives only where the answer can be verified with whole-number arithmetic.

5.4 Constraint-Aware Confabulation and Tool Collapse

hallucination at 5×5 is not random. We test this at scale: for every eigenvalue response tagged Complete_Collapse (1,014 cases), we parse the model's final answer and check it against two exact invariants of the given matrix — the trace (the eigenvalue sum) and the determinant (the eigenvalue product). Among the 579 cases yielding a full parsed set of the correct cardinality, 45.4% match the trace but only 8.5% match the determinant. The asymmetry is diagnostic: the trace is readable directly off the diagonal at a glance, while the determinant requires precisely the recursive computation the model has abandoned. Fabrication under overload is thus structured by whatever constraints remain cheaply checkable — and a benchmark evaluating only such necessary conditions would systematically misclassify these fabrications.

Constraint-awareness follows the capability hierarchy. Among models with at least 40 parsed fabrications, trace-match rates span 100% (Qwen3-235B), 87.9% (DeepSeek-V3), and 70.0% (Claude-4.5-Sonnet) down to 18.5% (Qwen2.5-72B) and 10.6% (Llama-3.3-70B), while determinant-match stays at or below 12.1% for every model. Stronger models fabricate more plausibly, not less. The remaining cases are themselves informative: 57 responses fabricate complex eigenvalues for matrices with certified all-real spectra, and 94 produce no final answer at all.

Across 784 Complete_Collapse instances at 5×5 eigenvalue, the failure follows a consistent three-step pattern: the model correctly identifies the task and sets up the characteristic equation, generates a meta-statement citing complexity, then fabricates eigenvalues via simulated tool invocation — citing Python, NumPy, or WolframAlpha it does not have access to. Tool use was disabled for all models; several models explicitly output the system constraint before violating it, confirming tool invocation is hallucinated rather than real. This contrasts sharply with models explicitly trained to call external APIs (Schick et al., 2023); the tool-use syntax present in pretraining data appears to be triggered as a collapse response when mathematical execution is overloaded. This simulated tool invocation likely stems from the prevalence of program-aided reasoning in pretraining mixtures (Gao et al., 2023); when internal computation fails, the model falls back on code-generation syntax.

Full model-specific collapse personas are documented in Appendix L.

6 INTERPRETATION

6.1 THE WORKING MEMORY ACCOUNT

The shift documented in Section 5 — execution errors giving way to abandonment — is a threshold, not a slope. We propose a working-memory account as a falsifiable hypothesis, not an identified mechanism: models fail when a computation requires carrying more dependent intermediate state than they can reliably hold in their own generated context. Four observations point the same way.

- **Same algorithm, more state.** Models describe cofactor expansion correctly at 5×5 and execute it correctly at 3×3; between the two, only the live intermediate state grows — knowledge constant, state up, accuracy collapsed. A knowledge deficit cannot produce this pattern; a capacity limit does.
- **Failure changes kind, not amount.** At 3×3 models compute and slip (arithmetic 33.1%, sign 15.6% inside real attempts); at 5×5, 57.9% of failures are fabricated or abandoned answers with no genuine computation attempted. A skill running out produces more of the same errors; a capacity being exceeded produces a change in behavior.
- **Long-chain error types.** Looping on a step, retrieving a stale intermediate, swapping live variables — these tags appear only where chains are long, including 3×3 irrational eigenvalues (55 of 57 small-matrix cases), and across every model tier and training paradigm, ruling out a training-data explanation: direct sightings of state dropped mid-chain.
- **Success follows the shortcut, not the problem.** Whole-number spectra admit a memory-cheap route (guess small integers, check trace and determinant; Appendix J); messy spectra force full execution. Models succeed 4.2× more often when the shortcut exists (8.4× at 5×5) while indifferent to matrix structure (symmetry 30.1% vs 32.6%) — the fingerprint of a memory limit, not a knowledge limit.
- **The exception: sign errors.** These persist at every size instead of growing with chain length — a tokenisation-level phenomenon (Nogueira et al., 2021; Brown et al., 2020) and a separate mechanism requiring a separate explanation. Architectural conjectures are documented in Appendix L.

6.2 TESTING THE ACCOUNT: FOUR EXPERIMENTS

A hypothesis this convenient must be made to risk failure. Each of the following experiments removes one rival explanation.

Experiment 1 — forced strategy: is it method choice? A determinant can be computed two ways: cofactor expansion or Gaussian elimination. Cofactor expansion is the lengthy one — roughly $n \cdot n!$ operations against n^3 for Gaussian elimination. Models that fail overwhelmingly use the lengthy method; models that succeed use the short one. Perhaps, then, collapse is merely a bad choice of algorithm. We took the five models whose failures were logged under cofactor expansion and prompted them to use Gaussian elimination on the very problems they had failed. Even when prompted onto the efficient path, accuracy did not recover: Figure 4 (Panel A) shows the fully collapsed models staying near 0%. Panel B shows why. It traces each run step by step and plots how many remain error-free: Gaussian elimination performs fewer operations but chains them tighter — every step feeds fractions into the next — and by step 2 only 27–67% of runs survive without error; one early wrong fraction poisons everything after it. Fewer operations, deeper dependence, same collapse: what matters is what must be carried, not what must be done (Appendix K).

Experiment 2 — operation count: is it volume? Perhaps models simply break after too many operations of any kind. We built problems where volume grows but the carried state does not: chains of k successive matrix–vector multiplications ($v_1 = A_1 \cdot v_0$, $v_2 = A_2 \cdot v_1$, …, $v_k = A_k \cdot v_{k-1}$, with k distinct 5×5 integer matrices), so k is a dial that raises the operation count while the model only ever carries five integers from one step to the next. Figure 5 (Panel A) plots accuracy against k: the faint background curves are the five non-reasoning models, which fade early; the four frontier curves hold high plateaus far past k = 11 — the length at which one chain's ≈500 operations already exceed the arithmetic of a full 5×5 eigenvalue solve. At that same budget these models hold 77–100% on chains yet score only 1.7–56.7% on 5×5 eigenvalue problems, dropping on chains only at extreme lengths (883 graded chain runs across nine models; Appendix N). Volume alone does not break them.

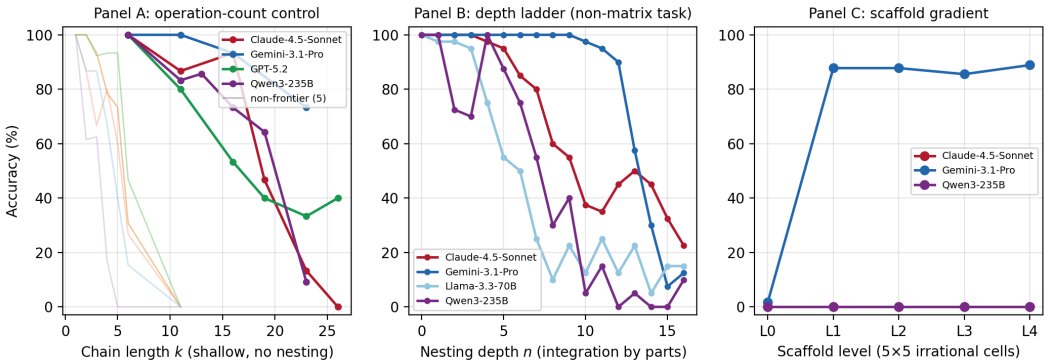

Figure 5: Bottleneck triangulation. Panel A: accuracy versus chain length k on chained matrix–vector multiplication (volume grows, carried state fixed); frontier models hold plateaus rather than degrading smoothly. Panel B: accuracy versus nesting depth n on integration-by-parts problems (2,720 calls); the depth collapse reappears outside linear algebra. Panel C: scaffold gradient on 5×5 irrational eigenvalue cells — Gemini-3.1-Pro climbs from 1.7% (L0) to 86–89% (L1–L4) while Claude-4.5-Sonnet and Qwen3-235B remain at 0% at every level.

Experiment 3 — depth in another domain: is it linear algebra? If carried state is the real limit, the collapse should follow depth into any domain. Integration by parts ($\int u\,dv = uv - \int v\,du$) gives a clean ladder: an integral of the form $\int x^n \sin(bx)\,dx$ requires exactly n+1 applications of the rule, each feeding its result into the next (n is the nesting depth — the level of recursion) — as n increases, the recursion deepens, with no matrices anywhere, reaching the same level of demand the 5×5 eigenvalue task imposes. Each depth level has 20 problems, each run twice (40 calls per model per level); in total, 340 integrals and 2,720 graded calls across four models. The same threshold collapse appears (Figure 5, Panel B): each model survives to its own depth, in the order Gemini-3.1-Pro ($n \approx 13$–14) > Qwen3-235B ($n \approx 11$–13) > Claude-4.5-Sonnet ($n \approx 9$) > Llama-3.3-70B ($n \approx 5$) — exactly the order these same models finish in on the benchmark's eigenvalue tasks, and on determinants too (Appendix O) — and the same error taxonomy classifies the failures with no new categories needed. The collapse repeats in a second domain: the limit travels with depth, not with the subject.

Experiment 4 — scaffolding: do instructions fix it? If the limit were planning or knowledge, telling the model exactly what to do should repair it. We used chain-of-thought prompting that spells out the exact path — which procedure to follow and each step to perform, at five levels of increasing detail, from unaided (L0) to fully externalised stage-by-stage execution (L4) — 4,320 graded calls in total. The outcome is model-dependent (Figure 5, Panel C): Gemini-3.1-Pro — already the strongest model on these tasks, with the highest 5×5 eigenvalue score in the benchmark (56.7%) and 100% on 5×5 determinants — climbs to 86–89% on the irrational cells from L1 onward, while Claude-4.5-Sonnet and Qwen3-235B remain at exactly 0% at every level — every step spelled out, and still nothing (Appendix P). Scaffolding helped only the model that could already do the mathematics; for the models that fail, no prompt supplies the ability to carry the chain. This heterogeneity is scope-defining: any claim that scaffolding recovers deep execution must be indexed by model.

The rivals are now each excluded by design: not knowledge (models state the algorithms they fail to run), not method choice (Experiment 1), not operation volume (Experiment 2), not something peculiar to linear algebra (Experiment 3), not missing instructions (Experiment 4). What remains standing is the depth of sequentially dependent state a model must carry through its own generated context.

7 Discussion

Taken together, the four experiments vary everything a skeptic might blame — the algorithm, the amount of arithmetic, the mathematical domain, the detail of the instructions — and only one variable moves the outcome: how much dependent intermediate state the model must carry from step to step. Raise that, and every model collapses at its own depth; raise anything else while keeping carried state small, and models are fine. Because each model's collapse depth is the same across

eigenvalues, chains, and integrals, the limit behaves like a fixed budget built into the model — not something the task, the subject, or the prompt can change. Ten models spanning every tier and training paradigm each fail with their own signature (Appendix L), yet all hit the same wall: the limit is architectural, not vendor-specific.

This has direct consequences for evaluation methodology. A benchmark evaluating determinants at a single matrix size would classify Llama-3.3-70B, GPT-4o, and Claude-4.5-Sonnet as capable — all achieve $\geq$98% at 3×3 — while concealing collapse to $\leq$6.3% at 5×5. The eigenvalue task sharpens the lesson: two problems of identical size can differ only in whether the answer is guessable, and a benchmark that does not control spectrum type will report shortcut search as execution — "got the right answer" and "did the computation" are different claims. Diagnostic instruments should locate the dial that causes failure, not only rank models; the released generators keep every finding testable on fresh, statistically matched problem sets, robust to training-data contamination.

The failure mode itself carries a deployment warning: models at their depth limit do not report difficulty. At 3×3 they fabricate answers; by 5×5 they mostly abandon — and the fabrications are constraint-consistent: eigenvalue answers built to match the trace and determinant, passing exactly the checks an expert would apply first. The failure is quietest where verification is hardest.

Several limitations bound these claims. First, the depth account is tested outside linear algebra exactly once: the integration-by-parts ladder reproduces the same collapse in the same model order, but chemistry, physics, and statistics remain untested. Second, the instrument covers dense, square, integer-entry matrices with real, distinct spectra at sizes 3×3–5×5, graded by exact match; sparse, structured, or complex-spectrum problems could move the thresholds, and numerical robustness is not a failure surface here. Third, the account is behavioral, not mechanistic: the working-memory limit is inferred from failure patterns and stated as a falsifiable hypothesis. Fourth, models with hidden chains of thought were excluded from the four probe experiments — a hidden scratchpad is an uncontrolled depth aid — so those conclusions speak for visible-trace models. Fifth, error labels come from a three-stage judge pipeline at 91.2% agreement with human review (93.0% at 5×5, where most failures lie); fine-grained subtype percentages inherit the remaining disagreement. Finally, tool availability and API-level constraints mean some model behaviours cannot be definitively attributed to specific implementation choices without access to inference infrastructure.

Three directions follow. Constraint-aware confabulation invites cross-domain tests: whether models fabricate constraint-consistent answers under computational overload in chemistry, physics, or statistics is directly checkable, with immediate evaluation-methodology implications. The working-memory account invites mechanistic study — for example, activation patching at the steps where carried state grows. And the scaffold result implies that multi-step reasoning curricula should be evaluated per model on execution robustness under externalised state, not just accuracy at fixed difficulty.

8 Conclusion

LinAlg-Bench recasts the evaluation of LLM linear-algebra computation as a controlled measurement problem: a 1,194-problem factorial instrument with exact verification, forensic error classification, and regenerable problem sets. The demonstration study yields a consistent picture. Failure is structurally constrained — determined by algorithm family, dimension, and, most sharply, by the structure of the answer itself: integer spectra are solved at 50.7% against 12.1% for irrational spectra, while matrix symmetry is null. Determinant performance bifurcates at 5×5 rather than degrading uniformly. Three designed probes triangulate the bottleneck to sequential execution depth rather than operation volume or method selection, and a five-level scaffold shows the limit is model-dependent: one model recovers to 86–89% under externalised state while others remain at exactly 0%. When computation is abandoned, fabrication is constraint-aware — 45.4% of parsed fabricated spectra match the trace but only 8.5% match the determinant — and more structured in stronger models. These are falsifiable claims that follow directly from the data and invite verification by replication on freshly generated problem sets. All data, model outputs, error labels, generators, and the judge pipeline are publicly released.

Data Availability. All benchmark problems, model outputs, error annotations, and results are publicly available at `https://anonymous-hf.com/a/may6smri97a3/`. Code

and evaluation scripts are available at `https://anonymous.4open.science/r/LinAlg-LLM-Benchmarking-Extended-57C2/`.

REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaïd Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International Conference on Machine Learning*, 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, 2021.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*, 2020.

Nayoung Lee, Kartik Sreenivasan, Jason D. Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers. *arXiv preprint arXiv:2307.03381*, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. MathBench: Evaluating the theory and application proficiency of LLMs with a hierarchical mathematics benchmark. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6884–6915, 2024.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, 2020.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35, 2022.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, et al. SymPy: Symbolic computing in Python. *PeerJ Computer Science*, 3:e103, 2017.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Chang, Tom Kwiatkowski, Chitta Baral Shiue, Sean Welleck, Chitta Baral, Yejin Choi, et al. Lila: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Investigating the limitations of transformers with simple arithmetic tasks. *arXiv preprint arXiv:2102.13019*, 2021.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42, 2019. doi: 10.18653/v1/P19-3007.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A BENCHMARK INFERENCE PROMPTS

All models were evaluated zero-shot at temperature 0. The prompt consists of a system prompt and a user prompt, both fixed per subcategory. A base zero-shot chain-of-thought (ZS-CoT) instruction was included to elicit reasoning traces ("Show all computation steps clearly"; "Show your work") but no few-shot examples, method guidance, or algorithm-specific instructions were provided. Tool use was disabled for all models via API configuration.

A.1 SYSTEM PROMPT

The system prompt is shared across all subcategories. Only the answer-type directive (underlined below) varies:

> You are a precise mathematical assistant.
> Show all computation steps clearly.
> Always put your final [answer-type] answer inside `\boxed{}`.

Table A.1: Answer-type directive by subcategory.

Subcategory	Answer-type word	Tasks
`det`, `trace`	numerical	Determinant, Trace
`eig`	answers	Eigenvalues
`rank`, `null`	integer	Rank, Nullity
`mult`, `pow`, `trans`	matrix	Mult., Power, Transpose
`vec`	vector	Matrix-Vector

A.2 USER PROMPT TEMPLATE

The user prompt template is identical across all nine subcategories:

> Solve this linear algebra problem.
> Show your work and give the final answer.
> {`problem_text`}
> {`problem_matrix`}

where {`problem_text`} is the task instruction (e.g. "Compute the determinant of matrix A") and {`problem_matrix`} is the matrix in one of three input formats described in Table A.2.

A.3 INPUT FORMAT EXAMPLES

The same 3×3 determinant problem is shown below in all three input formats used in the benchmark. Format sensitivity was evaluated at 4×4 and 5×5 only; the 3×3 dimension is shown here for brevity.

A.4 MODEL DETAILS AND SELECTION RATIONALE

Table A.3 reports the exact model identifiers and access configuration used for all benchmark inference runs, sourced directly from the production inference code. All models were evaluated zero-shot at temperature 0 with tool use and external code execution disabled. Token budgets were set to 8,192 for most subcategories and 16,384 for DeepSeek-V3, GPT-5.2, and Qwen3-235B. Responses reaching the token ceiling without producing a final answer were classified as GENERATION_TRUNCATION, strictly separating infrastructure limits from computational collapse.

Reproducibility window To establish a reproducible temporal baseline, all inference across all 10 models was conducted within a strict window between March 2026 and August 2026. Six models were accessed via the OpenRouter routing layer: DeepSeek-V3, Qwen3-235B and Qwen2.5-72B,

Table A.2: Three input format representations of the same determinant problem. All three encode identical numerical content; only the notation differs.

Format	{problem_matrix} value	Description
LaTeX	$A = \begin{bmatrix} 2 & -1 & 3 \\ 0 & 4 & -2 \\ -1 & 1 & 5 \end{bmatrix}$	Standard LaTeX bmatrix environment. Provides explicit algebraic notation.
Tabular	| 2 -1 3 | | 0 4 -2 | | -1 1 5 |	Pipe-delimited human-readable grid. Visual row-column structure without mathematical notation.
List	A = [[2, -1, 3], [0, 4, -2], [-1, 1, 5]]	Nested Python-style list. No explicit visual alignment.

Mistral-Large, Claude-4.5-Sonnet, and Llama-3.3-70B. OpenRouter does not guarantee snapshot pinning; the date window bounds this uncertainty. Four models were accessed via direct provider APIs: OpenAI-o1, GPT-4o, and GPT-5.2 via the OpenAI API, and Gemini-3.1-Pro via the Google AI Studio GenAI SDK.

Table A.3: Model identifiers and inference configuration.

Paper Name	Model ID	Access	Provider	Architecture
OpenAI-o1	`o1`	Direct	OpenAI	Dense, reasoning-optimised
Gemini-3.1-Pro	`gemini-3.1-pro-preview`	Direct	Google AI Studio	Dense, frontier (main benchmark; also probe experiments, Appendices N–P)
GPT-5.2	`gpt-5.2`	Direct	OpenAI	Dense, frontier
GPT-4o	`gpt-4o`	Direct	OpenAI	Dense, instruction-tuned
DeepSeek-V3	`deepseek/deepseek-v3.2`	OpenRouter	DeepSeek	MoE, instruction-tuned
Qwen3-235B	`qwen/qwen3-235b-a22b-2507`	OpenRouter	Alibaba	MoE, thinking/non-thinking hybrid
Mistral-Large	`mistralai/mistral-large-2512`	OpenRouter	Mistral AI	MoE, instruction-tuned
Claude-4.5-Sonnet	`anthropic/claude-sonnet-4.5`	OpenRouter	Anthropic	Dense, instruction-tuned
Qwen2.5-72B	`qwen/qwen-2.5-72b-instruct`	OpenRouter	Alibaba	Dense, instruction-tuned
Llama-3.3-70B	`meta-llama/llama-3.3-70b-instruct`	OpenRouter	Meta	Dense, instruction-tuned

Note. All model IDs copied verbatim from production inference code. Gemini-3.1-Pro (API ID gemini-3.1-pro-preview) is used for both the main benchmark and the probe experiments (Appendices N–P). Paper name DeepSeek-V3 refers to the base/chat MoE model deepseek-v3.2; this is not the DeepSeek-R1 reasoning model.

A.5 Model Selection Rationale

The 10 models were selected to cover five independent axes of variation, enabling the benchmark to test whether failure patterns are specific to particular training regimes, architectures, or providers.

A.6 Excluded Models

Qwen-2.5-Math-72B was accessed via a HuggingFace Space inference endpoint—not an API-controlled environment—and is excluded as results are not reproducible under the same inference protocol. Claude-3.5-Sonnet was registered as a standby model and replaced by Claude-4.5-Sonnet before benchmark runs began; no Claude-3.5-Sonnet results are included in the benchmark.

B Benchmark Task Descriptions

This appendix describes all nine subcategories in the LinAlg-Bench benchmark. For each task we show the exact instruction seen by the model, the algorithmic steps required for a correct solution, and how computational demand scales across the three matrix dimensions. Tasks are grouped by cognitive level.

Table A.4: Selection rationale. Models were selected to cover five independent axes of variation: training paradigm, architecture, provider, weight availability, and scale. No model was selected on the basis of expected performance outcome. Selection was finalised before any benchmark runs were conducted.

Axis	Coverage
Training paradigm	Reasoning-optimised (OpenAI-o1); thinking-mode hybrid (Qwen3-235B); standard RLHF dense (Llama-3.3-70B, Qwen2.5-72B, GPT-4o, Claude-4.5-Sonnet); balanced frontier (GPT-5.2, Gemini-3.1-Pro); MoE instruction-tuned (DeepSeek-V3, Mistral-Large).
Architecture	Dense: GPT-4o, GPT-5.2, OpenAI-o1, Claude-4.5-Sonnet, Qwen2.5-72B, Llama-3.3-70B, Gemini-3.1-Pro. MoE: Qwen3-235B (22B active / 235B total), Mistral-Large, DeepSeek-V3.
Provider	OpenAI (3); Google (1); Anthropic (1); Meta (1); Alibaba (2); Mistral AI (1); DeepSeek (1).
Weight availability	Closed-weight: GPT-4o, GPT-5.2, OpenAI-o1, Gemini-3.1-Pro, Claude-4.5-Sonnet. Open-weight: DeepSeek-V3, Qwen3-235B, Qwen2.5-72B, Llama-3.3-70B, Mistral-Large.
Scale	Active parameter range from 70B (Llama-3.3-70B, Qwen2.5-72B) to 235B total / 22B active (Qwen3-235B). Closed-weight models not directly comparable by parameter count.

1. Trace · Cognitive level: Reading **Instruction:** Compute the trace of matrix A. Put your final numerical answer inside `\boxed{}`.

Required steps: Sum the main diagonal entries $a_{11} + a_{22} + \cdots + a_{nn}$.

Expected output: Single integer scalar.

	3×3	**4×4**	**5×5**
Operations	2 additions	3 additions	4 additions

Primary failure at 5×5: *None — trace is format-invariant and dimension-invariant. Serves as ceiling baseline.*

2. Transpose · Cognitive level: Reading **Instruction:** Compute the transpose of matrix A. Put your final matrix answer inside `\boxed{}`.

Required steps: Reflect all entries across the main diagonal: entry (i, j) moves to (j, i).

Expected output: $n \times n$ matrix.

	3×3	**4×4**	**5×5**
Swaps required	3 off-diagonal pairs	6 off-diagonal pairs	10 off-diagonal pairs

Primary failure at 5×5: *None — purely positional, no arithmetic. Serves as ceiling baseline.*

3. Matrix-Vector Multiplication · Cognitive level: Arithmetic **Instruction:** Compute the matrix-vector product Ax. Put your final vector answer inside `\boxed{}`.

Required steps: For each row of A, compute the dot product with vector x: sum of $a_{ij} \cdot x_j$ across j.

Expected output: Column vector of length n.

	3×3	**4×4**	**5×5**
Dot products	3 (one per row)	4 (one per row)	5 (one per row)
Multiplications	9 total	16 total	25 total
Error exposure	Low	Moderate	Moderate

Primary failure at 5×5: *Arithmetic slips in the final summation — dot-product terms are computed correctly, but their sum is wrong (4 failures of 200, all Arithmetic-tagged).*

4. Matrix Multiplication · Cognitive level: Arithmetic **Instruction:** Compute the matrix product AB, where A and B are both $n \times n$ matrices. Put your final matrix answer inside $\texttt{\textbackslash boxed}\{\}$.

Required steps: For each output entry (i, j), compute dot product of row i of A with column j of B.

Expected output: $n \times n$ matrix.

	3×3	**4×4**	**5×5**
Output entries	9	16	25
Dot products	9 (length 3 each)	16 (length 4 each)	25 (length 5 each)
Multiplications	27 total	64 total	125 total

Primary failure at 5×5: *Arithmetic summation slips (12 of 15 failures) and miscopied input entries (3 of 15); errors are isolated per entry rather than compounding.*

5. Matrix Power · Cognitive level: Arithmetic **Instruction:** Compute A^k. Put your final matrix answer inside $\texttt{\textbackslash boxed}\{\}$.

Required steps: Compute $A^2 = A \cdot A$, then $A^3 = A^2 \cdot A$, repeating $k - 1$ multiplications. Each multiplication requires n^2 dot products of length n.

Expected output: $n \times n$ matrix.

	3×3	**4×4**	**5×5**
Multiplications	$k-1$ full matrix multiplications	$k-1$ full matrix multiplications	$k-1$ full matrix multiplications
Error propagation	Errors from step t affect all $t+1, t+2, \ldots$	Errors from step t affect all $t+1, t+2, \ldots$	Errors compound most severely at 5×5 (125 ops per step)

Primary failure at 5×5: *Input_Transcription — the model miscopies a matrix entry inside a dot product (11 of 12 failures); no sign errors observed at this size.*

6. Rank · Cognitive level: Sequential **Instruction:** Compute the rank of matrix A. Put your final integer answer inside $\texttt{\textbackslash boxed}\{\}$.

Required steps: Perform row reduction (Gaussian elimination) to echelon form. Count non-zero pivot rows.

Expected output: Integer in $\{0, 1, \ldots, n\}$.

	3×3	**4×4**	**5×5**
Row operations	Up to 3 elimination steps	Up to 6 elimination steps	Up to 10 elimination steps
Pivot decisions	(is row zero?)		
Arithmetic ops	$\sim$ 9 multiplications + subtractions	$\sim$ 24 multiplications + subtractions	$\sim$ 50 multiplications + subtractions

Primary failure at 5×5: *Arithmetic slips during row reduction (25 of 35 failures), plus row operations applied in the wrong direction (Operation_Direction, 7 cases).*

7. Nullity · Cognitive level: Sequential **Instruction:** Compute the nullity of matrix A. Put your final integer answer inside $\texttt{\textbackslash boxed}\{\}$.

Required steps: Compute rank via row reduction, then apply rank-nullity theorem: nullity $= n -$ rank.

Expected output: Integer in $\{0, 1, \ldots, n\}$.

	3×3	**4×4**	**5×5**
Depends on	Rank computation (same steps)	Rank computation (same steps)	Rank computation (same steps)
Extra step	One subtraction: $3 -$ rank	One subtraction: $4 -$ rank	One subtraction: $5 -$ rank

Primary failure at 5×5: *Mirrors rank — arithmetic slips during row reduction (26 of 34 failures); the failure rate (17.0%) tracks rank's (17.5%).*

8. Determinant · Cognitive level: Recursive **Instruction:** Compute the determinant of matrix A. Put your final numerical answer inside $\texttt{\textbackslash boxed}\{\}$.

Required steps: Cofactor expansion: $\det(A) = \sum_j a_{1j} \cdot (-1)^{1+j} \cdot M_{1j}$ where M_{1j} is the $(n-1) \times (n-1)$ minor. Alternatively: Gaussian elimination to upper triangular form, then multiply diagonal.

Expected output: Single integer scalar.

	3×3	**4×4**	**5×5**
Cofactor terms	sub-minors	$4 \times 3 = 12$ sub-minors	$5 \times 4 \times 3 = 60$ sub-minors
Parity checks	$(+ - +)$	(two levels of $+ - +-$)	$+$ (three nested levels)
Gaussian steps	pivot eliminations	pivot eliminations	pivot eliminations
Method advantage	Equal — both manageable	Gaussian $\sim$ 3× fewer ops	Gaussian $\sim$ 6× fewer ops

Primary failure at 5×5: *Alternating_Drift (cofactor models lose ± 1 checkerboard after 3–4 terms); Gaussian-dominant models maintain high accuracy. Strategy is the primary predictor of success at 5×5.*

9. Eigenvalues · Cognitive level: Compositional **Instruction:** Find all eigenvalues of matrix A. Put your final answers inside $\texttt{\textbackslash boxed}\{\}$.

Required steps: Stage 1: Form $A - \lambda I$. Stage 2: Expand $\det(A - \lambda I)$ to obtain characteristic polynomial of degree n. Stage 3: Find all roots of the polynomial.

Expected output: Set of n real eigenvalues (all benchmark spectra are real by construction — integer or irrational; see the stratification design at the end of this appendix).

	3×3	**4×4**	**5×5**
Characteristic poly	Degree 3 (cubic)	Degree 4 (quartic)	Degree 5 (quintic)
Cofactor terms	(each contains λ)	(each a polynomial in λ)	(each a polynomial in λ)
Root finding	Factorable by hand	Difficult, may require numerical methods	Integer-spectrum problems factor via the rational root theorem; irrational-spectrum problems have no rational roots and are graded numerically as unordered sets (see grading tolerance below)
Failure surfaces	(det expansion + root finding)	(compounded)	(fully compounded — det load × polynomial load)

Primary failure at 5×5: *Complete_Collapse dominates at 5×5 (85.4% of hallucination-tagged failures; 784 of 918 cases): models correctly identify the task, then abandon computation with meta-statements citing complexity, fabricating eigenvalues via simulated tool calls.* ***Gemini-3.1-Pro*** *and* ***OpenAI-o1*** *demonstrate the task remains tractable at this size (56.7% and 32.5% on 5×5 eigenvalue problems, respectively).*

Grading tolerance: Eigenvalue answers are graded as unordered sets against the reference spectrum, with a per-value tolerance of 0.011. The bound is empirical, taken from the released grading

data (model_outputs.parquet): of 1,070 accepted eigenvalue answers, 998 state numeric values — 93% deviate from the reference by at most 0.004, and 97% by at most 0.011. The judge additionally accepted a small number of low-precision rounded answers (deviations up to ≈ 0.09) and 72 answers that give the characteristic polynomial or root intervals instead of numeric values (judge-leniency cases; Appendix E). The grading protocol codifies 0.011 as an explicit rule.

Spectra are well separated by construction: the minimum eigenvalue gap across the set is 0.32, comfortably above the design floor of 0.3 and far above the grading tolerance, so set matching is unambiguous.

B.1 Eigenvalue Stratification: Symmetry × Spectrum Factorial Design

Two natural objections apply to any eigenvalue benchmark: performance may depend on matrix structure such as symmetry, and failure at 5×5 might simply reflect spectra that cannot be written in closed form rather than a computational limit. The eigenvalue set is therefore built as a 2×2 factorial design — **symmetry** (symmetric vs. non-symmetric) × **spectrum type** (integer vs. irrational eigenvalues) — fully crossed with matrix dimension, so both objections are measured rather than assumed.

The symmetry axis (Figure B.1). A symmetric matrix ($A = A^T$) mirrors across its main diagonal: the entry in row i, column j always equals the entry in row j, column i. This structure guarantees that every eigenvalue is real — the structurally "nice" case. A non-symmetric matrix breaks the mirror and carries no such guarantee, so every non-symmetric instance in the benchmark is explicitly constructed to have an all-real spectrum. Both arms therefore share the same answer space, and any accuracy difference between them is attributable to structure alone.

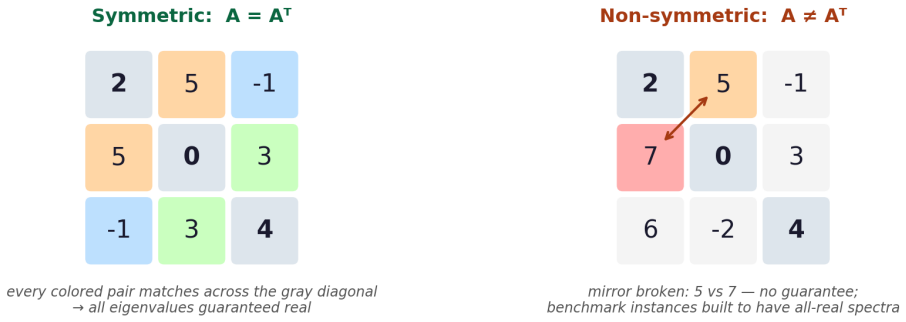

Figure B.1: **The symmetry axis.** Left: a symmetric matrix — each colored pair matches across the gray diagonal, which guarantees all-real eigenvalues. Right: a non-symmetric matrix — the mirror is broken (5 vs. 7), so real eigenvalues are not guaranteed; benchmark instances are built to have all-real spectra so both arms share the same answer space.

The spectrum axis (Figure B.2). An eigenvalue can lie anywhere on the real number line, negative or positive. In *integer-spectrum* problems every eigenvalue is an exact integer — for example -3, 2, and 7 — so it sits exactly on an integer point of the number line, and a model can confirm its answer with clean whole-number arithmetic (the values must, for instance, sum to the trace). In *irrational-spectrum* problems every eigenvalue is provably irrational, so it falls strictly between two integers — for example $1 - \sqrt{5} = -1.236\ldots$ — and never on a clean point: no simple arithmetic check exists, and numerical precision must survive the entire computation. This axis directly tests whether "the exact answer cannot be expressed simply" explains the 5×5 collapse.

What the factorial separates. If models fail equally in all four cells, failure is driven by dimension — the depth thesis of this paper. If failure instead concentrates in particular cells, it tracks matrix structure (symmetry) or spectrum type (whether the eigenvalues are integers or irrational). Table B.1 shows the four cells and their expected difficulty ordering; Tables B.2 and B.3 give the composition.

Composition. The eigenvalue set contains 360 problems: 180 symmetric and 180 non-symmetric, with each arm splitting into 90 integer-spectrum and 90 irrational-spectrum problems (Table B.2).

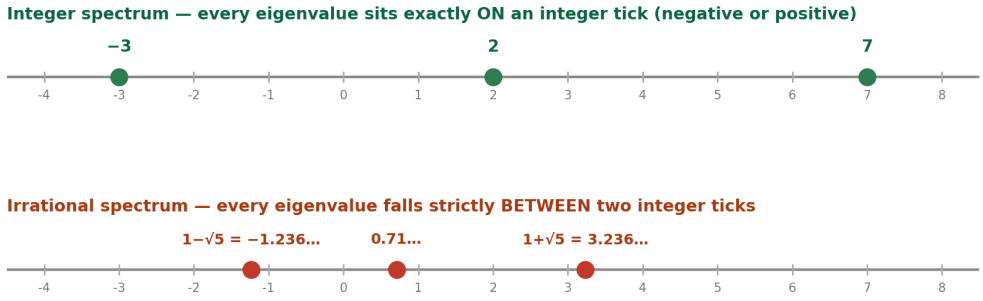

Figure B.2: **The two spectrum types.** Eigenvalues can lie anywhere on the line, negative or positive. Integer-spectrum eigenvalues (top) sit exactly on integer ticks — here -3, 2, and 7 — so clean arithmetic checks apply (for example, the values must sum to the trace). Irrational-spectrum eigenvalues (bottom) always fall strictly between two integer ticks — here $-1.236\ldots$, $0.71\ldots$, and $3.236\ldots$ — so no clean check exists and precision must survive the whole computation.

Table B.1: **The 2×2 factorial.** Green marks the expected easiest cell (nice structure, clean answers); red the hardest (neither). Crossed with three dimensions at 30 problems per cell: 4 cells $\times$ 3 dimensions $\times$ 30 = 360 problems.

	Integer answers	**Irrational answers**
Symmetric	Easiest cell: nice structure + clean answers	Structure nice, answers messy
Non-symmetric	Structure messy, answers clean	Hardest cell: messy structure + messy answers

Every dimension contains exactly 120 problems, 30 per cell, so each dimension splits 60/60 on both axes and accuracy comparisons across dimension, symmetry, or spectrum type carry equal statistical weight (Table B.3).

Table B.2: **Composition of the eigenvalue set.** 360 problems, split evenly on both axes: each cell of the factorial contributes 90 problems. (The eigenvalue task additionally includes 14 problems with a repeated eigenvalue — 7 at 3×3, 7 at 4×4 — which fit neither spectrum cell; they are counted in Tables C.1–C.3 but sit outside the factorial.)

	Integer spectrum	**Irrational spectrum**	**Total**
Symmetric	90	90	180
Non-symmetric	90	90	180
Total	**180**	**180**	**360**

Construction. Every cell is exact by construction. Symmetric-integer matrices are built by exact orthogonal similarity ($d^2 \cdot Q \cdot D \cdot Q^T + m \cdot I$) from a chosen integer spectrum; non-symmetric-integer matrices by unimodular similarity ($P \cdot D \cdot P^{-1}$ with $\det P = \pm 1$), which keeps integer entries and the exact integer spectrum; irrational cells use symmetrization ($A + A^T$) or rejection sampling over integer-entry matrices, accepting only spectra that are all-real with no rational root. All matrices have integer entries bounded by $|12|$, eigenvalues bounded by $|\lambda| \leq 20$, and a minimum eigenvalue gap of 0.3, so every problem is well conditioned for the unordered-set grading at the 0.011 tolerance described above.

C Full Accuracy Tables

Tables C.1–C.3 report complete per-model, per-task accuracy across all three matrix dimensions. All models were evaluated zero-shot at temperature 0. The benchmark comprises 9 tasks across 5 cognitive levels, with 317 (3×3), 442 (4×4), and 435 (5×5) questions (1,194 total). Ground truth

Table B.3: **Per-dimension balance.** Every dimension carries exactly 30 problems per cell — 120 in total, splitting 60 symmetric / 60 non-symmetric and 60 integer / 60 irrational.

Cell	**3×3**	**4×4**	**5×5**
Symmetric, integer	30	30	30
Symmetric, irrational	30	30	30
Non-symmetric, integer	30	30	30
Non-symmetric, irrational	30	30	30
Total	**120**	**120**	**120**

verified using SymPy symbolic computation. Model column order is fixed across all three tables. OpenAI-o1 and GPT-5.2 were run on capability-ordered subsets (258 and 699 of the 1,194 problems, respectively); problems they did not attempt are scored under the capability-ordering declared-pass protocol documented in the note at the end of this appendix. Entries for these two models therefore combine measured and protocol-scored problems.

Table C.1: Complete accuracy table — 3×3 matrices (317 questions, 10 models). Numbers show correct responses out of maximum per category. OpenAI-o1 and GPT-5.2 entries include declared-pass scoring (see note at end of Appendix C).

Level	**Category**	**Max**	**OpenAI-o1**	**Gemini-3.1-Pro**	**DeepSeek-V3**	**GPT-5.2**	**Qwen3-235B**	**Mistral-Large**	**Claude-4.5-Son.**	**Qwen2.5-72B**	**Llama-3.3-70B**	**GPT-4o**
Reading	Trace	20	20	20	20	20	20	20	20	20	20	20
	Transpose	20	20	20	20	20	20	20	20	19	20	20
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*97.5*	*100.0*	*100.0*
Arithmetic	Matrix-Vector	20	20	20	20	20	20	20	20	20	20	20
	Multiplication	20	20	20	20	20	20	20	20	20	20	19
	Matrix Power	20	20	20	20	20	13	20	20	20	19	20
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*88.3*	*100.0*	*100.0*	*100.0*	*98.3*	*98.3*
Sequential	Nullity	20	20	20	20	20	20	20	20	19	14	20
	Rank	20	20	20	20	20	19	20	20	19	19	19
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*97.5*	*100.0*	*100.0*	*95.0*	*82.5*	*97.5*
Deep Recursive	Determinant	50	50	50	50	50	48	50	50	46	50	49
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*96.0*	*100.0*	*100.0*	*92.0*	*100.0*	*98.0*
Built on Det.	Eigenvalues	127	110	107	93	82	73	68	64	37	18	24
Level Accuracy (%)			*86.6*	*84.3*	*73.2*	*64.6*	*57.5*	*53.5*	*50.4*	*29.1*	*14.2*	*18.9*
Total Correct		**317**	**300**	**297**	**283**	**272**	**253**	**258**	**254**	**220**	**200**	**211**
Overall Model Accuracy (%)			***94.6***	***93.7***	***89.3***	***85.8***	***79.8***	***81.4***	***80.1***	***69.4***	***63.1***	***66.6***

Table C.2: Complete accuracy table — 4×4 matrices (442 questions, 10 models). OpenAI-o1 and GPT-5.2 entries include declared-pass scoring (see note at end of Appendix C).

Level	**Category**	**Max**	**OpenAI-o1**	**Gemini-3.1-Pro**	**DeepSeek-V3**	**GPT-5.2**	**Qwen3-235B**	**Mistral-Large**	**Claude-4.5-Son.**	**Qwen2.5-72B**	**Llama-3.3-70B**	**GPT-4o**
Reading	Trace	20	20	20	20	20	20	20	20	20	20	19
	Transpose	20	20	20	20	20	20	20	20	18	19	20
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*95.0*	*97.5*	*97.5*
Arithmetic	Matrix-Vector	20	20	20	20	20	19	20	20	19	20	20
	Multiplication	20	20	20	20	20	19	20	20	19	19	20
	Matrix Power	20	20	20	20	20	18	20	20	17	18	20
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*93.3*	*100.0*	*100.0*	*91.7*	*95.0*	*100.0*
Sequential	Nullity	20	20	20	20	20	20	18	16	17	12	13
	Rank	20	20	20	20	20	19	17	18	13	11	12
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*97.5*	*87.5*	*85.0*	*75.0*	*57.5*	*62.5*
Deep Recursive	Determinant	175	175	175	161	173	167	165	146	46	148	27
Level Accuracy (%)			*100.0*	*100.0*	*92.0*	*98.9*	*95.4*	*94.3*	*83.4*	*26.3*	*84.6*	*15.4*
Built on Det.	Eigenvalues	127	93	103	52	50	54	34	28	7	1	8
Level Accuracy (%)			*73.2*	*81.1*	*40.9*	*39.4*	*42.5*	*26.8*	*22.0*	*5.5*	*0.8*	*6.3*
Total Correct		**442**	**408**	**418**	**353**	**363**	**356**	**334**	**308**	**176**	**268**	**159**
Overall Model Accuracy (%)			***92.3***	***94.6***	***79.9***	***82.1***	***80.5***	***75.6***	***69.7***	***39.8***	***60.6***	***36.0***

Declared-pass protocol: Because of the very high computation cost of running OpenAI-o1 (and, to a lesser extent, GPT-5.2) on the full benchmark, these two models were run on capability-ordered subsets: OpenAI-o1 attempted 258 and GPT-5.2 attempted 699 of the 1,194 problems. A problem a model did not attempt is scored as a declared pass: under the capability ordering it ranks below problems the model demonstrably solved, so it is credited as correct. Entries for these two models in Tables C.1–C.3 therefore combine measured and declared results; all other models attempted every problem.

Table C.3: Complete accuracy table — 5×5 matrices (435 questions, 10 models). OpenAI-o1 and GPT-5.2 entries include declared-pass scoring (see note at end of Appendix C).

Level	Category	Max	OpenAI-o1	Gemini-3.1-Pro	DeepSeek-V3	GPT-5.2	Qwen3-235B	Mistral-Large	Claude-4.5-Son.	Qwen2.5-72B	Llama-3.3-70B	GPT-4o
Reading	Trace	20	20	20	20	20	20	20	20	20	20	20
	Transpose	20	20	20	20	20	20	20	20	19	20	20
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*100.0*	*97.5*	*100.0*	*100.0*
Arithmetic	Matrix-Vector	20	20	20	20	20	20	20	20	16	20	20
	Multiplication	20	20	20	20	20	15	20	20	12	19	19
	Matrix Power	20	20	20	20	20	20	19	20	19	11	19
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*100.0*	*91.7*	*98.3*	*100.0*	*78.3*	*83.3*	*96.7*
Sequential	Nullity	20	20	20	20	19	20	14	15	11	13	14
	Rank	20	20	20	20	19	20	16	16	10	11	13
Level Accuracy (%)			*100.0*	*100.0*	*100.0*	*95.0*	*100.0*	*75.0*	*77.5*	*52.5*	*60.0*	*67.5*
Deep Recursive	Determinant	175	175	175	125	159	164	67	11	0	9	0
Level Accuracy (%)			*100.0*	*100.0*	*71.4*	*90.9*	*93.7*	*38.3*	*6.3*	*0.0*	*5.1*	*0.0*
Built on Det.	Eigenvalues	120	39	68	7	4	12	0	2	0	0	0
Level Accuracy (%)			*32.5*	*56.7*	*5.8*	*3.3*	*10.0*	*0.0*	*1.7*	*0.0*	*0.0*	*0.0*
Total Correct		**435**	**354**	**383**	**272**	**301**	**311**	**196**	**144**	**107**	**123**	**125**
Overall Model Accuracy (%)			***81.4***	***88.0***	***62.5***	***69.2***	***71.5***	***45.1***	***33.1***	***24.6***	***28.3***	***28.7***

D Format Sensitivity Analysis

Format sensitivity was evaluated across three input representations — LaTeX (standard bmatrix notation), List (nested Python-style list), and Tabular (pipe-delimited grid) — at 4×4 and 5×5 matrix dimensions only; the 3×3 dimension was excluded as pilot evaluation confirmed near-universal ceiling performance across all formats at that scale, making format the least interesting variable to study. $n = 200$ (4×4) / $n = 220$ (5×5) questions per model per format across 9 subcategories.

At 4×4 format sensitivity is negligible across all tiers. At 5×5 it is strictly complexity-gated:

- **Tier 1 models** remain format-robust. Gemini-3.1-Pro scores 50/50 (LaTeX) and 49/50 (List, Tabular) on 5×5 determinants — ceiling-level under every format.
- **Tier 2 models** show divergent model-specific sensitivity. GPT-5.2 incurs a LaTeX penalty on 5×5 determinants (24/50 LaTeX vs 45–46/50 List/Tabular); Mistral shows the opposite (27/50 LaTeX vs 13–16/50 List/Tabular). These are opposing effects, not systematic format bias.
- **Tier 3 models** collapse completely regardless of format — on 5×5 determinants GPT-4o scores at most 1/50 and Llama at most 2/50 under any of the three formats. The dimensional boundary, not input notation, is the binding constraint.

The key implication: format sensitivity is a **second-order effect** that manifests only when working memory is partially saturated (Tier 2 at 5×5). When sufficient (Tier 1) or fully saturated (Tier 3), format is irrelevant.

Table D.1: Format sensitivity raw correct counts — 4×4 matrices (200 questions per model per format, 9 subcategories). Three sub-columns per model: LaTeX, List, Tabular. All counts from a fully enumerated evaluation. † OpenAI-o1 excluded.

Category	Max	Gemini			Qwen 235B			DeepSeek			GPT-5.2			Mistral			Claude 4.5		
		LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.
Determinant	30	30	30	28	29	29	28	26	27	27	23	30	30	25	29	29	24	21	21
Eigenvalues	30	15	25	24	8	18	20	7	22	23	9	24	23	5	20	21	5	20	20
Multiplication	20	20	5	5	19	5	5	20	6	6	20	11	10	20	5	5	20	5	5
Rank	20	20	20	20	19	20	20	20	20	20	20	20	20	18	16	16	20	18	17
Trace	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
Matrix-Vector	20	20	20	20	19	20	19	20	20	20	20	20	20	20	20	20	20	20	20
Nullity	20	20	20	20	20	20	20	20	19	20	20	20	20	14	17	19	16	14	16
Transpose	20	20	20	20	20	20	20	20	20	20	20	20	19	20	20	20	20	20	20
Matrix Power	20	20	15	15	18	9	10	20	10	10	20	13	14	20	9	9	20	20	20
TOTAL	**200**	**185**	**175**	**172**	**172**	**161**	**162**	**173**	**164**	**166**	**172**	**178**	**176**	**162**	**156**	**159**	**165**	**158**	**159**

† *OpenAI-o1 excluded. Format sensitivity covers 9 models × 3 formats × 200 (4 × 4) or 220 (5 × 5) questions = 5,400 and 5,940 evaluations respectively.*

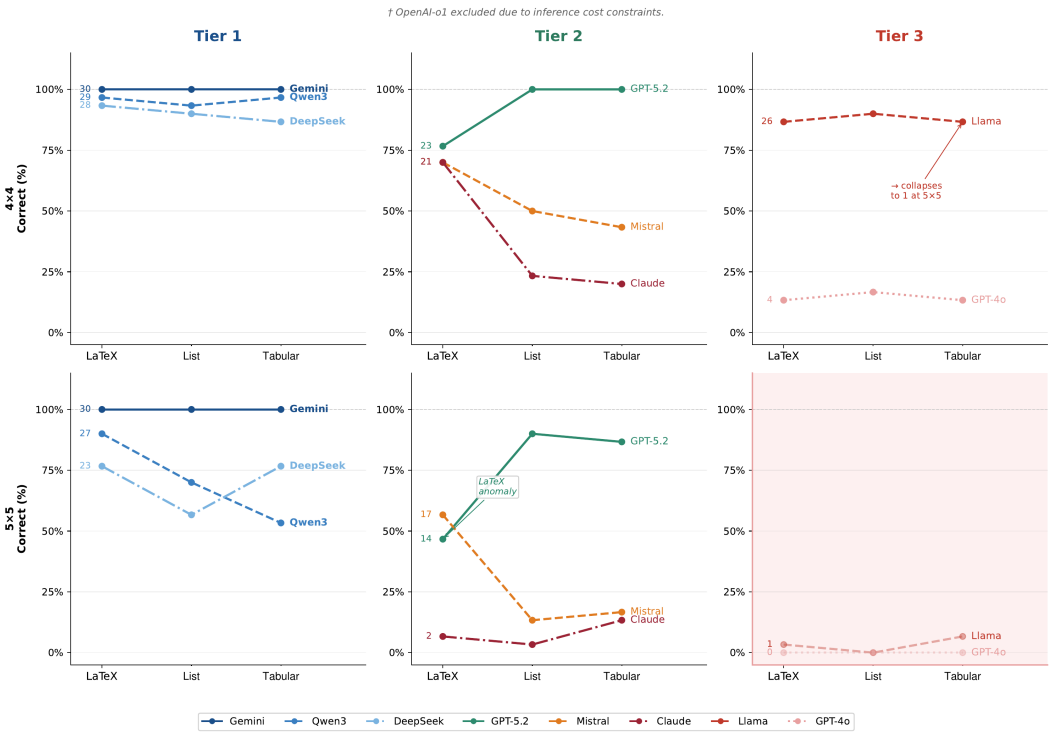

Figure D.1: Format sensitivity across matrix dimensions and model tiers (determinant subcategory, $n = 30$ (4×4) / 50 (5×5) per model per format). At 4×4 (top row) format variance is minimal. At 5×5 (bottom row) format sensitivity emerges but is strictly complexity-gated. Red-shaded panel indicates complete Tier 3 collapse. † OpenAI-o1 excluded due to inference cost constraints.

Table D.2: Format sensitivity raw correct counts — 4×4 matrices, Tier 3 models.

Category	Max	GPT-4o			Llama			Qwen2.5 72B		
		LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.
Determinant	30	5	5	4	26	27	26	7	6	6
Eigenvalues	30	5	18	18	4	19	19	5	19	19
Multiplication	20	20	5	5	19	4	4	19	5	5
Rank	20	11	11	9	7	10	10	12	9	7
Trace	20	20	20	19	20	20	20	20	20	20
Matrix-Vector	20	20	20	19	20	19	19	19	20	20
Nullity	20	12	15	12	12	11	10	18	17	14
Transpose	20	20	20	20	19	20	20	15	19	20
Matrix Power	20	20	15	15	18	7	8	19	19	19
TOTAL	**200**	**133**	**129**	**121**	**145**	**137**	**136**	**134**	**134**	**130**

Table D.3: Format sensitivity raw correct counts — 5×5 matrices (220 questions per model per format, 9 subcategories). Eigenvalue row shows near-universal zero across all models and formats, confirming eigenvalue collapse is format-invariant.

Category	Max	Gemini			Qwen 235B			DeepSeek			GPT-5.2			Mistral			Claude 4.5		
		LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.
Determinant	50	50	49	49	45	35	34	41	30	37	24	46	45	27	13	16	4	4	2
Eigenvalues	30	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
Multiplication	20	20	20	20	15	14	11	20	20	19	20	19	19	20	19	20	20	20	19
Rank	20	20	20	20	20	20	20	20	18	20	19	19	20	15	13	15	16	12	10
Trace	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	18	20	20	20
Matrix-Vector	20	20	20	20	20	17	19	20	20	20	20	20	20	20	20	20	20	13	13
Nullity	20	20	20	20	20	20	18	20	20	20	19	15	17	14	9	11	15	12	13
Transpose	20	20	20	20	20	18	19	20	20	20	20	20	20	20	19	18	20	18	16
Matrix Power	20	20	20	20	20	14	17	20	16	17	20	18	20	19	17	17	20	20	20
TOTAL	**220**	**190**	**189**	**189**	**180**	**158**	**158**	**184**	**164**	**173**	**162**	**177**	**181**	**155**	**130**	**135**	**135**	**119**	**113**

Table D.4: Format sensitivity raw correct counts — 5×5 matrices, Tier 3 models.

Category	Max	GPT-4o			Llama			Qwen2.5 72B		
		LaTeX	List	Tab.	LaTeX	List	Tab.	LaTeX	List	Tab.
Determinant	50	0	0	1	2	0	2	2	0	1
Eigenvalues	30	0	0	0	0	0	0	0	0	0
Multiplication	20	19	19	19	19	13	16	12	13	12
Rank	20	13	10	8	11	9	12	10	9	7
Trace	20	20	19	19	20	20	20	20	20	20
Matrix-Vector	20	20	13	12	20	13	13	16	10	10
Nullity	20	14	7	13	13	8	9	11	12	10
Transpose	20	20	20	20	20	17	18	19	19	20
Matrix Power	20	19	15	18	11	6	5	19	16	16
TOTAL	**220**	**125**	**103**	**110**	**116**	**86**	**95**	**109**	**99**	**96**

† *OpenAI-o1 excluded. Format sensitivity covers 9 models × 3 formats × 200 (4×4) or 220 (5×5) questions = 5,400 and 5,940 evaluations respectively.*

E JUDGE PIPELINE VALIDATION

The forensic classification pipeline operates in two independent stages. Both stages use **Gemini-3.1-Pro** at **temperature 0.0** for deterministic output.

E.1 BUILD JUDGE (STAGE 1 — FIRST-PASS CLASSIFICATION)

The Build Judge acts as a forensic auditor. For each failed response:

- For all subcategories except eigenvalues, **the Build Judge** independently computes the correct solution from scratch before reading the model response. Gemini-3.1-Pro achieves perfect accuracy on all non-eigenvalue subcategories across all three matrix dimensions, confirming it as a reliable computation oracle for the judge role. All judge outputs are additionally cross-validated against the SymPy-certified ground truth as a consistency check. For eigenvalues, the Build Judge receives the SymPy-certified answer directly as its reference and focuses solely on tracing where and how the model's reasoning diverged from the correct solution path.
- Traces the model response **step-by-step** to identify the **first erroneous value**, not just the final wrong answer.
- Applies the **Magnitude Rule** as the primary boundary enforcer between `sign_error` and `arithmetic`: if $|\text{wrong}| = |\text{correct}|$ (element-wise for matrices/vectors) $\rightarrow$ `sign_error`; if magnitudes differ $\rightarrow$ `arithmetic`.
- Records **Solution_Strategy** (e.g., cofactor expansion vs. Gaussian elimination) alongside the error tag.
- Outputs **First_Error_Step** and **First_Error_Description** for full forensic traceability.
- Requires a **confidence level** (`high` / `medium` / `low`) for every classification.

E.2 VALIDATE JUDGE (STAGE 2 — INDEPENDENT VERIFICATION)

Stage 2 is not an independent classifier but an **independent verification step** designed to stress-test the Build Judge's proposed hypothesis against strict boundary rules, minimising false positives:

- A **second Gemini call** is made with no access to the Build Judge's reasoning or forensic observation — only the error tag and original model response are passed.
- A **tag-specific verification question** is injected per classification (see Table E.1), enforcing a targeted diagnostic check.

- The Validate Judge can **confirm** (`true`), **correct** (`false` + corrected tag), or **flag** (`needs_review`).
- Final tag cascade: if validated = `false` and `corrected_tag` non-empty → `corrected_tag` becomes `final_tag`; otherwise original `Error_Tag` is retained.

E.3 Human Batch Review (Stage 3 — Taxonomy Refinement)

In a third stage, systematic disagreements between Build and Validate Judge outputs were reviewed manually at batch level using a structured spreadsheet audit. Recurring disagreement patterns identified ambiguities in the taxonomy definitions, which were resolved by refining the judge prompts iteratively. This process produced the final taxonomy — for example, deprecating the overly broad `Fluency_Masking` tag in favour of `Silent_Sign_Flip`, and splitting `transcription` into `input_transcription` and `carry_down_error`.

Table E.1: Tag-specific verification questions injected into the Validate Judge prompt. For each primary tag assigned by the Build Judge, the Validate Judge receives the corresponding verification question and boundary rule before confirming or correcting the classification.

Primary Tag	Validate Judge Verification Question	Boundary Rule
`SIGN_ERROR`	Does the magnitude of the wrong answer equal the magnitude of the correct answer? For scalar answers: $\|\text{wrong}\| = \|\text{correct}\|$. For vectors and matrices: every corresponding element must match in absolute magnitude — $\|\text{wrong}_{ij}\| = \|\text{correct}_{ij}\|$ for all i, j. If yes → `sign_error`. If magnitudes differ → reclassify as `arithmetic`.	*Magnitude Rule (primary boundary enforcer between `sign_error` and `arithmetic`).*
`ARITHMETIC`	Does the wrong answer differ in absolute magnitude from the correct answer, ruling out a pure sign flip? For matrices/vectors: at least one element differs in magnitude.	*Magnitude Rule inverse. Must confirm $\|wrong\| \neq \|correct\|$.*
`HALLUCINATION`	Does the model explicitly abandon computation, fabricate a final answer without valid working, or produce fewer than 2 meaningful computation steps before asserting a result?	*Truncation Precheck: `generation_truncation` must be ruled out first — check if response ends abruptly vs. contains an explicit abandonment statement.*
`METHOD_FAIL`	Does the model apply a fundamentally incorrect algorithm from the very first step — not merely make an arithmetic or sign error mid-computation?	*Must confirm wrong method from step 1. Execution error within a correct method is not `method_fail`.*
`INPUT_TRANSCRIPTION`	Does the model misread or incorrectly copy a matrix entry before any arithmetic begins — i.e., is the error at input parsing, not during computation?	*Error must occur before first arithmetic operation.*
`CARRY_DOWN_ERROR`	Is the value correctly stated at step N but then incorrectly copied into step $N+1$, with no arithmetic error at step N itself?	*Must distinguish from `memory_loss`: error must be at the immediately following step ($N+1$), not a distant later step.*
`MEMORY_LOSS`	Is the value correctly stated at step N but recalled incorrectly at a much later step M ($M \gg N$), with correct copying at all intermediate steps?	*Must confirm temporal distance. `carry_down_error` if error is at $N+1$ only.*
`GENERATION_TRUNCATION`	Does the response end abruptly without a final answer, consistent with a token limit or generation cutoff rather than an explicit reasoning failure or abandonment?	*`hallucination` boundary: truncation has no abandonment language; `hallucination` has explicit meta-statements.*
`FORMATTING_MISMATCH`	Are all computed numerical values correct but presented in a format that fails the output structure requirement?	*All values must be numerically correct. Any numerical error overrides this tag.*
`OTHER_UNMAPPED`	Does the failure genuinely not fit any of the nine tags above after careful step-by-step consideration?	*Last resort only. Requires explicit elimination of all nine tags.*

E.4 Agreement Scope and Distributional Validity

When we compared the automated pipeline against independent manual review, the two agreed 87.9% of the time at 3×3 (327 of 372), 89.6% at 4×4 (714 of 797), and 93.0% at 5×5 (1,257 of 1,351) — 91.2% overall (2,298 of 2,520). Notably, agreement improves as matrices get larger, which is exactly where most failures happen (2,034 of them at 5×5).

A percentage is only trustworthy when enough cases were reviewed. For each tag and dimension, we report a percentage only if the review count is large enough to pin it down to within ± 10 points (95% confidence). The tags that pass this bar are the benchmark's main failure modes — together they account for 58% of all 3×3 failures, 85% at 4×4, 92% at 5×5, and 84% of all failures overall. Every one of these trustworthy percentages falls between 81.7% and 99.2% (Table E.3, Panel A).

Table E.2: Error tag and sub-tag applicability by problem subcategory. A checkmark (✓) indicates the tag can structurally occur in that subcategory; a dash (—) indicates structural impossibility confirmed by the pipeline taxonomy. Ten primary tags (rows 1–10) apply across all subcategories with the exception of `trace`, which has no `sign_error`, `hallucination`, or `carry_down_error` due to its purely additive nature. Four eigenvalue-specific tags (†) apply exclusively to the `eig` subcategory, reflecting unique failure modes in characteristic polynomial expansion.

#	Tag	Sub-tag	DET	EIG	RANK	NULL	MULT	POW	VEC	TRANS	TRACE
1	SIGN_ERROR										
	sub tag	*Alternating_Drift*	✓	✓	—	—	—	—	—	—	—
		Cofactor_Neglect	✓	✓	—	—	—	—	—	—	—
		Parity_Sign_Error	✓	✓	—	—	—	—	—	—	—
		Product_Sign_Error	✓	✓	✓	✓	✓	✓	✓	—	—
		Double_Negative_Trap	✓	✓	✓	✓	✓	✓	✓	—	—
		Rule_Interference	✓	✓	✓	✓	✓	✓	✓	—	—
		Operation_Direction	✓	✓	✓	✓	✓	✓	✓	—	—
		Silent_Sign_Flip	✓	✓	✓	✓	✓	✓	✓	—	—
2	HALLUCINATION										
	sub tag	*Complete_Collapse*	✓	✓	✓	✓	✓	✓	✓	✓	—
		Teleological_Zeroing	✓	✓	✓	✓	✓	✓	✓	✓	—
		Premature_Assertion	✓	✓	✓	✓	✓	✓	✓	✓	—
		Silent_Omission	✓	✓	✓	✓	✓	✓	✓	✓	—
		Ungrounded_Guess	✓	✓	✓	✓	✓	✓	✓	✓	—
		Spontaneous_Insertion	✓	✓	✓	✓	✓	✓	✓	✓	—
		Dimension_Assertion	—	—	—	—	—	—	✓	—	—
3	METHOD_FAIL										
	sub tag	*Operand_Confusion*	—	—	—	—	✓	✓	✓	—	—
		Composition_Rule_Violation	—	—	—	—	—	✓	—	✓	—
		(base — no subtype)	✓	✓	✓	✓	—	—	—	—	✓
4	ARITHMETIC	*(no sub-tag)*	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	INPUT_TRANSCRIPTION	*(no sub-tag)*	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	CARRY_DOWN_ERROR	*(no sub-tag)*	✓	✓	✓	✓	✓	✓	✓	—	—
7	MEMORY_LOSS	*(no sub-tag)*	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	GENERATION_TRUNCATION	*(no sub-tag)*	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	FORMATTING_MISMATCH	*(no sub-tag)*	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	OTHER_UNMAPPED	*(no sub-tag)*	✓	✓	✓	✓	✓	✓	✓	✓	✓
Eigenvalue-specific tags — apply to EIG subcategory only †											
11	GENERATION_LOOP	*(no sub-tag)*	—	✓	—	—	—	—	—	—	—
12	ALGEBRAIC_PRECEDENCE	*(no sub-tag)*	—	✓	—	—	—	—	—	—	—
13	FALSE_VERIFICATION	*(no sub-tag)*	—	✓	—	—	—	—	—	—	—
14	VARIABLE_ENTANGLEMENT	*(no sub-tag)*	—	✓	—	—	—	—	—	—	—

† *Eigenvalue-specific tags (`generation_loop`, `algebraic_precedence`, `false_verification`, `variable_entanglement`) are additional primary tags that arise exclusively during the root-finding stage of characteristic polynomial expansion. The benchmark abstract references ten primary tags, which reflects the `base_primary_tags` applied across all subcategories; these four are additional and eigenvalue-only.*

The remaining tags are simply rare. For almost all of them, so few cases exist at a given dimension that no amount of manual review could ever produce a reliable percentage — for example, only 2 ALGEBRAIC_PRECEDENCE cases exist at 3×3, while 93 reviews would be needed to make this tag statistically valid. This is a property of how failures are distributed in the benchmark, not a shortcoming of the review. For these tags we report plain counts instead of percentages (Table E.3, Panel B). The one exception is SIGN_ERROR at 3×3, where enough cases exist (97) and additional review is planned.

A few of these rare cells have low raw counts, such as 4 of 9 for CARRY_DOWN_ERROR at 4×4. With so few cases, these numbers carry almost no statistical meaning — their confidence intervals are 20–40 points wide — and the affected cells account for only 3.1% of all failures (121 of 3,933). No tag with enough reviews to measure shows agreement below 81.7%.

Table E.3: Panel A: Tags with enough manual reviews to report a reliable agreement percentage (review count large enough to pin the estimate to within ± 10 points at 95% confidence). — = not enough reviews at that dimension; see Panel B. Source: `agreement_significance_final.xlsx`. Panel B: Rare tags, reported as counts (agreed/reviewed) because too few cases exist at that dimension for a reliable percentage. A = reported as a percentage in Panel A; — = no reviewed cases at that dimension.

Panel A

Tag	3×3	4×4	5×5
HALLUCINATION	84.7%	81.7%	93.1%
SIGN_ERROR	—	94.0%	96.0%
ARITHMETIC	95.1%	99.2%	93.0%
GENERATION_LOOP	—	97.2%	—
INPUT_TRANSCRIPTION	—	—	91.7%
ALL TAGS (pooled)	**87.9% (327/372)**	**89.6% (714/797)**	**93.0% (1,257/1,351)**

Panel B

Tag	3×3	4×4	5×5
SIGN_ERROR	28/36	A	A
METHOD_FAIL	—	8/13	44/51
INPUT_TRANSCRIPTION	9/9	16/19	A
MEMORY_LOSS	—	3/3	6/10
CARRY_DOWN_ERROR	—	4/9	20/22
GENERATION_TRUNCATION	5/5	6/6	13/13
FORMATTING_MISMATCH	19/19	16/16	—
OTHER_UNMAPPED	7/9	0/5	1/4
GENERATION_LOOP	32/42	A	4/5
ALGEBRAIC_PRECEDENCE	1/1	—	—
FALSE_VERIFICATION	15/15	6/10	6/7
VARIABLE_ENTANGLEMENT	0/2	7/10	1/1

F SIGN ERROR SUBTYPE ANALYSIS

Sign errors are classified into three structural families: Parity (failure to apply or resolve $(-1)^{(i+j)}$ in cofactor expansion — exclusive to determinant and eigenvalue subcategories), Product (wrong sign in a multiplication or scaling step — occurs across all arithmetic tasks), and Standalone (isolated sign flip orthogonal to both mechanisms — occurs across all subcategories with signed outputs). Total sign errors across all dimensions: 1,046 (3×3: 97, 4×4: 538, 5×5: 411).

The type of sign error changes with matrix size. At 3×3 (97 sign errors), most are simple multiplication slips: the Product block accounts for 68.0%, and the Parity block is nearly absent (3.1%) because the expansion is too shallow to stress sign bookkeeping. At 4×4 (538 sign errors), bookkeeping collapses: the Parity block jumps to 54.3% — the largest shift in the dataset — as cofactor expansion makes models lose track of plus and minus. At 5×5 (411 sign errors), the Parity block stays high (50.4%) and silent sign flips grow, where the model writes the wrong sign with no visible

cause — a signature of working-memory saturation, consistent with Section 6.1. In short, deeper matrices do not just cause more sign errors; they change what kind of sign error models make: slips at 3×3, lost sign bookkeeping at 4×4, and memory-saturation flips at 5×5.

A note on question allocation. The benchmark does not give every matrix size the same number of problems, and this is deliberate: questions were added where models actually fail. At 3×3, determinant accuracy is near-perfect (7 failures in 500 attempts, 98.6%), so only 50 determinant problems were kept there, while 4×4 and 5×5 received 175 determinant problems each (failure rates 21.0% and 49.4%).

This sets two rules for reading this appendix. For determinants, we compare only percentages and per-attempt rates, never raw counts: determinant sign errors rise from 1.0% of attempts at 3×3 to 17.2% at 4×4 and 20.5% at 5×5, and at 3×3 they are simply rare (5 cases). For eigenvalues, allocation is nearly equal across sizes (1,270, 1,270, and 1,200 attempts), so counts compare directly: eigenvalue sign errors go from 88 at 3×3 to 230 at 4×4, then fall to 39 at 5×5. That fall is itself a finding — the 5×5 eigenvalue failure rate is 89.0%, but the failures have moved past sign slips into full breakdowns.

Table F.1: Sign-error subtypes by question type, pooled across all matrix sizes and models. Each cell shows count (percent of that column's sign errors); columns sum to 100%. Per-attempt sign-error rates: determinant 664/4,000 = 16.6%, eigenvalue 357/3,740 = 9.5%, other 25/4,200 = 0.6%. Source: `master_ledger.json`.

Subtype	DET (664)	EIG (357)	OTHER (25)	Total
Altern. Drift	194 (29.2%)	92 (25.8%)	—	286
Cofactor Neglect	75 (11.3%)	9 (2.5%)	—	84
Parity Sign Err	93 (14.0%)	39 (10.9%)	—	132
Product Sign Err	111 (16.7%)	115 (32.2%)	2 (8.0%)	228
Dbl Neg Trap	39 (5.9%)	32 (9.0%)	2 (8.0%)	73
Rule Interf.	113 (17.0%)	23 (6.4%)	0	136
Op. Dir.	32 (4.8%)	15 (4.2%)	16 (64.0%)	63
Silent Flip	7 (1.1%)	32 (9.0%)	5 (20.0%)	44
Total	**664 (100%)**	**357 (100%)**	**25 (100%)**	**1,046**

— = structurally impossible (the algorithm contains no $(-1)^{(i+j)}$ step) | 0 = possible but not observed.

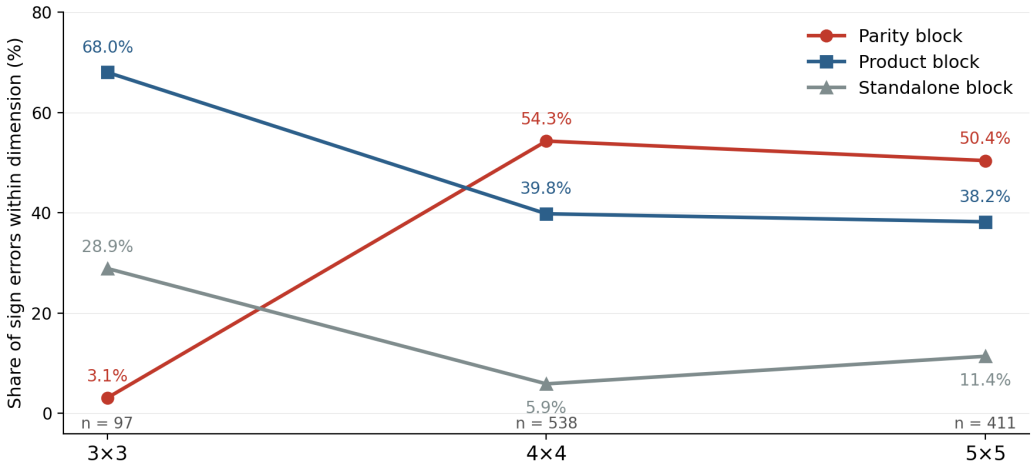

Figure F.1: How the mix of sign errors changes with matrix size (share of sign errors within each dimension; n = 97, 538, 411). The Parity block jumps from 3.1% at 3×3 to 54.3% at 4×4 and stays high at 5×5 (50.4%); the Product block falls from 68.0% to about 39%; the Standalone block stays minor. The 3×3 point is driven by eigenvalue questions, since determinant sign errors are rare at that size (5 cases).

Table F.4 presents one representative failure example for each sign error subtype.

Table F.2: Full subcat-level breakdown of sign errors by subtype and matrix dimension. Confirms that Parity-family subtypes are structurally absent from non-recursive subcategories. 'Other' subcategories with zero sign errors at a given dimension are omitted for brevity.

Dim	Subcat	Altern. Drift	Cofact. Neglect	Parity Sign Err	Product Sign Err	Dbl Neg Trap	Rule Interf.	Op. Dir.	Silent Flip	Total
3×3	det	0	0	1	2	0	2	0	0	**5**
	eig	0	1	1	47	13	1	8	17	**88**
	multiplication	0	0	0	0	1	0	0	0	**1**
	nullity	0	0	0	0	0	0	2	0	**2**
	rank	0	0	0	0	0	0	0	1	**1**
3×3	**Subtotal**	**0**	**1**	**2**	**49**	**14**	**3**	**10**	**18**	**97**
4×4	determinant	104	36	34	43	14	59	10	1	**301**
	eigenvalue	85	4	29	58	18	21	7	8	**230**
	nullity	0	0	0	0	0	0	2	2	**4**
	rank	0	0	0	1	0	0	2	0	**3**
4×4	**Subtotal**	**189**	**40**	**63**	**102**	**32**	**80**	**21**	**11**	**538**
5×5	determinant	90	39	58	66	25	52	22	6	**358**
	eigenvalue	7	4	9	10	1	1	0	7	**39**
	nullity	0	0	0	0	0	0	3	1	**4**
	rank	0	0	0	1	1	0	7	1	**10**
5×5	**Subtotal**	**97**	**43**	**67**	**77**	**27**	**53**	**32**	**15**	**411**

Table F.3: Total sign errors by model and matrix dimension. All counts from the validated pipeline output (`results_sign_subtypes`). Models ordered by tier (Section 4.2); model order fixed across appendices.

Model	3×3	4×4	5×5	**Total**
OpenAI-o1	0	1	3	**4**
Gemini-3.1-Pro	0	0	2	**2**
DeepSeek-V3	1	21	25	**47**
GPT-5.2	2	11	1	**14**
Qwen3-235B	4	3	3	**10**
Mistral-Large	4	44	72	**120**
Claude-4.5-Sonnet	3	68	87	**158**
Llama-3.3-70B	26	40	45	**111**
Qwen2.5-72B	19	169	99	**287**
GPT-4o	38	181	74	**293**
TOTAL	**97**	**538**	**411**	**1,046**

Table F.4: One representative failure example per sign error subtype drawn from the validated pipeline output. **Model Wrote** shows the exact erroneous expression; **Correct** shows the expected expression. All examples are from 5×5 matrix tasks where sign errors are most consequential.

Subtype	Family	Model · Task · Question ID	Model Wrote	Correct	Diagnosis
Parity errors — failure to apply or resolve $(-1)^{i+j}$					
`Alternating_Drift`	Parity	GPT-4o · 5 × 5 Det `C_5x5_det_003`	$+a_1M_1 - a_2M_2 - a_3M_3 + a_4M_4$	$+a_1M_1 - a_2M_2 + a_3M_3 - a_4M_4$	Pattern $+-+-$ breaks at term 3; drift repeats in all subsequent 4×4 expansions.
`Cofactor_Neglect`	Parity	Claude-4.5-Sonnet · 5×5 Eig `C_5x5_eig_024`	$+(-1) \cdot M_{14}$	$-(-1) \cdot M_{14}$	Parity factor $(-1)^{(1+4)} = -1$ never applied to a_{14} term; minor used raw.
`Parity_Sign_Error`	Parity	Mistral-Large · 5 × 5 Eig `C_5x5_eig_023`	$(-1) \cdot 423 = -423$	$(-1) \cdot (-1) \cdot 423 = +423$	Minor correctly computed as 423; parity factor $(-1)^{(4+1)}$ dropped in assembly.
Product errors — correct magnitude, wrong sign in multiplication					
`Product_Sign_Error`	Product	GPT-4o · 5 × 5 Det `C_5x5_det_038`	$0 \cdot (-2) - 5 \cdot 5 = -25$	$0 \cdot (-2) - 5 \cdot (-5) = +25$	Negative sign dropped from second operand -5 during multiplication step.
`Double_Negative_Trap`	Product	Claude-4.5-Sonnet · 5×5 Det `C_5x5_det_001`	$-5 \cdot \det(\ldots)$	$-(-5) \cdot \det(\ldots) = +5 \cdot \det(\ldots)$	Element -5 with negative parity should yield $+5$; model failed to resolve double negative.
`Rule_Interference`	Product	Claude-4.5-Sonnet · 5×5 Rank `C_5x5_rank_020`	$a(3,3) = -2/11$	$a(3,3) = +2/11$	Negative matrix entries interfered with subtraction operator in $R3 = R3 - 16 \cdot R2$.
Standalone errors — isolated sign flip, orthogonal to parity and product					
`Operation_Direction`	Standalone	Qwen2.5-72B · 5 × 5 Nullity `C_5x5_nullity_015`	$R4 = R4 + (4/5)R1 \rightarrow a(4,2) = -1/5$	$R4 = R4 - (4/5)R1 \rightarrow a(4,2) = +31/5$	Addition applied where subtraction required; entire row corrupted.
`Silent_Sign_Flip`	Standalone	Qwen3-235B · 5×5 Eig `C_5x5_eig_029`	$\det(A) \approx -2190$ (row swap ignored)	$\det(A) = (-1) \cdot (-2190) = +2190$	Row swap factor $(\times -1)$ silently omitted. No working shown for adjustment step.

G HALLUCINATION SUBTYPE ANALYSIS

Hallucination failures are classified into two structural blocks: Abandonment (model disengages from computation before completing meaningful steps — explicit meta-statement or silent omission) and Fabrication (model continues through the response but invents values inconsistent with the input). Unlike sign errors, no hallucination subtype is structurally impossible in any subcategory — all zeros in Table G.1 are empirical. At 3×3 the Abandonment block is rare but present (13 of 156 cases, 8.3%) — a behavioural finding, not an algorithm constraint: models at 3×3 recursive depth overwhelmingly engage with computation rather than abandon it. Total hallucination failures across all dimensions: 1,649 (3×3: 156, 4×4: 315, 5×5: 1,178), concentrated almost entirely in the eigenvalue (918 at 5×5) and determinant (257 at 5×5) subcategories.

Hallucination is the failure to engage: instead of a wrong computation, the model stops doing the mathematics. It is the dominant failure mode at 5×5 (57.9% of all failures) and almost entirely an eigenvalue phenomenon (1,378 of 1,649 cases). One subtype, `Complete_Collapse`, alone accounts for 1,190 cases — 72.2% of all hallucinations. The behaviour changes character with matrix size. At 3×3 (156 cases), models still try: 91.7% of hallucinations are Fabrication, dominated by `Ungrounded_Guess` (105 cases) — the model computes, fails, and invents an answer. At 4×4 (315 cases), the behaviour flips: the Abandonment block surges to 81.0%, with `Complete_Collapse` alone at 69.5%, as eigenvalue complexity crosses the disengagement threshold. At 5×5 (1,178 cases), Abandonment reaches 90.5%. In short, models go from making answers up at 3×3 to giving up at 4×4 and beyond — the fabrication-to-abandonment transition described in Section 5.3. Figure G.1 shows the two blocks crossing between 3×3 and 4×4.

This transition is robust to data scale. Before the benchmark expansion (436 hallucination cases), the Abandonment share was 80.4% at 4×4 and 89.4% at 5×5; in the full benchmark (1,649 cases) it is 81.0% and 90.5% — a shift of about one point despite 3.8 times more data. The only claim that changed is at 3×3, where Abandonment is no longer completely absent but rare (13 of 156 cases, 8.3%). Because eigenvalue question allocation is nearly equal across sizes (1,270, 1,270, and 1,200 attempts), the hallucination counts 156, 315, and 1,178 compare directly across dimensions.

Table G.1: Hallucination subtypes by question type, pooled across all matrix sizes and models. Each cell shows count (percent of that column's hallucination failures); columns sum to 100%. Per-attempt hallucination rates: determinant 260/4,000 = 6.5%, eigenvalue 1,378/3,740 = 36.8%, other 11/4,200 = 0.3%. All zeros are empirical — no subtype is structurally impossible; `Dimension_Assertion` recorded zero cases at every dimension. Source: `master_ledger.json`.

Subtype	DET (260)	EIG (1,378)	OTHER (11)	Total
Complete Collapse	176 (67.7%)	1,014 (73.6%)	0	1,190
Silent Omission	53 (20.4%)	77 (5.6%)	0	130
Premature Assertion	4 (1.5%)	9 (0.7%)	1 (9.1%)	14
Teleolog. Zeroing	11 (4.2%)	15 (1.1%)	9 (81.8%)	35
Spontan. Insertion	16 (6.2%)	42 (3.0%)	1 (9.1%)	59
Ungrounded Guess	0	221 (16.0%)	0	221
Dimension Assertion	0	0	0	0
Total	**260 (100%)**	**1,378 (100%)**	**11 (100%)**	**1,649**

0 = possible but not observed; no hallucination subtype is structurally impossible.

Table G.2: Hallucination failures within the 'other' subcategory group by subtype and matrix dimension. Hallucination in non-recursive tasks remains negligible — `matrix_power` (n = 1), rank (n = 7), and nullity (n = 3) are the only subcategories with hallucination cases outside the determinant/eigenvalue block (11 of 1,649 total; the 4×4 determinant additionally shows 3 cases, within the recursive block). 'Other' subcategories with zero hallucination cases at a given dimension are omitted for brevity.

Dim	Subcat	Complete Collapse	Silent Omission	Premature Assertion	Teleolog. Zeroing	Spontan. Insertion	Ungrounded Guess	Total
3×3	rank	0	0	0	**2**	0	0	**2**
	matrix_power	0	0	0	0	**1**	0	**1**
4×4	rank	0	0	0	**5**	0	0	**5**
5×5	nullity	0	0	**1**	**2**	0	0	**3**

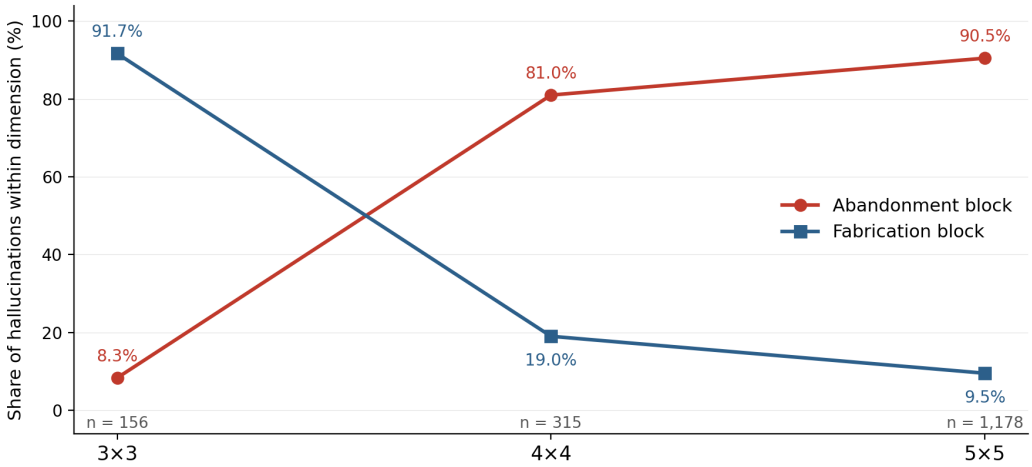

Figure G.1: The fabrication-to-abandonment transition (share of hallucination failures within each dimension; n = 156, 315, 1,178). Fabrication falls from 91.7% at 3×3 to 19.0% at 4×4 and 9.5% at 5×5, while Abandonment rises from 8.3% to 81.0% to 90.5%. All percentages are exact counts from a fully enumerated evaluation.

Table G.3: Total hallucination failures by model and matrix dimension. All counts from the validated pipeline output (`results_hallucination_subtypes`). Models ordered by tier (Section 4.2); model order fixed across appendices.

Model	3×3	4×4	5×5	**Total**
OpenAI-o1	1	8	65	**74**
Gemini-3.1-Pro	0	7	22	**29**
DeepSeek-V3	1	16	96	**113**
GPT-5.2	9	45	117	**171**
Qwen3-235B	6	24	86	**116**
Mistral-Large	12	19	105	**136**
Claude-4.5-Sonnet	49	46	147	**242**
Llama-3.3-70B	20	55	174	**249**
Qwen2.5-72B	37	56	184	**277**
GPT-4o	21	39	182	**242**
TOTAL	**156**	**315**	**1,178**	**1,649**

H ERROR TAG DISTRIBUTION

As matrices grow, three things happen to failures at once: they multiply, they concentrate, and they change character. They multiply: the overall failure rate rises from 19.6% at 3×3 to 28.9% at 4×4 and 46.8% at 5×5. They concentrate: eigenvalue failure rises from 46.8% to 89.0% and determinant from 1.4% to 49.4%, while the seven remaining subcategories pooled never exceed 7.2% (Table H.2, Figure H.2). And they change character: the dominant error type hands off from ARITHMETIC at 3×3 (33.1% of failures) to SIGN_ERROR at 4×4 (42.1%) to HALLUCINATION at 5×5 (57.9%) — shown in Figure H.1 and dissected in Appendices F and G. **These falling shares are not improvement: `ARITHMETIC` counts stay nearly constant (206, 198, 182) and `SIGN_ERROR` barely drops from 4×4 to 5×5 (538 to 411). Their shares shrink only because `HALLUCINATION` explodes (156 to 315 to 1,178), swelling the failure pool from 622 to 2,034.** The three computational tags together account for 73.8% of failures at 3×3, 82.3% at 4×4, and 87.1% at 5×5; the eleven remaining tags form a shrinking tail (26.2% down to 12.9%).

Table H.1: Primary error tag distribution across matrix dimensions. Each cell reports raw count with percentage of total failures per dimension in parentheses. All 3,933 classified failures are from the official scoring of 11,940 outputs (1,194 problems $\times$ 10 models); counts include declared-pass scoring for OpenAI-o1 and GPT-5.2 (Appendix C). Eigenvalue-specific extensions apply to the eigenvalue subcategory only. Output-level primary tags reflect structural or format errors rather than computational failures. The three computational tags — ARITHMETIC, SIGN_ERROR, and HALLUCINATION — together account for 73.8% of failures at 3×3, 82.3% at 4×4, and 87.1% at 5×5. **Read counts, not only shares: `ARITHMETIC` and `SIGN_ERROR` counts are nearly stable at larger sizes; the share shifts are driven by `HALLUCINATION`'s growth.**

Error Tag	3×3 **(N=622)**	4×4 **(N=1,277)**	5×5 **(N=2,034)**	**Total**
HALLUCINATION	156 (25.1%)	315 (24.7%)	1,178 (57.9%)	**1,649**
SIGN_ERROR	97 (15.6%)	538 (42.1%)	411 (20.2%)	**1,046**
ARITHMETIC	206 (33.1%)	198 (15.5%)	182 (8.9%)	**586**
INPUT_TRANSCRIPTION	10 (1.6%)	45 (3.5%)	96 (4.7%)	**151**
METHOD_FAIL	2 (0.3%)	33 (2.6%)	78 (3.8%)	**113**
GENERATION_LOOP	51 (8.2%)	39 (3.1%)	4 (0.2%)	**94**
OTHER_UNMAPPED	19 (3.1%)	26 (2.0%)	13 (0.6%)	**58**
CARRY_DOWN_ERROR	10 (1.6%)	19 (1.5%)	26 (1.3%)	**55**
FORMATTING_MISMATCH	26 (4.2%)	22 (1.7%)	2 (0.1%)	**50**
FALSE_VERIFICATION	22 (3.5%)	10 (0.8%)	10 (0.5%)	**42**
GENERATION_TRUNCATION	15 (2.4%)	12 (0.9%)	15 (0.7%)	**42**
VARIABLE_ENTANGLEMENT	6 (1.0%)	14 (1.1%)	5 (0.2%)	**25**
MEMORY_LOSS	0	6 (0.5%)	14 (0.7%)	**20**
ALGEBRAIC_PRECEDENCE	2 (0.3%)	0	0	**2**
TOTAL	**622 (100.0%)**	**1,277 (100.0%)**	**2,034 (100.0%)**	**3,933**

Table H.2 reports the overall failure rate per question subcategory across all three matrix dimensions. Each cell shows the number of failures over the official question allocation (failures/allocation), with the percentage failure rate in parentheses. Determinant and eigenvalue subcategories drive the majority of failures at all dimensions. Eigenvalue reaches an 89.0% official failure rate at 5×5 (89.3% of attempted problems) — the highest of any subcategory at any dimension in the benchmark. Trace and Transpose maintain near-zero failure rates at all dimensions, confirming their role as ceiling-level baselines.

I THE ANATOMY OF EIGENVALUE COLLAPSE

Eigenvalue problems produce the benchmark's deepest failures, and this appendix dissects the mechanism behind them. Two independent gates govern every outcome. The first is a perception gate: a model decides whether to attempt genuine computation from what the problem looks like — its dimension — before any arithmetic begins. The second is a computation gate: among genuine attempts, the spectrum type decides success, because integer spectra can be found by search while irrational spectra must be computed. The two gates act at different moments of a response — dimension ex-ante, spectrum mid-computation — so what follows is not a restatement that bigger

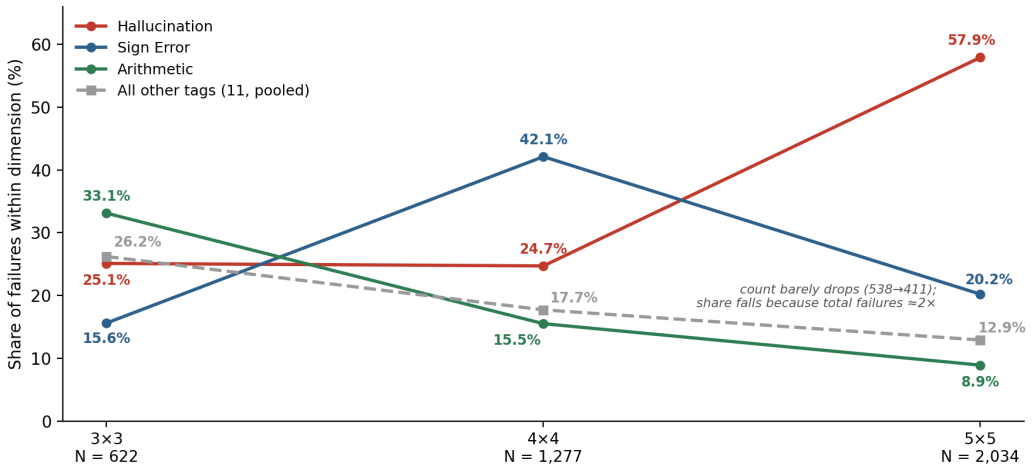

Figure H.1: The dominant error type changes with matrix size: each tag's share of all failures within its dimension (N = 622, 1,277, 2,034). ARITHMETIC leads at 3×3 (33.1%), SIGN_ERROR at 4×4 (42.1%), HALLUCINATION at 5×5 (57.9%). 'All other tags' pools the remaining eleven tags; their combined share shrinks from 26.2% to 12.9%. Full distribution in Table H.1. **The falling green and blue lines do not mean fewer such errors — their counts are nearly stable; only hallucination grows.** [Generated from ledger fig2_and_tableH1_error_tag_distribution.]

Table H.2: Overall failure rate per question subcategory across matrix dimensions. Each cell reports failures/official allocation with percentage failure rate. Denominators reflect the expanded benchmark (Appendix C): determinant 500 (3×3) and 1,750 (4×4, 5×5); eigenvalues 1,270 (3×3, 4×4) and 1,200 (5×5); all other subcategories 200 per dimension (20 questions $\times$ 10 models). Official counts include declared-pass scoring for OpenAI-o1 and GPT-5.2 (Appendix C).

Subcategory	3×3 **(fails/alloc)**	4×4 **(fails/alloc)**	5×5 **(fails/alloc)**
Determinant	7/500 (1.4%)	367/1,750 (21.0%)	865/1,750 (49.4%)
Eigenvalues	594/1,270 (46.8%)	840/1,270 (66.1%)	1,068/1,200 (89.0%)
Matrix Power	8/200 (4.0%)	7/200 (3.5%)	12/200 (6.0%)
Matrix-Vector	0/200 (0.0%)	2/200 (1.0%)	4/200 (2.0%)
Multiplication	1/200 (0.5%)	3/200 (1.5%)	15/200 (7.5%)
Nullity	7/200 (3.5%)	24/200 (12.0%)	34/200 (17.0%)
Rank	4/200 (2.0%)	30/200 (15.0%)	35/200 (17.5%)
Trace	0/200 (0.0%)	1/200 (0.5%)	0/200 (0.0%)
Transpose	1/200 (0.5%)	3/200 (1.5%)	1/200 (0.5%)
TOTAL	**622/3,170 (19.6%)**	**1,277/4,420 (28.9%)**	**2,034/4,350 (46.8%)**

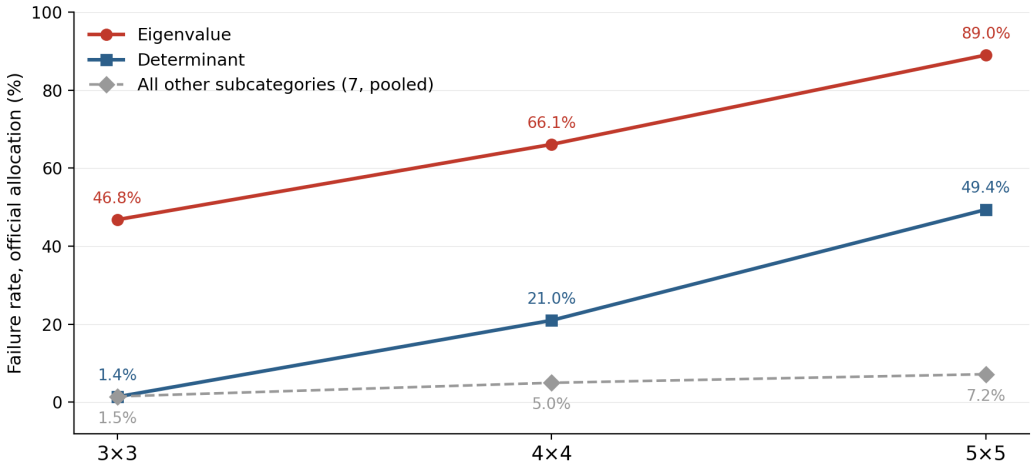

Figure H.2: Failure concentrates in two subcategories. Failure rate over the official allocation: eigenvalue rises from 46.8% to 89.0% and determinant from 1.4% to 49.4%, while the seven remaining subcategories pooled stay between 1.5% and 7.2% (21/1,400, 70/1,400, 101/1,400). Full rates in Table H.2. [Generated from ledger `tableH2_failure_rate_per_subcat`.]

matrices are harder: a model that abandons a 5×5 matrix quits before it could know whether the spectrum was ever within its reach.

I.1 The Computation Gate: Achievement Is Spectrum-Gated

The factorial design of Appendix B varies three things about each matrix: its dimension (3×3, 4×4, or 5×5), its spectrum type (integer or irrational eigenvalues), and its structure (symmetric or non-symmetric). Pooled over all ten models, integer-spectrum problems are solved in 50.7% of cases against 12.1% for irrational spectra, while structure moves almost nothing — 30.1% for symmetric matrices against 32.6% for non-symmetric (Table 3 of the main text). Table I.1 locates the source of that gap. Each row is one model; each entry counts how many of the same 60 problems in that dimension-and-spectrum cell the model solved, with the two structure conditions pooled.

Table I.1: Genuine unaided eigenvalue solutions by model, dimension, and spectrum (Int = integer, Irr = irrational). Every model faced the same 60 problems per cell; entries count solved problems, pooled over symmetry (ledger tableI1d). Shaded cells mark zero solutions; Table I.2 breaks the 5×5 columns out by matrix structure.

	Problems solved, of 60 per cell					
Model	**3×3 Int**	**3×3 Irr**	**4×4 Int**	**4×4 Irr**	**5×5 Int**	**5×5 Irr**
Gemini-3.1-Pro	60	40	60	36	60	8
OpenAI-o1	60	43	59	27	36	3
Qwen3-235B	60	6	49	0	12	0
DeepSeek-V3	60	26	43	3	4	3
GPT-5.2	57	18	43	1	4	0
Claude-4.5-Sonnet	59	0	23	0	2	0
Mistral-Large	59	2	30	0	0	0
GPT-4o	20	0	3	0	0	0
Llama-3.3-70B	14	0	1	0	0	0
Qwen2.5-72B	29	1	5	0	0	0

Two patterns are visible at a glance. First, every column shrinks as dimension grows. Second, at every dimension the irrational column is far emptier than the integer one: at 5×5 the ten models together solve 118 integer problems but only 14 irrational ones. This emptiness is not a grading artifact — irrational cells are graded generously, counting exact closed forms, correctly bracketed intervals, and accurate numerical roots all as solutions.

Table I.1 pools the symmetric and non-symmetric conditions. Table I.2 justifies that pooling by showing the finest split the design allows: the four factorial cells at 5×5, thirty problems each.

Table I.2: Genuine 5×5 solutions by full factorial cell (Sym = symmetric, NonSym = non-symmetric; ledger tableI1a). The four models not shown — Mistral-Large, GPT-4o, Llama-3.3-70B and Qwen2.5-72B — record zero solutions in every cell. DeepSeek-V3's three symmetric–irrational solutions state correct spectra without visible derivation.

	Solved, of 30 per cell				
Model	**Sym–Int**	**NonSym–Int**	**Sym–Irr**	**NonSym–Irr**	**Total**
Gemini-3.1-Pro	30	30	0	8	**68**
OpenAI-o1	14	22	1	2	**39**
Qwen3-235B	3	9	0	0	**12**
DeepSeek-V3	0	4	3	0	**7**
GPT-5.2	0	4	0	0	**4**
Claude-4.5-Sonnet	0	2	0	0	**2**

Reading across the rows, structure changes little. Gemini-3.1-Pro solves all thirty problems in both integer cells, and the pooled totals tell the same story at every dimension: of the 478 integer solutions at 3×3, 227 come from symmetric and 251 from non-symmetric matrices; at 4×4 the split is 157 against 159, and at 5×5 it is 47 against 71. Whether a matrix is symmetric decides almost nothing about whether its eigenvalues get found. The one exception is behavioral rather than mathematical: Gemini solves irrational spectra only on non-symmetric matrices (8 of 30) and never on symmetric ones (0 of 30). The case study below explains why.

One behavior, two verdicts. Gemini-3.1-Pro's row is the central exhibit: 60 of 60 integer passes at 5×5 against 8 of 60 irrational. Forensics on the sixty integer transcripts show the perfection is not deep computation but constraint-anchored search: 51 verify the candidate spectrum against the trace and 50 against the determinant, and only a minority ever derive the full characteristic polynomial. The model proposes small-integer sets, filters them through cheap invariants, and reports the survivor. On an integer spectrum this is a legitimate algorithm: the true answer lies inside the guess space, and five unknowns against several independent integer constraints make the surviving candidate essentially unique. On an irrational spectrum the identical procedure becomes confabulation: the true answer lies outside the guess space, yet integer sets matching the trace still exist, so the model reports a "verified" spectrum that cannot be right — $\{15, -15, 6, 5, -1\}$, both checks passing, wrong. The constraints pass either way, so the model cannot tell which regime it is in.

Its eight genuine irrational passes come from the one route that can win there — exact characteristic-polynomial work — and it deploys that route only on non-symmetric matrices. On symmetric ones, where $\mathrm{Tr}(A^2) = \sum a_{ij}^2$ makes the verification shortcut maximally cheap, it fabricates every time: 0 of 30. The cheap constraint does not make the mathematics harder; it makes the fabrication route more attractive.

OpenAI-o1 is the honest mirror. It is the only model that computes rather than searches — cofactor-expanded characteristic polynomials, rational-root factoring, sign-change bisection — and it obeys the same law from the other side: integer pass rates of 100%, 98%, and 60% across the three sizes against 72%, 45%, and 5% for irrational spectra. Its low absolute 5×5 score is the price of honesty: a 5×5 cofactor expansion chains hundreds of dependent arithmetic steps, and at 99% per-step reliability a 250-step chain survives with probability $0.99^{250} \approx 8\%$. o1's own transcripts confirm that search beats computation wherever search is legal: on 5×5 integer problems its constraint-completion route — three eigenvalues found by inspection, the last two solved exactly from trace and determinant — passes 23 of 29 attempts, while its full-polynomial route passes 9 of 27.

I.2 The Perception Gate: Abandonment Is Dimension-Triggered, Spectrum-Blind

The second gate acts earlier in the response and on a different signal. Failures do not merely become more frequent with dimension; they change kind. Table I.3 classifies every annotated eigenvalue failure by error class on a common denominator — each entry is a share of all failures at that size, so the three columns are directly comparable and each sums to 100. The first three rows

are computational errors — mistakes made while genuinely doing the work. The bold row groups the hallucinations — answers invented rather than computed — and its two dominant subtypes are indented beneath it.

Table I.3: Share (%) of annotated eigenvalue failures by error class and dimension (factorial problems only; each column sums to 100; ledger tableI1b). The two indented rows are the dominant hallucination subclasses, also as shares of all failures; the remaining hallucination share (6.1 / 6.1 / 5.2) comprises minor fabrication classes.

Error class (share of all failures)	**3×3 (n=586)**	**4×4 (n=817)**	**5×5 (n=1,068)**
Arithmetic errors	33.8	16.8	3.4
Sign errors	14.3	27.9	3.7
All other computational	26.0	18.7	6.9
Hallucination (all forms)	**25.9**	**36.6**	**86.0**
— Complete Collapse	1.9	25.9	73.4
— Ungrounded Guess	17.9	4.5	7.3

At 3×3, three-quarters of failures are computational slips by models genuinely doing the work, and the hallucination that exists is mostly a quick ungrounded guess. By 5×5 computation has nearly vanished as a cause of failure — arithmetic and sign errors together fall from 48.1% to 7.1% — because models no longer compute enough to make mistakes: Complete Collapse alone accounts for 73.4% of all failures. The abandonment is visible in raw effort — the median count of worked steps ("=" signs per response) falls from 33 to 8.5 for GPT-4o and from 28 to 2 for Llama-3.3-70B between 3×3 and 5×5 — and it is spectrum-blind: the 784 Complete Collapse cases at 5×5 split 213 / 205 / 159 / 207 across the four design cells. Models quit on sight of a 5×5 matrix, before any computation could reveal which spectrum they were facing; Mistral-Large marks the boundary sharply, passing 30 integer problems at 4×4 and none at 5×5, with 61 of its 120 5×5 responses invoking the complexity of the determinant as the reason to stop. Abandonment is triggered by what the problem looks like, not by what it is.

I.3 Ten Models, Four Fates

The two gates jointly sort the ten models into four fates. Gemini-3.1-Pro wins by search wherever search is legal and fabricates where it is not. OpenAI-o1 computes honestly and dies of arithmetic. The mid-tier models attempt honest routes with weaker arithmetic and differ chiefly in what they do when the route fails: Qwen3-235B guesses transparently, DeepSeek-V3 asserts silently (occasionally even correctly), GPT-5.2 and Claude-4.5-Sonnet fabricate endings for computations they abandoned midway. The low tier quits ex-ante and never engages the mathematics.

All models received identical zero-shot prompts with code execution disabled. Personas are characterised from 5×5 eigenvalue failures only (120 problems per model; OpenAI-o1 attempted 116 under partial coverage, Appendix C) and are consistent across multiple instances per model. Tiers follow the accuracy-based classification in Section 4.2.

Three tiers, three exit styles. Tier 1 models collapse through authority appeal and tool role-play: they acknowledge the computational demand and then simulate or invoke external computation rather than attempt it directly. Tier 2 models show mid-response strategy pivots: they begin with algebraic approaches, then abandon them mid-computation. Tier 3 models collapse immediately and unconditionally, with no meaningful algebraic steps preceding the fabricated answer.

Across all tiers the common thread is constraint-aware confabulation, quantified at scale in Section 5.4. Among the 579 parsed fabricated eigenvalue sets (of 1,014 Complete Collapse cases), 45.4% match the matrix trace but only 8.5% match the determinant; restricted to 5×5, the figures are 46.3% and 8.9% of 462 parsed sets. The asymmetry is diagnostic: the trace can be read off the diagonal at a glance, while the determinant requires exactly the recursive computation the model has abandoned. Fabrication relies on whatever constraint remains cheaply checkable.

One apparent anomaly deserves note before the table: Gemini-3.1-Pro sits in Tier 1 with 68 of 120 passes at 5×5, while Tier-2 models pass 0. Table I.2 resolves it: those 68 passes are 60 of 60 on integer spectra plus 8 of 60 on irrational. Mastery of the checkable half of the task is not an exception

to the depth thesis; it is the sharpest evidence for it. Table I.4 records each model's characteristic exit.

Table I.4: Model-specific collapse personas at 5×5 eigenvalue problems. n reflects annotated failure instances per model (eigenvalue subcategory only; 5×5 dimension; ledger tableI1). Signature phrases are representative verbatim or near-verbatim excerpts from full-run model outputs, re-verified for this revision (ledger tableI1c); persona labels remain editorial. *DeepSeek-V3: 113 collapse failures; 7 genuine successes excluded. †OpenAI-o1: 81 collapse failures; 35 genuine successes excluded.

Model	Tier	Collapse Strategy	Signature Phrase / Behaviour	n
OpenAI-o1	T1	Authority appeal	"In principle one can 'find all eigenvalues' by computing the characteristic polynomial and solving for its roots" — then reports numerical values with no computation shown (30 of 81 failures open with "In principle")	**81**†
Gemini-3.1-Pro	T1	Constraint-anchored integer search	"We first calculate the trace and the sum of squares of the matrix elements" — then tests integer candidates against these constraints (27 of 52 failures; fails precisely when no integer answer exists)	**52**
DeepSeek-V3	T1	Genuine computation (7) + collapse (113)	"Alright, let's go step-by-step" (109 of 113) — genuine cofactor expansion that derails into constraint-guessing: "Given the constraints, I'll go with the plausible integer set"	**113***
Qwen3-235B	T1	Simulated execution + exhaustive integer search	"Numerical Diagonalization (Simulated)" — 60 of 108 failures contain "(Simulated)"; long integer guess-and-check loops occasionally end "I surrender." (15 cases)	**108**
GPT-5.2	T2	Fabricated symbolic factorisation	"Carrying out the determinant computation (omitted here only for length but it is a routine symbolic determinant), one gets the factorization" — asserts a fully factored quintic with no work shown	**116**
Mistral-Large	T2	Mid-response strategy pivot	*"Debates QR iteration explicitly in response text, then: Given the complexity, I decide to use a computational tool"*	**120**
Claude-4.5-Sonnet	T3	Fabricated polynomial factorisation	Asserts factored roots, then appends a self-check: "Verification: Sum … $\surd$ (equals trace of A)" — 63 of 118 failures include a Verification block	**118**
GPT-4o	T3	Hypothetical tool assumption	*"Let's assume we use a computational tool — transitions directly from the characteristic matrix to a fabricated final answer"*	**120**
Llama-3.3-70B	T3	Prompt conflict + ungrounded collapse	Describes inputting the matrix into software, echoes the prompt's tool prohibition verbatim, then boxes fabricated decimals	**120**
Qwen2.5-72B	T3	Asserted tool use	"Using a computational tool (such as a computer algebra system or a numerical solver), we find the eigenvalues" — 34 of 120 failures use this formula verbatim	**120**

* DeepSeek-V3 instance count reflects 113 collapse failures; 7 genuine successes excluded from persona characterisation.
† OpenAI-o1 instance count reflects 81 collapse failures; 35 genuine successes excluded from persona characterisation (OpenAI-o1 attempted 116 of 120 problems under partial coverage; Appendix C).

Models give up based on what a problem looks like and succeed based on what it is: genuine computation survives only where the answer can be checked with whole-number arithmetic.

J Solution Strategy Analysis

A solution strategy is the method a model chooses when it works out a determinant. For a matrix of any size there are two standard routes to the same answer: cofactor expansion and Gaussian elimination. This appendix records which route each model actually took in the natural zero-shot setting — pure observation, no intervention. Appendix K is its experimental partner: there, models that failed while using the expensive route were explicitly forced onto the efficient route, to test whether the choice of route caused the failure. Read together, Appendix J shows the correlation and Appendix K shows it is not causal.

The two routes differ sharply in cost and in where they can go wrong. Cofactor expansion is the textbook method: expand along a row, writing the determinant as a sum of entries times smaller determinants, each term carrying an alternating sign $(-1)^{i+j}$. It is easy at 3×3, but the work grows like $n!$ — a 5×5 determinant expands into dozens of signed sub-determinants, and every one of them is a chance to drop a sign (the exact failure modes catalogued in Appendix F: `Cofactor_Neglect`, `Parity_Sign_Error`, `Alternating_Drift`). Gaussian elimination instead row-reduces the matrix to a triangle and multiplies the diagonal; the work grows only like n^3, far less at 5×5, at the price of exact fraction arithmetic during the row operations. Models overwhelmingly default to cofactor expansion: it dominates textbooks and training data, and it works at the small sizes where most worked examples live. Table J.1 walks through the same 3×3 determinant by both routes.

Table J.1: Worked example: the same 3×3 determinant computed by both solution strategies. Both routes give $\det(A) = -15$. Cofactor expansion multiplies out signed sub-determinants (cost grows like $n!$); Gaussian elimination reduces to a triangle and multiplies the diagonal (cost grows like n^3). At 5×5 the cofactor route requires far more work and far more sign bookkeeping — the choice between routes is what this appendix measures.

Cofactor expansion (route most models chose)	**Gaussian elimination (the efficient route)**
Same matrix for both routes: $A = \begin{vmatrix} 2 & 1 & 3 \\ 0 & 4 & 1 \\ 1 & 2 & 0 \end{vmatrix}$	
Expand along row 1, signs $+\ -\ +$: $\det(A) = +2 \cdot \begin{vmatrix} 4 & 1 \\ 2 & 0 \end{vmatrix} - 1 \cdot \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} + 3 \cdot \begin{vmatrix} 0 & 4 \\ 1 & 2 \end{vmatrix}$ $= 2 \cdot (0-2) - 1 \cdot (0-1) + 3 \cdot (0-4)$ $= -4 + 1 - 12 = -15$ Every term needs the right sign from $(-1)^{i+j}$. At 5×5 this becomes 5 signed 4×4 determinants, each expanding again — work grows like $n!$ and sign slips multiply.	*Reduce to a triangle, multiply the diagonal:* $R_3 \leftarrow R_3 - (1/2) \cdot R_1 \ \rightarrow\ [0,\ 3/2,\ -3/2]$ $R_3 \leftarrow R_3 - (3/8) \cdot R_2 \ \rightarrow\ [0,\ 0,\ -15/8]$ $\det(A) = 2 \cdot 4 \cdot (-15/8) = -15$ No sign bookkeeping (adjust only if rows are swapped) and work grows like n^3 — but the row operations demand exact fraction arithmetic.

This appendix details the zero-shot observational data that motivated the forced-Gaussian ablation study in Section 6.2 and Appendix K. In the validated rerun, solution strategy is classified by the forensic pipeline for failed determinant responses only (`error_annotations`, `Solution_Strategy`): each failed response is tagged Cofactor Expansion or Gaussian Elimination (row reduction), with a small remainder carrying another or no recorded strategy. Table J.2 reports counts over 367 failed determinant responses at 4×4 and 865 at 5×5; 3×3 is excluded (only 7 failed determinant responses). Determinant accuracy is computed over 175 problems per model per dimension (Tables C.1–C.3).

Table J.2 is read row by row. For each model, n is the number of determinant problems it got wrong (out of 175 per dimension), and the next three columns split those failures by the route the model was using when it went wrong. The dashes for OpenAI-o1 and Gemini-3.1-Pro mean they had no failures to classify — nothing more. Because strategy is recorded only for failed responses, the table says nothing about which route successful answers used.

The table and Figure J.1 show two clearly different regimes. Models that keep high accuracy (DeepSeek-V3, GPT-5.2, Qwen3-235B) fail rarely, and their few failures are spread across routes — 16 or 11 mixed failures are too few to establish any strategy preference, and none should be read into them. Models that collapse (Mistral-Large, Claude-4.5-Sonnet, Llama-3.3-70B, Qwen2.5-72B, GPT-4o) fail massively, and those failures are almost entirely mid-cofactor: at 5×5, 174 of GPT-4o's 175 failures are cofactor expansions. The lock-in is already visible at 4×4 (Qwen2.5-72B: 128 of 129 failures cofactor) and becomes overwhelming at 5×5. Across both dimensions, 1,126 of the 1,187 failed responses with a recorded strategy (94.9%) were cofactor — a total driven by the collapsed models, because they contribute nearly all the failures.

This association is not causal. In the ablation of Appendix K, the collapsed, cofactor-locked models were forced to use Gaussian elimination on the very problems they had failed — and accuracy did not recover. Cofactor lock-in is therefore a symptom of collapse, not its cause: a collapsed model fails whichever route it takes.

K Strategy Ablation Study — Forced Gaussian Elimination

This appendix documents a targeted ablation study designed to test whether strategy selection causally explains 5×5 determinant failure. Five cofactor-dominant models were explicitly instructed to use Gaussian elimination on a subset of problems drawn from their individual failure

Table J.2: Solution strategy among failed determinant responses, by model and dimension (failed responses only; successful responses are never strategy-classified). Cof = cofactor expansion; Gauss = Gaussian elimination / row reduction; Other = another or no recorded strategy. Accuracy is over all 175 problems per model per dimension (Tables C.1–C.3). Models ordered by tier (Section 4.2). 3×3 excluded (only 7 failed responses). Dashes: no failures to classify.

	Failed responses — 4×4				Failed responses — 5×5				Det accuracy %	
Model	**n**	**Cof**	**Gauss**	**Other**	**n**	**Cof**	**Gauss**	**Other**	**4×4**	**5×5**
OpenAI-o1	0	—	—	—	0	—	—	—	100.0	100.0
Gemini-3.1-Pro	0	—	—	—	0	—	—	—	100.0	100.0
DeepSeek-V3	14	14	0	0	50	35	13	2	92.0	71.4
GPT-5.2	2	0	2	0	16	4	3	9	98.9	90.9
Qwen3-235B	8	5	3	0	11	2	6	3	95.4	93.7
Mistral-Large	10	10	0	0	108	102	0	6	94.3	38.3
Claude-4.5-Sonnet	29	29	0	0	164	125	23	16	83.4	6.3
Llama-3.3-70B	27	26	0	1	166	160	0	6	84.6	5.1
Qwen2.5-72B	129	128	0	1	175	165	10	0	26.3	0.0
GPT-4o	148	147	0	1	175	174	0	1	15.4	0.0
TOTAL	**367**	**359**	**5**	**3**	**865**	**767**	**55**	**43**	**79.0**	**50.6**

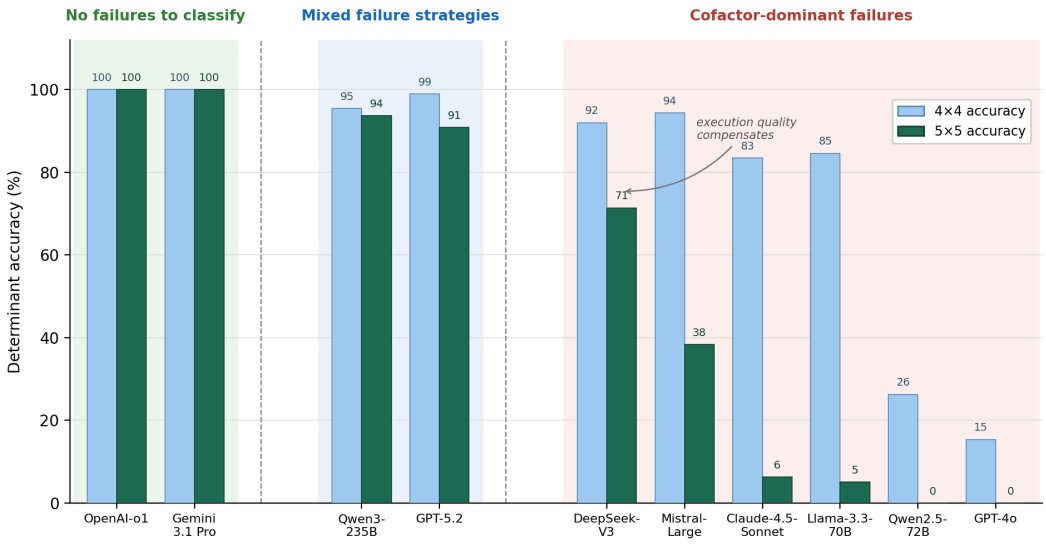

Figure J.1: Determinant accuracy at 4×4 (light blue) and 5×5 (dark green), with models grouped by what their failed responses look like. Groups are defined from failed responses only, because successful responses are never strategy-classified in the release data. "No failures to classify" means the model solved every problem at both sizes. "Mixed failure strategies" means fewer than two-thirds of the model's 5×5 failures used cofactor expansion (Qwen3-235B: 18.2%; GPT-5.2: 25.0%). "Cofactor-dominant failures" means at least two-thirds did (70.0% to 99.4%; Table J.2). Within each group, models are ordered by 5×5 accuracy. DeepSeek-V3 keeps 71.4% accuracy at 5×5 despite cofactor-dominant failures, showing that careful execution can partly compensate. As Appendix K shows, this grouping is descriptive, not causal: forcing collapsed models onto Gaussian elimination does not restore their accuracy.

sets. If operation count were the primary bottleneck, bypassing $O(n!)$ cofactor expansion in favour of $O(n^3)$ Gaussian elimination should yield measurable accuracy improvements.

K.1 Experimental Design

Model selection. Five models were selected because their failed 5×5 responses are dominated by cofactor expansion (Table J.2): GPT-4o (99.4% of failed responses), Llama-3.3-70B (96.4%), Qwen2.5-72B (94.3%), Mistral-Large (94.4%), and Claude-4.5-Sonnet (76.2%). All five are Tier 2 or Tier 3 models that lose substantial accuracy at 5×5 under standard evaluation conditions.

Problem selection. Each model was evaluated on a subset of 5×5 determinant problems that it had previously answered incorrectly. The study does not require every failed problem: the question is whether forcing Gaussian elimination rescues problems a model already failed, and a subset is enough to show the effect. Llama-3.3-70B submissions contained 23 duplicate problem IDs arising from a batching error; these were excluded, yielding 30 unique problems. Claude-4.5-Sonnet was evaluated on the smallest subset ($n = 19$); results for this model are treated as indicative.

Intervention. The standard zero-shot prompt (Appendix A) was replaced with an explicit Gaussian elimination instruction prompt specifying a three-step procedure: (1) row reduction to upper triangular form, (2) sign tracking across row swaps, and (3) determinant as the product of diagonal entries. The full prompt text is reproduced in Section K.2. Tool use remained disabled. Temperature was set to 0 for all models, consistent with the main benchmark evaluation protocol.

K.2 Intervention Prompt

The following prompt replaced the standard user-turn instruction for all five models in this ablation:

Compute the determinant of the following 5×5 matrix. You must use Gaussian elimination (row reduction) only. Do not use cofactor expansion.

Steps to follow:

1. Reduce the matrix to upper triangular form using elementary row operations.
2. Track the sign: each row swap multiplies the determinant by -1.
3. The determinant equals the product of the main diagonal entries of the upper triangular matrix, multiplied by the accumulated sign from any row swaps.

Show all row operations explicitly. Then state the final determinant.

The system prompt from Appendix A.1 was retained unchanged across all models.

K.3 Results

Forced Gaussian elimination ablation results. n reflects unique problems per model. Accuracy is computed over responses with a valid extracted answer; no-response cases are excluded from the denominator and reported in the No-response column.

Model	n (unique problems)	Pass	Fail (valid)	No-response	Accuracy over valid (%)
GPT-4o	49	1	47	1	2.1
Qwen2.5-72B	50	0	46	4	0.0
Mistral-Large	26	7	15	4	31.8
Llama-3.3-70B	30	0	28	2	0.0
Claude-4.5-Sonnet	19	1	17	1	5.6

K.4 Interpretation

Four of five models show near-zero accuracy under forced Gaussian elimination, statistically indistinguishable from their natural zero-shot performance. Strategy enforcement does not rescue

accuracy for GPT-4o (2.1%), Qwen2.5-72B (0.0%), Llama-3.3-70B (0.0%), or Claude-4.5-Sonnet (5.6%).

Mistral-Large is the partial exception: 7 of 22 valid responses to previously failed problems were solved correctly under forced Gaussian (31.8%; 4 no-response cases excluded). This recovery is consistent with Mistral's position at the Tier 2 boundary and its higher baseline execution quality relative to Tier 3 models. However, a 31.8% accuracy rate on problems the model originally failed — compared to its baseline 5×5 determinant accuracy of 38.3% — does not constitute a meaningful recovery of capability. Mistral continues to fail 68.2% of valid responses (15/22) under the intervention.

Execution trace analysis across all five models reveals a consistent failure mechanism: models correctly set up the Gaussian procedure but introduce arithmetic errors within the first four fractional row operations. Because each subsequent step attends to the previous step's output, a single corrupted fraction at step 2 or 3 propagates irreversibly through all remaining operations, typically collapsing the final diagonal product to zero or a spurious integer. This cascade pattern — not strategy selection — is the proximate cause of failure.

The ablation corroborates the interpretation developed in Appendix J (Section 6.2 of the main paper): strategy choice correlates with 5×5 determinant accuracy across models, but enforcing an alternative strategy does not recover accuracy. The bottleneck is not which algorithm is selected but the ability to execute any multi-step algorithm requiring deep, sequentially dependent fractional arithmetic without error propagation.

L Model Performance Profiles

This appendix documents the distinctive failure signature of each of the 10 models evaluated in LinAlg-Bench, drawn entirely from empirical data. Profiles report per-level accuracy across all three dimensions, dominant failure modes, solution strategy behaviour, and the most diagnostically significant finding for each model. No architectural speculation is included; all observations are grounded in the benchmark's error taxonomy and forensic classifications. Profiles are ordered by tier and then by overall 5×5 accuracy within tier.

Level accuracy values are computed as correct responses divided by maximum possible for that level and dimension. Det = Determinant (Recursive level). Eig = Eigenvalue (Compositional level). Dominant failure reports the primary error tags at each dimension (all subcategories pooled). Strategy shares refer to annotated failed determinant responses, the same convention as Appendix J. Full per-model accuracy tables are in Appendix C; strategy classifications are in Appendix J; collapse personas are in Appendix I; the prompt-scaffold study is in Appendix P.

L.1 Gemini-3.1-Pro — Tier 1

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	100	100	100	84.3	93.7	ARITHMETIC, FORMATTING_MISMATCH
4×4	100	100	100	100	81.1	94.6	HALLUCINATION, ARITHMETIC
5×5	100	100	100	100	56.7	88.0	HALLUCINATION

Strategy: Perfect determinant at every dimension; no failed 5×5 determinant responses to classify (Table J.2). 56.7% (68/120) eigenvalue at 5×5 despite 100% determinant — the two tasks represent distinct failure thresholds, and this is the highest 5×5 eigenvalue accuracy in the benchmark. Appendix P shows the capability runs deeper: with a worked example in the prompt, the same model reaches 81.1–100% on 5×5 eigenvalue problems.

Diagnostic signal: Eigenvalue collapse at 5×5 is partial (56.7%) despite perfect determinant — the most direct evidence that Recursive and Compositional are distinct thresholds. The scaffold study (Appendix P) lifts its irrational-spectrum accuracy from 13.3% to 88.9%, confirming the gap is prompt-recoverable for this model alone.

L.2 OPENAI-O1 — TIER 1

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	100	100	100	86.6	94.6	ARITHMETIC
4×4	100	100	100	100	73.2	92.3	ARITHMETIC
5×5	100	100	100	100	32.5	81.4	HALLUCINATION

Strategy: Explicit bisection for eigenvalue root-finding; iterates λ at interval boundaries before converging. No failed 5×5 determinant responses to classify (Table J.2). 5×5 eigenvalue accuracy is 32.5% (39/120), second only to Gemini-3.1-Pro (56.7%); six models score above zero at this level (Gemini-3.1-Pro 68, OpenAI-o1 39, Qwen3-235B 12, DeepSeek-V3 7, GPT-5.2 4, Claude-4.5-Sonnet 2, out of 120). Superior execution raises the working-memory ceiling but does not eliminate it.

Diagnostic signal: Perfect determinant accuracy across all dimensions. 5×5 eigenvalue failures are dominated by fabricated eigenvalue lists (HALLUCINATION, 80% of tagged errors) — the working memory threshold exists even for Tier 1.

L.3 QWEN3-235B — TIER 1

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	88.3	97.5	96	57.5	79.8	ARITHMETIC
4×4	100	93.3	97.5	95.4	42.5	80.5	GENERATION_LOOP, HALLUCINATION
5×5	100	91.7	100	93.7	10	71.5	HALLUCINATION

Strategy: Determinant trajectory is a shallow monotonic decline: 96.0% $\rightarrow$ 95.4% $\rightarrow$ 93.7%. Rank and nullity reach 100% at 5×5 — shared with OpenAI-o1, Gemini-3.1-Pro, and DeepSeek-V3 — consistent with its dual-mode thinking architecture and 36 trillion training tokens providing deep algorithm memorisation at sub-recursive complexity levels. Failure mode shifts from GENERATION_LOOP at 4×4 (27 of 86 tagged errors; it gets trapped testing rational roots for irrational eigenvalue polynomials) to HALLUCINATION at 5×5 (86 of 124) — crossing the abandonment threshold one dimension later than Tier 3 models. MoE architecture with 22B active parameters drops 8.3pp overall from 3×3 to 5×5, the second-smallest decline in the benchmark after Gemini-3.1-Pro (5.6pp).

Diagnostic signal: 100% Sequential at 5×5 and the second-smallest overall decline make Qwen3 a key data point for the training pipeline hypothesis; the failure-mode transition from loop to collapse tracks the working memory account. Prompt scaffolds do not help (Appendix P): its 5×5 eigenvalue accuracy does not improve at any prompt level.

L.4 DEEPSEEK-V3 — TIER 1

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	100	100	100	73.2	89.3	ARITHMETIC
4×4	100	100	100	92	40.9	79.9	ARITHMETIC, SIGN_ERROR
5×5	100	100	100	71.4	5.8	62.5	HALLUCINATION

Strategy: Fails only 50 of 175 determinant problems at 5×5 — too few failures to establish a strategy preference (Appendix J). Forensic inspection reveals systematic zero-seeking before expansion and explicit sub-minor simplification stated in working text, effectively reducing autoregressive depth within cofactor expansion. One of two open-weight models to pass any 5×5 eigenvalue problems (7/120; Qwen3-235B passes 12/120); uses bisection root-finding independently of OpenAI-o1. Degradation is gradual: a 9.4pp drop from 3×3 to 4×4 and 17.4pp from 4×4 to 5×5 — consistent with reasoning chain depth providing graceful degradation at the R1 distillation boundary.

Diagnostic signal: Critical test case for Section 6.2: execution quality partially compensates — 71.4% determinant at 5×5 with gradual degradation — but only to a ceiling (113 of 120 eigenvalue problems fail at 5×5, with HALLUCINATION the dominant tag).

L.5 GPT-5.2 — TIER 2

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	100	100	100	64.6	85.8	ARITHMETIC
4×4	100	100	100	98.9	39.4	82.1	HALLUCINATION
5×5	100	100	95	90.9	3.3	69.2	HALLUCINATION

Strategy: Only 16 failed determinant responses at 5×5 — too few to establish a strategy preference (Appendix J). Format sensitivity at the Recursive level is examined in Appendix D. Apparent tool circumvention noted at 5×5: GPT-5.2 explicitly outputs the no-tool constraint before producing tool-simulated outputs (Appendix I), confirming tool invocation is hallucinated rather than real. API-level inference constraints mean this cannot be definitively attributed to specific implementation choices without access to inference infrastructure.

Diagnostic signal: The widest determinant–eigenvalue gap in the benchmark at 5×5 (90.9% vs 3.3%): the Recursive level is essentially intact while the Compositional level is gone, making GPT-5.2 the clearest single-threshold case.

L.6 MISTRAL-LARGE — TIER 2

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	100	100	100	53.5	81.4	GENERATION_LOOP
4×4	100	100	87.5	94.3	26.8	75.6	SIGN_ERROR, ARITHMETIC
5×5	100	98.3	75	38.3	0	45.1	HALLUCINATION, SIGN_ERROR

Strategy: 94.4% of its failed 5×5 determinant responses are cofactor expansions (Table J.2) — among the most cofactor-locked in the benchmark. Despite this, it achieves 38.3% determinant at 5×5, the highest among the five collapsed cofactor-locked models. Forced Gaussian ablation: 31.8% accuracy (7/22 valid responses) on previously failed problems — the only model to show meaningful partial recovery, consistent with its position at the Tier 2 boundary. MoE architecture with 41B active parameters drops 36.3pp overall from 3×3 to 5×5 — substantially worse than Qwen3-235B at 22B active parameters despite similar architecture, suggesting training pipeline design matters more than parameter count.

Diagnostic signal: Same MoE architecture as Qwen3-235B but roughly twice (41B vs 22B) the active parameters and dramatically worse scale robustness — suggesting training pipeline design matters more than parameter count. Forced Gaussian partial recovery is the clearest execution-quality signal in the ablation.

L.7 CLAUDE-4.5-SONNET — TIER 3

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	100	100	100	50.4	80.1	HALLUCINATION
4×4	100	100	85	83.4	22	69.7	SIGN_ERROR, HALLUCINATION
5×5	100	100	77.5	6.3	1.7	33.1	HALLUCINATION, SIGN_ERROR

Strategy: 76.2% of its failed 5×5 determinant responses are cofactor expansions (Table J.2). Forced Gaussian ablation: 1 of 18 valid responses correct (5.6%; one no-response excluded). Explicitly outputs the no-tool constraint before simulating tool invocation at 5×5 eigenvalue (Appendix I). Arithmetic remains 100% at 5×5 — one of five models with perfect arithmetic accuracy at maximum dimension, confirming that shallow arithmetic is intact while recursive computation collapses. Prompt scaffolds do not help (Appendix P): 0% on 5×5 eigenvalue at every prompt level, even with the full algorithm and a worked example in the prompt.

Diagnostic signal: Most interesting Tier 3 profile because Reading stays at 100% at 5×5 alongside Arithmetic — suggesting input-transcription robustness at scale rather than general degradation. HALLUCINATION dominates failure at 5×5.

L.8 GPT-4o — TIER 3

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	98.3	97.5	98	18.9	66.6	SIGN_ERROR, ARITHMETIC
4×4	97.5	100	62.5	15.4	6.3	36	SIGN_ERROR
5×5	100	96.7	67.5	0	0	28.7	HALLUCINATION

Strategy: 99.4% of its failed 5×5 determinant responses are cofactor expansions (174 of 175; Table J.2) — the most cofactor-locked model in the benchmark. Early sequential collapse: 62.5% at 4×4 Sequential, the second-lowest at that level (above Llama-3.3-70B at 57.5%). Determinant falls to 15.4% already at 4×4 — the earliest and deepest determinant collapse in the benchmark — and reaches 0% at 5×5. Forced Gaussian ablation: 2.1% (1/48 valid) — near-zero, confirming strategy enforcement is irrelevant. Format sensitivity at 5×5 matrix–vector tasks is examined in Appendix D.

Diagnostic signal: Most severe degradation trajectory in the benchmark: Sequential falls to 62.5% and determinant to 15.4% already at 4×4, the deepest determinant collapse at that dimension.

L.9 LLAMA-3.3-70B — TIER 3

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	100	98.3	82.5	100	14.2	63.1	ARITHMETIC, SIGN_ERROR
4×4	97.5	95	57.5	84.6	0.8	60.6	HALLUCINATION, ARITHMETIC
5×5	100	83.3	60	5.1	0	28.3	HALLUCINATION

Strategy: 96.4% of its failed 5×5 determinant responses are cofactor expansions (Table J.2). Anomalous 4×4 profile: 84.6% determinant accuracy despite poor sequential performance (57.5%) — suggesting cofactor expansion bypasses row-operation state tracking. Lowest 3×3 eigenvalue accuracy in the benchmark (14.2%), the earliest Compositional failure signal. Forced Gaussian ablation: 0% across 28 valid responses of 30 unique problems (23 duplicate submissions excluded). Dense model with standard RLHF training; among the most complete collapses at 5×5 (28.3% overall), though Qwen2.5-72B falls lower (24.6%) and both GPT-4o (36.0%) and Qwen2.5-72B (39.8%) are already lower at 4×4.

Diagnostic signal: The 3×3 eigenvalue weakness (14.2%) is the strongest early predictor of 5×5 tier membership in the dataset. The 4×4 determinant anomaly (84.6% despite poor sequential) makes Llama the clearest case of cofactor-only execution masking underlying capability limits.

L.10 QWEN2.5-72B — TIER 3

Dim.	Reading	Arithmetic	Sequential	Det	Eig	Overall	Dominant failure (all subcats)
3×3	97.5	100	95	92	29.1	69.4	HALLUCINATION, ARITHMETIC
4×4	95	91.7	75	26.3	5.5	39.8	SIGN_ERROR
5×5	97.5	78.3	52.5	0	0	24.6	HALLUCINATION, SIGN_ERROR

Strategy: 94.3% of its failed 5×5 determinant responses are cofactor expansions (Table J.2). Early determinant collapse — already 26.3% at 4×4, against 83.4–84.6% for Claude-4.5-Sonnet and Llama-3.3-70B (GPT-4o, at 15.4%, falls harder still). Forced Gaussian ablation: 0% across 46 valid responses — no recovery. Arithmetic drops to 78.3% at 5×5, the most severe arithmetic degradation in the benchmark. Dense model with standard instruction tuning.

Diagnostic signal: Early determinant collapse (4×4: 26.3%) combined with arithmetic degradation at 5×5 makes Qwen2.5-72B the clearest case of generalised computational overload rather than recursive-only failure.

L.11 CROSS-MODEL DIAGNOSTIC SUMMARY

The following table summarises the key diagnostic signal for each model — the single most informative finding for mechanistic interpretability follow-up.

Cross-model diagnostic summary. All observations are empirical. Architectural hypotheses generated by these observations are framed as falsifiable conjectures for follow-up mechanistic work and are not confirmed claims.

Model	Tier	Primary diagnostic signal
Gemini-3.1-Pro	T1	Highest 5×5 eig (56.7%) alongside 100% det; scaffolding closes the remaining gap (Appendix P)
OpenAI-o1	T1	Perfect det all dims; 5×5 eig failures dominated by `HALLUCINATION` — the working memory ceiling exists even for Tier 1
Qwen3-235B	T1	Shallow monotonic det decline ($96.0 \rightarrow 95.4 \rightarrow 93.7$); 100% Sequential at 5×5; failure mode shifts from loop to collapse across dimensions
DeepSeek-V3	T1	71.4% det at 5×5 with gradual degradation; execution quality compensates to a ceiling
GPT-5.2	T2	Widest 5×5 det–eig gap (90.9% vs 3.3%): Recursive intact, Compositional gone; format sensitivity examined in Appendix D
Mistral-Large	T2	Highest forced Gaussian partial recovery (31.8%); MoE with 41B active params — worse than Qwen3 at 22B
Claude-4.5-Sonnet	T3	Reading and Arithmetic both 100% at 5×5 — input-transcription robustness at scale, not general degradation; prompt scaffolds give 0% (Appendix P)
GPT-4o	T3	Earliest and deepest det collapse (4×4: 15.4%) with early Sequential collapse (4×4: 62.5%); format sensitivity examined in Appendix D
Llama-3.3-70B	T3	Lowest 3×3 eig (14.2%) is the strongest early predictor of 5×5 tier; 4×4 det anomaly (84.6%) masks underlying limits
Qwen2.5-72B	T3	Early det collapse (4×4: 26.3%); arithmetic degradation at 5×5; clearest case of generalised computational overload

M Run-to-Run Stability

All headline results are reported from a single evaluation pass at temperature 0. To bound the noise attributable to a single pass, a fixed 660-problem subset of the benchmark was evaluated three times — the reported run plus two independent complete reruns — for all ten models, under the identical zero-shot prompt and grading pipeline (Appendices A and B). This appendix reports per-model accuracy across the three runs and verifies that the 5×5 regime shift is present in every run.

M.1 Per-Model Accuracy Across Three Runs

Table M.1: Overall accuracy (%) on the 660-problem three-run set across three complete runs at temperature 0. Mean $|\Delta|$ is the average absolute deviation of the two reruns from the reported run, in percentage points.

Model	Reported (%)	Rerun 1 (%)	Rerun 2 (%)	Mean $\lvert\Delta\rvert$ (pp)
OpenAI-o1	94.7	92.9	92.4	2.1
Gemini-3.1-Pro	93.2	91.4	91.4	1.8
DeepSeek-V3	89.8	87.7	88.2	1.9
GPT-5.2	89.5	89.5	90.2	0.4
Qwen3-235B	87.1	87.1	85.3	0.9
Mistral-Large	83.2	81.2	81.7	1.8
Claude-4.5-Sonnet	79.7	79.8	80.3	0.4
Llama-3.3-70B	71.4	70.6	69.7	1.3
GPT-4o	70.5	67.9	68.5	2.3
Qwen2.5-72B	67.7	67.3	67.3	0.4

Across all ten models the mean absolute deviation from the reported run is 1.3 percentage points; the largest single deviation is 2.6pp (GPT-4o, rerun 1). Rankings are stable: OpenAI-o1 and Gemini-3.1-Pro finish first and second in every run, and the only ordering change anywhere in the table is the pair DeepSeek-V3 / GPT-5.2, separated by 0.3pp in the reported run — within run-to-run noise.

M.2 Stability of the 5×5 Regime Shift

The regime shift is present in every run: reading and arithmetic stay near ceiling, sequential tasks degrade mildly, determinant collapses to under half, and eigenvalues collapse to near zero. In each rerun, no more than two 5×5 eigenvalue problems are answered correctly across all ten models combined. The reported results are not an artifact of a single sample.

Table M.2: Accuracy (%) at 5×5 by task level, aggregated over all ten models, in each of the three runs.

Task level	Reported (%)	Rerun 1 (%)	Rerun 2 (%)
Reading	99.8	99.2	99.8
Arithmetic	94.8	92.0	92.2
Sequential	82.8	79.2	76.0
Determinant	47.0	48.8	49.2
Eigenvalues	1.3	0.3	0.7

M.3 Protocol Notes

All three runs used the same fixed 660-problem set, spanning every task level at all three sizes: the complete problem set for the seven other task levels (420 problems), plus 150 determinant problems (50 per size) and 90 eigenvalue problems (30 per size). Model snapshots, prompts, and the grading pipeline were held fixed across all three runs.

N The Operation-Count Control — Depth, Not Volume

A reader of Appendix I may reasonably object that the eigenvalue collapse is a matter of sheer volume: a 5×5 eigenvalue computation involves several hundred arithmetic operations, and perhaps models simply fail once the total amount of arithmetic grows large, however that arithmetic is organized. This appendix reports the control experiment that separates the amount of arithmetic from its structure. The result, in one sentence: at the same operation budget, the same models range from 0% to 100% depending on how much intermediate state the steps force them to carry — so volume cannot be the cause.

N.1 Design: Matching the Arithmetic Budget

The control task is a chain of matrix–vector multiplications. Starting from an integer vector v_0, the model applies a fixed 5×5 integer matrix k times in sequence and reports the final vector (matrix entries drawn from $\{-2, -1, 1, 2\}$; v_0 entries from $[-3, 3]$; temperature 0; fifteen chains per model at each length; grading is exact integer match on the final boxed vector). Each individual step is easy — twenty-five multiplications and twenty additions — and the state carried from one step to the next is only five numbers. A chain therefore accumulates arithmetic volume without accumulating entangled intermediate state.

Chain length k is the dial. At $k = 11$ a chain totals 495 elementary operations, matching or exceeding the operation count of one full 5×5 eigenvalue solve; that rung is the point of comparison throughout. (At $k \geq 19$ intermediate entries can exceed 10^{10}; the volume argument rests on the $k = 11$ rung, where values remain near 10^6.) Nine of the ten benchmark models were tested. OpenAI-o1 was not run: its hidden chain of thought is a computational aid that can be neither observed nor controlled.

N.2 Chain Results: Where Each Model's Depth Runs Out

Table N.1 reports the five non-reasoning models. Each entry is chains solved over chains run. Reading left to right: every model is at or near a perfect score at $k = 1$ — a single matrix–vector multiplication — and every model is at zero by $k = 11$, the eigenvalue-equivalent budget. The collapse points differ along the way (Qwen2.5-72B fades by $k = 4$; Mistral-Large still solves fourteen of fifteen at $k = 5$) but the destination is shared: at the rung matching one eigenvalue computation's arithmetic, no non-reasoning model solves a single chain.

Table N.2 reports the four reasoning models over longer chains. All four hold 77–100% at $k = 11$ — the budget at which the non-reasoners sit at zero, and at which these same four models score between 1.7% and 56.7% on 5×5 eigenvalue problems. Their limits lie much further out, and differ in character: Claude-4.5-Sonnet stays above 85% through $k = 16$ and reaches zero by $k = 26$; Qwen3-235B degrades gradually through $k = 19$ and collapses at $k = 23$, past which no longer

Table N.1: Chain accuracy for non-reasoning models (chains solved / chains run; temperature 0; source: op-count_results_raw.csv). Cells with fewer than fifteen runs come from early smaller batches; shaded cells are zero.

Model	k=1	k=2	k=3	k=4	k=5	k=6	k=11
Qwen2.5-72B	11/11	8/13	5/8	1/7	0/10	0/11	0/8
Llama-3.3-70B	15/15	13/15	13/15	10/15	6/15	2/13	0/12
GPT-4o	15/15	13/15	10/15	12/15	9/15	4/15	0/15
DeepSeek-V3	15/15	15/15	14/15	11/14	11/15	4/13	0/13
Mistral-Large	15/15	15/15	12/15	14/15	14/15	7/15	0/15

lengths were scheduled; GPT-5.2 degrades but never fully collapses, still solving 40% at $k = 26$; Gemini-3.1-Pro remains at 73% at the longest length tested.

Table N.2: Chain accuracy for reasoning models. — = length not scheduled for that model: chain lengths were extended per model only while accuracy held up; once a model fell below 40%, at most one longer length was run to confirm the floor and the ladder was stopped there. Run counts above fifteen pool repeated batches.

Model	k=6	k=11	k=13	k=16	k=19	k=23	k=26
Gemini-3.1-Pro	15/15	15/15	—	14/15	—	11/15	—
Qwen3-235B	13/13	10/13	12/15	11/15	9/15	2/28	—
Claude-4.5-Sonnet	15/15	13/15	—	14/15	7/15	2/15	0/15
GPT-5.2	15/15	12/15	—	8/15	6/15	5/15	6/15

N.3 The Matched-Budget Comparison: Chains, Determinants, Eigenvalues

Table N.3 places three tasks side by side at comparable arithmetic budgets. A $k = 11$ chain costs about 500 operations but carries only five numbers between steps. A 5×5 determinant by cofactor expansion costs about 320 operations but forces one deeply nested recursion — every 4×4 minor feeds the same final sum, so intermediate results must all be held and combined. A 5×5 eigenvalue problem costs the most: it contains the determinant computation in symbolic form (on $A - \lambda I$) and adds root-finding on the result. If total arithmetic were what breaks models, the three columns would rise and fall together.

Table N.3: Accuracy at matched arithmetic budgets. Chain column: $k = 11$, of 15 chains. Determinant column: all 5×5 determinant problems, of 175. Eigenvalue column: official basis, of 120 (Appendix I, Table I.1). † OpenAI-o1 solved all 16 determinant problems it attempted; remaining problems follow the declared-pass protocol (Appendix C).

Model	**Chain,** k=11 **(≈500 ops)**	5×5 **determinant (≈320 ops)**	5×5 **eigenvalue**
Gemini-3.1-Pro	100.0%	100.0%	56.7%
OpenAI-o1	—	100.0% †	32.5%
Qwen3-235B	76.9%	93.7%	10.0%
DeepSeek-V3	0.0%	71.4%	5.8%
GPT-5.2	80.0%	90.9%	3.3%
Claude-4.5-Sonnet	86.7%	6.3%	1.7%
Mistral-Large	0.0%	38.3%	0.0%
GPT-4o	0.0%	0.0%	0.0%
Llama-3.3-70B	0.0%	5.1%	0.0%
Qwen2.5-72B	0.0%	0.0%	0.0%

They do not. At a fixed arithmetic budget, outcomes span 0–100% in both directions, and arithmetic volume predicts none of it:

- **Reasoning models pass chains and determinants, yet fail eigenvalues.** Qwen3-235B solves 93.7% of 5×5 determinants but 10.0% of 5×5 eigenvalue problems; GPT-5.2, 90.9% against 3.3%. The binding failure is not the determinant sub-computation but what

the eigenvalue task adds on top of it — carrying a symbolic parameter through that same recursion, then solving the resulting quintic.

- **More arithmetic with shallow state succeeds; less arithmetic with entangled state fails.** Claude-4.5-Sonnet solves 86.7% of $k = 11$ chains (about 500 operations) yet only 6.3% of numeric determinants (about 320 operations). The chain is more work but shallow; the determinant is less work but entangled.
- **The dissociation runs the other way too.** DeepSeek-V3 scores 0% on chains at $k = 11$ but 71.4% on determinants — its chains fail through accumulated slips over a long repetitive output, not through state overload. No volume account survives dissociations in both directions.

N.4 What This Rules Out

The operation-count objection fails twice over. Models that score essentially zero on 5×5 eigenvalue problems complete chains of the same arithmetic size at 80–100%, and models that handle the determinant — the largest arithmetic block inside the eigenvalue computation — still fail the eigenvalue task built on it. What separates success from failure at a fixed budget is the depth of entangled intermediate state the task forces the model to hold. Two companion experiments close the remaining escape routes: the IBP depth ladder (Appendix O) shows the same collapse law in a different mathematical domain, and the scaffold gradient (Appendix P) shows that spelling out the algorithm does not repair it.

O The Integration-by-Parts Depth Ladder — The Same Collapse Outside Linear Algebra

Appendix N showed that the eigenvalue collapse is not about the amount of arithmetic. A reader may still object that it is about linear algebra itself — something peculiar to matrices. This experiment reruns the depth test in a different subject entirely: first-year calculus. The problems below contain no matrix, no determinant, and no eigenvalue, yet they produce the same collapse, in the same model order.

O.1 The Task, From the Ground Up

- **One problem, one integral.** Every problem is one integral of the form $\int x^n \sin(bx)\,dx$ or $\int x^n \cos(bx)\,dx$, where n runs from 0 to 16 and b is a whole number from 1 to 10. Example: $\int x^2 \cos(2x)\,dx$.
- **One method, one step at a time.** Integration by parts is the standard classroom method, and each application of it peels exactly one power of x off the problem: it turns $\int x^n \sin(bx)\,dx$ into a known piece plus $(n/b)\int x^{n-1}\cos(bx)\,dx$. One application lowers n by one, swaps sine for cosine, and adds one coefficient to keep track of.
- **n is a depth dial, and nothing else.** So n is a depth dial. A problem with $n = 2$ takes about three chained steps; $n = 10$ takes about eleven; $n = 16$ takes about seventeen. Every step is the same kind of easy move — differentiate a power, integrate a sine or cosine, keep one sign and one factor of $1/b$ straight. Raising n adds more steps of identical difficulty; it never makes any single step harder.
- **Depth means carried state.** What grows with n is the state the model must carry. The final answer has the form (polynomial) $\times \sin(bx)$ + (polynomial) $\times \cos(bx)$, and every coefficient in those polynomials depends on every earlier step of the chain. One slip anywhere corrupts everything after it. This is the same burden the eigenvalue task imposes — a long chain of entangled intermediate values — but built out of calculus instead of matrices.

O.2 Setup

- 340 problems: 17 levels ($n = 0$ through 16), 20 problems per level — 10 with sine, 10 with cosine — and b spread over 1 to 10.

- Levels $n = 0$ and $n = 1$ are sanity rungs (one or two steps; everything should pass). The ladder proper is $n = 2$ through 16.
- Four models were run — Gemini-3.1-Pro, Claude-4.5-Sonnet, Qwen3-235B, Llama-3.3-70B — at temperature 0, each problem twice: 680 calls per model, 2,720 in total, all completed with zero API errors.
- Every ground-truth answer was verified before any model was called: each of the 340 answers was differentiated symbolically and checked to return exactly the original integrand.
- Grading is symbolic, not string matching. The grader takes the model's final boxed answer, differentiates it, and subtracts the integrand; a pass requires the difference to be exactly zero. Any correct form passes, no matter how it is written.
- Models with hidden chains of thought (o1, o3, DeepSeek-R1) are excluded: a hidden scratchpad is an uncontrolled depth aid, and this experiment measures depth.
- Clean subset. Responses whose boxed answer the symbolic grader could not read at all — no boxed answer, or LaTeX that does not parse — are removed from both the numerator and the denominator, so every accuracy is over readable answers only. Removed per model, of 680 calls: Claude 3, Gemini 14, Llama 104, Qwen3 331.

O.3 Results: Accuracy Down the Ladder

Every cell is one number: the accuracy, in percent, at that depth. Each level has 20 problems, each run twice, so a cell rests on up to 40 answers (20 problems $\times$ 2 repeats). Accuracy is computed over readable answers only: responses the symbolic grader could not read were removed before scoring (of 680 calls per model — Claude 3, Gemini 14, Llama 104, Qwen3 331). A star (*) marks a cell resting on fewer than 10 readable answers, where the percentage is unstable; a dash means no readable answers at that level. Shaded cells are below 50%.

Table P.1a: Accuracy (%) by depth, $n = 0$–8. Up to 40 answers per cell (20 problems $\times$ 2 repeats); accuracy over readable answers only; * = fewer than 10 readable answers; temperature 0; source: ibp_results_raw.csv.

Model	n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8
Gemini-3.1-Pro	100	100	100	100	100	100	100	100	100
Claude-4.5-Sonnet	100	100	100	100	98	95	85	80	60
Qwen3-235B	100	100	100	100	100	88	88	92	86
Llama-3.3-70B	100	98	98	95	75	55	51	29	12

Table P.1b: Accuracy (%) by depth, $n = 9$–16 (same conventions as Table P.1a).

Model	n=9	n=10	n=11	n=12	n=13	n=14	n=15	n=16
Gemini-3.1-Pro	100	100	100	97	64	32	8	13
Claude-4.5-Sonnet	55	38	35	45	51	46	32	23
Qwen3-235B	100	100*	60	0*	50*	0*	0	40
Llama-3.3-70B	26	19	32	17	39	7	22	20

O.4 What the Ladder Shows

- The sanity rungs pass everywhere. All four models are at or near perfect at $n = 0$ and $n = 1$ — 100% in every cell except one dropped problem for Llama-3.3-70B at $n = 1$ (39 of 40). The task is understood, the format is followed, one or two steps are no obstacle.
- Each model then collapses at its own depth. Reading the tables for the level where accuracy first falls below 50%: Llama-3.3-70B around $n \approx 5$, Claude-4.5-Sonnet around $n \approx 9$, Qwen3-235B around $n \approx 11$–13, Gemini-3.1-Pro around $n \approx 13$–14.
- **The ranking transfers across domains.** That order — Gemini and Qwen3 deepest, Claude next, Llama first to fall — is the same order these models finish in on the 5×5 eigenvalue benchmark, and the same order their chains collapse in Appendix N. Three tasks, three domains, one ranking.

- Failures look like depth failures, not confusion. Failing answers do not refuse or ramble: they keep the correct overall shape — a polynomial times sine plus a polynomial times cosine — and their leading coefficients are right. The errors appear midway down the chain, exactly where accumulated factors from earlier steps must be carried through later ones.
- One caution on thin cells. Removing unreadable responses leaves some Qwen3-235B cells at high n with very few survivors (as few as 2 of 40). Percentages in those cells are unstable; each such cell is marked with a star (*) so the reader can weigh it accordingly.
- Secondary analyses (logistic regression separating depth from coefficient size, and curves split by low and high b) are planned and marked NEEDS_DATA in the working draft.

O.5 What This Rules Out

The objection this appendix answers is "the collapse is a linear-algebra quirk." It is not. A depth dial made of nothing but powers of x and one trigonometric factor reproduces the eigenvalue collapse: perfect performance on shallow problems, a model-specific cliff, and failure driven by the length of the chain of entangled intermediate values — not by the difficulty of any single step, and not by anything specific to matrices. Together with the operation-count control (Appendix N) and the scaffold gradient (Appendix P), it leaves depth of state as the only surviving explanation.

P Scaffold Gradient Study — Does Better Prompting Fix 5×5 Eigenvalue Failure?

A scaffold is extra help written into the prompt — anything from a simple formatting instruction up to a complete, step-by-step statement of the algorithm the model should run. Levels L2 and L3 are chain-of-thought prompting in its standard form — the prompt spells out the reasoning steps the model should follow — and L4 goes further, writing out the complete algorithm. This appendix asks a single question: when a model fails 5×5 eigenvalue problems, can a better prompt fix it? The question matters because "just prompt it better" is the most natural objection to any claimed capability limit: if a stronger prompt could restore accuracy, the failures documented in this paper would be a prompting problem, not a capability limit.

P.1 Experimental design

- Models: Claude-4.5-Sonnet, Gemini-3.1-Pro, Qwen3-235B, run in the same standard configuration as the main benchmark, with their working written out in the visible response. Dedicated reasoning models (e.g., o1, DeepSeek-R1) are excluded: their long hidden reasoning pass is itself a scaffold that cannot be controlled across prompt levels.
- Problems: four pools (sets) of 5×5 eigenvalue problems from the main benchmark. In the two integer pools the correct eigenvalues are whole numbers; in the two irrational pools they are non-terminating decimals. Each is split into symmetric and non-symmetric matrices (30, 30, 15 and 15 problems).
- Prompt levels: L0 — the unchanged zero-shot benchmark prompt; L1 — the same task plus a demand for decimal answers (4 decimal places, no symbolic form); L2 — a three-step strategy hint (form $\det(A - \lambda I)$, expand, find the roots numerically) plus the decimal demand; L3 — an explicit five-step cofactor procedure; L4 — the complete Faddeev–LeVerrier algorithm, the strongest scaffold in the gradient, leaving the model no step to plan on its own.
- Protocol: temperature 0, 3 repeats per condition, 4,320 calls in total.
- Grading: locked before the first call — the boxed answer must be a list of decimals matching ground truth within the benchmark tolerance.
- Ceiling: Gemini already solves every integer-pool problem with the plain L0 prompt (100%). A perfect score leaves no room for a scaffold to show an effect, so these cells are excluded from the statistical tests (†).

Table P.1: Mean 5×5 eigenvalue accuracy (%) $\pm$ range over 3 repeats, by model, problem pool and scaffold level. † Gemini integer cells are at ceiling at L0 and excluded from hypothesis tests at all levels.

Model	Pool (5×5)	L0	L1	L2	L3	L4
Gemini-3.1-Pro	Irr, sym	3.3	86.7 $\pm$6.7	88.9 $\pm$6.7	81.1 $\pm$3.3	90.0 $\pm$10.0
	Irr, non-sym	0.0	88.9 $\pm$6.7	86.7 $\pm$6.7	90.0	87.8 $\pm$3.3
	Int, sym	100.0	100.0	100.0	97.8 $\pm$6.7	100.0 †
	Int, non-sym	100.0	100.0	95.6 $\pm$6.7	93.3	100.0 †
Claude-4.5-Sonnet	Irr, sym	0.0	0.0	0.0	0.0	0.0
	Irr, non-sym	0.0	0.0	0.0	0.0	0.0
	Int, sym	0.0	2.2 $\pm$6.7	0.0	0.0	0.0
	Int, non-sym	6.7	0.0	0.0	0.0	0.0
Qwen3-235B	Irr, sym	0.0	0.0	0.0	0.0	0.0
	Irr, non-sym	0.0	0.0	0.0	0.0	0.0
	Int, sym	6.7	0.0	6.7	0.0	4.4 $\pm$6.7
	Int, non-sym	33.3 $\pm$13.3	0.0	0.0	2.2 $\pm$6.7	20.0 $\pm$13.3

P.2 Results

Chain-of-thought prompting could not do much: the only model that improved was the one already strongest on the task, and the other two stayed at or near 0% at every level.

- Gemini-3.1-Pro is already the strongest model on these tasks in the main benchmark — 56.7% on 5×5 eigenvalue (the highest of the ten models) and 100% on 5×5 determinant. It can do the mathematics, and prompting let it show that: 100% on the integer pools at every level, and 81.1–90.0% on the irrational pools from L1 through L4.
- Claude-4.5-Sonnet solved nothing at any level: 0% on the irrational pools from L0 through L4, including L4, where the complete Faddeev–LeVerrier algorithm is written into the prompt. Its L4 responses follow the requested procedure and produce well-formatted decimal lists — the values are simply wrong.
- Qwen3-235B behaved the same: 0% on the irrational pools at every level. On its one non-zero pool (integer non-symmetric), scaffolds even hurt — 33.3% at L0 falls to 0% under L1–L2 and recovers only partially to 20.0% at L4.
- Takeaway: prompting helped only the model that could already do the mathematics. For the models that fail, no prompt in the gradient — up to and including the complete algorithm — moved them off 0%. This is the paper's central result seen from another angle: the barrier is carrying a long chain of dependent computation, and no prompt supplies that.

P.3 Comparison with the main benchmark

Table P.2 places the scaffold results next to each model's zero-shot accuracy in the main benchmark, split by spectrum. Out of six model $\times$ spectrum cells, prompting improved exactly one — Gemini on irrational spectra. Gemini was already flawless on integer spectra, and for Claude and Qwen scaffolding helped nowhere; on integer spectra it made both slightly worse.

Table P.2: Main-benchmark 5×5 eigenvalue accuracy (zero-shot) versus the best accuracy reached at any scaffold level L1–L4, by model and spectrum type. Benchmark cells contain 60 problems each; scaffold values are means over 3 repeats, pooled across the symmetric and non-symmetric pools (60 irrational, 30 integer problems) drawn from the same problem families.

Model	Spectrum	Benchmark (zero-shot, %)	Best with scaffold (%)	Did prompting help?
Gemini-3.1-Pro	Integer	100.0	100.0 (all levels)	Nothing to improve
	Irrational	13.3	88.9 (L4)	Yes — the one gain
Claude-4.5-Sonnet	Integer	3.3	1.1 (L1)	No — slightly worse
	Irrational	0.0	0.0	No
Qwen3-235B	Integer	20.0	12.2 (L4)	No — worse
	Irrational	0.0	0.0	No

The conclusion matches Appendix K from the opposite direction. Appendix K forced a better algorithm onto models that fail 5×5 determinants, and accuracy did not recover; here, handing models the complete eigenvalue algorithm does not recover accuracy either. Better prompts did not repair execution. The bottleneck is the model's ability to carry a long chain of dependent intermediate values without error, and no amount of instruction supplies that.

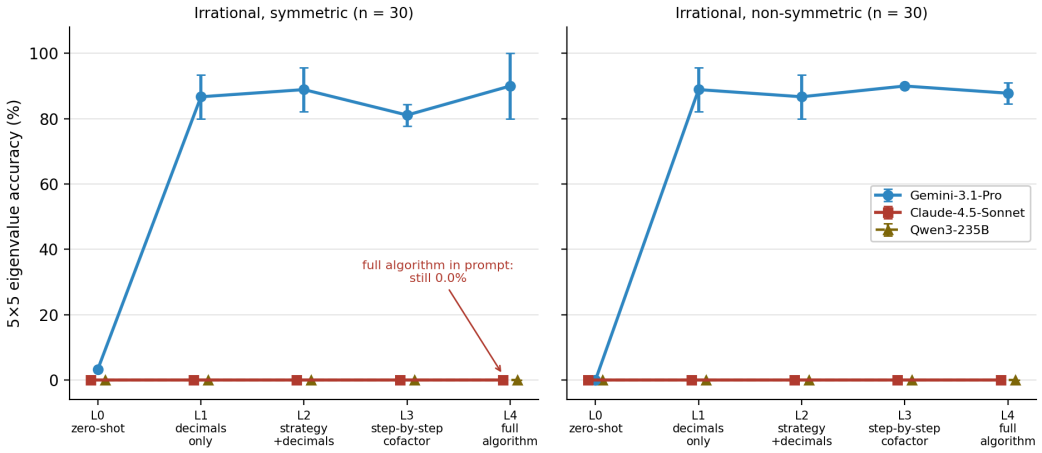

Figure P.1: Mean 5×5 eigenvalue accuracy ($\pm$ range over 3 repeats) across the scaffold gradient on the two irrational pools. Gemini-3.1-Pro (blue) reaches $\sim$87–90% from L1 onward and holds it through L4. Claude-4.5-Sonnet (red) and Qwen3-235B (dashed) remain at 0% at every level, including L4, where the complete algorithm is written into the prompt.