

The Architecture of Errors: From Universal Impossibility to Patch-Local LLM Reliability

Mikhail L. Arbuzov
Independent Researcher
mike.arbuzov54@gmail.com

Sisong Bei
Independent Researcher
qurining@gmail.com

Ziwei Dong
Independent Researcher
ziwei.dong@alumni.emory.edu

Dmitri Kalaev
Independent Researcher
kalaevdr@gmail.com

Alexey Shvets
Palo Alto Networks
ashvets@paloaltonetworks.com

Abstract

Reliability is the implicit goal of context management — what a model retrieves, remembers, and is scaffolded with is chosen so it fails less — yet it is usually analysed asymptotically, as if each token compounded the risk. We argue the object to track is not raw sequence length but a small, local catalogue of recurring failure modes. Universal LLM reliability is not a finite-library problem: across all possible tasks, tools, schemas, knowledge sources, and evaluator expectations, new intervention-distinguishable failure modes can appear without bound, so no finite intervention dictionary can guarantee bounded residual error for every such mode. But deployed systems do not operate over the whole universe. They operate inside operationally bounded patches (legal review, medical RAG, code repair, customer-support agents, contract extraction) with recurring tasks, schemas, tools, and evaluator expectations—the operational envelope that the context scaffold of retrieval, memory, tools, and orchestration defines and maintains. Within such patches, empirical evidence suggests failures are sparse, repetitive, and concentrated in a small recurring catalogue, so reliability becomes a local catalogue-discovery and intervention-coverage problem rather than an exponential token-length problem. We formalize this transition with two propositions and one corollary. Proposition 1 is the worst-case-mode-wise negative result: no finite intervention dictionary covers every distinguishable failure mode of an unbounded domain. Corollary 1 is the inverse-discovery implication: the logarithmic upper bound on mode discovery cannot accommodate linearly more distinct tail modes without exponentially more observed hard-failure events. Proposition 2 is the positive patch-local result: under log active-mode exposure and head-heavy coverage, a sufficient per-hard-decision intervention budget grows polylogarithmically in sequence length and becomes domain-constant once the patch catalogue saturates. The framework relocates rather than dissolves long-context difficulty: where the number of hard decisions itself grows with task length, reliability remains hard; the contribution is to identify the on-axis intervention rather than to make those regimes easy.

1 Introduction

The standard worry about long-context generation is exponential: if every token has independent error probability e , the chance of a fully correct n -token output is $(1 - e)^n$, collapsing

to zero for any nontrivial e (LeCun, 2023; Dziri et al., 2023). Prior work (Arbuzov et al., 2025) argued this is misplaced because errors are not uniform across tokens: only 5–10% are “key” (dependent on long-range context (Fang et al., 2025)), the rest near-deterministic once context accumulates. The two-rate model $P(\text{correct}) = (1 - e_{\text{key}})^k (1 - e_{\text{non}})^{n-k}$ with $k \ll n$ sublinear in n recovers long-context coherence and converts the question from “how does n grow?” to “how does k grow?”.

From sparse tokens to recurring patterns. This paper takes the next step. Sparsity tells us *where* errors live; the follow-up is *what* they are. Recent failure-mode atlases answer: errors are not only sparse but *repetitive* — ErrorAtlas (Ashury-Tahan et al., 2026) sorts failures into 17 head-concentrated categories, two error types cover 86.35% of HumanEval (Wen et al., 2024), MWPEs-300K categorises 304,865 math errors (Sun et al., 2025), and multi-hop QA (Zhang et al., 2026), agentic tool use (Cemri et al., 2025), and RAG (Wood & Forbes, 2024) show the same. This suggests a third architectural layer beyond key-token sparsity (Layer 1) and within-key manifold structure (Layer 2; Arbuzov et al. 2025): within the key tokens only a fraction β produce *hard* failures, and inside bounded patches those failures cluster into a finite or effectively capped catalogue whose size grows much more slowly than the number of observed events.

Contributions. The central contribution is a shift in the reliability object. Universal LLM reliability is not a finite-library problem, but patch-local reliability can be treated as catalogue discovery and intervention coverage. We formalise this transition with two propositions and one corollary.

Proposition 1 (negative) says no finite dictionary covers every intervention-distinguishable mode of an unbounded domain. Corollary 1 (inverse-discovery) says the logarithmic bound cannot accommodate linearly more tail modes without exponentially more observed failures, so open-domain tail discovery has diminishing returns. Proposition 2 (positive) says that inside a fixed patch, under log active-mode exposure and head-heavy coverage, a sufficient per-hard-decision budget satisfies $m \geq \lceil |C_{\text{eff}}|^{1-\varepsilon/e_{\text{hard}}} \rceil$ — doubly-logarithmic in sequence length pre-cap, domain-constant once the catalogue saturates. The sequence-level analogue is strictly tighter.

Around this transition the paper does four supporting things. It frames reliability in three layers — sparsity (α), hard-token stratification (β), and patch-local mode catalogue ($|C_D|$) — separating *where* errors occur, *what* forms they take, and *which* interventions address them (§3). It states logarithmic mode discovery as an empirical postulate, not a theorem, calibrated against ErrorAtlas, HumanEval, and MWPEs at $\sigma \in [0.87, 1.85]$ ($\sigma \approx 1.85$ as a conservative planning value). It synthesises ~ 60 published results for clustering, cluster-selective interventions, and sublinear length scaling, including a six-axis harvest of 28 quantitatively-anchored citations (Appendix B). And it re-audits the most-cited steep-decay counter-evidence (Appendix C), showing it decays over task-structure variables — compositional graph size, fact count, log-time horizon, capacity threshold, evidence scope — rather than raw token length.

Relevance to context management. The reframing for this workshop is direct: a deployment patch is an operational context envelope, and the levers a context-management system pulls — what to retrieve, what to keep in memory, which tools and validators to attach, how to orchestrate multi-turn and multi-session state — determine which failure modes in C_D are reachable and which are covered. The budget below bounds how much reliability a well-managed context buys inside a fixed patch; Corollary 1 bounds the cost of discovering the catalogue the scaffold must cover.

2 Related Work

Failure-mode taxonomies. ErrorAtlas (Ashury-Tahan et al., 2026), MWPEs-300K (Sun et al., 2025), the HumanEval categorisation (Wen et al., 2024), and RFMDataset (Guo et al., 2025) together suggest that at any corpus size the number of named failure modes is small

(typically 8–20) with high top-mode coverage; domain taxonomies for multi-hop QA (Zhang et al., 2026), multi-agent systems (Cemri et al., 2025), and tool agents (Yao et al., 2024) report the same.

Targeted interventions. Each named cluster has a focused countermeasure — Python execution, constrained decoding, execution feedback, process reward models, RAG, preference optimisation, and structured tool-call uncertainty (Gao et al., 2023a; Chen et al., 2023; Gou et al., 2024b; Suresh et al., 2025; Wang et al., 2025b; Dong et al., 2025; Shinn et al., 2023; Huang et al., 2023; Wang et al., 2024b; Lightman et al., 2024; Wood & Forbes, 2024; Karaman et al., 2024; Suri et al., 2025) — quantified in §4. Two structural facts recur: each intervention is *cluster-selective* (residuals land in a different cluster) and *additivity is approximate* (Patel et al., 2026; Le, 2026).

Length-decay benchmarks. A parallel literature measures the decay curve’s *shape*. Mild-decay results (Loong (Wang et al., 2024a), GSM- ∞ (Zhou et al., 2025), RULER (Hsieh et al., 2024), anchor LLMs (Pang et al., 2024)) report log-linear, sigmoidal, or threshold decay inconsistent with smooth $(1 - \epsilon)^n$; steep-decay results (Dziri et al., 2023; Kuratov et al., 2024; Kwa et al., 2025; Wan et al., 2026) cited as exponential-compounding evidence each, on re-audit, decay over a variable distinct from raw token length (Appendix C).

Gap. What is missing is a quantitative bridge between the small, repeating taxonomy catalogue and a polylog intervention budget. We provide that bridge in §3.

3 Theoretical Framework

Four levels of $|C|$. LLM error analysis routinely conflates four objects we keep apart: **L1** failure events (raw observed errors); **L2** empirical taxonomy categories (researcher labels grouping L1, e.g. ErrorAtlas’s 17 categories or HumanEval’s AssertionError/NameError); **L3** latent failure modes (the unobserved clusters L2 approximates); and **L4** capability axes / interventions (the engineering unit — a Python interpreter, a constrained decoder, a retrieval-augmented generator). Throughout this section $|C|$ is the L2 count, which is what published taxonomies report. Postulate 1 is *engineering-relevant* because L4 is coarser than L2 (one capability axis sweeps several L2 categories at once; §4, Claim B), and *epistemically* conditional because L2 is only a noisy proxy for the L3 catalogue, whose faithfulness is unmeasured.

Roadmap. The framework separates three questions often conflated: *where* errors occur (a sparse subset of key decisions), *what* recurs (a local catalogue of failure modes), and *what fixes them* (a smaller library of capability interventions). The two propositions of §3.4 formalise the transition from universal impossibility to patch-local tractability.

3.1 β -stratification of key tokens

The two-rate model of Arbuzov et al. (2025) distinguishes k key tokens (error rate e_{key}) from $n - k$ non-key tokens ($e_{\text{non}} \ll e_{\text{key}}$). Empirical atlases suggest that even within the key-token class errors are not uniformly distributed: most key tokens are “decisions” for which the model has stable representations; only a fraction concentrate the actual failures.

Definition 1 (Hard fraction). Partition the k key tokens of a sequence into easy and hard subsets,

$$k_{\text{hard}} = \beta k, \quad k_{\text{easy}} = (1 - \beta)k, \quad \beta \in (0, 1),$$

where hard key tokens have an elevated error rate e_{hard} corresponding to manifold-transition decisions in the sense of Arbuzov et al. (2025, §3.2), and easy key tokens have $e_{\text{easy}} \approx e_{\text{non}}$.

Under Definition 1, the composed sequence-level reliability becomes

$$P(\text{correct}) = (1 - e_{\text{hard}})^{\beta k} (1 - e_{\text{easy}})^{(1-\beta)k} (1 - e_{\text{non}})^{n-k}. \quad (1)$$

We do not assume iid token errors: the three rates are *conditional per-decision hazards*, and grouping by stratum gives the multiplicative survival expression. Since $e_{\text{easy}} \approx e_{\text{non}}$ once context accumulates, the rate is dominated by the βk hard tokens, so e_{hard} is load-bearing; β is *latent* (atlases report $\Pr(\text{category} \mid \text{error})$, not $\Pr(\text{hard} \mid \text{key decision})$), so we carry it symbolically.

3.2 Empirical postulate: logarithmic mode discovery

Stratification names the target — the βk hard-token decisions — but not how many distinct ways they fail. Failures repeat (§1): the question is how the catalogue grows with observed failures, since that is what an intervention library must keep up with.

Two type-token candidates exist: Heaps’ law (power-law, $|C| \approx K k_{\text{hard}}^b$, $b \in [0.4, 0.6]$) (Manning et al., 2008)) and logarithmic discovery. **Zipfian rank-frequency does not imply logarithmic growth** — Heaps’ law is the correct type-token consequence of Zipf, and it is power-law. We therefore state logarithmic mode discovery as an empirical postulate, defensible by direct measurement, not a theorem.

Two sample-size variables matter and are easily conflated. Let $h(n) = \beta k(n)$ be the number of hard-token decisions in a sequence of length n , and T the number of observed hard-failure events in a discovery corpus: $h(n)$ controls per-sequence exposure, T controls empirical discovery. We write C_D for the full reachable catalogue in patch D , $C_{\text{seen},D}(T)$ for the subset discovered after T failures, and $C_{\text{active},D}(n)$ for the subset one sequence of length n can activate. (Small-sample $\ln(1 + \cdot)$ forms and floor/expectation readings of discrete counts do not affect any rate claim below.)

Postulate 1 (Patch-indexed catalogue discovery). Within a fixed application domain D , the number of named recurring failure modes discovered after T sampled hard-failure events is bounded by

$$|C_{\text{seen},D}(T)| \leq \min(A_D + \sigma_D \ln T, |C_D|), \quad \sigma_D > 0, A_D \geq 0.$$

The cap $|C_D|$ is the domain-imposed ceiling. The postulate concerns catalogue *discovery*, not the number of hard decisions inside a single sequence.

Assumption 2 (Per-sequence active-mode exposure). For a single sequence of length n in domain D ,

$$|C_{\text{active},D}(n)| \leq \min(A'_D + \sigma'_D \ln h(n), |C_D|).$$

Primed constants are distinct from those of Postulate 1: corpus discovery and per-sequence activation need not share the same rate.

The two bounds answer different questions: $C_{\text{active},D}(n)$ governs Proposition 2’s sequence-length rate, while full library budgeting uses C_D ; once sampling or activation saturates the cap $|C_D|$, the budget becomes domain-constant.

Empirical calibration. ErrorAtlas, HumanEval-style code taxonomies (Wen et al., 2024), and MWPEs (Sun et al., 2025) place $|C|$ in the 8–20 range across 10^4 to 3×10^5 failures, yielding $\sigma \in [0.87, 1.85]$ under the simple $A = 0$ calibration; we carry $\sigma \approx 1.85$ as a conservative planning value. These are endpoint counts, not discovery curves — the missing test is a subsample-vs-discovered-modes measurement within a patch (Appendix E).

L2 → L4 as weighted set cover. Coverage is formally a weighted set cover (each L4 intervention removes a fraction of several modes’ mass; one maximises $\sum_i p_i \max_{j: I_j \in \mathcal{I}} r_{ij}$ over $|\mathcal{I}| \leq m$). Proposition 2 uses the tractable one-mode-per-intervention special case; since one capability covers several categories, this ranked-category budget is a conservative proxy, and Appendix B (the full harvest) treats overlap and additivity empirically.

Operational patch. A deployment patch D is not a topic label but an operational tuple fixing, over a time window, which failure modes are reachable: $D = (\mathcal{X}, \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{E}, \mathcal{P}, H, \tau)$ — task input distribution, schema family, user/client class, retrieval corpus, evaluator, policy

constraints, workflow horizon, and time window. Several $(\mathcal{R}, S, H)$ are exactly what a context-management system controls. A *patch shift* (enough change to alter C_D) separates legal review on a fixed jurisdiction from mixed jurisdictions, or curated medical RAG from arbitrary-web RAG; the patch-local claims apply within one such fixed tuple, not “the domain” in any looser sense.

Domain patches cap the engineering problem. Localized representations (Park et al., 2024; Li & Sarwate, 2025), cross-domain accuracy spreads (Wang et al., 2024c), and long-tail knowledge (Mallen et al., 2023; Kandpal et al., 2023) make model behaviour strongly domain-dependent without measuring $|C_D|$ directly; we therefore treat the patch’s reachable catalogue as finite or effectively capped as a modelling assumption, not a theorem (evidence in Appendix D).

Corollary 1 (Inverse Discovery Cost). *Reading Postulate 1 in the inverse direction, accommodating q distinct discovered modes requires at least $T \geq \exp((q - A_D)/\sigma_D)$ observed hard failures; equivalently, Δq further modes raise the minimum sample budget by a factor $\exp(\Delta q/\sigma_D)$.*

The lower bound holds unconditionally; under tightness at $\sigma_D \approx 1.85$, five extra modes cost $\approx 15\times$ more observed failures and ten cost $\approx 220\times$. Returns are asymmetric (the head cheap and high-mass, the tail expensive and low-yield), and the claim concerns *newly distinguishable* modes, not ordinary failures. The full derivation, Heaps variant, and a capability-gain sub-corollary are in Appendix A.6.

3.3 Coverage by a targeted intervention library

How much residual error does a library of m interventions close off? Let $p_1 \geq p_2 \geq \dots \geq p_{|C|}$ be the ranked hard-error masses ($\sum_i p_i = 1$); the exact top- m coverage is the empirical step function $F_{\text{emp}}(m) = \sum_{i=1}^m p_i$ (with $F_{\text{emp}}(|C|) = 1$). For closed-form analysis we use the continuum log-head approximation

$$F_{\log}(m; |C|) = \min\left(1, \frac{\ln m}{\ln |C|}\right), \quad m \geq 2, |C| \geq 2, \quad (2)$$

as a *planning approximation* to F_{emp} , not the true distribution. For the ErrorAtlas anchor ($|C| = 17$), $F_{\log}(5; 17) \approx 56.8\%$ and $F_{\log}(10; 17) \approx 81.3\%$ (a Zipf-1 reference is even more head-concentrated, so the log form is conservative). Proposition 2 is thus a closed-form planning approximation, not a distribution-free theorem; the qualitative polylog conclusion survives across coverage families (Appendix A.5), and anchoring F_{emp} to a measured curve in a patch is an explicit falsifiability test.

One caveat carries through: $F_{\log}$ tracks ranked L2 categories, not L4 capability axes, so where one capability sweeps several categories the true residual at a given m is lower than (2) predicts — the bound is valid but loose. After deploying the top- m library, the residual per-hard-token error rate is

$$e_{\text{res}}(m) = (1 - F_{\log}(m; |C|)) e_{\text{hard}} \quad (3)$$

under the log-head approximation, or the corresponding F_{emp} expression if the per-mode masses are known directly.

3.4 From universal impossibility to patch-local reliability

The finite-catalogue claim is patch-local: across all possible tasks, tools, schemas, knowledge sources, and evaluators, new intervention-distinguishable failures keep appearing, so a finite dictionary is not a well-posed universal target. The positive result begins only after a patch D is fixed and its task family, schemas, tools, and evaluators recur, changing the question from “can one library cover all LLM failures?” to “how large must the local library be to cover enough of C_D ?”

Proposition 1 (No Universal Finite Intervention Dictionary). *Fix a residual-error tolerance ε . If a domain D contains an infinite sequence of failures where each new failure is not covered by any*

finite intervention dictionary covering all earlier failures, then the ε -resolution failure catalogue C_D^ε is infinite. Consequently, no finite intervention dictionary can guarantee residual error below ε for every intervention-distinguishable mode in D .

Metric. This is a worst-case, mode-wise guarantee, not a claim about expected residual error: under a distributional metric an infinite uncovered tail may still carry arbitrarily small mass. The proposition rules out only the worst-case-mode-coverage reading of universal reliability — the reading implicit in any request for “a fixed list of interventions that covers LLM use.”

Proof sketch. Each new failure is intervention-distinguishable from its predecessors, so the sequence generates infinitely many distinct modes and any finite dictionary misses some later one; the full proof is in Appendix A.1.

Engineering meaning. A fixed list of interventions cannot cover open-ended LLM use; the rest of the paper is therefore about *patch-local*, not universal, reliability. Proposition 1 inoculates against the misread that a fixed list of ≈ 50 patterns covers LLM use in general.

The positive result composes β -stratification (Definition 1), the active-mode bound (Assumption 2), and the log-coverage form (Eq. 2) into a local budget. It is conditional engineering math — depending on the log-coverage approximation, the active-mode assumption, and the patch hypothesis that C_D is finite or capped (conditions stated in Appendix A.2) — not an unconditional theorem.

Proposition 2 (Patch-Local Sufficient Intervention Budget). *Fix a deployment patch D . Let $\varepsilon \in (0, e_{\text{hard}})$ be the target per-hard-decision residual error rate. Under the log-head approximation $F_{\log}$ of §3.3, a library covering the dominant local modes is sufficient for the target once*

$$m \geq \left\lceil |C_{\text{eff}}|^{1-\varepsilon/e_{\text{hard}}} \right\rceil, \quad (4)$$

where C_{eff} is either the active catalogue $C_{\text{active},D}(n)$ touched by a single sequence, or the full reachable catalogue C_D of the deployment patch. If $C_{\text{active},D}(n)$ grows logarithmically with the number of hard decisions and $k(n) = \Theta(\log n)$, then the pre-cap sufficient budget grows doubly-logarithmically in sequence length. Once the patch catalogue saturates, the sufficient budget becomes domain-constant in $|C_D|$. This is a sufficient, model-implied budget under the stated approximation, not a proof of the true minimal intervention library.

Proof sketch. The top- m library leaves residual $(1 - F(m; |C|)) e_{\text{hard}}$; requiring this $\leq \varepsilon$ and substituting the log-coverage form rearranges to the stated bound (full derivation, pre-cap and cap regimes, in Appendix A.2).

Engineering meaning. Once the patch is fixed, the question is no longer whether arbitrary future failures exist but how quickly the local catalogue is discovered and how much hard-error mass the top interventions remove. The per-hard-token budget is small and slowly growing, domain-constant in the cap regime, with its exact size set by the local discovery and rank-coverage curves rather than a universal prior.

Sequence-level caveat. Proposition 2 bounds residual error *per hard decision*. A one-shot sequence-level target is strictly stricter: as k grows the allowable per-decision residual shrinks and the required library approaches full-catalogue coverage, and in some regimes the non-hard-token mass alone already exceeds the budget. The full three-regime analysis is in Appendix A.3.

The “tens of interventions” rule of thumb is a per-hard-decision planning prior, not a constant derived from the model; the sequence-level analogue requires more interventions and tighter tail coverage.

These are propositions, not theorems: their force is conditional, formalising the consequences of the paper’s modelling assumptions rather than a universal law. The doubly-logarithmic

rate is the optimistic special case; Appendices A.5 and A.4 give the Heaps/saturation variants and the conditional reading.

4 Empirical Evidence

We collect evidence for three load-bearing claims: errors cluster into a small recurring set (A); each cluster is addressable by one targeted intervention (B); reliability decays sublinearly with output length (C).

Claim A: Failure-mode clustering. ErrorAtlas (Ashury-Tahan et al., 2026) is the strongest single result: 83 models $\times$ 35 datasets, $\gtrsim 10^4$ failures, 17 head-concentrated categories. Domain Paretos reproduce the shape (AssertionError+NameError cover 86.35% on HumanEval (Wen et al., 2024); MWPEs’s top-4 math categories dominate (Sun et al., 2025); multi-hop QA (Zhang et al., 2026), agentic tool use (Cemri et al., 2025; Yao et al., 2024), and RAG (Wood & Forbes, 2024) each show <20 modes), and it is *stable* across models (RFMDataset (Guo et al., 2025); EDIT’s $\approx 4.7\%$ key-step fraction (Dai et al., 2025)). Schaeffer et al. (2025) prove the observed power-law eval scaling requires a success-rate distribution heavy-tailed near $p = 0$, related to Postulate 1.

Claim B: Cluster-selective capability interventions. A dedicated harvest yields 28 capability-elimination citations across six independent axes (arithmetic, code execution, format/structure, perception/grounding, knowledge/RAG, verification), each confirmed by three to nine citations; the full table and three structural patterns (A by-construction, B strong-empirical-with-class-shift, C moderate-with-shift) are in Appendix B.

The cleanest cases span the six axes: PAL (Gao et al., 2023a) lifts GSM-Hard 20.1% $\rightarrow$ 61.5% (residuals in comprehension); constrained decoding zeroes invalid-token probability *by construction* (Suresh et al., 2025; Wang et al., 2025b; Dong et al., 2025); and execution feedback, process supervision, RAG, preference optimisation, and clarification each largely close their target cluster (e.g. Reflexion+AgentCoder 80% $\rightarrow$ 96.3% (Shinn et al., 2023; Huang et al., 2023); Math-Shepherd 28.6% $\rightarrow$ 43.5% (Wang et al., 2024b); full table with Acurai, POROver, and SAGE-Agent in Appendix B). The *class-shift* signature — residuals landing in structurally different classes — is the empirical content of the cluster-selectivity underwriting Proposition 2’s composition, and DebugBench (Tian et al., 2024) is a sharp negative control: execution feedback fixes syntax/reference errors but is “unhelpful for logic errors,” where the framework predicts provisioning fails.

Claim C: Sublinear length scaling. Sublinear (not exponential) decay shows up across very different designs. Loong (Wang et al., 2024a) has GPT-4o decline 81.6% $\rightarrow$ 32.9% across 10K $\rightarrow$ >200 K context, log-linear in $\log L$; GSM- ∞ (Zhou et al., 2025) finds exponentially more compute buys only *linear* AUC (DeepSeek-R1 at 10%+ at 130 operations where $(1 - \epsilon)^{130}$ predicts near-zero); and Press et al. (2023) report a compositionality gap roughly *constant* across the GPT-3 family. Structural correlates agree (Anchor LLMs’ $\approx 99\%$ K/V reduction (Pang et al., 2024); RULER’s threshold behaviour (Hsieh et al., 2024); METR’s logistic-in-log-length (Kwa et al., 2025)), while prominent steep-decay counter-evidence (Dziri et al., 2023; Kuratov et al., 2024; Wan et al., 2026; Karpinska et al., 2024) decays over variables distinct from raw token length (Appendix C); self-consistency (Wang et al., 2023) is suggestive, not diagnostic.

5 Practical Implications

Reliability engineering is local patch coverage. Within a fixed patch, Proposition 2 makes reliability a small-catalogue engineering problem rather than an asymptotic scaling one. A team budgets initially for a library of tens of interventions, then refines from the local discovery curve $C_{\text{seen},D}(T)$ and rank-frequency distribution; consistent with the head-mass figures in ErrorAtlas, HumanEval, and MWPEs, ≈ 50 named interventions cover the bulk of the per-hard-token failure mass in many measured domains. This is a planning prior,

not a universal constant: the same base model in cardiology RAG, legal drafting, and code review yields three different libraries because C_D , A_D , σ_D all change with the patch. Note the metric: per-hard-token residual error (continuous-correction systems) has a polylog budget, but sequence-level failure probability (one-shot systems) is strictly stricter, so SLAs must match the actual cost structure rather than read the per-token result as the production SLA.

Libraries are modular and capability-coarse. A library can be assembled module-by-module in any order as long as modules target distinct classes; layer-separated interventions (constrained decoding, retrieval, process supervision, tool calls) compose approximately additively (the Le (2026) caveat applies only within a shared prompt channel), so construction is highly parallelisable. It is also coarser than the error-class count: one Python interpreter collapses five math classes, one constrained decoder collapses format plus the structural “missing required element” ($> 20\%$ of ErrorAtlas), so fewer than 50 capability-axis interventions cover ≈ 50 named clusters — the six axes of Appendix B are the better accounting unit.

6 Discussion

By-construction elimination. Seven of the 28 citations (Appendix B: constrained decoders, proof kernels, syntax checks) achieve zero residual error *by construction* — a constrained decoder sets $P(\text{invalid token}) = 0$, emptying the grammar-violating class. The polylog bound is then merely loose (covered clusters contribute exactly zero), strengthening the framework, and Pattern A applies to the structural classes at the *head* of the distribution: the strongest mechanism lands where the catalogue is densest.

Counter-evidence relocates, not dissolves. Five prominent steep-decay papers cited as exponential-compounding evidence each decay over a variable distinct from raw token length — compositional graph size (Dziri et al., 2023), supporting-fact count (Kuratov et al., 2024), log-time horizon (Kwa et al., 2025), capacity threshold (Wan et al., 2026), evidence scope (Karpinska et al., 2024) — all quantities the framework already concentrates the action in ($k_{\text{hard}}, |C|$). These regimes remain hard; the value is directing intervention along the actual decay axis, not context-window expansion (re-audits in Appendix C).

Limitations. Postulate 1 is empirical, not derived; genuinely Heaps-power-law discovery would invalidate the doubly-logarithmic special case while preserving the qualitative polylog conclusion. The coverage form of §3.3 is an empirical best-fit, not theoretical confirmation; inter-cluster additivity is only approximate (Le 2026); the $\sigma \approx 1.85$ estimate is a single-point calibration; β is latent; and the L2–L3 granularity gap is unmeasured. The empirical anchor rests on three 2025–2026 taxonomies (general/code/math), untested on agentic workflows, long scientific reasoning, and multi-turn tool use over millions of tokens — where $|C|$ may grow faster than logarithmically and patch-shift changes $A_D, \sigma_D, \beta_D, |C_D|$ together, so a library calibrated on one patch under-covers the next. Most fundamentally, the framework *relocates* long-context difficulty rather than resolving it: where k_{hard} grows with task length, reliability remains hard, and the contribution is to name the on-axis intervention, not make those regimes easy.

7 Conclusion

LLM reliability is often framed as asymptotic scaling. Universal reliability is indeed not a finite-library problem (Proposition 1). But once an operationally bounded patch is fixed, reliability becomes local — within the sparse set of hard decisions, errors are *repetitive*, clustering into a finite catalogue whose size grows logarithmically with observed failures (§3.2), or as a small power under the Heaps alternative (Appendix A.5). The conditional consequence (Proposition 2) is a per-hard-decision budget that scales polylogarithmically in sequence length and becomes domain-constant once $|C_D|$ saturates; sequence-level targets are strictly tighter. Read inversely (Corollary 1), the same postulate explains why generic

frontier post-training faces diminishing reliability returns once a patch’s head modes are covered.

This shifts the question from “can we bound n -token error?” to “have we catalogued enough failure modes *inside the patch*?” — finite and addressable, with direct measurement of the mode-rate σ on new domains the natural empirical follow-up. The paper names the engineering object (a patch-local failure catalogue and the budget covering its head) but not how the deployment-time scaffold of instructions, tools, retrieval, memory, and orchestration governs that library over time; in that frame, reliability engineering is *governance of the scaffold rather than scaling of the weights*.

References

- Mikhail L. Arbuzov, Sisong Bei, Ziwei Dong, Dmitri Kalaev, and Alexey Shvets. Beyond exponential decay: Rethinking error accumulation in large language models. *arXiv preprint arXiv:2505.24187v2*, 2025. Posted 06 May 2026; CC BY 4.0 license.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024.
- Shir Ashury-Tahan, Yifan Mai, Elron Bandel, Michal Shmueli-Scheuer, and Leshem Choshen. ErrorMap and ErrorAtlas: Charting the failure landscape of large language models. *arXiv preprint*, 2026.
- Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? *arXiv preprint*, 2025. Introduces the MAST taxonomy (Multi-Agent System Failure Taxonomy).
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research (TMLR)*, 2023.
- Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of ACL*, 2024.
- Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. Capture the key in reasoning to enhance CoT distillation generalization. In *Proceedings of ACL*, 2025. Earlier arXiv version titled “Beyond Imitation: Learning Key Reasoning Steps from Dual Chain-of-Thoughts in Reasoning Distillation”.
- Alex Dantart. Reliability by design: Quantifying and eliminating fabrication risk in LLMs. from generative to consultative AI: A comparative analysis in the legal domain and lessons for high-stakes knowledge bases. *arXiv preprint*, 2026.
- Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ruihang Lai, Ziyi Xu, Yilong Zhao, and Tianqi Chen. XGrammar: Flexible and efficient structured generation engine for large language models. In *Proceedings of MLSys*, 2025.
- Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenyang Zhang, Stefanie Jegelka, Jianfeng Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? In *International Conference on Learning Representations (ICLR)*, 2025.

- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *Proceedings of ICML*, 2023a.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of EMNLP*, 2023b. Introduces the ALCE benchmark.
- Alex J. Goodell, Simon N. Chu, Dara Rouholiman, and Larry F. Chu. Large language model agents can use tools to perform clinical calculations. *npj Digital Medicine*, 8(1):163, 2025. doi: 10.1038/s41746-025-01475-8.
- Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *International Conference on Learning Representations (ICLR)*, 2025. ICLR 2025 Oral. Introduces the UGround model.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations (ICLR)*, 2024a.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *International Conference on Learning Representations (ICLR)*, 2024b.
- Dadi Guo, Jiayu Liu, Zhiyuan Fan, Zhitao He, Haoran Li, Yuxin Li, Yumeng Wang, and Yi R. Fung. Mathematical proof as a litmus test: Revealing failure modes of advanced large reasoning models. *arXiv preprint*, 2025. Introduces the RFMDataset (Reveal Failure Modes).
- Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What’s the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024.
- Dong Huang, Qingwen Bu, Jie M. Zhang, Michael Luck, and Heming Cui. AgentCoder: Multi-agent-based code generation with iterative testing and optimisation. *arXiv preprint*, 2023.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of ICML*, 2023.
- Batuhan K. Karaman, Ishmam Zabir, Alon Benhaim, Vishrav Chaudhary, Mert R. Sabuncu, and Xia Song. POROver: Improving safety and reducing overrefusal in large language models with overgeneration and preference optimization. *arXiv preprint*, 2024.
- Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. One thousand and one pairs: A “novel” challenge for long-context language models. In *Proceedings of EMNLP*, 2024.
- Yuri Kuratov, Aydar Bulatov, Pavel Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *NeurIPS Datasets and Benchmarks Track*, 2024.
- Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. METR technical report.
- Yifan Le. Schema key wording as an instruction channel in structured generation under constrained decoding. *arXiv preprint*, 2026.

- Yann LeCun. Auto-regressive LLMs are doomed. Tweet, March 26, 2023. <https://x.com/ylecun/status/1640122342570336267>, 2023.
- Linzhang Li, Yixin Dong, Guanjie Wang, Ziyi Xu, Alexander Jiang, and Tianqi Chen. XGrammar-2: Efficient dynamic structured generation engine for agentic LLMs. *arXiv preprint*, 2026.
- Xin Li and Anand D. Sarwate. Unraveling the localized latents: Learning stratified manifold structures in LLM embedding space with sparse mixture-of-experts. *arXiv preprint*, 2025.
- Yujia Li, David H. Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d’Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *Science*, 378(6624): 1092–1097, 2022. doi: 10.1126/science.abq1158.
- 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. In *International Conference on Learning Representations (ICLR)*, 2024.
- Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhua Chen, Nigel Collier, and Yasemin Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of ACL*, 2023.
- Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. OmniParser for pure vision based GUI agent. *arXiv preprint*, 2024.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of ACL*, 2023. Introduces the PopQA benchmark.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. Section on Heaps’ law: <https://nlp.stanford.edu/IR-book/html/htmledition/heaps-law-estimating-the-number-of-terms-1.html>.
- Ayana Niwa and Hayate Iso. AmbigNLG: Addressing task ambiguity in instruction for NLG. In *Proceedings of EMNLP*, 2024.
- OpenAI. Introducing structured outputs in the API. OpenAI Engineering Blog. <https://openai.com/index/introducing-structured-outputs-in-the-api/>, 2024. Published August 6, 2024.
- OpenAI. GPT-5 system card. <https://cdn.openai.com/gpt-5-system-card.pdf>, 2025. Published August 7, 2025.
- Jianhui Pang, Fanghua Ye, Derek F. Wong, Xun He, Wenxiang Chen, and Longyue Wang. Anchor-based large language models. In *Findings of ACL*, 2024.
- Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of ICML*, 2024.
- Khush Patel, Siva Surendira, Jithin George, and Shreyas Kapale. The six sigma agent: Achieving enterprise-grade reliability in LLM systems through consensus-driven decomposed execution. *arXiv preprint*, 2026.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*, 2023.

- Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint*, 2025.
- Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. How do large language monkeys get their power (laws)? In *International Conference on Machine Learning (ICML)*, 2025.
- Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. AgentSquare: Automatic LLM agent search in modular design space. *arXiv preprint*, 2024.
- Yuling Shi, Songsong Wang, Chengcheng Wan, Min Wang, and Xiaodong Gu. From code to correctness: Closing the last mile of code generation with hierarchical debugging. *arXiv preprint*, 2024. Introduces the MGDebugger method.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Yuhong Sun, Zhangyue Yin, Xuanjing Huang, Xipeng Qiu, and Hui Zhao. Error classification of large language models on math word problems: A dynamically adaptive framework. In *arXiv preprint*, 2025. Introduces the MWPEs-300K dataset (304,865 error samples from 15 LLMs across four MWP datasets).
- Tarun Suresh, Debangshu Banerjee, Shubham Ugare, Sasa Misailovic, and Gagandeep Singh. DINGO: Constrained inference for diffusion LLMs. *arXiv preprint*, 2025.
- Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for LLM agents. *arXiv preprint*, 2025. Introduces the SAGE-Agent method.
- Runchu Tian, Yining Ye, Yujia Qin, Xin Cong, Yankai Lin, Yinxu Pan, Yesai Wu, Haotian Hui, Weichuan Liu, Zhiyuan Liu, and Maosong Sun. DebugBench: Evaluating debugging capability of large language models. In *Findings of ACL*, 2024.
- Akihiko Wada, Yuya Tanaka, Mitsuo Nishizawa, Akira Yamamoto, Toshiaki Akashi, Akifumi Hagiwara, Yayoi Hayakawa, Junko Kikuta, Keigo Shimoji, Katsuhiko Sano, Koji Kamagata, Atsushi Nakanishi, and Shigeki Aoki. Retrieval-augmented generation elevates local LLM quality in radiology contrast media consultation. *npj Digital Medicine*, 8(1):395, 2025. doi: 10.1038/s41746-025-01802-z.
- Kaiyang Wan, Lang Gao, Honglin Mu, Preslav Nakov, Yuxia Wang, and Xiuying Chen. A Fano-style accuracy upper bound for LLM single-pass reasoning in multi-hop QA. In *International Conference on Learning Representations (ICLR)*, 2026.
- Benlu Wang, Iris Xia, Yifan Zhang, Junda Wang, Feiyan Ouyang, Shuo Han, Arman Cohan, Hong Yu, and Zonghai Yao. From scores to steps: Diagnosing and improving LLM performance in evidence-based medical calculations. In *Proceedings of EMNLP*, 2025a. Introduces the MedRaC benchmark.
- Darren Yow-Bang Wang, Zhengyuan Shen, Soumya Smruti Mishra, Zhichao Xu, Yifei Teng, and Haibo Ding. SLOT: Structuring the output of large language models. *arXiv preprint*, 2025b. SLOT = Structured LLM Output Transformer.
- Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of EMNLP*, 2024a.

- Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of ACL*, 2024b.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *NeurIPS Datasets and Benchmarks Track (Spotlight)*, 2024c.
- Hao Wen, Yueheng Zhu, Chao Liu, Xiaoxue Ren, Weiwei Du, and Meng Yan. Fixing function-level code generation errors for foundation large language models. *arXiv preprint*, 2024. Introduces the LlmFix method. Reports AssertionError 63.64% + NameError 22.71% = 86.35% of HumanEval failures across 14 LLMs.
- Michael C. Wood and Adam A. Forbes. 100% elimination of hallucinations on RAGTruth for GPT-4 and GPT-3.5 Turbo. *arXiv preprint*, 2024. Introduces the Acurai method.
- Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. Scaling computer-use grounding via user interface decomposition and synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. NeurIPS 2025 Spotlight. Introduces the Jedi grounding dataset and OSWorld-G benchmark.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint*, 2024.
- Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R. Dalal, Jennifer L. Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, Jack Boyd, Kathleen Boyd, Karen Hirsch, Curtis Langlotz, Rita Lee, Joanna Melia, Joanna Nelson, Karim Sallam, Stacey Tullis, Melissa Ann Vogelsong, John Patrick Cunningham, and William Hiesinger. Almanac — retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2): AIoa2300068, 2024. doi: 10.1056/AIoa2300068.
- Kexun Zhang, Hongqiao Chen, Lei Li, and William Wang. Don’t fine-tune, decode: Syntax error-free tool use via constrained decoding. *arXiv preprint*, 2023. Introduces the ToolDec method.
- Meiru Zhang, Zaiqiao Meng, and Nigel Collier. Failure modes in multi-hop QA: The weakest link effect and the recognition bottleneck. *arXiv preprint*, 2026.
- Yang Zhou, Hongyi Liu, Zhuoyan Chen, Yuandong Tian, and Beidi Chen. GSM-Infinite: How do your LLMs behave over infinitely increasing context length and reasoning complexity? *arXiv preprint*, 2025.

A Formal Proofs, Derivations, and Sensitivity Analysis

A.1 Proof of Proposition 1: No Universal Finite Intervention Dictionary

Fix a residual-error tolerance ε . Say that an intervention *covers* a failure mode if it reduces the residual error of that mode below ε . Coverage is mode-level: an intervention that covers a mode covers every failure event in that mode. The two conditions “ D is intervention-unbounded” and “ $|C_D^\varepsilon| = \infty$ ” are equivalent under mode-level coverage; an unbounded witnessing sequence is constructed by taking one representative per mode, and an infinite catalogue forces the existence of such a sequence.

Let D be a domain. Call D *intervention-unbounded* if it contains an infinite sequence of failures $f_1, f_2, f_3, \dots$ such that each new f_j is not covered by any finite intervention dictionary that covers all earlier failures $\{f_1, \dots, f_{j-1}\}$.

Step 1: assume the opposite. Suppose, for contradiction, that C_D^ε is finite. Then there are only finitely many intervention-distinguishable failure modes in D . Write them as $C_D^\varepsilon = \{c_1, \dots, c_M\}$ for some finite M .

Step 2: what finiteness means. If there are only M intervention-distinguishable modes, then after all M modes have appeared in the sequence, every later failure must belong to one of the already-seen modes.

Step 3: same mode means same intervention class. Modes are defined at intervention resolution ε . If a later failure belongs to the same mode as an earlier failure, then the intervention dictionary that covers the earlier representative of that mode also covers the later failure below residual tolerance ε .

Step 4: contradiction. Intervention-unboundedness says exactly the opposite: each new f_j is not covered by any finite dictionary that covers $\{f_1, \dots, f_{j-1}\}$. Therefore f_j cannot belong to any earlier intervention mode, so each f_j introduces a new intervention-distinguishable mode. The sequence $f_1, f_2, f_3, \dots$ induces infinitely many such modes, contradicting the assumption that C_D^ε is finite. Hence $|C_D^\varepsilon| = \infty$.

Step 5: no finite dictionary covers every mode of the domain. Suppose, again for contradiction, that some finite intervention dictionary $\mathcal{I}$ covers every mode of D . Then $\mathcal{I}$ covers every finite prefix $\{f_1, \dots, f_{j-1}\}$ for every j . By intervention-unboundedness, any dictionary covering that prefix fails to cover f_j . Therefore $\mathcal{I}$ does not cover f_j , contradicting the assumption that $\mathcal{I}$ covers every mode of D . $\square$

A.2 Derivation of Proposition 2: Patch-Local Sufficient Intervention Budget

Conditions used. Proposition 2 gives a sufficient, model-implied budget under the following four assumptions:

1. **Coverage model.** The cumulative hard-error mass covered by the top m modes is approximated by the log-head form

$$F_{\log}(m; |C|) = \min\left(1, \frac{\ln m}{\ln |C|}\right)$$

on the declared domain $m \geq 2, |C| \geq 2$. We write F for $F_{\log}$ throughout this appendix unless otherwise noted.

2. **Non-trivial, attainable target.** The residual target satisfies $\varepsilon \in (0, e_{\text{hard}})$. If $\varepsilon \geq e_{\text{hard}}$ no intervention is needed; if $\varepsilon \leq 0$ the target is unattainable unless all residual hard-token error is eliminated.
3. **Patch-local catalogue.** The result applies only after a deployment patch D has been fixed and its reachable catalogue is modelled as finite or effectively capped.
4. **Sequence-length claim.** The doubly-logarithmic rate further requires Assumption 2 and $k(n) = \Theta(\log n)$. Without these, Proposition 2 still gives a catalogue-size budget but not the same sequence-length scaling.

Step 1: define the target. Let e_{hard} be the baseline hard-token error rate and $\varepsilon \in (0, e_{\text{hard}})$ the target after intervention.

Step 2: define the effective catalogue. For per-sequence scaling, set $C_{\text{eff}} = C_{\text{active}, D}(n)$: how many failure modes can a single sequence of length n activate? For full deployment-library budgeting, set $C_{\text{eff}} = C_D$: how large must the library be to cover the recurring failure modes reachable inside D ?

Step 3: cumulative coverage. Let the ranked local failure modes have hard-error masses $p_1 \geq p_2 \geq \dots \geq p_{|C_{\text{eff}}|}$ with $\sum_i p_i = 1$. A library covering the top m modes removes cumulative hard-error mass $F(m; |C_{\text{eff}}|) = \sum_{i=1}^m p_i$, approximated by the log-coverage form of §3.3.

Step 4: residual after intervention. The uncovered mass fraction is $1 - F(m; |C_{\text{eff}}|)$, so the residual per-hard-token error rate is $e_{\text{res}}(m) = (1 - F(m; |C_{\text{eff}}|)) e_{\text{hard}}$.

Step 5: impose the target. Requiring $e_{\text{res}}(m) \leq \varepsilon$ and dividing by $e_{\text{hard}} > 0$ gives

$$1 - F(m; |C_{\text{eff}}|) \leq \frac{\varepsilon}{e_{\text{hard}}}, \quad \text{i.e.,} \quad F(m; |C_{\text{eff}}|) \geq 1 - \frac{\varepsilon}{e_{\text{hard}}}.$$

Step 6: substitute the log-coverage form. Since $\varepsilon < e_{\text{hard}}$, the required coverage $1 - \varepsilon/e_{\text{hard}} \in (0, 1)$, so the cap in F is inactive before saturation. Using $F = \ln m / \ln |C_{\text{eff}}|$:

$$\frac{\ln m}{\ln |C_{\text{eff}}|} \geq 1 - \frac{\varepsilon}{e_{\text{hard}}}.$$

Step 7: solve for m . Multiplying by $\ln |C_{\text{eff}}| > 0$ and exponentiating,

$$m \geq \exp \left[\left(1 - \frac{\varepsilon}{e_{\text{hard}}} \right) \ln |C_{\text{eff}}| \right] = |C_{\text{eff}}|^{1 - \varepsilon/e_{\text{hard}}}.$$

Taking the ceiling (since m is integer-valued) yields $m \geq \lceil |C_{\text{eff}}|^{1 - \varepsilon/e_{\text{hard}}} \rceil$, the bound stated in Eq. (4).

Step 8: sequence-length rate. For per-sequence scaling, set $C_{\text{eff}} = C_{\text{active}, D}(n)$. Assumption 2 bounds $|C_{\text{active}, D}(n)| \leq \min(A'_D + \sigma'_D \ln h(n), |C_D|)$. In the pre-cap regime, i.e., while $A'_D + \sigma'_D \ln h(n) < |C_D|$, the ceiling has not been reached and $|C_{\text{active}, D}(n)| = O(\ln h(n))$. With $h(n) = \beta k(n)$ and $k(n) = \Theta(\log n)$, we get $h(n) = \Theta(\log n)$ and $\ln h(n) = \Theta(\log \log n)$, hence

$$|C_{\text{active}, D}(n)| = O(\log \log n),$$

and substituting into the budget,

$$m = O((\log \log n)^{1 - \varepsilon/e_{\text{hard}}}).$$

This is the doubly-logarithmic pre-cap special case.

Step 9: cap regime. Once active-mode discovery saturates the patch catalogue, $|C_{\text{active}, D}(n)| = |C_D|$, so $m \geq \lceil |C_D|^{1 - \varepsilon/e_{\text{hard}}} \rceil$, which no longer depends on n . The intervention budget is then domain-constant. $\square$

A.3 Sequence-Level Reliability Derivation

Proposition 2 gives a per-hard-token residual target. Production systems often care about a stricter target: the probability that the entire sequence is correct.

Starting from the composed reliability of Eq. (1), and writing the post-intervention hard-token rate as e_{res} ,

$$P(\text{correct}) = (1 - e_{\text{res}})^{\beta k} (1 - e_{\text{easy}})^{(1 - \beta)k} (1 - e_{\text{non}})^{n - k}.$$

Define the non-hard-token survival factor

$$S_{\text{base}} = (1 - e_{\text{easy}})^{(1 - \beta)k} (1 - e_{\text{non}})^{n - k}, \quad (5)$$

so that $P(\text{correct}) = (1 - e_{\text{res}})^{\beta k} S_{\text{base}}$. The sequence-level target $P(\text{correct}) \geq 1 - \varepsilon_{\text{seq}}$ becomes

$$(1 - e_{\text{res}})^{\beta k} S_{\text{base}} \geq 1 - \varepsilon_{\text{seq}}, \quad \text{i.e.,} \quad (1 - e_{\text{res}})^{\beta k} \geq \frac{1 - \varepsilon_{\text{seq}}}{S_{\text{base}}}.$$

Three regimes follow.

Regime (i): non-hard-token errors already violate the target. If $S_{\text{base}} < 1 - \varepsilon_{\text{seq}}$, then even setting $e_{\text{res}} = 0$ cannot meet the target, because the maximum possible survival after eliminating all hard-token failures is only S_{base} . Hard-token interventions alone cannot meet the sequence-level SLA. This is the route by which the exponential-in- n concern re-enters and should be diagnosed before any catalogue-budgeting exercise.

Regime (ii): hard-token residual error determines feasibility. If $S_{\text{base}} \geq 1 - \varepsilon_{\text{seq}}$, the target may be feasible. Taking the $(1/(\beta k))$ -th power of the rearranged inequality and isolating e_{res} ,

$$e_{\text{res}} \leq 1 - \left(\frac{1 - \varepsilon_{\text{seq}}}{S_{\text{base}}} \right)^{1/(\beta k)} =: \tau_{\text{seq}}.$$

Applying Proposition 2 with ε replaced by τ_{seq} ,

$$m \geq \left\lceil |C_{\text{eff}}|^{1 - \tau_{\text{seq}}/e_{\text{hard}}} \right\rceil.$$

As the sequence-level target becomes stricter (ε_{seq} shrinks), $\tau_{\text{seq}} \rightarrow 0$ and the exponent $1 - \tau_{\text{seq}}/e_{\text{hard}} \rightarrow 1$, so $m \rightarrow |C_{\text{eff}}|$. Strict one-shot sequence-level reliability pushes the system toward full-catalogue coverage.

Regime (iii): baseline hard-token error is already acceptable. If $\tau_{\text{seq}} \geq e_{\text{hard}}$, the baseline hard-token rate already satisfies the sequence target (no intervention is required because $e_{\text{res}}(0) = e_{\text{hard}}$ from Eq. (3)).

Conclusion. Per-hard-token reliability is easier than sequence-level reliability. A library that gives a large reduction in residual hard-token error may still be insufficient for one-shot sequence-level guarantees when many hard decisions occur in a single output. This is why the main paper treats the “tens of interventions” rule of thumb as a per-hard-decision planning prior, not a one-shot sequence-level SLA.

A.4 Why these are propositions rather than unconditional theorems

Proposition 1 is a definitional impossibility result: once intervention-unboundedness is assumed, an infinite intervention-resolution catalogue follows. Its role is not to prove that every unbounded domain necessarily has infinite failure modes, but to show that open-ended domains cannot be assumed to admit finite dictionaries.

Proposition 2 is conditional engineering math. It does not prove that LLM failures universally obey logarithmic mode discovery. It proves that *if* a bounded patch has a finite or effectively capped reachable catalogue, *and if* cumulative intervention coverage follows the stated head-heavy form, *then* a sufficient per-hard-decision intervention budget grows slowly and becomes constant after catalogue saturation. It does not prove that the true minimal intervention library has the same scaling.

The empirical burden therefore lies not in the algebra but in measuring, for each deployment patch, the local discovery curve $C_{\text{seen},D}(T)$, the per-sequence activation $C_{\text{active},D}(n)$, the cumulative coverage $F(m; |C_D|)$, the hard-token fraction β_D , and the baseline hard-token rate e_{hard} . This is why the paper frames the results as a reliability-engineering scaffold rather than as universal theorems about LLM behaviour.

A.5 Heaps Power-Law Variant and Cluster-Count Sensitivity

A reader who prefers to derive cluster-count growth from standard Heaps’ law rather than from Postulate 1 obtains a qualitatively similar result. Let $|C|(k_{\text{hard}}) \approx K \cdot k_{\text{hard}}^b$ with $b \in (0, 1)$. Canonical fits give $b \approx 0.5$ for natural-language vocabularies; failure-mode taxonomies plateau much more sharply (ErrorAtlas stabilises at $|C| = 17$ across $10^4 +$ failures), implying a small effective $b \approx 0.24\text{--}0.32$ under a crude no-intercept endpoint read

(not a fitted discovery exponent) in our setting. Composing with $k = \Theta(\log n)$, we get $|C| = O((\log n)^b)$, and Proposition 2 becomes

$$m = O\left((\log n)^{b \cdot (1 - \varepsilon / e_{\text{hard}})}\right).$$

This is still polylogarithmic in n for any $b \in (0, 1)$ and any $\varepsilon < e_{\text{hard}}$. The paper’s qualitative claim survives either choice of cluster-count law.

Symbolic-form sensitivity across candidate laws. The polylog conclusion depends on which cluster-count law one accepts; available evidence is consistent with multiple candidates because no subsample-discovery curve has been published for any LLM failure-mode taxonomy at this writing. We therefore report symbolic rates rather than fitted constants. With $h(n) = \beta k(n)$:

- **Logarithmic:** $|C_{\text{active},D}(n)| = O(\log h(n))$. If $k(n) = \Theta(\log n)$, then $m = O((\log \log n)^{1 - \varepsilon / e_{\text{hard}}})$.
- **Heaps:** $|C_{\text{active},D}(n)| = O(h(n)^b)$ with $b \in (0, 1)$. If $k(n) = \Theta(\log n)$, then $m = O((\log n)^{b(1 - \varepsilon / e_{\text{hard}})})$.
- **Saturating:** $|C_{\text{active},D}(n)| \leq |C_D|$. Then $m = O(|C_D|^{1 - \varepsilon / e_{\text{hard}}})$, constant in n once the patch ceiling is reached.

The qualitative conclusion is robust **under the $k(n) = \Theta(\log n)$ regime**: m grows more slowly than any positive power of n under every candidate, and the directional claim (“a small library covers the head of the failure distribution in the per-hard-token regime”) survives. Only the exponent shifts: doubly-logarithmic under logarithmic discovery, $(\log n)^b$ with small b under Heaps, constant in the cap regime. The doubly-logarithmic rate is the optimistic special case. If $k(n)$ grows as a positive power of n , the Heaps variant inherits that power and the polylog-in- n language fails along that axis; the framework’s intervention prescription still applies, but its asymptotic-rate framing does not.

Numerical constants require a measured discovery curve $C_{\text{seen},D}(T)$ or $C_{\text{active},D}(n)$. Existing taxonomies provide endpoint category counts at a single corpus scale (ErrorAtlas at $|C| = 17$ for $\approx 10^4$ failures), not discovery curves. The subsample-discovery measurement remains the explicit empirical test that would either tighten the postulate or fall back to the Heaps variant. Until that measurement exists, the framework’s headline rate should be read as *polylogarithmic in the pre-cap regime, domain-constant in the cap regime*, with the specific exponent flagged as a falsifiability test rather than a fitted prediction.

A.6 Inverse Discovery Cost

The body Corollary 1 inverts the upper-bound discovery postulate into a sample-budget lower bound on novel-mode discovery. This appendix gives the algebra, the numerical anchors, the tightness assumption that converts the lower bound into an approximate inverse cost, sensitivity to the Heaps cluster-count alternative of §A.5, the saturation regime, and a separate mode-mediated gain sub-corollary that connects discovery cost to broad capability proxies.

Setup. Let $q(T) = |C_{\text{seen},D}(T)|$ be the number of distinct failure modes discovered in patch D after T observed hard-failure events. Postulate 1 is an upper bound,

$$q(T) \leq A_D + \sigma_D \ln T, \quad \sigma_D > 0,$$

on the discovered catalogue under the logarithmic upper-bound assumption.

Inversion (lower bound on T). The upper-bound postulate is monotone in T . Taking the inverse direction yields a *lower bound* on T required for the cap to accommodate q discovered modes: if $q > A_D$ distinct modes have been discovered after T events, then

$$T \geq \exp\left(\frac{q - A_D}{\sigma_D}\right).$$

This is the rigorous direction of the corollary. The cap cannot accommodate q until the sample budget has grown exponentially in q .

Tightness assumption. The lower bound above is unconditional under Postulate 1. The stronger reading is that observed hard failures at scale $T(q) \approx \exp((q - A_D)/\sigma_D)$ actually deliver q discovered modes, and that each additional Δq modes raises the sample budget by approximately $\exp(\Delta q/\sigma_D)$. This stronger reading requires an additional assumption that the empirical discovery curve is approximately *tight* against the bound at the relevant corpus scales. Without that assumption, the appendix gives a lower bound on T only. With it,

$$\frac{T(q + \Delta q)}{T(q)} \approx \exp\left(\frac{\Delta q}{\sigma_D}\right).$$

Numerical anchors under tightness. At the conservative calibration $\sigma_D \approx 1.85$ (§3.2), $\exp(5/1.85) \approx 14.9$ and $\exp(10/1.85) \approx 222$: five extra modes need roughly $15\times$ more observed hard failures, ten extra modes need roughly $220\times$ more. These are tightness-conditional anchors, not unconditional consequences of Postulate 1. The subsample-discovery measurement of §3.2 is precisely the test of whether tightness holds.

What this is and is not. The corollary describes *new distinct-mode discovery*. Ordinary failures inside already-discovered modes may remain common and cheap to observe; the exponential cost lives on the category-novelty axis. Conflating ordinary failure rate with novel-mode arrival rate would over-claim the result.

Heaps alternative. Under the Heaps cluster-count law of §A.5, $q(T) = KT^b$ with $b \in (0, 1)$, the corresponding inverse cost is polynomial rather than exponential:

$$T(q) = (q/K)^{1/b}.$$

The exponential inverse-cost reading is specific to the logarithmic upper bound; the broader qualitative claim that tail discovery has diminishing returns survives under any concave discovery curve. Sensitivity is therefore: exponential under logarithmic-and-tight, polynomial under Heaps, undefined past the patch ceiling.

Saturation regime. Once $q(T) \leq |C_D|$ has been saturated, discovery stops; the inversion applies only in the pre-cap regime. Inside saturated patches the corollary’s multiplicative-cost reading is vacuous because no novel modes remain to discover, which is itself a property of the patch and not a failure of the corollary.

Mode-mediated capability gain (sub-corollary). Suppose broad capability or reliability gain G inside the patch is approximately linear in the number of useful discovered modes, $G(q) = G_0 + \gamma q$ for some $\gamma > 0$. Composing with the logarithmic upper bound at tightness,

$$G(T) \approx G_0 + \gamma A_D + \gamma \sigma_D \ln T,$$

so G grows logarithmically in observed hard-failure exposure under tightness. Inverting,

$$T(G) \approx \exp\left(\frac{G - G_0 - \gamma A_D}{\gamma \sigma_D}\right).$$

A linear gain in mode-mediated broad reliability therefore corresponds to exponential growth in observed hard-failure exposure under the postulate at tightness. We deliberately phrase G as a *mode-mediated capability/reliability* proxy, not as “intelligence”: the corollary does not say frontier scaling is useless or that intelligence requires exponential data in any general sense. It says, more narrowly, that for fixed deployment reliability where improvement is mediated by discovering new useful modes, generic open-domain training pays a heavy data tax relative to direct patch-local measurement and intervention.

Engineering reading. The corollary explains, without invoking new mechanisms, two empirical signals: (a) why generic post-training shows diminishing reliability returns once a domain’s head modes are covered, and (b) why patch-local measurement combined with targeted tools, retrieval, validators, constrained decoding, and process supervision often outperforms more frontier-scale data on the deployment SLA. Frontier scaling and patch-local engineering solve different problems: scaling improves the substrate; patch-local engineering removes recurring deployment failure mass. $\square$

B Full Failure-Mode Taxonomy and the Capability-Elimination Harvest

The intervention literature provides at least one targeted countermeasure for each named cluster of §4. Two organising granularities exist: at the capability level (six axes) the cluster-selectivity property is most clearly visible; at the error-class level (twelve named clusters) the evidence aligns with the taxonomies of §4 but is finer-grained than the underlying capability mechanisms.

Six capability axes and three structural patterns. A dedicated harvest yields **28 quantitatively-anchored citations** across **six independent capability axes**: Arithmetic (Python/symbolic execution), Code Execution (REPL/sandbox feedback), Format/Structure (constrained decoding, FSMs, grammar engines), Perception/Grounding (visual grounding for GUI, charts, tables), Knowledge/RAG (dense retrieval and citation grounding), Verification (proof checkers, learned verifiers, classifier rerouting, process supervision). Each axis is independently confirmed by between three and nine citations. We stratify the 28 by kind of evidence into three patterns:

Pattern A: hard guarantees (by-construction). Seven citations achieve residual error rate equal to zero *by construction*, restricted strictly to structural/verifiable classes: constrained decoders set $P(\text{invalid token}) = 0$ at every step (Suresh et al., 2025; Zhang et al., 2023; Dong et al., 2025; Li et al., 2026; OpenAI, 2024); static syntax checks reject programs with a `SyntaxError` before execution (Wen et al., 2024); proof kernels reject any output failing type-checking (Ren et al., 2025). The class of grammar-violating outputs is mathematically empty under these mechanisms.

Pattern B: strong empirical reductions with class-shift signature. Roughly fourteen citations report empirical reductions of 80–100% in a named error class, with the post-intervention failure log dominated by structurally different residual classes. Program-of-Thoughts on GSM8K (Chen et al., 2023): calculation errors drop from 30% of failures to 0%, residuals are 62% reasoning + 36% misunderstanding. OpenMedCalc (Goodell et al., 2025): “only interpretation errors were identified.” Acurai (Wood & Forbes, 2024): 100% hallucination elimination on RAGTruth (95% CI 91–100%), strong empirical rather than by-construction.

Pattern C: moderate reductions (60–80%) with residuals shifting outside the target class. Legal RAG (Dantart, 2026): fabricated citations $> 30\% \rightarrow < 0.2\%$. GPT-5 SimpleQA with web access (OpenAI, 2025): 47% $\rightarrow$ 9.6% (inter-condition, not within-condition). CRITIC (Gou et al., 2024a): toxic generation -79.2% .

The 28 citations, organised by axis and pattern.

Axis	Citation & targeted class	Pre → Post	Pattern
Verification	Ren et al. (2025): Invalid Lean proof	any → 0% by constr.	A
Format	OpenAI (2024): JSON schema violation	>60% → 0% by constr.	A
Format	Zhang et al. (2023): Tool-call syntax	21–100% → 0%	A
Format	Suresh et al. (2025): JSON parse failure	13–82% → 0%	A
Format	Dong et al. (2025): Multi-format errors	20–38% → 0%	A
Format	Li et al. (2026): Malformed tool calls	33–78% → 0%	A
Arithmetic	Chen et al. (2023): Calculation on GSM8K	30% → 0% of failures	B
Arithmetic	Goodell et al. (2025): Clinical arithmetic	“only interpretation errors”	B
Knowledge/RAG	Wada et al. (2025): RAG hallucinations	8% → 0% ($p = 0.012$)	B
Knowledge/RAG	Wood & Forbes (2024): Context-conflict hallucinations	100% → 0% on subset	B
Knowledge/RAG	Dantart (2026): Fabricated legal citations	>30% → <0.2%	C
Code Exec	Wen et al. (2024): SyntaxError (HumanEval)	5.76% → 0.01%	A
Code Exec	Li et al. (2022): False-positive submissions	62% → 4%	B
Code Exec	Shi et al. (2024): 5 code-bug classes	100% repair (5/6 classes)	B
Knowledge/RAG	Gao et al. (2023b): Citation grounding (ELI5)	≈50% w/o complete support	C
Knowledge/RAG	OpenAI (2025): Factual errors	47% → 9.6% w/web	C
Knowledge/RAG	Zakka et al. (2024): Clinical citation errors	44% → 9%	C
Arithmetic	Wang et al. (2025a): Arithmetic in medical reasoning	426 → 74 errors	B
Code Exec	Wen et al. (2024): NameError repair	22.7% → 2.3%	C
Perception	Gou et al. (2025): GUI grounding errors	16.2% → 73.3% acc.	B
Perception	Xie et al. (2025): GUI task failures	5% → 27% SR (5.4×)	B
Perception	Cheng et al. (2024): Element mislocation	5.2% → 53.4% (10.3×)	B
Perception	Liu et al. (2023): Chart-reading errors	38.2% → 67.6% (+29.4 pp)	B
Verification	Gou et al. (2024a): Toxic generation	−79.2%	C
Verification	Wang et al. (2024b): Reasoning step errors	28.6% → 43.5% w/rerank	B
Knowledge/RAG	Asai et al. (2024): Unsupported facts	55.5% → 22% error	B
Perception	Lu et al. (2024): Icon/element errors	70.5% → 93.8% (79% red.)	B
Knowledge/RAG	Mallen et al. (2023): Long-tail entity errors	≈80% → ≈50% failure	B

Capabilities are coarser than error classes. Several entries in the harvest reveal “two-for-one” reductions where one capability addresses multiple named clusters: a single Python interpreter removes the execution-error component of arithmetic, unit conversion, simple counting, list manipulation, and date arithmetic; constrained decoding eliminates by construction both format violations and the structural component of “missing required element”; code execution feedback strongly reduces SyntaxError, NameError, and most TypeError together; RAG strongly reduces factual hallucinations, fabricated citations, and

outdated information jointly. The practical capability library required is therefore smaller than the count of named error categories.

The twelve named clusters.

Cluster	Axis	Intervention	Before → After	Patt.
A. Arithmetic	Arithmetic	PAL Python (Gao et al., 2023a)	20.1% → 61.5%	B
B. Unit conversion	Arithmetic	Folded into A (Python)	n/a	n/a
C. Counting	Arithmetic	Explicit counter (gap)	n/a	gap
D. Format/schema	Format	DINGO/SLOT (Suresh et al., 2025; Wang et al., 2025b)	up to +68 pp; 99.5% acc.	A
E. Code logic/type	Code Exec	Reflexion/AgentCoder (Shinn et al., 2023; Huang et al., 2023)	80% → 96.3% pass@1	A/B
F. Multi-hop drift	Knowledge	Entity-grounded rewriter	≈5–10 pp	C
G. Reasoning step	Verification	Math-Shepherd (Wang et al., 2024b)	28.6% → 43.5%	B
H. Spec misinterpret.	Verification	Clarification (Niwa & Iso, 2024)	ROUGE-L +15	C
I.a Struct. missing	Format	Subsumed by D (constr. de-code)	n/a	A
I.b Semantic missing	Verification	Subsumed by G/H	n/a	C
J. Hallucination	Knowledge	RAG (Wood & Forbes, 2024)	non-zero → 0%	B/C
K. Refusal	Verification	POROver (Karaman et al., 2024)	57.6% → 82.1%	B
L. Tool/API	Format+Verif.	SAGE-Agent (Suri et al., 2025)	36.5% → 65.2%	B

Eight of twelve categories have a strong citation with double-digit percentage-point improvement; three (B, C, I) lack a clean single-cluster ablation; one (F) has consistent medium-strength evidence. Category I (missing required elements) splits mechanistically into I.a (structural, absorbed by D’s constrained decoders) and I.b (semantic, absorbed by G/H). Categories B (units) and C (counting) remain technical gaps: B is naturally folded into A; C is the smallest residual gap.

Additivity and its limits. Patel et al. (2026) demonstrate that stacking interventions targeting orthogonal failure modes can produce large compound reliability gains: their parallel-consensus framework yields a $14,700\times$ improvement over single-pass baseline, evidence for compound gains from decomposition plus consensus aggregation under their specific parallel-voting regime rather than a direct demonstration that heterogeneous cluster-targeted interventions compose additively without voting. Shang et al. (2024) similarly show supra-additive gains when modules target distinct failure modes. Le (2026) reports that schema-level and prompt-level instructions interact *non-additively* when sharing a prompt channel: interventions on orthogonal processing layers (decoding constraint vs. retrieval vs. training-signal vs. inference-time tool call) compose near-additively, while those sharing a channel may interfere.

The irreducible-semantic residual. Of the 17 ErrorAtlas categories, 13 are addressed under Patterns A, B, or C by one of the six capability axes; four are not: residuals of inappropriate refusal beyond preference optimisation, specification misinterpretation not closed by clarification, problem-decomposition reasoning bottlenecks, and a “user wanted something different” semantic remainder. These are classes where the failure is in choosing what to do, not executing it. Proposition 2’s prediction of $O(\log)$ rather than $O(0)$ residual reliability reflects exactly this irreducible core. The framework does not claim 100% coverage of all failures; it claims polylog-bounded *capability-eliminable* failures, with the named semantic residual as the floor.

C Counter-Evidence Re-Audits

Five prominent papers are routinely cited as evidence that LLM reliability decays steeply with length (Dziri et al., 2023; Kuratov et al., 2024; Kwa et al., 2025; Wan et al., 2026; Karpinska et al., 2024). A careful re-reading shows that *every one* of these papers decays over a variable distinct from raw token length n . Identifying the decay axis is not the same as dissolving the practical concern: where compositional graph size, fact count, or evidence scope grow with problem length, the framework predicts steep failure curves. The contribution is to identify which interventions help (capability provisioning along the actual decay axis) and which do not.

Dziri et al. (2023), Faith and Fate. GPT-4 multi-digit multiplication accuracy drops from 59% (3-digit) to 4% (4-digit) zero-shot, with the authors theorising “probability of incorrect predictions converges exponentially to ≈ 1 for abstract compositional tasks.” The decay variable is *compositional graph size* N , not raw token length: the 3×3 graph has on the order of d^2 partial products plus carries, and multi-digit multiplication is engineered so that $k_{\text{hard}} \approx N$, with every node in the computation graph a hard decision. For natural-language tasks where $k_{\text{hard}} \ll n$, Dziri’s regime is the *boundary case* in which our framework reduces to their result. A clean $(1 - \varepsilon)^N$ exponential cannot simultaneously reproduce 59% at 3-digit and 4% at 4-digit for any single per-node ε : the observed drop is locally steeper than per-node iid exponential, consistent with a finite catalogue of failure modes exhausting as N grows. *Where the framework concedes:* in adversarial compositional tasks engineered so $k_{\text{hard}} \approx N$, the framework reduces to Dziri’s regime and does not relieve it; the relocation is informational, not magical.

Kuratov et al. (2024). The abstract reports “performance declines sharply with increased reasoning complexity” and models “effectively utilise only 10–20% of the context.” Read in detail, BABILong varies two axes: context length n (0K to 10M tokens) and number of supporting facts k (QA1 = 1 fact to QA3 = 3 facts). The sharp decline is in k , not n . For QA1 (single-fact), most models “perform well up to 4,000 tokens”: a plateau, not exponential decay. The famous RAG-flat-across-length result (60% on single-fact QA *independent of context length*) is the cleanest possible demonstration that when relevant evidence is in window, length does not matter. Recurrent Memory Transformers maintaining performance to 50M tokens further confirms effective k is determined by architecture, not raw n . The framework’s concession is narrow but real: for multi-hop tasks whose required fact count grows with task complexity, BABILong’s sharp k -axis decay is exactly what the framework predicts happens, not a counter-example.

Kwa et al. (2025) (METR). Per-model success fits a logistic in $\log(\text{human-task-duration})$: $S(t) = \sigma(\beta(\log t - \log h))$. Logistic-in-log-length is *mathematically sublinear* in length itself. The 80%-horizon being $4\text{--}6\times$ shorter than the 50%-horizon is a steep within-model cliff but compatible by construction with the polylog result: a cliff at a specific capacity threshold is a manifold-transition signature. The famous exponential (capability doubling every seven months) is an *inter-model* claim about how the horizon h moves across generations, orthogonal to within-model decay shape. An honest qualifier: the within-model logistic-in-log cliff is steep on a practical scale. Calling it “sublinear in length” is technically correct but engineering-useful only for systems designed to operate well below the 50% horizon.

Wan et al. (2026), Fano-style upper bound. This paper theorises super-linear information-demand growth and identifies an “accuracy cliff” at capacity overflow: “when the task’s information demand surpasses the model’s output capacity, performance does not degrade gracefully but instead collapses sharply.” The behavioural prediction is a *threshold*, not smooth $(1 - \varepsilon)^n$. A cliff at a specific capacity threshold is exactly the manifold-transition behaviour our framework predicts at the boundary between covered and uncovered modes; Wan et al.’s mechanism (capacity overflow) is one specific cause, complementary to ours. *Where the framework concedes:* Wan documents a mechanism by which $|C|$ effectively exceeds the model’s coverage in a single forward pass; Pattern A interventions (constrained de-

coding, formal verification) are the kind of response the framework expects but does not automatically supply.

Karpinska et al. (2024). GPT-4o achieves 55.8% pair accuracy across 1,001 minimally-different true/false claim pairs about long fictional books (mean length 127K tokens). The paper reports performance by *evidence scope*, not by context length: 59.8% on sentence-level retrieval, 47.6% on passage-level, 41.6% on global reasoning. No per-context-length curve is reported. The decay axis is the number of evidence pieces that must be integrated, a k -axis quantity rather than an n -axis one. NoCha is therefore not counter-evidence to a sublinear-in- n claim. Evidence-scope decay is exactly the k -axis observation the framework predicts cannot be addressed by scaling raw context length; it requires retrieval or process supervision along the actual decay axis. The relocation is informational, not magical.

Unifying observation. Each of the steep-decay counter-papers we re-audit decays over a variable other than n : compositional graph size, fact count, log-time horizon, capacity threshold, or evidence scope. The apparent rapid decay is, in each case, in a quantity our framework already concentrates the action in (k_{hard} and $|C|$), not in raw sequence length. This is a relocation, not a dissolution. Where k_{hard} grows with task length (adversarial compositional structure, multi-hop fact chains, long horizons that force more decisions), reliability remains hard; the framework’s value is directing intervention toward capability provisioning along the actual decay axis rather than toward context-window or compute-budget expansion that does not help.

D Patch Evidence Detail

The body’s claim that domain patches cap the engineering problem rests on a triangulation of indirect evidence. None of the items below *measures* the patch ceiling $|C_D|$ directly; they support the weaker claim that model behaviour is strongly domain-dependent and that operational neighbourhoods occupy bounded regions of the model’s behavioural space.

Evidence type	What it supports	What it does not prove
Low intrinsic-dimensional structure in representations (Park et al., 2024; Li & Sarwate, 2025)	Domains occupy localised structure in model representation space	Does not measure failure-catalogue size $ C_D $
Cross-domain accuracy spreads on a single base model (Wang et al., 2024c)	Same model behaves differently across domains; performance is patch-dependent	Does not imply finite failure modes
Long-tail / popularity thresholds in knowledge tasks (Mallen et al., 2023; Kandpal et al., 2023)	Coverage depends on domain frequency and training exposure	Does not prove logarithmic mode discovery
Patch-specific intervention performance (across §4, Appendix B)	Local tools, schemas, and capability libraries change residual error structure	Does not imply universal cross-patch transfer of the same intervention library

The purpose of this evidence is motivational. It supports patch-indexing of $\sigma_D, A_D, \beta_D, |C_D|$, but the finite (or effectively capped) reachable catalogue remains an empirical modelling assumption to be measured per deployment, not derived from any of the rows above.

E Empirical Calibration Detail

Three published LLM error taxonomies anchor the mode-rate parameter σ in the body’s $\sigma \in [0.87, 1.85]$ range. The simple calibration uses $|C| \approx A + \sigma \ln T$ with $A = 0$; this is a deliberately conservative readout that ignores any positive intercept and uses endpoint counts rather than discovery curves.

Source	Approx. observed failures T	Named categories $ C $	Implied σ at $A = 0$	Role
ErrorAtlas (Ashury-Tahan et al., 2026)	$\gtrsim 10^4$	17	≈ 1.85	Conservative cross-domain anchor (used as planning value)
HumanEval categorisation (Wen et al., 2024)	comparable scale	8–12	≈ 0.87 –1.30	Code-domain anchor
MWPES-300K (Sun et al., 2025)	$\approx 3 \times 10^5$	15–20	≈ 1.2 –1.6	Math-domain anchor (largest corpus)

These are endpoint counts, not discovery curves: they tell us *how many* named categories a taxonomer assigned at a single corpus scale, not how the count grew with T . They do not prove logarithmic mode discovery. Their role in this paper is twofold: they motivate the empirical postulate of §3.2, and they identify the explicit empirical test that would either tighten the postulate or move the analysis to the Heaps variant of Appendix A.5: repeatedly subsample failures from a fixed deployment patch D and plot discovered modes $|C_{\text{seen},D}(\tilde{T})|$ against T .