THE SHARED MEMORY BOTTLENECK: CAPACITY LIMITS AND OPERATIONAL RETENTION WITH EXTERNAL SCRATCHPADS

Anonymous authors
Paper under double-blind review

ABSTRACT

While chain-of-thought prompting and scratchpads are frequently conceptualized as external memory for reasoning tasks, this perspective breaks down when a final answer must be generated from a bounded, jointly retained state. We formulate this limitation through a minimal constraint-satisfaction model where an agent must exactly output N independent binary constraints using a shared internal and external memory configuration yielding at most 2^B distinguishable states. Under this framework, we mathematically prove a tight phase law: $p^\star(N, B) = \min\{1, 2^{B-N}\}$. This Joint Memory Budget Theorem establishes that each independent constraint exceeding the retained memory budget exponentially halves the probability of exact success. We further generalize this limit with adversarial robustness and a min-entropy extension that separates raw prompt length from effective independent information. To empirically test the qualitative predictions of this theory, we introduce a two-call retention protocol that forces large language models to reconstruct constraint hands solely from their bounded written representations. Comprehensive evaluations across modern long-context and reasoning models reveal sharp, capacity-dependent performance cliffs. While variables such as scratchpad formatting, effective task entropy, and answer-time external access can shift this phase boundary, they do not eliminate it. Ultimately, we demonstrate that long-context and scratchpad reasoning are fundamentally bottlenecked not by nominal context length, but by the independent information jointly available at the time of answer generation.

1 INTRODUCTION

Chain-of-thought prompting, zero-shot reasoning, scratchpads, self-consistency, decomposition, and program-aided reasoning all expand the visible workspace available to a language model during inference Wei et al. (2022); Kojima et al. (2022); Nye et al. (2021). These methods can improve search, decomposition, and intermediate computation, and they have encouraged the interpretation of reasoning traces as a form of external memory. However, a longer trace does not by itself guarantee that all task-relevant information remains jointly recoverable when the final answer is generated.

Long-context architectures and benchmarks similarly increase the amount of text that can be presented to a model Dai et al. (2019); Bai et al. (2024). Yet effective context utilization can remain far below nominal context length: models exhibit positional biases, retrieval-like shortcuts, and sharp degradation on reasoning tasks that require information to be combined across distant or numerous constraints Liu et al. (2024); Hsieh et al. (2024a). These findings raise a basic question: why can a model observe all required information, produce a long reasoning trace, and still fail to satisfy all constraints simultaneously? Retrieval-augmented and memory-augmented systems address a related but distinct problem by supplying information during answer generation Lewis et al. (2020). Such systems may enlarge the information available at answer time. Our focus is the complementary regime in which the original constraints are no longer accessible and the final answer must be decoded from a bounded, jointly retained configuration. This distinction separates visible context, retained information, and newly retrieved information as different resources.

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.

We study this regime as a memory-capacity problem. Our central observation is that internal activations and an external scratchpad are not independent stores once the answer is generated from their joint final configuration. If that configuration has bounded distinguishable capacity, neither arbitrary intermediate computation nor a different allocation between internal and external memory can preserve unlimited independent information. This final-state perspective complements theoretical analyses of attention and chain-of-thought expressivity Hahn (2020); Merrill & Sabharwal (2024), but isolates a simpler question: how many distinct constraint assignments can remain exactly recoverable at answer time?

To answer this question, we introduce a minimal constraint-satisfaction model in which an agent observes a sequence of independent binary constraints and must reproduce the complete sequence exactly. The agent may freely process the constraints and use a scratchpad, but its internal state and scratchpad are treated jointly as one retained memory configuration. This abstraction separates information availability from search quality, prompting strategy, and computational power. Under this shared-memory abstraction, we establish an exact phase law for all-constraint success. Perfect reconstruction is possible when the retained capacity covers all independent constraints. Once the number of independent constraints exceeds that capacity, optimal exact-success probability decreases exponentially, with each additional independent constraint halving the achievable probability. The result is tight, applies to arbitrary agent policies, yields a worst-case coverage failure, and extends from uniform independent constraints to non-uniform inputs through min-entropy. The relevant quantity is therefore not raw sequence length, but the amount of independent information that must remain jointly available.

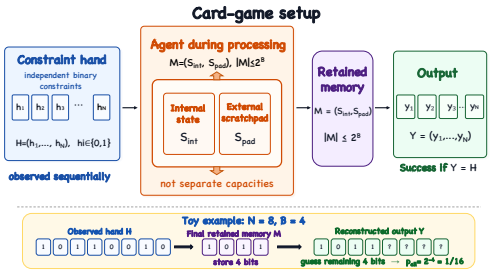


Figure 1: The answer-time information interface. Without the dashed channel, the original hand can influence the answer only through M. Fixed keys, model weights, and instructions are known in advance. Reintroducing an input-dependent response changes the channel being bounded.

We connect the theoretical abstraction to modern language models through a two-call retention protocol. In the first call, a model observes the full constraint hand and writes a bounded memory representation. In the second call, the original hand is removed, and the model must reconstruct it using only the written representation. Experiments across long-context and reasoning models reveal model-dependent retention cliffs. Structured key–value and compressed symbolic scratchpads move these cliffs by improving coding efficiency, while redundancy and latent structure move them by reducing effective entropy. Answer-time retrieval or tool access shifts the boundary differently by supplying information outside the retained state. Figure 1 illustrates the shared-memory abstraction. Our main contributions are as follows:

- We introduce a shared-memory formalization of scratchpad reasoning in which internal state and external workspace contribute to a single retained configuration at answer time.
- We derive a tight phase law for exact constraint satisfaction, together with worst-case and min-entropy extensions that distinguish surface length from effective independent information.
- We operationalize the theory using a two-call retention protocol and show that modern language models exhibit capacity-dependent cliffs, scratchpad-format shifts, entropy-dependent boundaries, and predictable gains from answer-time external access.

2 Related Work

2.1 Scratchpads and Chain-of-Thought Reasoning

Chain-of-thought prompting and scratchpads improve reasoning by exposing intermediate computation in natural language or auxiliary tokens Wei et al. (2022); Kojima et al. (2022); Nye et al. (2021). Subsequent work improves these traces through self-consistency, decomposition, executable reasoning, self-correction, tool feedback, and faithfulness analysis Wang et al. (2023b); Zhou et al. (2023);

Gao et al. (2023); Chen et al. (2023); Madaan et al. (2023); Shinn et al. (2023); Gou et al. (2024); Kamoi et al. (2024); Lyu et al. (2023); Paul et al. (2024). These studies focus on how intermediate traces support computation, search, or interpretation. In contrast, we study what information remains available when the final answer must be decoded from a bounded jointly retained state. Under this setting, scratchpads may improve organization and coding efficiency, but do not themselves enlarge the set of reachable outputs.

2.2 Long Context, Retrieval, and External Memory

Long-context models and retrieval-augmented systems expand the information available during inference, but nominal context length, effective context utilization, and retrieval access are distinct resources Sun et al. (2021); Liu et al. (2024); Hsieh et al. (2024b); Gao et al. (2024); Goldman et al. (2024); Modarressi et al. (2025); Li et al. (2025). Memory-augmented and retrieval-based models further provide information during answer generation rather than requiring all relevant content to remain in a fixed retained state Khandelwal et al. (2020); Lewis et al. (2020); Wu et al. (2022); Wang et al. (2023a). Our phase law applies when such answer-time access is removed and the final output depends only on retained information. Retrieval, tools, and verifiers can therefore shift the boundary by supplying information outside the bounded state.

2.3 Finite-Memory and Transformer-Theoretic Limits

Our analysis is related to information-theoretic, finite-memory, and streaming lower bounds, where a bounded state space limits how much input information can be preserved Cover & Thomas (2006); Alon et al. (1996). It also connects to theoretical work on the limitations and expressivity of transformers under attention, recurrence, sequence length, and bounded-state representations Hahn (2020); Yun et al. (2020); Chiang & Cholak (2022); Oren et al. (2024), as well as analyses of chain-of-thought expressivity, length, and memorization Merrill & Sabharwal (2024); Bavandpour et al. (2025); Yu et al. (2025). Unlike work characterizing the full computational power of transformers, we focus on the final stage of scratchpad reasoning. Once intermediate computation is compressed into at most 2^B jointly retained configurations, the decoder can produce at most 2^B distinct outputs and therefore exactly cover at most 2^B of the 2^N possible independent constraint hands.

3 A Shared-Memory Phase Law

We formalize scratchpad reasoning by the information available at answer time. Regardless of intermediate processing, a bounded final state can distinguish only finitely many inputs, turning exact constraint satisfaction into a coverage problem.

3.1 Shared-Memory Model and Exact Phase Law

Let a *constraint hand* be a binary string $H = (h_1, \ldots, h_N) \in \{0,1\}^N$, drawn uniformly unless otherwise stated. After observing H, the agent outputs $Y = (y_1, \ldots, y_N) \in \{0,1\}^N$ and succeeds only when $Y = H$. We measure all-constraint exact success by $p_{\text{all}} = \Pr[Y = H]$.

The agent may perform arbitrary computation, maintain an internal state, and write to an external scratchpad. We place no restriction on its intermediate computation. Instead, we assume that the final answer is decoded from a jointly retained state $M \in \mathcal{M}$, where internal state and scratchpad contribute to the same set of distinguishable final configurations. A shared memory budget of B bits therefore requires

$$|\mathcal{M}| \le 2^B. \tag{1}$$

Any deterministic policy can be represented by an encoder $f : \{0,1\}^N \to \mathcal{M}$ and a decoder $g : \mathcal{M} \to \{0,1\}^N$, such that $M = f(H)$ and $Y = g(M)$. The map f may summarize an arbitrarily complex interaction with the input and scratchpad. Randomized policies need not be considered separately because their success probability is an average over deterministic realizations. The scratchpad is not assumed to be useless; rather, it is counted together with the internal state as part of the information available at answer time.

We now characterize the optimal success probability under this final-state constraint.

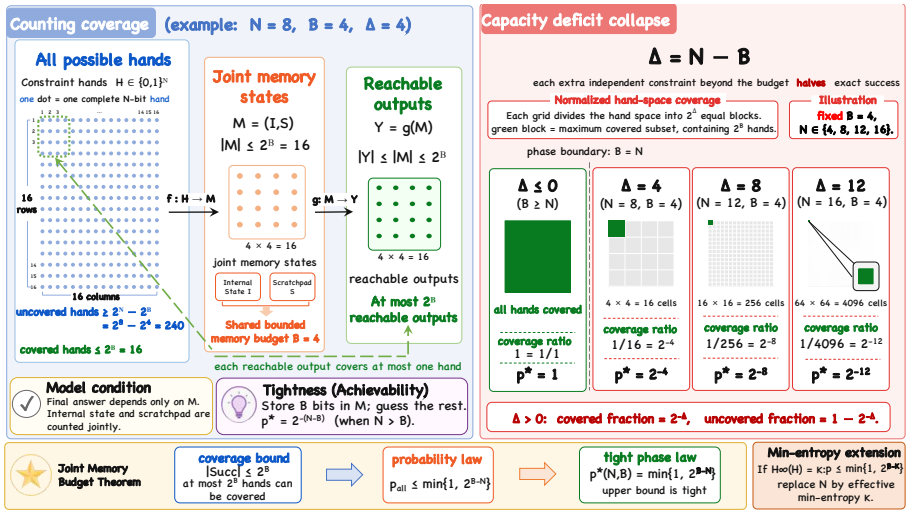


Figure 2: Counting interpretation of the Joint Memory Budget Theorem. At most 2^B retained states can induce at most 2^B exact outputs among 2^N possible hands. Consequently, perfect reconstruction is possible when $B \geq N$, whereas the optimal exact-success probability decays as $2^{-(N-B)}$ when $N > B$.

Theorem 1 (Joint Memory Budget Theorem)**.** *Let $H \sim \mathrm{Uniform}(\{0,1\}^N)$, and suppose the agent's final retained state satisfies Eq. (1). The optimal all-constraint success probability is*

$$p^\star(N,B) = \sup_{\substack{f,g:\\ |\mathcal{M}|\leq 2^B}} \Pr[g(f(H)) = H] = \min\left\{1, 2^{B-N}\right\}. \tag{2}$$

Thus, when $N > B$, $p^\star(N,B) = 2^{-(N-B)}$.

Proof sketch. For a fixed encoder–decoder pair, define the exact-success set and reachable outputs by

$$\mathcal{S} = \{h : g(f(h)) = h\}, \qquad \mathcal{Y} = g(\mathcal{M}). \tag{3}$$

If two successful hands shared one retained state, then $h_1 = g(f(h_1)) = g(f(h_2)) = h_2$; hence f is injective on $\mathcal{S}$, and

$$|\mathcal{S}| \leq |\mathcal{Y}| \leq |\mathcal{M}| \leq 2^B. \tag{4}$$

Uniformity then gives

$$\Pr[g(f(H)) = H] = \frac{|\mathcal{S}|}{2^N} \leq \min\{1, 2^{B-N}\}. \tag{5}$$

This condition is also sufficient: for $B < N$, assign distinct retained states to any 2^B hands (equivalently, store B selected bits and fix the remaining guesses), attaining $2^{-(N-B)}$; for $B \geq N$, store the complete hand. Randomization cannot improve the optimum because conditioning on its seed yields an ordinary deterministic policy and averaging preserves the bound. See the supplementary material for the necessary-and-sufficient finite-state proof, arbitrary-distribution optimum, and randomized-policy details.

Figure 2 visualizes the counting argument. A final state with B bits induces at most 2^B reachable outputs among 2^N possible hands. The result therefore characterizes the exact optimum under the retained-state budget, rather than the performance of a particular encoding strategy. The bound is independent of memory location. Internal-only, scratchpad-only, and mixed strategies obey the same law whenever they produce the same number of distinguishable final configurations. A scratchpad can improve organization, compression, or coding efficiency within the available budget, but it cannot enlarge the reachable output set unless it increases the information available at answer time.

3.2 Entropy, Measurement, and External Access

The binary uniform setting makes the phase boundary explicit, but the counting argument applies more broadly. For any deterministic policy with $N > B$, the reachable output set contains at most $2^B < 2^N$ hands. Therefore, there exists at least one hand $H^\star \notin \mathcal{Y}$ on which the agent necessarily fails. The limitation is thus a coverage shortfall, not merely an average-case effect.

Now suppose that H follows an arbitrary distribution P over $\{0,1\}^N$. Let its min-entropy be $\kappa = H_\infty(H) = -\log_2 \max_h P(H = h)$. Each hand then has probability at most $2^{-\kappa}$, while a policy can succeed on at most 2^B hands. Hence,

$$\Pr[g(f(H)) = H] \le \min\left\{1, 2^{B-\kappa}\right\}. \tag{6}$$

Equation (6) separates surface length from independent information. For N independent uniform bits, $\kappa = N$, and the exact law in Eq. (2) is recovered. If the constraints are redundant, predictable, or generated from a compact latent variable, then κ may be substantially smaller than the number of written constraints. The relevant boundary therefore depends on effective uncertainty rather than prompt length alone (see supplementary material for the full min-entropy proof and extensions to side information and larger alphabets).

Corollary 2 (Answer-Time Information Shift)**.** *Suppose the final answer is generated as $Y = g(M, E)$, where $|\mathcal{M}| \le 2^B$, the answer-time response E has at most 2^Q distinguishable values, and H has min-entropy κ. Then*

$$\Pr[Y = H] \le \min\left\{1, 2^{B+Q-\kappa}\right\}. \tag{7}$$

The proof is the same coverage argument applied to the joint state (M, E), which has at most 2^{B+Q} configurations (see supplementary material for proof). Thus, ideal answer-time access can shift the information boundary by at most Q bits; redundant responses, query-selection errors, and tool failures can make the empirical shift smaller.

The theoretical budget B is not directly observable in a language model. We therefore operationalize the final-state abstraction through a two-call retention protocol. In the first call, the model observes H and writes a bounded representation M. In the second call, the original hand is removed, and the model reconstructs $\widehat{H}$ from M alone. The protocol instantiates the Markov structure $H \to M \to \widehat{H}$, and its primary metric is $\widehat{p}_{\mathrm{all}} = \Pr[\widehat{H} = H]$.

At a fixed entropy level κ, we use

$$\widehat{B}_{\mathrm{point}}(\kappa) = \kappa + \log_2 \widehat{p}_{\mathrm{all}}(\kappa) \tag{8}$$

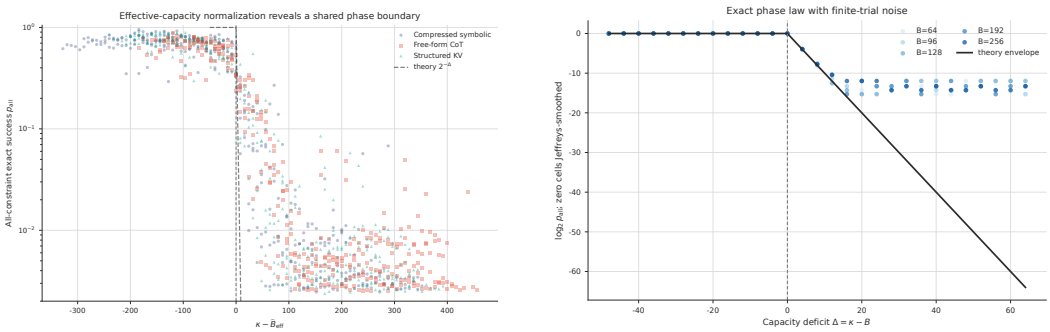
as a pointwise diagnostic of retained independent information. This quantity is not used as an independent normalization for the main empirical claims. When $\widehat{p}_{\mathrm{all}} = 0$, it is treated as censored below the finite-sample resolution. The primary evidence instead comes from directly measured retention boundaries, scratchpad-format ablations, entropy shifts, and external-access interventions.

The corresponding deficit satisfies the identity $\kappa - \widehat{B}_{\mathrm{point}}(\kappa) = -\log_2 \widehat{p}_{\mathrm{all}}(\kappa)$. Figure 3a visualizes the format-ablation measurements in these coordinates. The display is useful for reading operating ranges on a common information-deficit axis, but the apparent alignment is partly built into the transformation: its horizontal coordinate is algebraically determined by the plotted success rate. We therefore treat it as a descriptive diagnostic and not as independent evidence for the phase law.

4 Empirical Phase Law in Modern LLMs

The shared-memory law applies when the final answer depends only on M. If the model can retrieve the original constraints, query a database, invoke a verifier, or call an external tool during answer generation, the output instead depends on both retained and externally supplied information, which may be written as $Y = g(M, Z)$. Such access changes the resource model rather than violating the theorem.

This distinction separates three mechanisms that can move the empirical boundary. A more efficient scratchpad improves the encoding of information within M; redundant or structured constraints



(a) Descriptive capacity-normalization diagnostic.

(b) Finite-trial calibration of the exact phase law.

Figure 3: **Capacity-normalization diagnostics and phase-law calibration. Left:** The pointwise quantity $\widehat{B}_{\mathrm{point}}(\kappa) = \kappa + \log_2 \widehat{p}_{\mathrm{all}}(\kappa)$ summarizes the measured operating regimes; because the x-axis is algebraically related to the y-axis, it is not treated as independent evidence for the phase law. **Right:** The abstract model follows $\log_2 p_{\mathrm{all}} = -\max(0, \Delta)$, where $\Delta = \kappa - B$. Measured points track the theoretical envelope until zero-success cells reach the simulation resolution limit.

Table 1: **Model retention boundaries under compressed symbolic memory.** For the descriptive fit $\bar{p}(\kappa)$, $\widehat{\kappa}_\tau = \inf\{\kappa : \bar{p}(\kappa) \le \tau\}$. Values are approximate and rounded to the nearest 10 bits.

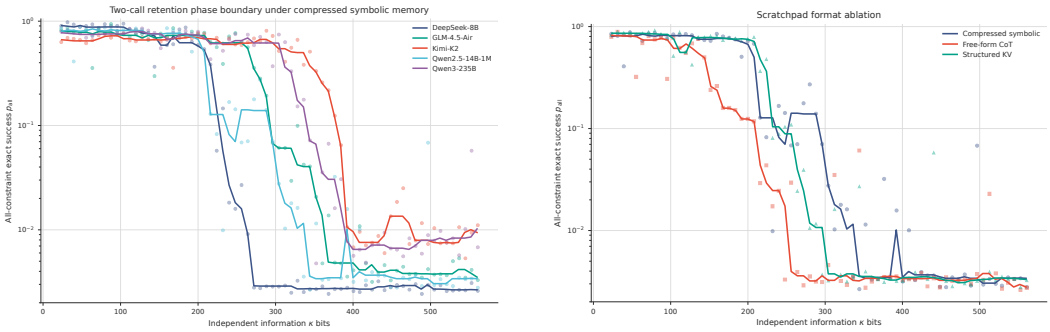
Model	$\widehat{\kappa}_{0.5}$	$\widehat{\kappa}_{0.1}$
DeepSeek-R1-Qwen3-8B	190	200
Qwen2.5-14B-1M	180	210
GLM-4.5-Air	240	260
Qwen3-235B-A22B	270	310
Kimi-K2-Thinking	310	340

reduce the effective entropy κ; and answer-time retrieval or tools introduce additional information Z. The shared-memory formulation makes these mechanisms explicit and identifies which resource is responsible for an observed improvement.

Before evaluating language models, we calibrate the finite-trial behavior of the exact law in the abstract model, where both B and κ are known. Figure 3b recovers the predicted boundary at $\kappa = B$ and the post-boundary slope -1 in $\log_2 p_{\mathrm{all}}$, up to the finite-sample floor. Simulation settings are provided in the supplementary material. We evaluate whether modern large language models exhibit the retained-capacity behavior predicted by Theorem 1. Following the two-call protocol introduced above, the first call observes a constraint hand H and writes a bounded memory representation M. The original hand is then removed, and the second call reconstructs $\widehat{H}$ from M alone. This procedure directly tests how much independent information remains recoverable at answer time.

Our primary metric is all-constraint exact success, $p_{\mathrm{all}} = \Pr[\widehat{H} = H]$. For selected analyses, we also report the pointwise retained-information diagnostic $\widehat{B}_{\mathrm{point}}(\kappa) = \kappa + \log_2 \widehat{p}_{\mathrm{all}}(\kappa)$. As discussed above, this quantity is interpreted only at a fixed operating point and is not used as an independent normalization of the main results. Malformed outputs, truncation, and refusals are never counted as successful reconstructions and are reported separately in Figure 5; detailed parsing and failure-labeling rules are provided in the supplementary material.

The five models in Table 1 were selected to cover distinct operating regimes rather than to construct a performance leaderboard. All models use the same constraint generators, reconstruction parser, and exact-success criterion. Unless otherwise stated, the first call is limited to 512 output tokens, the reconstruction call to 8192 tokens, and decoding is deterministic. We evaluate 40–100 independently generated hands per condition, depending on model cost. Full model snapshots, evaluation grids, prompt templates, parser rules, and raw per-condition counts are provided in the supplementary material. Figure 3a reports the capacity-normalization view only as a descriptive reparameterization and is not treated as independent evidence.



(a) Two-call exact reconstruction under compressed symbolic memory.

(b) Effect of scratchpad format on exact reconstruction.

Figure 4: **Exact reconstruction under compressed memory and scratchpad-format ablation. Left:** Two-call exact reconstruction under compressed symbolic memory. The horizontal axis measures independent information κ, and the vertical axis reports all-constraint exact success. Points denote measured conditions, while curves are descriptive smooth fits. **Right:** Effect of scratchpad format on exact reconstruction for Qwen2.5-14B-Instruct-1M. Structured key–value and compressed symbolic representations shift the retention boundary beyond that of free-form chain-of-thought, although all formats eventually collapse as κ increases.

Table 2: **Scratchpad-format boundary shifts on Qwen2.5-14B-1M.** The gain column reports the change in the 50%-success crossing relative to free-form CoT. Values are approximate and rounded to the nearest 10 bits.

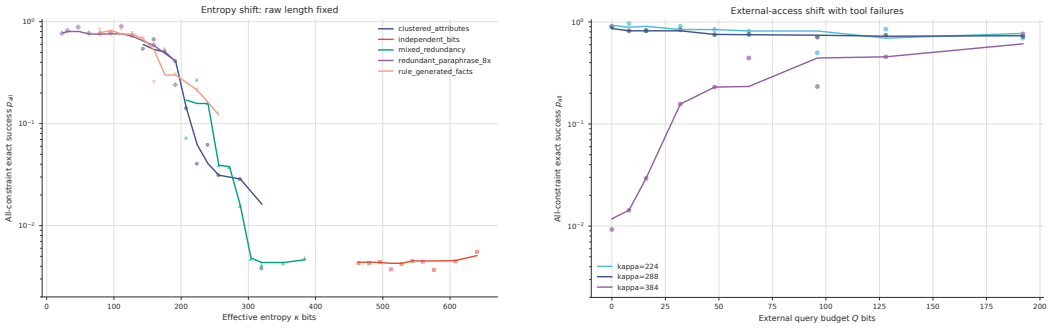
Format	$\widehat{\kappa}_{0.5}$	Gain	$\widehat{\kappa}_{0.1}$
Free-form CoT	130	–	190
Compressed symbolic	190	+60	220
Structured KV	200	+70	230

4.1 Retention Boundaries and Scratchpad Formats

Figure 4a reports exact reconstruction under compressed symbolic memory while varying the independent information κ in the constraint hand. Across all models, success remains high below a model-dependent threshold and declines sharply once the written representation can no longer preserve the required information reliably. Stronger long-context and agentic models move this threshold toward larger values of κ, but none eliminates the transition. The main difference across models is therefore the location of the empirical boundary rather than the existence of the boundary itself.

The fitted 50%-success crossing spans roughly 180–310 bits across models, a spread of about 130 bits, while the 10%-success crossing spans approximately 200–340 bits (Table 1). Kimi-K2-Thinking has the rightmost boundary in this evaluation, whereas DeepSeek-R1-Qwen3-8B and Qwen2.5-14B-1M cross near similar entropy levels despite their different nominal context regimes. This non-monotonicity cautions against treating context length or parameter scale alone as a retained-capacity estimate.

The shared-memory result does not imply that all scratchpad representations are equally effective. Different formats may preserve different amounts of information within the same writing budget. Figure **??** compares free-form chain-of-thought, structured key–value memory, and compressed symbolic memory under an otherwise identical protocol. Free-form chain-of-thought reaches the failure region earlier, indicating that verbose explanatory text leaves less of the available representation for preserving task constraints. Structured key–value and compressed symbolic formats retain exact reconstruction over larger values of κ, but both eventually exhibit the same qualitative collapse. Scratchpad design therefore changes coding efficiency and shifts the empirical threshold without removing the underlying retained-capacity limitation.



(a) Entropy shift **(b) External-access shift**

Figure 6: Factors that shift the empirical retention boundary. (a) With raw hand length fixed, redundancy and latent structure vary the effective entropy κ, moving the reconstruction threshold. (b) Answer-time query access supplies information beyond the retained representation M; larger query budgets improve reconstruction in high-entropy conditions.

Table 2 quantifies the format effect. On Qwen2.5-14B-1M, the estimated 50%-success boundary moves from about 130 bits with free-form CoT to 190 bits with compressed symbolic memory and 200 bits with structured KV, corresponding to gains of approximately 60 and 70 bits. At the 10%-success level, the corresponding crossings are about 190, 220, and 230 bits. The gain is therefore a boundary shift rather than removal of the low-success regime.

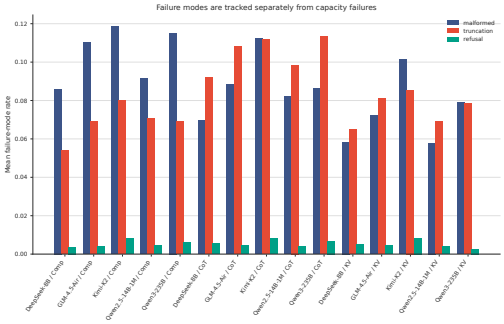


Figure 5: The answer-time information interface. Without the dashed channel, the original hand can influence the answer only through M. Fixed keys, model weights, and instructions are known in advance. Reintroducing an input-dependent response changes the channel being bounded.

To check whether the observed cliffs are dominated by output-format artifacts, Figure 5 reports malformed outputs, truncations, and refusals separately from wrong-value errors. Across the 15 model–format combinations, mean malformed-output rates are approximately 5.8–11.8%, truncation rates are 5.4–11.3%, and refusals remain below 0.8%. These diagnostics are non-negligible but remain separate from wrong-value capacity failures (see supplementary material for the complete parsing rules, primary-label priority, and raw records).

4.2 Boundary Shifts from Entropy and External Access

We next examine two interventions that move the empirical boundary by changing different resources. Figure 6(a) varies the effective entropy of the hand while holding the number of stated constraints fixed. Redundancy, paraphrasing, and latent generative structure reduce the amount of independent information that must be retained, allowing larger raw hands to remain reconstructable. In contrast, independently generated constraints become difficult at smaller raw lengths because their effective entropy is higher.

The curves do not collapse perfectly when plotted against κ. This is expected because min-entropy measures independent uncertainty but does not fully characterize how easily a language model can identify and encode the latent structure of a particular generator. Nevertheless, the observed transition tracks effective entropy more closely than the surface number of written constraints (see supplementary material for generator definitions and entropy-control details).

Figure 6(b) evaluates a different intervention. During reconstruction, the model receives an external query budget Q that returns information unavailable in the written representation M. At $\kappa = 384$, exact success rises from approximately 1.2% with $Q = 0$ to 15.5% with $Q = 32$, 44.3% with

$Q = 96$, and 61.2% with $Q = 192$. By contrast, the $\kappa = 224$ and 288 conditions already remain near 0.7–0.9 without substantial query-dependent gains. This pattern matches Corollary 2: external information is most valuable after retained memory alone becomes insufficient, although tool failures prevent the ideal Q-bit shift from being fully realized (see supplementary material for the query interface and budget accounting). Together, these results distinguish three mechanisms behind improved exact reconstruction. Stronger models and more efficient scratchpad formats preserve more information within the retained representation; redundant or structured inputs reduce the effective entropy that must be represented; and external queries provide additional information during answer generation. Although these mechanisms move the observed boundary in different ways, the results consistently support the shared-memory interpretation: exact success depends on how much independent information is jointly available when the final answer is produced.

5 Discussion and Conclusion

Discussion. The shared-memory view identifies the relevant resource as the information jointly available at generation time, rather than information that merely appears somewhere in the context. In the card-game model, this distinction is exact: once the interaction is compressed into a final retained configuration M with at most 2^B distinguishable states, exact satisfaction of κ bits of independent uncertainty is bounded by $p_{\text{all}} \leq \min\{1, 2^{B-\kappa}\}$. The two-call experiments operationalize this abstraction empirically. Different models and scratchpad formats preserve different amounts of information, but all-constraint success declines once the independent information in the hand exceeds what the written representation can reliably retain. This perspective reframes a common long-context failure mode. A model may have observed all relevant facts and may even have produced a long reasoning trace, yet still fail when the final answer must satisfy many independent constraints simultaneously. The bottleneck is therefore not merely access to individual tokens, but the joint retention and use of those constraints at answer time. In this sense, scratchpads are a shared resource: they can expose, organize, and compress intermediate reasoning, but when they remain part of the same bounded final state, they do not by themselves create unlimited retained capacity. This also explains why failures may appear abrupt rather than gradual. While the retained representation covers the required independent constraints, the model may remain reliable; once the required information exceeds the retained capacity, exact all-constraint success can collapse rapidly.

The experiments further clarify what moves this boundary. Stronger models or more efficient memory formats can increase the information preserved in M; structured key–value and compressed symbolic representations help because they use the available budget more effectively than unconstrained free-form traces. Task structure moves the boundary differently: when constraints are redundant, predictable, or generated from a compact latent seed, the surface number of facts overstates the independent uncertainty that must be retained. The relevant quantity is then the effective entropy κ, rather than prompt length alone. Finally, retrieval, tools, and verifiers change the setting by supplying information at answer time that is not contained in M. The phase-law view therefore separates improvements in retained information, effective entropy, and external access instead of treating them as a single undifferentiated increase in reasoning ability. This distinction matters because systems with similar observed accuracy may rely on different mechanisms: one may retain more information, another may compress the task more effectively, and a third may recover missing facts through external access. Without separating these mechanisms, it is difficult to explain why a method succeeds or predict when it will fail.

Conclusion. Scratchpads make reasoning visible, but visible workspace is not equivalent to retained capacity. When exact success requires many independent constraints to remain jointly available at answer time, a bounded final state induces a sharp capacity law. The theory characterizes this law in closed form, and the two-call experiments reveal its empirical signature in modern language models. This result does not imply that chain-of-thought, long context, or retrieval are ineffective. They help by changing specific resources: how efficiently information is encoded, how much independent entropy the task contains, or how much external information remains available during answer generation. Progress should therefore be assessed not only by producing longer reasoning traces, but also by determining whether those traces preserve the independent information required for the final answer. Identifying which resource has changed is essential for building systems that scale beyond merely longer traces.

REFERENCES

Noga Alon, Yossi Matias, and Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 20–29. ACM, 1996. doi: 10.1145/237814.237823. URL https://doi.org/10.1145/237814.237823.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.172. URL https://aclanthology.org/2024.acl-long.172/.

Alireza Amiri Bavandpour, Xinting Huang, Mark Rofin, and Michael Hahn. Lower Bounds for Chain-of-Thought Reasoning in Hard-Attention Transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3274–3306. PMLR, 2025. URL https://proceedings.mlr.press/v267/bavandpour25a.html.

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*, 2023. URL https://openreview.net/forum?id=YfZ4ZPt8zd.

David Chiang and Peter Cholak. Overcoming a Theoretical Limitation of Self-Attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7654–7664. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.527. URL https://aclanthology.org/2022.acl-long.527/.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2 edition, 2006. ISBN 978-0-471-24195-9. doi: 10.1002/047174882X. URL https://onlinelibrary.wiley.com/doi/book/10.1002/047174882X.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pp. 2978–2988. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1285. URL https://aclanthology.org/P19-1285/.

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 the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10764–10799. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Muhan Gao, TaiMing Lu, Kuai Yu, Adam Byerly, and Daniel Khashabi. Insights into LLM Long-Context Failures: When Transformers Know but Don't Tell. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7611–7625. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.447. URL https://aclanthology.org/2024.findings-emnlp.447/.

Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. Is It Really Long Context if All You Need Is Retrieval? Towards Genuinely Difficult Long Context NLP. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16576–16586. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.924. URL https://aclanthology.org/2024.emnlp-main.924/.

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*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Michael Hahn. Theoretical Limitations of Self-Attention in Neural Sequence Models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020. doi: 10.1162/TACL_A_00306. URL https://doi.org/10.1162/tacl_a_00306.

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 *First Conference on Language Modeling*, 2024a. URL https://openreview.net/forum?id=kIoBbc76Sy.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Found in the middle: Calibrating Positional Attention Bias Improves Long Context Utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14982–14995. Association for Computational Linguistics, 2024b. doi: 10.18653/V1/2024.FINDINGS-ACL.890. URL https://aclanthology.org/2024.findings-acl.890/.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When Can LLMs *Actually* Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024. doi: 10.1162/TACL_A_00713. URL https://doi.org/10.1162/tacl_a_00713.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklBjCEKvH.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Kuan Li, Liwen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Shuai Wang, and Minhao Cheng. LaRA: Benchmarking Retrieval-Augmented Generation and Long-Context LLMs - No Silver Bullet for LC or RAG Routing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36846–36867. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25dv.html.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/TACL_A_00638. URL https://doi.org/10.1162/tacl_a_00638.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful Chain-of-Thought Reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 305–329. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.IJCNLP-MAIN.20. URL https://aclanthology.org/2023.ijcnlp-main.20/.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

William Merrill and Ashish Sabharwal. The Expressive Power of Transformers with Chain of Thought. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNGlPh8Wh.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. NoLiMa: Long-Context Evaluation Beyond Literal Matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 44554–44570. PMLR, 2025. URL https://proceedings.mlr.press/v267/modarressi25a.html.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show Your Work: Scratchpads for Intermediate Computation with Language Models. *CoRR*, abs/2112.00114, 2021. URL https://arxiv.org/abs/2112.00114.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are Multi-State RNNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18724–18741. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.1043. URL https://aclanthology.org/2024.emnlp-main.1043/.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making Reasoning Matter: Measuring and Improving Faithfulness of Chain-of-Thought Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15012–15032. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.882. URL https://aclanthology.org/2024.findings-emnlp.882/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. Do Long-Range Language Models Actually Use Long-Range Context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 807–822. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.62. URL https://aclanthology.org/2021.emnlp-main.62/.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting Language Models with Long-Term Memory. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ebd82705f44793b6f9ade5a669d0f0bf-Abstract-Conference.html.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing Transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TrjbxzRcnf-.

Lijia Yu, Xiao-Shan Gao, and Lijun Zhang. Analyzing the Power of Chain of Thought through Memorization Capabilities. *CoRR*, abs/2511.01190, 2025. doi: 10.48550/ARXIV.2511.01190. URL https://doi.org/10.48550/arXiv.2511.01190.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are Transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

APPENDIX

A PROOF DETAILS

This supplementary material collects the proof of the shared-memory phase law and the extensions used in the main paper. We use the notation of the main paper: the hand is H, the final retained configuration is $M \in \mathcal{M}$, and the output is $Y = g(M)$. The only structural assumption is the final-state factorization $H \xrightarrow{f} M \xrightarrow{g} Y$, with $|\mathcal{M}| \le 2^B$. All logarithms are base two.

A.1 PROOF OF THE EXACT PHASE LAW

The main theorem can be stated exactly for any finite number K of retained configurations. This formulation makes both necessity and sufficiency explicit and removes integer-rounding ambiguity.

Lemma 3 (Success-set injection)**.** *Let $\mathcal{H}$ be a finite hand space and let $f : \mathcal{H} \to \mathcal{M}$, $g : \mathcal{M} \to \mathcal{H}$ be deterministic. Define $\mathcal{S} = \{h \in \mathcal{H} : g(f(h)) = h\}$. Then the restriction $f|_{\mathcal{S}}$ is injective. Consequently,*

$$|\mathcal{S}| \le |\mathcal{M}|.$$

Proof. Take $h_1, h_2 \in \mathcal{S}$ and suppose $f(h_1) = f(h_2) = m$. Since both hands are reconstructed exactly,

$$h_1 = g(f(h_1)) = g(m) = g(f(h_2)) = h_2.$$

Thus distinct successful hands cannot share a retained configuration, proving injectivity and the cardinality bound. □

Theorem 4 (Exact finite-state optimum)**.** *Let $H \sim \mathrm{Uniform}(\{0,1\}^N)$, and suppose $|\mathcal{M}| \le K$ for an integer $K \ge 1$. Then*

$$\sup_{f,g} \Pr[g(f(H)) = H] = \frac{\min\{K, 2^N\}}{2^N}.$$

Equivalently, under a real-valued budget B with $|\mathcal{M}| \le 2^B$, the exact optimum is

$$p^\star(N, B) = \min\left\{1, \frac{\lfloor 2^B \rfloor}{2^N}\right\}.$$

For integer B, this reduces to $p^\star(N, B) = \min\{1, 2^{B-N}\}$.

Proof. For any policy, Lemma 3 gives $|\mathcal{S}| \le \min\{K, 2^N\}$. Uniformity therefore implies

$$\Pr[g(f(H)) = H] = \frac{|\mathcal{S}|}{2^N} \le \frac{\min\{K, 2^N\}}{2^N}.$$

To attain the bound, choose any subset $\mathcal{C} \subseteq \{0,1\}^N$ of size $\min\{K, 2^N\}$, assign one distinct retained state m_h to each $h \in \mathcal{C}$, and set $g(m_h) = h$. Map every hand outside $\mathcal{C}$ to an arbitrary existing state. The policy succeeds exactly on $\mathcal{C}$, so its success probability equals the upper bound. This construction proves sufficiency as well as necessity. □

A more concrete achieving policy for integer $B < N$ stores any fixed B coordinates and uses fixed guesses for the remaining $N - B$ coordinates. It succeeds exactly when all unstored bits match those guesses, with probability $2^{-(N-B)}$. When $B \ge N$, it stores the complete hand.

Randomized policies. Let R denote all policy randomness. Conditional on $R = r$, the encoder and decoder are deterministic and Theorem 4 applies. Hence

$$\begin{aligned}
&\Pr[g_R(f_R(H)) = H] \\
&\quad = \mathbb{E}_R[\Pr[g_r(f_r(H)) = H \mid R = r]] \\
&\quad \le \frac{\min\{K, 2^N\}}{2^N}.
\end{aligned}$$

This argument also covers public randomness available to both calls: each fixed public seed obeys the same bound. If a sampled seed or random transcript must be retained until answer time, it is part of the final configuration and must be counted in K. Thus randomization cannot enlarge the optimal coverage under a fixed answer-time state budget.

The internal-memory-only case is recovered by taking the scratchpad component of M to be empty. Conversely, a scratchpad that remains accessible to the decoder is part of M; splitting $M = (M_{\text{int}}, M_{\text{pad}})$ changes no step of the proof because only the number of joint configurations matters.

A.2 Distributional and Structural Extensions

Adversarial hands. The uniform theorem is also a coverage statement. For any deterministic policy, the reachable output set $\mathcal{Y}$ has size at most 2^B. If $B < N$, then $|\mathcal{Y}| < 2^N$, so some hand $H^\star \in \{0,1\}^N \setminus \mathcal{Y}$ is never output by the policy. An adversary who selects this hand forces exact failure. Thus the limitation is not merely an artifact of averaging over random hands.

Exact optimum for an arbitrary distribution. Let $\mathcal{H} = \{h_1, \ldots, h_L\}$, and order the hand probabilities as $p_{(1)} \geq p_{(2)} \geq \cdots \geq p_{(L)}$. For an integer state budget K, the exact optimum is

$$\sup_{f,g:\,|\mathcal{M}|\leq K} \Pr[g(f(H)) = H] = \sum_{i=1}^{\min\{K,L\}} p_{(i)}. \tag{9}$$

Necessity follows because Lemma 3 limits the success set to at most K hands, whose total probability cannot exceed the mass of the K most likely hands. Sufficiency follows by assigning distinct states to those K hands and decoding them exactly. Equation (9) is the distribution-specific necessary-and-sufficient characterization; the min-entropy statement below is a distribution-free corollary.

Min-entropy corollary. Let

$$\kappa = H_\infty(H) = -\log_2 \max_{h\in\mathcal{H}} P(H = h).$$

Every hand has probability at most $2^{-\kappa}$. With $K = \lfloor 2^B \rfloor$, Eq. (9) yields

$$\Pr[g(f(H)) = H] \leq \min\{1, K2^{-\kappa}\} \leq \min\{1, 2^{B-\kappa}\}.$$

The min-entropy bound is tight whenever the distribution has at least K outcomes of equal maximal mass and the remaining mass is arranged consistently with normalization. It can be loose for highly nonuniform distributions, in which case Eq. (9) is the sharper statement.

Side information. Suppose both encoder and decoder observe side information O, and the state budget $|\mathcal{M}| \leq K$ holds for every value of O. For each o, order the conditional probabilities as $p_{(1|o)} \geq p_{(2|o)} \geq \cdots$. Applying Eq. (9) conditionally gives the exact bound

$$\Pr[g(f(H,O),O) = H] \leq \mathbb{E}_O \left[\sum_{i=1}^{\min\{K,|\mathcal{H}|\}} p_{(i|O)} \right].$$

A convenient worst-case conditional min-entropy is

$$\kappa_O^{\text{wc}} = -\log_2 \max_{o:P(o)>0,\,h} P(H = h \mid O = o).$$

It implies

$$\Pr[g(f(H,O),O) = H] \leq \min\{1, 2^{B-\kappa_O^{\text{wc}}}\}.$$

This formalizes how known templates, generation rules, or other priors reduce the uncertainty that must be retained. If the decoder does not observe O, it cannot in general exploit the conditional reduction; the relevant state must then include whatever information about O remains available at answer time.

Larger alphabets. The binary presentation is for clarity. If $H \in \mathcal{A}^N$ consists of independent uniform values over an alphabet $\mathcal{A}$, then the hand has min-entropy $\kappa = N \log_2 |\mathcal{A}|$. The same counting argument gives

$$\Pr[g(f(H)) = H] \le \min\left\{1, 2^{B - N \log_2 |\mathcal{A}|}\right\}.$$

This form is useful when experimental constraints carry multi-bit values rather than binary values.

A.3 Changing the Resource Model

The theorem assumes that the final answer is generated only from the retained state M. External information changes the available answer-time transcript rather than violating the counting argument.

Theorem 5 (Augmented answer-time state)**.** *Let H have min-entropy κ. Suppose the final output is $Y = g(M, E)$, with $|\mathcal{M}| \le K_M$ and $|\mathcal{E}| \le K_E$. Then*

$$\Pr[Y = H] \le \min\{1, K_M K_E 2^{-\kappa}\}.$$

In particular, if $K_M \le 2^B$ and $K_E \le 2^Q$, then

$$\Pr[Y = H] \le \min\{1, 2^{B+Q-\kappa}\}.$$

For a uniform N-bit hand and an unrestricted augmented channel with $K_M K_E$ joint configurations, the exact optimum is $\min\{1, K_M K_E / 2^N\}$.

Proof. Treat the complete answer-time transcript as a single state $T = (M, E)$. Its support has size at most $K_M K_E$. Applying Lemma 3 to the composite encoder $H \mapsto T$ and decoder $T \mapsto Y$ limits the success set to at most $K_M K_E$ hands. Multiplying by the maximum hand probability gives the min-entropy bound. Under a uniform distribution, assigning distinct joint transcripts to any $\min\{K_M K_E, 2^N\}$ hands attains the stated optimum. □

The upper bound remains valid for adaptive retrieval or multi-step tool use: E may denote the entire transcript available when the answer is produced, and only the number of distinguishable final transcripts matters. Tightness, however, requires an unrestricted channel capable of realizing arbitrary transcript assignments. A fixed retriever, noisy tool, verifier, or query policy may provide fewer than Q independent useful bits, so empirical gains can be strictly smaller than the ideal shift. Correlated or redundant responses should therefore be counted by their distinguishable answer-relevant information, not by raw response length.

Exact success versus per-constraint accuracy. Let C_i be the event that constraint i is reconstructed correctly, with $q_i = \Pr[C_i]$. Without assumptions on dependence,

$$\max\left\{0, 1 - \sum_{i=1}^{N} (1 - q_i)\right\} \le \Pr\left[\bigcap_{i=1}^{N} C_i\right] \le \min_i q_i.$$

If the correctness events are independent, exact success becomes $\prod_{i=1}^{N} q_i$, and equals q^N when all marginals are q. Thus high average per-constraint accuracy does not by itself imply high all-constraint success; the error-dependence structure matters. This is why the experiments use $p_{\text{all}} = \Pr[\widehat{H} = H]$ as the primary metric and report per-constraint or parser diagnostics separately.

B Scope of the Theoretical Model

The theorem is a final-state distinguishability result. It abstracts away the internal mechanics of an agent and asks only how many distinct configurations can remain available when the final answer is produced. This section clarifies how that abstraction should be read when relating the theorem to language models and to the two-call experiments.

B.1 The Final-State Abstraction

The mathematical model assumes that the final answer is produced from a retained configuration M: $Y = g(M)$. The interaction before M is formed may be arbitrarily complex. The agent may read constraints sequentially, write intermediate notes, compress summaries, use chain-of-thought tokens, or perform any computation. All such processing is absorbed into the map $f : H \mapsto M$. The theorem does not limit the computation used to construct M; it limits the number of distinguishable final states from which the answer can be generated.

The memory budget is shared. Internal activations, scratchpad content, cached summaries, or any other retained representation contribute to the same final configuration. The assumption $|\mathcal{M}| \leq 2^B$ therefore means that these resources jointly expose at most 2^B distinguishable states at answer time. A scratchpad that remains available at generation time is part of M. If an external store is available without being counted in M, then the final answer is being produced from a larger state.

The proof is stated for finite distinguishable configurations. Real neural systems use continuous activations, key–value caches, logits, and finite-precision arithmetic. The relevant abstraction is not that neural memory is literally stored as discrete bits, but that exact reconstruction of a discrete hand requires separating possible hands into distinguishable final states. In the experiments, this distinction is handled operationally: retained capacity is measured by what the model can reconstruct in the two-call protocol, rather than by directly inspecting its internal representation.

B.2 Entropy, Priors, and Structure

The cleanest theorem uses N independent uniform binary constraints, where the hand has N bits of min-entropy. This is a diagnostic worst case, while natural-language tasks often contain dependencies and semantic structure. When constraints are redundant, generated by a compact rule, or predictable from side information, the relevant quantity is the remaining uncertainty of the hand, denoted κ, rather than the number of written constraints.

This is also how prior knowledge enters the model. A pretrained model may exploit familiar formats, semantic regularities, or a known generation rule. Such information can reduce the effective entropy that must be retained, or it can improve the encoding used for M. The counting argument then applies after accounting for that side information: the remaining uncertainty must still be represented, queried, or guessed. Formally, using the worst-case conditional min-entropy $\kappa_O^{\mathrm{wc}} = -\log_2 \max_{o,h} P(H = h \mid O = o)$, the same bound applies with κ_O^{wc} in place of N.

The entropy-shift experiments are designed to probe this distinction. They fix the number of stated constraints while varying the effective entropy through redundancy and latent structure. The resulting boundary need not depend only on κ: real models also differ in how easily they recognize and encode a particular structure. The min-entropy view should therefore be read as a resource account of the independent information that must be retained, not as a complete behavioral model of every prompt format.

B.3 External Information and Empirical Interpretation

The strict shared-memory theorem applies when the final answer is determined by M. If a system queries a retriever, calls a verifier, re-reads the original constraints, or uses a tool that returns new information during answer generation, then the effective state is augmented. Writing this as $Y = g(M, E)$, where E is the answer-time external response, makes the accounting explicit. If E contributes at most Q bits of independent information, the same counting argument gives a boundary of the form

$$p_{\mathrm{all}} \leq \min\{1, 2^{B+Q-\kappa}\}.$$

External access is therefore modeled as additional answer-time information, or as a way to reduce the uncertainty that must be stored in M.

The empirical protocol is a diagnostic instantiation of this abstraction. The first call asks the model to write an explicit memory representation, and the second call asks it to reconstruct the hand from that representation alone. This does not claim to expose the model's physical number of bits. It measures an operational quantity: how much independent information survives through the written

Table 3: Model and decoding settings. "Context used" is the maximum context length allocated in our experiments, not necessarily the model's architectural maximum. "Hands/point" is the number of random hands evaluated at each entropy level and condition. The snapshot / endpoint field records the model identifier or API snapshot used for reproducibility.

Model	Access	Context	Max gen.	Decoding	Hands/pt	Snapshot / endpoint
DeepSeek-R1-0528-Qwen3-8B	local	32K	8K	$T = 0,\ p = 1.0$	100	`deepseek-ai/DeepSeek-R1-0528-Qwen3-8B`
Qwen2.5-14B-Instruct-1M	local	128K	8K	$T = 0,\ p = 1.0$	100	`Qwen/Qwen2.5-14B-Instruct-1M`
GLM-4.5-Air	local/API	128K	8K	$T = 0,\ p = 1.0$	50	`zai-org/GLM-4.5-Air`
Qwen3-235B-A22B-Instruct-2507	API	128K	8K	$T = 0,\ p = 1.0$	50	`Qwen/Qwen3-235B-A22B-Instruct-2507`
Kimi-K2-Thinking	API	128K	16K	$T = 0,\ p = 1.0$	40	`moonshotai/Kimi-K2-Thinking`

Table 4: Memory and output budgets used in the two-call protocol.

Condition	Writing budget	Reconstruction budget
Main retention	512 tokens	8192 tokens
Free-form CoT	512 tokens	8192 tokens
Structured KV	512 tokens	8192 tokens
Compressed symbolic	512 tokens	8192 tokens
Entropy shift	512 tokens	8192 tokens
External access	512 tokens + Q bits	8192 tokens

representation and can be used for exact reconstruction. For this reason, the main metric is all-constraint exact success $p_{\text{all}} = \Pr[\hat{H} = H]$. Partial per-constraint accuracy, malformed-output rate, truncation, and refusal rate are useful diagnostics, but they measure different failure modes.

C Experimental Setup

This appendix records the experimental settings used for the two-call retention experiments. The main text reports compact quantitative boundary tables; here we give the full decoding setup, evaluation grids, hand generators, memory budgets, and plotting conventions. Unless otherwise stated, all conditions use the same exact-success criterion: a trial succeeds only when the reconstructed hand $\hat{H}$ matches the ground-truth hand H on every required key and value.

C.1 Models, Decoding, and Evaluation Grid

Table 3 gives the model-level settings used in the reported experiments. We use deterministic decoding for the main runs, with one output per hand and no self-consistency voting. Local full-grid models are evaluated on more random hands per condition; larger API or frontier models are evaluated on a reduced boundary-focused grid.

Table 4 reports the first-call memory-writing budget and the second-call reconstruction budget. The first-call budget controls the retained representation M; the second-call budget is only used to allow the final JSON object to be emitted.

Table 5: Evaluation grid for the main-paper and supplementary figures. All reported metrics use all-constraint exact success p_{all}, except the failure-mode diagnostic.

Figure	Models / conditions	Grid	Metric
Main paper: model retention	all models; compressed symbolic memory	$\mathcal{K}_{\text{full}}$ for full-grid models; $\mathcal{K}_{\text{red}}$ otherwise	p_{all}
Main paper: format ablation	Qwen2.5-14B-Instruct-1M; CoT / KV / compressed	$\mathcal{K}_{\text{full}}$	p_{all}
Main paper: entropy shift	entropy generators with 512 stated items	target $\kappa \in \{96, 128, 160, 192, 224, 256, 320, 384, 448, 512, 640\}$	p_{all}
Main paper: external-access shift	external access at selected κ levels	$Q \in \{0, 8, 16, 32, 48, 64, 96, 128, 192\}$	p_{all}
Main paper: phase-law calibration	abstract card-game simulation	$\Delta = \kappa - B \in [-12, 16]$	$\log_2 p_{\text{all}}$
Main paper: capacity normalization	descriptive pointwise normalization	same measured points as format ablation	diagnostic
Main paper: failure modes	malformed / truncated / refusal rates	all main model–format conditions	failure rate

The full and reduced entropy grids are

$$\begin{aligned}\mathcal{K}_{\text{full}} &= \{32, 64, 96, 128, 160, 192, 224,\\ &\qquad 256, 288, 320, 384, 448, 512, 640\},\\ \mathcal{K}_{\text{red}} &= \{64, 128, 192, 256, 320,\\ &\qquad 384, 512, 640\}.\end{aligned}$$

For fixed-constraint experiments, the hand length is fixed at 512 items, and κ is varied by changing the value alphabet, redundancy, or latent generator.

Table 5 summarizes which settings are used for each reported figure.

C.2 Hand Generators and Entropy Control

Each hand is a list of key–value constraints of the form $\texttt{key}_i \mapsto v_i$. Keys are synthetic nonce identifiers, such as `k0001`, and carry no semantic information. Values are generated by one of the mechanisms below. The target entropy κ is measured in bits.

Independent binary hands. For binary hands, each $v_i \in \{0, 1\}$ is sampled independently and uniformly. A hand with N such values has $\kappa = N$. This generator is the closest empirical analogue of the theorem's clean card-game setting.

Multi-symbol hands. For multi-symbol hands, values are drawn independently from an alphabet $\mathcal{A}$. When the distribution is uniform, a hand with N values has $\kappa = N \log_2 |\mathcal{A}|$. This is used when the number of stated items is fixed but the target entropy exceeds the number of items.

Redundant hands. A latent seed Z is sampled first, and multiple stated constraints are generated as deterministic or near-deterministic functions of Z. This creates many written constraints with lower effective entropy. The model can succeed by retaining the latent information rather than memorizing every stated item independently.

Rule-generated hands. A compact rule is sampled from a finite family and then applied to nonce keys or indices to generate values. This tests whether the model can retain a rule-like representation. The effective entropy is the entropy of the latent rule plus any residual independent noise.

Clustered attributes. Keys are partitioned into clusters, and values within a cluster share latent attributes plus optional independent noise. This produces partial redundancy: the cluster structure lowers the entropy, while the residual noise still requires item-level retention.

In the entropy-shift experiment, the number of stated constraints is fixed at 512, but the value alphabet and latent generator vary. Therefore κ is not identified with the number of written items; it is the independent information in the generated hand.

C.3 Estimation, Smoothing, and Reporting

For each condition, we estimate exact success as

$$\widehat{p}_{\mathrm{all}} = \frac{s}{n},$$

where s is the number of exact reconstructions and n is the number of evaluated hands. For log-scale visualizations and finite-resolution diagnostics, zero-success cells are displayed using Jeffreys smoothing,

$$\widetilde{p}_{\mathrm{all}} = \frac{s + 1/2}{n + 1}.$$

The smoothed value is used only for plotting on logarithmic axes; the raw exact-success rate remains the reported metric.

Solid curves in the main figures are descriptive smooth fits to the measured points. They are included to make the boundary easier to read and are not used as evidence independent of the raw measurements. For monotone retention curves we fit the two-parameter logistic form

$$\bar{p}(\kappa) = \frac{1}{1 + \exp\{a(\kappa - b)\}}, \qquad a > 0,$$

by binomial maximum likelihood using the observed success counts (s, n). The positivity constraint enforces a nonincreasing retention curve. For non-monotone diagnostic plots we use locally weighted smoothing. All claims in the main text are based on the measured points and observed boundary shifts, not on the smoothing procedure.

For the compact numerical summaries in the main paper, the descriptive crossing entropy is

$$\begin{aligned}\widehat{\kappa}_\tau &= \inf\{\kappa : \bar{p}(\kappa) \le \tau\} \\ &= \widehat{b} + \frac{1}{\widehat{a}} \log\left(\frac{1}{\tau} - 1\right), \quad \tau \in \{0.5, 0.1\}.\end{aligned}$$

A crossing outside the evaluated entropy range is reported as left- or right-censored rather than extrapolated. Reported values are rounded to the nearest 10 bits because the evaluation grid is coarser than the fitted precision. They are readable summaries of measured curves, not independently estimated physical memory sizes. A bootstrap interval is obtained by resampling hands within each condition, refitting the logistic curve, and recomputing the crossing in each of 1,000 replicates.

Uncertainty intervals, when shown, are computed by nonparametric bootstrap over random hands within each condition. Unless otherwise stated, we use 1,000 bootstrap resamples and report 95% intervals. For pairwise comparisons of scratchpad formats, the same random hands are reused across formats whenever possible, so paired bootstrap intervals can be computed over hand-level differences.

Pointwise retained-information diagnostic. The quantity

$$\widehat{B}_{\mathrm{point}}(\kappa) = \kappa + \log_2 \widehat{p}_{\mathrm{all}}(\kappa)$$

is reported only as a pointwise diagnostic. It is not used to produce an independent phase-collapse claim in the main text, because plotting $\widehat{p}_{\mathrm{all}}$ against $\kappa - \widehat{B}_{\mathrm{point}}(\kappa)$ would be an algebraic reparameterization of the same point. The supplementary capacity-normalization section includes this visualization only as a descriptive diagnostic.

Failure categories. Malformed outputs, truncation, and refusals are logged separately from wrong-value errors. They are counted as failures for exact reconstruction, but their aggregate rates are reported separately in the main paper. This separation checks that phase boundaries are not artifacts of parser failures or output length limits.

Table 6: Descriptive crossing entropies reported in the main paper. Values are rounded to the nearest 10 bits; the underlying measured points and s/n counts remain the primary records.

Condition	$\widehat{\kappa}_{0.5}$	$\widehat{\kappa}_{0.1}$
DeepSeek-R1-Qwen3-8B / compressed	190	200
Qwen2.5-14B-1M / compressed	180	210
GLM-4.5-Air / compressed	240	260
Qwen3-235B-A22B / compressed	270	310
Kimi-K2-Thinking / compressed	310	340
Qwen2.5-14B-1M / free-form CoT	130	190
Qwen2.5-14B-1M / structured KV	200	230

Table 7: Raw-result summary fields provided for each plotted condition.

Field	Meaning
model	evaluated model name
format	memory format
κ	effective entropy level
Q	external query budget, if applicable
s/n	exact successes / total trials
malformed	malformed-output rate
truncated	truncation rate
refusal	refusal rate
seed range	random hand generator seeds

Raw-result records. For each plotted point, we store the hand generator seed, model name, memory format, entropy level κ, query budget Q when applicable, raw model output, parsed reconstruction, exact-success indicator, and failure label. The supplementary artifact reports aggregate s/n counts for every plotted condition.

D Additional Diagnostics

D.1 Abstract-Model Calibration Details

The main paper reports the finite-trial calibration of the exact phase law in the abstract card-game model, where B and κ are known. The simulation evaluates

$$\log_2 p_{\mathrm{all}} = -\max(0, \kappa - B)$$

over $\Delta = \kappa - B \in [-12, 16]$, using

$$B \in \{64, 96, 128, 192, 256\}.$$

Each point is estimated from independent random hands using the same exact all-constraint criterion as the language-model experiments. Zero-success cells are shown only through Jeffreys smoothing on the logarithmic axis; they are treated as censored at the finite-trial resolution rather than as exact zeros. The displayed agreement therefore tests the implementation and finite-sample behavior of the counting law, not the language-model capacity assumption.

D.2 Capacity-Normalization Derivation and Limitations

The main text does not use pointwise effective-capacity normalization as independent evidence. If

$$\widehat{B}_{\mathrm{point}}(\kappa) = \kappa + \log_2 \widehat{p}_{\mathrm{all}}(\kappa),$$

then

$$\kappa - \widehat{B}_{\mathrm{point}}(\kappa) = -\log_2 \widehat{p}_{\mathrm{all}}(\kappa).$$

A plot using this pointwise quantity on the x-axis and $\widehat{p}_{\mathrm{all}}$ on the y-axis is therefore a descriptive reparameterization, not a held-out test of the phase law.

The main paper now displays this diagnostic alongside the operational definition of $\widehat{B}_{\text{point}}$. Its role is deliberately limited. It can make the range of observed deficits easier to compare across scratchpad formats, but it cannot establish collapse independently because each plotted x-coordinate is computed from the same y-coordinate. A valid independent test would require a capacity estimate obtained from separate data or a held-out intervention, followed by a prediction of success on new entropy levels.

D.3 Failure-Mode Controls

Malformed outputs, truncation, and refusals are tracked separately from exact reconstruction errors. Across the 15 model–format aggregates shown in the main paper, malformed-output rates range from approximately 0.058 to 0.118, truncation rates from 0.054 to 0.113, and refusal rates remain below 0.008. These are aggregate diagnostics rather than substitutes for the hand-level exact-success metric. The parsing rules and mutually exclusive primary failure labels used to produce those rates are specified below.

E Prompt Templates and Parsing

This appendix gives the prompt templates used in the two-call retention experiments. The templates are parameterized by the hand, the memory format, the output budget, and, in the external-access condition, the query budget. The prompts are intentionally neutral: they do not mention the theorem, the phase law, or the expected boundary. Their purpose is to enforce the experimental abstraction $H \to M \to \hat{H}$. The imperative style in the boxes below is part of the model-facing experimental material, not the prose style of the paper.

E.1 Two-Call Prompt Templates

Each trial consists of two independent calls. The first call sees the full constraint hand and writes a bounded memory representation. The second call receives only that representation and reconstructs the full hand. The original constraints are never included in the second call.

Call 1: Write a Bounded Memory Representation

```
You are participating in a memory-reconstruction task.
You will be given a list of key-value constraints.  Each key is a
synthetic identifier, and each value is the value associated with that
key.  Your goal in this call is not to answer the task directly, but
to write a memory representation that will allow a later call to
reconstruct the full set of constraints.
Important instructions:  1.  Preserve as much information from the
constraints as possible.  2.  Use the requested memory format exactly.
3.  Stay within the specified memory budget.  4.  Do not output the
final reconstructed JSON in this call.  5.  Do not include discussion
of the experiment, uncertainty, or strategy unless the requested
memory format explicitly allows brief notes.  6.  Your output will be
used as the only input to a later reconstruction call.
Memory format:  {FORMAT}
Memory budget:  {BUDGET}
Constraint list:  {CONSTRAINT_LIST}
Write only the memory representation.
```

Call 2: Reconstruct from Memory Only

```
You are participating in the second step of a memory-reconstruction
task.
You will be given a memory representation written by a previous call.
The original constraint list is no longer available.  Your task is to
reconstruct the complete set of key-value constraints using only the
memory representation below.
```

```
Important instructions: 1. Return a strict JSON object. 2. Include
every required key exactly once. 3. Do not include keys that were
not part of the original constraint list. 4. Do not include
explanations, markdown, comments, or text before or after the JSON
object. 5. If the memory representation is compact, decode it
according to the rule given in the representation. 6. If a value is
uncertain, still output your best reconstruction in the required JSON
format.
Memory representation: {MEMORY_REPRESENTATION}
Required output format: {"key_0001": value, "key_0002": value,
"key_0003": value, ...}
Return only the JSON object.
```

E.2 Format and Access Variants

The two-call structure is fixed across all conditions. Only the memory-format instruction in Call 1 changes. The reconstruction prompt in Call 2 remains the same except in the external-access condition.

Free-Form CoT / Notes Format

```
Memory format: Free-form notes.
Write concise notes that would help a later call reconstruct the full
constraint list. You may use natural language, grouping, reminders,
or short intermediate reasoning. However, the goal is memory
preservation, not explanation.
Guidelines: 1. Prefer compact notes over verbose reasoning. 2.
Preserve exact key-value associations whenever possible. 3. If
grouping keys helps, define the grouping clearly. 4. If you infer a
rule or pattern, state it briefly and precisely. 5. Do not write the
final JSON answer. 6. Do not include irrelevant commentary.
The later call will receive only these notes, so the notes must
contain enough information to reconstruct the constraints.
```

Structured Key-Value Format

```
Memory format: Structured key-value memory.
Write the memory as a regular table or list of key-value entries. Use
a consistent structure throughout the output.
Preferred format: key | value
Guidelines: 1. Keep one retained key-value association per entry
unless a compact group is explicitly defined. 2. If multiple keys
share the same value, you may group them, but the grouped keys must be
recoverable. 3. Avoid natural-language explanations except for short
rules needed to decode the memory. 4. Do not output the final JSON
answer. 5. Do not include keys or values that are not in the
constraint list.
```

Compressed Symbolic Format

```
Memory format: Compressed symbolic memory.
Write the most compact representation you can while keeping the hand
decodable.
Guidelines: 1. Use the original key order unless you explicitly
define a different order. 2. For binary values, prefer a bitstring
in key order. 3. For multi-symbol values, define a short symbol map
and then write the encoded sequence. 4. If using hexadecimal, base64,
run-length encoding, or another compact code, state the decoding rule
briefly. 5. The decoding rule must be short but sufficient for
reconstructing the full hand. 6. Do not include explanatory prose
beyond the decoding rule. 7. Do not output the final JSON answer.
Example structure: ORDER: original MAP: A=0, B=1, C=2 CODE: ABBCAC...
```

External-Access Reconstruction Variant

```
You are participating in the second step of a memory-reconstruction
task.
You will be given a memory representation written by a previous call.
The original constraint list is no longer available.  In this
condition, you may request a limited amount of additional information
before producing the final answer.
Query budget:  {Q} bits
Rules for queries:  1.  Each query must specify one key.  2.  A query
returns the value associated with that key.  3.  Do not exceed the
query budget.  4.  Use queries only if they help reconstruct the final
hand.  5.  After querying, return the final answer as a strict JSON
object.  6.  Do not include explanations, markdown, or text before or
after the final JSON.
Memory representation:  {MEMORY_REPRESENTATION}
Query format:  QUERY key_id
Final output format:  {"key_0001":  value, "key_0002":  value,
"key_0003":  value, ...}
After using any queries, return only the JSON object.
```

In the external-access condition, Q counts the amount of independent value information returned by the external channel. The no-access condition is $Q = 0$. The final reconstruction in this variant may use both the written memory representation and the answer-time query responses.

E.3 Parsing, Normalization, and Failure Labels

The main metric uses strict parsing. A trial is an exact success only if the parsed output contains every required key exactly once, contains no extra keys, and assigns every key the correct value. Ordering of JSON keys is ignored.

A relaxed parser is used only for diagnostics. It attempts to recover an unambiguous key-value mapping from non-JSON text, line-based tables, or JSON-like outputs. Relaxed parsing does not guess missing values, resolve ambiguous duplicate keys, or infer values from explanations.

Table 8: Parsing and failure-labeling rules.

Label	Definition
Exact success	The output parses as a mapping with every required key exactly once, no extra keys, and all values matching the ground truth.
Wrong value	The required keys are present, but at least one value is incorrect.
Omission	One or more required keys are missing.
Extra key	The output contains a key that is not part of the target hand. Under strict parsing, this is a failure.
Malformed	No valid key-value mapping can be parsed, or the JSON object is syntactically invalid.
Truncation	The output stops before completing the mapping, or generation reaches the configured maximum output length.
Refusal	The model refuses, says reconstruction is impossible, or returns a meta-answer instead of attempting the required output.
Budget violation	The first-call memory representation exceeds the specified budget. The trial is retained for diagnostics and marked as a protocol violation.

If multiple failure labels apply, we assign one primary label using the priority order

$$\text{refusal} > \text{truncation} > \text{malformed} > \text{budget violation} > \text{omission} > \text{extra key} > \text{wrong value}.$$

For binary tasks, strict parsing accepts integer values 0 and 1. String values such as `"0"` and `"1"` are accepted only by the relaxed parser. For multi-symbol tasks, the allowed value set is fixed by the generator, and values must match the canonical symbols exactly under strict parsing. Case folding

or whitespace normalization is applied only when the generator defines string labels for which such normalization is unambiguous.

For token-budgeted memory writing, we count tokens using the tokenizer associated with the evaluated model when available; otherwise we use the serving backend's reported token count. For character-budgeted diagnostic runs, we count Unicode codepoints after trimming leading and trailing whitespace. The reconstruction call is given a separate maximum generation length large enough to hold the full JSON answer for the largest hand in the grid; outputs that hit this limit before completing the object are labeled as truncation.