Attention Deficits in Language Models Causal Explanations For Procedural Hallucinations

Ahmed Karim[1], Jad Awada[1], Sara Hammoud[1], Leon Chlon[1,2]
[1]Hassana Labs
[2]Oxford University, Torr Vision Group

Abstract

Large language models can follow complex procedures yet fail at a seemingly trivial final step: reporting a value they themselves computed moments earlier. We call this failure *procedural hallucination*: the model possesses the information but does not use it at the critical decision point.

Our central finding is that these failures decompose into two distinct stages. In *Stage 2A*, the model fails to recognize that it should produce an answer at all. In *Stage 2B*, the model correctly enters "answer mode" but selects the wrong value, often due to recency bias or competition among candidate statistics. Across three model families, Stage 2B errors dominate: linear probes recover the correct answer from the model's own hidden states with 74% accuracy on trials where the model outputs the wrong answer (chance: 2%). The information is present but not routed to the output.

We formalize this routing failure using information theory. We define *available information* (what the hidden state encodes about the answer) versus *used information* (what the output exploits), and prove tight bounds relating these quantities to error rates. We introduce *pseudo-priors* via causal interventions to quantify how much evidence the model needs to overcome its biases, and extend this framework to audit reasoning traces for unsupported claims.

Empirically, oracle checkpointing (restating the answer near the query) converts 0% accuracy to 99%+, confirming that failures reflect distance-limited information routing rather than fundamental incapacity. We release a toolkit for computing these diagnostics from API logprobs.

1 Introduction

When a language model fabricates a historical date or invents a citation, we call this a *factual hallucination*: the model lacks the relevant knowledge. But a different failure mode is equally important and far less understood. Consider the **strawberry counting task** (see Appendix A): **"How many r's are in strawberry?" The model shows its work via chain-of-thought**, enumerating letters and maintaining a running count. For short traces, this works. But as the trace grows longer, the final answer drifts: at length $k = 20$, the model outputs 6; at $k = 30$, it outputs 8. The computation was faithful throughout. The failure occurred at the final step, where the readout mechanism selected a wrong statistic (a cumulative or recency-weighted count) instead of the intended one. The model did not lack the information; it simply failed to use it.

We call this a **procedural hallucination**: an inconsistency between information explicitly available in the prompt and the model's output. Unlike factual hallucination, which concerns world knowledge, procedural hallucination concerns the model's ability to faithfully execute a specification given in its own context. This distinction matters because procedural hallucinations cannot be fixed by scaling or retrieval; the information is already there.

This is not a knowledge failure. The model computed the correct value; probing its hidden state on error trials recovers the answer with high accuracy. The information is present; it is not routed to the output. This observation raises two questions. First, can we decompose these failures into interpretable stages? Second, can we formalize "information is present but not used" in a way that yields provable guarantees? This paper answers both. Experiment code is available at https://github.com/leochlon/procedural_chlon2026.

1.1 Overview of Contributions

We develop a theory of procedural hallucinations grounded in information theory and causal reasoning, then validate it empirically across model families.

Our first contribution is a **stagewise decomposition** of slot-population errors. We observe that failures occur in two distinct modes: the model may fail to "enter answer mode" at all (Stage 2A), or it may enter answer mode but select the wrong candidate (Stage 2B). We define logprob-derived margins that diagnose which stage failed on any given trial. This decomposition matters because the two stages have different causes and different remedies.

Our second contribution is an **information-theoretic framework** that formalizes "present but not used." We define available information I_{avail} as the mutual information between the correct answer and the model's hidden state, and used information I_{used} as the mutual information between the correct answer and the model's output. The ratio $\eta = I_{\text{used}}/I_{\text{avail}}$ is the routing efficiency. We prove tight Fano-style bounds connecting error rates to used information, including an exact slack decomposition that characterizes when the bounds bite.

Our third contribution is a **causal framework for pseudo-priors**. We define a null intervention $\text{do}(E = \varnothing)$ that removes the binding evidence, inducing a pseudo-prior $\tilde{p}$ that captures the model's baseline bias (e.g., toward recency). We prove a tight Bernoulli-projected bound: shifting from pseudo-prior $\tilde{p}$ to posterior p requires at least $\text{KL}(\text{Ber}(p)\|\text{Ber}(\tilde{p}))$ nats of information. This gives concrete "bits-to-trust" certificates.

Our fourth contribution is an **extension to reasoning traces**. We treat each step of a chain-of-thought as a claim with a step-specific pseudo-prior (computed by scrubbing the cited evidence). Steps whose observed information falls below the required budget are flagged as hallucination risks.

Empirically, we validate these ideas across Qwen, Llama, and Gemma models. We find that Stage 2B dominates in binding tasks, that probing certifies information presence on error trials, that activation patching localizes the failure to late MLP layers, and that oracle checkpointing recovers near-perfect accuracy. We release a toolkit that computes Stage 2A/2B margins and trace-level budgets from API logprobs.

2 Related Work

Hallucination taxonomies and faithfulness. Existing taxonomies classify hallucinations by the relationship between outputs and sources, and faithfulness analyses in summarization separate factuality from adherence to input [6, 7]. We position procedural hallucination as an orthogonal, mechanistic category: information is present in context but fails to route to the output.

Long-context failures and "know but don't tell." Lost-in-the-middle results show position-dependent retrieval failures even when the answer is explicitly present [8]. Probing studies demonstrate that models often encode the correct answer while failing to produce it [9], and retrieval-head ablations localize the routing pathway [10]. Our stagewise decomposition and margins formalize how these failures arise despite available information.

Mechanistic interpretability and circuit tracing. Causal tracing and editing [5], automated circuit discovery [11], and detailed circuit analyses of binding tasks [12] provide tools for localizing information flow, grounded in a circuit-level framework for transformers [13]. We adapt these methods to slot-population failures and quantify where routing degrades.

Information-theoretic analyses. The information bottleneck formalizes how representations preserve task-relevant information [14], and neural estimators make mutual information measurement tractable in practice [15]. Building on classical information-theoretic tools [1], we define available versus used information and derive tight error bounds that diagnose routing inefficiency.

Chain-of-thought faithfulness. Reasoning traces can be unfaithful to the underlying computation: interventions show post-hoc rationalization and systematic unfaithful explanations [16, 17]. Our trace-level audits operationalize this gap by attaching information budgets to individual steps.

Binding and attention-based routing. Induction heads and in-context optimization describe how transformers bind variables to values [18, 19], while attention sinks and attention-flow analyses reveal routing bottlenecks in multi-layer attention [20, 21]. Our binding tasks and pseudo-prior framework target these routing failures directly.

3 Stagewise Slot Population

We now formalize the setting in a way that is both precise and operationally compatible with API-accessible outputs. The key insight is that slot-population errors can fail in two distinct ways, and distinguishing them clarifies both diagnosis and mitigation.

3.1 Prompt Structure

A prompt W_k has three components: (i) a binding region where keys are assigned values, (ii) k filler tokens, and (iii) a query requesting a specific value. Values come from a candidate set $\mathcal{C} \subset \mathcal{V}$ of single-token strings. We study two task variants:

competing_vars	**primacy_recency**

`KEY1 = [apple]`
[k filler tokens]
`KEY2 = [banana]`
[k filler tokens]
`What is KEY1?  KEY1 = [`
	`KEY = [alpha]`
	[k filler tokens]
	`KEY = [beta]`
	[k filler tokens]
	`KEY = [gamma]`
	[k filler tokens]
	`What was the FIRST value of KEY? KEY = [`

Different keys hold different values; the competitor is a distractor.	Same key reassigned; recency bias actively competes with the correct answer.

3.2 Stage 2A: Does the Model Enter Answer Mode?

Let $z_\theta(x \mid W_k)$ denote the next-token logit for token x at the readout position. We define:

$$Y_k := \arg\max_{v \in \mathcal{C}} z_\theta(v \mid W_k) \qquad \text{(best candidate)} \qquad (1)$$

$$\hat{Y}_k := \arg\max_{x \in \mathcal{V}} z_\theta(x \mid W_k) \qquad \text{(best overall token)} \qquad (2)$$

The indicator $G_k := \mathbf{1}\{\hat{Y}_k \in \mathcal{C}\}$ marks whether the model "enters answer mode" by outputting a candidate token. We quantify this with the **gate margin**:

$$\text{GateGap}(W_k) \;:=\; \max_{v \in \mathcal{C}} z_\theta(v \mid W_k) \;-\; \max_{x \notin \mathcal{C}} z_\theta(x \mid W_k).$$

When GateGap > 0, the model will output a candidate; when GateGap < 0, it will output something else entirely (e.g., punctuation or a hedging phrase). Negative gate margin indicates a Stage 2A failure.

3.3 Stage 2B: Does the Model Select the Right Candidate?

Conditional on entering answer mode, we ask whether the model selects correctly. Let $V \in \mathcal{C}$ denote the ground-truth value. The **value margin** is:

$$\text{ValueGap}(W_k) \;:=\; z_\theta(V \mid W_k) \;-\; \max_{v \in \mathcal{C} \setminus \{V\}} z_\theta(v \mid W_k).$$

Positive value margin means correct binding; negative value margin means the model prefers a wrong candidate. This is a Stage 2B failure.

3.4 Why This Decomposition Matters

The distinction between Stage 2A and Stage 2B is not merely taxonomic. The two failure modes have different signatures and different remedies:

- **Stage 2A failures** indicate that the model does not recognize the query as requesting a value from $\mathcal{C}$. This is a format or instruction-following problem.
- **Stage 2B failures** indicate that the model understands the task format but routes to the wrong value. This is an information-routing problem, often driven by recency bias.

We report accuracy $\mathbb{P}(\hat{Y}_k = V)$, candidate accuracy $\mathbb{P}(Y_k = V \mid G_k = 1)$, and stage fractions Frac-$2A = \mathbb{P}(G_k = 0)$ and Frac-$2B = \mathbb{P}(G_k = 1,\, Y_k \neq V)$.

4 Information-Theoretic Framework

We now formalize "information is present but not used." The key objects are available information (what the hidden state knows) and used information (what the output exploits). Proofs appear in B.

4.1 Available versus Used Information

Let H_k denote the model's internal state at the readout position (e.g., the final-layer residual stream). The data-generating process forms a Markov chain:

$$V \;\to\; W_k \;\to\; H_k \;\to\; Y_k,$$

where V is the ground-truth value, W_k is the prompt, H_k is the hidden state, and Y_k is the model's candidate-set decision.

Definition 1 (Available and used information). We define:

$$I_{\text{avail}}(k) := \mathrm{I}(V; H_k), \qquad I_{\text{used}}(k) := \mathrm{I}(V; Y_k),$$

and the **routing efficiency**:

$$\eta_k := \frac{I_{\text{used}}(k)}{I_{\text{avail}}(k)} \in [0, 1].$$

Proposition 1 (Data processing). *For any k, we have $0 \leq \eta_k \leq 1$.*

Routing efficiency captures how much of the available information the model actually exploits. Procedural hallucinations correspond to $\eta_k \ll 1$: the hidden state encodes the answer (I_{avail} is large), but the output ignores it (I_{used} is small).

4.2 From Error Rates to Information: Fano Bounds

Let $M := |\mathcal{C}|$ and define the within-candidate error rate $\varepsilon_B(k) := \mathbb{P}(Y_k \neq V \mid G_k = 1)$.

Theorem 1 (Fano lower bound). *If V is uniform on $\mathcal{C}$, then:*

$$I_{\text{used}}(k) \;\geq\; \log M \;-\; h(\varepsilon_B(k)) \;-\; \varepsilon_B(k) \log(M-1),$$

where $h(\cdot)$ is binary entropy in nats.

This bound says that high error implies low used information. But is the bound tight? The following proposition says yes, and characterizes exactly when.

Proposition 2 (Minimax tightness). *For every M and $\varepsilon \in [0, 1 - 1/M]$, the M-ary symmetric channel achieves equality in 1. This channel is minimax optimal: it minimizes $\mathrm{I}(V;Y)$ among all channels with error rate ε.*

When does the bound have slack? The following decomposition makes this precise.

Proposition 3 (Fano slack decomposition). *Under the conditions of 1, define $E = \mathbf{1}\{Y \neq V\}$. Then:*

$$\mathrm{I}(V;Y) = \underbrace{\log M - h(\varepsilon) - \varepsilon \log(M-1)}_{\textit{Fano lower bound}} + \underbrace{\big(h(\varepsilon) - \mathrm{H}(E \mid Y)\big)}_{\textit{Jensen slack } \geq 0} + \underbrace{\varepsilon \cdot \big(\log(M-1) - \mathrm{H}(V \mid Y, E=1)\big)}_{\textit{non-uniform confusion slack } \geq 0}. \tag{3}$$

The first slack term vanishes iff the error rate is constant across output values. The second vanishes iff, given an error, all wrong values are equally likely.

Corollary 1 (Fano inversion). *Define $\Phi_M(\varepsilon) := \log M - h(\varepsilon) - \varepsilon \log(M-1)$. Then $\mathrm{I}(V;Y) \geq \Phi_M(\varepsilon)$ implies $\varepsilon \geq \Phi_M^{-1}(\mathrm{I}(V;Y))$. Under the M-ary symmetric channel, this is an equality.*

4.3 Pseudo-Priors and Decompression Bounds

The Fano bounds connect error to used information. But how much information does the model *need*? This depends on its baseline bias. A model with strong recency bias needs more evidence to overcome that bias than an unbiased model would.

We formalize this via a causal intervention that removes the binding evidence.

Definition 2 (Pseudo-prior)**.** Let E denote the binding evidence (e.g., "KEY1 = [apple]"). Define a null distribution by the intervention:

$$\tilde{W}_k \sim \mathrm{do}(E = \varnothing),$$

which removes E while preserving the template, candidate set, and competing cues (e.g., recency). Let $\tilde{Y}_k$ be the model's decision under $\tilde{W}_k$. The **pseudo-prior** is:

$$\tilde{p}_k := \mathbb{P}(\tilde{Y}_k = V).$$

4.3.1 Operationalizing $\mathrm{do}(E = \varnothing)$: structure-preserving evidence ablation

The intervention $\mathrm{do}(E = \varnothing)$ is a *design pattern*: remove the *semantic support* for a binding while preserving the *structure* of the instance so that differences in behavior can be attributed to the missing evidence rather than to superficial prompt changes.

Design goal (invariances of the null). Our null operator is chosen to preserve: (i) the prompt template (role labels, span IDs, delimiters, formatting), (ii) the query and candidate set, and (iii)—as far as practical—length and locality statistics (so that "distance" and recency cues remain comparable). This answers the "empty string vs. noise vs. alternative binding" ambiguity: we target *structure-preserving evidence ablation*, not an arbitrary prompt corruption.

Operator used in experiments. In binding experiments, we implement $\mathrm{do}(E = \varnothing)$ by removing the key–value *content* while keeping the key line and delimiters (e.g., replacing `KEY1 = [apple]` with `KEY1 = [REDACTED]`). In trace auditing (6), we implement $\mathrm{do}(E = \varnothing)$ as *span scrubbing*: for a step citing spans S_i, we replace the *contents* of those spans with a fixed placeholder (default `[REDACTED]`) while preserving span labels and delimiters, and re-run the verifier on the scrubbed prompt to obtain $p_{0,i}$.

What the pseudo-prior means (and why `[REDACTED]` is not "cheating"). We emphasize that `[REDACTED]` is not distribution-neutral. It is an explicit marker for "evidence removed." Accordingly, $\tilde{p}$ should be interpreted as *the model/verifier's probability when explicitly denied access to that evidence*, which is exactly the counterfactual required for budgeting. Empirically, this tends to be conservative: scrubbing usually drives p_0 downward (more `UNSURE`/abstain behavior), which increases the required bits-to-trust and therefore *flags more*, not fewer, instances.

Robustness via a null family (envelope certification). To avoid over-committing to a single null implementation, we can define a small family of structure-preserving null operators $\mathcal{N}$ (e.g., delete-span, redact-span, same-length masking), and compute a pseudo-prior interval

$$p_0^{\min} := \min_{\nu \in \mathcal{N}} p_0^{(\nu)}, \qquad p_0^{\max} := \max_{\nu \in \mathcal{N}} p_0^{(\nu)}.$$

For budget tests of the form $\mathrm{KL}(\mathrm{Ber}(p_1)\|\mathrm{Ber}(p_0)) \geq \mathrm{KL}(\mathrm{Ber}(\tau)\|\mathrm{Ber}(p_0))$, we then certify against the *hardest* null (worst-case over $\nu \in \mathcal{N}$), i.e. require the inequality to hold for all $\nu \in \mathcal{N}$. This "envelope" move is directly analogous to the assumption-free, output-computable robustness layer used in compression-based ISR planners: we do not claim a single null is uniquely correct, we certify against a small, explicit family.

The pseudo-prior measures how likely the model is to guess correctly without the critical evidence. If recency bias is strong and the correct answer appeared first, $\tilde{p}_k$ will be small.

Theorem 2 (Bernoulli-projected decompression bound)**.** *Let P and Q be distributions with $P(A) = p$ and $Q(A) = \tilde{p}$ for some event A. Then:*

$$\mathrm{KL}(P\|Q) \;\geq\; \mathrm{KL}(\mathrm{Ber}(p) \,\|\, \mathrm{Ber}(\tilde{p})).$$

Moreover, for any $(p, \tilde{p})$ there exists a pair (P, Q) achieving equality.

Corollary 2 (Bits-to-trust)**.** *To achieve success probability p from pseudo-prior $\tilde{p}$, the model needs at least* $\mathrm{KL}(\mathrm{Ber}(p)\|\mathrm{Ber}(\tilde{p}))$ *nats.*

Concrete example. Suppose recency bias gives the correct answer a pseudo-prior of $\tilde{p} = 0.05$. To achieve 90% accuracy, the model needs at least $\mathrm{KL}(\mathrm{Ber}(0.9)\|\mathrm{Ber}(0.05)) \approx 2.25$ nats ≈ 3.2 bits of evidence. This is the "bits-to-trust" cost of overcoming the bias.

4.4 Certifying "Present but Not Used"

Estimating $I_{\text{avail}} = \mathrm{I}(V; H_k)$ requires access to hidden states. But we can certify routing failure without a tight estimate.

Proposition 4 (Certification via probing)**.** *For any probe f:*

$$\mathrm{I}(V; H_k) - \mathrm{I}(V; Y_k) \;\geq\; \mathrm{I}(V; f(H_k)) - \mathrm{I}(V; Y_k).$$

If a linear probe achieves higher mutual information with V than the model's own output does, then the model is provably failing to use available information. We do not need to estimate I_{avail} exactly; we only need a probe that outperforms the model.

4.5 Distance Dependence

Why do binding failures worsen with distance? We formalize this via strong data processing inequalities (SDPI).

Definition 3 (SDPI coefficient)**.** For a channel K mapping distributions on $\mathcal{X}$ to distributions on $\mathcal{X}'$:

$$\alpha(K) := \sup_{P \neq Q} \frac{\mathrm{KL}(PK \,\|\, QK)}{\mathrm{KL}(P \,\|\, Q)} \;\in\; [0, 1].$$

Theorem 3 (SDPI contraction)**.** *For a Markov chain $U \to X \to X'$ with channel K:*

$$\mathrm{I}(U; X') \;\leq\; \alpha(K)\, \mathrm{I}(U; X).$$

For a chain $V \to S_0 \to S_1 \to \cdots \to S_k \to H_k$:

$$\mathrm{I}(V; H_k) \;\leq\; \Big(\prod_{t=1}^{k} \alpha(K_t)\Big)\, \mathrm{I}(V; S_0).$$

Information decays geometrically with distance. This is not merely an upper bound; for "copy-or-noise" channels, the decay is exact (5 in Appendix).

Model	Task	k	Acc	Cand-Acc	Frac-2A	Frac-2B	GateGap	ValueGap
Qwen2.5-3B	competing_vars	256	0.107	0.158	0.350	0.650	0.56	-1.81
Qwen2.5-3B	primacy_recency	256	0.043	0.045	0.185	0.815	0.96	-3.40
Qwen2.5-3B-Instruct	competing_vars	256	0.276	0.360	0.402	0.598	0.61	-0.77
Qwen2.5-3B-Instruct	primacy_recency	256	0.061	0.059	0.072	0.928	2.10	-3.96
gemma-2-2b	competing_vars	256	0.995	0.995	0.000	1.000	3.03	2.28
gemma-2-2b	primacy_recency	256	0.801	0.804	0.013	0.987	1.65	0.47
Llama-3.2-3B	competing_vars	2048	0.385	0.369	0.018	0.982	2.01	-0.28
Llama-3.2-3B	primacy_recency	2048	0.001	0.014	0.461	0.539	-0.08	-5.14

Table 1: Stage decomposition ($n = 800$). For Llama we report $k = 2048$ to reach comparable difficulty. Stage 2B dominates in COMPETING_VARS; PRIMACY_RECENCY is harder and shows mixed patterns at extreme distance.

Implication for checkpointing. Checkpointing injects fresh copies of the evidence, resetting the distance. This increases $I_{\text{avail}}(k)$ and thus the attainable accuracy.

5 Empirical Results

Our experiments test three predictions of the framework:

1. Stage 2B errors (wrong candidate) should dominate over Stage 2A errors (no candidate) in binding tasks.
2. On Stage 2B error trials, probes should recover the correct answer at above-chance rates, certifying "present but not used."
3. Checkpointing should recover accuracy by shortening the effective evidence distance.

We also localize the failure mechanistically via activation patching. Full protocols appear in C.

Models and tasks. We evaluate Qwen2.5-3B (and its instruction-tuned variant), Llama-3.2-3B-Instruct, and Gemma-2-2b-it on COMPETING_VARS and PRIMACY_RECENCY tasks with distances $k \in \{256, 512, 1024, 2048\}$ tokens. Sample sizes are $n = 800$ for stage decomposition and probing, $n = 400$ for baseline comparisons and checkpointing, and $n \approx 160$ for patching (which requires paired clean/corrupt runs).

5.1 Prediction 1: Stage 2B Dominates

Table 1 shows the stage decomposition across models and tasks. In COMPETING_VARS, Stage 2B accounts for 65–100% of errors depending on model and distance. The model enters answer mode but selects the wrong candidate.

Table 2 shows a robustness hierarchy: Gemma > Llama ≫ Qwen. Gemma maintains near-perfect accuracy through $k = 1024$; Qwen collapses by $k = 256$.

5.2 Prediction 2: Probes Certify Information Presence

On trials where the model outputs the wrong candidate (Stage 2B errors), we train a linear probe on the final-layer residual stream to predict the correct answer. Table 3 shows the results.

Task	k	Qwen2.5-3B	Llama-3.2-3B-Inst.	Gemma-2-2b-it
competing_vars	256	0.203	**0.993**	**1.000**
	512	0.008	**0.993**	**1.000**
	1024	0.000	0.455	**0.998**
	2048	—	0.000	—
primacy_recency	256	0.060	**0.868**	**0.963**
	512	0.003	0.553	**0.798**
	1024	0.000	0.003	0.440
	2048	—	0.000	—

Table 2: Baseline accuracy ($n = 400$). All models eventually fail on PRIMACY_RECENCY at sufficient distance.

Task	k	Acc	Cand-Acc	Frac-2B	Probe@0	Probe@18	Probe@35
competing_vars	256	0.200	0.255	0.641	0.019	0.083	0.739
competing_vars	512	0.000	0.000	0.968	0.008	0.029	0.390
primacy_recency	256	0.069	0.074	0.754	0.020	0.040	0.354
primacy_recency	512	0.000	0.000	0.994	0.020	0.020	0.124

Table 3: Probing on Stage 2B errors (Qwen2.5-3B, $n = 800$). Layers 0/18/35 are embedding, mid, and final residual streams. Chance is ≈2% (candidate set size ≈50). At $k = 256$, the final-layer probe recovers the correct answer on 74% of error trials.

At $k = 256$ in COMPETING_VARS, the final-layer probe achieves 0.739 accuracy, which is 37× above chance, even though the model output the wrong answer. By 4, this certifies that $\mathrm{I}(V; H_k) > \mathrm{I}(V; Y_k)$: the information was present but not used.

5.3 Mechanistic Localization: Activation Patching

We localize the routing failure using activation patching. For each layer and component (attention vs. MLP), we patch activations from a "clean" run (where the model would be correct) into a "corrupt" run (where it errs) and measure how much the correct answer's logit margin recovers.

Table 4 reveals a consistent motif across all three model families: late attention layers restore correct bindings, while late MLPs corrupt them. Gemma's attention is approximately twice as strong as Qwen's, which may explain its greater robustness.

We also identify specific attention heads. Notably, Gemma has an "anti-recency" head (L25H6, ablation delta −0.51) that actively fights recency bias, counterbalancing its misbinding head (L25H2, +0.51). Qwen and Llama lack this compensatory mechanism.

5.4 Prediction 3: Checkpointing Recovers Accuracy

To test whether failures reflect distance rather than incapacity, we introduce oracle checkpointing: restating bindings every 128 tokens. Table 5 shows dramatic recovery.

Qwen recovers from 0% to 99.8% accuracy at $k = 1024$. The one exception is Llama at $k = 2048$ on PRIMACY_RECENCY, where checkpointing fails (0% → 0.3%). Inspection reveals that Stage 2A gating has collapsed: the model no longer enters answer mode. This confirms the stagewise picture: Stage 2B failures are distance-limited and recoverable; Stage 2A failures at extreme distance are a distinct breakdown.

Qwen2.5-3B-Instruct				**Llama-3.2-3B-Instruct**				**Gemma-2-2b-it**			
L	C	Rest	ΔM	L	C	Rest	ΔM	L	C	Rest	ΔM
32	attn	0.11	2.09	21	attn	**0.23**	0.06	22	attn	**0.31**	0.40
32	mlp	0.05	0.81	21	mlp	-0.04	0.00	22	mlp	-0.01	0.02
33	attn	**0.18**	3.44	25	attn	0.10	0.23	23	mlp	-0.16	0.07
33	mlp	-0.04	-0.81	25	mlp	**-0.10**	-0.19	25	attn	**0.36**	0.98
34	mlp	-0.11	-1.99	26	attn	**0.21**	0.24	25	mlp	**-0.19**	-0.47
35	mlp	**-0.20**	-3.30	27	attn	0.19	0.20				

Table 4: Layer-component patching ($n \approx 160$ pairs). *Rest*: fraction of margin restored. ΔM: logit margin change. All three models share a motif: late attention restores binding (positive), late MLPs corrupt it (negative).

Option-randomized naturalistic notes binding at 8B–9B scale. To address fixed-label confounds (e.g., a constant "correct letter") and test generality beyond 2–3B models, we run an option-randomized version of `notes_binding` in which the mapping from values to choice letters (A–D) is independently permuted per instance (so $H(V) > 0$ and $I(V;\hat{V})$ is well-defined). We evaluate Meta-Llama-3.1-8B-Instruct and Gemma-2-9b-it at distances $k \in \{0, 64, 256, 1024, 2048\}$ with and without checkpointing ($n = 200$ per cell). Table 6 summarizes results. Llama exhibits distance sensitivity (baseline accuracy 0.575 at $k = 64$ falling to 0.430 at $k = 2048$), while checkpointing partially recovers performance (to 0.600 at $k = 2048$). Gemma is nearly perfect under checkpointing (≥ 0.995 across all k) and, intriguingly, its baseline accuracy increases with distance (from 0.330 at $k = 0$ to 0.820 at $k = 2048$). This increase is driven primarily by a collapse in abstention ($P[\hat{Y} = Z]$) from 0.49 at $k = 0$ to 0.00 at $k = 2048$; conditional on answering (A–D), Gemma's accuracy also rises from 0.65 to 0.82. As an information-theoretic summary, Gemma's checkpointed runs achieve $I(V;\hat{V}) \approx 1.38$ nats (near $\log 4$), while Llama remains around 0.7–0.85 nats, consistent with persistent routing inefficiency.

6 Audited Reasoning Traces

We now extend the framework to reasoning traces. The model produces a chain-of-thought, and we audit whether each step is supported by the evidence it cites.

6.1 Trace Format

We prompt the model to produce a structured trace $T = \{(c_i, S_i, o_i)\}_{i=1}^{T}$ where c_i is an atomic claim, S_i is a set of cited span identifiers, and o_i is an optional confidence. A separate verifier labels each step as `ENTAILED`, `CONTRADICTED`, `NOT_IN_CONTEXT`, or `UNVERIFIABLE`. An executor then derives the answer using only verified steps.

6.2 Trace-Level Pseudo-Priors

For each step i, we define a pseudo-prior by scrubbing its cited spans:

$$\tilde{W}^{(i)} := \mathrm{do}(\text{spans in } S_i := \texttt{[REDACTED]}).$$

Let $p_{0,i}$ be the verifier's probability that step i is entailed under $\tilde{W}^{(i)}$, and $p_{1,i}$ be the probability under the full context. We define:

$$\mathrm{ReqBits}_i := \mathrm{KL}(\mathrm{Ber}(o_i) \,\|\, \mathrm{Ber}(p_{0,i})), \qquad \mathrm{ObsBits}_i := \mathrm{KL}(\mathrm{Ber}(p_{1,i}) \,\|\, \mathrm{Ber}(p_{0,i})).$$

Model	Task	k	Baseline	+Checkpoint	ΔAcc	ValueGap$_{\text{base}}$	ValueGap$_{\text{chk}}$
Qwen2.5-3B	competing_vars	256	0.203	**0.973**	+0.770	-1.02	+2.81
	competing_vars	512	0.008	**0.993**	+0.985	-5.75	+2.61
	competing_vars	1024	0.000	**0.998**	+0.998	-9.93	+2.88
	primacy_recency	256	0.060	**0.760**	+0.700	-2.88	+0.80
	primacy_recency	512	0.003	**0.573**	+0.570	-6.53	+0.26
	primacy_recency	1024	0.000	**0.763**	+0.763	-9.05	+1.01
Llama-3.2-3B-Inst.	competing_vars	256	0.993	0.998	+0.005	+8.89	+9.20
	competing_vars	1024	0.455	**0.998**	+0.543	+0.03	+8.96
	competing_vars	2048	0.000	**0.853**	+0.853	-3.78	+1.44
	primacy_recency	256	0.868	**0.995**	+0.127	+1.81	+3.48
	primacy_recency	512	0.553	**0.973**	+0.420	+0.16	+2.95
	primacy_recency	1024	0.003	**0.915**	+0.912	-4.30	+1.64
	primacy_recency	2048	0.000	0.003	+0.003	-4.02	-1.35
Gemma-2-2b-it	competing_vars	256	1.000	1.000	0.000	+8.82	+7.98
	competing_vars	512	1.000	1.000	0.000	+7.72	+8.10
	competing_vars	1024	0.998	1.000	+0.003	+6.55	+7.71
	primacy_recency	256	0.963	**0.993**	+0.030	+2.74	+3.80
	primacy_recency	512	0.798	**0.963**	+0.165	+1.37	+2.36
	primacy_recency	1024	0.440	**0.968**	+0.528	-0.62	+2.70

Table 5: Checkpointing results ($n = 400$). Qwen recovers from 0% to 99.8% at $k = 1024$. Exception: Llama at $k = 2048$ on PRIMACY_RECENCY shows checkpoint failure, as Stage 2A gating has collapsed and cannot be recovered by re-statement.

If $\text{ObsBits}_i < \text{ReqBits}_i$, the step is **under-budget**: the cited evidence does not justify the claimed confidence.

6.3 Empirical Validation

On synthetic binding tasks (Mistral-7B-Instruct, $k \in \{1024, 2048\}$, $n = 300$), the audit isolates errors effectively. At $k = 2048$, the pass subset achieves 99.5% accuracy while the flagged subset drops to 63.7% (Table 7).

On QuALITY reading comprehension, we use an evidence-only verifier (since correct answers often require inference rather than literal entailment). Table 8 shows that passing the audit is strongly predictive of correctness: 81–83% accuracy for pass vs. 59–62% for flagged.

7 Toolkit and Reproducibility

We release an open-source toolkit that implements the diagnostics described in this paper for any API model exposing token logprobs. Experiment code is available at https://github.com/leochlon/procedural_chlon2026.

The toolkit provides four capabilities. First, **Stage 2A/2B scoring**: given a prompt and candidate set, compute gate margin, value margin, and stage classification from the model's next-token logprobs. Second, **structured trace generation**: prompt the model to produce a JSON-formatted trace with span citations, using schema-constrained decoding. Third, **trace verification**: run a separate verifier model on each claim-span pair and compute p_0 (scrubbed) and

	Meta-Llama-3.1-8B-Instruct			**Gemma-2-9b-it**		
k	Acc (base)	$P[Z]$ (base)	Acc (+Chk)	Acc (base)	$P[Z]$ (base)	Acc (+Chk)
0	0.435	0.185	0.510	0.330	0.490	1.000
64	0.575	0.130	0.590	0.415	0.300	1.000
256	0.550	0.155	0.550	0.640	0.150	0.995
1024	0.435	0.105	0.580	0.740	0.015	1.000
2048	0.430	0.055	0.600	0.820	0.000	1.000

Table 6: Naturalistic `notes_binding` at 8B–9B scale with randomized choice ordering ($n = 200$ per cell). $P[Z]$ is the abstention rate (Stage 2A failure). Llama degrades at long distance and is partially recovered by checkpointing; Gemma is near-perfect under checkpointing and exhibits a large reduction in abstention as k increases.

Setting	k	Acc(%)	Pass(%)	Acc(pass)(%)	Acc(flag)(%)	Lift(pp)
Mistral-7B (baseline)	1024	98.3	54.8	100.0	97.0	3.0
Mistral-7B (baseline)	2048	87.0	68.6	99.5	63.7	35.8

Table 7: Trace-budget audit ($\tau = 0.75$). At large distance, errors concentrate in flagged traces.

p_1 (full context) from logprobs. Fourth, **budget computation**: calculate ReqBits and ObsBits for each trace step and flag under-budget claims.

We additionally include a "reviewer-closure" experiment suite implementing option-randomized long-context binding (including $k = 0$ controls and checkpointing) and output-only routing certificates.

The implementation currently supports OpenAI's API (including `gpt-4o-mini`) and can be extended to other providers. Activation-level analyses (probing, patching) require local access to open-weight models and are implemented separately using standard libraries.

8 Discussion

Procedural versus factual hallucination. Our results suggest that many structured-generation failures are not about missing knowledge but about *mis-commitment*. The model encodes the correct answer (probes recover it) but routes its output toward a biased competitor. This is qualitatively different from factual hallucination and requires different interventions.

Why the theory gives more than necessary conditions. Fano's inequality is a necessary condition: low information implies high error. But Propositions 2 and 3 show when it is *sufficient*, that is, when error rates are close to the information-theoretic limit, and provide a measurable slack decomposition. The Bernoulli decompression bound (2) is tight by construction, giving exact "bits-to-trust" costs.

Mitigation strategies. Our results point to two levers. First, *increase availability*: checkpointing, retrieval, and shorter contexts all increase I_{avail}. Second, *increase routing efficiency*: the patching results suggest that late MLPs are a bottleneck. Gemma's anti-recency head shows that architectural solutions exist; whether they can be trained or induced is an open question.

Variant	n	Pass(%)	Acc(pass)(%)	Acc(flag)(%)	Lift(pp)	$\mathbb{E}[p_1]$	$\mathbb{E}[p_0^{\mathrm{NE}}]$
baseline	369	21.1	83.3	61.5	21.8	0.398	0.036
rag	378	25.1	81.1	60.4	20.6	0.428	0.039
checkpoint	383	24.0	82.6	59.5	23.2	0.430	0.037

Table 8: QuALITY audit with null-family envelope certification ($\tau = 0.75$, Llama-3.1-8B-Instruct). Pass means the budget test holds for all null operators $\nu \in \mathcal{N}$ (redact, delete, same-length mask, and no-evidence). $\mathbb{E}[p_0^{\mathrm{NE}}]$ reports the no-evidence pseudo-prior.

9 Limitations

Our mechanistic claims (probing, patching) require activation access and thus apply directly only to open-weight models. For hosted APIs, we provide output-level and trace-level diagnostics but cannot estimate I_{avail} without activations.

Trace auditing detects certificate failures, that is, claims that lack sufficient evidential support, but does not prove faithfulness to the model's hidden chain-of-thought. A model could produce a valid-looking trace via post-hoc rationalization.

Our experiments focus on synthetic binding tasks designed to isolate the phenomena. Whether the same mechanisms explain failures in naturalistic long-context tasks (e.g., document QA) remains to be validated.

10 Conclusion

We have presented a rigorous framework for understanding procedural hallucinations: failures where the model possesses information but does not use it. The framework decomposes errors into gating (Stage 2A) and binding (Stage 2B) failures, formalizes "present but not used" via information-theoretic routing efficiency, provides tight bounds connecting error rates to information budgets, and extends to auditing reasoning traces.

Empirically, Stage 2B errors dominate in binding tasks. Probes certify that correct information is encoded on error trials. Activation patching localizes failures to late MLPs. Checkpointing recovers near-perfect accuracy by shortening the evidence path.

We release a toolkit for computing these diagnostics from API logprobs, enabling practitioners to diagnose and mitigate procedural hallucinations in deployed systems.

A Strawberry Counting Analysis

Prompt: "How many r's are in 'strawberry'?" The correct word-level statistic is $W = 3$. For long traces, the final readout misbinds to competing statistics: WORD (intended), TOTAL (total count in the full trace), or SUFFIX (a recent suffix window). Empirically, baseline misbinds for $k > 10$: at $k = 20$ the final output is 6 (TOTAL), and at $k = 30$ it is 8 (SUFFIX); larger k yields TOTAL. A binding intervention that forces the final to match the word-level statistic returns final $= 3$ for all tested k with perfect trace fidelity (character match and run-count consistency both 1.00).

B Proofs

B.1 Proof of 1

Proof. Since Y_k is a function of H_k, we have the Markov chain $V \to H_k \to Y_k$. By data processing, $\mathrm{I}(V; Y_k) \le \mathrm{I}(V; H_k)$. Nonnegativity of mutual information yields $0 \le \eta_k \le 1$. □

B.2 Proof of 1 and 2

Proof sketch. For uniform V on $\mathcal{C}$, $H(V) = \log M$. Fano's inequality gives $H(V \mid Y) \le h(\varepsilon) + \varepsilon \log(M-1)$. Therefore $\mathrm{I}(V; Y) = H(V) - H(V \mid Y) \ge \log M - h(\varepsilon) - \varepsilon \log(M-1)$.

Tightness: the M-ary symmetric channel has $P(Y = V) = 1-\varepsilon$ and $P(Y = y \neq V) = \varepsilon/(M-1)$. For this channel, $H(V \mid Y)$ attains the Fano upper bound. Minimax optimality follows because the symmetric channel maximizes $H(V \mid Y)$ given ε. □

B.3 Proof of 3

Proof. Let $E = \mathbf{1}\{Y \neq V\}$. Since E is deterministic given (V, Y), we have $\mathrm{H}(E \mid V, Y) = 0$. By chain rule:

$$\mathrm{H}(V \mid Y) = \mathrm{H}(E, V \mid Y) - \mathrm{H}(E \mid V, Y) = \mathrm{H}(E \mid Y) + \mathrm{H}(V \mid Y, E).$$

Since $\mathrm{H}(V \mid Y, E = 0) = 0$ (when correct, $V = Y$), we have $\mathrm{H}(V \mid Y, E) = \varepsilon\, \mathrm{H}(V \mid Y, E = 1)$. Thus:

$$\mathrm{H}(V \mid Y) = \mathrm{H}(E \mid Y) + \varepsilon\, \mathrm{H}(V \mid Y, E = 1).$$

Using $\mathrm{I}(V; Y) = \log M - \mathrm{H}(V \mid Y)$ and adding/subtracting Fano bounds yields (3). Nonnegativity: $\mathrm{H}(E \mid Y) \le h(\varepsilon)$ by Jensen; $\mathrm{H}(V \mid Y, E = 1) \le \log(M-1)$ by support size. □

B.4 Proof of 3

Proof. Express mutual information as average KL:

$$\mathrm{I}(U; X) = \mathbb{E}_u\big[\mathrm{KL}(P(X \mid U = u) \,\|\, P(X))\big].$$

Since $X \mapsto X'$ is channel K: $P(X' \mid U = u) = P(X \mid U = u)K$ and $P(X') = P(X)K$. By definition of $\alpha(K)$:

$$\mathrm{KL}(P(X' \mid U = u) \,\|\, P(X')) \le \alpha(K)\, \mathrm{KL}(P(X \mid U = u) \,\|\, P(X)).$$

Averaging yields $\mathrm{I}(U; X') \le \alpha(K)\mathrm{I}(U; X)$. Chain bound follows by iteration. □

B.5 Proof of 2

Proof sketch. Let $f(\omega) = \mathbf{1}\{\omega \in A\}$. By data processing for KL:

$$\mathrm{KL}(P \| Q) \ge \mathrm{KL}(P \circ f^{-1} \,\|\, Q \circ f^{-1}) = \mathrm{KL}(\mathrm{Ber}(p) \| \mathrm{Ber}(\tilde{p})).$$

Tightness: choose P as the I-projection of Q onto $\{P : P(A) = p\}$. □

B.6 Proof of 4

Proof. By data processing, $\mathrm{I}(V; f(H_k)) \leq \mathrm{I}(V; H_k)$. Subtracting $\mathrm{I}(V; Y_k)$ yields the result. ☐

Proposition 5 (Exact contraction for copy-or-noise)**.** *For the copy-or-noise channel with parameter α (copies input with probability α, else outputs noise from ν):*

$$\mathrm{I}(U; X') = \alpha\, \mathrm{I}(U; X).$$

Proof. Let $B \sim \mathrm{Ber}(\alpha)$ be independent of (U, X) and $Z \sim \nu$ be independent noise. Since B is independent of U:

$$\mathrm{I}(U; X') = \mathrm{I}(U; X' \mid B) = \alpha\, \mathrm{I}(U; X' \mid B = 1) + (1 - \alpha)\, \mathrm{I}(U; X' \mid B = 0).$$

When $B = 0$, $X' = Z$ independent of U, so $\mathrm{I}(U; X' \mid B = 0) = 0$. When $B = 1$, $X' = X$, so $\mathrm{I}(U; X' \mid B = 1) = \mathrm{I}(U; X)$. ☐

C Additional Experimental Results

Qwen2.5-3B-Instruct				**Llama-3.2-3B-Instruct**				**Gemma-2-2b-it**			
L	C	Rest	ΔM	L	C	Rest	ΔM	L	C	Rest	ΔM
28	attn	-0.03	-0.50	20	attn	-0.04	0.04	18	attn	0.02	-0.41
28	mlp	0.03	0.59	20	mlp	0.04	0.03	18	mlp	0.01	0.17
29	attn	0.06	1.02	21	attn	0.23	0.06	19	attn	0.10	0.12
29	mlp	-0.01	-0.24	21	mlp	-0.04	0.00	19	mlp	-0.03	-0.05
30	attn	-0.04	-0.71	22	attn	-0.03	0.09	20	attn	-0.01	0.02
30	mlp	0.07	1.34	22	mlp	0.01	0.01	20	mlp	-0.01	0.01
31	attn	-0.01	-0.19	23	attn	-0.03	-0.15	21	attn	-0.09	-0.55
31	mlp	0.08	1.45	23	mlp	0.07	0.14	21	mlp	0.00	0.21
32	attn	0.11	2.09	24	attn	0.02	-0.04	22	attn	0.31	0.40
32	mlp	0.05	0.81	24	mlp	0.05	0.05	22	mlp	-0.01	0.02
33	attn	0.18	3.44	25	attn	0.10	0.23	23	attn	0.03	0.03
33	mlp	-0.04	-0.81	25	mlp	-0.10	-0.19	23	mlp	-0.16	0.07
34	attn	0.01	0.21	26	attn	0.21	0.24	24	attn	-0.01	0.04
34	mlp	-0.11	-1.99	26	mlp	-0.01	-0.22	24	mlp	0.02	0.10
35	attn	0.01	0.10	27	attn	0.19	0.20	25	attn	0.36	0.98
35	mlp	-0.20	-3.30	27	mlp	0.07	0.18	25	mlp	-0.19	-0.47

Table 9: Complete layer-component patching results.

C.1 QuALITY Robustness

Table 11 shows robustness of the *null-family envelope* evidence-only audit across reliability targets $\tau \in \{0.65, 0.70, 0.75, 0.80\}$.

Task	Filler	k	Acc	Cand-Acc	Frac-2A	Frac-2B	GateGap	ValueGap
competing_vars	decoy_heavy	128	0.999	0.999	0.000	1.000	3.74	5.12
competing_vars	decoy_heavy	256	0.107	0.158	0.350	0.650	0.56	-1.81
competing_vars	decoy_heavy	512	0.003	0.001	0.023	0.977	1.99	-7.26
competing_vars	decoy_heavy	1024	0.000	0.000	0.033	0.968	1.79	-10.10
competing_vars	repeat	128	0.030	0.080	0.570	0.430	-0.07	-3.39
competing_vars	repeat	256	0.026	0.069	0.792	0.208	-0.80	-4.33
competing_vars	repeat	512	0.000	0.094	0.988	0.013	-2.97	-3.46
competing_vars	repeat	1024	0.000	0.249	1.000	0.000	-7.76	-1.83
decoy_injection	decoy_heavy	128	0.979	0.994	0.824	0.176	3.23	7.14
decoy_injection	decoy_heavy	256	0.240	0.907	0.993	0.007	-1.17	2.41
decoy_injection	decoy_heavy	512	0.000	0.294	1.000	0.000	-6.92	-1.24
decoy_injection	decoy_heavy	1024	0.000	0.149	1.000	0.000	-8.09	-2.57
decoy_injection	repeat	128	0.065	0.996	1.000	0.000	-2.03	5.03
decoy_injection	repeat	256	0.000	0.779	1.000	0.000	-6.98	1.51
decoy_injection	repeat	512	0.000	0.292	1.000	0.000	-9.58	-1.09
decoy_injection	repeat	1024	0.000	0.295	1.000	0.000	-10.53	-1.33
primacy_recency	decoy_heavy	128	0.734	0.755	0.146	0.854	1.68	0.64
primacy_recency	decoy_heavy	256	0.043	0.045	0.185	0.815	0.96	-3.40
primacy_recency	decoy_heavy	512	0.000	0.000	0.005	0.995	2.55	-7.77
primacy_recency	decoy_heavy	1024	0.000	0.000	0.049	0.951	1.31	-8.14
primacy_recency	repeat	128	0.036	0.046	0.370	0.630	0.33	-3.69
primacy_recency	repeat	256	0.020	0.046	0.807	0.193	-0.62	-3.86
primacy_recency	repeat	512	0.000	0.142	1.000	0.000	-3.42	-2.47
primacy_recency	repeat	1024	0.000	0.139	1.000	0.000	-7.06	-2.48

Table 10: Complete Stage 2A/2B results for Qwen2.5-3B ($n = 800$ per row).

τ	Variant	Pass(%)	Acc(pass)(%)	Acc(flag)(%)	Lift(pp)
0.65	baseline	24.7	83.5	60.4	23.1
0.65	checkpoint	27.4	81.0	59.0	22.0
0.65	rag	28.6	79.6	60.0	19.6
0.70	baseline	22.5	84.3	60.8	23.5
0.70	checkpoint	25.3	82.5	59.1	23.4
0.70	rag	26.2	81.8	59.9	22.0
0.75	baseline	21.1	83.3	61.5	21.8
0.75	checkpoint	24.0	82.6	59.5	23.2
0.75	rag	25.1	81.1	60.4	20.6
0.80	baseline	15.7	87.9	62.1	25.9
0.80	checkpoint	20.1	83.1	60.5	22.7
0.80	rag	21.2	86.2	60.1	26.2

Table 11: Robustness of the QuALITY null-family envelope audit across reliability targets τ. Passing is consistently associated with substantially higher answer accuracy.

Null operator	Variant	Pass(%)	Acc(pass)(%)	Acc(flag)(%)	Lift(pp)
no-evidence	baseline	22.8	79.8	62.1	17.7
no-evidence	checkpoint	25.1	81.2	59.6	21.7
no-evidence	rag	26.5	79.0	60.8	18.2
redact-span	baseline	27.4	81.2	60.4	20.7
redact-span	checkpoint	30.8	78.8	58.9	19.9
redact-span	rag	31.2	80.5	58.8	21.7
delete-span	baseline	29.3	82.4	59.4	23.0
delete-span	checkpoint	31.6	77.7	59.2	18.5
delete-span	rag	33.1	80.0	58.5	21.5
mask-same-len	baseline	26.0	84.4	59.7	24.7
mask-same-len	checkpoint	31.1	80.7	58.0	22.7
mask-same-len	rag	30.4	80.9	58.9	21.9
envelope (all)	baseline	21.1	83.3	61.5	21.8
envelope (all)	checkpoint	24.0	82.6	59.5	23.2
envelope (all)	rag	25.1	81.1	60.4	20.6

Table 12: Sensitivity of QuALITY audit to the concrete pseudo-prior operator at $\tau = 0.75$. "Envelope" requires the budget test to pass for all null operators; it provides a conservative, assumption-light certificate while retaining strong predictive lift.

k	n	Pass(%)	$\mathbb{E}[p_1]$	$\mathbb{E}[p_0^{\min}, p_0^{\max}]$
512	200	66.0	0.756	[0.179, 0.272]
1024	200	0.0	0.728	[0.097, 0.184]
2048	200	100.0	0.841	[0.181, 0.361]

Table 13: Null-family statistics for a synthetic binding audit ($\tau = 0.75$). Although p_0 varies across null operators, the pass/fail decision is invariant across the null family in this setting (envelope equals any single-null decision).

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, 2nd edition, 2006.

[2] R. M. Fano. Transmission of information: a statistical theory of communications. MIT Press, 1961.

[3] L. Chlon, S. Rashidi, Z. Khamis, and M. M. Awada. LLMs are Bayesian, In Expectation, Not in Realization. arXiv:2507.11768, 2025.

[4] L. Chlon, A. Karim, and M. Chlon. Predictable Compression Failures: Why Language Models Actually Hallucinate. arXiv:2509.11208, 2025.

[5] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and Editing Factual Associations in GPT. *NeurIPS*, 2022.

[6] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, and others. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12), 2023.

[7] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. On Faithfulness and Factuality in Abstractive Summarization. *ACL*, 2020.

[8] N. F. Liu, K. Lin, J. Hewitt, and others. Lost in the Middle: How Language Models Use Long Contexts. *TACL*, 2024.

[9] T. Lu, M. Gao, K. Yu, A. Byerly, and D. Khashabi. Insights into LLM Long-Context Failures: When Transformers Know but Don't Tell. *Findings of EMNLP*, 2024.

[10] W. Wu, Y. Wang, G. Xiao, H. Peng, and Y. Fu. Retrieval Head Mechanistically Explains Long-Context Factuality. *ICLR*, 2025.

[11] A. Conmy, A. Mavor-Parker, A. Lynch, and others. Towards Automated Circuit Discovery for Mechanistic Interpretability. *NeurIPS*, 2023.

[12] K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small. *ICLR*, 2023.

[13] N. Elhage, N. Nanda, C. Olsson, and others. A Mathematical Framework for Transformer Circuits. Transformer Circuits Thread, 2021.

[14] N. Tishby, F. C. Pereira, and W. Bialek. The Information Bottleneck Method. *Allerton Conference*, 1999.

[15] M. I. Belghazi, A. Baratin, S. Rajeshwar, and others. Mutual Information Neural Estimation. *ICML*, 2018.

[16] T. Lanham, A. Chen, A. Radhakrishnan, and others. Measuring Faithfulness in Chain-of-Thought Reasoning. arXiv:2307.13702, 2023.

[17] M. Turpin, J. Michael, E. Perez, and S. R. Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. *NeurIPS*, 2023.

[18] C. Olsson, N. Elhage, N. Nanda, and others. In-context Learning and Induction Heads. Transformer Circuits Thread, 2022.

[19] J. von Oswald, E. Niklasson, E. Randazzo, and others. Transformers Learn In-Context by Gradient Descent. *ICML*, 2023.

[20] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis. Efficient Streaming Language Models with Attention Sinks. *ICLR*, 2024.

[21] S. Abnar and W. Zuidema. Quantifying Attention Flow in Transformers. *ACL*, 2020.