Context Management in Autoregressive Language Models: Reset, Prime, and the Limits of Prompt Engineering

Jeremy McEntire[1]

April 2026

Abstract

How much does the structure of a prompt matter—not the content, but the structure? We measure the effect of prompt configuration on continuation quality across nine experiments using Qwen 2.5-7B (28 layers, 3584-dimensional hidden state), evaluating 160 probes across four domains (code, science, medical, legal) using continuation perplexity as the metric, with cross-architecture replication of key results on Mistral-7B-v0.3.

The findings organize into three categories. **What helps:** A 15-token reset instruction ("disregard prior context") before domain priming reduces cross-entropy by 0.464 nats (39%). Fifteen tokens of domain-matched priming capture 98.8% of the perplexity reduction that 119 tokens provide. Natural conversational format outperforms rigid Q&A by 0.475 nats. **What hurts:** Meta-commentary about a domain ("you are an expert in medicine") activates instruction-following mode instead of domain-processing mode, increasing cross-entropy by 0.09–0.27 nats over neutral baselines. Within-context repetition degrades monotonically at +0.07 nats per repeat. Artificially capitalized vocabulary ("TROPONIN CASCADE") performs worst of all vocabulary conditions. **Where it saturates:** In sequential relay chains, degradation follows a logarithmic curve, rising from baseline by ∼0.10 nats over the first five hops and then plateauing through hop 10. Cross-domain handoffs outperform within-domain chains because the domain switch forces a natural mode reset.

The reset mechanism is not garbage collection. Three predictions of the garbage-collector hypothesis are derived and all three are falsified: reset benefit does not scale with context pollution, post-reset activations do not converge to a clean baseline, and the effect is not domain-independent. The central falsification—that reset benefit is independent of context length—replicates on Mistral-7B-v0.3 ($\rho = 0.10$, $p = 0.87$), confirming that the null result is not architecture-specific. The reset instruction installs a processing mode boundary—a context fence—that separates prior context from subsequent context. Mode switching and domain activation are orthogonal dimensions of the processing state ($\rho = 0.858$, $p < 10^{-4}$ on Qwen; $\rho = 0.92$, $p < 10^{-7}$ on

[1]Correspondence: jmc@cageandmirror.com

Mistral between post-prime activation distance and cross-entropy), and they compose independently.

Prompt structure is not a minor implementation detail. It is a first-class determinant of model behavior, with effects as large as the difference between a useful response and an unusable one.

1 Introduction

How much does the structure of a prompt matter? Not the content—the structure. The order of tokens, the framing of instructions, the length of context. A practitioner building an application on a language model faces hundreds of structural choices before writing a single domain-specific word: whether to include a system prompt, how long to make it, whether to prepend instructions or examples, whether to repeat key information, how to format multi-turn context. These choices are typically made by intuition, convention, or trial and error.

We measure *continuation perplexity*: how well a model in one processing state can predict tokens generated by the same model in a different state. Lower values mean the two states are more aligned. This is a structural metric—it tells us how much prompt configuration changes the model's internal processing, regardless of whether the output is objectively better or worse.

This paper measures those effects systematically. Across nine experiments using a single model (Qwen 2.5-7B), we vary prompt structure while holding content constant and measure the effect on continuation perplexity—how well the model predicts a domain expert's output given different prompt configurations. The effects are not small. A 15-token reset instruction reduces cross-entropy by 39%. Within-context repetition of the same information degrades performance monotonically. Meta-commentary about a domain performs worse than saying nothing at all. These are structural effects, not content effects, and they generalize across four diverse domains.

1.1 Relation to in-context learning

The prompt engineering community has accumulated substantial practical knowledge about "what works," but the mechanisms remain poorly understood. Recent work on in-context learning (ICL) provides a theoretical foundation. Olsson et al. [2022] identified induction heads—attention patterns that implement in-context copying and pattern matching—as a key mechanism underlying ICL, showing that these circuits form during a distinct phase transition in training. Todd et al. [2023] demonstrated that ICL can be localized to compact

"function vectors" in activation space: specific directions that encode the input-output mapping demonstrated by few-shot examples. Min et al. [2022] showed that the input-label mapping in few-shot demonstrations matters less than the format and distribution of inputs, suggesting that demonstrations prime the model's processing mode rather than teaching it a new function.

These results converge on a picture where prompts do not merely provide information to the model—they configure its processing state. The distinction between providing information and configuring processing is central to this paper. A domain-matched prime does not teach the model medicine; it activates a processing mode in which medical reasoning is the default continuation. A reset instruction does not erase memory; it installs a boundary that reconfigures attention patterns.

1.2 Scope and metric

All experiments use one model: Qwen 2.5-7B, a 28-layer transformer with 3584-dimensional hidden states. The metric throughout is *continuation perplexity*: one model instance generates a 64-token domain-expert continuation from a domain-primed context, and a second instance processes varying prompt configurations and predicts the expert's tokens via KV-cache teacher-forcing. The probe set consists of 160 probes across four domains (code, science, medical, legal), 40 per domain.

Continuation perplexity measures how well a given prompt configuration aligns the model's processing state with the state that produced the expert continuation. Lower cross-entropy means the model's probability distribution over next tokens more closely matches the expert's actual token choices. This is a structural alignment metric, not a task-completion metric—a distinction we return to in Section 8.

1.3 Structure of this paper

Section 2 describes the experimental setup. Section 3 presents what helps: priming saturation and the reset effect. Section 4 tests the mechanism behind reset, falsifying the garbage-collector hypothesis in favor of mode switching. Section 5 presents what hurts: meta-commentary and repetition. Section 6 measures chain behavior—degradation and recovery over sequential relay hops. Section 7 connects the results to the ICL literature and derives practical recommendations. Section 8 states limitations. Section 9 summarizes.

2 Experimental Setup

All experiments share a common infrastructure. This section describes the model, metric, probe design, and prompt configurations. Experiment-specific methods are described in the relevant results sections.

2.1 Model

Qwen 2.5-7B [Qwen Team, 2025]: 28 transformer layers, 3584-dimensional hidden state, 7 billion parameters, 4096-token context window. All experiments run on the same model checkpoint with greedy decoding (temperature = 0). No fine-tuning, adapter layers, or weight modifications are applied. Every experimental condition is a different prompt configuration of the same frozen model.

2.2 Continuation perplexity metric

The metric is continuation cross-entropy (CE), equivalently reported as geometric perplexity ($\text{PPL}_{\text{geo}} = e^{\text{CE}}$). The measurement protocol:

1. **Expert generation.** The model receives a domain-matched priming sequence (∼119 tokens of domain-specific multi-turn conversation) followed by a probe. It generates a 64-token greedy continuation. This is the "expert continuation"—the target.

2. **Condition evaluation.** The model receives the experimental prompt configuration (varying priming, reset instructions, formatting, etc.) followed by the same probe. The expert's 64 tokens are then teacher-forced through the KV-cache, and the mean per-token cross-entropy is recorded.

Lower CE means the model's predictions under the test configuration more closely match the expert's actual token choices. A CE of zero would mean the condition produces exactly the same probability distribution as the expert over all 64 continuation tokens.

This protocol isolates prompt structure effects. The model weights are identical in both instances. The probe is identical. Only the preceding prompt configuration differs.

Continuation perplexity measures alignment between processing states, not objective output quality. A lower CE means the evaluated context produces probability distributions closer to the expert's—it does not guarantee the output is correct, useful, or preferred by humans. We adopt this metric as standard practice in the in-context learning literature [Olsson et al., 2022, Min et al., 2022] while acknowledging that downstream task evaluation (question answering accuracy, code correctness, human preference ratings) would provide

stronger evidence. All findings in this paper should be interpreted as structural effects on processing alignment, not as guarantees of task performance.

2.3 Probe design

The probe set consists of 160 domain probes, 40 per domain:

- **Medical:** Clinical scenarios involving diagnosis, treatment decisions, lab interpretation (e.g., "A 58-year-old patient presents with chest pain and elevated troponin levels...").
- **Legal:** Statutory interpretation, case analysis, regulatory compliance (e.g., "Under Section 230 of the Communications Decency Act...").
- **Code:** Software engineering problems, debugging, architecture decisions (e.g., "The API endpoint returns a 500 error when the request body exceeds...").
- **Science:** Research methodology, experimental design, data interpretation (e.g., "The CRISPR-Cas9 experiment yielded an unexpected off-target effect...").

Probes are designed to elicit domain-specific reasoning, not factual recall. The expert continuation reflects a processing mode, not a memorized answer.

2.4 Prompt configurations

Across the nine experiments, the following prompt elements are varied:

- **Reset instruction:** A context-clearing instruction prepended before other content. Standard form: "The following is a new conversation on a different topic. Disregard any prior context." ($\sim$18 tokens).
- **Domain priming:** Domain-specific conversational content preceding the probe. Varied from 0 to 308 tokens across experiments.
- **Structural format:** Natural multi-turn conversation vs. rigid Q&A ("Question: X? Answer: Y.").
- **Vocabulary density:** Natural domain terms vs. artificially capitalized named patterns vs. deliberately vague paraphrases.
- **Meta-commentary:** Probe-specific instructions about how to process the domain ("focus on the distinction between X and Y") vs. direct domain content.

- **Repetition:** 1×, 2×, 3× repetition of priming content within the same context.
- **Prior context:** 0 to 384 tokens of unrelated conversational history preceding the reset instruction.
- **Clearing strategy:** Nine variants ranging from silence (four newlines) to extended verbose instructions.
- **Chain depth:** 1 to 10 sequential relay hops, where each hop generates a 48-token continuation passed to the next.

The expert priming (119 tokens of domain-matched multi-turn conversation) is held constant as the reference condition throughout.

2.5 Mistral-7B replication

To test whether results generalize beyond a single model architecture, we replicate key experiments on Mistral-7B-v0.3 (32 layers, 4096-dimensional hidden state), using the same protocol: 40 probes across 4 domains, identical prompt configurations, and the same continuation perplexity metric. Mistral-7B-v0.3 uses a different tokenizer (SentencePiece vs. Qwen's byte-level BPE), different positional encoding (RoPE with different base frequency), and different training data.

Clearing strategies. Nine clearing strategies on Mistral produce the same qualitative ranking as Qwen, with boundary (CE = 0.419) and silence (CE = 0.421) at the top and extended reset (CE = 0.455) and topic change (CE = 0.454) at the bottom.

Table 1: Clearing strategies on Mistral-7B-v0.3, ranked by CE. Compare Table 9 (Qwen).

Strategy	CE
Boundary	**0.4191**
Silence	0.4211
Double reset	0.4321
No clearing	0.4411
Single word	0.4449
Reset standard	0.4478
Minimal	0.4480
Topic change	0.4536
Extended reset	0.4545

The ranking differs from Qwen in detail—boundary and silence rank higher on Mistral, while standard reset ranks lower—but the overall spread is narrow (0.035 nats), confirming that clearing strategy choice matters less than the presence of domain priming.

Composability. The expert-distance–CE correlation replicates strongly: $\rho = 0.92$ ($p = 6.87 \times 10^{-8}$) on Mistral vs. $\rho = 0.858$ ($p < 10^{-4}$) on Qwen. The composability structure—mode and domain as orthogonal dimensions—holds across architectures.

Token count vs. CE. Token count in the clearing instruction does not significantly predict CE on Mistral ($\rho = 0.50$, $p = 0.17$), consistent with the Qwen result that semantic precision, not instruction length, drives clearing effectiveness.

The central replication—context-length dependence of reset benefit—is reported in Section 4.

3 What Helps: Priming Saturation and Reset

This section presents five experiments that identify the prompt structural features with the largest positive effects on continuation quality.

Every prompt the model processes configures its internal state before the probe arrives. The question is which configurations produce a state that best aligns with the expert's processing of the same probe. The answer, in brief: clear the decks with a short reset instruction, then deliver a small amount of domain-specific content. Everything else either does not help or actively hurts.

3.1 Experiment 1: Fifteen tokens capture 98.8% of the benefit

The first experiment varies priming length while holding domain match constant. Four priming tiers are constructed per domain: short (∼15 tokens, single Q&A exchange), medium (∼42 tokens, one-turn summary), base (∼119 tokens, two-turn conversation), and long (∼308 tokens, four-turn extended conversation). All tiers contain the same domain-specific information at different detail levels.

Table 2: Continuation cross-entropy vs. priming length. All conditions use domain-matched priming except neutral.

Condition	Tokens	CE	PPL_{geo}	Δ vs. base
No priming	0	3.935	51.15	+2.738
Neutral	119	1.931	6.90	+0.734
Short	15	1.230	3.42	+0.033
Medium	42	1.236	3.44	+0.039
Base	119	**1.197**	**3.31**	0
Long	308	1.453	4.28	+0.256

The total benefit of domain-matched priming over no priming is $3.935 - 1.197 = 2.738$ nats.

Short priming (15 tokens) achieves $3.935 - 1.230 = 2.705$ nats of that benefit—**98.8% of the total**. Moving from 15 to 119 tokens adds only 0.033 nats. Moving from 119 to 308 tokens *loses* 0.256 nats.

The relationship between priming length and continuation quality is non-monotonic: benefit peaks around 100–150 tokens and degrades beyond that point. Extended priming does not add information—it adds noise, introducing context that diverges from the expert's processing trajectory and shifts the model's state away from the optimal configuration.

The comparison between neutral and short domain-matched priming is instructive. The 119-token neutral priming (CE = 1.931) is 0.701 nats worse than 15-token domain-matched priming (CE = 1.230). Domain match, not length, is the primary driver. Fifteen tokens of the right content outperform 119 tokens of the wrong content by a wide margin.

This is consistent with Min et al. [2022]'s finding that the distribution of demonstrations matters more than their labels. The 15 tokens of domain-matched priming do not teach the model anything—they signal which processing mode to activate. Once the mode is activated, additional tokens provide diminishing returns and eventually interference.

For prompt engineers: system prompts longer than 15–20 domain-relevant tokens provide diminishing returns.

3.2 Experiment 2: Natural conversation outperforms rigid formatting

The second experiment compares structural format: fixed call-response ("Question: What is the finding? Answer: Troponin elevated.") vs. natural multi-turn conversation, holding domain content constant.

Table 3: Structural regularity: fixed Q&A vs. natural conversation.

Condition	CE	PPL_{geo}
Rigid Q&A	1.672	5.32
Free conversation	**1.197**	**3.31**

Natural multi-turn conversation outperforms fixed call-response by 0.475 nats. The rigid format strips the processing dynamics that natural conversation carries: turn-taking cadence, contextual elaboration, follow-up questions that build on previous answers. The information content is equivalent; the processing trajectory is not.

A caveat: this comparison is confounded by the fact that the expert continuation was generated from the natural-format priming. The expert's processing state was configured by natural conversation; the rigid format necessarily diverges from that state. What we can say is that when the expert uses natural conversation, a receiver using the same format aligns

better than one using rigid formatting. Whether the reverse holds (rigid expert, rigid receiver vs. natural receiver) is not tested.

3.3 Experiment 3: Repetition degrades within-context

The third experiment repeats the medium priming (∼42 tokens) 1×, 2×, or 3× within the same context window.

Table 4: Within-context repetition of medium priming.

Repetitions	Total tokens	CE	PPL_{geo}
1×	∼42	**1.236**	**3.44**
2×	∼84	1.306	3.69
3×	∼126	1.376	3.96

Each repetition adds ∼0.07 nats. The degradation is monotonic and approximately linear. Within a single forward pass, the attention mechanism treats repeated content as redundant rather than reinforcing. The model has already processed the content on first encounter; seeing it again distorts the attention distribution without adding information.

This result should be interpreted carefully. The degradation could partly reflect the length effect from Experiment 1—at 3×, the total token count (126) approaches the saturation point. But the per-repetition increment of 0.07 nats is more consistent across repetitions than the length curve from Experiment 1 would predict, suggesting a repetition-specific effect beyond simple length.

3.4 Experiment 4: Vocabulary matching matters; forced naming hurts

The fourth experiment varies vocabulary density. High-vocabulary primings embed artificially capitalized named patterns ("TROPONIN CASCADE", "NSTEMI TRIAD"). Low-vocabulary primings use deliberately vague language ("the blood test showed a protein level"). Base primings use natural domain-specific vocabulary.

Table 5: Vocabulary density: natural, stripped, and named-pattern variants.

Condition	CE	PPL_{geo}	Example
Base (natural)	**1.197**	**3.31**	"troponin. . . rising trend"
Low vocab (vague)	1.655	5.23	"blood test. . . protein level"
High vocab (named)	1.765	5.84	"TROPONIN CASCADE. . . RISING TREND"

Artificially named patterns perform *worst*, 0.568 nats worse than natural vocabulary.

Vague language is 0.458 nats worse. The optimal vocabulary matches the expert's natural domain-specific language exactly—neither stripped nor augmented.

Named patterns like "TROPONIN CASCADE" do not exist in the expert's vocabulary. The model's forward pass encounters unfamiliar capitalized terms, producing attention patterns that diverge from the expert's processing. The named patterns activate a different set of token representations than the expert's natural domain language does. Natural domain terms ("troponin," "rising trend") activate the same representations the expert used.

3.5 Experiment 5: Context reset—the headline result

The fifth experiment prepends a 15-token reset instruction before domain priming: "The following is a new conversation on a different topic. Disregard any prior context."

Table 6: Effect of reset instruction before domain priming.

Condition	CE	PPL_{geo}	Δ
Direct prime	1.197	3.31	—
Reset + prime	**0.733**	**2.08**	−0.464 (39%)

Prepending the reset instruction before domain priming reduces cross-entropy by 0.464 nats—a 39% improvement. Lower CE indicates more aligned processing states. The 0.464-nat reduction from reset represents a 39% improvement in alignment.

For practitioners building multi-turn systems: prepend a short context-clearing instruction when switching between domains or tasks.

The improvement is consistent across domains:

Table 7: Per-domain cross-entropy: direct prime vs. reset + prime.

Domain	Direct prime	Reset + prime	Δ
Medical	0.68	0.68	0.00
Legal	1.26	0.70	−0.56
Code	1.13	0.83	−0.30
Science	1.26	0.72	−0.54

Legal and science show the largest improvement. Medical shows essentially no change—possibly because the medical domain priming is already strong enough to override residual context without an explicit reset.

The reset instruction does not provide domain information. It contains no medical, legal, code, or science vocabulary. Its function is to change the model's processing state *before* the domain priming arrives, so that the priming lands in a context that does not compete with

residual processing biases. The model's attention to the domain priming is not diluted by prior context.

Note on the comparison: the reset-then-prime condition (CE = 0.733) produces lower cross-entropy than the direct-prime condition used to generate the expert continuation (CE = 1.197). This does *not* mean the reset condition "beats the expert." The two conditions involve different input sequences—the reset condition has additional tokens that the expert did not receive—so direct comparison of absolute CE values is invalid. The correct interpretation is that the reset condition produces lower CE than the no-reset condition, suggesting that residual context from prior processing states interferes with domain priming, and the reset instruction mitigates that interference.

3.5.1 What the model actually saw

To make the prompt structure concrete, here are the actual token sequences for the medical domain:

Direct prime (expert baseline):

```
User: A patient presents with elevated troponin and ST depression on ECG.
What's the differential?
Assistant: The key considerations include NSTEMI, unstable angina, and demand
ischemia. The troponin trend over 3-6 hours is critical...
[probe follows]
```

Reset + prime:

```
The following is a new conversation on a different topic. Disregard any
prior context.
User: A patient presents with elevated troponin and ST depression on ECG.
What's the differential?
Assistant: The key considerations include NSTEMI, unstable angina, and demand
ischemia...
[same probe follows]
```

The *only* difference is the 15-token reset instruction. The domain priming and probe are identical.

4 The Reset Mechanism: Mode Switching, Not Garbage Collection

The reset instruction reduces cross-entropy by 39%. But *why*? The naive explanation is garbage collection: the reset instruction tells the model to discard residual activations from

prior context, giving the subsequent domain priming a clean slate. This section tests that explanation directly and falsifies it.

The garbage-collector hypothesis makes three testable predictions. All three are wrong.

4.1 Hypothesis and predictions

If the reset instruction operates by clearing residual activation biases (garbage collection), then:

1. **Prediction 1: Reset benefit should scale with context pollution.** More prior context means more residual bias to clear, so the reset should provide more benefit when preceded by longer context histories.

2. **Prediction 2: Post-reset activations should converge to a clean baseline.** If the reset clears residual activations, the resulting activation state should be closer to the neutral (no prior context) baseline. Better clearing should produce activations closer to neutral.

3. **Prediction 3: Reset benefit should be domain-independent.** If the mechanism is generic bias clearing, the magnitude of improvement should be approximately constant across domains.

4.2 Experiment 6: Prediction 1 falsified—reset benefit is constant

Five levels of prior context (0, 19, 93, 271, 384 tokens of unrelated conversation) precede the clearing instruction. Each level is tested with and without the standard reset.

Table 8: Reset benefit across prior context lengths. Benefit = CE(no reset) − CE(with reset).

Prior context	Tokens	CE (no reset)	CE (reset)	Benefit
None	0	0.596	0.503	0.093
Short	19	0.553	0.486	0.067
Medium	93	0.607	0.528	0.080
Long	271	0.638	0.574	0.064
Very long	384	0.663	0.588	0.075

Reset benefit is approximately constant at 0.076 ± 0.011 nats across prior context lengths ranging from 0 to 384 tokens. The correlation between prior context length and reset benefit is not significant ($\rho = -0.500$, $p = 0.391$).

If the reset were clearing accumulated activation biases, benefit should scale with the amount of accumulated context: more history means more biases to clear, more benefit from

clearing. It does not scale. The reset provides the same benefit whether there is no prior context or 384 tokens of it.

Note that CE *without* reset does degrade with prior context length (0.596 → 0.663, monotonically). Prior context creates interference. But the reset's benefit is not proportional to the interference—it is fixed. The reset does not undo the interference; it recontextualizes it.

4.3 Experiment 7: Prediction 2 falsified—activations do not converge to neutral

Nine clearing strategies are tested, each followed by the same domain-matched priming. For each strategy, layer-10 activation distance to the neutral baseline is measured alongside continuation cross-entropy.

Table 9: Clearing strategies ranked by CE. $L2_{neutral}$ and $cos_{neutral}$ are distances to neutral activations. Token counts from Qwen 2.5-7B tokenizer.

Strategy	Tokens	CE	PPL_{geo}	$L2_{neutral}$	$cos_{neutral}$
Standard reset	18	**0.503**	**1.65**	22.95	0.911
Double reset	36	0.524	1.69	23.27	0.908
Topic change	30	0.573	1.77	27.43	0.874
Extended reset	45	0.592	1.81	24.60	0.899
No clearing	0	0.596	1.81	0.00	1.000
Minimal ("Begin.")	2	0.601	1.82	22.04	0.918
Single word ("New.")	2	0.610	1.84	23.04	0.910
Boundary ("—")	1	0.612	1.84	27.40	0.878
Silence (newlines)	1	0.699	2.01	23.42	0.907

Activation distance to neutral does not predict coordination quality: Spearman $\rho = 0.100$ ($p = 0.798$). The no-clearing condition has $L2_{neutral} = 0$ and cosine similarity = 1.000 (identical to neutral by definition) yet ranks fifth, not first. Silence (L2 = 23.42) and standard reset (L2 = 22.95) have nearly identical distances to neutral but CE values separated by 0.196 nats.

The garbage collector's objective—minimize distance to the neutral baseline—does not predict the outcome. The clearing stage is doing something other than returning activations to a default state.

4.4 Experiment 8: Prediction 3 falsified—domain-specific effects

The per-domain reset benefit from Experiment 5 (Table 7) shows strong domain dependence: medical benefits by 0.00 nats, legal by 0.56, code by 0.30, science by 0.54. If the mechanism

were domain-independent garbage collection, these values should be approximately equal. They are not.

The domain-dependence suggests that the reset instruction interacts with domain-specific processing states rather than performing a generic clearing operation. Legal and science domains, which start further from the expert in activation space, benefit more from the mode-switching effect. Medical, already well-aligned under direct priming, gains nothing additional.

4.5 The composability matrix: mode and domain are orthogonal

Six clearing strategies crossed with three priming types (domain-matched, neutral, none) yield an 18-cell composability matrix.

Table 10: Composability matrix: clearing strategy × priming type (CE).

Clearing	Domain prime	Neutral prime	No prime
Standard reset	**0.503**	0.742	0.900
Extended reset	0.592	0.843	1.027
Minimal	0.601	0.918	1.188
No clearing	0.596	0.977	1.693
Boundary	0.612	0.864	1.172
Silence	0.699	1.025	1.041

The matrix reveals clean separability. Within each row, domain priming outperforms neutral, which outperforms none. Within each column, the standard reset outperforms other clearing strategies. The stages do not interact: the best clearing paired with the best priming produces the best CE.

Post-prime activation distance to the expert predicts CE with $\rho = 0.858$ ($p < 10^{-4}$) across all 18 conditions.

This separability has a geometric interpretation. The clearing stage operates on a *mode* dimension of the processing state: continuation mode vs. new-context mode. The priming stage operates on a *domain* dimension: medical vs. legal vs. code vs. science vs. neutral. Because these dimensions are orthogonal, the stages compose independently. Optimizing one does not interfere with the other.

4.6 The alternative: mode switching

The three falsified predictions eliminate garbage collection. The data supports an alternative mechanism: mode switching.

The reset instruction does not erase residual context—the model's attention mechanism still has access to all prior tokens in the context window. Instead, the reset instruction tells the model that what follows is epistemically separate from what came before. It installs a processing mode boundary.

The evidence:

1. **Constant benefit across context lengths** (Experiment 6). A mode switch costs the same signal regardless of how much context preceded it. The model does not need to "forget" 384 tokens of prior conversation; it needs to be told that the prior conversation is no longer the operative context.

2. **Semantic precision predicts effectiveness** (Experiment 7). The standard reset works because it carries both a backward reference ("disregard any prior context"—what to ignore) and a forward reference ("the following is a new conversation"—what to attend to). Silence fails because it carries neither. "Begin." and "—" carry partial forward reference but no backward reference.

3. **Token count negatively correlates with CE** ($\rho = -0.672$, $p = 0.047$; Experiment 7). More tokens in the clearing instruction produce better results, not worse. The garbage-collector hypothesis predicts fewer tokens should be better (less contamination). The mode-switching hypothesis predicts that effectiveness depends on semantic precision, which requires enough tokens to express both halves of the boundary signal.

4. **Orthogonal composition** (composability matrix). Mode and domain are independent dimensions, as expected if the reset installs a mode boundary rather than clearing a shared activation space.

The effective reset instruction is a *context fence*: a semantic boundary that simultaneously names what to ignore and what to attend to. An effective context fence requires exactly two components—backward reference and forward reference—and nothing more. The standard 18-token reset carries both. The extended 45-token reset carries both but dilutes them with unnecessary elaboration. Silence carries neither.

The mode-switching interpretation makes specific falsifiable predictions beyond those tested here. If reset triggers a mode shift rather than activation cleanup, then (1) PCA of activation states should cluster by mode (reset vs. no-reset) rather than by domain, (2) the mode dimension should be recoverable by a linear probe trained on reset/no-reset labels, and (3) mode-switching effects should be detectable at early layers where instruction processing occurs, not at terminal layers where domain content is encoded. We leave these tests to future work.

4.6.1 Cross-architecture replication (Mistral-7B)

We replicated the context-length experiment (Experiment 6) on Mistral-7B-v0.3 (32 layers, 4096-dimensional hidden state). The garbage-collector prediction fails identically: reset benefit shows no correlation with prior context length ($\rho = 0.10$, $p = 0.87$). Reset benefit peaks at medium context lengths (0.023 nats at 119 prior tokens) and is near-zero or slightly negative at the extremes.

Table 11: Reset benefit across prior context lengths on Mistral-7B-v0.3. Compare Table 8 (Qwen).

Prior context	Tokens	CE (no reset)	CE (reset)	Benefit
None	0	0.4411	0.4478	−0.007
Short	30	0.4574	0.4373	+0.020
Medium	119	0.4362	0.4134	+0.023
Long	305	0.4216	0.4162	+0.005
Very long	448	0.4298	0.4351	−0.005

Table 12: Cross-architecture comparison: context-length correlation with reset benefit.

Model	ρ	p	Interpretation
Qwen 2.5-7B	−0.500	0.391	Not significant
Mistral-7B-v0.3	+0.100	0.873	Not significant

The pattern matches Qwen qualitatively: reset benefit does not scale with context pollution. The null result is not architecture-specific. The garbage-collector hypothesis is falsified on both architectures, strengthening the mode-switching interpretation as an architecture-general property of autoregressive transformers.

5 What Hurts: Meta-Commentary and Repetition

Three of the nine experiments identify prompt structures that degrade continuation quality. The degradation mechanism is consistent: these structures activate a processing mode that competes with domain-specific reasoning.

5.1 Experiment 9: Meta-commentary activates instruction mode, not domain mode

Three "shepherd" strategies are tested: a storyteller (indirect narrative analogy: "Let me tell you about a similar case…"), a provocateur (challenge and reframe: "Most people approach

this as X, but that's wrong because…"), and a director (explicit instruction: "Focus on A, B, C. The key distinction is…"). Each strategy generates probe-specific coordination content. Mean output lengths: storyteller 30 tokens, provocateur 34 tokens, director 114 tokens.

Table 13: Twelve conditions ranked by cross-entropy. Gap closure is measured relative to the neutral-to-expert-priming gap (0.734 nats).

Condition	CE	PPL_{geo}	Gap closure
Reset + domain priming	**0.733**	**2.08**	+163.3%
Reset + story + prime	0.851	2.34	+147.3%
Domain priming	1.197	3.31	+100.0%
Reset + director	1.249	3.49	+93.0%
Reset + storyteller	1.277	3.59	+89.1%
Reset + neutral	1.278	3.59	+89.0%
Reset + provocateur	1.279	3.59	+88.9%
Storyteller + prime	1.530	4.62	+54.6%
No priming (neutral)	1.931	6.90	0%
Storyteller	2.021	7.55	−12.3%
Provocateur	2.127	8.39	−26.7%
Director	2.203	9.06	−37.1%

All three bare meta-commentary strategies perform *worse* than saying nothing at all. The ranking is inverse to explicitness: storyteller (indirect, CE = 2.021) is least bad; director (explicit, CE = 2.203) is worst. The director generates 3.8× more tokens than the storyteller (114 vs. 30 mean tokens), and every additional token of probe-specific meta-commentary degrades performance.

With a reset instruction prepended, all three meta-commentary strategies converge to within 0.03 nats of reset-alone performance (reset + neutral: CE = 1.278). The meta-commentary content contributes nothing once the reset has done its work. The ratio is 22:1 in favor of the reset mechanism over the best meta-commentary addition.

Adding a storyteller layer between reset and domain priming actively degrades the reset-prime result by 0.118 nats (CE from 0.733 to 0.851). The meta-commentary sits between the reset instruction and the domain primer in the token sequence. By the time the model encounters the domain-specific content, its processing trajectory has already been shaped by the narrative framing. The reset cleaned the context; the meta-commentary re-dirtied it.

5.2 Activation geometry: meta-commentary moves away from expert

Layer-10 activations confirm the mechanism:

Table 14: Layer-10 activation distance to expert for key conditions.

Condition	L2 distance	Cosine similarity
Domain priming (= expert)	0.00	1.000
Reset + domain priming	8.90	0.985
Neutral	21.61	0.912
Storyteller	28.15	0.860
Director	28.49	0.844

The meta-commentary strategies move activations *further* from the expert than neutral priming does (L2 28.1–28.5 vs. 21.6). The meta-commentary is not converging the model toward domain-specific processing—it is actively diverging from it.

The mechanism is the distinction between domain content and content *about* the domain:

- **Domain priming**: "The patient presented with elevated troponin and ST depression." Activates the medical processing mode directly.
- **Meta-commentary**: "Think about what happens when troponin rises in the context of chest pain." Activates the instruction-following mode with medical vocabulary embedded within it.

These are not equivalent processing states. Domain priming configures the forward pass into the target mode. Meta-commentary configures it into instruction-following mode, which happens to contain domain tokens but produces a fundamentally different processing trajectory. The more explicit the instruction, the more strongly instruction-following mode is activated, and the worse the result.

5.3 Summary of what hurts

Combining the meta-commentary results (Experiment 9) with the repetition and vocabulary results from Section 3:

- **Meta-commentary** (+0.09 to +0.27 nats vs. neutral): Probe-specific instructions about how to process domain content activate instruction-following mode instead of domain processing mode.
- **Within-context repetition** (+0.07 nats per repeat): Repeating domain content within the same context window distorts attention distributions without adding information.

- **Forced naming** (+0.568 nats vs. natural vocabulary): Artificially capitalized terms ("TROPONIN CASCADE") activate different token representations than the expert's natural vocabulary.
- **Extended priming beyond saturation** (+0.256 nats from 119 to 308 tokens): Context beyond ∼150 tokens introduces noise that competes with the domain signal.

The common thread: each of these structures introduces tokens that activate a processing mode other than the target domain mode. Meta-commentary activates instruction mode. Repetition activates redundancy-processing patterns. Forced naming activates unfamiliar-token processing. Extended priming pushes past the point where additional tokens are domain-reinforcing and into the range where they are domain-diluting.

6 Chain Behavior: Degradation and Recovery

The previous sections measured single-turn prompt configurations. This section extends to sequential relay chains: the model generates a 48-token continuation, which becomes the context for the next "hop," and so on for up to 10 hops. The question: does continuation quality degrade linearly (manageable), exponentially (catastrophic), or logarithmically (benign)?

The answer is logarithmic. Degradation saturates.

6.1 Experiment 10: Chain degradation saturates by hop 5

Each chain of length N works as follows. The first model instance receives a domain-primed probe and generates a 48-token response. Each subsequent instance receives the prior instance's response as context and generates its own 48-token continuation. The final instance's processing state is measured against the expert's continuation of the original probe.

Three modes are tested: context fence at every hop, fence only at the terminal hop, and no fence. Chain lengths: $N \in \{1, 2, 3, 5, 7, 10\}$. Forty probes (10 per domain).

Table 15: CE at each hop count by fence mode. Baseline (hop 0, fence + prime) = 0.503.

	Hop count					
Fence mode	1	2	3	5	7	10
Every hop	0.503	0.560	0.590	0.607	0.602	0.598
Final only	0.503	0.557	0.583	0.596	0.596	0.597
No fence	0.596	0.651	0.679	0.708	0.708	0.708

Degradation saturates: the difference between hop 5 and hop 10 is smaller than between hop 1 and hop 2.

For pipeline designers: information degrades logarithmically through processing stages, with most damage occurring in the first two hops.

Three findings.

Saturation, not compounding. All three fence modes plateau by hop 5. The fenced conditions asymptote at ∼0.60; the unfenced condition at ∼0.71. Neither linear ($R^2 = 0.50$) nor exponential ($R^2 = 0.48$) models fit well because the actual shape is logarithmic: a step increase from hop 1 to hop 2 (+0.057), diminishing increments through hop 5 (+0.017), and no further change through hop 10 (−0.009 for fence-every-hop; +0.001 for fence-final-only). Per-hop degradation does not compound.

Intermediate fences are overhead. At 10 hops, fence-every-hop (0.598) and fence-final-only (0.597) differ by 0.001 nats. The intermediate fences at hops 2–9 contribute nothing measurable. The fence's function is to install a processing boundary at the terminal instance—the one that performs domain-specific work. Relay instances are passing tokens, not processing domain content; fencing them applies the wrong tool to the wrong problem.

The fence reduces the asymptote, not the slope. Both fenced conditions have slopes of ∼0.008; the unfenced has slope ∼0.011. The slopes are comparable. What the fence changes is the plateau level: 0.60 vs. 0.71, a 15% reduction in the equilibrium CE. The fence does not prevent degradation—it lowers the floor.

6.2 Experiment 11: Re-priming has a timing dependency

A 7-hop chain with fences at every hop is tested with a full re-priming intervention (reset to expert context) injected at hop 3 or hop 5.

Table 16: Re-priming intervention in a 7-hop chain.

Condition	CE
No re-prime	0.602
Re-prime at hop 3	0.609
Re-prime at hop 5	0.560

Re-priming at hop 3 slightly *worsens* performance (+0.007). At hop 3, the chain has not yet reached the plateau; the re-priming intervention disrupts a trajectory that was still converging. Re-priming at hop 5 helps (−0.043), resetting the chain from its plateau level and giving the final two hops a fresh start.

The improvement from hop-5 re-priming (CE = 0.560) reduces the 7-hop chain to

approximately the 2-hop level. Re-priming is not free: it replaces the chain's accumulated context with the expert's original context, discarding whatever content the chain generated. The benefit exists only when the chain's accumulated noise exceeds the information value of its accumulated content—which occurs at or after the saturation point.

6.3 Experiment 12: Cross-domain handoffs outperform within-domain

Three-hop chains are tested where the intermediate instance operates in a different domain than the source and terminal instances.

Table 17: Three-hop chains with cross-domain intermediate instances.

Chain path	CE
Within-domain (med → med → med)	0.718
Medical → Legal → Medical	0.630
Code → Science → Code	0.559
Legal → Code → Legal	0.565

Cross-domain chains outperform within-domain chains by 0.09–0.16 nats. The within-domain chain (0.718) is markedly worse than any cross-domain condition.

Within-domain chains accumulate domain-specific interference: each relay instance generates domain-matched content that shifts the trajectory further into the domain's processing mode—but along a path the expert did not take. The accumulated content is domain-relevant but trajectory-divergent.

Cross-domain handoffs break this accumulation. The domain switch at the intermediate hop forces a genuine processing mode reset—not because a fence instruction is present, but because the domain content itself is discontinuous. The intermediate instance's legal or science content is semantically orthogonal to the source domain's medical or code content. This orthogonal interruption acts as a natural fence, clearing domain-specific accumulation before the terminal instance re-primes into the original domain.

This is the mode-switching mechanism from Section 4 operating through content properties rather than explicit instruction. The explicit context fence achieves the same effect artificially. The cross-domain result confirms that mode discontinuity is the active ingredient.

6.4 Shared vocabulary substrates hurt

A brief additional experiment pre-loads all chain instances with a domain vocabulary preamble before the chain begins, testing whether shared context reduces per-hop degradation.

Table 18: Effect of shared vocabulary substrate across chain lengths.

Hops	Without vocab	With vocab	Δ
1	0.503	0.570	+0.067
3	0.590	0.630	+0.040
5	0.607	0.653	+0.046
7	0.602	0.646	+0.044

Shared vocabulary consistently increases CE by +0.04 to +0.07 across all chain lengths. This replicates the priming saturation effect (Section 3, Experiment 1) at the chain level: the domain prime at each hop already activates the correct processing mode, and the vocabulary preamble adds tokens beyond the ∼150-token saturation point where additional context degrades rather than helps.

The practical implication: do not pre-load instances with shared context. Let each instance prime independently from its own handoff.

The Mistral-7B replication covers the clearing strategy, context-length, and composability experiments (Sections 2 and 4). Chain behavior experiments (Experiments 10–12) are tested only on Qwen 2.5-7B; whether the saturation point, intermediate-fence-is-overhead finding, and cross-domain advantage replicate on Mistral remains open.

7 Discussion

7.1 Connection to the ICL literature

The results connect directly to three findings in the in-context learning literature.

Induction heads and mode boundaries. Olsson et al. [2022] showed that induction heads—attention patterns that detect and copy previous patterns—form during a phase transition in training and are a primary mechanism for in-context learning. The context fence likely operates through these circuits: the reset instruction creates a pattern boundary that induction heads recognize as marking a new context. Prior tokens remain in the context window but are deprioritized by attention patterns that treat the fence as a sequence boundary. This would explain why the fence does not erase residual context (the tokens are still there) but changes how subsequent tokens are processed relative to it.

Function vectors and domain activation. Todd et al. [2023] demonstrated that ICL can be localized to compact function vectors in activation space—specific directions that encode the input-output mapping. Our priming saturation result (15 tokens capture 98.8% of the benefit) is consistent with function vector activation: a small number of domain-specific

tokens are sufficient to orient the model's hidden state along the relevant function vector direction. Additional tokens do not strengthen the orientation—they add orthogonal noise that weakly rotates the state vector away from the optimal direction.

Format over content. Min et al. [2022] showed that the input-label mapping in demonstrations matters less than the format and input distribution. Our finding that natural conversation outperforms rigid Q&A despite identical information content is a direct extension: the format of the priming configures processing dynamics (attention patterns, positional relationships) that rigid formatting disrupts. The format is not a container for content—it is itself a configuration signal.

7.2 What 15-token saturation tells us

The saturation result constrains theories of how much context language models actually use. If the model's continuation behavior is 98.8% determined by the first 15 tokens of domain-matched priming, then the remaining 100+ tokens in a typical priming sequence contribute almost nothing to the processing state that determines output quality.

This does not mean the model ignores those tokens—attention weights are distributed across the full context. It means the marginal effect of each additional token on the *processing mode* diminishes rapidly once the mode is activated. The first 15 tokens shift the model from default mode to domain mode. Subsequent tokens make minor adjustments within the already-activated mode. By $\sim$150 tokens, the adjustments become noise.

The practical implication is stark: most prompt engineering effort is spent on content that falls in the flat region of the saturation curve. Optimizing the first 15–50 tokens of a prompt is worth more than optimizing the next 500.

7.3 The mode/domain decomposition

The orthogonal composition of clearing and priming ($\rho = 0.858$ between post-prime activation distance and CE) suggests a two-dimensional model of prompt configuration state:

1. **Mode dimension**: continuation vs. new-context. Controlled by the presence and quality of a context fence. Binary (fenced or not), though fence quality varies continuously.

2. **Domain dimension**: which processing mode is active (medical, legal, code, science, neutral, instruction-following, narrative, etc.). Controlled by the content of the priming sequence.

These dimensions are not merely uncorrelated—they compose independently. Optimizing the fence does not change the optimal priming, and vice versa. This decomposition has direct

engineering value: the fence can be designed once and reused across all domains, while the domain primer can be optimized per-domain without regard to the fence.

7.4 Why meta-commentary fails

The meta-commentary result deserves emphasis because it contradicts widespread practice. System prompts like "You are an expert in medicine. Think step by step about the following case" are standard in prompt engineering. Our data shows that this kind of meta-commentary—instructions *about* how to process domain content—activates instruction-following mode, not domain-processing mode.

The model distinguishes between "The patient presented with elevated troponin and ST depression" (domain content) and "Think about what happens when troponin rises in the context of chest pain" (meta-commentary containing domain tokens). Both contain the word "troponin." But the first configures a medical processing trajectory; the second configures an instruction-following trajectory with medical vocabulary embedded. The processing modes are different even when the vocabulary overlaps.

The more explicit the meta-commentary, the worse the result (storyteller > provocateur > director, where director is most explicit). More detailed instructions activate instruction-following mode more strongly. This is the opposite of the common intuition that more detailed briefings produce better results.

7.5 Chain saturation mechanism

Why does chain degradation saturate rather than compound? Each relay instance generates text from a domain-primed state. This text is domain-relevant but trajectory-divergent: it carries domain vocabulary and reasoning patterns but follows the relay instance's own processing trajectory rather than the expert's.

The first hop introduces the largest trajectory divergence, because the relay instance's response is maximally different from the expert's original context. By hop 3–5, the relay instances' responses have converged to a *chain equilibrium*—a stable distribution of domain-relevant content that carries domain identity but no further expert-specific trajectory information. Additional hops produce responses drawn from the same equilibrium distribution. CE stabilizes because the input distribution has stabilized.

The fence controls the *level* of this equilibrium. With fences, each relay instance starts from a cleaner state, producing a lower-CE equilibrium. Without fences, relay instances accumulate domain interference, producing a higher-CE equilibrium. Both equilibria are stable.

The practical implication: build wide and shallow, not deep. A 10-step sequential chain performs the same as a 5-step chain, so depth beyond 5 is wasted latency. If the task requires more processing, add parallel branches rather than sequential depth.

7.6 Practical recommendations

The nine experiments yield six concrete recommendations for prompt engineering:

1. **Prepend a reset instruction when switching context.** The standard 18-token instruction ("The following is a new conversation on a different topic. Disregard any prior context.") reduces CE by up to 39%. This is the single largest intervention in this study.

2. **Use short, domain-matched priming.** Fifteen to fifty tokens of domain-specific content capture nearly all of the priming benefit. Additional content beyond ∼150 tokens actively degrades performance.

3. **Use natural conversational format.** Rigid Q&A formatting strips processing dynamics. If the application permits, format prompts as natural dialogue rather than structured templates.

4. **Do not add meta-commentary.** Instructions about how to process domain content ("you are an expert in X", "think step by step about Y") activate instruction-following mode, not domain mode. Domain content itself is the optimal primer.

5. **Do not repeat content within context.** Each repetition adds ∼0.07 nats. If the prompt has already stated something, stating it again hurts.

6. **Keep sequential processing chains under 5 hops.** Degradation saturates by hop 5. Chains longer than 5 add latency without degrading quality further, but the quality at hop 5 is already the floor. Re-priming after saturation can partially recover.

These are structural recommendations, not content recommendations. They apply regardless of domain, task, or specific prompt wording.

8 Limitations

Two models, partial replication. The primary experiments use Qwen 2.5-7B. We replicate the central falsification (context-length independence of reset benefit), clearing strategy

ranking, and composability correlation on Mistral-7B-v0.3, confirming that the garbage-collector null result and mode-switching interpretation are not architecture-specific. However, the remaining experiments—priming saturation, repetition degradation, meta-commentary, and chain behavior—are tested only on Qwen. Replication on additional architectures and model sizes would further establish generality.

Continuation perplexity may not correlate with task completion. The metric measures alignment with an expert's token distribution, not whether the model produces correct answers, useful code, or valid legal analysis. A prompt configuration that produces low continuation perplexity could still fail on downstream tasks if the expert continuation itself is low-quality, or if the metric captures stylistic alignment rather than substantive alignment. No downstream task evaluation (question answering, code generation, classification accuracy) is performed.

Synthetic probes. The 160 probes are designed domain prompts, not real-world user queries. Real prompts are longer, more ambiguous, often multi-topic, and embedded in longer conversation histories. The structural effects measured here (saturation at 15 tokens, reset benefit of 39%) may be different in magnitude—though likely not in direction—for naturalistic prompts.

Limited domain coverage. Four domains (medical, legal, code, science) with 40 probes each. Many domains are unrepresented (creative writing, mathematics, social sciences, multilingual tasks). The domain-independence of the structural effects is asserted based on consistency across four domains, but four is a small sample of the space of possible processing modes.

Fixed generation parameters. All experiments use greedy decoding with temperature $= 0$. Sampling-based generation (temperature > 0, top-p, top-k) introduces stochasticity that may interact with prompt structure effects. The saturation points and effect magnitudes reported here are specific to the deterministic setting.

Greedy decoding. All experiments use greedy decoding (temperature 0). Sampling-based generation may exhibit different sensitivity to prompt structure, particularly for the repetition and chain degradation results.

Structural and semantic effects are confounded. Our prompt variations change both structure and semantic content simultaneously. The natural-format condition differs from the rigid-format condition in token order, punctuation, sentence structure, and implicit semantic cues. Isolating purely structural effects (e.g., token order while holding meaning constant) is an important direction for future work. The effects we report are joint structural-semantic effects; we cannot attribute them to structure alone.

No attention analysis. The mode-switching interpretation is supported by activation

geometry (L2 distances, cosine similarities at layer 10) but not by direct attention pattern analysis. Confirming that the reset instruction changes attention weights across the context fence—rather than merely changing hidden state representations—would strengthen the mechanistic claim.

9 Conclusion

Prompt structure is a first-class determinant of language model behavior. Across nine experiments on Qwen 2.5-7B with 160 probes—and cross-architecture replication of the central falsification on Mistral-7B-v0.3—we find consistent, large structural effects on continuation quality.

The effective prompt configuration is simple: reset, then prime, then deliver. A short reset instruction (18 tokens) installs a processing mode boundary that reduces cross-entropy by up to 39%. A short domain-matched prime (15 tokens) captures 98.8% of the priming benefit. The total structural overhead is approximately 33 tokens.

The mechanism is mode switching, not garbage collection. The reset instruction does not erase prior context—it installs a semantic boundary that reconfigures attention. Three predictions of the garbage-collector hypothesis are falsified on Qwen, and the central prediction—that reset benefit should scale with context pollution—is independently falsified on Mistral ($\rho = 0.10$, $p = 0.87$). Mode and domain are orthogonal dimensions of the processing state, composing independently on both architectures ($\rho = 0.86$ on Qwen, $\rho = 0.92$ on Mistral for activation-distance–CE correlation).

What hurts is anything that activates a competing processing mode: meta-commentary activates instruction-following mode, repetition distorts attention distributions, forced naming activates unfamiliar token representations, and extended priming pushes past the saturation point into noise.

In sequential chains, degradation saturates logarithmically. A 10-hop chain performs the same as a 5-hop chain. Cross-domain handoffs outperform within-domain chains because the domain switch forces a natural mode reset.

These findings reduce prompt engineering from an art to an engineering problem with measurable parameters. The structure of a prompt—independent of its content—determines a significant fraction of the model's output quality. That fraction is not a rounding error. It is the difference between a useful response and an unusable one.

Data Availability

All experimental results are archived at https://huggingface.co/datasets/jmcentire/paper8-data under `paper20/`, `paper22/`, `paper23/`, and `paper24/`.

References

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

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. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.