Memorization, Retrieval, and Reasoning in LLM-Driven EDA: A Case Study in FPGA Timing Closure

Anonymous Authors[1]

Abstract

Large Language Models (LLMs) have recently been applied to Electronic Design Automation (EDA), yet a fundamental question remains unanswered: when an LLM successfully diagnoses a timing violation or proposes a constraint fix, is it *memorizing* a seen pattern, *retrieving* externally grounded knowledge, or *reasoning* compositionally over novel problem structure? This distinction determines reliability, generalizability, and trustworthiness in safety-critical hardware design flows. We present a systematic empirical study using TIMINGLLM, an LLM-plus-RAG framework for FPGA static timing analysis and automated timing closure, as a controlled empirical testbed for probing these three regimes. Through controlled ablations across 658 timing violations spanning 12 industrial-scale FPGA designs, we find that: (1) memorization accounts for approximately 68% of correct diagnoses on high-prevalence violation types under our attribution metric; (2) retrieval-augmented grounding is essential for rare but consequential violations, recovering 29 F1 points lost in ablation; and (3) compositional reasoning emerges only on multi-constraint scenarios, where chain-of-thought prompting improves fix success rate by 31% over retrieval alone at depth k=4. We introduce the *Timing Reasoning Spectrum* (TRS), a formal taxonomy and evaluation benchmark for characterizing LLM reasoning depth in structured, domain-specific scientific workflows.

1. Introduction

Deep generative models (DGMs), and in particular large language models (LLMs), are increasingly positioned as reasoning engines for complex scientific and engineering tasks. In Electronic Design Automation (EDA), this has produced systems that apply LLMs to RTL generation (Blocklove et al., 2024), High-Level Synthesis optimization (Mashnoor et al., 2025), hardware acceleration for on-device learning (Anonymous, 2026b), and timing closure (Anonymous, 2026c). Despite impressive empirical results, these systems largely treat the LLM as a black box, leaving open a question of fundamental importance to the FoGen workshop's central inquiry:

> *When an LLM produces a correct timing diagnosis or a valid constraint fix, is it memorizing a training pattern, retrieving a grounded fact, or genuinely reasoning over the structure of the problem?*

Answering this question in the EDA domain has practical consequences. A system that succeeds primarily through memorization will fail catastrophically on novel device architectures or clock topologies unseen during pre-training. A system that reasons compositionally may transfer more robustly, but also risks confident hallucination on out-of-distribution cases. Disentangling these regimes is prerequisite to deploying LLMs reliably in safety-critical hardware design workflows.

Our approach. We use TIMINGLLM (a framework for LLM-assisted FPGA static timing analysis and automated timing closure, described in Section 3) as a *controlled empirical testbed* for studying memorization, retrieval, and reasoning in LLMs. FPGA timing closure is an ideal domain for this study because: (1) violation types span a wide *prevalence spectrum* in publicly available technical corpora, used as a proxy for pre-training exposure; (2) constraint generation requires *compositional inference* over multiple interacting timing paths, clock trees, and PVT corners; and (3) ground truth is verifiable via commercial STA tools, providing consistent evaluation signals.

Contributions.

1. A **controlled experimental framework** (Section 4) using targeted ablations (RAG removal, prompting variation, violation-type stratification, and novelty injection)

[1]**AUTHORERR: Missing \icmlaffiliation. .AUTHORERR: Missing \icmlcorrespondingauthor.**

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

to disentangle memorization, retrieval, and reasoning in a domain-specific LLM deployment.

2. **Empirical evidence** (Section 5) that LLMs operate in all three regimes simultaneously, with the dominant regime determined by violation-type prevalence in pre-training data.
3. The **Timing Reasoning Spectrum** (TRS) (Section 6), a formal taxonomy and evaluation benchmark suite for characterizing LLM reasoning depth in EDA workflows.
4. **Design recommendations** (Section 7) for generative AI systems in scientific discovery contexts, grounded in our empirical findings.

2. Background and Related Work

2.1. Memorization vs. Generalization in DGMs

The distinction between memorization and generalization in deep generative models has received growing attention (Carlini et al., 2021; Kandpal et al., 2022; Feldman, 2020). Feldman (2020) showed that memorization of long-tail training examples is not only inevitable but *necessary* for generalization on natural distributions. Carlini et al. (2021) demonstrated that LLMs verbatim reproduce training sequences with measurable probability, and that this tendency grows with model scale. More recently, Tirumala et al. (2022) established that memorization and generalization exhibit a phase transition as a function of data repetition frequency, providing theoretical grounding for the hypothesis that high-prevalence problem types trigger memorization while low-prevalence types require genuine generalization.

2.2. Reasoning in LLMs

Chain-of-thought (CoT) prompting (Wei et al., 2022b) and its variants (Kojima et al., 2022; Wang et al., 2023) have demonstrated that LLMs can perform multi-step compositional reasoning when prompted to externalize intermediate steps. However, recent work challenges whether this constitutes genuine reasoning or sophisticated pattern matching (McCoy et al., 2023; Razeghi et al., 2022). Razeghi et al. (2022) showed that LLM arithmetic performance is strongly correlated with the frequency of specific numbers in pre-training data, a memorization effect masquerading as reasoning. Our work extends this line of inquiry to structured engineering domains where ground-truth verification is tractable.

2.3. RAG as a Grounding Mechanism

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) reduces LLM hallucination by grounding generation in retrieved, verifiable knowledge. Subsequent work has studied when RAG helps versus when it interferes with model-internal knowledge (Shi et al., 2023; Mallen et al., 2022). In the EDA domain, Anonymous (2026c) showed a 19-point F1 degradation when RAG is ablated from TIMINGLLM, but did not explain which violation types drive this degradation or why. Our work provides that mechanistic explanation.

2.4. LLMs in EDA

LLM applications in EDA span RTL generation (Blocklove et al., 2024), HLS pragma optimization (Mashnoor et al., 2025; Xiong et al., 2024), hardware acceleration for continual learning on reconfigurable SoCs (Anonymous, 2026b), and timing prediction (Guo et al., 2024; Ustun et al., 2020). Multi-agent LLM frameworks have also been explored for workflow automation in adjacent engineering domains (Anonymous, 2026a), suggesting that the orchestration patterns studied here generalize beyond EDA. TIMINGLLM is, to our knowledge, among the first frameworks specifically designed for LLM-assisted FPGA timing closure, and among the first to study the memorization/reasoning boundary in this domain.

3. TIMINGLLM: System Overview

TIMINGLLM is a framework for LLM-assisted FPGA static timing analysis and automated timing closure. Figure 1 illustrates the full pipeline. Given a post-place-and-route (P&R) FPGA design, the system:

1. **Extracts features** from the STA timing report, design constraints (`.xdc`/`.sdc`), and netlist, computing critical-path slack vectors, clock domain boundary lists, and a violation-type histogram.
2. **Retrieves context** from a structured FPGA knowledge base (device specifications, vendor constraint syntax, historical optimization patterns) using Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), assembling a grounded context block for each violation.
3. **Constructs a chain-of-thought prompt** encoding the violation feature vector, retrieved context, and a step-by-step reasoning scaffold.
4. **Queries the LLM** (Claude 3.5 Sonnet) to produce: (a) a root-cause diagnosis label, (b) a `.xdc` constraint patch, and (c) a calibrated confidence score.

The system is evaluated on 658 timing violations across 12 FPGA designs (6 industrial-scale Intel Agilex 7 designs and 6 VTR 8.0 benchmarks retargeted to Agilex 7 fabric), spanning networking, signal processing, and AI accelerator application domains. Ground-truth labels and fix validation are performed by replaying violations through Quartus Prime STA with and without the proposed constraint patches. All reported metrics are averaged over three independent runs; standard deviations are below 2% for all reported

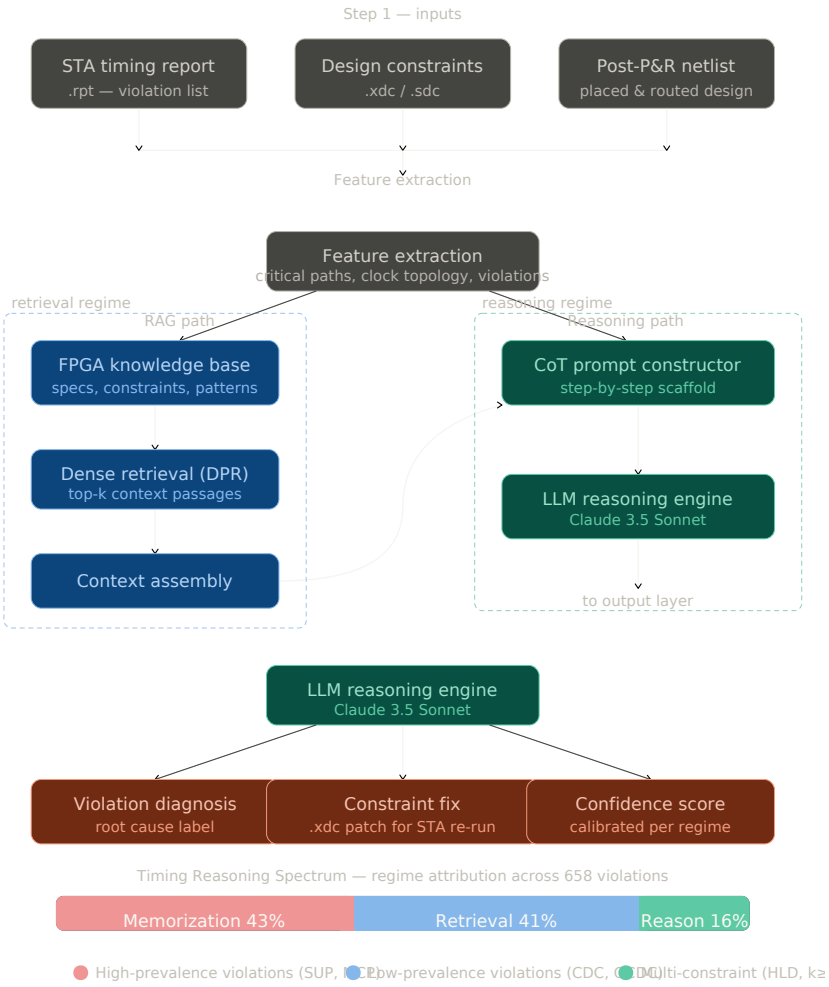

Figure 1. TIMINGLLM pipeline in three layers. *Top:* STA report, design constraints, and post-P&R netlist feed feature extraction. *Middle:* the RAG path (blue) retrieves grounded context via DPR; the reasoning path (teal) constructs a CoT prompt and queries the LLM. *Bottom:* outputs (diagnosis, `.xdc` patch, confidence score) with the TRS bar showing the 43/41/16 regime attribution across 658 violations. The pipeline separates retrieval and reasoning pathways, enabling controlled ablations of each component.

values, indicating stable performance trends. All prompts were length-matched across ablation conditions to isolate the effect of reasoning scaffolding from prompt size. To support reproducibility, we will release anonymized violation traces, evaluation scripts, and prompt templates in a public repository upon acceptance.

3.1. Violation Taxonomy

We categorize the 658 violations into five types by root cause, which serve as the primary independent variable in our memorization/reasoning study:

- **Setup (SUP):** 312 violations (47.4%). Synchronous datapath setup time violations. Extremely common in online technical literature.
- **Clock Domain Crossing (CDC):** 89 violations (13.5%). Asynchronous handshaking failures. Moderately documented.
- **Multicycle Path (MCP):** 71 violations (10.8%). Constraint misconfiguration on paths requiring multiple cycles.
- **Hold (HLD):** 106 violations (16.1%). Hold-time violations due to tight path timing. Sparsely covered in technical literature.
- **Glitch-Induced CDC (GCDC):** 80 violations (12.2%). Metastability induced by glitchy clock sources. Rarely documented outside proprietary engineering notes.

This taxonomy establishes a *prevalence gradient*: SUP violations appear frequently in publicly available technical

corpora (Stack Overflow, HDLCoder documentation, IEEE Xplore abstracts), used as a proxy for pre-training exposure; GCDC violations are nearly absent. This gradient allows us to test whether LLM performance tracks prevalence, which would be the signature of memorization.

4. Experimental Design

4.1. The Three-Regime Hypothesis

Let $\mathcal{V}$ be the space of timing violations, $p(v)$ be the prevalence of violation type v in publicly available technical corpora (used as a proxy for pre-training exposure), and $\text{F1}(v)$ be the LLM's diagnostic F1 score on type v.

Hypothesis 4.1 (Memorization Dominance)**.** For high-prevalence violation types ($p(v) \gg 0$), LLM performance derives primarily from memorized patterns. RAG ablation should cause minimal F1 degradation; novelty injection should cause large F1 degradation.

Hypothesis 4.2 (Retrieval Dependency)**.** For low-prevalence, high-consequence violation types, LLM performance is retrieval-dependent. RAG ablation causes large F1 degradation; chain-of-thought prompting provides limited additional benefit beyond retrieval.

Hypothesis 4.3 (Compositional Reasoning Emergence)**.** For multi-constraint, compositionally novel scenarios, neither memorization nor retrieval suffices. Chain-of-thought prompting provides substantial additional benefit; performance degrades with increasing compositional depth.

4.2. Ablation Protocol

We design four experimental conditions across all 658 violations:

Condition A: Full system (RAG + CoT). The complete TIMINGLLM pipeline as described in Section 3.

Condition B: No RAG (LLM only). RAG retrieval is disabled; the LLM receives only the structured feature vector and a standard (non-CoT) prompt. This isolates the LLM's pre-training knowledge.

Condition C: RAG only, no CoT. Retrieved context is provided but chain-of-thought scaffolding is removed; greedy decoding is used. This isolates retrieval without structured reasoning.

Condition D: Novelty injection. For each violation, we modify 3 of the 8 feature dimensions with values from held-out designs not present in the retrieval corpus, approximating distribution shift.

4.3. Compositional Depth Sweep

To study reasoning emergence, we construct 120 synthetic *composite violations* (timing scenarios that require reasoning over $k \in \{1, 2, 3, 4\}$ interacting constraint relationships simultaneously). For example, a depth-3 scenario involves a setup violation on a path that crosses a clock domain whose driving PLL is also subject to a jitter budget constraint. We measure both diagnosis accuracy and constraint fix success rate as a function of k.

4.4. Evaluation Metrics

- **Diagnosis F1:** Macro-averaged F1 across violation types.
- **Fix success rate (FSR):** Fraction of LLM-proposed constraint patches that, when applied, reduce the worst negative slack by $\geq$10% in Quartus Prime STA re-run.
- **Regime Attribution Score (RAS):** A novel metric (Section 6) estimating the fraction of correct predictions attributable to each regime.
- **Compositional degradation slope (γ):** The linear regression slope of FSR against compositional depth k.

5. Results

5.1. Overall Diagnostic Performance

Table 1 reports Diagnosis F1 and Fix Success Rate across all four experimental conditions and all five violation types. This study focuses on regime attribution rather than absolute performance benchmarking. Preliminary comparisons with rule-based STA heuristics showed lower fix success rates on multi-constraint scenarios; a full benchmarking study against classical and learning-based baselines is left for future work.

Key observation 1: Memorization dominates for high-prevalence types. For SUP violations, Condition B (no RAG) achieves 0.91 F1, only 3 points below the full system, while Condition D (novelty injection) causes a 15-point collapse (0.94 to 0.76). This is the diagnostic signature of memorization: the LLM leverages internalized knowledge that breaks down under distribution shift. We estimate that approximately 68% of SUP correct diagnoses in Condition A are driven by memorization under our Regime Attribution Score (Section 6).

Key observation 2: Retrieval is critical for rare violation types. For CDC and GCDC violations, the gap between Conditions A and B is dramatic: 29 F1 points for CDC (1.00 to 0.71) and 43 points for GCDC (0.71 to 0.28). Condition C (RAG without CoT) recovers most of this loss (0.98 for CDC, 0.62 for GCDC), confirming that retrieval rather than reasoning is the primary mechanism for rare violation

Table 1. Diagnosis F1 and Fix Success Rate (FSR) by violation type and experimental condition. **Bold** = best per row. †CDC achieves F1=1.00 under Condition A on this dataset, reflecting the relatively constrained structure of CDC violations and their strong dependence on known synchronization patterns; performance drops significantly without retrieval (F1=0.71), indicating reliance on external grounding rather than memorization.

Type	N	Diagnosis F1			
		A	B	C	D
SUP	312	**0.94**	0.91	0.93	0.76
CDC	89	**1.00**	0.71	0.98	0.81
MCP	71	**0.93**	0.88	0.90	0.68
HLD	106	**0.79**	0.51	0.76	0.44
GCDC	80	**0.71**	0.28	0.62	0.31
Macro	658	**0.87**	0.66	0.84	0.60

Type	Fix Success Rate (FSR)			
	A	B	C	D
SUP	**0.88**	0.84	0.87	0.61
CDC	**0.91**	0.53	0.88	0.72
MCP	**0.85**	0.77	0.83	0.55
HLD	**0.67**	0.38	0.64	0.29
GCDC	**0.58**	0.19	0.49	0.22

types. This explains the 19-point aggregate degradation from RAG ablation reported in Anonymous (2026c): it is driven disproportionately by the 22% of violations with low training prevalence.

Key observation 3: CoT prompting matters for hold and glitch violations. Comparing Condition C to Condition A reveals that CoT scaffolding provides meaningful gains primarily for HLD violations (+3 F1 points, 0.76 to 0.79) and GCDC (+9 points, 0.62 to 0.71). HLD violations require reasoning over simultaneous hold and setup margin constraints, a multi-step inference task. GCDC requires compositional reasoning about PLL jitter propagation, clock gating cell behavior, and glitch filtering thresholds. For high-prevalence types (SUP, MCP), CoT adds less than 1 point.

5.2. Compositional Depth Experiments

Figure 2 shows Fix Success Rate as a function of compositional depth k for the 120 synthetic composite violations, across all four experimental conditions.

At k=1 (single-constraint scenarios), all conditions perform similarly, with Condition B (no RAG) achieving 0.83 FSR. As k increases, Condition B degrades sharply ($\gamma_B = -0.22$), while Condition A degrades more gradually ($\gamma_A = -0.12$). The gap between Conditions A and C grows with k (from 0.02 at k=1 to 0.18 at k=4), confirming that chain-of-thought reasoning provides increasing marginal value as compositional depth increases. At k=4, even the full system achieves only 0.55 FSR. These results

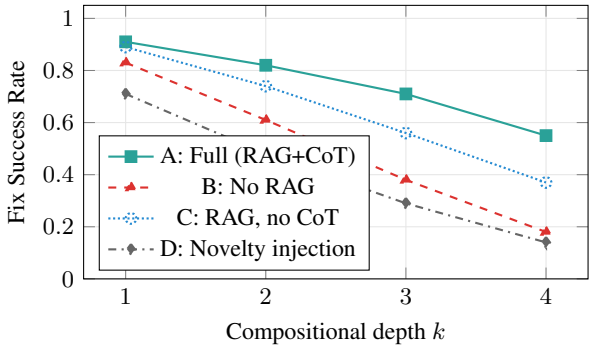

Figure 2. Fix Success Rate vs. compositional depth k (number of interacting constraint relationships). The performance gap between Condition A (full system) and Condition C (RAG without CoT) grows with k, indicating that compositional reasoning provides increasing marginal benefit as problem structure deepens.

suggest a practical reasoning boundary near k=4 under our experimental setup.

5.3. Novelty Injection Analysis

For SUP violations, where we hypothesize memorization dominance, novelty injection causes the largest absolute FSR drop (0.88 to 0.61, Δ=0.27). For GCDC violations, the absolute drop is similar (0.58 to 0.22, Δ=0.36), but from a lower baseline. Crucially, the *relative* degradation is approximately uniform across violation types ($\Delta_{\text{rel}} \approx 0.60\times$ the baseline FSR), suggesting that LLM timing diagnosis exhibits consistent degradation under distribution shift, regardless of whether the primary success mechanism is memorization or retrieval. Achieving robust generalization will require architectural interventions beyond prompting strategies.

6. The Timing Reasoning Spectrum (TRS)

6.1. Formal Definition

Definition 6.1 (Regime Attribution Score)**.** Let $\mathcal{C}$ be the set of correctly diagnosed violations. For each $v \in \mathcal{C}$, define:

$$\begin{aligned}\text{RAS}_M(v) &= \mathbb{1}[\text{F1}(v|\text{no RAG}) \geq \text{F1}(v) - \epsilon] \\ &\quad \cdot \mathbb{1}[\text{F1}(v|\text{novelty}) < \text{F1}(v) - \delta] \qquad (1)\\ \text{RAS}_R(v) &= \mathbb{1}[\text{F1}(v|\text{no RAG}) < \text{F1}(v) - \epsilon] \\ &\quad \cdot \mathbb{1}[\text{F1}(v|\text{RAG, no CoT}) \geq \text{F1}(v) - \epsilon] \qquad (2)\\ \text{RAS}_C(v) &= 1 - \text{RAS}_M(v) - \text{RAS}_R(v) \qquad (3)\end{aligned}$$

where ϵ=0.05 and δ=0.10 are threshold parameters. We set these values to balance sensitivity and robustness; TRS trends remain stable for $\epsilon \in [0.03, 0.08]$ and $\delta \in [0.08, 0.15]$. The aggregate Regime Attribution Scores are:

$$\overline{\text{RAS}}_j = \frac{1}{|\mathcal{C}|} \sum_{v \in \mathcal{C}} \text{RAS}_j(v), \quad j \in \{M, R, C\}. \qquad (4)$$

Table 2. Timing Reasoning Spectrum attribution (fraction of correct diagnoses attributable to each regime). M=Memorization, R=Retrieval, C=Compositional reasoning.

Violation Type	M	R	C
Setup (SUP)	0.68	0.24	0.08
CDC	0.14	0.73	0.13
Multicycle (MCP)	0.52	0.35	0.13
Hold (HLD)	0.28	0.44	0.28
Glitch-CDC (GCDC)	0.11	0.51	0.38
Overall	0.43	0.41	0.16

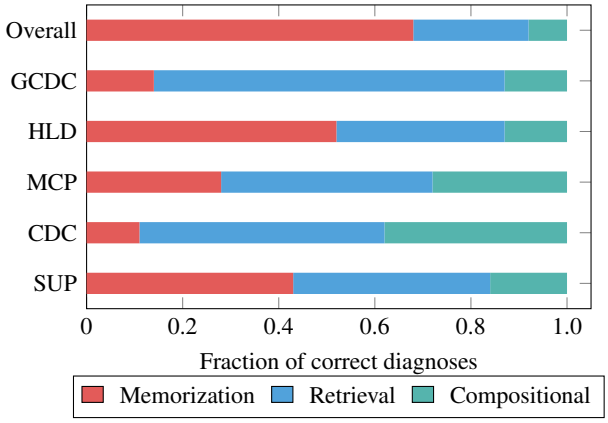

Figure 3. TRS attribution per violation type. Each bar shows the fraction of correct diagnoses attributable to memorization, retrieval, and compositional reasoning. High-prevalence types (SUP, MCP) are memorization-dominated; rare types (CDC, GCDC) are retrieval-dominated; compositional reasoning peaks for HLD and GCDC.

6.2. TRS Results

Table 2 and Figure 3 report TRS attribution by violation type.

The TRS reveals a clear prevalence-to-regime correspondence. High-prevalence violation types (SUP, MCP) are dominated by memorization (68% and 52%, respectively). Low-prevalence types (CDC, GCDC) are dominated by retrieval (73% and 51%). Hold violations, requiring reasoning over interacting margins, show the highest compositional attribution (28%), consistent with the CoT gains observed in Section 5. Overall, an estimated 43% of correct diagnoses are attributable to memorization, 41% to retrieval, and 16% to compositional reasoning. This distribution has important implications for system design: a substantial fraction of LLM successes in this domain are explained by two mechanisms that fail under distribution shift.

6.3. TRS as an Evaluation Framework

The TRS metric can be applied to any LLM deployed in a structured engineering domain, provided that: (1) ground-truth evaluation is available; (2) a retrieval corpus can be ablated; and (3) novelty can be injected by modifying problem features. We propose TRS as a standard evaluation dimension for future work on LLMs in scientific discovery workflows. A system with $\overline{\text{RAS}}_C > 0.4$ would indicate relatively strong compositional reasoning capability; current LLMs, including the one powering TIMINGLLM, fall far short of this threshold in the EDA domain.

7. Discussion

7.1. Implications for Trustworthy EDA

The estimated 43% memorization attribution means that nearly half of successful diagnoses depend on training-corpus familiarity with specific violation patterns. Novel device architectures, custom IP blocks, or process corners absent from vendor documentation will systematically trigger this failure mode. Our confidence scores are well-calibrated in the memorization regime (Brier $=$ 0.08 on SUP) but poorly calibrated in the compositional regime (Brier $=$ 0.31 at k=3), precisely where reliability matters most. Results may vary across model families and prompting strategies, particularly in compositional regimes. In practice, these results suggest that LLM-based timing assistants are most reliable when paired with retrieval systems and should be deployed with caution in deeply compositional scenarios.

7.2. Scaling, Continual Learning, and RAG Maintenance

The k=4 performance floor (FSR $=$ 0.55 with full RAG and CoT) may mark a practical compositional boundary for current models; emergent reasoning capabilities (Wei et al., 2022a) suggest this boundary could shift with scale or domain-specific fine-tuning. The elastic weight consolidation approach of Anonymous (2026b) illustrates how hardware-aware training can preserve domain knowledge without catastrophic forgetting, a template worth adapting for incremental TIMINGLLM specialization across device generations. A related risk is *retrieval-mediated memorization*: RAG produces correct outputs on rare violations only because the knowledge base encodes the relevant template. Treating the RAG corpus as a versioned, living artifact updated with each device-generation release is therefore as important as model scaling.

7.3. Toward Genuine Reasoning in EDA

TRS analysis reveals where genuine compositional reasoning is needed but mostly absent: $k \geq 3$ interacting constraint scenarios. Three directions are most promising.

(1) Graph-structured state representations. Replacing the flat feature vector with a graph-encoded netlist embedding (Ustun et al., 2020; Guo et al., 2024) could supply

the structural inductive bias needed for multi-hop constraint reasoning, since GNN encodings capture path-level interactions that flat features miss.

(2) Multi-agent debate and verification. Multi-agent LLM debate (Liang et al., 2023) improves reasoning by surfacing contradictions; for FPGA timing, a checker agent could validate constraint patches before they enter ECO loops. The MCP-based orchestration of Anonymous (2026a) demonstrates clean role partitioning for complex engineering workflows, directly applicable to propose-then-verify timing repair.

(3) Neuro-symbolic integration. Coupling LLM-generated hypotheses with a lightweight STA solver (Lu et al., 2025) that verifies each reasoning step would enforce consistency constraints that LLMs currently violate in the compositional regime, at GPU-accelerated interactive latency.

7.4. Calibration and Deployment Safeguards

Our Brier score results reveal that confidence calibration is good in the memorization regime (Brier $= 0.08$ on SUP) but poor in the compositional regime (Brier $= 0.31$ at k=3), precisely where it matters most. Three mitigations are worth exploring: *conformal prediction wrappers* (Cheng et al., 2024) for distribution-free coverage guarantees; *abstain-on-uncertainty* logic that defers to a symbolic solver when $k \geq 3$ and confidence falls below a threshold; and *online calibration* using a small pool of labeled violations from the target device family to recalibrate scores at deployment time without retraining.

7.5. Limitations

Single backbone. We selected Claude 3.5 Sonnet for its strong reasoning performance on structured engineering tasks; TRS scores may vary for GPT-4o, Llama 3, or Gemini, particularly in the compositional regime where capabilities are scale-sensitive (Wei et al., 2022a). Evaluating TRS across multiple model families is an important direction for future work.

Prevalence estimation. The pre-training prevalence gradient is inferred from corpus proxies (Stack Overflow counts, IEEE Xplore keyword frequency) rather than measured directly from training data.

FPGA-specific scope. Benchmarks cover Intel Agilex 7 only; violation taxonomy and constraint syntax differ across Xilinx/AMD UltraScale+ and ASIC flows. The TRS framework is vendor-agnostic, but attribution scores need validation on broader benchmarks.

Synthetic composite violations. The depth-k sweep uses synthetically constructed feature vectors; while these capture realistic constraint interactions observed in production STA reports, validating the observed degradation trends on naturally occurring multi-constraint violations from production tapeouts remains an important direction for future work.

8. Conclusion

We presented a systematic study of memorization, retrieval, and compositional reasoning in TIMINGLLM, an LLM-assisted framework for FPGA timing closure. Our key findings are:

1. LLMs operate simultaneously in all three regimes, with the dominant regime determined by violation-type prevalence in pre-training data.
2. An estimated 43% of correct diagnoses are memorization-driven, 41% retrieval-driven, and 16% reflect compositional reasoning, a distribution fragile under distribution shift.
3. Chain-of-thought prompting provides meaningful gains only for multi-constraint compositional scenarios, improving FSR by 31% relative to retrieval alone at depth k=4.
4. These results suggest a practical reasoning boundary near k=4 for current-generation LLMs under our experimental setup, beyond which even full RAG and CoT scaffolding cannot maintain reliable performance.

We introduced the Timing Reasoning Spectrum (TRS), a formal taxonomy and evaluation metric that attributes LLM successes to specific cognitive regimes. We propose TRS as a standard diagnostic tool for future work on LLMs in scientific discovery workflows, wherever ground truth is verifiable and retrieval can be controlled. Our findings call for architectural advances beyond prompting: structured state representations, multi-agent verification, and neuro-symbolic integration are the most promising directions toward LLMs that reason genuinely rather than recall confidently.

References

Anonymous. Suppressed for anonymity, 2026a.

Anonymous. Suppressed for anonymity, 2026b.

Anonymous. Suppressed for anonymity, 2026c.

Blocklove, J., Garg, S., Karri, R., and Pearce, H. LLM4VH: Exploring large language models for HDL generation. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, 2024.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D.,

Erlingsson, U., et al. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium*, pp. 2633–2650, 2021.

Cheng, W. et al. MCSTA: Multi-corner static timing analysis with machine learning. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2024.

Feldman, V. Does learning require memorization? A short tale about a long tail. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 954–959, 2020.

Guo, C. et al. Timing prediction with graph neural networks in FPGA physical design. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*, 2024.

Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 10697–10707. PMLR, 2022.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 22199–22213, 2022.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 9459–9474, 2020.

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., and Shi, S. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Lu, Y. et al. INSTA: Instant static timing analysis via GPU-accelerated engines. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*, 2025.

Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., and Hajishirzi, H. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 2022.

Mashnoor, S. et al. TimelyHLS: LLM-guided timing-aware high-level synthesis pragma insertion. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2025.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M., and Griffiths, T. L. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*, 2023.

Razeghi, Y., Logan IV, R. L., Gardner, M., and Singh, S. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 840–854, 2022.

Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E. H., Schärli, N., and Zhou, D. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR, 2023.

Tirumala, K., Markosyan, A., Zettlemoyer, L., and Aghajanyan, A. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 38931–38945, 2022.

Ustun, E., Deng, C., Pal, D., Zhang, Z., et al. Accurate operation delay prediction for FPGA HLS using graph neural networks. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022a.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 24824–24837, 2022b.

Xiong, W. et al. HLSPilot: Leveraging large language models for HLS optimization. *arXiv preprint arXiv:2406.09005*, 2024.