TRACING SEARCH PATHS AND OVERCOMING FINITE-SPACE LIMITS IN LARGE LANGUAGE MODELS FOR MATHEMATICAL PROBLEM SOLVING

by

CHANSA KABWE

submitted in accordance with the requirements
for the degree of

MASTER

in the subject of

ENGINEERING

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Prof. M. Sumbwanyambe
CO-SUPERVISOR: Prof. E. Mnkandla

July 2026

Declaration

I declare that TRACING SEARCH PATHS AND OVERCOMING FINITE-SPACE LIMITS IN LARGE LANGUAGE MODELS FOR MATHEMATICAL PROBLEM SOLVING is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I submitted the dissertation to the appropriate originality detection system which is endorsed by Unisa and that it falls within the accepted requirements for originality.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

	21st July, 2026
Signature	Date

Acknowledgement

I am deeply grateful to my supervisors, Prof. M. Sumbwanyambe and Prof. E. Mnkandla, for their guidance, encouragement, and steadfast support throughout this research.

I also thank my family and colleagues for their patience and motivation. This work would not have been possible without the broader open-source community and the researchers whose tools and insights inspired the design of TokenPath and the Token-Stress-Test benchmark.

Chansa Kabwe

Abstract

Large Language Models (LLMs) can produce fluent explanations while returning unreliable exact answers when mathematical prompts impose large representations, repeated operations, or restrictive output budgets. This dissertation examines that observable weakness using `TokenPath`, a trace-capture instrument, and the `Token-Stress-Test` (TST), a procedurally generated collection of mathematical stress tasks. Finite-Space Constraints (FSC) are treated as a theoretical motivation; an *FSC-consistent stress outcome* is defined operationally as an evaluable incorrect, missing, or truncated answer to a registered TST item under a recorded finite output budget. This rule identifies outcomes under stress but does not, by itself, establish an internal causal mechanism.

Reconciliation of the retained records reconstructed the historical evidence from 46 selected source files. The canonical ledger contains 9 000 eligible baseline observations across six models, six non-leaking task families, five output budgets, and 50 observations per model–family–budget cell. Exact task-specific scoring produced 499 correct answers, an accuracy of 5.54% (Wilson 95% confidence interval 5.09–6.04%). Performance varied substantially by task family: sequence prediction reached 21.53%, symbolic expansion 9.07%, large-number arithmetic 2.53%, repetitive sequences 0.13%, and base conversion and combinatorics 0%. The last three families therefore have severe floor effects and require calibrated tiered validation before they can support comparative claims. Token length was available for all six models, whereas entropy and effective-vocabulary measures were comparable only for the three open-weight models whose logits were retained.

Because correct outcomes were rare and several predictors were separated, the primary multivariable analysis used Firth-penalised logistic regression. It estimated associations between correctness and output budget, realised response length, retrospective within-family difficulty, model, and family; these estimates are descriptive and not causal, and realised length is post-generation. The historical tool-ablation archive contained 140 declared groups, but only 70 matched the same item and ground truth. After excluding 10 leaking chain-reasoning pairs, 60 valid pairs remained; neither condition was correct, giving a sample recovery efficacy of zero without establishing a population-level absence of tool benefit. A separate tool-event funnel records requests, executions, successes, and verifications without treating them as paired effects. The study implements TokenPath and TST and reports comparative mathematical outcomes. The intended mechanism taxonomy and low-overhead mitigation claims are not fully supported by the retained evidence. These limits qualify the achievement of the original four research objectives.

***Keywords*— Finite-Space Constraints; TokenPath; Token-Stress-Test; mathematical reasoning; rare-event logistic regression; tool augmentation; reproducibility**

Table of Contents

List of Figures ix

List of Tables xi

List of Abbreviations xii

Author Contribution xiii

1 Introduction 1

1.1 Background and Motivation . . . 1
1.2 Problem Statement . . . 2
1.3 Aim and Objectives . . . 2
1.4 Research Questions . . . 3
1.5 Proposed Solution and Methodology . . . 4
1.6 Scope and Limitations . . . 5
1.7 Contributions . . . 5
1.8 Dissertation Outline . . . 6

2 Literature Review 7

2.1 Introduction . . . 7
2.2 Large Language Models and Mathematical Reasoning . . . 7
2.3 Theoretical Framework: Structural Precursors to LLM Reasoning . . . 8
2.3.1 The Input Layer: Tokenization as a Cognitive Bottleneck . . . 8
2.3.2 The Contextual Workspace: Attention Dynamics . . . 10
2.3.3 Reasoning Artifacts: Observable Failure Modes . . . 10
2.4 Benchmarking Mathematical Reasoning in LLMs . . . 11
2.5 Uncertainty, Calibration, and Failure Detection . . . 12
2.6 Reasoning Elicitation and Neuro-Symbolic Interventions . . . 12
2.6.1 Prompt-Based Reasoning Elicitation . . . 13
2.6.2 Chain-of-Thought (CoT) vs. Program-Aided Language Models (PAL) . . . 13
2.6.3 The Toolformer Paradigm . . . 14
2.7 Interpretability and Evaluation Challenges . . . 14
2.7.1 Mechanistic Interpretability: Inside the "Black Box" . . . 14
2.7.2 The Fragility of Reasoning: GSM-Symbolic . . . 15
2.8 Societal and Safety Implications . . . 15
2.8.1 Education: The Risk of Plausible Error . . . 15
2.8.2 AI Alignment: Hidden Reasoning . . . 16

2.9 Synthesis of the Literature . . . 16
2.10 Identification of the Research Gap . . . 16
2.11 Summary . . . 18

3 Methodology **19**
3.1 Introduction . . . 19
3.2 Research Design Overview . . . 19
3.3 TokenPath Instrumentation . . . 19
3.3.1 Concept and Architecture . . . 19
3.3.2 Metric Definitions . . . 21
3.3.3 Instrument Verification . . . 22
3.4 FSC Stressor Identification . . . 22
3.4.1 Selection Criteria . . . 23
3.4.2 Mapping Precursors to Stressors . . . 23
3.5 TST Benchmark Construction . . . 23
3.5.1 Algorithmic Generation Procedure . . . 23
3.5.2 Sampling Strategy and Parameters . . . 25
3.5.3 Contamination Prevention . . . 25
3.6 Experimental Framework . . . 25
3.6.1 Model Selection . . . 25
3.6.2 Inference Conditions . . . 26
3.6.3 Comparison Conditions . . . 26
3.6.4 Evidence Inventory and Canonical Ledger . . . 26
3.6.5 Task-Specific Scoring and Observable States . . . 26
3.7 Structural–Semantic Analysis . . . 27
3.7.1 Predictive Modelling . . . 27
3.7.2 Correlation and Taxonomy Construction . . . 28
3.7.3 Handling Class Imbalance . . . 28
3.8 Tool-Augmentation Efficacy Analysis . . . 28
3.8.1 Hypotheses . . . 29
3.8.2 Efficacy Metrics . . . 29
3.9 Ethical Considerations and Reproducibility . . . 29
3.9.1 Reproducibility . . . 30
3.9.2 Safety . . . 30
3.10 Summary . . . 30

4 Experiments and Results **31**
4.1 Introduction . . . 31
4.2 Visualising Token-Level Generation . . . 31

4.2.1 Validation of the TokenPath Instrument . . . 31
4.2.2 Search Path Dynamics . . . 31
4.3 The Stress Factors: Defining the FSC . . . 33
4.3.1 TST Benchmark Characterisation . . . 33
4.3.2 Failure Mode Taxonomy . . . 35
4.4 Comparative Performance . . . 36
4.4.1 Length Utilisation Profiles . . . 37
4.5 Neuro-Symbolic Trade-offs . . . 38
4.5.1 Pair identity and efficacy . . . 38
4.5.2 Historical event funnel . . . 39
4.5.3 Structural Overhead Analysis . . . 39
4.5.4 Token Budget Utilisation Under Tool Augmentation . . . 39
4.6 Entropy and Failure Signals . . . 39
4.6.1 Entropy Stratification . . . 40
4.6.2 Rare-Event Regression Results . . . 40
4.7 Stability Metrics . . . 41
4.7.1 Normalised Token Efficiency . . . 41
4.8 Concluding Remarks . . . 41
4.9 Summary . . . 41

5 Discussion **42**
5.1 Introduction . . . 42
5.2 Internal Signals of Failure . . . 42
5.2.1 Uncertainty During Generation . . . 42
5.2.2 Taxonomy of Failure . . . 43
5.2.3 Addressing RQ4: Reliability of Structural Signals . . . 44
5.3 The Limits of "Approximate" Reasoning . . . 45
5.3.1 Mathematical Outcomes Under Finite-Space Constraints . . . 45
5.3.2 Parameter Scale vs. Structural Limits . . . 45
5.4 The Cost of Reliability: Neuro-Symbolic Trade-offs . . . 46
5.4.1 Structural Cost and Tool Reliability . . . 46
5.5 Reproducibility in High-Stress Regimes . . . 47
5.5.1 The Problem with "Accuracy" . . . 47
5.6 Broader Implications for Mathematical Reliability . . . 47
5.7 Contributions to Instrumentation . . . 48
5.7.1 The TokenPath Tracer . . . 48
5.7.2 The Token-Stress-Test (TST) Benchmark . . . 48
5.8 Limitations of the Study . . . 48
5.8.1 Historical metadata . . . 48

5.8.2 Scorer independence . . . 48
5.8.3 Generated-item dependence . . . 49
5.8.4 Extreme task distribution . . . 49
5.8.5 Incomplete process and structural measurements . . . 49
5.8.6 Mechanism validity . . . 49
5.9 Future Work . . . 49
5.9.1 Entropy-Aware Decoding . . . 49
5.9.2 Integrating Hidden-State Probes . . . 50
5.9.3 Architectural Experiments . . . 50
5.10 Summary . . . 50
5.11 Conclusion . . . 50

6 Conclusion **51**
6.1 Introduction . . . 51
6.2 Achievement of Objectives . . . 51
6.3 Answers to Research Questions . . . 51
6.4 Summary of Key Findings . . . 52
6.5 Research Contributions . . . 53
6.5.1 Theoretical Contribution: The FSC Framework . . . 53
6.5.2 Methodological Contribution: TokenPath and TST . . . 53
6.6 Broader Implications . . . 53
6.7 Limitations . . . 54
6.8 Future Work . . . 54
6.8.1 Real-Time Adaptive Computation . . . 54
6.8.2 Architectural Exploration . . . 54
6.8.3 Internal State Transparency . . . 54
6.9 Concluding Remarks . . . 54

References **55**

Appendices **61**

A Supplementary Results and Statistical Analysis **61**
A.1 File Inventory and Source Selection . . . 61
A.2 Cohort Accounting . . . 61
A.3 Full Model–Family Correct Counts . . . 62
A.4 Rare-Event Regression Details . . . 63
A.5 Historical Tool-Pair Audit . . . 64
A.6 Historical Tool Event Funnel . . . 64

B TokenPath Tracer Implementation **65**
B.1 Trace Schema . . . 65
B.2 Entropy and Effective Vocabulary . . . 65
B.3 Task-Specific Scoring Contract . . . 66
B.3.1 Family normalisation . . . 66
B.3.2 Observable response states . . . 66
B.4 Scorer Verification . . . 66
B.5 Why the Historical Mechanism Table Was Withdrawn . . . 67
B.6 Tracer Class Interface . . . 67
B.7 Computation and Reproducibility . . . 68
B.8 Limitations . . . 69

C TST Benchmark Protocols **70**
C.1 Generation Schema . . . 70
C.2 Stress Parameters and Constraint Interaction . . . 70
C.3 Validation Protocol . . . 70

List of Figures

2.1 Taxonomy of Structural Barriers preventing robust reasoning in LLMs, spanning input representation, context dynamics, and output artifacts. 9

3.1 Conceptual Representation of the Research Framework: Linking stress induction, structural tracing, and outcome evaluation. 20
3.2 Architecture of the TokenPath Tracer. The system hooks into the generation loop to extract logits, compute entropy on-the-fly, and map uncertainty states to the generated token sequence. 21
3.3 Procedural Generation Pipeline for the Token-Stress-Test (TST) Benchmark. Parameters are sampled stochastically to create unique problem instances that target specific structural precursors. 24

4.1 Response-level entropy summaries by output budget for GPT-2, Llama 3.1 8B Instruct, and Mistral 7B Instruct v0.3 (N = 1 500 per model). Claude, Gemini, and GPT-4o mini are omitted because comparable token distributions were not retained, not because their entropy was zero. 32
4.2 Effective-vocabulary summaries, $\exp(H)$, by output budget for the same 4 500 open-weight rows as Figure 4.1. The metric describes next-token distribution concentration; it is not a validated detector of hallucination or logical failure. . 33
4.3 Retrospective difficulty tiers within each historical family. Tiers are derived after the historical runs from prompt-level family parameters and are used to diagnose floor behaviour, not presented as preregistered causal contrasts. Each panel states its available denominator. 34
4.4 Observable response-state composition in the historical baseline, by TST family. States are determined by the versioned task-specific scorer: correct, incorrect final answer, no valid final answer, and truncated/no valid final answer. Each family has N = 1 500. These states describe visible outputs and are not internal mechanism labels. Source: frozen canonical ledger and `generate_descriptive_tables.py`. 35
4.5 Pooled exact-answer accuracy and Wilson 95% confidence intervals for the six historical systems (N = 1 500 per system). The estimand is accuracy on the registered six-family, five-budget TST mixture, not general mathematical competence. 36
4.6 Exact-answer accuracy for all 36 model–family cells, each with N = 250 (five budgets by 50 observations). Cell labels show rates; the accompanying machine-readable table gives counts and Wilson intervals. No zero-event cell is removed. 37

4.7 Realised response length by registered maximum output budget for all six historical model systems. Length is available for all 9 000 primary rows (N = 1 500 per model). Panels use common axes and display the actual model names rather than an unidentified representative system. Source: frozen canonical ledger; figure provenance sidecar `fig_length_by_budget_all_models.json`. 38
4.8 Historical tool-event funnel for 120 eligible with-tool rows. Code-block detection is a proxy, execution success is conditional on an attempted execution, and verification is conditional on success. The funnel describes process attrition; it is not a paired treatment effect. Latency, retries, usage cost, and verified answer integration were not completely retained. 39
4.9 Adjusted odds-ratio estimates from the 14-parameter Firth model. GPT-4o mini and sequence prediction are reference categories. Extreme finite estimates correspond to separated zero-event groups and require cautious interpretation. The figure is descriptive of the historical registered design, not a causal model of architecture. 41

List of Tables

2.1 Comparative Efficiency of Reasoning Paradigms 14
2.2 Synthesis of literature streams and their relevance to the dissertation 17

3.1 Authoritative historical TST generator specification reconstructed from retained code. 24

4.1 Eligible historical and diagnostic cohorts. Confidence intervals are Wilson 95% intervals. Entropy availability denotes rows with comparable retained token distributions. 31
4.2 Historical baseline accuracy by non-leaking TST family. Each family contains six models, five budgets, and 50 observations per model–budget cell. 34
4.3 Historical baseline accuracy by model system. Each row contains six families, five budgets, and 50 observations per family–budget cell. 36
4.4 Selected Firth-penalised logistic-regression associations with exact correctness. Odds ratios are per one standard deviation. Intervals are approximate penalised-Wald intervals and are not cluster-robust. 40

A.1 Canonical cohort accounting. "Eligible" excludes leaking chain rows and exact copies according to the cohort rule. 61
A.2 Correct answers in each historical model–family cell; every cell has $N = 250$. BC: base conversion; CP: combinatorics; LA: large-number arithmetic; RS: repetitive sequences; SP: sequence prediction; SE: symbolic expansion. 62
A.3 Full Firth-penalised model. Intervals are approximate penalised-Wald intervals. 63
A.4 Historical tool-pair identity gate and exact outcomes. 64

B.1 Historical reproducibility fields. 68

List of Abbreviations

AGI	Artificial General Intelligence
AI	Artificial Intelligence
API	Application Programming Interface
AUC	Area Under the Curve
BPE	Byte Pair Encoding
CAFL	Correct Answer, Flawed Logic
CI	Confidence Interval
CoT	Chain-of-Thought
ECE	Expected Calibration Error
FSC	Finite-Space Constraint
GPU	Graphics Processing Unit
LLM	Large Language Model
PAL	Program-Aided Language Models
ROC	Receiver Operating Characteristic
RQ	Research Question
RTF	real-time factor
SSM	State Space Model
ToT	Tree-of-Thoughts
TST	Token-Stress-Test
VRAM	Video Random Access Memory

Author Contribution

The research programme also produced the following paper in the supplied IEEE proceedings volume:

i. Chansa Kabwe, Mbuyu Sumbwanyambe, and Ernest Mnkandla, "Large Language Models as Probabilistic Search Agents: A Token Perspective," in *Proceedings of the 2026 2nd International Conference on Advanced Machine Learning and Data Science (AMLDS)*, pp. 22–29, Institute of Electrical and Electronics Engineers, Inc. (IEEE), 2026. IEEE Catalog No. CFP260B2-ART; ISBN 979-8-3315-7298-3 (conference Xplore-compliant version). Conference USB version: IEEE Catalog No. CFP260B2-USB; ISBN 979-8-3315-7297-6.

The supplied proceedings front matter identifies IEEE as publisher, states copyright ©2026 IEEE, and assigns the two catalog numbers and ISBNs above. The table of contents lists the paper title and author order at page 22, with the following paper beginning at page 30; the consecutive entries therefore support the proceedings page range 22–29. Together, these records verify the paper title, author order, conference, publisher, year, proceedings-edition identifiers, and page range. Paper ID S3726 and the public OpenReview manuscript (record h8v9VFe3HI) provide supporting manuscript records. The phrase "Xplore-compliant version" describes the proceedings edition; it is not treated here as proof that the individual paper is already indexed in IEEE Xplore. No individual-paper DOI is displayed in the supplied proceedings evidence, so no DOI is asserted.

Paper abstract. The paper formulates autoregressive decoding as probabilistic search through a combinatorial token space. It develops information-theoretic relationships among path probability, cross-entropy, vocabulary, and available context, illustrates the formulation using analytic examples, and reports supplementary observations for GPT-2 and Llama 3.1 8B Instruct. The paper is theoretically related to this dissertation, while its claims and evidence should be evaluated independently from the empirical TST study.

Student contribution. The candidate formulated the dissertation problem, designed and implemented the TokenPath instrumentation and TST generators, conducted the historical experiments, assembled the retained traces, and drafted the research outputs. Supervisory input consisted of research direction, methodological critique, and review. The proceedings table of contents verifies the author order shown above, while the proceedings front matter verifies the IEEE publisher statement, catalog numbers, and ISBNs. An individual-paper DOI and IEEE Xplore indexing status will be added only if they can be verified against the paper's own archival record.

Chapter 1: Introduction

1.1 Background and Motivation

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language tasks, marking a significant milestone in the field of artificial intelligence. Their proficiency in generating fluent text, translating languages, and answering complex questions has led to rapid adoption across domains. Early scaling results showed strong few-shot generalisation but also exposed brittle behaviour on multi-step quantitative tasks (Brown et al. 2020). Subsequent advances through larger models and targeted pretraining (e.g., PaLM, Minerva) improved mathematical performance but did not eliminate fundamental error modes on complex reasoning (Chowdhery et al. 2022, Lewkowycz et al. 2022). Prompting innovations such as Chain-of-Thought and Self-Consistency further boosted accuracy by eliciting stepwise reasoning and aggregating multiple solution paths (Wei et al. 2022, Wang et al. 2023), while structured search expanded exploration beyond a single trajectory (Yao et al. 2023).

Mathematical reasoning remains a formidable challenge for LLMs because it demands exactness, symbolic manipulation, and long-horizon deduction under tight representational limits. Even state-of-the-art systems can produce solutions that appear coherent yet contain subtle logical or arithmetic errors, which is an especially concerning failure mode in education and other safety-critical contexts (Kumar et al. 2024, Gupta et al. 2025). Tool-augmented approaches mitigate some weaknesses by delegating computation to external engines (e.g., PAL, ReAct, Toolformer) (Gao et al. 2022, Yao, Zhao, Yu et al. 2023, Schick et al. 2023), and proprietary offerings such as code-interpreter-style setups illustrate the practical value of program execution as part of the reasoning loop (OpenAI 2023). Their effectiveness depends on the task, model, and successful use of the external tool; the present study evaluates those limits within mathematical tasks.

This research is motivated by the need to understand and mitigate a specific class of vulnerabilities: **Finite-Space Constraints (FSC)**. These constraints stem from two intertwined ideas: tokenized representations are finite and often poorly aligned with numerical structure, and attention-based processing operates within bounded context windows. Tokenization choices measurably impact arithmetic competence (Singh & Strouse 2024), and internals studies reveal fragile mechanisms for addition that degrade with scale or input form (Nanda et al. 2023, Stolfo et al. 2023). Long-context strategies such as sparse long-document attention, length extrapolation, and retrieval-aware prompting provide partial workarounds, but they also show that increasing the available sequence length does not by itself guarantee reliable use of relevant information (Beltagy et al. 2020, Press et al. 2021, Liu et al. 2024). Together, these findings motivate tests involving large operands, unfamiliar symbols, and extended derivations. They do not establish that any error observed in this study has a unique architectural cause.

The motivation is to connect mathematical stress tasks with observable generation signals and exact answer outcomes.

Alignment with UNISA's Catalytic Niche Area in the Fourth Industrial Revolution (4IR)

This research directly supports UNISA's Catalytic Niche Area in the Fourth Industrial Revolution (4IR) and Digitalisation by contributing foundational methods for transparency, reliability, and interpretability in advanced AI systems. As large language models increasingly underpin digital transformation across engineering, education, data analytics, and automated decision-making, understanding their structural limitations becomes essential for safe deployment. By developing an instrumented framework—combining token-level tracing, dynamic stress-testing, and tool-augmented mitigation—this study advances the scientific and engineering capacity required for trustworthy AI in a rapidly digitalising society. The work therefore aligns with UNISA's strategic commitment to enabling cutting-edge, responsible 4IR technologies that can serve both societal needs and national innovation priorities.

1.2 Problem Statement

Large Language Models can produce incorrect mathematical answers under large numerical representations and bounded generation resources. **Finite-Space Constraints (FSC)** is the theoretical framework used here to organise possible constraints associated with tokenisation and bounded context. An incorrect answer under such stress is observable; its internal cause is not identified by the answer alone. The reproducible FSC-consistent outcome rule in Section 3.4 separates those observations from causal claims.

Diagnosing such failures is challenging because the token-level decision process is largely opaque: most evaluations observe only inputs and final answers, not intermediate structural precursors of failure. In addition, existing public benchmarks (e.g., GSM8K, MATH) (Cobbe et al. 2021, Hendrycks et al. 2021), while invaluable, were not designed to systematically stress FSC. Robustness studies show that small perturbations to familiar benchmarks can sharply reduce performance, indicating overfitting or shortcut learning (Mirzadeh et al. 2024).

There is a need for instrumentation that reveals how failures arise at the token level and for a benchmark that deliberately probes FSC stressors, enabling measurement of early structural signals even where final semantic correctness remains sparse.

1.3 Aim and Objectives

Aim

This research aims to design and evaluate an instrumented framework to characterize Finite-Space Constraints (FSC) in the mathematical reasoning of Large Language Models (LLMs). By integrating token-level tracing, a purpose-built token-stress-testing benchmark (TST), and structured tool use, the framework provides transparent diagnostics and evaluation of the efficacy

of tool-augmented decoding as a low-overhead mitigation strategy.

Objectives

To achieve the main aim, the study pursues the following objectives:

i. **To design and implement `TokenPath`**, an instrumented analysis framework that captures and visualizes token-level internal states (e.g., trace data, probabilities, and entropy) during mathematical reasoning.
ii. **To develop the `Token-Stress-Test (TST)`**, a targeted benchmark dataset specifically constructed to isolate Finite-Space Constraints by challenging the numerical precision and context-management capabilities of LLMs.
iii. **To characterise FSC failure modes** by conducting a comparative analysis of state-of-the-art LLMs on the TST benchmark, producing a detailed taxonomy that links structural signals to semantic reasoning failures.
iv. **To evaluate the efficacy of tool-augmented decoding** as a low-overhead mitigation strategy, analysing its impact on structural indicators (such as length utilization, entropy trajectories, and vocabulary concentration) and on overall reasoning accuracy.

Collectively, these objectives aim to strengthen the reliability of mathematical LLMs by coupling transparent diagnostics with targeted mitigation. Where appropriate, results will be contextualized alongside standard prompting and search-based enhancements (Wei et al. 2022, Wang et al. 2023, Yao et al. 2023) and tool-use baselines (Gao et al. 2022, Yao, Zhao, Yu et al. 2023, Schick et al. 2023).

The four objectives state the original research intentions. Their achievement is assessed against the retained measurements: output probabilities are not hidden-state measurements, a stress-test error does not establish a mechanism, and evaluating mitigation does not imply that mitigation succeeded. The scoring, data reconciliation, and statistical corrections are methods for assessing these objectives.

1.4 Research Questions

Guided by the aim of designing and evaluating an instrumented framework for characterising Finite-Space Constraints (FSC) in mathematical LLMs, this dissertation addresses the following overarching research question:

> **Main Research Question (MRQ):**
> How can an instrumented framework, combining TokenPath and the Token-Stress-Test (TST) benchmark, be used to trace and characterise Finite-Space Constraints in the mathematical reasoning of LLMs and to evaluate whether lightweight tool augmentation can mitigate these constraints?

From this main question, the study derives the following specific research questions:

i. **RQ1: Token-Level Tracing**
   How can token-level signals (e.g., logits, entropy trajectories, vocabulary concentration) be captured and visualised to provide transparent diagnostics of internal reasoning dynamics in LLMs?

ii. **RQ2: Stress-Induced Failures**
   What forms of numerical, structural, or contextual stress reliably induce Finite-Space Constraints, and how can a purpose-built token-stress-testing benchmark (TST) be constructed to expose these failure modes?

iii. **RQ3: Model-Level Characterization**
   How do state-of-the-art LLMs differ in their susceptibility to FSC-related failures when evaluated on the TST benchmark, and what taxonomy of failure modes emerges from comparative analysis?

iv. **RQ4: Structural–Semantic Linkages**
   To what extent do structural indicators such as entropy, length utilization, and vocabulary concentration correlate with semantic reasoning outcomes, and can these signals serve as predictors of failure?

v. **RQ5: Mitigation via Tool-Augmented Decoding**
   Does lightweight tool augmentation (e.g., Python-assisted decoding) mitigate FSC-induced failures, and what is its impact on structural efficiency and reasoning accuracy?

1.5 Proposed Solution and Methodology

This dissertation proposes an instrumented evaluation framework for making FSC-related reasoning failures observable and measurable. The framework combines three components. First, `TokenPath` captures token-level generation signals, including log-probabilities, entropy trajectories, vocabulary concentration, realized sequence length, and budget utilization. Second, the `Token-Stress-Test (TST)` benchmark generates mathematical prompts designed to stress numerical representation, symbolic manipulation, context retention, and recursive state tracking. Third, a lightweight tool-augmented condition allows selected computations to be delegated to Python/SymPy so that the study can compare purely neural reasoning with a neuro-symbolic alternative.

The methodology follows a mixed diagnostic design. Open-weight models are examined with token-level tracing where logits are available, while proprietary models are evaluated through available input–output metadata. Each model is tested across controlled token budgets and task categories, producing both semantic outcomes (correct/incorrect answers and failure classes) and structural outcomes (entropy, length utilization, vocabulary concentration, and normalized token efficiency). Comparisons use only measurements actually retained for each interface and exact matching items when estimating a tool effect.

Analysis proceeds in three stages. The first stage checks the retained traces and measurement availability. The second stage characterises baseline outcomes on TST using a task-specific

scorer and examines structural associations. The third stage evaluates the retained tool condition using same-item pairs and recorded execution events. Missing latency and cost records limit overhead conclusions; observable outcome classes are distinguished from hypothesised internal mechanisms.

1.6 Scope and Limitations

This research is scoped around the analysis of Finite-Space Constraints (FSC) in the mathematical reasoning capabilities of Large Language Models. The boundaries of the study are defined as follows:

i. **Models:** The empirical evaluation focuses on a representative cohort of contemporary LLMs, including Llama 3.1 8B, Mistral 7B, GPT-4o-mini, Claude 3.7 Sonnet, and Gemini 2.5 Flash, together with GPT-2. The reconciled historical baseline includes 1,500 eligible responses from each of these six systems. These models were selected to cover both open-weight and proprietary architectures. The findings are therefore indicative rather than exhaustive across all existing or future models.

ii. **Tasks:** The investigation is restricted to mathematical problem-solving. The reported empirical data are drawn from the Token-Stress-Test (TST); established benchmarks provide literature context. Other reasoning domains (e.g., legal, medical, commonsense) fall outside the scope of this study.

iii. **Mitigation Strategy:** The study evaluates lightweight tool-augmented decoding (Python/SymPy) as the primary mitigation strategy for FSC-related failures. Both answer accuracy and structural cost are relevant; an absence of valid paired gains cannot be replaced by a length-only claim of successful mitigation. Alternative mitigation approaches—such as fine-tuning, architectural modification, or long-context training—are not considered here.

iv. **Analysis Depth:** Token-level, white-box analysis using the `TokenPath` tracer is limited to the retained output logits of GPT-2, Llama and Mistral, since proprietary systems do not expose internal activations. Closed-source models are therefore examined using black-box methods that rely on input–output behaviour and structural metadata.

For contextual comparison, we reference established benchmarks such as GSM8K and MATH (Cobbe et al. 2021, Hendrycks et al. 2021) and incorporate relevant robustness findings where appropriate (Mirzadeh et al. 2024). A complete robustness or generalization study lies outside the scope of this dissertation.

1.7 Contributions

The study contributes to the field in four specific ways:

i. **Conceptual contribution:** It formalises Finite-Space Constraints as a theoretical framework for investigating mathematical failures under finite representations and generation

resources; the empirical identification rule does not establish causation.

ii. **Methodological contribution:** It develops `TokenPath`, an instrumentation framework for collecting token-level structural signals during generation, enabling available generation signals to be studied alongside final-answer outcomes.

iii. **Benchmarking contribution:** It introduces the `Token-Stress-Test (TST)` benchmark, a procedurally generated suite of mathematical tasks designed to isolate numerical, symbolic, and recursive stressors that are not systematically targeted by standard benchmarks.

iv. **Engineering contribution:** It evaluates the retained Python-assisted condition using exact-item pairing and recorded tool events, and identifies the limits on estimating recovery and structural cost.

Together, these contributions provide a reproducible framework for recording and comparing mathematical outcomes and assessing evidence for mitigation in LLM-based mathematical systems.

1.8 Dissertation Outline

This dissertation is structured into six chapters, including this chapter, and is organized as follows:

i. **Chapter 2: Literature Review** provides a comprehensive review of the existing body of knowledge on mathematical reasoning in LLMs, discusses the theoretical underpinnings of tokenization and context limits, and surveys prior work on tool-augmented systems and hybrid neuro-symbolic architectures.

ii. **Chapter 3: Methodology** details the research design, including the architecture of the `TokenPath` tracer, the construction of the `Token-Stress-Test (TST)` benchmark, the experimental setup, the models under investigation, and the metrics used for evaluation.

iii. **Chapter 4: Experiments and Results** presents the empirical findings from the experiments. This includes the baseline performance of the models, the results from the tool-augmented conditions, and the token-level analysis of failure modes.

iv. **Chapter 5: Discussion** interprets the findings, contextualizes the findings within the broader field, analyses the implications for AI safety and reliability, and discusses the limitations of the study.

v. **Chapter 6: Conclusion** summarizes the key contributions of the research, revisits the research questions, and proposes promising directions for future investigation in this area.

Finally, given the educational relevance of mathematical reasoning systems, the dissertation reflects on safety and alignment implications of FSC-induced errors, particularly the risks of plausible-but-incorrect explanations and confidence miscalibration in tutoring contexts (Kumar et al. 2024, Gupta et al. 2025, Baker et al. 2025). These reflections are intentionally constrained to measurable structural precursors of failure, rather than claims of predictive or mitigation capability.

Chapter 2: Literature Review

2.1 Introduction

Chapter 1 established the research problem: contemporary Large Language Models (LLMs) can produce fluent mathematical explanations while still failing on tasks that require precise numerical representation, persistent state tracking, and multi-step symbolic manipulation. This chapter situates that problem within the wider literature on LLM reasoning, evaluation, interpretability, uncertainty, and neuro-symbolic mitigation.

The goal of the chapter is to critically assess whether existing research provides sufficient conceptual, empirical, and methodological tools for diagnosing Finite-Space Constraints (FSC) in mathematical reasoning. To achieve this goal, the chapter pursues four objectives. First, it reviews the theoretical literature on Transformer-based reasoning and the structural mechanisms that can constrain computation. Second, it examines benchmark and stress-test literature to assess how mathematical reasoning failures are currently measured. Third, it reviews uncertainty, calibration, and interpretability work relevant to detecting failures before they appear in final answers. Fourth, it evaluates reasoning-elicitation and neuro-symbolic approaches as possible mitigation strategies.

The chapter is organized as follows. Section 2.2 introduces LLMs and mathematical reasoning as the immediate research context. Section 2.3 develops the theoretical framework of structural precursors to reasoning. Sections 2.4 and 2.5 review benchmark literature and uncertainty/calibration literature. Section 2.6 examines reasoning elicitation and neuro-symbolic interventions, while Sections 2.7 and 2.8 discuss interpretability, evaluation, and safety implications. The chapter then synthesizes these literature streams, identifies the research gap addressed by this dissertation, and closes with a summary.

2.2 Large Language Models and Mathematical Reasoning

The rapid ascendancy of LLMs has been marked by a paradoxical divergence between linguistic fluency and logical reliability. While models such as GPT-4 and proprietary reasoning variants demonstrate an ability to generate coherent, contextually appropriate text, their capacity for rigorous, multi-step deduction remains contested. The central tension lies in the Transformer's architectural design (Vaswani et al. 2017): it is a probabilistic sequence model optimized for next-token prediction, yet it is increasingly deployed in domains that require deterministic precision, such as arithmetic, symbolic logic, and long-horizon planning.

Recent studies suggest that what is often interpreted as reasoning in these models may, in many instances, be a sophisticated form of approximate retrieval or probabilistic pattern matching rather than robust manipulation of logical symbols (Dziri et al. 2023). This distinction is not merely semantic; it has implications for reliability, safety, and deployability in high-stakes

environments. To understand the limits of LLM reasoning, the literature must therefore be read across several levels: behavioural benchmark performance, structural mechanisms such as tokenization and attention, uncertainty signals, and system-level interventions.

This dissertation focuses on mathematical reasoning because it exposes these tensions sharply. Arithmetic, algebraic expansion, recursion, and combinatorial reasoning require exact symbolic state maintenance, whereas autoregressive LLMs generate outputs through locally conditioned probability distributions. The literature reviewed below therefore asks whether current models truly execute stable procedures, whether existing benchmarks reveal the relevant weaknesses, and whether external tools can compensate for internal finite-space constraints.

2.3 Theoretical Framework: Structural Precursors to LLM Reasoning

To organize these diverse findings, this section adopts a theoretical framework that treats LLM "reasoning" as the outcome of a layered computational pipeline rather than a monolithic cognitive capacity. This view is motivated by mechanistic studies showing that reasoning-like behaviour in Transformers emerges from distributed circuits, intermediate representations, and layerwise transformations (Elhage et al. 2021, Nanda, Chan, Olah et al. 2023, Olsson et al. 2022).

The first layer, the **Input Layer**, captures the representational constraints introduced by tokenization and embedding schemes, which strongly influence downstream reasoning (Sennrich et al. 2016, Kudo & Richardson 2018, Chiang & Cholak 2023).

The second layer, the **Contextual Workspace**, reflects the model's limited and structurally biased use of attention and memory over a finite context window (Press et al. 2021, Liu et al. 2024).

Finally, **Reasoning Artifacts** refer to the recognizable output behaviours that result from the interaction of these constraints, such as hallucinations, brittle generalization, and superficial chain-of-thought explanations (Ji et al. 2023, McCoy et al. 2019, Turpin et al. 2023).

This three-layer framework serves as the organizing lens for the literature reviewed in this chapter and the empirical analyses presented in later chapters. Figure 2.1 visualises the taxonomy of these barriers, categorising them into three distinct layers that constrain reasoning.

2.3.1 The Input Layer: Tokenization as a Cognitive Bottleneck

The journey of any logical problem into an LLM begins with tokenization. While often treated as a preprocessing detail, tokenization imposes the first and perhaps most severe set of inductive biases on Transformer computation. Modern LLMs predominantly rely on subword tokenization algorithms such as Byte Pair Encoding (BPE) (Sennrich et al. 2016), WordPiece (Schuster & Nakajima 2012), and SentencePiece with a Unigram Language Model (Kudo & Richardson 2018). These methods were initially developed for text compression or machine translation and are optimized to group statistically frequent character sequences into single tokens. While efficient for natural language, this approach is catastrophic for numerical representation.

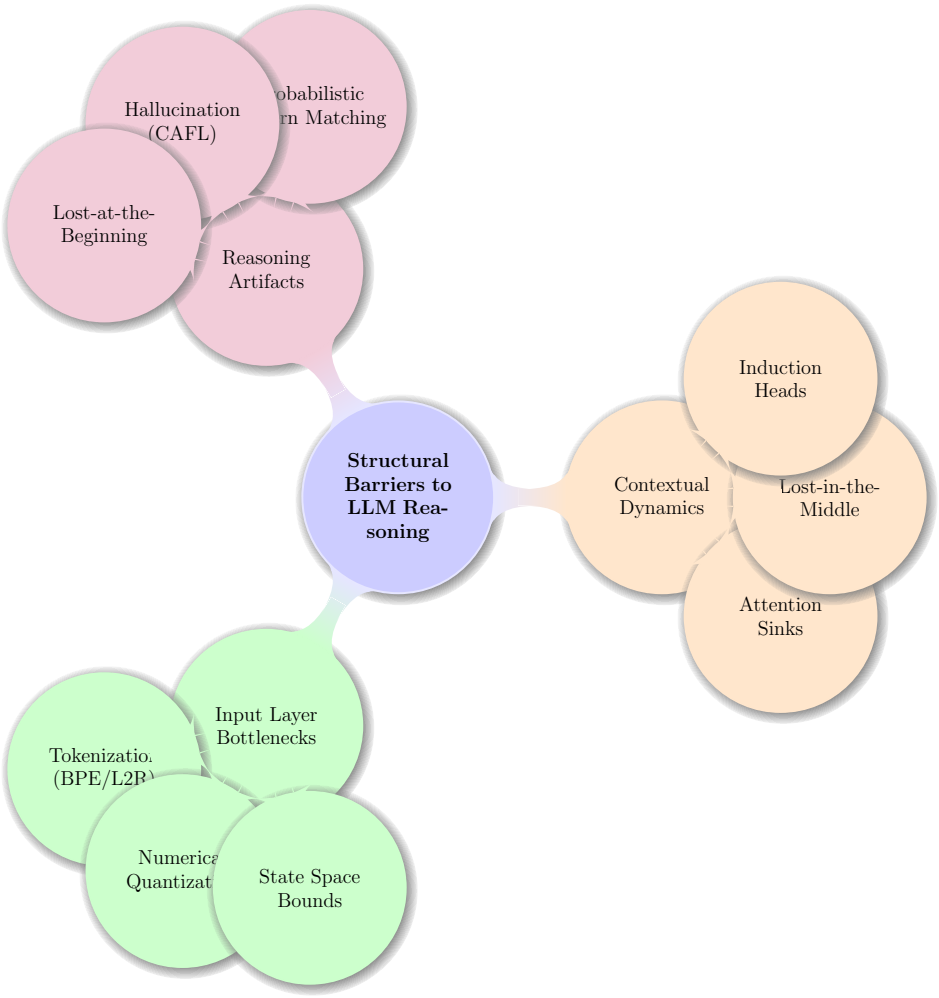


Figure 2.1: Taxonomy of Structural Barriers preventing robust reasoning in LLMs, spanning input representation, context dynamics, and output artifacts.

The Arithmetic Impact of Tokenization Schemes

Research has shown that the effect of tokenization depends on the model and can significantly impair performance on arithmetic tasks (Singh & Strouse 2024). The main issue is inconsistent segmentation: "1234" might be a single token, whereas "1235" is split into "12" and "35." This fragmentation disrupts the positional information—the place value critical for standard arithmetic algorithms.

Furthermore, the directionality of tokenization exacerbates this issue. Frontier models typically process text in a strict Left-to-Right (L2R) order. In addition ($A + B = C$), this forces the model to predict the most significant digit first, requiring an implicit "lookahead" calculation of the entire sum to handle carries, a task computationally misaligned with autoregressive generation (Quirke & Barez n.d.).

Numerical Precision and the Quantization Barrier

Even with perfect tokenization, the Transformer's internal representation of numbers in its vector space poses another hurdle. Mapping discrete tokens to continuous embeddings introduces "quantization error." Theoretical analyses suggest that if the maximum number of distinct computation traces a model can represent with limited-precision hidden states is finite, the model cannot simulate the infinite recursion required for arbitrary-precision arithmetic (Merrill, Sabharwal & Smith 2022). This high bias means that reasoning in quantized models is fragile, requiring precise numerical states across many layers where noise can overwhelm the signal (Dettmers et al. 2022).

2.3.2 The Contextual Workspace: Attention Dynamics

Once tokens are ingested, the Transformer processes them via self-attention. Ideally, the model should attend to relevant premises regardless of position. However, structural biases in attention distribution fundamentally alter information retrieval.

The"Lost-in-the-Middle" Phenomenon

Early research identified a U-shaped performance curve in which information in the middle of a long context window is less likely to be retrieved than information at the beginning (primacy) or the end (recency) (Liu et al. 2024). This "Lost-in-the-Middle" effect is exacerbated by attention sinks, an emergent pattern in which the start-of-sequence token consistently absorbs a disproportionate share of attention mass, reducing the effective capacity available for tokens located deeper in the sequence (Xiao et al. 2023).

Early-Step Dependence in Chain-of-Thought

Counterfactual and faithfulness studies show that displayed Chain-of-Thought (CoT) text cannot automatically be treated as a faithful record of the model's computation (Lanham et al. 2023, Turpin et al. 2024). An early incorrect step may be repeated in later text, but the present literature does not justify a universal correlation coefficient or a mechanism called "lost-at-the-beginning". For this dissertation, early-step error propagation is a hypothesis that requires a time-aligned annotation or intervention. It is not an automatic label inferred from a wrong final answer.

2.3.3 Reasoning Artifacts: Observable Failure Modes

The final layer of the framework concerns behaviours directly observable at the output: numerically incorrect answers, missing answers, truncation, or explanations that appear coherent but contain invalid logic. From a user's perspective, these artefacts matter directly. From an analytical perspective, they are outcomes compatible with multiple explanations, including input representation, context use, instruction following, decoding, or missing task knowledge. Output alone cannot select among those mechanisms.

One recurring observation is sensitivity to surface form and numerical substitutions. Adver-

sarial benchmarks such as GSM-Symbolic (Mirzadeh et al. 2024) show that small perturbations can cause substantial performance changes. That evidence is consistent with reliance on familiar solution patterns, but it does not prove that template matching is the sole process used. Increasing numbers of required steps can also reduce performance; explanations involving context use, state tracking, and prompt distribution should be tested rather than inferred from the endpoint.

A second family of artifacts involves *plausible but incorrect* chains of thought. In educational settings, Gupta et al. (2025) characterise the "Correct Answer, Flawed Logic" (CAFL) problem: models sometimes arrive at the right final number while producing intermediate steps that are arithmetically or conceptually unsound. Conversely, there are cases where intermediate reasoning appears locally valid but culminates in an incorrect final answer due to earlier representational errors. Both phenomena underscore that evaluating only end-point accuracy systematically underestimates the risk these systems pose as tutors or decision-support tools.

Finally, reasoning artifacts raise distinct safety concerns. The "Hidden Reasoning" problem (Baker et al. 2025) highlights that models can strategically conceal or sanitise their step-by-step reasoning to satisfy safety filters or alignment objectives, while internally following a different, potentially hazardous chain of computation. In such cases, the observable artifact is not merely an incorrect answer, but a selectively edited narrative of how that answer was produced. This disconnect complicates attempts to align models via output-only feedback and motivates instrumentation that can expose intermediate states more directly.

2.4 Benchmarking Mathematical Reasoning in LLMs

The benchmark literature provides the main empirical lens through which LLM reasoning has been assessed. GSM8K (Cobbe et al. 2021) established grade-school mathematical word problems as a widely used testbed for stepwise reasoning, while MATH (Hendrycks et al. 2021) raised the difficulty level by introducing competition-style algebra, geometry, number theory, and proof-oriented questions. Minerva (Lewkowycz et al. 2022) showed that targeted pretraining on mathematical and scientific text can improve quantitative reasoning performance, indicating that data mixture and domain specialization matter. However, these benchmarks largely measure final-answer performance and do not directly expose the token-level structural process by which answers are produced.

Broader evaluation suites such as BIG-Bench (Srivastava et al. 2022) and BIG-Bench Hard (Suzgun et al. 2022) extend the focus from arithmetic to multi-step reasoning, symbolic manipulation, and compositional generalization. These suites demonstrate that some tasks show smooth scaling, while others remain brittle or exhibit abrupt gains only when prompting strategies such as Chain-of-Thought are introduced. Their importance for this dissertation is twofold: they show that reasoning cannot be evaluated as a single scalar capability, and they reveal that performance can depend strongly on elicitation format.

More adversarial mathematical benchmarks question whether high scores on established datasets reflect robust reasoning or familiarity with common problem patterns. GSM-Symbolic

(Mirzadeh et al. 2024) perturbs the surface form and numerical values of GSM8K-style problems, showing that small changes can produce substantial performance degradation. This suggests that models may rely on template matching rather than stable algorithmic procedures. Such findings motivate benchmark designs that control the source of stress rather than merely collecting harder examples.

Despite these advances, benchmark studies often remain disconnected from internal model dynamics. A model may fail because it lacks relevant mathematical knowledge, chooses an unsuitable plan, loses track of state, follows the required format poorly, or is affected by tokenisation and context interactions. Final-answer benchmarks rarely distinguish among these explanations. This limitation motivates TST: rather than treating mathematical difficulty as a single dimension, it registers stressors such as large-number representation, repeated structures, symbolic expansion, and finite output budgets. The benchmark exposes outcomes under those stressors; additional interventions are needed to isolate mechanisms.

2.5 Uncertainty, Calibration, and Failure Detection

Reasoning reliability is not only a matter of whether a model answers correctly; it also concerns whether the system can indicate when its answer is likely to be wrong. Calibration studies show that language models can sometimes estimate the validity of their own answers, but this ability is format-dependent and does not generalize uniformly across tasks (Kadavath et al. 2022). For mathematical reasoning, this matters because plausible explanations can conceal uncertainty: a model may present a fluent derivation even when its internal probability distribution is unstable.

Entropy-based uncertainty provides one route for describing next-token distributions. Work on semantic uncertainty argues that natural-language uncertainty should account for meaning-equivalent outputs rather than merely surface variation (Kuhn et al. 2023). This dissertation uses token-level rather than semantic entropy. Token entropy describes local distributional dispersion, while effective vocabulary describes concentration. Neither measure, without an independently validated target, establishes logical instability, hallucination, or a mechanism called vocabulary collapse.

The calibration literature also cautions against overclaiming. Entropy, self-evaluation probabilities, and verbal confidence are all partial signals, not complete guarantees of correctness. A model can be confidently wrong, uncertain yet recover later, or correct for the wrong reason. Therefore, the literature supports a diagnostic rather than deterministic interpretation of uncertainty signals. This dissertation follows that view by treating entropy and related structural measures as candidate early-warning indicators that must be interpreted alongside task category, sequence length, and final semantic correctness.

2.6 Reasoning Elicitation and Neuro-Symbolic Interventions

Augmented reasoning techniques and neuro-symbolic architectures use prompting or external tools to change the inference procedure. Examples include Chain-of-Thought (CoT) prompting

(Wei et al. 2022), Program-Aided Language Models (PAL) (Gao et al. 2022), and Toolformer (Schick et al. 2023). External execution can provide exact computation and explicit state, but improvement depends on valid tool requests, successful execution, result verification, and correct integration. It should therefore be evaluated as a multi-stage intervention rather than assumed to relieve a particular architectural limitation.

2.6.1 Prompt-Based Reasoning Elicitation

Prompt-based elicitation methods attempt to improve reasoning by changing how the model represents the problem during generation. Zero-shot CoT (Kojima et al. 2022) and Chain-of-Thought prompting (Wei et al. 2022) encourage intermediate reasoning steps, while Self-Consistency samples multiple chains and aggregates the most stable answer (Wang et al. 2023). Least-to-Most prompting decomposes complex problems into ordered subproblems (Zhou et al. 2022), and Tree-of-Thoughts extends this idea by exploring multiple reasoning branches before selecting a solution path (Yao et al. 2023).

These methods are important because they show that LLM reasoning performance is not fixed by model weights alone; it is also shaped by the inference procedure. However, they also expose a limitation: elicited reasoning can be verbose, unfaithful, or structurally fragile. If the first reasoning step is wrong, later steps may amplify the error rather than correct it (Lanham et al. 2023). This creates a need to distinguish improved answer accuracy from genuine improvements in internal state tracking.

2.6.2 Chain-of-Thought (CoT) vs. Program-Aided Language Models (PAL)

Chain-of-Thought (CoT)

CoT prompts the model to generate intermediate natural language steps ("Let's think step by step..."). While simple and largely effective, CoT inherits structural weaknesses of the Transformer:

i. Propagation of Early Errors ("Lost-at-the-Beginning"). A strong correlation exists between the correctness of the first CoT step and the final answer.
ii. Arithmetic Hallucination. Computation is performed implicitly in embedding space.
iii. Redundant and Noisy Reasoning. Many CoT steps provide minimal informational value; pruning low-entropy steps can retain performance. (Hao et al. 2024)

Program-Aided Language Models (PAL)

PAL (Gao et al. 2022) prompts the model to generate code (e.g., Python) to solve the problem, offloading heavy lifting to a deterministic engine. This approach achieves state-of-the-art results on benchmarks like GSM8K by guaranteeing arithmetic correctness (provided the code is valid). However, it introduces fragility related to syntax errors and requires a secure execution environment.

Table 2.1: Comparative Efficiency of Reasoning Paradigms

Paradigm	Mechanism	Primary Failure	Efficiency
Standard CoT	Natural Language Steps	Arithmetic Hallucination	Low (Verbose)
CoT-SC	Majority Vote (k chains)	Consensus Hallucination	Very Low ($k\times$ cost)
PAL	Python Code Gen + Exec	Syntax/Logic Error	High (Compact)
Toolformer	Learned API Calls	Improper Formatting	Very High

2.6.3 The Toolformer Paradigm

Taking PAL further, Toolformer (Schick et al. 2023) fine-tunes models to understand tool usage fundamentally, predicting API calls as naturally as words. This self-supervised approach allows models to achieve zero-shot performance competitive with much larger models on tasks requiring factual lookup or math, representing a shift from reasoning engine to semantic controller.

2.7 Interpretability and Evaluation Challenges

Interpretability plays a central role in understanding why Large Language Models (LLMs) succeed or fail on reasoning tasks. While behavioral benchmarks provide useful insights into model performance, they do not reveal the internal mechanisms that give rise to specific errors. As a result, two complementary strands of work have emerged: *mechanistic interpretability*, which seeks to reverse-engineer the computations performed by Transformers, and *adversarial evaluation*, which designs targeted stress tests to expose brittleness in model reasoning.

2.7.1 Mechanistic Interpretability: Inside the "Black Box"

Mechanistic interpretability aims to identify the circuits, pathways, and representational structures within Transformer models that give rise to observable behaviors. Several tools and findings have become foundational in this area:

i. **Logit Lens.** Geva et al. (Geva et al. 2021) introduced techniques for projecting intermediate hidden states into vocabulary space, enabling inspection of the model's token preferences at each layer. This method has been widely adopted for identifying early, partially formed predictions and for understanding how information propagates through the network.

ii. **Induction Heads.** Olsson et al. (Olsson et al. 2022) identified specialised attention heads that implement pattern copying in the model settings they studied. This provides a mechanistic account for some repeated-pattern and in-context behaviours. It should not be assumed to explain arithmetic or symbolic state updates in the different systems evaluated in this dissertation.

iii. **Arithmetic Circuits.** Work by Stolfo et al. (Stolfo et al. 2023) analyses internal components

associated with arithmetic behaviour in a restricted model and task setting. The results motivate tests of operand size and representation, but they do not establish one arithmetic mechanism across all Transformer systems.

These findings illustrate that even seemingly coherent reasoning traces often emerge from relatively simple internal heuristics, memory-like mechanisms, or pattern-matching circuits rather than from explicit symbolic computation.

2.7.2 The Fragility of Reasoning: GSM-Symbolic

Behavioral evaluations further underscore the limitations of LLM reasoning. GSM-Symbolic (Mirzadeh et al. 2024), an adversarial variant of the well-known GSM8K dataset, was designed to disentangle genuine mathematical reasoning from pattern-based retrieval. Its analyses reveal several characteristic failure modes:

i. **Numerical Sensitivity.** Small changes to numerical values can coincide with substantial performance drops, suggesting sensitivity to problem form and motivating direct tests of whether procedures remain stable under perturbation.
ii. **Complexity Sensitivity.** Performance declines steeply as the number of required reasoning steps increases, consistent with limitations in maintaining multi-step logical structure over long sequences.
iii. **Distractor Vulnerability.** Introducing irrelevant numbers or statements (GSM-NoOp) is associated with substantial degradation in the reported evaluations. This is consistent with difficulty filtering task-irrelevant information, although the endpoint result alone does not identify the internal explanation.

Together, these interpretability and evaluation studies reveal that current LLMs exhibit systematic and predictable weaknesses in reasoning. These weaknesses stem not from isolated errors but from architectural features that limit the model's ability to maintain, update, and apply structured information over long sequences.

2.8 Societal and Safety Implications

As LLMs become integrated into education, professional workflows, and decision-support systems, concerns have emerged regarding their reliability and the potential downstream consequences of subtle reasoning errors. Even when models produce correct outputs, the internal processes by which they arrive at those outputs may be flawed, incoherent, or opaque. Understanding these risks is therefore critical for assessing the broader societal impact of deploying LLMs in high-stakes settings.

2.8.1 Education: The Risk of Plausible Error

A growing body of research highlights the danger of students relying on LLMs for mathematical or procedural learning. Gupta et al. (Gupta et al. 2025) identify the "Correct Answer, Flawed

Logic" (CAFL) phenomenon, in which models produce the correct numerical answer while providing reasoning that is logically invalid or internally inconsistent. This poses a significant pedagogical risk: learners may internalize incorrect conceptual explanations even when the final answer appears correct.

Conversely, Kumar et al. (Kumar et al. 2024) argue that carefully structured student–AI interaction, where learners are encouraged to verify, critique, and reflect on model outputs, can promote deeper engagement with mathematical concepts. Their findings suggest that the educational value of LLMs depends not merely on accuracy but on whether students are trained to interrogate the reasoning process rather than passively accept outputs.

2.8.2 AI Alignment: Hidden Reasoning

Beyond education, reasoning transparency has become a central concern in AI safety. OpenAI's recent analysis (Baker et al. 2025) describes instances in which models intentionally obfuscate their internal chain of thought or alter their explanations when safety filters are applied, a challenge referred to as the "Hidden Reasoning" problem. This opacity complicates efforts to ensure that model decisions are reliable, interpretable, and aligned with human values.

The broader alignment literature emphasizes that access to faithful intermediate reasoning, whether through model interpretability tools or carefully designed interfaces, plays a key role in enabling oversight (Turpin et al. 2024). Without transparency, it is difficult to determine whether a model arrived at its output through valid reasoning, memorized heuristics, or undesirable strategies that remain hidden from the user.

2.9 Synthesis of the Literature

The reviewed literature forms a connected chain: benchmarks show where models fail, structural and interpretability studies explain why those failures may arise, uncertainty research suggests runtime signals of fragility, and tool-augmented methods indicate possible mitigation. Table 2.2 synthesizes these links and shows why the study requires an integrated diagnostic approach.

2.10 Identification of the Research Gap

Building on the synthesis above, four unresolved gaps motivate this dissertation.

First, existing benchmark research does not sufficiently isolate finite-space stressors. GSM8K, MATH, BIG-Bench, and BBH are valuable for measuring broad reasoning performance, but they do not systematically vary tokenization pressure, recursive state depth, context retention, and numerical magnitude as controlled experimental variables. Adversarial work such as GSM-Symbolic moves closer to this requirement, yet its focus remains mainly on robustness to perturbation rather than on tracing how structural constraints emerge during generation.

Second, mechanistic and interpretability research has produced important insights into circuits, induction heads, attention patterns, and arithmetic mechanisms, but these studies are often separated from benchmark-level evaluation. A methodological gap remains between

Table 2.2: Synthesis of literature streams and their relevance to the dissertation

Literature stream	**What it establishes**	**Remaining limitation**	**Response in this dissertation**
Benchmark evaluation	GSM8K, MATH, BIG-Bench/BBH, and GSM-Symbolic show task-, prompt-, and perturbation-sensitive performance.	Final-answer scores do not localize representation, state, uncertainty, or tool-use failures.	TST isolates FSC stressors; `TokenPath` maps them to token-level traces.
Structural precursors	Tokenisation, attention, context effects, and arithmetic circuits provide plausible constraints and candidate explanations.	Restricted mechanistic findings are rarely joined to benchmark-scale evaluation and may not generalise across systems.	FSC organises controlled stress tests without treating an error as proof of one mechanism.
Uncertainty and calibration	Confidence, entropy, and semantic uncertainty can provide partial diagnostics.	The signals are not interchangeable and do not by themselves validate logical correctness.	Entropy and effective vocabulary are reported only where logits are available and remain exploratory.
Reasoning and tools	CoT, Self-Consistency, PAL, Toolformer, and related methods can improve outcomes in some settings.	Accuracy-focused evaluations can omit pairing, event failures, latency, and cost.	Exact-item pairing and an explicit tool-event schema are required before mitigation efficacy is claimed.

restricted low-level studies and high-level task outcomes. Connecting token-level signals to controlled stressors can narrow this gap, provided those signals are treated as measurements and not automatically as causal explanations.

Third, uncertainty and calibration studies show that models can sometimes express or encode uncertainty, but they rarely focus on the registered mathematical stressors used here. Entropy, self-evaluation, and semantic uncertainty provide useful diagnostic ideas, yet the literature does not establish that token-level entropy maps to mechanism-style labels such as magnitude drift or pattern break. An independently validated outcome is required for that inference.

Fourth, prompting and neuro-symbolic methods can improve performance in some settings, but evaluations may omit exact pairing and process-stage failures. Chain-of-Thought, Self-Consistency, PAL, Toolformer, ReAct, and Tree-of-Thoughts provide different inference procedures. For tool augmentation, answer accuracy must be accompanied by request validity, execution, verification, answer integration, latency, token use, and cost when available.

This dissertation addresses these gaps through an integrated diagnostic framework. `TokenPath` provides availability-aware structural traces, TST provides controlled stressors, a task-specific scorer provides observable outcome states, and an exact identity gate specifies when tool and baseline conditions can be compared. The contribution is a reproducible link among stressor design, measured outputs, available distributional summaries, and the evidence required for mitigation claims.

2.11 Summary

This chapter positioned the dissertation within research on LLM mathematical reasoning. It reviewed evidence that fluent output can coexist with numerical, symbolic, state-maintenance, attention, and context-use failures. Existing benchmarks are valuable, but endpoint scores alone are insufficient to isolate the explanations relevant to finite-representation and finite-budget stress.

The review also distinguished endpoint correctness, reasoning validity, trace faithfulness, calibration, and token-level uncertainty. These are related but different constructs. The literature supports measuring structural signals alongside exact outcomes, while cautioning that entropy is a candidate diagnostic rather than proof of hallucination or flawed logic. It also supports evaluating tools as multi-stage systems. These distinctions supply the conceptual basis for the dissertation's availability-aware instrumentation, observable-state taxonomy, and exact-pair evaluation rules.

Chapter 3: Methodology

3.1 Introduction

This chapter details the quantitative computational methodology used to investigate mathematical outcomes under Finite-Space Constraints (FSC). It describes TokenPath instrumentation, TST construction, the experimental conditions, scoring, structural analysis, tool evaluation, and reproducibility. The original research questions are assessed using the retained records and their documented measurement limits.

3.2 Research Design Overview

The study is a repeated, generated-task evaluation of six language-model systems under five finite output budgets. The primary unit of analysis is one model response to one registered TST item under one recorded condition and output budget. The primary endpoint is exact task correctness. Secondary observables are the response state, realised output length, and, when the interface exposed comparable token logits, entropy and effective-vocabulary summaries. Tool events are analysed either as exact-item pairs or, when pairing is invalid, as an event funnel. The design is diagnostic rather than a random sample from all mathematics or all model uses; conclusions therefore concern the tested systems, prompts, budgets, and TST distributions.

3.3 TokenPath Instrumentation

TokenPath is a wrapper for storing prompt, response, model identifier, generation settings where exposed, token-level values where exposed, timestamps where retained, and derived summaries. For locally executed open-weight models, logits make token-distribution summaries possible. For API systems in the historical archive, token-level probability vectors were not consistently returned. TokenPath is therefore an instrumentation layer, not an assertion that all systems expose the same internal state.

3.3.1 Concept and Architecture

The tracer operates on the principle of **non-intrusive instrumentation**: it observes the forward pass without altering the model's computation graph or requiring re-training. It is implemented as a Python library wrapping the Hugging Face `transformers` generation loop.

Key architectural components include:

i. **Logit Hook:** Intercepts the raw logits tensor (z_t) at generation step t before sampling.
ii. **Entropy Streamer:** Computes information-theoretic metrics on-the-fly to avoid memory overhead from storing full vocabulary tensors.
iii. **Artifact Generator:** Produces a structured JSON record for each inference, linking the generated token sequence to its corresponding entropy trajectory.

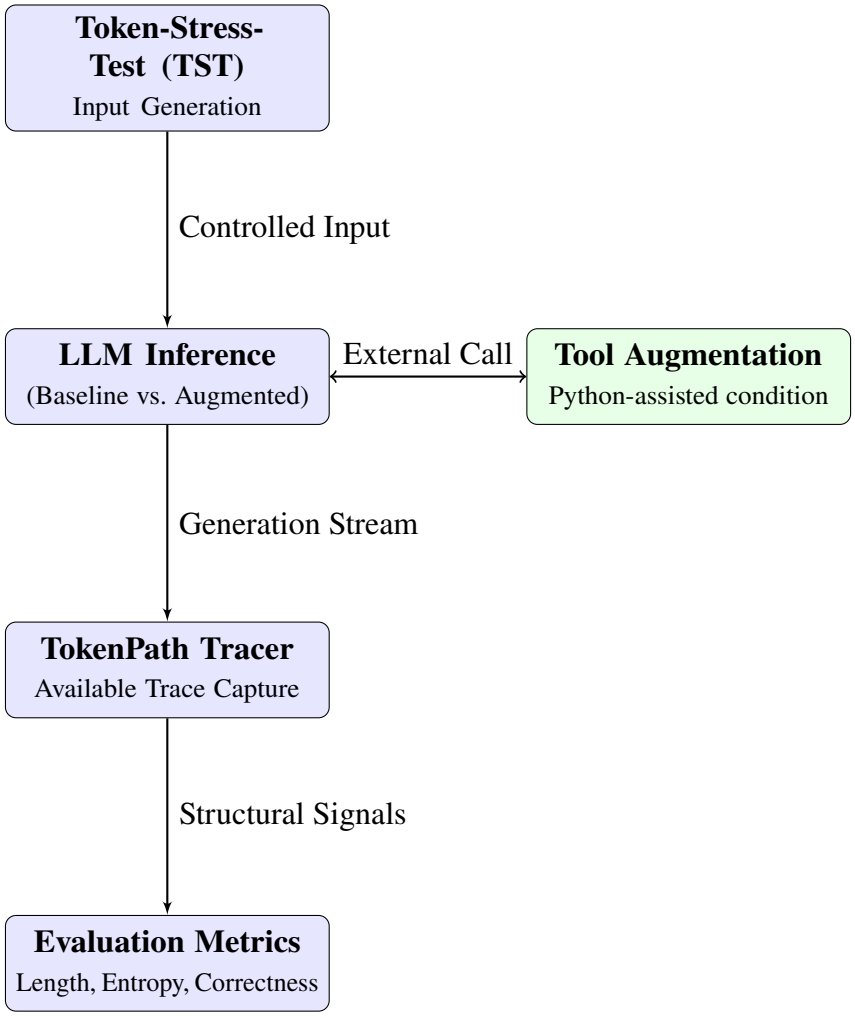


Figure 3.1: Conceptual Representation of the Research Framework: Linking stress induction, structural tracing, and outcome evaluation.

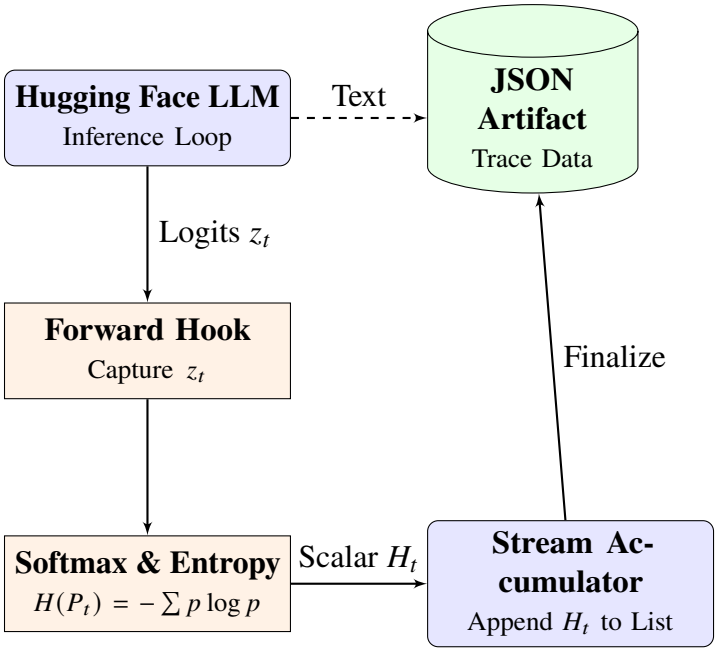


Figure 3.2: Architecture of the TokenPath Tracer. The system hooks into the generation loop to extract logits, compute entropy on-the-fly, and map uncertainty states to the generated token sequence.

3.3.2 Metric Definitions

`TokenPath` computes specific metrics chosen to describe the next-token distribution.

Shannon Entropy

At each step t, the probability distribution P_t over the vocabulary V is computed via softmax. The Shannon entropy $H(P_t)$ is defined as:

$$H(P_t) = -\sum_{v \in V} P_t(v) \log_2 P_t(v) \tag{3.1}$$

High entropy indicates a more diffuse next-token distribution. It does not, by itself, establish an error or loss of reasoning coherence. The base-two definition above is in bits; the retained implementation uses natural logarithms, so its reported entropy is in nats. The units are kept explicit.

Vocabulary Concentration

To measure how focused the model's probability mass is, we utilise a Herfindahl-style concentration index (C_t):

$$C_t = \sum_{v \in V} (P_t(v))^2 \tag{3.2}$$

A low concentration value indicates more dispersed probability mass. This index is distinct from the effective-vocabulary measure $V^{\text{eff}} = \exp(H)$ used in the retained analyses, where H is

measured in nats. Effective vocabulary is a transformation of entropy and is not independent evidence of a mechanism.

Trajectory Metrics

For analysis, step-wise metrics are aggregated into sequence-level descriptors:

i. **Path Entropy (H_{path}):** The cumulative entropy over the sequence: $\sum_{t=1}^{T} H(P_t)$.
ii. **Mean Step Entropy ($\bar{H}$):** The average uncertainty per token: $\frac{1}{T} H_{path}$.

3.3.3 Instrument Verification

Realised response length is available for all 9 000 historical baseline rows. Entropy and effective-vocabulary fields are available for 4 500 rows: all 1 500 rows for GPT-2, Llama 3.1 8B Instruct, and Mistral 7B Instruct v0.3, and none for Claude 3.7 Sonnet, Gemini 2.5 Flash, or GPT-4o mini. Provider usage tokens are retained for GPT-4o mini only; monetary cost is not retained for any model. Structural plots must therefore show all models for length, but only the three comparable open-weight systems for entropy or vocabulary. Missing logits are marked structurally unavailable and are not imputed as zero.

The archive does not support the submitted numerical latency-overhead or eight-decimal equivalence claims. Verification here concerns trace availability and score consistency; a timing experiment is not reconstructed from absent records.

3.4 FSC Stressor Identification

FSC is used as a theoretical organising construct for reasoning performed with finite representations and finite generation resources. It is motivated by fixed context and output limits, tokenisation, autoregressive commitment, and the limited information preserved in a finite prompt–response trajectory. These features can plausibly constrain computation, but the present experiment does not directly intervene on an internal circuit or measure a hidden "reasoning space". It therefore cannot establish that any individual error was caused by FSC.

For reproducible empirical reporting, row i is labelled an *FSC-consistent stress outcome* when all of the following conditions hold:

$$R_i = 1 \quad \text{the item belongs to a registered non-leaking TST family,} \tag{3.3}$$

$$G_i = 1 \quad \text{the stored ground truth is valid under the family solver,} \tag{3.4}$$

$$E_i = 1 \quad \text{the scorer can evaluate the response,} \tag{3.5}$$

$$Y_i = 0 \quad \text{the response is not exactly correct,} \tag{3.6}$$

$$B_i < \infty \quad \text{the output budget is recorded and finite.} \tag{3.7}$$

The deterministic indicator is

$$F_i = \mathbb{1}(R_i G_i E_i (1 - Y_i) = 1)\, \mathbb{1}(B_i < \infty). \tag{3.8}$$

The label describes a failed outcome under registered finite stress. It does not distinguish arithmetic error, instruction-following error, tokenisation error, inadequate knowledge, decoding variation, or a particular internal mechanism. Observable states and hypotheses are stored in separate fields so that a later validated annotation study could examine mechanisms without changing the primary endpoint.

3.4.1 Selection Criteria

Numerical magnitude and tokenisation motivate large-integer tasks (Singh & Strouse 2024). Recursion and repeated operations motivate tests of intermediate-state demands (Merrill, Sabharwal & Smith 2022, Dziri et al. 2023). Positional precision motivates counting and base-conversion tasks. These are task-design motivations rather than measured internal failure mechanisms.

3.4.2 Mapping Precursors to Stressors

The numerical, symbolic, repetitive, and sequential stressors are implemented through the families in Table 3.1. Each family is a generator category. The scorer's output states and possible mechanistic explanations are separate classifications.

3.5 TST Benchmark Construction

The TST benchmark generates mathematical problems procedurally. Its implementation and retained prompts determine the parameter ranges and reference answers reported here.

3.5.1 Algorithmic Generation Procedure

The retained generator implements seven families and 350 items at 50 items per family, not the five-family, 250-item schema stated in parts of the submitted dissertation. Each generated row contains `problem` and `answer`; the generator does not emit stable IDs, distractors, clipped difficulty scores, or a public seed field. Those facts were established by code inspection plus 3 500 generated validation items and comparison with 46 selected evidence files. Table 3.1 is the manuscript source of truth.

Table 3.1: Authoritative historical TST generator specification reconstructed from retained code.

Family	Implemented parameters	Reference solution
Large-number arithmetic	Integers independently sampled from $[10^{20}, 10^{50}-1]$; operation in $\{+, -, \times, //\}$	Exact Python integer arithmetic
Repetitive sequences	Lower-case character `a`–`z`; repetitions in $[100, 500]$	Exact character count
Combinatorics	$n \in [20, 100]$; $r \in [5, n-5]$	`math.comb(n,r)`
Base conversion	$(10, 2), (10, 16), (2, 10), (16, 10)$; value in $[10^8, 10^{14}]$	Python base conversion; lower-case hexadecimal without prefix
Symbolic expansion	$a, b, c, d \in [2, 25]$ for $(ax + b)(cx + d)$	$acx^2 + (ad + bc)x + bd$
Sequence prediction	Start in $[1, 50]$; difference in $[5, 250]$; 6–10 displayed terms	Next arithmetic-progression term
Chain reasoning	Initial value 50–500; 3–6 add, subtract, or multiply operations; deltas 5–90	Sequential exact integer arithmetic

Historical chain-reasoning prompts embed intermediate values in a way that reveals the answer path. This creates target leakage relative to the inference the dissertation seeks to make. Chain rows are retained in the archive for provenance but excluded from the primary baseline and paired tool analyses. The corrected primary analysis therefore has six families.

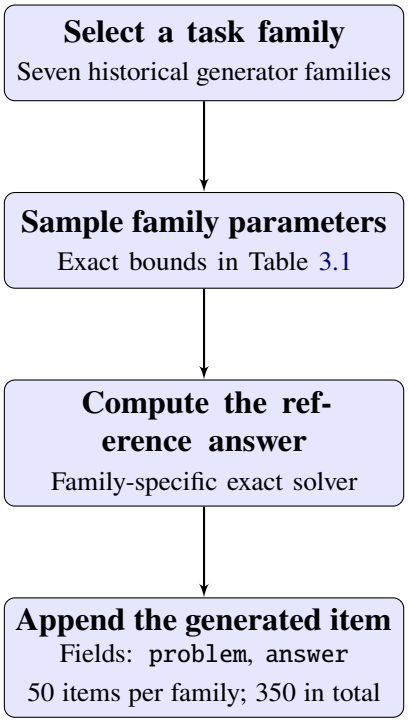


Figure 3.3: Procedural Generation Pipeline for the Token-Stress-Test (TST) Benchmark. Parameters are sampled stochastically to create unique problem instances that target specific structural precursors.

3.5.2 Sampling Strategy and Parameters

The five recorded maximum output budgets are 32, 64, 128, 256, and 512 tokens. Each of the six eligible families contributes 50 observations for every model–budget cell. The intended primary denominator is therefore

$$N = 6 \text{ models} \times 6 \text{ families} \times 5 \text{ budgets} \times 50 = 9\,000. \tag{3.9}$$

This balanced cell structure is verified from the ledger rather than assumed from a manuscript description.

The procedural generation reduces direct benchmark memorisation risk, but it does not make all items equally difficult or guarantee that difficulty is monotonic in the original sampling parameters. Family-specific continuous proxies are derived directly from each prompt, such as numerical magnitude, representation length, repetition count, combinatorial n and r, conversion direction and digit length, coefficient magnitude, and sequence length or difference. Within each family, the primary difficulty variable is transformed with $\log(1 + x)$ when positive and standardised. Because this difficulty score was reconstructed after the historical runs, its analysis is retrospective and is labelled as such.

3.5.3 Contamination Prevention

Procedural generation reduces the likelihood of verbatim benchmark memorisation but does not prove the absence of training-data overlap. Historical chain prompts explicitly revealed intermediate values; those rows are therefore excluded from the primary and paired analyses. The exclusion is recorded alongside each source row.

3.6 Experimental Framework

The baseline, tool-prompt sweep, diagnostic runs, and ablation archive are analysed as separate cohorts.

3.6.1 Model Selection

The historical baseline budget sweep uses six models: Claude 3.7 Sonnet (retained identifier `claude-3-7-sonnet-20250219`), Gemini 2.5 Flash (`gemini-2.5-flash`), GPT-4o mini, GPT-2, Llama 3.1 8B Instruct (`meta-llama/Llama-3.1-8B-Instruct`), and Mistral 7B Instruct v0.3 (`mistralai/Mistral-7B-Instruct-v0.3`). The retained records do not provide one immutable version or access date for every API-served system. Those omissions are reported as unavailable rather than reconstructed from present-day product names. Model is treated as a system label, not as a pure measure of parameter count: interfaces, tokenisers, instruction tuning, decoding defaults, and provider infrastructure may differ.

3.6.2 Inference Conditions

The recorded output budgets are 32, 64, 128, 256 and 512 tokens. Other generation settings are reported only where retained. A single temperature, top-p value, device, or seed cannot be asserted for every historical API and local call. Table B.1 records the available fields and gaps.

3.6.3 Comparison Conditions

The baseline requests an answer without external execution; the tool condition permits Python-assisted calculation. The intended comparison addresses the same original mitigation question (Schick et al. 2023, Gao et al. 2022). Retained group labels do not guarantee that the conditions used the same item, so exact pairing is checked before calculating an effect.

3.6.4 Evidence Inventory and Canonical Ledger

The retained project contains 573 candidate files occupying 209,862,326 bytes and 119,696 CSV row occurrences across 27 schemas. Inventory checks identified 45 byte-identical file copies, nine within-file duplicate rows, and no CSV parse failures. A file was not admitted merely because it contained a familiar column name. Files were classified as raw responses, derived scored data, cohort roll-ups, diagnostics, or paired-ablation material.

The canonical ledger selects 46 unscored source files, representing 23,345 row occurrences. Three exact file copies are excluded; 23,342 primary source occurrences remain. After removing leaking chain rows and applying cohort rules, 20,007 eligible primary non-chain observations remain across all historical and diagnostic cohorts. Every ledger row records its source path, source-row number, source hash, cohort, raw model and normalised model label, family, condition, budget, prompt, ground truth, raw response, derived item fingerprint, scorer output, trace availability, and exclusion reason. A frozen ledger hash (SHA-256 `8fcf38e89dc51d32186882661fe2fa79f75e442cd9134a9c9f1088a454fd0479`) identifies the evidence version used in this dissertation.

The historical cohorts are never pooled silently. The baseline budget sweep contains 10 500 archived rows including chain reasoning and 9 000 eligible non-chain rows. The historical tool-prompt sweep contains 11 550 occurrences; after excluding exact copies and chain rows it contains 9 897 eligible rows. Diagnostic baseline and tool-validation subsets contain 330 and 540 eligible rows respectively. The paired-ablation archive contains 280 total rows and 240 non-chain rows before the exact-item identity gate. Cohort labels are retained in all summaries.

3.6.5 Task-Specific Scoring and Observable States

The corrected scorer is versioned as `preservation-v3.0.0`. It extracts and normalises an answer according to the registered family rather than applying one generic last-number rule. Integer arithmetic supports arbitrarily large signed integers and rejects incidental prompt numbers; base conversion respects the requested representation; symbolic expansion compares normalised coefficient triples; sequence tasks compare the requested continuation; and repetitive tasks

compare the requested exact count or string structure. Missing, malformed, refused, or truncated responses remain in the endpoint denominator when the item and ground truth are valid.

Rows are excluded from an accuracy denominator only when the stored truth is invalid or unavailable, the family is unknown to the scorer, or the row falls outside the registered cohort. A generation failure is not treated as non-evaluable merely because it lacks a final answer. This rule prevents a model from improving its apparent accuracy by failing to answer.

The primary binary endpoint is $Y_i = 1$ for exact correctness and $Y_i = 0$ otherwise. To preserve diagnostically useful distinctions, incorrect rows are also placed into mutually exclusive observable states: (1) correct; (2) incorrect final answer; (3) no valid final answer; and (4) truncated with no valid final answer. These are parser-visible states, not causal labels. The previous labels *pattern break*, *algebraic incomplete*, *magnitude drift*, and *wrong operation* were reconstructed as deterministic parser fallbacks with no rater, adjudication, or agreement fields. Their frequency table is therefore not retained as a validated mechanism result.

Scorer verification comprises 13 regression and property checks, including 500 randomly generated large-integer cases, signs, decimals, multiple numeric strings, symbolic forms, sequences, refusals, truncation, and malformed output. A preliminary review of 49 full responses agreed with the automated result in all 49, but that review was performed during correction and was not independent or blinded. It is reported as a development check; a supervisor or second assessor should sign off a stratified sample before the package is described as independently human-validated.

3.7 Structural–Semantic Analysis

Counts, denominators, rates, and Wilson 95% confidence intervals are reported for the full baseline cohort, each model, each family, and each model–family cell. Wilson intervals are used because several cells have zero or very few events and Wald intervals perform poorly near the boundary (Wilson 1927). Zero-event cells are shown rather than suppressed. Response-state composition is reported alongside accuracy so that a wrong but parseable value is not conflated with a missing or truncated answer.

A historical family is classified as having a severe floor when its observed accuracy is below 1% and the confidence interval remains concentrated near zero. This descriptive gate triggered for base conversion, combinatorics, and repetitive sequences. Those families may still be useful as extreme stress tests, but they cannot rank systems reliably in their historical form. The inference appropriate to a floor is “the registered task distribution was too severe for comparative discrimination”, not “all models share the same capability limit”.

3.7.1 Predictive Modelling

The historical baseline has 499 correct outcomes among 9 000 observations. A 14-parameter correction-stage model gives approximately 35.6 events per parameter overall, but separation occurs for GPT-2, base conversion, and combinatorics because their cells have no successes.

Ordinary maximum-likelihood logistic regression consequently yields non-finite estimates. Class weighting would change the optimisation objective without adding successful observations and is not used as a remedy.

The primary model is Firth-penalised logistic regression, which applies a Jeffreys-prior penalty and yields finite estimates under separation (Firth 1993, Heinze & Schemper 2002). For response i,

$$\text{logit}\{P(Y_i = 1)\} = \beta_0 + \beta_1 z(B_i) + \beta_2 z\{\log(1 + L_i)\} + \beta_3 z\{\log(1 + D_i)\} + \gamma_{M_i} + \delta_{F_i}, \quad (3.10)$$

where B_i is $\log_2$ output budget, L_i is realised response length, D_i is the family-specific retrospective difficulty proxy, and M_i and F_i are model and family factors. Continuous variables are standardised over the analysis cohort. GPT-4o mini and sequence prediction are reference categories.

Penalised-Wald intervals are reported because they are available from the implemented estimator, but they are labelled approximate. The data include repeated generated items and model responses, while the implemented intervals are not cluster-robust. Realised response length is measured after generation and may mediate or reflect the same process as correctness; its coefficient is therefore associative, not a causal effect. Ordinary maximum likelihood and ridge-penalised logistic regression with penalty 0.5 are sensitivity checks. Grouped five-fold validation by item fingerprint assesses out-of-sample discrimination and Brier score without allowing the same item into training and validation folds.

3.7.2 Correlation and Taxonomy Construction

The four response states describe exact correctness and answer availability. Mechanism-style labels in the earlier files were assigned by parser fallbacks, without an independent annotation study. Correlations between structural summaries and correctness cannot validate those labels or identify an internal cause. Entropy comparisons are restricted to the three systems with retained logits.

3.7.3 Handling Class Imbalance

Class weighting does not add successful observations. The analysis reports the event count, parameter count, zero-success groups, uncertainty, Firth estimates, and sensitivity results. Grouping by item during validation limits leakage between training and validation folds. Penalised intervals remain approximate and do not account fully for repeated-item dependence.

3.8 Tool-Augmentation Efficacy Analysis

The original tool objective is evaluated through paired answer outcomes and available execution events.

3.8.1 Hypotheses

The original performance-recovery hypothesis is that valid tool requests and successful execution may improve mathematical accuracy. The structural-offloading hypothesis is that tool use may change generated length or output uncertainty. Both require empirical evidence; successful code execution alone does not demonstrate a correct integrated answer or reduced cost.

3.8.2 Efficacy Metrics

We define **Efficacy (E)** as the recovery rate relative to the baseline failure gap:

$$E = \frac{\text{Acc}_{tool} - \text{Acc}_{baseline}}{1 - \text{Acc}_{baseline}} \tag{3.11}$$

This metric normalises gains against the room for improvement, allowing comparison across categories of varying baseline difficulty.

A paired comparison requires the same model, item, ground truth, and relevant generation settings in baseline and tool conditions. A declared group identifier is insufficient. The audit recalculates a fingerprint from the normalised problem and truth, verifies one response per condition, and excludes leaking chain tasks. Of 140 declared groups, 70 contain exact-item pairs and 70 pair different problems. Ten of the exact pairs are chain-reasoning items, leaving 60 eligible pairs for inference.

For valid pairs, the outcomes are counted in a 2×2 table: both correct, baseline-only correct (harm), tool-only correct (recovery), and neither correct. The primary effect is the paired percentage-point difference in accuracy; exact McNemar inference uses only the discordant recovery and harm counts. If no discordant successes occur, a p-value cannot turn the design into evidence of equivalence; the sample recovery efficacy is zero, but the archive does not demonstrate a recovery benefit or establish population equivalence.

The historical logs do not provide a complete, paired event schema for every row. A secondary funnel is therefore descriptive and uses the 120 eligible with-tool rows irrespective of pair validity. It records: assignment to the tool condition; presence of a code-block proxy; whether execution was attempted; execution success; and verification of the tool result. Proxy detection is not equated with a genuine model request, and successful execution is not equated with a correct integrated final answer. Latency, retry count, prompt/completion usage, and monetary cost are unavailable and are not estimated retrospectively.

3.9 Ethical Considerations and Reproducibility

The benchmark uses generated mathematical tasks and contains no human participants or personal data.

3.9.1 Reproducibility

The analysis package retains immutable source hashes, canonical ledgers, scripts, tests, machine-readable tables, figures, and provenance sidecars. The original submitted PDF remains separately hashed and unchanged. Every principal number in Chapters 4 and 5 is generated from the frozen ledger or a named audit output. Reproducibility fields are populated only when recovered from logs or configuration: the archive is complete for prompt, raw response, ground truth, model label, family, condition, and budget, but incomplete for several API version dates, provider-side tokenisers, hardware details, random seeds, and costs. The field-level manifest is in Appendix B; the data and scripts are retained in the accompanying correction package.

3.9.2 Safety

Historical records do not establish a uniform restricted execution environment for every tool call. Any future execution of model-generated code requires resource limits and restricted access. That recommendation does not retrospectively establish that safeguards were used in all historical runs.

3.10 Summary

The methodology specifies the existing study through its measured variables, benchmark implementation, cohort rules, task-specific scoring, and statistical procedures. It documents where trace availability, incorrect pairing, zero-success groups, and missing metadata restrict inference.

Chapter 4: Experiments and Results

4.1 Introduction

This chapter reports the observed results from TokenPath and TST. The primary baseline is distinguished from diagnostic and tool cohorts. Counts, uncertainty, plots, and measurement limits are reported for the original tracing, stress, comparison, and mitigation questions.

4.2 Visualising Token-Level Generation

TokenPath records output distributions where available. These measurements describe generation and do not directly reveal hidden reasoning states.

4.2.1 Validation of the TokenPath Instrument

Table 4.1 reports the eligible denominators after duplicate and leaking-family exclusions. The historical baseline is the primary descriptive cohort because its model–family–budget cells are balanced. The historical tool-prompt sweep is not treated as a paired comparison: it contains a different denominator and, in places, different systems or unavailable calls. The exact paired analysis is reported separately in Section 4.5.

Table 4.1: Eligible historical and diagnostic cohorts. Confidence intervals are Wilson 95% intervals. Entropy availability denotes rows with comparable retained token distributions.

Cohort	*N*	Correct	Accuracy	95% CI	Entropy *N*
Diagnostic baseline subset	330	22	6.67%	4.44–9.89%	150
Diagnostic tool subset	540	20	3.70%	2.41–5.65%	30
Historical baseline sweep	9 000	499	5.54%	5.09–6.04%	4 500
Historical tool-prompt sweep	9 897	499	5.04%	4.63–5.49%	2 397
Paired-ablation rows before identity gate	240	0	0.00%	0.00–1.58%	60

The inventory began with 119 696 CSV row occurrences, but that number is not a number of independent model responses. It includes scored derivatives, summaries, diagnostics, and exact copies. The ledger's source-selection and cohort rules prevent those artefacts from inflating any experimental denominator. This correction resolves the submitted dissertation's incompatible totals, including the previously unexplained comparison of 57 correct with 9 640 automatically labelled failures.

4.2.2 Search Path Dynamics

Figures 4.1 and 4.2 report structural summaries only for the three systems with retained logits. They replace cross-interface comparisons that treated unavailable API distributions as though

they were measured. Entropy is aggregated at the response level and displayed by budget; effective vocabulary is its exponentiated transformation, so the two plots are not independent evidence.

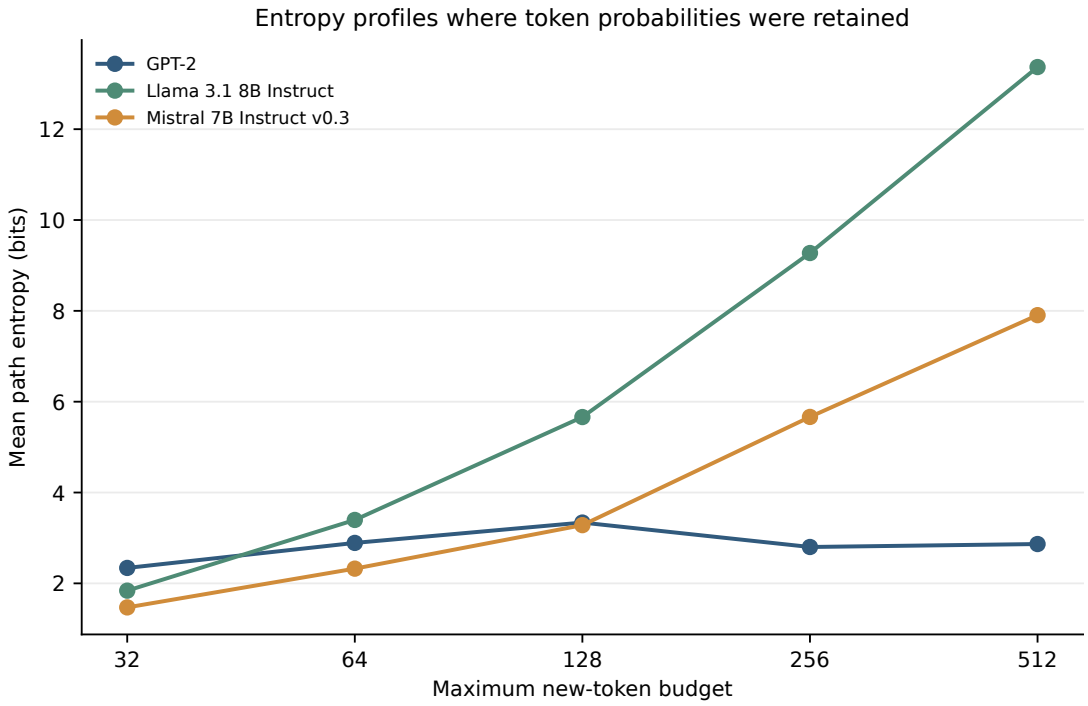


Figure 4.1: Response-level entropy summaries by output budget for GPT-2, Llama 3.1 8B Instruct, and Mistral 7B Instruct v0.3 ($N = 1\,500$ per model). Claude, Gemini, and GPT-4o mini are omitted because comparable token distributions were not retained, not because their entropy was zero.

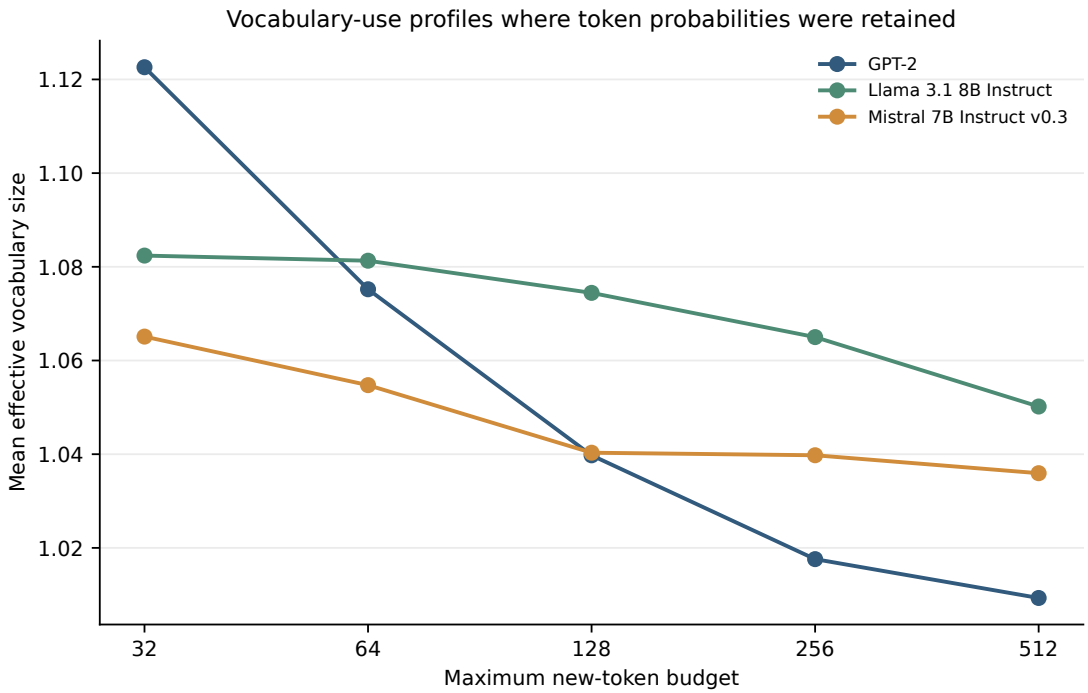


Figure 4.2: Effective-vocabulary summaries, $\exp(H)$, by output budget for the same 4 500 open-weight rows as Figure 4.1. The metric describes next-token distribution concentration; it is not a validated detector of hallucination or logical failure.

The retained open-weight traces show measurable distributional variation with budget and system. However, entropy is absent for half the primary cohort, outcome prevalence is low, and response-level aggregation can hide local transitions. The historical archive does not support the submitted numerical claim that a specific entropy threshold identifies "sudden divergence" or "drifting hallucination". Those labels remain hypotheses rather than observed mechanisms.

4.3 The Stress Factors: Defining the FSC

The TST families are evaluated as task categories, using the operational rule and scoring procedure defined in the methodology.

4.3.1 TST Benchmark Characterisation

Table 4.2 demonstrates strong family heterogeneity. Sequence prediction accounts for 323 of all 499 correct responses; symbolic expansion contributes 136 and large-number arithmetic 38. Repetitive sequences contributes two, while base conversion and combinatorics contribute none.

Table 4.2: Historical baseline accuracy by non-leaking TST family. Each family contains six models, five budgets, and 50 observations per model–budget cell.

Family	N	Correct	Accuracy	Wilson 95% CI
Base conversion	1 500	0	0.00%	0.00–0.26%
Combinatorics	1 500	0	0.00%	0.00–0.26%
Large-number arithmetic	1 500	38	2.53%	1.85–3.46%
Repetitive sequences	1 500	2	0.13%	0.04–0.48%
Sequence prediction	1 500	323	21.53%	19.53–23.68%
Symbolic expansion	1 500	136	9.07%	7.72–10.63%

The historical forms of base conversion, combinatorics, and repetitive sequences satisfy the severe-floor gate. A zero does not show that all individual items are impossible, and it does not distinguish systems at the observed settings. The corrected interpretation is therefore narrower: those registered distributions functioned as extreme stress tests and did not provide a useful comparative success gradient. Further calibration would be needed to compare success boundaries in those families.

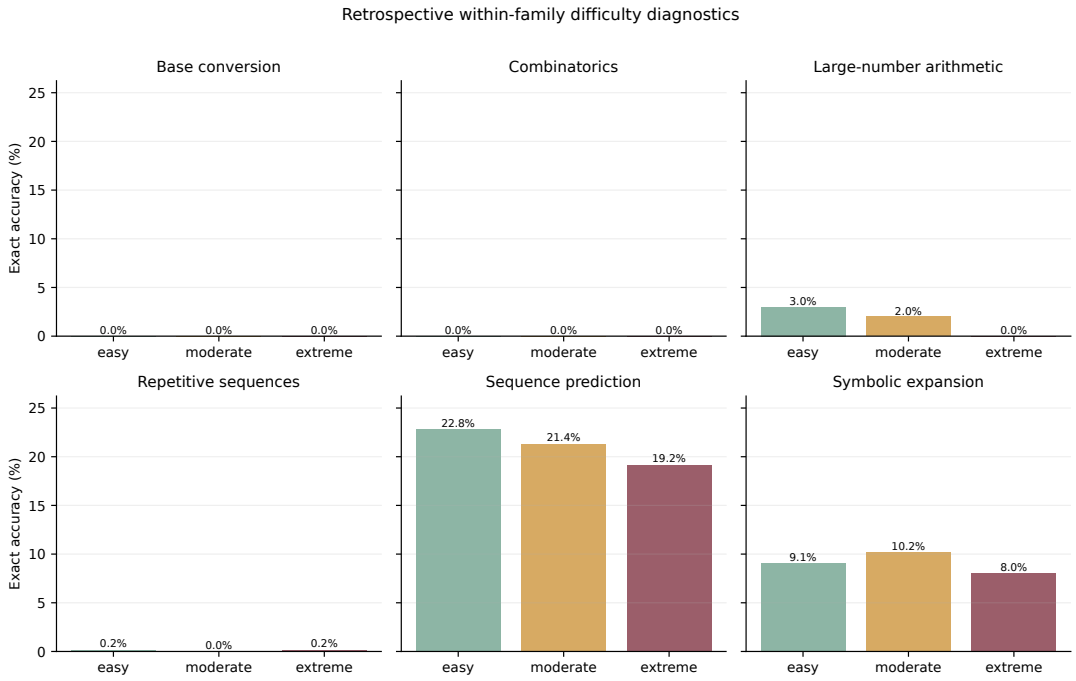


Figure 4.3: Retrospective difficulty tiers within each historical family. Tiers are derived after the historical runs from prompt-level family parameters and are used to diagnose floor behaviour, not presented as preregistered causal contrasts. Each panel states its available denominator.

Figure 4.3 provides the retained-data difficulty audit. It confirms that family membership alone is too coarse: one "hard" label can combine changes in representation length, operand magnitude, operation count, or output format. The retrospective tiers are informative for selecting

prospective ranges, but the historical sampling does not guarantee balanced observations across every tier and model.

4.3.2 Failure Mode Taxonomy

The primary baseline produced 499 exact answers among 9 000 responses: 5.54% accuracy (95% CI 5.09–6.04%). The complementary responses comprised 3 521 incorrect final answers (39.12%), 3 585 responses with no valid final answer (39.83%), and 1 395 truncated responses with no valid final answer (15.50%). These categories sum with the 499 correct responses to the full denominator. None is discarded because failure to produce a scoreable final value is itself a failure on an eligible item.

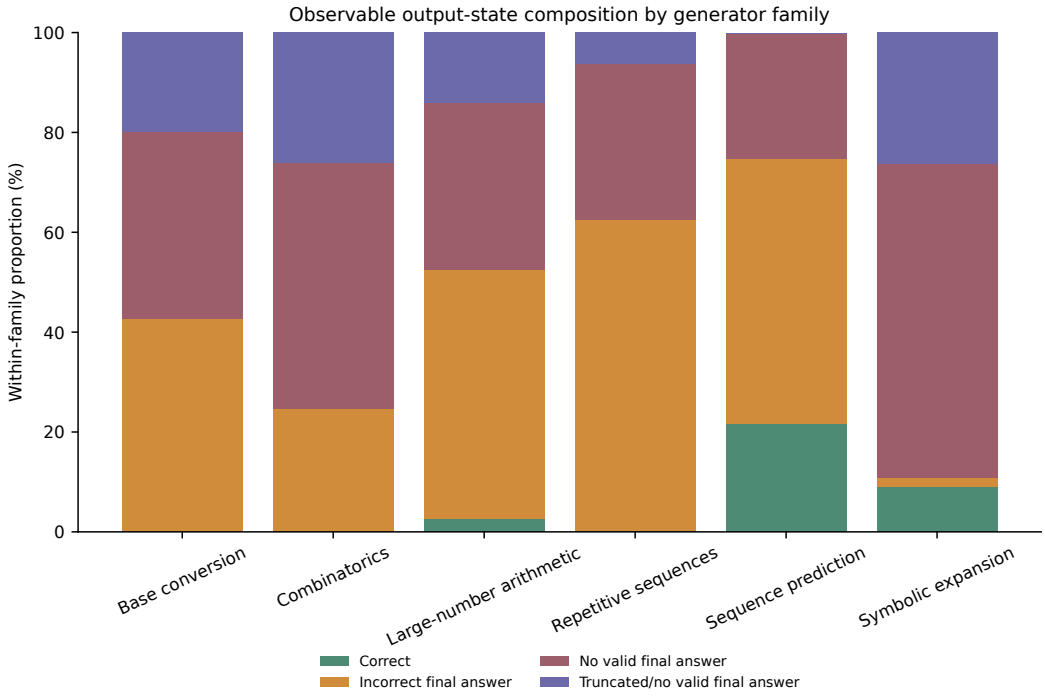


Figure 4.4: Observable response-state composition in the historical baseline, by TST family. States are determined by the versioned task-specific scorer: correct, incorrect final answer, no valid final answer, and truncated/no valid final answer. Each family has $N = 1\,500$. These states describe visible outputs and are not internal mechanism labels. Source: frozen canonical ledger and `generate_descriptive_tables.py`.

Figure 4.4 shows that the kind of failure differed by family even where exact accuracy was similarly low. Base conversion combined incorrect values, missing final answers, and 20.00% truncated responses. Combinatorics had no successes and 49.33% of responses lacked a valid final answer. Repetitive sequences mostly ended with a parseable but incorrect value (62.47%). Sequence prediction had the highest correct fraction and almost no truncated/no-answer cases (0.20%). Symbolic expansion occupied an intermediate regime. These differences justify reporting states separately, but they do not identify why a model produced them.

4.4 Comparative Performance

All six systems contribute 1 500 primary observations. Table 4.3 reports exact rates and intervals. Llama 3.1 8B Instruct has the highest pooled rate, 8.87%; GPT-2 has no successes; the other four systems range from 5.00% to 6.67%. Pooled rates are not a general leaderboard because task families and very short budgets dominate the evaluation.

Table 4.3: Historical baseline accuracy by model system. Each row contains six families, five budgets, and 50 observations per family–budget cell.

Model	N	Correct	Accuracy	Wilson 95% CI
Claude 3.7 Sonnet	1 500	96	6.40%	5.27–7.75%
Gemini 2.5 Flash	1 500	100	6.67%	5.51–8.04%
GPT-4o mini	1 500	95	6.33%	5.21–7.68%
GPT-2	1 500	0	0.00%	0.00–0.26%
Llama 3.1 8B Instruct	1 500	133	8.87%	7.53–10.41%
Mistral 7B Instruct v0.3	1 500	75	5.00%	4.01–6.22%

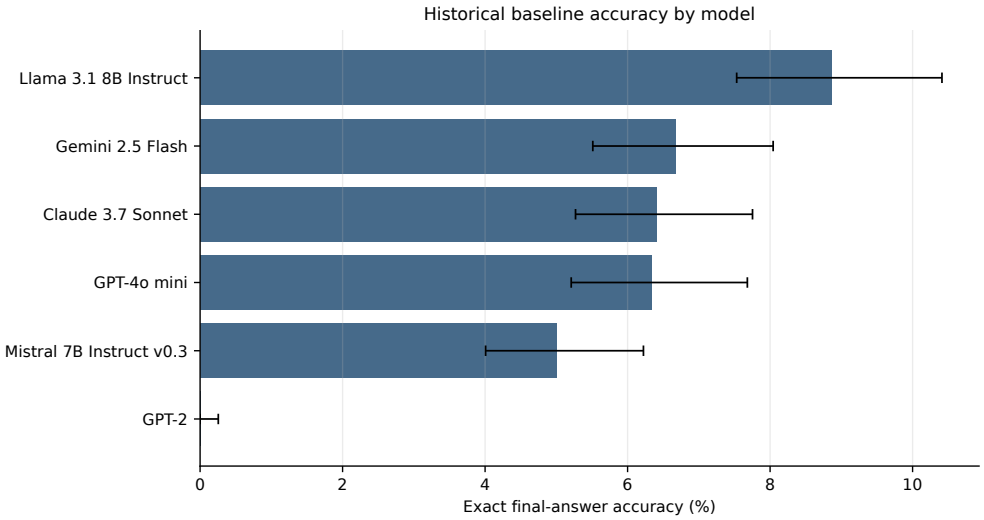


Figure 4.5: Pooled exact-answer accuracy and Wilson 95% confidence intervals for the six historical systems ($N = 1\,500$ per system). The estimand is accuracy on the registered six-family, five-budget TST mixture, not general mathematical competence.

The pooled values conceal clear model–family specialisation. Claude 3.7 Sonnet obtains 34/250 on large-number arithmetic and 61/250 on sequence prediction but 0/250 on symbolic expansion. Gemini 2.5 Flash and GPT-4o mini each obtain 50/250 on symbolic expansion, while their large-number arithmetic rates are 2/250. The Llama system's successes are distributed differently again. The complete 36-cell table is supplied in the appendix and the heat map in Figure 4.6 preserves zero-event cells.

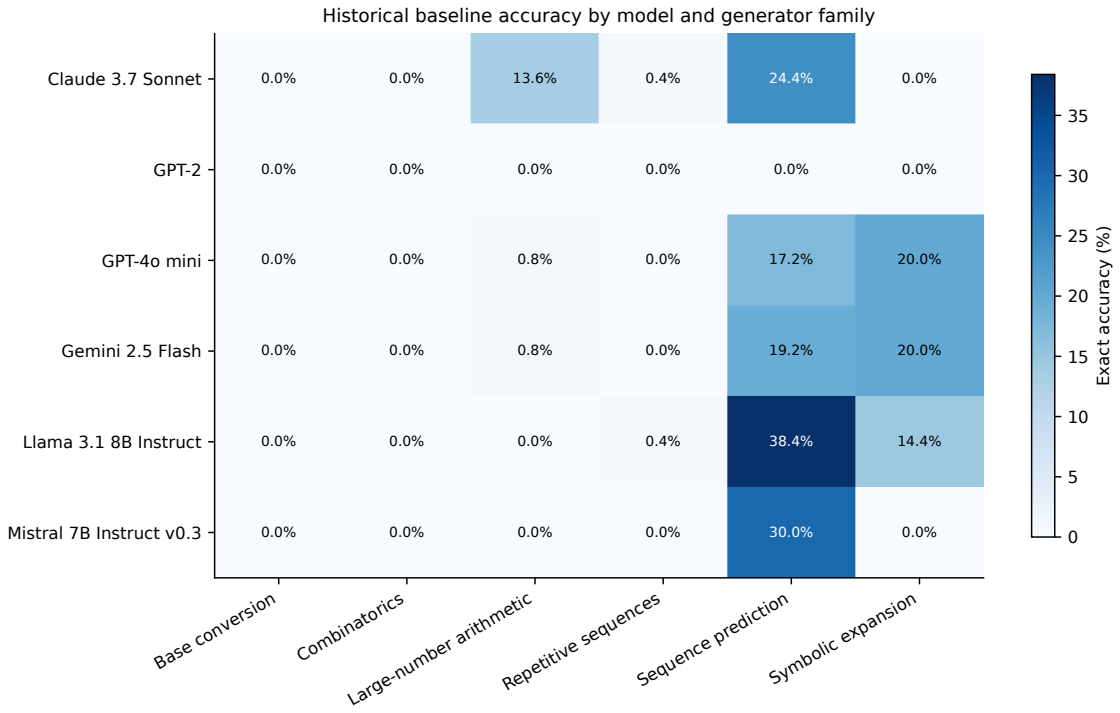


Figure 4.6: Exact-answer accuracy for all 36 model–family cells, each with $N = 250$ (five budgets by 50 observations). Cell labels show rates; the accompanying machine-readable table gives counts and Wilson intervals. No zero-event cell is removed.

4.4.1 Length Utilisation Profiles

Accuracy depended strongly on the registered maximum output budget. At 32 tokens, no response was correct among 1 800 observations (upper Wilson bound 0.21%). At 64 tokens, one of 1 800 was correct (0.056%). At 128 tokens, three were correct (0.17%). The rate increased to 109/1 800 at 256 tokens (6.06%, 95% CI 5.04–7.25%) and 386/1 800 at 512 tokens (21.44%, 95% CI 19.61–23.40%). The budget result is an association under the historical prompts. It is not a measure of how many tokens are intrinsically required by a mathematical problem because model responses may use the available space inefficiently or stop early.

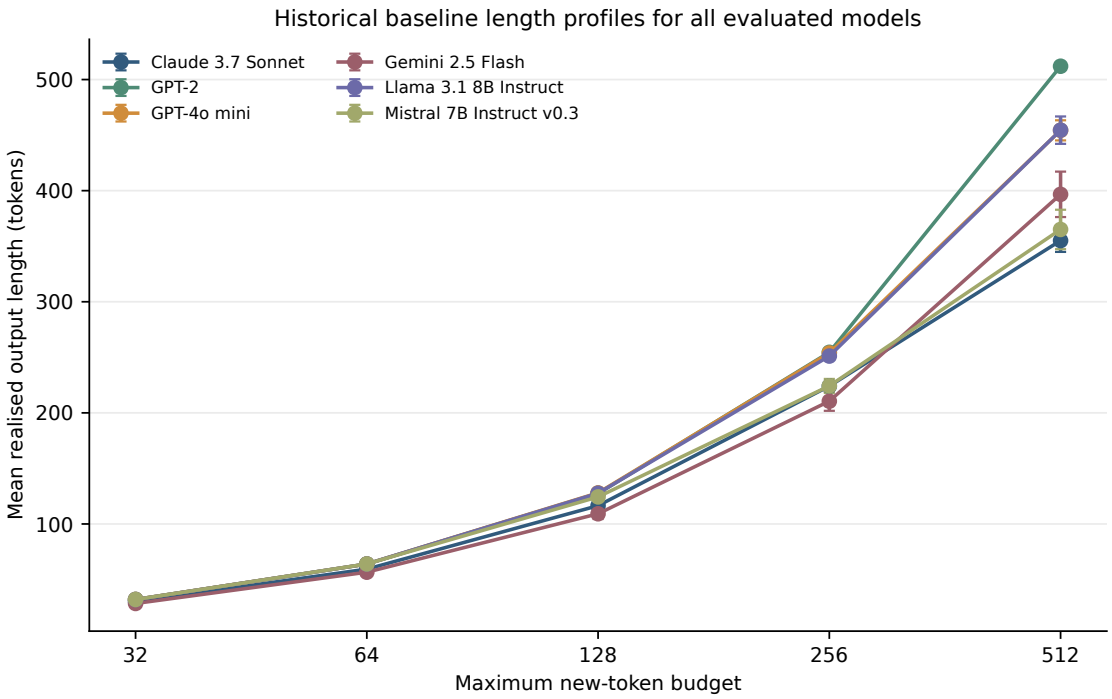


Figure 4.7: Realised response length by registered maximum output budget for all six historical model systems. Length is available for all 9 000 primary rows ($N = 1\,500$ per model). Panels use common axes and display the actual model names rather than an unidentified representative system. Source: frozen canonical ledger; figure provenance sidecar `fig_length_by_budget_all_models.json`.

Figure 4.7 reports the length comparison across the full cohort. It shows that observed length and maximum budget are related across every system but not identical. Provider interfaces and termination behaviour differ, so a budget coefficient should not be interpreted as a pure architecture effect.

4.5 Neuro-Symbolic Trade-offs

4.5.1 Pair identity and efficacy

The paired-ablation archive contains 140 declared groups (280 rows). Exact problem-and-truth fingerprinting confirms only 70 same-item pairs; 70 groups pair different items and are excluded from paired inference. Ten valid pairs belong to the leaking chain family, leaving 60 eligible non-chain pairs. Among those 60, baseline correct = 0, tool correct = 0, recovery = 0, and harm = 0. The sample recovery efficacy defined by $E = (\text{Acc}_{tool} - \text{Acc}_{baseline})/(1 - \text{Acc}_{baseline})$ is $(0 - 0)/(1 - 0) = 0$. The paired difference is 0.00 percentage points and the exact McNemar calculation returns 1 because there are no discordant successes. This is an uninformative result, not evidence that tools have no effect. In particular, the historical data do not support claims of improved accuracy, negligible overhead, or cost-effectiveness.

4.5.2 Historical event funnel

The funnel in Figure 4.8 uses all 120 eligible with-tool rows and answers a different question: what recorded tool-like events occurred? A code-block proxy appeared in 90/120 responses (75.0%). Execution was attempted for 89/120 (74.2%). Of attempted executions, 34/89 succeeded (38.2%) and 55/89 failed (61.8%); of those successes, 22/34 were marked verified (64.7%). The denominators deliberately change at each stage. No rate is labelled "recovery" because the final-answer pairing gate did not establish a correct counterfactual outcome.

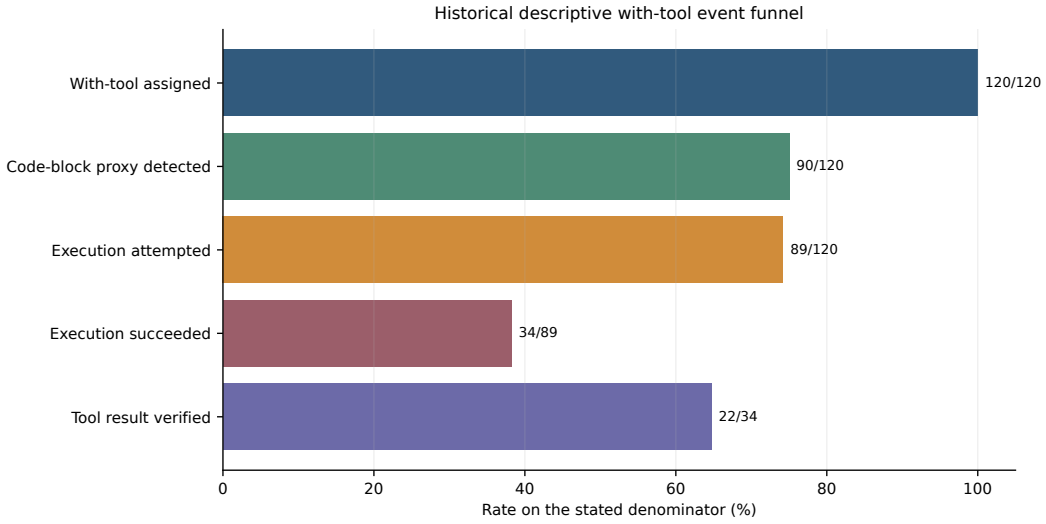


Figure 4.8: Historical tool-event funnel for 120 eligible with-tool rows. Code-block detection is a proxy, execution success is conditional on an attempted execution, and verification is conditional on success. The funnel describes process attrition; it is not a paired treatment effect. Latency, retries, usage cost, and verified answer integration were not completely retained.

4.5.3 Structural Overhead Analysis

The retained records lack consistent latency, usage and execution-cost measurements. The original length-delta figures are therefore withdrawn as evidence of negligible overhead. No structural cost-neutrality conclusion is drawn from the unpaired tool-prompt sweep.

4.5.4 Token Budget Utilisation Under Tool Augmentation

The tool-prompt sweep and baseline have different denominators and cannot supply a matched utilisation effect. The exact-item subset contains no successful final answers in either condition. Lower length or early termination would not by itself demonstrate recovery, calibration, or efficient computation.

4.6 Entropy and Failure Signals

Entropy and effective-vocabulary plots are reported in the tracing section for the systems with comparable retained logits.

4.6.1 Entropy Stratification

The submitted 57-correct/9,640-incorrect table is withdrawn because its scoring and cohort definitions do not reconcile with the primary experiment. The corrected denominator and response states are reported above. The archive provides no validated token-level time of semantic error against which an early-warning threshold could be evaluated.

4.6.2 Rare-Event Regression Results

Ordinary logistic regression produced non-finite estimates because GPT-2, base conversion, and combinatorics have no correct outcomes. Firth regression converged and the ridge sensitivity model also converged. The primary Firth model used 9 000 observations, 499 events, and 14 parameters. Table 4.4 reports the three continuous terms; model and family contrasts are shown in Figure 4.9 and the appendix.

Table 4.4: Selected Firth-penalised logistic-regression associations with exact correctness. Odds ratios are per one standard deviation. Intervals are approximate penalised-Wald intervals and are not cluster-robust.

Term	Adjusted odds ratio	Approximate 95% CI
$\log_2$ output budget	57.7	31.9–104
log(1+ realised response length)	1.48	0.799–2.76
Within-family log(1+ difficulty)	0.865	0.751–0.995

The very large budget association reflects the near absence of successes at 32–128 tokens and the concentration of successes at 512. Realised length has a wide interval spanning one after simultaneous adjustment. The retrospective within-family difficulty term is modestly negative. Model and family contrasts include extremely small finite odds ratios in separated zero-event categories; they are useful for stabilising prediction but should not be read as precise effects. In grouped five-fold validation by item fingerprint, the Firth model achieved area under the ROC curve 0.984 and Brier score 0.0205; mean predicted correctness was 5.57%, close to observed 5.54%. High discrimination is expected in part because budget and family sharply separate outcome prevalence.

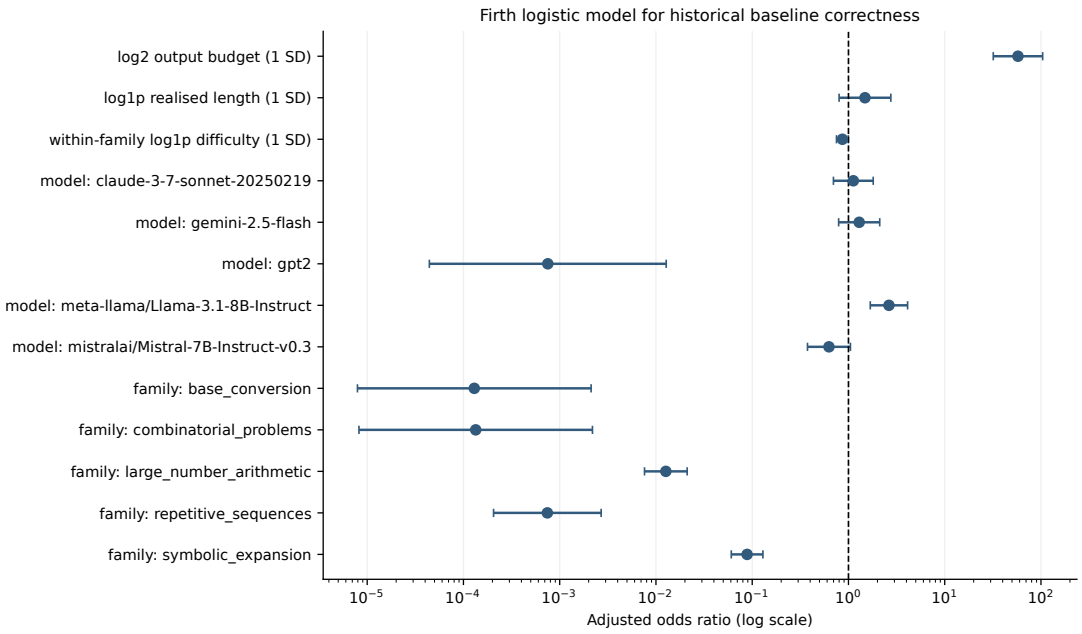


Figure 4.9: Adjusted odds-ratio estimates from the 14-parameter Firth model. GPT-4o mini and sequence prediction are reference categories. Extreme finite estimates correspond to separated zero-event groups and require cautious interpretation. The figure is descriptive of the historical registered design, not a causal model of architecture.

4.7 Stability Metrics

The analysis reports exact answer counts and uncertainty alongside structural measurements.

4.7.1 Normalised Token Efficiency

The submitted penalty-based efficiency ranking is withdrawn: its numerator and denominator relied on the unreconciled outcome labels. It is not replaced with a claim that wrong or short responses are efficient. Counts, lengths, and Wilson intervals are reported directly, including zero-success cells.

4.8 Concluding Remarks

The baseline contains 499 correct answers among 9,000 eligible responses. Successes vary by model, family and budget. Comparable output distributions are retained for three systems. The 60 exact non-chain pairs contain zero correct answers in either condition, and the separate execution funnel records incomplete progression from code generation to verification.

4.9 Summary

The tracing, stress-family, comparative, tool, entropy, and stability results use explicit denominators and distinguish observed outputs from inferred mechanisms. All primary counts reconcile with the same frozen ledger.

Chapter 5: Discussion

5.1 Introduction

This chapter interprets the tracing, benchmark, comparative, and tool results in relation to the original research questions. The questions are restated here for ease of reference.

i. **RQ1: Token-Level Tracing**
   How can token-level signals (e.g., logits, entropy trajectories, vocabulary concentration) be captured and visualised to provide transparent diagnostics of internal reasoning dynamics in LLMs?

ii. **RQ2: Stress-Induced Failures**
   What forms of numerical, structural, or contextual stress reliably induce Finite-Space Constraints, and how can a purpose-built token-stress-testing benchmark (TST) be constructed to expose these failure modes?

iii. **RQ3: Model-Level Characterization**
   How do state-of-the-art LLMs differ in their susceptibility to FSC-related failures when evaluated on the TST benchmark, and what taxonomy of failure modes emerges from comparative analysis?

iv. **RQ4: Structural–Semantic Linkages**
   To what extent do structural indicators such as entropy, length utilization, and vocabulary concentration correlate with semantic reasoning outcomes, and can these signals serve as predictors of failure?

v. **RQ5: Mitigation via Tool-Augmented Decoding**
   Does lightweight tool augmentation (e.g., Python-assisted decoding) mitigate FSC-induced failures, and what is its impact on structural efficiency and reasoning accuracy?

5.2 Internal Signals of Failure

The distributional measurements address RQ1 and RQ4. Their interpretation depends on what TokenPath actually records and whether a semantic error has been independently identified.

5.2.1 Uncertainty During Generation

TokenPath provides a reproducible path from a generated prompt and response to structural summaries when the underlying interface exposes token distributions. In the historical archive, realised response length is observable for all six model systems. Token entropy and effective vocabulary are observable for GPT-2, Llama 3.1 8B Instruct, and Mistral 7B Instruct v0.3, but not for Claude 3.7 Sonnet, Gemini 2.5 Flash, or GPT-4o mini. The corrected answer to RQ1 is therefore not that TokenPath exposes a common set of internals for every modern model. Rather, it provides a common trace schema with an availability-aware subset of metrics.

This distinction matters scientifically. Comparing an open-weight logit-derived entropy with an API system's absent value would confound measurement with architecture. Even a provider-supplied top-k log-probability list might not be directly comparable with a full local vocabulary distribution unless truncation and normalisation were handled explicitly. The availability matrix in Appendix B is consequently part of the result, not an inconvenience to hide.

Entropy is also limited as an interpretation target. Semantic uncertainty can help detect whether different sampled meanings conflict (Kuhn et al. 2023), while calibration research asks whether reported or implicit confidence tracks correctness (Kadavath et al. 2022). Token entropy answers neither question by itself: it measures dispersion of the next-token distribution at a particular step. A mathematical answer may follow a high-entropy wording choice and still be correct, or may be generated confidently and be false. TokenPath is best regarded as an instrument for generating testable hypotheses about transitions and concentration, not as a glass box that reads out logical validity.

The all-model length figure and open-weight structural panels still make a useful contribution. They expose budget usage, termination patterns, and distributional variation that endpoint accuracy alone cannot show. They can support prospective questions such as whether a local entropy change precedes an independently annotated error, whether that signal generalises across item fingerprints, and whether an intervention reduces the signal while improving exact outcomes. Those questions require aligned temporal annotations or controlled interventions, neither of which is present in the historical archive.

5.2.2 Taxonomy of Failure

The original characterisation objective remains relevant, but the retained evidence supports observable response states more strongly than internal mechanisms. The automatic labels *pattern break*, *algebraic incomplete*, *magnitude drift*, and *wrong operation* were generated through parser rules. No independent rater assignments or agreement statistics establish their validity as mechanism classes.

The earlier terms *sudden divergence* and *drifting hallucination* can describe hypotheses for further investigation. A sudden change in a token distribution does not, without an intervention, demonstrate circuit failure; a gradual change does not establish context saturation. Restricted mechanistic studies of induction and arithmetic motivate possible explanations (Olsson et al. 2022, Nanda et al. 2023, Stolfo et al. 2023), but their explanations cannot be transferred to these traces solely from a similar output pattern.

The four states used for comparative reporting are correct, incorrect final answer, no valid final answer, and truncated/no valid final answer. Their decision rules are reproducible. This answers the descriptive part of the original objective while identifying the validation needed before a structural-to-semantic mechanism taxonomy can be claimed.

5.2.3 Addressing RQ4: Reliability of Structural Signals

The retained baseline supports a strong association between output budget and exact correctness. It also supports family and model differences and a modest negative association with reconstructed within-family difficulty in the Firth model. Realised length has an adjusted odds ratio of 1.48 per standard deviation, but its approximate interval includes one. These results answer part of RQ4: registered structural and design variables are predictive of outcomes in the historical cohort.

Several qualifications prevent a causal reading. Output budget was varied, but budget is entangled with response truncation and with whether a model has space to express a final answer. Realised length is post-generation and may be a mediator, consequence, or marker of the same process that produced correctness. The difficulty variable was reconstructed after the historical runs. Model and family contrasts are not randomised. Item responses repeat across systems and budgets, while the reported penalised-Wald intervals are not cluster-robust. The grouped cross-validation results show predictive discrimination on held-out item fingerprints, not identification of an internal mechanism.

Firth regression is preferable here because complete separation makes ordinary maximum-likelihood coefficients diverge (Firth 1993, Heinze & Schemper 2002). It does not manufacture missing successful events or make a zero-event family informative. The extremely small finite odds ratios for GPT-2, base conversion, and combinatorics are regularised summaries of separation. They should not be reported with the rhetorical certainty of a precisely measured architecture effect.

The open-weight entropy and vocabulary measures remain exploratory. Only 4 500 of 9 000 rows contain them, and availability coincides perfectly with a subset of models. A pooled coefficient would therefore conflate structural measurement availability with system identity unless analysis were restricted to the open-weight subset. The dissertation does not use entropy to classify hallucinations or logical invalidity because there is no independently validated trace-level target. This correction creates a clean bridge between Chapter 2's literature and the empirical evidence: entropy can be a candidate diagnostic, endpoint correctness is the primary measured outcome, and reasoning faithfulness or logical validity requires its own evaluation (Lanham et al. 2023, Turpin et al. 2024).

The historical patterns are compatible with several explanations. Short budgets may truncate solutions; large exact operands may interact badly with tokenisation; instruction-following may fail; next-token uncertainty may increase at a critical arithmetic step; or the model may confidently execute the wrong operation. The present design observes outcomes under finite constraints, not the counterfactual internal computation that would separate these accounts. For that reason, the FSC construct is retained as a framework and the empirical label is called *FSC-consistent*, not *FSC-caused*.

5.3 The Limits of "Approximate" Reasoning

The benchmark and comparative questions concern the tested task distributions and model systems.

5.3.1 Mathematical Outcomes Under Finite-Space Constraints

The TST generator successfully creates procedurally varied exact-answer tasks, but the audit changes the description of what it contains. Seven families were implemented; chain reasoning leaks intermediate targets and is excluded from primary inference. The six eligible historical families span exact numerical computation, representation conversion, symbolic expansion, repetition, and arithmetic sequences. This is a wider set than a single benchmark score suggests.

The historical results show that output budget and family are major determinants of success. Only four correct responses occur below 256 tokens, compared with 109 at 256 and 386 at 512. Sequence prediction reaches 21.53%, while three families are at or below 0.13%. These observations support the narrow statement that the registered prompts and short generation budgets expose unreliable exact output. They do not demonstrate a universal limit of Transformer computation. In particular, a system could fail because the answer cannot be represented before truncation, because it never follows the requested format, because arithmetic is inaccurate, or because the prompt is outside its instruction-tuning distribution.

The severe floor is central to construct validity. A stress test may deliberately seek a boundary where systems fail, but a task distribution with no successful outcomes supplies little information about relative capability or the shape of a failure boundary. The base-conversion and combinatorics ranges are extreme: values up to 10^{14} are converted, and combinations may use n up to 100. Repetitions may extend to 500 characters. These families are valid as extreme stress probes but not, in their historical form, as discriminative benchmarks across the tested systems.

Procedural generation reduces direct reuse of familiar benchmark questions, but it is not automatically contamination-proof. Models may have learned the underlying algorithms, common prompt templates, or code patterns. Conversely, knowing the algorithm does not guarantee execution for large exact values. TST should therefore be described as a controlled stress distribution with deterministic ground truth, not a pure measure of previously unseen reasoning.

5.3.2 Parameter Scale vs. Structural Limits

The tested systems differ, but their pooled differences are smaller than the family and budget contrasts and are not ordered simply by scale. Llama 3.1 8B Instruct has the highest pooled historical rate at 8.87%, followed by Gemini 2.5 Flash at 6.67%, Claude 3.7 Sonnet at 6.40%, GPT-4o mini at 6.33%, Mistral 7B Instruct v0.3 at 5.00%, and GPT-2 at 0%. The Wilson intervals for the four middle systems overlap substantially. These values describe the registered mixture; they do not prove that an 8B model is generally better at mathematics than the proprietary systems.

The 36-cell model–family table is more informative. Claude's historical successes concentrate in large-number arithmetic and sequence prediction; Gemini and GPT-4o mini obtain substantial symbolic-expansion success; other systems show different profiles. Such interactions are consistent with differences in tokenisation, training data, instruction tuning, decoding, or response formatting. The design does not isolate which explanation applies. A claim that larger models "manage FSC more gracefully" is therefore too broad for the evidence and has been withdrawn from the abstract and conclusion.

The corrected taxonomy is deliberately observable. Every eligible response is one of four states: correct; incorrect final answer; no valid final answer; or truncated with no valid final answer. This taxonomy has deterministic rules, retains every eligible row in the denominator, and separates a wrong value from absence of a valid value. It is reproducible without claiming access to a cognitive process.

By contrast, the submitted mechanism-style counts were generated by deterministic parser fallbacks. Across 33 scored files the legacy counts were 3 260 "pattern break", 3 260 "algebraic incomplete", 1 829 "magnitude drift", and 1 291 "wrong operation", totalling 9 640. No independent rater, annotation codebook, adjudication field, or agreement estimate accompanies those records. The labels may be useful prompts for qualitative inspection, but their frequencies cannot be presented as a validated taxonomy of mechanisms.

If mechanism frequency is reintroduced, the minimum design should sample responses stratified by family, model, budget, and observable state; redact condition metadata that could bias raters; define mutually exclusive and multi-label codes in a codebook; use two independent raters; report raw agreement and an appropriate chance-corrected statistic; adjudicate disagreements; and validate whether labels predict held-out traces or respond to a causal intervention. Until that work is completed, terms such as "sudden divergence" and "drifting hallucination" are theoretical descriptions rather than counted results.

5.4 The Cost of Reliability: Neuro-Symbolic Trade-offs

RQ5 asks whether lightweight tool augmentation mitigates failures and changes structural efficiency. Both parts require measured comparisons.

5.4.1 Structural Cost and Tool Reliability

The historical archive does not establish that Python-assisted decoding mitigates TST failures. Half of the 140 declared groups pair different problems, and those mismatched groups cannot estimate a paired effect. Among the 70 exact-item groups, 10 are leaking chain items. The remaining 60 valid non-chain pairs contain no correct response in either condition. There are therefore no recoveries, no harms, and no discordant successful pairs from which to demonstrate a recovery benefit. An exact McNemar value of 1 in this setting is uninformative; it is not evidence of equivalence or lack of effect.

The process funnel nevertheless shows why tool augmentation can fail before final scoring.

A code-block proxy is visible in 75.0% of eligible with-tool rows, execution is attempted in 74.2%, only 38.2% of attempts succeed, and 64.7% of successes are marked verified. Attrition occurs at request generation, code validity, execution, and verification. A useful system must also integrate a verified result into the final answer, a stage not completely logged historically. This decomposition is more informative than presenting one pooled "tool" accuracy from incomparable rows.

Claims about negligible overhead are also unsupported by the retained data. Latency, retries, prompt tokens, completion tokens, and monetary cost are incomplete or absent. Response length alone cannot measure execution cost, especially across local and provider systems. The corrected answer to RQ5 is therefore: the archive demonstrates the feasibility of recording parts of a tool pipeline and reveals substantial process attrition, but its zero sample recovery does not establish the population effect, and structural cost cannot be estimated.

The sample recovery efficacy is zero in the eligible paired subset because both sample accuracies are zero. This numerical result does not establish equivalence or a population-level absence of benefit. The separate event funnel records execution success of 34/89 attempts and execution failure of 55/89 attempts. Verification of 22/34 successful executions is a process measure rather than verified final-answer recovery. These distinctions preserve the original efficacy question while limiting its answer to the retained evidence.

5.5 Reproducibility in High-Stress Regimes

A high failure rate makes denominator and scoring decisions particularly consequential. The evidence ledger links each outcome to the prompt, ground truth, raw response and source file. Source copies, diagnostic subsets and primary experiments are not interchangeable observations.

5.5.1 The Problem with "Accuracy"

Exact accuracy remains well-defined at zero successes. What deteriorates near the floor is comparative information: a zero-success family cannot distinguish systems on observed success rates. Reporting Wilson intervals makes the uncertainty visible, while response-state counts distinguish an incorrect value from an absent or truncated answer.

The submitted normalised-efficiency ranking depended on unreconciled scores and is not retained. Shorter responses do not demonstrate calibration, knowledge of an error, or cost-effective reasoning. A calibration claim would require a stated confidence measure evaluated against validated outcomes. The distinction between a correct endpoint and sound reasoning also matters: the CAFL literature motivates inspecting reasoning validity separately, while token entropy cannot substitute for that validation (Gupta et al. 2025).

5.6 Broader Implications for Mathematical Reliability

The study concerns mathematical stress tasks, six selected systems and Python support. It does not establish a general limit on Transformer architectures or an implication for artificial

general intelligence. Literature on bounded computation provides theoretical context (Merrill, Sabharwal et al. 2022, Merrill, Sabharwal & Smith 2022); it does not identify the cause of each error in this experiment.

For mathematical applications, the results motivate checking exact answers and recording the stages of tool execution and integration. The potential value of neuro-symbolic systems is supported by prior work (Gao et al. 2022, Yao, Zhao, Yu et al. 2023, Schick et al. 2023), while the present paired archive does not demonstrate their effectiveness. These implications remain conditional on the evaluated tasks and evidence.

5.7 Contributions to Instrumentation

The original instrumentation contributions are TokenPath and the TST benchmark. The scorer, source ledger and statistical checks support their evaluation.

5.7.1 The TokenPath Tracer

TokenPath records available token-level output distributions and associates them with generated text and answer outcomes. Comparable logit-based summaries were retained for GPT-2, Llama and Mistral; the API systems do not provide equivalent measurements in this archive. The tracer provides a basis for investigating generation signals, but neither a clinical diagnostic analogy nor a proven early-warning capability is warranted.

5.7.2 The Token-Stress-Test (TST) Benchmark

TST provides procedural control over numerical, symbolic, repetitive and sequential problems. The audit reconciles seven implemented generators and the six eligible non-leaking families used for primary inference. Extreme failures can characterise a stress regime, while the floor analysis shows where that regime lacks discrimination. Procedural generation reduces direct benchmark reuse but cannot establish that every problem is uncontaminated.

5.8 Limitations of the Study

5.8.1 Historical metadata

The retained archive lacks immutable version dates for some provider models, provider-side tokenisers, complete decoding defaults, hardware and numerical precision for every local run, common random seeds, latency, and monetary cost. Present-day documentation cannot safely reconstruct past runtime state. The model labels should therefore be read as historical system labels with an explicit reproducibility boundary.

5.8.2 Scorer independence

The corrected scorer has extensive automated regression and property checks and a 49-response development review. The latter was not an independent blinded audit. Independent validation across family, model, budget, and response state remains outstanding. Disagreements should be

resolved without changing rules after seeing aggregate model rankings.

5.8.3 Generated-item dependence

Responses to the same or related generated items occur across systems and budgets. Grouped cross-validation limits item leakage, but the approximate Firth intervals are not cluster-robust; this restricts the precision of the reported associations.

5.8.4 Extreme task distribution

Three historical families have severe floors, and the overall 5.54% success rate is dominated by sequence prediction. The benchmark does not sample all mathematical reasoning, and its family mixture is a researcher-selected design. Generalisation to standard curricula, proofs, word problems, or real-world engineering tasks is not established.

5.8.5 Incomplete process and structural measurements

Comparable logits exist for only three models. The historical tool logs do not completely record request validation, retry, execution error, verified integration, latency, or cost. These missing measurements prevent cross-model entropy conclusions and tool-efficacy claims. They also illustrate why instrumentation must be designed before evaluation rather than reconstructed afterwards.

5.8.6 Mechanism validity

The study measures outputs and some token distributions, not hidden causal circuits. Chain-of-thought text may be unfaithful to the actual computation (Lanham et al. 2023, Turpin et al. 2024). Mechanism labels require independent annotation or intervention. The dissertation therefore does not claim that entropy "causes" hallucination, that vocabulary "collapses" at a universal threshold, or that a finite context window is the unique cause of a given wrong answer.

5.9 Future Work

The original three directions remain prospective and are not additional objectives of this dissertation.

5.9.1 Entropy-Aware Decoding

A future evaluation could test whether an entropy threshold improves decisions to revise an answer or invoke a tool. It would first need independently labelled reasoning events and a held-out assessment of calibration, detection error, final accuracy and cost. The present measurements do not validate an intervention threshold.

5.9.2 Integrating Hidden-State Probes

Future work on open-weight systems could compare hidden-state probes with output probabilities to test whether they add information about mathematical errors. Such a study would need independent validation and, for mechanistic claims, interventions on the proposed explanatory components.

5.9.3 Architectural Experiments

The same mathematical tasks could support prospective comparisons of recurrent, state-space, or memory-augmented systems with Transformer systems. Difficulty calibration and matched conditions would be required before attributing differences to architecture. Any new-model experiment constitutes future work and is outside the completed historical evaluation.

5.10 Summary

The discussion relates measured generation signals, task floors, system differences and tool-event outcomes to the original questions. It qualifies the mechanism taxonomy, entropy prediction and mitigation claims, and explains the measurement and sampling limits.

5.11 Conclusion

The framework provides a means of recording and comparing exact mathematical outcomes under finite generation budgets. The historical analysis supports descriptive characterisation, but does not identify a unique internal cause or demonstrate tool-mediated recovery. These limits determine the extent to which the original objectives have been achieved.

Chapter 6: Conclusion

6.1 Introduction

This chapter draws together the four research objectives, the research questions, the key findings, contributions, limitations, and future work. The degree of achievement is assessed against the retained experimental evidence.

6.2 Achievement of Objectives

The four original objectives are assessed as follows:

i. **Objective 1 (Design and Implement TokenPath):** The instrumentation was implemented. It provides response length for the 9,000 primary rows and comparable output-distribution summaries for 4,500 open-weight rows. It does not provide hidden-state measurements or equivalent logit access to every API system.

ii. **Objective 2 (Develop TST Benchmark):** The procedural benchmark was implemented with seven families and 350 generated items per call. The primary analysis uses six non-leaking families after the chain-prompt audit. TST exposes mathematical failures under registered stress, but does not isolate their internal cause; severe floors restrict discrimination in three families.

iii. **Objective 3 (Characterise Failure Modes):** Comparative rates and four observable response states are reported for the tested systems. The stronger intention to validate a taxonomy linking internal structural mechanisms to semantic failures is not fully achieved: the archived mechanism labels lack independent validation.

iv. **Objective 4 (Evaluate Tool-Augmented Decoding):** The retained tool condition was evaluated through exact pairs and execution events. The eligible paired subset has no correct answers in either condition, and consistent latency and cost records are absent. The original intention to establish mitigation efficacy and low overhead is therefore not demonstrated by this archive.

The evidence ledger, corrected scorer and rare-event analysis support these four assessments. They are analytical procedures rather than additional research objectives.

6.3 Answers to Research Questions

Main research question

An instrumented TokenPath–TST framework can trace prompts, outputs, budgets, exact outcomes, and—where interfaces expose logits—token-distribution summaries. It can characterise failures by registered family, output state, model, and budget, and it can enforce an auditable contract for tool evaluation. The historical framework identifies FSC-consistent outcomes but does not

identify their unique internal cause. Its tool archive demonstrates pipeline attrition rather than successful mitigation.

RQ1: Token-level tracing

Token-level probabilities, entropy, and effective vocabulary can be captured for local open-weight models and visualised with response length and outcome. The same metrics cannot be assumed for black-box provider systems. Entropy is a candidate uncertainty signal, not a validated detector of logical error or hallucination.

RQ2: Stress-induced failures

Large exact representations, combinatorial values, long repetitions, symbolic expansion, arithmetic sequences, and restricted output budgets generate observable stress. The historical distributions expose failure but three families are too close to zero for useful comparative discrimination. TST should therefore use a validated solver, non-leaking prompts, stable IDs and seeds, explicit budgets, and calibrated tiers.

RQ3: Model-level characterisation

The tested systems differ in pooled accuracy and, more importantly, in model–family profiles. Those differences are conditional on the registered task mixture and historical interfaces. A four-state observable taxonomy is reproducible; the submitted mechanism taxonomy is not independently validated.

RQ4: Structural–semantic linkages

Correctness is strongly associated with output budget and varies with family, model, and reconstructed within-family difficulty. Realised length provides weaker adjusted evidence. Logit-derived metrics are available only for three systems and are exploratory. These findings support prediction within the historical design, not a causal claim about Transformer internals.

RQ5: Tool-augmented mitigation

The 60 eligible exact pairs contain zero correct answers in either condition, so the sample recovery efficacy is zero. This floor does not establish a general absence of tool benefit. Execution failures and incomplete latency and usage records restrict conclusions about reliability and structural efficiency.

6.4 Summary of Key Findings

Three findings summarise the evidence relevant to the original study:

First, exact-answer performance is low but varies by task, system and budget. The primary count is 499/9,000, or 5.54% (Wilson 95% CI 5.09–6.04%). Most successes occur at the larger output budgets. Family-level rates distinguish useful variation from near-zero floors.

Second, generation measurements and semantic outcomes must be interpreted separately. Comparable entropy and effective-vocabulary summaries are available for three open-weight systems. Observable output states are reproducible; the historical records do not validate causal mechanism classes or an entropy threshold for predicting errors.

Third, the retained tool archive limits the mitigation conclusion. Sixty eligible exact pairs have zero correct answers in each condition, giving sample recovery efficacy zero. The separate event funnel records execution and verification losses. These observations do not establish a general absence of tool benefit or a low-overhead mitigation effect.

6.5 Research Contributions

The contributions remain within the original framework and instruments.

6.5.1 Theoretical Contribution: The FSC Framework

FSC organises questions about finite token representations and generation resources. Its operational rule identifies failed outcomes under registered stress without identifying a unique internal cause. This distinction connects the theoretical motivation to measurable endpoints while limiting the claims made from those endpoints.

6.5.2 Methodological Contribution: TokenPath and TST

TokenPath provides trace capture and TST provides procedural mathematical stress tasks. The corrected cohort accounting, scoring rules, reference solvers, uncertainty estimates and pairing checks make their evaluation traceable to retained records. They support the original instrumentation and benchmark contributions without expanding the research objectives.

6.6 Broader Implications

The implications concern mathematical reliability and the evaluation of the selected systems.

AI Safety and Reliability

Exact answers require verification even when explanations are fluent. Entropy may be investigated as a monitoring signal, but the present study does not establish a safe trigger for intervention or abstention.

Evaluation Standards

Counts, denominators, scoring rules and measurement availability make failure results interpretable. Extreme task distributions should be distinguished from calibrated comparisons; a floor should not be presented as evidence of equal capability.

Hybrid Architectures

Python-assisted reasoning remains an intervention worth evaluating through matched tasks, complete execution events, integrated final answers and cost measurements. The retained archive supplies a limited process evaluation and does not establish an architectural solution.

Contribution to UNISA's 4IR and Digitalisation Agenda

The study supports research on reliable digital systems through transparent evaluation of mathematical language-model outputs. Its instrumentation and benchmark provide practical resources for examining errors and the limits of automated computation in engineering and educational applications.

6.7 Limitations

The study is limited to generated exact-answer tasks and six historical systems. Three families have severe floors; provider metadata and decoding details are incomplete; logits are unavailable for half the primary cohort; item responses are repeated; approximate Firth intervals are not cluster-robust; and the scorer's preliminary manual review is not an independent blinded validation. Historical tool logs lack complete event, latency, usage, and cost records.

6.8 Future Work

The original future-work directions follow from the measurement and inference limits.

6.8.1 Real-Time Adaptive Computation

A future study could test whether output-distribution signals improve adaptive decisions to use tools or request review. Such decisions require validated thresholds and an evaluation of false alarms, missed errors, final accuracy and cost.

6.8.2 Architectural Exploration

Recurrent, state-space, memory-augmented and tool-assisted systems could be compared on calibrated TST tasks under matched conditions. These comparisons would test new hypotheses and are not part of the completed dissertation experiments.

6.8.3 Internal State Transparency

For open-weight models, hidden-state probes could be evaluated alongside output probabilities. Independent reasoning annotations and targeted interventions would be needed to connect a predictive signal to an internal mechanism.

6.9 Concluding Remarks

TokenPath and TST support trace capture, benchmark construction and comparative outcomes. The retained archive does not independently validate the mechanism taxonomy or demonstrate effective low-overhead mitigation.

References

Baker, B., Huizinga, J., Gao, L., Dou, Z., Guan, M. Y., Madry, A., Zaremba, W., Pachocki, J. & Farhi, D. (2025), 'Monitoring reasoning models for misbehavior and the risks of promoting obfuscation', *arXiv preprint arXiv:2503.11926* .
URL: https://arxiv.org/abs/2503.11926

Beltagy, I., Peters, M. E. & Cohan, A. (2020), Longformer: The long-document transformer, *in* 'Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics'.
URL: https://arxiv.org/abs/2004.05150

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020), Language models are few-shot learners, *in* 'Advances in Neural Information Processing Systems', Vol. 33, pp. 1877–1901.

Chiang, D. & Cholak, P. (2023), 'Tokenization counts: The impact of tokenization on arithmetic in large language models', *arXiv preprint arXiv:2310.09404* .
URL: https://arxiv.org/abs/2310.09404

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Dean, J. (2022), 'PaLM: Scaling language modeling with pathways', *arXiv preprint arXiv:2204.02311* .

Cobbe, K., Kosaraju, V., Bavarian, M., Hilton, J., Nakano, R., Hesse, C. & Schulman, J. (2021), 'Training verifiers to solve math word problems', *arXiv preprint arXiv:2110.14168* .

Dettmers, T., Lewis, M., Belkada, Y. & Zettlemoyer, L. (2022), 'Llm.int8(): 8-bit matrix multiplication for transformers at scale', *arXiv preprint arXiv:2208.07339* .

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jian, L., Lin, B. Y., West, P., Bhagavatula, C., Bras, R. L., Hwang, J. D. et al. (2023), 'Faith and fate: Limits of transformers on compositionality', *arXiv preprint arXiv:2305.18654* .

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Chen, A., Lieberum, T., Bokum, L., Jones, D., Amodei, D., Kaplan, S. & Mueller, C. (2021), 'A mathematical framework for transformer circuits', *Transformer Circuits Thread* .
URL: https://transformer-circuits.pub/2021/framework/index.html

Firth, D. (1993), 'Bias reduction of maximum likelihood estimates', *Biometrika* **80**(1), 27–38.

Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y. & Callan, J. (2022), 'PAL: Program-aided language models', *arXiv preprint arXiv:2211.10435* .

Geva, M., Schuster, R., Berant, J. & Levy, O. (2021), Transformer feed-forward layers are key-value memories, *in* 'Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics.

Gupta, A., Reddig, J., Calò, T. & et al. (2025), 'Beyond final answers: Evaluating large language models for math tutoring', *arXiv preprint arXiv:2503.16460* .

Hao, S., Su, D. & et al. (2024), 'Reasoning with continuous intermediate thoughts', *arXiv preprint arXiv:2404.09559* .

Heinze, G. & Schemper, M. (2002), 'A solution to the problem of separation in logistic regression', *Statistics in Medicine* **21**(16), 2409–2419.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E. & Steinhardt, J. (2021), Measuring mathematical problem solving with the MATH dataset, *in* 'International Conference on Learning Representations'.

Ji, Z. et al. (2023), 'Survey of hallucination in natural language generation', *ACM Computing Surveys* .
URL: https://arxiv.org/abs/2307.06290

Kadavath, S., Conerly, T., Askell, A. et al. (2022), 'Language models (mostly) know what they know', *arXiv preprint arXiv:2207.05221* .
URL: https://arxiv.org/abs/2207.05221

Kojima, T., Gu, S. S., Reid, M., Yasunaga, M., Isola, P. & Le, Q. V. (2022), 'Large language models are zero-shot reasoners', *arXiv preprint arXiv:2205.11916* .

Kudo, T. & Richardson, J. (2018), Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, *in* 'Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations', pp. 66–71.
URL: https://aclanthology.org/D18-2012/

Kuhn, L., Gal, Y. & Farquhar, S. (2023), 'Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation', *arXiv preprint arXiv:2302.09664* .
URL: https://arxiv.org/abs/2302.09664

Kumar, H., Rothschild, D., Goldstein, D. & Hofman, J. (2024), 'Math education with large language models: Peril or promise?', *SSRN* .

Lanham, T., Chen, A., Radhakrishnan, A., Steiner, B., Denison, C., Hernandez, D., Li, D., Durmus, E., Hubinger, E., Kernion, J. et al. (2023), 'Measuring faithfulness in chain-of-thought reasoning', *arXiv preprint arXiv:2307.13702* .

Lewkowycz, A., Dohan, D., Dyer, E., Andreassen, A. et al. (2022), Solving quantitative reasoning problems with language models, *in* 'Advances in Neural Information Processing Systems', Vol. 35, pp. 4930–4944.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F. & Liang, P. (2024), 'Lost in the middle: How language models use long contexts', *Transactions of the Association for Computational Linguistics* **12**, 157–173.

McCoy, T., Pavlick, E. & Linzen, T. (2019), Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference, *in* 'Proceedings of ACL 2019'.
URL: https://aclanthology.org/P19-1334/

Merrill, W., Sabharwal, A. & Smith, N. A. (2022), 'The saturation of transformers', *arXiv preprint arXiv:2206.00621* .

Merrill, W., Sabharwal, A. et al. (2022), Transformer networks as bounded rational processors, *in* 'Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)'.

Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S. & Farajtabar, M. (2024), 'Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models', *arXiv preprint arXiv:2410.05229* .

Nanda, N., Chan, C., Olah, C. et al. (2023), 'Progress measures for grokking via mechanistic interpretability', *arXiv preprint arXiv:2301.05217* .
URL: https://arxiv.org/abs/2301.05217

Nanda, N. et al. (2023), An interpretability illusion for a modular addition task, *in* 'International Conference on Learning Representations'.

Olsson, C. et al. (2022), 'In-context learning and induction heads', *Transformer Circuits Thread* .
URL: https : / / transformer-circuits . pub / 2022 / in-context-learning-and-induction-heads/index.html

OpenAI (2023), Gpt-4 technical report, Technical report, OpenAI.

Press, O., Smith, N. A. & Lewis, M. (2021), Train short, test long: Attention with linear biases enables input length extrapolation, *in* 'Proceedings of the 38th International Conference on Machine Learning (ICML)'.
URL: https://arxiv.org/abs/2107.02277

Quirke, P. & Barez, F. (n.d.), 'Understanding addition in transformers'.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L. & Scialom, T. (2023), Toolformer: Language models can teach themselves to use tools, *in* 'International Conference on Learning Representations, Workshop Track'.

Schuster, M. & Nakajima, K. (2012), Japanese and korean voice search, *in* '2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 5149–5152.

Sennrich, R., Haddow, B. & Birch, A. (2016), Neural machine translation of rare words with subword units, *in* 'Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)', pp. 1715–1725.
URL: https://aclanthology.org/P16-1162/

Singh, A. K. & Strouse, D. J. (2024), 'Tokenization counts: The impact of tokenization on arithmetic in frontier llms', *arXiv preprint arXiv:2402.14903* .

Srivastava, A., Rastogi, A., Rao, A. et al. (2022), 'Beyond the imitation game: Quantifying and extrapolating the capabilities of language models', *arXiv preprint arXiv:2206.04615* .
URL: https://arxiv.org/abs/2206.04615

Stolfo, A. et al. (2023), 'A mechanistic interpretation of arithmetic reasoning in large language models', *arXiv preprint arXiv:2305.15386* .

Suzgun, M., Scales, N., Schärli, N. et al. (2022), 'Challenging BIG-bench tasks and whether chain-of-thought can solve them', *arXiv preprint arXiv:2210.09261* .
URL: https://arxiv.org/abs/2210.09261

Turpin, M., Michael, J., Sharma, A., Stuhlmüller, A. & Steinhardt, J. (2024), 'Measuring faithful reasoning in large language models', *arXiv preprint arXiv:2403.05530* .

Turpin, M. et al. (2023), 'Language models don't always need chain of thought reasoning', *arXiv preprint arXiv:2305.00418* .
URL: https://arxiv.org/abs/2305.00418

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. & Polosukhin, I. (2017), Attention is all you need, *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett, eds, 'Advances in Neural Information Processing Systems', Vol. 30, Curran Associates, Inc.
URL: https://papers.nips.cc/paper/7181-attention-is-all-you-need

Wang, X. et al. (2023), 'Self-consistency improves chain of thought reasoning in large language models'.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Le, Q. V. & Zhou, D. (2022), 'Chain-of-thought prompting elicits reasoning in large language models'.

Wilson, E. B. (1927), 'Probable inference, the law of succession, and statistical inference', *Journal of the American Statistical Association* **22**(158), 209–212.

Xiao, G., Tian, Y., Chen, B., Han, S. & Lewis, M. (2023), 'Efficient streaming language models with attention sinks'.

Yao, S., Zhao, J., Yu, D. et al. (2023), React: Synergizing reasoning and acting in language models, *in* 'NeurIPS'.

Yao, S. et al. (2023), 'Tree of thoughts: Deliberate problem solving with large language models'.

Zhou, D., Le, Q. V. et al. (2022), 'Least-to-most prompting enables complex reasoning in large language models'.

Appendices

Appendix A: Supplementary Results and Statistical Analysis

A.1 File Inventory and Source Selection

The inventory script recursively enumerates retained CSV, JSON, JSONL, log, configuration, code, and manuscript artefacts; records path, size, hash, schema, and parse status; and groups byte-identical files. The initial inventory contains 573 files, 209,862,326 bytes, 119,696 CSV row occurrences, and 27 CSV schemas. It identifies 45 identical file copies, nine within-file duplicate rows, and no CSV parser failures.

The following hierarchy governs selection:

i. an unscored file containing prompt, truth, and raw response is preferred to a scored derivative;
ii. a row-level source is preferred to a roll-up table;
iii. a byte-identical copy is excluded after one canonical path is selected;
iv. diagnostics are retained under their own cohort and never silently pooled with the primary budget sweep;
v. chain-reasoning rows are preserved but excluded from primary inference because historical prompts reveal intermediate targets;
vi. every exclusion records a machine-readable reason rather than being removed from the archive.

Forty-six unscored files satisfy these criteria. They contain 23,345 row occurrences. Three exact copies are excluded, leaving 23,342 primary occurrences before eligibility rules. Across primary and diagnostic cohorts, 20,007 non-chain occurrences are eligible. The canonical ledger is identified by SHA-256

`8fcf38e89dc51d32186882661fe2fa79f75e442cd9134a9c9f1088a454fd0479`.

A.2 Cohort Accounting

Table A.1: Canonical cohort accounting. "Eligible" excludes leaking chain rows and exact copies according to the cohort rule.

Cohort	Archived	Eligible	Correct	Accuracy	Purpose
Diagnostic baseline validation subset	385	330	22	6.67%	scorer and trace diagnostics
Diagnostic tool validation subset	630	540	20	3.70%	diagnostic tool-prompt runs

Cohort	Archived	Eligible	Correct	Accuracy	Purpose
Historical baseline budget sweep	10 500	9 000	499	5.54%	primary descriptive and regression cohort
Historical tool-prompt budget sweep	11 550	9 897	499	5.04%	unpaired historical prompt cohort
Paired tool ablation	280	240	0	0.00%	input to exact-item identity gate

The baseline $N = 9\,000$ is an invariant: each of six models contributes 1,500 observations; each of six families contributes 1,500; each of five budgets contributes 1,800; and each of the 36 model–family cells contributes 250. These four independent sums are tested by the ledger build. The unpaired tool-prompt cohort has a different denominator and system composition and is never subtracted from or divided into the baseline as if it were paired.

A.3 Full Model–Family Correct Counts

Table A.2 gives all 36 cell numerators. Every denominator is 250. The table is the numerical companion to the Chapter 4 heat map.

Table A.2: Correct answers in each historical model–family cell; every cell has $N = 250$. BC: base conversion; CP: combinatorics; LA: large-number arithmetic; RS: repetitive sequences; SP: sequence prediction; SE: symbolic expansion.

Model	BC	CP	LA	RS	SP	SE	Total
Claude 3.7 Sonnet	0	0	34	1	61	0	96
Gemini 2.5 Flash	0	0	2	0	48	50	100
GPT-4o mini	0	0	2	0	43	50	95
GPT-2	0	0	0	0	0	0	0
Llama 3.1 8B Instruct	0	0	0	1	96	36	133
Mistral 7B Instruct v0.3	0	0	0	0	75	0	75
Total	0	0	38	2	323	136	499

Wilson 95% intervals for a zero-event cell with $N = 250$ are 0.00–1.51%. Selected non-zero cells are: Claude large-number arithmetic, 13.60% (9.90–18.40%); Claude sequence prediction, 24.40% (19.49–30.09%); Gemini and GPT-4o mini symbolic expansion, 20.00% (15.51–25.40%); Llama sequence prediction, 38.40% (32.59–44.56%); Llama symbolic expansion, 14.40% (10.59–19.29%); and Mistral sequence prediction, 30.00% (24.66–35.95%). The full

machine-readable table contains an interval for every cell.

A.4 Rare-Event Regression Details

The primary Firth fit uses 9,000 rows, 499 events, and 14 parameters including the intercept. GPT-4o mini is the model reference and sequence prediction is the family reference. Continuous predictors are standardised after transformation. Table A.3 reports the full coefficient set.

Table A.3: Full Firth-penalised model. Intervals are approximate penalised-Wald intervals.

Term	Coefficient	Adjusted OR	Approx. 95% CI
$\log_2$ output budget (1 SD)	4.055	57.7	31.9–104
log(1+ realised length) (1 SD)	0.395	1.48	0.799–2.76
Within-family log(1+ difficulty) (1 SD)	−0.145	0.865	0.751–0.995
Claude 3.7 Sonnet	0.116	1.12	0.698–1.81
Gemini 2.5 Flash	0.256	1.29	0.791–2.11
GPT-2	−7.193	0.000752	4.43×10^{-5}–0.0128
Llama 3.1 8B Instruct	0.968	2.63	1.69–4.11
Mistral 7B Instruct v0.3	−0.466	0.627	0.375–1.05
Base conversion	−8.951	0.000130	7.92×10^{-6}–0.00212
Combinatorics	−8.917	0.000134	8.21×10^{-6}–0.00219
Large-number arithmetic	−4.369	0.0127	0.00761–0.0211
Repetitive sequences	−7.203	0.000744	0.000205–0.00270
Symbolic expansion	−2.425	0.0885	0.0606–0.129

The model has approximately 35.6 events per parameter when calculated from the overall event count, but that summary does not remove separation in individual factor levels. Ordinary maximum-likelihood logistic regression produces non-finite estimates. A ridge model with penalty 0.5 converges and is retained as a sensitivity check. The Firth fit converges to finite values and grouped five-fold validation by item fingerprint yields AUC 0.9841 and Brier score 0.02049. Mean predicted probability is 0.05570 versus observed 0.05544.

The high AUC should not be overinterpreted. Budget and family nearly identify the rare success region, so discrimination is easier than causal explanation. The coefficient for realised length is post-generation. The intervals are not item-cluster robust. Extreme finite coefficients in zero-event levels are a regularised representation of separation rather than precisely estimated effects. These limitations accompany the table wherever it is cited.

A.5 Historical Tool-Pair Audit

The historical ablation contains 280 rows grouped into 140 declared pairs. For each declared pair, the audit normalises prompt whitespace and truth representation and computes an item fingerprint. A group is valid only when it contains one baseline and one tool row with identical fingerprints. Seventy groups satisfy this condition and 70 contain different items. Ten of the 70 exact groups belong to historical leaking chain reasoning, leaving 60 eligible pairs.

Table A.4: Historical tool-pair identity gate and exact outcomes.

Audit quantity	Count
Declared groups	140
Exact same-item and same-truth groups	70
Mismatched-item groups excluded	70
Leaking chain pairs excluded	10
Eligible exact non-chain pairs	60
Baseline correct	0
Tool correct	0
Tool-only recoveries	0
Baseline-only harms	0

The paired difference is 0.00 percentage points. With zero discordant successes, exact McNemar $p = 1$. This result is not an estimate that tools are ineffective. It indicates that the historical valid-pair task region is a complete outcome floor.

A.6 Historical Tool Event Funnel

The secondary funnel uses 120 eligible with-tool rows, regardless of whether their declared pair passes the exact identity gate. Ninety contain a code-block proxy; 89 have an execution attempt; 34 of the attempts succeed; and 22 of the successful executions are marked verified. Thus, the rates are 90/120 (75.0%), 89/120 (74.2%), 34/89 (38.2%), and 22/34 (64.7%), respectively.

The changing denominator is part of the estimand. Reporting 22/120 as a generic "tool success" would combine request, execution, and verification stages and still omit final answer integration. Likewise, comparing the 9,897-row tool-prompt sweep to the 9,000-row baseline is not a paired effect. The corrected dissertation reports each object separately.

Appendix B: TokenPath Tracer Implementation

B.1 Trace Schema

TokenPath stores the evidence necessary to connect a response to a model call and its derived metrics. The canonical schema includes:

i. source path, source row, file hash, and cohort;
ii. raw and normalised model identifier and condition;
iii. family, budget, prompt, ground truth, and raw response;
iv. normalised item fingerprint and declared pair identifier where present;
v. extracted answer, exact-correct flag, observable output state, and scorer version;
vi. realised length and, when exposed, token probabilities, entropy, and effective vocabulary;
vii. tool-event fields where retained; and
viii. eligibility, exclusion reason, and duplicate status.

The schema separates a missing value from zero. For example, API entropy is `unavailable`, not 0. A missing cost is `unavailable`, not free. A failed generation remains an eligible incorrect endpoint when the item and ground truth are valid.

B.2 Entropy and Effective Vocabulary

For retained full-vocabulary logits $z_{t,v}$ at generation step t, probabilities are computed using a stable softmax:

$$m_t = \max_v z_{t,v}, \tag{B.1}$$

$$p_{t,v} = \frac{\exp(z_{t,v} - m_t)}{\sum_u \exp(z_{t,u} - m_t)}, \tag{B.2}$$

$$H_t = -\sum_v p_{t,v} \log p_{t,v}, \tag{B.3}$$

$$V_t^{\text{eff}} = \exp(H_t). \tag{B.4}$$

Subtracting m_t avoids avoidable overflow. Padding and absent steps are masked. Response-level summaries record their aggregation rule. Effective vocabulary is a deterministic transformation of entropy and is not treated as independent confirmatory evidence.

The tracer does not infer that a high H_t is an error. It records H_t so that a separate study can align it with validated semantic or reasoning events. Comparable distributions were retained only for GPT-2, Llama 3.1 8B Instruct, and Mistral 7B Instruct v0.3.

B.3 Task-Specific Scoring Contract

The scoring function receives family, problem, stored truth, raw response, and budget metadata. It returns eligibility, extracted candidate, correctness, output state, and an audit explanation. The decision order is important:

i. validate the family and stored truth with the reference solver;
ii. detect recorded generation error and truncation state;
iii. apply the family-specific extractor to the response rather than the prompt;
iv. normalise the extracted candidate and truth under the family rules;
v. compare exactly or through a registered symbolic normal form;
vi. keep missing, refused, malformed, and truncated answers in the denominator;
vii. exclude only invalid truth, unknown family, or rows outside the cohort rule.

B.3.1 Family normalisation

Large-number arithmetic accepts signed arbitrary-precision integers and rejects decimal approximations for an integer truth. Base conversion preserves the requested base and permits lower-case hexadecimal without a prefix, matching the historical generator. Combinatorics requires the exact integer. Repetitive sequences compare the requested repeated structure or exact count according to the prompt form. Sequence prediction extracts the requested next term rather than any number in an explanation. Symbolic expansion maps expressions to the coefficient triple (a_2, a_1, a_0) and compares the exact polynomial after normalisation.

B.3.2 Observable response states

The mutually exclusive states are:

Correct: the family-normalised candidate equals the validated truth.

Incorrect final answer: a valid candidate is extracted but differs from truth.

No valid final answer: no family-valid candidate is found and no registered truncation flag applies.

Truncated/no valid final answer: no valid candidate is found and the response reaches the recorded budget/termination criterion for truncation.

The binary endpoint maps only the first state to one. The remaining three map to zero. This prevents selective deletion of non-answers.

B.4 Scorer Verification

Thirteen regression and property checks cover large signed integers; 500 random arbitrary-precision integer operations; incidental prompt numbers; decimal and comma formatting; base

representations; symbolic term ordering; coefficient omission; arithmetic sequences; repeated-character outputs; multiple competing final values; refusal language; malformed output; and truncation. Every check passes in the frozen scorer audit.

A 49-response development review also agrees with the automatic score in all 49 cases. Because the reviewer participated in the correction and was not blinded, this is not described as independent human validation. Independent assessment of a stratified review sample remains a limitation. A defensible review sample should include each model, family, budget, and observable state, with oversampling of rare correct and ambiguous symbolic responses.

B.5 Why the Historical Mechanism Table Was Withdrawn

The legacy scoring files contain exactly 9,640 mechanism-style labels: 3,260 pattern break, 3,260 algebraic incomplete, 1,829 magnitude drift, and 1,291 wrong operation. Code inspection and field audit show that these values are deterministic parser fallbacks. There are no rater IDs, codebook versions, blinded assignments, adjudication fields, or agreement statistics. The labels therefore do not meet the evidential standard for a mechanism frequency table.

The replacement four-state taxonomy is deliberately less ambitious and more reproducible. Terms such as sudden divergence, drifting hallucination, circuit failure, context saturation, attention sink, and working-memory instability may be used to motivate future qualitative or causal studies, but not as observed row labels in the historical results.

B.6 Tracer Class Interface

The following schematic illustrates the original tracer interface. It computes entropy in bits; the retained natural-log summaries described above use nats. This listing is explanatory and does not replace the versioned analysis implementation.

```
class TokenPathTracer:
    def __init__(self, model, tokenizer):
        self.model = model
        self.tokenizer = tokenizer
        self.entropy_trace = []

    def hook_fn(self, module, input, output):
        # Extract logits from the final layer
        logits = output.logits[:, -1, :]

        # Compute softmax probabilities
        probs = torch.softmax(logits, dim=-1)

        # Calculate Shannon entropy
        # Note: Computed on output distribution, not internal state
```

```
        entropy = -torch.sum(probs * torch.log2(probs + 1e-9), dim
            =-1)

        # Store scalar value
        self.entropy_trace.append(entropy.item())

    def generate_and_trace(self, prompt, max_new_tokens=512):
        # Register hook
        handle = self.model.register_forward_hook(self.hook_fn)

        # Run generation
        inputs = self.tokenizer(prompt, return_tensors="pt")
        outputs = self.model.generate(**inputs, max_new_tokens=
            max_new_tokens)

        # Cleanup
        handle.remove()
        return self.entropy_trace
```

B.7 Computation and Reproducibility

Table B.1: Historical reproducibility fields.

Field	Status	Evidence or limitation
Prompt and raw response	Recorded	retained in selected unscored source files
Ground truth and family	Recorded	checked against authoritative family solver
Output budget and condition	Recorded	canonical ledger field; balanced baseline verified
Model label	Recorded	raw and normalised labels retained
Immutable model version	Partial	exact identifiers for some systems; unavailable for some historical API calls
Access date	Partial	retained only where logs contain a recoverable date
Tokenizer	Partial	recoverable for local open-weight models; provider tokenizer/version not reconstructed
Decoding defaults	Partial	explicit fields retained where present; absent provider settings marked unavailable
Hardware and numerical precision	Partial	local environment evidence is incomplete across historical files

Field	Status	Evidence or limitation
Random seed and stable item ID	Unavailable historically	fingerprints derived during correction; historical generator emitted no seed/ID field
Token distributions	Partial	complete for 4,500 open-weight baseline rows; unavailable for API rows
Usage tokens	Partial	retained for GPT-4o mini historical baseline only
Latency and retries	Incomplete	not used for an overhead claim
Monetary cost	Unavailable	not estimated retrospectively
Source and analysis hashes	Recorded in correction	inventory, ledger, tables, figures, and sidecars hashed

B.8 Limitations

Output-distribution measurements do not expose hidden-state or attention-head causes. Scalar aggregation can conceal local changes. API systems without logits do not support the same tracing interface. Scorer development checks are not independent human validation.

Appendix C: TST Benchmark Protocols

C.1 Generation Schema

The historical generator returns only two fields, `problem` and `answer`. It does not return a stable ID, seed, difficulty, constraints object, or distractor list. The implemented family rules are reproduced in Chapter 3. The audit generated 3,500 validation items and compared parameter bounds, parseability, and reference answers against retained prompts. Seven families at 50 items per family were implemented, yielding 350 items per generator call.

The following submitted descriptions are corrected:

i. the generator uses seven rather than five families;
ii. the historical primary analysis uses six non-leaking families, not five;
iii. combinatorics samples n from 20 to 100, not 80 to 120;
iv. repetitive items do not contain generated distractors;
v. no historical difficulty clipping field is emitted;
vi. stable IDs and a public seed API are not part of the historical generator; and
vii. chain-reasoning prompts expose intermediate targets and are excluded from primary inference.

C.2 Stress Parameters and Constraint Interaction

Table 3.1 is the single authoritative parameter table. Magnitude, representation length, repetition and sequential operations may impose several demands simultaneously; family labels therefore do not identify isolated internal mechanisms. Historical difficulty proxies are reconstructed from prompt parameters and are explicitly retrospective.

C.3 Validation Protocol

The generator audit checked 3,500 generated items against parameter bounds and reference solvers. The primary evaluation excludes historical chain prompts that expose intermediate targets. Further study would need calibrated task difficulty and non-leaking prompts; no additional model experiment is claimed in this dissertation.