

Automated Taxonomy Discovery for Interpretable Diagnosis of Multi-Hop Question Answering Datasets

Ryoji Osaki¹ Masaki Asada² Makoto Miwa^{1,2}

¹Toyota Technological Institute, Japan

²National Institute of Advanced Industrial Science and Technology (AIST), Japan

{sd26411, makoto-miwa}@toyota-ti.ac.jp masaki.asada@aist.go.jp

Abstract

As Large Language Models (LLMs) improve, multi-hop question answering (QA) has become a standard testbed. However, automated analyses of these datasets remain largely structural (chain depth, branching), and a dataset’s own design labels predict almost none of the variance in whether an LLM answers correctly. This paper introduces a framework for the automated, multi-perspective analysis of QA datasets and its application to diagnosing LLM answers. An Analyst LLM is utilized to (1) generate explicit sequential steps from question–context–answer triples in a zero-shot setting, retaining only sequences that re-derive the gold answer, and (2) synthesize a taxonomy of reasoning types over seven fixed analytical perspectives spanning educational, logical, and professional lenses. The resulting per-question binary features are then used to (3) identify feature combinations *associated* with Target LLM failures, using interpretable models with permutation and Fisher exact tests. Applied to FRAMES, Multihop-RAG, and MuSiQue, the analysis surfaces dataset-specific bottlenecks that neither structural analysis nor the datasets’ own labels expose: Implicit Math in FRAMES, where questions require calculation without explicit operand extraction, Conditioned Filtering in Multihop-RAG, and a Temporal Order Logic Paradox in MuSiQue.

1 Introduction

Multi-hop Question Answering (QA) (Yang et al., 2018) is a benchmark task that requires aggregating information across multiple documents and synthesizing a final answer. To effectively evaluate LLMs on these datasets and improve them, it is necessary to understand the datasets’ characteristics to identify which reasoning types are associated with errors in LLM outputs. However, QA datasets typically do not annotate the explicit sequential reasoning steps required for each query. Furthermore,

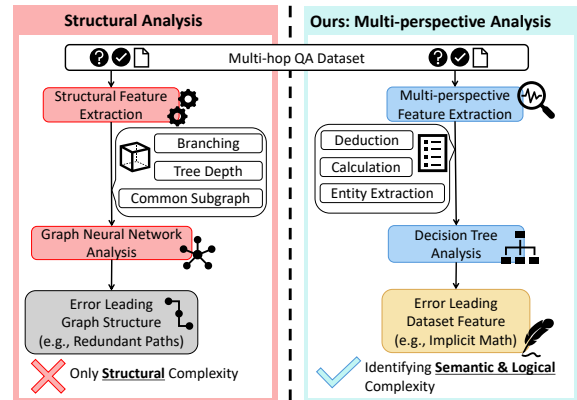


Figure 1: An overview of dataset analysis considering contents of reasoning.

defining criteria to analyze these latent steps is inherently challenging, leading to high variance and subjectivity among human evaluators.

Recent efforts have introduced automated methods for analyzing sequential steps. For instance, Liu et al. (2025) used LLMs to generate reasoning structures retroactively, and Jiang et al. (2025) analyzed structural complexity by converting Chain-of-Thought (CoT) (Wei et al., 2022) outputs into tree graphs. While these approaches discover some structural features, they largely ignore the diverse features from multiple perspectives contained within the sequential steps, such as differences in inference types (e.g., deduction vs. induction), domain-specific analytical perspectives (e.g., cognitive psychologist, data scientist), and logical fallacies (e.g., hasty generalization, red herring). Analyzing these steps from multiple perspectives is required to diagnose output errors, rather than relying solely on structural features.

The labels provided with a dataset do not fill this gap either. A benchmark’s own reasoning-type or hop-count annotations state what the dataset was *built* to test, and we find that they explain almost none of the variance in whether a Target

LLM answers correctly: $R^2 = 0.023$ on FRAMES, and pseudo- $R^2 \leq 0.008$ on MuSiQue and on Multihop-RAG once answer format is controlled (Section 5.5). Consider a FRAMES question asking for the number of days between a film’s release and its cinematographer’s death. Its design label (numerical reasoning) covers a large, mostly easy set, whereas the distinction our analysis discovers—whether the operands must be inferred rather than extracted from the context—separates 77% accuracy from 48% among the questions our taxonomy marks as requiring calculation (Section 5.2; this question is the failure audited in Appendix G).

We propose an automated multi-perspective analysis framework for QA datasets and apply it to diagnose LLM answering results (Figure 1). We hypothesize that different analytical lenses—cognitive demand, inferential structure, domain-specific operational requirements—surface complementary failure correlates that no single perspective captures alone. We use an Analyst LLM to derive a taxonomy over seven such perspectives, spanning educational theory (Bloom’s Taxonomy (Bloom, 1956)), formal logic, and professional domain personas, and treat the resulting classifications as features for interpretable models that isolate the reasoning types associated with LLM failure.

Our contributions are as follows:

- **Framework.** An automated method to generate and classify sequential steps from seven analytical perspectives, yielding per-question reasoning-requirement features that structural approaches do not capture (Section 3).
- **Diagnosis.** Interpretable models over these features isolate feature *combinations* associated with Target LLM failure, such as Implicit Math on FRAMES and the corresponding patterns in Multihop-RAG and MuSiQue (Section 5.2).
- **Validation.** We show that the association is reproduced across Targets and Judges but is sensitive to the Analyst: varying the Target, Judge, and Analyst roles (Section 5.3), human validation of the generated steps and labels (Section 4.4), and a comparison against the datasets’ own design labels (Section 5.5).

Throughout, we report *associations* between properties of questions and Target failures: the steps we analyze reconstruct what a question requires rather than transcribing the Target’s own

computation, so we do not claim that a feature caused a particular failure (Section 3.1; Limitations). Our code, the generated taxonomies and their definitions, all prompts, and the per-question features, correctness labels, and fitted trees for every (Analyst, Target, Judge) configuration are released at <https://github.com/zwititors/blackboxnlp2026-taxonomy-diagnosis>.

2 Related Work

Multi-Hop QA Benchmarks and Shortcut Learning. Multi-hop QA benchmarks evaluate the ability of models to aggregate information across documents (Yang et al., 2018). Early datasets like HotpotQA exhibited shortcut learning, where correct outputs are generated via entity co-occurrence or pattern matching rather than multi-step reasoning (Min et al., 2019); 2WikiMultiHopQA (Ho et al., 2020) and MuSiQue (Trivedi et al., 2022) were designed to enforce connected sequential steps, and FRAMES (Krishna et al., 2025) targets the temporal and numerical dependencies that defeat a simple retrieval strategy. These efforts specify difficulty at *construction* time through a small set of design categories; our work instead recovers a fine-grained requirement profile for each individual question after the fact, and Section 5.5 shows that the two are far from equivalent as predictors of model failure.

Automated Analysis and Taxonomy Generation. Analyzing LLM reasoning has traditionally relied on manual annotation or static question types, and recent work automates this with LLMs. LLM-as-a-Judge approaches (Zheng et al., 2023) evaluate response quality but are subject to biases such as position and self-preference (Wataoka et al., 2024); we therefore treat correctness labels as a varied component of the pipeline rather than as ground truth (Section 5.3). For multi-hop tasks, Liu et al. (2025) introduced PER, generating sequential steps to evaluate step-level correctness, and Jiang et al. (2025) proposed LCoT2Tree, converting CoT logs into tree structures to quantify structural complexity. These methods measure step correctness and structural complexity, respectively. In contrast, our work characterizes the *semantic and logical* requirements of each question from several fixed analytical perspectives at once, and links those requirements to failures with an explicit significance protocol rather than reporting descriptive structure alone.

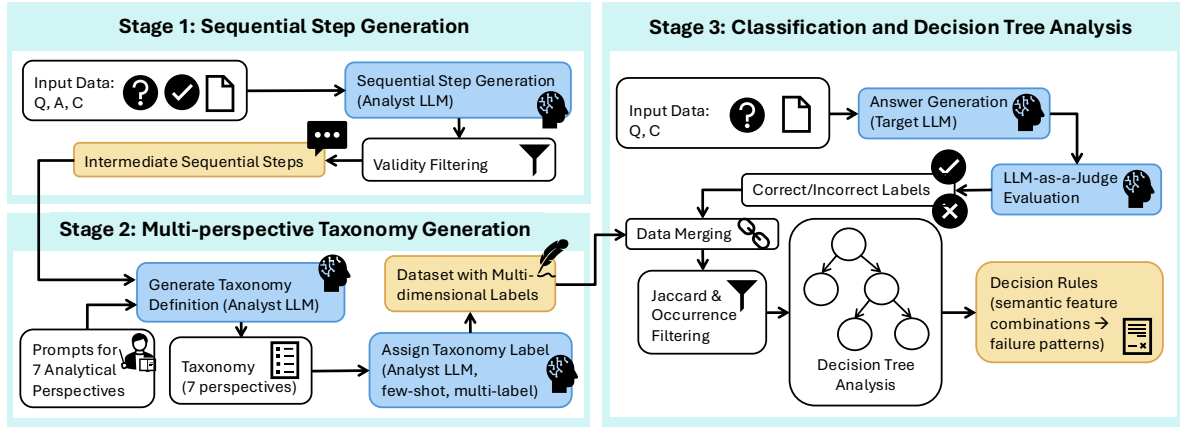


Figure 2: An overview of our automated dataset analysis pipeline.

3 Methodology

Our framework (Figure 2) consists of three stages: (1) Sequential Step Generation, (2) Multi-Perspective Taxonomy Generation and Classification, and (3) Decision Tree Analysis.

3.1 Sequential Step Generation

Standard multi-hop QA datasets provide question-answer pairs but leave the intermediate sequential steps unannotated. We use an Analyst LLM to retroactively generate an explicit sequential step R_i that connects each question Q_i to its answer A_i via the provided context C_i .

Formally stated, given a dataset $\mathcal{D} = \{(Q_i, A_i, C_i)\}_{i=1}^N$ where Q is the question, A is the answer, and C is the context (reference documents), we prompt the Analyst LLM in a zero-shot setting to decompose the derivation into numbered atomic steps (e.g., “Step 1: Identify entity X from document $D...$ ”, “Step 2: Compare dates...”). Concretely, for each question we (i) sample five candidate sequences from the Analyst LLM; (ii) read off the final derived answer of each candidate; (iii) compare it with the gold answer A_i ; and (iv) retain one candidate whose derived answer matches A_i , discarding the question if no candidate matches. This validity filter ensures that retained sequences represent plausible sequential steps rather than arbitrary narratives. Because taking the union of all validated candidates would make the feature vectors all-ones, which reduces the contrast between questions the analysis depends on, we generate one valid derivation per question.

The zero-shot setting avoids biasing the taxonomy toward human annotators’ vocabulary, at the cost of relying on the Analyst LLM’s parametric

priors and lexical preferences (Resnik, 2025); we mitigate this by verifying the taxonomy against human standards (Section 4.4) and by demonstrating cross-model stability (Section 5.3).

3.2 Multi-Perspective Taxonomy Generation and Classification

Standard QA types (e.g., “Comparison”) provide limited diagnostic granularity for fine-grained failures. To capture the diverse nature of reasoning, we generate a taxonomy by prompting the Analyst LLM with seven analytical perspectives, selected from LLM-generated candidates to span three complementary lenses: *cognitive demand* (whether the required computational step can be executed), *logical structure* (whether the inference pattern is valid), and *domain-specific operational requirements* (the procedural constraints that determine answer quality). The seven are Bloom’s Taxonomy, Bottom-Up, Logical Fallacies, Formal Logic, Expert Personas (a Formal Logician, a Cognitive Psychologist, and a Data Scientist), Top-Down, and Data Transitions; Appendix A gives the full definitions and the prompt for each.

The LLM generates definitions for these labels based on the specific dataset content, aligned for fine analysis of each dataset’s unique aspects. Importantly, the seven perspectives are fixed a priori and are identical for every dataset and every Analyst LLM; only the within-perspective vocabulary is generated per dataset. The perspective layer is therefore the unit that is comparable across datasets and models, whereas individual category names are not—a design choice we revisit in Section 5.3 and in the Limitations. A per-perspective ablation (Appendix B) shows that no single perspective alone matches the full feature set, while remov-

ing most individual perspective leaves failure-class F1 largely intact. Appendix L shows that 66% of categories are perspective-unique, the rest forming stable cross-perspective clusters.

After the taxonomy generation, an Analyst LLM classifies each generated sequential step R_i against the taxonomy labels in a few-shot, multi-label setting, producing a feature vector $\mathbf{x}_i \in \{0, 1\}^K$ where K is the number of labels. The classification prompt is shown in Figure 3 of Appendix C.

3.3 Decision Tree Analysis and Statistical Protocol

To isolate which features are associated with LLM failure, we use a decision tree model. To obtain target variables, we determine $y_i \in \{0, 1\}$ by evaluating the Target LLM’s answer to Q_i (correctness judged via LLM-as-a-Judge; see Section 4.5). Before fitting, we prune redundancy based on Jaccard similarity above a threshold j and remove low-frequency labels below f . The tree is then trained to predict y_i from $\mathbf{x}_i$, isolating the specific combinations of taxonomy labels associated with LLM failure.

Crucially, $\mathbf{x}_i$ describes what question Q_i requires and y_i records whether the Target answered it, so a split identifies a subpopulation of questions with an elevated failure rate, not a mechanism inside the Target. Because correctness is imbalanced and the feature space is high-dimensional, plain accuracy is uninformative, and we report a single protocol for all datasets and configurations: (i) the majority-class baseline; (ii) balanced accuracy under stratified 5-fold cross-validation (CV); (iii) a label-permutation test, permuting the correctness labels 500 times at the question level so that the grouping structure is preserved and refitting the model on each permutation, which gives a p -value for the null that no feature combination is associated with failure; and (iv) for each rule we discuss, a Fisher exact test on its 2×2 contingency table, Benjamini–Hochberg corrected across the eleven rules reported here (Benjamini and Hochberg, 1995). Preferring low-dimensional models, we take a depth-limited tree as primary and treat the full-depth tree as exploratory.

One preprocessing step affects how the numbers below must be read. Questions whose steps carry context-quality categories—labels stating that the supplied context is missing or insufficient—are removed before fitting, because they describe a defect of the provided context rather than a reasoning re-

quirement. On FRAMES this removes 232 of the 588 questions and, because such labels co-occur with failure, 127 of the 150 failures, leaving 356 questions whose majority-class baseline is 93.5%. We therefore report FRAMES both with the exclusion (what makes a question difficult when its context is adequate) and without it (what predicts failure over the dataset as sampled), and we do not mix the two. The same exclusion is applied to the other two datasets, where it is far smaller—19 of the 500 MuSiQue questions and a single Multihop-RAG question (one further Multihop-RAG question has no complete feature row)—so those two are reported in the excluded form only.

The choice of model family does not drive the findings: on identical features, an L_1 -regularized logistic regression and a gradient-boosted tree are indistinguishable from the decision tree within CV variability on FRAMES and Multihop-RAG (Appendix J). We keep the tree because its explicit conjunctive rules are the deliverable; classifiers reading raw question text can predict correctness but cannot express rules over taxonomy features, so they answer a different question.

4 Experimental Setup

4.1 Datasets

We used three multi-hop QA datasets.

FRAMES (Krishna et al., 2025) evaluates RAG systems; we used its 588 text-based multi-hop questions, excluding tabular tasks to maintain modality consistency. **Multihop-RAG** (Tang and Yang, 2024) is a news-based dataset requiring temporal and multi-document reasoning; we sampled 588 questions. **MuSiQue** (Trivedi et al., 2022) is designed to prevent shortcut learning through connected 2–4 hop sequential steps and provides gold supporting facts (no retrieval required); we sampled 500 questions.

4.2 Models

The **Analyst LLMs** are Gemini 2.5-Flash (Cormanici et al., 2025) (primary; step generation, taxonomy creation, and classification) plus DeepSeek V3.1 (671B) (DeepSeek-AI et al., 2025), Qwen3-235B (Yang et al., 2025), and GPT-OSS (120B) (OpenAI et al., 2025) as additional Analysts on FRAMES (Section 5.3). The **Target LLMs** are Gemini 2.5-Flash (primary; all three datasets) plus DeepSeek V3.1, GPT-OSS (120B), and Llama 3.3 (70B) (Grattafiori et al., 2024), each evaluated with

the same features, Judge, and tuning budget. The **Judge LLMs** are Gemini 2.5-Flash (primary) plus DeepSeek V3.1, a cross-family Judge that re-labels every answer on all three datasets.

4.3 Evaluation Metric

We employed **LLM-as-a-Judge** to evaluate the correctness of answers, with Gemini 2.5-Flash as the judge. The evaluation prompt is in Figure 4 of Appendix C.

4.4 Validation

Human Annotation. Two annotators (CS students) judged the overall validity of the generated sequential steps on 100 sampled questions, classifying each sequence as “Valid” or “Invalid” according to whether it logically connects the question to the correct answer. Adopting the lower rating on disagreement gives a conservative validity rate of 75%, with 93% agreement and Cohen’s $\kappa = 0.79$. To validate the classification, the same annotators assessed the 385 labels assigned to 100 randomly sampled questions as “Valid,” “Partially Valid,” or “Invalid,” yielding an 85% strict validity rate (326/385) and 98% (377/385) including partially valid labels. Detailed counts and guidelines are in Appendix D.

Answer-blind decomposition. Because the Analyst sees the gold answer when generating the steps (Section 3.1), we repeat the decomposition on 200 questions per dataset with the answer withheld, against a control that differs only in whether the answer is supplied; both conditions were run on the same day and scored by the same Judge. On FRAMES the answer-blind derivation reaches the gold answer in 69.5% of cases (139/200), against 99.0% for the control. The rate is 75.0% on Multihop-RAG and 82.0% on MuSiQue. Two properties of the three samples bear on the comparison. First, the gold answer occurs verbatim in its context for 200/200 MuSiQue questions, 105/200 FRAMES questions, and 69/200 Multihop-RAG questions; MuSiQue answers are spans of the supporting paragraphs. Second, 52.5% of the Multihop-RAG sample is Yes/No, and split by answer format the free-form half reaches 96.8% and the Yes/No half 55.2%.

4.5 Decision Tree Configuration

We tuned the decision tree hyperparameters—split criterion (Gini or entropy), `min_samples_leaf`,

`max_depth`, frequency pruning threshold, and Jacard deduplication threshold—using Optuna. The objective is the F1-score of the failure class ($y = 0$), since the diagnostic target is the minority class of incorrect answers and optimizing plain accuracy on these class distributions selects trees that predict the majority class everywhere. Each trial was evaluated via stratified 5-fold cross-validation grouped by question. Identical prompts, pruning, and tuning budgets are used for every Analyst, Target, and Judge configuration; no per-model prompt engineering or per-model tuning was performed.

5 Results and Analysis

5.1 Taxonomy Overview

On FRAMES, the taxonomy generation process produced 364 raw category labels across the seven perspectives, of which 144 reasoning features were retained after deduplication.¹ Perspective counts and representative categories for all three datasets are given in Appendices E, F, and G. Each question was assigned a median of 33 labels (range 14–64). The most frequent categories—Multi-step Chain Reasoning (574/588 questions), Attribute Extraction (573/588), Information Retrieval (567/588)—are operations common to most multi-hop questions, while others such as False Dilemma appeared only once, so the taxonomy captures both ubiquitous and rare patterns.

5.2 Reasoning Features Associated with Failure

We fit models on each dataset to identify which combinations of reasoning features are associated with Target LLM failure. Table 1 summarizes model performance under the protocol of Section 3.3.

Before detailing the specific failure patterns, we highlight two key observations in Table 1. First, the high cross-validation accuracy of the analyzed FRAMES subset is not evidence of a predictive model. That set retains only 23 of the 150 failures, so its majority baseline is 93.5%, its balanced accuracy is 0.49, and the permutation test cannot distinguish it from the same procedure applied to shuffled labels. Its splits remain informative as

¹We conducted these experiments with Japanese prompts, and category names are generated in both Japanese and English. Since the names are LLM-generated, they may describe the reasoning types of the dataset imprecisely.

Setting	N	Maj.	CV	Bal.	p
FRAMES, all (depth 2)	588	74.5	85.5	0.80	0.002
FRAMES, analyzed	356	93.5	91.0	0.49	0.88
Multihop-RAG	586	75.1	78.8	0.72	0.002
MuSiQue	481	74.0	62.8	0.51	0.33

Table 1: Primary results under the single protocol of Section 3.3. N is the number of questions actually fitted; Maj. is the majority-class (always-correct) baseline on that set; CV and Bal. are plain and balanced accuracy under stratified 5-fold CV; p is the label-permutation test. “FRAMES, all” is our primary FRAMES model and the only row taken before the context-quality exclusion of Section 3.3; the other three rows are post-exclusion sets, and “FRAMES, analyzed” is fitted with the hyperparameters of Table 12. Every other row re-selects hyperparameters on its own set. FRAMES uses 144 features, Multihop-RAG uses 152 features, and MuSiQue uses 181 after pruning.

within-sample enrichments, each tested individually with a Fisher exact test, but the statistical weight of the FRAMES analysis rests on the all-question model. Second, on all 588 questions a low-dimensional model suffices: a depth-2 tree attains balanced accuracy 0.80 with permutation $p = 0.002$, so a two-condition rule already carries most of the recoverable signal, and we treat the full-depth 144-feature tree as an exploratory device for extracting candidate rules. Its large train–CV gap reflects the high-dimensional setting (144 features vs. 588 samples), but the top-level splits are stable across all five random seeds (Appendix B), so we read only those. Tree visualizations, a Random Forest importance ranking, and per-dataset supplementary analyses are in Appendices E, F, G, and H.

The depth-2 model, across all question splits, first on Uncertainty Coping, the category assigned to steps that must cope with missing or insufficient information: questions carrying it but not Hypothesis Construction are answered 43.4% of the time (33/76), or 10.1% (7/69) when they also carry Hypothesis Construction, against 93.5% (344/368) for questions carrying neither it nor Information Gap & Completion. This is precisely the signal the context-quality exclusion removes, which is why the two FRAMES rows differ so sharply; the patterns discussed below come from the analyzed set, where context adequacy is controlled.

FRAMES. The most diagnostically informative interaction involves Simple Numerical

Calculation and Entity Extraction: among questions requiring numerical calculation, those in which entity extraction is absent are answered correctly 56% of the time, against 86% when both are present. We call this contrast the Implicit Math bottleneck: the prerequisite values must be inferred implicitly rather than extracted explicitly. Tested as a condition against the rest of the dataset rather than as a tree leaf, it is significant on the analyzed set (60% against 95%; Fisher exact, BH-corrected $p = 0.001$) and, more importantly, on all 588 questions, where the exclusion no longer applies (48% against 76%; corrected $p = 0.003$); restricting the comparison to questions that require calculation, so that only explicit extraction varies, gives 60% against 93% and 48% against 77% respectively. The all-question test covers the questions the Target actually failed as well as those removed by the context-quality exclusion, so the gap depends on neither. An LLM-based attribution of a primary cause to 118 of the 150 FRAMES failures does not separate questions inside the condition from those outside it (Fisher exact $p = 0.35$), so we do not rely on it. A second pattern, Intent-Heavy Filtering, appears when filtering requires resolving implicit question constraints (e.g., “movies that were considered successful”): [NOT Simple Numerical Calculation] AND Filtering/Refinement AND Intent Understanding is answered at 66%, suggesting that ambiguity resolution rather than retrieval difficulty is the operative constraint. It does not survive on all 588 questions (25.9% against 25.5% failure, corrected $p = 1.0$), so unlike Implicit Math we report it as specific to the analyzed subset. Both statements concern which questions are answered less often, not why the Target failed (Appendix G).

Multihop-RAG. Accuracy across the dataset’s built-in question types ranges from 64.5% (null_query) to 82.9% (temporal_query), a spread of 18.4pp (Table 7).² Question types correspond strongly to answer formats; once the format is controlled, the built-in labels become near-uninformative (Section 5.5). The pattern the analysis surfaces instead is Conditioned Filtering: failures concentrate where aggregated information must be verified against multiple contrasting conditions (e.g., “Is A greater than B AND did C

²Computed from the same correctness labels the models are fitted on; $N = 587$ because one question has no complete feature row.

happen before $D?$ ”), so judgment rather than retrieval is the operative constraint—a distinction not recoverable from the question-type labels alone (see Appendix E for the tree visualization).

MuSiQue. With gold supporting facts removing retrieval as a confound, the pipeline yields 181 features (pruned from 419 raw categories) and the Target answers 74.2% of the 500 sampled questions correctly, 481 of which survive the exclusion of Section 3.3. Because the MuSiQue model does not beat chance out of sample (balanced accuracy 0.51, permutation $p = 0.33$), we present its splits not as predictive rules but as within-sample enrichments, each Fisher-tested and BH-corrected across the reported rules (all $p \leq 0.014$). Deductive Rule Application is the primary root split; its absence shifts difficulty to temporal ordering and attribute matching. Temporal questions grounded entirely in document evidence succeed reliably (0/25 errors) but fail at 64% (7/11) when combined with Linguistic and Semantic Interpretation—the Temporal Order Logic Paradox, where semantic disambiguation rather than temporal ordering is the operative constraint. The highest error rate (7/10) involves Attribute Value Identification under implicit Constrained Selection, mirroring the implicit-constraint structure of Implicit Math in FRAMES (see Appendix F for the corresponding tree splits). These subpopulations are small and describe this dataset only.

5.3 Robustness Across the Analyst, Target, and Judge Roles

The pipeline uses an LLM in three places—Analyst, Target, and Judge—and in the primary configuration all three are Gemini 2.5-Flash. A finding could therefore reflect self-consistency of one model family rather than a property of the data. We vary each role in turn, keeping everything else fixed (Table 2).

Target. Re-running the full pipeline with DeepSeek V3.1, GPT-OSS (120B), and Llama 3.3 (70B) as Targets, using the shared Judge and identical features, reproduces the FRAMES association ($p = 0.002$ each). The three answer 82.0%, 80.8%, and 46.8% of FRAMES questions correctly, against the primary Target’s 74.5%. The DeepSeek V3.1 and GPT-OSS trees select Uncertainty Coping at the root, as the primary Target’s all-question tree does, and both place Simple Numerical Calculation—the feature

Role	Model	N	Bal.	p	Root
—	Gemini 2.5-Flash	588	0.80	0.002	UC
Target	DeepSeek V3.1	579	0.65	0.002	UC
Target	GPT-OSS 120B	579	0.68	0.002	UC
Target	Llama 3.3 70B	579	0.65	0.002	NumOp
Judge	DeepSeek V3.1	585	0.64	0.002	IDP
Analyst	DeepSeek V3.1	587	0.53	0.030	other
Analyst	Qwen3-235B	586	0.52	0.18	other
Analyst	GPT-OSS 120B	576	0.50	0.41	other

Table 2: Varying one role at a time on FRAMES, all questions, with everything else held fixed; the first row is the primary configuration. Bal. is cross-validated balanced accuracy and p the permutation test of Section 3.3. “Root” names the root feature: UC is Uncertainty Coping and IDP is Information Deficiency Presentation, two categories of the same context-insufficiency family; NumOp is Numerical Operation, which the feature filter keeps separate from Simple Numerical Calculation; “other” is a feature outside the context-insufficiency family. N differs across rows only because joining each configuration’s outputs to the feature table loses a few questions (one to twelve). The three additional Targets answer 82.0%, 80.8%, and 46.8% of FRAMES questions correctly against the primary Target’s 74.5%; on MuSiQue they give permutation $p = 0.20, 0.23$, and 0.25, the same null result as the primary Target.

defining Implicit Math—among their top five splits. The Llama 3.3 tree selects Numerical Operation at the root, a category the feature filter keeps separate from Simple Numerical Calculation, and does not place the latter in its top five. On MuSiQue all three give a null result ($p = 0.20, 0.23$, and 0.25), matching the primary Target. Llama 3.3 left the answer field empty on 17.3% of FRAMES and 32.8% of MuSiQue items, against under 2% for the other Targets; those items are scored incorrect. One asymmetry is worth stating: for the DeepSeek V3.1 Target the permutation test is significant on balanced accuracy but not on plain accuracy ($p = 0.25$), the expected pattern when a tree improves minority-class recall without improving majority-class predictions.

Judge. Judging every answer with DeepSeek V3.1 agrees with the primary Judge on 78.8%, 82.5%, 79.8% of items on FRAMES, Multihop-RAG, MuSiQue ($\kappa = 0.515, 0.573, 0.564$). For calibration, an independent re-run of the *same* Gemini Judge reproduces its own labels on only 90.6% (533/588) of FRAMES items, so much of the cross-family disagreement is judge stochasticity rather than a family effect. The residual disagreements

are systematic: DeepSeek is the stricter judge in 77%/71%/91% of them. Refitting the FRAMES tree on the DeepSeek labels preserves the association (balanced accuracy 0.64, $p = 0.002$) and again puts a context-insufficiency category at the root, though a different one: Information Deficiency Presentation rather than Uncertainty Coping. No reported conclusion depends on which judge is right, but which one is right on the contested items is not settled here.

Analyst. We extend the pipeline to three additional Analysts—DeepSeek V3.1, Qwen3-235B, and GPT-OSS (120B)—on FRAMES. To keep the comparison meaningful we do not regenerate the taxonomy: all Analysts classify against the common labels generated by Gemini 2.5-Flash in Section 5.1, and the rest of the pipeline is applied unchanged. Feature counts, tuned hyperparameters, accuracies, and per-model tree visualizations are in Appendix I.

On the full FRAMES dataset, under the protocol of Section 3.3, the primary Analyst reaches balanced accuracy 0.80 ($p = 0.002$), whereas the three additional Analysts stay at 0.53 ($p = 0.030$), 0.52 ($p = 0.18$), and 0.50 ($p = 0.41$); only the DeepSeek Analyst is nominally significant, and a balanced accuracy of 0.53 is not a usable model. Their plain CV accuracies show the same ordering (85.5% against 67–69%, the latter below the majority-class baseline). This is not an artifact of differential tuning: every Analyst receives identical prompts, pruning, and tuning budget (Section 4.5), and re-selecting hyperparameters on the common question set does not change the picture. The per-Analyst trees are therefore descriptive summaries of a particular labeling, and the predictive claims of this paper are made only for the primary Analyst.

What does transfer is the content of the diagnosis rather than the fit. At the level of feature groups, the temporal–numerical bundle accounts for 17.8–20.1% of total feature importance for all four Analysts. Pairwise agreement on the selected feature sets is high among the three additional Analysts (Jaccard 0.39–0.57; Spearman ρ of per-feature importance 0.66, 0.49, and 0.27) but low between the primary Analyst and each of them (Jaccard 0.03–0.09; ρ 0.21, 0.08, and -0.26). The gap is one of vocabulary rather than substance: the primary Analyst populates the shared label space with a different subset of names, so surface-form overlap understates the agreement, which is why we read

convergence at the bundle and perspective level. Three conclusions recur across all four trees under different label names—temporal–numerical alignment, contextual disambiguation versus parametric bias, and early sequence termination—with the per-Analyst labels in Appendix I. The perspective layer, not the category vocabulary, is what transfers, as the design in Section 3 intends.

5.4 Sequence-Aware Failure Analysis via N-grams

The models above treat labels as an unordered set, neglecting the ordering of reasoning operations (e.g., filtering *before* versus *after* integration). As a complementary, exploratory view we compare N-grams of the step-category sequences between the correct and incorrect groups. For the primary Analyst, Uncertainty Coping $\rightarrow$ Information Search & Extraction $\rightarrow$ Logical Completion is strongly overrepresented among failures (+36.4pp): on encountering missing context, the derivation forces a logical completion rather than withholding a conclusion. The remaining patterns, the per-Analyst tables, and exact tests on the rare high-error sequences are in Appendix K; all rest on few questions and serve as qualitative illustrations of *how* a derivation can go wrong, so no quantitative claim in this paper depends on them.

5.5 Does the Dataset’s Own Description Already Say This?

A natural objection is that these findings restate what each benchmark was built to test. We operationalize it by asking whether the labels encoding a dataset’s design predict Target correctness. Because y_i is binary we use **McFadden’s pseudo- R^2** (logistic regression MLE), comparing the built-in labels against our seven perspective groupings—not the 144 individual features—to avoid rewarding dimensionality.

Multihop-RAG, controlling for answer format.

Question types correspond strongly to answer formats: inference questions are almost exclusively open-ended (197 of 199), while comparison and temporal are predominantly Yes/No, so the apparent discriminative power of the built-in types is largely attributable to format rather than question content. Restricting to Yes/No questions ($N = 297$) removes the confound: the built-in labels become near-uninformative (pseudo- $R^2 < 0.001$) while all six taxonomy axes with sufficient

Grouping	McF. Pseudo- R^2
Question Type (baseline)	<0.001
Formal Logic	0.159
Bottom-Up	0.132
Data Transition	0.112
Top-Down	0.054
Logical Fallacy	0.024
Bloom’s Taxonomy	0.014
QType + Formal Logic	0.261

Table 3: McFadden’s Pseudo- R^2 for Yes/No questions only ($N = 297$), comparing question type classification with proposed taxonomy axes. All taxonomy axes exceed the near-zero baseline.

coverage exceed that baseline (Table 3), and the combined model reaches 0.261. A complexity-penalized check agrees: adjusted R^2 is higher for the taxonomy groupings (0.015) than for question types (0.007).

FRAMES and MuSiQue. The same comparison on FRAMES (linear R^2 , multi-label reasoning types) and MuSiQue (pseudo- R^2 , hop-count types) gives the same picture. On FRAMES the built-in reasoning types explain 2.3% of the variance in correctness ($R^2 = 0.023$), whereas our taxonomy axes explain up to 40.2% (Bloom’s Taxonomy, $R^2 = 0.402$), a $17\times$ difference. On MuSiQue ($N = 500$) the built-in hop-based types explain near-zero variance (pseudo- $R^2 = 0.008$) against up to 0.105 (Top-Down) for our axes, and 0.132 for the combined model.

Advantages over static dataset descriptions. Predefined static labels are either confounded by surface features such as answer format or too coarse to capture the relevant distinctions. The gain is specifically *conditional*: FRAMES states that it probes numerical reasoning, but not that failures concentrate on the numerical questions whose operands are not explicitly extractable (48% vs. 77%, Section 5.2), and it is that second distinction that names the questions on which a model is weak. A residual analysis, the per-perspective ablation, and a cross-perspective validity analysis of the taxonomy are in Appendices E, B, and L.

6 Discussion: From Dataset Diagnosis to Model Diagnosis

Characterizing a dataset and diagnosing a model are two readings of the same output. Each fitted model is the failure profile of one Target on one dataset. Applying the identical pipeline to several Targets separates two sources of difficulty: prop-

erties whose association with failure recurs across model families point to limitations shared by current LLMs—all four Targets’ FRAMES trees give a significant association ($p = 0.002$ each) and all four produce a null result on MuSiQue, and the three frontier-scale ones select Uncertainty Coping at the root with Implicit Math components among their top splits—whereas differences between profiles localize a model-specific weakness. Under the identical pipeline the Llama 3.3 tree roots in Numerical Operation rather than Uncertainty Coping. What the design does not license is the causal step from a recurring association to an inherent capability limit, for the reasons given in Section 3.1.

The output is actionable in two ways. As a *selector*, a discovered rule defines an evaluation subset on which current models are roughly 30 points weaker than on the rest of the same benchmark, obtained without new annotation. As a *specification*, it says which capability to fix: the audited FRAMES failure in Appendix G decomposes a date interval correctly and then mis-sums the days—a case for delegating arithmetic to a tool, not for more retrieval.

7 Conclusion

This study introduced an automated, multi-perspective framework for identifying reasoning types associated with errors in multi-hop QA. An Analyst LLM generates a multi-label taxonomy over seven fixed perspectives, and interpretable models over the resulting features isolate combinations that structural analysis and the datasets’ own labels both miss: Implicit Math in FRAMES, Conditioned Filtering in Multihop-RAG, and a Temporal Order Logic Paradox in MuSiQue. Future work should analyze the Target’s own reasoning traces alongside the reconstructed requirement profiles, incorporate step ordering, extend the analysis to additional datasets and languages, and use the labels to augment multi-hop QA datasets. The pipeline only needs (input, gold output, context) triples, so extending it to agentic and code-reasoning benchmarks, such as SWE-bench (Jimenez et al., 2024), is another possible application.

Limitations

We identify the following limitations.

- **Reconstructed requirements versus actual**

failures. The steps we analyze are not the Target’s own failed reasoning steps. Consequently, our results are associations between properties of questions and Target failures, and a split does not establish that a property caused a failure. Analyzing the Target’s own traces would close it.

- **Out-of-sample generalization is dataset- and configuration-dependent.** The FRAMES and Multihop-RAG models are significant under the permutation test, but the MuSiQue model does not beat chance out of sample, so its patterns are reported as within-sample enrichments only (Section 5.2). The same holds for the FRAMES model once the context-quality exclusion is applied. Out-of-sample significance also holds only for the primary Analyst (Section 5.3). More generally, the full-depth trees achieve high descriptive fit with modest generalization and should be read as descriptive models of specific dataset failure patterns.
- **Dataset-specific vocabularies.** The seven perspectives are fixed, but the category vocabulary is regenerated per dataset (144 features for FRAMES, 152 for Multihop-RAG, and 181 for MuSiQue), so category names are not directly comparable across datasets, and a practitioner applying the framework to a new dataset must rerun taxonomy generation. Fixing a single global vocabulary would restore comparability at the cost of the dataset-specific granularity that produces the findings reported here; constraining classification to a shared, enumerated label space is a direction for future work.
- **Scope of the claims.** The object of analysis is the dataset. A per-Target profile localizes where a model is weak on a given benchmark, and properties that recur across Targets suggest shared limitations (Section 6), but neither constitutes a certificate about a model’s inherent capabilities. Only four Targets and two Judges, all open or Gemini-family models, were examined; closed frontier models were not.
- **Single validated derivation per question.** We assume a single sequential derivation per question, whereas multi-hop questions may

admit several valid ones (Section 3.1). Aggregating over candidates is straightforward but raises the already high label count per question; enumerating and assessing multiple equivalent derivations remains future work.

- **Solver response quality.** Llama 3.3 leaves the answer field empty on 17.3% of FRAMES and 32.8% of MuSiQue items (Section 5.3). Our protocol scores these incorrect, so that Target’s correctness labels mix reasoning failure with failure to follow the output format, and its tree should be read accordingly.
- **The headline condition is defined by the Analyst’s own labels.** Membership in the Implicit Math condition is decided by the labels the Analyst assigned, not by an independent human reading of each question. We did not run such a re-judgement, so the reported contrast compares question groups as the Analyst draws them; the human validation we do report (Section 4.4) covers the steps and the labels, not the delimitation of the condition.
- **Judge disagreement.** The two Judges disagree on roughly a fifth of items (Section 5.3), and no reported conclusion depends on which is correct, but we did not adjudicate a sample of those items by hand, so the disagreement is bounded rather than resolved.
- **Label recall.** Our human validation (Section 4.4) assessed the precision of assigned labels but not recall, because defining the exhaustive set of applicable labels for a step is impractical.
- **Analyst label preference.** Different Analyst LLMs describe the same failure region with labels drawn from different perspectives (Section 5.3), so their names do not overlap by surface form.
- **Faithfulness of individual steps.** The validity filter accepts sequences that reach the correct answer; a sequence with a wrong intermediate step that nonetheless reaches the gold answer will be retained and contributes label noise.

Ethical Considerations

This research adheres to the ACL Code of Ethics.

- **Data Privacy and Licensing:** All datasets are publicly available under open licenses: FRAMES (Apache-2.0), Multihop-RAG (ODC-BY), and MuSiQue (CC BY 4.0). None contains personally identifiable information (PII) of private individuals. All datasets were used solely for research purposes, consistent with their intended use.
- **Human Annotation:** The annotation study involved evaluating LLM-generated label–reason pairs and did not collect personal data or subject annotators to any risk. Annotators participated voluntarily as academic peers without compensation. Formal ethics review was not required under our institutional guidelines for this type of quality evaluation task.
- **Environmental Impact:** The computational cost of this analysis is relatively low compared to pre-training. We used API-based inference.
- **Risk of Misinterpretation:** The “Logical Fallacies” perspective in our taxonomy may flag benign text as fallacious due to the probabilistic nature of LLMs. We emphasize that these labels are diagnostic heuristics, not absolute judgments of truth.
- **Dual Use:** We do not foresee any immediate dual-use risks. The methodology is designed to improve the robustness and interpretability of QA systems.
- **Use of AI Assistants:** AI assistants (LLM-based tools) were used throughout this research for coding, data analysis, and manuscript writing. All research decisions, experimental designs, and conclusions are the authors’ own.

Acknowledgements

This paper is partly supported by TOYOTA TECHNOLOGICAL INSTITUTE Research Center for Smart Information Technology and partly based on results obtained from the BRIDGE Program (R7-H05), implemented by the Cabinet Office, Government of Japan.

References

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the*

Royal Statistical Society: Series B (Methodological), 57(1):289–300.

Benjamin Samuel Bloom. 1956. *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook 1: Cognitive Domain*. Longman, London.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. [Deepseek-v3 technical report](#). *Preprint*, arXiv:2412.19437.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Gangwei Jiang, Yahui Liu, Zhaoyi Li, Wei Bi, Fuzheng Zhang, Linqi Song, Ying Wei, and Defu Lian. 2025. [What makes a good reasoning chain? uncovering structural patterns in long chain-of-thought reasoning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6490–6514, Suzhou, China. Association for Computational Linguistics.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. [SWE-bench: Can language models resolve real-world GitHub issues?](#) In *The Twelfth International Conference on Learning Representations*.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. [Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation](#). In *Proceedings of the 2025*

- Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759, Albuquerque, New Mexico. Association for Computational Linguistics.
- Qichuan Liu, Chentao Zhang, Chenfeng Zheng, Guosheng Hu, Xiaodong Li, and Zhihong Zhang. 2025. [Beyond the answer: Advancing multi-hop QA with fine-grained graph reasoning and evaluation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23433–23456, Vienna, Austria. Association for Computational Linguistics.
- Sewon Min, Eric Wallace, Ari Holtzman, Maarten Sap, Hannaneh Hajishirzi, and Yejin Choi. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4249–4255.
- OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastian Bubeck, and 108 others. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). Preprint, arXiv:2508.10925.
- Philip Resnik. 2025. [Large language models are biased because they are large language models](#). *Computational Linguistics*, 51(3):885–906.
- Yixuan Tang and Yi Yang. 2024. [Multihop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries](#). In *First Conference on Language Modeling*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [MuSiQue: Multi-hop questions via single-hop question composition](#). *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. [Self-preference bias in LLM-as-a-judge](#). In *Neurips Safe Generative AI Workshop 2024*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Jian Wu, Linyi Yang, Zhen Wang, Manabu Okumura, and Yue Zhang. 2025. [CofCA: A STEP-WISE counterfactual multi-hop QA benchmark](#). In *The Thirteenth International Conference on Learning Representations*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-bench and chatbot arena](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Appendix: Taxonomy Perspectives Details

We provide detailed definitions for the seven perspectives used in the taxonomy generation prompt. In the prompts, LLMs are given the perspectives and their sub-category, which are used to generate more specified category labels for the datasets (e.g., Remembering and Understanding for Bloom’s Taxonomy).

1. Bloom’s Taxonomy: Adapted for NLP tasks (Bloom, 1956).

- *Remembering*: Direct retrieval of facts, listing, or identifying specific entities from long-term memory or context.
- *Understanding*: Constructing meaning from messages; interpreting, summarizing, paraphrasing, or explaining concepts.
- *Applying*: Executing or implementing a procedure in a given situation; using rules or calculations.
- *Analyzing*: Breaking material into constituent parts and detecting relationships; differentiating, organizing, and attributing.
- *Evaluating*: Making judgments based on criteria and standards; checking, critiquing, or validating consistency.
- *Creating*: Putting elements together to form a novel, coherent whole; generating, planning, or producing a new structure.

2. **Bottom-Up:** Inductively constructs a classification framework in two steps:

- *Step 1 (Atomic Attributes):* Identifies independent “facets” such as “Number of Sources,” “Numerical Processing Necessity,” “Temporal Handling,” “Causal Inference,” “Information Conflict,” and “Direction of Inference.”
- *Step 2 (Archetype Synthesis):* Identifies frequent co-occurrence patterns of these attributes to define “Problem Archetypes” (e.g., “Time-Series Analysis with Conflict Resolution”).

3. **Logical Fallacy:** Analyzes sequential steps for patterns susceptible to informal fallacies:

- *Hasty Generalization:* Generalizing from insufficient evidence.
- *Questionable Cause (Post Hoc):* Concluding causality from simple correlation or temporal sequence.
- *Red Herring:* Introducing irrelevant information to divert the argument.
- *False Dilemma:* Presenting fewer options than actually exist.
- *Quoting Out of Context:* Distorting meaning by omitting context.
- *Fallacy of Composition/Division:* Erroneously transferring properties between whole and parts.
- *Straw Man:* Attacking a distorted version of an argument.

4. **Formal Logic:** Classifies the dominant inferential type:

- *Deductive Reasoning:* Deriving necessary conclusions from general rules/premises (e.g., Syllogism, Modus Ponens).
- *Inductive Reasoning:* Inferring general rules from specific observations (probabilistic).
- *Abductive Reasoning:* Inferring the best explanation for an observation (hypothesis generation).
- *Analogical Reasoning:* Transferring properties from a source to a target based on structural similarity.

5. **Expert Persona:** Simulate three analytical viewpoints:

- *Formal Logician:* Focuses on the validity of structure, identifying logical leaps, modus ponens/tollens usage, without regard for content truth.
- *Cognitive Psychologist:* Focuses on reasoning demand and analytical processes. Identifies operations like information retrieval, integration, comparison, conflict resolution, and inferential constraints.
- *Data Scientist:* Focuses on the “Data Pipeline.” Identifies steps as data operations: filtering, joining contexts, aggregation, transformation, and input/output mapping.

6. **Top-Down:** Constructs a hierarchical taxonomy starting from the most abstract level:

- *Level 1:* Defines mutually exclusive, comprehensive top-level categories covering fundamental reasoning differences.
- *Level 2:* Decomposes each Level 1 category into specific sub-categories.
- *Level 3 (Optional):* Further decomposition for complex patterns.

7. **Data Transition:** Focuses on the “edges” or transitions between sequential steps rather than the steps themselves:

- *Information Transformation:* Aggregation (many-to-one), Filtering (selection), Abstraction (concrete-to-rule), Instantiation (rule-to-concrete).
- *Logical Relationship:* Causality, Inclusion (set-element), Conflict/Resolution, Premise/Conclusion.
- *Task Dependency:* Sequential vs. Parallel Integration (independent sub-paths merging).

B Appendix: Split Stability and Perspective Ablation

B.1 Top-Level Split Stability Across Random Seeds

Table 4 shows the top-4 decision tree split features (BFS order) across five random seeds on the FRAMES analyzed subset (Gemini 2.5-Flash Analyst, optimal hyperparameters from Table 12). All seeds yield identical splits, confirming that the split features are not artifacts of random initialization. This checks stability with respect to the random

seed only; it does not bear on out-of-sample validity, which Table 1 shows is absent on this subset.

B.2 Per-Perspective Ablation Study

To quantify each analytical perspective’s contribution to failure prediction, we ablate the feature set by perspective—either restricting it to the categories of a single perspective, or removing one perspective at a time (leave-one-out)—and retrain the decision tree under each condition. Table 5 reports, for each ablation condition on the FRAMES dataset, the F1-score of the failure class ($y = 0$) averaged over 5-fold stratified cross-validation, using DeepSeek V3.1 as the Analyst ($N = 588$, optimal hyperparameters from Table 12). We use DeepSeek V3.1 because its per-viewpoint JSONL classification files are available, allowing exact category-to-perspective assignment; the main Gemini analysis merged all perspectives before storage. The single-perspective results confirm that no individual perspective alone matches the full-model baseline, while the leave-one-out results show that removing the *Logical Fallacy* perspective causes the largest drop ($0.229 \rightarrow 0.155$), identifying it as the most diagnostically irreplaceable axis.

C Appendix: Analyst and Judge Prompts

Path Analysis Classification Prompt. To ensure reproducibility, we provide the translated prompt structure (Figure 3) used by the Analyst LLM to classify the generated sequential steps into the taxonomy. This corresponds to the path analysis phase of our framework.

LLM-as-a-Judge Evaluation Prompt. To evaluate the correctness of the Target LLM’s responses against the ground-truth answers, we utilize Gemini 2.5-Flash as an automated judge. Below is the English translation of the prompt template (Figure 4) used for this evaluation (originally executed in Japanese).

D Appendix: Human Annotation Details

D.1 Details for Classification Annotation

Annotators were presented with a web-based interface displaying, for each question, all LLM-generated (category, reason) pairs produced by the classification stage. For each pair, annotators judged whether the stated reason correctly justified the category assignment using the following three-level scale:

You are an analytical assistant with advanced reasoning capabilities.

Instruction Based on the following ‘Classification Categories and Criteria’, analyze which category each given ‘Sequential Step’ belongs to, and output the result in JSON format.

Constraints

- **Multi-label Classification:** A single sequential step may fall into multiple categories. List all applicable categories.
- **Granularity:** The classification judgment should focus on the individual steps and elements that make up each sequential step.
- **Strictness:** Classifications must strictly follow the provided ‘Classification Categories and Criteria’.
- **Concreteness:** In the reason field, concretely describe which part of the sequential step meets the criteria for that category.

Classification Categories and Criteria: [Category Definitions]

Sequential Step (List format): [Generated Sequential Step]

Output Format (JSON): Please generate a JSON array strictly with the following structure:

```
[
  {
    "reasoning_path_id": "Question n",
    "classifications": [
      {
        "category": "[Category Name]",
        "reason": "[Concrete reasoning for the classification]"
      }
    ]
  }
]
```

Figure 3: A prompt used to classify generated sequential steps.

Seed	Node 0 (Root)	Node 1	Node 2	Node 3
42	Knowledge Application & Commonsense Reasoning	Aggregation & Comparison from Structured Data	Direct Deduction	Numerical Computation Error
123	Knowledge Application & Commonsense Reasoning	Aggregation & Comparison from Structured Data	Direct Deduction	Numerical Computation Error
456	Knowledge Application & Commonsense Reasoning	Aggregation & Comparison from Structured Data	Direct Deduction	Numerical Computation Error
789	Knowledge Application & Commonsense Reasoning	Aggregation & Comparison from Structured Data	Direct Deduction	Numerical Computation Error
1234	Knowledge Application & Commonsense Reasoning	Aggregation & Comparison from Structured Data	Direct Deduction	Numerical Computation Error

Table 4: Top-4 decision tree split features (BFS order) across 5 random seeds, FRAMES analyzed subset, Gemini 2.5-Flash Analyst ($N = 356$). All seeds produce identical splits.

=== Task === Please help evaluate the response provided by the LLM by comparing it to the ground-truth answer. Determine whether the ground-truth answer is contained within the LLM’s response. Analyze the provided data to make your judgment. === Instructions === - Carefully compare the “predicted response” with the “ground-truth answer”. - Examine the content of the response and look for equivalent information or correct answers. Do not obsess over exact phrasing unless the precise expression determines the meaning. - Make your final judgment based on whether the meaning and crucial facts of the “ground-truth answer” are included in the “predicted response”. === Input Data === - Question: {question} - Predicted Response: {response} - Ground-Truth Answer: {answer} === Output Format === Please provide your final evaluation in the following format: <reason> (Explain how you made your decision) </reason> <judge> (“TRUE” or “FALSE”) </judge> Please proceed with the evaluation.
--

Figure 4: A prompt used to evaluate Target LLM’s answer.

Setting	N Features	CV F1 (mean $\pm$ std)
<i>Baseline</i>		
All Features	107	0.229 $\pm$ 0.065
<i>Single perspective only</i>		
Only Bloom’s Taxonomy	11	0.014 $\pm$ 0.028
Only Bottom-Up	23	0.176 $\pm$ 0.031
Only Logical Fallacy	11	0.143 $\pm$ 0.076
Only Formal Logic	10	0.041 $\pm$ 0.053
Only Expert Persona	23	0.077 $\pm$ 0.074
Only Top-Down	12	0.041 $\pm$ 0.053
Only Data Transition	17	0.126 $\pm$ 0.051
<i>Leave-one-out</i>		
Without Bloom’s Taxonomy	96	0.266 $\pm$ 0.097
Without Bottom-Up	84	0.238 $\pm$ 0.083
Without Logical Fallacy	96	0.155 $\pm$ 0.043
Without Formal Logic	97	0.250 $\pm$ 0.070
Without Expert Persona	84	0.313 $\pm$ 0.075
Without Top-Down	95	0.235 $\pm$ 0.068
Without Data Transition	90	0.247 $\pm$ 0.093

Table 5: Per-perspective ablation on FRAMES (DeepSeek V3.1 Analyst). CV F1 is an F1-score of the failure class ($y = 0$) over 5 stratified folds. No single perspective matches the full baseline, confirming complementary coverage. Without Expert Persona exceeding the baseline indicates that Expert Persona features add noise rather than signal for DeepSeek’s taxonomy on this dataset.

- **Valid:** The reason correctly and fully justifies the category assignment based on the content of the sequential step.
- **Partially Valid:** The reason partially supports the category but with minor gaps, ambiguity, or over-generalization.
- **Invalid:** The reason does not support the category assignment, or the category is inappropriate for the given sequential step.

Annotators had access to the full definitions of each classification category during evaluation.

Evaluation	Count	Percentage
Valid	326	85%
Partially Valid	51	13%
Invalid	8	2%

Table 6: Human evaluation of the automated reasoning labels on a subset of 100 questions.

Question Type	N	Accuracy
temporal_query	123	82.9%
comparison_query	194	74.7%
inference_query	208	74.0%
null_query	62	64.5%
Overall	587	75.1%

Table 7: Per question-type accuracy on Multihop-RAG (Gemini 2.5-Flash, $N = 587$), computed from the same correctness labels the models are fitted on. An 18.4pp gap separates the easiest and the hardest type; Section 5.5 controls for answer format. One question is missing because it has no complete feature row.

Each annotator evaluated items independently. In cases of inter-annotator disagreement, the lower rating was adopted as the final label to provide a conservative estimate of label quality. The interface displayed the reasoning_path_id, the category name, its definition, and the LLM-generated reason for each classification entry. Table 6 shows the overall results of classification annotation.

E Appendix: Supplementary Analysis on Multihop-RAG

Per Question-Type Accuracy. Table 7 reports the primary Target’s accuracy on each of Multihop-RAG’s built-in question types, referenced from Section 5.2.

Temporal Reasoning and RAG. Unlike FRAMES, Multihop-RAG contains dense news articles. The decision tree highlighted Time Series Comparison as a primary splitting feature for failure. Models exhibit performance degradation when normalizing relative temporal expressions (e.g., last Tuesday) across documents. A visualization of the corresponding decision tree splits is provided in Figure 5.

Residual Analysis. To confirm that the taxonomy captures genuine semantic difficulty beyond answer format in Multihop-RAG, we computed each category’s expected accuracy from its answer format mix and compared it to the actual accuracy. The two base rates are the Target’s mean accuracy on

Yes/No questions, 77.0% ($n = 296$), and on open-ended questions, 73.2% ($n = 291$); a category with 60% Yes/No questions therefore has an expected accuracy of $0.6 \times 77.0\% + 0.4 \times 73.2\% = 75.5\%$. Because the two base rates differ by only 3.8 pp, answer format can account for at most that much of any category’s deviation. Among categories with $n \geq 20$ questions the largest residuals are Hypothesis Formation (−60.3 pp, $n = 20$), Temporal Order & Consistency Evaluation (−46.4 pp, $n = 87$), Time-Series Change Analysis (−41.3 pp, $n = 41$), and Consistency Evaluation (−38.0 pp, $n = 86$). The second of these is the root split of the Multihop-RAG tree (Figure 5), so the residual analysis and the tree isolate the same category independently.

Taxonomy Label Statistics. Table 8 shows the per-perspective raw category counts for the Multihop-RAG taxonomy generated by Gemini 2.5-Flash. The taxonomy comprises 193 unique raw categories (196 per-perspective total, with 9 cross-perspective overlaps), pruned to 152 features for decision tree training. Each question was assigned a median of 23 labels (range: 7–37). The most frequently assigned categories were Result Generation (98.1% of questions), Understanding (96.4%), and Element Extraction (91.0%), while 25 categories appeared in only a single question.

F Appendix: Supplementary Analysis on MuSiQue

Taxonomy Label Statistics. Table 9 shows the per-perspective raw category counts for the MuSiQue taxonomy. The taxonomy comprises 419 unique raw categories (452 per-perspective total, with 16 cross-perspective overlaps), pruned to 181 features. Each question was assigned a median of 29 labels (range: 10–63). The most frequently assigned categories were Multi-step Chain Reasoning (100% of questions), Attribute Extraction (100%), and Remembering (98.8%), while 10 categories appeared in only a single question. The Formal Logic perspective generates the largest number of raw categories (70).

Structural Comparison with FRAMES and Multihop-RAG. Comparing the three taxonomies reveals that MuSiQue generates markedly more features after pruning (181 vs. 144) despite a smaller question pool (500 vs. 588). The

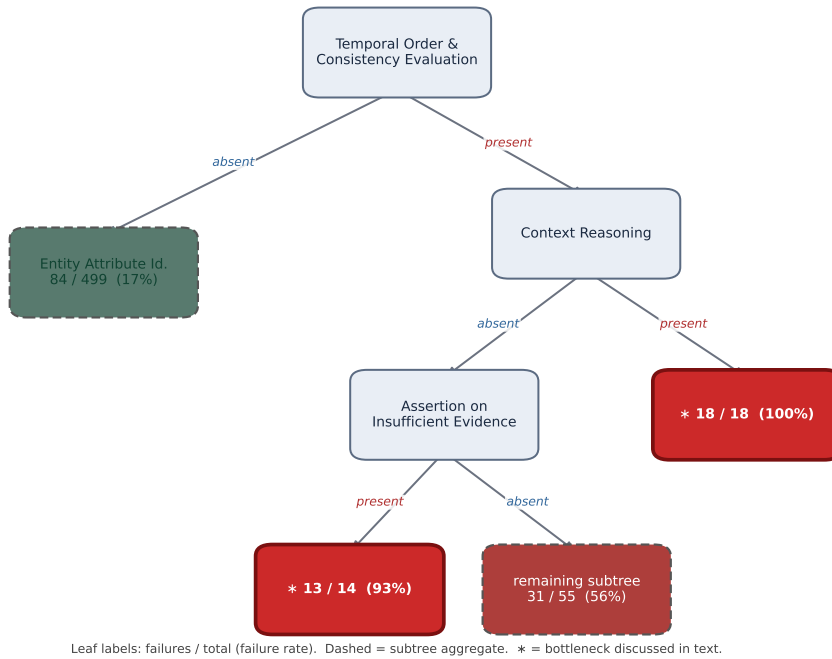


Figure 5: Key-split view of the Multihop-RAG decision tree (Gemini 2.5-Flash Analyst; 152 features, $N = 586$). The root split is Temporal Order & Consistency Evaluation: its absence covers the majority ($n=499$) at 17% error, whereas its presence ($n=87$) raises the error to 71% and cascades—via Context Reasoning and premature Assertion on Insufficient Evidence—into near-total-failure leaves (18/18 and 13/14). This isolates temporal comparison combined with a judgment shortcut as the operative constraint (not retrieval), matching the Time Series Comparison / Conditioned Filtering pattern discussed in Section 5.2. Solid boxes are leaves (failures / total; failure rate); dashed boxes are subtree aggregates; * marks the bottlenecks discussed in the text.

Perspective	#Cat.	Representative Categories
Bloom’s Taxonomy	2	Understanding, Remembering
Bottom-Up	27	Multi-Source Attribute Integration
Logical Fallacy	20	Source Misattribution
Formal Logic	50	Deductive Rule Application, Condition Matching
Expert Persona	46	Query Understanding, Information Extraction
Top-Down	20	Compound Multi-Fact Verification
Data Transition	31	Result Generation, Element Extraction
Total raw (unique)	193	
Total (after pruning)	152	

Table 8: Per-perspective raw category counts for the Multihop-RAG taxonomy (Gemini 2.5-Flash Analyst), computed over the 193 distinct raw categories in the decision-tree input ($N = 587$). Categories are assigned to perspectives via the per-perspective classification; the per-perspective counts sum to 196 because 9 categories are recovered by more than one perspective, and 6 of the 193 are not perspective-tagged. The pruned feature count of 152 corresponds to the main analysis (Table 1).

Perspective	#Cat.	Representative Categories
Bloom’s Taxonomy	65	Remembering, Understanding
Bottom-Up	53	Multi-step Chain Reasoning, Final Answer Construction
Logical Fallacy	64	Direct Information Extraction
Formal Logic	70	Attribute Extraction, Multi-step Deduction
Expert Persona	68	Context Join, Multi-step Cognitive Chain
Top-Down	52	Entity Identification, Inferential Combination
Data Transition	47	Relation Identification, Matching
Total raw (unique)	419	
Total (after pruning)	181	

Table 9: Per-perspective raw category counts for the MuSiQue ($N = 500$) taxonomy (Gemini 2.5-Flash Analyst). Overall raw category is pruned to 181 features for the decision tree (Table 1).

concentration of common categories (100% coverage for the top two) alongside a long tail of 10 singletons mirrors the pattern observed in FRAMES, confirming that the taxonomy robustly captures both universal multi-hop operations and dataset-specific edge cases regardless of collection method. Additionally, Figure 6 visualizes the key decision tree splits for MuSiQue discussed in Section 5.2.

G Appendix: Supplementary Analysis on FRAMES

Taxonomy Statistics per Perspective. Table 10 lists, for the FRAMES taxonomy referenced in Section 5.1, the number of categories retained per perspective after pruning together with representative examples. The corresponding statistics for Multihop-RAG and MuSiQue are in Appendices E and F.

Key Decision Tree Splits. Table 11 reports the Gini impurity reduction for the top nodes of the FRAMES tree, and Figure 7 shows the corresponding splits. Simple Numerical Calculation is

Perspective	#Cat.	Representative Cat.
Bloom’s Taxonomy	37	Analyzing, Evaluating
Bottom-Up	34	Time-Series Comparison
Expert Persona	58	Intent Understanding
Formal Logic	15	Multi-step Deduction
Logical Fallacy	7	Quoting Out of Context
Data Transition	18	Filtering/Refinement
Top-Down	20	Entity Extraction
Total	144	

Table 10: Categories (Cat.) per perspective after pruning, with FRAMES examples. Perspectives may share categories, so the counts do not sum to the deduplicated Total.

the first split on the path we trace, separating calculation-heavy from retrieval-heavy questions. The calculation path is not uniformly difficult: it splits further on Entity Extraction, and the failure rate drops to 13% (7/52) when explicit extraction is present.

Attribution of Actual Target Failures. Section 5.2 reports an LLM-based attribution of a primary cause to 118 of the 150 questions the primary Target answered incorrectly. Each output

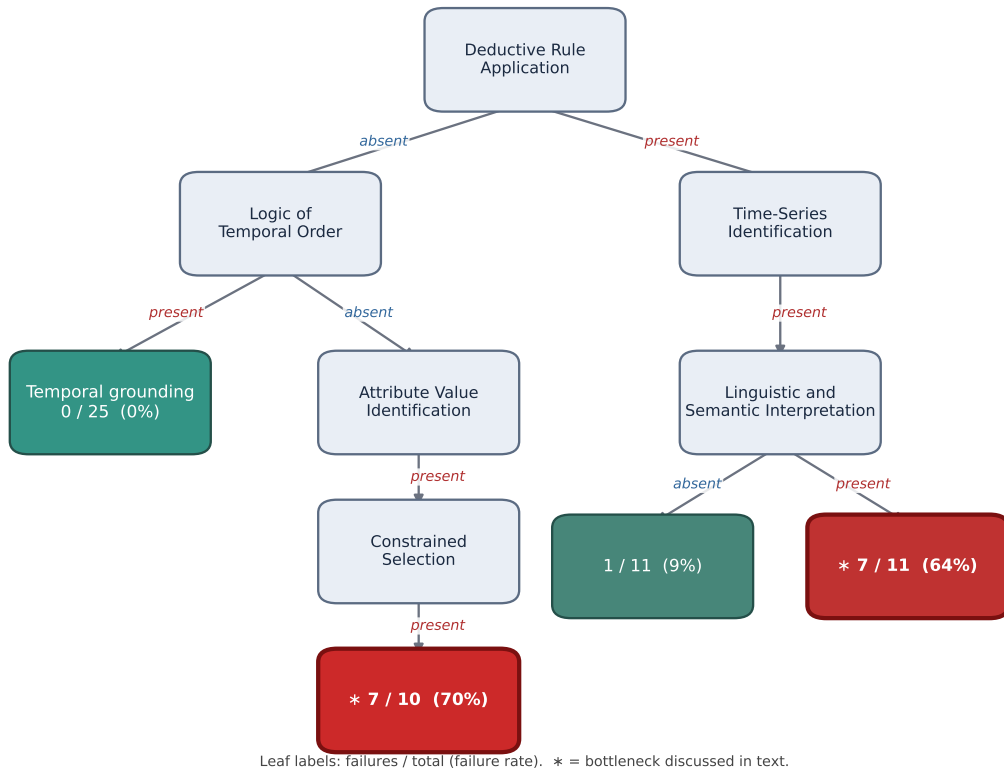


Figure 6: Key-split view of the MuSiQue decision tree (Gemini 2.5-Flash Analyst). Derived from the full tree by tracing the root-to-leaf paths analyzed in Section 5.2 and collapsing splits that retain $> 90\%$ of parent samples with $\Delta P_{\text{fail}} < 0.05$; internal nodes thus show split features only, and leaves report (failures / total; failure rate). Edges indicate feature presence/absence; * marks the two bottlenecks discussed in the text.

Feature	Gini	N
Simple Numerical Calc.	0.146	365
Filtering/Refinement	0.080	287
Intent Understanding	0.185	68
Entity Extraction	0.327	68
Multi-step Deduction	0.233	52

Table 11: Gini impurity reduction and sample counts (N) for key decision tree nodes.

was assigned one of four causes: a grounding gap (the answer is not supported by the retrieved context), capture by a distractor, a gold-answer artifact, or a reasoning error not covered by the first three. Grounding gaps dominate both inside and outside the Implicit Math condition (7/13 against 73/105), and no cause separates the two groups (Fisher exact $p = 0.35$ for grounding gaps, the smallest of the four contrasts being a gold-answer artifact at $p = 0.06$). The attribution therefore neither supports nor contradicts the Implicit Math association, and no claim in this paper rests on it.

Case Study: Calculation Failure. We present an instance where the model retrieves the necessary information but outputs an incorrect calculation. The question asks for the number of days between the US release of *The Lord of the Rings: The Return of the King* (December 17, 2003) and the death of its cinematographer, Andrew Lesnie (April 27, 2015). The model correctly identifies both dates and decomposes the calculation into two intervals:

1. **Dec 17, 2003 to Dec 17, 2014:** Calculated correctly as 4,015 days (11×365) plus 3 leap days (2004, 2008, 2012), totaling 4,018 days.
2. **Dec 17, 2014 to April 27, 2015:** The model sums the days of each month ($14 + 31 + 28 + 31 + 27$) but computes the sum as 143 instead of the correct 131.

As a result, the final answer (4,161) deviates from the ground truth (4,149). This arithmetic failure demonstrates that even with correct decomposition, LLMs can exhibit high error rates on basic arithmetic operations, necessitating the integration of external tools for robust numerical reasoning.

H Appendix: Feature Importance Analysis

We utilized a Random Forest (100 trees) to stabilize feature importance estimates. The top 10 features are listed below.

1. Structure-based Aggregation (0.0307): Aggregating data from structured lists.
2. Intent Understanding (0.0246): Inferring implicit constraints.
3. Simple Numerical Calculation (0.0239): Basic arithmetic (+, -, *, /).
4. Attribute Transformation (0.0209): Changing units or formats (e.g., date formats).
5. Logical Imputation (0.0193): Filling in missing information logically.
6. Inference to the Best Explanation (0.0190): Abductive reasoning.
7. Time Series Comparison (0.0171): Ordering events.
8. Attribute Extraction (0.0162): Identifying specific properties.
9. Comparative Analysis (0.0162): Comparing two entities.
10. Information Interpretation (0.0160): Converting text to meaning.

I Appendix: Detailed Results in Cross-Model Analysis

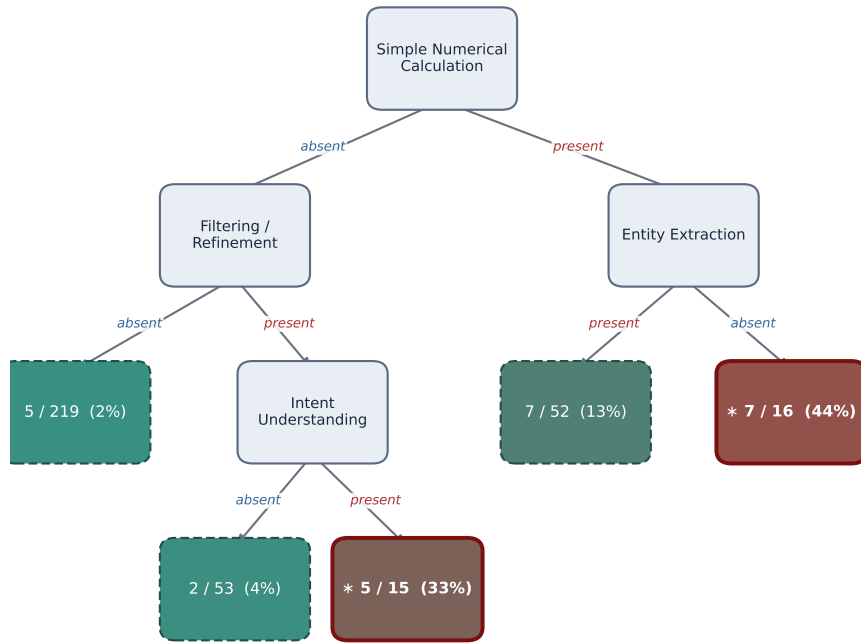
I.1 Tuning and Accuracies

Table 12 reports, for each of the four Analyst LLMs on FRAMES, the number of raw generated categories, the number of features retained after tuning-selected pruning, the selected hyperparameters, and the training and cross-validation accuracies discussed in Section 5.3.

I.2 Common Conclusion Across Analysts

Section 5.3 states that three conclusions recur across the four Analysts under different label names. The per-Analyst labels are as follows.

Temporal-numerical alignment. A primary failure correlate in all four trees: Detailed Temporal Calculation (Gemini 2.5-Flash), Temporal Handling (DeepSeek V3.1), Deductive Temporal Ordering (Qwen3-235B and, together with Implicit External Knowledge, GPT-OSS). The bottleneck is the alignment of temporal intervals across documents rather than the arithmetic itself.



Leaf labels: failures / total (failure rate; accuracy = 1 - failure). Solid boxes are leaves; dashed boxes are subtree aggregates. * = bottleneck discussed in text.

Figure 7: Key-split view of the FRAMES decision tree (Gemini 2.5-Flash Analyst). We trace the two failure paths analyzed in Section 5.2 and collapse a near-passthrough intermediate split, so that only decision-relevant splits are shown. Internal nodes are annotated with split features only, and leaves report (failures/total; failure rate), where accuracy = 1 - failure rate. Edges indicate whether a reasoning feature is present or absent. * marks the two bottlenecks discussed in the text: Simple Numerical Calculation without Entity Extraction (the Implicit Math bottleneck) and Filtering/Refinement combined with Intent Understanding (Intent-Heavy Filtering).

Analyst LLM	N	Tuned Feats.	Optimal Hyperparameters	CV Acc.	Train Acc.
Gemini 2.5-Flash	356	144	$\{f = 3, j = 0.93, \text{crit} = \text{Gini}, l = 7, d = 9\}$	$90.9 \pm 5.2\%$	94.57%
DeepSeek V3.1	499	107	$\{f = 5, j = 0.71, \text{crit} = \text{Gini}, l = 2, d = 10\}$	$67.2 \pm 6.7\%$	82.49%
Qwen3-235B	510	211	$\{f = 4, j = 0.88, \text{crit} = \text{Entropy}, l = 5, d = 10\}$	$68.9 \pm 3.6\%$	77.64%
GPT-OSS (120B)	566	80	$\{f = 3, j = 0.76, \text{crit} = \text{Entropy}, l = 1, d = \text{None}\}$	$67.5 \pm 2.7\%$	95.05%

Table 12: Hyperparameter tuning and decision tree results across four Analyst LLMs on FRAMES. CV Acc. reports the mean cross-validation accuracy. Optimal Hyperparameters denote the frequency threshold (f), Jaccard similarity threshold (j), splitting criterion, minimum leaf samples (l), and maximum depth (d). The N column shows that the context-quality exclusion of Section 3.3 removes many more questions under the primary Analyst’s vocabulary than under the others’, because the excluded category names are surface strings from that vocabulary. The CV accuracies in this table are therefore computed on different question sets and are not comparable across rows; the comparable numbers, computed on all questions under one protocol, are reported in Section 5.3 and Table 2.

Contextual disambiguation versus parametric bias. Where the context contains ambiguous or implicit entities, failures concentrate on questions whose steps invoke external knowledge: DeepSeek’s Implicit External Knowledge, Gemini’s Knowledge Application, and GPT-OSS’s Implicit External Knowledge Introduction. This echoes counterfactual benchmarks such as CofCA (Wu et al., 2025).

Early sequence termination. Questions demanding sequential verification—Qwen3’s Withholding Conclusion, Gemini’s Condition Suitability, GPT-OSS’s Leap of Logic from Insufficient Info—are associated with premature answers that bypass the remaining steps.

I.3 DeepSeek V3.1: Cognitive Bias and Fallacy-Driven Diagnosis.

The decision tree trained on the DeepSeek V3.1 taxonomy focuses on cognitive constraints and logical fallacies. The top-level splits prioritize features such as Implicit External Knowledge Introduction (feature importance: 0.057) and Ignoring Definition Ambiguity (0.044). The analysis reveals a distinct failure path with 100% training error (6/6 samples) characterized by:

Path D1: Query Reinterpretation → NOT Handling Temporal Information

This path represents cases where the query is reinterpreted without grounding in the temporal context provided in the documents. The DeepSeek V3.1 analysis also reveals a failure mode involving temporal extreme value tracking (88% error rate, 7/8 samples):

Path D2: NOT Time-Series Inference → Numerical Processing → Implicit External Knowledge → NOT Numerical Comparison

In these cases, arithmetic calculations are performed, but explicit numerical comparison is bypassed in favor of pre-existing parametric knowledge (implicit memory) rather than the context, leading to factual errors. Figure 8 situates both paths within the decision tree, which is rooted at Query Reinterpretation: D1 lies on its *present* branch and D2 on its *absent* branch.

I.4 Qwen3-235B: Procedural Short-Circuiting and Step Omission.

The Qwen3-235B taxonomy reflects procedural step execution. The decision tree reveals failures arising from the short-circuiting of sequential steps, such as skipping intermediate verification. The primary failure path with 100% training error (5/5 samples) is defined as:

Path Q1: Goal-Directed Reasoning → NOT Hypothesis Construction → NOT Withholding Conclusion → NOT Memory Retrieval (Answer-First)

This path represents an unsupported inference, where the final answer template is targeted without constructing a hypothetical intermediate state, withholding judgment, or verifying facts against the context. A bottleneck in deductive multi-stage reasoning is also identified (80% error rate, 4/5 samples):

Path Q2: Multi-stage Processing → Deductive Comparative Judgment

This indicates that even when multi-stage planning is executed correctly, the final deductive comparison across structured features remains error-prone. Figure 9 traces both paths from the tree root, Answer-First Reasoning: Q1 on its *present* branch and Q2 on its *absent* branch.

DeepSeek V3.1 (FRAMES) -- failure paths from the decision-tree root

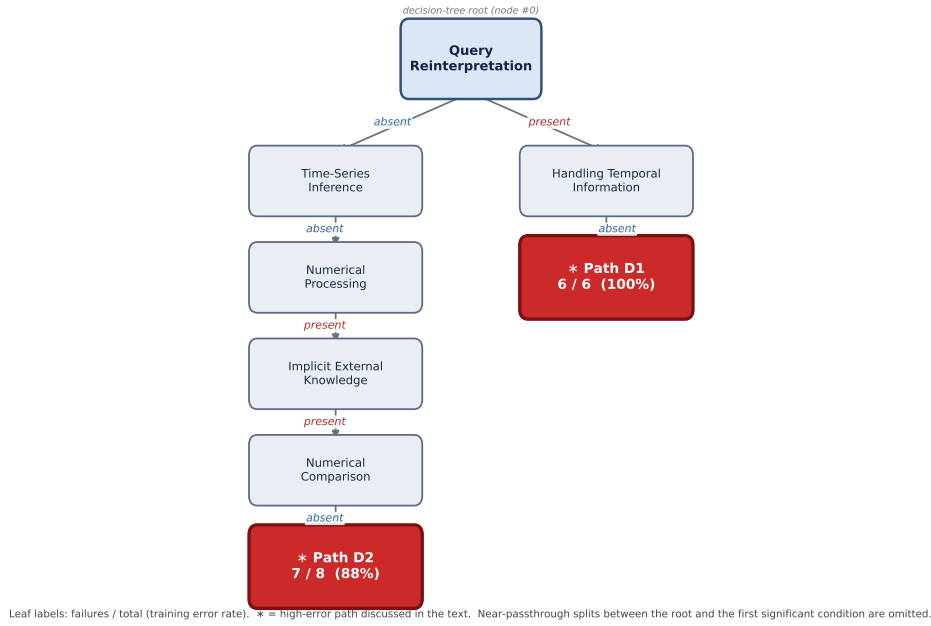


Figure 8: Decision-tree view of the two DeepSeek V3.1 failure paths on FRAMES (f1-tuned tree; root node #0 = Query Reinterpretation, $N = 497$, base error 26%). Path D1 lies on the *present* branch (Query Reinterpretation → NOT Handling Temporal Information; 6/6 errors); Path D2 on the *absent* branch (NOT Time-Series Inference → Numerical Processing → Implicit External Knowledge → NOT Numerical Comparison; 7/8 errors). Internal nodes show split features only; edges mark feature *presence/absence*; leaves report (failures / total; training error). Near-passthrough splits between the root and the first significant condition are collapsed, following Fig. 7. * marks the high-error paths discussed in the text.

I.5 Gemini 2.5-Flash: Knowledge Dependency Partitioning.

As a baseline, the Gemini 2.5-Flash decision tree places Knowledge Application and Commonsense Reasoning (0.223) and Intent Understanding (0.171) at the root of its classification rules. This partitions the dataset primarily by whether a question requires anchoring to external commonsense or implicit intent. When numerical calculations are combined with implicit intent, accuracy drops to 66%. This aligns with the Implicit Math bottleneck observed in the main analysis.

I.6 GPT-OSS (120B): Implicit External Knowledge Integration and Temporal Alignment Bias.

The decision tree trained on the GPT-OSS (120B) taxonomy focuses on deductive verification and contextual grounding constraints. The top-level splits prioritize features such as Analysis (feature importance: 0.078), Deductive Rule Application (0.052), Deductive Information Integration (0.041), and Implicit External

Knowledge Introduction (0.040), indicating that logical rigor and systematic analytical alignment are primary discriminators of correctness.

The analysis reveals a distinct failure path with 100% training error (3/3 samples) characterized by:

Path O1: Information Extraction → Direct Deduction → NOT Analysis → Implicit External Knowledge Introduction

This cascade highlights that when information is extracted and direct deduction is attempted without a rigorous analytical filter, the model falls back to implicit parametric memory, causing factual hallucinations.

The GPT-OSS analysis also reveals a failure mode where numerical calculation is executed without retrieval grounding and information integration constraints (100% training error, 3/3 samples):

Path O2: NOT Information Extraction → Deductive Numerical Calculation → NOT Memory → NOT Deductive Information Integration

In these cases, when information grounding is completely absent, deductive calculation proceeds on unsupported or empty premises, yielding severe cal-

Qwen3-235B (FRAMES) -- failure paths from the decision-tree root

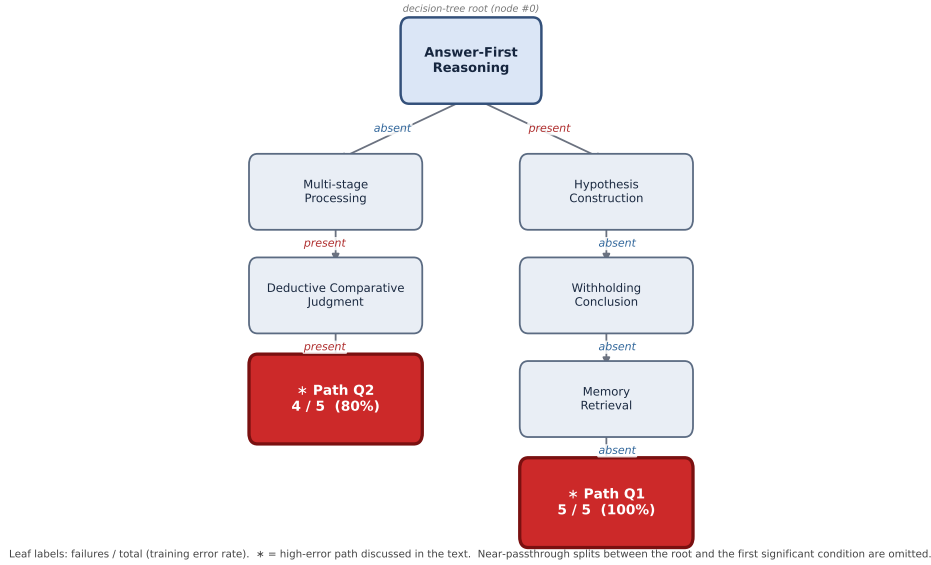


Figure 9: Decision-tree view of the two Qwen3-235B failure paths on FRAMES (f1-tuned tree; root node #0 = Answer-First Reasoning, $N = 492$, base error 26%). Path Q1 lies on the *present* branch (Answer-First Reasoning → NOT Hypothesis Construction → NOT Withholding Conclusion → NOT Memory Retrieval; 5/5 errors); Path Q2 on the *absent* branch (Multi-stage Processing → Deductive Comparative Judgment; 4/5 errors). Conventions follow Fig. 8.

culuation and logical errors. Figure 10 shows both paths branching from the tree root, Information Extraction: O1 on its *present* branch and O2 on its *absent* branch.

J Appendix: Alternative Classifiers on Identical Features

The decision tree is only one of several models that can be fit to the same binary feature matrix, so we check that the reported associations are not an artifact of the tree family. Using identical features, identical stratified 5-fold splits, and an identical tuning budget, we compare the tuned decision tree against an L_1 -regularized logistic regression (a sparse linear model over the same features) and a gradient-boosted decision tree (GBDT, a strong non-interpretable reference on tabular binary features). Table 13 reports the result.

On each set the models fall within cross-validation variability of one another (fold-level standard deviations are 0.03–0.06), so the choice of classifier does not drive the findings. The comparison also reproduces the contrast of Section 5.2 in a model-independent way: on the analyzed FRAMES subset every model sits at chance, whereas on the all-question set every model is well above it. We retain the decision tree as the pri-

Model	FRAMES	FR. ana.	M-RAG
Decision tree	0.804	0.486	0.720
L_1 logistic reg.	0.757	0.511	0.710
GBDT	0.756	0.515	0.692
Random forest	0.775	0.500	0.706

Table 13: Cross-validated balanced accuracy under identical features, splits, and pruning. FRAMES is the all-question set ($N = 588$), FR. (ana.) the 356-question analyzed subset, and M-RAG Multihop-RAG ($N = 586$).

mary model because its explicit conjunctive rules—not its predictive accuracy—are the deliverable of the analysis: an L_1 linear model gives per-feature coefficients but not the feature *combinations* that Section 5.2 reports, and GBDT gives neither.

We deliberately do not include classifiers that read the raw question text, such as a fine-tuned encoder or a prompted LLM. Such models can predict Target correctness, possibly better than any model over our features, but they cannot express a rule over taxonomy features and therefore cannot produce the interpretable diagnosis that is the goal here; a comparison against them would measure predictive power rather than diagnostic content.

GPT-OSS (120B) (FRAMES) -- failure paths from the decision-tree root

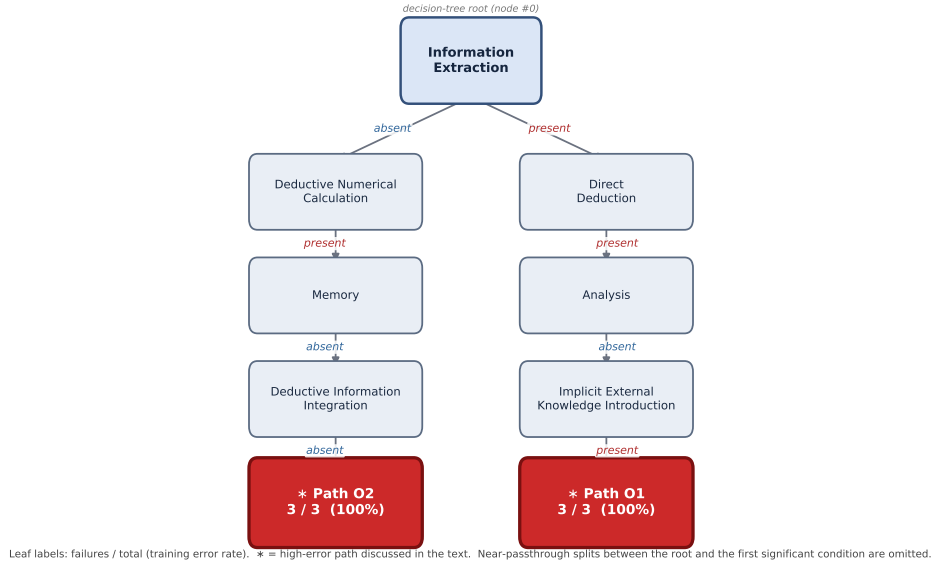


Figure 10: Decision-tree view of the two GPT-OSS (120B) failure paths on FRAMES (f1-tuned tree; root node #0 = Information Extraction; every decision path begins with this split). Path O1 lies on the *present* branch (Information Extraction → Direct Deduction → NOT Analysis → Implicit External Knowledge Introduction; 3/3 errors); Path O2 on the *absent* branch (NOT Information Extraction → Deductive Numerical Calculation → NOT Memory → NOT Deductive Information Integration; 3/3 errors). Conventions follow Fig. 8.

Analyst LLM	N-gram Sequence	Err. (%)	Corr. (%)
Gemini 2.5-Flash	Uncertainty Coping → Info Search & Ext → Logical Comp	38.67	2.28
	Info Defect → Uncertainty Coping → Info Search & Ext	32.00	1.37
	Uncertainty Eval → Hyp Form/Back Reasoning → Hyp Construction	11.33	0.00
DeepSeek V3.1	Info Ext → Ded Rule → Fact Recog	11.41	5.71
	Temporal Calc → Ext Know → Ded Num Calc	4.03	0.23
Qwen3-235B	Leap of Faith → Info Ext → Info Integ	2.74	0.48
	Num Misinterpret → Num Proc → Info Ext	2.05	0.24
GPT-OSS	Dir Info Ext → Multi-step Ded	10.07	5.15
	Ded Num Calc → Dur Calc & T-S → Num Calc	8.05	4.92

Table 14: N-gram frequencies in Error (Err.) vs. Correct (Corr.) step sequences across all Analyst LLMs on the FRAMES dataset. A higher Err. than Corr. rate means the sequence is overrepresented in failed sequences.

K Appendix: Sequence-Aware Failure Analysis via N-grams

This appendix expands the ordered view summarized in Section 5.4. We extract the sequential steps for each question and perform an N-gram analysis of the resulting category sequences across the successful and failed groups, which reveals patterns invisible to unordered feature representations. Table 14 lists the most discriminative transitions for all four Analyst LLMs on FRAMES. Table 15 lists the full category name for each abbreviation, grouped by Analyst LLM. Because each Analyst assigns a different subset of the shared label vocabulary, we report an abbreviation under the model whose sequences include it.

Beyond the primary Analyst’s pattern dis-

cussed in Section 5.4, the most discriminative sequences for the remaining Analysts are Temporal Calculation → Necessity of External Knowledge → Deductive Numerical Calculation for DeepSeek V3.1 (+3.8pp), Numerical Misinterpretation → Numerical Processing → Information Extraction for Qwen3-235B (+1.8pp), and the 2-gram Direct Information Extraction → Multi-step Deduction for GPT-OSS (+4.9pp), which bypasses intermediate bridging and matches the early-termination pattern of Appendix I.2.

Interpreting high-differential N-grams directly as causal variables introduces methodological errors. A qualitative review of the matched sequences shown in Table 14 categorizes the transitions into two primary types.

Common Structural Sequences. Specific transitions, such as Gemini 2.5-Flash’s Uncertainty Coping → Info Search & Ext → Logical Comp (matched 68 times), DeepSeek’s Info Ext → Ded Rule → Fact Recog (matched 42 times), Qwen3’s Leap of Faith → Info Ext → Info Integ (matched 6 times), and GPT-OSS’s Dir Info Ext → Multi-step Ded (matched 37 times), are observed in both correct and incorrect outputs. These

sequences represent standard computational sequences for multi-step processing. Their increased frequency within the incorrect output subset is consistent with error propagation: longer derivations with more intermediate transitions offer more opportunities for an early extraction error to propagate to the final output.

High-Error Sequence Transitions. Other sequences occur almost exclusively among incorrect outputs. Recounted over the full question set, Gemini 2.5-Flash’s uncertainty-evaluation sequence (Uncertainty Eval $\rightarrow$ Hyp Form/Back Reasoning $\rightarrow$ Hyp Construction) matches 17 questions, all of them failures (Fisher exact $p < 0.001$); DeepSeek V3.1’s temporal-external sequence matches 7 questions, 6 of them failures ($p = 0.003$); and Qwen3-235B’s numerical-misinterpretation sequence matches 4 questions, 3 of them failures ($p = 0.067$). We treat the first two as supported and the third as a suggestive trend only. Because all three subpopulations are small ($n = 4\text{--}17$), we present them as qualitative illustrations of failure mechanisms rather than as statistical results: a qualitative audit of the matched questions shows that when Qwen3 misinterprets numerical data (such as a postcode or a count of player entries) and applies arithmetic to it, the derivation is corrupted regardless of subsequent steps, and that when DeepSeek attempts a temporal comparison without the required external knowledge, it generates an unsupported calculation.

L Appendix: Cross-Perspective Validity Analysis

The per-perspective ablation (Table 5) shows that removing most individual perspectives leaves failure-class F1 largely intact—and removing Expert Persona even improves it—raising a natural concern: do the perspectives guide the LLM’s classification, or does the model produce similar categories regardless of the prescribed analytical perspectives (i.e., prompt insensitivity)? To address this, we quantified the *convergent validity* (shared structure across perspectives) and *discriminant validity* (perspective-specific categories) of the generated taxonomy.

Hierarchical Clustering of Categories. We computed pairwise Pearson correlations between all 189 category labels (filtered to ≥ 5 occurrences) based on their co-occurrence across questions. We

then performed agglomerative hierarchical clustering (average linkage) on the resulting distance matrix ($d = 1 - r$). The optimal distance threshold was determined via Silhouette analysis across the range $[0.2, 1.0]$, yielding 50 clusters.

Of these 50 clusters, 20 contain categories from **multiple perspectives**, confirming that certain reasoning constructs are independently recovered by different analytical lenses. The three largest cross-perspective clusters and their constituent categories are:

1. **Mathematical Reasoning** (16 categories, 5 perspectives): Simple Math (Bloom’s), Deductive Math (Formal Logic), Quantitative Analysis (Top-Down), Date Calculation (Bloom’s), Time-Series Computation (Bottom-Up).
2. **Uncertainty and Information Gaps** (14 categories, 6 perspectives): Hypothesis Construction (Bloom’s), Abductive Completion (Formal Logic), Absent Information Handling (Bottom-Up), Uncertainty Coping (Top-Down).
3. **Comparison and Ranking** (6 categories, 5 perspectives): Compare & Contrast (Bloom’s), Comparison and Ranking (Top-Down), Inductive Relative Comparison (Formal Logic).

Notably, the clustering of Hypothesis Construction (from Bloom’s Taxonomy) with Abductive Completion (from Formal Logic) into a single cluster ($r = 0.67$) provides empirical support for the theoretical equivalence of educational-psychological and formal-logical conceptions of hypothesis-driven reasoning.

Unique Category Rate. We measured the proportion of categories for each perspective that have *no strong cross-perspective correlate* ($|r| < 0.3$ with all categories from other perspectives). A high uniqueness rate indicates that the perspective contributes distinct analytical dimensions.

As shown in Table 16, **66.1%** of all categories are perspective-unique, demonstrating that each perspective contributes distinct analytical dimensions. The remaining 33.9% form stable cross-perspective clusters representing fundamental reasoning constructs (e.g., mathematical reasoning, uncertainty handling) that are independently discovered regardless of the prescribed analytical lens.

Abbreviation	Full Category Name
<i>Gemini 2.5-Flash</i>	
Info Search & Ext	Information Search & Extraction
Logical Comp	Logical Completion
Info Defect	Information Deficiency Presentation
Uncertainty Eval	Uncertainty Evaluation
Hyp Form/Back Reasoning	Hypothesis Formation / Backward Reasoning
Hyp Construction	Hypothesis Construction
<i>DeepSeek V3.1</i>	
Info Ext	Information Extraction
Ded Rule	Deductive Rule Application
Fact Recog	Fact Recognition & Identification
Temporal Calc	Temporal Calculation
Ext Know	Necessity of External Knowledge
Ded Num Calc	Deductive Numerical Calculation
<i>Qwen3-235B</i>	
Leap of Faith	Leap to Conclusion from Insufficient Information
Info Ext	Information Extraction
Info Integ	Information Integration
Num Misinterpret	Numerical Misinterpretation
Num Proc	Numerical Processing
<i>GPT-OSS (120B)</i>	
Dir Info Ext	Direct Information Extraction
Multi-step Ded	Multi-step Deduction
Ded Num Calc	Deductive Numerical Calculation
Dur Calc & T-S	Duration Calculation & Time-Series Analysis
Num Calc	Numerical Calculation

Table 15: Full category names for the abbreviated labels used in the N-gram sequences of Table 14 and Section 5.4, grouped by Analyst LLM.

Perspective	Total	Unique	Rate
Logical Fallacy	7	7	100.0%
Expert Persona	58	47	81.0%
Data Transition	18	13	72.2%
Bloom’s Taxonomy	37	23	62.2%
Bottom-Up	34	18	52.9%
Top-Down	20	10	50.0%
Formal Logic	15	7	46.7%
Overall	189	125	66.1%

Table 16: Unique category rate per perspective. A category is “unique” if its maximum absolute Pearson correlation with any category from another perspective is below 0.3.

This dual structure—convergent validity on core constructs and discriminant validity through perspective-specific categories—suggests that the perspectives are neither redundant (which would imply prompt insensitivity) nor entirely orthogonal (which would undermine the notion of stable reasoning constructs). Rather, they provide complementary views that share a common foundation while offering unique analytical depth.