A. Previous System Architecture

• The initial system focused on evaluating Chain-of-Thought (CoT) and Tree-of-Thought (ToT) prompting strategies for mathematical reasoning tasks.

• Mathematical datasets such as GSM8K and MGSM were used primarily for performance benchmarking.

• The selection of reasoning strategies was performed using static rules or manual configuration.

• The Tree-of-Thought module utilized full multi-branch reasoning, which resulted in high computational and token cost.

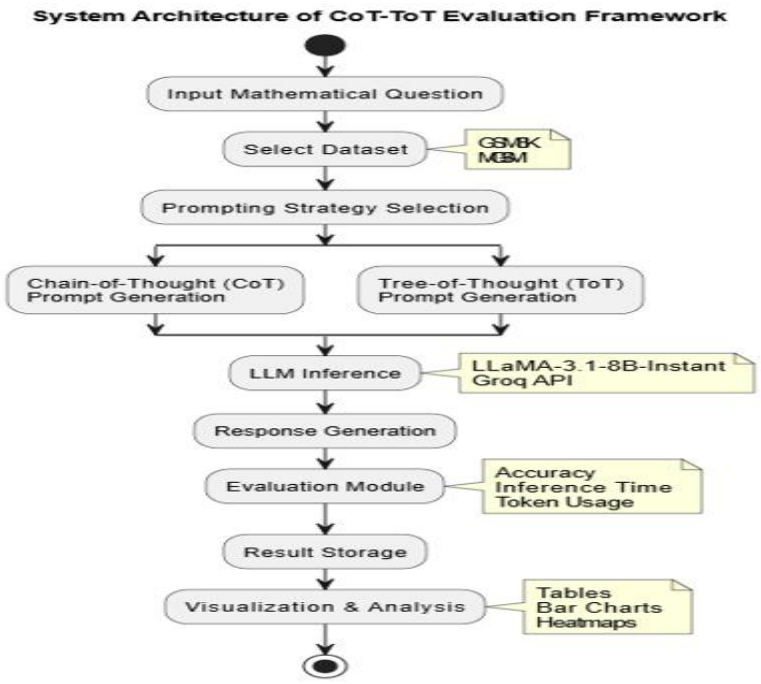
• The inference pipeline directly generated outputs from the language model without post-processing or validation.

• Performance evaluation was limited to accuracy, inference time, and token usage metrics.

• No automated difficulty classification mechanism was included in the original system.

• The architecture did not incorporate failure pattern analysis of large language models.

• The system was designed mainly as a comparative evaluation framework rather than an adaptive optimization pipeline



B. Updated System Architecture

- The updated framework introduces an adaptive hybrid reasoning pipeline that dynamically selects the optimal reasoning strategy for each input problem.

- Additional datasets, including SVAMP and Math23K, have been integrated to improve robustness and evaluation diversity.

- A machine learning–based router model has been added to automatically classify problems into Easy and Hard categories.

- A Smart Tree-of-Thought module with limited branching and early stopping has been implemented to reduce computational overhead while preserving reasoning accuracy.

- A preprocessing and feature extraction module has been incorporated to analyze input complexity prior to inference.

- An answer validation layer has been introduced to filter code outputs, numerical formatting errors, and invalid reasoning sequences.

- A failure analysis module has been added to study mathematical reasoning errors of modern LLMs such as ChatGPT and Gemini.

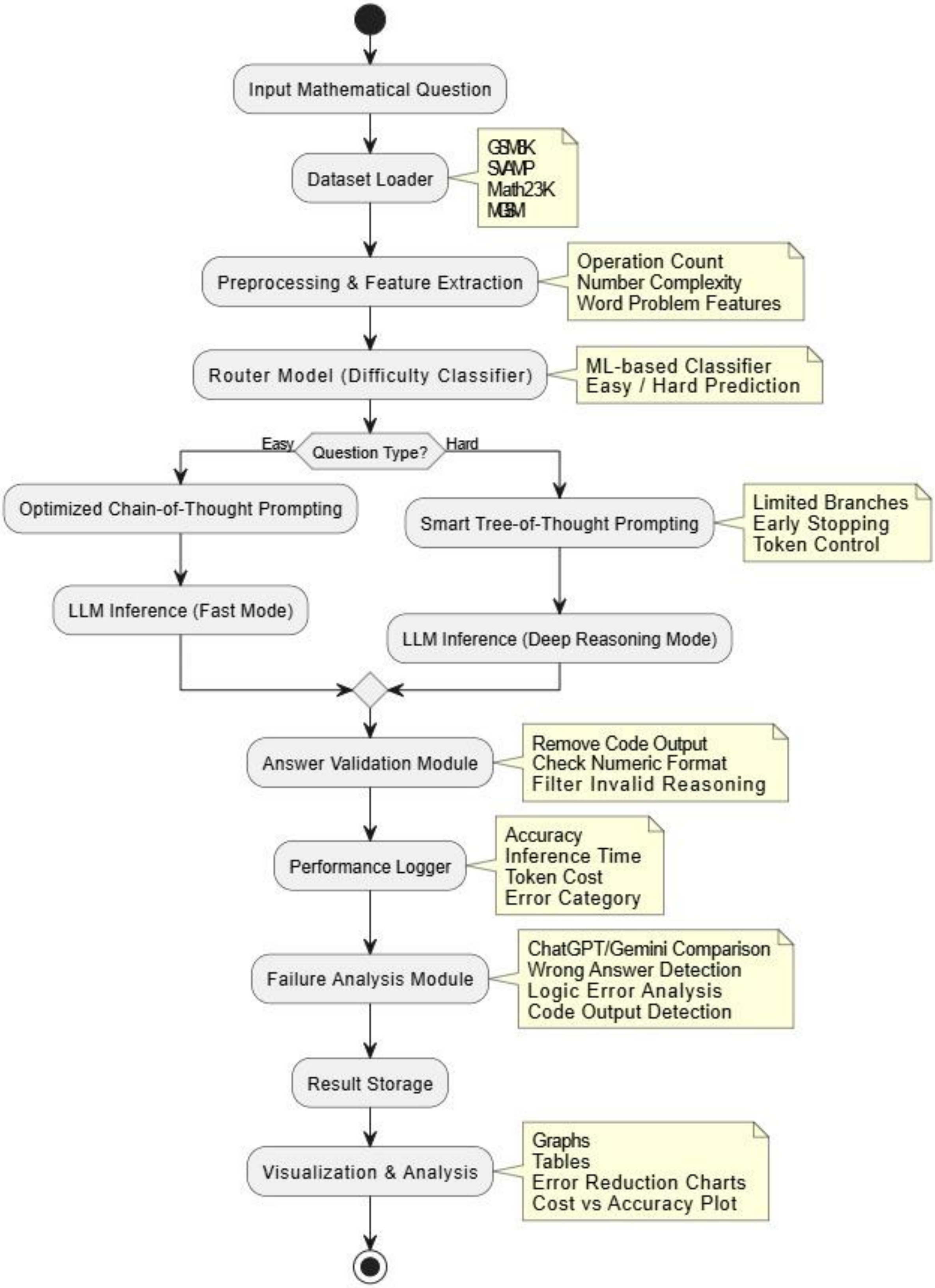
- The performance evaluation module has been extended to include error category tracking and hard-problem success rates.

- Automated performance logging has been implemented to support large-scale experimental analysis.

- The visualization module has been upgraded to generate error reduction plots and cost-versus-accuracy comparisons.

- The updated architecture is designed to support reproducible experimentation and academic publication standards.

Adaptive Hybrid Reasoning Framework (Updated System Architecture)


Input Mathematical Question
Dataset Loader
GSM8K
SVAMP
Math23K
MGSM
Preprocessing & Feature Extraction
Operation Count
Number Complexity
Word Problem Features
Router Model (Difficulty Classifier)
ML-based Classifier
Easy / Hard Prediction
Question Type?
Easy
Hard
Optimized Chain-of-Thought Prompting
LLM Inference (Fast Mode)
Smart Tree-of-Thought Prompting
Limited Branches
Early Stopping
Token Control
LLM Inference (Deep Reasoning Mode)
Answer Validation Module
Remove Code Output
Check Numeric Format
Filter Invalid Reasoning
Performance Logger
Accuracy
Inference Time
Token Cost
Error Category
Failure Analysis Module
ChatGPT/Gemini Comparison
Wrong Answer Detection
Logic Error Analysis
Code Output Detection
Result Storage
Visualization & Analysis
Graphs
Tables
Error Reduction Charts
Cost vs Accuracy Plot