Developing a Data-Driven Feedback System to Support Students' Multiple-Source Learning Processes in LearnNet

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Software Engineering
June 2026
Soumik Roy Bishal

UNIVERSITY OF TURKU
Department of Computing

Soumik Roy Bishal: Developing a Data-Driven Feedback System to Support Students' Multiple-Source Learning Processes in LearnNet

Master of Science (Tech) Thesis, 95 p., 9 app. p.
Software Engineering
June 2026

In today's digital educational environment, students are expected to find, evaluate, and synthesize information from multiple online resources. The ability to do so is termed "Multiple Source Comprehension" (MSC), and it remains one of the most difficult skills for students to develop. To master these complex tasks, students require continuous teacher support and instructional scaffolding, which is time-consuming for teachers to deliver individually. While artificial intelligence systems can provide some support through automated mechanisms, such as providing instant feedback, transforming students' digital trace records into supportable feedback is a technically challenging task. As such, the purpose of this research is to extend the LearnNet platform's ability to provide automated feedback through new process-oriented feedback formats and to evaluate the system-wide capability of generating AI-based interventions using real-world school contexts.

The primary development contribution of this study includes designing and integrating the Entire Task Feedback System. This system collects raw time-series event log data that represent the activities of students and transforms these logs into a set of quantitative indicators that measure the learning processes of students. Additionally, the full system, including all feedback modalities, was tested in live school trials in Finland.

To establish the factual accuracy of the system, its usefulness as a source of actionable feedback, and comprehensibility, an independent evaluation framework was developed that integrated metrics from the LLM as a Judge with interaction data collected via the user interface. During live trials, an important instructional bottleneck known as the performance inversion trap was identified. Early baseline engines provided high clarity feedback; however, they also had low actionability and delayed executions. A controlled offline experiment was implemented to determine how different models and reasoning budget constraints impacted feedback quality and latency. The results indicate that moving to the newer generation model substantially improved the tradeoff between feedback quality and latency. Specifically, 53.8% of generations delivered high-tier actionable scaffolding while maintaining structurally safe factual accuracy and reduced average round-trip latency from 22.38% to a classroom-acceptable 5.19 seconds per request. Finally, post-task surveys from the pilot group indicated that students perceived the grounded scaffolds as both clear and reliable and displayed no apparent symptoms of systemic AI dependency. Due to the relatively limited scope of the pilot, further research is needed to replicate these results on a larger scale.

Keywords: AI, Learning Analytics, Multiple-Source Comprehension, Large Language Models, Automated Feedback, Educational Design Research, Real-Time Systems.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my academic supervisor, Professor Tuomas Mäkilä from the Department of Computing at the University of Turku, for his constant guidance, insightful advice, and valuable feedback throughout this study.

I am equally grateful to Professor Mirjamaija Mikkilä-Erdmann from the Department of Teacher Education for providing me with the opportunity to work as a research assistant on the FINSCI project. Her leadership created the fundamental framework that made this research possible.

I wish to express my most sincere gratitude to Norbert Erdmann for his guidance, supervision, and feedback during the development and research process, which was invaluable to my academic, professional growth, and the thesis.

I would also like to acknowledge Shing Lau's continuing research advice and feedback on the thesis, as well as Kasperi Rautio's technical expertise and support throughout the development phase of the LearnNet project.

Finally, I want to thank the entire FINSCI research group. I am grateful for their hard work in organizing the school trials, which provided the essential data for this study. I am also thankful to the University of Turku for providing the resources and environment necessary to conduct this work.

Contents

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Problem Statement 3

1.3 Research Objectives 3

1.3.1 Academic Objective 4

1.3.2 Practical Objective 4

1.3.3 Research Questions 4

1.4 Methodology 5

1.5 Declaration of Generative AI 5

2 Technical Background **6**

2.1 LLMs as Probabilistic Engines 6

2.2 The API-Driven Interaction Model 7

2.3 Prompt Engineering as Methodology 7

2.3.1 Shot-Based Strategies 8

2.3.2 Chain-of-Thought (CoT) Reasoning 8

2.4 Reliability: Hallucination and Faithfulness 8

2.5 LLM-as-a-Judge Framework 9

2.6 The Ragas Evaluation Framework 10

2.7 Observability and System Performance 11

3 Pedagogical Foundations and AI Feedback 12

3.1 Multiple-Source Comprehension and Learning Strategies 12

3.2 Learning Analytics and the "Semantic Gap" 13

3.3 Data-Driven Feedback and Process Indicators 14

3.4 Theoretical Foundations of Digital Scaffolding 15

3.5 Structural and Instructional Characteristics of Well-Formed Educational Feedback 17

3.6 Pedagogical Views of AI-generated Feedback 18

3.6.1 Faithfulness in Pedagogical Settings 18

3.6.2 Diagnostic Gaps and Structural Quality 18

3.7 LLM as Judge in Education 19

3.8 Multi-Agents Governance and Benchmarks 19

3.9 Research Synthesis and Gaps 20

4 Research Methodology 21

4.1 Educational Design Research (EDR) framework 21

4.2 Data Sources and Data Collection Methods 22

4.2.1 System Input Data: Creating Context-specific Feedback 23

4.2.2 Evaluative Data: Validating Feedback Quality 23

4.2.3 Student Feedback Literacy Index Psychometrics: Baseline for Student Traits 24

4.3 Research Procedure: The Student Trial 25

4.3.1 General Procedure and Tasks 26

4.3.2 Environment and Participants of the Trial 26

4.4 Automated Evaluation System Ragas 27

4.5 Data Analysis and Statistical Framework 28

4.6 Ethical Considerations 30

5 Phase 1: Analysis and design 31

5.1 Data Modeling and Abstraction Framework 31

5.1.1 Raw Interaction Schema 32

5.1.2 Extraction of high-Level metrics 32

5.2 Pedagogical Operationalization of Quantitative Metrics 33

5.2.1 Temporal Patterns in the inquiry life-cycle 34

5.2.2 Strategic Search and Source Evaluation 34

5.2.3 Intertextual Integration Profiles 35

5.3 Conceptual Architecture of the Feedback System 35

5.3.1 Cognitive Grounding via the MD-TRACE Model 35

5.3.2 Scaffolding Mechanics: The Glow and Grow Framework 36

5.4 Optimizing Design Requirements for an LLM Prototype System . . . 38

5.4.1 Information Pipeline and Context Generation 38

5.4.2 LLM Quality constraints 38

6 Phase 2: Development 40

6.1 System Architecture Overview 40

6.2 Frontend Development (React & Mantine) 41

6.2.1 Modular Component Design for MSC Phases 42

6.2.2 The Feedback Interaction UI 42

6.2.3 State management and rendering via SSE 43

6.2.4 Telemetry Integration 43

6.3 Orchestration Backend (Node.js & Express) 43

6.3.1 Persistence Layer and Schema Design 44

6.3.2 Telemetry Pipeline: High Frequency Event Logging 44

6.3.3 Service Management And Security 45

6.4 Feedback Microservice (FastAPI) 45

6.4.1 The QPI / SAP Metric Engine 46

6.4.2 LLM Orchestration and Thinking Budgets 46
6.4.3 Implementation of Streaming Feedback (SSE) 46
6.4.4 Data Validation and Schema Safety 47
6.5 Observability and Telemetry 48
6.5.1 Real-time Trace Capturing 49
6.5.2 Linking Student Activity to AI Inference 49
6.5.3 User Feedback Loop and Error Monitoring 49
6.6 Independent Evaluation and Analytics Infrastructure 50
6.6.1 Stage 1: Multi-Source Data Alignment. 50
6.6.2 Stage 2: Evaluation with Custom Rubrics. 50
6.6.3 Stage 3: Mixed-Methods Statistical Synthesis. 51
6.6.4 Stage 4: Automated Behavioral Abstraction. 52
6.6.5 Stage 5: Reasoning-Effort Experimental Replays. 52
6.6.6 Stage 6: Final Report Generation. 52

7 Results And Evaluation 55

7.1 Evaluation Context Overview 55
7.1.1 Validation of Modeled Student Process Indicators (RQ1) . . . 56
7.1.2 Mapping Quantitative Profiles 56
7.1.3 Validation of High-Tier MSC Strategies 57
7.1.4 Validation of Low-Tier and Bottlenecked Strategies 58
7.2 Baseline Informational Faithfulness and Semantic Quality Audit (RQ2) 59
7.2.1 Baseline Field Performance: Ragas Audit and the Invariance Paradox 60
7.2.2 Qualitative Case Analysis and Field Grounding 63
7.2.3 The Limitations of the Faithfulness Audit 66
7.3 Experimental Reasoning Optimization and Performance Trade-offs (RQ2 & RQ3) 66

7.3.1 Aggregated Experimental Results 68
7.3.2 Semantic Quality and Instructional Safety (RQ2 Focus) 68
7.3.3 System Performance and Live Deployment Trade-offs (RQ3 Focus) 70
7.3.4 Architectural Synthesis and Deployment Recommendations . . 72
7.4 User Adaptation, Scaffold Triggers, and Literacy (RQ4) 72
7.4.1 Distribution and Sentiment of Interface Scaffolding Modalities 75
7.4.2 Conversational Navigation and Follow-up Sub-Question Selection 76
7.4.3 Analysis of Student Perception across Interface Modalities . . 78

8 Discussion **81**
8.1 Discussion of Key Findings and Synthesis 81
8.1.1 Modeling Complex Skill Frameworks from Raw Telemetry (RQ1) 81
8.1.2 Algorithmic Trade-offs and the Performance Inversion Trap (RQ2 & RQ3) 82
8.1.3 Interface Interventions, Trust, and Learning Friction (RQ4) . . 87
8.2 Implications for Theory and Practice 89
8.2.1 Theoretical Implications 89
8.2.2 Practical Implications 90
8.3 Limitations and Directions for Future Work 91

9 Conclusion **93**
9.1 Research Summary and Contributions 93
9.2 Key Technical and Pedagogical Lessons 94
9.3 Concluding Remarks 95

References **96**

Appendices

A Evaluation Instruments **A-1**

A.1 Feedback Literacy(FL) Questionnaire A-1

A.1.1 Feedback Literacy (FL) Scale Items A-1

A.2 Post-Task Survey Items A-2

B Trials Tasks **B-1**

B.1 Task 1: EN126 . . . . . B-1

B.2 Task 2: EN226 . . . . . B-2

C LLM As A Judge Evaluation Rubrics **C-1**

C.1 Faithfulness and Groundedness Rubric C-1

C.2 Actionability and Guidance Rubric C-2

C.3 Clarity and Comprehension Rubric C-3

List of Figures

5.1 System processing pipeline for converting primitive log events into validated formative feedback. 39

6.1 System Architecture: Vertical layout showing the direct frontend-to-API communication and the decoupled offline evaluation pipeline. . . 41

6.2 The Refined Two-Pillar Data Flow: Demonstrating how the Metric Engine orchestrates data from both the temporal log stream (`eventlogs`) and relational artifact tables to construct grounded prompts. 48

6.3 The expanded 6-stage Evaluation Infrastructure: Integrating statistical analysis with AI-driven behavioral and performance benchmarking. 54

7.1 DPI for Faithfulness 61

7.2 DPI for Actionability 61

7.3 DPI for Clarity 61

7.4 Baseline Distribution Propensity Index (DPI) mapping across replayed database text objects ($N_{\text{rows}} = 192$). Plots illustrate the empirical density allocations for Trial EN126 (light gray, $N = 93$) and Trial EN226 (dark gray, $N = 99$) across the primary evaluation dimensions. 61

7.5 System Architecture Replay Pipeline and Factorial Experimental Workflow. 67

7.6 Factorial performance scaling comparison highlighting non-linear computational latencies between engine generations ($N = 93$). 71

List of Tables

4.1 Summary of Data Sources and Purpose 25

5.1 Architectural Alignment of System Payloads and Evaluation Rubrics 36

5.2 System Metric Categories and Target Pedagogical Strategies 37

7.1 Summary of Trial Engagement and Feedback Generation 56

7.2 Extracted Quantitative Process Indicator (QPI) Profiles for Representative Students 57

7.3 Non-Parametric Descriptive Matrix and Mann-Whitney U Test Profiles for Baseline Ragas Metrics 60

7.4 Empirical Sampling Matrix of Real Database Baseline Evaluation Logs 66

7.5 Operational Statistics and Rubric Level Distributions for `gpt-5-mini` 68

7.6 Operational Statistics and Rubric Level Distributions for `gpt-5.4-mini` 69

7.7 Granular Student Feedback Sentiment by Evaluation Criterion 73

7.8 Distribution and Multi-Dimensional Student Sentiment across Interface Modalities 75

7.9 Student Follow-up Sub-question Selection and Sentiment Analysis . . 77

7.10 Student Post-Task Perceptions of LearnNet Interface Interventions . . 79

1 Introduction

1.1 Background and Motivation

The skill of transitioning from “learning to read” to “reading to learn” has become an increasingly important part of contemporary education, even at the elementary school level [1]. Since education is moving toward complex digital environments, where students at every educational level will be required to navigate conflicting information, assess the reliability of each piece of information, and synthesize various points of view, it is required for all students to have developed the skills necessary for what is referred to as Multiple-Source Comprehension (MSC). MSC is not only important for preparing students to succeed in post-secondary education but also for the long-term scientific literacy of future citizens [2].

Unfortunately, research indicates that this set of skills is a major area of concern for students at all levels, including prospective teachers. Students often produce syntheses based on irrelevant or false content. Not only do they have difficulty evaluating whether a source is reliable and valuable for a given task, but they also have difficulty finding and connecting the main ideas within several sources [3].

One reason why students may experience difficulties with these skills is due to the "black box" nature of traditional assessments, whereby educators are able to judge the final product produced by the student without being aware of the cognitive processes used during the completion of that product. The use of open-ended

learning environments (OELEs) offers one method to overcome this issue. OELEs capture extensive log data regarding how students interact with the environment, providing insight into inquiry-based processes that could never be captured by final products [4].

For learning purposes, Multiple-Source Comprehension (MSC) tasks are often designed structurally as an online inquiry workflow [5] with the purpose of providing guidance to students during the construction of their own mental representation from each document [6]. Therefore, if educators wish to use these digital environments to provide adaptive, temporary scaffolding throughout this process [7], the underlying technology must offer more than passive logging capabilities. The LearnNet platform, a specialized, closed learning environment developed by the Department of Teacher Education at the University of Turku, addresses this challenge. It provides a structured workspace where students search for information, evaluate source reliability, bookmark relevant materials, extract core textual snippets, and ultimately synthesize their findings into a final cohesive essay. By breaking down the complex process of online research into these manageable, successive stages, the platform maps a visible workflow onto students' cognitive comprehension strategies.

The focus during the development phase is on the Entire Task Feedback while also analyzing and evaluating the feedback mechanisms from other phases (Bookmark, Snippet, Synthesis) so as to assess how a student develops throughout the entire inquiry cycle (Searching, Bookmarking, Snippeting, and Synthesizing).

A dual-Analysis approach is utilized when generating feedback in the feedback system:

- ***Process Analysis***: Log data is used to evaluate behavioral patterns of the student, such as time-on-task, navigation sequences, and source selection strategies [8].

- ***Content Analysis***: Semantics-based evaluation of the quality of student

artifacts, such as whether a selected text snippet was relevant or whether a written synthesis was coherent.

This allows for holistic guidance to be provided by the system. A student who selects irrelevant sources can receive Bookmark Feedback; a student who has difficulty identifying main ideas within a text can receive Snippet Feedback. Thus, the goal of the system is to move away from retrospective judgment of final products towards continuous data-driven dialogue that supports the development of complex inquiry skills, and students learn complex phenomena like energy, animal adaptation, etc.

1.2 Problem Statement

The central problem addressed in this study is the need to create a data-driven feedback system to support students' multiple-source learning processes using the LearnNet platform. In educational contexts, Multiple-Source Comprehension (MSC) is applied through a set of inquiry skills (comprising searching, bookmarking, snippeting, and synthesizing), which serves as the visible workflow for students to enact their hidden, cognitive comprehension strategies [4], [8]. Research shows a gap in effective methods of processing unstructured log data generated by these systems in a generalized way [4]. Without a real-time processing method for these log data, automated interventions are restricted to analyzing past task performance metrics. This does not provide a dynamic level of strategic scaffolding needed for developing students' long-term self-regulated learning behaviors or digital literacy skills.

1.3 Research Objectives

To address the stated problem, this study defines both academic and practical Objectives:

1.3.1 Academic Objective

The primary academic objective of this study is to develop and validate a learning analytics methodology through creating a dual-layered modeling approach using quantitative process indicators from log data and semantic content indicators from student artifacts. Additionally, this study will assess complex Multi-Source Comprehension strategies and investigate the feasibility and effectiveness of using Large Language Models (LLMs) to generate constructive feedback while also evaluating the impact of model reasoning effort on diagnostic quality and age appropriateness.

1.3.2 Practical Objective

The practical objective of this study is to design, implement, and evaluate a fully integrated full-stack feedback system within LearnNet that provides targeted interventions across five specific dimensions: Bookmark, Snippet, synthesis, and task feedback. This will involve orchestrating interactions between process indicators and content Analysis to produce pedagogical guidance while optimizing system latency and identifying pedagogical bottlenecks through student feedback loops.

1.3.3 Research Questions

This study is directed toward the following four research questions.

1. **RQ1:** How can quantitative learning process indicators be automatically modeled and extracted from LearnNet student event logs?

2. **RQ2:** How accurately does AI-produced feedback use the real-time assignment data inputted by the student, and to what degree does it avoid "pedagogical hallucinations"?

3. **RQ3:** What effect does reasoning effort have on the quality, security, and latency of AI-generated feedback within a real-time learning environment?

4. **RQ4**: How do students evaluate/interact with the developed system?

    - **RQ4.1**: What are students' opinions about the usefulness and completeness of the provided feedback?
    - **RQ4.2**: How do students implement feedback into their current working environments?
    - **RQ4.3**: Which pedagogical obstacles to implementation does an evaluation by the students show?

1.4 Methodology

This study follows the Educational Design Research (EDR) framework, structured across three iterative phases: analysis of raw student telemetry, software prototyping and implementation, and systematic evaluation of the completed feedback pipelines. A comprehensive description of the operational parameters, data collection sources, and classroom deployment environments is provided in Chapter 4.

1.5 Declaration of Generative AI

Generative AI tools were used during the design and development process of this thesis in a limited, supervised, and well-controlled way. The AI was utilized to some extent to assist with improving the writing, create an academic tone to improve readability, and make other technical portions easier to read - such as improving how the system's evaluation metrics and architectural frameworks are presented.

2 Technical Background

The progression from raw behavioral data to educational guidance demands a technological pipeline able to interpret complex student behavior. The application of Generative AI (GenAI) - particularly large language models (LLM's) – represents an increasingly viable means by which educational applications achieve this goal.

Although the end purpose of this technology is educational in nature, its technical realization is dependent upon three unique computational paradigms; namely, the probabilistic nature of transformer-based models [9]; the structure of Application Programming Interface (API) architectures; and the emergent field of automated evaluation [10]. This chapter will provide the technical foundation necessary to bridge the "semantics gap" between low-level database log records and high-level natural language feedback.

2.1 LLMs as Probabilistic Engines

From a systems level, LLMs are functioning as probabilistic engines that predict the most probable next "token" given a specified context. Unlike deterministic algorithms, LLMs represent a flexible translation layer capable of transforming quantitative data, i.e., "process indicators," into qualitative, natural language feedback. Due to their probabilistic nature, system reliability is contingent upon proper calibration of the inference parameters to assure output stability [11].

2.2 The API-Driven Interaction Model

Following industry standard guidelines [12], the interaction with LLMs within GenAI features is typically implemented using a RESTful API. This paradigm separates global instruction from dynamic data in order to enable multiple roles to interact with a single LLM. The roles include:

1. **System Message:** This element specifies the operational "persona" and establishes global constraints. For example, in educational contexts, the system message can define the operational persona and enforce pedagogical constraints. Additionally, it can also define constraints related to tone and age appropriateness [13].

2. **User Message:** This is the dynamic payload that contains the particular context to be processed. Processed within a learning analytics pipeline, engineered process indicators [8] are typically injected into user messages to provide the model with a snapshot of student behavior.

3. **Inference Parameters:** These technical settings – e.g. Temperature – influence the variability of the model selecting a token. Generally speaking, lower temperatures (i.e., 0.1 to 0.3) are preferable for feedback systems in order to assure that generated responses remain consistent and repeatable [14].

2.3 Prompt Engineering as Methodology

Prompt engineering is described as the process of optimizing input formats to identify the reasoning path of a model without modifying the underlying model weights.

2.3.1 Shot-Based Strategies

Research has shown that "few-shot" prompts result in higher levels of structural alignment and stylistic consistency when compared to "zero-shot" instructional inputs [13]. Few-shot prompts provide a structural template that enables the model to map raw data to specific educational standards.

2.3.2 Chain-of-Thought (CoT) Reasoning

To reduce error associated with the interpretation of data, CoT is used to cause the model to generate intermediate reasoning tokens prior to producing the final output [15]. Specifically, by instructing the model to "reason step-by-step", the system causes the model to produce explicit intermediate reasoning tokens prior to producing the final output. CoT is especially useful for analyzing complex behavioral logs since it allows the model to "validate" each indicator in the logs prior to producing a summary.

In addition to simply providing step-by-step instructions, current reasoning models allow for the configuration of a thinking budget [16]. The thinking budget influences how much internal reasoning effort the model exerts prior to producing a response, making it critical for complex data interpretation tasks where validation of student indicators is necessary.

2.4 Reliability: Hallucination and Faithfulness

A significant problem with all generative systems is establishing “faithfulness” to the input data. *Hallucinations* refer to generated responses that are linguistically correct, but lack a factual basis; they include many types of responses that have been categorized by Jia et al. [17] as follows:

1. **Intrinsic Hallucination:** The response directly contradicts the data pro-

vided. For example, if an AI tutoring system states that a student finished a learning activity when the log files clearly indicate that the student did not complete it [17].

2. **Extrinsic Hallucination:** The model produces responses based upon its own internal training data "leaked" into a specific student interaction, and neither the produced response supports nor contradicts the actual data provided [17].

2.5 LLM-as-a-Judge Framework

As an additional solution to ensure quality control for large volumes of generative content, researchers developed the LLM-as-a-judge paradigm for automatic, reference-free semantic validation. This paradigm involves two models in a multi-agent architecture. First, there is a primary generative model that provides students with natural language-based outputs. Second, there is a high-parameter secondary model referred to as an “evaluator”. This evaluator uses structured rubrics to evaluate each component of the text based upon specific instructional qualities and then assigns a numerical score. Software development pipelines can use these evaluations to automatically identify low-scoring generations and eliminate them prior to their being delivered to users [11], [13], [18].

For measurement theorists who study how students learn through instructional technology, using a Large Language Model as an automated judge creates an evaluation environment governed by ordered, ordinal data attributes as opposed to uniform interval scales. For example, traditional natural language processing measures of performance, such as token perplexity or the distance between similar tokens (i.e., similarity distances), measure the magnitude of differences among discrete levels of a particular attribute. In contrast, rubric criteria used to assess educational programs track qualitative characteristics of behavior that exhibit a monotonic quality trend.

Therefore, since semantic thresholds associated with each level of a rubric represent nonlinear transformations rather than fixed mathematical distances, statistical estimates (e.g., mean [M]) computed over qualitative traits cannot be assumed to reflect the same level of mathematical difference between each successive scoring tier. Rather, these statistical estimates serve as an indicator of likelihood (or propensity), measuring how far away from the system's metric center-of-gravity each successive scoring tier lies relative to another.

2.6 The Ragas Evaluation Framework

Contemporary software development pipelines are increasingly using specially designed frameworks such as Ragas (Retrieval-Augmented Generation Assessment) [19] to implement reference-free semantic validation at scale. Traditional Natural Language Processing is usually evaluated with token matching metrics such as BLEU or ROUGE, but they measure how much of the original sentence was repeated, not what was said semantically, thus failing to capture the multiple dimensions needed to accurately validate an educational interface.

The Ragas platform eliminates this problem through formalized computational statements that define LLM-as-a-Judge Paradigm [18], utilizing a high-parameter secondary model to programmatically deconstruct a generated payload to verify the factual content of its base against the base context array provided to the generation engine [11]. As the evaluation process can be made independent of the string matching architecture used for each evaluation, it will allow for the systematic assessment of semantic attributes along monotonically increasing quality trends. It also gives us a mathematical way to determine where we have gone wrong during our testing, including when we hit structural bounds of the system and whether we see some form of inherent or extrinsic hallucination due to heavy load on context [17].

2.7 Observability and System Performance

To balance the competing demands placed upon reasoning quality versus timeliness within real-time applications, the system employs technical observation tools (e.g., Langfuse). These allow for exact tracking of latency, token usage and reasoning paths. One important consideration in this optimization process is the concept of the *thinking budget*, which represents the maximum number of internal tokens that can be created for intermediate verification purposes before providing feedback to students [16], [20].

3 Pedagogical Foundations and AI Feedback

While Chapter 2 provided an overview of the technical mechanics of large language models, this chapter provides a deeper examination of the theoretical frameworks of automated systems and research conducted using learning science and learning analytics methods. The cognitive challenges of multiple-source comprehension (MSC) are examined along with the role of process indicators in bridging the "semantic gap" and the potential pedagogical applications of generative AI in student-centered learning environments.

3.1 Multiple-Source Comprehension and Learning Strategies

Students today find themselves in environments where they are being asked to assess and use information found in many different, sometimes conflicting, sources. In other words, this process is called Multiple-Source Comprehension (MSC), and it requires learners to perform higher-order tasks such as connecting main ideas across texts while also assessing the credibility of each source [4]. Additionally, unlike when one reads a single text, MSC requires learners to identify conceptual relationships among multiple texts while at the same time evaluating the credibility of each source [2]. To

represent these complex cognitive steps, the MD-TRACE (Multiple-Document Task-Based Relevance Assessment and Content Extraction) model is used in cognitive psychology [21]. The multiple-source comprehension process is conceptualized by this model as a non-linear multi-step process. Learners develop an internal task representation, they search for and evaluate material, and extract specific content to synthesize a final artifact [21]

Research often uses the Inquiry-Based Learning (IBL) cycle to break down these tasks into five separate phases: Orientation, Conceptualization, Investigation, Conclusion, and Discussion [5]. In systems like LearnNet, the four core IBL phases correspond to observable digital behaviors: searching, source evaluation and bookmarking, creating snippets of main ideas, and finally writing a synthesis based on the snippets created [8]. Therefore, to support the development of effective processes throughout these stages of IBL, a feedback mechanism would need to be developed that supports all three levels of feedback: Feed Up (clarifying goals), Feed Back (evaluating progress to date), and Feed Forward (suggesting possible strategies for completing future tasks) [22], [23]. Although the IBL cycle establishes this framework, studies have shown that students are still unable to effectively create a number of deep connections between texts [3]; therefore, there is a growing trend toward developing personalized, AI-based scaffolds for use in post-secondary education [24].

3.2 Learning Analytics and the "Semantic Gap"

One of the greatest benefits of Open-Ended Learning Environments (OELEs) is their ability to allow researchers to look inside the “black box” of student process performance by analyzing detailed records of students’ digital behavior. These detailed records of digital behavior – referred to as digital traces enable the analysis of the skill performances of the students in real-time [4]. However, despite this wealth of data available for analysis, a major issue facing the field of Learning Analytics (LA)

is what is commonly referred to as the "semantic gap." The semantic gap refers to the challenge of converting digital traces into meaningful educational insights [4]. This study bridges this semantic gap by combining behavioral log analysis (analyzing how students worked) with artifact analysis (analyzing what students understood about their own critical thinking and/or self-regulated learning) relative to understanding the depth of a student's critical thinking or self-regulatory abilities [4].

Assessment is traditionally a backward-looking activity and typically focuses on the final product resulting from the learning process, rather than on the developmental processes used by the student to arrive at that product [4]. Process-oriented feedback represents an attempt to address this limitation by identifying specific points of low performance within an inquiry process that are likely responsible for producing less-than-optimal products [4]. For example, if a student produces a suboptimal synthesis, this could potentially result from inadequate evaluation of individual sources rather than due to some deficiency in writing skill [4]. Identifying these specific instances of low performance enables educators to develop targeted data-driven scaffolding that goes beyond simply collecting and recording student data and instead facilitates a form of active educational dialogue with students [8].

3.3 Data-Driven Feedback and Process Indicators

Bridging the semantic gap involves the application of two types of indicators: "process indicators" (log-based workflow proxies) and "content indicators" (artifact-based semantic-quality proxies; e.g., snippet relevance, source reliability) [8]. Indicators such as switching frequency (indicating inter-textual integration), time-on-task distribution (distinguishing between rushed syntheses), and source evaluation strategy (ratio of relevant to irrelevant bookmarks) can be engineered from logs and represent ways in which a system can characterize an individual student's inquiry approach without continuous human observation [8]. Recent research has shown

that large language models (LLMs) can analyze educational interaction data and produce results that exhibit high agreement with human expert evaluations, highlighting their potential for supporting educational assessment tasks [25].

3.4 Theoretical Foundations of Digital Scaffolding

Real-time intervention designs in open-ended learning environments are practically grounded in the methodologies of educational design research [26]. The theoretical foundation for designing such real-time interventions stems from pedagogical scaffolding, a concept developed in Vygotsky's sociocultural theory. Pedagogical scaffolding is defined as a temporary, collaborative structure used to support a learner working within their zone of proximal development (ZPD). It provides the necessary guidance that helps them reach the next zone that would not be possible independently [27]. Adaptive, computer-based learning systems have taken the dyadic relationships between the learner and the teacher and adapted them into learning utilities that can adapt to each learner's needs [28]. Digital multi-text inquiry uses learning analytics to develop adaptive scaffolding that provides real-time cognitive and metacognitive prompts tailored to the learner's workflow at specific points. These adaptive scaffolds use the learner's fine-grained process trace data to determine if additional scaffolding is needed [4], [8]. Process-oriented discourse in education allows researchers like Long, Luo, and Zhang (2024) to analyze how large language models (LLMs) interpret these workflow patterns, illustrating that generative agents can achieve high agreement rates with human expert coders when identifying complex classroom dialogic structures and pedagogy [25].

However, deploying generative models to provide real-time scaffolds poses significant instructional risks. Instructional designers must ensure that automated scaffolding provides balanced structural frameworks with appropriate formative feedback dimensions such as praise, critique, advice, and clarification requests to guide

student progress effectively [29]. Because complex online inquiry tasks are inherently difficult, theoretically a reduction can be expected to see in extraneous cognitive load when students have access to structured workflows and organizational frameworks for completing work (i.e., step-by-step processes and templates for taking notes) that will help clarify students' sub-goals and lower their chances of experiencing cognitive overload [30] [31]. Additionally, while these forms of support will primarily benefit novice learners, they may also provide excessive assistance and therefore be redundant or even counterproductive to more experienced learners [32]. A primary pedagogical requirement for an automated educational agent is its ability to accurately identify and remediate student mistakes while providing actionable strategies without prematurely revealing final solutions or key content parameters to the learner [33]. When deployed without explicit algorithmic constraints, standard prompt engines frequently fail to generate well-formed or contextual interventions, leading to significantly impaired structural performance. The factual faithfulness deficiency can manifest as ungrounded or hallucinated claims about student artifacts [17], and the tutoring capability bottleneck can cause systems to become prescriptive "spoon-feeding" by outputting direct answers or lists of instructions rather than supporting independent problem solving [14]. Confirmatory research conducted by Seßler et al. (2025) demonstrated that while frontier llm's display a high utility baseline in science protocols, their diagnostic depth and pedagogical precision decrease dramatically when scaffolding lower secondary cohorts engaging in complex experimental tasks [23]. Therefore, to maintain students' productive engagement within their own learning zone, digital interfaces must deliberately add learning friction. Disabling direct text editing capabilities and automated rewriting capabilities ensures that students actively process hints provided by scaffolding and complete all required text revision work independently [34], [35], [36].

3.5 Structural and Instructional Characteristics of Well-Formed Educational Feedback

Feedback generated by AI is determined both by how accurately it diagnoses a problem and structurally. In higher education research, there is an increasing adoption of the "well-formedness" framework presented by Hughes, Smith, and Creese (2015) to assess the structural characteristics of educational responses [29]. This framework establishes that for an educational response to be considered "well-formed," it must consist of four structural elements, each of which supports the development of formative learning experiences.

1. **Praise (P)** - General positive statements made with the intent to promote motivation among students, although it has been reported that such praise can serve a double purpose, where excessive use can result in appearing insincere [29], [37].

2. **Critique (C)** - Identification of areas in the task, categorized as follows: correction of formal errors, critiques of the content itself, and critiques of the student's methodological approach to completing the task [29].

3. **Advice (A)** - Actionable guidance provided for future tasks, which can include suggestions for improvement for the student regarding either the current task or additional broader advice related to the student's overall learning experience [29], [37].

4. **Clarification Requests (Q)** - Prompts used to facilitate a deeper pedagogical interaction with the student by requesting justification or further clarification relative to unclear aspects of the student's work [29].

Numerous studies employing the above-mentioned profiling tool on a wide range of LLMs indicate that all models will typically generate well-formed Praise and Advice;

however, LLMs vary significantly in their ability to create sophisticated corrective feedback and detect errors in student assignments [37].

3.6 Pedagogical Views of AI-generated Feedback

Although LLMs represent an effective method of providing personalized feedback at a large-scale level, they also introduce unique pedagogical challenges when implemented within educational environments. Some notable challenges include creating a dependency gap wherein students may enhance immediate performance, but upon removal of the AI-based assistance, the students may not retain long-term mastery [36].

3.6.1 Faithfulness in Pedagogical Settings

In a pedagogical setting, one measure of the value of feedback relates to its faithfulness, i.e., whether or not the feedback provided reflects what the student performed in terms of their actual work [17]. Several studies demonstrate that LLMs may produce "hallucinations" for approximately 24% of all instances, thereby criticizing students for errors that were never committed [17]. Hallucinations undermine student confidence and can solidify incorrect assumptions about subject matter [17]. Consequently, grounding techniques are required to limit LLM reliance on a student's log files and artifacts to ensure that the LLM reasonings are based directly on them [10], [11].

3.6.2 Diagnostic Gaps and Structural Quality

Researchers have identified a diagnostic gap in the performance of LLMs; specifically, LLMs tend to perform exceptionally well regarding linguistically clear and positively toned responses. However, recent empirical work by Seßler et al. (2025) demon-

strated that while models can successfully simulate human-like feedback when evaluating expert-level science lab protocols using structured rubrics, they often struggle to provide the same depth of pedagogical insight for middle school(equivalent to the lower secondary level in the Finnish context) cohorts [23]. For an effective feedback strategy to occur, instructional attributes (praise, critique, advice, and clarification requests) must be balanced appropriately to provide contextual feedback [29], [37]. Researchers recommend CoT (Chain-of-Thought) reasoning to require LLMs to review specific process logs before advising students [24]..

When a diagnosis cannot be resolved due to the diagnostic gap, it creates a pedagogical bottleneck and prevents the system from effectively resolving the student's true challenge. Therefore, this study uses a classification scheme to identify "bottlenecks" experienced by students who report "dislikes", such as clarity issues (e.g., confusing language) and scaffolding alignment problems (i.e., providing hints that do not align with the student's immediate needs).

3.7 LLM as Judge in Education

There is a shift from evaluating NLP applications using traditional evaluation metrics (BLEU, ROUGE) towards developing "LLM as judge" frameworks that enable reference-free semantics-based evaluations of feedback quality [10], [11].

A widely referenced framework in this area is the Dean of LLM Tutors Project's 16-dimensional taxonomy that provides an evaluation of feedback along three dimensions: Content, Effectiveness, and Types of Hallucinations [10].

3.8 Multi-Agents Governance and Benchmarks

To enable high-quality monitoring at scale, the Dean of LLM Tutors (DeanLLM) Architecture employs a multi-agent governance model whereby a highly successful

model serves as a "gatekeeper" to screen out untrustworthy or unhelpful responses [10].

Empirical assessments have demonstrated that fine-tuning LLMs enables them to reach human expert levels in identifying pedagogically relevant errors [10], [13].

To evaluate the high-level tutoring skills of a model, benchmarking tools have been developed, namely TutorBench (2025) and MRBench (2025). TutorBench measures a model's ability to provide adaptive explanation while guiding students through productive struggle without providing premature resolution [14], [33].

Both benchmarks consistently show that while frontier models possess extensive knowledge bases and capabilities, they often require calibration in terms of adapting instruction to meet individual students' knowledge deficiencies [14].

3.9 Research Synthesis and Gaps

There is currently a gap between how well platforms such as LearnNet capture behavioral data [8] and how frameworks such as DeanLLM provide rigorous evaluations [10]. The real-time incorporation of raw, quantifiable measures (i.e., switch rate) into the qualitative reasoning of an LLM has been unexplored. This study will build on the two previous studies to convert raw quantitative inquiry logs into a pedagogically relevant and contextually adaptable feedback system that is factually faithful.

4 Research Methodology

The methodology for this study utilizes the Educational Design Research (EDR) framework [8][26], with an emphasis upon developing an iterative design cycle based upon analysis, design, development, and evaluation. Specifically, the focus of this study is on developing a data-driven feedback system for the LearnNet platform, which will utilize multiple-sourced comprehension and scaffold each student's ability to perform such tasks. In this chapter, we detail the methodologies utilized for collecting the data for the study, the methods employed for evaluating the efficacy of the platform utilizing Ragas as an automated evaluator for generative feedback quality, and the design requirements for providing each user with real-time pedagogical support.

4.1 Educational Design Research (EDR) framework

To achieve its research objectives, this study has adopted the Educational Design Research (EDR) framework [26]. The EDR framework represents a combination of applied software engineering techniques, together with rigorous scientific investigation of learning processes as they occur in real classrooms. While traditional software engineering lifecycles have focused primarily on assessing the technical feasibility and performance of software systems, EDR seeks to integrate the cognitive demands placed upon students when completing their schoolwork and to develop both practical tools and theoretical foundations for education [26].

For purposes of conducting this study, the fluid iterations of the EDR method have been structured into three discrete thesis phases so that they may be conducted with sufficient rigor and technical viability:

1. **Phase 1: Analysis and Design:** A thorough examination of all student activity in the LearnNet database was conducted to determine where students would experience significant difficulties in navigating multi-source readings. The Results of this analysis were then used to inform the design of the various system indicators.

2. **Phase 2: Development:** The system prototype was developed via a combination of front-end code written in React, back-end orchestration layers implemented with Node.js, and a microservice-based architecture employing FastAPI to provide automated pedagogical feedback to students.

3. **Phase 3: Evaluation and Results:** An additional Evaluation pipeline was designed to process previously recorded classroom data offline via ragas [11], [17], [19] to create a basis for auditing machine-generated text against accepted educational standards.

This approach creates a mature, low-latency feedback system inside LearnNet, while improving our theoretical understanding of how LLMs can evaluate student learning strategies at scale.

4.2 Data Sources and Data Collection Methods

The data collection efforts serve two primary goals: to collect input data for generating feedback and to validate the effectiveness of the system. These are categorized as *System Input Data* and *Evaluative Data* [8].

4.2.1 System Input Data: Creating Context-specific Feedback

To develop relevant and useful feedback, the system collects input data from two different levels of abstraction within the LearnNet environment [5], [8]:

1. **LearnNet Event Logs (Process Data):** All events initiated by students during their use of the system are stored in a high-volume time-stamped PostgreSQL database. Examples include page view, bookmark click, and snippet creation. This raw log data is transformed into quantitative process indicators that are used to assess the student's MSC strategy [4], [8].

2. **Student Artifacts (Content Data):** The content generated by students during their engagement with the system – e.g., search terms, evaluation of source credibility, extracted snippets, etc.- represents the "what" aspect of the student's learning process [2]. Analysis of this generated content permits the identification of coherence among inter-textual links and the relevance of extracted information [2], [3].

4.2.2 Evaluative Data: Validating Feedback Quality

To enhance the quality of generated feedback and ultimately meet educational standards, four forms of evaluative data are collected [38]:

1. **Pedagogical Quality and Ragas Metrics for LLMs:** To ensure that generated feedback is consistent with established pedagogical principles, Ragas is used to assess fidelity and answer relevance(see Section 4.4). Use of ragas prevents model hallucinations while maintaining adherence to pedagogical frameworks [11], [17], [19].

2. **System Performance and Observability:** Efficiency metrics related to technical performance are evaluated via Langfuse and customized benchmark scripts. The former evaluates end-to-end latency under conditions of varying complexity in process indicators, prompts, thinking budgets, etc. [12], [16], while the latter measures token usage; backend throughput; etc. [20], [23].

3. **Perceptual and Usability Data (Student Surveys):** Structured survey instruments utilizing Likert scales are administered post-inquiry task to gather quantifiable perceptual responses from students about dimensions of usefulness, clarity of feedback provided, motivation, etc. [24], [37].

4. **Real-Time Feedback Evaluation:** A binary "Like/Dislike" telemetry mechanism was exposed via the user interface in order to obtain contextual validation metrics during task execution. This enabled the students to judge each line of feedback based on two separate sub-criteria:

    - **Actionable Utility (Feed-Forward):** Evaluated via the prompt: *"Auttoiko tämä ymmärtämään, mitä tehdä seuraavaksi?"* (Did this help you understand what to do next?).
    - **Linguistic Clarity (Readability):** Evaluated via the prompt: *"Oliko palaute selkeää ja helppoa ymmärtää?"* (Was the feedback clear and easy to understand?).

4.2.3 Student Feedback Literacy Index Psychometrics: Baseline for Student Traits

This paper uses the Feedback Literacy (FL) index developed by the LearnNet research initiative at the University of Turku [39] to connect individual student traits with active session telemetry. This link is implemented through the matching of an individual student's questionnaire scores to their related, specific events recorded on

the database based on a unique student ID. Combining both datasets enables the analysis of how students' overall self-regulated learning behavior patterns compare to their actual behaviors while utilizing the software (e.g., frequency of clicking on feedback, amount of time spent reading vs. writing). The FL index is a single composite metric derived from a well-established seven-item Likert scale instrument used during the cohort research cycle (see Appendix A), where each item corresponds to one or more of the primary components of self-regulated learning, namely, attentive listening, proactive tracking, and emotional regulation.

Raw scores were converted using the repository preprocessing protocols in the unified project database, thus reversing coding all scores so a score of 5 represented the highest level of behavioral alignment (`Strongly Agree`) in relation to each item. In general, higher mean values indicate greater levels of self-regulated learning. As such, it provides a valid psychological measure that pairs with LearnNet's process measures to provide the foundation for Chapter 7's qualitative examination of the user cases studied.

Table 4.1: Summary of Data Sources and Purpose

Data Source	**Data Type**	**Purpose**
LearnNet Logs	Event Logs	Extracting MSC process indicators [8]
Student Artifacts	Text/Snippets	Evaluating semantic quality and synthesis [2]
LLM Outputs	Ragas Scores	Validating accuracy and faithfulness [19]
Student Surveys	Likert/Interviews	Assessing usability and perceived usefulness
Like/Dislike	Binary	Feedback evaluation on individual level

4.3 Research Procedure: The Student Trial

A controlled student trial was carried out inside the LearnNet learning environment to assess the feedback mechanism in a real-world scenario. Instead of using simulated user personas or an isolated laboratory setting, it was important to execute this field trial in a live classroom setting to establish the ecological validity of the system. The

fact that the system would be exposed to real cognitive loading, real spontaneous multiple-source learning issues, and real student behaviors is a factor that cannot be represented by the controlled environments typically used in controlled offline settings.

4.3.1 General Procedure and Tasks

The students have been given the tasks of searching, evaluating, and synthesizing information retrieved from various digital files. The feedback mechanism initiates the "My Process Feedback" modal at defined points (for example, after creating some snippets, and/or when completing the draft of synthesis) for students. Following this, students can interact with the evaluation tools (Like/Unlike) and fill in the final perception to provide the evaluative data as described in Section 4.2.2.

4.3.2 Environment and Participants of the Trial

The participants were 9th-grade students who had a good pre-knowledge of energy topics. Both trials enrolled 20 students each, all of whom were familiar with the LearnNet platform. Both classroom teachers and three researchers observed the sessions. However, participation numbers vary across the different data collection instruments reported in Chapter 7. In EN226, four students did not reach a feedback trigger point during the session, resulting in 16 students generating feedback instances rather than 20. Similarly, 18 of 20 EN126 students and 19 of 20 EN226 students completed the post-task survey. These differences reflect natural variation in task engagement within a live classroom setting and are reported transparently within each respective analysis.

- **Trial EN126:** In Trial EN126, students completed a 60-minute online inquiry task requiring them to write a descriptive, informational text identifying and categorizing different energy sources.

- **Trial EN226:** Trial EN226 involved an advanced 70-minute argumentative task where students wrote a deeply reflective, evaluative essay on responsible energy production strategies specifically tailored to the Finnish context.

4.4 Automated Evaluation System Ragas

The evaluation system uses Ragas (Retrieval-Augmented Generation Assessment)[19], which provides reliability for generated feedback.

Metrics Used for Evaluation: To confirm the educational value of the feedback, three custom rubric-based metrics were developed. These metrics are based on the "LLM-as-a-Judge" paradigm and will be utilized for independent assessment referred to in Chapter 6.

1. **Indicator Faithfulness:** Indicates how accurate the LLM-generated feedback is with respect to the student data.
2. **Actionability and Usefulness:** How effective the feedback is in guiding the student towards the next steps.
3. **Linguistic Clarity and Tone:** Assessment of how comprehensive and appropriate the feedback text is for the target students.

These 1.0–5.0 rubric scores represent essentially an ordered sequence, rather than a continuous interval scale; they monitor qualitatively monotonous progressions where the intervals between levels are nonlinear. Therefore, continuous average (M) values calculated over these evaluation profiles do not indicate mathematically identical intervals between rubric levels but operate practically as a statistical indicator of tendency, measuring a central point of gravity of the quantitative quality axis for the system's metrics.

4.5 Data Analysis and Statistical Framework

When quantitatively assessing the systems evaluation rubrics, a pair of non-parametric statistical analysis tools is employed that are specifically designed for ordinal or interval data. The first tool is the Mann-Whitney U test, which is used to assess if there are differences in the distributions of ordinal rankings of two independent groups of implementations [40]. The second tool is Spearman's rank order correlation coefficient (ρ) [41], which is used to examine the direction and strength of the monotonic relationship between student evaluation of the feedback and ordinal feedback quality score. Even though both of these methodologies provide mathematical capability to detect ordinal trend shifts across a discrete evaluation scale, ranked analysis has inherent structural constraints that may hide substantial local variation in quality generated [10].

A major limitation of utilizing the Mann-Whitney U Test is that it creates a "flattening" effect because it uses cumulative ranks rather than individual ranks. This means that a uniform distribution centered about the scale mid-point (i.e., 100% of students scoring at Level 3), and a highly skewed distribution (e.g., 50% of students fail to score at Level 1 and 50% succeed at Level 5) will produce the same rank sum and thus have a non-significant p-value ($p>.05$). Therefore, in automated educational platforms, such as those discussed herein, the artifacts produced by the Mann-Whitney U Test can hide large decreases in performance at the low end of a system's scale due to an appearance of consistent performance [23].

Similar to the Mann-Whitney U Test, Spearman's ρ correlates rank orderings but does not consider the actual value or position on a given scale. Thus, a perfectly positively correlated pair of variables (ρ=1.0) exists regardless of how far down a particular scale each variable lies [2]. As such, standard rank-based analyses require additional parameterization of density boundaries to maintain both analytical accuracy and educational relevance.

The Generalized Distribution Propensity Index (DPI): To resolve the statistical blind spots of traditional rank tests without violating the mathematical constraints of ordinal scales, this study introduces the Distribution Propensity Index (DPI). Let X be a discrete random variable representing an evaluation score assigned to an automated feedback instance within an ordered rubric scale $R = \{r_1, r_2, \ldots, r_k\}$. For a sample population N, where n_i represents the frequency count at level r_i, the empirical probability mass function is defined as $P(X = r_i) = n_i/N$.

The generalized DPI maps data dispersion by partitioning the distribution into m mutually exclusive, pedagogically distinct operational segments bounded by index thresholds t:

$$\mathrm{DPI}_m = \sum_{i=t_{m-1}+1}^{t_m} P(X = r_i) = \frac{1}{N} \sum_{i=t_{m-1}+1}^{t_m} n_i \tag{4.1}$$

where the sum of all segments equals unity ($\sum \mathrm{DPI}_m = 1$). While this framework adapts to multi-tiered configurations across different evaluation tasks, the baseline system evaluation in this study utilizes a two-tier configuration ($m = 2$). This isolates tracking into a Lower-Bound Grounding Risk ($\mathrm{DPI}_{\mathrm{lower}}$, grouping Levels 1–3 to monitor critical system failures) and an Upper-Bound Scaffolding Density ($\mathrm{DPI}_{\mathrm{upper}}$, grouping Levels 4–5 to monitor high-tier process support). By isolating these metrics, the index exposes data movements that leave population medians completely unchanged.

It should be noted that the DPI is not a validated psychometric instrument, it is a descriptive tool. Its threshold boundaries are operationally defined based on the pedagogical interpretation of the rubric levels described in Appendix C, rather than derived from empirical calibration. The index is therefore best read as a way of exposing distributional patterns that rank-based statistics tend to obscure, and not as an independent measure of validity in its own right.

4.6 Ethical Considerations

Adhering to ethical principles for AI in K-12 education, this study prioritizes absolute student privacy and pedagogical safety [38]. The collection and management of all digital traces follow the strict quality assurance and data protection standards mandated by the University of Turku.

Crucially, this research does not collect, process, or track individual student identities, personal characteristics, or identifiable psychological traits. All session telemetry and feedback literacy indices are completely anonymized, utilizing randomized database identifiers (*varatunnukset*) to examine behavioral archetypes at the aggregate level rather than profiling individual learners. This design guarantees transparency and accountability in the automated scaffolding while maintaining strict compliance with institutional data protection guidelines.

5 Phase 1: Analysis and design

The analytical design patterns developed to provide a basis for translation of digital user traces to pedagogical scaffolding are described in this chapter. The system design will follow a two-tiered process to translate raw user interaction logs to high-Level strategy indicators and then apply them to an automated feedback generation framework[8], [21].

5.1 Data Modeling and Abstraction Framework

The design of the feedback engine relies on the combined processing of two classes of data payloads. As such, it provides for the generation of interventions based on both students' process history and synthesized learning artifacts:

1. **Quantitative Process Indicators (QPI):** Statistical summaries derived from interaction frequency and duration – behavioral proxies for self-regulation and workflow.

2. **Semantic Artifact Payloads (SAP):** Text-based data produced directly by the student, including search queries, document bookmarks, and snippets of text written in drafts.

5.1.1 Raw Interaction Schema

Input data consists of time-stamped user logs captured within the inquiry space. To track progress through an instructional cycle, each primitive log message includes structural attributes mapping onto stages of inquiry-based learning [5]. Attributes used for subsequent data Modeling include:

- `action_type`: Signature of event operations (e.g., Execute a search query or save document abstract).
- `from_phase` & `to_phase`: Markers to show the student navigates across workspaces between search results to document reading panels, etc.
- `eventtimestamp`: Temporal values that are very accurate are used to compute delta metrics and step durations.
- `detail`: Metadata providing contextual information about actual strings entered into interface input fields, target data objects.

5.1.2 Extraction of high-Level metrics

A raw log stream has thousands of row-Level entries due to basic interface actions that contain high-frequency noise. Therefore, the system utilizes a processing framework to parse and aggregate inputs. This component filters out extraneous interface records and aggregates sequential interaction chains into structured objects. These objectives transform thousands of row-Level entries into concise behavioral indices. The extracted metrics can be classified into temporal distributions and counts of procedures forming standardized feature vectors describing individual student learning behaviors.

5.2 Pedagogical Operationalization of Quantitative Metrics

In an open-ended digital learning environment, a student's processing strategy is defined as the organized sequence of actions they perform to complete an informational task. Rather than looking only at a student's final text or essay in isolation, a data-driven system analyzes raw database logs to understand the path a student takes as they read and write [4].

By engineering specific features from these raw interaction logs, the software can determine whether a student is working systematically or experiencing a procedural problem [8]. Based on these log patterns, the system categorizes student behavior into three primary strategies:

- **Information Searching Strategy:** To track this strategy, both the total number of searches made and the semantic changes in search terms are measured. If a student uses very few different search terms, then it may indicate a repetitive trial-and-error strategy. Therefore, the lack of keyword expansion may be an area where the student needs assistance [2].

- **Source Evaluation Strategy:** This is calculated by determining how many of the web pages a student views are bookmarked. When a student saves many web pages but spends little time reading those pages, it typically suggests a shallow source collection rather than a critical evaluation strategy [2].

- **Synthesis and Integration Strategy:** We monitor this type of behavior by evaluating how often a student switches between their captured snippets and the final text editor. When a student makes few transitions between their captured snippet and the final text editor, it is likely to indicate that the student is writing without integrating their captured notes. As a result, there

will also likely be poor synthesis of the sources referenced in the final product.

Translating raw trace logs into these strategy profiles allows the system to bridge the gap between simple click data and pedagogical goals. This structural baseline provides the necessary context for the automated feedback engine to select targeted hints that address the student's actual working workflow.

5.2.1 Temporal Patterns in the inquiry life-cycle

Time spent allocating resources among digital workspaces indicates task engagement choices and cognitive planning [4]. The system evaluates these patterns based on two indicators:

- **Task Understanding Delay:** Duration spent analyzing initial instructions before triggering first external activity. Typically, brief delays indicate rushed planning while extended delays suggest structured orientation to tasks.
- **Phase Allocation Balance:** Calculated ratio of time spent active in source selection compared to source analysis, revealing whether learner over-indexes on document collection at the expense of comprehensive evaluation.

5.2.2 Strategic Search and Source Evaluation

The system evaluates search performance by examining how select thresholds align with successful retrieval methods [2]:

- **Search Specificity Index:** Measure tracking the uniqueness of a query by calculating the semantic variation between successive search phrases. Low variability suggests repetitive behavior and a need for refined hints.
- **Evaluation Selectivity Ratio:** Relationship between the total number of documents viewed and volume of bookmarks created. A high ratio indicates rigorous evaluation, while a low ratio indicates indiscriminant saving habits.

5.2.3 Intertextual Integration Profiles

Counts of transitions between distinct workspaces highlight cognitive strategy driving synthesis tasks [28]:

- **Integration Iteration Density:** Measures frequency of transitions between gathering data (bookmarking/note taking) and creating active draft compilations. Frequent adjustments indicate high information integration.
- **Metacognitive Revision Loops:** Tracks instances where the student moves back to previously archived source text after completing the final synthesis workspace. Provides a procedural indicator for verification/fact-checking behaviors.

5.3 Conceptual Architecture of the Feedback System

This represents an advancement in comparison to conventional assessment models, which are primarily based on post-hoc grading as opposed to real-time formative instructional dialogue [22], [24].

This architecture is aimed to provide visibility to elementary-aged students regarding their implicit inquiry strategies by providing them with formatively aligned, automated, and contextualized feedback.

5.3.1 Cognitive Grounding via the MD-TRACE Model

The feedback architecture is cognitively grounded to match the process steps of the MD-TRACE (Multiple-Document Task-Based Relevance Assessment and Content Extraction) model [21], such that when the systems intervene, it will be targeted to specific cognitive process steps involved during the task completion cycle:

- **Internal Task Representation:** Each of the feedback modules has been designed to intervene at the internal representation stage of the task cycle, specifically after the exploratory phase, to promote goal-oriented search and preliminary planning.
- **Information Searching and Evaluation:** System boundaries around each student's searching activity and bookmarks assist the student in identifying relevant vs. Non-relevant information sources.
- **Information Extraction and Integration:** When the student compiles snippets from various materials and makes revisions to their first draft, injected contexts support integration and synthesis of the evidence.

Table 5.1: Architectural Alignment of System Payloads and Evaluation Rubrics

Feedback Dimension	**Injected Payload Context**	**Instructional Scaffolding Target**
Source Selection	Search Terms + Saved Snippets	Evaluates if source collection matches explicit task goals or relies on superficial keyword matching.
Content Processing	Excerpt Highlights + Reference Data	Analyzes textual extraction accuracy against verified baseline document segments.
Synthesis Composition	Active Draft Text + Saved Snippets	Verifies if assertions within the written text are logically supported by previous excerpts.
Task Strategy	Complete Aggregated QPI Log Vector	Assesses macro-level habits, including linear workflows versus unstructured trial-and-error patterns.

5.3.2 Scaffolding Mechanics: The Glow and Grow Framework

Every feedback intervention uses a structured two-stage pedagogical structure to provide students with both support and clarity [29]:

- ***Glow Segment (Feedback/Affirmative):*** Identifies at least one positive behavioral strategy based upon a review of the quantitative log information or textual content from artifacts. For example, *"You spent enough time reading the article regarding the biodiversity of ecosystems before you made a note."*.

- ***Grow Segment (Scaffolding)****:* Uses procedural steps to identify at least one critical strategy gap without providing solutions. For example, *"To improve your search results, please include geographic terms for the specified area in the task instructions."*

Data Analysis Metrics have been defined so that they can be used to define rules. If any Data Analysis Metric exceeds a pre-defined limit, then the template loader will automatically load dynamic scaffolding criteria in the context window. In this way, the engine is functioning as an objective diagnostic tool that provides an immediate response to student behavior. See Table 5.2, which illustrates how metric thresholds correspond to target strategies.

Table 5.2: System Metric Categories and Target Pedagogical Strategies

Target Metric Category	**System Boundary Condition**	**Target Remediation Strategy**
Phase Transition Ratio	Low navigation linearity	Scaffolds a more linear workflow; helps students shift from trial-and-error to systematic reading.
Search Phrase Variance	High query repetition	Triggers query expansion hints; suggests diverse terminology based on reference metadata.
Evaluation Selectivity	High bookmark-to-source ratio	Counters over-collection habits; encourages evaluating source relevance before bookmarking.
Drafting Integration Index	Low text-to-snippet ratio	Addresses disconnected draft writing; prompts a review of saved notes to verify factual support.

5.4 Optimizing Design Requirements for an LLM Prototype System

The process of developing a viable, stable software system based on theoretical designs involves meeting specific operational or design requirements. The system architecture balances automated evaluation with demographic boundaries to ensure accessible data transformation.

5.4.1 Information Pipeline and Context Generation

One key requirement in designing the system is transforming individual row-level event sequence information into a structured format without losing any pedagogically relevant meaning in the context. Therefore, the system will use a three-layered pipeline to fill the semantic gap between event sequencing and contextualization:

1. **Layer: Data Ingestion:** This layer extracts active workspace text block(s) and their associated log sequence(s) from the core persistence layer.

2. **Layer: Feature Translation:** The feature translation layer calculates high-level strategy profiles by taking the raw tracking input and passing it through the system analyzer engine.

3. **Layer: Context Formatting:** This layer generates structured output prompts that contain student behavior metric(s), explicit criterion rule(s), and/or priority attribute(s) to direct the engine's attention towards identified critical strategy gaps.

5.4.2 LLM Quality constraints

To help provide a mapping of quantitative bounds to meaningful pedagogical response outputs, the engine's output formatting profiles were created to adhere to

rigorous instructional quality constraints:

- **Actionable Procedure:** Instead of generating generic “advice” (such as "write more"), the engine will generate actionable procedural steps; thus avoiding the provision of too general guidance.
- **Preservation of Cognitive Processes:** The engine will not produce explicit reference to correct answer(s); thereby preserving student ownership of the cognitive extraction processes.
- **Management of Cognitive Load:** Engine-generated output will be limited to a single targeted strategy adjustment, thus preventing overwhelming young learners [22].

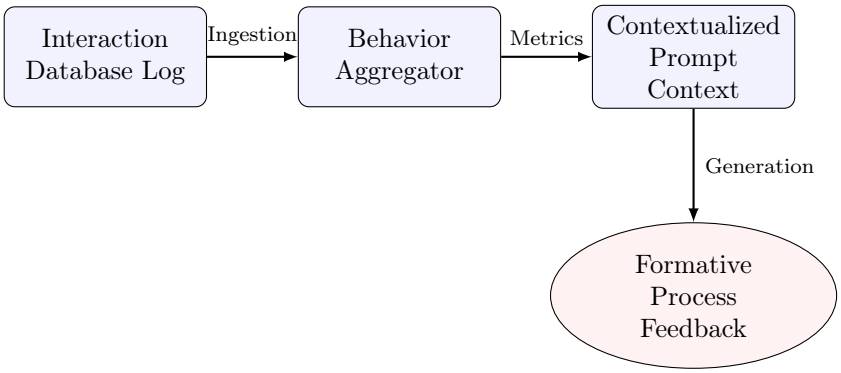

Figure 5.1: System processing pipeline for converting primitive log events into validated formative feedback.

6 Phase 2: Development

This chapter describes the technical development of the LearnNet feedback system. The focus was on building a fast, highly-modular microservices-based architecture to operate within the existing LearnNet framework, while addressing two major bottlenecks: multi-source data aggregation (logs and artifacts) and Large Language Model (LLM) inference latency.

6.1 System Architecture Overview

The overall design strategy employs an architectural model based on distributed microservices, which enables the separation of core education functions from specialized learning analytics and AI processing. Figure 6.1 illustrates these three key functional layers of the system:

- **Frontend Layer (React):** This manages the client-side UI, renders text passage content, and displays asynchronous state changes to provide real-time streaming feedback.
- **Core Backend (Node.js/Express):** This acts as the secure gateway and controls all interactions with our persistent database layers and streams log information into PostgreSQL.
- **Feedback API (FastAPI/Python):** This is a specialized microservice which

performs metrics calculation, constructs prompts, and manages LLM reasoning budgets.

- **Evaluation Infrastructure (Python/Ragas):** This is an entirely self-contained off-line pipeline used to evaluate previously recorded classroom session data, without impacting the performance of the live system.

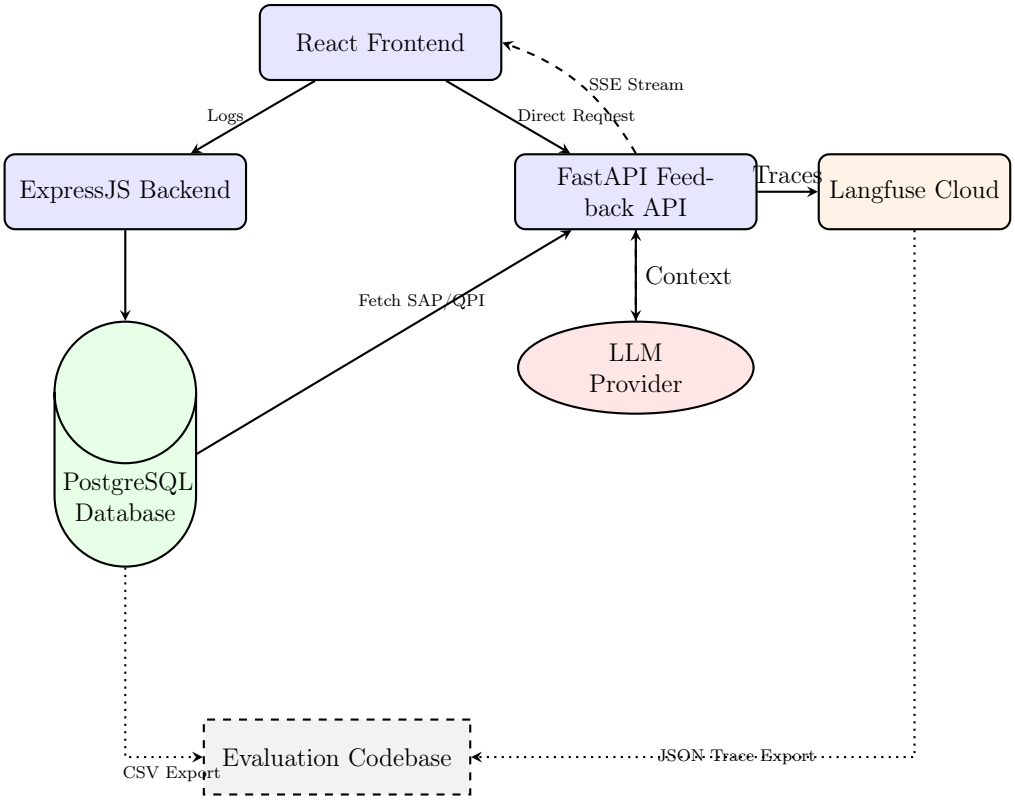

Figure 6.1: System Architecture: Vertical layout showing the direct frontend-to-API communication and the decoupled offline evaluation pipeline.

6.2 Frontend Development (React & Mantine)

The frontend system has been built with a clear goal to reduce the high cognitive load of the MSC tasks by providing an organized, tab-based environment for our students. The system was developed on top of React 18 and the Mantine library [42],

focusing on developing a fully responsive and accessible interface for lower secondary school students.

6.2.1 Modular Component Design for MSC Phases

The user interface is divided into functionally modular components; each of these modules follows the student's transition through the inquiry lifecycle according to the MD-TRACE model.

- **Search and Evaluation:** Search functionality has been implemented within the search module in `Search.jsx` and `SerpItem.jsx`. The module provides full-text searches over the entire exercise corpus. The module is responsible for the rendering of search results and the logic behind triggering bookmarking Events.

- **Snippet Extraction:** The snippet extraction module has been implemented utilizing `snippets.jsx` and `draggablesnippetcard.jsx`. Students may organize key ideas into thematic groups, which are later used as *Semantic Artifact Payloads* (SAP) for the feedback engine.

- **Synthesis Editor:** The synthesis editor has been developed utilizing rich-text framework (`editor/index.jsx`). The module allows students to draft their final answers while maintaining a sidebar view of their extracted snippets. Thus providing an opportunity for inter-textual integration.

6.2.2 The Feedback Interaction UI

One of the core challenges of developing a non-disruptive integration with AI interventions was solved by creating a persistent overlay for feedback. There are two main components in the feedback overlay:

1. **FloatingFeedbackButton.jsx:** A context-aware trigger that serves as the entry point for the "My Process Feedback" panel.

2. **FeedbackPanel.jsx:** A specialized panel that manages the display of AI-generated content. It is designed to handle the multi-part "Glow and Grow" structure and the follow-up question mechanism.

6.2.3 State management and rendering via SSE

Because of the high latency reasoning heavy LLM (discussed in Chapter 7), the front-end uses **TanStack Query** for asynchronous State management [43].

To avoid freezing the interface during "thinking" phases, the `feedbackpanel` implements a listener for **Server-Sent Events (SSE)** [44]. As the FastAPI service streams feedback tokens, the frontend uses custom utility `feedbackParser.js` to render text progressively. This implementation significantly improves *Perceived Responsiveness* of the system, allowing students to see "Glow" (positive feedback) while the model still calculates "Grow" (Improvement Suggestions).

6.2.4 Telemetry Integration

Every interaction, from text selections to tab switches, is timestamped and dispatched to the ExpressJS backend via `tracking/index.js` module. Each dispatched event receives a unique `trace_id`.

6.3 Orchestration Backend (Node.js & Express)

This Node.js service, based on the Express.js framework, is the primary API gateway for all interactions between the front-end, assignment data management, and student progress tracking. It is also responsible for securely coordinating telemetry streams.

6.3.1 Persistence Layer and Schema Design

For managing database relationships between student activities, the system uses PostgreSQL 13. Based on the implementation in the schema SQL, the database is structured to provide two types of representations of the inquiry process:

- **Identity & State:** Tables `users` and `assignment` contain information about students' identities and states, respectively. The `users` table tracks the students currently logged into the platform, while the `assignment` table provides an inventory of what task each student has been assigned.
- **Pedagogical Artifacts:** Three tables (`snippets`, `bookmarks`, and `synthesis`) manage the semantic artifact payloads. Each type of payload links the student's work to a specific text or assignment. Constraints in the SQL schema enforce this relationship.
- **Feedback Traceability:** Table `feedback` maintains a detailed log of every time the AI was called to provide feedback. Each entry includes the identifier for the particular thread being processed, the context for providing feedback, and the actual response received from the LLM.

6.3.2 Telemetry Pipeline: High Frequency Event Logging

As previously mentioned, one of the key requirements for the system (RQ1) was to obtain fine-grained behavioral data. This need is satisfied through the utilization of the `/tracking` endpoint defined in `routes/tracking.js`:

1. **Capture Data:** The front end sends JSON-encoded event objects to the backend, which include a `trackingEvent` and a corresponding `payload`.
2. **Serialize:** The backend maps these event objects onto entries within the `eventlogs` table. The schema indicates that this table records phase transi-

tions (both from and to) so that a student's traversal of the "Search", "Read", and "Synthesis" modules may be temporally reconstructed.

3. **Optimize Performance:** Since there could potentially be numerous requests from each member of an entire classroom cohort simultaneously, a pool (`db/index.js`) is used to manage connections to the database. This ensures that logging events will not block the main thread during normal operation.

6.3.3 Service Management And Security

The dual-API approach enables fast and efficient communication between services. Although FastAPI is utilized for inference by the front end, the Express backend always represents the authoritative view of the current state of each student.

- **Authentication :** Passport.js is used to perform authentication. Both local credentials (for development) and Haka (SAML/OIDC) are supported for authenticating students within Finnish education systems [45] [46].

- **Cross Origin Resource Sharing (CORS):** Due to various micro-services running behind different ports (e.g., 3000, 3001, 8000), a strict CORS policy is enforced by the backend to ensure that only valid LearnNet origins are permitted to request AI-based feedback.

6.4 Feedback Microservice (FastAPI)

The intelligence layer of the system, the fastapi-based microservice, is created to provide a highly concurrent Python environment with data transformation capabilities from student log files to LLM input requests.

6.4.1 The QPI / SAP Metric Engine

Inside the FastAPI microservice, there is a conceptually modeled metric engine. It has been implemented as an asynchronous processing layer. In order to process time-series logs from a relational table, it uses pandas.

In addition to transforming unprocessed row records into structured payloads for prompt creation, it also runs the necessary calculations for each payload. As opposed to having the data processing and aggregations occur within the main student workspace, this allows the system to avoid the potential of slowing down the application due to the large amount of processing occurring when calculating aggregated queries.

6.4.2 LLM Orchestration and Thinking Budgets

It provides an abstraction for the provider-agnostic inference layer in llm_requests.py, which will enable the ability to switch between different models such as GPT-5 mini, Gemini 3 flash, and reasoning-heavy models such as Gemini 2.0 Flash Thinking.

One new feature in this system is the "Thinking Budget Management". When performing complex synthesis tasks, the service will be allowed to create a reasoning window (an internal Chain-of-Thought) for the model before creating the student-facing response. This is done through the thinking_budget_tokens parameter, so that the model reflects on the student's QPI indicators in order to recognize certain pedagogical patterns prior to writing the final "Glow and Grow" advice.

6.4.3 Implementation of Streaming Feedback (SSE)

Because the computational intensity of reasoning-heavy models creates long response times of over 15 seconds, a positive user experience (UX) is maintained by utilizing Server-Sent Events (SSE) through the use of the StreamingResponse class in FastAPI.

As shown in streaming_utils.py, it sends back the feedback in a stream of data chunks. This allows the React frontend to start rendering both the initial praise (the "glow") and pedagogical markers as soon as they are available, thus giving an immediate visual cue to the student that their request is being processed. Due to its asynchronous nature, SSE is crucial for scaling the system for use in classrooms where potentially multiple students can submit reasoning-intensive feedback at the same time.

6.4.4 Data Validation and Schema Safety

In order to keep the system stable, all inputs and outputs are governed by Pydantic V2 schemas[47]. This implementation will prevent "model drift" or malformed LLM output from crashing the frontend. If a model produces a response that does not conform to the expected JSON structure (i.e., it misses follow-up questions), then it will capture the error, send it off to Sentry for logging, attempt a "schema correction" retry, and finally return a fallback response to the student.

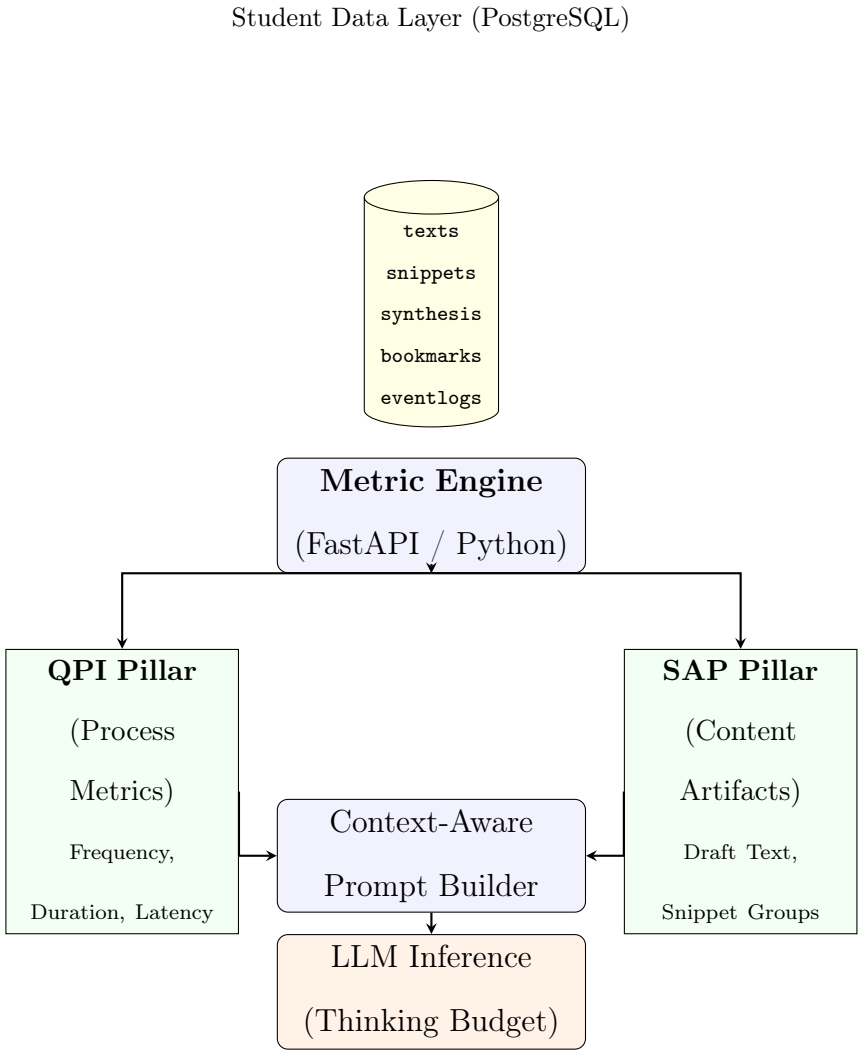

Figure 6.2: The Refined Two-Pillar Data Flow: Demonstrating how the Metric Engine orchestrates data from both the temporal log stream (`eventlogs`) and relational artifact tables to construct grounded prompts.

6.5 Observability and Telemetry

To confirm the reliability of the LLM interventions and provide the necessary foundation for the retrospective evaluation in Chapter 7, a robust and transparent ***observability*** layer was included in the Feedback API utilizing ***langfuse***. This layer allows for real-time monitoring of both the "Two-pillar" transformation process of

the data and how the model uses its internal logic.

6.5.1 Real-time Trace Capturing

With `app/llm_requests.py` being utilized to implement the system, this includes the use of the `@observe` decorator to surround all inference functions. This automatically generates a hierarchical "trace" for each feedback request, consisting of:

- **Input Context:** Raw qpi metrics and SAP artifacts provided to the model.
- **Chain-of-Thought:** Reasoning tokens produced by the model during the "thinking budget" window.
- **Metadata:** Session ids, version of model used (i.e., gpt-5 mini vs gemini-2.5-flash), etc.

6.5.2 Linking Student Activity to AI Inference

One of the key contributions of the technology design is linking a specific student activity (i.e., updating a synthesis draft) to the AI response and/or any subsequent student inquiry. The `trace_id` from the React frontend is propagated through the Express orchestration layer and then to the FastAPI service. Therefore, there exists end-to-end tracing that identifies when an exact pedagogical indicator caused a specific intervention.

6.5.3 User Feedback Loop and Error Monitoring

In addition to automatic generation of traces, the system captures explicit user feedback. The api includes an endpoint where the frontend submits user "likes" or "dislikes," accompanied by qualitative reasons why they made those selections. At

the same time, Sentry monitors runtime exceptions (for example, api rate limits, Pydantic validation errors), thereby providing high availability of the system during classroom trials [48].

6.6 Independent Evaluation and Analytics Infrastructure

To measure the pedagogical effectiveness of the real-time feedback loop and assess the technical efficiency of the real-time feedback loop, an evaluation pipeline was created independently. This pipeline serves as a post-processing engine inside an architectural decoupling of analytical infrastructure. It extracts, normalizes, and validates previously logged user traces and inference payloads without interacting with active production database threads. All steps of execution follow a modular six-stage workflow coordinated between distinct software utilities.

6.6.1 Stage 1: Multi-Source Data Alignment.

The first step joins relational logs of student activities (assignment targets, click events on student interface, etc.) with full context frames from the observability logging tier (raw prompt inputs, internal reasoning tokens produced by the model, text payloads, etc.). With this step completed, aligned data points are serialized into a uniform dataset optimized for processing and analysis.

6.6.2 Stage 2: Evaluation with Custom Rubrics.

After successful alignment of sequences, each sequence is analyzed using a fully automated semantic evaluation engine built utilizing the Ragas framework. This engine uses the LLM-as-a-judge paradigm to evaluate machine-generated responses against three core rubrics:

- **Indicator Faithfulness:** The accuracy of the generated output with respect to the traces created by students can be assessed using the taxonomy of hallucination presented by [49]. It will thus distinguish intrinsic hallucinations (the internal inconsistencies of the data of the log) and extrinsic hallucinations (non-verifiable inventions).

- **Actionability and Usefulness :** This rubric uses the model "Feed Up, Feed Back, Feed Forward" [22] of formative assessment. Feedback that is effective should go further than simply evaluating and providing support to enable the student to make progress toward a specific objective. Therefore, our rubric emphasizes the Feed Forward aspect, i.e., it provides procedural scaffolding to help to bridge the gap between what a student is currently able to do and the demands of the task.

- **Clarity and Comprehension:** This rubric was developed based on Cognitive Load Theory [30]. In order to keep cognitive load in line with the level of understanding required by school students, this rubric will therefore discourage the use of complex or dense vocabulary while encouraging the use of clear and scaffolded language.

6.6.3 Stage 3: Mixed-Methods Statistical Synthesis.

Metrics collected during the audit are automatically imported by the quantitative analytics framework. This component runs summary distributions and comparative analyses, such as t-tests and non-parametric tests, mapping automated rubric scores against actual perceptual markers of satisfaction (active likes and dislikes) by students. This synthesis identifies statistical correlations between configuration adjustments made to technical settings (i.e., changing the reasoning budget) and actual student interaction satisfaction.

6.6.4 Stage 4: Automated Behavioral Abstraction.

The abstract module categorizes sequential patterns of actions into larger functional windows (for example, source document selection, note extraction, active composition of summaries) using a 30-minute maximum session gap rule. The interpretation model processes these temporal segments to generate text-based narrative summaries, grouping student behaviors into operational profiles such as strategic synthesizer or hurried writer.

6.6.5 Stage 5: Reasoning-Effort Experimental Replays.

An experiment is designed to establish the quantitative relationship between the latency involved in generating a response to a question posed by a student and the degree to which that generated response corresponds to the correct answer. To accomplish this goal, the system will utilize a re-run engine, as part of stage-5, to re-run previously saved (archived) contexts using different levels of reasoning. This allows the system to determine what level of "reasoning" was required, within its computational model, to produce responses that were at least partially correct.

Furthermore, because all prior stages have utilized varying degrees of reasoning and logic, the system has been able to track both the total time it took to generate each response (i.e., latency), as well as the distribution of tokens among students for each response type, and how often students retrieved pre-generated answers from their local caches (context cache hits). Once the responses are generated, they are sent back to Stage-2's evaluation process, so that the effect on overall accuracy can be determined.

6.6.6 Stage 6: Final Report Generation.

Finally, the system utilizes an automated reporting tool (consolidation tool) to aggregate and combine data collected during the experimental runs and text profiling

of the student cohorts, along with multi-dimensional rubrics to create a comprehensive, dynamic HTML-based research report dashboard. This single, consolidated data artifact serves as the primary source of information used to support the empirical analyses described in Chapter 7.

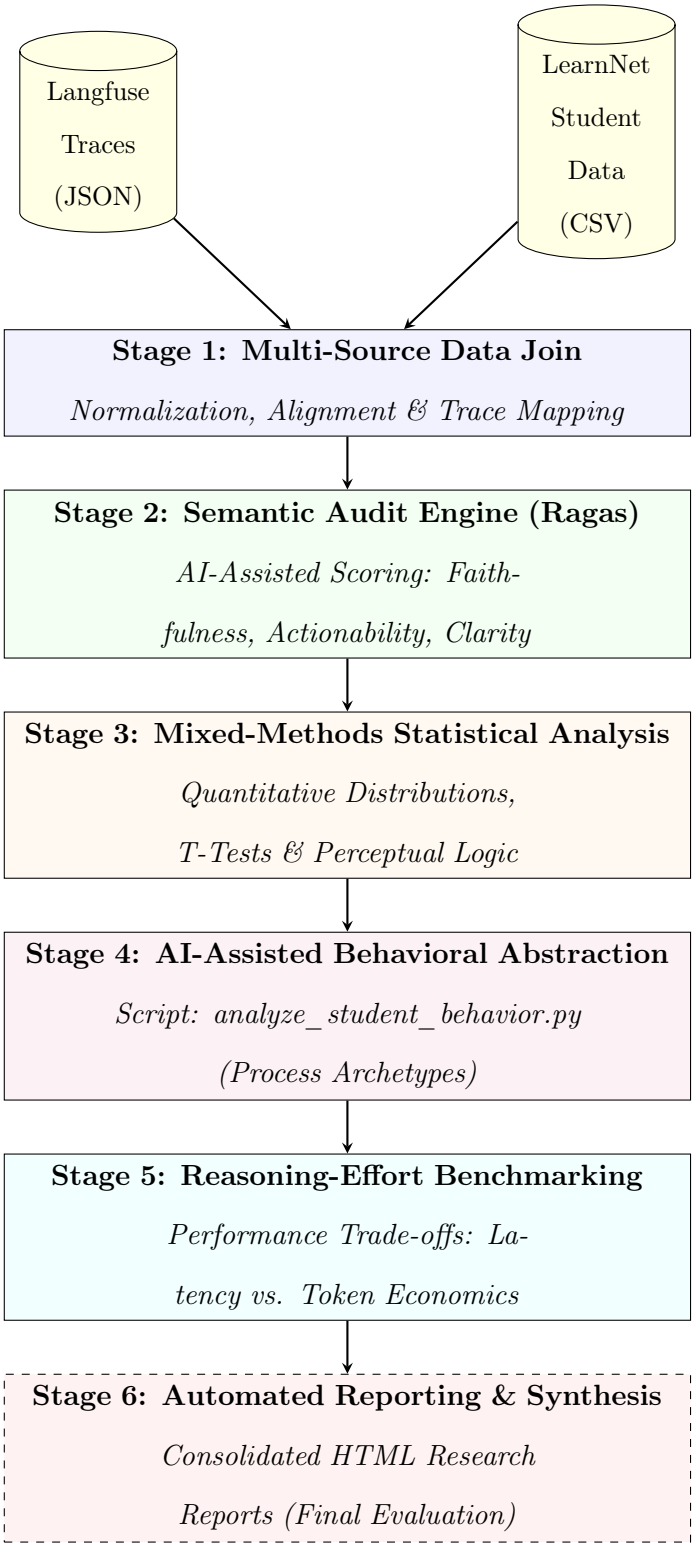

Figure 6.3: The expanded 6-stage Evaluation Infrastructure: Integrating statistical analysis with AI-driven behavioral and performance benchmarking.

7 Results And Evaluation

This chapter describes the results of the empirical study that followed the implementation of the data-driven feedback system as well as its accompanying analytical support structure. This chapter will be organized in a way to answer the research question, which were developed based on quantitative interaction data gathered during two separate sets of live classroom trials. These same trials were used to develop qualitative insights about how students interacted with their learning environment via AI-based behavioral abstraction and controlled reasoning experiments.

7.1 Evaluation Context Overview

These evaluations utilized the previously described six-stage telemetry and analysis pipeline as outlined in section 6.6. To ensure that there is a solid empirical basis for the findings, the system utilized interaction traces from two separate trials. Three key factors are measured related to these interactions, namely the amount of feedback generated by the system, the user's subjective sentiment towards the system, and the various model execution trade-offs associated with the use of each type of model.

Trial Assignment And Task Configuration

The empirical data for this study were obtained in two different deployment trial environments located within LearnNet. Those environments were EN126 and EN226.

Each of those trials consisted of 9th-grade students completing a multiple-source comprehension (MSC) task involving open-ended assignments regarding sustainable energy. While both trials required students to complete MSC tasks, there existed significant differences in terms of operational complexity and explicit cognitive demands placed upon the students across the two trials.(See Appendix B)

The baseline participation indicators and the quantitative volume of unique feedback instances generated by the system are summarized in Table 7.1.

Table 7.1: Summary of Trial Engagement and Feedback Generation

Trial ID	**Enrolled (N)**	**Students Generating Feedback**	**Total Session Events**	**Unique Feedback Rows**	**Avg. Feedback / Student**
EN126	20	20	4,671	93	4.65
EN226	20	16	4,881	99	6.19
Total	**40**	**36**	**9,552**	**192**	**5.33**

7.1.1 Validation of Modeled Student Process Indicators (RQ1)

To ensure that student interaction logs are representative of students' actual multi-Source learning processes before evaluating the quality of the student responses generated by the automated process, it was first necessary to assess whether the indicators derived from the processed logs represent actual learning strategies used by students or simply result from random system interactions. As a result, the sequential order of events for each of four different students was extracted from the overall dataset as part of this assessment.

7.1.2 Mapping Quantitative Profiles

The metrics pipeline collected and processed the raw data from the database logs into a set of quantifiable performance measures for each tracking time frame. The

four students who were selected as the basis for assessing the value of the indicator were represented by the four non-linear MSC performance characteristics shown in Table 7.2.

Table 7.2: Extracted Quantitative Process Indicator (QPI) Profiles for Representative Students

Student ID	Dur. (min)	Total Actions	Read Act.	Write Act.	Snip. Act.	FB Req.	Abstracted Behavioral Archetype
varatunnus01	34.1	198	44	92	16	5	**Strategic Synthesizer**: Systematic search-to-draft flow; high feedback responsiveness.
varatunnus03	35.9	210	42	148	0	3	**Premature Drafter**: Rapid writing with superficial source interaction; off-task browsing.
varatunnus07	48.4	214	59	67	24	2	**Linear Collector**: Heavy initial note-taking and snippet clustering; list-like text synthesis.
varatunnus18	53.8	254	44	161	14	7	**Iterative Corrector**: Highly active revision cycles; immediate removal of irrelevant material.

Note: Dur. = Duration, Act. = Actions, Read = Reading, Write = Writing, Snip. = Snippet, FB Req. = Feedback Requests.

7.1.3 Validation of High-Tier MSC Strategies

To validate whether the final states of the students' written drafts match the retrieved database indicators of each individual's interaction within the tool, a mapping process is conducted to compare the high-engagement strategies with disorganized writing workflow patterns:

1. **Structured Synthesis Profile (`varatunnus01`):** The student's log profile showed very good task orientation (4.2 min.) along with systematic source saving (12 evaluation events). The pipeline documented a large number of informational integrations (16). The structured organization of the content in the final draft, which included concept sections and clearly defined sub-headings,

validated the hypothesis that high numbers of informational integrations are a reliable indicator of a well-structured composition of text.

2. **Iterative Fact-Checking Profile (`varatunnus18`):** Student varatunnus18 utilized a significant amount of time for his initial orientation (5.0 min.), but then engaged in several rounds of fact checking using his personal draft and the open-source documentation (9 revisions). A clear pattern emerged from the event logs indicating that he continuously went back and forth between his working copy and the open-source documents to verify data prior to entering new data into the database.

7.1.4 Validation of Low-Tier and Bottlenecked Strategies

In addition to validating the high-tier strategies, the quantitative indicators also identified problematically shallow learning strategies during the execution of tasks requiring multiple sources:

1. **Bypassed Processing Profile (`varatunnus03`):** This profile maps the lower boundaries of the system metrics. The student bypassed the assignment instructions (0.5 minutes), recorded minimal source evaluation (2 events), and generated zero information integration markers (0). The transaction record indicated that the student typed phrases without support as soon as he executed general searches; therefore, low levels of each metric were effective in identifying surface-level text creation.

2. **The Integration Bottleneck Profile (`varatunnus07`):** Student varatunnus07 exhibited a type of learning bottleneck. Each of their indicator measured task engagement (3.1 min.) and source collection activity (nine evaluation events), however, there was little to no evidence of informational integration (four) or text revision (two). Although they effectively compiled data

fragments, the metric successfully identified an area of failure in the writing phase where the student simply created a list based on their notes rather than developing a synthesized structure.

With the raw click-based data now transformed into these valid indicator matrices, the system has established a reliable data base for automatically providing help when needed. To assess both the language safety and educational value of the feedback provided by each of the profiles examined here, a reference-free assessment method will be employed as discussed in the next section.

7.2 Baseline Informational Faithfulness and Semantic Quality Audit (RQ2)

This section addresses RQ2 after establishing the logging and processing pipelines for the auditing process, focusing on the generation of natural language feedback blocks for the instructional feedback format developed through live classroom deployments. In addition to technical challenges faced during deployment, one of the most significant operational challenges encountered was a noticeable decline in user sentiment toward the content presented; although students in the initial lookup trial (`EN126`), where students were introduced to the instructional feedback format, had positive perceptions of the format itself, only 60.0% of users felt that the text panel provided sufficient utility or value during the subsequent complex cross-document synthesis trial (`EN226`).

In order to identify the source of this pedagogical "bottleneck," this section analyzes the baseline performance of each of the various configurations of prompts used with active system instances. By combining quantitative algorithmic measures of text quality with qualitative telemetry data on student perceptions, the structural characteristics that determine instructional reliability and safety under different lev-

els of cognitive load are identified.

7.2.1 Baseline Field Performance: Ragas Audit and the Invariance Paradox

Table 7.3: Non-Parametric Descriptive Matrix and Mann-Whitney U Test Profiles for Baseline Ragas Metrics

	EN126 (Search, $N_{\mathbf{rows}} = 93$)	**EN226 (Synthesis, $N_{\mathbf{rows}} = 99$)**	**Mann-Whitney Test**	
Evaluation Metric	**Median & Density**	**Median & Density**	***U***	***p***
Informational Faithfulness	**4.00** L1–2: 5.4%, L3: 20.4% L4: 61.3%, L5: 12.9%	**4.00** L1–2: 6.1%, L3: 21.2% L4: 59.6%, L5: 13.1%	4568.0	0.925
Actionability / Usefulness	**3.00** L1–2: 24.7%, L3: 55.9% L4: 15.1%, L5: 4.3%	**3.00** L1–2: 23.2%, L3: 57.6% L4: 16.2%, L5: 3.0%	4493.5	0.771
Clarity / Comprehension	**4.00** L1–2: 0.0%, L3: 11.8% L4: 83.9%, L5: 4.3%	**4.00** L1–2: 1.0%, L3: 15.2% L4: 80.8%, L5: 3.0%	4337.0	0.485

Note: Significance threshold set at $\alpha = 0.05$. N_{rows} tracks individual feedback text objects replayed through the evaluation pipeline ($N_{\text{total}} = 192$). Distribution density tracks the explicit percentage allocation across qualitative rubric levels (Levels 1–5) directly beneath the median index.

The degree to which our "in-the-wild" system provided faithful information was assessed by evaluating the generated feedback data from the live system using the Ragas [17] framework. The received feedback was evaluated on its informational faithfulness, actionability, and linguistic clarity with a rating from 1.0 to 5.0. A summary of the statistical metrics based upon ranks and the U-test profiles for the deployment of the field trials is shown in Table 7.3 (`EN126`, N = 93; `EN226`, N = 99).

As indicated by the results of the U-test, there were no statistically significant differences between the two cohorts of users who used different versions of the tool

(`EN126`: Search-based; `EN226`: Synthesis-based) across the three axes ($p > .05$). Therefore, it is possible to conclude that the prompt engine produced a static quality metric regardless of the complexity of the problem being solved.

However, macro-level median scores mask an operational defect visible through the Distribution Propensity Index (DPI). A deeper look at the rubric distribution exposes an architectural bottleneck, which is illustrated in Figure 7.4.

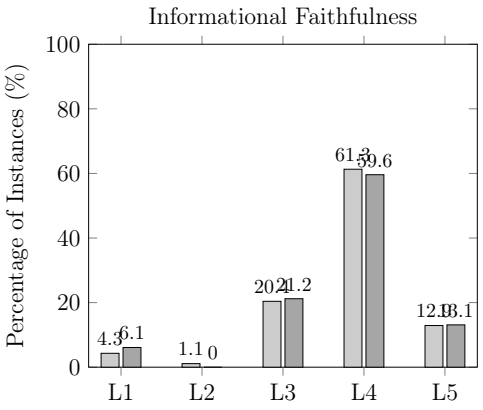

Figure 7.1: DPI for Faithfulness

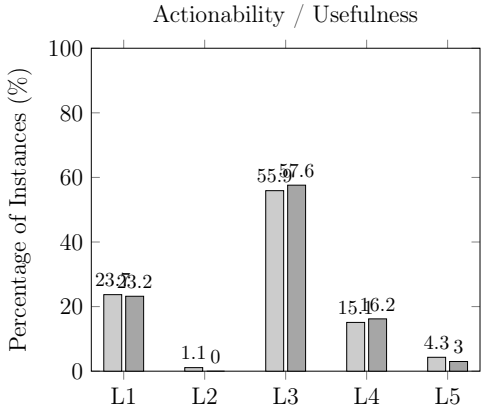

Figure 7.2: DPI for Actionability

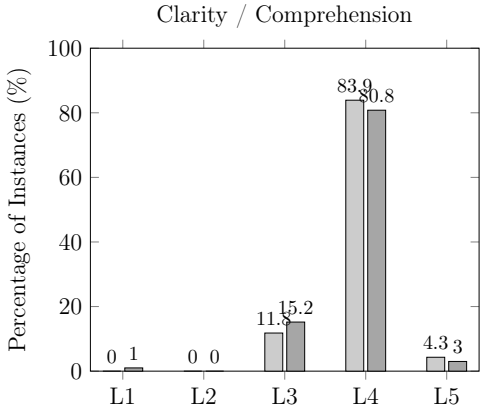

Figure 7.3: DPI for Clarity

Figure 7.4: Baseline Distribution Propensity Index (DPI) mapping across replayed database text objects ($N_{\text{rows}} = 192$). Plots illustrate the empirical density allocations for Trial EN126 (light gray, $N = 93$) and Trial EN226 (dark gray, $N = 99$) across the primary evaluation dimensions.

The cross-analysis of the density cluster indicates that there exists an architectural limit to which the baseline model can operate: the **Level 3 Trap**. In terms

of actionability, nearly 56 percent (or greater) of the total number of all generated feedback examples across both trials (`EN126`: 55.9 percent; `EN226`: 57.6 percent) fall within this particular level of the rubric. According to the evaluation framework, level three represents *spoon-feeding or overwhelming guidance*. Instead of providing accurate, focused hints that promote independent cognitive processing, the default mode of the baseline prompt generator produced lists that were excessively lengthy or even provided the exact wording of the text structures.

Additionally, the structural limitation of the baseline model further reduced informational safety by creating a clear pedagogical barrier that the system was unable to meet the expectations of students. Consequently, along the information faithfulness metric, the system frequently fell below the preferred threshold of level four (*highly grounded*) for each trial. The calculated DPI reveals that 20.4% of instances in `EN126` and 21.2% in `EN226` introduced ungrounded statements or assumptions regarding user actions (*Level 3: Minor Overreach*), thus illustrating that under heavy context loading conditions, the baseline prompt architecture has a constant tendency toward factual overreach. Therefore, as demonstrated above, under heavy context loading conditions, the baseline prompt structure will continue to incorrectly criticize users for omissions or errors that they never made while actively working with their personal workspaces.

The Automated Evaluation Blind Spot and Perceptual Limitations

A Spearman rank correlation (ρ), which measures the strength and direction of the relationship between two variables measured on at least an ordinal scale, was used to measure the degree of agreement (correlation) between the continuous Ragas text parameters (readability and actionability) and the discrete interface click log values recorded by the students.

The computation revealed no meaningful correlation for actionability ($\rho = 0.05$,

$p = 0.808$, $N_{\text{cases}} = 26$) and an absolute absence of association for readability ($\rho = 0.00$, $p = 1.000$, $N_{\text{cases}} = 24$). This statistical decoupling exposes an **Automated Evaluation Blind Spot**. The Ragas evaluator judges the feedback text by examining sentence patterns in isolation, confirming that the output is clear and grammatically fluent, as validated by the high baseline Clarity density, where over 80% of instances reach Level 4. However, it cannot model the situational task constraints of a lower secondary school student trying to manage multiple conflicting documents simultaneously.

In terms of design limitations under live workspace constraints, the basic system produced general summary statements and long lists of instructional advice. Although the machine judge determined that the sentences were structurally consistent and stable, students using the advanced writing tasks rejected the large amounts of instructional content displayed in the user interface as unhelpful (`not_understood`) or simply ignored them altogether to prevent disrupting their writing process. These findings demonstrate that a standard format for presenting prompts does not provide enough internal "processing" space to determine whether a student has taken into consideration all relevant information across multiple document sources.

7.2.2 Qualitative Case Analysis and Field Grounding

A fine-grained qualitative audit of individual database records from the active deployment explains the relationship between machine-rated fluency and human utility. By analyzing specific text generations alongside user telemetry, we can verify the boundary conditions where the automated system provides valid scaffolding or falls into abstract, unverified criticism.

Case 1: Grounded Pedagogical Scaffolding (ID: 0d5331d...) During the `EN126` informational text trial, the generation engine successfully processed high-

volume interaction logs to produce a highly grounded response. The real feedback text delivered to the student workspace was recorded as follows:

> *"Olet valinnut todella relevantteja lähteitä ja poiminut hyviä katkelmia; työskentelysi on aktiivista. Syntetisointi selittää uusiutuvat ja uusiutumattomat sekä esittelee ydin-, tuuli-, vesi- ja bioenergian sekä päästö- ja huoltovarmuusnäkökulmat. Lisää vielä konkreettinen lause: kerro, että Suomessa bioenergia usein tulee metsätalouden sivutuotteista ja mainitse vesivoiman säätörooli."*

The evaluation framework evaluated this row positively, giving it a 4.0/5.0 Indicator Faithfulness rating. This matches the highly grounded indicator faithfulness rating in the evaluation framework rubrics. All source activity claims and all operational indicator claims contained only slight changes (linguistic) to the original log entries.

For Actionability and Usefulness, the system received a 3.0/5.0 rating. When performing a structured informational lookup task, adding explicit prescriptive instructions (e.g., add forestry by-product, stabilize hydropower), provided direction to continue without overloading cognitive processing. Although such ratings may suggest that an evaluator felt too much guidance was being given based on some of the evaluation rules used, interface telemetry showed otherwise. Positive user alignment clicks (⊕ `Like`), were recorded in all task utility metrics.

Case 2: The Abstract Prompt Trap (ID: 3397107...) In contrast, during complex synthesis phases, the system encountered a diagnostic blind spot where high readability failed to translate into student utility. The feedback text presented to the student stated:

> *"Hyvä alku: kerrot energialähteiden jaottelun ja käyttötavat selkeästi. Teksti jää kuitenkin melko pintapuoliseksi — siitä puuttuu monia keskeisiä*

yksityiskohtia, esimerkkejä ja ympäristövaikutusten sekä eri lähteiden välisten yhteyksien käsittelyä. Lisää konkreettisia esimerkkejä ja yhdistä eri tekstien ideat loogiseksi kokonaisuudeksi."

This feedback scored well on clarity and comprehension with a 4.0/5.0 score by the Ragas audit, showing short, clear, and useful sentences. The student, however, rejected the interface model and logged a negative interaction event ($\ominus$ `not_understood`).

The decoupling highlights an important classification risk when using standard text metrics. As it was shown, the student's text was very fluent, but due to the system's faithfulness penalty, the student's final level was reduced from Level 1 (No Overreach) to Level 3 (Minor Overreach). Although the prompt engine did criticize the draft for failing to include "environmental impact" and "connection between sources", before providing feedback, it should have checked the current active database state, and/or checked what source references the student has been able to extract.

As another mistake, the Actionability score fell to a poor Level 2 score (Vague Advice), as instructing a lower secondary school pupil to add "concrete example(s)" or "link together their ideas in a logical way" is a generic critique, rather than providing specific step-by-step instructions for how they can progress. With no guidance being provided, the student will remain burdened with cognitive overload. Therefore, the automated advice was rejected.

Table 7.4 presents a representative sample of real evaluation records spanning high, moderate, and low student-alignment states.

This baseline ceiling illustrates how standard surface-level assessments will not be able to identify where students experience instructional drops off while engaged in high-load learning activities. This ceiling also demonstrates why the system has to have an internal reasoning step to validate active student drafts using multiple

Table 7.4: Empirical Sampling Matrix of Real Database Baseline Evaluation Logs

Feedback ID (Short)	Trial	Type	Faith.[a]	Act.[b]	Clar.[c]	Student Utility[d]	Student Clarity[e]
f0d5331d...	EN126	entire_task	4.0	3.0	4.0	⊕ (Like)	⊕ (Like)
54127d07...	EN126	synthesis	4.0	4.0	4.0	⊕ (Like)	⊕ (Like)
3e89f7ca...	EN126	synthesis	4.0	2.0	4.0	⊖ (`not_understood`)	*[Not Rated]*
c3397107...	EN126	synthesis	3.0	2.0	4.0	⊖ (`not_understood`)	*[Not Rated]*
d7667f63...	EN226	synthesis	4.0	4.0	4.0	⊖ (`not_understood`)	*[Not Rated]*
e0de8e66...	EN226	synthesis	4.0	3.0	4.0	⊖ (`not_understood`)	*[Not Rated]*
b6cc2f21...	EN226	synthesis	4.0	4.0	4.0	*[Not Rated]*	*[Not Rated]*
93880107...	EN126	snippet	4.0	2.0	4.0	*[Not Rated]*	*[Not Rated]*

[a] Informational Faithfulness rubric score (1–5 scale). [b] Actionability and Usefulness rubric score (1–5 scale). [c] Linguistic Clarity rubric score (1–5 scale). [d] Student binary interface rating for "Auttoiko tämä ymmärtämään, mitä tehdä seuraavaksi" (Next Steps). [e] Student binary interface rating for "Oliko palaute selkeää ja helppoa ymmärtää" (Readability).

reference models at the same time, to enhance its ability to act upon those drafts and eliminate excessive factual information.

7.2.3 The Limitations of the Faithfulness Audit

Although the Ragas evaluation framework presents a systematic way to compare automated feedback, the faithfulness audit is limited because it relies on reference-free metrics. Therefore, the tools that provide this assessment measure linguistic and structural faithfulness; however, they do not provide all aspects of what a teacher would consider part of their pedagogical intent or the complexities of the multi-model nature of a student's inquiry. In addition, although the study was based on one specific lower secondary school population and subject area, the study's results may differ in secondary and higher education populations or in subject areas that have more stringent factual requirements (for example, medical or law).

7.3 Experimental Reasoning Optimization and Performance Trade-offs (RQ2 & RQ3)

To address the previously described limitations of experimental evaluation and structural performance limits of live classroom deployments, an offline replay experiment

was designed. The main purpose of this decoupled experiment is to isolate the influence of generating feedback on real-time classroom network configurations by evaluating student workspace states from the trial (`EN126`), under various computational conditions that were systematically varied.

The optimization experiment uses a 2 x 3 factorial design to vary the two primary architectural factors: model generation series (`gpt-5-mini` vs. `gpt-5.4-mini`) and the internal computational cost of reasoning used in producing each output (*Minimal*, *Low*, and *Medium*). The dataset for this experiment consists of 93 feedback generation instances derived from the 20 students who participated in the EN126 classroom trial, each instance representing a distinct workspace state captured at the point when the system was triggered to generate feedback. This replay approach allows all six model-configuration combinations to be evaluated against identical historical inputs, isolating the effect of model and reasoning budget while controlling for context variability. A reference-free evaluation system was used to evaluate all generated outputs using the same rubric applied during the live trials.

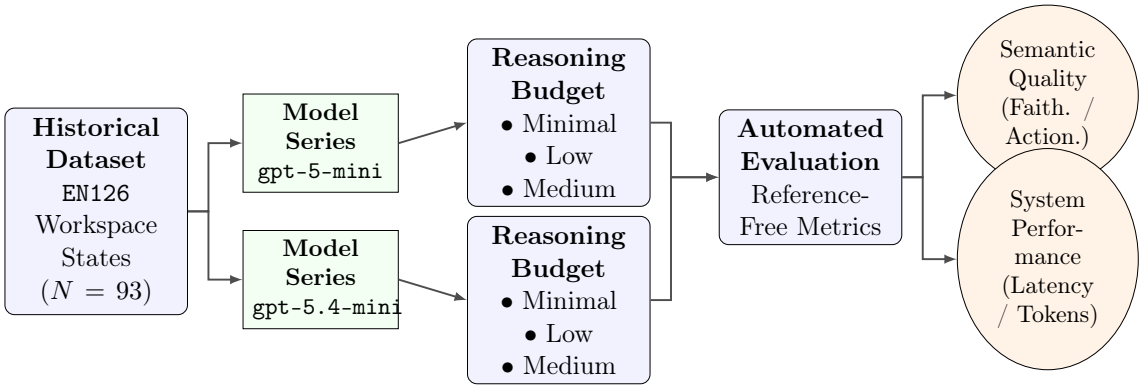

Figure 7.5: System Architecture Replay Pipeline and Factorial Experimental Workflow.

7.3.1 Aggregated Experimental Results

Tables 7.5 and 7.6 isolate the descriptive non-parametric metric distributions (Median ± Interquartile Range) alongside the non-functional system execution vectors across all six experimental states.

Table 7.5: Operational Statistics and Rubric Level Distributions for `gpt-5-mini`

Reasoning Effort	**Rubric Score Level Distribution (%)**			**Latency**	**Total Tokens**
	Level 1–2	**Level 3**	**Level 4–5**	(s, $M \pm SD$)	($M \pm SD$)
Dimension 1: Informational Faithfulness					
Minimal	1.1%	40.9%	58.0%	4.369 ± 1.563	$3,560.5 \pm 1,090.2$
Low	1.1%	23.7%	75.3%	9.115 ± 3.802	$3,502.6 \pm 1,082.1$
Medium	0.0%	25.8%	74.2%	22.376 ± 7.319	$3,499.3 \pm 1,085.7$
Dimension 2: Actionable Utility					
Minimal	24.7%	34.4%	40.9%	4.369 ± 1.563	$3,560.5 \pm 1,090.2$
Low	29.0%	41.9%	29.0%	9.115 ± 3.802	$3,502.6 \pm 1,082.1$
Medium	40.9%	29.0%	30.1%	22.376 ± 7.319	$3,499.3 \pm 1,085.7$
Dimension 3: Linguistic Clarity					
Minimal	0.0%	24.7%	75.3%	4.369 ± 1.563	$3,560.5 \pm 1,090.2$
Low	2.2%	18.3%	79.6%	9.115 ± 3.802	$3,502.6 \pm 1,082.1$
Medium	0.0%	6.5%	93.5%	22.376 ± 7.319	$3,499.3 \pm 1,085.7$

Note: Data tracking is based on $N = 93$ feedback generation instances. The baseline prompt template includes an average context envelope of $M = 432$ tokens. The input workload contained an average of 38.4 ± 10.2 individual student Data Process Indicators (DPI parameters) for the Minimal tier, rising to 42.6 ± 12.3 entries for the Medium tier configuration.

7.3.2 Semantic Quality and Instructional Safety (RQ2 Focus)

The results indicate meaningfully different scaling behaviour between the two model series as the reasoning budget increases.

Table 7.6: Operational Statistics and Rubric Level Distributions for `gpt-5.4-mini`

Reasoning Effort	**Rubric Score Level Distribution (%)**			**Latency**	**Total Tokens**
	Level 1–2	**Level 3**	**Level 4–5**	(s, $M \pm SD$)	($M \pm SD$)
Dimension 1: Informational Faithfulness					
Minimal	4.3%	37.6%	58.1%	2.700 ± 1.147	$3,481.8 \pm 1,080.1$
Low	1.1%	34.4%	64.5%	3.009 ± 0.687	$3,518.8 \pm 1,077.4$
Medium	2.2%	28.0%	69.9%	5.189 ± 1.225	$3,517.2 \pm 1,059.9$
Dimension 2: Actionable Utility					
Minimal	31.2%	21.5%	47.3%	2.700 ± 1.147	$3,481.8 \pm 1,080.1$
Low	32.3%	20.4%	47.3%	3.009 ± 0.687	$3,518.8 \pm 1,077.4$
Medium	24.7%	21.5%	53.8%	5.189 ± 1.225	$3,517.2 \pm 1,059.9$
Dimension 3: Linguistic Clarity					
Minimal	2.2%	38.7%	59.1%	2.700 ± 1.147	$3,481.8 \pm 1,080.1$
Low	2.2%	24.7%	73.1%	3.009 ± 0.687	$3,518.8 \pm 1,077.4$
Medium	1.1%	22.6%	76.3%	5.189 ± 1.225	$3,517.2 \pm 1,059.9$

Note: Data tracking is based on $N = 93$ replayed workspace contexts. The baseline prompt template includes an average context envelope of $M = 432$ tokens. The input workload contained an average of 38.4 ± 10.2 individual student Data Process Indicators (DPI parameters) for the Minimal tier, rising to 42.6 ± 12.3 entries for the Medium tier configuration.

For *gpt-5-mini*, the Medium reasoning configuration maximizes Linguistic Clarity (Med = 4.00), with 93.5% of instances reaching Levels 4–5. It also maintains solid Informational Faithfulness (Med = 4.00), with 74.2% at Levels 4–5. However, this comes at a direct cost to Actionability (Med = 3.00): 40.9% of responses fall into the lower bounds (Levels 1–2), and only 30.1% reach the upper tier. This is the core of the Performance Inversion Trap, the model exhausts its reasoning capacity on safety validation and formatting constraints, leaving insufficient token budget for generating specific, process-oriented suggestions.

The pattern worsens as reasoning effort decreases. Under Minimal reasoning, Actionability remains at Med = 3.00, with 34.4% of responses locked at Level 3 and a further 24.7% in the lower-bound risk region. Faithfulness also degrades at this setting relative to Low effort, with Level 3 instances rising from 23.7% to 40.9%, suggesting the model produces more hedged, less grounded output when the reasoning budget is severely constrained.

The *gpt-5.4-mini* series resolves this inversion. Under Medium reasoning effort, Actionability improves substantially (Med = 4.00), with 53.8% of responses reaching Levels 4–5, up from 30.1% for *gpt-5-mini* under equivalent conditions, and the lower-bound risk falling from 40.9% to 24.7%. Faithfulness holds at a safe level (69.9% at Levels 4–5), and Clarity remains strong at 76.3%. Critically, this is achieved at a mean latency of 5.189 ± 1.225 seconds, a 76.8% reduction from the 22.376 ± 7.319 seconds recorded for *gpt-5-mini* at Medium effort.

7.3.3 System Performance and Live Deployment Trade-offs (RQ3 Focus)

Scaling internal reasoning budget influence on semantic feature does introduce a system boundary, which is an important factor for constraining the system performance discussed in RQ3. Execution metric shows that there exists a strong latency

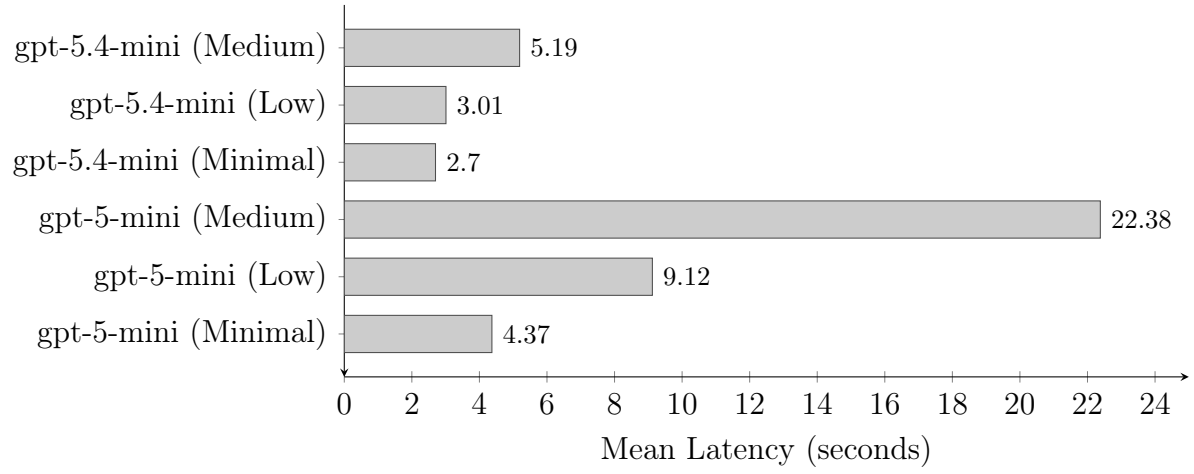

Figure 7.6: Factorial performance scaling comparison highlighting non-linear computational latencies between engine generations ($N = 93$).

tradeoff when generating models.

GPT-5 Mini engine has a large, nonlinear relationship with scaling Latency. Model generation takes a very short time using a minimal configuration (M=4.369 ± 1.563 s), but scales significantly faster than expected when scaled to the default Medium reasoning effort, with a significant increase in round-trip execution time to 22.376 ± 7.319 s. A 22-second feedback loop in an interactive classroom setting creates a clear break in real-time system usability requirements and creates potential risks for student engagement loops and timeouts at the application level.

GPT-5.4 Mini Engine successfully circumvents the performance limitation by optimizing the inference mechanism to decouple the number of reasoning steps from the delay associated with those steps. GPT-5.4 Mini Engine runs both its Minimal Tier in 2.700±1.147 seconds and the optimal Medium configuration in 5.189±1.225 seconds. More importantly, Total Token Consumption measured at the User Interface remains consistent across all Experimental Configurations ($M \approx 3500$). Therefore, these results demonstrate that the increased system latency observed in the experiments ($4.369s \rightarrow$22.376 s) was due to an increase in computational time spent executing the model's multiple-step validation checks, recursive attention mappings,

and token-level reasoning loops before delivery of the final Data Payload.

7.3.4 Architectural Synthesis and Deployment Recommendations

The empirical findings from this replay experiment provide a clear architectural layout for scaling data-driven feedback loops in open-ended multiple-source environments. The original classroom configuration (`gpt-5-mini` at Medium effort) reached an active performance ceiling. It maintained good clarity but was limited by a 22.38-second processing delay and lower actionability scores.

The data strongly suggests that the optimized solution for breaking through the synthesis bottleneck is the implementation of the `gpt-5.4-mini` model series under a *Medium* reasoning effort budget. This specific configuration concurrently scales Informational Faithfulness and Actionability while safely preserving a fast classroom response speed (5.189 s), delivering the necessary scaffolding depth within viable operational constraints.

7.4 User Adaptation, Scaffold Triggers, and Literacy (RQ4)

Analyzing student responses directly at the criterion level resolves analytical ambiguities caused by overlapping tags (e.g., cases where a student values the structural readability of a text snippet but rejects its immediate strategic utility). Table 7.7 details the verified distribution of these target telemetry traces.

The findings from this analysis illustrate some key pedagogical implications. System evaluation metrics in the highly exploratory `EN126` trial were very similar, and very high, on both structural clarity (85.0%) and utility (85.7%). However, as we move into the complex synthesis task of `EN226`, the results reveal a significant disso-

Table 7.7: Granular Student Feedback Sentiment by Evaluation Criterion

Trial ID	Feedback Evaluation Criterion	Likes	Dislikes	Total Rated	Approval Rate
EN126	Actionable Utility (Next Steps)	18	3	21	85.7%
	Linguistic Clarity (Readability)	17	3	20	85.0%
EN226	Actionable Utility (Next Steps)	3	2	5	60.0%
	Linguistic Clarity (Readability)	4	0	4	100.0%
Combined	Actionable Utility (Next Steps)	21	5	26	80.8%
	Linguistic Clarity (Readability)	21	3	24	87.5%

nance: although all of the respondents to the survey indicated that the synthesized text was clearly written at a linguistic level (100.0%), they rated the usefulness of the feedback much lower than before (60.0%, 3 likes, and 2 dislikes).

The findings from this analysis point to some early pedagogical patterns, though the small volume of active ratings—particularly in EN226-limits how far these can be generalized. In trial EN126, student-initiated evaluation rates were relatively high, with approval rates of 85.7% for actionability and 85.0% for clarity across 21 and 20 rated interactions, respectively. However, in the more complex synthesis task of EN226, active evaluation dropped sharply: only 5 students rated the feedback for utility, and 4 for clarity. Among those who did engage, there was a notable dissonance; linguistic clarity was rated at 100.0%, but actionable utility fell to 60.0% (3 likes, 2 dislikes). Given the very low interaction volume in EN226, these figures should be treated as directional indicators rather than stable estimates.

More importantly, the students' decreased ratings of utility are associated with significantly less engagement overall. The system created and stored 99 automated feedback messages (N_{rows}) throughout the synthesis trial, but the students provided only 5 responses (N_{cases}) when evaluating how useful the feedback was. The dramatic

decrease in data points collected represents a systemic failure in terms of behavioral response; instead of having to take the time to evaluate or dispute the system's assessment, nearly all students simply ignored the feedback section so that they would be able to continue writing without interruption.

Students who do choose to engage with the system's feedback are supported by the objective data, illustrating that their self-assessment of its utility is supported by their actions taken in relation to their writing. For example, varatunnus18 actively utilized feedback in their writing. Following a trigger from the system that identified an unrelated reference as being unhelpful for completing an assignment, they immediately deleted the text related to the unhelpful reference. Thus, there is evidence of real-time editing behavior. Conversely, varatunnus03 demonstrated complete non-use of the system; despite being issued three consecutive warnings regarding a lack of fact-based support in his writing, they made no subsequent changes based on those warnings.

These two examples demonstrate that in situations where contextual complexity is high, dissatisfaction and low use rates do not relate to clarity of communication of the system but rather to instructional utility. The default to general summaries or instruction lists repeatedly by the baseline prompt engine, a trend illustrated by the previously discussed semantic quality audit trap, demonstrated an inability to provide an actionable path for completing an assignment. Students such as varatunnus18 were capable of utilizing specific warning prompts to inform their writing actions. Students such as varatunnus03 were unable to make progress due to the instructional bottleneck presented by the system's repeated instructions, which resulted in a decision to utilize the system as little as possible. Consequently, the data indicate that there was a clear decline in the number of evaluations made by the students as the task complexity and contextual workload increased.

7.4.1 Distribution and Sentiment of Interface Scaffolding Modalities

The LearnNet interface enables four distinct instructional modalities through student actions while in their "active" workspace: *Synthesis*, which is deep reference integration support; *Snippet*, which identifies specific workspace sources to check for source extraction errors; *Bookmark*, which will provide users with alerts about how relevant a given source is to the task at hand; and *Entire Task*, which provides macro views of students' work toward completing an entire task. Table 7.8 presents data regarding the distribution of these triggers across both field trials as well as Liked/Disliked user preferences.

Table 7.8: Distribution and Multi-Dimensional Student Sentiment across Interface Modalities

Modality	**Evaluation**	**EN126 (Search, $N = 93$)**			**EN226 (Synthesis, $N = 99$)**		
	Criterion	Total (n)	Likes ($\oplus$)	Dislikes ($\ominus$)	Total (n)	Likes ($\oplus$)	Dislikes ($\ominus$)
Synthesis	Actionable Utility	66	8	2	50	1	2
	Linguistic Clarity		9	1		3	0
Snippet	Actionable Utility	15	5	1	17	0	0
	Linguistic Clarity		4	1		0	0
Bookmark	Actionable Utility	9	4	0	20	1	0
	Linguistic Clarity		3	1		1	0
Entire Task	Actionable Utility	3	1	0	12	1	0
	Linguistic Clarity		1	0		0	0
Total Logs		**93**	**35**	**6**	**99**	**7**	**2**

Note: Total volume logs (n) map onto the system trigger profiles. The individual item metrics track the distribution of explicit interaction markers logged for Actionable Utility (*Next Steps*) and Linguistic Clarity (*Readability*) independently.

The empirical distribution shown in Table 7.8 illustrates very clearly how different modalities of scaffolding trigger. In terms of the advanced task (EN226), the distribution becomes more diverse. For example, bookmarks, which are used to alert users to content relevant to their current focus on the assignment, had relevance alerts double from 10.00% to 20.00% ($n = 20$). The number of entire-task level triggers also increased four times over from 3.00% to 12.00% ($n = 12$).

The user interaction trace shows that the number of student evaluation entries for synthesis feedback is significantly higher than other types of feedback from the student to the teacher. Synthesis feedback has been used the most overall in each trial. However, when looking at specific tool preferences during the early data-gathering phases of EN126, the snippet ($5 \oplus$ utility, $4 \oplus$ clarity) and bookmark ($4 \oplus$ utility, $3 \oplus$ clarity) modalities achieved the highest relative approval density. This localized preference suggests that while students interacted most frequently with deep reference support, they found automated context tracking highly intuitive when performing lookup activities, a process likely made smoother by the cohort's strong domain-specific pre-knowledge on energy topics.

In contrast, when evaluating performance in both modalities across all modalities during the advanced EN226 trial, a sharp decline in the total number of active telemetry events recorded (a total of 7 cumulative likes and 2 dislikes). Since the experimental session time limits were adequate enough for students to complete each of the assigned essays, this decline indicates either a motivational barrier or a decision by students to strategically optimize their efforts.

7.4.2 Conversational Navigation and Follow-up Sub-Question Selection

To foster feedback literacy, the interface generated three targeted, conversational follow-up questions allowing students to request tailored help. When a student

clicked an interface suggestion, the session expanded into a *Follow-up Scope*. Table 7.9 isolates the structural pathways chosen by students across both trials, illustrating how users navigated between initial prompts (*Normal Scope*) and specialized follow-up options (*Clarity*, *Concept*, and *Action* sub-goals).

Table 7.9: Student Follow-up Sub-question Selection and Sentiment Analysis

Interaction Scope &	**EN126 Trial**			**EN226 Trial**		
Pedagogical Goal	Rows (n)	Share (%)[a]	Like (%)[b]	Rows (n)	Share (%)[a]	Like (%)[b]
Normal Scope	66	—	73.68	78	—	80.00
Follow-up Scope	27	—	83.33	21	—	50.00
Clarity	5	18.52	100.00	4	19.05	0.00
Concept	12	44.44	100.00	9	42.86	0.00
Action	10	37.04	50.00	8	38.10	50.00

Note: Sentiment percentages represent rows that received active evaluation marks from users within that specific conversational branch. [a] represents the specific sub-goal's proportion of total follow-up rows generated within that trial. [b] Represents the percentage of positive satisfaction marks (`True`) recorded across active evaluations.

Beginning with the conversational structures outlined above, we can clearly see how students were defining local information requirements based upon their immediate social environment. Both trials resulted in very similar distribution patterns of follow-up requests. Students continued to prioritize knowledge gaps at a very high level; they selected concept linkages greater than 42.0% of the time ($n = 12$, in EN126; $n = 9$, in EN226) and used the directive "action" option as a secondary choice approximately 37.5% of the time. Only occasionally did students seek assistance from basic vocabulary help through Clarity pathways (approximately 19.0%).

In addition, the trial results also showed that conversational follow-up loops resulted in an exceptionally high student satisfaction rating (83.33% like rate among all the rows evaluated); both Clarity and Concept paths were rated as 100.0%.

Conversely, students exhibited lower satisfaction ratings (50.0%) when complet-

ing a complex task (EN226). The decrease was due to a significant decline in the usefulness of the directive pathway within the Action option (50.0% satisfied). These findings indicate that while students continue to choose action-oriented sub-question prompts to serve as a guide for their next steps; however, the baseline prompt engine has difficulty generating specific guidance when confronted with heavy contextual load. The findings provide additional evidence to confirm that students have a high preference for conversational scaffolding, but ultimately, it will be dependent upon the degree of the system's reasoning budget available for use, which is directly related to the offline reasoning enhancements described in Section 7.3.

7.4.3 Analysis of Student Perception across Interface Modalities

To evaluate how the students perceived the instructional utility of these distinct interface interventions, post-task survey results were triangulated alongside background transaction traces from the platform database. Raw survey scores were reverse-coded using the linear transformation:

$$X_{\text{re-coded}} = 6 - X_{\text{raw}} \tag{7.1}$$

Through this transformation, a value of 5 represents maximum structural alignment (Strongly Agree), while a value of 1 represents absolute disagreement (Strongly Disagree). Table 7.10 maps these scale-corrected descriptive ratings across both field trial assignments (`EN126` and `EN226`) for the participant cohorts.

Table 7.10: Student Post-Task Perceptions of LearnNet Interface Interventions

	Perceptual Evaluation Metric	`EN126` ($N = 18$)		`EN226` ($N = 19$)	
	(Survey Item Statement)	Mean (M)	Median (Med)	Mean (M)	Median (Med)
Item 1	Feedback was easy to understand	4.22	4.00	3.58	4.00
Item 2	Feedback was generally helpful	4.44	5.00	4.16	4.00
Item 3	Feedback seemed reliable	4.11	4.00	4.32	4.00
Item 4	Helped spot deficiencies in draft	4.06	4.00	4.21	4.00
Item 5	Frequently modified text based on AI	4.18	4.00	3.63	4.00
Item 6	Accepted AI feedback unreflectively[a]	3.39	3.00	3.58	4.00
Item 7	The amount of feedback was appropriate	4.00	4.00	4.05	4.00
Item 8	Confirmed I was on the right track	3.94	4.00	4.11	4.00

Note: Descriptive scores track scale-corrected values where 5 = Strongly Agree and 1 = Strongly Disagree. [a] Item 6 functions as an inverted control check; high values indicate unreflective dependency behavior.

An important behavioral pattern that directly addresses **RQ4** is the scale-corrected tracking data demonstrating how students positively responded to the clarity, trustworthiness, and usefulness of the system (all $M \geq 3.58$) throughout both environments (EN126 & EN226).

Moreover, the substantial average value ($M_{\text{EN126}} = 4.44$, $Med = 5.00$) reported by students for Item 2 in the initial environment suggests that real-time contextual feedback has significant instructional worth when used to support beginning learners' instructional tasks.

In addition, the results of the analysis of the interaction parameter indicated an important self-regulatory strategy profile. Students were often making modifications to their drafts based on the scaffolding provided by the system ($M_{\text{EN126}} = 4.18$;

$M_{\text{EN226}} = 3.63$); however, there was no indication of dependence as related to automation of these strategies. Instead, the students showed evidence of active parsing/evaluating of the recommendations provided by the system. The median values for Item 6 ($Med_{\text{EN126}} = 3.00$; $Med_{\text{EN226}} = 4.00$) provide further support for this conclusion.

Taken together, these results suggest a potentially beneficial learning configuration in this pilot context: students appeared to use the interface to identify gaps in their source coverage and to confirm their writing direction ($M_{\text{EN226}} = 4.11$), while maintaining control over their own content decisions. The post-task perceptual data and system logs together point to iterative revision behaviors, such as removing unhelpful source materials and moving between snippets and drafts, that are consistent with autonomous self-regulation. This pattern is encouraging, though the cohort sizes involved (N = 18 and N = 19 for post-task survey; far fewer for live telemetry ratings) mean these observations should be regarded as preliminary. Larger-scale, multi-cohort trials are necessary before stronger claims about the system's effect on student autonomy can be made.

8 Discussion

This chapter will discuss all the results obtained in the empirical study presented in this thesis, and will also include information about both the technical artefacts developed and the experiences collected during the trials. The purpose here is to synthesize the results and assess the quality of the system, and to analyze the users' perception of the real-time feedback mechanism proposed to support the MSC skill through the LearnNet platform. Section 8.1 will provide the evaluation of the core research questions concerning (i) telemetry distillation; (ii) trade-off in terms of the reasoning model's performance; (iii) users' experience. Then, section 8.2 will present theoretical and practical implications. Finally, section 8.3 will report on some limitations and future works, and section 8.4 will conclude the chapter.

8.1 Discussion of Key Findings and Synthesis

8.1.1 Modeling Complex Skill Frameworks from Raw Telemetry (RQ1)

Research Question 1 aimed at determining whether low-level, high-volume log data could be converted into stable, pedagogically relevant process indicators. To date, log analysis in open-ended environments has mainly been limited to post-hoc analyses that do not allow timely intervention in response to students' behavior [4].

The event-based engine designed in this work resolves this problem by transform-

ing raw browser telemetry into behavioral ratios. More precisely, it monitors transitions between the snippet list and the synthesis word processor and measures the temporal writing-to-reading ratio to establish dynamic surrogates for self-regulated text synthesis. Contrary to considering absolute time-on-task as a direct measure of cognitive abilities, a simplification that was refuted by Goldhammer et al. [50], this allows for a better understanding of how strategic sequence patterns emerge from raw event logs.

Therefore, this conversion of raw event logs into Quantitative Process Indicators (QPIs), which corresponds to the work of Wang et al. [4], emphasizes that open-ended environments collect fine-grained, time series data while product-oriented evaluations fail to account for these aspects. The use of operational thresholds in the LearnNet Metric Engine to compute exploratory sequences is similar to what Goldhammer et al. found regarding the nonlinear relation between task execution time and student comprehension.

For example, if there is a long dwell time after receiving an assignment prompt and then a first search string is executed, it may indicate goal-directed reading preparation [51]; conversely, if a transition occurs immediately from reading sources to editing words in a processor without any intermediate annotations or manipulations of snippets, it indicates that a copy/paste behavior occurred. Therefore, the processing layer that translates raw user actions into structured representations closes the semantic gap described in current learning analytics literature [8], where raw data records are mapped to higher-level educational strategies.

8.1.2 Algorithmic Trade-offs and the Performance Inversion Trap (RQ2 & RQ3)

The intersection of Research Questions 2 and 3 highlights a major contribution of this thesis: mapping the non-linear relationship between an LLM's internal reasoning

budget, its output pedagogical quality, and system execution latency. The majority of existing research has treated foundation models as homogeneous evaluators and assumed that increasing the number of reasoning steps will always lead to better-quality feedback [11]. However, the experimental results obtained from the full baseline dataset ($N = 192$) feedback instances across both classroom trials in Table 7.3, and from the ($N = 93$) workspace-state replays drawn from the EN126 cohort in Table 7.5, show that there exists no such direct relationship, but instead present a structural weakness known as the *Performance Inversion Trap*.

Research Question 2 examined the limits of precision of the quality of automatic feedback and its faithfulness, and whether an increase in an LLM's cognitive budget can create structural distortion with respect to user interaction history [17]. While researchers have previously addressed concerns about maintaining absolute information security in educational AI systems [17], the risk of creating ungrounded statements is still very high. Jia et al. showed that LLM-generated feedback has been shown to be inaccurate and contains unfaithful comments not backed by students' actual work. This type of factual inaccuracy will reduce the trustworthiness of the feedback and thus its usability for assessing student performance.

Because the 1.0—5.0 rubric scores are used as ordinal rather than interval scales, the differences between discrete levels are based on qualitative pedagogical behaviors showing a monotonically increasing trend of text quality rather than equal quantitative increases. Thus, using averages may result in a statistical illusion of complete homogeneity. As a means of providing sufficient description while not exceeding measurement constraints, this study uses a **Distribution Propensity Index (DPI)**, a descriptive density tool defined in Section 4.5 to map the percentage of feedback instances falling within operationally defined upper and lower rubric zones, exposing the distributional structure that the rank-based tests alone cannot resolve.

As shown in the baseline field trial data in Table 7.3, the production system

running with a standard pipeline for prompts without any reasoning capabilities exhibited statistically equivalent markers for different cohorts completing tasks. In terms of information collection, with regard to cohort *EN126* ($N = 93$), the automated judge had indicator fidelity scores that were statistically equal to 4.00, actionability scores of 3.00, and clarity scores of 4.00. With regards to increased task load with respect to the advanced causal writing requirements of cohort *EN226* ($N = 99$), performance boundaries continued to be stable with regard to faithfulness, actionability, and clarity scores, all having medians of 4.00, 3.00, and 4.00. Using a two-tailed Mann-Whitney U-test ranking analysis revealed that there was absolutely no evidence for statistically significant variance with respect to faithfulness ($p = 0.925$), actionability ($p = 0.771$), and clarity ($p = 0.485$).

Although many would interpret these findings as evidence for the system being stable according to traditional statistical methods, cross-analysis of DPI density clusters reveals structural pedagogical bottlenecks that lie beneath the apparent flat medians. A center-of-gravity for actionability at exactly 3.00 is far from satisfactory. By design within the evaluation module (`metrics.py`), level three on the system rubric represents *Spoon-Feeding or Overwhelming Guidance*-an undesirable situation where the engine either provides a student with the answer to a question directly, thus removing target process-oriented cognitive effort from the student, or overwhelms a student with an excessive amount of guidance that causes instantaneous cognitive overload [22].

The data presented in this section on how the baseline DPI was broken down demonstrates that over 55% of all generated instances, in both cohorts (*EN126*: 55.9%, *EN226*: 57.6%), were trapped in a very narrow band of this extremely high Level 3 threshold. This clear output ceiling explains the severe alignment conflict observed between the automated text evaluations and the live user evaluation. Although the Ragas metrics demonstrated that the feedback text ranked consistently

across both domains, students' approvals of the actionability of the feedback fell from 85.7% in *EN126*, to just 60.0% in *EN226*, and users logged many examples where they stated that their response was *not_ understood*.

However, it's also important to understand that student evaluations were significantly lower for the advanced task (a total of 9 user evaluations in EN226 vs. 41 in EN126). This decline highlights how task differences affect automated scaffolding [7]. Although there were no changes in the learner profiles, background texts, or the classroom context for both trials, there were significant differences in the trial tasks. For instance, trial *EN126*, which required students to collect facts about a given topic and write a synthesis by organizing the facts. The trial *EN226*, however, asked students to analyze the collected information and form complex arguments.

This means that the effectiveness of the feedback provided is largely dependent upon internal student-related factors such as prior conceptual understanding, evidence-based reasoning, and intentional (active) self-regulation processes through actions such as revising their writing and re-searching relevant material [50]. Due to its inability to consider these various complex variables, students experiencing high levels of cognitive load could have chosen to ignore the evaluation system altogether and focused solely on their writing [30].

The fact that there exists a statistical paradox supports the notion that the baseline prompt engine did not adapt to the higher cognitive load requirements of synthesizing multiple documents. It produced the same overly rigid and too simplistic instructional templates, which, although acceptable for simple lookup tasks, would create barriers to developing strategic processes for complex tasks. The propensity to be faithful to indicators, as measured by DPI, also showed that the system often operated below Level 4 (*Highly Grounded*) for indicator faithfulness. An examination of the DPI distribution clearly identifies a statistically occurring trend in errors across environments; specifically, 20.4% of the instances created

unsupported assumptions concerning the student's metacognitive intent under heavy context loads [17].

A systematic assessment of whether increasing internal reasoning processing capabilities could remove the existing baseline ceiling that has been addressed through the controlled offline replay experiments listed in the Table 7.5. The findings of those experiments indicate that for earlier generation architectures (`gpt-5-mini`), extending the size of the thinking token pool causes a significant performance inversion. At the Minimal reasoning level, the model produces a consistent position (Med = 3.00) for Actionability; however, it reduces instructional safety by increasing its distribution variability, with 24.7% of responses falling into the lower-bound risk region (Levels 1–2) and a further 34.4% locked at Level 3.

At maximum reasoning effort (Medium), the model maximizes Linguistic Clarity (93.5% at Levels 4–5) and maintains solid Informational Faithfulness (74.2% at Levels 4–5), yet loses instructional Actionability entirely. Specifically, 40.9% of responses fall into the lower bounds (Levels 1–2) and only 30.1% reach the upper tier, a direct consequence of the model exhausting its reasoning capacity on safety validation and formatting constraints rather than generating specific, process-oriented suggestions.

This architectural constraint is largely resolved by the newer generation architecture (`gpt-5.4-mini`). As shown in the Table 7.6, the new model configurations exhibit a more stable scaling profile across all three dimensions simultaneously. As the reasoning budget expands from Minimal to Medium, Actionability upper-tier density increases from 47.3% to 53.8%, while the lower-bound risk decreases from 31.2% to 24.7%. The model also reaches Med = 4.00 for Actionability at Medium effort, a level the `gpt-5-mini` series fails to reach under any configuration. Faithfulness upper-tier density rises from 58.1% at Minimal to 69.9% at Medium, while Clarity improves from 59.1% to 76.3%. Therefore, this systematic improvement in-

dicates that we have transitioned from producing overly rigid or vague instructional templates towards truly process-oriented scaffolds without introducing contradictions or unsupported instructional claims [10]. Additionally, this improvement allows us to utilize independent validation frameworks or high-parameter automated gatekeeping solutions, such as the multi-agent filtering models proposed by Qian et al. (2025), to block unfaithful feedback iterations before deployment.

In addition, this generational shift effectively circumvents the execution latency penalty issues (referenced in RQ3) associated with deep-reasoning educational systems. In a live classroom setting, responsiveness is paramount; execution times greater than 10-15 seconds will disrupt a student's mental flow and decrease their reliance upon digital tools [23]. For example, our `gpt-5-mini` model operating at Medium effort resulted in an unworkable execution delay of 22.376 ± 7.319. Under equivalent environmental settings, our `gpt-5.4-mini` model achieved completion of its ideal Medium reasoning cycle in a stable time frame of 5.189 ± 1.225. This represented a 76.8% reduction in operation latency and thus establishes that optimization of architectural routing can de-link complex pedagogical reasoning from interactive classroom constraints.

8.1.3 Interface Interventions, Trust, and Learning Friction (RQ4)

Research Question 4 analyzed the lower secondary cohort (grade 9) interaction with the real-time UI intervention. It evaluated how the feedback loops generated by automatic UI influence students' feedback literacy and long-term self-regulation. The results from the psychometric scales in Table 7.7 ($M \geq 3.58$ across both cohorts) suggest that, within this pilot setting, younger students showed positive levels of evaluative trust toward the system. Survey responses indicate that students generally perceived the feedback as valid and useful for identifying structural problems

in their essays. Given that these conclusions rest on cohort sizes of N = 18 and N = 19, respectively, they should be interpreted as preliminary rather than definitive evidence of broader student trust in AI-generated educational feedback.

These results are consistent with the results of many studies that compared human evaluation to machine-generated feedback. For example, Sessler et al. (2025) found that conversational language models could simulate human teacher input and provide positive, clear, and contextually relevant terms to middle school cohorts [23]. However, researchers also noted that "out of the box" prompting systems frequently create a diagnostic gap and fail to provide sufficient strategic information to help young students navigate a process of productive struggle toward an answer without immediately showing them what they should do [35].

As significant, while there was a high level of initial trust in the system, this trust did not lead to an over-reliance on automation or a lack of reflection. One of the key concerns that has been raised regarding the use of AI-based writing and programming tools is the potential for a copy-and-paste trap. A copy-paste trap occurs when students take the suggestions made by machines in order to reduce their current immediate cognitive load [34]. The LearnNet UI avoided this type of situation by including an inverted control scale item (item 6, table 7.7) labeled "accepted AI feedback unreflectively", which had a median of approximately 3.0 to 4.0. This suggests that the UI structure introduced **productive learning friction**.

Because the system highlighted *process errors* (for example, suggesting a student go back and read again because their source coverage was too shallow), but did not directly correct content or suggest specific text blocks for students to write, students were required to reflect on and implement the feedback themselves. In essence, the AI served as a mirror to their learning strategy, requiring students to perform the actual revisions to their work.

This structural dynamic can be easily seen through the use of qualitative case

studies taken from the EDR cycle. The study on `varatunnus18` provided clear evidence of a productive misalignment: although `varatunnus18` had returned very low subjective survey scores for both overall satisfaction and stylistic phrasing of the AI's critiques and was strongly in disagreement with the AI, they still showed enormous objective gains in both the breadth and depth of their ability to synthesize and integrate the sources. For `varatunnus18`, the system acted as expected: it was able to act as an independent validation mechanism, which verified the Student's critique of the AI. Through the process of explaining why they did not agree with the AI, `varatunnus18` was forced into greater levels of reflection about themselves, and therefore developed a written justification to explain their choices.

On the other hand, the defensive friction shown in `varatunnus03` (who rejected all baseline prompts from the AI that were deemed to be "un-reasoned" or flawed) illustrates the importance of having explanation tracing when using formative feedback systems. When students are given generic, flat directives from an automated agent that contradict their personal assessment of what they have done, they immediately respond emotionally to resist and reject those suggestions. Therefore, formative feedback has to be based upon explicit explanations for how the feedback is related to specific traces of the student's past behavior if it is going to be seen as credible and meaningful.

8.2 Implications for Theory and Practice

8.2.1 Theoretical Implications

This study contributes to the learning sciences by developing a valid method for modeling collaborative human-AI agency in open-ended environments. The multimedia learning and multiple-source comprehension frameworks, including the MD-trace model [21], have generally viewed information problem solving as a cognitive

process that is internal, multi-step, and regulated by self-regulatory standards. This study demonstrates how real-time log analysis may provide access to these hidden cognitive processes and alter them via systematic, goal-directed scaffolding. This structure-based relationship supports the theoretical trends identified in Barzilai et al. (2023) [2], who noted that explicit digital triggers and descriptive scaffolds are used to support younger students developing coherent cross-document mental models.

In addition, this thesis extends our knowledge of the concept of student feedback literacy in the context of generative AI. Rather than defining feedback literacy as the ability to interpret human-generated comments passively, it defines it as an active competency in critiquing, filtering, and adapting to non-deterministic automated cues without becoming either algorithmically compliant or defensively avoidant.

8.2.2 Practical Implications

For system designers, software engineers, and education administrators, this research provides several clearly articulated design principles for implementing large-scale automated feedback systems:

1. **Audit for Performance Inversions:** Developers designing generative systems must continually audit their systems to monitor for performance inversions. Developers cannot assume that increasing the number of parameters or the logical depth of their models will lead to higher levels of educational value.

2. **Prioritize Process over Product Scaffolding:** To preserve student autonomy and to encourage reflective use of AI rather than unreflective reliance on AI, the interfaces designed for students must limit the generation of content edits generated by automated agents. Rather than having AI automatically generate content edits, AI should prompt students to go back to their original

learning action (i.e., review the credibility of their sources; expand upon their notes).

3. **Embed Explainable Telemetry within the Payload:** To reduce defensive resistance (as illustrated in the `varatunnus03` profile), the automated prompts provided to students must include direct references or quotes to the quantitative process metric(s) that were utilized to trigger the intervention (i.e., "you have allocated 80% of your time reviewing one source..."). Thus, instead of receiving generic claims made by automated agents, students receive data-grounded insights that enable them to understand what led to the need for an intervention.

8.3 Limitations and Directions for Future Work

Although this research has conducted comprehensive engineering validations and well-documented behavioral patterns over multiple rounds of iterative design, there are also various constraints that limit the generalization of its findings. The first limitation is related to the sample sizes of both the offline reasoning budget experiments that were based on a high-quality historical data set ($N = 93$), and the live psychometric survey, which was limited by school-based sampling methods ($N < 20$ per cohort). Due to the small sample size in the survey, the use of parametric statistical tests to determine whether changes in student learning behaviors could be attributed to the use of AI tutoring was inappropriate because of a lack of sufficient statistical power. Therefore, large-scale, multi-school longitudinal studies will need to be implemented to statistically differentiate the impact of AI tutoring from traditional classroom instruction with respect to how students regulate their own learning behavior.

The second limitation relates to the structural assumptions associated with the

telemetry processing layer. The layer assumes consistent, linear browser sessions. However, if students are experiencing high levels of variability in their session usage patterns such that they may have multiple tabs open at one time, or when working together on shared workspaces, then the structural indicators (i.e., writing-to-reading temporal ratios) may be altered by the amount of idle activity occurring in background processing.

Therefore, future iterations of this project should include some type of multi-agent validation framework that would allow an independent review agent (e.g., similar to what has been proposed by Qian et al. [10] as DeanLLM) to evaluate the generated feedback payload against real-time telemetry validation before being displayed within the client user interface. Additionally, expanding the prompt pipeline to provide real-time cross-language translations for users representing different languages provides a critical path forward toward developing universal, accessible AI tutoring systems for all global education systems.

A third limitation concerns the scope of evaluation itself. This study assesses the quality and perceived usefulness of the AI-generated feedback, but does not measure whether use of the system leads to measurable improvement in students' MSC skills. Without a pre/post-assessment of student learning outcomes or a control group completing the same tasks without the feedback system, it is not possible to establish whether the scaffolding provided translates into actual skill development. This is a recognized boundary of the EDR methodology at this stage of the design cycle, where the priority is establishing system functionality and formative validity rather than conducting a summative learning effectiveness trial. Longitudinal studies incorporating standardized MSC assessments would be a natural and necessary next step.

9 Conclusion

9.1 Research Summary and Contributions

The primary focus of this thesis was to develop a real-time, data-driven feedback system in the LearnNet open-ended learning environment to support lower secondary students in completing their multi-document literacy tasks. To accomplish this, a customized event-processing engine was developed. Using the engine, the raw system log files were transformed into quantifiable process indicators. These process indicators enabled the system to make its AI-generated prompts based on objective measures of student behavior, instead of simply based on final product scores or simplistic time-on-task measures.

This study resulted in three new contributions to the fields of Learning Analytics and Educational Technology through four rounds of iterative Educational Design Research (EDR):

1. **Real-Time Telemetry Pipeline**: An engineering solution that allows for real-time processing of student activity (for example, reading to writing ratios, snippet saves), without negatively impacting overall system performance.

2. **Empirical Mapping of LLM Behavior**: Benchmark data for understanding how model size, internal reasoning patterns, and response constraint requirements impact the quantity and quality of feedback generated automatically by LLMs.

3. **Preliminary Instructional Evidence**: Pilot data from two real-world classroom trials suggesting that process-focused, non-directive automated hinting is perceived positively by young learners and does not appear to encourage over-reliance on AI tools. Measuring its direct impact on MSC skill development remains an open empirical question for future work.

9.2 Key Technical and Pedagogical Lessons

Both technical and pedagogical implications arose during the development and evaluation of the LearnNet feedback architecture. Technologically, the off-line controlled studies identified an important engineering limitation referred to as the *Performance Inversion Trap*. The trap is described as follows: when you increase the internal reasoning parameters of an earlier generation LLM (e.g., gpt-5-mini), it spends all of its remaining tokens on internal safety checks and formatting constraints, resulting in very little actionable value in its responses. In addition to poor actionable value, high latency also arises from the trap. Both limitations were completely overcome with the use of the newer version, gpt-5.4-mini, whose internal reasoning parameters scaled smoothly, which reduced run time latency by 76.8%; and provided true process-focused hints versus previously vague and overwhelming direction.

Pedagogically, the findings from the system studies indicated that productive learning friction needs to be introduced when using LLMs in education. Specifically, by disabling writing/editing functionality within the automated agent so that students remained active participants in the learning cycle, the developers ensured students would produce thoughtful, original revisions. Additionally, by enabling students to trust the analysis capabilities of the automated agent but requiring students to engage in their own revisionary processes, the system prevented students from falling prey to common "copy-paste" pitfalls associated with many AI writing assistants currently available today. The result was a technology designed to

support authentic skill acquisition and self-regulated learning, one that, within this pilot context, students engaged with reflectively rather than passively.

9.3 Concluding Remarks

In conclusion, this thesis presents early-stage evidence that large language models can function as a complementary tool for classroom instruction when properly aligned with instructional research principles and validated through real-time data. The pilot findings suggest that, when designed with appropriate constraints, these tools have the potential to assist teachers in providing scalable and contextualized scaffolding for inquiry-based education. As digital classrooms continue to evolve, the design framework and empirical lessons from this study contribute a foundation for future work on automated systems that aim to preserve student autonomy and support young learners as they navigate complex information environments.

References

[1] W. Kintsch, *Comprehension: A paradigm for cognition*. Cambridge university press, 1998.

[2] S. Barzilai, D. Tal-Savir, F. Abed, S. Mor-Hagani, and A. R. Zohar, "Mapping multiple documents: From constructing multiple document models to argumentative writing", *Reading and Writing*, vol. 36, pp. 809–847, 2023. DOI: 10.1007/s11145-021-10208-8.

[3] M. Heikkilä, M. Häkkinen, A. Vidbäck, M. Mikkilä-Erdmann, and I. E. Sääksjärvi, "Student teachers' conceptual understanding of biodiversity loss in a multiple-source learning environment", *Environmental Education Research*, vol. 31, no. 7, pp. 1356–1370, 2025. DOI: 10.1080/13504622.2024.2436110. eprint: https://doi.org/10.1080/13504622.2024.2436110. [Online]. Available: https://doi.org/10.1080/13504622.2024.2436110.

[4] K. D. Wang, J. M. Cock, T. Käser, and E. Bumbacher, "A systematic review of empirical studies using log data from open-ended learning environments to measure science and engineering practices", *British Journal of Educational Technology*, vol. 54, no. 1, pp. 192–221, DOI: 10.1111/bjet.13289. eprint: https://bera-journals.onlinelibrary.wiley.com/doi/pdf/10.1111/bjet.13289. [Online]. Available: https://bera-journals.onlinelibrary.wiley.com/doi/abs/10.1111/bjet.13289.

[5] M. Pedaste et al., "Phases of inquiry-based learning: Definitions and the inquiry cycle", *Educational Research Review*, vol. 14, pp. 47–61, 2015, ISSN: 1747-938X. DOI: 10.1016/j.edurev.2015.02.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1747938X15000068.

[6] C. A. Perfetti, J.-F. Rouet, and M. A. Britt, "Toward a theory of documents representation", *The construction of mental representations during reading*, vol. 88108, 1999.

[7] J. Van de Pol, M. Volman, and J. Beishuizen, "Scaffolding in teacher–student interaction: A decade of research", *Educational psychology review*, vol. 22, no. 3, pp. 271–296, 2010.

[8] N. Erdmann and K. Rautio, "Creating process indicators from log data to foster complex skills", in *Programme and Abstracts of the Finland-Germany International Conference on Artificial Intelligence in Education (FLAIEC 2024)*, Conference Abstract, Joensuu, Finland, 2024, pp. 18–19.

[9] A. Vaswani et al., "Attention is all you need", in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010, ISBN: 9781510860964.

[10] K. Qian et al., *Dean of llm tutors: Exploring comprehensive and automated evaluation of llm-generated educational feedback via llm feedback evaluators*, 2025. arXiv: 2508.05952 [cs.CY]. [Online]. Available: https://arxiv.org/abs/2508.05952.

[11] H. Seo et al., "Large language models as evaluators in education: Verification of feedback consistency and accuracy", *Applied Sciences*, vol. 15, no. 2, p. 671, 2025.

[12] *Gpt-4o system card*, OpenAI, 2024.

[13] E. Mazzullo and O. Bulut, "Automated feedback generation for open-ended questions: Insights from fine-tuned llms", in *Proceedings of Large Foundation Models for Educational Assessment*, S. Li, Z. Cui, J. Lu, D. Harris, and S. Jing, Eds., ser. Proceedings of Machine Learning Research, vol. 264, PMLR, 15–16 Dec 2025, pp. 103–120. [Online]. Available: https://proceedings.mlr.press/v264/mazzullo25a.html.

[14] R. S. Srinivasa et al., *Tutorbench: A benchmark to assess tutoring capabilities of large language models*, 2025. arXiv: 2510.02663 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2510.02663.

[15] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models", in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, New Orleans, LA, USA: Curran Associates Inc., 2022, ISBN: 9781713871088.

[16] J. Li, W. Zhao, Y. Zhang, and C. Gan, *Steering llm thinking with budget guidance*, 2025. arXiv: 2506.13752 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2506.13752.

[17] Q. Jia et al., "On assessing the faithfulness of llm-generated feedback on student assignments", in *Proceedings of the 17th International Conference on Educational Data Mining (EDM)*, 2024, pp. 491–499.

[18] L. Zheng et al., "Judging llm-as-a-judge with mt-bench and chatbot arena", in *Advances in Neural Information Processing Systems*, 2023. [Online]. Available: https://arxiv.org/abs/2306.05685.

[19] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval augmented generation", in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, N. Aletras and O. De Clercq, Eds., St. Ju-

lians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 150–158. DOI: 10.18653/v1/2024.eacl-demo.16. [Online]. Available: https://aclanthology.org/2024.eacl-demo.16/.

[20] U. Lee et al., *Rewarding how models think pedagogically: Integrating pedagogical reasoning and thinking rewards for llms in education*, 2026. arXiv: 2601.14560 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2601.14560.

[21] J.-F. Rouet and M. A. Britt, "Relevance processes in multiple document comprehension", in *Text relevance and learning from text*, Information Age Publishing, 2011, pp. 19–52.

[22] J. Hattie and H. Timperley, "The power of feedback", *Review of Educational Research*, vol. 77, no. 1, pp. 81–112, 2007.

[23] K. Seßler, A. Bewersdorff, C. Nerdel, and E. Kasneci, *Towards adaptive feedback with ai: Comparing the feedback quality of llms and teachers on experimentation protocols*, 2025. arXiv: 2502.12842 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2502.12842.

[24] E. Panadero and A. A. Lipnevich, "A review of feedback models and diverse types of feedback in higher education", *Educational Research Review*, vol. 35, p. 100 416, 2022.

[25] Y. Long, H. Luo, and Y. Zhang, "Evaluating large language models in analysing classroom dialogue", *npj Science of Learning*, vol. 9, no. 1, p. 60, 2024. DOI: 10.1038/s41539-024-00273-3.

[26] S. McKenney and T. C. Reeves, *Conducting Educational Design Research*, 2nd. Routledge, 2012.

[27] L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press, 1978, ISBN: 9780674576292.

[28] J. Frerejean, J. L. H. van Strien, P. A. Kirschner, and S. Brand-Gruwel, "Embedded instruction to learn information problem solving", *Computers in Human Behavior*, vol. 90, pp. 117–130, 2019. DOI: 10.1016/j.chb.2018.08.037.

[29] G. Hughes, H. Smith, and B. Creese, "Not seeing the wood for the trees: Developing a feedback analysis tool to explore feed forward in modularised programmes", *Assessment & Evaluation in Higher Education*, vol. 40, no. 8, pp. 1079–1094, 2015. DOI: 10.1080/02602938.2014.969187.

[30] J. Sweller, "Cognitive load during problem solving: Effects on learning", *Cognitive Science*, vol. 12, no. 2, pp. 257–285, 1988. DOI: 10.1207/s15516709cog1202_4. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1202_4. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1202_4.

[31] J. J. Van Merrienboer and J. Sweller, "Cognitive load theory and complex learning: Recent developments and future directions", *Educational psychology review*, vol. 17, no. 2, pp. 147–177, 2005. DOI: 10.1007/s10648-005-3951-0.

[32] S. Kalyuga, P. Ayres, P. Chandler, and J. Sweller, "The expertise reversal effect.", *Educational Psychologist*, 2003.

[33] K. K. Maurya, K. A. Srivatsa, K. Petukhova, and E. Kochmar, "Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors", 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)*, Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 1234–1251. DOI: 10.18653/v1/2025.naacl-long.57. [Online]. Available: https://aclanthology.org/2025.naacl-long.57.

[34] N. Scholz, M. H. Nguyen, A. Singla, and T. Nagashima, "Partnering with ai: A pedagogical feedback system for llm integration into programming education", in *Two Decades of TEL. From Lessons Learnt to Challenges Ahead*. Springer Nature Switzerland, 2025, pp. 243–248, ISBN: 9783032038739. DOI: 10.1007/978-3-032-03873-9_31. [Online]. Available: http://dx.doi.org/10.1007/978-3-032-03873-9_31.

[35] M. Kapur, "Productive failure", *Cognition and Instruction*, vol. 26, no. 3, pp. 379–424, 2008. DOI: 10.1080/07370000802212669.

[36] H. Bastani, O. Bastani, A. Sungu, H. Ge, Ö. Kabakcı, and R. Mariman, "Generative ai without guardrails can harm learning: Evidence from high school mathematics", *Proceedings of the National Academy of Sciences*, vol. 122, no. 26, e2422633122, 2025. DOI: 10.1073/pnas.2422633122. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2422633122. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2422633122.

[37] D. Agostini and F. Picasso, *Evaluation of large language models' educational feedback in higher education: Potential, limitations and implications for educational practice*, 2026. arXiv: 2602.02519 [cs.CY]. [Online]. Available: https://arxiv.org/abs/2602.02519.

[38] C. Adams, P. Pente, G. Lemermeyer, and G. Rockwell, "Ethical principles for artificial intelligence in k-12 education", *Computers and Education: Artificial Intelligence*, vol. 4, p. 100 131, 2023. DOI: 10.1016/j.cacai.2023.100131.

[39] C. S. Lau, M. Mikkilä-Erdmann, and N. Erdmann, "Toward partnership with AI: A mixed-methods study of feedback literacy and student engagement", in *Proceedings of the EdMedia + Innovate Learning 2025 Conference*, Turku, Finland: Association for the Advancement of Computing in Education (AACE), Dec. 2025.

[40] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other", en, *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, Mar. 1947, ISSN: 0003-4851. DOI: 10.1214/aoms/1177730491. [Online]. Available: http://projecteuclid.org/euclid.aoms/1177730491.

[41] C. Spearman, "The proof and measurement of association between two things", *The American Journal of Psychology*, vol. 15, no. 1, p. 72, Jan. 1904, ISSN: 00029556. DOI: 10.2307/1412159. [Online]. Available: https://www.jstor.org/stable/1412159?origin=crossref.

[42] Meta Platforms, Inc., *React: A javascript library for building user interfaces*, 2022. [Online]. Available: https://react.dev/blog/2022/03/29/react-v18.

[43] TanStack, *Tanstack query: Powerful asynchronous state management for ts/js*, 2023. [Online]. Available: https://tanstack.com/query/latest.

[44] W3C, *Server-sent events*, W3C Recommendation, 2015. [Online]. Available: https://www.w3.org/TR/eventsource/.

[45] CSC – Tieteen tietotekniikan keskus Oy, *Haka-käyttäjätunnistusjärjestelmä*, Eduuni-wiki, 2024. Accessed: Jun. 12, 2026. [Online]. Available: https://wiki.eduuni.fi/spaces/CSCHAKA/pages/27297845/Haka-k%C3%A4ytt%C3%A4j%C3%A4tunnistusj%C3%A4rjestelm%C3%A4.

[46] J. Hanson, *Passport: Simple, unobtrusive node.js authentication*, 2024. [Online]. Available: https://www.passportjs.org/.

[47] S. Colvin, *Pydantic: Data validation and settings management using python type hints*, 2023. [Online]. Available: https://github.com/pydantic/pydantic.

[48] Sentry Team, *Sentry: Open-source error tracking and performance monitoring*, 2024. [Online]. Available: https://sentry.io/.

[49] Y. Zhang et al., "Siren's song in the AI ocean: A survey on hallucination in large language models", *Computational Linguistics*, vol. 51, no. 4, Dec. 2025. DOI: 10.1162/coli.a.16. [Online]. Available: https://aclanthology.org/2025.cl-4.9/.

[50] F. Goldhammer, J. Naumann, A. Stelter, K. Tóth, H. Rölke, and E. Klieme, "The time on task effect in reading and problem solving is moderated by task difficulty and skill: Insights from a computer-based large-scale assessment", *Journal of Educational Psychology*, vol. 106, no. 3, pp. 608–626, 2014. DOI: 10.1037/a0034716. [Online]. Available: https://doi.org/10.1037/a0034716.

[51] M. A. Britt, T. Richter, and J.-F. Rouet, *Reading as a Strategic and Evaluative Process: Essays in Honor of Charles A. Perfetti*. New York: Routledge, 2014. DOI: 10.4324/9781315871271.

Appendix A Evaluation Instruments

This appendix presents the exact data-collection instruments and questionnaires used to gather student responses for this research.

A.1 Feedback Literacy(FL) Questionnaire

The instrument consists of seven feedback literacy items evaluated on a 5-point Likert scale ranging from 1 (Täysin eri mieltä / Strongly Disagree) to 5 (Täysin samaa mieltä / Strongly Agree).

A.1.1 Feedback Literacy (FL) Scale Items

The following items measure how students process, reflect upon, and implement instructional feedback to manage their learning process:

1. **Ques-F1:** Yleensä kuuntelen palautteen huolellisesti ja otan sen huomioon seuraavissa tehtävissä.
   (*English:* I usually listen to feedback carefully and take it into account in future tasks.)

2. **Ques-F2:** Pyydän palautetta työstäni tietyistä asioista, joissa haluan kehittyä.
   (*English:* I ask for feedback on specific aspects of my work in which I want to improve.)

3. **Ques-F3:** Yleensä pohdin palautetta tarkasti ennen kuin päätän, miten hyödynnän sitä tehtävän tekemisessä.
   (*English:* I usually reflect on feedback carefully before deciding how I will use it when completing a task.)

4. **Ques-F4:** Yleensä pohdin, miten palaute vastaa tehtävän tavoitteita ja arviointikriteerejä.
   (*English:* I usually consider how the feedback aligns with the goals and assessment criteria of the task.)

5. **Ques-F5:** Yleensä suhtaudun rauhallisesti ja avoimesti palautteeseen, vaikka se olisi kriittistä.
   (*English:* I usually respond to feedback calmly and openly, even when it is critical.)

6. **Ques-F6:** Yleensä pystyn hyödyntämään palautetta, vaikka se ensin tuntuisi epämiellyttävältä.
   (*English:* I am usually able to make use of feedback even if it initially feels uncomfortable.)

7. **Ques-F7:** Tarkistan, onko työni parantunut sen jälkeen, kun olen ottanut palautteen huomioon.
   (*English:* I check whether my work has improved after I have taken the feedback into account.)

A.2 Post-Task Survey Items

The post-task questionnaire captures user evaluations immediately after completing the energy-themed multiple-source learning assignment. The following items are evaluated using a 5-point Likert scale: 1 = Täysin eri mieltä (Strongly Disagree), 2 =

Jokseenkin eri mieltä (Somewhat Disagree), 3 = Ei samaa eikä eri mieltä (Neutral), 4 = Jokseenkin samaa mieltä (Somewhat Agree), and 5 = Täysin samaa mieltä (Strongly Agree).

1. **Post-L1:** Palaute oli pääosin hyvin ymmärrettävää.
   (*English:* The feedback was mostly understandable.)

2. **Post-L2:** Palaute oli pääosin hyödyllistä.
   (*English:* The feedback was mostly useful.)

3. **Post-L3:** Palaute vaikutti luotettavalta.
   (*English:* The feedback seemed reliable.)

4. **Post-L4:** Palaute auttoi minua huomaamaan puutteita tai epäkohtia omassa vastauksessani.
   (*English:* The feedback helped me notice deficiencies or shortcomings in my answer.)

5. **Post-L5:** Muokkasin usein vastaustani tekoälyn palautteen perusteella.
   (*English:* I often modified my answer based on the feedback from AI.)

6. **Post-L6:** Hyväksyn yleensä tekoälyn palautteen sellaisenaan ilman, että pohdin sitä enempää.
   (*English:* I generally accept the feedback from AI as it is, without thinking much further about it.)

7. **Post-L7:** Käyttämäni tekoäly palautteen määrä tuntui sopivalta tehtävän tekemiseen.
   (*English:* The amount of feedback from AI I used felt suitable for completing the task.)

8. **Post-L8:** Palaute vahvisti minulle, että olen oikealla tiellä ratkaisun löytämiseksi.

(*English:* The feedback confirmed to me that I am on the right track to find a solution.)

9. **Post-L9:** Palautteet tallennetuista sivuista olivat minulle tärkeitä.
   (*English:* Feedback regarding saved source pages was important to me.)

10. **Post-L10:** Palautteet pääkohdista olivat minulle tärkeitä.
    (*English:* Feedback regarding main points was important to me.)

11. **Post-L11:** Palautteet yhteenvedosta olivat minulle tärkeitä.
    (*English:* Feedback regarding the summary was important to me.)

12. **Post-L12:** Koko tehtävää eli prosessia koskeva palaute oli minulle tärkeintä.
    (*English:* Feedback regarding the entire task process was the most important to me.)

Appendix B Trials Tasks

B.1 Task 1: EN126

"Hei,

Olen kunnan nuorten ilmasto- ja energiapaneelin koordinaattori. Valmistelemme verkkosivuille tietopakettia, jonka tarkoituksena on auttaa nuoria ymmärtämään, mistä energiaa saadaan ja miten eri energialähteet vaikuttavat ympäristöön ja yhteiskuntaan. 9.-luokkalainen Aurora on lähettänyt paneelille kysymyksen: Mitä eri energialähteitä on olemassa?

Kirjoita selkeä ja yleistajuinen tietoteksti otsikolla "Mitä eri energialähteitä on ja miten ne eroavat toisistaan?" Teksti julkaistaan verkkosivuillamme.

Etsi kolme hyödyllistä ja luotettavaa sivua LearnNetistä ja laadi tekstisi niiden pohjalta.

Huomioi kirjoittaessasi seuraavat asiat:

- *Käytä käsitteitä, jotka sopivat nuorille suunnattuun tietosisältöön.*
- *Esittele ja luokittele löytämäsi energialähteet selkeästi.*

Ystävällisin terveisin,

Niina Laitinen

Nuorten ilmasto- ja energiapaneelin koordinaattori"

B.2 Task 2: EN226

"Hei,

Olen kunnan nuorten ilmasto- ja energiapaneelin koordinaattori. Valmistelemme verkkosivuillemme tietopakettia, jonka tarkoitus on auttaa nuoria ymmärtämään, miten energiaa voidaan tuottaa kestävästi.

9.-luokkalainen Aurora on lähettänyt paneelille kysymyksen: Miten energiaa voisi tuottaa vastuullisesti Suomessa?

Kirjoita selkeä ja pohtiva tietoessee, joka julkaistaan verkkosivuillamme.

Tekstin otsikko on: ***Miten energiaa voitaisiin tuottaa vastuullisesti Suomessa?***

Etsi neljä hyödyllistä lähdetekstiä LearnNetistä ja hyödynnä niitä pohdintasi tukena.

Huomioi kirjoittaessasi seuraavat asiat:

- *Käytä asiallisia käsitteitä, jotka sopivat nuorille suunnattuun tietosisältöön.*
- *Perustele näkemyksiäsi ja tuo esiin syy–seuraussuhteita (esimerkiksi ympäristövaikutukset, huoltovarmuus ja taloudelliset näkökulmat).*
- *Pohdi erilaisia energianlähteitä ja niiden vastuullisuutta Suomen olosuhteissa.*

Ystävällisin terveisin,

Niina Laitinen

Nuorten ilmasto- ja energiapaneelin koordinaattori"

Appendix C LLM As A Judge Evaluation Rubrics

C.1 Faithfulness and Groundedness Rubric

The faithfulness metric evaluates the degree of factual adherence to the authoritative signals provided in the student activity logs and artifact dictionaries. This metric specifically checks for intrinsic and extrinsic hallucinations in the tutor output.

Score 1: Direct Contradiction. The feedback explicitly contradicts the provided student logs or artifacts, introducing distinct intrinsic hallucinations (e.g., praising a student for reading a source that logs show was skipped).

Score 2: Major Fabrication. The feedback does not explicitly contradict the logs, but invents major actions, claims, or student achievements that are entirely unsupported by the provided baseline data.

Score 3: Minor Overreach. The core instructional message remains grounded in the source data, but the model makes unsupported assumptions regarding student intent or introduces 1–2 minor claims not validated by the data.

Score 4: Highly Grounded. The feedback is factual and aligns with the logs and student artifacts, containing only slight linguistic embellishments that stray marginally from the raw tracking metrics.

Score 5: Flawless Faithfulness. Every claim, praise, or critique maps perfectly and exclusively to the provided student logs and artifacts, introducing zero external facts or unsupported assumptions.

C.2 Actionability and Guidance Rubric

The actionability metric measures the pedagogical usefulness of the generated message, ensuring that it provides scaffolding rather than giving away answers or remaining purely critical.

Score 1: Purely Evaluative. The feedback only judges or grades the work (e.g., *"This is wrong"* or *"Good job"*) while providing zero advice on how the student should proceed.

Score 2: Vague Advice. The text suggests that an improvement is needed but provides no procedural path or strategy (e.g., *"Try to find better sources"*).

Score 3: Spoon-feeding / Overwhelming. The model provides a specific next step, but either reveals the exact correct answer (robbing the student of cognitive effort) or provides an overwhelming, unprioritized list of instructions.

Score 4: Actionable Guidance. The feedback provides a single, clear next step that requires active cognitive work from the student, pointing them in the right direction without revealing the final solution.

Score 5: Masterful Scaffolding. The model identifies the specific behavioral bottleneck and provides one prioritized strategy, framing the advice as a guided question that triggers productive struggle.

C.3 Clarity and Comprehension Rubric

The clarity metric ensures that the linguistic complexity and affective tone of the feedback are developmentally appropriate for the target middle school student persona.

Score 1: Hostile or Incomprehensible. The tone is excessively critical or relies on dense academic jargon that a primary or middle school student cannot understand.

Score 2: Developmentally Inappropriate. The meaning is recoverable, but the vocabulary is too advanced, sentences are too long, or the structure is too disorganized for a younger learner.

Score 3: Passable but Flat. The text is understandable but overly wordy or repetitive, reading like a rigid robotic system rather than a supportive tutor.

Score 4: Clear and Supportive. The vocabulary is highly accessible, the sentence structure is concise, and the tone is genuinely encouraging for the student.

Score 5: Perfect Persona. The text reads exactly like a supportive human learning coach speaking to a middle school student, utilizing highly readable sentences, excellent formatting, and confidence-building language.