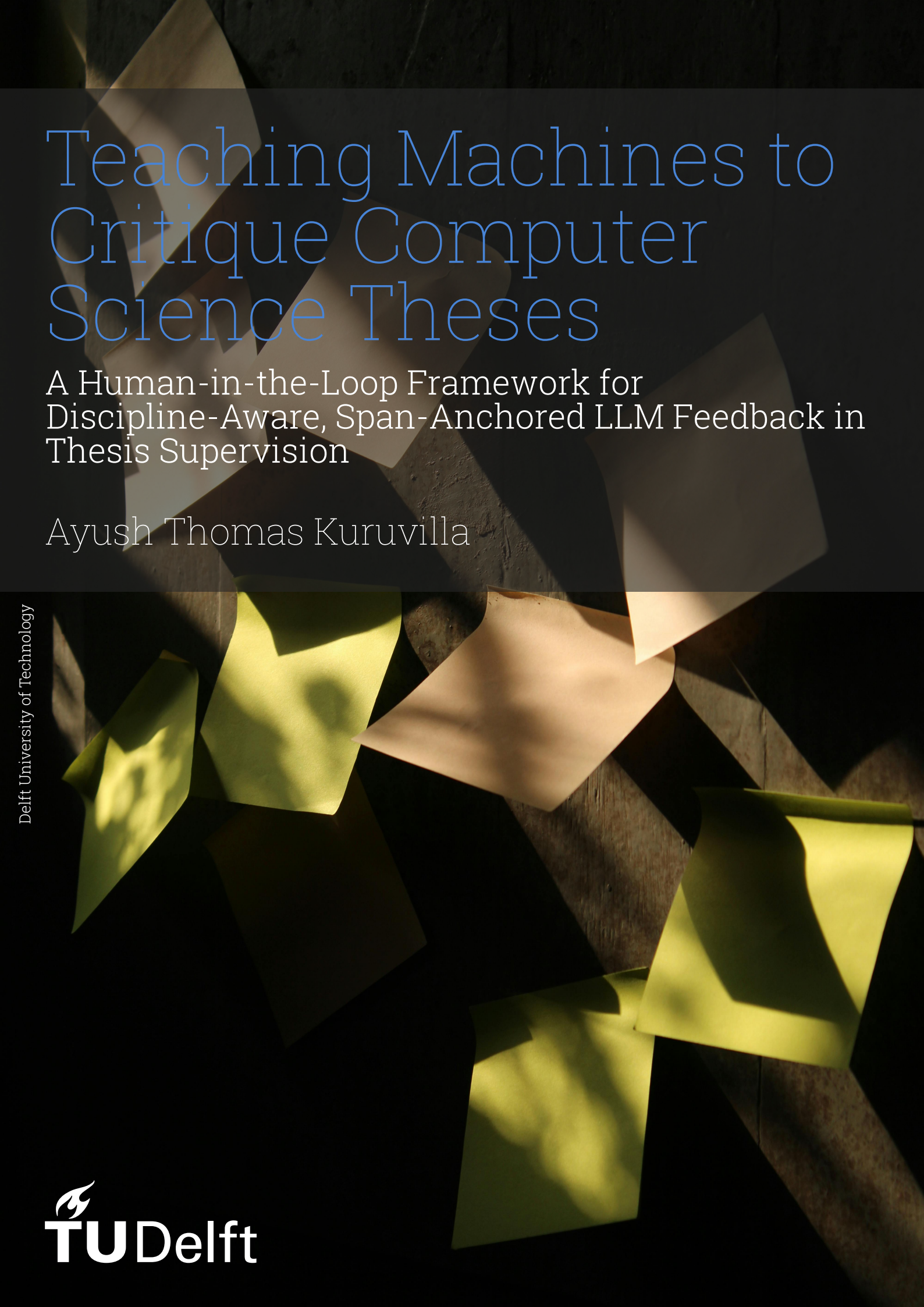

Teaching Machines to Critique Computer Science Theses

A Human-in-the-Loop Framework for Discipline-Aware, Span-Anchored LLM Feedback in Thesis Supervision

Ayush Thomas Kuruvilla

Delft University of Technology

TU Delft

Teaching Machines to Critique Computer Science Theses

by

Ayush Thomas Kuruvilla

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday, June 15, 2026 at 11:00.

Student number: 6207235
Project duration: November 10, 2025 – June 15, 2026
Thesis committee: Gosia Migut, TU Delft, supervisor
Tom Viering, TU Delft, supervisor
Stephanie Tan, TU Delft, committee member

Cover: "Sticky Notes on Wall with Natural Light and Shadows" by Jahra Tasfia Reza

TU Delft

Preface

I began this thesis hoping to find a project that I found genuinely interesting, that could contribute to a research community, and that would still let me build something concrete before returning to software development in industry, the work I most enjoy. After many conversations and discarded ideas, this project clicked. It combined a scientific question with the chance to build for a community I care about. My mother, my aunt, and many of the people I grew up around were teachers and lecturers, so I have always had a close view of the time and care that teaching requires. Supporting that work, even at the scale of a single university course, mattered to me. It is also why preserving the balance between the model side and the interface that supervisors could actually use became important.

Many people contributed to this work along the way.

I am deeply grateful to my supervisors, *Gosia Migut* and *Tom Viering*, for giving me the freedom to scope the project as a paired model-and-interface study, for their kindness and support in brainstorming and thinking through ideas with me, and for the time they invested in reading imperfect drafts. I also thank *Stephanie Tan* for kindly taking the time to be part of my thesis committee.

The two user studies in this work would not exist without the professors, PhD candidates, and postdoctoral researchers who evaluated the feedback comments and the interface. Their qualitative comments on the thesis-feedback task and the interface were among the most helpful parts of the studies.

I am grateful to *Sowmya*, who listened patiently and was a steady source of emotional support throughout this work.

I am also grateful to *Smrithi*, *Pratik*, and my nephew *Kian*, for being my family and my home away from India over these years. I am thankful for the peace, the food, and the sense of comfort I find with them.

I thank my family, without whom I would not have had the opportunity to pursue a master's at TU Delft, for their support and prayers throughout. I also thank my friends for standing alongside me during this thesis.

This thesis was supported computationally by the *DelftBlue* high-performance computing cluster, which made it possible to train and serve open-weight 27B and 70B-parameter models inside institutionally controlled infrastructure without depending on external commercial APIs for any step that touched student-derived material. That choice was important to me, and the work would have looked quite different without it.

Ayush Thomas Kuruvilla
Delft, June 2026

Contents

List of Figures

List of Tables

Glossary

Acronyms

CSE3000 TU Delft Computer Science and Engineering bachelor Research Project, the supervision context for which the thesis-feedback system was designed.

DSAIT Data Science and Artificial Intelligence Technology, the master's programme under which this thesis was carried out.

GDPR General Data Protection Regulation (EU Regulation 2016/679).

JSON JavaScript Object Notation, the structured output format used by the generation pipelines.

LLM Large Language Model.

LoRA Low-Rank Adaptation, a parameter-efficient supervised fine-tuning method.

NLP Natural Language Processing.

PDF Portable Document Format.

RAG Retrieval-Augmented Generation.

RQ Research Question.

SFT Supervised Fine-Tuning.

SUS System Usability Scale [1].

UI User Interface.

Key Terms

Span-anchored feedback A feedback comment that is attached to a specific span of text from the thesis draft, paired with surrounding section context. Used as the supervised training format and as the unit of human and automatic evaluation throughout the thesis.

Held-out evaluation pool The five annotated thesis documents reserved before any fine-tuning and reused across the LLM-as-a-judge benchmark, the blind feedback-quality study, and qualitative inspection.

Generation strategy The broader test-time configuration that determines what context the model sees and how multi-step orchestration is structured. The three strategies compared are *whole-document prompting*, *section-aware two-stage generation*, and *agentic review*.

Prompt family The set of prompt variants used in this thesis that share the same task instruction, rubric-derived priorities, exclusion rules, and structured output schema, and that differ only in how they supply context.

LLM-as-a-judge An automatic evaluation pipeline in which one LLM scores another LLM's output against an explicit rubric, used here for strategy-level comparison.

Supervisor suitability An evaluation dimension distinct from descriptive feedback quality, capturing whether a comment would be appropriate for a supervisor to send to a student as written or after only minor editing.

Notation

s Anchor span from the thesis draft.

c Surrounding section context, including section headings and neighboring text.

y Feedback comment associated with the span.

(s, c, y) One supervised training example.

p Course-guideline prompt.

Introduction 1

Writing a thesis is a central part of higher education. The student must take technical or scholarly work and turn it into a coherent written report whose argument, methodology, evidence, and conclusion can stand independent scrutiny. Supervisor feedback shapes that process: it points the student toward unsupported claims, unjustified methodological choices, or conclusions that do not follow from the evidence, and it does so across multiple levels at once, from local writing-level corrections to broader structural and argumentative critique [2, 3, 4, 5, 6, 7]. Useful feedback goes beyond surface correction. It tells the student what to change, why the change matters for the argument, and how the revision should look [3, 8, 9, 10].

Providing this feedback is time-consuming. Drafts are long, and each comment requires careful reading and a deliberate choice of what to comment on. This effort becomes more difficult to sustain when many drafts must be reviewed in a short period. The supervision context for this work is the TU Delft CSE3000 bachelor research project, which enrols close to four hundred students per year and requires supervisors to provide detailed written feedback on student drafts within a ten-week project window [11]. Supervisors carry this workload alongside their teaching, research, and departmental responsibilities. Studies of academic workloads have consistently reported that these demands compete for limited faculty time [12, 13, 14, 15]. This creates a practical motivation for automated assistance that prepares reviewable draft comments while leaving supervisory judgment intact.

Supervision context and source corpus

TU Delft CSE3000 Research Project: a 10-week bachelor research project in computer science with approximately 400 students per year under distributed supervision [11].
Source corpus for this work: 1,140 span-feedback pairs recovered from 55 supervisor-annotated thesis drafts. Five additional drafts are held out before fine-tuning and reused for both LLM-as-judge and blind human evaluation.

Large language models make automated first-pass feedback more plausible than earlier rule-based approaches. They can read long passages, recognise patterns of academic writing, and draft comments that resemble supervisor feedback [16, 17]. The same capability can also produce critique that is incorrect, too generic to guide revision, or focused on the wrong issue [18, 19]. Recent work on AI-generated writing feedback [20, 21, 22, 23, 24, 25, 26, 27] and on AI-supported scholarly review [28, 29, 30, 31, 32] reports both outcomes at once: useful local critique alongside shallow, mis-prioritised, or weakly grounded comments. A recent systematic review of AI-based automated written feedback similarly finds that current systems address surface-level issues well but deal weakly with deeper, contextual revision [33, 34, 35]. In thesis supervision, the supervisor remains responsible for the feedback that reaches the student. An incorrect comment can therefore create additional review risk, which is why this work treats the supervisor, not the model, as the decision-maker on what feedback the student receives.

This thesis studies how open-weight large language models can produce reviewable draft feedback for computer-science bachelor theses inside a supervisor-controlled workflow. The feedback is treated as *span-anchored*: every generated comment is attached to a specific passage of the thesis, so that a human reader can see the exact text the comment refers to and can judge its relevance, correctness, and specificity. Anchoring a comment to a visible referent has both an empirical and a formal motivation: writers act on comments more reliably when the comment names the problem location and a possible fix [36, 37], and the W3C Web Annotation Model formalises annotation as a body–target pair with a selector pointing into a text segment [38]. Span anchoring serves three roles in this work. As a

training signal, it matches the format of the supervisor comments in the source corpus, which are already attached to passages. As an evaluation unit, it lets raters score each comment in context against its anchor. As an interaction unit, it allows a supervisor-facing PDF interface to display generated comments as editable suggestions that the supervisor can accept, revise, or dismiss before export.

Example: a span-anchored thesis-feedback item

Anchor span: *"important"*
Surrounding context: *"In machine learning, learning curves are an **important** metric that plots performance versus training set size. They inform decisions about data acquisition, model selection, and hyperparameter tuning."*
Supervisor-style feedback: *"Subjective: everything can be called important. Instead, explain why you think this is important and what depends on it."*
The model is trained to produce one such comment per anchor, grounded in the surrounding section context. The feedback above is from the CSE3000 source corpus and illustrates the form of comment the model learns to imitate.

The work is conducted within a number of practical constraints. Thesis drafts and supervisor comments contain personal educational data, which makes external commercial APIs less suitable for both training and inference and brings the work under UNESCO guidance on generative AI in education [39], the EU AI Act's transparency and human-oversight requirements for high-risk educational uses [40], and GDPR principles of data minimisation [41]. This regime motivates the use of open-weight models that can be fine-tuned and served on institutional infrastructure rather than queried through retention-bound external APIs, in line with arguments that closed proprietary models obscure training data, model changes, and evaluation artefacts and so make scientific audit harder [42]. Theses are long documents, which means that a model may need to process tens of thousands of tokens at once; long-context studies show that simply increasing the context window does not guarantee the relevant information is used [43, 44], and recent comparisons of retrieval-augmented generation and long-context prompting indicate that they trade off depending on task structure and budget [45, 46, 47]. Good thesis feedback is also multidimensional: a comment can be relevant but incorrect, specific but unhelpful, or accurate but inappropriate to send to a student. Preset criteria cannot fully represent this kind of qualitative judgment [48], which is why the peer-feedback and peer-review literature scores feedback along several axes in parallel rather than against a single reference comment [49, 8, 50, 51].

The investigation is organised around four research questions:

RQ1: To what extent can fine-tuned span-anchored feedback match or improve on human supervisor feedback under blind human rating, across relevance, change clarity, correctness, actionability, specificity, and supervisor suitability?

RQ2: How do base and fine-tuned open-weight models compare across whole-document, section-aware two-stage, and agentic review strategies for long thesis-feedback generation?

RQ3: How usable is a supervisor-facing interface that presents generated comments as editable, span-anchored draft annotations inside a PDF review workflow?

RQ4: Which feedback-quality and workflow risks persist when LLM comments are used as reviewable draft feedback under supervisor control?

The thesis is organised in two parts. Chapter 2 contains the scientific paper, which presents the problem, the methodology, the evaluation, and the main results. Chapter 3 provides supplemental materials: a more thorough literature review, background on the underlying concepts, the dataset construction pipeline, implementation details for the models and the supervisor-facing interface, the pilot evaluation that shaped the deployed configuration, the user-study methodology including the human-research-ethics procedure, additional analyses of the LLM-as-a-judge benchmark, and a discussion of how the work relates to the surrounding research field. The appendices preserve the study instruments, the full prompt templates, the CSE3000 source guidelines, the interface screenshots, and the additional judge tables.

Scientific Article 2

Teaching Machines to Critique Computer Science Theses: Span-Anchored Feedback with Open-Weight Models and Supervisor-in-the-Loop Review

Ayush Thomas Kuruvilla
Delft University of Technology

Abstract

To produce reviewable draft feedback on long computer-science thesis drafts under supervisor control, we present a controlled three-way comparison of LLM generation strategies on open-weight backbones, paired with a supervisor-facing PDF review interface. We compare base and LoRA-fine-tuned Llama 3.3 70B and Qwen 3.5 27B across whole-document, section-aware two-stage, and agentic review strategies, evaluated through an LLM-as-judge benchmark over all twelve configurations, a blind human rating study, and an interface usability study. Whole-document generation obtains the highest judge-macro score. Fine-tuning helps in only one of six matched base-versus-fine-tuned comparisons, indicating that supervised adaptation on a small span-anchored corpus shifts response distribution without expanding critique ability. In the blind study, the deployed section-aware generation is rated significantly higher than held-out supervisor feedback on change clarity ($p = .027$) and specificity ($p = .018$), while correctness shows no significant difference ($p = .516$). On the comments scored by both, three independent LLM judges rate the system above the human raters on every dimension (mean bias 0.41 points), most strongly on supervisor suitability. The PDF review interface reaches a mean System Usability Scale (SUS) of 77.5. The results support using open-weight LLMs to produce reviewable draft feedback on long technical theses under supervisor inspection, editing, and export control.

1 Introduction

Writing a thesis requires students to produce a long, self-directed piece of technical writing that develops and supports a research contribution. It is where students learn to frame a research problem, defend methodological choices, and connect evidence to claims, and supervisor feedback is a central channel through which that argument is shaped [1, 2, 3]. Written supervisor feedback on drafts ranges from local writing-level comments on wording and clarity to broader critique of organization, argument, and subject-matter coverage [3, 2].

We study this problem in the setting of a large university thesis course, where detailed feedback must be provided repeatedly and under time pressure. The Delft University of Technology (TU Delft) CSE3000 Research Project is a 10-week bachelor course in computer science with nearly 400 students per academic year under distributed supervision [4]. Each supervisor is expected to give detailed written feedback on student drafts over the course of the project, in addition to teaching, research, and service responsibilities that already constrain faculty time [5]. Under that load, automation that prepares a reviewable first pass over a long draft could help supervisors focus on higher-level structural and methodological critique, while leaving them responsible for selection and revision. The goal is not to replace supervisors, but to produce reviewable draft comments that a supervisor can inspect, edit, and approve before any feedback is returned to students.

Prior work points to three properties that make automating thesis feedback difficult. First, feedback quality is multidimensional: useful comments differ not only in what they address, but also in how localized, specific, justified, accurate, actionable, and helpful they are [6, 7, 8, 9, 10]. A comment can therefore be specific but wrong, accurate but unhelpful, or relevant but too vague to act on. This makes single-reference evaluation a weak proxy for usefulness [11]: a held-out supervisor comment can serve as a reference example of useful feedback, but not as the only valid answer. Different reviewers can reasonably prioritize different problems in the same passage.

Second, modern large language models (LLMs) can draft plausible comments over arbitrary text, yet recent work on AI-generated writing feedback

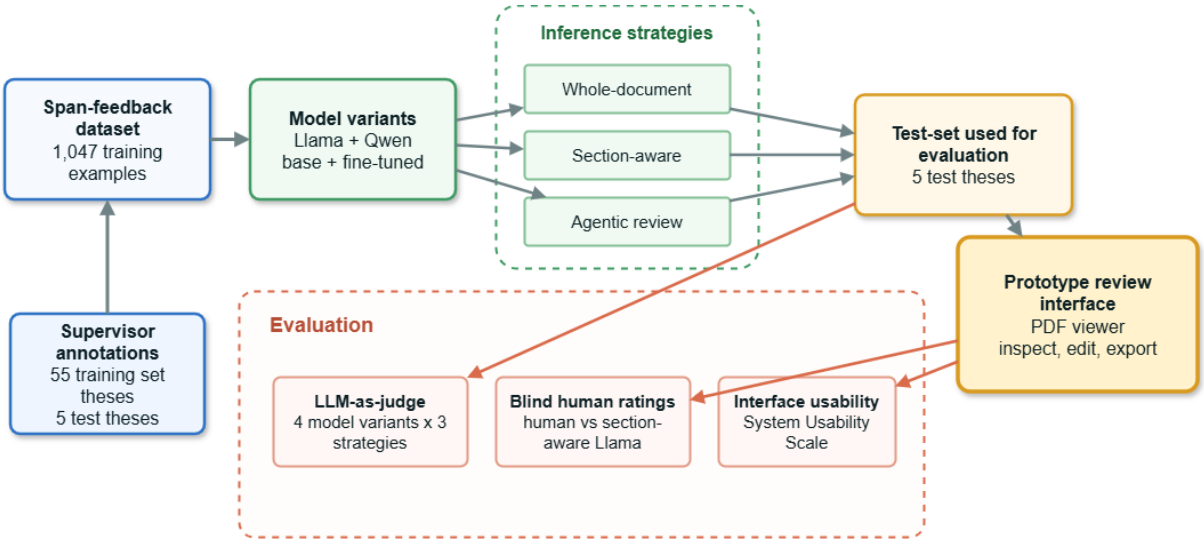

Figure 1: End-to-end pipeline. Five annotated theses are held out before fine-tuning. Evaluation combines blind human ratings, judge-based comparison, and interface usability testing.

and scholarly review reports that the resulting critique is frequently relevant but shallow, misprioritized, or weakly grounded [12, 13, 14, 11, 15]. Most of this evidence comes from closed-source models, and the writing-feedback subset typically targets shorter texts or bounded writing tasks. Open-weight models for long-document thesis feedback under supervisor control remain underexplored. Prior work on AI-generated writing feedback also shows that feedback usefulness depends not only on the generated text, but also on whether users can inspect, adapt, and integrate that feedback into an actual revision workflow [12, 16, 17].

Third, drafts and supervisor comments are personal educational data. UNESCO guidance on generative AI in education emphasizes privacy and institutional validation [18], the EU AI Act imposes transparency and human-oversight requirements on high-risk educational uses [19], and General Data Protection Regulation (GDPR) principles such as data minimization apply to draft-assistance tools by default [20]. These privacy, transparency, and human-oversight requirements make closed proprietary interfaces difficult to justify and motivate locally deployable open-weight models, where institutions can retain control over data handling and adaptation.

A final design issue is context selection. Thesis drafts are long, so a feedback system must decide how much of the document to show the model when generating each comment. Larger context windows reduce truncation but do not guarantee that the relevant evidence is used [21, 22]; retrieval-augmented generation and other context-selection methods can narrow the input around a target passage [23, 24]. Whole-document generation, section-aware two-stage generation, and agentic review (Section 3.4) occupy different points on this local-grounding-versus-global-context trade-off.

Figure 1 gives an overview of the entire pipeline. To address these gaps, we study span-anchored draft feedback with base and fine-tuned open-weight LLMs, meaning models whose trained parameters can be downloaded and run locally. We also test the resulting workflow through a PDF review interface developed for supervisors.

This setup leads to four research questions.

RQ1: To what extent can fine-tuned span-anchored feedback match or improve on supervisor comments under blind human rating?

RQ2: How do base and fine-tuned open-weight models compare across whole-document, section-aware two-stage, and agentic review strategies on long thesis drafts?

RQ3: How usable is a supervisor-facing PDF review interface, as measured by the System Usability Scale (SUS)?

RQ4: Which feedback-quality and workflow risks persist when LLM comments are used under supervisor control?

To our knowledge, this is the first work to combine these elements into a single workflow

for LLM-generated thesis feedback. Specifically, we contribute: (i) a controlled three-way comparison of whole-document, section-aware two-stage, and agentic generation strategies on the same fine-tuned open-weight backbones, isolating the context-scope trade-off that earlier work assembles across baselines from different systems; (ii) a reproducible on-premise pipeline for fine-tuning and serving open-weight 27B–70B models on supervisor-annotated thesis data using 4-bit-quantised LoRA on a single A100, with anonymisation and JSON-validated span-anchored inference; (iii) a supervisor-facing PDF review interface with backbone-agnostic model routing via LiteLLM, so the same workflow can adopt newer open-weight or commercial models without re-engineering; and (iv) a dual evaluation pairing an LLM-as-judge benchmark across all twelve system configurations with a blind human rating study on the deployed system, including explicit judge–human calibration.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 defines the task, data pipeline, fine-tuning setup, and generation strategies. Section 4 describes the supervisor-facing PDF review interface. Section 5 presents the evaluation design, and Section 6 reports the results. Section 7 discusses the findings and deployment implications, Section 8 covers limitations and ethics, and Section 9 concludes.

2 Related Work

Educational-feedback research treats feedback as useful when it changes how learners revise and think, not just when it points out errors [1, 25, 26]. In thesis supervision specifically, feedback also carries disciplinary norms, and what supervisors emphasize varies widely between individuals and programs [3, 2, 27]. That variation is the reason thesis feedback cannot be evaluated against a single gold-standard comment: several different comments can be useful for the same passage.

Existing work on AI-generated feedback consistently evaluates along several axes at once, such as correctness, specificity, revision usefulness, and user preference [12, 13, 28, 29, 14, 30, 15, 31]. This literature establishes useful evaluation dimensions, but it less often studies long technical drafts, open-weight local models, or supervisor-controlled review workflows.

Scholarly-review research is the closest related literature: it also generates critique over long academic documents [32, 33, 34, 35, 36]. At scale, useful critique overlaps only partly across human reviewers, which is why this literature evaluates by utility rather than by agreement with one reviewer [11]. Thesis feedback still differs from peer review in ways that shape our design: it is formative, not editorial; course-specific, not venue-specific; and usually attached to passages in a draft instead of written as a standalone report.

Anchoring a comment to a visible passage has both an empirical and a formal motivation. Empirically, writers act on comments more reliably when the comment names the problem location and a possible fix, and justified comments correlate with better learning outcomes [6, 7]. For LLM-generated feedback, anchoring also provides a guardrail against unsupported critique by tying each comment to visible evidence [37, 38]. Formally, the Web Annotation model defines an annotation as a body–target pair with a selector pointing into a text segment [39]; our span-anchored outputs follow this body–target structure, with the anchor span as target and the generated comment as body.

Anchoring does not resolve the long-document problem. The model must still balance a local passage against enough of the surrounding draft to judge contribution and consistency. Long-context prompting passes the whole draft to the model but suffers from recall loss on information placed in the middle of the input [21] and effective context windows smaller than advertised [22]; retrieval-based methods instead condition generation on selected passages [23], trading global coverage for tighter local grounding, with hybrid schemes combining both [24]. This evidence comes from knowledge-intensive question answering and synthetic long-context probes, where a single correct answer can be retrieved or localized; thesis feedback has no such reference and may require either local or global context per comment, motivating an empirical comparison of whole-document, section-aware two-stage, and agentic generation.

Evaluating generated feedback introduces a separate challenge. Sadler [40] argues that preset criteria cannot fully represent qualitative assessment judgment, and the peer-feedback literature confirms this empirically: there is no settled single definition of quality [6, 7, 8, 41, 9, 42]. We therefore evaluate feedback along multiple dimensions, not against a reference comment. The rubric combines dimensions common in writing-feedback and

review-utility work: whether a comment is relevant to the passage, clear about the required change, correct and grounded in the text, actionable for revision, specific, and suitable for supervisor use. Our dimension set is closest to Sadallah et al. [10], who operationalize review-comment utility through actionability, grounding, verifiability, and helpfulness.

3 Methodology

This section defines the span-anchored feedback task, describes how training examples were recovered from CSE3000 supervisor-annotated PDFs, gives the LoRA fine-tuning setup, and specifies the three generation strategies that the rest of the paper compares.

3.1 Task Formulation and Output Format

We define *span-anchored thesis-feedback generation* as the task of producing one feedback comment for a highlighted thesis passage. Given an anchor span s from a thesis, surrounding section context c, and a course-guideline prompt p, the model generates one feedback item y: a concise, revision-oriented comment grounded in the highlighted text. A well-formed feedback output addresses the selected passage and, in the structured output schema, returns the anchor text as a verbatim copy of the excerpt. An example of the task input and output format is provided in Appendix C.

Verbatim copying is enforced because LLMs are known to fabricate plausible-looking content that does not exist in the input, including bibliographic references and source quotations [43, 44, 45]. We adopt the verbatim-quoting principle of Menick et al. [46], who enforce it through reinforcement learning on quote-verified answers; here we apply it via fine-tuning on span-anchored supervisor data and a prompt-level format constraint that requires an exact-substring match against the provided excerpt. This rules out paraphrased or fabricated anchors.

In our setup, each anchor span connects three stages of the workflow. It defines what the model comments on, what raters use as evidence, and where the generated suggestion appears in the PDF interface. This anchor-centric design implements prior work on localized, justified feedback [6, 7].

3.2 Training Examples and Held-Out Split

The course data consist of CSE3000 student thesis drafts with supervisor feedback stored in annotated PDFs, in three forms: comments attached to highlighted text, sticky notes, and, in some cases, separate text-file feedback linked to the draft. The preprocessing pipeline first anonymized the PDFs by removing personally identifiable student and supervisor information, then recovered examples of the form (s, c, y), where s is the highlighted thesis span, c is the surrounding section context, and y is the corresponding supervisor comment. Thesis main text was extracted using layout-aware PDF parsing [47, 48] to preserve section hierarchy and paragraph boundaries. Highlighted comments were then aligned automatically by mapping each highlight's geometry to the extracted text. Sticky notes and text-file feedback lack explicit text targets and were aligned manually by exact-quote selection against the extracted section text.

Five documents were randomly selected and held out as a test set before filtering. The remaining documents formed the training pool. During preprocessing, feedback items shorter than five words were removed when they did not provide enough information to function as interpretable standalone comments in context. After filtering, the fine-tuning split contained 1,140 span–feedback examples from 55 documents. The held-out test set was used for both model comparison and human evaluation.

3.3 Prompt Design and Fine-Tuning

Each generation strategy uses the same core instruction prompt. The prompt asks the model to prioritize thesis-writing issues related to the research objective, methodology, argumentation, results interpretation, and scholarly support; follow exclusion rules for low-value comments such as placeholders, `TODO`-style drafting notes, and minor proofreading; and return an exact copied text span with one concise, actionable suggestion. Per-strategy prompt structure and post-filters are detailed in Appendix A.

We use Llama 3.3 70B Instruct and Qwen 3.5 27B as open-weight backbones. Both run on local university infrastructure, support long-context inputs, and are adapted in our setup with low-rank adaptation (LoRA) [49, 50, 51]; in pilot runs they were the two models that most reliably followed the structured output schema and copied anchor spans verbatim. Local open-weight deployment also enables auditing and keeps student data under institutional control [52, 18, 20]. Both backbones are fine-tuned with response-only supervised

LoRA fine-tuning under 4-bit quantization (rank 16, $\alpha = 32$, dropout 0) for one epoch at a maximum sequence length of 8,192 tokens, with learning rate 5×10^{-5} and effective batch size 32; full hyperparameters are reported in Appendix B. Because fine-tuning gains depend on model size, data, and task [53, 54], we evaluate both base and fine-tuned variants.

3.4 Generation Strategies

We compare three test-time strategies for the same feedback task: whole-document generation, section-aware two-stage generation, and agentic review. They differ in how much of the thesis the model sees, how the prompt is structured, and whether generation is split into multiple steps.

Whole-document generation. The full parsed thesis is passed to one long-context prompt. This global-context baseline tests whether direct access to the whole draft helps the model reason about contribution, method, and consistency. Relevant local evidence can still be missed inside a long input, especially when the effective context is smaller than the advertised window [21, 22].

Section-aware two-stage generation. This strategy narrows the input before generation by separating passage selection from feedback writing. A first call selects a candidate sentence from a section window; a second call writes feedback using that sentence and its surrounding context. This avoids asking one prompt to both identify where feedback is needed and generate the feedback, while following context-selection approaches that condition generation on selected evidence [23, 24]. This is the method used in the blind human study, since it was the deployed system at the time and produces one local comment per inspected span.

Agentic review. The agentic pipeline uses role-specific prompts for clarity, structure, methodology, results, and consistency. Candidate comments are checked for support, optionally revised, and deduplicated before output. This strategy tests whether role specialization and self-checking change coverage and actionability relative to a single prompt, following multi-agent work that decomposes complex tasks into specialized roles and verification steps [55, 56, 57].

4 Supervisor Review Interface

The prototype is a React/TypeScript frontend over a FastAPI backend. A supervisor can upload a thesis PDF, request generated feedback per section, edit or remove individual comments, and export the annotated PDF. Figure 2 shows the resulting workflow: generated comments enter as editable draft annotations beside the PDF and remain under supervisor control until export.

5 Evaluation Design and Analysis

The evaluation pairs a blind human rating study on held-out theses with an LLM-as-judge benchmark across all twelve system configurations, and an interface usability study; together these target the four research questions while letting human and automatic measurements calibrate each other.

5.1 RQ1/RQ4: Blind Feedback-Quality Study

The first study measures generated-feedback quality (RQ1) and adoption risks noticed by raters (RQ4). It uses five held-out theses. Participants were supervisors, postdoctoral researchers, and PhD candidates recruited from TU Delft and adjacent institutions, representing expert or near-expert users familiar with academic draft review. Participants saw the thesis PDF, one candidate feedback comment at a time, and the excerpt to which the comment was anchored. They could inspect the surrounding document context before rating the comment. Source labels were hidden, so participants did not know whether a comment came from a human supervisor or the LLM. Each completed participant rated 15 comments across three PDFs: six human comments from the held-out thesis test set and nine LLM comments from the fine-tuned Llama 3.3 70B section-aware pipeline.

Participants rated each comment on the six dimensions in Table 1. The same rubric was used for blind human ratings and the LLM-as-judge benchmark. Relevance, change clarity, correctness, actionability, and specificity used a symmetric five-point agreement scale from strongly disagree (-2) to strongly agree ($+2$), with neutral at 0. This coding keeps positive and negative judgments comparable across dimensions. Supervisor suitability instead used a four-point use-decision scale: not suitable (-2), suitable after major revision (-1), suitable after minor editing ($+1$), and suitable as-is ($+2$). This scale captures the practical question of whether a supervisor could send the comment

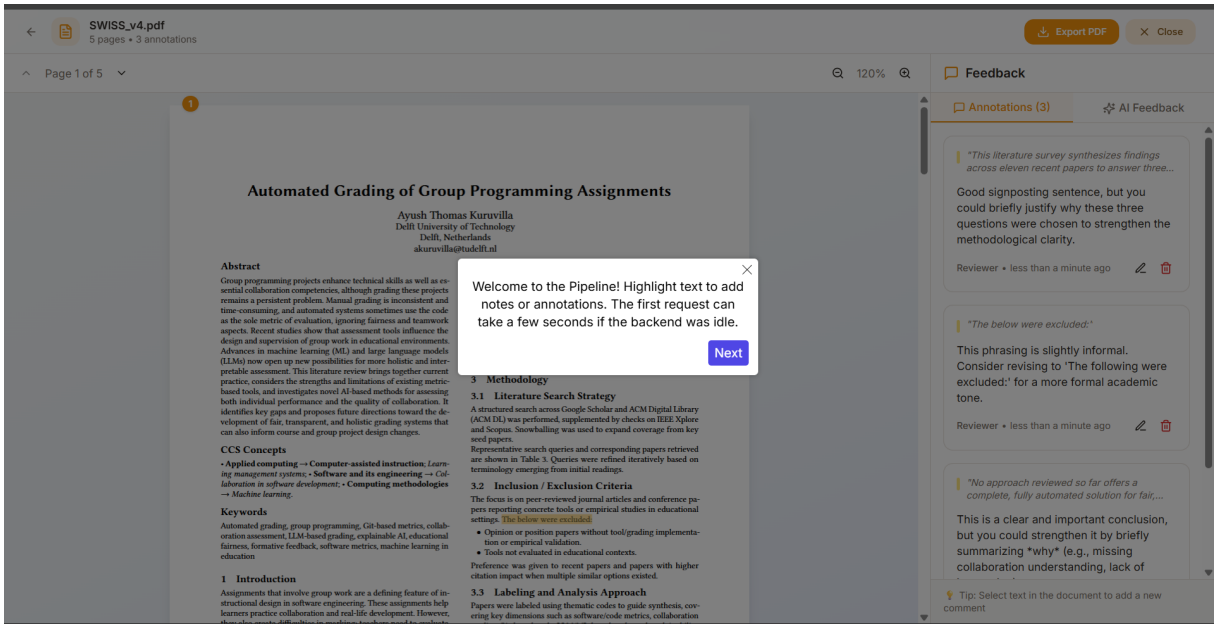

Figure 2: Supervisor-facing thesis-feedback interface. Generated suggestions are shown beside the thesis PDF, so supervisors can inspect the anchor text before revising, removing, or exporting a comment. Model output enters the workflow as editable draft feedback for supervisor review.

to a student as written, after editing, or not at all. The completed dataset contains 22 raters and 330 comment-level ratings (132 human-comment and 198 LLM-comment).

5.2 RQ2: LLM-as-Judge Strategy Comparison

The judge-based benchmark compares generation strategies (RQ2) across twelve system configurations on the five held-out theses, spanning base and fine-tuned Llama and Qwen variants under each of the three strategies. The benchmark includes all combinations of model backbone, fine-tuning status, and generation strategy, allowing us to compare both individual factors and complete system configurations.

Three local Ollama judges score each candidate comment: Mistral Small, DeepSeek-R1 32B Qwen Distill, and Gemma 4 31B. All three use the same six-dimension rubric from the blind study, yielding 1,491 successful item-level judgments across the two judge runs. Because the judges differ in scoring severity, we report *judge-macro* means: approach-level means are first computed separately for each judge and then averaged across judges.

5.3 RQ3/RQ4: Interface Usability Study

The interface study measures usability (RQ3) and the workflow risks supervisors notice in use (RQ4). Participants completed a task-based session that involved uploading a thesis PDF, reviewing LLM-generated comments, editing or removing individual comments, and exporting the annotated PDF. They then answered the ten-item System Usability Scale (SUS) in Qualtrics [58]. We report SUS on its standard 0–100 scale, against the common benchmarks of 68 (average) and 80 (good to excellent) [59]. The interface study includes 21 completed SUS responses. The interface-study pool overlaps almost entirely with the 22 blind-study raters, with one participant present in the blind study who did not complete the interface session.

5.4 Analysis Plan

For the blind study, we report participant-weighted means and bootstrap raters' LLM–human differences over participants, with comment scores nested within raters. Paired Wilcoxon signed-rank tests are run on the same participant-level contrasts; we use Wilcoxon instead of a paired t-test because the rubric is ordinal on a $[-2, 2]$ five-point scale and the participant-level contrasts are not constrained to be Gaussian. For SUS, we compute the per-participant 0–100 score and bootstrap the mean over respondents; bootstrap intervals are reported instead of a parametric normal approximation because the SUS distribution is bounded and typically skewed at the upper end. Open-ended comments from both studies are coded thematically and used to explain the numeric pattern, not to score it.

Table 1: Rubric used for both blind human ratings and the LLM-as-judge benchmark, adapted from prior work on writing feedback, review utility, and thesis-style scholarly review [12, 15, 14, 10, 29, 36].

Dimension	Operational definition
Relevance	The feedback directly addresses the shown excerpt and stays on-topic.
Change clarity	The feedback makes it clear whether the student should keep, revise, remove, or otherwise change the text.
Correctness	The feedback is factually and contextually justified by the visible excerpt and does not introduce unsupported claims.
Usefulness / actionability	The feedback gives concrete guidance that could help the student improve the text instead of only pointing out a problem.
Specificity	The feedback is grounded in specific words, phrases, claims, or parts of the excerpt instead of remaining overly general.
Supervisor suitability	The feedback would be suitable for a thesis supervisor to send to a student as written, or after only minor editing.

For the LLM-as-judge benchmark, systems produced different numbers of feedback comments, so individual comment scores were not treated as independent evidence for strategy-level significance. Instead, we first averaged the six rubric scores within each judged comment, then averaged comments within each thesis–judge–model block for each strategy. Statistical comparisons were made on paired thesis-level differences, using Wilcoxon signed-rank tests and bootstrap confidence intervals over the five held-out theses. This treats the thesis, not the generated comment, as the main unit of inference.

6 Results

Results are organised by research question: strategy ranking from the LLM-as-judge benchmark, blind human ratings of the deployed system against held-out supervisor comments, and System Usability Scale ratings for the supervisor-facing interface.

6.1 RQ2: Inference Strategy Comparison

Figure 3 reports judge-macro means for the two fine-tuned models only, where a *block* is one thesis–judge–model cell of averaged comment scores and a model is one of the four base or fine-tuned Llama 3.3 and Qwen 3.5 variants. Whole-document prompting obtains the highest overall score (1.34), followed by agentic review (1.32) and section-aware two-stage generation (1.28). Whole-document prompting leads on relevance, correctness, and supervisor suitability; agentic review leads on actionability; and section-aware generation remains strongest on change clarity and specificity but trails on correctness and actionability.

Across all base and fine-tuned model variants, whole-document prompting has the highest thesis-balanced judge-macro score, followed by section-aware generation and agentic review. However, thesis-level pairwise tests do not show a statistically significant strategy effect at the conventional $p < .05$ threshold. The same conclusion holds when the comparison is restricted to the two fine-tuned models: pairwise comparisons between whole-document prompting, section-aware generation, and agentic review are all non-significant. The strategy comparison is therefore best interpreted as descriptive evidence of a context-scope trade-off rather than conclusive evidence that one strategy is reliably superior.

Figure 4 addresses the fine-tuning question directly by comparing base and fine-tuned variants within each model family and inference strategy. Fine-tuning does not produce a uniform gain in judge score: in several settings the base model remains competitive or higher, indicating that fine-tuning changes the generated-feedback profile but does not consistently improve judge-macro scores.

The matched thesis-level tests support this interpretation. None of the six base–fine-tuned comparisons within a fixed model family and generation strategy reaches statistical significance. When the matched comparisons are pooled across model family and generation strategy, the fine-tuned variants score slightly lower on average than their base counterparts, but this trend is not significant ($\Delta = -0.09$, $p = .062$). Thus, the judge benchmark does not provide evidence that fine-tuning reliably improves overall judge-macro performance on the held-out theses.

6.2 RQ1/RQ4: Blind Feedback Quality

The blind study compares generated-comment quality with held-out supervisor comments and records remaining risks (RQ1, RQ4). Figure 5 visualizes participant-weighted ratings for human comments

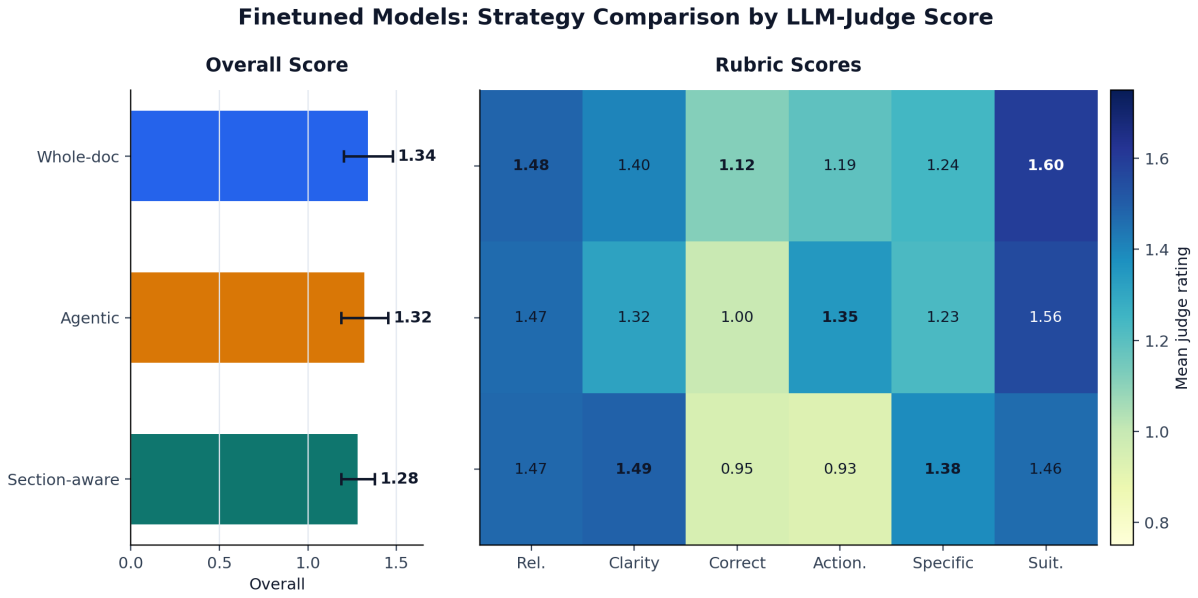

Figure 3: LLM-as-judge strategy comparison for the two fine-tuned generators (fine-tuned Llama and fine-tuned Qwen). Scores use the $[-2, 2]$ rubric scale (higher is better) and are macro-averaged over judge, thesis, and model blocks. The left panel shows overall score with 95% bootstrap confidence intervals; the right panel breaks each approach down by rubric dimension.

and the deployed section-aware Llama system on the six dimensions defined in Table 1; Appendix Table 3 reports full means, bootstrap confidence intervals, and paired Wilcoxon signed-rank tests.

Across 22 raters, LLM feedback was rated higher than human feedback on five of six dimensions, but only specificity ($+0.47$, $p = .018$) and change clarity ($+0.40$, $p = .027$) reached significance. Correctness was effectively tied (-0.04, $p = .516$); relevance, actionability, and supervisor suitability were positive on average but inconclusive.

Figure 6 reports the inter-rater agreement between the LLM judges and the human raters on the subset of LLM comments that can be matched directly between the human study and judged outputs. On this matched subset the judges are more generous than the humans on every dimension (mean bias $+0.41$, mean absolute error (MAE) 0.53), most strongly on supervisor suitability ($+0.85$), so judge-based strategy comparisons should be interpreted as relative system diagnostics, not as a replacement for human acceptability judgments. Open-ended remarks collected alongside the ratings are analyzed thematically and used to interpret these patterns in Section 7.

6.3 RQ3/RQ4: Interface Usability

The interface received a mean SUS score of 77.5 (median 85.0, standard deviation (SD) 16.58), with a bootstrap 95% confidence interval of $[70.0, 83.9]$ over the 21 respondents who completed every SUS item. Using standard SUS interpretation, this places the interface in the above-average usability band [59].

Figure 7 shows the reverse-coded item means, where higher values are better for all ten items. Participants scored the interface especially high on not needing technical support ($Q4 = 4.71$) and learning it quickly ($Q7 = 4.33$). The weakest item is intended frequency of use ($Q1 = 3.10$), suggesting that the interface is usable even if routine adoption remains uncertain. Open-ended comments on interface use are analyzed thematically and translated into the deployment requirements reported in Section 7.

7 Discussion

7.1 Principal findings

Prior work on LLM-generated thesis and paper feedback does not characterise how much of a long draft the model should see, and does not report results for open-weight models in this setting. We tested three approaches spanning the local-versus-global context choice (whole-document,

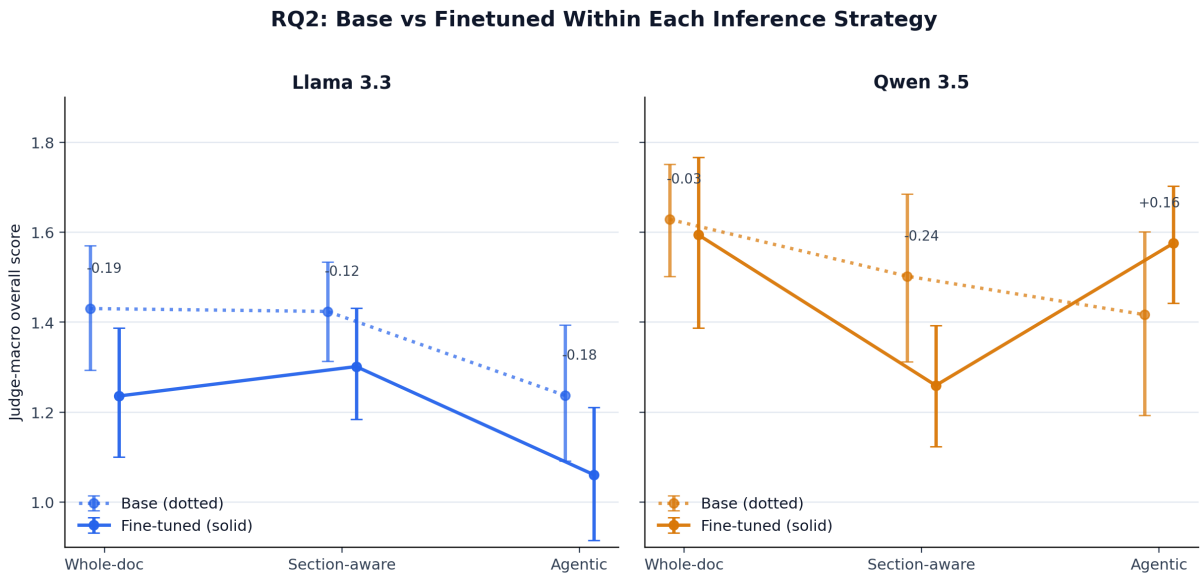

Figure 4: Base versus fine-tuned judge-macro overall scores within each model family and inference strategy, on the $[-2, 2]$ rubric scale (higher is better). Error bars show 95% bootstrap intervals over matched judge–thesis blocks. The annotated values above each strategy show fine-tuned minus base score.

section-aware two-stage, and agentic review) on two open-weight backbones, and the main finding is a context-scope trade-off (Figure 3): each approach won on the dimensions its context shape favoured. The trade-off is therefore a structural property of span-anchored draft feedback, not of any one system. Under blind rating against held-out supervisor comments, the deployed system improved significantly on the local dimensions (specificity, change clarity) and was tied on correctness, meaning open-weight LLMs can produce useful local draft feedback on long technical theses but do not yet substitute for global supervisory judgment. Fine-tuning on the span-anchored corpus was numerically higher in only one of six matched comparisons (Qwen 3.5 under agentic review); in the other five the base model scored at or above its fine-tuned counterpart (Figure 4), and the supervised pool primarily shifted the response distribution without expanding underlying critique ability, an outcome consistent with prior fine-tuning analyses on small narrow-target corpora [53, 54]. Finally, on the comments scored by both, the three independent LLM judges rated the system above the blind human raters on every dimension (mean bias $+0.41$, MAE 0.53) and most strongly on supervisor suitability ($+0.85$). The judges are therefore systematically more generous than humans, with the gap largest on the deployment-relevant dimension.

7.2 Relation to prior work on automated review and AI writing feedback

Most LLM-as-reviewer work targets conference papers with closed proprietary models: Kang et al. [32] benchmark acceptance prediction and aspect scoring, Liu and Shah [33] generate GPT-4 reviews that handle simple checks but lose accuracy where a global view is needed, and Liang et al. [11] find GPT-4 reviews partially overlap human reviewers yet are rated helpful by authors. Zhou et al. [34] caution that single-judge LLM evaluation is not reliable enough for deployment claims, and Sun et al. [36] review master's theses on a small sample without a supervisor-facing workflow. Our work differs on three axes that matter for deployment. The artefact is a long bachelor thesis inside live supervision, so we score actionability and supervisor suitability rather than predictors of acceptance. The backbones are open-weight and run on-premise, where these studies depend on closed APIs that cannot meet the privacy and oversight constraints of student data. And our evaluation is dual: a blind comparison against held-out supervisor comments, where the LLM scores higher on local dimensions (specificity, change clarity) but not correctness, matching Liang et al. [11]; and a multi-judge calibration in which the optimism we measured (mean bias $+0.41$, rising to $+0.85$ on supervisor suitability) persisted across three independent judges, extending Zhou et al. [34]'s single-

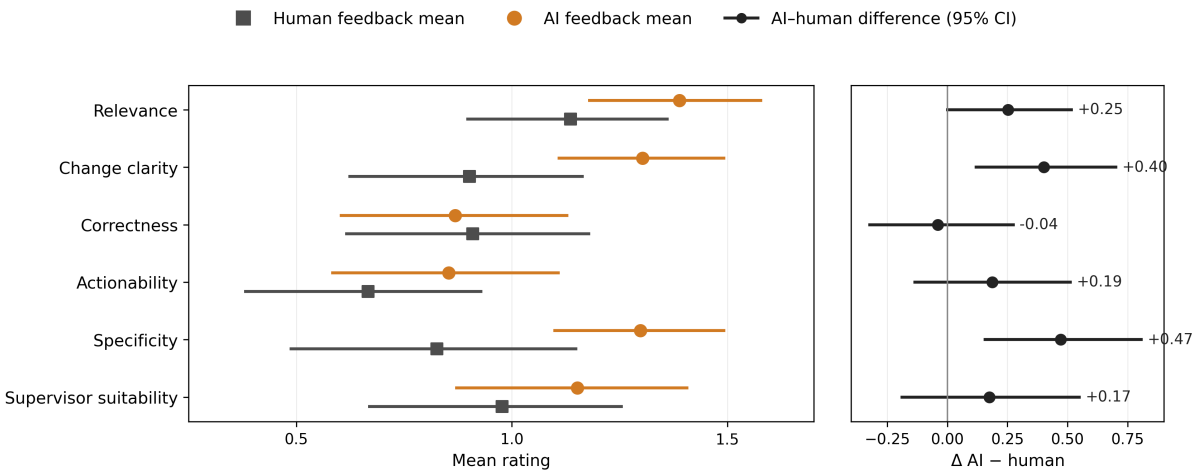

Figure 5: Participant-weighted blind-study ratings for human and LLM feedback across six evaluation dimensions. Scores use the $[-2, 2]$ rubric scale (higher is better). The left panel shows source means with 95% bootstrap confidence intervals; the right panel shows paired participant-level differences (Δ LLM$-$human). Exact numerical results and paired Wilcoxon signed-rank tests are reported in Appendix Table 3.

judge concern to the judging task itself.

The AI writing-feedback literature provides the vocabulary for our rubric. Studies of AI feedback on short texts evaluate outputs along several dimensions at once, such as correctness, specificity, actionability, and helpfulness [12, 14, 30, 15, 31, 29, 28]; Sadallah et al. [10] is closest to our six-dimension rubric, formalising review-comment utility along similar axes. We extend this multi-dimension approach to a setting the literature does not cover: full thesis drafts of fifteen to forty pages, open-weight backbones, and a three-way strategy comparison sharing one backbone, prompt format, and rubric. The long-context literature [21, 22, 23, 24] characterises the same context-selection trade-offs we observed, but on knowledge-intensive question answering and synthetic long-context probes where one correct answer can be localised. Thesis feedback has no such reference, motivating our empirical test of three strategies inside the same backbone and rubric.

7.3 Practical implications and beneficiaries

Our interface ratings and open-ended responses converge on a small set of requirements for course-level deployment. Raters consistently flagged a missing *why* as the marker separating supervisor-grade feedback from locally accurate but shallow comments, the property educational-feedback work associates with revision uptake [6, 7], and our rubric does not yet score it. Supervisors valued inspecting, editing, and removing each comment before export, and exported PDFs need to visually distinguish LLM-authored from supervisor-authored annotations. Generation should also be conditioned on existing supervisor comments so it does not duplicate critique already in the draft. We treat personalisation as a data-quality problem, since high-level feedback is often given verbally and never written into the PDF, leaving supervisor corpora only a partial signal. Across all of these, the supervisor stays the decision-maker rather than the calibrator [40].

We see three groups of beneficiaries. For institutions under data-protection and high-risk-AI obligations [18, 19, 20], we show a fully on-premise pipeline, from anonymising supervisor-annotated PDFs to 4-bit-quantised LoRA fine-tuning on a single A100 and JSON-validated span-anchored inference, that runs on institutional hardware without sending student drafts to a closed API. For supervisors, our React/FastAPI interface routes models through LiteLLM and is backbone-agnostic, so the workflow can adopt newer models without re-engineering. For researchers, we test a controlled three-way strategy comparison on one backbone family with multi-judge calibration against blind human ratings.

8 Limitations and Ethics

The study uses a single TU Delft course, five held-out theses, and 22 blind raters drawn from supervisors, postdoctoral researchers, and PhD candidates. The ratings reflect informed expert judg-

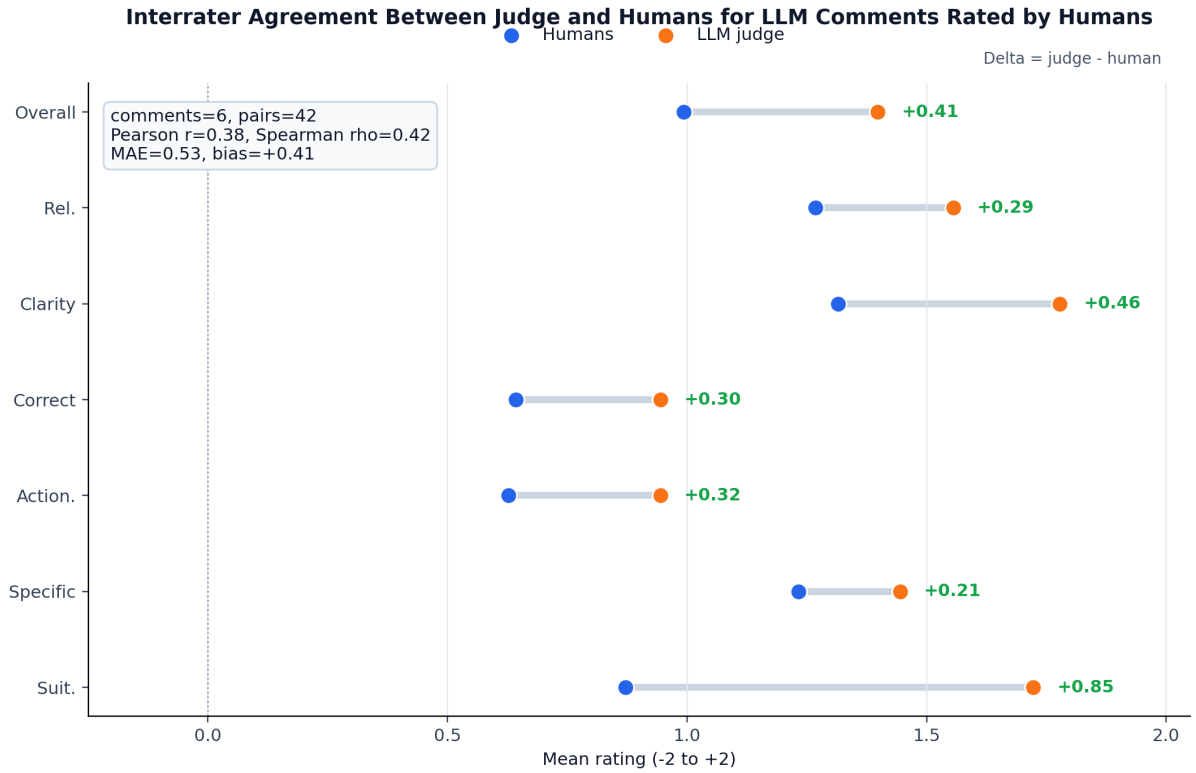

Figure 6: Interrater agreement between LLM judges and human raters for LLM comments that were rated by humans and matched to judged outputs. Each row compares the human mean rating with the corresponding judge mean on the same $[-2, 2]$ rubric scale (higher is better); the right-side value is judge minus human.

ment but not the broader population of users (students, supervisors from other disciplines, external examiners), and we did not measure learning gains or supervisor workload reduction. Only the deployed section-aware Llama route appears in the blind study; the remaining strategies were compared through the LLM-as-judge benchmark, an automatic proxy whose calibration diverges most on supervisor suitability (on the matched comments, judges scored LLM suitability 0.85 above blind raters).

The rubric does not score revision rationale, which raters identified as the main reason locally accurate feedback could still appear shallow; future evaluations should score it separately and test whether students revise better with the feedback than without. The evaluation is also text-only: figures, tables, equations, and code listings were not analyzed, and the training data themselves capture only written PDF comments, while substantive supervisor feedback is often delivered in meetings. The model is therefore trained on a partial signal of supervisor judgment, which constrains both current performance and any supervisor-personalized extension.

Thesis excerpts and feedback were anonymized, and participants in the human studies provided informed consent before taking part. Participants were recruited by email invitation, targeted at the CSE department and at supervisors of the course; participation was unpaid and based on interest. Course deployment would require provenance on every LLM-authored annotation, retention rules, student opt-out, and supervisor sign-off before export, in line with GDPR and AI Act obligations for high-risk educational use [18, 20, 19]. The risk that LLM-generated feedback is polished but unsupported is partly mitigated by keeping LLM annotations editable, rejectable, and visibly marked in the developed interface.

9 Conclusion

This research investigated whether fine-tuned open-weight LLMs can produce reviewable draft feedback on long bachelor-thesis drafts under supervisor control. Under blind rating, the deployed section-aware Llama 3.3 70B route was rated above held-out supervisor comments on specificity ($+0.47$, $p = .018$) and change clarity ($+0.40$, $p = .027$), with no statistically significant dif-

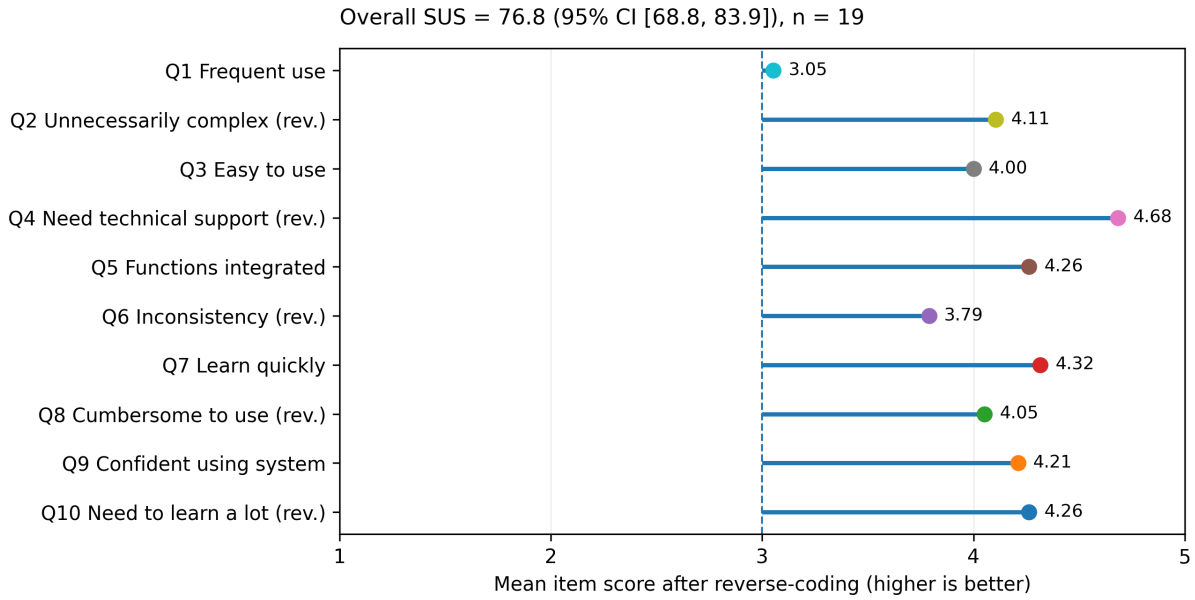

Figure 7: System Usability Scale results for the interface study after reverse-coding the negatively phrased items, so higher values are better for all items. The dashed vertical line marks the neutral midpoint of 3 on the 1–5 item scale. Overall SUS is 77.5 with a 95% bootstrap confidence interval of $[70.0, 83.9]$ over the 21 respondents who completed every SUS item.

ference on correctness (-0.04). The LLM-as-judge benchmark ranked whole-document generation first across all three judges, and the PDF review interface reached a mean SUS of 77.5 across 21 respondents [59]. Within this course setting and held-out thesis pool, the results support a narrower claim: open-weight LLMs can produce useful first-pass draft feedback for supervisor review on long technical theses. In this study, the system was strongest on local, span-level issues and did not replace higher-level supervisory critique of argument, method, or contribution. More broadly, the pattern reported here, a controlled three-way comparison of generation strategies on the same fine-tuned open-weight backbone paired with blind human rating and judge–human calibration, gives institutions a reusable template for evaluating LLM feedback systems on long technical writing under data-protection constraints.

Future work. Several near-term directions follow from these results. The most important is to develop and evaluate a complementary higher-level generation mode so that the system covers contribution-, methodology-, and argument-level critique alongside its current local strengths. Because fine-tuning helped in only one of six matched comparisons, a parallel direction is to improve the supervised data itself by collecting more, longer, and rationale-rich supervisor comments, so that adaptation becomes a more consistent contributor instead of a stylistic shift; the rubric should also add an explicit revision-rationale dimension and the supervised targets should include the *why* alongside the actionable suggestion. A further direction is to test whether students revise drafts better with the generated feedback than without it, since the present study measures perceived comment quality, not learning outcomes. Finally, the calibration gap concentrated on supervisor suitability suggests training or selecting LLM judges against human suitability ratings, not only against the multi-dimension rubric, before any judge benchmark is used as a deployment-readiness signal.

References

[1] John Hattie and Helen Timperley. The power of feedback. *Review of Educational Research*, 77(1): 81–112, 2007. doi: 10.3102/003465430298487.

[2] John Bitchener, Helen Basturkmen, and Martin East. The focus of supervisor written feedback to thesis/dissertation students. *International Journal of English Studies*, 10(2):79–97, 2010. doi: 10.6018/ijes/2010/2/119201.

[3] Vijay Kumar and Elke Stracke. An analysis of written feedback on a PhD thesis. *Teaching in Higher Education*, 12(4):461–470, 2007. doi: 10.1080/13562510701415433.

[4] Gosia Migut, Aleksander Buszydlik, and Mathijs M. De Weerdt. The research project in computer science bachelor education: Undergraduate research experience at scale. In *ITiCSE 2025: Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 410–416, New York, NY, 2025. Association for Computing Machinery. doi: 10.1145/3724363.3729101.

[5] Andrew S. Griffith and Zeynep Altinay. A framework to assess higher education faculty workload in U.S. universities. *Innovations in Education and Teaching International*, 57(6):691–700, 2020. doi: 10.1080/14703297.2020.1786432.

[6] Melissa M. Nelson and Christian D. Schunn. The nature of feedback: How different types of peer feedback affect writing performance. *Instructional Science*, 37(4):375–401, 2009. doi: 10.1007/s11251-008-9053-x.

[7] Sarah Gielen, Ellen Peeters, Filip Dochy, Patrick Onghena, and Katrien Struyven. Improving the effectiveness of peer feedback for learning. *Learning and Instruction*, 20(4):304–315, 2010. doi: 10.1016/j.learninstruc.2009.08.007.

[8] Yong Wu and Christian D. Schunn. From feedback to revisions: Effects of feedback features and perceptions. *Contemporary Educational Psychology*, 60:101826, 2020. doi: 10.1016/j.cedpsych.2019.101826.

[9] Wenjing He and Ying Gao. Explicating peer feedback quality and its impact on feedback implementation in EFL writing. *Frontiers in Psychology*, 14:1177094, 2023. doi: 10.3389/fpsyg.2023.1177094.

[10] Abdelrahman Sadallah, Tim Baumgärtner, Iryna Gurevych, and Ted Briscoe. The good, the bad and the constructive: Automatically measuring peer review's utility for authors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28991–29021, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1476.

[11] Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, Daniel A. McFarland, and James Zou. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8), 2024. doi: 10.1056/AIoa2400196.

[12] Wei Dai, Yi-Shan Tsai, Jionghao Lin, Ahmad Aldino, Hua Jin, Tongguang Li, Dragan Gašević, and Guanliang Chen. Assessing the proficiency of large language models in automatic feedback generation: An evaluation study. *Computers and Education: Artificial Intelligence*, 7:100299, 2024. doi: 10.1016/j.caeai.2024.100299.

[13] Juan Escalante, Austin Pack, and Alex Barrett. Ai-generated feedback on writing: insights into efficacy and enl student preference. *International Journal of Educational Technology in Higher Education*, 20 (57), 2023. doi: 10.1186/s41239-023-00425-2.

[14] Kathrin Seßler, Arne Bewersdorff, Claudia Nerdel, and Enkelejda Kasneci. Towards adaptive feedback with ai: Comparing the feedback quality of llms and teachers on experimentation protocols, 2025.

[15] Hannah Rashkin, Elizabeth Clark, Fantine Huot, and Mirella Lapata. Help me write a story: Evaluating llms' ability to generate writing feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25827–25847, 2025.

[16] Chao Zhang, Kexin Ju, Zhuolun Han, Yu-Chun Grace Yen, and Jeffrey M. Rzeszotarski. Synthia: Visually interpreting and synthesizing feedback for writing revision. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, 2025. doi: 10.1145/3746059.3747703.

[17] Maximilian Sölch, Felix T. J. Dietrich, and Stephan Krusche. Direct automated feedback delivery for student submissions based on llms. In *Companion Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 2025. doi: 10.1145/3696630.3727247.

[18] Fengchun Miao and Wayne Holmes. Guidance for generative AI in education and research, 2023. URL https://www.unesco.org/en/articles/guidance-generative-ai-education-and-research.

[19] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence, 2024. URL https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

[20] European Parliament and Council of the European Union. Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data, 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[21] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[22] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models?, 2024.

[23] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2020.

[24] Zhuowan Li, Cheng Xu, Mohit Iyyer, and Yashar Mehdad. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2024. doi: 10.18653/v1/2024.emnlp-industry.66.

[25] Valerie J. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008. doi: 10.3102/0034654307313795.

[26] David J. Nicol and Debra Macfarlane-Dick. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2):199–218, 2006. doi: 10.1080/03075070600572090.

[27] Susan Carter and Vijay Kumar. Ignorance, uncertainty and confusion: Writing feedback and doctoral supervision. *Innovations in Education and Teaching International*, 54(1):68–77, 2017. doi: 10.1080/14703297.2015.1123104.

[28] Mickie de Wet, Margarita Oja Da Silva, and René Bohnsack. Can AI give good feedback on essay-type assignments? an explorative case study of LLMs in higher education. *Innovations in Education and Teaching International*, 62(5):1484–1499, 2025. doi: 10.1080/14703297.2025.2536609.

[29] Hana Mohamed Nassar. Comparing the quality of ai-generated and instructor feedback in a university writing program. Master's thesis, The American University in Cairo, 2025. URL https://fount.aucegypt.edu/etds/2468.

[30] Seyyed Kazem Banihashem, Nafiseh Taghizadeh Kerman, Omid Noroozi, Jewoong Moon, and Hendrik Drachsler. Feedback sources in essay writing: peer-generated or ai-generated feedback? *International Journal of Educational Technology in Higher Education*, 21(23), 2024. doi: 10.1186/s41239-024-00455-4.

[31] İbrahim Rıza Hallaç and Hasan Oğul. From prompting to fine-tuning: A comprehensive evaluation of llm strategies for automatic essay assessment, 2025. URL https://ssrn.com/abstract=5699710. SSRN preprint.

[32] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, 2018.

[33] Ryan Liu and Nihar B. Shah. Reviewergpt? an exploratory study on using large language models for paper reviewing, 2023.

[34] Ruiyang Zhou, Lu Chen, and Kai Yu. Is llm a reliable reviewer? a comprehensive evaluation of llm on automatic paper reviewing tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9340–9351, 2024.

[35] Osama Mohammed Afzal, Preslav Nakov, Tom Hope, and Iryna Gurevych. Beyond "not novel enough": Enriching scholarly critique with llm-assisted feedback. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2648–2671, Rabat, Morocco, 2026. Association for Computational Linguistics.

[36] Pengfei Sun, Deqing Huang, Qingyuan Wang, Na Qin, and Liming Cai. Llm-based scholarly paper review for master's theses. In *2025 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Macao, China, 2025.

[37] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models. *ACM Computing Surveys*, 2025. doi: 10.1145/3703155.

[38] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, 2021. doi: 10.18653/v1/2021.findings-emnlp.320.

[39] Robert Sanderson, Paolo Ciccarese, and Herbert Van de Sompel. Web annotation data model. W3C Recommendation, 2017. URL https://www.w3.org/TR/annotation-model/.

[40] D. Royce Sadler. Indeterminacy in the use of preset criteria for assessment and grading. *Assessment & Evaluation in Higher Education*, 34(2):159–179, 2009. doi: 10.1080/02602930801956059.

[41] Kwangsu Cho and Charles MacArthur. Student revision with peer and expert reviewing. *Learning and Instruction*, 20(4):328–338, 2010. doi: 10.1016/j.learninstruc.2009.08.006.

[42] Cecilia Superchi, José Antonio González, Ivan Solà, Erik Cobo, Darko Hren, and Isabelle Boutron. Tools used to assess the quality of peer review reports: A methodological systematic review. *BMC Medical Research Methodology*, 19(1):48, 2019. doi: 10.1186/s12874-019-0688-x.

[43] William H. Walters and Esther Isabelle Wilder. Fabrication and errors in the bibliographic citations generated by ChatGPT. *Scientific Reports*, 13(1):14045, 2023. doi: 10.1038/s41598-023-41032-5.

[44] Mehul Bhattacharyya, Valerie M. Miller, Debjani Bhattacharyya, and Larry E. Miller. High rates of fabricated and inaccurate references in ChatGPT-generated medical content. *Cureus*, 15(5):e39238, 2023. doi: 10.7759/cureus.39238.

[45] Jake Linardon, Hannah Jarman, Zoe McClure, Cleo Anderson, Claudia Liu, and Mariel Messer. Influence of topic familiarity and prompt specificity on citation fabrication in mental health research using large language models: Experimental study. *JMIR Mental Health*, 12:e80371, 2025. doi: 10.2196/80371.

[46] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022.

[47] C. Ramakrishnan, A. Patnia, E. Hovy, and G. A. P. C. Burns. Layout-aware text extraction from full-text PDF of scientific articles. *International Journal on Document Analysis and Recognition*, 15(1):7–20, 2012. doi: 10.1007/s10032-011-0163-8.

[48] Yue Zhang, Zhihao Zhang, Yuzhuo Wang, Zhenyu Xi, Siyuan Yuan, and Xuanjing Huang. PDF-to-Tree: Parsing PDF text blocks into a tree. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10702–10715, 2024.

[49] Meta. Llama 3.3 70b instruct model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

[50] Qwen Team. Qwen3.5-27b model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-27B.

[51] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[52] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. OLMo: Accelerating the science of language models, 2024.

[53] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[54] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.779.

[55] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework, 2023.

[56] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Juergen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024.

[57] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[58] John Brooke. Sus: A "quick and dirty" usability scale. In Patrick W. Jordan, Bruce Thomas, Ian L. McClelland, and Bernard Weerdmeester, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor and Francis, London, 1996.

[59] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24:574–594, 2008. doi: 10.1080/10447310802205776.

A Prompt Design Details

All prompt variants share a common skeleton. The system message instructs the model to act as a thesis-feedback reviewer, ignore placeholders and `TODO`-style drafting notes, prioritise high-impact academic-writing issues over minor proofreading, and return concise, actionable feedback anchored to exact thesis text. The user message contains the course-guideline prompt, the surrounding section context, and (for whole-document or agentic modes) the parsed thesis text. All variants return structured JSON: a verbatim anchor span and a one-sentence comment.

Whole-document generation. The entire parsed thesis is passed to a single review prompt that requests supervisor-style critique across the complete document. A post-filter removes items whose anchor sentence is not an exact substring of the source, items whose anchor matches a placeholder pattern, and items that duplicate an already-accepted entry.

Section-aware two-stage generation. The parsed thesis is split into top-level sections; each section is partitioned into sliding windows of at most W characters with O characters of overlap. A Stage 1 selector prompt asks for at most one exact-quoted candidate sentence per window; the candidate is validated for exact occurrence in the section text, against a truncation heuristic, and

against a global deduplication set. A Stage 2 generator prompt then writes one concise, actionable comment for that candidate, conditioned on the entire section text and the selected sentence. A post-filter rejects comments that are too short, too vague, or match a list of generic feedback prefixes.

Agentic review. A coordinator assigns work to five role-specific reviewers (clarity, structure, methodology, results, and consistency), prioritising sections by heading-keyword priority and pending-role coverage. Each reviewer proposes a candidate span and comment for its role. A deterministic check enforces exact span match, placeholder avoidance, and duplicate-anchor rejection. A semantic verifier returns ACCEPT, REFINE, or REJECT; REFINE triggers up to $R_{\max}$ refinement attempts. A final deduplication step removes near-identical accepted comments. The loop terminates when the agenda is empty, when a patience counter elapses without new accepts, or when the iteration budget is reached.

Full prompt strings and the exact post-filter regular expressions are included in the supplemental thesis materials.

B Fine-Tuning Configuration

Both backbones use the same supervised LoRA configuration; only the chat template differs between Llama and Qwen. Settings are summarised in Table 2.

Table 2: Supervised LoRA fine-tuning configuration.

Hyperparameter	Value
Quantisation	4-bit (Unsloth NF4-style)
Compute dtype	`bfloat16` on compute capability ≥ 8.0, `float16` otherwise
LoRA rank r	16
LoRA α	32
LoRA dropout	0.0
Loss target	assistant response only
Max sequence length	8,192 tokens
Per-device train batch size	1
Gradient-accumulation steps	32 (effective batch size 32)
Learning rate	5×10^{-5}
Number of epochs	1.0
Optimiser	AdamW (TRL default)
Random seed	42
Hardware	DelftBlue, single NVIDIA A100 80 GB

One epoch over the 1,140-record corpus completes in approximately ten wall-clock hours under this configuration. The single-epoch budget was chosen because additional epochs in pilot runs increased the risk of memorising supervisor-style register without improving span-anchored generation, and the held-out judge benchmark confirmed that further optimisation on this corpus did not change the dominant strategy-level finding. The LoRA rank ($r = 16$), scaling ($\alpha = 32$), and learning rate (5×10^{-5}) follow common defaults reported in the Hugging Face TRL examples for 4-bit-quantised LoRA fine-tuning of 7B–70B instruction-tuned models, and were retained because pilot runs on the development split did not show a clear gain from a sweep within the typical ranges (rank $\in \{8, 16, 32\}$, $\alpha \in \{16, 32, 64\}$). Dropout was held at zero because LoRA at this rank already constrains the adaptation space; adding dropout under-fit on the development split.

C Example of Span-Anchored Feedback Generation

D Detailed Blind-Study Scores

E Detailed LLM-Judge Tables

F Assets, Licenses, Compute, and LLM Usage

Code and data availability. Anonymised supervisor-annotated data, fine-tuning splits, and analysis artefacts are released on OSF at `https://osf.io/d7fra/` (anonymous view-only link: `https://osf.io/d7fra/?view_only=3c84e234d6644ad2820fa8e8139818ee`). The fine-tuning and evaluation pipeline is available at `https://gitlab.ewi.tudelft.nl/dsait5000/tom-viering/msc-thesis-akuruvilla/thesis-feedback-ft`, and the supervisor-facing PDF review interface at `https://github.com/Ayushkuruvilla/thesis_interface`.

Open-weight backbones. The fine-tuned generators use Llama 3.3 70B Instruct [49] under the Llama 3.3 Community License Agreement and Qwen 3.5 27B Instruct [50] under the Apache 2.0 license. Both checkpoints were obtained from the official Hugging Face repositories with checkpoint identifiers and SHA digests recorded alongside the LoRA adapter manifests. Use of both models in this work is consistent with the licenses' permitted academic-research uses, and no model weights are redistributed by this paper.

Judge models. The LLM-as-judge benchmark uses three locally hosted Ollama models: Mistral

Anchor span s
important

Section context c
In machine learning, learning curves are an important metric that plots performance versus training set size. They inform decisions about data acquisition, model selection, and hyperparameter tuning. Despite their importance, recent research suggests that our understanding of learning curve behavior remains limited. In this work, we investigate learning curves from a classification perspective to better understand their structural properties.

Course-guideline prompt p
Provide one concise feedback comment for the highlighted passage. Focus on whether the wording is precise, justified, and useful for revision. Ground the feedback in the highlighted span and surrounding passage.

Feedback output y
Subjective, everything can be important. Instead: explain why you think its important.

Figure 8: Example of span-anchored thesis-feedback generation, adapted from an annotated thesis passage. This is the prompt format used in fine-tuning. The model receives a highlighted span, surrounding section context, and a standardized course-guideline prompt, and produces one concise revision-oriented comment grounded in the selected passage.

Table 3: Participant-weighted blind-study scores on a $[-2, 2]$ scale. Each of the 22 completed raters scored six human comments and nine LLM comments across the held-out study documents. Confidence intervals are bootstrap intervals over participant means. Wilcoxon tests are exact paired tests over participant-level LLM–human contrasts.

Dimension	Human mean [95% CI]	LLM mean [95% CI]	Δ LLM–human [95% CI]	Wilcoxon p
Relevance	1.14 [0.89, 1.37]	1.39 [1.18, 1.59]	+0.25 [-0.01, 0.53]	.117
Change clarity	0.90 [0.62, 1.17]	1.30 [1.11, 1.49]	+0.40 [0.11, 0.71]	.027
Correctness	0.91 [0.63, 1.18]	0.87 [0.60, 1.13]	-0.04 [-0.33, 0.28]	.516
Actionability	0.67 [0.38, 0.93]	0.85 [0.59, 1.12]	+0.19 [-0.14, 0.51]	.429
Specificity	0.83 [0.48, 1.15]	1.30 [1.10, 1.50]	+0.47 [0.15, 0.81]	.018
Supervisor suitability	0.98 [0.67, 1.26]	1.15 [0.87, 1.40]	+0.17 [-0.19, 0.56]	.500

Small (Apache 2.0), DeepSeek-R1 32B Qwen Distill (the DeepSeek-R1 license; the Qwen 2.5 base from which it was distilled is Apache 2.0), and Gemma 4 31B (the Gemma Terms of Use). All three are used only for inference inside the institutional cluster and are not redistributed.

Other assets. LoRA fine-tuning uses `peft` and `trl` (Apache 2.0). PDF parsing uses PyMuPDF (AGPL-3.0) and Docling (MIT). The interface uses React (MIT), FastAPI (MIT), and LiteLLM (MIT). Statistical analysis uses NumPy, SciPy, and scikit-learn (all BSD-3-Clause).

Compute envelope. Each LoRA fine-tune runs on one NVIDIA A100 80 GB on the institutional DelftBlue cluster and completes one epoch over the 1,140-record corpus in approximately ten wall-clock hours; two reported fine-tunes (Llama 3.3 70B, Qwen 3.5 27B) therefore account for approximately 20 GPU-hours. Inference for the deployed section-aware route processes one thesis in a few minutes on the same hardware. The 12-system LLM-as-judge benchmark over five held-out theses yielded 1,491 successful item-level judgments across the three judges and two judge runs, with each judge call costing single-digit seconds on the same partition; the total benchmark consumed on the order of 40–60 GPU-hours. Pilot runs that did not enter the reported results (alternative backbones evaluated and discarded, a single-call section pipeline variant, and decoding-temperature pilots) add an estimated further 60–80 GPU-hours, so the full research project consumed roughly 150 ± 50 A100 80 GB GPU-hours.

Declaration of LLM usage. LLMs are a core, non-standard component of the methods reported here: Llama 3.3 70B Instruct and Qwen 3.5 27B are the systems under study; Mistral Small, DeepSeek-R1 32B Qwen Distill, and Gemma 4 31B are used as evaluation judges. All model identities, fine-tuning settings, and prompt-design details are reported in Sections 3 and 5 and Appendices A and B.

Separately, in line with TU Delft's publishing policies,[1] the author acknowledges the use of AI tools, including ChatGPT, to support the writing

[1] TU Delft publishing policies: https://www.tudelft.nl/library/actuele-themas/open-publishing/about/policies

Table 4: Per-judge approach ranking by average score across the six rubric dimensions. The ranking is stable even though the judges differ in absolute severity.

Judge	Approach	Rank	Overall mean
DeepSeek-R1 32B Qwen Distill	Whole-document generation	1	1.56
DeepSeek-R1 32B Qwen Distill	Agentic review	2	1.49
DeepSeek-R1 32B Qwen Distill	Section-aware two-stage	3	1.41
Gemma 4 31B	Whole-document generation	1	1.67
Gemma 4 31B	Agentic review	2	1.50
Gemma 4 31B	Section-aware two-stage	3	1.48
Mistral Small	Whole-document generation	1	1.20
Mistral Small	Agentic review	2	1.14
Mistral Small	Section-aware two-stage	3	1.10

and preparation of this work. These tools were used for language refinement, drafting assistance, LaTeX editing support, and code-related assistance during experimentation and analysis. AI tools were used as assistive tools only; the research design, implementation choices, experiments, interpretation of results, and final submitted text remain the author's own responsibility. All AI-assisted text and code suggestions were reviewed, edited, and checked against the implemented experiments and recorded results before inclusion, and no experimental result, table value, citation, or conclusion was accepted solely on the basis of AI-generated output.

Supplemental Materials 3

This chapter provides additional detail on the work presented in the scientific paper. It opens with an extended literature survey across six neighbouring strands (Section 3.1) and the governance constraints that shape the system design (Section 3.2). The remaining sections document the dataset construction pipeline (Section 3.3), the implementation of the models, inference strategies, and supervisor-facing interface (Section 3.4), the pilot evaluation that shaped the deployed configuration (Section 3.5), the two user studies and their human-research-ethics procedure (Section 3.6), an extended analysis of the LLM-as-a-judge benchmark (Section 3.7), and a final discussion of how the work relates to the surrounding research and deployment context (Section 3.8).

3.1. Related Work

3.1.1. Educational Feedback and Writing Pedagogy

Research on educational feedback characterises useful feedback in terms of its effect on learning. Hattie and Timperley [2] describe feedback as information that reduces the gap between current performance and a desired standard, and identify three useful question-types for the learner: where am I going, how am I going, and where to next. Shute [3] reviews formative-feedback design and identifies properties associated with learning gains: feedback that is specific to the task, that is timed appropriately, and that supports rather than threatens the learner. Nicol and Macfarlane-Dick [4] place feedback inside self-regulated learning and argue that feedback should help students develop their own internal capacity to self-evaluate. The dimensions used in the evaluation rubric for this work (relevance, change clarity, correctness, actionability, specificity, and supervisor suitability) trace back to this literature: each captures an aspect of what makes feedback usable for revision.

Thesis feedback specifically has been studied in the doctoral and master's setting. Kumar and Stracke [6] document supervisor comments across content, organisation, and disciplinary content, with substantial variation in what supervisors choose to emphasise. Bitchener, Basturkmen, and East [5] analyse written supervisor feedback on doctoral writing and report variation in language, uncertainty, and scholarly standards. Carter and Kumar [7] situate feedback inside the broader supervisory relationship. The shared finding is that several different comments can be useful for the same passage and that exact agreement with a single gold-standard comment is a poor evaluation target. The blind feedback-quality study described in Section 3.6.1 is designed against this background: human comments are evaluated as one possible response to a passage rather than as the canonical answer.

3.1.2. AI-Generated Feedback for Educational Writing

A growing body of work compares LLM-generated feedback against peer, instructor, or human-authored feedback. Dai et al. [20] compare ChatGPT against human teachers on student assignments and find that the model's feedback is rated highly on readability and detail but is less personal. Escalante, Pack, and Barrett [22] study AI versus human tutors in a writing classroom and find that students improve under both, with no significant difference in student preference. Work on essay feedback in higher education [24] and on university writing more broadly [25] reports that AI-generated feedback is consistently strong on local correction and weaker on the kind of guidance that requires understanding the student's broader argument.

A second strand evaluates the alignment of model feedback with rubric criteria. Seßler et al. [23] study adaptive AI feedback and decompose feedback quality into multiple dimensions. Banihashem et al. [26] compare AI to peer feedback sources and emphasise multidimensional quality. Story- and essay-feedback studies likewise separate correctness, specificity, and revision orientation [21, 27]. The shared finding

across this literature is that fluency is not a sufficient evaluation criterion and that useful evaluation needs to disaggregate quality into properties such as accuracy, specificity, and revision-orientation. The same disaggregated approach is adopted here.

3.1.3. Scholarly and Academic Review with LLMs

Scholarly review provides an adjacent literature for critique generation on long technical documents. Kang et al. [52] introduced PeerRead as a benchmark for review analysis and prediction; the benchmark scopes evaluation to acceptance prediction and aspect scoring rather than open-ended comment generation, which limits its direct transfer to a setting where the comment text itself is the artefact of interest. Liu and Shah [29] extend the line to LLM-generated paper reviews and find that the model is competitive on simple verification tasks but loses fidelity on tasks that require global understanding of the paper. Zhou, Chen, and Yu [30] examine the reliability of model-generated reviews and report that single-judge reliability is insufficient for deployment claims, motivating multi-judge or human-calibrated evaluation. Afzal et al. [31] introduce literature-aware critique that conditions the model on retrieved related work; the design improves grounding when the relevant context is external to the document, but transfers less directly to a single-thesis setting where most of the relevant context is internal. Sun et al. [32] examine LLM-supported review of master's theses and report local-strength patterns similar to those reproduced here, but evaluate on a smaller pool and without a workflow study. Liang et al. [28] compare GPT-4-generated reviews to human reviews on Nature-family and ICLR papers and report that the model's points overlap only partially with human reviewers, while more than half of surveyed authors rated the AI feedback as helpful. Their finding that useful critique varies even across human reviewers supports an evaluation strategy based on utility rather than on agreement with a single ground-truth reviewer.

Thesis supervision differs from scholarly review in audience and purpose. A reviewer's comments inform an editorial decision; a supervisor's comments help the student revise toward a graded final draft. Supervisor feedback is also part of an ongoing relationship rather than a one-shot review. The interface described in Section 3.4.4 is designed against this difference: it shows LLM comments for inspection and revision.

3.1.4. Long-Context Generation and Retrieval-Augmented Strategies

Generating useful comments for a long thesis depends on how the model is exposed to the document. Long-context decoder-only models can in principle consume tens of thousands of tokens in one prompt, but their ability to use that context degrades with position and length. Liu et al. [43] demonstrated the lost-in-the-middle phenomenon: model performance on retrieval-style tasks drops substantially when the relevant evidence is in the middle of a long input. The result is a structural property of decoder-only attention rather than a defect of any particular model, and it transfers to thesis feedback: a model given the whole thesis as a single prompt is more likely to surface comments anchored at the start or end of the document than in the middle. Hsieh et al. [44] introduced the RULER benchmark and reported that the effective length of long-context models, defined as the longest input on which the model maintains a chosen accuracy threshold, is consistently shorter than the advertised window. The implication for whole-document prompting on a long thesis is that the strategy may approach the model's effective usable context length even when the input fits in the nominal context.

Retrieval-augmented generation [45] treats context selection as a separate step that chooses passages to include in the model's prompt. The framework was developed for open-domain settings where the relevant context is spread across a large external corpus; its limitation in the present setting is that the relevant context is internal to the document, so dense retrieval contributes less than in the open-domain case. Li et al. [46] compare retrieval against long-context prompting on long-document QA and reasoning, and report a contingent answer: retrieval helps when the answer depends on a small, locally identifiable region; long-context prompting helps when the answer depends on aggregating distributed evidence. Long-context retrieval extensions [53] push the boundary on input length but do not directly address the lost-in-middle bias. Self-reflective retrieval [47] introduces a generation-time decision about whether to retrieve, which is more expressive but adds a second source of error that has to be controlled. Thesis feedback combines both regimes: a comment about an unsupported claim depends on a local span, while a comment about whether a contribution is consistent with the methodology depends on distributed evidence. The strategy comparison in the scientific paper tests

three points along this trade-off rather than committing to a single regime.

3.1.5. Span-Anchored Annotation and Grounded Critique

Annotation systems that attach comments to spans of text have a long history in document-review tooling. The Web Annotation Data Model formalises annotations as relationships between a body and a target, with selectors that identify a text segment in the source [38]. In educational and review settings, span anchoring has been associated with better learner uptake. Nelson and Schunn [36] report that comments which include problem location, solutions, and summaries are associated with revision. Gielen et al. [37] report that justified peer comments are linked to better learning outcomes. Span anchoring constrains generation to visible evidence, which reduces inspection effort and gives raters a concrete object to score. The same rationale motivates the span-anchored task formulation used here.

3.1.6. Human-in-the-Loop Writing and Feedback Systems

Recent systems work studies how generated feedback enters and is used in writer workflows. Zhang et al. [54] show how visual organisation can help writers interpret and synthesise feedback; the setting is short-form writing rather than long-document supervision, so the visualisation patterns transfer only partially to a thesis-length artefact. Sölch, Dietrich, and Krusche [55] study how automated feedback enters submission workflows but treat the model output as a fixed artefact, omitting the editing layer that supervisor-controlled deployment requires. Zyska et al. [56] treat peer feedback and system support as an interaction rather than a standalone prediction problem, which aligns with the present design but is evaluated on essay-length material rather than on long technical writing. The shared insight across this literature is that the practical value of generated feedback depends on whether comments can be inspected, edited, accepted, rejected, and exported within a real review workflow. The supervisor-facing interface described in Section 3.4.4 is built on the same insight and adds an explicit editing layer for the long-document supervision setting: it surfaces generated comments as editable PDF annotations rather than as a fixed output.

3.2. Educational AI Governance

3.2.1. Operational Implications

Three sources of guidance constrain a system that processes student-derived material in an educational setting. The UNESCO guidance on generative AI in education [39] emphasises privacy, validation by qualified educators, transparency to learners, and institutional governance of model adaptation. The European AI Act [40] and its implementation guidance [57] treat AI used in educational assessment as a high-risk category and impose obligations around risk management, data governance, transparency, automatic logging, and human oversight. The General Data Protection Regulation [41] applies to thesis drafts and supervisor comments because both contain personal data, and its principles of data minimisation and privacy by design [58] constrain what the dataset may contain and how the system may log its activity.

Together, the three frames point toward the same operational picture for a deployment in a course like CSE3000. Processing must remain on infrastructure the institution governs. Model adaptation must use anonymised data. The supervisor must remain the final author of any feedback that reaches a student. The workflow must keep AI-authored and supervisor-authored annotations distinguishable. The choice of open-weight models and the choice to host both fine-tuning and inference on the institutional DelftBlue cluster follow directly from these requirements. So does the interface design pattern in which generated comments enter as editable suggestions rather than as fixed outputs. So, finally, does the residual-risk taxonomy reported in the discussion of the scientific paper.

3.3. Dataset Construction

3.3.1. Annotation Sources

Supervisor feedback in the source corpus arrived in five forms.

- *Highlight with comment.* A PDF highlight annotation whose quadrilateral vertices delineate a text rectangle, with a comment payload attached. The geometric shape indicates what the comment is about.
- *Sticky note.* A PDF text annotation placed at a single page coordinate without an explicit text target. The reader infers the target from the placement, but the inference is unreliable in two-column layouts and on pages with figures.
- *Free-form marking.* A non-text drawing annotation (rectangle, polygon, or ink) around a passage. Visually unambiguous but geometrically ambiguous: the bounding box of a freehand circle around a paragraph rarely coincides with the paragraph's text rectangle.
- *Text-file feedback.* A separate plain-text file accompanying a draft, containing supervisor comments without span anchors. A subset contained quoted excerpts that could be aligned by exact-string match.

Highlights and the quoted parts of text-file feedback are recoverable automatically. Sticky notes, free-form markings, page-level notes, and the unquoted parts of text-file feedback require manual alignment to recover the intended target.

3.3.2. Extraction and Alignment Pipeline

The pipeline runs in two stages. The first extracts the thesis main text into a structured representation that preserves section heading hierarchy, paragraph boundaries, and placeholders for tables and figures. Heading detection uses font-size thresholds, bold-weight detection, and numbered-heading regular expressions. Paragraph reflow merges line-broken sentences within a paragraph and inserts explicit paragraph breaks between paragraphs. Tables and figures are replaced by explicit placeholders rather than extracted as text, because their textual rendering varies substantially between PDF producers and supervisor comments rarely refer to their content directly.

The second stage links each annotation to a span in the structured representation. Highlights are aligned automatically with PyMuPDF: for each highlight, the extractor iterates over the annotation's quadrilateral list, calls the page text-recovery routine on each rectangle, and joins the recovered fragments into the anchor span. Three normalisation steps handle predictable failure modes. Hyphenated line breaks are merged with the regular expression `(\w)-\n(\w)` so that an anchor spanning a hyphenated word matches the same anchor when the PDF is re-rendered in the interface. Unicode normalisation collapses curly versus straight quotation marks, en-dash versus hyphen, and non-breaking versus regular spaces to canonical forms; the same normalisation is applied at the interface side so that the two ends of the pipeline produce identical anchor strings. Whitespace is collapsed to single spaces and trimmed at the edges.

For sticky notes, free-form markings, and the unquoted parts of text-file feedback, alignment is manual. A reviewer opened each unanchored annotation alongside the corresponding extracted section text and recorded the intended target span by exact-quote selection. The reviewer was instructed to prefer the smallest anchor span that captures the passage the comment is plausibly about, to choose the span that the comment text references most directly when the placement is ambiguous, and to discard the annotation entirely when neither rule resolves the ambiguity. Discards are recorded with a reason code so that the audit pass can verify consistent application of the rule. The manual alignment was performed by a single reviewer, which is a documented limitation of the corpus: an inter-rater pass over a sample of borderline alignments would produce inter-reviewer agreement statistics that the current pipeline cannot, and is recorded as a follow-up item rather than a current property of the data.

3.3.3. Anonymisation

Bachelor thesis drafts and supervisor comments contain personally identifiable information. The anonymisation protocol has three components, all applied before any model training or evaluation.

Cover-page exclusion. Front pages contain student names, registration numbers, supervisor names, and submission dates. Front pages were either deleted from the working PDF or blacked out with a full-page rectangle annotation that the extractor treats as empty; the choice between deletion and redaction is recorded per document.

In-text name detection. An NLP-based name-detection pass was run over the extracted thesis text and over the supervisor comments. Detected names were replaced with role-stable placeholders (`[STUDENT]`, `[SUPERVISOR]`, `[THIRD_PARTY]`) so that the surrounding sentence remained syntactically coherent. The same placeholder vocabulary is used in the prompts so that the model does not learn to treat placeholders as anomalies.

Comment-specific scrubbing. Comments that became uninterpretable after scrubbing, because they were primarily addressed to the student by name, were dropped rather than retained as fragmentary placeholders.

The protocol does not promise re-identification resistance against an adversary with access to TU Delft cohort records. It promises that the dataset, used within institutional boundaries, contains no direct identifiers and no obviously identifying contextual phrases.

3.3.4. Statistics and Audit

Two audits were carried out. The first was a rule-based pass over an early 846-record snapshot used to surface preprocessing issues. The second was a refined pass over the final 1,140-record corpus used for fine-tuning. Tables 3.1 and 3.2 summarise both audits.

Table 3.1: Comparison of the two dataset audits.

Statistic	Initial audit	Refined audit
Total records	846	1,140
Records with section context	839	1,138
Mean feedback length (words)	18.84	19.10
Median feedback length (words)	13	13
Feedback with < 5 words	147	136
Question-heavy feedback	266	329

Table 3.2: Top feedback categories in the refined audit. Categories are produced by a rule-based classifier intended for distributional summary.

Category	Count	Share
other	551	48.3%
question_probe	164	14.4%
positive_accept	101	8.9%
logic_interpretation	95	8.3%
revision_instruction	75	6.6%
methodology	68	6.0%
evidence_citation	50	4.4%
english_or_formatting	28	2.5%
structure_organization	8	0.7%

The most consequential property of the refined distribution is the dominance of short clarification-style and probing-style comments. Long methodological critiques are scarce. A model trained on this corpus will learn to produce comments of the same shape, and the qualitative finding that the deployed system produces locally specific comments with limited explicit rationale is partly a consequence of the training distribution rather than of the model itself.

This matters for interpreting fine-tuning. Instruction-tuning studies show that small supervised datasets can strongly teach response format and style when the examples are carefully curated, while low-quality or mismatched examples can also reduce downstream performance [59, 60]. The present corpus is high value because it is authentic supervisor feedback, but it is not a balanced sample of all supervisory judgment. It over-represents short written annotations that were recoverable from PDFs and under-represents longer oral, meeting-based, or document-level critique. Fine-tuning on this data should therefore be expected to teach the model the local span-comment genre more reliably than it teaches higher-level thesis review.

Five thesis documents were reserved as the held-out evaluation pool before any filtering or training run. The selected identifiers are preserved in the analysis artefacts so that any re-run produces directly comparable results. The pool was selected by drawing five identifiers from the document index uniformly at random under a fixed seed.

3.4. Implementation

3.4.1. Models and Fine-Tuning

Two open-weight backbones are used. Llama 3.3 70B Instruct is the deployed model in the supervisor-facing interface and in the blind feedback-quality study. Qwen 3.5 27B Instruct is the principal alternative backbone in the LLM-as-a-judge benchmark. Both are obtained from their official Hugging Face checkpoints, with the exact checkpoint names and SHA digests recorded alongside the LoRA adapter manifests so that any later resolution to a different revision can be detected.

Adapters are produced with LoRA over Unsloth's 4-bit-quantised loader and TRL's `SFTTrainer`. The adapter configuration is shared between the Llama and Qwen routes; only the chat-template selector differs. Adapters are applied to the attention $q/k/v/o$ projections, the feed-forward gate/up/down projections, the input embeddings, and the language-model head. The loss is computed on the assistant response only, with the chat template applied with `add_generation_prompt=False` so that the assistant message is preserved as the supervision target. Table 3.3 reports the full hyperparameter set.

Table 3.3: Supervised fine-tuning configuration. The Qwen route uses identical values; only the chat-template selector differs.

Hyperparameter	Value
Quantisation	4-bit (Unsloth NF4-style)
Compute dtype	`bfloat16` on compute capability ≥ 8.0, `float16` otherwise
LoRA rank r	16
LoRA α	32
LoRA dropout	0.0
Loss target	assistant response only
Max sequence length	8,192 tokens
Per-device train batch size	1
Gradient-accumulation steps	32 (effective batch size 32)
Learning rate	5×10^{-5}
Number of epochs	1.0
Optimiser	AdamW (TRL default)
Save interval	every 5 optimiser steps (resume-aware)
Random seed	42
Hardware	DelftBlue GPU partition

The single-epoch budget reflects two considerations. The supervised pool is small enough that additional epochs increased the risk of memorising supervisor-style register without improving span-anchored generation, and the held-out judge benchmark confirmed that further optimisation on this corpus would not change the dominant strategy-level finding. Dropout is held at zero because LoRA at this rank already constrains the adaptation space; adding dropout under-fit on the development split.

Compute envelope. Each fine-tune was executed on the DelftBlue GPU partition on a single NVIDIA A100 80 GB. One epoch over the 1,140-record corpus completed in ten wall-clock hours under the configuration in Table 3.3, with the resume-aware checkpointing every five optimiser steps adding negligible overhead while allowing recovery from node failures without re-running the entire epoch. Inference for the deployed section-aware route processes a single thesis in a few minutes on the same hardware once the backbone is resident.

3.4.2. Inference and Decoding

All three generation strategies share a deterministic decoding configuration: temperature 0, top-p 0.8, top-k 20, repetition penalty 1.05, and the Ollama JSON-format constraint. The context window

requested at inference is 16,384 tokens. Per-call `num_predict` budgets are 120 tokens for the selector and the generator stages in the section-aware route, larger for whole-document and agentic calls. The HTTP timeout is 1,800 seconds for the section-aware runner and 600 seconds for the docling-based runner and the agentic pipeline. Ollama's `keep_alive` value is set to ten minutes, which keeps the model resident across consecutive calls so that per-call overhead is dominated by token generation rather than by model loading.

These settings affect reproducibility. Lowering the context window forces aggressive chunking and changes the per-window evidence the model sees. Raising the temperature increases the rate of anchor mismatches, since deterministic decoding is what allows the post-filter to enforce the exact-quote rule reliably. Reducing the timeout makes long-context calls fail at irregular boundaries inside a thesis.

3.4.3. Generation Strategies

Strategy axis. The three generation strategies test different ways of exposing a long thesis to the model. A bachelor thesis in the CSE3000 corpus is between fifteen and forty single-column pages plus figures, tables, and references; after parsing, the longest held-out drafts produce prompts in the tens of thousands of tokens. The long-context and context-selection literature reviewed in Section 3.1.4 therefore leads to a practical design question: should feedback be generated from the whole document, from selected local evidence, or from a structured multi-step review process? Whole-document generation tests direct long-context prompting. Section-aware two-stage generation narrows the evidence to a local section window and explicit anchor. Agentic review adds role-specialised critique and verification on top of section-level evidence. The deployed supervisor-facing route is the section-aware two-stage pipeline because it produces exact anchor spans that can be re-attached to the PDF and returns one local suggestion per window, matching the interface's accept, edit, and dismiss operations.

Whole-document generation. The PDF is converted with a Docling-based pipeline (`section_and_full_review_docling.py`) into an XML-tagged representation that preserves section and subsection boundaries. Cleaned text is passed verbatim to a single chat completion under a constrained JSON schema. This strategy gives the model maximal access to thesis-level context and is therefore best suited to comments about contribution, methodology, and consistency across sections. Its risk is the one identified in long-context studies: fitting the input inside the nominal context window does not guarantee that the model will use the relevant middle sections reliably. A post-filter removes items whose anchor sentence is not an exact substring of the excerpt, items whose anchor matches a placeholder pattern, and items that duplicate an already-accepted entry.

Section-aware two-stage generation. The PDF is converted to a structured text stream with heading detection and paragraph reflow (`section_review_pdf_dir_two_stage.py`). The result is split into top-level sections; each section is partitioned into sliding windows of at most W characters with O characters of overlap. For each window, a Stage 1 selector returns at most one exact-quoted candidate sentence; the candidate is validated for exact occurrence in the section text, against a truncation heuristic, and against a global deduplication set. Stage 2 then issues a generator call for that candidate using the entire section text plus the selected sentence. This separation avoids asking one prompt to both decide where feedback is needed and write the feedback, and it makes the anchor sentence an explicit intermediate artefact. A post-filter rejects comments that are too short, too vague, or match a list of generic feedback prefixes.

Agentic review. The agentic pipeline tests whether decomposition and verification improve coverage and actionability compared with a single prompt. The design follows multi-agent LLM work in which a task is split into specialised roles with coordination, debate, or verification steps [61, 62, 63, 64]; it is also related to critique-specific work that trains or organises models around feedback from multiple critics [65]. In this implementation, the `agentic_review` module uses one backbone for all stages and decomposes review into five role-specific reviewers: clarity, structure, methodology, results, and consistency. A deterministic coordinator scores sections by heading-keyword priority and pending-role coverage, then dispatches at most B tasks per iteration. Each reviewer proposes a candidate span and comment for its assigned role. The candidate first passes deterministic checks for exact span match, placeholder avoidance, and duplicate anchors. It then passes through a semantic verifier that returns

ACCEPT, REFINE, or REJECT; a refine verdict triggers up to $R_{\max}$ refinement attempts, while a reject verdict is recorded as failure-mode evidence. A final deduplicator removes near-identical accepted comments. The loop terminates when the agenda is empty, when a patience counter elapses without new accepts, or when the iteration budget is reached. In the benchmark, this strategy is expected to trade additional inference cost for broader issue coverage and stronger actionability, because role prompts can direct model behaviour toward methodology, results, and consistency issues that a generic prompt may not prioritise.

Methodological contribution of the strategy comparison. Whole-document, section-aware, and agentic strategies are usually studied in separate papers and on different tasks. Evaluating all three within the same backbone family, prompt format, held-out pool, and calibrated multi-judge benchmark isolates the strategy axis more cleanly than a comparison assembled from independent baselines. If each strategy had used a different backbone family, the result would have confounded generation strategy with model capability. If the benchmark had relied on a single judge, the ranking would have inherited the idiosyncrasies of one evaluator. The contribution of this comparison is therefore not that any individual component is new in isolation, but that the components are held fixed tightly enough for the strategy-level result to be interpretable.

3.4.4. Supervisor-Facing Interface

The interface is a React/TypeScript frontend with a FastAPI backend. LiteLLM provides a model-routing layer so that the same frontend can call either institutionally hosted open-weight endpoints or a closed-source API behind the same registry. The frontend separates document handling, manual annotations, AI-generated feedback, and study-specific rating flows so that the same substrate supports both supervisor use and the embedded blind study.

The design follows a mixed-initiative pattern rather than a fully automated review pattern. Mixed-initiative interface work argues that automated services should be easy to invoke, easy to terminate, and easy for users to refine when the system's inference is uncertain [66]. Human-AI interaction guidelines similarly emphasise showing system status, supporting efficient correction, and making clear when users remain responsible for the outcome [67]; human-centred AI work frames reliable systems as ones that preserve meaningful human control rather than replacing it with opaque autonomy [68]. These principles map directly onto the interface: generated comments are not inserted into the final PDF automatically, but enter as draft suggestions that can be inspected against the anchor span, edited, accepted, rejected, and exported only after supervisor action.

The supervisor workflow has four operations.

1. *Upload and view a thesis PDF.* The PDF is rendered in the browser with the original annotations preserved.
2. *Request generated suggestions.* The frontend extracts sections from the rendered PDF, sends the selected section context to the backend model route, and receives generated comments as draft annotations.
3. *Edit, accept, or dismiss annotations.* The interface presents AI suggestions as a list beside the document. Each comment can be inspected against its anchor span on the rendered PDF, edited in place, accepted into the supervisor's own annotation layer, or dismissed.
4. *Export the reviewed PDF.* The export contains the supervisor's accepted and edited comments.

Screenshots of the interface are present in Appendix C. The study backend stores participant responses under an opaque participant identifier and keeps the rating flow separate from model inference. The schema, environment-driven storage configuration, and export scripts are committed in the repository.

Code availability. Both repositories are available for inspection and reuse. The model-side code (data conversion, fine-tuning, inference pipelines for the three generation strategies, and the LLM-as-a-judge runner) is hosted at `https://gitlab.ewi.tudelft.nl/dsait5000/tom-viering/msc-thesis-akuruvilla/thesis-feedback-ft`. The supervisor-facing interface (React frontend, FastAPI backend, study flow, and PDF annotation handling) is hosted at `https://github.com/Ayushkuruvilla/thesis_interface`. Each repository contains a README that maps the experiments

reported in the scientific paper to specific scripts, the prompts in their source form, and the configuration files used for the runs.

3.5. Pilot Evaluation and Iteration

The deployed configuration is the result of a sequence of pilot runs that shaped both the model side and the interface. Two pilot decisions are worth recording because they constrained the evaluation that followed.

Schema-conformance pilots motivated dropping two backbones. GPT-OSS 20B and Qwen 2.5 70B were prepared for inclusion in the strategy comparison but produced inconsistent behaviour under the structured JSON output schema required for span-anchored generation. The most common failure modes were chat-template inconsistency between checkpoint releases, free-form prefixes preceding the JSON payload, and occasional truncation of the closing brace under tight `num_predict` budgets. Including them would have introduced model-specific reliability dimensions into a benchmark designed to separate strategy from backbone. They were therefore dropped from the central comparison and recorded as exploratory negative results.

Single-call section pipelines were dropped in favour of the two-stage formulation. An earlier section-level pipeline issued one call per window in which the model both selected the anchor sentence and produced the comment in a single response. The output anchor frequently failed an exact-substring check against the section text, which broke the interface's PDF re-attachment. Splitting the pipeline into a separate selector call followed by a separate generator call solved the problem at the cost of one additional call per window. The deployed two-stage pipeline is the result of that decision.

Together, these pilot decisions converge on the deployed configuration: a section-aware two-stage pipeline on a Llama 3.3 70B fine-tune, with the dropped backbones and the dropped single-call section pipeline retained as exploratory negative results that bound what the strategy comparison can be expected to show. Each decision narrowed the experimental scope toward a comparison whose result is interpretable rather than confounded by reliability or anchoring artefacts of the discarded variants.

3.6. User Evaluation

3.6.1. Blind Feedback-Quality Study

Participants were recruited from the population of thesis supervisors, postdoctoral researchers, and PhD candidates at TU Delft and adjacent institutions through email invitation. Recruitment continued until twenty-two participants had completed the full protocol. Participation was voluntary and unpaid; participants could stop at any point without giving a reason.

The study presented each participant with three of the five held-out theses, in randomised order. Each thesis was shown together with five candidate feedback comments: two human comments drawn from the held-out annotations and three AI comments generated by the deployed section-aware Llama route. Source labels were hidden, so participants did not know which comments were human-authored and which were model-generated. For each comment, the interface displayed the anchor span and the surrounding section context; participants could open the underlying PDF to inspect the broader document. Each comment was rated on the six dimensions of the rubric. The first five (relevance, change clarity, correctness, actionability, specificity) used a symmetric five-point agreement scale encoded as $-2, -1, 0, +1, +2$. Supervisor suitability used a four-point ordered scale: not suitable (-2), suitable after major revision (-1), suitable after minor editing ($+1$), suitable as-is ($+2$).

The data export contained twenty-two completed raters and 330 comment-level ratings, including 132 human-comment ratings and 198 AI-comment ratings. No rows were excluded from the quantitative analysis. The full rubric wording shown to participants is reproduced in Appendix A.

3.6.2. Interface Usability Study

The interface usability study used the standard System Usability Scale [1]. Participants completed a task-based interaction with the supervisor-facing pipeline that included PDF inspection, AI-feedback review, editing, and export, then answered the standard ten-item SUS instrument in Qualtrics. The task instructions and the SUS items are reproduced in Appendix A.

The SUS export contained twenty-one consented, finished responses with all ten items completed. The thesis reports these twenty-one complete SUS responses as the headline usability analysis. A task-completion sensitivity check retains the nineteen responses that explicitly marked all tasks complete; this subset does not change the interpretation. SUS is reported on the standard 0–100 scale.

3.6.3. Human Research Ethics

Both studies involve human participants and were therefore designed under the Human Research Ethics policy of TU Delft. An application was submitted to the Human Research Ethics Committee (HREC) before participant recruitment began, and approval was obtained for the protocol described above. The application documented the study design, the rubric, the consent flow, the data-storage location, the data-retention period, and the participant-withdrawal procedure.

The protocol minimises participant risk. No personally identifiable information is collected from participants beyond the consent record, which links a participant identifier to a consent timestamp and is stored separately from the rating data. Participants are informed of the study's purpose at the start of the consent flow and are given the option to withdraw at any time without giving a reason. Withdrawals delete all data associated with the participant identifier. All study data are stored on TU Delft infrastructure; no participant data are transferred to external services.

The held-out thesis material shown to participants was anonymised before the study under the protocol described in Section 3.3.3. Original visible annotations were removed from the PDFs shown in the blind study so that participants would not infer the source of a comment from the presence of an existing supervisor annotation.

3.7. Extended Analysis of the LLM-as-a-Judge Benchmark

3.7.1. Per-Judge Agreement

The three judges (Mistral Small, DeepSeek-R1 32B Qwen Distill, and Gemma 4 31B) place the strategies in the same order despite differing in absolute severity. Mistral Small is the strictest. Across the 1,491 successful item-level judgments, its grand mean of 1.15 across the six rubric dimensions lies 0.42 and 0.47 points below the means produced by DeepSeek-R1 (1.49) and Gemma 4 (1.55) respectively. All three judges nonetheless rank whole-document prompting first, agentic review second, and section-aware two-stage generation third on overall macro score, and the rank does not flip on any rubric dimension. The judge-macro means used for strategy comparison reflect a stable ordering rather than an artefact of one generous evaluator.

A second reliability diagnostic concerns failure rates inside the judge run. Across the two judge runs, twelve item-level judgments failed schema validation and were excluded from score means but retained as coverage failures. All twelve failures were produced by Gemma 4 31B; none came from Mistral Small or DeepSeek-R1.

3.7.2. Base versus Fine-Tuned Per-System Effects

Aggregating systems into three strategy families is appropriate for an overall comparison but hides a pattern that emerges when base and fine-tuned variants are separated within each backbone. Table 3.4 reports paired base–finetuned macro scores per backbone and strategy.

Three observations follow. First, fine-tuning on the span-anchored corpus does not uniformly improve automatic scores: in five of the six paired comparisons the fine-tuned variant scores lower than its base. Second, the family ordering is preserved across base and fine-tuned tiers (Qwen 3.5 scores above Llama 3.3 in every paired comparison), suggesting that backbone capability dominates the effect of supervised adaptation on this benchmark. Third, the deployed section-aware Llama route used in the blind study is

Table 3.4: Per-backbone judge-macro overall scores for base and fine-tuned variants, with Δ = finetuned − base on the [−2, +2] scale.

Strategy	Family	Base	Fine-tuned	Δ
Whole-document	Llama 3.3	1.42	1.17	−0.25
Whole-document	Qwen 3.5	1.63	1.59	−0.03
Section-aware two-stage	Llama 3.3	1.39	1.27	−0.12
Section-aware two-stage	Qwen 3.5	1.52	1.28	−0.25
Agentic review	Llama 3.3	1.23	1.11	−0.12
Agentic review	Qwen 3.5	1.45	1.58	+0.13

therefore not the highest-scoring system on the judge benchmark; it was deployed because it produces inspectable spans, returns comments under the JSON schema required for interface integration, and is the route on which the workflow study was actually conducted.

The data audit gives a plausible explanation for the non-uniform fine-tuning effect. The base models already contain broad academic-writing and reasoning capabilities from pretraining and instruction tuning. The local fine-tuning corpus mainly teaches a narrower output distribution: short span-anchored comments, many of them phrased as clarification questions or concise revision prompts. This can improve interface compatibility and stylistic similarity to the source annotations while reducing the model's tendency to produce broader critique. The base–fine-tuned result should therefore be read as a data-fit result, not as evidence that supervised fine-tuning is intrinsically unhelpful for thesis feedback.

3.7.3. Error Patterns in the Deployed Route

The qualitative comments from the blind study converge on three recurring failure modes for the deployed section-aware Llama system, all consistent with the dimension-level numbers.

Local-but-shallow critique. Comments correctly identify a surface problem in the highlighted span without explaining why the change matters for the thesis argument. This is consistent with the gap between high specificity (+0.47) and weaker correctness (−0.04) in the same data.

Missed global concern. Span anchoring concentrates the model's evidence on the highlighted excerpt and its immediate section context, which makes the system structurally less able to comment on contribution clarity, methodological coherence, or relationships between sections.

Near-duplicate feedback across sections. The two-stage pipeline deduplicates within a single thesis using exact anchor matching, but it cannot detect that two distinct anchor sentences in different sections raise the same conceptual issue.

A fourth pattern is less a failure than a property of the dataset. Sixty per cent of the supervised target comments are shorter than twenty words; only 6.6% are at least fifty words. The model has learned to produce concise rather than explanatory comments. Together with the missing rationale dimension surfaced by raters, this suggests a concrete data-improvement target: training examples that include the why component alongside the actionable suggestion may reduce local-but-shallow feedback.

3.8. Reflections and Broader Implications

3.8.1. Position in the Broader Field

This thesis tests candidate comments, rather than final feedback decisions, as the unit of automation. The model produces a span-anchored draft comment that is tied to visible evidence in the document. The supervisor then inspects, edits, accepts, or rejects that comment before export. This positions the contribution less as a claim that LLMs can "review theses" and more as a workflow pattern for making generated critique reviewable by a qualified human reader.

The results also clarify a tension that matters beyond this specific course setting. Span anchoring

supports grounding, specificity, and inspection, but it can bias the system toward local comments and away from contribution-level, methodological, or cross-section critique. Whole-document and agentic strategies address the same problem from different directions, yet the benchmark shows that broader context and stronger orchestration do not remove the need for human judgment. Grounding and global assessment are therefore complementary requirements rather than a single solved capability.

The evaluation design reflects that position. The LLM-as-a-judge benchmark is useful for comparing many system configurations under the same rubric, but the blind human study and judge–human calibration are needed to test whether automatic scores track human acceptability. The largest calibration gap is supervisor suitability, the dimension most directly tied to deployment. For this reason, the thesis treats automatic judges as scalable diagnostics and human ratings as the stronger evidence for whether generated comments belong in a supervisory workflow.

3.8.2. Implications for Stakeholders

The workflow affects several distinct groups, and the same design choice can have different implications for each group.

Supervisors. Supervisors are the primary user and the population from which both human-study pools were drawn. The interface is most relevant for supervisors whose marginal cost of writing the first draft of a comment is high, either because they are time-constrained, because they are seeing many drafts in a short window, or because they want to ensure consistent coverage of standard issues across all students. It is less relevant for supervisors whose comments are already highly individualised and explanatory, because the editing cost of a locally specific suggestion with limited rationale can exceed the cost of writing a comment from scratch. A responsible deployment is therefore opt-in at the supervisor level rather than mandated at the course level.

Students. Students do not interact with the system directly. They receive its output indirectly, through the supervisor's annotated PDF. What matters to them is whether the comments they receive, after the supervisor's review, help them understand what to revise and why. A comment that says "revise this sentence" is less useful than one that says "revise this sentence because the claim it makes is not supported by the evidence in the previous paragraph". The current system provides limited rationale partly because the training distribution contains many short comments. The transparency obligation is also stronger for students than for any other group: they cannot calibrate their reading of a comment unless they know whether it originated with the supervisor or with the model.

Course coordinators. Course coordinators operate between individual supervisors and institutional policy. The most important risk from this perspective is uneven adoption. CSE3000 enrols close to four hundred students per year [11]. Because this thesis does not measure supervisor workload reduction, course-level use should be framed as a review-support workflow rather than as an efficiency guarantee. Any deployment would need explicit usage guidance, audit checks that confirm the supervisor interacted with the suggestions, and visible distinction between AI-authored and supervisor-authored annotations in the exported feedback.

Institutions. The institution sets the data-protection, retention, and AI-governance rules within which the course operates. The system was designed inside the constraints that the TU Delft institutional governance imposes: open-weight models, institutional hosting on DelftBlue, anonymisation before training, consent-based human evaluation. These constraints were preconditions for the project to be approved at all. The same design pattern could be evaluated in other courses or supervision contexts at the same institution under similar data-protection assumptions. Portability to a different institution is a separate question that depends on that institution's own rules.

3.8.3. Risks and Conditions for Responsible Deployment

Three risks deserve to be named explicitly. The first is over-reliance: a fluent comment can sound authoritative without being correct, and a supervisor who treats AI suggestions as a checklist can produce shallower feedback than one who writes from scratch. The interface reduces this risk by requiring acceptance before export, but it cannot ensure that every accepted suggestion has been

carefully checked. The second is inequity: supervisors who adopt the system inconsistently can produce uneven feedback quality across students, which is a substantive concern in a course that enrols several hundred students per year. The third is opacity to students: a student who receives AI-influenced feedback without knowing it cannot calibrate how much weight to put on each comment.

These risks define the conditions under which the contribution is defensible. Generated comments must enter the workflow as editable suggestions and must require a supervisor accept action before any export. Provenance must be preserved in export so that students can see which annotations are AI-authored and which are supervisor-authored. The current prototype marks AI suggestions during review, but a production version should preserve that provenance more explicitly in every exported annotation, including highlights whose anchor geometry is resolved successfully. Adoption must be opt-in at the supervisor level. Audit data must be retained without retaining student-identifying material. The rationale dimension that the blind-study raters surfaced, and that the current rubric does not score, should be added in any future iteration. Each of these is a precondition for course-level deployment rather than an optional feature, and together they point at the conditions under which open-weight LLMs can take a useful but bounded role in thesis supervision.

References

[1] John Brooke. "SUS: A "Quick and Dirty" Usability Scale". In: *Usability Evaluation in Industry*. Ed. by Patrick W. Jordan, Bruce Thomas, Ian L. McClelland, and Bernard Weerdmeester. London: Taylor and Francis, 1996, pp. 189–194.

[2] John Hattie and Helen Timperley. "The Power of Feedback". In: *Review of Educational Research* 77.1 (2007), pp. 81–112. DOI: 10.3102/003465430298487.

[3] Valerie J. Shute. "Focus on Formative Feedback". In: *Review of Educational Research* 78.1 (2008), pp. 153–189. DOI: 10.3102/0034654307313795.

[4] David J. Nicol and Debra Macfarlane-Dick. "Formative Assessment and Self-Regulated Learning: A Model and Seven Principles of Good Feedback Practice". In: *Studies in Higher Education* 31.2 (2006), pp. 199–218. DOI: 10.1080/03075070600572090.

[5] John Bitchener, Helen Basturkmen, and Martin East. "The Focus of Supervisor Written Feedback to Thesis/Dissertation Students". In: *International Journal of English Studies* 10.2 (2010), pp. 79–97. DOI: 10.6018/ijes/2010/2/119201.

[6] Vijay Kumar and Elke Stracke. "An analysis of written feedback on a PhD thesis". In: *Teaching in Higher Education* 12.4 (2007), pp. 461–470. DOI: 10.1080/13562510701415433.

[7] Susan Carter and Vijay Kumar. "Ignorance, uncertainty and confusion: Writing feedback and doctoral supervision". In: *Innovations in Education and Teaching International* 54.1 (2017), pp. 68–77. DOI: 10.1080/14703297.2015.1123104.

[8] Yong Wu and Christian D. Schunn. "From Feedback to Revisions: Effects of Feedback Features and Perceptions". In: *Contemporary Educational Psychology* 60 (2020), p. 101826. DOI: 10.1016/j.cedpsych.2019.101826.

[9] Ernesto Panadero and Anastasiya A. Lipnevich. "A review of feedback models and typologies: Towards an integrative model of feedback elements". In: *Educational Research Review* 35 (2022), p. 100416. DOI: 10.1016/j.edurev.2021.100416.

[10] Cam Brooks, Annemaree Carroll, Robyn M. Gillies, and John Hattie. "A Matrix of Feedback for Learning". In: *Australian Journal of Teacher Education* 44.4 (2019). DOI: 10.14221/ajte.2018v44n4.2.

[11] Gosia Migut, Aleksander Buszydlik, and Mathijs M. De Weerdt. "The Research Project in Computer Science Bachelor Education: Undergraduate Research Experience at Scale". In: *ITiCSE 2025: Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. New York, NY: Association for Computing Machinery, 2025, pp. 410–416. DOI: 10.1145/3724363.3729101.

[12] Don Houston, Luanna H. Meyer, and Shelley Paewai. "Academic Staff Workloads and Job Satisfaction: Expectations and Values in Academe". In: *Journal of Higher Education Policy and Management* 28.1 (2006), pp. 17–30. DOI: 10.1080/13600800500283734.

[13] Julia Miller. "Where Does the Time Go? An Academic Workload Case Study at an Australian University". In: *Journal of Higher Education Policy and Management* 41.6 (2019), pp. 633–645. DOI: 10.1080/1360080X.2019.1635328.

[14] Andrew S. Griffith and Zeynep Altinay. "A Framework to Assess Higher Education Faculty Workload in U.S. Universities". In: *Innovations in Education and Teaching International* 57.6 (2020), pp. 691–700. DOI: 10.1080/14703297.2020.1786432.

[15] John Kenny and Andrew Edward Fluck. "Emerging Principles for the Allocation of Academic Work in Universities". In: *Higher Education* 83 (2022), pp. 1371–1388. DOI: 10.1007/s10734-021-00747-y.

[16] Enkelejda Kasneci, Kathrin Sessler, Stefan Kuechemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Guennemann, Eyke Huellermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Juergen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. "ChatGPT for good? On opportunities and challenges of large language models for education". In: *Learning and Individual Differences* 103 (2023), p. 102274. DOI: 10.1016/j.lindif.2023.102274.

[17] Yuhong Shi, Kun Yu, Yifei Dong, and Fang Chen. "Large language models in education: a systematic review of empirical applications, benefits, and challenges". In: *Computers and Education: Artificial Intelligence* 10 (2026), p. 100529. DOI: 10.1016/j.caeai.2025.100529.

[18] Qinjin Jia, Jialin Cui, Ruijie Xi, Chengyuan Liu, Parvez Rashid, Ruochi Li, and Edward Gehringer. "On Assessing the Faithfulness of LLM-generated Feedback on Student Assignments". In: *Proceedings of the 17th International Conference on Educational Data Mining (EDM 2024)*. 2024.

[19] Qinjin Jia, Jialin Cui, Haoze Du, Parvez Rashid, Ruijie Xi, Ruochi Li, and Edward Gehringer. "LLM-generated Feedback in Real Classes and Beyond: Perspectives from Students and Instructors". In: *Proceedings of the 17th International Conference on Educational Data Mining (EDM 2024)*. 2024.

[20] Wei Dai, Yi-Shan Tsai, Jionghao Lin, Ahmad Aldino, Hua Jin, Tongguang Li, Dragan Gašević, and Guanliang Chen. "Assessing the proficiency of large language models in automatic feedback generation: An evaluation study". In: *Computers and Education: Artificial Intelligence* 7 (2024), p. 100299. DOI: 10.1016/j.caeai.2024.100299.

[21] Hannah Rashkin, Elizabeth Clark, Fantine Huot, and Mirella Lapata. "Help Me Write a Story: Evaluating LLMs' Ability to Generate Writing Feedback". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2025, pp. 25827–25847.

[22] Juan Escalante, Austin Pack, and Alex Barrett. "AI-generated feedback on writing: insights into efficacy and ENL student preference". In: *International Journal of Educational Technology in Higher Education* 20.57 (2023). DOI: 10.1186/s41239-023-00425-2.

[23] Kathrin Seßler, Arne Bewersdorff, Claudia Nerdel, and Enkelejda Kasneci. *Towards Adaptive Feedback with AI: Comparing the Feedback Quality of LLMs and Teachers on Experimentation Protocols*. 2025. DOI: 10.48550/arXiv.2502.12842. arXiv: 2502.12842 [cs.CL].

[24] Mickie de Wet, Margarita Oja Da Silva, and René Bohnsack. "Can AI give good feedback on essay-type assignments? An explorative case study of LLMs in higher education". In: *Innovations in Education and Teaching International* 62.5 (2025), pp. 1484–1499. DOI: 10.1080/14703297.2025.2536609.

[25] Hana Mohamed Nassar. "Comparing the Quality of AI-generated and Instructor Feedback in a University Writing Program". MA thesis. The American University in Cairo, 2025. URL: https://fount.aucegypt.edu/etds/2468.

[26] Seyyed Kazem Banihashem, Nafiseh Taghizadeh Kerman, Omid Noroozi, Jewoong Moon, and Hendrik Drachsler. "Feedback sources in essay writing: peer-generated or AI-generated feedback?" In: *International Journal of Educational Technology in Higher Education* 21.23 (2024). DOI: 10.1186/s41239-024-00455-4.

[27] İbrahim Rıza Hallaç and Hasan Oğul. *From Prompting to Fine-Tuning: A Comprehensive Evaluation of LLM Strategies for Automatic Essay Assessment*. SSRN preprint. 2025. URL: https://ssrn.com/abstract=5699710.

[28] Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, Daniel A. McFarland, and James Zou. "Can Large Language Models Provide Useful Feedback on Research Papers? A Large-Scale Empirical Analysis". In: *NEJM AI* 1.8 (2024). DOI: 10.1056/AIoa2400196.

[29] Ryan Liu and Nihar B. Shah. *ReviewerGPT? An Exploratory Study on Using Large Language Models for Paper Reviewing*. 2023. DOI: 10.48550/arXiv.2306.00622. arXiv: 2306.00622 [cs.CL].

[30] Ruiyang Zhou, Lu Chen, and Kai Yu. "Is LLM a Reliable Reviewer? A Comprehensive Evaluation of LLM on Automatic Paper Reviewing Tasks". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 9340–9351.

[31] Osama Mohammed Afzal, Preslav Nakov, Tom Hope, and Iryna Gurevych. "Beyond "Not Novel Enough": Enriching Scholarly Critique with LLM-Assisted Feedback". In: *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. Rabat, Morocco: Association for Computational Linguistics, 2026, pp. 2648–2671.

[32] Pengfei Sun, Deqing Huang, Qingyuan Wang, Na Qin, and Liming Cai. "LLM-Based Scholarly Paper Review for Master's Theses". In: *2025 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. Macao, China, 2025.

[33] Huawei Shi and Vahid Aryadoust. "A systematic review of AI-based automated written feedback research". In: *ReCALL* 36.2 (2024), pp. 187–209. DOI: 10.1017/S0958344023000265.

[34] L. H. Alghamdi and T. M. Alghizzi. "Educators' reflections on AI-automated feedback in higher education: a structured integrative review of potentials, pitfalls, and ethical dimensions". In: *Frontiers in Education* 10 (2025), p. 1704820. DOI: 10.3389/feduc.2025.1704820.

[35] Wayne Holmes, Kaska Porayska-Pomsta, Kenneth Holstein, Emma Sutherland, Toby Baker, Simon Buckingham Shum, Olga C. Santos, Maria Rodrigo, Mutlu Cukurova, Ig Ibert Bittencourt, and Kenneth R. Koedinger. "Ethics of AI in Education: Towards a Community-Wide Framework". In: *International Journal of Artificial Intelligence in Education* 32 (2022), pp. 504–526. DOI: 10.1007/s40593-021-00239-1.

[36] Melissa M. Nelson and Christian D. Schunn. "The nature of feedback: How different types of peer feedback affect writing performance". In: *Instructional Science* 37.4 (2009), pp. 375–401. DOI: 10.1007/s11251-008-9053-x.

[37] Sarah Gielen, Ellen Peeters, Filip Dochy, Patrick Onghena, and Katrien Struyven. "Improving the effectiveness of peer feedback for learning". In: *Learning and Instruction* 20.4 (2010), pp. 304–315. DOI: 10.1016/j.learninstruc.2009.08.007.

[38] Robert Sanderson, Paolo Ciccarese, and Herbert Van de Sompel. *Web Annotation Data Model*. W3C Recommendation. 2017. URL: https://www.w3.org/TR/annotation-model/ (visited on 05/08/2026).

[39] Fengchun Miao and Wayne Holmes. *Guidance for Generative AI in Education and Research*. 2023. URL: https://www.unesco.org/en/articles/guidance-generative-ai-education-and-research (visited on 05/08/2026).

[40] European Parliament and Council of the European Union. *Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence*. 2024. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj (visited on 05/08/2026).

[41] European Parliament and Council of the European Union. *Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data*. 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj (visited on 05/08/2026).

[42] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. *OLMo: Accelerating the Science of Language Models*. 2024. DOI: 10.48550/arXiv.2402.00838. arXiv: 2402.00838 [cs.CL].

[43] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. *Lost in the Middle: How Language Models Use Long Contexts*. 2023. DOI: 10.48550/arXiv.2307.03172. arXiv: 2307.03172 [cs.CL].

[44] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. *RULER: What's the Real Context Size of Your Long-Context Language Models?* 2024. DOI: 10.48550/arXiv.2404.06654. arXiv: 2404.06654 [cs.CL].

[45] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2020. DOI: 10.48550/arXiv.2005.11401. arXiv: 2005.11401 [cs.IR].

[46] Zhuowan Li, Cheng Xu, Mohit Iyyer, and Yashar Mehdad. "Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 2024. DOI: 10.18653/v1/2024.emnlp-industry.66.

[47] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*. 2023. DOI: 10.48550/arXiv.2310.11511. arXiv: 2310.11511 [cs.CL].

[48] D. Royce Sadler. "Indeterminacy in the Use of Preset Criteria for Assessment and Grading". In: *Assessment & Evaluation in Higher Education* 34.2 (2009), pp. 159–179. DOI: 10.1080/02602930801956059.

[49] Abdelrahman Sadallah, Tim Baumgärtner, Iryna Gurevych, and Ted Briscoe. "The Good, the Bad and the Constructive: Automatically Measuring Peer Review's Utility for Authors". In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, 2025, pp. 28991–29021. DOI: 10.18653/v1/2025.emnlp-main.1476.

[50] Wenjing He and Ying Gao. "Explicating Peer Feedback Quality and Its Impact on Feedback Implementation in EFL Writing". In: *Frontiers in Psychology* 14 (2023), p. 1177094. DOI: 10.3389/fpsyg.2023.1177094.

[51] Cecilia Superchi, José Antonio González, Ivan Solà, Erik Cobo, Darko Hren, and Isabelle Boutron. "Tools Used to Assess the Quality of Peer Review Reports: A Methodological Systematic Review". In: *BMC Medical Research Methodology* 19.1 (2019), p. 48. DOI: 10.1186/s12874-019-0688-x.

[52] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. "A Dataset of Peer Reviews (PeerRead): Collection, Insights and NLP Applications". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, pp. 1647–1661.

[53] Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Han, and Meng Jiang. "The Impact of Retrieval Augmentation on Long Context Language Models". In: *Proceedings of the Thirteenth International Conference on Learning Representations*. 2025.

[54] Chao Zhang, Kexin Ju, Zhuolun Han, Yu-Chun Grace Yen, and Jeffrey M. Rzeszotarski. "Synthia: Visually Interpreting and Synthesizing Feedback for Writing Revision". In: *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 2025. DOI: 10.1145/3746059.3747703.

[55] Maximilian Sölch, Felix T. J. Dietrich, and Stephan Krusche. "Direct Automated Feedback Delivery for Student Submissions based on LLMs". In: *Companion Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*. 2025. DOI: 10.1145/3696630.3727247.

[56] Dennis Zyska, Alla Rozovskaya, Ilia Kuznetsov, and Iryna Gurevych. *Exposía: Academic Writing Assessment of Exposés and Peer Feedback*. 2026. DOI: 10.48550/arXiv.2601.06536. arXiv: 2601.06536 [cs.CL].

[57] European Commission. *Frequently Asked Questions | AI Act Service Desk*. Accessed May 2026. 2026.

[58] European Data Protection Board. *Guidelines 4/2019 on Article 25 Data Protection by Design and by Default*. 2020.

[59] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. *LIMA: Less Is More for Alignment*. 2023. DOI: 10.48550/arXiv.2305.11206. arXiv: 2305.11206 [cs.CL].

[60] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. *AlpaGasus: Training a Better Alpaca with Fewer Data*. 2023. DOI: 10.48550/arXiv.2307.08701. arXiv: 2307.08701 [cs.CL].

[61] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. *AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework*. 2023. DOI: 10.48550/arXiv.2308.08155. arXiv: 2308.08155 [cs.AI].

[62] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Juergen Schmidhuber. "MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework". In: *Proceedings of the Twelfth International Conference on Learning Representations*. 2024.

[63] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. "Improving Factuality and Reasoning in Language Models through Multiagent Debate". In: *Proceedings of the 41st International Conference on Machine Learning*. 2024.

[64] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. *Large Language Model Based Multi-Agents: A Survey of Progress and Challenges*. 2024. arXiv: 2402.01680 [cs.CL].

[65] Tian Lan, Wenwei Zhang, Chengqi Lyu, Shuaibin Li, Chen Xu, Heyan Huang, Dahua Lin, Xian-Ling Mao, and Kai Chen. "Training Language Models to Critique With Multi-Agent Feedback". In: *Findings of the Association for Computational Linguistics: EMNLP 2025*. Suzhou, China: Association for Computational Linguistics, 2025, pp. 1474–1501. DOI: 10.18653/v1/2025.findings-emnlp.78. URL: https://aclanthology.org/2025.findings-emnlp.78/.

[66] Eric Horvitz. "Principles of Mixed-Initiative User Interfaces". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1999, pp. 159–166. DOI: 10.1145/302979.303030.

[67] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. "Guidelines for Human-AI Interaction". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019. DOI: 10.1145/3290605.3300233.

[68] Ben Shneiderman. "Human-Centered Artificial Intelligence: Reliable, Safe and Trustworthy". In: *International Journal of Human–Computer Interaction* 36.6 (2020), pp. 495–504. DOI: 10.1080/10447318.2020.1741118.

Study Instruments and Rubrics A

This appendix preserves the human-study instruments used by the thesis. The first study evaluates individual feedback comments under blind conditions. The second study evaluates the usability of the supervisor-facing interface. They are kept together here because both are human-centered, but they answer different empirical questions.

A.1. Blind Feedback-Quality Study Rubric

Participants rated each feedback item without knowing whether it was written by a human supervisor or generated by the fine-tuned model. The first five dimensions used a five-point agreement scale from strongly disagree to strongly agree, encoded as −2, −1, 0, +1, and +2.

Dimension	Wording shown to participants
Relevant and on-topic	This feedback directly addresses the given excerpt and stays focused on its content. It does not introduce unrelated comments.
Clear about whether change is needed	This feedback makes it clear what the student should do, such as whether the text should be kept, revised, or removed. The recommended action is understandable.
Correct	The feedback is factually and contextually accurate. Its claims are supported by the excerpt and do not introduce incorrect or unjustified points.
Useful / actionable	This feedback provides guidance that would help a student make a concrete improvement to the text. It goes beyond identifying an issue and supports revision.
Specific	This feedback refers to specific words, phrases, claims, or parts of the excerpt. It is grounded in the actual text and not overly general.

A.2. Supervisor Suitability Scale

Supervisor suitability used a separate ordered scale because the question was not only whether the comment was locally good, but whether it would be appropriate for a thesis supervisor to provide.

Value	Label
−2	Not suitable
−1	Suitable after major revision
+1	Suitable after minor editing
+2	Suitable as-is

A.3. SUS Interface-Usability Instrument

The second user study evaluated the main supervisor-facing thesis pipeline with the System Usability Scale. Participants rated each item from strongly disagree (1) to strongly agree (5).

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Prompt Templates B

B.1. Prompt Provenance

The main review prompt used in this thesis was derived from the writing expectations given to students in the TU Delft CSE3000 bachelor thesis course. Two local source documents were used as the main basis:

1. the CSE3000 research paper template;
2. the CSE3000 research-paper checklist.

These materials emphasize clear research purpose, justified claims, reproducible methodology, interpretable results, sound argumentation, responsible research, strong paragraph structure, and correct use of references, tables, and figures. The prompt translates these expectations into operational model instructions for span-anchored review.

B.2. Main Review Prompt

```
You are a professor reviewing a Bachelor Thesis excerpt for TU Delft
CSE3000-style academic writing quality.

Important exclusion rule:
Some excerpts may contain placeholders, drafting notes, author reminders,
revision instructions, or self-declared missing content rather than actual
thesis prose. These are NOT part of the thesis content and must be ignored
completely.

Treat any text like the following as non-reviewable placeholder text:
- lines starting with or containing: "TODO", "TO DO", "TBD", "FIXME",
  "NOTE TO SELF"
- author reminders such as "to be added", "to be written",
  "still needs to be added", "still needs to be written", "pending",
  "unfinished", "draft", "placeholder"
- bullet points or notes that explicitly describe what the author plans to
  write later
- checklist items, reminders, or comments to the author
- text that only states something is missing or will be completed later

Never produce feedback on such placeholder text.
Never select such text as the "sentence".
Never tell the student to add, complete, summarize, acknowledge, write, or
expand something when the excerpt already explicitly says that this content is
still to be written or added later.

If a sentence is a placeholder, reminder, drafting note, checklist item, or
revision instruction, skip it and look for real writing issues elsewhere.
If the excerpt consists only of placeholders or drafting notes, return [].
```

```
Your task is to identify the most important issues in the excerpt based only
on the provided text.

Judge the text using the following academic writing expectations:
- clear research purpose, contribution, or line of argument
- justified claims and interpretations
- clear and motivated methodology
- clearly reported results
- well-structured and source-grounded argumentation
- conclusions linked back to the research question or objective
- responsible research, ethics, and reproducibility where relevant
- well-structured paragraphs and topic sentences
- objective and academically precise style
- correct use of references, figures, and tables

Focus on high-impact issues such as:
- unclear or missing research question, motivation, contribution, or conclusion
- weak argumentation or reasoning gaps
- unsupported, vague, or overstated claims
- missing explanation of why a method was chosen
- methodology or experimental setup that is too unclear to understand or
  reproduce
- results or interpretations that are incomplete, unclear, or unsupported
- conclusions that do not clearly follow from earlier material
- weak paragraph structure or poor flow
- missing critical engagement with prior work or sources
- unclear responsible research, ethics, or reproducibility discussion
- ambiguous references to figures, tables, sections, prior work, or
  terminology
- tables or figures that are not interpretable on their own, not properly
  referred to, or insufficiently explained
- imprecise, non-objective, ambiguous, or academically weak language

Return ONLY JSON.
Return a JSON array with at most {NUM} items.

Each item must contain:
- "sentence": the exact problematic text span copied verbatim from the thesis
  excerpt
- "actionable_feedback": one concise, specific, and actionable improvement
  suggestion
```

B.3. Role of the Prompt in the Thesis

This prompt is important because it operationalizes the target review behavior used throughout the thesis. It links course-level academic-writing guidelines to the model-level generation task and provides the conceptual basis for both prompting and evaluation.

Interface Screenshots C

This appendix documents the implemented interface artifact. The main thesis text discusses the interface as a workflow for supervisor-controlled feedback; these screenshots preserve the concrete visual state of the website used for the studies and demonstration.

C.1. Landing Page

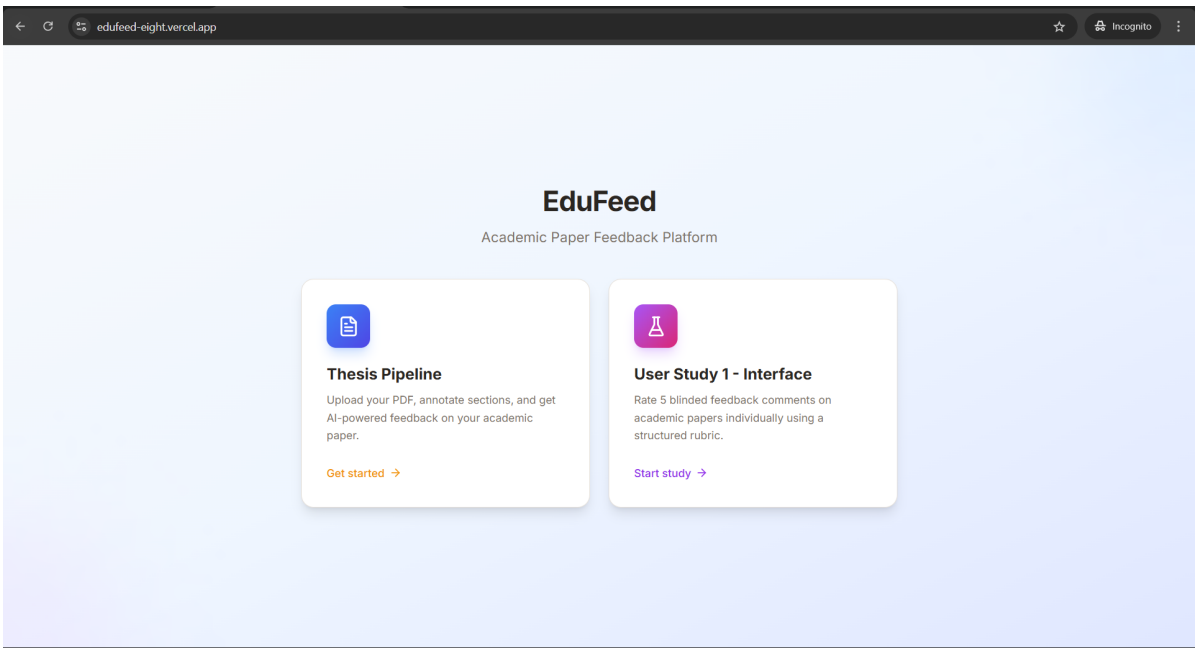

Figure C.1: Interface landing page showing the two implemented paths: the thesis feedback pipeline and the blind feedback-quality study interface.

C.2. Supervisor Pipeline

C.3. Blind Feedback-Quality Study

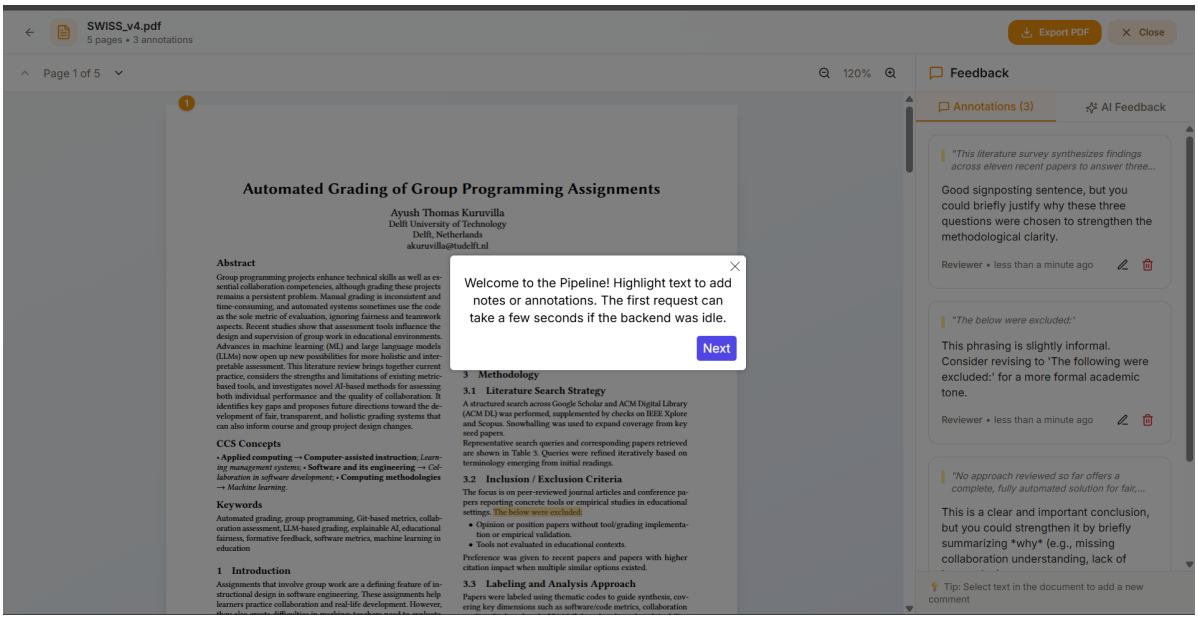

Figure C.2: Supervisor-facing PDF pipeline with document view, annotation cards, AI-feedback controls, and export affordance.

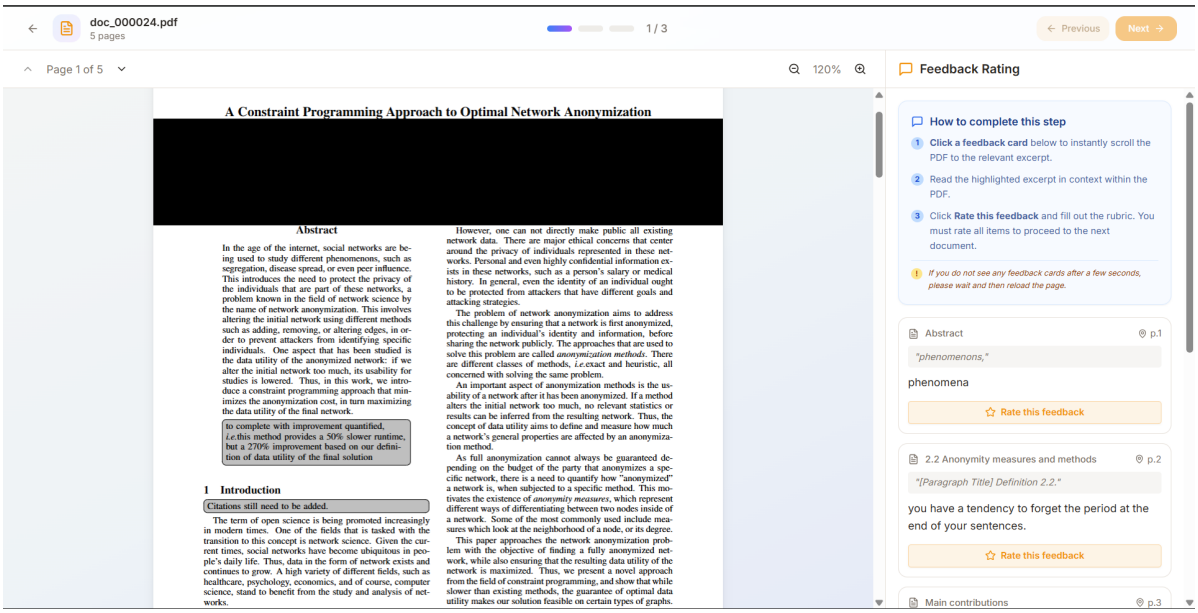

Figure C.3: Blind feedback-quality study page showing a held-out thesis document and feedback cards to be rated without source labels.

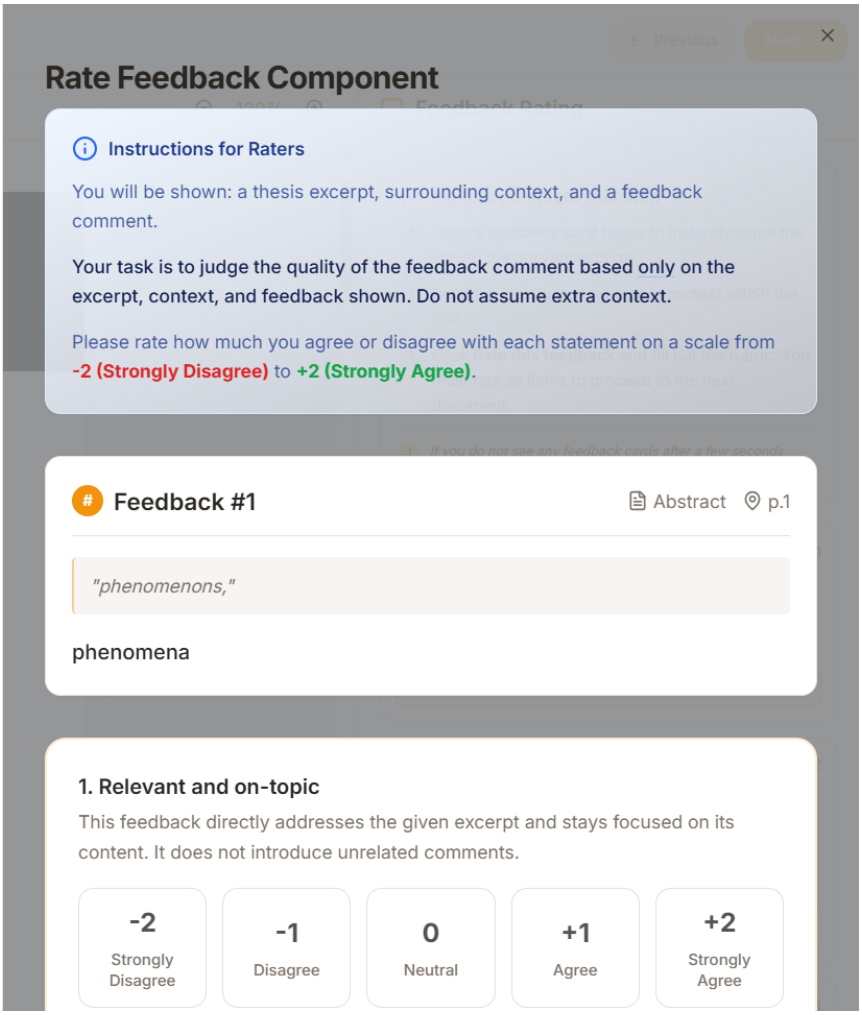

Figure C.4: Rating modal for the blind feedback-quality study. The modal presents the rubric criteria on a -2 to +2 Likert scale while keeping the feedback source hidden.

Guideline Basis for Prompt Design D

D.1. CSE3000 Writing Expectations

The prompt and evaluation framing used in this thesis were derived from the written guidance provided to students in the CSE3000 bachelor thesis course. The two main source documents were the course research paper template and the associated checklist.

The checklist emphasizes the following expectations for a strong research paper:

- an abstract that includes motivation, main question, method, and conclusions;
- an introduction with background, motivation, a clear research question or objective, and article structure;
- a methodology that is clear, justified, and reproducible;
- results that are fully reported and supported by figures or tables where needed;
- argumentation that is clear, structured, critical, and grounded in trustworthy sources;
- responsible research and reproducibility;
- conclusions clearly linked back to the research question;
- correct use of references, quotations, paraphrases, and bibliography;
- well-structured paragraphs and topic sentences;
- objective, clear, and grammatically sound style;
- interpretable tables, figures, and overall layout.

D.2. Why These Guidelines Matter for the Thesis

These guidelines are not included merely as background material. They define the normative writing standard that the review prompt attempts to approximate. In other words, the LLM is not being asked to produce generic writing comments, but to produce feedback consistent with the expectations of the CSE3000 thesis-writing context.

D.3. Connection to the Evaluation Framework

The link between the guidelines and the evaluation framework is direct. The prompt uses the checklist-derived expectations to determine what kinds of issues should be surfaced, while the blind human study and the LLM-as-a-judge pipeline evaluate whether the resulting feedback is relevant, clear, correct, actionable, and specific. The guidelines therefore anchor both generation and evaluation in the same academic-writing standard.

Additional LLM-as-a-Judge Results E

This appendix preserves judge-benchmark artefacts that supplement the strategy comparison in the scientific article and the per-system diagnostics in Section 3.7 with the per-judge rankings, the top individual systems, and the failure breakdown.

E.1. Per-Judge Approach Rankings

The judges differ in absolute severity but place the three strategies in the same order. Table E.1 reports the per-judge ranking with the corresponding overall macro score on the [−2, +2] scale.

Judge model	Approach	Rank	Overall mean
DeepSeek-R1 32B Qwen Distill	Whole-document prompting	1	1.56
DeepSeek-R1 32B Qwen Distill	Agentic review	2	1.49
DeepSeek-R1 32B Qwen Distill	Section-aware two-stage	3	1.41
Gemma 4 31B	Whole-document prompting	1	1.67
Gemma 4 31B	Agentic review	2	1.50
Gemma 4 31B	Section-aware two-stage	3	1.48
Mistral Small	Whole-document prompting	1	1.20
Mistral Small	Agentic review	2	1.14
Mistral Small	Section-aware two-stage	3	1.10

Table E.1: Per-judge approach ranking by overall macro score across the six rubric dimensions. Mistral Small is substantially stricter than DeepSeek and Gemma in absolute terms, but the ranking is identical across judges.

E.2. Top Individual Systems

Table E.2 lists the six highest-scoring systems across all backbones and strategies. The table is dominated by Qwen 3.5 variants and spans all three strategy families. The `qwen25-*` system identifiers used throughout this appendix are retained from the original storage paths and all denote the Qwen 3.5 27B backbone.

System	Strategy	Tier	Overall mean
`full_context_all/base/qwen25-27-base`	Whole-document prompting	base	1.63
`full_context_all/finetuned/qwen25-27-finetuned`	Whole-document prompting	finetuned	1.59
`agentic_review_dir_v3_qwen27`	Agentic review	finetuned	1.58
`ollama_chunking_base/qwen25-70-base`	Section-aware two-stage	base	1.52
`agentic_review_dir_v3_qwen25-27-base`	Agentic review	base	1.45
`full_context_all/base/llama33-base`	Whole-document prompting	base	1.42

Table E.2: Top individual systems by judge-macro overall score. The `qwen25-70-base` system is mapped into the Qwen 3.5 27B base bucket during canonicalization.

E.3. Fine-Tuning Effect Heatmap

Figure E.1 visualizes the per-dimension difference between fine-tuned and base variants for each backbone and strategy. Cells coloured negative indicate that fine-tuning reduced the judge's average score on that dimension.

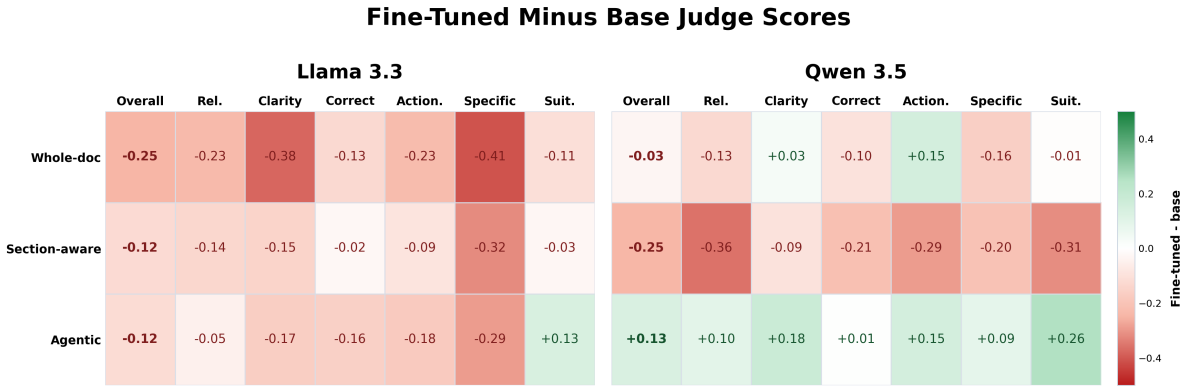

Figure E.1: Heatmap of fine-tuned minus base judge-macro scores per rubric dimension, by backbone and strategy. Cells are score *differences* on the underlying [−2, +2] rubric scale: a positive (good) value means fine-tuning raised the judge score on that dimension, a negative (bad) value means it lowered it, and zero means no change. Most cells are negative or near zero; the largest positive cells are in the Qwen 3.5 agentic row.

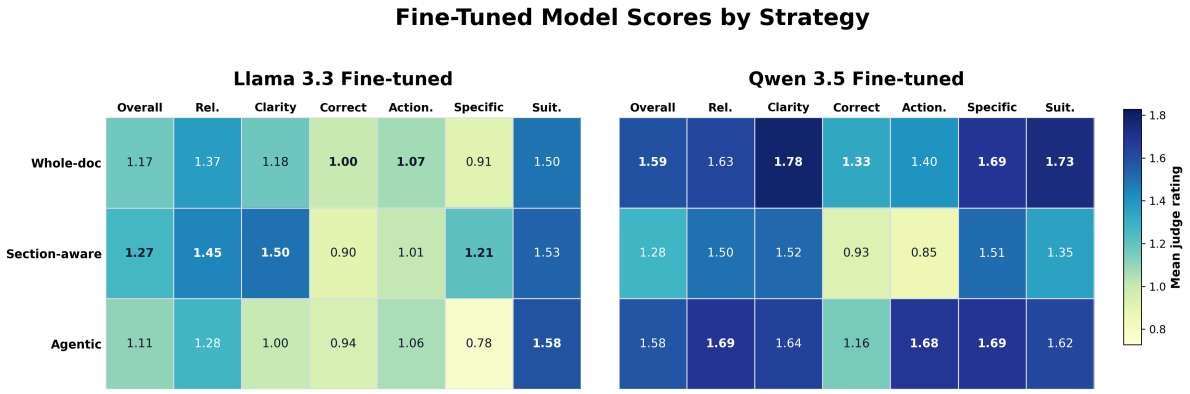

Figure E.2: Fine-tuned per-system judge-macro scores by approach and rubric dimension. Scores are on the [−2, +2] rubric scale, where higher is better (+2 is the best possible score, −2 the worst, and 0 neutral). The Qwen 3.5 fine-tuned agentic configuration is the strongest fine-tuned cell on actionability and supervisor suitability.

E.4. Coverage and Failure Breakdown

The two judge runs together produce 1,491 successful item-level judgments and 12 invalid-score failures after canonicalization. Mistral Small and DeepSeek-R1 32B each judge their full item pools without invalid-score failures; all 12 failures originate from Gemma 4 31B. The canonicalization step that produced the analysis pool excluded 267 duplicate-system records on the success side and 6 on the failure side, and remapped 21 items from `qwen25-70-base` into the `qwen25-27-base` bucket so that base and fine-tuned variants align under a single backbone label. These counts are preserved in the analysis artefacts together with the run identifiers `judge_run_9715946` and `judge_run_9719554`.

E.5. Held-Out Document Identifiers

Five thesis documents were reserved as the held-out evaluation pool before any model adaptation or filtering. Their stable identifiers are retained in the analysis artefacts, and the same pool is reused across the judge benchmark, the blind feedback-quality study, and qualitative inspection. Re-running any of the strategies on the same held-out set therefore produces results directly comparable to those reported

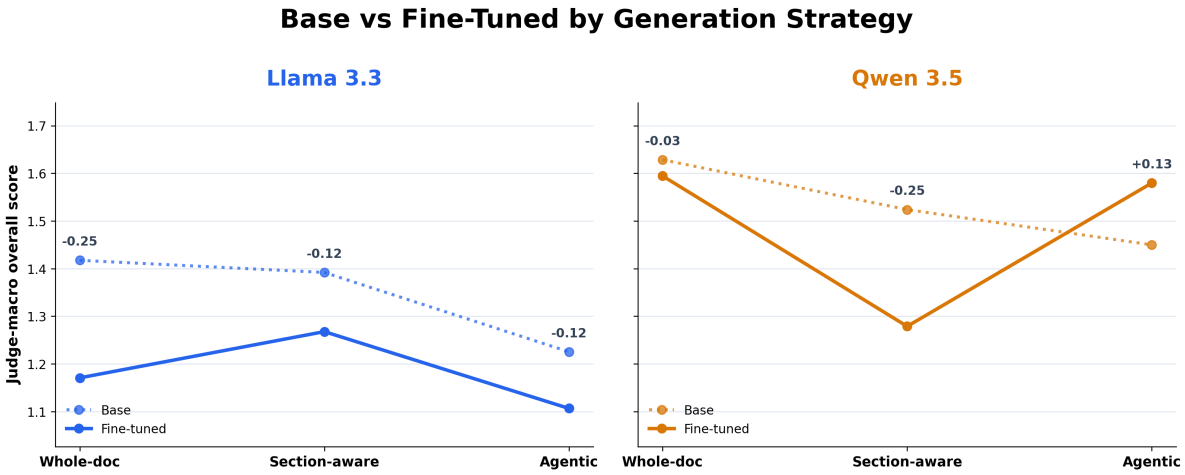

Figure E.3: Per-strategy comparison between base and fine-tuned variants for each backbone, showing where fine-tuning shifts the per-dimension scores up or down. The vertical axis is the [−2, +2] rubric scale, where higher is better (+2 best, −2 worst, 0 neutral), so a line moving upward indicates an improvement and a line moving downward a degradation.

in this thesis.

Generative AI Usage F

Generative AI tools were used as limited support tools during the thesis process. They were not used as authoritative sources and did not replace the author's responsibility for the research design, implementation, analysis, or conclusions.

- **Writing and structure.** Generative AI was used to suggest alternative paragraph structures, improve flow between sections, shorten text for the ACL-format paper, and rephrase draft sentences for clarity. Suggested text was edited, rewritten, or rejected by the author.
- **LaTeX and formatting.** Generative AI was used for LaTeX troubleshooting, build-log inspection, and organisation of compilation artefacts.