University of Tübingen

Faculty of Science

Department of Computer Science

Master Thesis in Machine Learning

Evaluating AI-Supported Planning for Differentiated Mathematics Homework

Apoorv Agnihotri

July 2026

Reviewers

Wieland Brendel

(Machine Learning)

ELLIS Institute Tübingen

Max Planck Institute for Intelligent Systems

Matthias Bethge

(Machine Learning)

University of Tübingen

Apoorv Agnihotri:

Evaluating AI-Supported Planning for Differentiated Mathematics Homework

Master Thesis in Machine Learning

University of Tübingen

Thesis period: 07.01.2026 – 07.07.2026

Abstract

This thesis studies whether an AI-supported planning tool can help novice teachers create mathematics homework for classes where students work at different levels. The focus is on German secondary schools. Phase A used teacher interviews and teaching material to identify three needs: task search, clear options for student groups, and teacher control when selecting, ordering, and adapting existing tasks.

In Phase B, pre-service mathematics teachers tested a demo editor manually or with an AI planning assistant. The evaluation combines questionnaire responses, final homework plans, interaction logs, and chat logs to examine early teacher-facing usefulness.

AI support produced mixed results in this prototype. It helped some participants generate ideas and find tasks, but also increased tool complexity and created checking and repair work. Manual participants produced clearer three-way task branches for different student groups. AI-assisted plans more often contained support material, catalogue-mismatched task references, or structures needing repair. These findings are specific to this prototype and its implementation constraints, including both the use of a smaller, lower-cost model and an unoptimized AI stack, and should not be generalized to all AI-assisted planning systems.

The design conclusion is that teacher-facing AI planning tools should help teachers search, organise, and adapt existing materials without taking over teacher control. Their outputs need to be understandable, checkable, and easy to correct. Effective AI support depends first on learnable workflows and low integration friction. This points to faster prototype iterations and tighter feedback collection.

Acknowledgements

I would like to thank my supervisor, Dr. Wieland Brendel, for his guidance, support and trust throughout this project and giving me the opportunity to work on this project. I am immensely grateful to Alex Braun for always helping me with all kinds of problems, from technical issues to personal challenges. I am nothing but delighted to have had his support and encouragement throughout our collaboration. I would immensely like to thank Dr. Thomas Klein for his invaluable insights in organizing the work and breaking down the problem into smaller pieces and focusing on the right details at the right time. Including all the kudos I recieved on my runs, which certainly keep me going.

This work would not have been possible without the constant support I received from my family and friends. I am extremely grateful to my sister, Kavya, for being my coffee break buddy and helping me keep my spirits up during the most challenging times. I am thankful to my parents for their unconditional love and support. I would also like to thank Abhavya Chandra for his consistent and never-ending support since I met him more than a decade ago. I am grateful to Lara Müller for pushing me and for her candid mockery whenever I suggested unrealistic timelines. I thank Nilsu Saglam for her amazing and detailed feedback on the drafts.

I am extremely thankful to Prof. Andreas Lachner and Pauline Frick for their help with the study design and evaluation ideas. I would also like to thank Martí Quixal for his help in validating the task for the study and also his amazing spirits. Phase A work would have been impossible without Heike Martina Russ' help in connecting me to mathematics teachers who were comfortable in English for formative interviews. I am grateful to all the teachers and particpants who provided valuable feedback.

Lastly, I am grateful to everyone I worked with during this thesis who I couldn't mention here.

Contents

Chapter 1

Introduction

Mathematics teachers often need to design homework for a whole class while accounting for students with different prior knowledge, confidence, and learning speed. In classroom teaching, teachers already plan for such variation: they define a learning goal, choose and order tasks, anticipate difficulties, and prepare support or extension material. This thesis transfers that planning logic to homework. Because students work away from the teacher, a homework assignment can be more than a list of exercises. It can become a planned route through shared work, level-specific alternatives, and optional support.

This thesis calls that route a *homework learning path*. A homework learning path is a teacher-authored structure for assigning homework: it contains task nodes, points where students can receive different tasks, and optional support material for students who need help. Its purpose is to make the teacher's planning decisions visible and editable. Figure 1.1 shows the basic structure: students can begin with shared tasks, move through a temporary differentiated section for basic, expected, or challenge work, and then return to shared follow-up tasks. The thesis investigates whether an AI-supported planning tool can reduce this planning burden while keeping teachers in control of the resulting homework learning path. The tool supports teacher-side planning and is evaluated through teacher experience, process evidence, and produced homework artefacts.

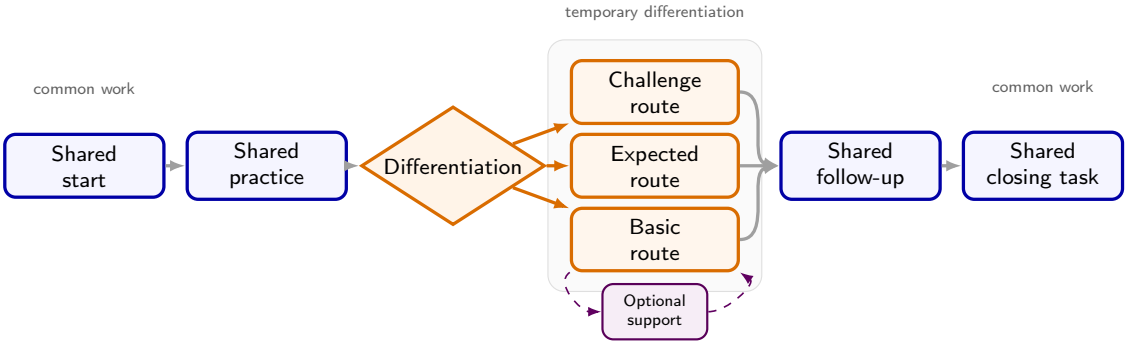


Figure 1.1: Simplified structure of a homework learning path with shared work, temporary differentiated routes, optional support, and shared follow-up tasks.

1.1 Thesis Argument and Key Takeaway

The central argument is that AI support for differentiated homework should be evaluated as planning support, not as automatic homework generation. Teachers still select tasks, organise sequences, and decide where differentiation is needed. A tool therefore makes task fit, differentiation logic, and support material easy to inspect and edit.

The key takeaway is cautious. AI assistance may help with ideation, task search, and drafting, but it can also introduce checking, repair, and usability burden. The evaluation therefore combines usability, catalogue grounding, differentiation structure, interaction traces, produced artefacts, and participant feedback. The formative interviews are also a contribution: they identify practical constraints around differentiated homework planning that designers of similar teacher-facing AI systems would otherwise have to rediscover.

1.2 Background and Problem Context

1.2.1 Problem Context

Differentiated instruction is a well-established response to mixed-ability classrooms [27, 30]. The practical planning work behind differentiation is demanding because teachers have limited time to find, choose, and adapt suitable tasks for a specific class [14, 16].

In mathematics classes, students often differ strongly in prior knowledge and learning speed. In this thesis, *task levels* means assigning different tasks to weaker, expected-level, and stronger students. The study uses an initial controlled task catalogue based on ALICE and Mathebattle tasks [21, 29]. *Support paths* means optional explanations, worked hints,

links, or short videos that help students when they get stuck. Together, task levels and support paths turn a single homework assignment into a structured planning problem. Teachers must prepare material that fits the class profile, learning goals, and available planning time [12, 14].

The prototype addresses this planning work. It supports task search, selection, and structuring while leaving pedagogical control with the teacher. Conversational copilots may help teachers search content, draft structures, and refine plans [12, 14].

1.3 Needs Discovery and Design Input (Phase A)

Phase A examined teachers' existing planning practices through semi-structured interviews and a review of curriculum- and textbook-related material.

Common materials already include some differentiation, for example levelled or column-based textbook structures and current Klett mathematics materials for Grade 6 [13]. However, the interviews suggested that teachers still need to select, combine, or adapt tasks for a particular class.

Phase A therefore motivates the prototype design and defines the practical criteria used later to interpret whether the prototype addresses a real teacher-planning problem.

1.4 Research Gap

Many discussions about educational AI focus on eventual classroom outcomes. For an early-stage planning tool, this thesis evaluates the planning process and teachers' perception of the resulting material. This is also theoretically relevant: Backfisch et al. [2] found that teachers' perceived utility-value of educational technology mediated differences in the instructional quality and technology exploitation of technology-enhanced lesson plans. This thesis addresses that gap with a two-phase approach: first derive requirements from teacher practice, then test whether the prototype meets those requirements in a controlled planning task.

1.5 Research Question and Scope

1.5.1 Primary Research Question

Does AI-assisted planning help novice teachers create differentiated homework learning paths more efficiently, while maintaining or improving perceived homework quality, compared with manual planning?

1.5.2 Operational Scope and Delimitations

The scope is intentionally narrow: planning support for differentiated homework in a controlled task setting. The evaluation focuses on participants' experience and what they produced. This includes interaction traces, questionnaire responses, and homework learning-path artefacts. Long-term adoption, classroom implementation, and direct effects on student learning remain outside this evaluation.

1.6 Thesis Contributions

This thesis makes four contributions:

1. It synthesises formative teacher interviews and material review to identify practical constraints around task discovery, three-level differentiation, teacher control, student-side technology logistics, and diagnostic visibility (Section 4.1.1; see also Section 5.2).

2. It translates those constraints into a teacher-facing homework learning-path planning prototype for differentiated mathematics homework design (Chapter 3).

3. It evaluates the prototype in a controlled two-condition study comparing manual and AI-assisted planning (Section 4.1.2 and Chapter 5).

4. It shows where prompting-based AI support is useful and where stronger structural constraints are needed before such systems can reliably support novice teachers (Sections 6.2 and 6.5.2).

1.7 Thesis Structure

Chapter 2 presents the literature review and positions the research gap. Chapter 3 describes the implemented learning-path platform, including the manual editor, data model, AI planning workflow, and study instrumentation. Chapter 4 presents the two-phase methodology, evaluation framework, study design, and metrics. The remaining chapters present results, discussion, and conclusions.

Chapter 2

Literature Review

This chapter links four areas—pedagogy, teacher workflows, AI support, and evaluation methods—to the design problem: AI-supported planning of differentiated homework learning paths with an existing task catalogue, visible learner-group structures, and teacher control.

2.1 Pedagogical Foundations for Differentiation

Differentiated instruction is a well-established response to mixed-ability classrooms [30]. In mathematics, this often means assigning different challenge levels while keeping a shared learning goal. Scaffolding adds adjustable support that matches learners' current level and changes as they become more capable [33, 36]. Recent secondary-school evidence links differentiated instruction to improved outcomes when implementation quality and teacher support are strong [27].

Cognitive activation is another useful lens for judging mathematics task quality. A task is cognitively activating when it asks students to reason, connect representations, or apply ideas beyond procedural repetition. Research in mathematics education links cognitive activation and individual learning support to student progress [3]. Task-design research also argues that the tasks students receive influence what they have the opportunity to learn [28]. Together with differentiation and scaffolding, this suggests that homework learning paths should be judged by both task coverage and the meaningful organisation of tasks and supports for different learners.

Finally, lesson-plan quality depends on more than content coverage. Teacher expertise

and planning quality also shape how useful technology-enhanced plans become in practice [2]. Taken together, this literature suggests that homework learning paths should be evaluated both for task suitability and for how clearly they represent differentiation choices and support options to the teacher.

2.2 Teacher Workflow and Practical Constraints

Lesson planning is a practical workload problem. Teachers often need to search for, select, adapt, and sequence activities while also accounting for a diverse class. This burden helps explain why recent systems explore AI support for lesson planning and customisation [12, 14]. Related work shows adjacent workload pressures: special-education teachers face substantial paperwork and non-teaching tasks [16], while AI may redistribute teachers' cognitive load rather than simply reduce it [26].

A second constraint is feedback and assessment. AI systems can draft lesson or homework plans, suggest next teaching moves, or help interpret formative-assessment evidence. However, teachers still need to review these outputs and decide whether they fit the learning goal, the available material, and the needs of their class. This matters for tool design because AI or analytics outputs are more useful when they support action rather than adding more information to inspect [8, 35]. AI can support feedback and teaching decisions, but it should remain a tool that teachers review and control [38].

A third constraint is technology logistics and adoption. Field studies and acceptance research show that AI support is shaped by context: infrastructure, language, administrative demands, and whether teachers feel able and willing to use the tool [4, 12]. In this thesis, that supports a planning workflow that avoids student-device requirements. The workflow problem is therefore to help teachers find, select, sequence, and revise differentiated homework material.

2.3 Local School Context and Material Ecosystem

The local context matters because the prototype does not start from a lack of educational content. Phase A interviews and material review indicated that teachers already work with differentiated materials, including levelled or column-based exercises and QR-linked

resources; current Grade 6 Klett mathematics materials provide one concrete textbook example of this material ecosystem [13]. Existing materials still leave teachers with class-specific selection, combination, sequencing, and adaptation work. Teachers described differentiation through low-, medium-, and high-level alternatives, wanted generated material to remain editable, and noted student-device logistics. The requirement-level version of these findings is reported in Section 4.1.1, and their role in the results analysis is made explicit in Section 5.2.

These findings translate into a narrow prototype requirement: AI-assisted homework planning should help teachers organise available tasks into visible levelled alternatives while keeping the output editable and teacher-controlled.

2.4 AI Assistance for Lesson Planning

Recent systems show growing interest in AI support for teacher planning and instruction. LessonPlanner reports better lesson-plan quality and lower workload for novice teachers in a controlled comparison with a ChatGPT baseline [14]. Shiksha Copilot reports reduced planning burden and contextual adaptation in low-resource settings [12]. Teacher-education work also argues that generative AI can contribute to lesson planning when it supports planning, critical thinking, and reflective openness under teacher judgement [31]. Related work on LLM-based personalized learning-path planning illustrates the technical feasibility of prompt-engineered path generation for learner-path personalization; this thesis focuses on teacher-side homework planning [23]. Tutor CoPilot is adjacent rather than a lesson-planning system: it demonstrates that human-AI instructional support can improve live tutoring outcomes, especially for lower-rated tutors, while still requiring attention to suggestion quality and contextual fit [35].

Across these studies, a consistent pattern appears: AI can propose drafts and options, but teachers need final control. This matches teacher-trust literature and explainable-AI arguments in education [20, 22].

Over-reliance remains a risk. If users accept imperfect suggestions uncritically, errors can go unnoticed even when a system seems to make the work easier [25]. For this reason, this thesis evaluates AI assistance as a planning partner inside a structured editor.

2.5 Evaluating Early-Stage Educational AI Prototypes

The previous sections motivate teacher-controlled, catalogue-grounded, and inspectable AI planning support. The remaining question is how to evaluate such support at an early prototype stage.

Direct student learning outcomes are important evidence for tutoring and instructional systems [32]. This thesis evaluates an earlier step: a planning prototype before classroom use. Evaluation literature for technology-enhanced learning and intelligent tutoring systems supports this evidence base. At this stage, it is reasonable to study the process of use, usability, participant or instructor perceptions, learning analytics, and other system-level measures [9, 17]. Design-Based Research is compatible with this logic because it treats the design and study of technology-enhanced learning environments as iterative and practice-facing [1, 34].

A prototype study also evaluates the particular version of the tool that participants use. By analogy with implementation-fidelity research, if the implemented version differs from the intended design, the findings must be interpreted carefully: weak results may reflect the idea, the implementation, or both [6, 24]. Prototype research makes a related point: a prototype tests selected aspects of a design, such as its role, look and feel, or implementation [19]. For this thesis, this means that interaction problems, invalid suggestions, and repair work are treated as evidence about the current prototype's readiness.

For interaction-focused AI tools, logs are useful because they show what participants actually had to do. Examples in this thesis include suggestion use, repair effort, and conversation turns. These logs are most useful when they are read together with produced artefacts, questionnaire responses, and qualitative feedback, rather than as isolated success indicators [8, 12, 14, 35]. This justifies the evaluation approach used in this thesis: the study examines whether the prototype supports a usable, inspectable, and teacher-controlled planning process.

2.6 Usability, Acceptance, and Trust

Usability and adoption are separate but related questions. The System Usability Scale (SUS) provides a compact usability measure [5], while workload instruments such as

NASA-TLX motivate attention to mental demand, effort, and frustration rather than time alone [18].

Technology acceptance research gives a second, more limited lens. The original TAM identifies perceived usefulness and perceived ease of use as central acceptance variables, and a recent AI-in-education TAM meta-analysis reports strong positive TAM relationships in AIEd contexts [11, 37]. For education specifically, AI used by teachers also raises trust and context questions. Studies of teachers' trust and AI acceptance show that willingness to use AI tools is shaped by trust, perceived usefulness, personal experience, technical equipment, and training or professional-development opportunities [4, 22]. More specific teacher-AI studies point to related factors, including pedagogical beliefs, perceived trust, perceived usefulness, ease of use, and the understandability of AI recommendations [7, 15, 39].

For this thesis, TAM provides conceptual background; the measurement evidence comes from a short SUS-style usability battery, use-intention and quality perceptions, and qualitative reflections on confidence and control.

2.7 Synthesis and Research Gap

The review connects broad literature and local discovery evidence to the prototype requirements. Table 2.1 summarises that connection. The Phase A column is grounded in the formative requirement elicitation in Section 4.1.1 and the later traceability analysis in Section 5.2.

The reviewed literature supports three points. First, differentiated planning is a persistent teacher challenge. Second, AI assistance can help when it reduces search and drafting effort without removing teacher control. Third, early-stage evaluation with indirect process and quality measures is defensible when claims are scoped carefully.

The remaining practical gap is how to design and evaluate AI planning support for differentiated homework learning paths.

This gap motivates the two-phase methodology used in this thesis. Phase A identifies grounded requirements from practitioner conversations and material review. Phase B tests those requirements in a controlled comparison of manual and AI-assisted planning.

Table 2.1: Synthesis from literature and discovery to thesis requirements.

Reviewed area	**What the literature establishes**	**What Phase A adds locally**
Differentiated instruction and scaffolding	Mixed-ability classrooms require tasks and supports that match learner readiness while preserving shared learning goals [27, 30, 33, 36].	Teachers described differentiation as a practical three-level planning problem: weak/expected/strong or low/medium/high task alternatives.
Teacher workflow and workload	Planning support is valuable when it reduces search, adaptation, and sequencing effort while preserving teacher judgement [12, 14, 35].	Teachers already had materials; the bottleneck was selecting, comparing, sequencing, and adapting usable tasks for a concrete class.
Local material and technology context	AI tools for teachers need contextual fit, trust, and adoption conditions that reflect the setting in which teachers work [4, 12, 22].	Textbook-style levelled exercises and QR-linked resources existed, and student-side technology logistics made planning support the coherent first scope.
AI copilots and teacher agency	AI planning tools can support drafting and ideation, with trust, verification, explainability, and automation bias as central risks [12, 14, 20, 22, 25].	Teachers wanted output that remains editable, inspectable, and subordinate to professional judgement.
Early-stage evaluation	Early-stage prototypes can be evaluated through process, usability, artefact quality, and practitioner feedback when the empirical focus is tool use and produced artefacts [1, 9, 17, 19, 34].	The study evaluates final homework learning paths, interaction traces, questionnaire responses, open-ended feedback, and interview-derived requirements.

Chapter 3

Technical Implementation

This chapter describes the concrete teacher-facing prototype evaluated in the study. It clarifies how the platform represented learning paths, how participants interacted with the editor, and how the AI planning loop received context, returned structured output, and validated changes before they affected the saved path.

The platform was implemented as a web application. In the task catalogue, both ALICE and Mathebattle are treated as task sources. ALICE refers to an existing TUM mathematics task source used here for fraction exercises, while Mathebattle contributes interactive mathematics tasks through its task environment and renderer [21, 29]. Inside the prototype, both are represented in the same way. Each task is a catalogue item with a stable reference such as `ALICE/...` or `MATHEBATTLE/...` denoted by `contentRef`. The editor therefore works with references to these tasks.

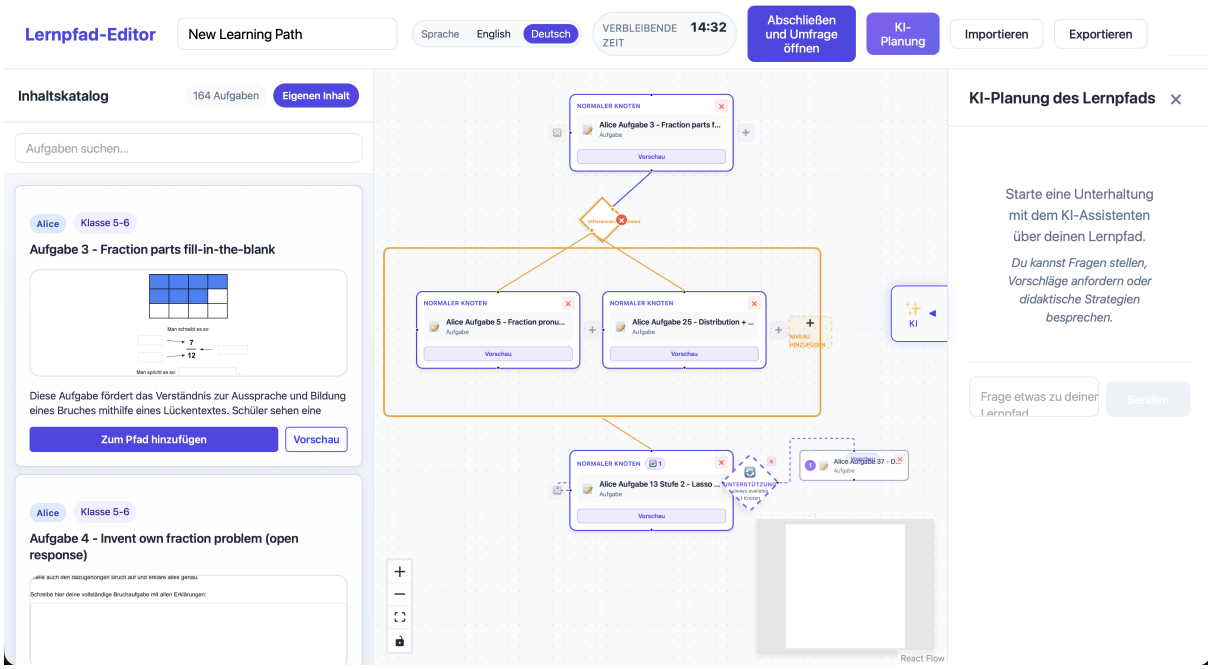


Figure 3.1: Participant-facing learning-path editor in the AI-assisted condition. The left panel contains the bounded task catalogue, the centre canvas visualises the learning path as a graph, and the right panel hosts the conversational AI planning interface. The header exposes study controls such as the timer, import/export, survey handoff, and AI planning access.

Figure 3.1 shows the editor as participants experienced it. Both study conditions shared the same catalogue, graph canvas, data model, timer, autosave, and survey handoff; only the `AI_assisted` condition exposed the conversational planning panel. The implementation therefore kept the condition difference concentrated in one additional AI-supported workflow.

3.1 System Architecture

The prototype uses a three-part architecture: a React/Vite browser client, an Express backend, and Supabase/Postgres for persistent study data. The AI assistant is implemented through the backend, so participant token checks, study condition checks, prompt construction, catalogue validation, and chat persistence remain server-side. Figure 3.2 summarises the main runtime boundaries.

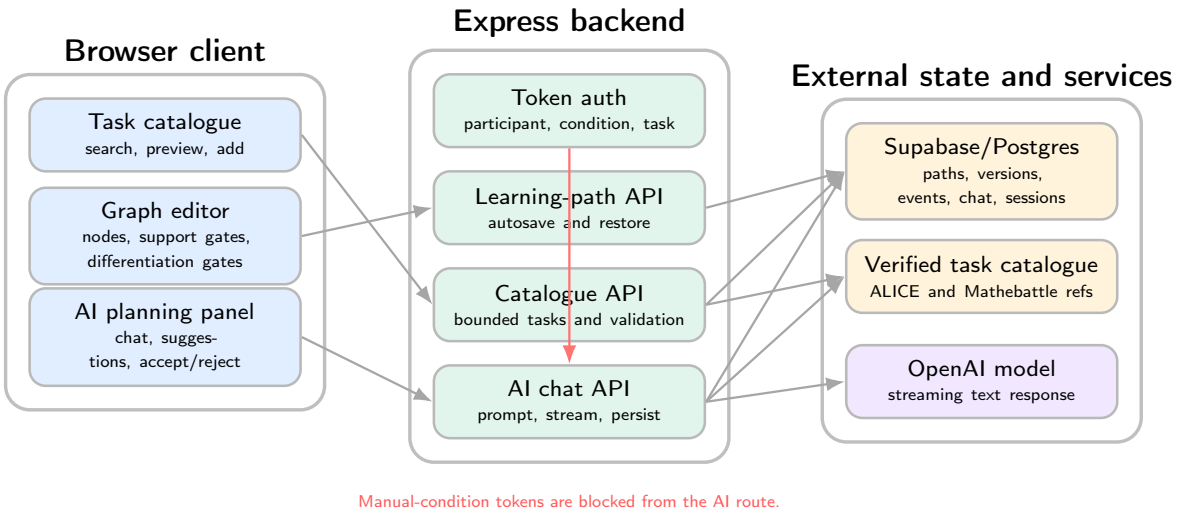


Figure 3.2: Runtime architecture of the evaluated prototype. The browser client owns the interactive editing experience, the Express backend protects study routes and constructs AI requests, and Supabase stores the path, session, telemetry, and chat data used in the evaluation.

The access token in each participant link identifies the participant pseudonym, assigned condition, and task. This token is used by the backend for every study-relevant route, including catalogue loading, path persistence, telemetry submission, session completion, and AI chat. The AI endpoint explicitly rejects manual-condition tokens. As a result, the same deployed application could serve both study conditions while keeping the AI feature condition-gated.

The backend also serves the verified task catalogue used by both conditions; participants could extend it with custom content in the editor.

3.2 Learning Path Representation

The central implementation object is a `LearningPath`. It contains metadata such as title and topic, and its `mainPath` is an ordered list of `NormalNode` objects. A `NormalNode` can contain a task directly. It can also act as a differentiation point: in that case, it has no task of its own, but stores options such as low, medium, and high, where each option points to the task for that branch. Figure 3.3 shows the schema used by the editor.

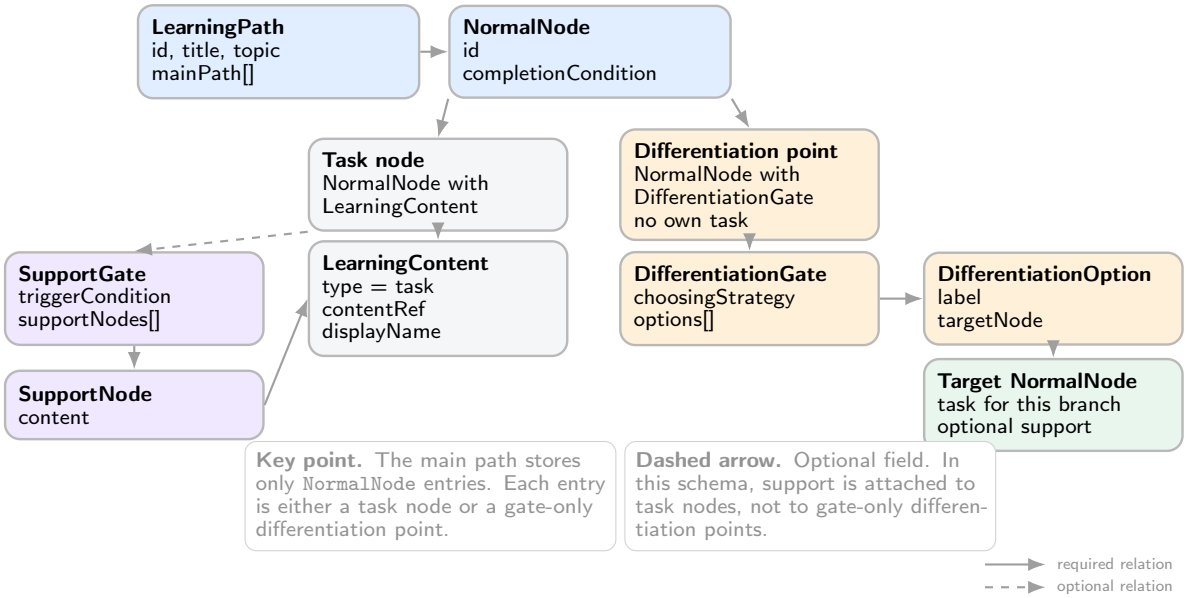


Figure 3.3: Simplified learning-path data model. The main path is an ordered list of `NormalNode` entries. A `NormalNode` is either a task node, which may have optional support, or a gate-only differentiation point whose options point to branch tasks.

The model follows a simple "nodes and gates" design. Task nodes hold the selected exercise reference and can have an optional support gate for additional help. Differentiation points do not hold an exercise themselves; they store labelled options such as low, medium, and high, and each option points to a branch task, which can also have its own support.

The saved path keeps only the stable `contentRef` and display name for each task; the actual ALICE or Mathebattle task is rendered from the catalogue. Because support and differentiation are saved as structured fields, the same JSON could later be used for artefact analysis.

3.3 Manual Editing Workflow

The manual editor is the foundation for both study conditions. Participants search or filter the task catalogue, preview tasks, and add selected tasks to the graph. The graph is rendered with React Flow, but the underlying state remains the `LearningPath` schema from Figure 3.3. React Flow is therefore only the canvas layer; the saved learning path remains the domain object described above.

Table 3.1 gives the implementation view of the main teacher operations. When a participant creates a differentiation gate or support gate, that interaction is saved and

Table 3.1: Manual editor operations and their representation in the data model.

Teacher action	Implementation effect	Study relevance
Add catalogue task	Creates a `NormalNode` with task `contentRef`.	Makes task choice observable and auditable.
Reorder path nodes	Reorders the `mainPath` array.	Makes task order explicit in the saved path.
Add support material	Creates or extends a `SupportGate`.	Represents scaffolding for struggling students.
Create differentiation gate	Converts a task node into a gate-only decision node and moves the existing task into the first option.	Makes three-level differentiation explicit.
Add differentiation option	Adds a labelled `DifferentiationOption` with a target node.	Supports low, expected, and high performance groups.

can be analysed later.

The editor autosaves the current path to the backend after changes. During the study, autosave was important since it made it possible to recover the final path in case of corruption.

3.4 AI Planning Workflow

The AI assistant is implemented as a chat panel inside the editor. When the teacher sends a message, the backend combines it with the current path and the available catalogue tasks. The model then returns either ordinary advice or structured artefacts that the user can inspect and apply. Figure 3.4 shows the main steps in one AI turn.

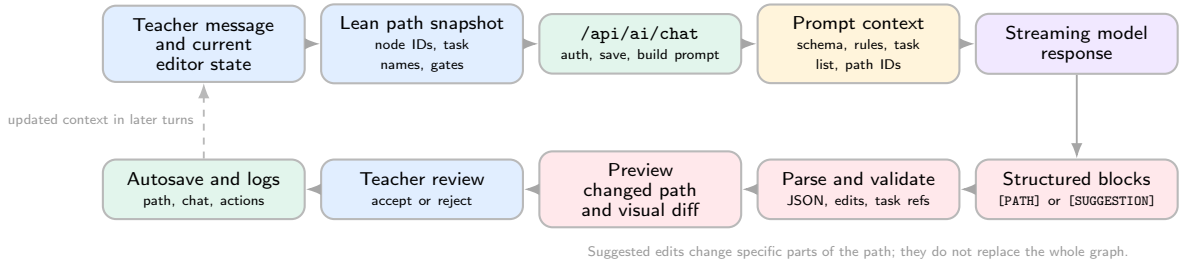


Figure 3.4: One AI turn in the evaluated prototype. The model receives the teacher message, a lean snapshot of the current path, schema rules, and available catalogue tasks. Structured responses are parsed into path artefacts or diff suggestions, but the teacher still decides whether to apply them.

The client sends the chat transcript to `/api/ai/chat` together with a lean snapshot

of the current learning path. It keeps only a compact path summary: path ID, title, topic, node IDs, task names, support-gate titles, support-node task names, and differentiation-option IDs and names. The purpose is to give the model enough context to refer to existing nodes by ID while avoiding a large request body.

The backend then builds a system prompt with four important parts. First, it describes the role of the assistant as an educational content designer for German mathematics teachers. Second, it specifies the learning-path structure and the schema for the path. Third, it lists available catalogue tasks and tells the model not to invent task IDs. Fourth, it adds the current path structure with node IDs so that suggestions can target the existing graph. The catalogue part is a compact metadata summary rather than the full task content: each task is represented by its stable `contentRef`, display name, short description, and, when available, associated knowledge-component identifiers. The assistant therefore reasons over catalogue metadata and must return known task references, which the editor later resolves back to renderable ALICE or Mathebattle tasks. Appendix A.4.1 summarises this prompt structure and the main limitations it implies for AI performance. The configured model identifier in the evaluated deployment was `gpt-5.4-nano-2026-03-17`. The model response is streamed back to the browser so participants see the answer progressively.

The assistant can produce two structured output types. A full `[PATH]` block contains a complete new learning path. A `[SUGGESTION]` block contains one or more diff operations to apply to the current path. The prompt instructs the model to use `[SUGGESTION]` blocks for modifications and reserve `[PATH]` blocks for entirely new paths. Each suggestion contains a short description and an `operations` array. This means that the model proposes a compact edit script over the `LearningPath` object. In the browser, this edit script is parsed, applied to a copy of the current path, and shown as a before/after change before the user applies it. Diff operations include adding or removing nodes, modifying nodes, adding or removing support gates, adding support nodes, and adding or removing differentiation gates. Table 3.2 gives the short version of this turn. The last column points to the matching rows in the appendix prompt-structure table where the prompt construction is documented in more detail.

When the browser receives an assistant message, it removes the structured marker blocks from the displayed prose, parses the JSON, checks that the operation types and

Table 3.2: Information flow and prompt construction in one AI planning turn.

Stage	**How it is used in the AI turn**	**Prompt detail**
Teacher request and current path	The browser sends the teacher message, chat transcript, and compact path snapshot to the backend. These become dynamic context for the model.	Appendix Table A.3, row "Current path context".
Stable prompt instructions	The backend adds the assistant role, catalogue grounding, path schema rules, and constraints against invented task references.	Appendix Table A.3, rows "Role and style", "Catalogue grounding", and "Path schema and invariants".
Structured response contract	The prompt tells the model when to return ordinary advice, a complete `[PATH]`, or an editable `[SUGGESTION]` block.	Appendix Table A.3, rows "Structured output" and "Rationale".
Browser review state	The browser parses any structured block into a candidate path or before/after diff, then shows it as a review card and graph highlights.	Not part of prompt construction; handled after the response.
Stored study evidence	Chat messages, path records, and telemetry events are persisted for later process and artefact analysis.	Not part of prompt construction; used in the evaluation.

required fields are present, and applies the operations to a copy of the current path. The resulting after-state is compared with the before-state by node IDs. This produces counts of added, removed, and modified nodes, plus the underlying before/after paths used for review. The graph overlay renders candidate after-state elements as non-editable preview nodes and adds status classes to existing nodes. This is why participants see a review card and graph highlights rather than raw JSON.

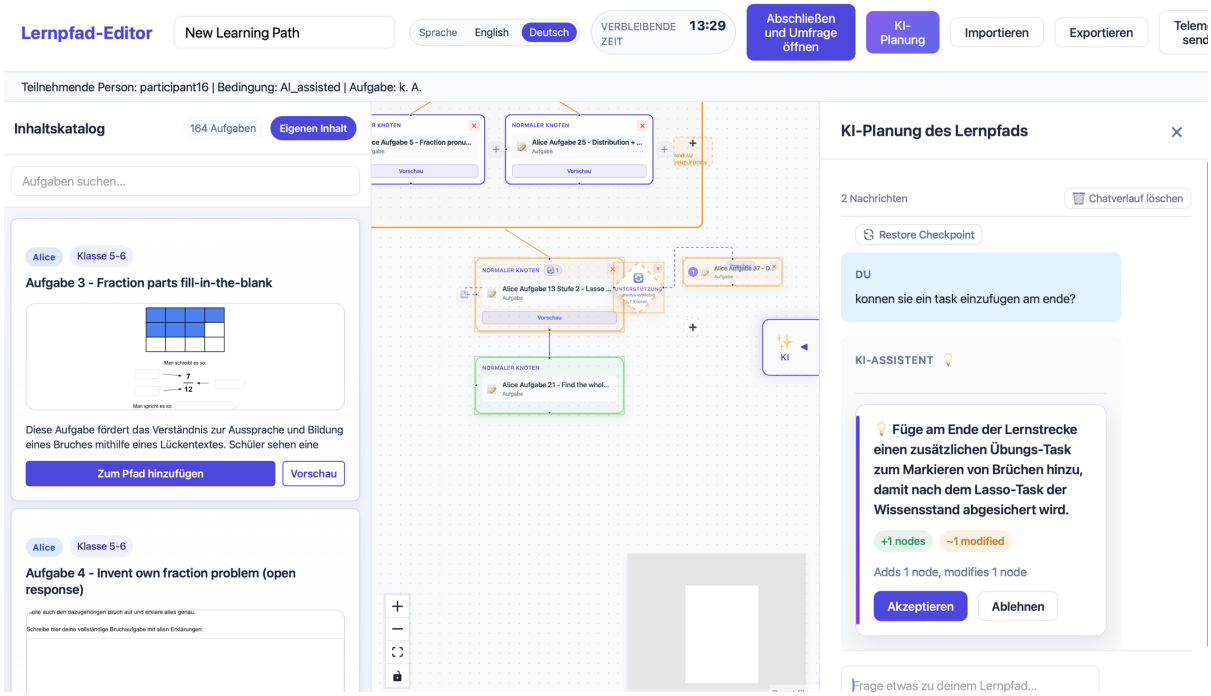


Figure 3.5: AI suggestion review in the evaluated prototype. The chat card summarises the parsed diff and provides accept/reject controls. On the canvas, green highlights show proposed added graph elements and orange highlights show modified elements; these overlays preview the candidate after-state before it becomes the saved editor state.

Figure 3.5 shows the review state created from one parsed suggestion. Only after the participant accepts a suggestion does the changed path become the editor state. Accepting, rejecting, applying, and normalising suggestions are logged as study events, which later made it possible to interpret where the AI workflow helped and where it shifted work into inspection and repair.

Several guardrails were implemented around this workflow. Generated task references are normalised and checked against the verified catalogue reference set. If a suggestion introduces a new task reference outside the catalogue, it is rejected rather than applied. The diff applicator also repairs a specific structural problem that occurred in AI outputs: if a main-path node incorrectly contains both task content and a differentiation gate, the content is moved into an option and the parent is normalised into a gate-only node. This normalisation is logged because it is both a usability support and evidence that the assistant produced structurally imperfect output.

The chat panel also stores local checkpoints. Before a user sends a message, the current path is copied into a checkpoint. If the conversation later leads to an undesirable edit, the user can restore an earlier path and trim the chat history to that point.

3.5 Persistence and Study Instrumentation

The platform records study-relevant data directly during use, including saved paths, telemetry events, chat messages, and completion records. The backend links these records, together with version records and survey handoff payloads, to a pseudonymous participant token.

Telemetry is collected through a client-side event bus. Components emit structured events for actions such as catalogue search, adding a node, creating a differentiation gate, accepting or rejecting an AI suggestion, timer events, and session submission. The event bus batches events and periodically flushes them to the backend.

When participants finish or the timer expires, the app creates a session-completion record and builds a survey URL with hidden fields for token, participant pseudonym, condition, task, and language. This keeps the handoff deterministic while preserving pseudonymous linkage between the platform session and the later survey response.

3.6 Implementation Trade-Offs

The implementation makes several trade-offs that are important for interpreting the study. First, the AI component is prompt-based without extensive experimentation. The results therefore evaluate one deployed prompting workflow and model configuration, not the upper bound of what a stronger model, retrieval system, or constrained generator could achieve.

Second, the catalogue grounding is pragmatic rather than perfect. The prompt lists available catalogue tasks and the application validates task references, but the model can still produce invalid or awkward suggestions that require rejection, normalisation, or repair. This limitation became visible in the results and is part of what the prototype evaluation revealed.

Third, the graph model is structured but not fully pedagogically constrained. The editor can represent support and differentiation, but it does not force every path to contain exactly three performance-group branches. The platform made differentiation visible and editable, but it did not yet enforce all task-specific pedagogical requirements when using AI.

The next chapter evaluates that implemented planning platform under controlled study

conditions.

Chapter 4

Methods and Evaluation

This thesis uses a two-phase Design-Based Research (DBR) methodology. Phase A is formative: it identifies the practical constraints that teachers and education stakeholders associate with differentiated mathematics planning. Phase B is evaluative: it compares manual and AI-assisted use of the learning-path editor in a controlled teacher-side planning task. The combined design keeps the evaluation close to the context that motivated the prototype. The study asks whether the tool helps novice teachers find suitable tasks, build visible differentiation, stay in control, and work within realistic planning constraints. Figure 4.1 summarises this two-phase structure and shows how Phase A requirements feed into Phase B evidence streams.

The study asks a narrow question: how did participants experience the planning workflow, what did they do in the editor, what final homework-learning paths did they produce, and what did they report afterward? In evaluation terms, these are early-stage process and artefact measures. Student learning gains, classroom implementation, long-term adoption, and model comparisons are treated as future research targets. This choice is consistent with Cook and Ellaway's technology-enhanced learning framework, which includes needs analysis, process and product documentation, usability testing, participant experience, learning outcomes, and sustainability as evaluation activities [9]. It is also consistent with intelligent-tutoring-system evaluation work that treats system behaviour, learning curves, educational data mining, and learning analytics as legitimate evaluation evidence [17]. The DBR framing is important for interpretation because it treats prototype development and evaluation as practice-facing, iterative inquiry [1, 34]. Accordingly, the analysis treats the implemented prototype's usability, reference quality, and need for

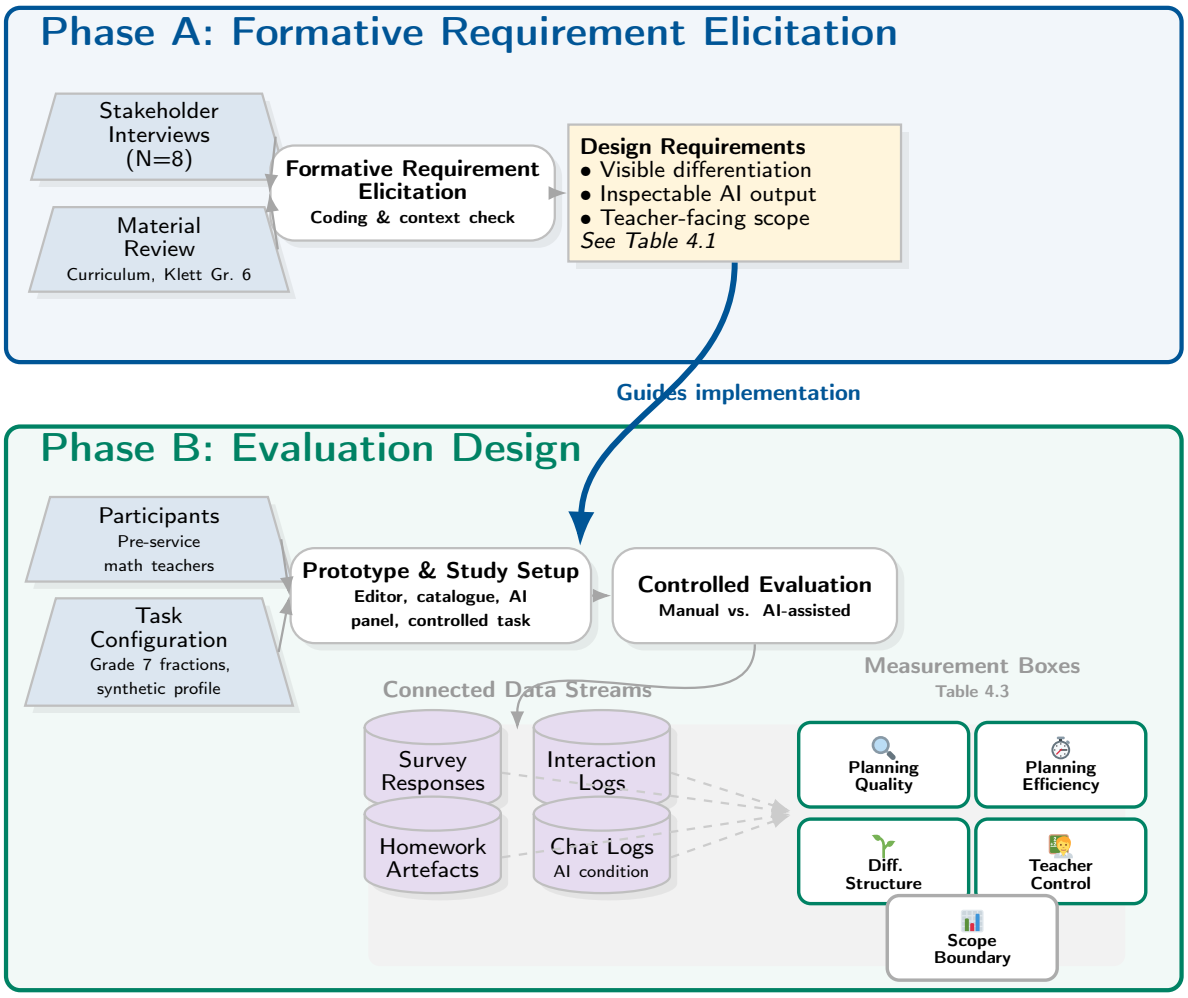


Figure 4.1: Input-output structure of the two-phase methodology. Phase A converts contextual evidence into requirements; Phase B evaluates the resulting planning workflow through connected evidence streams. The Phase B output cues correspond to the measurement boxes in Table 4.3.

facilitator repair as evidence about its current design readiness.

4.1 Two-Phase Research Design

4.1.1 Phase A: Formative Requirement Elicitation

Phase A identifies the planning problem before treating AI as a solution. Its evidence base consists of semi-structured interviews with teachers, teacher trainees or former teachers, and education stakeholders with experience in lesson planning, differentiation, assessment, or classroom technology. The interviews focus on mixed-ability classrooms, existing approaches to differentiation, the practical cost of finding and adapting tasks, the materials teachers already use, and the circumstances under which AI support would or would not be acceptable. The formative corpus consists of eight conversation notes and short synthesis memos. Because the roles vary and the phase is exploratory, the corpus is used to identify design constraints rather than to estimate prevalence or claim thematic saturation. Interview notes and synthesis memos were analysed through a single-researcher descriptive coding pass followed by pattern coding into design requirements.

A source-by-source overview of the formative corpus is preserved in Appendix A.1. The main text focuses on how the formative evidence was converted into prototype and evaluation requirements.

The analysis was conducted as a single-researcher formative coding process. First, each note was read for descriptive codes such as task discovery, differentiation levels, material availability, teacher control, student-device logistics, assessment feedback, and diagnostic visibility. Second, related codes were grouped into pattern codes that could become prototype requirements. To check the coding, the resulting requirement set was compared against the original notes, available transcript excerpts, and the curriculum/textbook material review. This check was used to avoid basing requirements on isolated comments when other sources contradicted them. Phase A provides design constraints for the prototype and evaluation.

The requirement set was also checked against curriculum and textbook-related material, including Baden-Wuerttemberg curriculum summaries and examples of common mathematics textbooks such as current Grade 6 Klett materials [13]. The material review serves as a contextual check. It examines current textbook-based differentiation, such as

levelled or column-based exercises and QR-linked resources. The resulting interpretation shapes the evaluation: the problem is not framed as a lack of educational content, but as a planning, selection, sequencing, and adaptation problem. Table 4.1 summarises how the formative evidence is converted into prototype and evaluation requirements.

Table 4.1: Formative Phase A findings and resulting design requirements.

Finding from interviews and material review	**Interpretation for differentiated planning**	**Requirement carried into the prototype/evaluation**
Existing materials already contain some differentiation, for example levelled textbook exercises and QR-linked resources.	The problem is selecting, comparing, sequencing, and adapting usable tasks for a specific class.	Evaluate task discovery, catalogue use, task references, and whether final paths remain grounded in available material.
Teachers often reason in practical levels such as basic/intermediate/advanced or weak/expected/strong.	Differentiation needs to be visible as parallel alternatives, not only as extra support material.	Use the task-specified three-group homework structure and audit exact three-option differentiation gates.
Teachers want support but not opaque replacement of professional judgement.	AI output must remain inspectable, editable, and subordinate to the teacher's pedagogical decision.	Treat agency, usability, chat repair, suggestion application, and invalid references as central evaluation evidence.
Student-side technology creates classroom logistics: devices, batteries, internet, input difficulty, and access inequality.	A teacher-facing planning tool is a realistic first step before attempting a student-facing adaptive tutor.	Keep the evaluated prototype teacher-facing and use synthetic class profiles.
Teachers value evidence about student understanding, especially from tests, observations, and progress dashboards.	Planning and diagnosis are connected in practice, but implementing the full loop is larger than this thesis.	Acknowledge diagnostic visibility as an unmet requirement and future-work direction, not as a claim of the current prototype.

4.1.2 Phase B: Evaluation Design

Phase B translates these requirements into a controlled between-subjects planning study. Its purpose is to evaluate the resulting planning workflow under comparable conditions. Each participant is assigned to one of two conditions. In the manual condition (`A_manual`), participants use the learning-path platform and task catalogue without conversational AI support. In the AI-assisted condition (`AI_assisted`), participants use the same platform, task catalogue, and planning task with a conversational AI assistant enabled. The comparison isolates the participant-visible availability of the conversational AI planning workflow while keeping the base editor, content pool, and task constant. The phase produces four connected evidence streams: questionnaire responses, interaction logs, final homework-learning-path artefacts, and qualitative feedback, which are integrated in the analysis strategy described in Section 4.5. The study is exploratory: with a small sample,

the aim is to identify patterns, design mechanisms, and validity threats.

4.2 Prototype and Study Conditions

The implemented prototype is described in Chapter 3. For the evaluation design, the important point is that both study conditions used the same teacher-facing learning-path editor, task catalogue, timer, autosave, and submission flow. The editor let participants assemble a homework learning path from content nodes, differentiation gates, and support gates. A differentiation gate is a branching point where students are assigned to different tasks according to performance group; in the study task, each such gate was expected to contain exactly one task for each of the three groups. A support gate is an optional help branch attached to a task, intended for students who need an explanation or extra practice before returning to the main path.

The manual condition (`A_manual`) used this base editor without access to the AI planning route. The AI-assisted condition (`AI_assisted`) used the same editor with an additional conversational planning panel. The panel could propose tasks, path structures, and edits that participants could inspect and apply. The AI implementation was prompt-based, without extensive experimentation or model comparison. Consequently, the results should be read as evidence about this implemented prototype and configured deployment, not as evidence about the best possible AI planner. Additional interface screenshots are provided in Appendix A.6.

4.3 Participants, Procedure, and Data Sources

The Phase B participants were pre-service secondary mathematics teachers and teacher trainees. This population was appropriate because participants were actively learning to plan mathematics lessons and differentiated materials, and the prototype is partly aimed at supporting novice teachers [14, 31]. Recruitment began with an interest form sent to mathematics teacher-education students. After duplicate submissions were removed, the anonymised recruitment pool contained 24 unique registrations; 21 registered people were later invited to try the demo. All 24 unique registrants selected German as a comfortable language, so the participant-facing study flow used German for the familiarisation docu-

ment, task text, platform interface, and post-condition questionnaire. Full task wording and instrument details are documented in Appendix A.2 and Appendix A.3.

Fourteen participants completed the study and post-condition survey. One participant, participant20, was later identified as a protocol deviation because the participant had seen both study conditions despite the planned between-subjects design. The between-subjects analysis therefore excludes participant20 and uses 13 participants: 6 in the manual condition and 7 in the AI-assisted condition. Debug, pilot, and incomplete session records were excluded from the condition comparisons. Fine-grained recruitment, linkage, and planned-versus-realised measurement details are preserved in Appendix A.4.

4.3.1 Task and Procedure

All participants completed the same planning task on Grade 7 fraction consolidation. The task asked participants to construct a differentiated homework learning path for a heterogeneous Gymnasium-level class in Baden-Wuerttemberg. The homework had to be feasible in approximately 45 minutes and focus on equivalence, expanding, simplifying, and applying fractions as preparation for rational numbers. Participants were instructed to use the predefined catalogue of ALICE and Mathebattle tasks, to include differentiation nodes, and to assign exactly one task per performance group at each differentiation node. They could also add support materials, such as short explanations, links, or videos, to assist students who failed to complete a corresponding task.

The synthetic class profile contained three performance groups: high-performing students, students at the expected level, and students below the expected level. Participants first received a familiarisation document, then worked within a short fixed platform window before the demo timer prompted them to submit their path and continue to the post-condition survey. The full study flow also included consent, a post-condition questionnaire, short SUS-style usability items, and open-ended reflections.

4.3.2 Data Sources and Instrumentation

The analysis combines four data streams. The first is self-report data from the post-condition Tally survey, including Likert responses, short usability items, and open-ended written responses. The second is artefact data from the selected final homework learning paths stored in Supabase. The third is interaction data from logged user actions, session

events, and catalogue-search behaviour. The fourth is AI-condition conversational data from chat messages between participants and the assistant. Table 4.2 summarises the source and analytical role of each stream.

Table 4.2: Data streams used in the mixed-methods analysis.

Data stream	**Source**	**Analytical role**
Survey responses	Latest completed Tally submissions	Perceived quality, task discovery, usability, time pressure, and open-ended reflections
Homework learning paths	Supabase learning-path records linked to participant tokens and sessions	Structural and exploratory artefact audits of final paths
Interaction logs	Supabase session and user-action records	Process evidence, including search, node creation, gate creation, and AI suggestion application
Chat logs	Supabase chat-message records in the AI condition	Qualitative evidence for steering, repair, validation, and AI-control issues

The logging pipeline was designed around pseudonymous participant tokens. Each participant received a study link associated with a token, assigned condition, task, and language context. When the participant completed the planning task, the application stored a session-completion record and the selected homework learning-path identifier. The survey link carried hidden pseudonymous fields so that survey responses could be linked to the corresponding session and condition without using direct personal identifiers. The raw artefacts were preserved, and all analysis outputs were generated as derived files so that the original study data remained auditable.

4.4 Measures and Hypotheses

The measures turn the primary research question into four observable questions: what participants thought about the produced path, how manageable planning felt, what differentiation structure appeared in the final path, and how much control participants kept. A fifth box is included as a scope boundary because diagnostic visibility into student needs was identified in Phase A but not implemented in this prototype. The easiest way to read the map is therefore as four measurement boxes plus one explicit boundary, rather than as a long list of variables. Table 4.3 gives the map; Appendix A.4 documents the full

metric inventory and implementation caveats.

Table 4.3: Measurement map for the Phase B analysis.

Cue	Box	Reader question	Main Phase B evidence
🔍	Planning quality	Did participants think the produced homework path fit the task?	Task-goal fit, overall quality, differentiation quality, curriculum fit, use intention, and open-ended quality comments.
⏱️	Planning efficiency	Did the workflow make planning feel easier or more manageable?	Task-discovery efficiency, perceived time sufficiency, catalogue-search logs, construction-action counts, AI steering effort, and time-pressure comments.
🌱	Differentiation structure	Did the final path show the required three-group homework structure?	Differentiation gates, exact three-option gates, difficulty spread, target fraction-skill coverage, support-gate placement, duplicates, and invalid task references.
👩‍🏫	Teacher control	Did participants stay in control of the plan, especially with AI support?	Agency and usability items, AI accept/apply events, AI-suggestion normalisation events, chat turns, chat repair evidence, and comments about steering or editing AI output.
📊	Scope boundary	Which important teacher need was not implemented in this prototype?	Diagnostic visibility into student needs was retained as future work; no real student data, assessment data, or progress-dashboard evidence was collected.

The four measurement boxes should be read together. None of them proves pedagogical quality on its own. Because the study is scoped to teacher-side planning rather than student learning outcomes, they are interpreted as early-stage indicators of whether the workflow is useful, understandable, and controllable. Efficiency does not mean that one group simply finished faster. All participants worked within the same short planning window, and some log timestamps included repeated visits or stale path states. For that reason, efficiency is read through perceived time sufficiency, interaction counts, and comments about time pressure.

The following directional expectations guided the analysis:

- **Expectation 1 (Perceived Quality):** Participants in the AI-assisted condition report higher perceived task-goal fit and overall homework-learning-path quality than participants in the manual condition.

- **Expectation 2 (Efficiency under time constraint):** Participants in the AI-assisted condition experience task discovery and construction as less effortful than participants in the manual condition within the same fixed planning window, reflected in perceived task-discovery efficiency, perceived time sufficiency, and lower interaction burden.

4.5 Analysis Strategy and Evaluation Boundaries

The analysis followed a simple chain from Phase A to Phase B: Phase A defined design requirements, and Phase B checked whether those requirements appeared in participant experience and final artefacts. Survey responses, interaction logs, path structures, chat evidence, and open-ended reflections were interpreted as different views of the same planning problem. In methodological terms, this is a convergent mixed-methods design [10]: quantitative results established the key patterns, while written responses, chat evidence, and artefact inspection helped explain those patterns.

Survey responses were summarised by condition using means, medians, and AI-minus-manual differences. Given the small sample size and exploratory nature of the study, the reporting emphasises descriptive patterns and practical interpretation over binary significance claims. For exploratory robustness checks, continuous and Likert outcomes were compared using exact rank-permutation Mann–Whitney tests based on the observed ranks. Welch's t-test was used only as a sensitivity check. Binary audit and open-response indicators were compared using Fisher's exact test. Because no multiple-comparison correction was applied, p-values are interpreted as exploratory signals. They help scope the descriptive and mixed-methods patterns, but they are not treated as confirmatory evidence. Time-related evidence is interpreted together with perceived time sufficiency, action logs, and qualitative comments rather than as a standalone efficiency outcome.

The final homework learning paths were analysed in three layers. First, the analysis counted visible structural elements, including content nodes, differentiation gates, differentiation options, support gates, and support nodes. Second, each path was checked against criteria drawn from Phase A and the study task: at least one exact three-option differentiation gate, low/medium/high difficulty spread, coverage of target fraction-skill buckets (equivalence, expanding, simplifying, comparison, and application), absence of

invalid catalogue references, bounded path length relative to the 45-minute homework target, and duplicate task references. Third, an exploratory sequence and branch-design audit examined common starts, broad visual-to-symbolic ordering, branch-level challenge and application tasks, support attached to differentiated branch tasks, and long or duplicate-heavy paths. These measures describe visible properties of the submitted paths; they do not replace independent expert ratings.

Open-ended responses were analysed using a keyword-assisted coding pass followed by inspection of representative excerpts. The code set was derived from the formative interviews and from issues that appeared during the analysis, such as metadata needs, onboarding, language mismatch, time pressure, and AI steering or recovery. AI-condition chat logs were analysed qualitatively after exact deduplication to identify mechanisms such as invalid catalogue references, missing three-level differentiation, unexpected branch counts, stale path state, and difficulty applying or editing AI-generated suggestions.

4.5.1 Ethics and Data Protection

The study used pseudonymous participant identifiers and tokenised access links. No real student data was collected; the planning task used a synthetic class profile. Participants were informed that their interaction traces, chat messages in the AI condition, final homework learning paths, and survey responses would be analysed for research purposes. Data handling followed DSGVO/GDPR principles by separating direct personal identifiers from analysis data and by storing study records under pseudonymous keys.

4.5.2 Methodological Limitations

Several methodological limits shape the interpretation of the results. The sample is small and consists primarily of novice or pre-service teachers, so the findings are best understood as formative evidence for design iteration. The study uses one mathematics topic and one bounded planning task. This improves comparability, but it limits generalisation to other topics or classroom contexts. The artefact audits are based on observable proxies rather than independent expert ratings. The AI-assisted condition also included rough interactions and known demo problems, so the study cannot fully separate whether weak AI-condition results came from the design idea, from the buggy version of the demo, or from both. Finally, the usability battery is SUS-style but not a complete validated SUS

instrument. These limitations are addressed in the results and discussion by treating the findings as triangulated design evidence rather than as definitive proof of AI effectiveness.

Chapter 5

Results

The results are easiest to read as a design diagnosis for teacher-facing AI tools. AI assistance changed the kind of planning work participants did, but it did not make the final homework-learning-path artefacts clearer in this prototype. It helped some participants with ideation and task discovery, yet the clearest artefact-level signal ran in the other direction: all six manual paths contained at least one exact three-option differentiation gate, compared with three of seven AI-assisted paths.

The measured difference is a shift in structure and work: fewer explicit differentiation gates in the final AI-assisted paths, more logged suggestion mediation, and more participant reports of AI steering or recovery. For a tool builder, the practical implication is that AI support needs better-enforced catalogue grounding, branch-level differentiation constraints, and inspectable apply/edit controls before its ideation benefits can plausibly improve final paths.

5.1 Analysis Cohort and Reading Guide

The between-subjects analysis uses the completed Phase B cohort after excluding protocol deviations from the condition comparison. Fourteen participants completed the post-condition survey. Participant20 was later identified as a crossover exposure case because the participant had seen both conditions despite the intended between-subjects design. Participant20 is therefore excluded from the condition comparisons; its exploratory boundary case is documented in Appendix A.5.2. The main analysis cohort contains 13 participants: 6 in the manual condition (`A_manual`) and 7 in the AI-assisted condition

(`AI_assisted`). Table 5.1 is included as a coverage check rather than as a substantive result: it shows that every retained participant contributed survey, final-path, and interaction evidence, while conversational evidence exists only for the AI-assisted condition. Debug, pilot, incomplete, and other non-analysis records were excluded.

Table 5.1: Coverage of evidence streams used in the main comparison.

Evidence stream	**Manual**	**AI-assisted**
Participants in between-subjects analysis cohort	6	7
Completed post-condition surveys	6	7
Selected final homework learning paths	6	7
Logged interaction actions	204	489
AI chat evidence (messages / participant turns)	–	104 / 52

Table 5.2 gives the shortest reading of the chapter. The detailed sections that follow unpack each row with survey, artefact, process, chat, and open-response evidence.

Table 5.2: Results at a glance for a teacher-tool reader.

Question	**Phase B result**	**Evidence**	**Design action**
Did AI improve perceived path quality?	No clear improvement; AI was lower on differentiation quality, curriculum fit, and willingness to use the path.	Post-condition survey means and usability battery.	Treat AI output as draft material that must be inspectable, not as finished planning.
Did AI reduce planning burden?	Only a small task-discovery signal appeared; time sufficiency, learnability, and usability moved against AI.	Task-discovery item, time-sufficiency item, usability items, action logs.	Search support is useful, but it must not add a larger validation and control burden.
Did final artefacts show practical differentiation?	Manual paths made the task-specified three-group structure more visible.	Differentiation gates, exact-three gates, path audit, sequence audit.	Generate exactly three levelled branches by default and make missing differentiation visible.
What work did AI change?	Direct construction decreased, but AI steering, applying, normalising, and repair became central.	Logged actions, chat turns, chat excerpts, open responses.	Build separate branch-level accept, edit, and reject controls, plus undo and clear path-state handling.
What did participants still need?	Better metadata, filters, difficulty labels, onboarding, reordering, and pedagogical rationale remained important.	Open-response design codes.	Improve the base planning workflow as well as the AI layer.

The rest of the chapter follows this reader logic. Section 5.2 explains the interview-derived lens used to interpret the evidence. Section 5.3 reports perceived quality, efficiency, and usability. Section 5.4 examines the final homework learning paths. Section 5.5 shows the process mechanism through interaction and chat evidence. Section 5.6 turns participant comments into design needs. Section 5.7 reports robustness checks and evaluation boundaries.

5.2 Interview-Derived Analysis Lens

The Phase B results were interpreted using the teacher needs identified in Phase A, not only generic AI-performance metrics. This matters because the interviews showed that the core design problem was not simply whether AI can generate a plausible homework learning path. Teachers described a more concrete problem: finding usable tasks under time pressure, making practical three-level differentiation, keeping control over generated material, avoiding student-side device logistics, and eventually gaining better visibility into student needs. The full traceability table is preserved as Table A.2 in Appendix A.1.

This link back to Phase A changes how the homework learning-path artefacts are interpreted. A path is not better simply because it has more nodes. From the interviews, a useful path should be grounded in available material, feasible for teacher use, visibly differentiated into meaningful levels, and still editable by the teacher. The exact three-option differentiation gate is therefore treated as a visible requirement, not merely as a technical count: it shows whether the path gives the three performance groups a concrete branching point in the homework design.

5.3 Perceived Quality and Planning Burden

Table 5.3 summarises the central post-condition Likert items. Scores range from 1 to 5. For positively worded items, higher values indicate a more favourable response. For the two negative usability items, "unnecessarily complex" and "had to learn a lot", higher values indicate a less favourable response. The full Likert item wording is documented in Appendix A.3; Table 5.3 uses shortened labels for readability.

The questionnaire does not support the simple hypothesis that AI assistance improved perceived homework-learning-path quality. Task-goal fit was essentially identical across conditions (manual $M = 3.33$, AI $M = 3.43$), and overall quality was also similar but low (manual $M = 2.50$, AI $M = 2.29$). The AI condition was lower on differentiation quality ($\Delta = -1.00$), curriculum fit ($\Delta = -0.67$), and willingness to use the created homework assignment ($\Delta = -0.71$). Thus, the first expectation is not supported descriptively in this dataset.

The efficiency picture is mixed in a way that is important for teacher-facing AI design. AI participants reported only slightly better task discovery efficiency ($M = 2.71$ versus

Table 5.3: Post-condition questionnaire means.

Item	Manual M	AI M	Manual Md	AI Md	Δ AI–Manual	Mean profile
Perceived homework quality						
Task-goal fit	3.33	3.43	3.5	3.0	0.10	
Overall quality	2.50	2.29	2.0	2.0	-0.21	
Differentiation quality	4.00	3.00	4.5	3.0	-1.00	
Would use homework	3.00	2.29	3.0	3.0	-0.71	
Curriculum fit	3.67	3.00	4.0	3.0	-0.67	
Planning efficiency under time constraint						
Task discovery efficiency	2.67	2.71	2.0	3.0	0.05	
Parallel groups	3.50	2.43	3.5	3.0	-1.07	
Enough time	2.17	1.14	1.5	1.0	-1.02	
Exploratory usability battery						
Easy to use [SUS]	3.67	2.00	4.0	1.0	-1.67	
Felt confident [SUS]	3.00	1.71	3.0	1.0	-1.29	
Unnecessarily complex [SUS]	1.50	2.86	1.0	3.0	1.36	
Functions integrated [SUS]	3.83	2.57	4.0	2.0	-1.26	
Had to learn a lot [SUS]	1.83	4.00	1.5	4.0	2.17	

Note. All item scores use the common 1–5 Likert scale. The [SUS] tag marks System Usability Scale (SUS)-style usability items. Delta-cell shading is an interpretive aid, not a significance marker: expected direction, against expectation, and exploratory item with no formal directional hypothesis.

$M = 2.67$), but they also reported lower perceived time sufficiency ($M = 1.14$ versus $M = 2.17$) and substantially worse usability. The planning-time event spans were not used as a primary outcome because some sessions include repeated visits, stale path states, or cross-session timer events. For that reason, the second expectation is not supported overall. Only the task-discovery subcomponent points slightly in the expected direction, while perceived time sufficiency and usability point against a reduction in experienced planning burden. At the questionnaire level, the core pattern is already visible: a small search benefit is paired with a larger control and learnability cost.

5.3.1 Usability Battery

The implemented Tally form did not contain the full 10-item System Usability Scale. Therefore, this subsection does not report a valid SUS score. Instead, the available usability evidence is treated as a short, non-validated usability battery. On these items, the AI condition was clearly more difficult for participants: AI participants rated the system as less easy to use ($M = 2.00$ versus $M = 3.67$), reported lower confidence ($M = 1.71$ versus $M = 3.00$), rated the system as more unnecessarily complex ($M = 2.86$ versus

$M = 1.50$), and agreed more strongly that they had to learn a lot before getting started ($M = 4.00$ versus $M = 1.83$). This usability result is especially informative because both conditions shared the same base editor and catalogue. The additional AI layer appears to have introduced interaction and control costs rather than simply reducing work.

5.4 Final Homework-Learning-Path Artefacts

The final homework-learning paths make the survey results easier to interpret. AI participants did not only rate the AI condition lower; their finished paths also showed less visible differentiation. Both conditions produced paths of similar top-level length, but the manual paths more often showed the required three-group structure.

Table 5.4 reports descriptive counts for the selected final paths. These counts are not expert quality scores. They show how much participants built, how often they used differentiation and support structures, and whether differentiation gates followed the task requirement of exactly three options. Top-level nodes count only the direct main-path nodes; recursive path nodes count nodes reached by traversing into differentiation branches as well.

Table 5.4: Structural metrics of selected final homework learning paths.

Metric	Manual	AI-assisted	Count profile
Mean count per final path			
Top-level nodes	6.83	6.57	
Recursive path nodes	15.33	11.57	
Differentiation gates	3.00	1.57	
Differentiation options	8.50	5.00	
Support gates	0.83	1.29	
Support nodes	1.00	1.43	
Invalid three-option gates	0.83	0.86	

Note. Visual profiles show manual (gray, top) and AI-assisted (blue, bottom) condition means on a common 0–16 count axis. The profiles make the same pattern visible as the numeric means: AI-assisted paths have similar top-level length but fewer differentiation gates/options and more support material on average.

The most visible structural difference is not total path length. AI-assisted paths had a similar number of top-level nodes to manual paths (6.57 versus 6.83), but fewer recursive path nodes overall (11.57 versus 15.33). More importantly for the study task, manual paths contained more explicit differentiation structure: 3.00 differentiation gates and 8.50 differentiation options on average, compared with 1.57 gates and 5.00 options in

the AI-assisted condition. AI-assisted paths, by contrast, contained more support-gate structure and support nodes.

This pattern suggests that the AI layer did not simply make participants produce more differentiated homework learning paths. Instead, it appears to have shifted some structure toward remedial support or generated support material. That can be pedagogically useful, but it is not identical to the study requirement: differentiated homework should route students into parallel ability-appropriate tasks through differentiation gates. This finding is consistent with the questionnaire result in which AI participants rated differentiation quality lower than manual participants.

5.4.1 Interview-Derived Path Audit

To connect the path analysis back to the interviews, each final path was checked for visible features teachers had said mattered. This check did not replace expert judgement. It asked a simpler question: did the submitted path show the practical qualities teachers had highlighted and the study task specified? Those qualities were exact three-option differentiation for the three performance groups, low/medium/high task-difficulty spread, coverage of the required fraction skills, catalogue grounding, and bounded length for the study homework task. Appendix A.4.2 documents how the table values were calculated, including the five audit-score criteria and the target fraction-skill buckets used for coverage. Table 5.5 reports the condition-level summary of this interview-derived audit.

This audit reinforces and sharpens the earlier structural result. All manual paths contained at least one exact three-option differentiation gate, compared with three of seven AI-assisted paths. Manual paths also more often showed a low/medium/high difficulty spread and broad target fraction-skill coverage. The AI-assisted paths were more bounded in length, but they had substantially more duplicate task references on average (3.00 versus 0.67), suggesting that AI-supported construction sometimes repeated tasks rather than diversifying the path. One AI path contained an invalid catalogue reference, matching the chat evidence in which a participant reported that the generated path used a task outside the allowed catalogue.

Table 5.5: Interview-derived path audit summary by condition.

Audit criterion	Manual	AI-assisted	Profile
Bounded count summaries (0 = none, 5 = all)			
Mean audit criteria met	4.67	3.71	
Mean target fraction-skill buckets covered	3.67	3.43	
Open-ended duplicate-count summary (lower is better)			
Mean duplicate task references per path	0.67	3.00	
Paths meeting binary audit criterion (%)			
At least one exact three-option differentiation gate	100%	42.9%	
Low, medium, and high difficulty tasks	83.3%	71.4%	
At least three of five target fraction-skill buckets	100%	71.4%	
No invalid task-catalogue references	100%	85.7%	
Within the 45-minute homework length proxy	83.3%	100%	

Note. The criteria translate Phase A concerns into observable properties of the final learning paths. Visual profiles show manual (gray, top) and AI-assisted (blue, bottom), matching the coloured column headers. The first two mean rows are bounded 0–5 counts. The duplicate-reference row uses its own open-ended count axis; the arrow indicates that larger values are possible. Percentage rows use separate 0–100 bars.

5.4.2 Exploratory Sequence and Branch-Design Audit

Because the interviews emphasised task selection, ordering, and adaptation, a second exploratory audit looked at sequence and branch design. This audit does not judge whether a path is pedagogically excellent. It only checks for visible features that should matter in this task: a shared starting point, movement from concrete to symbolic work, meaningful differentiated branches, targeted support, and avoidance of repeated tasks. Appendix A.4.3 documents how these proxy indicators were derived from the final path structure and task catalogue metadata. Table 5.6 summarises these sequence and branch-design proxies by condition.

The sequence audit does not show an AI advantage in overall ordering. Both conditions showed a similar amount of shared work before the first differentiation point, and the conservative common-start proxy was met by half of the manual paths and three of seven AI-assisted paths. The broad visual-to-symbolic proxy was high in both conditions. It should therefore be read only as evidence that participants often included concrete or visual material before later symbolic work, not as validation of expert-quality sequencing.

The more informative differences appeared inside differentiated branches. Manual paths more often placed high-challenge tasks and application/context tasks within differentiation branches. Manual paths also more often attached support material to a task inside a differentiated branch, whereas none of the AI-assisted paths met this exploratory proxy.

Table 5.6: Exploratory sequence and branch-design proxies by condition.

Proxy indicator	Manual	AI-assisted	Profile
Open-ended mean count summary			
Mean shared top-level task references before first differentiation gate	2.83	2.71	
Paths meeting proxy indicator (%)			
Clear common start (≥ 2 shared top-level task references before first differentiation)	50.0%	42.9%	
Clear visual-to-symbolic ordering proxy	100.0%	85.7%	
Support material attached inside a differentiation branch	33.3%	0.0%	
High-challenge task in a differentiation branch	100.0%	71.4%	
Application/context task in a differentiation branch	66.7%	57.1%	
Long or duplicate-heavy path	16.7%	42.9%	

Note. These exploratory indicators describe observable path features and not expert judgements of pedagogical quality. Visual profiles show manual (gray, top) and AI-assisted (blue, bottom), matching the highlighted column headers. The shared-task row uses an open-ended count axis; the arrow indicates that larger values are possible. Percentage rows use separate 0–100 bars. The long-or-duplicate-heavy proxy is met when a path has more than 18 task references or at least three duplicate references.

This does not mean that support material must always appear inside branches. It only checks whether support, when present, was targeted to a learner-specific route rather than attached elsewhere in the path. The earlier structural results showed that AI-assisted paths contained more support-gate structure overall, but this exploratory audit suggests that the support was not necessarily integrated into differentiated learner routes. AI-assisted paths were also more often long or duplicate-heavy. Together, these results suggest that AI assistance helped generate structure, but did not reliably translate that structure into targeted, branch-level differentiation.

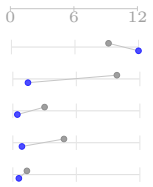
5.5 Process Mechanism: Steering and Repair

The artefact results raise a practical question: if AI participants could ask for task selection, sequencing, and differentiation help, why were their final paths not more clearly differentiated? The interaction and chat evidence point to a mechanism. AI assistance reduced some direct construction actions, but it introduced a suggestion pipeline that participants had to steer, apply, validate, and sometimes repair.

The interaction logs show that AI assistance changed the type of work participants performed. AI participants did not stop searching the catalogue. They logged more catalogue searches per participant than manual participants (12.00 versus 9.17), while

performing far fewer direct construction actions such as adding nodes (1.43 versus 9.83) or creating differentiation gates (0.86 versus 4.83). Table 5.7 reports the comparable shared actions and the AI-specific workflow events that define this work shift.

Table 5.7: Selected logged actions, normalised by condition size.

Action	Manual	AI-assisted	Action profile
Comparable shared actions per participant			
Catalogue search	9.17	12.00	
Add node	9.83	1.43	
Delete node	3.00	0.43	
Create differentiation gate	4.83	0.86	
Add support content	1.33	0.57	
AI-specific workflow events per AI participant			
AI accept-and-apply	–	2.43	*AI-only*
AI suggestion normalisation	–	39.71	*AI-only*

Note. Visual profiles show manual (gray, top) and AI-assisted (blue, bottom) condition means for comparable shared actions on a common 0–12 per-participant axis. The same pattern as the former shared-action figure is visible here: AI participants still searched the catalogue heavily but manually added and created fewer nodes/gates. AI-specific workflow events are not plotted on the shared-action axis because they occur only in the AI-assisted condition and use a different scale.

The AI-specific logs provide one explanation for the poor usability scores. Across the 7 AI participants in the between-subjects cohort, the system recorded 17 accept-and-apply actions and 278 AI-suggestion normalisation actions. Here, normalisation means that the software converted AI suggestions into the graph format expected by the editor. This does not mean participants manually performed 278 separate repairs; it shows that the application repeatedly had to turn generated suggestions into valid path structures. In practical terms, the AI condition made participants work through a suggestion pipeline. Some direct editing burden decreased, but a new burden appeared: interpreting, applying, validating, and sometimes repairing generated structures. This is the process-level version of the artefact result. Participants created fewer structures by hand, but the generated structures still had to be made valid, task-aligned, and usable inside the editor.

5.5.1 Conversational Steering Evidence

The AI condition produced 144 raw chat-message records in the between-subjects cohort. After exact deduplication by participant, role, turn index, and content hash, the chat corpus used for analysis contained 104 messages. The turns-per-path analysis counts deduplicated participant messages as conversational turns because those messages represent

the prompts through which participants steered the assistant. Across the seven AI-assisted final homework learning paths, participants wrote 52 turns in total ($M = 7.43$, median = 7, range = 6–10). This is used as descriptive evidence of steering effort, not as an inferential metric or as evidence that fewer turns would necessarily mean better collaboration.

Several chat excerpts explain why the AI condition could help with task discovery while still lowering perceived quality and usability. Participants asked the AI to repair invalid or misaligned structures, to add the task-specified three-level differentiation, or to resolve editor-state problems. Table 5.8 gives representative examples of these steering and recovery requests.

Table 5.8: Representative AI-condition chat evidence.

Observed issue	**Representative excerpt**
Invalid catalogue references	"der Pfad ist ungueltig wegen: Der generierte Pfad nutzt Aufgaben ausserhalb des erlaubten Katalogs: ALICE/1910"
Missing three-level differentiation	"wo ist jetzt die Differenzierung in drei Schwierigkeitsgrade? bitte ergaenze das"
Unexpected differentiation structure	"warum gibt es jetzt 4 differenzierungen?"
Apply/edit friction	"Loesche alles. Ich kann leider nichts bearbeiten."
Stale path state	"Okay mir wird noch ein alter Pfad angezeigt."

These excerpts do not imply that all AI interactions failed. They show a recurring mechanism behind the quantitative pattern: the AI was useful as a conversational planning and search partner, but participants had to monitor validity, enforce the task-specified differentiation structure, and recover from interaction-state issues. This is a plausible explanation for the combination of slightly higher task-discovery ratings and substantially lower ease-of-use ratings in the AI condition.

5.6 Teacher Voice and Design Needs

The post-condition open responses show that participants experienced the study not only as an AI comparison, but as a planning-workflow problem. This section therefore reports the comments after the artefact and process evidence: the written responses help translate the observed mechanism into design needs. The coding was keyword-assisted and then inspected through representative excerpts. It should therefore be interpreted as exploratory qualitative evidence rather than as a validated thematic analysis. Table 5.9

summarises the coded design themes by condition.

Table 5.9: Open-response design themes by condition.

Design theme	Manual	AI-assisted	Total	Count profile
Participants mentioning theme				
Need for reordering or restructuring	3	1	4	
Need for metadata, filters, or difficulty labels	6	5	11	
German/English language mismatch	2	1	3	
Need for onboarding or practice	4	3	7	
AI steering, repair, or recovery issue	0	6	6	
Need for pedagogical context or rationale	4	3	7	
Time pressure	4	3	7	

Note. Counts indicate participants with at least one matching open-ended response; coding is exploratory and keyword-assisted. The profile uses a common 0–7 participant-count axis.

The strongest cross-condition open-response theme was task discovery. Eleven of 13 between-subjects participants mentioned a need for better metadata, filters, task structure, difficulty labels, or search support. This directly echoes the formative interviews: teachers were not only looking for more material, but for material that could be found, compared, and sequenced quickly for a particular class. The remaining language mismatch also mattered: despite the German study flow and translated interface, three participants explicitly described friction caused by German planning intentions and English catalogue labels or search terms.

Seven participants mentioned onboarding, practice, explanations, or difficulty understanding the platform. This helps explain why time pressure was prominent in both conditions. Participants spent part of the study learning how the editor, catalogue, and differentiation nodes worked. The reordering theme further shows that the base editor constrained pedagogical revision: several participants wanted to rearrange tasks after seeing the emerging path.

The clearest AI-specific open-response theme was steering and recovery. Six of seven AI-assisted participants mentioned the AI assistant, generated suggestions, repair, errors, unsatisfactory adjustments, or difficulty making the AI change the path as intended. This aligns with the chat evidence and the usability battery. The AI was useful for brainstorming and narrowing a large task catalogue, but participants still needed reliable ways to steer, inspect, undo, and repair generated structures.

For a teacher-tool builder, the teacher voice adds one more caution. The AI layer is not the only intervention surface. Even without AI, participants wanted clearer task

metadata, difficulty signals, path reordering, and a faster route from pedagogical intention to a valid path. The next version therefore needs both a better planning editor and a more constrained AI workflow.

5.7 Robustness Checks and Evaluation Boundaries

The results above are best interpreted as formative design evidence. This section reports exploratory robustness checks that scope the main reading. Additional audit-trail details on the protocol-deviation crossover case, data linkage, and automated proxy metrics are preserved in Appendix A.5.2, Appendix A.5.1, and Appendix A.4.4.

5.7.1 Exploratory Robustness Checks

Because the revised between-subjects cohort is small ($n = 6$ manual, $n = 7$ AI-assisted), statistical tests are reported as exploratory checks rather than as the main basis for interpretation. For Likert and continuous outcomes, exact rank-permutation Mann–Whitney tests were computed from the observed ranks. Welch's t-tests were also computed as sensitivity checks, but the Mann–Whitney results are reported as the primary tests. For binary audit and open-response indicators, Fisher's exact test was used. Because no multiple-comparison correction was applied, p-values are interpreted as exploratory signals. They help scope the descriptive and mixed-methods patterns, but they are not treated as confirmatory evidence. Table 5.10 reports these exploratory between-condition checks.

Most condition differences did not reach the conventional $p < .05$ threshold. This includes selected survey and usability outcomes: overall quality ($p = .829$), differentiation quality ($p = .143$), task discovery ($p = .911$), enough time ($p = .168$), ease of use ($p = .081$), and perceived complexity ($p = .081$). The direction of these effects is still substantively important for a formative study, but the small sample does not support strong claims that these differences generalise beyond this cohort.

Three exploratory tests did cross $p < .05$. AI-assisted participants agreed more strongly that they had to learn a lot before getting started ($p = .015$), AI-assisted paths contained fewer differentiation gates ($p = .032$), and AI steering or recovery issues appeared in the open responses much more often in the AI condition ($p = .0047$). The exact-three-option

Table 5.10: Exploratory between-condition checks by metric scale.

Metric	Manual	AI-assisted	Δ	r_{rb}	p	Test
Mean scores (1–5 Likert scale)						
Overall quality	2.50	2.29	-0.21	-0.07	0.8287	MW
Differentiation quality	4.00	3.00	-1.00	-0.50	0.1434	MW
Task discovery	2.67	2.71	0.05	0.05	0.9108	MW
Enough time	2.17	1.14	-1.02	-0.41	0.1678	MW
Easy to use	3.67	2.00	-1.67	-0.62	0.0810	MW
Felt confident	3.00	1.71	-1.29	-0.43	0.2389	MW
Unnecessarily complex	1.50	2.86	1.36	0.62	0.0810	MW
Had to learn a lot	1.83	4.00	2.17	0.81	0.0152	MW
Mean count summaries						
Differentiation gates	3.00	1.57	-1.43	-0.64	0.0321	MW
Differentiation options	8.50	5.00	-3.50	-0.60	0.0728	MW
Interview-derived score	4.67	3.71	-0.95	-0.62	0.0606	MW
Duplicate task references	0.67	3.00	2.33	0.05	0.9534	MW
Percentage indicators						
Exact three-option differentiation gate	100.0	42.9	-57.1	–	0.0699	Fisher
AI steering or recovery issue	0.0	85.7	85.7	–	0.0047	Fisher

Note. Δ is AI-assisted minus manual. MW denotes exact rank-permutation Mann–Whitney tests; Fisher denotes Fisher's exact test for binary indicators. Lens icons mark exploratory rows crossing $p < .05$: learnability/efficiency, differentiation, and AI-control friction. For negatively worded usability rows, higher means more friction.

gate rate was directionally large but did not cross the threshold (manual 100%, AI 42.9%, Fisher $p = .070$). The inferential checks taken together point to the interpretation that the clearest exploratory signals are about learnability, AI-control friction, and explicit differentiation structure.

The protocol-deviation case adds nuance without estimating a treatment effect, the linkage caveats affect individual audit trails rather than the condition-level pattern, and the Backfisch-inspired automated proxies point in the same direction while remaining unvalidated quality indicators.

5.8 Actionable Results Summary

Overall, the Phase B results support a cautious formative conclusion. The AI condition showed some promise for task discovery: AI participants reported task finding as slightly more efficient and used the chat to ask for task selection, sequencing, and differentiation help. However, the study does not show that AI assistance improved final perceived quality or produced stronger differentiated homework-learning-path artefacts. Manual participants rated differentiation quality, curriculum fit, usability, confidence, and willingness to use the final homework path more favourably.

The evidence streams align around one mechanism. The survey results show no

descriptive AI advantage in perceived quality and a clear AI disadvantage in usability. The artefact metrics show that AI-assisted paths had fewer differentiation gates and options but more support structures. The interview-derived path audit shows that manual paths more often made the task-specified three-group differentiation explicit and covered a broad set of skills, while AI paths had more duplicate task references. The exploratory sequence audit suggests that AI-generated support was not necessarily integrated into differentiated branches and that AI paths were more often long or duplicate-heavy. The open responses show that task metadata, onboarding, time pressure, and editability were cross-condition issues, while steering and recovering AI suggestions was specific to the AI condition. The interaction and chat logs then show how this happened: participants in the AI condition spent effort applying, normalising, validating, correcting, and conversationally steering AI-generated structures. Table 5.11 translates these findings into priorities for the next prototype iteration.

Table 5.11: Actionable summary for the next prototype iteration.

Result finding	**Evidence in this chapter**	**Design action**
Task discovery remained the strongest cross-condition need.	11 of 13 participants mentioned metadata, filters, task structure, difficulty labels, or search support; AI task-discovery ratings improved only slightly.	Add difficulty labels, topic/skill filters, language-aligned search, and validated, catalogue-bound AI retrieval.
Three-level differentiation was not reliably preserved by AI assistance.	Manual paths all contained at least one exact three-option gate; AI-assisted paths did so in three of seven cases.	Make low/expected/high branches an explicit schema and warn when a path lacks task-specified differentiation.
AI shifted work from construction to supervision.	AI participants performed fewer direct construction actions but produced 17 accept-and-apply actions, 278 normalisation events, and 52 participant chat turns.	Provide branch-level accept/edit/reject controls, visible undo, state freshness indicators, and validation before applying suggestions.
Support generation did not automatically mean targeted differentiation.	AI-assisted paths had more support structure overall, but none met the exploratory proxy for support attached to differentiated branch tasks.	Attach supports to learner-specific routes and explain which difficulty or misconception each support addresses.
Artefact metrics need expert calibration.	Automated proxies point in the same direction as other evidence, but expert validation is not yet available.	Use teacher review to turn the audit criteria into a path health check for validity, feasibility, and pedagogical clarity.

The strongest defensible claim is therefore not that AI improved differentiated homework planning outright. Rather, AI assistance helped some participants get ideas and navigate an unfamiliar task catalogue, but the current prototype did not reliably transform that help into valid, differentiated paths that teachers could control. For the next design iteration, structurally valid differentiation gates, validated, catalogue-bound task suggestions, branch-level editing, and stronger path-state handling are prerequisites for a fairer test of the AI concept.

Chapter 6

Discussion

6.1 Interpretation of Core Findings

This study provides a formative, design-oriented answer to the research question. The prototype showed that AI assistance can support parts of teacher-facing homework-learning-path planning, especially ideation, sequencing, and catalogue navigation. At the same time, the evaluation shows that these benefits do not automatically translate into stronger final artefacts unless the system is structurally reliable, catalogue-grounded, and easy for teachers to control. The main contribution of this iteration is therefore not a finished product claim, but a clearer account of the conditions under which AI-assisted planning can become useful for differentiated instruction.

The core mechanism is visible across the evidence streams. Manual participants created more task-aligned differentiation structures: every manual path contained at least one exact three-option differentiation gate for the three performance groups, while only three of seven AI-assisted paths did. This matters because the study task asked for explicit alternatives for different ability groups, and Phase A showed that teachers often reason about differentiation through practical levels such as basic, intermediate, and advanced. The AI-assisted paths were not empty or irrelevant; several covered useful fraction skills and contained support structures. However, they were more likely to repeat tasks, omit the task-specified exact three-option structure, or shift visible structure toward support material rather than parallel ability-level branches.

The efficiency result points in the same direction. AI assistance changed, but did not solve, the catalogue problem. Participants still needed to inspect task fit, check whether

references belonged to the allowed catalogue, and verify whether generated structures had been applied as intended. The AI condition therefore changed the shape of the work: it helped with the blank-page and search problem for some participants, but it also created validation and repair work. Any benefit in ideation could be reduced if participants had to spend that time checking invalid references, unexpected branch counts, repeated tasks, stale path state, or difficult apply/edit behaviour.

Teacher agency is the design issue that ties these findings together. Phase A repeatedly pointed toward malleable, teacher-controlled support. Teachers wanted help, but not an opaque system that decides what their class should do. The AI condition made this balance fragile. When participants had to debug why a generated path was invalid or why a proposed structure did not match the required three groups, supervision became technical rather than pedagogical. Good teacher–AI collaboration should still involve teacher judgement, but that judgement should be about task appropriateness, difficulty progression, and support quality, not about recovering from system state or validating whether a generated suggestion contained unavailable tasks.

6.2 Coherence from Phase A to Phase B

The Phase B results are coherent with the Phase A requirement logic. Phase A identified practical constraints around task discovery, visible three-level differentiation, teacher control, student-device logistics, and the need to connect planning with evidence about student understanding. Phase B directly tested the constraints around task discovery, visible differentiation, and teacher control in a teacher-facing planning workflow. It also deliberately avoided the student-device problem by evaluating planning rather than a student-facing adaptive tutor. The diagnostic feedback loop remains outside the implemented prototype and should be treated as a future design direction.

This framing explains why the prototype focused on differentiated homework planning. Teachers described device access, batteries, internet availability, input difficulty, and classroom logistics as barriers to student-side deployment. A teacher-side planning tool is therefore a coherent first step. At the same time, Phase A sources I2, I4, I5, and I7 showed that teachers often want planning to connect back to evidence about student understanding. The current system supports the planning side of this loop, but not yet

the diagnostic side.

The design theory refined by this study is:

> AI assistance is most promising when it helps teachers search, organise, and adapt available tasks inside a structured planning workflow; quality gains depend on enforced catalogue grounding, visible three-level differentiation, and transparent teacher control.

This is more precise than a broad claim that AI improves planning. It states the conditions under which AI support may become useful for differentiated instruction and why this prototype did not yet meet those conditions reliably.

6.3 Relation to Prior Work

The findings align with prior work on AI lesson-planning copilots in an important respect: conversational AI can reduce the blank-page problem and support early-stage ideation. Participants in the AI-assisted condition used the chat to ask for task selection, sequencing, and differentiation support. This resembles the role of AI copilots in lesson-planning systems where the model acts as a drafting and brainstorming partner.

The study also adds a constraint that is especially important for structured educational artefacts. A homework learning path is not only a text document. It is a graph with validity constraints, task references, support gates, and differentiation gates. For such artefacts, fluent suggestions are insufficient. The system must preserve catalogue grounding and graph validity while still aligning with pedagogical intent. Invalid references, duplicated tasks, and missing three-option gates are therefore not minor interface issues; they are quality risks for AI-supported learning-path construction.

The results also fit a Design-Based Research interpretation. The study set out to test whether AI assistance could improve homework-learning-path planning. Because the prototype itself caused friction, the results cannot show the general effect of AI assistance in a smoother or more mature tool. This does not mean that the iteration was incomplete. Instead, the evaluation completed one design loop by showing that possible benefits depend not only on the language model, but also on how well the surrounding platform supports the work. Teachers need to be able to communicate pedagogical intent clearly, the AI needs access to a structured and current representation of the path and catalogue

constraints, and suggested changes need to be applied transparently and reliably. In that formative sense, the study was productive: it showed promise for ideation and task discovery, while showing that the next prototype must make the handoff between teacher, platform, and AI much easier.

6.4 Boundary Conditions and Validity

The evaluation is strongest as triangulated formative evidence. The conclusions do not rest on a single survey item. Survey responses, interaction logs, final path structures, path-audit criteria, chat evidence, and open-ended responses point toward the same interpretation: AI assistance created useful planning affordances, but also introduced reliability and control costs. The end-to-end linkage from Phase A requirements to Phase B evidence keeps the interpretation grounded in teacher-articulated needs.

The main boundary is sample size and participant profile. Fourteen participants completed the study, participant20 was excluded from the between-subjects comparison after seeing both conditions, and the retained cohort contained 13 participants: 6 in the manual condition and 7 in the AI-assisted condition. These numbers support design interpretation, not population-level claims. The Phase B participants were novice or pre-service teachers, while the formative interviews included experienced teachers. That is defensible because the tool targets planning support for novice teachers, but experienced-teacher judgement is still needed before making stronger claims about final artefact quality.

The prototype itself is also part of the evidence. Some AI-assisted participants encountered invalid task references, unexpected branch structures, stale path state, or difficult apply/edit behaviour. This means the study cannot fully separate the broader design idea from the implemented version participants actually used. The safest reading is that the current workflow limited how clearly possible AI benefits could appear. The analysis therefore treats rough interactions as findings about design readiness [6, 24].

There are also limits in topic, measurement, and artefact validation. The study used a bounded planning task on Grade 7 fractions and a controlled catalogue, which improved comparability but limits generalisation to other topics and school contexts. The usability items were SUS-style rather than a full validated SUS instrument. The path and sequence

audits used observable proxies derived from Phase A and the study task; they are useful formative indicators, but they do not replace independent expert ratings. The study also evaluated the deployed prompt-based prototype with the configured model, not the upper bound of AI-assisted planning with stronger retrieval, validation, or model capability.

6.5 Design Implications and Next Iterations

6.5.1 Implications for Teacher Education

For teacher education, the findings suggest that AI planning tools should be introduced as critique-and-refinement partners, not as automatic homework-learning-path generators. Novice teachers may benefit from AI-generated starting points, but they also need to learn how to inspect task fit, difficulty progression, differentiation structure, and support placement. The tool could therefore become useful in teacher education if it prompts reflection on why a task fits a group, what skill a node addresses, and whether support material scaffolds learning rather than replacing differentiation.

6.5.2 Implications for Product and System Design

The next prototype should treat structure as the main product problem. The evaluation shows that fluent chat is not enough when the output is a constrained graph rather than a text draft. AI support must therefore operate inside the editor's rules. Task references need stronger enforcement than prompt context alone; differentiation gates should default to exactly three levelled branches; and invalid references, duplicates, missing differentiation, weak skill coverage, and stale state should be surfaced before teachers have to find them manually.

The central interface requirement is branch-level control. Teachers should be able to accept, reject, revise, and undo parts of an AI suggestion without losing their own changes. This keeps supervision pedagogical rather than technical: teachers judge task fit, difficulty progression, and support quality instead of repairing graph state.

The next evaluation step should involve experienced teachers. Their review of the final paths can turn the current audit criteria into a compact rubric for curriculum alignment, difficulty fit, differentiation clarity, feasibility, support quality, and catalogue validity. The

same rubric can later become an in-product path-health check.

The longer-term direction is to connect planning to diagnosis. Phase A showed that teachers see planning, feedback, and progress visibility as connected. A future prototype should therefore help teachers move from evidence of student difficulty to differentiated homework and back to follow-up feedback, but only after the basic planning workflow is reliable, inspectable, and easy to revise.

Chapter 7

Conclusion

7.1 Summary of the Thesis

This thesis asked whether AI assistance can help novice teachers plan differentiated mathematics homework learning paths. The work was deliberately grounded in teacher practice before the prototype was evaluated. Phase A used interviews and material review to identify practical requirements: teachers need help finding suitable tasks, representing clear levels of differentiation, keeping control over suggested material, and working within realistic classroom technology constraints. Phase B then tested a runnable learning-path editor in a controlled comparison between manual planning and AI-assisted planning.

7.2 Answer to the Research Question

The research question asked whether AI-assisted planning helps novice teachers create differentiated homework learning paths more efficiently, while maintaining or improving perceived homework quality, compared with manual planning. The answer from this study is cautious. In this prototype, AI assistance helped some participants with brainstorming and task discovery, but it did not clearly improve final perceived homework quality or reduce experienced planning burden. The clearest visible difference was structural: every manual path contained at least one exact three-option differentiation gate, while fewer than half of the AI-assisted paths did.

This should be read as evidence about the implemented prototype, not as a rejection of AI-assisted planning in general. The study shows that AI support needs stronger

structural reliability and teacher control before its benefits can be expected to appear in final path quality.

7.3 Main Contributions

The first contribution is a teacher-grounded requirement set for differentiated homework planning. The interviews showed that the problem is not simply a lack of educational content. Teachers already use textbooks, task collections, and digital resources. The harder problem is selecting, sequencing, adapting, and checking material for a specific class.

The second contribution is a working prototype that turns these requirements into a teacher-facing planning workflow. The prototype combines a task catalogue, a graph-based learning-path editor, differentiation gates, support gates, and an optional AI planning panel.

The third contribution is a mixed-methods evaluation of the prototype. The study links survey responses, final learning paths, interaction logs, chat evidence, and open-ended feedback. This makes it possible to see not only whether participants liked the tool, but also what they built and where the workflow created friction.

7.3.1 Educational Technology Contribution

For educational technology, the thesis shows that teacher-facing AI tools need more than fluent suggestions. Differentiated homework planning is a structured design task. The tool must preserve valid task references, make differentiation visible, support meaningful difficulty levels, and allow teachers to edit the result without losing control. AI can be useful as a planning partner, but it should support teacher judgement rather than move teachers into technical repair work.

7.3.2 Methodological and Machine Learning Contribution

This thesis also contributes to methodology and machine learning system design. The methodological contribution is that it shows how an early-stage educational AI prototype can be evaluated responsibly before classroom deployment. Because the tool was tested as a teacher-side planning aid, the study does not claim effects on student learning.

Instead, it evaluates the planning process through final learning paths, usability responses, interaction logs, chat evidence, and participant reflections. The claims are therefore limited to planning quality and workflow experience in this prototype.

The machine learning contribution is a clearer design target for ML-based planning systems. The study shows that AI support for differentiated homework should be treated as planning support, not automatic homework generation. In this setting, the model should help teachers search, organise, and adapt existing materials while keeping the teacher's planning decisions visible and editable. This means that model output should be handled as structured suggestions over a task catalogue and learning-path graph, not as free text that teachers must repair afterwards. Useful AI assistance therefore needs enforced catalogue grounding and should preserve the differentiation structure required by the teacher's planning context. The system should also check whether AI suggestions are valid, explain why tasks fit particular difficulty levels, and let teachers easily accept, change, or reject the suggestions or parts of them.

7.4 Practical Takeaways

The practical takeaway is that the next version should make valid structure hard to lose and easy to inspect. This includes enforced catalogue boundaries, visible three-level differentiation, branch-level editing, and clearer path-health feedback.

The study also shows that better search and metadata matter in both conditions. Participants repeatedly asked for clearer difficulty labels, filters, task descriptions, and ways to reorder the path. These are not secondary interface details. They are part of the planning problem teachers actually face.

7.5 Final Outlook

The broader promise of AI in this thesis is not automatic lesson planning. The more realistic promise is a tool that helps teachers search, organise, adapt, and reflect while keeping professional judgement in the teacher's hands. This study shows that such a tool is plausible, but also that the current prototype is not yet enough. The next step is to strengthen the planning workflow first, then evaluate whether its benefits appear more

clearly.

In that sense, the main result is productive rather than purely negative. The study identifies where AI support already helps and, more importantly, what must be fixed before it can support differentiated mathematics planning in a dependable way: valid catalogue references, visible three-level differentiation, branch-level editability, and teacher control over the final path.

Appendix A

Study Materials and Evaluation Details

This appendix preserves the detailed study materials and evaluation documentation that support Chapters 4 and 5. The main chapters keep the methodological and results arguments compact; the appendix retains the task wording, instrument details, AI prompt structure, logging description, metric caveats, and audit-trail details needed for auditability.

A.1 Formative Corpus and Traceability Details

Table A.1 summarises the Phase A sources used for requirement elicitation. The table reports role and data type rather than treating the interviews as a representative sample. Some conversations were recorded and transcribed or summarised in detail, while others were documented as field notes or synthesis memos.

Table A.2 records how the Phase A themes were translated into Phase B evidence.

Table A.1: Phase A formative corpus.

Source	Role/context	Data type	Main contribution to requirements
I1	Former teacher / education researcher	Field notes	Curriculum context, school types, and access to Baden-Wuerttemberg curriculum material.
I2	Experienced mathematics teacher, Realschule, grades 5–10	Recorded interview, condensed notes, transcript excerpt	Three-level textbook differentiation, Klett material use, QR-linked resources, short-test workflow, and need for diagnostic visibility.
I3	Mathematics teacher / student, Realschule experience	Interview notes	Three-level grouping, teacher-created worksheets, technology logistics, and teacher-side AI scope.
I4	Mathematics teacher, Realschule	Interview notes	Homework planning practices, enough existing material, exam-oriented task selection, and interest in progress dashboards.
I5	Mathematics teacher, Gemeinschaftsschule/Gymnasium context	Interview notes	Adaptive teaching routines, diagnostic tests, device limitations, and teacher control over correction/feedback.
I6	Mathematics teacher, Realschule	Interview notes	Performance-level structure, revision needs, homework checking practices, and student motivation constraints.
I7	Former teacher / educator outside German mathematics context	Interview notes	General teaching workload, grading evidence, motivation, curriculum rigidity, and learner heterogeneity.
I8	Design/education stakeholder	Synthesis memo	Need for lightweight, malleable prototypes and iterative teacher-facing design.

Table A.2: Traceability from formative interviews to Phase B analyses.

Phase A theme	Design requirement	Phase B evidence	Result interpretation
Task discovery and material selection are costly	Help teachers find suitable fraction tasks inside the planning workflow	Task-discovery item, catalogue-search logs, final task references	AI helped search only modestly; AI participants still searched the catalogue heavily.
Three-level differentiation must be practical	Represent the task-specified weak/expected/strong groups explicitly	Differentiation gates, exact-three gates, task difficulty spread	Manual paths contained more explicit differentiation; AI paths made this structure explicit less often.
Teachers need malleable output and control	Keep suggestions editable and subordinate to teacher judgement	Usability/confidence items, chat repair evidence, suggestion-normalisation logs	The AI layer introduced control and validation costs.
Student-side technology is logistically fragile	Keep the prototype teacher-facing	Prototype scope and study task	The evaluation scope is coherent with teacher-side deployment constraints.
Teachers want visibility into student needs	Connect planning to diagnosis and progress evidence in future work	Feature coverage and unmet-needs analysis	The current prototype supports planning, not the diagnostic feedback loop teachers also wanted.

A.2 Participant Task Text

The following text documents the substantive assignment. The German version was the participant-facing wording; the English version is a reporting translation.

English translation.

> Create a differentiated homework assignment for a Grade 7 Gymnasium class in Baden-Wuerttemberg based on the curriculum, focusing on the review, consolidation, and application of fraction-calculation skills (equivalence, expanding, simplifying) as preparation for working with rational numbers.
>
> A differentiated homework assignment assigns tasks to students according to their performance level. The homework should be designed so that it can be completed in about 45 minutes.
>
> Within the platform, a predefined task pool is available, including tasks from ALICE (TUM) and Mathebattle. Please create the homework assignment only on the basis of these tasks. In addition, you can add support materials to individual tasks, for example short explanations, links, or videos, to support students when needed.
>
> The class consists of:
>
> - 6 high-performing students,
> - 14 students at the expected level,
> - 6 lower-performing students.
>
> Your homework assignment should:
>
> - contain differentiation nodes at which students are assigned to different performance groups,
> - assign exactly one task per group at each differentiation node,
> - ensure that each group receives tasks appropriate to its performance level.
>
> The tasks should support students in:
>
> - recognising and forming equivalent fractions,
> - simplifying and expanding fractions,
> - applying these skills in typical application situations, for example comparing fractions, word problems, and preparation for algebraic terms.

Support materials are shown to students when they cannot complete a task independently.

German original.

Erstellen Sie eine differenzierte Hausaufgabe für eine 7. Klasse am Gymnasium in Baden-Württemberg auf Grundlage des Bildungsplans mit dem Fokus auf der Wiederholung, Festigung und Anwendung von Bruchrechenfertigkeiten (Äquivalenz, Erweitern, Kürzen) als Vorbereitung auf den Umgang mit rationalen Zahlen.

Eine differenzierte Hausaufgabe weist den Schülerinnen und Schülern Aufgaben entsprechend ihrem Leistungsniveau zu. Die Hausaufgabe sollte so gestaltet sein, dass sie in etwa 45 Minuten bearbeitet werden kann.

Innerhalb der Plattform steht Ihnen ein vordefinierter Aufgabenpool (u. a. Aufgaben aus ALICE (TUM) und Mathebattle) zur Verfügung. Bitte erstellen Sie die Hausaufgabe ausschließlich auf Basis dieser Aufgaben. Zusätzlich können Sie zu einzelnen Aufgaben Unterstützungsmaterialien (z. B. kurze Erklärungen, Links oder Videos) hinzufügen, um Schülerinnen und Schüler bei Bedarf zu unterstützen.

Die Klasse besteht aus:

- 6 leistungsstarken Schülerinnen und Schülern,
- 14 Schülerinnen und Schülern auf dem erwarteten Niveau,
- 6 leistungsschwächeren Schülerinnen und Schülern.

Ihre Hausaufgabe soll:

- Differenzierungsknoten enthalten, an denen Schülerinnen und Schüler verschiedenen Leistungsgruppen zugewiesen werden,
- an jedem Differenzierungsknoten genau eine Aufgabe pro Gruppe zuweisen,
- sicherstellen, dass jede Gruppe Aufgaben erhält, die ihrem Leistungsniveau entsprechen.

Die Aufgaben sollen die Schülerinnen und Schüler dabei unterstützen:

- gleichwertige Brüche zu erkennen und zu bilden,
- Brüche zu kürzen und zu erweitern,

- diese Fertigkeiten in typischen Anwendungssituationen anzuwenden, zum Beispiel Brüche vergleichen, Sachaufgaben und Vorbereitung auf algebraische Terme.

Unterstützungsmaterialien werden den Schülerinnen und Schülern angezeigt, wenn sie eine Aufgabe nicht selbstständig bearbeiten können.

A.3 Questionnaire and Open-Ended Prompts

Immediately after the assigned condition, participants completed a short questionnaire capturing planning efficiency, perceived quality, usability, and control. Each item used a 5-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree). The instrument was documented with matched English and German wording, but the main Phase B study used the German version because the interest form showed German as the shared comfortable language.

Analysed Likert items shown in both conditions:

1. "How familiar are you with task collections like Alice or Mathebattle?" (*Prior Catalog Familiarity*)
2. "The tasks I selected fit the learning goal well." (*Task-Goal Fit*)
3. "I am satisfied with the overall quality of the homework assignment I created." (*Perceived Homework Quality*)
4. "The homework assignment adequately addresses different student ability levels." (*Differentiation Quality*)
5. "I would personally use this homework assignment." (*Use Intention*)
6. "The created homework assignment aligns well with the Baden-Wuerttemberg Bildungsplan." (*Curriculum Fit*)
7. "I was able to find appropriate tasks for the homework assignment efficiently." (*Task Discovery Efficiency*)

8. "I was able to create parallel task groups without much effort." (*Parallel Group Construction*)
9. "I had enough time to complete the planning task to my satisfaction." (*Perceived Time Sufficiency*)
10. "I felt in control of the lesson-planning process." (*Agency*)
11. "The system was easy to use." (*Ease of Use*)
12. "I felt confident using the system." (*Confidence*)
13. "The system was unnecessarily complex." (*Perceived Complexity*)
14. "The functions were well integrated." (*Functional Integration*)
15. "I had to learn a lot before I could get started." (*Learnability Burden*)

Some additional survey fields, such as study programme, semester, prior planning experience, and perceived task realism, were used to describe the sample and context rather than as primary between-condition outcomes. Core outcome items are compared across conditions using Mann–Whitney U tests where exploratory tests are reported. Prior catalogue familiarity is reported descriptively to contextualise whether participants were already comfortable with the task sources; it is not used for subgroup testing in the realised analysis.

At the end of each session, participants provided open-ended written responses online. Each participant answered five prompts about their experience in the assigned condition. Three prompts were shared across both conditions, and two prompts branched by condition:

1. **Shared prompt (planning and discovery):** How did the participant find and select tasks, and what was most helpful versus most difficult?
2. **Shared prompt (quality and differentiation):** How well did the final path address the student profiles, and what should be changed?
3. **Shared prompt (agency):** How did the participant experience the balance between own pedagogical decisions and tool support?

4. **Condition-specific prompt 1:** Condition AI: Did the AI surface tasks the participant would likely not have found manually? Condition A: What challenges arose when searching for suitable tasks without AI support?

5. **Condition-specific prompt 2:** Condition AI: What role did the AI play in planning (e.g., brainstorming, task selection, sequencing, differentiation)? Condition A: What additional support would have made manual task discovery easier?

The System Usability Scale (SUS) [5] was used as a design reference for short usability items covering ease of use, confidence, complexity, integration, and learnability. The implemented post-condition form did not contain the complete 10-item SUS instrument. Therefore, the study reports these items as a short, non-validated SUS-style usability battery rather than calculating a 0–100 SUS score or comparing against the standard SUS benchmark. Because both conditions used the same base platform, these usability items captured both shared editor friction and condition-specific friction introduced by the AI layer. The original evaluation design also considered NASA-TLX [18], a full TAM battery [11], and the Nazaretsky et al. Teacher Trust in AI-EdTech scale [22], but these were dropped to reduce response burden and avoid ambiguous interpretation at the current sample size. In particular, workload self-reports can blur productive pedagogical effort and interface friction, while longer multi-item scales would add statistical overhead without directly strengthening the core condition comparison.

A.4 Logging, Metrics, and Realised Evaluation Details

Interaction metrics were derived from structured event logs captured by the system backend. The schema supported several event types:

`SUGGESTION_GENERATED` Planned support for generated suggestion records.

`USER_ACTION` Action type, target identifiers, source, and AI-specific actions such as `accept-and-apply` and `normalize-ai-suggestion`.

`EDIT_SNAPSHOT` Export-level snapshots such as node count and exported path metadata.

`CHAT_TURN` Speaker (user/AI), message content, turn index, timestamp.

`SESSION_EVENT` Session start/end, participant pseudonym, and assigned condition.

Logs were written to the study backend database during the session and batch-processed at analysis time. No raw participant identifiers appeared in the log.

A.4.1 AI Prompt Structure

The AI chat endpoint used one server-built system prompt that was rebuilt for each turn. Teacher messages were sent as chat messages; the prompt added stable instructions plus dynamic context from the current catalogue and path. Table A.3 summarises the stable prompt structure rather than reproducing the full prompt text, because the available task list and path summary changed with each request.

Table A.3: Prompt structure used by the evaluated AI planning workflow.

Prompt part	**Evaluated implementation**	**Possible improvement point**
Role and style	Frames the assistant as an educational content designer for German mathematics teachers and asks for concise answers.	Add a stronger task-specific planning rubric instead of relying mainly on the general role description.
Catalogue grounding	Adds available catalogue tasks with `contentRef`, display name, optional description, and knowledge-component IDs. The model is told not to invent task references.	Retrieve richer task text, difficulty metadata, and curriculum tags when selecting tasks.
Path schema and invariants	Describes the schema used by the evaluated prototype, including support gates, differentiation gates, and the gate-only invariant for differentiation points.	Use schema-constrained output or typed tool calls to reduce repair and normalisation work.
Current path context	Adds path ID, title, topic, node IDs, task names, support-gate titles, support-node task names, and differentiation-option IDs and names.	Include the full participant task brief and a branch-quality checklist in every turn.
Structured output	Requests `[SUGGESTION]` blocks with diff operations for edits and reserves `[PATH]` blocks for complete new paths.	Require clearer per-operation rationale and validation checks before suggestions are shown.
Rationale	Requires a short "Why these tasks" section before complete `[PATH]` blocks.	Require rationale for each suggested branch, especially for weak, expected, and strong performance groups.

The server-built prompt was therefore a pragmatic grounding prompt. It constrained which task references the model could use and how structured outputs should be shaped. However, it did not inject the full participant task text, did not provide a detailed pedagogical scoring rubric, and did not force a self-check for the three performance groups in every turn. These omissions are plausible improvement points for later versions of the AI workflow.

The original evaluation plan included a full suggestion-acceptance metric, in the same spirit as log-based process measures used in other copilot evaluations [12]. In the realised

study, the logged evidence was narrower. The analysis therefore does not report a survival-based Suggestion Acceptance Rate in which accepted AI units are traced into the final exported artefact. Instead, it uses the AI-condition `accept-and-apply` action count as evidence that participants chose to apply AI-generated suggestions during the session.

The realised logs also contain `normalize-ai-suggestion` actions. These events were emitted when the application had to translate AI-generated structures into the graph format expected by the editor, for example by enforcing differentiation-structure constraints. They are interpreted as implementation-level evidence that the AI workflow required structural mediation, not as actions manually performed one-by-one by participants. Together with chat evidence and open responses, these logs help explain whether AI assistance reduced planning work or shifted work into validation, steering, and repair.

The original plan also considered graph-aware edit effort and text-level edit distance as a way to quantify how much participants changed AI drafts. That complete edit-distance analysis was not realised. Instead, repair and refinement are interpreted through logged construction actions, AI suggestion normalisation events, accept-and-apply events, chat turns, and open-ended comments about applying or changing AI suggestions. This realised evidence is descriptive and qualitative. It is useful for identifying where the AI workflow created friction, but it should not be read as a complete edit-effort metric.

The study task used a fixed planning window. For that reason, elapsed session duration is not treated as a primary between-condition efficiency outcome: most participants were constrained by the same timebox, and session timestamps may also include repeated visits or stale path states. Instead, efficiency is interpreted through perceived time sufficiency, time-pressure comments, and interaction burden. An efficiency advantage in Condition AI would therefore not mean that participants finished earlier. It would mean that they felt less time pressure or needed fewer effortful interaction steps while producing artefacts of comparable or better quality. Table A.4 summarises how the original evaluation plan was realised in the final study.

Table A.4: Planned teacher-side evaluation measures and realised implementation.

Planned component	**Realised in this thesis**	**Interpretive consequence**
Full usability/workload batteries	Reduced to short SUS-style usability items and open responses.	No valid SUS or NASA-TLX score is reported.
Planning time as efficiency outcome	Short fixed planning window and noisy timestamps made elapsed time unsuitable as a primary outcome.	Efficiency is interpreted through perceived time sufficiency, action logs, and comments.
Independent expert quality ratings	Not completed; structural and interview-derived proxy audits were used instead.	Artefact results are formative indicators, not definitive expert ratings.
AI process metrics such as acceptance and edit effort	Partly realised through accept/apply logs, suggestion-normalisation logs, and chat evidence.	AI mechanisms are explained qualitatively and descriptively rather than as a complete edit-distance analysis.

A.4.2 Path Audit Calculation Notes

Table 5.5 was calculated from one final homework learning path per retained participant. The primary artefact was the session-linked selected path; if that record was empty, the latest non-empty path for the same study token was used and flagged in the linkage notes. The audit traversed top-level path nodes, nested differentiation-option nodes, and support nodes, then matched each task reference against the exported task catalogue using its normalised `ALICE/...` or `MATHEBATTLE/...` identifier. References in the explicit `CUSTOM/...` namespace were retained as custom material, while catalogue-like references that did not resolve to the allowed task pool were counted as invalid references.

The five-point interview-derived score is a count of five binary criteria:

1. at least one differentiation gate with exactly three options, corresponding to the three performance groups in the task;
2. at least one low-, one medium-, and one high-difficulty catalogue task, using task difficulty means bucketed as low (≤ 0.40), medium (> 0.40 and ≤ 0.60), and high (> 0.60);
3. coverage of at least three of the five target fraction-skill buckets;
4. no invalid task-catalogue references;
5. a bounded-length proxy for a 45-minute homework path, operationalised as no more than 18 recursive normal path nodes.

The target fraction-skill buckets were equivalence, expanding, simplifying, comparison, and application. They were inferred from task names, task descriptions, knowledge-

component identifiers, and custom-task display names. Additional detected buckets such as representation and operations were preserved in the audit data, but they did not contribute to the target-bucket count because Table 5.5 reports coverage of the skills named in the study task. The row "Mean number of target fraction-skill buckets covered" is therefore the condition mean of a 0–5 distinct-bucket count, while "Paths covering at least three of five target fraction-skill buckets" is the percentage of paths whose count was at least three. Duplicate task references were calculated after normalising task identifiers; repeated uses beyond the first contributed one duplicate each.

A.4.3 Sequence and Branch-Design Audit Calculation Notes

Table 5.6 used the same selected final paths as the path audit, but asked a narrower ordering question: did the visible path structure show a plausible common start, progression, and differentiated branch design? The audit flattened each path into three location groups: top-level tasks, tasks inside differentiation-option branches, and support tasks attached to those nodes. Task references were matched to the catalogue where possible so that difficulty and broad task-type proxies could be inferred from task names, descriptions, and knowledge-component identifiers.

The sequence and branch-design indicators were calculated as follows:

1. **Shared top-level tasks before first differentiation:** the number of top-level task references before the first top-level node that contains a differentiation gate. The clear-common-start proxy was met when this count was at least two.

2. **Visual-to-symbolic ordering:** catalogue tasks were classified as visual/concrete, symbolic, and/or application/context tasks using task text and knowledge-component identifiers. The clear proxy was met when a visual/concrete task appeared before a later symbolic task in the flattened non-custom, non-invalid sequence.

3. **Support inside differentiated branches:** met when a differentiation-option target node had support material attached to it.

4. **High-challenge branch task:** met when at least one task inside a differentiation branch had catalogue difficulty at or above 0.60.

5. **Application/context branch task:** met when at least one task inside a differentiation branch was classified as application/context based on task text or knowledge-component identifiers.

6. **Long or duplicate-heavy path:** met when the flattened path contained more than 18 task references or at least three duplicate task references after normalising task identifiers.

These indicators are intentionally conservative proxies. They describe visible structure in the exported paths, not expert judgements about whether the exact sequence, task choices, or support placement would be optimal in classroom use.

A.4.4 Backfisch-Inspired Automated Proxy Calculation Notes

Table A.5 combines the path audit and sequence-design audit into a small set of observable artefact proxies inspired by Backfisch et al.'s distinction between instructional quality and technology exploitation. These scores are not expert ratings and should not be read as the original Backfisch et al. rubric. They are automated point counts over visible final-path features.

The cognitive activation proxy has a maximum of five points. One point is awarded for each of the following: a clear common start, visual-to-symbolic ordering, broad target fraction-skill coverage, at least one high-challenge task inside a differentiation branch, and at least one application/context task inside a differentiation branch. The instructional support proxy has a maximum of four points: exact three-way differentiation, at least one support gate, support attached inside a differentiated branch, and support on the lowest-difficulty branch. The instructional-quality proxy is the sum of those two sub-scores, with a maximum of nine points.

The platform-affordance use proxy is a breadth score with a maximum of five points. It awards one point each for a non-empty task path, use of at least one differentiation gate, use of at least one support gate, use of at least one custom task reference, and use of tasks from more than one source bucket. This is the quantity/use indicator: it asks which editor affordances appear in the final artefact, not whether they are pedagogically well integrated.

The platform-affordance integration proxy is the stricter structured-use score with a

maximum of four points. It is ordinal rather than additive: a non-empty path receives one point; a path using at least one non-linear affordance such as differentiation, support, or custom material receives two; a path with exact three-way differentiation, catalogue grounding, and either support gates or broad skill coverage receives three; and a path additionally containing support inside differentiated branches, broad skill coverage, catalogue grounding, and no long-or-duplicate-heavy structure receives four. For this reason, the word *proxy* is essential: the score approximates integrated use of platform affordances from observable structure, but it does not directly measure platform-affordance quality by expert judgement.

The evaluation plan included possible validation of automated artefact metrics against human or expert judgement. Those validation results are not yet available. Consequently, the thesis does not report human–human agreement, LLM–human agreement, or an expert-validated Structural Alignment Score. This is an evaluation boundary, not a positive result. However, it is still useful as an appendix check to inspect whether automated artefact proxies point in the same direction as the survey and structural results.

Table A.5: Backfisch-inspired automated artefact proxies by condition. These are observable path proxies, not validated expert ratings.

Proxy metric	**Manual**	**AI-assisted**	**AI − manual**	**Normalised profile**
Condition means in raw points; profile normalised to each row maximum				0% 50% 100%
Cognitive activation proxy (0–5)	4.17	3.29	-0.88	
Instructional support proxy (0–4)	2.00	1.00	-1.00	
Instructional quality proxy (0–9)	6.17	4.29	-1.88	
Platform affordance use proxy (0–5)	3.00	2.71	-0.29	
Platform affordance integration proxy (0–4)	3.33	2.29	-1.05	

Note. Profiles divide each condition mean by the row's maximum possible score. Raw table values and deltas remain in points. The platform-affordance use proxy counts breadth of editor features represented in the final path; the platform-affordance integration proxy is the stricter structured-use proxy.

These automated proxies are not a substitute for expert judgement. They do not assess whether a task is mathematically well chosen, whether the sequencing is pedagogically optimal, or whether a support node would actually scaffold a struggling student. Nevertheless, the pattern is informative: the manual condition scores higher on the observable instructional-quality proxy ($M = 6.17$ versus $M = 4.29$) and on the stricter platform-affordance integration proxy ($M = 3.33$ versus $M = 2.29$). Thus, the automated proxy layer does not reveal a hidden AI advantage in the final artefacts. It reinforces the more

cautious interpretation that AI assistance helped some participants with ideation and construction, but did not reliably translate into stronger observable homework-learning-path structure in this prototype.

Table A.6 summarises the realised evaluation metrics and their sources.

Table A.7 summarises the homework-learning-path artefact metrics used in the analysis.

Direct student learning outcomes, classroom implementation, validated automated homework-learning-path quality scoring, long-term teacher adoption, and comparisons between different internal AI implementations are explicitly out of scope. These are legitimate future research directions, but they answer different questions from the core comparison in this thesis: does AI assistance improve teacher-side planning outcomes over a manual workflow?

A.5 Recruitment and Linkage Audit Notes

Recruitment proceeded in two steps. First, an interest form was sent to mathematics teacher-education students. The exported interest-form data contained 26 submissions; after duplicate submissions were removed, this corresponded to 24 unique registrations. Second, after roughly three weeks, 21 registered people were invited to try the demo. The invitation contained two links: a Google Doc with screenshots and explanatory text for platform familiarisation, and a tokenised access link to the assigned study condition. Participants were asked to spend about 15 minutes familiarising themselves with the platform through the linked Google Doc, about 15 minutes using the platform to build a differentiated homework learning path, and about 15 minutes completing the post-condition survey. After the short platform task, the demo timer prompted participants to submit their path and continue to the survey. Participants were compensated with grocery vouchers, and the study was supported by ELLIS. Table A.8 reports the anonymised characteristics available from the duplicate-cleaned interest-form pool.

A.5.1 Data Linkage Caveats

Before analysis, the survey, path, interaction, and chat data were linked through participant pseudonyms, token identifiers, session identifiers, and selected homework learning-path identifiers. The latest completed Tally submission snapshot contained 14 completed

Table A.6: Summary of evaluation metrics.

Metric	Category	Data Source	Measures	Precedent
Human–AI Interaction				
AI accept-and-apply count (Condition AI only)	Interaction	System log	Applied AI suggestions during the session	Log-based copilot evaluation [12]
AI suggestion normalisation count (Condition AI only)	Interaction	System log	Structural mediation required by generated suggestions	Custom
Perceived Time Sufficiency	Survey	Post-condition Likert item	Time adequacy under the short fixed planning window	Custom
Interaction Burden	Interaction	System log	Search, construction, editing, and AI-steering effort	[14]
AI chat turns per final path (Condition AI only)	Interaction	Chat log	Conversational steering effort	[14]
Usability and Perceived Quality				
Post-Condition Questionnaire	Survey	Familiarity, homework-quality, planning, time, control, and usability Likert items	Perceived quality, differentiation, task discovery, time sufficiency, usability, and control	Custom
Short usability battery	Survey	SUS-style items	Ease of use, confidence, complexity, integration, learnability	[5]
Qualitative				
Open-Ended Survey	Qualitative	Written responses (shared + condition-specific prompts; German fielded, English documented)	Task discovery mechanisms, planning quality, agency, condition-specific experience	Exploratory coding

Table A.7: Summary of homework learning-path artefact metrics.

Metric	**Data source**	**Measures**
Structural path metrics	Final learning paths	Nodes, differentiation gates, support gates, and option counts.
Exact three-option differentiation gate	Final learning paths	Whether the path visibly satisfies the task-specified three-group structure.
Interview-derived path audit	Final learning paths	Catalogue grounding, difficulty spread, skill coverage, valid references, bounded length, and duplicate task use.
Exploratory sequence and branch-design audit	Final learning paths	Common start, visual-to-symbolic ordering, branch-level support, challenge tasks, and duplicate-heavy structure.
Backfisch-inspired automated proxies	Final learning paths and audit outputs	Instructional-quality components and platform-affordance use/integration proxies.

Table A.8: Anonymised characteristics of the interest-form recruitment pool after duplicate removal.

Interest-form item	**Aggregate result**
Language comfort	24/24 selected German; 2/24 also selected English.
Study subjects	All 24 listed mathematics; 7 listed mathematics only and 17 listed mathematics with one or more additional school subjects.
Expected Referendariat start	2 within 6–12 months, 10 within 1–2 years, and 12 in more than two years.
Current programme	5 Bachelor of Education and 5 Master of Education responses; 14 unknown.
AI-tool use for studies	5 reported regular use and 5 occasional use; 14 unknown.

post-condition surveys. Thirteen of these were retained for the between-subjects condition comparison after excluding participant20 as a crossover exposure case. Participant20 was retained only for exploratory comparison with participant01, because those two pseudonymous records corresponded to the same participant seeing both conditions. Four early survey submissions did not contain the hidden participant identifier in the raw Tally payload. These submissions were linked to participant pseudonyms using the study access-token mapping and submission context. Participant07 had two completion rows; the later completion row was used. Participant03's session-linked path was empty; for this case, the latest non-empty path for the same token was used and flagged as a fallback. Participant26's selected path had been updated after the session-completion timestamp; after manual inspection, the latest non-empty selected path was retained and the caveat was recorded. These caveats affect the audit trail for individual cases, but they do not drive the condition-level pattern reported in Chapter 5.

A.5.2 Protocol-Deviation Crossover Case

Participant20 was excluded from the between-subjects condition comparison because this participant had seen both conditions. The case is useful only as a brief boundary check; it cannot estimate a treatment effect because prior exposure, practice, and AI assistance are confounded.

The participant's later AI-assisted record showed the same work-shift pattern as the main AI condition. Compared with the corresponding manual record, the AI-assisted path was larger and better perceived on overall quality and differentiation quality, but ease of use, confidence, and learnability were worse. The path contained more support structure, but it did not contain more differentiation gates or options. The interaction trace also shifted away from direct construction and toward AI suggestion mediation: the AI-assisted session included 6 accept-and-apply actions and 134 AI-suggestion normalisation events. The participant's written reflection was consistent with this pattern: the AI created a rough structure, but smaller adjustments were difficult to make satisfactorily.

This case therefore adds nuance without changing the main result. AI assistance may help a returning user produce a more elaborate draft, but it does not automatically solve the central control problem.

A.6 Additional Prototype Screenshots

Figures A.1, A.2, and A.3 document the prototype views that support the interface description in Chapter 3.

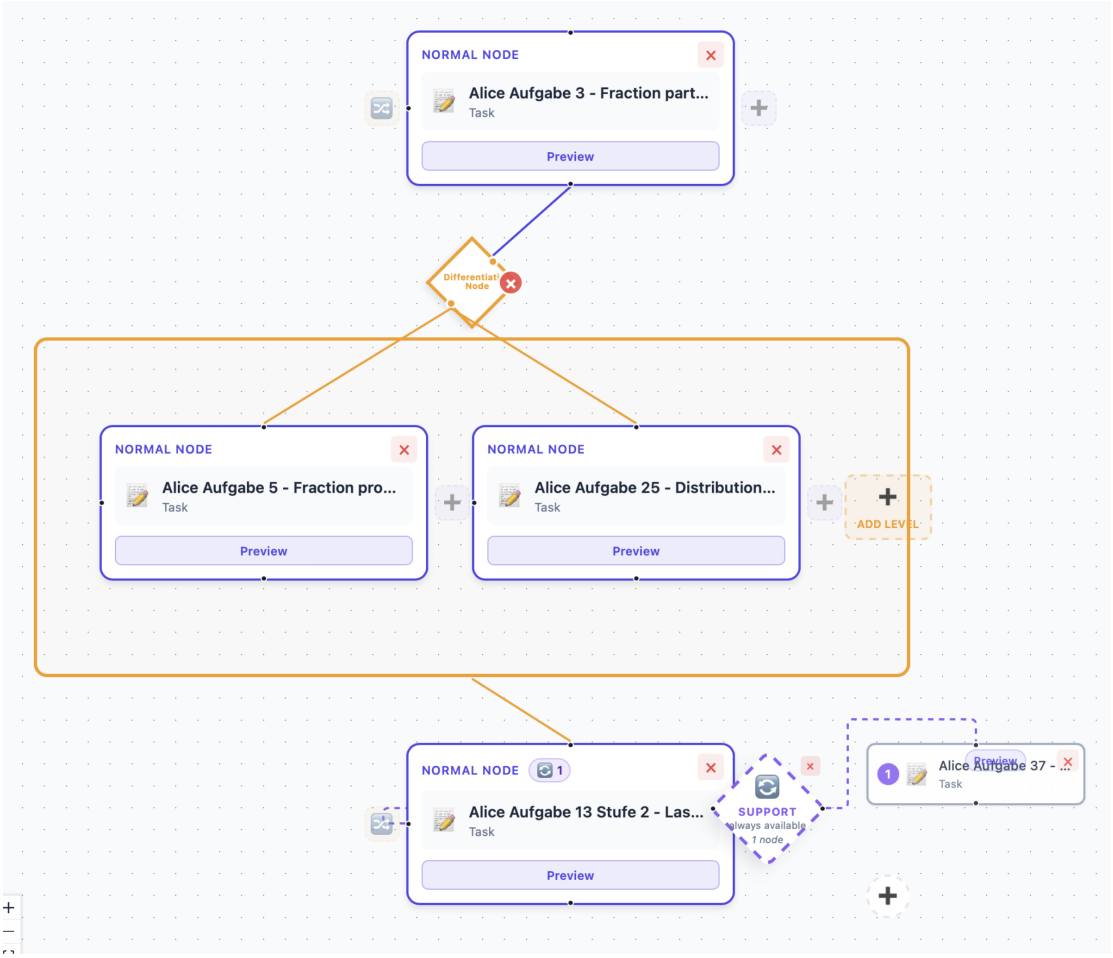


Figure A.1: Detailed screenshot of a homework learning path in the prototype editor. The view shows task nodes, a differentiation gate, grouped differentiated branches, and an optional support branch.

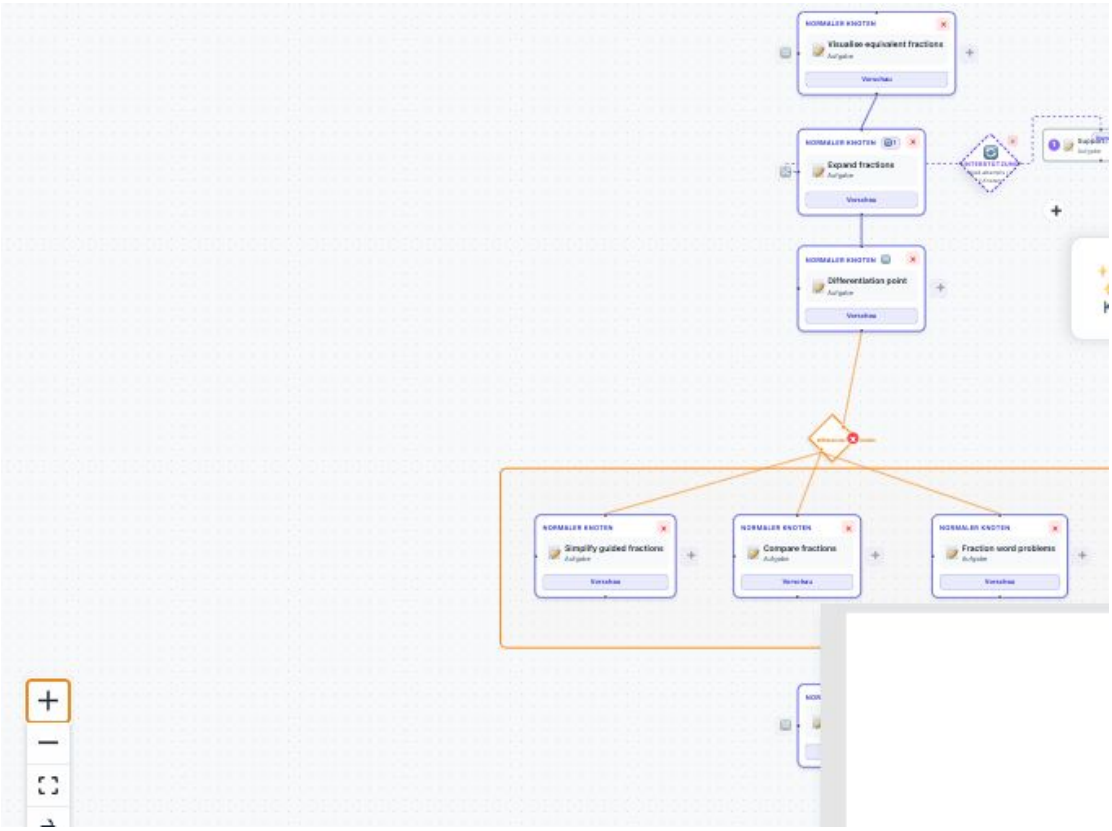


Figure A.2: Illustrative screenshot of the graph editor with a support branch and a three-option differentiation gate.

Figure A.3: Illustrative screenshot of the AI planning panel available only in the AI-assisted condition.

Bibliography

[1] Terry Anderson and Julie Shattuck. Design-based research: A decade of progress in education research? *Educational Researcher*, 41(1):16–25, 2012. doi: 10.3102/0013189X11428813.

[2] Iris Backfisch, Andreas Lachner, Christoff Hische, Frank Loose, and Katharina Scheiter. Professional knowledge or motivation? investigating the role of teachers' expertise on the quality of technology-enhanced lesson plans. *Learning and Instruction*, 66:101300, 2020. doi: 10.1016/j.learninstruc.2019.101300.

[3] Jürgen Baumert, Mareike Kunter, Werner Blum, Martin Brunner, Thamar Voss, Alexander Jordan, Uta Klusmann, Stefan Krauss, Michael Neubrand, and Yi-Miau Tsai. Teachers' mathematical knowledge, cognitive activation in the classroom, and student progress. *American Educational Research Journal*, 47(1):133–180, 2010. doi: 10.3102/0002831209345157.

[4] Gerhard Brandhofer and Karin Tengler. Acceptance of AI applications among teachers and student teachers. *Discover Education*, 4(1):213, 2025. doi: 10.1007/s44217-025-00637-w.

[5] John Brooke. SUS: A "quick and dirty" usability scale. In Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor & Francis, London, 1996. doi: 10.1201/9781498710411-35.

[6] Christopher Carroll, Malcolm Patterson, Stephen Wood, Andrew Booth, Jo Rick, and Shashi Balain. A conceptual framework for implementation fidelity. *Implementation Science*, 2(40), 2007. doi: 10.1186/1748-5908-2-40.

[7] Seongyune Choi, Yeonju Jang, and Hyeoncheol Kim. Influence of pedagogical beliefs and perceived trust on teachers' acceptance of educational artificial intelligence tools. *International Journal of Human-Computer Interaction*, 39(4):910–922, 2023. doi: 10.1080/10447318.2022.2049145.

[8] Doug Clow. The learning analytics cycle: Closing the loop effectively. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12)*, pages 134–138, New York, NY, USA, 2012. ACM. doi: 10.1145/2330601.2330636.

[9] David A. Cook and Rachel H. Ellaway. Evaluating technology-enhanced learning: A comprehensive framework. *Medical Teacher*, 37(10):961–970, 2015. doi: 10.3109/0142159X.2015.1009024.

[10] John W. Creswell and J. David Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, Thousand Oaks, CA, 5th edition, 2017.

[11] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989. doi: 10.2307/249008.

[12] Deepak Varuvel Dennison, Bakhtawar Ahtisham, Kavyansh Chourasia, Nirmit Arora, Rahul Singh, Rene F. Kizilcec, Akshay Nambi, Tanuja Ganu, and Aditya Vashistha. Shiksha Copilot: Teacher-AI collaboration for curating and customizing lesson plans in low-resource schools, 2025. URL https://arxiv.org/abs/2507.00456.

[13] Ernst Klett Verlag. *Schnittpunkt Mathematik 6G: Differenzierende Ausgabe Baden-Wuerttemberg ab 2025. Schulbuch mit Medien.* Ernst Klett Verlag, Stuttgart, 2026. ISBN 978-3-12-745341-6. URL https://www.klett.de/produkt/isbn/978-3-12-745341-6. Official Klett product metadata lists Klasse 6 (Grundlegendes Niveau) and Erscheinungsdatum 2026-05-06.

[14] Haoxiang Fan, Guanzheng Chen, Xingbo Wang, and Zhenhui Peng. LessonPlanner: Assisting novice teachers to prepare pedagogy-driven lesson plans with large language models. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*, pages 1–17, Pittsburgh, PA, USA, October 2024. ACM. doi: 10.1145/3654777.3676390.

[15] Yael Feldman-Maggor, Mutlu Cukurova, Carmel Kent, and Giora Alexandron. The impact of explainable AI on teachers' trust and acceptance of AI edtech recommendations: The power of domain-specific explanations. *International Journal of Artificial Intelligence in Education*, 35(5):2889–2922, 2025. doi: 10.1007/s40593-025-00486-6.

[16] Samantha R. Goldman, Juli Taylor, Adam Carreon, and Sean J. Smith. Using AI to support special education teacher workload. *Journal of Special Education Technology*, 39(3):434–447, 2024. doi: 10.1177/01626434241257240.

[17] Jim Greer and Mary Ann Mark. Evaluation methods for intelligent tutoring systems revisited. *International Journal of Artificial Intelligence in Education*, 26(1):387–392, 2015. doi: 10.1007/s40593-015-0043-2.

[18] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, pages 139–183. Elsevier, Amsterdam, 1988. doi: 10.1016/S0166-4115(08)62386-9.

[19] Stephanie Houde and Charles Hill. What do prototypes prototype? In Martin G. Helander, Thomas K. Landauer, and Prasad V. Prabhu, editors, *Handbook of Human-Computer Interaction*, pages 367–381. Elsevier, 1997. doi: 10.1016/B978-044481862-1.50082-0.

[20] Hassan Khosravi, Simon Buckingham Shum, Guanliang Chen, Cristina Conati, Yi-Shan Tsai, Judy Kay, Simon Knight, Roberto Martinez-Maldonado, Shazia Sadiq, and Dragan Gašević. Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 3:100074, 2022. doi: 10.1016/j.caeai.2022.100074.

[21] Land Baden-Württemberg, Zentrum für Schulqualität und Lehrerbildung. MatheBattle. https://mathebattle.de/cms_pages/view/about_mathebattle, 2026. Accessed 2026-06-23.

[22] Tanya Nazaretsky, Moriah Ariely, Mutlu Cukurova, and Giora Alexandron. Teachers' trust in AI-powered educational technology and a professional development program to improve it. *British Journal of Educational Technology*, 53(4):914–931, 2022. doi: 10.1111/bjet.13232.

[23] Chee Ng and Yuen Fung. Educational personalized learning path planning with large language models. *arXiv preprint*, arXiv:2407.11773, 2024. URL https://arxiv.org/abs/2407.11773.

[24] Carol L. O'Donnell. Defining, conceptualizing, and measuring fidelity of implementation and its relationship to outcomes in K–12 curriculum intervention research. *Review of Educational Research*, 78(1):33–84, 2008. doi: 10.3102/0034654307313793.

[25] Raja Parasuraman and Dietrich Manzey. Complacency and bias in human use of automation: An attentional integration. *Human Factors*, 52(3):381–410, 2010. doi: 10.1177/0018720810376055.

[26] Irfan Ahmed Rind. Conceptualizing the impact of AI on teacher knowledge and expertise: A cognitive load perspective. *Education Sciences*, 16(1):57, 2026. doi: 10.3390/educsci16010057.

[27] Annemieke E. Smale-Jacobse, Anna Meijer, Michelle Helms-Lorenz, and Ridwan Maulana. Differentiated instruction in secondary education: A systematic review of research evidence. *Frontiers in Psychology*, 10:2366, 2019. doi: 10.3389/fpsyg.2019.02366.

[28] Margaret Schwan Smith and Mary Kay Stein. Reflections on practice: Selecting and creating mathematical tasks: From research to practice. *Mathematics Teaching in the Middle School*, 3(5):344–350, 1998. doi: 10.5951/mtms.3.5.0344.

[29] Technische Universität München. ALICE:Bruchrechnen. https://www.alice.edu.tum.de/, 2026. Accessed 2026-06-23.

[30] Carol Ann Tomlinson. *How to Differentiate Instruction in Mixed-Ability Classrooms*. ASCD, Alexandria, VA, 2nd edition, 2001.

[31] Geesje van den Berg and Elize du Plessis. ChatGPT and generative AI: Possibilities for its contribution to lesson planning, critical thinking and openness in teacher education. *Education Sciences*, 13(10):998, 2023. doi: 10.3390/educsci13100998.

[32] Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011. doi: 10.1080/00461520.2011.611369.

[33] Lev S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978. Edited by Michael Cole, Vera John-Steiner, Sylvia Scribner, and Ellen Souberman.

[34] Feng Wang and Michael J. Hannafin. Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4): 5–23, 2005. doi: 10.1007/BF02504682.

[35] Rose E. Wang, Ana T. Ribeiro, Carly D. Robinson, Susanna Loeb, and Dorottya Demszky. Tutor CoPilot: A human-AI approach for scaling real-time expertise. *arXiv preprint*, arXiv:2410.03017, 2024. URL https://arxiv.org/abs/2410.03017.

[36] David Wood, Jerome S. Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100, 1976. doi: 10.1111/j.1469-7610.1976.tb00381.x.

[37] Liangyong Xue, Jazihan Mahat, and Norliza Ghazali. Technology acceptance model in artificial intelligence in education: A meta-analysis. *SAGE Open*, 2026. doi: 10.1177/21582440251409441.

[38] Xiaoming Zhai and Ross H. Nehm. AI and formative assessment: The train has left the station. *Journal of Research in Science Teaching*, 2023. doi: 10.1002/tea.21885.

[39] Chengming Zhang, Jessica Schießl, Lea Plößl, Florian Hofmann, and Michaela Gläser-Zikuda. Acceptance of artificial intelligence among pre-service teachers: A multigroup analysis. *International Journal of Educational Technology in Higher Education*, 20 (1):49, 2023. doi: 10.1186/s41239-023-00420-7.

Declaration

According to resolutions of the Boards of Examiners for Bioinformatics, Computer Science, Computer Science Teaching Degrees, Cognitive Science, Machine Leaning, Media Informatics and Medical Informatics of the University of Tübingen dated 05.02.2025. Valid for final theses (B.Sc./M.Sc./B.Ed./M.Ed.) in the corresponding subjects. In the case of student research projects and term papers, please do so in accordance with the respective examiner.

1. General declarations

I hereby declare that:

- I have written this thesis independently and I have not used any sources or resources other than those specified.
- I have clearly identified as such all matter from other works, whether cited verbatim or paraphrased.
- The thesis was not the subject of any other examination procedure neither in its entirety nor in significant parts.
- In case I submitted an electronic file and one or more printed and bound copies (e.g. because the examiner(s) requested this): The electronically submitted file is identical to the printed and bound copy(s) I submitted.

2. Declaration concerning publications

A publication is often a sign of quality (e.g. publication in a scientific journal, a conference, preprint, etc.). However, it must be stated correctly. Please tick the box that applies to your work:

- ☒ The work/thesis has not yet been published in full or in part.
- ☐ The work/thesis has been published in part or in full. A complete table with bibliographic details can be found in the appendix.

3. Using methods of artificial intelligence (AI, e.g. ChatGPT, DeepL, etc.)

Using AI can be useful. However, such usage must be declared correctly and can influence the focus of the evaluation of the thesis. Please tick all variants that are applicable to your thesis and please note that variants 3.4 – 3.6 require prior consultation with the supervisor:

- ☐ 3.1. No usage: I did not use AI to create my thesis.
- ☐ 3.2. Spelling & grammar correction: I used AI for correcting spelling and grammar without any relevant text generation or translations, i.e., I have had texts written by me corrected in the same language. These are purely linguistic corrections so that the meaning I originally intended has not been changed or expanded significantly. In case of doubt, I discussed this with my supervisor. All programmes used and their version numbers are listed in a table in the appendix of my thesis.
- ☒ 3.3. Support with developing software: I used AI to help me write code in software development. This is merely used as an aid and not to automatically generate larger programme parts. In case of doubt, I discussed this with my supervisor. All programmes used and their version numbers are listed in a table in the appendix of my thesis.

☒ 3.4. Translation: I used AI for translating texts I had written in a different language *by prior consultation and with the permission of my supervisor*. Each such translation has been labelled in the running text and the appendix of my work contains a table with a complete list of translated text passages and the programmes used with version number.

☒ 3.5. Code generation: I used AI for generating code in software development *by prior consultation and with the permission of my supervisor*. The appendix of my thesis contains a table with a complete record of all such uses, the programmes used with version numbers and the prompts used.

☒ 3.6. Text generation: I used AI for generating text in my work *by prior consultation and with the permission of my supervisor*. Each such usage of AI has been labelled in the running text and the appendix of my work contains a table with a complete record of all such uses, the programmes used with version number and the prompts used.

In case I have used AI in any form (see above), then I declare:

I am aware that I am responsible if the use of AI results in incorrect content, violations of data protection law, copyright law or scientific misconduct (e.g. plagiarism).

4. Degree and signature(s)

I am aware that a breach of this declaration may have consequences under examination law and in particular may result in the examination being assessed as 'insufficient' or the coursework being assessed as 'failed'. Furthermore, in the event of multiple or serious attempts to cheat, this may result in deregistration ("Exmatrikulation"), or proceedings may be initiated to withdraw any academic title awarded and affected by this.

APOORV, AGNIHOTRI	TÜBINGEN, 25-06-26	
First name, Last name Student	Location, Date	Signature

Items 3.4 - 3.6 require the examiner's agreement. Should you have ticked these, then your supervisor has to sign here:

I have agreed to the above-mentioned use of AI for the preparation of the thesis.

WIELAND, BRENDEL	TÜBINGEN, 26-06-26	
First name, Last name Examiner	Location, Date	Signature