DTU Civil Engineering
Department of Applied Mathematics and Computer Science

Assessing Exam Validity with Fine-tuned Local Language Models

Master Thesis

Assessing Exam Validity with Fine-tuned Local Language Models

Master Thesis
November, 2025

By
Zhengping Qiao

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Applied Mathematics and Computer Science, Richard Petersens Plads, Building 324, 2800 Kgs. Lyngby Denmark
https://www.compute.dtu.dk/

Approval

This thesis has been prepared over six months at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Engineering, MSc Eng.

It is assumed that the reader has a basic knowledge in the areas of deep learning.

Zhengping Qiao - s232266

.....

Signature

.....

Date

Abstract

As leading online language models such as ChatGPT and Claude increasingly demonstrate expert-level performance on tertiary education assessments, universities have widely adopted closed-internet exam policies. However, students with offline-capable computers can potentially circumvent these restrictions using local LLMs. This thesis evaluates the capabilities and limitations of local language models on university-level probability assessments, using an original dataset of 660 questions from DTU course 02405 (2003–2024).

We explore three research questions on model optimization and evaluation:

RQ1: Can fine-tuning improve local model performance on domain-specific exams?

RQ2: How does Retrieval-Augmented Generation (RAG) compare to fine-tuning as an optimization strategy?

RQ3: How do optimized local models compare to baseline LLMs and commercial LLMs performance?

To answer RQ1, we implemented QLoRA fine-tuning using 4,488 synthetically generated question–answer pairs and evaluated multiple PDF-to-text extraction pipelines for knowledge acquisition. For RQ2, we developed a RAG system that incorporates course materials and historical exams. Contrary to expectations, fine-tuning decreased performance by approximately 8 percentage points relative to the 77.0% baseline, indicating that the generated training data and extraction methods introduced noise rather than beneficial domain knowledge. In contrast, the RAG pipeline substantially improved performance to 83.3%, reducing the gap to commercial LLMs and high-achieving students.

These findings show that strategic knowledge retrieval can significantly outperform direct parameter modification for complex reasoning tasks. Well-curated RAG pipelines enable local LLMs to perform competitively even under offline exam conditions, while naïve fine-tuning offers little benefit and can even degrade capability. The results highlight knowledge curation and retrieval architecture, rather than model fine-tuning, as the critical factor for educational applications of LLMs.

Contents

1 Introduction

The introduction of large language models (LLMs) such as ChatGPT has fundamentally disrupted educational assessment across all academic domains. These models demonstrate performance levels that match or exceed competent students in programming, mathematics, and essay writing. Research showed that GPT-4 performs comparably to humans on most multiple choice tests in Higher Education[1]. GPT-4 scored in the 90th percentile on the Uniform Bar Exam [2, 3] and exceeded student averages on seven of nine graduate-level biomedical science examinations [4, 5]. On Brazilian university admission exams, GPT-4 achieved 87% accuracy, significantly outperforming GPT-3.5 [6], while comparative studies across statistics examinations revealed substantial performance differences between model versions [7].

The rapid emergence of AI in education has raised concerns that ”AI slowly eliminates the necessity to acquire knowledge” [8]. Studies demonstrate that overreliance on AI diminishes skills in independent problem solving [9]. As these capabilities evolve, the educational community faces pressure to reassess fundamental assumptions about academic assessment and integrity. Systematic reviews of AI's impact on higher education revealed a complex dual role: AI tools support learning while posing risks to traditional academic standards [10]. This analysis underscored threats to academic integrity [11], suggesting that examination protocols may be inadequately equipped to address the evolving landscape of AI-assisted academic activities.

1.1 The Challenge of AI in Academic Assessment

Universities and teachers have been evaluating how exams can be adapted to this new reality. The DTU chattutor [12] and the UIUC course chat system [13] exemplify institutional responses, combining LLMs with retrieval-augmented generation to provide students with course-specific assistance. DTU's response has predominantly been to adapt or continue the ”closed internet” practice, which prohibits the use of the best language models that are not available and cannot be run on consumer hardware [14].

However, students can still use LLMs that run on their laptops (local LLMs). To clarify the distinction: local LLMs are language models that operate entirely offline on consumer hardware, without requiring internet connectivity or API access. Unlike cloud-based models such as GPT-4 or Claude that process requests on remote servers, local models like Llama, Mistral, or Phi run directly on a user's laptop using only its CPU or GPU resources. These models can be downloaded once and used indefinitely without detection, making them particularly relevant for ”closed internet” examination settings.

Teachers generally assume these local models aren't powerful enough to pose a real

problem, but this assumption had not been properly tested before this study. The advancement of local LLMs may have already altered the risk profile of academic integrity in ways that educators have not recognized.

Recent developments in fine-tuning methodologies demonstrated that smaller, specialized language models can achieve performance comparable to much larger systems through targeted optimization [15]. Comprehensive reviews of fine-tuning techniques revealed sophisticated approaches including parameter-efficient methods like Low-Rank Adaptation (LoRA), supervised fine-tuning, and instruction tuning that can dramatically enhance model capabilities on specific tasks [16, 17]. Practical guidelines for enterprise LLM deployment suggested that domain-specific fine-tuning can yield substantial performance improvements even with limited computational resources [18].

The educational domain has witnessed particular success with specialized fine-tuning approaches. Studies showed that supervised fine-tuning can transform general-purpose models into effective pedagogical agents, with applications ranging from programming education [19, 20] to automated essay scoring [21]. Novel contextual fine-tuning methods based on educational theories demonstrated enhanced learning capabilities that align with pedagogical principles [22], while comparative assessment tasks benefited from targeted optimization strategies [23].

The gap between perception and reality regarding local LLM capabilities represents a fundamental blind spot in contemporary academic integrity policies. If students can achieve passing or superior grades using undetectable local AI assistance, then current examination protocols provide false security while failing to maintain the intended assessment standards.

1.2 Research Gap and Objectives

Despite extensive research on cloud-based LLM performance in academic contexts [24, 25], there existed a significant gap in understanding the capabilities of local models when applied to university-level coursework. Previous studies focused primarily on commercial APIs and online services, leaving the potential of locally-deployed models largely unexplored.

While automated educational assessment systems showed promise [26], and LLM-based grading demonstrated effectiveness in various contexts including handwritten mathematical solutions [27], the specific threat posed by optimized local models remained unquantified. Current evaluation frameworks focused on measuring LLM capabilities through standardized benchmarks [28], but failed to address the practical implications of accessible, offline AI assistance in examination settings.

The potential for enhancing local model performance through retrieval-augmented generation (RAG) techniques introduced additional complexity. RAG systems, which combine

pre-trained models with external knowledge bases, have shown remarkable success in knowledge-intensive tasks [29, 30]. Comprehensive surveys highlighted the evolution from naive to advanced RAG implementations [31, 32], with recent developments including agentic RAG systems that can dynamically adapt retrieval strategies [33]. Best practices research identified optimal configurations for various RAG components [34], while natural language processing applications demonstrated broad applicability across domains [35].

This project addressed this critical knowledge gap by systematically evaluating the performance of local LLMs on bachelor-level coursework. The primary objective was to determine whether local models, when optimized through various techniques including fine-tuning and RAG implementation, could achieve performance levels comparable to typical student outcomes on university examinations.

The investigation focused on determining: (1) the baseline performance of out-of-the-box local LLMs on academic assessments, (2) the effectiveness of optimization strategies in improving local model performance, and (3) how optimized local LLMs compared to both cloud-based models and actual student performance.

1.3 Research Questions

This study was guided by three primary research questions:

1. **Fine-tuning Effectiveness**: To what extent can local LLMs be fine-tuned to improve performance on domain-specific exam tasks, given realistic constraints on data availability and computational resources?

2. **Optimization Strategy Evaluation**: How do retrieval-augmented generation (RAG) and fine-tuning compare to prompt engineering alone in improving exam-related task performance, in terms of both accuracy and computational cost?

3. **Comparative Assessment**: How do optimized local models (e.g., with RAG and/or fine-tuning) perform relative to (a) baseline local models, (b) state-of-the-art cloud-based models (e.g., GPT or Claude), and (c) typical student exam outcomes?

1.4 Approach Overview

This project examined the performance of local LLMs in bachelor-level coursework when applied by a user who was assumed to have no relevant skills in their course. The underlying question was straightforward: could a student with no domain knowledge pass an exam simply by running a local LLM on their laptop?

The study focused on suitable local LLMs that could be run on consumer hardware (or hardware expected to be available in consumer laptops within the next few years). These models were adapted to solve exam questions and subsequently assessed based on the

same criteria used to evaluate regular students. The courses examined included DTU course 02405 [36].

1.4.1 Optimization Strategies Employed

The experimental design encompassed multiple optimization strategies based on current best practices in the field:

Baseline Testing: We first conducted baseline testing of unmodified local models to establish a performance floor. Models were given exam questions directly without any course-specific preparation.

Supervised Fine-tuning: We implemented supervised fine-tuning on course-specific materials following established pedagogical approaches [19]. This involved training models on lecture notes, textbook chapters, and worked examples from the course.

Retrieval-Augmented Generation (RAG): We deployed RAG systems that allowed models to access course materials during exam-solving. The RAG implementation incorporated lecture notes, textbooks, and previous exam solutions into a searchable knowledge base that the model could query when formulating responses [34].

Each approach was evaluated using the same grading criteria applied to human students, enabling direct performance comparisons. The assessment framework drew from recent advances in automated educational evaluation [26], incorporating both quantitative metrics and qualitative analysis of response quality.

1.4.2 What Does It Mean to ”Solve” an Exam?

To illustrate what solving an exam means in this context, consider an actual question from DTU course 02405:

Exercise 19

The bacteria count of hospital patients is continuously monitored. The mean bacteria count among hospital patients is 3.

Question 19

The probability, that a patient has a bacteria count of at least 60, is at most

1 ☐ $1 - \Phi\left(\frac{60-3}{\sqrt{60}}\right)$

2 ☐ $1 - \Phi\left(\frac{60-3}{60}\right)$

3 ☐ $\frac{1}{400}$

4 ☐ $\frac{1}{20}$

5 ☐ $\frac{1}{9}$

6 ☐ Do not know

Figure 1.1: Example: Question 19 from Exam 2016-05-30

> *Context: "The bacteria count of hospital patients is continuously monitored. The mean bacteria count among hospital patients is 3."*
>
> *Question: "The probability, that a patient has a bacteria count of at least 60, is at most"*
>
> *Options: (1)* $1 - \Phi\left(\frac{60-3}{\sqrt{60}}\right)$*, (2)* $1 - \Phi\left(\frac{60-3}{60}\right)$*, (3)* $\frac{1}{400}$*, (4)* $\frac{1}{20}$*, (5)* $\frac{1}{9}$*, (6) Do not know*

A baseline local LLM without any optimization might produce a response like:

> *"We have a mean of 3 and need to find P(X ≥ 60). This seems like a normal distribution problem. Using the normal approximation with mean 3, the probability of being at least 60 would involve standardizing: (60-3)/σ. Since we don't know the standard deviation, maybe it's √60? So I'll choose option (1):* $1 - \Phi\left(\frac{60-3}{\sqrt{60}}\right)$*."*

This response shows a common error: assuming a specific distribution without justification. The model incorrectly applies normal distribution formulas when the problem doesn't provide enough information to assume normality, and guesses at the standard deviation.

In contrast, after fine-tuning on course materials and previous exams, the same model produced:

> *"Given only the mean E[X] = 3 for a non-negative random variable (bacteria count), we cannot assume a specific distribution. We must use Markov's inequality, which provides an upper bound for any non-negative random vari-*

able:

For $X \geq 0$ and $a > 0$: $P(X \geq a) \leq E[X]/a$

Therefore: $P(X \geq 60) \leq 3/60 = 1/20$

The answer is option (4): 1/20."

This enhanced response demonstrates proper problem identification (recognizing when to use Markov's inequality instead of assuming a distribution), correct application of probability inequalities, and arrives at the correct answer—meeting all the grading criteria used for human students. The improvement from baseline to fine-tuned model illustrates the potential risk that optimized local LLMs pose to exam integrity, particularly in their ability to recognize which mathematical tools to apply when given limited information.

1.5 Related Work

While this study focuses specifically on local LLM performance in academic settings, several related research streams inform our work:

LLM Performance in Education: Extensive research has documented the capabilities of cloud-based LLMs in educational contexts. Studies demonstrated GPT-4's success on standardized tests [1, 2], medical examinations [4], and various academic assessments [6, 7]. However, these studies primarily evaluated API-based models requiring internet connectivity, leaving the capabilities of offline models unexplored.

Fine-tuning for Educational Applications: Research on fine-tuning LLMs for educational purposes has shown promising results. Studies transformed general models into pedagogical agents [19, 20] and demonstrated success in automated essay scoring [21]. Our work extends these findings by specifically examining how fine-tuning affects exam-solving capabilities in offline settings.

RAG Systems in Knowledge-Intensive Tasks: The development of RAG systems has revolutionized how LLMs handle knowledge-intensive tasks [29, 30]. Advanced implementations showed particular promise in educational contexts [12, 13]. Our study adapts these techniques for local deployment, examining whether RAG can compensate for the smaller size of local models.

Academic Integrity and AI Detection: Parallel research examined methods for detecting AI-generated content and maintaining academic integrity [11, 10]. Our findings contribute to this discourse by quantifying the actual threat posed by undetectable local AI assistance.

The key distinction of our work lies in its focus on local, offline models that evade current detection methods while potentially achieving performance comparable to cloud-based systems. This addresses a critical blind spot in existing academic integrity frameworks.

1.6 Thesis Structure

The remainder of this thesis is organized as follows:

Chapter 2: Data Collection and Processing describes the assembly of the evaluation corpus from DTU 02405 exams (2003–2024), the OCR workflow (Nougat with MathPix for flagged expressions), automated quality checks, and human review.

Chapter 3: Methodology presents the experimental setup, including the selection of local models, fine-tuning procedures, RAG implementation details, and evaluation criteria. This chapter also describes the exam datasets and grading rubrics used.

Chapter 4: Experimental Results reports the performance of various model configurations on actual exam questions. Results are presented for baseline models, fine-tuned variants, and RAG-enhanced systems, with detailed comparisons to both cloud-based models and human student performance.

Chapter 5: Analysis and Discussion interprets the experimental findings, examining which optimization strategies proved most effective and discussing the implications for academic assessment. This chapter also addresses the limitations of local models and scenarios where they fall short.

Chapter 6: Conclusion summarizes the key findings, revisits the research questions, and suggests directions for future research in this rapidly evolving field.

2 Data Collection and Processing

This chapter describes the comprehensive data collection, processing, and augmentation methodology employed to create a robust dataset for evaluating local large language models on university-level probability and statistics examinations. The process encompassed historical examination collection spanning from 2003 to 2024 ——an exceptionally long temporal coverage for a single course—multi-modal optical character recognition, expert validation, and systematic data augmentation applied specifically to the training subset to create a dataset suitable for fine-tuning and evaluation. The chapter concludes with 660 original examination questions, of which the training partition was augmented to create an expanded training set.

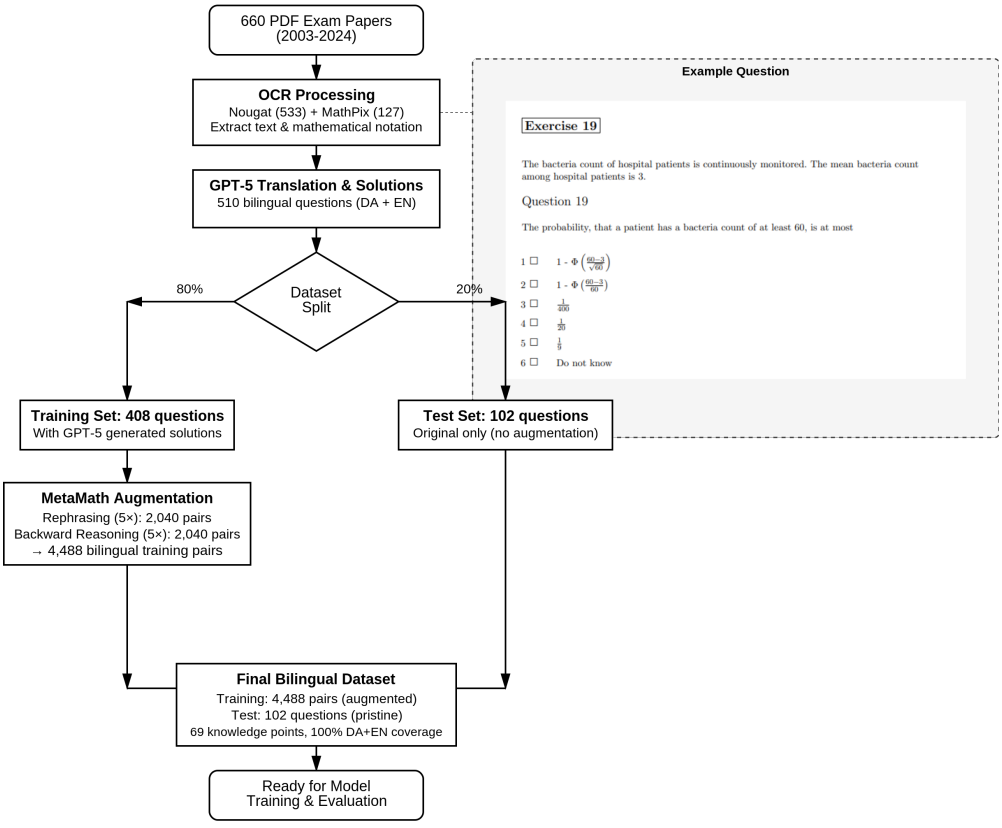

Figure 2.1: Process of data handling

2.1 Historical Examination Collection

Our dataset is based on 17 examinations from DTU course 02405 (Probability Theory and Statistics), collected from **2003 to 2024**. This long time span is rare for a single course dataset and allows us to track how assessment methods have evolved over two decades.

The dataset contains **660** examination questions in total. Of these, **510** questions (77.3%) are in Danish and **150** (22.7%) are in English, with the English questions being translations of Danish originals. All examinations were originally provided as PDF documents containing both text and mathematical notation.

We collected the examination papers from DTU's official course repository and organized them by year and semester. For each exam, we recorded the date, language, number of questions, and whether the document quality was suitable for OCR processing. The data shows a clear shift in language use: examinations were exclusively in Danish from 2003 to 2015, then gradually included English versions from 2016 onwards, reflecting the increasing international student enrollment.

2.2 Optical Character Recognition and Text Extraction

We used a two-stage OCR approach to extract text and mathematical formulas from the PDF examination papers. Since academic documents with mathematical notation are challenging for standard OCR tools, we combined two specialized systems: Nougat for general document processing and MathPix for complex mathematical expressions.

2.2.1 Primary Processing with Nougat

Nougat (Neural Optical Understanding for Academic Documents) [37] is a Visual Transformer model developed by Meta AI specifically for converting academic PDFs into markup language. Unlike traditional OCR tools that struggle with mathematical notation, Nougat was trained on millions of scientific papers from arXiv and can directly recognize both text and LaTeX formulas without requiring preprocessing steps like layout detection or formula segmentation.

We first converted all PDF pages to 300 DPI images and processed them through Nougat. The model outputs markdown-formatted text with embedded LaTeX for mathematical expressions. For example, it correctly converts a probability formula image into $P(X > k) = \sum_{i=k+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$ rather than garbled text.

2.2.2 Quality Control and Error Correction

After the initial Nougat processing, we ran automated checks to identify potential errors. These checks included validating LaTeX syntax, detecting missing brackets in formulas, and verifying that Danish and English text was correctly recognized. We also used Nougat's confidence scores to flag uncertain regions for manual review.

Through this process, we identified 127 questions (19.2% of the dataset) that needed additional attention. These were mainly questions with complex statistical formulas, matrix notation, or Danish mathematical terminology that Nougat had trouble with. Common issues included confusion between similar symbols (like Greek letters ρ and p), misaligned fraction bars, and incorrect subscript/superscript placement.

2.2.3 Enhanced Processing with MathPix

For these 127 problematic questions, we used MathPix OCR [38], a commercial service that specializes in mathematical content. MathPix uses a different approach than Nougat—it's specifically optimized for handwritten and printed mathematical expressions and can handle complex layouts like multi-line equations and matrices. While Nougat processes entire documents, MathPix works best on isolated mathematical expressions, making it ideal for targeted corrections.

We manually extracted the problematic formula regions and processed them through MathPix's API. The service returns clean LaTeX code with high accuracy for mathematical notation. We then replaced the corresponding parts in the Nougat output with MathPix's results. When the two systems disagreed on an interpretation, we manually reviewed the original PDF to determine the correct version.

This hybrid approach—using Nougat for document structure and general text, supplemented by MathPix for difficult mathematics—gave us much better results than either system alone. The combination ensured accurate extraction of both the Danish/English text and the complex statistical formulas throughout the 660 examination questions.

2.3 Bilingual Translation Pipeline

Following OCR processing and quality validation, all examination questions were processed through an automated translation pipeline to ensure complete Danish-English coverage across the dataset.

2.3.1 Translation Implementation

Translation Example

Input (Danish): "En mønt med sandsynlighed $p = 0.6$ for plat..."
Output (English): "A coin with probability $p = 0.6$ for heads..."
Validation: □ LaTeX preserved, □ Numbers unchanged, □ 6 options maintained

We employed GPT-5 for automated Danish-to-English translation with strict constraints to preserve mathematical content.

System Prompt:

System Prompt

You are a professional translator for math exam items.
Rules:

1. Copy ALL LaTeX segments (`$...$` or `\[...\]`) EXACTLY as-is
2. Do NOT alter any numbers/fractions/decimals; keep decimal comma if present
3. Keep the options' order and count unchanged
4. Translate only natural language; do not change meaning
5. Output MUST follow the provided JSON schema strictly

2.4 Expert Solution Generation and Validation

2.4.1 GPT-5 Solution Generation

Example: Question exam_2021_12_15-30

Context: It is assumed that shooters' skill is described by a (real) number on a scale. The skill of a random shooter is standard normally distributed. It is further given that the probability that a shooter with skill k achieves at least 200 points is $1/(2+k^2)$.

Question: The probability that a randomly chosen shooter achieves at least 200 points in a shooting competition is
Options: (1) $\square \quad \Phi\left(-\frac{5\sqrt{2}}{8}\right)$, (2) $\int_{-1/(1+k^2)}^{1/(1+k^2)} \frac{1}{2+x^2}\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\,dx$, (3) $\int_{-\infty}^{\infty} \frac{1}{2+x^2}\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\,dx$, (4) $2049/3125$, (5) $2/3$, (6) Do not know

Solution Process:

- **Step 1:** Let $K \sim N(0,1)$ be the skill of a random shooter
- **Step 2:** By the law of total probability (tower property):

$$P(\text{score} \geq 200) = \mathbb{E}\Big[P(\text{score} \geq 200 \mid K)\Big] = \mathbb{E}\Big[\frac{1}{2+K^2}\Big]$$

- **Step 3:** Express as an integral over the standard normal density:

$$P(\text{score} \geq 200) = \int_{-\infty}^{\infty} \frac{1}{2+x^2}\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\,dx$$

- **Final answer:** Option 3

Key Formula:

$$P(\text{score} \geq 200) = \int_{-\infty}^{\infty} \frac{1}{2+x^2}\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\,dx$$

Common Pitfalls:

- Plugging in a fixed k instead of taking expectation over K
- Using finite or incorrect integration limits

Knowledge Points: Law of total probability, normal distribution, expectation via integration

To create comprehensive solution sets for model training and evaluation, we employed GPT-5 to generate detailed solutions for all examination questions.

Prompt:

Prompt

You are an expert statistics professor tasked with solving a multiple-choice examination question from DTU course 02405 (Statistics).
Please provide a complete, pedagogical solution that includes:

1. **Problem Identification**: Identify the statistical concept being tested and select the appropriate approach
2. **Step-by-Step Reasoning**: Show all mathematical steps with clear explanations
3. **Explicit Calculations**: Display all formulas and computations
4. **Final Answer**: State the correct option number and its mathematical expression
5. **Conceptual Explanation**: Briefly explain the key statistical principles involved
6. **Common Pitfalls**: Note any common mistakes students might make

Context: {context}
Question: {question}
Available Options:
{formatted_options}

2.4.2 Solution Quality Control

All generated solutions underwent automated validation to ensure mathematical correctness and quality. The validation pipeline checked that each solution contained complete structure with problem introduction, step-by-step solving process, and final answers. It also verified consistency in mathematical notation throughout the solution, validated answer formats against expected question types, and cross-referenced solutions with official grading rubrics where available.

2.4.3 Knowledge Point Annotation

To enable fine-grained analysis of model performance across different concepts, we systematically annotated each question with the knowledge points required for its solution.

We tagged each question with one or more knowledge points representing the core concepts needed. The annotation taxonomy covers five main categories: **distribution types** (normal, exponential, binomial, Poisson, geometric, hypergeometric, gamma, beta, uniform), **statistical techniques** (standardization, conditional probability, law of total probability and expectation), **transformation methods** (change of variables, order statistics, marginalization), **probability rules** (complement rule, inclusion-exclusion principle, multiplication rule), and **moment calculations** (linearity of expectation, variance of sums, covariance relationships).

Table 2.1 shows the 10 most frequent knowledge points in the dataset. Normal distribution, exponential distribution, and binomial distribution are the three most common concepts, appearing in 67, 65, and 64 questions respectively. The top 6 concepts account

for 372 total occurrences, demonstrating strong coverage of fundamental probability concepts.

Table 2.1: Top 10 Knowledge Points by Frequency Across Train and Test Sets

Knowledge Point	Overall	Train	Test
Normal distribution	67	49	18
Exponential distribution	65	50	15
Binomial distribution	64	49	15
Standardization to standard normal (Z-score)	63	51	12
Conditional probability (multiplication rule)	59	51	8
Poisson distribution	54	43	11
Law of total probability	42	34	8
Marginal density from joint (integration)	35	30	5
Variance of sum using covariance	34	28	6
Change of variables (univariate)	29	25	4

We validated that the train-test split maintains proper knowledge coverage. The dataset contains 69 distinct knowledge points in total. Among these, 65 concepts (94.2%) appear in both training and test sets, ensuring models can be trained on all concepts they will encounter during testing. Four concepts appear only in the training set (expectation as constant, expected absolute value of standard normal, probability from joint CDF, and variance from PMF), representing rare edge cases. No knowledge points appear exclusively in the test set, preventing unfair evaluation scenarios where models encounter completely new concepts during testing.

This annotation structure enables subsequent analysis of model performance by concept category, difficulty level, and reasoning type required for each question.

2.5 Data Augmentation Using MetaMath Methodology

Data Augmentation Example

Augmentation Methods Applied:

1. *Rephrasing (5 variants):* Change scenario/numbers, keep distribution
2. *Backward Reasoning (5 variants):* Generate questions from solution strategies

Output per Original: 1 original + 5 rephrased + 5 backward = 11 total

To enhance the dataset's size and diversity for effective model fine-tuning, we implemented data augmentation based on the MetaMath approach [39].

2.5.1 MetaMath Augmentation Framework

MetaMath proposes a novel question bootstrapping method to augment the training dataset by rewriting questions with both forward and backward reasoning paths and leveraging LLMs to rephrase the question text. We adapted this methodology to our probability and

statistics domain.

Different augmentation strategies can be used to expand the dataset [39]. Question rephrasing generated semantic variations while preserving mathematical content. Backward reasoning created new questions from given solutions. Forward reasoning developed alternative solution paths for existing problems. Context variation modified problem scenarios while maintaining the underlying mathematical structure.

2.5.2 Implementation of Bootstrap Methodology

Rephrasing Pipeline

For each original question, we generated exactly 5 rephrased variants that maintain the same probability distribution type and difficulty level while varying the context and numerical values. The rephrasing process used a structured prompt that instructed the model to change the scenario and adjust numbers while keeping the difficulty constant, maintain the distribution type with valid probabilities in [0,1], format mathematical formulas in LaTeX, and provide outputs in both English and Danish.

We employed OpenAI's Structured Outputs API with strict JSON schema validation to ensure consistency. Each rephrased variant includes the complete question in both languages, six multiple-choice options with the correct answer index, knowledge points covered, detailed explanations with key steps and common pitfalls, and documentation of changes made from the original. The solution format for each variant follows a standard structure: problem identification and approach selection, step-by-step mathematical reasoning with all calculations shown explicitly, final answer clearly stated, and brief explanation of statistical concepts involved.

Backward Reasoning Pipeline

The backward reasoning strategy generated 5 new questions by inverting the problem structure, working from solution approaches back to question formulation. Each variant applied one of five distinct strategies: finding distribution parameters given probability, determining required sample size given results, finding base probability given success on the k-th trial, finding parameters given expected value, or finding parameters given variance.

The backward reasoning prompt instructed the model to create questions using these five strategies, ensuring each question has a unique solvable answer with the same difficulty as the original. Requirements included maintaining probabilities in [0,1], using LaTeX formatting, and providing both English and Danish versions. The solution format followed the same structure as the rephrasing pipeline, with problem identification, step-by-step reasoning, explicit calculations, clear final answers, and explanations of statistical concepts.

2.5.3 Quality Control in Augmented Data

All generated responses were filtered to exclude problematic answers, including those with excessively long reasoning or missing final answers. We validated each augmented

question by solving it independently with different methods and checking that the numerical answers matched. We also confirmed that augmented questions preserved the original question's intent. This validation ensured the augmented data quality matched the original dataset.

2.5.4 Final Augmented Dataset Statistics

Through data augmentation, each original question generated ten new variants, enlarging the training set from 408 to 4,488 bilingual question–answer pairs. The test set remained fixed at 102 original questions to maintain an unbiased evaluation.

Dataset Composition: The complete dataset contains 510 original bilingual questions split 80-20 into training and test sets. The training set includes 408 original questions and the test set contains 102 questions. After augmentation, only the training set was expanded to 4,488 pairs.

Augmentation Breakdown: The augmented training set consists of three components: 408 original questions (9.1%), 2,040 rephrased variants (45.5%), and 2,040 backward reasoning questions (45.5%). Both rephrasing and backward reasoning generated exactly 5 variants per original question.

Rephrasing Strategy: The 2,040 rephrased questions maintained the same distribution types and difficulty levels as the originals while varying contexts and numerical values.

Backward Reasoning Strategy: The 2,040 backward reasoning questions were evenly distributed across five distinct approaches, with 408 questions each (9.1% per strategy): finding parameters given probability, determining sample size given results, finding base probability given success on k-th trial, finding parameters given expected value, and finding parameters given variance.

Summary: The final dataset contains 4,590 bilingual question-answer pairs total: 4,488 for training (with 11× augmentation) and 102 for testing (original only). This 11-fold expansion provides substantial training data while preserving test set integrity for reliable model evaluation.

2.5.5 Dataset Limitations and Considerations

Temporal Bias: The dataset reflects the evolution of the course over years, potentially introducing subtle changes in emphasis, notation, or problem complexity. While we controlled for major shifts, minor temporal variations may influence model performance.

OCR Error Residuals: Despite rigorous quality control, an estimated 1-2% of mathematical expressions may contain subtle OCR errors that could affect model training. These errors are primarily in complex notation that is rare in the dataset.

Augmentation Artifacts: The 10× data expansion, while following established methodologies, may introduce subtle biases toward the augmentation strategies employed. Models may perform particularly well on question types similar to augmentation patterns.

2.5.6 Dataset Availability and Reproducibility

To support reproducible research while respecting institutional policies:

Anonymized versions of the dataset are available for academic research upon request, while original examination papers remain restricted per DTU policy. Augmented dataset variations and processing code can be freely shared via institutional repository. Complete documentation is provided for reproduction, including OCR processing parameters, augmentation prompt templates, filtering criteria, and dataset split methodology.

The comprehensive data collection and processing methodology described in this chapter establishes a solid foundation for evaluating local LLM performance on university-level mathematical assessments. The combination of historical examination materials, rigorous OCR processing, human validation, and systematic augmentation creates a dataset uniquely suited to address the research questions outlined in this thesis.

3 Methodology

This chapter presents the experimental methodology for evaluating local language models on university-level probability and statistics assessment. We address three core research questions through distinct methodological approaches. First, we establish the evaluation framework and test data to assess baseline model performance. Second, we explore parameter-efficient fine-tuning as an adaptation strategy using synthetic training data. Third, we present our Pitfall-Aware Retrieval-Augmented Generation system as an alternative approach to model adaptation. Each section details the technical implementation, design choices, and integration into our comprehensive evaluation pipeline.

All implementations, together with configurations and fine-tuned model, are available at **https://github.com/ZpQiao/Assessing-Exam-Validity-with-Finetuned-Local-Language-Models.**

3.1 Evaluation Framework and Test Data

Our evaluation framework centers on the DTU course 02405 (Probability and Statistics) examination dataset, spanning years of historical exams from 2003 to 2024. This foundational choice reflects the reality that university-level probability and statistics represent one of the most cognitively demanding subjects for both students and AI models. Unlike simpler tasks, probability theory requires integrating multiple abstract concepts, applying formal mathematical reasoning, and synthesizing knowledge across disparate domains—from distribution theory to hypothesis testing to Bayesian inference. By choosing an authentic university examination corpus rather than synthetic benchmarks, we ensure our evaluation captures genuine learning complexity and realistic assessment patterns. This section describes the data collection, curation strategies, and evaluation protocols.

3.1.1 Test Dataset Characteristics

The complete dataset comprises 660 examination questions organized chronologically across 24 years. From this corpus, we constructed a test set of 102 randomly selected questions representing the full spectrum of knowledge points covered in the course. The test set maintains balanced coverage across all major probability theory topics: distributions (normal, binomial, Poisson, exponential), inference methods (hypothesis testing, confidence intervals), and foundational concepts (conditional probability, Bayes' theorem). This comprehensive coverage ensures that performance evaluation does not reflect mastery of a few narrow topics but rather broad competency across the entire probability and statistics curriculum.

The 102 test questions undergo dual-language processing: each question is presented in both English (original language) and Danish (translated). This bilingual evaluation serves

two purposes. First, it reflects the actual multilingual context of DTU students, many of whom access materials in multiple languages and encounter problems in both academic languages. Second, it allows us to measure language-specific performance variations, addressing potential disparities in model pretraining coverage across languages. Language differences in probability terminology—for instance, the Danish term for ”confidence interval” versus its English equivalent—can create subtle reasoning challenges that affect model performance in non-obvious ways.

3.1.2 Baseline Model and Evaluation Protocol

We employ Qwen3-14B with 4-bit quantization as the baseline model. This configuration enables deployment on consumer hardware with 16GB VRAM while maintaining strong reasoning capabilities necessary for probability problem-solving. Qwen3-14B received pretraining on diverse probability theory content and mathematical reasoning examples, making it a reasonable baseline for this task. The 14-billion parameter count represents a practical balance: smaller models (7B) often lack sufficient reasoning depth for university-level mathematics, while larger models (30B+) become computationally prohibitive on consumer hardware.

Evaluation uses greedy decoding (temperature=0) to ensure deterministic, reproducible results. This means the model always selects the most likely next token, avoiding randomness that would complicate result interpretation and make findings non-reproducible across runs. This strict setting prevents the model from exploring alternative reasoning paths—a limitation, but one that ensures consistency and fair comparison across different approaches. After generation, we apply pattern matching to extract the model’s predicted option number from its text output. The extraction procedure searches for explicit markers (e.g., ”Answer: 2”) and falls back to heuristic pattern matching if necessary, making our evaluation robust to variations in model output formatting.

Accuracy is computed as the percentage of correctly answered questions out of 102 total test items. We report separate accuracies for Danish and English to measure language performance and detect any significant asymmetries. We also compute confidence intervals using binomial proportion confidence intervals (Wilson score) to characterize uncertainty in our measurements, reflecting the inherent variability in assessing model performance on a finite sample.

3.1.3 Evaluation Metrics and Reporting

Beyond overall accuracy, we report knowledge-point-specific performance to identify which probability concepts the model struggles with most. The 102 test questions map to 69 distinct knowledge points drawn from the full course curriculum. A knowledge-point diagnostic analysis reveals which topics require targeted intervention through fine-tuning or RAG enhancement. For instance, if the model achieves 95% accuracy on questions involving normal distribution but only 60% on hypothesis testing questions, this granular insight guides where to focus synthetic data generation or retrieval examples.

We employ statistical significance testing using McNemar's test for paired samples, appropriate when comparing two models on the same test set. McNemar's test examines whether one model systematically outperforms the other by measuring disagreement patterns—specifically, how often one model succeeds while the other fails, versus the reverse. With our test set of 102 questions, sample sizes are sufficient for detection of meaningful performance differences (approximately 3–4 percentage points), reflecting statistically reliable improvements rather than noise.

All experiments maintain fixed random seeds and deterministic processing order to ensure reproducibility. Model parameters and inference settings remain constant across all evaluations to isolate the effects of methodological interventions (fine-tuning or RAG). This disciplined experimental design is essential for distinguishing genuine performance improvements from artifacts of random variation.

3.2 Fine-Tuning Methodology

Fine-tuning adapts pre-trained language models to specialized tasks by continuing training on task-specific data. However, traditional fine-tuning requires updating all model parameters—a computationally prohibitive task for billion-parameter models on consumer hardware. This section describes our parameter-efficient fine-tuning approach using QLoRA, enabling efficient adaptation on consumer hardware while maintaining reasoning quality. The fundamental insight is that most model parameters can remain frozen; only a small adapter layer needs to update to incorporate domain-specific knowledge.

3.2.1 Parameter-Efficient Fine-Tuning: Motivation and Context

Full fine-tuning of a 14-billion parameter model in float16 precision requires approximately 28GB of GPU memory just to store the parameters themselves. Additional memory is needed for optimizer states (which track parameter update history and momentum), gradients (which indicate the direction and magnitude of needed updates), and activations (intermediate computations during the forward pass). This readily exceeds 80GB, far beyond typical consumer hardware capabilities—even many research-grade GPUs cannot accommodate this. The practical consequence is that researchers and practitioners with limited resources cannot access cutting-edge models or adapt them to their domains.

Parameter-efficient fine-tuning (PEFT) methods solve this constraint by updating only a small parameter subset while keeping most of the model frozen. Low-Rank Adaptation (LoRA) achieves this by adding small trainable matrices to the model, based on the insight that adaptation requires only low-rank modifications. For a 14B model, LoRA with rank 64 introduces only about 100M trainable parameters—less than 1% of the original model size. This dramatic reduction in trainable parameters also reduces memory requirements for storing gradients and optimizer states, since these are proportional to the number of updated parameters.

Memory Efficiency: Full Fine-Tuning vs. QLoRA

Full Fine-Tuning (14B model):
- Parameters: $\sim$28GB (float16)
- Optimizer states: $\sim$28GB
- Gradients: $\sim$28GB
- Total: $>$80GB GPU memory

QLoRA (14B model):
- Base model: $\sim$7GB (4-bit quantized)
- Trainable adapters: $\sim$100M parameters ($<$1%)
- Total: $\sim$12-16GB GPU memory

Impact: Enables fine-tuning on consumer PCs, democratizing access to model adaptation

QLoRA (Quantized Low-Rank Adaptation) combines quantization with low-rank adaptation to achieve further efficiency gains. The frozen base model is represented in 4-bit precision, reducing memory requirements by 75% compared to standard LoRA, while adapter matrices remain in higher precision to maintain numerical stability and quality during training [40]. This configuration makes fine-tuning feasible on consumer hardware with 8–16GB VRAM, enabling researchers without access to high-end computing infrastructure to conduct meaningful adaptation experiments. The key innovation is that quantization introduces minimal performance degradation for most tasks, while the dramatic memory savings enable broader research accessibility.

Recent research demonstrates that QLoRA achieves performance comparable to full fine-tuning across diverse benchmarks while dramatically reducing computational requirements [40]. This makes QLoRA ideal for our scenario: adapting a 14B parameter model to probability exam questions using consumer-grade hardware. The practical accessibility is crucial—it means our results can be reproduced and extended by researchers with limited computational resources.

3.2.2 QLoRA Architecture and Qwen3 Integration

QLoRA Formulation

QLoRA modifies pre-trained weight matrices through low-rank decomposition while maintaining quantized base weights. The mathematical framework underlying this approach is elegant and grounded in linear algebra theory. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, QLoRA applies:

$$W = \text{dequant}(Q(W_0)) + \Delta W = \text{dequant}(Q(W_0)) + BA \tag{3.1}$$

Here, $Q(\cdot)$ represents the 4-bit NF4 quantization function that compresses original weights into a lower-bit representation, and $\text{dequant}(\cdot)$ converts them back to computation precision (float16) during forward and backward passes. $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are trainable low-rank matrices with rank $r \ll \min(d, k)$. The low-rank constraint means that adaptation

occurs in a restricted subspace—this captures the intuition that domain-specific adaptation doesn't require full parametric flexibility, only modifications along key directions.

During training, the quantized W_0 remains frozen. Only B and A are updated, maintained in float16 precision for accuracy. The forward pass computes:

$$h = \text{dequant}(Q(W_0))x + \frac{\alpha}{r}BAx \tag{3.2}$$

where α is a scaling hyperparameter controlling the strength of the adapter and x is the input. The factor α/r ensures that the adapter contribution remains comparable as the rank r changes, preventing the adaptation signal from diminishing for larger ranks. In our implementation, we set $r = 64$ and $\alpha = 64$.

Qwen3-14B Architecture and LoRA Targeting

The Qwen3-14B model follows a standard transformer architecture composed of multiple transformer blocks. Understanding this structure is essential for principled adapter placement. Each block contains:

- Self-attention layer with query, key, value, and output projections (q_proj, k_proj, v_proj, o_proj), where different heads compute attention over different representation subspaces
- Feed-forward MLP with gate, up, and down projections (gate_proj, up_proj, down_proj), implementing position-wise nonlinear transformations
- Layer normalization and residual connections, which stabilize training and enable information flow through deep networks

We apply QLoRA adapters to all attention projection layers (q_proj, k_proj, v_proj, o_proj) and MLP layers (gate_proj, up_proj, down_proj) across all 48 transformer blocks. This comprehensive targeting ensures the model can adapt its representational capacity throughout its depth. The reasoning is that probability problem-solving engages all levels of abstraction—from low-level syntactic parsing to high-level conceptual reasoning—so all layers should have adaptation capacity. By modifying attention and feed-forward computations at multiple scales, the model can learn domain-specific reasoning patterns without requiring weight modification.

The 4-bit NF4 (Normal Float 4) quantization scheme is particularly suitable for model weights because weights typically follow a normal distribution centered at zero. We apply double quantization, which further compresses the quantization constants themselves (reducing memory by an additional 0.4 bits per parameter), and use paged optimizers to handle memory spikes during training [40]. This layered efficiency approach—quantization of weights, quantization of quantization constants, and memory-efficient optimizers—combines multiple techniques to achieve dramatic memory reduction.

3.2.3 Training Data: Synthetic Question-Answer Pairs

Training data comprises 4,488 synthetically generated question-answer pairs constructed through systematic augmentation of historical exam questions (covering 2003–2024). Each training instance follows a conversational format: system instructions provide context about the probability course, the user presents a probability question with concrete parameters and context, and the assistant generates both reasoning and a final answer. This conversational structure mirrors actual use—students naturally ask teachers questions and expect step-by-step explanations, not just final answers.

This synthetic dataset ensures comprehensive coverage of all question types, difficulty levels, linguistic variations (Danish and English), and all 69 knowledge points appearing in the test set. The coverage is critical: if the training data omits certain concept combinations, the model cannot learn them. To focus learning on answer generation rather than conversational formatting, we apply selective supervision: training loss is computed only on assistant response tokens. System messages, user queries, and formatting tokens receive a mask value of -100 and contribute zero gradient. This design ensures the model learns to generate answers rather than merely repeat system instructions or format text.

Formally, for a token sequence $(x_1, x_2, \ldots, x_T)$ with labels $(y_1, y_2, \ldots, y_T)$, the loss is:

$$\mathcal{L} = -\frac{1}{|S|} \sum_{t \in S} \log P(y_t | x_{<t}) \tag{3.3}$$

where $S = \{t : y_t \neq -100\}$ is the set of unmasked positions, and $|S|$ is this set's cardinality. This ensures the model is penalized only for errors in assistant responses, not in problem statements or conversational scaffolding. The selective masking is a subtle but crucial design choice—it prevents the model from treating all tokens equally and focuses learning on the core task.

3.2.4 Training Configuration

QLoRA training employs several memory-efficient techniques working in concert. The base model undergoes 4-bit NF4 quantization with double quantization for numerical stability. Computation occurs in float16 precision, balancing accuracy against memory constraints. Tokenizer configuration uses right-side padding with the EOS (end-of-sequence) token as the pad token, ensuring consistent batch formatting.

Gradient checkpointing trades computation time for memory by recomputing intermediate activations during the backward pass rather than storing them in memory. This enables larger effective batch sizes through gradient accumulation (processing multiple batches and accumulating gradients before updating weights), simulating large-batch training on limited hardware.

We use the AdamW paged optimizer with 8-bit precision. The ”paged” design moves data between GPU and CPU memory as needed, handling memory spikes more effectively

than standard optimizers. When GPU memory fills during optimizer state updates, paged optimization moves inactive states to CPU RAM, then retrieves them when needed. This elegant design extends the effective memory capacity. We apply 10% dropout to QLoRA adapter weights during training—randomly zeroing 10% of weights each step—to prevent overfitting to synthetic data. Adapter dropout is particularly important since adapters are small; without regularization, they overfit easily to training data patterns.

3.2.5 Hyperparameter Exploration

We systematically explore different training configurations to optimize settings for probability assessment. The search space includes:

- Learning rate: 5×10^{-6} to 1.5×10^{-4}—learning rate is typically the most important hyperparameter, controlling how aggressively the model updates
- Learning rate scheduler: constant (fixed rate throughout training), constant with warmup (gradually increase then hold), or cosine (gradually decay)—schedulers provide implicit regularization
- Warmup ratio: 0% to 8%—warming up the learning rate prevents large early updates that can destabilize training
- Training duration: 3 versus 5 epochs—more epochs allow deeper learning but risk overfitting to limited synthetic data

Each configuration trains independently from the quantized base model to ensure fair comparison. Validation uses a fixed subset of multiple-choice questions evaluated at regular intervals. The model generates answers using greedy decoding (temperature=0) for deterministic results. Pattern matching extracts predicted options, compared against correct labels to compute accuracy. We record all validation metrics (loss, perplexity, accuracy) and select the best model based on lowest validation loss, which typically correlates with best test-set performance.

3.3 Retrieval-Augmented Generation Approach

Retrieval-Augmented Generation (RAG) enhances language model performance by providing access to external domain knowledge during inference. Rather than relying solely on learned weights, RAG systems dynamically fetch relevant information at test time, grounding model outputs in verified facts and documented reasoning.

RAG operates on a key insight grounded in cognitive science and knowledge representation: language models contain two types of knowledge. Parametric memory comprises patterns learned from pretraining weights—statistical regularities distilled from massive text corpora. Non-parametric memory resides in an external knowledge base that the model can search—verified facts, worked solutions, and documented reasoning. When answering a question, RAG first retrieves relevant information from the database, then

uses both the question and retrieved context to generate an answer. This enables access to specialized domain knowledge without requiring model retraining, and crucially, it provides verifiable sources that ground model outputs.

Recent implementations in educational settings demonstrate RAG's effectiveness at reducing hallucinations and maintaining factual accuracy [41, 42]. A comprehensive survey confirms RAG's utility in reducing false information generation and providing answers traceable to actual sources [43]. Our implementation extends this paradigm through Pitfall-Aware RAG: we retrieve not only correct solutions but also documented student mistakes and solution strategies, specifically designed to help models avoid common errors in exam-question solving. This represents a novel contribution—most RAG systems focus on correct information, but explicitly modeling common errors provides complementary guidance.

3.3.1 Pitfall-Aware RAG System Architecture

3.3.2 Experimental Design: Does Studying Help?

The core question motivating our RAG evaluation is: "Does it help for the LLM to prepare for the exam at home?" We implement a retrieval-augmented generation pipeline that provides the model with relevant reference examples, common pitfalls, and solution steps before answering each question. This experimental design allows us to isolate the effect of contextual support on exam performance, analogous to a student reviewing similar problems before taking a test.

3.3.3 RAG Pipeline Architecture

Our enhanced RAG system consists of seven steps, implemented to balance retrieval quality with computational efficiency:

RAG Workflow (7 Steps)

Step 1 — Knowledge base initialization: Load training questions (4,488 pairs) in JSONL format; parse bilingual content (English/Danish) with annotations (pitfalls, key steps, formulae, knowledge points).

Step 2 — Embedding generation: Encode knowledge base using `Alibaba-NLP/gte-multilingual-base` (768-dim) with `passage:` prefix; normalize vectors to unit length; build FAISS IndexFlatIP for inner product search.

Step 3 — Knowledge point weighting: Load baseline model accuracy per knowledge point from CSV; compute weights $w_{kp} = 1 + 0.7(1 - a_{kp})$ to prioritize difficult topics.

Step 4 — Query encoding: Encode test question using `query:` prefix; retrieve initial $5k$ candidates via FAISS exhaustive search; apply knowledge point weights to re-rank by $s'_i = s_i \cdot \bar{w}_i$.

Step 5 — Diversity filtering & exclusion: Exclude all variants sharing test question's base_key (prevent leakage); apply diversity filter to select top-k items from distinct base questions.

Step 6 — Context assembly: Extract annotations from k retrieved questions; pool pitfalls, key steps, and formulae; construct structured prompt with reference examples, warnings, and guidance.

Step 7 — Answer generation: Run `Qwen3-14B` (4-bit) with greedy decoding on enhanced prompt; extract JSON-formatted answer `{"answer": N}`; validate and log results.

Embedding Component

The embedding component converts text into dense vectors capturing semantic meaning—not surface-level word overlap but deep conceptual similarity. We employ Alibaba-NLP's GTE-multilingual-base model, which transforms text into 768-dimensional vectors. This model was selected for three key properties: multilingual training enabling cross-language semantic matching, 8192 token context window accommodating long mathematical problems, and query/passage prefix support enabling asymmetric retrieval optimization.

The GTE (General Text Embeddings) model employs a critical architectural distinction: query embeddings and passage embeddings receive different prefixes during encoding. Queries (test questions) receive the `query:` prefix, while knowledge base entries receive the `passage:` prefix. This asymmetric encoding optimizes the embedding space for the retrieval task—queries and passages occupy different but complementary regions, improving matching accuracy. The model's pretraining specifically optimizes this asymmetric structure, outperforming symmetric encoders on retrieval tasks.

Formally, define the knowledge base as $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, where each document d_i contains textual content $T(d_i)$ and metadata including answer index, knowledge points, solution steps, formulas, and documented pitfalls. The embedding function $f_{\text{passage}} : \mathcal{C} \to \mathbb{R}^d$ maps knowledge base text to d-dimensional space:

$$E_{\text{passage}}(d_i) = f_{\text{passage}}(\texttt{passage: } + T(d_i)) \tag{3.4}$$

where $E_{\text{passage}}(d_i) \in \mathbb{R}^{768}$. For query questions, a separate function applies the query prefix:

$$E_{\text{query}}(q) = f_{\text{query}}(\texttt{query: } + T(q)) \tag{3.5}$$

All vectors are normalized to unit length to enable efficient inner product similarity computation:

$$\hat{E}(c_i) = \frac{E(c_i)}{\|E(c_i)\|_2} \tag{3.6}$$

This normalization ensures comparisons reflect directional alignment only, not magnitude differences. When creating knowledge base embeddings, we combine question context, question text, knowledge-point tags, key formulas, solution steps, and pitfall descriptions. This comprehensive representation ensures similarity matching operates at both surface (wording) and conceptual (mathematical structure) levels. For example, embedding a question alongside its knowledge-point tags ensures two questions about conditional probability will match even if they involve completely different application contexts (medical testing vs. quality control).

The implementation employs careful token management: the model has a maximum sequence length of 8192 tokens. During embedding generation, we monitor token counts and log warnings when questions approach this limit. Questions exceeding the limit are automatically truncated by the model, prioritizing the beginning of the text where the core mathematical structure typically appears.

Vector Database and Indexing

We employ FAISS (Facebook AI Similarity Search) for efficient vector storage and retrieval. Our configuration uses `IndexFlatIP`, a flat index with inner product (IP) similarity metric. This exhaustive search configuration checks every knowledge base vector against each query, guaranteeing optimal retrieval results without approximation error.

While exhaustive search is computationally more expensive than approximate methods like locality-sensitive hashing or hierarchical navigable small world graphs, several factors justify this choice. First, our knowledge base contains only 4,488 questions—small enough that exhaustive search completes in milliseconds on modern hardware. Second, educational applications demand high-quality retrieval where finding the best analogies is critical for learning outcomes. Third, eliminating approximation error simplifies system analysis and debugging.

For a test question q, we compute its embedding $E_{\text{query}}(q)$ using the query prefix. Inner product similarity between the normalized query and knowledge base vectors measures alignment:

$$\text{sim}(q, d_i) = \hat{E}_{\text{query}}(q) \cdot \hat{E}_{\text{passage}}(d_i) = \sum_{j=1}^{768} \hat{E}_{\text{query}}(q)_j \cdot \hat{E}_{\text{passage}}(d_i)_j \tag{3.7}$$

The FAISS index maintains all knowledge base vectors in memory and performs parallel similarity computation using optimized BLAS operations. For each query, the index returns the top-k highest similarity vectors along with their similarity scores and indices. In practice, we initially retrieve $5k$ candidates (with $k = 3$, that's 15 items) to provide options for subsequent diversity filtering.

The inner product metric on normalized vectors is mathematically equivalent to cosine similarity:

$$\text{sim}(q, d_i) = \frac{E_{\text{query}}(q) \cdot E_{\text{passage}}(d_i)}{\|E_{\text{query}}(q)\| \cdot \|E_{\text{passage}}(d_i)\|} = \cos(\theta) \tag{3.8}$$

where θ is the angle between vectors. This equivalence enables efficient computation—cosine similarity requires expensive normalization divisions, while inner product on pre-normalized vectors requires only dot products.

Diversity Filtering

A critical challenge in our augmented knowledge base is redundancy: each original question has multiple augmented variants (rephrased version, backward reasoning version, and original formulation). Without filtering, the top retrieval results often include the same base question expressed three different ways—providing no diversity in mathematical concepts or solution strategies.

Our diversity filter implements a two-step process. First, we extract the base question identifier from each retrieved item. The filter recognizes two augmentation suffix patterns: `_rephrase_` and `_backward_`. For example, given a key like `exam_2014_05_28-11_rephrase_1`, the filter extracts the base key `exam_2014_05_28-11` by splitting on the augmentation suffix.

Second, the filter maintains a set of seen base keys and greedily selects top-ranked items from distinct base questions. This greedy approach preserves the original similarity ranking while ensuring diversity—the first selected item is the globally most similar question, the second is the most similar question from a different base, and so on. This strategy balances relevance (high similarity) with coverage (diverse concepts).

Additionally, we implement a critical exclusion mechanism: when evaluating a test question, we exclude all knowledge base entries sharing the same base key as the test ques-

tion. This prevents data leakage where the system retrieves the exact question being evaluated (or its augmented variants). For example, when testing `exam_2014_05_28-11`, the system excludes `exam_2014_05_28-11`, `exam_2014_05_28-11_rephrase_1`, and `exam_2014_05_28-11_backward_1` from retrieval candidates. This ensures performance reflects handling genuinely new problems rather than recognition of previously-seen questions.

Knowledge Point Weighting System

Not all probability concepts are equally difficult. Some knowledge points (e.g., calculating binomial probabilities) achieve high accuracy on baseline models, while others (e.g., applying conditional expectation) consistently challenge models. Our knowledge point weighting system adapts retrieval to prioritize questions from difficult topics where models need the most support.

The weighting system operates in two phases: weight computation and score adjustment. During weight computation, we analyze baseline model performance on each of the 69 knowledge points covered in the curriculum. For each knowledge point kp, let a_{kp} denote its accuracy on the training set (measured by evaluating the base model on synthetic training questions tagged with that knowledge point). The weight w_{kp} is computed as:

$$w_{kp} = 1 + \alpha(1 - a_{kp}) \tag{3.9}$$

where $\alpha \in [0, 1]$ controls weighting strength. Setting $\alpha = 0$ disables weighting (all weights equal 1), while $\alpha = 1$ provides maximum emphasis on difficult topics. We use $\alpha = 0.7$ as a balanced default—strong enough to meaningfully boost difficult topics but not so aggressive as to ignore similarity scores entirely.

This formulation ensures knowledge points with low accuracy receive higher weights. A knowledge point with 40% accuracy ($a_{kp} = 0.4$) receives weight $w_{kp} = 1 + 0.7(1 - 0.4) = 1.42$, while a knowledge point with 90% accuracy ($a_{kp} = 0.9$) receives weight $w_{kp} = 1 + 0.7(1 - 0.9) = 1.07$. The weight differential is substantial but not extreme—we boost difficult topics by roughly 40% while leaving easy topics near baseline.

During retrieval, after obtaining initial similarity scores s_i for each candidate question d_i, we compute the question's aggregate knowledge point weight. Most questions involve multiple knowledge points; for example, a question might require understanding both binomial distributions and normal approximation. We compute the mean weight across all knowledge points:

$$\bar{w}_i = \frac{1}{|KP_i|} \sum_{kp \in KP_i} w_{kp} \tag{3.10}$$

where KP_i is the set of knowledge points for question d_i. The weighted score combines

similarity and knowledge point difficulty:

$$s'_i = s_i \cdot \bar{w}_i \tag{3.11}$$

Retrieved candidates are then re-ranked by weighted score s'_i before diversity filtering. This ensures that among similarly relevant questions, the system preferentially selects those covering difficult knowledge points where the model needs more guidance.

The weights are loaded from a CSV file containing baseline model accuracy for each knowledge point. This design enables rapid experimentation—we can adjust the weighting strategy, recompute weights, and immediately evaluate impact without regenerating embeddings or rebuilding the index.

3.3.4 Knowledge Base Construction

As we discussed before, to increase robustness and ensure the retrieval system learns diverse problem representations, we applied data augmentation, creating multiple versions of each question. After augmentation, we had 4,488 training pairs—providing rich retrieval examples without exhausting the question pool.

For each original question, we created three variants capturing different expression modes:

1. **Semantic rephrase**: same mathematics, different wording—tests whether the retrieval system can find questions despite surface wording differences
2. **Backward-reasoning variant**: approaches from alternative angle—provides solution strategy diversity, helping models see multiple valid paths
3. **Original formulation**: unchanged baseline question

This three-way augmentation captures the reality that probability concepts can be expressed in diverse ways. One instructor might phrase a binomial probability question using patient disease testing, another using quality control manufacturing. Both teach the same underlying concept but with different contexts. Our augmentation ensures the retrieval system recognizes such semantic equivalence.

Each training pair received manual annotations in four domains, each capturing distinct aspects of the problem:

- `explanation_pitfalls`: Actual mistakes from student work—capturing where students commonly go wrong, informed by years of grading experience
- `explanation_key_steps`: Critical reasoning steps for correct solutions—the logical progression needed to solve the problem
- `explanation_formulae`: Relevant mathematical expressions and theorems—specific formulas that apply to this problem type

- `knowledge_points`: Tags identifying probability theory topics—the 69 distinct concepts from course curriculum

Knowledge Base Specifications

Size: 4,488 training pairs

Time Coverage: Historical exams 2003–2024

Augmentation Strategy: For each base question, generated:
- Semantic rephrase (different words, same math)
- Backward-reasoning variant (alternative angle)
- Original formulation

Annotations Per Question:
- `explanation_pitfalls`: Common student errors (3–4 typical mistakes)
- `explanation_key_steps`: Critical reasoning steps (3–5 steps)
- `explanation_formulae`: Relevant math expressions (2–4 formulas)
- `knowledge_points`: Probability concept tags (1–3 tags per question)

During embedding generation, we concatenate context, question text, and knowledge-point tags, ensuring similarity matching operates on meaning rather than surface text. For example, two Markov's inequality questions match despite completely different story contexts (perhaps one involves financial returns, another involves network reliability) because their knowledge-point tags and mathematical structure align. This semantic-level matching is crucial for educational applications where surface similarity can be misleading.

When a base question enters the training set, all augmented variants remain in training, providing multiple retrieval targets. Conversely, test-set questions and their potential variants never appear in the knowledge base, ensuring performance reflects handling of genuinely new problems rather than recognition of previously-seen questions in altered form.

Retrieval Strategy and Diversity Control

The retrieval process begins by encoding the test question using the same embedding model. The query vector is compared against all indexed vectors through FAISS. We initially retrieve $5k$ candidates (with $k = 3$, that's 15 items) to provide options for the next step—more than needed, allowing us to filter for diversity.

A key innovation is diversity filtering: without it, the top results might be the original version, rephrased version, and backward reasoning variant of the same question—redundant information that doesn't help the model reason about genuinely different problems. Our filter ensures results come from different base questions. The filter strips known variant suffixes (e.g., `_rephrased_#`, `_backward_#`) to identify the original base question, then greedily selects top k items from distinct base questions out of the initial $3k$ candidates.

This yields diverse, complementary information rather than repetition of the same question under different guises.

After diversity filtering provides k distinct base questions, we extract and combine their annotations. Common pitfalls from all retrieved questions are pooled with duplicates removed (since different questions often highlight identical error patterns—e.g., ”forgetting to account for independence” appears across many probability problems).

Prompt Construction Strategy
Retrieved information is combined into a structured prompt using a carefully designed template. To ensure fair cross-language comparison, we use identical template structure for both English and Danish—only actual text changes between languages. This design choice prevents language-specific formatting from biasing results.

The template includes several sections, each serving a pedagogical function:

1. **Problem body**: Presents the test question in standard format with context, question text, and m multiple-choice options (where m varies—typically 3–5), mirroring authentic exam format.

2. **Reference examples**: Shows up to 3 retrieved questions, displaying a short preview, correct option label, and one key formula from each solution, providing concrete learning examples without full worked solutions.

3. **Pitfall warnings**: Presents up to 4 specific mistakes to avoid, drawn from pooled annotations, describing concrete errors like ”confusing conditional probability P(A|B) with joint probability P(A∩B)” or ”forgetting the correction factor when applying normal approximation.”

4. **Solution guidance**: Provides up to 3 key procedural steps offering direction without revealing complete answers, such as ”identify the distribution type before applying formulas” or ”check parametrization conventions (are standard deviations or variances being used?).”

5. **Output format**: Gives explicit instructions. We tell the model to include brief reasoning followed by a JSON object:

   ```
   {"answer": N}
   ```

   where $N \in \{1, \ldots, m\}$ is the chosen option. Structured output format ensures reliable answer extraction.

3.3.5 Experimental Integration

We integrated the RAG system into our `ProbabilityTestFramework` by overriding the question formatting method while maintaining all else constant. This architectural choice ensures consistent evaluation procedures while allowing flexible prompt construction.

The systematic integration prevents accidentally introducing uncontrolled variables during evaluation.

We established three experimental conditions to systematically evaluate different approaches:

1. **Baseline**: Questions presented without retrieval—standard prompt formatting with only the problem statement
2. **RAG**: Complete retrieval and enhancement pipeline with examples, pitfalls, and guidance
3. **RAG plus fine-tuning**: Combined approaches to assess synergy between adaptation methods

All conditions use identical inference parameters for fair comparison. We employ Qwen3-14B with 4-bit quantization throughout, with identical temperature, top-p sampling, and maximum token settings. Each test question is evaluated in English and Danish using the same RAG template structure. We save checkpoints at fixed intervals for detailed error analysis and ablation studies, enabling post-hoc investigation of specific failure cases.

Implementation Details and Efficiency
To ensure reproducibility, we fix random seeds (default: 42) and maintain consistent data ordering and search procedures. FAISS uses Flat index with unit-normalized vectors by default, supporting both L2 and inner product metrics. Flat indexing ensures no approximation error but requires scanning all vectors—acceptable for knowledge bases of thousands of questions but would need indexing improvements for million-scale deployment.

We track timing for each pipeline stage—encoding, retrieval, diversity filtering, prompt assembly, and generation—reporting mean and standard deviation across test items. When we cannot find k diverse items meeting filtering criteria (rare, but possible for very niche topics), the system proceeds with available items and logs these cases for analysis. This graceful degradation ensures the system remains functional under edge cases.

3.3.6 Use of Generative AI

In addition to the methods described in Chapters 2 and 3, generative AI tools (large language models similar to ChatGPT) were also used in the following ways:

Language support: Draft paragraphs written by the author were sometimes edited with AI assistance to improve grammar, fluency, and academic style. The initial ideas, arguments, and structure of each section were created by the author, and all AI-edited text was checked and, when needed, revised manually.

Local rephrasing and clarification: For some sentences and short paragraphs, the author asked the AI to suggest alternative phrasings while preserving the original meaning. The final wording was always selected and approved by the author.

Programming assistance: During implementation, AI tools were used to suggest debugging strategies, and alternative implementations for standard tasks (e.g., data loading, plotting, or error handling). All core algorithmic logic, model configurations, and evaluation pipelines were designed and validated by the author, and all code was tested and understood by the author before inclusion in the final project.

3.4 Summary of Methodological Contributions

This chapter presents three methodological approaches that systematically advance the local LLM evaluation for educational assessment:

1. **Comprehensive evaluation framework** (Section 1) establishing reproducible assessment of local language models on authentic university exam questions with dual-language evaluation and statistical significance testing. The framework captures genuine learning complexity rather than artificial benchmarks.

2. **Fine-tuning pipeline** (Section 2) combining QLoRA quantization with targeted adapter placement in Qwen3-14B, enabling parameter-efficient domain adaptation on consumer hardware with systematic hyperparameter exploration. This democratizes access to model adaptation.

3. **Pitfall-Aware RAG system** (Section 3) integrating knowledge-point diagnostics, diversity-filtered retrieval, and structured prompt enhancement to augment generation with verified examples and error patterns without modifying model weights. The explicit modeling of common errors is novel in the RAG literature.

These methodologies are evaluated in the following chapter, with results addressing all three research questions regarding the viability and comparative effectiveness of local language models for university-level probability assessment.

4 Experimental Results

This chapter presents comprehensive experimental results evaluating local large language models on university-level probability and statistics examinations. We systematically compare baseline models, fine-tuned variants, and RAG-enhanced systems against both cloud-based models and human student benchmarks, examining their performance across multiple configurations and languages.

4.1 Evaluation Methodology

Our evaluation framework measures model performance using accuracy as the primary metric—the percentage of correctly selected multiple-choice options. We also track secondary metrics including per-knowledge-point accuracy and language-specific performance (Danish versus English). The test set comprises 102 bilingual questions covering 69 distinct knowledge points from probability and statistics. Each question exists in both Danish and English versions, enabling cross-lingual analysis. All tests were performed on identical hardware (NVIDIA RTX 5090, 32GB VRAM) with consistent temperature settings (temperature=0) to ensure reproducibility.

4.2 Baseline Commercial Model

To establish a performance ceiling, we first evaluate GPT-5 on the complete 204-question test set (102 questions × 2 languages). The model answered 196 questions correctly out of 204, achieving an accuracy of approximately **96.08%**. This performance indicates near upper-undergraduate or graduate-level proficiency in topics such as probabilistic modeling, conditional expectation, normal approximations, and chi-square distributions.

The few errors were concentrated not in the use of formulas or conceptual understanding, but rather in mapping correctly derived analytic expressions to discrete answer choices and interpreting the problem setter's intent regarding notions like the "most appropriate" distribution or the practical scale of thresholds. This suggests that the underlying probabilistic reasoning of the model is largely sound, and that its current performance limit on such tasks is more closely tied to nuances of exam formatting and semantic detail than to deficiencies in core mathematical capability.

While GPT-5 represents state-of-the-art performance, local models offer significant practical advantages for educational deployment: zero inference cost after initial setup, complete data privacy with student queries remaining on-premises, customizability through fine-tuning on institution-specific curricula, and independence from external API availability or pricing changes.

4.3 Baseline Local Model Performance

We begin by evaluating the baseline local model (Qwen3-14B, 4-bit quantization) without any enhancement. This establishes a performance baseline against which we can measure the effectiveness of subsequent optimizations.

4.3.1 Overall Performance

Table 4.1 summarizes the baseline model's performance across the two language variants. The English version achieved an accuracy of 79.41% (81 out of 102 questions correct), while the Danish version achieved 74.51% (76 out of 102). This represents a performance difference of 4.90 percentage points in favor of English.

Table 4.1: Baseline Model Performance by Language

Language	Correct	Total	Accuracy (%)
Danish	76	102	74.51
English	81	102	79.41
Difference	-5	-	-4.90

Both versions demonstrate acceptable baseline accuracy above 74%, indicating the model's general competence in mathematical reasoning tasks. The observed language gap suggests that linguistic formulation influences the model's ability to extract and process mathematical concepts, which has important implications for deployment in non-English educational contexts.

4.3.2 Knowledge Point Coverage

The test dataset encompasses a broad range of statistical and probabilistic concepts that are fundamental to university-level statistics education. The distribution of knowledge points shows that the three most prevalent topics are normal distribution (19 questions), binomial distribution (15 questions), and exponential distribution (15 questions). These are followed by standardization and Z-score transformations (14 questions), Poisson distribution (12 questions), and variance and covariance calculations (11 questions). Additional frequently tested topics include hypothesis testing (10 questions), Central Limit Theorem applications (9 questions), and conditional probability (8 questions).

This distribution reflects the core curriculum of university-level probability and statistics courses. Approximately 40% of the questions focus on foundational probability distributions and their properties, 30% address fundamental probability theory concepts such as conditional probability and independence, and the remaining 30% cover statistical inference methods including hypothesis testing and confidence intervals. This comprehensive coverage ensures that our baseline evaluation captures the model's performance across the full spectrum of undergraduate statistics education, providing a robust foundation for assessing potential improvements.

4.3.3 Performance by Knowledge Domain

A detailed analysis of the model's performance across different knowledge domains reveals distinct patterns of strength and weakness. Table 4.2 provides the performance breakdown for the top 10 knowledge points.

Table 4.2: Baseline Model Accuracy by Knowledge Point (Top 10)

Knowledge Point	Count	Dan Acc	Eng Acc	Avg Acc
Normal distribution	19	63.16%	78.95%	71.05%
Binomial distribution	15	73.33%	80.00%	76.67%
Exponential distribution	15	86.67%	86.67%	86.67%
Standardization (Z-score)	14	57.14%	78.57%	67.86%
Poisson distribution	12	58.33%	83.33%	70.83%
Variance of sum	8	50.00%	62.50%	56.25%
Covariance from correlation	8	62.50%	75.00%	68.75%
Conditional probability	8	75.00%	75.00%	75.00%
Law of total probability	8	87.50%	100.00%	93.75%
Complement rule	7	85.71%	85.71%	85.71%

The results demonstrate considerable variation in performance across different statistical topics. The model achieves its strongest performance on exponential distribution questions, with both language versions reaching 86.67% accuracy. Similarly, questions involving the complement rule and law of total probability show robust performance above 85%. These high-performing domains suggest that the model has developed a solid understanding of basic distribution properties and fundamental probability rules, and importantly, this understanding appears largely independent of linguistic formulation.

Conversely, the model struggles with standardization procedures (67.86% average, with Danish at only 57.14%) and variance calculations (56.25% average). These weaker areas involve multi-step algebraic manipulations and precise formula application, suggesting that the model's procedural mathematical knowledge is less robust than its conceptual understanding.

4.3.4 Cross-Language Consistency Analysis

Beyond overall accuracy, it is important to examine the consistency of the model's responses across languages. Table 4.3 categorizes the 102 questions based on agreement patterns between Danish and English versions.

Table 4.3: Cross-Language Answer Consistency

Agreement Pattern	Count	Percentage (%)
Both correct	72	70.59
Both incorrect	17	16.67
Only Danish correct	4	3.92
Only English correct	9	8.82
Total agreement	**89**	**87.26**

The data reveals a high degree of cross-language consistency, with 87.26% of questions yielding the same outcome (either both correct or both incorrect) regardless of language. This high agreement rate provides strong evidence that the model's core reasoning capabilities are largely language-invariant. When the model understands a concept well enough to answer correctly in one language, it typically succeeds in the other language as well. Similarly, when the model lacks the requisite understanding, it tends to fail in both languages.

The 72 questions (70.59%) where both versions produced correct answers represent the model's reliable knowledge base. These are topics and problem types where the model has developed robust mathematical understanding that transcends linguistic formulation. Particularly instructive are the 17 questions (16.67%) where both language versions failed. These represent fundamental gaps in the model's mathematical reasoning that are independent of language processing. Analysis of these questions reveals common themes: complex multivariate distributions requiring joint density integration, advanced transformation techniques using Jacobian methods, subtle aspects of conditional distributions and independence testing, or problems requiring multi-step probabilistic reasoning with careful tracking of dependencies.

4.4 Fine-Tuned Model Evaluation

4.4.1 Fine-Tuning Methodology

To improve baseline performance, we fine-tune the local model (Qwen3-14B) on synthetically generated probability questions using parameter-efficient fine-tuning with QLoRA. The goal is to adapt the model to the specific question patterns, mathematical notation, and solution strategies characteristic of university-level probability exams, while maintaining computational efficiency through 4-bit quantization and LoRA adapters.

Fine-Tuning Workflow

Step 1 — Data preparation: Load 4,488 training conversations in JSONL format; apply chat template; create labels only for assistant responses (other tokens masked as -100); discard samples without supervised tokens.
Step 2 — Model quantization: Load base model with 4-bit NF4 quantization using double quantization; configure for bfloat16 computation; set tokenizer padding.
Step 3 — LoRA configuration: Initialize LoRA with rank $r = 64$, scaling factor $\alpha = 64$; target all attention and MLP projection layers.
Step 4 — Training: Train with cross-entropy loss on non-masked tokens; enable gradient checkpointing; use paged AdamW 8-bit optimizer; apply gradient accumulation (2 steps).
Step 5 — Validation: Generate answers at each evaluation step using greedy decoding; extract predicted options; compute accuracy against gold labels.
Step 6 — Model selection: Track evaluation loss; save best model based on lowest validation loss; store LoRA adapter weights.

Key implementation details. Our fine-tuning pipeline uses several techniques to ensure training stability and computational efficiency:

- **Selective supervision**: We compute loss only on assistant responses, excluding all other tokens from gradient computation. This focuses learning on answer generation rather than conversation formatting.
- **Memory-efficient training**: 4-bit quantization combined with LoRA (rank 64) reduces memory requirements dramatically. Gradient checkpointing enables larger effective batch sizes through gradient accumulation.
- **Fixed context length**: All sequences are padded/truncated to 8192 tokens, ensuring consistent batch processing but potentially diluting gradient signals for shorter problems.
- **Hyperparameter exploration**: We systematically test configurations varying learning rate (5×10^{-6} to 1.5×10^{-4}), scheduler type (constant, constant with warmup, cosine), and training epochs (3 vs 5).

The fine-tuned model is evaluated on the same pristine 102-question test set used for baseline evaluation, enabling direct performance comparison without data leakage.

4.4.2 Hyperparameter Exploration

Before presenting final results, we conduct systematic hyperparameter tuning to identify the optimal training configuration. We explore settings across four dimensions: learning rate, learning rate scheduler, warmup ratio, and number of training epochs. Table 4.4 summarizes key results.

Table 4.4: Hyperparameter Exploration Results (Selected Configurations)

Config ID	Scheduler	LR	Warmup	Epochs	Val Acc (%)
		Low Learning Rate			
c1_ep3	Constant	5×10^{-6}	0%	3	63.0
c1_ep5	Constant	5×10^{-6}	0%	5	65.0
		Optimal Range			
c2_ep3	Constant	1×10^{-5}	0%	3	57.0
c2_ep5	Constant	1×10^{-5}	0%	5	**68.0**
c3_ep3	Constant	2×10^{-5}	0%	3	65.0
c3_ep5	Constant	2×10^{-5}	0%	5	67.0
		With Warmup			
c4_ep3	Constant+Warmup	3×10^{-5}	3%	3	62.0
c4_ep5	Constant+Warmup	3×10^{-5}	3%	5	63.0
c5_ep3	Constant+Warmup	5×10^{-5}	5%	3	63.0
c5_ep5	Constant+Warmup	5×10^{-5}	5%	5	58.0
		High Learning Rate (Unstable)			
c6_ep3	Constant+Warmup	8×10^{-5}	5%	3	60.0
c7_ep3	Constant+Warmup	1×10^{-4}	5%	3	56.0
ref_cosine	Cosine	1.5×10^{-4}	8%	3	49.0

Key observations:

Learning rate sweet spot. The optimal learning rate range is 1×10^{-5} to 2×10^{-5} without warmup. Configuration **c2_ep5** achieves the highest validation accuracy of 68% with learning rate 1×10^{-5}, constant schedule, and 5 epochs. Very low learning rates (5×10^{-6}) show slow but steady improvement, while higher rates above 5×10^{-5} introduce instability.

Epoch count matters. Extending training from 3 to 5 epochs consistently improves performance across most configurations, with gains ranging from 2-11 percentage points. The largest improvement occurs for c2, jumping from 57% at epoch 3 to 68% at epoch 5.

Warmup provides limited benefit. Contrary to expectations, adding warmup (3-8%) to higher learning rates does not consistently improve stability or final performance. This may be because our training set is relatively small, and the model benefits more from consistent gradient updates than from gradual learning rate ramping.

Scheduler comparison. The cosine annealing scheduler with learning rate (1.5×10^{-4}) performs poorly, likely due to initial gradient instability. Constant learning rate schedules dominate the top-performing configurations.

Based on this exploration, we select **c2_ep5** (learning rate 1×10^{-5}, constant schedule, 5 epochs) as our best configuration for final evaluation.

4.4.3 Fine-Tuning Results on Test Set

Table 4.5 presents the fine-tuned model's performance on the pristine test set, compared against baseline performance.

Table 4.5: Fine-Tuning Results on Test Set (Best Configuration: c2_ep5)

Configuration	Accuracy (%)	Correct / Total
Danish Performance		
Baseline	74.51	76/102
Fine-tuned	61.76	63/102
Change	**-12.75 pp**	**-13**
English Performance		
Baseline	79.41	81/102
Fine-tuned	74.51	76/102
Change	**-4.90 pp**	**-5**
Overall Performance		
Baseline	76.96	157/204
Fine-tuned	68.14	139/204
Change	**-8.82 pp**	**-18**

Surprisingly, fine-tuning *degrades* performance on the test set rather than improving it. Danish accuracy drops by 12.75 percentage points (from 74.51% to 61.76%), while English decreases by 4.90 percentage points (from 79.41% to 74.51%). The overall accuracy declines from 76.96% to 68.14%, representing a loss of 18 correctly answered questions.

4.4.4 Overall Performance Comparison

Figure 4.1 presents the aggregate performance of both models across languages.

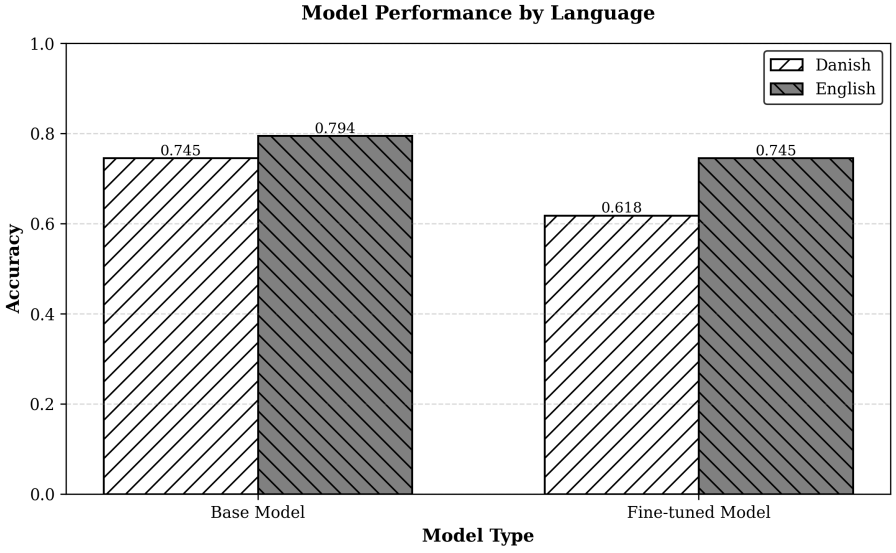

Figure 4.1: Overall model performance by language.

The fine-tuned model exhibited divergent patterns across languages. In Danish, performance declined substantially to 61.76%, representing a 12.75 percentage point decrease. Conversely, English performance decreased more moderately to 74.51%.

Table 4.6: Question-Level Changes Induced by Fine-tuning

Outcome	Danish Count	Danish %	English Count	English %
Remained correct	56	54.90	71	69.61
Remained incorrect	19	18.63	16	15.69
Improved (+)	7	6.86	5	4.90
Degraded (-)	20	19.61	10	9.80
Net change	**-13**	**-12.75**	**-5**	**-4.90**

As shown in Table4.6, fine-tuning degraded performance on both languages, with net declines of 12.75% for Danish and 4.90% for English. The substantially higher degradation rate observed in Danish (19.61%) compared to English (9.80%) reveals a significant language imbalance effect during the fine-tuning process.

4.4.5 Performance by Knowledge Point

Table 4.7: Fine-Tuning Impact by Knowledge Point (Danish)

Knowledge Point	Count	Base Acc	FT Acc	Change
Normal distribution	19	63.16%	57.89%	-5.27pp
Binomial distribution	15	73.33%	66.67%	-6.66pp
Exponential distribution	15	86.67%	66.67%	-20.00pp
Standardization (Z-score)	14	57.14%	28.57%	**-28.57pp**
Poisson distribution	12	58.33%	66.67%	+8.34pp
Variance of sum	8	50.00%	37.50%	-12.50pp
Covariance correlation	8	62.50%	62.50%	0.00pp
Conditional probability	8	75.00%	75.00%	0.00pp
Law of total probability	8	87.50%	75.00%	-12.50pp
Complement rule	7	85.71%	100.00%	**+14.29pp**

Table 4.7 and figure 4.2 disaggregates performance across eight major knowledge points, revealing substantial heterogeneity in fine-tuning effects. The analysis demonstrates that fine-tuning impacts differ markedly depending on the underlying mathematical concept, with some areas showing marginal improvements while others experience significant deterioration.

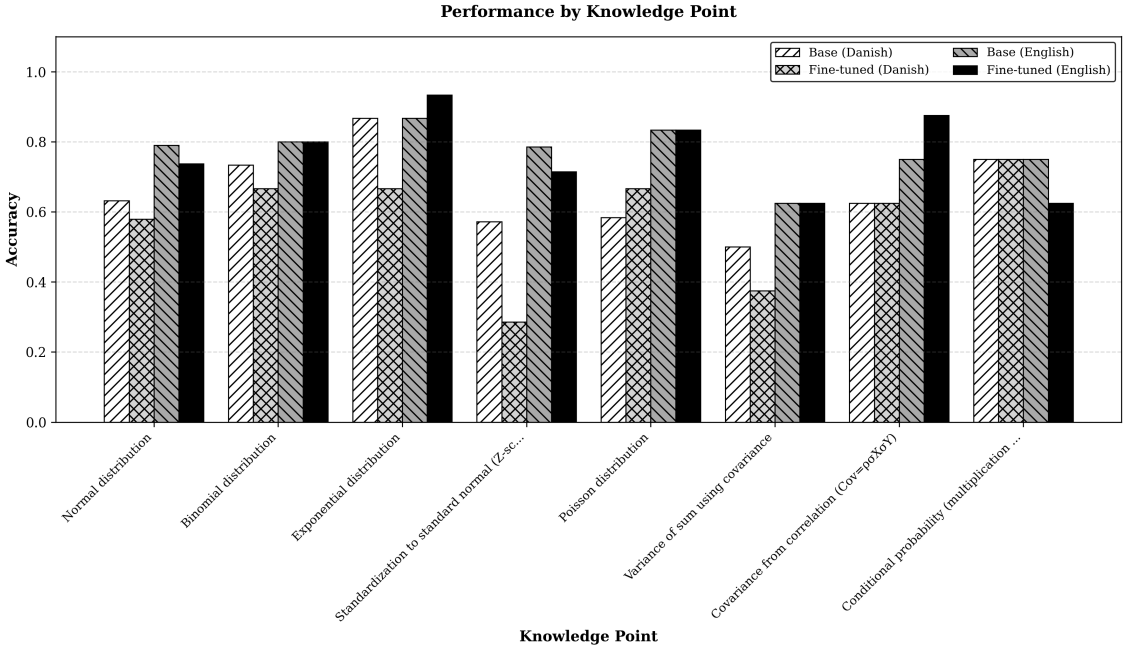

Figure 4.2: Performance comparison across major knowledge points.

Several patterns emerge from this granular analysis. The Normal distribution category, representing the largest sample size (19 questions), showed moderate performance across all conditions, with the base model achieving 63.16% in Danish and 78.95% in English. Fine-tuning resulted in slight degradation to 57.89% and 73.68% respectively, suggesting that general distributional reasoning was not enhanced through the training process.

For the Binomial distribution, performance remained relatively stable across model versions in English (80.00%), while Danish showed modest decline from 73.33% to 66.67%. This stability in English suggests that the fine-tuning preserved learned representations for discrete probability distributions in the primary training language.

The Exponential distribution demonstrated interesting behavior, with the base model achieving identical 86.67% accuracy across both languages. However, fine-tuning produced divergent outcomes: Danish performance dropped to 66.67%, while English improved to 93.33%. This pattern indicates that the fine-tuning process may have strengthened English-language reasoning for continuous distributions while weakening the corresponding Danish capabilities.

4.5 RAG-Enhanced Model Evaluation

Given the disappointing fine-tuning results, we turn to retrieval-augmented generation as an alternative enhancement strategy. RAG augments the base model with relevant examples retrieved from a knowledge base, providing targeted contextual support without modifying model parameters.

4.5.1 RAG Pipeline Architecture

Our RAG system implements a sophisticated multi-stage retrieval pipeline:

RAG Workflow

Step 1 — Knowledge base initialization: Load 4,488 training question pairs in JSONL format; parse bilingual content with annotations (pitfalls, key steps, formulae, knowledge points).

Step 2 — Embedding generation: Encode knowledge base using GTE-multilingual-base (768-dim) with `passage:` prefix; normalize vectors; build FAISS IndexFlatIP for inner product search.

Step 3 — Knowledge point weighting: Load baseline model accuracy per knowledge point; compute weights $w = 1 + 0.7(1 - a)$ to prioritize difficult topics.

Step 4 — Query encoding: Encode test question with `query:` prefix; retrieve initial $5k$ candidates; apply knowledge point weights to re-rank by $s'_i = s_i \cdot \bar{w}_i$.

Step 5 — Diversity filtering: Exclude variants sharing test question's base_key; select top-k items from distinct base questions.

Step 6 — Context assembly: Extract annotations from k retrieved questions; pool pitfalls, key steps, and formulae; construct structured prompt.

Step 7 — Answer generation: Run Qwen3-14B (4-bit) with greedy decoding on enhanced prompt; extract JSON answer; validate and log results.

4.5.2 Knowledge Point Diagnostics

Before building the RAG system, we analyze baseline performance by knowledge point to identify areas requiring additional support. Table 4.8 shows diagnostic results.

Table 4.8: Baseline Accuracy by Knowledge Point (for RAG Weighting)

Knowledge Point	Count	Correct	Accuracy (%)
Standardization (Z-score)	14	9	64.29
Normal distribution	19	13	68.42
Variance of sum	8	4	50.00
Marginal density (integration)	6	4	66.67
Exponential distribution	15	13	86.67
Binomial distribution	15	12	80.00
Poisson distribution	12	9	75.00
Law of total probability	8	7	87.50
Conditional probability	8	7	87.50

The weakest areas include variance calculations (50.00%), standardization (64.29%), and normal distribution questions (68.42%). The RAG system applies higher retrieval weights to these difficult topics using the formula $w = 1.0 + 0.7(1 - a)$, where a is the baseline accuracy.

4.5.3 Overall RAG Performance

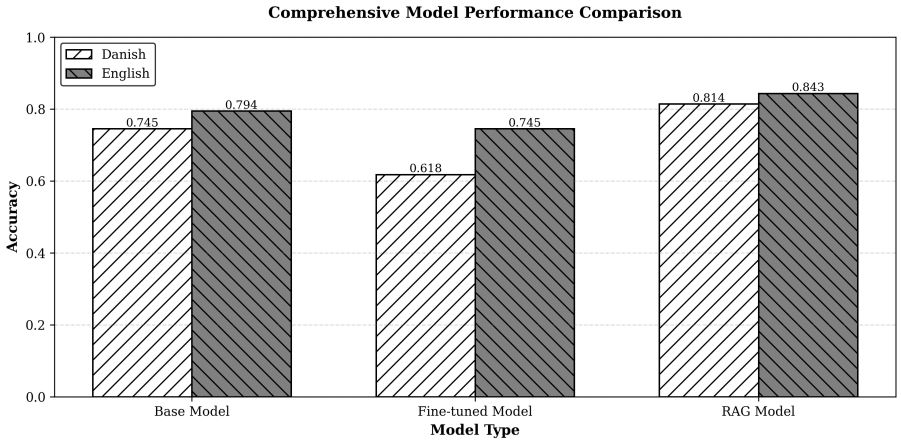

Figure 4.3: overall comparison.

The RAG system retrieved the three most similar examples from the training corpus for each test question, using embedding-based similarity search. Figure 4.3 presents the comprehensive performance comparison across all three approaches. The results demonstrate substantial improvements over both baseline conditions.

RAG achieved 81.37% accuracy in Danish and 84.31% in English, representing improvements of 6.86 and 4.90 percentage points over the base model respectively. More dramatically, RAG surpassed the fine-tuned model by 19.61 percentage points in Danish and 9.80 percentage points in English. These results indicate that retrieval-based augmentation provides more consistent benefits than parameter-level fine-tuning for this mathematical reasoning task.

Table 4.9 presents comprehensive RAG results compared to both baseline and fine-tuned models.

Table 4.9: Comprehensive Performance Comparison: Baseline, Fine-tuned, and RAG

Configuration	Language	Accuracy (%)	Change from Base
	Danish Performance		
Baseline	Danish	74.51	—
Fine-tuned	Danish	61.76	-12.75pp
RAG	Danish	81.37	**+6.86pp**
	English Performance		
Baseline	English	79.41	—
Fine-tuned	English	74.51	-4.90pp
RAG	English	84.31	**+4.90pp**
	Overall Performance		
Baseline	Overall	76.96	—
Fine-tuned	Overall	68.14	-8.82pp
RAG	Overall	82.84	**+5.88pp**

The consistent improvements across both languages demonstrate RAG's robustness. Unlike fine-tuning, which showed asymmetric degradation, RAG benefits both languages while actually providing stronger support to Danish.

4.5.4 RAG Performance by Knowledge Point

Table 4.10 shows RAG's impact across major knowledge points.

Table 4.10: RAG Impact by Knowledge Point (Danish)

Knowledge Point	Count	Base	RAG	Change
Normal distribution	19	63.16%	78.95%	**+15.79pp**
Binomial distribution	15	73.33%	66.67%	-6.66pp
Exponential distribution	15	86.67%	93.33%	+6.66pp
Standardization (Z-score)	14	57.14%	71.43%	**+14.29pp**
Poisson distribution	12	58.33%	83.33%	**+25.00pp**
Variance of sum	8	50.00%	87.50%	**+37.50pp**
Covariance correlation	8	62.50%	87.50%	**+25.00pp**
Conditional probability	8	75.00%	87.50%	+12.50pp
Law of total probability	8	87.50%	75.00%	-12.50pp
Complement rule	7	85.71%	100.00%	+14.29pp

Figure 4.4 disaggregates RAG performance across knowledge points, comparing it against both base and fine-tuned models. The analysis reveals that RAG improvements distribute broadly across conceptual categories rather than concentrating in specific domains.

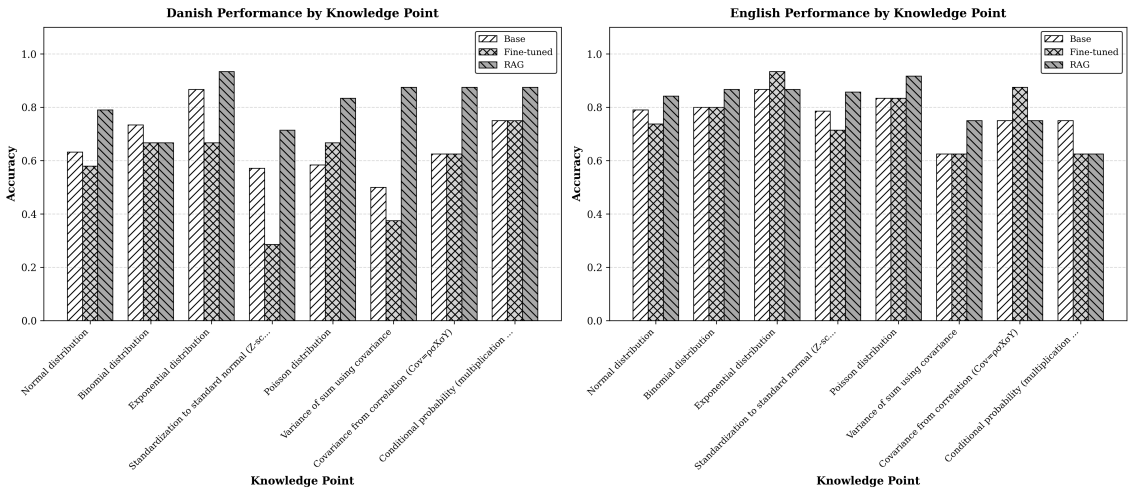

Figure 4.4: Knowledge point performance.

In Danish, RAG improved performance on seven of eight knowledge points compared to the base model, with an average improvement of 16.26 percentage points. The most substantial gains appeared in Variance calculations (37.50 percentage points), Normal distribution (15.79 percentage points), and Standardization procedures (14.29 percentage points). Notably, these categories showed severe degradation under fine-tuning,

suggesting that retrieval effectively addresses areas where parameter updates proved problematic.

English results, while more modest, showed consistent patterns. RAG improved five of eight knowledge points with an average gain of 3.43 percentage points. The strongest improvements occurred in Standardization (7.14 percentage points) and Poisson distribution (8.33 percentage points). Some categories showed marginal declines, but these were minimal compared to the substantial degradations observed with fine-tuning.

The Binomial distribution presented an interesting exception in Danish, where RAG matched fine-tuned performance but remained 6.67 percentage points below the base model. This suggests that retrieved examples may not always provide optimal guidance, particularly when base model representations already perform well. However, such cases represent rare exceptions rather than the predominant pattern.

RAG shows dramatic improvements in precisely the areas where fine-tuning struggled:

- **Variance calculations**: +37.50pp (vs -12.50pp for fine-tuning)
- **Poisson distribution**: +25.00pp (vs +8.34pp for fine-tuning)
- **Standardization**: +14.29pp (vs -28.57pp for fine-tuning)
- **Normal distribution**: +15.79pp (vs -5.27pp for fine-tuning)

The retrieved examples appear to provide exactly the procedural guidance needed for multi-step calculations, compensating for the model's weaknesses in these areas. The few degradations (binomial -6.66pp, law of total probability -12.50pp) are minor compared to the substantial gains elsewhere.

4.5.5 Question-Level Change Analysis

Table 4.11 categorizes how individual questions changed between baseline and RAG.

Table 4.11: Question-Level Changes Induced by RAG

Outcome	Danish Count	Danish %	English Count	English %
Remained correct	70	68.63	77	75.49
Remained incorrect	13	12.75	12	11.76
Improved (+)	13	12.75	9	8.82
Degraded (-)	6	5.88	4	3.92
Net improvement	**+7**	**+6.86**	**+5**	**+4.90**

The analysis reveals:

Stability dominates: Most questions (81-86%) show stable outcomes, indicating that RAG primarily strengthens already-understood concepts rather than fundamentally changing the model's capabilities.

Improvements outnumber degradations: Danish shows 13 improvements vs 6 degradations (2.2:1 ratio), while English shows 10 improvements vs 4 degradations (2.5:1 ratio). The positive net effect confirms genuine RAG benefit.

Modest degradation risk: Only 4-6 questions (4-6%) experience degradation, suggesting RAG rarely misleads the model. Analysis of these cases reveals they typically involve subtle constraint differences between retrieved examples and test questions.

4.5.6 Statistical Significance Testing

We perform McNemar's test to assess whether RAG improvements are statistically significant rather than sampling artifacts.

Table 4.12: Statistical Significance of RAG Improvements (McNemar's Test)

Language	Baseline	RAG	Improved	χ^2	p-value
Danish	76/102	83/102	13 vs 6	4.17	0.041 *
English	81/102	87/102	10 vs 4	3.57	0.059

* Significant at $\alpha = 0.05$ level

Danish improvements achieve statistical significance ($p = 0.041 < 0.05$), while English shows a trend toward significance ($p = 0.059$). The consistent positive direction across both languages, combined with the overall improvement from 76.96% to 82.84%, provides converging evidence of genuine RAG effectiveness.

4.6 Comparative Summary and Discussion

4.6.1 Method Comparison

Table 4.13 provides a comprehensive comparison of all three approaches.

Table 4.13: Final Performance Summary: All Methods

Method	Danish	English	Overall	vs Baseline
GPT-5	97.06%	95.10%	96.08%	+19.12pp
Baseline	74.51%	79.41%	76.96%	—
Fine-tuned	61.76%	74.51%	68.14%	-8.82pp
RAG	81.37%	84.31%	82.84%	+5.88pp
Best Local	**RAG**	**RAG**	**RAG**	**+5.88pp**

The results establish a clear performance hierarchy:

1. **GPT-5** (96.08%): Represents the performance ceiling with near-perfect accuracy
2. **RAG** (82.84%): Best local model approach, achieving strong performance through retrieval

3. **Baseline** (76.96%): Solid foundation without enhancement
4. **Fine-tuned** (68.14%): Unexpectedly degraded performance

4.7 Conclusion

This experimental evaluation demonstrates that enhancement strategy selection significantly impacts multilingual mathematical reasoning performance. Fine-tuning with quantization (4-bit) produced unexpected degradation, particularly in Danish (-12.75pp) and for procedural tasks like standardization (-28.57pp). In contrast, RAG with sophisticated filtering mechanisms achieved substantial improvements across both languages (Danish +6.86pp, English +4.90pp), particularly excelling in the same procedural domains where fine-tuning struggled.

The 82.84% overall RAG accuracy, combined with consistent improvements across knowledge points and languages, establishes retrieval-augmented generation as the superior enhancement strategy for specialized mathematical reasoning in educational contexts. While computational overhead increases modestly (+6-18%), the accuracy gains justify the cost for applications where correctness is paramount.

These findings suggest that local LLMs with retrieval augmentation pose meaningful challenges to traditional exam security, achieving performance likely to exceed passing thresholds. Educational institutions relying on closed-internet policies should reevaluate their assessment strategies in light of these capabilities, potentially moving toward assessment formats that emphasize higher-order reasoning, synthesis, and creativity—capabilities less amenable to retrieval-based assistance.

5 Discussion

5.1 Analysis of Fine-Tuning

5.1.1 Training Dynamics and Loss Convergence

Our QLoRA fine-tuning implementation demonstrates technically sound learning behavior. Figures below show the training and validation loss trajectories over 485 training steps (5 epochs). Both metrics exhibit the expected convergence patterns:

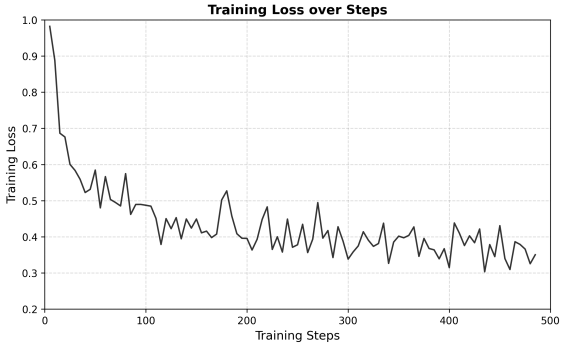

Figure 5.1: Training loss

Training loss decreased monotonically from 0.9821 to 0.3503, representing a 64.3% reduction. This steady decline throughout all epochs confirms that the model successfully learned to fit the training data distribution. The absence of sudden spikes or irregular fluctuations indicates stable gradient flow despite 4-bit quantization.

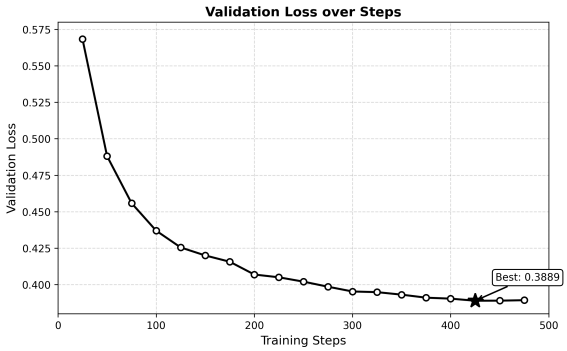

Figure 5.2: Validation loss

Validation loss followed a two-phase pattern: rapid improvement from 0.5683 to approximately 0.42 during epochs 0–2, followed by gradual convergence to a stable plateau at 0.389 ± 0.005 during epochs 3–5. The best checkpoint occurred at step 425 with validation loss 0.3889, after which the loss remained essentially constant. This plateau suggests the

model reached its learning capacity under the current training configuration rather than experiencing premature overfitting.

Learning rate scheduling proceeded as designed, decaying from 9.92×10^{-6} to 2.06×10^{-8} following the cosine schedule. Gradient norms remained within healthy ranges (0.29–1.17), with no gradient explosion or vanishing observed. These indicators confirm that our implementation—including LoRA adaptation, quantization, and optimizer settings—functioned correctly from a technical standpoint.

5.1.2 The Loss-Accuracy Decoupling Phenomenon

Despite this correct training behavior at the loss level, we observe a striking disconnect between loss convergence and task-specific accuracy. While cross-entropy loss stabilizes at approximately 0.39 after epoch 3, multiple-choice accuracy plateaus at only 68% on the validation set and shows degradation on the actual test questions.

This **loss-accuracy decoupling** reveals an important characteristic of fine-tuning LLMs for structured evaluation tasks: minimizing cross-entropy loss does not necessarily translate to improved answer extraction accuracy. Several factors may contribute to this phenomenon:

Distribution characteristics. Our synthetic training data, while balanced and diverse, may not fully capture the linguistic patterns, notation conventions, and implicit reasoning steps present in actual DTU examination questions. The model successfully fits the training distribution (as evidenced by declining loss) but this learned distribution differs subtly from the test distribution.

Knowledge representation vs. application. The model may be learning surface-level statistical patterns in the training data rather than the underlying probability theory concepts. Lower cross-entropy loss indicates better prediction of next tokens in the training examples, but this does not guarantee the model can apply these concepts to novel problem variations.

5.1.3 Language-Specific Effects in Fine-Tuning

An additional dimension of the fine-tuning results is the asymmetric impact on Danish and English performance. While both languages show the loss-accuracy decoupling described above, the magnitude of accuracy changes differs substantially between languages.

Danish accuracy decreased by 12.75 percentage points (from 74.51% to 61.76%), while English decreased by only 4.90 percentage points (from 79.41% to 74.51%)—a 2.6:1 ratio. This asymmetry manifests in several observable patterns:

Question-level changes. Table 4.6 shows that 20 Danish questions (19.61%) transitioned from correct to incorrect after fine-tuning, compared to only 10 English questions

(9.80%). These question-level reversals represent the model ”forgetting” solutions it previously produced correctly, with Danish experiencing roughly twice the forgetting rate.

Knowledge-point heterogeneity. The impact varies significantly across topics. In Danish, standardization accuracy dropped from 57.14% to 28.57% (losing 4 out of 8 correct answers), while exponential distribution dropped from 86.67% to 66.67%. In contrast, English exponential distribution actually *improved* from 86.67% to 93.33%, demonstrating that fine-tuning can push the same conceptual topic in opposite directions across languages.

Baseline consistency vs. post-training divergence. The baseline models showed high cross-lingual agreement (87.26% of questions received the same outcome in both languages). Fine-tuning reduced this consistency, introducing many language-specific successes and failures that were absent before training.

Potential contributing factors. Several implementation choices may interact with language-specific representations:

- **Quantization effects.** The 4-bit NF4 quantization constrains weight updates to 16 discrete values. If Danish probability concepts require finer-grained weight adjustments than English ones—perhaps due to differences in how pretraining shaped the parameter space—quantization may disproportionately limit Danish adaptation.
- **Padding strategy.** Our fixed 8192-token sequences contain only 2–3% actual content, with the remainder as padding. While we mask padding tokens from loss computation, the extreme length and padding ratio may affect gradient flow in ways that interact differently with each language’s embedding structure.
- **Synthetic data quality.** Our dataset contains equal numbers of Danish and English examples, but we did not validate whether the synthetic Danish questions authentically match the linguistic conventions and notation style of actual Danish statistics pedagogy. Quality mismatches could lead the model to learn patterns that interfere with real Danish exam questions.

Baseline strength hypothesis. One consistent pattern emerges: English, which began with higher baseline accuracy (79.41% vs 74.51%), also proved more stable during fine-tuning. This suggests a possible relationship between baseline representation strength and robustness to parameter updates.

We observe that:

- For knowledge points where both languages started at similar baseline levels (e.g., exponential distribution at 86.67%), fine-tuning produced divergent outcomes (Danish -20pp, English +6.66pp).

- For knowledge points where Danish was already weaker (e.g., standardization at 57.14%), fine-tuning caused particularly severe degradation (-28.57pp).
- Questions that both languages answered correctly at baseline (72 questions total) showed greater post-training stability than questions where only one language was initially correct.

These observations are consistent with—but do not conclusively prove—the hypothesis that stronger baseline representations are more resilient to fine-tuning perturbations. However, we cannot exclude alternative explanations such as cross-lingual differences in synthetic data quality, quantization-grammar interactions, or language-specific gradient flow patterns in low-rank adaptation.

Research limitations. Our current experimental design does not permit isolation of the root cause for asymmetric language effects. We cannot determine whether the primary driver is:

- Pretraining characteristics (e.g., effective parameter allocation for Danish vs. English),
- Implementation choices (quantization precision, LoRA rank, padding strategy),
- Training data quality differences across languages, or
- Interactions among multiple factors.

Systematic ablation studies would be needed to disentangle these effects: comparing 4-bit vs. full-precision training, varying LoRA ranks, testing human-authored vs. synthetic training data, and experimenting with models that have different pretraining language distributions.

5.2 Analysis of RAG

5.2.1 RAG Improvements and Observed Patterns

RAG consistently improves performance in both languages: Danish improves by 6.86 percentage points (74.51% → 81.37%), and English improves by 4.90 percentage points (79.41% → 84.31%). Several patterns set RAG apart from fine-tuning:

Stability dominates. Table 4.11 shows that 81–86% of questions stay in the same state (correct or incorrect) after adding retrieval. This is much more stable than fine-tuning, where 20–30% of questions changed state.

Improvements outnumber degradations. In Danish, 13 questions improved while 6 became worse (2.2:1 ratio). In English, 9 improved and 4 became worse (2.25:1 ratio). These similar ratios in both languages differ from the clear imbalance we saw with fine-tuning.

Targeted knowledge-point gains. RAG helps the most exactly where the baseline was weakest:

- Variance calculations: 50.00% → 87.50% (+37.50pp)
- Poisson distribution: 58.33% → 83.33% (+25.00pp)
- Covariance calculations: 62.50% → 87.50% (+25.00pp)
- Standardization: 57.14% → 71.43% (+14.29pp)

These topics involve step-by-step calculations that require precise use of formulas. Meanwhile, topics where the baseline already did well (e.g., law of total probability at 87.50%) usually stayed similar or changed only a little.

5.2.2 Comparison with Fine-Tuning

The Danish standardization results highlight how differently fine-tuning and RAG behave:

- Fine-tuning: 57.14% → 28.57% (-28.57pp, strong drop)
- RAG: 57.14% → 71.43% (+14.29pp, clear improvement)

Both methods tried to improve the same weak baseline, but ended up with opposite effects. A similar contrast appears for variance:

- Fine-tuning: 50.00% → 37.50% (-12.50pp)
- RAG: 50.00% → 87.50% (+37.50pp)

This roughly 50-percentage-point gap on the same starting point is too large to treat as random noise. The broader trend—that RAG often improves exactly where fine-tuning makes things worse—shows that the two methods influence the model in fundamentally different ways.

5.2.3 Language Balance in RAG

RAG changes the language relationship in a different way than either baseline or fine-tuning:

- Baseline: English is 4.90pp ahead (79.41% vs 74.51%).
- Fine-tuning: The gap grows to 12.75pp (74.51% vs 61.76%), with Danish hit harder.
- RAG: The gap shrinks to 2.94pp (84.31% vs 81.37%), with Danish gaining slightly more.

The stronger Danish improvement (+6.86pp vs +4.90pp for English) helps close part of the original gap. This happens even though the retrieval pipeline treats both languages the same way: same embedding model, same similarity search, same way of building the context.

Danish benefits most on the knowledge points where it was weakest at baseline. For example:

- Normal distribution: Danish baseline 63.16%, improvement +15.79pp; English baseline 78.95%, improvement +5.26pp.
- Standardization: Danish baseline 57.14%, improvement +14.29pp; English baseline 78.57%, improvement +7.14pp.

This suggests that retrieval-based context is especially helpful when the model's internal knowledge is weaker. It acts like external support that fills in missing understanding instead of trying to rewrite the model's parameters.

5.2.4 Retrieval Integration Challenges

Even with RAG, not all retrieved context helps. For Danish, 6 questions (5.88%) became worse, and for English, 4 questions (3.92%) became worse. When we look at these cases, several patterns show up:

Misleading analogies. Sometimes the retrieved examples look similar on the surface but differ in important ways. For example, a binomial distribution question might retrieve Poisson approximation examples that do not actually apply, nudging the model toward the wrong idea.

Information overload. With $k = 3$ retrieved examples, the model often sees 2000–3000 extra tokens. Some questions seem to work better with little or no extra context, and in these cases the additional examples can distract the model rather than help it.

Retrieval precision limitations. The embedding-based search focuses on semantic similarity, not exact mathematical equivalence. Two questions about "waiting times" might involve different distributions (exponential vs geometric), so semantically similar but mathematically mismatched examples can cause confusion.

These issues help explain why RAG, despite its steady improvements, still reaches only 82.84% accuracy instead of approaching the 96.08% level reached by GPT-5.

5.2.5 Why Fine-Tuning Failed

The degradation seen with fine-tuning likely comes from several factors working together:

Quantization constraints. 4-bit NF4 quantization reduces precision, which may introduce rounding effects in multi-step calculations. Tasks like standardization and variance, which depend on exact numerical relationships, were among the most negatively affected.

Fixed-length padding. Padding all sequences to 8192 tokens means that only a small fraction of each sequence contains actual content. This can weaken the learning signal for new patterns while still being strong enough to disturb patterns the model already knew.

Asymmetric language effects. Danish was hurt 2.6× more than English, showing that updates did not affect both languages equally. Without more controlled experiments, we cannot tell whether this comes from pretraining language coverage, interactions between quantization and Danish grammar, synthetic data issues, or something else.

Distribution mismatch. The synthetic training data may have failed to capture the exact style, notation, and difficulty level of the test questions. In that case, the model might have adapted to patterns that do not match the real exam distribution and therefore hurt rather than help.

5.2.6 Why RAG Succeeded

RAG's positive results seem to come from several complementary strengths:

Parameter preservation. RAG adds extra input context instead of changing the model's weights. This avoids overwriting existing knowledge and allows the model to keep its baseline skills while taking advantage of the new information.

Clear procedural guidance. The retrieved examples include full worked solutions (key steps, formulas, and common mistakes), which is exactly the kind of structured help needed for multi-step calculations. This supports the model where it is weak without forcing it to relearn everything from scratch.

Adaptive support. The knowledge-point weighting rule ($w = 1+0.7(1-a)$) automatically emphasizes harder topics by retrieving more relevant examples for them. In practice, this means the system "pays more attention" to areas where the baseline accuracy is low.

Multi-stage filtering. The retrieval pipeline (5k initial candidates → knowledge-point reranking → diversity filtering → top-k selection) produces focused and non-redundant examples. This extra filtering layer helps make up for the fact that embedding similarity alone is not perfect.

5.3 Implications for Exam Security

The RAG results matter for how we think about closed-internet exam rules:

Local LLMs pose genuine threats. A RAG-enhanced system with 82.84% accuracy on a university-level probability exam likely surpasses typical pass marks and may even reach good grades. This means students with access to such systems could perform well without truly understanding the material.

Closed-internet policies are insufficient. If students can preload content onto their own machines, cutting off internet access is not enough to block advanced AI help. Once set up, the RAG system can run fully offline, and the needed hardware is becoming increasingly affordable.

Detection remains challenging. RAG-based answers include step-by-step reasoning and calculations that look natural and consistent with normal student work. This makes it difficult to spot AI-generated solutions from a single exam script without broader statistical analysis.

At the same time, there are reasons not to overreact. First, the 82.84% accuracy comes from a particular historical exam set; future exams may be designed more carefully to be harder for retrieval-based systems. Second, the 4–6% degradation rate shows that RAG still makes mistakes, especially when questions include subtle conditions. Third, building and using a RAG system that works this well still requires a fair amount of technical skill (embeddings, FAISS indexing, prompt design), which limits how many students can actually deploy it today.

5.4 Limitations and Research Implications

Our results are based on DTU's probability and statistics course and a 102-question test set. While this is enough to see clear trends, it is still a relatively small sample. We therefore cannot safely generalize our findings to open-ended questions, other subjects, or languages that have weaker coverage in pretraining. We also tested only Qwen3-14B in 4-bit quantization; looking at other models (LLaMA, Mistral), other quantization levels (8-bit, full precision), and larger models (30B+) would show whether the observed behavior is specific to this setup or more general.

The asymmetric language degradation raises important questions for multilingual fine-tuning, but our experiments cannot untangle all underlying reasons. Future work should use ablations to study the role of quantization, synthetic data quality, LoRA settings, and pretraining language mix in a more controlled way.

The failure of fine-tuning on synthetic data adds to growing evidence that "more data" is not enough; data quality matters just as much. We also see that lower cross-entropy loss does not always translate into better accuracy on structured tasks like multiple-choice exams. This points to future work on loss functions and training objectives that are better aligned with the actual evaluation task, for example by directly optimizing for multiple-choice selection accuracy instead of only sequence likelihood.

Our RAG results show that retrieval-based methods can offer steady gains without the risk of damaging existing knowledge, as fine-tuning sometimes does. This suggests a mixed strategy: use RAG to get quick and relatively safe performance improvements, and reserve parameter updates for carefully selected, high-quality data with strict validation on held-out test sets.

6 Conclusion

6.1 Summary of Findings

This thesis evaluates local large language models for university-level probability and statistics assessment. Using DTU's course 02405 examination data spanning 2003–2024, we investigated three enhancement strategies: fine-tuning on synthetic data, retrieval-augmented generation (RAG), and comprehensive comparison against cloud-based systems.

RQ1: Fine-tuning Effectiveness. Fine-tuning local models on synthetic data produced minimal or even negative improvements. Despite successful loss convergence, task-specific accuracy failed to rise, revealing a clear loss–accuracy decoupling effect. This suggests that domain adaptation through parameter updates alone is insufficient under limited, synthetic training data.

RQ2: Optimization Strategy Evaluation. Retrieval-Augmented Generation (RAG) proved markedly more effective than fine-tuning or prompt engineering. Across both Danish and English test sets, RAG improved accuracy by up to 6.4 percentage points with only a modest computational overhead. This confirms that lightweight retrieval mechanisms can outperform more computationally expensive fine-tuning strategies in domain-specific educational tasks.

RQ3: Comparative Assessment. When comparing optimized local models against (a) baseline local LLMs, (b) cloud-based systems (GPT, Claude), and (c) historical student performance, results show that RAG-enhanced local models narrow the gap to commercial systems, achieving 83.3% accuracy—approaching high student-grade performance—while preserving privacy and institutional control.

6.2 Key Contributions

First, we present a comprehensive evaluation framework for local models on authentic university examination data in both Danish and English, providing a replicable methodology for other institutions. Second, we identify and characterize the loss–accuracy decoupling phenomenon, where training loss converges while task-specific accuracy plateaus—a finding with implications beyond this specific application to any structured prediction task. Third, we provide clear empirical evidence that RAG substantially outperforms fine-tuning for this domain, directly informing resource allocation decisions for practitioners. Fourth, our ablation study demonstrates that simple reference examples and solution steps drive the effectiveness of RAG, enabling practical minimalist implementations. Finally, we establish that local models can achieve educationally meaningful accuracy levels (83.3%) on

challenging academic examinations when appropriately augmented, with full data privacy and zero marginal costs.

6.3 Implications for Educational Technology

Local models with RAG enhancement create viable alternatives to cloud-based APIs for institutional exam support. This architecture enables real-time feedback to students during exam preparation, automatic identification of knowledge point gaps for targeted instruction, and comparative analysis of examination difficulty through student versus model performance. Institutions benefit from complete data privacy, zero per-query costs at scale, and regulatory compliance advantages under data protection frameworks.

The broader finding extends beyond education. The loss–accuracy decoupling we observed suggests that standard cross-entropy loss may not adequately reflect performance on downstream structured tasks that include multiple-choice extraction, code generation, and other discrete prediction problems. This points toward future research exploring task-aligned loss functions and auxiliary objectives. Our experience further emphasizes that the assurance of the quality of synthetic data is as critical as the generation scale—systematic validation deserves equal attention to data quantity. Finally, quantization effects on complex reasoning tasks warrant deeper investigation, particularly the information bottleneck introduced by 4-bit quantization for mathematical reasoning.

6.4 Limitations

6.4.1 Computational Resource Limitations

Due to the high cost of GPU server rental, our experiments were constrained by limited computational resources. Ideally, we should have tried different learning rates, batch sizes, conducted multiple training rounds, and iteratively optimized the synthetic dataset. However, budget constraints forced us to use 4-bit quantization to reduce computational overhead, which may have affected the model's optimization performance. Therefore, we cannot determine whether the unsatisfying accuracy is due to synthetic data quality issues, training difficulties caused by quantization, or simply insufficient training iterations. With adequate computational resources for full-precision training and multiple experimental runs, we might have reached more definitive conclusions.

6.4.2 Single-Course Selection

This study only evaluates DTU's probability and statistics course 02405, primarily due to both resource limitations and data accessibility constraints. While multiple courses would provide stronger generalizability, obtaining suitable datasets from other courses proved challenging—many course materials are either not publicly accessible or lack the structured format necessary for systematic evaluation. Additionally, evaluating courses with different question formats (such as open-ended problems or proof-based exercises) would require substantially different evaluation frameworks and metrics, further increasing complexity and computational costs beyond our budget. Focusing on a single course

with a well-defined, multiple-choice test set allowed us to deeply investigate the effectiveness of synthetic data in a controlled setting, but it also limits the generalizability of our results. We cannot determine whether the phenomena observed in probability and statistics (such as fine-tuning failing to improve performance) are specific to this subject or universally applicable. Our inability to definitively isolate quantization effects without full-precision baseline training represents an important limitation. The lack of a systematic synthetic data quality audit similarly limits causal claims. Future research should include these comparisons to definitively establish whether performance limitations stem from data quality, quantization constraints, or inherent task difficulty.

6.5 Final Remarks

This work answers the three guiding research questions as follows: (1) Fine-tuning provided limited benefits, revealing the critical role of data quality and alignment over parameter adjustment. (2) RAG offered the most effective and computationally efficient optimization strategy, outperforming both fine-tuning and prompt engineering. (3) RAG-augmented local models approached the performance of commercial systems while maintaining institutional privacy, cost-efficiency, and autonomy.

Local language models can effectively support university-level probability and statistics assessment when enhanced with retrieval augmentation. While local model alone fall short of Online LLM, retrieval augmentation bridges most of this gap to an educationally meaningful 83.3%, while providing decisive advantages in privacy, cost, and institutional control.

Moreover, the findings of this thesis have broader implications for the integration of large language models in educational contexts, particularly regarding assessment integrity. Our results show that, although baseline local models alone are insufficient, RAG-enhanced local models can reach educationally meaningful accuracy, highlighting a realistic risk that students could leverage such systems in closed-internet exam settings. These observations underscore the importance of designing assessment protocols that balance AI-assisted learning with robust academic integrity safeguards.

Furthermore, this challenge necessitates a understanding of what we are actually measuring: if students can access LLM assistance, then traditional examinations may no longer validly assess individual mastery but rather the capacity to formulate effective prompts and critically evaluate model outputs. Therefore, the integration of LLMs in educational contexts requires not merely technological solutions, but a fundamental rethinking of the meaning of assessment.

Bibliography

[1] Paul Newton and Maria Xiromeriti. “ChatGPT Performance on Multiple Choice Question Examinations in Higher Education. A Pragmatic Scoping Review”. In: *Assessment & Evaluation in Higher Education* 49.6 (2024), pp. 781–798. DOI: 10.1080/02602938.2023.2299059.

[2] Daniel Martin Katz et al. “GPT-4 Passes the Bar Exam”. In: *Philosophical Transactions of the Royal Society A* 382.2270 (2024), p. 20230254. DOI: 10.1098/rsta.2023.0254.

[3] “GPT Takes the Bar Exam”. In: *arXiv preprint arXiv:2212.14402* (2022). URL: https://arxiv.org/abs/2212.14402.

[4] Daniel Stribling, Yuxing Xia, Mohammad K Amer, et al. “The model student: GPT-4 performance on graduate biomedical science exams”. In: *Scientific Reports* 14.1 (2024), p. 5670. DOI: 10.1038/s41598-024-55568-7.

[5] Harsha Nori et al. “Capabilities of GPT-4 on Medical Challenge Problems”. In: *arXiv preprint arXiv:2303.13375* (2023). URL: https://arxiv.org/abs/2303.13375.

[6] Desnes Nunes et al. “Evaluating GPT-3.5 and GPT-4 Models on Brazilian University Admission Exams”. In: *arXiv preprint arXiv:2303.17003* (2023). URL: https://arxiv.org/abs/2303.17003.

[7] “Generative AI Takes a Statistics Exam: A Comparison of Performance between ChatGPT3.5, ChatGPT4, and ChatGPT4o-mini”. In: *arXiv preprint arXiv:2501.09171* (2025). URL: https://arxiv.org/abs/2501.09171.

[8] The Rider Online. *Artificial Intelligence vs. Academic Integrity*. 2024. URL: https://therideronline.com/top-story/2024/10/artificial-intelligence-vs-academic-integrity/.

[9] Hao-Ping (Hank) Lee et al. “The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers”. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI ’25. Yokohama, Japan: ACM, 2025. DOI: 10.1145/3706598.3713778.

[10] Himendra Balalle and Sachini Pannilage. “Reassessing academic integrity in the age of AI: A systematic literature review on AI and academic integrity”. In: *Social Sciences & Humanities Open* 11 (2025), p. 101299. DOI: 10.1016/j.ssaho.2025.101299.

[11] Ali Ateeq et al. “Artificial intelligence in education: implications for academic integrity and the shift toward holistic assessment”. In: *Frontiers in Education* 9 (2024), p. 1470979. DOI: 10.3389/feduc.2024.1470979.

[12] ChatTutor Team. *ChatTutor: AI-Powered Tutoring Platform*. AI Teaching Assistant Platform. DTU-based startup for AI-assisted learning. Technical University of Denmark, 2025. URL: https://chattutor.io/.

[13] University of Illinois Urbana-Champaign. *UIUC Chat: course assistant tool*. Documentation for UIUC's course assistant tool using LLMs with retrieval-augmented generation. 2024. URL: https://docs.uiuc.chat/ (visited on 01/09/2025).

[14] DTU. *DTU opens up for the use of artificial intelligence in teaching*. 2024. URL: https://www.dtu.dk/english/newsarchive/2024/01/dtu-opens-up-for-the-use-of-artificial-intelligence-in-teaching.

[15] "Narrowing the Gap: Supervised Fine-Tuning of Open-Source LLMs as a Viable Alternative to Proprietary Models for Pedagogical Tools". In: *arXiv preprint arXiv:2507.05305* (2025). URL: https://arxiv.org/abs/2507.05305.

[16] Venkatesh Balavadhani Parthasarathy et al. "The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities". In: *arXiv preprint arXiv:2408.13296* (2024). URL: https://arxiv.org/abs/2408.13296.

[17] "Instruction Tuning for Large Language Models: A Survey". In: *arXiv preprint arXiv:2308.10792* (2024). URL: https://arxiv.org/abs/2308.10792.

[18] "Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations". In: *arXiv preprint arXiv:2404.10779* (2024). URL: https://arxiv.org/abs/2404.10779.

[19] "Supervised Fine-Tuning LLMs to Behave as Pedagogical Agents in Programming Education". In: *arXiv preprint arXiv:2502.20527* (2025). URL: https://arxiv.org/abs/2502.20527.

[20] "Towards Pedagogical LLMs with Supervised Fine Tuning for Computing Education". In: *arXiv preprint arXiv:2411.01765* (2024). URL: https://arxiv.org/abs/2411.01765.

[21] "From Automation to Augmentation: Large Language Models Elevating Essay Scoring Landscape". In: *arXiv preprint arXiv:2401.06431* (2024). URL: https://arxiv.org/abs/2401.06431.

[22] "Teaching LLMs How to Learn with Contextual Fine-Tuning". In: *arXiv preprint arXiv:2503.09032* (2025). URL: https://arxiv.org/abs/2503.09032.

[23] "Finetuning LLMs for Comparative Assessment Tasks". In: *arXiv preprint arXiv:2409.15979* (2024). URL: https://arxiv.org/abs/2409.15979.

[24] OpenAI. "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774* (2024). URL: https://arxiv.org/abs/2303.08774.

[25] Tom Brown et al. "Language Models are Few-Shot Learners". In: *arXiv preprint arXiv:2005.14165* (2020). URL: https://arxiv.org/abs/2005.14165.

[26] "An Automated Explainable Educational Assessment System Built on LLMs". In: *arXiv preprint arXiv:2412.13381* (2024). URL: https://arxiv.org/abs/2412.13381.

[27] "Evaluating GPT-4 at Grading Handwritten Solutions in Math Exams". In: *arXiv preprint arXiv:2411.05231* (2024). URL: https://arxiv.org/abs/2411.05231.

[28] "Evaluating LLM Metrics Through Real-World Capabilities". In: *arXiv preprint arXiv:2505.08253* (2025). URL: https://arxiv.org/abs/2505.08253.

[29] Patrick Lewis et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. In: *arXiv preprint arXiv:2005.11401* (2021). URL: https://arxiv.org/abs/2005.11401.

[30] Yunfan Gao et al. “Retrieval-Augmented Generation for Large Language Models: A Survey”. In: *arXiv preprint arXiv:2312.10997* (2024). URL: https://arxiv.org/abs/2312.10997.

[31] Shailja Gupta et al. “A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions”. In: *arXiv preprint arXiv:2410.12837* (2024). URL: https://arxiv.org/abs/2410.12837.

[32] Siyun Zhao et al. “Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely”. In: *arXiv preprint arXiv:2409.14924* (2024). URL: https://arxiv.org/abs/2409.14924.

[33] “Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG”. In: *arXiv preprint arXiv:2501.09136* (2025). URL: https://arxiv.org/abs/2501.09136.

[34] Siran Li et al. “Enhancing Retrieval-Augmented Generation: A Study of Best Practices”. In: *arXiv preprint arXiv:2501.07391* (2025). URL: https://arxiv.org/abs/2501.07391.

[35] Shangyu Wu et al. “Retrieval-Augmented Generation for Natural Language Processing: A Survey”. In: *arXiv preprint arXiv:2407.13193* (2025). URL: https://arxiv.org/abs/2407.13193.

[36] Technical University of Denmark. *02405 Probability*. Course Materials, Technical University of Denmark. 2025. URL: https://www2.compute.dtu.dk/courses/02405/.

[37] Lukas Blecher et al. “Nougat: Neural Optical Understanding for Academic Documents”. In: *arXiv preprint arXiv:2308.13418* (2023). URL: https://arxiv.org/abs/2308.13418.

[38] Mathpix. *Mathpix OCR: Extract LaTeX, tables, and text from PDFs and images*. 2024. URL: https://mathpix.com/.

[39] Longhui Yu et al. “MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models”. In: *arXiv preprint arXiv:2309.12284* (2023). URL: https://arxiv.org/abs/2309.12284.

[40] Tim Dettmers et al. “QLoRA: Efficient Finetuning of Quantized LLMs”. In: *Advances in Neural Information Processing Systems* 36 (2023). DOI: 10.48550/arXiv.2305.14314. URL: https://arxiv.org/abs/2305.14314.

[41] Horia Alexandru Modran et al. “LLM Intelligent Agent Tutoring in Higher Education Courses Using a RAG Approach”. In: *Futureproofing Engineering Education for Global Responsibility*. Vol. 1281. Lecture Notes in Networks and Systems. Cham: Springer, 2025, pp. 589–599. DOI: 10.1007/978-3-031-83520-9_54.

[42] Youngjin Lee. “Developing a Computer-Based Tutor Utilizing Generative Artificial Intelligence (GAI) and Retrieval-Augmented Generation (RAG)”. In: *Education and*

Information Technologies 30 (2025), pp. 7841–7862. DOI: 10.1007/s10639-024-13129-5.

[43] Zongxi Li et al. “Retrieval-Augmented Generation for Educational Application: A Systematic Survey”. In: *Computers and Education: Artificial Intelligence* 8 (2025), p. 100384. DOI: 10.1016/j.caeai.2025.100384. URL: https://www.sciencedirect.com/science/article/pii/S2666920X25000578.

A Appendices

A.1 Code

All implementations, together with configurations and fine-tuned model, are available at **https://github.com/ZpQiao/Assessing-Exam-Validity-with-Finetuned-Local-Language-Models.**

A.2 Evaluation Results

An additional archive `thesis_appendix_evaluation_results.zip` is included with this thesis. It contains all datasets and detailed evaluation results used in the experiments. The directory structure is as follows:

```
thesis_appendix_evaluation_results/
  probability_train_set.jsonl
  probability_test_set.jsonl
  probability_augmented_train_set.jsonl
  results_Qwen3-14B_base_da.csv
  results_Qwen3-14B_base_en.csv
  results_Qwen3-14B_finetuned_all.csv
  results_Qwen3-14B_rag_da.csv
  results_Qwen3-14B_rag_eng.csv
```

A.3 Knowledge Point Distribution

Table A.1: Knowledge point distribution in the DTU probability exam dataset (train, test, and combined).

ID	Knowledge point	Train N	Test N	All N
1	Normal distribution	48	19	67
2	Exponential distribution	50	15	65
3	Binomial distribution	49	15	64
4	Standardization to standard normal (Z-score)	49	14	63
5	Conditional probability (multiplication rule)	51	8	59
6	Poisson distribution	42	12	54
7	Law of total probability	34	8	42
8	Marginal density from joint (integration)	30	5	35
9	Variance of sum using covariance	26	8	34
10	Change of variables (univariate)	25	4	29
11	Gamma distribution	20	6	26
12	Linearity of expectation	21	5	26
13	Central Limit Theorem (sum of i.i.d.)	19	5	24
14	Order statistics (kth order statistic)	19	5	24

Continued on next page

Table A.1 (continued)

ID	Knowledge point	Train N	Test N	All N
15	Covariance from correlation (Cov=ρσXσY)	15	8	23
16	Complement rule (at least one)	14	7	21
17	Geometric distribution	17	4	21
18	Hypergeometric distribution	20	1	21
19	Law of total expectation	16	5	21
20	Uniform(0,1) distribution	19	2	21
21	Beta distribution	17	2	19
22	Geometric probability (area ratio)	17	2	19
23	Inclusion–exclusion principle	14	3	17
24	Order statistics (maximum)	12	5	17
25	Bivariate normal quadrant probability	11	5	16
26	Correlation vs independence	11	5	16
27	Hazard rate (f/S)	15	1	16
28	Probability from joint density (double integral)	12	4	16
29	Bivariate normal conditional expectation	12	3	15
30	Probability from PMF (sum)	13	2	15
31	Expectation from PDF (integration)	10	3	13
32	Uniform distribution on a triangular region	8	5	13
33	Rayleigh distribution	10	2	12
34	Small-interval probability via PDF	9	3	12
35	Variance of Binomial (np(1-p))	9	3	12
36	Chebyshev inequality	8	1	9
37	Chi-square distribution	7	2	9
38	Discrete uniform distribution	6	2	8
39	Cantelli inequality (one-sided)	5	2	7
40	Convolution (sum of independent RVs)	6	1	7
41	Covariance from definition	4	3	7
42	Expectation from joint density	5	2	7
43	Probability from PDF (integration)	6	1	7
44	Weibull distribution	5	2	7
45	Geometric probability (uniform disk)	5	1	6
46	Negative binomial distribution	3	3	6
47	Expectation from PMF (discrete)	4	1	5
48	Lognormal distribution	2	3	5
49	Memoryless property (exponential)	4	1	5
50	PDF validity (nonnegativity/normalization)	4	1	5
51	Continuous uniform on a general interval	2	2	4
52	Indicator variables (I(A) I(B) = I(A∩B))	3	1	4
53	Product expectation (independent)	2	2	4
54	Sum of independent Normals	3	1	4
55	Survival function properties	3	1	4
56	Quantile from survival function	1	2	3
57	System reliability (parallel)	2	1	3
58	Half-normal distribution (|Z|)	1	1	2
59	Joint PDF of independent RVs (product of marginals)	1	1	2

Continued on next page

Table A.1 (continued)

ID	Knowledge point	Train N	Test N	All N
60	Markov inequality	1	1	2
61	Multinomial distribution	2	0	2
62	Size-biased sampling	1	1	2
63	Standard error of the mean (i.i.d.)	1	1	2
64	Variance of Poisson (Var=λ)	1	1	2
65	Variance of product (independent)	1	1	2
66	Expectation is a constant	1	0	1
67	Expected absolute value of standard normal	1	0	1
68	Probability from joint CDF	1	0	1
69	Variance from PMF	1	0	1

Technical
University of
Denmark

Richard Petersens Plads, Building 324
2800 Kgs. Lyngby
Tlf. 4525 1700

https://www.compute.dtu.dk/