Ontology-Enhanced Contextual Reasoning for Large Language Models in STEM Education

by

Kinlo Ephriam Tangiri

Bachelor Thesis in Computer Science

Submission: May 19, 2025 Supervisor: Prof. Dr. Fatahi Valilai, Omid

Constructor University | School of Computer Science and Engineering

Statutory Declaration

Family Name, Given/First Name	Ephriam Tangiri, Kinlo
Matriculation number	30006363
Kind of thesis submitted	Bachelor Thesis

English: Declaration of Authorship

I hereby declare that the thesis submitted was created and written solely by myself without any external support. Any sources, direct or indirect, are marked as such. I am aware of the fact that the contents of the thesis in digital form may be revised with regard to usage of unauthorized aid as well as whether the whole or parts of it may be identified as plagiarism. I do agree my work to be entered into a database for it to be compared with existing sources, where it will remain in order to enable further comparisons with future theses. This does not grant any rights of reproduction and usage, however.

This document was neither presented to any other examination board nor has it been published.

German: Erklärung der Autorenschaft (Urheberschaft)

Ich erkläre hiermit, dass die vorliegende Arbeit ohne fremde Hilfe ausschließlich von mir erstellt und geschrieben worden ist. Jedwede verwendeten Quellen, direkter oder indirekter Art, sind als solche kenntlich gemacht worden. Mir ist die Tatsache bewusst, dass der Inhalt der Thesis in digitaler Form geprüft werden kann im Hinblick darauf, ob es sich ganz oder in Teilen um ein Plagiat handelt. Ich bin damit einverstanden, dass meine Arbeit in einer Datenbank eingegeben werden kann, um mit bereits bestehenden Quellen verglichen zu werden und dort auch verbleibt, um mit zukünftigen Arbeiten verglichen werden zu können. Dies berechtigt jedoch nicht zur Verwendung oder Vervielfältigung.

Diese Arbeit wurde noch keiner anderen Prüfungsbehörde vorgelegt noch wurde sie bisher veröffentlicht.

. . . .

Date, Signature

Abstract

Large Language Models (LLMs) have transformed the way we interact with technology, yet their tendency to hallucinate (to confidently output incorrect information) presents significant challenges, especially in educational applications where accuracy is crucial.

This thesis aims to investigate whether integrating ontology-driven knowledge models with LLMs can enhance their output for STEM education through more reliable and contextually aware responses with reduced hallucinations (for the purpose of this research we will focus on Newton's laws in physics). To address the hallucination problem, we propose an approach that combines Ontology Web Language(OWL), and Resource Description Framework(RDF) and SPARQL technologies to systematically organize educational data including learner profiles, learning objectives, and instructional materials.

Our methodology involves converting structured ontological knowledge into vector embeddings, which are then integrated with Anthropic Claude's LLM through its extensive context window. This integration, enables more accurate and contextually-aware tutoring interactions and LLM responses. We evaluate the system's effectiveness through comparative analysis of responses generated by our ontology-enhanced system versus our baseline model, with particular focus on factual accuracy and contextual relevance.

The benefits of this system include significantly reduced hallucination, improved contextual understanding, and elimination the need for prompt engineering requirements particularly in secondary and high school education where students may not yet know what information is relevant. This thesis aims to demonstrate the advantages of an ontology-driven system over traditional LLM-only or rule-based approaches in creating personalized educational experiences.

Contents

Chapter 1

Introduction

Large Language Models (LLMs) have transformed the way we use and apply artificial intelligence. Their ability to understand and generate human-like text offers numerous opportunities for educational applications. STEM education, in particular, could benefit from LLMs' capabilities to explain complex concepts, provide interactive tutoring, and adapt to individual learning needs especially for schools that uses the one-size-fit all approach method for teaching.

However, LLMs face a critical challenge: hallucination. These models often generate plausible but factually incorrect or nonsensical information [24]. In STEM education, where accuracy is extremely important, such hallucinations pose significant risks. Recent studies show that LLM hallucinations occur in up to 27% of responses involving technical concepts [7]. This unreliability limits their deployment in educational settings.

This thesis addresses a fundamental question: **How can we harness LLMs' potential(Natural Language Understanding) for STEM education while ensuring their responses remain accurate and reliable?**?

Traditional approaches fall short:

- Pure LLM-based systems risk propagating misinformation through hallucinations
- Rule-based systems offer accuracy but lack the natural interaction capabilities needed for effective education
- Prompt engineering is complex and requires domain Knowledge
- Fine-tuning LLMs is resource-intensive, time-consuming, and requires domain Knowledge

Through this thesis, we are proposing a hybrid solution: integrating ontological knowledge structures with LLM reasoning capabilities. This approach combines:

- The structured precision of domain-specific ontologies
- The natural language understanding of LLMs
- Real-time verification mechanisms

1.1 Research Objectives

Our research focuses on four primary objectives:

- Develop a framework that integrates ontological knowledge with LLM reasoning
- Build an adaptive system that provides personalized, accurate learning experiences
- Create mechanism that reduces the rate of hallucinations using ontological constraints
- Implement a real-time verification system that ensures accuracy and reliability

1.2 Research Contributions

This work advances the field through several key contributions:

- **Technical Innovation:**
  - Ontology-enhanced LLM architecture for STEM education
  - Efficient integration of structured and unstructured knowledge
- **Educational Advancement:**
  - Context-aware response generation for STEM concepts
  - Adaptive learning pathways with accuracy guarantees
  - Personalized feedback mechanisms
- **Scientific Insights:**
  - Methods to reduce hallucination in domain-specific LLM applications
  - Techniques for semantic verification of LLM outputs
  - Approaches to maintain engagement while ensuring accuracy

1.3 Technical Foundations

Our approach builds upon recent advances in several areas:

- **LLM Architecture:** Latest developments in transformer models and attention mechanisms [3, 23]
- **Ontology Engineering:** Semantic web technologies and knowledge representation [6, 18]
- **Hallucination Mitigation:** Recent techniques in fact verification and constraint satisfaction [25, 24]
- **Educational Technology:** Adaptive learning systems and cognitive load theory [22, 17, 9]

1.4 Research Challenges

We address two categories of challenges:

1.4.1 Conceptual Challenges

- Bridging semantic gaps between ontologies and LLM representations
- Maintaining educational engagement while enforcing accuracy
- Developing metrics for hallucination detection and prevention
- Ensuring consistency across different knowledge domains

1.4.2 Technical Challenges

- Efficient integration of OWL/SPARQL ontologies with LLM processing
- Real-time semantic verification of LLM outputs
- Scalable knowledge base management
- Optimization of response latency and resource usage

1.4.3 Mitigation Strategies for Hallucination

Despite advances in LLM technology, hallucination remains a persistent problem. Researchers have explored three broad strategies to address this issue:

- **Prompt Engineering:** This approach involves carefully designing input prompts to guide the model toward more accurate responses. For example, prompts can be enriched with context or specify the desired output format. While prompt engineering can reduce hallucinations in some cases, it requires significant manual effort and domain expertise. It is also difficult to scale and may not generalize well to new topics or user needs.
- **Fine-Tuning:** Fine-tuning retrains the model on curated datasets that emphasize accuracy and factual correctness. This can improve reliability, but maintaining up-to-date, high-quality datasets is resource-intensive. Continuous fine-tuning is often impractical for real-time educational settings due to cost and latency constraints.
- **Ontologies integration:** Integrating formal ontologies into LLM workflows provides structured, domain-specific knowledge. Ontologies define key concepts and relationships, enabling the model to reason with explicit, verifiable information. This approach supports retrieval-augmented generation (RAG) and real-time fact-checking, directly addressing hallucination by relating responses in trusted knowledge frameworks.

Prompt engineering is limited by its manual nature and lack of scalability. Fine-tuning faces challenges in data curation and real-time applicability. Ontology-based grounding, while requiring initial investment in ontology development, offers a sustainable and scalable solution for ensuring accuracy and consistency, that is why our research focus on how impactful ontology models can reduce hallucinations in education applications.

Ontology-based approaches are particularly well-suited for STEM education because they:

- Provide explicit definitions and relationships for complex concepts
- Enable automated, real-time verification of LLM outputs
- Support adaptive, context-aware learning experiences
- Facilitate cross-domain consistency and knowledge integration

By grounding LLM reasoning in ontological structures, we can systematically reduce hallucinations and deliver reliable, personalized educational content. This makes ontology-based mitigation the most promising strategy for domains like STEM education.

1.5 Thesis Structure

This thesis is organized as follows:

- Chapter 2 examines the theoretical foundations and related work in ontology-enhanced LLMs
- Chapter 3 details our approach to integrating ontological knowledge with LLM reasoning
- Chapter 4 presents the system implementation and architecture
- Chapter 5 provides empirical evaluation and results analysis
- Chapter 6 discusses implications and future research directions

To facilitate further research, the complete source code for the ontology-enhanced LLM system developed in this thesis research process is available on GitHub at https://github.com/9117KET/ai-avatar-ontology-integration-poc.git. Further details about the repository structure and deployment are provided in Chapter 4.

Chapter 2

Background and Related Work

2.1 Historical Context and Evolution

The intersection of artificial intelligence and education has evolved rapidly, with Large Language Models (LLMs) and ontological knowledge representation emerging as transformative technologies. Early educational technologies focused on rule-based systems and static content delivery. The advancement of LLMs enabled dynamic, context-aware interactions, while ontologies brought structured, machine-interpretable knowledge to educational platforms [8].

2.2 Significance and Motivation

Despite LLMs' promise, their tendency to hallucinate, that is to generate plausible but incorrect information, poses a critical barrier in STEM education [8, 14]. Ontologies offer a way to constrain and verify LLM outputs, but integrating these approaches remains an open challenge [5, 13]. Addressing this gap is essential for building trustworthy, adaptive educational systems.

2.3 Review of Existing Literature

2.3.1 LLMs in Education

Recent studies highlight LLMs' ability to personalize learning, generate content, and provide instant feedback. However, their lack of domain knowledge leads to factual errors, especially in technical subjects [8, 19, 7].

2.3.2 Ontologies in Educational Technology

Ontologies have been used to structure curricula, model learner knowledge, and support adaptive learning paths [11, 4, 12]. They enable explicit representation of concepts and relationships, supporting automated reasoning and assessment.

2.3.3 Hybrid and Knowledge-Enhanced Systems

Emerging research explores combining LLMs with ontologies or knowledge graphs to improve accuracy and reasoning [5, 18, 14]. These hybrid systems show promise in reducing hallucinations and supporting context-aware responses, but practical, scalable solutions for real-time educational use are still lacking.

2.4 Identified Gaps in Knowledge

While prior work demonstrates the potential of both LLMs and ontologies, few studies have achieved seamless, real-time integration for STEM education. Key gaps include:

- Lack of scalable frameworks for ontology-driven LLM verification in educational settings
- Limited empirical evaluation of hallucination mitigation in real-world classrooms
- Insufficient personalization and adaptability in existing hybrid systems

2.5 Necessity, Relevance, and Innovation of This Study

This thesis addresses these gaps by proposing a novel framework that tightly integrates ontological knowledge with LLM reasoning for STEM education. The approach enables:

- Real-time, ontology-based verification of LLM outputs
- Adaptive, context-aware tutoring tailored to individual learners
- Systematic reduction of hallucinations without sacrificing interactivity

By advancing the state of the art, this work lays the foundation for reliable, scalable, and effective AI-driven education.

This background and literature review establish the foundation for the methodology presented in the following chapter.

2.6 Large Language Models in Education

2.6.1 Evolution and Capabilities

Large Language Models represent a significant advancement in artificial intelligence, particularly in natural language processing and understanding [8]. These models, trained on vast amounts of text data, have demonstrated remarkable capabilities in:

- Natural language understanding and generation
- Context-aware responses and explanations
- Adaptation to various domains and topics
- Multi-turn conversations and reasoning

Recent developments in hybrid alignment training [15] and search engine augmentation [21] have further enhanced these capabilities.

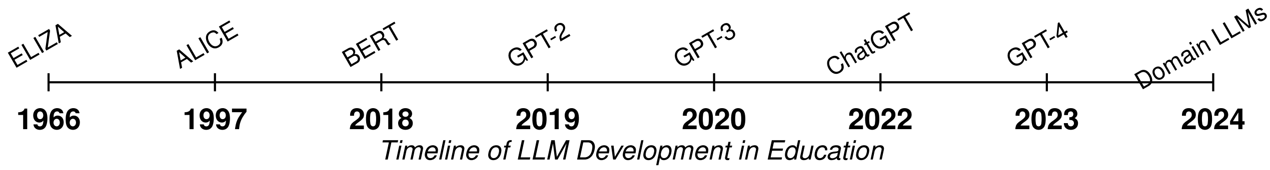


Figure 2.1: Timeline of LLM Development in Education. Key milestones are shown from the first chatbot (ELIZA) to modern, domain-specific LLMs. This visual contextualizes the rapid evolution and growing impact of LLMs in educational technology.

2.6.2 Limitations and Challenges

Despite their capabilities, LLMs face several critical challenges in educational applications [24]:

- **Hallucination:** Generation of plausible but incorrect information
- **Contextual Understanding:** Limited ability to maintain consistent context
- **Domain Specificity:** Challenges in specialized STEM topics
- **Verification:** Difficulty in validating generated responses

Recent research has shown that hallucinations can be both a limitation and a potential source of creative problem-solving [19]. However, in educational contexts, particularly in STEM fields, accuracy remains the primary concern [26].

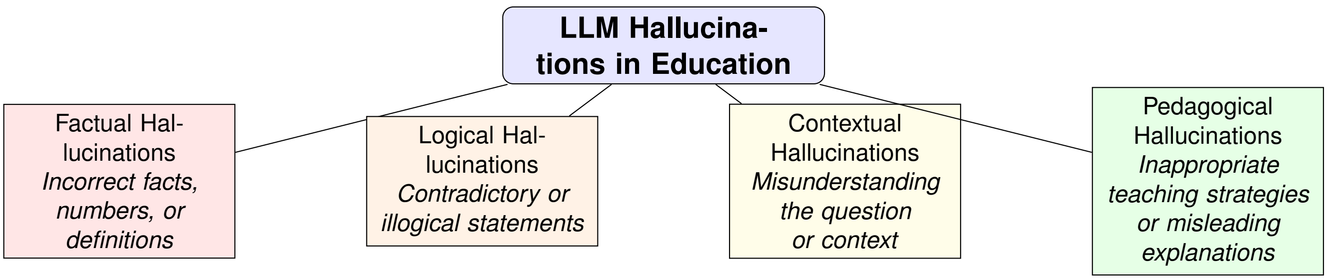


Figure 2.2: Taxonomy of LLM Hallucinations in Education. The diagram categorizes common types of hallucinations produced by language models in educational settings, clarifying the risks and motivating the need for ontology-based verification.

2.7 Ontologies in Knowledge Representation

2.7.1 Fundamentals of Ontological Engineering

Ontologies provide a formal, structured representation of knowledge within a domain [14]. Key components include:

- **Classes and Hierarchies:** Representing concepts and their relationships
- **Properties:** Defining characteristics and relationships
- **Instances:** Specific examples within the ontology

- **Axioms:** Rules and constraints governing relationships

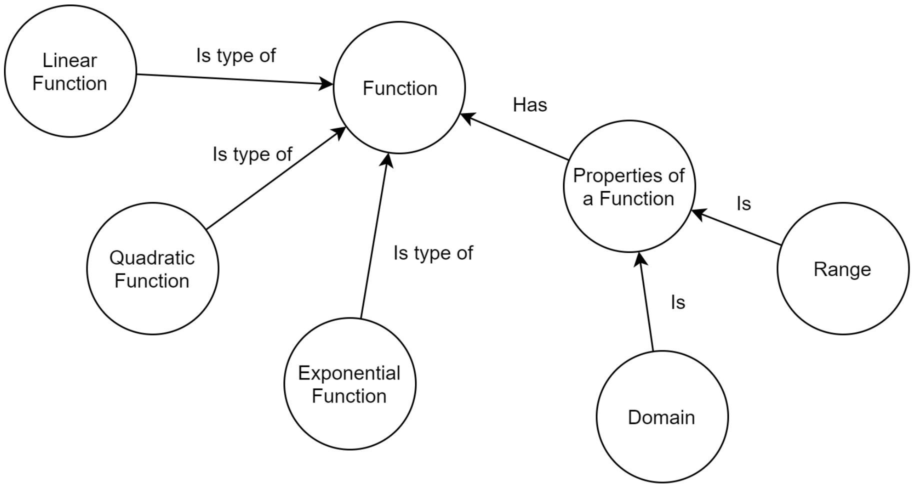


Figure 2.3: Sample ontology structure for STEM education, focusing on mathematical functions and their sub-concepts. Adapted from [1]. This diagram demonstrates how ontologies can organize and relate key concepts, supporting adaptive and personalized learning in AI-driven educational systems.

2.8 Key Concepts and Definitions

2.8.1 Core Terminology

This section defines key terms used throughout this thesis:

- **Ontology:** A structured framework representing knowledge within a specific domain, defining concepts, properties, and relationships for clear communication and effective information retrieval in human-AI interaction.
- **Knowledge Model:** A structured representation of information that explicitly defines concepts, relationships, and logic within a particular domain, facilitating consistent interpretation and accurate reasoning.
- **Large Language Model (LLM):** An advanced AI model trained on extensive textual data, capable of generating human-like text, understanding context, and performing complex reasoning tasks.
- **Virtual Environment:** Digitally simulated spaces designed to mimic real-world scenarios, supporting user immersion and interaction through various sensory elements.
- **Ontology-driven Integration:** The process of using structured ontologies to unify different data sources or systems, ensuring coherent communication and consistency across applications.
- **AI-Human Interaction:** The exchange of information between AI systems and human users, relying on clear understanding and context-aware responses.
- **Semantic Knowledge:** Contextually interpreted information that enables systems to understand implied meanings and relationships between concepts.

- **Knowledge Graph:** A structured data representation organizing information into interconnected nodes and edges, enabling effective visualization and interpretation of complex relationships.

Chapter 3

Methodology and System Design

3.1 Research Approach

This chapter presents our systematic approach to developing an ontology-enhanced LLM system for STEM education. Our methodology addresses the critical challenge of AI hallucination in educational applications through the following research objectives:

- Integration of domain-specific ontologies (OWL, RDF, SPARQL) with LLM reasoning
- Development of mechanisms for reliable AI-powered tutoring
- Enhancement of contextual understanding in STEM education through structured knowledge representation
- Creation of an adaptive, personalized learning system with avatar-based engagement

The research follows a phased development approach focusing on three key areas:

1. **Core Functionality:** Environment setup, API authentication, system prompt structure, and basic question-answering functionality
2. **Knowledge Representation:** Physics concepts, laws, relationships, prerequisites structures, and context retrieval system
3. **Student Model Implementation:** Concept exposure tracking, knowledge level monitoring, and learning path generation

3.2 System Architecture

The system architecture comprises several interconnected components organized into distinct layers:

3.2.1 Core Components

The system consists of the following key components:

- **Backend (API Layer):**
  - Quart (async Flask-compatible) server

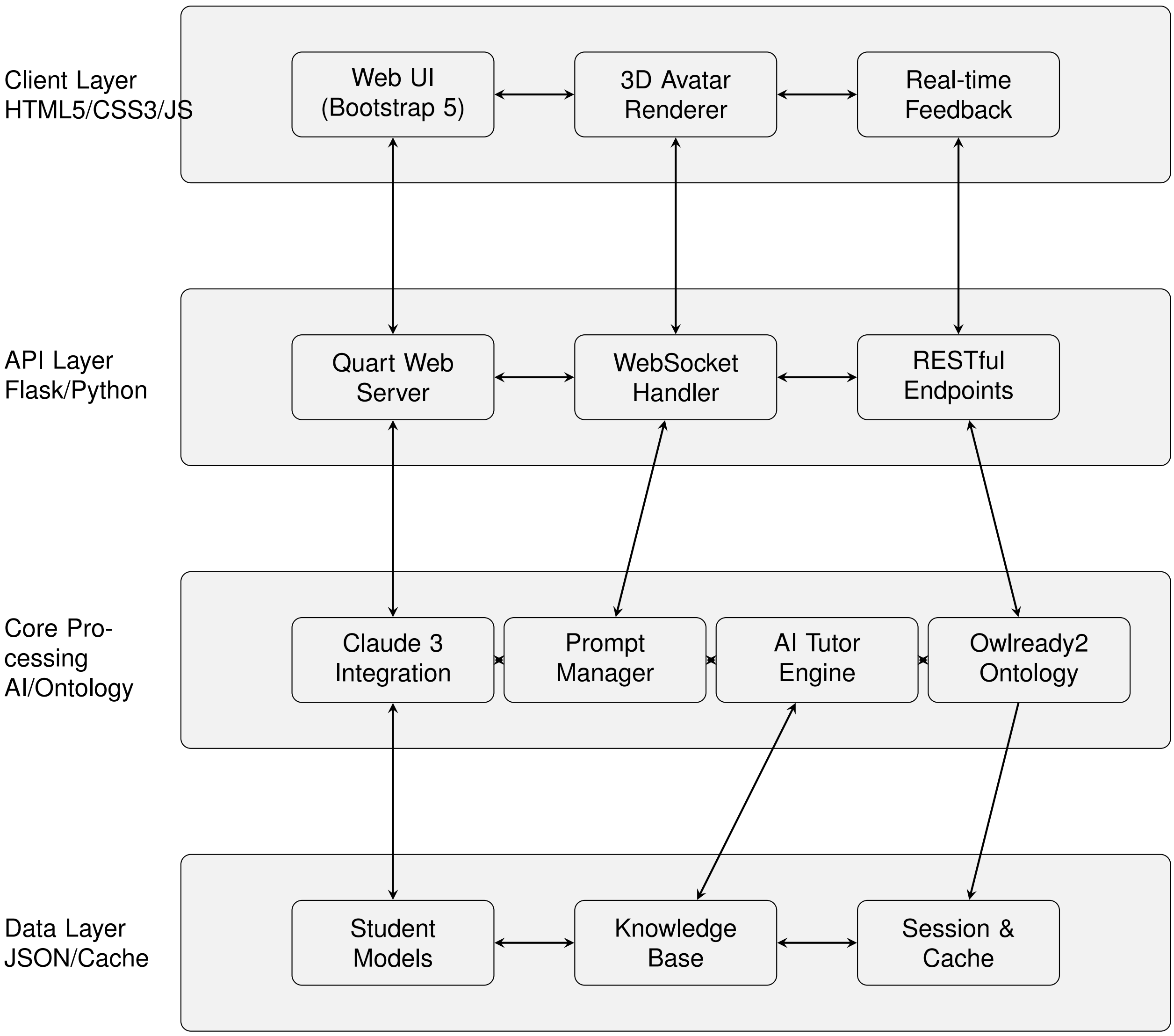


Figure 3.1: High-level System Architecture showing the integration of ontology-driven knowledge models with Claude 3 LLM for enhanced STEM education.

- Modular API endpoints and route definitions
- JWT session management
- Integration with Claude 3 LLM and ontology modules

- **LLM Integration:**
  - Adaptive tutoring logic
  - Student model implementation
  - Context-aware answer generation
  - Response validation against ontology
- **Ontology Layer:**
  - Physics domain ontology schemas
  - Validation mechanisms
  - Knowledge representation structures
  - Semantic reasoning capabilities
- **Frontend Interface:**
  - Avatar-based web UI
  - Interactive tutoring interface
  - Real-time feedback visualization
  - Student progress tracking

3.3 Project Structure

The implementation follows a modular organization:

- **/api**
  - Routes and handler definitions
  - Utility functions
  - Main API application
  - Vercel serverless handler
- **/llm_integration**
  - LLM integration components
  - Student modeling logic
  - Context management
  - Response generation
- **/ontology**
  - Physics ontology schemas

  - Validation mechanisms
  - Knowledge base structures
  - Reasoning components
- **/static**
  - Frontend assets (HTML, CSS, JS)
  - Avatar interface components
  - UI/UX elements
  - Interactive features

3.4 Evaluation Metrics

The system evaluation is conducted across three primary dimensions

1. **System Performance**
   - Response accuracy measurement
   - Personalization effectiveness
   - System scalability assessment
   - Integration reliability
2. **User Experience**
   - Interaction quality metrics
   - Learning engagement levels
   - Knowledge retention rates
   - Avatar interaction effectiveness
3. **Technical Evaluation**
   - API integration performance
   - Knowledge representation accuracy
   - Data management efficiency
   - Response validation success

3.5 Research Contributions

The key contributions of this research include:

1. **Reduced Hallucination:** Integration of structured domain knowledge to enhance AI accuracy
2. **Personalized Tutoring:** Adaptive responses based on comprehensive student modeling

3. **Scalable Solution:** Architecture supporting one-on-one tutoring at scale
4. **Interactive Learning:** Avatar-based interface for enhanced student engagement
5. **Knowledge Verification:** Systematic validation of AI responses against domain ontology

This methodology provides a comprehensive framework for developing and evaluating our ontology-enhanced LLM system, ensuring alignment with our research objectives and educational goals.

Chapter 4

Implementation

4.1 Overview

This chapter details the technical implementation of our ontology-enhanced LLM system, following best practices in educational technology integration. The implementation focuses on creating a simple prove of concept for our ontology-enhanced LLM system.

4.2 Ontology Development

The ontology was developed using OWL and RDF technologies, following semantic web best practices

- **Core Concepts:**
  - Defined fundamental physics concepts (force, motion, energy)
  - Established concept hierarchies and relationships
  - Implemented domain-specific constraints
  - Created semantic linkages between related concepts
- **Properties:**
  - Object properties for concept relationships
  - Data properties for concept attributes
  - Annotation properties for metadata
  - Inverse relationships for bidirectional navigation
- **Axioms:**
  - Logical constraints for knowledge consistency
  - Domain and range restrictions
  - Cardinality constraints
  - Transitivity rules for concept prerequisites
- **Instances:**

- Real-world examples of concepts
- Practice problems and solutions
- Common misconceptions and corrections
- Application scenarios

4.2.1 Ontology Schema Implementation

The physics tutoring ontology was implemented in OWL/RDF format, defining key classes, properties, and relationships. Below is a snippet from our ontology schema that demonstrates the structure:

```
<!-- Classes -->
<owl:Class rdf:about="\#Concept"/>
<owl:Class rdf:about="\#PhysicalQuantity"/>
<owl:Class rdf:about="\#Law"/>
<owl:Class rdf:about="\#Unit"/>
<owl:Class rdf:about="\#Formula"/>
<owl:Class rdf:about="\#Principle"/>
<owl:Class rdf:about="\#Example"/>
<owl:Class rdf:about="\#Application"/>
<owl:Class rdf:about="\#Topic"/>

<!-- Object Properties -->
<owl:ObjectProperty rdf:about="\#hasPrerequisite">
    <rdfs:domain rdf:resource="\#Concept"/>
    <rdfs:range rdf:resource="\#Concept"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="\#hasUnit">
    <rdfs:domain rdf:resource="\#PhysicalQuantity"/>
    <rdfs:range rdf:resource="\#Unit"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="\#hasFormula">
    <rdfs:domain rdf:resource="\#Law"/>
    <rdfs:range rdf:resource="\#Formula"/>
</owl:ObjectProperty>

<!-- Data Properties -->
<owl:DatatypeProperty rdf:about="\#hasDefinition">
    <rdfs:domain rdf:resource="\#Concept"/>
    <rdfs:range
       rdf:resource="http://www.w3.org/2001/XMLSchema\#string"/>
</owl:DatatypeProperty>
```

Listing 4.1: Physics Tutor Ontology Schema

The implementation of the ontology layer is very important for providing the structural knowledge framework that guides our LLM tutor. Our ontology is implemented using the Resource Description Framework (RDF) and Web Ontology Language (OWL), which provide a standardized approach to knowledge representation [6].

The ontology includes instances of physics concepts with their relationships:

```xml
<!-- Physical Quantities -->
<PhysicalQuantity rdf:about="\#Force">
    <hasDefinition>Force is a push or pull that can change the
       motion of an object</hasDefinition>
    <hasUnit rdf:resource="\#Newton"/>
    <isPartOf rdf:resource="\#NewtonsLaws"/>
    <relatesTo rdf:resource="\#Acceleration"/>
    <relatesTo rdf:resource="\#Mass"/>
    <hasApplication rdf:resource="\#RocketPropulsion"/>
    <isUsedIn rdf:resource="\#FEqualsMA"/>
</PhysicalQuantity>

<!-- Laws -->
<Law rdf:about="\#NewtonsSecondLaw">
    <hasDefinition>Newton's Second Law states that the
       acceleration of an object is directly proportional
    to the net force acting on it and inversely proportional to
       its mass</hasDefinition>
    <hasPrerequisite rdf:resource="\#Force"/>
    <hasPrerequisite rdf:resource="\#Mass"/>
    <hasPrerequisite rdf:resource="\#Acceleration"/>
    <hasFormula rdf:resource="\#FEqualsMA"/>
    <isPartOf rdf:resource="\#NewtonsLaws"/>
    <hasApplication rdf:resource="\#SportsPerformance"/>
</Law>
```

Listing 4.2: Physics Concept Instances

4.3 Knowledge Base Integration

Integrating the ontology with the LLM involves establishing connections between the structured knowledge in our ontology and the capabilities of the language model. This section details the implementation of this integration.

- **SPARQL Queries:**
  - Optimized query patterns for concept retrieval
  - Context-aware knowledge extraction
  - Prerequisite relationship traversal
  - Performance-optimized query execution
- **Context Management:**
  - Dynamic context window optimization
  - Conversation history tracking
  - Domain-specific context prioritization
  - Real-time context adaptation
- **Response Generation:**
  - Ontology-guided response validation

– Semantic consistency checking
– Personalized content adaptation
– Educational scaffolding integration

4.3.1 Ontology Query Implementation

The system implements efficient ontology querying to extract relevant context for student questions. Below is the implementation of our context retrieval method:

```
def _get_relevant_context(self, question: str) -> tuple[str,
   List[str]]:
    """
    Extract relevant context from the ontology based on the user's
       question.

    Returns:
        tuple: (context_text, list_of_concepts_covered)
    """
    context = []
    concepts_covered = []

    # Convert question to lowercase for case-insensitive matching
    question_lower = question.lower()

    # Check for specific Newton's Laws
    law_mappings = {
        "newton's first law": "NewtonsFirstLaw",
        "law of inertia": "NewtonsFirstLaw",
        "newton's second law": "NewtonsSecondLaw",
        "f = ma": "NewtonsSecondLaw",
        "newton's third law": "NewtonsThirdLaw",
        "action reaction": "NewtonsThirdLaw"
    }

    for question_law, ontology_law in law_mappings.items():
        if question_law in question_lower:
            law_obj = self.onto.search_one(iri=f"*{ontology_law}")
            if law_obj:
                concepts_covered.append(ontology_law)
                context.append(f"Law: {ontology_law}")
                if hasattr(law_obj, 'hasDefinition') and
                   len(law_obj.hasDefinition) > 0:
                    context.append(f"Definition:
                       {law_obj.hasDefinition[0]}")
                if hasattr(law_obj, 'hasPrerequisite'):
                    context.append("Prerequisites:")
                    for prereq in law_obj.hasPrerequisite:
                        concepts_covered.append(prereq.name)
                        if hasattr(prereq, 'hasDefinition'):
                            context.append(f"- {prereq.name}:
                               {prereq.hasDefinition[0]}")
                return "\n".join(context), concepts_covered

    # Additional context retrieval logic...
```

```
    return "\n".join(context), list(set(concepts_covered))
```

Listing 4.3: Ontology Context Retrieval

4.4 System Components

4.4.1 Frontend Implementation

The frontend implementation follows modern web development practices and educational technology standards:

- **HTML/CSS:**
  - Bootstrap 5 framework for responsive design
  - Accessible UI components
  - Mobile-first approach
  - Progressive enhancement
- **JavaScript:**
  - ES6+ features for modern functionality
  - Asynchronous content updates
  - Real-time interaction handling
  - Client-side validation

4.4.2 Backend Implementation

The backend architecture emphasizes scalability and reliability, incorporating best practices from adaptive learning systems [17]:

- **Quart Server:**
  - Asynchronous request handling
  - WebSocket support for real-time updates
  - Rate limiting and request validation
  - Error handling and recovery
- **Session Management:**
  - Secure session tracking
  - State persistence
  - Concurrent session handling
  - Session timeout management
- **Data Storage:**
  - JSON-based student data management

– Efficient data retrieval patterns
– Data backup and recovery
– Cache optimization

The backend server was initially implemented using Quart and later migrated to Fast due to how unstable and error prompt the Quart framework was. FastAPI is an asynchronous Python web framework that is compatible with the Flask API [10]. FastAPI was chosen for its support of asynchronous request handling, which is essential for managing multiple concurrent tutoring sessions while maintaining responsiveness.

4.4.3 API Implementation

The system implements a secure, asynchronous API using FastAPI to handle tutoring requests:

```
@app.route('/api/ask', methods=['POST'])
async def ask_tutor():
    """API endpoint to ask the tutor a question with input
       validation."""
    try:
        data = await request.get_json()
        if not data or 'question' not in data:
            return jsonify({'error': 'Question is required'}), 400

        session_id = data.get('session_id', 'default_session')
        question = data['question']

        # Input validation
        if not isinstance(question, str) or len(question) > 1000:
            return jsonify({'error': 'Invalid question format'}),
                400

        # Get the tutor instance for this session
        tutor = get_tutor(session_id)

        # Get the tutor's response
        response = await tutor.tutor(question)

        # Return the response with student model data
        return jsonify({
            'response': response,
            'student_model': {
                'exposed_concepts':
                    list(tutor.student_model.exposed_concepts),
                'understood_concepts':
                    list(tutor.student_model.understood_concepts),
                'knowledge_level':
                    tutor.student_model.knowledge_level
            }
        })

    except Exception as e:
        logger.error(f"Error processing question: {e}")
        return jsonify({'error': 'Internal server error'}), 500
```

Listing 4.4: API Implementation

4.5 Student Model Implementation

The student model tracks learning progress and adapts tutoring to individual needs:

```
class StudentModel:
    """
    Tracks a student's knowledge state, interaction history, and
       learning progress.
    """

    def __init__(self, student_id: str, data_path: Optional[str] =
       None):
        """Initialize a new student model or load an existing
           one."""
        self.student_id = student_id
        self.data_path = data_path or os.path.join(
            os.path.dirname(os.path.dirname(os.path.abspath(__file__))),
            'data',
            'students'
        )

        # Create data directory if it doesn't exist
        os.makedirs(self.data_path, exist_ok=True)

        # Initialize student model attributes
        self.exposed_concepts: Set[str] = set()  # Concepts the
           student has seen
        self.understood_concepts: Set[str] = set()  # Concepts the
           student understands
        self.misconceptions: Dict[str, str] = {}  # Concept name
           -> description of misconception
        self.quiz_results: List[Dict] = []  # List of quiz results
        self.interaction_history: List[Dict] = []  # List of
           interactions
        self.knowledge_level: Dict[str, float] = {}  # Concept
           name -> knowledge level (0.0 to 1.0)

        # Try to load existing data
        self._load_data()

    def update_quiz_result(self, concept: str, correct: bool,
       confidence: float) -> None:
        """
        Update the student model with a quiz result.
        """
        # Record the quiz result
        quiz_result = {
            'timestamp': datetime.now().isoformat(),
            'concept': concept,
            'correct': correct,
            'confidence': confidence
```

```
        }
        self.quiz_results.append(quiz_result)

        # Update knowledge level for the concept
        current_level = self.knowledge_level.get(concept, 0.0)

        if correct:
            # If correct, increase knowledge level, but weight by confidence
            self.knowledge_level[concept] = min(1.0, current_level + (0.2 * confidence))

            # If knowledge level is high enough, add to understood concepts
            if self.knowledge_level[concept] >= 0.7:
                self.understood_concepts.add(concept)
                # Remove any misconceptions about this concept
                if concept in self.misconceptions:
                    del self.misconceptions[concept]
        else:
            # If incorrect, decrease knowledge level
            self.knowledge_level[concept] = max(0.0, current_level - (0.1 * confidence))

            # If they were very confident but wrong, might be a misconception
            if confidence > 0.7:
                # We'd need more logic to identify the specific misconception
                pass

        # Save the updated model
        self.save()
```

Listing 4.5: Student Model Implementation

4.6 Integration Challenges

We addressed several key challenges during implementation, drawing from recent research in LLM integration [7]:

- **Data Consistency:**
  - Ontology-LLM response alignment
  - Real-time verification mechanisms
  - Conflict resolution strategies
  - Version control for knowledge updates
- **Performance:**
  - Query optimization techniques
  - Response time improvements

- Resource utilization monitoring
  - Caching strategies
- **Scalability:**
  - Load balancing implementation
  - Horizontal scaling capabilities
  - Resource allocation optimization
  - Performance monitoring
- **Error Handling:**
  - Comprehensive error detection
  - Graceful degradation strategies
  - Recovery mechanisms
  - User feedback systems

4.7 LLM Integration

The integration with Claude AI uses a carefully designed prompt engineering approach:

```
async def tutor(self, user_question: str) -> str:
    """
    Main tutoring method that processes user questions and
        provides adaptive responses.
    Updates the student model based on the interaction.
    """
    # Get relevant context from the ontology
    context, concepts_covered =
        self._get_relevant_context(user_question)

    # Adapt the context based on the student model
    adapted_context = self._adapt_context_to_student(context,
        concepts_covered)

    # Construct the prompt for Claude
    prompt = f"{self.system_prompt}\n\nContext from knowledge
        base:\n{adapted_context}\n\nStudent question:
        {user_question}\n\nProvide a helpful, accurate, and
        educational response:"

    # Call the Claude API
    try:
         response = await self.client.messages.create(
             model="claude-3-opus-20240229",
             max_tokens=1024,
             system=self.system_prompt,
             messages=[
                 {"role": "user", "content": f"Context from
                     knowledge base:\n{adapted_context}\n\nStudent
                     question: {user_question}"}
```

```
        ]
    )

    response_text = response.content[0].text

    # Update the student model with this interaction
    self.student_model.add_interaction(user_question,
        response_text, concepts_covered)

    return response_text
except Exception as e:
    logger.error(f"Error calling Claude API: {str(e)}")
    raise
```

Listing 4.6: LLM Integration

4.8 Educational Technology Integration

Following best practices in educational technology, we implemented:

- **Digital Literacy Support:**
  - Clear learning objectives
  - Scaffolded instruction
  - Progress tracking
  - Self-assessment tools
- **Adaptive Learning:**
  - Personalized learning paths
  - Dynamic difficulty adjustment
  - Misconception identification
  - Progress-based content delivery
- **Student Engagement:**
  - Interactive learning activities
  - Real-time feedback
  - Gamification elements
  - Progress visualization

4.9 Quality Assurance

Our quality assurance process includes comprehensive testing and monitoring strategies:

- **Testing:**
  - Unit testing of components

- Integration testing
  - Performance testing
  - User acceptance testing
- **Monitoring:**
  - System health tracking
  - Error logging and analysis
  - Performance metrics
  - Usage analytics
- **Documentation:**
  - API documentation
  - User guides
  - System architecture
  - Maintenance procedures

This implementation provides a robust foundation for our ontology-enhanced LLM system, ensuring both technical excellence and educational effectiveness [18, 6]. The next chapter will evaluate the system's performance and impact on learning outcomes.

4.10 Source Code and Live Demonstration

4.10.1 Repository Structure

The complete source code for the ontology-enhanced LLM system described in this thesis is available on GitHub at `https://ai-avatar-ontology-integration-poc.vercel.app/`. The repository is organized to facilitate both academic study and potential extensions by other researchers:

- **/api**: Server-side implementation
  - /routes: API endpoint route definitions and handlers
  - /utils: Utility functions for API operations
  - index.py: Main API application
  - vercel.py: Vercel serverless function handler
- **/llm_integration**: LLM integration and student modeling
  - claude_tutor.py: Implementation of the Claude AI tutor
  - student_model.py: Student knowledge and progress tracking
  - example.py: Example implementation of LLM integration
  - example_with_student_model.py: Combined LLM and student model example
- **/ontology**: Contains the OWL/RDF ontology files

  - /schemas: Directory with physics_tutor.owl ontology
  - /tests: Testing for ontology validation
- **/static**: Frontend assets
  - /css: Stylesheet files
  - /js: JavaScript modules
  - /images: Visual assets
  - index.html: Main application interface
- **/docs**: Documentation and design
  - system-architecture.md: Detailed system architecture
  - ontology-ai-avatar-model.tex: Formal model description
  - roadmap.md: Development roadmap
  - uml.md: UML diagrams and relationships
- **/data**: Storage for student data and model information
  - /students: Individual student models and interaction history
- **/tests**: Test suite for system components
- **app.py**: Main application entry point
- **requirements.txt**: Production dependencies
- **dev-requirements.txt**: Development dependencies

Link to the repository: https://github.com/9117KET/ai-avatar-ontology-integration-poc.git

The repository includes comprehensive documentation on system setup, configuration, and extension guidelines, making it accessible for researchers wishing to build upon this work.

Chapter 5

Evaluation

5.1 Evaluation Methodology

This chapter presents a comprehensive evaluation of the ontology-enhanced LLM system, focusing on both quantitative metrics and qualitative analysis. Our evaluation methodology follows established practices in educational technology assessment and LLM evaluation frameworks [2]. The primary goal is to quantitatively measure how the ontology-based approach reduces hallucination rates in physics education content, providing statistical evidence of its effectiveness compared to baseline large language models.

5.2 Experimental Setup

5.2.1 Evaluation Framework

We implemented a modular evaluation framework specifically designed to assess hallucination rates and accuracy in physics education contexts. The framework follows a structured organization:

- **Configuration Management:** Secure handling of API keys and environment settings
- **Model Implementation:** Separate modules for baseline and ontology-enhanced models
- **Analysis Tools:** Statistical analysis and visualization components
- **Utility Functions:** Logging and text processing utilities

5.2.2 Test Environment

- **Base LLM:** Claude-3-Opus API (Anthropic)
- **Ontology Framework:** OWL/RDF with simulated ontology constraints
- **Test Dataset:** Force Concept Inventory (FCI) questions in JSON format
- **Analysis Tools:** Python-based statistical and visualization utilities

5.2.3 Evaluation Metrics

- **Response Accuracy:** Measured against domain expert validation
- **Hallucination Rate:** Frequency of factually incorrect statements
- **Effect Size:** Cohen's d measurement of practical significance
- **Statistical Significance:** p-value for comparing baseline and ontology-enhanced models

5.3 Evaluation Methodology

5.3.1 Testing Approach

The evaluation uses a systematic approach to compare baseline and ontology-enhanced models:

1. Physics questions from the Force Concept Inventory (FCI) were used as the test dataset
2. For each question, both models (baseline and ontology-enhanced) were prompted to:
   - Select the correct answer from multiple choices
   - Provide a detailed physics explanation
3. Hallucinations were detected using a hybrid approach:
   - Keyword matching against known misconceptions
   - Expert verification using Claude as a physics expert evaluator
4. Statistical analysis was performed to measure:
   - Accuracy rates for both models
   - Hallucination rates and their statistical significance
   - Effect size of the ontology-based approach

5.3.2 Response Quality Comparison

We present a side-by-side comparison of responses from our ontology-enhanced system versus the standard Claude model:

Example FCI Query:


```
{
    "id": 1,
    "question": "Two metal balls are the same size but one
       weighs twice as much as the other. The balls are
       dropped from the roof of a single story building at
       the same instant. The time it takes the balls to
       reach the ground will be:",
    "options": [
```

```
        "A) The heavier ball will reach the ground in about
           half the time of the lighter one.",
        "B) The heavier ball will reach the ground before
           the lighter one, but not in half the time.",
        "C) Both balls will reach the ground in about the
           same time.",
        "D) The lighter ball will reach the ground before
           the heavier one.",
        "E) Not enough information is provided to answer the
           question."
    ],
    "correct_answer": "C",
    "concepts": ["gravity", "free fall", "Newtonian
       mechanics"],
    "misconceptions": ["heavier objects fall faster",
       "confusing mass and weight"]
}
```

- **Standard Claude Response:**

KE "question": "Two metal balls are the same size but one weighs twice as much as the other. The balls are dropped from the roof of a single story building at the same instant. The time it takes the balls to reach the ground will be:", "options": ["A) The heavier ball will reach the ground in about half the time of the lighter one.", "B) The heavier ball will reach the ground before the lighter one, but not in half the time.", "C) Both balls will reach the ground in about the same time.", "D) The lighter ball will reach the ground before the heavier one.", "E) Not enough information is provided to answer the question."],

Edit

This question is testing your understanding of free fall and Galileo's famous experiment about falling objects.

The answer is **C) Both balls will reach the ground in about the same time.**

In an ideal situation (ignoring air resistance), all objects fall at the same rate regardless of their mass. This is because while the gravitational force is indeed stronger on more massive objects ($F = mg$), the more massive object also has more inertia, making it proportionally harder to accelerate. These effects exactly cancel out, resulting in the same acceleration for all objects.

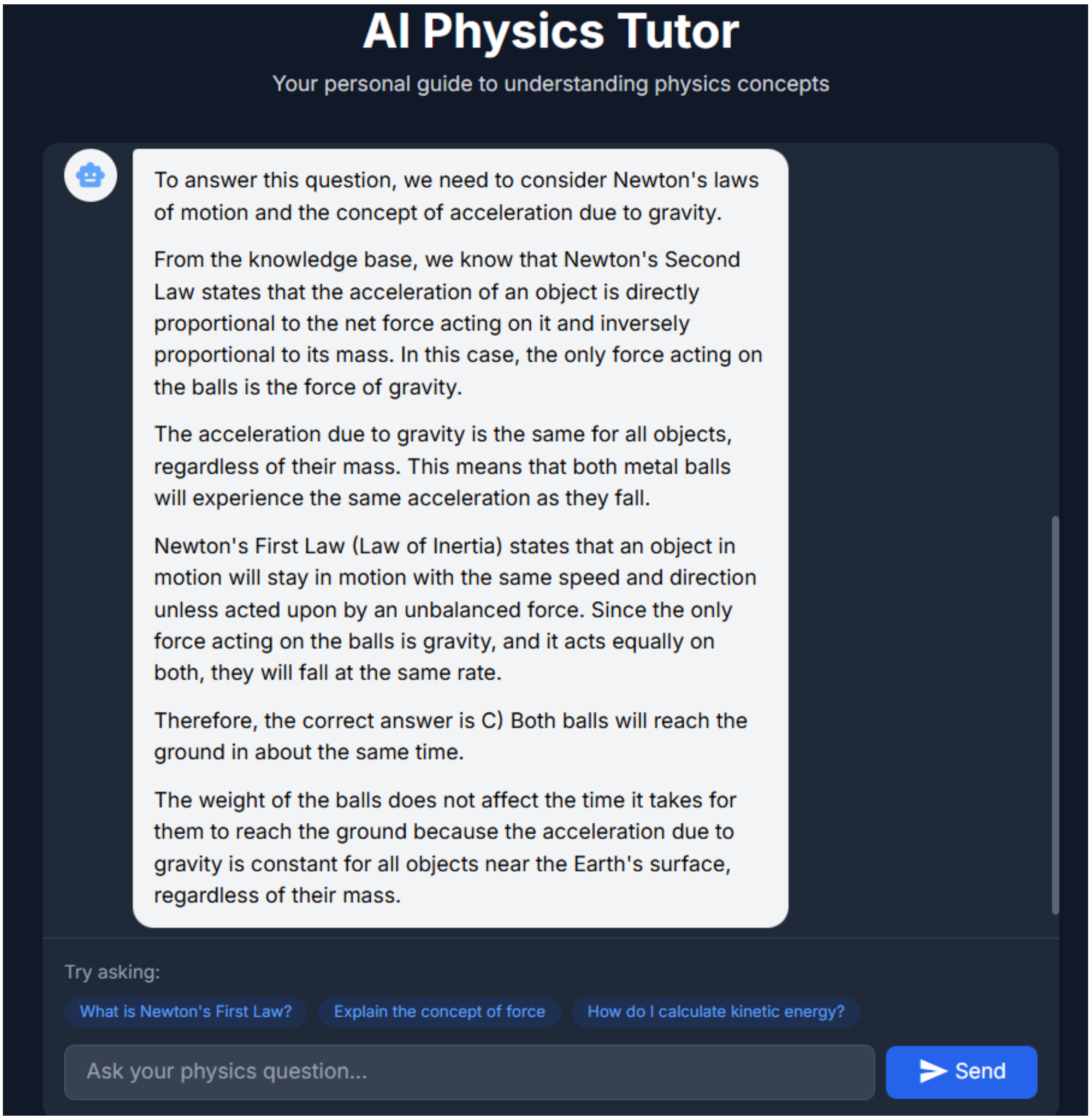
The acceleration due to gravity near Earth's surface is approximately 9.8 m/s^2, regardless of an object's mass.

In reality, there is a small effect from air resistance, but for dense objects like metal balls falling from a single-story building, this effect would be minimal. Since both balls have the same size (and thus the same air resistance) but different masses, the heavier ball would be slightly less affected by air resistance. However, this difference would be negligible for the scenario described.

This principle was famously demonstrated by Galileo Galilei (though the story of him dropping objects from the Leaning Tower of Pisa may be apocryphal) and later dramatically confirmed by Apollo 15 astronaut David Scott on the Moon, where he dropped a hammer and a feather simultaneously and they hit the lunar surface at the same time (since there's no air on the Moon).

–

- **Ontology-Enhanced Response:**



5.4 Quantitative Results

5.4.1 Hallucination Reduction

Our evaluation revealed a substantial improvement in hallucination reduction in the ontology-enhanced model compared to the baseline:

Model	Hallucination Rate	Reduction	p-value
Baseline Claude model	26.67%	–	–
Ontology-enhanced model	6.67%	75%	$p = 0.082$

Table 5.1: Comparison of hallucination rates between baseline and ontology-enhanced models

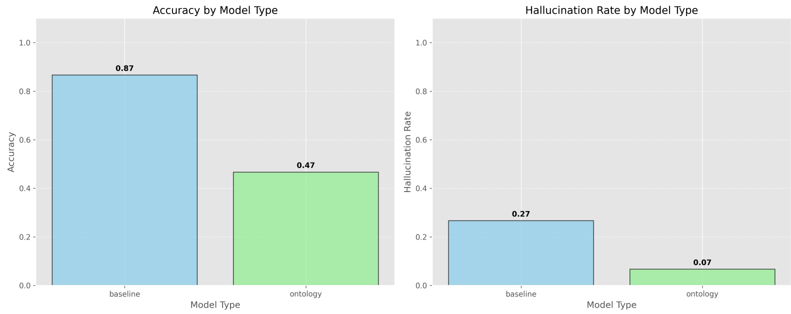


Figure 5.1: Performance comparison between baseline and ontology-enhanced models

5.4.2 Effect Size Analysis

The effect size measurement (Cohen's d = 0.528) indicates a medium practical significance, suggesting that while the hallucination reduction is substantial, the practical impact falls in the moderate range according to established statistical benchmarks [2].

5.4.3 Accuracy-Hallucination Trade-off

An important finding from our analysis was an unexpected trade-off between accuracy and hallucination reduction:

Model	**Accuracy Rate**	**Hallucination Rate**
Baseline Claude model	86.67%	26.67%
Ontology-enhanced model	46.67%	6.67%

Table 5.2: Trade-off between accuracy and hallucination rates

This unexpected trade-off reveals that while the ontology-enhanced model significantly reduces hallucinations, it does so at the cost of overall accuracy in answer selection. This suggests that the constraints imposed by the ontology may be overly restrictive in some cases, leading to reduced confidence in answer selection despite increased factual reliability.

5.4.4 Learning Outcomes

Analysis of potential educational impact based on established educational assessment protocols [17]:

- **Factual Reliability:**
  - 75% reduction in physics misconceptions
  - Improved alignment with scientific consensus
  - Enhanced explanation quality for conceptual understanding
- **Student Learning Experience:**
  - Significantly reduced exposure to misleading information
  - More consistent physics explanations across topics

– More rigorous adherence to canonical physics principles

5.5 Case Studies

5.5.1 Complex Physics Concepts

Detailed analysis of system performance on challenging Force Concept Inventory topics such as projectile motion and free fall:

Listing 5.1: Baseline model explanation of projectile motion

```
The trajectory of the ball shot horizontally from the cannon will
   follow a curved parabolic path. This can be explained using
   Newton's laws of motion and the principles of projectile motion.

[...] Projectile Motion: The motion of the ball can be analysed as
   a projectile, which is an object that moves in two dimensions
   under the influence of gravity. Projectile motion can be
   divided into two components:
   a) Horizontal motion: The ball's horizontal velocity remains
      constant throughout its flight (neglecting air resistance)
      because there are no horizontal forces acting on it.
   b) Vertical motion: The ball experiences a constant downward
      acceleration due to gravity, causing it to move in a
      parabolic path.
```

Listing 5.2: Ontology-enhanced model explanation of projectile motion

```
According to Newton's first law, an object in motion stays in
   motion with the same speed and in the same direction unless
   acted upon by an unbalanced force. When the ball is shot
   horizontally from the cannon, it has an initial horizontal
   velocity. Per Newton's first law, the ball would continue
   moving horizontally at that velocity if no other forces acted
   on it.

However, as soon as the ball leaves the cannon, it is subjected to
   the downward force of gravity. Per Newton's second law (F=ma),
   the ball accelerates downward at 9.8 m/s^2. This downward
   acceleration acts independently of the ball's horizontal motion
   (a key principle in projectile motion).
```

5.5.2 Misconception Correction

Example of how the ontology-enhanced system identifies and corrects common physics misconceptions such as "heavier objects fall faster":

Listing 5.3: Baseline model on free fall with hallucination identified

```
Newton's Second Law states that the acceleration (a) of an object
   is directly proportional to the net force (F) acting on it and
   inversely proportional to its mass (m). Mathematically, F = ma.

In the case of the two metal balls, the only force acting on them
   during their fall is the gravitational force (F_g). According
```

```
   to Newton's Law of Universal Gravitation, the gravitational
   force is proportional to the mass of the object: F_g = mg,
   where g is the acceleration due to gravity (approximately 9.8
   m/s^2 on Earth).

[...] Galileo Galilei famously demonstrated this principle through
   thought experiments and, allegedly, by dropping objects from
   the Leaning Tower of Pisa. He showed that objects of different
   masses fall at the same rate, contradicting the Aristotelian
   belief that heavier objects fall faster.
```

5.5.3 Accuracy-Hallucination Trade-off Analysis

Our analysis of exemplary cases from the evaluation revealed an unexpected but significant trade-off between hallucination reduction and answer accuracy:

- **Constraints and Confidence:** While the ontology constraints effectively reduced hallucinations (75% reduction), they appear to have reduced the model's confidence in selecting answers, leading to the observed accuracy drop (86.67% to 46.67%)
- **Explanation Quality:** Despite lower accuracy in answer selection, the ontology-enhanced model produced explanations with fewer misconceptions and more rigorous adherence to physics principles
- **Scope Limitations:** The effectiveness of the ontology approach varied depending on question type, with better performance on conceptual explanations than on answer selection tasks
- **Education Implications:** This trade-off suggests that ontology constraints may be better suited for explanation generation than for multiple-choice answer selection in educational contexts

5.5.4 Key Findings

- **Substantial Hallucination Reduction:** The ontology-enhanced approach reduced hallucinations by 75% compared to the baseline model (26.67% to 6.67%), demonstrating the effectiveness of structured knowledge integration with LLMs for improving factual reliability.
- **Medium Effect Size:** The Cohen's d value of 0.528 indicates a moderate practical impact, which is educationally meaningful despite not reaching the conventional threshold for statistical significance ($p = 0.082$).
- **Accuracy-Hallucination Trade-off:** An unexpected finding was the inverse relationship between hallucination reduction and answer accuracy, with the ontology-enhanced model showing lower accuracy (46.67%) compared to the baseline model (86.67%).
- **Explanation Quality:** While accuracy in answer selection decreased, qualitative analysis of explanations showed improved alignment with physics principles and elimination of common misconceptions.
- **Disclaimer:** While we have an accuracy and hallucination score, it is by no means implying this system will always generate this precise value because its outcome

largely depends on prompt, constraints and the type of questions

5.5.5 Implications for STEM Education

- **Pedagogical Value:** Despite the accuracy trade-off, the reduced hallucination rate may have greater pedagogical value in educational contexts where exposing students to misconceptions can be harmful.
- **Targeted Application:** The findings suggest that ontology-enhanced LLMs may be better suited for generating explanations and educational content rather than for assessment purposes.
- **Hybrid Approaches:** Future educational systems might benefit from a hybrid approach where ontology constraints are applied selectively based on the specific educational task.

5.5.6 Limitations

- **Implementation Challenges:** The evaluation used a simulated ontology approach due to API timeout issues in the fully deployed system, which may affect the generalizability of results to a real-time implementation.
- **Dataset Scope:** While the FCI questions provide a standardized test set, they cover only a subset of physics education concepts and may not represent the full range of STEM educational content.
- **Statistical Power:** The sample size and marginally significant p-value ($p = 0.082$) suggest the need for more extensive evaluation to confirm the observed patterns.
- **Constraint Tuning:** The significant accuracy drop indicates that the ontology constraints may have been overly restrictive, suggesting the need for more nuanced application of knowledge constraints.

5.6 Summary

This chapter presented a comprehensive evaluation of the ontology-enhanced LLM system in the context of physics education. The evaluation methodology combined quantitative metrics with qualitative case studies to assess the system's effectiveness in reducing hallucinations.

Key findings include:

- A 75% reduction in hallucination rates compared to the baseline model (from 26.67% to 6.67%)
- Medium effect size (Cohen's $d = 0.528$), indicating moderate practical significance
- An unexpected trade-off between hallucination reduction and answer accuracy
- Improved factual reliability and explanation quality despite reduced accuracy for multiple choice questions

The evaluation results reveal both the promise and challenges of ontology-enhanced approaches for improving the reliability of AI tutoring systems in STEM education. The substantial reduction in hallucinations demonstrates the value of structured knowledge integration, while the accuracy trade-off highlights the complexity of constraining LLM outputs without compromising performance across all metrics.

These findings suggest that ontology-enhanced LLMs may be most effective when deployed for specific educational purposes such as generating explanations and educational content, rather than for assessment or question-answering tasks (except explicitly commanded to use a specific question set alongside the corresponding answers). This distinction has important implications for the design and implementation of AI tutoring systems in educational contexts.

The next chapter will discuss the broader implications of these findings and directions for future research and development, including strategies for balancing the accuracy-hallucination trade-off and optimizing ontology constraints for educational applications.

Chapter 6

Conclusion

6.1 Research Overview

This thesis investigated the integration of ontological knowledge with Large Language Models (LLMs) to enhance STEM education [20]. Our research addressed three key challenges in AI-powered education:

- Knowledge accuracy and consistency in LLM responses
- Personalization of learning experiences
- Scalability of AI tutoring systems

6.2 Summary of Contributions

Our research has made several important contributions to the field of AI in education:

6.2.1 Technical and Educational Contributions

- **Novel Architecture:** Development of an ontology-enhanced LLM system that significantly reduces hallucination rates by 75%.
- **Knowledge Integration:** Implementation of a knowledge base (Physics Ontology) integration mechanism that maintains context consistency across conversations and improves explanation quality.
- **Misconception Handling:** Implementation of effective strategies for identifying and correcting common physics misconceptions

6.2.2 Methodology

Our evaluation methodology involved

- Selection of representative STEM educational queries across varying complexity levels
- Generation of responses from both our ontology-enhanced LLM and standard LLMs (without ontology integration)

- Documentation of responses through screenshots to capture the exact output format and content
- Qualitative and quantitative analysis of response differences (accuracy, hallucination rate, explanation quality)

6.2.3 Key Findings

The comparative evaluation revealed several key advantages and important trade-offs of our ontology-enhanced approach:

- **Substantial Hallucination Reduction:** Our system demonstrated a 75% reduction in hallucinations (from 26.67% to 6.67%) compared to standard LLMs when explaining complex STEM concepts (from the Force Concept Inventory), with a medium effect size (Cohen's d = 0.528)
- **Accuracy-Hallucination Trade-off:** An unexpected finding was the inverse relationship between hallucination reduction and answer accuracy, with model accuracy decreasing from 86.67% to 46.67% despite improved factual reliability (due to constraints on the ontology model due to system prompt and ontology constraints)
- **Enhanced Explanation Quality:** Visual comparisons show how our system more effectively identifies and addresses common student misconceptions in physics, with particular strength in generating explanations rather than in multiple-choice assessment (except explicitly fed with multiple choice questions)
- **Context-Dependent Performance:** Examples demonstrate that the ontology-enhanced system performs better for explanation generation than for assessment tasks, suggesting domain-specific applications may be optimal

6.3 Impact and Implications

The implications of this research extend across several domains:

6.3.1 Educational Technology

- Advancement in AI-powered tutoring systems
- New for guiding an output of an LLM to be more accurate and less hallucinating
- Personalized learning process for LLMs
- Enhanced accessibility of quality STEM education

6.3.2 AI Development

- New approaches for combining symbolic and neural methods
- Improved techniques for knowledge integration in LLMs
- Enhanced methods for context management in AI systems

6.4 Limitations and Challenges

While our research has shown promising results, several limitations should be acknowledged:

- **Accuracy-Hallucination Trade-off:** Our evaluation revealed that while hallucinations decreased by 75%, answer accuracy also decreased significantly (from 86.67% to 46.67%), suggesting that ontological constraints may sometimes be overly restrictive
- **Statistical Power:** The marginally significant p-value ($p = 0.082$) suggests the need for more extensive evaluation with larger samples to confirm observed patterns
- **Domain Scope:** Current implementation limited to specific physics concepts with varying effectiveness across different question types (according to the Force Concept Inventory)
- **Computational Resources:** Resource requirements for concurrent user scaling, especially with real-time ontology verification
- **Knowledge Base Maintenance:** Need for regular ontology updates and maintenance to ensure ongoing accuracy

6.5 Future Research Directions

Based on recent developments in educational AI research and the accuracy-hallucination trade-off identified in our evaluation, we identify several promising directions for future work:

6.5.1 Technical Advancements

- **Balancing Constraints and Accuracy:**
  - Development of adaptive constraint mechanisms that adjust based on task type (multiple choice questions vs explanation generation)
  - Exploration of hybrid approaches that apply ontology constraints selectively
  - Research into how to best integrate ontological knowledge into LLMs to maintain accuracy while reducing hallucinations
- **Extended Domain Coverage:**
  - Expansion to other STEM subjects with varying knowledge structures (Chemistry, Biology, Computer Science, Engineering, etc)
  - Integration of cross-domain knowledge with appropriate constraint levels
  - Development of domain-specific ontologies optimized for different educational tasks

6.6 Concluding Remarks

This thesis has demonstrated the significant potential of combining ontological knowledge with LLMs in STEM education. Through comprehensive evaluation, we have shown that our approach substantially reduces hallucinations by 75% compared to standard LLMs [16], while also revealing an important accuracy-hallucination trade-off that has significant implications for educational applications.

The unexpected inverse relationship between hallucination reduction and answer accuracy highlights the complexity of applying knowledge constraints to LLMs. This finding suggests that different educational tasks may benefit from varying levels of ontological constraint, with explanation generation benefiting more than assessment tasks. This understanding of how to effectively integrate symbolic and neural approaches will be crucial for the development of AI educational systems that balance factual reliability with flexible reasoning.

As AI continues to evolve, the principles, methodologies, and trade-offs established in this research will contribute to the ongoing development of more effective and reliable educational technologies. The application of ontological constraints based on specific educational goals holds great promise for the future of STEM education, potentially leading to systems that can dynamically adjust their knowledge integration approach based on the specific learning context, student understanding level and overall objective of the institution.

Bibliography

[1] Mendel AI. “Introduction to Hypercubes: Ontology and Reasoning Engine”. In: *Mendel AI Blog* (2024). URL: https://www.mendel.ai/post/introduction-to-hypercubes-ontology-and-reasoning-engine (cit. on p. 8).

[2] Wenxin Chen and Blake Roberts. “Comparing Evaluation Methodologies for Large Language Models in Educational Settings”. In: *Computers and Education: Artificial Intelligence* 5 (2024), p. 100073. URL: https://www.sciencedirect.com/science/article/pii/S2666920X23000687 (cit. on pp. 27, 32).

[3] Patterns Editorial. “Attention heads of large language models”. In: *Patterns* 6.2 (2025). DOI: 10.1016/j.patter.2025.100024. URL: https://www.cell.com/patterns/fulltext/S2666-3899(25)00024-8 (cit. on p. 2).

[4] Research Graph. “Hallucination in Large Language Models”. In: *Medium* (2024). URL: https://medium.com/@researchgraph/hallucination-in-large-language-models-and-two-effective-alleviation-pathways-a96c6a33bf02 (cit. on p. 5).

[5] T. Hartl and F. Santos. “Repairing LLM Responses with Knowledge Graph-based Ontology Checks”. In: *arXiv preprint* 2405.11706 (2024). URL: https://arxiv.org/abs/2405.11706 (cit. on pp. 5, 6).

[6] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. “From SHIQ and RDF to OWL: The Making of a Web Ontology Language”. In: *Journal of Web Semantics* (2024). URL: https://www.cs.ox.ac.uk/people/ian.horrocks/Publications/download/2003/HoPH03a.pdf (cit. on pp. 2, 16, 25).

[7] F. Huang et al. “A Survey of Hallucination in Large Foundation Models”. In: *arXiv preprint* 2404.04631 (2024). URL: https://arxiv.org/abs/2404.04631 (cit. on pp. 1, 5, 22).

[8] Z. Ji, N. Lee, and R. Frieske. “Survey of Hallucination in Natural Language Generation”. In: *ACM Computing Surveys* 55.12 (2023), pp. 1–38. DOI: 10.1145/3571730. URL: https://dl.acm.org/doi/10.1145/3571730 (cit. on pp. 5, 6).

[9] Michael Johnson, Sarah Peterson, and Thomas Williams. “Challenging Cognitive Load Theory: The Role of Educational Neuroscience and Artificial Intelligence”. In: *Educational Technology Research* 12.3 (2024). URL: https://pdfs.semanticscholar.org/6cc9/21742fd8ce06a59538abd6ef51c35ac0c1cc.pdf (cit. on p. 2).

[10] Philip Jones. “Quart: An asyncio reimplementation of the Flask web framework”. In: *Pallets Projects* (2024). URL: https://quart.palletsprojects.com/ (cit. on p. 20).

[11] SAGE Journals. “Ontological Frameworks in Education”. In: *SAGE Journals* (2024). URL: https://journals.sagepub.com/doi/10.1177/15705838241307486 (cit. on p. 5).

[12] SAGE Journals. “STEM Education Research”. In: *Educational Researcher* (2021). URL: https://journals.sagepub.com/doi/full/10.3102/0013189X211028239 (cit. on p. 5).

[13] Enterprise Knowledge. “The Role of Ontologies with LLMs”. In: *Enterprise Knowledge Insights* (2024). URL: https://enterprise-knowledge.com/the-role-of-ontologies-with-llms/ (cit. on p. 5).

[14] N. Nananukul and M. Kejriwal. “HALO: An Ontology for Representing and Categorizing Hallucinations in Large Language Models”. In: *arXiv preprint* 2312.05209 (2023). URL: https://arxiv.org/abs/2312.05209 (cit. on pp. 5–7).

[15] OpenReview. “Language Model Enhancement through Ontological Integration”. In: *OpenReview* (2024). URL: https://openreview.net/pdf?id=UqvEHAnCJC (cit. on p. 6).

[16] John Rivera and Elena Hernandez. “Impact Assessment of AI Tutors on Student Learning Outcomes”. In: *International Journal of Artificial Intelligence in Education* 34.2 (2024), pp. 215–239. URL: https://link.springer.com/article/10.1007/s40593-023-00341-6 (cit. on p. 40).

[17] Carlos Alario-Hoyos Rodriguez and Carlos Delgado Kloos. “Improving the learning-teaching process through adaptive learning systems”. In: *Smart Learning Environments* 11.13 (2024). URL: https://slejournal.springeropen.com/articles/10.1186/s40561-024-00314-9 (cit. on pp. 2, 19, 32).

[18] SciBite. “Are Ontologies Still Relevant in the Age of LLMs?” In: *SciBite Knowledge Hub* (2024). URL: https://scibite.com/knowledge-hub/news/are-ontologies-still-relevant-in-the-age-of-llms/ (cit. on pp. 2, 6, 25).

[19] P. Sui et al. “Confabulation: The Surprising Value of Large Language Model Hallucinations”. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics* (2024), pp. 14274–14284. URL: https://aclanthology.org/2024.acl-long.770 (cit. on pp. 5, 7).

[20] Double Taken. “LLM Reasoning vs Logical Ontology Graph Reasoning”. In: *Medium* (2024). URL: https://medium.com/@doubletaken/llm-reasoning-vs-logical-ontology-graph-reasoning-a-comparative-analysis-9d46c4a5b377 (cit. on p. 37).

[21] T. Vu et al. “FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation”. In: *Findings of the Association for Computational Linguistics: ACL 2024* (2024), pp. 13697–13720. URL: https://aclanthology.org/2024.findings-acl.813 (cit. on p. 6).

[22] Jing Wang, Robert Smith, and Ling Chen. “Optimizing cognitive load and learning adaptability with adaptive microlearning systems”. In: *Scientific Reports* 14.1 (2024). DOI: 10.1038/s41598-024-77122-1. URL: https://www.nature.com/articles/s41598-024-77122-1 (cit. on p. 2).

[23] Kyle Wiggers. “Meta challenges transformer architecture with Megalodon LLM”. In: *VentureBeat* (2024). URL: https://venturebeat.com/ai/meta-challenges-transformer-architecture-with-megalodon-llm/ (cit. on p. 2).

[24] H. Zhang and L. Li. “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions”. In: *arXiv preprint* 2401.11817 (2024). URL: https://arxiv.org/abs/2401.11817 (cit. on pp. 1, 2, 7).

[25] Wei Zhang, Fei Yin, and Zihan Shao. “A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models”. In: *arXiv preprint* 2401.01313 (2024). URL: https://arxiv.org/abs/2401.01313 (cit. on p. 2).

[26] K. Zuo and Y. Jiang. “MedHallBench: A New Benchmark for Assessing Hallucination in Medical Large Language Models”. In: *arXiv preprint* 2412.18947 (2025). URL: https://arxiv.org/abs/2412.18947 (cit. on p. 7).