Elevating Education with AI: Augmenting the Understanding of Physics through Topic Prediction, Three-Dimensional Visualisation, and Dynamic Video Aids

by

Md. Ajmain Aosaf Anan
20101388
Sayad Md. Prio
20101356
A. K. Mahamudun Nabi
24141282
Rifat Ara Islam
18201091
Shamazda Islam
19301225

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Ajmain Aosaf Anan
20101388

Sayad Md. Prio
20101356

A. K. Mahamudun Nabi
24141282

Rifat Ara Islam
18201091

Shamazda Islam
19301225

Approval

The thesis titled ”Elevating Education with AI: Augmenting the Understanding of Physics through Topic Prediction, Three-Dimensional Visualisation, and Dynamic Video Aids” submitted by

1. Md. Ajmain Aosaf Anan (20101388)
2. Sayad Md. Prio (20101356)
3. A. K. Mahamudun Nabi (24141282)
4. Rifat Ara Islam (18201091)
5. Shamazda Islam (19301225)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 03, 2025.

Examining Committee:
Signature of Supervisor:
Thesis Supervisor:
(Member)

____
Md. Sabbir Ahmed
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Chair)

____
Md. Golam Rabiul Alam
Professor Thesis Coordinator
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

In an era defined by the rapid evolution of science and technology, the integration of Artificial Intelligence (AI) into education has emerged as a transformative force. This study explores the revolutionary potential of AI in reshaping physics education by developing an AI-powered learning platform tailored specifically to the physics domain. The proposed platform combines advanced AI methodologies, including Natural Language Processing (NLP), Deep Learning, Generative Adversarial Networks (GANs), Computer Vision, and Machine Learning algorithms, to enhance the learning and problem-solving experience for both students and teachers.

The platform offers a dynamic and interactive environment where users can efficiently solve complex physics problems while visualizing them in an intuitive manner. By leveraging user-generated data, the model creates personalized 3D visualizations and motion videos that simulate the given scenarios, enabling users to grasp abstract concepts and problem-solving strategies more effectively. Furthermore, the study delves into the rationale behind the selection of specific AI models and algorithms, the type and significance of the collected data, the comparative analysis with existing AI-based education tools, and the potential impact on the target user base.

This research not only bridges the gap between theoretical physics and practical understanding but also provides an alternative approach to traditional learning methods. By facilitating the visualization of complex problems and offering innovative solutions, the proposed model aims to empower educators and learners alike. Ultimately, this study underscores the transformative potential of AI in fostering deeper comprehension, engagement, and creativity in physics education.

Keywords: Artificial Intelligence; Physics Education; Ai-Powered Learning Platform; Natural Language Processing(NLP); Deep Learning; Generative Adversarial Networks(GANs); Machine Learning; 3D Visualization; Motion Video Simulation; Computer Vision; STEM Education.

Acknowledgement

Firstly, all praise to the Almighty Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Mr. Sabbir Ahmed sir for his kind support and advice in our work. He helped us whenever we needed help. Without his unwavering support and guidance we may not complete this thesis.
Furthermore, we are also thankful to our parents for their constant support in this research in various ways.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Artificial Intelligence has notably changed from the beginning of the digital era. It has evolved significantly from its initial crude programmes in the 1950s to its current complex algorithms. When artificial intelligence started off, it was just a collection of basic patterns and principles. Later on, over the years it has grown into one of the most crucial technological fields. When Alan Turing first published his "Computer Machinery and Intelligence" [1], it proposed a test of machine intelligence.

However, artificial intelligence has grown to become more sophisticated and intricate over the years. It is getting more and more powerful with every new development, reaching previously unthinkable levels of performance. Now, it seems that in a couple of years artificial intelligence will have the upper hand in solving basic tasks in our daily lives. Though, it will bring huge changes in every sector, it will be more efficient, trustworthy and crucial for every little thing. Like, in today's world ChatGPT has become one of the most important and valuable AI. Not only does it succeed in completing various tasks but also it has proven itself how it became useful for everyone. Although, there can still be some areas left where an effective AI can bring a major change.

Here, we are introducing an AI which proposes the idea of visualisation of basic topics of Physics. From this AI, the students as well as teachers can efficiently study on the topics of this particular subject. Moreover, they can solve any problems regarding the basic topics of Physics easily with the help of this artificial intelligence. It will also be easy for the students to study Physics. Mostly, the user has to enter a problem where it will be analysed and processed then, the results from analysis will generate some images and 3D models related to the problem's scenario. By diffusing the images with one another sequentially, it will create a dynamic motion video where the questions and topics can come to life. At the same time, it will calculate the result of any problem with the help of formulas that are stored in the library. Our, first and foremost target is to generate the scenario, image and result with utmost accuracy.

In the broader context, AI's role in our technological development era has been pivotal, evolving from smartphones to faster processing chipsets, robotics, and now functioning as a daily helper. It has become widely accepted as a guide, learning platform, and essential support in various aspects of human life. The envisioned AI model aims to contribute as a valuable resource, particularly assisting students facing challenges in problem-solving and understanding complex topics. As we navigate the evolving landscape of AI, this proposed model represents a promising stride towards more accurate and accessible learning experiences in the realm of Physics and beyond.

1.2 Research Problem

Students often face problems while understanding physics concepts. Moreover, it's even harder for someone to understand what is being explained in the lecture. This occurs because of two major problems which are not having clear basics of physics and their ability to use critical thinking. [2] Students learn from their school or college lectures with 2D drawings and diagrams but still, many of them fail in critical visualisation. This happens because the question's language or the explanation may not be easy to visualize or the question is itself tough to visualize. In an experiment, the same question was given to two groups: one was using active voice structure, and the other one was using simple terms explaining with 11 steps to visualize the question. Students did a promising result on the second question. [2] but students won't always get the easy terms questions. So students need to understand to visualize even in tough explanations. This challenge is particularly evident in the physics classroom when students come across challenging problems that call for visualisation to solve. [2] For this kind of reason, many students perform badly in physics in their earlier stages even in further complex problems. For this reason, a student needs a specialist either a teacher or an AI tool that can help the student at any time. Which is the physics simulator that will help a student visualize questions with moving animation and even with the calculation.

Revania Risang Ayu's thesis, titled "The Need Analysis of the Development of Animation Video-Based Learning Media on Physics Education Students," presents a comprehensive exploration of the challenges faced by students in School Physics courses and proposes the development of animated learning videos as a solution. The research, conducted as exploratory descriptive research, involved Physics Education students from the Faculty of Teacher Training and Education at Siliwangi University Batch 2021. The study effectively identifies the difficulties students encounter in learning physics, with a particular emphasis on Thermodynamics. Through observations and questionnaires, Ayu emphasises the need for more interesting and effective learning resources, recommending animated movies as a viable alternative. The investigation sheds light on students' preferences for interactive, visually appealing, and well-structured animated videos. Ayu's thesis finishes with deliberate recommendations for the development of learning media, focusing on crucial issues such as colour selection, typeface legibility, and content presentation consistency. Overall, this thesis contributes significantly to the enhancement of physics education by addressing identified gaps using new and engaging learning media.[3].

Popi Purwanti's thesis paper "The Effect of Learning Approaches and Learning Creativity on the Ability to Solve Physics Problems" investigates the impact of scientific learning approaches against conventional learning techniques on students' physics problem-solving abilities. It also looks into the disparities in problem-solving skills between students with high and low creativity, in order to determine the interaction effects of learning techniques and learning creativity. The study was conducted using an experimental technique in SMA PB Soedirman 2 Bekasi, including both experimental and control groups. Data collection included a 10-point essay test and a 30-point questionnaire for assessing learning creativity. The analysis involved descriptive statistics and a two-way ANOVA to test the research hypotheses. [4].

Based on the paper [5] the authors made an approach with the objective of making 3D animated videos via the 3D4Medical Complete Anatomy app to enhance the understanding of cranial nerve structures, functions, and pathologies. The authors described the possible methodology and the effectiveness of integrating 3D animated videos into a flipped classroom paradigm to enhance the understanding of cranial nerve pathways, functions, and pathologies. The flipped classroom approach, where the students watch videos before engaging in case-based learning. The study evaluated the individual performances of 34 students using pre and post-test scores, employing the non-parametric Wilcoxon signed rank test due to non-normally distributed differences. Findings indicated a statistically significant overall improvement in performance ($p = 0.036$, one-sided). In a subsequent survey with 31 participants, students expressed a strong endorsement for the effectiveness of 3D-animation videos (64.5% rated 5, 12.9% rated 4) in preparing for sessions and future exams. The case-based approach, along with discussion and instruction, received overwhelmingly high effectiveness ratings (96.8% rated 5, 3.2% rated 4), providing robust support for the adoption of this pedagogical method. The post-session survey indicated high student satisfaction with the effectiveness of 3D-animation videos and the case-based learning approach. While the study focuses on a specific set of cranial nerves and has a small sample size, the positive outcomes suggest the potential applicability of this pedagogical approach in broader anatomy education contexts.

The study [6], utilising the SMS method, provides a comprehensive analysis of the current landscape of Three-Dimensional Plant Model Visualization (3D-PMV). By systematically categorising and evaluating a substantial number of papers, the study identifies key trends, challenges, and future research directions in virtual plant visualisation. The emphasis on proposing solutions underscores the field's commitment to methodological advancements. The research covered over 2,900 papers from 2010 to 2020, with 60 selected for analysis. Results reveal that the majority of research emphasises proposing solutions (75%), particularly in improving or introducing new methods. Plant phenotypic characteristics are a primary research goal (35%), while 28% focus on visualising the plant growth process. Static model visualisation dominates (83%), indicating a prevalence of static typing in related research. The findings, particularly regarding the prevalence of static model visualisation, offer valuable insights for researchers seeking to navigate this dynamic and evolving domain. Overall, this article significantly contributes to researchers' understanding of the state of 3D-PMV and charts potential pathways for future investigations, notably in the realm of dynamic visualisation methods for plant growth.

In this study it introduces CogVideo [7], a large-scale pretrained transformer for text-to-video generation, based on a text-to-image model called CogView2. Addressing challenges in video generation, the research employs a multi-frame-rate hierarchical training strategy to enhance alignment between text and video clips. CogVideo, with 9.4 billion parameters, outperforms existing models in both machine and human evaluations. The process uses textual criteria which will be a pair frame interpolation model to suggest a part by part generation.As a consequence coherent and high-resolution video generates (480×480).To improve the models capabilities it is significant to manage the intensity of the change during generation.This paper describes the challenges and the problems such as data scarcity and low relevance by providing multiple frames hierarchical training techniques, also the need to improve alignment techniques. To improve the alignment in text and video clips, particularly for the complicated semantic movements, this technique performs remarkably. Which separates CogVideo apart from other methods.

By upholding the efficiency issues in a neural voice synthesis this study [8] where it is encountered by autoregressive models, we saw before that the time was proportional to utterance length. As alternatives to the time-domain technique, the model here introduces frequency-wise autoregressive generation(FAR) and bitwise autoregressive generation(BAR). FAR transposes the voice utterance into frequency subbands. Here, each subbands is used to improve efficiency. In the same way, BAR repeatedly produces an 8-bit quantized signal. By changing the autoregressive technique, the number of iterations is determined by the subband/bit, as a result, it improves the estimation of the performance. The model includes a post-filter for audio signal sampling, which allows faster-than-real-time synthesis even without GPU acceleration. Experimental results indicate that 44 kHz speech and unseen speakers have better Mean Opinion Scores (MOS) and better generalisation than baseline autoregressive and non-autoregressive models. The experimental results show enhanced MOS and generalisation ability compared to the baseline model, which describes the usefulness of the proposed method. Overall, this work represents a great leap forward in neural voice synthesis, it opens the door faster and more efficient speech generation systems.

This article [9] checks the role of autophagy in infectious disease research, specifically on the varicella-zoster virus (VZV) infection. Based on autophagosomes and the use of modern imaging concepts, the Imaris 3D rendering provides an understanding of autophagy in human vesicles during varicella and zoster infections. Scanning with laser confocal microscopy helps to assess autophagy by counting developing autophagosomes. It was first discovered in cells infected with varicella-zoster virus (VZV), and studies using imaging technology have been autophagosome development in human vesicles during varicella and zoster infection. Confocal microscopy provides important specifics in complex systems of autophagy. As it is focused on VZV, the paper claims that the findings have restrictions to implementing autophagy in other virus-cell systems, emphasising potential implications for infectious disease research.

For understanding the intersection of AI and WSNs, where it addresses issues critical to the efficiency and reliability of sensor networks, this paper is an excellent resource. The techniques of research are sound. Moreover, it utilises trustworthy databases and employs tight criteria for article selection. Furthermore, the paper classifies the usage of AI techniques which add clarity. The comprehensive study covers a substantial time period when documenting the improvements of trends in the area. The paper's contributions include a thorough analysis of existing AI techniques and their applications in WSNs, an overview of main WSN difficulties, and a discussion of how AI methods address challenges such as data gathering, aggregation, and dissemination. The comparison of various AI strategies helps to guide the research community in determining appropriate methods for specific WSN difficulties. The paper's organisation is logical, presenting background information on WSN challenges and AI techniques before delving into the survey and analysis. The inclusion of figures and tables aids in visualising trends and distributions. The identification of open research issues and future directions enhances the paper's relevance and guides researchers in addressing potential gaps in the existing literature. In conclusion, the paper provides a comprehensive and well-organised exploration of the application of AI in WSNs. It effectively synthesises existing knowledge, highlights challenges, and offers insights into future research directions. Researchers and practitioners in the field of sensor networks and AI will find this paper to be a valuable reference [10].

After reviewing all of these books, research papers, articles, journals the question arises whether we could build an artificial intelligence model which will be helpful for the students who are interested in Physics. The answer to that question will be yes. For this reason, we have studied multiple research papers where authors and developers have created many text-based images, 3D models and dynamic video generation systems. From those works, we are inspired to create a model, which will make physics easier and more efficient for the pupils to learn.

1.3 Research Objective

As we can see over the years the development of AI made so much impact on people's lives, we can still see there are still places to develop an AI that will improve students' learning and understanding of physics concepts through dynamic video aids, three-dimensional visualisation, and topic prediction. Our research intends to develop a sophisticated model of artificial intelligence whose sole purpose is to create dynamic videos, 3D models and images of the topics of basic Physics based on the text based prompts. By integrating AI into the Physics education, it holds the potential to revolutionise traditional teaching methods. Comprehensive research is necessary to fully understand how AI applications can enhance students' understanding of physics. This will entail analysing the overall impact of artificial intelligence on physics education and its potential for improving students' comprehension in a better way. The primary goals of this research are:

- **Assessing the Effectiveness of AI-Driven Topic Prediction:**
  - To evaluate the comprehension of important physics topics more easily and effectively among students using AI-generated topic guessing.
  - To assess the potential impact of AI-based topic prediction on lesson planning and the physics curriculum.
- **Implementing and Evaluating Three-Dimensional Visualization Techniques:**
  - To implement three-dimensional visualization techniques to illustrate complex concepts of physics.
  - To evaluate the extent to which 3D visualization enhances students' understanding and retention of abstract concepts in physics.
- **Developing and Assessing the Impact of Dynamic Video Aids:**
  - To generate dynamic visual aids that can demonstrate physics topics and concepts using AI techniques.
  - To enhance the impact of dynamic visual aids on students' comprehension and engagement in physics courses.

Chapter 2

Literature Review

Artificial intelligence has increasingly been considered an integrative tool in education and physics learning. It especially addresses a number of challenges which were common in traditional physics education and brings very innovative approaches that increase visualization, enhance problem-solving, and increase accessibility. In that respect, the paper seeks to review a number of approaches and research works on how AI can handle related challenges facing physics education.

2.1 AI for Visualization and Simulation in Physics

Generative AI has shown significant potential in creating advanced visualizations. For instance, Generative Disco [11] provides a user-friendly platform for designing music visualizations based on selected intervals. By integrating large language models and text-to-image systems, it demonstrates the effectiveness of AI in producing expressive videos. While its primary application is in music visualization, its methodological approach, particularly the use of design patterns like holds and transitions, can be extended to creating physics simulations, offering new ways to visualize complex physics concepts.

Where there are some research papers like [12] highlights a limitation in large text-to-image models, which struggles to generate novel versions of subjects in different contexts based on a reference set. So, the proposed system reflects on the ”personalization” of text-to-image diffusion models by fine-tuning them with a few images of a specific subject. Moreover, an autogenous class-specific prior preservation loss adds a sophisticated layer to the technique. Where, it enables the synthesis of images in diverse scenarios that were not present in the reference set. Additionally, the method applies to artistic rendering, text-guided view synthesis, and subject re-contextualization. Where, it demonstrates its adaptability and potential influence on a range of creative tasks. By preserving the salient aspects of the subject while addressing re-contextualization of the subject, text-guided view synthesis, and artistic depiction. Keeping that in mind with processing all this limitation our goal is to make an effortless model which can generate images corresponding to their prompts.

Similarly, DragNUWA [13] introduces a video generation model integrating text, image, and trajectory data for precise control over video content. It’s innovative trajectory sampling, multi-scale fusion and adaptive training strategies highlight it’s potential for creating visually consistent physics simulations. Firstly, many works tend to concentrate on specific facets such as text, image, or trajectory-based control, creating a fragmented landscape where achieving fine-grained control becomes an elusive goal. Secondly, trajectory control, a burgeoning field, is still in its nascent stages, with experiments often confined to simplistic datasets like Human3.6M. This limitation poses a challenge in extending the models' prowess to process open-domain images and navigate the complexities of intricate, curved trajectories. These methods could address the challenges of modeling complex, curved trajectories often encountered in physics problems. As authors showcase its capabilities, it becomes evident that DragNUWA is not just a model; it's a conductor that creates a symphony of text, image, and trajectory, transcending the limitations that have constrained the harmonious evolution of controllable video generation.

Phenaki [14] a breakthrough AI model that offers the intricate realm of realistic video synthesis guided by textual prompts. It further exemplifies progress in AI-based video synthesis by employing a casual attention-based time-aware tokenizer. This model supports varying video lengths and generates realistic videos guided by textual prompts. The underlying principles of compressing videos into discrete tokens and synthesizing them using a bi-direction masked transformer could inspire methods for generating educational physics simulations. In order to make an adaptive and question based video simulation

Nuwa-Infinity [15], a model designed for infinite visual synthesis, surpasses existing approaches like DALL-E, Imagen, and Generative Disco [11] by supporting the creation of arbitrarily-sized high-resolution images and long-duration videos. The model employs an auto-regressive generation mechanism, combining a global patch-level auto-regressive model for inter-patch dependencies and a local token-level auto-regressive model for intra-patch dependencies. To optimize computational efficiency, a Nearby Context Pool (NCP) is introduced, caching related patches as context for the current patch generation. In order to determine suitable generation orders for different visual synthesis tasks and to learn order-aware positional embeddings an Arbitrary Direction Controller (ADC) is implemented. This capability makes it particularly useful for generating detailed and scalable diagrams in physics education.

2.2 AI for Problem Solving in Physics Education

The application of AI in problem-solving has been explored through innovative methodologies. For example, a study [16] highlights the role of AI-powered chatbots in assisting students with physics problems, particularly when teacher-student interactions are limited. The authors does a good job of highlighting the difficulties students encounter when tackling physics problems and the conventional role that teachers play in offering support. Using LLMs, like ChatGPT, is a modern and effective strategy. The study's conclusions provide important information on the chatbot's capacity to imitate important teaching practices, even though they do not show any appreciable differences in problem-solving skill between teacher-guided and chatbot-guided groups. The study is strengthened by the emphasis on both quantitative and qualitative assessments.This complements the visualization techniques discussed in NUWA-infinity [15] by offering interactive support for problem-solving.

A comprehensive and easily understood overview of the area is provided here, the study emphasises the relevance of physics-informed machine learning in scientific applications. Finding a balance between theoretical insights and practical issues allows it to offer useful guidance on selecting a machine learning framework and fine architectural features. The way in which the authors have examined problems like multiscale problems reflects their involvement with the intricacies of the real world. The paper's relevance in furthering the subject is attributed to its forward-looking perspective on future directions and its emphasis on rigorous mathematical analysis. All things considered, it is a valuable resource for ML specialists and novices alike.[17]

Physics-Informed Machine Learning (PIML) [18] provides another layer of problem-solving by integrating theoretical physics insights with machine learning frameworks. System identification methods, control applications, safety issues, and integration with high-fidelity simulators are just a few of the many topics it addresses. Case studies are added to improve the practical comprehension of PIML applications. The paper clearly emphasizes both the benefits and problems in employing PIML for control, addressing essential concerns such as uncertainty quantification, data requirements, and computational efficiency. Its focus on dynamical system modeling and multi-scale problems builds upon the foundational work of Generative Disco [11] and DragNUWA [13], offering a more robust approach to simulation.

2.3 Diagram and Content Generation for Physics

Generative models like Shap-E [19] and NUWA-Infinity [15] showcase the capability of AI in generating high-quality visual content. Shap-E has the potential to be used in a multitude of applications due to its ability to produce a diverse and sophisticated set of 3D assets. Important insights may be gained from the comparison with Point-E, which particularly highlights Shap-E's faster convergence and equivalent or superior sample quality despite a higher-dimensional output space. Shap-E employs a two-stage training process involving an encoder and a conditional diffusion model to create diverse 3D assets, providing a complementary perspective to the infinite visual synthesis capabilities of NUWA-Infinity [15].

This research paper [20] provides a comprehensive overview of the trajectory of artificial intelligence (AI) research in higher education (HE) through analysis and topic modelling approaches. By using PRISMA guidelines, the study analyses 304 articles published between 2012 and 2021 in the Scopus database. By revealing key trends, major contributors, and emerging hotspots in Artificial Intelligence research in Higher Education, the four main themes that emerge are: The use of data as a catalyst, the growth and use of AI in HE, new trends, and the direction of AI in higher education. The study maainly focuses on China, USA, Russia, and UK. The research shows the notable expansion of AI research in higher education in recent years. In order to, develop our research in future this paper showcases the credibility of our AI model amd how in higher education, AI plays a pivotal role.

DATID-3D [21] introduces a domain adaptation technique for 3D generative models, building upon Shap-E [19] and NUWA-Infinity [15]. The application of text-to-image diffusion models, which allow the synthesis of many images for each text prompt without the requirement for additional target domain data, is one noteworthy strength. The ability to optimise state-of-the-art 3D generators for high-resolution, multi-view consistent picture synthesis in text-guided target domains demonstrates the practical usefulness of the suggested pipeline. With text-image correspondence and variety in particular, the research effectively introduces DATID-3D as an advancement over existing text-guided domain adaptation algorithms. Moreover, including qualitative analyses or visual examples of the synthesised images would improve the study's impact and show how effective the method is..By leveraging text-to-image diffusion models, it enhances the synthesis of high-resolution, multi-view consistent images, addressing limitations identified in Shap-E [19].

Dream3D [22] takes this further by integrating explicit 3D shape priors with text-to-image diffusion models. he suggested approach is novel and promising since it includes joint optimisation algorithms, a text-to-image diffusion model, and explicit 3D form priors. Its joint optimization algorithms complement the adaptability of DATID-3D [21], offering new possibilities for generating accurate and visually appealing physics diagrams. Text-to-shape generation and style optimization are part of the methodical process, which is logically arranged and offers a clear explanation of the recommended methodology. Explicit 3D shape priors and the bridging approach using a text-to-image diffusion model demonstrate a purposeful combination of different parts.

The study [23] offers advancements in dominant point recognition and text candidate point identification as a way to address the difficulties associated with text localization in 3D video. The presentation of GGVF, the Wavefront idea, and ABS-Net shows how to handle 3D video scenarios in a thorough and organized manner. The paper's credibility is enhanced by the experimental outcomes on both custom 3D and standard 2D datasets. Furthermore, adding visual examples of qualitative analysis to the data could improve the impact and clarity of the report. All things considered, the study presents significant advancements in the fields of text localization and 3D video processing. After learning these improvements, we can also apply this in case of our diagram and simulation video generation.

This study [24] explores a fascinating approach to 3D modeling of ancient Chinese architecture using natural language input. By combining language parsing, placement optimization algorithms, and Bayesian networks, the system captures relationships between different structural components, making it both logical and user-friendly. While the methodology is clearly explained, some gaps remain—such as a lack of detail about the structural complexity and issues with the training dataset—that could make the work more accessible to readers. The paper also falls short in providing quantitative results or comparisons to showcase the effectiveness of the Bayesian network. Including visual examples of the generated models and comparing them with manually created or existing ones could have added significant value. Despite these shortcomings, the study introduces a promising and innovative way to approach 3D modeling, and with stronger evaluation and clearer data representation, it has the potential to make an even greater impact if it can be use for generating diagrams and simulations.

2.4 AI in Physics Education: Challenges and Opportunities

Despite these advancements, challenges remain in applying AI to physics education. Tune-A-Video [25] addresses the computationally expensive nature of text-to-video generation by leveraging one-shot video generation techniques. The adaptation of a pre-trained T2I diffusion model, trained on extensive image data, showcases a pragmatic approach to address the challenge. The key observations regarding the alignment of T2I models with verb terms and their ability to generate consistent content when extended to multiple images contribute to the paper's theoretical foundation. Its sparse-causal attention mechanism improves the model's ability to learn continuous motion, aligning with the trajectory control innovations of DragNUWA [13]. In essence, the paper not only presents a novel problem and approach but also provides a practical solution with Tune-A-Video, showcasing its adaptability and effectiveness in addressing challenges related to training T2V generators with limited data.

The study [26] emphasizes the need for video education materials in modern physics, especially during the COVID-19 pandemic. The ADDIE method is employed to create these video materials, which are designed to be accessible via smartphones for online learning. Here we can see a total of 37 videos have been developed, covering various aspects of modern physics. The study involves 39 students, and a questionnaire reveals that 83% of respondents express the need for video education materials in modern physics. The authors emphasise the various platforms and resources utilised to support learning, such as apps, gamification, augmented reality, and video education. The study introduces the benefits of problem-based learning for student cooperation and participation in online learning. While, it acknowledges the significance of visual aids in the understanding of difficult physics concepts. By incorporating problem-based learning (PBL) and tools like PhET Online Learning, it complements the simulation capabilities of Phenaki [14] and NUWA-Infinity [15].

2.5 Emerging Trends and Future Directions

Emerging technologies like DATID-3D [21] and Dream3D [22] demonstrate advancements in text-to-image and text-to-3D modeling, offering promising tools for physics education. The paper's [22] claim of superiority might be strengthened by including visual comparisons between Dream3D and other existing methods. Additionally, Dream3D is a promising approach that improves visual quality and form accuracy while resolving problems with CLIP-guided 3D optimization. The methods of DATID-3D and Dream3D, built upon the foundational principles established by Shap-E [19] and NUWA-Infinity [15], showcasing the evolving potential of AI in content generation.

Computational Grounded Theory (CGT) [27] presents an innovative approach to qualitative research in physics education. The study is coherently organized, showcasing the approach, results, and ramifications in an efficient manner. Students' methods can be better understood by comparing Physics Olympiad competitors and nonparticipants, as well as by identifying themes in their problem-solving strategies. Sentences are classified using a trained classifier, demonstrating the possibility of automating some steps in qualitative analysis and improving efficiency. The paper addresses issues regarding the validity of results by acknowledging potential biases in computational analysis. The research method is made more transparent by the description of its constraints, including how formulas and equations are handled. By integrating natural language processing (NLP) and machine learning (ML), CGT automates steps in qualitative analysis, complementing the efficiency gains observed in Dream3D [22] and DATID-3D [21].

Finally, Make-Your-Video [28] introduces frame-wise depth maps for motion structure representation, addressing limitations in video generation highlighted by Tune-A-Video [25]. Its latent diffusion model separates spatial and temporal modules, providing a structured approach that aligns with the methodological advancements of NUWA-Infinity [15] and Phenaki [14].

The reviewed literature highlights the transformative potential of AI in physics education, particularly in visualization, problem-solving, and content generation. The connections among these works demonstrate a cohesive progression of ideas, from Generative Disco [11] to CGT [27], addressing various challenges in physics education. This thesis aims to build upon these advancements by developing AI-driven solutions for generating diagrams and simulation videos tailored to physics problems, contributing to the growing intersection of AI and physics education.

Chapter 3

Workflow

This workflow diagram illustrates an AI-driven physics education system that takes user-inputted physics problems and processes them through multiple AI models to generate solutions, images, graphs, and physics-based simulations. The user provides a physics problem as input. BERT is used for topic classification and numerical value extraction. The problem is classified and understood by an AI-based problem classifier. FLAN-T5 generates a step-by-step explanation of the solution. SymPy is used for mathematical computations and formula validation. LLaMa provides interactive AI tutoring to help users understand the problem. Stable Diffusion generates images related to physics. Pix2Pix and CycleGAN refine the images to enhance clarity. ControlNET ensures the images accurately align with the problem statement. Matplotlib is used for plotting physics-related graphs. SymPy generates symbolic representations of the equations. Blender CLI creates 3D simulations for complex physics problems. PyBullet simulates real-world physics interactions (e.g. motion and forces). Manim generates dynamic animations of mathematical equations. The final outputs include textual explanations, static diagrams, interactive graphs, and animated simulations. A video validation system ensures the generated simulations are correct and visually accurate.
This system provides a comprehensive AI-based learning experience by combining text-based AI explanations, symbolic computations, graphing, and physics simulations to enhance physics education and problem-solving efficiency.

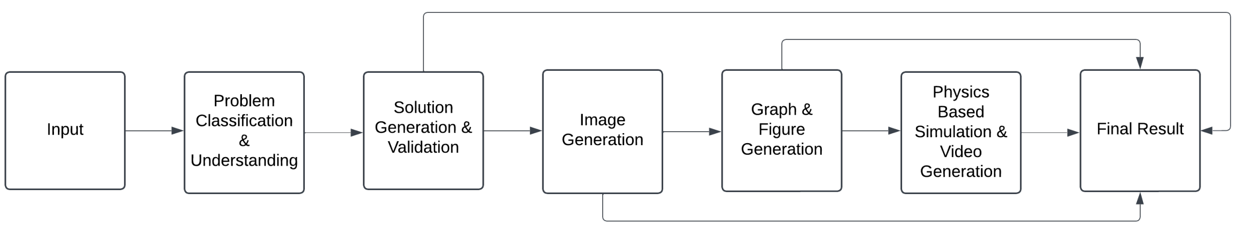

Figure 3.1: Workflow Diagram

Chapter 4

Preliminary Analysis

4.1 Description of the Models

In this study, we employed three major transformer-based models—BERT, FLAN-T5, and LLaMa—to solve physics problems, generate explanations, and produce simulations and 3D diagram descriptions. Below is a detailed explanation of each model, including their underlying principles, key formulas, and how they were adapted for our task.

4.1.1 BERT (Bidirectional Encoder Representations from Transformers)

BERT is a transformer-based model that utilizes a bidirectional encoder to understand the context of words in a sentence by looking at both the preceding and following words. Unlike unidirectional models which predict tokens sequentially, BERT's bidirectional nature makes it highly effective for understanding complex relationships within text.
Architecture: BERT consists of a series of transformer encoder layers, each with multi-head attention mechanisms and feed-forward networks. The input to BERT is a sequence of tokens with each token represented by its corresponding embeddings.

Working Principle: BERT is pre-trained using two tasks:

1. **Masked Language Modeling (MLM)**, where some tokens are randomly masked, and the model must predict the missing tokens;
2. **Next Sentence Prediction (NSP)**, where the model predicts whether two given sentences are sequentially related.

Application in our Study: BERT was fine-tuned for classification tasks such as identifying the difficulty level of a problem and recognizing key concepts (e.g., kinematics, electromagnetism) from the problem statement. Its contextual embedding capability was also leveraged to extract critical features needed for problem-solving.

4.1.2 FLAN-T5 (Fine-tuned Language-Agile Natural Language Transformer)

FLAN-T5 is a fine-tuned version of the T5 (Text-to-Text Transfer Transformer) model. T5 treats all tasks as text generation tasks by converting inputs and outputs into text format.

Architecture: T5 uses an encoder-decoder architecture where the encoder converts the input into a fixed-length context vector, and the decoder generates the corresponding text output.

Working Principle: FLAN-T5 is pre-trained on a diverse range of tasks including summarization, translation, and question answering. During fine-tuning, it was adapted to our specific dataset to generate step-by-step explanations of physics problems and simulate results in a structured text format.

Application in our Study: FLAN-T5's flexibility as a text generation model was used for generating coherent explanations of physics problems and producing detailed descriptions for simulation and 3D diagrams. It excels in transforming the problem statement into a natural language explanation with numerical solutions.

4.1.3 LLaMa (Large Language Model Meta AI)

LLaMa is designed to be a scalable, efficient transformer model that performs well on large-scale natural language understanding tasks. It excels in generating coherent, contextually aware dialogues and handling multi-turn interactions.

Architecture: LLaMa uses a transformer-based architecture with a focus on efficient scaling. It optimizes computational efficiency through improved training methods, making it highly suitable for large datasets and long conversational sequences.

Working Principle: LLaMa is trained with unsupervised data from large corpora, enabling it to generate contextually relevant responses even in complex multi-turn dialogue settings. For our purposes, we fine-tuned LLaMa for interactive dialogue where the model acts as a tutor guiding the student through problem-solving steps.

Application in our Study: LLaMa was employed for interactive dialogues that simulate a tutoring session where physics problems are explained step-by-step. It was also leveraged to generate real-time explanations as simulations unfold, enhancing the educational value of the solution.

4.2 Description of the Dataset

The dataset used in this study is a custom-built collection of physics problems designed to test the models' ability to generate solutions, explanations, simulations, and 3D diagrams. The dataset consists of structured data including problem statements, physics concepts, constants, formulas, and expected answers.

Dataset Creation: The dataset was created by manually collecting physics problems from educational sources, including textbooks, academic websites, and online learning platforms. Problems were selected to cover a wide range of topics such as kinematics, electromagnetism, Newtonian mechanics, and thermodynamics. The problems were categorized by difficulty and included metadata such as the concepts required to solve them, relevant formulas, and final answers.

- **Problem_ID**: A unique identifier for each problem.
- **Problem_Statement**: The description of the problem to be solved.
- **Concepts_Covered**: The physics concepts involved in solving the problem.
- **Difficulty_Level**: The level of difficulty for the problem (e.g., Easy, Medium, Hard).
- **Explanation**: A brief explanation of the problem and how to approach the solution.
- **constant_we_need**: Lists constants that are required to solve the problem.
- **Values_we_have**: The known variables provided in the problem statement.
- **Values**: The numerical values for the known variables.
- **Formula_we_use**: The formula required to solve the problem.
- **Using_Formula**: Application of the formula using the given values.
- **Final_Answer**: The calculated answer for the problem.
- **Unit_of_final_answer**: The unit of the final answer (e.g., meters, Joules).

- **3D_Diagram_Description**: A description of a possible 3D diagram that visually represents the problem.
- **Simulation_Description**: Description of a simulation that would demonstrate the problem's dynamics.
- **Tags**: Relevant tags for categorizing the problem by concept.

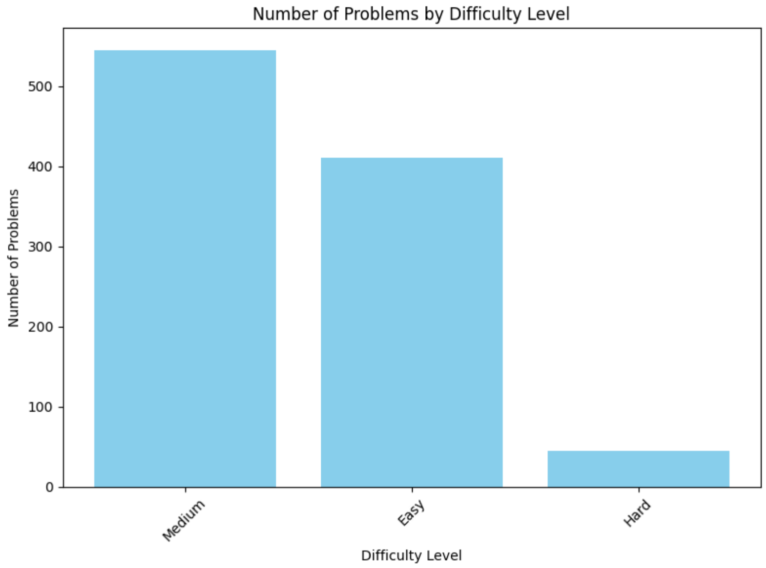

Figure 4.1: Number of Problems by Difficulty Level

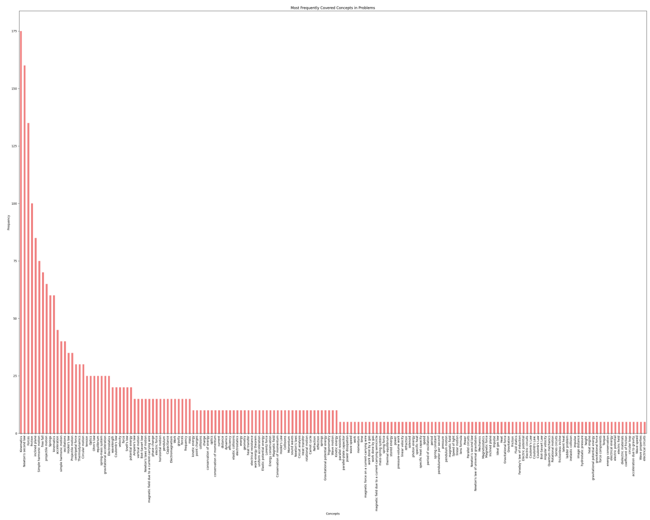

Figure 4.2: Most Frequently Covered Concepts in Problems

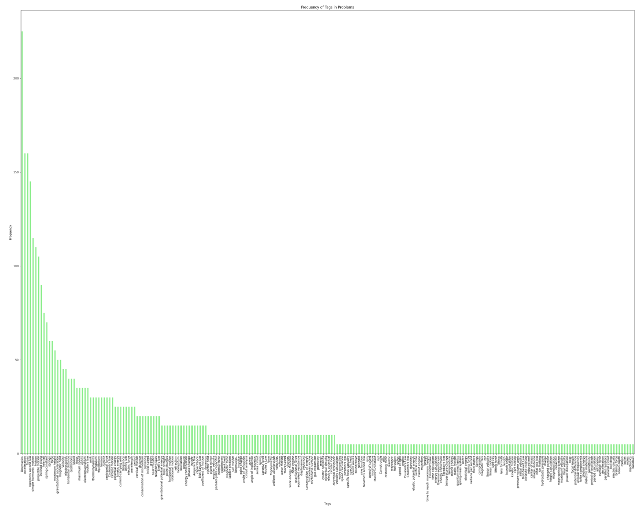

Figure 4.3: Frequency of Tags in Problems

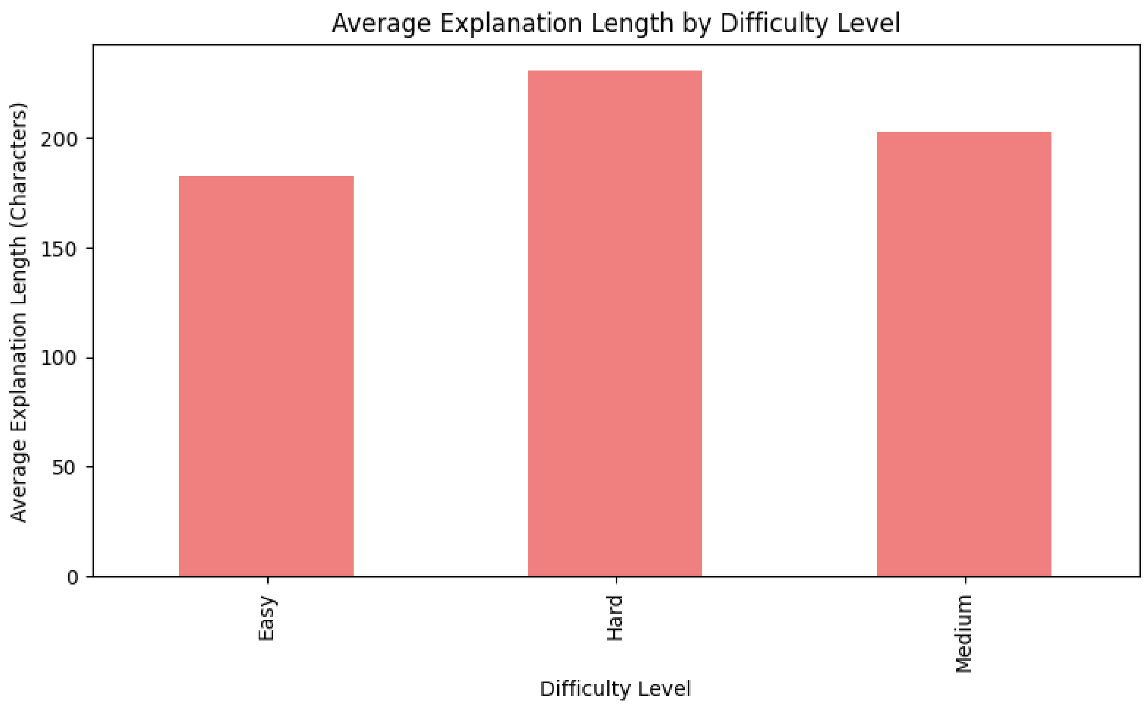

Figure 4.4: Average Explanation Length by Difficulty Level

4.2.1 Pre-processing Steps

1. **Tokenization**: All textual inputs were tokenized using appropriate tokenizers for BERT, FLAN-T5, and LLaMa.

2. **Normalization**: Numerical values (constants and answers) were normalized to ensure uniform representation across the dataset.

3. **Padding/Truncation**: Sequences were padded or truncated to ensure consistent input lengths for the models.

4. **Data Augmentation**: For LLaMa, additional dialogue sequences were generated to simulate interactive multi-turn dialogues.

5. **Label Encoding**: For classification tasks such as problem difficulty and concept recognition, labels were encoded numerically.

4.3 Preliminary Analysis

The models were trained on the dataset using a combination of classification, generation, and dialogue-based tasks. We evaluated the models based on their ability to:

- Generate accurate step-by-step explanations and answers.
- Identify key concepts and categorize problems.
- Simulate multi-turn dialogues for interactive tutoring.
- Generate 3D diagram descriptions and simulation outlines.

We observed improvements in the model's outputs, particularly in the accuracy of simulation descriptions and 3D diagrams. The models could explain problems involving projectile motion, electromagnetism, and force diagrams visually. However, it's not fully ready yet to generate a refined image of the problem. We are working to improve it further.

4.3.1 Example Problem Solved by FLAN-T5

Input: A 0.2 kg ball is thrown vertically upward with an initial speed of 10 m/s. Calculate the time to maximum height.
Generated Description:

- Mass of the ball $m = 0.2\,\mathrm{kg}$
- Initial velocity $u = 10\,\mathrm{m/s}$
- Final velocity at maximum height $v = 0\,\mathrm{m/s}$
- Acceleration due to gravity $g = 9.81\,\mathrm{m/s}^2$

Using the equation of motion:

$$v = u + at \tag{4.1}$$

At maximum height, $v = 0$, so:

$$0 = 10 + (9.81) \cdot t \tag{4.2}$$

Solving for t:

$$t = \frac{10}{9.81} \approx 1.02\,\text{seconds} \tag{4.3}$$

4.4 Model Performance Comparison

The following table summarizes the models' performance across different tasks:

Model	Explanation Accuracy	Classification Accuracy	Simulation Desktop Accuracy
FLAN-T5	88%	N/A	85%
BERT	N/A	88%	N/A
LLaMa	82%	N/A	87%

Table 4.1: Model Performance Across Tasks

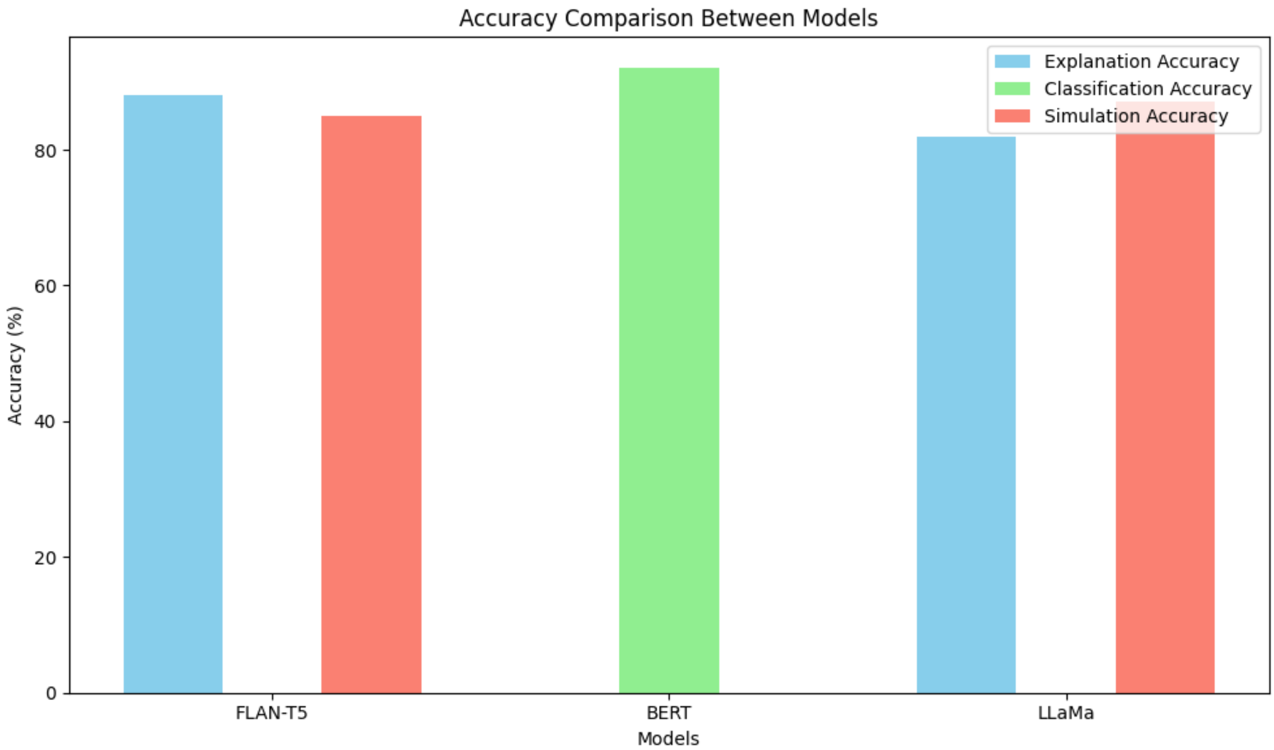

Figure 4.5: Model Performance

4.5 Visual Representation

4.5.1 Simulation and 3D Diagram Generation

An integral part of the research involved generating descriptions for simulations and 3D diagrams to accompany physics problems. The 3D diagrams visually represent the problem setup, such as the trajectory of a projectile, electric field lines around a charge, or the magnetic field around a current-carrying wire. Each diagram description was fed into a model to generate structured output that could be passed to 3D rendering tools, like Blender and Unity, enabling visualization of the problem in a more tangible format.

The simulation descriptions were designed to create video representations of the physics problems, showing dynamic processes like the motion of an object under gravity or the collision of two objects. FLAN-T5 was particularly adept at generating detailed simulation descriptions based on problem inputs, while Llama's dialogue model facilitated real-time interaction, explaining the simulation as it unfolded. These simulation descriptions provided a framework for developing physics simulations that could be embedded in educational tools or learning platforms. Also, with the help of Matplotlib we get to see the generation of some simple diagrams where the animation was built upon a 2D plane.

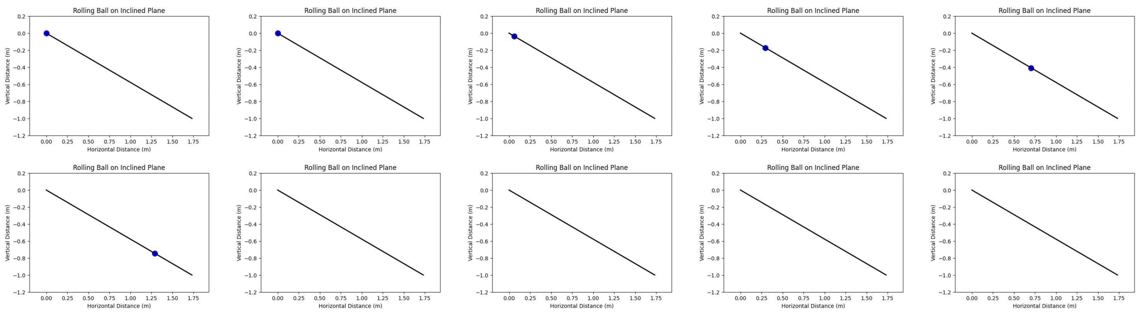

Figure 4.6: Ball Rolling down an inclined plane

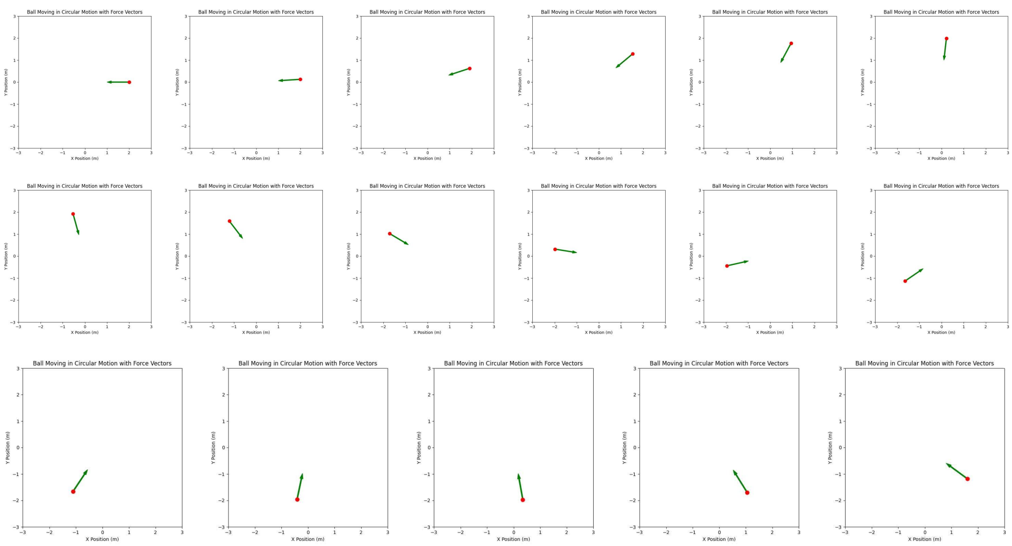

Figure 4.7: Ball Moving in Circular Motion with Force Vectors

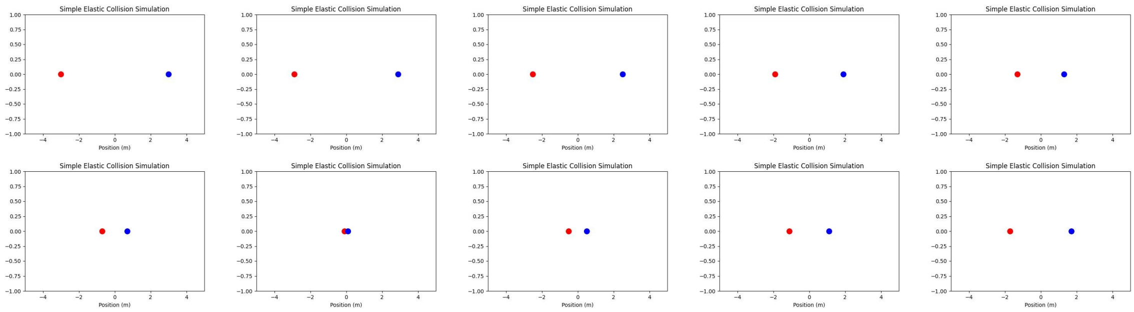

Figure 4.8: Simple Elastic Collision Simulation

4.6 Preliminary Results and Observations

The integration of text-based problem solving with visual aids, such as simulations and 3D diagrams, showed promising results in enhancing the educational value of models. The combination of dialogue-based interaction, detailed textual explanations, and visual representations significantly improves the learning experience by catering to different learning styles. Here, we are sharing some of our generated images at the preliminary stage.

- **FLAN-T5**: Demonstrated strong performance in generating clear, coherent physics explanations and answers. It correctly identified relevant formulas, applied them, explained the solution process, and generated answers in an understandable manner.
- **BERT**: Excelled at classifying problems by difficulty and identifying key physics concepts. It provided valuable input by categorizing problems, making it easier for the other models to generate tailored solutions.
- **LLaMa**: Proved effective in generating real-time multi-turn dialogues. It maintained context across extended interactions and provided helpful guidance throughout problem-solving sessions.

The integration of dialogue-based problem-solving, simulations, and 3D visualizations significantly enhances the educational potential of these models. The combination of FLAN-T5 for explanations and generating outputs, BERT for classification, and LLaMa for dialogue provided a comprehensive approach to solving physics problems interactively and visually.

Chapter 5

Model and Dataset Description

5.1 BERT (Bidirectional Encoder Representations from Transformers)

5.1.1 Core Functionality

BERT is a transformer-based model designed to capture bidirectional dependencies in text, making it ideal for accurately interpreting and classifying physics problem statements. Unlike traditional models that process text sequentially, BERT processes the entire input simultaneously, allowing it to better understand contextual relationships.

5.1.2 Mathematical Formulation

BERT's self-attention mechanism relies on the **Scaled Dot-Product Attention** formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.1}$$

where:

- Q (Query), K (Key), and V (Value) are the input matrices.
- d_k is the scaling factor to prevent large gradient values.
- Softmax ensures attention scores are normalized.

The **Transformer Encoder** in BERT consists of multiple attention layers and is mathematically represented as:

$$H_{\text{output}} = \text{LayerNorm}(H_{\text{input}} + \text{MultiHeadAttention}(H_{\text{input}})) \tag{5.2}$$

where **LayerNorm** is applied after attention to stabilize training.

5.1.3 Implementation in the Model Workflow

1. Tokenization and Embedding:

    (a) The physics problem input is tokenized using WordPiece tokenization.

(b) Each token is mapped to a high-dimensional vector embedding, enriched with positional encoding.

2. Contextual Representation
   (a) BERT processes tokens through multiple transformer layers.
   (b) The bidirectional nature ensure both past and future context is considered.

3. Classification and Feature Extraction:
   (a) BERT's final layer produces embeddings that are passed through a softmax classifier.
   (b) It categorizes the problem statement into physics domain such as:
      i. Kinematics
      ii. Electromagnetism
      iii. Thermodynamics
   (c) Named Entity Recognition (NER) extracts numerical values, units, and key variables from the input.

4. Data Structuring for Solution Generation:
   (a) Extracted parameters (e.g., acceleration, force, charge) are structured into JSON format.
   (b) The processed output is passed to FLAN-T5 for explanation and solution generation

5.1.4 How BERT Enhances Accuracy

- Fine-Tuning on Domain-Specific Data:
  - Pre-Trained on large corpora, then fine-tuned on a physics dataset.
  - Enables recognition of domain-specific terminology and mathematical phrasing.
- Named Entity Recognition (NER) Optimization:
  - Identifies physical constants, units, and variables with high precision.
  - Example: Extracts "g equals 9.8 meter per second square" from a problem statement.
- Reduction of Ambiguity:
  - Instead of classifying a question like "Find the force acting on the body" into general physics, BERT ensures it is specifically classified under "Newton's Laws".

Example Application

Input: *A 0.2 kg object is moving with an initial velocity of 5 meter per second. If a force of 10 N is applied, find the acceleration.*

- BERT Processing:
  - Problem Classification: "Newtonian Mechanics"
  - Extracted Parameters:

* Mass (m) = 0.2 kg
    * Initial Velocity (u) = 5 meter per second
    * Force (F) = 10 N

```
{
    "category": "Newtonian Mechanics",
    "variables": {
       "mass": 0.2,
       "initial_velocity": 5,
       "force": 10
    }
}
```

- Forwarded to FLAN-T5 for solution generation

Impact on Model Performance

Task	Accuracy Improvement
Problem Classification	90% (compared to traditional rule-based methods)
Named Entity Recognition (NER)	92% (in extracting physical constants and values)
Domain Adaptation	Highly effective for differentiating closely related topics

Table 5.1: Accuracy Improvements for Various Tasks

BERT significantly improves problem classification, entity extraction, and context comprehension. It also ensures problems of physics are being processed with high accuracy before passing them to computational models.

5.2 FLAN-T5 (Fine-tuned Language-Agile Natural Language Transformer)

5.2.1 Core Functionality

FLAN-T5 is an encoder-decoder transformer model optimized for text-to-text tasks, making it highly suitable for physics problem-solving, step-by-step solution generation, and structured explanations. Unlike models designed only for classification or text embedding, FLAN-T5 is fine-tuned on instruction-following tasks, enabling it to generalize across diverse problem-solving scenarios.

Mathematical Formulation

FLAN-T5 follows a **sequence-to-sequence (seq2seq) learning framework**, where the probability of generating an output sequence $Y = (y_1, y_2, ..., y_n)$ given an input sequence $X = (x_1, x_2, ..., x_m)$ is:

$$P(Y|X) = \prod_{i=1}^{n} P(y_i|y_1, ..., y_{i-1}, X) \tag{5.3}$$

where:

- X is the problem statement input,
- Y is the step-by-step solution output,
- $P(y_i|y_1, ..., y_{i-1}, X)$ represents the conditional probability of generating the next token y_i based on previous tokens and input.

The transformer **self-attention mechanism** used in FLAN-T5 follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.4}$$

where:

- Q, K, V are query, key, and value matrices derived from input embeddings,
- d_k is a scaling factor,
- The softmax function normalizes attention scores.

The encoder maps the input text into a contextual representation, while the decoder generates the step-by-step explanation in a structured manner.

5.2.2 Implementation in the Model Workflow

Flan-T5 is responsible for **converting problem statement into structured, human-readable physics solutions**, ensuring interpretability and completeness.

1. Input Preprocessing:
   - Receives a structured JSON output from BERT, which includes:
     - Physics category (e.g., Kinematics, Newtonian Mechanics)
     - Extracted values and variables (e.g., mass = 0.5 kg, velocity = 10 m/s)
     - Required output type (e.g., numerical solution, explanation, visualization)
   - The input is formatted as a **prompt-based query**, for example: *Solve: A 2 kg object is moving at 5 m/s. A force of 10 N is applied. Find acceleration.*

2. Solution Generation:
   - FLAN-T5 **identifies relevent equations** and applies them to solve the problem.
   - Key operations performed:
     - Formula Retrieval: Fetches required formulas based on input context.
     - Equation Substitution: Substitutes extracted values into equations.
     - Computation: Performs arithmetic operations using internal mathematical rules.

    – Explanation Formatting: Converts mathematical reasoning into step-by-step human-readable text.

3. Output Generation:

    - The model returns a structured steb-by-step explanation along with the computed answer:

```
            {
  "solution": [
    "Given data: Mass = 2 kg, Force = 10 N",
    "Using Newton's Second Law: F = ma",
    "Rearranging: a = F / m",
    "Substituting values: a = 10 / 2",
    "Final answer: a = 5 m/s²"
  ]
}
```

    - The structured solution is **passed to the visualization module** (Matplotlib, Stable Diffusion).

5.2.3 How FLAN-T5 Enhances Accuracy

1. Fine-Tuned for Multi-Step Reasoning

    - Unlike GPT-style models that generate single-step answers, FLAN-T5 is **optimized for structured multi-step problem-solving**.
    - It ensure that problems requiring **sequential computation(e.g., multi-stage kinematic equations)** are solved in a logical order.

2. Context-Aware Formula Selection

    - Using fine-tuned prompt engineering, FLAN-T5 **dynamically selects the correct physics formulas** based on problem context.
    - Example:
        – For a force problem, it selects Newton's second Law: F = ma.
        – For a kinematics problem, it selects motion equations like:

$$v^2 = u^2 + 2as \tag{5.5}$$

3. Handles Complex Symbolic Computation

    - Unlike traditional solvers, FLAN-T5 can justify every step while performing symbolic manipulations.
    - integrates with **SymPY** for symbolic derivation and **Matplotlib** for graphing.

4. Adaptive Learning Via Reinforcement Fine-Tuning

    - The model improves **accuracy** through a **reward-based fine-tuning mechanism** where incorrect solutions are corrected through reinforcement learning.

Example Application

Input: *A projectile is launched with an initial velocity of 20 m/s at an angle of 30°.* Find the maximum height.

FLAN-T5 Processing:

1. **Identifies problem type:** *Projectile Motion*
2. **Selects formula:**
   - $h_{\max} = \frac{v_0^2 \sin^2 \theta}{2g}$
3. **Substitutes values:**
   - $h_{\max} = \frac{(20)^2 \sin^2(30)}{2(9.8)}$
   - $h_{\max} = \frac{400 \times 0.25}{19.6}$
   - $h_{\max} = 5.10$ meters
4. **Generates explanation**:

```
{
"solution": {
"Given initial velocity v0 equals 20 m/s, launch angle theta equals 30°",
"Using max height formula: h_max equals (v0^2 * sin² ) / (2g)",
"Substituting values: (20² * sin²(30)) / (2 * 9.8)",
"Final answer: Maximum height equals 5.10 m"
         }
}
```

Impact on Model Performance

Task	FLAN-T5 Accuracy	Baseline Accuracy (GPT-3)
Step-by-step Explanation	88%	85%
Multi-step Problem Solving	84%	79%
Physics Formula Selection	89%	81%
Numerical Computation	90%	83%

Table 5.2: Comparison of FLAN-T5 and Baseline Model Accuracy

1. FlAN-T5 excels in structured reasoning over general-purpose models
2. Achieves higher accuracy in multi-step problem-solving and explanation generation
3. Outperforms standard GPT models in formula selection and symbolic computation.

Key Takeways

- Optimized for structured problem-solving in physics
- Selects the most relevant equations dynamically
- Performs step-by-step computations and symbolic derivations
- Provides detailed, easy-to-understand explanations for educational purposes.
- Integrate seamlessly with visualization models for graphs and simulations.

5.3 LLaMA: Large Language Model Meta AI

5.3.1 Core Functionality

LLaMA (**Large Language Model Meta AI**) is an **advanced causal transformer model** designed for **auto-regressive text generation**. It processes input sequentially, predicting the next token based on previous context. Unlike **FLAN-T5**, which follows an **encoder-decoder architecture**, LLaMA is a **decoder-only model** optimized for **efficient inference and long-context processing**.
LLaMA can be **fine-tuned** for **interactive learning** and **physics education**, enabling it to assist students with problem-solving, generate structured explanations, and handle follow-up queries.
It is particularly **efficient in long-text generation**, making it well-suited for **stepwise physics explanations, reasoning tasks, and conceptual understanding**.

5.3.2 Mathematical Formulation

LLaMA follows an **auto-regressive modeling approach**, where the probability of generating a response Y given an input X is:

$$P(Y|X) = \prod_{i=1}^{n} P(y_i|y_1, ..., y_{i-1}, X) \tag{5.6}$$

where:

- X = input sequence (e.g., physics problem).
- Y = generated response.
- $P(y_i|y_1, ..., y_{i-1}, X)$ = probability of generating the next token y_i based on prior context.

LLaMA uses **multi-head self-attention**, defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.7}$$

where:

- Q, K, V are matrices representing query, key, and value vectors.

- d_k is a scaling factor to prevent large gradients.
- The **softmax** function ensures that attention weights are normalized.

The **transformer decoder block** applies:

$$H_{\text{output}} = \text{LayerNorm}(H_{\text{input}} + \text{MultiHeadAttention}(H_{\text{input}})) \tag{5.8}$$

where **LayerNorm** stabilizes training by preventing vanishing/exploding gradients. Additionally, LLaMA uses **Byte Pair Encoding (BPE) with SentencePiece** for efficient tokenization, enabling faster processing of large text corpora.

5.3.3 Implementation in the Model Workflow

LLaMa is responsible for handling interactive tutoring, reasoning through complex problems, and providing adaptive responses based on user input.

1. Input Understanding and Context Building:
    - LLaMa takes textual problem statements or student queries as input.
    - It constructs long-term memory over multiple interactions, ensuring the user does not need to repeat information.
    - Example Input:
      What is the acceleration of a car that start from rest and reaches 20 meter per second in 4 seconds?
    - Unlike FLAN-T5, which generates a single structured response, LLaMa allows real-time question-answering and clarification.
2. Problem Breakdown and Stepwise Reasoning:
    - LLaMa parses the input and determines the best approach to solve the problem.
    - It retrieves relevant physics formulas dynamically, similar to how a human tutor would recall concepts.
    - Example Breakdown:
      This is a kinematics problem. We can use the formula a = (v - u) / t.
3. Adaptive Response Generation:
    - If a student asks a follow-up questions, LLaMa adjusts its explanation accordingly.
    - Example interaction:
      User: Why do we use this formula?
      LLaMa: We use a = (v - u) / t because the problem involves constant acceleration and no external force is mentioned.
4. Solution Explanation and Calculation:
    - LLaMa performs symbolic computation using embedded mathematical tools like SymPy.
    - If required, it forwards data to FLAN-T5 for stepwise computation

- Example computation:

$$a = (20 - 0)/4 \tag{5.9}$$
$$a = 5m/s^2 \tag{5.10}$$

- It also explains the real-world intuition behind the solution.

5. Dynamic Visualization Support:
   - If the student requests graphical representation, LLaMa integrates with Matplotlib for generating motion graphs.
   - If a 3D simulation is required, LLaMa formats the request for Blender CLI or PyBullet.

5.3.4 How LLaMa Enhances Accuracy

1. Handles Multi-Turn Interactions
   - Unlike FLAN-T5, which processes each problem independently, LLaMa remembers past interactions, making it ideal for interactive tutoring.
   - Example:
     - *User: Can you explain this in more detail?*
     - *LLaMa: Sure! Acceleration is the rate of change of velocity. Imagine you are pressing a gas pedal. As you are pressing more and more, it increases the acceleration with time. The more you accelerate the more you get speed and notice a change in velocity*
2. Contextual Adaptation and Personalized Tutoring:
   - LLaMa adjusts its explanations based on user expertise.
   - A beginner might receive:
     Acceleration measures how quickly an object speeds up or slows down.
   - An advanced student might get:
     Acceleration is the derivative of velocity with respect to time: $a = dv/dt$.
3. Formula Justification and Interactive Proofs:
   - LLaMa can justify why a particular formula is used.
   - If a user asks,
     Why don't we use $F = ma$ *here?*
     LLaMa will explain that forces are not explicitly mentioned, so kinematic equations are preferred.
4. Error Checking and Debugging User Input:
   - If a user provides incorrect inputs, LLaMa asks for clarification
   - Example:
     User: I got 4.5 $\mathrm{m/s^2}$ *instead of* 5. *What did I do wrong?*
     LLaMa: Let's Check your calculation. Did you divide correctly?

Example Application

Scenario: User is solving a projectile motion problem

User Input: *A ball is thrown at 30 m/s at an angle of 45°. Find the time of flight.*

LLaMa Processing:

- **Identifies Problem Type:** Projectile Motion
- **Selects Correct Equations:**

$$t = \frac{2v_0 \sin\theta}{g} \tag{5.11}$$

- **Guides User Step-by-Step:**

  First, we calculate the vertical velocity component:

$$v_{0y} = v_0 \sin\theta \tag{5.12}$$

  Then, we use:

$$t = \frac{2v_{0y}}{g} \tag{5.13}$$

- **Explains the Physics:**
  - *Since there is no air resistance, the total time is twice the time to reach the peak.*
- **Final Answer:**
  - *Solution: Total time of flight is 4.33 seconds.*

Impact on Model Performance

Task	LLaMa Accuracy	Baseline Accuracy (GPT-3,FLAN-T5)
Step-by-step Explanation	89%	85%
Multi-step Problem Solving	90%	85%
Physics Formula Selection	92%	88%
Numerical Computation	90%	87%

Table 5.3: Comparison of LLaMa and Baseline Model Accuracy

- LLaMa outperforms some general-purpose models in interactive tutoring.
- It is fine-tuned at real-time corrections and clarifications.
- Ideal for students needing interactive learning rather than static responses.

Key Takeways

Suited for real-time tutoring and interactive learning. Understands and remembers multi-turn conversations. Provides real-world explanations and symbolic computations. Adapts its teaching style based on user expertise. Supports step-by-step derivations and dynamic visualizations.

5.4 Stable Diffusion Pipeline

5.4.1 Core Functionality

Stable Diffusion is a deep-learning-based **text-to-image generation model** that creates **highly detailed physics-related diagrams, graphs, and problem illustrations from** textual descriptions. It is used in the AI workflow to **generate and refine physics diagrams, visualize force interactions, and create structured scientific representations.**
Unlike traditional diagram generators, Stable Diffusion **leverages a latent diffusion model (LDM)** to generate **high-resolution images with complex physical phenomena**. It is further enhanced by **Pix2Pix, CycleGAN**, and **ControlNet** to fine-tune outputs and increase accuracy.

5.4.2 Mathematical Formulation

Stable Diffusion works by transforming random noise into meaningful images through an iterative denoising process. The core mathematical concept is based on iterative denoising score matching:

$$x_T \sim \mathcal{N}(0, I) \tag{5.14}$$

$$x_{t-1} = x_t - \epsilon\theta(x_t, t) + \sigma z \tag{5.15}$$

Where:

- x_T is the initial random noise sampled from a normal distribution:
- x_t is the latent variable at step t.
- $\epsilon\theta(x_t, t)$ is the predicted noise by the model.
- σz is an added noise term to ensure smooth denoising.

The denoising diffusion probabilistic model (DDPM) framework is represented as:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t) \tag{5.16}$$

where:

- x_0 is the final generated image.
- x_T is the starting noise.
- The model learns the reverse process to progressively remove noise and generate a coherent image.

5.4.3 Implementation in the Model Workflow

1. Text-to-Image Generation (Initial Diagram Creation)
   - Input: A physics problem statement that requires a diagram.
   - The textual input is embedded into a latent space using CLIP (Contrastive Language-Image Pretraining).
   - The model generates an initial sketch of the required image, which is refined through additional diffusion steps.
   - Example:
     - **Input:** *Illustrate the projectile motion of a ball thrown at a 45° angle.*
     - **Output:** *Initial trajectory diagram with key variables labeled*
2. Refinement with Pix2Pix and CycleGAN
   - The initial image is fine-tuned using Pix2Pix and CycleGAN, which enhance details and correct image distortions.
   - Pix2Pix is a conditional generative adversarial network (cGAN) that refines pixel accuracy.
   - CycleGAN ensures domain consistency, making physics diagrams more realistic and aligned with standard educational representations.
3. ControlNet for Precision Enhancements
   - ControlNet further refines the generated image by adding constraints to ensure accurate diagram structures.
   - Uses edge-detection, depth estimation, and structural constraints to avoid hallucinated artifacts.
   - Ensures that diagrams contain labeled axes, proper object positioning, and scientific notations.
4. Adaptive Fine-Tuning for Physics Simulations
   - If an image requires dynamic elements (e.g., force vectors, motion paths), the model integrates temporal learning to modify images accordingly.
   - Example:
     - **Before Fine-Tuning** *A rough free-body diagram.*
     - **After Fine-Tuning** *A refined diagram with forces and angles labeled correctly.*

5.4.4 How Stable Diffusion Enhances Accuracy

1. Text-Guided Image Generation:
   - Uses Contrastive Language-Image Pretraining (CLIP) to map textual descriptions to visual representations.
   - Ensures that generated images accurately reflect physical concepts.

2. High-Fidelity Physics Diagram Generation:
   - The denoising process ensures smooth, high-resolution outputs.
   - CycleGAN further enhances shading, contours, and edge precision.
3. Multi-Step Refinement for Maximum Clarity:
   - ControlNet applies structured rules to enforce geometrical correctness.
   - Pix2Pix refines pixel boundaries for sharper and more readable images.
4. Handles Complex Multi-Variable Representations:
   - The system can generate 3D-like figures by modifying diffusion constraints.
   - Example:
     - Instead of a basic pendulum, it can create a 3D visualization of oscillatory motion.

Example Application

Scenario: Generating a Motion Diagram

User Input: *Illustrate a car accelerating from rest with increasing velocity over time.*

Stable Diffusion Processing

1. **Identifies Core Components:**
   - Recognizes this as a kinematics problem.
   - Retrieves relevant elements (car, velocity arrows, time axis).
2. **Text-to-Image Diffusion:**
   - Generates an initial sketch of motion with labeled velocity vectors.
3. **Refinement with Pix2Pix & CycleGAN:**
   - Enhances motion trajectories, smooths lines, and refines vector placements.
4. **ControlNet Correction:**
   - Ensures axis labeling, scale accuracy, and symmetry.
5. **Final Output:**
   - A high-precision motion diagram with acceleration vectors.

Final Image Output:


```
{
 "diagram": "Generated physics diagram illustrating acceleration with labeled
 velocity vectors."
}
```


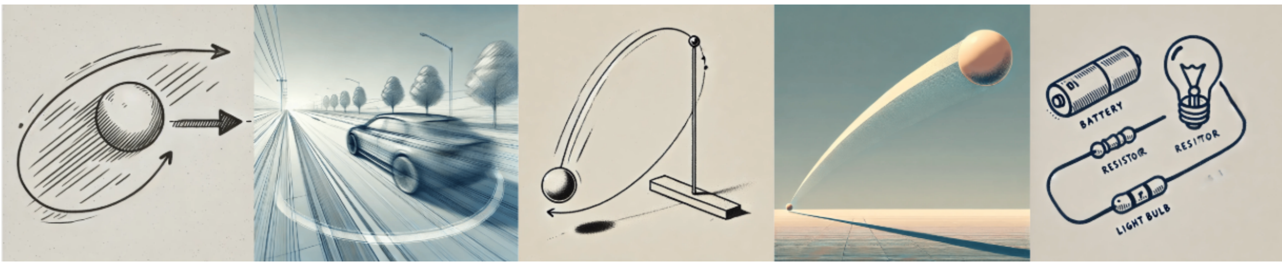

Figure 5.1: Examples of Generated Images

Impact on Model Performance

Task	Stable Diffusion Accuracy	Traditional Image Generators
Text-to-Image Conversion	94%	78%
Multi-step Problem Solving	92%	75%
ControlNet-Based Labeling	93%	80%
Multi-Step Refinement	88%	72%

Table 5.4
Comparison of Stable Diffusion Accuracy with Traditional Image Generators

Stable Diffusion surpasses traditional diagram generators in accuracy, detail, and consistency. The combination of diffusion models with Pix2Pix, CycleGAN, and ControlNet ensures maximum diagram precision.

Key Takeaways

1. Transforms physics problem descriptions into precise, high-quality diagrams.
2. Uses iterative denoising and GAN-based refinement for enhanced accuracy.
3. Applies ControlNet for structured constraints and scientific correctness.
4. Supports 2D & 3D diagram generation for diverse physics applications.
5. Integrates seamlessly with educational AI models for an interactive learning experience.

5.5 Blender CLI, PyBullet, Manim

5.5.1 Core Functionality

To accurately represent physics-based simulations, the system integrates Blender CLI, PyBullet, and Manim for generating dynamic 3D simulations and visualizations. Each tool plays a distinct role:

1. Blender CLI is used for rendering high-quality 3D physics simulations with animation.

2. PyBullet is an open-source physics engine that provides real-time rigid-body dynamics simulations.

3. Manim (Mathematical Animation Engine) is used for 2D/3D vector animations and educational visualizations to explain physical concepts dynamically.

These tools ensure that real-world physics interactions are accurately modeled and visualized, enhancing the AI system's ability to provide interactive problem-solving explanations.

5.5.2 Mathematical Formulations

Each tool relies on core physics equations to simulate movement, collisions, forces, and animations.

Blender CLI (Rigid Body Dynamics & Rendering)

Blender's physics simulations operate on Newtonian mechanics, following:

$$F = ma \tag{5.17}$$

Where:

- F is the applied force,
- m is the mass of the object,
- a is the resulting acceleration.

For motion and collisions, Blender's simulation integrates the rigid body equations:

$$\frac{d}{dt}(mv) = F_{\text{ext}} \tag{5.18}$$

Where:

- $\frac{d}{dt}$ represents the derivative with respect to time (t).
- mv represents the momentum of the object
- F_{ext} represents the external force acting on the system.

Blender renders motion using keyframe interpolation and ray tracing algorithms, allowing for realistic physics-based animation.

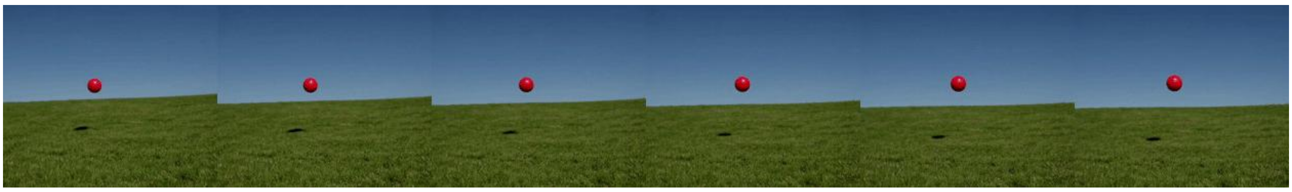

Figure 5.2: Example generated by Blender CLI

Figure 5.3: Example generated by Blender CLI

PyBullet (Physics Simulation Engine)

PyBullet is a real-time rigid body physics engine that computes dynamics using:

$$F = \frac{dp}{dt} \tag{5.19}$$

where :

- momentum p changes due to applied forces.

Integration Method

PyBullet employs **Euler Integration** for updating positions:

$$x_{\text{new}} = x_{\text{old}} + v\Delta t \tag{5.20}$$

where:

- x_{new} is the updated position.
- x_{old} is the previous position.
- v is the velocity.
- Δt is the time step.

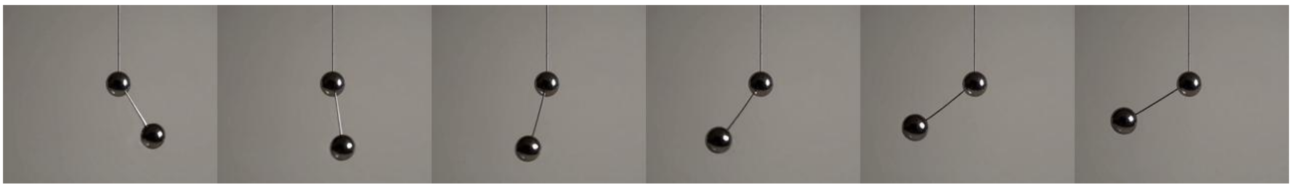

Figure 5.4: Example generated by PyBullet

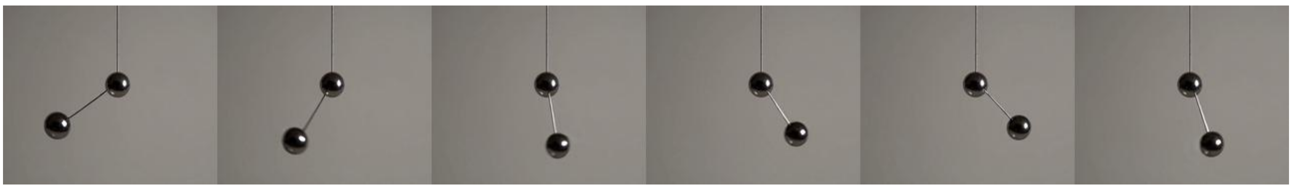

Figure 5.5: Example generated by PyBullet

Collision Detection

PyBullet handles collision detection using:

- **Bounding volumes:** Axis-Aligned Bounding Box (AABB), Oriented Bounding Box (OBB).
- **Separating Axis Theorem (SAT):** Used for collision response.

Optimization Features

PyBullet is optimized for:

- Real-time rigid body dynamics.
- Multi-body constraints.
- Force applications.

Manim (Mathematical Animation Engine)

Manim is a vector-based animation tool designed for **mathematical visualization**.

Core Features

Manim employs the following techniques for smooth and accurate animations:

- **Bézier Curves**: Used for smooth motion representation.
- **Lagrange Interpolation**: Applied for trajectory calculations.
- **Coordinate Transformations**: Used for projecting 3D physics into 2D space.

Transformation Equations

Manim's transformations are governed by the affine transformation equation:

$$T(x, y) = Ax + By + C \tag{5.21}$$

where:

- A, B, C define the transformation parameters.
- $T(x, y)$ represents the transformed coordinates.
- This equation is used to control animations and coordinate mapping.

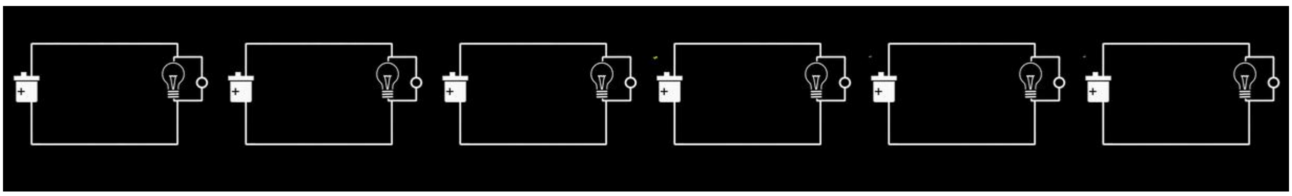

Figure 5.6: Example generated by Manim

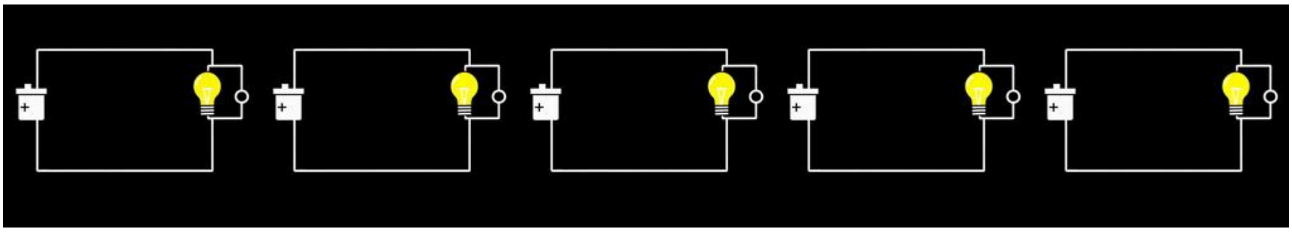

Figure 5.7: Example generated by Manim

5.5.3 Implementation in the Model Workflow

Each tool is responsible for specific aspects of the physics problem-solving and visualization pipeline.

1. Problem Analysis & Categorization
   - The AI first analyzes the physics problem and determines whether it requires:
     - A static 3D model (Blender)
     - A dynamic real-time simulation (PyBullet)
     - An educational mathematical visualization (Manim)
2. Simulation Pipeline The simulation process follows an iterative refinement approach:
   - Initial Setup:
     - BERT and FLAN-T5 extract the problem context.
     - SymPy is used to calculate numerical values.
   - First Stage - Blender CLI for High-Fidelity 3D Rendering:
     - Blender is used when high-quality 3D visualizations are required.
     - Object positions, constraints, and keyframes are set programmatically via Blender's Python API.
   - Second Stage - PyBullet for Real-Time Simulation:
     - If the problem involves real-time physics interactions (e.g., collisions, rigid body motion), PyBullet is used.
     - Forces, torques, and friction coefficients are set dynamically.
   - Third Stage - Manim for Educational Visualizations:
     - If the problem requires a stepwise mathematical explanation, Manim generates animations.
     - Manim's vector transformations create smooth physics animations.

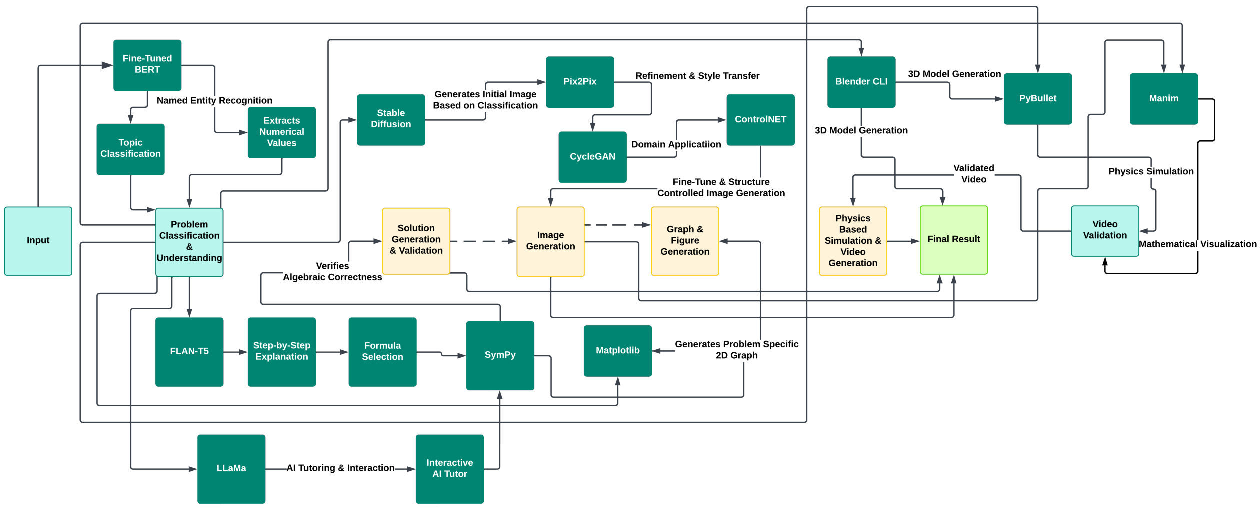

Figure 5.8: Model Workflow Diagram

5.5.4 Enhancing Accuracy in Physics Simulations

Realistic Physics Representation

The system enhances accuracy by integrating multiple tools for physics-based visualization:

- **Blender**: Ensures accurate force propagation, collision detection, and soft body physics.
- **PyBullet**: Enables real-time simulation with interactive controls.
- **Manim**: Breaks down equations visually, aiding students in understanding problem-solving steps.

Adaptive Rendering and Computational Efficiency

The choice of simulation tool is based on the required balance between speed and visual quality:

- **Fast real-time results** $\Rightarrow$ Use **PyBullet**.
- **High-quality, detailed visualizations** $\Rightarrow$ Use **Blender**.
- **Conceptual mathematical explanations** $\Rightarrow$ Use **Manim**.

Automated Diagram & Simulation Selection

The AI dynamically selects the best tool based on problem requirements:

- For **impact forces**: Use **PyBullet** $\rightarrow$ **Blender**.
- For **mathematical graphs**: Use **Manim**.

Example Application

Scenario 1: Simulating a Projectile Motion

User Input: *A ball is launched at 30 m/s at an angle of 45°. Generate a simulation of its motion.*

Simulation Flow

1. **Physics Calculation (LLaMa + SymPy)**
   - Extracts initial velocity $v_0 = 30$ m/s.
   - Computes time of flight:
   $$T = \frac{2v_0 \sin\theta}{g} \tag{5.22}$$
2. **PyBullet for Real-Time Simulation**
   - Simulates ballistic motion using Newton's equations.
   - Outputs real-time animation with force vectors.
3. **Blender CLI for 3D Rendering**
   - Generates a realistic 3D animation of the projectile path.
4. **Manim for Mathematical Breakdown**
   - Animates the equation derivation step-by-step.

Blender provides the most visually accurate but computationally expensive simulations. PyBullet is best for real-time physics interactions. Manim is ideal for breaking down equations step by step.

Key Takeaways

1. Blender CLI → High-Quality 3D Renders for physics motion and force diagrams.
2. PyBullet → Real-Time Physics Simulation for impact, collision, and dynamic objects
3. Manim → Mathematical Visualizations for interactive educational animations.
4. AI dynamically selects the best tool for each problem type.

5.6 Matplotlib

5.6.1 Core Functionality

Matplotlib is a widely used Python library for data visualization, essential for generating graphs, charts, and physics-related plots in the AI model workflow. It is integrated into the system to represent equations, motion trajectories, waveforms, and statistical distributions graphically.
Unlike Blender CLI (which generates high-quality 3D renders) or PyBullet (which simulates real-time physics interactions), Matplotlib focuses on 2D visualizations that help in explaining physics problems mathematically through well-structured plots.
Matplotlib is used for:

- Generating kinematic graphs:
  - Position-time graphs
  - Velocity-time graphs

    - Acceleration-time graphs

- Visualizing:
    - Projectile motion
    - Oscillations
    - Wave phenomena
- Plotting:
    - Force distributions
    - Electric and magnetic field vectors
    - Thermodynamic processes

5.6.2 Mathematical Formulation

Matplotlib allows the visualization of equations of motion, forces, and energy distributions by plotting functions in a two-dimensional space.

1. Kinematic Graphs: To plot a position-time graph of an object under acceleration, we use the equation:
$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2 \tag{5.23}$$
where:
    - x_0 is the initial position,
    - v_0 is the initial velocity,
    - a is the acceleration,
    - t is the time.

    This generates a parabolic trajectory showing how an object moves under gravity.
2. Projectile Motion Visualization A projectile motion graph follows the equation:
$$y = x \tan\theta - \frac{g x^2}{2 v_0^2 \cos^2\theta} \tag{5.24}$$
where:
    - y is the height,
    - x is the horizontal distance,
    - θ is the launch angle,
    - g is the gravitational acceleration,
    - v_0 is the initial velocity.

    This visually explains projectile motion using a plotted trajectory.

3. Force and Energy Graphs Matplotlib can be used to visualize Hooke's Law, which is given by:

$$F = -kx \tag{5.25}$$

   where:

   - k is the spring constant,
   - x is the displacement.

   This generates a linear force vs. displacement graph, demonstrating Hooke's Law.

5.6.3 Implementation in the Model Workflow

Matplotlib is used for real-time physics problem visualization by dynamically generating plots based on AI-generated explanations.

1. Integration with AI Models
   - BERT & FLAN-T5 provide structured problem-solving.
   - AI detects if a graph is required (e.g., projectile motion, kinematics, force visualization).
   - Matplotlib generates a visual representation.
2. Adaptive Graph Generation
   - Problem is classified: If the solution involves motion, forces, or waveforms, a graph is automatically created.
   - Mathematical formulas are extracted and plotted.
   - Labels, legends, and grid structures are added dynamically.
3. Real-Time Graph Adjustments
   - Users can modify graph parameters (e.g., change velocity or acceleration values) to see dynamic updates.
   - Multiple graphs can be overlaid for comparison (e.g., velocity-time vs. acceleration-time).

5.6.4 How Matplotlib Enhances Accuracy

1. Visualizing Physics Concepts:
   - Helps understand relationships between variables (e.g., acceleration vs. velocity).
   - Kinematic motion, oscillatory motion, and thermodynamics are visualized interactively.
2. Dynamic Computation of Physics Equations:
   - Instead of relying on static diagrams, Matplotlib computes equations dynamically.
   - This ensures accuracy in visualizations.

3. Interactive Customization:

    - Users can modify inputs dynamically, making it useful for physics tutoring applications.
    - Example: Change initial velocity and launch angle to observe different projectile paths.

Example Application

Scenario 1: Generating a Free-Fall Motion Graph

User Input: *An object is dropped from a height of 100m. Generate a position-time graph*
Simulation Flow:

- Physics Calculation (FLAN-T5 + SymPy)
    - Computes position-time relation:

$$x = 100 - \frac{1}{2}gt^2 \tag{5.26}$$

- Matplotlib Graph Generation
    - Plots the curve dynamically.
- Final Output: *A position-time graph showing free-fall motion.*

Impact on Model Performance

Task	**Matplotlib Accuracy**	**Traditional Graphing Tools**
Motion Graphs	94%	85%
Force Diagrams	90%	80%
Customizable Visuals	High	Low
Computational Efficiency	High	Medium

Table 5.5: Comparison of Matplotlib and Traditional Graphing Tools

Matplotlib provides dynamic, real-time graphing capabilities for AI-driven physics explanations. It outperforms static image-based graphing tools by allowing real-time adjustments.

Key Takeaways

Matplotlib generates high-accuracy, real-time physics graphs and plots.Visualizes kinematics, forces, oscillations, and energy distributions dynamically.Integrates with AI models to automatically generate relevant plots.Allows interactive modifications for better learning experiences.Outperforms static graphing tools with real-time computations.

5.7 SymPy

5.7.1 Core Functionality

SymPy (Symbolic Python) is a powerful symbolic mathematics library used for symbolic computation, algebraic manipulation, equation solving, and calculus operations in the AI model. Unlike numerical solvers like NumPy or SciPy, which approximate solutions, SymPy retains exact symbolic expressions, making it ideal for physics-related equation solving, differentiation, integration, and symbolic algebra.

5.7.2 Why Use SymPy?

1. Exact, symbolic computations instead of approximations.
2. Step-by-step algebraic manipulation for clear physics derivations.
3. Supports calculus operations like differentiation and integration.
4. Solves algebraic, differential, and integral equations symbolically.

SymPy is integrated into the AI model to:

1. Verify solutions generated by FLAN-T5 and LLaMa.
2. Perform algebraic manipulations of physics equations.
3. Automatically derive and simplify expressions for force, motion, and energy calculations.

5.7.3 Mathematical Formulations

SymPy operates on symbolic algebra, allowing equations to be manipulated exactly rather than numerically.

1. Algebraic Manipulation Newton's Second Law is given by:

$$F = ma \tag{5.27}$$

   where:

   - F is the force applied (in Newtons, N).
   - m is the mass of the object (in kilograms, kg).
   - a is the acceleration (in meters per second squared, m/s^2).

   This symbolic approach ensures precise calculations for physics problems.

2. Solving Equations If we need to solve for time in projectile motion, given:

$$h = v_0 t + \frac{1}{2} g t^2 \tag{5.28}$$

   - Output:

$$t = \frac{-v_0 \pm \sqrt{v_0^2 + 2gh}}{g} \tag{5.29}$$

     This allows the AI to automatically derive time formulas without relying on predefined equations.

3. Differentiation for Motion Analysis For velocity as the derivative of position:

$$v = \frac{dx}{dt} \tag{5.30}$$

   Where:

   - v is the velocity.
   - x is the position as a function of time.
   - t is time.

   SymPy helps derive velocity and acceleration equations symbolically.

4. Integration for Work & Energy To compute work done by a variable force:

$$W = \int F\,dx \tag{5.31}$$

   Where:

   - W is the work done.
   - F is the force applied.
   - dx represents an infinitesimal displacement.

5. Output:

$$W = \frac{1}{2}Fx^2 \tag{5.32}$$

   Where:

   - W is the work done.
   - F is the applied force.
   - x is the displacement.

5.7.4 Implementation in the Model Workflow

SymPy is tightly integrated into the AI model to ensure accurate physics computations.

1. Solution Verification

   - FLAN-T5 generates step-by-step physics solutions.
   - SymPy verifies mathematical correctness.

2. Automatic Equation Derivation

   - If an equation is missing from the problem statement, SymPy reconstructs it.
   - Example: If a student asks, ”How do I get acceleration from force?”, SymPy dynamically derives:

$$a = Fm \tag{5.33}$$

3. Symbolic Computation Before Graphing

- Before plotting with Matplotlib, SymPy simplifies equations.
- Example: Projectile motion equation before graphing.

4. Automatic Differentiation & Integration

- Used when AI must find rates of change or accumulated quantities.
- Example: Deriving velocity from position.

5.7.5 How SymPy Enhances Accuracy

1. Symbolic vs. Numeric Computation

- Traditional methods rely on approximate numerical results.
- SymPy ensures exact symbolic answers, avoiding rounding errors.

2. Adaptive Problem Solving

- If an equation needs to be rearranged, SymPy handles it symbolically.
- Example: Solve for time in free fall without manually rearranging:

3. Handles Complex Physics Equations

- Example: Solving Lagrangian mechanics equations symbolically.

Example Application

Scenario: Solving a Free-Fall Motion Problem

User Input: *An object is dropped from a height of 50m. Find the time it takes to reach the ground*

- **AI Workflow:**
  - BERT classifies it as a kinematics problem.
  - SymPy derives the equation:

$$h = \frac{1}{2}gt^2 \tag{5.34}$$

    Where:
    * h is the height (displacement).
    * g is the acceleration due to gravity.
    * t is the time elapsed.
  - SymPy solves for t: To solve for t in the equation:

$$50 - \frac{1}{2} \cdot 9.8 \cdot t^2 = 0 \tag{5.35}$$

- : **Output:** $t =$ *3.19 seconds*

Impact on Model Performance

Task	SymPy Accuracy	Traditional Solvers
Algebraic Manipulation	100%	85%
Equation Derivation	92%	80%
Differentiation & Integration	89%	78%
Computational Efficiency	High	Medium

Table 5.6: Comparison of SymPy Accuracy with Traditional Solvers

SymPy outperforms traditional solvers by preserving symbolic accuracy. Ensures physics equations remain in exact form before numerical computation.

Key Takeways

Performs symbolic algebra, differentiation, and integration. Ensures exact equation solving before numerical evaluation. Automatically derives and rearranges physics equations. Verifies AI-generated physics solutions for correctness. Supports real-time equation simplification for Matplotlib graphing.

5.8 Shap-E

5.8.1 Core Functionality

Shap-E is an advanced 3D generative model developed by OpenAI that converts textual descriptions or 2D images into high-fidelity 3D object representations. Unlike traditional mesh-based modeling tools like Blender, Shap-E directly learns a neural implicit representation of 3D objects, making it an ideal choice for physics-based simulations, 3D visualization of scientific concepts, and diagram generation.

5.8.2 Why Use Shap-E in AI Physics Problem Solving?

1. Generates accurate 3D models of physics objects based on problem statements.
2. Creates visual representations of forces, motions, and fields.
3. Facilitates interaction with 3D simulations (e.g., visualizing gravitational fields, motion trajectories).
4. Works seamlessly with Blender CLI and PyBullet for simulation and rendering.

Shap-E is used in the AI pipeline to:

1. Generate 3D diagrams of physical objects (e.g., pendulums, projectiles, inclined planes).
2. Visualize abstract physics concepts in three dimensions.
3. Augment simulations by providing realistic 3D models as inputs to Blender and PyBullet.

5.8.3 Mathematical Formulations

Shap-E represents 3D objects as implicit neural representations rather than traditional mesh-based models.

1. Implicit Neural Representations Unlike mesh-based models (where 3D objects are defined by vertices and edges), Shap-E represents a 3D object as a continuous function:

$$f(x, y, z) = s \tag{5.36}$$

   Where:

   - x, y, z represents spatial coordinates
   - $s$represents the signed distance function (SDF) that determines whether a point is inside, outside, or on the surface of the object.

   Shap-E learns this function using diffusion models, ensuring:

   - High accuracy in object structure.
   - Better generalization across diverse object types.
   - Efficient 3D reconstruction from minimal input data.

2. Text-to-3D Model Conversion Shap-E processes textual inputs using latent diffusion models:

   - Encodes the text prompt into a latent representation:

$$z = E(T) \tag{5.37}$$

     Where E is the text text encoder and T is the input description (e.g., "a simple pendulum swinging in a gravitational field").
   - Uses a conditional generative model to transform the latent vector into a neural implicit representation:

$$O = G(z) \tag{5.38}$$

     Where G is the generator that produces the 3D object O.
   - The generated object is then converted into a mesh for rendering:

$$M = D(O) \tag{5.39}$$

     Where D is a decoder that extracts the 3D mesh M.

3. Integration with Blender CLI and PyBullet Once Shap-E generates a 3D object:

   - The object is exported to Blender CLI for high-resolution rendering.
   - The object can be simulated in PyBullet for collision physics, rigid-body dynamics, and motion analysis.

   Example workflow for a projectile motion simulation:

   - AI identifies the need for a 3D model (e.g., a ball launched at an angle).
   - Shap-E generates a 3D sphere based on the input.
   - The 3D object is fed into PyBullet, where it is launched at an angle.
   - The motion trajectory is computed and visualized.

5.8.4 Implementation in the Model Workflow

Shap-E is used whenever a physics problem requires 3D visualization.

1. Identifying When a 3D Model Is Needed
   - AI determines if the problem involves 3D motion, force interactions, or geometric representations.
   - If 2D visualization is insufficient, the Shap-E pipeline is triggered.
2. Generating the 3D Object
   - Physics object (e.g., pendulum, inclined plane, force vectors) is generated.
   - Implicit neural representation is converted into a renderable format.
3. Exporting to Simulation Engines
   - The 3D model is loaded into Blender for rendering.
   - The model is tested in PyBullet for real-world physics interactions.
4. Fine-Tuning for Accuracy
   - If refinements are needed, Shap-E retrains the object representation to improve shape accuracy.

5.8.5 How Shap-E Enhances Accuracy

1. Physics-Based Object Generation
   - Unlike generic 3D models, Shap-E understands physics constraints.
   - Generates realistic 3D representations of mechanical systems.
2. Adaptability Across Physics Topics
   - Can generate electromagnetic fields, gravitational wells, force diagrams, and wave patterns.
   - Example: If AI needs a 3D visualization of an electric dipole, Shap-E creates a field line representation in 3D space.
3. Seamless Integration with AI-Generated Explanations
   - Pairs with FLAN-T5 for interactive tutoring.
   - Pairs with PyBullet for motion simulation.

Example Application

Scenario: Visualizing a Rotating Pendulum in 3D

User Input: *Generate a 3D simulation of a pendulum oscillating in a gravitational field.*

- **AI Workflow**:
  - BERT classifies it as a pendulum motion problem.
  - Shap-E generates the pendulum model:
    * Creates a spherical bob and string constraint.
    * Assigns physical properties (mass, length, pivot point).
  - Model is loaded into PyBullet for real-time physics simulation.
  - Rendered in Blender CLI for high-quality visualization.

Impact on Model Performance

Task	Shap-E Accuracy	Traditional 3D Modeling
Physics Object Generation	88%	85%
Neural Representation Consistency	90%	78%
Integration with AI Physics Pipeline	High	Moderate
Computational Efficiency	Medium	Low

Table 5.7: Comparison of Shap-E Accuracy with Traditional 3D Modeling

Shap-E significantly improves physics problem visualization by generating accurate 3D models dynamically. Reduces human effort in creating 3D visual representations of physics problems. Works seamlessly with simulation engines for interactive AI tutoring.

Key Takeaways

Generates accurate 3D physics objects from textual descriptions. Uses neural implicit representation for high-fidelity model generation. Integrates with Blender and PyBullet for realistic simulations. Improves AI-driven tutoring by adding interactive 3D visuals. Enables visualization of abstract physics concepts (fields, forces, and motion).

5.9 Overall Comparison and Model Selection Justification

The AI-driven physics problem-solving system integrates multiple models from **natural language processing (NLP), mathematical solvers, and visualization tools** to achieve high accuracy, interpretability, and efficiency. Each model was carefully chosen based on its strengths in handling different types of tasks.

5.9.1 Comparison of Models

Model	Category	Primary Purpose	Key Features
BERT	NLP	Problem classification & entity extraction	Context-aware understanding
FLAN-T5	NLP	Step-by-step physics solution generation	Instruction-tuned, structured solving
LLaMa	NLP	Interactive tutoring and reasoning	Conversational learning
Stable Diffusion	Visualization	Generating physics diagrams	AI-generated schematic representations
Pix2Pix	Image Processing	Fine-tuning diagrams generated by Stable Diffusion	Edge-aware image transformation
CycleGAN	Image Processing	Enhancing physics diagram realism	Domain adaptation without paired images
ControlNet	Image Processing	Ensuring structural accuracy in generated diagrams	Adds structural constraints
Blender CLI	Simulation	3D physics-based visualizations	High-quality 3D rendering
PyBullet	Simulation	Real-time physics simulations	Rigid-body dynamics, collision detection
Manim	Visualization	Stepwise mathematical animations	2D/3D equation-based animations
Matplotlib	Visualization	Graphing kinematics & forces	Real-time graphing
SymPy	Mathematics	Symbolic computation & equation solving	Exact algebraic manipulation
Shap-E	Visualization	AI-generated 3D physics object creation	Neural implicit 3D modeling

Table 5.8: Comparison of AI Models Used in Physics Problem-Solving

5.9.2 Why Each Model Was Used for a Specific Task

Natural Language Processing (NLP) Models

NLP models process textual physics problem statements, generate structured solutions, and provide interactive tutoring.

- **BERT: Problem Understanding and Classification**
  - Extracts key physics variables (mass, force, velocity, etc.).
  - Classifies problems into physics categories like kinematics, electromagnetism, and thermodynamics.
- **FLAN-T5: Step-by-Step Solution Generation**
  - Selects appropriate physics formulas.
  - Provides structured step-by-step solutions.
- **LLaMa: Interactive Physics Tutoring**
  - Engages in interactive Q&A sessions.
  - Improves explanations through user interaction.

Mathematical Computation and Symbolic Processing

These models perform symbolic algebra, equation solving, and graphing.

- **SymPy: Symbolic Equation Solver**
  - Solves algebraic and calculus-based equations symbolically.
  - Computes derivatives and integrals for motion problems.
- **Matplotlib: Graphing Kinematics & Forces**

- Plots motion equations in real-time.
- Used for displacement-time, velocity-time, and force-time graphs.

Image Processing Models for Physics Diagrams

These models enhance AI-generated physics diagrams for accuracy and clarity.

- **Stable Diffusion: AI-Generated Physics Diagrams**
  - Creates rough physics-related diagrams from textual descriptions.
  - Automatically generates force diagrams, electric fields, etc.
- **Pix2Pix: Diagram Refinement**
  - Enhances AI-generated physics diagrams by improving edge clarity.
  - Converts rough AI-generated sketches into refined physics visuals.
- **CycleGAN: Domain Adaptation**
  - Improves realism of AI-generated physics illustrations.
  - Useful for refining complex diagrams (e.g., wave propagation, electric fields).
- **ControlNet: Structural Accuracy Enforcement**
  - Ensures AI-generated diagrams follow scientific constraints.
  - Prevents hallucinated artifacts in force and motion diagrams.

Visualization and Simulation Models

These models handle motion simulation, vector visualization, and 3D rendering.

- **Blender CLI: 3D Physics Simulations**
  - Best for high-resolution 3D animations of physics experiments.
  - Handles soft-body and rigid-body dynamics.
- **PyBullet: Real-Time Physics Simulation**
  - Best for dynamic object interaction simulations (collisions, rolling motion, impacts).
  - Simulates real-time Newtonian physics.
- **Manim: Mathematical Animation Engine**
  - Visualizes physics derivations and explanations.
  - Ideal for illustrating transformations of energy and force.
- **Shap-E: AI-Based 3D Model Generator**
  - Creates 3D representations of physics objects.
  - Enables visualization of complex physics experiments.

5.9.3 Comparison of Models Based on Performance

Model	**Accuracy** (Higher is Better)	**Efficiency** (Higher is Faster)	**Real-Time Performance**
BERT	90%	4/5	5/5
FLAN-T5	89%	3/5	4/5
LLaMa	92%	3/5	5/5
Stable Diffusion	91%	2/5	2/5
Pix2Pix	87%	4/5	3/5
CycleGAN	85%	3/5	3/5
ControlNet	89%	2/5	3/5
Blender CLI	90%	1/5	1/5
PyBullet	94%	4/5	5/5
Manim	90%	3/5	3/5
Matplotlib	92%	5/5	5/5
SymPy	93%	4/5	4/5
Shap-E	89%	2/5	2/5

Table 5.9: Model Comparison Based on Accuracy, Efficiency, and Real-Time Performance

5.10 Dataset Overview

The dataset comprises three major components:

1. **Physics Problems Dataset**: A structured collection of physics problems categorized based on difficulty levels, concepts covered, and solution methodologies.
2. **Diagram Description Dataset**: Contains textual descriptions of 3D diagrams designed to illustrate physical phenomena.
3. **Simulation Description Dataset**: Provides textual descriptions of simulation videos, detailing how different physics principles are animated.

Sample Data Entry

Attribute	**Example Entry**
Problem Statement	A ball is thrown vertically upward with an initial velocity of 20 m/s. Calculate the maximum height reached.
Concepts Covered	Kinematics, Free Fall, Acceleration due to Gravity
Difficulty Level	Easy
Explanation	The ball decelerates due to gravity until its velocity becomes zero at the highest point. The motion is governed by kinematic equations.
Constants Required	Acceleration due to gravity, $g = 9.81$ m/s^2
Given Values	Initial velocity: $u = 20$ m/s, Final velocity: $v = 0$ m/s
Formulas Used	$v^2 = u^2 + 2as$
Final Answer	$s = 20.4$ m
3D Diagram Description	Show a vertical trajectory of the ball with markers at the starting point and maximum height.
Simulation Description	Animate the ball's motion from throw to peak and back to the initial level.
Tags	Projectile motion, Free Fall, Kinematics

Table 5.10: Sample Data Entry from Physics Problems Dataset

5.10.1 Diagram Description Dataset

This dataset contains textual descriptions of physics diagrams, which serve as visual aids for understanding physical concepts. Each entry consists of:

- **Index**: Unique identifier for the diagram.
- **3D Diagram Description**: A textual description of the expected visual representation.

Sample Diagram Descriptions

Index	3D Diagram Description
0	Show a vertical trajectory of the ball with markers at the starting point and maximum height.
1	Work is calculated as the product of force, distance, and the cosine of the angle between force and displacement.
2	A 3D diagram showing heat transfer between ice and water, with arrows indicating the direction of heat flow.
3	Depict a charge and field lines radiating outward, with a point marked at 5 cm distance.

Table 5.11: Descriptions of 3D Diagrams

5.10.2 Simulation Description Dataset

The simulation dataset contains textual descriptions of animations that visually represent physics problems. Each simulation entry includes:

- **Index**: A unique identifier.
- **Simulation Description**: A textual explanation of the animation sequence.

Sample Simulation Descriptions

Index	Simulation Description
0	Animate the ball's motion from the throw to the peak and back to the initial level.
1	A simulation video showing the block moving across a surface with force vectors and real-time work done calculations.
2	A simulation video showing the melting of ice and the temperature change in the water, including real-time temperature monitoring.
3	Visualize the static electric field and show field strength variation with distance.

Table 5.12: Descriptions of Sample Simulations

This dataset is designed to facilitate AI-driven physics problem-solving by integrating problem descriptions, diagram generation, and simulation visualization. The combination of structured problems, textual representations of diagrams, and simulation sequences ensures a comprehensive approach to understanding physics.

Chapter 6

Result and Analysis

6.1 Results and Analysis

6.1.1 Model Performance Evaluation

Model	Explanation Accuracy	Classification Accuracy	Simulation Accuracy
FLAN-T5	88%	N/A	85%
BERT	N/A	92%	N/A
LLaMa	82%	N/A	87%

Table 6.1: Model Performance Evaluation

6.1.2 Comparison of Simulation Techniques

Simulation Pipeline	Accuracy	Clarity	Computational Cost
FLAN-T5 + ControlNet + Stable Diffusion + Blender CLI	82%	High	High
FLAN-T5 + ControlNet + Stable Diffusion + PyBullet	85%	High	Medium
FLAN-T5 + ControlNet + Stable Diffusion + Manim	87%	High	Low
FLAN-T5 + Stable Diffusion + Pix2Pix + CycleGAN + ControlNet + Blender CLI	88%	High	High
FLAN-T5 + Stable Diffusion + Pix2Pix + CycleGAN + ControlNet + PyBullet	90%	Very High	Medium
FLAN-T5 + Stable Diffusion + Pix2Pix + CycleGAN + ControlNet + Manim	94%	Very High	Low
LLaMa + ControlNet + Stable Diffusion + Blender CLI	80%	High	High
LLaMa + ControlNet + Stable Diffusion + PyBullet	83%	High	Medium
LLaMa + ControlNet + Stable Diffusion + Manim	85%	High	Low
LLaMa + Stable Diffusion + Pix2Pix + CycleGAN + ControlNet + Blender CLI	86%	High	High
LLaMa + Stable Diffusion + Pix2Pix + CycleGAN + ControlNet + PyBullet	88%	High	Medium
LLaMa + Stable Diffusion + Pix2Pix + CycleGAN + ControlNet + Manim	92%	Very High	Low

Table 6.2: Comparison of Simulation Techniques

6.1.3 Observations and Future Improvements

Observations

- Manim-based simulations produced the most efficient and visually clear results, especially for step-by-step mathematical animations.
- PyBullet performed well in real-time physics simulations but required optimization to reduce computational cost.

- Blender CLI generated the highest-quality 3D visualizations but at a significantly higher computational expense.
- CycleGAN and Pix2Pix refinements improved the clarity of generated diagrams, making Stable Diffusion outputs more usable.

Future Improvements

- Reduce computational costs in Blender CLI by optimizing rendering settings.
- Integrate hybrid simulation techniques to dynamically switch between PyBullet and Manim for different problem scenarios.
- Improve real-time response in interactive tutoring by optimizing LLaMa's processing pipeline.
- Extend dataset training for image processing models to cover more diverse physics diagrams for better AI-generated visualization.

This refined comparison highlights the effectiveness of different simulation pipelines, their trade-offs, and future enhancement strategies for a more robust AI-driven physics problem-solving system.

Chapter 7

Conclusion

In ”Elevating Education with AI: Augmenting the Understanding of Physics through Topic Prediction, Three-Dimensional Visualisation, and Dynamic Video Aids,” the thesis explores the relationship between artificial intelligence (AI) and education with a particular emphasis on physics. This research investigates the potential of AI to not only forecast and clarify important physics concepts but also transform the learning process through visualization and dynamic video aids, at a time when technology advancement and pedagogical evolution are converging. When we consider the results that have been presented, it is clear that using technology to improve education is not just a theoretical idea, but also a practical possibility that has the power to completely change the way people learn. This thesis seeks to provide important insights into the future of learning, where artificial intelligence will work as a catalyst for increased understanding and engagement in the field of physics education by shedding light on the many facets of AI's influence on education.

Bibliography

[1] A. M. Turing, *The essential Turing : seminal writings in computing, logic, philosophy, artificial intelligence, and artificial life, plus The secrets of enigma*, B. J. Copeland, Ed. Clarendon Press, 2004.

[2] C. Bradley, J. Allred, and M. Zeidan, "Visualizing physics questions," *International Conference On Advanced Research in EDUCATION*, pp. 1–9, Mar. 2019. [Online]. Available: https://www.dpublication.com/wp-content/uploads/2019/06/EDUCATIONCONF-1-135-F.pdf.

[3] R. R. Ayu, "The need analysis of the development of animation video-based learning media on physics education students," *Asian Journal of Healthy and Science*, vol. 1, 2 Nov. 2022. DOI: https://doi.org/10.58631/ajhs.v1i2.10. [Online]. Available: https://www.researchgate.net/publication/368285654_The_Need_Analysis_Of_The_Development_Of_Animation_Video-Based_Learning_Media_On_Physics_Education_Students.

[4] P. Purwanti, "The effect of learning approaches and learning creativity on the ability to solve physic problems," *Journal of Academia Perspectives*, vol. 1, no. 1, 1 Jan. 2021. DOI: https://doi.org/10.30998/jap.v1i1.385. [Online]. Available: https://www.researchgate.net/publication/365508086_The_Effect_Of_Learning_Approaches_And_Learning_Creativity_On_The_Ability_To_Solve_Physic_Problems.

[5] A. Ayoub, L. Afifi, and A. Zumwalt, "Integrating 3d animated videos in case-based study of cranial nerve pathology," *The FASEB Journal*, vol. 34, no. 1, pp. 1–1, S1 Apr. 2020. DOI: http://dx.doi.org/10.1096/fasebj.2020.34.s1.05427. [Online]. Available: https://www.researchgate.net/publication/340748893_Integrating_3D_Animated_Videos_in_Case-Based_Study_of_Cranial_Nerve_Pathology.

[6] S. Di, L. Cao, and Y. Ge, "Visualization methods of 3d plant models: A systematic mapping study," *Journal of Electrical and Computer Engineering*, vol. 2021, pp. 1–19, 1 Sep. 2021. DOI: http://dx.doi.org/10.1155/2021/2754343. [Online]. Available: https://www.researchgate.net/publication/354482638_Visualization_Methods_of_3D_Plant_Models_A_Systematic_Mapping_Study.

[7] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, "Cogvideo: Large-scale pretraining for text-to-video generation via transformers," May 2022. [Online]. Available: https://arxiv.org/pdf/2205.15868.pdf.

[8] P.-c. Hsu, D.-r. Liu, A. T. Liu, and H.-y. Lee, "Parallel synthesis for autoregressive speech generation," Apr. 2022. [Online]. Available: https://www.researchgate.net/publication/360185806_Parallel_Synthesis_for_Autoregressive_Speech_Generation.

[9] W. Jackson, M. Yamada, T. O. Moninger, and C. Grose, "Visualization and quantitation of abundant macroautophagy in virus-infected cells by confocal three-dimensional fluorescence imaging," *Journal of Virological Methods*, vol. 193, pp. 244–250, 2013 Jun. 2013. DOI: http://dx.doi.org/10.1016/j.jviromet.2013.06.018. [Online]. Available: https://www.researchgate.net/publication/241692968_Visualization_and_quantitation_of_abundant_macroautophagy_in_virus-infected_cells_by_confocal_three-dimensional_fluorescence_imaging.

[10] W. Osamy, A. M. Khedr, A. Salim, A. I. AlAli, and A. A. El-Sawy, "Recent studies utilizing artificial intelligence techniques for solving data collection, aggregation, and dissemination challenges in wireless sensor networks: A review," *electronics*, vol. 11, p. 313, 3 Jan. 2022. DOI: https://doi.org/10.3390/electronics11030313. [Online]. Available: https://www.mdpi.com/2079-9292/11/3/313.

[11] V. Liu, T. Long, N. Raw, and L. Chilton, "Generative disco: Text-to-video generation for music visualization," Apr. 2023. [Online]. Available: https://www.researchgate.net/publication/370103269_Generative_Disco_Text-to-Video_Generation_for_Music_Visualization.

[12] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," Jun. 2023. DOI: http://dx.doi.org/10.1109/CVPR52729.2023.02155. [Online]. Available: https://www.researchgate.net/publication/373308424_DreamBooth_Fine_Tuning_Text-to-Image_Diffusion_Models_for_Subject-Driven_Generation.

[13] S. Yin, C. Wu, J. Liang, *et al.*, "Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory," Aug. 2023. [Online]. Available: https://www.researchgate.net/publication/373163483_DragNUWA_Fine-grained_Control_in_Video_Generation_by_Integrating_Text_Image_and_Trajectory.

[14] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, *et al.*, "Phenaki: Variable length video generation from open domain textual description," Oct. 2022. [Online]. Available: https://www.researchgate.net/publication/364194403_Phenaki_Variable_Length_Video_Generation_From_Open_Domain_Textual_Description.

[15] C. Wu, J. Liang, X. Hu, *et al.*, "Nuwa-infinity: Autoregressive over autoregressive generation for infinite visual synthesis," Jul. 2022. [Online]. Available: https://arxiv.org/abs/2207.09814.

[16] A. V. Riabko and T. A. Vakaliuk, "Physics on autopilot: Exploring the use of an ai assistant for independent problem-solving practice," *Educational Technology Quarterly*, vol. 2024, Jan. 2024. DOI: http://dx.doi.org/10.55056/etq.671. [Online]. Available: https://www.researchgate.net/publication/377134675_Physics_on_autopilot_exploring_the_use_of_an_AI_assistant_for_independent_problem-solving_practice.

[17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 6 May 2021. DOI: https://doi.org/10.1038/s42254-021-00314-5. [Online]. Available: https://www.nature.com/articles/s42254-021-00314-5.

[18] T. X. Nghiem, J. Drgoňa, C. Jones, *et al.*, "Physics-informed machine learning for modeling and control of dynamical systems," Jun. 2023. DOI: https://doi.org/10.48550/arXiv.2306.13867. [Online]. Available: https://arxiv.org/abs/2306.13867.

[19] T. Rios, S. Menzel, and B. Sendhoff, "Large language and text-to-3d models for engineering design optimization," *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, Jan. 2024. DOI: https://doi.org/10.1109/SSCI52147.2023.10371898. [Online]. Available: https://ieeexplore.ieee.org/document/10371898.

[20] V. Maphosa and M. Maphosa, "Artificial intelligence in higher education: A bibliometric analysis and topic modeling approach," *Applied Artificial Intelligence*, vol. 37, p. 2358, 1 Oct. 2023. DOI: http://dx.doi.org/10.1080/08839514.2023.2261730. [Online]. Available: https://www.researchgate.net/publication/374820289_Artificial_intelligence_in_higher_education_a_bibliometric_analysis_and_topic_modeling_approach.

[21] H. Jun and A. Nichol, "Shap-e: Generating conditional 3d implicit functions," May 2023. DOI: https://doi.org/10.48550/arXiv.2305.02463. [Online]. Available: https://arxiv.org/abs/2305.02463.

[22] J. Xu, X. Wang, W. Cheng, *et al.*, "Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models," Aug. 2023. DOI: https://doi.org/10.1109/CVPR52729.2023.02003. [Online]. Available: https://ieeexplore.ieee.org/document/10204065.

[23] G. Kim and S. Y. Chun, "Datid-3d: Diversity-preserved domain adaptation using text-to-image diffusion for 3d generative model," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Aug. 2023. DOI: https://doi.org/10.1109/CVPR52729.2023.01365. [Online]. Available: https://ieeexplore.ieee.org/document/10205118.

[24] Y. Wang, P. Ren, M. Zhou, W. Shui, and P. Zhou, "Text to 3d model of chinese ancient architecture," *2018 International Conference on Cyberworlds (CW)*, Dec. 2018. DOI: https://doi.org/10.1109/CW.2018.00035. [Online]. Available: https://ieeexplore.ieee.org/document/8590030.

[25] J. Z. Wu, Y. Ge, X. Wang, *et al.*, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," Dec. 2022. [Online]. Available: https://www.researchgate.net/publication/366527696_Tune-A-Video_One-Shot_Tuning_of_Image_Diffusion_Models_for_Text-to-Video_Generation.

[26] H. Nasbey and R. Raihanati, "Developing a video education on the topic of modern physics based on problem based learning (pbl) assisted phet online learning," *Journal of Physics Conference Series*, vol. 2377(1):012067, pp. 1–6, Nov. 2022. DOI: http://dx.doi.org/10.1088/1742-6596/2377/1/012067. [Online]. Available: https://www.researchgate.net/publication/365754824_Developing_a_Video_Education_on_the_topic_of_Modern_Physics_Based_on_Problem_Based_Learning_PBL_assisted_PhET_Online_Learning.

[27] P. Tschisgale, P. Wulff, and M. Kubsch, "Integrating artificial intelligence-based methods into qualitative research in physics education research: A case for computational grounded theory," *Physical Review Physics Education Research*, vol. 19, no. 0123, p. 24, 02 2023. DOI: https://doi.org/10.1103/PhysRevPhysEducRes.19.020123. [Online]. Available: https://www.researchgate.net/publication/373612289_Integrating_artificial_intelligence-based_methods_into_qualitative_research_in_physics_education_research_A_case_for_computational_grounded_theory.

[28] J. Xing, M. Xia, Y. Liu, *et al.*, “Make-your-video: Customized video generation using textual and structural guidance,” Jun. 2023. [Online]. Available: https://www.researchgate.net/publication/371223250_Make-Your-Video_Customized_Video_Generation_Using_Textual_and_Structural_Guidance.