

# 생성형 인공지능 기반 실시간 3D 물리 시뮬레이션 콘텐츠 생성: 직선운동 문제 사례를 중심으로<sup>1</sup>

박재우<sup>0</sup>      이용선      이택희\*

(주)웨일텍, 한국공학대학교

(jjj901913, heebung@whaletek.net, watersp@tukorea.ac.kr)

## Generative AI-Based Real-Time 3D Simulation Content Generation for Physics Education: Focused on Linear Motion Problems

Jaewoo, Park      Yongsun, Lee      Taekhee, Lee\*

Whaletek, Tech University of Korea

### 요약

본 연구는 생성형 인공지능 중 하나인 GPT 기반 모델을 활용하여 이미지로 저장된 과학 문제를 자동으로 학습용 실시간 3D 과학 콘텐츠를 생성하는 기법을 제안한다. 콘텐츠는 미리 생성된 3D 과학 실험 모델링 데이터와 시뮬레이터, 그리고 인공지능으로부터 생성된 결과를 활용하여 생성된다. 이 과정에서 인공지능 활용의 핵심은 이미지를 분석하고 시뮬레이터에 필요한 파라미터를 자동으로 추출하는 것이다. 인공지능이 적용되지 않은 방법은 사용자가 문제 조건을 해석하고 파라미터를 직접 입력해야 하는 불편함이 있으며, 이로 인해 학습 과정에서의 오류나 비효율이 발생할 수 있다. 제안된 시스템은 과학 문제의 이미지를 분석 후, 핵심 조건들을 추출하여 시뮬레이터에 필요한 위치, 속도 등의 파라미터를 자동으로 구성함으로써 문제 상황에 적합한 실시간 3D 시뮬레이션을 생성한다. 이를 통해 학습자는 시각적이고 직관적인 방식으로 문제를 이해할 수 있으며, 문제 풀이에 대한 몰입도와 효율성을 높일 수 있다. 본 연구는 고등학교 물리학1 교과서 기반의 문항을 활용하여 실제 문제 상황에 맞는 시뮬레이터가 자동 생성될 수 있음을 확인하였다. 이러한 접근은 생성형 AI의 활용 가능성을 실시간 3D 시뮬레이션 기반 학습으로 확장하는 새로운 사례로 볼 수 있다.

### Abstract

This study proposes a method for automatically generating real-time 3D science learning content using GPT-based generative artificial intelligence. The content is constructed by combining pre-built 3D science experiment models, a simulation engine, and outputs generated by the AI. The core function of the AI in this process is to analyze images of science problems and automatically extract the parameters required for the simulator. In conventional methods without AI support, users must manually interpret problem conditions and input parameters, which can lead to inefficiencies and errors during the learning process. The proposed system analyzes images of science problems, extracts key conditions, and automatically generates parameters such as positions and velocities required for the simulator, thereby creating real-time

<sup>1</sup> 학부생 주저자 논문임

\*corresponding author: Taekhee, Lee/ Tech University of Korea (watersp@tukorea.ac.kr)

3D simulations tailored to the problem scenario. This allows learners to understand the problems in a more visual and intuitive manner, enhancing engagement and learning efficiency. The proposed method was tested using physics problems from high school Physics I textbooks, and it was confirmed that the simulator can be automatically generated in accordance with the given scenarios. This approach demonstrates the potential of generative AI in expanding into real-time 3D simulation-based learning and is expected to contribute to the development of diverse science education content in the future.

**키워드:** 생성형 인공지능, GPT, 실시간 3D 과학 콘텐츠, 실시간 3D 시뮬레이션

**Keywords:** generative artificial intelligence, GPT, real-time 3D science learning content, real-time 3D simulation

## 1. 서론

최근 인공지능(AI) 기술은 비약적인 발전을 거듭하며 다양한 산업 분야에 도입되고 있다. 특히 GPT(Generative Pre-trained Transformer)기반으로 개발된 ChatGPT의 등장은 AI 기술의 발전에 있어 중요한 변화를 이끌어 내고 있다. ChatGPT는 대규모 사전 학습을 통해 자연어를 처리할 수 있는 능력을 갖춘 모델로, 대화형 AI 형태로 서비스를 제공한다[1]. 초기에는 텍스트 생성 및 이해에 그쳤으나, 이미지 분석과 생성, 그리고 동영상 생성까지 기술적 범위를 확장하고 있으며 이에 따른 다양한 AI 서비스가 등장하고 있다. 이러한 흐름 속에서 AI는 자료 요약, 데이터 분석, 프로그래밍 등 다양한 작업의 효율성을 높이며, 생활, 의료, 산업 등 여러 분야에 적용하기 위한 연구가 진행 중이다[2][3]. 교육 분야에서도 AI 기술의 도입은 이루어지고 있지만, 현재까지의 대부분의 활용 사례는 학습 방향을 잡아주거나 진로를 컨설팅 하는 역할에 머무르고 있다. 즉, 학습자의 학습 그 자체를 돕는 도구로서 AI를 직접적으로 활용하는 시도는 상대적으로 부족하다[4].

수동으로 제작되는 학습용 물리 콘텐츠의 경우, 기존에는 사용자가 문제의 조건을 직접 해석하고, 파라미터를 수동으로 입력하여 시뮬레이터를 생성하고 실행해야 하는 번거로움이 있었다. 이 과정에서 사용자가 문제를 제대로 이해하지 못하면 시뮬레이터의 설정이나 결과 해석에 오류가 발생할 수 있어 학습 효율 저하가 될 우려가 있다. 본 연구는 이러한 한계를 극복하기 위해 GPT API를 활용한 학습용 물리 시뮬레이터 자동 생성 시스템을 제안한다. 제안된 시스템은 직선운동 문제 사례에 한정된 물리 문제의 이미지를 GPT API를 통해 분석하고, 문제 이해를 바탕으로 시뮬레이터 생성에 필요한 핵심 파라미터를 자동으로 생성한다. 생성된 파라미터는 미리 제작된 3D 모델링 데이터에 자동으로 적용되며 이를 통해 사용자는 별도의 파라미터 설정 없이도 즉각적으로 실시간 3D 학습용 콘텐츠를 구동할 수 있다. 더 나아가 학습자가 문제를 해석하는 초기 단계에서의 오류를 방지하고 학습 효율성을

높일 것으로 기대된다. 완성된 시뮬레이터는 실시간 3D 콘텐츠 형태이며 이러한 형태는 문제에 설정된 수치를 변화시켜 가면서 학습 이해도를 높이는 등의 목적으로 활용될 수 있다.

## 2. 선행 연구

생성형 인공지능은 등장 이후 교육 분야 전반에서 빠르게 적용되고 있다. 수업 준비, 학습 활동, 평가에 이르기까지 교사와 학습자의 역할을 보조하거나 확장할 수 있는 결과물을 생성해 주기 때문이다.

교육 현장에서 ChatGPT를 활용하는 주요 방식 중 하나는 수업 설계의 보조 도구로 활용하는 것이다. 기존에는 교사가 일일이 구상하던 수업 내용이나 문제 제작 과정이 생성형 AI를 통해 자동화 및 고도화되면서 업무 부담을 줄일 수 있다. 예컨대, 중등 영어 교사들을 대상으로 한 연구에서는 ChatGPT가 수능형 문항의 지문 생성, 어휘 난이도 조절, 보기에 대한 구조화 등에서 효과적으로 활용되었으며, 참가자들 대부분이 문항 개발의 질적, 양적 향상에 긍정적인 반응을 보였다는 연구결과가 있다[5].

교사 대상의 교육에서도 수업지도안 작성을 위한 협업 과정에 ChatGPT를 도입한 사례가 존재한다. AI가 단순한 도우미를 넘어 협력적 피드백을 제공하는 일종의 동료 교사 역할을 수행했다. 특히 협업 수준이 높을수록 수업 내용의 구조나 논리 흐름이 더욱 정교화되는 경향이 뚜렷하게 나타났으며, 이는 교사의 교육 설계 능력을 향상시킬 수 있다는 연구가 진행되었다[6].

수업 실행 단계에서의 활용을 위한 학생의 학습 몰입도와 맞춤형 피드백 제공을 가능하게 하는 AI 기반 학습 플랫폼에 대한 연구도 진행되었다. 초등학교 대상 영어 수업에 ChatGPT API를 기반으로 개발된 웹 애플리케이션을 도입한 사례에서는, 학생 개개인의 수준에 맞춘 실시간 피드백과 콘텐츠 제공이 이루어졌으며, 결과적으로 학습자 반응성과 교사 만족도 모두에서 긍정적 효과가 확인되었다[7].

고등 과학 교과에서는 ChatGPT의 문제 해결 능력과 그 한계를 검증하려는 연구들이 이루어지고 있다. 대학 수준의 일반물리학 문제를 제시했을 때, 최신 인공지능 모델들은 높은 정답률을 기록하며 개념 이해와 수치 계산에서 우수한 성과를 보였다. 그러나 설명 과정이나 응답 구조에서 여전히 확률적 오류가 나타나기도 하였으며, 이는 AI가 제공하는 해답이 언제나 완전하거나 교육적으로 적합하진 않다는 점을 시사한다[8][9].

학생 평가에 ChatGPT를 적용한 시도도 이루어졌다. 특히 과학 교과의 서술형 문항 채점 과정에 활용되었을 때, AI가 제시한 기준과 모범답안은 일정 수준에서 적절성을 보였지만, 교사 채점 결과와의 일관성은 복잡한 문항일수록 낮아지는 경향을 보였다. 이는 채점 보조 도구로는 활용이 가능하나, 독립적인 평가 판단을 맡기기엔 아직 이르다는 연구가 이루어졌다[10].

AI 기반으로 생성된 학습 도구의 수용 가능성은 기술수용모델(TAM)을 통해 분석되기도 했다. 인지된 유용성, 사용 용이성, 그리고 학습 중 몰입 경험은 ChatGPT에 대한 긍정적 태도와 실제 활용 의도에 유의미한 영향을 미치는 요소로 확인되었으며, 이는 교사나 학습자가 단순히 기술을 이해하는 수준을 넘어 심리적 신뢰감과 효용성을 경험할 때 비로소 기술 채택이 일어남을 의미한다[11].

생성형 AI는 수업 설계 지원 도구와 학습 보조 매체로 양분되어 다양한 교과에 적용되고 있다. 국내에서는 초중등 중심, 국외에서는 대학 중심으로 연구가 진행되고 있으며, AI 활용의 인지 수준 역시 단순 정보 검색에서 창의적 사고 유도까지 폭넓게 확장되고 있다. 하지만 교사의 지도 없이 AI에만 의존할 경우, 학습자의 고차사고 발달이 제한될 수 있다는 점도 지속적으로 제기되고 있다[12].

현장의 실제 목소리를 반영한 실천적 접근도 존재한다. 교사, 학생, 학부모를 대상으로 생성형 AI에 대한 인식을 조사한 결과, 대부분이 이 기술이 미래 교육의 유연성 향상과 정보 활용 역량 강화에 기여할 것으로 기대하고 있으며, 특히 교사의 수업 준비 및 학생의 자기주도학습 도구로서의 유용성이 강조되었다[4].

이처럼 교육 현장에서 생성형 AI가 빠르게 도입되고 있는 흐름은, 단순한 기술 도입을 넘어 우리 사회 전반에 생성형 AI가 미치는 영향력이 크게 증가하고 있음을 보여준다. AI 기술은 교육 분야만이 아니라 산업, 미디어, 일상생활 등 다양한 분야에 걸쳐 인간의 역할과 판단, 그리고 가치 평가의 기준에 영향을 미치며 기존의 틀을 변화시키고 있다. 따라서 교육 분야에서의 AI 활용 논의는 단지 기술의 효율성이나 활용 가능성에 국한되지 않고, 교육의 철학적 방향, 공정성,

학습자의 권리와 같은 근본적인 가치와 함께 논의되어야 한다[13].

앞서 살펴본 바와 같이 다양한 선행연구들은 생성형 인공지능, 특히 ChatGPT가 수업 설계, 학습 활동, 평가 보조 등 교육의 여러 영역에서 일정 수준 이상의 효용성을 지니고 있음을 보여준다. 다만 대부분의 연구가 언어 기반의 콘텐츠 생성이나 문항 설계, 혹은 피드백 제공에 초점을 맞추고 있으며, 비언어적이고 추상적인 개념이 포함된 과학, 수학 등의 영역에서 ChatGPT가 어떻게 학습자의 개념 이해를 도울 수 있는지에 대한 실질적 접근은 아직 미흡한 실정이다. 특히 물리학과 같은 교과에서는 공식, 변수, 조건의 조합에 따라 문제 이해가 크게 달라지며, 학습자의 사고 과정 역시 시각적이고 질차적인 시뮬레이션을 필요로 하는 경우가 많다. 이러한 특성에도 불구하고, 현재까지의 생성형 AI 활용 연구는 물리학 문제를 보다 쉽게 구조화하거나, 문제 풀이 과정을 시각적으로 안내하는 데 직접적으로 적용된 사례가 거의 없는 실정이다.

본 연구는 ChatGPT를 활용하여 사용자가 입력한 물리학 문제로부터 핵심 파라미터를 추출하고, 이를 기반으로 상호작용 가능한 3D 시뮬레이터에 연동되는 구조화된 데이터를 자동 생성하여 콘텐츠를 만들어 주는 시스템을 제안한다. 이 콘텐츠를 통해 학습자가 문제 상황을 시각적으로 이해하고, 직관적 사고를 통해 문제 해결 전략을 구성할 수 있도록 도움을 줄 수 있다. 이러한 접근은 단순한 콘텐츠 생성에 그치지 않고, 언어 기반 AI와 시뮬레이션 기반 학습 도구의 융합을 통해 생성형 AI의 활용 영역을 새로운 방식으로 확장하는 시도로서, 물리 교육에 있어 ChatGPT의 실질적 활용 가능성을 한층 더 구체화하는 데 기여할 수 있을 것이다.

### 3. 시뮬레이션 생성 구조

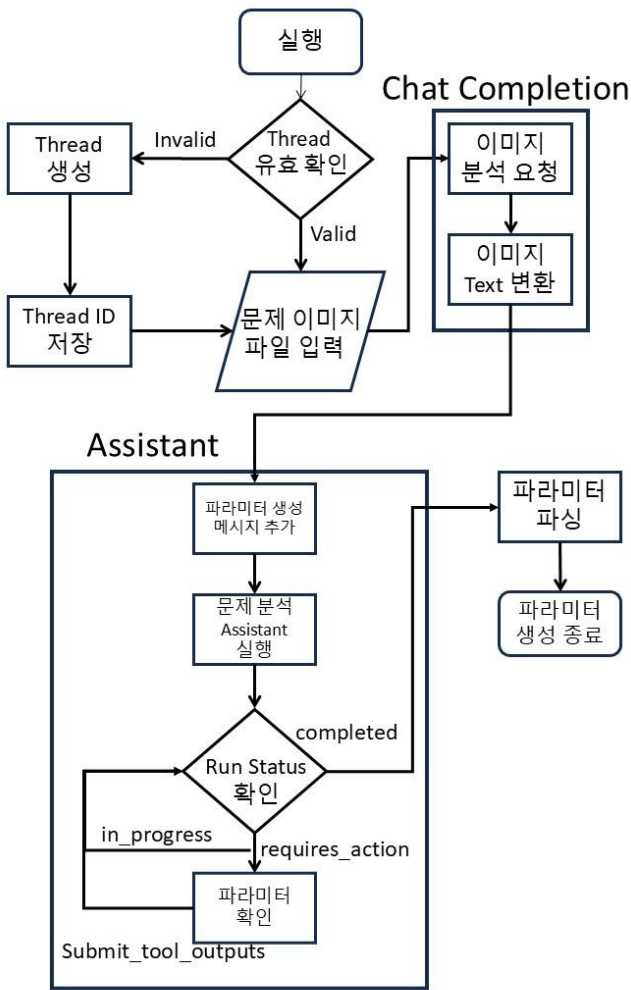


Figure 1. Simulation parameter extraction flow

본 연구에서 구현한 시뮬레이션 생성 시스템은 언리얼 엔진 5.3 을 기반으로 제작되었으며, 전체 흐름은 Figure 3 와 같다. 시뮬레이션 생성 과정은 다음과 같은 단계로 이루어진다. 먼저 GPT API Thread 의 유효성을 검증한 후, 문제 이미지를 분석하고 문제 설명 텍스트를 추출한다. 추출된 텍스트는 Assistant 에 메시지로 추가되고, Assistant 는 이를 기반으로 시뮬레이션 파라미터를 생성한다.

### 3.1 Thread 유효성 판단

Assistant 기능은 대화 상태를 유지하기 위해 Thread 를 기반으로 동작한다. 시뮬레이션 파라미터 생성의 첫 단계는 이 Thread 의 유효성을 검증하는 것이다. 유효성 검증은 GET 방식으로 `api.openai.com/v1/threads/THREAD_ID` 에 요청을 보내 응답 코드가 200 이면 Thread 가 유효하며, 404 이면 새로운 Thread 를 생성해야 한다. 새 Thread 생성은 POST 방식으로 `api.openai.com/v1/threads` 에 요청을 보내며, 응답에서 반환된 Thread ID 를 후속 단계에서 사용한다.

### 3.2 문제 텍스트 추출

유효한 Thread 가 준비되면, GPT API 의 Chat Completion 기능을 통해 문제 이미지를 분석하고 문제 설명 텍스트를 추출한다. 요청은 Post 방식으로 `api.openai.com/v1/chat/completion` 에 요청을 전송해야 하며, 이미지 파일을 포함한 메시지를 전달한다. 이미지는 첨부될 때 사용자 메시지의 content 필드에 텍스트와 이미지가 혼합된 형태의 JSON 객체 배열로 작성된다. 이미지 객체는 type 과 image\_url 로 구성되며 각각 스트링 필드와 오브젝트 필드이다. type 은 "image\_url"이다. image\_url 필드는 `data:<mime>;base64,<image>` 형태의 Data URL 이다. image 부분에 Base64 로 인코딩한 이미지 데이터를 넣고 확장자에 따라 "image/<mime>"을 mime 부분에 넣는다. 만들어진 객체를 메시지의 content 필드에 삽입하여 응답 생성을 요청한다. 요청 처리 완료 후, 사전에 바인딩된 콜백 함수가 호출되고 문제 텍스트가 응답으로 반환된다. 반환된 텍스트는 이후 Assistant 에 전달되어 시뮬레이션 파라미터 생성의 입력으로 사용된다.

### 3.3 파라미터 생성

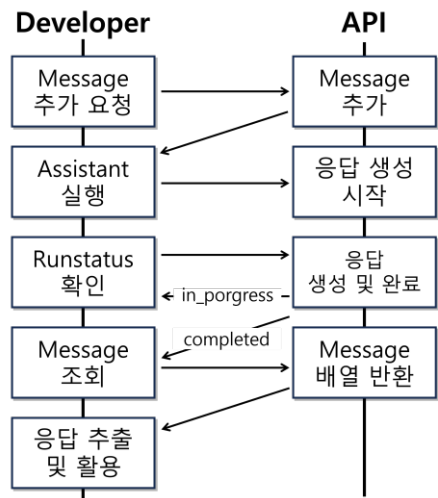


Figure 2. Assistant flow

Figure 2 는 Assistant 기반 시뮬레이션 파라미터 생성 흐름을 구체적으로 보여준다. Chat Completion 과 달리 Assistant 는 단일 요청으로 응답을 생성하지 않고, 메시지 추가, 실행, 응답 생성 등 다단계 요청과 응답 과정을 거친다. 먼저, 문제 텍스트를 POST 방식으로 `api.openai.com/v1/threads/THREAD_ID/messages` 에 요청해 추가한다. 이때 메시지의 role 은 user 로 설정하고 content 에 문제 설명 텍스트를 입력한다. file\_ids 는 참조할 업로드 된 파일 ID 를 지정할 수 있으나, 본 연구에서는 Chat

Completion 에서 이미지 분석을 처리했고 Fine-tuning 의 경우 이미지 처리의 기능은 지원되지 않기 때문에 생략한다. 메시지 추가 후, Assistant 실행을 위해 Post 방식으로 `api.openai.com/v1/threads/THREAD_ID/runs` 에 요청을 보내고 응답에서 `run_id` 를 받는다. Assistant 는 메시지 리스트 내 모든 메시지를 참고하여 응답을 생성한다. 이후 `api.openai.com/v1/threads/THREAD_ID/runs/RUN_ID` 에 Get 방식으로 요청하여 생성 상태를 확인한다. 본 연구에서는 `completed`, `in_progress`, `requires_action` 세 가지 상태를 처리하도록 구현하였다.

상태	설명
completed	답변의 생성이 완전히 끝난 상태.
in_progress	답변이 생성 중인 상태
requires_action	호출되어 사용자의 응답을 기다리는 상태

Table 1. Run status used in generation

Table 1 은 사용하는 각 Run status 의 설명이다. 그중 `requires_action` 상태는 앞서 설명한 Function 기능이 호출된 상태로 이 때 구조화된 파라미터가 반환되며, 이어서 응답을 생성하기 위해서 요청이 필요하다. `api.openai.com/v1/threads/THREAD_ID/runs/RUN_ID/submit_tool_outputs` 에 POST 방식으로 `tool_call_id(requires_action` 응답에 포함)와 `output` 을 전달하면 GPT 가 Function 호출 결과를 반영해 응답을 생성한다. 이 단계에서 실질적으로 시뮬레이션 생성에 필요한 파라미터가 도출되어 역할이 끝나며 `output` 엔 Success 문자열을 포함시켜 작업이 끝났음을 알리고, 이후 `run` 상태가 `completed` 로 전환되면 전체 시뮬레이션 생성 과정이 종료된다. 전 과정은 비동기 처리를 통해 사용자 경험을 유지하면서 진행된다.

## 4. ChatGPT API

ChatGPT API 는 자연어 처리, 이미지 분석, 텍스트 생성 등 다양한 작업을 손쉽게 수행할 수 있도록 제공되는 인터페이스이다.[4] 본 연구에서는 언리얼 엔진 내부에서 HTTP 요청을 통해 GPT API 와 연동하였으며, 생성 방식은 Chat Completion 과 Assistant 방식 두 가지를 병행하여 활용하였다.

### 4.1 Chat Completion

Chat Completion 방식은 매 요청마다 독립적인 메시지 리스트를 서버에 전달하여 응답을 생성하는 구조이다.

사용자는 시스템 메시지(프롬프트)와 사용자 메시지 리스트로 구성하고, 이 리스트를 `context` 로 하여 서버에 요청한다. 서버는 이전 대화 기록을 저장하지 않고 매번 새로운 `context` 만을 참고하여 응답을 생성한다. 따라서 일관된 분석을 위해 매 요청 시 시스템 메시지를 함께 포함시켜야 한다.

본 연구에서는 문제 이미지 분석 단계에서 Chat Completion 방식을 채택하였다. 이유는 Assistant 와 달리, 이전 메시지의 영향을 받지 않기 때문에 이미지 기반 문제 분석의 정확성을 유지할 수 있기 때문이다. 실제 실험에서도 Assistant 를 사용한 경우 Thread 에 쌓인 이전 메시지로 인해 분석 결과가 정확하지 않는 문제가 발생했다. 반면, Chat Completion 은 독립된 요청 구조 덕분에 문제 텍스트 추출의 정확성이 높게 유지되었다.

### 4.2 Assistant

Assistant 방식은 서버 측에서 `context` 을 유지하는 Thread 기반 대화 관리 기능을 제공한다. Thread 는 메시지를 저장하는 컨테이너이며, 여러 메시지를 묶어 하나의 `context` 를 형성한다. 각 메시지는 `role`, `content` 로 구성되며, 필요하다면 추가로 파일을 업로드하여 `file_ids` 를 추가할 수 있다. Assistant 는 Runs 요청을 받으면 Thread 내 `context` 를 기반으로 응답을 생성하여 Thread 에 메시지를 추가하고 사용자는 추가된 메시지를 확인하여 답변을 확인할 수 있다. 이로 인해 사용자와의 상호작용이 지속적이고 일관된 상태로 이어질 수 있으며, 복잡한 문제 처리 및 다단계 요청 처리에 적합하다.

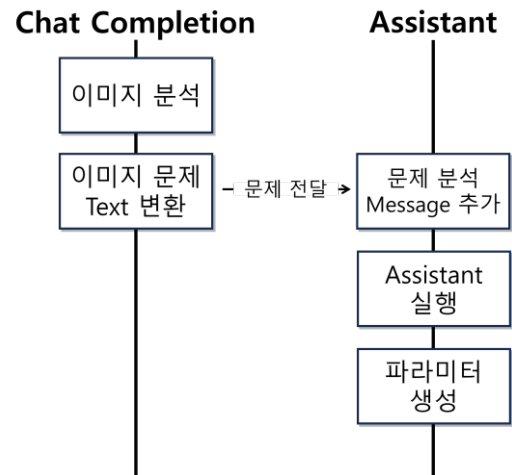


Figure 3. Transfer the question text to Assistant

본 연구에서는 Assistant 를 물리 문제 분석 후 시뮬레이터 생성 파라미터를 추출하는 역할에 활용하였다. Figure 3 에서처럼 이미지 분석 단계를 거쳐 추출된 문제 텍스트를 Assistant 의 Thread 에 입력 메시지로 전달하여 실행하면, Assistant 는 해당 문제를 분석하고 시뮬레이터 생성에 필요한 파라미터(예: 속도, 가속도, 위치 등)를 자동으로 도출한다.

이러한 파라미터는 시뮬레이터 생성 함수에 직접 전달해 객체들을 생성할 수 있다.

### 4.3 Function

Function 기능은 파라미터 생성에 필요한 핵심 기능으로 Assistant 가 단순한 텍스트 응답을 넘어, 호출한 기능(Function)을 통해 구조화된 데이터를 반환하거나 외부 처리 흐름을 연결할 수 있도록 한다. 이를 통해 문제 분석 단계에서 추출된 파라미터를 명확히 구조화하고, 시뮬레이터 생성 엔진과의 연계가 용이하도록 할 수 있다.

Function 기능의 기본 개념은, 사용자가 GPT 에 "이 문제의 조건을 분석하고 필요한 파라미터를 반환해라"라는 요청을 했을 때, GPT 가 단순히 텍스트 설명을 생성하는 대신, 주어진 함수의 정의(스키마)에 맞게 호출 제안을 응답으로 생성한다. 사용자는 Assistant 생성 때 Function 호출을 위해 함수의 정의를 Assistant 에 등록하며, 이 정의에는 호출할 함수 이름, 입력 파라미터 형식 등이 포함된다. GPT 는 context 와 함수의 정의를 바탕으로 해당 함수의 호출 제안을 생성하고, 사용자는 이 제안을 바탕으로 실제 함수를 호출하거나 외부 API 를 이용하는 등 결과를 처리할 수 있다. GPT 자체가 함수 호출을 실행하거나 결과를 반환하지는 않지만, 구조화된 데이터 출력을 위한 중간 역할을 수행한다. 이 기능을 사용하지 않았을 경우 데이터 전 후에 부연 설명이 추가되거나 정확한 값이 필요한 부분에 변수로 대체하는 등의 문제가 있으나 구조화된 데이터 출력으로 이 문제를 완벽하게 보완하여 시뮬레이터 생성의 핵심 기능의 역할을 수행한다.

```
{
  "CarName": "A",
  "StartVelocity": 2,
  "StartDistance": 0,
  "Accel": 1,
  "Reverse": false,
  "Points": [
    {
      "X": 0,
      "Y": 1
    }
  ]
}
{
  "CarName": "B",
  "StartVelocity": 0,
  "StartDistance": 0,
  "Accel": 1,
  "Reverse": false,
  "Points": [
    {
      "X": 0,
      "Y": 1
    }
  ]
}
```

Figure 4. Parameter JSON

Figure 4 는 Function 으로 반환된 응답의 JSON 구조를 보여준다. 반환된 JSON 을 이용해 사용할 수 있는 형태의 데이터로 가공하여 시뮬레이터 객체 생성에 사용한다.

### 4.4 Fine-tuning

GPT 및 기타 대규모 언어 모델(LLM)의 성능은 주로 대규모 범용 데이터셋에 기반한 사전 학습(Pre-training)을 통해 확보된다. 그러나 이러한 범용 모델은 특정 도메인이나 문제 유형에 대한 최적화가 부족할 수 있으며, 특정 작업(예: 특정 유형의 문제 파라미터 추출)에서 정확성과 일관성을 높이기 위해 추가적인 학습 단계가 필요하다. 이때 적용되는 것이 바로 Fine-tuning(미세조정)이다.

Fine-tuning 은 사전 학습된 LLM 에 대해 특정 목적의 데이터셋을 추가 학습시켜, 모델의 파라미터를 미세하게 조정하는 과정이다. 이 과정을 통해 모델은 특정 도메인, 특정 데이터 형식, 특정 문제, 특정 문제 유형 등 특정한 작업에 더욱 최적화된 성능을 발휘할 수 있게 된다. Fine-tuning 은 전통적인 학습 방식과 달리 기존 사전 학습된 모델의 학습 파라미터를 그대로 유지한 채, 새로운 데이터셋을 기반으로 추가 학습을 수행하여 모델의 응답 스타일과 정보 처리 방식을 조정한다.

본 연구에서는 GPT-4o 기반으로 Fine-tuning 을 적용한 버전과, 적용하지 않은 버전을 비교하였다. Fine-tuning 을 적용한 버전은 2 가지로 데이터 셋을 각각 10 개 학습한 모델과 20 개 학습한 모델이다. 학습에 사용된 물리 문제는 실제 학생들을 상대로 유통되는 문제집의 문제 중 선형 운동에 관련된 문제를 선택하여 사용했으며[14], 해당 물리 문제에서 파라미터를 추출할 때, 특정 문제와 그에 맞는 반환해야 할 Function 의 파라미터를 학습 데이터로 제공하여 GPT 의 문제 분석과 파라미터 추출 성능을 최적화하였다. 이를 통해 파라미터 추출의 일관성과 정답률을 높이고자 하였다.

## 5. Fine-tuning 결과 비교

Fine-tuning 의 성능을 비교하기 위해 세 가지 모델을 대상으로 실험을 수행하였다.

명칭	설명
GTP-4o	Fine-tuning 모델의 기본 모델
FT10	GTP-4o 기반으로 데이터 셋 10 개를 학습시킨 Fine-tuning 모델
FT20	GTP-4o 기반으로 데이터 셋 20 개를 학습시킨 Fine-tuning 모델

Table 2. Models used in test

### 5.1 실험 조건

실험에 사용된 문제의 조건은 다음과 같다. “자동차 A 와 B 는 동일한 위치에서 출발하여 등가속도 운동을 하며, 자동차 A 의 초기 속도는  $2m/s$  이다. 운동을 시작한 후  $t$  초가 되는 시점에 두 차량은 각각  $30m, 18m$  의 위치에 도달한다. 두 자동차의 가속도는 크기와 방향이 동일하다.”[15]

이 문제는 시간( $t$ )과 가속도( $a$ )가 주어지지 않은 상태에서, 주어진 조건만으로 이를 정확히 추론해야 하는 것이 핵심이다. 조건에 따르면 이 문제의 가속도는  $1m/s^2$  총 시뮬레이션 시간은 6 초이다. 실험 방법은 해당 조건이 포함된 문제 이미지를 입력으로 제공하고, 모델이 생성한 자동차의 파라미터(초기 속도, 가속도 등)를 기반으로 시뮬레이션을 수행한다. 이후, 시뮬레이션 결과에서 6 초 후 자동차 A 와 B 가 각각 30m, 18m 위치에 도달하는지를 기준으로 정답 여부를 판별한다.

## 5.2 정답률 비교

각 모델은 하나의 Thread 내에서 동일한 조건으로 10 번의 응답을 생성하였다. Table 3 은 Thread 별 정확도이다.

Trial Number	GPT-4o Model	FT Model 10 sets	FT Model 20 sets
1	70%	70%	100%
2	10%	0%	70%
3	60%	0%	100%

Table 3. Accuracy of each model (10 trials per Thread)

FT10 모델은 기본 모델보다 정확도가 오히려 낮게 나타나는 경우가 있었으나, 응답의 오답 유형이 일관되고 제한적이라는 특성이 있었다. 반면, GPT-4o 기본 모델은 상대적으로 높은 정확도를 보이는 경우도 있었으나, 오답의 편차가 크고 다양한 유형의 오류를 포함하고 있어 응답의 일관성이 낮았다.

FT20 모델의 경우, 정확도에서 두 모델 모두를 상회하였고, 대부분의 응답에서 정답 또는 출발 위치 설정 오류에 따른 경미한 오차만 나타났다. 특히 가속도 및 초기 속도는 정확히 분석한 결과였다. 이로써 Fine-tuning 성능은 학습 데이터의 양에 따라 유의미하게 향상됨이 검증되었다.

## 5.3 각 모델의 세부 응답 비교

Table 4, Table 5, Table 6 는 각 모델이 생성한 시뮬레이션 결과를 세부적으로 비교한 것이다. A 가 30, B 가 18 에 도달한 경우를 정답으로 판단하였다. 응답이 짙은 색으로 강조된 경우는 정답이다.

GPT-4o Model			
Trial Number	Thread A	Thread B	Thread C
1	A:48 B:36	A:39 B:27	A:30 B:18
2	A:48 B:36	A:48 B:36	A:30 B:18
3	A:48 B:36	A:48 B:36	A:26.4 B:14.4
4	A:30 B:18	A:26.4 B:14.4	A:30 B:18
5	A:30 B:18	A:40.8 B:28.8	A:26.4 B:14.4
6	A:30 B:18	A:48 B:36	A:30 B:18

7	A:30 B:18	A:48 B:36	A:32 B:20
8	A:30 B:18	A:30 B:18	A:30 B:18
9	A:30 B:18	A:40.8 B:28.8	A:30 B:18
10	A:30 B:18	A:40.8 B:28.8	A:26.4 B:14.4
Accuracy	70%	10%	60%

Table 4. Test Results of GPT-4o model

Fine-tuning Model 10 sets			
Trail Number	Thread D	Thread E	Thread F
1	A:48 B:36	A:48 B:36	A:0 B:18
2	A:48 B:36	A:48 B:36	A:0 B:18
3	A:48 B:36	A:48 B:36	A:0 B:18
4	A:30 B:18	A:48 B:36	A:0 B:18
5	A:30 B:18	A:48 B:36	A:48 B:36
6	A:30 B:18	A:48 B:36	A:48 B:36
7	A:30 B:18	A:48 B:36	A:48 B:36
8	A:30 B:18	A:48 B:36	A:48 B:36
9	A:30 B:18	A:48 B:36	A:48 B:36
10	A:30 B:18	A:48 B:36	A:48 B:36
Accuracy	70%	0%	0%

Table 5. Test results of Fine-tuning model 10 sets

Fine-tuning Model 20 sets			
Trail Number	Thread G	Thread H	Thread I
1	A:30 B:18	A:30 B:268	A:30 B:18
2	A:30 B:18	A:30 B:268	A:30 B:18
3	A:30 B:18	A:30 B:268	A:30 B:18
4	A:30 B:18	A:30 B:18	A:30 B:18
5	A:30 B:18	A:30 B:18	A:30 B:18
6	A:30 B:18	A:30 B:18	A:30 B:18
7	A:30 B:18	A:30 B:18	A:30 B:18
8	A:30 B:18	A:30 B:18	A:30 B:18
9	A:30 B:18	A:30 B:18	A:30 B:18
10	A:30 B:18	A:30 B:18	A:30 B:18
Accuracy	100%	70%	100%

Table 6. Test results of Fine-tuning model 20 sets

FT10 모델은 전체적으로 정답률이 낮았지만, 모든 Thread 에서 유사한 오답 패턴을 반복했다는 점에서 응답의 예측 가능성 및 안정성은 향상된 것으로 볼 수 있다. 반면, GPT-

40는 Thread 마다 응답 편차가 심했으며, 동일한 조건에서도 일관되지 않은 결과를 도출하였다.

FT20 모델은 대부분 정확한 값을 반환하였으며, 일부 Thread 에서 B의 출발 위치가 잘못 지정된 경우(예: 초기 위치 250m + 시뮬레이션 거리 18m)를 제외하면, 정답률 및 수치 정확도 모두 우수한 성능을 보였다.

### 5.4 단일 시행 테스트 결과

GPT 모델은 Thread 내의 여러 메시지를 참고하여 응답을 생성하는 특성이 있어, 이전 응답의 영향을 받을 가능성이 있다. 이를 배제하기 위해 각 모델에 대해 Thread 당 단 1 회만 응답하도록 설정한 단일 시행 실험을 수행하였다. 결과는 Table 7에 요약되어 있다.

GPT-4o		FT Model 10 Sets		FT Model 20 Sets	
Thread	Result	Thread	Result	Thread	Result
Thread A-1	A: 66 B: 54	Thread B-1	A: 48 B: 36	Thread C-1	A: 30 B: 18
Thread A-2	A: 66 B: 54	Thread B-2	A: 0 B: 18	Thread C-2	A: 30 B: 18
Thread A-3	A: 66 B: 54	Thread B-3	A: 0 B: 18	Thread C-3	A: 30 B: 18
Thread A-4	A: 84 B: 72	Thread B-4	A: 48 B: 36	Thread C-4	A: 30 B: 18
Thread A-5	A: 12 B: 0	Thread B-5	A: 30 B: 18	Thread C-5	A: 30 B: 18
Thread A-6	A: 48 B: 36	Thread B-6	A: 48 B: 36	Thread C-6	A: 30 B: 18
Thread A-7	A: -6 B: -18	Thread B-7	A: 48 B: 36	Thread C-7	A: 30 B: 18
Thread A-8	A: 48 B: 21.6	Thread B-8	A: 48 B: 36	Thread C-8	A: 30 B: 18
Thread A-9	A: 76.8 B: 43.2	Thread B-9	A: 48 B: 36	Thread C-9	A: 48 B: 36
Thread A-10	A: 66 B: 54	Thread B-10	A: 48 B: 36	Thread C-10	A: 30 B: 18
Accuracy	0%	Accuracy	10%	Accuracy	90%

Table 7. Test result of single trial of each model

GPT-4o 모델은 단일 시행에서 정확도가 현저히 낮아졌고 오답 편차가 컸으며, 일관성도 확인되지 않았다. 반면 FT10 모델은 응답이 대부분 오답이었음에도 동일한 형태의 수치를 반복적으로 반환하여 앞선 실험과 동일한 일관성 있는 패턴을 보였다.

FT20 모델은 대부분의 경우에서 정답에 근접하거나 정확한 응답을 도출하여, 데이터셋의 규모가 Fine-tuning 성능에 결정적인 영향을 미친다는 점을 실험적으로 입증하였다.

## 6. 결과 및 향후 연구

본 연구에서는 GPT API의 다양한 기능을 활용하여, 물리 문제 이미지로부터 시뮬레이션 파라미터를 자동 추출하고,

언리얼 엔진 기반 시뮬레이션에 적용하여 자동 생성하는 시스템을 제안하였다. 기존 방식처럼 사용자가 문제 조건을 수동으로 입력하지 않아도, GPT의 자연어 이해 및 Function 기능을 통해 구조화된 데이터를 얻을 수 있도록 하였다.

특히, Chat Completion과 Assistant를 병행하여 문제 분석과 파라미터 생성의 정확성을 확보하였으며, Function 기능을 통해 시뮬레이터 구성에 필요한 데이터를 명확하고 일관되게 반환하도록 설계하였다. 또한, Fine-tuning을 통해 문제 유형에 특화된 학습을 수행하였으며, 데이터셋의 양이 성능에 미치는 영향을 실험을 통해 검증하였다. 20개 데이터셋으로 Fine-tuning한 모델은 정답률과 응답의 정밀도 면에서 가장 우수한 성능을 보여 그 효과를 실증하였다.

**07** 그림과 같이 등가속도 운동을 하는 자동차 A가 시간  $t=0$ 일 때 기준선 P를 속력  $2\text{ m/s}$ 로 지나는 순간 P에 정지해 있던 자동차 B가 등가속도 운동을 시작하여  $t=t_0$ 일 때 A, B는 P로부터 각각  $30\text{ m}$ ,  $18\text{ m}$  떨어진 기준선 R, Q를 지난다. A, B의 가속도의 방향은 같고, 가속도의 크기는  $a_0$ 이다.

이에 대한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은? (단, 자동차의 크기는 무시한다.)

• 보기 •

- ㄱ.  $t_0=6$ 초이다.
- ㄴ.  $a_0=2\text{ m/s}^2$ 이다.
- ㄷ. 3초일 때, P와 A 사이의 거리는 P와 B 사이의 거리보다 6m만큼 크다.

① ㄱ    ② ㄴ    ③ ㄱ, ㄷ    ④ ㄴ, ㄷ    ⑤ ㄱ, ㄴ, ㄷ

Figure 5. Physics question example

Figure 5는 2025 학년도 수능특강 과학탐구영역 물리학 1 문제집 25p 7번 문제로 실제 연구과정에서 사용된 문제이다[15].

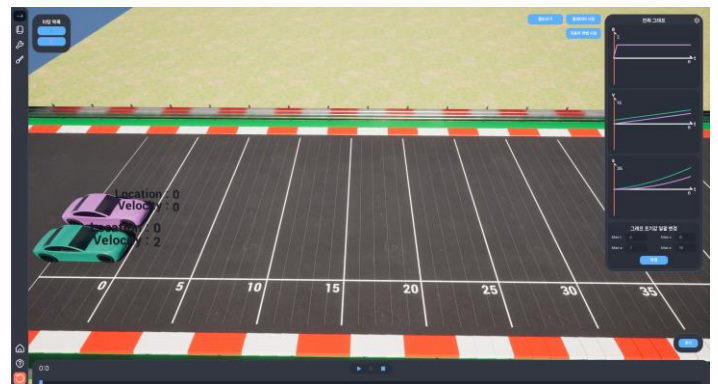


Figure 6. After the creation of the simulator with AI

Figure 5의 문제 이미지를 그대로 입력하여 생성을 시도했고 Figure 6와 같이 자동차가 생성되었다.



Figure 7. Simulation result

시뮬레이션을 실행하여 얻어진 결과가 Figure 7 로 총 6초간 주행하여 A, B 각각 30m, 18m 에 위치했다. 이 결과는 문제의 조건에 정확히 맞아 떨어지며 해당 결과를 통해 Figure 5 의 (7)이 참임을 알 수 있다.



Figure 8. Simulated for 3 seconds

실제로 시간을 3초로 설정하고 시뮬레이션 한 결과 각각 10.5m, 4.5m 에 위치했고, 이는 Figure 5 의 (7)의 조건이 참임을 알 수 있으며 그래프에서 가속도는  $2m/s^2$ 에 도달하지 못한 것을 확인할 수 있다. 따라서 (7)은 거짓이므로 3 번 (7),(7)이 정답임을 시뮬레이션을 통해 유도할 수 있다. 이 결과를 통해 AI 가 생성한 시뮬레이터 파라미터를 통해 문제를 풀 수 있음을 확인했다. 제시된 내용은 대표적인 케이스이며 해당 문제와 비슷한 유형의 다른 문제의 경우 <https://whaletek.net> 을 통해 결과 확인이 가능하다.

이러한 결과는 ChatGPT API 를 활용한 자동 시뮬레이션 생성 시스템이 학습용 도구로서 충분한 실용성을 가질 수 있음을 보여준다. 다만 연구된 물리 문제는 직선운동에 국한된 콘텐츠 생성에 적용되었다는 한계가 있다. 향후 연구에서는 다양한 문제 유형과 물리 영역(예: 역학, 전자기, 파동, 전자회로 등)의 파라미터를 생성하는 Fine-tuning 데이터셋을 구성하고, 각 문제의 유형을 판단하여 해당 유형을 담당하는 AI 로 분석을 넘기는 AI 끼리의 상호작용 시스템으로 발전시킬 예정이다.

다만, Fine-tuning 에 사용될 수 있는 문제의 수가 다른 분야에 비해 상대적으로 작아 정답률을 높이는데 한계가 있다. 이는 성공적으로 생성된 문제의 파라미터 수치를 다변화하여 다시 학습시키는 형태 등의 개선방향이 필요할 것으로 보인다. 또한 정답률을 분석하기 위한 수치를 표 형태로 구성하였는데, 이는 직관적인 판단을 어렵게 하는 단점이 있다. 이는 그래프 등을 활용하여 추세를 직관적으로 표현, fine-tuning 결과를 쉽게 분석할 수 있게 하는 등의 개선이 필요하다.

## References

- [1] 김태원 ChatGPT는 혁신의 도구가 될 수 있을까? : ChatGPT 활용 사례 및 전망 한국지능정보사회지능 원 The AI Report, 2023.
- [2] 목경열. 한의사의 진료분야와 의료 적용분야의 AI 도입과 유용도에 대한 인식조사 연구. 보건의료생명과학논문지, 10(2), 413-421. 2022.
- [3] 윤제정. AI를 활용한 바이오산업 분야 미래와 전망. 전자공학회지, 26-32. (2024).
- [4] 김정아, 강두식, 고용철. 생성형 AI의 교육적 활용 방안 연구 - ChatGPT 활용을 중심으로. 정보교육학회논문지, 27(6), 691-704, 2023.
- [5] 신동광. AI 도구를 활용한 중등교사 영어 출제 연수 사례 연구: ChatGPT를 중심으로. Language Research, 59(1), 21-42. 2023.
- [6] 박희경, 박정우. AI와의 협업을 통한 초등예비교사의 물리 수업지도안 작성: ChatGPT를 중심으로. New Physics: Sae Mulli, 74(2), 215-226. 2024
- [7] 이재화, 정우성, 이은주. ChatGPT 기반 초등학생 대상 영어 교육 AI 플랫폼 개발 및 적용. 컴퓨터교육학회논문지, 28(4), 75-86. 2025.
- [8] 강동열. 거대언어모델 기반 인공지능의 일반물리학 학습 도구로서의 가능성 탐색. 컴퓨터교육학회논문지, 27(4), 147-160. 2024
- [9] 김주인, 유훈. 물리 교육을 위한 ChatGPT 활용 방안 탐색: 고등학교와 일반물리 수업 중심으로. 현장과학교육, 17(3), 216-239. 2023
- [10] 이동원, 심현표, 백종호. 과학 교과 학생 평가에서 ChatGPT의 활용 가능성 및 교사 인식 탐색. 한국과학교육학회지, 44(1), 119-130. 2024
- [11] 김지은, 김미량. 기술수용모델을 적용한 학습 도구로서의 ChatGPT 이용 의도에 관한 탐구. 컴퓨터교육학회논문지, 27(3), 75-86. 2024
- [12] 이주아, 유선아. 생성형 AI를 활용한 과학교육 분야의 국내외 연구 동향: 수업 설계와 학습 도구로서의 활용 분석.

한국과학교육학회지, 45(1), 19-31. 2025

[13] 김주은. 인공지능이 인간사회에 미치는 영향에 대한 연구.  
The Journal of the Convergence on Culture Technology, 5(2),  
177-182. 2019

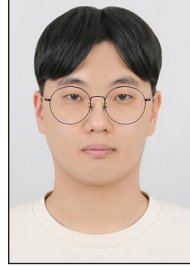
[14] 마더텅 편집부. 2026 마더텅 수능기출문제집 물리학 1

[15] EBS. 2025학년도 수능특강 과학탐구영역 물리학 1

## 〈 저자 소개 〉

### 박재우

- 2018~2024 한국공학대학교 게임공학부 학사
- 2024~현재 (주)웨이텍 연구원
- 관심분야 : 게임개발, 게임엔진, VR
- <https://orcid.org/0009-0000-2730-4563>



### 이용선

- 2013~2019 한국공학대학교 게임공학부 학사
- 2019~2021 한국공학대학교  
미디어융합디자인공학 석사
- 2021~현재 (주)웨이텍 선임연구원
- 관심분야 : 컴퓨터 그래픽스, 게임개발,  
게임엔진, VR, XR
- <https://orcid.org/0000-0003-3412-5058>



### 이택희

- 1997~2001 서울대학교 전기컴퓨터공학부  
학사
- 2001~2009 서울대학교 컴퓨터공학부 박사
- 2009~2012 삼성전자 무선사업부
- 2017~현재 한국공학대학교 게임공학과 부교수
- 관심분야 : VR, AR, MR, XR, 체감형 콘텐츠
- <https://orcid.org/0000-0003-2985-3229>

