

## A Comparative Study on 3D Model Generation Performance Using Generative AI

Byong-Kwon Lee\*

\*Professor, School of media contents, Seowon University, Chungbuk, Korea

### [Abstract]

The rapid advancement of generative AI technology has led to its increasing reliance across various industries and everyday applications. However, most generative AI solutions have focused primarily on generating text or 2D images, with relatively little attention paid to 3D model generation. This study conducts a comparative analysis of the performance of three generative AI models ChatGPT, Copilot, and Gemini in generating 3D models. The methodology involved generating scripts for use in the open-source Blender graphics tool, which operates under a GPL license, using each of the three AI models. The performance of these models was then evaluated using metrics such as Accuracy, Recall, Precision, and F1 Score. The results showed that ChatGPT and Copilot outperformed Gemini, which exhibited lower performance. This discrepancy in performance is likely due to differences in the training data related to 3D model generation. This study demonstrates that the performance of generative AI models varies significantly depending on the domain of application, highlighting the need for further development and upgrades to improve their effectiveness in specific areas like 3D modeling.

▶ **Key words:** Generative AI, 3D Model Generation, Performance Comparison, F1 Score, Training Data

### [요 약]

생성형 인공지능 기술의 급속한 발전은 산업 및 일상생활 전반에 걸쳐 그 의존도를 높이고 있다. 그러나 대부분의 생성형 인공지능 솔루션은 주로 텍스트 또는 2D 이미지 생성에 중점을 두고 있으며, 3D 모델 생성 분야에서는 상대적으로 연구가 부족한 상황이다. 본 연구에서는 ChatGPT, Copilot, Gemini 세 가지 생성형 AI 모델을 대상으로 3D 모델 생성 성능을 비교 분석하였다. 연구 방법으로는 오픈소스 GPL 라이선스를 따르는 Blender 그래픽 제작 도구에서 사용하는 스크립트를 ChatGPT, Copilot, Gemini 모델을 통해 자동 생성한 후, 각각의 성능을 정확도(Accuracy), 재현율(Recall), 정밀도(Precision), F1 점수(F1 Score) 지표로 평가하였다. 평가 결과, ChatGPT와 Copilot은 우수한 성능을 보인 반면, Gemini는 상대적으로 낮은 성능을 나타냈다. 이러한 성능 차이는 3D 모델 생성과 관련한 학습 데이터의 차이에서 기인한 것으로 추정된다. 본 연구는 생성형 인공지능 모델이 특정 분야에 따라 성능이 상이함을 확인하였으며, 향후 성능 향상을 위해 각 모델에 대한 추가적인 학습과 업그레이드가 필요할 것으로 판단된다.

▶ **주제어:** 생성형 인공지능, 3D 모델 생성, 성능 비교, F1 점수, 학습 데이터

- First Author: Byong-Kwon Lee, Corresponding Author: Byong-Kwon Lee
- Byong-Kwon Lee (sonic747@daum.net), School of media contents, Seowon University
- Received: 2024. 10. 24, Revised: 2024. 11. 20, Accepted: 2024. 11. 27.

### I. Introduction

생성형 인공지능은 프롬프트에 반응하여 텍스트, 이미지, 미디어를 생성할 수 있는 인공지능 기술로, 기존 데이터를 단순히 분석하는 데 그치지 않고 새로운 콘텐츠를 창작하는 데 중점을 두고 있다[1]. 이 기술은 2022년에 ChatGPT의 출시를 시작으로, 이후 미드저니와 같은 여러 모델들이 연이어 공개되었다. 인공지능 기술 분야는 일반적으로 산업용 AI와 프론트 AI로 구분될 수 있으며, 산업용 AI는 특정 산업의 필요에 맞추어 개발된 응용 프로그램으로 잘 알려진 AlphaGo가 그 대표적인 사례이다. 반면 프론트 AI는 현재 유행 중인 GPT 기반의 인공지능으로, 일반 사용자들이 쉽게 접근할 수 있도록 설계되어 있으며, 그 대표적인 예로 ChatGPT가 있다[2][3]

Table 1. Two Types of AI Utilization

Industrial AI	Front AI
AI used for specific purposes (in companies)	AI services for general users
Factory assembly AI, packing AI, inspection AI, AlphaGo	AI Assistant, chatGPT, Gemini, Alexa, Copilot, Siri, Hello Google

과거에 주로 사용한 인공지능과 현재 유행인 생성형 인공지능은 다르다. Fig. 1은 분류 및 추론 기반의 전통적인 인공지능 방식인 변별(Discriminative Model)과 생성형 모델(Generative Model)이다. 생성형 모델은 현재 생성형 인공지능으로 단어, 문장 및 이미지 등의 입력으로 미리 학습된 데이터 셋(Data Set)을 통해 원하는 결과를 얻는 방식이다[4][5].

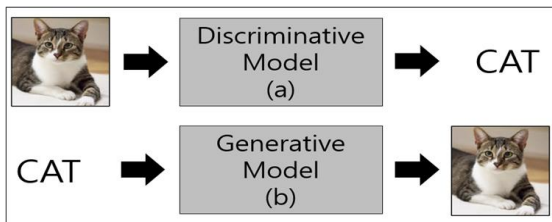


Fig. 1. Discriminative(a) Vs Generative(b) Model

본 연구에서는 생성형 인공지능 기술을 적용한 서비스인 ChatGPT, Copilot, Gemini를 사용해 3D 콘텐츠를 제작하고 정확도(Accuracy), 재현율(Recall), 정밀도(Precision), F1 점수(F1 Score)를 분석해 성능을 측정한다. 본 논문의 구성은 개요, 관련연구, 제안내용, 실험 및 결론으로 구성된다.

### II. Preliminaries

#### 1.1 Discriminative and Generative Model

전통적인 인공지능 모델인 Discriminative Model은 추론 및 판별을 목적으로 하는 머신러닝 분류 알고리즘이다. 머신러닝 분류 알고리즘은 어떤 입력값(input)  $x$ 가 주어졌을 때 결과값(label)이  $y$ 일 확률을 알아내는 것  $p(y | x) \Rightarrow x$ 라는 값이 의미 있는 데이터일 때 데이터를 기반으로  $y$ 가 0일지 1일지를 확률적으로 예측해 결과 알아내는 방식이다. Discriminative 머신러닝은 (1) 입력 값  $x$ 에 대한 조건부 확률 분포(Conditional Probability Distribution) 생성하고, (2) 조건부 확률 분포에 근거해서  $x$  input과  $y$  label(output)로 벡터를 생성한다. (3)  $x$ 값 성질(결과 값, 즉  $y$ 값)이 0인 것과 1인 것을 구분하고 선(직선 혹은 곡선으로 decision boundary)을 생성한다. (4) 선을 기반으로 새로운 데이터( $x'$ )가 입력되었을 때 선으로부터 음 혹은 양의 방향으로의 거리를 측정해 확률을 구하는 방식이다[6].

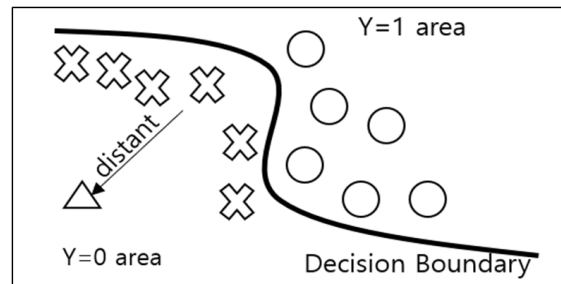


Fig. 2. Discriminative Model Algorithm

Fig. 2에서 X, O는 입력  $x$ 를 나타내며, 각각  $y=0$ 과  $y=1$ 을 의미한다. 곡선은 결정 경계를 나타내고, 세모는 새로운 입력을 나타낸다. 이때 “distant”는 결정 경계로부터의 거리를 의미하며, 이 거리가 음의 방향인지 양의 방향인지를 구분할 수 있다. 이러한 판별 모델의 대표적인 알고리즘은 로지스틱 회귀(Logistic Regression)가 있다. 이 외에도 조건부 확률 장(field, Conditional Random Field), 서포트 벡터 머신(Support Vector Machine), 선형 회귀(Linear Regression), 신경망(Neural Network) 등 다양한 알고리즘이 존재한다. 이러한 알고리즘들은 데이터가 충분할 경우 우수한 성능을 발휘하는 장점이 있다. 그러나 이들 알고리즘은 데이터를 구분하는 데 주목적을 두고 있기 때문에, 데이터의 본질적 형태를 이해하는 데 한계가 있다. 이러한 방식으로 입력값과 출력값을 직접적으로 연결하는 접근법을 변별 알고리즘(Discriminative Algorithm)이라고 한다. 생성형 인공지능에 사용되는 생

성 모델(Generative Model) 알고리즘은 판별 모델(Discriminative Model)의 구분 기능을 확장하여[7], 각 분류 카테고리에 맞는 데이터를 생성하는 방식이라고 할 수 있다. 즉, 생성되는 입력과 결과 데이터는 각 분류 클래스별로 특정한 통계적 분포를 따른다고 가정한다. 주어진 입력값  $x$ 와 결과값(라벨)  $y$ 에 대해, 이들은 일정한 분포 규칙 내에서 존재하게 된다. 이때 분포 규칙은 각 분류 클래스마다 상이하게 설정되며, 이러한 특성을 기반으로 새로운 데이터를 생성하는 것이 생성 모델의 핵심이다. 이러한 분포 규칙은 Normal Distribution 통계적인 방법론을 따른다. 분포 규칙은 수식적으로는 결합확률분포- $p(x, y)$ 로 표현한다. (결합확률분포-joint probability distribution) 실제로 분류에 필요한 확률  $p(y | x)$ 은  $p(x, y)$ 로 부터 유도할 수 있다. 핵심은 카테고리별로 분포 규칙을 표현하는 확률 분포 모델의 파라미터들을 찾아내는 것이다. 이렇게 입력값과 출력값 사이에 분포 규칙의 개념을 적용한 방식을 Generative(생성) Algorithm이라고 한다. Generative 방식의 머신러닝은 다음의 과정으로 이루어진다. (1) 주어진 데이터와 결과 값을 이용해 모든  $x$ 와 모든  $y$ 에 대해 결합 확률 분포 (Joint Probability Distribution)를 생성. (2) 결합확률 분포로부터 어떤 확률 분포모델이 데이터 분포모델에 적합한지 측정 (estimation). (3) 측정해낸 확률 분포모델을 기반으로 확률을 알아낼 수 있다.  $p(y|x) = p(x|y) * p(y) / p(x)$ , 확률 = 결합확률( $p(x, y)$ ) /  $p(x)$ . (4)  $x$ 가 주어질 경우,  $y$ 의 확률을 예측하려면,  $x, y$ 에 대한 결합확률분포공식(모델)과  $x$ 의 확률(경계확률)을 구할 수 있다. Fig. 3은 Generative Model 알고리즘으로  $X, O$ 는  $x$ 를 의미,  $X/O$ 은  $y=0/y=1$ 을 의미, 세모는 새로운 입력을, 큰 타원은 확률 분포를 의미한다[8].

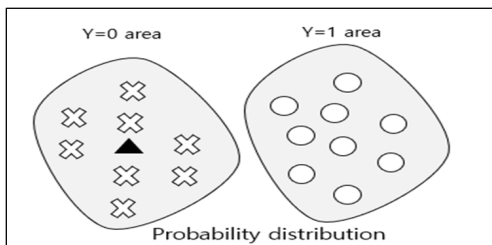


Fig. 3. Generative Model Algorithm

대표적인 방식은 Naive Bayes가 있으며 이외에도 Gaussian discriminant Analysis(GDA), Gaussian Mixture Model(GMM), Hidden Markov Model(HMM), Latent Dirichlet Allocation(LDA), Restricted

Boltzmann machine(RBM), Generative Adversarial Network(GAN)등이 있다. 이러한 알고리즘의 장점은 데이터 셋이 적어도 가능하며, 데이터 자체의 특성을 파악하기에 좋고, 데이터를 생성해 내어 새로운 결과물을 얻어낼 수 있다. 하지만, 단점은 데이터가 많은 경우, Discriminative에 비해 성능이 떨어질 수 있다.

### 1.2 GPT(Generative Pre-trained Transformer)

GPT(Generative Pre-trained Transformer)는 미국의 인공지능 연구 기관인 오픈AI(OpenAI)가 2018년에 발표한 대형 언어 모델(LLM) 계열에 속한다. GPT 모델은 레이블링되지 않은 대량의 텍스트 데이터셋을 기반으로 미리 훈련되며, 인간과 유사한 텍스트를 생성할 수 있도록 설계된 변환기 아키텍처에 기반한 인공 신경망이다. 2023년 기준으로, 대부분의 LLM은 이러한 특성을 가지고 있으며, 일반적으로 GPT로 통칭된다. Table 2는 버전별 GPT의 특징을 나타내며, GPT-4의 경우 일부 정보가 공개되지 않은 상태이다. 현재는 GPT-5의 출시가 예정되어 있다 [9][10][11].

Table 2. GPT Features by Version

Model	Parameter	Arch	Data(Gb)
GPT.1	117m	softmax	4.5
GPT.2	1,500m	GPT-1	40
GPT.3	175,000m	GPT-2	570
GPT.4	Private	RLHF	Private

GPT-1

2018

GPT-2

2019

GPT-3

2020

GPT-3.5

2022

GPT-4.0

2023

특히 GPT-4는 기존의 텍스트 입력 방식에서 벗어나 이미지 입력 및 음성 입력을 동시에 지원하는 특징을 가지고 있다. 또한, 강화 학습 기반 인간 피드백(RLHF, Reinforcement Learning from Human Feedback)을 적용하여 성능을 향상시키고 있다. RLHF는 두 단계로 구성되며, 첫 번째 단계에서는 지도 학습을 통해 모델이 초기 학습을 수행하고, 두 번째 단계에서는 우선순위 강화 학습을 통해 보다 우수한 답변을 도출하도록 한다. 더불어, GPT-4는 기존에 문제로 제기되었던 안전성과 신뢰성 문제를 강화하여, 도덕적으로 부적절한 질문에 대한 답변을 회피하는 기능을 개선하였다. 이러한 변화는 모델의 윤리적 사용을 촉진한다. Fig. 4는 휴먼 피드백 강화 학습(RLHF)의 동작 방식을 보여주며, 두 단계의 학습 과정을 진행한다. 예를 들어, "초등학교 학생에게 중력을 설명해 줘?"라는 질문에 대한 답변에서는 두 번째 단계에서 여러

개의 답변 중 초등학생이 이해하기 쉬운 우선순위가 높은 답변을 선정해 제공하는 방식이다. 이러한 접근을 통해, 기존의 방식보다 진보된 형태로 생성형 인공지능의 성능을 고도화하고 있다[12][13].

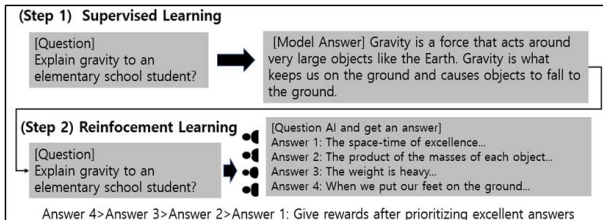


Fig. 4. RLHF Algorithm

### 1.3 Generative AI services and 3D Contents

생성형 AI는 텍스트, 이미지, 비디오 등 다양한 형태의 새로운 콘텐츠를 생성하는 인공지능 모델이다. 이러한 모델들은 방대한 양의 데이터를 학습하여, 사용자의 요구에 맞는 결과물을 만든다. Table 3에는 대표적인 생성형 AI 모델로 DALL-E 2, Midjourney[14], Stable Diffusion[15] 과 같은 이미지 생성 모델과 ChatGPT, GPT-3, Gemini와 같은 텍스트 생성 모델, 그리고 RunwayML[16], Synthesia와 같은 비디오 생성 모델이 포함되어 있다. 이 외에도 개발자를 위한 AI인 Copilot과 같이 특정 분야에 특화된 모델도 존재한다. 이러한 생성형 AI 모델들은 예술, 디자인, 콘텐츠 제작, 소프트웨어 개발 등 다양한 분야에서 활용될 가능성이 크며, 인간의 창의성을 증폭시키고 새로운 가치를 창출하는데 이바지할 것으로 기대된다. 그러나 저작권 문제와 윤리적 고려사항 등 해결해야 할 과제가 있다.

Table 3. Generative AI services

Category	Product Name	Key Features
Image	DALL-E 2	Generates realistic images from textual descriptions
	Midjourney	Generates high-quality artistic images, operated on Discord
	Stable Diffusion	Open-source model, capable of generating images in various styles
	RunwayML	Offers various features including image editing, generation, and animation
Text	ChatGPT	Provides various functions such as natural conversation, text generation, and translation
	GPT-3	Capable of performing various text generation tasks, excellent language understanding ability
	Jasper.ai	Specializes in generating various text content such as marketing content and blog posts
	Gemini	Offers various language models, text

		generation, translation, code generation, and more
	Copilot	AI pair programmer for developers, offering code autocompletion, code generation, and code explanation
Video	RunwayML	Generates videos based on images, style transfer, etc.
	Synthesia	Generates realistic videos based on text
	DeepMind	Develops AI-based video editing technology

Table 4는 Blender와 Unity3D와 같은 3D 제작 도구에서 생성형 AI를 적용할 수 있는 목록이다. 현재는 ChatGPT, Gemini, Copilot과 같은 언어모델을 활용하여 3D 모델링 과정을 간접적으로 지원하는 방법이 주로 사용되고 있다. 이러한 모델들은 3D 모델에 대한 설명을 제공하거나, 스크립트를 작성하는 등 다양한 방식으로 3D 모델 제작을 한다. 하지만 아직까지는 Blender나 Unity3D와 직접 연동되어 3D 모델을 생성하는 상용 도구는 적다. 대부분의 생성형 AI 모델은 이미지나 텍스트 생성에 특화되어 있으며, 3D 모델 생성은 아직 연구 단계에 있는 경우가 많다. 따라서, 현재는 생성된 이미지나 텍스트를 바탕으로 3D 모델을 제작하는 방식이 주로 사용되고 있다. 앞으로는 3D 모델 생성에 특화된 생성형 AI 도구들이 더욱 많이 등장할 것으로 예상된다. 이러한 도구들은 Blender나 Unity 3D와의 직접적인 연동을 통해 더욱 편리하고 효율적인 3D 모델 제작이 가능하다[17][18].

Table 4. Generative AI for 3D Contents Modeling

Company	Product Name	Key Features	Potential Integration with Blender/Unity
OpenAI	ChatGPT	Text generation, information retrieval	Indirect integration possible (script writing)
Google	Gemini	Text generation, translation, code generation	Indirect integration possible (script writing, Q&A)
Microsoft	Copilot	Code autocompletion, code generation,	Indirect integration possible (C# script support)
RunwayML	RunwayML	Image generation, video editing, style transfer	Indirect integration possible (import generated images into Blender/Unity)
Midjourney	Midjourney	Generates high-quality artistic images, Discord-based	Indirect integration possible (import generated images into Blender/Unity)
Stable Diffusion	Stable Diffusion	Open-source model, capable of generating images in various styles	Indirect integration possible (import generated images into Blender/Unity)

### III. The Proposed Scheme

이 절에서는 Table 5에서 ChatGPT, Copilot, Gemini의 특징, 장점, 단점을 비교 분석하였다. ChatGPT는 주로 자연어 처리와 코드 작성에 최적화되어 있으며, 폭넓은 지식 기반과 뛰어난 자연어 처리 능력이 장점이다. 그러나 전문 지식이 부족할 수 있고, 오류가 발생할 가능성이 상대적으로 높다는 단점이 있다. Copilot은 코드 자동 완성 및 오류 수정 기능이 특징으로, 코드의 품질을 높이고 생산성을 향상시킨다. 하지만 특정 프로그래밍 언어나 프레임워크에 의존하는 경향이 있고, 새로운 코드를 생성하는 데 어려움이 있을 수 있다. Gemini는 다국어 지원과 복잡한 문제 해결에 강점이 있지만, 모델 크기가 크고 연산 비용이 높다는 단점이 있다.

Table 5. Features of ChatGPT, Copilot and Gemini

Model	Key Features	Strengths	Weaknesses
ChatGPT	Natural language processing, code generation	Broad knowledge base, strong natural language processing capabilities	Lack of specialized knowledge, potential for errors
Copilot	Code autocompletion, function recommendation, error correction	Improved code quality, increased productivity	Dependence on specific languages and frameworks, difficulty generating entirely new code
Gemini	Support for multiple languages, complex problem-solving	Ability to solve complex problems, code explanation generation	Large model size, high computational cost

또한, 연구에서는 ChatGPT, Copilot 및 Gemini에 대하여 F1 Score를 측정했다. F1 Score는 정밀도와 재현율의 조화 평균으로, 정확도와 유사하게 모델의 성능을 종합적으로 측정방식이다[19]. Accuracy는 올바르게 예측된 데이터의 수를 전체 데이터의 수로 나눈 값으로 식은 (1)과 같다.

$$Accuracy = \frac{TP}{TP + TN + FN + FP} \quad (1)$$

식 (2)는 실제로 True인 데이터를 모델이 True라고 인식한 데이터의 수 판단하는 Recall이다.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

식 (3)은 모델이 True로 예측한 데이터 중 실제로 True인 데이터의 수 Precision이다.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

- TP(True Positives) : 모델이 양성(Positive)으로 예측하고, 실제로 해당 샘플이 양성인 경우
- TN(True Negatives) : 모델이 음성(Negatives)으로 예측하고, 실제로 해당 샘플이 음성인 경우
- FP(False Positives) : 모델이 양성(Positive)으로 예측하고, 실제로 해당 샘플이 음성인 경우
- FN(False Negatives) : 모델이 음성(Negatives)으로 예측하고, 실제로 해당 샘플이 음성인 경우

### IV. Experiment and Analysis

본 연구의 실험을 Scikit-learn을 사용해 ChatGPT, Copilot, Bard가 생성한 블렌더 큐브 스크립트의 성능을 F1 Score로 평가하는 코드이다. 이 코드는 각 AI가 생성한 큐브 스크립트를 기준으로 Precision, Recall, F1 Score를 계산한다. Fig. 5는 ChatGPT, Copilot 및 Gemini에 대한 F1-Score를 측정하는 파이썬 프로그램이다. Fig. 5에서 "1"은 올바르게 생성된 큐브, "0"은 잘못 생성되었거나 누락된 큐브를 나타낸다. ChatGPT\_generated는 10개의 큐브 중 8개는 정확하고 2개는 속성 오류. copilot\_generated는 8개만 생성되고, 2개가 누락. gemini\_generated는 10개가 모두 정확하게 생성되고, True Labels는 정답 큐브는 모두 10개이며, 각 큐브가 올바르게 생성되어야 하므로 "1"로 표현된다. 본 연구에서는 생성형 AI 모델인 ChatGPT, Copilot, Gemini가 작성한 블렌더 스크립트를 평가하기 위해 Precision, Recall, Accuracy, F1 Score와 같은 성능 지표를 계산하였다. 각 모델이 제공한 스크립트를 블렌더(Blender) 소프트웨어에 입력한 후 출력된 결과를 바탕으로 평가를 진행하였다. Table 5는 각 모델이 제시한 스크립트를 블렌더에서 실행한 결과를 요약한 것이다.

```

from sklearn.metrics import precision_score, recall_score, f1_score
# Example: Each AI generated 10 cubes, and we assess how many cubes were correctly generated.
# 1 means a correctly generated cube, 0 means an incorrectly generated or missing cube.
# AI-generated cube results
# ChatGPT generates 10 cubes, but 2 cubes overlap (incorrect position or attributes)
chatgpt_generated = [1, 1, 1, 1, 1, 1, 1, 1, 0, 0]
# Copilot generates only 8 cubes, missing 2 cubes
copilot_generated = [1, 1, 1, 1, 1, 1, 1, 0, 0]
# Gemini generates exactly 10 cubes, all correctly generated
gemini_generated = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
# Ground truth: The correct number of cubes is 10, all correctly generated
true_labels = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
# Function to calculate Precision, Recall, and F1 Score
def evaluate_model(generated, true_labels):
    precision = precision_score(true_labels, generated)
    recall = recall_score(true_labels, generated)
    f1 = f1_score(true_labels, generated)
    return precision, recall, f1

# Evaluate ChatGPT's performance
chatgpt_precision, chatgpt_recall, chatgpt_f1 = evaluate_model(chatgpt_generated, true_labels)
print(f"ChatGPT - Precision: {chatgpt_precision}, Recall: {chatgpt_recall}, F1 Score: {chatgpt_f1}")

# Evaluate Copilot's performance
copilot_precision, copilot_recall, copilot_f1 = evaluate_model(copilot_generated, true_labels)
print(f"Copilot - Precision: {copilot_precision}, Recall: {copilot_recall}, F1 Score: {copilot_f1}")

# Evaluate Gemini's performance
gemini_precision, gemini_recall, gemini_f1 = evaluate_model(gemini_generated, true_labels)
print(f"Gemini - Precision: {gemini_precision}, Recall: {gemini_recall}, F1 Score: {gemini_f1}")
    
```

Fig. 5. ChatGPT, Copilot and Gemini for F1-Score

Q1: "서로 5M씩 떨어진 큐브 10개를 랜덤하게 생성하는 블렌더 스크립트 작성해줘"(Can you write a blender script that randomly generates 10 cubes 5m apart from each other?) 이 질문에 대한 각 생성형 AI의 응답은 서로 5미터씩 떨어진 큐브 10개를 무작위로 생성하는 블렌더 스크립트를 포함하고 있었다. 모델들은 큐브 배치의 무작위성과 규칙성을 반영한 다양한 방법으로 답변을 제시하였으며, 각각의 스크립트는 블렌더에서 성공적으로 실행되었다.

Q2: "등받이 1개이고 다리가 4개인 의자를 만드는 블렌더 스크립트 작성해줘"(Can you write a blender script to make a chair with one back and four legs?) 두 번째 질문에서는 각 모델이 하나의 등받이와 네 개의 다리를 갖춘 의자를 생성하는 스크립트를 작성하였다. 이 역시 각 스크립트는 블렌더에서 정상적으로 실행되었으며, 생성된 객체는 요구사항을 충족하였다. Table 6은 두 개의 질문에 대해 블렌더 스크립트를 생성하고 실행한 결과를 나타낸다. ChatGPT와 Copilot은 유사한 방식으로 스크립트를 생성하여 성공적으로 실행되었으나, Gemini의 경우 두 번째 질문에서 스크립트 생성에 문제가 발생한 것으로 확인되었다.

Table 6. Generative AI Text script Test

Text	Q1	Can you write a blender script that randomly generates 10 cubes 5m apart from each other ?	
	Q2	Can you write a blender script to make a chair with one back and four legs ?	
AI Tools	Result		
	Q1	Q2	
ChatGPT4.0 for OpenAI (Q1:21 Line) (Q2:38 Line)			
Copilot for Microsoft (Q1:17Line) (Q2:33Line)			
Gemini for google (Q1:38Line) (Q2:46Line)			

Table 7은 AI 모델(ChatGPT, Copilot, Gemini)이 생성한 결과를 실제 라벨과 비교하여 각각의 정확도(Accuracy), 재현율(Recall), 정밀도(Precision), F1 점수(F1 Score)를 계산한 결과이다. 또한, ChatGPT4.0, Copilot, Copilot 세 가지 모델의 예측값을 비교하여 평가하고, 각 모델의 성능을 시각화했다. 평가한 결과 값들은 각 모델별로 정확도, 재현율, 정밀도, F1 점수로 나누어 도표 작성했다.

Table 7. F1 Score for ChatGPT, Copilot and Gemini

Text	Q1	Can you write a blender script that randomly generates 10 cubes 5m apart from each other ?		
	Q2	Can you write a blender script to make a chair with one back and four legs ?		
Avg	Average value for two questions			
AI Tools	Accuracy	Recall	Precision	F1 Score
ChatGPT4.0 for OpenAI	0.70	0.78	0.88	0.82
	0.70	0.70	1.00	0.82
Copilot for Microsoft	0.60	0.67	0.86	0.75
	0.90	0.90	1.00	0.95
Gemini for google	0.75	0.79	0.93	0.85
	0.50	0.56	0.83	0.67
	0.20	0.20	1.00	0.33
	0.35	0.38	0.92	0.50

Fig. 6, 7은 막대그래프와 선 그래프를 활용해 정확도 (Accuracy), 재현율(Recall), 정밀도(Precision), F1 점수 (F1 Score)를 표시한 것이다. 본연구의 관련성은 큐브를 생성하는 정확도를 평가하는 요소로 사용된다. ChatGPT는 예측 성능이 비교적 좋고 대부분의 예측이 실제 값과 일치했다. Copilot은 예측된 값들이 실제 라벨과 높은 일치도를 보이고, 정확도와 재현율에서 우수했다. 마지막으로 Gemini의 경우 성능이 다소 낮고 다수의 0값을 예측했으며, 특히 재현율이나 정확도에서 낮은 점수를 보였다. 3가지 모델을 실험한 결과 항상 Gemini 모델이 안 좋다고 판단하기 어렵다. 본 실험에서는 단지 블렌더의 스크립트 생성에 대한 부분만 비교한 것으로 일반적인 다른 유형 성능에서는 다를 것으로 사료된다.

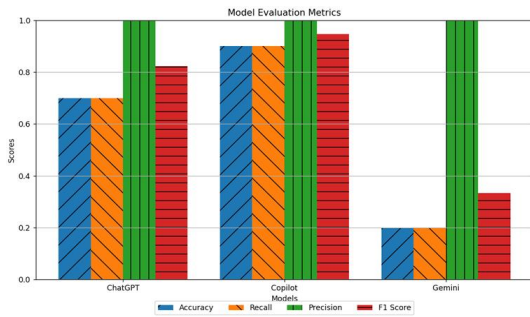


Fig. 6. Bar graph for F1-Score

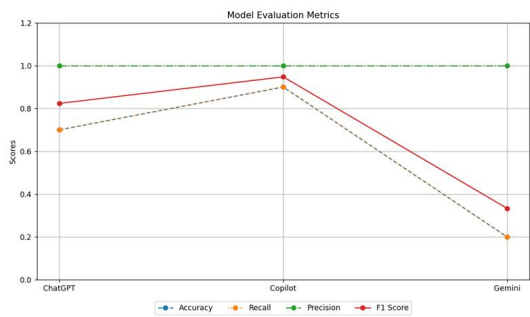


Fig. 7. Line graph F1-Score

## V. Conclusions

본 연구에서는 생성형 AI에 대표적인 서비스인 ChatGPT, Copilot, Gemini 세 가지 AI 모델을 대상으로 3D 모델 생성 성능을 비교 분석하였다. 연구 방법으로는 오픈소스 GPL 라이선스를 따르는 Blender 그래픽 제작 도구에서 사용하는 스크립트를 ChatGPT, Copilot, Gemini 모델을 통해 자동 생성한 후, 각각의 성능을 정확도(Accuracy), 재현율(Recall), 정밀도(Precision), F1 점

수(F1 Score) 지표로 평가하였다. 평가 결과, ChatGPT의 경우 두 개의 질문에 대한 평균 0.82, Copilot은 0.85, Gemini 0.50의 성능을 보였으며, 결론적으로 우수한 성능을 ChatGPT, Copilot이고, Gemini는 상대적으로 낮은 성능을 나타냈다. 이러한 성능 차이는 3D 모델 생성과 관련한 학습 데이터의 차이에서 기인한 것으로 추정된다. 하지만, 본 연구는 3D 모델을 생성하는 과정에 대한 평가로 Gemini 모델이 다른 모델보다 성능 낮음을 보여주는 것은 아니다. 스크립트 생성에서 사용하는 프롬프트의 내용에 따라 성능은 다시 달라질 수 있다. 향후 연구 방향으로서는 기본 유료화된 서비스 대상으로 평가를 진행하고, 다양한 생성형 AI에 대한 비교 분석이 필요하다. 또한, Open Source를 제공하는 Stable Diffusion과 같은 도구를 활용하여 기존 3D 애플리케이션에 플러그인 형태로 통합하는 방법 연구가 필요하다.

## REFERENCES

- [1] G. Rani, J. Singh and A. Khanna, "Comparative Analysis of Generative AI Models," 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), Faridabad, India, 2023, pp. 760-765, doi: 10.1109/ICAICCIT60255.2023.10465941.
- [2] M. Hollender, C. Xu and R. Tan, "Engineering Challenges in Industrial AI," 2024 IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI (CAIN), Lisbon, Portugal, 2024, pp. 41-42.
- [3] P. P. Bonissone and A. Liani, "Industrial AI [Industrial and Governmental Activities]," in IEEE Computational Intelligence Magazine, vol. 18, no. 1, pp. 10-12, Feb. 2023, doi: 10.1109/MCI.2022.3223460.
- [4] P. Dreu, G. Heigold and H. Ney, "Confidence-Based Discriminative Training for Model Adaptation in Offline Arabic Handwriting Recognition," 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009, pp. 596-600, doi: 10.1109/ICDAR.2009.116.
- [5] G. Rani, J. Singh and A. Khanna, "Comparative Analysis of Generative AI Models," 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), Faridabad, India, 2023, pp. 760-765, doi: 10.1109/ICAICCIT60255.2023.10465941.
- [6] P. G. Shivakumar, J. Kolehmainen, Y. Gu, A. Gandhe, A. Rastrow and I. Bulyko, "Discriminative Speech Recognition Rescoring With Pre-Trained Language Models," 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Taipei,

- Taiwan, 2023, pp. 1-7, doi: 10.1109/ASRU57964.2023.10389787.
- [7] S. Warudkar and R. Jalit, "Unlocking the Potential of Generative AI in Large Language Models," 2024 Parul International Conference on Engineering and Technology (PICET), Vadodara, India, 2024, pp. 1-5, doi: 10.1109/PICET60765.2024.10716156.
- [8] J. Hwang and S. Oh, "A Brief Survey of Watermarks in Generative AI," 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2023, pp. 1157-1160, doi: 10.1109/ICTC58733.2023.10392465.
- [9] S. G. Prasad, V. C. Sharmila and M. K. Badrinarayanan, "Role of Artificial Intelligence based Chat Generative Pre-trained Transformer (ChatGPT) in Cyber Security," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 107-114, doi: 10.1109/ICAAIC56838.2023.10141395
- [10] S. -S. Chen, T. -W. Pai and C. -Y. Sun, "Applying the Diamond Model of Intrusion Analysis with Generative Pre-trained Transformer 3," 2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), PingTung, Taiwan, 2023, pp. 289-290, doi: 10.1109/ICCE-Taiwan58799.2023.10226923.
- [11] N. Rawal, P. Tavva and P. Selvakumar, "Enhancing Large Language Model Performance with Reinforcement Learning from Human Feedback: A Comprehensive Study on Q&A, Summarization, and Classification," 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET), Sydney, Australia, 2024, pp. 1-6, doi: 10.1109/ICECET61485.2024.10698396.
- [12] Y. Cao, B. Ivanovic, C. Xiao and M. Pavone, "Reinforcement Learning with Human Feedback for Realistic Traffic Simulation," 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 2024, pp. 14428-14434, doi: 10.1109/ICRA57147.2024.10610878.
- [13] X. Zhang, L. Zang, Q. Liu, S. Wei and S. Hu, "Triple-Based Data Augmentation for Event Temporal Extraction via Reinforcement Learning from Human Feedback," 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 2024, pp. 1627-1632, doi: 10.1109/CSCWD61410.2024.10580274.
- [14] H. -C. Chen, "Harnessing AI for Scientific Illustration: Exploring Tornado Dynamics Through Midjourney," 2023 12th International Conference on Awareness Science and Technology (iCAST), Taichung, Taiwan, 2023, pp. 136-141, doi: 10.1109/iCAST57874.2023.10359310.
- [15] A. Horváth, "Stable Diffusion with Memristive Cellular Neural Networks," 2024 IEEE International Symposium on Circuits and Systems (ISCAS), Singapore, Singapore, 2024, pp. 1-5, doi: 10.1109/ISCAS58744.2024.10558282.
- [16] S. S. Nair, "Redefining Creativity in Design: Exploring the Impact of AI-Generated Imagery on Design Professionals and Amateurs," 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2024, pp. 1723-1728, doi: 10.23919/INDIACom61295.2024.10498341.
- [17] J. Lv et al., "GPT4Motion: Scripting Physical Motions in Text-to-Video Generation via Blender-Oriented GPT Planning," 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2024, pp. 1430-1440, doi: 10.1109/CVPRW63382.2024.00150.
- [18] A. Madhubalan, A. Gautam and P. Tiwary, "Blender-GAN: Multi-Target Conditional Generative Adversarial Network for Novel Class Synthetic Data Generation," 2024 International Conference on Smart Applications, Communications and Networking (SmartNets), Harrisonburg, VA, USA, 2024, pp. 1-7, doi: 10.1109/SmartNets61466.2024.10577645.
- [19] A. Madhubalan, A. Gautam and P. Tiwary, "Blender-GAN: Multi-Target Conditional Generative Adversarial Network for Novel Class Synthetic Data Generation," 2024 International Conference on Smart Applications, Communications and Networking (SmartNets), Harrisonburg, VA, USA, 2024, pp. 1-7, doi: 10.1109/SmartNets61466.2024.10577645.

## Authors



Byong-Kwon Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanbat, Hannam and Chungbuk University Korea, in 2000, 2003 and 2007, respectively.

My main areas of interest are embedded systems, virtual and augmented reality (VR/AR), and artificial intelligence (AI). The field currently being studied is the construction of an exhibition hall using virtual reality. It is a technology that combines AI with cultural uniform restoration technology as a future research field.