2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 6-10, 2024. Sarawak, Malaysia

AutoForma: A Large Language Model-Based Multi-Agent for Computer-Automated Design

JianXing Liao[1], JunYan Xu[1], SiCheng He[1], ZeKe Chen[1], Shui Yu[1], and Yun Li*[1,2], Fellow, IEEE
[1]Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China
[2] i4AI Ltd, London WCIN3AX, United Kingdom

2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-1020-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SMC54092.2024.10831565

***Abstract*—With the proliferation of artificial intelligence, Computer-Aided Design (CAD) is being transformed into Computer-Automated Design (CAutoD). In this paper, the advent of Large Language Models (LLMs) introduces new opportunities for CAutoD. This study develops AutoForma, an LLM-based multi-agent system, for automatic conversion from natural language descriptions to 3D models. By harnessing the comprehension capabilities of LLMs, AutoForma streamlines the CAutoD workflow by efficiently translating design intents into precise models in CAD. Through a comprehensive set of evaluations, AutoForma is seen to offer automation performance across various design tasks, particularly in generating non-standard parts that meet specific requirements, with higher efficiency and accuracy than using just an LLM like GPT-4.**

***Index Terms*—CAutoD, LLM, Multi-Agent, Automated Design**

I. Introduction

Computer-Aided Design (CAD) leverages computer technology for the drafting, modeling, and simulation of physical designs. CAD software is a powerful tool that empowers engineers and designers to visualize, simulate, and analyze three-dimensional objects within a digital environment. This capability is crucial for creating detailed and precise models, which are essential for manufacturing, construction, and product development. With the advent of the Industry 4.0 era, the design and manufacturing sectors are undergoing a profound transformation towards digitization and intelligence [1]. The primary objectives of this shift include reducing production costs through modern technological innovations, more effectively meeting market and customer demands, and further liberating human resources by automating the design and manufacturing processes to decrease the cost of solution provision.

With the help of artificial intelligence, CAD is being transformed into Computer-Automated Design (CAutoD). As Large Language Models (LLMs) [2] such as GPT-3 [3], GPT-4, and Gemini continue to advance, they herald a technological revolution within the field of Natural Language Processing (NLP). The pre-training on extensive corpora equips LLMs with a foundational understanding of world knowledge and proficient coding capabilities. Concurrently, CAD software has incorporated the capability for code-based CAD, paving the way for the automation of model generation through coding.

Furthermore, LLM agents represent a significant breakthrough, enabling LLMs to autonomously invoke appropriate tools and functions to fulfill received tasks [4]. Mei et al. [5] have developed AIOS, an LLM agent operating system focused on optimizing resource allocation, facilitating context switching between agents, enabling concurrent execution, providing tool services, and maintaining access control. X. Wang et al. [6]introduced the CodeAct framework, which allows LLMs to generate executable Python code, enhancing coding efficiency through integration with the Python interpreter. Z. He et al. [7]'s ChatEDA employs the large language model AutoMage as an autonomous EDA agent, simplifying the design process from RTL to GDSII through task planning, script generation, and execution management. Collectively, the essence of these agent systems lies in transitioning from traditional human-centered decision-making models to intelligent decision processes led by large language models.

In this paper, we develop AutoForma, an LLM Multi-Agent system for the CAutoD, aiming at facilitating the automated transformation from natural language descriptions to 3D models. AutoForma employs LLM as the cornerstone for decision-making and execution, enabling users to effortlessly generate corresponding 3D models through straightforward textual descriptions, thereby significantly optimizing design process costs. The principal contributions of this research encompass:

1) Proposing AutoForma as the first LLM-based methodology for CAutoD;
2) Establishing a Multi-Agent system, setting a new standard for automation in the design domain;
3) Demonstrating AutoForma's superior performance in a series of comprehensive evaluations, where it outperforms renowned models like GPT-4 across various tasks.

This investigation not only propels advancements in CAD technology but also equips the Industry 4.0 era with strong

*Corresponding Author
Project supported by the Major Research Plan of the National Natural Science Foundation of China (Grant No. 92270105)

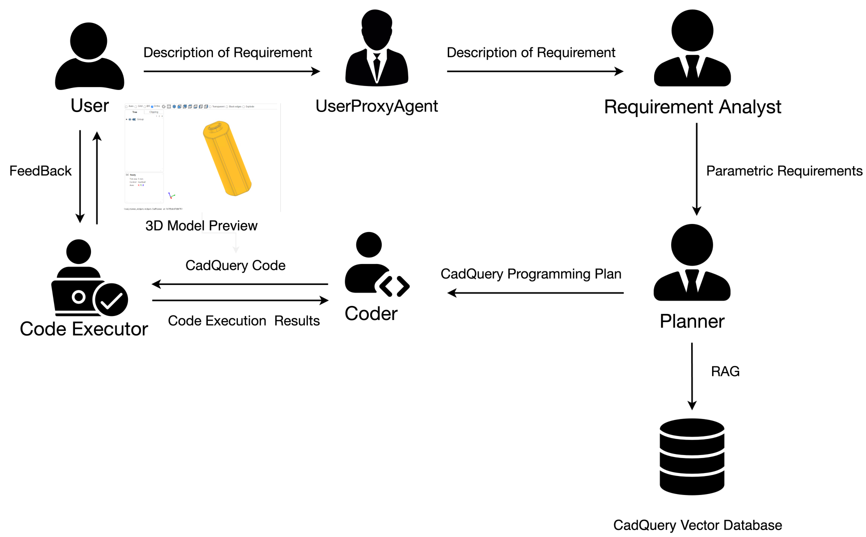


Fig. 1. Overview of AutoForma LLM agency structure. The workflow consists of four stages: requirements analysis, plan formulation, code writing, and execution verification. A multi-agent system composed of Requirement Analyst, Palner, Coder, and Executor.

technological foundations for the automation of design and manufacturing workflows.

II. AutoForma

AutoForma, a LLM agent tool, positions LLMs as the central drivers within the CAD application and decision-making spectrum. It introduces an efficient user interface that requires no prior knowledge of CAD modeling or operations from its users. By simply submitting textual modeling requests to AutoForma, the system leverages a Multi-Agent framework to autonomously interpret and fulfill these requests, subsequently producing 3D models that align with user expectations. Seamlessly integrated within AutoForma, CadQuery, a Python-based open-source 3D geometric modeling utility [8], facilitates rapid CAD model creation, analysis, and simulation. Through generating CadQuery scripts, AutoForma enables the effortless crafting of bespoke, parameterized models, thus enhancing the precision and versatility of CAD designs.

The workflow of the AutoForma LLM agent framework is illustrated in Fig. 1, encompassing four primary stages: requirement analysis, planning, code writing, and execution verification. Initially, the Requirement Analyst agent conducts a thorough analysis of the user's textual description, extracting key modeling parameters, and translating them into precise CAD design parametric requirements. Following this, the Planner agent devises comprehensive. CadQuery coding plans based on these requirements, identifying feasible requests, and subsequently providing specific execution steps to CADQuery engineers. Thereafter, the Coder agent transforms these plans into Python code, employing the CadQuery library to script solutions that fulfill the design requirements. Finally, the Executor agent implements these scripts, verifies the accuracy of the code, and presents the 3D model to the user for further modifications or confirmation of the design outcome.

The construction of AutoForma's Multi-Agent system utilizes Microsoft's AutoGen framework [9]. AutoGen introduces a multi-agent conversational framework as a sophisticated abstraction layer, facilitating the streamlined construction of LLM workflows with enhanced convenience. The Multi-Agent system is comprised of the following agents: Requirement Analyst, Planner, Coder, and Executor.

A. *Requirement Analyst*

In the CAD design process, the transformation of user requirements into parameterized forms is crucial for ensuring design precision and efficiency. By specifying the exact specifications and performance requirements, parameterized demands provide precise input parameters for the automated design workflow, thereby significantly reducing errors during the design process and enhancing the quality of the final outcomes. Leveraging LLMs to analyze textual descriptions submitted by users and transform them into parameterized requirements for the CAD modeling process, this method delves deeply into and understands users' modeling needs, accurately extracting key information related to the geometric dimensions, shapes, and internal structures of parts or components. This provides foundational support for subsequent planning and code writing.

B. *Planner*

Within the AutoForma LLM agent framework, the Planner agent plays a crucial role in transforming abstract parameterized requirements, refined during the requirement analysis phase, into actionable operational plans. For those feasible requirements, the Planner agent is tasked with formulating a series of detailed, sequential action plans. For each step, the Planner can utilize the CadQuery documentation retrieval function to provide use cases for the function, aiming to offer clear standard coding guidelines and recommendations to CadQuery code writing engineers. This approach ensures the precise translation of users' design intentions into actual CAD models.

1) Retrieval-Augmented Generation Based on CadQuery: In the AutoForma framework, retrieval-augmented generation (RAG) based on CADQuery plays a crucial role in optimizing the automated design process. By employing web scraping, CADQuery documentation, function descriptions, and use cases are collected, vectorized, and stored in a database. A pre-processing mechanism parses and extracts structured information, outputting JSON-formatted results. The Planner utilizes RAG to query and embed this information into planning documents, providing precise guidance for subsequent coding and model construction, thereby enhancing the accuracy and efficiency of the automated design process.For a detailed workflow, refer to Fig. 2.

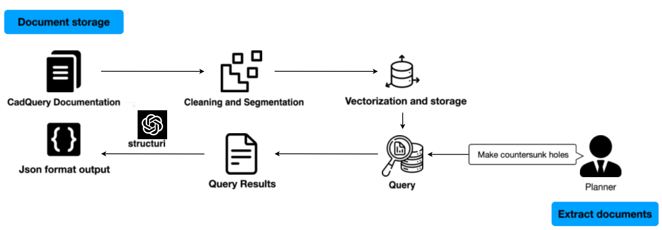


Fig. 2. Illustrates the document retrieval and processing workflow within the AutoForma LLM agent framework, depicting the entire process from document storage to extraction.

C. Coder

The Coder agent is responsible for converting the Planner's plans into executable Python code. Designed as a technical expert proficient in Python programming and the CADQuery library, the Coder develops comprehensive, standalone scripts for specific CAD modeling tasks. Additionally, the Coder collaborates with the Executor agent to ensure accurate code execution by running scripts, addressing errors, and resubmitting optimized code as needed. This coordination between the Coder and Executor ensures the generation of reliable and efficient Python scripts for the automated CAD modeling process.

D. Executor

The Executor agent is tasked with running 3D modeling code generated by the Coder and displaying the resulting CAutoD model. It employs automated methods to verify code accuracy without user intervention. If errors occur during execution, the Executor captures and relays error information to the Coder for necessary adjustments. Upon successful execution, the Executor presents the 3D model to the user, who can propose modifications or terminate the program. This interactive process bridges technical execution and user engagement, ensuring design outcomes align with user requirements and expectations.

III. EXPERIMENTS

We assess AutoForma's proficiency in generating CadQuery code by engaging it in a suite of user-defined CAD modeling tasks designed to test its versatility and accuracy. These tasks range from generating basic geometric shapes and executing advanced geometric operations to constructing simple models. For this evaluation, we have selected a suite of models for testing, which includes AutoForma , GPT-4, GPT-3.5, Gemini Pro, and Tongyi Qwen. AutoForma, driven by GPT-4, will be deployed in conjunction with GPT-4 in a series of ablation studies to elucidate its effectiveness across the following tasks.

To objectively evaluate the custom-generated 3D models, we have devised a comprehensive set of assessment criteria that combine subjective judgment with objective metrics. These criteria are divided into four levels:

- Grade A (10 points): Full compliance, with successful code execution and adherence to all predefined standards.
- Grade B (6-9 points): Models that function properly, maintaining essential functionality and design intent, albeit with minor deviations in non-critical parameters.
- Grade C (1-5 points): Code executes, but models require further adjustments to meet basic design standards due to failure in achieving several key parameters.
- Grade D (0 points): Code execution fails or models significantly deviate from requirements, resulting in unable to use.

Each SubTask will be executed ten times. he scoring formula is as follows:

$$\text{Total Score} = \frac{\sum_{i=1}^{n} \text{Score of SubTask}_i}{n} \times 10 \qquad (1)$$

where n is the total number of SubTask, and $\text{Score of SubTask}_i$ represents the score obtained for the ith sub-task. The proportions of difficulty levels for each SubTask are presented in Fig. 5.

A. Task 1: Basic CAD Operations

In Task 1, we investigate the performance of various language models in executing fundamental CAD operations, which include high-precision geometric operations and the construction of complex components. We will execute the two SubTasks outlined in Table II and calculate the average score as well as the distribution of scores across different grades in Table .

AutoForma demonstrates effective model generation that meets operational conditions through reasoned parameter

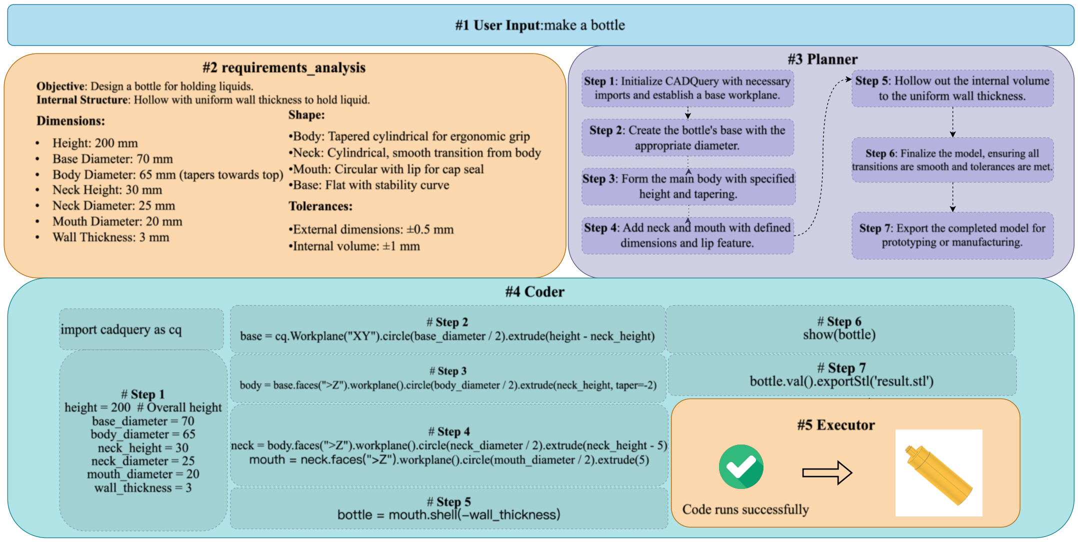


Fig. 3. The AutoForma uses the Muti-Agent system to automate the process of designing bottles

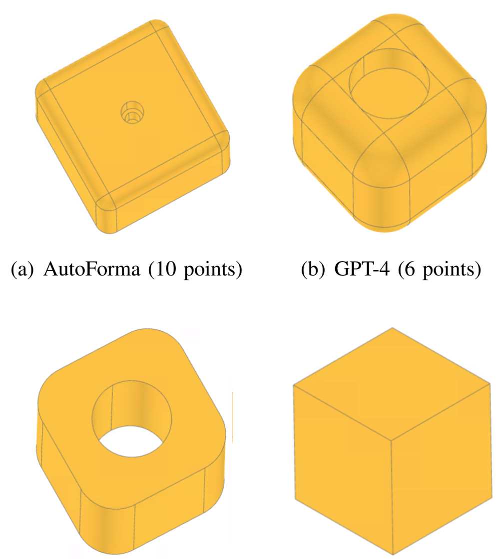
TABLE I
TASK 1 SUBTASK LIST

Task Number	Object	Operation	Parameters
SubTask 1	Cube	Counterbore and Fillet Application	Edge Length: 20 mm
			Fillet Radius: 5 mm
			Hole Diameter: 2 mm
			Counterbore Diameter: 3 mm
			Counterbore Depth: 5 mm
			Hole Depth: 10 mm
SubTask 2	Cylinder Top	Countersink Hole Machining	Diameter: 6 mm
			Depth: 10 mm
			Feature: Flat bottom

analysis and planning. In contrast, GPT-4 primarily produces foundational models in SubTasks 1 and 2, failing to perform CAD operations, which results in lower scores, see table III for task1 scores.

As illustrated in SubTask 1, the best 3D models generated by LLMs are depicted in the figure 4. The requirement is to create a model that conforms to the parameters specified for Counterbore and Fillet Application. AutoForma impeccably completes the task, achieving a score of 8.2 and producing two models rated 'A' in the tests, thanks to the close collaboration within its Multi-Agent framework that generates code strictly in line with task requirements. GPT-4, however, falls short in dealing with more complex tasks, unable to effectively meet the established standards across various parameters, only managing to produce 'B' grade models thrice, without ever perfectly meeting all parameter requirements, hence scoring merely 2.2.

The lower performance of models such as GPT-3.5, Gemini, and ProTongyi Qwen is primarily attributed to an insufficient success rate in code generation. This outcome suggests a need for enhanced comprehension and code-generation capabilities in handling complex CAutoD modeling operations.



(a) AutoForma (10 points) (b) GPT-4 (6 points)

(c) GPT-3.5 (3 points) (d) Gemini (1 points)

Fig. 4. Best results for Task1——SubTask1 and the 3D models scores. TongYi Qwen failed to generate the model in this SubTask.

B. *Task 2: Non-Standard Part Model Generation*

Within the framework of this study, the automatic generation of non-standard components (non-standard parts) is considered a core component. This test involves two SubTasks that examine the large language model's ability to generate non-standard part models. The tasks are listed in table II.

1) SubTask 1: SubTask 1 entails the design of a reinforced pipe connector with counterbore holes, demanding precise

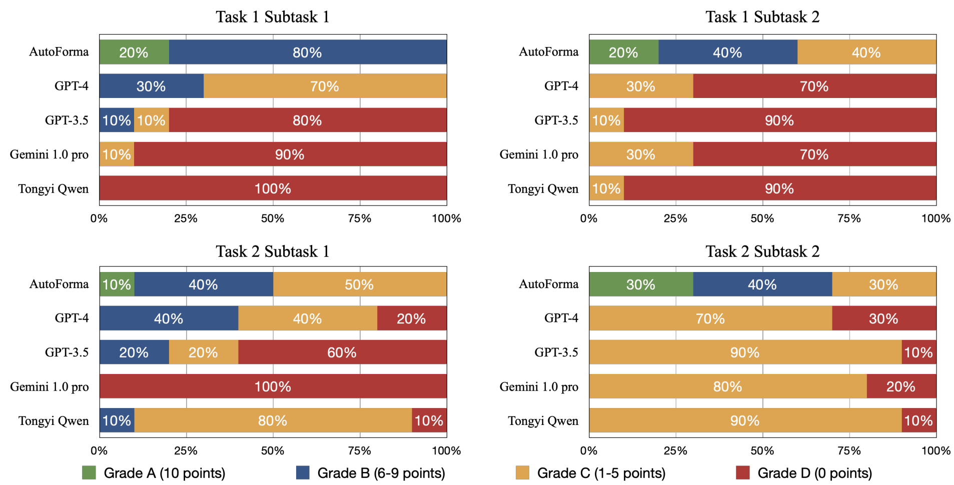


Fig. 5. The performance comparison between AutoForma and other LLMs)across individual SubTasks.

TABLE II
TASK 2 SUBTASK LIST

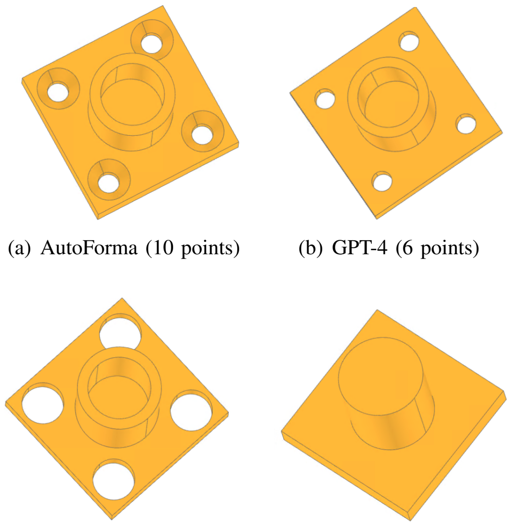
Task Number	Object	Parameters	
SubTask1	Reinforced Pipe Connector with Countersunk Holes	Base Plate	Length: 15.0 mm
			Width: 15.0 mm
			Thickness: 2.0 mm
		Countersunk Holes	Countersink Diameter: 4.0 mm
			Countersink Angle: 82 degrees
			Hole Diameter: 2.0 mm
			Position: Located at the corners of a 10.0 mm x 10.0 mm rectangle on the top face of the base plate.
		Pipe	The pipe needs to be connected to the base plate
			Diameter: 8.0 mm
			Height: 10.0 mm
			Central Hole Diameter: 6.0 mm
SubTask2	Generate a bottle	No parameters are specified, and no operations are specified.	

geometry, dimensional accuracy, and multi-component assembly.In the evaluation of SubTask 1, significant differences were observed in the reinforced tube joint models generated by five language models. We take the highest score for analysis, and the average score is shown in Table III .As depicted in Figure 6, AutoForma (Fig. 6-a) successfully produced a model that met all specified parameter requirements. All countersunk holes were precisely placed with correct dimensions and locations, and the central tube's size complied with the predetermined specifications, thus earning a perfect score of 10. GPT-4 (Fig. 6-b) only achieves a score of 6 due to deviations in the precision and positioning of counterbored holes. Additionally, the dimensions of the central tube did not precisely meet the preset specifications. For GPT-3.5 (Fig. 6-c)and TongYi Qwen (Fig. 6-d), the models exhibited lower quality, with dimensions and positions not aligning with the requirements, and the construction of the central tube also failed to conform to the specified parameters.Gemini did not successfully execute the code in this test, resulting in a score of 0.

TABLE III
THE SCORE OF EACH MODEL ON EACH TASK

models	task1			task2		
	SubTask 1	SubTask 2	Score	SubTask1	SubTask2	Score
AutoForma	8.2	5.1	**66.5**	7.8	6.3	**70.5**
GPT-4	2.2	1.2	17	4.5	3.6	34
GPT-3.5	1	0.2	6	2.5	2.3	20
Gemini 1.0 pro	0.2	0.8	5	0	0.5	2.5
Tongyi Qwen	0	0.2	1	0.4	1.6	10

2) SubTask 2: In the evaluation of Sub-Task 2, large language models were tasked with generating 3D models of a water bottle, without predefined parameters and operations. The assessment focused on the models' ability to precisely analyze the parameters of the bottle and generate code accordingly. The AutoForma generation process, illustrated in Fig. 3, commences with a user input to create a bottle. Subsequently, the AutoForma system conducts a requirement analysis to define the design objective, taking into account dimensions and shape features. Tolerances for external dimensions and internal capacity are rigorously defined to ensure precision. The planning phase delineates the necessary steps for model generation, ranging from initializing CadQuery to constructing the bottle base, forming the main body, and meticulously adding the neck and mouth of the bottle. In the coding phase, a programmer agent transforms the plan into executable code. Upon successful execution of the code, the model is confirmed to have been accurately created and is represented through a generated 3D model. The model produced by GPT-4 not only meeting the design requirements but also achieving functional necessity by hollowing out the interior cavity for liquid storage. However, it also generated some unrelated auxiliary accessories. In contrast, despite resembling a bottle, other models failed to fulfill



(a) AutoForma (10 points) (b) GPT-4 (6 points)

(c) GPT-3.5 (4 points) (d) TongYi Qwen (3 points)

Fig. 6. Best results for Task2-subTask1 and the 3D models scores. Gemini failed to generate the model in this subTask.

the basic functional requirement of containing liquid due to their solid internal structure, resulting in lower scores. The Gemini system was only capable of generating basic cylindrical shapes and was ineffective in accomplishing the bottle modeling task.

IV. Conclusion

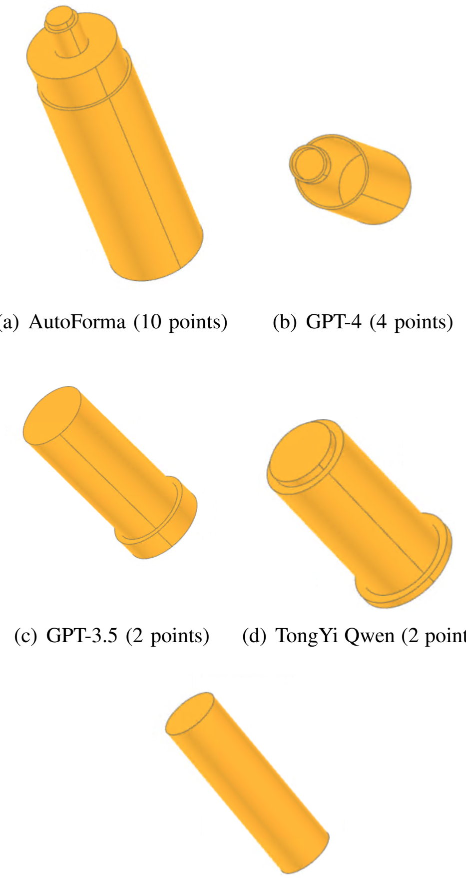
Enhancing design efficiency, CAutoD is a pivotal area of research. This paper introduces AutoForma, an innovative Multi-Agent system underpinned by LLMs, designed to automate the conversion of natural language descriptions into 3D CAutoD models. By harnessing the advanced natural language processing capabilities of LLMs, AutoForma markedly enhances the CAutoD design workflow, simplifying the design process and enabling the efficient creation of precise, non-standard part models with remarkable accuracy. Our research demonstrates that AutoForma outperforms current LLM models in automating complex CAutoD tasks, signifying a significant advancement in CAutoD technology. It exemplifies the transformative potential of LLMs in the field of design, heralding a new era of more intuitive and user-friendly CAutoD solutions.

References

[1] Y. Chen and Y. Li, *Computational intelligence assisted design: in industrial revolution 4.0*. CRC Press, 2018.

[2] A. Birhane, A. Kasirzadeh, D. Leslie, and S. Wachter, "Science in the age of large language models," *Nature Reviews Physics*, vol. 5, no. 5, pp. 277–280, 2023.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.



(a) AutoForma (10 points) (b) GPT-4 (4 points)

(c) GPT-3.5 (2 points) (d) TongYi Qwen (2 points)

(e) Gemini 1.0 pro (1 points)

Fig. 7. Best results for Task2-SubTask2 and the 3D models scores. Bottles generated by AutoForma and GPT are hollow inside, while bottles generated by other models are solid inside.

[4] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, pp. 1–26, 2024.

[5] K. Mei, Z. Li, S. Xu, R. Ye, Y. Ge, and Y. Zhang, "Aios: Llm agent operating system," *arXiv e-prints*, pp. arXiv–2403, 2024.

[6] X. Wang, Y. Chen, L. Yuan, Y. Zhang, Y. Li, H. Peng, and H. Ji, "Executable code actions elicit better llm agents," no. arXiv:2402.01030, Feb. 2024, arXiv:2402.01030 [cs].

[7] Z. He, H. Wu, X. Zhang, X. Yao, S. Zheng, H. Zheng, and B. Yu, "Chateda: A large language model powered autonomous agent for eda," in *2023 ACM/IEEE 5th Workshop on Machine Learning for CAD (MLCAD)*. IEEE, 2023, pp. 1–6.

[8] A. Urbańczyk *et al.*, "Cadquery: A python parametric cad scripting framework," 2024, available online: https://github.com/CadQuery/cadquery (accessed on April 1, 2024).

[9] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang, "Autogen: Enabling next-gen llm applications via multi-agent conversation framework," *arXiv preprint arXiv:2308.08155*, 2023.