Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 29-30, 2026

Automated Constructive Solid Geometry Based McCARD Input Generation Using Representative Large Language Models

Tae Min Kim [a] and Ho Jin Park [a*]
a Kyung Hee University, 1732 Deokyoungdaero, Giheung-gu, Yongin-Si, Gyeonggi-do, Korea, 17104
**Corresponding author: parkhj@khu.ac.kr*

***Keywords:** Large Language Model, Monte Carlo, McCARD, Automated Input Generation, Chat KHU

1. Introduction

Next-generation nuclear power systems, envisioned as carbon-free and enhanced-safety power sources, must often be designed in the absence of experimental data. For the development and verification of such systems, or of systems without prior design analyses, Monte Carlo (MC) particle transport methods, which allow accurate modeling of geometry and continuous-energy nuclear reaction cross sections, are widely used as a reference. However, obtaining accurate results from high-fidelity MC transport analysis requires precise geometry modeling, and the repetitive construction and modification of the complex Constructive Solid Geometry (CSG) inputs typically used in MC transport codes is a bottleneck that requires substantial time and manpower resources.

To solve this problem, this study preliminarily investigates whether a Large Language Model (LLM) can be used to automatically generate inputs for CSG-based MC analyses (i.e., McCARD [1]) from natural language instructions. In this study, three LLMs, Claude (Anthropic) [2], Gemini (Google) [3], and ChatGPT (OpenAI) [4], were used to generate MC inputs for VERA benchmark problems [5].

2. Methods

2.1 VERA Benchmark Overview

The VERA (Virtual Environment for Reactor Applications) Core Physics Benchmark Progression is a set of benchmark problems [5] proposed by the Consortium for Advanced Simulation of Light Water Reactors (CASL), based on the Westinghouse 17×17 pressurized water reactor (PWR) Watts Bar Unit 1. The progression consists of ten problems, ranging from a two-dimensional single fuel pin cell (i.e., problem #1) to full-core depletion calculations (problem #10), with each stage designed to incrementally validate the modeling capabilities required of a reactor analysis code. In this study, problem #1 (2D single fuel pin cell), problem #2 (2D fuel lattice), problem #3 (3D single fuel assembly), and problem #4 (control rod worth test on a 3×3 assembly array) were used as benchmarks to assess LLM capabilities for McCARD input generation. In addition to the VERA benchmark problems, Flattop-233 critical experiment benchmark [6] was also used. This benchmark consists of a spherical ^{233}U fuel core surrounded by a normal-uranium reflector.

2.2 LLM-Based Input Generation Workflow

In this study, LLMs were accessed through Chat KHU [7], Kyung Hee University's integrated platform for multiple LLMs, including Claude, ChatGPT, and Gemini. Figure 1 shows the Chat KHU Access Interface screen. Chat KHU provides free access to various versions of LLMs from multiple vendors within a specified token limit.

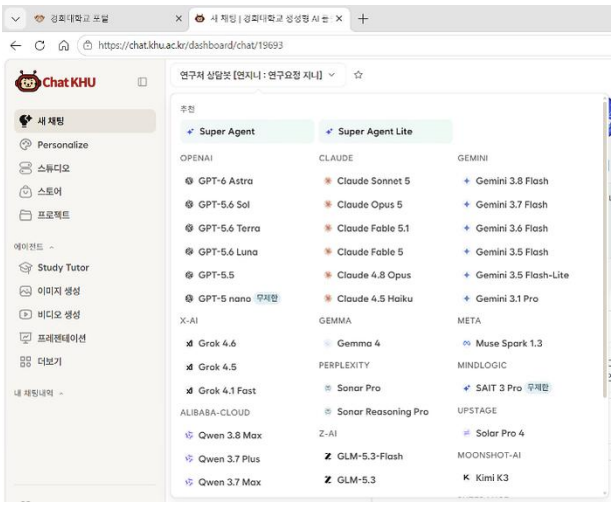


Fig. 1. Chat KHU main interface

Three steps were considered to convert natural-language instructions into McCARD input files: (1) analysis of the problem description, including the relevant physics, material specifications, and geometric conditions; (2) generation of a McCARD input file from the analysis results and user instructions; and (3) validation of the McCARD results obtained with the generated input file. This approach eliminated differences in image and document extraction capabilities across LLMs as a potential confounding factor in the assessment of input-generation capability. A standardized text format for all benchmark problems ensured an equal basis for model comparison. The McCARD User's Manual Version [8] was provided in full to all three LLMs as reference material for McCARD input generation and validation. All three LLMs received identical natural-language instructions in accordance with the three-step procedure described above. An input file was deemed successful if McCARD executed it without input-related errors and produced a physically

valid multiplication factor.

2.3 Comparison of Claude, Gemini, and ChatGPT

The comparison in this study represents the performance of the complete LLM-based workflow system rather than the reasoning capability of each LLM alone. The workflow comprises each model and its associated code execution and error-diagnosis environment. Claude and ChatGPT were evaluated together with an agentic environment capable of directly executing and self-correcting McCARD, whereas Gemini was evaluated purely on its text-generation ability without such an execution environment. Accordingly, the results of this comparison should be interpreted as reflecting differences between input-generation workflows rather than differences in the intrinsic reasoning capability of LLMs.

For each LLM provider, the latest model available on Chat KHU as of August 2026 was selected. All three models were specialized for code generation. However, as described below, their support for agentic execution environments differed by model. Claude (Claude Code, Sonnet 5) and ChatGPT (Codex, 5.6 Sol) were able to run McCARD directly with their generated inputs and independently diagnose and correct errors, whereas Gemini (Gemini 3.1 Pro, using the Gem feature) could not run McCARD directly, requiring the user to supply the execution results back to the model for validation. This is because Claude Code and Codex provide agentic execution environments capable of operating on an actual file system and terminal, whereas Gemini's Gem feature can only perform limited Python calculations within its own sandbox, without permission to execute external programs. Figure 2 shows the Claude Code execution screen.

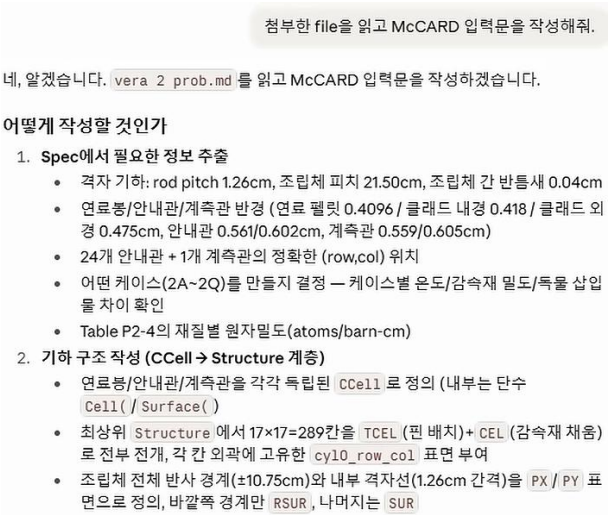


Fig. 2. Screen for Claude Code LLM executions

No limit was placed on the number of iterative debugging attempts for Claude and ChatGPT. During the initial stage, the generated inputs were manually reviewed by the user, who identified incorrect or missing information and instructed the models to revise the inputs. After the initial stage, Claude and ChatGPT directly executed McCARD, analyzed the resulting output files, and iteratively corrected their inputs within their agentic environments. Because Gemini could not execute McCARD directly, the user executed its generated inputs externally and provided the resulting output files to Gemini for correction and validation. The VERA subproblems were evaluated sequentially within the same conversation, with the context from preceding subproblems retained. Representative early-stage errors included omission of required temperature values and the use of 300 K instead of the specified 293 K.

3. Results

Overall, Claude and ChatGPT consistently produced valid McCARD inputs with smaller k_{eff} deviations than Gemini across the benchmark problems. This performance gap widened substantially as problem complexity increased, and Gemini failed to produce valid results for problems beyond VERA problem #1. Table I compares the errors in effective multiplication factor (k_{eff}) for the inputs generated by Claude, ChatGPT, and Gemini.

Table I: Errors in k_{eff} relative to the reference solutions for Flattop and VERA problem #1 (pcm)

Problem	Error in multiplication factor (pcm)		
	Claude	ChatGPT	Gemini
Flattop	69	69	69
Problem #1A	57	84	175
Problem #1B	16	13	125
Problem #1C	55	65	153
Problem #1D	51	55	143
Problem #1E	17	34	57
RMS errors	43.4	55.9	136.6

* The stochastic errors for k_{eff} are less than 10 pcm

Table II: Errors in k_{eff} relative to the reference solutions for VERA Problem #2 (pcm)

Problem	Error in multiplication factor (pcm)		
	Claude	ChatGPT	Gemini
Problem #2A	77	71	Failed
Problem #2B	18	20	
Problem #2C	60	45	
Problem #2D	34	25	
Problem #2E	1	1	
Problem #2F	35	49	
Problem #2G	7	3	
Problem #2H	49	41	
Problem #2I	30	27	
Problem #2J	20	41	
Problem #2K	26	33	
Problem #2L	27	11	
Problem #2M	18	10	
Problem #2N	4	7	

Problem #2O	37	37	
Problem #2P	78	97	
Problem #2Q	20	52	
RMS errors	38.8	41.6	

* The stochastic errors for k_{eff} are less than 10 pcm

On simple problems, all three models were effectively indistinguishable: for the Flattop problem, all three produced identical 69 pcm differences from the reference solution. However, even a slight increase in problem complexity produced a clear performance gap. Across the five subproblems of VERA problem #1, Claude achieved the lowest RMS error of 43.4 pcm, followed by ChatGPT at 55.9 pcm, whereas Gemini produced a substantially larger RMS error of 136.6 pcm—more than twice as large.

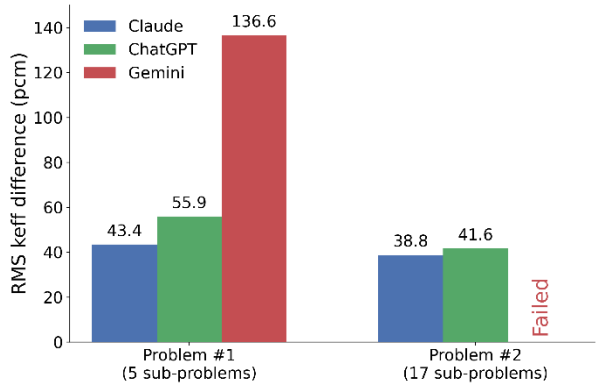


Fig. 3: RMS differences in k_{eff} for each LLM in VERA problems #1 and #2 (Gemini was excluded from problem #2 because it failed to produce a valid McCARD input.)

Table II compares the errors in k_{eff} in VERA problem #2, while Figure 3 shows the RMS differences in k_{eff} for each LLM in problems #1 and #2. The performance gap observed in VERA problem #1 became decisive in VERA problem #2. Gemini failed to produce a valid McCARD input from problem #2A onward, whereas Claude and ChatGPT produced valid inputs for all 17 subproblems, with comparable RMS errors of 38.8 and 41.6 pcm, respectively. This result demonstrates that the gap between Gemini and the other two models does not just widen but eventually becomes a binary success/failure gap as problem complexity increases.

Table III: Errors in k_{eff} relative to the reference solutions for VERA problems #3 and #4

Problem	**Error in multiplication factor (pcm)**		
	Claude	**ChatGPT**	**Gemini**
Problem #3A	42	6	Failed
Problem #3B	98	67	
Problem #4	76	66	
RMS errors	75.6	54.4	

* The stochastic errors for k_{eff} are less than 10 pcm

This pattern held at the highest complexity level tested: for VERA problem #3 (3D single fuel assembly) and problem #4 (control rod worth test on a 3×3 assembly array), Claude and ChatGPT again generated valid inputs with k_{eff} differences of 6-98 pcm (RMS errors of 75.6 pcm and 54.4 pcm for Claude and ChatGPT, respectively), while Gemini—having already failed at problem #2—could not be evaluated at all. For most successfully executed cases, the geometry and material specifications of the generated inputs were manually checked against the corresponding benchmark descriptions, and no major discrepancies were identified. However, agreement in k_{eff} alone does not necessarily indicate that all geometry and material definitions are correct, because different modeling errors may compensate for each other and produce similar multiplication factors. Therefore, careful verification of LLM-generated inputs by an experienced user remains necessary.

Table IV: Elapsed time and token usage for generating inputs for VERA Problem #1A-1E (median of 3 trials)

Model	**Elapsed Time**	**Tokens Used**
Claude	3 min 49 s	~6,065 tokens
ChatGPT	4 min 40 s	~3,173 tokens
Gemini	2 min 13 s	~3,850 tokens
Gemini (debugging)	1 min 23 s	~3,700 tokens

Table IV compares the elapsed time and token usage for generating inputs for VERA problem #1A-1E. Claude required 3 min 49 s and approximately 6,065 tokens; ChatGPT required 4 min 40 s and approximately 3,173 tokens; and Gemini required 2 min 13 s and approximately 3,850 tokens. However, an error in Gemini's initial input required an additional 1 min 23 s and approximately 3,700 tokens for error correction. Because token definitions differ among LLMs, direct token-count comparison does not provide a reliable measure of McCARD input-generation performance. Therefore, token consumption was assessed according to the model-specific standards established by Chat KHU.

It should also be noted that the McCARD calculations in this study used the ENDF/B-VII.1 nuclear data library, whereas the VERA benchmark reference solutions were computed using the ENDF/B-VII.0 library. Since cross-section updates for certain nuclides differ between these library versions, this library mismatch may have partially contributed to the observed differences in k_{eff}. In addition, the VERA benchmark reference solutions were computed using 1.1×10^{8} particle histories (100,000 particles per generation over 1,100 generations, with 100 generations skipped), which is reported to yield a statistical uncertainty of approximately 8 pcm in k_{eff}. Accordingly, part of the pcm differences observed in this study may be attributable to the library mismatch and the inherent statistical uncertainty of the MC calculations, rather than solely reflecting the accuracy of the generated inputs.
To reduce the effects of these differences and provide a more direct comparison, researcher-authored McCARD

inputs were additionally prepared for VERA problems #1 and #2. These inputs and the LLM-generated inputs were calculated using the same McCARD code and ENDF/B-VII.1 nuclear data library. Table V compares the k_{eff} differences between the LLM-generated inputs and the researcher-authored inputs.

Table V: Errors in k_{eff} relative to the researcher-authored McCARD inputs for VERA problems #1 and #2

Problem	Error in multiplication factor (pcm)		
	Claude	ChatGPT	Gemini
Problem #1A	22	5	96
Problem #1B	14	11	123
Problem #1C	6	16	104
Problem #1D	22	26	114
Problem #1E	0	17	74
RMS errors	15.5	16.5	103.6
Problem #2A	0	6	Failed
Problem #2B	16	14	
Problem #2C	11	26	
Problem #2D	2	11	
Problem #2E	0	0	
Problem #2F	4	10	
Problem #2G	2	11	
Problem #2H	30	22	
Problem #2I	10	7	
Problem #2J	4	17	
Problem #2K	14	21	
Problem #2L	11	5	
Problem #2M	10	2	
Problem #2N	8	11	
Problem #2O	26	26	
Problem #2P	23	42	
Problem #2Q	49	17	
RMS errors	18.0	17.8	

* The stochastic errors for k_{eff} are less than 10 pcm

For VERA problem #1, the RMS differences relative to the researcher-authored McCARD inputs were 15.5 pcm for Claude, 16.5 pcm for ChatGPT, and 103.6 pcm for Gemini. For VERA problem #2, the corresponding RMS differences were 18.0 pcm for Claude and 17.8 pcm for ChatGPT, while Gemini failed to generate a valid McCARD input. Compared with the published VERA reference solutions, the RMS differences for Claude and ChatGPT were substantially reduced. This result suggests that part of the differences observed in the previous comparison may be associated with differences in the transport code and nuclear data library rather than solely with the generated inputs.

4. Conclusions

In this study, the preliminary results confirm that LLMs can produce CSG-based input files for the MC particle transport code McCARD from natural-language instructions. In particular, the ability of Claude and ChatGPT to execute McCARD and correct their own errors enabled a more autonomous workflow than Gemini, for which the user had to execute McCARD externally and provide the resulting outputs to the model. This performance gap widened further from VERA problem #2 onward. The comparison with researcher-authored McCARD inputs further showed smaller k_{eff} differences for Claude and ChatGPT under consistent code and nuclear data conditions.

These results suggest that LLMs can serve as a powerful auxiliary tool for reducing human error and lowering the barrier to input generation in reactor design workflows. However, LLM-generated MC code inputs may contain errors, particularly during the early stages of implementation. Accordingly, careful user review remains essential to identify and correct such errors. Therefore, LLMs have the potential to mitigate the time and manpower costs associated with CSG-based MC code input construction. Future work will include the generation and verification of inputs for more complex next-generation nuclear system models.

Acknowledgment

This work was supported by the 2026 University Innovation Support Project funded by the Ministry of Education (MOE).

REFERENCES

[1] H. J. Shim et al., McCARD: Monte Carlo Code for Advanced Reactor Design and Analysis. *Nucl. Eng. Technol.*, Vol. 44, pp.151-176, 2012.
[2] Anthropic, “Claude,” Anthropic, https://claude.ai/, accessed Aug. 17, 2026.
[3] Google, “Gemini,” Google, https://gemini.google.com/, accessed Aug. 17, 2026.
[4] OpenAI, “ChatGPT,” OpenAI, https://chatgpt.com/, accessed Aug. 17, 2026.
[5] A. T. Godfrey, *VERA Core Physics Benchmark Progression Problem Specifications*, Rev. 4, CASL-U-2012-0131-004, Consortium for Advanced Simulation of Light Water Reactors, Aug. 29, 2014.
[6] R. W. Brewer, Benchmark Critical Experiment of a Uranium-233 Sphere Reflected by Normal Uranium with Flattop, U233-MET-FAST-006, Rev. 1, NEA/NSC/DOC(95)03/V, Sep. 30, 2000.
[7] Kyung Hee University Library, “ChatKHU User Guide,” Mar. 23, 2026.
[Online]. Available: https://libguides.khu.ac.kr/ai/chatKHU
[8] H. J. Shim and C. H. Kim, McCARD User’s Manual, Rev02, Seoul National University, 2018.