Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #45533

BOARD #134: Results and Evaluation of an Early LLM Benchmarking of our ECE Undergraduate Curriculums

Dr. Peter Jamieson, Miami University

Dr. Jamieson is an assistant professor in the Electrical and Computer Engineering department at Miami University. His research focuses on Education, Games, and FPGAs.

Dr. George D. Ricco, Miami University

George D. Ricco is an engineering education educator who focuses on advanced analytical models applied to student progression, and teaching first-year engineering, engineering design principles, and project management.

Brian A Swanson, Miami University
Dr. Bryan Van Scoy, Miami University

Bryan Van Scoy is an assistant professor in the Department of Electrical and Computer Engineering at Miami University whose research focuses on algorithms in optimization and control.

Results and Evaluation of an Early LLM Benchmarking of our ECE Undergraduate Curriculums

Abstract

The rapid integration of Artificial Intelligence (AI) into engineering practice necessitates critically examining our educational approaches. This paper presents an investigation into the performance of Large Language Models (LLMs) within the context of our Electrical Engineering (EE) and Computer Engineering (CpE) undergraduate curricula at Miami University. Our study addresses a fundamental question: How do current AI tools perform on typical course assessments, and what implications does this have for curriculum design?

We introduce a systematic methodology for benchmarking LLM performance on our course assessments, including exams, assignments, and projects. Utilizing state-of-the-art LLMs, we evaluate their capabilities across core courses in our EE and CpE programs. This includes Circuits I (ECE 205), Digital Design (ECE 287), Energy Systems (ECE 291), and Signals and Systems (ECE 306). Our benchmarking results reveal the strengths and limitations of these AI tools in engineering education tasks, providing insights for curriculum adaptation. We discuss how these results might inform the evolution of engineering education, highlighting areas where AI could enhance learning and where human skills should be reinforced. This work contributes to the ongoing dialogue on AI integration in engineering education. It offers a first step in providing a replicable framework for continuously assessing AI capabilities in academic settings and how this activity can aid educators. As we navigate the transforming landscape of engineering practice and education, such benchmarking efforts are essential for ensuring our curricula remain relevant and effective in preparing the next generation of engineers for an AI-augmented profession.

1 Introduction

Artificial Intelligence (AI) is rapidly evolving, reshaping industries, and challenging traditional paradigms across various fields, including education. As educators, we face a dual responsibility: to harness AI's potential in our own careers and to prepare our students for theirs. The emergence of Large Language Models (LLMs) like GPT-4 and Claude represents a significant leap in AI capabilities, where most individuals in the first world can easily access these tools. These AI technologies necessitate reevaluating our teaching methodologies, curriculum design, and the skills we impart to our students to ensure their readiness for this modern workforce [1].

To effectively integrate AI into our educational frameworks and curricula, we must first establish

a baseline understanding of how current AI technologies, particularly LLMs, perform within our existing educational structures. This understanding is crucial as it provides insight into the strengths and limitations of AI in tackling the types of problems and assessments we currently use to evaluate student learning. If anything, we should understand why we ask students to perform different assessments and explain why an AI tool might not benefit the student's development. By benchmarking LLM performance against our current curriculum, we can identify areas where AI excels, where it falls short, and how it might influence the future direction of our educational programs.

This paper presents an early benchmarking study of LLM performance across various courses in our Electrical Engineering (EE) and Computer Engineering (CpE) undergraduate curricula. We have systematically evaluated how advanced LLMs perform on various assessments in our subset of courses. Our results, though preliminary and covering only a subset of our courses, provide valuable insights into the current capabilities of AI in tackling engineering education tasks. These findings suggest areas where our curriculum may need to evolve to both leverage AI capabilities and teach skills that remain uniquely human.

The contributions of this work are:

1. An applied methodology for benchmarking LLM performance against engineering course assessments.
2. Initial performance data of an advanced LLM on EE and CpE course materials.
3. Insights into potential curriculum adjustments in light of AI advancements.
4. A framework for ongoing evaluation of AI capabilities in an educational context.

The remainder of this paper is organized as follows: Section 2 provides background on AI models, curriculum analysis, and design. Section 3.1 introduces Jamieson's LLM Prompt Taxonomy, which we use to classify our prompts. Section 3 details our benchmarking methodology. In Section 4, we present an overview of our EE and CpE curricula and note which courses have been evaluated in this study. Section 5 showcases our benchmarking results. We discuss the implications of our findings in Section 6 and conclude with future directions in Section 7.

2 Background – AI model, Curriculum Analysis and Curriculum Design

To place this work into context, we briefly provide our brief description of what AI models are focusing on the emergence of LLM-based chatbots, we provide a brief idea of what a learning objective is, and we briefly look at curriculum design and analysis as well as electrical and computer engineering curriculums.

2.1 AI and LLMs – what are they?

This work focuses on using the capabilities of modern LLMs to perform the assessments in a course/curriculum. We use Agrawal, Gans, and Goldfarb's [1] model of AI as "prediction

machines". Most of us with smartphones have access to chatbots (LLM-based AI). LLMs are prediction machines that can interact with a human prompter (via their training), and the intersection between computational power and the training set allows LLMs to be trained where they are very effective (and improving) in responding to our queries.

2.2 Learning Objectives and Designing Courses

A *Learning Outcome* (LO) is an educational goal for a learner such that they can perform the outcome once they have learned it. Typically, an LO is described at a level from Bloom's Taxonomy [2] and applies the process to some content related to a field of study. Bloom's taxonomy provides a hierarchy of cognitive processes from "lower-order thinking skills", such as recall and classification, to "higher-order thinking skills" such as creating or planning. "Understanding by Design" [3] provides a design methodology for courses where the course designer starts from LOs, determines how to assess if a student has achieved the LO, and designs class activities around preparing a student for the assessment.

2.3 Curriculum Design and Analysis

Walker defines curriculum [4] as:

> A curriculum is a particular way of ordering content and purposes for teaching and learning in schools.

For our sake, we will start with this definition, but we will leave the debates of curriculum theory to Walker's treatise. Instead, our focus will be on engineering curriculum and how it is designed and analyzed – both under the assumption that the goal of a curriculum is to develop people who can professionally perform as engineers.

The easiest undergraduate engineering curriculum to explore is CpE. Since this curriculum is much narrower than EE, we will state that a CpE is a specialization branch of EE. The market demand for CpEs was significant enough that we drew a line and defined the space as CpE. From a how to create a CpE curriculum, we can look at the most recent document for CpE curriculum [5] developed by several people and released in coordination with both the IEEE and ACM in 2016. EE curriculum is much more complex as the expertise spaces in EE are much broader. Still, the core ideas in both CpE and EE are shared across both spaces, and we can talk broadly about how practicing CpE and EE focuses on harnessing the electromagnetic spectrum to solve problems in designing systems related to this spectrum. This means that both EE and CpE need to have a basic understanding of our current models of electromagnetism and the physical components that interact with it, hence, why we see these curriculums focusing on advanced mathematics, Maxwell's equations, relativity, Fourier's mathematics, etc.

Now, how do we design these curriculums? In most cases, we start with objectives [6] or aims [7] determined by accrediting agencies, universities, countries, and professors and practitioners (and companies that hire these practitioners) in a respective field. Next, we determine a period of study and the progression of courses/activities the learner does in the curriculum. When designing a curriculum a case study provides insights on the process such as Patil and Ghatage's [8].

From the perspective of analyzing how a curriculum works, a broad range of researchers have explored improving and documenting the failures of our curriculum. Recently, researchers such as Molontay *et al.* [9] have analyzed the prerequisite chains (using directed graphs) within engineering curriculums. Padhye *et al.* [10] use similar data analytic techniques to determine design patterns within a curriculum to determine how complex the curriculum is. Note, there are unintended aspects of curriculums that are not necessarily written down; this is known as Hidden Curriculum, which the reader can learn more about in works such as Villanueva *et al.* [11] work in the engineering education space.

3 Our Benchmarking Methodology

This work aims to have an LLM chatbot perform each of the assessments for each course in the curriculum, assuming a prompt can be created. Our methodology is the following:

1. Using our prompt skeleton, we create prompts to generate chatbot results for each course assessment. This will include recording the associated Jamieson's LLM prompt taxonomy for each prompt (as we will describe in the next section).
2. Run the prompts created and record the results from the chatbot.
3. With the results from the prompt, the course instructor will evaluate the result for the assessment as if they were grading a student – what should emerge is a letter grade or numerical point result. This assessment can be done using a rubric or not, following the instructor's approach when evaluating students. Note that this approach is biased because the instructor knows that the LLM is performing the assessment.
4. Based on the results, we can aggregate the data to evaluate how the chatbot performs in the course. We can make claims such as "How good is the LLM at doing a course."
5. For the evaluated portion of the course that cannot be prompted into the chatbot, we will treat these assessments as a "0".
6. The instructor will also create an "Assumed LLM score" that will make assumptions of how the LLM would score in the category. These assumptions are speculative but will provide instructor insight into how an LLM will score going forward with more advanced architectures and capabilities.

Each course will be organized and evaluated independently.

3.1 Jamieson's LLM Prompt Taxonomy

In the above methodology, we record the prompt taxonomy. Jamieson's LLM Prompt Taxonomy is a three-level classification system for LLM prompts based on existing research and presented in [12]. The taxonomy consists of:

1. **LLM Shot Type** [13]
    - *zero-shot*: Prediction without specific training [14]
    - *few-shot*: Prediction with example actions [15]
    - *multi-shot*: Multiple separate actions, can combine with other types [16, 17]
2. **LLM Reasoning of Thought**
    - nothing-of-thought (NoT): Baseline without reflection

- self-improved of thought (SoT): Reflects and improves on the prompt [18, 19]
- chain of thought (CoT): Linear steps with reasoning [20]
- tree of thought (ToT): Branching paths for alternatives [21]
- graph of thought (GoT): Network of ideas with dependencies [22]
- program of thought (PoT): Separates reasoning from computation [23]

3. **LLM Agent Architecture** [24]
    - Prompt/Response: Simple input-output model
    - Multi-Prompt/Automated Response: Response generates additional prompts
    - Human-in-the-LLM-loop (HiLL): Human guides task based on LLM responses
    - Retrieval Augmented Generation (RAG) systems [25, 26] with memory and retrieval agents

The complete taxonomy for a prompt is represented as: {*architecture; reasoning; prompt-shot-level*}

This taxonomy allows prompts to be classified based on their complexity and capabilities. More advanced prompts typically involve more complex architectures, sophisticated reasoning techniques, and few-shot or multi-shot approaches. This taxonomy can be used to document and analyze the types of prompts used in course assessments in the context of curriculum benchmarking.

4 Miami's ECE and CpE Curriculums

This project aims to apply LLMs to our engineering education space from the perspective of being an assessed student/learner. The overarching questions of greatest significance regarding AIs and their application are:

1. How does the existence of LLM tools change what engineers do?
2. Due to this change, how must our curriculum and courses change?

Though these questions are important, they are hard to contemplate without understanding our current curriculum and how the LLM performs in them. This is the starting point of this work, and we will describe our EE and CpE curriculums from the perspective of departmental mandatory courses. We will leave out our prereqs from mathematics and science, and we will not explore electives, typically taken in the 3rd and 4th years of a 4-year degree.

4.1 EE and CpE Curriculum at Miami

Figures 1 and 2 show the two curriculums of interest in this LLM benchmarking work. We have added "*" to the images to show which courses have been benchmarked in this work. These identified courses will be benchmarked in the next section, and we provide space for the benchmarkers to describe insights into the benchmarking in this exercise.

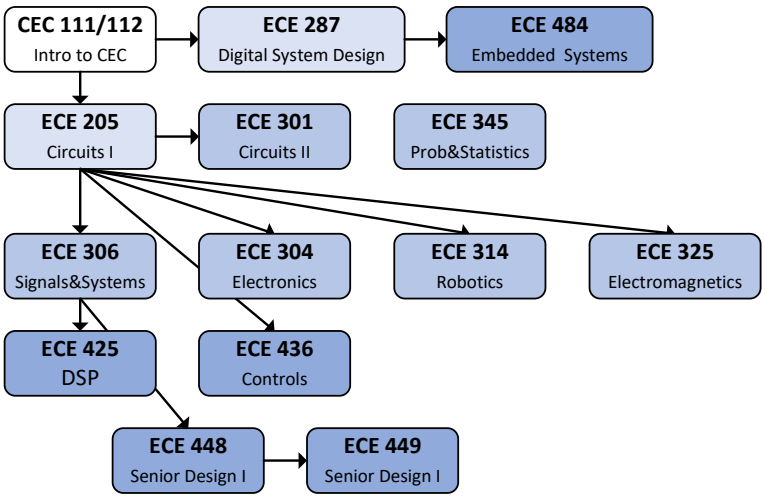

Figure 1: EE Curriculum Flowchart for courses offered in our department

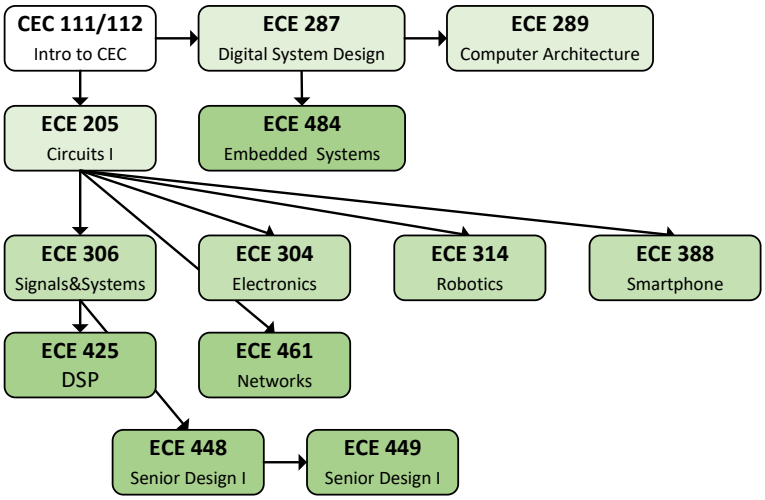

Figure 2: CpE Curriculum Flowchart for courses offered in our department

5 Benchmarking Results for our Courses

5.1 Benchmarking ECE 205 – Circuits 1

Assessment	Grade Percentage	LLM Score	Jamieson's LLM Prompt Taxonomy
Labs	30%	0%	N/A
Quizzes	15%	70%	Prompt/Response; CoT; multi-shot-4
Midterm Exam 01	15%	68%	Prompt/Response; CoT; multi-shot-4
Midterm Exam 02	15%	51%	Prompt/Response; CoT; multi-shot-4
Final Exam	25%	61%	Prompt/Response; CoT; multi-shot-4
Totals	**100**%	**43.6%**	

Table 1: LLM Performance in ECE 205

Assessment	Grade Percentage	Student Score	Jamieson's LLM Prompt Taxonomy
Labs	30%	90%	N/A
Quizzes	15%	70%	Prompt/Response; CoT; multi-shot-4
Midterm Exam 01	15%	68%	Prompt/Response; CoT; multi-shot-4
Midterm Exam 02	15%	51%	Prompt/Response; CoT; multi-shot-4
Final Exam	25%	61%	Prompt/Response; CoT; multi-shot-4
Totals	**100**%	**70.6%**	

Table 2: Student using LLM Performance in ECE 205

Our prompt template is used with Claude 3.5 to evaluate the performance of an LLM in ECE 205. The performance of the LLM on assessments in ECE 205 is shown in Table 1. In Table 1, the first column lists the assessment; the second column lists the grade percentage for each assessment in the course; the third column lists the score (in percentage) that the LLM achieves for each assessment; and the fourth column lists the prompt taxonomy used for each assessment. The final score for the LLM is provided in the final column.

Looking at the performance of the LLM in ECE 205, the LLM can take all quizzes and exams but cannot perform the lab component of the course. The LLM cannot use the physical hardware (oscilloscopes, multimeters, etc.) to complete lab assignments. Homework is assigned to the class throughout the semester but is ungraded and does not count toward the overall grade. To complete the quizzes and exams, the LLM is provided with a depiction of the circuit diagram as an attachment to the prompt, and the attachment is referenced in the prompt. Throughout the examination process, the LLM can recognize most elements of the provided circuit diagrams.

Throughout the evaluation process, there were a couple of issues that were encountered:

- When explaining its chain of thought, the LLM randomly leaves out components in its initial circuit identification or calculations. Thus, the LLM is limited by its inability to correctly identify and process circuit components, which was the leading cause of it performing relatively poorly on quizzes and exams.

- The LLM always gives itself a perfect self-assessment score. This could have disastrous consequences for students who blindly use LLMs without verifying the veracity of the presented solution(s), especially if the students trust that the self-assessment by the LLM.

After the LLM completed all quizzes and exams, the LLM was determined to have earned an **F** for the course, as shown in Table 1. However, if we were to assume that a student only used an LLM on their assessments and completed all labs, that student may achieve an average grade in the class. An average student is expected to earn 90% in the lab portion of the class. Looking at Table 2, an average lab grade of 90% would allow an average student using only LLMs to achieve a grade of 70.6%, earning them a **C-** grade. Due to this grade improvement, in ECE 205, we might consider reducing the weight of lab assignments and focusing more on other assessments.

5.2 Benchmarking ECE 287 – Digital System Design

Course Assessment	**Actual Points**	**LLMs Score**	**Jamieson's LLM Prompt Taxonomy**	**Assumed LLM Score**
Participation	4		NA	4
Exam I	6	6	Prompt/Response; CoT; zero-shot-4	6
Exam II	6	8	Prompt/Response; CoT; zero-shot-4	8
Exam III	8	6.5	Prompt/Response; CoT; zero-shot-4	6.5
Labs	38	28.5	Prompt/Response; CoT; multi-shot-4	29.5
Assignments	15	14	Prompt/Response; CoT; zero-shot-4	15
Quiz Preps	10	8.5	Prompt/Response; CoT; zero-shot-4	10
Project	15		NA	8
Totals	**102**	**71.5**		**87**

Table 3: LLM performance in ECE 287

Using our prompt template, we evaluate Digital Design (ECE 287). Table 3 shows the performance of Claude 3.5 on ECE 287. Columns 1 and 2 show the course assessment and the points they are worth. Column 3 shows Claude's evaluated response score, which shows whether the LLM can do it (green) or not (red). Column 4 shows the prompts level in Jamieson's LLM prompt taxonomy presented earlier. Finally, the last column makes assumptions of how the tool would perform given either a more complex LLM architecture or our judgment.

Regarding the "Assumed LLM Score," we assume the tool would get 4 points for participation, as these points are not difficult for students to earn. Also, we assume that the LLM tool with a human (HILL architecture) would get 8 of 15 points, as most students do not get lower than 8 points as long as they demonstrate a working project with some complexity.

For the letter grade mapping the instructor uses in his syllabus, we can say that the LLM for assessments would score a letter grade of a **C** for the artifacts it generates and an **A-** for the assumed LLM score. Exams I and II are done live in the classroom, but all the activities are done in the lab or out of class.

One important point to make here is that for any diagrams in the assignments, labs, or exams, the

prompt used a written notation to describe circuits as nodes and edges in a non-standard format. Similarly, the output of the chatbot could describe the circuit in the same format as a valid solution instead of drawing a schematic.

The main points of contention for what the chatbot responded with were:

- Karnaugh map results were regularly wrong. K-maps are used as a simple two-level logic optimization technique for optimizing small circuits (this visual tool is too hard to use for circuits with greater than five inputs). K-map-like optimization can be implemented as algorithms, and the idea of logic optimization in the 2nd year is more to have a simple technique to optimize circuits than to understand how the optimization is done. Our tool flow, for example, uses complex algorithms to optimize logic, but we have not learned how. Therefore, the failure of the chatbot to get K-maps correct hurts the chatbot for assignments by a small score, but with more advanced architectures, it would not be a problem.
- Some of the Verilog HDL code generated by the tool was either not synthesizable (as this was not part of the prompt) or used advanced Verilog syntax that we do not teach to students. In the case of non-synthesizable Verilog (which means the design, when processed by the tool flow, can be programmed to an FPGA) we modified the prompt with "synthesizable to an FPGA". We did not penalize the chatbot for using more complex Verilog.

Course	**LLM Base Letter Grade**	**Assumed Letter Grade**
ECE 287	B-	A-

Table 4: LLM grade performance in EE and CpE curriculum at Miami University

5.3 Benchmarking ECE 291 – Energy Systems Engineering

Assessment	Grade Percentage	Student Score	LLM Score	Jamieson's LLM Prompt Taxonomy
Homework	20%	90.4%	97.7%	Prompt/Response; CoT; multi-shot-4
Reflections	15%	91.8%	21.2%	Prompt/Response; CoT; multi-shot-4
Notes	15%	93.4%	90.0%	Prompt/Response; CoT; multi-shot-4
Team Activities	30%	96.0%	N/A	Prompt/Response; CoT; multi-shot-4
Active Learning Activities	20%	92.3%	N/A	Prompt/Response; CoT; multi-shot-4
Totals	**100%**	**70.6%**		

Table 5: Student with LLM Performance in ECE 291

Energy Systems and Sustainability is a course that focuses heavily on student-led discussion and curriculum, team-based active learning activities, and experiential site visits. We initially used GPT-4o for our benchmarking but were not pleased with the results. After switching to Claude 3.5, we noticed significantly better outputs. Table 5 shows the results (in similar formats to the previous) for the LLM's performance in this course.

In the *Homework* and *Notes* categories, the AI performed at the same level as or better than the students. The AI did not perform well in the *Reflections* category. Most of the *Team Activities* section involves pre/post research and notes from students performed on-site at various power

generation sites. Due to the proprietary nature of the materials involved in and generated by this analysis, we cannot feed them into an AI for analysis at this time.

The following items are of note:

- A limited number of hallucinations were observed in both the GPT-4o and Claude 3.5 AIs, and this was not expected given the qualitative nature of reflections and notes assignments, in general—for example, one reflection involved pulling information from a report published by a university sustainability council. At first, we were impressed by the summary the AI developed, as the report is public but not easily accessible. Upon further examination, it was apparent that the AI did not have specifics endemic to the report.
- In one *Notes* assignment, the AI provided a response related to the chapter in question but without the actual chapter uploaded. Once again, at first glance, it appeared this was a legitimate treatment of the material until a closer read of the results indicated otherwise.
- The *Notes* category produced some interesting questions. The AI in multiple notes assignments posed supplementary questions about the material that we had never seen developed by students
- In *Homework* responses, the AI provided quantitative and qualitative responses on par with what would be expected from second and third-year students taking a course on energy. It performed unit and order of magnitude conversions well, but had difficulty reading Sankey diagrams.
- In one *Homework* question on nuclear physics, the AI corrected an error in the text that indicated B+ instead of B- decay, and performe the correct analysis.

5.4 Benchmarking ECE 306 – Signals and Systems

Assessment	**Grade Percentage**	**LLM Score**	**Jamieson's LLM Prompt Taxonomy**
Homework	25%	100%	Prompt/Response; CoT; zero-shot-4
Audio Filtering	10%	100%	Prompt/Response; CoT; zero-shot-4
Exam 1	15%	67%	Prompt/Response; CoT; zero-shot-4
Exam 2	15%	69%	Prompt/Response; CoT; zero-shot-4
Exam 3	15%	66%	Prompt/Response; CoT; zero-shot-4
Final Exam	20%	87%	Prompt/Response; CoT; zero-shot-4
Total	**100%**	**82.7%**	

Table 6: LLM performance in ECE 306

We evaluate the performance of the LLM on ECE 306 in Table 6. The LLM used was GPT-4o. The first column shows the type of assessment, the second column shows the percentage for the assessment in calculating the final grade, the third column shows the LLM's score, and the fourth column shows the level in Jamieson's LLM Prompt Taxonomy.

The homework score is based on completion, while all other assessments are based on the correctness of the results (with partial credit assigned based on reasoning). The audio filtering

assessment consists of five assignments. For each assignment, students are provided a MATLAB Live Script that consists of text and code that explains a concept (such as the frequency response) and how to implement the idea in code on a given audio signal (such as plotting the frequencies in the audio signal). The assessment is then based on a short-answer quiz consisting of up to 10 questions.

Based on the standard letter grade mapping used for the course, the LLM would receive a **B-** for the course.

The main difficulties with the LLM were as follows:

- Several problems are required using the z-transform, which is an alternative representation of a discrete-time signal. A common discrete-time signal is the exponential a^k where $k \geq 0$ is the discrete time index. The z-transform of this signal is $z/(z-a)$. The LLM, however, consistently used the z-transform of this signal as $1/(z-a)$, which is incorrect. Interestingly, when asked specifically about this z-transform pair, the LLM correctly identified the z-transform of a^k, the inverse z-transform of $z/(z-a)$, and the inverse z-transform of $1/(z-a)$. We conclude that the LLM knew how to derive this z-transform pair but commonly "remembered" an incorrect pair when used in longer calculations.
- A problem on Exam 3 required solving a difference equation in the time domain by solving the homogeneous equation, finding a particular solution, and then summing the two and applying the initial conditions. While the LLM followed the appropriate process, it incorrectly guessed the form of the particular solution, leading to a contradiction. Instead of realizing that the contradiction meant that something was incorrect, the LLM insisted on using the incorrect particular solution. After arriving at the same contradiction twice, a mathematical error was then introduced to avoid the contradiction and to continue with the (then incorrect) solution. We suspect that the LLM had been insufficiently trained on this particular edge case and could not adapt its solution.

6 Discussion

Having completed the benchmarking of four EE and CpE courses we provide some insight into our experience. First, we will discuss the tool's performance concerning the course. Second, we will provide insights into the exercise's value for the educator to do this benchmarking and ideas of how a student might use the tool to help them. Finally, we will briefly discuss how we believe these tools will impact engineering and engineering education.

6.1 LLM's in our Courses

In our brief benchmarking exercise, we evaluated three fundamental courses in the EE and CpE curriculums in the 2nd and 3rd years. These three courses follow typical engineering courses in that they assess students based on quantitative problem-solving techniques, and correct or incorrect solutions can easily define the results of assessments. One difference between Digital Design and the other two (Circuits and Signals) is that Digital Design more quickly shifts to aspects of Engineering design, whereas the other two are more analysis. In all cases, what's

interesting is that these courses provide learners with basic skills in analysis and problem-solving. The complexity of our engineering space means that even with the challenges of these courses, a student needs to quickly learn basics that mix many of the mathematical mechanics they have learned elsewhere and apply these mechanics to analyzing and designing the physical systems we have learned to model with the mathematics. Still, from what an engineer does, these processes are far from what an engineer does daily (with exceptions). Still, from the learner's perspective, these courses are a moment of tying the physics of the world with many other ideas and skills – a momentous moment.

In these quantitative fields, however, the LLMs have problems because they lack calculation accuracy (mechanics of computation such as model, solving, and verifying). What's interesting is that most learners in this space suffer from this as well. Their questions lie in how I model my system (the question) with valid equations, how I analyze the equations once structured, and whether the final results make sense for the system. The approach to improve in these spaces is, typically, to reverse engineer problems that are solved to see the steps and be able to pattern match and replicate the forward analysis.

We will argue that the accuracy of calculations will only improve as these LLM tools are given capabilities to use other tools such as calculators, Mathematica, and other calculation tools (which in some professional versions they already do) that in the more advanced models (RAG-like) the LLM will eventually solve this. A second weakness of the LLM is the challenge of seeing our visual representations, such as schematics, graphs, visualizations, etc., and interpreting them and mapping them to our mathematical models. We, similarly, will argue this will improve LLMs as we, arguably, have trained these tools with the corpus of human text and now need to shift to visual observation techniques to train further and improve these tools.

6.2 Benchmarking Exercise

From the exercise of benchmarking our courses, two key insights are drawn as the educator. Basically, by doing this exercise, a teacher gets a stronger understanding of what the assessment is, and by going backward (like the approach taken in Understanding by Design [3]), we learn deeper insight into what we're teaching and how the learner might perform the assessment. In other words, even though the LLM is not human, it is an "intelligent" agent that provides the educator with a model of learning such that we can see how the prompt (in some way a metaphor to teaching) and response can be tuned and what the agent sees the assessment as. As educators, we have never had this capability at our fingertips, and for this experience alone, we suggest educators perform this exercise.

The second insight into the exercise is seeing how the tool responds to our prompts and then imagining what it is like being a learner who is in this space but needs help – whether that be on how to do a problem and understanding the steps in it. These LLM tools are powerful and are used extensively by undergraduates. Now, however, imagine being the learner who can't assess the response of the prompt when used as a tutor. Can you trust the response? This is not based on our experiences; we know this because we are experts and can evaluate responses. What do you do as a beginner? In this way, we believe that what learning engineers need to be taught is the one-shot answer approach to prompting an LLM needs to be done at a deeper level, which is to

say, the learner and the LLM become integrated tighter such that instead of using it as a prompt and done method, the interaction between the two needs to be a back and forth such that the learner gets started with the LLM on how to approach the problem but then goes back and forth with it to breakdown the problem and solve it. Teaching students to do this is a challenge, but we think it is a critical skill for students moving forward.

6.3 LLMs and Engineering Education

With the above insights and what we are hearing from the world, we propose that moving forward, we need to teach the use of these tools to produce competitive engineers. So what does this mean? First, we go back to the early definition of AI in this paper: these are "prediction" machines (at "low" cost). The metaphor we have been using with LLMs is that they now provide us with a "calculator" tool for producing English (in our case) communications. Still, they have emerging skills given that they are trained in the corpus of human text – which is incredible. This emerging tool is still hype, but we are already seeing industries respond by asking how these tools improve efficiency. This translates to the economics of hiring engineers in that, if companies can double the productivity of their current engineers, we will see a decline in demand. In coding (which is deeply tied to language), these LLMs will see early success, and not surprisingly, software engineering will be hit first with this demand decrease. We might even call the prompting of an LLM the High-High-Level programming language.

Therefore, from the perspective of engineering education, we need to produce the next generation of engineers who know how to use these tools powerfully. This still starts with what we call "first-principle" learning, which is what our early curricula courses that capture how mathematics, physics, modeling, analysis, and design link together should still be assessed via traditional approaches. Why? When done by humans, these exercises are the only method of producing engineers who can informally reason in this space. By informal reasoning, we mean that within a human brain, the human can envision/picture these equations and physical systems that are more ideas than part of a typical human experience. Developing this informal reasoning with the LLM tools will be hard since we barely understand the first part of this sentence. However, if we can do this, our future engineers will, hopefully, take us to new and unimagined spaces, tackling the grand problems that our societies are facing.

7 Conclusions

In this work, we performed a benchmarking exercise in our respective courses in EE and CpE curriculums. We described our methodology, provided a taxonomy to classify prompts, and then used this approach to benchmark our respective courses. As each instructor performed their benchmarking, they provided insights into this process. Based on these results, we hypothesize that the benchmarking exercise helps educators and learners in these spaces. Our thoughts on this was provided in the discussion section.

Our major conclusion is that as an education space for future engineers, we must embrace and play with these tools to understand how our education research and goals must adapt. Our approach provides educators with an exercise we believe each should take with their

courses.

8 Acknowledgements

Miami University's College of Engineering and Computing supported this project.

We acknowledge the use of Claude 3.5 (https://claude.ai/, Accessed August-October 2024) to improve this document's organization and academic writing. We prompted the tool with various ideas and used generated results as starting points for aspects of our writing, noting that the ideas in the prompts were our own and that the authors edited and checked responses. We also acknowledge the use of Grammarly as a tool to improve our writing. No part of this document was written by an AI tool alone.

References

[1] A. Agrawal, J. Gans, and A. Goldfarb, *Prediction Machines, Updated and Expanded: The Simple Economics of Artificial Intelligence*. Harvard Business Press, 2022.

[2] L. W. Anderson, D. R. Krathwohl, and B. S. Bloom, *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Allyn & Bacon, 2001.

[3] G. P. Wiggins, J. McTighe, L. J. Kiernan, and F. Frost, *Understanding by design*. Association for Supervision and Curriculum Development Alexandria, VA, 1998.

[4] D. F. Walker, *Fundamentals of curriculum: Passion and professionalism*. Routledge, 2002.

[5] J. Impagliazzo, S. Conry, J. Hughes, L. Weidong, L. Junlin, A. McGettrick, V. Nelson, E. Durant, H. Lam, and R. Reese, "Curriculum guidelines for undergraduate degree programs in computer engineering," *CE2016). December*, vol. 15, 2016. [Online]. Available: https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf

[6] T. Dion and K. C. Bower, "Integrating learning outcomes throughout the civil engineering curriculum to meet site engineering prerequisite needs," in *ASEE Southeast Section Annual Conference, Louisville, KY*, 2007.

[7] J. Prasad, A. Goswami, B. Kumbhani, C. Mishra, H. Tyagi, J. H. Jun, K. K. Choudhary, M. Kumar, N. James, V. R. S. Reddy *et al.*, "Engineering curriculum development based on education theories," *Current Science*, pp. 1829–1834, 2018.

[8] S. R. Patil and P. S. Ghatage, "Curriculum development of an engineering pg program at an autonomous institute–a case study," *Journal of Engineering Education Transformations*, vol. 32, no. 4, 2019.

[9] R. Molontay, N. Horváth, J. Bergmann, D. Szekrényes, and M. Szabó, "Characterizing curriculum prerequisite networks by a student flow approach," *IEEE Transactions on Learning Technologies*, vol. 13, no. 3, pp. 491–501, 2020.

[10] S. M. Padhye, D. Reeping, and N. Rashedi, "Analyzing trends in curricular complexity and extracting common curricular design patterns," in *2024 ASEE Annual Conference & Exposition*, 2024.

[11] I. Villanueva, "What does hidden curriculum in engineering look like and how can it be explored?" in *Proceedings of the American society of engineering education annual conference and exposition, minorities in engineering division*, 2018.

[12] removed for blind review.

[13] S. Kanti Karmaker Santu and D. Feng, "Teler: A general taxonomy of llm prompts for benchmarking complex tasks," *arXiv e-prints*, pp. arXiv–2305, 2023.

[14] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," *Advances in neural information processing systems*, vol. 22, 2009.

[15] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[16] 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.

[17] M. Mizrahi, G. Kaplan, D. Malkin, R. Dror, D. Shahaf, and G. Stanovsky, "State of what art? a call for multi-prompt llm evaluation," *CoRR*, 2024.

[18] J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han, "Large language models can self-improve," *arXiv preprint arXiv:2210.11610*, 2022.

[19] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[20] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[21] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[22] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk *et al.*, "Graph of thoughts: Solving elaborate problems with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 682–17 690.

[23] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," *arXiv preprint arXiv:2211.12588*, 2022.

[24] 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, p. 186345, 2024.

[25] A. Krithara, A. Nentidis, K. Bougiatiotis, and G. Paliouras, "Bioasq-qa: A manually curated corpus for biomedical question answering," *Scientific Data*, vol. 10, no. 1, p. 170, 2023.

[26] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," *arXiv preprint arXiv:2007.01282*, 2020.