

LLM Prompting Methodology and Taxonomy to Benchmark our Engineering Curriculum

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.

Suman Bhunia, Miami University

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.

LLM Prompting Methodology and Taxonomy to Benchmark our Engineering Curriculums

Abstract

As Large Language Models (LLMs) continue to transform our educational landscape, we educators find ourselves challenged by how we should use these tools and how we should help future learners use them. Engineering education has the fortune of already including high levels of Bloom's taxonomy learning outcome (i.e., create), which LLMs cannot consistently perform. Still, we need to adapt to these tools' emergence and capabilities. How will we adjust our curriculums to this rapidly evolving technology? This paper proposes benchmarking our courses against LLM capabilities to benchmark our curriculums. By introducing Jamieson's LLM Prompt Taxonomy, a three-level classification system for LLM prompts, we create a framework to evaluate the performance of these AI tools within our existing educational structures.

Our methodology involves crafting prompts for course assessments, categorizing them using the proposed taxonomy, and comparing LLM performance to traditional student outcomes. We provide an example use of this with an undergraduate course on Digital System Design. We demonstrate how this benchmarking process offers insights into the strengths and limitations of LLMs in an educational context.

This work serves as a call to action for educators across disciplines. By systematically benchmarking our curriculums against LLM capabilities, we can better understand the evolving relationship between AI and education. This understanding will allow us to refine our teaching methods, emphasize uniquely human skills, and prepare our students for a future where collaboration with AI is expected. As we move forward, it is crucial that we, as educators, take charge of shaping how these powerful tools are integrated into our classrooms and beyond. This work will illuminate the need for curriculum-based learning outcomes at high levels in Bloom's taxonomy.

1 Introduction

Our engineering and higher education curriculum's complexity is greater than this project contends to explore. However, there are many trends in modern society that we believe must be addressed in our curriculums in the short term for our students' benefit:

1. A broad education that includes understanding how computing technologies as tools and essential infrastructure of our world impacts all of us.
2. Modern industry seeks a diverse workforce as this diversity impacts all facets of design. We need to help teach this diverse workforce. This includes pushing the narrative so that all walks of human life can see themselves as engineer, which seems to be directly linked to early efficacy within the fields [1] [2].
3. Providing an undergraduate engineering education with an understanding that future professionals want to impact the world in many ways, including what we have motivated in the past, such as personal success and creating “cool” technology (traditionally motivated by robots, games, cars, etc.), but there are many who are looking for other social impacts in their careers including human welfare and environmental sustainability [3].
4. AI in modern society has seen the cost of using prediction technology go down to a point where the technology is democratized to a level where any user in the first world can access these tools [4].
5. The democratization of engineering and computing fabrication allows most people in the first world to access (at reasonable cost) fabrication technology and computation at the scale of an individual — this trend, pushed by artists [5] and engineers, makes the public and our students not only aware but experienced in building things using these tools. However, the assumption that a student, because they lived before or during a technology emergence, is strongly skilled with that technology is false [6].

The biggest of these trends that we address in this work is trend 4, commonly referred to in the general population as chatbots such as ChatGPT — the continued emergence of AI capabilities — specifically, the emergence of Large Language Models (LLMs) [7] means our curriculums need to be viewed as how AI capabilities change the educational landscape.

This work aims to provide a methodology for benchmarking our engineering curriculums with these LLM-based chatbots. To do this, we define an LLM prompt taxonomy to quantify the LLM and its related prompts. With this taxonomy and a model of how courses are created with learning outcomes and assessments of learning outcomes, we provide a methodology to benchmark our existing courses. This is a first step in evaluating how these emerging technologies will impact our curriculums.

2 Background - Models of Education and Artificial Intelligence

Before discussing our methodology to benchmark our curriculums, we start by providing some assumptions of what Artificial Intelligence (AI) and LLMs are in a model that might be better for us all to understand. Generally, the definitions of intelligence, education, learning, and what our brains do are extremely complex, and the wide variety of scientific fields (Cognitive Psychology, Neuropsychology, Educational Psychology, Artificial Intelligence, etc.) that work in this space provides a broad glimpse of the complexity of the questions and includes many definitions. Therefore, we will provide starting points based on models and definitions to create a curriculum/course benchmark.

2.1 Educable - a definition of intelligence?

First, we use Valiant's "Educable" definition [8]:

- (a) "learning from experience."
- (b) "acquiring theories through instruction."
- (c) "applying what one has acquired through (a) and (b) in an integrated way."

This definition helps us understand what learning is in terms of (a) and (b), and from an engineering perspective, the application and skills of engineering and design relate to (c). At present, we can argue that humans and our ability to perform (a), (b), and (c) provide our superabilities in information processing and transformation compared to animals and existing intelligent agents.

2.1.1 AI and LLMs - what are they?

Now, in terms of AI, we will use Agrawal, Gans, and Goldfarb's [4] model of AI as "prediction machines". Our current AI excitement of this technology (that we might argue is in the "early majority phase" if not further in the adoption curve [9]) and its benefits and drawbacks are being experienced and discussed in almost every space. Higher Education is no exception. Most citizens are seeing a democratization (from a first-world smartphone owner perspective) of the technology in the form of chatbot LLMs such that most people in North America can interact with these tools at the touch of their fingers. LLMs are prediction machines that can interact via language prompts — the intersection and the vast data that LLMs are trained on means that the prediction engine can respond to our queries. Going deeper, Bowman's eight things to know about LLMs [10] are:

1. LLMs predictably get more capable
2. Many important LLM **behaviors emerge unpredictably** — Arora and Goyal [11] would describe these as emerging skills
3. LLMs often appear to learn and use representations of the outside world
4. There are no reliable techniques for steering their behavior
5. Experts are not able to interpret the inner workings
6. Human performance on a task is not an upper bound on LLM performance
7. LLMs need not express the values of their creators
8. Brief interactions with LLMs are often misleading

This list provides insights into this technology, where we can no longer explain the technology for particular instances/queries (AI explainability [12]). In general, though, LLMs as turn-key solutions is a technology that can be prompted into a task and responds with a solution to that task, typically as an artifact of text.

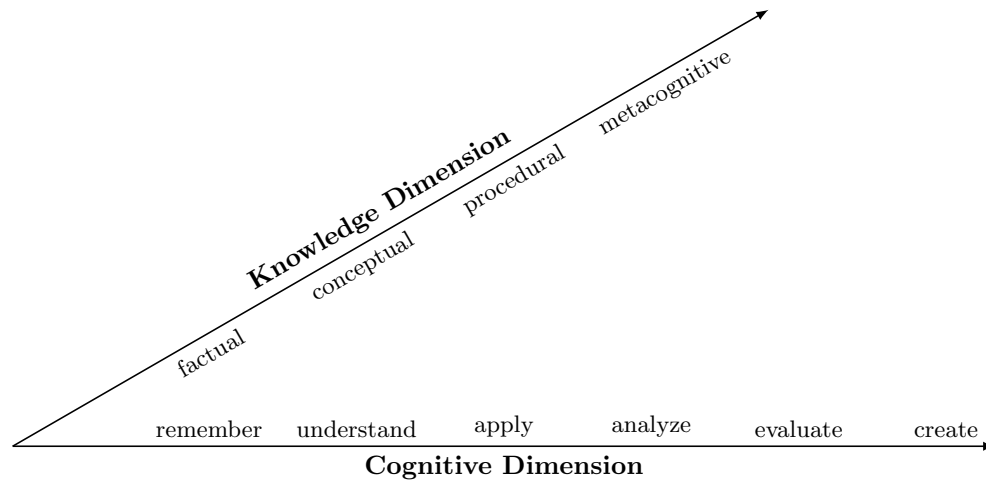


Figure 1: The two dimensions of Bloom's taxonomy [19]

2.1.2 Bloom's Taxonomy and Learning Outcomes

In previous work by Jamieson, Bhunia, and Rao [13], we defined Learning Outcomes (LOs) and Bloom's Taxonomy in the following ways. We use the lenses of Bloom's Taxonomy [14] and Understanding by Design's [15] filter of concepts to understand the Learning Outcomes (LOs) in a particular curriculum and its respective courses. In our previous work, we used a similar approach to understand the use of games in a course [16] to help educators evaluate if they should spend time creating a game to teach a concept. Here, the choice of a LO provides an understanding of assessments in a course, which is the intersection of LLMs and our courses. We will then use the assessment points to develop our benchmarks.

We further define learning in this work with Valiant's educable definition, using ideas from [17], [18]: "Learning is the stabilization of neural networks based on a desired response". This definition allows us to relate the idea of learning things to the necessity of trying and doing things since if the neurons are not firing, your neurons are not wiring (Donald Hebb stated in 1949, "neurons that fire together wire together"). This learning model fits into our earlier definition of "educable" but has a higher connection to the physical aspect of learning.

A *Learning Outcome* (LO) is an educational goal for a learner such that they are able to perform the outcome once they have learned it. Typically, an LO is described by a cognitive process taken from Bloom's Taxonomy [14] and applying the method 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. The "higher-order" thinking skills tend to be what society wants of our workers. Still, because of the professional nature of learning, these higher-order skills are much more challenging to develop and take significantly more time than the lower-order thinking skills to teach and learn. Figure 1 shows two dimensions of Bloom's taxonomy: cognitive and knowledge dimensions [19].

However, this distinction between "higher" and "lower" can confuse educators in somehow evaluating the quality of an LO based on where it sits in the taxonomy. As illuminated by Lang

[20], professors can fall into the trap of simply wanting to teach “higher-order thinking skills” and assume that the lower skills can be looked up using technology such as search engines and the like (including LLM tools like ChatGPT). In reality, achieving skills such as “create” and “plan” in Bloom’s taxonomy requires a human to keep and think about ideas and concepts within a field of study that requires the “remember” and “understanding”. Learning and achieving LOs at a higher Bloom’s level, which we typically believe happens in the later portions of our curriculum and courses, is a holistic process that requires achieving LOs at all Bloom’s taxonomy levels. We call this misunderstanding by professors learning about teaching **Bloom’s Folly**.

An LO is defined in a sentence where a verb is taken from Bloom’s taxonomy [14], and the object of the sentence describes some knowledge or skill. For example, a sample LO might be, “A student should be able to *construct* a program that will then generate a desired output with a proper sequence of program statements.” This outcome is *apply* in Bloom’s taxonomy cognitive process dimension [14] and is *procedural* in the knowledge dimension [19]. This LO would be considered a “middle-order thinking skill” from the taxonomy.

To further determine how significant an LO is as relevant to a curriculum or course, we use Wiggins and McTighe’s filter [15]: (1) worth being familiar with; (2) important to know and do; (3) enduring understanding; where the importance of an LO is more important to a student’s future as the above number increases. This filter allows teachers to consider their LOs and course goals and qualify their importance. This is useful because just because an outcome ranks as a “low-order thinking skill,” if it is an enduring concept for that student in their respective field, then it is important to emphasize and should be achieved by the student. Similarly, depending on the time in the curriculum as related to a field, different ideas will need to be learned to progress in a field holistically.

3 Proposed Benchmarking of our Curriculums

This project aims to provide a methodology on how to apply LLMs to our engineering education space from the perspective of being a student/learner. We argue that the two questions of greatest significance in this regard are:

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

These two questions are “point solutions” as described in the book “Power and Prediction” by Agrawal *et. al.* [21].

1. **Point solutions:** Enhance isolated components for local efficiencies. Example: Replacing steam engines with electric generators in factories.
2. **Application solutions:** Redesign interdependent parts for significant efficiencies. Example: Installing individual electric motors on machines for on-demand power usage.
3. **System solutions:** Overhaul entire processes for maximum efficiency. Example: Electricity enables flexible factory layouts, leading to assembly line production.

The question of how LLMs and AI might disrupt higher education lies in a “System solution”, which is beyond the scope of this work.

Though of great significance, these questions are hard to contemplate without an understanding of our current curriculum and how an LLM performs in it. We use benchmarks to evaluate a system quantitatively. Even though the system, in this case, the assessment of a learner or an AI, is complex, this project focuses on creating these initial benchmarks.

3.1 LLM Benchmarks for a Course and Our Curriculum

The main goal of this work is to take courses in our engineering curriculums and create prompts as assessment benchmarks for LLMs to understand what a course is assessing and gauge how an LLM might perform in that course. Looking at engineering undergraduate education and Bloom’s taxonomy, one of the key paradigms of the space is that our assessments tend to separate into exam/assignment/quiz assessments. These map to Bloom’s levels Remember, Understand, Apply, and Analyze versus more open design project questions typically aligned with Analyze, Create, and Evaluate. LLMs are not yet performing at these later Bloom taxonomy levels in terms of being able to complete a task in a single prompt. However, when a human can iterate and interact with the LLM with multiple prompts, these higher-level aspects can be solved well.

Next, in this work, we will propose an LLM prompt taxonomy that we believe will correlate with Bloom’s Taxonomy.

3.2 Jamieson’s LLM Prompt Taxonomy

We describe our taxonomy for a prompt to an LLM based on existing research in this space. Below, we propose a composition of a three-level taxonomy for prompts and will use this taxonomy to label our benchmarking prompts. This taxonomy for LLM prompting is based on three levels: architecture (in sub-section 3.2.3), type of LLM thought (in sub-section 3.2.2), and shot type (in sub-section 3.2.1). We will discuss these reverses as they go from simplest to more complex.

3.2.1 LLM Shot Type

Table 1 shows the prompt taxonomy provided by Kanti *et. al.* [22]. The level they specify is shown in column 1, and column 3 describes the details of the prompt. These levels are for a single prompt generation. In column 2, we rely on these levels with the nomenclature *shot_type-level_number* such that zero-1 represents the level 1 LLM prompt as a zero-shot action/prompt. The prompt shot definitions of *shot_type* are:

- **zero-shot** - is a prediction action/prompt where the action has not been trained for specifically by the agent [23].
- **few-shot** - is a prediction action/prompt where the action includes examples of the form of the action [24].
- **multi-shot** - is a prediction action/prompt(s) where the action is broken into several separate actions. These multiple actions can now be distinguished into additional multi-shot

Level	prompt-shot-level	Directive-Type
0	zero-shot-0	No directive, just data.
1	zero-shot-1	Simple one-sentence directive expressing the high-level goal.
2	zero-shot-2	Multi-sentence (paragraph-style) directive expressing the high-level goal and the sub-tasks that need to be performed to achieve the goal.
3	zero-shot-3	Complex (bulleted-list-style) directive expressing the high-level goal along with a detailed bulleted list of sub-tasks to be performed.
4	zero-shot-4	A complex directive that includes the following: 1) Description of High-level goal; 2) A detailed bulleted list of sub-tasks; 3) A guideline on how LLM output will be evaluated/Few-Shot Examples
5	multi-shot-5	A complex directive that includes items from the previous level and 4) Additional relevant information gathered via retrieval-based techniques.
6	multi-shot-5 blend with chain-of-thought	A complex directive that includes items from the previous level and 5) An explicit statement asking LLM to explain its output.

Table 1: LLM prompt level taxonomy from [22] relabelled with prompt-shot-level.

categories such as cycles of actions and linear steps, and these fall into our next distinguishing categories such that you can have multi-shot with few-shot [7] [25].

In our prompt taxonomy, we will not consider level 6 as Kanti *et. al.* does in their work, and instead, we believe that the reasoning of thought for the prompt is its classification of a prompt.

3.2.2 LLM Reasoning of Thought

Of Thought	Prompt Thought Type
nothing-of-thought (NoT)	Baseline type of thought where the prompt is not asked to be reflective or informed on its prompting.
self-improved of thought (SoT)	The action/prompt includes a request to reflect and improve on the prompt [26] [27].
chain of thought (CoT)	The action/prompt includes a request to analyze the prompt in terms of the linear chain of steps that will be output, including the reasoning for each step [28].
tree of thought (ToT)	Extends CoT by allowing branching paths for exploring alternative solutions [29].
graph of thought (GoT)	A flexible approach that models information as a network of ideas with connections representing dependencies [30].
program of thought (PoT)	separate reasoning from computation, where the LLM defines problem-solving steps and an external program does calculations [31].

Table 2: LLM prompts type of thought

Here, we propose a deeper taxonomy for LLM prompts beyond Kanti *et. al.*, which we will call Jamieson’s LLM prompt Taxonomy. However, we take Kanti’s taxonomy levels and label those

as prompt-shot-level for the taxonomy described in the previous section. The next taxonomy level is the type-of-thought, broadly viewed in Table 2. We include basic thought models that are currently employed in prompt engineering. Different thought types have been shown to improve the prompt output quality concerning the requested action/prompt.

3.2.3 LLM Agent Architecture

ChatBot as Agent Architecture	Description
Prompt/Response	The agent is provided a prompt and gives a response.
Multi-Prompt/Automated Response	In this model (such as PoT), a prompt response is provided where the response generates additional prompts.
Human-in-the-LLM-loop (HiLL)	Prompt are provided, and the human uses the response to guide the task forward.

Table 3: LLM prompts as agents in an architecture

Finally, as has been eluded to in some of the definitions from above, we define different approaches to using the prompting chatbot in a system — or the architecture. Table 3 shows the various models that have been currently used [32]. In these architectures, the LLM chatbot is considered an agent and is part of a more extensive system architecture. The architecture of that system is very briefly classified above. However, the architecture can include much more complexity regarding capabilities such as memory, function calls to other tools/agents, and planning. For example, an architecture we might consider is the recent Retrieval Augmented Generation (RAG) based systems [33] [34] that blend LLMs with the retrieval of and memory of queries.

The taxonomy of a prompt consists of three levels: {architecture; reasoning; prompt-shot-level}. We will argue that these three levels, from left to right, define the capability of the prompting approach (more complex prompts have a more complicated architecture, use advanced reasoning techniques, and are few-shot templated prompts). From a benchmarking of LLMs in our curriculum perspective, we propose to have instructors/project participants work at a prompt for their assessments and document “Jamieson’s LLM prompt taxonomy” instead of enforcing a specific approach. In direct relation to this, we believe that the {create, analyze, and evaluate} levels of Bloom’s taxonomy will correlate more closely to the more complex prompts with complex agent/system architectures that have not yet been created.

4 Sample Benchmarking a Course

This work proposes that we educators need to observe how well an LLM performs within our curriculums. The ideal result is to make benchmarks for our courses that could be run on various LLM tools (of which there are many and will be many more). The motivation for creating this methodology comes from the work by Kevian *et. al.* [35], where they benchmarked control courses at the University of Illinois and UC San Diego and provided early ideas on benchmarking a course topic. Several other efforts are emerging in this regard, including benchmarking

mathematics by Zou *et. al.* [36] and a comprehensive question of how LLM chat-bots perform in engineering by Borges *et. al.*[37].

Several efforts are underway to benchmark LLMs against each other and in specific intelligent tasks. Our goal in this work is not to test the different LLM chatbots to quantify which is best but to understand how LLMs perform in our curriculums and courses. However, some of the ideas from the benchmarking of LLMs apply to this work. Two popular benchmarks for LLMs include LMentary [38], and BIG-bench [39]. These benchmarks have been used in various ways to understand the qualities of LLMs (and are not the only ones in existence).

No formalized benchmarking paths have emerged in higher education, but a range of researchers in almost all domains have been researching and publishing small-scale explorations. For example, Balhorn *et. al.* [40] evaluated ChatGPT’s answering capabilities in the natural sciences and engineering. They used a rubric to assess the responses and compared various aspects in this space to human-level students at the undergraduate, master, and Ph.D. levels. Rein *et. al.* [41] performed a similar benchmark (in this case, multiple choice questions) to evaluate GPT-4. These two examples are from the larger attempts, but many results are at the individual course/topic level.

We provide a base benchmark using a course we teach called ECE 287 —Digital System Design at Miami University. This second-year course focuses on teaching students how to create a digital system on an FPGA to implement algorithms.

1. Using the course ECE 287 as a template, we will create prompts to generate chatbot results for each of our assessments. This will include the associated Jamieson’s LLM prompt taxonomy for each prompt.
2. Next, with the results from the prompt, the benchmarker will evaluate the result as if they were grading a student — what should emerge is a letter grade or numerical point result (maybe using a rubric or not, but following the approach the instructor follows)¹.
3. Each participant will need to look at the assessments in their course and create prompts for those they can, as provided by the template.
4. Based on the results, we can aggregate the data and make claims such as “How well the LLM performs in a course.”

4.1 Basic Prompt Benchmark Template

We suggest taking course assessments and putting them in the following prompt template for a starting prompting point. From Jamieson’s LLM prompt taxonomy, this is a {Prompt/Response; CoT; multi-shot-5} prompt template that can be made into a zero-shot-4 if there are no examples (for the multi-shot). In this particular 2nd year course, the LLM can achieve the exams and assignments with this level of prompting, but as we will show, as Bloom’s taxonomy level goes higher, the LLM cannot complete the assessment.

¹We note that this approach is biased because the instructor knows that the LLM is doing the assessment.

Algorithm 1 Template for {Prompt/Response; CoT; multi-shot-5} Prompt

- 1: **Problem/Task:** XXX-[Provide the full text of the problem or task]
 - 2: **Prompt:** Please solve the above problem using the following steps encased in the step numbers:
 - 1) Analyze the problem and identify key components.
 - 2) Provide a step-by-step solution, explaining your reasoning for each step.
 - 3) Evaluate your solution and consider alternative approaches.
 - 4) Summarize your final answer and justify your reasoning.
 - 5) Reflect on the problem-solving process and suggest potential improvements.
 - 6) Assess your problem response based on the assessment technique provided.
 - 3: **Exemplars:** Here are two examples of similar problems with their solutions. Using these examples as a guide, please proceed with solving the given problem by self-correcting and providing reasoning for your solution steps:
[Example 1 - XXX]
[Example 2 - XXX]
 - 4: **Assessment Technique:** This question is assessed via XXX-[Describe the assessment method, e.g., rubric details]
-

Note, if there are no examples, this can be removed and becomes a {Prompt/Response; CoT; single-shot-4}.

4.1.1 Example Prompt Template from ECE 287

Here is a simple example of an assignment problem with the prompt in yellow (edited for the paper) and the response in green. From the author's perspective, he would have assessed this at the same rank as Claude-3.5 LLM did.

Problem/Task

What is the transistor count and truth table for the following circuit described as a graph of nodes where nodes are gates, primary inputs, primary outputs, Gnd, and Vcc:

- 2AND (inputs: A, B, output: 2AND)
- NOT (inputs: 2AND, output: NOT)
- 2NOR (inputs: B, C, output: 2NOR)
- 2XOR (inputs: NOT, 2NOR, output: OUT)
- A (type: primary_input)
- B (type: primary_input)
- C (type: primary_input)
- Vcc (type: power_supply)
- Gnd (type: power_gnd)
- OUT (type: primary_output)

Prompt

Please solve the above problem using the following steps encased in the step numbers:

1. Analyze the problem and identify key components.
2. Provide a step-by-step solution, explaining your reasoning for each step.
3. Evaluate your solution and consider alternative approaches.
4. Summarize your final answer and justify your reasoning.
5. Reflect on the problem-solving process and suggest potential improvements.
6. Assess your problem response based on the assessment technique provided.

Assessment

This question is assessed via a rubric with the following:

- **0.4 pts - Excellent:** Question completely solved (work shown where appropriate) and no errors.
- **0.32 pts - Very Good:** Question answered with only one or two minor errors.
- **0.25 pts - Average:** Question attempted, but a significant error or miscalculation (where the learner has not reviewed) has led to an okay (but incorrect) solution.
- **0.19 pts - Below Average:** Some of the question has been solved, but missing major portions of the solution.
- **0.1 pts - Poor:** A poor attempt at the question with no progress toward a solution.
- **0 pts - No Marks:** Either no attempt, or the attempt seems to have no direction to solving the problem.

Response - Claude 3.5

Let's solve this problem step by step:

1. Analyze the problem and identify key components:

- We need to determine the transistor count and truth table for the given circuit.
- The circuit consists of a 2AND gate, a NOT gate, a 2NOR gate, and a 2XOR gate.
- There are three primary inputs: A, B, and C.
- The output is labeled OUT.
- We need to consider the connections between these gates.

2. Step-by-step solution:

1. Transistor count:

- 2AND gate: typically 6 transistors (4 for AND, 2 for output inverter)
- NOT gate: 2 transistors
- 2NOR gate: 4 transistors
- 2XOR gate: typically 12 transistors
- Total transistor count: $6 + 2 + 4 + 12 = 24$ transistors

2. Truth table construction:

- We have 3 inputs (A, B, C), so we'll have $2^3 = 8$ rows.
- Let's evaluate each gate's output for all input combinations:

A	B	C	2AND	NOT	2NOR	OUT (2XOR)
0	0	0	0	1	1	0
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	0	1	0	1
1	0	0	0	1	1	0
1	0	1	0	1	0	1
1	1	0	1	0	0	0
1	1	1	1	0	0	0

3. Evaluation and alternative approaches:

- This solution provides both the transistor count and truth table as requested.
- An alternative approach could involve using Boolean algebra to simplify the circuit before creating the truth table, but this wasn't necessary for the given task.

4. Final answer summary:

- Transistor count: 24
- Truth table: As shown above, with 8 rows covering all possible input combinations and their corresponding outputs.
- The transistor count is based on typical implementations of these gates in CMOS technology. The truth table is derived by following the signal through each gate in the circuit for all possible input combinations.

5. Reflection and potential improvements:

- The solution directly addresses both parts of the problem.
- To improve, we could provide a more detailed explanation of how each gate's output is determined, especially for the 2XOR gate.
- We could also discuss how the transistor count might vary with different implementations of these gates.

6. Self-assessment based on the provided rubric:

- I believe this solution merits the "Excellent" rating (0.4 pts) because:
 - The question is completely solved with work shown.
 - Both the transistor count and truth table are provided.
 - The reasoning for the transistor count is explained.
 - The truth table is constructed step-by-step.
 - There are no apparent errors in the solution.

Using prompts like these, we evaluated our course ECE 287. Table 4 shows the performance of Claude 3.5 in the course. Columns 1 and 2 show the course assessment and the points they are

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

Table 4: LLM performance in ECE 287

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.

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. Of note, exams I and II are done live in the classroom, but all the activities are done in the lab or out of class.

5 Discussion

So far, we have specified a method for benchmarking engineering courses for LLM chatbots. In general, we believe that engineering educators should, at a minimum, be playing with this technology. Starting with the simple question of "what can these tools help me achieve in my normal work day?" is a valid space to start. It is pretty clear that our students are using these technologies, and the reality is that there will be changes for practicing engineers, which means that we need to ask "how should we change what we teach?"

The above approach will give each teacher insight into how well the technology works in their subject matter. Additionally, from the lens of Wiggins and McTighe's filter [15]: (1) worth being familiar with; (2) important to know and do; (3) enduring understanding; we need to honestly address our students when an activity needs to be done for the sake of developing intellectual skills in design and theory. In education, there have always been instances of us knowing how to solve something. Still, doing it provides and reinforces all forms of valuable development. As educators, we need to effectively evaluate these moments and justify why the exercise is of value to the student.

Finally, a side product of benchmarking our courses allows us to inspect our courses from the inside out and potentially improve our teaching. When the LLM is treated like a virtual student and is prompted to provide its reasoning, the teacher can derive more profound insight into how an intelligence understands something. These mechanics provide insight into how the solution process might be taught in a logical format.

6 Conclusions

In this work, we looked at how to create benchmark prompts to be assessed in the courses in our engineering curriculums. We defined what AI is and what education is and described our prompt taxonomy for LLMs, which is the technology behind the commercial democratization of intelligent chatbots. With these definitions, we provide a methodology to benchmark the individual courses that, when done at scale, would allow us to benchmark an undergraduate engineering curriculum. We give an example of creating a benchmark of prompts for a course and provide the results of benchmarking a single course — Digital System Design. We believe that in higher education, we should benchmark all of our curriculums at an undergraduate level to help us understand what our courses are providing to the learner and to be able to communicate to our learners why they should complete course exercises while openly being able to describe how well existing LLMs perform in a course. This will provide us with an understanding of where our curriculums are and how to improve them.

7 Acknowledgements

Miami University’s College of Engineering and Computing provided support for this project.

We acknowledge the use of Claude 3.5 (<https://claude.ai/>, Accessed August-December 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. Venugopal, R. Singh, and Devika, “Role of self-efficacy in the learning output of engineering education,” in *Enhancing future skills and entrepreneurship: 3rd Indo-German Conference on Sustainability in Engineering*. Springer, 2020, pp. 233–242.
- [2] P. Jamieson, “Why we should be using our stories and the hero’s journey to help recruit a more diverse population for engineering,” in *2023 ASEE Annual Conference & Exposition*, 2023.
- [3] A. Diekmann, E. Clark, M. Johnston, R. Brown, and M. Steinberg, “Malleability in communal goals and beliefs influences attraction to stem careers: Evidence for a goal congruity perspective,” *Journal of Personality and Social Psychology*, vol. 101, no. 5, pp. 902–918, 2011.
- [4] A. Agrawal, J. Gans, and A. Goldfarb, *Prediction Machines, Updated and Expanded: The Simple Economics of Artificial Intelligence*. Harvard Business Press, 2022.
- [5] P. Jamieson, “Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?” in *Proc. FECS*, 2010, pp. 289–294.
- [6] G. E. Kennedy, T. S. Judd, A. Churchward, K. Gray, and K.-L. Krause, “First year students’ experiences with technology: Are they really digital natives?” *Australasian journal of educational technology*, vol. 24, no. 1, 2008.

- [7] 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.
- [8] L. Valiant, *The Importance of Being Educable: A New Theory of Human Uniqueness*. Princeton University Press, 2024.
- [9] E. M. Rogers and R. Adhikarya, “Diffusion of innovations: An up-to-date review and commentary,” *Annals of the International Communication Association*, vol. 3, no. 1, pp. 67–81, 1979.
- [10] S. R. Bowman, “Eight things to know about large language models,” 2023. [Online]. Available: <https://arxiv.org/abs/2304.00612>
- [11] S. Arora and A. Goyal, “A theory for emergence of complex skills in language models,” *arXiv preprint arXiv:2307.15936*, 2023.
- [12] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial intelligence*, vol. 267, pp. 1–38, 2019.
- [13] P. Jamieson, S. Bhunia, and D. M. Rao, “With chatgpt, do we have to rewrite our learning objectives-case study in cybersecurity,” in *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2023, pp. 1–5.
- [14] 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.
- [15] G. P. Wiggins, J. McTighe, L. J. Kiernan, and F. Frost, *Understanding by design*. Association for Supervision and Curriculum Development Alexandria, VA, 1998.
- [16] P. Jamieson and L. Grace, “A framework to help analyze if creating a game to teach a learning objective is worth the work,” in *2016 IEEE Frontiers in Education Conference*, 2016, pp. 1–7.
- [17] J. Zull, “The Art of Changing the Brain: Enriching the Practice of Teaching by Exploring the Biology of Learning,” 2002.
- [18] D. A. Kolb, *Experiential Learning*. Englewoods Cliff, NJ: Prentice Hall Inc., 1984.
- [19] D. R. Krathwohl, “A revision of bloom’s taxonomy: An overview,” *Theory into practice*, vol. 41, no. 4, pp. 212–218, 2002.
- [20] J. M. Lang, *Small teaching: Everyday lessons from the science of learning*. John Wiley & Sons, 2021.
- [21] A. Agrawal, J. Gans, and A. Goldfarb, *Power and prediction: The disruptive economics of artificial intelligence*. Harvard Business Press, 2022.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.

- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] G. Izacard and E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” *arXiv preprint arXiv:2007.01282*, 2020.
- [35] D. Kevian, U. Syed, X. Guo, A. Havens, G. Dullerud, P. Seiler, L. Qin, and B. Hu, “Capabilities of large language models in control engineering: A benchmark study on gpt-4, claude 3 opus, and gemini 1.0 ultra,” *arXiv preprint arXiv:2404.03647*, 2024.
- [36] C. Zou, X. Guo, R. Yang, J. Zhang, B. Hu, and H. Zhang, “Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models,” *arXiv preprint arXiv:2411.00836*, 2024.
- [37] B. Borges, N. Foroutan, D. Bayazit, A. Sotnikova, S. Montariol, T. Nazaretzky, M. Banaei, A. Sakhaeirad, P. Servant, S. P. Neshaei *et al.*, “Could chatgpt get an engineering degree? evaluating higher education vulnerability to ai assistants,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 49, p. e2414955121, 2024.
- [38] A. Efrat, O. Honovich, and O. Levy, “Lmentry: A language model benchmark of elementary language tasks,” *arXiv preprint arXiv:2211.02069*, 2022.
- [39] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Sholeh, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso *et al.*, “Beyond the imitation game: Quantifying and extrapolating the capabilities of language models,” *arXiv preprint arXiv:2206.04615*, 2022.
- [40] L. Schulze Balhorn, J. M. Weber, S. Buijsman, J. R. Hildebrandt, M. Ziefle, and A. M. Schweidtmann, “Empirical assessment of chatgpt’s answering capabilities in natural science and engineering,” *Scientific Reports*, vol. 14, no. 1, p. 4998, 2024.
- [41] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, “Gpqa: A graduate-level google-proof q&a benchmark,” *arXiv preprint arXiv:2311.12022*, 2023.