2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #52480

Evaluating Creativity and Novelty in Large Language Model–Generated Design Assignments for Digital Systems Education

Mrs. Alicia Baumann, Arizona State University

Alicia Baumann is an Associate Teaching Professor at the Ira A. Fulton Schools of Engineering at Arizona State University. With over a decade of experience in engineering education and prior industry experience in systems engineering at General Dynamics, her work focuses on first-year engineering, project-based learning, and inclusive, student-centered pedagogy. She has led large-scale mentorship and instructional initiatives, developed widely adopted curricular innovations, and published on student motivation and engagement in engineering education. Baumann has received multiple teaching and service awards, including ASEE Best Paper recognition and Top 5% Professor, and is committed to fostering environments where all students feel seen and supported.

Sai Vignesh Naragoni, Arizona State University

Sai Vignesh Naragoni is an undergraduate student in Computer Science at Arizona State University. His research interests include large language models, artificial intelligence in education, and agentic systems. He has worked on projects exploring LLM-generated educational content, Multi-Agent systems, and graph-based retrieval systems.

Evaluating Creativity and Novelty in Large Language Model–Generated Design Assignments for Digital Systems Education

Introduction

The increasing adoption of artificial intelligence (AI) technologies within higher education has prompted renewed attention to their role not only as learning tools for students but also as productivity and design aids for instructors. In engineering education, where project-based learning is central to developing problem-solving and design skills, faculty face recurring challenges in creating assignments that are simultaneously rigorous, engaging, and novel across academic terms. As large language models (LLMs) become more capable of generating structured and context-aware text, their potential to support academic assignment design warrants systematic investigation, particularly in domain-constrained engineering contexts.

While prior work has demonstrated that LLMs are effective in generating open-ended educational prompts and questions [1], [2], recent surveys note that their application to domain-constrained engineering assignments remains underexplored [3], [4]. Engineering assignments involving digital logic and hardware design require adherence to formal specifications, feasibility within curricular scope, and sufficient novelty to discourage reuse or solution memorization. Consequently, understanding whether LLMs can generate assignment prompts that satisfy these requirements is essential for their responsible integration into engineering education assignment design.

This paper evaluates the capability of four large language models to generate constrained, project-based assignment prompts for a freshman-level digital logic fundamentals course; EEE120: Digital Design Fundamentals at Arizona State University (ASU). The course introduces Boolean algebra, combinational and sequential logic, and basic microprocessor concepts, culminating in the design of a custom synchronous finite state machine (FSM). Because each offering of the course requires a distinct project prompt that remains feasible within defined logic constraints, the final project provides an ideal testbed for assessing LLM-generated academic assignments. Specifically, this study examines whether LLMs can produce project prompts that are both novel, in the sense of being meaningfully distinct from prior faculty-created assignments, and technically viable within the digital logic domain.

The primary objective of this work is to assess the usefulness of LLMs as collaborative tools for engineering educators in assignment creation. By systematically comparing model outputs and analyzing their creativity, novelty, and constraint adherence, this research aims to provide evidence-based guidance on how AI tools may be leveraged to enhance student engagement while reducing instructor workload. Ultimately, this paper seeks to empower faculty to adopt LLMs as informed partners in curriculum and assignment design, contributing to more sustainable, innovative, and academically sound engineering education practices.

From a pedagogical perspective, project-based assignments in digital logic courses are critical for developing students' ability to translate abstract Boolean concepts into structured system designs. The quality of these assignments directly influences students' engagement, depth of understanding, and ability to apply design principles in open-ended contexts. As such, evaluating LLM-generated assignments requires not only assessing novelty and feasibility, but also considering how well these prompts support meaningful learning experiences, including problem formulation, design trade-offs, and system-level reasoning.

To guide this investigation, the following research questions were formulated to evaluate the effectiveness of large language models in generating constrained, creative, and instructionally viable assignment prompts:

1) How effectively can large language models (LLMs) generate project-based assignment prompts that satisfy domain-specific technical constraints in a freshman-level digital logic course?
2) How do different LLMs (GPT-4, Gemini, LLaMA, and DeepSeek R1) compare in their ability to produce assignment prompts that balance novelty and feasibility?
3) What is the impact of retrieval-augmented generation (RAG) versus non-RAG prompting on the creativity (novelty) and pedagogical appropriateness (feasibility) of LLM-generated assignments?

Literature Review

The rapid advancement of large language models (LLMs) has expanded their potential applications across numerous domains, including educational content creation. Prior research has demonstrated that LLMs are capable of generating fluent, coherent, and contextually appropriate educational materials such as explanations, reflective prompts, and open-ended assessment questions [5], [6]. However, the extent to which these models can generate *novel, technically valid assignments* within the strict logical and design constraints characteristic of engineering education remains an open research question. This literature review examines existing work on LLM creativity, reasoning, and educational applications, with particular emphasis on identifying gaps related to constraint-aware creativity and systematic novelty evaluation in engineering contexts.

Early foundational work by Brown et al. demonstrated that large-scale language models trained on extensive corpora exhibit strong few-shot learning capabilities, enabling them to generate meaningful outputs from limited prompting [5]. Building on this, Wei et al. showed that as LLMs scale in size and training complexity, they display emergent abilities such as improved reasoning, abstraction, and creative recombination of knowledge [7]. These findings suggest that LLMs possess the underlying representational capacity necessary to support creative educational tasks. However, these studies primarily evaluate general language performance and reasoning

benchmarks, leaving open questions regarding how such capabilities translate to domain-specific engineering tasks that impose formal logical and structural constraints.

Subsequent work has explored methods to improve the reasoning reliability of LLM outputs. Wei et al. introduced chain-of-thought (CoT) prompting, a technique that encourages models to generate intermediate reasoning steps prior to producing a final answer [8]. CoT prompting has been shown to significantly improve performance on tasks requiring multi-step reasoning, mathematical problem solving, and logical consistency [8]. In the context of engineering assignment generation, such prompting techniques are particularly relevant, as they offer a mechanism for improving adherence to design constraints and internal coherence. However, while CoT improves reasoning transparency and correctness, it does not directly address whether the resulting outputs are *novel* relative to existing assignments or merely reformulations of learned patterns.

Parallel to advancements in reasoning techniques, recent research has examined the role of retrieval-augmented generation (RAG) versus non-retrieval-based LLM implementations. Wei et al. describe non-RAG LLMs as relying entirely on parametric knowledge encoded during training, which limits their ability to verify novelty and increases the risk of reproducing training-derived content [7]. In contrast, retrieval-augmented approaches incorporate external knowledge sources at inference time, enabling models to ground outputs in curated or task-specific datasets [7]. While RAG-based methods have demonstrated improvements in factual accuracy and domain grounding, their implications for creative educational content generation remain underexplored. In assignment design contexts, retrieval may improve feasibility and constraint adherence while simultaneously biasing outputs toward existing examples, potentially limiting novelty. As a result, evaluating LLM-generated assignments under non-RAG conditions provides insight into the models' intrinsic creative capabilities independent of external content grounding.

Within educational research, Zawacki-Richter et al. provide a comprehensive review of artificial intelligence applications in higher education, identifying content generation and instructional support as prominent use cases [6]. While the review highlights AI's potential to enhance student engagement and personalize learning experiences, it also raises concerns regarding originality, academic integrity, and instructor trust. These concerns are particularly salient in engineering education, where repeated reuse of similar assignments can diminish learning outcomes and increase opportunities for solution sharing. Consequently, the ability of LLMs to generate genuinely novel assignment prompts has direct implications for both pedagogy and academic integrity.

Despite demonstrated strengths in language generation, critiques by Marcus argue that LLMs often lack deep conceptual understanding and robust reasoning, particularly in tasks requiring strict adherence to domain-specific rules [9]. This limitation raises questions about whether

LLM-generated assignments can reliably reflect the underlying engineering principles they are intended to assess. Related work by Petroni et al. characterizes LLMs as implicit knowledge bases, showing that while they excel at recalling and recombining known information, their outputs remain constrained by the distribution of their training data [10]. These findings suggest that LLM creativity frequently manifests as recombination rather than true conceptual novelty, reinforcing the need for explicit evaluation frameworks capable of distinguishing between superficial variation and meaningful innovation.

Quantitative evaluation of novelty and uniqueness has been explored in both natural language processing and engineering design literature through vector-space similarity analysis. Reimers and Gurevych introduced Sentence-BERT, a framework for generating semantically meaningful sentence embeddings that enable efficient similarity comparison using cosine distance [11]. Their work demonstrates that embedding-based cosine similarity provides a robust mechanism for measuring semantic overlap between textual artifacts, allowing researchers to distinguish between genuinely distinct content and paraphrased or minimally altered text. In educational and generative contexts, such approaches enable objective comparison of AI-generated outputs against existing corpora, supporting automated assessments of originality beyond surface-level lexical variation.

Complementing this natural language processing perspective, Chakrabarti and Khadilkar proposed a formal novelty metric for engineering design that evaluates how distinct a proposed solution is relative to prior designs within a defined solution space [12]. Their framework quantifies novelty based on the degree of deviation from existing design features rather than unrestricted creativity, emphasizing that meaningful innovation must be evaluated relative to domain constraints and functional feasibility. Although originally applied to physical product design, this novelty assessment paradigm aligns closely with embedding-based similarity analysis, where cosine distance can serve as a proxy for design differentiation in abstract feature spaces. Together, these studies provide a theoretical and methodological foundation for using vector cosine similarity to quantify the uniqueness of LLM-generated assignment prompts relative to faculty-created exemplars, enabling structured, repeatable evaluation of novelty within constrained engineering education contexts.

Collectively, prior work establishes that LLMs are capable of generating fluent educational content and that techniques such as chain-of-thought prompting and retrieval augmentation can improve reasoning and grounding [7], [8]. However, the literature also reveals a significant gap in systematic evaluation of LLM-generated assignments within constrained engineering domains, particularly with respect to measurable creativity and novelty [6], [7]. This gap motivates the present study, which applies structured novelty evaluation methods to assess whether LLMs can generate feasible, constraint-compliant, and genuinely novel project prompts for digital logic design courses.

Digital Design Project Description

The digital design course at Arizona State University culminates in an individual capstone-style project in which each student designs a synchronous finite state machine (FSM) based on a real-world system description. To ensure sufficient design complexity, the FSM must incorporate at least two binary inputs and a minimum of five distinct states. Because both states and inputs are represented in binary, this requirement necessitates the use of at least three state bits and two input bits, resulting in a five-variable truth table and corresponding Karnaugh maps. This constraint establishes the minimum design complexity required for students to demonstrate mastery of core digital logic concepts.

Although synchronous FSMs often follow intuitive state progressions, real-world behavior frequently introduces non-linear transitions based on changing inputs. For example, a vending machine FSM may progress through states as money is inserted, but additional behaviors, such as refunds, overpayment, or cancellations can cause transitions to skip, repeat, or reverse states. These non-linear behaviors must be explicitly accounted for in the design, requiring students to reason beyond simple counting mechanisms. This can be seen in a simplified state transition diagram below showing basic transitions while omitting several custom assumptions.

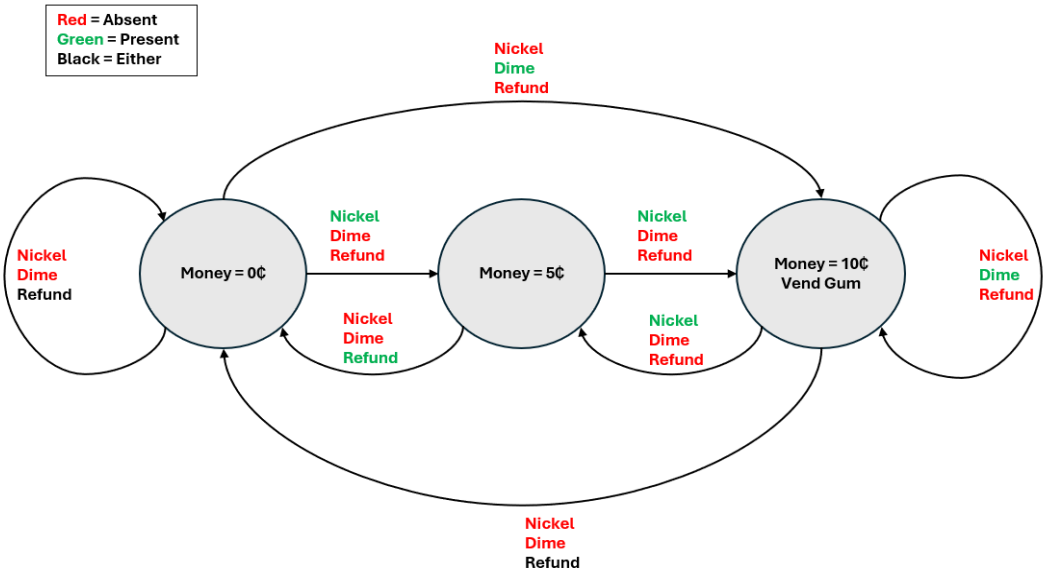

Figure 1: Simple Vending Machine FSM State Diagram

Designing effective project prompts for this course therefore requires careful balancing of constraints and openness. Prompts must be sufficiently constrained to limit state explosion and maintain feasibility within course scope, yet open-ended enough to allow for multiple valid interpretations and student-specific design decisions. Over time, the course has accumulated a diverse repository of FSM project prompts, including systems such as vending machines, scoreboards, gas pumps, automated wheelchairs, and key fob controllers. Below is a sample prompt:

"A car seat is designed to ensure that a parent does not forget a sleeping child in the back of their car. There are two sensors in the car seat. The first one is located directly under the child. The purpose of this sensor is to detect if the seat is being occupied by a child. The other sensor detects if a fob key is 10 meters away from the seat. If the key is over 10 meters away, the sensor will give a high reading. If the key is less than 10 meters away, the key will give a low reading. There are two outputs; a notification that is sent to the parent if the child is left in the car, and a cooling system that turns on if the child remains in the car while the parent is away.

If the child is in the seat and the parent walks away with the fob key, then the circuit will move into a first violation state and a notification will be sent to the parent. If this same reading is made on the next clock cycle, then the machine will move into a second violation state and another notification will be sent. If this happens again the machine will move toward an emergency state where both the cooling system is turned on and the notification is sent. In this emergency state, the cooling system will only be turned off if the child is removed from the seat. However, if a parent approaches a car, within 10 meters, while in the emergency state the machine will not send a notification, but it will keep the cooling system on."

Each prompt provides a concise description of the problem space, including required inputs and outputs, while intentionally leaving certain behaviors unspecified. This ambiguity encourages students to make and justify design assumptions, resulting in unique implementations even when based on the same prompt. As a result, the creation of new FSM design prompts each semester is both pedagogically valuable and time-intensive, making this course an appropriate testbed for evaluating LLM-generated assignment prompts under realistic engineering constraints.

Methodology

This study employs a mixed-methods evaluation framework to assess the capability of large language models (LLMs) to generate novel and feasible project assignment prompts for a freshman-level digital logic course. Four LLMs (GPT-4, Gemini, LLaMa, and DeepSeek R1) were evaluated under controlled prompting conditions to examine differences in creativity, uniqueness, and technical feasibility. Both quantitative and qualitative analyses were conducted to capture complementary aspects of assignment quality.

All models were prompted using an identical base prompt that specified course level, technical constraints, and deliverable expectations. The prompt instructed the model to generate a project assignment suitable for a freshman-level digital logic course and explicitly required the use of a synchronous FSM. Model temperature and generation length were held constant across trials to reduce variance attributable to sampling parameters.

LLM Selection and Experimental Design

The four LLMs selected for this study represent a range of widely used proprietary and open-source architectures with demonstrated performance in reasoning and content generation tasks.

GPT-4 and Gemini were selected as representative state-of-the-art commercial models, while LLaMa and DeepSeek R1 were included to evaluate open and research-oriented models with varying architectural and training characteristics.

Each model was evaluated under two configurations:

1) **Non-retrieval-augmented generation (non-RAG)**, in which the model relied solely on its internal parametric knowledge
2) **Retrieval-augmented generation (RAG)**, in which the model was provided access to a curated corpus of prior faculty-created project prompts and course-relevant documentation during generation.

This comparison was motivated by prior work demonstrating that retrieval augmentation can improve grounding and factual accuracy, while potentially constraining creative divergence due to exposure to existing examples [7]. For each model and configuration, three independent trials were conducted using a standardized base prompt. This resulted in a total of 24 generated project prompts (4 models × 2 configurations × 3 trials). Trials were conducted independently to account for stochastic variation in model outputs and to assess consistency across generations.

Quantitative Novelty and Creativity Evaluation

Novelty and creativity were evaluated quantitatively using a vector-space similarity approach grounded in prior work on semantic embedding analysis and engineering design novelty metrics. Each generated assignment prompt was embedded into a high-dimensional semantic vector space using a sentence embedding model capable of capturing semantic similarity beyond surface-level lexical overlap [11]. Specifically, embeddings were generated using the all-MiniLM-L6-v2 sentence transformer. Faculty-created historical project prompts were embedded using the same model to form a reference corpus of projects used in prior offerings of EEE120 at ASU.

The all-MiniLM-L6-v2 model was selected due to its strong performance on semantic similarity benchmarks and its ability to produce consistent sentence-level embeddings with low computational overhead. Given the relatively short length and structured nature of assignment prompts, this model provides an appropriate balance between semantic sensitivity and efficiency, making it well-suited for comparative similarity analysis across a moderate-sized corpus.

Cosine similarity was computed between each LLM-generated prompt and all prompts in the reference corpus. Lower cosine similarity values indicate greater semantic distance and, consequently, higher uniqueness relative to existing assignments. For each generated prompt, the minimum cosine similarity score against the corpus was used as the primary novelty metric, reflecting the closest existing analogue. Novelty was calculated as 1 minus the minimum cosine similarity score.

The use of the minimum cosine similarity as the basis for the novelty metric reflects a conservative approach, where each generated prompt is evaluated against its closest existing analogue. This ensures that even partial overlap with prior assignments is captured, aligning with the instructional goal of minimizing reuse or near-duplicate prompts.

Creativity was operationalized as a function of semantic uniqueness rather than unconstrained ideation. This definition aligns with engineering design frameworks that characterize novelty as deviation from existing solutions within a constrained solution space [12]. By using cosine distance as a quantitative proxy for differentiation, this method provides a repeatable and objective measure of novelty while remaining sensitive to domain-relevant semantic structure.

Qualitative Feasibility Assessment

While novelty captures creative differentiation, feasibility is essential for instructional validity in engineering education. To evaluate feasibility, each generated prompt was assessed qualitatively using a structured rubric consisting of six binary (yes/no) criteria derived from the instructional and technical requirements of EEE120: Digital Design Fundamentals at ASU:

1) Does the prompt require at least two distinct human inputs (e.g., buttons, switches)?
2) Does the prompt produce at least one observable output (e.g., LED, signal)?
3) Does the prompt meet all design requirements, including the implementation of a synchronous finite state machine (FSM) with at least five states?
4) Does the prompt provide sufficient assumptions and contextual information to support implementation?
5) Is the prompt aligned with the knowledge and skills taught in EEE120?
6) Is the prompt clear and feasible for an EEE120 student to complete within the course scope?

Each criterion was scored independently, resulting in a feasibility score ranging from 0 to 6 for each prompt. Binary scoring was selected to reduce subjectivity and ensure consistent evaluation across prompts, particularly given the small sample size and the need for clear instructional thresholds. A prompt was considered instructionally viable only if it satisfied all core technical requirements, including FSM implementation and observable input-output behavior, reflecting minimum expectations for successful student completion within the course. This threshold-based approach prioritizes baseline pedagogical validity over relative ranking. Prompts that failed to satisfy the FSM requirement or output generation criterion were classified as infeasible regardless of total score. This approach is consistent with prior critiques, noting that LLM outputs may appear fluent while failing to satisfy underlying technical constraints [8].

These feasibility criteria are not only technical requirements but also proxies for instructional effectiveness, as they ensure that generated prompts require students to engage in core learning activities such as state modeling, input-output mapping, and iterative design reasoning.

Data Analysis and Comparison

Quantitative novelty scores and qualitative feasibility scores were analyzed across models, configurations (RAG vs. non-RAG), and trials. Comparisons were used to evaluate trade-offs between creativity and feasibility, as well as the impact of retrieval augmentation on both metrics. By combining semantic similarity analysis with structured feasibility evaluation, this methodology enables a comprehensive assessment of whether LLM-generated assignment prompts are not only unique but also pedagogically and technically appropriate for use in a constrained engineering education context. The purpose of this analysis is exploratory rather than inferential. Given the limited number of models and trials, the study emphasizes comparative patterns and observable trends across configurations rather than statistical significance testing or performance ranking.

This study focuses on the generation and evaluation of assignment prompts and does not include deployment in an active classroom setting; therefore, student learning outcomes were not directly measured.

Results

Results are presented using descriptive statistics and visual comparisons to highlight trends in novelty and feasibility across models and prompting configurations. These results are intended to reveal relative behaviors and trade-offs rather than to establish statistically significant performance differences.

Qualitative Results - Feasibility Analysis

Feasibility was evaluated using a six-item binary rubric reflecting core instructional and technical requirements of a freshman-level digital logic course. Each generated assignment prompt received a feasibility score ranging from 0 to 6, corresponding to the number of rubric criteria satisfied. Figure 2 summarizes the mean feasibility scores aggregated across all models under non-RAG and RAG configurations. Under non-RAG conditions, feasibility scores were low across models, indicating that most generated prompts failed to satisfy multiple core technical requirements. In contrast, RAG-based generation resulted in substantially higher feasibility scores, with average performance exceeding five out of six rubric criteria.

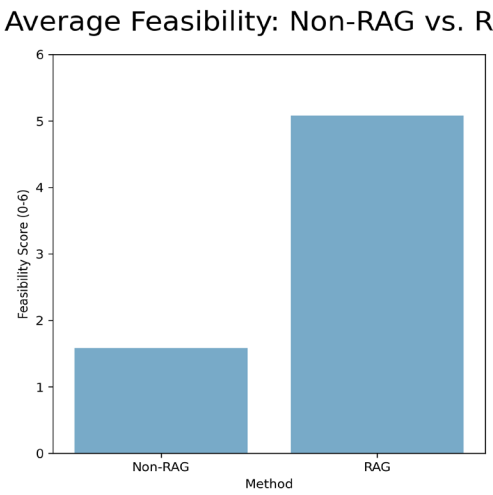

Figure 2: Average Feasibility Comparison per Configuration

To examine the feasibility at the level of individual rubric criteria, Figure 3 presents criterion level scores for each model under non-RAG conditions. While models frequently satisfied basic structural requirements such as the inclusion of at least one input or output, they consistently failed during the stricter technical constraints. Particularly, requirements related to synchronous FSM implementation and sufficient state complexity were rarely satisfied, and contextual assumptions were often underspecified.

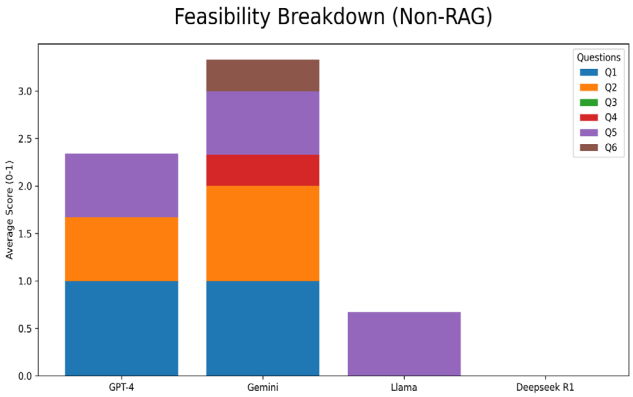

Figure 3: Non-RAG Feasibility Breakdown

In contrast, Figure 4 shows that retrieval augmentation led to consistent improvements across all rubric criteria. Prompts generated under RAG conditions more frequently satisfied FSM requirements, incorporated multiple inputs and observable outputs, and provided clearer contextual framing. These improvements were observed across all evaluated models, though absolute feasibility levels varied by model.

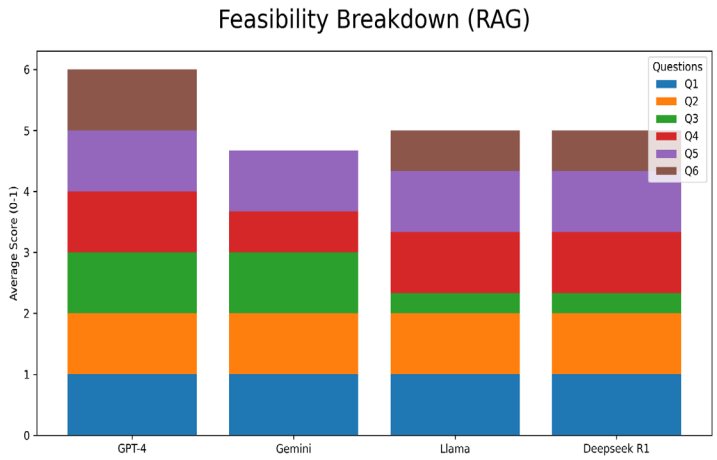

Figure 4: RAG Feasibility Breakdown

Together, these results demonstrate that retrieval augmentation substantially improves the instructional feasibility of LLM-generated project prompts by increasing adherence to domain-specific technical constraints.

Quantitative Results - Novelty Analysis

Each generated prompt was analyzed for its novelty using a cosine similarity computation. After all three trials on all four LLMs were conducted, the average novelty score was calculated to analyze if a substantial difference exists between the non-RAG prompts versus the RAG prompts. As seen below in Figure 5, the method of creation had mixed results depending on which LLM was employed. While DeepSeek R1 and GPT-4 had slightly more creative results with the RAG approach, Gemini and LLaMa had better novelty scores while using the non-RAG implementation, indicating that by training the LLM with prior samples, the LLM loses some originality of thought.

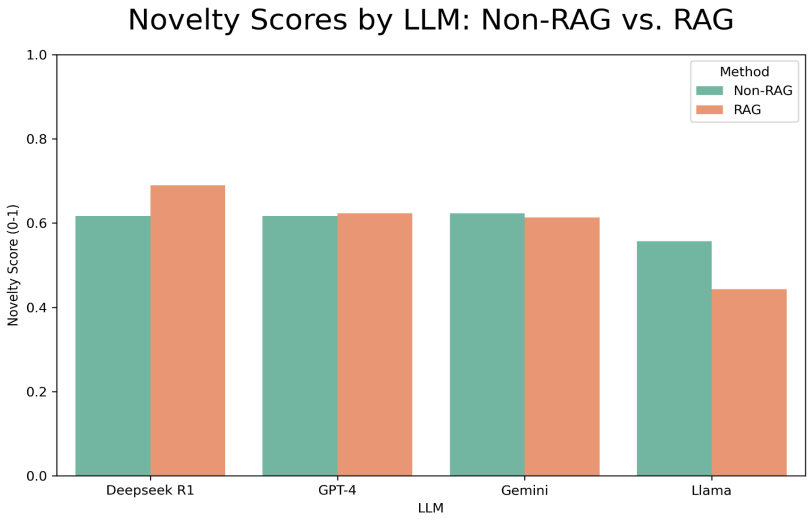

Figure 5: Novelty Scores by LLM Comparison

Integrated Analysis: Novelty vs. Feasibility

The novelty versus feasibility graph shown in Figure 6 below, provides a visual summary of the trade-offs observed across LLM-generated assignment prompts. Plotting these dimensions together highlights that novelty alone is insufficient for instructional use if feasibility is compromised, and conversely, that highly feasible prompts may offer limited pedagogical value if they closely resemble existing assignments. As such, Figure 6 reveals that the RAG implementation of GPT-4 and Deepseek R1 scored collectively well in both feasibility and novelty. Meanwhile, the non-RAG implementations of all four LLMs scored within a similar cluster distribution of each other indicating mediocre feasibility and similar novelty.

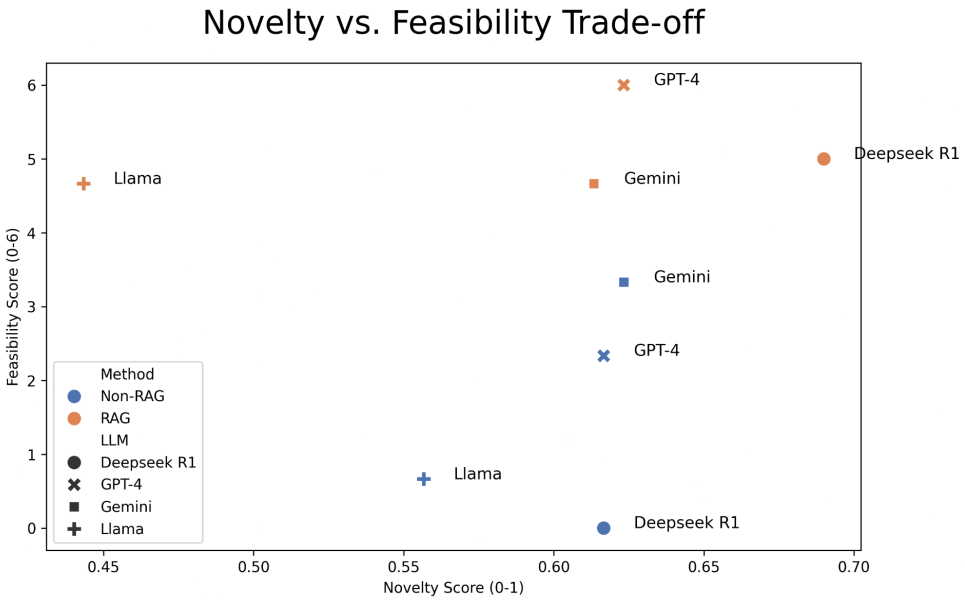

Figure 6: Novelty vs. Feasibility Comparison

From a pedagogical standpoint, the balance between novelty and feasibility has direct implications for student learning. Highly novel but infeasible prompts may introduce ambiguity that exceeds students' current skill level, leading to confusion rather than productive struggle. Conversely, highly feasible but low-novelty prompts may limit opportunities for creative problem solving and reduce student engagement due to familiarity with prior examples. The results suggest that prompts generated under RAG configurations more consistently fall within an instructional "sweet spot," where sufficient structure supports successful implementation while maintaining enough variation to encourage independent reasoning and design exploration.

While the quantitative metrics provide an objective comparison of novelty and feasibility across models, they do not fully capture how these differences manifest in the structure and clarity of individual assignment prompts. To complement the numerical analysis, representative sample outputs from each model are presented below to illustrate qualitative differences in prompt formulation and technical framing.

Sample response from Deepseek R1 (RAG - Trial 3):
This response, while lengthy and verbose showing all train-of-thought, resulted in a Feasibility score of 6 out of 6 and a Novelty score of .74.

"Alright, let me break this down step by step. So, I need to come up with a unique Capstone project idea for the Digital Design course. The project has to involve a synchronous FSM with the Moore model, have at least 5 internal states, 2 human inputs, and 1 output. Plus, it needs to be distinct from past projects and align with the syllabus objectives. Okay, let's start by understanding what's needed.

First, the project should use both Moore and Mealy models. Wait, the syllabus mentions both, but the project only needs to focus on one, probably Moore since the user specified. So I'll stick with Moore. Moore machines have outputs that depend only on the current state, so I'll need to design that.

Next, at least 5 internal states. That means I'll need a state diagram with 5 distinct states. Let me think about a common application for such a FSM. Voting systems are a classic, but the user mentioned that's been done. Maybe something else, like a traffic light controller, but that's also been used. Hmm. What about a simple calculator? But that might involve more complex operations. Maybe a state machine for a small robot, but that could get complicated.

Wait, what about a FSM that controls a simple elevator? But elevator might have more states. Let's think of something simpler. Maybe a FSM for a small home automation system, like controlling a lamp based on certain inputs. But that's too simple. Need something that's a bit more involved but still manageable for a capstone project.

Oh, how about a FSM that controls a basic traffic light controller with pedestrian detection? That involves multiple states: green, yellow, red, maybe with pedestrian detection adding more complexity. Let's see:

- Green light with pedestrian detection: Maybe pedestrian waiting or passing.
- Yellow light: Change to red.
- Red light: Pedestrian detection could allow them to cross or stop.
But wait, that might only give me a few states. Need at least five. Maybe include different pedestrian detection modes or pedestrian presence.

Alternatively, think of a FSM for a simple elevator controller. States could be: stop, going up, going down, stopped at floor, arrived at floor. That's five states. But I need two human inputs and one output.

Wait, let's clarify: the project needs two human inputs, which could be buttons or switches. So, for the elevator, maybe inputs are 'up' and 'down' buttons, and the output is the direction the elevator is moving.

But let's outline the requirements again:

- 5 internal states
- 2 human inputs (each can be 0 or 1)
- 1 output (0 or 1)
- Moore FSM (outputs depend only on current state)
- At least one assumption, like synchronous operation
- State definitions, state transition table, current state next state transitions with inputs and outputs.

Let me outline the states for an elevator:

1. Stopped at current floor (S0)
2. Moving up (S1)
3. Moving down (S2)
4. At top floor (S3)"

Sample Response from LLaMa 3.2 (RAG - Trial 3):
This response does not have enough detail to fully understand the purpose of the machine or what states could exist. This resulted in a Feasibility score of 4 and Novelty score of .44.

"- Summary: A music player is designed with a synchronous sequential machine to control the playback of a song.
- Inputs: 'C' (0: song is playing, 1: song is paused), 'D' (0: volume is at maximum, 1: volume is at minimum).
- Output: 'Y' (1: song is playing), 0 (song is paused).
- Assumption: The system assumes the user will not press the pause button while the song is playing.

- Summary: The music player controls the playback of a song based on user input.
- Inputs: C and D, 00 (playing), 01 (paused), 10 (volume max), 11 (volume min).
- Output: Y, 0 (playing),"

Sample response from GPT-4 (RAG - Trial 3):
While this response is difficult to read given the format, it is a high scoring result with a Feasibility score of 6 and a Novelty score of .56.

*"**Project Title: Digital Logic Design for a Smart Pet Feeder Control System**\n\n**Project Description:**\nThis project involves designing a synchronous finite state machine (FSM) using a Moore model to control a smart pet feeder. The system will manage feeding schedules and portion sizes based on user inputs. The purpose is to automate pet feeding tasks while ensuring proper feeding times and amounts, enhancing convenience for pet owners.\n\n**Inputs:**\n1. **Time Input (T):** A binary input signal representing scheduled feeding times (0 for off-schedule, 1 for scheduled feeding time).\n2. **Portion Size Selector (P):** A binary input signal to choose between two feeding portion sizes (0 for small portion, 1 for large portion).\n\n**Output:**\n- **Feeder Activation (A):** A binary output signal that controls the feeder mechanism (1 to activate feeder, 0 to deactivate).\n\n**State Design:**\nThe FSM will have at least five states to manage various feeding scenarios:\n1. **Idle State:** Default state when no feeding is scheduled.\n2. **Small Portion Feeding State:** Activates the feeder for a small portion when the scheduled time arrives.\n3. **Large Portion Feeding State:** Activates the feeder for a large portion when the scheduled time arrives.\n4. **Waiting State:** A transitional state that ensures the feeder completes its operation before returning to Idle.\n5. **Error Handling State:** Manages any errors like simultaneous portion requests or mechanical faults.\n\n**Purpose and Functionality:**\n- The FSM will monitor the time input to determine if it's a scheduled feeding time.\n- Depending on the portion size selector input, the FSM will transition to either the small or large portion feeding state.\n- Once the feeder is activated for the specified portion, the FSM will transition to the waiting state to ensure the feeding cycle completes before returning to idle.\n- If any errors are detected, such as invalid input combinations, the FSM will transition to the error handling state to manage the issue.\n\n**Adherence to Course Objectives:**\nThis project aligns with the course objectives by requiring students to analyze, design, and debug a digital logic circuit with a synchronous FSM. It involves creating a state transition diagram, simplifying logic using Boolean algebra, and constructing a practical digital system that integrates combinational and sequential logic components. The project also encourages students to determine optimal design solutions through the selection of efficient state transitions and logic simplification.\n\nThis unique project idea provides a practical application for digital design concepts, distinct from traditional projects like traffic controllers or voting systems, and does not overlap with past semester projects."*

Sample response from Gemini (RAG - Trial 1)

This LLM approach provided the cleanest and most user-friendly result to interpret. It scored a 4 in Feasibility as it did not provide sufficient assumptions or context, and the overall prompt is lacking notable information to understand the correct sequence. This scored a .39 in novelty as patterned sequences are very commonly taught within digital logic FSM examples.

"- Summary of the project and its purpose: Design a digital combination lock with a rolling code feature. The lock opens only if the correct three-digit sequence is entered within a specified number of attempts. Each correct digit entered advances the state. If

an incorrect digit is entered, the system resets to the initial state. A limited number of incorrect sequence attempts will permanently lock the safe until reset.
- Inputs:
- `DIGIT` (2-bit input - 00, 01, 10, 11): Represents the digit entered by the user. These are the 4 possible digits in the combination."
- `ATTEMPT` (1-bit input): A signal from the user indicating that a digit has been entered. This signal debounces the input and allows time between digit entries.
- Output: `UNLOCK` (1-bit output): 1 indicates the lock is open; 0 indicates the lock is locked.
- Assumption: The correct three-digit sequence is pre-programmed into the FSM, and the attempt counter is also pre-programmed."

After analyzing all prompts from all trials, GPT-4 under RAG conditions continually achieved high feasibility scores along with novelty scores above .6. The non-RAG implementation of LLaMa 3.2 exhibited the lowest alignment with rubric criteria. It is seen that RAG significantly improved feasibility (average score: 5.08 vs. 1.58 for Non-RAG), with GPT-4 and Deepseek R1 achieving perfect scores (6/6) in RAG mode, indicating strong adherence to context constraints. Novelty remained stable overall (0.60 vs. 0.60), though model-specific trends emerged: Deepseek R1's novelty rose (0.62 to 0.69), while LLaMa's dropped (0.56 to 0.44). Bar charts confirmed RAG's feasibility boost, scatter plots revealed a novelty-feasibility balance (RAG clustering higher), and radar charts highlighted GPT-4 and Deepseek R1 as top performers. These results validate RAG's enhancement of practical prompt generation.

Lessons Learned

RAG significantly enhances LLM-generated project prompts for digital logic courses, improving feasibility (5.08 vs. 1.58 for Non-RAG) while maintaining novelty (0.60 for both). GPT-4 and Deepseek R1 excel in RAG mode, achieving perfect feasibility (6/6) and high novelty (0.62-0.69), making them ideal for faculty use. The feasibility breakdown reveals RAG consistently improves adherence to technical criteria (e.g., Q3: FSM requirements). However, models often generated basic projects (e.g., traffic light controllers), likely due to training data biases, though some novel ideas emerged. This confirms context-aware LLMs balance creativity and usability, offering a superior tool for educational prompt design.

From an instructional perspective, this study highlighted that effective use of large language models for assignment creation requires understanding the distinct strengths and limitations of each model rather than treating AI as a uniform tool. The evaluation process revealed clear idiosyncrasies in how different LLMs interpret constraints, balance creativity with feasibility, and respond to contextual grounding, underscoring the importance of informed model selection and prompt design. More importantly, this research provided a practical roadmap for integrating AI into curriculum development: starting with clearly articulated technical constraints,

selectively incorporating retrieval-based context to improve feasibility, and applying systematic evaluation metrics to vet outputs before classroom adoption. These lessons emphasize that AI is most effective when used as a collaborative design aid supporting instructor creativity and efficiency, rather than as an autonomous content generator.

Conclusions

The primary objective of this study was to examine how large language model (LLM) prompting strategies influence creativity and pedagogical appropriateness in digital systems design assignments. The findings demonstrate that LLMs can effectively support context-specific assignment creation when guided by clearly defined technical constraints and instructional goals. In particular, GPT-4 and DeepSeek, when deployed using retrieval-augmented generation (RAG), consistently produced project prompts that balanced high feasibility with meaningful creative variation relative to previously used assignments.

Although many LLM-generated prompts required additional refinement to match the level of structure and instructional clarity typically produced by faculty, the models nevertheless introduced a wide range of novel project concepts—including pet monitoring systems, encryption-based controllers, and elevator control logic—that expand the available design space for instructors. Access to such varied ideas is valuable for sustaining student engagement, reducing academic integrity concerns associated with repeated assignments, and introducing students to contemporary and diverse application domains.

Overall, this work suggests that the gap between AI-generated content and faculty-designed instructional materials is narrowing as LLMs increasingly demonstrate improved conceptual understanding and adherence to domain-specific constraints. When used as collaborative design aids rather than autonomous content generators, LLMs offer a promising pathway toward more sustainable, innovative, and responsive assignment development in engineering education.

While this work evaluates the feasibility and novelty of LLM-generated assignments, these prompts were not implemented in a live course setting. The impact on student learning remains an area for investigation. Future work will involve implementing this framework in an active classroom setting where LLM-generated assignment prompts will be incorporated into course offerings and evaluated based on student performance, engagement, and learning outcomes. By comparing student responses to AI-generated versus traditionally developed assignments, this next phase aims to assess the instructional effectiveness of AI-assisted curriculum design and validate whether increased novelty and feasibility translate to improved learning experiences.

Acknowledgements

AI-based tools were used to assist with drafting and language refinement; however, all research design, analysis, results, interpretations, and conclusions presented in this paper are the original work of the authors.

References

[1] S. Al Faraby, A. Romadhony, and Adiwijaya, "Analysis of large language models for educational question classification and generation," *Computers & Artificial Intelligence*, vol. 43, no. 2, pp. 1–14, 2024.

[2] N. Scaria, S. Dharani Chenna, and D. Subramani, "Automated educational question generation at different Bloom's skill levels using large language models," *arXiv preprint*, arXiv:2408.04394, 2024.

[3] S. Wang, T. Xu, H. Li, C. Zhang, J. Liang, J. Tang, P. Yu, and Q. Wen, "Large language models for education: A survey and outlook," *arXiv preprint*, arXiv:2403.18105, 2024.

[4] L. Yan, L. Sha, L. Zhao, Y. Li, R. Martinez-Maldonado, G. Chen, X. Li, Y. Jin, and D. Gašević, "Practical and ethical challenges of large language models in education: A systematic scoping review," *arXiv preprint*, arXiv:2303.13379, 2023.

[5] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[6] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education—Where are the educators?" *Int. J. Educational Technology in Higher Education*, vol. 16, no. 1, p. 39, 2019.

[7] J. Wei *et al.*, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022. [Online]. Available: https://openreview.net/forum?id=yzkSU5zdwD

[8] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2023.

[9] G. Marcus, "GPT-3, bloviator: OpenAI's language generator has no idea what it's talking about," *MIT Technology Review*, Aug. 2020. [Online]. Available: https://www.technologyreview.com/

[10] F. Petroni *et al.*, "Language models as knowledge bases?" in *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, China, 2019, pp. 2463–2473.

[11] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[12] A. Chakrabarti and P. Khadilkar, “A measure for assessing product novelty,” in *Proc. Int. Conf. on Engineering Design (ICED 2003)*, Stockholm, Sweden, Aug. 2003, pp. 1–8.