

**Volume 1, Issue 1**

**Research Article**

**Date of Submission:** 16 May, 2025

**Date of Acceptance:** 14 Aug, 2025

**Date of Publication:** 20 Aug, 2025

## Beyond Data: How Simulation Can Empower LLMs to Revolutionize Science

**M. Garbey\***

Care Constitution Corp, USA  
LaSIE UMR-CNRS 7356 University of La Rochelle, France

**\*Corresponding Author:**

M. Garbey, Care Constitution Corp – Houston, TX, USA: LaSIE UMR-CNRS 7356 University of La Rochelle, France

**Citation:** Garbey, M. (2025). Beyond Data: How Simulation Can Empower LLMs to Revolutionize Science. *J AI VR Hum Comput*, 1(1), 01-11.

### Abstract

We hypothesize that large language models (LLMs) can effectively generate ‘thought experiments’ that can be evaluated using virtual simulators. This approach has the potential to automate knowledge generation, accelerate scientific discovery, and provide a systematic method for training subsequent LLMs with verifiable and reproducible datasets. To illustrate this concept, we will use the clinical workflow optimization challenge as a case study.

**Keywords:** Artificial Intelligence, Heuristic Computer Reasoning, Thought Experiment, Numerical Simulation, Scientific Discovery, Generative Language Model, Semantic

### Thanks

This manuscript has taken advantage of the generative language model Gemini from Google in multiple ways, from the basic “conversation” reported below that serves as the material for our experiences to some editing. We thank Google for making Gemini freely available to users.

### Introduction

To describe how Large Language Models (LLMs) operate is not simple. The first phase is the training phase where the language models use unsupervised learning to predict the most probable next word in a sequence based on the context provided by the preceding words. The term “generative” refers to this capacity to extrapolate with new words to extend a sentence. An essential component of the system includes the transformer architecture, which allows the system to capture long-range dependencies in the input data [1]. A large body of fine-tuning methods and sampling techniques complete the work – see the AI-Open Journal for example.

However, it is recognized today that while the principle of LLMs is somewhat well-known and well-published, nobody truly understands why it works to the extent that we observe today. In fact, LLMs mimic human behavior so well that one should be concerned about bias and hallucination systematically [2]. Consequently, LLMs in their current state may not be suitable for generating scientific discoveries.

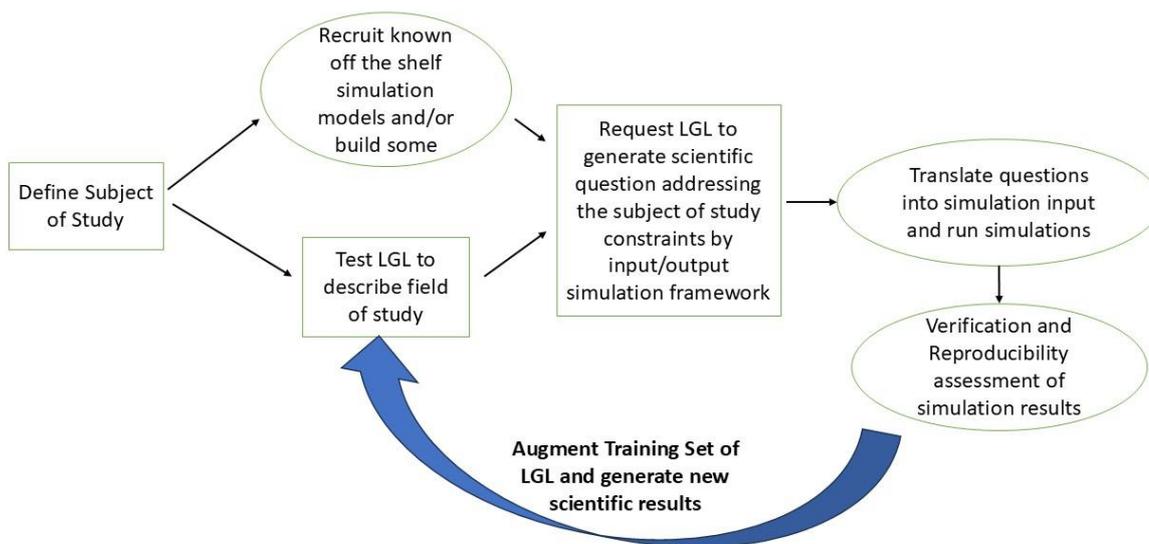
The convergence of the following ideas may support the potential use of generative language for science discovery: Asking questions is the most important activity that prefigures most discovery [3].

- “If I had an hour to solve a problem and my life depended on the solution, I would spend the first 55 minutes determining the proper question to ask.” – quote attributed to Albert Einstein
- While LLMs may not be adept at asking the most relevant questions, they can grab and “extrapolate” on large quantities of relevant questions that have been “floating” in the immense body of data used for training, some of them

being right on the spot [4]. We can use the brute force of LLMs to generate dozens of questions related to any given field of science quickly.

- It is not rare when the same scientific discovery is made by two different teams independently at about the same time: examples include the theory of evolution with Charles Darwin and Alfred Russel Wallace, General Relativity with A. Einstein and D. Hilbert, the periodic table of elements with D. Mendeleev and J.L. Meyer, etc. This demonstrates somehow the ubiquity of good questions that led to these discoveries even if they are not easy to find [5].
- Using LLMs to generate questions is less risky than using LLMs to generate answers, provided those questions can be answered by a different method (than LLMs) that is verifiable [6].
- LLMs are by design multidisciplinary engines trained across multiple specialties that can generate questions with a broad-based "extrapolation" that favors innovation.
- Answering scientific questions does not always require lab work at first: the concept of thought experiments can be traced back to ancient Greek philosophers like Plato and Aristotle [7]. Ernst Mach, a 19th-century Austrian physicist and philosopher, emphasized the importance of sensory experience and observation in scientific inquiry. Albert Einstein is often credited with popularizing and effectively using thought experiments in the field of physics.
- Today, we have an artifact that can be a substitute for thought experiments: computer simulations [8]. These simulations might be based on mathematical modeling, first principles from mechanics, physics, chemistry, etc., agent-based models, digital twins, etc. These simulations are coded with organized knowledge underpinned by algorithmic principles not necessarily available in such a structured format in LLMs.
- Simulation results can be verified and should be reproducible, which is not part of the organizational principle of LLM output [9].
- Modeling and Numerical Simulation has been recognized as one of the main tools to generate scientific discovery. Molecular dynamics and Bioinformatics are good examples. Digital twins have been the most recent development in this venture [10].

Articulating these concepts leads us to propose the following scheme to augment LLMs to support science discovery - see Figure 1.



**Figure 1: Concept**

While this goal is very ambitious, we will restrict ourselves to an example to illustrate the concept and prepare for a much broader software project that could use simulation to train LLMs under strict guidance for future science discovery in a second part of this manuscript yet to come.

### Method

Selecting the example to demonstrate the concept is critical. A good example to start this study should be an application that:

- is not trivial and/or can be tuned to an increasing level of complexity.
- has practical value.
- is understood mathematically to some level.
- gives some hint on how to generalize the concept.

The design constraints of this example with (i) to (iii) are easy to assess, while (iv) seems intrinsic to the nature of the research at best.

In this study, we have chosen to work within the field of queue modeling. Queue modeling is used extensively in virtually every aspect of our economy from manufacturing to transport, from

communication to healthcare, as well as in fundamental biology or sociology [11-14]. We will refer to our own experience with clinical workflows, which are very challenging industrial processes with complexity dominated by human factors [15-18]. To start with something simple enough, we consider a Single-Channel Queueing Model [19,20]. To illustrate our purpose, this can be a queue model of patients arriving at the clinic who need to go through consecutive phases such as registration, pre-op preparation, procedure, post-op recovery, and discharge, to take one example. The granularity of each phase can be increased at will by introducing subtasks related to staff responsibilities. To start simply, we suppose each task has an elapsed time that is given by a normal distribution.

We can write a simple simulation tool of this model with the following input parameters:

- Frequency of patient arrival
- Number of Phases
- Mean and standard deviation of the normal distribution of elapse time of task processing for each phase
- Capacity of the facility to set how many patients can be processed in parallel for each phase.

Models may vary according to the probability distribution framework used to describe the number of arrivals in an interval of time. Model parameters can be static or can change dynamically over time. The latter allows for future feedback optimization implementations or supports a study on transient behavior versus steady state. The model becomes increasingly complex if the elapsed time for each phase is explicitly stated with an agent-based model that considers human factors to achieve a digital twin description [18].

### **The Output of the Model Can be, Without Limitation,**

- Global: throughput, and efficiency,
- Patient specific: waiting time in queue,
- Focusing on a specific phase of the process:
- Elapse time per task per stage
- Hidden time between tasks at each stage

With strong hypothesis, the single channel multiple phase model can be analyzed mathematically to compute steady state output. Otherwise, one uses stochastic numerical simulation with random number generators.

As the model of our field study has been described, let us formalize what question might look like using the concept of experimental thoughts

### **Formalization of Questions**

This formulation framework was provided by L.B. Yeates as a classification of thought experiments [21]:

- Prefactual: "What will be the outcome if event X occurs?"
- Counter Factual: "What might have happened if X had happened instead of Y?"
- "Even though X occurs instead of Y, would Z still occur?"

We can substitute to X and Y "events", with X and Y parameter values or range of parameter values; X and Y can be also rules coded into subroutines of the simulation, in order to represent different probability distribution for example for task elapse time or time of arrival of patients.

- Predictive: "Can we provide forecasting from Stage Z?"
- Hind Casting: "Can we provide forecasting from stage Z with new event X?" 4 & 5 are clearly related to the stability of the simulation. Linear and Nonlinear stability metrics are well defined mathematically.
- Retrodiction: "past observations, events and data are used as evidence to infer the process(es) that produced them"
- Backcasting: "moving backwards in time, step-by-step, in as many stages as are considered necessary, from the future to the present to reveal the mechanism through which that particular specified future could be attained from the present.

The formulation of these questions with "events" is best fitted to a detailed agent-based model where events at each phase are related to staff and patients: we then need a two-level model where the task elapsed time depends on these staff/patient events using an agent-based formulation, instead of being a generic normal law. In our previous work, we used this approach to build an autonomous AI system that can companion the work of a gastroenterology clinic [18]. This mathematical framework was very useful in the context of optimizing workflow but seems too rigid to lead to unexpected innovations.

We will use Gemini first as the LLM prototype. Gemini can be free access for now and usually provides well-structured answers, which is much appreciated in science.

We started from scratch trying to let Gemini tell us more about the field of study. This is not trivial because Gemini can be easily misinterpreted if the mathematical framework we provide is not clear. This is what we meant by using the step "test LLM capabilities to describe field of study" in the graph of

Figure 1 But once we have the proper framework using the key sentence: "single-channel multiple-phase queueing system," the answers of Gemini become consistent, and we can ask Gemini to generate a finite number of scientific questions without any specification. There are multiple ways of asking Gemini to generate interesting questions. As we go and learn, we somehow ask Gemini to look at "challenging questions," questions that are often overlooked for a specific purpose such as performance or complexity, etc.

To be More Specific, we Intend to Use in our Method the "Conversation" with Gemini Summarized Below

**Question 1:** pseudo mathematical formulation: can you edit the formulation of the following problem please:

"We investigate a queue process for a workflow that corresponds to a pipeline of  $N$  steps. The queue process is fed with a large discrete number of  $n$  tasks for . The elapse time allocated for each step takes seconds where The resource allocated at each stage to process individual steps has a maximum capacity which means that no more than subtasks can be processed in parallel.

We focus on the situation where the number of tasks saturates the system, i.e. we get a slightly larger number of tasks than the capacity of the system can handle. The intent is to input the system with just enough tasks to maximize the processing units time and minimize the idle time. In principle the input flow of tasks is uniformly distributed in time, i.e. a new task comes into the pipeline with a given time frequency  $F$ .  $F$  is of the order of "

**Question 2:** What is the solution to that problem, please?

**Question 3:** Please provide ten meaningful scientific questions for a single-channel multiple-phase queueing system.

**Question 4:** Is there a mathematical model of a single-channel multiple-phase queueing system?

**Question 5:** What can go wrong in a clinical facility that can be modeled as a single-channel multiple-phase queue?

**Question 6:** We consider a single-channel multiple-phase queue model of a clinic workflow, which can be modeled with the following parameters: frequency of patient arrival ( $F$ ), number of phases in patient treatment, normal distribution parameters of processing time at each phase, capacity of the clinic for each phase, i.e., the number of patients that can be handled in parallel for each phase. Please formulate 10 interesting questions to explore the performance of the system using these parameters.

**Question 7:** Please answer these questions in the same order.

**Question 8:** Are there new challenges in understanding how single-channel multiple-phase models work?

**Question 9:** What are the questions that are often overlooked when one attempts to optimize a single-channel multiphase queueing process?"

A priori, this plan for a conversation is a mixed bag of questions and requests with respect to the type of answer expected, level of complexity, and purpose. We will report next on our results: as the answers of Gemini are indeed unexpected, we have to expand this conversation as reported in the Appendix.

## Result

LLMs do not understand the mathematics at hand and can be very easily misled depending on the formulation of the question provided in **Question 1:** one single word can bifurcate the answer, i.e., the system's answers are very unstable.

One should notice, however, that the mathematical problem is not perfectly defined. Asking Gemini to edit the problem's mathematical statement was an attempt to check that the problem is "well-posed" for Gemini. In fact, Gemini starts answering by extrapolating the content of classical textbooks in queue theory without focusing on the specificity of the model, which is a single-channel multiple-phase queueing system. Needless to say, Gemini did not answer Question 2 properly.

Once the proper scientific vocabulary tag is in the question, and the question has been simplified, the system is great at recovering basic knowledge: here, the key subject matter we should use is "single-channel multiple-phase queueing system." However, it is not clear how one can know such a key without being a specialist. But LLMs were able to provide these references after few attempts with variations of (1).

Still, Gemini does not bring properly mathematical solutions per se to answer **Question 2 & Question 7**. It can, however, easily grab some answers and references to a non-ambiguous question like **Question 4**. This is both useful and expected from a well-trained LLM, since the answers to that question have been published many times in tutorials and/or basic textbooks.

Exercising LLMs to generate questions with **Question 3 & 5 & 6, and Question 8 & 9** is by far the most exciting piece of the conversation. LLM answers to **Question 3** are somehow related to a small subset of the framework by L.B. Yeates – see A1 to A10 in the appendix – without being so formalized [21].

Not only are LLMs incredibly fast and efficient at generating questions, but half of them in our test seem relevant and can be answered by a simulation, provided the simulation code is correct.

Most amazingly, LLMs are sensitive to the formulation of questions asking for "what can go wrong?", "new challenging questions," and "overlooked topics," which is the bread and butter of scientific discovery. We can anticipate that journal publications or PR related to the problem we target would use these keywords to get the reader's attention: Gemini seems to be able to extrapolate further on such material.

The difficulty is to articulate a robust strategy to answer all these Gemini-generated questions with the simulation software without redundancy and with a good enough preprocessor that prepares the simulation input and a sound postprocessor that interprets the simulation output. We will leave for now this part that goes beyond this first essay.

But probably the most amazing development of the results is when I started to ask Gemini directly to deal with this new problem to make the concept "Formulating Thought Experiment Questions for Simulation Engines" operational. Gemini provides a generic answer that makes sense: "To effectively feed a mental experiment question into a simulation engine, we need to break down the question into its core components and translate them into a structured format that the engine can understand"; as Gemini ask if I need a follow up question, I gladly answer yes and let Gemini proposed to look at the specific example we used in that manuscript – see appendix.

## **Discussion Technology Standpoint**

To ensure the reliability of Large Language Models (LLMs) like Large Generative Language LGL), a crucial step is to implement robust mechanisms for double-checking their outputs. This approach could be inspired by healthcare practices where patient safety is paramount, aims to minimize the risk of disseminating inaccurate or inappropriate information. We have followed that model with neuromuscular diseases to lower the risk of inappropriate answers given to patients [22].

A multi-faceted approach to verification is necessary, leveraging existing tools in Natural Language Processing (NLP) and semantic analysis to identify potential inconsistencies, factual errors, and biases.

However, current LLMs, including LGL, exhibit significant limitations in rigorous mathematical reasoning, a critical capability for conducting meaningful thought experiments [23].

Overcoming this limitation and developing more robust verification systems are essential for advancing the responsible and reliable use of LLMs in various domains.

Our current method for generating good questions from an LLM involves brute-force generation, where a large number of questions are produced without any a priori discrimination. To improve this approach, we propose filtering questions based on criteria such as grammatical correctness, the use of common words, and the specificity of the question. This filtering step aims to eliminate questions that are unlikely to yield meaningful results, thus preventing the waste of valuable simulation computing time.

However, this approach raises concerns regarding energy consumption. As AI energy consumption is rapidly increasing, the additional energy required for both LLM-based question generation and subsequent scientific simulations presents a significant challenge [24].

To mitigate these concerns, we believe our approach should be applied selectively within specific fields of study. Furthermore, a robust system for tracking previously generated questions and their corresponding simulation inputs should be implemented to prevent redundant simulations and optimize resource utilization. As a matter of fact, any question would correspond to a specific set of input of the simulation, and it is relatively easy to keep those entries in a data base along with computed output for further reuse.

The rapid advancement of LLMs like Gemini necessitates a highly adaptable implementation of the concepts presented in this manuscript. A key challenge lies in automatically mapping diverse and disparate user questions to the simulation engine. To encourage innovation, we deliberately avoid rigid question formats, allowing the LLM to operate with minimal constraints. Another challenge involves automatically extracting relevant conclusions from the simulation results.

Our long-term vision is an LLM capable of performing semantic analysis of both user questions and simulation outputs, establishing the necessary connections between them, and even autonomously generating the code required to interact with the simulation software As suggested by Gemini itself – see Appendix - "Addressing these challenges requires a combination of analytical techniques, numerical methods, simulation modeling, and a deep understanding of the underlying theory concepts" of the scientific field of interest.

## **Epistemology Standpoint**

We propose a novel approach to developing large language models that transcends traditional training methods reliant solely on pre-existing datasets. Our primary focus is to leverage training to generate insightful questions within any targeted scientific domain where robust first-principles-based simulators exist. Instead of directly seeking answers from the Gemini, we advocate for its utilization to interrogate these computational science simulators. This approach offers the significant advantage of leveraging validated and verified simulation outputs.

The development of simulation software encapsulates our most refined understanding of a given scientific domain. Employing these simulations to address scientific inquiries represents a modern paradigm for advancing discovery while

minimizing the need for costly and time-consuming physical experiments. Discrepancies between simulation outputs generated by our methodology for sets of related questions should serve as catalysts for reevaluating the fundamental principles underpinning the simulations, thereby driving advancements within the respective scientific field.”

A salient feature of Large Language Models lies in their broad knowledge base, transcending the confines of any specific field or scientific discipline. The emergence of unexpected questions can be attributed to the interdisciplinary nature of LGLs and their inherent resistance to rigid preconceptions, which can be characterized as a form of “scientific reasoning incompetence [25,26].

Historically, significant scientific breakthroughs often challenge existing paradigms, necessitating a departure from conventional thinking. LGLs, trained on vast amounts of internet data, can potentially circumvent the limitations imposed by strict and narrow mathematical or computer science pre-assumed frameworks. By employing heuristic methods, these models may be able to generate unexpected questions that challenge established scientific consensus, thus fostering innovative research directions.

### Some Grand Challenges to Come

We give here three examples of simulation frameworks that Gemini may use to generate and perhaps address important questions:

- Open question in quantum physics using simulation of particle collider [<https://pythia.org/>] or [<https://herwig.hepforge.org/>]?
- What can go wrong in critical human space exploration mission using a digital twin of a mission to Mars that would be verified and validated?
- How ecosystem would be affected by various scenarios of global climate modeling using stochastic agent-based model in ecology of population?

Developing these simulation software packages requires significant effort. We anticipate that our hybrid approach, integrating Large Language Models (LLMs) with simulation engines, will unlock novel and unexpected thought experiments, thereby maximizing the potential of these powerful tools.

### Conclusion

In this essay, we have presented a novel approach to enhance Large Language Models like Gemini to facilitate scientific discovery. We particularly emphasize Gemini’s unique interdisciplinary capabilities, enabling it to generate questions that surpass the scope of any individual human researcher.

Even modest attempts to further enhance LLMs, like this one, may open wider the Pandora’s Box of “super intelligence” in a way we may regret. By establishing a robust and controlled interface between LLMs and certified simulation software, which encapsulates the collective algorithmic knowledge and expertise of generations of scientists, we may potentially maintain a degree of control over the overall system.

### References

1. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.
2. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12), 1-38.
3. National Research Council (2014). *Convergence: Facilitating Transdisciplinary Integration of Life Sciences, Physical Sciences, Engineering, and Beyond*. National Academies Press, Washington, D.C.
4. Hubert, K. F., Awa, K. N., & Zabelina, D. L. (2024). The current state of artificial intelligence generative language models is more creative than humans on divergent thinking tasks. *Scientific Reports*, 14(1), 3440.
5. Lamb, David; Easton, S. M. (1984). “Originality in art and science [chap. 9]”. *Multiple Discovery: The Pattern of Scientific Progress*. Amersham: Avebury Publishing. ISBN 978-0861270255.
6. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., ... & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2), 1-55.
7. Gendler, T. S. (2004). Thought experiments rethought—and re-perceived. *Philosophy of Science*, 71(5), 1152-1163.
8. Peter J Denning, *Computational Thinking in Science*, *American Scientist*, January-February 2017, Volume 105, Number 1, Page 13.
9. Babuska, I., & Oden, J. T. (2004). Verification and validation in computational engineering and science: basic concepts. *Computer methods in applied mechanics and engineering*, 193(36-38), 4057-4066.
10. *The Rise of Digital Twins*, *Nature Computational Science* 26 March 2024.
11. Cookson, N. A., Mather, W. H., Danino, T., Mondragón-Palomino, O., Williams, R. J., Tsimring, L. S., & Hasty, J. (2011). Queueing up for enzymatic processing: correlated signaling through coupled degradation. *Molecular systems biology*, 7(1), 561.
12. Shi, C., Jiang, Y., & Zhou, T. (2020). Queueing models of gene expression: Analytical distributions and beyond. *Biophysical Journal*, 119(8), 1606-1616.
13. Kloska, S., Pałczyński, K., Marciniak, T., Talaśka, T., Nitz, M., Wysocki, B. J., ... & Wysocki, T. A. (2021). Queueing theory model of Krebs cycle. *Bioinformatics*, 37(18), 2912-2919.

14. Furnham, A., Treglown, L., & Horne, G. (2020). The psychology of queuing.
15. Huang, A. Y., Joerger, G., Salmon, R., Dunkin, B., Sherman, V., Bass, B. L., & Garbey, M. (2016). A robust and non-obtrusive automatic event tracking system for operating room management to improve patient care. *Surgical endoscopy*, 30(8), 3638-3645.
16. Garbey, M., Joerger, G., Rambourg, J., Dunkin, B., & Bass, B. (2017). Multiscale modeling of surgical flow in a large operating room suite: Understanding the mechanism of accumulation of delays in clinical practice. *Procedia Computer Science*, 108, 1863-1872.
17. Garbey, M., Joerger, G., Furr, S., & Fikfak, V. (2020). A model of workflow in the hospital during a pandemic to assist management. *Plos one*, 15(11), e0242183.
18. Marc Garbey, Guillaume Joerger, Shannon Furr, Application of Digital Twin and Heuristic Computer Reasoning to Workflow Management: Gastroenterology Outpatient Centers Stud., *Journal of Surgery and Research*. 6 (2023): 104-129.
19. J.G.C. Templeton, G.I. Falin, *Retrial Queues*, Chapman & Hall/CRC Monographs on Statistics and Applied Probability) 1st Edition, 1997.
20. Eniola Ezekiel, *Theory and Practice of Queuing System*, Independently published, July 29, 2022 • ISBN-13. 979-8842995080.
21. Yeates, L. B. (2004). Thought experimentation: A cognitive Approach. Graduate Diploma in Arts (By Research) dissertation, University of New South Wales.
22. M.Garbey, G.Joeger and H. Kaminsky, Self-Assessment Neurological Health Care System, PCT/US24/41422.
23. Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., & Farajtabar, M. (2024). Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. arXiv preprint arXiv:2410.05229.
24. Desislavov, R., Martínez-Plumed, F., & Hernández-Orallo, J. (2023). Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38, 100857.
25. Socol, Y., Shaki, Y. Y., & Yanovskiy, M. (2019). Interests, bias, and consensus in science and regulation. *Dose-Response*, 17(2), 1559325819853669.
26. Barrio, J. R. (2009). Consensus science and the peer review. *Molecular imaging and biology*, 11(5), 293-293.

## Appendix: "Element of a Conversation with Gemini"

### Question A: Please Provide 10 Meaningful Scientific Questions for a Single Channel Multiple Phase Queue System

#### Optimal Phase Sequencing

- How should phases be sequenced to minimize overall system time or waiting time?

#### Phase Capacity and Bottlenecks

- How does varying the capacity (or service rate) of individual phases impact system performance?
- Are there specific phases that tend to become bottlenecks?

#### Customer Routing and Prioritization

- Can customer routing or prioritization schemes be implemented to improve system efficiency?
- How do different prioritization rules affect performance metrics?

#### Balking and Reneging

- How do balking (customers deciding not to join the queue) and reneging (customers leaving the queue before being served) impact system behavior?
- What factors influence the likelihood of balking and reneging?

#### Bulk Arrivals and Departures

- How does the arrival of customers in batches (bulk arrivals) or the departure of customers in groups (bulk departures) affect system performance?

#### Phase-Dependent Service Times

- If service times vary across different phases, how does this impact system behavior?
- Can phase-dependent service times be modeled effectively?

#### Non-Markovian Arrivals and Departures

- How do non-Markovian arrival and departure processes (e.g., non-exponential distributions) affect system performance?

#### Finite Buffer Systems

- If the system has a finite buffer capacity, how does this limit the number of customers that can be accommodated?

- What are the implications of buffer overflow and underflow?

### **System Stability and Transient Behavior**

- Under what conditions is the system stable?
- How does the system behave during transient periods (e.g., when there is a sudden influx of customers)?

### **Applications and Case Studies**

- In what real-world scenarios are single channel multiple phase queue systems applicable?
- Can existing case studies be analyzed to validate theoretical models and insights?

## **Question B: What can Go Wrong in a Clinical Facility that can be Model as a Single Queue Multiple Phase Queue?**

### **Overcrowding and Long Wait Times**

- **Excessive Patient Influx:** When the rate of patient arrivals exceeds the system's capacity, it can lead to long wait times and overcrowding.
- **Inadequate Staffing or Resources:** Insufficient staff or resources can contribute to delays in patient processing and increased wait times.

### **Priority Mismanagement**

- **Incorrect Prioritization:** If patients are not prioritized based on the severity of their conditions, it can result in delays for critical cases.
- **Ineffective Triage:** A flawed triage system can misclassify patients, leading to suboptimal treatment sequencing.

### **Phase-Dependent Bottlenecks**

- **Slow Phases:** If certain phases, such as imaging or laboratory tests, have long processing times, they can become bottlenecks and cause delays in the entire system.
- **Limited Resources:** Insufficient resources (e.g., equipment, personnel) in specific phases can hinder patient flow.

### **Inefficient Resource Allocation**

- **Overallocation:** Allocating excessive resources to low-priority phases can lead to underutilization of resources and increased costs.
- **Underallocation:** Insufficient resources in critical phases can result in delays and patient dissatisfaction.

### **Poor Patient Experience**

- **Frustration and Anxiety:** Long wait times and overcrowding can lead to patient frustration and anxiety.
- **Suboptimal Care:** Delays in treatment can negatively impact patient outcomes.

### **Staff Burnout**

- **Overwork:** Excessive workload and long hours can contribute to staff burnout and decreased productivity.
- **Stressful Environment:** A chaotic and stressful environment can negatively impact staff morale and job satisfaction.

## **Question C: We Consider a Queue Single Channel Multiple Phase Model of a Clinic Workflow, that can be Model with the Following Parameters: Frequency of Patient Arrival $F$ , Number of Phases in Patient Treatment, Normal Distribution Parameters of Processing Time at Each Phase, Capacity of the Clinic for Each Phase, i.e the Number of Patients that can be Handle in Parallel for Each Phase. Please Formulate 10 Interesting Questions to Explore the Performance of the System Using these Parameters**

### **General Performance Questions**

- **Average Waiting Time:** How long does the average patient wait before starting their treatment?
- **System Utilization:** What is the overall utilization of the clinic's resources (e.g., doctors, nurses)?
- **Throughput:** How many patients can the clinic process per unit time?

### **Phase-Specific Questions**

- **Bottleneck Phase:** Which phase of treatment is the most likely to cause delays or backups?
- **Phase Utilization:** What is the utilization of each individual phase?
- **Phase Waiting Time:** How long does the average patient wait at each phase?

### **Capacity-Related Questions**

- **Capacity Constraints:** Are there any phases where the current capacity is insufficient to meet demand?
- **Sensitivity Analysis:** How does the system performance change if we increase or decrease the capacity of a specific phase?

### **Arrival and Service Time Questions**

- **Arrival Rate Impact:** How does a change in the frequency of patient arrivals ( $F$ ) affect the system performance?
- **Service Time Variability:** How does the variability in processing times at each phase (standard deviation of the normal distribution) impact the system performance?

### **Question D: What are Often the Questions that are Overlooked when One Attempt Optimizing a Single Channel Multiphase Queue Process**

#### **Resource Utilization**

- Are there Idle Periods in the System? If so, can the resources be reallocated or adjusted to reduce idle time?
- Is the Bottleneck Phase Identified Correctly? Misidentifying the bottleneck can lead to ineffective optimization efforts.

#### **Customer Satisfaction**

- Are there Specific Customer Expectations or Tolerances for Waiting Times? Failing to meet these expectations can lead to customer dissatisfaction and lost business.
- Does the Queueing System Prioritize Certain Customer Types or Orders? If so, are these priorities being effectively implemented?

#### **System Flexibility**

- How Adaptable is the System to Changes in Demand or Resource Availability? A rigid system may struggle to respond to fluctuations, leading to inefficient performance.
- Can the System Handle Unexpected Events or Disruptions? A robust system should be able to recover quickly from problems.

#### **Cost Considerations**

- What is the Cost of Adding or Removing Resources? This includes the cost of hiring or firing employees, purchasing or selling equipment, and renting additional space.
- How do Changes in Queueing Policies Affect Operational Costs? For example, implementing a priority queueing system may require additional staffing or technology.

#### **Long-Term Goals**

- How does the queueing system align with the organization's overall strategic objectives? Is it supporting growth, improving customer satisfaction, or reducing costs?
- Are there potential future changes in the business environment that should be considered when optimizing the queueing system?

### **Question E: Are there New Challenge in Understanding How Single Channel Multiple Phase Model Work?**

#### **Model Complexity**

- **Multiple Phases:** As the number of phases increases, the model becomes more intricate, making it harder to analyze and predict system behavior.
- **Non-Markovian Processes:** If the time spent in each phase is not exponentially distributed (i.e., non-Markovian), the model becomes more mathematically complex and difficult to solve analytically.

#### **Real-World Variations**

- **Non-Stationary Arrivals:** In many real-world situations, customer arrivals are not constant over time. Fluctuations in demand can make it challenging to accurately model the system.
- **Batch Arrivals:** When customers arrive in groups, the model becomes more complex, as the service time for a batch may depend on the size of the batch.
- **Balking and Reneging:** Customers may decide not to join the queue (balking) or leave the queue before being served (reneging) due to long wait times or other factors. This behavior can significantly impact system performance.

#### **Data Limitations**

- **Insufficient Data:** Obtaining accurate and comprehensive data on system behavior can be difficult, especially for new or infrequently used systems.
- **Data Quality:** The quality of the data can affect the accuracy of the model. Outliers or errors in the data can lead to misleading results.

#### **Dynamic Environments**

- **Changing Conditions:** Systems often operate in dynamic environments where conditions can change over time. This can make it challenging to develop models that accurately predict future performance.
- **Uncertainty:** There is often uncertainty about future events, such as changes in demand or resource availability. This can make it difficult to incorporate these factors into the model.

## Computational Challenges

- **Computational Complexity:** Solving complex multiphase queueing models can be computationally intensive, especially when dealing with large systems or non-Markovian processes.
- **Numerical Stability:** Numerical methods used to solve these models can be sensitive to numerical errors, which can affect the accuracy of the results.

## Question F. Formulating Mental Experiment Questions for Simulation Engines

To effectively feed a mental experiment question into a simulation engine, we need to break down the question into its core components and translate them into a structured format that the engine can understand.

### Here's a General Approach:

#### Identify the Core Variables

- **Input Variables:** These are the factors that can be manipulated in the simulation.
- **Output Variables:** These are the outcomes or results we want to observe.
- **Constraints and Assumptions:** These are the limitations or conditions that define the simulation environment.

#### Define the Experiment's Goal

- What is the specific question or hypothesis we want to test?
- What kind of insights or predictions do we hope to gain?

#### Translate the Question into a Formal Query

- **Formal Language:** Use a formal language or scripting language that the simulation engine can interpret.
- **Structured Query:** Break down the question into a series of statements and conditions.

#### Example:

**Mental Experiment Question:** "What would happen if we doubled the global population and reduced natural resources by 20%?"

#### Formal Query:

```
SIMULATE {
  INPUT:
    population = 2 * current_population
    resources = 0.8 * current_resources

  OUTPUT:
    economic_growth
    environmental_impact
    social_stability

  CONSTRAINTS:
    technology_level = current_technology_level
    climate_change_model = standard_model

  RUN SIMULATION
}
```

#### Key Considerations:

- **Simulation Engine Capabilities:** The specific format and complexity of the query will depend on the capabilities of the simulation engine.
- **Data and Models:** Ensure that the simulation engine has access to the necessary data and models to accurately represent the real-world system.
- **Validation and Calibration:** Validate the simulation results against real-world data to ensure accuracy and reliability.
- **Sensitivity Analysis:** Conduct sensitivity analysis to understand how changes in input variables affect the output variables.

By following these steps and adapting them to the specific simulation engine, you can effectively formulate mental experiment questions and gain valuable insights into complex systems.

Follow Up question F: Would You Like to Delve Deeper Into a Specific Simulation Engine or Explore a Particular Mental Experiment?

## Mental Experiments Addressing Queue Single Channel Phase Model Challenges in Clinical Workflows

The Queue Single Channel Phase Model, while a foundational tool in operations research, can face limitations when

applied to complex clinical workflows. Here are a few mental experiments to address these challenges:

### **Variable Service Times**

- **Experiment:** Simulate a system where service times for each patient are drawn from a distribution (e.g., exponential, normal) rather than being fixed.
- **Question:** How does variability in service times impact waiting times, queue length, and overall system performance?
- **Goal:** Identify optimal strategies for managing variable service times, such as flexible staffing or priority queuing systems.

### **Patient Prioritization**

- **Experiment:** Implement a priority-based queuing system where patients are assigned priorities based on urgency (e.g., critical, urgent, routine).
- **Question:** How does priority-based queuing affect waiting times for different patient categories and overall system efficiency?
- **Goal:** Develop strategies for prioritizing patients effectively to minimize wait times and improve patient satisfaction.

### **Batch Service**

- **Experiment:** Model a system where multiple patients are processed together in batches (e.g., group therapy sessions, lab tests).
- **Question:** How does batch service impact system performance, particularly in terms of utilization and waiting times?
- **Goal:** Identify optimal batch sizes and scheduling strategies to maximize efficiency and minimize patient wait times.

### **Patient Abandonment**

- **Experiment:** Incorporate patient abandonment into the model, where patients may leave the queue if their wait time exceeds a certain threshold.
- **Question:** How does patient abandonment affect system performance, and what strategies can be implemented to reduce abandonment rates?
- **Goal:** Develop strategies to improve patient satisfaction and reduce lost opportunities.

### **Multiple Service Phases**

- **Experiment:** Model a system with multiple service phases (e.g., triage, consultation, treatment).
- **Question:** How does the sequence and duration of service phases impact overall system performance?
- **Goal:** Identify bottlenecks and opportunities for improvement in multi-phase workflows.

By conducting these mental experiments and analyzing the simulation results, healthcare providers can gain valuable insights into optimizing their clinical workflows, improving patient satisfaction, and reducing costs.