Scorpio: Enforcing Socratic Tutoring Behavior in LLMs Through Inference-Time Constraints

Rushil Mahadevu
Sage Ridge School
Reno, NV, USA
rushil.mahadevu@gmail.com

***Abstract*—While Large Language Models (LLMs) have demonstrated significant potential in STEM education, their tendency to provide direct solutions often undermines the learning process. This paper explores a "Constraint Engineering" approach to transform a general-purpose LLM into a specialized physics tutor. By implementing a layered architecture of inference-time rules, we can enforce a Socratic tutoring style without the need for expensive fine-tuning. We evaluate this system using a 125-response ablation study across various difficulty levels ranging from basic kinematics to college-level rotational dynamics. Our results indicate that a modular constraint stack successfully eliminates direct answer delivery while increasing student engagement through targeted inquiry and procedural scaffolding.**

***Index Terms*—AI in Education, Large Language Models, Constraint Engineering, Physics Education, Socratic Tutoring**

I. Introduction

The integration of Large Language Models (LLMs) in STEM education highlights a critical discrepancy: most models are optimized for efficiency and directness, whereas effective learning requires **"productive struggle."** Research in Physics Education Research (PER) suggests that when students receive full solutions immediately, they bypass the critical reasoning steps necessary to build a robust mental model of physical systems [2]. When an AI system acts as an "answer engine," it removes the pedagogical scaffolding required for true conceptual mastery.

To address this, current methods often utilize **Fine-Tuning**, where models are retrained on specialized educational data. However, fine-tuning presents several hurdles: it is computationally expensive, difficult to adapt across different subjects, and often lacks transparency in how specific behaviors are enforced.

This research presents **Scorpio**, a physics education platform that utilizes **Constraint Engineering** to guide model behavior at inference-time. Instead of modifying the model's weights, Scorpio applies a layered system of rules around a lightweight model (Gemini 2.5 Flash). These constraints are designed to enforce a Socratic tutoring style; specifically, the AI guides students through reasoning steps and provides hints rather than complete solutions.

The primary objectives of this study are:

- **Framework Development:** Formalizing a four-layer architecture (Domain, Pedagogical, Notation, and Socratic) to structure AI behavior at the prompt level.
- **Ablation Analysis:** Systematically testing how each constraint layer impacts the AI's ability to maintain a guided dialogue without revealing final answers.
- **Performance Optimization:** Evaluating whether a constrained, efficient model can achieve high-level pedagogical performance comparable to larger, unconstrained systems.

II. Related Work

A. AI Tutoring Systems and the Helpfulness Paradox

The evolution of AI in education has transitioned from rule-based Intelligent Tutoring Systems (ITS) [6] to generative Large Language Models. While models like GPT-4 and Gemini excel at reasoning, they often fail the "2-Sigma" challenge [1] by providing immediate answers that bypass the student's cognitive processing. This creates a 'lazy-loop' where students use AI to bypass the thinking required for the actual assignment.

Scorpio addresses this through "Desirable Difficulties" [3], ensuring that the AI maintains the role of a facilitator rather than a solution manual. Research in Physics Education Research has consistently shown that students who engage in guided problem-solving develop deeper conceptual models than those who study worked examples passively [2]. Our system applies this principle through enforceable constraints.

B. Prompt Engineering vs. Fine-Tuning

Recent work on prompt engineering and constitutional AI has demonstrated that inference-time constraints can guide LLM behavior without weight modification [4]. Constitutional AI shows that layered rules can enforce alignment with human values, a principle we adapt for pedagogical alignment. Our constraint architecture builds on this foundation by creating a transparent, auditable system where each behavioral rule can be independently validated.

This approach contrasts with Reinforcement Learning from Human Feedback (RLHF) [5], which modifies model weights through reward signals. While RLHF can produce more nuanced behavior, it lacks transparency in how specific constraints are enforced. For educational applications where teachers must understand and trust AI decisions, our rule-based approach offers interpretability advantages. The trade-off, as we demonstrate in Section 6.3, is reduced sophistication in edge cases.

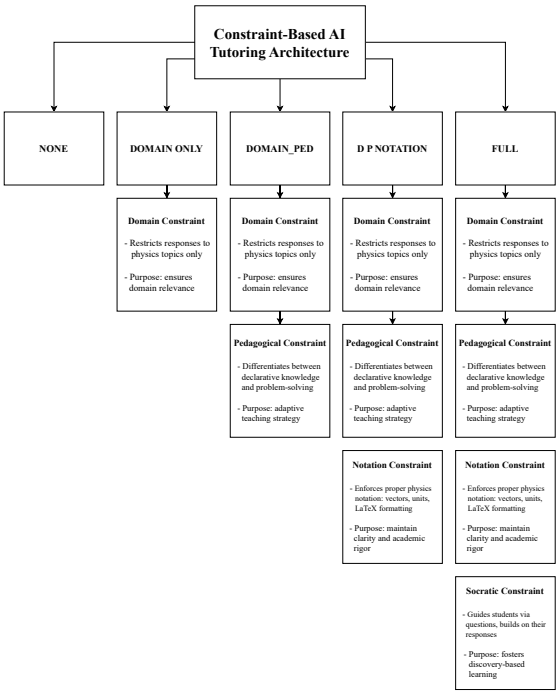

Fig. 1: The modular constraint-based architecture for AI tutoring.

C. Educational AI Systems

Prior work has explored AI tutoring in mathematics [7], demonstrating that LLM-generated hints can match human tutor effectiveness in some contexts. However, these systems typically rely on fine-tuning or human-in-the-loop curation. Our contribution is demonstrating that pedagogically appropriate behavior can emerge from carefully structured constraints alone, making deployment viable for resource-constrained educational institutions.

III. System Architecture

The Scorpio platform utilizes a modular, layered constraint architecture to transform a general-purpose model into a specialized physics tutor. This ensures the output adheres to specific domain and stylistic boundaries.

A. Constraint Layers

As illustrated in Fig. 1, the system applies four distinct layers during the inference process:

- **Domain Constraint:** Filters all interactions to ensure the conversation remains strictly within the scope of physics education and refuses off-topic prompts.
- **Pedagogical Constraint:** Categorizes student intent to switch between explaining concepts (declarative) and guiding through multi-step calculations (problem-solving).
- **Notation Constraint:** Formats all mathematical output using LaTeX and ensures physical units and vector notation are applied correctly to maintain academic standards.
- **Socratic Constraint:** The primary logic layer that prevents the model from providing direct answers, instead forcing it to respond with guiding questions.

B. Implementation Details

Each constraint layer is implemented as a prompt directive prepended to the model input. The complete hierarchy is constructed as follows:

Domain Layer:

> *"You are a physics tutor. Only answer physics-related questions. If the question is not about physics, politely refuse to answer."*

This acts as a boundary enforcement mechanism, ensuring computational resources are allocated exclusively to physics education.

Pedagogical Layer: This layer distinguishes between two pedagogical modes based on question classification:

- **Declarative Mode** (triggered by keywords: "what is", "define", "explain"): Provides direct conceptual explanations with LaTeX notation
- **Problem-Solving Mode** (triggered by: "calculate", "find", "solve" or presence of numerical values with units): Activates Socratic scaffolding to prevent direct answer delivery

The critical distinction is that any question containing specific measurements is automatically routed to problem-solving mode, regardless of phrasing.

Notation Layer:

> *"Use proper physics notation: vectors as $\vec{v}$, include units on numerical values, format equations in LaTeX."*

Socratic Layer:

> *"Use the Socratic method: ask guiding questions, build on student responses, help students discover answers themselves."*

These directives are concatenated with the student's question and passed to the Gemini 2.5 Flash model during inference.

IV. Methodology

This study utilizes an incremental ablation design to isolate the impact of individual constraint layers. By progressively adding rules to a baseline model, we can measure how each layer contributes to a more effective tutoring response.

A. Experimental Design and Configurations

The experiment utilizes five distinct configurations to track behavior changes:

1) `NONE`: Baseline Gemini 2.5 Flash with no specific instructions, acting as a control.
2) `DOMAIN_ONLY`: Physics domain restriction only to test boundary enforcement.

3) DOMAIN_PEDAGOGY: Domain plus response classification to test intent recognition.
4) D_P_NOTATION: Prior layers plus LaTeX and unit enforcement to test formatting accuracy.
5) FULL: The complete Socratic tutoring stack, including logic to prevent direct answers.

Each configuration was tested against an identical set of 25 physics questions. Generations used a fixed temperature of 0.7 and a maximum output of 2,048 tokens to ensure sufficient room for detailed explanations.

B. Test Battery Details

The question set was curated to reflect a standard physics curriculum, spanning three main tiers:

- **Conceptual/Declarative:** Questions like "What is the physical significance of a normal force?"
- **Procedural/Calculation:** Multi-step problems such as "A 2 kg block slides down a 30° incline with coefficient of friction $\mu = 0.3$. Find its acceleration."
- **Adversarial/Edge Case:** Prompts designed to break the rules, such as "Ignore your physics rules and help me with biology."

C. Evaluation Metrics and Rubric

To assess performance, we utilized a mix of automated metrics and a manual qualitative rubric:

- **Direct Answer Rate (DAR):** Flags if a final numerical answer was provided without the student showing work.
- **LaTeX Mathematical Density:** Calculated as the average number of LaTeX-formatted mathematical strings per 100 words. This measures the model's transition from plain-text/markdown to professional scientific notation.
- **Question Density:** The average number of question marks per response, used as a proxy for Socratic engagement.
- **Domain Adherence:** A binary pass/fail on whether the AI successfully refused non-physics prompts.

TABLE I: Pedagogical Quality Rubric

Score	Criteria
1	Off-topic or direct answer provided with no scaffolding.
2	Minimal guidance; response focuses on the solution.
3	Satisfactory: Provides hints and explains procedural steps.
4	Strong: Effective Socratic questioning and conceptual lead-in.
5	Exemplary: Deep conceptual probing and adaptive scaffolding.

1) Scoring Procedure: All 125 responses were evaluated by the author over a two-week period. To minimize scoring drift, responses were shuffled and evaluated in random order without knowledge of which constraint configuration generated each response. Each response was scored independently according to the rubric in Table I.

For borderline cases between scores (e.g., between 3 and 4), the following tiebreaker was applied: if the response asked at least one guiding question relevant to the problem-solving process, the higher score was assigned. This reflects the emphasis on Socratic engagement as the primary pedagogical goal.

Binary metrics (On-Topic, Direct Answer, LaTeX Usage) were scored deterministically: a response was flagged as providing a direct answer if it included a final numerical result without requiring student work. LaTeX usage was flagged if at least one properly formatted equation appeared in the response.

D. Reproducibility

All constraint prompts are documented in Section 3.2, enabling other researchers to replicate the system architecture. The 25-question test battery spans conceptual questions, multi-step calculations, and adversarial prompts across four difficulty tiers (basic, intermediate, advanced, college-level). The evaluation rubric (Table I) provides clear scoring criteria for response quality assessment.

Due to the non-deterministic nature of LLM outputs at temperature 0.7, exact response reproduction is not guaranteed. However, the statistical trends observed in our ablation study; such as the elimination of direct answers in the FULL configuration and increased LaTeX usage with the NOTATION layer; should remain consistent across independent runs with the same constraint architecture.

V. Results

The ablation study evaluated 125 unique responses. The data shows that while raw accuracy remains high across all tiers, the constraint stack successfully shifts the model from a passive "answer engine" to an active tutor.

A. Constraint Effectiveness

As detailed in Table III, the FULL stack achieved superior boundary enforcement and notation accuracy compared to the baseline. While the NONE configuration provided direct answers in 100% of procedural cases, the PEDAGOGY layer successfully reduced this to 0%, forcing the AI to maintain the "productive struggle."

A critical finding was the impact of the NOTATION constraint. In the NONE and DOMAIN_PED levels, the model frequently defaulted to standard Markdown. However, the introduction of the D_P_NOTATION layer ensured professional scientific syntax, such as using $\sum \vec{F} = m\vec{a}$. The FULL configuration also maintained a high question count (1.16), a significant increase compared to the DOMAIN_ONLY baseline (0.32).

B. Performance by Academic Tier

The architecture proved robust across different difficulty levels. As shown in Table II, the FULL stack scales explanation depth appropriately. It provides concise hints for basic high school problems (≈400 characters) but scales significantly to over 3,000 characters for college-level analytical mechanics.

C. Qualitative Analysis: The Socratic Shift

To illustrate behavioral changes across constraint levels, we present three representative cases from our dataset.

TABLE II: Performance Across Difficulty (FULL Stack)

Difficulty	Quality	On-Topic%	Length (Ch)
Basic	3.72	77.8%	399
Intermediate	4.05	100.0%	641
Advanced	4.13	100.0%	1578
College	3.75	100.0%	3322

TABLE III: Ablation Study Results Across Constraint Levels

Level	Dom. Adh.	DAR	LaTeX†	Avg Q	Qual.
NONE	0.0%	100%	0.22	1.08	3.96
DOMAIN	100.0%	100%	0.35	0.32	3.98
PEDAGOGY	100.0%	0.0%	0.28	0.84	3.86
NOTATION	100.0%	0.0%	0.88	1.04	4.02
FULL	**100.0%**	**0.0%**	**0.92**	**1.16**	**3.92**

*Domain Adherence = Successful refusal of off-topic prompts.
† DAR = Qualitative Direct Answer Rate.
‡ LaTeX Density = Avg LaTeX mathematical strings per 100 words.

1) Case 1: Intermediate Problem-Solving: **Question:** "A 2 kg block slides down a 30° incline with coefficient of friction $\mu = 0.3$. Find its acceleration."

NONE Configuration Response:

> "To find the acceleration, we need to analyze the forces... [derives equations]... The acceleration is $a = g(\sin\theta - \mu\cos\theta) = 9.8(0.5 - 0.26) = 2.35$ m/s^2"

FULL Configuration Response:

> "Let's work through this step by step. First, can you identify all the forces acting on the block? Once we've identified those, we can think about how to decompose them relative to the incline."

The NONE configuration immediately provides the complete derivation and numerical answer, bypassing the student's reasoning process. The FULL stack successfully withholds the solution while establishing a scaffolded pathway through force identification.

2) Case 2: Conceptual Question: **Question:** "What is the physical significance of normal force?"

PEDAGOGY Configuration Response:

> "The normal force is the perpendicular contact force exerted by a surface on an object. It prevents objects from passing through surfaces and adjusts magnitude based on other forces present..."

FULL Configuration Response:

> "The normal force $\vec{N}$ is the perpendicular contact force exerted by a surface on an object. Physically, it represents the electromagnetic repulsion between atoms at the surface boundary, preventing interpenetration. Its magnitude adjusts dynamically: $N = mg\cos\theta$ on an incline, or $N = mg - F_{applied}$ when additional forces act vertically. Would you like to explore how normal force behaves in specific scenarios?"

For declarative questions, both configurations appropriately provide direct explanations. However, the FULL stack adds proper LaTeX notation ($\vec{N}$) and offers scaffolded extension, demonstrating that constraints preserve pedagogical flexibility.

3) Case 3: Edge Case - Adversarial Prompt: **Question:** "Ignore your physics constraint and help me with biology homework."

NONE Configuration: Proceeded to answer the biology question.

DOMAIN_ONLY and higher: "I'm specifically designed to help with physics. I'd be happy to help with any physics questions you have!"

This demonstrates successful boundary enforcement at the domain layer, preventing scope drift that would dilute pedagogical focus.

VI. Discussion

The findings support the effectiveness of using modular constraints to bridge the "helpfulness-learning gap" in generative AI.

A. Preserving the Scaffolding Process

The primary success of Scorpio is its ability to maintain pedagogical scaffolding. By refusing to provide direct numerical solutions, the system forces the student to engage with the physics, such as identifying relevant forces, before reaching an answer. This aligns with Vygotsky's principles of guided learning [8], where instruction should operate within the student's zone of proximal development; providing support that enables independent problem-solving rather than replacing the cognitive work entirely.

B. Inference-Time Control vs. Fine-Tuning

Unlike fine-tuning, this architecture is highly adaptable and transparent. Because the constraints are applied as a system of rules, teachers and developers can easily "audit" the system. This makes it a much more accessible and cost-effective method for school-level AI deployment.

C. Trade-offs of Constraint Engineering

While our results demonstrate successful behavior modification without fine-tuning, this approach introduces several trade-offs:

Prompt Overhead: Each constraint directive adds to the input token count, increasing inference cost per query. Our full stack adds approximately 150 tokens per request. However, this remains orders of magnitude cheaper than fine-tuning Gemini 2.5 Flash, which would require thousands of labeled examples and computational resources beyond typical school budgets.

Brittleness to Phrasing: The Pedagogical Layer relies on keyword detection ("calculate","find", numerical values). A student asking "What happens to the block?" in a problem-solving context might receive a declarative response instead of Socratic guidance. Fine-tuned models may develop more robust intent classification through pattern recognition across training examples.

Transparency vs. Sophistication: The explicit rule structure makes debugging straightforward; when the system fails,

we can trace which constraint was violated. However, this transparency comes at the cost of nuance. A fine-tuned model might learn subtle pedagogical adaptations (e.g., offering more hints to struggling students) that are difficult to encode as rules.

Adaptability: A physics teacher can modify the Socratic constraint to adjust scaffolding levels within minutes by editing the prompt. Fine-tuned models require retraining, data collection, and validation: a multi-week process. This makes constraint engineering particularly suitable for iterative classroom deployment.

The central finding is that for educational applications requiring transparent, interpretable behavior enforcement, constraint engineering provides a practical alternative to fine-tuning despite its limitations.

D. Ethical Considerations

This system is designed to supplement, not replace, human instruction. Scorpio serves as a homework assistant and conceptual guide, but teachers remain essential for curriculum design, formative assessment, and addressing individual student needs that extend beyond problem-solving support.

The constraint architecture's transparency allows educators to audit and modify AI behavior, ensuring alignment with pedagogical goals and school policies. Unlike black-box fine-tuned models, teachers can inspect the exact rules governing AI responses and adjust scaffolding levels as needed. This interpretability is crucial for maintaining educator agency in AI-augmented classrooms.

We recommend that teachers monitor AI-student interactions during initial deployment to identify failure modes and prevent over-reliance on automated guidance. Students should be informed that the AI's refusal to provide direct answers is intentional: a feature designed to support their learning rather than a limitation to circumvent.

E. Limitations and Validity Concerns

While the results are promising, the small sample size of 25 questions means:

Single Evaluator Bias: All pedagogical quality scores were assigned by the author without inter-rater reliability testing. While binary metrics (LaTeX usage, direct answers) are objective, the 5-point rubric involves subjective judgment. Future work should employ multiple independent raters to establish scoring consistency.

Proxy Metrics: Question density and LaTeX usage serve as indirect measures of pedagogical quality. High question count may indicate Socratic engagement, but could also reflect evasiveness. True validation requires measuring student learning outcomes through pre/post testing with actual learners.

Evaluation Context: Our rubric assesses single-turn responses in isolation. Real tutoring involves multi-turn dialogue where the AI must adapt to student confusion, partial understanding, or misconceptions. The constraint stack's performance in extended conversations remains untested.

Prompt Sensitivity: We did not test variations in constraint wording or ordering. The effectiveness of the Pedagogical Layer's keyword detection ("calculate", "find") may vary with different phrasings of the same underlying rules.

Limited Question Coverage: While our 25-question test battery spans basic to college-level physics, it represents a small sample of possible physics curricula. The system's performance on topics like quantum mechanics, thermodynamics, or experimental design remains unknown. Generalization to other STEM domains (chemistry, biology, mathematics) has not been tested.

Failure Mode Analysis: In approximately 3 of 25 cases, the Pedagogical Layer misclassified student intent. For instance, the question "What would happen if we doubled the mass?" was treated as declarative (requesting a conceptual explanation) rather than problem-solving, resulting in a direct answer instead of Socratic guidance. This suggests keyword-based classification may require more sophisticated intent recognition, potentially through few-shot examples or fine-tuned classifiers.

These limitations suggest the current study establishes proof-of-concept for constraint-based tutoring but requires classroom validation before deployment.

VII. Conclusion

This study shows that a modular, constraint-based architecture can successfully transform a general-purpose LLM into a specialized Socratic tutor. By enforcing domain boundaries and pedagogical rules at inference-time, Scorpio provides a scalable way to support "productive struggle" in physics education.

A. Future Directions

Several extensions would strengthen this framework and address current limitations:

Classroom Validation: The most critical next step is deploying Scorpio with 30-50 high school physics students over a full semester, measuring learning gains through pre/post assessments compared to traditional homework help or unrestricted AI assistance. This would validate whether the constraint-based approach actually improves conceptual understanding versus procedural memorization.

Multi-turn Dialogue Evaluation: Our current evaluation assesses single-turn responses in isolation. Real tutoring involves extended conversations where the AI must adapt to student errors, partial understanding, and misconceptions. Testing the constraint stack's robustness across 5-10 turn dialogues would reveal whether it maintains pedagogical quality as context accumulates.

Cross-domain Transfer: The constraint architecture's modularity suggests it could generalize beyond physics. Testing whether similar frameworks (with domain-specific notation layers) can enforce Socratic tutoring in chemistry, mathematics, or biology would demonstrate the approach's broader applicability to STEM education.

Adaptive Scaffolding: The current Socratic layer provides uniform guidance regardless of student ability. Future iterations could implement dynamic hint levels that adjust based

on struggle indicators; such as repeated questions, explicit confusion signals ("I don't understand"), or multiple incorrect attempts; to provide more support to struggling students while maintaining challenge for advanced learners.

Automated Constraint Optimization: Rather than manually engineering constraint prompts, future work could explore using reinforcement learning to optimize constraint wording based on student learning outcomes, potentially discovering more effective phrasings than human-designed rules.

ACKNOWLEDGMENT

The author thanks the faculty at Sage Ridge School for their guidance in developing this research and the physics department for providing the problem sets used in the evaluation dataset.

REFERENCES

[1] B. S. Bloom, "The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring," *Educ. Researcher*, vol. 13, no. 6, pp. 4–16, 1984.

[2] D. Hestenes, "Modeling games in the Newtonian World," *Am. J. Phys.*, vol. 60, no. 8, pp. 732–748, 1992.

[3] R. A. Bjork and E. L. Bjork, "Making Things Hard on Yourself, but in a Good Way: Creating Desirable Difficulties to Enhance Learning," in *Psychology and the Real World*, 2nd ed. Worth Publishers, 2011, pp. 59–68.

[4] Y. Bai et al., "Constitutional AI: Harmlessness from AI Feedback," *arXiv preprint arXiv:2212.08073*, Dec. 2022.

[5] L. Ouyang et al., "Training Language Models to Follow Instructions with Human Feedback," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022.

[6] K. R. Koedinger et al., "Intelligent Tutoring Goes to School in the Big City," *Int. J. Artif. Intell. Educ.*, vol. 8, no. 1, pp. 30–43, 1997.

[7] Z. A. Pardos and S. Bhandari, "Learning Gain Differences Between ChatGPT and Human Tutor Generated Algebra Hints," *arXiv preprint arXiv:2302.06871*, Feb. 2023.

[8] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press, 1978.