

# Algorhythmic RAG: Socratic Process Retrieval-Augmented Reasoning for Multi-Agent Sociological Simulation

Del Coburn

University of Toronto Scarborough

`del.coburn@utoronto.ca`

## Abstract

Generative AI is reconfiguring the relationship between pedagogy, assessment, and automation in higher education. This leads us to question how learning materials are being handled by AI systems to enhance learning experiences, as opposed to simply informing generation of the correct information in response to student queries. While Retrieval-Augmented Generation (RAG) has emerged as the dominant strategy for grounding LLM outputs in verifiable resources, most implementations focus on content retrieval—fetching facts or examples. This paper proposes a novel extension of RAG, introducing Algorhythmic RAG, a framework for Process Retrieval-Augmented Reasoning (PRAR) that shifts the retrieval target from static information to dynamic pedagogical schemas.

The term "algorhythmic" intentionally combines algorithmic logic with conversational rhythm, foregrounding how pedagogical processes must be both formally specified and dialogically enacted. We operationalize this through the Reflect-Connect-Ask (RCM) model, which retrieves a structured *Process Corpus* comprising a Required Values Index (RVI) and Step-by-Step Guide—to enforce strict Socratic constraints. Unlike standard RAG, the system does not retrieve answers; it retrieves the next appropriate question and the constraints that prevent the LLM from generating student work. We demonstrate this framework through a university-level multi-agent sociological simulation course, where the system scaffolds students in operationalizing classical theory without overwriting their agency. Finally, we present a generalized *Process Corpus*

*Construction Toolkit*, showing how this architecture can be ported to domains such as clinical reasoning and legal argumentation.

## Introduction

The landscape of pedagogy is being reconstituted by generative Artificial Intelligence (AI). In addition to extending human presence through digital tools, large language models (LLMs) now automate key aspects of instructional interaction itself, raising questions about how educational systems should structure, constrain, and evaluate AI mediation in the context of learning. Recent scholarship highlights both the promise and the risks of this shift: AI can extend access, personalization, and feedback, but it also threatens to flatten pedagogical praxis into generic tutoring patterns that are detached from theory and dislocated from the social context of the classroom. This is not to sound an already well-struck alarm. Rather, we argue for a reconsideration of this technology in light of the history of modern pedagogy in the face of exponential innovation. In 1967, Texas Instruments broke centuries of mathematical tradition with the release of Cal Tech, the first hand-held calculator. In that moment, we held in our hands the entirety of a computational tradition—Pythagoras in our palms. Generative AI presents us with a contemporary reenactment of such innovation. Instead of restructuring calculation, our focus now shifts to technology that allows for the extension of reasoning across domains. In a word, AI in learning holds similar promise as innovations that came before and were absorbed by pedagogical methods. Thus, our urgency is not alarmist but stems from the recognition of potential to advance systems and strategies for learning and engaging with knowledge such as those explored by (Mattalo, 2024) and, (Hu et al., 2025).

It is with such a philosophy that this paper conceives AI as a means to a new order of pedagogy, where technology allows for discourse that is both interactive and grounded in best practices. However, utilizing AI for in depth learning requires overcoming a fundamental architectural limitation. Within computer science and STEM education, Retrieval-Augmented Generation (RAG) has emerged as a dominant strategy for tying LLM output to verifiable resources (Jacobs and Jaschke, 2024; Levonian et al., 2023; Zhou et al., 2024). Yet, most implementations focus on *content retrieval*—fetching documents, transcripts, or code snippets to answer a user’s query. In education, this often exacerbates the “oracle problem”: the AI becomes a better answer-retriever, bypassing the productive struggle essential to learning.

To address this, we introduce Algorithmic RAG, a method that shifts the retrieval target from information to pedagogical process. We call this framework *Process Retrieval-Augmented Reasoning* (PRAR). Instead of retrieving a static fact to generate an answer,

PRAR retrieves a *Process Corpus*—a structured set of pedagogical constraints and schemas that dictate the next appropriate move in a learning sequence.

The term "algorhythmic" is intentional. It combines algorithmic structure—formalized in a *Required Values Index* (RVI) and Step-by-Step Guide—with conversational rhythm. This rhythm is operationalized through the Reflect-Connect-Ask (RCM) model, a retrieval loop that forces the AI to mirror requirements, connect them to student context, and ask a single, targeted question without ever generating the solution.

This project advances Algorhythmic RAG as a generalizable framework for both single-agent and multi-agent LLM applications. As a concrete testbed, we instantiate the framework in a multi-agent sociological simulation workflow, where the system scaffolds students in operationalizing classical theory. More broadly, the framework offers a literature-grounded approach to pedagogical RAG that is: *Dialogic and Socratic*, aligned with contemporary work on dialogic pedagogy for LLMs (Beale, 2025; Hu et al., 2025; Liu et al., 2025); *Process-aware*, building on metacognitive and interaction-focused RAG for reasoning and debugging (Zhou et al., 2024; Al-Hossami and Bunesco, 2025); and *Evaluation-oriented*, informed by empirical studies of RAG trade-offs between groundedness, preference, and usability (Levonian et al., 2023; Jacobs and Jaschke, 2024).

What follows articulates the technical architecture of the Process Corpus and the RCM retrieval logic, demonstrating how Algorhythmic RAG transforms the LLM from an answer-engine into a Socratic orchestrator of student reasoning.

## Background: Dialogic Pedagogy and Socratic LLMs

Dialogic pedagogy emphasizes learning as a process of structured dialogue, conceptual conflict, and co-construction of meaning rather than one-way information transfer. Contemporary work on conversational AI explicitly argues that LLM-based tutoring must be aligned with these principles if it is to support in-depth learning rather than efficient answer-giving (Beale, 2025; Hu et al., 2025). However, operationalizing these principles requires moving beyond prompt engineering to architectural constraints.

Recent evaluations of "Socratic LLM tutors" reveal a critical gap between intent and performance. Liu et al. (2025) show that effective instructional guidance depends on three distinct capabilities: *Perception* (inferring learner state), *Orchestration* (adapting pedagogical strategies), and *Elicitation* (stimulating reflection). Their findings indicate that while models possess high domain knowledge, they often fail to "discriminate, sequence, and adapt" prompts effectively, defaulting to generic tutoring patterns or revealing answers prematurely. This suggests that the *content* of the model is sufficient, but its *process control* is deficient.

Within programming education, [Al-Hossami and Bunesu \(2025\)](#) propose "Socratic debugging" to address this, using LLMs to guide students through "reasoning trajectories" rather than providing code fixes. They demonstrate that capturing these trajectories—stepwise chains of hints, reflections, and edits—provides a richer basis for learning than assessing one-shot correctness. This concept of "trajectory-level" pedagogy is central to our proposal: effective scaffolding requires the retrieval of a sequence of moves, not just a static resource.

Mattalo’s analysis of AI in legal education further situates this challenge, arguing that AI should function not as a "solutionist" tool but as a catalyst for "pedagogical enhancement" and "breakthrough" ([Mattalo, 2024](#)). He warns that without explicit pedagogical theory, AI risks flattening education into efficient task completion.

Taken together, these works motivate a specific design imperative. To achieve the "Pedagogical Breakthrough" Mattalo describes, and to solve the "Orchestration" failures identified by Liu et al., LLM systems must retrieve and enforce pedagogical processes. They need a mechanism to impose the "reasoning trajectories" that novice learners cannot yet generate for themselves. Algorithmic RAG answers this imperative by formalizing these trajectories into a retrievable Process Corpus.

## From Content RAG to Pedagogical RAG: Addressing Dialogic Limitations

Retrieval-Augmented Generation (RAG) has become the dominant strategy for mitigating hallucination and grounding LLM outputs in curated corpora ([Lewis et al., 2020](#); [Gao et al., 2024](#)). In education, this has been productively applied to programming feedback and lecture integration. [Jacobs and Jaschke \(2024\)](#) use RAG to link programming feedback to specific lecture segments, demonstrating that students value feedback situated in course materials. Similarly, [Levonian et al. \(2023\)](#) find that RAG involves nuanced trade-offs between groundedness and preference, while [Zhou et al. \(2024\)](#) propose Metacognitive RAG to dynamically select retrieval strategies.

These strands converge on what we call *pedagogical RAG*: systems where retrieval supports learners in navigating domain structures rather than just locating facts ([Kasneci et al., 2023](#); [Jacobs and Jaschke, 2024](#)). However, as [Beale \(2025\)](#) demonstrates, content-focused RAG systems inherit three critical architectural limitations that a Process Corpus is specifically designed to address:

**Over-directness vs. Productive Struggle:** Standard LLMs are optimized for task completion, often providing answers that short-circuit the cognitive struggle essential for

learning (Beale, 2025). Content RAG exacerbates this by making correct answers even more accessible. Addressing this requires a system that retrieves negative constraints—explicit instructions on what *not* to generate—rather than just positive content. In our framework, this is formalized as the Socratic Constraint Specification, ensuring the model retrieves a refusal strategy (“Do not define the concept”) alongside the relevant pedagogy.

**Assessment Deficiency:** LLMs lack persistent mechanisms to assess learner understanding beyond immediate context. Without a structured schema, scaffolding frequently overshoots or undershoots the Zone of Proximal Development. Standard RAG retrieves content based on query similarity (what the user just said) rather than learner readiness (what the user needs to do next). To solve this, the retrieval target must shift from semantic matches to workflow states. By retrieving a Required Values Index (RVI)—a schema of necessary conceptual fields—the system can diagnose the learner’s state based on missing values rather than just text matching.

**Asymmetric Feedback Patterns:** Empirical evaluations reveal that while LLMs effectively affirm correct responses, they fail dramatically at redirecting errors, often providing vague, non-committal commentary (Liu et al., 2025). This asymmetry persists because models lack specific definitions of “wrongness” in the retrieved context. Effective remediation requires retrieving an Error Pattern Catalog—a taxonomy of common misconceptions and targeted correction schemas—so the model can recognize and address specific failures in reasoning (e.g., “conflating Marx and Tocqueville”) rather than offering generic encouragement.

Pedagogical RAG must therefore move beyond retrieving static artifacts (slides, videos, prior answers) to retrieving process constraints and state definitions. It requires a shift from an informational architecture—“What does the database say about X?”—to a procedural architecture—“Where is the student in the workflow, and what constraint applies to this step?” This architectural pivot is the foundation of Algorythmic RAG.

## Defining Algorythmic RAG

We use the term *Algorythmic RAG* to describe this process-oriented retrieval paradigm. The slightly “mis-spelled” form is intentional: it combines *algorithmic* and *rhythmic* to foreground the two distinct retrieval targets that constitute the system.

1. **Algorithmic (The Schema):** What is retrieved are algorithmic structures of practice—formalized as a *Process Corpus*. In our implementation, this consists of two primary artifacts: a *Required Values Index* (RVI), which defines the typed conceptual fields (e.g., “Concept Definition,” “Experimental Lever”) that must be specified; and a

*Step-by-Step Guide*, which encodes the if-then logic of task progression. Unlike content RAG, which retrieves answers, the algorithmic layer retrieves the *schema of the next step*.

2. **Rhythmic (The Protocol):** These algorithms are enacted as conversational rhythms through the **Reflect-Connect-Ask (RCM)** model. Each interaction follows a strictly enforced beat: *Reflect* the retrieved requirement, *Connect* it to the student’s prior inputs, and *Ask* exactly one targeted question. This rhythm is technically enforced by a “One-Question-at-a-Time” constraint in the system prompt, ensuring the LLM orchestrates inquiry rather than delivering a “wall of text” or a complete solution (Beale, 2025; Hu et al., 2025).

Algorithmic RAG, then, names a design pattern where the primary retrieval targets are *algorithms of interaction*. In a teaching context, these algorithms surface the tacit instructional design decisions usually buried in rubrics or instructors’ habits of questioning. Once formalized into the Process Corpus, they structure LLM behavior in principled ways.

**Theoretical Justification for the Neologism:** This terminological choice signals a fundamental reconceptualization of RAG from static information retrieval to dynamic process orchestration—a shift that existing vocabulary inadequately captures. Educational theory has long recognized the productive use of metaphorical language to encode complex pedagogical concepts: Vygotsky’s “scaffolding,” Bruner’s “spiral curriculum,” and Freire’s “banking model” all use vivid imagery to crystallize theoretical insights. Similarly, “algorithmic” encapsulates the dual requirements of formal specification (the algorithm) and adaptive enactment (the rhythm) that distinguish pedagogical process retrieval from traditional RAG paradigms. This reorientation prepares the ground for Process Retrieval-Augmented Reasoning (PRAR), in which LLMs reason not over retrieved text, but over retrieved constraints. In PRAR, the model’s core task is to maintain the rhythm of the process: to determine which step of the guide to retrieve, which value in the index is missing, and which Socratic constraint must be applied to the next generation.

## Algorithmic RAG and Process Retrieval-Augmented Reasoning (PRAR)

Algorithmic RAG is a pattern for organizing LLM applications around formalized processes rather than only topical documents. A *process* here is not merely a sequence of actions, but a retrievable state machine of pedagogical moves—for example: “Elicit student concept,” “Surface theoretical tension,” “Prompt operationalization,” and “Run interaction.”

*Process Retrieval-Augmented Reasoning* (PRAR) operationalizes this pattern by generalizing the RAG mechanism. Instead of retrieving knowledge chunks (texts, transcripts, code), the system retrieves *process schemas*: algorithmic templates, constraint specifications, and decision trees. The LLM reasons within this retrieved structure, where each generation is bounded by procedural patterns and required values (e.g., explicit hypothesis, concept tensions, predicted outcomes).

Where Metacognitive RAG treats retrieval *policy* as an object of metacognition—deciding how to search (Zhou et al., 2024)—PRAR extends this to *pedagogical metacognition*. The system effectively asks: "Where is the learner in the workflow? Which step must be retrieved next? And what constraints apply to this specific transition?"

This logic is instantiated through three core artifacts that collectively function as the *Process Corpus*:

1. **The Required Values Index (RVI)**: A schema that enumerates the minimal conceptual fields students must specify (e.g., [Concept A], [Experimental Lever], [Agent Goal]). It acts as the "database schema" for the student's developing project.
2. **The Step-by-Step Guide**: A linear logic file that encodes the phases of the project (Conceptualization → Experiment Design → Drafting). It links each step to specific RVI fields, defining the criteria for advancement.
3. **The Socratic System Prompt**: A "controller" script that enforces non-creative, format-only behavior. It ensures the LLM never generates student content but constantly retrieves the next empty RVI field and "reflects" it back to the learner via the RCM protocol.

In this architecture, the Process Corpus replaces the textbook as the primary retrieval target. Instead of embedding lecture notes to generate answers, the system embeds the course's *algorithmic pedagogy*—a structured logic of what students must do to transform abstract theory into concrete experimental design.

## PRAR and the Perception-Orchestration-Elicitation Framework

Recent work on Socratic LLM evaluation (Liu et al., 2025) identifies three critical dimensions of instructional guidance that align precisely with PRAR's objectives: *Perception* (inferring learner state), *Orchestration* (adapting pedagogical strategies), and *Elicitation* (stimulating proper reflection). This framework emerged from empirical findings that while LLMs excel at answering queries, they often fail to provide effective adaptive scaffolding when learners



experience confusion. We operationalize each dimension through specific process-retrieval mechanisms:

**Perception via Schema Validation:** The system uses the Required Values Index (RVI) as a diagnostic schema to infer learner state. Rather than relying on generic "understanding checks," the system validates student inputs against typed fields (e.g., [Concept Definition], [Experimental Lever]). This addresses the "limited sensitivity to implicit knowledge states" identified by Liu et al. (2025). For example, if a student proposes "inequality" for [Concept A], the system does not merely ask for clarification. It detects a validation failure—specifically, the "Generic Terminology" pattern defined in the Error Pattern Catalog—and retrieves a targeted diagnostic sequence: (1) reflect the vague term, (2) contrast it with the theorist’s specific vocabulary, and (3) prompt for the precise textual definition.

**Orchestration via Workflow Logic:** The Step-by-Step Guide functions as a state machine for pedagogical orchestration. It encodes adaptive branching logic calibrated to learner progress. When a student demonstrates confusion about complex design elements (an implicit state of cognitive overload), the system retrieves simplification strategies—breaking a multi-agent task into single-agent components. Conversely, when a student proposes a weak hypothesis, it retrieves "constraint reminders" that reconnect the design to primary theoretical texts. This directly addresses Liu et al. (2025)’s finding that LLMs lack "flexibility to implicit states." By formalizing common error patterns into the Process Corpus, PRAR enables consistent orchestration adaptation (OSA), shifting strategies based on the structural quality of student work rather than just explicit requests for help.

**Elicitation via the RCM Protocol:** The System Prompt enforces the Reflect-Connect-Ask (RCM) method to govern elicitation depth. This operationalizes what Liu et al. (2025) distinguish as "heuristic questioning" (for confused states) and "strategic questioning" (for comprehension states). Crucially, the system is bound by a strict "One-Question-at-a-Time" rule. For instance, when a learner is in a positive state (having successfully defined [Concept A]), the retrieved process triggers a strategic question: "How would this concept manifest differently in [Fictional Setting]?" When in a negative state (confusion about operationalization), it retrieves a heuristic prompt: "Think about a concrete example from the readings—what evidence did the theorist point to?"

Empirical evaluation suggests that even advanced models like GPT-4 exhibit substantial gaps in these dimensions, particularly in redirecting errors (P-Redirect) and adapting question depth (ESA). Algorithmic RAG addresses this by externalizing pedagogical intelligence into the Process Corpus, ensuring that Perception, Orchestration, and Elicitation are treated as retrievable architectural constraints rather than emergent properties of the



language model.

## PRAR as Metacognitive Process Orchestration

Where Metacognitive RAG treats retrieval *policy* as an object of metacognition—dynamically choosing among search strategies like BM25 or dense retrieval (Zhou et al., 2024)—PRAR extends this to *pedagogical metacognition*. In this framework, the system’s metacognitive task is to determine *which process step* must be executed next, based on the learner’s position within the Step-by-Step Guide and the completion status of the Required Values Index.

Consider a scenario from the sociological simulation workflow: if a learner has successfully defined [Concept A] and [Concept B] but struggles to articulate a measurable [Agent Goal], the system does not simply retrieve informational content about ”goals.” Instead, it metacognitively assesses the learner’s state—identifying a ”failure to operationalize”—and retrieves a specific three-step scaffolding process:

1. A *connection prompt* that forces a link between the abstract concept and the agent’s function: ”How would this agent’s objectives reflect the tension between [Concept A] and [Concept B]? What would success or failure look like from the perspective of your theory?”
2. A *constraint reminder* retrieved from the rubric constraints: ”Agent goals must be measurable and tied to observable outcomes. What specific action would indicate progress toward this goal?”
3. An *exemplar structure* (not content) from the process corpus, demonstrating the required syntax without providing the answer: ”[Agent Goal] should follow this pattern: '[Agent Name]’s goal is to [measurable outcome] which would demonstrate [theoretical concept] by [observable behavior].’”

This retrieval sequence is itself a metacognitively selected *process schema*—the ”operationalization scaffolding” pattern—rather than a fixed response. If the learner instead demonstrates strong conceptual grounding but weak methodological specificity (a different state), the system retrieves a different process: ”Challenge Questions” that probe the validity of the experimental lever.

In this sense, PRAR generalizes Zhou et al. (2024)’s insight. The system maintains awareness of multiple process schemas in the corpus (conceptual connection, methodological specification, theoretical grounding, error correction) and selects which to instantiate based on continuous assessment of the learner’s trajectory. This dual-level metacognition—managing

both the retrieval of constraints and the progression of pedagogy—represents the core innovation of the PRAR architecture.

## Socratic-RCM for Multi-Agent Sociological Simulation

We instantiate Algorithmic RAG through the Socratic Reflect-Connect Model (RCM), a protocol that governs the conversational rhythm of the system. This model ensures that every system output advances the learner’s reasoning without crossing the boundary into content generation:

1. **Reflect:** The system initiates every turn by mirroring the relevant requirement from the Required Values Index. Instead of asking generic questions (“What do you want to do?”), it reflects the constraints of the current step (e.g., “The assignment requires a definition of [Concept A] that cites a primary text”). This aligns with dialogic recommendations that LLMs should ground inquiry in shared norms rather than open-ended generation (Beale, 2025; Mattalo, 2024).
2. **Connect:** Once the requirement is established, the system connects it to the learner’s specific context—their chosen theorist, setting, or previous inputs. This “process grounding” ensures that scaffolding is not generic but situated in the learner’s unique project trajectory (Hu et al., 2025).
3. **Constrain:** Perhaps most critically, the system retrieves explicit negative constraints from the Socratic Constraint Specification. It is strictly prohibited from generating theoretical content, agent descriptions, or scenario narratives. It checks whether the student has filled the required placeholders; if not, it refuses to proceed, enforcing the “productive struggle” often bypassed by standard LLMs (Liu et al., 2025).
4. **Simulate and Analyze:** The resulting artifacts—student-authored agent prompts and round structures—are then fed into the multi-agent simulation platform. The simulation functions not as a “chat” but as an experimental apparatus. Students run their design and interpret the resulting transcripts as empirical data, paralleling how RAG-based lecture systems link feedback to concrete artifacts (Jacobs and Jaschke, 2024), but here extended to interactional data generated by the students’ own theoretical models.

This design treats the AI not as a knowledge oracle, but as a *process orchestrator*. In Algorithmic RAG terms, the system is a Socratic PRAR controller: its function is to retrieve

the next required process step (e.g., "Step 1.2: Define Tension") and elicit the student’s fulfillment of that step, ensuring that the final simulation is entirely the product of the learner’s reasoning.

## Relation to Existing Pedagogical RAG Work

Algorithmic RAG and PRAR build on, yet architecturally diverge from, existing RAG-based educational systems. Like lecture-linked feedback systems (Jacobs and Jaschke, 2024), Algorithmic RAG integrates structured corpora—but here the corpus serves as a repository of *process logic* (Required Values, stepwise guides) rather than content artifacts (lecture transcripts). Like Metacognitive RAG (Zhou et al., 2024), PRAR foregrounds dynamic control of retrieval, but it shifts the object of metacognition from *retrieval strategies* (e.g., selecting vector search vs. keyword search) to *pedagogical steps* (e.g., selecting concept definition vs. tension articulation). Like the Socratic Playground (Hu et al., 2025) and recent Socratic LLM studies (Liu et al., 2025), the framework emphasizes multi-turn, question-driven scaffolding. However, PRAR applies this specifically to the design of simulation experiments, embedding theory-method links directly into the retrieved workflow. In this sense, Algorithmic RAG can be understood as a *domain-specialized* pedagogical RAG: it encodes disciplinary concepts (e.g., "tyranny of the majority"), theoretical tensions, and experimental levers into a reusable Process Corpus that LLMs can retrieve and enforce as structural constraints.

## Relation to Multi-Modal Pedagogical Systems

The Socratic Playground (Hu et al., 2025) represents the state-of-the-art in multi-modal pedagogical AI, offering distinct interaction modes and Expectation-Misconception-Tailored (EMT) scaffolding. Like Algorithmic RAG, it retrieves pedagogical processes, but it operates at a different granularity and enforces fundamentally different constraints.

### Architectural Similarities:

- **Process-Level Organization:** Both systems move beyond content retrieval to retrieve structured pedagogical sequences. Where the Socratic Playground retrieves EMT feedback patterns (hint → prompt → assertion) calibrated to response quality, Algorithmic RAG retrieves RCM sequences (reflect → connect → ask) calibrated to the completion of the Required Values Index (RVI).
- **State-Aware Adaptation:** Both systems assess learner state to modulate system behavior. The Playground uses Learner Characteristics Curves (LCC) to decompose

responses into relevant/irrelevant components. PRAR uses the RVI as a diagnostic schema to detect "implicit state" errors—such as checking if a defined [Experimental Lever] logically manipulates the stated [Concept A].

- **Metacognitive Scaffolding:** Both emphasize metacognition through reflection. The Playground’s Teachable Agent mode asks students to explain concepts to a peer; PRAR’s constraint-enforcement approach requires students to articulate why their design choices align with theoretical principles before the system allows them to proceed.

### Critical Distinctions:

**Scaffolding vs. The Non-Generation Principle:** The Socratic Playground scaffolds learning by *generating* tailored hints and explanations. When a student struggles with a concept, the system provides analogies or guided questions to smooth the path. Algorythmic RAG, by contrast, utilizes a Socratic Constraint Specification to retrieve *negative constraints* that prevent generation entirely. When a learner struggles to define a concept, the system does not provide paraphrases or examples; it retrieves a "refusal-and-redirect" process that forces the student to return to primary texts. This prioritizes the "productive struggle" of articulation over the efficiency of understanding.

**Adaptive Modes vs. Enforced Process:** The Playground adapts across five modes (e.g., Tutoring vs. Gaming) to match learner readiness, maximizing accessibility. Algorythmic RAG, conversely, enforces a single, invariant process structure (Conceptualization → Drafting → Review) via the Step-by-Step Guide. While questioning tactics adapt within this structure, the workflow itself is rigid. This rigidity serves a specific pedagogical purpose: ensuring all students engage with the complete "algorithmic" logic of theoretical operationalization—a higher-order competency that cannot be bypassed.

**Complementarity:** These systems address different layers of the pedagogical challenge space. The Socratic Playground excels at guiding learners through *content mastery* via adaptive scaffolding—helping students understand established facts through iterative explanation. Algorythmic RAG excels at guiding learners through the *creative application of frameworks*—helping students design original models or arguments. Future research might productively combine these approaches: using Playground-style scaffolding for foundational knowledge building, then transitioning to PRAR-style constraint enforcement for application and synthesis tasks.

## Methods: Constructing the Process Corpus

The implementation of Algorithmic RAG is grounded in a deliberately constructed Process Corpus—a set of structured text files that serve as the retrieval target for the LLM. Rather than treating prompts as ad hoc engineering artifacts, these materials formalize the sociological method into modular, retrievable components.

### The Schema: Required Values Index (RVI)

The Required Values Index (RVI) functions as the database schema for the student’s project. It systematically enumerates the conceptual and procedural fields that must be “filled” to constitute a valid design. These fields include:

- **Theoretical Fields:** Primary concepts (e.g., [Concept A]), canonical sources, and the focal tension to be investigated.
- **Design Fields:** Agent identifiers, roles, goal structures, and the experimental lever (the variable being manipulated).
- **Constraint Fields:** Definitions of “baseline” vs. “experimental” conditions.

Each value is specified with a definition and local constraints (e.g., “[Agent Goal] must be measurable and tied to [Concept A]”). The RVI allows the system to perform “state detection” by simply checking which fields remain null.

### The State Machine: Step-by-Step Guide

A separate Step-by-Step Guide encodes the workflow logic. It articulates the phases of work—from high-level conceptualization to detailed agent drafting and final analysis. Crucially, each step is explicitly linked to entries in the RVI, creating a dependency graph (e.g., “Cannot proceed to [Agent Prompts] until [Concept A] is defined”). This document provides the “algorithmic backbone,” determining the sequence of retrieval.

### The Controller: TA System Prompt and Socratic Constraints

The System Prompt functions as the runtime controller, encoding the Socratic Constraint Specification as hard behavioral rules:

- **The Non-Generation Principle:** The agent is strictly forbidden from generating theoretical content, agent descriptions, or narratives. It retrieves these only as structural templates (e.g., "Use this format: [Role]... [Goal]...") but never fills the content.
- **The One-Question-at-a-Time Rule:** To enforce conversational rhythm, the agent is constrained to ask exactly one question per turn, preventing cognitive overload.
- **Refusal and Redirect:** When a student offers vague acceptance (e.g., "sure"), the agent retrieves a refusal pattern that demands explicit wording tied to a specific RVI field.

This prompt operationalizes the Reflect-Connect-Ask rhythm: reflect the retrieved requirement, connect it to the student’s current value, and ask for the next move.

## The Experimental Layer: Multi-Agent Simulation

Finally, the simulation platform uses these same process artifacts as scaffolds for the experiment itself. Agents (e.g., Worker, Manager, Moderator) are defined entirely by the student-authored prompts that resulted from the PRAR dialogue. In this way, the methods of classical sociological theory—concept formation, tension articulation, and hypothesis testing—are encoded as a reusable Process Corpus that structures both the design and analysis of simulations.

## Generalizability Beyond the Reference Implementation

Although the present implementation is grounded in sociological theory, the core architecture of Algorithmic RAG is platform- and domain-independent. To demonstrate portability, we map PRAR’s core components to four distinct pedagogical domains (Table 1), showing how process retrieval can be adapted while maintaining the invariant PRAR logic.

The key insight from Table 1 is that Algorithmic RAG’s portability depends not on domain-neutral algorithms but on the *formalizability* of disciplinary expertise into retrievable process structures. Three factors predict successful PRAR implementation:

1. **Established Pedagogical Scaffolding:** Domains with well-codified instructional sequences (like the medical diagnostic protocol or legal IRAC) are particularly amenable to Process Corpus construction.

Domain	Process Corpus	Required Values	Socratic Constraint
<b>Clinical Reasoning</b>	Diagnostic decision trees, differential diagnosis protocols	[Chief Complaint], [Diff. Diagnoses], [Diagnostic Tests], [Evidence Synthesis], [Treatment Rationale]	Never provides diagnoses; guides through systematic hypothesis generation and evidence evaluation.
<b>Legal Argumentation</b>	IRAC framework (Issue-Rule-Application-Conclusion)	[Legal Issue], [Applicable Rules], [Fact Pattern], [Rule Application], [Counter-Arguments], [Conclusion]	Never drafts arguments; elicits rule identification, fact-to-law mapping, and reasoning through ambiguities.
<b>Experimental Design</b>	Hypothesis formation protocols, variable isolation principles	[Research Question], [Hypothesis], [Independent Variables], [Controls], [Predicted Outcomes]	Never proposes hypotheses; guides from observation to testable prediction, probes confound identification.
<b>Narrative Development</b>	Story structure frameworks (e.g., Hero’s Journey), character arcs	[Protagonist Goal], [Central Conflict], [Inciting Incident], [Climax], [Thematic Question]	Never generates plot points or dialogue; probes narrative coherence and character motivation consistency.

Table 1: Algorithmic RAG adaptation across pedagogical domains. Each domain requires domain-specific process corpora and required values, but the core PRAR architecture (retrieve process → constrain generation → elicit student reasoning) remains invariant.

2. **Process-Content Distinction:** Effective implementation requires separating *how to think* (process) from *what to think* (content). In clinical reasoning, the diagnostic process can be retrieved independently of specific disease content.
3. **Assessable Intermediate Artifacts:** PRAR works best when reasoning can be captured in discrete, evaluable fields (Required Values) rather than only in final products. Medical students can specify [Differential Diagnoses] before a final diagnosis; law students can identify [Applicable Rules] before drafting arguments.



# Limitations and Challenges

This work opens several critical areas for future investigation, particularly regarding the tension between algorithmic constraint and pedagogical freedom.

## Learner Agency and the Risk of Over-Scripting

First, there is an inherent tension between scaffolding and agency. While Algorithmic RAG aims to preserve student autonomy by refusing to generate content, it limits the structural choices available to the learner. By imposing a rigid Step-by-Step Guide, the system runs the risk of creating "pseudo-agency," where students nominally control the content but are subtly herded down pre-determined pathways (Mattalo, 2024). Future implementations must explore adaptive constraint relaxation, allowing the system to loosen process bounds as the learner demonstrates expertise.

## Empirically Observed Limitations in Process Enforcement

Beyond theoretical concerns, early deployment of Algorithmic RAG reveals concrete challenges aligned with recent empirical research on Socratic LLMs (Liu et al., 2025). Systematic evaluation identifies consistent failure patterns that PRAR implementations must actively mitigate:

**Asymmetric Feedback (P-Affirm vs. P-Redirect):** Liu et al. (2025) find that LLMs achieve high accuracy ( $> 0.85$ ) in affirming correct responses but fail dramatically (often  $< 0.55$ ) at redirecting errors. In pilot testing, our system exhibited this asymmetry: strong definitions were immediately affirmed, but weak definitions elicited generic requests for "more detail" rather than targeted theoretical probes. **Mitigation Strategy:** We address this by enhancing the Error Pattern Catalog with specific "anti-patterns" (e.g., conflating distinct theoretical terms). This transforms the redirection task from generic prompting to pattern-matched retrieval, ensuring the system can "diagnose" a vague definition as effectively as it validates a correct one.

**Implicit State Blindness (Orchestration Failures):** The system responds effectively to explicit confusion but struggles with *implicit* contradictions—for instance, a student proposing a "Marxian" experimental lever for a "Tocquevillian" problem. Standard RAG validates the individual fields without detecting the logical misalignment between them. **Mitigation Strategy:** This requires implementing consistency diagnostics within the Required Values Index. By defining dependency rules (e.g., "IF [Theory] = B THEN [Lever] MUST NOT = Class Conflict"), the system can proactively flag theoretical incoherence even

when individual fields are linguistically valid.

**Question Depth Homogeneity (Elicitation Failures):** Liu et al. (2025) measure Elicitation Strategy Adaptivity (ESA)—the degree to which questioning depth varies across learner states. Early implementations showed low ESA ( $< 0.15$ ): questions maintained similar complexity whether students demonstrated mastery or confusion. **Mitigation Strategy:** We enrich the Step-by-Step Guide with graduated questioning templates. "Heuristic" questions (for confused states) and "Strategic" questions (for advanced states) are explicitly separated in the retrieval corpus, making ESA a function of retrieval logic rather than emergent model behavior.

## Evaluation at the Process Level

Third, rigorous evaluation remains an open challenge. Existing studies typically focus on answer quality or user preference (Levonian et al., 2023; Zhou et al., 2024). By contrast, Algorithmic RAG requires process-level metrics to evaluate the efficacy of the scaffolding itself:

- **Constraint Adherence Rate:** The frequency with which the TA-agent successfully refuses to generate content (e.g., zero violations of the "Non-Generation Principle").
- **Trajectory Completeness:** The percentage of students who successfully populate all fields in the Required Values Index without bypassing steps.
- **Integration Depth:** A qualitative measure of how effectively students link retrieved concepts (e.g., [Concept A]) to operational choices (e.g., [Agent Goal]) in their final transcripts.

Validating these metrics will require mixed-methods studies combining automated log analysis with rubric-based assessment of student artifacts.

## Generalization Across Domains and Platforms

Finally, while the framework is intended to be generalizable, its reference instantiation is grounded in classical sociological theory. Extending Algorithmic RAG to domains such as law, medicine, or policy will require co-designing new Process Corpora with domain experts (Mattalo, 2024). Future work includes: (a) Cross-course replication, testing whether the "RVI + Guide" architecture can scaffold different forms of inquiry; (b) Multi-agent PRAR, where distinct LLM agents (e.g., a "Tutor" vs. a "Critic") share a common Process Corpus but occupy different dialogic roles; and (c) Metacognitive Integration, giving the system explicit

self-monitoring capabilities to tighten or relax constraints based on real-time assessment of learner confidence (Zhou et al., 2024).

## Conclusion

The Algorithmic RAG framework and its Socratic PRAR instantiation contribute to the emerging field of pedagogical AI by shifting attention from content retrieval to process retrieval, and from answer-generation to process orchestration. Grounded in contemporary work on dialogic pedagogy, Socratic LLMs, and metacognitive RAG, the framework demonstrates how theoretical reasoning can be operationalized into simulation design without letting the AI overwrite student agency (Mattalo, 2024; Beale, 2025; Hu et al., 2025; Zhou et al., 2024; Liu et al., 2025). As educational institutions increasingly adopt generative AI, the challenge is not simply to "add RAG" to existing systems, but to encode pedagogy itself as a retrievable, enforceable process. Algorithmic RAG offers a concrete, rigorously defined path toward that goal.

## References

- Al-Hossami, E., & Bunesu, R. (2025). Reasoning trajectories for Socratic debugging of student code: From misconceptions to contradictions and updated beliefs. *arXiv preprint arXiv:2511.00371*.
- Beale, R. (2025). Dialogic pedagogy for large language models: Aligning conversational AI with proven theories of learning. *arXiv preprint arXiv:2506.19484*.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.
- Henkel, O., Levonian, Z., Postle, M. E., & Li, C. (2024). Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference. In *Proceedings of the 17th International Conference on Educational Data Mining (EDM 2024)* (pp. 315–320). Atlanta, GA: International Educational Data Mining Society.
- Holstein, K., McLaren, B. M., & Aleven, V. (2017). Intelligent tutors as teachers' aides: Exploring teacher needs for real-time analytics in blended classrooms. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference (LAK 2017)* (pp. 257–266). New York: ACM.

- Hu, X., Xu, S., Tong, R., & Graesser, A. (2025). Generative AI in education: From foundational insights to the Socratic playground for learning. *arXiv preprint arXiv:2501.06682*.
- Jacobs, S., & Jaschke, S. (2024). Leveraging lecture content for improved feedback: Explorations with GPT-4 and retrieval augmented generation. *arXiv preprint arXiv:2405.06681*.
- Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Stadler, M., Weller, J., Kuhn, J., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274.
- Levonian, Z., Li, C., Zhu, W., Gade, A., Henkel, O., Postle, M. E., & Xing, W. (2023). Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference. *arXiv preprint arXiv:2310.03184*.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459–9474).
- Liu, Y., Li, C., Zhang, T., Wang, M., Zhu, Q., Li, J., & Huang, H. (2025). Discerning minds or generic tutors? Evaluating instructional guidance capabilities in Socratic LLMs. *arXiv preprint arXiv:2508.06583*.
- Mattalo, B. (2024). Artificial intelligence: The future of pedagogy. *Journal of Legal Studies Education*, 41(1), 49–71.
- Zhou, Y., Liu, Z., Jin, J., Nie, J. Y., & Dou, Z. (2024). Metacognitive retrieval-augmented large language models. In *Proceedings of the ACM Web Conference 2024 (WWW '24)* (pp. 1453–1463). New York: ACM.