2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #53325

WIP: A Virtual Teaching Assistant to Engage in Effective Pedagogical Methods

Dr. Jonathan Steffens, Washington State University

Jonathan received a B.S. in Mechanical Engineering from Rochester Institute of Technology and a Ph.D. in Mechanical Engineering from Cornell University. He worked at the U.S. Environmental Protection Agency as a post-doctoral fellow. He was a visiting professor at Lafayette College. Currently, he is an Assistant Professor in the Mechanical and Materials Engineering Department at Washington State University.

Prof. Narasimha Boddeti
Yan Yan
Zijian Zhang, Washington State University

WIP: A Virtual Teaching Assistant to Engage in Effective Pedagogical Methods

Abstract

This work presents the development and behavioral evaluation of an AI agent acting as a Virtual Teaching Assistant (VTA) for large STEM courses. The VTA is designed to engage students using Socratic dialogue and inquiry-based learning rather than direct-answer responses. The project emphasizes scalable, open-source technology that addresses challenges posed by large class sizes and limited instructor availability.

The VTA follows a multi-agent design. Its core Socratic Dialogue agent is built on Qwen2.5-7B-Instruct [1], an open-weight base model, fine-tuned with Low-Rank Adaptation (LoRA) on a domain-specific Socratic dialogue dataset that we generated by adapting the SocraticLM framework [2] to university-level thermo-fluid sciences. A planned Scaffolder agent will handle orchestration in subsequent work, including retrieval-augmented generation (RAG) over course-specific content, dialogue routing, and progress tracking. The system can be embedded in learning management systems such as Canvas or Blackboard.

Planned features include anonymized interaction logging, on-demand summary reports for students, and aggregated reports for instructors that surface gaps in understanding across the class.

In a preliminary held-out comparison, the LoRA-trained checkpoint produced Socratic guidance in 4/4 cases while a P-Tuning v2 checkpoint trained on the same data reverted to direct problem-solving in 4/4 cases. Planned future work will pilot the VTA in multiple course sections and conduct mixed-methods evaluation of pedagogical effectiveness, learning outcomes, and student perceptions relative to traditional human-TA support.

1. Introduction

Generative artificial intelligence (AI) is increasingly used in education, offering capabilities for content development and interactive learning [3], [4]. In rigorous STEM disciplines, personalized support is critical due to the abstract nature of problem-solving. However, as class sizes grow and student-to-faculty ratios worsen [5], [6], providing individualized attention becomes increasingly challenging. While AI-based tutors are becoming more common [7], their potential often remains underutilized, frequently failing to leverage recent compact open-weight language models such as Llama 3.2 [8], Mistral [9], or Phi-4 [10].

The default “answer-first” behavior of most generative AI systems can conflict with instructional goals in engineering education, where the objective is to develop problem-solving skills, metacognition, and persistence. To understand the trajectory of these tools, we look to the history of computer-based education, which has evolved from early Computer Assisted Instruction (CAI) programs like SCHOLAR in the 1970s [11] to the second generation of Intelligent Tutoring Systems (ITS) that boosted student performance by approximately one standard

deviation [12], [13]. Despite these gains, ITS have faced high authoring-time costs, estimated at 50 to 300 hours per hour of instruction, and have yet to consistently match the “two-sigma” boost attributed to one-to-one human tutoring [14].

To address these limitations, commercial platforms and independent developers have introduced specialized study modes, such as those in Google’s Gemini [15] and OpenAI’s ChatGPT [16], which use guiding questions and step-by-step scaffolding rather than answer-first responses. These products share important pedagogical goals with the present work; the differentiation here is technical and institutional rather than pedagogical-philosophical: domain-specific reproducible fine-tuning on thermo-fluid problems, weight-level access for self-hosting on institutional or personal hardware, and explicit behavioral evaluation of Socratic adherence rather than reliance on undisclosed alignment procedures. Open-source initiatives such as DeepTutor [17] represent early efforts toward framework-level tutoring systems with these properties.

We frame current LLM-based tutors as a potential third generation of computer tutoring systems, driven by transformer architectures and Large Language Models (LLMs) that can dialogue with learners in a constructive manner [7]. This fits into the interactive learning paradigm defined by the ICAP framework, which posits that interactive engagement is more effective than constructive, active, or passive modes [18]. This project prioritizes an open-source imperative for such systems, differentiating itself from general-purpose assistants in four ways:

- **Open-Weight Architecture:** Unlike proprietary models, our VTA uses open-weight models to enable data privacy through institutional self-hosting, eliminate per-token costs, and allow weight-level inspection.
- **Small Language Model (SLM) Optimization:** Rather than relying on massive, all-in-one platforms, we focus on fine-tuning compact SLMs (1B to 8B parameters) that can be hosted locally on institutional or personal hardware. Systematic evaluation of which model sizes meet pedagogical requirements at acceptable compute cost is itself a planned research output (Section 3.3).
- **Course-Specific Specialization:** The VTA is designed as a specialized tool tailored to specific engineering curricula, rather than a generalist model that may hallucinate outside of its training distribution.
- **Multi-Agent / Agentic Architecture:** Rather than treating the VTA as a single monolithic LLM endpoint, the system is designed as a multi-agent pipeline in which a Scaffolder agent orchestrates retrieval, dialogue routing, and progress tracking, while a separately fine-tuned Socratic Dialogue agent handles the conversational mechanics of Socratic questioning. Separating pedagogical decision-making from dialogue generation distinguishes our approach from monolithic chatbot deployments and keeps each component independently evaluable, swappable, and updatable.

By utilizing open-weight models, researchers and instructors gain greater inspectability, local control, and reproducibility than proprietary API-only systems provide. This project develops and pilots a Virtual Teaching Assistant (VTA), a term we use to encompass both the administrative tasks of a traditional TA and the personalized instruction of a tutor. The VTA is

designed to respond with scaffolded prompts and targeted hints that keep students cognitively engaged. The specific contributions of this paper are (1) a thermo-fluid Socratic-dialogue dataset of 2,899 multi-turn dialogues constructed by adapting SocraticLM's Dean-Teacher-Student framework, (2) a step-rewriting stage that translates generic textbook solution steps into domain-specific guiding questions, motivated by the observation that SocraticLM's decomposition behavior does not transfer cleanly to thermo-fluid prompts, and (3) a behavioral comparison of LoRA and P-Tuning v2 fine-tuning of Qwen2.5-7B-Instruct on this dataset. Broader system components (RAG, LMS integration, classroom evaluation) are described as planned future work. While future plans for classroom implementation are detailed throughout this work, those evaluations will not be implemented in time for the current conference. In particular, this paper does not claim a completed measurement of pedagogical impact (e.g., transfer of reasoning or metacognitive change); rather, it documents the technical VTA development and a preliminary evaluation framework for future classroom study.

2. Technical progress: dataset construction and Socratic fine-tuning

We adapted SocraticLM [2], a Socratic teaching model originally developed for elementary mathematics (GSM8K and MAWPS), to university-level thermo-fluid sciences. Because the released implementation did not provide a directly transferable thermo-fluid dialogue-generation workflow, we rebuilt the pipeline for this domain, substituted the source problems for a domain-appropriate dataset, and re-ran fine-tuning on a current open-weight base model. Figure 1 shows the dataset construction pipeline, from raw problems through step rewriting to dialogue generation. Figure 2 shows the two parallel fine-tuning runs of Qwen2.5-7B-Instruct on the resulting corpus and the contrasting behavioral outcomes.

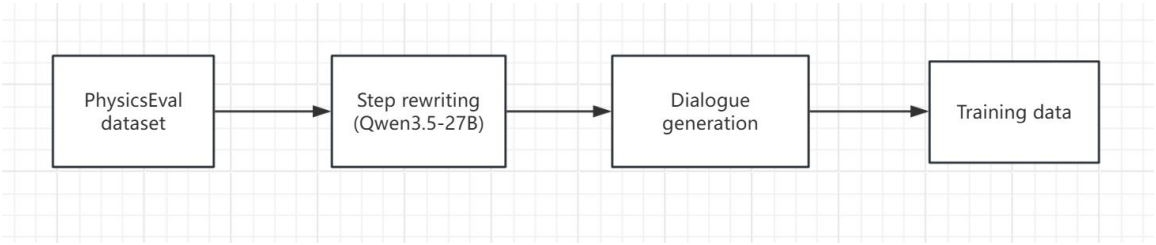

Figure 1. Dataset construction pipeline. Filtered PhysicsEval problems are passed through a step-rewriting stage (Qwen3.5-27B) that converts generic solution steps into domain-specific guiding questions, then a Teacher-Student dialogue generator (a simplified two-agent instantiation of SocraticLM's Dean-Teacher-Student framework) produces the training corpus.

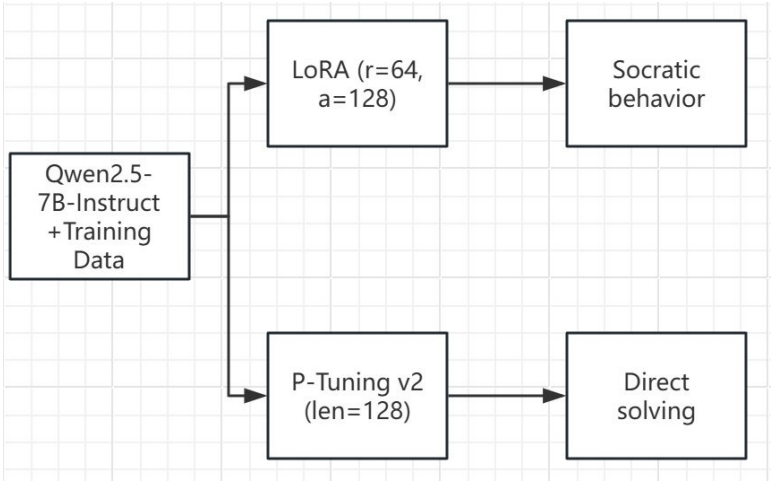

Figure 2. Parallel parameter-efficient fine-tuning of Qwen2.5-7B-Instruct on the Teacher-Student dialogue dataset. The LoRA branch (rank 64, alpha 128) produced Socratic dialogue behavior, while the P-Tuning v2 branch (prefix length 128) reverted to direct problem-solving on the same held-out test cases.

2.1 Dataset: PhysicsEval, filtered to high-difficulty thermo-fluid problems

We drew source problems from PhysicsEval [19], a public dataset of 19,609 physics problems with worked step-by-step solutions, each annotated with a difficulty score on a 1-to-10 scale. We filtered to higher-difficulty problems suitable for university-level instruction and applied domain keyword matching for thermodynamics, fluid mechanics, and heat transfer, retaining 2,992 problems suitable for tutoring-dialogue generation. By comparison, the original SocraticLM used 11,147 elementary-mathematics problems drawn from GSM8K and MAWPS.

2.2 Two-stage dialogue generation pipeline

The dialogue-generation pipeline runs in two stages, both implemented with Qwen3.5-27B [20] as the underlying language model.

Step rewriting. PhysicsEval ships with generic, language-model-style solution steps (for example, “apply the relevant conservation law”). For each problem, Qwen3.5-27B rewrites these into domain-specific sub-questions that a tutor would actually ask (for example, “What conservation law applies when fluid moves through a horizontal pipe of varying cross-section?”). This stage is necessary because, while SocraticLM’s Socratic conversational *format* transfers cleanly to thermo-fluid prompts, its problem *decomposition* behavior does not: in preliminary tests, applying SocraticLM directly to thermo-fluid problems caused it to rephrase the problem statement verbatim rather than break it into a sequence of guiding sub-questions. Explicit, domain-specific step rewriting at dataset-construction time is our solution.

Teacher-Student dialogue generation. Adapting SocraticLM’s Dean-Teacher-Student framework [2] but simplified to a two-agent Teacher-Student configuration in this first-pass implementation, Qwen3.5-27B drives both roles: the Teacher poses Socratic questions grounded in the rewritten sub-questions, and the Student responds. The Teacher has access to the full

solution, but the Student does not, mirroring the information asymmetry of a real classroom. SocraticLM's Dean role, which oversees and revises Teacher utterances at data-generation time, is omitted in this implementation; reintroducing it alongside the planned runtime Scaffolder agent (Section 3.4) is on the roadmap. After data preprocessing and surface-level quality filtering, 2,899 dialogues passed validation (96.9% of the 2,992 source problems), averaging approximately 9 turns each.

Each Teacher utterance, paired with its preceding dialogue context, was treated as one supervised training sample, yielding approximately 16,400 samples in total, compared with about 208,000 single-round samples for the original SocraticLM. The smaller training-set size is a deliberate first-pass choice: we generated only one student profile, corresponding to SocraticLM's persona type (1) ("understanding difficulty"), rather than the six cognitive states defined in SocraticLM's student cognitive state system. The implications of this choice for behavioral coverage are discussed in Section 2.4 below.

2.3 Fine-tuning Qwen2.5-7B-Instruct with LoRA

We initially compared two parameter-efficient fine-tuning (PEFT) methods on Qwen2.5-7B-Instruct: Low-Rank Adaptation (LoRA) and P-Tuning v2, the latter being the method used in the original SocraticLM paper. LoRA was more promising in this preliminary screen on both axes that matter for this project. On compute, LoRA finished training in approximately 87 minutes versus approximately 262 minutes for P-Tuning v2 on the same single-H100 hardware and training data, a roughly 3× speedup. On behavioral output, the LoRA checkpoint asked guiding questions and withheld direct answers in 4/4 held-out test cases (three drawn from the thermo-fluid in-domain set and one drawn from outside the training distribution) across single-turn and multi-turn protocols, whereas the P-Tuning v2 checkpoint reverted to direct problem-solving in 4/4 cases. The two methods are not directly comparable in trainable parameter count (LoRA's 161M versus P-Tuning v2's 15M), since they parameterize different parts of the model; a budget-matched head-to-head is left for future work. Both methods reached comparable losses in our setup; the exact magnitudes are less important than the observation that loss alone did not separate the two methods, while behavioral evaluation did. This is a small (n = 4) but suggestive preliminary observation: at these parameter budgets, training loss did not predict downstream Socratic adherence, and behavioral evaluation gave a much sharper signal. We treat this as a preliminary screen that warrants verification on a larger held-out set in future work. The remainder of this section focuses on the LoRA configuration in detail, since LoRA is the method we carried forward.

LoRA fine-tunes a frozen base model by inserting trainable low-rank matrices into selected weight projections of the transformer, leaving the original pretrained parameters unchanged. This design fits the present setting in three respects. First, because the base weights are frozen, the model's broad pretraining knowledge of physics, engineering, and natural language is preserved without catastrophic forgetting, which would be a real risk if a 7B model were fully fine-tuned on a 16K-sample domain corpus. Second, the resulting adapter is small (a few hundred megabytes) and can be redistributed independently of the base model, which is consistent with the open-source release plan: downstream users can pull the base from one repository and our

pedagogical adapter from another, then load them together at inference time. Third, multiple adapters can be swapped at inference time without reloading the base, which opens a path to one shared base model serving multiple courses, each with its own behavior or domain module.

Table 1 summarizes the LoRA configuration used in this work and the resulting training statistics. The 161M trainable adapter parameters represent 2.1% of the base model and were sufficient to induce Socratic dialogue behavior in the held-out test cases of Section 2.4.

Table 1. LoRA fine-tuning configuration for Qwen2.5-7B-Instruct on the Teacher-Student dialogue dataset.

Parameter	Value
Base model	Qwen2.5-7B-Instruct
PEFT method	Low-Rank Adaptation (LoRA)
Training data	2,899 Teacher-Student dialogues, ~16.4K supervised samples
Hardware	1× NVIDIA H100
LoRA rank (r)	64
LoRA alpha	128
LoRA dropout	0.05
Target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj
Learning rate	2×10^{-4}
LR scheduler	cosine, 5% warmup
Optimizer	AdamW (Trainer default)
Batch size	4 × 4 grad accum (16 effective)
Epochs	2
Precision	bf16
Trainable LoRA parameters	161M / 7.8B (2.1%)
Training wall-clock	~87 min
Final train_loss	~4.05

2.4 Sample interactions and known limitations

To make the LoRA checkpoint's behavior concrete, we present two illustrative cases. The first shows a successful multi-turn Socratic walkthrough; the second documents a known failure mode that motivates the next phase of dataset construction.

Table 2. Successful Socratic guidance: pipe-flow continuity and Bernoulli problem (LoRA, multi-turn).

Speaker	Dialogue Content
Problem	Water flows through a horizontal pipe narrowing from 0.04 m² to 0.01 m². Speed in the wider section is 2 m/s. Find the speed in the narrower section and the pressure difference.
Teacher	What principle relates the velocity of water to the cross-sectional area in this horizontal pipe?
Student	I think it has something to do with the flow rate being constant?
Teacher	That's right! The continuity equation states that the volume flow rate remains constant if the fluid is incompressible and the pipe is horizontal. How can you use this to find the velocity in the narrower section?
Student	$A_1 \cdot v_1 = A_2 \cdot v_2$, so $v_2 = A_1 \cdot v_1 / A_2$.
Teacher	Exactly! Now plug in the given values for A_1, v_1, and A_2 to calculate v_2. What do you get?
Student	$v_2 = 0.04 \cdot 2 / 0.01 = 8$ m/s.
Teacher	Great job! Now, what equation connects pressure difference with velocity change in a horizontal pipe?
Student	I can use Bernoulli's equation: $P_1 + \frac{1}{2}\rho v_1^2 = P_2 + \frac{1}{2}\rho v_2^2$.
Teacher	Correct! Since the pipe is horizontal, gravity doesn't factor into the equation. How can you rearrange it to find the pressure difference?
Student	$P_1 - P_2 = \frac{1}{2} \cdot 1000 \cdot (64 - 4) = 30{,}000$ Pa.
Teacher	Exactly! You've correctly calculated both the velocity and the pressure difference. Well done!

The Teacher does not reveal the final answers before the student produces them. The dialogue runs in two stages: the student first applies continuity ($A_1v_1 = A_2v_2$) to find $v_2 = 8$ m/s, then Bernoulli's equation to find $P_1 - P_2 = 30{,}000$ Pa, with the horizontal-pipe simplification provided by the Teacher. The opening turn is notable: the student offers a hedged, partially correct framing ("something to do with the flow rate being constant") and the Teacher confirms the underlying intuition before naming the principle, rather than rejecting the hedge or supplying the answer. This case demonstrates the LoRA checkpoint's typical behavior on problems where the student begins with a partial intuition; harder failure modes, where the student volunteers a wrong intermediate result, are documented in Table 3.

On problems where the student begins with no recognizable foothold (for example, responses such as "I don't know" or "I don't remember the formula"), the Teacher shifts toward more direct conceptual support, naming the relevant law, providing formulas, and reducing question depth, while still requiring the student to perform the substantive arithmetic. This adaptive shift is consistent with established scaffolding theory but represents a different operating mode from the Socratic engagement shown in Table 2; a more thorough characterization of this mode, including when it is and is not pedagogically appropriate, is left for future work.

Known limitation: incorrect-answer recognition. The LoRA checkpoint follows the Socratic format reliably but does not catch student computational errors. Table 3 documents a representative case in which a deliberately wrong intermediate step is accepted without challenge.

Table 3. Heat exchanger problem with an intentionally incorrect student response.

Speaker	Dialogue Content
Problem	Hot water 90°C → 50°C, cold water enters 20°C. Find outlet temperature and heat transfer rate.
Teacher	What equation relates the heat transfer rate to the temperature change of the hot water and its specific heat?
Student (intentionally wrong)	The mass flow rate is 3 kg/s, and ΔT = 90 − 20 = 70°C. (Correct ΔT for the hot stream is 90 − 50 = 40°C; the student has used the cold-water inlet instead of the hot-water outlet.)
Teacher	Good! Now calculate the heat lost by the hot water.

The model accepted the incorrect ΔT without challenging it. This failure mode maps directly to SocraticLM's Incorrect Answer Recognition Accuracy (IARA) metric. We hypothesize that the primary contributor is dataset coverage: we trained only with persona type (1) ("understanding difficulty"), whereas SocraticLM additionally generates persona type (3) ("calculation errors") along with approximately 10K augmentation samples specifically labeled "Incorrect-reply." Other contributors are also plausible (the underlying base model's calculation-checking ability, or limitations of the SFT objective for error detection), so closing this gap will be both a dataset and a methodology question. Generating the missing personas and augmentation data is the primary near-term motivation for Section 3.3.

3. Planned technical development

The broader system architecture planned for future development emphasizes an agentic VTA that integrates behavioral fine-tuning with course-specific domain knowledge under a multi-agent orchestration layer. This technical development is targeted for completion during the Spring 2026 semester. The technical plan focuses on four primary pillars: domain grounding, LMS integration, conversational refinement, and multi-agent orchestration, all built exclusively on an open-source stack.

3.1 Domain grounding and knowledge retrieval

The Socratic LoRA adapter from Section 2.3 shapes how the VTA talks (guiding questions, withheld answers, scaffolded inquiry) but is not designed to carry course-specific content knowledge. Course knowledge is intended to be supplied at inference time by a retrieval-augmented generation (RAG) pipeline built with the open-source LlamaIndex framework. The two components are intentionally complementary: fine-tuning sets the conversational style, RAG is intended to ground the substance, and together they aim to mitigate the hallucination and context-awareness failure modes that arise when either component is used in isolation. RAG can

ground responses in retrieved context but does not by itself guarantee correctness; the evaluation plan in Section 4 will measure whether grounding improves student-perceived accuracy in practice.

A key challenge in STEM-focused RAG is preserving non-prose content. In this initial text-based implementation, mathematical expressions will be represented using Markdown-formatted LaTeX (for example, $F = ma$). For geometric details and figures, the current VTA will rely on textual alt-text descriptions stored in the vector database. We recognize that text-only retrieval is fundamentally limited for thermo-fluid sciences, where T-s diagrams, P-h diagrams, free-body diagrams, and control-volume sketches carry information that resists prose summarization. Future iterations will transition to a multimodal RAG architecture using an open-weight vision-language base (e.g., Qwen-VL or LLaVA-Next), with figures retrieved alongside text and grounded in the same retrieval index.

The implementation is organized into three primary stages:

- **Knowledge Base Construction:** Course-specific materials, including lecture notes, homework assignments, textbooks, and Open Educational Resources (OER), will be processed into vector embeddings. Embeddings will be generated using the open-weight nomic-embed-text model and stored in the open-source Milvus vector database.
- **Contextual Retrieval:** When a student submits a query, the VTA will retrieve the most relevant document chunks from the vector database. This retrieved context will then be injected into the LLM prompt alongside the pedagogical instructions.
- **Optimized Retrieval Strategy:** To improve accuracy while controlling computational cost, the system will include an open-source reranking step. Initial retrieval results will be reordered by relevance score before being passed to the LLM.

3.2 Learning Management System (LMS) integration

For the VTA to be an effective classroom tool, it must be accessible through existing student workflows. We are developing an integration pathway that uses the Learning Tools Interoperability (LTI) 1.3 standard [21], ensuring that our open-source solution can be embedded into any major LMS. Because LMS integration entails handling student interaction data, deployment will follow institutional IRB review and FERPA-compliant data-handling procedures, including de-identification before research use (see Section 4.2) and instructor-controlled retention policies for raw interaction logs.

- **LTI-Based Web Application:** A dedicated web application, developed using the Flask framework and the open-source pylti1.3 module [22], will serve as the bridge between the VTA and the LMS (e.g., Canvas or Blackboard).
- **User Interface:** The front-end will be powered by Open WebUI [23], which provides a familiar, chat-centric interface. This interface also provides administrative tools for instructors to manage user access and security. It supports both Markdown and LaTeX for equation formatting.

- **Instructor Insights:** The integration will include the open-source LLM engineering platform Langfuse [24] for tracing and analytics. This allows instructors to monitor student progress anonymously and receive automated summary reports.

3.3 Conversational quality, error recognition, and model benchmarking

A critical near-term objective is to close the IARA gap documented in Section 2.4 and to move beyond the current first-pass dialogue style.

- **Persona expansion and "Incorrect-reply" augmentation:** We will extend the Teacher-Student dialogue pipeline to generate persona type (3) ("calculation errors") and an additional set of "Incorrect-reply" augmentation samples in the spirit of SocraticLM, so that the fine-tuned model learns to detect and challenge wrong intermediate steps rather than accept them.
- **Persona and flow refinement:** Future iterations will reduce the rigidity often observed in automated Socratic dialogues by introducing more varied response templates and dynamic persona management.
- **Pareto-frontier characterization of compute versus pedagogical quality:** The LoRA-vs-P-Tuning v2 contrast in Section 2.3 is one data point on a much larger compute-versus-output trade-off. We plan to benchmark a panel of open-weight base models spanning roughly 1B to 30B+ parameters, each fine-tuned via LoRA on the same Teacher-Student dialogue dataset, and to map their position on the Pareto frontier of training and inference compute cost versus Socratic-quality metrics (rate of guiding-question issuance, IARA, frequency of direct-answer leakage, and similarity to held-out reference dialogues). Two outcomes matter: identifying the smallest model that still meets pedagogical requirements, since smaller models can run on student-grade or mobile hardware and improve equity of access, and identifying the regime in which additional compute stops yielding pedagogical gains. Best-performing checkpoints across the frontier will be released on public model repositories for general use by the engineering education community.
- **Direct comparison against published Socratic baselines:** A natural baseline for our adapted system is the published SocraticLM checkpoint applied to thermo-fluid problems. We plan a head-to-head evaluation on a shared held-out set, scored with the SocraticLM evaluation suite (Overall Quality, IARA, CARA, SER, SRR), to make the value of domain-specific adaptation explicit and to provide a reference point for downstream researchers.
- **Cross-domain pipeline extension:** The Teacher-Student dialogue pipeline and step-rewriting stage are domain-agnostic. We plan to apply the same construction protocol to adjacent STEM domains (statics, dynamics, materials science, and electrical engineering) to test whether one pedagogical adapter generalizes across courses or whether course-specific adapters are required.

3.4 Multi-agent orchestration

The planned deployed VTA architecture will extend beyond the fine-tuned Socratic Dialogue agent of Section 2.3 to a multi-agent system. A Scaffolder (orchestrator) agent will handle orchestration: reading the student's input, querying the RAG knowledge base when domain content is needed, routing between Socratic dialogue and direct explanation depending on the student's state, tracking progress across turns, and surfacing recurring misconceptions to the instructor. The fine-tuned Socratic Dialogue agent will handle only the conversational mechanics of Socratic questioning. This separation of concerns will scope the fine-tuned model's behavior tightly, making it easier to evaluate, swap out, or upgrade as Pareto-frontier work (Section 3.3) identifies better base models. The orchestration logic, by contrast, will live in a more interpretable and easily updatable layer that does not require retraining when course content or pedagogical strategies change. This division also opens a natural path for reintroducing data-time supervision (analogous to SocraticLM's Dean role) as a separate quality-control agent within the Scaffolder framework.

4. Future evaluation plan for classroom implementation

While the current phase focuses on the technical realization of the VTA tool, a comprehensive evaluation strategy is planned for subsequent academic terms to quantify impact and guide iteration. This plan will compare a control baseline with the VTA implementation. Because this is a Work-in-Progress focused on technical development, the pedagogical measurement components described below are presented as a preliminary protocol (including proposed coding rubrics and reliability procedures) rather than completed results. Accordingly, this paper does not present a finalized participant recruitment target or a formal a priori power analysis; the initial classroom deployment is intended as a pilot/feasibility implementation that will inform effect-size and variance estimates for a later, fully powered study design.

4.1 Comparative study design

The project uses a structured comparison to isolate the impact of the intervention:

- **Control Group:** This group establishes the "business-as-usual" baseline using pre-implementation data collected from semesters with traditional human TA support and conventional instructional workflows. We acknowledge that comparison against historical data introduces confounders (cohort, instructor, exam-difficulty drift, and post-pandemic enrollment effects), so pilot-stage findings will be treated as feasibility evidence rather than causal attribution. A properly powered study with within-cohort assignment is reserved for the follow-up phase informed by these pilot results.
- **Experimental Groups:** The VTA will be introduced as a course-embedded support tool. This includes a Socratic-enabled VTA configured for inquiry-based learning and guided questioning to evaluate the specific pedagogical impact of the behavior-shaped model.
- **Pilot Sampling and Analysis Scope:** Participant counts for the initial deployment will be determined by feasible enrollment and implementation logistics in participating course sections, with recruitment and consent rates reported. Any subgroup or stratified analyses (e.g., by course section or baseline preparation level) will be treated as exploratory and

only conducted if sample sizes are adequate; otherwise, results will be reported in aggregate and used to refine the subsequent study protocol.

4.2 Qualitative and subjective metrics

- **Anonymized Interaction Logs:** Student-VTA interactions will be logged through the Open WebUI interface and de-identified prior to research analysis to enable periodic, privacy-preserving summaries. De-identification will include removal or replacement of direct identifiers (e.g., names, email addresses, LMS usernames or IDs) with study IDs, minimization of nonessential metadata, and redaction of self-disclosed identifying information in free-text logs before logs are shared with raters or used for coding. In future classroom studies, sampled logs will be coded with a rubric designed to capture proximal pedagogical indicators of Socratic scaffolding (e.g., prompting quality, student self-explanation, evidence of reasoning progression, and metacognitive reflection statements) in addition to technical accuracy. Two raters will independently code a stratified sample, with interrater reliability reported using Cohen's κ or Krippendorff's α before consensus coding is applied to the remainder.
- **Student Feedback Surveys:** Pre- and post-intervention surveys will draw on Likert-scale items and open-ended prompts administered before and after deployment. In addition to general perceptions (e.g., clarity and usefulness), the future survey design will include items targeting constructs more directly aligned with the pedagogical mechanism under study, such as perceived support for reasoning transfer, self-explanation, and metacognitive awareness during problem solving. Open-ended prompts will be used to gather qualitative insights into student experiences with the inquiry-based dialogue.

4.3 Quantitative and performance metrics

- **Proximal pedagogical measures:** To capture mechanism-relevant effects of Socratic scaffolding rather than only summative achievement, we will track measures more directly tied to the intervention: pre/post concept inventories on thermodynamics and fluid mechanics, items targeting common misconceptions in the domain (e.g., confusing intensive and extensive properties, or misapplying conservation laws), worked-problem transfer tasks, direct-answer leakage rate from interaction logs, and incorrect-answer recognition rate. These are treated as proximal indicators of whether the Socratic interaction is doing the pedagogical work the design intends.
- **Course-Level Outcomes:** Evaluation will track exam scores and overall course grade distributions. These quantitative outcomes will be compared against the historical control baseline to measure shifts in academic achievement. They are treated as indicators rather than standalone evidence of Socratic scaffolding quality.
- **Long-term exploratory outcome (FE exam):** As a long-term exploratory outcome, student performance on the Fundamentals of Engineering (FE) exam in thermodynamics and fluid mechanics sub-disciplines may be tracked when matched longitudinal data are available. FE performance is distant from the immediate intervention and subject to many

confounds; it is reported here only as a downstream signal, not as a primary measure of Socratic interaction quality.

- **Actionable Instructor Feedback:** Summary reports will be generated to inform the course instructor of potential gaps in understanding and recurring student misconceptions, providing actionable feedback to refine instruction during the term.

5. Development and implementation timeline

The project follows a phased timeline. The technical development presented in this paper represents the first phase of this effort.

- **Current phase (Spring 2026): Model Refinement and Baseline Collection.** Current focus is on persona expansion and Incorrect-reply augmentation to close the IARA gap, RAG pipeline development, Scaffolder agent development for multi-agent orchestration, model benchmarking across a panel of open-weight bases for the compute-versus-quality Pareto frontier, and establishing control-group parameters.
- **Fall 2026: Pilot Implementation and Refinement.** The VTA will be deployed in a single pilot section to test LTI integration and pedagogical effectiveness.
- **Spring 2027: Multi-Section Implementation and Scale-Up.** Broad implementation across multiple disciplines to facilitate the comparative study and public release of the refined model weights.

6. Conclusion

This Work-in-Progress reports a reproducible dataset construction pipeline that adapts SocraticLM to thermo-fluid sciences, two parallel parameter-efficient fine-tuning runs of Qwen2.5-7B-Instruct, and a preliminary behavioral evaluation in which LoRA produced Socratic dialogue behavior across all four held-out test cases while P-Tuning v2 reverted to direct problem-solving. A practical takeaway is that comparable training and evaluation losses can mask very different behavioral outcomes, which has implications for how Socratic-style tutors should be evaluated during development. By using an open stack throughout, this work addresses transparency, security, and institutional autonomy gaps in the deployment of generative AI for engineering education.

We recognize that “Socratic form” does not by itself guarantee effective pedagogy, and that the current checkpoint still fails to reliably catch student computational errors. The next phase has several concurrent threads: closing the IARA gap through persona expansion and Incorrect-reply augmentation, characterizing the compute-versus-quality Pareto frontier across a panel of open-weight bases, integrating RAG-based domain grounding under a Scaffolder-orchestrated multi-agent runtime, and conducting mixed-methods classroom evaluation in thermo-fluid sciences courses. The dataset, adapter weights, and evaluation problems will be released for use by the engineering education community.

References

[1] A. Yang, B. Yang, B. Hui, et al., “Qwen2.5 Technical Report,” arXiv preprint arXiv:2412.15115, 2024.

[2] J. Liu, Z. Huang, T. Xiao, J. Sha, J. Wu, Q. Liu, S. Wang, and E. Chen, “SocraticLM: Exploring Socratic Personalized Teaching with Large Language Models,” Advances in Neural Information Processing Systems, vol. 37, pp. 85693-85721, 2024.

[3] M. Kazemitabaar et al., “CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs,” in Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1-20. doi: 10.1145/3613904.3642773.

[4] D. Zapata-Rivera et al., “Editorial: Generative AI in education,” Frontiers in Artificial Intelligence, vol. 7, 2024. doi: 10.3389/frai.2024.1532896.

[5] D. J. Hornsby and R. Osman, “Massification in higher education: large classes and student learning,” Higher Education, vol. 67, no. 6, pp. 711-719, 2014. doi: 10.1007/s10734-014-9733-1.

[6] E. Buckner and Y. Zhang, “The quantity-quality tradeoff: a cross-national, longitudinal analysis of national student-faculty ratios in higher education,” Higher Education, vol. 82, no. 1, pp. 39-60, 2021. doi: 10.1007/s10734-020-00621-3.

[7] J. Stamper, R. Xiao, and X. Hou, “Enhancing LLM-Based Feedback: Insights from Intelligent Tutoring Systems and the Learning Sciences,” in Artificial Intelligence in Education, A. M. Olney et al., Eds. Cham: Springer Nature Switzerland, 2024, pp. 32-43. doi: 10.1007/978-3-031-64315-6_3.

[8] Meta AI, “Llama 3.2: Revolutionizing edge AI and vision with open, customizable models,” 2024. [Online]. Available: https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/. [Accessed: 06-Jan-2025].

[9] A. Q. Jiang et al., “Mistral 7B,” arXiv, 2023. doi: 10.48550/arXiv.2310.06825.

[10] M. Abdin et al., “Phi-4 Technical Report,” arXiv, 2024. doi: 10.48550/arXiv.2412.08905.

[11] J. R. Carbonell, “AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction,” IEEE Transactions on Man-Machine Systems, vol. 11, no. 4, pp. 190-202, 1970. doi: 10.1109/TMMS.1970.299942.

[12] K. VanLehn, “The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems,” Educational Psychologist, vol. 46, no. 4, pp. 197-221, 2011. doi: 10.1080/00461520.2011.611369.

[13] J. A. Kulik and J. D. Fletcher, “Effectiveness of Intelligent Tutoring Systems: A Meta-Analytic Review,” Review of Educational Research, vol. 86, no. 1, pp. 42-78, 2016. doi: 10.3102/0034654315581420.

[14] B. S. Bloom, “The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring,” Educational Researcher, vol. 13, no. 6, pp. 4-16, 1984. doi: 10.3102/0013189X013006004.

[15] Google, “Guided Learning in Gemini: From answers to understanding,” 2025. [Online]. Available: https://blog.google/products-and-platforms/products/education/guided-learning/.

[16] OpenAI, “Introducing study mode,” Jul. 29, 2025. [Online]. Available: https://openai.com/index/chatgpt-study-mode/. [Accessed: 29-Apr-2026].

[17] HKUDS, “DeepTutor: An Open-Source Intelligent Tutoring System,” 2024. [Online]. Available: https://github.com/HKUDS/DeepTutor.

[18] M. T. H. Chi and R. Wylie, “The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes,” Educational Psychologist, vol. 49, no. 4, pp. 219-243, 2014. doi: 10.1080/00461520.2014.965823.

[19] O. Siddique, J. M. A. U. Alam, M. J. R. Rafy, S. R. Raiyan, H. Mahmud, and M. K. Hasan, “PhysicsEval: Inference-Time Techniques to Improve the Reasoning Proficiency of Large Language Models on Physics

Problems," arXiv preprint arXiv:2508.00079, 2025. [Online]. Available: https://huggingface.co/datasets/IUTVanguard/PhysicsEval.

[20] Qwen Team, "Qwen/Qwen3.5-27B," Hugging Face, 2026. [Online]. Available: https://huggingface.co/Qwen/Qwen3.5-27B. [Accessed: 29-Apr-2026].

[21] 1EdTech, "Learning Tools Interoperability," 2019. [Online]. Available: https://www.1edtech.org/standards/lti. [Accessed: 06-Jan-2025].

[22] D. Viskov, "dmitry-viskov/pylti1.3," 2025. [Online]. Available: https://github.com/dmitry-viskov/pylti1.3. [Accessed: 14-Jan-2025].

[23] Open WebUI, "open-webui," 2025. [Online]. Available: https://github.com/open-webui/open-webui. [Accessed: 14-Jan-2025].

[24] Langfuse, "langfuse," 2025. [Online]. Available: https://github.com/langfuse/langfuse. [Accessed: 14-Jan-2025].