# Science of Reading Teacher Training Simulator: An Interactive VR Platform for Literacy Education

Hemasree Pujari
Department of Computer Science and Engineering
University at Buffalo, The State University of New York
hemasree@buffalo.edu
*Advised by: Dr. Karthik Dantu*

***Abstract*—**The Science of Reading (SoR) framework has become the empirical foundation for early literacy instruction, but only 31% of U.S. fourth graders meet 2024 NAEP reading proficiency standards. Part of the problem is that pre-service teachers do not get enough chances to practice SoR-aligned instruction in realistic classroom conditions. The simulators that do exist either need motion-capture rigs, trained human role-players or work off scripted dialogue and none of them scale to the volume of practice teachers actually need. We built the Science of Reading Teacher Training Simulator, a Virtual Reality platform where teachers rehearse lessons against AI-driven student agents inside a Unity classroom rendered on Meta Quest headsets. Three components make the system work: a custom speech pipeline that returns International Phonetic Alphabet (IPA) phonemes rather than only words, with eight accent profiles; a three-agent LLM architecture that handles a child's verbal response, emotional state and physical action separately and a Retrieval-Augmented Generation layer grounded in SoR curriculum and transcribed teacher conversations. We describe each component, the engineering choices behind them and the design issues that came up along the way.

## I. INTRODUCTION

The Science of Reading (SoR) is the body of research that explains how children learn to read. It is organized around five pillars: phonemic awareness, phonics, vocabulary, fluency and comprehension [1], [2]. After the New York State Education Department mandated in 2023 that elementary teacher preparation programs align with SoR principles, instructors of literacy methods courses now need to give pre-service teachers a lot of practice delivering phoneme-explicit instruction. The supply of practice opportunities has not kept up with the demand.

Field placements are limited and uneven. Peer micro-teaching does not feel anything like teaching a real four-year-old. Human-actor simulators produce convincing dialogue but cost too much per session to scale to the practice volume teachers actually need. Off-the-shelf chatbots reply in adult prose, with no phoneme awareness, no emotion and no body so they do not help either. The space between these extremes is where we built our system.

A teacher puts on a Meta Quest headset, walks into a Unity classroom and starts teaching a lesson. The child avatars hear what the teacher says down to the level of individual phonemes, respond in kid-like voices, react with appropriate emotion and motion, stay grounded in actual SoR lesson content. The contributions of this work are:

- A speech pipeline that produces IPA phoneme transcriptions with eight accent-specialized recognizers, so the simulator works for the internationally diverse candidate population.
- A three-agent LLM architecture (Dialogue, Emotion, Action) that separates what the child says from how they feel and what they do. We found that decomposing the response this way matters more for perceived believability than the choice of LLM itself.
- A Retrieval-Augmented Generation (RAG) grounding layer built on SUNY New Paltz SoR curriculum and transcribed teacher conversations which keeps the simulated children on-pillar and on-level.
- A Unity and Meta Quest deployment that runs with no motion capture and no live human interactors, which removes the per-session cost that has historically limited large-scale teacher rehearsal.

## II. BACKGROUND AND RELATED WORK

### *A. The Science of Reading*

The five-pillar SoR framework rests on decades of cognitive and neuroscientific research showing that fluent reading begins with the ability to map graphemes to phonemes (phonics) and to manipulate phonemes in spoken words (phonemic awareness) [1]. Vocabulary, fluency and comprehension build on top of this foundation. The practical consequence is that effective SoR teaching has to operate at the phoneme level. A teacher needs to know that the first sound in "man" is /m/, not "em" and they need to catch and correct it when a child substitutes one phoneme for another. Fig. 1 shows the five pillars and the sub-skills our simulator targets within each.

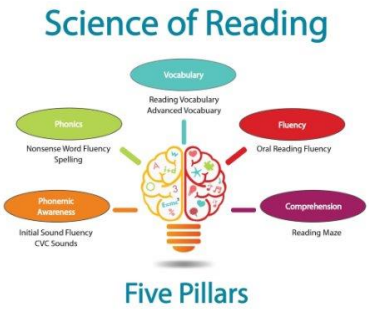


***Fig. 1.*** *The five pillars of the Science of Reading and the sub-skills targeted within each pillar (Initial Sound Fluency and CVC sounds; Nonsense Word Fluency and spelling; reading and advanced vocabulary; Oral Reading Fluency; and Reading Maze).*

### *B. Existing Teacher-Training Simulators*

Platforms that use trained human interactors behind student avatars and have them voice the avatar in real time. The dialogue is convincing because there is a person on the other end. The cost is also a person on the other end, which caps how many hours of practice a single trainee can get. Asynchronous video platforms let's teachers watch but not interact. Pure-LLM chatbots talk back, but they reply like adults, drift off-topic, ignore articulation errors and have nothing to show on screen. None of these options match the volume, fidelity and cost requirements that the New York State mandate has created.

## III. SYSTEM ARCHITECTURE

Fig. 2 shows the overall flow. The teacher wears a Meta Quest headset and speaks to one of the child avatars seated in the Unity classroom. The audio is streamed to a backend that runs five things, partly in parallel:

- Speech-to-text together with the custom phoneme pipeline (Section IV), which produces both a word-level transcript and a precise IPA phoneme sequence.
- Vector retrieval against a ChromaDB store of SoR curriculum and transcribed teacher dialogues (Section VI).
- Three concurrent calls to OpenAI LLMs that produce the verbal reply, the emotional state and the physical action (Section V).
- Text-to-speech synthesis using a child voice profile, with prosody modulated by the predicted emotion.
- Unity animation of the avatar's lip-sync, face and body, driven by the action and emotion outputs.

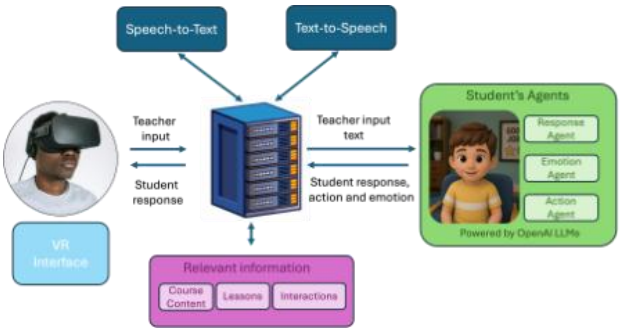


***Fig. 2.*** *High-level system architecture. Teacher audio enters the headset and flows through the speech pipeline, RAG retrieval and the three agents in parallel before TTS and Unity render the response.*

### *A. Latency*

The biggest engineering challenge has been latency. A real four-year-old usually answers within a second or two and anything noticeably slower than that breaks the illusion. We had to design the pipeline so that retrieval, the three LLM calls and TTS could overlap rather than running one after the other. The standard STT and the custom phoneme pipeline run side by side so the agents can start reasoning over the word-level transcript as soon as it arrives and only have to wait on the IPA sequence to finalize. The three agents are dispatched together; the controller waits for the slowest of them, not for all three in series. TTS streams into Unity at the phoneme level which lets the avatar's mouth start moving on the first phoneme of the reply.

### *B. Backend Services*

The backend is split into four services, each independently deployable. A speech service hosts the standard STT, the accent classifier and the phoneme recognizer. A retrieval service wraps ChromaDB and exposes a typed interface to the agents. An agent service holds the prompts and the OpenAI client logic for the three cooperating agents. A rendering bridge turns agent output into Unity animation calls and TTS phoneme streams. The split makes it possible to swap any single piece without touching the rest which matters because we expect to replace the OpenAI backend with a self-hosted model once the latency picture allows it.

## IV. PHONEME-AWARE SPEECH PIPELINE

### *A. Why Words Are Not Enough*

A standard speech-to-text system gives back whole words like "man walks ahead." For most applications that is fine, but for an SoR lesson the exact sound the teacher made is what matters. If the lesson target is the phoneme /m/ and the teacher actually says "muh" (/mʌ/), the correct child response is different from the response if the teacher had said the clean /m/. Word-level transcription erases that distinction and any agent that reads only the spelled transcript will miss

articulation errors that a real pre-K student would notice and copy.

### *B. Pipeline Stages*

The pipeline runs in three stages, shown in Fig. 3:

- A standard STT produces a word-level transcript. This is used to seed the phoneme recognizer and to align the audio with the retrieval step.
- A custom pipeline runs an accent classifier that routes the audio to one of eight accent-specialized phoneme recognizers. The output is an IPA phoneme sequence rather than a sequence of words.
- The IPA sequence (for example, /m æ n/ for "man") is sent to the agents along with the word-level transcript so the model can react to articulation rather than spelling.

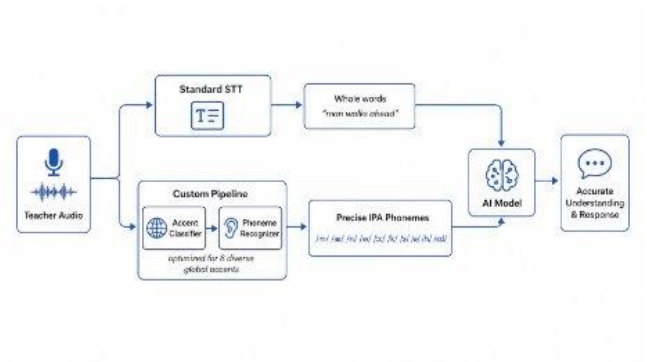


***Fig. 3.*** *The phoneme-aware speech pipeline. Teacher audio is processed in parallel by a standard STT and a custom accent-routed phoneme recognizer. Both outputs feed the downstream LLM agents.*

### *C. Eight Accent Profiles*

A single English phoneme model trained on American speech degrades quickly on non-native phonological substitutions: /v/ for /w/, retroflex consonants, tonal influence and so on. The pre-service teacher population is internationally diverse and we did not want the simulator to penalize candidates for their natural articulation while still surfacing mispronunciations that would matter for a child learner. The eight profiles cover the most common backgrounds we observe. New profiles can be added by training a recognizer on the relevant phonological data and adding a route in the accent classifier.

## V. MULTI-AGENT BELIEVABLE-STUDENT ARCHITECTURE

Asking a single LLM to "respond as a four-year-old" produces text that reads like an adult writing baby talk. The student feels scripted, the emotional content drifts between turns and there is no body to put on screen. We split the simulated child into three cooperating agents that share input context but reason independently. Fig. 4 shows the dataflow.

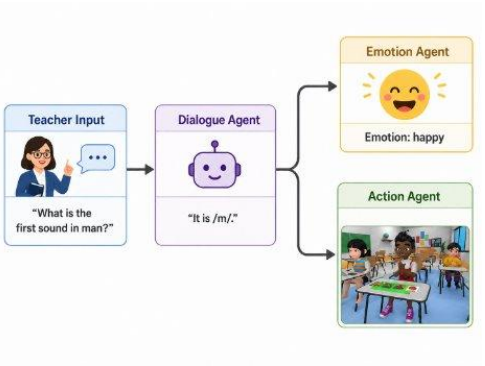


***Fig. 4.*** *The three cooperating agents. Teacher input fans out to the Dialogue, Emotion, and Action agents in parallel. Their outputs are reconciled by a lightweight controller and routed to TTS and the Unity animator.*

### *A. Dialogue Agent*

The Dialogue Agent takes the teacher's utterance, the IPA phoneme sequence from Section IV, the retrieved curriculum context and the running conversation history. It produces a short verbal reply encoded in ARPABET phonemes (for example, {K} for /k/) so the downstream TTS can render the exact phonetic articulation rather than a spelling-based approximation. Without it, a child whose intended utterance is the isolated phoneme /m/ comes out of TTS as the word "em."

### *B. Emotion Agent*

The Emotion Agent reads the same input bundle and outputs a discrete emotional label (happy, confused, frustrated, proud, sleepy and so on) along with an intensity. The label drives the avatar's facial blend shapes in Unity and modulates TTS prosody. Keeping emotion separate from dialogue lets the simulator do something pre-K children actually do: express frustration in tone and face while the verbal response stays a one-syllable answer.

### *C. Action Agent*

The Action Agent picks a physical behavior such as raising a hand, looking away or smiling, from a fixed library of pre-K classroom behaviour's. Actions are timed to start slightly before or during the verbal reply, which is what real children do.

### *D. How the Agents Stay in Sync*

The agents are not strictly independent. A teacher utterance that is on-pillar, articulate and gentle should generally produce a child response that is verbally engaged, emotionally positive and physically attentive. We get loose coordination through a shared context bundle that is built once per teacher turn and broadcast to all three agents. The bundle holds the standard transcript, the IPA sequence, the retrieved curriculum passages, the avatar's persistent profile (phonological tendencies, baseline temperament,

engagement style) and a compact summary of the conversation so far. Each agent prompt tells the model to respect the bundle and optimize its own output type.

## VI. GROUNDED IN REAL SCIENCE OF READING CONTENT

LLMs left to themselves will improvise. They will give reading behavior that is too advanced, too articulate, or just inconsistent with the lesson at hand. A four-year-old who suddenly defines "phoneme" in a complete sentence is not useful for teacher training. To prevent this, every utterance is grounded against retrieved content from a vector database (ChromaDB) populated with The SUNY New Paltz Science of Reading Fundamentals certification curriculum [2].

For each teacher utterance, the system computes an embedding, retrieves the top-k most relevant passages and includes them in the prompt sent to the three agents. This keeps the simulated child on-pillar (operating within the lesson's intended SoR pillar) and on-level (responding with vocabulary, articulation and metacognitive sophistication appropriate to ages four and five). It also keeps lesson content out of the prompts themselves, so adding a new lesson or a new student profile is a matter of adding documents to the vector store rather than editing the system prompts.

## VII. THE VIRTUAL CLASSROOM

The classroom is built in Unity and deployed to Meta Quest headsets. Three design choices set it apart from previous VR teacher-training systems.

### *A. No Motion Capture, No Live Interactors*

Unlike Mursion-class systems, our simulator does not require a human in the loop or a motion-capture suit. Avatar motion is procedural, driven by the Action Agent and standard Unity animation blend trees. Removing the per-session human-actor cost is what makes large-scale repeated practice realistic for any individual pre-service teacher. Fig. 5 shows a teacher in the headset alongside the in-headset view of a child avatar.

***Fig. 5.*** *A teacher wearing a Meta Quest headset (left) and the in-headset view of a child avatar during a lesson (right). The avatar's motion and facial expression are generated procedurally from the agent outputs.*

### *B. Extensible Avatar Profiles*

Each child avatar has a profile with baseline phonological tendencies, an emotional baseline, and an engagement style. New avatars can be added without code changes, which is what we need to model, for example, an English-language learner or a child with a specific articulation pattern. The same profile schema is what would let the platform extend to other clinical-training domains in future work (Section VIII), without changes to the underlying engine.

### *C. Lesson Selection and Difficulty*

Teachers choose lessons by SoR pillar (Phonemic Awareness, Phonics, Vocabulary, Fluency, Comprehension) and by difficulty level. Fig. 6 shows the in-headset lesson-selection interface. The same lesson can be replayed against different avatar profiles, so a trainee gets to see how the same instructional move plays out with different students. This is the kind of varied practice that field placements rarely provide because a single classroom does not contain the full distribution of student profiles a teacher will eventually meet.

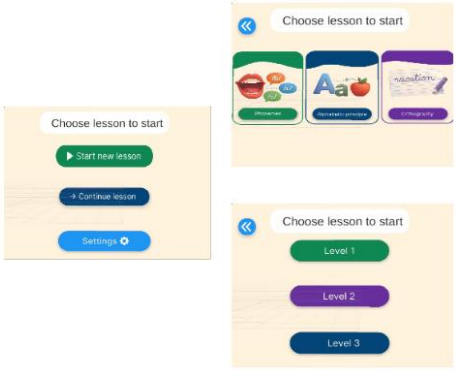


***Fig. 6.*** *In-headset lesson-selection UI. Teachers pick a pillar (phonics, vocabulary, phonemic awareness, etc.) and a difficulty level (Level 1–3) before entering the classroom.*

*Fig. 7. Wide view of the Unity classroom with seated child avatars. The classroom layout, lighting, and avatar positioning are designed to mimic a typical pre-K instructional setting.*

## VIII. CONCLUSION AND FUTURE WORK

We have built a Virtual Reality simulator for Science of Reading teacher training that puts together accent-aware phoneme recognition, a three-agent believable-student model, RAG grounding against real SoR content and Unity classroom rendering on Meta Quest. The platform sits in the gap between human-actor simulators that cannot scale and pure-LLM chatbots that are not believable and it is the first system we are aware of that operates at the phoneme level rather than the word level for SoR practice.

### *A. Limitations*

The eight accent profiles cover the backgrounds we see most often, but they do not exhaust the space. Trainees with profiles outside the supported set may see degraded recognition until we extend the profile library. The avatar profiles in the current version were authored by the project team rather than calibrated against a developmental corpus, so individual avatars are believable in isolation but the set as a whole probably does not span the full distribution of real pre-K behavior. We have not yet run a controlled user study; the only evaluations to date are demo-day walkthroughs and informal feedback from partner faculty. The system also depends on LLM inference, which adds latency variance.

### *B. Future Work*

Several directions follow naturally. On the teacher-evaluation side, we are designing automatic feedback that scores instructional moves against SoR rubrics. We are also planning a formal user study with pre-service teachers in our partner literacy methods courses, where we will measure whether time spent in the simulator transfers to live classroom performance and how it compares to incumbent video-based and human-actor platforms.

The same platform structure (Unity classroom, multi-agent stack, RAG grounding, accent-aware speech) should also adapt to Speech-Language Pathology and special-education clinical training. Existing tools in that space, such as the Master Clinician Network and SimuCase, rely on pre-recorded video and do not support live participation, real-time feedback, or unpredictable student behavior. Adapting our system to that domain would require new avatar profiles encoding common pediatric speech-sound disorders, a domain-specific RAG corpus and supervisor review hooks. We plan to explore this extension under the AI4ExceptionalEd funding stream.


## ACKNOWLEDGMENT

I would like to thank my teammates, Aditya Verma, Rose Vuluvabeeti, Daulet Mukan, Kirupanandan Jagadeesan Liza Kuzmina, Utkarsh Bansal for their contributions to the Science of Reading Teacher Training Simulator. Their work on the Unity classroom environment, backend architecture, speech-processing pipeline, multi-agent system, retrieval infrastructure, testing and integration was essential to the completion of this project. I am also grateful to our advisors, Dr. Karthik Dantu and Ranga Setlur for their guidance, feedback and support throughout the project.


## REFERENCES


[1] National Center for Education Statistics, “The Nation’s Report Card: 2024 Reading Assessment,” U.S. Department of Education, 2024. [Online]. Available: https://www.nationsreportcard.gov/reading

[2] SUNY New Paltz, “Science of Reading Fundamentals,” Online Certification Program. [Online]. Available: https://learn.newpaltz.edu

[3] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Advances in Neural Information Processing Systems, 2020.

[4] Chroma DB: Chroma, "Chroma - the AI-native open-source embedding database," 2024. [Online]. Available: https://www.trychroma.com

[5] Unity Technologies, “Unity Real-Time Development Platform,” 2024. [Online]. Available: https://unity.com

[6] Meta Platforms, Inc., “Meta Quest Developer Documentation,” 2024. [Online]. Available: https://developer.oculus.com