One Local Model, Fifteen Spoken Students: An Industrial Lecture Hall

Shyamal Suhana Chandra
Domino Data Systems
dominodatasystems@gmail.com

***Abstract*—Training rooms and product rehearsals need a lecturer who stays on one topic, an audience that can interrupt, and a record that can be replayed. Staffing that room with a separate language model for every student is expensive and makes the cast hard to reproduce. This industrial-track report describes a local lecture hall that uses one Ollama model for the lecturer and for fifteen students. The students differ by temperature and top-k, not by weights. A seed fixes the seating, the voices, a five-to-ten-minute speech budget, and the times at which hands rise. Speech is macOS `say`, so a raised hand stops the lecturer mid-sentence. The lecturer then thinks in a larger prompt budget and answers aloud. Each spoken paragraph is sent, in full, to a second Ollama server for a picture. The paper reports the architecture, the scheduler measured by replaying it, one image-generation timing from development, and the operating lessons. It does not report a learning-outcome study.**

***Index Terms*—industrial track, local language models, lecture simulation, top-k sampling, speech, image generation**

I. Introduction

A lecture is a timed performance. The speaker has a subject, a voice, and a clock. Students break that clock with questions that are sometimes careful, sometimes probing, and sometimes unrelated to the subject. Classroom systems have long tried to hold this kind of mixed-initiative talk [1]. Recent agent simulations show that a language model can play many people if each person has memory and a plan [2]. Both lines of work are heavier than a rehearsal tool that has to run on one machine, next to the model the team already downloaded.

The hall in this paper is that lighter tool. It is a Rust application with a pixel-style window [3]. One local model, served by Ollama [4], speaks as the lecturer and as every student. Role is a sampling setting plus a prompt, which is the same idea Fan *et al.* used when they drew open-ended text from a truncated distribution instead of searching for the single most likely sentence [5]. The hall makes the truncation a property of the seat: a definition-minded student samples from a colder, smaller set than a student who is supposed to leave ordinary reality.

Three practical constraints shaped the build.

- The lecture clock is speech, not tokens. A hand has to be able to cut off audio that is already playing.
- The model should not reload when the speaker changes. Context length is therefore one loaded window, while the text placed in that window changes with the job.
- Pictures of the lecture have to come from the same vendor interface the operators already use. On the machine used here, that required a second, older Ollama process, because the current server refused image models [6].

The rest of the paper describes the system, shows the cast and the hand schedule as plots taken from the scheduler itself, and records what broke when text generation and image generation shared a deployment.

II. Related systems

AutoTutor holds a mixed-initiative dialogue, with an animated agent, and was built to be deployed in classrooms [1]. The hall is not a tutor in that sense. It does not score a human learner. It produces a spoken lecture and a seeded audience so that a topic can be heard, interrupted, and illustrated.

Generative agents give each character a memory stream, reflections, and a daily plan, and they show that social behavior can emerge from those parts [2]. The hall does not give students a day or a memory beyond the transcript. Diversity is purchased at decode time. Top-k sampling keeps each draw inside the k most likely tokens [5]. Nucleus sampling later argued for a dynamic cutoff instead of a fixed k [7]. The hall stays with temperature and top-k because those are the two knobs Ollama exposes per request, and because a fixed pair can be written on a seat and reproduced from a seed.

Long contexts are not used evenly. Liu *et al.* found that relevant material in the middle of a long prompt is easier to miss than material at either end [8]. The hall therefore does not dump an unbounded transcript into every call. It keeps a tail that fits a stated token budget, and it gives answers a larger budget than lecture paragraphs.

The default text model in the command line is `gemma3:1b`. Gemma 3 is a family of open models from 1B to 27B parameters, and the 1B model is specified with a 32K context [9]. The hall does not hard-code that number. It reads the context length the local server reports and sizes its budgets from the report. Passage pictures default to Z-Image-Turbo, a 6B distilled text-to-image model aimed at few-step photorealistic generation [10], called through Ollama's experimental image interface [6].

III. Architecture

Figure 1 is the whole deployment. A command line supplies the topic, an optional seed, the chat model, the image model, the image host, and an optional question rate. The window shows those values and offers Begin and Stop. It does not edit them. A session thread then owns the lecture.

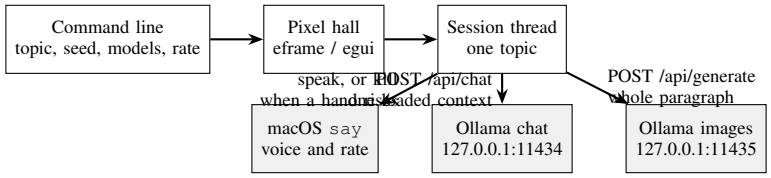


Fig. 1. The hall keeps one text model loaded and sends passage illustrations to a second Ollama server. Speech is a real process, so a raised hand can stop the lecturer in the middle of a sentence.

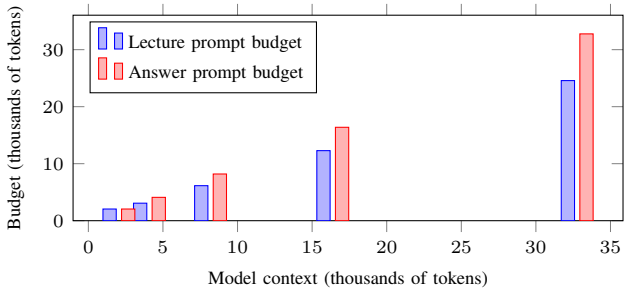


Fig. 2. Prompt budgets. Every call still loads `num_ctx` at the model maximum, so the runner is not restarted between speakers. When that maximum is above 2,048, the lecture prompt is cut to three quarters of it. The answer and the private notes use the full window. For the 32,768-token context reported for `gemma3:1b`, those budgets are 24,576 and 32,768.

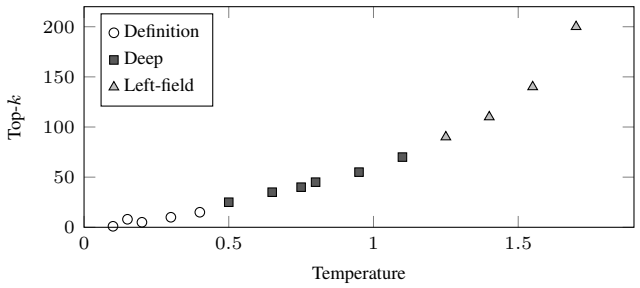


Fig. 3. The fifteen fixed sampling pairs. Low temperature and a small top-k tend toward definition questions. High temperature and a large top-k tend toward left-field questions. A seed assigns these pairs to names; it does not invent new pairs.

TABLE I
SAMPLING USED FOR EACH SPEAKING JOB. THE LOADED CONTEXT IS ALWAYS THE MODEL MAXIMUM.

Job	T	Top-k	Limit
Lecture paragraph	0.55	40	140 tokens
Student question	seat	seat	80 tokens
Private notes	0.20	8	280 tokens
Spoken answer	0.40	24	220 tokens

Chat requests are HTTP posts to `/api/chat` on `127.0.0.1:11434`. Image requests are posts to `/api/generate` on a host that defaults to `127.0.0.1:11435`. Speech is the platform `say` command, which uses the system speech synthesizer [11]. The process identifier is kept so Stop, or a raised hand, can kill that process.

The loaded context size is the model maximum on every chat call, including student questions. Ollama treats a change of `num_ctx` as a reason to reload the runner. Reloading between a student and the lecturer would insert a stall into a live talk. Prompt length is controlled separately, by trimming the transcript before it is sent. The lecture trim is three quarters of the model maximum once that maximum exceeds 2,048 tokens. Answers and private notes may use the full window. Figure 2 plots that split. On the `gemma3:1b` server used with this hall, the reported maximum was 32,768, so the lecture budget was 24,576 tokens and the answer budget was the full 32,768. Question prompts are further capped at the smaller of the lecture budget and 8,192 tokens.

Table I is the cast of jobs rather than the cast of people. The lecturer's public voice is warmer than the private notes. The notes are a low-temperature pass whose only audience is the next prompt. They are shown in the window and are not spoken. The spoken answer is a second call, still on the same model, with the notes and the question in front of it.

IV. A SEEDED ROOM

Fifteen names and fifteen English system voices are fixed. Fifteen sampling pairs are also fixed, and they are the only source of student personality. Figure 3 plots them. Five pairs are marked as definition seats, six as deep seats, and four as left-field seats. Temperature runs from 0.10 to 1.70. Top-k runs from 1 to 200. No two pairs share a temperature or a top-k. Spoken rate for a student is $138 + 28T$ words per minute, rounded. The lecturer is separate: voice Alex, 156 words per minute, temperature 0.55, top-k 40.

A seed is either a positive integer, with zero rewritten to one, or an arbitrary phrase hashed with FNV-1a. From that integer the hall shuffles pairs onto names, shuffles voices onto names, draws a speech duration of 300 plus a uniform draw from 0 through 300 seconds, and draws a hand count of 3, 4, or 5. The same seed rebuilds the same room. Changing the phrase changes the room. The window can be closed and reopened without drifting the cast.

Hands are not a Poisson process. The first legal second is 45. The last legal region ends 20 seconds before the speech budget. The span between them is cut into as many slots as there are hands, and one second is drawn inside each slot, forced to stay strictly after the previous hand. The student is taken from a shuffled pool, wrapping after fifteen hands when a question rate asks for more interruptions than there are students. With probability $1/5$ the question kind is replaced by a uniform

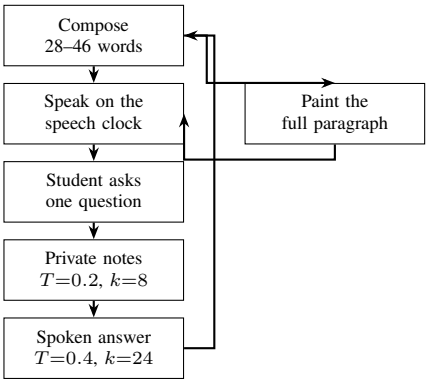


Fig. 4. Each lecture paragraph is spoken and illustrated together. A hand ends the utterance, the student asks, and the lecturer thinks before answering. The loop then resumes on the same topic.

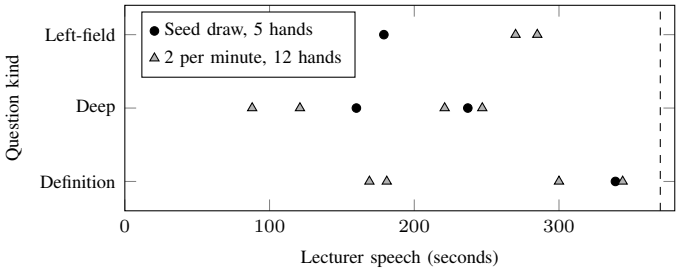


Fig. 5. Hands for seed 42. The lecture speech lasts 370 seconds, marked by the dashed line. The seed draw places five hands. Setting two questions per minute on the same seed places twelve, at 61, 88, 111, 121, 169, 181, 221, 247, 270, 285, 300, and 344 seconds. Kinds still come from the seated student, with a one-in-five replacement.

draw over the three kinds. Otherwise the student asks in the kind written on the seat.

The prompts ask for one question. A definition question asks what a term means. A deep question asks for an assumption, a hard case, a consequence, or a tension. A left-field question is required to ignore the topic and to be impossible in ordinary life, and the lecture transcript is withheld from that prompt so the subject is not sitting in the context. Nothing in the sampler can force the model to obey those instructions. The kind is a request, checked by ear, not a classifier score.

V. Speech is the clock

Figure 4 is one cycle. The lecturer writes a paragraph of 28 to 46 words. The width of that band is the seconds remaining until the next hand, capped at 18, doubled, and then clamped. Short gaps still produce a speakable paragraph rather than a fragment. The text is spoken with `say`. The clock advances by the audio that actually elapsed.

If the hand's second arrives during speech, the `say` process is killed. The transcript stores the prefix of the paragraph that fits the elapsed time at 156 words per minute, not the paragraph the model drafted and the room did not hear. The student then speaks the question in that student's voice and rate. The lecturer's notes are generated, then the spoken answer. Wall-clock time spent in questions and answers is outside the five-to-ten-minute speech budget. The budget counts lecturer exposition only. After the answer, the next paragraph is told that a question just happened, so the lecture continues instead of repeating the answer.

Stop sets a cancel flag and silences any current utterance. Closing the window does the same. There is no second lecturer and no overlapping student speech. One mouth is a product constraint: a rehearsal that talks over itself is hard to audit.

VI. Questions per minute

Operators asked to raise the number of questions per minute without editing the source. The flag `--questions-per-minute` replaces the seed's draw of 3, 4, or 5. The count is the rate times the speech duration in minutes, rounded, and at least one. It is clamped to the number of distinct seconds between second 45 and 20 seconds before the end, so a very high rate cannot schedule two hands on one second or a hand after the lecture has ended.

Figure 5 is seed 42 under both policies. The seed draws a 370-second lecture and five hands, at 102, 160, 179, 237, and 339 seconds. Two questions per minute on that same duration is $\mathrm{round}(2 \times 370/60) = 12$ hands. The seating and the speech length stay put. Only the slot boundaries move. Figure 6 shows the same function over seeds 1 through 500 and over four explicit rates. Durations fill the legal 300–600 second range. Default hand counts were 188 lectures with three hands, 149 with four, and 163 with five. The right-hand curves are straight because the flag is a rounded proportion of the speech length, until a rate would demand more seconds than the legal span contains.

The series in Figs. 5 and 6 are not a sample of model behavior. They are the scheduler, replayed outside the window and checked, for seed 42, against the Rust function the application calls. `docs/data/replay_scheduler.py` regenerates the files.

VII. Pictures of each paragraph

When a paragraph is ready, the hall starts an image request and starts speaking. The prompt is the topic, an instruction to draw one scene and not to draw letters, captions, or signs, and then the entire paragraph. The request asks for a 512×512 image, a seed derived from the room seed and the line number, and a ten-minute keep-alive so the image weights stay resident between paragraphs [6]. The returned field is a base64 PNG. The window decodes it and shows it on the lectern and in the call log. Student questions and spoken answers are not illustrated.

The image call does not fail the lecture. A refusal becomes a short note on that passage, and speech continues. This mattered in deployment. The Ollama application already running on port 11434, version 0.40.2, answered image requests with the error that image generation models are not currently supported. The same request to Ollama 0.32.5, bound to `127.0.0.1:11435` and pointed at the existing model di-

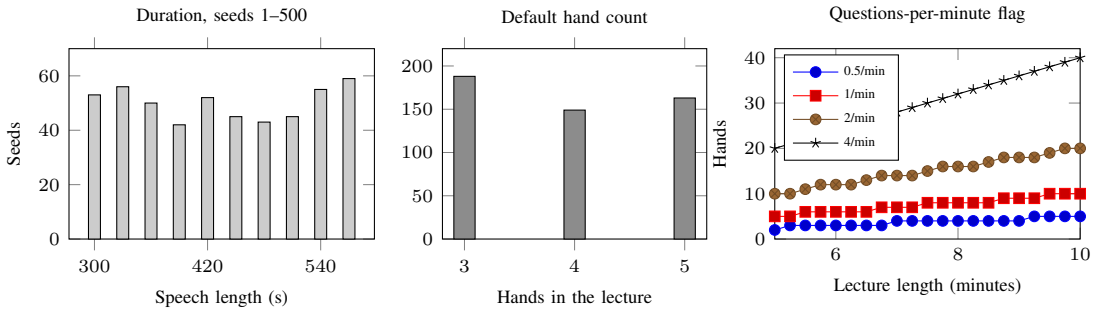


Fig. 6. Operating envelope of the scheduler, replayed from the same function the hall uses. Across seeds 1 through 500 the spoken lecture is spread through 300–600 seconds, and the default hand count is 3 (188 seeds), 4 (149), or 5 (163). The right-hand plot is the command-line rate after rounding and after clamping to the seconds between the first legal hand and the end of the lecture.

rectory, returned a PNG. The hall's image host is therefore a separate flag from the chat server, which stays on port 11434.

One development call, not a benchmark, was recorded on an Apple M3 Ultra. Model `x/z-image-turbo`, 512×512, seed 42, through 0.32.5: the server reported 11.12 seconds end to end, of which 2.58 seconds was weight load. Later paragraphs in a lecture reuse the keep-alive and skip that load. The hall joins the image thread before it composes the next paragraph, so this process does not overlap a chat call with an image call. The two servers can still contend for the GPU. That join replaced an earlier design that prefetched the next paragraph during speech. Prefetch won a little latency and lost the picture, because the two models were scheduled against each other.

Z-Image-Turbo is used because it is the experimental Ollama image model already present on the machine, it is small enough to sit beside a 1B chat model, and its weights are released for commercial use [6], [10]. The prompt forbids rendered words because the picture is an annotation of a paragraph the lecturer is already saying. Readable text inside the image would compete with the transcript.

VIII. What the Command Line Must Carry

Moving the controls out of the window made a lecture a single reproducible command. The topic is required. The seed is optional and, if omitted, is drawn from the system random source and printed, so the room can be repeated later. The chat model defaults to `gemma3:1b`. The image model defaults to `x/z-image-turbo`. The image host defaults to `127.0.0.1:11435`. The question rate is optional; omitting it restores the three-to-five hand draw. A run with the defaults and a faster audience is:

```
cargo run -- --topic "ocean tides" --seed 42
--questions-per-minute 2
```

The double dash is Cargo's, not the hall's. Unknown flags are rejected before the window opens.

IX. Lessons from Operating It

One context size, two budgets. Reloading a local model between turns is visible in a spoken demo. Keeping `num_ctx` constant and trimming prompts was the change that made speaker turns feel like turns.

Kill the audio, then repair the transcript. Stopping a synthesizer does not by itself tell the model what the room heard. The word-prefix rule is crude, and it is good enough that the next prompt does not answer a sentence that was never played.

Image generation is a server-version problem, not a prompt problem. The first image failures were immediate HTTP errors, before any sampling. A second Ollama, pinned to a release that still implements the published image API, was the fix. Sharing the model directory worked. Sharing the port did not.

Do not prefetch text while a picture is in flight. Both calls are large. Serializing them behind the speech of the current paragraph kept the lecture understandable and the illustration attached to the right line.

A seed has to cover the room, not only the sampler. Temperature without the voice, the duration, and the hand times does not replay a lecture. Operators compare runs by seed, topic, and question rate.

X. Limits

The hall speaks only through macOS `say`, and only in the installed English voices. It has not been timed on Windows or Linux, and the image path depends on an Ollama release that current application builds reject. Question kind is prompted, not verified. Left-field questions are specified to be unreal; they are a stress on the lecturer, not a source of facts. The five-to-ten-minute figure is lecturer speech, so a curious room runs longer on the wall clock. No study here measures whether a person learns the topic. The plots measure the scheduler and one image call.

XI. Conclusion

A fifteen-student lecture can run on one local model if personality is a sampling pair, if the clock is the audio that was actually played, and if illustrations are allowed to use a second server when the first server has dropped image generation. The seed makes that room repeatable. The question-rate flag changes how often hands rise without reseating the class. The

system is a rehearsal instrument for a single topic, with the limits of a local speech stack and of an image API that is still experimental.

REFERENCES

[1] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney, "AutoTutor: An intelligent tutoring system with mixed-initiative dialogue," *IEEE Transactions on Education*, vol. 48, no. 4, 2005.

[2] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3586183.3606763

[3] egui contributors, "egui: An immediate-mode GUI library for Rust," https://github.com/emilk/egui, 2026, accessed 10 October 2026.

[4] Ollama, "Ollama," https://github.com/ollama/ollama, 2026, accessed 10 October 2026.

[5] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018. [Online]. Available: https://aclanthology.org/P18-1082/

[6] Ollama, "Image generation (experimental)," https://ollama.com/blog/image-generation, Jan. 2026, accessed 10 October 2026.

[7] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rygGQyrFvH

[8] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024.

[9] Gemma Team, "Gemma 3 technical report," arXiv:2503.19786, 2025. [Online]. Available: https://arxiv.org/abs/2503.19786

[10] Tongyi MAI, "Z-image: An efficient image generation foundation model with single-stream diffusion transformer," arXiv:2511.22699, 2025. [Online]. Available: https://arxiv.org/abs/2511.22699

[11] Apple, "NSSpeechSynthesizer," https://developer.apple.com/documentation/appkit/nsspeechsynthesizer, 2026, the hall speaks through the `say` command. Accessed 10 October 2026.