Episode Book

EPISODE #6659

Updating Speech Models Weekly Without Forgetting

PUBLISHED	17 November 2026 at 11:21 UTC
DURATION	24:18
LISTEN	myweirdprompts.com/episode/incremental-asr-fine-tuning-80298/
HOSTS	Corn, Herman, Dorothy, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.35
SOURCES	22 cited by the research (listed at the end)
BOOK GENERATED	17 November 2026 at 14:42 UTC

SYNOPSIS

Frequent, lightweight updates to a speech recognition model require separating list problems from weights problems. Runtime biasing frameworks like BR-ASR handle up to 200,000 custom vocabulary entries without touching model weights. When accents or systematic errors do require training, AWS AI Labs' sequential-editing recipe uses ten epochs at one-tenth the learning rate. To prevent catastrophic forgetting across dozens of cycles, researchers use adapters, wake-and-sleep consolidation, TIES-merging, and SVD-rehearsal requiring just one utterance per past task.

TRANSCRIPT



Corn

Every AI assistant Daniel asks gives him the same answer, and it's the wrong one. That's today's whole premise.



Herman

I'm Herman Poppleberry, and yes, I've been waiting all morning for this one.



Corn

Daniel's prompt is about rhythm, not recipes. He fine-tunes a speech recognition model — his Parakeet, trained on about seventy hours of his own voice — and he wants to add a little vocabulary, run a short training round, push out the weights, and do it again next week. No word-error-rate benchmark every epoch. He benchmarks separately. What he wants is the fine-tuning strategy for a very light, very frequent update cycle — and every helper he asks hands him the full treatment instead, the ceremonial run, the machinery, the epoch-by-epoch validation.



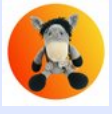
Herman

The "complete the twelve-week training montage" answer.



Corn

Right. His actual question is whether there's a formula built for cadence rather than for one big careful run. But before we get to the recipe, there's a prior question — whether most of these updates should be training runs at all.



Herman

And that's where I want to start, because the field has a name for what Daniel's describing. Continual learning, sometimes lifelong learning, for ASR. There's a real literature — adapters, low-rank factorization, weight merging, replay, reduced learning rates. It's not exotic. But the literature starts by asking something Daniel's question skips past: when you say "add a little new vocabulary," which of two problems are you holding?



Corn

I'd say both, depending on the week.



Herman

There's a list problem — names, product terms, jargon that changes with each release, the new tool that didn't exist when the model was pretrained. And there's a weights problem — accents, code-switching, systematic error patterns, formatting conventions. They look the same on the transcript, because both show up as "the model got my word wrong." But they're different problems with different tools.



Corn

And the list problem never needed a training run.



Herman

That's the thesis I'd defend today. The list problem is a runtime, config-push problem and should almost never touch the weights. The weights problem is where the incremental-training literature actually applies — and its answers are more specific than the generic advice Daniel keeps getting. The practitioner literature lands on this split independently of the research literature, which is always a good sign. Gladia, Amazon Science, the biasing papers — three separate lines all arriving at "don't retrain, bias at runtime."



Corn

Gladia being the speech-to-text API company.



Herman

Right. Their guidance on custom vocabulary is practically a table with two columns. Runtime vocabulary lists: fast to deploy, included in the base rate, best for dynamic vocabularies that change with product releases. Model fine-tuning: requires labeled domain audio, slower, heavier. Their recommendation is to update the custom vocabulary list on the same cadence as your product release cycle, test each update against a golden test set using keyword error rate rather than overall word error rate, and — this is the rule I like most — prune the terms the base model already transcribes correctly.



Corn

Prune before you push. The list shrinks as the model improves underneath it.



Herman

And it keeps the biasing from fighting the model. If you bias toward a word the model already knows, you're not helping, you're just creating opportunities for false positives. Amazon Science made the same argument from the other end — they wrote a whole piece called "Teaching speech recognizers new words without retraining," for exactly the case where the operational vocabulary keeps changing. New names appearing, users adding contacts to address books. Their phrase for the alternative is "prohibitively time-consuming" — retraining on new-word datasets every time the vocabulary moves.



Corn

A phrase like that is doing a lot of load-bearing work. Prohibitively for whom, though? Daniel's runs are short.



Herman

For a company shipping to thousands of users, sure. For Daniel, "prohibitively" is more like "unnecessarily." The research version of the same argument is BR-ASR, from Interspeech last year — a retrieval framework that pulls bias candidates and integrates them into the ASR system with no fine-tuning at all. The numbers are the striking part. It scales to two hundred thousand bias entries with about zero point three percent absolute WER degradation on general speech, a bit under three percent on the bias words themselves, a pruning rate above ninety-nine percent, and about twenty milliseconds per query.



Corn

Two hundred thousand entries. Daniel's list is maybe forty Hebrew tech terms and a street name.



Herman

Which means his list problem has headroom for about 199,960 more terms before the technique even notices. There's also streaming CTC word-spotting biasing work from this year that's explicitly designed around not modifying the acoustic model at all — no additional training, deployable as-is. That's the whole pitch.



Corn

So the first answer to "what's the best formula for tiny incremental fine-tuning" is: if the thing you're adding is shaped like a list, the light frequent update cycle is a config push. Save the training runs for what a list can't fix — an accent, code-switching, a systematic mis-transcription that shows up fifty times a day.



Herman

And now we can talk about those runs properly, because the literature on them is consistent on one architectural point. Freeze the base, train a small add-on.



Corn

You have a favorite.



Herman

I have findings, Corn. PELE, from Interspeech a couple of years back, compared the parameter-efficient approaches for extending a multilingual model, and the result was blunt — methods that adjust weight parameters or input features were significantly inferior to inserting a lightweight module between layers. An adapter. In their experiments, adapters beat LoRA-style weight-parameter methods for this job.



Corn

That cuts against the default. LoRA is what every tutorial hands you.



Herman

It's the default because it's convenient, not because it's best for repeated rounds. And under repetition it gets worse — hold that thought for part two. The template paper for the freeze-the-base approach is OWSM-Biasing: take a frozen OWSM v3.1, bolt contextual biasing on top, and it works even with a small dataset. Small matters here, because Daniel's increments are small by design.



Corn

And when a training round does happen, what does it look like? Round numbers, Herman. He wants the formula.



Herman

The most transferable recipe I know comes from the AWS AI Labs sequential-editing paper, on a Conformer CTC model. Initial training: sixty epochs at a learning rate of five times ten to the minus three. Then, for every addition of new data, ten more epochs at a tenth of that — five times ten to the minus four. That's it. That's the formula.



Corn

Ten rounds, one tenth the learning rate.



Herman

The ten-times learning-rate drop is the whole trick, and nobody says it loudly. Every incremental-round paper I've read converges on roughly this shape: short round, low rate, continue from the previous weights rather than restarting from base. The fancy machinery — the merging, the subspace restriction — is all scaffolding around that one idea, so the small step doesn't undo the big one.



Corn

And now, a brief return to Dorothy's History of Toast.

D

Dorothy

Hello dear, it's Dorothy, with Dorothy's History of Toast.

D

Dorothy

In nineteen ninety five, the physicist Robert Matthews published a paper explaining why toast really does tend to land butter side down, mostly because of the height of the table.

D

Dorothy

And it won him an Ig Nobel Prize the next year. So there you are, sweetheart. Even the butter has a plan and it isn't yours.

D

Dorothy

Anyway, I've got the kettle on. Be good.



Herman

Mum and her toast physics. Delighted as always.



Corn

The recipe works — until you run it sixty-six times in a row and the model starts forgetting what it knew.



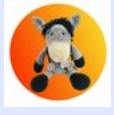
Herman

Which is precisely part two. The forgetting problem is real, and the literature is honest about it in a way the generic AI advice never is. The Pacific Indigenous ASR work from this year's Interspeech says it in one line: while LoRA adapts well initially, it suffers from catastrophic forgetting during sequential learning. That's a direct warning for exactly Daniel's loop. Small doesn't mean safe.



Corn

A small step in a bad direction is still a step in a bad direction.



Herman

So the recipe is necessary but not sufficient. Now, the pattern I find most interesting — the closest thing to a formal statement of Daniel's rhythm — is what Ugan, Pham and Waibel call wake and sleep. Two phases. A factorization phase, where new knowledge gets learned into multiple scattering low-rank adapters. Then a centralization phase, where those adapters are merged back into the model. They frame it on the human sleep cycle, which is a bit precious, but the engineering is sound: small frequent updates, periodically



Herman (continued)

consolidated. Rehearsal-free, works without the original training data, language-agnostic.



Corn

I nap, therefore I consolidate. Finally research in my image.



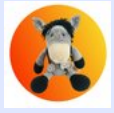
Herman

The principled version of "can't break anything" comes from the same lab, actually — PECL, from Interspeech this year. You take the singular value decomposition of each pretrained weight matrix, split it into a head and a tail — the big singular values versus the small ones — and restrict your adaptation to approximate rotations in the low-energy tail. The dominant directions of the pretraining barely move, because you've only allowed movement in directions the pretraining didn't care much about. For subsequent tasks, the rotations get combined by weight averaging.



Corn

So you're only allowed to redecorate the rooms nobody uses.



Herman

That's a better sentence than the paper's. Then there's the merging family. Task arithmetic, TIES-merging — the AWS paper uses this on sequential accent addition: fine-tune on the new data source, then merge the resulting task vector back into the accumulated model. On CommonVoice English multi-accent, that gave a fifteen percent WER reduction over the plain fine-tuning baseline, versus six percent for prior lifelong-learning approaches — with no access to the old datasets and no extra parameters at inference.



Corn

Merging as the cheap forgetting fix.



Herman

Cheap, and — there's a follow-up paper that says this — heuristic. Inverse Hessian regularization, from ICASSP this year, points out that weight averaging ignores the underlying loss landscapes. So they add a Kronecker-factored inverse-Hessian approximation of the previous task to the merge step, so the model moves mainly in directions less harmful to past performance. Still lightweight, no memory of old data.



Corn

And replay? The boring answer.



Herman

Boring, reliable, and now nearly free in memory. The SVD-rehearsal work from this year: fine-tune on the new task, then decompose the changes in the linear layers and retrain only gating vectors on the singular values, controlling how much of the update you accept. The headline is that it works even when you're limited to a single utterance per previous task. One sentence of the old domain is enough rehearsal to matter. For Daniel, that's a rounding error of storage.



Corn

One utterance per old topic. That's the kind of number you build a whole habit around.



Herman

Now — the study I'd hand Daniel first. KIT and CMU ran sixty-six sequential learning cycles over sixty-six computer science lecture talks. They pulled new words from the slides — about thirty-nine hundred of them, roughly twenty-two hundred unique — used a memory-enhanced model to find utterances containing those words, collected those utterances as the adaptation set, trained low-rank adapters, and repeated. Sixty-six times.



Corn

That's a year of Tuesdays.



Herman

New-word recall climbed above eighty percent as words recurred. General performance was preserved — no catastrophic forgetting across all sixty-six cycles. And the comparison is the part that matters. The baselines that just kept adding words to the bias memory, without training, saw WER increase consistently — more than twelve percent and seven percent relative respectively — because they had to distinguish between more and more words in memory, producing false positives. The continual-learning variant held WER flat against a four point three nine percent baseline.



Corn

So both of Daniel's instincts are validated, just not simultaneously. The list grows, WER degrades. The weights get trained, WER holds flat.



Herman

That's the cleanest experimental statement of the tradeoff I've seen. And their stated failure modes read like a checklist for a frequent-update loop — errors in word extraction propagate through the cycles; morphological variants of known words increase false positives; similar-looking words cause trouble. Their example: when ISCA is learned followed by ISA, which is written similarly, a small supervised intervention could help.



Corn

ISA walks in wearing ISCA's coat.



Herman

Now the practitioner side, and this is where it gets personal, because the repo documenting this is Daniel's own. ASR-Recursive-Improvement-Loop — his daily loop of mining transcript history, confirming errors, recording new sentences, fine-tuning, shipping to the app. It's the most concrete public description of a small-frequent-release ASR workflow I know of.



Corn

We get the receipts straight from the source.



Herman

The rules in it are worth quoting almost verbatim. "Train small and often. Continue from the last weights on new recordings plus a replay set, rather than retraining from the base model. A recent run here took about eighteen and a half minutes end to end." "Mix in replay data. Old-domain audio in every run limits forgetting." "Fix one pattern at a time. Once vocabulary is strong, take one error pattern, record about 150 sentences for it, train, and check the next day's history."



Corn

A hundred fifty sentences per pattern. That's an evening of talking, not a data-collection campaign.



Herman

"Include contrast sentences. When fixing a format, also record the neighbouring form that must not change, or the fix overcorrects." That one's quietly brilliant — it's the replay idea applied within a single fix. "Only train what the audio can decide. If only the speaker's intent separates two forms, training will not fix it." "Record the model version on every transcript. Without it you cannot tell which version fixed what." And "hold out a test set, term-disjoint from training, for periodic benchmarks, because daily history reading is a hint and not a measurement."



Corn

Version stamps on every transcript. The one piece of infrastructure that makes a frequent-release regime debuggable at all.



Herman

The repo is also honest about its own gaps, which I respect. Incremental runs currently publish without an evaluation step — which is exactly Daniel's point, he benchmarks separately — failing terms are fed back by hand, and the error list, in its own words, "will not shrink in a straight line, because a growing vocabulary brings new failures with it."



Corn

We did an episode ages ago on what comes after fine-tuning — deployment, the part everyone skips. Everything in that repo is the deployment part. The training run is eighteen minutes; the versioning is forever.



Herman

On architectures — which ones suit this workflow. CTC and Conformer-CTC with adapters, first. CTC is what makes streaming word-spotting biasing possible without touching the acoustic model, and the AWS work is a Conformer CTC. Then attention encoder-decoder models with a bias list — there's work scoring candidate entities using multi-token prediction logits, no extra entity encoders, no cross-attention layers, up to fifty percent relative improvement on named-entity WER. Non-autoregressive with dynamic vocabulary: DYNAC puts the dynamic vocab into intermediate layers of a self-conditioned CTC and cuts real-time factor by eighty-one percent at a tenth of a point of WER. Then the frozen-foundation-plus-external-biasing template, OWSM being the example.



Corn

And the awkward one?



Herman

Speech LLMs. They hold the state of the art on general ASR, but they still fail to accurately transcribe bias words that appear rarely or never in the training data — the very words a personal model cares about most — and their context windows limit how many bias entities you can inject at once. For Daniel's use case, the fancy architecture is the wrong one.



Corn

The cutting edge can't remember your name. There's a whole field in that sentence.



Herman

One more thing before we land, because it cuts against everything above. The earnings-call study this year — adapting Whisper-small to Earnings22 — found the opposite of the forgetting story. Adaptation improved held-out financial speech rather than degrading anything, which made regularization counterproductive. Their conclusion: assess task complementarity before applying continual-learning constraints.



Corn

Meaning: if the new domain is a friend of the old domain, don't build walls between them.



Herman

And adjacent domains are usually friends. General speech and financial jargon overlap heavily, the way Daniel's general English and his tech vocabulary overlap. So the right amount of safety machinery is an empirical question, not a settled one — measure whether forgetting is happening before you pay to prevent it.



Hilbert

I don't retrain mine at all. I keep a list. Paper, in the green notebook, the one with the elastic band. Every word the phone's dictation gets wrong goes on it, and when the list reaches a certain length — the length is my business — I read it aloud into a fresh recording. Once, slowly. Then I cross it out and start the next page.



Herman

You record the list itself?



Hilbert

The whole list, Herman, in one go. You'd like the way you two say words like "adapter," like the word comes with a certificate. Anyway. The model learns from it, and the error rate goes down. There's a street near my sister's place, Vechtweg, took the phone four months to stop rendering it as "fetch bag." After the third reading it comes out right every time.



Corn

You've never actually checked whether the phone improved. That's a different recording from the transcript it produces.



Hilbert

Checking would spoil it. Corn's mic is a touch hot, by the way, I'll pull it down at the edit. The pages are dated and kept, every one, in a shoebox, because someday somebody will want proof that the words were real.



Corn

Proof for whom, Hilbert?



Hilbert

Whoever asks. Vechtweg was real. Fetch bag never was.



Herman

The recordings do go into your voice profile on some of these systems. So the ritual isn't nothing — you may be feeding it audio either way.



Hilbert

Then I've been doing it right for the wrong reasons, which is my usual method. Two seconds on the levels and we go again.



Corn

Whether anyone has actually measured the cadence, though. That's the gap I keep landing on. There's no published study of very light, very frequent ASR update cycles as engineering practice — nobody's measured daily ten-minute fine-tunes against weekly hour-long ones. The sixty-six-cycle experiment is the nearest thing, and its data volume isn't framed as deployment cadence. Daniel's loop is, as far as I can tell, the experiment.



Herman

Which means the version-stamped transcripts aren't bookkeeping. They're the dataset.



Corn

If you take one thing from today: "add a little new vocabulary" is usually a list, and a list is a config push, not a training run. The full treatment every assistant hands Daniel is a solution to a problem he mostly doesn't have.



Herman

And when the weights do need to move — ten epochs, a tenth the learning rate, replay mixed in, one pattern at a time.



Corn

That's the show. Thanks as always to our producer, Hilbert Flumingtop, for holding the levels and the notebook.



Herman

If today's episode earned its eighteen and a half minutes, leave us a review — it's how other people find the rabbit holes. If you want more of this, try episode five, Fine-Tuning ASR For Maximal Usability; episode eleven, How Does Fine Tuning Work Anyway; and episode nine, Benchmarking Custom ASR Tools - Beyond The WER. This has been My Weird Prompts, the human-AI collaboration podcast.



Corn

We'll be back soon. See you then.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Sequential Editing for Lifelong Training of Speech Recognition Models, AWS AI Labs (Kulshreshtha et al.), arXiv v1, submitted 2024-06; retrieved 2026-11-17. [primary]**
   https://arxiv.org/abs/2406.17935v1

2. **Continuously Learning New Words in Automatic Speech Recognition, KIT / CMU (Ugan, Pham, Waibel et al.), arXiv v2; retrieved 2026-11-17. [primary]**
   https://arxiv.org/html/2401.04482v2

3. **Improving a personal ASR model by using it every day, repo README, last updated 2026-11-04; retrieved 2026-11-17. [primary]**
   https://github.com/danielrosehill/ASR-Recursive-Improvement-Loop

4. **Gladia, Custom vocabulary support in speech-to-text; retrieved 2026-11-17.**
   https://www.gladia.io/blog/custom-vocabulary-support-in-speech-to-text-how-to-teach-the-model-your-terms

5. **Amazon Science, Teaching speech recognizers new words**
   without retraining; retrieved 2026-11-17.
   https://www.amazon.science/blog/teaching-speech-recognizers-new-words-without-retraining

6. **Ugan, Pham, Waibel, Weight Factorization and Centralization for Continual Learning in Speech Recognition, Interspeech 2025; retrieved 2026-11-17.**
https://arxiv.org/abs/2506.16574v1

7. **Vander Eeckt & Van hamme, Parameter-Efficient Continual Learning for Automatic Speech Recognition, Interspeech 2026; retrieved 2026-11-17.**
https://arxiv.org/abs/2606.09342v1

8. **Vander Eeckt & Van hamme, Inverse-Hessian Regularization for Continual Learning in ASR, ICASSP 2026; retrieved 2026-11-17.**
https://arxiv.org/abs/2601.14751v1

9. **Vander Eeckt & Van hamme, Efficient Rehearsal for Continual Learning in ASR via Singular Value Tuning, IEEE TASLP 2026; retrieved 2026-11-17.**
https://arxiv.org/abs/2601.18266v1

10. **Vander Eeckt & Van hamme, CGaLore: Curvature-Guided GaLore, SLT 2026; retrieved 2026-11-17.**
https://arxiv.org/abs/2609.21336v1

11. **Vander Eeckt & Van hamme, Unsupervised Online Continual Learning for Automatic Speech Recognition, Interspeech 2024; retrieved 2026-11-17.**
https://arxiv.org/abs/2406.12503v1

12. **Liu et al., A Parameter-efficient Language Extension Framework for Multilingual ASR (PELE), Interspeech 2024; retrieved 2026-11-17.**
https://arxiv.org/abs/2406.06329v1

13. **Gong et al., BR-ASR: Efficient and Scalable Bias Retrieval Framework for Contextual Biasing ASR in Speech LLM, Interspeech 2025; retrieved 2026-11-17.**
https://arxiv.org/abs/2505.19179v1

14. **Sudo et al., OWSM-Biasing: Contextualizing Open Whisper-Style Speech Models, Interspeech 2025; retrieved 2026-11-17.**
https://arxiv.org/abs/2506.09448v1

15. **Sudo et al., DYNAC: Dynamic Vocabulary based Non-Autoregressive Contextualization, Interspeech 2025; retrieved 2026-11-17.**
https://arxiv.org/abs/2506.00422v1

16. **Tsai et al., Contextual Biasing for Streaming ASR via CTC-based Word Spotting, arXiv v2, revised 2026-05-19; retrieved 2026-11-17.**
https://arxiv.org/abs/2605.18222v2

17. **Novitasari et al., Contextual Biasing for ASR in Speech LLM with Common Word Cues and Bias Word Position Prediction, 2026-04-14; retrieved 2026-11-17.**
https://arxiv.org/abs/2604.12398v1

18. **Selvakumar et al., Peeking Into The Future For Contextual Biasing, 2025-12-19; retrieved 2026-11-17.**
https://arxiv.org/abs/2512.17657v1

19. **On & Beigi, Does Financial Speech Need Continual Learning Regularization? A Study of EWC and mini-KFAC on Earnings Call ASR, ICNLSP 2026; retrieved 2026-11-17.**
https://arxiv.org/abs/2610.13432v1

20. **Xiao et al., Continual Adaptation for Pacific Indigenous Speech Recognition, Interspeech 2026, revised 2026-06-17; retrieved 2026-11-17.**
https://arxiv.org/abs/2603.06310v2

21. **Guo et al., COALA: Robust Contextualized Speech-augmented Language Modeling for ASR, Interspeech 2026; retrieved 2026-11-17.**
https://arxiv.org/abs/2607.08117v1

22. **Show HN: Gemma 4 Multimodal Fine-Tuner for Apple Silicon, 2026-04-07; retrieved 2026-11-17.**
https://news.ycombinator.com/item?id=47680309