Episode Book

EPISODE #6366

Why Training Recipes Matter More Than Model Size

PUBLISHED	30 October 2026 at 12:44 UTC
DURATION	28:07
LISTEN	myweirdprompts.com/episode/speech-model-training-recipes-60639/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	openai/gpt-6.1-sol
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.25
SOURCES	18 cited by the research (listed at the end)
BOOK GENERATED	31 October 2026 at 05:58 UTC

SYNOPSIS

Fine-tuning speech recognition models often conceals massive architectural decisions inside ordinary setup notebooks. From checkpoint averaging and batch size dynamics to optimizer interactions, minor configuration details can drastically swing error rates without altering the underlying audio dataset. Exploring recent findings across Whisper and NeMo architectures, this discussion examines curriculum ordering, tokenizer reuse constraints, and why structured training recipes routinely outperform brute-force parameter scaling when adapting speech models to complex domains like code-switching.

TRANSCRIPT



Corn

Daniel has been fine-tuning Parakeet for code-switching, and the thing bothering him is why the instructions have so much influence over the result. He gathers a few more hours of speech, starts another run on Modal, and has Claude help supervise it. Recently, an NVIDIA example introduced averaging of selected checkpoints, and his results improved. Same general job, different recipe, better transcription. Why should the notebook have that much say?



Herman

The notebook chooses the path through training, then chooses which weights you actually deploy. Those are substantial decisions hiding inside something presented as an example.



Corn

My objection begins with the word example.



Herman

Mine begins when the example quietly becomes production. I'm Herman Poppleberry, and I think Daniel has noticed something important about the interface. Dataset in, weights out describes what the process consumes and produces. It says almost nothing about what happens between them.



Corn

His runs also keep changing underneath him. More audio, slightly different coverage, another chance to fix the previous mistakes. He wants those runs to accumulate knowledge rather than merely accumulate bills. I suspect the averaging discovery matters because it improved the result without requiring him to collect another afternoon of speech.



Herman

It also means the best result of a training run might never have appeared as an individual checkpoint during that run.



Corn

A GitHub search for Whisper fine-tuning returned nine hundred forty-four repositories. That sounds less like a standard procedure and more like nine hundred forty-four people deciding the previous instructions needed work.



Herman

Some are solving different problems outright. One supports training without timestamp annotations. Another is marked work in progress. Another hasn't been updated since twenty twenty-three. Their titles can look interchangeable while their assumptions about supervision, hardware, and model versions differ.



Corn

Three years of stability, or three years of abandonment. Same badge.



Herman

The same base model and audio can produce different weights depending on learning rate, ordering, schedule, precision, and which parameters are allowed to change. Even the phrase same data needs scrutiny. Are long recordings filtered? Are numbers written as digits or words? Are batches formed by utterance count or audio duration?



Corn

I would have put those under preparation, not fine-tuning.



Herman

They influence the gradients, so the boundary is administrative. If one notebook silently drops the hardest recordings, it has changed the training problem before the optimizer begins.



Corn

Then comparing notebooks by the final error rate can reward the notebook that quietly avoided the job.



Herman

Potentially. What would convince you a recipe was better, Corn, if you had two runs and one reported a lower error?



Corn

The actual exported models transcribing the same unseen recordings, with the same scoring. Then I'd look at the mistakes I care about. A code-switching model could improve ordinary English while still replacing every borrowed term with a respectable English word. I'd want to hear that before congratulating the progress bar.



Herman

That distinction will matter with averaging. Better selection, better weights, and different evaluation can all move the reported number. You need to know which happened.



Corn

You are about to fall in love with the averaging mechanism.



Herman

A little. You are about to defend keeping the old checkpoint.



Corn

It already exists. Excellent manufacturing standard.



Herman

Keep it, then. We can test both. That's one rare occasion when your attachment to the old version makes the experiment stronger.



Herman

NVIDIA's own fine-tuning documentation gives a learning rate from one ten-thousandth to one hundred-thousandth. That's a tenfold range, one order of magnitude. It's sometimes described as two orders because the exponents differ, but the ratio is ten.



Corn

Even the recipe needs arithmetic supervision.



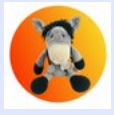
Herman

The important point is that the vendor doesn't provide one setting. Its newer agent instructions suggest one ten-thousandth for large or mixed runs. For small domain adaptation, especially below roughly twenty hours of valuable target audio, they suggest three hundred-thousandths.



Corn

Valuable is doing actual work there. Twenty hours containing the same easy phrases won't be equivalent to twenty hours containing the switches that keep breaking.



Herman

Correct, and later refinement or unstable training gets a lower suggestion, around one hundred-thousandth or below. Those recommendations depend on corpus size and training history. Both change in Daniel's workflow.



Corn

Then his next run isn't automatically a repeat with more fuel. The new recordings might change which approach makes sense.



Herman

Also, are these fresh starts from the pretrained model, or continued training from his previous adaptation? Those are different experiments. We shouldn't infer that from the fact that he runs again every few days.



Corn

I would compare recipes from the same starting weights. Otherwise the apparent improvement could be the earlier run's work.



Herman

For a clean comparison, yes. Continuing from a previous checkpoint can be useful operationally, but it carries its history. You cannot treat that history as invisible.



Corn

The same vendor offers a tenfold learning-rate range, then conditional advice within it. Daniel wants to know how much the choice of recipe changes the outcome, and already the vendor's answer depends on what happened last time.
Ivan: Call sign Marrow. Two seven, four one. Nine three, zero eight. Two seven, four one. The garden marrow is bitter but the kettle sings at dusk. The marrow counts its own seeds twice. Close on Marrow.



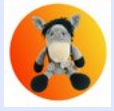
Herman

Wrong number.



Corn

What on earth just came through?



Herman

Scheduling adds another dependency. The documentation describes fifty to a hundred epochs as typical for domain adaptation. The agent instructions instead say to schedule by optimizer steps. An epoch grows when your corpus grows, so repeating an epoch count



Herman (continued)

can quietly increase the amount of training.



Corn

If Daniel adds audio and keeps the epochs constant, he's changed the data and the update budget together. That makes the comparison messy before anyone touches the learning rate.



Herman

Steps make the update budget more explicit. They don't settle everything, because batch composition matters, but they separate questions that an epoch count bundles together.



Corn

I'd keep exposure to the old recordings visible too. Continuing on only the new material could teach the model that this week's vocabulary is the entire language.



Herman

Rehearsal data can help retain old capabilities. Its composition matters as much as the headline ratio. Broad clean speech tests a different kind of retention from older examples of Daniel's own switching patterns.



Herman

There are architectural choices before any of this. NeMo can initialize from a published model, a local model package, or selected components. You can load the encoder and preprocessor while excluding incompatible output components.



Corn

Which means two notebooks can both say fine-tune Parakeet while one preserves substantially more of the original model.



Herman

Changing the tokenizer can require rebuilding the corresponding decoder or output components. The precise components depend on the architecture. You aren't simply nudging the same parameter set anymore.



Corn

I wouldn't replace the tokenizer just because the vocabulary contains unfamiliar names. Existing subword pieces can often spell them already. I'd first ask whether the model has learned to choose those pieces in the right context.



Herman

That restraint fits the Riva tutorial's advice to reuse the pretrained tokenizer below about fifty hours of training data. A new tokenizer creates things you must learn again with a small corpus.



Corn

A vocabulary renovation with no budget for the furniture.



Herman

Would you ever change it for code-switching?



Corn

If the existing representation mangled the target writing system or made important words absurdly fragmented, perhaps. But I'd want that demonstrated. Unfamiliar to the model isn't the same as impossible for the tokenizer.



Herman

Especially because multilingual transcription errors can come from different places. The acoustics may be recognized well, but the model favors a common word from the other language. Or the pronunciation really is acoustically unfamiliar. The intervention should follow the mistake.



Herman

Recipe effects can be large. In the Vividh-ASR work on Whisper, changing learning-rate timing alone improved global word error by twelve absolute percentage points. That means twelve points off the error rate, not a twelve percent relative reduction.



Corn

That is a considerable amount of accuracy to leave inside the timetable.



Herman

Their reverse multi-stage approach used larger updates early and a hard-to-easy curriculum. A parameter-efficient model with two hundred forty-four million parameters matched or exceeded conventionally fine-tuned counterparts with seven hundred sixty-nine million.



Corn

Roughly three times the parameters, beaten by getting the training order right.



Herman

Under those experiments, yes. It doesn't promise that gain for Daniel's Parakeet. Whisper has a different architecture. It does show why treating schedule as a minor preference can be expensive.



Corn

What I find useful is the direction. People often save the difficult material for later, as though the model needs encouragement first. Here the difficult examples helped establish the adaptation, then the easier ones refined it.



Herman

Their representation analysis also suggested effective schedules concentrated change in the decoder while preserving the pretrained encoder's acoustic structure. The recipe influenced where adaptation happened.



Corn

For code-switching, preserving strong acoustics while changing which word sequences get favored sounds attractive. Though I wouldn't assume every accent or language pair wants the encoder left alone.



Herman

Nor would I. That's the limit of transferring the intuition. Parakeet's prediction and joint components aren't Whisper's text decoder. The diagnostic question transfers better than the exact intervention.



Herman

Optimizers interact with batch size too. In a Whisper Small LoRA comparison, AdamW beat Adam at a batch size of eight, with word error around eight percent versus eight and a half.



Corn

Half a point might tempt someone to declare AdamW the winner and publish the notebook.



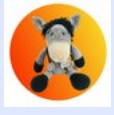
Herman

At a batch size of four, the ranking reversed. Adam was around six percent error, while AdamW was around eight. A recommendation calibrated to one batch setup didn't survive the other.



Corn

The GPU ran out of memory and changed the scientific conclusion.



Herman

Well ... the experiment changed more than memory pressure. Smaller batches alter gradient noise and the number of updates for a given corpus exposure. Accumulation changes some of that again. The lesson is to record the effective batch and update



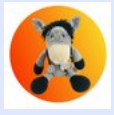
Herman (continued)

budget, not merely the number the notebook calls batch size.



Corn

You installed something last week because its memory use looked better.



Herman

Its memory use was better.



Corn

The working part was less competitive.



Herman

I got it working. Eventually. That is why I favor recording environment versions before enthusiasm edits them.



Corn

I assume adapter choice becomes another enormous search space.



Herman

It can, although the fashionable distinction isn't always the valuable one. In a single-speaker dysarthric ASR comparison, LoRA and DoRA were effectively tied across repeated seeds. The simpler LoRA was a reasonable choice.



Corn

One fewer syllable to maintain.



Herman

Quantized LoRA performed worse on every seed and both tested base models, without saving memory at that scale. Full fine-tuning remained more accurate. A targeted feed-forward LoRA configuration, though, came within about two-thirds of a character-error point using only a small fraction of the storage.



Corn

So cheap inference artifacts and cheap training aren't automatically the same objective. Nor is low memory guaranteed by a technique's name.



Herman

Right … no, let me narrow that. Quantization can save substantial memory in appropriate settings. That particular result says measure the actual configuration, especially when adapters, activations, and framework overhead dominate. It doesn't indict quantization generally.



Corn

Daniel needs a shortlist of plausible approaches, not an obligation to audition every acronym. Scheduling, data consistency, and export correctness seem more likely to repay attention before an adapter beauty contest.



Herman

NVIDIA's checkpoint example is a good reason to start with vendor guidance, without treating it as the winner by definition. A NeMo developer said in twenty twenty-three that averaging the best saved checkpoints beat the best individual one ninety-five percent of the time.



Corn

That percentage sounds more measured than the sentence behind it.



Herman

It was developer testimony, not a controlled success-rate estimate. Useful experience, but I wouldn't attach statistical confidence to it. The current agent instructions are more conditional. Average the selected checkpoints, evaluate the result, and retain it only if it beats the best individual checkpoint.



Corn

Mechanism first, then. We're averaging the weights, not asking several models to vote on the transcript?



Herman

Weight averaging. Several checkpoints from the same training trajectory often represent nearby solutions. Averaging can reduce local variation and land on a solution that generalizes better. You deploy one model, so it doesn't carry the inference cost of running an ensemble.



Corn

Provided the checkpoints belong together.



Herman

Same architecture and compatible weights, and usually a coherent region of the same run. Arbitrarily averaging independently trained models can destroy useful structure because their parameters need not line up.



Corn

Five individually competent transcripts would at least remain transcripts. Five incompatible weight files can produce one very confident appliance.



Herman

Daniel's improvement is consistent with nearby checkpoints benefiting from smoothing. It doesn't tell us whether averaging will win on every future corpus. The later instruction to test the result is stronger operational guidance than the old percentage.



Corn

I like that the sample improved what he already paid to train. That makes post-training selection part of the recipe's value, not housekeeping after the real work.



Herman

What would you keep from that run for the next one, Corn?



Corn

The exact starting checkpoint, the data revision, the effective batch, schedule, and the final artifacts' identities. Then a short explanation of what changed and which errors moved. Not a diary saying looked promising.



Herman

I'd add learning curves and how checkpoints were selected. An early best checkpoint followed by worsening validation suggests a different next experiment from a run that was still improving when its budget ended.



Corn

Neither tells you the cause alone. Worsening could mean too much training, too aggressive an update, or a tiny validation set getting noisy.



Herman

You'd change one plausible cause first?



Corn

When comparing recipes, yes. When simply trying to ship, sometimes you bundle changes. But then you don't pretend you learned which one worked.



Herman

That distinction is useful with Claude supervising on Modal. Let the agent execute an explicit comparison, capture the resolved configuration and software revision, and summarize failed hypotheses too. Otherwise it may helpfully repair three things and leave you unable to reproduce the improvement.



Corn

An agent that changes everything until the number drops has learned how to end the conversation.



Herman

Artifact handling is a surprisingly large separator between good recipes and pretty notebooks. A NeMo model package saved most recently may have overwritten an earlier package, even when the earlier checkpoint had the better validation score.



Corn

Latest has a wonderfully undeserved confidence.



Herman

The highest-ranked training checkpoint, the final exported package, and the averaged export can all differ. Evaluate each actual artifact independently. Don't assume the number beside a checkpoint automatically belongs to the file you deploy.



Corn

This is where I prefer fingerprints over filenames. Call something best and it can coast on that name through several bad decisions.



Herman

Compatibility matters too. Some averaging scripts are deprecated. PyTorch changed its default checkpoint-loading behavior starting with version two point six, favoring weights-only loading. Older checkpoints can require an explicit compatibility override.



Corn

Which should be a deliberate decision for a trusted checkpoint, since older serialization can execute code during loading. Fixing the loading error shouldn't become permission for every downloaded file.



Herman

Yes. Pin the environment, understand what the override permits, and distinguish a loader problem from a model problem. None of those changes should get credited as a learning-rate improvement.



Corn

The exported model needs to be tested outside the training loop. I don't trust a contestant who supplies its own finishing photograph.



Herman

The agent instructions call for standalone transcription evaluation for the baseline, best checkpoint, final package, and averaged model. They also specify explicit bfloat sixteen computation without automatic mixed precision for evaluation. The purpose is a stable inference setup, not a second experiment accidentally nested inside the first.



Corn

Scoring too. Strip punctuation for one run and retain it for another, and the model can win by changing the referee.



Herman

Their default scoring contract strips capitalization and punctuation. Code-switching needs additional explicit choices. Are alternative spellings accepted? Is a borrowed word expected in its original script or transliterated? Does the reference consistently represent spoken numbers?



Corn

I'd keep a normalized aggregate score, plus separate views for language switches, names, and ordinary retained speech. Otherwise the largest bucket wins the argument. Most recordings may contain stretches that were already easy.



Herman

Hold out complete recording sessions where possible, not adjacent snippets from the same source. Then keep the final test set out of repeated recipe selection. If you keep picking winners against it, you've made it another validation set.



Corn

You can memorize a test set without ever showing it to the optimizer. It just takes more meetings.



Herman

Cheap preflight checks matter before tuning. Mixed transcript conventions can make the model spend capacity learning contradictions. Lhotse filtering can silently remove examples, so inspect what remains after duration and token-rate limits.



Corn

Particularly the switch-heavy examples. They may look unusual precisely because they're the point of the exercise.



Herman

A fixed list of diagnostic clips helps catch that. A tiny smoke test can verify audio loading, target text, export, and inference before the long run. That's cheaper than discovering a manifest mistake after tuning six schedules.



Corn

I would keep failed runs in the comparison table too. Otherwise the record becomes a collection of winners with no indication of how much searching bought them.



Herman

For the next schedule, the vendor suggests a short warmup, roughly one to two percent of the total update budget, followed by cosine decay. The scheduler and trainer must agree about the total steps.



Corn

A learning rate decaying toward a finish line the trainer never reaches would explain some mysterious underachievement.



Herman

Save several low-validation-error checkpoints, not just the last. Five is their suggested retained set. Precision is another measured choice. True bfloat sixteen can improve memory headroom compared with mixed mode, but fall back if the run becomes unstable.



Corn

Extra headroom is useful if you spend it consciously. Increasing the batch because memory became available changes the experiment again.



Herman

There are code-switching approaches that change the data requirement, rather than merely the optimizer. One reinforcement-learning result matched full-data LoRA supervised tuning using a tenth of the training data across ten language pairs.



Corn

What was it rewarding that ordinary transcription loss missed?



Herman

Recognition accuracy plus writing-system fidelity. It penalized output in the wrong script. That matters when an acoustically plausible result is written in a form the task doesn't want. The gains also transferred to human-recorded code-switching, though that doesn't establish a drop-in Parakeet recipe.



Corn

It makes the target preference explicit rather than hoping the model infers it from a small corpus.



Herman

Synthetic speech offers another route. A text-to-speech and transcription flywheel generated about twenty-two thousand entity-heavy code-mixed utterances for under fifty dollars in marginal generation cost.



Corn

Less than the administrative cost of naming the experiment.



Herman

Entity hit rate improved seventeenfold against the tested open baseline, but ordinary read-speech word error could worsen by up to six point six absolute points. That's a meaningful cost, not a footnote.



Corn

I'd want the actual entity list and baseline performance before becoming impressed by the multiplier. Improving from almost never to sometimes can produce a magnificent ratio.



Herman

Agreed … the result is a candidate strategy for targeted vocabulary, not proof that synthetic speech replaces genuine switching. Daniel's new recordings still expose pronunciation, timing, and context that a generator may miss.



Corn

There's also diminishing return. Additional hours should earn their place through the errors they cover, rather than merely making the dashboard larger.



Herman

In the dysarthric adaptation work, five minutes captured nearly half the character-error reduction achieved with thirty minutes. That curve belongs to that task, but it illustrates why the next hour's content matters more than its existence.



Corn

A controlled comparison across each corpus revision sounds useful, though keeping all those exports and rerunning them manually becomes impractical. Eventually you need to decide which one to keep…



Hilbert

…the average, Corn. That's what I keep running. I've been averaging my dictation checkpoints for years. The setup is a mess. Training runs in rented compute, evaluation runs on another machine, and the upload form still wants a description I fill with the same sentence. Your scheduler explanation was shorter.



Herman

A low bar, apparently.



Hilbert

Yossi at the computer shop handles the uploads when the browser gets awkward. I've dealt with him far too long to change shops now. This week he averaged five exports for me. The best individual one scored lower on my ordinary reading sheet. The average got more of the names in my supplier messages right. I use those messages. I don't send anybody the reading sheet.



Corn

The reading sheet was measuring a job you weren't asking it to do.



Hilbert

It was clean speech. Sensible sentences. No stock codes. I dictate the rejected transcripts too, with the shop shutter half down. That's the sound I want. Yossi opens it for me before he makes his tea. He knows I can't manage that little latch with these anteater claws. I think the developer who said averaging usually wins was describing work he'd actually done. The newer instruction sounds like somebody once shipped the wrong export. I'd trust it for choosing the file.



Herman

Your supplier messages could be a second evaluation set, provided the same messages aren't feeding the next training run. Then the lower reading-sheet score and better names would both remain visible.



Corn

Though the shutter recordings may be testing background noise as much as names. Separating clean supplier messages from noisy ones would tell you which improvement you're getting.



Hilbert

The shutter rattles. It doesn't stay in the same place.



Herman

That makes the recording conditions worth tracking alongside corpus size. New hours can change the noise distribution, vocabulary, and speaking style together. A recipe that coped well last month may be answering a different problem this month.



Corn

Daniel can carry forward the comparison method more confidently than the winning number. Fixed evaluation, identifiable exports, known starting weights, and one clear hypothesis buy him reusable knowledge. The precise learning rate may expire.



Herman

Vendor recipes supply informed starting points and implementation knowledge. The strongest ones expose their assumptions and verify the deployed artifact. They don't promise that the same settings remain optimal as the corpus grows.



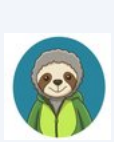
Corn

His successive runs make sense, then. The useful tinkering separates gains from more data, gains from training, and gains from choosing the output correctly. Averaging belongs in that last comparison until it earns its place.



Herman

I'm curious what happens as NVIDIA puts more of this decision-making into agent skills. Instructions that once sat beside a notebook can now guide the agent running it, including which artifact to evaluate and when to reject an average. The missing context may be as mundane as whether this week's audio was recorded beside a rattling shutter.



Corn

Thanks to Hilbert Flumingtop for producing the show.



Herman

For more along these lines, there's episode five, Fine-Tuning ASR For Maximal Usability; episode nine, Benchmarking Custom ASR Tools - Beyond The WER; and episode six, How To Fine Tune Whisper. This has been My Weird Prompts, the human-AI collaboration podcast.



Corn

You can find the archive at my weird prompts dot com.



Herman

We'll be back soon.



Corn

See you tomorrow.

SOURCES

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

1. **NVIDIA NeMo repo, `main` branch, accessed 2026-10-30 [primary]**
https://github.com/NVIDIA-NeMo/NeMo/blob/main/.claude/skills/nemo-speech-asr-finetune/references/training-evaluation.md

2. **NVIDIA NeMo repo, `main` branch, accessed 2026-10-30 [primary]**
https://github.com/NVIDIA-NeMo/NeMo/blob/main/.claude/skills/nemo-speech-asr-finetune/references/setup-checkpoints.md

3. **NVIDIA NeMo Speech docs, Fine-Tuning page, accessed 2026-10-30 [primary]**
https://docs.nvidia.com/nemo/speech/latest/asr/fine_tuning.html

4. **NVIDIA Riva 2-24-0 tutorial [primary]**
https://docs.nvidia.com/deeplearning/riva/archives/2-24-0/tutorials/asr-finetune-parakeet-nemo.html

5. **NVIDIA Riva tutorials, updated 2025-12-24 [primary]**
https://github.com/nvidia-riva/tutorials/blob/main/asr-finetune-parakeet-nemo.ipynb

6. **Hugging Face model card, created 2025-08-04 [primary]**
https://huggingface.co/nvidia/parakeet-tdt-0.6b-v3

7. **NVIDIA Developer Forums, post dated 2023-01-18**
https://forums.developer.nvidia.com/t/fine-tune-the-hind-nvidia-nemo/239264/21

8. **Vividh-ASR: A Complexity-Tiered Benchmark and Optimization Dynamics for Robust Indic Speech Recognition, Interspeech 2026**
https://arxiv.org/abs/2605.13087v2

9. **Choosing a PEFT Variant for Per-Patient Dysarthric ASR, submitted 2026-09-02**
https://arxiv.org/abs/2609.02735v1

10. **Reinforcement Learning for Data-Efficient Code-Switched ASR, Interspeech 2026**
https://arxiv.org/abs/2607.02757v1

11. **The TTS-STT Flywheel, submitted 2026-05-04**
https://arxiv.org/abs/2605.03073v1

12. **Optimizer-Aware Fine-Tuning of Whisper Small with Low-Rank Adaption, *Information* 16(11):928**
https://www.mdpi.com/2078-2489/16/11/928

13. **TechCrunch, 2026-10-28**
https://techcrunch.com/2026/10/28/voice-ai-in-vogue-nvidia-releases-new-open-transcription-model/

14. **The New Stack, 2026-10-28**
https://thenewstack.io/nvidia-nemotron-promptable-transcription/

15. **GitHub, 1,221, last push 2026-05-08**
https://github.com/yeyupiaoling/Whisper-Finetune

16. **GitHub, 221, last push 2026-07-02**
https://github.com/jumon/whisper-finetuning

17. **GitHub, created 2026-06-02**
https://github.com/kim-el/nvidia-parakeet-vastai-training

18. **GitHub, last push 2026-09-17**
https://github.com/kristijonasatpro/kmynas