# Episode Book

EPISODE #6657

# Why Fine-Tuned Models Keep the Same File Size

| | |
|---|---|
| PUBLISHED | 17 November 2026 at 11:19 UTC |
| DURATION | 21:59 |
| LISTEN | myweirdprompts.com/episode/fine-tuning-safetensors-file-size-73574/ |
| HOSTS | Corn, Herman, Dorothy, Hilbert |
| SCRIPT MODEL | z-ai/glm-5.3 |
| VOICES | omnivoice |
| PIPELINE | V5.35 |
| SOURCES | 25 cited by the research (listed at the end) |
| BOOK GENERATED | 17 November 2026 at 14:42 UTC |

## SYNOPSIS

After 70 hours of voice fine-tuning, a Parakeet TDT speech model learns new vocabulary while its 2.5-gigabyte safetensors file barely grows. The format is a fixed-size photograph, not a diary: training nudges existing 32-bit floats rather than adding storage, while discarding 10 gigabytes of live optimizer state. That fixed frame also hides catastrophic forgetting, where new accents and words overwrite old ones. Practical fixes include weight averaging, Elastic Weight Consolidation, and decode-time contextual biasing.

# TRANSCRIPT



### Corn

Seventy hours of your own voice, poured into a model, one training round after another, and the file that comes out the other end has grown by about as much as a photograph of your lunch. That's Daniel's situation, and he's noticed something odd. The Parakeet model he's fine-tuning is clearly getting better. He can watch it pick up new vocabulary as he goes. Yet the safetensors file, the one thing that represents the model, has put on a few megabytes and nothing more. So his question is the right one. Where is the learning stored, if not in the file?



### Herman

And it's a good puzzle, because the intuition being violated here is deep. We're used to learning meaning accumulation. A notebook fills up. A library gets an extension. Daniel's model apparently learned dozens of new words and the building didn't get any bigger.



### Corn

Which either means the model is lying about learning, or our idea of where learning lives is wrong.



### Herman

It's the second one, and the answer is almost embarrassingly concrete once you look at what a safetensors file actually is. Parakeet TDT, the version three checkpoint, has six hundred twenty-seven million and change parameters, stored at thirty-two bit floating point. Four bytes each. Call it two and a half gigabytes of raw tensor data, plus a header measured in kilobytes. That's the whole file.



### Corn

So the size was decided before Daniel recorded a single hour.



### Herman

The day the architecture was drawn, yes. The file size is parameter count times bytes per parameter. Nothing else. The format itself is almost aggressively simple. Eight bytes at the front telling you how long the header is, then the header, which is plain JSON listing every tensor's name, its data type, its shape, and its offsets into the file. Then the rest is just the



### Herman (continued)

numbers. And the spec insists the buffer can't contain holes. It's a serialization format. It has no slot for "things learned."



### Corn

You've used this line before, but it's the one that carries the whole episode. The file is a photograph, not a diary.



### Herman

A diary gets thicker as you write. A photograph doesn't care how long you stood in front of the scene. Every time Daniel saves a checkpoint, the camera clicks again. Same frame, same size, different light.



### Corn

Okay. So the photograph idea tells me the size is pinned. It doesn't yet tell me where the seventy hours went.



### Herman

Then let's open the camera. During training, those six hundred twenty-seven million slots are the same slots. Fine-tuning doesn't add slots. It nudges the values in the existing ones. The FastConformer encoder and the TDT decoder keep their shapes, the header keeps listing the same shapes, the byte buffer keeps its exact length, and every single float inside it can be different at the end. The file is the same size and is a different model. Those two facts are not in tension.



### Corn

The only ways the file can grow, then, are the boring ones.



### Herman

More tensors, bigger tensors, or a bigger header. If you bolt on adapter matrices, the file grows by the size of those matrices. If you append rows to a vocabulary embedding, it grows a little. If your training script starts stamping step counts and timestamps into the metadata, the JSON header grows by a few hundred bytes. A few megabytes of growth over many rounds is consistent with metadata creep plus maybe a small adapter or some appended vocabulary rows. It is not consistent with the model acquiring new storage capacity, because the format has no mechanism for that.



### Corn

I want to be careful with one word, though. You said "nudge." Seventy hours of audio is a lot of nudging. What does a new word actually look like in there?



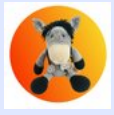
### Herman

Not a word. That's the part people get wrong. There's no cell anywhere that now contains "Parakeet." A new word is a shift in the geometry of the weight matrices. The encoder's acoustic representations and the decoder's token and duration predictions jointly re-weight, so that a particular sound pattern now maps to a particular token sequence. The weights are multipliers, not records. Think of... no, actually, skip the metaphor. The numbers are volumes on an enormous mixing desk, and learning is turning existing knobs. Nobody installs a new knob for a new word.



### Corn

Which is strange, when you phrase it that way, because it works. Daniel says he can watch vocabulary appear.



### Herman

He can, and there's a result that explains why so little of the model needs to move. Aghajanyan, Zettlemoyer and Gupta, the intrinsic dimensionality work. They showed that pre-trained models have a very low intrinsic dimension, meaning there's a low-dimensional reparameterization that does almost everything full fine-tuning does. Their headline number: training only two hundred parameters, randomly projected back into the full space, got RoBERTa to ninety percent of full fine-tuning performance on a standard benchmark.



**Corn**

Two hundred.



**Herman**

Out of over a hundred million. The pre-trained model has already built a space where the adaptation you need lives in a tiny corner of it. Daniel's seventy hours aren't filling a warehouse. They're adjusting a few hundred dimensions of a room that was already furnished.



**Corn**

Here's the number I keep circling, though. During the actual training, this isn't a two and a half gigabyte affair, is it?



**Herman**

No, and this is the part I find beautiful. Full fine-tuning with AdamW in mixed precision runs on the order of sixteen bytes per parameter. The weights themselves, the gradients, and two optimizer moment estimates. For six hundred twenty-seven million parameters, that's roughly ten gigabytes of live training state. Ten gigabytes of machinery doing the work, and at checkpoint time almost all of it is thrown away. The gradients, gone. The optimizer moments, gone. What gets saved is the weights, the residue. Daniel is looking at the residue and asking where the process went. The process went in the bin.



**Corn**

The photograph doesn't even show the developing fluid.



**Herman**

The LoRA people proved the same point by construction, from the other end. You train small added matrices alongside frozen weights, and afterwards you multiply them out and fold the result back into the original weight matrix. The adapter disappears, the file is exactly the original size, the model runs at original speed, and it still knows the new domain. Learning happened and the file did not grow by one byte. That's not a trick. That's the nature of the thing.



**Corn**

There's a detail in all this that I suspect explains Daniel's few megabytes, and it's checkable. The header is plain JSON at the start of the file. He can open it himself.



**Herman**

He can. Every tensor name, every shape, every offset, readable in any text editor. If the growth is metadata, he'll see step counts and config strings. If it's an adapter, he'll see new tensor names. If it's vocabulary rows, he'll see a shape with a bigger first dimension. The mechanism is fully documented. The exact composition of his particular growth is the one thing only he can look at.



**Corn**

That's a satisfying answer to where the learning isn't. Values, not bytes. Photograph, not diary. I'd almost leave it there, except Daniel said something in passing that I can't stop chewing on. He trains incrementally. Round after round.



**Herman**

You're going at the forgetting.



**Corn**

I am. Because the file will never show him what he's losing, and that's the part of this episode that nobody asks about.



**Herman**

Then let's name it. Catastrophic forgetting. Every serious paper on sequential fine-tuning of speech models leads with it. Vander Eeckt and Van hamme put it bluntly: even monolingual ASR models cannot be extended to new accents, dialects, or topics without suffering it. The model doesn't just add Daniel's voice. It rebalances. And the rebalancing is paid for out of everything it used to know.



### Corn

Hannah raised a version of this once, about what happens to a fine-tune when the base model moves on. This is the mirror image. What happens to the base model's other abilities when Daniel's fine-tune keeps moving in.



### Herman

And the file gives him no warning at all. A model that has forgotten half of everyone else's accents weighs exactly the same as the day it was downloaded. There's no shrinkage, no warning light, no "gotten worse" bit in the header. The photograph still shows a healthy model.



### Corn

Here's what I want to pin down, though. There's a tension in the numbers. NVIDIA's own adapter recipe for fine-tuning Parakeet says ten to fifty hours of data, frozen pretrained weights, and it explicitly can't change vocabulary. Daniel is at seventy hours and watching vocabulary appear. Those two facts don't sit together.



### Herman

They sit together only if he's doing something other than the frozen-encoder adapter. My reading is that he's doing full or near-full fine-tuning, which is exactly the regime where forgetting bites hardest. The adapter can't change vocabulary because the output layer is frozen. If vocabulary is changing, the output pathway moved.



### Corn

Which means the good news and the bad news are the same news.



### Herman

Yes. The learning is real and deep, and so is the exposure. Though I'll say the data scale itself isn't reckless. There's a nice comparison: a team fine-tuned Distil-Whisper with LoRA on about seventy hours of air traffic control transmissions, rank thirty-two, and got under four percent word error rate. Same data scale as Daniel. The parameter-efficient route was plenty.



**Corn**

So suppose he keeps his rounds and doesn't want to retrain the world every time. What does the literature say a man in his position should actually do?



**Herman**

The most practical thing, and I like it because it's almost insultingly simple, is weight averaging. Plantinga, Yoo and Dhir showed that taking several models, each fine-tuned from the same generalist base, and linearly interpolating their parameters, produces one model that does well on all of it. One model for inference. No tower of adapters. For Daniel's workflow, that's averaging the round-N checkpoint back toward the base, or toward earlier rounds. You can do it in a few lines of code.



**Corn**

Interpolating between photographs.



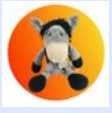
**Herman**

It works because they were all taken of the same room. The tensors correspond one to one, so the arithmetic is meaningful in a way it wouldn't be across different architectures.



**Corn**

What's the catch?



**Herman**

It's heuristic. It ignores the shape of the loss landscape, is the polite way the follow-up papers put it. The fancier methods add curvature information, or consolidate what's important. Elastic Weight Consolidation and Synaptic Intelligence, for instance, gave about five percent relative word error reduction over a plain fine-tuning baseline in an online continual learning setup. And sequential model editing, Kulshreshtha and colleagues, got up to fifteen percent over a fine-tuning baseline with no extra parameters and no access to past data.



**Corn**

No extra parameters. Same file size.



**Herman**

The whole continual learning field has quietly converged on Daniel's observation as a design goal. Keep learning without growing. But here's the lever I'd argue nobody in his position thinks of, because it doesn't involve training at all. Contextual biasing. At decode time, you hand the model a word list, and it biases its output toward those words.



**Corn**

How well does that work?



**Herman**

On proper nouns, up to fifty-one percent relative word error reduction. And the model file changes zero bytes. The weights never move. Which means no forgetting, because there's nothing to forget. If the goal is specifically vocabulary, that's a completely different lever than weight updates, and it's the one with no side effects.



**Corn**

Hold on. Fifty-one percent from a word list, and Daniel has spent seventy hours of recording to get vocabulary into the weights.



**Herman**

To be fair, they're not the same vocabulary. Biasing helps the model pick a word it already knows when context points there. It won't teach it a new accent, or his particular speaking style. But if the new vocabulary is domain words, terms, names, the biasing result is the number I'd tape to the monitor first.



**Corn**

Alright. Next up it's Dorothy's Kitchen, with Corn's mother on the line. Mum, what are we cooking?

D

**Dorothy**

Hello dear, it's your mother. I'm doing Dorothy's Kitchen today, so put your little microphone hat on.



**Herman**

She's got us on the good microphones today, I can tell.

D

**Dorothy**

Porridge, plain. You want one cup of rolled oats, three cups of water, and a pinch of salt. That's it. That's the whole terrifying list.



**Corn**

Three parts water to one part oats. Noted.

D

**Dorothy**

Now the method. You bring the water to a boil and you watch it, because water will boil over the moment you turn your back, and then you have a mess and a burnt smell and a lesson.



**Herman**

Watching the water. I've scorched enough saucepans in my day to confirm that step.

D

**Dorothy**

Then in go the oats, and you stir, slow and steady, for ten minutes, and you do not wander off. Things have gone wrong before. I'd rather not say how.



**Corn**

We've all chosen not to ask, Mum.

D

**Dorothy**

It should be soft but not gluey. If it's gluey, add a splash of milk. If it's lumpy, well, dear, we eat it anyway.



**Herman**

That's the spirit of the whole field, honestly.

D

**Dorothy**

Anyway. Your father's socks are still on the radiator and I've told him twice. Eat something warm, sweetheart. Bye bye.



**Corn**

Bye, Mum. So, the rounds keep going, and the file stays flat, and there's one more wrinkle I wanted to pull on. Daniel says he can watch the vocabulary improve round over round. Does the loss agree with him?



**Herman**

Not necessarily, and there's recent work on exactly that gap. Li, Fan and Zhou looked at grokking in pretraining and found that the internal pathways keep evolving, from instance-specific and scattered to structured and transferable, despite converged loss. The headline number stops moving and the behavior keeps improving underneath.



**Corn**

The photograph looks identical and the diary entry got better.



**Herman**

Which means Daniel's eyeball observation is more trustworthy than a loss curve would be. If he's seeing new words land in round eleven, they landed. The model is still reorganizing even after the training metric looks bored.



**Corn**

Though "converged" and "finished" aren't the same thing either. There's a failure where very long training runs look perfect on the training side and quietly collapse on generalization.



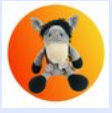
**Herman**

Late-stage collapse. It's real and it's why round thirty should earn its keep with actual transcripts, not just vibes. You test on held-out audio, full stop.



**Corn**

I want to know what you'd actually do, though, Herman. If the file can't tell you anything, what's your instrument panel?



**Herman**

A fixed reference set of audio, and it should not all be Daniel. Some of it everyone else's voices, some of it a different domain, and I'd run every checkpoint against it and plot both curves. The Daniel curve should go up. The everyone-else curve is the one that tells you whether you're still fine-tuning or just specializing. The file size tells you nothing. The test set tells you everything.



**Hilbert**

Corn keeps saying photograph. It's a ledger.



**Hilbert**

My uncle Eamonn kept a workshop ledger, forty years of repairs, every job written in the same hardback notebook. After the first year the notebook never got thicker. The entries just got denser. Overwritten in the margins, crossed through, done again smaller. He'd say the book was full in March of the first year and everything since was handwriting.



**Herman**

Does he still have the ledger?



**Hilbert**

I read it last spring. The early entries are the most legible ones, because they were written when there was room. Big clear hand, space between the lines. The later ones are cramped, nearly gone, ink on ink. That's what I'd worry about with Daniel's model. Not the pages he's adding. The pages getting squeezed.



**Herman**

That's the forgetting, though he'd never use the word. The dense entries are eating the legible ones.



**Hilbert**

Hold on, you've got a level peaking. Right. Sorry. Where was I. The uncle's notebook, yes. I keep a small one myself, for the phone. Every word the voice assistant mishears, date, what I actually said. There's a pattern and it isn't flattering. Half of it is my diction and I know which half, I'm just not saying which. The rest is the assistant, and the same six words, every week, wrong the same way. Herman, you talked about knobs earlier. Nobody has turned those knobs since the day it shipped.



**Corn**

Yours or the assistant's?



**Hilbert**

Mine are original equipment and holding fine. The assistant's are the problem. But you two were on the averaging trick before I came in. Blending the round-N photograph with the base one. Eamonn did that by accident, he'd copy a settled entry into a new book every few years, only the clean ones. Lost half the margin notes. Kept the legible hand.



**Herman**

So he averaged and forgot the fine detail on purpose.



**Hilbert**

He kept the version he could read. The detail he lost was the detail nobody came back for. But the tidy book ran out of room the same as the old one. Your averaging buys him legibility, Herman, it doesn't buy him pages.



**Corn**

No, but it buys him the early entries back. That's the part I'd actually want.



**Herman**

Then that's where the panel matters. The fixed test set, both curves, every checkpoint. Daniel can check the header JSON himself and see what those few megabytes are, but the file will never tell him what the model's quietly gotten worse at. Only held-out audio will.



**Corn**

There's a whole episode's worth of lesson in the fact that the two of you just built the same point from opposite ends. The file is a photograph of the present. The learning is real, it's in the values, and the cost was paid somewhere the file never records.



**Herman**

And the place to look for what's lost isn't the byte count. It's other people's voices.



**Corn**

That's where we'll leave Daniel, with a word list in one hand and a test set in the other. Thanks as always to Hilbert Flumingtop for producing the show. If this was your kind of episode, go back for episode eleven, How Does Fine Tuning Work Anyway; episode three, Safetensors or something else; and episode fifteen, AI Gets Personal. This has been My Weird Prompts, where no question is too obscure and no rabbit hole too deep. If you enjoyed this one, leave us a review, it's the cheapest way to help the show along. See you tomorrow, brother.



**Herman**

Tomorrow, brother. Tell the model I said congratulations on the vocabulary.

## SOURCES

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


1. **Hugging Face / NVIDIA model card , created 2025-08-04, last modified 2026-08-05, retrieved 2026-11-17. [primary]**
   https://huggingface.co/nvidia/parakeet-tdt-0.6b-v3

2. **safetensors repository README and format specification , retrieved 2026-11-17. [primary]**
   https://github.com/huggingface/safetensors

3. **NVIDIA NeMo-Speech fine-tuning documentation , retrieved 2026-11-17. [primary]**
   https://docs.nvidia.com/nemo/speech/latest/asr/fine_tuning.html

4. **NVIDIA Riva Parakeet fine-tuning notebook , last updated Dec 24, 2025 per page, retrieved 2026-11-17. [primary]**
   https://github.com/nvidia-riva/tutorials/blob/main/asr-finetune-parakeet-nemo.ipynb

5. **Hugging Face org listing for NVIDIA Parakeet family (15 models with param counts and creation dates), retrieved 2026-11-17.**
   https://huggingface.co/nvidia

6. **Google Machine Learning Crash Course, LLMs: Fine-tuning, distillation, and prompt engineering, retrieved 2026-11-17.**
   https://developers.google.com/machine-learning/crash-course/llm/tuning

7. **DeepWiki summary of nvidia-riva adapter-based fine-tuning tutorial, retrieved 2026-11-17.**
   https://deepwiki.com/nvidia-riva/tutorials/4.3.3-adapter-based-fine-tuning

8. **community Parakeet fine-tuning pipeline, retrieved 2026-11-17.**
https://github.com/furkanksl/parakeet-asr-finetuning-pipeline

9. **Aghajanyan, Zettlemoyer, Gupta, Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning, submitted 2020-12-22, retrieved 2026-11-17.**
https://arxiv.org/abs/2012.13255v1

10. **Vander Eeckt & Van hamme, Continual Learning for Monolingual End-to-End Automatic Speech Recognition, EUSIPCO 2022, retrieved 2026-11-17.**
https://arxiv.org/abs/2112.09427v4

11. **Vander Eeckt & Van hamme, Rehearsal-Free Online Continual Learning for Automatic Speech Recognition, Interspeech 2023, retrieved 2026-11-17.**
https://arxiv.org/abs/2306.10860v1

12. **Plantinga, Yoo, Dhir, Continual Learning for End-to-End ASR by Averaging Domain Experts, submitted 2023-05-12, retrieved 2026-11-17.**
https://arxiv.org/abs/2305.09681v1

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

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

15. **Ahadzi et al., Continuous Learning for Children's ASR: Overcoming Catastrophic Forgetting with EWC and Synaptic Intelligence, Interspeech 2025, retrieved 2026-11-17.**
https://arxiv.org/abs/2505.20216v1

16. **Kulshreshtha et al., Sequential Editing for Lifelong Training of Speech Recognition Models, Interspeech 2024, retrieved 2026-11-17.**
https://arxiv.org/abs/2406.17935v2

17. **Pokel et al., Variational Low-Rank Adaptation for Personalized Impaired Speech Recognition, revised 2026-03-16, retrieved 2026-11-17.**
https://arxiv.org/abs/2509.20397v2

18. **Wagner et al., Personalized Fine-Tuning with Controllable Synthetic Speech..., Interspeech 2025, retrieved 2026-11-17.**
https://arxiv.org/abs/2505.12991v1

19. **Kim et al., Gated Low-rank Adaptation for personalized Code-Switching ASR on low-spec devices, ICASSP 2024 Workshop, retrieved 2026-11-17.**
https://arxiv.org/abs/2406.02562v1

20. **Dahal et al., Mixture of Phonetic Experts Based Low-Rank Adaptation of Conformer Models for Accented English ASR, Interspeech 2026, retrieved 2026-11-17.**
https://arxiv.org/abs/2610.26888v1

21. **Bagat, Illina, Vincent, Mixture of LoRA Experts for Low-Resourced Multi-Accent ASR, Interspeech 2025, retrieved 2026-11-17.**

https://arxiv.org/abs/2505.20006v1

22. **Wang et al., Towards Contextual Spelling Correction for Customization of End-to-End Speech Recognition Systems, IEEE TASLP, retrieved 2026-11-17.**

https://arxiv.org/abs/2203.00888v2

23. **Mirzaei, Arzate, Vijay, Enhancing Aviation Communication Transcription: Fine-Tuning Distil-Whisper with LoRA, submitted 2025-03-13, retrieved 2026-11-17.**

https://arxiv.org/abs/2503.22692v1

24. **Li, Fan, Zhou, Grokking in LLM Pretraining? Monitor Memorization-to-Generalization without Test, ICLR 2026, retrieved 2026-11-17.**

https://arxiv.org/abs/2506.21551v4

25. **Kazanskii, How to Tame Grokking: Representation Geometry as a Control Signal, submitted 2026-07-13, retrieved 2026-11-17.**

https://arxiv.org/abs/2607.11666v1