Episode Book

EPISODE #6610

What Goes Into a Character-Voice LoRA Adapter

PUBLISHED	15 November 2026 at 15:30 UTC
DURATION	29:31
LISTEN	myweirdprompts.com/episode/character-voice-lora-adapters-63565/
HOSTS	Corn, Herman, Dorothy, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.30
SOURCES	17 cited by the research (listed at the end)
BOOK GENERATED	15 November 2026 at 17:52 UTC

SYNOPSIS

A character-voice LoRA adapter can weigh just twelve megabytes, yet it is only half of a voice—reference-conditioned models like MOSS and F5-TTS still need a reference audio clip on every generation. This episode unpacks the de facto PEFT LoRA format, how rank controls capacity, and why training data requirements span from seven-second NanoVoice adaptations to twenty-hour F5-TTS runs. It also covers curriculum splitting, acoustic diversity over raw volume, and the open debate over where to place adapters.

TRANSCRIPT



Corn

Twelve megabytes. That's the size of the file that makes me sound like me.



Herman

Roughly. Depends who you ask — the same project that trained our cast also says six. But yes, on the order of a floppy disk.



Corn

For listeners just joining us, that's not a metaphor I picked for fun. Daniel wrote in this week — he'd finally connected the dots that the voices on this show, mine, Herman's, Hilbert's, Dorothy's, are LoRA adapters sitting on top of a shared text-to-speech base, and that this is one of the main ways the whole field does custom voices now. So he wants the full picture. Is there a standard format or recipe for a character-voice LoRA, how much audio does one actually take, what should you know before building a whole cast this way — and then the big one. One adapter per character, which is what we do, versus a single model that holds the whole cast and you just pick the character at generation time.



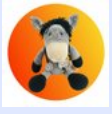
Herman

He said "the big comparison" and he meant it, because that fork decides your whole pipeline.



Corn

But before we can compare the two routes, we need to be precise about what the adapter actually is — because the mental model most people carry is wrong, and everything downstream depends on it.



Herman

The mechanism is elegant, and I say that having resisted it for about a week. You take a linear layer in the network, with its big weight matrix, W-zero. You freeze it. Then you bolt on a low-rank update — two small matrices, B and A, whose product is a matrix the same shape as W-zero but with rank r, tiny. The layer's effective weights become W-zero plus B times A. During training, only B and A move. The base never changes.



Corn

And the rank is how much room the adapter has to be wrong about the base model.



Herman

That's — actually not a bad compression. Rank is capacity. Rank sixteen means the update to any given layer lives in a sixteen-dimensional subspace of what could be a thousand-dimensional space.



Corn

The lineage matters here, doesn't it. This didn't start as adapters.



Herman

The direct ancestor is UnitSpeech, which fine-tuned every parameter in the model on one target speaker's recordings. VoiceTailor was the step that said, no, we can freeze the whole model and inject a LoRA into the linear layers of the attention modules, train only the adapter, and get higher speaker similarity than you'd expect for the cost. In TTS, the adapter lands on the attention projections of a pretrained multi-speaker model, so what it learns is this speaker's acoustic identity — how this voice colors speech — not how speech works. The base already knows speech.



Corn

Now the correction, because this is the part the explainers get wrong and Daniel will hit it the moment he builds anything. The adapter doesn't replace the reference clip.



Herman

Not on a reference-conditioned base, no. The MOSS release puts it in caps in their own card — the adapter carries the identity, but the base model is still reference-conditioned: generate without a reference clip and you get a random speaker. The LoRA sharpens identity. The conditioning audio is still part of the prompt. Every single generation.



Corn

So the adapter isn't a portable character file. It's half of one.



Herman

It's a key without the lock, or — I'm bad at these. It's the tuning and not the instrument. The point stands. Keep that in your head for the whole episode, because it changes the cast arithmetic later.



Corn

The arc, then: recipe and data first, cast-building second, the architectural fork last.



Herman

Let's do format first, because that one's the easy yes. There is no standardised character-voice LoRA spec — nobody has published a standard — but there is a de facto container everyone converged on, which is the PEFT LoRA adapter. Two files: a config and a safetensors weights file. You load it with the standard PEFT loader and you're running. Both the MOSS voice-profile release and the F5-TTS LoRA work use exactly that, to the letter.



Corn

Which means the character voice is interoperable by accident.



Herman

By accident, yes, but reliably. The wrinkle is that the container is exact and the stack around it is fussy. The MOSS quickstart pins specific versions of torch, transformers, PEFT, accelerate — down to a specific Python point release, three-eleven — and you have to run with trust remote code enabled, because the modelling code ships in the base model's repo rather than in the library.



Corn

"Trust remote code" is the phrase where the whole security model of your pipeline goes to die quietly.



Herman

It's a real trade. You're executing whatever the base-model author put in the repo. In a cast pipeline you do that once, on the base, and the adapters themselves are inert data — so the exposure is bounded, at least.



Corn

Recipe next. If there's no standard, what are people actually running?



Herman

The cleanest published recipe is the F5-TTS reference run. Rank sixteen, alpha sixteen — the scaling factor equal to rank, so a plain update — targeting the attention projections only: the query, key, value and output projections. Learning rate one times ten to the minus four. One epoch, about twelve hundred and fifty updates, batch size one, on a single RTX 3090 Ti. Out of that comes a twelve-megabyte adapter.



Corn

From a three-hundred-forty-million-parameter model.



Herman

Three hundred thirty-nine million, nine hundred eighty thousand and change. Two point eight million trainable — that's zero point eight five percent of the model. And here's the discrepancy I flagged at the top: the same project's own README says the adapters come out around six megabytes, not twelve. Same people, same run, two figures. My guess is different rank or dtype settings between the guide and the fork, but honestly they never reconcile it, and I'd rather leave it sitting there as a caution than pretend it's settled.



Corn

A character voice whose size depends on who in the project you ask.



Herman

It's a three-hundred-and-forty-million-parameter model, though — your average listener's mental image of a "small file" already broke.



Corn

Rank. You said rank is capacity. How real is that knob in practice?



Herman

Real enough that people ship both settings of it deliberately. The MOSS single-voice release — the one called Velvet Sage Baritone — publishes two adapters for the same voice. Rank sixteen, alpha thirty-two, that's thirty-four point four million trainable parameters. Rank thirty-two, alpha sixty-four, sixty-eight point seven million. They're the same character, doubled. And there's a JSON file recording which one they recommend and why.



Corn

So a character voice is a thirty-four-million-parameter object or a sixty-nine-million-parameter object, depending on how much room you gave it to be that character.



Herman

And the recipe that produced them is more interesting than the rank. Two stages. Stage one trains on the better half of every group of takes, ranked by the group's own reward score — the top sixteen of thirty-two. Stage two warm-starts from that and trains on the winners only, the single best take per group, at a lower learning rate. Curriculum. Coarse first, best last.



Corn

The part I want pulled out is the split.



Herman

Forty-six groups held out entirely, and the split is by group, not by clip. That sounds pedantic until you hear why: the thirty-two candidates in a group are the same sentence recorded under the same condition. Hold out individual clips and the model has literally heard the line before. You'd get a beautiful validation score that means nothing.



Corn

Leakage, wearing the costume of rigour.



Herman

If you take one methodological thing from this whole segment into your own build, it's that — split by source, not by sample. It applies to any audio you curate. And while we're on the recipe, caption handling: the captions were resampled every epoch from stored per-clip measurements rather than baked in. Even the text side of the training loop has craft in it.



Corn

Data, then. The honest question. How much audio do you need?



Herman

The honest answer is that the published numbers span two orders of magnitude, and almost all of the disagreement is about what question you're asking. Zero-shot reference conditioning, no adapter at all: XTTS v2 clones a voice from six to ten seconds of reference audio. That's the floor for the no-training route.



Corn

And the moment you train an adapter?



Herman

For XTTS v2 fine-tuning, the AllTalk wiki says two to three minutes minimum, five plus recommended. TTS Forge says ten minutes minimum, twenty to thirty recommended, and they report a working run on eighty-nine voice samples — about five minutes total — on a six-gigabyte laptop GPU. Then a Coqui guide from this year says five minutes is enough and fifteen plus shows diminishing returns.



Corn

Three guides, three numbers, one model.



Herman

And that's the consensus end of the spectrum. The F5-TTS maintainer was asked directly whether thirty minutes suffices, and his answer was, we haven't tried with ten hours, but pretty sure that twenty hours works. Pressed on under three hours: not sure, we haven't tried, but think there's a chance.



Corn

The most honest sentence in the field.



Herman

It really is. And to be clear, the actual F5-TTS LoRA run used nearly eleven thousand rows from a national speech corpus — a corpus speaker, not a bespoke recording — and the authors say outright that this does not establish a minimum dataset size. Then at the extreme low end there's NanoVoice, which adapts a speaker from one reference clip. One. Five hundred iterations, rank two, twenty-one thousand trainable parameters per speaker, and the adaptation takes about seven point six seconds on an A40.



Corn

Seven point six seconds.



Herman

Not a typo. It's a different regime entirely — tiny adapters, one per speaker, adapted at the edge. Rank two. Our cast voices are rank thirty-two, so we're carrying a thousand times more adapter than that.



Corn

Which is a strange thing to picture. What separates the seven-second voice from the twenty-hour voice?



Herman

What the study from this year found, fine-tuning an LLM-based TTS backbone, is that the gains are governed less by volume than by the characteristics of the data. Speakers whose training data had enough acoustic variety gained up to zero point four two DNS-MOS points — that's a meaningful perceptual jump — and signal-to-noise rose by as much as a third. Every evaluated speaker improved on fidelity. But with uniform data, the model can amplify the noise. Same adapter, same mechanism, opposite outcomes, decided entirely by what you fed it.



Corn

Diversity over dosage.



Herman

I've been saying that about medication for thirty years and nobody put it on a sticker either. The mixing is the prescription.



Corn

One loose end from the recipe before we move to casts. Where you put the LoRA — is that settled?



Herman

It is not, and it's the sharpest open disagreement in the literature we've got. The F5-TTS practitioners put the LoRA on the attention projections — to-q, to-k, to-v, the output projection — and report success. There's an adapter-extension paper that tested LoRA in self-attention for new speakers and found it performed terribly: a speaker-similarity score of zero point one four one, against zero point five six eight for vanilla adapters and zero point six for full fine-tuning. That's not a rounding disagreement, that's a contradiction.



Corn

Same technique, same target, opposite verdicts.



Herman

Different base models, different scoring, possibly different rank — nobody has reconciled it. So if you're building a voice, treat placement as something you test, not something you know.



Corn

Well, I think that's enough about adapters and ranks for one sitting. Let's go to the kitchen, I believe Dorothy is on the line.

D

Dorothy

Hello dear, it's your mother, and it's time again for Dorothy's Kitchen. Today I am going to walk you through a slice of bread. Unbuttered.



Herman

Mum, it's lovely to hear you. Unbuttered — is that a clarification or a warning?

D

Dorothy

The ingredients. Bread. That is the first thing, and honestly the last thing, but we list them properly. One slice. And a plate, which is not an ingredient, but you will want one, and you will want it nearer than you think.



Corn

How near are we talking, Mum?

D

Dorothy

The method. You set the bread on the plate. Then you watch it. Do not walk away, sweetheart. Things have gone wrong before, and I would rather not say how, but I will say this, the bread knows when you have lost interest.

D

Dorothy

Then you wait, and you do not touch the butter at all, because butter can turn on you, it really can, and there is no recovering from a butter that has decided something. Timing is everything.

D

Dorothy

And that is a slice of bread, done properly. Now go and put a jumper on, it is cold out, and ring me when you get in.



Herman

One slice, one plate, watched throughout. I'm putting that in my notes, Mum.



Corn

Where we were: building one voice is a recipe question. Building a cast is a different thing entirely.



Herman

And this is where the field gets strange, because the biggest public experiment in it doesn't treat a voice as one adapter at all. The MOSS ecosystem splits a voice into orthogonal adapters stacked on one frozen base. Who speaks — that's the voice profile, what we'd call the character. Feeling — forty separate emotion adapters. Timbre — a hundred and fourteen knob adapters, fifty-seven dimensions each in two directions. Non-speech — sixty-four vocal burst adapters, sobs, sharp inhales, that kind of thing. Character archetype — a hundred and twenty clusters. And domain style — German broadcast, sports commentary, explicitness.



Corn

So my voice isn't a file. It's a stack.



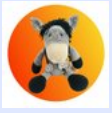
Herman

In that design, yes — identity plus mood plus timbre knobs plus whatever bursts the script calls for, all merged into the same frozen base at generation time. And the stacking is where people get burned, because activating an adapter applies it at its trained strength, which — their own words — for most of these is the wrong dose. You dial each one. The voice profile goes in at one. Emotions go in at half strength for moderate, up to nearly two for intense. Timbre knobs live between a quarter and one and a quarter. Vocal bursts at half. And the domain styles have their own doses — German broadcast emotion at a quarter, explicitness at zero point eight.



Corn

The sports commentary dose is a whole one.



Herman

Full strength, yes.



Corn

That tracks.



Herman

The point for a cast builder is arithmetic: six characters isn't six adapters, it's six adapters plus every merge-weight decision you make per line. That's a mixing desk, not a library.



Corn

You'd know. Gain staging.



Herman

I was going to resist bringing that up, but yes — it's exactly gain staging. You've got channels that sound fine alone and muddy together, and the fix is never "more channels," it's knowing what each one adds at what level. The difference is that with a mixing desk, the fader does one thing. Here the dose changes what the adapter learned, not how loud it is.



Corn

What does the stack cost you in the voice itself?



Herman

Identity, bought with expressiveness. Across their pilot, the adapters raise speaker similarity far above the base model and lower genuineness. The voices are more themselves and less alive. For a scripted cast, that's arguably a feature — you want Corn to sound like Corn in every episode. For improv, for banter, it's a real cost. The unadapted base sounds more spontaneous, just less like anybody in particular.



Corn

We're a scripted show with banter aspirations. Both, then.



Herman

We're a scripted show with banter aspirations. So we pay it in the middle. The other thing a cast builder should know is how dirty the training data gets. In that same release, forty-three percent of the training corpus fell below their own speaker-similarity floor of zero point four. In the intense-emotion block, fifty-seven percent. And their identity check is ranked, not gated — nothing stops a take that drifts off the reference voice from entering training. The adapters improve that a lot. They do not remove it.



Corn

So every character inherits some of the base model's drift.



Herman

And you find out where when a listener says the narrator sounded like Larry on line forty. Then the evaluation gap, which I find bracing. The MOSS card leads with this: no human has listened to any of this in a controlled study. Every number on the page — similarity, reward, genuineness — is the output of a learned scorer, and they've been observed disagreeing with listening judgements. Treat them as relative signals between runs, never as absolute quality. The Instavar F5-TTS work says the same thing from the other side — a completed run is not proof of improvement without a preserved base, a checkpoint, and held-out listening comparisons.



Corn

For a show whose product is how the voices sound, that's the entire gap between the literature and the job.



Herman

It is. The scorers are the map, and we are the territory.



Corn

Thank you, that's going on the marketing.



Herman

Don't you dare. Now — the fork. One LoRA per character, which is our setup, versus a single model holding the whole cast with the character as a parameter you choose at generation time.



Corn

Make our case first. You're the one who argued us into it.



Herman

I did, and I'd argue it again, but let me lay it out fairly. The per-character case has three legs. Isolation: each adapter is a separate, small, separately distributable object — twelve megabytes in the F5-TTS case, against roughly two point seven gigabytes for a full fine-tune of the same model. You can ship, roll back, or A/B one character without touching the others. When Dorothy's adapter needs retraining, nobody else's voice moves.



Corn

We've done exactly that. Twice, quietly.



Herman

Quality per voice, that's the second leg. NanoVoice's own framing is that per-speaker adaptation enhances robustness against out-of-distribution data, whereas zero-shot often struggles with unusual voices. For deliberately weird character voices — and a sloth with a Mongolian backstory qualifies — that's the argument. And third, the base is protected: the backbone stays frozen during speaker adaptation, so the speakers already in the base are unaffected by design. You can't break the other characters while fixing one.



Corn

The costs?



Herman

N adapters to train, evaluate, version, and load. N merge-weight decisions if you stack. And adapter swapping per request at inference — production loads the base once and switches, which is exactly what our pipeline does, but it's still a moving part per voice.



Corn

Now the other side. One model, character as a parameter.



Herman

The efficiency case is real and it's the headline of the whole NanoVoice line of work: they adapt forty reference voices in parallel, four times faster, using forty-five percent fewer parameters for the adaptation, with performance comparable to the baselines. Note that word. Comparable. Not better.



Corn

That's the tell in the whole pitch, isn't it.



Herman

It is. The single-model-with-a-parameter route is mostly a training-cost pitch, not a quality pitch. If you're adapting forty speakers and every extra hour of GPU time costs you, parallel adaptation with shared structure is a genuine win. If you build a cast once and ship it for a year, the efficiency argument evaporates and isolation starts looking like the whole ballgame. But there's craft in how the sharing works, and it's not symmetric. NanoVoice ablated which LoRA matrix to share across speakers. Share nothing, speaker similarity zero point nine three eight. Share the B matrix, zero point nine three three — nearly free. Share A, zero point nine zero two. Share both, zero point eight nine eight.



Corn

Wait. B is the big one?



Herman

B is roughly two-thirds of the trainable parameters, so sharing B cuts parameters by nearly a factor of three at almost no cost. Sharing A — the small one — costs you real similarity. It's backwards from the naive intuition of sharing whichever's convenient. And at the far end, a single LoRA shared across all speakers is simply a bad idea: one comparison has similarity dropping to zero point eight one three with both matrices shared, against zero point eight seven without.



Corn

So "one model, character as a parameter" still means per-speaker parameters. The sharing has to be structural.



Herman

That's the honest reading. The character doesn't become a dropdown in a fixed model — you're still carrying per-speaker state, just organised differently. What differs is who pays and when. Per-character adapters pay at training time, one voice at a time, and get isolation forever. The shared-parameter approach pays once, in parallel, and gives up the isolation.



Corn

Is the zero-shot route the cheap way out, then? No adapters at all, one big conditioning model?



Herman

Not as cheap as it looks. NanoVoice measured this: XTTS v2 uses roughly fifty times their pre-training data and still shows degraded speaker similarity, and CosyVoice needs nearly three hundred times the dataset to reach comparable scores. Big multi-speaker models are expensive to build. A small base plus adapters is cheap to extend — which is the same asymmetry again, from the other direction.



Corn

And the gotcha from the top of the show comes back for the finale. Reference conditioning.



Herman

For reference-conditioned bases like ours, the adapter is not a self-contained voice. You need the conditioning clip at generation time, every time. So a cast pipeline manages N adapters plus N reference clips, not N adapters. Two objects per character, and the second one is the one everyone forgets to back up.



Hilbert

"Carries the identity." I've been listening to you two hand that phrase around all afternoon and I don't think either of you has said the most important sentence in the pile. The one from the card, about the base still being reference-conditioned. That's the whole business.



Herman

I did flag it. Top of the show.



Hilbert

You flagged it, Herman, and then you spent forty minutes on ranks. The clip is the character. The adapter is the tuning. A cousin of mine did voice-overs for railway safety films, forty years of sessions, and when the studios went digital he asked one question at every contract signing: who keeps the raw audio. He never once handed over a session master. Re-recorded the pickup lines himself on his own deck rather than let the tapes travel. He's dead now, the deck isn't, and I've got both.



Corn

You have his session tapes.



Hilbert

I have the deck and I have the tapes, and they're in the flat, next to the desk, and the deck is plugged into the same strip as your mixing rig, which is why the lights dip when I rewind. I keep a spreadsheet of every character voice I hear in a video game — whether I think it's an adapter, a full fine-tune, or a zero-shot clone off some actor. Four hundred and eleven entries. I've been wrong on more of them than I've been right, and I only ever find out when someone posts the credits. I don't care for it. There's a woman at the market who does the tannoy announcements, Marlene, and I've had her down as zero-shot for a year because she sounds like four people I know.



Corn

Four people who know each other, presumably.



Hilbert

Four people who know me. Anyway. Your per-character call is right, I've no argument with it. What I want to know is whether one of these stacks would let me run Marlene's voice through my cousin's deck first, then into the model, so the adapter's hearing the tape. Not the other way round.



Herman

In principle you can condition on any clip the tokenizer accepts. The adapter was trained on a particular chain, though, so anything you put in front of it is a distribution shift.



Hilbert

Then I'd want the shift. Four hundred and eleven rows say my ear is no good cold; I'd rather hear it warm.



Herman

Leave the spreadsheet with us and we'll put two of your guesses against the credits.



Hilbert

Rows one and two. Row one is a racing game from two thousand and nine, the man's name is in the notes column. Don't touch the formatting, the columns are load-bearing.



Corn

Four hundred and eleven verdicts on other people's voices, and the columns are load-bearing.



Herman

He's more rigorous about the spreadsheet than the studios were about the tapes. You know what his cousin's instinct was, though? The thing he wouldn't give up was the reference audio. Not the model, not the fine-tune — the clips. That's the instinct the whole research pack points at and nobody states.



Corn

The cutting-room floor, before we wrap. The detail that didn't fit: the biggest public reference dataset in this space — four hundred thousand takes, eight hundred and fifty-plus hours, fifty shards — has two measured text defects. Nearly every burst tag in it, ninety-nine point nine six percent, was title-cased, so the model spells the word growl out letter by letter instead of performing it. And the burst density came out at thirty-four percent against the fifty they intended. Nobody heard it. A learned scorer didn't catch the casing. It took someone reading the annotations.



Herman

Eight hundred and fifty hours of audio, and the bug was in the punctuation.



Corn

Which is the open question I'd leave with Daniel, and it's not ours alone — nobody has run the head-to-head. One LoRA per character versus one multi-speaker model with a character parameter, same base, same cast, same listeners. The closest thing out there is NanoVoice's per-speaker versus shared-parameter comparison, and that's a different question. The field hasn't filled that gap. Our bet, standing by it: for a cast that ships on a schedule and needs one voice rolled back without the others moving, the per-character adapter wins — though if Hilbert's spreadsheet is anything to go by, half of us couldn't tell you which technique we're listening to anyway.



Herman

And as the bases get bigger and conditioning gets better, the adapter may shrink from a necessity to a preference. We're not there yet. Meanwhile the practical rule for anyone building a cast: two objects per character, the adapter and the clip, and the clip is the one you'll lose.



Corn

Thanks as always to Hilbert Flumingtop at the desk, producing the show and guarding the columns. If you enjoyed this one, try episode four, If Your Voice Ages, Does Your Fine-Tune Become Useless; episode three, Safetensors or something else; and episode five, Fine-Tuning ASR For Maximal Usability. This has been My Weird Prompts, the human-AI collaboration podcast. If you've built a cast of voices, leave a review and tell us how you handled the reference clips — and whether you've ever guessed wrong on what a voice actually was. You'll find us at my weird prompts dot com.



Herman

See you soon, everybody.



Corn

Ring your mother. She asked.

SOURCES

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

1. **Hugging Face, , created 2026-08-14, last modified 2026-08-15, retrieved 2026-11-15. [primary]**
   https://huggingface.co/TTS-AGI/moss-voice-profile-loras

2. **Hugging Face, , created/modified 2026-08-13, retrieved 2026-11-15. [primary]**
   https://huggingface.co/TTS-AGI/moss-voice-lora-velvet-sage-baritone

3. **Hugging Face, , retrieved 2026-11-15. [primary]**
   https://huggingface.co/instavar/f5-tts-v1-lora-female01

4. **GitHub, , retrieved 2026-11-15. [primary]**
   https://github.com/instavar/f5-tts-lora-finetuning

5. **Unsloth docs, , retrieved 2026-11-15. [primary]**
   https://unsloth.ai/docs/basics/text-to-speech-tts-fine-tuning

6. **NanoVoice: Efficient Speaker-Adaptive Text-to-Speech for Multiple Speakers (Samsung Electronics / Korean Govt. supported), 2024, retrieved 2026-11-15.**
   https://arxiv.org/abs/2409.15760

7. **When Fine-Tuning Fails and when it Generalises: Role of Data Diversity and Mixed Training in LLM-based TTS, Purwar & Choudhary, submitted 2026-03-11, retrieved 2026-11-15.**
   https://arxiv.org/abs/2603.10904v1

8. **An Exploration of ECAPA-TDNN and x-vector Speaker Representations in Zero-shot Multi-speaker TTS, Kunešová et al., TSD 2025, retrieved 2026-11-15.**
https://arxiv.org/abs/2506.20190v1

9. **Fed-PISA: Federated Voice Cloning via Personalized Identity-Style Adaptation, ICASSP 2026, retrieved 2026-11-15.**
https://arxiv.org/abs/2509.16010v2

10. **Instavar, Wei Jie Chee, 28 Mar 2026, claim v1 checked 7 Aug 2026, retrieved 2026-11-15.**
https://instavar.com/research/tts/f5-tts-fine-tuning

11. **GitHub issue, opened 2025-03-25, closed 2025-04-13, retrieved 2026-11-15.**
https://github.com/SWivid/F5-TTS/issues/910

12. **(Simple-Version)**
AllTalk TTS wiki, 17 Nov 2024, retrieved 2026-11-15.
https://github.com/erew123/alltalk_tts/wiki/XTTS-Model-Finetuning-Guide-

13. **TTS Forge, 15 Dec 2025, retrieved 2026-11-15.**
https://chandanbhagat.com.np/tts-forge-build-your-custom-voice-cloning-pipeline/

14. **Botmonster, 2026-05-12, retrieved 2026-11-15.**
https://botmonster.com/ai/train-custom-text-to-speech-voice-coqui-tts/

15. **Coqui TTS discussion, 11 Dec 2023, retrieved 2026-11-15.**
https://github.com/coqui-ai/TTS/discussions/3402

16. **Residual Adapters for Few-Shot Text-to-Speech Speaker Adaptation, retrieved 2026-11-15.**
https://ar5iv.labs.arxiv.org/html/2210.15868

17. **Adapter-Based Extension of Multi-Speaker Text-To-Speech Model for New Speakers, retrieved 2026-11-15.**
https://ar5iv.labs.arxiv.org/html/2211.00585