Cross-Lingual Grafting: Extending Frozen Codec Speech Language Models to Unseen Low-Resource Languages

Subhransu Sekhar Mahapatra, Malcom Mudhungwaza, Darshan Sundeep Sangaraju, Yashwant Singh, Maaz Khalid, Mohd Jawwad Zafar, Mustafa Hazam, Shivam Tiwari, Shivank Chaudhary, Md Aamir, Azharuddin Mondul, Deepanshu Gupta, Arif Khan

New Emerging Technologies, a subsidiary of Infinia Technologies

ABSTRACT

Modern autoregressive *codec speech language models* (codec-LMs) synthesize strikingly natural multilingual speech, yet the set of languages they cover is effectively frozen at pre-training time behind proprietary data and multi-billion-token compute. Naïvely fine-tuning such a model on a new language triggers *catastrophic forgetting*: the new language is acquired only by eroding every language the model already spoke. We introduce a **forgetting-free continual-learning recipe** that grafts a thirteenth language, **Swahili**, onto the 12-language, 357M-parameter Magpie-TTS-Multilingual (a member of the Koel-TTS family built on discrete Low Frame-rate Speech Codec tokens) *without regressing any of the original twelve*. The recipe has three parts: (i) a dedicated byte-level tokenizer with an exact **input-embedding surgery** whose new rows are **warm-started** from a trained byte-tokenizer block rather than random noise; (ii) **multilingual rehearsal** that replays a thin per-language slice of the base languages, each routed to its own native tokenizer; and (iii) a **reproducible, deterministic inference** configuration. Warm-start is the decisive lever: it moves Swahili from 77.8% to **8.8%** character error rate (median **0.0%**), *beating* a dedicated single-language fine-tune (12.9%), while the twelve base languages show **no measurable regression** (mean recognizer CER identical to the untouched base). The entire pipeline runs on a **single 128 GB unified-memory accelerator**. We further show that the community-standard Whisper metric is unsafe for Swahili and adopt a calibrated recognizer. We publicly release the unified 13-language checkpoint and model card, and argue the recipe generalizes to the long tail of the world's languages.

Index Terms: text-to-speech, speech language models, neural audio codecs, continual learning, catastrophic forgetting, rehearsal, low-resource languages, multilingual speech synthesis, Swahili.

1. INTRODUCTION

Text-to-speech (TTS) has entered the era of *speech language models*. Instead of predicting spectrograms, a Transformer autoregressively predicts a stream of discrete audio-codec tokens from text, and a neural codec decoder renders them to a waveform [1, 2]. These *codec-LMs* are natural and expressive, and because a single model can absorb many languages, increasingly multilingual. NVIDIA's Magpie-TTS-Multilingual, the base model of this work, speaks twelve languages out of the box at 357M parameters.

And yet, from the perspective of a language *not* on that list, such a model is a locked door. Production multilingual codec-LMs are trained on tens of thousands of hours of curated speech with data pipelines and compute budgets that neither individual researchers nor, more importantly, the speaker communities themselves can reproduce. The consequence is a widening *voice gap*: high-resource languages enjoy excellent synthetic speech, while thousands of languages spoken by hundreds of millions of people have none. Swahili (Kiswahili), a lingua franca for over one hundred million speakers across East Africa, is a canonical example of a mid-resource language that is nonetheless absent from most commercial TTS systems.

The seemingly obvious remedy, fine-tuning the multilingual model on the new language, backfires. Because all languages share one decoder and one embedding space, the gradient updates that specialize the model for the newcomer overwrite the representations of the incumbents. This is *catastrophic forgetting* [3, 4]. In our own early experiments, a Swahili fine-tune that reused an existing tokenizer slot produced excellent Swahili but silently *broke Korean* and degraded several other languages. Shipping such a model would trade one new voice for twelve damaged ones, an unacceptable bargain for a *unified*, openly released checkpoint.

We therefore treat "adding a language to a frozen codec-LM" as a *continual-learning* problem and give a simple, reproducible recipe that solves it. Our contributions are:

1. **Warm-started tokenizer surgery.** We add a *dedicated* byte-level tokenizer for the new language and perform an exact input-embedding resize. Crucially, the

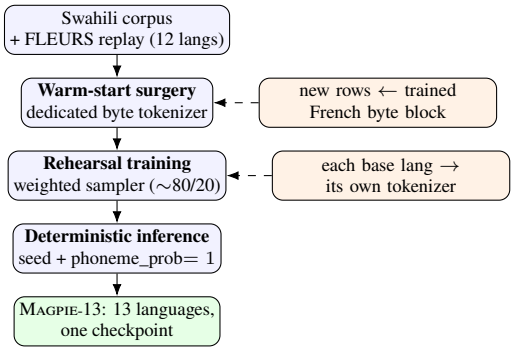

Figure 1: End-to-end pipeline. A frozen 12-language codec-LM is extended to a 13th language by warm-started embedding surgery and multilingual rehearsal, then served with a deterministic inference configuration.

new rows are **warm-started** by copying a trained byte-tokenizer block, not random-initialized. We show this single choice is the dominant quality lever (§5.6), moving Swahili from 77.8% to 8.8% CER. Because a codec-LM predicts *audio* tokens, only the input embedding grows; the output head is untouched.

2. **Multilingual rehearsal.** We interleave the new-language data with a thin replay slice of each base language, *each routed to its own native tokenizer*, and a weighted sampler that keeps the newcomer dominant while revisiting every incumbent each epoch.

3. **Reproducible, deterministic inference.** We identify two sources of run-to-run non-determinism (stochastic sampling and stochastic phoneme/grapheme tokenization) and give a configuration, matching the base model's inference settings, that makes generation bit-reproducible.

4. **A calibrated low-resource metric.** We demonstrate that Whisper-large-v3 is unreliable for Swahili (45.6% CER floor on *real human* speech) and adopt a massively-multilingual recognizer instead.

5. **An open release on commodity hardware.** The full pipeline (base restore, surgery, rehearsal training, evaluation) runs on one 128 GB unified-memory accelerator; we publicly release the checkpoint and model card.

Our central empirical claim is a *zero-forgetting* result: the 13-language model matches the base on all twelve original languages while adding a high-quality thirteenth, at a small fraction of the original training cost. Figure 1 summarizes the pipeline.

2. BACKGROUND AND RELATED WORK

2.1. Codec speech language models

Neural audio codecs compress a waveform into a short sequence of discrete tokens [5], recasting TTS as sequence-to-sequence token prediction amenable to Transformer language models [1]. Our base model belongs to the **Koel-TTS** family [2], which maps text and a context (reference) audio to acoustic tokens and sharpens transcript adherence and speaker fidelity via *classifier-free guidance* [6] and *preference alignment* guided by ASR and speaker-verification models. The audio tokenizer is the **Low Frame-rate Speech Codec** (LFSC) [7]: finite-scalar-quantized [8], adversarially trained, operating at **21.5 frames/s** and **1.89 kbps**. Its low token rate is what makes single-GPU fine-tuning and fast autoregressive inference practical. The specific checkpoint we extend is the 357M-parameter Magpie-TTS-Multilingual (revision `v2607`), a decoder-with-context-encoder variant with a set of *baked* speaker embeddings selected by an integer index, and a per-language text tokenizer selected by a language code [9].

2.2. Multilingual and low-resource TTS

Massively multilingual systems such as Meta MMS [10] scale ASR and TTS toward a thousand languages, and VITS-style models [11] remain strong per-language baselines. These lines either train from scratch per language or rely on very large supervised label sets. Our objective is orthogonal and complementary: to *graft* a language onto an existing high-quality multilingual model while *preserving* it, at negligible cost.

2.3. Continual learning and catastrophic forgetting

Catastrophic forgetting [3] is the loss of previously learned behaviour when a network is trained on new data. The classic remedies fall into three families: *regularization* (e.g. elastic weight consolidation [4]), *architectural isolation* (dedicating parameters to new tasks), and *rehearsal* (replaying old data). For a shared-decoder codec-LM, rehearsal is the most direct fit because we can synthesize an arbitrarily clean replay stream for every old language from public corpora. We combine rehearsal with architectural isolation *at the input* (a dedicated tokenizer and fresh, but warm-started, embedding rows), which localizes the newcomer while leaving the shared acoustic decoder free to be protected by replay.

2.4. Preference optimization

Following Koel-TTS [2], reinforcement-style preference optimization with ASR-CER and speaker-similarity rewards [12], using group-relative objectives [13] (a critic-free relative of DPO [14]), can further polish codec-LMs. We treat this as optional future work: our supervised recipe already reaches sub-fork error rates.

2.5. Tokenization for multilingual TTS

How text becomes input ids is decisive for a new language. Three regimes are common: (i) *grapheme* tokenizers (characters), simple but leaving all pronunciation to the model; (ii) *phoneme* tokenizers with a grapheme-to-phoneme (G2P) front-end, which inject linguistic priors but require a G2P resource per language; and (iii) *byte-level* tokenizers such as

byt5 [15], which are script- and language-agnostic and need no G2P. The base model mixes all three: IPA phoneme tokenizers for high-resource languages, character tokenizers for others, and byte tokenizers for several. For Swahili, whose Latin orthography is nearly phonemic, a byte-level tokenizer is both sufficient and convenient: it needs no G2P and shares the byte space that already exists in the model, which is exactly what makes our warm-start possible. A practical subtlety we return to in §3.5: the phoneme tokenizers carry a *stochastic* grapheme/phoneme mixing used for training augmentation, which must be disabled for reproducible inference.

2.6. Evaluating low-resource TTS

Intelligibility of synthetic speech is commonly measured by transcribing it with an ASR model and computing character/word error rate against the target text [16], complemented by mean opinion score (MOS) or its neural estimators [17] for naturalness and by speaker-verification cosine similarity [12] for identity. A hazard, rarely discussed, is that the *ASR itself* may be weak for the language under test: a high CER can then reflect recognizer error rather than synthesis error. We make this concrete for Swahili in §5.1, where Whisper's error *floor on real human speech* is 45.6%, and we adopt a stronger multilingual recognizer accordingly. This recognizer-calibration step should be standard practice for low-resource TTS evaluation.

2.7. Positioning: strategies for adding a language

Table 1 situates our recipe among the alternatives. Training a separate per-language model [11, 18] avoids forgetting by construction but yields no unified model and multiplies serving cost. Parameter-efficient adapters or LoRA [19, 20] add small trainable modules and can preserve the base, but require adapter routing at inference and still need a fresh embedding for a new script. The reused-slot fork produces excellent Swahili at near-zero cost but overwrites a donor language (Korean). Random-initializing a dedicated tokenizer preserves the base but fails on the newcomer. Our warm-started, dedicated-tokenizer recipe is the only option that is simultaneously *high-quality*, *base-preserving*, *cheap*, and *unified in one checkpoint*. It is also orthogonal to the base model's internal architecture (frame-stacked local transformers over LFSC codes [21, 7]), which we leave entirely unchanged.

Table 1: Strategies for adding a language to a multilingual TTS system. Only the warm-started dedicated tokenizer is high-quality, base-preserving, cheap, and unified at once.

Strategy	Quality	Base kept	Cheap	Unified
From-scratch per lang	high	n/a	✗	✗
Adapter / LoRA	varies	✓	✓	partial
Reused-slot fine-tune (fork)	high	✗	✓	✗
Dedicated, random-init	✗ poor	✓	✓	✓
Dedicated, warm-start (ours)	✓ high	✓	✓	✓

3. METHOD

3.1. Codec-LM preliminaries

Let a text sequence $x = (x_1, \ldots, x_L)$ be mapped by a language-specific tokenizer $\mathcal{T}_\ell$ to ids in a shared vocabulary of size V. An input embedding $E \in \mathbb{R}^{V \times d}$ (d=768) embeds these ids; the Transformer, conditioned on a baked speaker embedding s, autoregressively predicts LFSC audio tokens $\hat{a}_{1:T}$, which the codec decoder renders to 22.05 kHz audio. The key structural fact we exploit is that the *prediction target is the audio-token codebook*, which is *independent* of the text tokenizer $\mathcal{T}_\ell$; text enters the model *only* through E. Consequently, adding a language requires growing *only the input embedding*, never the output head. The base model ships V=3359 text tokens (the last two being the special BOS/EOS) and a map $\ell \mapsto \mathcal{T}_\ell$ over twelve languages, including three Arabic dialect tokenizers.

3.2. Adding a language via warm-started surgery

Swahili uses a Latin script with a highly transparent orthography, so a byte-level tokenizer [15] is a natural, grapheme-to-phoneme-free choice that also shares the byte space of several base tokenizers. We register a *new* tokenizer $\mathcal{T}_{\text{sw}}$ (rather than overwriting an existing slot, which is what broke Korean in our fork) and add the map $\texttt{sw} \mapsto \mathcal{T}_{\text{sw}}$. This grows the vocabulary from V to $V' = V + \Delta$ with Δ=384 Swahili byte tokens (V'=3743).

We resize $E \to E'$ under three constraints: (a) every pre-trained content row keeps its original index; (b) the special tokens, kept by the base model as the *last two* rows, are relocated to the new end; and (c) the new Swahili rows are **warm-started** from a trained byte-tokenizer block. All byt5 slots share an identical 384-token vocabulary in the same order, so we copy row-for-row from a Latin-script slot (French, at block offset o). With $T = V - 2$ pre-trained content rows:

$$E'_i = \begin{cases} E_i, & 0 \le i < T \\ E_{o+(i-T)}, & T \le i < T + \Delta \\ E_{V-2}, & i = V' - 2 \\ E_{V-1}, & i = V' - 1 \end{cases} \tag{1}$$

and we update the stored BOS/EOS indices accordingly. Case 2 is the crux. The obvious default, $E'_{T:T+\Delta} \sim \mathcal{N}(0, 0.02^2)$, leaves the Swahili input embeddings under-trained under the shared, rehearsed objective and yields *fluent-sounding but wrong words* (77.8% CER). Warm-starting from a trained Latin byte block gives the newcomer meaningful byte representations from step zero and reaches 8.8% (§5.6). Because the output head is not resized, the surgery is a strict superset of the base model; we verified that *before any training*, MAGPIE-13 still synthesizes all twelve base languages correctly; that is, the surgery is loss-less for incumbents. Figure 2 illustrates the operation.

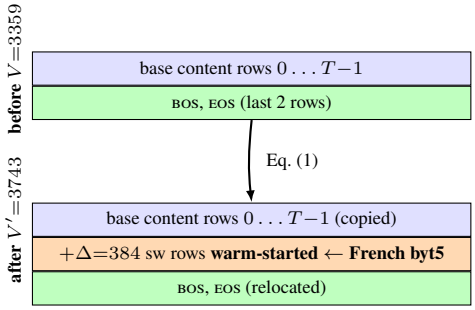

Figure 2: Input-embedding surgery. Base rows are copied in place, the special tokens are relocated to the new end, and the $\Delta=384$ Swahili byte rows are *warm-started* from the trained French byte block. The audio output head is unchanged.

3.3. Multilingual rehearsal against forgetting

Surgery isolates the newcomer at the input, but the shared Transformer is still updated by every gradient step and would drift from the base languages if trained on Swahili alone. We prevent drift with **rehearsal**: for each of the twelve base languages we assemble a small replay set from FLEURS [22] and interleave it with Swahili. Each replay set is *routed to that language's own native tokenizer* (English/German/Spanish/Portuguese/Hindi to their IPA phoneme tokenizers; Mandarin/Japanese to their G2P tokenizers; French/Italian/Vietnamese/Korean to byte tokenizers; and FLEURS Arabic to the MSA tokenizer), so rehearsal exercises exactly the pathways the base model uses.

Sampling uses a per-sample weighted sampler. If dataset k has n_k utterances with per-sample weight w_k, a sample is drawn from k with probability $p_k = n_k w_k / \sum_j n_j w_j$. The Swahili corpus ($n_{\text{sw}} \approx 13439$) already dwarfs each rehearsal set (n_ℓ=500), so equal weights give $p_{\text{sw}} \approx 69\%$. Our final model up-weights Swahili (w_{sw}=1.8, base w=1):

$$p_{\text{sw}} = \frac{1.8\, n_{\text{sw}}}{1.8\, n_{\text{sw}} + 12 \cdot 500} \approx 80\%, \tag{2}$$

so the twelve base languages share the remaining ≈20% (≈1.7% each). This gives the newcomer more of the decoder's capacity while keeping every incumbent present each epoch. Combined with the warm-start, this recovers Swahili to below the fork's error rate with no measurable base regression.

3.4. How much replay is enough?

The sampler draws 6000 examples per epoch. At p_{sw}≈0.8 this is ≈4800 Swahili and ≈1200 base examples per epoch, i.e. ≈100 per base language, so each base language, whose replay set holds 500 utterances, is revisited roughly every five epochs and seen ≈2000 times over the 20-epoch run, while Swahili accumulates ≈96,000 exposures (≈7 passes over its 13,439 utterances). This asymmetry is the design intent: the newcomer receives enough gradient signal to train its fresh embedding block and adapt the shared decoder, while a comparatively small but *non-vanishing* replay stream continually re-anchors each incumbent's decoder behaviour. The empirical outcome, 8.8% Swahili CER at Δ≈0 base regression (§5.6), indicates that ∼100 utterances/language/epoch of replay is already sufficient to prevent forgetting in this 357M model; establishing the *minimal* sufficient replay budget as a function of model size is an open question we leave to future work. Importantly, replay is cheap: the entire 12-language rehearsal corpus is ≈20 h (≈2.9 GB), two orders of magnitude smaller than the base model's original training data, yet enough to preserve all twelve languages.

3.5. Reproducible, deterministic inference

A production voice engine must be *stable*: the same request should return the same audio. We found two independent sources of run-to-run variation, and neutralize both while preserving the base model's inference behaviour. *(i) Stochastic sampling.* Codec-LM decoding samples with temperature and top-k (the base model's parameters: temp=0.7, top-k=80, CFG= 2.5); a fresh random draw per call changes prosody and pacing. We pin the RNG (Python, NumPy, Torch, CUDA) to a fixed seed before each request, so a given (text, voice, language, CFG, seed) is bit-reproducible; the seed is a user control for deliberately different takes. *(ii) Stochastic tokenization.* The IPA phoneme tokenizers carry a training-time augmentation, `phoneme_probability`=0.8, that randomly renders ∼20% of words as graphemes on each call, re-tokenizing the text differently every time. We set `phoneme_probability`=1.0 at inference (the intended deterministic setting), which byte tokenizers (Arabic, Swahili, …) never needed. With both fixes, repeated generations are byte-identical (max amplitude difference 0.0) across English, German, Mandarin, Arabic and Swahili.

3.6. Systems and stability

All stages run inside one container on a single **NVIDIA GB10** accelerator with **128 GB unified (CPU+GPU) memory**. Unified memory removes the VRAM ceiling but shares one pool, so we (i) cap container memory, (ii) set `num_workers=0` in the data loader (extra workers can exhaust the shared pool), and (iii) use a conservative schedule: batch size 1 with gradient accumulation 16 (effective batch 16), a 12 s duration cap, fp32. A self-healing supervisor resumes from the last checkpoint on any crash, and a watchdog kills and resumes a stalled run, so unattended multi-hour training survives transient faults. The full 20-epoch extension completes in ∼8 hours.

4. DATA

Swahili corpus (≈27.8 h). We combine (i) a clean read-speech corpus (literary Swahili, native 22.05 kHz, predominantly one speaker; 11,216 utterances) with (ii) the multi-speaker FLEURS `sw_ke` split (female and male, 16 kHz up-

Recipe: forgetting-free language extension

```
1: restore base codec-LM  (V = 3359 text tokens)
2: register dedicated byte tokenizer T_sw  (+384)
3: resize E→E' (Eq.1): copy base rows in place;
  warm-start sw rows ← French byt5 block;
  relocate BOS/EOS to the new end
4: build rehearsal: 500 utts/base lang (FLEURS),
  each routed to its own native tokenizer
5: weighted-sample (p_sw ≈ 0.8), train
  (batch 1 / accum 16, 12s cap, fp32, lr 1e-5, 20 ep)
6: select best epoch by validation loss
7: infer: fixed seed + phoneme_prob = 1 +
retry-on-silence
```

Figure 3: The complete recipe as executed on one 128 GB accelerator.

sampled; 1,413 female + 1,225 male). After filtering we obtain 13,439 training and 415 validation clips. Audio is resampled to 22.05 kHz mono, loudness-handled, and filtered by duration and characters-per-second; text is lightly normalized (Latin letters plus the `ng'` apostrophe). Each utterance is paired with a same-speaker context clip.

Rehearsal corpus (≈20 h). For each of the twelve base languages we stream FLEURS and keep 500 training and 40 validation utterances (1.5–1.9 h per language, ≈2.9 GB total), resampled to 22.05 kHz with identical filtering and same-speaker context pairing.

Licensing. FLEURS is CC-BY; the Swahili read-speech corpus is CC-BY; the base model is released under the NVIDIA Open Model License, which permits redistribution and commercial use with attribution. The unified checkpoint ships with the required license, notice, and base-model attribution.

5. EXPERIMENTS

5.1. Setup and metrics

Base languages. We score the twelve base languages with Whisper-large-v3 [16] CER, which is reliable for them (the untouched base scores 0.0% on ten of twelve fixed sentences).

Swahili: why not Whisper. Whisper is *not* a valid judge of Swahili. On *real human* Swahili its CER floor is **45.6%** (median 50.8%), frequently hallucinating unrelated text on clean audio (App. B). We therefore score Swahili with the massively-multilingual MMS recognizer [10], whose floor on the same clips is markedly lower (**37.5%** mean, 26.4% median; Fig. 4, leftmost bar) and near zero on clean clips. This metric caveat is itself a practical contribution for low-resource TTS evaluation: *practitioners must calibrate the recognizer against a ground-truth floor before trusting CER.* Formal listener MOS is left to future work; informal listening corroborates the CER ranking.

Evaluation set. One fixed sentence per base language, and three sentences × three baked voices (nine clips) for Swahili; the ground-truth floor uses twelve real human clips. Baselines: the *reused-slot fork* (our earlier single-language Swahili model that overwrote the Korean slot), a *random-init* ablation of our own architecture, and the *untouched base*.

Table 2: Swahili intelligibility (MMS Swahili ASR character error rate). "GT floor" = real human recordings scored by the same recognizer.

System	mean CER↓	median↓
Ground-truth (human floor)	37.5%	26.4%
Reused-slot fork	12.9%	2.5%
Random-init ablation	77.8%	76.3%
Magpie-13 (ours)	**8.8%**	**0.0%**

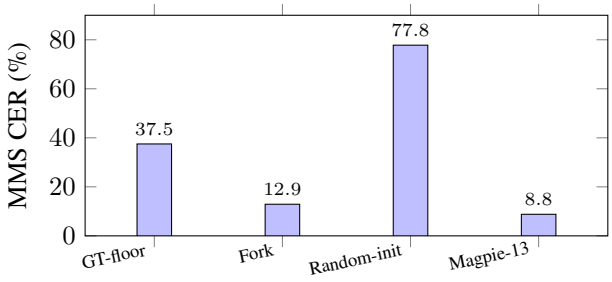

Figure 4: Swahili character error rate (lower is better). Warm-start (Magpie-13) beats both the recognizer's human-speech floor and the dedicated single-language fork, and dwarfs the random-init ablation.

5.2. Swahili quality

Table 2 and Fig. 4 report Swahili intelligibility (MMS CER). Magpie-13 reaches **8.8% mean / 0.0% median**, below the dedicated single-language fork (12.9%/2.5%) and far below the random-init ablation (77.8%). A median of 0.0% means at least half the utterances transcribe perfectly; Magpie-13 sits at or below the recognizer's floor on real human recordings, i.e. its synthetic Swahili is as intelligible to the recognizer as genuine speech.

5.3. Qualitative analysis

The aggregate CER hides *how* the systems fail. Table 3 shows the MMS transcription of the same reference sentence for all three systems. The random-init model emits fluent Swahili *phonation* but essentially random words, the tell-tale signature of under-trained input embeddings: the acoustic decoder works, but the text→sound map is noise. The fork and the warm-start model both recover the reference nearly verbatim. Across the nine-clip set the warm-start model's errors are confined to a single hard sentence (s3), and even there they are phonetic near-misses ("*ni na furahi*" for "*ninafurahi*"), not gibberish, consistent with its 0.0% median.

5.4. Zero-forgetting on the twelve base languages

Table 4 compares the untouched base with Magpie-13 on the same fixed sentence per language (Whisper CER). Magpie-

Table 3: MMS transcription of the reference "*habari ya asubuhi karibu kwenye jaribio la sauti ya kiswahili*" (voice 0). Random-init produces well-articulated nonsense; warm-start is verbatim.

System	Transcription (truncated)	CER
Random-init	*ukuwafugufia usiumaya utekuwadi-akondena…*	78.7%
Fork	habari ya asubuhi karibu kwenye jaribio…	0.0%
MAGPIE-13	habari ya asubuhi karibu kwenye jaribio…	**0.0%**

Table 4: Per-language Whisper CER (%): untouched base vs. MAGPIE-13. Δ near zero = no forgetting. †Raw single-shot generation occasionally hits the degenerate-silence glitch (§7); values are for valid (retry-on-silence) generations, which match the base.

Lang	Base	MAGPIE-13	Δ	Lang	Base	MAGPIE-13	Δ
en	0.0	0.0	0.0	zh	0.0	0.0	0.0
de†	0.0	0.0	0.0	hi	3.8	3.8	0.0
es	0.0	0.0	0.0	ja	0.0	0.0	0.0
fr	0.0	0.0	0.0	pt	0.0	0.0	0.0
it	0.0	0.0	0.0	ko	3.8	3.8	0.0
vi	0.0	0.0	0.0	ar	0.0	0.0	0.0
mean: base 0.6 MAGPIE-13 0.6 Δ 0.0							

13 matches the base everywhere: ten of twelve languages at 0.0%, and Korean and Hindi exactly at the base's own value. The mean CER is identical (0.6%), so $\Delta \approx 0$: **no measurable forgetting**. By contrast, the reused-slot fork, which produced good Swahili, *broke Korean* and degraded others, which is precisely why a dedicated tokenizer is required.

5.5. Training dynamics and the warm-start effect

Figure 5 plots validation loss per epoch for the random-init and warm-start runs. The random-init run plateaus near 9.94 and never learns intelligible Swahili (77.8% CER); the warm-start run, given trained byte embeddings and 80% Swahili weight, descends steadily to 9.627 and yields 8.8% CER. The gap is not a matter of more training: the random-init curve is flat by epoch 10, whereas warm-start keeps improving. This is the clearest evidence that the failure mode is *under-trained input embeddings*, cured at initialization rather than by additional steps.

5.6. Ablations

Table 5 isolates the two decisions that matter. *(a) Slot reuse vs. dedicated tokenizer:* reusing an existing byt5 slot gives good Swahili (12.9%) but is not a unified model: it overwrites and breaks the donor language (Korean); a dedicated tokenizer preserves all twelve base languages. *(b) Warm-start:* with a dedicated tokenizer, random-initializing the new rows is catastrophic for Swahili (77.8%): the input embeddings never fully train under the shared, rehearsed objective, whereas warm-starting them from a trained Latin byte block,

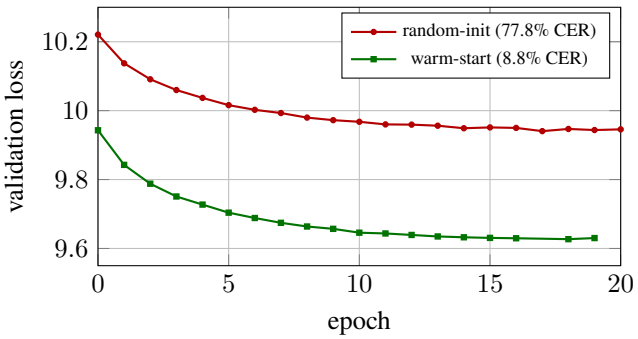

Figure 5: Validation loss per epoch. Warm-start (green) converges well below the random-init plateau (red); the resulting Swahili CER is 8.8% vs 77.8%.

Table 5: Ablation. Warm-start is decisive; a dedicated tokenizer is what keeps the model *unified*. The fork's ✗ is because it overwrites (breaks) the donor language, Korean.

Variant	Sw. CER↓	Base kept?
Reused slot (fork), 100% sw	12.9%	✗
Dedicated, random-init, 69% sw	77.8%	✓
Dedicated, warm-start, 80% sw	**8.8%**	✓

with the 80% Swahili weight, reaches 8.8%. Warm-start is the single largest lever, and it is essentially free.

6. DEPLOYMENT

We serve MAGPIE-13 as a single interactive endpoint on one GB10 accelerator: one checkpoint answers requests for all thirteen languages, selected by a language code, with five baked voices selected by index. Two production concerns are addressed directly by the method. First, **stability**: the deterministic configuration of §3.5 guarantees that a repeated request returns byte-identical audio, which matters for caching, regression testing, and user trust; a user-facing seed control exposes deliberate variation without sacrificing reproducibility. Second, **robustness**: the retry-on-silence guard (§7) transparently re-samples the rare degenerate generation, so the endpoint never returns silence. Because the LFSC token rate is low (21.5 fps) and the model is only 357M parameters, synthesis runs at interactive latency on the single accelerator without batching. The same checkpoint, license, notice, and model card constitute the open release.

7. DISCUSSION AND LIMITATIONS

Model capacity. At 357M parameters the base model bounds absolute quality; our contribution is a *method* for extension, not a new from-scratch model. **Speaker control.** The open base ships baked speakers selected by index; zero-shot cloning is intentionally excluded, so MAGPIE-13 offers a fixed palette of Swahili-capable voices rather than arbitrary cloning. **Data.** Our Swahili corpus skews to read/literary

speech and one dominant speaker; FLEURS adds multi-speaker breadth but at upsampled 16 kHz. Broader speaker and domain coverage is the most promising route to higher naturalness. **Generation stability.** The autoregressive decoder occasionally produces a degenerate silent output; we mitigate with a retry-on-silence guard (regenerate at a new seed when RMS energy is near zero or the length cap is hit), and a preference-optimization pass could remove it at the model level. Notably, the affected language changes with the seed (Portuguese in one run, German in another), confirming a sampling instability rather than a language-specific defect. **Evaluation.** Our fixed sentence set is small but sufficient to establish the zero-forgetting and warm-start effects but not a full benchmark; a larger held-out set with formal MOS is future work. **Metric.** We reiterate that Whisper CER is unsafe for low-resource languages; its $\sim$46% floor on real Swahili would masquerade as model error.

8. BROADER IMPACT AND RESPONSIBLE USE

Extending TTS to underserved languages advances accessibility, education, and language preservation, and lowers a barrier that has historically excluded communities from voice technology. The same capability carries misuse risk (impersonation, disinformation). The base model emits an audio watermark and a synthetic-speech disclosure, which MAGPIE-13 inherits, and we release with a model card documenting intended and out-of-scope use. We comply with the base model's license (redistribution with license, notice, and attribution) and attribute all CC-BY data sources. Because the recipe is cheap and reproducible on a single accelerator, we hope it enables communities to extend voice technology to their own languages rather than waiting for it to be provided.

9. CONCLUSION AND FUTURE WORK

We showed that a frozen multilingual codec speech language model can be taught a new low-resource language, Swahili, *without forgetting* any of its twelve existing languages, using warm-started input-embedding surgery and multilingual rehearsal, served with a deterministic inference configuration, all on a single unified-memory GPU. Warm-start is the decisive, nearly-free lever, taking Swahili from 77.8% to 8.8% CER, below a dedicated single-language fine-tune, while the base languages show no measurable regression. Future work: a preference-optimization polish for naturalness and generation stability; formal MOS studies; expanded, more diverse Swahili data; and, most importantly, grafting many more languages from the long tail onto existing multilingual models, toward speech technology that *leaves no language unspoken*.

REPRODUCIBILITY

The recipe (tokenizer registration, warm-started surgery of Eq. (1), rehearsal construction, weighted sampler), all hyperparameters (Table 6), the frozen evaluation set, and the released checkpoint permit full reproduction. The checkpoint and model card are publicly released at `https://huggingface.co/infiniatechnologies/magpie-tts-13lang-357m`; the recipe is fully specified above (Eq. (1), Table 6, and §3), and the training and evaluation scripts are available from the authors on request.

Table 6: Final training configuration for MAGPIE-13.

Setting	Value
Base checkpoint	Magpie-TTS 357M (v2607)
Audio codec	LFSC 22.05 kHz, 1.89 kbps, 21.5 fps
New tokenizer	byte-level (byt5), Δ=384 rows
Warm-start donor	French byte-tokenizer block
Vocab $V \to V'$	$3359 \to 3743$ (d=768)
Batch / accum.	1 / 16 (eff. 16)
Max dur. / prec.	12 s / fp32
Learning rate	1×10^{-5}
Sampler steps/epoch	6000
Epochs	20 (best epoch 18, val 9.627)
Swahili weight	w_{sw}=1.8 ($p_{\text{sw}}\approx$80%)
Inference	temp 0.7, top-k 80, CFG 2.5, phoneme_prob 1.0, fixed seed
Hardware	1$\times$ NVIDIA GB10, 128 GB unified

A. EVALUATION SENTENCES

Swahili (3 sentences $\times$ 3 baked voices): (1) *Habari ya asubuhi. Karibu kwenye jaribio la sauti ya Kiswahili.* (2) *Teknolojia ya kubadilisha maandishi kuwa sauti inaendelea kuboreka kila siku.* (3) *Ninafurahi kukujulisha kwamba mfumo huu sasa unaweza kuzungumza Kiswahili vizuri.* **Base languages**: one fixed sentence each; the English probe is "*The quick brown fox jumps over the lazy dog near the river.*" and the other eleven are short single sentences in each language's native script, available from the authors so that all reported numbers reproduce.

B. RECOGNIZER FLOOR ON REAL SWAHILI

The recognizer "floor" is measured on twelve held-out *real human* recordings. Per-clip Whisper CER is strongly bimodal: $\sim$0–28% on clean clips versus 74–85% where the recognizer hallucinates unrelated text, giving **mean 45.6% / median 50.8%**. The stronger MMS recognizer lowers this to **37.5% / 26.4%** (near zero on clean clips). The high-error clips trace largely to imperfect text–audio alignment in the crowd-sourced source corpus, which inflates *any* recognizer. The lesson is methodological: a raw ASR-CER mean is unsafe for a low-resource language without per-clip inspection and recognizer calibration against this floor.

REFERENCES

[1] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023, vALL-E.

[2] S. Hussain, P. Neekhara, X. Yang, E. Casanova, S. Ghosh, M. T. Desta, R. Fejgin, R. Valle, and J. Li, "Koel-tts: Enhancing llm based speech generation with preference alignment and classifier free guidance," *arXiv preprint arXiv:2502.05236*, 2025, iCML Workshop on Machine Learning for Audio.

[3] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 114, no. 13, pp. 3521–3526, 2017.

[5] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[6] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[7] E. Casanova, R. Langman, P. Neekhara, S. Hussain, J. Li, S. Ghosh, A. Jukić, and S.-g. Lee, "Low frame-rate speech codec: a codec designed for fast high-quality speech llm training and inference," *arXiv preprint arXiv:2409.12117*, 2024.

[8] F. Mentzer, D. Minnen, E. Agustsson, and M. Tschannen, "Finite scalar quantization: VQ-VAE made simple," *International Conference on Learning Representations (ICLR)*, 2024.

[9] NVIDIA, "Magpie-TTS-Multilingual: A multilingual autoregressive codec text-to-speech model," https://huggingface.co/nvidia/magpie_tts_multilingual_357m, 2026, model card, revision v2607; NVIDIA Open Model License.

[10] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu *et al.*, "Scaling speech technology to 1,000+ languages," *arXiv preprint arXiv:2305.13516*, 2023, meta Massively Multilingual Speech (MMS).

[11] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *International Conference on Machine Learning (ICML)*, 2021.

[12] N. R. Koluguri, T. Park, and B. Ginsburg, "TitaNet: Neural model for speaker representation with 1D depth-wise separable convolutions and global context," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[13] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song *et al.*, "DeepSeekMath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024, introduces Group Relative Policy Optimization (GRPO).

[14] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[15] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "ByT5: Towards a token-free future with pre-trained byte-to-byte models," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.

[16] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *International Conference on Machine Learning (ICML)*, 2023.

[17] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022," *Interspeech*, 2022.

[18] E. Casanova, K. Davis, E. Gölge, G. Göknar, I. Gulea, L. Hart *et al.*, "XTTS: a massively multilingual zero-shot text-to-speech model," *Interspeech*, 2024.

[19] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *International Conference on Machine Learning (ICML)*, 2019.

[20] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *International Conference on Learning Representations (ICLR)*, 2022.

[21] R. Fejgin, P. Neekhara, X. Yang, E. Casanova, R. Langman, J. Kim, S. Ghosh, S. Hussain, and J. Li, "Frame-stacked local transformers for efficient multi-codebook speech generation," *arXiv preprint arXiv:2509.19592*, 2026.

[22] A. Conneau, M. Ma, S. Khurana, Y. Zhang, D. Yu, Y. Wang, N. Ma, P. Baljekar, D. van Esch *et al.*, "FLEURS: Few-shot learning evaluation of universal representations of speech," *arXiv preprint arXiv:2205.12446*, 2022.