Audio8 TTS Preview 0.6B arrived with an unusually attractive promise: multilingual speech and zero-shot voice cloning from a compact 601-million-parameter model.

The practical question is simpler. Can it run on an ordinary Apple Silicon laptop as well as a consumer NVIDIA GPU, and what does the first real output tell us?

We pinned the model and source code, reviewed the custom Python files, and ran text-to-speech and synthetic-reference voice-cloning probes on a 16 GB M2 MacBook Pro and an RTX 3090 Ti. We then added a small LoRA training path and adapted the model on 128 cleaned clips from our NSC-derived FEMALE_01 voice profile.

60-second takeaway

- **It ran on the 16 GB M2 Mac.** CPU and MPS both produced valid 44.1 kHz audio without swap activity in our short tests.
- **It fit easily on the RTX 3090 Ti.** The highest observed whole-GPU memory reading was 2,938 MB during a two-item English and Chinese batch.
- **The short speech was textually clear.** Whisper large-v3-turbo recovered all seven tested outputs with 0.0 WER or CER after punctuation-insensitive normalization.
- **MPS was not a clear speed win.** It finished slightly sooner end to end, but its measured generation loop was not faster than CPU in these single short runs.
- **Voice cloning worked, but the quality verdict remains open.** We used a synthetic macOS reference rather than a person's voice. ECAPA speaker similarity landed between 0.564 and 0.604, which is a diagnostic signal rather than a human listening score.
- **The 128-clip LoRA pilot moved toward the target voice.** Mean ECAPA cosine similarity against 16 paired FEMALE_01 recordings increased from 0.092 for the base model to 0.189 after 100 steps.
- **Lower validation loss hid a serious failure.** A 500-step run kept improving on validation loss, but later checkpoints often failed to stop speaking. Step 300 failed to end normally on 12 of 16 prompts.
- **The model is promising, not production-proven.** Blind listening, larger clean datasets, long passages, and broader checkpoint tests remain necessary.

What Audio8 TTS Preview 0.6B is

Audio8 uses a DualAR design inspired by Fish Audio S2 Pro.

The slow autoregressive transformer predicts one semantic token for each audio frame. A smaller fast transformer predicts the remaining codec codebooks for that frame. The bundled neural codec then turns those tokens into a 44.1 kHz waveform.

Component	Released configuration
Main model	601,159,424 parameters, excluding the codec
Slow transformer	24 layers, width 896
Fast transformer	4 layers, width 896
Acoustic representation	10 codebooks with 4,096 entries each
Frame rate	About 21.5 frames per second
Output	44.1 kHz audio
License	Apache 2.0

The model card lists Cantonese, Chinese, Dutch, English, French, German, Italian, Japanese, Korean, Polish, and Spanish as its recommended languages. We only generated English and Chinese in this first test.

What we tested

We used exact revisions so that a later change to the repository does not silently change the benchmark:

- model: Audio8/Audio8-TTS-Preview-0.6b at 1b17c91db5f4dccb6914aa4aa5cb0e56661a6c17
- code: Audio8-AI/Audio8_TTS at 3346560df718d33096ac2fef7e5c7984ee5248e6

The upstream README still points to an older AutoArk-AI namespace. The live model we tested is under Audio8.

Hardware and runtime

Host	Runtime
Mac	M2 MacBook Pro, 16 GB unified memory, Python 3.12.13, PyTorch 2.13.0, Transformers 4.57.6

GPU host	RTX 3090 Ti 24 GB, PyTorch 2.6.0+cu124, Transformers 4.57.6

The Mac ran float32 on CPU and MPS. The NVIDIA host ran BF16 on CUDA.

Prompts

The GPU no-reference batch contained one English sentence and one Chinese sentence:

> A useful voice model must sound clear, run efficiently, and preserve the meaning of every sentence.

The Mac no-reference probe used the shorter sentence:

> A useful voice model must sound clear.

For voice cloning, we created a reference with the built-in macOS Samantha voice. This avoided copying a person's identity into a public benchmark.

Reference:

> The morning train arrived exactly on time.

Target:

> A small team can turn a clear idea into a useful product.

Every run used seed 42, temperature 0.8, top-p 0.95, and top-k 50.

Results on the 16 GB M2 Mac

Probe	Device	Total time	Audio duration	Maximum RSS	Peak memory footprint	Swaps
Short English	CPU float32	20.82 s	2.833 s	3.25 GB	5.48 GB	0
Short English	MPS float32	19.45 s	2.833 s	4.07 GB	6.96 GB	0
Synthetic-reference	CPU float32	20.25 s	3.762 s	5.63 GB	5.94 GB	0

clone						
Synthetic-reference clone	MPS float32	19.93 s	3.529 s	4.07 GB	7.00 GB	0

Both paths worked. That is the important result.

The timing difference is too small and the sample count is too low to call MPS faster. In fact, the progress timer around generation was slightly slower on MPS than CPU. Model loading and other setup made the total MPS run slightly shorter.

For this release, a 16 GB Mac is a workable local experimentation machine. It is not a real-time TTS host on the tested upstream PyTorch path. The short runs took roughly 5 to 7 times the generated audio duration when model loading was included.

Results on the RTX 3090 Ti

Probe	Total time	Generated audio	Highest observed GPU memory
English and Chinese, two-item batch	13.969 s	13.235 s	2,938 MB
Synthetic-reference clone	7.423 s	3.622 s	2,612 MB

These times include loading the model inside each command. They are not warmed-server latency measurements.

The bilingual batch produced about as much audio as the command took to finish. That makes the GPU path much more practical than the Mac path for repeated content production, but it does not establish concurrent serving capacity.

Did it say the right words?

We ran every output through Whisper large-v3-turbo, then calculated English word error rate and Chinese character error rate after removing punctuation differences.

Checked output	Result
GPU English, no reference	0.0 WER
GPU Chinese, no reference	0.0 CER
GPU synthetic clone	0.0 WER
Mac CPU and MPS, no reference	0.0 WER for both
Mac CPU and MPS, synthetic clone	0.0 WER for both

This is a useful but narrow result. It means one independent ASR model recovered the target text exactly from these short files. It does not tell us whether the voice sounded natural, expressive, pleasant, or free from subtle artifacts.

How closely did the clone follow the synthetic voice?

We compared the three clone outputs with the synthetic macOS reference using SpeechBrain's ECAPA-TDNN speaker encoder.

Clone	ECAPA cosine similarity
RTX 3090 Ti BF16	0.585
Mac CPU float32	0.564
Mac MPS float32	0.604

Do not read this table as a human voice-quality ranking.

The reference is only 2.43 seconds long. It is synthetic. The target sentence differs from the reference sentence. One speaker encoder can also reward or punish acoustic features that people notice differently.

The result establishes that the voice-cloning path ran and preserved a measurable amount of speaker information. A blind listening test is still needed before choosing the model for a real product voice.

What happened when we fine-tuned Audio8

The inference test answered whether Audio8 could run on our machines. The next question was whether a small adapter could move the model toward a voice profile we already use for TTS research.

We used the cleaned FEMALE_01 split derived from the IMDA National Speech Corpus. FEMALE_01 is a dataset namespace and voice profile. We have not established that every recording came from one person, so we do not describe this as a verified single-speaker clone.

For the first pilot, we selected:

- 128 training clips containing 535.6 seconds of speech
- 32 held-out validation clips containing 141.1 seconds of speech
- clips between 1.5 and 8 seconds long
- no empty transcripts in either selected manifest
- seed 42 for repeatable selection and training order

The model's audio codec encoded all 160 clips successfully. We then trained a rank-8 LoRA adapter on the RTX 3090 Ti.

LoRA is not part of the checked upstream repository. We added a small PEFT extension around the released supervised fine-tuning script. The upstream path also needed three corrections before it would train in our environment:

- gradient checkpointing is enabled by default, but the released `ArkttsModel` does not support it
- Accelerate 1.1.1 was incompatible with the installed Transformers trainer; Accelerate 1.14.0 worked
- the documented launcher points to a missing `configs/deepspeed_zero2.json` file

Our successful runs disabled gradient checkpointing, upgraded the isolated environment, and called the Python trainer without DeepSpeed. This means the released fine-tuning code is a useful starting point, but it was not ready to run exactly as documented in our checked revision.

Training setting	Value
Trainable parameters	4,730,880, or 0.78% of the model
Precision	BF16
Batch size	1
Maximum sequence length	512
Learning rate	0.00002
Target modules	`wqkv`, `wo`, `w1`, `w2`, `w3`
Selected checkpoint	Step 100
Highest sampled whole-GPU memory	12,091 MB, including existing host use

The 100-step training loop took 11.31 seconds. Including evaluation and export, the command took 18.46 seconds. These are single-host measurements, not general training-speed claims.

Did the adapter move toward FEMALE_01?

We generated the same 16 held-out prompts with the base model and the step-100 adapter. Each prompt also had a paired FEMALE_01 recording for comparison.

Measurement	Base model	Step-100 LoRA
Prompts ending normally	16 of 16	16 of 16
ASR word errors	0 of 85	1 of 85
Normalized exact ASR matches	16 of 16	15 of 16
Mean ECAPA similarity to paired FEMALE_01 recording	0.092	0.189

The adapted output had higher ECAPA similarity on 15 of 16 prompts. That is evidence of early movement toward the target voice profile, not proof that listeners will prefer it.

ASR only checks whether an independent model can recover the words. ECAPA is a speaker-embedding proxy trained on another corpus. The paired reference also contains the same words as the generated clip, so shared phonetic content may affect the score. We therefore prepared a blind, level-matched listening comparison before considering a larger run.

Why we rejected the checkpoint with the lowest loss

We also ran a fresh 500-step schedule. Validation loss fell from 10.777 at step 50 to 9.101 at step 500. If we had selected a checkpoint from loss alone, step 500 would have won.

Generation showed the opposite.

Checkpoint	One-prompt end-of-speech probe
100	Ended normally
150	Did not stop by 500 frames
200	Did not stop by 500 frames

250 Ended normally

300 Ended normally

350 Did not stop by 500 frames

400 Did not stop by 500 frames

450 Did not stop by 500 frames

500 Did not stop by 500 frames

We gave step 300 a broader chance because it passed the first probe and had lower validation loss than step 100. Across all 16 prompts, only 4 ended normally. The remaining 12 reached the 500-frame cap. The final step-500 export produced roughly 93 seconds of audio on its first prompt before we stopped that batch.

The practical lesson is clear: validation loss is necessary but insufficient for selecting an autoregressive TTS checkpoint. Every candidate also needs held-out generation checks for correct words, normal duration, end-of-speech behavior, voice similarity, and human listening quality.

Installation notes that matter

The released model contains about 2.4 GB of files on disk in our remote checkout. The main model is compact, but the bundled 44.1 kHz codec accounts for a large part of the download.

The model requires Python 3.10 or newer and Transformers 4.57.x:

```
python3 -m venv .venv
source .venv/bin/activate
pip install "torch>=2.5.0" "torchaudio>=2.5.0" \
  "transformers>=4.57.0,&lt;5" "soundfile>=0.12" \
  "safetensors>=0.4" "tqdm>=4.66"
```

The model also requires `trust_remote_code=True`. Review the pinned custom Python files before loading them.

In our bounded review, we checked the model configuration, processor, model, and codec files for shell commands, subprocess execution, explicit networking, file deletion, broad writes, dynamic `eval` or `exec`, and unsafe checkpoint loading. We did not find suspicious operations in those files. The codec loader uses `torch.load(..., weights_only=True)`.

That is not a complete dependency or supply-chain audit. Pin the exact revision rather than trusting future remote code automatically.

How to interpret the upstream benchmark

Audio8 reports strong Seed-TTS and CV3 results for a model of this size. Those figures are worth studying, but we did not reproduce those benchmark suites.

The model card itself notes that some baseline numbers came from other official reports, Fish S2 Pro used its own normalizer, and Higgs Audio v2 was evaluated locally. Different normalizers and evaluators make those numbers useful references rather than a strict matched ranking.

Our first-party evidence is narrower:

- the pinned model loads and generates on the two tested hosts
- short English and Chinese outputs were recovered exactly by one ASR system
- short synthetic-reference cloning completed on CPU, MPS, and CUDA
- the 24 GB GPU had substantial memory headroom in these probes

Where Audio8 fits today

Need	Current reading
Experiment locally on a 16 GB Mac	Viable for short offline runs
Generate repeated narration on a consumer GPU	Promising, based on low observed memory and near-real-time batch completion
Zero-shot voice cloning	Technically working, quality selection still pending
Long-form production narration	Not established
Replace our selected IndexTTS2 voice	Not justified by this first test
LoRA fine-tune on a small NSC-derived slice	Technically viable; early voice-profile movement measured at step 100
Select checkpoints from validation loss alone	Unsafe; later low-loss checkpoints often failed to stop

Audio8 is most interesting as a compact DualAR research and product candidate. It brings a Fish Speech style architecture down to a much smaller main model, keeps an Apache 2.0 license, includes both inference and supervised fine-tuning tools, and runs on hardware we already own.

That combination earns it a place in the next comparison. It does not yet earn a production promotion.

What we would test next

1. Complete the blind comparison between the base model, step-100 adapter, and FEMALE_01 references before scaling the dataset.
2. If listeners confirm the measured movement, expand the fixed training subset while preserving the same held-out prompts and end-of-speech checks.
3. Compare the selected Audio8 checkpoint against IndexTTS2, Qwen3-TTS, Supertonic, and Fish Speech using matched text.
4. Add names, numbers, abbreviations, Singapore English, Mandarin, Cantonese, and code-switching prompts.
5. Test 30-second, 2-minute, and 5-minute passages for repetition, skipped words, termination, and pacing drift.
6. Review the source and dependency chain more deeply before any production use of remote custom code.

Bottom line

Audio8 TTS Preview 0.6B passed the first practical inference and adaptation tests.

It generated clear short English and Chinese speech on an RTX 3090 Ti, and it completed both ordinary synthesis and synthetic-reference voice cloning on a 16 GB M2 Mac through CPU and MPS. The short samples used manageable memory, produced valid 44.1 kHz WAV files, and preserved the target text well enough for Whisper large-v3-turbo to recover every sentence exactly.

The 128-clip LoRA pilot also moved the model toward our NSC-derived voice profile on one speaker-embedding proxy. But longer training exposed unstable end-of-speech behavior even as validation loss improved. That failure changes

how we evaluate this model: a saved checkpoint is not a usable voice until it passes generation and listening tests.

The model is still a Preview release. Our evidence does not yet establish naturalness against alternatives, perceived voice similarity, long-form reliability, or production concurrency.

The right conclusion is not that Audio8 is already the best compact TTS model. It is that Audio8 can run and adapt on hardware we own, and that its step-100 LoRA checkpoint deserves blind listening before we spend more compute on a larger training set.

Related Instavar research

- Which TTS Model Should You Use? A Decision Tree (2026)
- How Open-Source TTS Architectures Differ
- Voice Cloning on a 24GB GPU
- Supertonic 3 On-Device TTS Reality Check on macOS
- LoRA Fine-Tuning Qwen3-TTS for Custom Voices

Sources

- Audio8 TTS Preview 0.6B model card
- Audio8 TTS source repository
- Apache License 2.0 in the upstream repository
- First-party benchmark evidence: reports/benchmarks/audio8-tts-preview-0.6b-macos-rtx3090ti-2026-07-30.md