60-second takeaway
VoxCPM 1.5, Qwen3-TTS 1.7B, IndexTTS2, and CosyVoice3 all fit within 24GB VRAM on an RTX 3090 Ti for both training and inference.
The real constraint is not memory - it is having a working recipe. LoRA paths (VoxCPM, Qwen3-TTS) have the most mature recipes. Full SFT paths (IndexTTS2, CosyVoice3) work but need explicit checkpoint management.
If you have a 24GB GPU and want to start today, VoxCPM 1.5 LoRA is the path of least resistance.

Who this is for

This guide is for engineers who have a single consumer or prosumer GPU (RTX 3090, RTX 3090 Ti, RTX 4090, or similar 24GB class) and want to fine-tune a TTS model for custom voice cloning. We ran all benchmarks on an RTX 3090 Ti (24 GB VRAM) on a Tailscale-connected remote desktop.

The question we're answering: **which open-source TTS models can you actually run - training and inference - on a single 24GB GPU in 2026?**

The short answer

Model	VRAM fit (24GB)	Training mode	Recipe maturity	Deployable result?
VoxCPM 1.5	Fits	LoRA	Mature	Yes
Qwen3-TTS 1.7B	Fits	LoRA	Mature	Yes
IndexTTS2	Fits	Full SFT	Mature (with recovery)	Yes
CosyVoice3	Fits	Full SFT via LoRA path	Available	Rerun pending

All four models fit. The differentiator is not memory - it is recipe stability and checkpoint management discipline.

VoxCPM 1.5 on 24GB

VRAM profile: Fits comfortably within 24GB for both LoRA training and inference.

Training mode: LoRA fine-tuning. The 44.1 kHz audio prep requirement is the main setup step - resample your dataset before training.

Recipe: Standard LoRA train/val split. No custom modifications required for 24GB. Training to step 9000 is feasible in a single session.

What to watch:

- Audio resampling to 44.1 kHz is mandatory. Skip this and training diverges.
- Validation loss is a reliable guide here - step 4000 was the best in our benchmark, but your dataset may differ.
- No-prompt inference is more stable than prompted inference for clean output.

Expected runtime: Training to step 9000 on FEMALE_01 completed in a single GPU session without memory pressure.

Qwen3-TTS 1.7B on 24GB

VRAM profile: Fits within 24GB for LoRA training and inference.

Training mode: LoRA. Requires JSONL dataset prep and codec preprocessing before training. The `sft_12hz.py` script handles the codec path.

Recipe: LoRA train/val/test split. The `lora_scale` parameter at inference time is the key tuning knob - not just checkpoint selection.

What to watch:

- Run the codec preprocessing step before training. Skipping it causes silent failures.
- Use SDPA (Scaled Dot-Product Attention) backend for inference - it reduces VRAM pressure and improves long-text stability.
- Scale sweep at inference: test 0.2, 0.3, 0.35, 0.5. Scale 1.0 almost always over-steers.
- Deterministic decode (fixed seed) is required for reproducible listening comparisons.

Expected runtime: Training to epoch 10 on FEMALE_01 fits within one GPU session. The codec prep adds ~15 minutes of CPU time upfront.

IndexTTS2 on 24GB

VRAM profile: Fits within 24GB for full SFT training and inference.

Training mode: Full SFT with resume support. This is more memory-intensive than LoRA paths but still fits 24GB without quantisation.

Recipe: Process manifests (`FEMALE_01_44k` format)   full SFT with explicit resume management. The training loop has a crash-prone resume path in some versions - use explicit checkpoint save paths and keep crash logs.

What to watch:

- Do not rely on automatic checkpoint retention. Keep all checkpoints manually until you have done a listening sweep.
- The best validation region in our run was around step ~13800, but the nearest saved checkpoint was step 14000. This is typical - save more frequently than you think you need to.
- Crashes during training are recoverable with the right resume path. Keep detailed logs.

Expected runtime: Training to step 15000+ was achievable on 24GB, but required crash recovery in our run.

CosyVoice3 on 24GB

VRAM profile: Fits within 24GB for the full SFT LoRA path.

Training mode: Full SFT via `train_cosyvoice3_lora.py`. The CosyVoice team provides explicit LoRA tooling for the 3.x line.

Recipe: Available (`train_cosyvoice3_lora.py`, `infer_cosyvoice3_lora.py`). The recipe works at the hardware level - the issues in our run were checkpoint management and prompt handling, not VRAM.

What to watch:

- Our first run did not produce production-ready output. See CosyVoice LoRA Fine-Tuning: What Worked, What Didn't for the full diagnostics.
- Tighter checkpoint gating (explicit save every N epochs) is required before the run can be evaluated properly.
- Long-text generation (>20s) was unstable in the current run configuration.

Expected runtime: Training fits within 24GB, but a proper checkpoint gating discipline adds setup time before the first reliable evaluation.

Hardware notes: RTX 3090 Ti specifics

All runs were on an RTX 3090 Ti with 24 GB GDDR6X. A few GPU-specific observations:

- **Thermal throttling:** Long training runs (4+ hours) on the 3090 Ti can trigger thermal throttling under poor airflow. Monitor GPU temperature and ensure adequate case ventilation.
- **Memory bandwidth:** The 3090 Ti's 936 GB/s bandwidth is a significant advantage for full SFT runs over the standard 3090's 936 GB/s (they are effectively the same here). The real difference vs a 3090 is power consumption and slight clock boost.
- **A100/H100 comparison:** These 24GB consumer runs are roughly 2–4x slower than equivalent runs on an A100 80GB. For production-scale fine-tuning (larger datasets, more epochs), a cloud A100 is significantly faster. The 24GB path is viable for prototyping and single-speaker benchmarking.

Practical setup checklist

Before starting any of these runs on a 24GB GPU:

Verify CUDA version matches model requirements (check README)
Pre-process and resample dataset to model-required sample rate
Set explicit checkpoint save paths (do not rely on defaults)
Confirm available disk space (full SFT checkpoints are 1–5 GB each)
Set up crash recovery / resume path before starting long runs
Run a 10-minute smoke test (1 epoch, small batch) before committing to full training
Keep a training log for each run (model, dataset, LR, steps, VRAM peak)

FAQ

Can I use a 16GB GPU (RTX 3080, RTX 4080) instead?

Not without modification. VoxCPM 1.5 and Qwen3-TTS LoRA may be achievable with reduced batch size and gradient checkpointing, but we have not tested this. IndexTTS2 and CosyVoice3 full SFT are unlikely to fit 16GB without quantisation.

How long does a full benchmark run take on a 3090 Ti?

Rough estimates for FEMALE_01-scale datasets (single speaker, ~2–5 hours of audio):

- VoxCPM LoRA to step 9000: 3–6 hours
- Qwen3-TTS LoRA to epoch 10: 4–8 hours
- IndexTTS2 full SFT to step 15000: 8–16 hours (with crash recovery overhead)
- CosyVoice3 full SFT: similar to IndexTTS2, plus checkpoint gating overhead

Is LoRA always better than full SFT for 24GB runs?

Not necessarily. LoRA is faster and uses less VRAM, but full SFT can produce more stable long-form output for some voice profiles. In our benchmark, the LoRA models (VoxCPM, Qwen3-TTS) produced deployable results with less friction. Full SFT (IndexTTS2) also worked but required more operational discipline. Choose based on your iteration speed requirements.

What is IMDA NSC FEMALE_01?

IMDA NSC is Singapore's National Speech Corpus. `FEMALE_01` is a single-speaker set with natural Singaporean English. We use it as a benchmark because the accent profile stress-tests speaker similarity in voice cloning. See IMDA NSC Voice Cloning Finetuning Benchmark 2026 for the full methodology.

Sources and related posts

- Full benchmark results: Best Open-Source TTS Models for Production in 2026
- Full benchmark matrix: IMDA NSC Voice Cloning Finetuning Benchmark 2026
- CosyVoice diagnostics: CosyVoice LoRA Fine-Tuning: What Worked, What Didn't
- VoxCPM run notes: VoxCPM 1.5 LoRA Finetuning on IMDA NSC FEMALE_01
- Qwen3-TTS run notes: Qwen3-TTS LoRA Fine-Tuning: Scale Sweeps, Checkpoints, and Production Defaults
- IndexTTS2 run notes: IndexTTS2 Finetuning on IMDA NSC FEMALE_01