60-second takeaway
There are now nine credible open-source TTS models you could deploy. The problem is not finding one that works - it is picking the one that fits your constraints.
We benchmarked most of them on IMDA NSC FEMALE_01 using an RTX 3090 Ti (24GB). This article gives you a decision tree: start with your use case (real-time streaming, audiobook, edge deployment, multilingual, fine-tuned voice cloning) and land on a specific model with a specific configuration.
If you just want a quick answer: Qwen3-TTS for real-time, CosyVoice 3 or VoxCPM 1.5 for pre-produced content, Chatterbox for fast fine-tuning, Kokoro for edge.

Where this fits

- **For founders:** Your team is about to pick a TTS model. This decision tree prevents the most expensive mistake - choosing a model that is technically impressive but wrong for your deployment constraints. A real-time product cannot tolerate CosyVoice 3's compute overhead. An audiobook pipeline does not need Qwen3-TTS's 97ms latency. Match the model to the use case before writing any integration code.
- **For engineers:** This is the routing logic we use internally. Each recommendation is grounded in first-party benchmarks - not paper claims, not leaderboard scores, not vibes. We include the specific checkpoints, LoRA scales, and failure modes we observed so you can reproduce or skip straight to deployment.

How to use this decision tree

Start from your use case, not from a model name. The models below overlap in capability - most of them can do voice cloning, most of them fit on a 24GB GPU, and most of them produce decent output in zero-shot mode. The differences emerge when you add constraints: latency budget, fine-tuning requirements, target language, or deployment hardware.

Read the decision tree table first. If your situation maps cleanly to one row, jump to that model's deep-dive section. If you are torn between two models, the deep-dive sections include the trade-offs we observed.

The decision tree

Your situation	Recommended model	Why
Need real-time streaming (< 100ms latency)	Qwen3-TTS 1.7B	97ms first-packet latency, robust to formatting variation
Pre-produced content (audiobook, video A-roll)	CosyVoice 3 (zero-shot) or VoxCPM 1.5 (fine-tuned)	Flow matching quality for zero-shot; VoxCPM if fine-tuning is needed
Deploy something today, minimal setup	VoxCPM 1.5	Lowest friction LoRA path, step 4000 deployable
Need LoRA adapter control post-training	Qwen3-TTS 1.7B	Scale parameter (0.3–0.35) tunes output without retraining
Full SFT for maximum voice fidelity	IndexTTS2	Most reproducible full-SFT baseline, best checkpoint at step 14000
Edge / on-device deployment	Kokoro (82M)	82M parameters, fraction of compute cost
Multilingual (EN + CN + JP)	Fish Speech S2 Pro	300K+ hours multilingual training data, ELO 1339 in TTS Arena, LoRA fine-tuned on FEMALE_01
Non-American English accent retention	VoxCPM 1.5 or IndexTTS2	Both retained IMDA NSC FEMALE_01 Singaporean English accent well
Fine-tuning with minimal reference audio	Qwen3-TTS 1.7B	3-second minimum, 10–15s optimal, then plateau
Fastest fine-tuning turnaround	Chatterbox (0.5B)	512 samples in 2 min 20s on a single GPU - fastest fine-tuning of any model here
Maximum expressiveness + voice cloning	Higgs Audio V2 (3B)	10M+ hours pre-training, top trending on HuggingFace, Llama 3.2 backbone

Real-time streaming: Qwen3-TTS 1.7B

If your product needs first-packet latency under 100ms, Qwen3-TTS is the only model in this list that reliably delivers it. We measured 97ms first-packet latency on our RTX 3090 Ti benchmark - fast enough for conversational interfaces, live dubbing, and interactive voice agents.

What makes it work for real-time:

- Streaming-native architecture - the model generates audio incrementally, not as a single batch
- 10 languages supported out of the box
- 3-second voice cloning - you can enrol a new speaker from a single short clip
- Robust to input formatting variation - handles punctuation, numbers, and abbreviations without special preprocessing

LoRA fine-tuning: Supported, and this is where the model shines post-training. The `lora_scale` parameter (0.3–0.35 optimal in our benchmark) lets you control how much the adapter influences output without retraining. This is a deployment-time knob, not a training-time decision. Run a 5-sample listening test at scales 0.2, 0.3, 0.35, and 0.5 before committing.

Current limitation: Single-speaker fine-tuning only. Multi-speaker LoRA is not yet supported. If you need multiple fine-tuned voices, you need separate adapters.

Known pitfalls (from our fine-tuning runs and companion repo):

Pitfall	Symptom	Fix
Double label-shift bug in `sft_12hz.py`	Speech progressively accelerates each epoch until unintelligible	Apply PR #178 - replace with `F.cross_entropy()` to avoid HuggingFace's internal shift
Missing `text_projection` call (line 93)	Hard crash on 0.6B model; silent wrong embeddings on 1.7B	Apply PR #188 (commit `680d4e9`)
Default LR too high (2e-5)	Pure noise output, infinite generation (no EOS), apparent divergence	Use **2e-6** instead (validated across GitHub issue #39)
Audio not at 24kHz	Crash deep in training with no early warning	Resample all audio to 24kHz before codec prep: `ffmpeg -i`

		in.wav -ar 24000 out.wav
LoRA scale 1.0 at inference	Over-steered, forced-sounding output	Use **0.3–0.35**; run 5-sample listening test before committing
EOS token failures (~0.5% of inferences)	Infinite token generation, hangs	Set explicit eos_token_id list and max_new_tokens cap
Cold-start decoder distortion	First inference in a new process produces corrupted audio	Prepend silence codec tokens as warm-up, then trim
Progressive timbre shift across chunks	Voice changes between long-text chunks	Fix random seed before each chunk; extract speaker embedding once and reuse
Val evaluation crash on small val sets	RuntimeError: zero-dimensional tensor cannot be concatenated	Bug in evaluation function - needs guard for empty loss tensor
Inference segfaults mid-epoch sweep	Process crashes partway through checkpoint evaluation	Batch inference defensively; do not assume a loop completes
Val loss plateaus after epoch 10	Train loss keeps dropping but val loss stalls at ~10.3	**Stop at epoch 10** - further training overfits without quality gain

The double label-shift bug is the most impactful: it affects every training run on the official script and is not documented in the upstream README. If your fine-tuned output sounds increasingly fast with each epoch, this is almost certainly the cause.

For the full fine-tuning walkthrough, see LoRA Finetuning Qwen3-TTS for Custom Voices.

Pre-produced content: CosyVoice 3 + VoxCPM 1.5

Pre-produced content (audiobooks, video narration, podcast intros) does not need sub-100ms latency. It needs the highest possible naturalness and consistency across long passages. Two models fit this use case, and which one you pick depends on whether you need fine-tuning.

CosyVoice 3 - best zero-shot quality

CosyVoice 3 uses a flow matching architecture that produces extremely consistent zero-shot output. If you have a reference clip and do not want to fine-tune, this is the model to start with.

Strengths:

- Flow matching produces smooth, natural prosody even on first attempt
- Extremely high speaker consistency in zero-shot mode
- Strong on long-form passages - audiobook chapters, 5-minute narration blocks

Trade-offs:

- Higher compute cost than Qwen3-TTS or VoxCPM at inference time
- Our fine-tuning run did not reach production quality (rerun pending) - if you need fine-tuning, use VoxCPM instead
- Not suitable for real-time streaming due to compute overhead

For the CosyVoice 3 benchmark details, see CosyVoice 2 vs CosyVoice 3 on IMDA NSC FEMALE_01.

VoxCPM 1.5 - best fine-tuned pre-produced quality

If zero-shot is not enough and you need a fine-tuned voice for pre-produced content, VoxCPM 1.5 is the path of least resistance.

Strengths:

- Lowest setup friction of any LoRA fine-tuning path we tested
- Best checkpoint (step 4000) produced deployable output in our first run
- No-prompt generation was the cleanest - prompted inference copied room noise from the reference clip
- Strong accent retention on Singaporean English (IMDA NSC FEMALE_01)

Trade-offs:

- LoRA only - no full SFT path if you need deeper model adaptation
- Requires 44.1 kHz audio resampling before training (skip this and training diverges)

Known pitfalls:

- **44.1 kHz resampling required** - VoxCPM expects 44.1 kHz audio (not 24 kHz like Qwen3-TTS). Skip this and training diverges silently - loss looks

normal but output quality degrades.
- **Prompted inference copies room noise** - if the reference clip has any background noise, it bleeds into the output. Use no-prompt generation for production; only use prompted mode when strong speaker lock is required.

For the full VoxCPM fine-tuning guide, see VoxCPM 1.5 LoRA Finetuning on IMDA NSC FEMALE_01.

Edge deployment: Kokoro

Kokoro is the outlier in this list - 82M parameters, compared to 1.7B for Qwen3-TTS. If your deployment target is a mobile device, an embedded system, or any environment where you cannot run a multi-billion-parameter model, Kokoro is the only viable option here.

What makes it work for edge:

- 82M parameters - fits on devices with limited RAM and no GPU
- Quality is comparable to larger models at a fraction of the compute cost
- Suitable for on-device voice assistants, kiosk applications, and offline scenarios

Trade-offs:

- Smaller model capacity means less flexibility for unusual prosody or complex text
- Fewer languages than Fish Speech S2 Pro or Qwen3-TTS
- Fine-tuning ecosystem is less mature than the larger models

When to use Kokoro over a cloud-hosted larger model: When latency to a server is unacceptable (offline scenarios, high-frequency short utterances), when compute budget per inference is extremely constrained, or when you need to keep audio generation entirely on-device for privacy reasons.

Multilingual requirements: Fish Speech S2 Pro

If your product needs to serve English, Chinese, and Japanese from a single model, Fish Speech S2 Pro is the strongest option. It was trained on 300K+ hours of multilingual data - an order of magnitude more than most open-source TTS models.

First-party experience: We fine-tuned Fish Speech S2 Pro with LoRA on IMDA NSC FEMALE_01 (March 2026). The model is a 4.6B parameter DualAR Transformer - only 18.4M parameters are trainable via LoRA, making fine-tuning feasible on a single 24GB GPU. The training pipeline requires three preparation steps (VQ extraction, protobuf building, then LoRA training) - more setup than Chatterbox or VoxCPM but well-documented. Our pilot run (64 steps) completed in 6 minutes 25 seconds.

What makes it work for multilingual:

- DualAR (Dual Auto-Regressive) architecture handles cross-language prosody transitions
- Strong on EN, CN, and JP - the three languages with the deepest training coverage
- ELO 1339 in TTS Arena, which tracks human preference across multilingual scenarios
- Active community and regular model updates
- LoRA fine-tuning supported - 18.4M trainable params out of 4.6B total

Trade-offs:

- Three-step data pipeline (VQ   protobuf   train) adds setup complexity compared to simpler models
- DualAR architecture is more complex to deploy than standard autoregressive models
- If you only need English, the compute overhead of multilingual capability is wasted
- 4.6B base model requires more VRAM at inference than smaller alternatives

Known pitfalls:

- **Three-step data pipeline has version sensitivity** - the VQ extraction   protobuf building   LoRA training pipeline requires matching protobuf versions. Our first protobuf build attempt failed with `ImportError: cannot import name 'builder' from 'google.protobuf.internal'`. Upgrading protobuf fixed it, but this is not obvious from the docs.
- **VQ extraction is slow** - processing 12,057 files (14.45 hours of FEMALE_01 audio) took ~22 minutes. Budget this into your pipeline setup time.

- **4.6B base model, 18.4M trainable** - the LoRA approach keeps most parameters frozen, but the base model still needs to fit in VRAM for inference. Fits on 24GB, but leave headroom.

When to use Fish Speech over Qwen3-TTS for multilingual: Qwen3-TTS supports 10 languages, but Fish Speech's deeper training data on EN/CN/JP produces more natural cross-language output for those three specifically. If your use case is primarily EN+CN+JP, Fish Speech wins. If you need broader language coverage (10 languages), Qwen3-TTS is more versatile.

Fast fine-tuning: Chatterbox

Chatterbox is a 0.5B parameter model built on Llama that has emerged as the fastest fine-tunable TTS model in the open-source ecosystem. In blind listening tests, it beats ElevenLabs at a 63.75% preference rate (Resemble AI benchmark). It was the #1 trending TTS model on HuggingFace.

First-party experience: We fine-tuned Chatterbox on IMDA NSC FEMALE_01 (512 samples, March 2026). Training completed in **2 minutes 20 seconds** on a single GPU - orders of magnitude faster than IndexTTS2 or VoxCPM fine-tuning. Two checkpoints were saved (step 384 and step 512), with a final training loss of 1.26.

What makes it work for rapid iteration:

- 0.5B parameters means fine-tuning is extremely fast and fits comfortably on 24GB
- The training pipeline uses standard HuggingFace Trainer - no custom training loop required
- Supports both TTS and voice conversion (VC) modes
- Multilingual support including English, Chinese, and more

Trade-offs:

- Smaller model (0.5B) means less capacity for complex prosody compared to 1.7B+ models
- Fine-tuning ecosystem is newer than Qwen3-TTS or IndexTTS2 - fewer community recipes
- Quality evaluation on FEMALE_01 is still pending full listening comparison against our other benchmarked models

Known pitfalls:

- **Minimum dataset size** - our pilot run with 64 samples produced `loss=0.0` (no meaningful gradients). The model needs a minimum of ~256 samples to learn. Our 512-sample run produced loss 1.26 and generated usable checkpoints.
- **Quality evaluation pending** - we have fine-tuned Chatterbox on FEMALE_01 but have not yet completed a formal listening comparison against IndexTTS2 or VoxCPM on the same corpus. The training metrics look healthy but training loss alone does not predict perceptual quality.

When to use Chatterbox: When you need to iterate on fine-tuning rapidly - testing multiple speaker profiles, dataset sizes, or training configurations. The 2-minute training cycle means you can run 30 experiments in the time it takes IndexTTS2 to complete one. Start with Chatterbox for exploration, then validate the best configuration against IndexTTS2 or VoxCPM for production deployment.

Expressive generation: Higgs Audio V2

Higgs Audio V2 is a 3B parameter model built on Llama 3.2, pre-trained on over 10 million hours of audio data. It is currently the top trending TTS model on HuggingFace (as of March 2026), positioned as an industry-leading model for expressive audio generation and multilingual voice cloning.

What makes it notable:

- 10M+ hours of pre-training data - the largest training corpus of any open-source TTS model
- Llama 3.2 3B backbone provides strong language understanding
- Expressive generation: captures whisper, vibrato, breathiness, and emotional variation
- Multilingual voice cloning from short reference clips

Trade-offs:

- 3B parameters requires more VRAM than Chatterbox (0.5B) or Kokoro (82M)
- Newer model - community recipes and fine-tuning guides are still emerging
- We have not yet run IMDA NSC benchmarks on this model - the data below is from community evaluations, not first-party

When to consider Higgs Audio V2: When expressiveness matters more than latency - audiobook narration, character voices, or content where emotional range is a quality differentiator. The 10M-hour pre-training corpus gives it a broader stylistic range than models trained on smaller datasets. If you need fine-grained control over speaking style without fine-tuning, Higgs Audio V2 is worth evaluating.

Status: Community-benchmarked only. We plan to add IMDA NSC FEMALE_01 benchmarks in a future update.

Fine-tuning: when zero-shot is not enough

Zero-shot voice cloning has improved dramatically - CosyVoice 3 and Qwen3-TTS both produce usable output from a single reference clip. But "usable" is not "production-ready" for every use case. Fine-tuning is worth the effort when:

- You need consistent output across hundreds of utterances (audiobook-length content)
- The target voice has distinctive characteristics that zero-shot does not capture (regional accent, specific speaking rhythm)
- You are building a branded voice that must sound identical every time

Reference audio requirements

This is the most common question we get. The answer is more nuanced than "more is better":

Reference audio length	What to expect
3 seconds	Minimum viable for Qwen3-TTS voice cloning. Speaker identity is captured but prosody is approximate.
10–15 seconds	Optimal range. Captures speaker identity, natural rhythm, and accent characteristics.
15+ seconds	Diminishing returns. Quality plateaus - additional audio does not meaningfully improve output.
30+ minutes (full dataset)	Required for full SFT (IndexTTS2). LoRA paths (VoxCPM, Qwen3-TTS) do not need this much.

The practical takeaway: For LoRA fine-tuning, prepare 10–15 seconds of clean reference audio per speaker. For full SFT, prepare a full dataset (IMDA NSC FEMALE_01 is ~30 minutes of labelled speech). Going beyond the optimal range adds training time without improving output quality.

Full SFT: IndexTTS2

IndexTTS2 is the model to use when you need maximum voice fidelity and are willing to invest in full SFT training. In our benchmark, it outperformed SOTA on WER, speaker similarity, and emotional fidelity. The official IndexTTS2 repo provides inference only - we wrote and open-sourced the fine-tuning pipeline: instavar/indextts2-finetuning.

Key details:

- Best checkpoint: step 14000
- Requires explicit checkpoint retention - do not rely on automatic deletion
- Crash recovery during training requires careful resume management
- Keep all checkpoints until you have completed a listening evaluation sweep

Known pitfalls:

- **Checkpoint auto-deletion** - the default retention policy deletes older checkpoints before you can evaluate them. Keep ALL checkpoints until listening eval is complete. The best checkpoint (step 14000 in our run) was not the final step (15949).
- **`transformers` version pinning** - requires exactly `transformers==4.52.1`. Older versions throw `KeyError: 'qwen3'` during model loading due to the Qwen emotion model inside IndexTTS2.
- **Crash recovery requires explicit management** - if training crashes mid-run, resume logic needs manual intervention. Log the last successful step and resume from there.

For the full IndexTTS2 fine-tuning guide, see IndexTTS2 Finetuning on IMDA NSC FEMALE_01.

F5-TTS

F5-TTS is an emerging fine-tunable model with a smaller but growing community. It is worth evaluating if you are running voice cloning experiments and want an alternative to VoxCPM or IndexTTS2. The model is capable, but the ecosystem (recipes, community checkpoints, debugging resources) is less mature than the top-tier options.

When to consider F5-TTS: When you have already tried VoxCPM or IndexTTS2 and want to compare outputs, or when the F5-TTS community has published a recipe specifically matching your language or accent.

Hardware constraints

All seven models in this guide fit on a 24GB GPU for inference. Fine-tuning constraints are tighter:

Model	Parameters	Inference (24GB GPU)	Fine-tuning (24GB GPU)	Notes
Qwen3-TTS 1.7B	1.7B		(LoRA)	SDPA backend recommended for long-text inference
CosyVoice 3	-		Rerun pending	Flow matching is compute-heavy; inference fits but is slower
IndexTTS2	-		(Full SFT)	Keep all checkpoints; save more frequently than default
VoxCPM 1.5	-		(LoRA)	44.1 kHz resampling required before training
Kokoro	82M			Fits on much less than 24GB
Fish Speech S2 Pro	4.6B (18.4M trainable)		(LoRA)	Three-step data pipeline; DualAR adds inference overhead
F5-TTS	-			Community recipes still maturing
Chatterbox	0.5B			512 samples in 2min 20s; fastest fine-tuning of any model
Higgs Audio V2	3B			Larger model; 10M+ hours pre-training

Consumer GPU reality check: An RTX 3090, RTX 3090 Ti, or RTX 4090 (all 24GB class) can run every model here for both training and inference. You do not need an A100 or H100 to get started. The constraint is recipe availability and checkpoint management, not VRAM.

For detailed VRAM profiles and training times, see Voice Cloning on a 24GB GPU - What Actually Works in 2026.

Cross-model patterns: what broke the same way everywhere

After fine-tuning nine models on the same IMDA NSC corpus, three patterns appeared consistently:

1. The best checkpoint is never the last one. VoxCPM peaked at step 4000 (not later steps). IndexTTS2 peaked at step 14000 (not the final 15949). Qwen3-TTS peaked at epoch 10 (not epoch 17). This is not a coincidence - TTS models overfit to training prosody quickly, and the last checkpoint has the lowest training loss but not the best perceptual quality. **Keep all checkpoints. Evaluate by listening, not by loss curve.**

2. Sample rate mismatches are the #1 silent failure. Qwen3-TTS requires 24 kHz. VoxCPM requires 44.1 kHz. IndexTTS2 works with both but prefers 44.1 kHz. Fish Speech defaults to its own codec sample rate. If you switch between models without checking sample rate, training may appear to work (loss decreases normally) but output quality is degraded. Always verify sample rate before every training run.

3. Scale and LR defaults are wrong for fine-tuning. Qwen3-TTS default LR (2e-5) causes noise; use 2e-6. Qwen3-TTS default LoRA scale (1.0) over-steers; use 0.3–0.35. These are not edge cases - they affect every fine-tuning run. No model's default hyperparameters are tuned for single-speaker fine-tuning on a small corpus. Always sweep before committing.

FAQ

CosyVoice 3 or Qwen3-TTS - which should I pick?

They solve different problems. CosyVoice 3 produces the best zero-shot quality for pre-produced content - audiobooks, video narration, anything where you batch-generate and review before publishing. Qwen3-TTS is the real-time model - 97ms first-packet latency, streaming-native, and the only option here if your product needs conversational response times. If latency does not matter, CosyVoice 3 for zero-shot, VoxCPM 1.5 for fine-tuned.

How much reference audio do I need for voice cloning?

3 seconds minimum (Qwen3-TTS), 10–15 seconds optimal (all models). Beyond 15 seconds, quality plateaus for LoRA fine-tuning. Full SFT (IndexTTS2) benefits from a complete dataset (~30 minutes), but the marginal gain per additional minute drops sharply after the first 15 seconds of high-quality audio.

Zero-shot or fine-tuned - when does fine-tuning become worth it?

Fine-tune when: you need consistent output across 50+ utterances, the target voice has a distinctive accent that zero-shot misses, or you are building a branded voice. Stay with zero-shot when: you are prototyping, the voice is a standard accent, or you cannot invest the 2–4 hours of training and evaluation time.

Which model retains Singaporean English accent best?

VoxCPM 1.5 and IndexTTS2 both retained the IMDA NSC FEMALE_01 accent well after fine-tuning. CosyVoice 3 zero-shot also handles non-American accents - it does not flatten to General American the way some models do. We specifically benchmark on Singaporean English because accent retention is a failure mode that most English-centric benchmarks miss entirely.

Can I run these on a consumer GPU?

Yes. All seven models fit on a 24GB GPU (RTX 3090, RTX 3090 Ti, RTX 4090) for both inference and fine-tuning. Kokoro (82M) fits on much less. You do not need data-centre hardware to get started. See our 24GB GPU guide for exact VRAM profiles.

What about proprietary models like ElevenLabs or PlayHT?

This guide covers open-source models only. Proprietary APIs (ElevenLabs, PlayHT, Azure Neural TTS) are viable but introduce vendor lock-in, per-character pricing, and data residency concerns. If you need full control over voice data, on-premise deployment, or want to avoid per-inference costs at scale, open-source is the path. The models in this guide match or exceed proprietary quality for single-speaker fine-tuned use cases.

Sources

All recommendations in this article are grounded in first-party benchmarks run on IMDA NSC FEMALE_01 using an RTX 3090 Ti (24GB). For detailed per-model results:

- Best Open-Source TTS Models for Production in 2026 - the hub comparison across all models
- IMDA NSC Voice Cloning and Finetuning Benchmark 2026 - full benchmark methodology and dataset details
- LoRA Finetuning Qwen3-TTS for Custom Voices - Qwen3-TTS fine-tuning walkthrough
- CosyVoice 2 vs CosyVoice 3 on IMDA NSC FEMALE_01 - CosyVoice generation comparison
- IndexTTS2 Finetuning on IMDA NSC FEMALE_01 - IndexTTS2 full SFT guide
- VoxCPM 1.5 LoRA Finetuning on IMDA NSC FEMALE_01 - VoxCPM fine-tuning guide
- Voice Cloning on a 24GB GPU - What Actually Works in 2026 - hardware constraints and VRAM profiles