Offline Filipino Text to Speech in QnVoice for iOS with Quantized VoxCPM2

Ona Pacholo Asis
QnSub Software Solutions
Research preprint · November 2026

Abstract

The QnVoice iOS app synthesizes Filipino speech offline with quantized VoxCPM2 and a synthetic Tagalog voice reference. We compare the configured VoxCPM2 and earlier SpeechT5 pipelines on 20 matched desktop prompts and evaluate an iPhone runtime optimization in a separate experiment. Whisper character error was 8.54% for VoxCPM2 and 15.40% for SpeechT5. Word error was 52.31% and 53.85%, respectively. On an iPhone 15 Pro, median playback readiness over five warm requests fell from 7.87 s to 2.48 s after changes to vocoding and reference encoding. The first optimized request after session release took 13.63 s. A separate six-clip Mac probe isolated the vocoder change and found matching sample counts with a maximum per-clip root mean square difference of 1.36×10^{-5} of full scale. These results quantify the behavior of the tested configurations. Naturalness, pronunciation, and broader device performance remain open questions.

Keywords Filipino, Tagalog, QnVoice, iOS, iPhone, text to speech, VoxCPM2, on-device inference

1 Introduction

QnVoice is an iOS app for on-device voice tools and translation. Its Filipino voice handles informal spellings and punctuation in typed text. When a user types “kamusta dyan, san tayo kakain,” the interface retains that text and sends “kumusta diyan, saan tayo kakain” to the synthesizer. A recording of the same phrase follows a separate waveform-conversion path to preserve the speaker’s words. This distinction affects what a speech evaluation should measure.

VoxCPM2 provides multilingual speech synthesis with a reference voice [1, 2]. Its language and waveform components create a substantial mobile inference workload. The app must allocate the model, prepare the synthetic speaker reference, generate speech, and publish the result to the audio player. A short warm request and a first request after releasing the model therefore have different timing profiles.

This study reports the iOS packaging and routing design, a matched transcription comparison of two deployed speech pipelines, and separate measurements of phone playback and Mac vocoding. The companion repository and pinned release identify the files needed to reproduce the speech path locally [9, 10].

2 Related Work and Study Scope

2.1 Filipino speech resources and mobile TTS

The Philippine Languages Database describes multilingual speech corpora for research on speech technologies in the Philippines [3]. QnVoice's historical SpeechT5 pipeline used a Filipino checkpoint whose model card identifies that dataset [8]. The VoxCPM2 configuration in this study uses a published multilingual model and an original synthetic reference designed for QnVoice [2, 10]. The desktop experiment compares these two configured systems. Their speakers, sample rates, and model families differ.

VoxCPM2 combines autoregressive latent generation with a diffusion-based local module and a waveform autoencoder [1]. Its reference encoder consumes audio at 16 kHz while the decoder can generate at 48 kHz. MobileSpeech explores a separate approach to efficient zero-shot synthesis for mobile devices [4]. Those studies establish architecture and deployment context. The present work evaluates a concrete iOS integration with a fixed Filipino preset and a single target device.

2.2 Speech, translation, and conversion

The application can translate English with a compact Marian model [5]. Translation quality is a separate property from the pronunciation of its output. OpenVoice performs voice conversion on recordings [6]. Whisper supplies a common transcription probe across synthesis variants and a regression signal for selected recordings [7]. Using a recognizer gives a repeatable score with a clear reference transcript. It also makes the score sensitive to the recognizer's own language and spelling decisions.

2.3 Research questions

Table 1 Evaluation questions and controls

Question	Protocol
How closely do the two desktop systems match intended words?	20 paired prompts with one recognizer and the same normalization for both systems.
How much does the phone runtime change playback delay?	Six matched requests in fixed order, including the first request after a session release.
Does the faster vocoder change the waveform?	Six paired Mac waveforms with the same weights, settings, and reference encoding.
Do representative app inputs reach speech output?	Catalog and tokenizer audit, targeted phone tests, and a recorded fixture.

Transcription agreement and runtime performance are separate endpoints. Fluent listener ratings would supply evidence about naturalness and intelligibility in use.

3 iOS Text to Speech System

3.1 Input routing and normalization

Typed text, phrasebook entries, and edited transcripts provide text directly. A language check routes Filipino text to the synthesizer. Free-form English uses an int8 ONNX export of the OPUS-MT English-to-Tagalog model. An exact match to certain localized presets uses their reviewed Filipino wording. The input layer standardizes curly and full-width punctuation, handles dash-like marks and peso amounts, and can expand common informal spellings for the voice. The on-screen text preserves the original spelling.

Table 2 Speech routes in QnVoice

Input	Processing path	Output
Typed Filipino	Normalize speech input and synthesize with VoxCPM2.	Generated Filipino speech.
Typed English	Translate with Marian or use a reviewed preset, then synthesize.	Generated Filipino speech and translated text.
Filipino recording	Convert original audio with OpenVoice and check wording with Whisper.	Converted or fallback audio from the recording.

If conversion changes the detected language or normalized words, the app can use the source recording and lower its pitch toward the preset when possible. It stages completed audio before making it available to playback. This avoids exposing a partly synthesized file when a request is cancelled.

3.2 VoxCPM2 assets and resource budget

The speech release contains a 1,002,374,880-byte BaseLM in Q4_K_M and a 1,077,740,192-byte acoustic file. QnVoice applies Q8_0 to eligible matrices in ResidualLM, LocEnc, and LocDiT. The waveform decoder and stop predictor retain source precision. The current preset uses a 4.48 s synthetic reference at 16 kHz mono. The app’s runtime uses a 1,024-token context and produces mono audio at 48 kHz. The weight files and reference are pinned by digest [10].

The load guard checks for at least 5.5 billion bytes of physical memory and 2.6 billion bytes of available process memory. The app releases the speech session after 20 s of inactivity. These are load-time guard thresholds. Peak resident memory and device-to-device performance require separate measurements.

4 Experimental Methods

4.1 Desktop transcription comparison

The two systems rendered the same 20 project-authored prompts. The set includes short requests and longer phrases with names, places, and a small amount of code-switching. The historical SpeechT5 graph produced 16 kHz audio. The native VoxCPM2 harness produced 48 kHz audio using the reference from the build 5 period and called the synthesizer directly. This comparison excludes app normalization and reflects each speech pipeline as configured.

Whisper Base transcribed each clip with Tagalog selected, temperature set to zero, and conditioning on previous text disabled. The scorer case-folds text, strips diacritics apart from the tilde in ñ, and forms tokens from a–z, ñ, and digits. It counts Levenshtein word edits on tokens and character edits after joining the tokens. The denominators are 130 words and 656 characters. We add a descriptive paired bootstrap that draws 20 prompts with replacement in each of 10,000 resamples and recalculates candidate-minus-baseline error rates. The interval reflects variation among these prompts.

4.2 iPhone playback timing

On an iPhone 15 Pro running iOS 27.0, each configuration ran the same six requests once in fixed order using the build 10 test project. The harness released the speech session before request one. Each request rendered a new take. The five later requests are labeled warm and include a repeat of the first prompt. Playback readiness is the time from submission until the player reports playback. A second timer records the first reading of at least 0.2 on the app’s output meter. Both endpoints occur at the start of playback.

4.3 Mac vocoder probe and app regressions

The two phone configurations differed in vocoding and reference reuse. The separate Mac probe isolates the waveform decoder implementation with the same weights, settings, reference encoding, and four-patch batches. The optimized implementation uses a matrix multiply and overlap-add for a transposed convolution whose kernel width equals twice its stride. The probe compares complete 48 kHz waveforms and records render time after loading.

App regression checks exercise the recorded phrase fixture, typed input, punctuation, localized presets, and catalog tokenization. These tests check routing and playback behavior.

5 Results

5.1 Desktop transcription agreement

Table 3 Whisper errors on 20 matched prompts

Synthesis pipeline	Character edits	CER	Word edits	WER
Historical SpeechT5	101 / 656	15.40%	70 / 130	53.85%
Quantized VoxCPM2	56 / 656	8.54%	68 / 130	52.31%

VoxCPM2 produced fewer character edits on 13 prompts, more on six, and the same number on one. Word edit counts fell on nine prompts, rose on eight, and tied on three. The stronger change in character error motivated a closer look at how the recognizer segments words.

Table 4 Descriptive differences across paired desktop prompts

Metric	Change (pp)	Paired 95% interval (pp)	Fewer / tied / more edits
CER	-6.86	-11.42 to -2.60	13 / 1 / 6
WER	-1.54	-16.15 to +12.14	9 / 3 / 8

Differences are VoxCPM2 minus SpeechT5. The paired intervals use the 20 development prompts as resampling units. The intervals describe variation within the tested prompt set.

For “Hindi, bukas na lang tayo magkita,” the VoxCPM2 clip yielded “Hindi, bukas nalang tayo mag kita.” The scorer counts four word edits but zero character edits because the differences disappear when the normalized tokens are joined. This is one reason a lower character score can coexist with a high word error rate. The origin of the segmentation difference remains uncertain.

Some prompts moved in the opposite direction. The sentence about turning before crossing a bridge had one character edit with the historical system and three with VoxCPM2. Appendix A includes examples of both directions. The comparison supports closer character agreement for this set and recognizer. The paired word interval spans zero, and perceived naturalness remains unmeasured.

5.2 Playback Timing on iPhone

Figure 1 shows the matched six-request sequence. The earlier first request took 17.37 s to playback readiness and the optimized first request took 13.63 s. The first request includes loading after the synthesis session was released. The later requests had loaded sessions and generated new audio.

Table 5 Playback timing on the iPhone 15 Pro

Endpoint	Earlier	Optimized
First request to readiness	17.37 s	13.63 s
Warm readiness median	7.87 s	2.48 s
Warm readiness range	6.47–13.20 s	2.10–3.96 s
Warm meter threshold median	8.14 s	2.73 s
Warm meter threshold range	6.78–13.40 s	2.42–4.15 s

Every matched optimized warm request reached the meter threshold within 4.15 s. The six matched output durations ranged from 1.12 to 2.72 s and were the same in both phone runs.

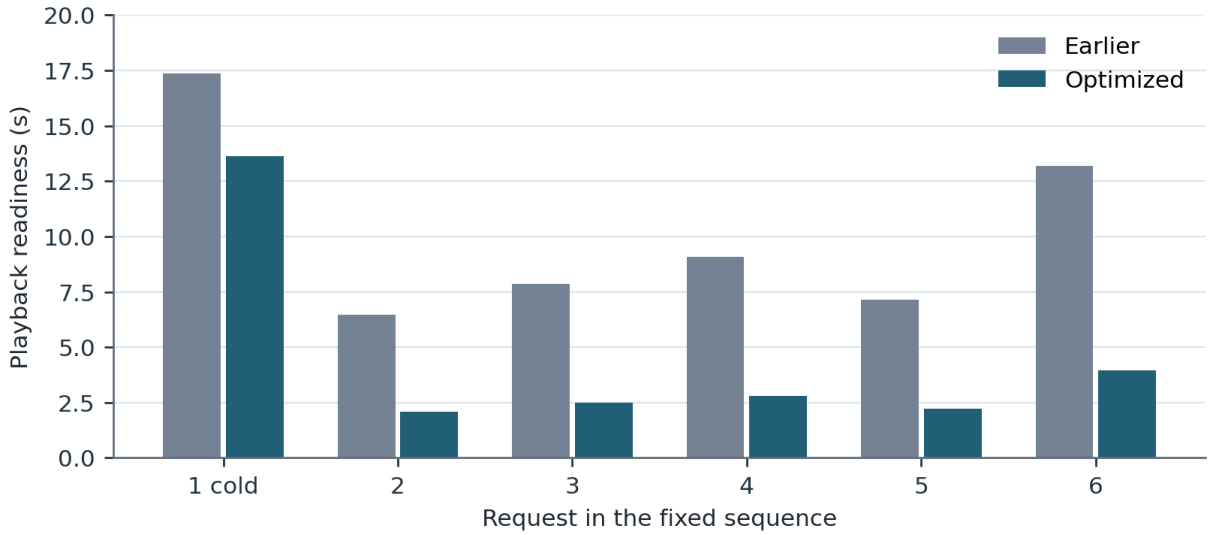


Figure 1 Time to playback readiness for six matched requests. Request 1 follows session release. Requests 2–6 are warm. The measurements are one run per configuration.

The optimized run also included two unmatched dash examples. The Filipino request reached readiness in 2.23 s. The English request, which used translation, took 7.56 s. They are reported separately because their processing paths differ.

5.3 Vocoder and App Regression Results

Vocoder waveform comparison

The isolated Mac comparison produced the same sample count in each of six paired clips. Median render time after loading fell from 4.09 s to 1.90 s. Instrumented median time attributed to vocoding fell from 2.54 s to 0.35 s. The waveform difference was computed sample by sample after verifying matching lengths.

Table 6 Waveform checks on six paired 48 kHz outputs

Check	Result
Pairs with equal sample count	6 / 6
Minimum per-clip RMS difference	1.04×10^{-5} of full scale
Maximum per-clip RMS difference	1.36×10^{-5} of full scale
Maximum absolute sample difference	2.41×10^{-4} of full scale

The sampled outputs remained close numerically. The probe assesses the specific waveform implementation and configurations used in the comparison.

Input and recording checks

The build 12 Mac audit exercised 119 English catalog entries, 94 selected symbol probes, and 969 sample–locale pairs across 51 locales. Of the sample–locale pairs, 893 matched a localized string and resolved to the expected reviewed Filipino wording. The remaining 76 already matched the source string and passed tokenization. A pasted phrase containing guillemets, a zero-width space, full-width punctuation, and a peso amount normalized to the same token sequence as its English equivalent.

Five selected build 12 phone tests passed, covering localized sample lookup, phrasebook tokenization, tokenizer parity and unknown-symbol handling, dash-only input, and typed playback with punctuation and a peso amount. The tokenizer still rejected the tested emoji and free-form Chinese and Russian text. The audit measures input handling and routing for the tested cases.

The build 7 recorded fixture retained the original 2.40 s audio and produced an output of the same duration. Whisper recovered the five expected words from that output and the corresponding typed phrase after applying the documented spelling equivalences. The phone tests used a fixture through the capture callback. This is a narrow regression check for wording preservation.

6 Discussion

6.1 Interpretation

The transcription comparison describes two synthesis pipelines as they were configured. Character agreement improved more than word agreement. The prompt examples show why the two measures can disagree. Segmenting Filipino words and handling informal spelling or mixed-language text can change the word score even when the sequence of letters matches. Fluent listeners can help determine which differences reflect pronunciation.

The phone results distinguish a warm request from the delay after the synthesis session has been released. A user who synthesizes several short requests can see a different delay from someone opening the voice for the first time. Both vocoding and reference reuse changed between the phone configurations. The isolated waveform comparison offers separate evidence for the vocoder change. A full causal split would need the same randomized phone protocol with each change enabled separately.

6.2 Limitations and follow-up evaluation

The study includes 20 development prompts, one iPhone, and one fixed-order run per phone configuration. It has limited coverage of Taglish, regional accents, long-form speech, and numeric expressions. The recorded evidence centers on one colloquial fixture. In some logs, the source of the final audio is ambiguous, which limits estimates of wording preservation by conversion alone.

A follow-up could sample natural Filipino and English–Tagalog code-switching using a documented corpus protocol [11]. Fluent listeners could transcribe clips and rate naturalness under blinded conditions with human readings as controls. Reporting accepted spelling variants, participant recruitment, and listener-level uncertainty would make the results interpretable [12, 13]. A device comparison could measure warm and first-request time, thermal state, and peak memory on several iPhone models.

6.3 Privacy and model use

The Filipino preset uses a synthetic voice reference. The application performs synthesis and the tested recognition locally. The repository publishes model assets and the synthetic reference with their stated terms. The historical SpeechT5 configuration remains a documented comparison and uses a separate checkpoint and licensing history.

7 Conclusion and Availability

The tested Filipino VoxCPM2 pipeline improved Whisper character agreement over the historical configuration. The optimized iPhone runtime reduced delay for the matched short requests and retained a substantial first-request cost. The waveform and input checks define the scope of the engineering results. Listener judgments and wider coverage across phones remain to be established.

The companion repository provides the synthetic reference, hashes, and a downloader that pins the published Hugging Face revision. The GGUF weights use Apache 2.0. The separate voice-reference terms require an app that uses the reference for synthesis to cite this paper. The project's development archive retains the historical logs and waveform probes used in the comparisons [9, 10].

GitHub companion project · Pinned Hugging Face weights · Zenodo versions

References

[1] Zhou, Y., et al. VoxCPM2 Technical Report. arXiv:2606.06928, 2026.

[2] OpenBMB and ModelBest. VoxCPM2 model card and Apache 2.0 release.

[3] Guevara, R. C. L., Cajote, R. D., Bayona, M. G. A. R., and Lucas, C. R. G. Philippine Languages Database: A multilingual speech corpora for developing systems for low-resource languages. SIGUL, 2024.

[4] Ji, S., Jiang, Z., Wang, H., Zuo, J., and Zhao, Z. MobileSpeech: A fast and high-fidelity framework for mobile zero-shot text-to-speech. ACL, 2024.

[5] Helsinki-NLP. OPUS-MT English to Tagalog model card.

[6] Qin, Z., Zhao, W., Yu, X., and Sun, X. OpenVoice: Versatile instant voice cloning. arXiv:2312.01479, revised 2024.

[7] Radford, A., et al. Robust Speech Recognition via Large-Scale Weak Supervision. ICML, 2023.

[8] internetoftim. speecht5_tts-pld-fil-ONNX model card.

[9] QnVoice. Filipino speech model companion repository and reproducibility instructions.

[10] Asis, O. P. QnVoice Filipino VoxCPM2 GGUF. Pinned release, 2026.

[11] Santa Maria, A., and Enghels, R. The Corpus of English–Tagalog Code-switching. Scientific Data, 2026.

[12] ITU-T. Subjective performance assessment of speech voice output devices. Recommendation P.85.

[13] Chiang, C.-H., Huang, W.-P., and Lee, H.-Y. Why We Should Report the Details in Subjective Evaluation of TTS More Rigorously. Interspeech, 2023.

Appendix A Evaluation Samples

The phone sequence starts with a fresh speech session. Requests 2–6 are matched warm renders. The two dash examples appear only in the optimized run. The times below are measured to playback readiness.

Table A1 Typed iPhone requests and playback readiness

Request	Typed phrase	Earlier (s)	Optimized (s)
1	Salamat po.	17.37	13.63
2	Kumusta ka?	6.47	2.10
3	Magandang umaga.	7.87	2.48
4	Paalam po.	9.09	2.81
5	Salamat po.	7.14	2.21
6	Kamusta dyan, san tayo kakain.	13.20	3.96
7	Salamat — po.	—	2.23
8	Thank you — for your help.	—	7.56

Request 5 repeats the first phrase. The harness requests a new render, so it measures the warm speech path. The English dash example also invokes translation.

Table A2 Selected desktop prompts and character edits

Test input	Reference characters	SpeechT5 edits	VoxCPM2 edits
Gusto ko ng isang baso ng tubig.	25	8	0
Hindi. Bukas na lang tayo magkita.	27	1	0
Ipa-update ang delivery address sa Biyernes.	37	9	3
Lumiliko ang bus bago tumawid sa tulay.	32	1	3

Run the Published Voice

The GitHub downloader saves both GGUF files and the reference, then checks their digests. The documented VoxCPM2 runtime renders speech locally after setup.

```
python3 fetch-model.py
python3 fetch-model.py --verify
```

Pinned runtime and CLI options · Model files and voice reference