

Concatenative Neural Speech Synthesis

Karl El Hajal^{*1,2}, Sanat Kumar Agrawal^{*1,2,3}, Enno Hermann¹, Mathew Magimai.-Doss¹

¹Idiap Research Institute, Martigny, Switzerland ²EPFL, Lausanne, Switzerland ³IIT Bombay, Mumbai, India
{karl.elhajal, enno.hermann, mathew}@idiap.ch sanat.agrawal@iitb.ac.in

Abstract—Speech synthesis has evolved from knowledge-driven methods, such as concatenative speech synthesis, to data-driven ones, such as neural speech synthesis. Both can yield commercial-grade speech synthesis systems. In this paper, we propose a target-domain training-free multi-speaker approach that combines concatenative speech synthesis with a recently proposed neural method based on self-supervised learning representations. Through experiments on multiple voices in the CMU ARCTIC corpus, we show that this combination opens new ways to exploit prior knowledge and auxiliary data to reduce target data requirements and develop interpretable and controllable neural speech synthesis systems. Subjective and objective evaluations confirm improved naturalness compared to concatenative synthesis alone, at the cost of intelligibility.

Index Terms—Text-to-Speech Synthesis, Unit Selection, Concatenative TTS, kNN, Retrieval, Voice Conversion

I. INTRODUCTION

Text-to-speech synthesis (TTS) has evolved from a knowledge-driven approach to a data-driven one. It can be argued that TTS invariably requires solving four fundamental problems:

- P1: Finding an intermediate representation that bridges the (input) text and the (output) speech signal.
- P2: How to map text to the intermediate representation?
- P3: How to map the intermediate representation to the speech signal?
- P4: How to deal with the information rate difference between the input sequence (text) and the output sequence (speech)?

Concatenative speech synthesis [1] (a) uses a linguistic and acoustic feature space as the intermediate representation (P1), with synthesis units such as diphones; (b) maps text to that feature space via text analysis, including letter-to-sound conversion and prosody prediction (P2); and (c) selects the optimal diphone sequence from a database of candidates by minimizing a weighted sum of target costs (how well a candidate unit matches the desired phonetic and prosodic context) and join costs (how smoothly adjacent units connect at their boundaries) via a Viterbi search [2], then concatenates the selected units; alternatively, a single stored instance may be concatenated and the resulting signal modified through speech analysis–synthesis methods such as PSOLA [3] (P3 and P4). While yielding commercial-grade systems, this approach

offers limited flexibility, for example in generating different voices. This issue can be addressed through voice conversion.

Statistical parametric speech synthesis (SPSS) emerged from the understanding that hidden Markov models (HMMs) are statistical sequence analysis–synthesis models [4], [5]. More precisely, SPSS (a) uses HMM state distributions over acoustic (vocoder) parameters as the intermediate representation (P1); (b) maps text to context-dependent HMM states via linguistic analysis and decision-tree-based parameter prediction (P2); (c) explicitly models the HMM state duration (P4) and source and system parameters, i.e., vocoder parameters, through HMM state emission distributions; and (d) synthesizes speech by generating a sequence of source and system parameters given the HMM state sequence and passing it through a vocoder (P3 and P4). Although SPSS provides flexibility, such as the generation of new voices through speaker adaptation, it is typically inferior in terms of speech quality when compared to concatenative synthesis, primarily due to the vocoding process. This issue was addressed by replacing the signal processing-based vocoder with a neural one [6].

Fully neural end-to-end speech synthesis approaches emerged concurrently, starting with Tacotron [7], [8], where (a) the intermediate representation is the learned encoder output, computed from grapheme embeddings (P1), (b) the input graphemes are encoded by the text encoder (P2), and (c) the encoder output is decoded into a mel spectrogram and vocoded (initially via signal processing, Griffin-Lim; later using a neural vocoder [8]) (P3 and P4). End-to-end systems such as VITS further eliminate the spectrogram intermediate step by jointly training acoustic generation and vocoding [9].

In parallel with these advances in neural speech synthesis, self-supervised learning (SSL)-based feature representations have emerged [10]–[13]. [14] demonstrated that speech can be encoded into SSL representations and synthesized back using a neural vocoder trained on those representations, an operation referred to as copy-synthesis. This understanding led to the development of k-nearest neighbor voice conversion (kNN-VC) using SSL representations [15]. More recently, kNN-TTS [16] has been proposed, where zero-shot multi-speaker speech synthesis is achieved by replacing mel spectrograms or jointly learned acoustic neural embeddings in end-to-end TTS with frozen SSL intermediate representations. This leads to a modular framework where a text-to-embedding model is first learned on single-speaker data in an end-to-end manner, and speaker characteristics are then modified akin to kNN-VC. However, kNN-TTS still requires training this text-to-embedding model on paired text-audio data, even in a single-

* Equal contribution.

This work was partially supported by the Swiss National Science Foundation through the project “Pathological Speech Synthesis (PaSS)”, grant agreement no. 219726.

speaker setting, leaving it dependent on the availability of such target-domain training data.

In this paper, we build upon kNN-TTS and kNN-VC to propose a flexible, target-domain training-free concatenative neural speech synthesis approach that combines prior knowledge-driven concatenative speech synthesis (P1, P2, and P4) with the SSL-based data-driven speech analysis–synthesis framework (P3). Through experiments on the CMU ARCTIC database [17], we explore whether classical concatenative structure, operating in SSL feature space rather than the waveform domain, can produce intelligible and natural speech without any neural acoustic model trained on text–speech pairs, and we analyze the naturalness–intelligibility trade-off that arises from concatenation in feature space.

II. BACKGROUND

In this section, we provide background on speech analysis–synthesis, voice conversion, and the kNN-TTS framework that our approach builds upon.

A. Speech analysis-synthesis

TTS system development invariably relies on the availability of a speech analysis–synthesis method. Earlier TTS approaches such as concatenative speech synthesis and SPSS were based on source-system decomposition-based speech analysis–synthesis. Linear prediction coding [18] and cepstral processing [19] are two popular source-system decomposition-based speech analysis–synthesis methods in the literature. As illustrated in Fig. 1, the aforementioned copy-synthesis can be visualized as a neural-based speech analysis–synthesis framework, analogous to a source-system decomposition-based framework. The four fundamental problems in TTS introduced earlier emerge from this analysis–synthesis perspective, where the input speech is replaced by text input.

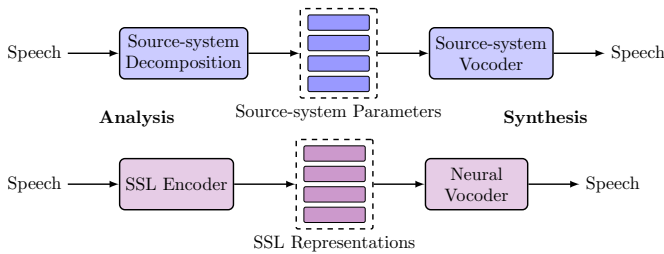


Fig. 1. Contrasting signal processing-based and neural-based speech analysis–synthesis

B. Voice conversion

Similar to TTS, voice conversion system development also relies on speech analysis–synthesis. Fig. 2 contrasts spectral representation-based voice conversion and kNN-VC. In spectral representation-based voice conversion, the mapping was traditionally achieved by collecting parallel utterances from the source and target speakers [20], [21], a requirement later alleviated by CycleGAN-VC [22]. kNN-VC [15] instead performs the mapping non-parametrically: each source SSL

frame is replaced by the average of its nearest neighbors retrieved from a target-speaker feature database, requiring no training and thereby enabling zero-shot multi-speaker voice conversion.

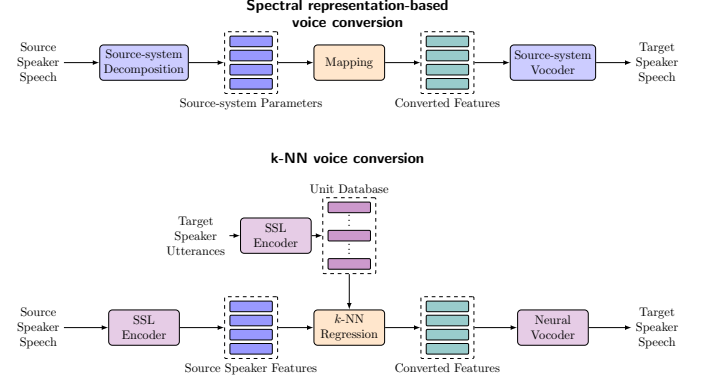


Fig. 2. Contrasting spectral representation-based voice conversion and kNN-VC

C. kNN-TTS

The recently developed kNN-TTS approach [16] brings these elements together, as illustrated in Fig. 3: given a pre-trained SSL model and a neural vocoder pre-trained on the corresponding SSL representations, a text-to-SSL module (P2 and P4) is trained on single-speaker data and combined with kNN-VC to add flexibility, such as generating multiple speaker voices in a zero-shot manner.

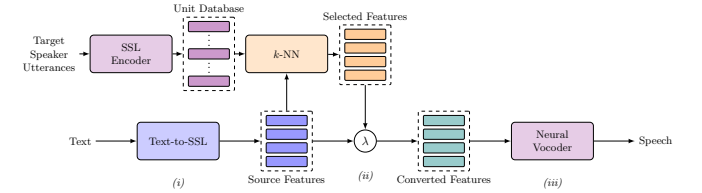


Fig. 3. kNN-TTS framework. The three stages are: (i) Text-to-SSL feature generation (realized by a supervised neural acoustic model in the original framework); (ii) feature-space voice conversion via kNN retrieval against a target-speaker database; and (iii) neural vocoding from SSL features to waveform.

III. PROPOSED APPROACH

We build upon the kNN-TTS framework to propose a target-domain training-free TTS approach that unifies concatenative speech synthesis, neural speech synthesis, and voice conversion. As illustrated in Fig. 4, our key modification replaces the learned text-to-SSL predictor (P2 and P4) with a concatenative generator: instead of predicting an SSL feature sequence from text, we construct it by selecting and concatenating SSL feature segments from a recorded unit database. The resulting sequence is passed through kNN-VC and decoded to waveform by a neural vocoder. This removes the need for any supervised text-to-SSL model training; we use *target-domain training-free* to indicate that all system components are employed as

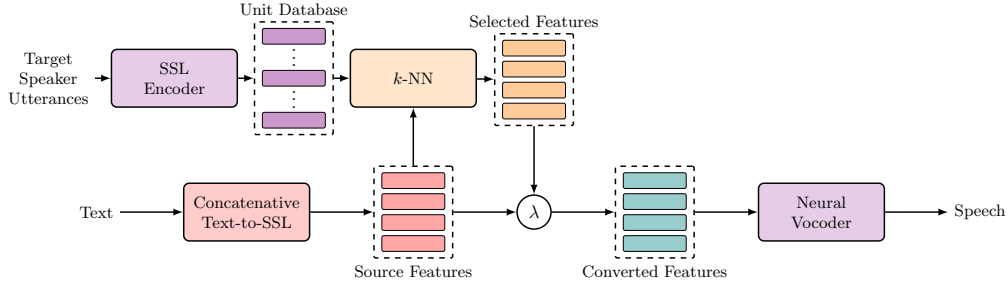


Fig. 4. Proposed concatenative speech synthesis approach. (i) Concatenative synthesis generates an SSL feature sequence directly from text; (ii) kNN-VC converts these features using a target-speaker unit database; (iii) a neural vocoder synthesizes speech from the converted features.

pretrained, with no training or fine-tuning on target-speaker data required. Multi-speaker synthesis is enabled by kNN-VC, which performs zero-shot voice conversion given only a target-speaker feature database.

Beyond the speech analysis-synthesis perspective outlined above, our approach is further motivated by the following considerations:

- 1) Although unit selection concatenative synthesis systems historically produced high-quality output [7], [23], their reliance on fixed inventories limits controllability and expressive variation compared to neural methods; nevertheless, concatenative synthesis remains in real-world use. Our approach combines the strengths of both: the interpretability of knowledge-driven concatenative synthesis and the flexibility of data-driven neural synthesis.
- 2) Speech carries multiple types of information, including message, speaker identity, and paralinguistic content. Since text primarily encodes the message, a text and a recorded *speech instance* are, from a message perspective, interchangeable. Concatenative synthesis is inherently instance-based, whereas SPSS and neural synthesis are statistical-model-based¹; combining the two lets us fall back on instances to capture variation that a statistical model may fail to generalize to, such as irregular name pronunciations.
- 3) Because our text-to-SSL stage only needs to capture the message, delegating speaker identity, rhythm, and paralinguistic variation to voice conversion [15], [24], the proposed approach offers flexibility in controlling both the resource requirements and how different types of information in the speech signal are generated. For example, we can replace unit selection’s multiple inventories with a single diphone inventory, substantially reducing the resources required for TTS development.

We next describe each of the three stages illustrated in Fig. 4 in turn: concatenative text-to-SSL, kNN-VC, and the vocoder.

A. Concatenative Text-to-SSL

Given input text, we apply a standard concatenative TTS front-end to produce a sequence of synthesis units and their

¹An analogous distinction exists in speech recognition between instance/template-based and statistical model-based approaches.

durations. SSL features are pre-extracted offline for each unit in the concatenative inventory; at inference time, we generate an SSL feature sequence by directly concatenating the corresponding pre-indexed feature segments, avoiding waveform synthesis and re-encoding.

B. kNN-VC

To enable zero-shot multi-speaker synthesis, we apply the same off-the-shelf kNN-VC component originally proposed for zero-shot voice conversion [15] to the concatenated SSL feature sequence. In our framework, kNN-VC serves two roles. First, it transfers the synthesized content to an arbitrary target voice, thus enabling zero-shot multi-speaker synthesis. Second, as a side-effect of frame-level retrieval against a coherent target-speaker database, it can smooth discontinuities introduced by concatenating feature segments from different source units.

C. Vocoder

Finally, we reconstruct waveform audio from the resulting SSL feature sequence using an off-the-shelf pre-trained neural vocoder, trained on a diverse multi-speaker dataset so that it generalizes to voices beyond those seen in the concatenative inventory.

IV. EXPERIMENTAL SETUP

We conduct two main experiments: (1) investigating how SSL representations and kNN-VC impact the quality of concatenative synthesis, and (2) evaluating the proposed method in a zero-shot multi-speaker setting. We first describe implementation details shared across both experiments, then present Experiment 1 and its six synthesis configurations, followed by Experiment 2, and finally the evaluation metrics and listening test protocol used in both.

A. Implementation

Unit Selection: We use Festival [25] as our concatenative TTS front-end for text analysis and unit selection, with two unit selection engines, each used with its default settings: CLUnits [26], which selects units from phonetically and prosodically clustered groups, and Multisyn [27], which selects diphone units by comparing stress, syllable position, part-of-speech, and predicted duration. Both determine the final unit sequence

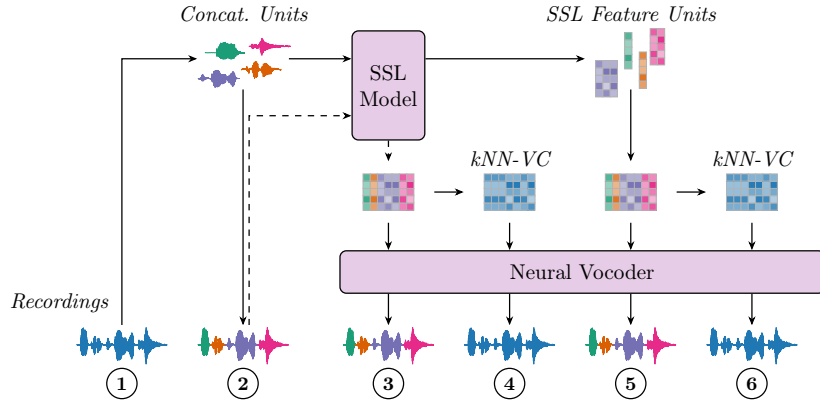


Fig. 5. The six synthesis configurations compared in Experiment 1: (1) original recordings; (2) concatenative TTS (waveform concatenation baseline); (3) concatenative TTS re-encoded through the SSL encoder-decoder; (4) concatenative TTS re-encoded and passed through kNN-VC self-retrieval; (5) NeuralConcat: direct SSL feature concatenation; (6) NeuralConcat with kNN-VC self-retrieval.

via a Viterbi search over target and join costs. We retain this unit selection pipeline unchanged across all configurations, so that the representation domain (waveform vs. SSL) is the sole variable between our system and the waveform-domain baseline; performing unit selection via costs computed directly in SSL feature space is left as future work.

SSL Encoder: We use WavLM-Large [12], extracting representations from layer 6, which has demonstrated strong capabilities for generation and kNN-based voice conversion [15]. It produces a sequence of vectors with a 25 ms window and a 20 ms hop. Speaker-specific information in the features determines the voice produced by the vocoder; when kNN-VC is applied, it replaces the source speaker identity by retrieving frames from a target-speaker database, enabling zero-shot voice conversion.

kNN-VC: kNN-VC [15] operates via frame replacement: each source frame is mapped, using cosine distance, to its $k = 4$ nearest neighbors in the target-speaker feature database, and the converted feature is obtained by uniformly averaging the retrieved vectors.

Vocoder: Waveform reconstruction is performed using a HiFi-GAN V1 [28] checkpoint from [15], trained using the pre-matched paradigm to synthesize 16 kHz waveforms directly from WavLM-Large layer 6 representations.

B. Experiment 1: SSL Representations and kNN-VC

We examine how operating in SSL representation space and applying kNN-VC affect traditional concatenative TTS in terms of naturalness and intelligibility. Fig. 5 shows the six synthesis configurations compared in this experiment, obtained by varying: (i) whether concatenation happens in the waveform or SSL feature domain, (ii) whether the waveform output is re-encoded through the SSL encoder-decoder chain, and (iii) whether kNN-VC self-retrieval is applied. The experiment uses single-speaker synthesis so that speaker identity remains fixed across conditions.

Configurations. (1) Original recordings: ground-truth utterances from the voice inventory. (2) Concatenative TTS: raw

TABLE I
CMU ARCTIC VOICES USED IN OUR EXPERIMENTS.

Voice ID	Gender	Accent	Hours
cmu_us_slt_arctic	F	American	0.94
cmu_us_rms_arctic	M	American	1.10
cmu_us_clb_arctic	F	American	1.07
cmu_us_awb_arctic	M	Scottish	1.33
cmu_us_jmk_arctic	M	Canadian	0.90
cstr_us_awb_arctic_multisyn	M	Scottish	1.33
cstr_us_jmk_arctic_multisyn	M	Canadian	0.90

waveform output of the front-end (baseline). (3) Concatenative TTS + SSL encode-decode: the waveform from (2) is passed through the frozen SSL encoder and decoded back to waveform by the neural vocoder; this isolates the contribution of the SSL encoder-decoder chain alone, independent of feature-space concatenation. (4) Concatenative TTS + kNN self-retrieval: kNN-VC self-retrieval (using the source speaker’s own feature database as target) is applied to the re-encoded features from (3) before vocoding. (5) *Direct SSL feature concatenation* (NeuralConcat): SSL features are pre-extracted offline for the concatenative unit inventory and concatenated directly at inference time, without any waveform synthesis. (6) NeuralConcat + kNN self-retrieval: kNN-VC self-retrieval is applied to the features from (5) before vocoding.

Comparing (2) vs. (4) and (5) vs. (6) isolates the contribution of kNN-VC; comparing (2), (3), and (5) isolates the effect of moving concatenation from the waveform domain to the SSL feature domain.

Voices. We evaluate seven publicly available voices from the CMU ARCTIC corpus [17], varying in synthesis method (CLUnits vs. Multisyn), gender, and accent (Table I).

C. Experiment 2: Zero-Shot Multi-Speaker Synthesis

We evaluate the NeuralConcat + kNN-VC configuration from Experiment 1 in a zero-shot multi-speaker setting. We use jmk_multisyn as the source voice, which employs the Multisyn unit selection engine designed for open-domain

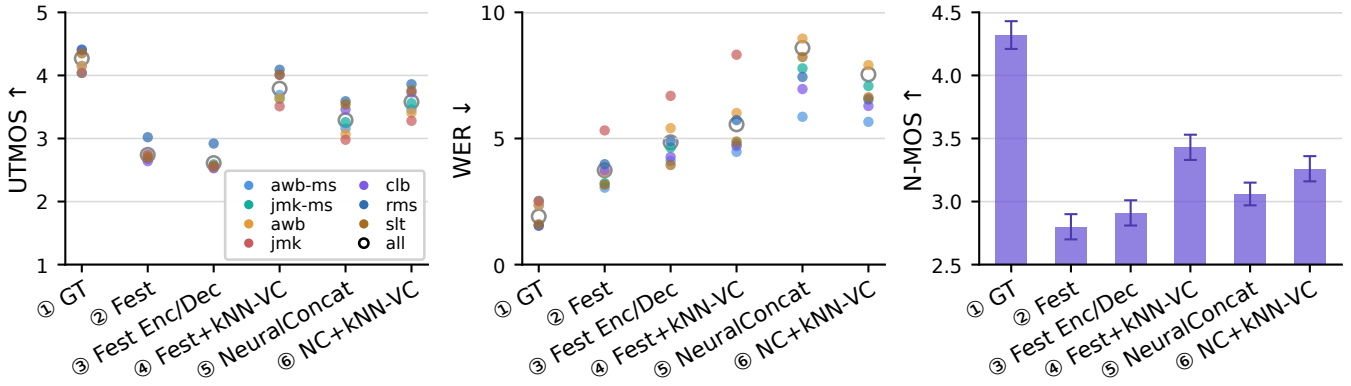


Fig. 6. Results of Experiment 1. We report objective metrics (UTMOS and WER) and subjective naturalness (N-MOS) for the six synthesis configurations numbered as in Fig. 5. GT denotes ground truth recordings, Fest denotes Festival TTS, NC denotes NeuralConcat, and “ms” voices correspond to Multisyn. Error bars on N-MOS denote 95% confidence intervals.

use. Instead of self-retrieval, kNN-VC is used to convert the concatenated features to new target speakers. We compare this approach against recent neural zero-shot TTS methods.

Target Speakers. Reference utterances are drawn from the LibriSpeech [29] *test-clean* split, comprising 20 male and 20 female speakers, each with approximately 8 minutes of speech.

Baselines. We compare against: XTTS [30], trained on 27,282 hours of transcribed multilingual speech; HierSpeech++ [31], trained on 2,796 hours of transcribed English and Korean speech across 7,299 speakers; and GlowkNN-TTS [16], a kNN-TTS variant using a GlowTTS [32] text-to-SSL model trained on the LJSpeech [33] dataset comprising 24 hours of single-speaker speech. For XTTS and HierSpeech++, we compute a global speaker embedding for each target speaker by averaging style embeddings across all available reference utterances. All baseline models are evaluated using the official checkpoints and default configurations provided by the authors [34]–[36].

D. Evaluation Metrics

For each system, we synthesize 100 English sentences per speaker using text randomly sampled from FLOres+ [37].

Objective Evaluation. Naturalness is estimated using UTMOS [38]. Intelligibility is measured using Whisper-Large-v3 [39] by computing Word Error Rate (WER). Speaker similarity (Experiment 2 only) is assessed using ECAPA2 [40] via Speaker Encoder Cosine Similarity (SECS).

Subjective Evaluation. We conduct crowd-sourced listening tests via Prolific, with 10 raters per sample. Participants are required to be native English speakers based in the United States, have $\geq 98\%$ task approval rate, and ≥ 500 approved tasks. Clear instructions are provided at the beginning and alongside each question. Attention checks and trap samples are included. Raters failing attention checks or assigning implausible scores to ground-truth samples are excluded from analysis. All ratings use a 1–5 Likert scale. For Experiment 1, we perform a naturalness MOS evaluation using the same ten randomly selected utterances across all voices and configurations. For

Experiment 2, we conduct naturalness (N-MOS) and speaker similarity (S-MOS) evaluations over 20 speakers (10 male, 10 female) with 3 utterances per speaker, yielding 60 utterances per model. For speaker similarity, raters are provided with a reference recording of the target speaker.

V. RESULTS

Fig. 6 presents the results of Experiment 1, comparing the six synthesis configurations (Fig. 5) across voices in terms of UTMOS, WER, and subjective N-MOS. Among the synthesized conditions, Festival TTS (2) yields the lowest perceived quality but the strongest intelligibility. Re-encoding through the SSL encoder–decoder alone (3) does not improve naturalness over (2), while applying kNN-VC self-retrieval (4) substantially improves perceived quality, but with further degraded WER. NeuralConcat (5), which concatenates directly in SSL feature space, improves naturalness relative to Festival TTS even without kNN-VC, but with a larger intelligibility penalty than either waveform-domain configuration. Adding kNN-VC (6) partially mitigates this trade-off, increasing UTMOS and N-MOS while reducing WER compared to NeuralConcat alone. Contrasting (2 vs. 3) with (2 vs. 5) suggests that SSL-space concatenation smooths unit boundaries and improves naturalness, at the cost of fine phonetic detail that could account for the intelligibility degradation. Across both concatenation domains, applying kNN-VC as post-processing consistently improves perceived quality.

Fig. 7 presents the results of Experiment 2. The proposed system, NeuralConcat (jmk_multisyn), uses only 0.9 hours of source speech and requires no training of a neural acoustic model on text-speech pairs, whereas GlowkNN-TTS, HierSpeech++, and XTTS are trained on 24, 2,796, and 27,282 hours of speech, respectively. In the objective evaluation, NeuralConcat trails the larger neural baselines in naturalness and intelligibility, but remains competitive in speaker similarity. The subjective results show the same pattern: NeuralConcat achieves lower perceived naturalness, but is more competitive in speaker similarity, consistent with the use of kNN-VC for

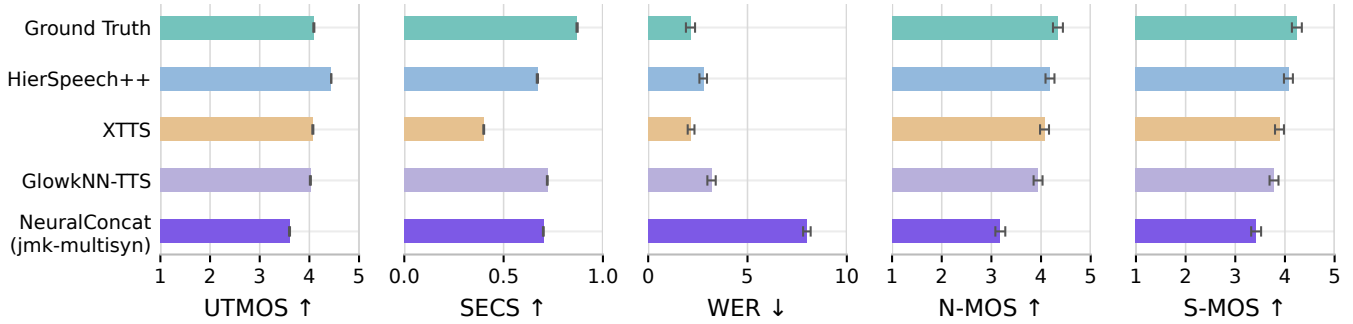


Fig. 7. Results of Experiment 2. We report objective metrics (UTMOS, SECS, WER) and subjective ratings (N-MOS and S-MOS). Error bars denote 95% confidence intervals.

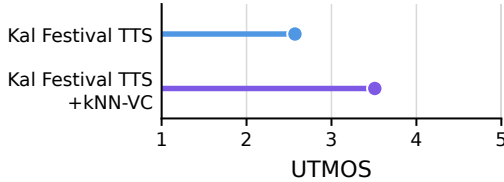


Fig. 8. Additional analysis on the `cmu_us_kal_diphone` voice. Applying kNN-VC self-retrieval to the Festival TTS output improves UTMOS over direct Festival TTS synthesis.

speaker adaptation. Audio examples for each experiment are available online².

VI. DISCUSSION AND CONCLUSIONS

This work shows that concatenative synthesis can be reformulated in an SSL feature space and combined with retrieval-based voice conversion. Across experiments, we observe that feature-space concatenation improves perceived quality over waveform-domain concatenation, while kNN-VC further improves naturalness and enables speaker adaptation. Intelligibility remains an area for improvement in the current NeuralConcat pipeline, likely requiring stronger unit selection and boundary handling in feature space. The present implementation represents a minimal concatenative pipeline; earlier commercial systems showed that concatenative synthesis can scale to production systems. Our goal in this work is mainly to demonstrate the viability of a concatenative neural synthesis framework.

A key advantage of this framework is its low data requirements: since all components are used off-the-shelf, no target-domain training is needed to adapt to a new speaker or a minimal unit inventory. In an additional analysis on the `cmu_us_kal_diphone` voice (Fig. 8), a minimal diphone inventory recorded as nonsense syllables and known to yield lower-bound synthesis quality, kNN-VC self-retrieval still improves UTMOS over direct concatenative synthesis.

The zero-shot multi-speaker experiment further illustrates this low-resource advantage. Using only 0.9 hours of source

speech from a single voice and no neural acoustic model trained on text-speech pairs, the proposed system remains competitive in speaker similarity against GlowkNN-TTS, HierSpeech++, and XTTS, trained on 24, 2,796, and 27,282 hours of speech, respectively. This is possible because target-speaker adaptation is delegated entirely to kNN-VC, which requires only a reference database rather than any additional training. NeuralConcat trails these larger neural systems in naturalness and intelligibility, consistent with the trade-off already observed in Experiment 1 and reflecting the output quality of this minimal concatenative synthesis implementation, rather than an upper bound on what concatenative synthesis or the zero-shot mechanism can achieve more broadly. Beyond speaker similarity, decoupling synthesis from voice modification in this way also enables flexible downstream control, for example through kNN-VC for speaker adaptation or RnV [24] for rhythm modification.

More broadly, the proposed framework revisits classical analysis–synthesis ideas. Earlier concatenative systems relied on signal-domain modification methods such as PSOLA, whereas our approach uses SSL analysis and neural vocoding as a feature-domain analysis–synthesis backbone, with conversion steps providing synthesis flexibility. This enables unit selection in a learned representation space while retaining explicit control over intermediate stages. Such modularity may prove attractive in constrained settings such as clinical or dysarthric speech synthesis, where collecting large amounts of speaker-specific training data for a neural acoustic model is impractical.

Future work includes evaluating the system in such low-resource and constrained settings, and extending unit selection itself to the SSL domain by computing target and join costs directly in SSL feature space, closing the loop between the representation domain and the selection criterion to potentially improve both naturalness and intelligibility. Additional directions include SSL-domain boundary smoothing, and evaluation on larger and more diverse unit inventories. Overall, these results position neural concatenation as a promising low-resource direction that can benefit from ongoing advances in speech representations and voice modification.

²<https://karlhajal.github.io/NeuralConcatTTS/>

Generative AI tools (Claude and ChatGPT) were used for language proofreading and assistance with implementing figures in TikZ.

REFERENCES

- [1] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.
- [2] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, 1996.
- [3] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in *Proc. ICASSP*, 1986.
- [4] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.
- [5] J. Dines, J. Yamagishi, and S. King, "Measuring the gap between HMM-based ASR and TTS," *IEEE Journal of Selected Topics in Signal Processing*, 2010.
- [6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016.
- [7] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomvrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017.
- [8] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018.
- [9] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. ICML*, 2021.
- [10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.
- [11] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [12] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1505–1518, 2022.
- [13] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe, T. N. Sainath, and S. Watanabe, "Self-supervised speech representation learning: A review," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1179–1210, 2022.
- [14] S. Wang, G. E. Henter, J. Gustafson, and E. Szekely, "On the use of self-supervised speech representations in spontaneous speech synthesis," in *Proc. ISCA Speech Synthesis Workshop (SSW)*, 2023.
- [15] M. Baas, B. van Niekirk, and H. Kamper, "Voice conversion with just nearest neighbors," in *Proc. Interspeech*, 2023.
- [16] K. El Hajal, A. Kulkarni, E. Hermann, and M. Magimai-Doss, "kNN retrieval for simple and effective zero-shot multi-speaker text-to-speech," in *Proc. NAACL*, 2025.
- [17] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Proc. ISCA Speech Synthesis Workshop (SSW)*, 2004.
- [18] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*. Springer-Verlag, 1976.
- [19] D. Childers, D. Skinner, and R. Kemeraït, "The cepstrum: A guide to processing," *Proceedings of the IEEE*, vol. 65, no. 10, pp. 1428–1443, 1977.
- [20] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. ICASSP*, 1988.
- [21] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.
- [22] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," in *Proc. EUSIPCO*, 2018.
- [23] X. Gonzalvo, S. Tazari, C.-a. Chan, M. Becker, A. Gutkin, and H. Silen, "Recent advances in Google real-time HMM-driven unit selection synthesizer," in *Proc. Interspeech*, 2016.
- [24] K. El Hajal, E. Hermann, S. Hovsepian, and M. Magimai-Doss, "Unsupervised rhythm and voice conversion to improve ASR on dysarthric speech," in *Proc. Interspeech*, 2025.
- [25] P. Taylor, A. W. Black, and R. Caley, "The architecture of the Festival speech synthesis system," in *Proc. ESCA TTS Workshop*, 1998.
- [26] A. W. Black and P. A. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proc. Eurospeech*, 1997.
- [27] R. A. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.
- [28] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, 2020.
- [29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.
- [30] E. Casanova, K. Davis, E. Gölge, G. Gökner, I. Gulea, L. Hart, A. Aljafari, J. Meyer, R. Morais, S. Olayemi, and J. Weber, "XTTS: A massively multilingual zero-shot text-to-speech model," in *Proc. Interspeech*, 2024.
- [31] S.-H. Lee, H.-Y. Choi, S.-B. Kim, and S.-W. Lee, "HierSpeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 10, pp. 18 422–18 436, 2025.
- [32] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Proc. NeurIPS*, 2018.
- [33] K. Ito and L. Johnson, "The LJ speech dataset," <https://keithito.com/LJ-Speech-Dataset/>, 2017.
- [34] Idiap Research Institute, "kNN-TTS," <https://github.com/idiap/knn-tts>, 2026.
- [35] —, "Coqui-AI TTS," <https://github.com/idiap/coqui-ai-TTS>, 2026.
- [36] S.-H. Lee *et al.*, "HierSpeech++," <https://github.com/sh-lee-prml/HierSpeechpp>, 2026.
- [37] M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard *et al.*, "No language left behind: Scaling human-centered machine translation," *arXiv preprint arXiv:2207.04672*, 2022.
- [38] T. Saecki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022," in *Proc. Interspeech*, 2022.
- [39] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023.
- [40] J. Thienpondt and K. Demuynck, "ECAPA2: A hybrid neural network architecture and training strategy for robust speaker embeddings," in *Proc. ASRU*, 2023.