Investigation of VITS Text-to-Speech for the Lithuanian Language

Vytautas Lėveris, Gražina Korvel

Vilnius University, Faculty of Mathematics and Informatics, Institute of Data Science and Digital Technology, Akademijos str. 4, Vilnius, Lithuania
vytautas.leveris@mif.vu.lt

Abstract. This study investigates the performance of the VITS model for Lithuanian speech synthesis under different training configurations. Experiments were conducted using datasets with phoneme-based and grapheme-based text representations, accented text, and both single-speaker and multi-speaker setups. The goal was to evaluate how linguistic pre-processing and speaker diversity influence synthesis quality. Model outputs were compared using objective measures. The results provide insights into the impact of phoneme representation and accent information on the quality of Lithuanian neural TTS systems.

Keywords: Text-to-Speech, Lithuanian language, VITS, Speech Synthesis, Phoneme-based modeling.

1 Introduction

Early approaches to text-to-speech (TTS) relied on concatenative or parametric methods. In recent years, advances in deep learning have enabled neural models to produce speech that is significantly more natural and expressive. Modern neural TTS architectures increasingly adopt end-to-end approaches that jointly learn text representation, acoustic modeling, and waveform generation. Among these approaches, the Variational Inference with Adversarial Learning for End-to-End Text-to-Speech (VITS) model has attracted significant attention due to its ability to generate high-quality speech while maintaining efficient parallel inference. VITS integrates a conditional variational autoencoder with adversarial training and normalizing flows, enabling the model to learn expressive latent representations of speech and generate natural-sounding waveforms directly from text input [1].

Despite significant progress in neural TTS systems, most research and publicly available models focus on widely used languages such as English

DOI: https://doi.org/10.15388/LMITT.2026.15

or Mandarin. In contrast, relatively few studies address languages with smaller speech resources, such as Lithuanian. Lithuanian presents several challenges for speech synthesis due to its rich inflectional morphology, complex stress patterns, and the importance of accentuation for correct pronunciation and meaning.

This study aims to investigate how different dataset configurations and text representations affect the performance of the VITS model for Lithuanian speech synthesis. Specifically, the experiments include single-speaker models trained with accented grapheme-based input (using two different speakers), and multi-speaker models trained with two alternative phoneme-based representations. The resulting models are evaluated using objective acoustic metrics to assess differences in synthesized speech quality.

The paper is organized as follows. Section 2 reviews related work on neural TTS architectures, with a focus on Lithuanian and low-resource language synthesis. Section 3 describes the datasets used in the experiments. Section 4 outlines the experimental setup and the four training configurations evaluated. Section 5 details the VITS model architecture and training parameters. Section 6 presents and discusses the objective evaluation results. Finally, Section 7 summarizes the findings and outlines directions for future work.

2 Related Work

In recent years, several neural TTS architectures have been proposed, including Tacotron-based models, FastSpeech variants, Glow-TTS, and VITS. Among these approaches, Tacotron 2 and VITS are the most commonly used architectures in recent Lithuanian speech synthesis research [3, 4]. A study conducted in 2022 compared Tacotron 2 and VITS models for Lithuanian speech synthesis and reported that VITS consistently outperformed Tacotron 2 in subjective Mean Opinion Score (MOS) evaluations for both plain and stressed text [4]. Later research focused on developing transformer-based models for Lithuanian text accenting. In that work, VITS models trained on accented text were also compared with models trained on phoneme-based representations. The results indicated that models trained on accented text achieved better synthesis quality than those trained on phoneme-based input representations [3].

Another important reason for selecting the VITS architecture is its fully end-to-end design. Unlike traditional two-stage TTS systems, which first generate intermediate representations such as mel-spectrograms and then use a separate neural vocoder to synthesize audio, VITS performs waveform generation directly within a single architecture [1]. Eliminating the separate vocoder stage simplifies the training pipeline and reduces potential error propagation between model components [2].

VITS has also been successfully applied to speech synthesis in other low-resource languages. For example, a recent study developed a Bengali speech synthesis system using a VITS-based architecture trained on a custom dataset [5]. Experimental results demonstrated that the VITS-based system produced highly natural speech despite limited training resources and the linguistic complexity of the Bengali language. Due to its ability to handle linguistic variability and generate natural speech, VITS has become increasingly popular in research involving phoneme-based and accent-aware TTS systems [6].

3 Dataset

In this paper, we use speech corpora from the Liepa-2 project [7], which was developed to support Lithuanian text-to-speech and speech-to-text applications. For the experiments, a subset of four speakers was selected, with approximately three hours of recordings per speaker. The statistics for each speaker are provided in Table 1.

Table 1. Training and test dataset statistics for each speaker.

Speaker ID	Speaker gender	Set	Number of Words	Number of Unique Words	Number of Phrases	Duration
0	female	Train	25452	13604	4934	3:33:12
		Test	705	586	100	5:31
1	male	Train	25495	13611	4933	3:05:01
		Test	710	588	100	4:56
2	female	Train	25663	13677	5038	2:58:30
		Test	710	588	100	4:40
3	male	Train	25462	13610	4934	3:17:17
		Test	706	586	100	5:14

For the multi-speaker setup, speaker datasets were merged and assigned unique speaker IDs (0-3). The multi-speaker test set contains 400 utterances (100 per speaker).

4 Experimental Setup

We conducted four experiments under a controlled setup, where the model architecture and training/evaluation hyperparameters were kept fixed across experiments. Only dataset composition and text representation settings varied between experiments.

The four configurations were selected to systematically investigate two key factors that influence Lithuanian TTS quality: speaker setup (single-speaker vs. multi-speaker) and text representation (grapheme-based vs. phoneme-based). Single-speaker models trained on accented grapheme input allow us to assess how well VITS captures speaker-specific acoustic characteristics when the input text carries explicit stress information. Two speakers (Speaker0 and Speaker3) were selected to examine whether synthesis quality varies across individuals, reflecting the sensitivity of the model to speaker-specific properties. The two multi-speaker configurations, differing in their phoneme-encoding schemes, were included to evaluate whether the choice of phoneme representation affects alignment and acoustic quality when the model must simultaneously handle multiple speakers. Together, these four configurations enable a structured comparison across both dimensions, providing insights relevant to the broader goal of building robust Lithuanian TTS systems.

The performed experiments are as follows:

1. Single-speaker VITS, Speaker0 voice, accented grapheme input.
2. Single-speaker VITS, Speaker3 voice, accented grapheme input.
3. Multi-speaker VITS, original Lithuanian phoneme representation.
4. Multi-speaker VITS, ASCII-mapped single-symbol phoneme representation.

For multi-speaker experiments, two phoneme encodings were evaluated. The original representation follows the Lithuanian phoneme system proposed by Kasparaitis, where words and syllables are explicitly segmented [8]. According to his method, there are 92 unique phonemes. Words and syllables are separated using the '+' symbol for words and the '-' for syllables, so 2 additional mappings were added. A phoneme can be

written within 1, 2, or 3 letters. For another experiment, phonemes were mapped to ASCII symbols so that each phoneme would be expressed as a single symbol. This way, we considered it would be easier for the model to map an audio fragment to a particular phoneme and potentially simplify sequence-to-acoustic alignment. For this representation, ASCII symbols from 33 to 126 were used (excluding '|' as it is a character used for dataset column separation), and also the '€' sign to cover all possible phoneme cases.

5 VITS Model

Most of the model settings follow the original VITS implementation. Table 2 summarizes the VITS parameters used for all experiments, including those for data preprocessing, optimization, and model architecture.

Table 2. Shared VITS training and model parameters.

Group	Parameter	Value
Data	sampling_rate	22050
Data	filter_length (FFT)	1024
Data	hop_length	256
Data	win_length	1024
Data	n_mel_channels	80
Data	mel_fmin	0.0
Data	mel_fmax	null (Nyquist)
Data	add_blank	true
Train	Seed	1234
Train	learning_rate	2e-4
Train	Betas	(0.8, 0.99)
Train	Eps	1e-9
Train	fp16_run	true
Train	lr_decay	0.999875
Train	segment_size	8192
Train	c_mel	45
Train	c_kl	1.0
Model	inter_channels	192
Model	hidden_channels	192
Model	filter_channels	768
Model	n_heads	2

Group	Parameter	Value
Model	n_layers	6
Model	kernel_size	3
Model	p_dropout	0.1
Model	resblock	1
Model	resblock_kernel_sizes	[3, 7, 11]
Model	resblock_dilation_sizes	[[1,3,5], [1,3,5], [1,3,5]]
Model	upsample_rates	[8, 8, 2, 2]
Model	upsample_initial_channel	512
Model	upsample_kernel_sizes	[16, 16, 4, 4]
Model	n_layers_q	3
Model	use_spectral_norm	false

Training was performed on the Vilnius University Faculty of Mathematics and Informatics (MIF) HPC supercomputer. Each training run used 4 GPU cores and 4 CPU cores for the training pipeline, data loading, and related preprocessing. For all the experiments, we trained models for approximately 4 days with a batch size of 32.

6 Results

For objective evaluation, synthesized speech was generated for utterances in the test set. Absolute duration error (ADE, in seconds), spectral distance (DTW-based L2), Mel-cepstral distortion (MCD, in dB), and pitch error (F0 RMSE, in Hz) were computed. The metrics were aggregated across utterances and are reported as the mean and standard deviation.

Table 3 reports objective metrics for single-speaker experiments, while Table 4 summarizes multi-speaker results (N denotes the number of utterances used for the test). Values are presented as mean +/- standard deviation.

Among the single-speaker models, Speaker3 consistently outperforms Speaker0 across all metrics. It achieves lower spectral distortion (4.68 vs. 5.53), lower MCD (16.74 vs. 18.61 dB), and a significantly lower F0 RMSE (7.96 vs. 22.67 Hz). This performance gap likely reflects differences in the acoustic properties of the two speakers' recordings, such as pitch range, speaking rate consistency, or recording conditions, rather than any difference in model architecture or training procedure, as both models were trained under identical configurations. The higher pitch error for Speaker0 in

particular suggests that this speaker's F0 contour may be more variable or harder to model with the available training data.

Table 3. Objective evaluation results for single speaker models.

Experiment	N	ADE (s)	DTW-based L2	MCD (dB)	F0 RMSE (Hz)
Speaker0	100	0.0753 +/- 0.0589	5.5306 +/- 0.2206	18.6107 +/- 0.8642	22.6737 +/- 6.6824
Speaker3	100	0.0697 +/- 0.0676	4.6822 +/- 0.3071	16.7373 +/- 1.3806	7.9581 +/- 3.6416

Table 4. Objective evaluation results for multi-speaker models

Experiment	N	ADE (s)	DTW-based L2	MCD (dB)	F0 RMSE (Hz)
Multi-speaker phonemes (original)	400	0.1733 +/- 0.1487	7.0692 +/- 0.9589	23.2295 +/- 2.2800	15.1006 +/- 7.5160
Multi-speaker phonemes (ASCII)	400	0.1813 +/- 0.1617	6.9651 +/- 0.9764	22.7871 +/- 2.3148	15.3243 +/- 8.1417

The multi-speaker models show notably higher error across all metrics than the single-speaker models. This is expected, as multi-speaker training requires the model to simultaneously learn acoustic representations for multiple voices, thereby increasing the complexity of the conditioning task and potentially leading to less precise modeling of each individual speaker. The greater standard deviations observed in multi-speaker results further reflect the added variability introduced by speaker diversity.

Comparing the two multi-speaker configurations, the ASCII phoneme mapping achieves slightly lower spectral and cepstral distortion (MCD 22.79 vs. 23.23 dB; DTW-based L2 6.97 vs. 7.07), while the original phoneme encoding yields slightly better duration and pitch accuracy. The ASCII representation converts each phoneme into a single symbol, which may simplify the alignment between input tokens and acoustic frames, potentially benefiting spectral reconstruction. However, the original encoding preserves more explicit linguistic structure through syllable and word boundary markers, which may help the model better capture prosodic timing and pitch patterns. The differences between the two encodings are small overall,

suggesting that neither representation has a decisive advantage in the current setup, and that further investigation with larger datasets or subjective evaluation would be needed to draw stronger conclusions.

7 Conclusions

This study evaluated four VITS configurations for Lithuanian TTS using objective acoustic metrics. The results demonstrate clear differences between single-speaker and multi-speaker configurations, as well as between individual speakers and phoneme-encoding schemes.

Among the single-speaker models, Speaker3 consistently outperformed Speaker0 across all evaluated metrics. Specifically, Speaker3 produced a lower spectral distortion (DTW-based L2: 4.68 vs. 5.53, a difference of ~15%), lower Mel-cepstral distortion (MCD: 16.74 vs. 18.61 dB, ~10% improvement), and substantially lower pitch error (F0 RMSE: 7.96 vs. 22.67 Hz, a ~65% reduction). These differences suggest that model performance is sensitive to speaker-specific acoustic properties, such as pitch range and prosodic variability, rather than to model architecture or training procedure, which were identical for both configurations.

Multi-speaker models showed higher error across all metrics than their single-speaker counterparts. For instance, the multi-speaker model with the best performance (ASCII phoneme encoding) achieved an MCD of 22.79 dB and a DTW-based L2 of 6.97. This is roughly 36% and 49% higher than the respective values of 16.74 and 4.68 for the best single-speaker model (Speaker3). This degradation is consistent with the increased complexity of multi-speaker conditioning, where the model must learn a shared acoustic space across multiple voices simultaneously.

Comparing the two multi-speaker phoneme encodings, the ASCII-mapped representation achieved slightly lower spectral and cepstral distortion (MCD: 22.79 vs. 23.23 dB; DTW-based L2: 6.97 vs. 7.07). However, the original phoneme encoding produced better duration accuracy (ADE: 0.1733 vs. 0.1813 s) and pitch error (F0 RMSE: 15.10 vs. 15.32 Hz). The differences between the two encodings are small, at less than 2% across all metrics. This suggests that neither representation has a clear advantage in the current setup, and that both should be investigated further in future experiments.

Direct comparison between single-speaker and multi-speaker models is not straightforward due to differences in dataset preparation. However,

the objective results clearly indicate that both multi-speaker configurations underperform single-speaker models in the current setup.

Future work should include controlled subjective listening tests (e.g., MOS), broader hyperparameter search, and stronger speaker-conditioning methods to improve multi-speaker performance. Additional analysis of accent quality and phonological error patterns would further clarify representation-specific behaviour. To enable a fair direct comparison between single-speaker and multi-speaker models, future experiments should also ensure that both are prepared under identical data conditions, which would allow more reliable selection of the preferred accentuation and phonetisation algorithm.

Acknowledgment

The authors are thankful for the high-performance computing resources provided by the Information Technology Open Access Center of Vilnius University.

References

[1] Kim, J., Kong, J., & Son, J. (2021). Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. International Conference on Machine Learning, 5530–5540.

[2] Radzevičius, A. (2022). Lithuanian speech synthesis using neural networks. Doctoral dissertation, Vilnius University.

[3] Mackevič, R. (2026). Transformer-based Lithuanian text stressing for speech synthesis. Doctoral dissertation, Vilnius University.

[4] Katumba, A., Kagumire, S., Nakatumba-Nabende, J., Quinn, J., & Murindanyi, S. (2025). Building text-to-speech models for low-resourced languages from crowdsourced data. Applied AI Letters, 6(2), e117.

[5] Kumar, S., Kumar, S., Sathe, K., & Pati, J. (2025). Advancing Bangla text-to-speech synthesis using a VITS-based model with a custom dataset and comprehensive evaluation. Discover Computing, 28(1), 183.

[6] Matoušek, J., Tihelka, D., & Tihelková, A. (2023, November). VITS, Tacotron or FastSpeech? Challenging some of the most popular synthesizers. In Asian Conference on Pattern Recognition, 322-335.

[7] Liepa-2 project. Available: https://raštija.lt/liepa-2/apie-projekta-liepa-2/ [Accessed: Mar. 21, 2026]

[8] Kasparaitis, P. (2005). Diphone Databases for Lithuanian Text-to-Speech Synthesis. Informatica, 16(2), 193-202.

[9] Cai, X., Xu, T., Yi, J., Huang, J., & Rajasekaran, S. (2019). DTWNet: A dynamic time warping network. Advances in neural information processing systems, 32.

[10] Abdul, Z. K., & Al-Talabani, A. K. (2022). Mel frequency cepstral coefficient and its applications: A review. IEEE Access, 10, 122136-122158.