# PITS: Variational Pitch Inference Without Fundamental Frequency for End-to-End Pitch-Controllable TTS

**Junhyeok Lee** [1 †] **Wonbin Jung** [2 †] **Hyunjae Cho** [3 4] **Jaeyeon Kim** [3 4] **Jaehwan Kim** [3 4]

## Abstract

Previous pitch-controllable text-to-speech (TTS) models rely on directly modeling fundamental frequency, leading to low variance in synthesized speech. To address this issue, we propose PITS, an end-to-end pitch-controllable TTS model that utilizes variational inference to model pitch. Based on VITS, PITS incorporates the Yingram encoder, the Yingram decoder, and adversarial training of pitch-shifted synthesis to achieve pitch-controllability. Experiments demonstrate that PITS generates high-quality speech that is indistinguishable from ground truth speech and has high pitch-controllability without quality degradation. Code, audio samples, and demo are available at https://github.com/anonymous-pits/pits.

## 1. Introduction

Text-to-speech (TTS) is a generative task that synthesizes human-like speech waveforms from a given text input. Since it is a one-to-many mapping problem, there are some challenges to model diverse duration, pitch, and *etc*. in speeches. Recent deep neural network-based TTS models (Wang et al., 2017; Shen et al., 2018; Kim et al., 2021; Donahue et al., 2021; Ren et al., 2021; Ju et al., 2022; Shirahata et al., 2022) have significantly increased their synthesis quality. Among them, VITS (Kim et al., 2021) is the most successful end-to-end (E2E) TTS, which is modeling the one-to-many nature of TTS by adopting conditional variational auto-encoder (cVAE) (Kingma & Welling, 2014) and normalizing flow (Chen et al., 2019) to provide diverse pitches and rhythms. Especially in rhythms, J. Kim *et al*. (Kim et al., 2021) propose a stochastic duration predictor and monotonic alignment search (MAS) that finds the maximum evidence lower bound (ELBO) path between the latent variables from the short-time Fourier transform (STFT) encoder and the text encoder to train a model without annotated duration labels.

On the other hand, there have been several efforts to model and control the pitch of synthesized speech. FastSpeech 2 (Ren et al., 2021) adopts a variance adaptor with a pitch predictor that predicts fundamental frequency ($f_0$) at the frame-level to provide pitch information in the speech. FastPitch (Łańcucki, 2021) improves upon FastSpeech 2 by proposing character-level pitch prediction, and FastPitchFormant (Bak et al., 2021) builds on these previous methods by incorporating pitch information to model speech according to the source-filter theory (Fant, 1971). These models are deterministic in pitch predictions and only rely on the variance in dropout. VarianceFlow (Lee et al., 2022b) applies normalizing flow for variance, but it also has low variance due to directly modeling $f_0$. Although text-to-pitch is a one-to-many mapping, these pitch-controllable TTS models tend to have low pitch variance while they are directly modeling ground truth $f_0$.

There are also some variants that have been developed to adopt pitch predictors in VITS. VISinger (Zhang et al., 2021), a singing voice synthesizer, incorporates frame prior networks and $f_0$ predictor for enhanced acoustic variation and more natural singing. However, the use of frame prior networks requires the duration predictor to be trained with annotated labels without the use of MAS. Period VITS (Shirahata et al., 2022), builds on the modifications introduced in VISinger and adds a periodicity generator. These models have limitations in that it requires external annotated duration labels and pitch contours which are expected to reduce the diversity of speech synthesis.

Since entangled pitch information, especially $f_0$, in linguistic information degrades the quality of speech synthesis, there have been several efforts to disentangle them in voice conversion (Polyak et al., 2021; Choi et al., 2021; Kim et al., 2022; Wang et al., 2021). A. Polyak *et al*. (Polyak et al., 2021) attempt to separate these two factors using vector quantization (VQ) models, HuBERT (Hsu et al., 2021) and VQ-VAE (van den Oord et al., 2017). H. Choi *et al*. (Choi et al., 2021) use wav2vec 2.0 (Baevski et al., 2020) for language information and introduced Yingram, an acoustic

[†] Work done at maum.ai. [1]Supertone Inc. [2]Korea Advanced Institute of Science and Technology (KAIST) [3]maum.ai Inc. [4]Seoul National University (SNU). Correspondence to: Junhyeok Lee <jun.hyeok@supertone.ai>.

*Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling.*, Honolulu, Hawaii, USA. PMLR 202, 2023. 

feature inspired by YIN algorithm (de Cheveigné & Kawahara, 2002), that captures pitch information including harmonics. Yingram is designed to address the limitations of $f_0$, which is not well-defined in some cases (Ardaillon & Roebel, 2020). This feature enables the training of pitch-controllable models in a fully self-supervised manner. In addition, the Yingram-based model shows better preference than the $f_0$-based model (Choi et al., 2021).

To overcome the limitations of previous pitch-controllable TTS models, which have low variance and rely on external $f_0$ extractors and labeled phoneme durations, we propose *PITS*, variational Pitch Inference for Text-to-Speech, an E2E model that incorporates variational pitch inference without $f_0$. PITS models pitch information by using cVAE and normalizing flow for the Yingram, instead of directly modeling $f_0$. Additionally, PITS utilizes two posterior encoders, the STFT encoder, and an additional Yingram encoder, unlike the original VITS, which only utilizes a single posterior encoder. We also introduce several new features to PITS to improve pitch-controllability, including the Yingram decoder and pitch-shifted waveform synthesis during training. To support these new features, we propose additional loss terms, including Yingram reconstruction loss, Yingram decoding loss, and adversarial pitch-shifted loss. Through our proposed method, PITS can generate speech with high naturalness, almost indistinguishable from recorded speech. In addition, PITS has pitch-controllability that could shift its relative pitch without quality degradation.

## 2. PITS

Based on the VITS generator (Kim et al., 2021), we add several modifications to build a pitch-controllable model. Different from VITS, PITS has Yingram encoder, Yingram decoder, Q-VAE, and synthesizing pitch-shifted waveforms during training. Figure 1 is an overall illustration of PITS during training and inference.

### 2.1. Yingram Encoder

We propose an additional posterior encoder, the Yingram encoder, to model pitch information including harmonics. Similar to previous works aimed at disentangling linguistic information and pitch information (Polyak et al., 2021; Choi et al., 2021), our proposed model PITS uses both the STFT encoder and the Yingram encoder to encode linguistic information and pitch information, respectively. The architecture of the Yingram encoder is identical to the STFT encoder without the number of channels. We set the number of input and output channels to be the same as 80 channels.

To achieve pitch-controllability, we crop the latent variables from the Yingram encoder, instead of cropping the Yingram as done in NANSY (Choi et al., 2021). While voice conversion tasks have access to a source speech to calculate the Yingram, TTS does not have access to any source speech during inference. Therefore, we crop the middle (16th - 65th channels) of the latent variables in the channel dimension, similar to cropping middle-frequency bins of Yingram in NANSY. We set the scope of the latent variables for the decoder as 50 channels, while we are utilizing 80 channels Yingram. To start, the latent variables from the Yingram encoder, $z_{\mathrm{yin}}$, are cropped to default scope (16th - 65th channels) as $z_{\mathrm{crop}}$. Additionally, scope-shift is determined by the randomly sampled integer $s$, which represents scope-shift, from a uniform distribution between $[-15, 15]$, and crops the scope-shifted (($s$+16)th - ($s$+65)th channels) latent variables $z_{\mathrm{crop}}^{\mathrm{shift}}$. The cropped latent variables from the Yingram encoder are concatenated with the latent variables from the STFT encoder and concatenated latent variables are fed to the decoder.

For the flow and MAS, we concatenate the variational latent variables from the Yingram encoder and the STFT encoder and feed them to the normalizing flow. The ELBO or Kullback–Leibler (KL) divergence calculations are also the same as in the original VITS without concatenation.

The Yingram encoder can be employed for voice conversion (VC) by extracting pitch information from the source samples. VC details are available in Appendix A.

### 2.2. Yingram Decoder

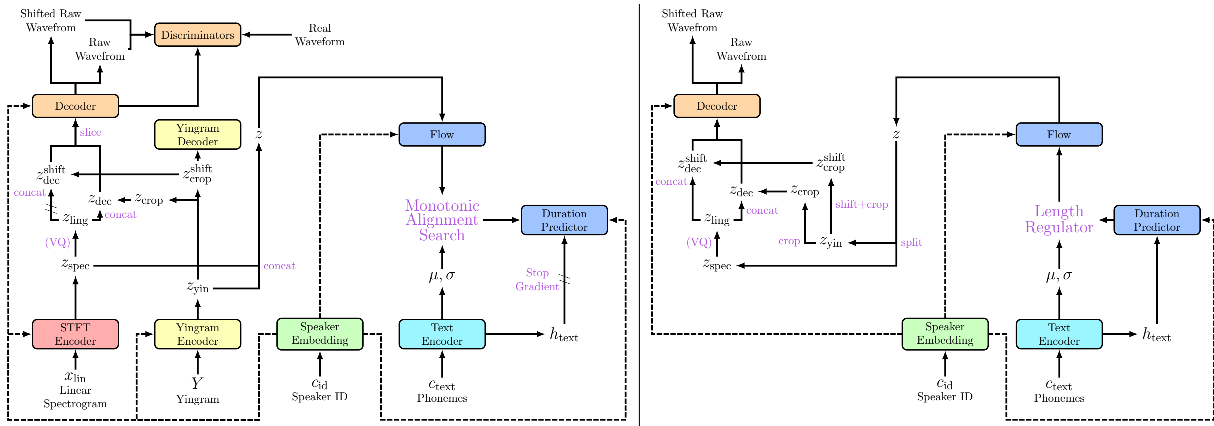
Building pitch-controllability by shifting the scope of latent variables requires the Yingram encoder to be translation equivariant in the channel direction. However, we found that outside of the default scope was not trained and remained as Gaussian. To enforce this translation equivariance, we design the Yingram decoder to reconstruct shifted scopes of Yingram from shifted latent variables. The Yingram decoder then reconstructs the Yingram of ($s$+16)th - ($s$+65)th channels from the scope-shifted latent variables $z_{\mathrm{crop}}^{\mathrm{shift}}$. $L^1$ norm is used to measure this reconstruction, the Yingram decoding loss $\mathcal{L}_{yd}$ is represented as:

$$\mathcal{L}_{yd} = \lambda_{\mathrm{yin}}\mathbb{E}\left[\left\|Y_{\mathrm{crop}}^{\mathrm{shift}} - \mathrm{Dec}_{\mathrm{yin}}\left(z_{\mathrm{crop}}^{\mathrm{shift}}\right)\right\|_1\right], \quad (1)$$

where $Y$ is the original Yingram, $Y_{\mathrm{crop}}$ is the cropped Yingram with default scope, $Y_{\mathrm{crop}}^{\mathrm{shift}}$ is the cropped Yingram with scope-shift, $\mathrm{Dec}_{\mathrm{yin}}$ is the Yingram decoder, and $\lambda_{\mathrm{yin}}$ is a multiplier for yin reconstruction loss and set to 45.

### 2.3. Q-VAE

We try Q-VAE (Sun et al., 2020; Qiang et al., 2022) for the output of the STFT encoder $z_{\mathrm{spec}}$ to disentangle the outputs of the Yingram encoder and the STFT encoder. We anticipate that quantizing the latent variables from the STFT encoder will effectively separate pitch information from




(a) PITS training procedure

(b) PITS inference procedure

Figure 1: Overall system diagram of (a) training procedure and (b) inference procedure. Vector quantization (VQ) in a diagram could be omitted in setups without Q and $z_{\text{ling}}$ represents either $z_{\text{spec}}$ or $z_{\text{VQ}}$ depending on Q ablation.

linguistic features, while pitch information is typically represented as a continuous variable. However, previous TTS models including VQ (Sun et al., 2020; Du et al., 2022; Siuzdak et al., 2022) have a limitation in that they are multiple-stage models. To remove multiple stages including VQ extractor and vocoder training, we apply Q-VAE to the VITS training scheme (Kim et al., 2021) by quantizing the output latent variable of the STFT encoder $z_{\text{spec}}$. With this implementation, the VQ extractor is trained along the waveform decoder, flow, text encoder, and duration predictor.

During applying Q-VAE (Sun et al., 2020; Qiang et al., 2022), we observed that codebook loss $\mathcal{L}_{\text{vq}}$ and commit loss $\mathcal{L}_{\text{commit}}$ diverge. It seems that our scheme is different from normal VQ-VAE since $z_{\text{spec}}$ gets multiple gradients from flow and codebooks. To prevent the diverging loss, we scale the gradient of $z^{\text{shift}}_{\text{crop}}$ from codebooks to $1/\lambda_{\text{mel}}$, where $\lambda_{\text{mel}}$ is a multiplier for mel-spectrogram reconstruction loss and set to 45. We use codebooks of size 160 and apply two codebooks as multiple codebooks in previous quantization modules (Hsu et al., 2021; Baevski et al., 2020).

### 2.4. Pitch-Shifted Waveform Synthesis

The cropped latent variables from the Yingram encoder, $z_{\text{crop}}$ and $z^{\text{shift}}_{\text{crop}}$, are concatenated with the latent variables from the STFT encoder, $z_{\text{vq}}$ or $z_{\text{spec}}$, depending on whether quantization is applied, to form $z_{\text{dec}}$ and $z^{\text{shift}}_{\text{dec}}$ for input to the decoder. For brevity, $z_{\text{vq}}$ or $z_{\text{spec}}$ is replaced to $z_{\text{ling}}$. In the case of scope-shifted variables, $z^{\text{shift}}_{\text{crop}}$ is concatenated with $z_{\text{ling}}$ after applying gradient stop to disentangle pitch information from it.

To construct a stable pitch-controllable model, PITS synthesizes pitch-shifted waveforms during training. While the default synthesized raw waveforms are trained to minimize mel-spectrogram loss to reconstruction loss, pitch-shifted waveforms do not have a corresponding ground-truth mel-spectrogram as the default synthesized signals do. To ensure that pitch-shifted synthesized speeches are indeed "pitch-shifted" from the default synthesized speeches, we propose two additional loss terms, the Yingram reconstruction loss, and the adversarial pitch-shifted loss.

First, since the ground truth mel-spectrogram of pitch-shifted speech is not available, we apply the Yingram reconstruction loss to both of normal synthesized signals and pitch-shifted synthesized signals for reconstruction loss. As YIN (de Cheveigné & Kawahara, 2002) and Yingram (Choi et al., 2021) are autocorrelation-based algorithms, they can be made differentiable. However, unlike the Yingram decoding, the ideal Yingram of the pitch-shifted speech could not be perfectly translation equivariant due to the presence of linguistic information. Thus, we apply a negative exponential to the cropped Yingrams, as values close to zero are more critical for representing harmonics. Yingram reconstruction loss $\mathcal{L}_{\text{yin}}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\text{yin}} =& \lambda_{\text{yin}}\mathbb{E}\Big[\Big\|e^{(-Y_{\text{crop}})} - e^{(-\mathcal{Y}_{\text{crop}}(G(z_{\text{dec}})))}\Big\|_1 \\ &+ \Big\|e^{(-Y^{\text{shift}}_{\text{crop}})} - e^{(-\mathcal{Y}_{\text{crop}}(G(z^{\text{shift}}_{\text{dec}})))}\Big\|_1\Big], \end{aligned} \tag{2}$$

where $\mathcal{Y}_{\text{crop}}$ is the Yingram crop function and $G$ is the waveform decoder. In addition, while we apply additional Yingram reconstruction loss, we modify the notation of mel-spectrogram reconstruction loss from $\mathcal{L}_{\text{recon}}$ to $\mathcal{L}_{\text{mel}}$.

Second, we encountered quality degradation in the pitch-shifted synthesized speech by the model that was trained using only $\mathcal{L}_{\text{yin}}$. To prevent this degradation, we also feed pitch-shifted synthesized speech to the discriminators for adversarial feedback to pitch-shifted speeches. Since waveform synthesis GANs (Kim et al., 2021; Kong et al., 2020; Kumar et al., 2019; Bak et al., 2022) get a paired input, we also pair the pitch-shifted waveforms with their original real waveforms. The time-aligned linguistic features of the pitch-shifted waveforms and real waveforms are identical, despite the difference in pitch, which makes the paired adversarial training between them reliable. Identical to waveform synthesis GANs, we apply the least-square loss (Mao et al., 2017) $\mathcal{L}_{\text{adv}}^{\text{shift}}(G)$ for the GAN and feature matching loss (Larsen et al., 2016) $\mathcal{L}_{\text{fm}}^{\text{shift}}(G)$ for the generator.

For the discriminators, we apply CoMBD and SBD from Avocodo (Bak et al., 2022) instead of MPD (Kong et al., 2020). We expect that CoMBD's collaborative structure, with multi-scale and hierarchical inputs, could enhance speech quality. To improve the pitch-shifted synthesis, we also feed hierarchical outputs of pitch-shifted synthesis to the CoMBD. To prevent periodicity artifacts, we apply PhaseAug (Lee et al., 2022a) during training. We do not apply PhaseAug to hierarchical outputs from the generator.

### 2.5. Total Loss

With our proposed method, the total loss for training PITS can be represented as follows:

$$\begin{aligned}\mathcal{L}_{\text{total}} =&\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{dur}} + \mathcal{L}_{\text{adv}}(G) + \mathcal{L}_{\text{fm}}(G)\\ &+ \mathcal{L}_{\text{yin}} + \left(\mathcal{L}_{\text{adv}}^{\text{shift}}(G) + \mathcal{L}_{\text{fm}}^{\text{shift}}(G)\right)\\ &+ (\mathcal{L}_{\text{yd}}) + (\mathcal{L}_{\text{vq}} + \mathcal{L}_{\text{commit}}),\end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{mel}}$, $\mathcal{L}_{\text{KL}}$, $\mathcal{L}_{\text{dur}}$, $\mathcal{L}_{\text{adv}}$, and $\mathcal{L}_{\text{fm}}$ are originated from loss of VITS (Kim et al., 2021). Terms enclosed in parentheses can be omitted in the model architecture search.

## 3. Experiments

### 3.1. Datasets

We conduct experiments with the VCTK dataset (Veaux et al., 2016), which includes 44 hours of recorded speech by 108 speakers with text labels. We preprocess the dataset by resampling all audio files to 22.05 kHz sampling rate data and transcribing all text into ARPABET without stress using the internal grapheme-to-phoneme model. The dataset is then split into training, validation, and test sets, with 43,225, 324, and 324 files, respectively.

### 3.2. Model Setups

To determine the optimal architecture, we perform an architecture search with four combinations. Although the Yingram encoder and $\mathcal{L}_{\text{yin}}$ are necessary for constructing a pitch-controllable model, we do not include them in the architecture search. The four variations of PITS are (A+D+Q), (A+D), (A+Q), and (D+Q), where A, D, and Q respectively denote adversarial loss for pitch-shifted signal, Yingram decoding loss, and Q-VAE. To evaluate the quality of pitch-controllability, we generate the pitch-shifted speeches by shifting the scope of normal speech, corresponding to a scope-shift of 0, with scope-shift values of [8,6,4,2,-2,-4,-6,-8], which correspond to pitch-shift values of [-4,-3,-2,-1,+1,+2,+3,+4] semitones, respectively.

We compare normal speech synthesis of PITS with previous state-of-the-art TTS models, VITS (Kim et al., 2021) and FastSpeech 2 + HiFi-GAN (FS2) (Ren et al., 2021; Kong et al., 2020). We trained VITS with ARPABET without stress by modifying the official implementation. For FS2, we adopted an unofficial modified implementation with unsupervised duration (Badlani et al., 2022) and its pre-trained model[1] (Lee, 2022). HiFi-GAN was trained by ourselves using the official implementation.

All models, except for FS2, were trained for 3,000 epochs from scratch on four V100 GPUs. Automatic mixed precision was applied during training. We set the batch size to 48 for each GPU. Each training took between 14 to 18 days.

### 3.3. Implementation Details

The flow model consists of 192 channels, the STFT encoder consists of 112 channels, and the decoder gets 162 channels of input. Aside from these modifications, the remaining details of the architecture are identical to the original VITS setup (Kim et al., 2021). Our Yingram setup is different from that used in H. Choi *et al.* (Choi et al., 2021). In our setup, 24 notes represent an octave for fine pitch modeling and note 69 is fixed at 440 Hz. We generate Yingram with 80 channels from notes ranging between -5 to 74, which corresponds to 30.8 to 508 Hz.

### 3.4. Evaluation

To compare the models, we generate speeches from the models only with the test split's text and speaker identity. We conduct a subjective evaluation using 5-scale mean opinion score (MOS). Each of MOS is performed with 324 samples and 1,620 ratings, with 95% confidence interval (CI). To quantitatively evaluate the quality of TTS models, we measure equal error rate (EER) and character error rate (CER) to assess their speaker similarity and speech intelligibility, respectively. EER is measured by official pre-trained RawNet3 (Jung et al., 2022). CER is calculated by a pre-trained Silero xlarge model (Team, 2021). To measure EER and CER, we resample all speeches to 16 kHz.

To evaluate the quantitative performance of pitch-shift, we use the shifted part of the Yingram reconstruction loss as $\mathcal{L}_{\text{yin}}^{\text{shift}} = \lambda_{\text{yin}}\mathbb{E}\left[\left\| e^{\left(-\mathcal{Y}_{\text{crop}}^{\text{shift}}(z_{\text{dec}})\right)} - e^{\left(-\mathcal{Y}_{\text{crop}}(G(z_{\text{dec}}^{\text{shift}}))\right)}\right\|_1\right]$. Unlike the loss calculation during training, where the ground truth is available, we compute the difference between the Yingram of normal synthesized speech and scope-shifted speech to obtain a $\mathcal{L}_{\text{yin}}^{\text{shift}}$.

[1]The pre-trained modified FastSpeech 2 model was trained on the VCTK and ARPABET datasets with stress.

Table 2: The evaluation results of pitch-shifted synthesized speeches. The best value in each pitch-shift is shown in bold. In cases where the standard deviation of two values overlaps, The values are paired in bold. While the (D+Q) model, which is expected to overfitting in $\mathcal{L}_{\text{yin}}^{\text{shift}}$, yields the best results in $\mathcal{L}_{\text{yin}}^{\text{shift}}$, the second-best value is also highlighted in bold.

| | **MOS ± CI** (↑) | | | | **CER (%)** (↓) | | | | **EER (%)** (↓) | | | | $\mathcal{L}_{\text{yin}}^{\text{shift}}$ (↓) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s$ | A+D+Q | **A+D** | A+Q | D+Q | A+D+Q | **A+D** | A+Q | D+Q | A+D+Q | **A+D** | A+Q | D+Q | A+D+Q | **A+D** | A+Q | **D+Q** |
| 8 | $3.39_{\pm 0.05}$ | $\mathbf{3.87}_{\pm 0.04}$ | $3.72_{\pm 0.05}$ | $2.39_{\pm 0.06}$ | 57.1 | **5.76** | 20.9 | 52.0 | 22.2 | **7.10** | 10.3 | 38.6 | 1.34 | **1.01** | 1.08 | **0.927** |
| 6 | $3.41_{\pm 0.05}$ | $\mathbf{3.94}_{\pm 0.04}$ | $3.77_{\pm 0.04}$ | $2.48_{\pm 0.06}$ | 56.4 | **5.00** | 19.7 | 51.5 | 14.0 | **3.86** | 7.25 | 35.8 | 1.34 | **0.959** | 1.04 | **0.934** |
| 4 | $3.45_{\pm 0.05}$ | $\mathbf{3.92}_{\pm 0.04}$ | $3.80_{\pm 0.04}$ | $2.68_{\pm 0.06}$ | 58.8 | **3.98** | 18.3 | 51.3 | 10.2 | **2.62** | 5.25 | 32.4 | 1.32 | **0.982** | 1.11 | **0.938** |
| 2 | $3.49_{\pm 0.05}$ | $\mathbf{4.01}_{\pm 0.04}$ | $3.85_{\pm 0.04}$ | $2.73_{\pm 0.06}$ | 55.4 | **3.64** | 17.5 | 50.8 | 5.40 | **1.85** | 3.09 | 26.5 | 1.52 | **1.15** | 1.28 | **0.927** |
| 0 | $3.70_{\pm 0.05}$ | $\mathbf{4.01}_{\pm 0.04}$ | $3.90_{\pm 0.04}$ | $3.65_{\pm 0.05}$ | 42.3 | **3.27** | 18.8 | 49.6 | 2.16 | **0.926** | 1.39 | 2.16 | – | – | – | – |
| -2 | $3.48_{\pm 0.05}$ | $\mathbf{3.94}_{\pm 0.04}$ | $3.79_{\pm 0.04}$ | $2.57_{\pm 0.06}$ | 56.2 | **3.66** | 18.1 | 52.4 | 6.17 | **2.16** | 4.32 | 22.7 | 1.53 | **1.20** | 1.29 | **0.957** |
| -4 | $3.51_{\pm 0.05}$ | $\mathbf{3.84}_{\pm 0.04}$ | $\mathbf{3.78}_{\pm 0.04}$ | $2.52_{\pm 0.06}$ | 58.0 | **4.17** | 19.2 | 53.0 | 12.7 | **3.40** | 6.63 | 32.4 | 1.34 | **1.05** | 1.16 | **1.01** |
| -6 | $3.51_{\pm 0.05}$ | $\mathbf{3.79}_{\pm 0.05}$ | $\mathbf{3.73}_{\pm 0.05}$ | $2.53_{\pm 0.06}$ | 58.1 | **4.83** | 20.7 | 54.3 | 20.2 | **5.71** | 12.0 | 37.5 | 1.36 | **1.06** | 1.12 | **1.05** |
| -8 | $3.50_{\pm 0.05}$ | $\mathbf{3.78}_{\pm 0.05}$ | $3.64_{\pm 0.05}$ | $2.50_{\pm 0.06}$ | 58.7 | **5.84** | 23.0 | 54.5 | 24.4 | **6.94** | 15.1 | 40.9 | 1.44 | **1.10** | 1.18 | **1.08** |

## 4. Results

Table 1 presents the results of normal speech synthesis. The evaluation metrics, including MOS, CER, and EER, indicate that PITS (A+D) outperforms VITS and FS2. In addition, the MOS of PITS (A+D) does not have a statistically significant difference from that of the ground truth samples.

Table 1: The evaluation results of models.

| **Model** | **MOS ± CI** (↑) | **CER (%)** (↓) | **EER (%)** (↓) |
|---|---|---|---|
| Ground Truth | 4.04 ± 0.04 | 2.63 | 1.54 |
| VITS | 3.89 ± 0.04 | 4.35 | 1.08 |
| FS2 | 3.84 ± 0.04 | 3.78 | 2.62 |
| PITS (A+D+Q) | 3.70 ± 0.05 | 42.3 | 2.16 |
| **PITS (A+D)** | **4.01 ± 0.04** | **3.27** | **0.926** |
| PITS (A+Q) | 3.90 ± 0.04 | 18.8 | 1.39 |
| PITS (D+Q) | 3.65 ± 0.05 | 49.6 | 2.16 |

Table 2 reports the evaluation results of the pitch-shifted speech. While PITS (A+D) is also the best combination in pitch-shifted synthesis without quality degradation, there are some noticeable points. The results demonstrate that the Q-VAE leads to a degradation in the quality and intelligibility of synthesized speech, as evidenced by a significant increase in the CER of the Q-VAE-included models. Moreover, combining the Yingram decoding loss with Q-VAE results in further deterioration in the quality and intelligibility of synthesized speech. The results for EER indicate that pitch-shift affects the speaker's identity. Although PITS (A+D) exhibits the smallest EER among all shifts, it increases as the pitch-shift increases. A comparison between (A+D+Q) and (D+Q) indicates that the adversarial loss for pitch-shift enhances the naturalness of pitch-shifted synthesis. However, there is no significant difference in the naturalness of synthesized speech during normal inference. In addition, when the A option is not used, we can observe that PITS (D+Q) focuses on reducing $\mathcal{L}_{\text{yin}}^{\text{shift}}$ without quality improvements. Without PITS (D+Q), PITS (A+D) also outperforms other models in $\mathcal{L}_{\text{yin}}^{\text{shift}}$.

Figure 2 depicts the pitch contours of normal and pitch-shifted synthesized speeches, extracted by PYIN (Mauch & Dixon, 2014). The figure illustrates that the pitch contours are almost parallel in the log scale, which confirms that our proposed method has high pitch-controllability. Figure 3 illustrates that PITS could generate speeches with diverse pitches from identical text and speaker identity by modeling pitch information using a variational Yingram encoder.

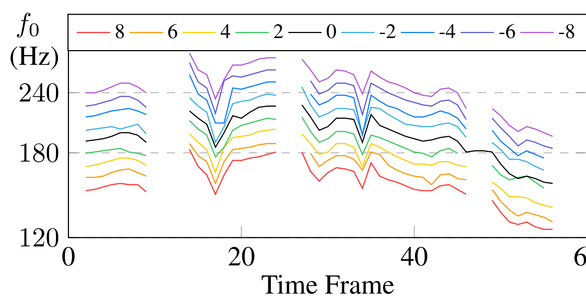


Figure 2: Pitch contours of pitch-shifted speeches.

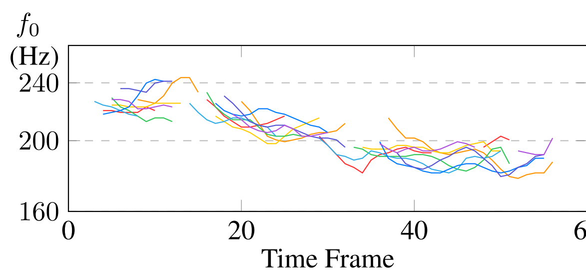


Figure 3: Pitch contours of synthesized speech with identical text, "How much variation is there?", and identity, p277.

## 5. Conclusion

In this paper, we propose PITS, an end-to-end variational pitch inference TTS. With our introduced method, the Yingram encoder, the Yingram decoding loss, and adversarial pitch-shifted training could build a pitch-controllable TTS model without directly modeling fundamental frequency, which is not well-defined in some cases. We also introduce Q-VAE with single-stage training, however, it reduces speech quality and intelligibility significantly. With our experiments, we could find the best architecture and losses, PITS (A+D), which could generate high naturalness and intelligibility as ground truth speeches. In addition, PITS could generate pitch-shifted speech by scope shift of latent variables without quality degradation.

## References


Ardaillon, L. and Roebel, A. GCI Detection from Raw Speech Using a Fully-Convolutional Network. In *IEEE ICASSP*, pp. 6739–6743, 2020.

Badlani, R., Łańcucki, A., Shih, K. J., Valle, R., Ping, W., and Catanzaro, B. One TTS Alignment to Rule Them All. In *IEEE ICASSP*, pp. 6092–6096, 2022.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *NeurIPS*, pp. 12449–12460, 2020.

Bak, T., Bae, J.-S., Bae, H., Kim, Y.-I., and Cho, H.-Y. FastPitchFormant: Source-Filter Based Decomposed Modeling for Speech Synthesis. In *Proc. Interspeech*, pp. 116–120, 2021.

Bak, T., Lee, J., Bae, H., Yang, J., Bae, J.-S., and Joo, Y.-S. Avocodo: Generative Adversarial Network for Artifact-free Vocoder. *arXiv preprint arXiv:2206.13404*, 2022.

Chen, R. T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. Residual Flows for Invertible Generative Modeling. In *NeurIPS*, pp. 9916–9926, 2019.

Choi, H.-S., Lee, J., Kim, W., Lee, J. H., Heo, H., and Lee, K. Neural Analysis and Synthesis: Reconstructing Speech from Self-Supervised Representations. In *NeurIPS*, pp. 16251–16265, 2021.

de Cheveigné, A. and Kawahara, H. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111 4:1917–30, 2002.

Donahue, J., Dieleman, S., Binkowski, M., Elsen, E., and Simonyan, K. End-to-end Adversarial Text-to-Speech. In *ICLR*, 2021.

Du, C., Guo, Y., Chen, X., and Yu, K. VQTTS: High-Fidelity Text-to-Speech Synthesis with Self-Supervised VQ Acoustic Feature. In *Proc. Interspeech*, pp. 1596–1600, 2022.

Duan, Z., Fang, H., Li, B., Sim, K. C., and Wang, Y. The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech. In *APSIPA*, pp. 1–9, 2013.

Fant, G. *Acoustic Theory of Speech Production*. De Gruyter Mouton, 1971.

Hsu, W.-N., Tsai, Y.-H. H., Bolte, B., Salakhutdinov, R., and Mohamed, A. Hubert: How much can a bad teacher benefit asr pre-training? In *IEEE ICASSP*, pp. 6533–6537, 2021.

Ju, Y., Kim, I., Yang, H., Kim, J.-H., Kim, B., Maiti, S., and Watanabe, S. TriniTTS: Pitch-controllable End-to-end TTS without External Aligner. In *Proc. Interspeech*, pp. 16–20, 2022. doi: 10.21437/Interspeech.2022-925.

Jung, J.-w., Kim, Y., Heo, H.-S., Lee, B.-J., Kwon, Y., and Chung, J. S. Pushing the limits of raw waveform speaker recognition. In *Proc. Interspeech*, pp. 2228–2232, 2022.

Kim, J., Kong, J., and Son, J. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. In *ICML*, pp. 5530–5540, 2021.

Kim, K.-W., Park, S.-W., Lee, J., and Joe, M.-C. ASSEM-VC: Realistic Voice Conversion by Assembling Modern Speech Synthesis Techniques. In *IEEE ICASSP*, pp. 6997–7001, 2022.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

Kong, J., Kim, J., and Bae, J. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. In *NeurIPS*, pp. 17022–17033, 2020.

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., de Brébisson, A., Bengio, Y., and Courville, A. C. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. In *NeurIPS*, pp. 14910–14921, 2019.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, pp. 1558–1566, 2016.

Lee, J., Han, S., Cho, H., and Jung, W. PhaseAug: A Differentiable Augmentation for Speech Synthesis to Simulate One-to-Many Mapping. *arXiv preprint arXiv:2211.04610*, 2022a.

Lee, K. Comprehensive-Transformer-TTS, 2 2022. URL https://github.com/keonlee9420/Comprehensive-Transformer-TTS.

Lee, Y., Yang, J., and Jung, K. Varianceflow: High-Quality and Controllable Text-to-Speech using Variance Information via Normalizing Flow. In *IEEE ICASSP*, pp. 7477–7481, 2022b.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. Least squares generative adversarial networks. In *IEEE ICCV*, pp. 2794–2802, 2017.

Mauch, M. and Dixon, S. PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *IEEE ICASSP*, pp. 659–663, 2014.

Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.-N., Mohamed, A., and Dupoux, E. Speech Resynthesis from Discrete Disentangled Self-Supervised Representations. In *Proc. Interspeech*, pp. 3615–3619, 2021.

Qiang, C., Yang, P., Che, H., Wang, X., and Wang, Z. Style-Label-Free: Cross-Speaker Style Transfer by Quantized VAE and Speaker-wise Normalization in Speech Synthesis. In *ISCSLP*, pp. 61–65, 2022.

Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. In *ICLR*, 2021.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., Saurous, R. A., Agiomvrgiannakis, Y., and Wu, Y. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. In *IEEE ICASSP*, pp. 4779–4783, 2018.

Shirahata, Y., Yamamoto, R., Song, E., Terashima, R., Kim, J.-M., and Tachibana, K. Period VITS: Variational Inference with Explicit Pitch Modeling for End-to-end Emotional Speech Synthesis. *ArXiv*, abs/2210.15964, 2022.

Siuzdak, H., Dura, P., van Rijn, P., and Jacoby, N. WavThruVec: Latent speech representation as intermediate features for neural speech synthesis. In *Proc. Interspeech*, pp. 833–837, 2022.

Sun, G., Zhang, Y., Weiss, R. J., Cao, Y., Zen, H., Rosenberg, A., Ramabhadran, B., and Wu, Y. Generating Diverse and Natural Text-to-Speech Samples Using a Quantized Fine-Grained VAE and Autoregressive Prosody Prior. In *IEEE ICASSP*, pp. 6699–6703, 2020.

Team, S. Silero models: pre-trained enterprise-grade stt / tts models and benchmarks (v0.4.1). https://github.com/snakers4/silero-models, 2021.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural Discrete Representation Learning. In *NeurIPS*, 2017.

Veaux, C., Yamagishi, J., MacDonald, K., et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit(Version 0.92), 2016. URL https://datashare.ed.ac.uk/handle/10283/3443.

Wang, D., Deng, L., Yeung, Y. T., Chen, X., Liu, X., and Meng, H. VQMIVC: Vector Quantization and Mutual Information-Based Unsupervised Speech Representation Disentanglement for One-Shot Voice Conversion. In *Proc. Interspeech*, pp. 1344–1348, 2021.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. Tacotron: Towards End-to-End Speech Synthesis. In *Proc. Interspeech*, pp. 4006–4010, 2017.

Zhang, Y., Cong, J., Xue, H., Xie, L., Zhu, P., and Bi, M. VISinger: Variational Inference with Adversarial Learning for End-to-End Singing Voice Synthesis. *IEEE ICASSP*, pp. 7237–7241, 2021.

Łańcucki, A. Fastpitch: Parallel text-to-speech with pitch prediction. In *IEEE ICASSP*, pp. 6588–6592, 2021.

## A. Voice Conversion

We undertake voice conversion (VC) using the pretrained PITS model, since our model disentangles linguistic and pitch information following the principles in VC studies. We perform an any-to-many voice conversion, which aims to convert the speech of an unseen source speaker to the speech of a specific seen speaker while preserving the linguistic and pitch information of the source speech. To better preserve the linguistic information of speech, we utilized both voice and text as inputs. However, different from the seen speakers, we do not have the speaker embedding of the unseen source speakers, which is required by modules such as the posterior encoders and the flow. Therefore, we optimize the speaker embedding of the source speaker $e_s$ by minimizing the KL divergence from VITS (Kim et al., 2021) as:

$$e_s = \operatorname*{argmin}_{e_s} \log q_\phi(z|S, Y, e_s) - \log p_\theta(z|c_{\text{text}}, A) \quad (4)$$

where $S$ and $Y$ represent linear spectrogram and Yingram of source speech, respectively, $c_{\text{text}}$ is a phonemes of input speech, $A$ is a optimal alignment by MAS. Furthermore, this process enables the acquisition of the optimal alignment $A$ between text and speech, which is utilized to expand the prior distribution $\mu$ and $\sigma$ from the text encoder to $\mu_A$ and $\sigma_A$. We then obtain $z_p$ from $\mathcal{N}(z_p; \mu_A, \sigma_A)$, which is subsequently fed into the inverse flow module $f^{-1}$. Empirically, one minute of training for source speaker embedding is sufficient to obtain a high-quality alignment.

To preserve the linguistic information of the text, we obtain $z_{\text{spec}}$ by splitting $z = f^{-1}(z_p, e_t)$ where $e_t$ is the target speaker embedding. Furthermore, to maintain the pitch information, we directly utilize $z_{\text{yin}}$ obtained from the source speech through the Yingram encoder with speaker condition $e_t$. Finally, pitch-guided speaker-converted speech is synthesized via the decoder from the concatenated latent $z$, which is composed of $z_{\text{spec}}$ and cropped Yingram latent $z^{\text{shift}}_{\text{crop}} = \text{shift\_crop}(z_{\text{yin}}, s)$ with scope shift $s$ where $\text{shift\_crop}(z, s) = z[:, 15 + s : 65 + s]$.

Direct utilization of $z^{\text{shift}}_{\text{crop}}$ from the unseen source speaker preserved pitch information, but degraded the audio quality of the converted speech. We first tried to optimize $z^{\text{shift}}_{\text{crop}}$ by applying an exponential regularization and $L_1$ loss between the decoded Yingram and the cropped Yingram $Y^{\text{shift}}_{\text{crop}}$ of the source speech $y$ with scope shift $s$.

$$\mathcal{L}^{\text{VC}}_{\text{yin}} = \left\| Y^{\text{shift}}_{\text{crop}} - \text{Dec}_{\text{yin}}\left(z^{\text{shift}}_{\text{crop}}\right) \right\|_1 + \lambda \left\| \exp\left(z^{\text{shift}}_{\text{crop}}\right) \right\|_1$$

While the quality of the generated audio is improved through the above modification, it still failed to achieve the quality of speech generated by PITS for TTS purpose.

The iterative synthesis illustrated in Algorithm 1 generates more natural results. In each $i$th iteration, the $i$th generated audio $y_i$ is produced by feeding the $z_{\text{crop}_i}$, cropped Yingram latent of the preceding audio output $y_{i-1}$, instead of $z^{\text{shift}}_{\text{crop}}$ when creating the concatenated latent $z_i$. However, substituting $z_{\text{spec}}$ in the $i$th iteration with the output $\text{Enc}_{\text{spec}}(\mathcal{S}(y_i), e_t)$ of the STFT encoder led to the vanishing of linguistic information. Therefore, we used the same $z_{\text{spec}}$ from the 0th iteration for every iteration to preserve linguistic information. Three iterations are sufficient for generating high-quality, pitch-conserving speech.

With pretrained PITS, it has been observed that when the pitch of the source speech falls outside the pitch range of the target speaker, the generated speech occasionally exhibits pitch deviations from the source speech by multiple octaves. To address this issue, we conducted finetuning using the NUS-48E singing voice dataset (Duan et al., 2013) with the original PITS objective. This resulted in an improvement in the expressiveness and expanded pitch range capabilities of the model, resulting in a reduction in the occurrence of octave-scale pitch differences between the source and generated speech. In addition, shifting $z_{\text{yin}}$ to match each speaker's pitch range effectively mitigates octave-scale pitch differences and prevents quality degradation.

**Algorithm 1** Voice Conversion by PITS

**Input:** source speech $y$, phoneme $c_{\text{text}}$, target speaker embedding $e_t$, scope shift $s$, number of iteration $N$
**Module:** Yingram operator $\mathcal{Y}$, spectrogram operator $\mathcal{S}$, prior encoder $\text{Enc}_{\text{p}}$, Yingram encoder $\text{Enc}_{\text{yin}}$, STFT encoder $\text{Enc}_{\text{spec}}$, flow $f$, decoder $G$
**Output:** Converted voice $y_N$

Calculate $Y \leftarrow \mathcal{Y}(y)$, $S \leftarrow \mathcal{S}(y)$
Calculate prior distribution $(\mu, \sigma) \leftarrow \text{Enc}_{\text{p}}(c_{\text{text}})$
Initialize source speaker embedding $e_s \sim \mathcal{N}(0, I)$.
**repeat**
  Calculate optimal alignment $A$ by MAS
  Update $e_s$ through SGD with KL-divergence objective
$$\log q_\phi(z|S, Y, e_s) - \log p_\theta(z|c_{\text{text}}, A)$$
**until** $e_s$ is converged
Expand $\mu$ and $\sigma$ with alignment $A$ resulting in $\mu_A$ and $\sigma_A$, respectively.
Sample $z_p \sim \mathcal{N}(z_p; \mu_A, \sigma_A)$
$z \leftarrow f^{-1}(z_p, e_t)$
$z_{\text{spec}} \leftarrow \text{split}(z)[0]$
$z^{\text{shift}}_{\text{crop}} \leftarrow \text{shift\_crop}(\text{Enc}_{\text{yin}}(Y, e_t), s)$
$z_0 \leftarrow [z_{\text{spec}}, z^{\text{shift}}_{\text{crop}}]$
$y_0 \leftarrow G(z_0, e_t)$
**for** $i = 1$ **to** $N$ **do**
  $z_{\text{crop}_i} \leftarrow \text{shift\_crop}(\text{Enc}_{\text{yin}}(\mathcal{Y}(y_{i-1}), e_t), 0)$
  $z_i \leftarrow [z_{\text{spec}}, z_{\text{crop}_i}]$
  $y_i \leftarrow G(z_i, e_t)$
**end for**
**return** $y_N$