Supplementary Material of Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech

A. Monotonic Alignment Search

We present pseudocode for MAS in Figure 4. Although we search the alignment which maximizes the ELBO not the exact log-likelihood of data, we can use the MAS implementation of Glow-TTS as described in Section 2.2.1.

```
def monotonic_alignment_search(value):
    """Returns the most likely alignment for the given log-likelihood matrix.
    Args:
        value: the log-likelihood matrix. Its (i, j)-th entry contains
        the log-likelihood of the j-th latent variable
        for the given i-th prior mean and variance:
        .. math::
            value_{i,j} = log N(f(z)_{j}; \mu_{i}, \sigma_{i})
        (dtype=float, shape=[text_length, latent_variable_length])
    Returns:
        path: the most likely alignment.
        (dtype=float, shape=[text_length, latent_variable_length])
    """
    t_x, t_y = value.shape # [text_length, letent_variable_length]
    path = zeros([t_x, t_y])

    # A cache to store the log-likelihood for the most likely alignment so far.
    Q = -INFINITY * ones([t_x, t_y])

    for y in range(t_y):
        for x in range(max(0, t_x + y - t_y), min(t_x, y + 1)):
            if y == 0: # Base case. If y is 0, the possible x value is only 0.
                Q[x, 0] = value[x, 0]
            else:
                if x == 0:
                    v_prev = -INFINITY
                else:
                    v_prev = Q[x-1, y-1]
                v_cur = Q[x, y-1]
                Q[x, y] = value[x, y] + max(v_prev, v_cur)

    # Backtrack from last observation.
    index = t_x - 1
    for y in range(t_y - 1, -1, -1):
        path[index, y] = 1
        if index != 0 and (index == y or Q[index, y-1] < Q[index-1, y-1]):
            index = index - 1

    return path
```

Figure 4. Pseudocode for Monotonic Alignment Search.

B. Model Configurations

In this section, we mainly describe the newly added parts of VITS as we followed configurations of Glow-TTS and HiFi-GAN for several parts of our model: we use the same transformer encoder and WaveNet residual blocks as those of Glow-TTS; our decoder and the multi-period discriminator is the same as the generator and multi-period discriminator of

HiFi-GAN, respectively, except that we use different input dimension for the decoder and append a sub-discriminator.

B.1. Prior Encoder and Posterior Encoder

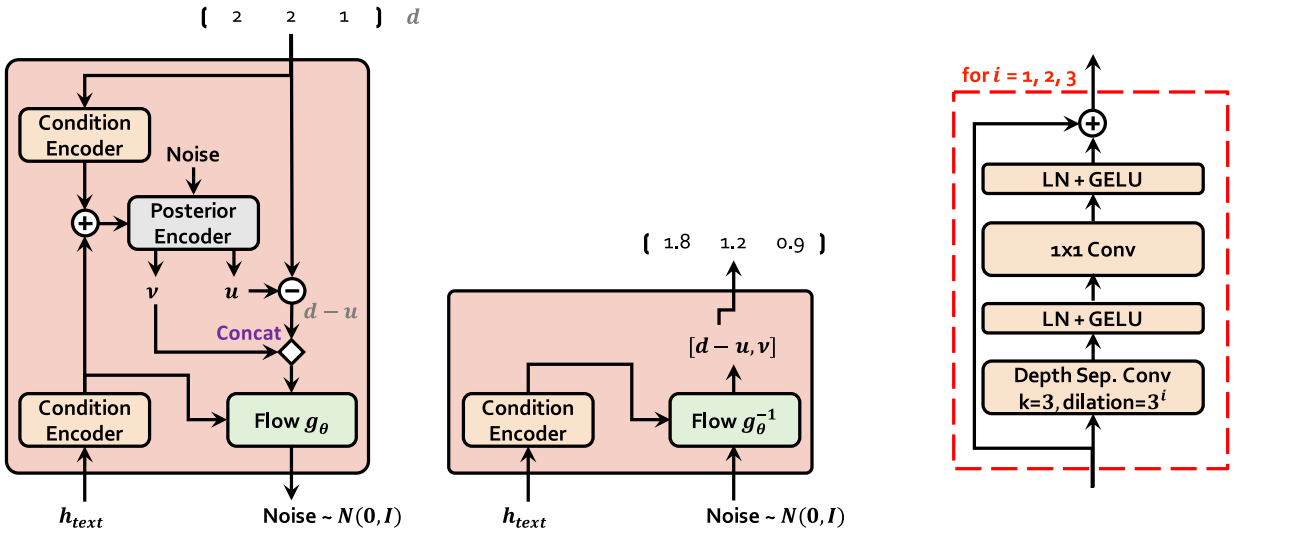
The normalizing flow in the prior encoder is a stack of four affine coupling layers, each coupling layer consisting of four WaveNet residual blocks. As we restrict the affine coupling layers to be volume-preserving transformations, the coupling layers do not produce scale parameters.

The posterior encoder, consisting of 16 WaveNet residual blocks, takes linear-scale log magnitude spectrograms and produce latent variables with 192 channels.

B.2. Decoder and Discriminator

The input of our decoder is latent variables generated from the prior or posterior encoders, so the input channel size of the decoder is 192. For the last convolutional layer of the decoder, we remove a bias parameter, as it causes unstable gradient scales during mixed precision training.

For the discriminator, HiFi-GAN uses the multi-period discriminator containing five sub-discriminators with periods $[2, 3, 5, 7, 11]$ and the multi-scale discriminator containing three sub-discriminators. To improve training efficiency, we leave only the first sub-discriminator of the multi-scale discriminator that operates on raw waveforms and discard two sub-discriminators operating on average-pooled waveforms. The resultant discriminator can be seen as the multi-period discriminator with periods $[1, 2, 3, 5, 7, 11]$.



(a) Training procedure

(b) Inference procedure

(c) Dilated and depth-wise separable convolutional residual block

Figure 5. Block diagram depicting (a) training procedure and (b) inference procedure of the stochastic duration predictor. The main building block of the stochastic duration predictor is (c) the dilated and depth-wise separable convolutional residual block.

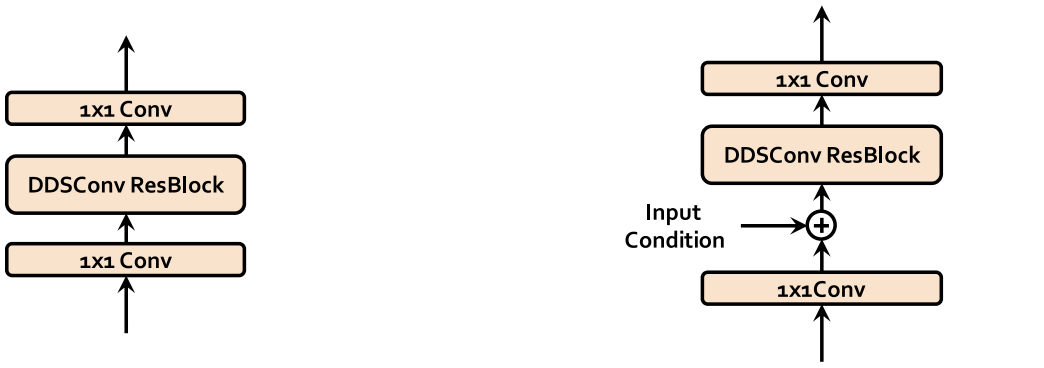
B.3. Stochastic Duration Predictor

Figures 5a and 5b show the training and inference procedures of the stochastic duration predictor, respectively. The main building block of the stochastic duration predictor is the dilated and depth-wise separable convolutional (DDSConv) residual block as in Figure 5c. Each convolutional layer in DDSConv blocks is followed by a layer normalization layer and GELU activation function. We choose to use dilated and depth-wise separable convolutional layers for improving parameter efficiency while maintaining large receptive field size.

The posterior encoder and normalizing flow module in the duration predictor are flow-based neural networks and have the

similar architecture. The difference is that the posterior encoder transforms a Gaussian noise sequence into two random variables ν and u to express the approximate posterior distribution $q_\phi(u, \nu|d, c_{text})$, and the normalizing flow module transforms $d - u$ and ν into a Gaussian noise sequence to express the log-likelihood of the augmented and dequantized data $\log p_\theta(d - u, \nu|c_{text})$ as described in Section 2.2.2.

All input conditions are processed through condition encoders, each consisting of two 1x1 convolutional layers and a DDSConv residual block. The posterior encoder and normalizing flow module have four coupling layers of neural spline flows. Each coupling layer first processes input and input conditions through a DDSConv block and produces 29-channel parameters that are used to construct 10 rational-quadratic functions. We set the hidden dimension of all coupling layers and condition encoders to 192. Figure 6a and 6b show the architecture of a condition encoder and a coupling layer used in the stochastic duration predictor.



(a) Condition encoder in the stochastic duration predictor

(b) Coupling layer in the stochastic duration predictor

Figure 6. The architecture of (a) condition encoder and (b) coupling layer used in the stochastic duration predictor.

C. Side-by-Side Evaluation

We conducted 7-point Comparative Mean Opinion Score (CMOS) evaluation between VITS and the ground truth through 500 ratings on 50 items. Our model achieved -0.106 and -0.270 CMOS on the LJ Speech and the VCTK datasets, respectively, as in Table 5. It indicates that even though our model outperforms the best publicly available TTS system, Glow-TTS and HiFi-GAN, and achieves a comparable score to ground truth in MOS evaluation, there remains a small preference of raters towards the ground truth over our model.

Table 5. Evaluated CMOS of VITS compared to the ground truth.

Dataset	CMOS
LJ Speech	-0.106
VCTK	-0.262

D. Voice Conversion

In the multi-speaker setting, we do not provide speaker identities into the text encoder, which makes the latent variables estimated from the text encoder learn speaker-independent representations. Using the speaker-independent representations, we can transform an audio recording of one speaker into a voice of another speaker. For a given speaker identity s and an utterance of the speaker, we can attain a linear spectrogram x_{lin} from the corresponding utterance audio. We can transform x_{lin} into a speaker-independent representation e through the posterior encoder and the normalizing flow in the prior encoder:

$$z \sim q_\phi(z|x_{lin}, s) \tag{12}$$

$$e = f_\theta(z|s) \tag{13}$$

Then, we can synthesize a voice $\hat{y}$ of a target speaker identity $\hat{s}$ from the representation e through the inverse transformation of the normalizing flow f_θ^{-1} and decoder G:

$$\hat{y} = G(f_\theta^{-1}(e|\hat{s})|\hat{s}) \tag{14}$$

Learning speaker-independent representations and using it for voice conversion can be seen as an extension of the voice conversion method proposed in Glow-TTS. Our voice conversion method provides raw waveforms rather than mel-spectrograms as in Glow-TTS. The voice conversion results are presented in Figure 7. It shows a similar trend of pitch tracks with different pitch levels.

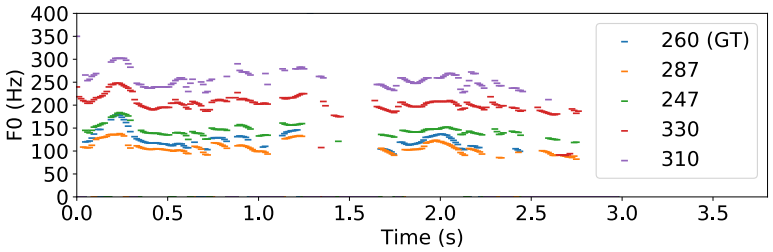


Figure 7. Pitch tracks of a ground truth sample and the corresponding voice conversion samples with different speaker identities.