Interspeech 2022
18-22 September 2022, Incheon, Korea

# EdiTTS: Score-based Editing for Controllable Text-to-Speech

*Jaesung Tae*[1*], *Hyeongju Kim*[2*], *Taesu Kim*[3]

[1]Yale University
[2]Supertone, Inc.
[3]Neosapience, Inc.

jake.tae@yale.edu, hyeongju@supertone.ai, taesu@neosapience.com

## Abstract

We present EdiTTS, an off-the-shelf speech editing methodology based on score-based generative modeling for text-to-speech synthesis. EdiTTS allows for targeted, granular editing of audio, both in terms of content and pitch, without the need for any additional training, task-specific optimization, or architectural modifications to the score-based model backbone. Specifically, we apply coarse yet deliberate perturbations in the Gaussian prior space to induce desired behavior from the diffusion model while applying masks and softening kernels to ensure that iterative edits are applied only to the target region. Through listening tests and speech-to-text back transcription, we show that EdiTTS outperforms existing baselines and produces robust samples that satisfy user-imposed requirements.[1]
**Index Terms**: speech editing, pitch control, content replacement, score-based modeling

## 1. Introduction

Score-based generative models [1] and denoising diffusion probabilistic models [2] have proven remarkably successful in various generative tasks, achieving state-of-the-art performance in image generation [3, 4] and speech synthesis [5, 6, 7, 8, 9]. This emerging class of generative models follows an iterative training scheme in which a complex data distribution is gradually corrupted with Gaussian noise, and the model learns to estimate gradient fields to reverse noisy priors back into their original distribution. Recent breakthroughs in image editing [10, 11] also utilize score-based generative models to synthesize photo-realistic images from user edits, such as rough sketches or scribbles. In this setup, the pretrained model is effectively prompted to interpolate realistic details from coarse user input, producing an edited image.

Bearing strong resemblance to image editing, speech editing is a conditional generation task in which a model modifies some selected audio segments according to user instructions. Despite advancements in neural text-to-speech (TTS), fine-grained generation remains a non-trivial task, and it is often difficult to obtain a sample that exactly satisfies the user's needs, especially when it comes to prosody and tone. This limitation highlights the need for a robust speech editing framework that can provide greater control over the final output. Previous works have used unit selection [12] or context-aware prosody correction [13], but their reliance on traditional algorithms such as concatenation or TD-PSLA [14] complicates the synthesis pipeline while compromising output quality. Alternative deep learning-based methods typically require training an additional editing model, such as a bidirectional fusion module [15] on top of an existing TTS pipeline, incurring time and computation costs.

We propose EdiTTS, an off-the-shelf score-based speech editing framework that leverages a pretrained score-based backbone to enable granular editing of pitch and audio content. Unlike score-based image editing methods, which directly modify the input with user edits, EdiTTS induces editing behavior by applying coarse perturbations to diffused Gaussian priors. The score estimator then denoises the perturbed latents along the trajectory that best aligns with the data distribution to generate an edited sample.

Our contribution can be summarized as follows: (1) We present a new method of editing speech, EdiTTS, which exploits the score-based TTS model to refine a coarsely modified mel-spectrogram prior. This conception allows for a unified approach to speech editing, where a single model is used to perform various editing tasks. (2) We show that EdiTTS is able to produce more natural, human-sounding speech without any conditional inputs such as pitch, thanks to the robust denoising process of the score-based TTS model. (3) EdiTTS is easy to implement since it leverages the existing TTS pipeline as-is without requiring any additional auxiliary module, fine-tuning, or changes to training objectives. In this work, we demonstrate our method using Grad-TTS [8] as the model backbone, focusing on pitch shift and content replacement as popular applications of speech editing.

## 2. Background

### 2.1. Score-based Generative Modeling

In score-based generative models, the original data distribution is corrupted with Gaussian noise according to an Itô SDE

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt + \mathbf{g}(\mathbf{x}, t)\, d\mathbf{w}, \tag{1}$$

where $\mathbf{w}$ is the standard Wiener process, $t$ is a value sampled from a finite time interval $[0, T]$, and $\mathbf{f}(\cdot, t)$ and $\mathbf{g}(\cdot, t)$ are drift and diffusion coefficients of $\mathbf{x}$, respectively.

A property of diffusion is that its reverse is also a diffusion process. Let $\bar{w}$ denote the reverse-time Wiener process. The reverse-time SDE [16] is given by

$$\begin{aligned} d\mathbf{x} = &[\mathbf{f}(\mathbf{x}, t) - \mathbf{g}(\mathbf{x}, t)\mathbf{g}(\mathbf{x}, t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\, dt \\ &+ \mathbf{g}(\mathbf{x}, t)\, d\bar{\mathbf{w}}. \end{aligned} \tag{2}$$

Furthermore, Kolmogorov's forward equation of Eq. (1) equals that of a more simplified probability flow ordinary differential equation [4]. The implication of this observation is that both equations ultimately describe an identical statistical process, albeit in different form.

[*]Equal contribution. Work performed at Neosapience.
[1]Code and audio samples at https://editts.github.io.

Copyright © 2022 ISCA

10.21437/Interspeech.2022-6

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt - \frac{1}{2}\nabla \cdot [\mathbf{g}(\mathbf{x}, t)\mathbf{g}(\mathbf{x}, t)^\top]\, dt - \frac{1}{2}\mathbf{g}(\mathbf{x}, t)\mathbf{g}(\mathbf{x}, t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\, dt. \tag{3}$$

Therefore, given a time-dependent model $\mathbf{s}_\theta(\mathbf{x}, t)$ that can approximate the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, we can sample $\mathbf{x}_T \sim p_T$ and compute reverse steps according to Eq. (3) to obtain $\mathbf{x}_0 \sim p_0$ from the original data distribution.

### 2.2. Grad-TTS

In this section, we expound Grad-TTS, the score-based TTS model we use to demonstrate EdiTTS. A more in-depth discussion of its components, such as monotonic alignment search [17], can be found in the original paper [8].

The forward SDE of Grad-TTS is defined as

$$d\mathbf{x} = \frac{1}{2}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{x})\beta_t\, dt + \sqrt{\beta_t}\, d\mathbf{w}, \tag{4}$$

where $\boldsymbol{\mu}$ is the mean of a prior distribution, $\boldsymbol{\Sigma}$ is the covariance matrix, and $\beta_t$ is the noise schedule. The mean is estimated by a prior text encoder. In Grad-TTS, the covariance is assumed as an identity matrix to simplify computation; we keep this variable in the exposition for the sake of completeness.

Since the drift coefficient $\mathbf{f}(\cdot, t)$ is affine, the transition kernel $p_{0t}(\mathbf{x}_t|\mathbf{x}_0)$ is a Gaussian distribution whose mean and variance can be obtained in closed form. Let $\gamma_t = \exp(-\int_0^t \beta_s\, ds)$. The transition kernel can be written as

$$p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\{(\mathbf{I} - \sqrt{\gamma_t})\boldsymbol{\mu} + \sqrt{\gamma_t}\mathbf{x}_0, \boldsymbol{\Sigma}(\mathbf{I} - \gamma_t)\}. \tag{5}$$

In an infinite time horizon where $\lim_{t\to T} \gamma_t = 0$, the transition kernel simply converges to $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In other words, the forward process guarantees that diffusion will yield samples from the prior distribution.

The reverse SDE of Grad-TTS can be derived using Eq. (2) and Eq. (4).

$$d\mathbf{x} = \frac{1}{2}\left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{x}) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right)\beta_t\, dt. \tag{6}$$

Therefore, given a score estimator $\mathbf{s}_\theta$ and an initial condition from the prior distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we can compute Eq. (6) to sample from the learned data distribution.

### 2.3. SDEdit

SDEdit [10] is an SDE-based image editing framework that allows for stroke-based image synthesis. An image is first perturbed by coarse user edits, such as color strokes or rough hand-drawn sketches. The edited image is then diffused to obtain noisy priors from a Gaussian distribution. Throughout the reverse steps that follow, a binary mask is repeatedly applied to the intermediate representation at each step such that the non-edited region of the image retains its original form, whereas the edited portion is transformed by gradients computed by a score estimator that simulates the reverse SDE. When reversing the priors, the score estimator only relies on low frequency components to infer high frequency information according to its learned data distribution, thus synthesizing realistic images from the edited input. SDEdit can be applied to a vast array of tasks without any task-specific fine-tuning or a custom loss function.

**Algorithm 1** Pitch shift algorithm for EdiTTS

**Require:** $\tilde{\boldsymbol{\mu}}$ (time-aligned prior), $\boldsymbol{m}$ (mask for edited region), $\boldsymbol{K}_{ps}$ (pitch shift kernel), $\boldsymbol{S}_g$ (mask for softening gradient), $\beta_1, ..., \beta_T$ (noise schedule), $T$ (total denoising steps)
$\tilde{\boldsymbol{\mu}}_{\text{edit}} \leftarrow \boldsymbol{m} \odot (\boldsymbol{K}_{ps} * \tilde{\boldsymbol{\mu}}) + (\mathbf{1} - \boldsymbol{m}) \odot \tilde{\boldsymbol{\mu}}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{x} \leftarrow \tilde{\boldsymbol{\mu}} + \boldsymbol{\epsilon}$
$\mathbf{x}_{\text{edit}} \leftarrow \tilde{\boldsymbol{\mu}}_{\text{edit}} + \boldsymbol{\epsilon}$
**for** $t = T, ..., 1$ **do**
  $d\mathbf{x}_1 \leftarrow \frac{\beta_t}{2T}(\tilde{\boldsymbol{\mu}} - \mathbf{x} - \mathbf{s}_\theta(\mathbf{x}, \tilde{\boldsymbol{\mu}}, t))$
  $d\mathbf{x}_2 \leftarrow \frac{\beta_t}{2T}(\tilde{\boldsymbol{\mu}}_{\text{edit}} - \mathbf{x}_{\text{edit}} - \mathbf{s}_\theta(\mathbf{x}_{\text{edit}}, \tilde{\boldsymbol{\mu}}_{\text{edit}}, t))$
  $\mathbf{x} \leftarrow \mathbf{x} - d\mathbf{x}_1$
  $\mathbf{x}_{\text{edit}} \leftarrow \mathbf{x}_{\text{edit}} - ((\mathbf{1} - \boldsymbol{S}_g) \odot d\mathbf{x}_1 + \boldsymbol{S}_g \odot d\mathbf{x}_2)$
**end for**
**return** $\mathbf{x}_{\text{edit}}$

## 3. Proposed Method

Unlike score-based image models, which use the standard normal $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as their prior, Grad-TTS uses spectrogram-like mean vectors predicted by a prior text encoder. Therefore, we cannot directly modify the input mel-spectrogram as in image editing, as modifying the input would result in a mismatch between the prior predicted from text and the input spectrogram. To circumvent this issue, we apply coarse edits to the encoded priors. Specifically, the input mel-spectrogram is first diffused without any user edits. Then, perturbations are directly applied on the sampled priors before the score estimator simulates the reverse SDE. These coarse perturbations approximate the effects of editing, similar to how hand-drawn color strokes are poor estimations of photo-realistic details.

### 3.1. Pitch Shift

Pitch shifting is a task in which the pitch of an audio segment is modified according to user demand. Motivated by the simple definition of spectrograms as representations of frequencies over time, we directly shift the spectrogram-like latents $\tilde{\boldsymbol{\mu}}$ produced by the prior network along the frequency axis. An upward shift roughly corresponds to higher pitch, and vice versa. In practice, we pad the priors and convolve a weighted averaging kernel $\boldsymbol{K}_{ps}$ over $\tilde{\boldsymbol{\mu}}$ along the frequency axis while applying a binary mask $\boldsymbol{m}$ to ensure that non-editable regions of the spectrogram remain unchanged. $\mathbf{x}$ and $\mathbf{x}_{\text{edit}}$ are each sampled from the original and perturbed prior distributions, respectively. At each reverse step, non-editable regions of $\mathbf{x}_{\text{edit}}$ generally follow $\mathbf{x}$, whereas editable regions are updated by score estimates computed from the perturbed prior. We employ a gradient softening mask $\boldsymbol{S}_g$ to better ensure that junctures between edited and non-edited segments are smooth and continuous. The pitch shift algorithm is schematically shown in Alg. 1.

### 3.2. Content Replacement

Content replacement is a task in which an audio segment in one source is substituted with that from another. Here, we only consider a standard replacement task involving two audio files, but the approach can be extended to any number of audio sources. First, the two input waveforms are appropriately sliced to obtain $\tilde{\boldsymbol{\mu}}^{\boldsymbol{a}}_{\text{src}}, \tilde{\boldsymbol{\mu}}^{\boldsymbol{b}}_{\text{trg}}$ and $\tilde{\boldsymbol{\mu}}^{\boldsymbol{c}}_{\text{src}}$, where superscript $b$ denotes an editable region, and $a$ and $c$ are adjacent segments. We concatenate the three chunks using a soft mask $\boldsymbol{S}_c$ to produce a perturbed prior, $\tilde{\boldsymbol{\mu}}_{\text{edit}}$. Similar to pitch shifting, both $\mathbf{x}$ and $\mathbf{x}_{\text{edit}}$ are denoised

**Algorithm 2** Content replacement algorithm for EdiTTS

**Require:** $\tilde{\boldsymbol{\mu}}_{\text{src}}$ (time-aligned source prior), $\tilde{\boldsymbol{\mu}}_{\text{trg}}$ (time-aligned target prior), $\boldsymbol{m}_{\text{src}}$ (mask for edited region in $\tilde{\mu}_{\text{src}}$), $\boldsymbol{m}_{\text{trg}}$ (mask for extracted region in $\tilde{\boldsymbol{\mu}}_{\text{trg}}$), $\boldsymbol{S}_c$ (mask for softening concatenation), $\boldsymbol{S}_g$ (mask for softening gradient), $\beta_1, ..., \beta_T$ (noise schedule), $T$ (total denoising steps)

$\tilde{\boldsymbol{\mu}}^{\boldsymbol{a}}_{\text{src}}, \tilde{\boldsymbol{\mu}}^{\boldsymbol{c}}_{\text{src}} \leftarrow \texttt{split_source_prior}(\tilde{\boldsymbol{\mu}}_{\text{src}}, \boldsymbol{m}_{\text{src}})$

$\tilde{\boldsymbol{\mu}}^{\boldsymbol{b}}_{\text{trg}} \leftarrow \texttt{extract_target_chunk}(\tilde{\boldsymbol{\mu}}_{\text{trg}}, \boldsymbol{m}_{\text{trg}})$

$\tilde{\boldsymbol{\mu}}_{\text{edit}} \leftarrow \texttt{soft_concat}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{a}}_{\text{src}}, \tilde{\boldsymbol{\mu}}^{\boldsymbol{b}}_{\text{trg}}, \tilde{\boldsymbol{\mu}}^{\boldsymbol{c}}_{\text{src}}, \boldsymbol{S}_c)$

$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\mathbf{x}_{\text{trg}} \leftarrow \tilde{\boldsymbol{\mu}}_{\text{trg}} + \boldsymbol{\epsilon}$

$\mathbf{x}_{\text{edit}} \leftarrow \tilde{\boldsymbol{\mu}}_{\text{edit}} + \boldsymbol{\epsilon}$

**for** $t = T, ..., 1$ **do**

$d\mathbf{x}_1 \leftarrow \frac{\beta_t}{2T}(\tilde{\boldsymbol{\mu}}_{\text{trg}} - \mathbf{x}_{\text{trg}} - \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_{\text{trg}}, \tilde{\boldsymbol{\mu}}_{\text{trg}}, t))$

$d\mathbf{x}_2 \leftarrow \frac{\beta_t}{2T}(\tilde{\boldsymbol{\mu}}_{\text{edit}} - \mathbf{x}_{\text{edit}} - \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_{\text{edit}}, \tilde{\boldsymbol{\mu}}_{\text{edit}}, t))$

$\mathbf{x}_{\text{trg}} \leftarrow \mathbf{x}_{\text{trg}} - d\mathbf{x}_1$

$\mathbf{x}_{\text{edit}} \leftarrow \mathbf{x}_{\text{edit}} - (\texttt{slice_and_reshape}(d\mathbf{x}_1, \boldsymbol{S}_g) + (\mathbf{1} - \boldsymbol{S}_g) \odot d\mathbf{x}_2)$

**end for**

**return** $\mathbf{x}_{\text{edit}}$

simultaneously, with gradient softening masks applied throughout the reverse process. Alg. 2 delineates these steps in detail.

Empirically, we found that soft concatenation reduced the occurrence of infelicitous mixing between the coda and onset of combined segments. Mispronunciation is attributed to the fact that the text encoder employs self-attention and convolutions, which facilitate token mixing on both local and global levels. As a result, a single frame not only contains information about the phoneme in its immediate index, but also bleeds hints about adjacent phonemes. Soft concatenation serves to weaken signals from non-immediate components near frame junctures.

# 4. Experiments

## 4.1. Setup

EdiTTS was evaluated on the LJ speech dataset [18]. We conducted two experiments to separately assess pitch shifting and content editing algorithms, employing HiFi-GAN [19] as the vocoder. We also used the official open-sourced checkpoint from Grad-TTS [8]. The number of reverse steps $T$ was set to 1000 during inference. The value of the softening gradient mask $\boldsymbol{S}_g$ corresponding to editable regions of the audio was set to 1, whereas non-editable segments were masked according to $\sum_{k=i}^{16} 2^{16-k} / \sum_{k=0}^{16} 2^k$, where $i \in [1, 16]$ is the positional difference between the current index and the nearest editable region. The remainder of $\boldsymbol{S}_g$ was set to 0. Similarly, non-editable regions of the softening concatenation mask $\boldsymbol{S}_c$ were set to $0.1 \times (10 - j)$, where $j \in [1, 9]$ is the positional difference as defined above. The remainder of $\boldsymbol{S}_c$ was set to zero. For upward pitch shifting, we set $\boldsymbol{K}_{ps}$ to (0.2, 0.2, 0.6, 0, 0); downward kernel, (0, 0, 0.6, 0.2, 0.2).

## 4.2. Pitch Shift

We compared EdiTTS with two existing baselines: FastPitch [20] and WORLD [21]. Additionally, we considered an ablated formulation of EdiTTS, denoted as "Mel shift" in Table 1, in which only coarse mel-spectrogram shifting was applied on the output of Grad-TTS without denoising reversal. We conducted a mean opinion score (MOS) test with 18 participants through Amazon Mechanical Turk, a crowd-sourcing platform.

Table 1: *Mean Opinion Scores with 95% confidence interval*

| Method | Pitch Up | Pitch Down | Overall |
|---|---|---|---|
| WORLD [21] | $2.79 \pm 0.12$ | $2.98 \pm 0.13$ | $2.88 \pm 0.09$ |
| FastPitch [20] | $3.77 \pm 0.10$ | $3.78 \pm 0.10$ | $3.78 \pm 0.07$ |
| Mel shift | $3.75 \pm 0.10$ | $3.69 \pm 0.09$ | $3.73 \pm 0.07$ |
| EdiTTS (Ours) | $\mathbf{4.04 \pm 0.09}$ | $\mathbf{3.86 \pm 0.10}$ | $\mathbf{3.95 \pm 0.07}$ |

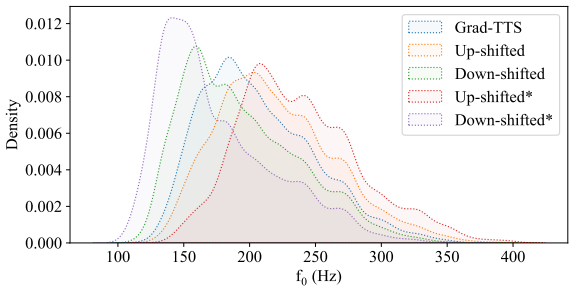


Figure 1: $f_0$ *statistics of pitch-altered samples*

15 sentences were randomly chosen from the LJ speech dataset test split. For each sentence, we randomly sampled an editable target region, which was used to generate pitch-unchanged, pitch-up, and pitch-down utterances. Given a pair of pitch-unchanged and adjusted samples, participants were asked to provide a holistic score on a five-point scale, based on (1) how natural the pitch-adjusted sample is overall, and (2) how discernible the edited pitch is compared to the reference sample.

As shown in Table 1, EdiTTS outperforms WORLD and FastPitch. In particular, EdiTTS's pitch-up score lies beyond the confidence interval in comparison to the baseline methods, suggesting statistical significance. The ablated baseline performed on par with FastPitch. We hypothesize that this is due to the relatively unnatural prosodic patterns exhibited in FastPitch samples. On the other hand, the ablated baseline leverages well-formed mel-spectrograms generated by Grad-TTS, albeit with some artifacts introduced by kernel convolution. The score discrepancy between EdiTTS and the mel shift baseline demonstrates that post-synthesis denoising edit is an integral component of EdiTTS, and that spectral perturbation alone is insufficient to achieve satisfactory output quality.

To further investigate the effectiveness of the pitch shifts created by EdiTTS, we extracted $f_0$ values of the edited regions using CREPE [22]. We scrutinized the relationship between the pitch shift kernel and the magnitude of the resulting pitch shift. To this end, EdiTTS was configured with two different kernels: the default $\boldsymbol{K}_{ps}$ at (0.2, 0.2, 0.6, 0, 0), and the more aggressive mode at (0.4, 0.4, 0.2, 0, 0), denoted in Fig. 1 with an asterisk (*). The density plot shows a modal progression from aggressive-down, default-down, Grad-TTS, default-up, and aggressive-up configurations. In particular, we found that the pitch variation of the modified setup is approximately double that of the default mode, suggesting that the configuration of the pitch shift kernel offers direct control over the magnitude of the pitch shift in the edited sample.

We performed additional sample-level qualitative experiments for closer examination of pitch shift patterns. Fig. 2 illustrates a plot of an up-down and down-up pitch alteration sequence with two discrete editable regions on a single randomly chosen sentence. Shaded regions represent edited segments. We

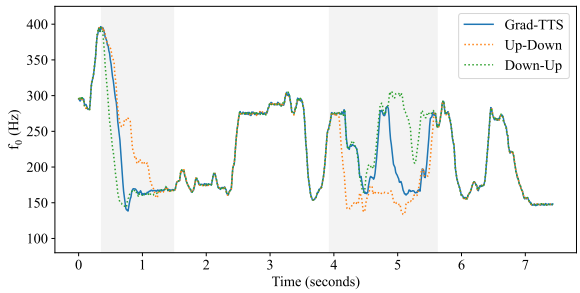


Figure 2: *$f_0$ plot of up-down and down-up samples*

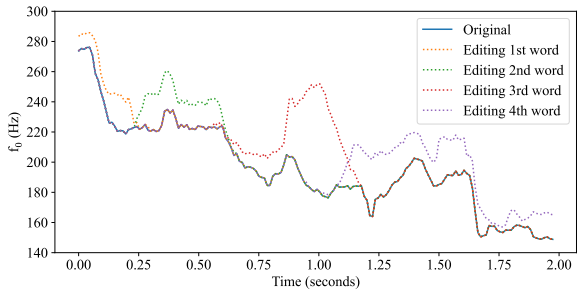


Figure 3: *$f_0$ plot of word-level pitch altered samples*

were able to empirically verify that EdiTTS can apply multiple edits to a single sample, and that doing so does not affect other non-editable regions unspecified by the user.

Another topic of interest was the granularity of edits afforded by EdiTTS. A highly realistic application of speech editing is the modification of $f_0$ to remove or create accentuation in a recording. We applied pitch-up shifts on each word of a LJ Speech test sentence, *'from August nineteen sixty-two'* and report the result in Fig 3. We observed that the pitch of each target word increased properly while leaving the remaining parts unchanged. In particular, a non-trivial pitch shift was achieved in the third word, creating discernible accentuation.

### 4.3. Content Replacement

For content edit evaluation, we compared EdiTTS against Grad-TTS [8] and naive spectrogram concatenation, in which spectrogram frames from disparate audio sources were simply joined together to emulate the effects of content substitution.

We prepared 80 test sentences from the LJ speech dataset by replacing one random word with its synonym using the `nltk` library [23]. The resulting sentences were manually reviewed by a native speaker to ensure that they read naturally. For each sentence, we synthesized two audio samples: one through Grad-TTS using the synonym-replaced script, and another through EdiTTS, where the selected word in the original audio sample was replaced with a synthesized utterance of its synonym. We used the Google Cloud speech-to-text API [24] to examine the word error rate (WER) and the character error rate (CER) of content edits performed by EdiTTS.

As shown in Table 2, EdiTTS substantially outperforms the naive mel concatenation baseline and marginally lags behind Grad-TTS. Note that the error rates achieved by Grad-TTS can be viewed as upper bounds for our method since EdiTTS itself employs Grad-TTS for spectrogram synthesis. We observe a considerable gap between the error rates given by the baseline

Table 2: *Word and character error rates*

| Method | WER (%) | CER (%) |
|---|---|---|
| Mel concat | 11.6129 | 4.4241 |
| Grad-TTS [8] | 8.2258 | 2.8215 |
| EdiTTS (Ours) | 9.1129 | 3.2187 |

and Grad-TTS. Concatenating mel-spectrograms extracted from different utterances introduces prosodic discontinuities and audible artifacts, which adversely affect the error rate. In contrast, EdiTTS processes the concatenated input via denoising reversal, which smoothens such irregularities. The discrepancy between the error rates of naive spectrogram concatenation and EdiTTS indicates that score-based denoising of perturbed priors is integral to improving the quality of the final output.

## 5. Limitations

We discuss some limitations of EdiTTS and ways to improve it.

**Speed.** Generating samples from diffusion models is slow due to the iterative nature of their sampling process. Because EdiTTS relies on the reverse process of the score-based backbone, the statement also applies. Improving the sampling speed of diffusion models is an ongoing area of research [25], and we fully expect these developments to give rise to better diffusion-based TTS and speech editing methodologies.

**Specificity.** Although EdiTTS enables different degrees of pitch shifts depending on the pitch shift kernel, the exact amount of shift cannot be specified. This contrasts with FastPitch or WORLD, which explicitly accepts $f_0$ values to generate samples that follow the input condition. One possible step towards improvement in this direction would be to apply different kernels for each micro-segment within the editable region instead of fixing a single $\boldsymbol{K}_{ps}$ for all frames.

**Alignment.** EdiTTS relies on text-to-speech alignments found by Grad-TTS to modify Gaussian latents that correspond to user-selected text. While the error rates reported in Table 2 suggest that misalignments rarely occur, EdiTTS could edit irrelevant portions of the audio if inaccurate alignments are found. We believe ongoing advancements in robust TTS alignments [26] will help improve not only EdiTTS, but also non-autoregressive TTS pipelines in general.

## 6. Conclusion

We propose EdiTTS, a score-based speech editing methodology for fine-grained pitch and content editing. EdiTTS applies coarse perturbations in the diffused prior space to induce desired edits from the model during denoising reversal. A marked advantage of EdiTTS is its ease of applicability: we leverage the TTS backbone as-is without any additional training or auxiliary module for each editing task. Through listening tests and error rate analyses, we demonstrate that EdiTTS achieves competitive edge over simple mel-spectrogram engineering, as well as performing on par with the original Grad-TTS and outperforming existing baselines on standard speech editing tasks.

## 7. Acknowledgement

This research was supported by the Korean Ministry of Culture, Sports and Tourism and the Korea Creative Content Agency (project number: R2021050007).

# 8. References


[1] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *NeurIPS*, 2019.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020.

[3] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," 2021.

[4] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *ICLR*, 2021.

[5] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform generation," in *ICLR*, 2021.

[6] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," in *ICLR*, 2021.

[7] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, and N. S. Kim, "Diff-TTS: A denoising diffusion model for text-to-speech," in *INTERSPEECH*, 2021.

[8] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-TTS: A diffusion probabilistic model for text-to-speech," in *ICML*, 2021.

[9] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. Kudinov, and J. Wei, "Diffusion-based voice conversion with fast maximum likelihood sampling scheme," 2021.

[10] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "SDEdit: Image synthesis and editing with stochastic differential equations," 2021.

[11] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "ILVR: Conditioning method for denoising diffusion probabilistic models," in *ICCV*, 2021.

[12] Z. Jin, G. J. Mysore, S. Diverdi, J. Lu, and A. Finkelstein, "VoCo: Text-based insertion and replacement in audio narration," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073702

[13] M. Morrison, L. Rencker, Z. Jin, N. J. Bryan, J.-P. Caceres, and B. Pardo, "Context-aware prosody correction for text-based speech editing," in *ICASSP*, 2021.

[14] É. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," in *Speech Commun.*, 1989.

[15] D. Tan, L. Deng, Y. T. Yeung, X. Jiang, X. Chen, and T. Lee, "EditSpeech: A text based speech editing system using partial inference and bidirectional fusion," in *ASRU*, 2021.

[16] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[17] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *NeurIPS*, 2020.

[18] K. Ito and L. Johnson, "The LJ speech dataset," 2017.

[19] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *NeurIPS*, 2020.

[20] A. Lańcucki, "FastPitch: Parallel text-to-speech with pitch prediction," in *ICASSP*, 2021.

[21] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 7, 2016.

[22] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *ICASSP*, 2018.

[23] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.

[24] "Google cloud speech-to-text." [Online]. Available: https://cloud.google.com/speech-to-text/

[25] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*, 2021.

[26] R. Badlani, A. Lancucki, K. J. Shih, R. Valle, W. Ping, and B. Catanzaro, "One tts alignment to rule them all," 2021.