Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

# SLASH: Self-Supervised Speech Pitch Estimation Leveraging DSP-derived Absolute Pitch

*Ryo Terashima*[1], *Yuma Shirahata*[1], *Masaya Kawamura*[1]

[1]LY Corporation, Japan

ryo.terashima@lycorp.co.jp, yuma.shirahata@lycorp.co.jp, kawamura.masaya@lycorp.co.jp

## Abstract

We present SLASH, a pitch estimation method of speech signals based on self-supervised learning (SSL). To enhance the performance of conventional SSL-based approaches that primarily depend on the relative pitch difference derived from pitch shifting, our method incorporates absolute pitch values by 1) introducing a prior pitch distribution derived from digital signal processing (DSP), and 2) optimizing absolute pitch through gradient descent with a loss between the target and differentiable DSP-derived spectrograms. To stabilize the optimization, a novel spectrogram generation method is used that skips complicated waveform generation. In addition, the aperiodic components in speech are accurately predicted through differentiable DSP, enhancing the method's applicability to speech signal processing. Experimental results showed that the proposed method outperformed both baseline DSP and SSL-based pitch estimation methods, attributed to the effective integration of SSL and DSP.
**Index Terms**: self-supervised learning, pitch estimation, voiced/unvoiced detection

## 1. Introduction

Pitch estimation is a fundamental task in the field of speech signal processing. It has been applied to many kinds of applications, such as text-to-speech and emotion recognition, among others [1, 2, 3]. Historically, this task has been performed using digital signal processing (DSP)-based methods [4, 5, 6], but with the advent of deep neural networks (DNNs), data-driven approaches have been proposed [7, 8, 9, 10].

One of the prominent methods using DNNs is the supervised approach [8, 10, 11]. In this approach, a model is trained to estimate the fundamental frequency ($F_0$)[1] from audio data using a simple framework where $F_0$ serves as the target label. Although this approach is effective, it faces challenges such as the difficulty in obtaining reliable target $F_0$ annotations and the tendency of the inference results to be overly dependent on the provided ground truth $F_0$.

These challenges have been addressed by methods based on self-supervised learning (SSL), which do not require labeled data. Various SSL-based methods have been proposed [7, 8, 12], among which one of the most notable is the methods that utilize pitch shift, exemplified by SPICE [8]. These methods leverage the relative values of pitch shifts as the main objective, effectively bypassing the need for manually labeled data. On the other hand, relative pitch difference is not the only clue for pitch prediction. Even when the target pitch labels are unavailable, clues for absolute pitch are also available by properly using DSP techniques. Leveraging these absolute pitch clues in the SSL-based pitch estimation schema is expected to further enhance the model performance.

[1]In general, the term $F_0$ represents a physical property, while pitch represents a perceptual property of sound. In this paper, for convenience, the term "pitch" is used interchangeably with $F_0$.

Therefore, we propose SLASH, a novel pitch estimation model that integrates absolute pitch information into SSL-based model. SLASH incorporates absolute pitch in two ways leveraging DSP. The first approach involves using a DSP technique based on subharmonic summation (SHS) [13] to determine the prior distribution of the $F_0$. This distribution is used to regularize the estimation results, which encourage the model to incorporate the course $F_0$ structure into the pitch estimation. The second approach optimizes the $F_0$ through gradient descent, using a loss between the target spectrogram and a spectrogram generated by a differentiable DSP (DDSP) module. Although this type of $F_0$ optimization is reported to be inefficient [14, 15], the proposed method overcomes the problem by introducing a novel spectrogram generation method. Specifically, the method directly generates a spectrogram of periodic component from $F_0$, without generating a time-domain waveform. The direct generation of spectrogram from $F_0$ simplifies the backpropagation process, enabling the optimization of $F_0$ using gradient descent. Furthermore, because the proposed method targets speech, it also aims to estimate the aperiodicity and voicing flags (V/UV) alongside the $F_0$. By passing the $F_0$, aperiodicity and spectral envelope to a DDSP synthesizer, the aperiodicity can also be optimized using gradient descent.

To demonstrate the effectiveness of the proposed method, experiments on pitch estimation for speech were conducted. Compared to DSP and SSL-based baselines, the proposed method showed superior results across all objective metrics, including $F_0$ RMSE and V/UV Error. Additionally, ablation studies confirmed that each component of the proposed method contributes to improved estimation accuracy.

## 2. SLASH

### 2.1. Model overview

Figure 1 illustrates the architecture of SLASH. As shown in the figure, only the Pitch Encoder is a DNN module, while all other modules are composed of DSP. When viewed as a whole, SLASH functions as an analysis-synthesis system based on the source-filter model [16, 17, 18]. Specifically, the model decomposes the waveform $\boldsymbol{w}$ into $F_0$, spectral envelope, and aperiodicity[2] and then reconstructs the waveform $\tilde{\boldsymbol{w}}$ from them. During training, spectral envelope is calculated by the Spec Env. Estimator, while $F_0$ and Aperiodicity are predicted by the Pitch Encoder. These features are then passed to the DDSP Synthesizer similar to [19], to reconstruct the waveform. In addition, V/UV are estimated through the V/UV Detector. During inference, vocoder features and V/UV are estimated through these modules. In the subsequent sections, we present a series of modules and methods aimed at enhancing prediction accuracy and stabilizing the training process.

[2]We call the group of $F_0$, spectral envelope, and aperiodicity as the *vocoder features* for simplicity.


10.21437/Interspeech.2025-1453

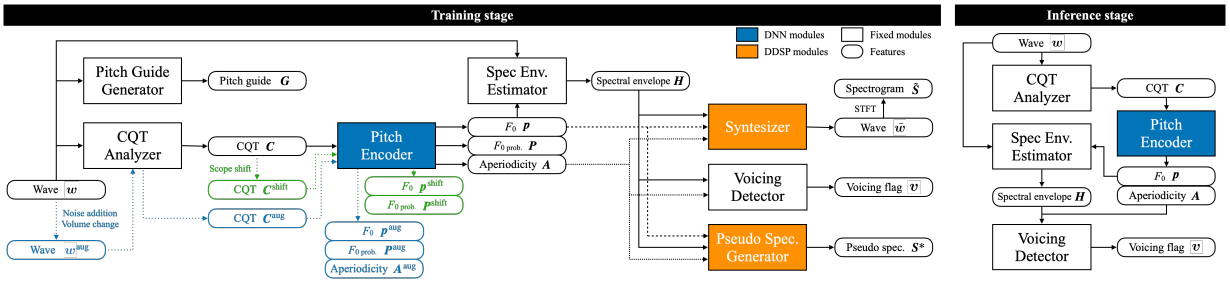



Figure 1: *Model architecture of SLASH. The different types of arrows (dashed, dotted, and solid) are used solely for visibility purposes.*

### 2.2. Pitch consistency loss

Following the success in previous works [7, 8, 12], we adopted the loss of pitch consistency. The key idea of this loss is to predict the pitch difference between the original and the pitch-shifted waveform to learn the relative pitch. For training, Constant-Q Transform (CQT) [20] $\boldsymbol{C} \in \mathbb{R}^{T \times F_c}$ is first extracted from the input waveform $\boldsymbol{w}$, and then a pitch-shifted CQT $\boldsymbol{C}^{\text{shift}}$ is prepared by shifting the scope of $\boldsymbol{C}$. Here, $T$, and $F_c$ are the number of frames and the number of frequency bins in CQT, respectively. $\boldsymbol{C}$ and $\boldsymbol{C}^{\text{shift}}$ are fed into the Pitch Encoder to obtain two sequences of $F_0$ $\boldsymbol{p} \in \mathbb{R}^{T \times 1}$ and $\boldsymbol{p}^{\text{shift}}$, which are used to calculate the pitch consistency loss defined as follows:

$$L_{\text{cons}} = \frac{1}{T} \sum_{t=1}^{T} h \left( \left| \log_2 \boldsymbol{p}_t - \log_2 \boldsymbol{p}_t^{\text{shift}} + \frac{d}{12} \right| \right), \quad (1)$$

where $h(\cdot)$ denotes the Huber norm [21]. $d$ is the frequency axis shift in semitones, converted to octaves by dividing by 12. The architecture of the Pitch Encoder is based on the Pitch encoder of [12], with slight modifications in the output. Specifically, unlike the original module that predicts the amplitudes of periodic and aperiodic components, the Pitch Encoder in SLASH outputs the probability matrix of $F_0$ $\boldsymbol{P} \in \mathbb{R}^{T \times F}$ and an $b$-dimensional Band Aperiodicity (BAP) $\boldsymbol{B} \in \mathbb{R}^{T \times b}$. Here, $F$ denotes the number of frequency bins. $\boldsymbol{p}$ is calculated by taking the weighted average of $\boldsymbol{P}$. Aperiodicity $\boldsymbol{A} \in \mathbb{R}^{T \times K}$ is then calculated by linearly interpolating $\boldsymbol{B}$ on the logarithmic amplitude, where $K$ represents the number of frequency bins. For $\boldsymbol{C}^{\text{shift}}$, $\boldsymbol{P}^{\text{shift}}$ is predicted accordingly.

### 2.3. Pitch guide

Although the pitch consistency loss is effective to learn the relative pitch, it is difficult to predict the absolute pitch with this loss alone. To address this issue, we introduce a pitch guide $\boldsymbol{G} \in \mathbb{R}^{T \times F}$, which is a frame-level prior distribution of $F_0$ calculated by the DSP-based Pitch Guide Generator. $\boldsymbol{G}$ is obtained through the following steps. 1) The fine structure spectrum in logarithmic domain $\psi(\boldsymbol{S}) \in \mathbb{R}^{T \times K}$ is calculated as:

$$\psi(\boldsymbol{S}) = \log(\boldsymbol{S}) - W(\log(\boldsymbol{S})), \quad (2)$$

where $\boldsymbol{S} \in \mathbb{R}^{T \times K}$ denotes the input amplitude spectrogram, and $W(\cdot)$ represents the calculation of the spectral envelope by lag-window method [22]. 2) SHS [13] is applied to $\exp(\psi(\boldsymbol{S}))$, to obtain $\boldsymbol{G}' \in \mathbb{R}^{T \times F}$. Note that the frequency scale is changed from $K$ to $F$ in this process. 3) $\boldsymbol{G}'$ is normalized so that the maximum of every time frame equals 1, resulting in $\boldsymbol{G}$. Figure 2 shows an example of pitch guide $\boldsymbol{G}$ with the corresponding spectrogram. As shown in the figure, the pitch guide well captures the harmonic structure of speech in voiced region, and is expected to serve as a clue for determining the absolute pitch. To encourage the Pitch Encoder to use $\boldsymbol{G}$ as a reference of absolute pitch, the pitch guide loss is defined as follows:

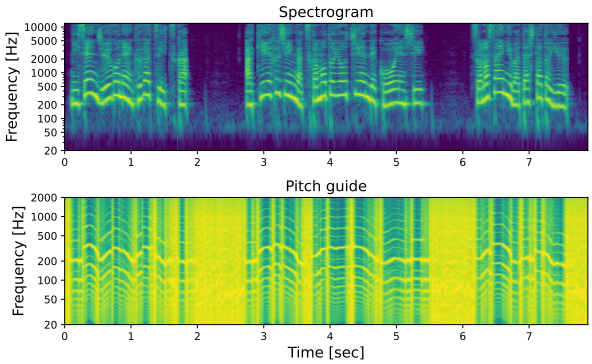



Figure 2: *An example of pitch guide $\boldsymbol{G}$ with log spectrogram.*

$$L_{\text{g}} = \frac{1}{T} \sum_{t=1}^{T} \max \left( 1 - \sum_{f} \boldsymbol{P}_{t,f} \cdot \boldsymbol{G}_{t,f} - m, 0 \right), \quad (3)$$

where $\boldsymbol{P}_{t,f}$ represents $\boldsymbol{P}$ at time frame $t$ and frequency $f$, and $m$ is the hinge parameter to relax the constraint. Pitch guide loss is also applied to $\boldsymbol{P}^{\text{shift}}$. In this case, the pitch guide is shifted according to the amount of pitch shift $\Delta f$:

$$L_{\text{g-shift}} = \frac{1}{T} \sum_{t=1}^{T} \max \left( 1 - \sum_{f} \boldsymbol{P}_{t,f}^{\text{shift}} \cdot \boldsymbol{G}_{t,f-\Delta f} - m, 0 \right).$$

### 2.4. $F_0$ optimization with gradient descent

Another approach to learn the absolute pitch is to synthesize the target waveform from vocoder features through DDSP or DNNs and update the input $F_0$ through gradient descent [23]. However, the gradient descent of $F_0$ from waveform is reported to be ineffective due to the limited perception of gradient orientation relative to $F_0$, and the presence of numerous local minima [14, 15]. To overcome this issue, we propose a novel spectrogram generation method, which directly generate the spectrogram of periodic component from $F_0$, bypassing the waveform generation. This approach is expected to stabilize $F_0$ back propagation by avoiding complex waveform generation processes. Particularly, the $F_0$ gradient descent is performed through DDSP-based Pseudo Spec. Generator, which receives predicted vocoder features as input. The gradient descent is performed by the following steps.

**Excitation spectrogram generation:** The Pseudo Spec. Generator generates a pseudo spectrogram of the periodic excitation signal $\boldsymbol{E}_{\text{p}}^{*} \in \mathbb{R}^{T \times K}$ from $\boldsymbol{p}$. First, the phase of the signal at time $t$ is calculated using the $F_0$ $\boldsymbol{p}_t$, sampling rate $f_s$, and frequency index $k \in \{1, 2, \ldots, K\}$, and $K$:

$$\Phi_{t,k} = \frac{f_s}{2 \boldsymbol{p}_t K} k.$$

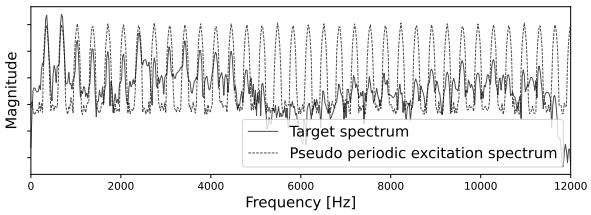



Figure 3: *An example of the target spectrum $\boldsymbol{S}_t$ and the pseudo periodic excitation spectrum $\boldsymbol{E}_t^*$ generated from optimized $\boldsymbol{p}_t$.*

Then, a triangle wave oscillating $\boldsymbol{X}$ between $-1$ and 1 is generated based on $\Phi_{t,k}$:

$$\boldsymbol{X}_{t,k} = \begin{cases} -1, & \text{if } \Phi_{t,k} < 0.5 \\ 4\left|\Phi_{t,k} - \lfloor \Phi_{t,k} \rfloor - 0.5\right| - 1, & \text{otherwise,} \end{cases}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. The pseudo periodic excitation spectrogram $\boldsymbol{E}_{\text{p}}^*$ is then obtained as:

$$\boldsymbol{E}_{\text{p}}^* = \max(\boldsymbol{X}, \varepsilon)^2 + |\boldsymbol{Z} \cdot \varepsilon|\,, \tag{4}$$

where $\varepsilon$ denotes a small magnitude, and $\boldsymbol{Z} \in \mathbb{R}^{T \times K}$ denotes a matrix with each element being independently drawn from standard normal distribution.

**Loss calculation:** After $\boldsymbol{E}_{\text{p}}^*$ is obtained, the entire pseudo spectrogram $\boldsymbol{S}^* \in \mathbb{R}^{T \times K}$ is calculated as follows:

$$\boldsymbol{S}^* = (\boldsymbol{E}_{\text{p}}^* \odot \boldsymbol{H} \odot (1 - \boldsymbol{A})) + (\mathcal{F}(\boldsymbol{e}_{\text{ap}}) \odot \boldsymbol{H} \odot \boldsymbol{A})\,, \tag{5}$$

where $\odot$ denotes the Hadamard product, $\boldsymbol{H} \in \mathbb{R}^{T \times K}$ denotes the spectral envelope, $\boldsymbol{e}_{\text{ap}}$ represents the aperiodic excitation signal in time-domain, and $\mathcal{F}$ represents the Fourier transform. The term $\mathcal{F}(\boldsymbol{e}_{\text{ap}}) \odot \boldsymbol{H} \odot \boldsymbol{A}$ represents the spectrogram of aperiodic component. It is obtained by applying an STFT to the aperiodic component of the Synthesizer output $\tilde{\boldsymbol{w}}$ [3].

Finally, the loss is defined as an L1 norm of the predicted and the target log fine structure spectrum:

$$L_{\text{pseudo}} = \left\|\psi(\boldsymbol{S}^*) - \psi(\boldsymbol{S})\right\|_1 \odot (\boldsymbol{v} \times \boldsymbol{1}_K), \tag{6}$$

and $\boldsymbol{v} \in \mathbb{R}^{T \times 1}$ is a mask of V/UV, which will be introduced in eq. (9). This loss aims to align the peak positions of the harmonics in the excitation signal spectrum $\boldsymbol{E}_t^*$ with those in the target spectrum $\boldsymbol{S}_t$. Figure 3 shows the case where $\boldsymbol{p}_t$ is perfectly optimized. In this case, we can see that the peak positions of $\boldsymbol{E}_t^*$ and $\boldsymbol{S}_t$ are perfectly aligned. Note that the computational graphs of features other than $\boldsymbol{p}$ were detached to focus on the $F_0$ prediction.

### 2.5. Aperiodicity prediction

Since SLASH focuses on speech signals, aperiodicity $\boldsymbol{A}$ and V/UV $\boldsymbol{v}$ are predicted along with the $F_0$ $\boldsymbol{p}$. $\boldsymbol{A}$ is optimized through the gradient descent using the loss between the target spectrogram $\boldsymbol{S}$ and another spectrogram $\tilde{\boldsymbol{S}} \in \mathbb{R}^{T \times F}$, which is obtained from $\tilde{\boldsymbol{w}}$ through STFT. In the frequency domain, $\tilde{\boldsymbol{S}}$ can be expressed by the following equation[3]:

$$\tilde{\boldsymbol{S}} = (\mathcal{F}(\boldsymbol{e}_{\text{p}}) \odot \boldsymbol{H} \odot (1 - \boldsymbol{A})) + (\mathcal{F}(\boldsymbol{e}_{\text{ap}}) \odot \boldsymbol{H} \odot \boldsymbol{A}), \tag{7}$$

where $\boldsymbol{e}_{\text{p}}$ represents a time-domain periodic excitation signal, created by combining multiple sine waves based on $\boldsymbol{p}_t$ and its harmonics [19]. Instead of the simple L1 norm, a generalized energy distance (GED) [24] is used to stabilize optimization:

$$L_{\text{recon}} = \left\|\psi(\tilde{\boldsymbol{S}}^1) - \psi(\boldsymbol{S})\right\|_1 - \alpha \left\|\psi(\tilde{\boldsymbol{S}}^1) - \psi(\tilde{\boldsymbol{S}}^2)\right\|_1, \tag{8}$$

[3] In practice, the periodic and aperiodic component of $\tilde{\boldsymbol{w}}$ is generated from $\boldsymbol{e}_{\text{p}}$ and $\boldsymbol{e}_{\text{ap}}$ in **time**-domain using minimum-phase response. The frequency representation is used in eq. (5) and (7) for brevity.

where $\tilde{\boldsymbol{S}}^1$ and $\tilde{\boldsymbol{S}}^2$ denote two different generated spectrograms, and $\alpha$ is a weight parameter that controls the influence of the repulsive term in the GED. Similar to $L_{\text{pseudo}}$, the loss is calculated by focusing on the fine structure spectrum.

The Voicing Detector calculates the V/UV $\boldsymbol{v}$ as:

$$\boldsymbol{v}' = \frac{M_{\text{p}}}{M_{\text{p}} + M_{\text{ap}}}, \boldsymbol{v} = \begin{cases} 1, & \text{if } \boldsymbol{v}' \geq \theta \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Here, $M_{\text{p}}$ and $M_{\text{ap}}$ are the magnitude of periodic and aperiodic components, which are calculated as the weighted sum of spectral envelope $\boldsymbol{H}$ by $\boldsymbol{A}$. $\theta$ is a threshold parameter.

### 2.6. Noise robust training

To enhance the robustness to noisy inputs, we prepare augmented waveforms by randomly adding noise and changing the volume for input waveforms [7]. $\boldsymbol{C}^{\text{aug}}$ is calculated through the CQT Analyzer, and $\boldsymbol{p}^{\text{aug}}$, $\boldsymbol{P}^{\text{aug}}$, and $\boldsymbol{A}^{\text{aug}}$ are obtained through the Pitch Encoder. Three losses are defined for these features. The first loss $L_{\text{aug}}$ is similar to $L_{\text{cons}}$, which is defined as the Huber norm between $\boldsymbol{p}$ and $\boldsymbol{p}^{\text{aug}}$. The second loss $L_{\text{g-aug}}$ is almost the same as $L_{\text{g}}$, except that $\boldsymbol{P}$ is substituted with $\boldsymbol{P}^{\text{aug}}$. These losses are expected to enhance the robustness of pitch estimation against noisy inputs. The last loss $L_{\text{ap}}$ is introduced to enhance the noise robustness of aperiodicity prediction:

$$L_{\text{ap}} = \left\|\log(\boldsymbol{A}^{\text{aug}}) - \log(\boldsymbol{A})\right\|_1.$$

## 3. Experiments

### 3.1. Experimental setup

**Datasets:** In our experiments, we used two datasets. The first one is **LibriTTS-R** [25], a high-quality multi-speaker English corpus of 585 hours of read English speech designed for text-to-speech use. The second one is **MIR-1K** [26], consisting of 1,000 song clips extracted from 110 karaoke songs. It contains both vocal and instrumental tracks and provides pitch and voicing flag annotations for the vocals. For the evaluation of all systems, we used 250 randomly selected phrases from the MIR-1K dataset. During the experiments, all data were resampled to a sampling rate of 24 kHz.

**Model details:** The CQT was configured with a frame shift of 5 ms, a minimum frequency $f_{\text{min}}$ of 32.70 Hz, and a total of 205 bins, with 24 bins per octave. To accommodate rapid pitch changes, the filter scale was set to 0.5. The pitch encoder processes the central 176 bins. The shift range for $\boldsymbol{C}^{\text{shift}}$ is set to $\pm 14$ bins. $\boldsymbol{C}^{\text{aug}}$ was computed by applying random noise (maximum SNR of -6 dB) and volume variations ($\pm 6$ dB) to the input waveform. The pitch encoder is designed to output a 1024-dimensional $F_0$ probability distribution (i.e. $F = 1024$) mapped onto a logarithmic frequency scale ranging from 20 Hz to 2 kHz, along with an 8-dimensional BAP. The parameter $m$ in eq. (3) was set to 0.5, $\alpha$ in eq. (8) was set to 0.1, $\theta$ in eq. (9) was set to 0.5, and $\varepsilon$ in eq. (4) was set to 0.001. The weights for the core losses were set as follows: $L_{\text{cons}}$ and $L_{\text{pseudo}}$ were set to 10.0, $L_{\text{recon}}$ was set to 5.0. Additionally, $L_{\text{g}}$, $L_{\text{g-shift}}$, $L_{\text{aug}}$, $L_{\text{g-aug}}$, and $L_{\text{ap}}$ were set to 1.0. During the training, we utilized a dynamic batching [27] with an average batch size of 17 samples to create mini-batches. The model was trained for 100,000 steps using the AdamW optimizer [28] with a learning rate of 0.0002.

**Comparison models:** We prepared two SLASH models trained on LibriTTS-R and MIR-1K. The latter was used to better align with the baseline model conditions. A total of 700 phrases were used for training and 50 phrases for validation. As the comparison models, we used methods based on DSP, SSL, and supervised learning. DIO [29] and Harvest [30] are DSP-based $F_0$ estimation methods implemented in the WORLD [17] speech analysis-synthesis system. For sections where the estimated $F_0$ by DIO and Harvest is 0 Hz, linear interpolation of $F_0$

Table 1: *Evaluation results of SLASH compared to digital signal processing, self-supervised learning, and supervised learning baselines.* ***Bold*** *font denotes the best score among models except supervised learning model, which is grayed out.*

| Model | Trained on | Pitch | | | V/UV |
|---|---|---|---|---|---|
| | | RPA (RCA) 50c ↑ | RPA (RCA) 100c ↑ | log-$F_0$ RMSE ↓ | V/UV ER ↓ |
| DIO | - | 0.943 (0.943) | 0.976 (0.976) | 0.030 | 0.052 |
| Harvest | - | 0.945 (0.947) | 0.972 (0.974) | 0.045 | 0.083 |
| SLASH | LibriTTS-R | **0.969 (0.969)** | **0.990 (0.990)** | 0.018 | **0.033** |
| | MIR-1K | 0.967 (0.967) | **0.990 (0.990)** | **0.017** | **0.033** |
| PESTO | MIR-1K | 0.962 (0.966) | 0.982 (0.986) | 0.057 | 0.098 |
| CREPE full | many (supervised) | 0.978 (0.978) | 0.989 (0.990) | 0.038 | 0.101 |

was performed on a logarithmic scale. PESTO is an SSL-based baseline model. We used the official pre-trained model of PESTO[4], which is trained on the MIR-1K dataset. CREPE [9] is a model using supervised learning. We used the official pre-trained model with Viterbi smoothing[5], which is trained on various datasets[26, 31, 32, 4, 33, 34], including MIR-1K. Note that CREPE uses some ground-truth pitch labels during training, while other models do not. For the V/UV estimation of PESTO and CREPE, we used confidence values. Frames with a confidence level of 0.5 or higher were classified as voiced.

**Evaluation metrics:** We predicted pitch and V/UV for the test set of MIR-1K, and evaluated the results with several metrics. Raw pitch accuracy (RPA) measures the proportion of voiced frames where the estimated pitch is within 50 cents of the true pitch. We additionally consider cases within 100 cents [35]. Raw chroma accuracy (RCA) accounts for the proportion of RPA that allows for octave errors [35]. The log-$F_0$ RMSE represents the root mean square error between the true and estimated $F_0$ in logarithmic domain, calculated over voiced frames. Lastly, the voiced/unvoiced error rate (V/UV ER) indicates the error rate in V/UV classification.

### 3.2. Results

#### *3.2.1. Clean signals*

Table 1 summarizes the evaluation results on MIR-1K. Among the DSP and SSL-based methods, which do not use the target labels during training, SLASH achieved the best scores in all metrics. It also showed competitive performance against CREPE, which uses the ground-truth pitch labels in MIR-1K. These results confirm the high pitch and V/UV estimation performance of SLASH, which takes the advantage of both SSL and DSP. In terms of the metrics, SLASH particularly performed well for the log-$F_0$ RMSE compared to the RPA (RCA). This result suggests that the prediction of SLASH rarely deviates significantly from the true value even when it does not fall within the accuracy range of RPA. In addition, SLASH outperforms the baseline SSL model PESTO in all metrics. This suggests that introducing loss functions for absolute pitch is effective to enhance the pitch prediction accuracy. In terms of the training dataset, SLASH trained on LibriTTS-R demonstrated performance nearly equivalent to that trained on MIR-1K, although the former is an out-of-domain dataset. This indicates that training SLASH on a large amount of dataset like LibriTTS-R leads to a good generalization performance to out-of-domain dataset such as singing voice.

#### *3.2.2. Robustness to noise*

Figure 4 illustrates the RPA and log-$F_0$ RMSE under conditions with added white noise. The results demonstrate that SLASH exhibits high robustness even in noisy environments. Notably, at an SNR of 0 dB, SLASH experienced minimal deterioration, while other systems showed significant degradation in log-$F_0$ RMSE. This is likely due to the waveform augmentation and related loss functions as described in Sec.2.6.

[4]https://github.com/SonyCSLParis/pesto

[5]https://github.com/marl/crepe

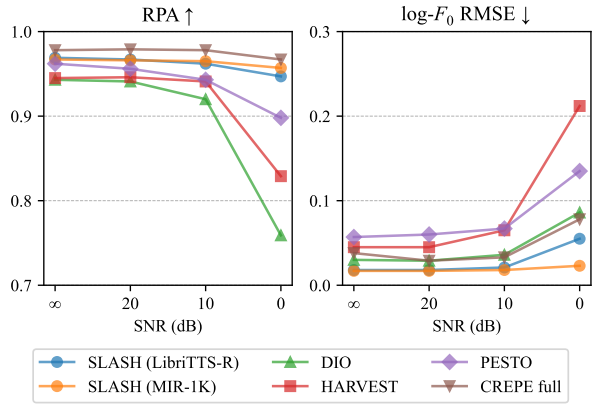



Figure 4: *RPA and log-$F_0$ RMSE in different SNR conditions.*

Table 2: *Evaluation results of the ablation study.*

| Model | Pitch | | V/UV |
|---|---|---|---|
| | RPA ↑ | log-$F_0$ RMSE ↓ | V/UV ER ↓ |
| SLASH | 0.969 | 0.018 | 0.033 |
| w/o $L_g$, $L_{\text{pseudo}}$ | 0.000 | 0.768 | 0.703 |
| w/o $L_{\text{pseudo}}$ | 0.934 | 0.031 | 0.037 |
| w/o GED, $L_{\text{ap}}$ | 0.913 | 0.187 | 0.243 |

#### *3.2.3. Ablation study*

Table 2 shows the results of the ablation study. Without $L_g$ and $L_{\text{pseudo}}$ (i.e., without absolute pitch losses), the scores significantly worsened. This is likely because an absolute reference for pitch is not adequately provided and the model was unable to predict the absolute pitch. Without $L_{\text{pseudo}}$, i.e., when $L_g$ is added, RPA and log-$F_0$ RMSE significantly improve, but not to the same extent as SLASH. These results suggest that while $L_g$ helps capture overall pitch trends, $L_{\text{pseudo}}$ also plays an essential role in enhancing the pitch prediction accuracy. Finally, w/o GED, $L_{\text{ap}}$ represents the case where the aperiodicity is optimized by a simple L1 norm in eq. (8). In this case, the V/UV ER worsened, which confirms the necessity of GED and aperiodicity regularization. In addition, other pitch objectives are also worsened. This is likely due to the use of unreliable V/UV in the calculation of $L_{\text{pseudo}}$ (as shown in eq. (6)).

## 4. Conclusion

We proposed SLASH, an SSL-based pitch estimation method with DSP-derived absolute pitch information. By incorporating the absolute pitch into the model, SLASH enhanced the pitch prediction accuracy of conventional SSL-based methods, which depend on relative pitch objectives. Future work includes expanding SLASH to real-time pitch estimation.

# 5. References


[1] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.

[2] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, "Speech emotion recognition using deep learning techniques: A review," *IEEE Access*, vol. 7, pp. 117 327–117 345, 2019.

[3] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 29, pp. 132–157, 2020.

[4] M. Mauch and S. Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. ICASSP*, 2014, pp. 659–663.

[5] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 124, no. 3, pp. 1638–1652, 2008.

[6] D. Talkin and W. B. Kleijn, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.

[7] A. Riou, S. Lattner, G. Hadjeres, and G. Peeters, "PESTO: Pitch estimation with self-supervised transposition-equivariant objective," in *Proc. ISMIR*, 2023.

[8] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 28, pp. 1118–1128, 2020.

[9] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. ICASSP*, 2018, pp. 161–165.

[10] S. Singh, R. Wang, and Y. Qiu, "DeepF0: End-to-end fundamental frequency estimation for music and speech signals," in *Proc. ICASSP*, 2021, pp. 61–65.

[11] K. Subramani, J.-M. Valin, J. Büthe, P. Smaragdis, and M. Goodwin, "Noise-robust DSP-assisted neural pitch estimation with very low complexity," in *Proc. ICASSP*, 2024, pp. 11 851–11 855.

[12] H.-S. Choi, J. Yang, J. Lee, and H. Kim, "NANSY++: Unified voice synthesis with neural analysis and synthesis," *arXiv preprint arXiv:2211.09407*, 2022.

[13] Y. Ikemiya, K. Itoyama, and K. Yoshii, "Singing voice separation and vocal F0 estimation based on mutual combination of robust principal component analysis and subharmonic summation," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 24, no. 11, pp. 2084–2095, 2016.

[14] B. Torres, G. Peeters, and G. Richard, "Unsupervised harmonic parameter estimation using differentiable DSP and spectral optimal transport," in *Proc. ICASSP*, 2024, pp. 1176–1180.

[15] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," *arXiv preprint arXiv:2001.04643*, 2020.

[16] H. Kawahara and M. Morise, "Technical foundations of tandem-straight, a speech analysis, modification and synthesis framework," *Sadhana*, vol. 36, pp. 713–727, 2011.

[17] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. on Inf. and Syst.*, vol. 99, no. 7, pp. 1877–1884, 2016.

[18] T. Nakano and M. Goto, "A spectral envelope estimation method based on F0-adaptive multi-frame integration analysis," in *Proc. SAPA-SCALE 2012*, 2012, pp. 11–16.

[19] S. Nercessian, "Differentiable world synthesizer-based neural vocoder with application to end-to-end audio style transfer," *arXiv preprint arXiv:2208.07282*, 2022.

[20] J. C. Brown, "Calculation of a constant Q spectral transform," *J. Acous. Soc. Am.*, vol. 89, no. 1, pp. 425–434, 1991.

[21] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.

[22] Y. Tohkura, F. Itakura, and S. Hashimoto, "Spectral smoothing technique in PARCOR speech analysis-synthesis," *IEEE Trans. on Acoustics, Speech, and Signal Process.*, vol. 26, no. 6, pp. 587–596, 1978.

[23] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Proc. NIPS*, vol. 29, 2016.

[24] A. Gritsenko, T. Salimans, R. van den Berg, J. Snoek, and N. Kalchbrenner, "A spectral energy distance for parallel speech synthesis," in *Proc. NIPS*, vol. 33, 2020, pp. 13 062–13 072.

[25] Y. Koizumi, H. Zen, S. Karita, Y. Ding, K. Yatabe, N. Morioka, M. Bacchiani, Y. Zhang, W. Han, and A. Bapna, "LibriTTS-R: A restored multi-speaker text-to-speech corpus," in *Proc. Interspeech*, 2023, pp. 5496–5500.

[26] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 18, no. 2, pp. 310–319, 2009.

[27] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proc. ICASSP*, 2020, pp. 7654–7658.

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[29] M. Morise, H. Kawahara, and H. Katayose, "Fast and reliable F0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech," in *Proc. AES 35th International Conference*, 2009.

[30] M. Morise *et al.*, "Harvest: A high-performance fundamental frequency estimator from speech signals," in *Proc. Interspeech*, 2017, pp. 2321–2325.

[31] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 18, no. 8, pp. 2121–2133, 2010.

[32] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research." in *Proc. ISMIR*, vol. 14, 2014, pp. 155–160.

[33] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez Gutiérrez, and J. P. Bello, "An analysis/synthesis framework for automatic F0 annotation of multitrack datasets," in *Proc. ISMIR*, 2017.

[34] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *Proc. ICML*, 2017, pp. 1068–1077.

[35] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 2, pp. 118–134, 2014.