



Neurodyne: Neural Pitch Manipulation with Representation Learning and Cycle-Consistency GAN

Yicheng Gu^{1,2}, Chaoren Wang², Zhizheng Wu², Lauri Juvela¹

¹Acoustic Lab, Aalto University, Espoo, Finland

²School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

yicheng.gu@aalto.fi

Abstract

Pitch manipulation is the process of producers adjusting the pitch of an audio segment to a specific key and intonation, which is essential in music production. Neural-network-based pitch-manipulation systems have been popular in recent years due to their superior synthesis quality compared to classical DSP methods. However, their performance is still limited due to their inaccurate feature disentanglement using source-filter models and the lack of paired in- and out-of-tune training data. This work proposes Neurodyne to address these issues. Specifically, Neurodyne uses adversarial representation learning to learn a pitch-independent latent representation to avoid inaccurate disentanglement and cycle-consistency training to create paired training data implicitly. Experimental results on global-key and template-based pitch manipulation demonstrate the effectiveness of the proposed system, marking improved synthesis quality while maintaining the original singer identity.

Index Terms: Pitch manipulation, singing voice synthesis, generative adversarial networks (GAN), representation learning

1. Introduction

Pitch manipulation is an essential process in music production where the producer can adjust the pitch of an audio segment to correct out-of-tune notes and improve the intonation. However, the performance of existing pitch-manipulation systems is still limited, which will generate unnatural modified audio with audible artifacts. Thus, building a system that can generate high-fidelity audio given the modified pitch contour is needed.

Classical DSP-based methods can be classified into signal and parametric modification systems. Signal modification systems, such as TD-PSOLA [1], relocate pitch periods in the time domain, whereas parametric modification systems, such as WORLD [2], decompose the signal into acoustic parameters and resynthesize the signal from the modified ones. However, due to inaccuracies in DSP modeling, such systems often have metallic artifacts and timbre inconsistency in the modified audio. Recent advances in deep learning (DL)-based systems offer improved audio quality. Specifically, [3] adapts a two-stage generation scheme conditioned on modified pitch contours, followed by DiffPitcher [4] using a diffusion-based architecture. These systems operate solely on acoustic features and lack audio-level optimization, which limits their synthesis quality. In contrast, DeepAutotuner [5] and KaraTuner [6] generate audio in an end-to-end way, followed by SiFiGAN [7] and HarmonicNet [8] adapt the neural vocoders [9, 10] for decoding, obtaining state-of-the-art (SOTA) performance. To get a faster inference speed, PeriodGrad [11] applies a diffusion model on the time-domain signal, and FIRNet [12] utilizes a harmonic-plus-noise synthesizer to reduce the model size.

Two central problems remain with the existing systems. First, current models [4, 7, 8] rely on a source-filter decomposition [2] for disentanglement, and assume independence between pitch and spectral envelope. However, the independence is not complete, which can lead to timbre inconsistency in modified audio. Second, ground truth paired data with in- and out-of-tune examples are unavailable, significantly limiting the system's generalization ability, especially in extreme modifications.

To address these issues, we propose *Neurodyne*. Instead of using inaccurately disentangled features, our system employs adversarial representation learning to learn a pitch-independent latent representation, following [13] and [14]. Moreover, we apply two kinds of cycle-consistency training schemes, including the regular inversion cycle-consistency [15], and a novel composition cycle-consistency. This allows us to create paired data implicitly based on three widely used pitch-manipulation operations in real-world music production (key, variance, and transient manipulation), making our model more robust in different scenarios. Experiment results in both global-key [7] and template-based [4] pitch-manipulation confirm the effectiveness of our system, which outperforms the baselines in audio quality, singer similarity, and pitch accuracy.

2. Methodology

This section details the model architecture, training scheme, and pitch-manipulation strategies of *Neurodyne*.

2.1. Model Architecture

As shown in Fig. 1a, Neurodyne consists of an encoder, a pitch predictor, a decoder, a multi-scale oscillator, and four different discriminators. The encoder includes an initial Conv1D layer and five CNN-based Resblocks following DAC [16]. The pitch predictor consists of a CNN block with a Conformer [17] encoder, following TorchFCPE³. The decoder includes five CNN-based Resblocks with the oversampling technique applied on the activation function to alleviate aliasing effects following BigVGAN [10] and a final Conv1D layer. The multi-scale oscillator consists of four NSF [18]-based excitation oscillators in different sampling rates followed by four Conv1D layers, where the pitch information will first be converted to excitations in different resolutions and added to the intermediate features after the upsampling layer via the Conv1D layer to avoid aliasing effects in the conditional embedding. We mix time-domain and time-frequency-representation-based discriminators following [19] to obtain a better synthesis quality without sacrificing the inference speed, which are: Multi-Period Discriminator (MPD) [9], Multi-Scale Discriminator (MSD) [9], Multi-Scale STFT Discriminator (MS-STFTD) [20], and Multi-Scale Sub-Band CQT Discriminator (MS-SB-CQTD) [21].

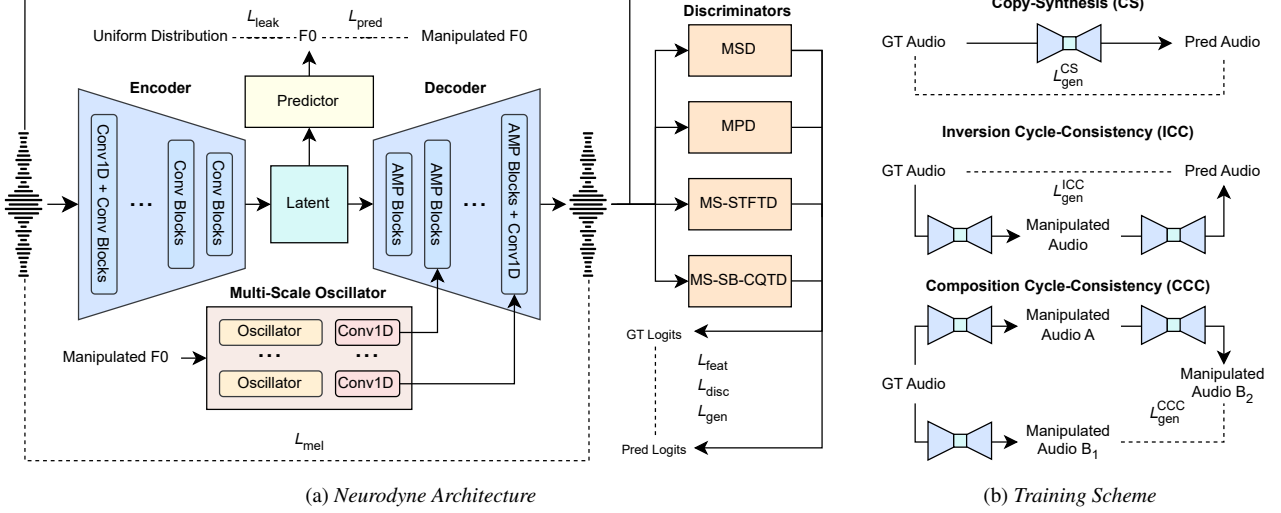


Figure 1: Architecture and training schemes for Neurodyne. The model consists of an encoder, a pitch predictor, a decoder, a multi-scale oscillator, and four different discriminators. The training scheme adopts inversion and composition cycle-consistency training to construct paired in- and out-of-tune data implicitly.

2.2. Adversarial Representation Learning

We apply adversarial representation learning to obtain a pitch-independent latent representation following the Hider-Finder-Combiner (HFC) [13] approach. Specifically, the pitch predictor aims to extract the pitch information from the latent representation. In contrast, the encoder aims to avoid pitch information leakage to the latent representation, giving:

$$\begin{aligned} L_{\text{pitch}}(y, \hat{y}) &= L_{\text{BCE}}(y, \hat{y}), \\ L_{\text{leak}}(\hat{y}) &= n_{\text{bins}}^{2/(n_{\text{bins}}-1)} \text{Var}(\hat{y}), \end{aligned} \quad (1)$$

where n_{bins} is the number of pitch bins, y is the ground truth and \hat{y} is the predicted pitch vector following CREPE [22].

2.3. Cycle-Consistency Training

The proposed cycle-consistency training scheme consists of three complementary training goals, as described below.

2.3.1. Copy-Synthesis (CS)

As illustrated in the top of Fig. 1b, copy-synthesis training aims to help the decoder generate high-fidelity audio segments while distilling the pitch information from the latent representation. Specifically, the generator seeks to reconstruct the input audio that cannot be distinguished by the discriminators while avoiding pitch information leakage to the latent representation, the pitch predictor aims to extract the pitch information from the latent representation, and the discriminator seeks to distinguish the ground truth audio and the predicted audio. Thus, we have:

$$\begin{aligned} L_{\text{gen}}^{\text{CS}} &= L_{\text{leak}}(\hat{y}) + 15L_{\text{mel}}(\text{mel}, \hat{\text{mel}}) \\ &+ \sum_{m=1}^M [L_{\text{adv}}(G; D_m) + 2L_{\text{feat}}(G; D_m)], \\ L_{\text{pred}}^{\text{CS}} &= L_{\text{pitch}}(y, \hat{y}), \\ L_{\text{disc}}^{\text{CS}} &= \sum_{m=1}^M L_{\text{adv}}(D_m; G), \end{aligned} \quad (2)$$

where L_{mel} is the multi-scale mel-spectrogram loss adapted from DAC [16], D_m is the m_{th} discriminator. G is the Neurodyne. The L_{adv} and L_{feat} are the adversarial losses and the feature matching loss following HiFiGAN [9].

2.3.2. Inversion Cycle-Consistency (ICC)

As illustrated in the middle of Fig. 1b, inversion cycle-consistency training aims to enhance the pitch-manipulation ability of the model by implicitly creating paired in- and out-of-tuned data in the training process. The main idea is that the pitch-manipulated audio should still be able to be converted back, given the ground truth pitch. Under this scenario, the discriminator loss remains the same as the copy-synthesis training, while the generator and predictor losses become:

$$\begin{aligned} L_{\text{gen}}^{\text{ICC}} &= L_{\text{leak}}(\hat{y}) + L_{\text{leak}}(\hat{y}_{\text{manipulated}}) + 15L_{\text{mel}}(\text{mel}, \hat{\text{mel}}) \\ &+ \sum_{m=1}^M [L_{\text{adv}}(G; D_m) + 2L_{\text{feat}}(G; D_m)], \\ L_{\text{pred}}^{\text{ICC}} &= L_{\text{pitch}}(y, \hat{y}) + L_{\text{pitch}}(y_{\text{manipulated}}, \hat{y}_{\text{manipulated}}). \end{aligned} \quad (3)$$

2.3.3. Composition Cycle-Consistency (CCC)

As illustrated in the bottom of Fig. 1b, composition cycle-consistency training ensures the audio manipulated to a specific pitch contour in one or two steps are identical. Under this scenario, there is no discriminator loss since there is no ground truth audio, while the generator and predictor losses become:

$$\begin{aligned} L_{\text{gen}}^{\text{CCC}} &= L_{\text{leak}}(\hat{y}_A) + L_{\text{leak}}(\hat{y}_{B_1}) \\ &+ L_{\text{leak}}(\hat{y}_{B_2}) + 15L_{\text{mel}}(\hat{\text{mel}}_{B_1}, \hat{\text{mel}}_{B_2}), \\ L_{\text{pred}}^{\text{CCC}} &= L_{\text{pitch}}(y_A, \hat{y}_A) + L_{\text{pitch}}(y_{B_1}, \hat{y}_{B_1}) \\ &+ L_{\text{pitch}}(y_{B_2}, \hat{y}_{B_2}), \end{aligned} \quad (4)$$

where B_1 denotes modifying to the target pitch contour in one step, A denotes the intermediate pitch contour in the two-step pitch manipulation, and B_2 denotes modifying to the target pitch contour in two steps via the intermediate pitch contour.

Table 1: *Global-key pitch manipulation results of different systems. The best and the second best results of every column (except those from Ground Truth) are **bold** and underlined. The MUSHRA scores are within 95% Confidence Interval (CI).*

System	FORMSE (\downarrow)							Q-MUSHRA (\uparrow)							S-MUSHRA (\uparrow)						
	-12	-6	-3	0	+3	+6	+12	-12	-6	-3	0	+3	+6	+12	-12	-6	-3	0	+3	+6	+12
Ground Truth	/	/	/	0.0	/	/	/	/	/	/	93.0 \pm 2	/	/	/	/	/	/	93.0 \pm 2	/	/	/
BigVGAN	/	/	/	16.0	/	/	/	/	/	/	87.5 \pm 3	/	/	/	/	/	/	89.7 \pm 3	/	/	/
WORLD	20.2	24.8	28.1	31.7	37.8	46.4	96.3	42.3 \pm 5	52.7 \pm 5	59.1 \pm 6	75.7 \pm 5	65.7 \pm 5	63.8 \pm 5	55.0 \pm 6	41.5 \pm 5	57.2 \pm 5	67.3 \pm 5	83.2 \pm 5	72.9 \pm 5	68.4 \pm 5	59.8 \pm 5
TD-PSOLA	29.0	<u>21.2</u>	20.8	<u>20.0</u>	28.6	<u>38.4</u>	<u>90.2</u>	42.3 \pm 5	55.4 \pm 6	69.0 \pm 4	90.0\pm2	71.1 \pm 4	60.8 \pm 5	58.3 \pm 5	43.6 \pm 5	59.1 \pm 5	73.4 \pm 4	91.8\pm2	78.8 \pm 4	65.3 \pm 5	62.4 \pm 5
DiffPitcher	32.0	39.4	45.9	52.8	65.1	83.6	163.0	25.5 \pm 4	36.6 \pm 4	44.5 \pm 5	57.2 \pm 6	44.8 \pm 5	41.5 \pm 5	26.5 \pm 4	32.5 \pm 5	47.9 \pm 5	61.0 \pm 5	75.1 \pm 5	63.4 \pm 5	53.5 \pm 6	39.5 \pm 6
SiFi-GAN	<u>19.8</u>	24.2	26.4	29.2	36.4	47.2	102.2	58.0 \pm 6	<u>66.6\pm4</u>	78.0 \pm 4	89.2 \pm 3	<u>78.9\pm3</u>	<u>73.7\pm6</u>	65.9\pm6	51.3 \pm 6	65.0 \pm 5	79.6 \pm 3	90.2 \pm 3	80.0 \pm 4	73.9 \pm 5	66.2\pm5
PC-NSF	23.1	26.4	27.0	25.8	36.2	59.6	231.5	<u>59.4\pm6</u>	<u>66.1\pm5</u>	<u>80.1\pm3</u>	<u>89.8\pm3</u>	78.9 \pm 4	72.3 \pm 5	51.9 \pm 6	<u>52.7\pm6</u>	<u>66.2\pm4</u>	<u>80.1\pm4</u>	<u>90.6\pm2</u>	81.8\pm4	73.6 \pm 5	56.9 \pm 6
Neurodyne	17.6	20.3	<u>22.1</u>	24.0	<u>29.6</u>	36.1	81.3	<u>72.5\pm5</u>	<u>77.7\pm5</u>	81.1\pm3	88.4 \pm 4	79.0\pm4	76.3\pm4	<u>62.0\pm6</u>	64.4\pm6	74.9\pm5	80.6\pm3	90.5 \pm 3	<u>80.1\pm4</u>	75.4\pm5	<u>64.4\pm6</u>

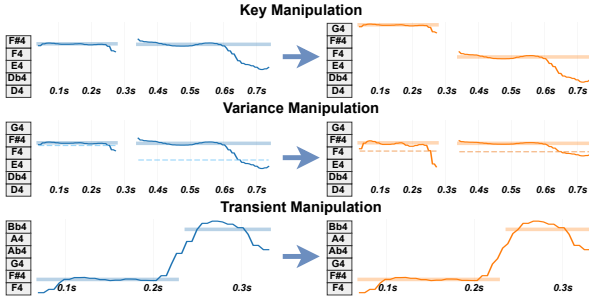


Figure 2: *The pitch manipulation strategy.*

2.4. Pitch Manipulation Strategy

As illustrated in Fig. 2, the pitch-manipulation strategy consists of three operations between the pitch contour and their center MIDI notes. The note value and rough segmentation are obtained from SOME¹, and the detailed segmentation is derived following CREPENotes [23]. Specifically, 1) the key manipulation will randomly shift the pitch contour regarding each note up or down, 2) the variance manipulation will randomly shift the variance computed by subtracting the pitch contour and the note value up or down, 3) the transient manipulation will randomly adjust the variance of the pitch contour between steady-state notes and the linear ramp between the notes.

3. Experiments

We evaluate the effectiveness of our proposed system in two settings. We apply global-key [7] and template-based [4] pitch manipulation to compare the robustness and evaluate the application in real-world scenarios. In global-key pitch manipulation, utterances will be globally manipulated by a specific amount of semitones. In template-based pitch manipulation, an out-of-tune audio segment will be adjusted based on an in-tune reference. The audio samples are available on our demo page².

3.1. Experiment Setup

Datasets: Following [24, 25, 26, 27], we collected all publicly available academic datasets to obtain a diversified large-scale data mixture, which is illustrated in Table 2. We use the paired recordings in PopBuTFy [3] for evaluating template-based pitch manipulation, where 200 utterance pairs with significant pitch differences were selected. We randomly selected 1% samples from the remaining dataset to form the test set for global-key pitch manipulation. All the remaining samples were used in training, resulting in a 580-hour data mixture.

¹<https://github.com/openvpi/SOME>

²<https://www.yichenggu.com/Neurodyne/>

Table 2: *Statistics of the singing voice datasets.*

Dataset	Dur. (hour)	Style	Lang.
NUS-48E [28]	2.8	Children/Pop	ZH
Opera [29]	2.6	Opera	IT/ZH
VocalSet [30]	8.8	Opera	EN
CSD [31]	4.6	Children	EN/KO
PJS [32]	0.5	Pop	JA
NHSS [33]	4.1	Pop	EN
OpenSinger [34]	51.8	Pop	ZH
Kiritan [35]	1.2	Pop	JA
KiSing [36]	0.9	Pop	ZH
PopCS [37]	5.9	Pop	ZH
M4Singer [38]	29.7	Pop	ZH
PopBuTFy [3]	30.7	Pop	EN
Opencpop [39]	5.2	Pop	ZH
SingStyle111 [40]	12.8	Children/Folk/Jazz Opera/Pop/Rock	EN/IT/ZH
GOAT [41]	4.5	Opera	ZH
ACESinger [42]	321.8	Pop	EN/ZH
GTSinger [43]	96.8	Folk/Jazz Opera/Pop	ZH/EN/JA KO/RU/ES FR/DE/IT

Preprocessing: We resampled all the training data to 44.1kHz. We use the TorchFCPE³ to extract the F0. For computing the input features, we use an FFT size of 2048, hop size of 512, window length of 2048, Mel filters of 128, and Mel-coefficient orders of 39. The mel-spectrogram is normalized in log-scale with values $\leq 1e-5$ clipped to 0.

Training: All the models are trained using the AdamW [44] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.998$, and an learning rate of $1e-4$, and the exponential decay scheduler with a factor $\gamma = 0.999996$. All the experiments are conducted on 8 MI250X GPUs with the maximum available batch size for 1M steps.

Configurations: For baseline systems, BigVGAN [10] is used as the benchmark for copy-synthesis. We use WORLD [2] and TD-PSOLA [1] as the DSP-based baselines. We use DiffPitcher [4], SiFi-GAN [7], and HarmonicNet [8] as the DL-based baselines. For SiFi-GAN, we use the 44.1 kHz configuration⁴. For HarmonicNet, we use the implementation open-sourced by the OpenVPI team⁵, denoted as PC-NSF. For Neurodyne, the encoder adjusts the DAC [16] with an output dimension of 128; the pitch predictor adjusts the TorchFCPE³ with the latent representation as the input; the decoder applies $2\times$ over-sampling to the NSF-HiFiGAN [45]⁵; the multi-scale oscillator uses sampling rates of [5512.5, 11025, 22050, 44100]; the MS-SB-CQT Discriminator [21] computes 10 octaves to conform the 44.1 kHz training, with other discriminators unchanged.

³<https://github.com/CNChTu/FCPE>

⁴<https://github.com/tonnetonne814/>

SiFi-VITS2-44100-Ja

⁵<https://github.com/openvpi/SingingVocoders>

Table 3: *Template-based pitch manipulation results of different systems. The best and the second best results of every column (except those from Ground Truth) are bold and underlined. The MUSHRA scores are within 95% Confidence Interval (CI).*

System	FPC (\uparrow)	F0RMSE (\downarrow)	Q-MUSHRA (\uparrow)	S-MUSHRA (\uparrow)
WORLD	0.951	40.7	61.9 \pm 3	67.8 \pm 3
TD-PSOLA	<u>0.970</u>	<u>30.6</u>	67.3 \pm 3	71.1 \pm 3
DiffPitcher	0.868	77.2	41.9 \pm 3	53.9 \pm 4
SiFi-GAN	0.969	31.3	77.5 \pm 2	78.4 \pm 2
PC-NSF	0.966	34.8	<u>78.2\pm2</u>	<u>78.7\pm2</u>
Neurodyne	0.978	27.1	78.3\pm2	78.8\pm2

3.2. Evaluation Metrics

Objective Evaluation: We use the Amphion [46] toolkit for objective evaluation. We investigate objective metrics focusing on F0 accuracy, audio quality, and singer similarity. For F0 accuracy, we use the F0 Root Mean Square Error (F0RMSE) and F0 Pearson Correlation Coefficient (FPC) [45, 47]. We use a pre-trained MOS predictor [48] to obtain the predicted audio quality score (MOS-Pred); we use all open-sourced SSL models from [49] to compute the singer similarity (SIM-O) and report their average score for the ablation study.

Subjective Evaluation: We use a MUSHRA-like test to evaluate the audio quality and singer similarity subjectively. A total of 35 and 10 utterances will be assessed in each setting individually. Listeners were asked to give quality and similarity scores (denoted as Q- and S-MUSHRA) between 1 and 100 for global-key or template-based pitch-manipulated utterances from different systems given the ground truth or in- and out-of-tune audio as the reference. We invited 15 volunteers who are experienced in audio generation with the ability to distinguish out-of-tune segments to attend the evaluation.

3.3. Global-Key Pitch Manipulation

The global-key pitch manipulation evaluation results are illustrated in Table 1. It can be observed that 1) neural-vocoder-based systems perform significantly better than the two-stage synthesis and DSP-based systems, confirming the effectiveness of adversarial training with audio-level losses; 2) in copy-synthesis, Neurodyne performs better subjectively compared with BigVGAN, which shows the effectiveness of the less information loss brought by the adversarial representation learning, making it easier to generate high-fidelity audio; 3) in different pitch-manipulation scenarios, Neurodyne performs better than the baseline systems with smaller F0RMSE and higher quality and similarity MUSHRA scores. This illustrates the effectiveness of feature disentanglement and robustness, which is brought by cycle-consistency training and adversarial representation learning. Specifically, the network will automatically adjust the latent representation regarding the modified F0 to avoid timbre inconsistency brought by the inaccurate disentanglement. Meanwhile, extreme pitch-manipulation scenarios are seen in the training stage due to the implicit data construction, which leads to improved F0 accuracy and audio quality.

3.4. Template-Based Pitch Manipulation

The template-based pitch manipulation evaluation results are illustrated in Table 3. Our system performs better in subjective synthesis quality, singer similarity, and objective pitch accuracy in both absolute value and relative trajectory, which shows the effectiveness of our model architecture, training schemes, and pitch-manipulation strategies.

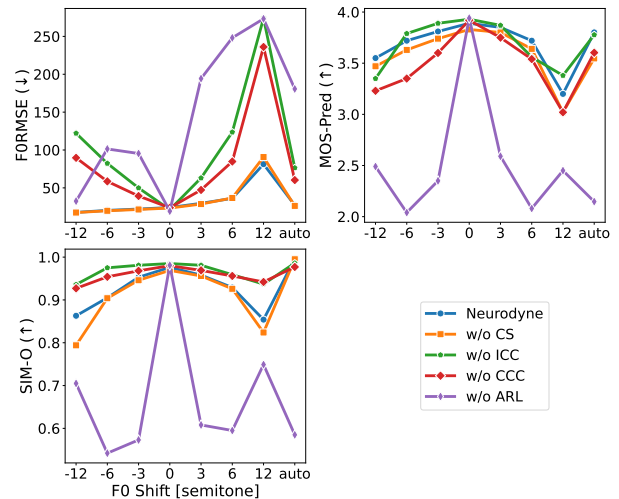


Figure 3: *The evaluation results of the ablation study. “CS” means copy-synthesis, “ICC” means inversion cycle-consistency, “CCC” means composition cycle-consistency, and “ARL” means adversarial representation learning.*

To further illustrate the effectiveness of our proposed system, we also compared Neurodyne with two SOTA commercial softwares. Specifically, we manually selected 4 Chinese and 4 English representative paired samples and manually conducted pitch manipulation. We post the audio samples on our demo page², and it can be heard that our model performs equivalently or even better compared with these commercial plugins.

3.5. Ablation Study

We conducted an ablation study to illustrate the effectiveness of cycle-consistency training and adversarial representation learning, as shown in Fig. 3. Firstly, the system without copy-synthesis training has worse synthesis quality and singer similarity scores, showing it is necessary to conduct copy-synthesis training to maintain the model performance. Moreover, the system without inversion or composition cycle-consistency training will fail to manipulate pitch according to the large F0RMSE values. This is because the constrain for the overall system is loosened, and the decoder will ignore the given pitch condition and utilize the pitch information leaked from the encoder to the latent representation instead, generating either audio with the original pitch contour or unnatural audio comprising singing voices in both pitches (original and shifted). Lastly, the system without adversarial representation learning will only generate reasonable audio in the copy-synthesis scenario since the pitch information is all encoded in the latent representation. The model will be confused when the pitch condition is different, thus generating metallic noises, as illustrated in the unreasonably large F0RMSE, MOS-Pred, and SIM-O values.

4. Conclusion

This paper introduces Neurodyne, a novel pitch manipulation system that utilizes adversarial representation learning and cycle-consistency training. It obtains optimized pitch-independent features that avoid the artifacts brought by source-filter-model-based disentanglement. It implicitly creates paired in- and out-of-tune training data to enhance the robustness in different pitch manipulation scenarios. The evaluation results in both global-key and template-based pitch manipulation demonstrate that our proposed system outperforms existing ones regarding audio quality, pitch accuracy, and singer similarity.

5. Acknowledgement

We acknowledge the computational resources provided by the Aalto Science-IT project. We acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LUMI, hosted by CSC (Finland) and the LUMI consortium through a EuroHPC Regular Access call. This work is also supported by the 2023 Shenzhen stability Science Program, the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2023ZT10X044), and the Shenzhen Science and Technology Program (ZDSYS20230626091302006)

6. References

- [1] Francis Charpentier and M. G. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in *ICASSP*, 1986.
- [2] Masanori Morise, et al., "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans Inf Syst*, vol. 99, no. 7, pp. 1877–1884, 2016.
- [3] Jinglin Liu, et al., "Learning the Beauty in Songs: Neural Singing Voice Beautifier," in *ACL*, 2022.
- [4] Jiarui Hai and Mounya Elhilali, "Diff-Pitcher: Diffusion-Based Singing Voice Pitch Correction," in *WASPAA*, 2023.
- [5] Sanna Wager, et al., "Deep Autotuner: A Pitch Correcting Network for Singing Performances," in *ICASSP*, 2020.
- [6] Xiaobin Zhuang, et al., "KaraTuner: Towards End-to-End Natural Pitch Correction for Singing Voice in Karaoke," in *Interspeech*, 2022.
- [7] Reo Yoneyama, et al., "Source-Filter HiFi-GAN: Fast and Pitch Controllable High-Fidelity Neural Vocoder," in *ICASSP*, 2023.
- [8] Keisuke Matsubara, et al., "Harmonic-Net: Fundamental Frequency and Speech Rate Controllable Fast Neural Vocoder," *TASLP*, 2023.
- [9] Jiaqi Su, et al., "HiFi-GAN: High-Fidelity Denoising and Dereverberation Based on Speech Deep Features in Adversarial Networks," in *Interspeech*, 2020.
- [10] Sang-gil Lee, et al., "BigVGAN: A Universal Neural Vocoder with Large-Scale Training," in *ICLR*, 2023.
- [11] Yukiya Hono, et al., "PeriodGrad: Towards Pitch-Controllable Neural Vocoder Based on a Diffusion Probabilistic Model," in *ICASSP*, 2024.
- [12] Yamato Ohtani, et al., "FIRNet: Fundamental Frequency Controllable Fast Neural Vocoder With Trainable Finite Impulse Response Filter," in *ICASSP*, 2024.
- [13] Jacob J. Webber, et al., "Hider-Finder-Combiner: An Adversarial Architecture for General Speech Signal Modification," in *Interspeech*, 2020.
- [14] Zeqian Ju, et al., "NaturalSpeech 3: Zero-Shot Speech Synthesis with Factorized Codec and Diffusion Models," in *ICML*, 2024.
- [15] Jun-Yan Zhu, et al., "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *ICCV*, 2017.
- [16] Rithesh Kumar, et al., "High-Fidelity Audio Compression with Improved RVQGAN," in *NeurIPS*, 2023.
- [17] Anmol Gulati, et al., "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech*, 2020.
- [18] Xin Wang, et al., "Neural Source-filter-based Waveform Model for Statistical Parametric Speech Synthesis," in *ICASSP*, 2019.
- [19] Yicheng Gu, et al., "An Investigation of Time-Frequency Representation Discriminators for High-Fidelity Vocoders," *TASLP*, 2024.
- [20] Alexandre Défossez, et al., "High Fidelity Neural Audio Compression," *Trans. Mach. Learn. Res.*, 2023.
- [21] Yicheng Gu, et al., "Multi-scale sub-band constant-q transform discriminator for high-fidelity vocoder," in *ICASSP*, 2024.
- [22] Jong Wook Kim, et al., "Crepe: A Convolutional Representation for Pitch Estimation," in *ICASSP*, 2018.
- [23] Xavier Riley and Simon Dixon, "CREPE Notes: A new method for segmenting pitch contours into discrete notes," *arXiv:2311.08884*, 2023.
- [24] Haorui He, et al., "Emilia: An Extensive, Multilingual, and Diverse Speech Dataset for Large-Scale Speech Generation," in *SLT*, 2024.
- [25] Haorui He, et al., "Emilia: A Large-Scale, Extensive, Multilingual, and Diverse Dataset for Speech Generation," 2025.
- [26] Yicheng Gu, et al., "SingNet: Towards a Large-Scale, Diverse, and In-the-Wild Singing Voice Dataset," *OpenReview*, 2024.
- [27] Yicheng Gu, et al., "Solid State Bus-Comp: A Large-Scale and Diverse Dataset for Dynamic Range Compressor Virtual Analog Modeling," 2025.
- [28] Zhiyan Duan, et al., "The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *APSIPA*, 2013.
- [29] Dawn AA Black, et al., "Automatic identification of emotional cues in Chinese opera singing," *ICMPC*, 2014.
- [30] Julia Wilkins, et al., "VocalSet: A Singing Voice Dataset," in *ISMIR*, 2018.
- [31] Soonbeom Choi, et al., "Children's song dataset for singing voice research," in *ISMIR*, 2020.
- [32] Junya Koguchi, et al., "PJS: phoneme-balanced Japanese singing-voice corpus," in *APSIPA*, 2020.
- [33] Bidisha Sharma, et al., "NHSS: A speech and singing parallel database," *Speech Commun.*, 2021.
- [34] Rongjie Huang, et al., "Multi-Singer: Fast Multi-Singer Singing Voice Vocoder With A Large-Scale Corpus," in *ACM MM*, 2021.
- [35] Itsuki Ogawa and Masanori Morise, "Tohoku kiritan singing database: A singing database for statistical parametric singing synthesis using japanese pop songs," *AST*, 2021.
- [36] Jiatong Shi, et al., "Muskits: an End-to-end Music Processing Toolkit for Singing Voice Synthesis," in *Interspeech*, 2022.
- [37] Jinglin Liu, et al., "DiffSinger: Singing Voice Synthesis via Shallow Diffusion Mechanism," in *AAAI*, 2022.
- [38] Lichao Zhang, et al., "M4Singer: A Multi-Style, Multi-Singer and Musical Score Provided Mandarin Singing Corpus," in *NeurIPS*, 2022.
- [39] Yu Wang, et al., "Openpop: A High-Quality Open Source Chinese Popular Song Corpus for Singing Voice Synthesis," in *Interspeech*, 2022.
- [40] Shuqi Dai, et al., "SingStyle111: A Multilingual Singing Dataset With Style Transfer," in *ISMIR*, 2023.
- [41] Meizhen Zheng, et al., "FT-GAN: Fine-Grained Tune Modeling for Chinese Opera Synthesis," in *AAAI*, 2024.
- [42] Jiatong Shi, et al., "Singing voice data scaling-up: An introduction to ace-openpop and kising-v2," *arXiv:2401.17619*, 2024.
- [43] Yu Zhang, et al., "Gtsinger: A global multi-technique singing corpus with realistic music scores for all singing tasks," *arXiv:2409.13832*, 2024.
- [44] Ilya Loshchilov and Frank Hutter, "Decoupled Weight Decay Regularization," in *ICLR*, 2019.
- [45] Wen-Chin Huang, et al., "The Singing Voice Conversion Challenge 2023," vol. arXiv:2306.14422, 2023.
- [46] Xueyao Zhang, et al., "Amphion: An Open-Source Audio, Music and Speech Generation Toolkit," in *SLT*, 2024.
- [47] Xueyao Zhang, et al., "Leveraging Content-based Features from Multiple Acoustic Models for Singing Voice Conversion," *CoRR*, vol. abs/2310.11160, 2023.
- [48] Wen-Chin Huang, et al., "MOS-Bench: Benchmarking Generalization Abilities of Subjective Speech Quality Assessment Models," *arXiv:2411.03715*, 2024.
- [49] Bernardo Torres, et al., "Singer Identity Representation Learning Using Self-Supervised Techniques," in *ISMIR*, 2023.