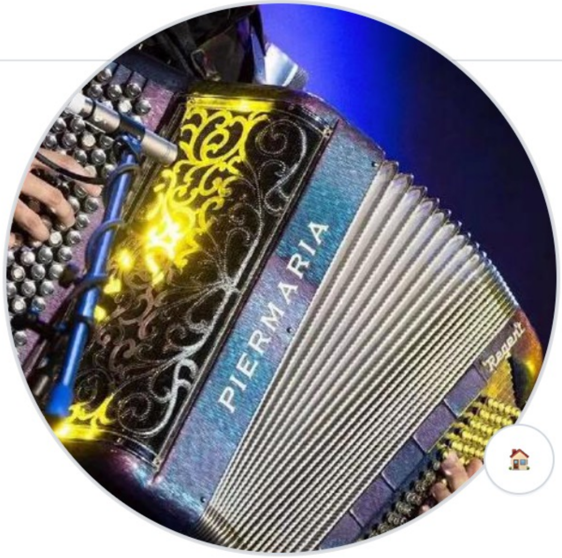
Singing Voice Synthesis: Data Curation, Modeling, and Evaluation

Jiatong Shi

jiatongs@cs.cmu.edu

About Me

- 4th Year Ph.D. Student
- Main research focus:
  - speech representation learning and its application
- Broad interests in many downstream tasks:
  - Typical speech tasks: ASR & TTS & ST & SLU
    - Architectures
    - Decoding
    - Aspects in low-resource and multilingual
  - Related music tasks
    - Singing voice synthesis
    - Singing voice conversion
    - Music generation
  - Recent focus:
    - Speech, music, and general audio evaluation



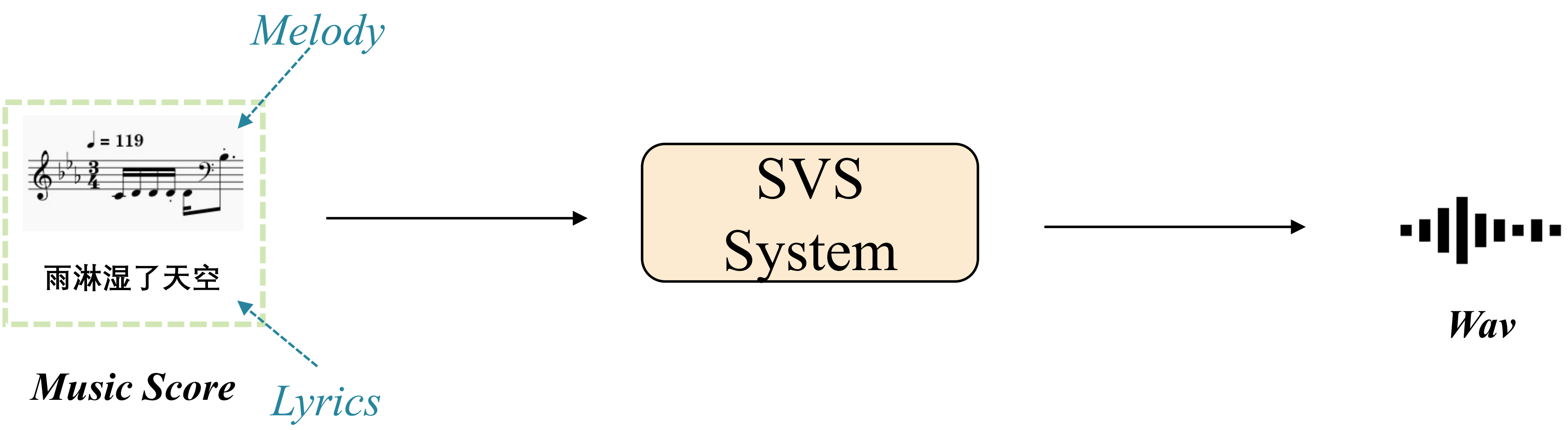
Jiatong
ftshijt

Content

- Singing Voice Synthesis (SVS), an Overview
- Data Curation for SVS
  - Introduction to ACE-Opencpop and ACE-KiSing (Interspeech 2024)
- Modeling for SVS (Discrete SVS)
  - TokSing
  - Multi-resolution discrete token learning: SingOMD
- Evaluation for SVS
  - SingMOS
  - VERSA

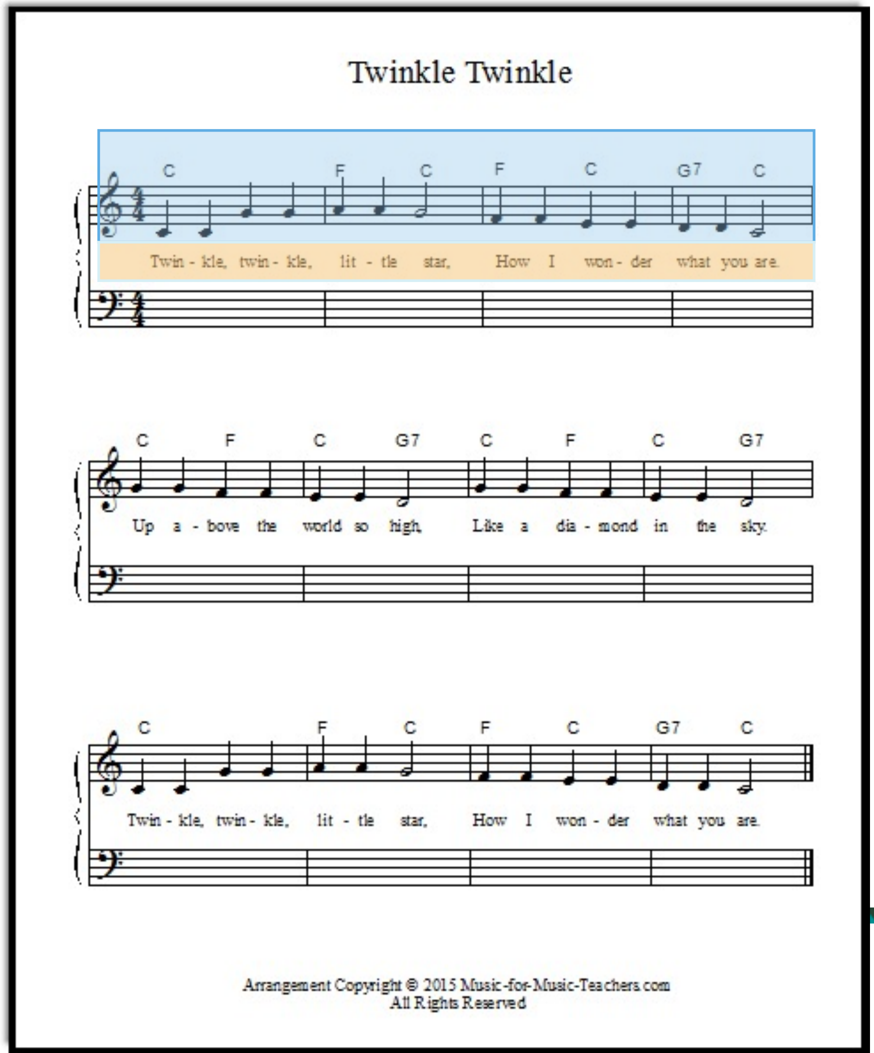
Singing Voice Synthesis (SVS)

- Utilize music score (i.e., melody and lyrics) to synthesize voice



The Format of Music Score

- Music Note
- Lyrics (in syllables?)
- Duration



Download from https://www.music-for-music-teachers.com/twinkle-twinkle.html (free for educational purpose)

Challenges in SVS

- Data Curation → More data!
- Modeling → Better modeling!
- Evaluation → Easier and comprehensive evaluation!

Data Curation

Singing Voice Data Scaling-up: An Introduction to ACE-Opencpop and ACE-KiSing

Jiatong Shi[1]*, Yueqian Lin[2], Xinyi Bai[3,4], Keyi Zhang, Yuning Wu[5], Yuxun Tang[6], Yifeng Yu[5], Qin Jin[5], Shinji Watanabe[1]

[1] Carnegie Mellon University, [2] Duke Kunshan University, [3] Cornell University, [4] Multimodal Art Projection Community, [5] Renmin University of China, [6] Georgia Institute of Technology

Motivation

- Compared to text-to-speech (TTS), SVS naturally has its difficulties in data collection:
  - **Stricter** usage guidelines and copyright concerns
  - Requires a professional, high-quality recording environment
  - Takes longer to record
  - Requires detailed annotations (e.g., music score, lyrics alignment)

Motivation

- Compared to text-to-speech (TTS), SVS naturally has its difficulties in data collection:
  - Stricter usage guidelines and copyright concerns
  - Requires a **professional, high-quality** recording environment
  - Takes longer to record
  - Requires detailed annotations (e.g., music score, lyrics alignment)

Motivation

- Compared to text-to-speech (TTS), SVS naturally has its difficulties in data collection:
  - Stricter usage guidelines and copyright concerns
  - Requires a professional, high-quality recording environment
  - Takes **longer** to record
  - Requires detailed annotations (e.g., music score, lyrics alignment)

Motivation

- Compared to text-to-speech (TTS), SVS naturally has its difficulties in data collection:
  - Stricter usage guidelines and copyright concerns
  - Requires a professional, high-quality recording environment
  - Takes longer to record
  - Requires **detailed** annotations (e.g., music score, lyrics alignment)

Real-world Production of Singing voice

- In real-world music production,
  - the singing voice is usually NOT used directly
  - but after **substantial efforts (i.e., mixing and mastering)** from the music producer.
- Widely-used procedures include:
  - Digital signal processing techniques
  - Parametric vocoders
  - Empirical strategies in audio smoothing, note correction, etc.
  - Voice control related to other sources (mixing)

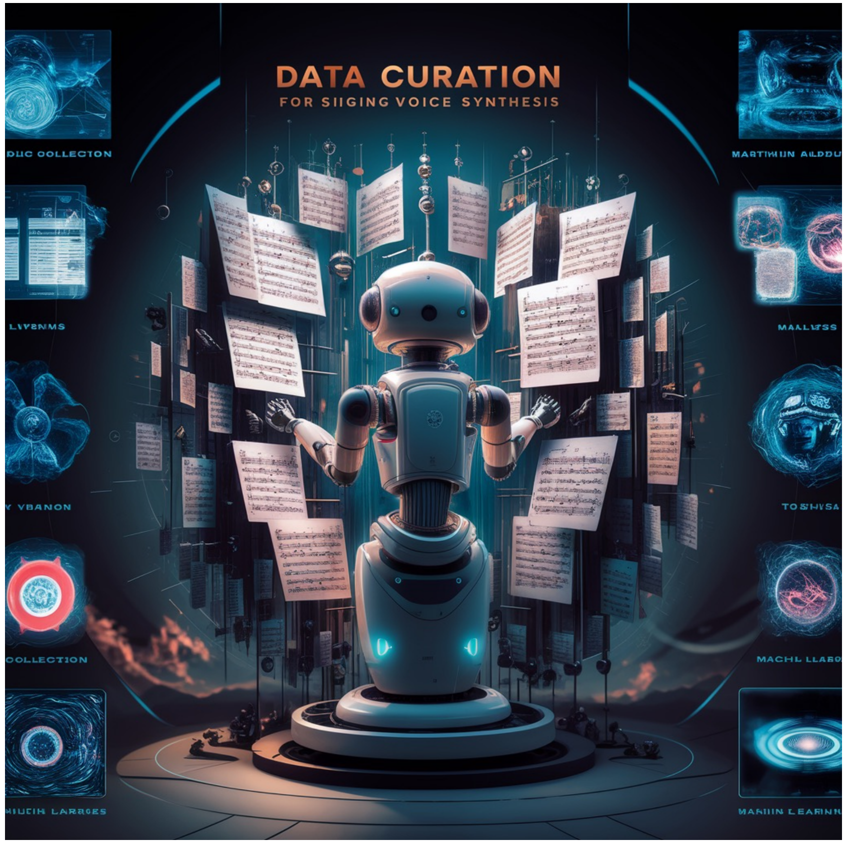
Q: Can we utilize **this** concept in production for better SVS data generation and preparation?

Key Contributions

- Introduce a unique data curation method for new SVS data
  - that incorporates a singing synthesizer and manual tuning
- Release two large-scale multi-singer SVS corpora:
  - ACE-Opencpop and ACE-KiSing
- Demonstrate three use-cases of the corpora, which shows to improve the performance of SVS modeling.

Corpora Details – Curation Process

- Data Preparation
- Information Verification and Correction
- Tuning for Voice Match
- Tuning for Singer Adaptation



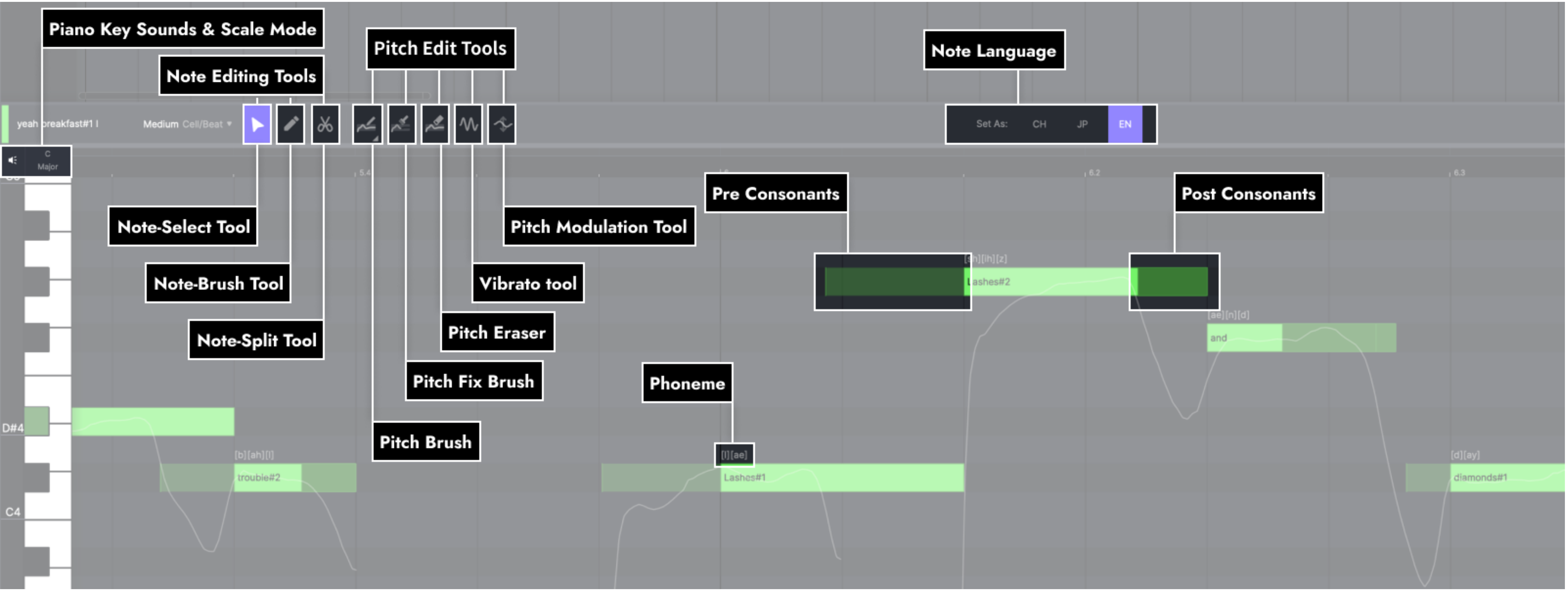
Corpora Details – Curation Process

- Data Preparation
  - Typical singing synthesis data preparation includes:
    - Musical notes, duration, syllable assignment, phone duration within a syllable
- Information Verification and Correction
- Tuning for Voice Match
- Tuning for Singer Adaptation

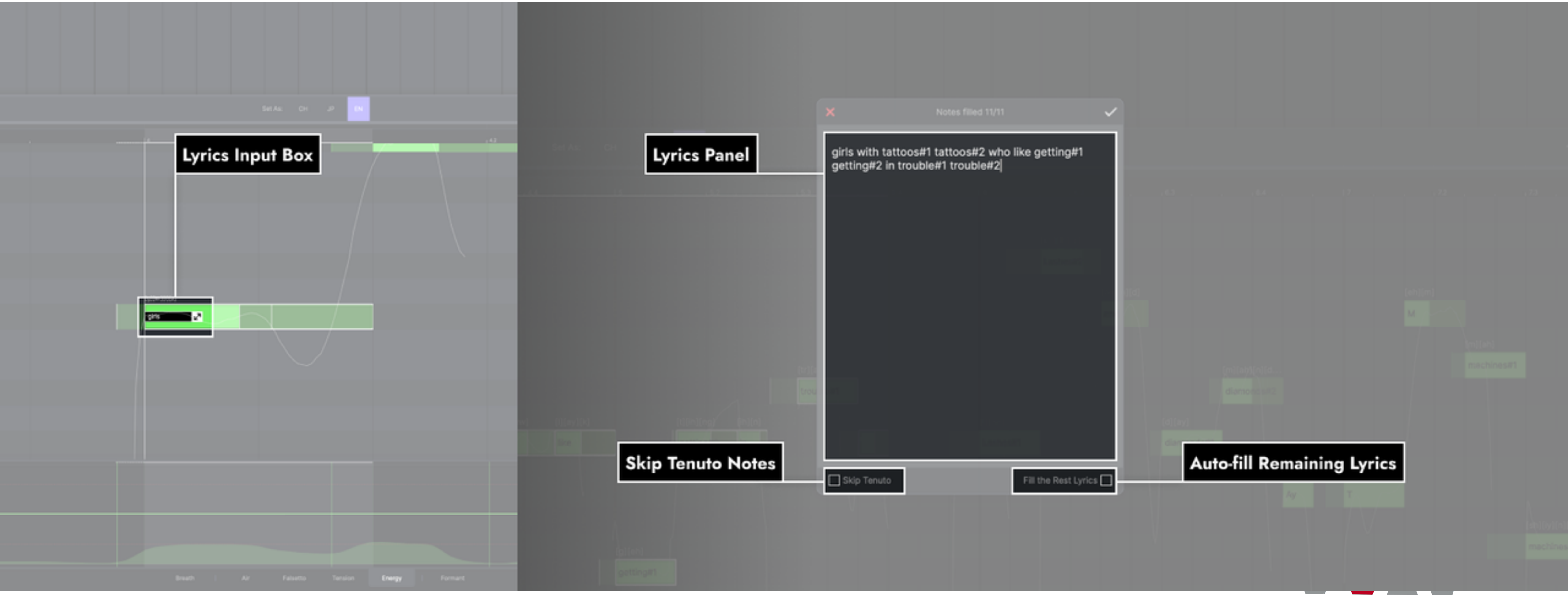
Corpora Details – Curation Process

- Data Preparation
- Information Verification and Correction
  - Error correction through ACE-Singer Interface
- Tuning for Voice Match
- Tuning for Singer Adaptation

Corpora Details – Curation Process – Music Note Editing

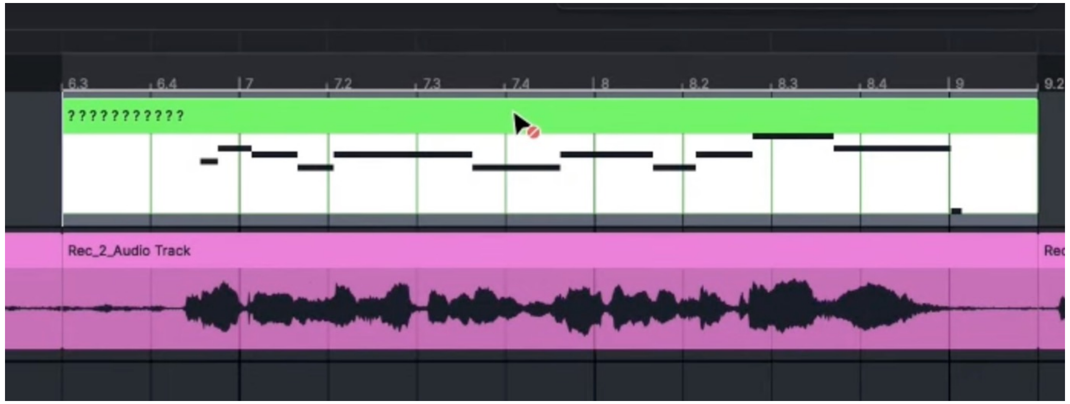


Corpora Details – Curation Process – Lyrics Editing

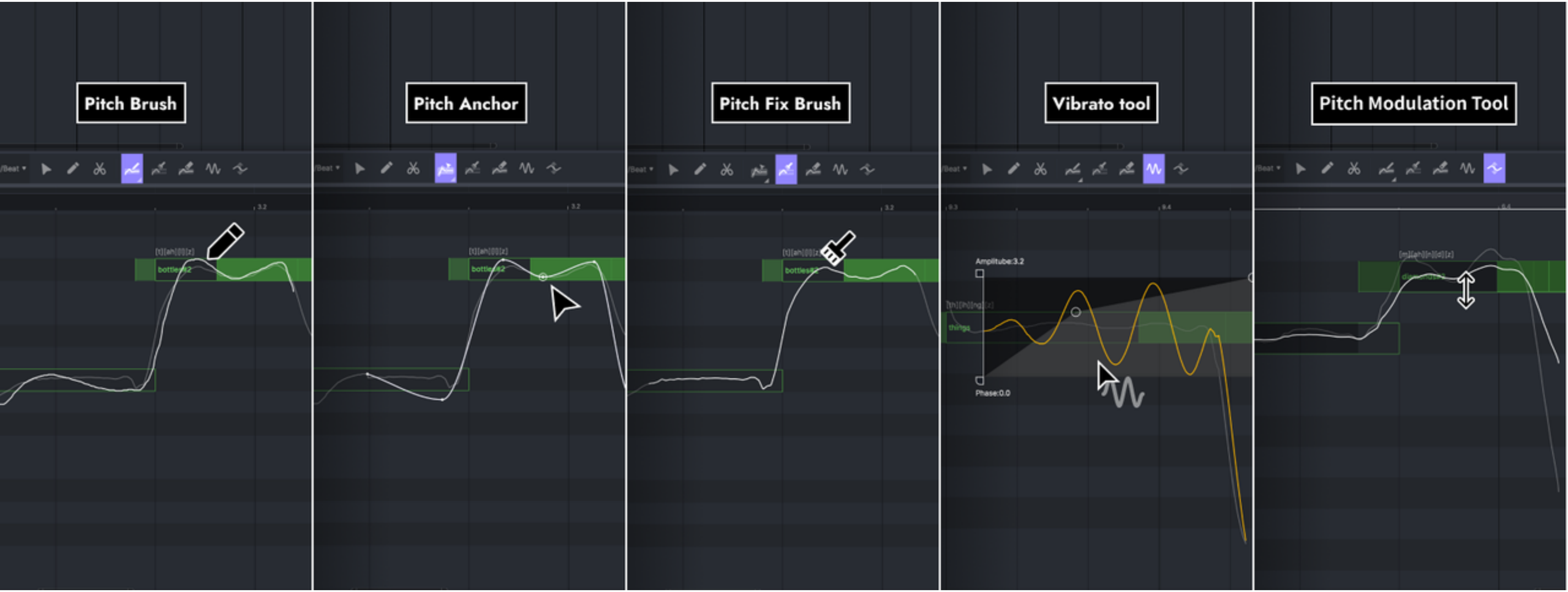


Corpora Details – Curation Process

- Data Preparation
- Information Verification and Correction
- Tuning for Voice Match
    - Iterative process to match the synthesized singing to the original singing
    - Key steps include:
        - F0 contour modification
        - Adding breath sounds
        - Adjusting the vibrato
        - Fine-tuning syllable duration
- Tuning for Singer Adaptation



Corpora Details – Curation Process – F0 Contour Editing



Corpora Details – Curation Process

- Data Preparation
- Information Verification and Correction
- Tuning for Voice Match
- Tuning for Singer Adaptation
  - Filtering unnatural singing phrases
    - → Not all singers can be proficient in all singing phrases!

Resulting Datasets

ACE-Opencpop

- Sourced from Opencpop*
- Directly uses the provided music score
- Common Chinese pop-style songs

Opencpop

* Wang, Y., Wang, X., Zhu, P., Wu, J., Li, H., Xue, H., Zhang, Y., Xie, L., Bi, M. (2022) Opencpop: A High-Quality Open Source Chinese Popular Song Corpus for Singing Voice Synthesis. Proc. Interspeech 2022, 4242-4246, doi: 10.21437/Interspeech.2022-48

Resulting Datasets (Cont'd)

ACE-KiSing

- Source from KiSing* but with additional new songs
- Re-transcribed the music scores
- Specialty: multi-genre, high melisma

* http://shijt.site/index.php/2021/05/16/kising-the-first-open-source-mandarin-singing-voice-synthesis-corpus/

Comparison to other multi-singer corpora

Dataset	Year	Language	Duration	Music Score	SVS	SVC	License
NUS-48E	2013	ENG	1.9	❌	❌	✅	Research-only
NHSS	2019	ENG	4.8	❌	❌	✅	Research-only
JVS-MuSiC	2020	JPN	2.3	❌	❌	✅	CC
OpenSinger	2021	CMN	50.0	❌	❌	✅	CC-NC
M4Singer	2022	CMN	29.8	✅	✅	✅	CC-NC
SingStyle111	2023	CMN/ENG/ITA	12.8	✅	❌	✅	Restricted by request
ACE-Opencpop	2024	CMN	**128.9**	✅	✅	✅	CC-NC
ACE-KiSing	2024	CMN/ENG	32.5	✅	✅	✅	CC-NC

Experiments

- Direct SVS
  - Directly training SVS with the corpora
- Transfer learning
  - Using pre-trained SVS model as initialization for other methods
- Joint training
  - Jointly using the data with other corpora to train multi-singer SVS

Experiments

- Direct SVS
  - ACE-KiSing
  - ACE-Opencpop
- Transfer learning
  - [In-domain transfer] ACE-Opencpop -> Opencpop
  - [Out-of-domain transfer] ACE-Opencpop -> Kiritan
- Joint training
  - ACE-KiSing + KiSing

Experiments

- Direct SVS
  - ACE-KiSing
  - ACE-Opencpop
- Transfer learning
  - [In-domain transfer] ACE-Opencpop -> Opencpop
  - [Out-of-domain transfer] ACE-Opencpop -> Kiritan
- Joint training
  - ACE-KiSing + KiSing

SVS Model

- Xiaoice-Sing2 (Wang et al. 2022)
- VISinger2 (Zhang et al. 2023)

Based on the open-source implementation in ESPnet-Muskits (Shi et al. 2022)

Experiments

Evaluation Metrics:

- Mel Cepstral Distortion (MCD)
- Semitone Accuracy (S. Acc)
- F0 Root Mean Square Error (F0 RMSE)
- Speaker Similarity (SECS)*
- Mean Opinion Score (MOS) with 95% Confidence Interval.

- Direct SVS
  - ACE-KiSing
  - ACE-Opencpop
- Transfer learning
  - [In-domain transfer] ACE-Opencpop -> Opencpop
  - [Out-of-domain transfer] ACE-Opencpop -> Kiritan
- Joint training
  - ACE-KiSing + KiSing

*Powered by ESPnet-SPK Rawnet-based speaker embedding.

Direct SVS (ACE-KiSing)

Model	MCD	S. Acc.	F0 RMSE	SECS	MOS
Xiaoice	6.10	62.93	0.199	0.77	3.29 ± 0.06
VISinger2	**5.24**	**64.50**	**0.185**	**0.80**	3.64 ± 0.06
G.T.	-	-	-	-	4.49 ± 0.05
Source G.T.	-	-	-	-	**4.51** ± 0.07

G.T. is the test set prepared in ACE-Opencpop.
Source G.T. is the test set in the original Opencpop dataset.

Direct SVS (ACE-KiSing)

Model	MCD	S. Acc.	F0 RMSE	SECS	MOS
Xiaoice	6.10	62.93	0.199	0.77	3.29 ± 0.06
VISinger2	**5.24**	**64.50**	**0.185**	**0.80**	3.64 ± 0.06
G.T.	-	-	-	-	4.49 ± 0.05
Source G.T.	-	-	-	-	**4.51** ± 0.07

Comparing G.T. and Source G.T., there is still a minor **gap** in MOS quality after the manual tuning.

Direct SVS (ACE-KiSing)

Model	MCD	S. Acc.	F0 RMSE	SECS	MOS
Xiaoice	6.10	62.93	0.199	0.77	3.29 ± 0.06
VISinger2	**5.24**	**64.50**	**0.185**	**0.80**	3.64 ± 0.06
G.T.	-	-	-	-	4.49 ± 0.05
Source G.T.	-	-	-	-	**4.51** ± 0.07

However, we also observe that with recently proposed singing synthesizers, the G.T., though also artificial, shows **a much better MOS score** (indicating better quality).

Direct SVS (ACE-Opencpop)

Model	MCD	S. Acc.	F0 RMSE	SECS	MOS
Xiaoice	5.81	64.33	0.162	0.75	3.53 ± 0.05
VISinger2	**5.08**	**65.19**	**0.140**	**0.78**	3.81 ± 0.05
G.T.	-	-	-	-	4.35 ± 0.06
Source G.T.	-	-	-	-	**4.69** ± 0.06

We can observe **similar findings** with ACE-Opencpop as with ACE-KiSing.

Transfer Learning (In-domain)

Model	MCD	S. Acc.	F0 RMSE	MOS
Xiaoice	9.26	59.22	0.185	2.78 ± 0.03
Xiaoice*	8.78	60.72	0.182	3.08 ± 0.05
ViSinger2	7.54	64.50	0.172	3.63 ± 0.07
VISinger2*	**7.26**	**65.18**	**0.162**	**3.66** ± 0.06
G.T.	-	-	-	4.69 ± 0.06

For in-domain scenarios, pre-training on ACE-Opencpop could **benefit** both Xiaoice and VISinger2 models.

Transfer Learning (Out-of-domain)

Model	MCD	S. Acc.	F0 RMSE	MOS
ViSinger2	**7.47**	49.33	0.123	3.58 ± 0.07
VISinger2*	7.54	**50.88**	**0.122**	**3.68** ± 0.07
G.T.	-	-	-	4.57 ± 0.07

In out-of-domain scenarios (cross-lingual + cross-singing styles), we also observe subjective improvements.

Joint-training as a multi-singer augmentation (ACE-KiSing + KiSing)

Model	MCD	S. Acc.	F0 RMSE	MOS
ViSinger2	5.55	68.12	0.168	3.58 ± 0.07
VISinger2*	**4.99**	**72.51**	**0.170**	**3.68** ± 0.07
G.T.				4.57 ± 0.07

We also observe significant improvements when using the data together with KiSing (a multi-style 1-hour dataset + melisma → more challenging).

Take-Home Message

ACE-Opencpop

ACE-KiSing

- We release two large-scale singing synthesis corpora, ACE-Opencpop and ACE-KiSing, with three common use cases:
  - Direct SVS system training
  - Transfer learning
  - Joint-training
- We showcase the use of manual tuning in dataset curation, which could be a feasible way to expand the dataset into multi-singer/multi-domain corpora.

Acknowledgement

We would like to specially thank Shengyuan Xu (Timedomain) and Pengcheng Zhu (Netease) for their support in the data license)

The experiments of the work used Bridges2 system at PSC and Delta system at NCSA.

- Part of the images are generated with Dall-E or Ideogram for research and educational presentation purpose only.
- We utilize some ACE-Studio manual images to demonstrate the curation process.

Modeling

A New Trend in Signal Generation

- Going discrete!

Transitioning to Discrete Representation

- Benefits from discrete representation

- Scaling up (e.g., TTS with VALL-E (Wang et al. 2023))

	Conventional TTS Systems	**VALL-E**
Intermediate Representation	Continuous spectral representation	Audio codec code
Objective Function	Continuous Regression	Language Model
Training Data	< 600 hours	60k hours
In-context Learning	❌	✅

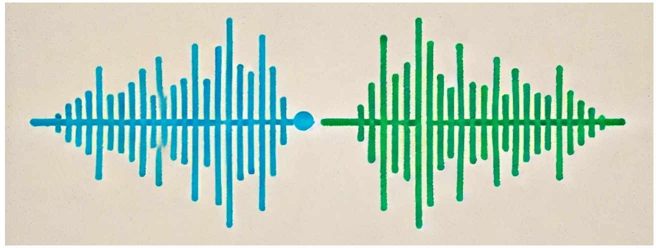
Transitioning to Discrete Representation

- Benefits from discrete representation

- Improved storage efficiency (e.g., discrete ASR (Chang et al. 2023))

Data Format	**Data size (bits)**
Raw waveform	16 x 16000 x T
Acoustic Features	32 x D x 100 x T
Self-supervised learning representation	32 x 1024 x 50 x T
Discrete tokens	**12 x 50 x T**

Transitioning to Discrete Representation

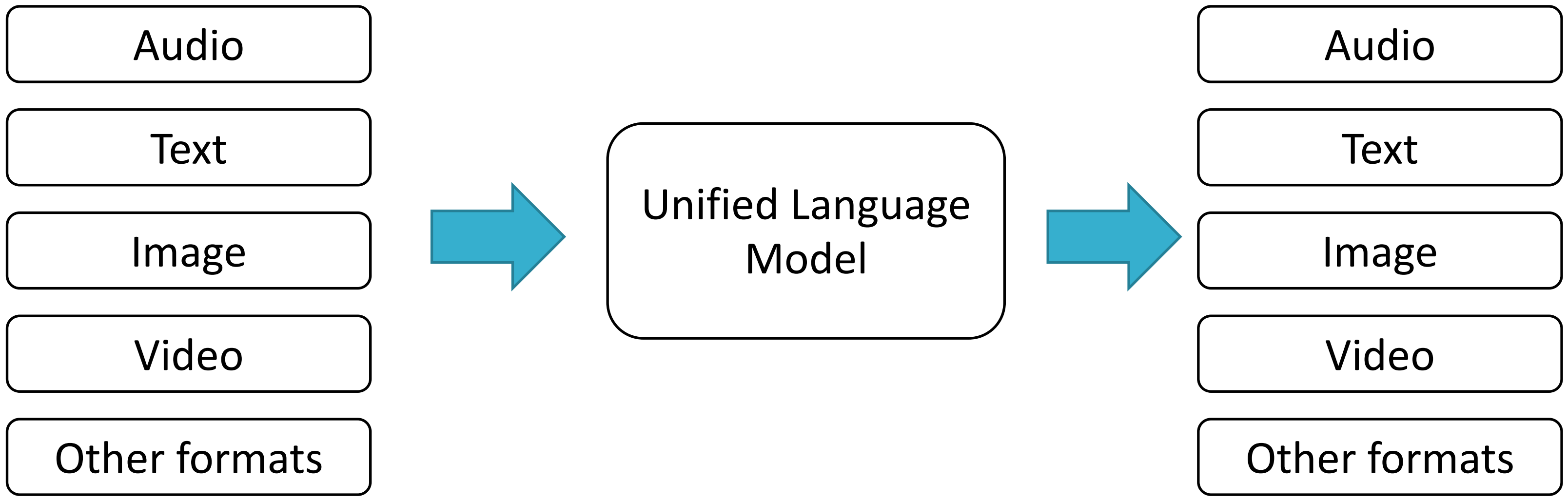


- Benefits from discrete representation
- Enhanced efficiency in computation (e.g., speech enhancement with CodecFormer (Yip et al. 2024))

Model	GMACs	Training Time (h/epoch)
SepFormer	77.3	2.7
CodecFormer	**1.5**	**1.0**

Transitioning to Discrete Representation

- Benefits from discrete representation
- Potential for integration with various modalities (e.g., Multimodal LLM with AnyGPT (Zhang et al. 2024))



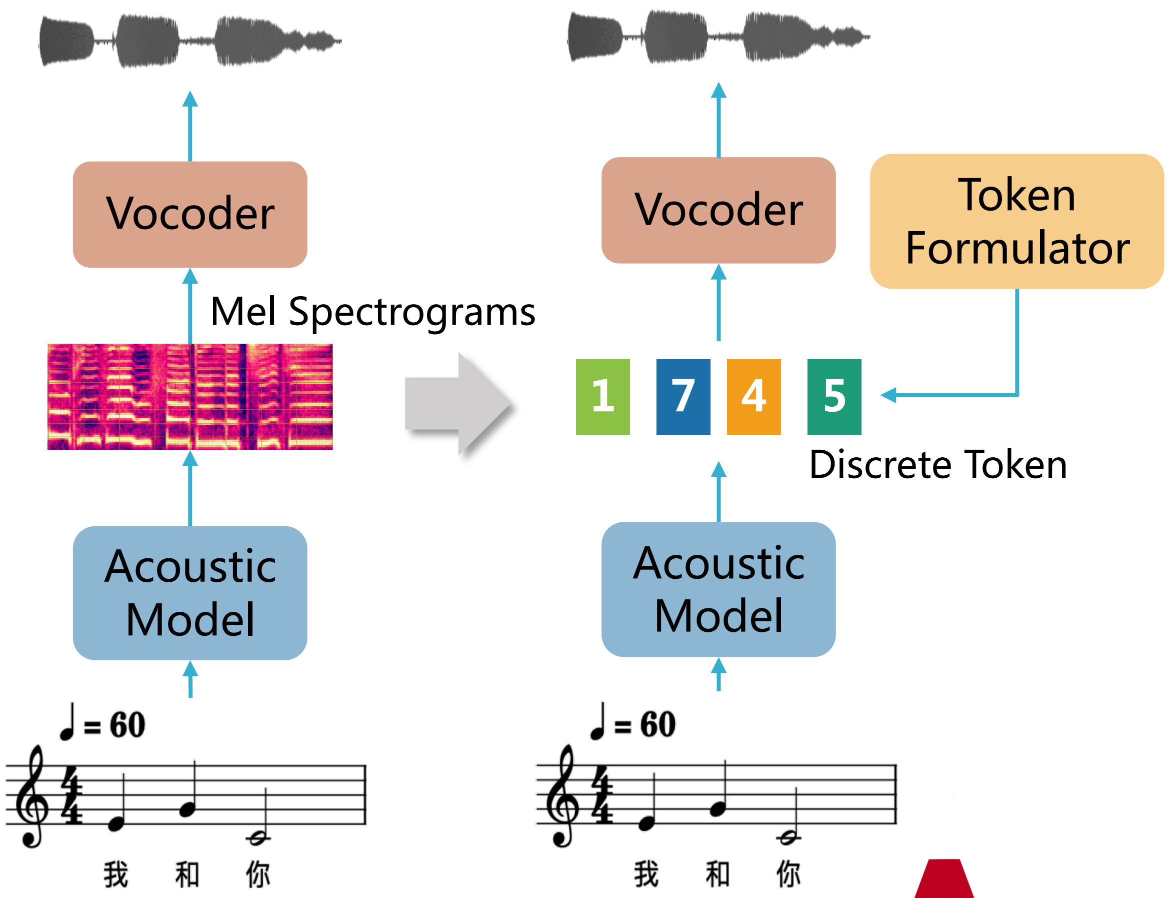
TokSing: Singing Voice Synthesis based on Discrete Tokens

Yuning Wu[1] , Chunlei Zhang[2], Jiatong Shi[3], Yuxun Tang[1], Yang Shan[2], Qin Jin[1]

[1] Renmin University of China, [2] Tencent, [3] Carnegie Mellon University

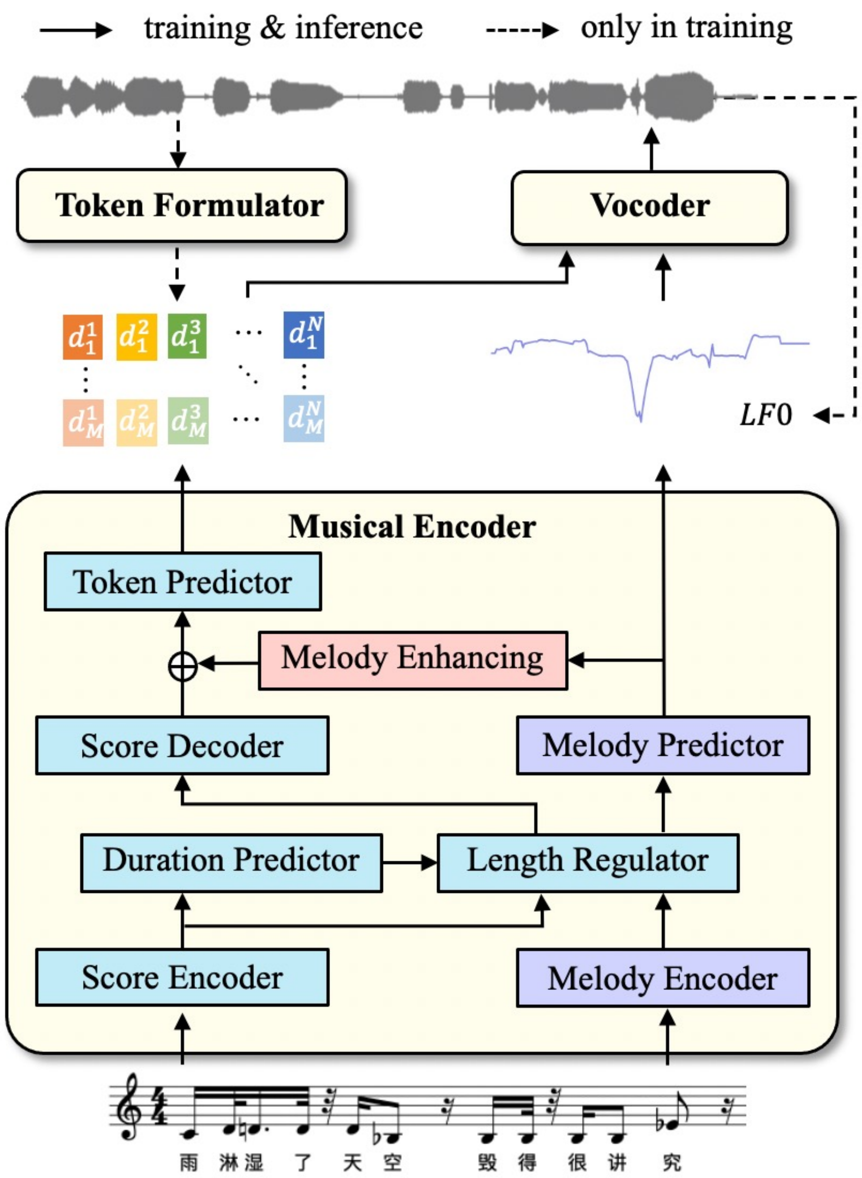
Challenges in Discrete SVS

- <u>Acoustic details</u>' loss during discretization
- High demand in <u>melody</u>



TokSing Framework

- **<u>Where</u>** tokens extracted from
- **<u>How</u>** tokens are formulated



Vocoder (token2wav)

↑ construction quality

Musical Encoder (score2token)

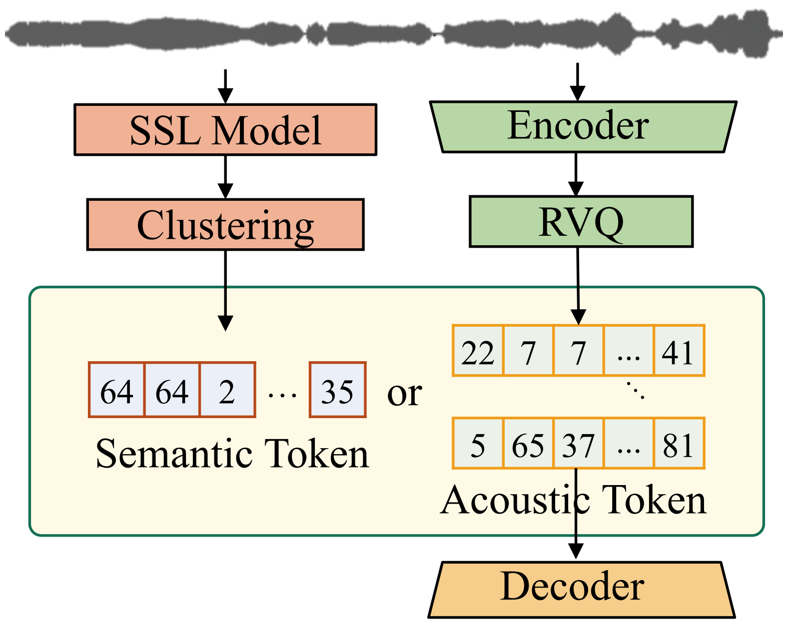
↑ acoustic prediction

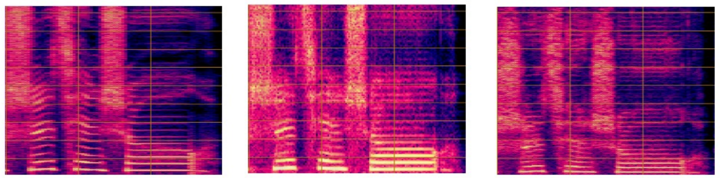
Token Formulation

- ➢ **<u>Where</u>** tokens extracted from
- ➢ <u>How</u> tokens are formulated
    - Semantic token
    - Acoustic token

💡 *For a better construction quality, we choose <u>semantic tokens</u> and train a <u>discrete-based vocoder</u>.*

- For more implementation of codec, see our latest work in ESPnet-Codec (Shi et al. 2024)





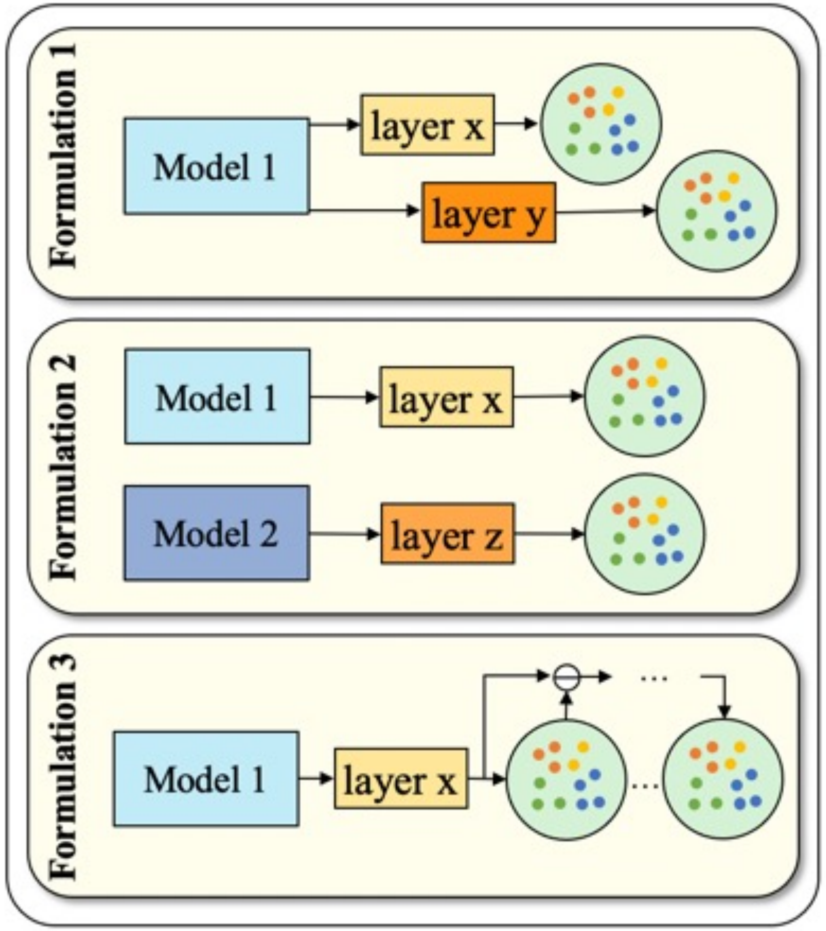
(d) GT (e) EnCodec (non-causal) (f) EnCodec (causal)

Token Formulation (Cont'd)

- ➢ **<u>Where</u>** tokens extracted from
- ➢ <u>How</u> tokens are formulated

  - single-layer
  - **multi-layer (MMM-based framework)**

  - *For more information of the MMM-based framework, see our concurrent work in MMM (Shi et al. 2024)*



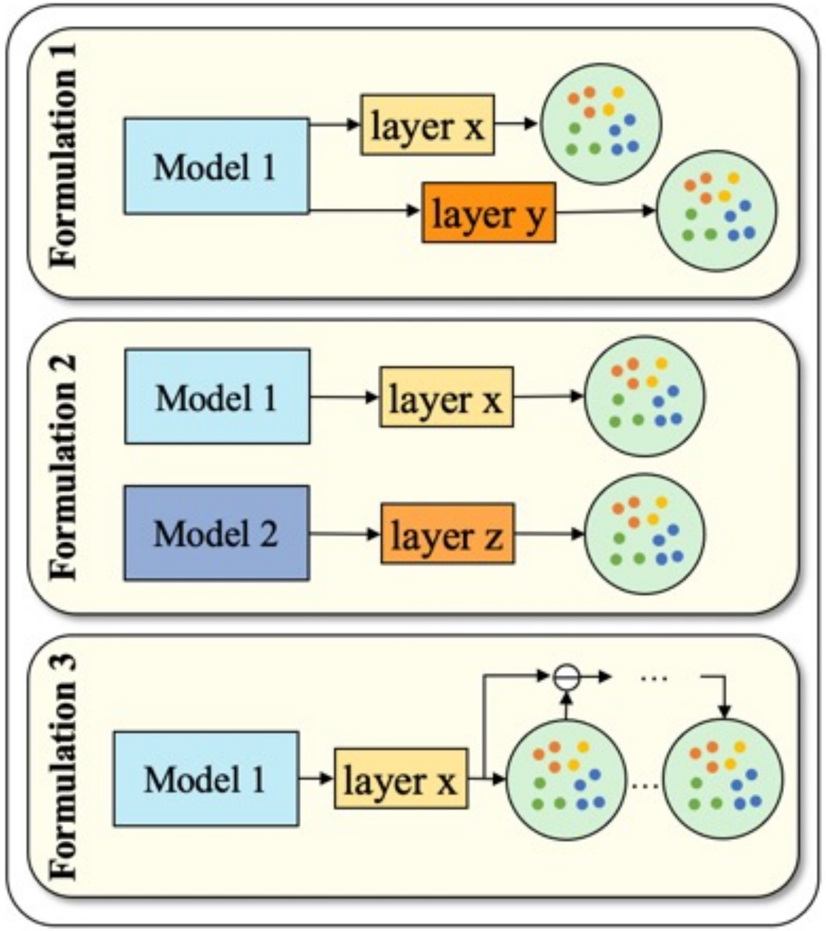
prediction accuracy improves on all layers

decrease when too deep

Token Formulation (Cont'd)

- ➢ **<u>Where</u>** tokens extracted from
- ➢ <u>How</u> tokens are formulated

  - single-layer
  - **multi-layer (MMM-based framework)**



prediction accuracy improves on all layers

decrease when too deep

💡 *<u>Which model and which layer</u> ?*

verify through weighted sum on vocoder

→ the 6th and 23rd layers of SSL models (HuBERT, WavLM and etc.)

Token Formulation - Abalation

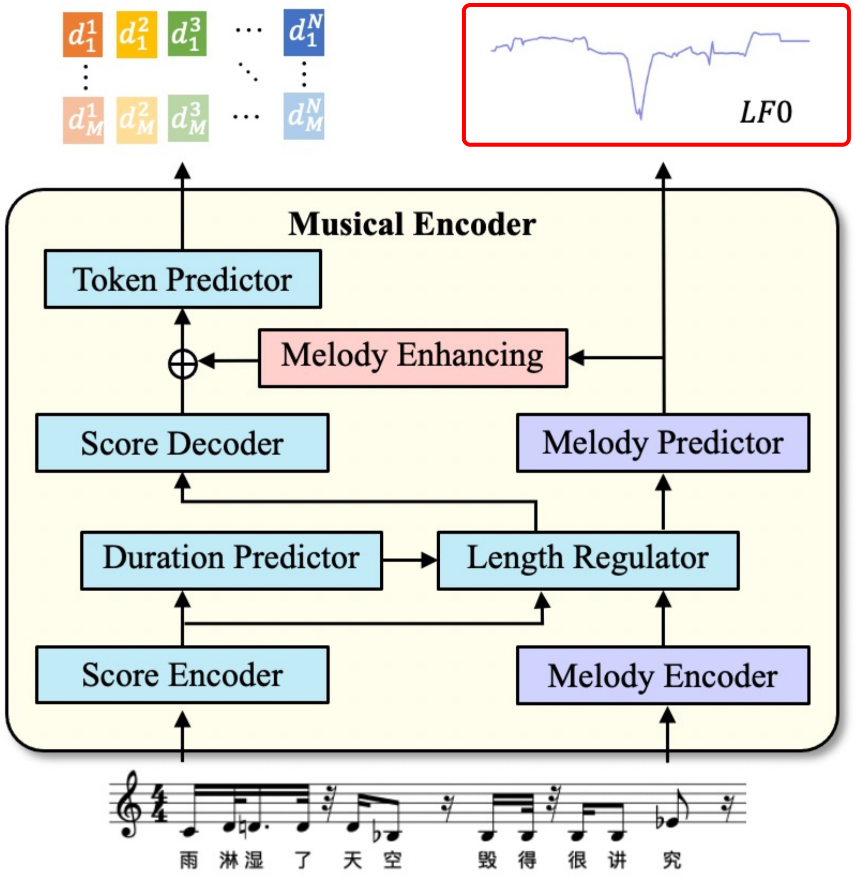
Representaion	Vocoder MCD ↓	Vocoder F0 ↓	+ Acoustic MCD ↓	+ Acoustic F0 ↓	+ Acoustic MOS ↑
single	6.59	0.17	7.56	**0.17**	3.70
Formulation 1	6.51	0.17	7.59	0.18	**3.74**
Formulation 2	6.39	0.16	**7.50**	**0.17**	3.61
Formulation 3	**5.96**	**0.15**	7.65	0.18	3.40

Experiments on 5.2hrs single singer:
Opencpop (Wang et al. 2022)

- Effective in both acoustic model and vocoder.
- Can be utilized individually or combined strategically.

F0 here refers to F0 root mean square error (supported by ESPnet-TTS)

Musical Encoder – Enhance the Melody

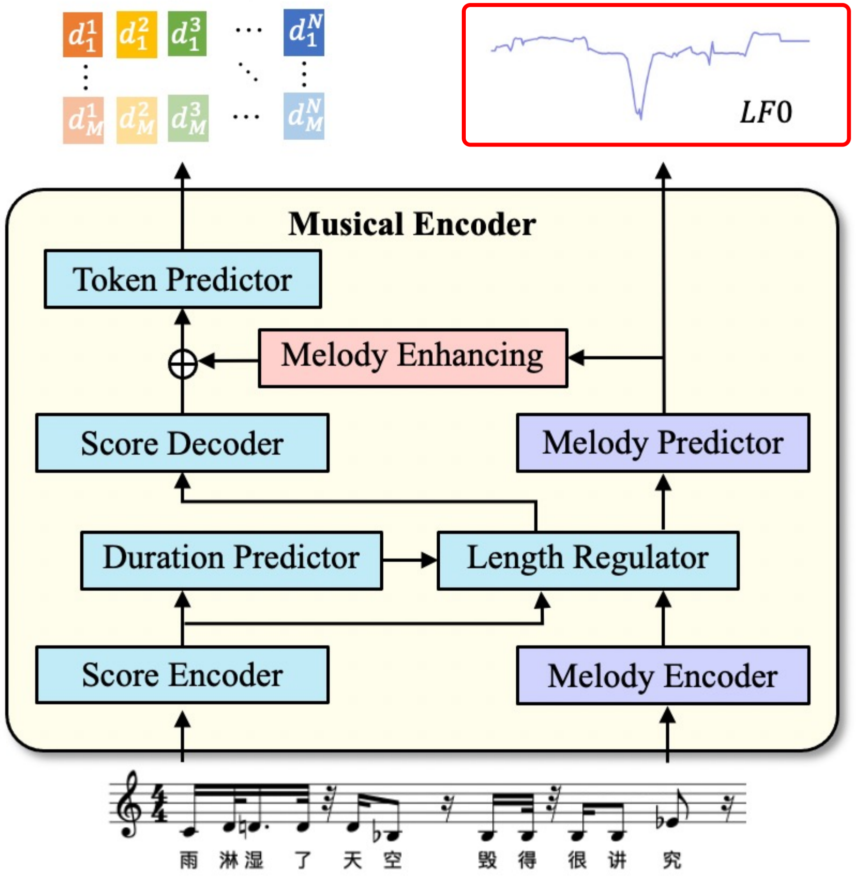


- To alleviate melody degradation during discretization

→ include a melody signal ($logF0$)

Representation	Vocoder MCD ↓	Vocoder F0 ↓	+ Acoustic MCD ↓	+ Acoustic F0 ↓
SSL Feat.	**3.18**	**0.14**	-	-
token only	9.10	0.27	9.60	0.26
token **+** $LF0$	6.59	0.17	**7.56**	**0.17**
Codec token + $LF0$	7.56	0.17	-	-
Codec decoder	6.35	0.24	-	-
Mel spectrogram	3.32	0.15	8.00	0.19

Musical Encoder – Enhance the Melody



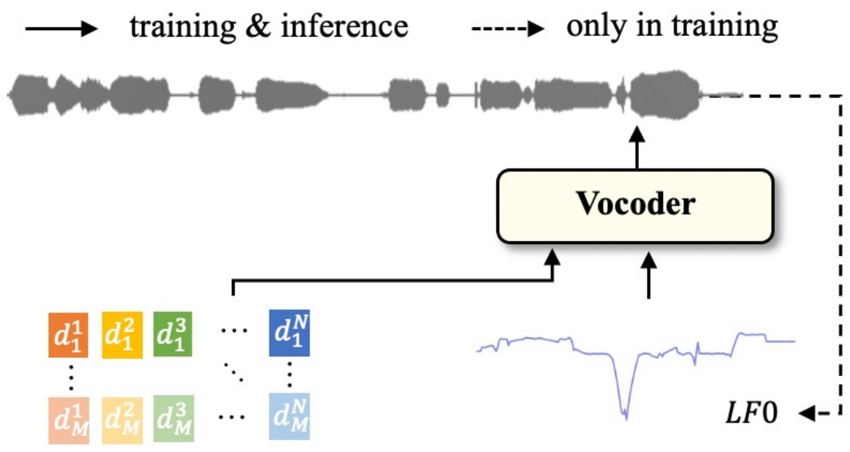
- Tokens encapsulate certain pitch-related details

→ integrate predicted melody into token prediction

Melody Prediction	Melody Enhanced	Objective MCD ↓	F0 ↓	SA ↑	Subjective Melody ↑	MOS ↑
✗	✗	8.07	0.19	44%	2.07	3.18
✗	✓	7.61	0.21	59%	2.32	3.65
✓	✓	**7.56**	**0.17**	**61%**	**2.38**	**3.70**

SA refers to semitone accuracy.

Vocoder Enhancement



- Vocoder becomes the new bottleneck

Representation	Vocoder MCD ↓	Vocoder F0 ↓	+ Acoustic MCD ↓	+ Acoustic F0 ↓
SSL Feat.	**3.18**	**0.14**	-	-
token only	9.10	0.27	9.60	0.26
token **+** $LF0$	6.59	0.17	**7.56**	**0.17**
Codec token + $LF0$	7.56	0.17	-	-
Codec decoder	6.35	0.24	-	-
Mel spectrogram	3.32	0.15	8.00	0.19

- **Transfer learning**

vocoder pretrained on 150h multi-speaker dataset

Representation	Vocoder MCD ↓	Vocoder F0 ↓	+ Acoustic MCD ↓	+ Acoustic F0 ↓
ACE-Opencpop	**5.60**	**0.14**	-	-
Opencpop-origin	6.51	0.16	8.15	0.19
Opencpop-transfer	6.11	0.16	**7.43**	**0.17**

Benefits of Going Discrete?

Subjective evaluation from different angles.

Evaluations between SVS systems

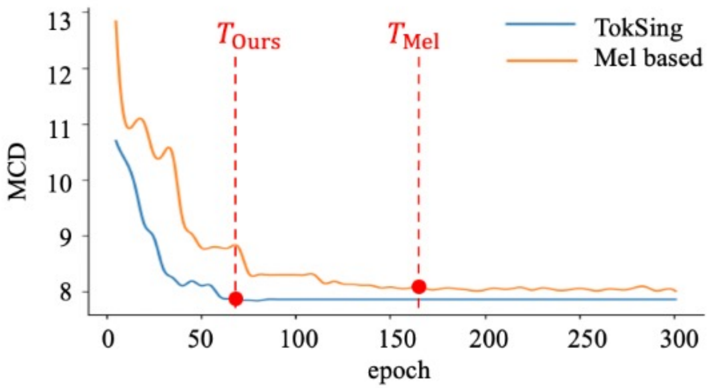
Representation	Objective Evaluations MCD ↓	F0 ↓	SA ↑	Subjective Evaluations Pron ↑	Melody ↑	Tech ↑	MOS ↑	Bitrate/bps
Mel spectrogram	8.00	0.19	59%	2.59	2.23	2.20	3.42	204800
Latent variance	7.76	0.18	**62%**	2.74	2.34	**2.37**	3.68	491520
Discrete token	**7.56**	**0.17**	61%	**2.73**	**2.38**	**2.37**	**3.70**	**1950**
GT	-	-	-	2.93	2.83	2.82	4.59	-

→ lower bitrate

Better objective and subjective performance with the discrete token!

Benefits of Going Discrete?

Convergence speed



→ higher convergence speed

Mel cepstral distortion measure in the y-axis

Some Examples

GT	baseline	TokSing	Lyric	
			“能不能给我一首歌的时间”	*Could you give me the time of one song?*
			“静静地把那拥抱变成永远”	*Turn the embrace into forever*
			“小酒窝长睫毛”	*Little dimples and long eyelashes*

Baseline here is the model that directly use the discrete units from speech pre-trained models

Take Home Messages

- ✓ Propose a discrete based SVS framework, TokSing, with melody enhancement by integrating melody control signals and improving acoustic prediction.
- ✓ Introduce a token formulator and provide multiple token formulations, offering flexibility in token sourcing and blending.
- ✓ Achieves better performance with lower storage cost and higher convergence speed than Mel systems.

Training code and pre-trained models are currently in ESPnet Pull Request!

SingOMD: Singing Oriented Multi-resolution Discrete Representation Construction from Speech Models

Yuxun Tang[1], Yuning Wu[1] , Jiatong Shi[2], Qin Jin[1]

[1] Renmin University of China, [2] Carnegie Mellon University

Improve Discrete Units for Singing?

- TokSing presents a practical solution to use existing speech models
  - Discrete units are extracted from pre-trained self-supervised speech models
- Can we acquire discrete representations for singing?

Potential Issues in Discrete SVS

- No existing self-supervised learning (SSL) representations for singing
  - Various reasons, but mostly with data (mentioned earlier):
    - Strict copyright issue of singing data
    - Constraints in scaling of singing data

Potential Issues in Discrete SVS

- Domain mismatch (singing vs. speech)
  - Nuanced pitch variations
  - Broader spectrum of vocal ranges
  - Flexible duration
- Fixed resolution (i.e., 20ms)
  - Suboptimal?
  - Insights from (Multiresolution HuBERT, Shi et al. 2024)

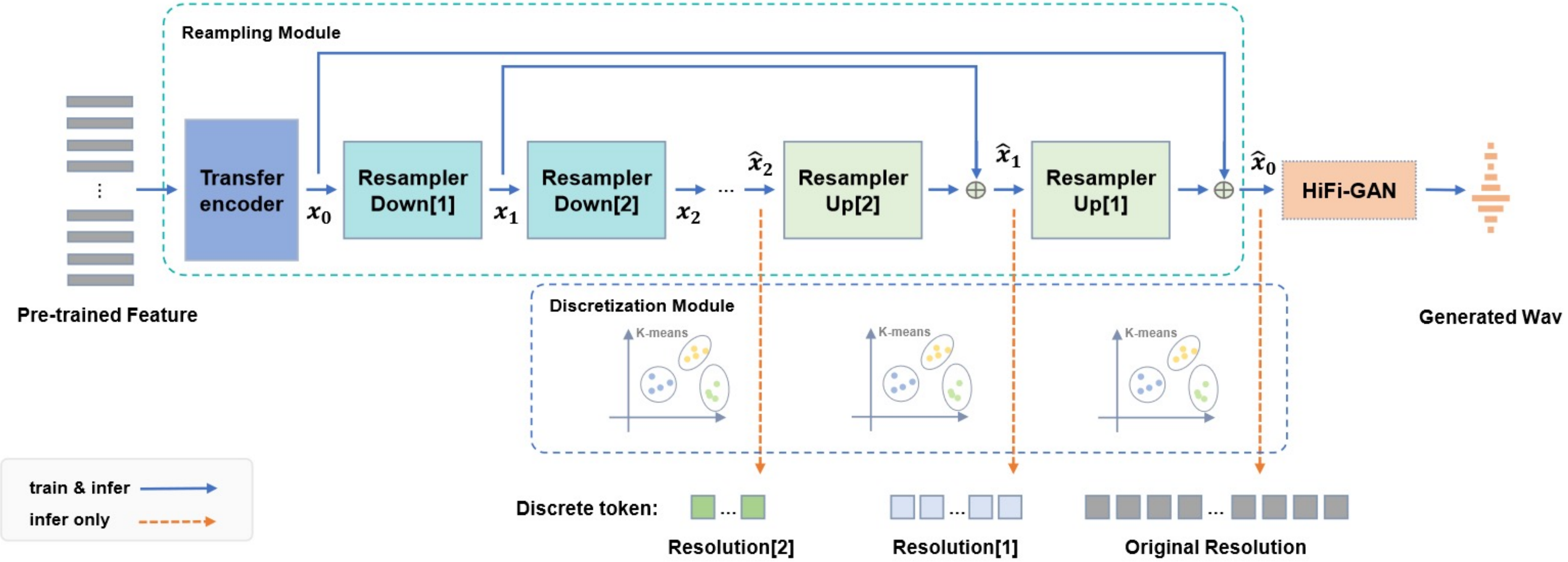
Potential Issues in Discrete SVS

- No existing SSL for singing
- Domain mismatch (singing vs. speech)
- Fixed resolution (i.e., 20ms)

A new framework, namely **SingOMD** to create new discrete tokens for singing

Overview:

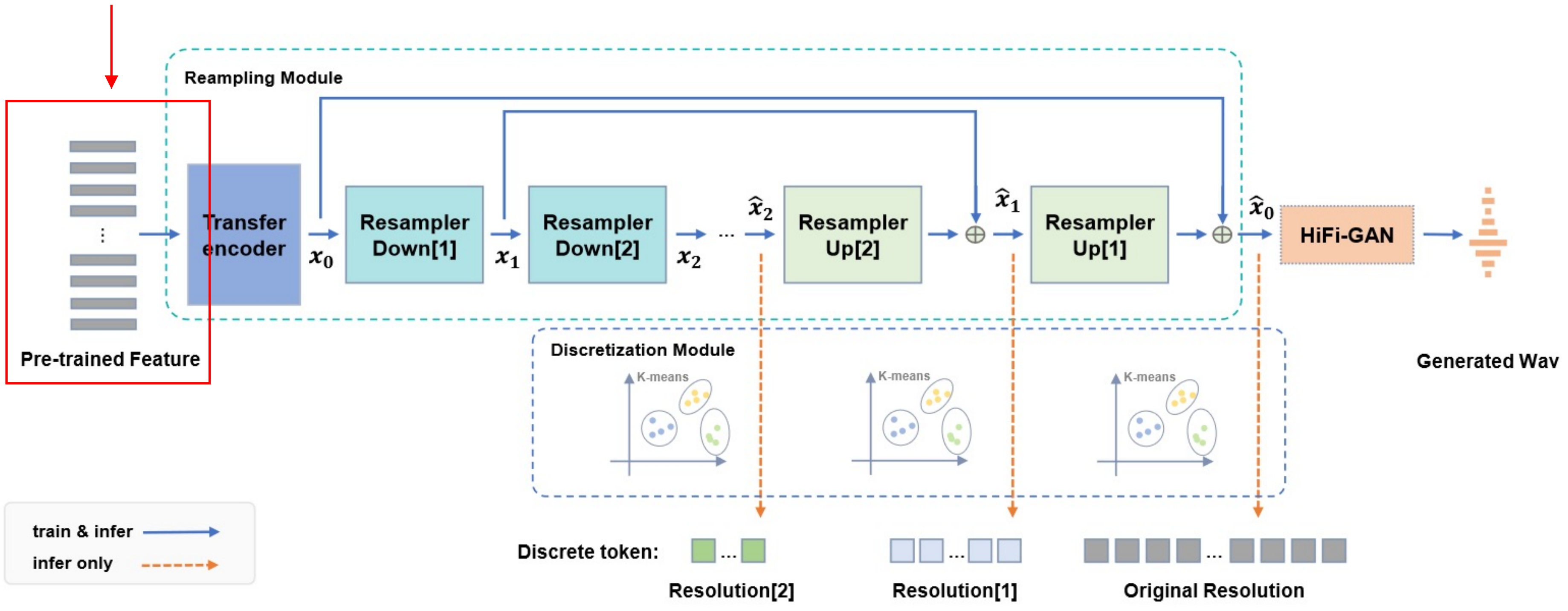
$$s = \mathrm{SSL}(y)$$
$$\hat{x} = \mathrm{Resampling}(s)$$
$$\hat{y} = \mathrm{Vocoder}(\hat{x})$$



Step 1: extract features from SSL models

$$s = \mathrm{SSL}(y)$$

Extracted from frozen pretrained speech SSL models



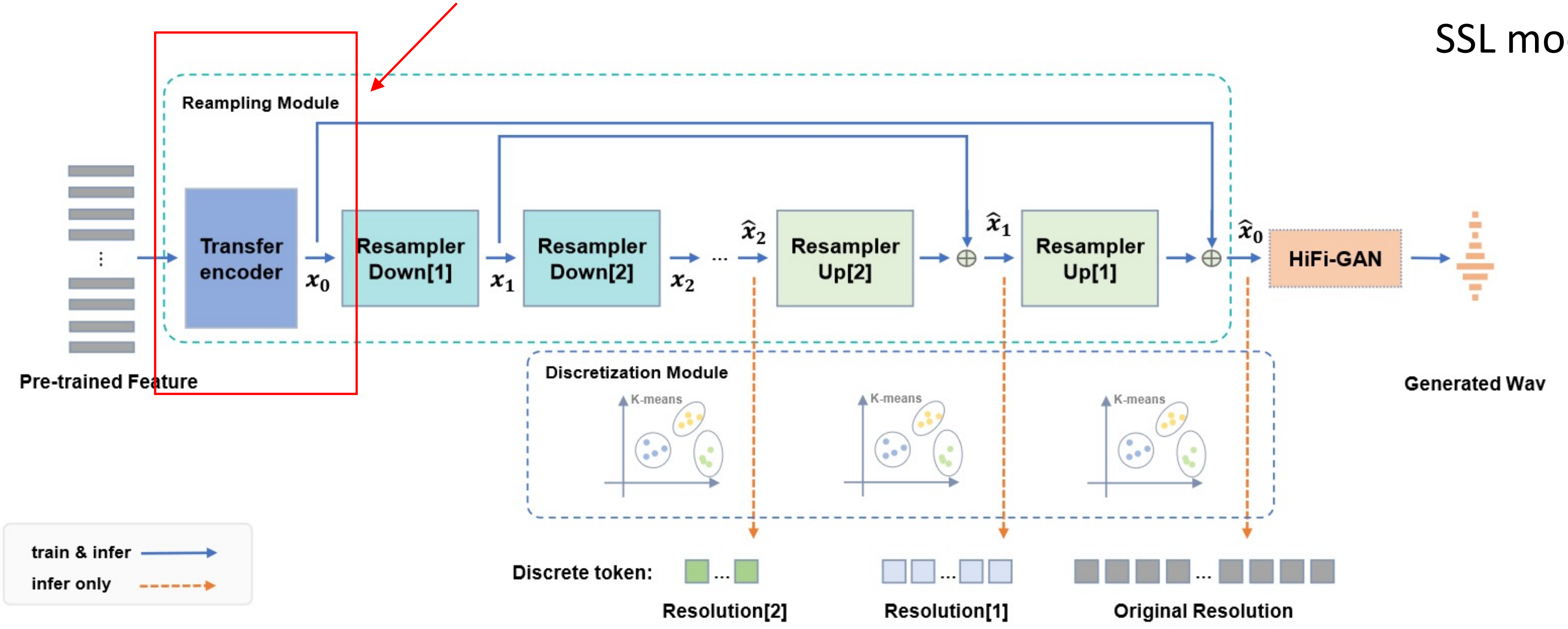
Deal with the problem:

- No SSL models for singing

Step 2: transfer and resample features

$$\hat{\boldsymbol{x}} = \text{Resampling}\left(\boldsymbol{s}\right)$$

Transfer speech features to singing features



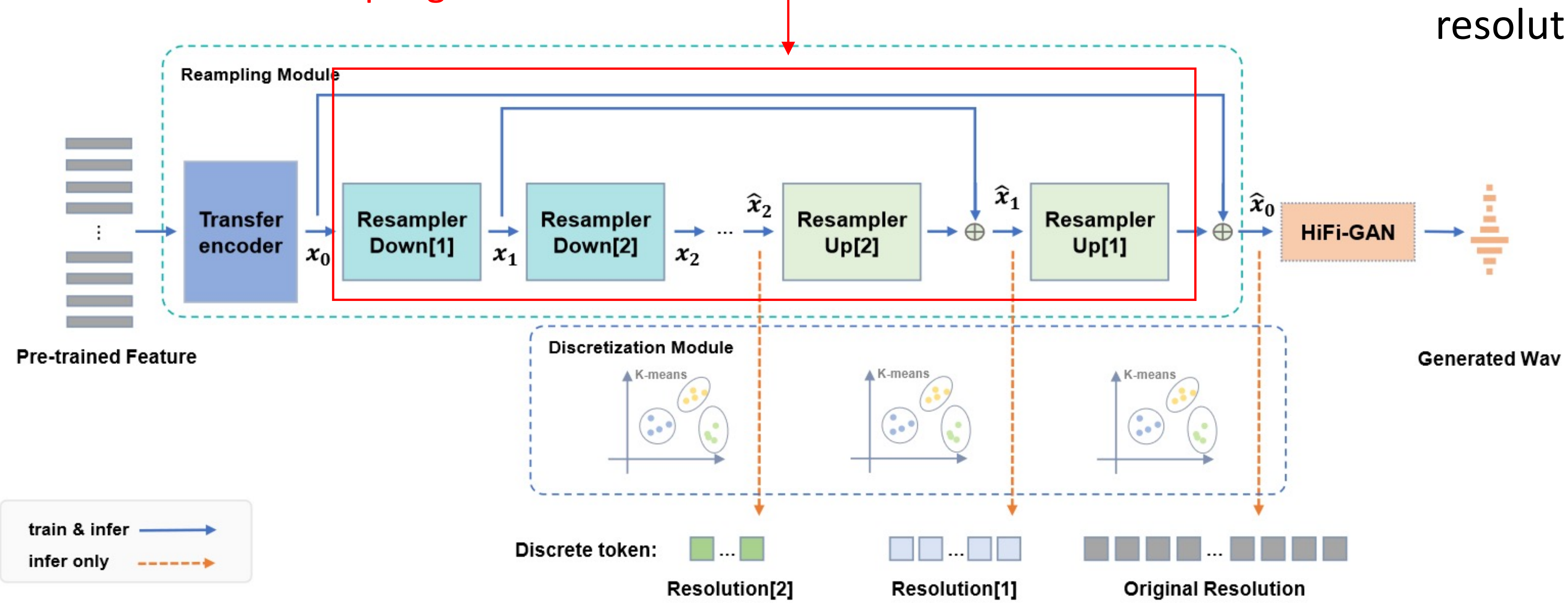
Deal with the problem:

- Domain gap between speech and singing when using speech SSL models directly

Step 2: transfer and resample features

$$\hat{\boldsymbol{x}} = \text{Resampling}\,(\boldsymbol{s})$$

Incorporate multi resolution features in a Unet-based resampling module



Deal with the problem:

- Suboptimal performance for representations in a fixed resolution

Step 3: resynthesize waveform

$$\hat{\boldsymbol{y}} = \text{Vocoder}\ (\hat{\boldsymbol{x}})$$

Resynthesize waveform

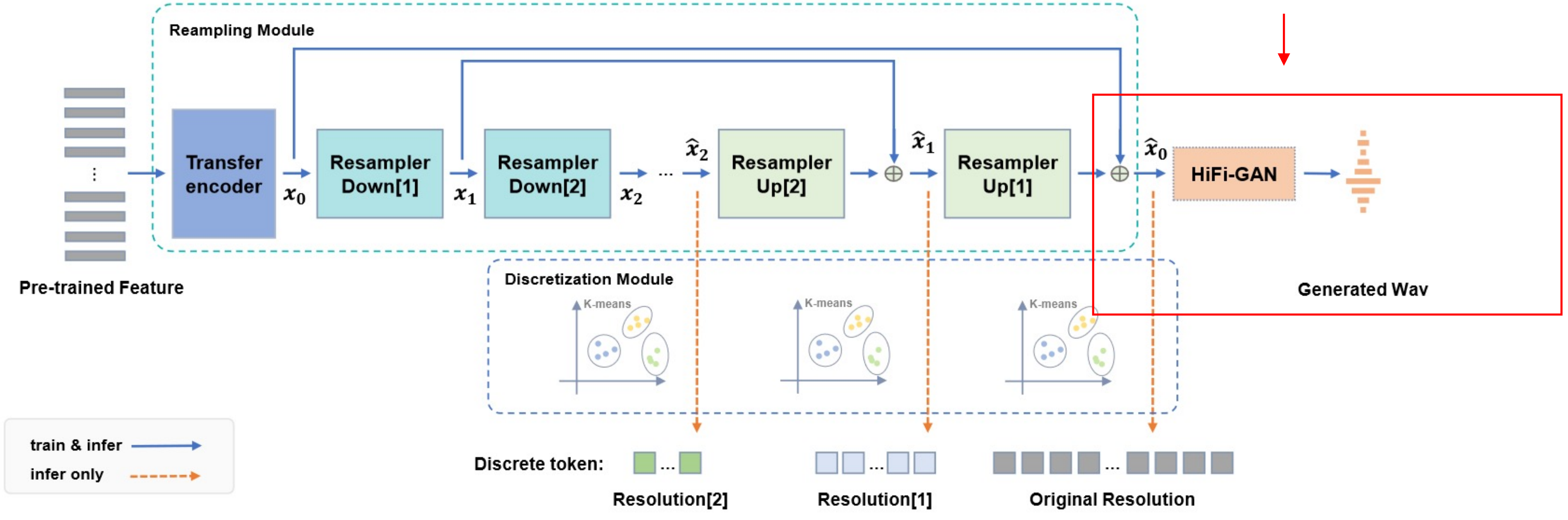
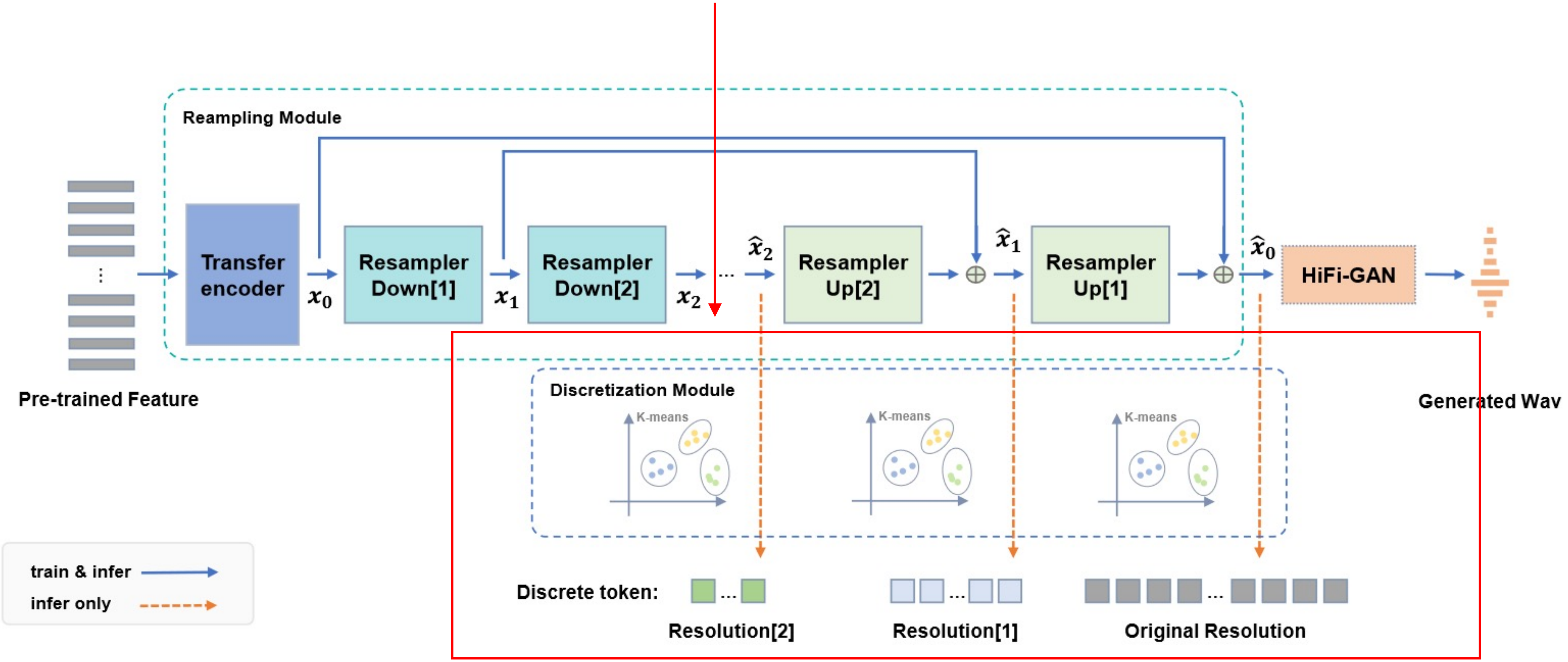


Figure 1: *Illustration of the overall workflow of our proposed SingOMD.*

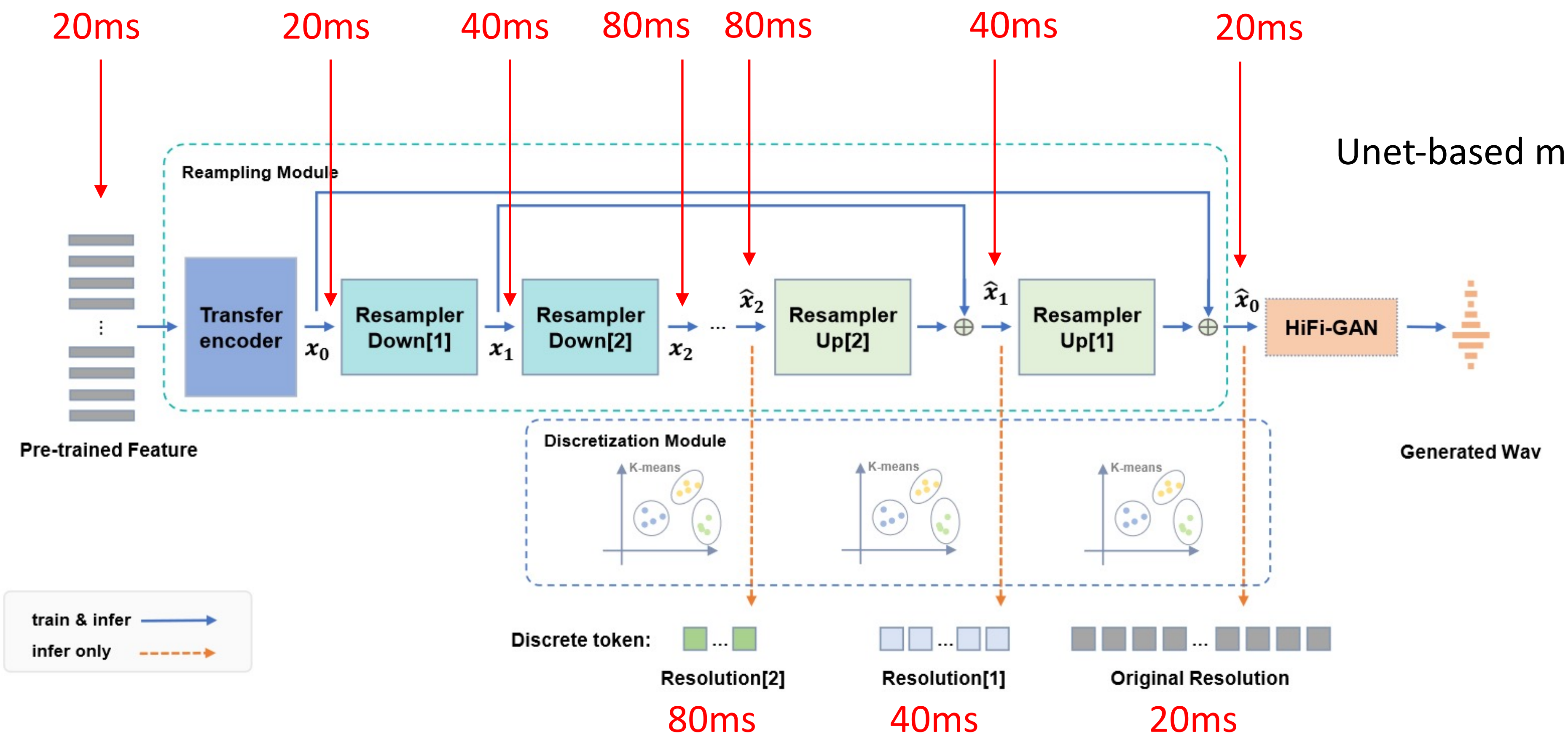
Step 4: obtain multi resolution discrete tokens

After training, cluster over multi resolution features to get corresponding discrete tokens



An example: Down[1] = Down[2] = 2, SSL output features in 20ms

Unet-based module: Down[i] = Up[i]



Training Details

- SingOMD Pre-training Datasets (210h):
  - ACE-Opencpop
  - M4Singer
  - Opencpop
  - OpenSinger

- Speech SSL base
  - HuBERT-base on Librispeech

- Downstream tasks (with Opencpop):
  - Singing resynthesis (i.e., vocoder)
  - SVS

Task 1: Singing Resynthesis

Note: to fully investigate the performance of discrete tokens, we **do not include F0** as input (which is different from TokSing)

- Task: resynthesize waveform from input discrete token
- Model: Discrete HiFiGAN

	Method	SSL	Resolution	MCD ↓	F0 RMSE ↓	S. ACC.↑	VUV Error ↓	MOS ↑
1	Baseline	HuBERT-base/3	(20)	8.7103	0.2192	25.40%	9.93%	2.46 ($\pm$ 0.06)
2	Baseline	HuBERT-base/3+10+11	(20)	8.8802	0.2922	27.42%	8.74%	2.34 ($\pm$ 0.05)
3	Baseline	HuBERT-base/sum	(20)	7.6427	0.1847	38.90%	**7.66%**	2.78 ($\pm$ 0.06)
4	SingOMD (ours)	HuBERT-base/sum	(20,)	6.9693	0.2167	60.32%	8.24%	3.39 ($\pm$ 0.06)
5	SingOMD (ours)	HuBERT-base/sum	(20, 40)	6.6414	**0.1806**	64.02%	8.41%	3.48 ($\pm$ 0.06)
6	SingOMD (ours)	HuBERT-base/sum	(20, 40, 80)	**6.5766**	0.1828	**64.83%**	8.16%	**3.55 ($\pm$ 0.07)**
7	Ground Truth	-	-	-	-	-	-	4.66 $\pm$ 0.06

Task 1: Singing Resynthesis

- Task: resynthesize waveform from input discrete token
- Model: Discrete HiFiGAN

	Method	SSL	Resolution	MCD ↓	F0 RMSE ↓	S. ACC.↑	VUV Error ↓	MOS ↑
1	Baseline	HuBERT-base/3	(20)	8.7103	0.2192	25.40%	9.93%	2.46 (± 0.06)
2	Baseline	HuBERT-base/3+10+11	(20)	8.8802	0.2922	27.42%	8.74%	2.34 (± 0.05)
3	Baseline	HuBERT-base/sum	(20)	7.6427	0.1847	38.90%	**7.66%**	2.78 (± 0.06)
4	SingOMD (ours)	HuBERT-base/sum	(20,)	6.9693	0.2167	60.32%	8.24%	3.39 (± 0.06)
5	SingOMD (ours)	HuBERT-base/sum	(20, 40)	6.6414	**0.1806**	64.02%	8.41%	3.48 (± 0.06)
6	SingOMD (ours)	HuBERT-base/sum	(20, 40, 80)	**6.5766**	0.1828	**64.83%**	8.16%	**3.55 (± 0.07)**
7	Ground Truth	-	-	-	-	-	-	4.66 ± 0.06

the effectiveness of transferring encoder in single resolution

Task 2: SVS

- Model:
  - Discrete cascaded SVS system
  - Mel-spectrograms cascaded SVS system (XiaoiceSing as acoustic model)

Model	MCD ↓	F0 RMSE ↓	MOS ↑
Mel spectrogram	**6.9283**	0.2610	3.04 ± 0.06
HuBERT-base/3	9.5528	0.2321	2.34 ± 0.06
HuBERT-base/3+10+11	9.7585	0.3200	2.34 ± 0.05
DiscreteSVS+SingOMD	7.7234	**0.1941**	$\mathbf{3.10 \pm 0.06}$
Ground Truth	-	-	4.66 ± 0.06

Mel spectrogram:

Discrete SingOMD:

More Demo at: https://interspeech2024singomd.github.io/

Take Home Messages

- Propose SingOMD, a novel method to construct singing-oriented discrete representations for singing generation by leveraging speech SSL models

- Experiments demonstrate the robustness, efficiency, and effectiveness of SingOMD tokens

Training code and pre-trained models are currently in ParallelWaveGAN Pull Request!

Evaluation

SingMOS: An Extensive Open-Source Singing Voice Dataset for MOS Prediction & An Exploration on Singing MOS Prediction

Yuxun Tang[1], Jiatong Shi[2], Yuning Wu[1] , Qin Jin[1]

[1] Renmin University of China, [2] Carnegie Mellon University

Evaluation is Hard

Show less correlation with audio quality

- How to evaluate a singing clip?

- **Objective Metrics**:
  - Mel cepstral distortion (MCD)
  - Logarithmic F0 root mean square error (F0 RMSE)
  - Semitone accuracy (S. Acc.)
  - Voice/Unvoice Error (V/UV E.)
- **Subjective Metrics**:
  - Mean Opinion Score (MOS)

MOS Predictor (in speech)

- Several works in speech focusing on MOS prediction
  - Direct neural networks:
    - MOSNet (Lo. et al. 2019)
    - MBNet (Leng et al. 2021)
    - LDNet (Huang et al. 2021)
    - …
  - SSL-based
    - SSL-MOS (Huang et al. 2022)
    - UTMOS (Saeki et al. 2022)
    - DDOS (Tseng et al. 2022)
    - LE-SSL-MOS (Qi. et al. 2023)
    - …

Trained with data from :

- Speech Voice Conversion Challenges
- Speech Challenges for TTS
- …

MOS Predictor (in speech)

- Several works in speech focusing on MOS prediction
  - Direct neural networks:
    - MOSNet (Lo. et al. 2019)
    - MBNet (Leng et al. 2021)
    - LDNet (Huang et al. 2021)
    - …
  - SSL-based
    - SSL-MOS (Huang et al. 2022)
    - UTMOS (Saeki et al. 2022)
    - DDOS (Tseng et al. 2022)
    - LE-SSL-MOS (Qi. et al. 2023)
    - …

Trained with data from :

- Speech Voice Conversion

	Sim-O ↑	Sim-R ↑	WER↓	WER* ↓	UTMOS ↑	CMOS↑	SMOS↑
Ground Truth	0.68	-	1.94	0.68	4.14	+0.08	3.85
VALL-E ♥	-	0.58	5.90	-	-	-	-
VALL-E ♦	0.47	0.51	6.11	4.87	3.68	-0.60	3.46
NaturalSpeech 2♠	0.55	0.62	1.94	1.24	3.88	-0.18	3.65
Voicebox♠	0.64	0.67	2.03	1.81	3.82	-0.23	3.69
Voicebox♦	0.48	0.50	2.14	1.24	3.73	-0.32	3.52
Mega-TTS 2♠	0.53	-	2.32	2.17	4.02	-0.20	3.63
UniAudio♠	0.57	0.68	2.49	1.81	3.79	-0.25	3.71
StyleTTS 2♣	0.38	-	2.49	1.58	3.94	-0.21	3.07
HierSpeech++♣	0.51	-	6.33	4.97	3.80	-0.41	3.50
NaturalSpeech 3	**0.67**	**0.76**	**1.81**	**1.13**	**4.30**	**0.00**	**4.01**

Speech MOS predictor starts to be utilized to evaluate samples.

(Naturalspeech 3, Ju et al. 2024)

What About Singing Voices?

- Challenges of Singing MOS Predictor
  - Data! Data! Data!
- Why not start collecting the data?

Data Construction

- Data collection and selection

Dataset (12):

- zh-datasets (5): Opencpop, M4Singer, ACE-Opencpop, Kising, SingGen
- jp-datasets (7): JVSMusic, Kiritan, Ofuton, Amaboshi, Natsume, Oniku, Namine

Model (33+1):

- Singing Voice Synthesis (12)
- Singing Voice Conversion (7)
- Vocoder (9)
- Codec (5)
- Ground-Truth

Setting (6):

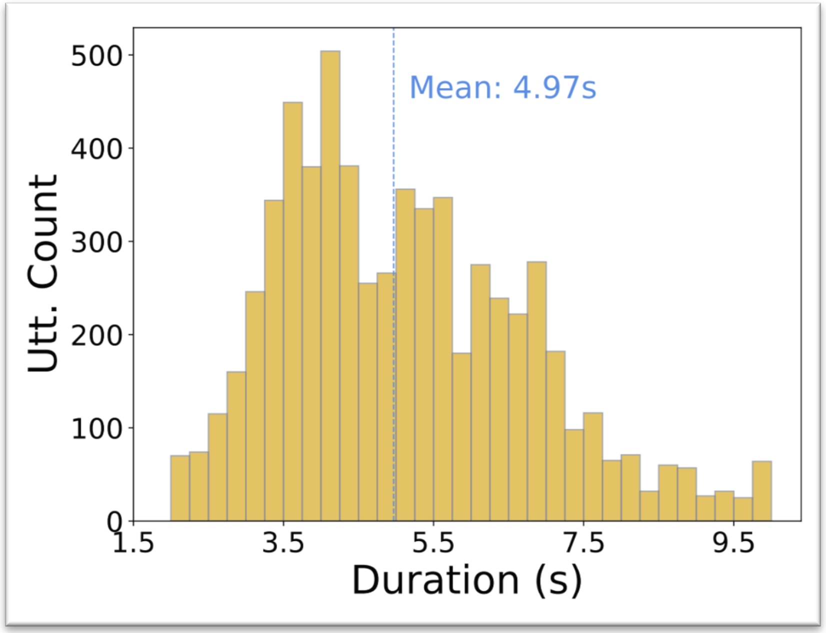
- Sample rating (3): 44kHz, 24kHz, **16kHz**
- Codebook number in codec (6): 4, 8, **32**

Data Construction (Cont’d)

- Data collection and selection

Statistic:

- 105 Systems (Dataset-Model-Setting)
- 6583 singing samples: 6175 synthetic samples, 408 GT samples
  - 2236 samples in 27 SVS systems
  - 1263 samples in 17 SVC systems
  - 1406 samples in 28 vocoder systems
  - 1270 samples in 27 codec systems
  - 408 samples in 6 GT systems



Data Construction (Con’td)

- Data Annotation

VoiceMOS Challenge 2024

  - V1: 31 systems, 90-100 samples / system, 5 annotators
    - Used in VoiceMOS Challenge 2024 (Track 2, the official set)
  - V2: 75 systems, 50 samples / system, 25 annotators

Dataset Construction

- Data Split
  - Main subset (3793): split 70% , 10%, 20% into train / dev / eval set
  - Unseen set (2763): eval set
    - unseen model (4): zh (2), jp (3), including a generated dataset
    - unseen dataset: 14 models cover SVS, SVC, vocoder, codec

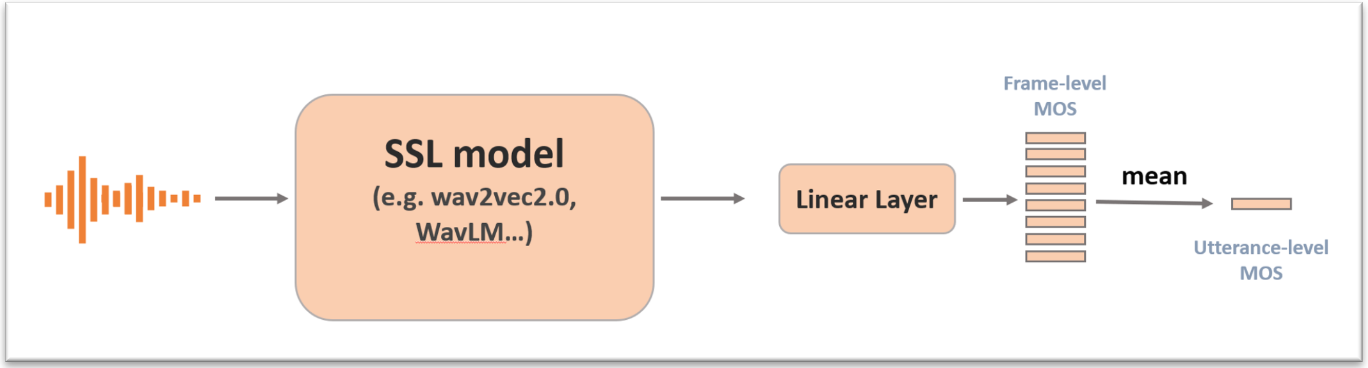
Dataset Analyses

- **Ovearll performance:**
- GT > vocoder > SVC > SVS > Codec

- Top performing systems:
  - **Vocoder**: Diffwave, HiFiGAN
  - **SVC**: sovits_contentvec, sovits_contentvec_nsf-hifigan_enhance
  - **SVS**: nnsvs_unk, acesinger, visinger
  - **Codec**: amuse_dac

Baseline on SingMOS

- Experimental Settings:
  - baseline model: SSL-MOS with loss margin
  - finetune up to 50 epochs
  - three random seeds to calculate the average



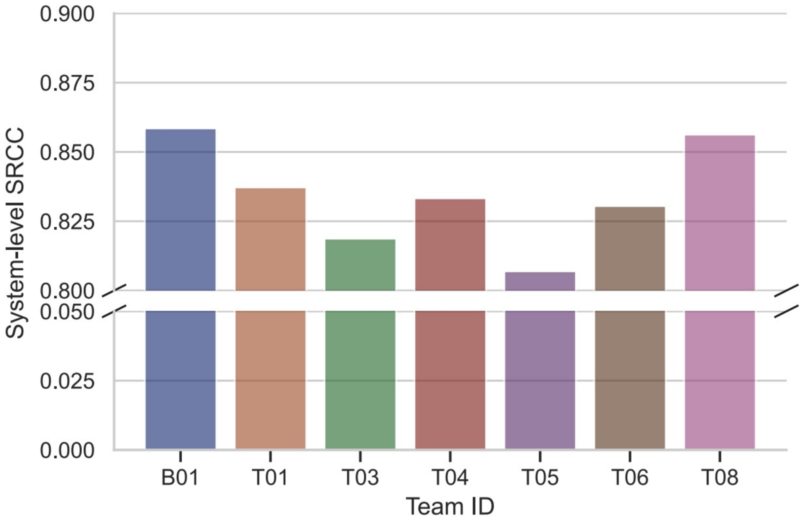
Baseline Experiments (Seen Dataset)

SSL Model	Utterance				System			
	Test Error↓	LCC↑	SRCC↑	KTRCC↑	Test Error↓	LCC↑	SRCC↑	KTRCC↑
wav2vec2.0	0.389	**0.659**	**0.644**	**0.478**	**0.029**	**0.943**	**0.905**	**0.751**
wav2vec2.0*	0.482	0.554	0.574	0.419	0.065	0.873	0.815	0.633
XLS-R*	**0.385**	0.603	0.595	0.434	0.034	0.915	0.834	0.653
HuBERT	0413	0.582	0.557	0.408	0.070	0.839	0.820	0.647
HuBERT*	0.422	0.559	0.556	0.405	0.062	0.848	0.845	0.653
WavLM	0.442	0.561	0.566	0.407	0.042	0.882	0.828	0.657
WavLM*	0.393	0.591	0.590	0.430	0.046	0.892	0.852	0.663

* indicates large models (300M version)

Larger is not better? Multilingual is not better?

Baseline Experiments (Seen Dataset)



* indicates large mod

Interesting Fact:
The Baseline we provided achieved the best performance in the VoiceMOS 2024 challenge

Larger is not better? Multilingual is not better?

Baseline Experiments (Unseen Dataset)

SSL Model	Utterance				System			
	Test Error↓	LCC↑	SRCC↑	KTRCC↑	Test Error↓	LCC↑	SRCC↑	KTRCC↑
HuBERT	0.574	0.675	0.527	0.386	0.221	0.866	0.611	0.455
HuBERT*	**0.503**	**0.704**	0.568	0.417	**0.206**	**0.894**	0.657	0.487
wav2vec2.0	0.658	0.685	0.525	0.383	0.298	0.881	0.635	0.485
wav2vec2.0*	0.709	0.682	0.502	0.363	0.305	**0.894**	0.585	0.430
WavLM	0.594	0.657	0.565	0.415	0.218	0.882	**0.759**	0.604
WavLM*	1.180	0.132	0.069	0.047	0.755	0.483	0.438	0.331
XLS-R*	0.540	0.688	**0.581**	**0.427**	0.247	0.888	0.753	**0.615**

* indicates large models (300M version)

The order has totally changed and results are mixed!

Take Home Messages

- We released the first MOS prediction dataset for singing voice.

- The data is open-sourced at huggingface:
  - https://huggingface.co/datasets/TangRain/SingMOS

- The pre-trained predictor is open-sourced at Github
  - https://github.com/South-Twilight/SingMOS/tree/main

Evaluation Beyond Singing Voice

VERSA: A Versatile Evaluation Toolkit for Speech, Audio, and Music

Jiatong Shi[1], Hyejin Shim[1], Jinchuan Tian[1], Siddhant Arora[1], Haibin Wu[2], Darius Petermann[3], Jia Qi Yip[4], You Zhang[5], Yuxun Tang[6], Wangyou Zhang[7], Dareen Alharthi[1], Yichen Huang[1], Koichi Saito[8], Jionghao Han[1], Yiwen Zhao[1], Chris Donahue[1], Shinji Watanabe[1],

[1]Carnegie Mellon University, [2]Microsoft, [3]Indiana University, [4]Nanyang Technological University, [5]University of Rochester, [6]Renmin University of China, [7]Shanghai Jiaotong University, [8]Sony AI

Up to **64 metrics** in speech and audio evaluation supported currently

More Funs in Singing Voice?

- Singing voice conversion
  - We hosted the first Singing Voice Conversion Challenge (SVCC) in ASRU2023
    - collaboration with Nagoya University and Tencent
- Singing voice deepfake detection
  - We hosted the Singing Voice Deepfake Detection (SVDD) Challenge in SLT2024 and MIREX

More Funs in Singing Voice?

SINGING VOICE CONVERSION CHALLENGE

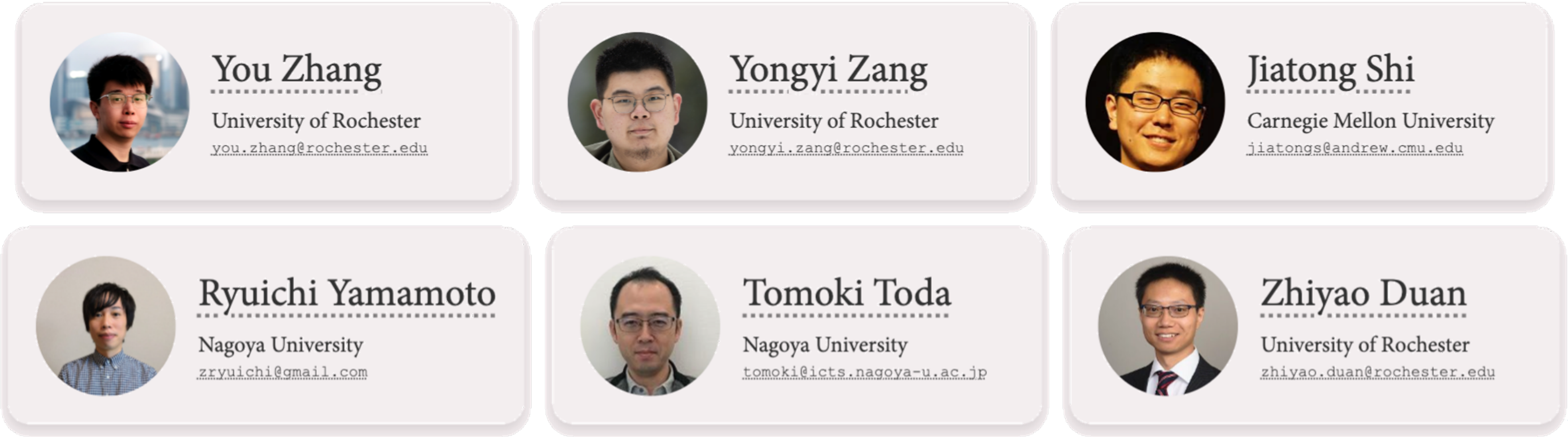
- Singing voice conversion
  - We hosted the first Singing Voice Conversion Challenge (SVCC) in ASRU2023
    - collaboration with Nagoya University and Tencent

- Tomoki Toda & Wen-Chin Huang & Lester Violeta (Nagoya University)
- Songxiang Liu (Tencent AI Lab)
- Jiatong Shi (Carnegie Mellon University)

More Funs in Singing Voice?

- Singing voice deepfake detection

  - We hosted the singing voice deepfake detection at SLT2024 and MIREX
    - collaboration with University of Rochester and Nagoya University



Thanks for Listening!