

---

# LOW-LATENCY REAL-TIME VOICE CONVERSION ON CPU

---

**Konstantine Sadv**  
Koe AI  
ksadv@koe.ai

**Matthew Hutter**  
Koe AI  
mhutter2@washcoll.edu

**Asara Near**  
Koe AI  
asara@koe.ai

## ABSTRACT

We adapt the architectures of previous audio manipulation and generation neural networks to the task of real-time any-to-one voice conversion. Our resulting model, LLVC (**L**ow-latency **L**ow-resource **V**oice **C**onversion), has a latency of under 20ms at a bitrate of 16kHz and runs nearly 2.8x faster than real-time on a consumer CPU. LLVC uses both a generative adversarial architecture as well as knowledge distillation in order to attain this performance. To our knowledge LLVC achieves both the lowest resource usage as well as the lowest latency of any open-source voice conversion model. We provide open-source samples, code, and pretrained model weights at <https://github.com/KoeAI/LLVC>.

**Keywords** voice conversion · streaming · low-latency · model distillation · open-source

## 1 Introduction

Voice conversion is the task of rendering speech in the style of another speaker while preserving the words and intonation of the original speech[1]. "Any-to-one" voice conversion converts speech from an arbitrary input speaker which may not have been seen during training to speech in the style of a single fixed speaker. Practical applications of voice conversion include speech synthesis, voice anonymization, and the alteration of one's vocal identity for personal, creative, or professional purposes.

The core challenges of voice conversion are ensuring similarity to the target speaker and creating natural-sounding output. Real-time voice conversion presents additional challenges that existing high-quality speech synthesis networks are ill-suited for: not only must the network operate faster than real time, but it also must operate with low latency and with minimal access to future audio context. Lastly, real-time voice conversion networks intended for widespread consumer usage must also be able to operate in low-resource computational environments.

This paper proposes an any-to-one voice conversion model based on the Waveformer architecture[2]. While Waveformer is designed to perform real-time sound extraction, LLVC is trained on an artificial parallel dataset of speech from various speakers which have all been converted to sound like a single target speaker with the objective of minimizing perceptible difference between the model output and the synthetic target speech. LLVC is presented as the first open-source model which can convert voices in a streaming manner on consumer CPUs with a latency as low as 20ms.

## 2 Related work

### 2.1 Voice conversion

Early approaches to voice conversion used Gaussian mixture models[3], with more recent approaches using artificial neural networks[4] and contemporary architectures commonly including variational autoencoders (VAEs) and generative adversarial networks (GANs)[5]. Recent approaches are generally made to operate on non-parallel datasets, referring to datasets where the speakers are not required to perform identical utterances. This is often achieved by a type of bottleneck in the architecture, such as the bottleneck in a VAE[6], adaptive instance normalization[7], k-nearest neighbors[8] or with the inclusion of pre-trained models which separate content and style, such as automatic speech recognition (ASR) or phonetic posteriorgrams (PPGs)[9].

## 2.2 Real-time voice conversion

There exists several published voice conversion architectures capable of operating at high enough speed to make real-time conversion on consumer hardware feasible. MMVC<sup>1</sup>, so-vits-svc<sup>2</sup>, DDSP-SVC<sup>3</sup> and RVC<sup>4</sup> are incorporated into the popular real-time voice-changer<sup>5</sup> application repository on Github.

Despite their inclusion in an application dedicated to real-time voice conversion, none of the cited architectures are trained to operate on low-latency streaming audio. Naively converting short sequential segments of audio results in perceptually-degraded output, so the networks are instead adapted for the streaming task by prefixing new input with previous audio context, trading computational efficiency for increased conversion quality.

QuickVC[10] is capable of running efficiently on CPU and can be adapted to real-time conversion using the same process as the architectures above. Regardless, the absence of streaming-specific architecture leaves this model subject to the same quality and efficiency trade-off as the previously cited models.

The above models share an encoder-decoder structure inspired by VITS[11]. The encoder is comprised of a pre-trained encoder, usually contentvec[12] or hubert-soft[13], which are designed to encode speech content without encoding input speaker characteristics such as pitch and timbre. The decoders of MMVC, so-vits-svc, DDSP-SVC and RVC are based on the architecture of HiFi-GAN, while QuickVC uses a vocoder based on the inverse short-time Fourier transform operation[14].

## 2.3 Streaming audio processing

Neural audio codecs such as LPCNet[15], and EnCodec[16] are designed to operate in low-resource streaming settings and have a similar encoder-decoder structure to the real-time voice conversion systems described above. However, these audio codec encoders seek to preserve input speaker identity along with speech content in order to ensure the fidelity of reconstructed audio, and are thus unsuitable for the task of voice conversion.

Waveformer's encoder-decoder architecture is designed to modify input audio by constructing a mask which is added to the input audio signal in order to isolate a type of sound present in the training set, i.e acoustic guitar, coughing, gunshot. While the encoder's initial convolution provides access to a small amount context, dilated causal convolutions (DCC) in the encoder and a masked transformer that attends only to present and past tokens in the decoder ensure that the model's inference is based mostly on past data. This makes the architecture well-adapted for a streaming setting, where requiring future context introduces additional latency. Additionally, the causal nature of the encoder and decoder allow intermediate calculations to be cached for future inference passes, which gives the network access to past context without requiring the entire context to be run through every part of the network, increasing inference speed.

## 2.4 Knowledge distillation

Model distillation in the realm of deep learning refers to the process of utilizing a larger, more complex "teacher" model to supplement the training of a smaller "student" model [17]. This methodology is rooted in harnessing the predictive power of intricate neural architectures while ensuring computational efficiency, especially in scenarios where computational resources are scarce, or when real-time responses are imperative, such as on mobile or edge devices [18]. Model distillation has recently been utilized to great effect with imitation-trained language models, which use high-quality output of large proprietary models to perform instruction-tuning on smaller open-source language models[19].

The conventional distillation process involves a teacher model, typically characterized by its large size or loose training constraints, which is trained to perform a specific task using a given dataset, followed by training a student model to mimic the teacher's output distribution, often softened by a higher temperature in the softmax function to encapsulate more nuanced information beyond hard labels [17].

In the scenario of non-parallel data, the landscape of model distillation extends to an innovative paradigm. A teacher model is trained on non-parallel data, leveraging its large, complex architecture to sift through and assimilate representations from the inherently unstructured and unaligned data. Following this, a synthetic parallel dataset is engineered based on the teacher's acquired knowledge, which in turn serves as the training ground for the student model [20, 21].

<sup>1</sup>[https://github.com/isletennos/MMVC\\_Trainer](https://github.com/isletennos/MMVC_Trainer)

<sup>2</sup><https://github.com/svc-develop-team/so-vits-svc>

<sup>3</sup><https://github.com/yx111c/DDSP-SVC>

<sup>4</sup><https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI>

<sup>5</sup><https://github.com/w-okada/voice-changer>

Although parallel speaker datasets have historically been challenging to create, introducing additional challenges such as aligning the utterances in time[22], the quality of modern voice conversion networks is now high enough such that they can now be artificially created. This can be done by using a pre-existing any-to-one or any-to-many voice conversion network to generate time-aligned parallel voice datasets. These artificial datasets can scale to arbitrary size simply by increasing the amount of input and output pairs generated from inference. After a parallel dataset has been obtained, smaller models can be trained on this dataset which require fewer parameters and less architectural complexity.

### 3 LLVC

#### 3.1 Architecture

Our proposed model is composed of a generator and a discriminator. Only the generator is used at inference time.

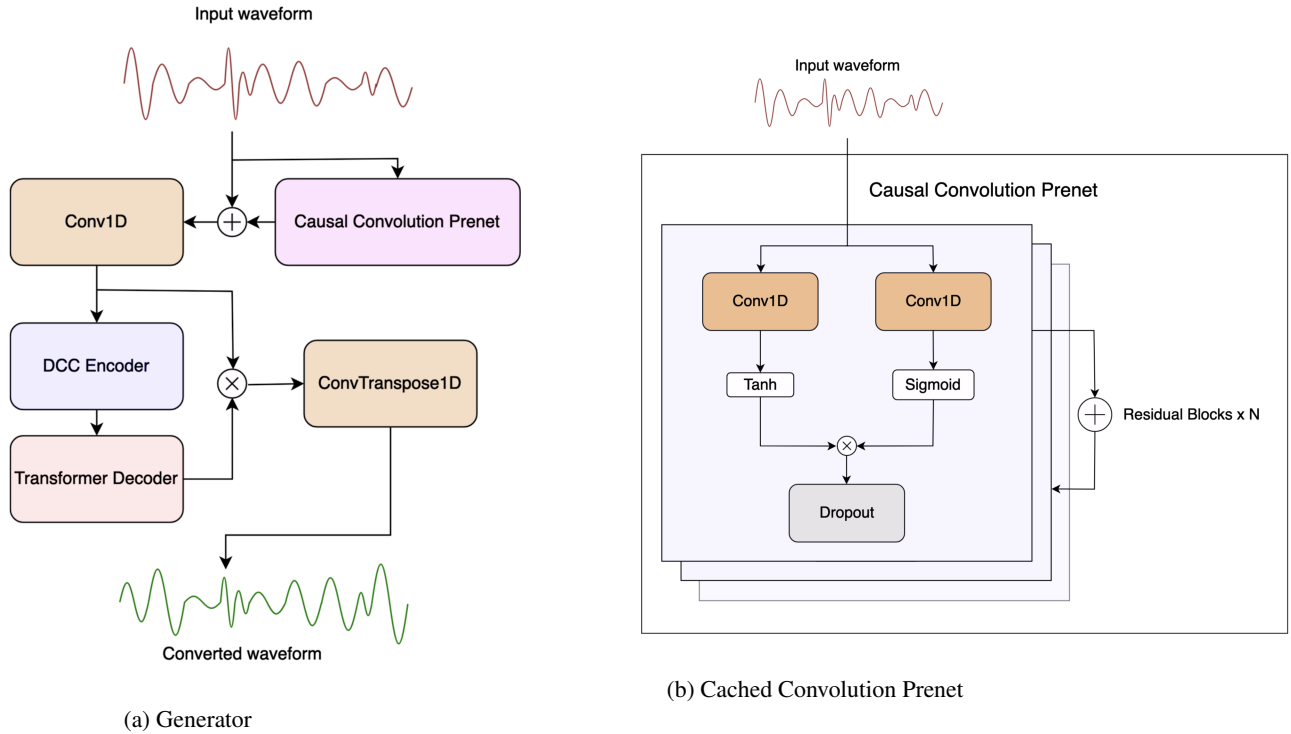


Figure 1: Generator (figure 1a) and Causal Convolution Prenet architecture (figure 1b). For details on the DCC Encoder and Transformer Decoder architectures, see the Waveformer paper. Note that the Waveformer’s Transformer Decoder takes a label query vector as input, which we do not use.

##### 3.1.1 Generator

Our generator is derived from Waveformer’s streaming encoder-decoder model. We adopt Waveformer’s 512-dimensional encoder and 256-dimensional decoder as the base for our model, though we decrease encoder depth from 10 to 8 layers and decreasing lookahead to 16 samples for lower inference latency and computation speed. Based on the success of causal U-Nets for speech modeling and enhancement[23, 24], we prefix the model with a prenet composed of causal convolutions.

##### 3.1.2 Discriminator

We adopt the multi-period discriminator architecture of VITS<sup>6</sup>, with discriminator periods of [2, 3, 5, 7, 11, 17, 23, 37] inspired by RVC’s<sup>7</sup> v2 discriminator.

<sup>6</sup><https://github.com/jaywalnut310/vits>

<sup>7</sup><https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI>

### 3.2 Dataset

We take the LibriSpeech clean 360 hour train split as input to our model[25]. This dataset consists of audio independently recorded by 922 English speakers with diverse speech characteristics and is thus a reasonable starting point for an any-to-one voice conversion system. We hold back a random sample of 2% of the files in this dataset from the training set to use for validation. We additionally use the dev-clean split, which contains a disjoint set of speakers from the 360 split in order to validate conversion on unseen input speakers.

We generate parallel utterances in the style of a single target speaker by converting the LibriSpeech files with an RVC v2 model trained on 39 minutes of audio from LibriSpeech speaker 8312, obtained from the librivox.org website. We fine tune a 32k RVC v2 base model<sup>8</sup> for 325 epochs on the target speaker data, using the RMVPE pitch extraction method [26]. The typical RVC pipeline includes a step where encoded input speaker data is mixed with encoded target speaker data retrieved from indexed ground-truth data. We choose to omit this step because we found it to decrease performance and intelligibility without improving conversion quality or resemblance. We downsample the 32kHz converted audio to 16kHz to match the sample rate of the unconverted input.

### 3.3 Training

We trained our model for 500,000 steps (53 epochs) on a single RTX 3090 GPU for 3 days at batch size 9. We used an AdamW optimizer and an exponential learning rate scheduler with gradient normalized to 1 to stabilize training. We set the learning rate to  $5e-4$ , learning rate decay to 0.999, AdamW momentum to 0.8, 0.999, and AdamW epsilon to  $1e-9$ .

#### 3.3.1 Loss

Our discriminator uses the same loss as the discriminator from VITS. Our generator uses a weighted sum of the VITS generator and feature loss as well as mel spectrogram and self-supervised speech representation based losses. The mel spectrogram loss is derived from the VITS mel loss, though we replace the VITS implementation with multi-resolution mel spectrogram loss from the auraloss library[27]. The self-supervised representation loss is inspired by Close, et al. (2023)[28], which found that loss based on L1 distance between features encoded by the pretrained fairseq HuBERT Base model was effective for speech enhancement.

### 3.4 Inference

The LLVC streaming inference procedure follows Waveformer’s chunk-based inference with lookahead. A single chunk is composed of  $dec\_chunk\_len * L$  samples. Inference additionally requires lookahead of  $2L$  samples, for a total latency of

$$(dec\_chunk\_len * L + 2L) / F_s \quad (1)$$

seconds, where  $F_s$  is the audio sample rate in Hz. It is also possible to run the network with  $N$  chunks at a time, increasing latency in order to improve the real-time factor of conversion. The file `infer.py` in the associated Github repository demonstrates the implementation of streaming inference at variable latency.

## 4 Experiments

In addition to the architecture described above, we trained two additional variants of our model. Hyperparameters for our runs can be found in the `.json` configuration files in the `experiments/` directory of the linked repository.

### 4.1 No causal convolution prenet

The causal convolution prenet adds latency to the model’s forward pass, so we performed an ablation run to test its impact on output quality. Hyperparameters and training steps were identical to that of the main LLVC model. We label this experiment LLVC-NC in our comparisons.

### 4.2 HiFi-GAN discriminator

Compared to the VITS discriminator, the HiFi-GAN discriminator uses fewer multi-period sub-discriminators at smaller period sizes, and more multi-scale sub-discriminators at larger period sizes. We reduced the training batch size to 7 but otherwise keep hyperparameters and training step count identical to LLVC. We label this experiment LLVC-HFG in our comparisons.

<sup>8</sup>[https://huggingface.co/lj1995/VoiceConversionWebUI/tree/main/pretrained\\_v2](https://huggingface.co/lj1995/VoiceConversionWebUI/tree/main/pretrained_v2)

## 5 Results

### 5.1 Evaluation Dataset

We used the LibriSpeech test-clean files as input for conversion. We used N-second clips from the LibriSpeech dataset for speaker 8312 to test quality and self-similarity of the ground truth dataset.

### 5.2 Comparison

We select two models for comparison with LLVC with the criterion of minimizing inference latency on CPU.

- No-F0 RVC: Pitch estimation creates a performance bottleneck for RVC, but the RVC developers provide pre-trained models that do not take pitch as input. We fine-tuned the RVC v2 32k no-f0 models on the 39 minutes of speaker 8312 data for 300 epochs.
- QuickVC: We fine-tuned the pre-trained QuickVC model linked in the official repository for 100 epochs on the 39 minutes of speaker 8312 data downsampled to 16kHz.

### 5.3 Performance

All models were evaluated on a Intel(R) Core(TM) i9-10850K CPU @ 3.60GHz. For No-F0 RVC and QuickVC, we aimed to achieve the lowest latency and highest amount of context that would allow the models to consistently run at above 1x real-time: a new content window of 100ms with a context buffer of 1024 samples for No-F0 RVC, and a window of 50ms and 2048 samples for QuickVC. LLVC was tested with the smallest new content window that the architecture could accommodate: about 15ms, as per ??.

We obtain performance numbers by averaging inference latency and the real-time factor (RTF) for conversions performed on the 2620 files LibriSpeech test-clean dataset, where RTF is the seconds of speech generated in 1 second of wall time. LLVC and LLVC-HFG have identical generator architectures, so differences in performance have no bearing on the efficiency of the underlying models. The lowest end-to-end latency and highest RTF scores have been bolded.

	End-to-End Latency (ms)	RTF
No-F0 RVC	189.772	1.114
QuickVC	97.616	1.050
LLVC (ours)	19.696	2.769
LLVC-NC (ours)	<b>18.327</b>	<b>3.677</b>
LLVC-HFG (ours)	19.563	2.850

### 5.4 Naturalness and Target-Speaker Similarity

We followed Guo et al. (2023)[10] to obtain Mean Opinion Scores (MOS) for naturalness and similarity to the target speaker of the converted speech. We recruited subjects on Amazon Mechanical Turk. 15 subjects evaluated naturalness of 4 utterances from the dataset and 4 converted utterances per model. 15 subjects individually evaluated the similarity of 2 utterances from the ground-truth dataset and the similarity of 4 converted utterances to 2 clips from the ground-truth dataset. The highest scores among the converted audio are in bold.

	Naturalness	Similarity
Ground Truth	3.7	3.88
No-F0 RVC	3.58	3.35
QuickVC	3.28	3.26
LLVC (ours)	3.78	3.83
LLVC-NC (ours)	3.73	3.7
LLVC-HFG (ours)	<b>3.88</b>	<b>3.9</b>

## 5.5 Objective Metrics

We use the Resemblyze<sup>9</sup> and WVMOS<sup>10</sup> libraries[29] in order to obtain metrics for target-speaker similarity and quality for the entire test-clean dataset. We obtain a baseline for comparison by evaluating 10 different 10-second clips from the ground truth against each other. The highest scores among the converted audio have been bolded.

	Resemblyze	WVMOS
Ground Truth	0.898	3.854
No-F0 RVC	<b>0.846</b>	2.465
QuickVC[10]	0.828	2.828
LLVC (ours)	0.829	3.605
LLVC-NC (ours)	0.821	<b>3.677</b>
LLVC-HFG (ours)	0.819	3.543

## 6 Conclusion and Further Work

Our work demonstrates the feasibility of ultra-low-latency low-resource voice conversion. LLVC is able to run in a streaming manner on devices that lack a dedicated GPU such as laptops and mobile phones.

We performed dataset preparation and training on a single consumer-grade GPU, using data and pretrained models freely available online. While we trained our own RVC v2 model, any pretrained RVC v2 model can be used to create a dataset for LLVC training. By open-sourcing our code, we hope to provide a broadly accessible option for creating and using real-time voice changing models.

Our choice of training data contained only clean English speech, even though our method of constructing the parallel dataset is language-independent and relatively robust to noise. Incorporating multi-lingual and noisy speech could create a model that generalizes better across diverse speakers. Conversely, our model could be fine-tuned on a dataset comprised of only a single input speaker converted to a target voice in order to create a personalized voice conversion model.

## Acknowledgments

Koe AI<sup>11</sup> provided compute and funding for this research. We thank Dr. Kyle Wilson at Washington College and Dr. Lorenz Diener for providing feedback on the first draft of the preprint.

## References

- [1] Tomasz Walczyna and Zbigniew Piotrowski. Overview of voice conversion methods based on deep learning. *Applied Sciences*, 13(5):3100, Feb 2023.
- [2] Bandhav Veluri, Justin Chan, Malek Itani, Tuochao Chen, Takuya Yoshioka, and Shyamnath Gollakota. Real-time target sound extraction, 2023.
- [3] Stephen Shum. Probabilistic voice conversion using gaussian mixture models, 2008.
- [4] Tzu hsien Huang, Jheng hao Lin, Chien yu Huang, and Hung yi Lee. How far are we from robust voice conversion: A survey, 2021.
- [5] Kaizhi Qian, Zeyu Jin, Mark Hasegawa-Johnson, and Gautham J. Mysore. F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2020.
- [6] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss, 2019.
- [7] Yen-Hao Chen, Da-Yi Wu, Tsung-Han Wu, and Hung yi Lee. Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization, 2020.
- [8] Matthew Baas, Benjamin van Niekerk, and Herman Kamper. Voice conversion with just nearest neighbors, 2023.

<sup>9</sup><https://github.com/resemble-ai/Resemblyzer>

<sup>10</sup><https://github.com/AndreevP/wvmos>

<sup>11</sup><https://koe.ai/>

- [9] Songxiang Liu, Yuwen Cao, Disong Wang, Xixin Wu, Xunying Liu, and Helen Meng. Any-to-many voice conversion with location-relative sequence-to-sequence modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1717–1728, 2021.
- [10] Houjian Guo, Chaoran Liu, Carlos Toshinori Ishi, and Hiroshi Ishiguro. Quickvc: Any-to-many voice conversion using inverse short-time fourier transform for faster conversion, 2023.
- [11] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech, 2021.
- [12] Kaizhi Qian, Yang Zhang, Heting Gao, Junrui Ni, Cheng-I Lai, David Cox, Mark Hasegawa-Johnson, and Shiyu Chang. Contentvec: An improved self-supervised speech representation by disentangling speakers, 2022.
- [13] Benjamin van Niekerk, Marc-Andre Carbonneau, Julian Zaidi, Matthew Baas, Hugo Seute, and Herman Kamper. A comparison of discrete and soft speech units for improved voice conversion. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2022.
- [14] Masaya Kawamura, Yuma Shirahata, Ryuichi Yamamoto, and Kentaro Tachibana. Lightweight and high-fidelity end-to-end text-to-speech with multi-band generation and inverse short-time fourier transform, 2023.
- [15] Jean-Marc Valin and Jan Skoglund. Lpcnet: Improving neural speech synthesis through linear prediction, 2019.
- [16] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression, 2022.
- [17] Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. Knowledge distillation: A survey. *CoRR*, abs/2006.05525, 2020.
- [18] Abdolmaged Alkhulaifi, Fahad Alsahli, and Irfan Ahmad. Knowledge distillation in deep learning and its applications. *PeerJ Computer Science*, 7, 2021.
- [19] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.
- [20] Tohru Nagano, Takashi Fukuda, and Gakuto Kurata. Knowledge distillation leveraging alternative soft targets from non-parallel qualified speech data. *ArXiv*, abs/2112.08878, 2021.
- [21] Shaolin Zhu, Shangjie Li, Shiwei Gu, and Lin Xu. Mining parallel sentences from internet with multi-view knowledge distillation for low-resource language pairs. *Knowledge and Information Systems*, 2023.
- [22] Elina Helander, Jan Schwarz, Jani Nurminen, Hanna Silén, and M. Gabbouj. On the impact of alignment on voice conversion performance. In *Interspeech*, 2008.
- [23] Daniel Stoller, Mi Tian, Sebastian Ewert, and Simon Dixon. Seq-u-net: A one-dimensional causal u-net for efficient sequence modelling, 2019.
- [24] Xinlei Ren, Xu Zhang, Lianwu Chen, Xiguang Zheng, Chen Zhang, Liang Guo, and Bin Yu. A causal u-net based neural beamforming network for real-time multi-channel speech enhancement. In *Interspeech*, 2021.
- [25] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.
- [26] Haojie Wei, Xueke Cao, Tangpeng Dan, and Yueguo Chen. Rmvpe: A robust model for vocal pitch estimation in polyphonic music, 2023.
- [27] Christian J. Steinmetz and Joshua D. Reiss. auraloss: Audio focused loss functions in PyTorch. In *Digital Music Research Network One-day Workshop (DMRN+15)*, 2020.
- [28] George Close, William Ravenscroft, Thomas Hain, and Stefan Goetze. Perceive and predict: Self-supervised speech representation based loss functions for speech enhancement. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, jun 2023.
- [29] Pavel Andreev, Aibek Alanov, Oleg Ivanov, and Dmitry Vetrov. HIFI: A unified framework for bandwidth extension and speech enhancement. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, jun 2023.