PaddleSpeech: An Easy-to-Use All-in-One Speech Toolkit

Hui Zhang[1], **Tian Yuan**[1], **Junkun Chen**[3], **Xintong Li**[2], **Renjie Zheng**[2],
Yuxin Huang[1], **Xiaojie Chen**[1], **Enlei Gong**[1], **Zeyu Chen**[1], **Xiaoguang Hu**[1],
Dianhai Yu[1], **Yanjun Ma**[1], **Liang Huang**[2,3]
[1]Baidu Inc., Beijing, China
[2]Baidu Research, Sunnyvale, CA, USA
[3]Oregon State University, Corvallis, OR, USA
{zhanghui41, yuantian01, xintongli, renjiezheng,
huangyuxin, chenxiaojie06, gongenlei, chenzeyu01}baidu.com

Abstract

PaddleSpeech is an open-source all-in-one speech toolkit. It aims at facilitating the development and research of speech processing technologies by providing an easy-to-use command-line interface and a simple code structure. This paper describes the design philosophy and core architecture of PaddleSpeech to support several essential speech-to-text and text-to-speech tasks. PaddleSpeech achieves competitive or state-of-the-art performance on various speech datasets and implements the most popular methods. It also provides recipes and pretrained models to quickly reproduce the experimental results in this paper. PaddleSpeech is publicly avaiable at https://github.com/PaddlePaddle/PaddleSpeech.[1]

1 Introduction

Speech processing technology enables humans to directly communicate with computers, which is an essential part of enormous applications such as smart home devices (Hoy, 2018), autonomous driving, and simultaneous translation (Zheng et al., 2020). Open-source toolkits boost the development of speech processing technology by lowering the barrier of application and research in this area (Young et al., 2002; Lee et al., 2001; Huggins-Daines et al., 2006; Rybach et al., 2011; Povey et al., 2011; Watanabe et al., 2018; Han et al., 2019; Wang et al., 2020; Ravanelli et al., 2021; Zhao et al., 2021).

However, the current prevailing speech processing toolkits presume that their users are experienced practitioners or researchers, so beginners might feel baffled when developing their exciting applications. For example, to prototype new speech applications with Kaldi (Povey et al., 2011), the users have to be comfortable reading and revising the provided recipes written in Bash, Perl, and Python scripts and be proficient at C++ to hack its implementation. The more recent toolkits, such as Fairseq S2T (Wang et al., 2020) and NeurST (Zhao et al., 2021), become more flexible by building on general-purpose deep learning libraries. But their complicated code styles also make it time-consuming to learn and hard to migrate from one to another. So, we have developed PaddleSpeech, providing a command-line interface and portable functions to make the development of speech-related applications accessible to everyone.

Notably, the Chinese community has many developers eager to contribute to the community. However, nearly all deep learning libraries, such as Pytorch (Paszke et al., 2019) and Tensorflow (Abadi et al., 2016), target the English community mainly, so it significantly increases the difficulty for Chinese developers. PaddlePaddle, as the only fully-functioning open-source deep learning platform targeting both the English and Chinese community, has accumulated more than 500k commits, 476k models, and is used by 157k enterprises. So, we expect PaddleSpeech, developed with PaddlePaddle can remove the barriers between the English and Chinese communities to boost the development of speech technologies and applications.

Developing speech applications for the industry is not the same scenario as conducting research in academia. The research papers mainly focus on developing novel models to perform better on specific datasets. However, a clean dataset usually does not exist when applying a speech product. So, PaddleSpeech provides on-the-fly preprocessing for the raw audios to make PaddleSpeech directly usable in product-oriented applications. Notably, some preprocessing methods are exclusive in PaddleSpeech, such as rule-based Chinese text-to-speech frontend, which can significantly benefit the performance of synthesized speech.

Performance is the cornerstone of all applica-

[1] Demo video: https://paddlespeech.readthedocs.io/en/latest/demo_video.html

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations, pages 114 - 123
July 10-15, 2022 ©2022 Association for Computational Linguistics

Task	Description	Techniques	Datasets
Sound Classification	*Label sound class*	Finetuned PANN (Kong et al., 2020b)	ESC-50 dataset (Piczak, 2015)
Speech Recognition	*Transcribe speech to text*	Deepspeech2 (Amodei et al., 2016) Conformer (Zhang et al., 2020) Transformer (Zhang et al., 2020)	Librispeech (Panayotov et al., 2015) AISHELL-1 (Bu et al., 2017)
Punctuation Restoration	*Post-add punctuation to transcribed text*	Finetuned ERNIE (Sun et al., 2019)	IWSLT2012-zh (Federico et al., 2012)
Speech Translation	*Translate speech to text*	Transformer (Vaswani et al., 2017)	MuST-C (Di Gangi et al., 2019)
Text To Speech	*Synthesis speech from text*	**Acoustic Model** Tacotron 2 (Shen et al., 2018) Transformer TTS (Li et al., 2019) SpeedySpeech (Vainer and Dušek, 2020) FastPitch (Łańcucki, 2021) FastSpeech 2 (Ren et al., 2020) **Vocoder** WaveFlow(Ping et al., 2020) Parallel WaveGAN (Yamamoto et al., 2020) MelGAN (Kumar et al., 2019) Style MelGAN (Mustafa et al., 2021) Multi Band MelGAN (Yang et al., 2021) HiFi GAN (Kong et al., 2020a)	CSMS (DataBaker) AISHELL-3 (Shi et al., 2020) LJSpeech (Ito and Johnson, 2017) VCTK (Yamagishi et al., 2019)

Table 1: List of speech tasks and corpora that are currently supported by PaddleSpeech.

tions. PaddleSpeech achieves state-of-the-art or competitive performers on various commonly used benchmarks, as shown in Table 1.

Our main contributions in this paper are two-folds.

- We introduce how we designed PaddleSpeech and what features it supports.

- We provide the implementation and reproducible experimental details that result in state-of-the-art or competitive performance on various tasks.

2 Design of PaddleSpeech

Figure 1 shows the software architecture of PaddleSpeech. As an easy-to-use speech processing toolkit, PaddleSpeech provides many complete recipes to perform various speech-related tasks and demo usage of the command line interface. Getting familiar with the top level should be enough for building speech-related applications.

The second level faces researchers in speech and language processing. The design philosophy of PaddleSpeech is model-centric to simplify the learning and development of speech processing methods. For a specific method, all computations of a specific model are included in two files under `PaddleSpeech/<task>/models/<model>`.[2]

PaddleSpeech has implemented most of the commonly used and well-performing models. A model architecture is implemented in a standalone file named by the method. Its corresponding training step and evaluation step are implemented in another `updater` file. Generally, reading or hacking these two files is enough to understand or design a model. More advanced hacking on more fine data processing or more compli-

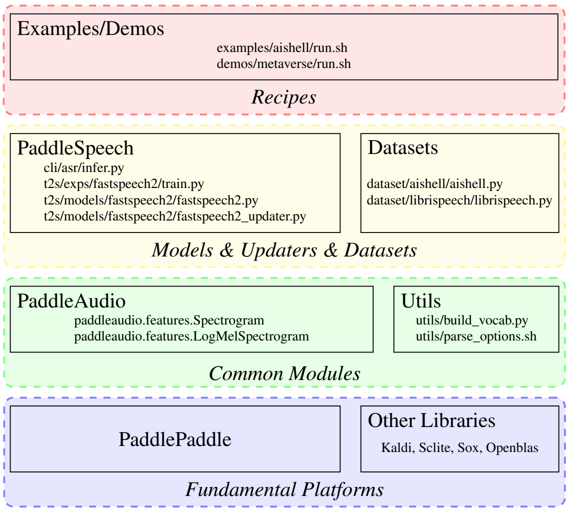
Examples/Demos
examples/aishell/run.sh
demos/metaverse/run.sh
Recipes
PaddleSpeech
cli/asr/infer.py
t2s/exps/fastspeech2/train.py
t2s/models/fastspeech2/fastspeech2.py
t2s/models/fastspeech2/fastspeech2_updater.py
Datasets
dataset/aishell/aishell.py
dataset/librispeech/librispeech.py
Models & Updaters & Datasets
PaddleAudio
paddleaudio.features.Spectrogram
paddleaudio.features.LogMelSpectrogram
Utils
utils/build_vocab.py
utils/parse_options.sh
Common Modules
PaddlePaddle
Other Libraries
Kaldi, Sclite, Sox, Openblas
Fundamental Platforms

Figure 1: Software architecture of PaddleSpeech.

[2] `<task>` includes `s2t` and `t2s` which stands for speech-to-text and text-to-speech respectively.

cated training/evaluation loop is also available at `PaddleSpeech/<task>/exps/<model>`. The original datasets can be obtained by scripts in corresponding `dataset/<dataset>/`. PaddleSpeech supports distributed multi-GPU training with good efficiency.

The standard modules, such as audio and text feature transformation and utility scripts, are implemented as libraries in the third level. The backend of PaddleSpeech is mainly PaddlePaddle with some functions from third-party libraries as shown in the fourth level. PaddleSpeech provides multiple ways to extract multiple types of speech features from raw audios using PaddleAudio and Kaldi, such as spectrogram and filterbanks, which can be varied according to the needs of the tasks.

3 Experiments

In this section, we compare the performance of models in PaddleSpeech with other popular implementations in five speech-related tasks, including sound classification, speech recognition, punctuation, speech translation, and speech synthesizing. The toolkit can reach SOTA on most tasks. All experiments in this section include details on data preparation, evaluation metrics, and implementation to enhance reproducibility.[3]

3.1 Sound Classification

Sound Classification is a task to recognize particular sounds, including speech commands (Warden, 2018), environment sounds (Piczak, 2015), identifying musical instruments (Engel et al., 2017), finding birdsongs (Stowell et al., 2018), emotion recognition (Xu et al., 2019) and speaker verification (Liu et al., 2018).

Datasets In this section, we analyze the performance of PaddleSpeech in Sound Classification on ESC-50 dataset (Piczak, 2015). The ESC-50 dataset is a labeled collection of 2000 environmental 5-second audio recordings consisting of 50 sound events, such as "Dog", "Cat", "Breathing" and "Fireworks", with 40 recordings per event.

Data Preprocessing First, we resample all audio recordings to 32 kHz, and convert them to monophonic to be consistent with the PANNs trained on AudioSet (Kong et al., 2020b). And then, we transform the recordings into log mel spectrograms by applying short-time Fourier transform on the waveforms with a Hamming window of size 1024 and a hop size of 320 samples. This configuration leads to 100 frames per second. Following Kong et al. (2019), we apply 64 mel filter banks to calculate the log mel spectrogram.

Model	Accuracy
AST-P (Gong et al., 2021)	95.6 ± 0.4
PANNs-CNN14	95.00
PANNs-CNN10	89.75
PANNs-CNN6	88.25

Table 2: 5-fold cross validation accuracy of ESC-50.

Implementation PANNs (Kong et al., 2020b) is one of the pre-trained CNN models for audio-related tasks, which is characterized in terms of being trained with the AudioSet (Gemmeke et al., 2017). PANNs are helpful for tasks where only a limited number of training clips are provided. In this case, we fine-tune all parameters of a PANN for the environment sounds classification task. All parameters are initialized from the PANN, except the final fully-connected layer which is randomly initialized. Specifically, we implement CNNs with 6, 10 and 14 layers, respectively (Kong et al., 2020b).

Results We report 5-fold cross validation accuracy values on ESC-50 dataset. As shown in Table 2, PANNs-CNN14 achieves 0.9500 5-fold cross validation accuracy that is comparable to the current state-of-the-art method (Gong et al., 2021).

3.2 Automatic Speech Recognition

Automatic Speech Recognition (ASR) is a task to transcribe the audio content to text in the same language.

Datasets We conduct the ASR experiments on two major datasets including Librispeech[4] (Panayotov et al., 2015) and Aishell-1[5] (Bu et al., 2017). Librispeech contains 1000 hours speech data. The whole dataset is divided into 3 training sets (100h clean, 360h clean, 500h other), 2 validation sets (clean, other), and 2 test sets (clean, other). Aishell contains 178 hours speech data. 400 speakers from different accent areas in China participate in the recording. The dataset is divided into the training

[3] `https://github.com/PaddlePaddle/PaddleSpeech/tree/develop/examples`

[4] `http://www.openslr.org/12/`

[5] `http://www.aishelltech.com/kysjcp`

Data	Model	Streaming	Test Data	Language Model	CER	WER
Aishell	WeNet Conformer†* (Yao et al., 2021)	✓			5.45	-
	WeNet Conformer† (Yao et al., 2021)				4.61	-
	WeNet Transformer† (Yao et al., 2021)				5.30	-
	ESPnet Conformer† (Inaguma et al., 2020)				5.10	-
	ESPnet Transformer† (Inaguma et al., 2020)				6.70	-
	SpeechBrain Transformer† (Ravanelli et al., 2021)				5.58	-
	Deepspeech 2	✓		char 5-gram	6.66	-
	Deepspeech 2			char 5-gram	6.40	-
	Transformer				5.23	-
	Conformer*	✓			5.44	-
	Conformer				4.64	-
Librispeech	WeNet Conformer† (Yao et al., 2021)		test-clean		-	2.85
	SpeechBrain Transformer† (Ravanelli et al., 2021)		test-clean	TransformerLM	-	2.46
	ESPnet Transformer† (Inaguma et al., 2020)		test-clean	TransformerLM	-	2.60
	Deepspeech 2		test-clean	word 5-gram	-	7.25
	Conformer		test-clean		-	3.37
	Transformer		test-clean	TransformerLM	-	2.40

† denotes the results are reported in their public repositories.

* denotes the results are streaming with chunk size 16.

Table 3: WER/CER on Aishell, Librispeech for ASR Tasks.

set (340 speakers), validation set, (40 speakers) and test set (20 speakers).

Data Preprocessing Deepspeech 2 takes character-level vocabularies for both English and Mandarin tasks. For other models, we use character-level vocabulary for Mandarin. And English text is preprocessed with SentencePiece (Kudo and Richardson, 2018). Both two kinds of datasets are added four additional characters, which are <' >, <space>, <blank> and <eos>. For cepstral mean and variance normalization (CMVN), a subset of or full of the training set is selected and be used to compute the feature mean and standard error. For feature extraction, we have several methods implemented, such as linear spectrogram, filterbank, and mfcc. Currently, the Deepspeech 2 model uses linear spectrogram or filterbank, but Transformer and Conformer models use filterbank. For a fair comparison, we take additional 3 dimensional pitch features into Transformer to be consistent with ESPnet.

Implementation We implement both streaming and non-streaming Deepspeech 2 (Amodei et al., 2016). The non-streaming model has 2 convolution layers and 3 LSTM layers. The streaming model has 2 convolution layers and 5 LSTM layers. The Conformer and Transformer models are implemented following Zhang et al. (2020) with 12 encoder layers and 6 decoder layers.

Results We report word error rate (WER) and character error rate (CER) for Librispeech (English) and Aishell (Mandarin) speech recognition, respectively. As shown in Table 3, Conformer and Transformer are better than Deepspeech 2. Our best models achieve comparable performance on both datasets compared with related works.

3.3 Punctuation Restoration

Punctuation restoration is a post-processing problem for ASR systems. It is crucial to improve the readability of the transcribed text for the human reader and facilitate down-streaming NLP tasks.

Datasets We conduct experiments on IWSLT2012-zh[6] dataset, which contains 150k Chinese sentences with punctuation. We select comma, period, and question marks as restore targets in this task, so we replace other punctuation with these three marks before training a model. We split the data into training, validation and testing sets with 147k, 2k, and 1k samples, respectively.

Implementation We formulate the problem of punctuation restoration as a sequence labeling task with four target classes including EMPTY, COMMA, PERIOD, and QUESTION (Nagy et al., 2021b). ERNIE (Sun et al., 2019), as a pretrained language model, achieves new state-of-the-art results on five Chinese natural language processing tasks, including natural language inference, semantic similarity,

[6] https://hltc.cs.ust.hk/iwslt/

Frameworks	De	Es	Fr	It	Nl	Pt	Ro	Ru
ESPnet-ST (Inaguma et al., 2020)	22.9	**28.0**	32.8	**23.8**	**27.4**	28.0	21.9	**15.8**
fairseq-ST (Wang et al., 2020)	22.7	27.2	32.9	22.7	27.3	28.1	21.9	15.3
NeurST (Zhao et al., 2021)	22.8	27.4	**33.3**	22.9	27.2	28.7	**22.2**	15.1
PaddleSpeech	**23.0**	27.4	32.9	22.9	26.7	**28.8**	**22.2**	15.4

Table 4: Case-sensitive detokenized BLEU scores on MuST-C *tst-COMMON*.

model	COMMA	PERIOD	QUESTION	Overall
BERTLinear†	0.4646	0.4227	0.7400	0.5424
BERTBiLSTM†	0.5190	0.5707	0.8095	0.6330
ERNIELinear	0.5142	0.5447	0.8406	0.6331

† denotes the results come from our reproduced models.

Table 5: F1-score values on IWSLT2012-zh dataset.

named entity recognition, sentiment analysis, and question answering. So, we finetune an ERNIE model for this task. More specifically, all parameters are initialized from the ERNIE pretrained model, except the final shared fully-connected layer, which is randomly initialized.

Results We report F1-score values on IWSLT2012-zh dataset. As shown in Table 5, our ERNIELinear model achieves 0.6331 overall F1-score, which is comparable with the previous work (Nagy et al., 2021a).

3.4 Speech Translation

Speech translation, where translating speech in a source language to text in another language, is beneficial in human communications.

Datasets In this section, we analyze the performance of speech-to-text translation with PaddleSpeech on MuST-C dataset (Di Gangi et al., 2019) with 8 different language translation pairs, which take the English speech as the source input.

Implementation We process the raw audios with Kaldi (Povey et al., 2011) and extract 80-dimensional log-mel filterbanks stacked with 3-dimensional pitch feature using a 25ms window size and a 10ms step size. Text is firstly tokenized with Moses tokenizer[7] and then processed by SentencePiece (Kudo and Richardson, 2018) with a joint vocabulary whose size is 8K for each language pair. We employ Transformer (Vaswani et al., 2017) as the base architecture for the speech translation experiments. In detail, the Transformer model has 12 encoder layers that follow 2 layers of 2D convolution with kernel size of 3 and stride size of 2, and 6 decoder layers. Each layer contains 4 attention heads with a size of 256. The encoder is initialized from a pretrained ASR model.

Results We report detokenized case-sensitive BLEU[8]. As shown in Table 4, PaddleSpeech can achieve competitive results compared with other frameworks.

3.5 Text-To-Speech

A Text-To-Speech (TTS) system converts given language text into speech. PaddleSpeech's TTS pipeline includes three steps. We first convert the original text into the characters/phonemes through the text frontend module. Then, through an Acoustic model, we convert the characters or phonemes into acoustic features, such as mel spectrogram, Finally, we generate waveform from the acoustic features through a Vocoder. In PaddleSpeech, the text frontend is a rule-based model inspired by expert knowledge. The Acoustic models and Vocoders are trainable.

Datasets In PaddleSpeech, we mainly focus on Mandarin and English speech synthesis. We have benchmarks on CSMSC[9], AISHELL-3[10], LJSpeech[11], VCTK[12]. Due to the limit of space, we only list the experimental results on CSMSC, which includes 12 hours speech audio corresponding to 10k sentences.

Text Frontend A text frontend module is used to extract linguistic features, characters and phonemes from given text. It mainly includes: Text Segmentation, Text Normalization (TN), Word Segmentation (WS), Part-of-Speech Tagging, Prosody Prediction and Grapheme-to-Phoneme (G2P) (see Table 6).

[7] https://github.com/moses-smt/mosesdecoder

[8] https://github.com/mjpost/sacrebleu

[9] https://www.data-baker.com/open_source.html

[10] http://www.aishelltech.com/aishell_3

[11] https://keithito.com/LJ-Speech-Dataset/

[12] https://datashare.ed.ac.uk/handle/10283/3443

	Module	Result								
PaddleSpeech	*Text*	*jīn tiān* 今天 today	shì 是 is	2020/10/29	,	zuì dī 最低 lowest	wēn dù 温度 temperature	shì 是 is	-3°C	。
	TN	今天	是	*èr líng èr líng nían shí yùe èr shí jiǔ rì* 二 零 二 零 年 十 月 二 十 九 日 2 0 0 2 year 10 month 29 day	,	最低	温度	是	*líng xià sān dù* 零下 三 度 *negative three degree*	。
	WS	今天 /	是 /	二零二零年 / 十月 / 二十九日	,	/ 最低	温度 /	是 /	零下 / 三度	。
	G2P	jin1 tian1 shi4 er4 ling2 er4 ling2 nian2 shi4 yue4 er4 shi2 jiu3 ri4				zui4 di1	wen1 du4	shi4	ling2 xia4 san1 du4	
ESPnet		jin1 tian1 shi4		2020/10/29		zui4 di1	wen1 du4	shi4	-3°C	

Table 6: An example of the text preprocessing pipeline for Mandarin TTS of PaddleSpeech and ESPnet. **TN** stands for the text normalization module, **WS** stands for the word segmentation module, **G2P** stands for the grapheme-to-phoneme module. The text normalization module for mandarin of ESPnet is not able to correctly handle dates (2020/10/29) and temperatures (-3°C).

For Mandarin, our G2P system consists of a polyphone module, which uses pypinyin and g2pM, and a tone sandhi module which uses rules based on chinese word segmentations. To the best of our knowledge, our Mandarin text frontend system is the most complete one compared with other publicly released works.

Data Preprocessing PaddleSpeech TTS uses the following modules for data preprocessing[13]: First, we use Montreal-Forced-Aligner to get the duration for corresponding phonemes. Second, we extract mel spectrograms as the features (additional pitch and energy features for Fastspeech 2). Last, we conduct the statistical normalization for each feature.

Acoustic Model Acoustic models can be mainly classified into autoregressive and non-autoregressive models. The decoding of the autoregressive model relies on previous predictions at each step, which leads to longer inference time but relatively better quality. While the non-autoregressive model generates the outputs in parallel, so the inference speed is faster, but the quality of generated result is relatively poor.

As shown in Table 1, PaddleSpeech has implemented the following commonly used autoregressive acoustic models: Tacotron 2 and Transformer TTS, and non-autoregressive acoustic models: SpeedySpeech, FastPitch and FastSpeech 2.

Vocoder As shown in Table 1, PaddleSpeech has implemented the following vocoders: WaveFlow, Parallel WaveGAN, MelGAN, Style MelGAN, Multi Band MelGAN, and HiFi GAN.

	Acoustic Model	Vocoder	MOS ↑
ESPnet	Fastspeech 2	PWGAN	2.55 ± 0.19
PaddleSpeech	Tacotron 2	PWGAN	3.69 ± 0.11
	Speedyspeech	PWGAN	3.79 ± 0.09
	Fastspeech 2	PWGAN	4.25 ± 0.09
	Fastspeech 2	Style MelGAN	4.32 ± 0.10
	Fastspeech 2	MB MelGAN	4.43 ± 0.09
	Fastspeech 2	HiFi GAN	4.72 ± 0.08

Table 7: The MOS evaluation with 95% confidence intervals for TTS models trained using CSMSC dataset. PWGAN stands for Parallel WaveGan, MB MelGAN stands for Multi-Band MelGAN.

Implementation The PadddleSpeech TTS implementation of FastSpeech 2 adopts some improvement from FastPitch and uses MFA to obtain the forced alignment (the original FastSpeech paper uses Tacotron 2). Notably, the speech feature parameters of the acoustic model and the vocoder of one TTS pipeline should be the same. Detailed settings can be found in the sample config file[14] on CSMSC dataset.

Results We report the mean opinion score (MOS) for naturalness evaluation in Table 7. We use the crowdMOS toolkit (Ribeiro et al., 2011), where 14 Mandarin samples (see Appendix A) from these 7 different models were presented to 14 workers on Mechanical Turk. As shown in Table 7, PaddleSpeech can largely outperform ESPnet on Mandarin TTS. The main reason is that PaddleSpeech TTS has a better text frontend as shown in Table 6. Compared with other models, Fastspeech 2 with

[13] https://github.com/PaddlePaddle/PaddleSpeech/blob/develop/examples/csmsc/tts3/local/preprocess.sh

[14] https://github.com/PaddlePaddle/PaddleSpeech/blob/develop/examples/csmsc/tts2/conf/default.yaml

HiFi GAN can achieve the best results.

4 Conclusion

This paper introduces PaddleSpeech, an open-source, easy-to-use, all-in-one speech processing toolkit. We illustrated the main design philosophy behind this toolkit to conduct development and research on various speech-related tasks accessible. A number of reproducible experiments and comparisons show that PaddleSpeech achieves state-of-the-art or competitive performance with the most popular models on standard benchmarks.

5 Acknowledgment

We sincerely thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Key Research and Development Project of China (2020AAA0103503).

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5. IEEE.

Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2012–2017. Association for Computational Linguistics.

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *ICML*.

Marcello Federico, Mauro Cettolo, Luisa Bentivogli, Paul Michael, and Stüker Sebastian. 2012. Overview of the iwslt 2012 evaluation campaign. In *IWSLT-International Workshop on Spoken Language Translation*, pages 12–33.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780.

Yuan Gong, Yu-An Chung, and James R. Glass. 2021. Ast: Audio spectrogram transformer. *ArXiv*, abs/2104.01778.

Kun Han, Junwen Chen, Hui Zhang, Haiyang Xu, Yiping Peng, Yun Wang, Ning Ding, Hui Deng, Yonghu Gao, Tingwei Guo, Yi Zhang, Yahao He, Baochang Ma, Yulong Zhou, Kangli Zhang, Chao Liu, Ying Lyu, Chenxi Wang, Cheng Gong, Yunbo Wang, Wei Zou, Hui Song, and Xiangang Li. 2019. DELTA: A DEep learning based Language Technology plAtform. *arXiv e-prints*.

Matthew B Hoy. 2018. Alexa, siri, cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*, 37(1):81–88.

David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W Black, Mosur Ravishankar, and Alexander I Rudnicky. 2006. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. Espnet-st: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 302–311.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020a. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *arXiv preprint arXiv:2010.05646*.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020b. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yong Xu, Wenwu Wang, and Mark D. Plumbley. 2019. Cross-task learning for audio tagging, sound event detection and spatial localization: Dcase 2019 baseline systems. *ArXiv*, abs/1904.05635.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*.

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE.

Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. 2001. Julius—an open source real-time large vocabulary recognition engine. *EUROSPEECH2001: the 7th European Conference on Speech Communication and Technology*.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713.

Bin Liu, Shuai Nie, Yaping Zhang, Shan Liang, and Wenju Liu. 2018. Deep segment attentive embedding for duration robust speaker verification.

Ahmed Mustafa, Nicola Pia, and Guillaume Fuchs. 2021. Stylemelgan: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE.

Attila Nagy, Bence Bial, and Judit Ács. 2021a. Automatic punctuation restoration with bert models. *arXiv preprint arXiv:2101.07343*.

Attila Matyas Nagy, Bence Bial, and Judit Ács. 2021b. Automatic punctuation restoration with bert models. *ArXiv*, abs/2101.07343.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. 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. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Karol J. Piczak. 2015. Esc: Dataset for environmental sound classification. *Proceedings of the 23rd ACM international conference on Multimedia*.

Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. 2020. Waveflow: A compact flow-based model for raw audio. In *International Conference on Machine Learning*, pages 7706–7716. PMLR.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. 2021. Speechbrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Flávio Ribeiro, Dinei Florêncio, Cha Zhang, and Michael Seltzer. 2011. Crowdmos: An approach for crowdsourcing mean opinion score studies. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2416–2419. IEEE.

David Rybach, Stefan Hahn, Patrick Lehnen, David Nolden, Martin Sundermeyer, Zoltan Tüske, Simon Wiesler, Ralf Schlüter, and Hermann Ney. 2011. Rasr-the rwth aachen university open source speech recognition toolkit. In *Proc. ieee automatic speech recognition and understanding workshop*.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2020. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. *arXiv preprint arXiv:2010.11567*.

Dan Stowell, Yannis Stylianou, Mike Wood, Hanna Pamula, and Hervé Glotin. 2018. Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge. *ArXiv*, abs/1807.05812.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *ArXiv*, abs/1904.09223.

Jan Vainer and Ondřej Dušek. 2020. Speedyspeech: Efficient neural speech synthesis. *arXiv preprint arXiv:2008.03802*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq S2T: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *ArXiv*, abs/1804.03209.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. 2018. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*.

Haiyang Xu, Hui Zhang, Kun Han, Yun Wang, Yiping Peng, and Xiangang Li. 2019. Learning alignment for multimodal emotion recognition from speech. *CoRR*, abs/1909.05645.

Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. 2019. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE.

Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. 2021. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 492–498. IEEE.

Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei. 2021. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. *arXiv preprint arXiv:2102.01547*.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. 2002. The htk book. *Cambridge university engineering department*, 3(175):12.

Binbin Zhang, Di Wu, Zhuoyuan Yao, Xiong Wang, Fan Yu, Chao Yang, Liyong Guo, Yaguang Hu, Lei Xie, and Xin Lei. 2020. Unified streaming and non-streaming two-pass end-to-end model for speech recognition. *arXiv preprint arXiv:2012.05481*.

Chengqi Zhao, Mingxuan Wang, Qianqian Dong, Rong Ye, and Lei Li. 2021. NeurST: Neural speech translation toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 55–62, Online. Association for Computational Linguistics.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, Jiahong Yuan, Kenneth Church, and Liang Huang. 2020. Fluent and low-latency simultaneous speech-to-speech translation with self-adaptive training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3928–3937.

A TTS Examples

We use the following sentences as the MOS evaluation test set in Table 7.

- 早上好，今天是2020/10/29，最低温度是-3°C。
- 你好，我的编号是37249，很高兴为您服务。
- 我们公司有37249个人。
- 我出生于2005年10月8日。
- 我们习惯在12:30吃中午饭。
- 只要有超过3/4的人投票同意，你就会成为我们的新班长。
- 我要买一只价值999.9元的手表。
- 我的手机号是18544139121，欢迎来电。
- 明天有62%的概率降雨。
- 手表厂有五种好产品。
- 跑马场有五百匹很勇敢的千里马。
- 有一天，我看到了一栋楼，我顿感不妙，因为我看不清里面有没有人。
- 史小姐拿着小雨伞去找她的老保姆了。
- 不要相信这个老奶奶说的话，她一点儿也不好。