Carnegie Mellon University
Language Technologies Institute

ESPnet: End-to-End Speech Processing Toolkit

ESPnet

Presenter: Jiatong Shi

Carnegie Mellon University

Note: Most slides are from Prof. Shinji Watanabe and main developers of ESPnet Team

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding
- More in ESPnet

ESPnet ESPnet, launched in December 2017

- **The Founder**

Shinji Watanabe

Our initial report at Interspeech 2018

ESPnet: End-to-End Speech Processing Toolkit

Shinji Watanabe[1], *Takaaki Hori*[2], *Shigeki Karita*[3], *Tomoki Hayashi*[4], *Jiro Nishitoba*[5], *Yuya Unno*[6], *Nelson Enrique Yalta Soplin*[7], *Jahn Heymann*[8], *Matthew Wiesner*[1], *Nanxin Chen*[1], *Adithya Renduchintala*[1], *Tsubasa Ochiai*[9],

[1]Johns Hopkins University, [2]Mitsubishi Electric Research Laboratories, [3]NTT Communication Science Laboratories, [4]Nagoya University, [5]Retrieva, Inc., [6]Preferred Networks, Inc., [7]Waseda University, [8]Paderborn University, [9]Doshisha University

shinjiw@jhu.edu

Abstract

This paper introduces a new open source platform for end-to-end speech processing named ESPnet. ESPnet mainly focuses on end-to-end automatic speech recognition (ASR), and adopts widely-used dynamic neural network toolkits, Chainer and PyTorch, as a main deep learning engine. ESPnet also follows the Kaldi ASR toolkit style for data processing, feature extraction/format, and recipes to provide a complete setup for speech

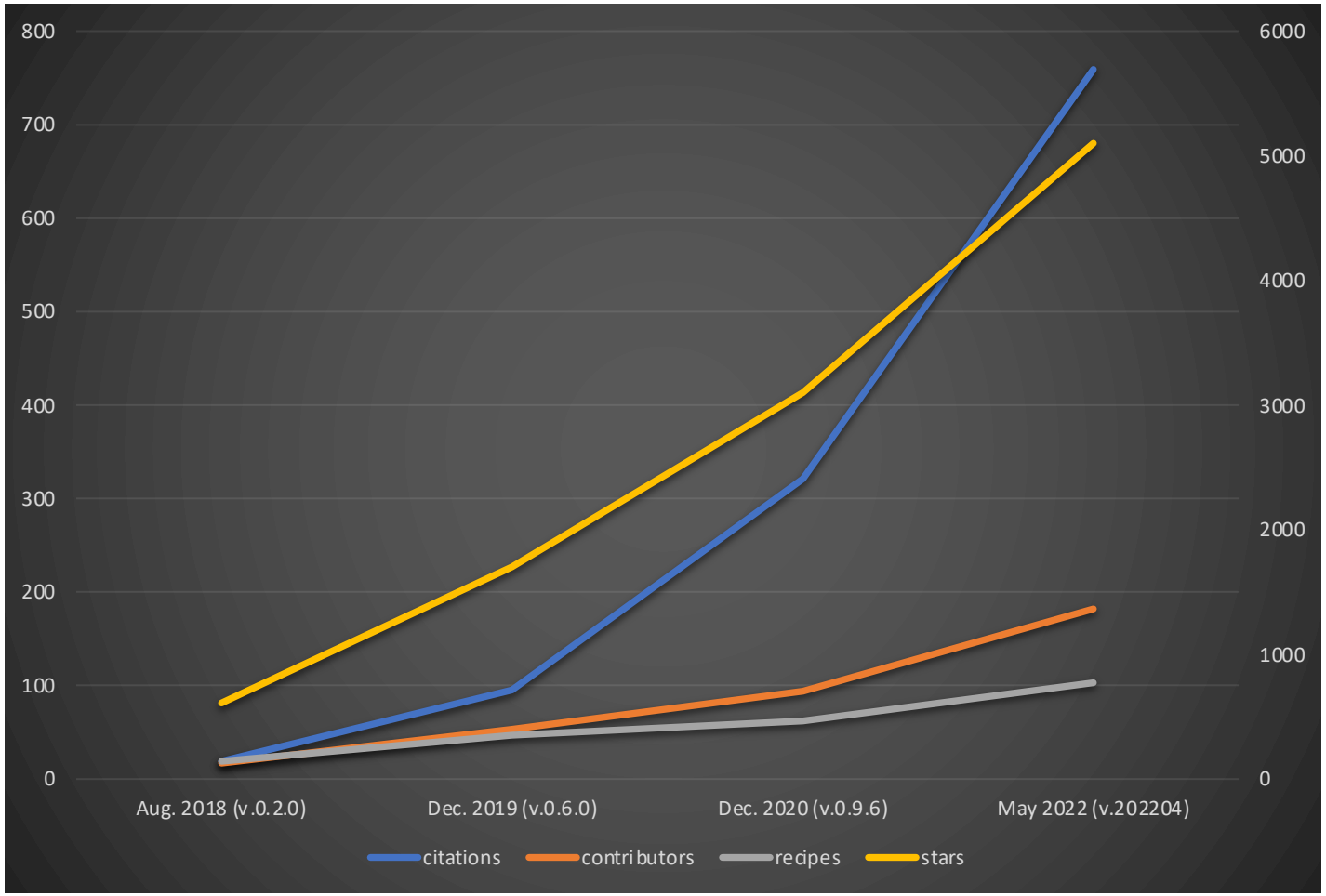
network [13, 14, 15, 16]. Attention-based methods use an attention mechanism to perform alignment between acoustic frames and recognized symbols, while CTC uses Markov assumptions to efficiently solve sequential problems by dynamic programming. ESPnet adopts hybrid CTC/attention end-to-end ASR [17], which effectively utilizes the advantages of both architectures in training and decoding. During training, we employ the multiobjective learning framework to improve robustness on irregular alignments and achieve fast convergence. During de-

ESPnet ESPnet, launched in December 2017

https://github.com/espnet/espnet

- **Open source** (Apache2.0) end-to-end speech processing toolkit
- Major concept: **Reproducibility**
  - Leverage our end-to-end ASR experience to the community
  - Accelerates end-to-end research for speech researchers
- Initially Chainer but later PyTorch based dynamic neural network toolkit as an engine
  - Easily develop novel neural network architecture
- Follows the **famous speech recognition toolkit, Kaldi, style**
  - Smoothly port ASR experiments from Kaldi to ESPnet with the common data processing, feature extraction/format
  - Recipes to provide a complete setup for speech processing experiments
- **The project is greatly accelerated in these four years**

Activity statistics (from 2018 to 2022)



- **Citations**, **contributors**, **recipes** (examples), and **stars** are all growing

i.e.,

- Developers have increasingly supported the development of ESPnet
- has been used in various research groups and contributed a lot to speech research activities
- ESPnet 5.2K stars (today)
  - Kaldi 11.7K stars
  - Nvidia NeMo 4.4K stars
  - SpeechBrain 4.1K stars
  - Google Lingvo 2.5K stars

Major change in the internal framework
From ESPnet1 to ESPnet2

ESPnet2: a new system for DNN training to extend our system from v.0.7.0

Mostly refactoring, but which enables major update to deal with

- **Distributed training**
- **On-the-fly feature extraction** from the raw waveform
- Improved the **scalability**

Improved software workflow by enhancing the **continuous integration**, **enriching documentation**, supporting the **docker**, **pip install**, and **model zoo** functions via HuggingFace.

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

What kind of research topics in speech research?

- Speech recognition
- Speech synthesis (Text to Speech (TTS))
- Speech…
- Spoken…

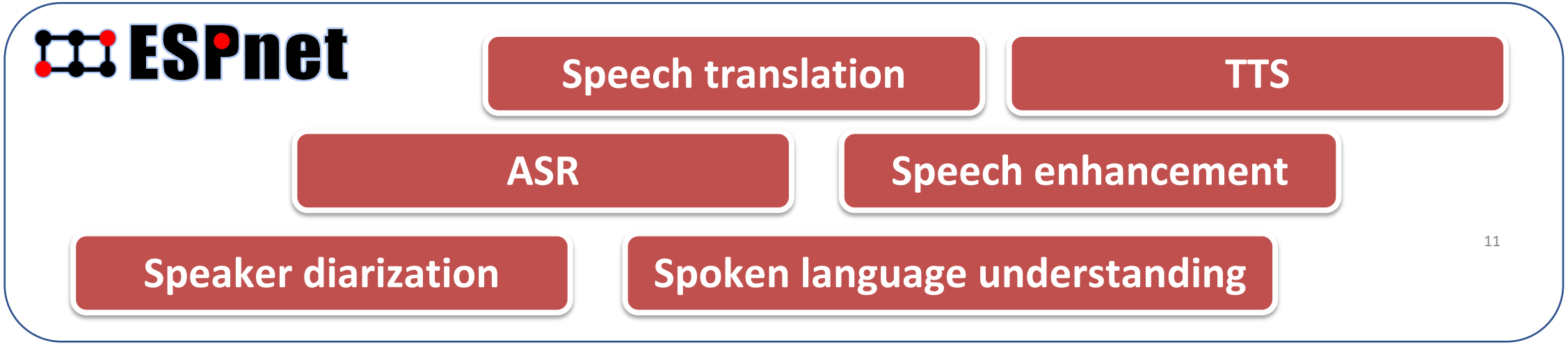
What kind of research topics in speech research?

- Speech recognition
- Speech synthesis
- Voice conversion
- Speaker recognition
- Language recognition
- Speech emotion recognition
- Speaker diarization
- Speech coding
- Speech perception
- Speech enhancement
- Microphone array processing
- Audio event classification and detection
- Speech separation
- Spoken language understanding
- Spoken dialogue systems
- Speech translation
- Multimodal processing
- Speech corpus

From the topic list in Interspeech

Broadened Applications

- ESPnet (**ASR+X**) covers the following topics complementally



- Why can one toolkit support such wide-ranges of applications?

What kind of research topics in speech research?

- **Speech recognition**
- **Speech synthesis**
- **Voice conversion**
- Speaker recognition
- Language recognition
- **Speech emotion recognition**
- **Speaker diarization**
- Speech coding
- Speech perception
- **Speech enhancement**
- **Microphone array processing**
- **Audio event classification and detection**
- **Speech separation**
- **Spoken language understanding**
- Spoken dialogue systems
- **Speech translation**
- **Multimodal processing**
- Speech corpus

Unified form → Unified software design

We design ESPnet by leveraging a **unified** mathematical **form** of **sequence (X) to sequence (Y) transformation f**

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

Unified view → Unified software design

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

Speech
Text
English Speech
Noisy Speech

ESPnet

ESPnet: End-to-end speech processing toolkit

$$f(\cdot)$$

Text
Speech
German Text
Clean Speech

Unified view → Unified software design

Speech recognition

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

ESPnet

ESPnet: End-to-end speech processing toolkit

$$f(\cdot)$$

Speech
Text
English Speech
Noisy Speech

Text
Speech
German Text
Clean Speech

Unified view → Unified software design

TTS

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

ESPnet

ESPnet: End-to-end speech processing toolkit

Speech
Text
English Speech
Noisy Speech

Text
Speech
German Text
Clean Speech

$$f(\cdot)$$

Unified view → Unified software design

Speech translation

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

ESPnet

ESPnet: End-to-end speech processing toolkit

$$f(\cdot)$$

Speech
Text
English Speech
Noisy Speech

Text
Speech
German Text
Clean Speech

Unified view → Unified software design

Speech enhancement

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

ESPnet

ESPnet: End-to-end speech processing toolkit

Speech
Text
English Speech
Noisy Speech

Text
Speech
German Text
Clean Speech

$$f(\cdot)$$

Unified form → Unified software design

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

ESPnet

ESPnet: End-to-end speech processing toolkit

$$f(\cdot)$$

CTC
Attention
Joint
CTC/Attention
RNN-T
Transformer, etc.

Unified form → Unified software design

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

ESPnet

ESPnet: End-to-end speech processing toolkit

- Many speech processing applications can be **unified** based on seq2seq
- **Nemo, Fairseq, Lingvo, Espresso, SpeechBrain, Asteroid** and other toolkits also fully make use of these functions
- We are closely collaborating/interacting with them

Today’s talk

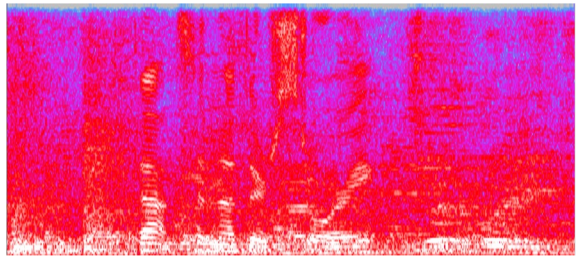
- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - RNN-transducer
    - Non-autoregressive modeling
- Text to speech (TTS)
- Voice conversion
- Speech translation
- Speech enhancement

Automatic speech recognition (ASR)

- Mapping **speech** sequence to ***character*** sequence

$$X = (x(l) \in \mathbb{Z} | l = 1, \ldots, L)$$

$$L = 43263$$



$$X = (\mathbf{x}_t \in \mathbb{R}^D | t = 1, \ldots, T)$$

$$T = 268$$

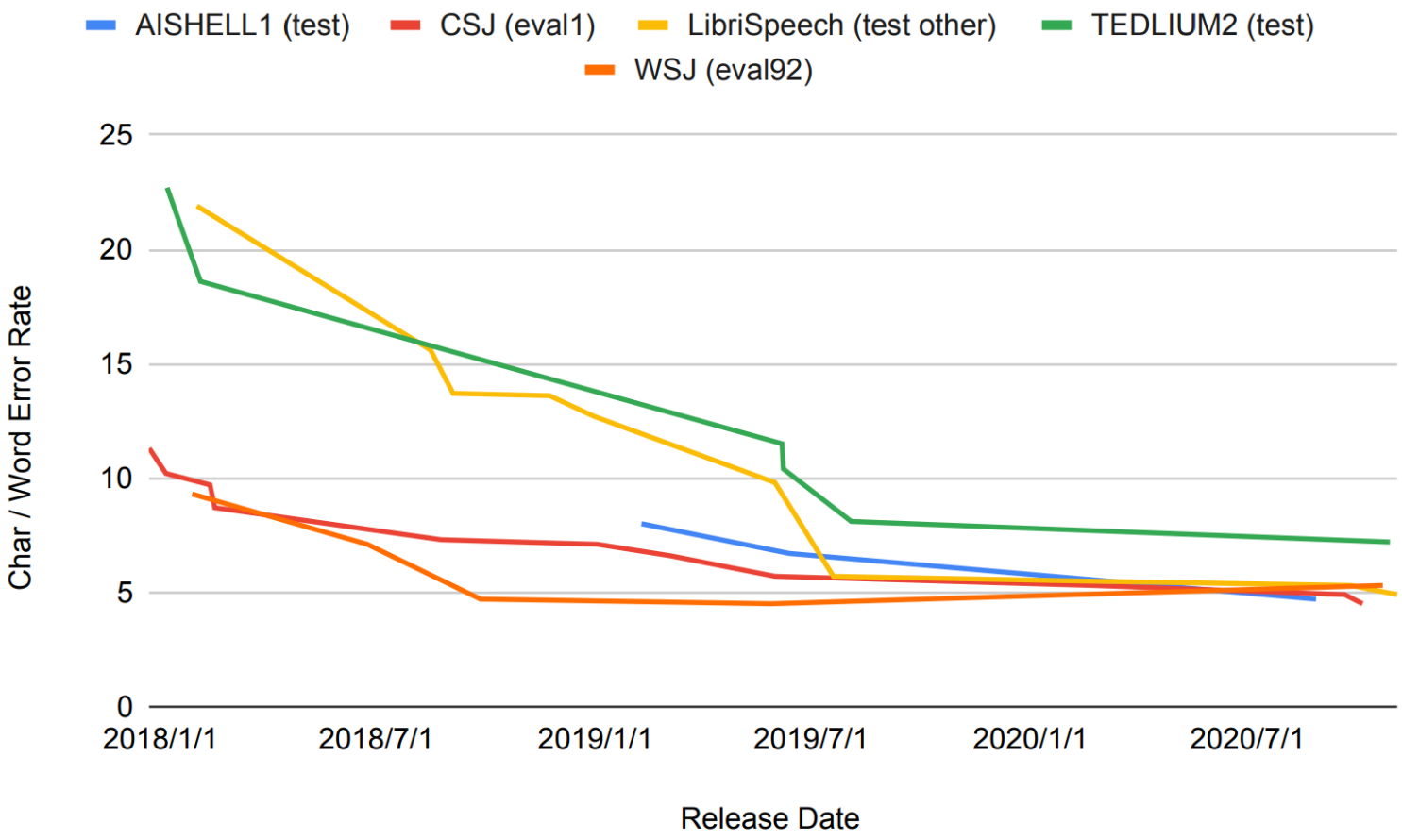
"That's another story"

$$W = (\mathbf{w}_n \in \mathcal{V} | n = 1, \ldots, N)$$

$$N = 18$$

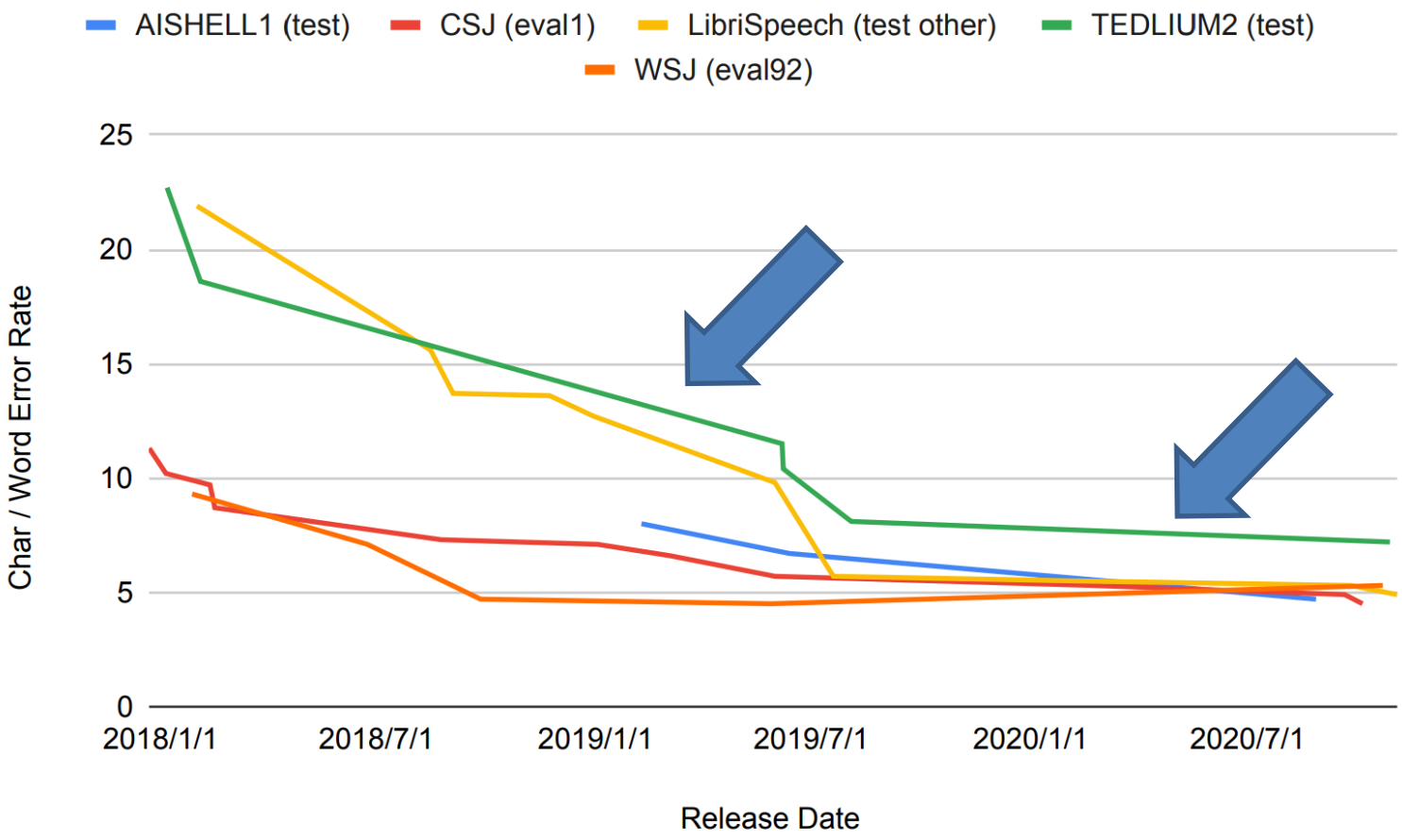
Maintaining state-of-the-art performance in ASR

Error rate:
Lower is better

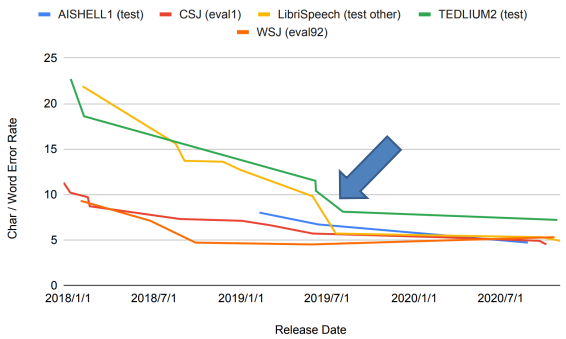


Maintaining state-of-the-art performance in ASR

Error rate:
Lower is better



ESPnet Transformer



A COMPARATIVE STUDY ON TRANSFORMER VS RNN IN SPEECH APPLICATIONS

Shigeki Karita[1],
(Alphabetical Order) *Nanxin Chen*[3], *Tomoki Hayashi*[5,6], *Takaaki Hori*[7], *Hirofumi Inaguma*[8], *Ziyan Jiang*[3], *Masao Someki*[5], *Nelson Enrique Yalta Soplin*[2], *Ryuichi Yamamoto*[4], *Xiaofei Wang*[3], *Shinji Watanabe*[3], *Takenori Yoshimura*[5,6], *Wangyou Zhang*[9]
[1]NTT Communication Science Laboratories, [2]Waseda University, [3]Johns Hopkins University, [4]LINE Corporation, [5]Nagoya University, [6]Human Dataware Lab. Co., Ltd., [7]Mitsubishi Electric Research Laboratories, [8]Kyoto University, [9]Shanghai Jiao Tong University

ABSTRACT

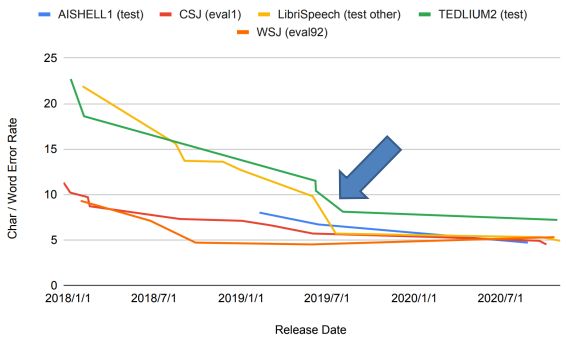
Sequence-to-sequence models have been widely used in end-to-end speech processing, for example, automatic speech recognition (ASR), speech translation (ST), and text-to-speech (TTS). This paper focuses on an emergent sequence-to-sequence model called Transformer, which achieves state-of-the-art performance in neural machine translation and other natural language processing applications. We undertook intensive studies in which we experimentally compared and analyzed Transformer and conventional recurrent

In our speech application experiments, we investigate several aspects of Transformer and RNN-based systems. For example, we measure the word/character/regression error from the ground truth, training curve, and scalability for multiple GPUs.

The contributions of this work are:

- We conduct a larges-scale comparative study on Transformer and RNN with significant performance gains especially for the ASR related tasks.
- We explain our training tips for Transformer in speech applica-

ESPnet Transformer



A COMPARATIVE STUDY ON TRANSFORMER VS RNN IN SPEECH APPLICATIONS

Shigeki Karita[1],
(Alphabetical Order) *Nanxin Chen*[3], *Tomoki Hayashi*[5,6], *Takaaki Hori*[7], *Hirofumi Inaguma*[8], *Ziyan Jiang*[3], *Masao Someki*[5], *Nelson Enrique Yalta Soplin*[2], *Ryuichi Yamamoto*[4], *Xiaofei Wang*[3], *Shinji Watanabe*[3], *Takenori Yoshimura*[5,6], *Wangyou Zhang*[9]
[1]NTT Communication Science Laboratories, [2]Waseda University, [3]Johns Hopkins University, [4]LINE Corporation, [5]Nagoya University, [6]Human Dataware Lab. Co., Ltd., [7]Mitsubishi Electric Research Laboratories, [8]Kyoto University, [9]Shanghai Jiao Tong University

ABSTRACT

Sequence-to-sequence models have been wid
end speech processing, for example, automatic
(ASR), speech translation (ST), and text-to-
paper focuses on an emergent sequence-to-sec
Transformer, which achieves state-of-the-art pe
machine translation and other natural language processing applications. We undertook intensive studies in which we experimentally compared and analyzed Transformer and conventional recurrent

and RNN with significant performance gains especially for the ASR related tasks.

- We explain our training tips for Transformer in speech applications: ASR, TTS and ST

- One of the first success in the speech areas
- The performance was boosted

Transformer boosted the performance

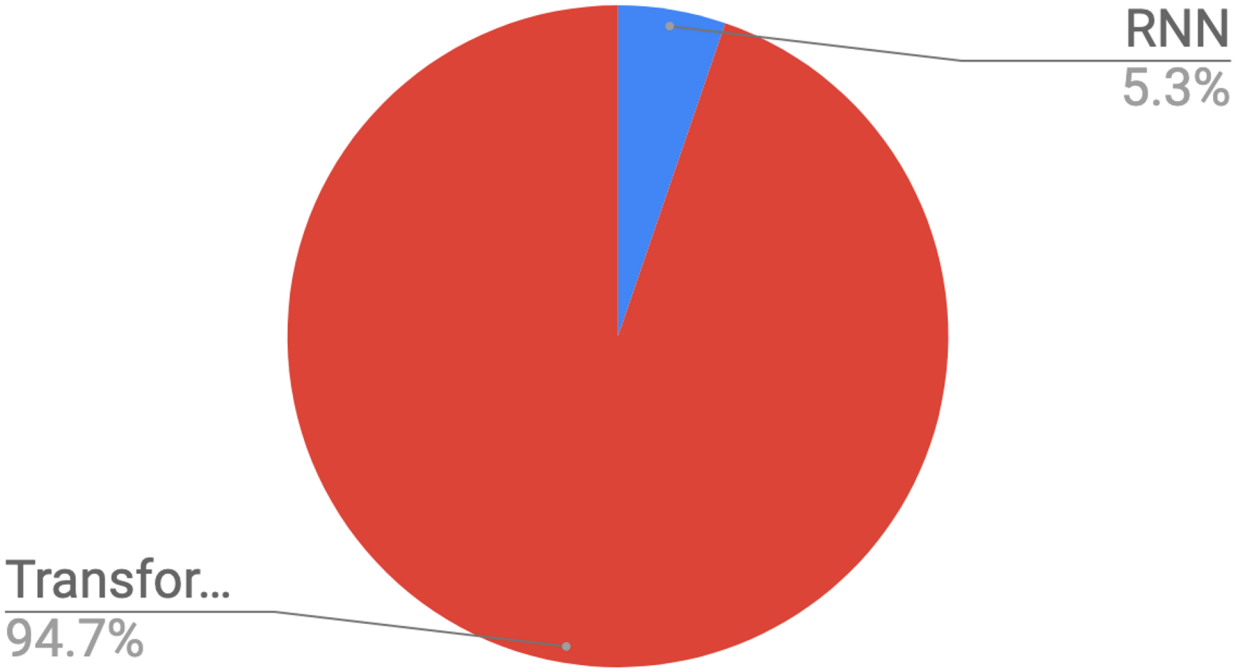
- Improve the performance from RNN with **13** ASR tasks among 15 tasks

dataset	token	error	Kaldi	Our RNN	Our Transformer
AISHELL	char	CER	N/A / 7.4	6.8 / 8.0	**6.0 / 6.7**
AURORA4	char	WER	(*) 3.6 / 7.7 / 10.0 / 22.3	3.5 / 6.4 / 5.1 / 12.3	**3.3 / 6.0 / 4.5 / 10.6**
CSJ	char	CER	(*) 7.5 / 6.3 / 6.9	6.6 / 4.8 / 5.0	**5.7 / 4.1 / 4.5**
CHiME4	char	WER	**6.8 / 5.6 / 12.1 / 11.4**	9.5 / 8.9 / 18.3 / 16.6	9.6 / 8.2 / 15.7 / 14.5
CHiME5	char	WER	**47.9 / 81.3**	59.3 / 88.1	60.2 / 87.1
Fisher-CALLHOME Spanish	char	WER	N/A	27.9 / 27.8 / 25.4 / 47.2 / 47.9	**27.0 / 26.3 / 24.4 / 45.3 / 46.2**
HKUST	char	CER	23.7	27.4	**23.5**
JSUT	char	CER	N/A	20.6	**18.7**
LibriSpeech	BPE	WER	3.9 / 10.4 / 4.3 / 10.8	3.1 / 9.9 / 3.3 / 10.8	**2.2 / 5.6 / 2.6 / 5.7**
REVERB	char	WER	18.2 / 19.9	24.1 / 27.2	**15.5 / 19.0**
SWITCHBOARD	BPE	WER	**18.1 / 8.8**	28.5 / 15.6	26.0 / 14.0
TED-LIUM2	BPE	WER	**9.0** / 9.0	11.2 / 11.0	9.3 / **8.1**
TED-LIUM3	BPE	WER	**6.2 / 6.8**	14.3 / 15.0	9.7 / 8.0
VoxForge	char	CER	N/A	12.9 / 12.6	**9.4 / 9.1**
WSJ	char	WER	**4.3 / 2.3**	7.0 / 4.7	6.8 / 4.4

Transformer boosted the performance

- Improve the performance from RNN with **13** ASR tasks among 15 tasks

RNN vs. Transformer



dataset	Our Transformer
AISHELL	**6.0 / 6.7**
AURORA4	**3.3 / 6.0 / 4.5 / 10.6**
CSJ	**5.7 / 4.1 / 4.5**
CHiME4	9.6 / 8.2 / 15.7 / 14.5
CHiME5	60.2 / 87.1
Fisher-CALLHOME Spanish	**7.0 / 26.3 / 24.4 / 45.3 / 46.2**
HKUST	**23.5**
JSUT	**18.7**
LibriSpeech	**2.2 / 5.6 / 2.6 / 5.7**
REVERB	**15.5 / 19.0**
SWITCHBOARD	26.0 / 14.0
TED-LIUM2	9.3 / **8.1**
TED-LIUM3	9.7 / 8.0
VoxForge	**9.4 / 9.1**
WSJ	6.8 / 4.4

Experiments (~ **1000** hours) Librispeech at April 2019

Toolkit	dev_clean	dev_other	test_clean	test_other
Facebook wav2letter++	3.1	10.1	3.4	11.2
RWTH RASR	2.9	8.8	3.1	9.8
Nvidia Jasper	2.6	7.6	2.8	7.8
Google SpecAug.	N/A	N/A	**2.5**	5.8

- Very sensational results by Google

Experiments (~ **1000** hours) Librispeech at August 2019

Toolkit	dev_clean	dev_other	test_clean	test_other
Facebook wav2letter++	3.1	10.1	3.4	11.2
RWTH RASR	2.9	8.8	3.1	9.8
Nvidia Jasper	2.6	7.6	2.8	7.8
Google SpecAug.	N/A	N/A	**2.5**	5.8
ESPnet	**2.2**	**5.6**	2.6	**5.7**

- **Reached Google's best performance by community-driven efforts**

GAFAM

GAFAM
ESPnet

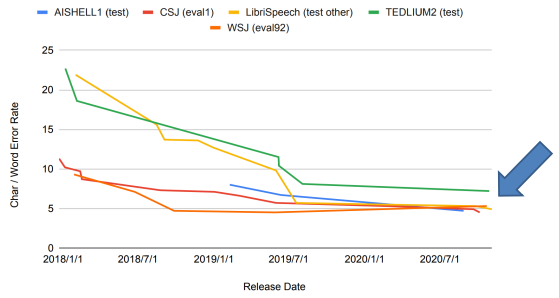
Experiments (~ **1000** hours) in August 2019 Librispeech

Toolkit	dev_clean	dev_other	test_clean	test_other
Facebook wav2letter++	3.1	10.1	3.4	11.2
RWTH RASR	2.9	8.8	3.1	9.8
Nvidia Jasper	2.6	7.6	2.8	7.8
Google SpecAug.	N/A	N/A	**2.5**	5.8
ESPnet	**2.2**	**5.6**	2.6	**5.7**

Experiments (~ **1000** hours) in March 2020 Librispeech

Toolkit	dev_clean	dev_other	test_clean	test_other
Facebook wav2letter++	3.1	10.1	3.4	11.2
RWTH RASR	2.9	8.8	3.1	9.8
Nvidia Jasper	2.6	7.6	2.8	7.8
Google SpecAug.	N/A	N/A	2.5	5.8
ESPnet	2.2	5.6	2.6	5.7
Google **Conformer**	N/A	N/A	**1.9**	**3.9**

ESPnet Conformer



RECENT DEVELOPMENTS ON ESPNET TOOLKIT BOOSTED BY CONFORMER

Pengcheng Guo[1,4], *Florian Boyer*[2,3], *Xuankai Chang*[4], *Tomoki Hayashi*[5], *Yosuke Higuchi*[6]
Hirofumi Inaguma[7], *Naoyuki Kamo*[8], *Chenda Li*[9], *Daniel Garcia-Romero*[4], *Jiatong Shi*[4]
Jing Shi[4,10], *Shinji Watanabe*[4], *Kun Wei*[1], *Wangyou Zhang*[9], *Yuekai Zhang*[4]

[1]Northwestern Polytechnical University, [2]LaBRI, University of Bordeaux, [3] Airudit
[4]Johns Hopkins University, [5]Human Dataware Lab. Co., Ltd.
[6]Waseda University, [7]Kyoto University, [8]NTT Corporation [9]Shanghai Jiao Tong University
[10]Institute of Automation, Chinese Academy of Sciences

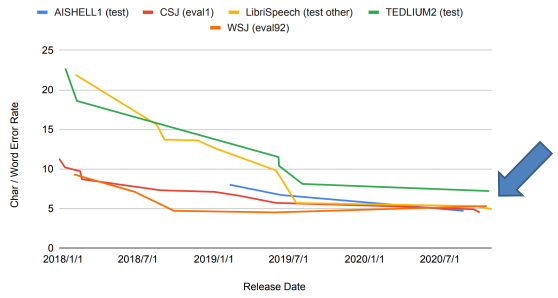
ABSTRACT

In this study, we present recent developments on ESPnet: End-to-End Speech Processing toolkit, which mainly involves a recently proposed architecture called Conformer, Convolution-augmented Transformer. This paper shows the results for a wide range of end-to-end speech processing applications, such as automatic speech recognition (ASR), speech translations (ST), speech separation (SS) and text-to-speech (TTS). Our experiments reveal various training tips and significant performance benefits obtained with the Conformer on different tasks. These results are competitive or even outperform the current state-of-art Transformer models. We are preparing to release all-in-one recipes using open source and publicly available corpora for all the above tasks with pre-trained models of publicly available corpora and try our best to share the practical guides (e.g., learning rate, hyper-parameters, network structure) on the use of Conformer. We also prepare to release the reproducible recipes and state-of-the-art setups to the community to succeed our exciting outcomes.

The contributions of this study include:

- We extend the Conformer architecture to various end-to-end speech processing applications and conduct comparative experiments with Transformer.
- We share our practical guides for the training of Conformer, like learning rate, kernel size of Conformer block, and model architectures, etc.

ESPnet Conformer



RECENT DEVELOPMENTS ON ESPNET TOOLKIT BOOSTED BY CONFORMER

Pengcheng Guo[1,4], *Florian Boyer*[2,3], *Xuankai Chang*[4], *Tomoki Hayashi*[5], *Yosuke Higuchi*[6]
Hirofumi Inaguma[7], *Naoyuki Kamo*[8], *Chenda Li*[9], *Daniel Garcia-Romero*[4], *Jiatong Shi*[4]
Jing Shi[4,10], *Shinji Watanabe*[4], *Kun Wei*[1], *Wangyou Zhang*[9], *Yuekai Zhang*[4]

[1]Northwestern Polytechnical University, [2]LaBRI, University of Bordeaux, [3] Airudit
[4]Johns Hopkins University, [5]Human Dataware Lab. Co., Ltd.
[6]Waseda University, [7]Kyoto University, [8]NTT Corporation [9]Shanghai Jiao Tong University
[10]Institute of Automation, Chinese Academy of Sciences

ABSTRACT

In this study, we present recent developments
End Speech Processing toolkit, which mainly
proposed architecture called Conformer, Co
Transformer. This paper shows the results for
to-end speech processing applications, such
recognition (ASR), speech translations (ST), sp
and text-to-speech (TTS). Our experiments r
ing tips and significant performance benefits obtained with the Conformer on different tasks. These results are competitive or even outperform the current state-of-art Transformer models. We are preparing to release all-in-one recipes using open source and publicly available corpora for all the above tasks with pre-trained

of publicly available corpora and try our best to share the practical

- We try to follow Google's conformer work
- Also apply conformer to ST, TTS, as well as ASR

periments with Transformer.

- We share our practical guides for the training of Conformer, like learning rate, kernel size of Conformer block, and model architectures, etc.

Experiments (~ **1000** hours) in October 2020 Librispeech

Toolkit	dev_clean	dev_other	test_clean	test_other
Facebook wav2letter++	3.1	10.1	3.4	11.2
RWTH RASR	2.9	8.8	3.1	9.8
Nvidia Jasper	2.6	7.6	2.8	7.8
Google SpecAug.	N/A	N/A	2.5	5.8
ESPnet	2.2	5.6	2.6	5.7
Google Conformer	N/A	N/A	**1.9**	**3.9**
ESPnet Conformer	**1.9**	4.6	2.1	4.7

We continue to work on catching up SOTA, but we still have a gap

→ We revisit several configurations **(with limited resources)**

Experiments (~ **1000** hours) in October 2020 Librispeech

Toolkit	dev_clean	dev_other	test_clean	test_other
Facebook wav2letter++	3.1	10.1	3.4	11.2
RWTH RASR	2.9	8.8	3.1	9.8
Nvidia Jasper	2.6	7.6	2.8	7.8
Google SpecAug.	N/A	N/A	2.5	5.8
ESPnet	2.2	5.6	2.6	5.7
Google Conformer	N/A	N/A	**1.9**	**3.9**
ESPnet Conformer	1.9	4.6	2.1	4.7
ESPnet Conformer (tuning)	**1.8**	**3.9**	2.0	4.2

Further gains based on various tuning (with limited resources)

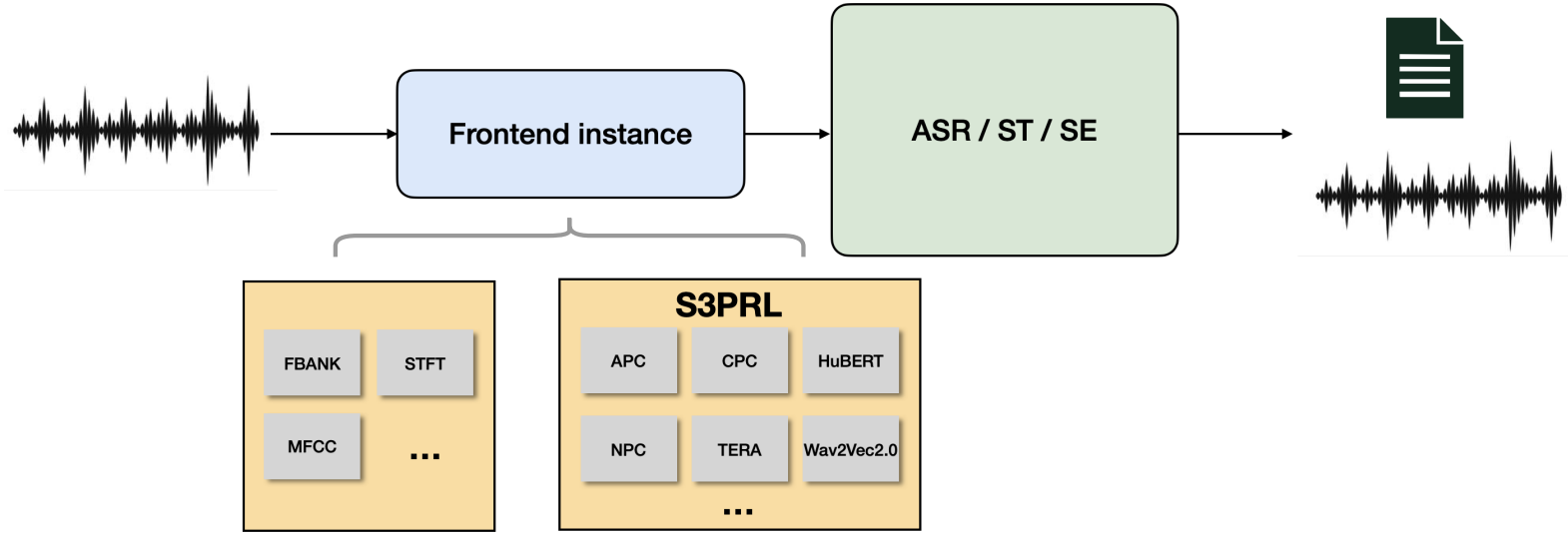
Support of self-supervised learning models

- SUPERB benchmark

  https://superbbenchmark.org/

Combine ESPnet with S3PRL [Yang+ 2021, Chang+ 2021]

- Self-supervised pretraining on speech data have achieved a lot of progress, like wav2vec2.0 [Baevski+ 2020], Hubert [Hsu+ 2021], etc.
- S3PRL* toolkit provides an integration of pretrained speech representation models and speech tasks, e.g., wav2vec2.0 + LSTM acoustic model for ASR.
- Support combine the pretrained models with advanced end-to-end speech processing models in a simple way.



*: https://github.com/s3prl/s3prl

ASR Experiments (80 hours English task)

- Word Error Rate (%) on WSJ corpus

Models	dev93	dev92
Kaldi Chain Model	4.3	2.3
ESPnet Conformer	6.6	4.4
ESPnet Conformer + wav2vec2.0	**2.8**	**1.8**
ESPnet Conformer + Hubert	3.1	**1.8**

Reach the state-of-the-art results.

ASR Experiments (960 hours English task)

- Word Error Rate (%) on Librispeech corpus

Toolkit	dev_clean	dev_other	test_clean	test_other
Google Conformer	2.1	4.3	1.9	3.9
ESPnet Conformer	1.9	4.6	2.1	4.7
Facebook wav2vec2.0 (60k LibriVox)	**1.6**	**3.0**	**1.8**	**3.3**
Facebook Hubert	1.7	3.0	1.9	3.5
ESPnet Conformer + wav2vec2.0	1.9	5.4	2.2	5.2
ESPnet Conformer + Hubert	**1.7**	**3.4**	**1.8**	**3.6**

Obtain further improvements with the help of self-supervised pretrained models.

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

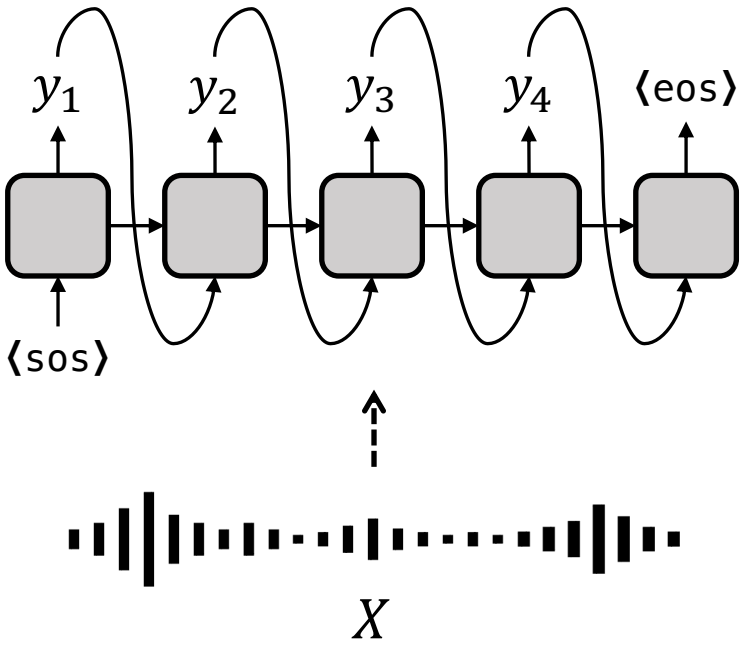
Introduction

- State-of-the-art end-to-end ASR models are mostly autoregressive
  - Inherently sequential, leading to slow decoding speed at inference
- Increasing interest in non-autoregressive (NAR) end-to-end ASR
  - Want to improve performance while keeping inference speed fast

	Accuracy	Inference speed
Autoregressive Model RNN-Transducer (RNN-T) [Graves+, 2013] Attention-based Sequence-to-Sequence [Bahdanau+, 2015]	😀	😩
Non-autoregressive Model Connectionist temporal classification (CTC) [Graves+, 2006]	😩	😀

Autoregressive sequence generation

+ **Accurate:** Each token prediction is conditioned on previously generated tokens
− **Slow:** Require incremental calculations for predicting a single token at a time



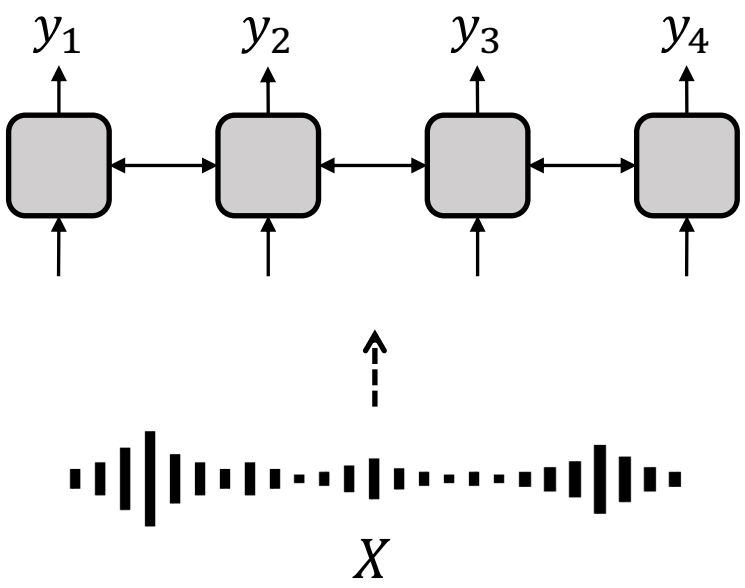
$$P(Y|X) = \prod_{l=1}^{L} P_{\mathrm{ar}}(y_l | y_{<l}, X)$$

Ex.)

- Attention-based encoder-decoder [Bahdanau+, 2015]
- RNN-Transducer [Graves+, 2013]

Non-autoregressive sequence generation

+ **Fast:** Multiple tokens can be generated in parallel
− **Inaccurate:** Difficult to capture contextual dependencies between output tokens



$$P(Y|X) = \prod_{l=1}^{L} P_{\mathrm{nar}}(y_l|X)$$

Ex.)

- Connectionist temporal classification [Graves+, 2006]

NAR models on ESPnet

A COMPARATIVE STUDY ON NON-AUTOREGRESSIVE MODELINGS FOR SPEECH-TO-TEXT GENERATION

Yosuke Higuchi[1], *Nanxin Chen*[2], *Yuya Fujita*[3], *Hirofumi Inaguma*[4], *Tatsuya Komatsu*[5], *Jaesong Lee*[6], *Jumon Nozaki*[4,5], *Tianzi Wang*[2], *Shinji Watanabe*[7]

[1]Waseda University, [2]Johns Hopkins University, [3]Yahoo Japan Corporation, [4]Kyoto University, [5]LINE Corporation, [6]Naver Corporation, [7]Carnegie Mellon University

ABSTRACT

Non-autoregressive (NAR) models simultaneously generate multiple outputs in a sequence, which significantly reduces the inference speed at the cost of accuracy drop compared to autoregressive baselines. Showing great potential for real-time applications, an increasing number of NAR models have been explored in different fields to mitigate the performance gap against AR models. In this work, we conduct a comparative study of various NAR modeling methods for end-to-end automatic speech recognition (ASR). Experiments are performed in the state-of-the-art setting using ESPnet. The results on various tasks provide interesting findings for developing an

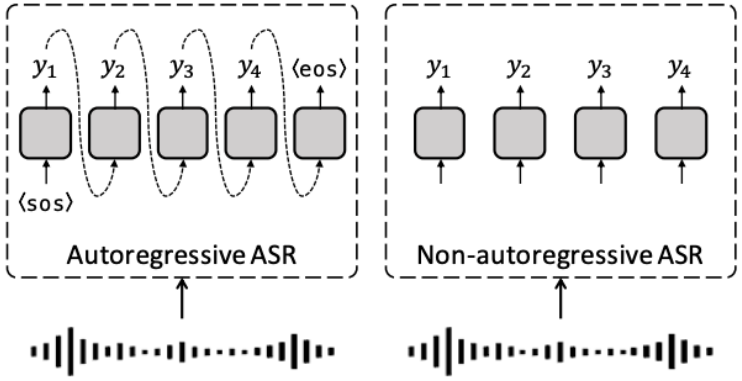


Fig. 1. Illustrations of autoregressive and non-autoregressive ASR.

Comparative study

- Comparable performance to autoregressive models (i.e., Seq-to-Seq and Transducer) while achieving significantly fast inference speed

Model		#iter	Inference speed		LS-100 (WER)				TED2 (WER)		CSJ-APS (CER)		
			RTF	Speedup	dev clean	dev other	test clean	test other	dev	test	eval1	eval2	eval3
AR	CTC/attention	L	0.341	1.00×	6.8	18.8	7.4	19.0	11.6	8.7	5.4	4.0	9.8
	+ beam-search	$> L$	3.419	0.10×	**6.3**	**18.2**	**6.8**	**18.5**	10.4	8.4	**5.1**	**3.8**	**9.0**
	Transducer	L	0.069	4.94×	7.3	19.9	7.6	19.9	9.6	9.2	6.3	4.5	10.6
	+ beam-search	$> L$	0.234	1.46×	6.4	18.8	**6.8**	18.9	**8.6**	**8.2**	5.2	4.1	10.0
NAR	CTC	1	0.059	5.78×	7.4	20.5	7.8	20.8	8.9	8.6	5.4	4.0	9.6
	Mask-CTC	10	0.063	5.41×	7.2	20.3	7.5	20.6	8.9	8.5	5.6	4.0	9.6
	Improved Mask-CTC	5	0.072	4.74×	7.0	19.8	7.3	20.2	8.8	8.3	5.5	4.0	9.5
	Align-Denoise	1	0.073	4.67×	8.0	22.3	8.4	22.5	9.0	8.7	5.4	**3.7**	**9.1**
	Intermediate CTC	1	0.059	5.78×	6.9	19.7	7.1	20.2	**8.5**	8.3	5.6	4.1	9.8
	Self-conditioned CTC	1	0.059	5.78×	**6.6**	**19.4**	**6.9**	**19.7**	8.7	**8.0**	**5.3**	**3.7**	**9.1**
	KERMIT	$\simeq \log_2(L)$	0.361	1.06×	7.1	19.7	7.4	20.2	9.1	8.2	5.4	**3.7**	9.5
	Insertion Transformer	$\simeq \log_2(L)$	0.083	4.11×	16.0	27.3	16.2	27.4	–	–	–	–	–
	CIF-NA†	1	0.073	4.67×	15.4	34.0	15.7	34.6	–	–	–	–	–

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

Introduction

- https://shinjiwlab.github.io/cmu_multilingual_speech/

Note that this is a tentative information

We’re working on enriching this DB

Multi-Lingual ASR in ESPnet

1. A standard pipeline to build an E2E-ASR;
2. Generalize to most languages;
3. Minimum efforts to learn;
4. Some advanced techniques;
5. Easy to make the work publicly available;
   a. Results
   b. Pre-trained models

General resource in ESPnet

1. Open-source
2. Standard recipes to follow
   - Most efforts would be: 1) Data 2) Hyperparameter
3. Benefit from advanced techniques
   - Self-supervised learning
     i. Wav2Vec, XLS-R, HuBERT, WavLM, etc.
   - Transfer-learning
   - Feature fusion
4. Clear instructions (notebook)
5. Rich help sources (Github issues)

Feature fusion for low-resource ASR

Combining Spectral and Self-Supervised Features for Low Resource Speech Recognition and Translation

Dan Berrebbi[1], *Jiatong Shi*[1], *Brian Yan*[1], *Osbel López-Francisco*[2], *Jonathan D. Amith*[3], *Shinji Watanabe*[1]

[1]Language Technologies Institute, Carnegie Mellon University
[2]Universidad Nacional Autónoma de México, Iztacala
[3]Dept. of Anthropology, Gettysburg College

dberrebb@andrew.cmu.edu, jiatongs@andrew.cmu.edu

Feature fusion for low-resource ASR (Cont'd)

Table 1: *ASR and ST results over models described in Sec. 4.2. The two first experiments are our FBANK and SSL baselines. The following lines are the proposed Linear, Convolutional, co-Attention, and Mixture of Experts models.*

	CER ↓		BLEU ↑
Exp	Totonac	Arabic	Mboshi-French
Base	17.2	15.4	10.9
SSL	14.2	8.1	10.6
Linear	14.0	6.6	**11.6**
Conv.	13.9	7.2	11.3
co-Att.	**13.4**	**5.4**	10.9
MoE	13.7	6.2	11.2

Potential external collaborators

- Existing Recipes (May 2022)
  - ASR: 60 out of 103 recipes
  - ~ 100 languages
- Open to collaborations
  - 29 corpora, more to be added

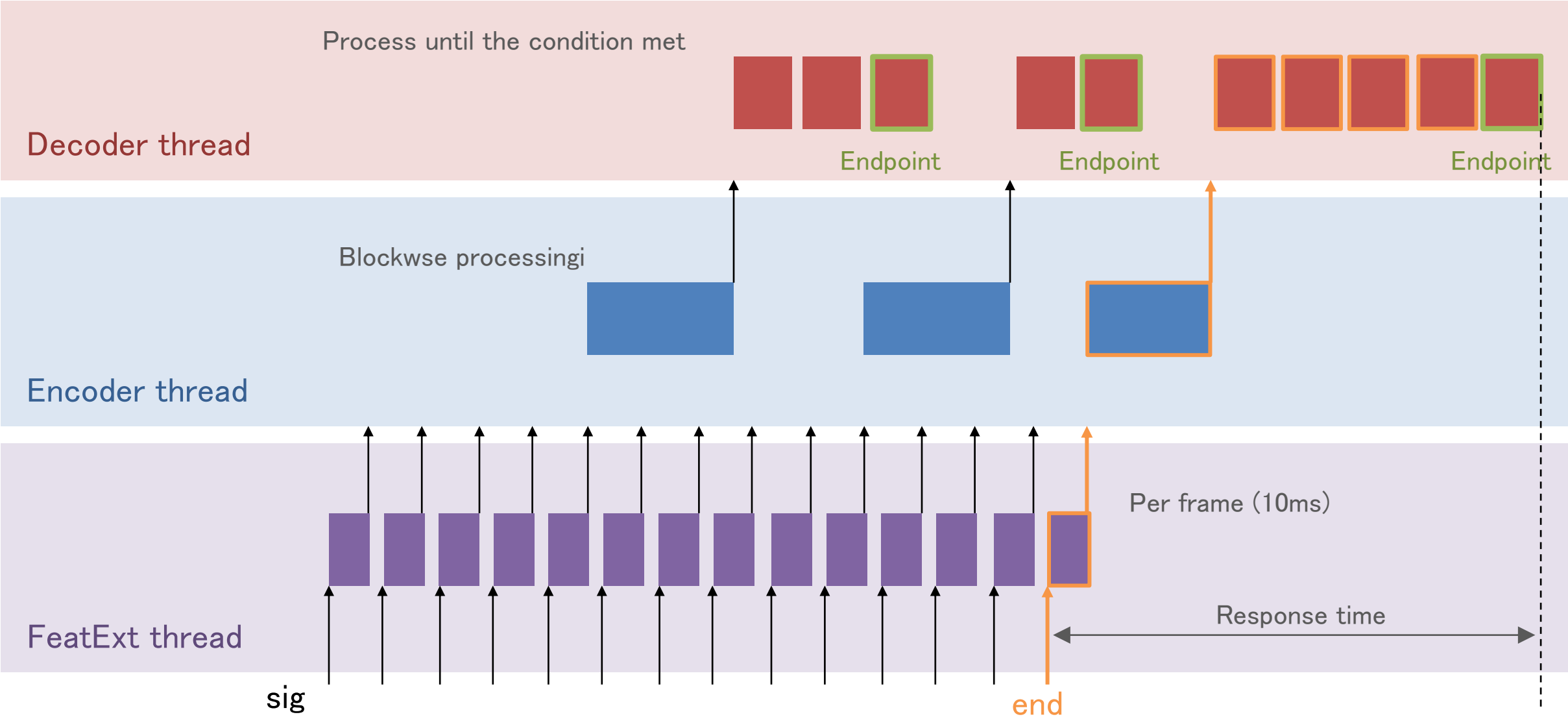
NISP (indian accent ASR)	https://arxiv.org/pdf/2007.06021v1.pdf
aishell2	https://github.com/espnet/espnet/tree/master/egs
hub4_spanish	https://github.com/espnet/espnet/tree/master/egs
openasr20	https://sat.nist.gov/openasr20
vystadial recipe	http://www.openslr.org/6/
thuyg-20	http://www.openslr.org/22/
Iban	http://www.openslr.org/24/
ALFFA	http://www.openslr.org/25/
neroico	http://www.openslr.org/39/
tunisian_msa	https://www.openslr.org/46/
sinhala_openslr52	http://www.openslr.org/52/
nepali_openslr54	http://www.openslr.org/54/
acceted_french_openslr56	http://www.openslr.org/57/
pansori_ted_x_kr	http://www.openslr.org/58/
parlament_parla	http://www.openslr.org/59/
tedx_spanish	http://www.openslr.org/67/
magicdata_mandarin_read_speech	http://www.openslr.org/68/
Puerto Rican Spanish	http://www.openslr.org/74/
Gujarati	http://openslr.org/78/
kannada	https://openslr.org/79/
russian_librispeech	http://www.openslr.org/96/
ksc	http://www.openslr.org/102/
nicolingua_0003_african_radio	http://www.openslr.org/105/
nicolingua_0004_african_va	http://www.openslr.org/106/
samromur_2105	http://www.openslr.org/112/
seoul_corpus	http://www.openslr.org/113/
golos	http://www.openslr.org/114/
LDCIL ()	https://data.ldcil.org/index.php?route=common/home
commonvoice Guaraní	https://commonvoice.mozilla.org/en/datasets
Sichuan Dialect Scripted Speech Corpus	https://magichub.com/datasets/sichuan-dialect-scripted-speech-corpus-daily-use-sentence/

https://docs.google.com/spreadsheets/d/1JUFoIsP7sPteP212nB7g48YcGhiyj7V-fuV0XlZHJmc/edit?usp=sharing

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

Streaming Transformer



Streaming Transformer

- We propose a novel transformer encoder-decoder for streaming processing
- We use the **blockwise** processing
  - Control the performance and latency with the block size
- Blockwise synchronous (BS) beam search [Tsunoo2020]
  - Block boundary detection (BBD) by computing reliability score for each hypothesis

Implementation in ESPnet

- Tutorials
  - Training
  - Decoding
- Demo (notebook)
  - https://github.com/espnet/notebook/blob/master/espnet2_streaming_asr_demo.ipynb
- Applications in the other speech processing
  - Speech translation
  - Spoken language understanding
- ONNX and Quantization
  - https://github.com/Masao-Someki/espnet_onnx

Streaming ASR to streaming SLU and ST

Blockwise Streaming Transformer for Spoken Language Understanding and Simultaneous Speech Translation

Keqi Deng[1], *Shinji Watanabe*[2], *Jiatong Shi*[2], *Siddhant Arora*[2]

[1]University of Chinese Academy of Sciences, China
[2]Carnegie Mellon University, USA

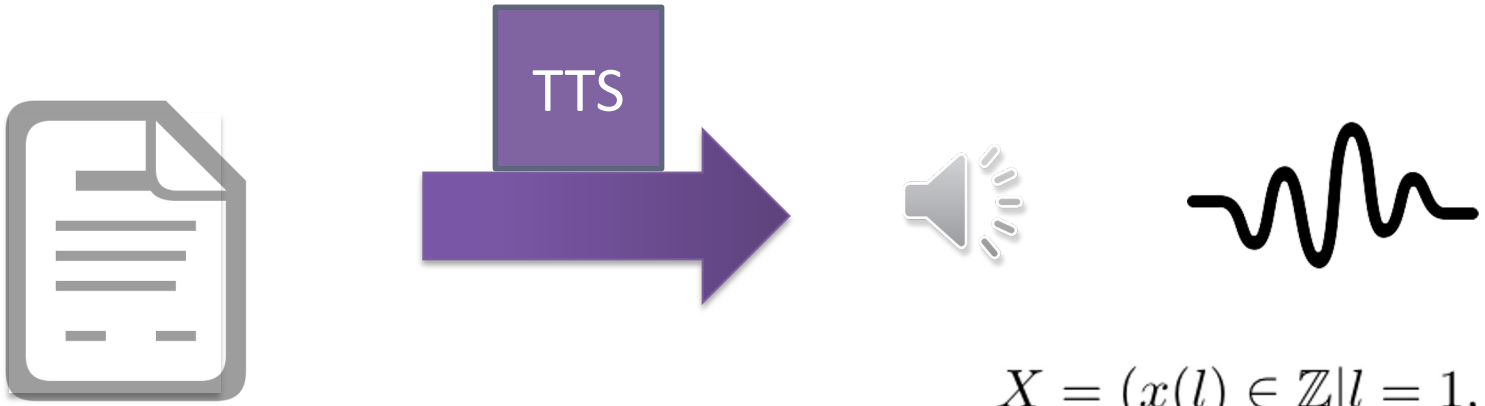
dengkeqi20@mails.ucas.ac.cn, shinjiw@cmu.edu, {jiatongs, siddhana}@cs.cmu.edu

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

Text to speech (TTS)

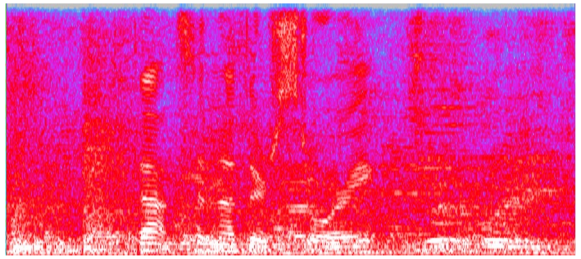
- Mapping **character** sequence to **speech** sequence



"That's another story"

$$W = (\mathbf{w}_n \in \mathcal{V} | n = 1, \ldots, N)$$
$$N = 18$$

$$X = (x(l) \in \mathbb{Z} | l = 1, \ldots, L)$$
$$L = 43263$$



$$X = (\mathbf{x}_t \in \mathbb{R}^D | t = 1, \ldots, T)$$
$$T = 268$$

ESPnet2-TTS

ESPNET2-TTS: EXTENDING THE EDGE OF TTS RESEARCH

Tomoki Hayashi[1,2], *Ryuichi Yamamoto*[3], *Takenori Yoshimura*[4], *Peter Wu*[5], *Jiatong Shi*[5], *Takaaki Saeki*[6], *Yooncheol Ju*[7], *Yusuke Yasuda*[2], *Shinnosuke Takamichi*[6], *Shinji Watanabe*[5]
[1]Human Dataware Lab. Co., Ltd., [2]Nagoya University, [3]LINE Corp., [4]Nagoya Institute of Technology, [5]Carnegie Mellon University, [6]The University of Tokyo, [7]AIRS Company, Hyundai Motor Group

v1 [cs.CL] 15 Oct 2021

ABSTRACT

This paper describes ESPnet2-TTS, an end-to-end text-to-speech (E2E-TTS) toolkit. ESPnet2-TTS extends our earlier version, ESPnet-TTS, by adding many new features, including: on-the-fly flexible pre-processing, joint training with neural vocoders, and state-of-the-art TTS models with extensions like full-band E2E text-to-waveform modeling, which simplify the training pipeline and further enhance TTS performance. The unified design of our recipes enables users to quickly reproduce state-of-the-art E2E-TTS results. We also provide many pre-trained models in a unified Python interface for inference, offering a quick means for users to generate baseline samples and build demos. Experimental evaluations with English and Japanese corpora demonstrate that our provided models synthesize utterances comparable to ground-truth ones, achieving state-of-the-art TTS performance. The toolkit is available online at `https://github.com/espnet/espnet`.

***Index Terms*—** Text-to-speech, end-to-end, open-source, joint training, text-to-waveform

Model Zoo, and state-of-the-art models with our own extensions. The contributions of this paper are summarized as follows:

- We introduce the ESPnet2-TTS toolkit, containing new features like a unified task design, flexible on-the-fly pre-processing, and a simple Python interface to quickly use many pre-trained models in the Model Zoo.
- We introduce our provided state-of-the-art models, including E2E text-to-waveform (T2W) models and joint training with neural vocoders, which simplify the training pipeline and enhance TTS performance. Furthermore, we provide extensions such as the Conformer architecture [20], full-band waveform modeling, and zero-shot adaptation with pre-trained speaker embeddings.
- We conduct experimental evaluations with English and Japanese corpora, and investigate the performance in various settings, including single speaker, multi-speaker, and adaptation. Evaluation results demonstrate that the model can achieve state-of-the-art performance comparable with the ground-truth.

2. RELATED WORKS

ESPnet2-TTS

- **Supports various models including text2wav models**
  - Text2mel models
    - AR models (Tacotron2 and Transformer-TTS)
    - Non-AR models (FastSpeech and FastSpeech2)
  - Text2wav models
    - VITS, JETS, and full-band extensions (44.1 kHz)
  - Of course, we can easily switch to the base architecture (RNN, transformer, or conformer)
- **Supports joint-training of text2mel and mel2wav**
  - Users can combine arbitrary text2mel and mel2wav models
- **Supports multi-speaker extensions**
  - Speaker one-hot, X-vector, and global style token (GST)
  - We can perform zero-shot speaker adaptation
- **Supports various languages**
  - Japanese, English, Mandarin, Korean, Germany, French, Russian, and more !

- Users can quickly develop the **state-of-the-art baseline systems** for the research purpose
- A lot of **examples and demonstration systems**, which works in **real-time** for various languages, https://colab.research.google.com/github/espnet/notebook/blob/master/espnet2_tts_realtime_demo.ipynb

Examples

```python
import time
import torch

# decide the input sentence by yourself
print(f"Input your favorite sentence in {lang}.")
x = input()

# synthesis
with torch.no_grad():
    start = time.time()
    wav = text2speech(x)["wav"]
rtf = (time.time() - start) / (len(wav) / text2speech.fs)
print(f"RTF = {rtf:5f}")

# let us listen to generated samples
from IPython.display import display, Audio
display(Audio(wav.view(-1).cpu().numpy(), rate=text2speech.fs))
```

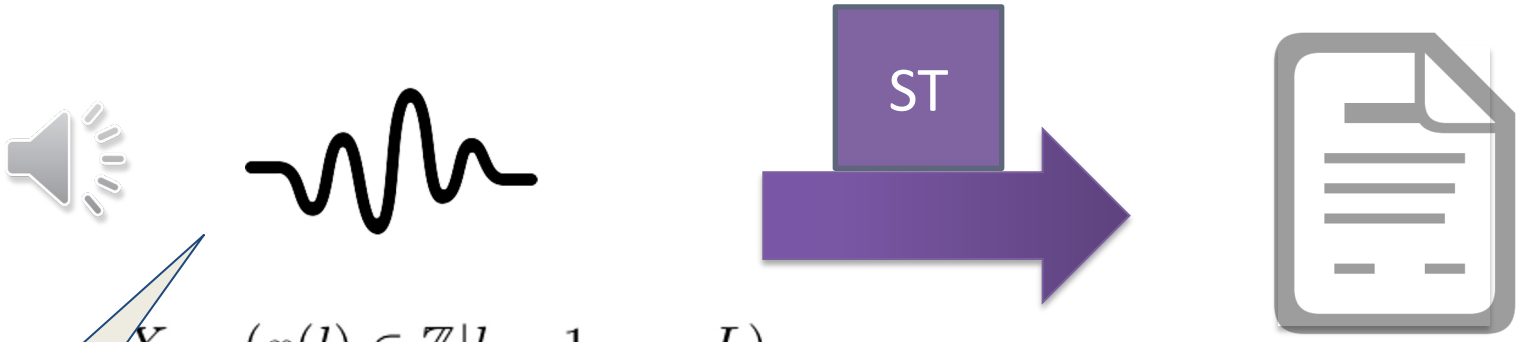
Powered by つくよみちゃん

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

Speech to text translation (ST)

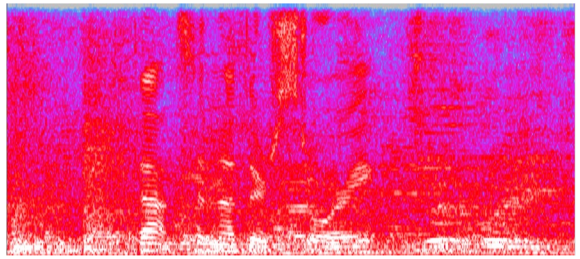
- Mapping **speech** sequence in a **source** language to ***character*** sequence in a **target** language



That's another story

$$X = (x(l) \in \mathbb{Z} | l = 1, \ldots, L)$$

$$L = 43263$$



$$X = (\mathbf{x}_t \in \mathbb{R}^D | t = 1, \ldots, T)$$

$$T = 268$$

"Das ist eine andere Geschichte"

$$W = (\mathbf{w}_n \in \mathcal{V} | n = 1, \ldots, N)$$

N=31

Recipes (7)
IWSLT{19, 21}, Fisher_callhome_Spanish, Must-C, libri_trans, HOW2, Mboshi_French

Functionality
- Fixed frontend
- Single BPE system for src-tgt langs

Models (2)
- Cascaded ST (ASR + MT)
- Encoder-decoder-based ST

Recipes (11)
CoVOST2, Must-C, fisher_callhome_Spanish, HOW2, IWSLT{19, 21, 22}, MTedX, Mboshi_French, libri_trans, Puebla-Nahuatl

Functionality
- OTF frontend (including spectral, SSL feature fusion)
- Option for multiple BPE system for src and tgt langs
- ST Ensemble

Models (4)
- Cascaded ST (ASR + MT)
- Encoder-decoder-based ST
- Blockwise streaming ST
- Multi-decoder ST

IWSLT Challenge 2022

- 1st Place in IWSLT2022 Challenge of dialectal translation.

CMU's IWSLT 2022 Dialect Speech Translation System

Brian Yan[1] **Patrick Fernandes**[1,2] **Siddharth Dalmia**[1] **Jiatong Shi**[1]
Yifan Peng[3] **Dan Berrebbi**[1] **Xinyi Wang**[1] **Graham Neubig**[1] **Shinji Watanabe**[1,4]
[1]Language Technologies Institute, Carnegie Mellon University, USA
[2]Instituto Superior Técnico & LUMLIS (Lisbon ELLIS Unit), Portugal
[3]Electrical and Computer Engineering, Carnegie Mellon University, USA
[4]Human Language Technology Center of Excellence, Johns Hopkins University, USA
{byan, pfernand, sdalmia, jiatongs}@cs.cmu.edu

IWSLT Challenge 2022

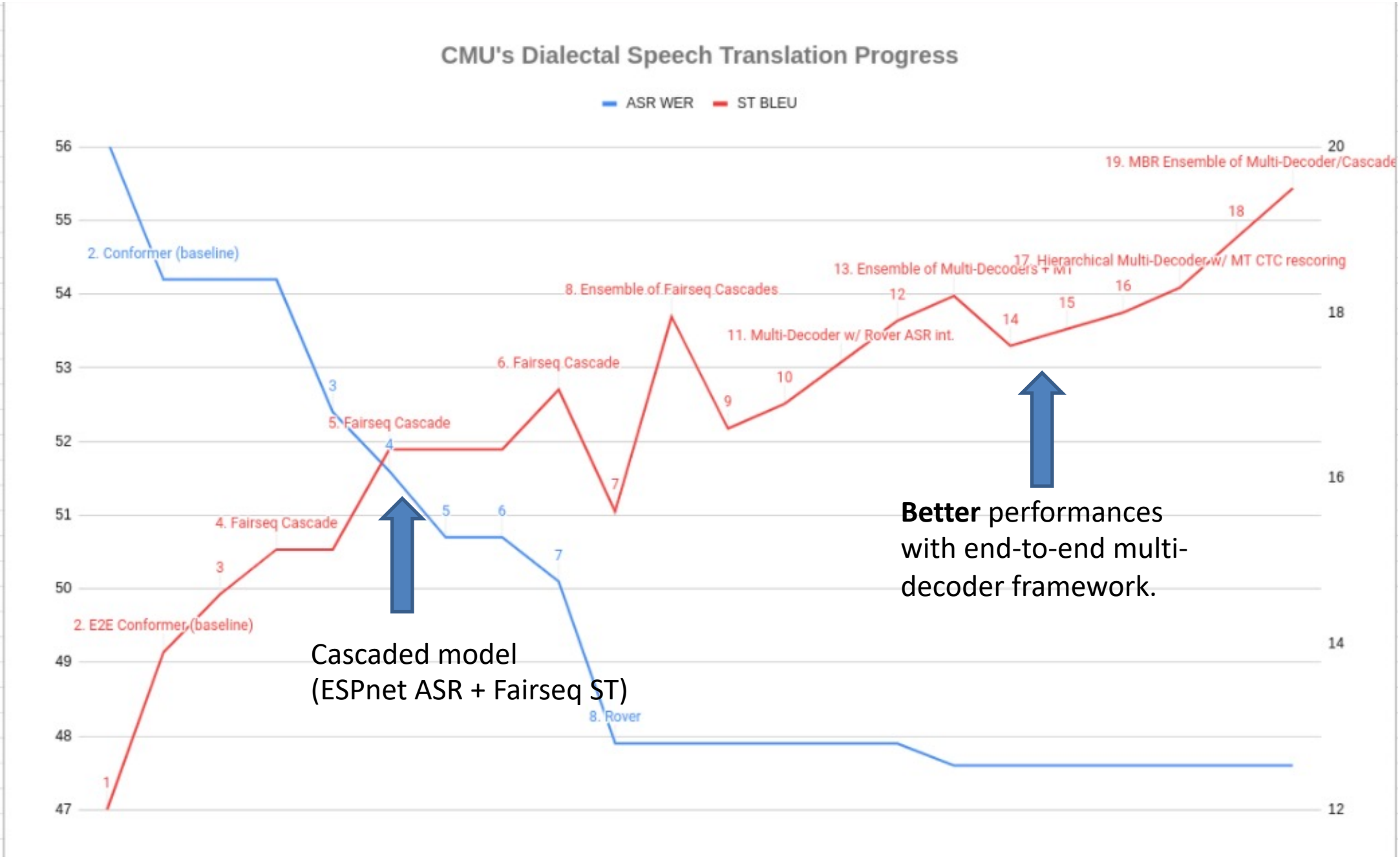
- 1st Place in IWSLT2022 Challenge of dialectal translation.
- The best performance in both objective and subjective evaluation.

Human Evaluation Results

Tunisian Arabic→English

Rank	Ave.	Ave. z	Team / Condition / System
1	76.6	0.457	translator-A
2-3	66.5	0.119	CMU / dialect adapt / contrastive (D6)
2-3	66.5	0.114	CMU / dialect adapt / primary (E2)
4-5	62.7	-0.032	JHU / dialect adapt / primary
4-5	60.7	-0.093	JHU / basic condition / primary
6-7	56.1	-0.271	ON-TRAC / unconstrained / primary
6-7	55.3	-0.302	ON-TRAC / unconstrained / contrastive1

Team / Condition / System	Architecture	Training Data	BLEU
CMU / basic / E1	Mix	TA/EN	20.4
CMU / dialect adapt / E2	Mix	TA/EN + MSA/EN	20.8
JHU / basic / primary	Cascaded	TA/EN	17.1
JHU / dialect adapt / primary	Cascaded	TA/EN + MSA/EN	18.9
ON-TRAC / basic /primary	End-to-End	TA/EN	12.4
ON-TRAC / unconstrained / post-eval	Cascaded	TA/EN + MSA/EN	14.4


CMU's Dialectal Speech Translation Progress
ASR WER
ST BLEU
56
55
54
53
52
51
50
49
48
47
20
18
16
14
12
2. Conformer (baseline)
3
4
5
6
7
8. Rover
1
2. E2E Conformer (baseline)
3
4. Fairseq Cascade
5. Fairseq Cascade
6. Fairseq Cascade
7
8. Ensemble of Fairseq Cascades
9
10
11. Multi-Decoder w/ Rover ASR int.
12
13. Ensemble of Multi-Decoders + MT
14
15
16
17. Hierarchical Multi-Decoder w/ MT CTC rescoring
18
19. MBR Ensemble of Multi-Decoder/Cascade
Cascaded model
(ESPnet ASR + Fairseq ST)
Better performances
with end-to-end multi-
decoder framework.

ESPnet-ST in Endangered language documentation

Highland Puebla Nahuatl–Spanish Speech Translation Corpus for Endangered Language Documentation

Jiatong Shi
The Johns Hopkins University
jiatong_shi@jhu.edu

Jonathan D. Amith
Gettysburg College
jonamith@gmail.com

Xuankai Chang, Siddharth Dalmia, Brian Yan, and Shinji Watanabe
Carnegie Mellon University
shinjiw@ieee.org

is the system with multi-task learning; S1 is the system
with searchable intermediates.

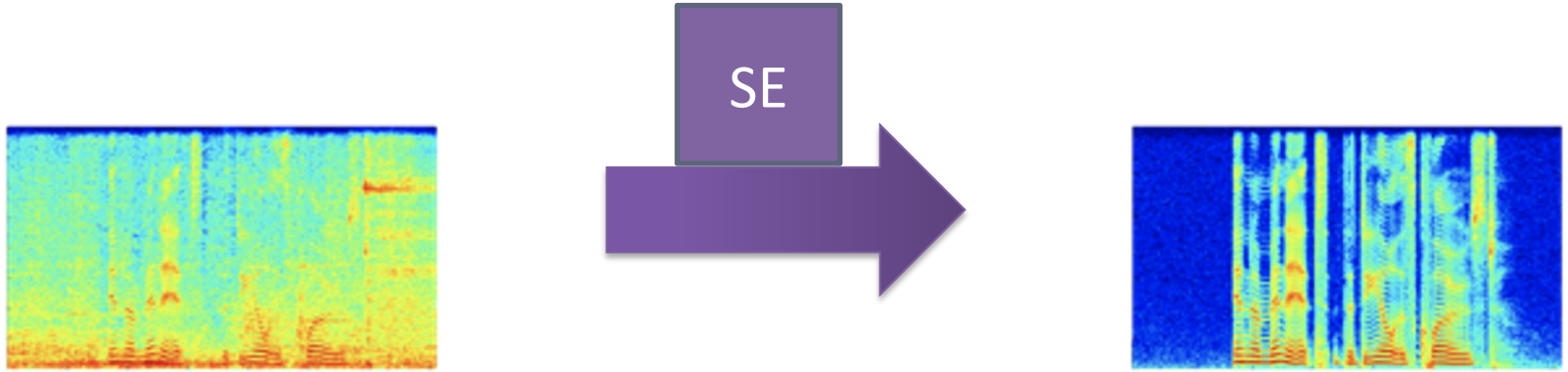
From Wikipedia

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

Speech enhancement (SE)

- Mapping **noisy** speech sequence to **clean** speech sequence



$$X = (\mathbf{x}_t \in \mathbb{R}^D | t = 1, \ldots, T)$$
$$T = 268$$

$$X' = (\mathbf{x}'_t \in \mathbb{R}^D | t = 1, \ldots, T)$$
$$T = 268$$

ESPnet-SE

ESPNET-SE: END-TO-END SPEECH ENHANCEMENT AND SEPARATION TOOLKIT DESIGNED FOR ASR INTEGRATION

Chenda Li[1*], *Jing Shi*[2,3*], *Wangyou Zhang*[1*], *Aswin Shanmugam Subramanian*[3], *Xuankai Chang*[3], *Naoyuki Kamo, Moto Hira*[4], *Tomoki Hayashi*[5,6], *Christoph Boeddeker*[7], *Zhuo Chen*[8], *Shinji Watanabe*[3]
[1]Shanghai Jiao Tong University, [2]Institute of Automation, Chinese Academy of Sciences,
[3]Johns Hopkins University, [4]Facebook AI, [5]Nagoya University,
[6]Human Dataware Lab. Co., Ltd., [7]Paderborn University, [8]Microsoft Research

ABSTRACT

We present ESPnet-SE, which is designed for the quick development of speech enhancement and speech separation systems in a single framework, along with the optional downstream speech recognition module. ESPnet-SE is a new project which integrates rich automatic speech recognition related models, resources and systems to support and validate the proposed front-end implementation (i.e. speech enhancement and separation). It is capable of processing both single-channel and multi-channel data, with various functionalities including dereverberation, denoising and source separation. We provide all-in-one recipes including data pre-processing, feature extraction, training and evaluation pipelines for a wide range of benchmark

In this paper, we introduce a new E2E-SE toolkit named ESPnet-SE[1], which is an extension of the open-source speech processing toolkit ESPnet [8]. ESPnet-SE fully considers the various forms of speech input in the front-end scenes and meanwhile flexibly and organically integrated with the downstream automatic speech recognition (ASR) task, making it a user-friendly toolkit to easily build totally end-to-end robust ASR systems, even without need for clean speech signals. The toolkit provides adaptability to different speech data, including (1) single and multiple speakers, (2) single and multiple channels, (3) anechoic and reverberant conditions. Moreover, thanks to the ripe and efficient ASR modules in ESPnet, rich speech recognition related models, resources and systems can be optionally concatenated after the E2E-SE system, enabling evaluation and joint

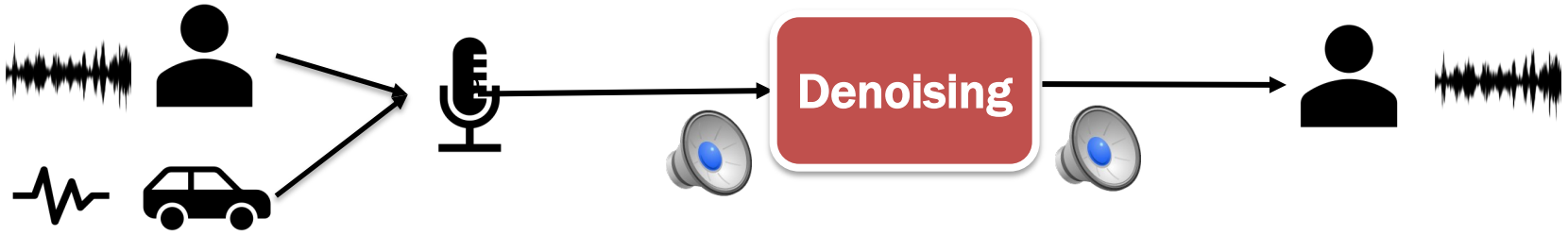
ESPnet-SE

- One of the biggest changes in ESPnet
- Include **ALL** speech enhancement functions
- Demonstration

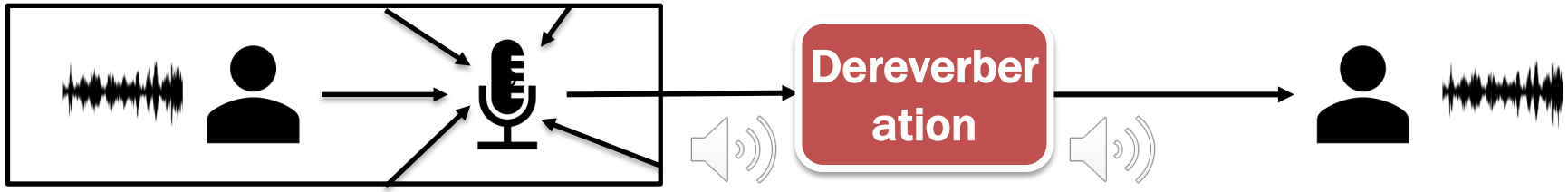
https://colab.research.google.com/drive/1fjRJCh96SoYLZPRxsjF9VDv4Q2VoIckI?usp=sharing

Speech enhancement
Several types of problems

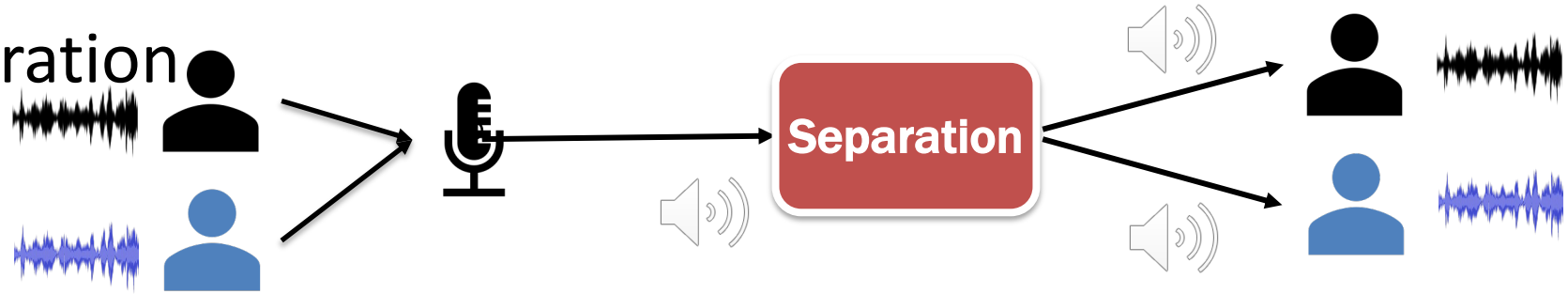
- Denoising (people mainly call it speech enhancement)



- Dereverberation

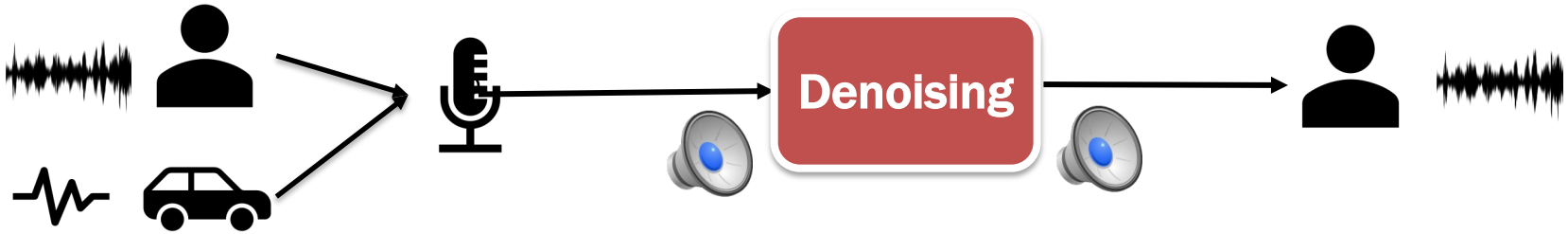


- Separation



Speech enhancement
Several types of problems

- Denoising (people mainly call it speech enhancement)



Enhance the single-channel real noisy speech in CHiME4

```python
# play the enhanced single-channel speech
wave = enh_model_sc(mixwav_sc[None, ...], sr)
print("Input real noisy speech", flush=True)
display(Audio(mixwav_sc, rate=sr))
print("Enhanced speech", flush=True)
display(Audio(wave[0].squeeze(), rate=sr))
```

```
Input real noisy speech
```

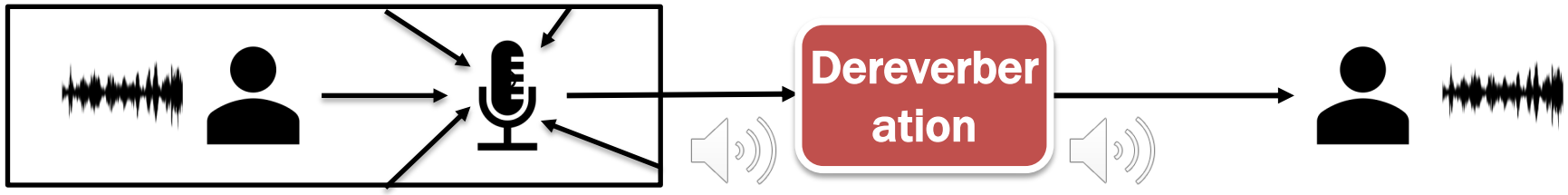
0:00 / 0:05

```
Enhanced speech
```

0:00 / 0:05

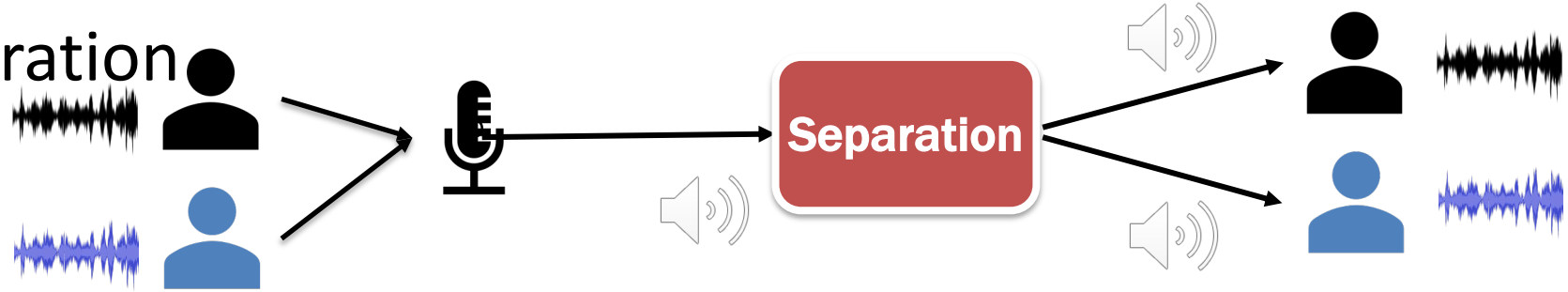
Speech enhancement
Several types of problems

- Dereverberation



Speech enhancement
Several types of problems

- Separation



Separate the example in wsj0_2mix testing set

```
!gdown --id 1ZCUkd_Lb7pO2rpPr4FqYdtJBZ7JMiInx -O /content/447c020t_1.2106_422a0112_-1.2106.wav

import os
import soundfile
from IPython.display import display, Audio

mixwav, sr = soundfile.read("447c020t_1.2106_422a0112_-1.2106.wav")
waves_wsj = separate_speech(mixwav[None, ...], fs=sr)

print("Input mixture", flush=True)
display(Audio(mixwav, rate=sr))
print(f"========= Separated speech with model {tag} =========", flush=True)
print("Separated spk1", flush=True)
display(Audio(waves_wsj[0].squeeze(), rate=sr))
print("Separated spk2", flush=True)
display(Audio(waves_wsj[1].squeeze(), rate=sr))
```

```
Downloading...
From: https://drive.google.com/uc?id=1ZCUkd_Lb7pO2rpPr4FqYdtJBZ7JMiInx
To: /content/447c020t_1.2106_422a0112_-1.2106.wav
100% 184k/184k [00:00<00:00, 5.73MB/s]
100%|██████████| 13/13 [00:00<00:00, 28.27it/s]Input mixture
```

0:00 / 0:11

```
========= Separated speech with model Chenda Li/wsj0_2mix_enh_train_enh_conv_tasnet_raw_valid.si_snr.
Separated spk1
```

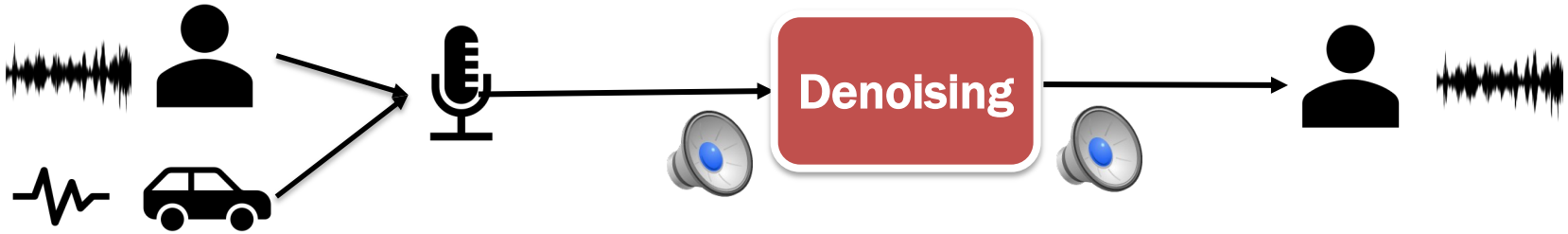
0:00 / 0:11

```
Separated spk2
```

0:00 / 0:11

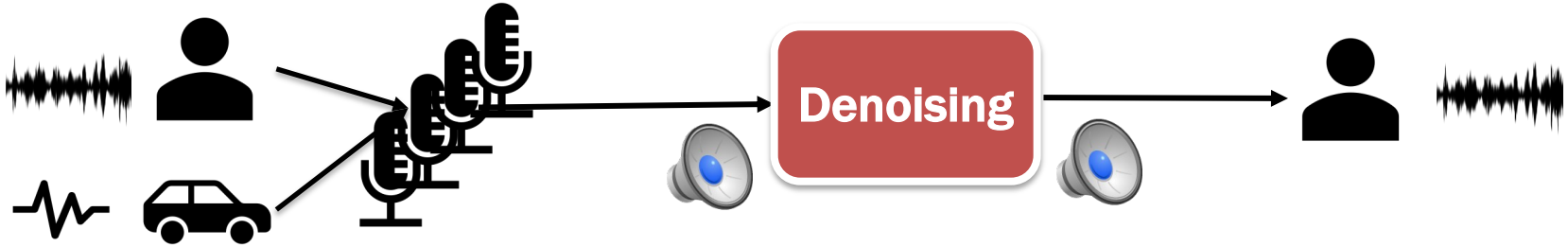
Microphone array processing
Single to multiple microphones

- Denoising (people mainly call it speech enhancement)



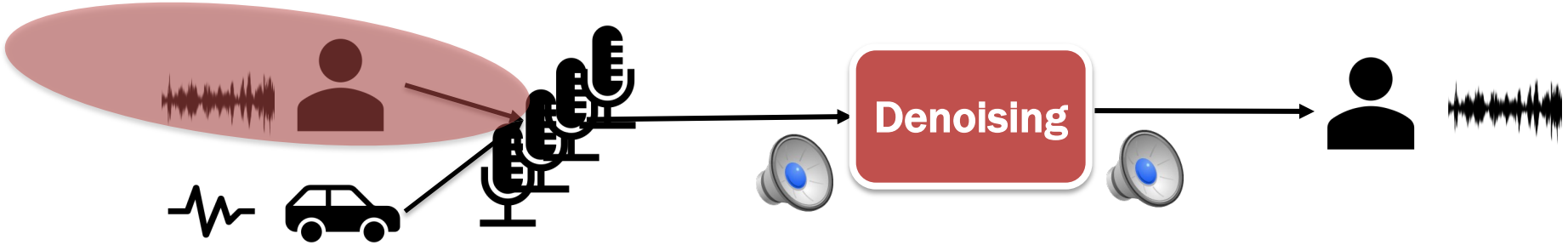
Microphone array processing
Single to multiple microphones

- Denoising (people mainly call it speech enhancement)



Microphone array processing
Single to multiple microphones

- Denoising (people mainly call it speech enhancement)



Make a spatial **beam** (beamforming) to only pick up desired signals

Enhance the multi-channel real noisy speech in CHiME4

```
wave = enh_model_mc(mixwav_mc[None, ...], sr)
print("Input real noisy speech", flush=True)
display(Audio(mixwav_mc.T, rate=sr))
print("Enhanced speech", flush=True)
display(Audio(wave[0].squeeze(), rate=sr))
```

```
/usr/local/lib/python3.7/dist-packages/torch/functi
  normalized, onesided, return_complex)
Input real noisy speech
/usr/local/lib/python3.7/dist-packages/torch/functi
  normalized, onesided, length, return_complex)
```

0:00 / 0:05

```
Enhanced speech
```

0:00 / 0:05

ESPnet-SE++: Speech Enhancement for Robust Speech Recognition, Translation, and Understanding

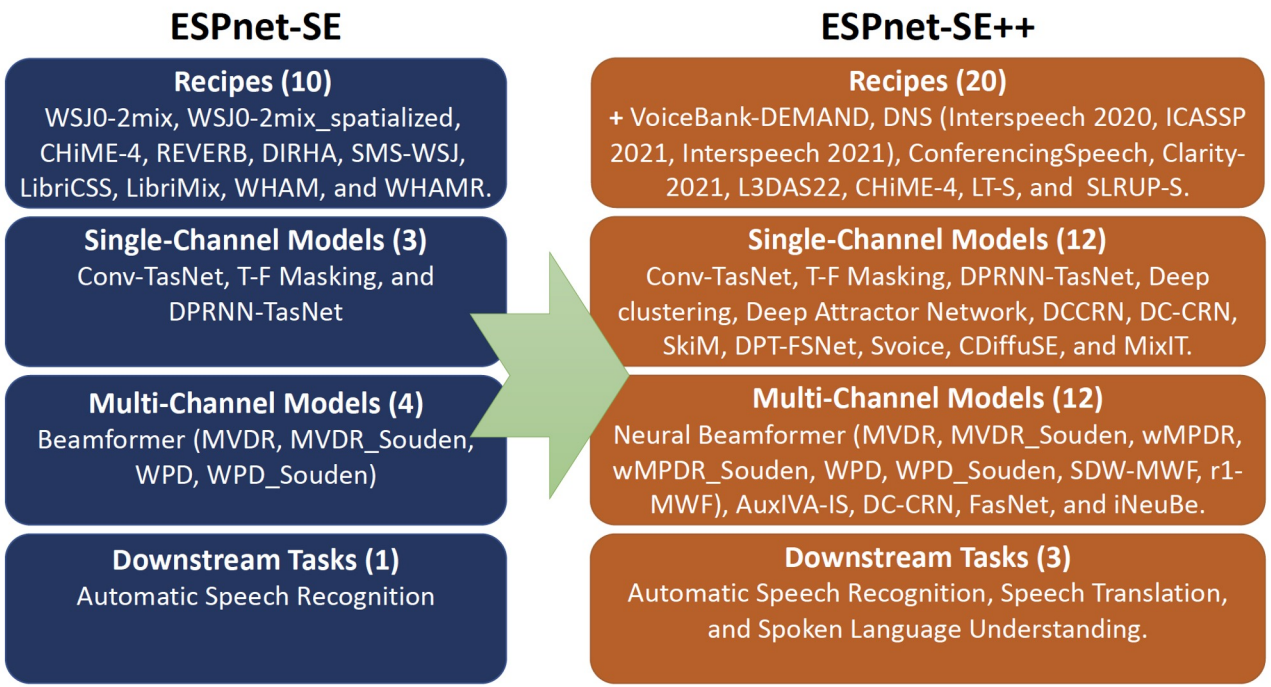
Yen-Ju Lu, Xuankai Chang, Chenda Li, Wangyou Zhang, Samuele Cornell, Zhaoheng Ni, Yoshiki Masuyama, Brian Yan, Robin Scheibler, Zhong-Qiu Wang, Yu Tsao, Yanmin Qian, Shinji Watanabe

Open-source: ESPnet-SE++

- Extension of ESPnet-SE
  - **New recipes/models** for several **speech separation and enhancement (SSE)** corpora and challenge.
  - **Unsupervised separation (MixIT)** and **generative enhancement (CDiffuSE)** approaches.
- Combination of SSE with back-end tasks
  - Speech enhancement with **recognition, translation, and understanding.**
  - **New spatialized multi-channel noisy-reverberant corpora** derived from Libri-trans and SLURP as benchmark.

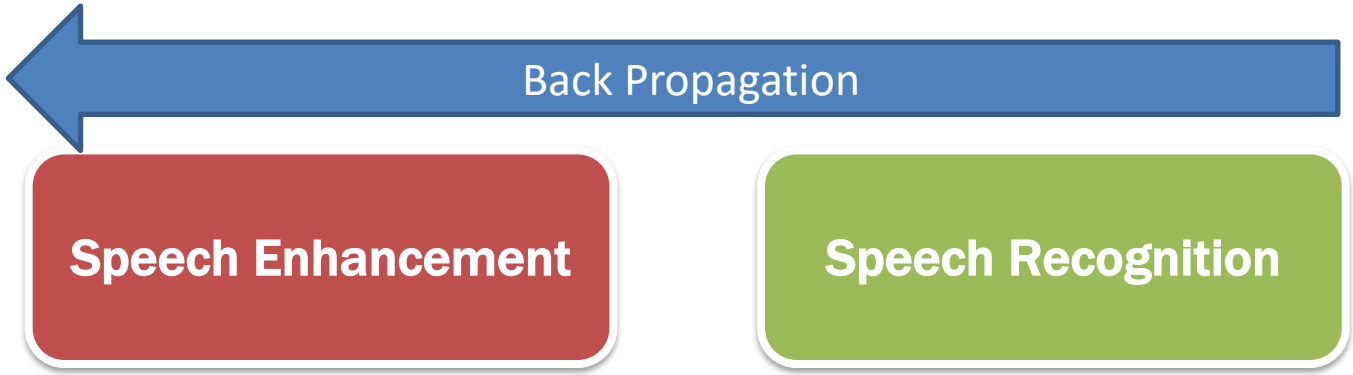
Extension of ESPnet-SE and New Benchmarks

- **10 new recipes** and **17 new models** in ESPnet-SE
- New Downstream Tasks: **Speech Translation (ST), Spoken Language Understanding (SLU)**
- New Datasets
  - **Spatialized Libri-Trans**
  - **Spatialized SLURP**



Differentiable speech enhancement frontend

- ESPnet SE can be used as an **independent** enhancement module
  - Denoising, Dereveberation, Separation
  - Single channel or multichannel
- ESPnet SE can be used as a **differentiable** enhancement module



- We can realize a cocktail party effect by a machine

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

TOWARDS LOW-DISTORTION MULTI-CHANNEL SPEECH ENHANCEMENT: THE ESPNET-SE SUBMISSION TO THE L3DAS22 CHALLENGE

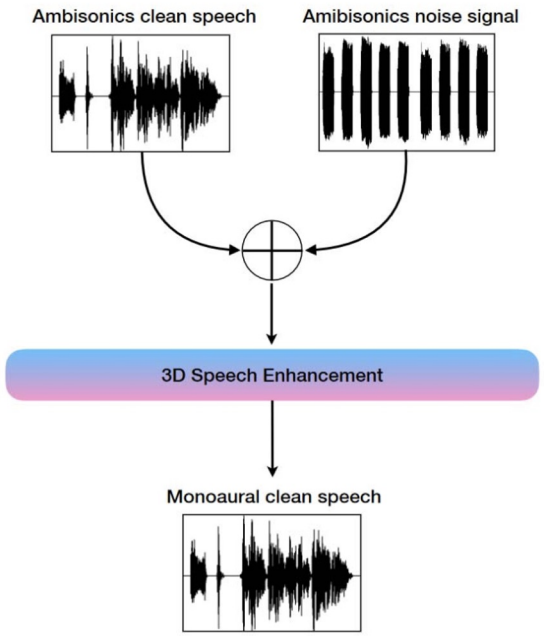
Yen-Ju Lu[1,5], *Samuele Cornell*[2], *Xuankai Chang*[1], *Wangyou Zhang*[4], *Chenda Li*[4], *Zhaoheng Ni*[3], *Zhong-Qiu Wang*[1], *Shinji Watanabe*[1]

[1]Carnegie Mellon University, USA [2]Università Politecnica delle Marche, Italy [3]Meta AI, USA
[4]Shanghai Jiao Tong University, Shanghai [5]Academia Sinica, Taipei

{cornellsamuele, neilyenjulu, nizhaoheng, wang.zhongqiu41}@gmail.com

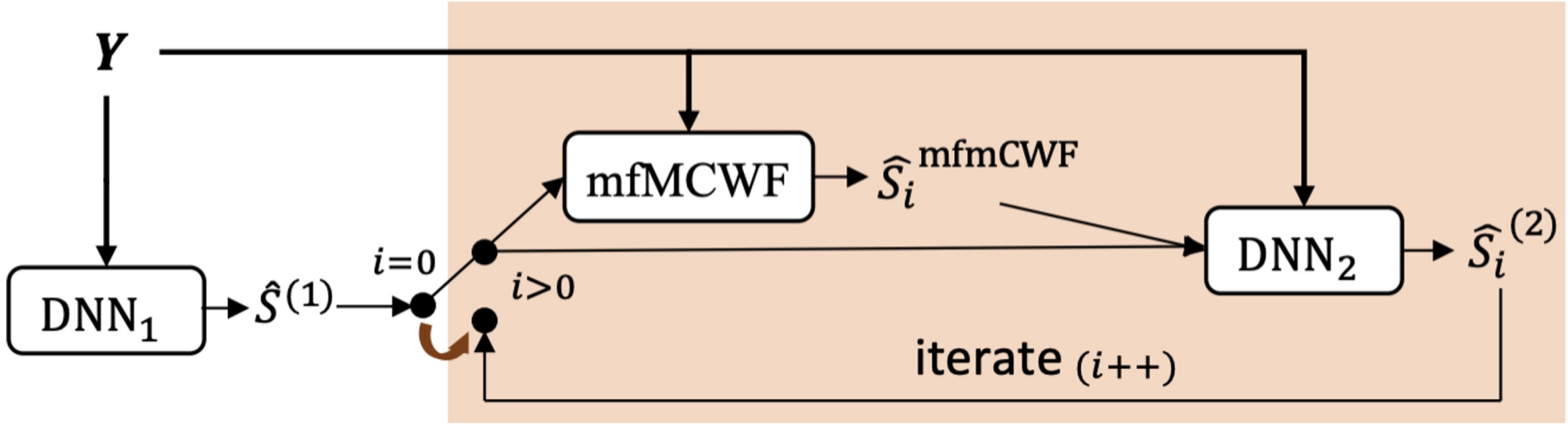
Introduction

- Multi-channel Speech Enhancement
  - Aims at estimating clean speech from noisy-reverberant audio recordings by multiple microphones.
- L3DAS22 Challenge
  - Multi-channel mixture data
    - 8-channels 16kHz wav files
    - 2 sets of first-order B-Format Ambisonics microphone
  - Evaluation Metric
    - Score = (STOI + (1-WER))/2
    - WER* is computed by a pre-trained Wav2Vec2 AS



iNeuBe: iterative Neural Beamforming Enhancement

- Iterative Neural Beamforming Enhancement

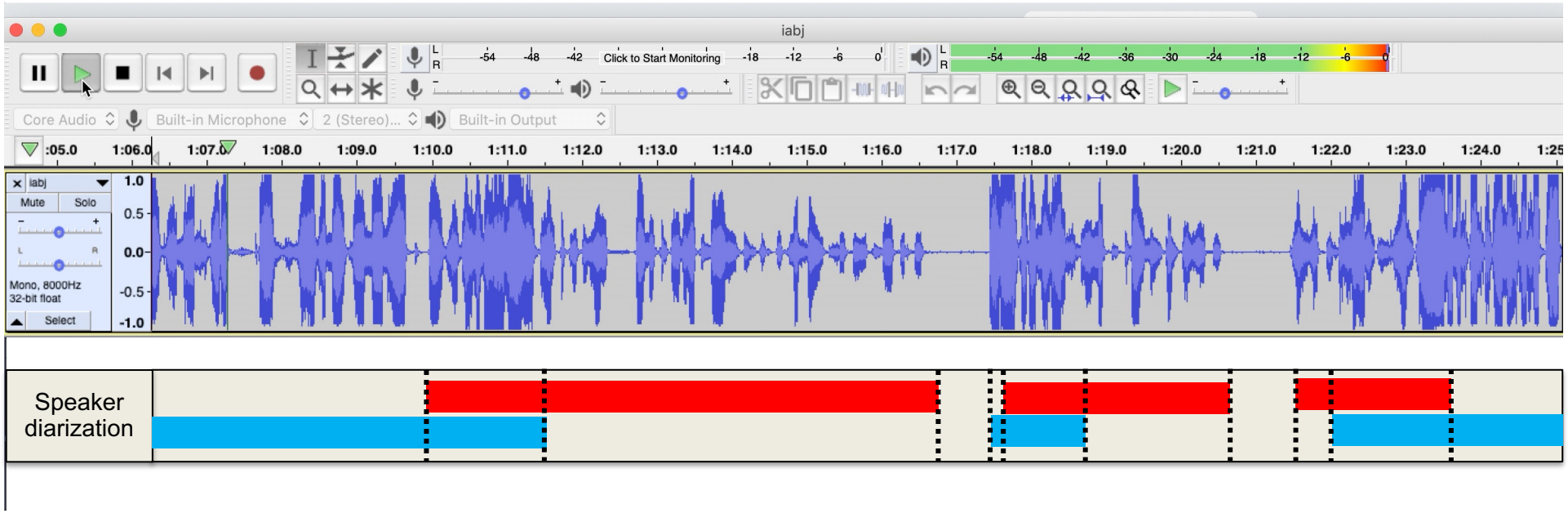


  - First stage: multichannel speech enhancement based on DNN_1 to provide a mask
  - Second stage: mask-based beamforming
  - Third stage: post filtering based on DNN_2

Today’s talk

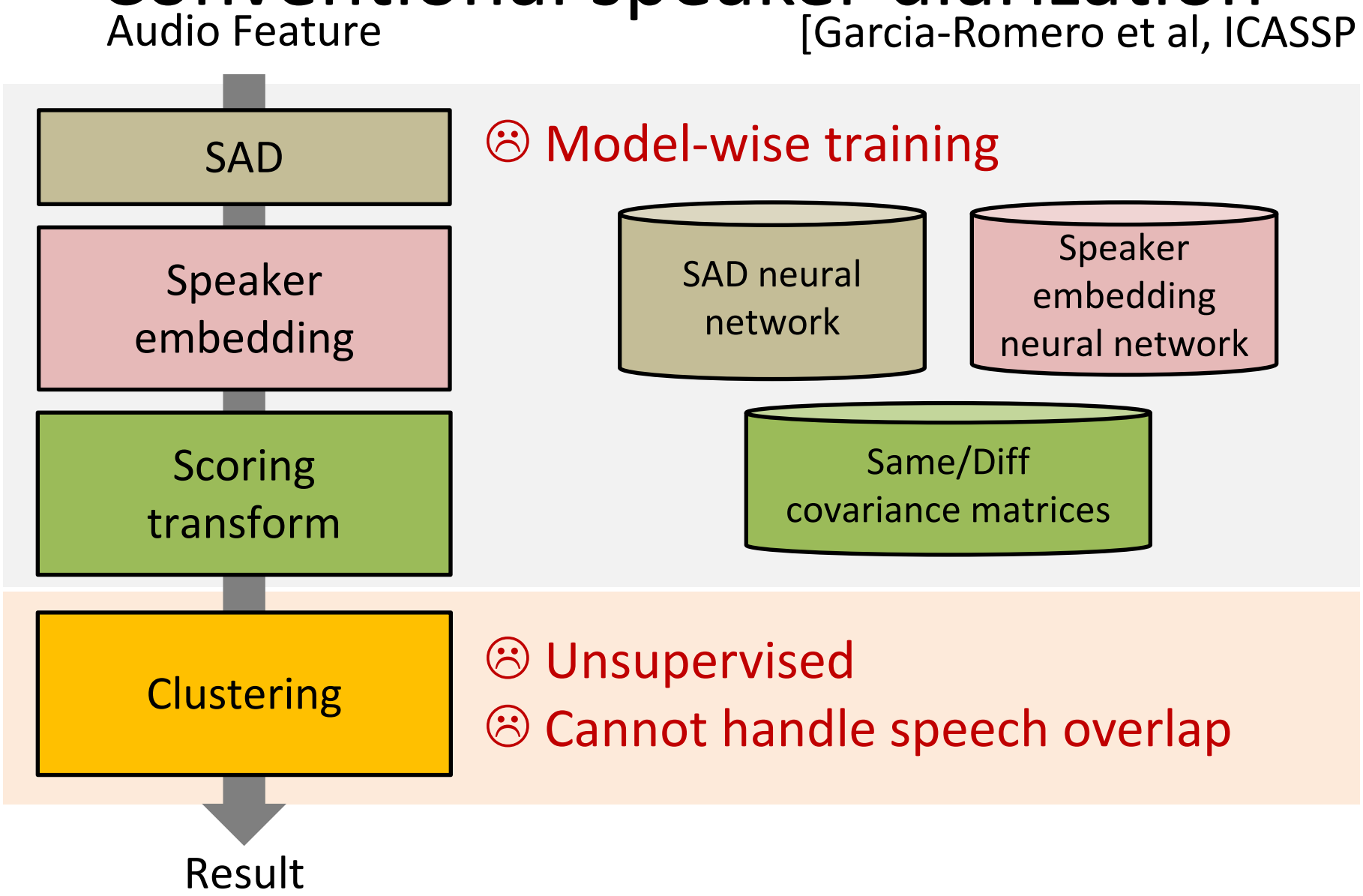
- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

Speaker diarization



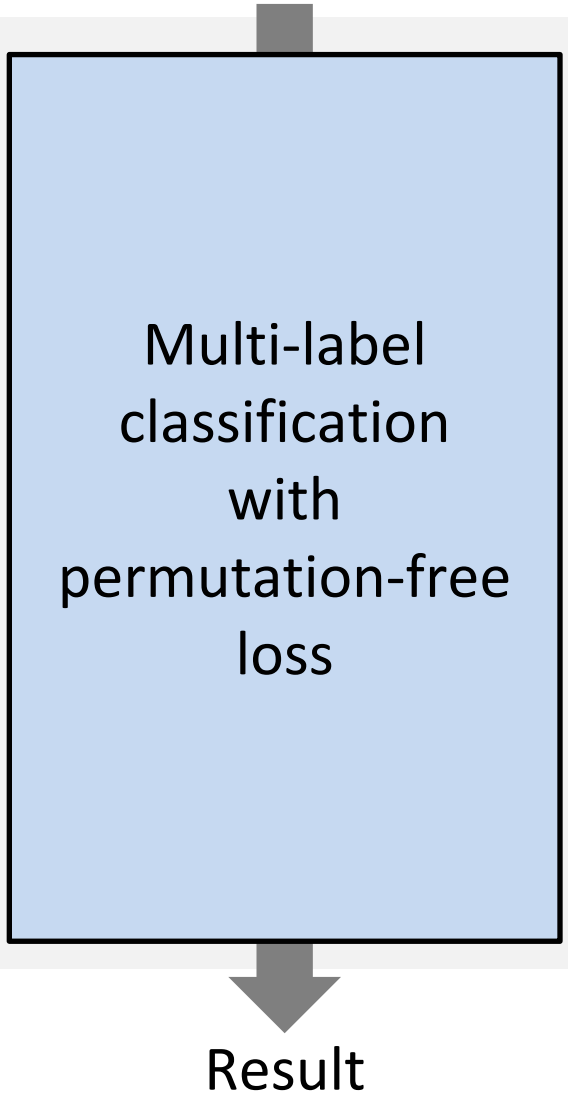
- Solves “who is speaking when”

Conventional speaker diarization

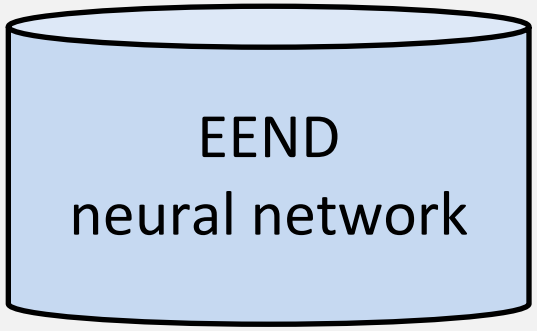


End-to-End Neural Diarization (EEND)

[Fujita, Interspeech 2019]



☺ Only one network to be trained †



☺ Fully-supervised †

☺ Can handle speech overlap

EEND in ESPnet

- Supports 3 EEND variants
  - Self-Attentive EEND (**SA-EEND**): Fixed number of speakers
  - EEND with Encoder-Decoder Attractor calculation (**EEND-EDA**): Can handle flexible numbers of speakers
  - Joint EEND and Speech Separation (**EEND-SS**): Joint modeling of EEND-EDA and Speech Separation
- Currently supports 2 recipes
  - mini-LibriSpeech dataset & LibriMix dataset
  - More and more recipes will be supported soon!

Joint speech separation and speaker diarization

Libri2Mix (2 speakers)

	DER	STOI	SDR	SI-SDR
SA-EEND	7.26	-	-	-
EEND-EDA	5.93	-	-	-
Conv-TasNet	-	0.824	8.62	7.54
EEND-SS	5.08	0.826	8.79	7.76

Libri2Mix + Libri3Mix (2 +3 speakers mixture)

	DER	STOI	SDR	SI-SDR
EEND-EDA	10.16	-	-	-
Conv-TasNet	-	0.756	5.59	4.25
EEND-SS	6.27	0.760	5.91	4.39

Today’s talk

- Introduction of ESPnet, end-to-end speech processing toolkit
- Broadened applications
- Automatic speech recognition (ASR)
  - Performance improvement
  - New features
    - Non-autoregressive modeling
    - Multilingual ASR
    - Streaming functions
- Speech synthesis
- Speech translation
  - IWSLT challenge
- Speech enhancement
  - L3DAS challenge
- Speaker diarization
- Spoken language understanding

ESPnet-SLU

ESPNET-SLU: ADVANCING SPOKEN LANGUAGE UNDERSTANDING THROUGH ESPNET

Siddhant Arora[1], *Siddharth Dalmia*[1], *Pavel Denisov*[2], *Xuankai Chang*[1], *Yushi Ueda*[1], *Yifan Peng*[1], *Yuekai Zhang*[3], *Sujay Kumar*[1], *Karthik Ganesan*[1], *Brian Yan*[1], *Ngoc Thang Vu*[2], *Alan W Black*[1], *Shinji Watanabe*[1]

[1]Carnegie Mellon University, [2]University of Stuttgart, [3]Zoom Video Communications

ABSTRACT

As Automatic Speech Processing (ASR) systems are getting better, there is an increasing interest of using the ASR output to do downstream Natural Language Processing (NLP) tasks. However, there are few open source toolkits that can be used to generate reproducible results on different Spoken Language Understanding (SLU) benchmarks. Hence, there is a need to build an open source standard that can be used to have a faster start into SLU research. We present ESPnet-SLU, which is designed for quick development of spoken language understanding in a single framework. ESPnet-SLU is a project inside end-to-end speech processing toolkit, ESPnet, which is a widely used open-source standard for various speech processing tasks like ASR, Text to Speech (TTS) and Speech Translation (ST). We enhance the toolkit to provide implementations for various SLU benchmarks that enable researchers to seamlessly mix-and-match different ASR and NLU models. We also provide pretrained models with intensively tuned hyper-parameters that can match or even outperform the current state-of-the-art performances. The toolkit is

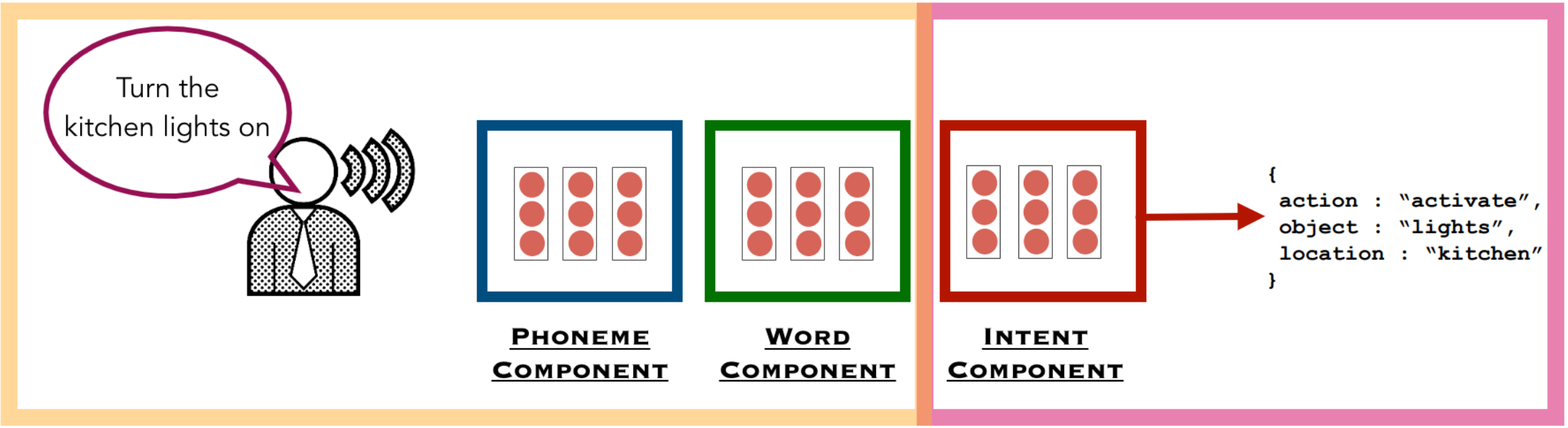
are also useful for low resource languages [12] where there is not enough training data or access to reliable transcripts to separately train ASR and NLU components.

With the increase in SLU datasets and methodologies proposed [3, 9, 13], there is a growing need for an open-source SLU toolkit which would help standardize the pipelines involved in building an SLU model like data preparation, model training, and its evaluation. Our goal is to provide an open-source standard where researchers can easily incorporate previously proposed technologies, compare and contrast new ideas with the existing methodologies. In this work, we introduce a new E2E-SLU toolkit built on an already existing open-source speech processing toolkit ESPnet [14–16]. ESPnet supports a variety of speech processing tasks ranging from front-end processing like enhancement and separation to recognition and translation. Having ESPnet-SLU would help users build systems for real-world scenarios where many speech processing steps need to be applied before running the downstream task. ESPnet also provides an easy access to other speech technologies being developed like data-

Spoken Language Understanding

Definition: As ASR systems are getting better, there is an increasing interest of using the ASR output to do downstream NLP tasks

- Example: Spoken Language Understanding (SLU) [1] = ASR + NLU



ESPnet-SLU

ESPnet-SLU

A new End to End SLU toolkit built on an already existing open-source speech processing toolkit ESPnet

Contributions

1. Provide pipeline that cover all experiment processes for various SLU tasks.
2. Incorporate the usage of pretrained ASR like Hubert, Wav2vec2 and NLU models like BERT, MPNet as feature extractors
3. Implementations of various speech processing tasks that can be used in a pipeline manner
4. Release an open-source toolkit and provide easy access to the trained models that match SOTA performance on various SLU datasets

Design Features of ESPnet-SLU

- Easy access to speech technologies being developed
  - Supports various type of encoders (Transformer/Conformer) and Decoder
- Support Multitasking with ASR
  - Predict both intent and ASR transcript
- Support Multitasking with NLU
  - Deliberation encoder based 2 pass model
- Support using pretrained ASR and NLP models as feature extractors
- Various language support
  - En / Jp / Zn / Nl / And more...
- Supports using context from previous utterances
- Supports using other tasks like SE in pipeline manner

Benchmark Results

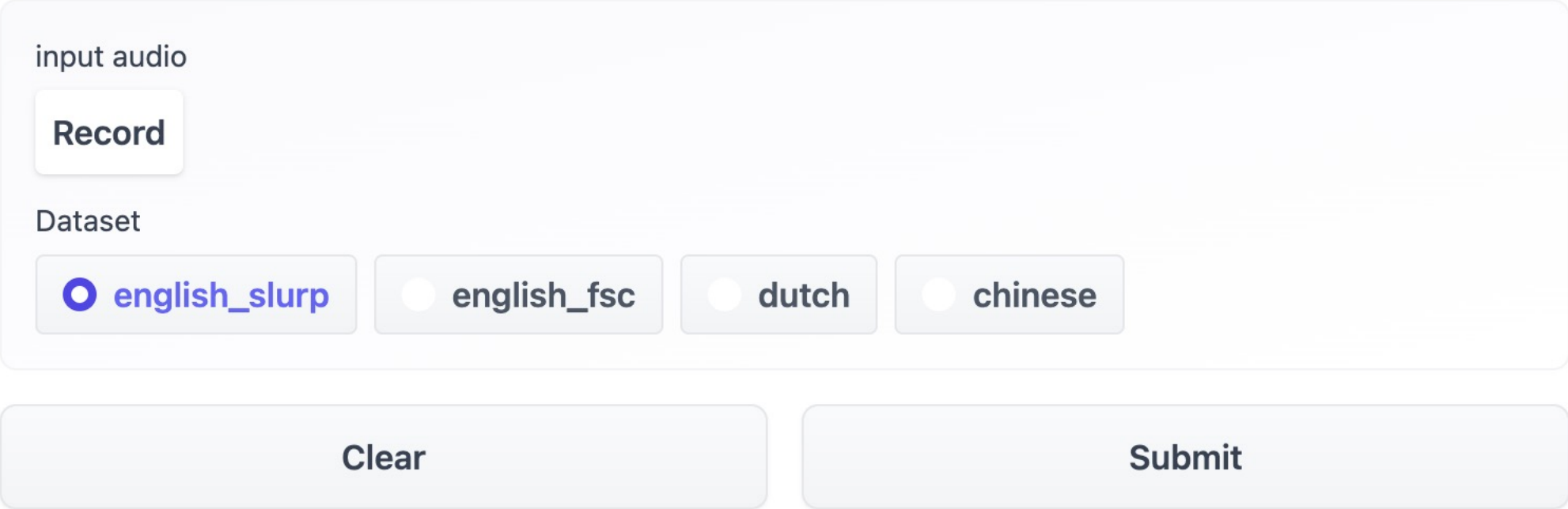
Table 2. Supported tasks and datasets in ESPnet-SLU along with their reported performance in the original paper and our toolkit. We show the metrics used in the original paper. We match or outperform SOTA performance across a variety of SLU benchmarks.

Task	Dataset	Metric	Paper Results	**ESPnet-SLU**
IC	SLURP [4]	Acc.	78.3	86.3
	FSC [3]	F1	98.8	99.6
	FSC Unseen (S) [3, 40]	Acc.	94.2	98.6
	FSC Unseen (U) [3, 40]	Acc.	88.3	86.4
	FSC Challenge (S) [3, 40]	Acc.	92.3	97.5
	FSC Challenge (U) [3, 40]	Acc.	78.3	78.5
	SNIPS [13]	F1	91.7	91.7
	HarperValleyBank [41]	Acc	45.5	47.1
	Grabo [12, 42]	Acc.	94.5	97.2
	CAT-SLU MAP [27, 43]	Acc.	79.8	78.9
	Speech Commands [44]	Acc.	88.2	98.4
SF	SLURP [4]	SLU-F1	70.8	71.9
DA	Switchboard [45, 46]	Acc.	68.7	67.5
	HarperValleyBank [41]	Acc.	45.5	47.1
ER	IEMOCAP [6, 47]	5-fold Acc.	67.6	69.4

Recipes for over 10 SLU corpora, for multiple languages and task types, with performance nearing or exceeding the prior SOTA.

Integrated to Huggingface Spaces with Gradio

See SLU demo on multiple languages:
https://huggingface.co/spaces/Siddhant/ESPnet2-SLU



Summary and future work

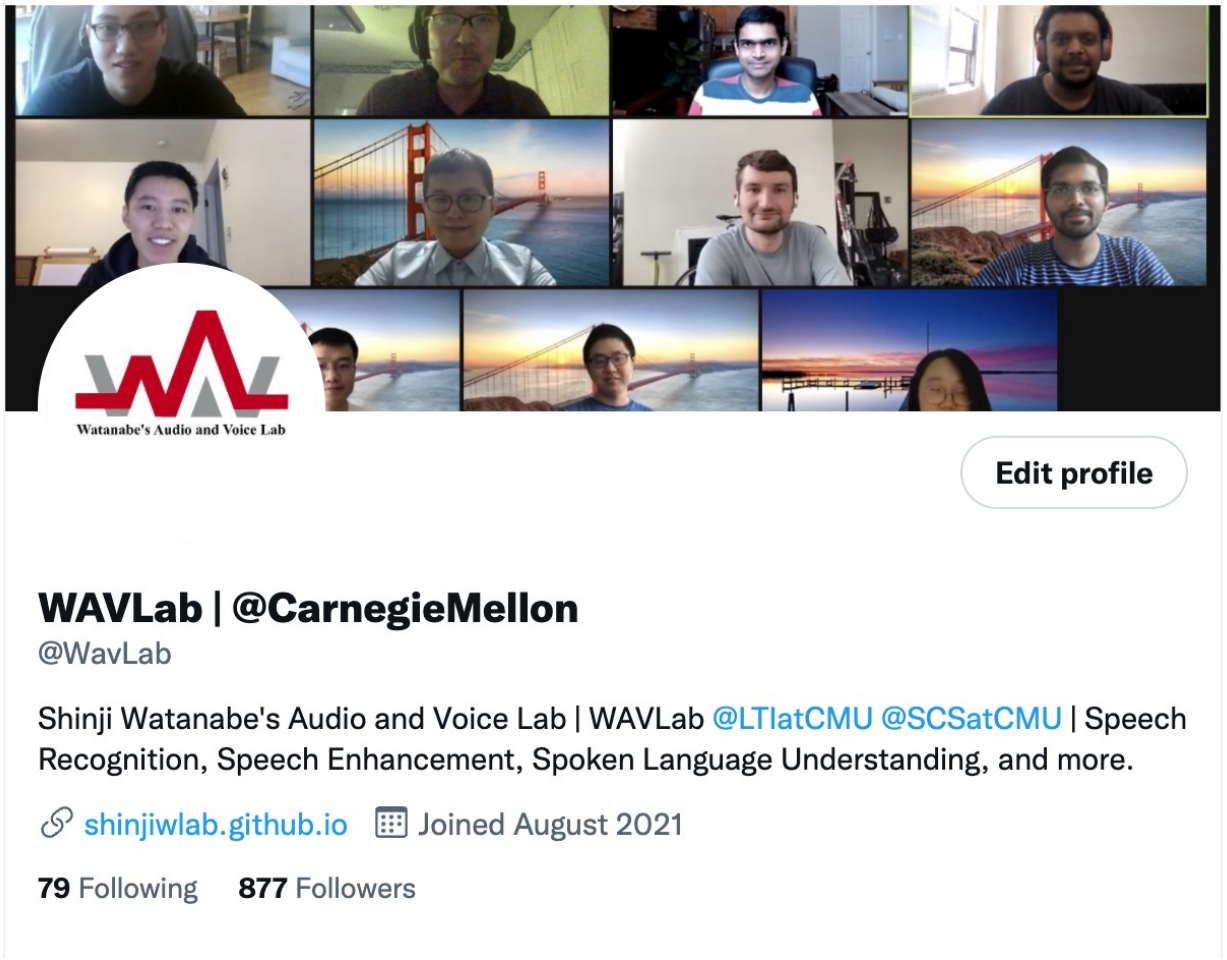
- End-to-end speech processing has a lot of potentials
- ESPnet provides **state-of-the-art** and **reproducible** research in various speech processing applications
- Future work
  - More applications, e.g., speaker verification, audio event detection, music processing
  - Enhance real-time/streaming applications
  - Continues to follow the state-of-the art performance

We are working on this project for the community contribution!!

Let’s work together!!

Follow our Twitter account to receive latest ESPnet Info

- https://twitter.com/WavLab



Acknowledgement

- Most slides are from Prof. Shinji Watanabe and core ESPnet contributors
- The content presented is a brief summary of the work. Please look to specific papers for details

Thanks a lot!!!