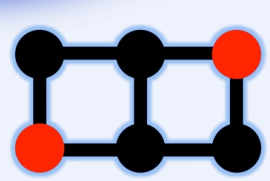
# ESPnet-SE toolkit Introduction


汇报人：钱彦旻、李晨达、张王优

汇报时间：2022.11.13

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY



# ESPnet

## End-to-end Speech Processing toolkit

Apache-2.0 license

5.7k stars

181 watching

1.8k forks

### Versatile framework

1. E2E speech recognition
2. Text-to-speech (TTS)
3. Voice conversion (VC)
4. Speech translation (ST)
5. Speech enhancement (SE & SS)
6. Speaker diarization (SD)
7. Spoken language understanding (SLU)
8. etc.

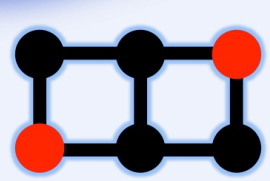
### All-in-one recipes

1. Data preparation
2. Feature extraction
3. Model training
4. Inference and evaluation
5. Model sharing via Hugging Face

### Rich building blocks

1. Available architectures:
   - RNN, Transformer, Conformer, Branchformer, TCN, U-Net, etc.
2. On-the-fly feature extraction and preprocessing during training
3. Supporting multiple nodes training and integrated with Slurm or MPI
4. Off-the-shelf iterators (seq, chunk)
5. etc.

https://github.com/espnet/espnet



# ESPnet

## End-to-end Speech Processing toolkit

Apache-2.0 license

5.7k stars

181 watching

1.8k forks

### Ever-updating tutorials

1. Official documentation: https://espnet.github.io/espnet/tutorial.html
2. Interactive tutorials via Google Colab or Jupyter Notebook: https://github.com/espnet/notebook
3. Tutorial slides from Interspeech 2019: https://github.com/espnet/interspeech2019-tutorial
4. Lecture materials and online videos at CMU (2021 & 2022):
    - Overview of the main features in ESPnet: https://www.youtube.com/watch?v=2mRz3wH1vd0
    - Usage of ESPnet (ASR as an example): https://www.youtube.com/watch?v=YDN8cVjxSik
    - How to add new models/tasks to ESPnet: https://www.youtube.com/watch?v=Css3XAes7SU

https://github.com/espnet/espnet

# ESPnet-SE (Since 2020)

Introduce SE functions to ESPnet: denoising, dereverberation, and separation


[1] C. Li, J. Shi, W. Zhang, A. S. Subramanian, X. Chang, N. Kamo, M. Hira, T. Hayashi, C. Boeddeker, Z. Chen, and S. Watanabe, "ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration," in Proc. IEEE SLT, 2021, pp. 785–792.

[2] S. Watanabe, F. Boyer, X. Chang, P. Guo, T. Hayashi, Y. Higuchi, T. Hori, W.-C. Huang, H. Inaguma, N. Kamo, S. Karita, C. Li, J. Shi, A. S. Subramanian, and W. Zhang, "The 2020 ESPnet update: New features, broadened applications, performance improvements, and future plans," in IEEE Data Science and Learning Workshop (DSLW), 2021, pp. 1–6.

[3] Y.-J. Lu, X. Chang, C. Li, W. Zhang, S. Cornell, Z. Ni, Y. Masuyama, B. Yan, R. Scheibler, Z.-Q. Wang, Y. Tsao, Y. Qian, and S. Watanabe, "ESPnet-SE++: Speech enhancement for robust speech recognition, translation, and understanding," in Proc. ISCA Interspeech, 2022, pp. 5458–5462.


https://github.com/espnet/espnet

CONTENT

# 目录

Milestones

01 发展历程

Features

02 特色功能

Demos

03 样例展示

Outlook

04 未来展望

# Milestones

## 01 发展历程

# 01 发展历程

Milestones

2020.01-2020.06

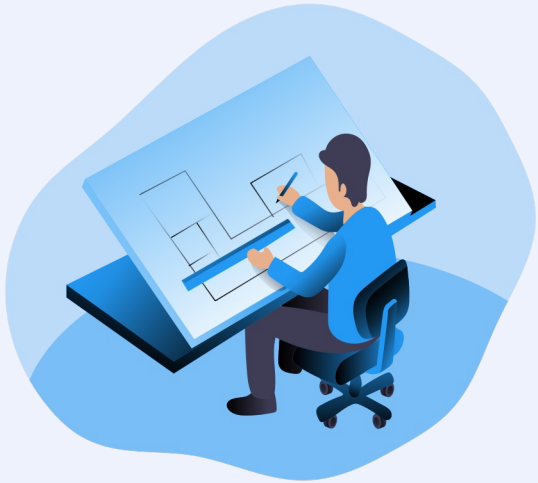
**Preparation of the JSALT 2020 workshop**

Gather ideas and experienced members

2020.06.11

**ESPnet-SE initial team**

Make development plans



2020.11.07

**ESPnet-SE was released.**

Provide basic speech enhancement (SE) and separation (SS) functions;

Compatible to `Asteroid`;

Support joint training with ASR

2022.07.19

**ESPnet-SE++ was released.**

Enrich SE and SS functions with SOTA techniques;

Support joint training with more tasks (ST, SLU, etc.)

01

# 发展历程

Milestones

非常感谢所有的贡献者！

Thanks to all the contributors!

**sw005320** (Shinji Watanabe) #6

507 commits 88,203 ++ 15,498 --

**Emrys365** (Wangyou Zhang) #9

259 commits 301,944 ++ 98,112 --

**LiChenda** (Chenda Li) #11

188 commits 28,689 ++ 19,930 --

**simpleoier** (Xuankai Chang) #16

104 commits 57,816 ++ 82,056 --

**popcornell** (Samuele Cornell) #23

73 commits 3,943 ++ 2,307 --

**shincling** (Jing Shi) #25

68 commits 12,711 ++ 2,696 --

100

**neillu23** (Yen-Ju Lu) #29

60 commits 554,223 ++ 546,156 --

**YoshikiMas** (Yoshiki Masuyama) #47

29 commits 1,333 ++ 771 --


**sas91** (Aswin Shanmugam Subramanian) #62

16 commits 1,151 ++ 137 --



**earthmanylf** (Linfeng Yu) #68

12 commits 1,149 ++ 147 --



**Johnson-Lsx** (Shaoxiong Lin) #80

10 commits 328 ++ 230 --

**nateanl** (Zhaoheng Ni) #87

7 commits 806 ++ 74 --

**fakufaku** (Robin Scheibler)

commits

# Features

## 02 特色功能

# ESPnet-SE

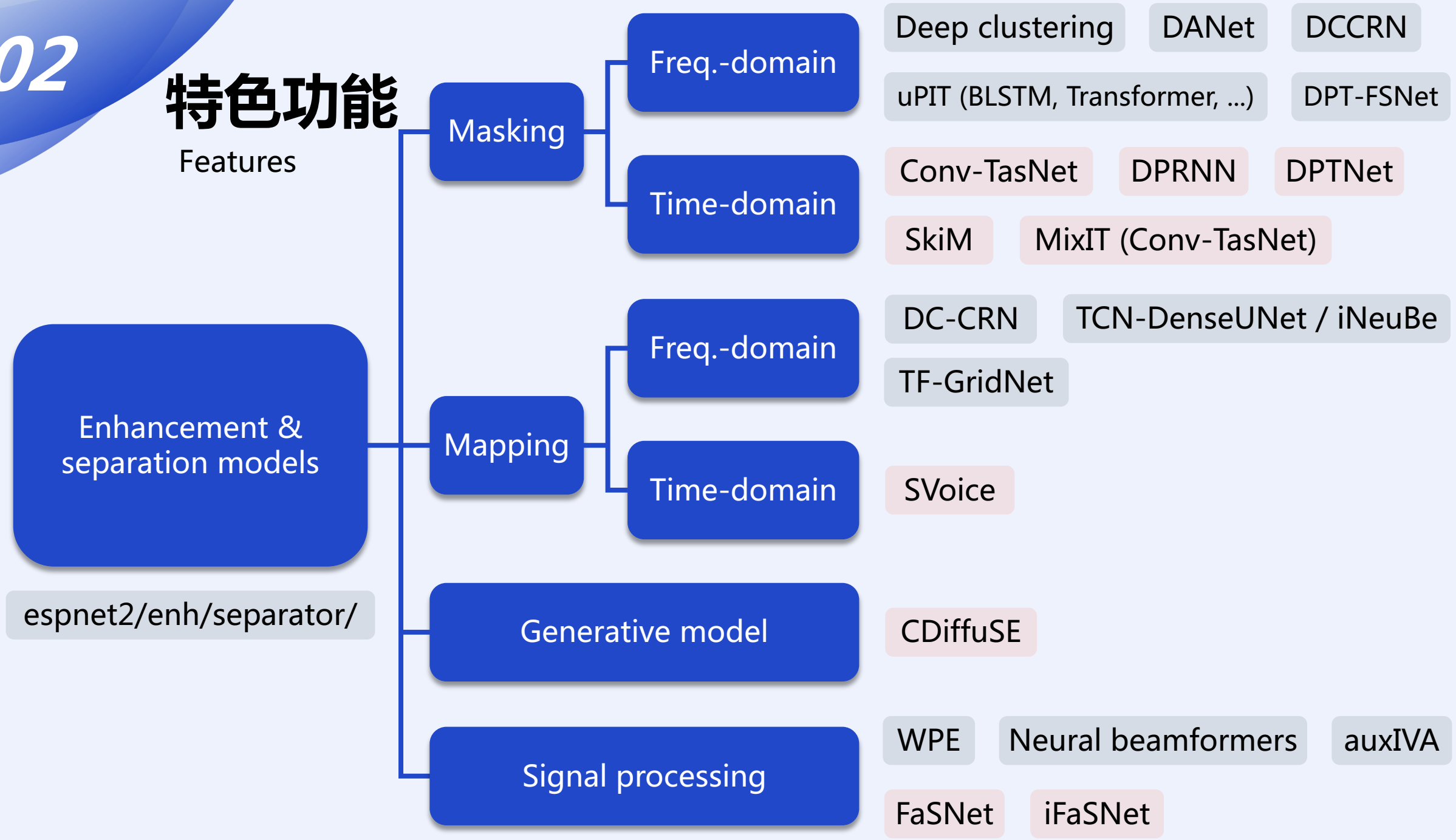
## Speech enhancement toolkit for robust speech recognition, translation, and understanding


https://github.com/espnet/espnet

02

# 特色功能

Features



02

# 特色功能

Features

Recipes for popular corpora

espnet2/egs2/*/enh1/

## Speech enhancement

chime4 conferencingspeech21

clarity21 dns_ins20 dns_icassp21

dns_ins21 l3das22 vctk_noisy

vctk_noisyreverb

## Speech separation

aishell4 lt_slurp_spatialized

librimix sms_wsj wham whamr

wsj0_2mix wsj0_2mix_spatialized

02

# 特色功能

Features

Unified interface for integration with more speech-to-text tasks

espnet2/tasks/enh_s2t.py

- Automatic speech recognition (ASR)
- Speech translation (ST)
- Spoken language understanding (SLU)

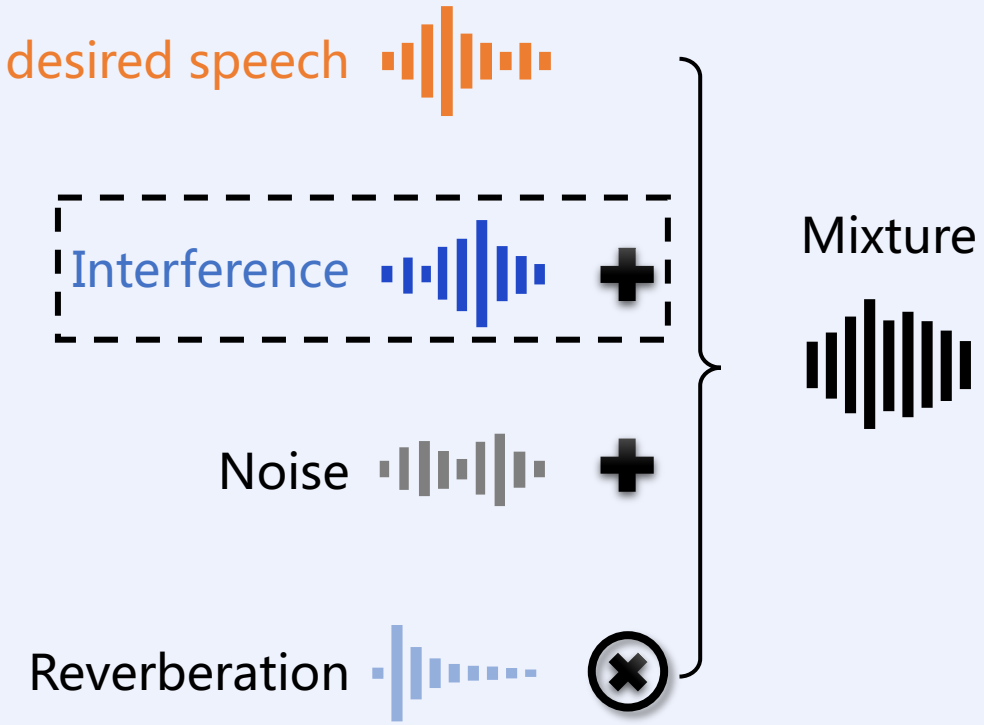
**Recipes**

- egs2/chime4/enh_asr1
- egs2/wsj0_2mix_spatialized/enh_asr1
- egs2/lt_slurp_spatialized/enh1
- egs2/slurp_spatialized/asr1

02

# 特色功能

Features




**Data simulation**

1 Offline mixing

`[recipe] local/data.sh`

```
$ ./run.sh --stage 1 --stop-stage 1
```

2 Dynamic mixing (on-the-fly)

`espnet2.tasks.enh.preprocessor_choices`

egs2/wsj0_2mix/enh1/conf/tuning/train_enh_skim_tasnet_noncausal_dm.yaml#L39-L44

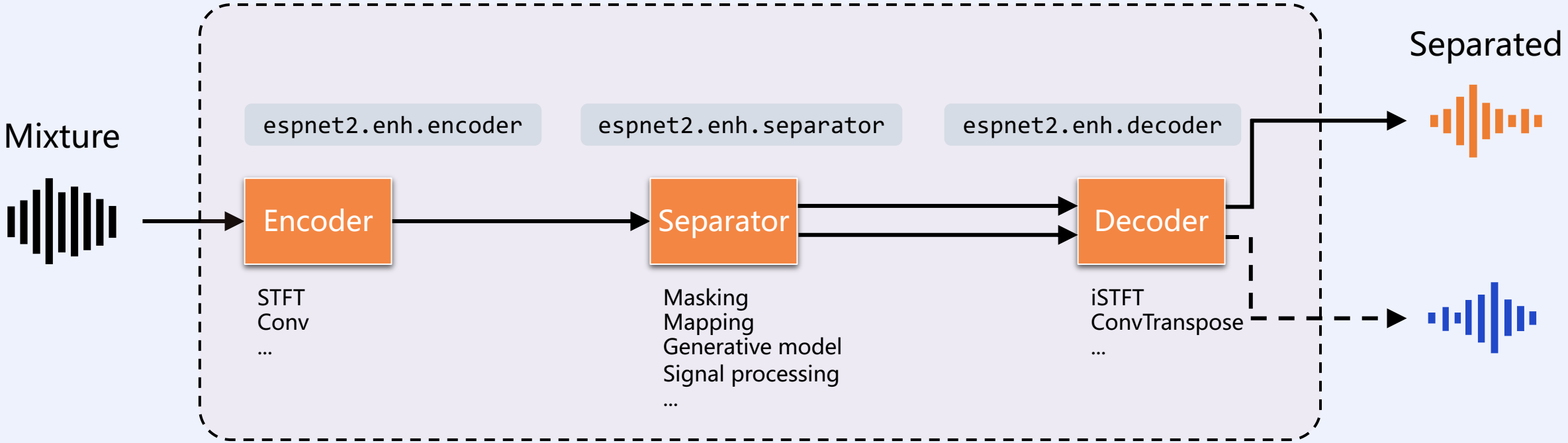
```yaml
preprocessor: dynamic_mixing
preprocessor_conf:
  ref_num: 2
  dynamic_mixing_gain_db: 0.0
  source_scp_name: "spk1.scp"
  mixture_source_name: "speech_mix"
```

02

# 特色功能

Features

## Model design `espnet2.enh.espnet_model`

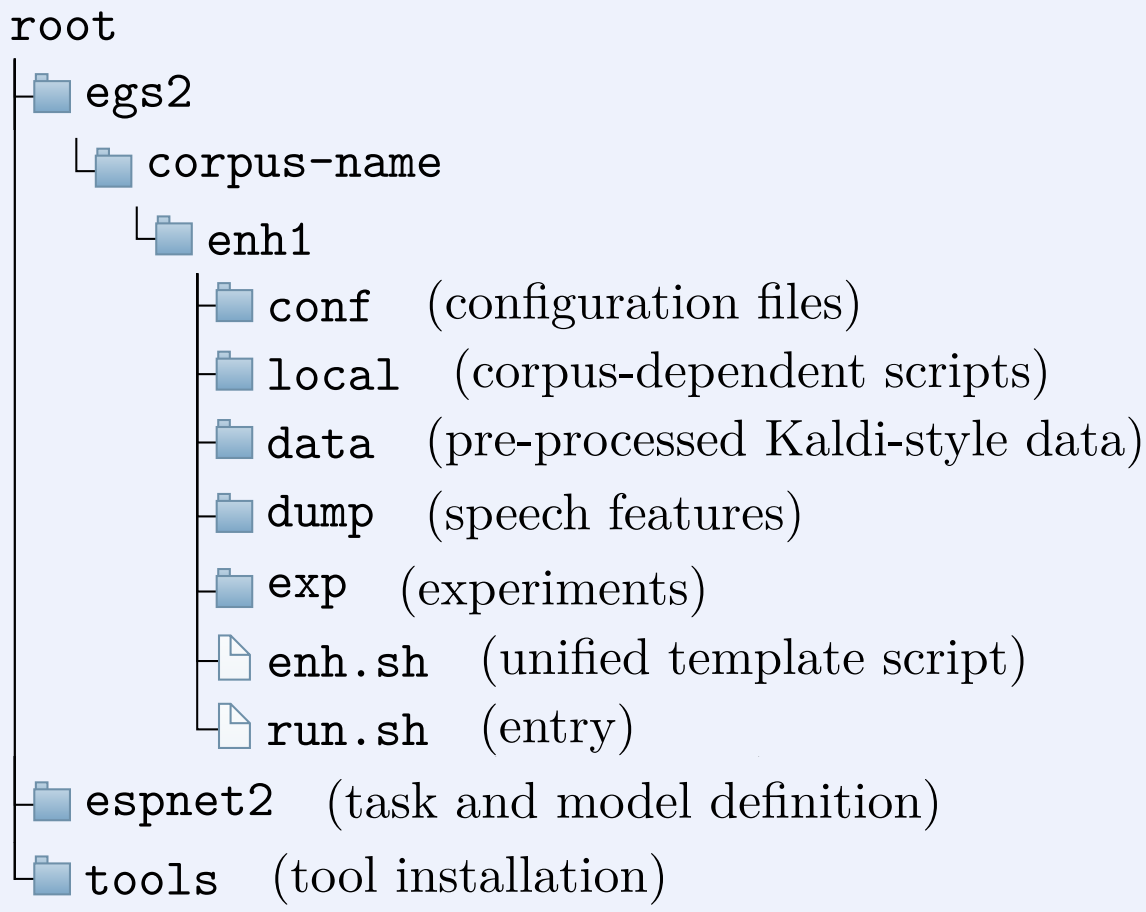


02

# 特色功能

Features

Code structure
(Recipe)



# Demos

## 03 样例展示

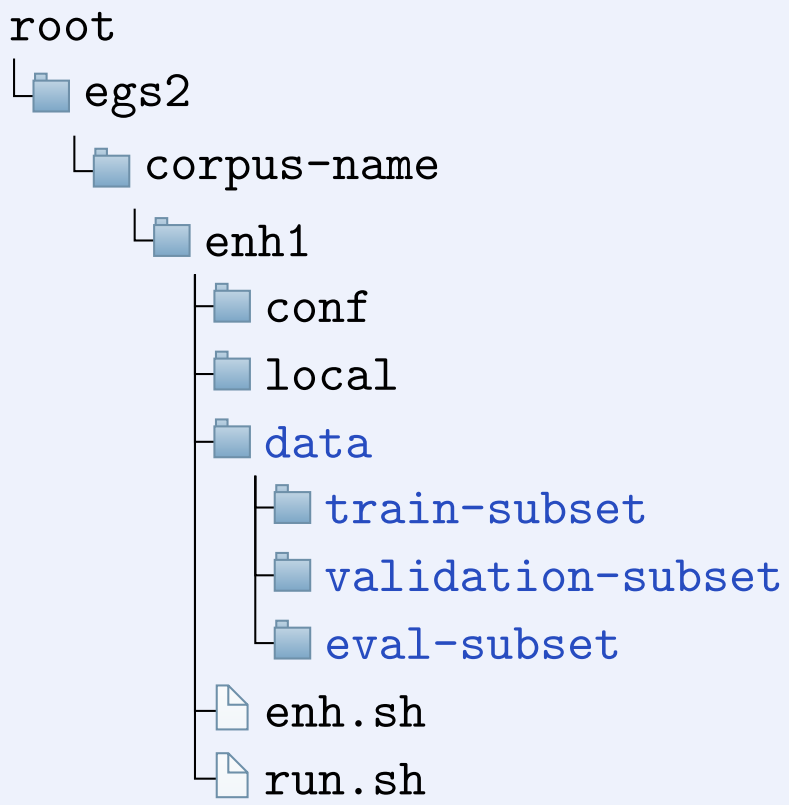
03

# 样例展示

Demos

## ① Run through the wsj0_2mix recipe

```
$ ./run.sh --stage 1 --stop-stage 1
```




Generate Kaldi-style data files for each subset in `data/`:

- wav.scp
- utt2spk
- spk2utt
- spk1.scp, spk2.scp, etc.
- text_spk1, text_spk2, etc. (if available)
- noise1.scp (if available)

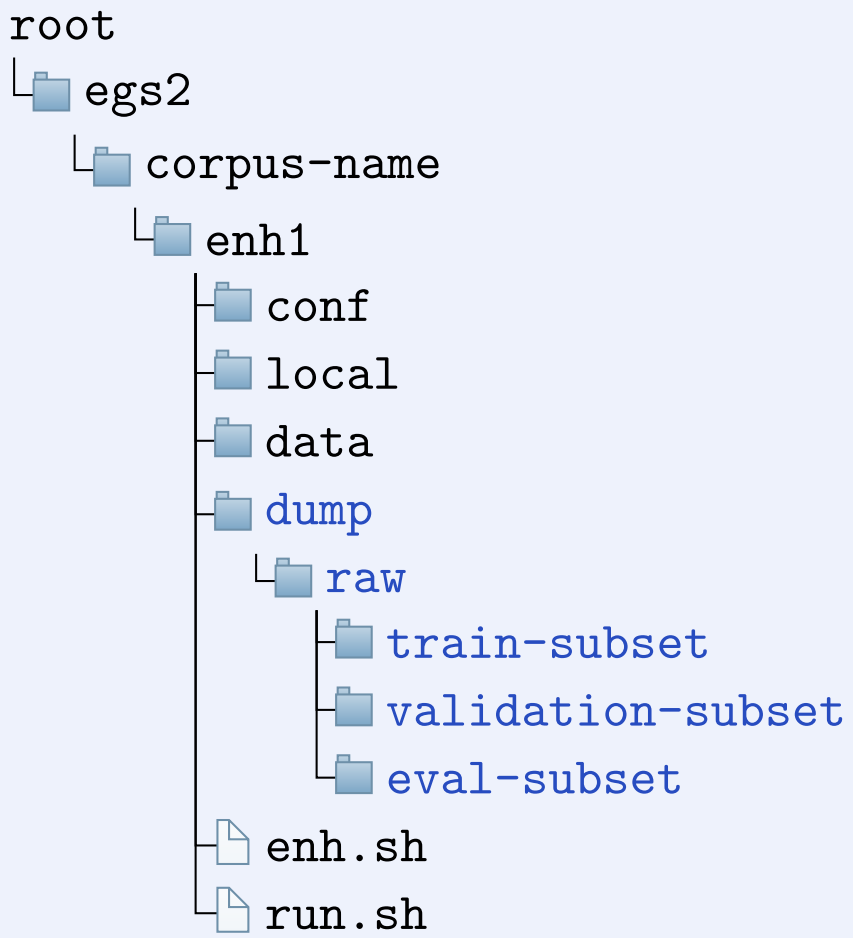
03

# 样例展示

Demos

## ① Run through the wsj0_2mix recipe

```
$ ./run.sh \
    --stage 3 \
    --stop-stage 4 \
    --ref_num 2
```




Prepare the final data files in `dump/`:

- unify audio format
- unify sampling rate
- exclude too short/long samples

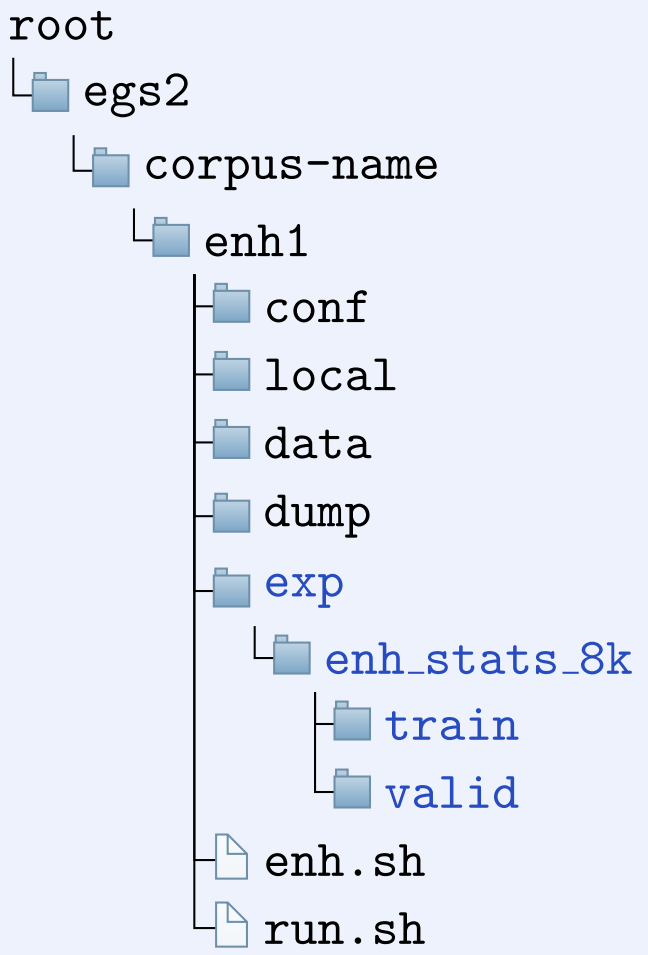
03

# 样例展示

Demos

## ① Run through the wsj0_2mix recipe

```
$ ./run.sh \
    --stage 5 \
    --stop-stage 5 \
    --ref_num 2 \
    --enh_config conf/tuning/train_enh_dprnn_tasnet.yaml
```




Prepare the statistics (sample length) of training and validation subsets in `exp/enh_stats_8k`:

- speech_mix_shape
- speech_ref1_shape, speech_ref1_shape, etc.
- feats_lengths_stats.npz
- feats_stats.npz

03

# 样例展示

Demos

## ① Run through the wsj0_2mix recipe

```
$ ./run.sh \
    --stage 6 \
    --stop-stage 6 \
    --ref_num 2 \
    --enh_config conf/tuning/train_enh_dprnn_tasnet.yaml \
    --num_nodes 1 \
    --ngpu 4
```

```
root
└─ egs2
   └─ corpus-name
      └─ enh1
         ├─ conf
         ├─ local
         ├─ data
         ├─ dump
         ├─ exp
         │  ├─ enh_stats_8k
         │  └─ enh_train_enh_dprnn_tasnet_raw
         │     ├─ config.yaml
         │     ├─ checkpoint.pth
         │     ├─ 96epoch.pth
         │     ├─ latest.pth -> 96epoch.pth
         │     ├─ valid.loss.best.pth -> 96epoch.pth
         │     ├─ train.log
         │     ├─ images
         │     └─ tensorboard
         ├─ enh.sh
         └─ run.sh
```

Start training in `exp/enh_train_enh_dprnn_tasnet_raw`:

- use a single node with 4 GPUs
- By default, only the latest and the best model checkpoints are stored.

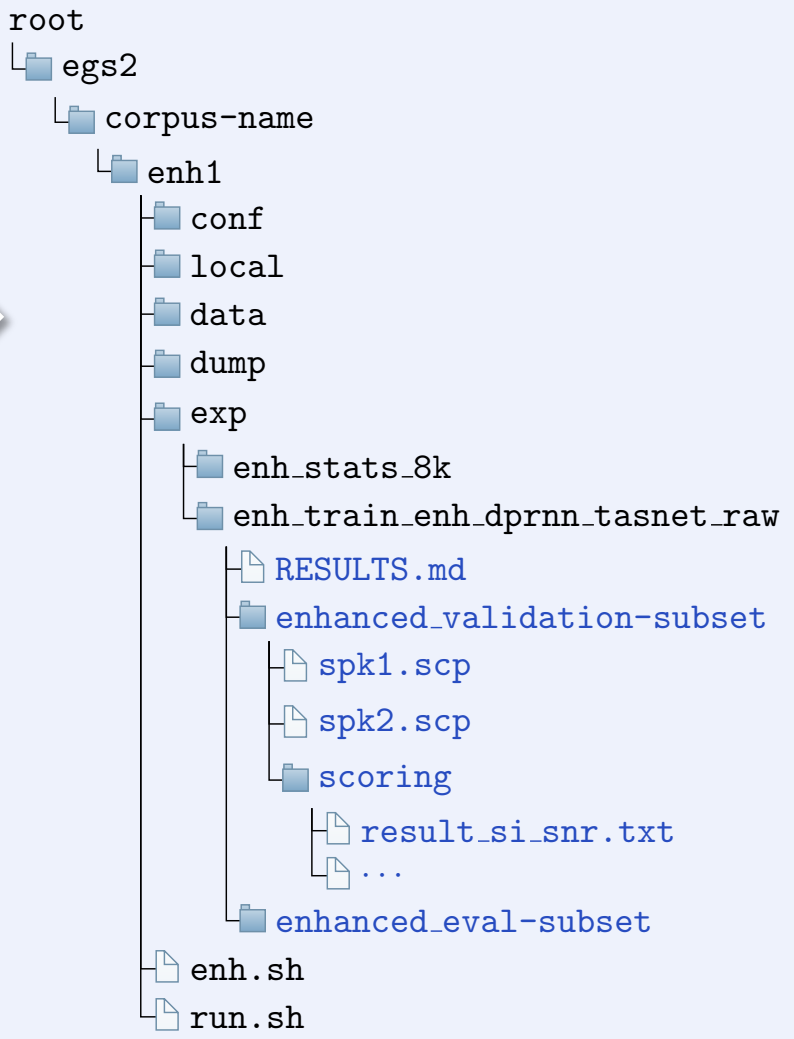
03

# 样例展示

Demos

## ① Run through the wsj0_2mix recipe

```
$ ./run.sh \
    --stage 7 \
    --stop-stage 8 \
    --ref_num 2 \
    --enh_config conf/tuning/train_enh_dprnn_tasnet.yaml \
    --scoring_protocol "STOI SDR SAR SIR SI_SNR" \
    --inference_model valid.loss.ave.pth \
    --inference_nj 32 \
    --gpu_inference false
```




Start inference and scoring on validation and evaluation subsets.

Intermediate results and enhanced audios are stored in `exp/enh_train_enh_dprnn_tasnet_raw/enhanced_*`.

Final results are written in `exp/enh_train_enh_dprnn_tasnet_raw/RESULTS.md`.

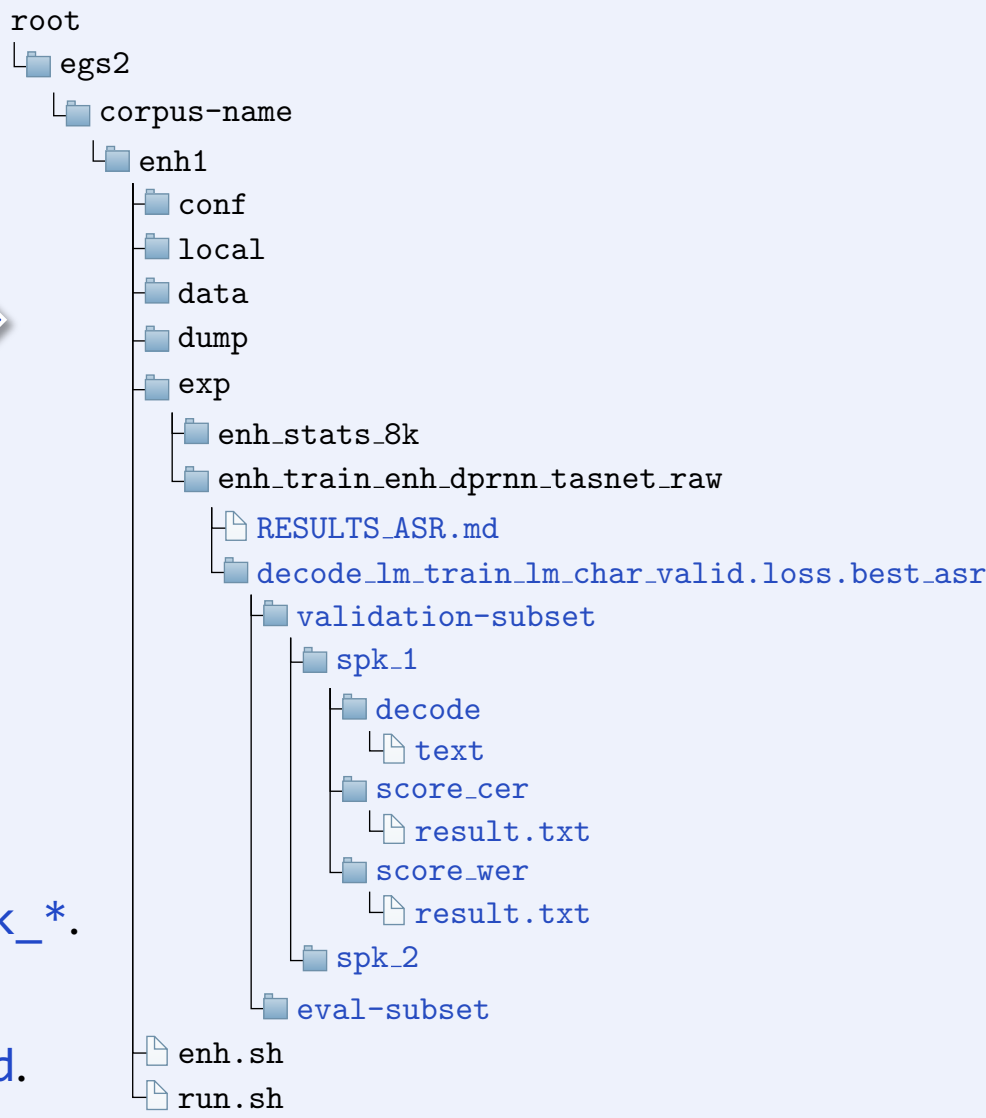
03

# 样例展示

Demos

## ① Run through the wsj0_2mix recipe

```
$ ./run.sh \
    --stage 9 \
    --stop-stage 10 \
    --ref_num 2 \
    --enh_config conf/tuning/train_enh_dprnn_tasnet.yaml \
    --score_with_asr true \
    --inference_asr_config conf/decode.yaml \
    --asr_exp /path/to/pretrained_asr_model_dir \
    --inference_asr_model valid.acc.best.pth \
    --lm_exp /path/to/pretrained_lm_model_dir \
    --inference_lm valid.loss.best.pth \
    --inference_nj 32 \
    --gpu_inference false
```




Decoding and scoring with a pretrained ASR model

Intermediate results are stored in
`exp/enh_train_enh_dprnn_tasnet_raw/*/[subset]/spk_*`.

Final ASR results are written in
`exp/enh_train_enh_dprnn_tasnet_raw/RESULTS_ASR.md`.

# 03

# 样例展示

Demos

## ② Hands-on demonstration

```
$ pip install espnet
$ pip install -q espnet_model_zoo
```

```
from espnet_model_zoo.downloader import ModelDownloader
from espnet2.bin.enh_inference import SeparateSpeech
import soundfile

tag = "espnet/Wangyou_Zhang_chime4_enh_train_enh_conv_tasnet_raw"
d = ModelDownloader()
cfg = d.download_and_unpack(tag)

separate_speech = SeparateSpeech(
      train_config=cfg["train_config"],
      model_file=cfg["model_file"],
      normalize_output_wav=True,
)

input_wav, sr = soundfile.read("input.wav")
enhanced_wav = separate_speech(input_wav[None, ...], fs=sr)[0]
```

A few steps for SE:

1. Install ESPnet with pip
2. Download and load pre-trained models
3. Inference with convenient APIs

input_wav:

enhanced_wav:

# 03

# 样例展示

Demos

## ② Hands-on demonstration

```
$ pip install espnet
$ pip install -q espnet_model_zoo
```

```
from espnet_model_zoo.downloader import ModelDownloader
from espnet2.bin.enh_inference import SeparateSpeech
import soundfile

tag = "lichenda/wsj0_2mix_skim_noncausal"
d = ModelDownloader()
cfg = d.download_and_unpack(tag)

separate_speech = SeparateSpeech(
      train_config=cfg["train_config"],
      model_file=cfg["model_file"],
      normalize_output_wav=True,
)

input_wav, sr = soundfile.read("mix.wav")
separated_wavs = separate_speech(input_wav[None, ...], fs=sr)
```

A few steps for SE:

1. Install ESPnet with pip
2. Download and load pre-trained models
3. Inference with convenient APIs

Utt-level speech separation based inference

input_wav:

separated_wavs[0]:

separated_wavs[1]:

# 03 样例展示

## ② Hands-on demonstration

Demos

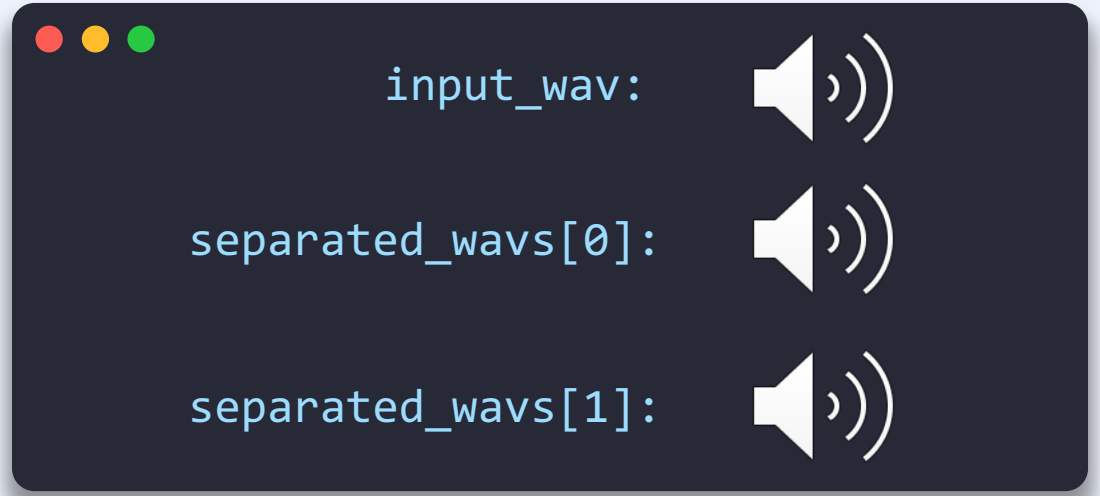
```
$ pip install espnet
$ pip install -q espnet_model_zoo
```

```
from espnet_model_zoo.downloader import ModelDownloader
from espnet2.bin.enh_inference import SeparateSpeech
import soundfile

tag = "lichenda/wsj0_2mix_skim_noncausal"
d = ModelDownloader()
cfg = d.download_and_unpack(tag)

separate_speech = SeparateSpeech(
      train_config=cfg["train_config"],
      model_file=cfg["model_file"],
      normalize_output_wav=True,
      normalize_segment_scale=True,
      segment_size=2.4, # process 2.4-sec audio at a time
      hop_size=0.8,     # 0.8-sec window hop
)
input_wav, sr = soundfile.read("mix.wav")
separated_wavs = separate_speech(input_wav[None, ...], fs=sr)
```

A few steps for SE:

1. Install ESPnet with pip
2. Download and load pre-trained models
3. Inference with convenient APIs

Continuous speech separation based inference



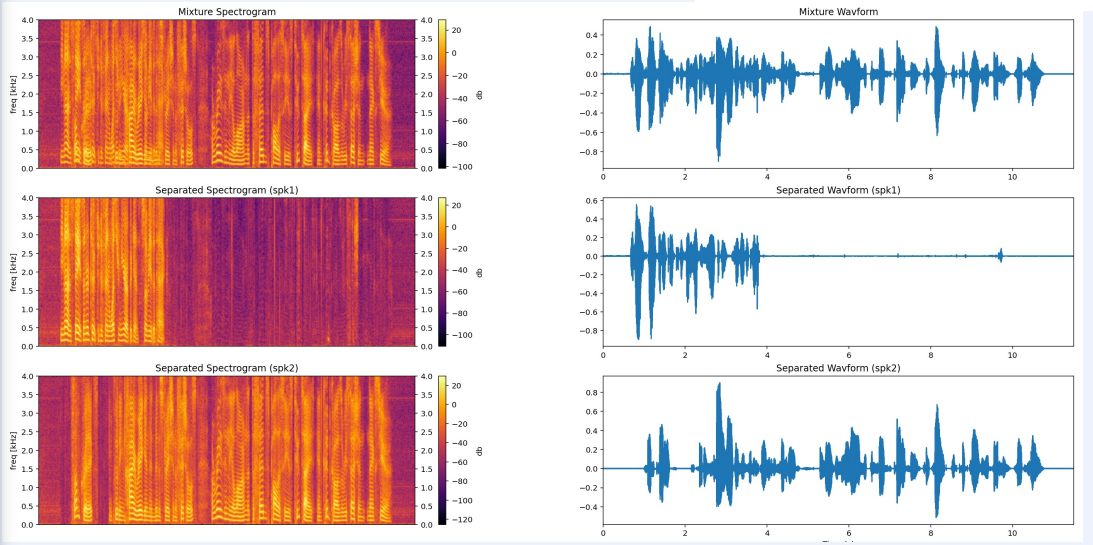
# 03

# 样例展示

Demos

## ② Hands-on demonstration

```python
from espnet_model_zoo.downloader import ModelDownloader
from espnet.asr.asr_utils import plot_spectrogram
from espnet2.bin.asr_inference import Speech2Text


tag = "lichenda/wsj_asr_train_asr_transformer_raw_char_8k"
d = ModelDownloader()
speech2text = Speech2Text(**d.download_and_unpack(tag))

text_est = [None, None]
text_est[0], *_ = speech2text(separated_wavs[0].squeeze())[0]
text_est[1], *_ = speech2text(separated_wavs[1].squeeze())[0]
```




```
--------------- Text 1 ---------------
REF: SOME CRITICS INCLUDING HIGH REAGAN ADMINISTRATION OFFICIALS ARE RAISING THE ALARM THAT THE FED'S POLICY IS TOO TIGHT AND COULD CAUSE A RECESSION NEXT YEAR
HYP: SOME CRITICS INCLUDING HIGH REAGAN ADMINISTRATION OFFICIALS ARE RAISING THE ALARM THAT THE FED'S POLICY IS TOO TIGHT AND COULD CAUSE A RECESSION NEXT YEAR
ERR:
Edit Distance = 0

--------------- Text 2 ---------------
REF: THE UNITED STATES UNDERTOOK TO DEFEND WESTERN EUROPE AGAINST SOVIET ATTACK*****
HYP: THE UNITED STATES UNDERTOOK TO DEFEND WESTERN EUROPE AGAINST SOVIET ATTACK SUCH
ERR:                                                                          IIIII
Edit Distance = 5
```

# 03 样例展示 ② Hands-on demonstration

Demos

```
$ cd <espnet-root>/tools
$ make TH_VERSION=1.10.1 CUDA_VERSION=11.3
```

```
$ scripts/utils/enhance_dataset.sh \
    --spk_num 1 \
    --gpu_inference true \
    --inference_nj 4 \
    --fs 16k \
    --id_prefix "" \
    dump/raw/et05_real_isolated_6ch_track \
    data/et05_real_isolated_6ch_track_enh \
    exp/enh_train_enh_beamformer_mvdr_raw/valid.loss.best.pth
```

```
$ scripts/utils/calculate_speech_metrics.sh \
    --ref_channel 0 \
    --nj 4 \
    dump/raw/et05_real_isolated_6ch_track/wav.scp \
    data/et05_real_isolated_6ch_track_enh/wav.scp \
    SNR \
    data/et05_real_isolated_6ch_track_enh/snr.scp
```

Portable scripts for processing the entire dataset

```
root
└── egs2
    └── chime4
        └── enh1
            ├── scripts -> ../../TEMPLATE/enh1/scripts
            ├── data
            │   └── et05_real_isolated_6ch_track_enh
            │       ├── wavs
            │       ├── wav.scp
            │       ├── text
            │       ├── utt2spk
            │       ├── spk2utt
            │       ├── utt2uniq
            │       └── snr.scp
            ├── dump
            │   └── raw
            │       └── et05_real_isolated_6ch_track
            └── exp
                └── enh_train_enh_beamformer_mvdr_raw
                    └── valid.loss.best.pth
```

# 03 样例展示

Demos

## ③ Useful links

ESPnet model hub:

https://huggingface.co/espnet

https://zenodo.org/communities/espnet

Demo pages:

https://colab.research.google.com/drive/1fjRJCh96SoYLZPRxsjF9VDv4Q2VoIckI (ESPnet-SE)

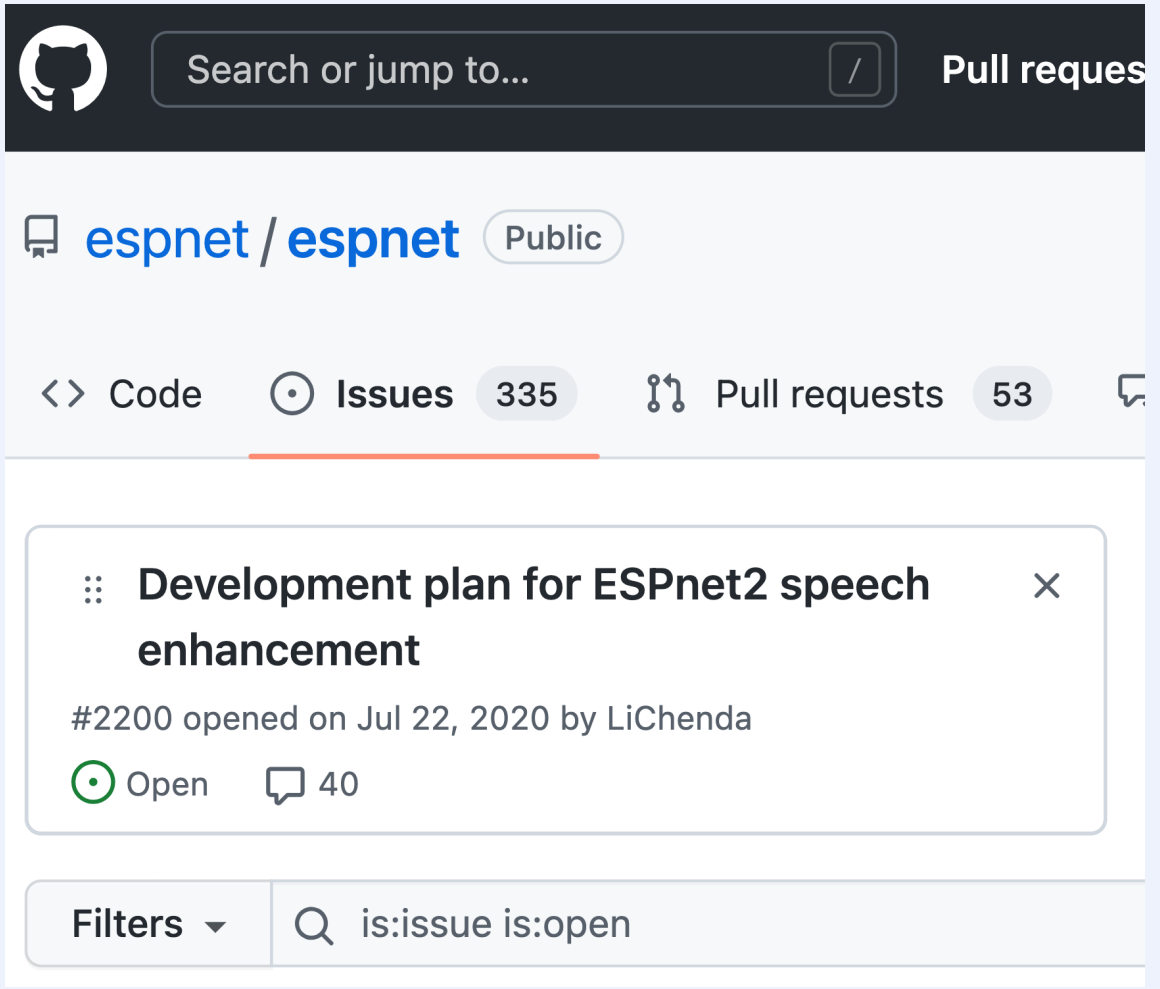
https://colab.research.google.com/drive/1hAR5hp8i0cBIMeku8LbGXseBBaF2gEyO (ESPnet-SE++)

Documentation:

https://github.com/espnet/espnet/blob/master/egs2/TEMPLATE/enh1/README.md

Suggestions and discussion:

https://github.com/espnet/espnet/issues/2200



# Outlook

## 04 未来展望

04

# 未来展望

Outlook

期待更多的贡献者！

Any contribution is welcome!

Target speaker extraction (TSE) / Personalized speech enhancement (PSE)

Recipes/Models for more realistic scenarios (far-field, real-time/streaming, etc.)

Further integration with other tasks (e.g. TTS，Multi-talker ASR #4753 )

More signal processing methods

# THANK YOU

汇报人：钱彦旻、李晨达、张王优

汇报时间：2022.11.13


上海交通大學
SHANGHAI JIAO TONG UNIVERSITY