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

Yen-Ju Lu∗1, Xuankai Chang∗2, Chenda Li3, Wangyou Zhang3, Samuele Cornell4, Zhaoheng Ni3, Yoshiki Masuyama2,7, Brian Yan2, Robin Scheibler5, Zhong-Qiu Wang2, Yu Tsao1, Yanmin Qian3, Shinji Watanabe2

1Academia Sinica, Taipei 2Carnegie Mellon University, USA 3Shanghai Jiao Tong University, Shanghai 4Università Politecnica delle Marche, Italy 5Meta AI, USA 6LINE Corporation, Japan 7Tokyo Metropolitan University, Japan

Abstract

This paper presents recent progress on integrating speech separation and enhancement (SSE) into the ESPnet toolkit. Compared with the previous ESPnet-SE work, numerous features have been added, including recent state-of-the-art SPEH Enhancement models with their respective training and evaluation recipes. Importantly, a new interface has been designed to flexibly combine speech enhancement front-ends with other tasks, including automatic speech recognition (ASR), speech translation (ST), and spoken language understanding (SLU). To showcase such integration, we performed experiments on carefully designed synthetic datasets for noisy-reverberant multi-channel ST and SLU tasks, which can be used as benchmark corpora for future research. In addition to these new tasks, we also use CHiME-4 and WSJ0-2Mix to benchmark multi-end single-channel SE approaches. Results show that the integration of SE front-ends with back-end tasks is a promising research direction even for tasks besides ASR, especially in the multi-channel scenario. The code is available online at https://github.com/ESPnet/ESPnet. The multi-channel ST and SLU datasets, which are another contribution of this work, are released on HuggingFace.

Index Terms: speech enhancement, speech recognition, speech translation, spoken language understanding

1. Introduction

Speech separation and enhancement (SSE) aims at extracting a target speech signal from noise, reverberation, and interfering speakers. It is essential to robust speech recognition [1, 2], assistive hearing [3], and robust speech recognition [4]. SSE methods have benefited greatly from the recent development of deep learning (DL) approaches, and DL-based methods are now the de-facto standard. This opens up the possibility for end-to-end integration of SSE methods with many downstream speech processing back-end tasks [e.g., automatic speech recognition (ASR), keyword spotting, and speech translation (ST), to name a few]. In fact, many works have been recently exploring this research direction, mainly concerning ASR [5–8].

To accelerate research in SSE, ESPnet-SE toolkit [9] was developed and currently supports multiple state-of-the-art enhancement approaches and various corpora. In the meantime, there is currently a lot of effort and interest for robust ASR, ST, and spoken language understanding (SLU) in noisy and possibly distant speech scenarios [10, 11], such as ones encoun-

tered by smart-speaker devices. This motivated us to extend ESPnet-SE into ESPnet-SE++, which has a re-designed interface focused on modularity. This new interface allows for a seamless combination of different front-end SSE models with various downstream tasks such as ASR, ST and SLU.

The contributions of ESPnet-SE++ are summarized below:

• We significantly extend ESPnet-SE by providing new recipes for several enhancement corpora and challenges.

• In addition to new recipes, several state-of-the-art single-channel and multi-channel enhancement approaches have been added. Including unsupervised separation [12] and generative speech enhancement [13] approaches.

• A redesigned modular interface allows a flexible, “plug-and-play” combination of SE front-ends with different ESPnet back-end tasks such as ASR, SLU, ST, etc. As we showcase in Section 4 for ST and SLU, this easily allows to test and possibly fine-tune (as done for SLU) the same pre-trained front-end for multiple applications.

• We develop two multi-channel noisy-reverberant datasets derived from SLURP and Libri-Trans, respectively. We simulate potential applications of SLU and ST in a distant speech setting and showcase how tight integration of SE front-end with the back-end opens up promising research directions for other tasks besides ASR.

An extensive experimental evaluation is performed to showcase the flexibility of ESPnet-SE++. In particular, we conducted different experiments using four different datasets: CHiME-4, WSJ0-2mix, and the two purposefully developed SLU and ST multi-channel datasets mentioned previously. Results show that speech enhancement can improve the performance in a wide variety of downstream tasks.

2. Related Works

In this section, we briefly compare ESPNet-SE with other open-source deep learning-based speech enhancement and separation toolkits, such as nussl (North-western University Source Separation Library) [14], Onssen (An Open-source Speech Separation and Enhancement Library) [15], Asteroid (Audio source separation on Steroids) [16], and SpeechBrain [17]. While nussl provides several state-of-the-art SE methods, the data preparation and experiments of nussl and Onssen are not easily configurable from the command line [16]. On the other hand, Asteroid and SpeechBrain provide a whole pipeline from data preparation to enhancing and evaluating the testing speech.
However, Asteroid is exclusively designed for SE front-end processing. On the other hand, SpeechBrain supports both SSE front-end and back-end (e.g. ASR) techniques along with corresponding recipes but does not yet integrate multiple speech processing tasks into a single recipe. Although the application of SE can already be beneficial, especially in the multi-channel case, various studies suggest that joint-optimization with downstream tasks can further boost the performance [10, 18, 19]. We summarize the difference between ESPnet-SE and ESPnet-SE++ in Figure 1. ESPnet-SE++ owns 20 SSE recipes, with 24 different enhancement/separation models.

### 3. Features of ESPnet-SE++

ESPnet-SE covers a wide range of speech enhancement/separation models and recipes, including (1) single-channel and multi-channel; (2) single-source and multi-source; (3) time-domain and frequency-domain enhancement models. ESPnet-SE++, is extended to support the latest state-of-the-art models and the latest enhancement/separation recipes, and also provides the combination of enhancement with other downstream tasks, including ASR, ST, and SLU. In addition, ESPnet-SE++ supports the native complex datatype built by PyTorch for complex time-frequency domain processing instead of relying on custom workarounds built on top of PyTorch real-valued tensors.

#### 3.1. Recipes

ESPnet-SE provides various recipes for several SSE benchmark corpora which are derived from WSJ [20] and LibriSpeech [21] corpora. ESPnet-SE++, on top of these, adds 10 new recipes, including speech enhancement/separation corpora: Voicebank-Demand [22], speech enhancement challenges: DNS (Interspeech 2020 [23], ICASSP 2021 [24], and Interspeech 2021 [25]), ConferenceSpeech 2021 [26], Clarity 2021 [27], and LSDAS22 [28]. In addition, we develop a new recipe that enables an all-in-one combination of multiple speech processing components with a front-end SSE module. This includes robust ASR (CHiME-4, ST (Libri-Trans [29] mixture), and SLU (SLURP [30] mixture) datasets. These recipes contain the data preparation steps for both enhancement and downstream speech processing tasks, the joint-training of the front-end and back-end models, and the output of enhanced/separated speech with the recognition/translation/understanding results.

The function of each stage in the unified pipeline is depicted in Figure 2.

#### 3.2. Models

In ESPnet-SE++, we add various classical and state-of-the-art supervised SSE models, including (1) single-channel models: Deep Clustering [31], Deep Attractor Network [32], DCCRN [33], DC-CRN [34], SkM [35], DPT-FFSN [36], Svoice [37], and (2) multi-channel models: neural beamformer with more supported types (c.f. Section 3.4), FaSNet [38], and iNeuBe [39, 40]. In addition, we further extend the ESPnet-SE framework to support various training and inference procedures, including (3) generative enhancement model, CDiffuSE [13], and (4) unsupervised separation model, MixIT [12].

#### 3.3. Training Objectives

The training objective can vary a lot between different speech enhancement and separation tasks. To make it more flexible and customizable, we disentangle the training objective into two abstract classes: a criterion and a wrapper.

The criterion is an implementation of a simple loss function, e.g. SI-SNR [41, 42], CI-SDR [43, 44], mean square error (MSE) on time-frequency masks, etc. These losses usually take both target and estimated tensors as input and output a scalar loss value.

The wrapper instead deals with a post-processing of the criterion output. For example, the procedure to find the best permutation in the PIT [45] algorithm is implemented as a wrapper. We have also added MixIT [12] wrapper for unsupervised speech separation. This modular design allows researchers to add complex training objectives conveniently by writing custom wrapper classes and combining it with a suitable criterion.

Furthermore, ESPnet-SE++ supports multiple training objectives. Different training objectives can be combined with different weights in a multi-task learning (MTL) fashion. The specific implementation of training objectives, including their wrapper, criterion and MTL weight, can be selected and configured directly in the ESPnet yaml configuration file.

#### 3.4. Single- and Multi-channel Speech Enhancement

As mentioned above, ESPnet-SE++ adds several models on top of ESPnet-SE. For single-channel models, DCCRN [33] and DC-CRN [34] are added. To catch up with the latest development of deep learning-based multi-channel speech enhancement systems, we further added several multi-channel SSE models, including FaSNet [38], DC-CRN [34], and iNeuBe [39, 40]. The first one is a time-domain multi-channel model, while the other two operate in the complex time-frequency domain. In addition, there are also various
Table 1: Results of single- and multi-channel speech enhancement approaches on CHiME-4.

| Models                  | PESQ | STOI | SI-SNR (dB) |
|-------------------------|------|------|-------------|
| No processing           | 2.18 | 0.870| 7.51        |
| Wang et al. [50]        | 3.68 | 0.986| 22.00       |

Single-channel Models

| Models                  | PESQ | STOI | SI-SNR (dB) |
|-------------------------|------|------|-------------|
| Conv-TasNet (baseline)  | 2.53 | 0.892| 11.57       |
| *DCCRN                  | 2.59 | 0.895| 12.57       |
| *DC-CRN                 | 2.43 | 0.880| 11.59       |

Multi-channel Models

| Models                  | PESQ | STOI | SI-SNR (dB) |
|-------------------------|------|------|-------------|
| MVDR                    | 2.61 | 0.954| 13.98       |
| MVDR_Souden             | 2.66 | 0.954| 15.25       |
| *wMPDR                  | 2.60 | 0.951| 13.53       |
| Neural Beamformer       | 2.64 | 0.951| 15.24       |
| WPD                     | 2.60 | 0.950| 13.43       |
| WPD_Souden              | 2.64 | 0.950| 14.89       |
| *SDW-MWF                | 2.43 | 0.922| 11.87       |
| *r1-MWF                 | 2.67 | 0.953| 15.08       |
| *AuxIVA-ISS             | 2.49 | 0.900| 10.34       |
| *DC-CRN                 | 2.95 | 0.948| 17.04       |
| *FaSNet                 | 2.70 | 0.935| 14.83       |
| *iNeuBe (DNN)           | 3.24 | 0.969| 19.52       |

The symbol * denotes models newly-added to ESPnet-SE++.

Table 2: Results of single-channel speech separation models on WSJ0-2mix.

| Models                  | PESQ | STOI | SI-SNR (dB) |
|-------------------------|------|------|-------------|
| No processing           | 2.01 | 0.738| 0.00        |
| Conv-TasNet (baseline)  | 3.25 | 0.953| 15.94       |
| DPRNN-TasNet            | 3.47 | 0.968| 17.91       |
| *Deep Clustering        | 2.15 | 0.845| 7.91        |
| *Deep Attractor Network | 2.68 | 0.893| 10.30       |
| *DC-CRN                 | 3.11 | 0.935| 13.01       |
| *SkiM                   | 3.47 | 0.966| 18.45       |
| *MixIT (Conv-TasNet)    | 3.00 | 0.938| 13.50       |

The symbol * denotes models newly-added to ESPnet-SE++.

Table 3: WER for system combination of speech enhancement with speech recognition on CHiME-4 corpus.

| Models                  | SIMU (%) | REAL (%) |
|-------------------------|----------|----------|
| No processing           | 19.7     | 18.0     |

Single-channel Models

| Models                  | SIMU (%) | REAL (%) |
|-------------------------|----------|----------|
| Conv-TasNet             | 17.4     | 15.2     |
| DCCRN                   | 16.3     | 15.8     |

Multi-channel Models

| Models                  | SIMU (%) | REAL (%) |
|-------------------------|----------|----------|
| FaSNet                  | 15.7     | 23.8     |
| Neural Beamformer       | 10.8     | 13.7     |
| *iNeuBe (DNN)           | 9.0      | 35.8     |

4. Combination tasks of ESPnet-SE++

ESPnet-SE++ allows for a tight and easy integration of SE front-end processing techniques and back-end tasks as introduced in Sec. 3.1. To showcase this, we provide three combined recipes, CHiME-4, LT-S, and SLURP-S, with speech enhancement (SE) front-end subtask, followed by a back-end ASR/ST/SLU subtask. We performed several experiments with different techniques to assess if and how SE could improve the results of the back-end tasks even when multi-condition training is employed.

4.1. Data Simulation

We created two multi-channel noisy-reverberant datasets based on SLURP and Libri-Trans (LT): SLURP-S and LT-S, where S stands for spatialized. We augmented the original SLURP and LT datasets using room impulse responses generated via Pyroomacoustics [53], and noises from FSD50k [54], SINS [55]. In detail, for each original utterance from SLURP and LT we simulate a smart-speaker scenario where the target signal is captured by a 4-microphone circular array with a diameter of 10 cm. For each utterance, we sample a room size from uniform distribution \( U(10, 100) \) m\(^2\) and a reverberation time (T60) from \( U(0.2, 0.6) \) s, typical of most indoor settings [56].

\(^1\)https://github.com/kamo-naoyuki/pytorch_complex

\(^2\)https://pytorch.org/audio/main/functional#multi-channel
Room height is sampled from $U(2.5, 4)$ m. The target speech is contaminated by 1 up to 4 point-source localized noises from SINS. Diffuse noise is simulated using the technique outlined in [57]. The positions of the array, point-source noises and target are sampled randomly in the virtual room with the constraint of being at least 0.5 m apart from each other and the walls. The Signal-to-Noise ratio (SNR) for target versus point-source noises is sampled from $U(0, 15)$ dB, while for diffuse noise is sampled from $U(12, 35)$ dB.

Regarding SLURP, since it already comprises noisy-reverberant examples, we use DNSMOS model [58] to select only a subset of sufficiently clean utterances (with estimated $B_\text{AK} \geq 3.2$) from training real, development and test, to use for SLURP-S. This process leaves 24497 utterances for training real, and 6387 and 6099 for development and test respectively. No utterance is discarded from training synthetic. The discarded utterances are kept but treated as single-channel examples with no oracle SE supervision, and are left for possible future work on semi/self-supervised joint SE+SLU. Instead, LT-S retains the same number of utterances of the original LT as it features only clean speech.

Finally, in order to assess generalization to unseen noise conditions, we also generate, for both datasets, an additional test set with diffuse noise derived from QUT [59] instead of SINS.

### 4.2. Combination of SE with ASR on CHiME-4

In Table 3 we report the results obtained by combining different SE front-ends with an ASR model on the CHiME-4 corpus [11]. CHiME-4 consists of both real and simulated noisy speech signals and is particularly suited for this analysis. We used a transformer encoder-decoder as the E2E ASR model that is pre-trained using the official ESPNet CHiME-4 recipe. The SE models are instead pre-trained using simulated data. The performance of these systems in terms of signal-based metrics is summarized in Sec. 3.4. After pre-training, the entire SE+ASR system is fine-tuned together with both SE and ASR losses. On the simulated dataset, all SE+ASR systems improve performance with respect to using solely the ASR model. This result confirms the effectiveness of jointly fine-tuning SE and ASR systems for robust speech recognition. Meanwhile, some models resulted in worse WERs on the real data. This is a commonly observed phenomenon when evaluating SE models on simulated and real data [19, 60], especially when they are not jointly optimized with the downstream ASR task. The neural beamformer achieved the best performance in both simulated and real sets.

### 4.3. Combination of SE with ST on LT-S

In the left part of Table 4, we explore the combination of SE with a ST task performed on the simulated LT-S dataset outlined in Section 4.1. For the ST model, we used a conformer-encoder transformer-decoder as the back-end model. The ST model was trained on the original clean Libri-Trans dataset and achieved a BLEU score of 16.7 on the test set. The front-end SE models were pre-trained using both LT-S and SLURP-S datasets using the anechoic clean speech as target (thus performing joint denoising and dereverberation). For the evaluation, we directly concatenate the pretrained SE models with the ST model in an end-to-end manner. Note that the whole system was not fine-tuned. We can see that the addition of the SE front-end generally improves the performance against the noisy speech even without fine-tuning. Among the SE models considered, both iNeuBe models based on DNN2 and multi-frame multi-channel Wiener filter (mfmcwf), bring the largest performance gain. In the single-channel case, DCCRN, as expected, outperforms Conv-Tasnet.

### 4.4. Combination of SE with SLU on SLURP-S

In the right part of Table 4, we explore instead the combination of SE with a SLU back-end and report the intent classification accuracy. Regarding the SLU model, we used the conformer-encoder transformer-decoder trained with multi-condition training (clean and noisy-reverberant) on the SLURP-S corpus. After SLU back-end pre-training, we fine-tuned the SE and SLU systems together. The SE models are the same as used in the ST experiments in Sec. 4.3. Notably, we can see that in this instance, contrary to what is observed on CHiME-4 which is arguably less acoustically challenging than SLURP-S, Conv-TasNet and FaSNet lead to degraded performance. On the other hand, both DCCRN and iNeuBe are able to outperform the multi-condition SLU model. This confirms that models that rely on complex spectral mapping [40, 50] are more robust in challenging acoustical conditions.

### 5. Conclusions

In this work we presented the ESPnet-SE++ toolkit. Built on top of ESPNet-SE, it includes new state-of-the-art models, losses, and recipes for speech enhancement corpora and challenges in various scenario. The toolkit interface has also been overhauled and improved with greater modularity. This allows a flexible combination of speech enhancement with ESPnet back-end tasks such as ASR, ST, and SLU. As an additional contribution, to showcase such integration of front-end and back-end components, we developed two multi-channel noisy-reverberant corpora based on SLURP and Libri-Trans. Experiments show that the use of SE front-end models improves both signal-based evaluation metrics and back-end tasks such as ASR, ST, and SLU, even when the back-end is trained on a multi-condition fashion. Future work could investigate further integration of different SE techniques with back-end tasks, including self-supervised pre-training via MixIT and generative enhancement models, with the goal of improving generalization to unseen acoustical conditions.

### 6. References

[1] Z. Wang et al., “Rank-1 constrained multichannel Wiener filter for speech recognition in noisy environments,” Computer Speech & Lang., vol. 49, pp. 37–51, 2018.
