NASTAR: Noise Adaptive Speech Enhancement with Target-Conditional Resampling

Chi-Chang Lee[1,2*], *Cheng-Hung Hu*[3*], *Yu-Chen Lin*[1,2], *Chu-Song Chen*[1,3], *Hsin-Min Wang*[3], *Yu Tsao*[2]

[1]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[3]Institute of Information Science, Academia Sinica, Taipei, Taiwan

r08922a27@csie.ntu.edu.tw, n124345679976@citi.sinica.edu.tw, f04922077@csie.ntu.edu.tw, chusong@csie.ntu.edu.tw, whm@iis.sinica.edu.tw, yu.tsao@citi.sinica.edu.tw

Abstract

For deep learning-based speech enhancement (SE) systems, the training-test acoustic mismatch can cause notable performance degradation. To address the mismatch issue, numerous noise adaptation strategies have been derived. In this paper, we propose a novel method, called noise adaptive speech enhancement with target-conditional resampling (NASTAR), which reduces mismatches with only one sample (one-shot) of noisy speech in the target environment. NASTAR uses a feedback mechanism to simulate adaptive training data via a noise extractor and a retrieval model. The noise extractor estimates the target noise from the noisy speech, called pseudo-noise. The noise retrieval model retrieves relevant noise samples from a pool of noise signals according to the noisy speech, called relevant-cohort. The pseudo-noise and the relevant-cohort set are jointly sampled and mixed with the source speech corpus to prepare simulated training data for noise adaptation. Experimental results show that NASTAR can effectively use one noisy speech sample to adapt an SE model to a target condition. Moreover, both the noise extractor and the noise retrieval model contribute to model adaptation. To our best knowledge, NASTAR is the first work to perform one-shot noise adaptation through noise extraction and retrieval.

Index Terms: speech enhancement, noise adaptation, contrastive learning, source separation, acoustic retrieval

1. Introduction

Speech enhancement (SE) is commonly used as a pre-processor in speech-related applications, such as hearing aids [1, 2], automatic speech recognition [3–5], and speech emotion recognition [6]. Recently, various deep learning (DL) models have been used to formulate a regression function for SE [7–13]. Practically, we intend to collect a broad spectrum of noise types to train a unified DL-based SE model that performs well in diverse conditions. However, in real-world scenarios, the unified SE model inevitably encounters unseen noise types. If the training data do not entirely align with the testing distribution, a training-test mismatch occurs, limiting the achievable enhancement performance.

Thus far, numerous attempts have been made to reduce the training-test mismatch issue of SE. To prevent mismatch under specific noise types, noise-aware strategies [14–17] were proposed to specify noise information into conditional embeddings to guide SE models to achieve better enhancement results. Meanwhile, ensemble learning strategies [18–20] were used in SE systems by preparing multiple component models, each of which is trained with a subset of the training data; during testing, the outputs of these component models are combined with a fusing/gating mechanism to dynamically handle noisy conditions. Despite confirmed effectiveness in many tasks, noise-aware and ensemble learning SE systems may yield sub-optimal performance when the training data are not directly sampled for the target condition. To handle the above issue, noise adaptation approaches have been proposed to handle mismatches by fine-tuning SE models to properly match target conditions. For supervised adaptation methods, the pre-trained SE system is modified with the available noisy-clean paired signals [21–24]. For unsupervised adaptation methods, it is assumed that only noisy inputs are accessible. Several studies [25–27] applied domain adversarial training (DAT) [28] techniques to convert input speech signals to noise-invariant features for processing by the decoder of the SE model. In addition, to avoid the catastrophic forgetting problem, Lee et al. [29] proposed the SERIL system based on a regularization-based incremental learning strategy. Meanwhile, multi-task learning approaches [7, 30] employ auxiliary tasks to provide positive effects for training adaptive SE models. All of these existing strategies require batches of data for adaptation. However, we usually only get a very small amount of data from the target environment.

In this paper, we propose a novel method called noise adaptive speech enhancement with target-conditional resampling (NASTAR[†]) to effectively handle the mismatch problem with the least amount of target data. Instead of using batches of extra data, our work is the first to reduce the mismatch problem by filtering out out-of-target data. Given one segment of noisy speech in the target environment, the NASTAR uses the sample to simulate the adaptation training data via a noise extractor and a retrieval model. The noise extractor estimates the target noise from the noisy speech, termed pseudo-noise. The noise retrieval model retrieves a relevant noise set, termed relevant-cohort, from an existing pool of noise signals based on the given noisy speech sample. The pseudo-noise and the relevant-cohort set are used in a combined sampling scheme and then mixed with the source speech corpus to simulate training data for noise adaptation. Compared to existing approaches, the main advantages of NASTAR include:

1) Data availability: NASTAR requires no additional target data beyond a given "one-shot" from the target environment. In noise adaptive SE research, this is the first work to re-utilize existing datasets instead of collecting additional data.

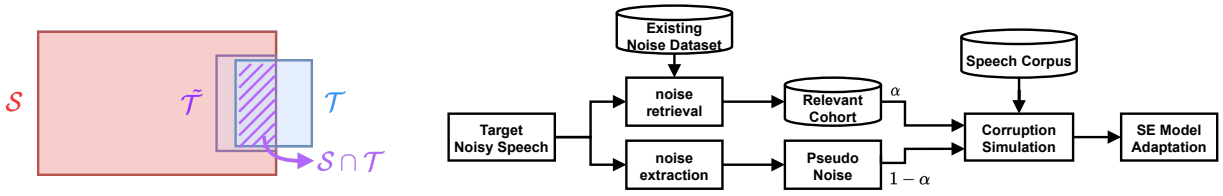
2) Promising performance: Experiment results confirm that NASTAR yields consistent and significant improvements (validated by the dependent t-test shown in Sec. 4.2) over a pre-trained model with few speech samples.

3) High training stability: NASTAR performs adaptation in a simple supervised manner, and the objective is committed to be aggressive towards handling the target noise type. We note that NASTAR has a more stable training process than the adaptive SE systems trained with multi-task objectives.

2. Background and Motivation

Based on the nature of noise interference, given a clean utterance $s(t)$ and a noise signal $n(t)$, we can simulate the noisy signal

[*]These authors contributed equally to this work.
[†]The codes are availabel at https://github.com/ChangLee0903/NASTAR



(a) (b)

Figure 1: *(a) Illustration of the subset relation between the accessible source noise data set ($\mathcal{S}$), the target-similar noise set $\tilde{\mathcal{T}}$, and the true target noise set ($\mathcal{T}$); (b) The flowchart of our NASTAR system. Given a query signal, the noise extractor directly estimates the pseudo-noise, and the retrieval model retrieves a relevant-cohort set.*

by $s(t) + n(t)$ and thus prepare noisy-clean paired data of the target condition. In general, we assume that the recorded data from the target environment contain a mixture of speech and noise components. Thus, we can use a noise extractor to estimate the noise signal, termed pseudo-noise, from the noisy speech to simulate target noisy speech for model adaptation. Nevertheless, the pseudo-noise may not be sufficient to cover the environment characteristics if the target noise is highly non-stationary, as there is only one segment. Therefore, we need to leverage more target-similar noise signals for data simulation. Although the prepared training data may cover a wide range of noise types, real-world SE systems may encounter few specific noise types not fully involved in all the training data, which inevitably leads to mismatches between training and test data, thereby limiting the SE performance.

To address the mismatch issue, this study investigates retrieving similar samples from the existing source noise dataset to enrich adaptation data. Our main idea focuses on filtering out out-of-target data and finding "what to train" for the specific target environment to reduce the mismatch. As shown in Fig. 1(a), the proposed NASTAR system intends to collect the target-similar noise set $\tilde{\mathcal{T}}$ from the source noise dataset $\mathcal{S}$ and the pseudo-noise signal to sufficiently overlap with the true target noise set $\mathcal{T}$. Since our resampling method selects noise signals from $\tilde{\mathcal{T}}$ instead of $\mathcal{S}$, this scheme is expected to reduce the mismatch between training and target noisy environments. In this way, the combination of noise extractor and noise retrieval model can provide better noise adaptation results in a parallel training manner. Meanwhile, the NASTAR system only needs one target noisy speech.

In our experimental setup, we firstly constructed a custom noise set, which included 5 different noise signals as different types. Each type of the custom noise set corresponds to one target noise signal. We cut each target noise signal in half. The first half was used to contaminate a randomly selected test utterance as the accessible noisy speech (at SNR 0dB); the second half was used to contaminate other test utterances.

3. The NASTAR System

3.1. System overview

Fig. 1b shows the overall flow of our proposed NASTAR system. First, we assume that a segment of target noisy speech (one-shot) is available and serves as the query signal. Next, the noise extractor estimates the noise from the query signal as pseudo-noise. Meanwhile, the noise retrieval model retrieves a relevant-cohort set from the source noise dataset based on the query signal. In this work, the relevant-cohort set consists of 250 noise signals closest to the query signal in terms of cosine similarity. The collection of the pseudo-noise and the relevant-cohort set forms a noise pool that potentially covers the noise characteristics of the target condition. By using a collaborative sampling mechanism, the pseudo-noise signal has the opportunity of "1 - α", and each noise signal in the relevant-cohort set has the opportunity of "α/250" to be selected to corrupt the clean speech. Finally, we adapt the pre-trained SE model to the target condition by using the sampled noise signals. α was set to 0.9.

3.2. Noise extractor

To construct a noise extractor, we used the DEMUCS model, which has been shown to yield state-of-the-art results on the music separation and SE tasks [31, 32]. Referred to [32], we follow the same settings of the architecture and objective function to estimate the noise signal as pseudo-noise from the query signal. The DEMUCS model consists of an encoder-decoder architecture with skip-connections and adopts a multi-resolution STFT objective function capturing the information of different time-frequency resolutions more effectively. The objective function is defined as:

$$\frac{1}{T}[||y - \hat{y}||_1 + \sum_{i=1}^{M} L_{stft}^{(i)}(y, \hat{y})], \tag{1}$$

where y and $\hat{y}$ are the target noise signal and the predicted noise signal, respectively; M is the number of STFT losses; L_{stft} is the addition of the spectral convergence loss $L_{sc}(y, \hat{y})$ and the magnitude loss $L_{mag}(y, \hat{y})$. $L_{sc}(y, \hat{y})$ and $L_{mag}(y, \hat{y})$ are respectively calculated by Eq. 2 and Eq. 3:

$$L_{sc}(y, \hat{y}) = \frac{|||STFT(y)| - |STFT(\hat{y})|||_F}{|||STFT(y)|||_F}, \tag{2}$$

$$L_{mag}(y, \hat{y}) = \frac{1}{T}\| \log|STFT(y)| - \log|STFT(\hat{y})|\|_1. \tag{3}$$

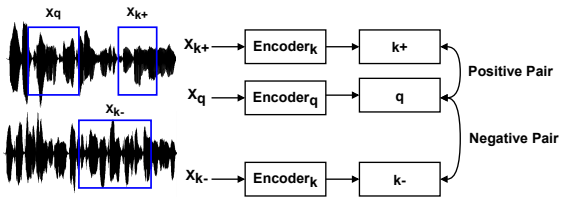


Figure 2: *Formation of positive pairs and negative pairs for training the noise retrieval model.*

3.3. Noise retrieval model

Given a target noisy speech as the query signal, a noise retrieval model aims to determine whether a compared noise signal is relevant to the query signal. The key lies in a good representation of the query and compared signals. In the representation space, the relevant signals are closer to the query than the irrelevant signals. In addition, noise signals are diverse, complex, and thus it is challenging to define a noise type as a specific class.

Some studies [33, 34] have explored self-supervised contrastive learning for speaker verification as an unsupervised learning framework. In [34], the commonly used contrastive learning

frameworks SimCLR [35], MoCo [36], and ProtoNCE [37] are compared for extracting speaker representations. To learn discriminative noise representations, we refer to these frameworks and design a pretext task to make paired data to train our noise retrieval model, as shown in Fig. 2. Our pretext task used to produce paired segments from noise signals include truncation, duplication, and shifting operations. Each sampled noise signal is randomly processed into segments, and each segment ranges from 24,000 to 80,000 sample points. Since different segments in the same noise signal have similar characteristics, these segments can be considered to be relevant to each other. As shown in Fig. 2, given a segment (the query segment x_q) excerpted from a noise signal, we can take another segment (the positive segment x_{k^+} to be compared) from the same noise signal to form a positive pair. Meanwhile, we assume that any two noise signals come from different noise conditions. Therefore, given a segment (the query segment x_q), we can randomly select a segment from another noise signal (the negative segment x_{k^-} to be compared) to form a negative pair. In addition, each segment being compared has a 50% chance of being mixed with a speech signal at a random SNR level (from -8dB to 8dB in 2dB steps). With these noisy speech segments involving both noise and speech components, our noise retrieval model can learn to filter out non-noise (speech) components. Therefore, at inference time, we can directly use "noisy speech" as the query signal to find those acoustically similar noise samples to construct the relevant-cohort set.

In training, we combine the negative pair usage mechanisms of the SimCLR [35] and MoCo [36] frameworks. In the early stage, in order to stabilize the training, we adopt the SimCLR method to generate negative pairs from the segments in a batch. After 5,000 steps, we follow MoCo to use the queue to access the past embeddings, and generate negative pairs from the past embeddings and the current embeddings. For a mini-batch, there are two embedding sets $\mathcal{B}_q$ and $\mathcal{B}_k$ generated by the query encoder network f_q and the key encoder network f_k, respectively, and a queue $\mathcal{N}$ that stores embeddings in previous steps. The loss function is defined as:

$$l_{q,k^+} = -\log \frac{\exp(\phi(q,k^+)/\tau)}{\sum\limits_{k^- \in \mathcal{B}_q \cup \mathcal{B}_k \cup \mathcal{N}} \mathbb{1}_{[k^- \neq k^+]} \exp(\phi(q,k^-)/\tau)}, \tag{4}$$

where ϕ is the similarity function, and τ denotes the temperature parameter. The query embedding q is obtained by $q = f_q(x_q)$, and x_q is the query segment. The positive key embedding k^+ is obtained by $k^+ = f_k(x_k^+)$, and x_k^+ is a positive key segment. For the negative key embedding k^- coming from $\mathcal{B}_q$ or $\mathcal{B}_k$, k^- is obtained by $k^- = f_q(x_k^-)$ or $k^- = f_k(x_k^-)$, and x_k^- is a negative key segment. The negative key embedding k^- can also be sampled from the previous embeddings $\mathcal{N}$. Given an embedding pair $(q,k) \in \mathbb{R}^d$, we use cosine similarity in the similarity function ϕ.

Our noise retrieval model consists of three bidirectional LSTM layers and one cascaded MLP projection layer. Denoting the parameters of f_k as θ_k and those of f_q as θ_q, we update θ_k by $\theta_k = \mu\theta_k + (1-\mu)\theta_q$, where μ is the momentum coefficient. To stabilize training in the early stage, those negative keys stored in $\mathcal{N}$ are not sampled until 5,000 steps.

4. Experiments

4.1. Experimental setup and implementation details

The datasets used in the experiments include: (1) the DNS-Challenge [38] noise dataset consisting of 65,303 background and foreground noise samples; (2) the Librispeech-360 [39] corpus consisting of 104,014 utterances; (3) the Voice Bank Corpus (VBC) [39] consisting of 11,572 utterances of 28 speakers; and (4) the custom noise set consisting of five noise signals of five common noise types, namely *AC/Vacuum*, *Babble*, *CafeRestaurant*, *Car*, and *MetroSubway*, selected from FreeSound [40].

The NASTAR system consists of four modules: the noise extractor, the noise retrieval model, the pre-trained SE model, and the adapted SE model. To train the noise extractor and the pre-trained SE model, the clean speech utterances from Librispeech-360 were contaminated by randomly sampled noise signals from DNS-Challenge at five SNR levels (from 0dB to 12dB in 3dB steps). We prepared noisy-noise and noisy-speech pairs to train the noise extractor and the pre-trained SE model, respectively. The models were trained by the Adam optimizer with β_1 = 0.9 and β_2 = 0.999, a learning rate of 0.0002, and a batch size of 8 with 500,000 steps. The noise extractor and the pre-trained SE model shared the same architecture and hyper-parameter settings. To train the noise retrieval model, the training data were prepared from DNS-Challenge and Librispeech-360. The process of preparing positive and negative pairs is presented in Sec. 3.3 and Fig. 2. The noise retrieval model was trained by contrasting positive and negative pairs. We used the Adam optimizer with β_1 = 0.9 and β_2 = 0.999, a learning rate of 0.00025, and a batch size of 256 to train the model for 100,000 steps. The hyper-parameters were set as follows, the temperature coefficient τ = 0.1, the momentum coefficient μ = 0.9, and the queue size $|\mathcal{N}|$ = 32,768. To train the adapted SE model, the pseudo-noise and the relevant-cohort set were derived based on the given query signal. The pre-trained SE model served as the initial checkpoint. Each clean utterance from Librispeech-360 was mixed with a target-similar noise signal sampled from the pseudo-noise and relevant-cohort at 7 SNR levels (from -4dB to 8dB in 2dB steps). The SE model was updated by the Adam optimizer with β_1 = 0.9 and β_2 = 0.999, a learning rate of 0.0001, and a batch size of 8 with 20,000 steps.

For each noise signal in the custom noise set, we used its first half and a test speech from the test-clean set of Librispeech to form the 0dB query noisy speech and accordingly trained the adaptive SE model. That is, there are five adaptation models, each for one noise type. For evaluation, we used the clean speech of the VBC corpus as our test set. For each noise condition, each clean utterance was separately mixed with the second half of the corresponding target noise signal at four SNR levels (from -5dB to 10dB in 5dB steps) as the test noisy speech. Therefore, there are 11,572×4 test noisy utterances for each noise condition. Three standardized evaluation metrics were used to measure the SE performance: narrow band perceptual evaluation of speech quality (PESQ_{nb}) [41], short-time objective intelligibility measure (STOI) [42], and scale-invariant signal-to-distortion ratio (SI-SDR) [43]. PESQ_{nb} was designed to evaluate the quality of processed speech, ranging from -0.5 to 4.5. STOI was designed to compute the speech intelligibility, ranging from 0 to 1. SI-SDR was designed to measure the energy ratio between speech and non-speech components. For all three metrics, higher scores indicate better performance.

4.2. Results

Table 1 shows the STOI, PESQ_{nb}, and SI-SDR scores of NASTAR and other related SE methods. **PTN** represents the original pre-trained SE model. $\textbf{DAT}_{\text{one}}$ represents the adaptive SE model trained with DAT using one sample of the test set. $\textbf{DAT}_{\text{full}}$ represents the adaptive SE model trained with DAT using the full test set. To investigate the effect of the noise extractor and the noise retrieval model in our NASTAR model, we prepared the paired data for adaptation in different ways: (1) only based on the estimated pseudo-noise (termed **EXTR**, α = 0); (2) only based on the ground truth of pseudo-noise (termed **GT**, α = 0); (3) based on the estimated pseudo-noise and randomly sampled noise signals from DNS-Challenge (termed **ALL**, α = 0.9); and (4) only based on the 250 noise signals in relevant-cohort (termed **RETV**, α = 1). In addition, we used the second half of the target noise signal to adapt the pre-trained SE model; that is, the adaptation and

Table 1: *Average $PESQ_{nb}$, STOI, and SI-SDR scores of different models. The red and orange numbers denote the first and second place results for each condition, respectively.*

	ACVacuum			Babble			CafeRestaurant			Car			MetroSubway		
method	STOI	$PESQ_{nb}$	SI-SDR	STOI	$PESQ_{nb}$	SI-SDR	STOI	$PESQ_{nb}$	SI-SDR	STOI	$PESQ_{nb}$	SI-SDR	STOI	$PESQ_{nb}$	SI-SDR
NOISY	0.8407	1.9671	10.8404	0.8469	2.2791	12.3311	0.8955	2.7286	10.8916	0.8706	2.4135	11.9138	0.9359	3.3332	21.2817
PTN	0.8815	2.8496	17.6623	0.8862	2.8116	16.4004	0.9199	3.3711	17.5752	0.925	3.4061	16.8882	0.9488	3.8564	21.2581
DAT_{one}	0.8823	2.8694	17.8753	0.8897	2.8017	16.9236	0.9215	3.3794	17.4182	0.9276	3.4170	17.2144	0.9504	3.8169	22.3700
DAT_{full}	0.8712	2.5442	15.5812	0.884	2.7615	16.1574	0.9094	3.2511	16.7324	0.9089	3.0483	15.9469	0.9359	3.7047	20.5600
NASTAR	0.8929	2.9482	18.2684	0.8916	2.8655	17.2072	0.9244	3.428	18.6476	0.930	3.4602	18.0286	0.9524	3.9121	22.3703
EXTR	0.8914	2.9047	18.6487	0.8771	2.6580	15.9827	0.9144	3.2114	14.9499	0.9113	3.1036	17.3962	0.9439	3.7233	21.5684
GT	0.8929	2.9062	18.7776	0.8779	2.6576	16.1166	0.9167	3.2485	16.3388	0.9106	3.0888	17.2334	0.9457	3.7811	23.3171
ALL	0.8905	2.9330	18.2512	0.8903	2.8260	16.8602	0.9225	3.4186	18.4044	0.9287	3.4225	17.4977	0.9504	3.8201	22.0051
RETV	0.8883	2.9469	18.1033	0.8913	2.8672	17.1955	0.9244	3.438	18.3673	0.9282	3.4949	18.2736	0.9524	3.8879	22.7825
OPT	0.9017	3.0561	18.9691	0.9111	3.0662	18.5652	0.9348	3.5684	19.4992	0.9364	3.5641	19.4694	0.9568	3.9346	26.5219

test noise signal are the same, and α is 0. The resulting model is called **OPT**. The oracle performance by **OPT** represents the upper-bound of NASTAR's performance. Furthermore, our test set can be divided into 20 groups based on 5 noise types and 4 SNR levels. We performed a dependent t-Test on the paired SE results based on these 20 groups. The p-values for NASTAR to outperform **PTN** in STOI, $PESQ_{nb}$, and SI-SDR scores are 3.31×10^{-8}, 5.81×10^{-11}, and 3.35×10^{-12}, respectively. Based on a standard threshold of 0.05, the improvement of NASTAR over **PTN** is confirmed to be significant.

4.2.1. *Comparison with other adaptive models*

Next, we compare the performance of NASTAR and the adaptive SE model with domain adversarial training (cf. $\textbf{DAT}_{one}$ and $\textbf{DAT}_{full}$ in Table 1). From Table 1, we observe that $\textbf{DAT}_{one}$ and $\textbf{DAT}_{full}$ are not always better than **PTN**, and our NASTAR model consistently outperforms $\textbf{DAT}_{one}$ and $\textbf{DAT}_{full}$ in all metrics under all noise conditions. The main reason for the results is that the help of the auxiliary adversarial training loss cannot effectively improve SE even with complete test data. In contrast, our NASTAR model is trained to the original SE objective by adjusting the sampling scheme of the existing training data.

4.2.2. *Ablation study*

Then, we investigate the effect of the noise extractor and the noise retrieval model in NASTAR. From Table 1, we observe that **EXTR** performs mostly worse than other variants of the NASTAR model. The results show that due to the estimation error of the noise extractor, the low-quality pseudo-noise is not sufficient to cover the environmental characteristics, and the complement of relevant-cohort is needed. Comparing NASTAR with **ALL**, we confirm that adaptive data sampled from relevant-cohort are more useful than randomly sampled data from DNS-Challenge. This is because the noise samples in relevant-cohort are closer to the target condition than the randomly sampled data from DNS-Challenge, thus reducing the mismatch between training and test conditions. **GT** performs mostly worse than NASTAR and **RETV**, from which we can see that the sample diversity is not enough to cope with the variation of target noise. In contrast, the noise samples in relevant-cohort used in NASTAR and **RETV** are close to the target condition, thus enriching the adaptive data. Furthermore, **GT** is not always better than **EXTR**. This result shows that the performance of the noise extractor is acceptable. Overall, NASTAR performs the best, followed by **RETV**. The results confirm the effectiveness of the noise extractor and noise retrieval model in NASTAR. NASTAR is slightly worse than **OPT**. There is still room for improvement.

4.2.3. *Relative improvement rate*

From Table 1, we note that the STOI improvement of the pre-trained SE model (**PTN**) over noisy speech is 0.0408 (0.8815 - 0.8407) in *AC/Vacuum*, and the improvement of NASTAR over **PTN** is 0.0114 (0.8929 - 0.8815) in *AC/Vacuum*. We also note that NASTAR has a significant improvement over **PTN** in terms of $PESQ_{nb}$ and SI-SDR scores. As for the other four noise types, we observe the same results that NASTAR consistently outperforms **PTN** in terms of $PESQ_{nb}$, STOI, and SI-SDR scores. To further evaluate the improvement brought by adaptation, we define a relative improvement rate as $\frac{s - s_{NOISY}}{s_{PTN} - s_{NOISY}}$, where s is the metric score obtained by an adaptation method, and s_{NOISY} is the metric score of the unprocessed noisy speech, and s_{PTN} is the metric score obtained by the pre-trained SE model (**PTN**). The relative improvement rate indicates how much further improvement the adaptation method can provide compared to **PTN**. Fig. 3 shows the average relative improvement rate in STOI and $PESQ_{nb}$ scores for NASTAR and other SE systems across five noise types. It is clear from the figure that NASTAR reached top performance in both STOI and $PESQ_{nb}$ compared to other SE methods, confirming the effectiveness of NASTAR.

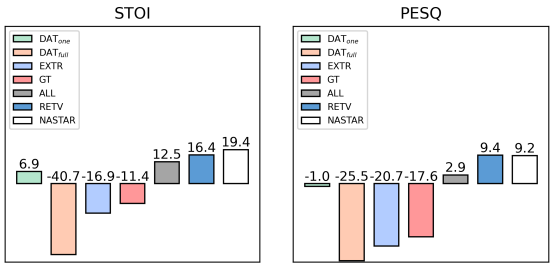


Figure 3: *The average relative improvement rate in STOI and $PESQ_{nb}$ scores for different SE methods.*

5. Concluding Remarks

Existing noise adaptation strategies for SE require batches of target noisy speech signals; however, only one-shot can be obtained in most cases. In this paper, we have proposed a novel NASTAR system to reduce the acoustic mismatch with only one target noisy utterance. NASTAR uses an accessible sample to simulate adaptation data via a noise extractor and a noise retrieval model. The noise extractor estimates the target noise, termed pseudo-noise, in the noisy speech sample. The noise retrieval model retrieves a relevant noise set, termed relevant-cohort, from a noise signal pool according to the given noisy speech sample. The pseudo-noise and relevant-cohort set are sampled and mixed with the source speech corpus to prepare simulated data for SE model adaptation. Experimental results show that NASTAR can effectively utilize a single target sample and outperforms several existing methods under different target conditions. To our best knowledge, NASTAR is the first work to apply resampling to leverage existing source data, which is more precise, efficient, and effective. Furthermore, our adaptation method can be combined with existing noise-aware and ensemble strategies in various scenarios.

6. References

[1] K. Tan, X. Zhang, and D. Wang, "Real-time speech enhancement using an efficient convolutional recurrent network for dual-microphone mobile phones in close-talk scenarios," in *Proc. of ICASSP*, 2019.

[2] I. Fedorov, M. Stamenovic, C. Jensen, L. Yang, A. Mandell, Y. Gan, M. Mattina, and P. N. Whatmough, "Tinylstms: Efficient neural speech enhancement for hearing aids," in *Proc. of Interspeech*, 2020.

[3] A. Nicolson and K. K. Paliwal, "Deep Xi as a front-end for robust automatic speech recognition," in *Proc. of CSDE*, 2020.

[4] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *Proc. of ICASSP*, 2017.

[5] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *Latent Variable Analysis and Signal Separation*, E. Vincent, A. Yeredor, Z. Koldovský, and P. Tichavský, Eds. Springer International Publishing, 2015, pp. 91–99.

[6] A. Triantafyllopoulos, G. Keren, J. Wagner, I. Steiner, and B. W. Schuller, "Towards robust speech emotion recognition using deep residual networks for speech enhancement," in *Proc. of Interspeech*, 2019.

[7] S. Wang, W. Li, S. M. Siniscalchi, and C. Lee, "A cross-task transfer learning approach to adapting deep speech enhancement models to unseen background noise using paired senone classifiers," in *Proc. of ICASSP*, 2020.

[8] Y.-C. Lin, Y.-T. Hsu, S.-W. Fu, Y. Tsao, and T.-W. Kuo, "IA-NET: Acceleration and compression of speech enhancement using integer-adder deep neural network," in *Proc. of Interspeech*, 2019.

[9] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Proc. of Interspeech*, 2013.

[10] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM TASLP*, vol. 23, no. 1, pp. 7–19, 2014.

[11] D. Liu, P. Smaragdis, and M. Kim, "Experiments on deep learning for speech denoising," in *Proc. of Interspeech*, 2014.

[12] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. of ICASSP*, 2015.

[13] A. Kumar and D. Florencio, "Speech enhancement in multiple-noise conditions using deep neural networks," in *Proc. of Interspeech*, 2016.

[14] "Noise perturbation for supervised speech separation," *Speech Communication*, vol. 78, pp. 1–10, 2016.

[15] B. O. Odelowo and D. V. Anderson, "A study of training targets for deep neural network-based speech enhancement using noise prediction," in *Proc. of ICASSP*, 2018.

[16] H. Li and J. Yamagishi, "Noise Tokens: Learning Neural Noise Templates for Environment-Aware Speech Enhancement," in *Proc. of Interspeech*, 2020.

[17] J. Lee, Y. Jung, M. Jung, and H. Kim, "Dynamic noise embedding: Noise aware training and adaptation for speech enhancement," in *Proc. of APSIPA ASC*, 2020.

[18] X.-L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.

[19] S. E. Chazan, J. Goldberger, and S. Gannot, "Deep recurrent mixture of experts for speech enhancement," in *Proc. of WASPAA*, 2017.

[20] J. Le Roux, S. Watanabe, and J. R. Hershey, "Ensemble learning for speech enhancement," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[21] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Cross-language transfer learning for deep neural network based speech enhancement," in *Proc. of ISCSLP*, 2014.

[22] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *Proc. of APSIPA ASC*, 2015.

[23] Y. Koizumi, K. Niwa, Y. Hioka, K. Kobayashi, and Y. Haneda, "Dnn-based source enhancement to increase objective sound quality assessment score," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[24] ——, "Dnn-based source enhancement self-optimized by reinforcement learning using sound quality measurements," in *Proc. of ICASSP*, 2017.

[25] C.-F. Liao, Y. Tsao, H.-Y. Lee, and H.-M. Wang, "Noise adaptive speech enhancement using domain adversarial training," in *Proc. of Interspeech*, 2019.

[26] T. Higuchi, K. Kinoshita, M. Delcroix, and T. Nakatani, "Adversarial training for data-driven speech enhancement without parallel corpus," in *Proc. of IEEE ASRU*, 2017.

[27] H.-Y. Lin, H.-H. Tseng, X. Lu, and Y. Tsao, "Unsupervised noise adaptive speech enhancement by discriminator-constrained optimal transport," in *Proc. of NeurIPS*, 2021.

[28] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, vol. 17, no. 1, pp. 2096–2030, 2016.

[29] C.-C. Lee, Y.-C. Lin, H.-T. Lin, H.-M. Wang, and Y. Tsao, "SERIL: noise adaptive speech enhancement using regularization-based incremental learning," in *Proc. of Interspeech*, 2020.

[30] Y. Bando, K. Sekiguchi, and K. Yoshii, "Adaptive neural speech enhancement with a denoising variational autoencoder." in *Proc. of Interspeech*, 2020, pp. 2437–2441.

[31] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[32] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," in *Proc. of Interspeech*, 2020.

[33] M. Ravanelli and Y. Bengio, "Learning Speaker Representations with Mutual Information," in *Proc. of Interspeech*, 2019.

[34] W. Xia, C. Zhang, C. Weng, M. Yu, and D. Yu, "Self-supervised text-independent speaker verification using prototypical momentum contrastive learning," in *Proc. of ICASSP*, 2021.

[35] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. of ICML*, 2020.

[36] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. of CVPR*, 2020.

[37] J. Li, P. Zhou, C. Xiong, and S. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. of ICLR*, 2021.

[38] C. K. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun *et al.*, "The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," in *Proc. of Interspeech*, 2020.

[39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. of ICASSP*, 2015.

[40] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. of the 21st ACM International Conference on Multimedia*, 2013.

[41] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. of ICASSP*, 2001.

[42] C. Taal, R. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. of ICASSP*, 2010.

[43] J. Le Roux, S. Wisdom, H. Erdogan, and J. Hershey, "SDR-half-baked or well done?" in *Proc. of ICASSP*, 2019.