ANALYSIS OF NOISY-TARGET TRAINING FOR DNN-BASED SPEECH ENHANCEMENT

Takuya Fujimura, Tomoki Toda
Nagoya University, Japan

ABSTRACT

Deep neural network (DNN)-based speech enhancement usually uses a clean speech as a training target. However, it is hard to collect large amounts of clean speech because the recording is very costly. In other words, the performance of current speech enhancement has been limited by the amount of training data. To relax this limitation, Noisy-target Training (NyTT) that utilizes noisy speech as a training target has been proposed. Although it has been experimentally shown that NyTT can train a DNN without clean speech, a detailed analysis has not been conducted and its behavior has not been understood well. In this paper, we conduct various analyses to deepen our understanding of NyTT. In addition, based on the property of NyTT, we propose a refined method that is comparable to the method using clean speech. Furthermore, we show that we can improve the performance by using a huge amount of noisy speech with clean speech.

Index Terms— Single-channel speech enhancement, deep neural network, Noise2Noise, unsupervised learning

1. INTRODUCTION

Speech enhancement aims to extract target speech from noisy observation. It plays an important role in a variety of applications such as automatic speech recognition (ASR) systems [1, 2, 3]. In recent years, speech enhancement has made great advancements by using a deep neural network (DNN) [4, 5, 6, 7, 8]. Most of the DNN-based speech enhancement method is based on Clean-target Training (CTT) as shown in Fig. 1 (a). In CTT, we input noisy speech into a DNN and train it to predict the corresponding clean target speech. With this approach, many methods have achieved superior performance to the traditional signal processing methods.

Although the CTT is a proper strategy, it has one major problem: collecting a huge amount of data is hard. Because audio signals are easily degraded by environmental noise and reverberation, clean speech signals should be recorded in a particular setting such as an anechoic chamber. It makes the recording very costly and time-consuming and prohibits us from collecting a large-sized dataset. This problem has limited the performance of speech enhancement since current deep learning methods perform better with more data [9, 10, 11]. Therefore, it is essential to relax the limitation on the amount of data by utilizing signals obtained easily.

To overcome the limitation, some studies consider approaches that do not require clean speech [12, 13, 14]. MixIT separates the mixture of two mixtures into each source and trains a DNN to minimize the error between the original mixture and the mixture generated by remixing separated mixtures [12]. MetricGAN-U trains a DNN to improve the speech quality metric, where the metric does not require clean speech [13]. A simpler approach than these is Noisy-target Training (NyTT) as shown in Fig. 1 (b). In the NyTT, a DNN predicts a noisy target speech from a more noisy signal synthesized by mixing the noisy target speech and an additional noise signal. In the inference, we input an unprocessed noisy speech and obtain the enhanced speech signal. In other words, NyTT trains a DNN using noisy speech instead of clean speech. Here, the model and training procedure are the same for CTT and NyTT. Therefore, NyTT is important in terms that it can directly benefit from the advanced CTT model, e.g., its performance and processing speed. Despite its importance, detailed analysis other than experimental proof that this strategy can train DNNs is insufficient.

In this paper, we conduct the analysis of NyTT and deepen our understanding of it. Our contributions are as follows: (i) proposing a new method based on the property of NyTT and showing its effectiveness, (ii) clarifying the conditions under which NyTT and its newer version work effectively focusing on the mismatch of noise environment between given data, and (iii) showing that we can improve the performance of the DNN-based speech enhancement by adding a large-sized dataset of noisy speech to a clean dataset to increase the amount of the training data.
2. NOISY-TARGET TRAINING

Noisy-target Training is the training strategy for DNN-based speech enhancement that does not require clean speech signals at all. NyTT utilizes the observation \( x = s + n_{\text{obs}} \) where \( s \) and \( n_{\text{obs}} \) denote clean speech and observed noise, respectively. In the training, we synthesize a more noisy signal \( y \) by mixing a noisy target \( x \) and an additional noise \( n_{\text{add}} \) as \( y = x + n_{\text{add}} \). Then, we train a DNN \( f(\cdot) \) to minimize the following prediction error \( L \):

\[
L = E[D(f(y; \theta), x)],
\]

where \( E \) is the expectation operator, \( \theta \) is the set of parameters of the DNN \( f(\cdot) \), and \( D \) is the mean-squared-error (MSE). A previous study has experimentally shown that NyTT can train a DNN for speech enhancement [14].

The reason why NyTT works has been believed to be because NyTT can be interpreted as Noise2Noise which is the training strategy for image denoising that does not require clean signals [15]. In the Noise2Noise, we train a DNN to learn the mapping from noisy signal \( s + n^{(1)} \) to another noisy signal \( s + n^{(2)} \). Since a DNN cannot exactly predict random noise components, their expected values are predicted as the noise components when we use MSE as a loss function. Therefore, by assuming that the noise of the target signal has zero-mean distribution, this training yields a DNN that predicts a clean target signal. For NyTT, it can be viewed as Noise2Noise with \( y = s + (n_{\text{obs}} + n_{\text{add}}) \) and \( s + n_{\text{obs}} \) where the noise signal \( n_{\text{obs}} \) of the target signal \( x \) has zero-mean distribution in the time domain. Thus, we can interpret that the DNN is trained to predict the clean speech signal \( s \), and we can say that NyTT performs speech enhancement in a Noise2Noise manner.

3. MOTIVATION AND CONTENTS OF INVESTIGATION

Although we experimentally showed that NyTT could train a DNN without clean signals [14], its detailed behavior is still unclear. Therefore, we conduct various investigations of NyTT. In this section, we summarise the investigation items.

Validity of interpretation of NyTT: So far, NyTT has been interpreted as a Noise2Noise framework. On the other hand, NyTT can also be interpreted as the removal of \( n_{\text{add}} \). Therefore, in Section 4.2, we verify whether NyTT is Noise2Noise or not. If NyTT was not Noise2Noise, the assumption of zero-mean noise and the use of MSE would be no longer necessary and it enables us to use a variety of loss functions. Also, it is expected that NyTT will be clearly affected by the mismatch between \( n_{\text{add}} \) and noise of the test set \( n_{\text{test}} \).

Improvement of NyTT by iteration: A previous experiment has shown that the performance of NyTT is increased as the SNR of the noisy target increases [14]. Based on this property, we propose an Iterative NyTT (IterNyTT) that achieves higher-quality speech enhancement using a teacher-student learning approach [16, 17, 18].

First, IterNyTT trains a DNN by NyTT with a noisy target dataset. Next, we apply speech enhancement to the noisy target dataset using the DNN trained in the first step. Then, IterNyTT trains a new DNN using the enhanced noisy target (which can be viewed as a pseudo-clean target). Repeating this process is expected to improve the SNR of noisy targets, and thus the speech enhancement performance. We verify the effectiveness of IterNyTT in Section 4.3.

Effects of mismatch between each noise: In the NyTT, there are three noise data: \( n_{\text{obs}}, n_{\text{add}}, \) and \( n_{\text{test}} \). If NyTT was the removal of \( n_{\text{add}} \), it could not work when there was a mismatch between \( n_{\text{add}} \) and \( n_{\text{test}} \). In Section 4.4, we investigate the various effects of mismatch among these noise on NyTT.

Evaluation using a large-sized noisy dataset: It is difficult to improve speech enhancement performance by increasing clean data due to its recording costs. In contrast, we can obtain noisy signals anywhere. In Section 4.5, we practically investigate whether we can improve the performance by using a large-sized dataset of noisy speech.

4. EXPERIMENTS

4.1. Experimental setups

In the experiments, we used LibriSpeech [19] as clean speech and CHiME3 [20], the training noise dataset of VoiceBank-DEMAND [21], and the train set of DCASE 2016 Challenge Task2 (DCASE) [22] as noise signals. The 10,000 and 90,000 utterances from train set of LibriSpeech were used as Libri1 and Libri2, respectively. The background noise data of CHiME3 was cut every 10 seconds and 7.83 hours of that was divided into two sets: CHiME1 and CHiME2. Also, the training noise dataset of VoiceBank-DEMAND was divided into two sets: DEMAND1 and DEMAND2. The test set was generated by mixing 1,000 utterances from the test set of LibriSpeech and 0.56 hours of CHiME3 (CHiME) at SNR randomly selected from 2.5, 7.5, 12.5, and 17.5 dB. The noisy target dataset was generated by mixing clean speeches and noise signals \( n_{\text{obs}} \) at SNR randomly selected from 0, 5, 10, and 15 dB. We refer to this dataset as “clean set name–noise set name”, i.e., Libri1-CHiME1 means a mixture set of Libri1 and CHiME1.

The DNN estimating a complex-valued T-F mask [5] consisted of CNN-BLSTM used in [23]. The input feature was log-amplitude spectrogram of the input signal. For the short-time Fourier transform parameters, frame shift, window size, and DFT size were set to 128-, 512-, and 512-samples with the Hamming window, where the sampling frequency was 16 kHz. We trained the DNN 200 epochs with mini-batch size 50 using Adam optimizer [24] with a fixed learning rate 0.0001. For a loss function, we used MSE calculated in the time domain.

For NyTT, we generated more noisy signals...
4.2. Validity of interpretation of NyTT

In this section, we conducted two analyses to confirm whether NyTT is Noise2Noise. First, we analyzed the output signal of the DNN when we input the more noisy signal. If NyTT was Noise2Noise, the output signal would be the estimate of clean speech. In contrast, if NyTT was the removal of $n^{\text{add}}$, the output signal would be the estimate of noisy target $x$. For this analysis, we generated the more noisy signals using training and testing data, respectively. More noisy signals of the train set were generated using 1,000 utterances from Libril-CHiME1 as the noisy target and CHiME2 as $n^{\text{add}}$, where they were used during the training. More noisy signals of the test set were generated using 1,000 utterances of the testing noisy data and CHiME. Table 1 shows the evaluated scores of the noisy targets $x$, more noisy signals $y$, and output signals $f(y; \theta)$ of the both train and test set. The output is closer to the noisy target than the clean target, considering its speech quality. An example of a spectrogram is shown in Fig. 2. The figure also suggests that the output of a DNN was an estimate of a noisy target rather than that of a clean target.

Second, we compared the performance when the loss function, MSE, was calculated in the time domain (Time) with the performance when it was calculated in the amplitude spectrogram domain (Spec). If NyTT was not Noise2Noise, Spec could also perform speech enhancement, even though it cannot assume zero-mean noise. In this experiment, the DNN estimated the real T-F mask to be applied to the amplitude spectrogram, and the phase of the input signal was used to convert it to the time domain. Table 2 shows both Time and Spec conditions improved the evaluated scores of the input signal. This result indicates that NyTT worked without the assumption about zero-mean noise and MSE.

As the metrics, we used scale-invariant signal-to-distortion ratio (SI-SDR) and PESQ. These are the standard metrics for evaluation of speech enhancement.

4.3. Effectiveness of IterNyTT

In this experiment, we investigated whether the performance of IterNyTT improves during five iterations. Figure 3 shows the experimental results. IterNyTT improved SI-SDR of the noisy target, and the score of IterNyTT approached that of CTT as the iteration increased. We can also see that IterNyTT already achieved a high score at the second iteration. In the following analyses, we set the iteration of IterNyTT to three.

4.4. Effects of mismatch between each noise

In this section, we investigated the effects of the mismatch of noise on NyTT. In the experiment, we used CHiME1, DEMAND1, and DCASE as $n^{\text{obs}}$ and CHiME2 and DEMAND2 as $n^{\text{add}}$. The noisy target consisted of Libril and each $n^{\text{obs}}$ data. Table 3 summarizes the SI-SDR in each combination for $n^{\text{obs}}$ and $n^{\text{add}}$ where $n^{\text{est}}$ is CHiME. We analyzed

| Table 1. Evaluation of the signals treated in NyTT |
|----------------|----------------|----------------|
| Data | Metric | Noisy target | More noisy | Output |
| Train set | SI-SDR | 8.33 | 1.37 | 6.73 |
| | PESQ | 1.34 | 1.07 | 1.28 |
| Test set | SI-SDR | 10.69 | 1.72 | 7.49 |
| | PESQ | 1.53 | 1.08 | 1.33 |

| Table 2. Comparison of a loss function. |
|----------------|----------------|----------------|
| Input | Time | Spec |
| SI-SDR | 10.69 | 15.03 | 14.78 |
| PESQ | 1.53 | 2.23 | 2.09 |

by mixing noisy target $x$ and noise $n^{\text{add}}$ at SNR randomly selected between -5 to 5 dB, where SNR was measured by considering $x$ as the target signal and $n^{\text{add}}$ as noise. For a comparison method CTT and IterNyTT on or after the second iteration, we generated the input signals by mixing clean or enhanced noisy target and noise $n^{\text{add}}$ at SNR randomly selected from 0, 5, 10, and 15 dB.
this result from three perspectives: mismatches between $n_{\text{add}}$ and $n_{\text{test}}$, $n_{\text{add}}$ and $n_{\text{obs}}$, and $n_{\text{obs}}$ and $n_{\text{test}}$. First, for each $n_{\text{obs}}$, we can see that higher scores were obtained when $n_{\text{add}}$ was CHIME2 which matches $n_{\text{test}}$. Second, the table shows that IterNyTT improved the SI-SDR by more than 1.3 dB from NyTT when there was no mismatch between $n_{\text{add}}$ and $n_{\text{obs}}$. On the other hand, there was no or slight improvement when there was a mismatch. To further analyze the effectiveness of IterNyTT in consideration of practical use, we evaluated the IterNyTT that switches $n_{\text{add}}$ according to the removal of the $n_{\text{obs}}$ and the $n_{\text{test}}$. In the case shown at the bottom of the table, we started IterNyTT using DEMAND2 for $n_{\text{add}}$ and finished it using CHIME2 for $n_{\text{add}}$ in the third iteration, where DEMAND2 matches DEMAND1 of $n_{\text{obs}}$ and CHIME2 matches CHIME of $n_{\text{test}}$. In this way, we could obtain comparable performance to CTT and IterNyTT of the situation where $n_{\text{obs}}$ and $n_{\text{add}}$ were CHIME1 and CHIME2. Finally, we focus on the mismatch between $n_{\text{obs}}$ and $n_{\text{test}}$. For each $n_{\text{add}}$, we can see that the lowest scores were obtained when $n_{\text{obs}}$ was CHIME1. We believe this result was caused by the training of the DNN to include $n_{\text{obs}}$ which matches $n_{\text{test}}$ in the output.

Summing up the above results, in the framework of NyTT, (i) the performance will be degraded similarly to CTT when there is a mismatch between $n_{\text{add}}$ and $n_{\text{test}}$, (ii) IterNyTT is comparable to CTT when there is no mismatch between $n_{\text{add}}$ and $n_{\text{obs}}$, and (iii) the performance will be degraded when there is no mismatch between $n_{\text{obs}}$ and $n_{\text{test}}$. In addition, these results are also consistent with the interpretation that NyTT predicts noisy target $x$ by removing $n_{\text{add}}$.

### 4.5. Evaluation using a large-sized noisy dataset

In this section, we conducted two evaluations using a large-sized dataset of noisy speech. First, we confirmed whether NyTT and IterNyTT using noisy Libri2 outperformed CTT using Libri1, where the amount of Libri2 is nine times that of Libri1. Table 4 summarizes the SI-SDR, where IterNyTT with Libri2-DEMAND1 switched $n_{\text{add}}$ between DEMAND2 and CHIME2 according to $n_{\text{obs}}$ and $n_{\text{test}}$. From the table, we can see that even normal NyTT outperformed CTT when there was a mismatch between $n_{\text{obs}}$ and $n_{\text{test}}$ (CHIME). IterNyTT improved the performance and approached the score of the ideal case (CTT using Libri2).

Second, we discuss a situation where not only a large-sized noisy dataset but also a clean dataset Libri1 are available. In this case, the performance can be improved by using both noisy and clean speech as targets. For the joint use of clean and noisy targets, we have two options. The first is simple: use original noisy targets with clean targets. The second is to use enhanced noisy data. If we had $n_{\text{add}}$ corresponding to $n_{\text{obs}}$ of noisy data, we could enhance it by CTT that used Libri1 and the $n_{\text{add}}$. Therefore, we also considered the option of using enhanced noisy data. Table 5 summarizes the SI-SDR and PESQ where the joint use improved performance even without speech enhancement of the noisy target. However, we can see that adding Libri2-CHIME1 was less effective because there was no mismatch between $n_{\text{obs}}$ and $n_{\text{test}}$. Furthermore, the table shows that adding the enhanced noisy target was very effective. These results suggest that we can scale up the training data and the performance of DNN-based speech enhancement by utilizing a huge amount of noisy data.

### 5. CONCLUSION

In this study, we conducted various analyses of NyTT to reveal its behavior. Our experiments showed that (i) NyTT could be interpreted as training to predict the noisy target by removing $n_{\text{add}}$, (ii) IterNyTT was comparable to CTT when $n_{\text{add}}$ corresponding to $n_{\text{obs}}$ was available, (iii) the mismatch between $n_{\text{obs}}$ and $n_{\text{test}}$ was preferable, and (iv) we could improve the performance of CTT by adding a large-sized noisy dataset. In addition, the experiment suggests that NyTT can be applied to any CTT method without the terms of zero-mean noise and MSE because NyTT was not Noise2Noise.

Acknowledgements: This work was partly supported by JST CREST Grant Number JPMJCR19A3, and JSPS KAKENHI Grant Number JP20H00102.
6. REFERENCES

[1] A. Narayanan and D. Wang, “Ideal ratio mask estimation using deep neural networks for robust speech recognition,” in Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP). IEEE, 2013, pp. 7092–7096.

[2] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, et al., “The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices,” in Proc. IEEE Autom. Speech Recognit. Underst. Workshop (ASRU). IEEE, 2015, pp. 436–443.

[3] K. Kinoshita, T. Ochiai, M. Delcroix, and T. Nakatani, “Improving noise robust automatic speech recognition with single-channel time-domain enhancement network,” in Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP). IEEE, 2020, pp. 7009–7013.

[4] 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. Int. Conf. Acoust. Speech Signal Process. (ICASSP), 2015, pp. 708–712.

[5] D. S. Williamson, Y. Wang, and D. Wang, “Complex ratio masking for monaural speech separation,” IEEE/ACM Trans. Audio Speech Lang. Process., vol. 24, no. 3, pp. 483–492, 2016.

[6] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation,” IEEE/ACM trans. audio speech lang. process. (TASLP), vol. 27, no. 8, pp. 1256–1266, 2019.

[7] K. Wang, B. He, and W.-P. Zhu, “TSTNN: Two-stage transformer based neural network for speech enhancement in the time domain,” in Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP). IEEE, 2021, pp. 7098–7102.

[8] Y. Koizumi, S. Karita, S. Wisdom, H. Erdogan, J. R. Hershey, L. Jones, and M. Bacchiani, “DF-Conformer: Integrated architecture of Conv-TasNet and Conformer using linear complexity self-attention for speech enhancement models via bootstrapped remixing,” IEEE J. Sel. Top. Signal Process. (J-STSP), 2022.

[9] L. Drude, D. Hasenklever, and R. Haeb-Umbach, “Unsupervised training of a deep clustering model for multichannel blind source separation,” in Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP). IEEE, 2019, pp. 695–699.

[10] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in Proc. IEEE/CVF conf. comput. vis. pattern recognit. (CVPR), 2020, pp. 10687–10698.

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP). IEEE, 2015, pp. 5206–5210.

[12] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third CHiME’speech separation and recognition challenge: Dataset, task and baselines,” in Proc. IEEE Autom. Speech Recognit. Underst. Workshop (ASRU). IEEE, 2015, pp. 504–511.

[13] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, “Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech,” in Proc. ISCA Speech Synth. Workshop (SSW), 2016, pp. 146–152.

[14] “IEEE DCASE 2016 Challenge,” http://www.cs.tut.fi/sgn/arg/dcase2016/, 2016.

[15] M. Kawanaka, Y. Koizumi, R. Miyazaki, and K. Yatabe, “Stable training of DNN-based speech enhancement based on perceptually-motivated black-box cost function,” in Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP), 2020, pp. 7524–7528.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.