Air Traffic Control Speech Enhancement Method Based on Improved DNN-IRM

Yuezhou Wu *, Pengfei Li * and Siling Zhang

School of Computer Science, Civil Aviation Flight University of China, Guanghan 618307, China; zsl20000306@163.com
* Correspondence: wuyuezhou@cafuc.edu.cn (Y.W.); cafucpfd@163.com (P.L.)

Abstract: The quality of air traffic control speech is crucial. However, internal and external noise can impact air traffic control speech quality. Clear speech instructions and feedback help optimize flight processes and responses to emergencies. The traditional speech enhancement method based on a deep neural network and ideal ratio mask (DNN-IRM) is prone to distortion of the target speech in a strong noise environment. This paper introduces an air traffic control speech enhancement method based on an improved DNN-IRM. It employs LeakyReLU as an activation function to alleviate the gradient vanishing problem, improves the DNN network structure to enhance the IRM estimation capability, and adjusts the IRM weights to reduce noise interference in the target speech. The experimental results show that, compared with other methods, this method improves the perceptual evaluation of speech quality (PESQ), short-term objective intelligibility (STOI), scale-invariant signal-to-noise ratio (SI-SNR), and speech spectrogram clarity. In addition, we use this method to enhance real air traffic control speech, and the speech quality is also improved.

Keywords: speech enhancement; deep neural network; ideal ratio mask; air traffic control speech

1. Introduction

The airspace operation environment is becoming more and more complex, air traffic flow continues to increase, and the problems arising from various aspects, such as air transportation efficiency and flight safety, are becoming more and more serious [1], demanding higher standards for air traffic control. Air traffic control primarily relies on radio communication between air traffic control officers and pilots, where controllers transmit instructions, pilots read them back, and the controllers confirm them before the pilots execute the instructions. As the initial step in executing air traffic control directives, clear and accurate voice communication between controllers and pilots is a crucial factor in ensuring flight safety [2]. According to statistical data released by the European Organization for the Safety of Air Navigation and the European Telecommunications Standards Institute for Aviation, as many as 30% of aviation incidents are related to misheard voice instructions [3].

During airplane flights and air traffic control procedures, communication between controllers and pilots is often subject to various disturbances due to restricted communication conditions. These disturbances originate partly from external environmental sources such as cabin instrument noise, aircraft engine roar, and air traffic control center noise, as well as noise from the communication process itself, including radio noise and high-frequency communication noise. Due to these disturbances, conversations between the parties frequently suffer from suboptimal quality, resulting in unsafe occurrences such as unclear semantic expression, misinterpretations, and missed information. Therefore, researching methods for speech enhancement in air traffic control communication holds significant practical importance. This research ensures that controllers can convey instructions, aids pilots in better understanding these instructions, enhances flight safety, reduces communication issues and risks, and promotes more effective aviation traffic management.
Traditional speech enhancement methods can be broadly categorized into those based on time–frequency (T-F) domain filtering and those based on statistical models. T-F domain-filtering methods include spectral subtraction [4], subspace methods [5], Wiener filtering [6], and minimum mean square error estimation [7]. These techniques operate under the assumption that “noise is stationary,” aiming to reduce noise by reducing its spectrum. Methods based on statistical models encompass Hidden Markov models [8] and Gaussian mixture models [9]. These statistical models rely on specific acoustic assumptions and model structures. However, noise often exhibits variability in natural environments, rendering the acoustic assumptions about noise invalid. Consequently, the denoising effectiveness of traditional speech enhancement methods is limited, and these methods tend to generate artifacts resembling musical interference.

With the rise of deep learning, various neural network models have been introduced and applied to the field of speech enhancement. Xu et al. [10,11] proposed a supervised method based on a DNN, aiming to directly map enhanced speech by seeking the mapping function between the noise and clean speech signals. Wang et al. [12,13] employed a DNN to estimate two different targets, namely, the ideal binary mask (IBM) and the IRM, aiming to reduce noise interference in the target speech signal. Zhou et al. [14] integrated the estimation of the IRM with the target binary mask (TBM) to obtain an improved mask for speech enhancement. The performance of the fused mask was found to be superior to the masks estimated individually. Liu et al. [15] proposed an amplitude fusion method based on deep learning, which effectively improves the effect of speech dereverberation by combining the mapping method with the ideal amplitude masking (IAM) method. The IBM is simple to implement but may cause speech distortion; the IAM offers smooth transitions and improved sound quality but has higher computational complexity and may retain some noise; and the IRM provides the best results in smoothness and noise suppression of the three masks but is computationally complex and requires accurate estimation of speech and noise energy.

This research aims to solve the quality problem of air traffic control speech in complex noisy environments. Specifically, the following research questions and hypotheses are proposed. First, we explore how to effectively enhance the clarity and intelligibility of air traffic control speech in intense noise environments. Second, it is assumed that an improved DNN-IRM method can reduce the distortion of target speech in a noisy environment and outperform the traditional method. In addition, it is assumed that introducing the LeakyReLU activation function can alleviate the gradient vanishing problem and improve the accuracy of the IRM estimation, and thus improve the score of the relevant speech quality. We hope to verify these hypotheses through experiments and prove the effectiveness of the proposed method in actual air traffic control speech enhancement.

This paper’s main contributions are as follows:

- We propose an air traffic control speech enhancement method based on an improved DNN-IRM. It uses the LeakyReLU activation function to mitigate the gradient vanishing issue, refines the DNN architecture to boost the IRM estimation accuracy, and modifies the IRM weights to minimize noise interference with the target speech.
- We select some pure noise segments from the air traffic control speech database as noise, add them to other clean speech, and create speech pairs of clean speech and noisy speech. Additionally, we conduct three sets of comparative experiments using the PESQ [16], short-time objective intelligibility (STOI) [17], the structural similarity index measure (SSIM) [18] and the scale-invariant signal-to-distortion ratio (SI-SDR) [19] as evaluation metrics. The SSIM is utilized to measure the similarity between the original speech spectrogram and the enhanced speech spectrogram.
- We use air traffic control speech in a natural environment to test the performance of the method proposed in this article and also use the speech quality non-reference evaluation method to test the overall score of the enhanced speech, respectively, deep noise suppression mean opinion score (DNSMOS) [20] and non-intrusive speech quality assessment (NISQA) [21].
• We investigate the impact of different parameter values on the experimental results when adjusting the IRM weights.

The rest of the article is organized as follows. In Section 2, we begin by explaining the relevant concepts covered in the paper, followed by a detailed exposition of the adopted method’s process and network structure. Section 3 compares the proposed method with other speech enhancement methods and explores the impact of adjusting the mask weight on the experimental results. Section 4 provides a comprehensive summary of the entire document and outlines prospects for future research directions.

2. Concepts and Method

In this section, we first explain some concepts related to the method, then introduce the specific process of the proposed method, and finally analyze the network structure of the proposed method.

2.1. Relevant Concepts

This section first introduces the activation function needed in this method, then introduces the mask used in the method, and finally describes in detail how to adjust the mask.

2.1.1. Activation Function

Activation function is crucial for the learning process in neural networks, as neural networks perform linear operations, but human perception of speech signals is nonlinear. Additionally, sensitivity to low-frequency signals is higher compared to high-frequency signals. By incorporating an activation function, neural networks acquire the ability to perform nonlinear operations, enabling the model to capture nonlinear relationships within the data, thus enhancing the network’s learning capability.

The activation functions needed for this experiment are ReLU [22], LeakyReLU [23], and Sigmoid [24], whose respective equations are:

\[ \text{ReLU}(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \]  
(1)

\[ \text{LeakyReLU}(x) = \begin{cases} x, & x > 0 \\ \alpha \cdot x, & x \leq 0 \end{cases} \]  
(2)

\[ \text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \]  
(3)

where \( x \) represents input features, with the value of \( \alpha \) being 0.01. Traditional DNN-IRM-based speech enhancement methods usually use ReLU as the activation function, which may ignore the impact of negative values. The introduction of LeakyReLU can solve this problem more effectively because it allows partial information transfer even when the input of the network layer is negative, which not only helps to alleviate the gradient disappearance problem during the training process but also improves learning efficiency. As the IRM values range from 0 to 1, the utilization of the Sigmoid function is required for data correction in the final network output.

2.1.2. Ideal Ratio Mask

We assume the equation for the noisy speech signal is:

\[ y = n + s \]  
(4)

where \( s \) represents the clean speech, \( n \) stands for the noise signal, and \( y \) denotes the noisy speech.
The key idea of the IRM is to determine the relative contribution of the clean speech and noisy speech energies within each T-F unit based on their ratio. When the IRM approaches 1, it signifies a greater presence of the speech signal in that frequency component, while an IRM close to 0 indicates a higher presence of the noise signal in that frequency component. This is expressed in the following equation:

$$IRM(t, f) = \left( \frac{S(t, f)}{S(t, f) + N(t, f)} \right)^\beta$$  \hspace{1cm} (5)

where $S(t, f)$ and $N(t, f)$, respectively, represent the clean speech energy and noisy speech energy of the T-F unit. $\beta$ is an adjustable parameter used to scale the mask, and when its value is set to 0.5, it provides a more accurate signal estimation.

2.1.3. Adjusting IRM Weights

The purpose of this operation is to further reduce the interference of noise in the noisy speech, as depicted in Figure 1. For the network-estimated IRM, it aims to preserve the T-F units primarily containing target speech energy while attenuating those that do not predominantly contain target speech energy. This yields the adjusted IRM. The process can be expressed as:

$$\tilde{IRM}(t, f) = \begin{cases} 
   \tilde{IRM}(t, f), & \tilde{IRM}(t, f) > \delta \\
   \gamma \cdot \tilde{IRM}(t, f), & \tilde{IRM}(t, f) \leq \delta 
\end{cases}$$  \hspace{1cm} (6)

where $\tilde{IRM}(t, f)$ represents the network-estimated IRM and $\tilde{IRM}(t, f)$ represents the adjusted IRM. The parameter ($\delta$) ranges from 0 to 1, used to differentiate between noisy speech and clean speech. Specifically, the signals where $\tilde{IRM}(t, f) > \delta$ are considered as clean speech, while the signals where $\tilde{IRM}(t, f) \leq \delta$ are regarded as noisy speech. Another parameter ($\gamma$), also ranging from 0 to 1, aims to mitigate the impact of noisy speech on clean speech. $\gamma$ and $\delta$ selected in this article are both 0.5.

![Schematic diagram of adjusting IRM weights.](image)

Figure 1. Schematic diagram of adjusting IRM weights.

When $\tilde{IRM}(t, f) > \delta$, that is, the weights represented by the red squares in Figure 1, there is no need to adjust the weights; when $\tilde{IRM}(t, f) \leq \delta$, that is, represented by the orange and yellow squares at the top of Figure 1, it is necessary to multiply the original weights by $\gamma$ to get the corresponding squares at the bottom of Figure 1. The upper orange squares correspond to the lower yellow squares, and the upper yellow squares correspond to the lower green squares. In Figure 1, the value of $\gamma$ is 0.5, and we can observe that the weight is reduced by half.
2.2. Method Process

The process of the air traffic control speech enhancement method based on improved DNN-IRM is illustrated in Figure 2.

Figure 2. Proposed method diagram.

The processing of the method includes a training stage and an enhancement stage. The input of the training stage is noisy speech and clean speech, and its purpose is to train the DNN and narrow the gap between the noisy speech and clean speech. In the enhancement stage, the noisy speech is directly input into the trained DNN, and the enhanced speech is finally obtained through DNN decoding and related processing. The specific details of the training stage and speech enhancement stage will be introduced in Sections 2.2.1 and 2.2.2.

2.2.1. Training Stage

During the training stage, we begin by extracting features from the speech signal to obtain log power spectrum (LPS) features, which are then fed into the DNN for training. The network produces the estimated IRM values, which are subsequently compared with actual IRM values. Throughout this process, there is continuous adjustment of the network parameters to achieve outputs closer to the actual mask values.

Due to the nonlinear nature of human auditory perception of audio, directly processing the amplitude spectrum after the short-time Fourier transform (STFT) is inappropriate. Therefore, it is necessary to use LPS features as input features for training. The specific principles behind extracting the LPS feature parameters and calculating the IRM are illustrated in Figure 3.

Figure 3. Schematic diagram of feature extraction principle and IRM calculation.
The specific processing steps are as follows:

(a) Taking \( y \) as the noisy speech, \( y \) undergoes framing and windowing operations to obtain the short-time signal for each frame, denoted as \( y(n) \). The purpose of framing is to use shorter single frames as steady-state signals so that parameters between frames can transition more smoothly; the purpose of windowing is to reduce leakage in the frequency domain.

(b) STFT transforms the time-domain data into frequency-domain data, resulting in the amplitude spectrum and phase spectrum for each frame. The equations are as follows:

\[
Y(t, f), \theta(t, f) = \sum_{n=-\infty}^{\infty} y(n) \omega(n - t) e^{-j \frac{2\pi}{N} f n}
\tag{7}
\]

where \( Y(t, f) \) represents the amplitude spectrum, \( \theta(t, f) \) represents the phase spectrum, \( \omega(n) \) is a Hamming Window with length \( N \), \( t \) is a time-sensitive index, \( f \) is a frequency index, \( N \) is the length of each band, and \( e^{-j \frac{2\pi}{N} f n} \) is the kernel that can be transformed.

(c) The energy spectrum \( (E(t, f)) \) for each frame after STFT transformation is calculated as follows:

\[
E(t, f) = |Y(t, f)|^2
\tag{8}
\]

(d) The LPS features corresponding to the energy spectrum \( (LPS(t, f)) \) are calculated as follows:

\[
LPS(t, f) = \log[E(t, f)]
\tag{9}
\]

(e) Steps (a) to (c) are repeated for the clean speech \( (s) \) and the noisy speech \( (n) \) to obtain the energy of the noisy speech \( (N(t, f)) \) and the energy of the clean speech \( (S(t, f)) \). Then, the IRM values are calculated using Equation (5).

Using the Mean Squared Error (MSE) as the loss function during the training process, the model continuously updates and optimizes its parameters to minimize the disparity between the actual IRM and the estimated IRM. The definition of MSE is as follows:

\[
E(t, f) = \sum_{t,f} (IRM(t, f) - \hat{IRM}(t, f))^2
\tag{10}
\]

where \( IRM(t, f) \) represents the actual IRM and \( \hat{IRM}(t, f) \) represents the IRM estimated by the DNN.

2.2.2. Enhancement Stage

During the enhancement stage, the LPS features of the noisy speech are fed into a trained DNN, which outputs \( \hat{IRM}(t, f) \). Subsequently, the weights of \( \hat{IRM}(t, f) \) are adjusted to obtain \( \tilde{IRM}(t, f) \). After multiplying \( IRM(t, f) \) with the amplitude spectrum \( Y(t, f) \) of the noisy speech, and then applying the natural exponential operation \( (Exp) \), the enhanced amplitude spectrum \( (\tilde{Y}(t, f)) \) is obtained. \( \tilde{Y}(t, f) \) is then multiplied with the phase spectrum \( (\theta(t, f)) \) of the speech under processing. After performing the inverse short-time Fourier transform (ISTFT), the enhanced speech \( (x) \) is obtained.

\[
\tilde{Y}(t, f) = \text{Exp} \left[ Y(t, f) \circ \tilde{IRM}(t, f) \right]
\]

\[
X(t, f) = \tilde{Y}(t, f) \circ \theta(t, f)
\]

\[
x = \frac{\sum_{t,f} X(t, f) \omega(n - t) e^{-j \frac{2\pi}{N} f n}}{\sum_{t} \omega^2(n - t)}
\]

where \( \circ \) represents the Hadamard product, \( X(t, f) \) is the frequency after amplification, \( \omega(n - t) \) is the transformation of the window function, and \( \sum_{t} \omega^2(n - t) \) is a multiplication
function to ensure the width consistency when the signal is repeated. The process of speech reconstruction is illustrated in Figure 4.

This section has introduced the specific processing flow of the proposed method, and also explained in detail the specific data flow in the speech feature extraction stage and the speech reconstruction stage. The following sections will introduce the internal structure of the DNN in detail.

2.3. Network Architecture

This section outlines the network architecture of the DNN in the method process, as illustrated in Figure 5. In the diagram, "linear" represents the alteration in the number of input and output units. Each layer’s network parameters are detailed in Table 1. The model comprises one input layer, three hidden layers, and one output layer, with 1799 units in the input layer, 2048 units in each hidden layer, and 257 units in the output layer. LeakyReLU serves as the activation function for every hidden and output layer. The negative slope of the LeakyReLU function is set to 0.1. As the desired estimated IRM values need to be constrained within the range of 0 to 1, a Sigmoid function is appended to the final layer of the network to regulate the output data range. The model utilizes MSE as its loss function. It employs the Adaptive Moment Estimation algorithm for network optimization and integrates Dropout to enhance the model’s noise generalization ability. The Dropout rate for the input layer and each hidden layer is set to 0.1, while the output layer’s Dropout rate is 0. The network utilizes an adaptive learning rate, initially set at 0.01, gradually decreasing to 0.001 as the number of training iterations increases. Additionally, Batch Normalization (BN) is applied to all layers. The network underwent 100 training iterations and underwent a series of steps continuously fine-tuning the network parameters, ultimately resulting in the DNN model used for speech enhancement.

![Figure 4. Speech reconstruction process diagram.](image)

![Figure 5. The network architecture for speech enhancement.](image)
Table 1. Network parameters for each layer.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Units</th>
<th>Activation</th>
<th>Dropout</th>
<th>BN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>1799</td>
<td>LeakyReLU</td>
<td>0.1</td>
<td>Yes</td>
</tr>
<tr>
<td>Hidden1</td>
<td>2048</td>
<td>LeakyReLU</td>
<td>0.1</td>
<td>Yes</td>
</tr>
<tr>
<td>Hidden2</td>
<td>2048</td>
<td>LeakyReLU</td>
<td>0.1</td>
<td>Yes</td>
</tr>
<tr>
<td>Hidden3</td>
<td>2048</td>
<td>LeakyReLU</td>
<td>0.1</td>
<td>Yes</td>
</tr>
<tr>
<td>Output</td>
<td>257</td>
<td>LeakyReLU, Sigmoid</td>
<td>0</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Additionally, to leverage the inter-frame correlation among input features and enhance the model’s noise suppression capability, frame expansion operations are employed in the input features. Considering the $i$-th frame as the central frame, the three preceding and succeeding frames of the noisy speech’s LPS are included as model inputs. Therefore, the input sequence $Y$ satisfies the following equation:

$$Y = [Y(i-3), Y(i-2), Y(i-1), Y(i), Y(i+1), Y(i+2), Y(i+3)]$$ (12)

where $Y(i)$ represents the LPS of the $i$-th frame of the noisy speech. $Y'(i)$ stands for the IRM estimated by the network.

3. Experiments and Results Analysis

This section first introduces the experimental setup, then gives a detailed explanation of the air traffic control speech dataset, and finally shows the experimental results of the comparative experiments and each group of experiments.

3.1. Experimental Setup

The experiment is conducted using the PyTorch 1.8.0 framework. Table 2 provides the specific details of the experimental setup.

Table 2. Experimental setup.

<table>
<thead>
<tr>
<th>Experimental Setup</th>
<th>Specific Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>13th Gen Intel(R) Core(TM) i5-13490F (Intel, Santa Clara, CA, USA)</td>
</tr>
<tr>
<td></td>
<td>NVIDIA RTX4060Ti (Nvidia, Santa Clara, CA, USA)</td>
</tr>
<tr>
<td>GPU</td>
<td>Nvidia, Santa Clara, CA, USA</td>
</tr>
<tr>
<td>OS</td>
<td>Windows 11 64-bit (Redmond, WA, USA)</td>
</tr>
</tbody>
</table>

3.2. Dataset

Both the training and test speech data come from the air traffic control speech database provided by the Civil Aviation Administration of China, totaling 50,000 pieces of speech data. The characteristics of air traffic control speech are a fast speaking speed and bilingual broadcasting in Chinese and English. In addition, due to the limitation of voice calls, it is impossible to collect completely clean speech data. Therefore, we assessed the NISQA score for all the speech data, and regarded the speech signals with a score greater than 3.0 as “clean speech data”, and obtained 20,000 speech data that could be used for the experiments. Each piece of speech data is about 3 s long and the total length is about 17 h. Overall, 70% were randomly selected as training data, and the remaining 30% were used as test data; some pure noise segments with a score less than 1.5 were selected from the speech data and they were extended to the same length as the clean speech signal. The noise includes more than 130 types of noise, such as aircraft cockpit noise, high-frequency communication noise, and air traffic control (ATC) center noise. After the dataset was collected, all the acquired data were uniformly processed into mono, single-channel waveform files with a sampling rate of 16 kHz. Subsequently, noise was added to the clean speech at SNRs of $-5$ dB, 0 dB, 5 dB, and 10 dB to generate noisy training and test speech.
3.3. Contrast Experiment

To verify the performance of the method proposed in this paper, four sets of experiments were set up to provide results for discussion, as follows:

(a) R-UnAdj: This method adopts the traditional speech enhancement method based on DNN-IRM, uses ReLU as the activation function, and does not adjust the IRM weights;
(b) R-Adj: This method is based on R-UnAdj, uses ReLU as the activation function, but adjusts the IRM weights;
(c) LR-UnAdj: This method is based on R-UnAdj, uses LeakyReLU as the activation function, and does not adjust the IRM weights;
(d) LR-Adj: This method is based on R-UnAdj, uses LeakyReLU as the activation function, and adjusts the IRM weights. LR-Adj is considered as the proposed method in this paper.

In addition, we trained the networks using ReLU and LeakyReLU for 100 epochs each, and recorded the loss for each epoch of both networks, as shown in Figure 6. We observed that the network trained with LeakyReLU was more stable and had lower losses.

![Figure 6. Loss of two networks after 100 epochs of training.](image)

3.4. Results Analysis

To evaluate the actual performance of the air traffic control speech enhancement method based on the improved DNN-IRM proposed in this article, the experimental data results were compared and analyzed from two aspects: reference evaluation and non-reference evaluation. We used the SSIM, PESQ, STOI, and SI-SDR as the parametric evaluation methods. The SSIM was used to judge the similarity between the original speech spectrogram and the enhanced speech spectrogram; for the non-parametric evaluation method, we chose DNSMOS and NISQA, and chose non-parametric evaluation. The purpose of the method was to test the performance of the proposed method in a natural air traffic control speech environment. Finally, we also discuss the impact on the experimental results when the parameter $\delta$ and the parameter $\gamma$ take different values when adjusting the IRM weights. In the process of conducting the comparative experiments, when analyzing one parameter, we fixed the other parameter.

3.4.1. Spectrogram Comparison and SSIM

The spectrogram plays a crucial role in speech analysis, providing a visual understanding of speech data. Figure 7 shows the spectrogram of the clean speech used in the experiment, while Figures 8–10 display the spectrograms of the same speech data under different SNR conditions, affected by aircraft cockpit noise, ATC center noise, and high-frequency communication (HFcommunication) noise, respectively. Additionally, the enhanced spectrograms after applying the four aforementioned methods are presented. Each column in the figures corresponds to a set of instances. Taking Figure 8 as an example, Figure 8a to Figure 8e, respectively, show the spectrogram with $-5 \text{ dB}$ aircraft cockpit
noise, the spectrogram enhanced by R-UnAdj, the spectrogram enhanced by R-Adj, the spectrogram enhanced by LR-UnAdj, and the spectrogram enhanced by LR-Adj. From Figure 8, it can be observed that aircraft cockpit noise is mainly concentrated in specific frequency bands. At −5 dB noise levels, LR-Adj effectively eliminates high-frequency noise, but some noise remains in the low-frequency range, especially in pure noisy speech segments. ATC center noise gradually weakens from low to high frequencies, and LR-Adj and LR-UnAdj show the best enhancement at 10 dB, while preserving more speech details. Aircraft cockpit noise is concentrated in specific frequency bands, and all methods exhibit the worst enhancement at −5 dB noise levels. LR-Adj significantly reduces noise in pure noisy speech segments at 0 dB, 5 dB, and 10 dB. In summary, through the observation of the three figures, it can be seen that the LR-Adj method generally outperforms the other three methods, especially in low SNR conditions, where it effectively eliminates low-frequency noise.

**Figure 7.** Spectrogram of clean speech.

**Figure 8.** Comparison of aircraft cockpit noise spectrum and speech spectrum enhanced by 4 methods at different SNR levels.
Figure 9. Comparison of ATC center noise spectrum and speech spectrum enhanced by 4 methods at different SNR levels.

Figure 10. Comparison of HF communication noise spectrum and speech spectrum enhanced by 4 methods at different SNR levels.

The SSIM measures the similarity between the clean speech spectrograms and enhanced speech spectrograms. It considers three aspects of information: brightness,
The SSIM measures the similarity between the clean speech spectrograms and enhanced speech spectrograms. It considers three aspects of information: brightness, contrast, and structure. Figure 11 shows the SSIM score of the speech spectrogram after four enhancement methods at different SNR levels.

The SSIM score is obtained by comparing the enhanced speech spectrum with the clean speech spectrum. The ordinate of each chromatogram in the figure represents the SSIM score of the four methods, and the abscissa represents the added noise level. It can be observed that the speech spectrum enhanced by the LR-Adj achieved the highest SSIM scores across all noise levels. From Figure 11b, it can be observed that the LR-Adj has a significant effect on processing low SNR ATC center noise. From Figure 11a,c, it can be seen that although the LR-Adj does not have a significant effect on processing low SNR aircraft cockpit noise and HF communication noise, the ability of the proposed method to process these two noises is improved at high SNR.

3.4.2. PESQ and STOI

The main purpose of the PESQ is to simulate listeners’ perception of speech quality and provide a numerical score to represent the quality of the speech. The PESQ generates a quality score between $-0.5$ and $4.5$, where a higher score indicates better speech quality and a lower score indicates poorer quality. We selected airplane takeoff and landing noise, aircraft cabin noise, and electric current noise to test the performance of each method. The PESQ scoring results of each method at different noise types and SNR levels are shown in Table 3.
Table 3. PESQ scores of each method for different noise types and at different SNR levels.

<table>
<thead>
<tr>
<th>Noise Types</th>
<th>SNR</th>
<th>Noise</th>
<th>R-UnAdj</th>
<th>R-Adj</th>
<th>LR-UnAdj</th>
<th>LR-Adj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>10 dB</td>
<td>1.36</td>
<td>2.27</td>
<td>2.35</td>
<td>2.48</td>
<td>2.72</td>
</tr>
<tr>
<td></td>
<td>5 dB</td>
<td>1.14</td>
<td>1.77</td>
<td>1.98</td>
<td>2.07</td>
<td>2.29</td>
</tr>
<tr>
<td></td>
<td>0 dB</td>
<td>1.06</td>
<td>1.44</td>
<td>1.54</td>
<td>1.71</td>
<td>1.82</td>
</tr>
<tr>
<td></td>
<td>−5 dB</td>
<td>1.03</td>
<td>1.18</td>
<td>1.27</td>
<td>1.31</td>
<td>1.44</td>
</tr>
<tr>
<td>Current</td>
<td>10 dB</td>
<td>1.28</td>
<td>2.18</td>
<td>2.37</td>
<td>2.57</td>
<td>2.77</td>
</tr>
<tr>
<td></td>
<td>5 dB</td>
<td>1.15</td>
<td>1.72</td>
<td>1.95</td>
<td>2.21</td>
<td>2.27</td>
</tr>
<tr>
<td></td>
<td>0 dB</td>
<td>1.08</td>
<td>1.36</td>
<td>1.49</td>
<td>1.72</td>
<td>1.80</td>
</tr>
<tr>
<td></td>
<td>−5 dB</td>
<td>1.04</td>
<td>1.21</td>
<td>1.28</td>
<td>1.36</td>
<td>1.44</td>
</tr>
<tr>
<td>Cabin</td>
<td>10 dB</td>
<td>1.47</td>
<td>2.34</td>
<td>2.59</td>
<td>2.68</td>
<td>2.81</td>
</tr>
<tr>
<td></td>
<td>5 dB</td>
<td>1.27</td>
<td>1.98</td>
<td>2.20</td>
<td>2.27</td>
<td>2.39</td>
</tr>
<tr>
<td></td>
<td>0 dB</td>
<td>1.07</td>
<td>1.55</td>
<td>1.74</td>
<td>1.86</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>−5 dB</td>
<td>1.05</td>
<td>1.27</td>
<td>1.42</td>
<td>1.57</td>
<td>1.68</td>
</tr>
</tbody>
</table>

The black bold number indicates the best result. Airplane represents aircraft takeoff and landing noise, Current represents electronic current noise, and Cabin represents aircraft cabin noise.

According to the data in Table 3, the enhancement effects of each method vary under different types of noise and SNR levels. We used the score for the enhanced speech minus the score of the noisy speech, and then divided it by the score of the noisy speech to obtain the improvement offered by the method. For each noise type, the improvement effect under all the SNR values was averaged to get the average improvement effect of the method under the noise type. Under aircraft cabin noise, the average enhancement effects for R-UnAdj, R-Adj, LR-UnAdj, and LR-Adj were 43.08%, 53.26%, 63.11%, and 78.35%, respectively. Under electric current noise, the average enhancement effects for each method were 40.04%, 53.50%, 70.08%, and 79.48%, respectively. Under airplane takeoff and landing noise, the average enhancement effects for each method were 42.92%, 53.38%, 65.17%, and 78.60%, respectively. Overall, the LR-Adj method generally exhibited better performance, followed by LR-UnAdj, then R-Adj, and finally the R-UnAdj method. These results demonstrate that the improved DNN-IRM method significantly improves the PESQ scores across different noise types compared to the original methods, highlighting its superiority in enhancing speech quality. Furthermore, we observed that using LeakyReLU rather than ReLU could significantly improve the performance of the model with adjusted IRM weights because the loss in the network trained by LeakyReLU is smaller than that in the network trained by ReLU. Moreover, under high SNR, the mask value estimation is relatively accurate, and we can improve the speech quality by reducing the weight of the noise mask.

STOI is an objective assessment method that measures speech clarity and intelligibility. Scores range from 0 to 1, where 1 represents complete intelligibility and 0 represents complete unintelligibility. Higher STOI scores indicate clearer and more understandable speech, while lower scores indicate poorer intelligibility. The evaluation results for STOI for each method at different noise types and various SNR levels are shown in Table 4.

According to the data in Table 4, we can see that LR-Adj and LR-UnAdj have similar performance in terms of STOI, among which LR-Adj shows the seven best indicators. The advantages of LR-Adj are mainly reflected in high SNR levels. LR-Adj is generally higher than the three other methods for STOI scores at 5 dB and 10 dB. In contrast, LR-UnAdj performs well on six indicators and shows better enhancement effects at −5 dB and 0 dB. The reasons for the excellent effect of the LR-Adj method are the same as before. On the one hand, the network loss trained by LeakyReLU is low, and, on the other hand, reducing the weight of the noise mask can improve speech quality.
Table 4. STOI scores for each method at different noise types and different SNR levels.

<table>
<thead>
<tr>
<th>Noise Type</th>
<th>SNR</th>
<th>Noise R-UnAdj</th>
<th>R-Adj</th>
<th>LR-UnAdj</th>
<th>LR-Adj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>10 dB</td>
<td>0.85</td>
<td>0.91</td>
<td>0.92</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>5 dB</td>
<td>0.78</td>
<td>0.85</td>
<td>0.86</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>0 dB</td>
<td>0.64</td>
<td>0.79</td>
<td>0.78</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>−5 dB</td>
<td>0.51</td>
<td>0.68</td>
<td>0.65</td>
<td>0.74</td>
</tr>
<tr>
<td>Current</td>
<td>10 dB</td>
<td>0.83</td>
<td>0.91</td>
<td>0.92</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>5 dB</td>
<td>0.75</td>
<td>0.85</td>
<td>0.86</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>0 dB</td>
<td>0.65</td>
<td>0.78</td>
<td>0.78</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>−5 dB</td>
<td>0.53</td>
<td>0.67</td>
<td>0.68</td>
<td>0.73</td>
</tr>
<tr>
<td>Cabin</td>
<td>10 dB</td>
<td>0.87</td>
<td>0.90</td>
<td>0.91</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>5 dB</td>
<td>0.81</td>
<td>0.87</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>0 dB</td>
<td>0.71</td>
<td>0.81</td>
<td>0.83</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>−5 dB</td>
<td>0.63</td>
<td>0.71</td>
<td>0.75</td>
<td>0.79</td>
</tr>
</tbody>
</table>

The black bold number indicates the best result. Airplane represents aircraft takeoff and landing noise, Current represents electronic current noise, and Cabin represents aircraft cabin noise.

3.4.3. Comparisons with Other Methods and Testing Real Speech

To further verify the performance of this method, we also tested the effects of other masks on this method.

The IBM at each time and frequency point is marked as belonging to target speech or noisy speech. If a time and frequency point contains mainly speech information, then the point is marked as 1 in the ideal binary mask; otherwise, it is marked as 0, as follows:

$$IBM(t,f) = \begin{cases} 1, & S(t,f) > N(t,f) \\ 0, & S(t,f) \leq N(t,f) \end{cases}$$

where $S(t,f)$ represents the target speech and $N(t,f)$ represents the noisy speech.

The ideal amplitude mask (IBM) is similar to the IRM, but it considers the impact of the target speech and noisy speech on the results, as follows:

$$IAM(t,f) = \left[ \frac{S(t,f)}{Y(t,f)} \right]^{0.5}$$

where $S(t,f)$ represents the target speech and $Y(t,f)$ represents the mixed speech.

We used the same data to train the IBM and IAM, and then randomly mixed the noise of the test data between 0 and 5 dB. The generated noise data was denoised through the three methods to obtain the enhanced speech. Finally, we used three evaluation indicators (PESQ, STOI, SI-SDR) to measure the performance of each method, and the results are shown in Table 5. Under the three different noise types, our method excelled in all three evaluation metrics (PESQ, STOI, SI-SDR). The improvement value is calculated as the difference between the noisy speech score and the enhanced speech score. For the PESQ, our method achieved an average improvement of 1.04, significantly better than the DNN-IBM’s 0.31 and the DNN-IAM’s 0.94. In STOI, our method showed an average improvement of 0.143, also surpassing the DNN-IBM and DNN-IAM, both at 0.13. For the SI-SDR, our method achieved an average improvement of 11.09, slightly higher than the DNN-IAM’s 11.06 and the DNN-IBM’s 10.90. This demonstrates that our method consistently provided better speech quality and intelligibility while effectively suppressing noise across various noise environments, showcasing its broad superiority.

In addition, we randomly selected 50 pieces of abandoned original control speech data. These data had not been processed by noise type, and the noise in the speech came from real ATC center noise. These data were used to test the performance of the proposed method. The DNSMOS and NISQA were used as evaluation indicators. The results are shown in Table 6. It can be seen that compared with the other two methods, the method proposed in this article shows a good performance using both the DNSMOS and NISQA.
Table 5. Scores of different mask conditions.

<table>
<thead>
<tr>
<th>Noise Type</th>
<th>Metric</th>
<th>Noise</th>
<th>DNN-IBM</th>
<th>DNN-IAM</th>
<th>Our Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cockpit</td>
<td>PESQ</td>
<td>1.10</td>
<td>1.36</td>
<td>2.07</td>
<td>2.10</td>
</tr>
<tr>
<td></td>
<td>STOI</td>
<td>0.71</td>
<td>0.85</td>
<td>0.84</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>SI-SDR</td>
<td>2.54</td>
<td>12.00</td>
<td>12.98</td>
<td>12.84</td>
</tr>
<tr>
<td>HFcommunication</td>
<td>PESQ</td>
<td>1.13</td>
<td>1.46</td>
<td>1.98</td>
<td>2.24</td>
</tr>
<tr>
<td></td>
<td>STOI</td>
<td>0.70</td>
<td>0.83</td>
<td>0.84</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>SI-SDR</td>
<td>2.53</td>
<td>12.33</td>
<td>12.61</td>
<td>12.76</td>
</tr>
<tr>
<td>ATCcenter</td>
<td>PESQ</td>
<td>1.17</td>
<td>1.51</td>
<td>2.17</td>
<td>2.17</td>
</tr>
<tr>
<td></td>
<td>STOI</td>
<td>0.76</td>
<td>0.87</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>SI-SDR</td>
<td>2.54</td>
<td>14.98</td>
<td>15.19</td>
<td>15.27</td>
</tr>
</tbody>
</table>

The black bold number indicates the best result. Cockpit stands for aircraft cockpit noise, ATC center stands for ATC center noise, and HFcommunication stands for high-frequency communication noise.

Table 6. Scores of each method after evaluation by DNSMOS and NISQA.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Noise</th>
<th>DNN-IBM</th>
<th>DNN-IAM</th>
<th>Our Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNSMOS</td>
<td>2.664</td>
<td>2.822</td>
<td>2.894</td>
<td>2.982</td>
</tr>
<tr>
<td>NISQA</td>
<td>2.713</td>
<td>2.881</td>
<td>2.907</td>
<td>2.987</td>
</tr>
</tbody>
</table>

The black bold number indicates the best result.

3.4.4. Impact of Parameter Values

This section investigates the impact of the parameter values, δ and γ, on the experimental outcomes during the adjustment of the IRM weights. The experiment was still conducted on data from the air traffic control speech database, and the average PESQ obtained at three noise conditions was used as the evaluation criterion. By employing a method with fixed parameters, the study delved deeper into analyzing the individual parameter’s influence on the experimental outcomes.

Table 7 shows the comparison of the PESQ values within the LR-Adj method for different γ values when δ = 0.5. Among these, the γ values of 0.4 and 0.5 achieved the optimal average PESQ score, both reaching 2.15. γ = 0.6 exhibited the best PESQ performance at 0 dB. The optimal scores were achieved at −5 dB when the values were set to 0.3 and 0.4. Additionally, γ = 0.4 was also effective at 10 dB. The poorest average PESQ score was seen at γ = 0, scoring only 1.47. In addition, when γ = 1, it produced almost the same results as LR-UnAdj.

Table 7. PESQ comparison of different γ for LR-Adj when δ = 0.5.

<table>
<thead>
<tr>
<th>γ</th>
<th>−5 dB</th>
<th>0 dB</th>
<th>5 dB</th>
<th>10 dB</th>
<th>AVERAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.14</td>
<td>1.29</td>
<td>1.56</td>
<td>1.89</td>
<td>1.47</td>
</tr>
<tr>
<td>0.1</td>
<td>1.34</td>
<td>1.61</td>
<td>1.98</td>
<td>2.38</td>
<td>1.83</td>
</tr>
<tr>
<td>0.2</td>
<td>1.44</td>
<td>1.83</td>
<td>2.23</td>
<td>2.66</td>
<td>2.04</td>
</tr>
<tr>
<td>0.3</td>
<td>1.55</td>
<td>1.86</td>
<td>2.35</td>
<td>2.83</td>
<td>2.15</td>
</tr>
<tr>
<td>0.4</td>
<td>1.55</td>
<td>1.86</td>
<td>2.35</td>
<td>2.83</td>
<td>2.15</td>
</tr>
<tr>
<td>0.5</td>
<td>1.54</td>
<td>1.89</td>
<td>2.35</td>
<td>2.81</td>
<td>2.15</td>
</tr>
<tr>
<td>0.6</td>
<td>1.51</td>
<td>1.91</td>
<td>2.29</td>
<td>2.76</td>
<td>2.12</td>
</tr>
<tr>
<td>0.7</td>
<td>1.49</td>
<td>1.83</td>
<td>2.24</td>
<td>2.77</td>
<td>2.08</td>
</tr>
<tr>
<td>0.8</td>
<td>1.47</td>
<td>1.81</td>
<td>2.21</td>
<td>2.66</td>
<td>2.04</td>
</tr>
<tr>
<td>0.9</td>
<td>1.43</td>
<td>1.77</td>
<td>2.19</td>
<td>2.67</td>
<td>2.00</td>
</tr>
<tr>
<td>1</td>
<td>1.41</td>
<td>1.76</td>
<td>2.17</td>
<td>2.58</td>
<td>1.96</td>
</tr>
</tbody>
</table>

The black bold number indicates the best result.

Table 8 presents the comparison of PESQ values within the LR-Adj for different δ values when γ = 0.5. The optimal average PESQ scores were observed at δ values of 0.8 and 0.9, both reaching 2.17. Conversely, the least favorable average PESQ performance
occurred at $\delta = 1$, scoring only 1.95. Moreover, for $\delta$ values of 0.5, 0.6, and 0.8, they were suitable for handling speech at a $-5 \text{ dB}$ noise environment, with scores reaching 1.54.

Table 8. PESQ comparison of different $\delta$ for LR-Adj when $\gamma = 0.5$.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>-5 dB</th>
<th>0 dB</th>
<th>5 dB</th>
<th>10 dB</th>
<th>AVERAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.41</td>
<td>1.72</td>
<td>2.11</td>
<td>2.56</td>
<td>1.95</td>
</tr>
<tr>
<td>0.9</td>
<td>1.52</td>
<td><strong>1.92</strong></td>
<td>2.36</td>
<td><strong>2.86</strong></td>
<td><strong>2.17</strong></td>
</tr>
<tr>
<td>0.8</td>
<td><strong>1.54</strong></td>
<td>1.91</td>
<td><strong>2.39</strong></td>
<td>2.85</td>
<td><strong>2.17</strong></td>
</tr>
<tr>
<td>0.7</td>
<td>1.53</td>
<td>1.89</td>
<td>2.36</td>
<td>2.82</td>
<td>2.15</td>
</tr>
<tr>
<td>0.6</td>
<td><strong>1.54</strong></td>
<td>1.9</td>
<td>2.36</td>
<td>2.82</td>
<td>2.16</td>
</tr>
<tr>
<td>0.5</td>
<td>1.54</td>
<td>1.87</td>
<td>2.35</td>
<td>2.80</td>
<td>2.14</td>
</tr>
<tr>
<td>0.4</td>
<td>1.53</td>
<td>1.83</td>
<td>2.32</td>
<td>2.78</td>
<td>2.12</td>
</tr>
<tr>
<td>0.3</td>
<td>1.47</td>
<td>1.80</td>
<td>2.23</td>
<td>2.70</td>
<td>2.05</td>
</tr>
<tr>
<td>0.2</td>
<td>1.44</td>
<td>1.77</td>
<td>2.22</td>
<td>2.59</td>
<td>2.00</td>
</tr>
<tr>
<td>0.1</td>
<td>1.38</td>
<td>1.74</td>
<td>2.08</td>
<td>2.56</td>
<td>1.94</td>
</tr>
<tr>
<td>0</td>
<td>1.41</td>
<td>1.71</td>
<td>2.20</td>
<td>2.52</td>
<td>1.96</td>
</tr>
</tbody>
</table>

The black bold number indicates the best result.

Figure 12 illustrates the average PESQ scores from Tables 5 and 6. The red line represents the average PESQ scores for different $\gamma$ values when $\delta = 0.5$, while the green line represents the average PESQ scores for different $\delta$ values when $\gamma = 0.5$.

This section has mainly explored the impact of different values of the parameters on the experimental results when adjusting the IRM weight. From Figure 12, it can be observed that at $\gamma = 0.5$ and $\delta = 0.8$, the speech enhancement performs optimally under various noise conditions, achieving the highest average scores. When the parameter value is close to 0 or close to 1, the PESQ score is very low, especially when $\gamma = 0$ and $\delta = 0.5$. At this time, the adjusted mask is similar to the IBM, and the noise signal is assigned to 0, but unlike the IBM, the target signal is not assigned to 1.

4. Discussion and Conclusions

Aiming at the quality and clarity issues of air traffic control speech in complex noisy environments, an air traffic control speech enhancement method based on an improved DNN-IRM is proposed by using noise and speech from the air traffic control speech database. The interference of noise in the target speech is reduced by refining the network architecture, using LeakyReLU as the activation function, and adjusting the network output IRM weight.

The experimental results show that the proposed method has made some progress in the PESQ, STOI, SI-SNR, and SSIM indicators, proving that adjusting the weight of the network output IRM is feasible for improving speech quality, because this step reduces
the weight of the noise mask. It also proves that LeakyReLU is more suitable for speech enhancement tasks than ReLU, because the loss of the network trained with LeakyReLU is smaller than that of the network trained with ReLU. In addition, when enhancing actual air traffic control speech, the performance of the proposed method is also better than that of the IBM-based method and the IAM-based method.

The amplitude spectrum and phase spectrum can be obtained after Fourier transform of the speech signal, but the mask used in this paper only considers the influence of the amplitude spectrum, and does not consider the influence of the phase spectrum on speech enhancement. Future experiments will consider more advanced masks, such as the complex ratio mask [25], to ensure that the original information in the speech is not lost.

**Author Contributions:** Conceptualization, Y.W. and P.L.; methodology, P.L.; software, P.L.; validation, Y.W., P.L., and S.Z.; formal analysis, Y.W. and S.Z; investigation, P.L.; writing—original draft preparation, Y.W. and P.L.; writing—review and editing, Y.W., P.L., and S.Z.; project administration, Y.W.; funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded in part by the National Key R&D Program of China (program no. 2021YFF0603904) and in part by the Fundamental Research Funds for the Central Universities (program no. ZJ2022-004).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available for privacy reasons.

**Conflicts of Interest:** The authors declare that there are no conflicts of interest regarding the publication of this paper.

**Abbreviations**

The following abbreviations are used in this manuscript:

- **DNN**: Deep neural network
- **IRM**: Ideal ratio mask
- **IBM**: Ideal binary mask
- **IAM**: Ideal amplitude mask
- **PESQ**: Perceptual evaluation of speech quality
- **STOI**: Short-time objective intelligibility
- **SSIM**: Structural similarity index measure
- **T-F**: Time–Frequency
- **SNR**: Signal-to-noise ratio
- **BN**: Batch Normalization
- **LPS**: Log power spectrum
- **STFT**: Short-time Fourier transform
- **ISTFT**: Inverse short-time Fourier transform
- **MSE**: Mean Squared Error
- **SI-SNR**: Scale-invariant signal-to-noise ratio
- **NISQA**: Non-intrusive speech quality assessment
- **DNSMOS**: Deep noise suppression mean opinion score
- **ATC**: Air traffic control

**References**

2. Wu, Y.; Li, G.; Fu, Q. Non-Intrusive Air Traffic Control Speech Quality Assessment with ResNet-BiLSTM. *Appl. Sci.* 2023, 13, 10834. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.