

Article

Robust SNR Estimation Based on Time–Frequency Analysis and Residual Blocks

Longqing Li , Wenjun Xie, Deming Hu *, Jingke Nie , Fei Xie, Zhiping Huang and Yongjie Zhao 

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China; longqing_li@nudt.edu.cn (L.L.); xiewenjun22@nudt.edu.cn (W.X.); jingke_nie@nudt.edu.cn (J.N.); fei.xie@nudt.edu.cn (F.X.); huangzhiping65@nudt.edu.cn (Z.H.); zhaoyongjie17@nudt.edu.cn (Y.Z.)

* Correspondence: hudeming23@nudt.edu.cn

Abstract

Signal-to-noise ratio (SNR) estimation plays a crucial role in communication systems, directly impacting the quality and reliability of signal transmission. This paper proposes a novel deep learning framework aimed at enhancing the accuracy and robustness of SNR estimation. The framework converts received signals into time–frequency matrices as feature inputs, effectively capturing both temporal and spectral characteristics through time–frequency analysis. Extensive experimental results across an SNR range of -5 dB to 15 dB demonstrate that our method achieves a mean squared error (MSE) that closely approaches the theoretical Cramér–Rao bound (CRB), comparable to data-aided (DA) maximum likelihood methods. A quantitative analysis reveals that, even under challenging conditions, such as a low SNR of -5 dB, the model maintains superior accuracy with a mean absolute error (MAE) as low as 0.352 , significantly outperforming traditional M2M4 and NDA estimators. The model's performance was systematically evaluated in a wide range of scenarios, encompassing various signal modulation formats, upsampling factors, multipath fading channels, frequency offsets, phase shifts, and roll-off factors. The evaluation highlights its exceptional generalization capability and robustness, with high performance and stability maintained even in challenging and dynamic environments.

Keywords: deep learning; SNR estimation; time–frequency analysis; residual block; robustness

1. Introduction

Signal-to-noise ratio (SNR) is a core metric for measuring signal quality in the field of communication [1], directly impacting demodulation quality, communication efficiency, and overall system performance. In signal communication, SNR estimation is crucial for applications such as advanced error-correction coding systems [2], automatic modulation recognition [3], and adaptive coding and modulation communication systems [4]. For instance, iterative decoding processes require real-time online estimated SNR values to calculate the soft prior information of received bits. Modulation recognition methods need SNR as prior information to improve the accuracy of signal identification. In cellular and satellite communication systems, SNR information is indispensable for system switching, power control, and channel allocation algorithms. Furthermore, the study of SNR estimation techniques holds significant theoretical and practical value for enhancing the performance and reliability of communication systems as it is not only vital for the algorithms themselves but also provides key reference information for many signal



Academic Editor: Constantine Kotropoulos

Received: 12 January 2026

Revised: 3 February 2026

Accepted: 14 February 2026

Published: 4 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

processing procedures, such as equalization and bit error rate estimation [5]. Therefore, SNR estimation plays an indispensable role in modern communication systems, and its study and application have a profound impact on ensuring the efficiency and reliability of communication.

Traditional SNR estimation methods are mainly divided into two categories: data-aided (DA) [6–8] and non-data-aided (NDA) [9–12]. DA methods rely on training sequences or pilot symbols known to both the transmitter and receiver for SNR estimation. These methods offer relatively accurate SNR estimation results and are generally straightforward to implement, making them particularly suitable for real-time communication environments. Nevertheless, they are constrained by the need to allocate some data transmission resources for SNR estimation, leading to lower bandwidth efficiency and reduced flexibility. This limitation hinders their ability to quickly adapt to sudden environmental changes. In contrast, NDA methods do not require any additional prior information and can complete SNR estimation based solely on the received data stream. This characteristic endows NDA methods with higher bandwidth efficiency and stronger adaptive capabilities, especially in dynamically changing channel conditions. However, these methods typically come with higher computational complexity, and their estimation accuracy may be limited under low-SNR conditions. Additionally, NDA methods are more sensitive to non-Gaussian noise, which somewhat restricts their application range. While both DA and NDA methods exhibit certain advantages in specific scenarios, their adaptability to various modulation types is limited, and the range of SNR estimation is relatively narrow. More critically, these methods often depend on ideal conditions for carrier recovery and timing synchronization [13]. In real-world communication environments, complex conditions such as multipath effects and frequency-selective fading are frequently encountered [14], and the signal format may be unknown. In these challenging scenarios, a universal SNR reference is essential, yet traditional DA and NDA methods may fall short of meeting this requirement.

With the continuous advancement of deep learning technology, its application scope has been widely extended to the field of signal processing, covering critical areas such as signal detection [15], parameter estimation [16], and modulation format recognition. Similarly, many researchers have introduced deep learning into NDA SNR estimation, bringing new vitality to this domain. Based on deep learning, SNR estimation methods leverage their superior adaptability and capability in handling complex scenarios, offering more universal and robust solutions. These methods can directly extract effective features from raw data through end-to-end learning, significantly reducing reliance on prior knowledge and demonstrating higher estimation accuracy and robustness under low-SNR conditions. In recent research, Zheng [17] proposed an innovative SNR estimation model that uses the average power spectrum (APG) as a feature and adopts a residual network as the regression model, conducting extensive simulation experiments. During the experiments, the researchers thoroughly considered the adaptability to different noise distributions (including white and colored noise) and multipath channels (covering Rayleigh and Rician distributions), ultimately achieving lower mean squared error and algorithmic complexity. However, the paper only discussed the recognition effect for one modulation format and did not comprehensively consider the impact of factors like frequency offset, phase shift, and timing errors in real-world environments. Additionally, it lacked systematic research on the generalization performance of the network. Given the limitations of the aforementioned research, this paper proposes a new deep learning framework aimed at further enhancing the accuracy and robustness of SNR estimation. Specifically, this framework first segments the received signals into 1024 sampling points and converts them into time–frequency matrices as feature inputs to the network. Subsequently, using the trained network model, it regenerates test data under various signal modulation formats,

upsampling factors, channel conditions, frequency offsets, phase shifts and roll-off factors to comprehensively evaluate the model's performance. The experimental results show that this framework achieves better SNR estimation effects across multiple scenarios. In summary, the main contributions of this paper are:

- We selected signals of finite length and computed their time–frequency matrices as feature inputs for the deep network. A significant advantage of time–frequency features is that they do not depend on the specific modulation format of the signal. This is because most modulated signals, after undergoing short-time Fourier transform (STFT), appear as continuous segments on the time–frequency plot. Particularly, changes in signal bandwidth or significant frequency deviations are especially prominent in this representation.
- We have developed a novel deep learning framework utilizing residual blocks, which demonstrates superior performance in SNR estimation by achieving lower mean absolute error (MAE) and mean squared error (MSE) compared to current state-of-the-art methods.
- Our training set exclusively comprises signals generated under Gaussian white noise conditions. To further evaluate the robustness of our model, we tested its estimation performance in multipath fading environments, specifically Rayleigh and Rician channels. Additionally, we analyzed the network's generalization capability under varying bandwidth conditions, examining the impact of different oversampling factors and roll-off factors on signal bandwidth. Moreover, we investigated how frequency offsets, phase shifts, and timing errors affect the network's generalization performance under these conditions.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work on SNR estimation. Section 3 presents the framework of the proposed network. Section 4 provides a detailed description of the training dataset, along with the test datasets used to validate the model's generalization performance and the evaluation metrics employed. Section 5 showcases the experimental results. Finally, Section 6 concludes the paper.

2. Related Work

This section briefly reviews the existing traditional methods and deep learning-based approaches for SNR estimation.

2.1. Traditional SNR Estimation Methods

Pauluzzi and Beaulieu [13] compared various SNR estimation techniques, including the split-symbol moments estimator (SSME) [18], maximum-likelihood estimator (MLE) [19], squared signal-to-noise variance (SNV) estimator [20], second- and fourth-order moments (M2M4) estimator [21], and signal-to-variation ratio (SVR) estimator [22]. They derived the Cramér–Rao bound (CRB) [23,24] for complex channels and compared it with that for real channels. Their results indicated that, under data-aided conditions, the ML estimator performs optimally, whereas, under non-data-aided conditions, the performance of SSME and other high-order statistics-based estimators degrades, particularly in high-SNR regions. Sevillano et al. [25] proposed a new non-data-aided SNR estimation algorithm tailored for QPSK data transmission systems, demonstrating its robustness to carrier frequency offsets. This algorithm does not rely on symbol timing recovery or carrier phase knowledge, making it suitable for a wide range of applications. Michael Rice [26] introduced DA and NDA ML SNR estimators for continuous phase modulation (CPM) signals. The study found that DA ML estimators perform better at low SNR levels, while DA, NDA ML estimators, and moment matching (MM) estimators exhibit comparable

performance at medium to high SNR levels. Bellili et al. [27] first proposed an ML SNR estimation method for time-varying single-input multiple-output (SIMO) channels, applicable to linearly modulated signals. Through simulations, they verified that the new estimator accurately estimates the instantaneous SNR for each antenna and demonstrates superior performance under multipath fading conditions. Rugini and Banelli [28] investigated the equivalence between minimum mean squared error (MMSE) and maximum SNR estimators under additive non-Gaussian noise and quantized observations. They showed that, as quantization resolution increases, the performance of suboptimal MMSE estimators approaches that of optimal MMSE estimators. These studies provide valuable theoretical insights but are often focused on specific modulation formats and noise models, which may limit their direct applicability to broader scenarios.

2.2. Deep Learning-Based SNR Estimation Methods

Yang et al. [29] proposed a non-data-aided (NDA) deep learning-based SNR estimation technique that significantly outperforms traditional M2M4 estimators in terms of performance and robustness. This method is applicable to various modulation types and intermediate frequency (IF) signals, demonstrating superior performance across different SNR conditions and strong robustness in non-cooperative environments with phase or frequency offsets. Xie et al. [30] developed a DL-based SNR estimation method using constellation diagrams, implemented through three deep neural networks (AlexNet, InceptionV1, and VGG16). This approach excels in low-SNR scenarios, effectively handling SNR estimation under diverse modulation schemes and channel conditions. The experimental results showed improved estimation accuracy and faster computation speeds, suitable for GPU acceleration. Xu et al. [31] introduced an SNR estimation method based on long-term recurrent convolutional networks (LRCNs), focusing on analyzing micro-Doppler echoes from conical targets. By combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs) with STFT feature extraction, this method achieves highly accurate and robust SNR estimation. Zheng et al. [17] developed a DL regression framework using power spectral density as input, validated through extensive simulations. Their method demonstrated superior accuracy in various noise and multipath channel conditions, exhibiting stronger adaptability with fewer samples and lower computational complexity compared to IQ input methods. These DL-based SNR estimation methods have achieved significant breakthroughs in accuracy and robustness across a wide range of applications. They excel in handling diverse modulation types, channel conditions, and environmental challenges. However, there remains room for improvement in adapting to unknown signal formats and complex real-world environments.

3. Methodology

3.1. System Model

In wireless communication systems, signals are modulated to map binary bit streams into analog signals suitable for wireless transmission. Common modulation methods include phase shift keying (PSK), frequency shift keying (FSK), and quadrature amplitude modulation (QAM). These techniques convert information bits into different signal states or symbols to adapt to the characteristics of wireless channels and enhance transmission efficiency. Subsequently, the modulated signals are passed through pulse-shaping filters to limit the signal bandwidth and reduce inter-symbol interference (ISI). Commonly used pulse-shaping filters, such as the root raised cosine filter (RRC), optimize spectral utilization while maintaining signal integrity. Next, the signals are upconverted to high-frequency carrier signals and transmitted through antennas. During transmission through the wireless channel, signals are subject to various factors, including path loss, multipath effects,

Doppler effects, and noise interference, which significantly impact the quality of signal transmission. Finally, at the receiver end, the received high-frequency signals are down-converted back to baseband signals and undergo a series of processing steps, such as timing recovery, carrier synchronization, and demodulation, to restore the original signals. Without considering subsequent processing like matched filtering, the received signal $y(t)$ can be expressed as

$$y(t) = x(t)e^{j(2\pi f_0 t + \varphi_0)} * h(t) + n(t), \quad (1)$$

where f_0 represents the frequency offset, φ_0 represents the phase offset, $x(t)$ is the modulated high-frequency signal, $h(t)$ is the channel response of the transmitted signal, and $n(t)$ is additive white Gaussian noise (AWGN) with a mean of 0 and variance σ^2 . The modulated high-frequency signal $x(t)$ can be specifically represented as

$$x(t) = \sum_n a_n g_T(t - nT_s - \varepsilon T_s) e^{j(2\pi f_c t + \varphi_n)}. \quad (2)$$

In this expression, a_n denotes the complex value of the n -th symbol, and $g_T(t)$ is the impulse response of the pulse-shaping filter, which limits the signal bandwidth and reduces inter-symbol interference (ISI). ε represents the timing error, T_s is the symbol period, and the term $e^{j(2\pi f_c t + \varphi_n)}$ models the upconversion process, with f_c denoting the carrier frequency and φ_n the phase of the n -th symbol. The bandwidth B of the signal is given by

$$B = \frac{1 + \alpha}{T_s}, \quad (3)$$

where α is the roll-off factor of the pulse-shaping filter, ranging between 0 and 1. The number of samples per symbol (SPS), denoted as N , is defined as the ratio of the sampling frequency f_s to the symbol rate $1/T_s$: $N = \text{SPS} = f_s T_s$. Consequently, the sampling period is defined as $T = 1/f_s = T_s/N$. Given M as the total number of modulated symbols, the total length of the sampled signal is MN , where the sample index n ranges from 0 to $MN - 1$.

The discrete-time transmitted signal $x[n]$, sampled at intervals of T , is expressed as

$$x[n] = x(nT) = \sum_k a_k g_T(nT - kT_s - \varepsilon T_s) e^{j(2\pi f_c nT + \varphi_k)}. \quad (4)$$

Considering the channel impulse response $h[n]$, the channel noise $n[n]$, and the quantization noise $n_q[n]$ introduced by the ADC, the discrete-time received signal $y[n]$ is modeled as

$$y[n] = \left(x(nT) e^{j(2\pi f_0 nT + \varphi_0)} * h[n] \right) + n[n] + n_q[n], \quad (5)$$

where f_0 and φ_0 represent the frequency and phase offsets, respectively. For computational simplicity, this can be approximated as

$$y[n] = s[n] * h[n] + w[n], \quad (6)$$

where $s[n]$ represents the effective signal component and $w[n]$ represents the aggregate noise.

The objective is to estimate the signal-to-noise ratio (SNR) and the ratio of symbol energy to noise power spectral density (E_s/N_0) from the received samples. The relationship between E_s/N_0 and SNR for complex signals is given by

$$E_s/N_0 \text{ (dB)} = 10 \log_{10} \left(\frac{T_s}{T} \right) + \text{SNR (dB)} = 10 \log_{10}(N) + \text{SNR (dB)}. \quad (7)$$

Specifically, the total energy of the received signal component is calculated as

$$E_{signal} = \sum_{n=0}^{MN-1} |s[n]|^2. \quad (8)$$

Consequently, the estimated SNR, defined as the ratio of total signal energy to total noise energy, can be written as

$$\text{SNR (dB)} = 10 \log_{10} \left(\frac{\sum_{n=0}^{MN-1} |s[n]|^2}{\sum_{n=0}^{MN-1} (|n[n]|^2 + |n_q[n]|^2)} \right). \quad (9)$$

3.2. Data Format

In this paper, we propose a method where the received signal, after being subjected to the STFT, is used as the input for a neural network. Specifically, the STFT of the signal $y(t)$ is performed, with the expression given by

$$Y(m, k) = \sum_{n=0}^{MN-1} y[n] w[n - mR] e^{-j(2\pi \frac{k}{L} n)}. \quad (10)$$

In this expression, $Y(m, k)$ represents the STFT of the original signal, where m denotes the time frame index and k denotes the frequency index. The window function $w[n - mR]$ is applied to the original signal, with R being the hop size, the number of samples between successive windows, which also determines the overlap between adjacent windows. L is the number of points used for the discrete Fourier transform (DFT), matching the number of points used in the original STFT.

After applying the STFT to the data, the energy of the transformed signal can be calculated using the following expression:

$$E_s(m, k) = \sum_m \sum_k |S(m, k)|^2, \quad (11)$$

where $S(m, k)$ denotes the STFT of the signal $s(n)$.

According to Parseval's energy theorem,

$$\sum_{n=0}^{L-1} |x[n]|^2 = \frac{1}{L} \sum_{k=0}^{L-1} |X[k]|^2, \quad (12)$$

the energy of the signal and noise in the discrete time domain is equal to the energy in the discrete frequency domain. Thus,

$$\rho = 10 \log_{10} (E_s(m, k) / E_n(m, k)) = 10 \log_{10} (E_s / N_0). \quad (13)$$

Consequently, performing the STFT on the signal does not alter its SNR. While the STFT transforms the signal into a time–frequency representation, it preserves the relative proportions of signal and noise energies, ensuring that the SNR remains consistent across domains.

3.3. Structure of the Network

The proposed network architecture, as illustrated in Figure 1, primarily consists of three residual blocks [32] and a spatial attention mechanism. The input to the network is a time–frequency matrix of size $227 \times 227 \times 1$, which captures the spectral characteristics of the input signal over time. This matrix first passes through two convolutional layers with kernel sizes of 3×3 , each followed by a batch normalization (BN) layer to regularize

the model and accelerate training, and a rectified linear unit (ReLU) activation function to introduce non-linearity. After these initial convolutional layers, the processed data flows into a basic residual block, which helps to mitigate the vanishing gradient problem and facilitates the learning of complex features. Following the residual block, a spatial attention mechanism selectively emphasizes important spatial regions of the feature maps, enhancing the network's ability to focus on relevant features while suppressing noise. Subsequently, the data passes through two additional residual blocks to further refine the learned features. An average pooling layer then reduces the spatial dimensions of the feature maps, summarizing the most significant features. Finally, the output from the average pooling layer is flattened and passed through three fully connected layers, which perform the final regression task to predict the SNR.

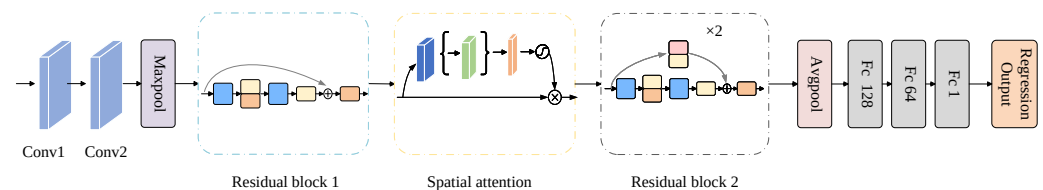


Figure 1. The proposed network architecture.

3.3.1. Residual Block

To address the potential degradation problem associated with increasing network depth and to ensure effective feature transmission, we employ residual blocks (BasicBlocks) as the core building units. Each residual block contains two 3×3 convolutional layers, a kernel size selected to achieve a balance between parameter efficiency and the ability to capture local time–frequency patterns. The specific structure, illustrated in Figure 2, is detailed as follows: First, the input feature map passes through a 3×3 convolutional layer with a stride of 1 or 2. When a stride of 2 is used, it adjusts the spatial dimensions, effectively increasing the receptive field to capture broader spectral characteristics essential for global SNR estimation. Next, a batch normalization layer stabilizes the training process by reducing internal covariate shift, followed by an ReLU activation function to introduce non-linearity. Subsequently, the feature map passes through another 3×3 convolutional layer with a stride of 1, maintaining the spatial dimensions. Finally, a skip connection adds the processed output to the input feature map. This residual connection is scientifically critical: it mitigates the vanishing gradient problem, allowing the gradient to flow through the network directly. This ensures that the model can learn identity mappings and preserves fine-grained signal features that might otherwise be lost during deep non-linear transformations. If necessary, the skip connection includes a downsample unit to ensure that the input and output feature maps match in both size and number of channels.

In the latter two BasicBlocks, a downsample operation is introduced to facilitate smooth connections between different residual blocks. Specifically, the downsample unit consists of a 1×1 convolutional layer and a batch normalization layer. The 1×1 convolutional layer adjusts the number of channels, while a 3×3 convolutional layer with a stride of 2 reduces the spatial dimensions. Through this downsample operation, we reduce the computational complexity while further expanding the receptive field, enabling the network to aggregate global noise statistics relative to the signal energy.

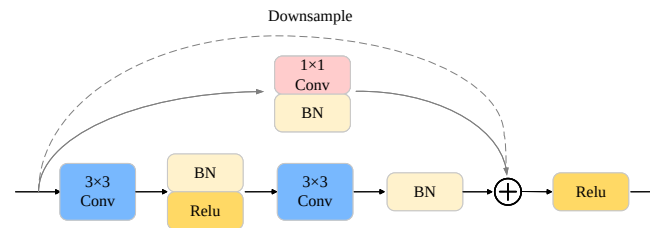


Figure 2. The basic structure of the residual block.

3.3.2. Spatial Attention

SNR estimation relies heavily on distinguishing effective signal energy regions from background noise within the time–frequency domain. Since signal features in time–frequency maps often exhibit local continuity while noise is globally distributed, we designed a spatial attention mechanism to enhance the network’s sensitivity to these informative regions. Firstly, a 1×1 convolutional layer $Q_1 \in \mathbb{R}^{1 \times 1}$ is applied to aggregate channels of the input feature map F , producing a two-dimensional feature mapping. This step serves as a bottleneck to fuse cross-channel information and reduce computational overhead. This is followed by introducing non-linearity with the ReLU activation function f . Then, two 3×3 dilated convolution kernels $Q_2, Q_3 \in \mathbb{R}^{3 \times 3}$ (where $Q_2 = Q_3$) are used to further process the generated two-dimensional feature mapping. The selection of dilated convolutions here is scientifically motivated: they allow the network to obtain a larger receptive field and capture multi-scale spatial information—such as the continuity of signal harmonics—without reducing spatial resolution via pooling. The ReLU activation function f is also applied after each convolutional layer. Following this, another 1×1 convolutional layer $Q_4 \in \mathbb{R}^{1 \times 1}$ (identical to Q_1) is concatenated for adjusting the number of channels in the feature mapping and again passed through the ReLU activation function f . Finally, the output is fed into the Sigmoid activation function σ , generating a spatial attention weight map $M_s(F)$ normalized to the $[0, 1]$ interval. This map acts as a feature filter, suppressing noise-dominant regions and emphasizing signal-dominant regions. The map is then element-wise multiplied (skip connection) with the original feature map F to achieve dynamic weighting. As illustrated in Figure 3, through the steps outlined above, the spatial attention mechanism effectively captures the importance distribution of different positions within the feature map, which can be formulated as

$$M_s(F) = \sigma(Q_4 * f(Q_3 * f(Q_2 * f(Q_1 * F)))). \quad (14)$$

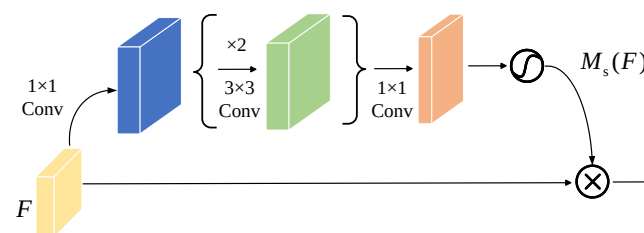


Figure 3. The structure of the spatial attention mechanism.

3.3.3. Network Output and Regression Prediction

In the final stages of the network architecture, the processed data first passes through an average pooling layer to aggregate spatial information. The average pooling layer reduces the spatial dimensions of the feature maps while retaining important statistical characteristics, generating a fixed-size feature vector that summarizes global information for subsequent fully connected layers. Following the average pooling layer, the flattened feature vector is fed into three fully connected layers with 128, 64, and 1 neurons, respec-

tively. These layers progressively reduce the dimensionality to map high-level features to specific prediction values. Each layer uses activation functions to introduce non-linearity, enabling the model to learn complex mappings. The last layer outputs a single prediction value. Finally, a regression layer processes the network's output to produce the final prediction. This regression layer calculates the difference between predicted and true values using the root mean squared error (RMSE) as the loss function to evaluate model performance. The RMSE formula is given by

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\rho_i - \hat{\rho}_i)^2} \quad (15)$$

where N is the number of samples, ρ_i is the true value of the i -th sample, and $\hat{\rho}_i$ is the predicted value of the i -th sample.

4. Experiments

4.1. Training Parameter Settings

Model training and simulations were performed on a computing system equipped with an Intel Core i9-9750H CPU, 64 GB of RAM, and an NVIDIA RTX 4000 GPU, utilizing MATLAB 2023b for all experiments. GPU acceleration was fully leveraged to expedite the computational processes. A dataset for network training was generated through simulations, containing 100 data entries under identical conditions. The dataset was divided into training, validation, and test sets in a 7:1.5:1.5 ratio, with detailed dataset parameters outlined in Table 1. During the network training process, the Adam optimizer was employed, with the maximum number of training epochs set to 20, 46 samples processed per batch, an initial learning rate of 1×10^{-5} , and random shuffling of the data at the beginning of each epoch to ensure data diversity. Additionally, the transformed validation dataset (transformedValFds) was utilized to periodically evaluate model performance, and a training progress chart was enabled for real-time monitoring of the training process, ensuring visualization and controllability of the training outcomes. Upon completion of training, the model's performance was assessed using the designated portion of the test set. The network training process and the evaluation of network performance on a portion of the test set are detailed in Algorithm 1 and Algorithm 2, respectively.

Table 1. Training dataset parameters.

Parameter	Value
SNR range	[−5 15] dB with interval 0.5 dB
fs	61.44 M
Modulation	8PSK
SPS	[2 20] with interval 2
Frequency offset	0
Phase offset	0
α	0.5
Channel	AWGN
Training set	28,700
Validation set	6150
Test set	6150

Algorithm 1 SNR Estimation Based on STFT

Require: Training set $\mathcal{T} = \{(y_t^{(i)}, \rho_t^{(i)})\}_{i=1}^m$, validation set $\mathcal{V} = \{(y_v^{(j)}, \rho_v^{(j)})\}_{j=1}^n$, mini-batch size N_t , learning rate η , maximum number of iterations λ

Ensure: Trained model $\mathcal{G}_P(\bullet)$

- 1: Preprocess each sample in $\mathcal{T}$ and $\mathcal{V}$
- 2: Initialize network parameters
- 3: Initialize iteration counter $t = 0$
- 4: **while** $t < \lambda$ **do**
- 5: Draw a mini-batch N_t of samples from $\mathcal{T}$
- 6: Compute loss function value and update network parameters
- 7: Evaluate model performance on $\mathcal{V}$
- 8: **if** early stopping criteria are met **then**
- 9: Break loop
- 10: **end if**
- 11: Increment iteration counter $t = t + 1$
- 12: **end while**
- 13: **return** The trained model $\mathcal{G}_P(\bullet)$

Algorithm 2 Testing Procedure

Require: Test set $\mathcal{S} = \{(y_s^{(k)}, \rho_s^{(k)})\}_{k=1}^p$, the trained model $\mathcal{G}_P(\bullet)$

Ensure: Predicted SNR ρ_e and its error analysis

- 1: Preprocess each test sample as done during training
- 2: **for** each test sample in $\mathcal{S}$ **do**
- 3: Preprocess the sample
- 4: Predict SNR ρ_e using $\mathcal{G}_P(\bullet)$
- 5: **end for**
- 6: Calculate evaluation metrics (e.g., MAE and MSE) based on predicted ρ_e and actual ρ
- 7: Plot the relationship between actual SNR and predicted SNR, along with error distribution under different SNR conditions
- 8: **return** Predicted SNR ρ_e and its corresponding error analysis charts

4.2. Generalization Testing Parameter Settings

To further evaluate the model's robustness, new test data was generated that the original model had not seen before, with different training parameters. By applying the control variable method, only one parameter at a time was varied while keeping the others consistent with the original model to assess the impact of changes in that parameter on performance. Specifically, the estimation performance was tested in multipath fading environments, particularly under Rayleigh and Rician channel conditions. The network's generalization capability under varying bandwidth conditions was analyzed by examining the effects of different SPS and roll-off factors on signal bandwidth. Additionally, the influence of frequency offsets and phase shifts on the network's generalization performance under these conditions was investigated. To ensure reliability, 100 independent data samples were generated under identical conditions for testing purposes. Detailed parameter settings are provided in Table 2, and the algorithmic process is outlined in Algorithm 3.

Table 2. Generalization test parameters.

Parameter	Value
SNR range	[−5 15] dB with interval 1 dB
f_s	61.44 M
Modulation	16QAM/DQPSK/GMSK/8PSK
SPS	[3 19] with interval 2
Frequency offset	± 10 MHz, ± 1 kHz, 0 Hz

Table 2. *Cont.*

Parameter	Value
Phase offset	$\pm \frac{\pi}{2}, \pm \frac{\pi}{4}, 0$
α	[0.1 0.9] with interval 0.1
Channel	Rayleigh/Rician/AWGN

Algorithm 3 Generalized Performance Testing for Various Parameters

Require: Pre-trained network $\mathcal{N}$, test data directory, parameters to test (e.g., channel type and SPS)

Ensure: Performance metrics and analysis for each parameter

- 1: Load pre-trained network $\mathcal{N}$
 - 2: Create file datastore from test data directory
 - 3: **for** each parameter to test **do**
 - 4: Select relevant samples based on the current parameter from the test set
 - 5: Transform selected data into the format required by $\mathcal{N}$
 - 6: **while** test set has data **do**
 - 7: Predict SNR using $\mathcal{N}$
 - 8: Collect predictions and true labels for further analysis
 - 9: **end while**
 - 10: Calculate performance metrics (e.g., MAE and MSE) for the current parameter
 - 11: Plot results for visualization
 - 12: **end for**
 - 13: **return** Analysis of model’s generalized performance across different parameters
-

4.3. Evaluation Metrics

To quantitatively assess the algorithm’s performance, we utilize mean absolute error (MAE) and mean squared error (MSE). MAE measures the discrepancy between the estimated SNR ($\hat{\rho}_i$) and the true SNR (ρ_i). In our simulations, the true SNR ρ_i is rigorously defined by adjusting the signal energy (E_s) and AWGN variance (σ^2) according to specified intervals, thereby establishing an exact benchmark for evaluation. Furthermore, MSE is included to capture both the bias and variance of the estimator. The specific formulas are defined as follows:

$$MAE\{\hat{\rho}\} = \frac{1}{L} \sum_{i=1}^L |\hat{\rho}_i - \rho_i| \quad (16)$$

$$MSE\{\hat{\rho}\} = \frac{1}{L} \sum_{i=1}^L (\hat{\rho}_i - \rho_i)^2 \quad (17)$$

where L is the total number of samples, $\hat{\rho}_i$ is the estimated SNR for the i -th sample, and ρ_i is the actual SNR for the i -th sample.

To further assess the absolute performance of each estimator, we employed the Cramér–Rao bound (CRB), which provides a theoretical lower bound on the variance of any unbiased estimator. The formula for the CRB is given as follows [26]:

$$CRB = \left(\frac{10}{\ln(10)} \right)^2 \frac{1}{L} \left[\frac{2N}{E_s/N_0} + 1 \right]. \quad (18)$$

5. Simulation Results and Discussion

5.1. Comparison with Traditional Estimation

Maximum likelihood estimation and the M2M4 method are two classical approaches for estimating SNR. In this study, we evaluate our trained 8PSK and GMSK signals with eight SPS against several established methods. These include the TXDA method based on

MLE as proposed by Pauluzzi [13], the DA and NDA MLE methods for GMSK signals as described in Rice's work [26], and the M2M4 technique [22]. Each algorithm was rigorously assessed through Monte Carlo simulations, consisting of 100 independent runs.

For 8PSK modulation, the performance is detailed in Figure 4. Figure 4a shows that the predicted SNR mean of the proposed method aligns well with the true SNR. However, the M2M4 method struggles to deliver accurate estimates for 8PSK signals at an SPS of 8, as reflected by the higher error rates in Figure 4b. Notably, Figure 4c presents the mean squared error (MSE) comparison. The enlarged detail in Figure 4c specifically highlights the performance in the range of 0 to 15 dB, revealing that, while the M2M4 estimator exhibits significant deviation, the proposed method maintains an extremely low MSE that closely approaches the theoretical Cramér–Rao bound (CRB), comparable to the data-aided (DA) ML method.

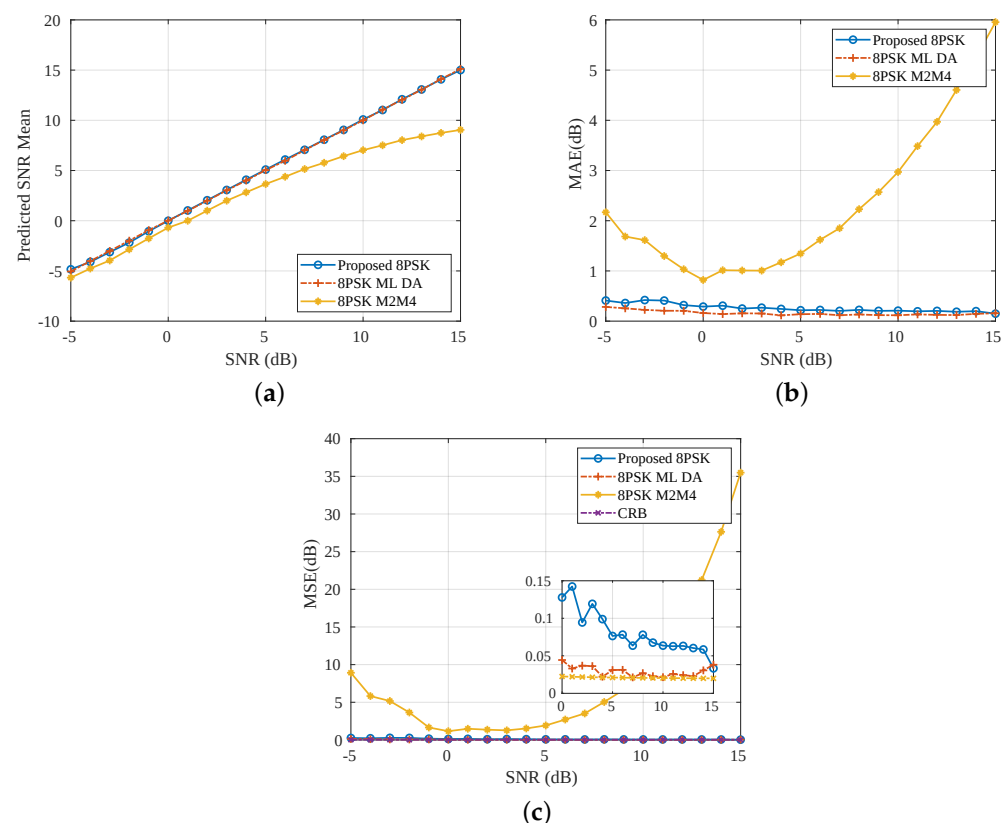


Figure 4. Performance comparison with traditional methods for the 8PSK modulation. (a) Predicted SNR mean; (b) MAE; (c) MSE.

For GMSK signals, which were subjected to a frequency offset of 1 kHz and a phase offset of $\frac{\pi}{4}$, the results are illustrated in Figure 5. Figure 5a confirms the linearity of the predicted means. The advantages of the proposed method are further emphasized in the error analysis. Figure 5b depicts the mean absolute error (MAE), where the enlarged view demonstrates that the proposed method achieves lower and more stable error values compared to the traditional NDA ML method in the high-SNR region. Similarly, Figure 5c illustrates the MSE, with the enlarged inset further confirming that the proposed method's accuracy converges effectively towards the CRB, outperforming the NDA baselines.

Collectively, these figures demonstrate that, while conventional methods tend to offer superior estimation precision under high-SNR conditions, the newly proposed method exhibits a distinct advantage in terms of MAE and MSE, particularly in low-SNR scenarios. An important feature of the novel approach is its independence from prior knowledge of the transmitted signal specifics, which contrasts with traditional SNR estimation meth-

ods that exhibit a higher dependency on modulation formats. By presenting these detailed comparative analyses, we underscore the robustness of the proposed method across diverse conditions.

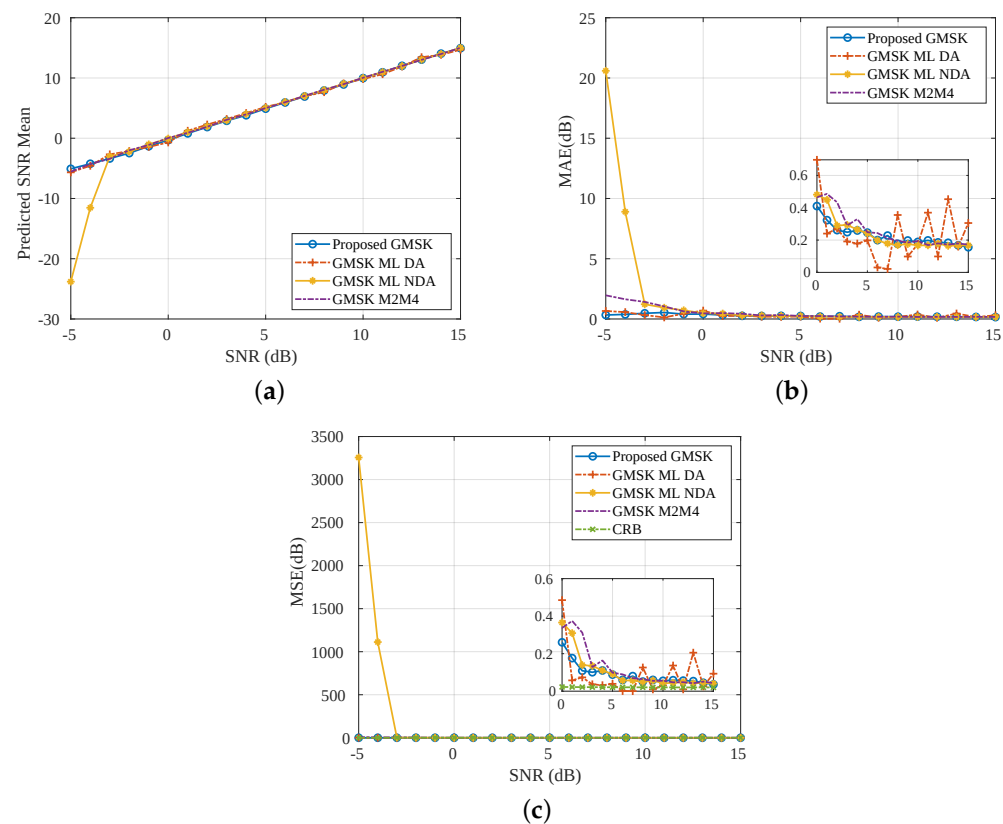


Figure 5. Performance comparison with traditional methods for the GMSK modulation. (a) Predicted SNR mean; (b) MAE; (c) MSE.

5.2. Comparison of Different Networks

To evaluate the performance of different networks, we conducted a comparative analysis against two existing methods: the APG method introduced by Zheng [17], which takes the averaged power spectral periodogram as its input, and the LRCN method proposed by Xu [31], which uses the spectrogram generated by the STFT as its input.

As shown in Figure 6b, the proposed method exhibits lower MAE values across various SNR levels compared to the other methods. This suggests that it achieves smaller systematic errors in the estimation process. Furthermore, as depicted in Figure 6c, our method maintains low MSE values at different SNR levels, with these MSE values approaching the CRB. These results indicate that the estimation performance of the proposed method is close to the theoretical limit. The comparison highlights the robustness and efficiency of the proposed method, showing its ability to maintain high accuracy under varying conditions. However, it is important to note that each method has its own strengths and may perform differently depending on specific application scenarios.

To evaluate computational cost, we compared the parameters and FLOPs of the proposed method with APG and LRCN in Table 3.

Table 3 shows that, while APG is the lightest, its accuracy is limited. LRCN requires huge storage, whereas our method is much more compact. Although our method has higher FLOPs to process detailed time–frequency features, it achieves the best balance between storage efficiency and high-precision robust estimation.

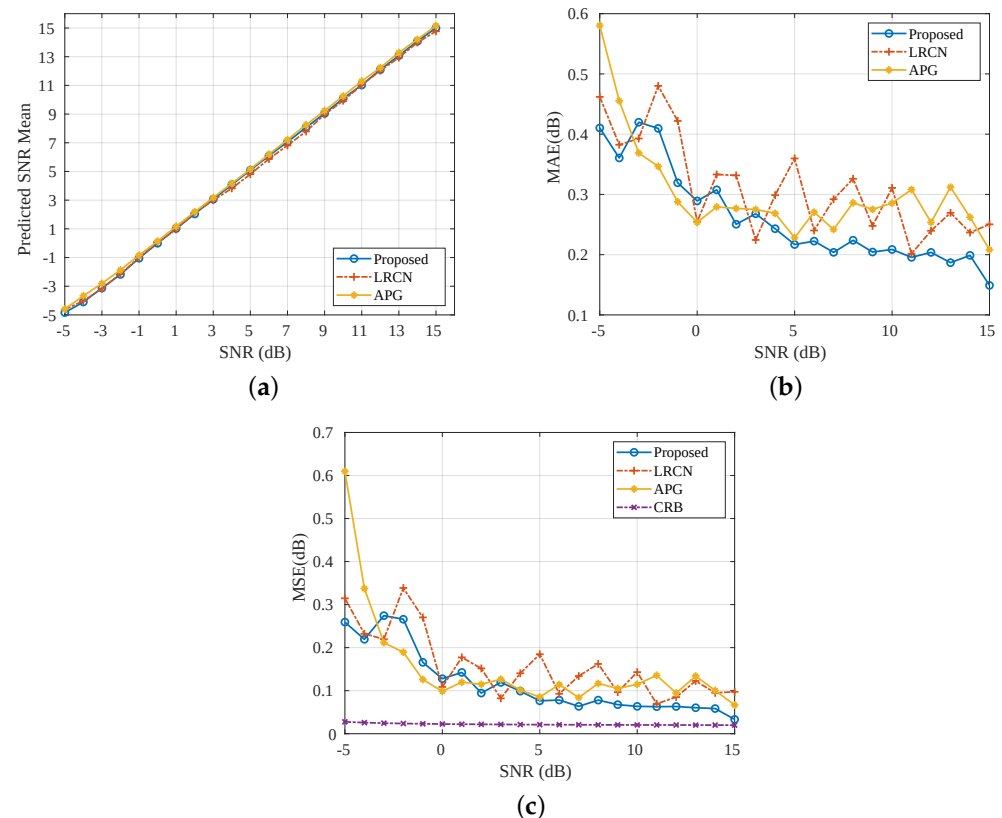


Figure 6. Performance of different networks: (a) predicted SNR mean; (b) MAE; (c) MSE.

Table 3. Computational complexity comparison.

Method	Params (M)	FLOPs (G)
Proposed	1.63	9.72
LRCN	85.80	2.48
APG	0.13	0.0194

5.3. Performance Evaluation of Various Modulation Formats

To evaluate the generalization performance of the proposed deep learning framework across different modulation formats, we conducted extensive tests using data from unseen modulation formats. This framework was trained exclusively on data from the 8PSK modulation format and then applied to generate and test data from a variety of modulation formats, including 8PSK, 16QAM, GMSK, and DQPSK. As demonstrated in Figure 7, the experimental results show that the proposed method can accurately estimate the SNR of signals across different modulation formats, thereby validating its strong generalization capability.

Specifically, for GMSK modulation, which features continuous phase frequency offset and exhibits more significant variations in its time–frequency representation compared to other modulation formats, the method shows robust performance. Under low-SNR conditions (below 0 dB), the MAE and MSE for GMSK are relatively higher, reflecting the challenge of distinguishing between signal and noise. However, at SNRs above 0 dB, the estimation accuracy for GMSK outperforms other modulation formats. Overall, the proposed method can effectively estimate the SNR of most modulation formats without prior knowledge of the modulation scheme, making it particularly suitable for real-world applications where received signals may have unknown modulation formats.

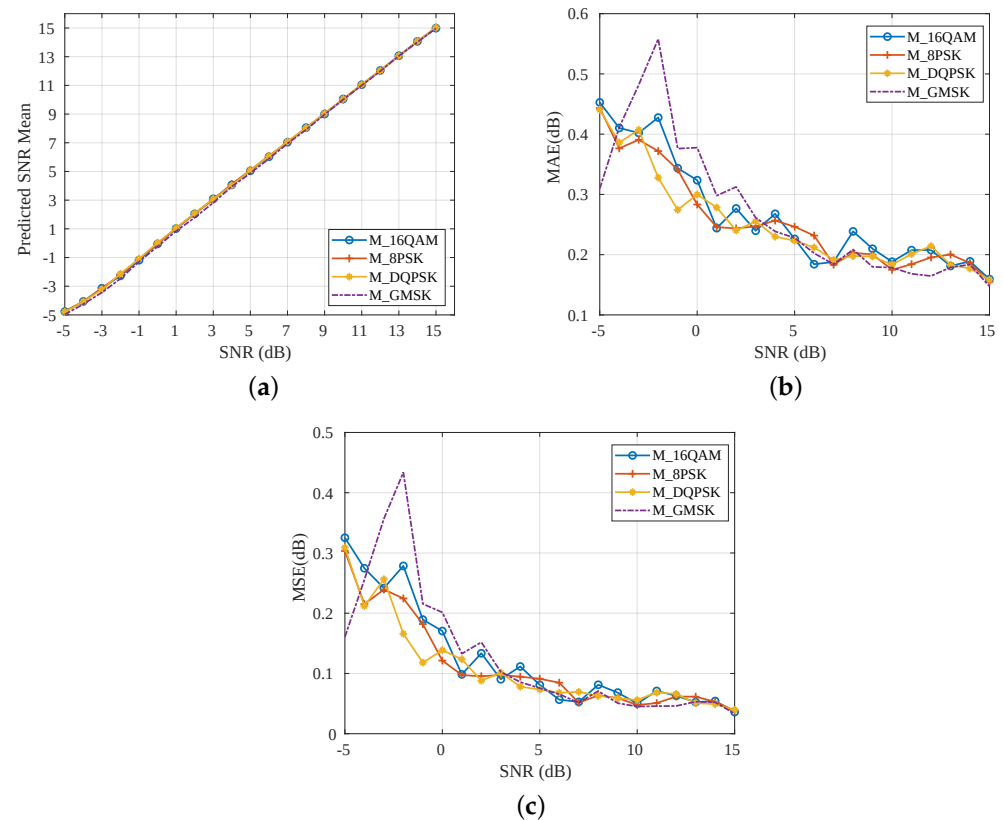


Figure 7. Performance comparison of different modulation formats: (a) predicted SNR mean; (b) MAE; (c) MSE.

5.4. Performance Evaluation in Various Channel Conditions

To validate the performance of the proposed network under channel fading conditions, we generated a test set where signals were transmitted through AWGN, Rician, and Rayleigh channels. The specific parameters for the Rician and Rayleigh channels are detailed in Table 4 and Table 5, respectively. The test results are shown in Figure 8, demonstrating the network’s performance across different channel conditions. Specifically, the AWGN channel exhibits the best estimation performance, with the lowest MAE and MSE across all the SNR intervals, particularly stabilizing at high SNR levels. This superior performance is primarily due to the network being trained exclusively on AWGN channel data. In contrast, the Rician channel shows the poorest estimation accuracy, especially under low-SNR conditions, where the MAE and MSE values are higher and decrease more slowly. The Rayleigh channel’s performance falls between the two extremes; it outperforms the Rician channel at low SNR and approaches the performance of the AWGN channel at high SNR.

Table 4. Parameter settings for Rayleigh channel.

Item	Value
SampleRate	61.44×10^6
PathDelays	$[0 \ 1 \times 10^{-4}]$
AveragePathGains	[0 1]
MaximumDopplerShift	30
PathGainsOutputPort	false
NormalizePathGains	true

Overall, these figures illustrate the impact of different channel conditions on SNR estimation. Despite this, the proposed network provides reliable SNR estimation results across various channel conditions, showcasing its strong generalization and robustness. Notably, given that the network was trained only on AWGN channel data, this performance highlights the method's potential value in practical communication systems.

Table 5. Parameter settings for Rician channel.

Item	Value
SampleRate	61.44×10^6
PathDelays	$[0.5 \times 10^{-5} \ 1 \times 10^{-5}]$
AveragePathGains	$[0.1 \ 0.5 \ 0.2]$
MaximumDopplerShift	30
KFactor	3
DirectPathDopplerShift	5
DirectPathInitialPhase	0.5

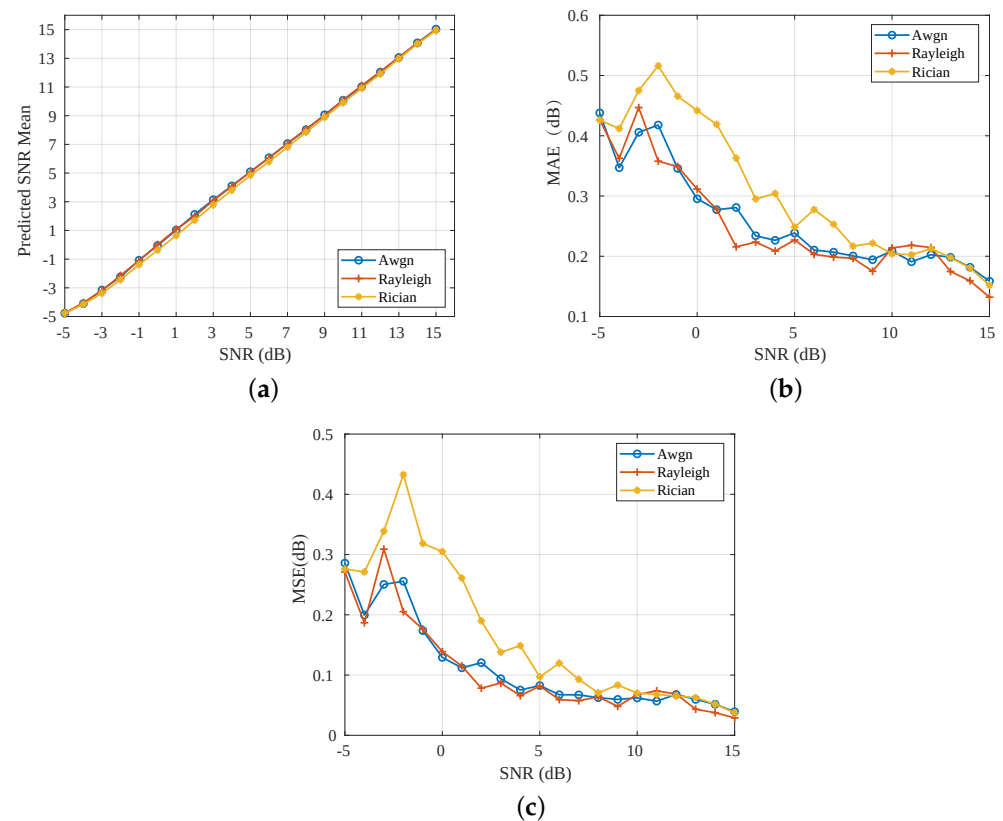


Figure 8. Performance evaluation in various channel conditions: (a) predicted SNR mean; (b) MAE; (c) MSE.

5.5. Performance Testing at Different SPS

To evaluate the network's generalization capability under different signal bandwidths, we generated signals with SPS parameters different from those in the training set, ranging from 3 to 19 with an interval of 2. According to Equation (3), a smaller SPS value results in a wider signal bandwidth. We extracted the MAE and MSE values at SNR levels of -5 dB and 15 dB, with the test results shown in Tables 6 and 7. Under low-SNR (-5 dB) conditions, both MAE and MSE are relatively high and exhibit significant fluctuations as SPS changes, indicating that low-SNR environments have a substantial impact on estimation performance. In contrast, under high-SNR (15 dB) conditions, MAE and MSE significantly decrease and stabilize as SPS increases, demonstrating better estimation accuracy and

robustness. Selected SPS values are plotted in Figure 9, showing that SPS has a considerable effect on SNR estimation; particularly, the estimation error is larger when $SPS = 3$, and the precision improves with increasing SPS before eventually stabilizing. When SPS is small, the signal bandwidth becomes wider, eventually occupying the entire time–frequency spectrum, making this method unsuitable for SPS values below 3. However, in most practical communication systems, higher SPS values are generally used to ensure better noise resistance and higher estimation accuracy.

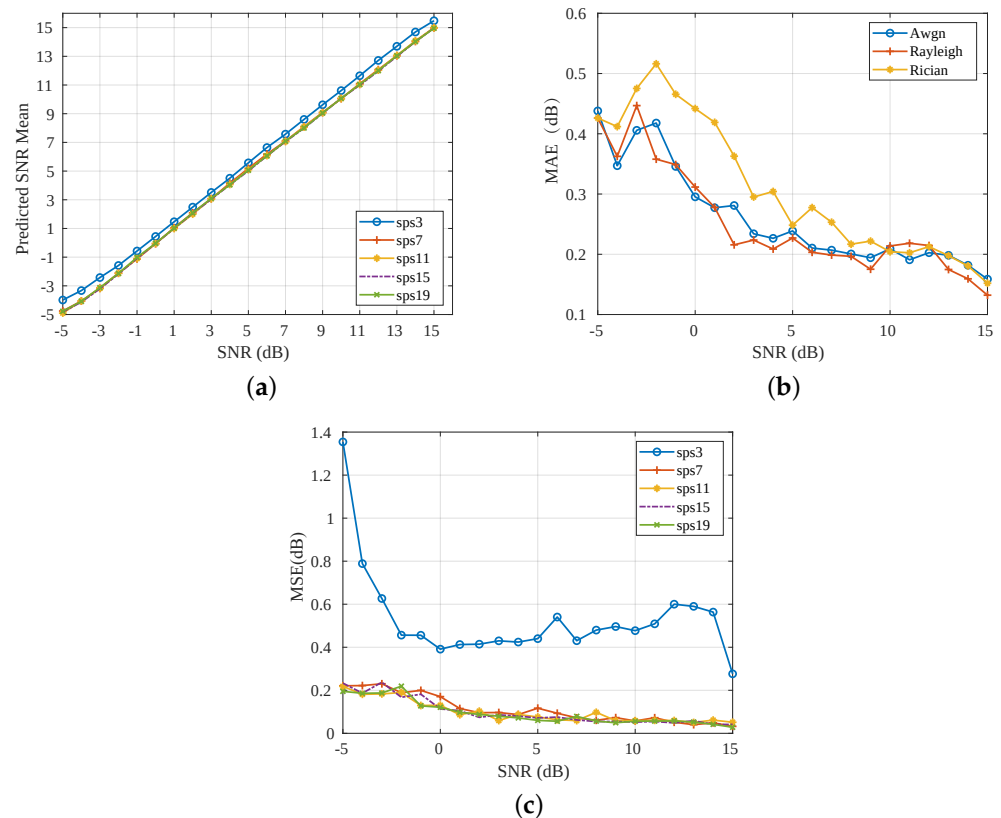


Figure 9. Performance of different SPS: (a) predicted SNR mean; (b) MAE; (c) MSE.

Table 6. Test results at $SNR = -5$ dB.

SPS	MAE Value	MSE Value
3	1.025	1.354
5	0.531	0.398
7	0.381	0.221
9	0.384	0.241
11	0.365	0.218
13	0.398	0.248
15	0.381	0.233
17	0.384	0.230
19	0.352	0.195

Table 7. Test results at $SNR = 15$ dB.

SPS	MAE Value	MSE Value
3	0.486	0.276
5	0.143	0.033
7	0.145	0.032
9	0.169	0.046

Table 7. *Cont.*

SPS	MAE Value	MSE Value
11	0.186	0.052
13	0.146	0.035
15	0.167	0.041
17	0.150	0.034
19	0.141	0.028

5.6. Performance Evaluation Across Various Roll-off Factors

In addition to SPS, the roll-off factor is another critical element influencing signal bandwidth. To evaluate the network's generalization capability across varying roll-off factors, we utilized a network trained exclusively on data with a roll-off factor of 0.5 and generated a test set with roll-off factors ranging from 0.1 to 0.9, simulating the potential scenarios of pulse-shaping filter roll-off factors at the transmitter. As shown in Figure 10, the x-axis represents the roll-off factor, while the y-axis shows both the MAE and the MSE, with data selected for analysis under five different SNR conditions.

The results indicate that, under low-SNR conditions, the roll-off factor significantly impacts MSE and MAE, particularly when the roll-off factor is low, leading to higher errors. As the roll-off factor increases, the error decreases gradually and stabilizes, suggesting that a higher roll-off factor can enhance estimation accuracy in low-SNR environments. However, as SNR continues to increase, both MSE and MAE decrease substantially, and the differences between various roll-off factors become negligible, demonstrating that the system achieves highly precise and stable estimates in high-SNR conditions, almost unaffected by changes in the roll-off factor. Overall, these findings highlight the influence of the roll-off factor on the performance of signal estimation, especially in low-SNR environments where selecting a relatively higher roll-off factor can significantly improve system accuracy. Meanwhile, under high-SNR conditions, regardless of variations in the roll-off factor, the system maintains excellent robustness and stability. This insight holds significant implications for the design and optimization of practical communication systems.

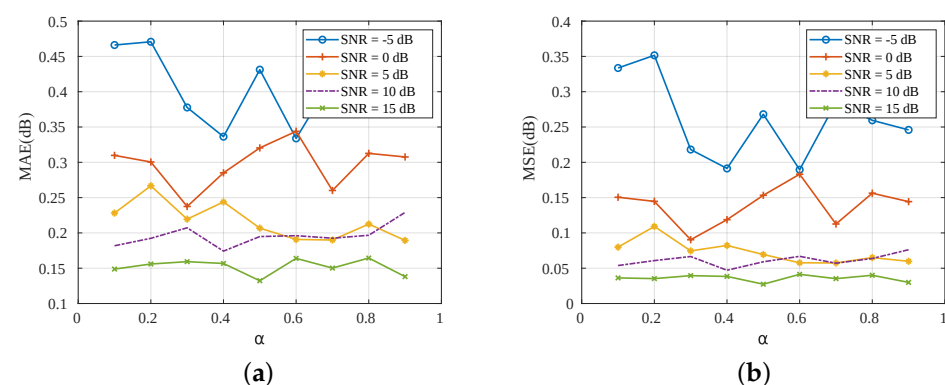


Figure 10. Performance evaluation of different roll-off factors: (a) MAE; (b) MSE.

5.7. Performance Testing Under Different Frequency Offsets

A frequency offset may exist between the receiver and transmitter, leading to issues such as ISI, phase errors, and other synchronization problems, which can significantly impact the overall system performance. To comprehensively evaluate the effect of frequency offset on signal estimation performance, we generated test data with frequency offsets of ± 100 Hz, ± 10 kHz, and ± 1 MHz. For small frequency offsets (± 100 Hz), both MSE and MAE values remain very low across the entire SNR range, demonstrating excellent robustness and stability. This indicates that the method has strong resilience against minor frequency

offsets, maintaining superior performance under various SNR conditions. As shown in Figure 11, the error metrics for small frequency offsets are consistently low, highlighting the method's robustness. Even under larger frequency offsets (± 1 MHz), MAE and MSE only show slight increases at low SNR levels but quickly return to low levels as SNR increases. This suggests that the method maintains high estimation accuracy and stability even in the presence of significant frequency offsets, showcasing its adaptability. The figures also illustrate this trend, where the performance degradation is minimal at higher SNR levels.

Overall, the performance differences between small and large frequency offsets are minimal, especially at high SNR levels where error metrics under different frequency offsets are nearly identical. This demonstrates that the proposed method is highly robust to frequency offsets, providing consistent and reliable estimation results across varying offset conditions. As illustrated in Figure 11a,b, the curves for different frequency offsets converge at higher SNR levels, confirming the method's stability. In summary, these test results indicate that the method not only exhibits strong resilience against minor frequency offsets but also performs excellently under significant frequency offsets. This finding provides valuable insights for optimizing practical communication systems, ensuring they maintain high performance and reliability in complex and dynamic environments. As shown in the figures, the method's robustness and adaptability are clearly demonstrated, making it suitable for a wide range of applications.

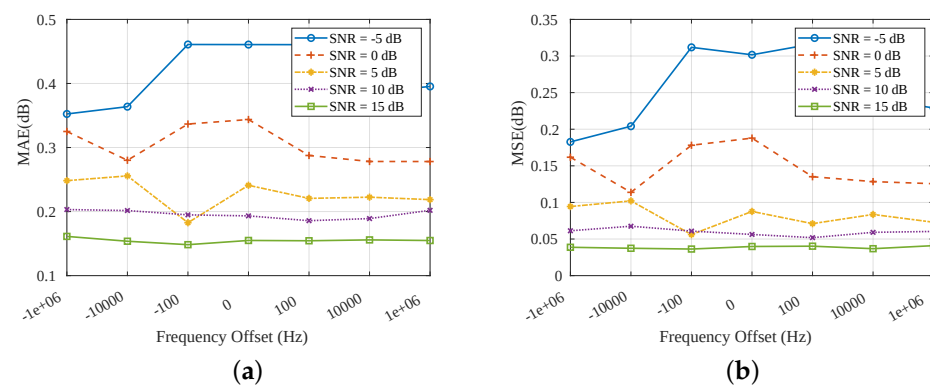


Figure 11. Performance of different frequency offsets: (a) MAE; (b) MSE.

5.8. Performance Evaluation with Varying Phase Offsets

To comprehensively evaluate the impact of phase offset on signal estimation performance, we simulated and generated test data under various phase-offset conditions. According to the specific analysis based on Figure 12, under low-SNR conditions, different phase offsets have a more significant effect on MSE and MAE. Especially when the phase offset is larger, the error noticeably increases. However, as SNR improves, the errors rapidly decrease and stabilize, indicating that the system not only demonstrates good robustness and adaptability under low SNR but also that the impact of different phase offsets on the error becomes negligible in high-SNR environments. This result further verifies the stability and reliability of this method in the face of varying phase offsets. Regardless of whether it is under low- or high-SNR conditions, the performance of the system proves its ability to provide consistent and reliable estimation results amidst changes in phase offset, thereby ensuring the high performance and stability of communication systems in practical applications.

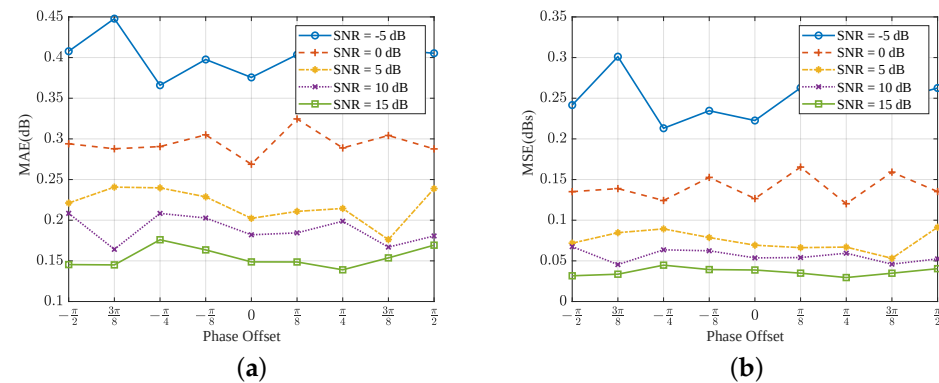


Figure 12. Performance evaluation with varying phase offsets: (a) MAE; (b) MSE.

6. Conclusions

This paper proposes a novel deep learning framework aimed at enhancing the accuracy and robustness of signal-to-noise ratio (SNR) estimation. The framework segments the received signals into 1024 sampling points and converts them into time–frequency matrices as feature inputs, which can effectively capture both the temporal and spectral characteristics of the signals. The experimental results confirm that, compared with the existing APG network and LRCN network, the proposed network achieves lower mean absolute error (MAE) and mean squared error (MSE), and its estimation performance approaches the theoretical limit. Specifically, robustness tests show that the method remains effective across unseen modulation formats (16QAM, GMSK, and DQPSK) and significant frequency offsets up to ± 1 MHz, with minimal performance degradation in high-SNR scenarios. The numerical validation results demonstrate its high estimation precision, achieving an MSE of approximately 0.028 at a signal-to-noise ratio of 15 dB with samples per symbol (SPS) = 19. The aforementioned studies indicate that the proposed model not only exhibits superior generalization ability and robustness but also maintains high performance and stability in various complex and harsh environments.

However, this study still has certain limitations. First, compared with other networks, the framework has a more complex structure, which may increase the implementation difficulty and computational cost. Second, the estimation performance is suboptimal under low-SPS conditions, which limits its applicability in some application scenarios.

In summary, this research validates the efficiency and stability of the novel framework in complex environments and confirms its broad applicability under various signal conditions. In response to the research limitations and the requirements proposed in the reviewer comments, future work will focus on the following aspects: first, optimizing the model architecture to reduce complexity and computational overhead, thereby promoting its practical implementation in edge computing scenarios and solving the practical problem of limited computing power of edge devices; second, in-depth exploration and identification of the motivating application scenarios of the framework in real-world systems, and verifying its application value through real-scenario tests; third, targeted improvement of the estimation performance under low-SPS conditions, breaking through the existing application limitations and further enhancing the practicality and versatility of the model.

Author Contributions: All authors contributed significantly to the research and development of this manuscript. L.L.: conceptualization, methodology, software, data curation, writing—original draft, visualization. W.X.: supervision, project administration. D.H.: software, data curation, visualization and validation. J.N.: formal analysis, resources, data curation. F.X.: writing—review and editing, funding acquisition. Z.H.: formal analysis, investigation. Y.Z.: visualization and validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Youth Foundation of China under Grant No. 62201598.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hu, D.; Zhao, Y.; Xie, W.; Xiao, Q.; Li, L. A squeeze-and-excitation network for SNR estimation of communication signals. *IET Commun.* **2025**, *19*, e70006. [CrossRef]
2. Li, S.; Zhou, J.; Huang, Z. TS-WHT: A Two-Step Walsh–Hadamard Transform Approach for Blind Error Correcting Code Classification. *IEEE Commun. Lett.* **2023**, *27*, 1689–1693. [CrossRef]
3. Qu, Y.; Lu, Z.; Hui, B.; Wang, J.; Wang, J. Contrastive Language-Signal Prediction for Automatic Modulation Recognition. *IEEE Wirel. Commun. Lett.* **2024**, *13*, 3242–3246. [CrossRef]
4. Al Mahbub, A.; Ahmad, I.; Shin, S. Comprehensive Analysis of CNN-Based Models for SNR Estimation. *IEEE Access* **2025**, *13*, 213346–213361. [CrossRef]
5. Chache, F.M.; Maxon, S.; Narayanan, R.M.; Bharadwaj, R. Effects on Age of Information From Forward Error Correction in LoRa Transmissions. *IEEE Sens. J.* **2024**, *24*, 30853–30862. [CrossRef]
6. Wang, M.; Zhang, Z.; Zhang, H.; Ghassemlooy, Z.; Zhang, T. Symmetry of constellation diagram-based intelligent SNR estimation for visible light communications. *Opt. Lett.* **2024**, *49*, 3138–3141. [CrossRef]
7. Gappmair, W. Data-Aided SNR Estimation for Bandlimited Optical Intensity Channels. *Sensors* **2022**, *22*, 8660. [CrossRef]
8. Xue, R.; Cao, Y.; Wang, T. Data-Aided and Non-Data-Aided SNR Estimators for CPM Signals in Ka-Band Satellite Communications. *Information* **2017**, *8*, 75. [CrossRef]
9. Li, C.; Zhang, T.; Liu, S. MVSC: Mamba Vision Based Semantic Communication for Image Transmission With SNR Estimation. *IEEE Commun. Lett.* **2025**, *29*, 1406–1410. [CrossRef]
10. Zhao, Y.; Ju, C.; Wang, D.; Liu, N.; Guan, L.; Xie, P. SNR Estimation for 5G Satellite-to-Ground Communication in Ultra-Low SNR Environment Based on Channel Frequency Response Reconstruction. *IEEE Commun. Lett.* **2024**, *28*, 357–361. [CrossRef]
11. Yadav, A.; Kumar, A.; Shitiri, E.; Kumar, S.; Cho, H.S. Non-Data-Aided SNR Estimation for Molecular Communication Systems in Internet of Bio-Nano Things. *IEEE Internet Things J.* **2025**, *12*, 595–604. [CrossRef]
12. Wang, W.; Shen, Y.; Wang, Y. Low-Complexity Non-Data-Aided SNR Estimation for Multilevel Constellations. *IEEE Commun. Lett.* **2020**, *24*, 113–116. [CrossRef]
13. Pauluzzi, D.; Beaulieu, N. A Comparison of SNR Estimation Techniques for the AWGN Channel. *IEEE Trans. Commun.* **2000**, *48*, 1681–1691. [CrossRef]
14. Lee, J.; Lou, H.I.; Toupakaris, D.; Cioffi, J.M. SNR Analysis of OFDM Systems in the Presence of Carrier Frequency Offset for Fading Channels. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 3360–3364. [CrossRef]
15. Li, G.; Liu, F.; Yang, H. A New Detection Model of Ship-Radiated Noise Signal. *Ocean. Eng.* **2024**, *297*, 117081. [CrossRef]
16. Lv, M.; Yan, X.; Wang, K.; Hao, X.; Dai, J. Adaptive Measurement and Parameter Estimation for Low-SNR PRBC-PAM Signal Based on Adjusting Zero Value and Chaotic State Ratio. *Mathematics* **2024**, *12*, 3203. [CrossRef]
17. Zheng, S.; Chen, S.; Chen, T.; Yang, Z.; Zhao, Z.; Yang, X. Deep Learning-Based SNR Estimation. *IEEE Open J. Commun. Soc.* **2024**, *5*, 4778–4796. [CrossRef]
18. Simon, M.K.; Mileant, A. SNR Estimation for the Baseband Assembly. The Telecommunications and Data Acquisition Report 1986. Available online: <https://ntrs.nasa.gov/citations/19860018814> (accessed on 1 January 2024).
19. Kerr, R.B. On Signal and Noise Level Estimation in a Coherent PCM Channel. *IEEE Trans. Aerosp. Electron. Syst.* **1966**, *AES-2*, 450–454. [CrossRef]
20. Athanasios, D.; Kalivas, G. SNR Estimation for Low Bit Rate OFDM Systems in AWGN Channel. In *International Conference on Systems & International Conference on International Conference on Networking*; IEEE: Piscataway, NJ, USA, 2006.
21. Matzner, R.; Englberger, F. An SNR estimation algorithm using fourth-order moments. In *1994 IEEE International Symposium on Information Theory*; IEEE: Piscataway, NJ, USA, 1994; p. 119. [CrossRef]
22. Ghose, A.; Yang, N. In-Service Monitoring of Multipath Delay and Cochannel Interference for Indoor Mobile Communication Systems. In *Serving Humanity Through Communications, Vol.2: 1994 International Conference on Communications (ICC'94)*; IEEE (Institute of Electrical and Electronics Engineers): Piscataway, NJ, USA, 1994.

23. Gappmair, W. Cramer-Rao Lower Bound for Non-Data-Aided SNR Estimation of Linear Modulation Schemes. *IEEE Trans. Commun.* **2008**, *56*, 689–693. Correction in *IEEE Trans. Commun.* **2010**, *58*, 318–318. <https://doi.org/10.1109/TCOMM.2010.5397926>. [CrossRef]
24. López-Valcarce, R.; Villares, J.; Riba, J.; Gappmair, W.; Mosquera, C. Cramér-Rao Bounds for SNR Estimation of Oversampled Linearly Modulated Signals. *IEEE Trans. Signal Process.* **2015**, *63*, 1675–1683. [CrossRef]
25. Sevillano, J.F.; Velez, I.; Leyh, M.; Lipp, S.; Irizar, A.; Fontan, F. In-Service SNR Estimation without Symbol Timing Recovery for QPSK Data Transmission Systems. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 3202–3207. [CrossRef]
26. Rice, M. Data-Aided and Non-Data-Aided Maximum Likelihood SNR Estimators for CPM. *IEEE Trans. Commun.* **2015**, *63*, 4244–4253. [CrossRef]
27. Bellili, F.; Meftehi, R.; Affes, S.; Stéphenne, A. Maximum Likelihood SNR Estimation of Linearly-Modulated Signals Over Time-Varying Flat-Fading SIMO Channels. *IEEE Trans. Signal Process.* **2015**, *63*, 441–456. [CrossRef]
28. Rugini, L.; Banelli, P. On the Equivalence of Maximum SNR and MMSE Estimation: Applications to Additive Non-Gaussian Channels and Quantized Observations. *IEEE Trans. Signal Process.* **2016**, *64*, 6190–6199. [CrossRef]
29. Yang, K.; Huang, Z.; Wang, X.; Wang, F. An SNR Estimation Technique Based on Deep Learning. *Electronics* **2019**, *8*, 1139. [CrossRef]
30. Xie, X.; Peng, S.; Yang, X. Deep Learning-Based Signal-To-Noise Ratio Estimation Using Constellation Diagrams. *Mob. Inf. Syst.* **2020**, *2020*, 8840340. [CrossRef]
31. Xu, X.; Feng, C. A Long-Term Recurrent Convolutional Network for SNR Estimation of Cone-Shaped Target. *IEEE Antennas Wirel. Propag. Lett.* **2023**, *22*, 1863–1867. [CrossRef]
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778. [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.