

# 目录

CONTENTS

01

研究背景

02

理论基础

03

网络模型搭建

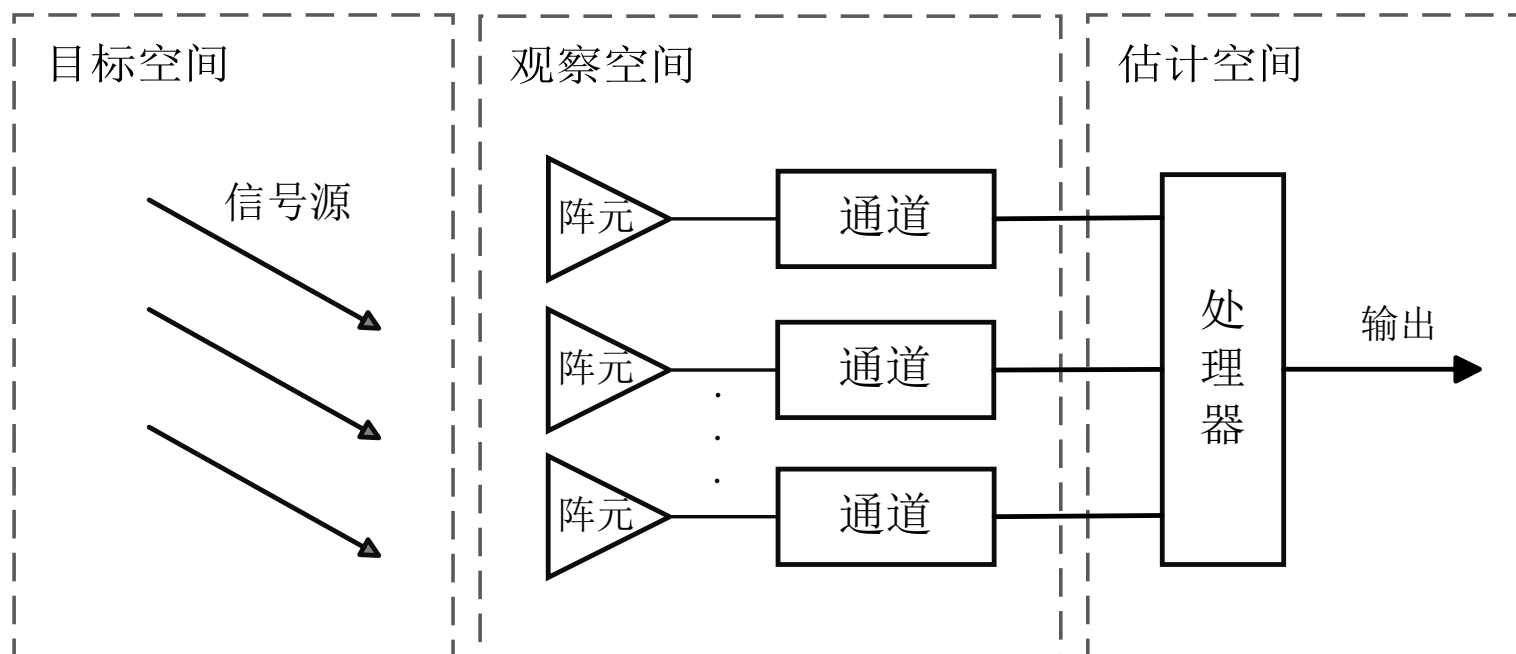
04

仿真实验分析



# 01 研究背景

西安理工大学



# 1. 研究背景

## 传统的空间谱估计

基于模型驱动、需提前建模

模型失配  
动态环境适应差

## 基于深度学习的空间谱估计

基于数据驱动、自动学习

性能可靠  
模型适应性强

# 02 理论基础

分类型神经网络  
+  
空间谱估计

2.1

阵列模型

2.2

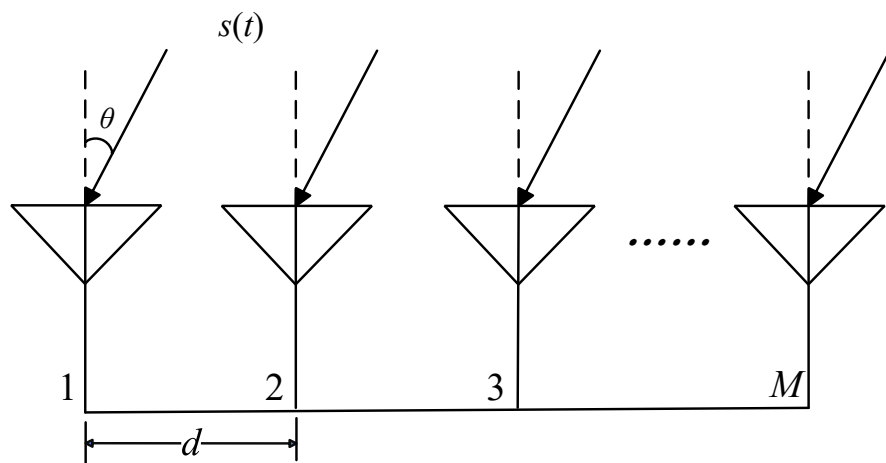
神经网络

2.3

网络的输入与输出

## 2.1 阵列模型

线性阵列:



\$K\$ 个入射信号  
\$M\$ 个阵元

阵列接收信号:  $\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t) + \mathbf{N}(t)$

$$= \begin{bmatrix} e^{-j\omega_0\tau_{1,1}} & e^{-j\omega_0\tau_{1,2}} & \dots & e^{-j\omega_0\tau_{1,K}} \\ e^{-j\omega_0\tau_{2,1}} & e^{-j\omega_0\tau_{2,2}} & \dots & e^{-j\omega_0\tau_{2,K}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j\omega_0\tau_{M,1}} & e^{-j\omega_0\tau_{M,2}} & \dots & e^{-j\omega_0\tau_{M,K}} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_K(t) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_M(t) \end{bmatrix}$$

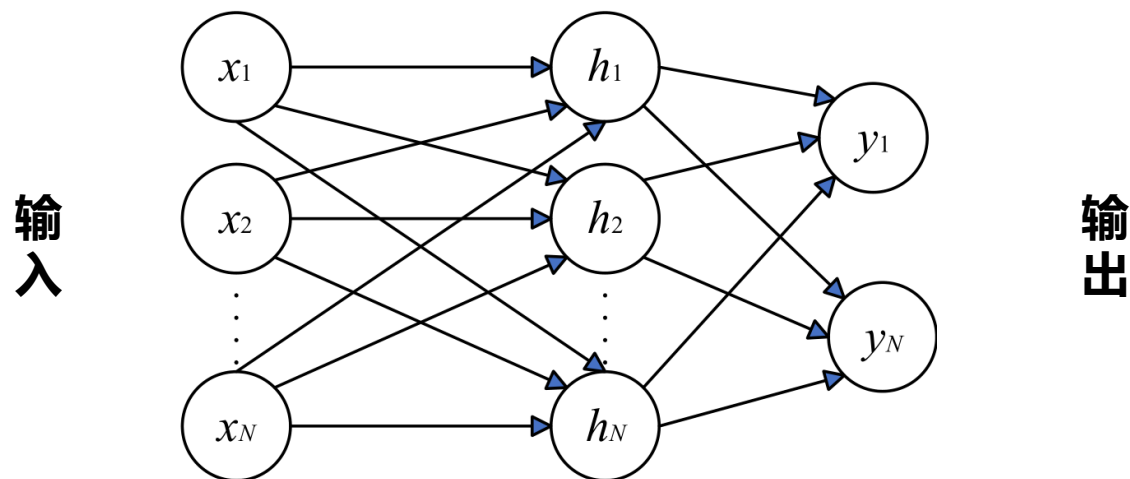


协方差矩阵

$$\mathbf{R} \approx \hat{\mathbf{R}} = \frac{1}{L} \sum_{i=1}^L \mathbf{X}\mathbf{X}^H$$

(包含信号空间谱信息)

## 2.2 神经网络



网络输出, 激活函数+ 前向传播 → 网络输出 → 损失函数

梯度下降, 反向传播 → 调整网络参数  $w$ 、 $b$

重复更新参数  
得到最优模型

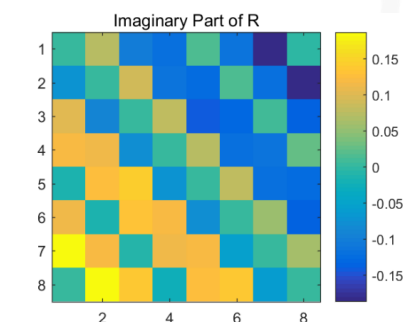
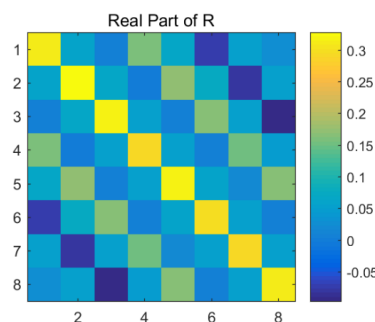
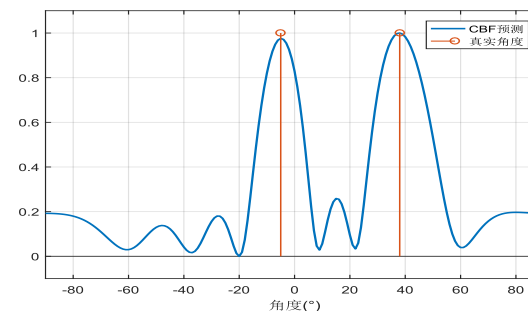
$$y = f(x; w; b) \quad (\text{输入与输出映射关系})$$

## 2.3 确定输入与输出

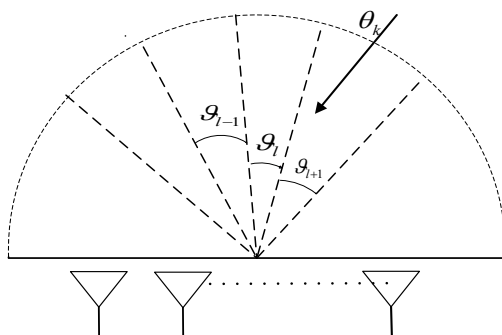
两种网络输入特征

一维输入：阵列接收信号→CBF算法  
181维向量

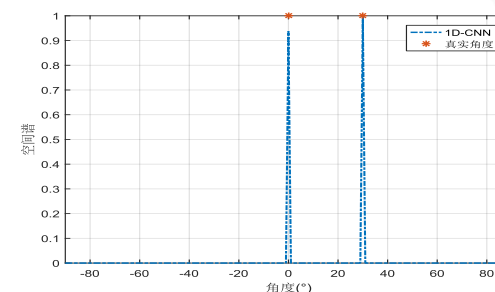
二维输入：协方差矩阵  
 $M \times M$  ( $8 \times 8$ )



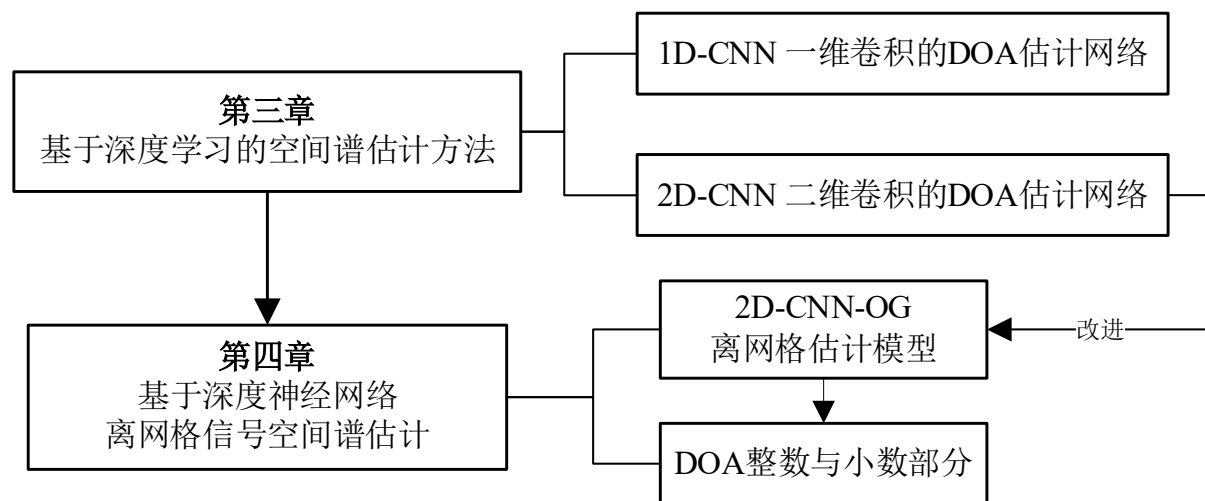
角度量化为181格



网络输出



# 03 网络模型



3.1

模型1D-CNN

3.2

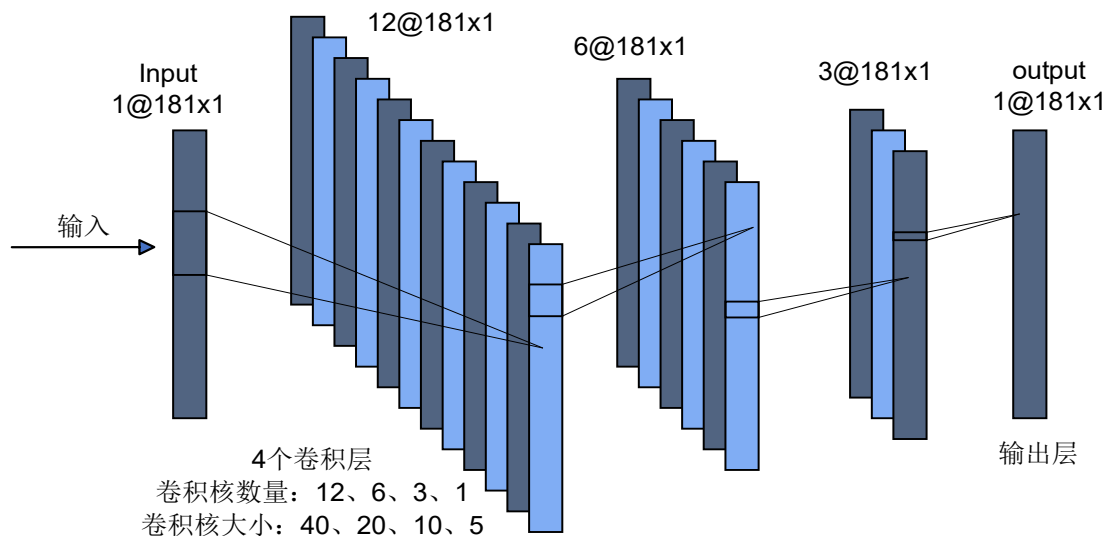
模型2D-CNN

3.3

2D-CNN-OG



## 3.1 模型1D-CNN



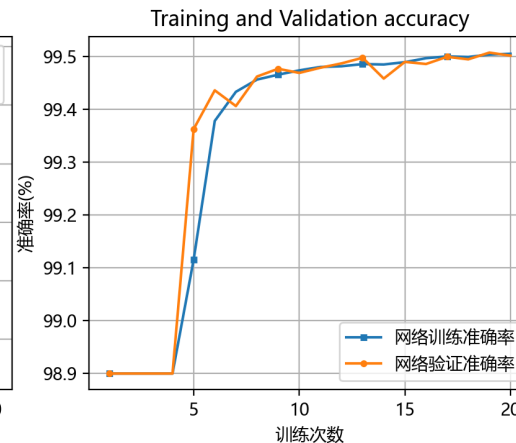
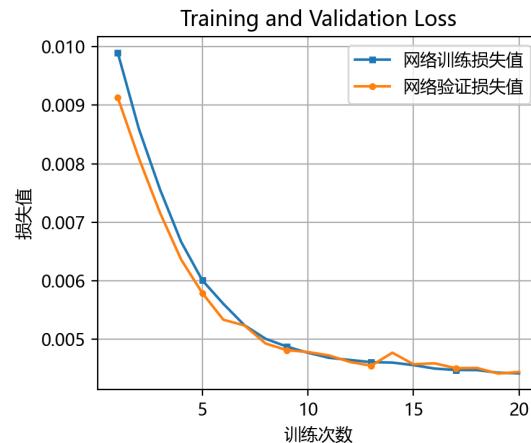
激活函数: ReLU

损失函数: MSE (均方误差)

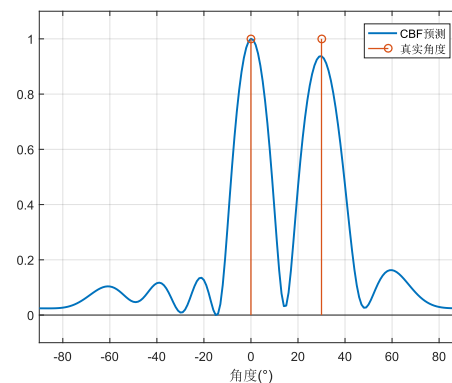
优化算法: Adam算法

训练次数: 20次

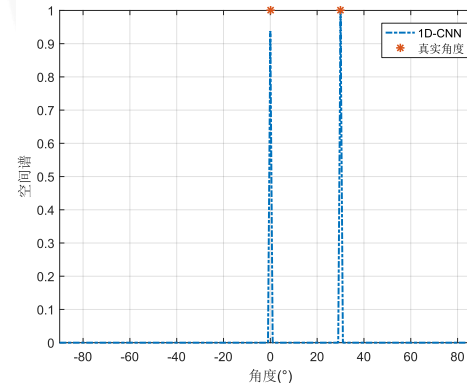
### 模型1D-CNN训练表现:



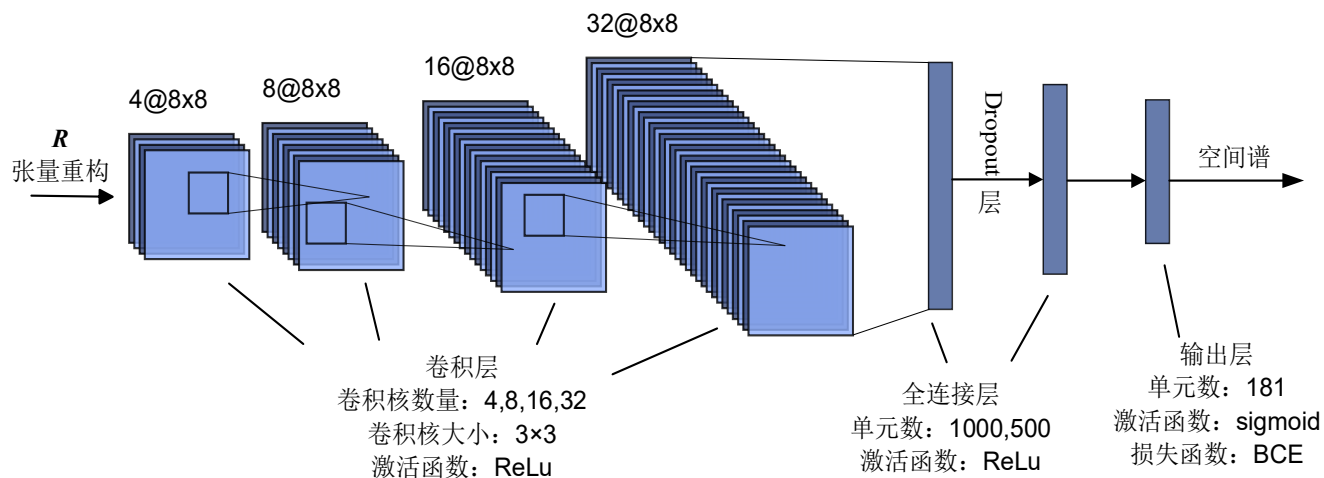
输入:



输出:



## 3.2 模型2D-CNN



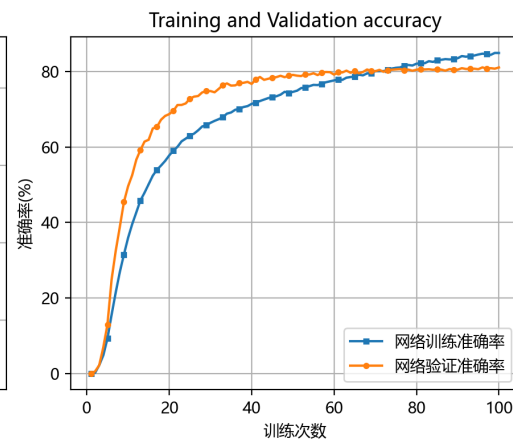
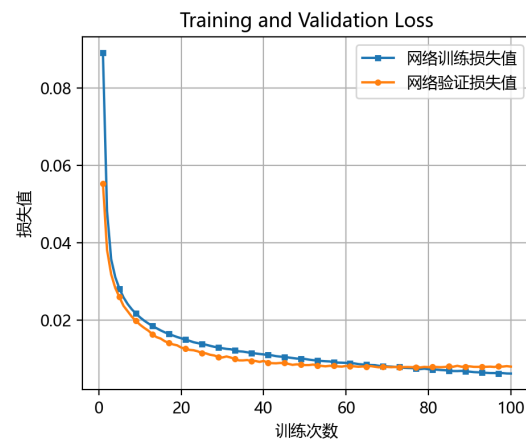
激活函数: ReLU (前六层), sigmoid (最后一层)

损失函数: BCE (二元交叉熵)

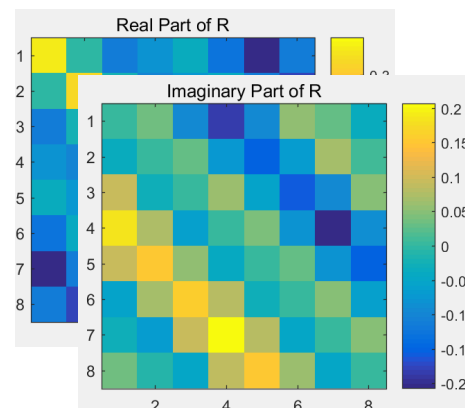
优化算法: Adam算法

训练次数: 100次

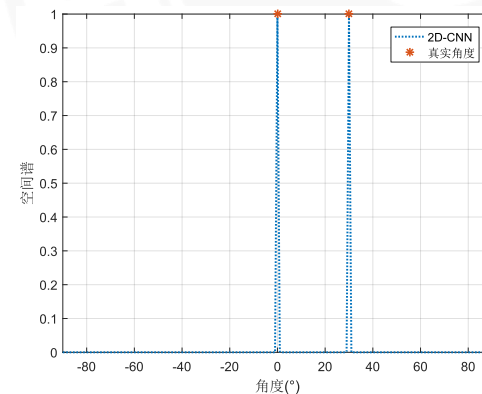
### 模型2D-CNN训练表现:



输入:

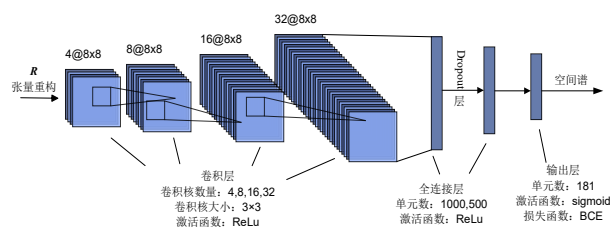


输出:



## 3.3 2D-CNN-OG

基于2D-CNN模型



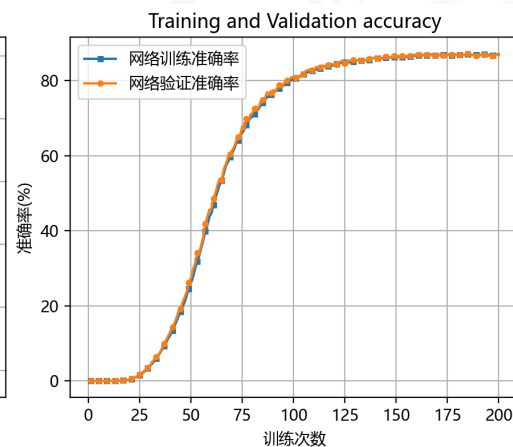
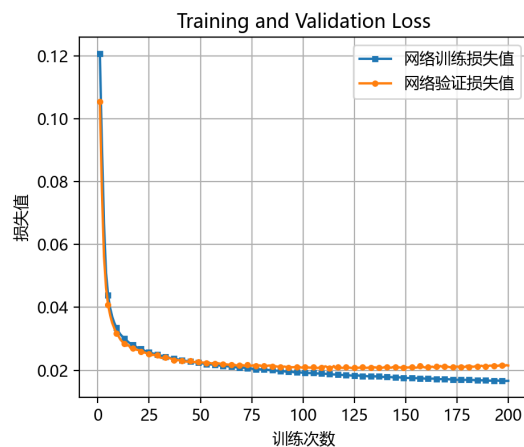
小数标签

$$p_i = \begin{cases} \theta_k - \lceil \theta_k \rceil, & \text{若 } \exists \theta_k, \text{ 且 } \lceil \theta_k \rceil = \Theta_i \\ -1, & \text{其他} \end{cases}$$

有信号处  $[-0.5, 0.5]$

无信号处 -1

模型2D-CNN-OG训练表现:



激活函数: ReLU (前四层), sigmoid (后两层)

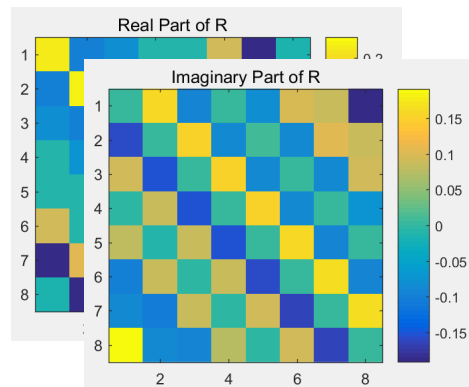
最后输出再经过 tanh函数激活  $s = \tanh(z)$

损失函数: 改进BCE:  $L(z, p) = -\frac{1}{N} \sum_{i=1}^N \left[ \mathbf{1}_{[p_i \neq 1]} \ln\left(\frac{1}{1 + e^{g(z_i)}}\right) + \mathbf{1}_{[p_i = 1]} \ln\left(\frac{e^{-z_i}}{1 - e^{-z_i}}\right) \right]$

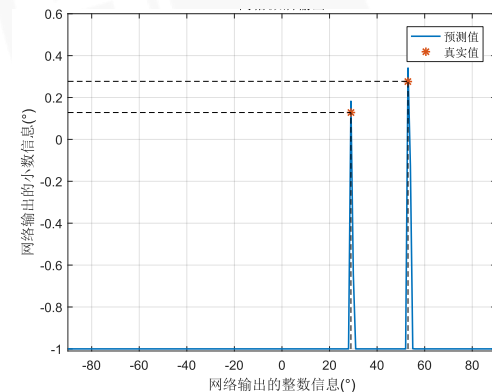
正则化函数:  $g(z_i) = \left| \frac{z_i + \varepsilon}{p_i + \varepsilon} - 1 \right|, p_i \neq -1$

训练次数: 200次

输入:



输出:





# 04 仿真实验分析

**4.1**

**仿真测向结果**

**4.2**

**网络估计误差分析**

**4.3**

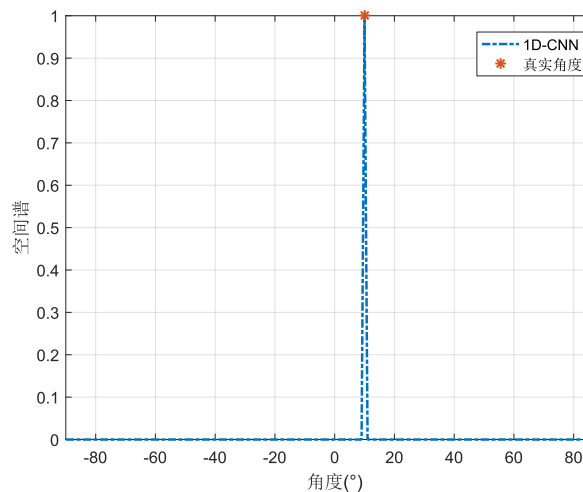
**综合性能分析**

## 4.1 仿真测向结果

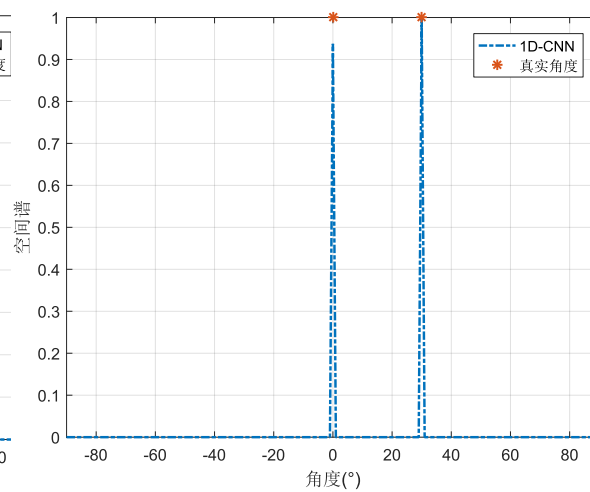
### 1D-CNN

信号个数泛化能力:

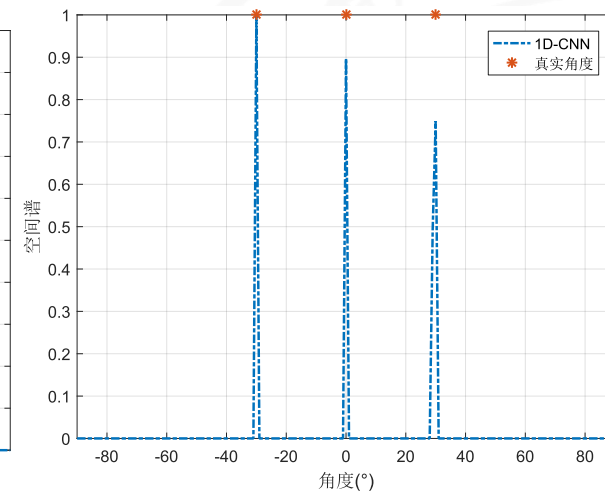
单信号 $10^\circ$



两信号  $0^\circ$  和  $30^\circ$

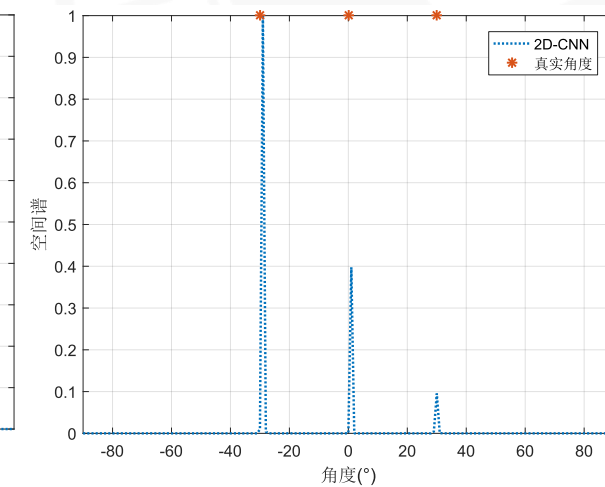
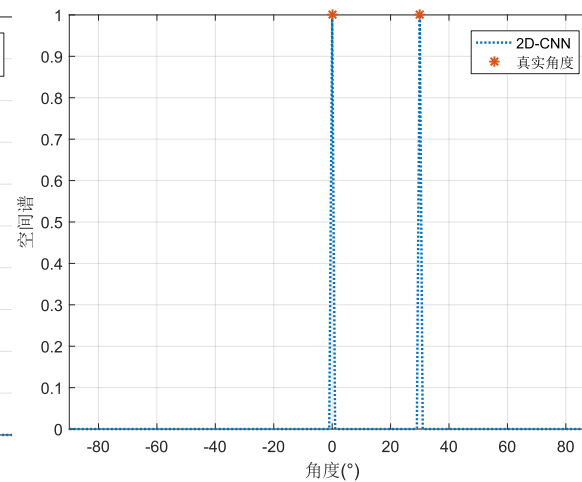
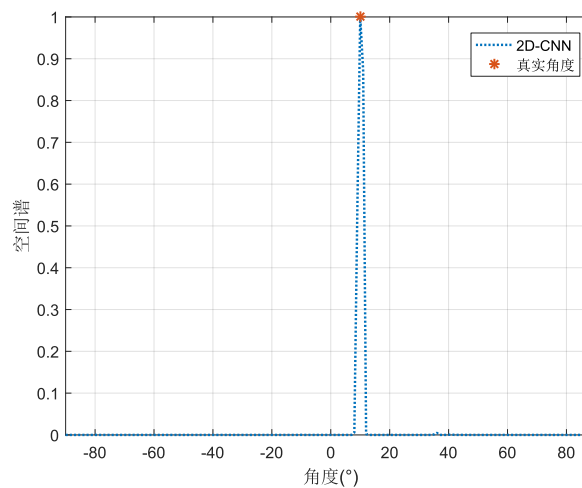


三信号  $-30^\circ$ 、 $0^\circ$  和  $30^\circ$



### 2D-CNN

信号个数泛化能力:

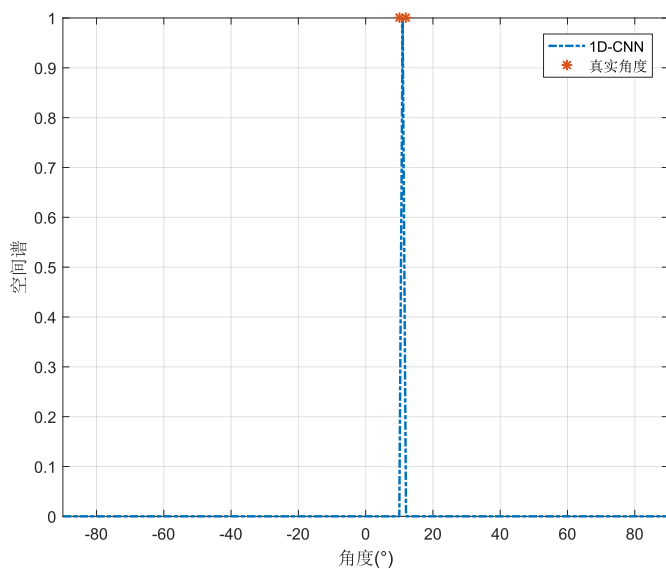


## 4.1 仿真测向结果

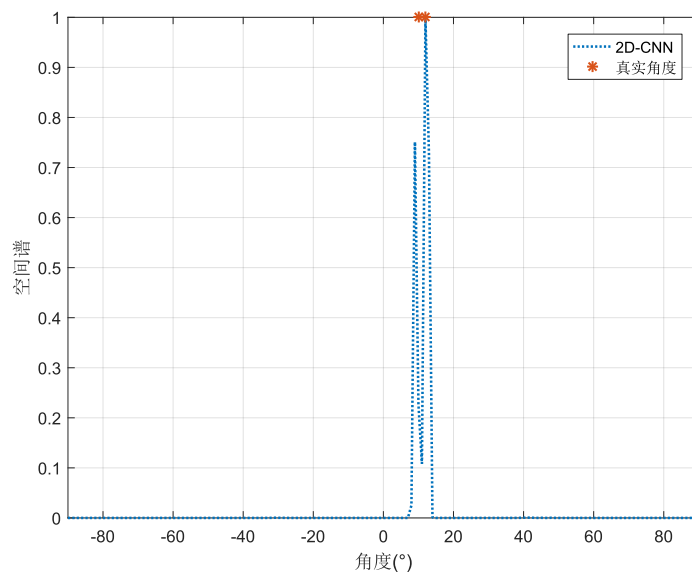
**模型的超分辨能力分析：**测试样本为两远场窄带信号

真实信号角度间隔为2度，均信噪比为0dB

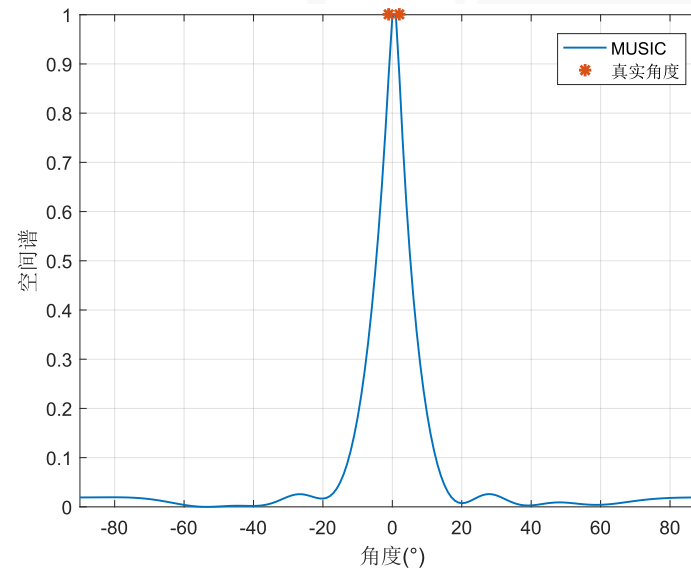
可见2D-CNN的分辨能力最好



(a) 1D-CNN分辨图



(b) 2D-CNN分辨图

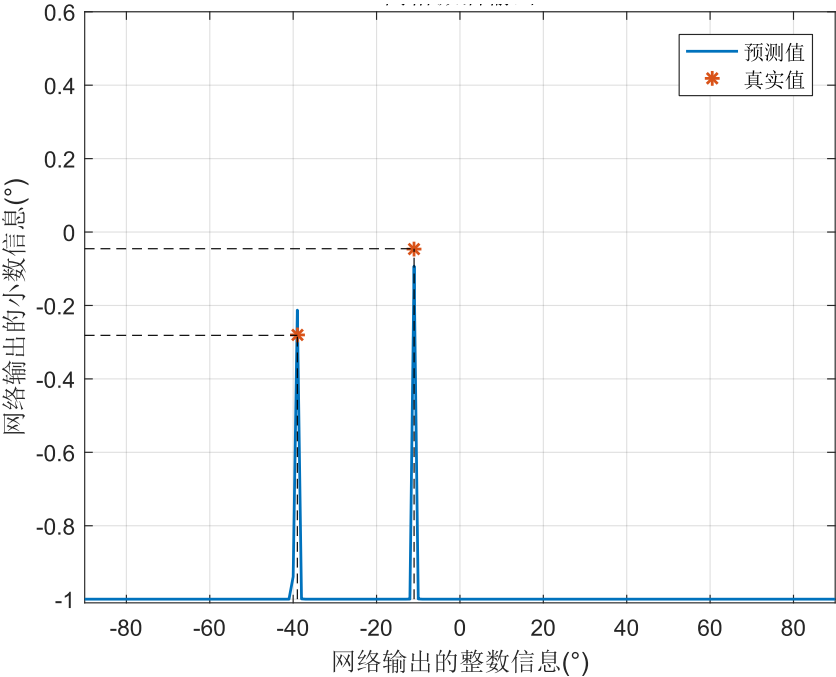


(c) MUSIC分辨图

# 4.1 仿真测向结果

2D-CNN-OG模型离网输出谱:

小数  
(回归任务)



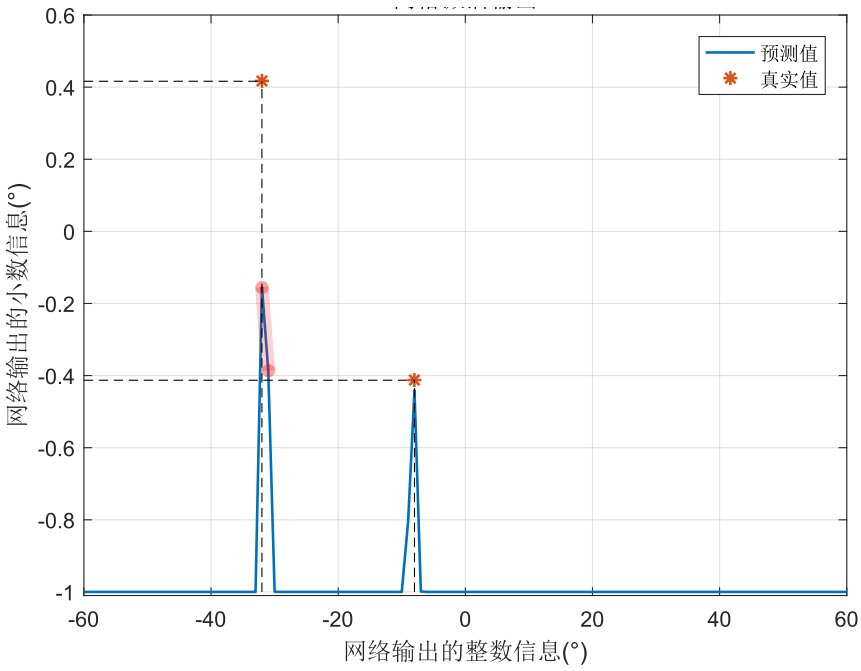
整数  
(分类任务)

整数信息 + 小数信息  
即为信号的入射角度估计值，估计结果见表

DOA 信息↵	信号 1↵	信号 2↵	↵
信号的真实 DOA↵	-39.2814° ↵	-11.0455° ↵	↵
2D-CNN 输出 DOA↵	-39.2126° ↵	-11.0928° ↵	↵

# 4.1 仿真测向结果

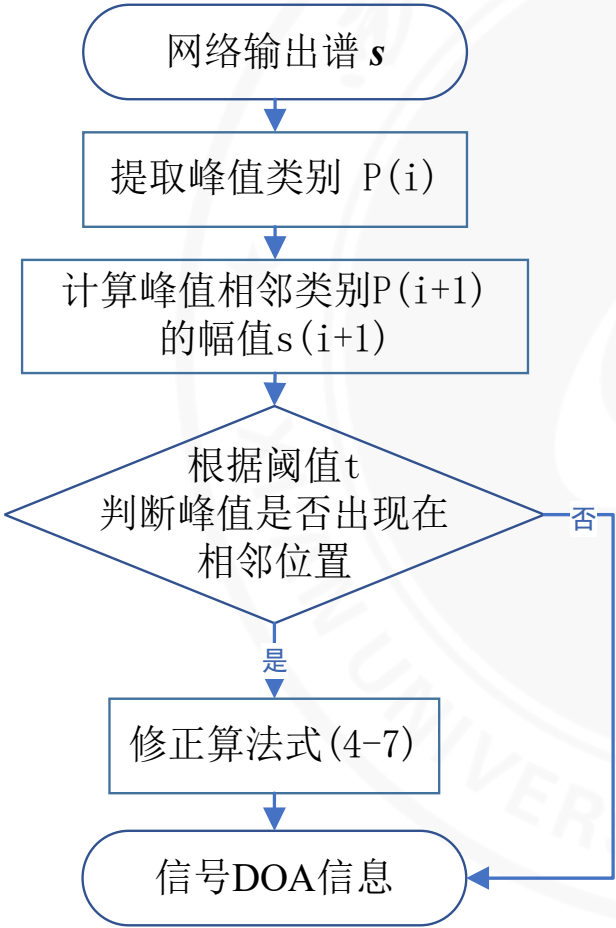
针对2D-CNN-OG离网格输出分类错误:



加权平均算法修正式:

$$\hat{\theta}_i = \frac{(s_i + 1)P_i + (s_{i+1} + 1)P_{i+1}}{s_i + s_{i+1} + 2}$$

误差自动修正程序:



DOA 信息↵	信号 1↵	信号 2↵	↵
信号的真实 DOA↵	-31.5837° ↵	-8.4129° ↵	↵
2D-CNN-OG 输出 DOA↵	-32.1545° ↵	-8.4403° ↵	↵
自动修正后 DOA↵	-31.5796° ↵	-8.4403° ↵	↵



## 4.2 网络误差分析

-60 °~ 60 °全角度测向与误差分析

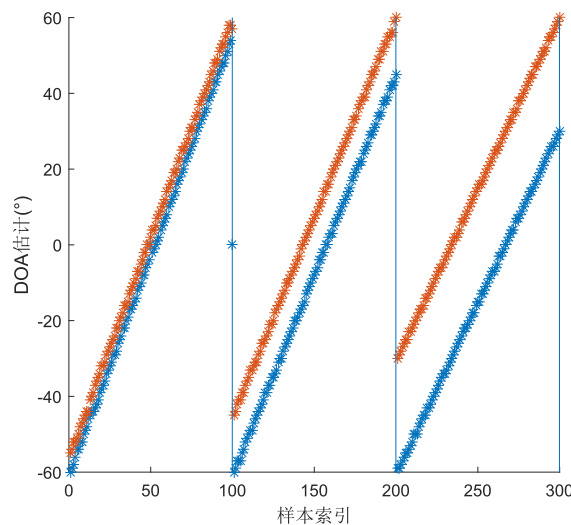
样本都为两信号输入

分3组，每组角度间隔为5.5 °、15.6 °、30.7 °

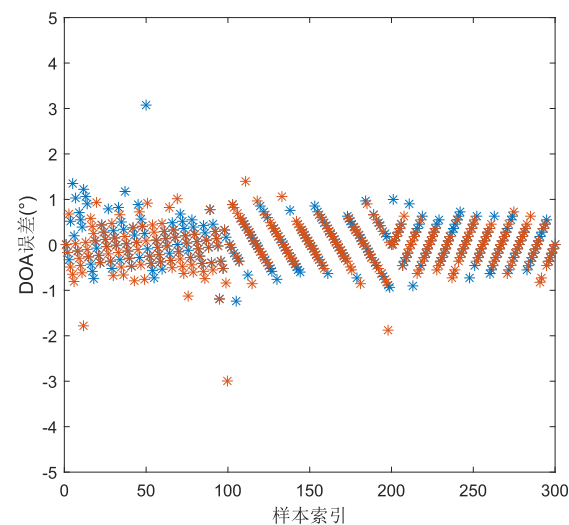
信噪比都为0dB

2D-CNN输出误差集中在-1°~1°之间

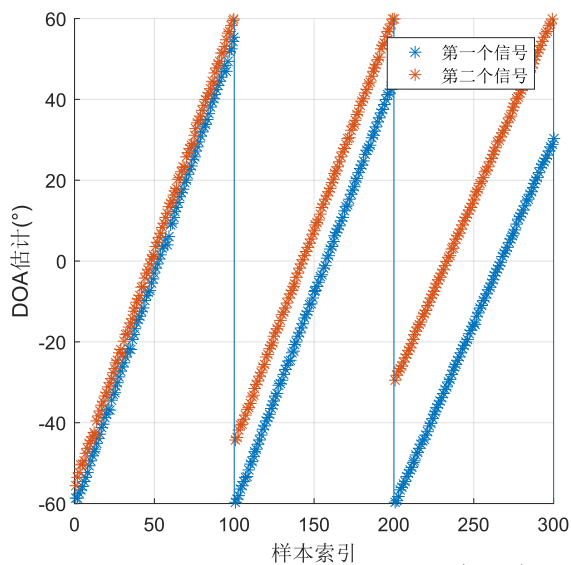
2D-CNN-OG输出误差集中在-0.5°~0.5°之间



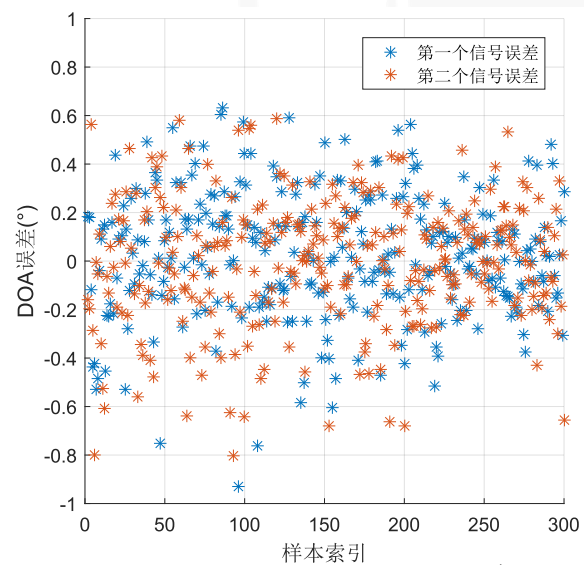
(a) 2D-CNN 测向结果



(b) 2D-CNN测向误差

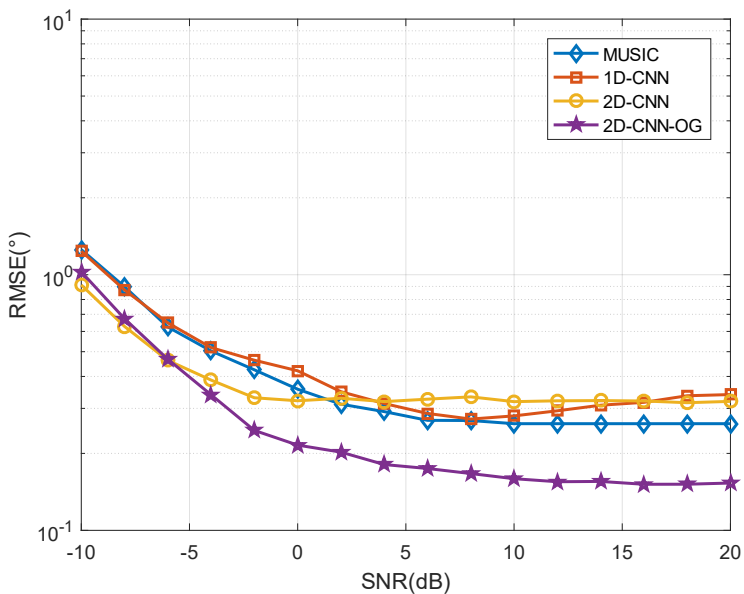


(c) 2D-CNN-OG 测向结果

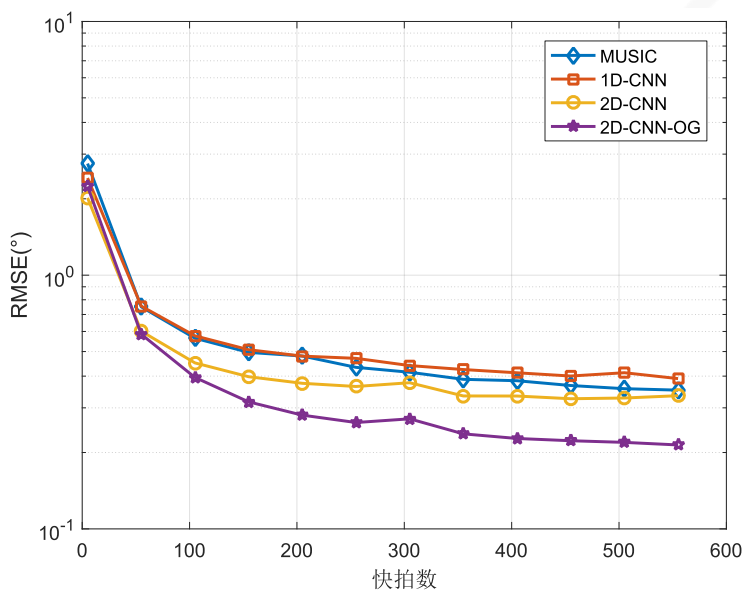


(d) 2D-CNN-OG测向误差

## 4.3 综合性能分析



(a) 不同信噪比对DOA估计性能的影响



(b) 不同快拍数对DOA估计性能的影响

16×300次  
测试实验

估计方法↵	MUSIC 算法↵	1D-CNN ↵	2D-CNN↵	2D-CNN-OG↵
模型训练时间↵	↵	249.83 s↵	2119.81 s↵	4278.12 s↵
总测试时间↵	19.02 s↵	450.44 s↵	456.06 s↵	454.36 s↵
平均单次时间↵	3.96 ms↵	93.8 ms↵	95.01 ms↵	94.65 ms↵