**Supplementary materials of SI-CNN**

## A. Proof Process of Differences in Signal Quality

To illustrate the Differences in Signal Quality (DSQ) phenomenon, we analyzed the results of the CNN model[1] on simulated data with a signal-to-noise ratio (SNR) of 15. The analysis focused on three types of signals: the clean signal $Clean_{sn}$, the noisy signal $Noisy_{sn}$, and the predicted signal $Pred_{sn}$. Based on these signals, three metrics were calculated to evaluate the quality of individual signals: $Noise$, $Fitting\ Error$, and $Reconstruction\ Error(Recon\ Error)$.

Because the closeness of a signal to $Clean_{sn}$ serves as an indicator of its quality, $Noise$ can be used to evaluate the quality of $Noisy_{sn}$. $Recon\ Error$ can be used to evaluate the quality of $Pred_{sn}$. It should be noted that smaller values of $Noise$ and $Recon\ Error$ indicate that the corresponding $Noisy_{sn}$ or $Pred_{sn}$ is closer to $Clean_{sn}$, and therefore has higher quality.

These metrics can be calculated as follows:

$$Noise = \sqrt{(Noisy_{sn} - Clean_{sn})^2} \tag{1}$$

$$FittingError = \sqrt{(Pred_{sn} - Noisy_{sn})^2} \tag{2}$$

$$ReconError = \sqrt{(Pred_{sn} - Clean_{sn})^2} \tag{3}$$

In the CNN[1] simulation experiment, we calculated the mean Noise and mean Recon Error of all signals and plotted their variations with training epochs, as shown in Fig. 1. Noise refers to the difference between the original noisy signals and the clean signals; its mean value remains constant across epochs and therefore appears as a horizontal line in the figure. Recon Error, in contrast, denotes the difference between the predicted signals and the clean signals. As the number of epochs increases, the predicted signals are continuously updated. Therefore, the mean Recon Error varies with the Epoch.

At epoch 1, the mean Recon Error of the model is higher than the mean Noise. Because smaller values of Noise and Recon Error indicate that the corresponding $Pred_{sn}$ and $Noisy_{sn}$ are closer to $Clean_{sn}$, reflecting higher overall signal quality. Therefore, at epoch 1, the overall quality of $Pred_{sn}$ is lower than that of $Noisy_{sn}$. This can be attributed to the fact that the model is still in the early training stage and has not yet effectively learned the signal features, resulting in a lower quality of $Pred_{sn}$. As training progresses, the mean Recon Error gradually decreases, indicating that the overall quality of $Pred_{sn}$ improves and eventually surpasses that of $Noisy_{sn}$ in later epochs. This phenomenon can be summarized as **Phenomenon 1**: As training progresses, the predicted signals progressively surpass the noisy signals in overall quality.

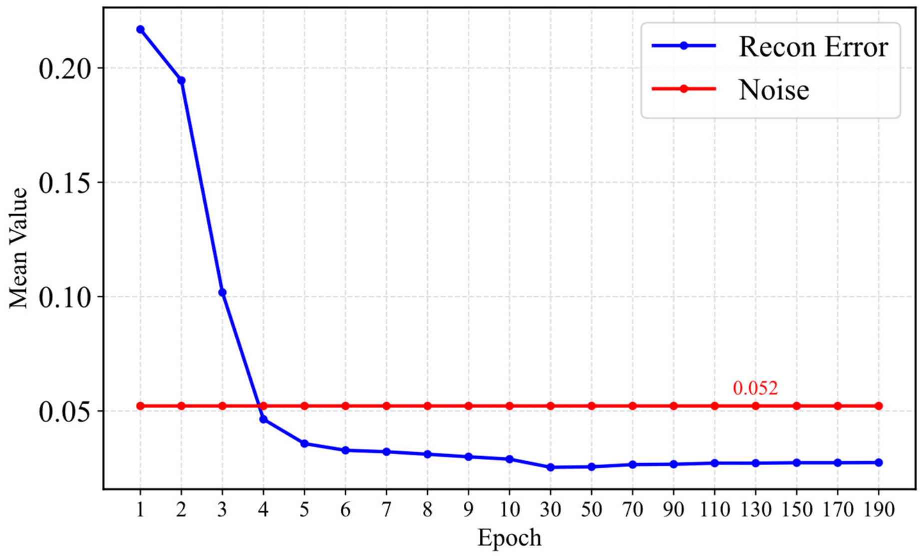



Figure 1: Mean Noise (red) and mean Recon Error(blue) of all signals across training epochs, computed according to Eq. 2 and Eq. 4. After epoch 10, Recon Error becomes stable. Therefore, metrics are reported every 20 epochs starting from epoch 10. Data are derived from the CNN[1] simulation experiment with SNR = 15.

To further investigate the signal characteristics within a single epoch, a refined analysis was conducted based on Fig. 1. Specifically, all signals within one epoch were grouped according to their Fitting Error, which was calculated as Eq. 3. It should be noted that while the overall mean Noise across all signals is fixed, the mean Noise within each subgroup varies due to differences in the sampled signals. To enhance the reliability of the conclusions, the relationships among Noise, Recon Error, and Fitting Error were examined across multiple epochs (30, 50, 100). The results are presented in Fig. 2.

From Fig. 2, another phenomenon can be observed: as the Fitting Error increases, the mean Noise increases significantly, while the mean Recon Error decreases slightly. This phenomenon can be observed for all epochs greater than 10. Because smaller values of Noise and Recon Error indicate signals closer to the clean signal, reflecting higher overall quality, this phenomenon can be summarized as **Phenomenon 2**: After extended training, as the Fitting Error increases, the overall quality gap between predicted signals and noisy signals increases.

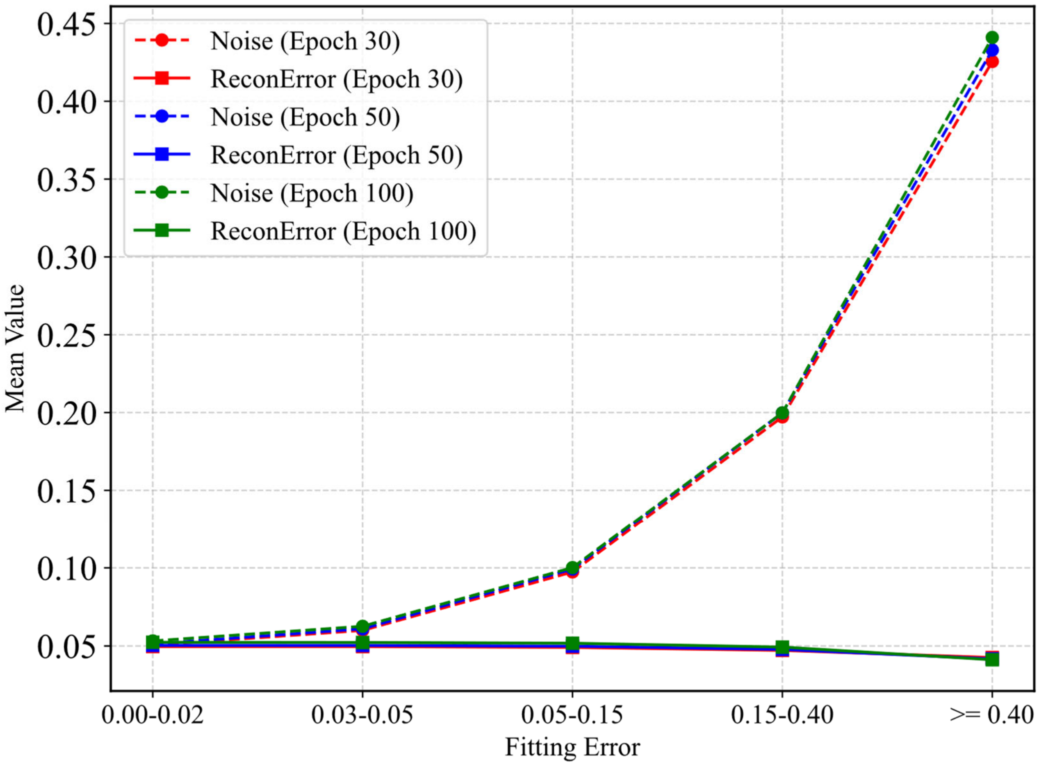



Figure 2: Mean Noise (dashed) and mean Recon Error (solid) of signals grouped by Fitting Error within a single epoch. Fitting Error intervals are left-closed, right-open. Data from epochs 30, 50, and 100 are shown in red, blue, and green curves, respectively. Data are derived from the CNN[1] simulation experiment with SNR = 15.

The verification process for Phenomena 1 and 2 is as follows:

Consider n clean signals across all voxels and all b-values, with the normalized clean signals denoted as $s_i$ and the noise introduced during acquisition denoted as $\epsilon_i$. The noisy signal is represented as $n_i = s_i + \epsilon_i$. The predicted signal obtained from an unsupervised model using $n_i$ as input is denoted as $p_i$, and the corresponding fitting error is defined as $f_i = p_i - n_i$.

To reflect the overall quality of the predicted signals, we define the metric $Q_p$ as follows:

$$Q_p = \frac{1}{n}\sum_{i=1}^{n}|p_i - s_i| \tag{4}$$

Since $Q_p$ represents the differences between the predicted signals $p_i$ and the clean signals $s_i$, a smaller value of $Q_p$ indicates that $p_i$ is closer to $s_i$, reflecting higher overall signal quality. Similarly, we define $Q_n$ to quantify the overall quality of the noisy signals $n_i$:

$$Q_n = \frac{1}{n}\sum_{i=1}^{n}|n_i - s_i| \tag{5}$$

From the definitions of $f_i$ and $n_i$, we have $p_i = n_i + f_i$ and $n_i - s_i = \epsilon_i$, which leads to:

$$|p_i - s_i| = |n_i + f_i - s_i| = |(n_i - s_i) + f_i| = |\epsilon_i + f_i| \tag{6}$$

$$Q_p = \frac{1}{n}\sum_{i=1}^{n}|p_i - s_i| = \frac{1}{n}\sum_{i=1}^{n}|\epsilon_i + f_i| \tag{7}$$

$$Q_n = \frac{1}{n}\sum_{i=1}^{n}|n_i - s_i| = \frac{1}{n}\sum_{i=1}^{n}|\epsilon_i| \tag{8}$$

The study by Zhang et al[2] demonstrated that, after extended training, the error in the loss function is primarily attributable to noise. To reduce the loss, the model tends to learn the noise $\epsilon_{\mathrm{i}}$ and suppress its influence, thereby driving the predicted signal closer to the clean signal. Since $p_i = n_i + f_i$, the fitting error $f_i$ can be regarded as the correction made by the model when processing the noisy signal $n_i$. Consequently, after sufficient training, for the vast majority of voxels and b-value signals, $f_i$ is oriented in the opposite direction of $\epsilon_{\mathrm{i}}$, with its magnitude being smaller than that of $\epsilon_{\mathrm{i}}$, thus:

$$|\epsilon_i + f_i| < |\epsilon_i| \tag{9}$$

Therefore:

$$|p_i - s_i| = |\epsilon_i + f_i| < |\epsilon_i| = |n_i - s_i| \tag{10}$$

On average over all signals, we have:

$$Q_p = \frac{1}{n}\sum_{i=1}^{n}|p_i - s_i| < \frac{1}{n}\sum_{i=1}^{n}|\epsilon_i| = Q_n \tag{11}$$

Smaller values of $Q_p$ and $Q_n$ indicate that the corresponding predicted signals $p_i$ and noisy signals $n_i$ exhibit higher overall quality. This finding strongly supports Phenomenon 1: As training progresses, the predicted signals progressively surpass the noisy signals in overall quality.

The quality metric of a single predicted signal is defined as $q_p = |p_i - s_i|$, while the quality metric of a single noisy signal is defined as $q_n = |n_i - s_i|$. The quality difference between the two is represented as $\Delta_i = q_p - q_n$.

Smaller values of $q_p$ and $q_n$ indicates that the corresponding predicted signal $p_i$ or noisy signal $n_i$ is closer to the clean signal $s_i$, thereby reflecting higher quality. Therefore, if $\Delta_i<0$, the predicted signal $p_i$ has higher quality than the noisy signal $n_i$; if $\Delta_i>0$, the predicted signal $p_i$ has lower quality than the noisy signal $n_i$, with larger values of $|\Delta_i|$ indicating a greater quality difference between $p_i$ and $n_i$.

Expanding $\Delta_i$ as:

$$\Delta_i = q_p - q_n = |p_i - s_i| - |n_i - s_i| \tag{12}$$

Substituting Eq. 11 into Eq. 13:

$$\Delta_i = |\epsilon_i + f_i| - |\epsilon_i| \tag{13}$$

Discussion by cases:

$$\Delta_i = \begin{cases} -|f_i|, |\epsilon_i + f_i| < |\epsilon_i| \\ \ \ |f_i|, |\epsilon_i + f_i| > |\epsilon_i| \end{cases} \tag{14}$$

From Eq. 10, it follows that after extended training, for the vast majority of voxels and b-value signals, $|\epsilon_i + f_i| < |\epsilon_i|$, in this case $\Delta_i = -|f_i|$. That means the larger the absolute value of the fitting error $|f_i|$, the larger $|\Delta_i|$ and the smaller $\Delta_i$, indicating a more significant quality improvement of the predicted signal $p_i$ relative to the noisy signal $n_i$.

Since this relationship holds for the vast majority of voxels and b-value signals, the predicted signals exhibit higher overall quality than the noisy signals across all voxels and b-values. Moreover, the larger the absolute value of the fitting error $|f_i|$, the more pronounced this advantage becomes. This finding strongly supports Phenomenon 2: After extended training, as the Fitting Error increases, the overall quality gap between predicted signals and noisy signals increases. It should be noted that, in the actual statistical analysis, the fitting error is assumed to be positive by default; therefore, fitting error in Phenomenon 2 refers to $|f_i|$ rather than $f_i$.

## B. Convergence plot

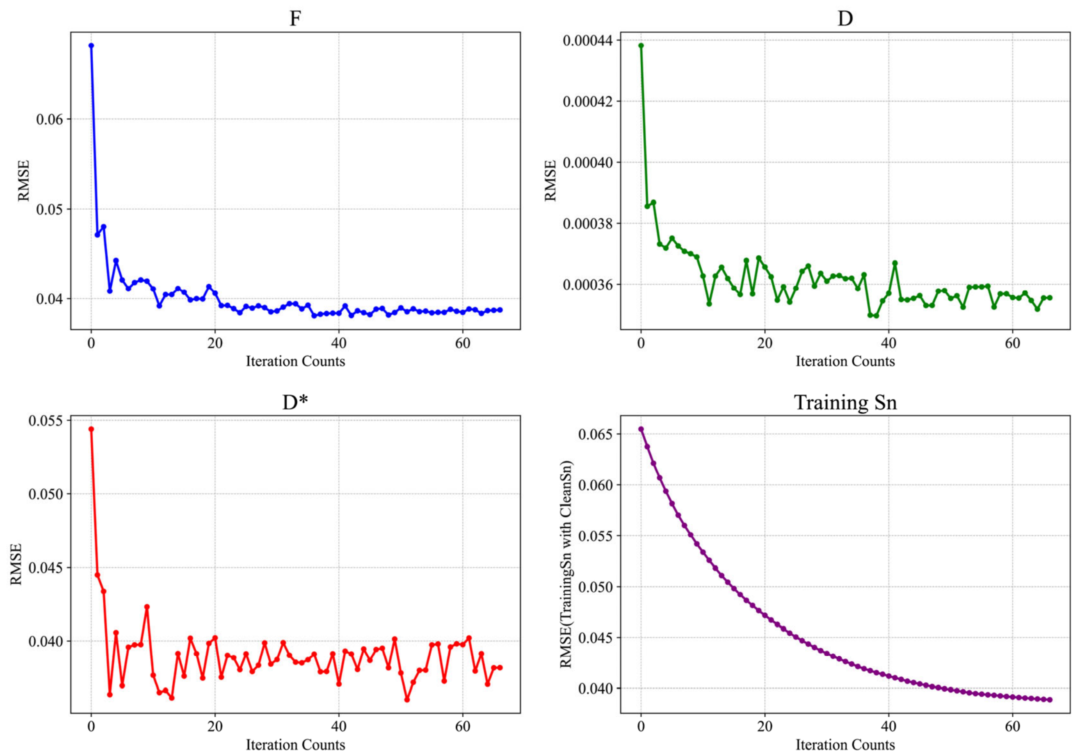



Figure 3: RMSE of SI-CNN in predicting F, D, and D* parameters (computed as the mean squared error between the predicted parameters and the ground truth) and RMSE of training signals (computed as the mean squared error between the training signals and the clean signals) as a function of the times of iterations for simulated abdominal data at SNR = 15. The x-axis indicates the times of data iterations.。

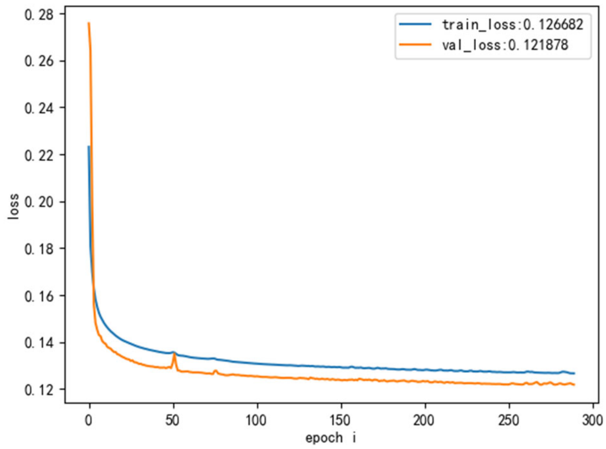



Figure 4: Training and Validation Loss.

## C. **Ablation study**

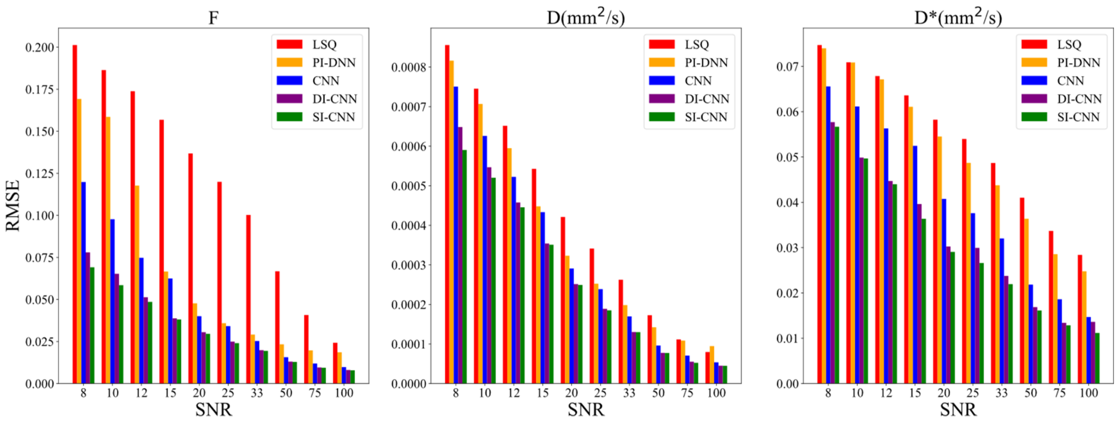



Figure 5: Comparison of the RMSE of predicted parameters for different methods under various SNR conditions.The DI-CNN represent the study using only the "Iterative Optimization" module.

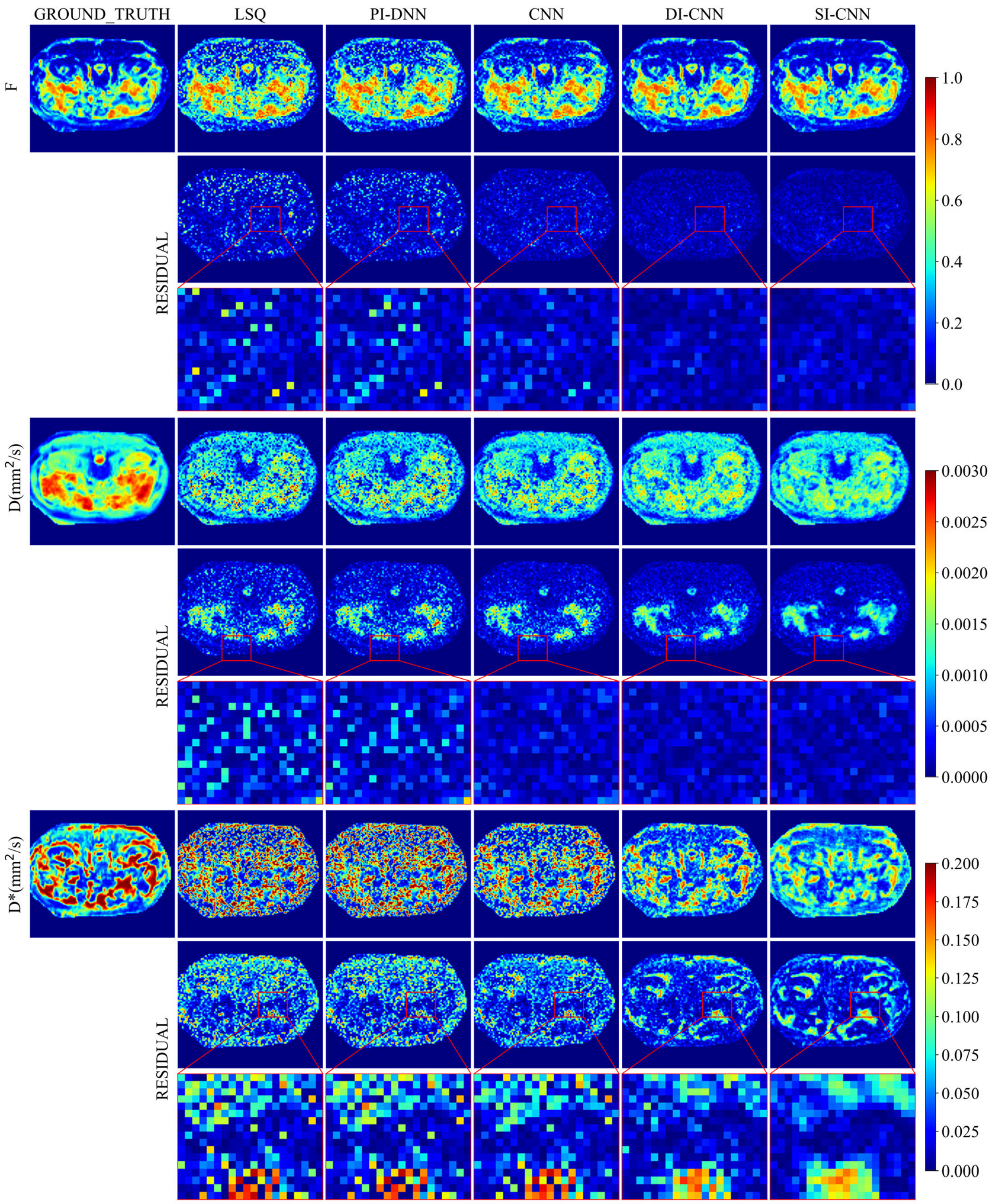



Figure 6: Parameter maps obtained by different fitting methods for a simulated abdomen in SNR=10. The DI-CNN represent the study using only the "Iterative Optimization" module.

## D. Related Experimental Logs

# Simulation experiment

# 2025.4.10

# ssh running study

| sicnn lamuda05 simulation snr | status | |
|---|---|---|
| 8 | over | |
| 10 | over | |
| 12 | over | |
| 15 | over | |
| 20 | over | |
| 25 | over | |
| 33 | over | |
| 50 | over | |
| 75 | over | |
| 100 | over | |

# 2025.4.11

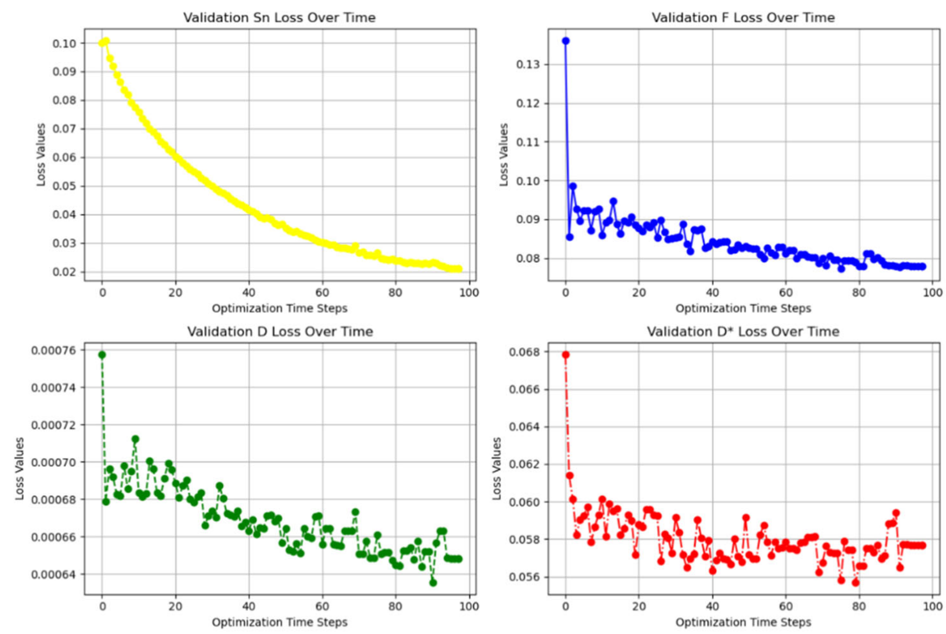
# running

| sicnnaround simulation snr 8 | training time(minutes) | val_sn_loss | val_f_loss | val_d_loss | val_d*_loss |
|---|---|---|---|---|---|
| lamuda0.01 | 43，46 | 0.020934 | 0.077988 | 0.000648 | 0.057689 |
| lamuda0.5 | 7，10 | 0.031245 | 0.123295 | 0.001034 | 0.082674 |
| lamuda0.001 | 118，134 | 0.020476 | 0.075563 | 0.000744 | 0.058039 |
| lamuda0.1 | 19，17 | 0.021232 | 0.074345 | 0.000723 | 0.056163 |
| lamuda0.05 | 27, 27 | 0.020923 | 0.076196 | 0.000749 | 0.056033 |

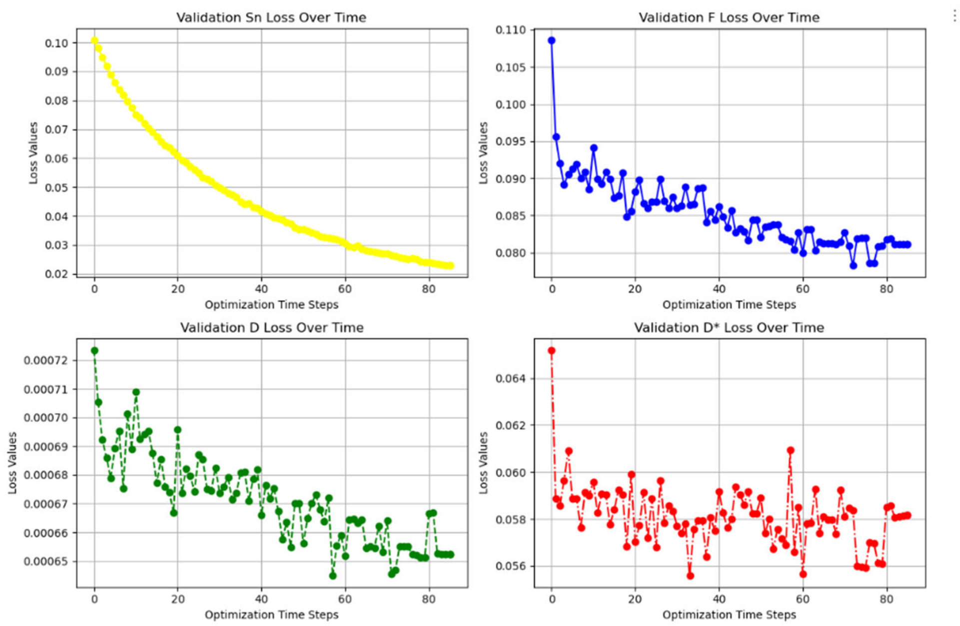
| sicnn use for show time simulation snr | 当前状态 | |
|---|---|---|
| 8 | training | over |
| 10 | training | over |
| 12 | training | over |
| 15 | training | over |
| 20 | training | over |
| 25 | training | over |
| 33 | training | over |
| 50 | training | over |
| 75 | training | over |
| 100 | training | over |

# 2025.4.14

## sicnn snr8  result




## sicnn snr8 fitting error 0.03,distance 0.02



## 2025.5.28

si_cnn(fix_new_net,dropout0.01,1e-4,batch25,lamuda01,fitting error 0.03,distance 0.02): adjust for best parameters

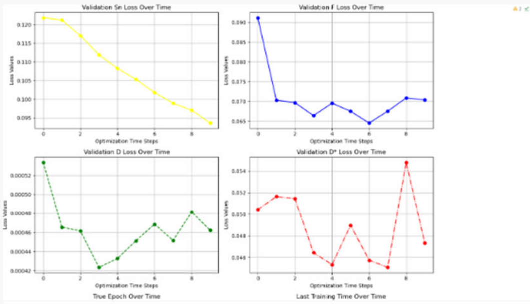
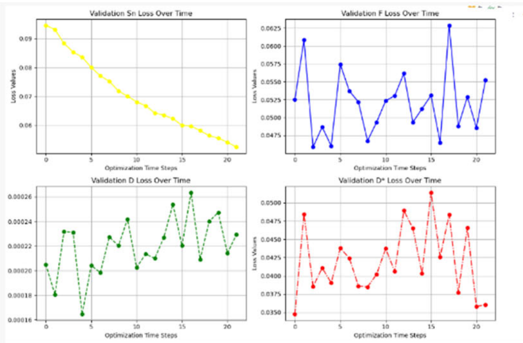
| simulation : around_si_cnn | training status | parameters | others | compare with other (fitting error and distance) |
|---|---|---|---|---|
| 8 | over | snr8,lr1e-4,batch25,delta005,fitting error 0.03,distance 0.02,lamuda01 | | yes |
| 10 | OVER | snr10,lr1e-4,batch25,delta005,fitting error 0.03,distance 0.02,lamuda01 | | yes |
| 12 | over | snr12,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,lamuda01 | | F,D better，d*worse |
| 15 | over | snr15,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,lamuda01 | | F,Dbetter,d*worse |
| 20 | over | snr20,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,lamuda01,autosave | | yes |
| 25 | over | snr25,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,lamuda01,autosave | | yes |
| 33 | over | snr33,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,lamuda01,autosave | | yes |
| 50 | over | snr50,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,autosave,limit200 | | yes |
| 75 | over | snr75,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,lamuda01,autosave,limit200 | | yes |
| 100 | test | snr100,lr1e-4,batch5,delta005,fitting error 0.03,distance 0.02,lamuda01,autosave,limit200 | | yes |
| ALL | | | around,fitting_max influence heavy | |

# 2025.5.31 keep trying for better result

| simulation : around_si_cnn | training status | parameters | others | result |
|---|---|---|---|---|
| 8 | | snr8,lr1e-4,batch25,delta005,around003,fitting_max002,lamuda001 | | |
| 10 | | snr10,lr1e-4,batch25,delta005,around003,fitting_max002,lamuda001 | | |
| 12 | | snr12,batch25,delta005,around003,fitting_max002,lamuda001 | | val_f_loss: 0.0485508106648922<br/> val_d_loss: 0.00044553776388056576<br/> val_dstar_loss: 0.04398658126592636 |
| 15 | | snr15,batch25,delta005,around003,fitting_max002,lamuda001 | | val_f_loss 0.038111<br/> val_d_loss 0.000351<br/> val_dstar_loss 0.036368 |
| 20 | | snr20,batch25,delta005,around003,fitting_max002,lamuda001,autosave | | |
| 25 | | snr25,batch25,delta005,around003,fitting_max002,lamuda001,autosave | | val_f_loss: 0.024040494114160538<br/> val_d_loss: 0.00018500156875234097<br/> val_dstar_loss: 0.026606647297739983 |
| 33 | | snr33,batch25,delta005,around003,fitting_max002,lamuda001,autosave | | val_f_loss 0.019424<br/> val_d_loss 0.000130<br/> val_dstar_loss 0.021947 |
| 50 | test | snr50,batch25,delta005,around003,fitting_max002,lamuda001 | | val_f_loss 0.012880<br/> val_d_loss 0.000077<br/> val_dstar_loss 0.016127 |
| 75 | test | snr75,batch25,delta005,around003,fitting_max002,lamuda001,autosave | | val_f_loss 0.009439<br/> val_d_loss 0.000052<br/> val_dstar_loss 0.012861 |
| 100 | | snr100,batch25,delta005,around003,fitting_max002,lamuda001 | | val_f_loss: 0.007904388941824436<br/> val_d_loss: 4.495037137530744e-05<br/> val_dstar_loss: 0.01116608455771683 |

# 2025.6.3 true experiment

sicnn batch5,delta005,fitting error 003,fitting_max002,lamuda001

| sicnn | status | results | results |
| --- | --- | --- | --- |
| snr12 | over | val_f_loss 0.065930<br/> val_d_loss 0.000441<br/> val_dstar_loss 0.036248 | val_f_loss: 0.06088801845908165<br/> val_d_loss: 0.00040284631540998816<br/> val_dstar_loss: 0.03552285581827164 |
| snr12 | wrong | |   |
| snr33 | over | val_f_loss 0.043301<br/> val_d_loss 0.000200<br/> val_dstar_loss 0.025823 | |
| snr33 | test | |   |
| snr33 | test | | |
| after validation, around_diff = 0.02,fitting_max=0.03,result better | | | val_f_loss 0.041317<br/> val_d_loss 0.000177<br/> val_dstar_loss 0.020673 |
| snr33 around_si_cnn_limit80 | | | |
| snr33<br /> around_si_cnn_batchsize40 | | | |
| snr33<br /> around_si_cnn_batchsize60 | | | |

**Reference**


1. Huang, H. M. (2022). An unsupervised convolutional neural network method or estimation of intravoxel incoherent motion parameters. Physics in Medicine & Biology, 67(21), 215018.
2. Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE transactions on image processing, 26(7), 3142-3155.