APCC 2022, Jeju Island, Korea

Tutorial 2:
Prof. Tomoaki Ohtsuki (Keio University)
Prof. Guan Gui (Nanjing University of Posts and Telecommunications)
Title: Deep Learning Aided Intelligent Sensing and Identification for Secure Wireless Communications

Deep Learning-Aided Intelligent Sensing and Identification for Secure Wireless Communications

Guan Gui
College of Telecommunications and Information Engineering
Nanjing University of Posts and Telecommunications
Nanjing, China
E-mail: guiguan@njupt.edu.cn

Outline

I. Background

II. Deep Learning-based AMC Methods

III. Deep Learning-based SEI Methods

IV. Deep Learning-based CSI Inferring Methods

V. Deep Learning-based Beamforming Design Methods

VI. Concluding Remarks

I. Background

Solve classical and new problems well

—— B5G and 6G demand more powerful tools

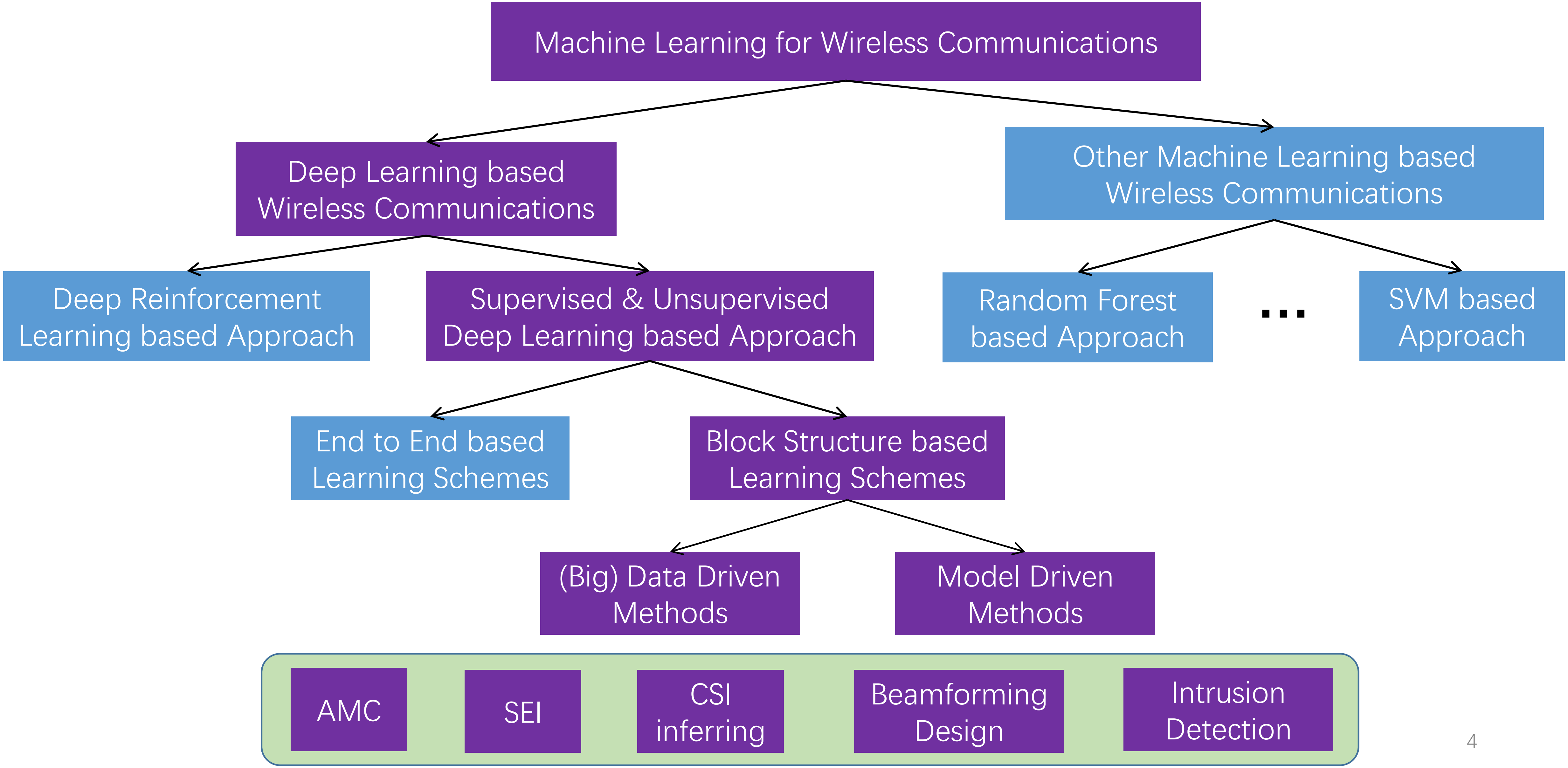
- serving with more devices and applications
- generating more amounts of data
- requiring lower communication delay
- facing more complicated situations
- demanding much smarter decision making skills
- more vulnerable to security and privacy threats

Compared with traditional algorithms—— Deep neural networks and deep learning have become the most effective and efficient machine learning technologies for various applications

- compatible to GPU and TPU
- stronger capability of fitting unknown and complex functions as black boxes
- better performances on feature learning automatically

G. Gui, M. Liu, F. Tang, N. Kato, F. Adachi, "6G: Opening New Horizons for Integration of Comfort, Security, and Intelligence," *IEEE Wireless Communications Magzine*, vol. 27, no. 5, pp. 126-132, 2020.

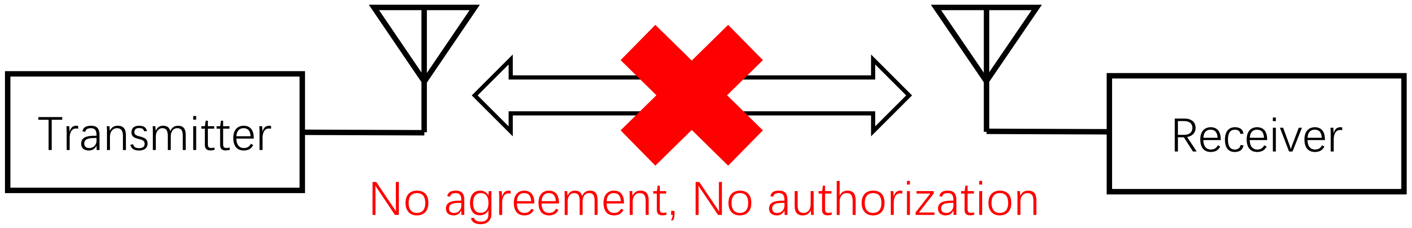
I. Background



II. Deep Learning-based AMC Methods

Background of AMC

- **Automatic Modulation Classification (AMC)** - a key technique for non-cooperative communication systems to recognize different modulation types relying on received signals. There are generally no agreement and authorization between transmitter and receiver.



- Recently, deep learning (DL)-based AMC has outperformed these traditional methods in both **performance and efficiency**.
- DL-based AMC is generally modeled as multi-classification problem. Based on maximum a posteriori (MAP) criterion, it can be written as follows.

$$\tilde{m} = \arg\max_{m \in \boldsymbol{M}} F_{DL}(m|\boldsymbol{R})$$

variable notations for modelling the AMC problem

$\boldsymbol{R}$:	The received signal
m:	The real modulation type
$\tilde{m}$:	The predicted modulation type
$\boldsymbol{M}$:	The modulation type pooling
F_{DL}:	The DL model

II. Deep Learning-based AMC Methods

Our work scope in deep learning-based AMC methods:

- **Deep Learning for Automatic Modulation Classification in SISO Systems**
  - Lightweight Automatic Modulation Classification (LightAMC)
  - Federated Automatic Modulation Classification (FedeAMC)
- **Deep Learning for Automatic Modulation Classification in MIMO System**
  - Multi-Antenna Cooperative Automatic Modulation Classification (Co-AMC)
  - CSI and Zero Forcing-aided Automatic Modulation Classification (ZF-AMC)
  - Transfer Learning-based Automatic Modulation Classification (TL-AMC)

II. Deep Learning-based AMC Methods

Lightweight Automatic Modulation Classification (LightAMC)

Y. Wang, J. Yang, M. Liu, and G. Gui, “LightAMC: Lightweight Automatic Modulation Classification via Deep Learning and Compressive Sensing,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3491-3495, 2020.

II. Deep Learning-based AMC Methods

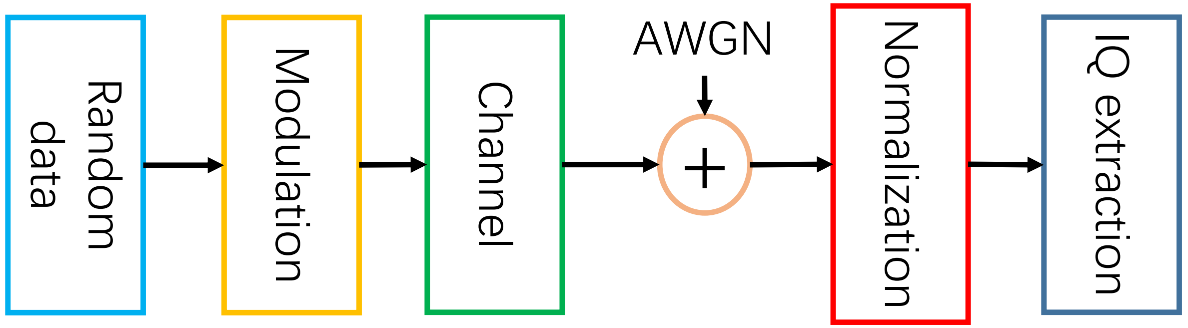
(1) Signal model

$$r(n) = Ae^{j\left(\Delta\theta + 2\pi\Delta f \frac{n}{N}\right)} s(n) + w(n), 0 \le n \le N-1$$

$r(n)$:	The received complex baseband signal
$s(n)$:	The modulation signal
$w(n)$:	Additive white Gaussian noise (AWGN)
A:	Channel gain, and it is a real value in $(0,1]$
$\Delta\theta$:	Time-varying phase offset, and $\Delta\theta \sim \mathrm{U}(0, \frac{\pi}{16})$
Δf:	Normalized frequency offset $(\Delta f = 0.1)$
N:	The number of sampling points

variable notations for modelling the modulation signals

(2) Dataset generation



- $\boldsymbol{R_{IQ}} = \begin{bmatrix} real(r(0)) & real(r(1)) & \ldots & real(r(N-1)) \\ imag(r(0)) & imag(r(1)) & \ldots & imag(r(N-1)) \end{bmatrix}$ and $\boldsymbol{R_{IQ}}$ is a real matrix with dimensionality 2×N (N=128).
- The modulation candidate pool: $\boldsymbol{\Theta}_1 = \{\mathrm{BPSK}, \mathrm{QPSK}, 8\mathrm{PSK}\}, \boldsymbol{\Theta}_2 = \{\mathrm{BPSK}, \mathrm{QPSK}, 8\mathrm{PSK}, 16\mathrm{QAM}\}$.
- SNR is random, and $\mathrm{SNR} \sim \mathrm{U}\ (-10, 10)\ \mathrm{dB}$

II. Deep Learning-based AMC Methods

(3) Introduction of lightweight methods

Task

Faster Feedforward Computing Speed

Smaller Model Size

Stable Performance

Lightweight method for Network Accelerating and Compression

Technology

Efficient Structure Design

- Group Convolution
  - ShuffleNet
  - MobileNet
- Separable Convolution
  - Bottleneck
  - SqueezeNet

Neuron Pruning and Sparsification

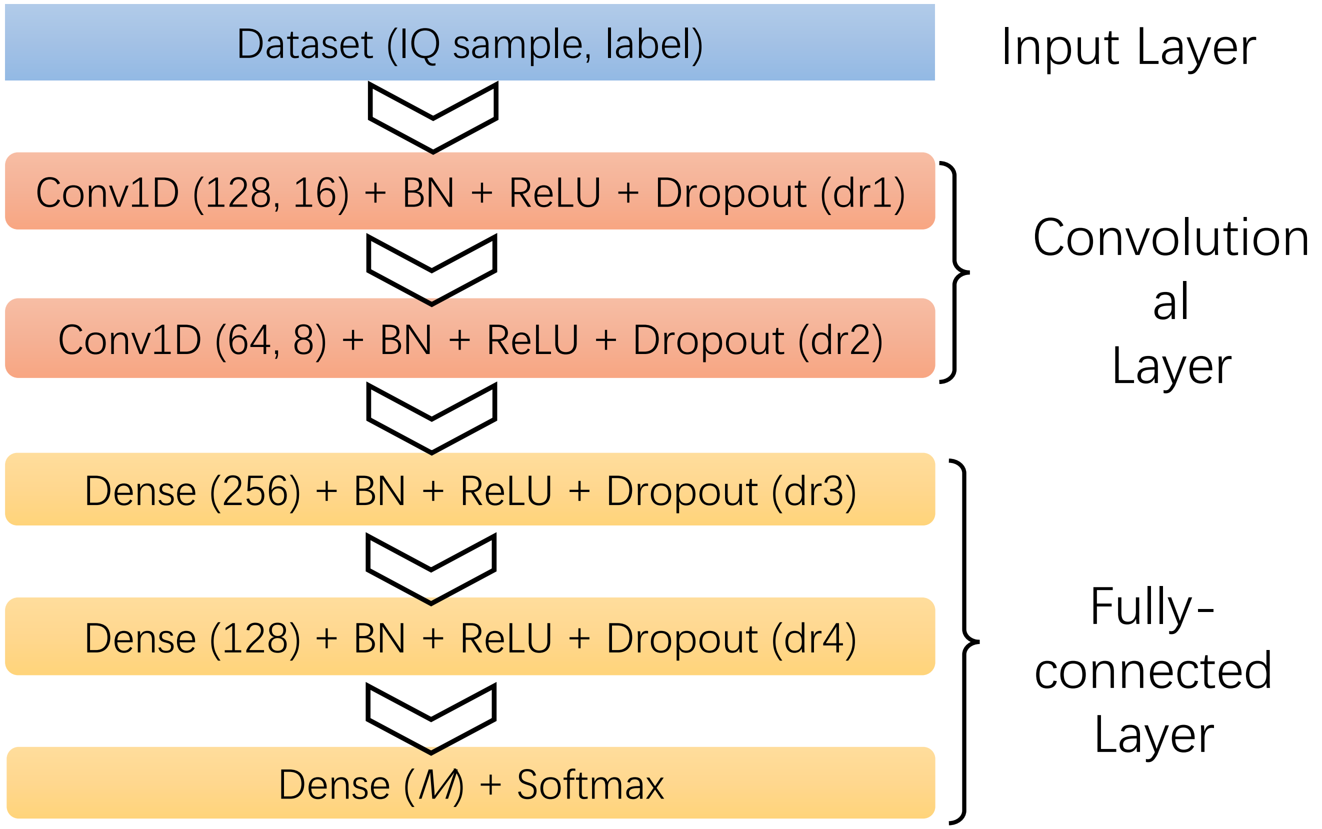
- Model Pruning
  - Weight sum (WS)
  - Average percentage of zero activation (APOZ)
  - **Sparsity regularization**
- Kernel Sparsification
  - Sparse constraint

Model Quantization

- Binary neural network (BNN)
- XNOR-Net
- Ternary weight network (TWN)
- quantized neural network (QNN)

II. Deep Learning-based AMC Methods

(4) Our original Deep Neural Network for AMC



Original CNN structure for the AMC technology.

Training Tips:

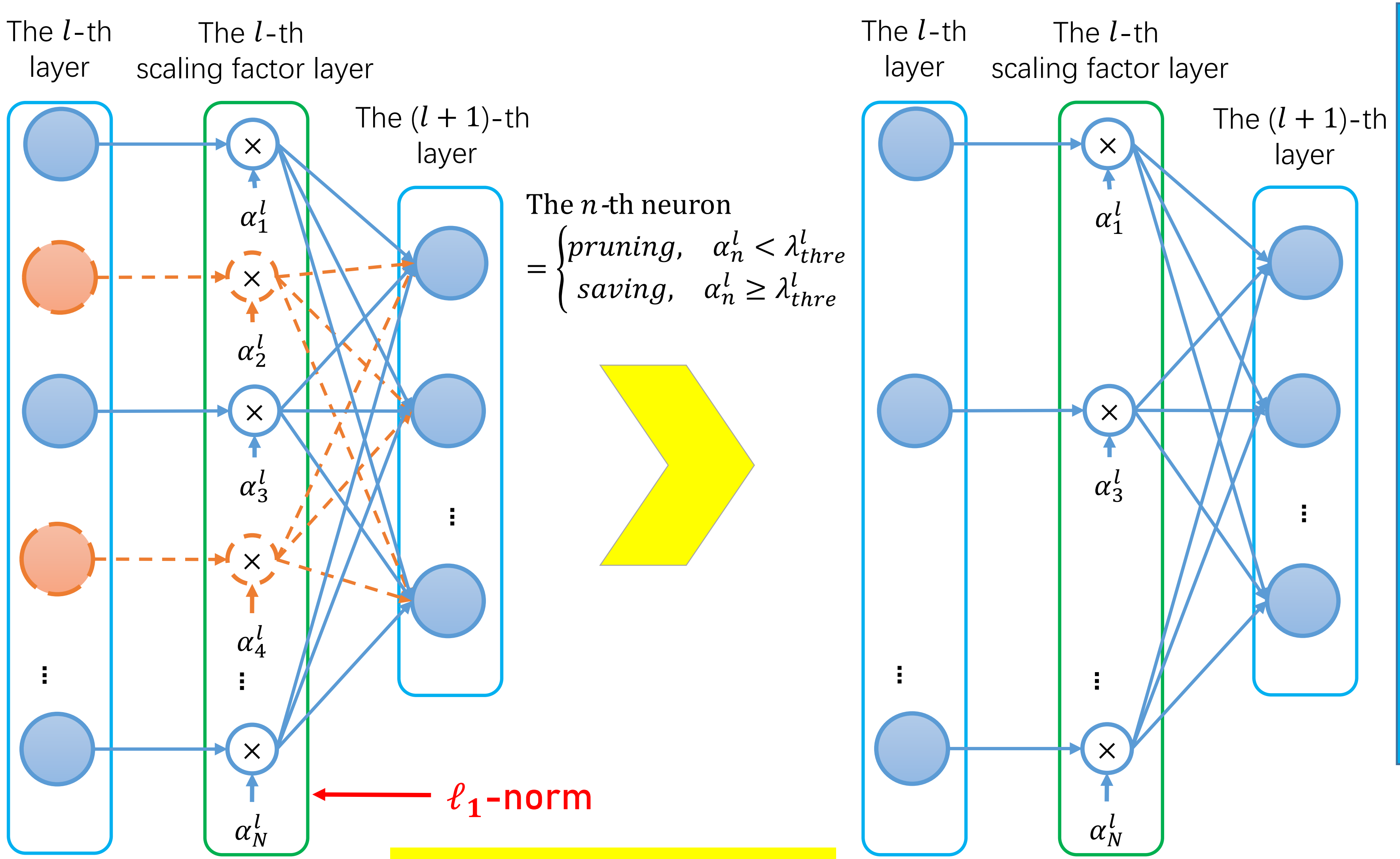
- ✓ 5 layers: 2 convolutional layers and 3 fully connected layers.
- ✓ Apply Batch Normalization (BN) and dropout : to avoid overfitting and accelerating training.
- ✓ Convolutional Neural Network representation:

$$x^l_{output} = f_{ReLu}\left\{\gamma^l \cdot BN_{\mu^l,\sigma^l,\epsilon^l}\left(W^l * x^l_{input} + b^l\right) + \beta^l\right\}$$

- $BN_{\mu^l,\sigma^l,\epsilon^l}(z) = \frac{z-\mu^l}{\sqrt{(\sigma^l)^2+\epsilon^l}}$
- $\mu^l = \frac{1}{N}\sum_{i=1}^{N} z(i)$
- $(\sigma^l)^2 = \frac{1}{N}\sum_{i=1}^{N}[z(i) - \mu^l]^2$

II. Deep Learning-based AMC Methods

(5) Our Lightweight Deep Neural Network for AMC



Neuron pruning for LightAMC.

Training Tips:

✓ Add scaling factor for CNN:

$$x_{output}^l = \alpha \cdot f_{ReLu}\{\gamma^l \cdot BN_{\mu^l,\sigma^l,\epsilon^l}(W^l *$$

✓ Objective function:

Define training samples: $T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_S, y_S)\}$,

$$\arg\min_{W,b,\gamma,\beta,\alpha} \sum_{s=1}^{S} l[f_{CNN}(x_s; W, b, \gamma, \beta, \alpha), y_s] + \lambda \|\alpha\|_1$$

II. Deep Learning-based AMC Methods

(4) Experimental results for LightAMC

Model	Structure/Θ_1	Model size/Θ_1	Structure/Θ_2	Model size/Θ_2
CNN-based AMC	128-64-256-128	15.5MB	128-64-256-128	15.5MB
LightAMC (Proposed)	77-18-49-44	1.0MB (93.5%↓)	81-19-63-49	1.3MB (91.6%↓)

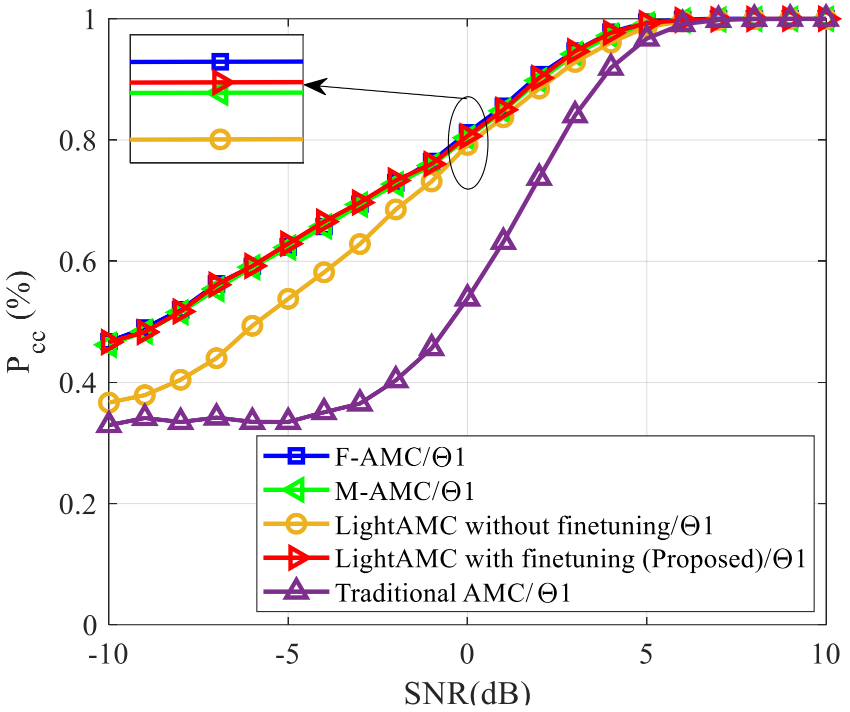
- Compared with M-AMC, our proposed LightAMC only has less than **7%** and **9%** of original CNN model sizes in $\boldsymbol{\Theta_1}$ and $\boldsymbol{\Theta_2}$, respectively.

Model	$\bar{T}_C$ (us) / Θ_1	$\bar{T}_C$ (us) / Θ_2
CNN-based AMC	44.2	44.2
Traditional AMC	200.7	312.9
LightAMC (Proposed)	33.3 (24.6%↓)	33.6 (23.9%↓)

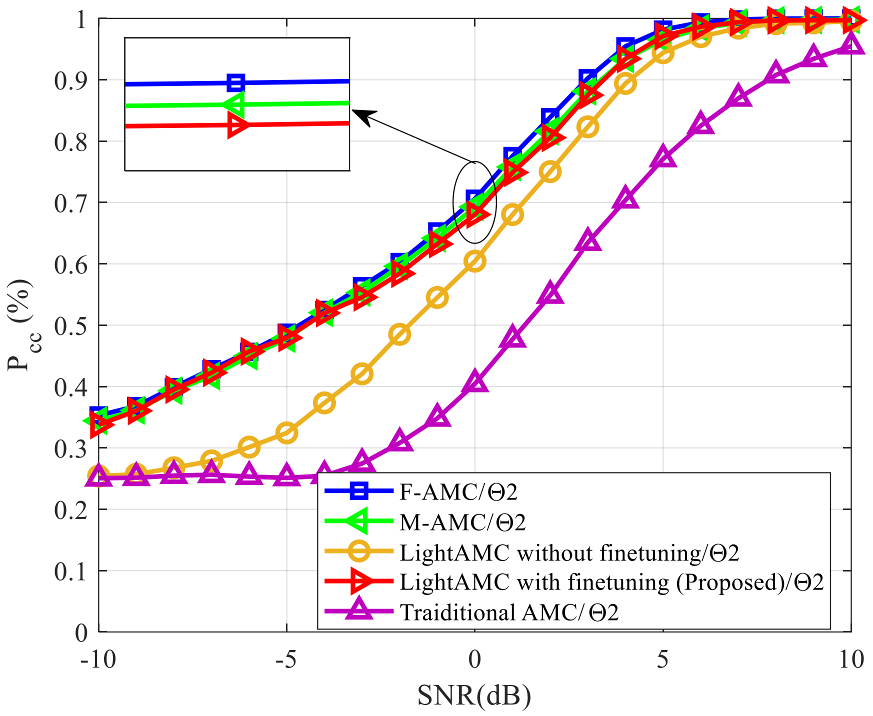
- Compared with CNN-based AMC, the computing time of our proposed LightAMC gets further reduction, and it has been reduced by nearly **24%** in both two datasets.

II. Deep Learning-based AMC Methods

$\bar{P}_{cc}$ (%) / Θ_1 with different SNR and AMC methods.



$\bar{P}_{cc}$ (%) / Θ_2 with different SNR and AMC methods.



Model	$\bar{P}_{cc}$ (%) / Θ_1	$\bar{P}_{cc}$ (%) / Θ_2
CNN-based AMC	78.63	70.35
Traditional AMC	62.93	51.12
LightAMC (Proposed)	78.93	70.10

- Compared with traditional AMC (HOC+SVM), CNN-based AMC has huge performance advantages.
- Our proposed LightAMC has similar performance with CNN-based AMC.

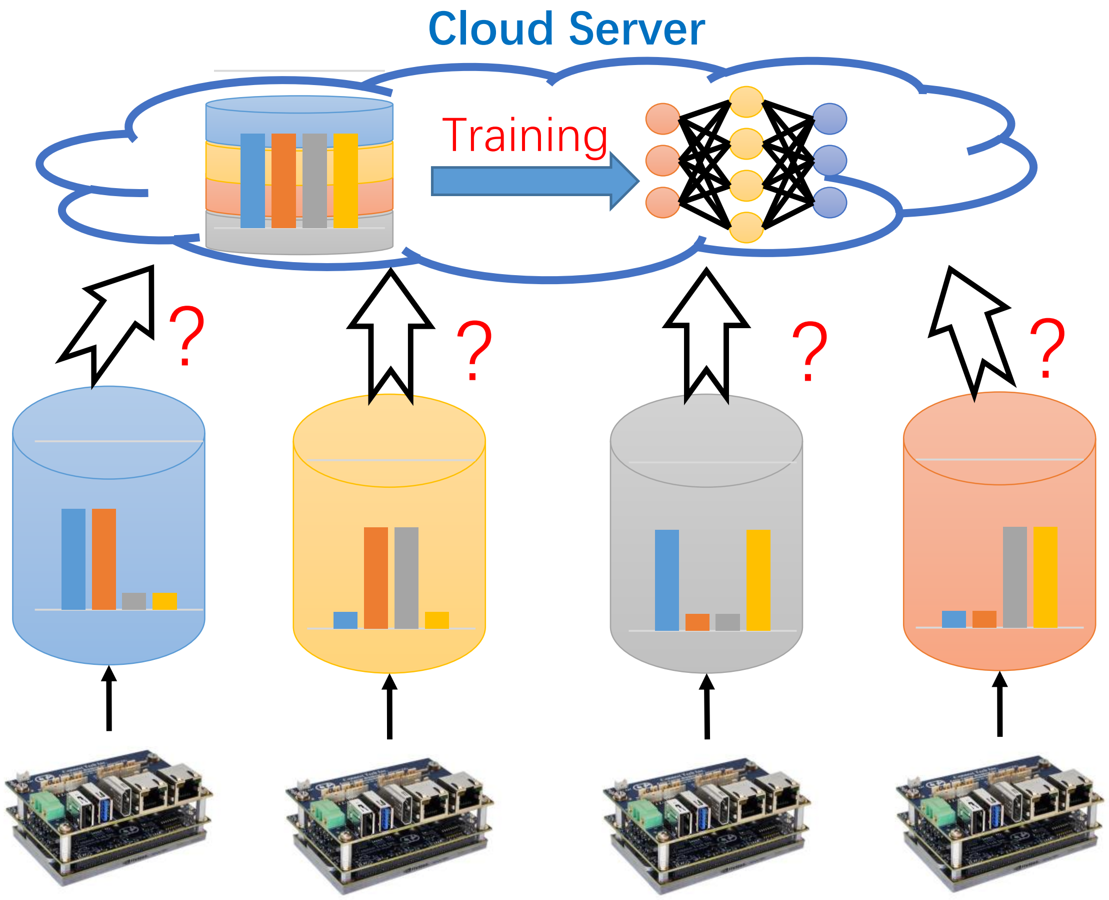
II. Deep Learning-based AMC Methods

Federated Automatic Modulation Classification (FedeAMC)

Y. Wang, G. Gui, H. Gacanin, B. Adebisi, H. Sari, and F. Adachi, "Federated Learning for Automatic Modulation Classification Under Class Imbalance and Varying Noise Condition," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 86-96, 2022.

II. Deep Learning-based AMC Methods

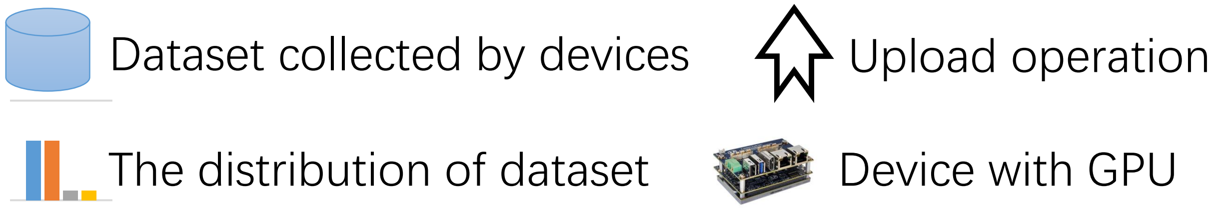
(1) Background and Problem



Background of the federated learning based AMC.

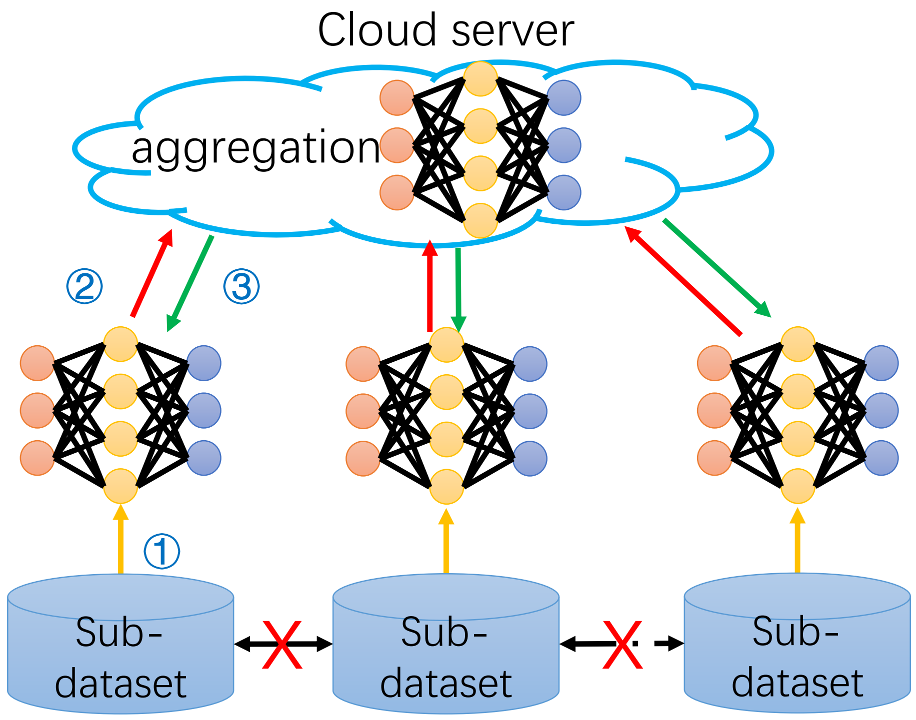
- From the left figure, the perfect DL model is trained on cloud server and based on **huge and balance** samples, uploaded from each device.
- Uploading each sample maybe impossible:

  (1) **High communication cost caused by too much data**;
  (2) **data privacy**.
- **How can we train a perfect DL model jointly without data sharing?**



II. Deep Learning-based AMC Methods

(2) System model based on federated learning (FL)



① **Train the sub-model**
② **Upload the key knowledge of the sub-model**
③ **Update the sub-model**

Background of the federated learning based AMC.

Federated learning (FL)

share knowledge rather than data

Steps:

1. Cloud server choose and initialize a DL model, and send it to each device;
2. Devices train the DL model on each sub-dataset;
3. Devices upload the learned knowledge (such as model weights);
4. Cloud server aggregate this knowledge and send it to each devices
5. Repeat Step 2 to Step 4.

II. Deep Learning-based AMC Methods

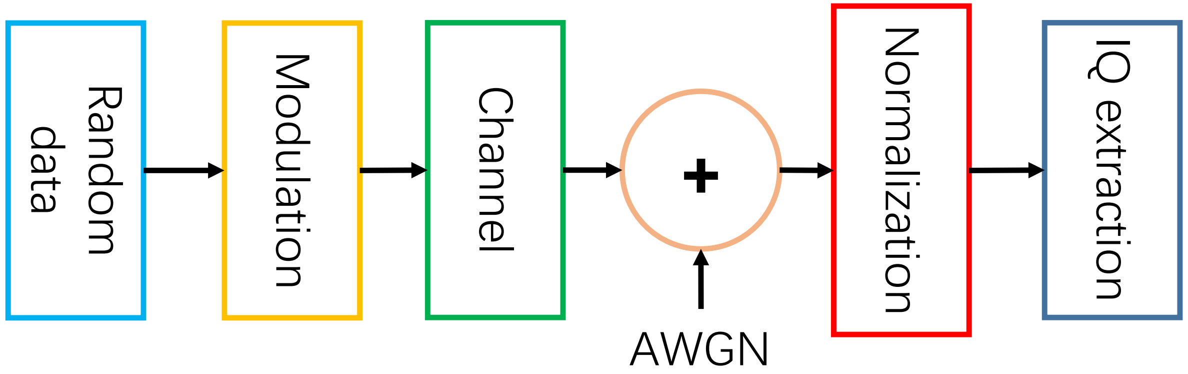
(3) Signal model

$$r(n) = Ae^{j(\Delta\theta + 2\pi\Delta f \frac{n}{N})} s(n) + w(n), 0 \le n \le N-1$$

variable notations for modelling the modulation signals

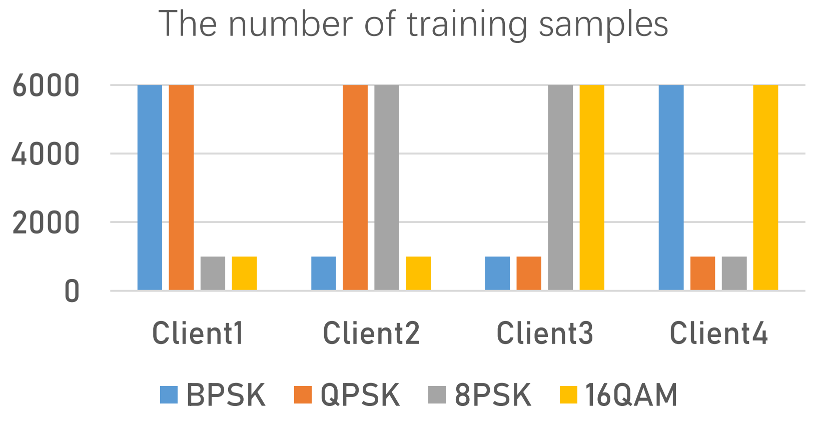
$r(n)$:	The received complex baseband signal
$s(n)$:	The modulation signal
$w(n)$:	Additive white Gaussian noise (AWGN)
A:	Channel gain, and it is a real value in $(0,1]$
$\Delta\theta$:	Time-varying phase offset, and $\Delta\theta \sim \mathrm{U}(0, \frac{\pi}{16})$
Δf:	Normalized frequency offset ($\Delta f = 0.1$)
N:	The number of sampling points

(4) Dataset with class imbalance



Simulation model for generating the dataset.

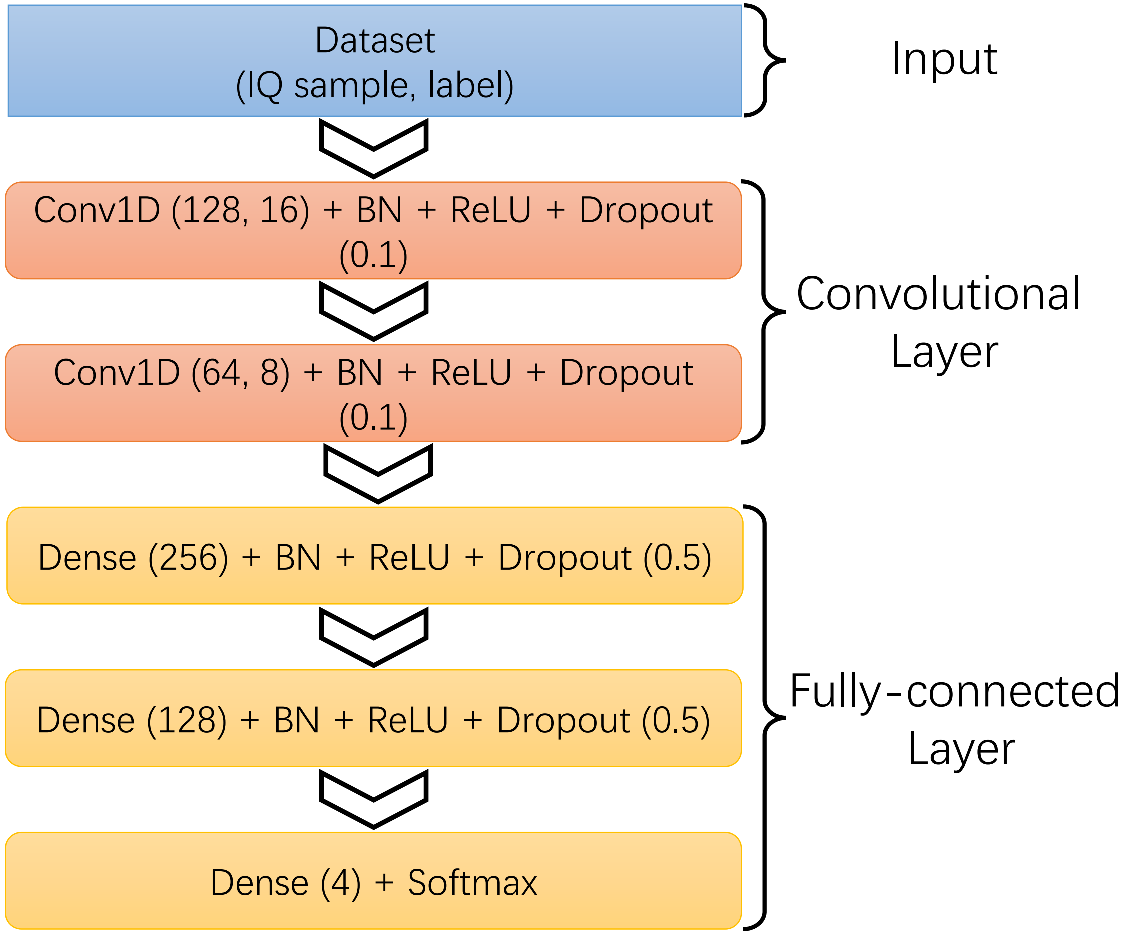
- The modulation candidate pool: $\mathfrak{M} = \{\mathrm{BPSK}, \mathrm{QPSK}, 8\mathrm{PSK}, 16\mathrm{QAM}\}$.
- SNR is random, and $\mathrm{SNR} \sim \mathrm{U}(-10, 10)\mathrm{dB}$
- We prepare four sub-dataset (with class imbalance) for simulations of four IoT devices, and their distributions are shown on the right.



Data distributions of four IoT devices

II. Deep Learning-based AMC Methods

(5) CNN structures for FedeAMC



CNN structure for the FedeAMC technology.

Training Tips:

- ✓ A common 5-layer CNN contains two "Conv1D" and 3 "Dense" .
- ✓ Batch normalization (BN) and dropout are adopted to prevent overfitting.
- ✓ Two learning algorithms are applied.
  - ✓ **Synchronous SGD (SSGD)**: share the gradients
  - ✓ **Model average (MA)**: share the model weights
- ✓ Assume that
  - w_t is the model weight at t-th epoch,
  - K is the number of devices,
  - T is the all training epochs,
  - η_t is the learning rate at t-th epoch,
  - M is the communication interval,
  - B is the number of batch in a epoch,

II. Deep Learning-based AMC Methods

(6) The descriptions of two FL algorithms: SSGD and MA

Algorithm1: Federated learning-based AMC (SSGD)

Initialize $w_t, K, T, \eta_t, and\ B$

w_t is the model weight at t-th epoch,
K is the number of devices,
T is the all training epochs,
η_t is the learning rate at t-th epoch,
B is the number of batch in a epoch,

for $t = 0,1,2,\ldots,T-1$ do

Load the current model w_t;

for $b = 0,1,2,\ldots,B-1$ do

Compute the current gradient at $k-th$ device $\nabla f_{k,b}(w_t)$;

Obtain gradients of all devices through synchronous

communication, $\{\nabla f_{1,b}(w_t), \nabla f_{2,b}(w_t), \ldots, \nabla f_{K,b}(w_t)\}$;

Update $w_{t+1} = w_t - \frac{\eta_t}{K}\sum_{k=1}^{K} \nabla f_{k,b}(w_t)$

end for

end for

Algorithm2: Federated learning-based AMC (MA)

Initialize $w_t, K, T, \eta_t, B\ and\ M$

for $t = 0,1,2,\ldots,T-1$ do

Load the current model $w_t^k = w_t$;

for $m = 0,1,2,\ldots,M-1$ do

for $b = 0,1,2,\ldots,B-1$ do

Compute the current gradient at $k-th$ device

$\nabla f_{k,m,b}(w_t^k)$;

Update $w_t^k = w_t^k - \eta_{t,m} \nabla f_{k,m.b}(w_t^k)$;

end for

end for

Obtain weights of all devices through synchronous communication,

Update $w_{t+1} = \frac{1}{K}\sum_{k=1}^{K} w_t^k$

end for

II. Deep Learning-based AMC Methods

(7) Loss function for class imbalance and its equivalent skill

Assume sample and labels are $\{(x_i, y_i)\}_{i=1}^{N_B}$ in a training batch:

N^m: The number of training samples with the modulation type m

α^m: class balance factor

$N^{max} = \max_{m\in M}(N^m)$

Variable notations

- Cross-entropy (CE) loss function

$$l_{CE} = -\frac{1}{N_B}\sum_{i=1}^{N_B} y_i \log\left(f_{CNN}(\boldsymbol{\theta}; x_i)\right) = -\frac{1}{N_B}\sum_{m\in M}\sum_{i=1}^{N^m} y_i^m \log\left(f_{CNN}(\boldsymbol{\theta}; y_i^m)\right)$$

class imbalance

Tips: increase the weight in the loss of class with small samples

- Balanced cross-entropy (BCE) loss function

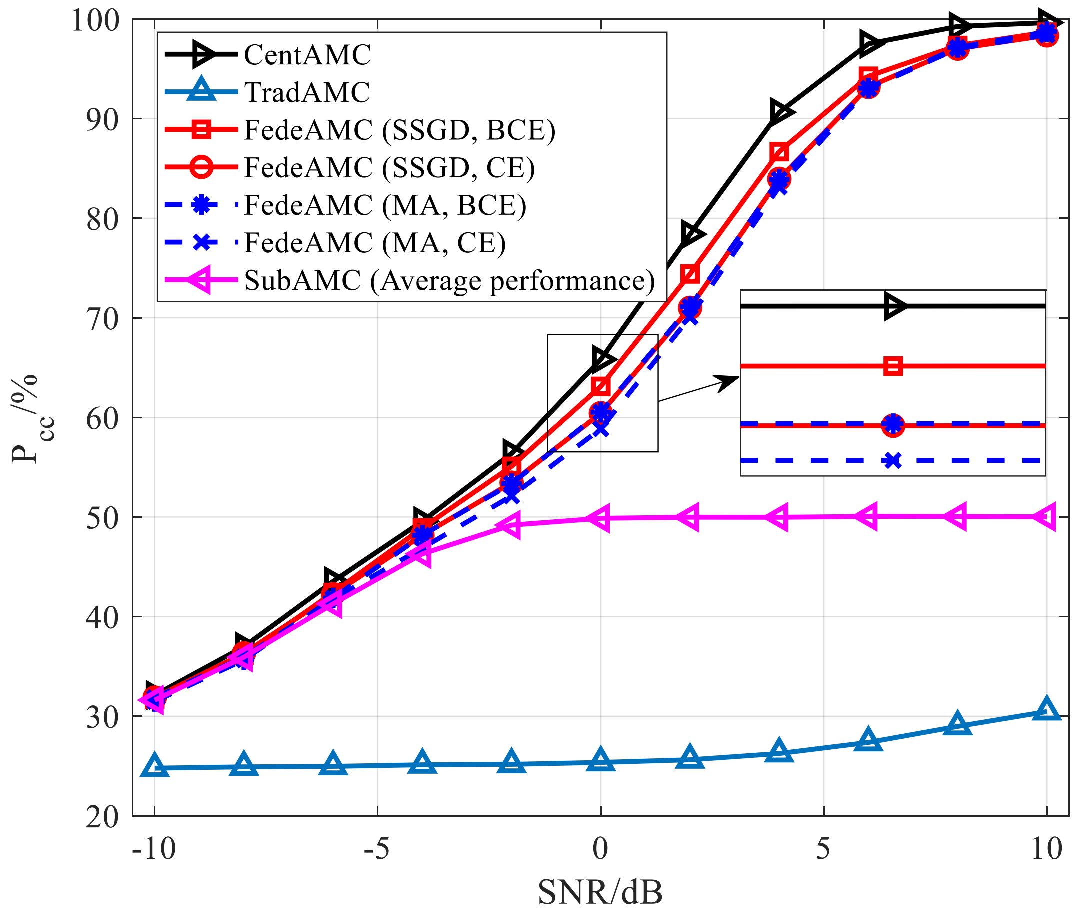
$$l_{BCE} = -\frac{1}{N_B}\sum_{m\in M}\sum_{i=1}^{N^m} \alpha^m y_i^m \log\left(f_{CNN}(\boldsymbol{\theta}; y_i^m)\right) = \underline{-\frac{1}{N_B}\sum_{m\in M}\frac{N^{max}}{N^m}\sum_{i=1}^{N^m} y_i^m \log\left(f_{CNN}(\boldsymbol{\theta}; y_i^m)\right)}$$

Equivalent to

CE + Data repeated expansion

II. Deep Learning-based AMC Methods

(8) Experimental results for FedeAMC

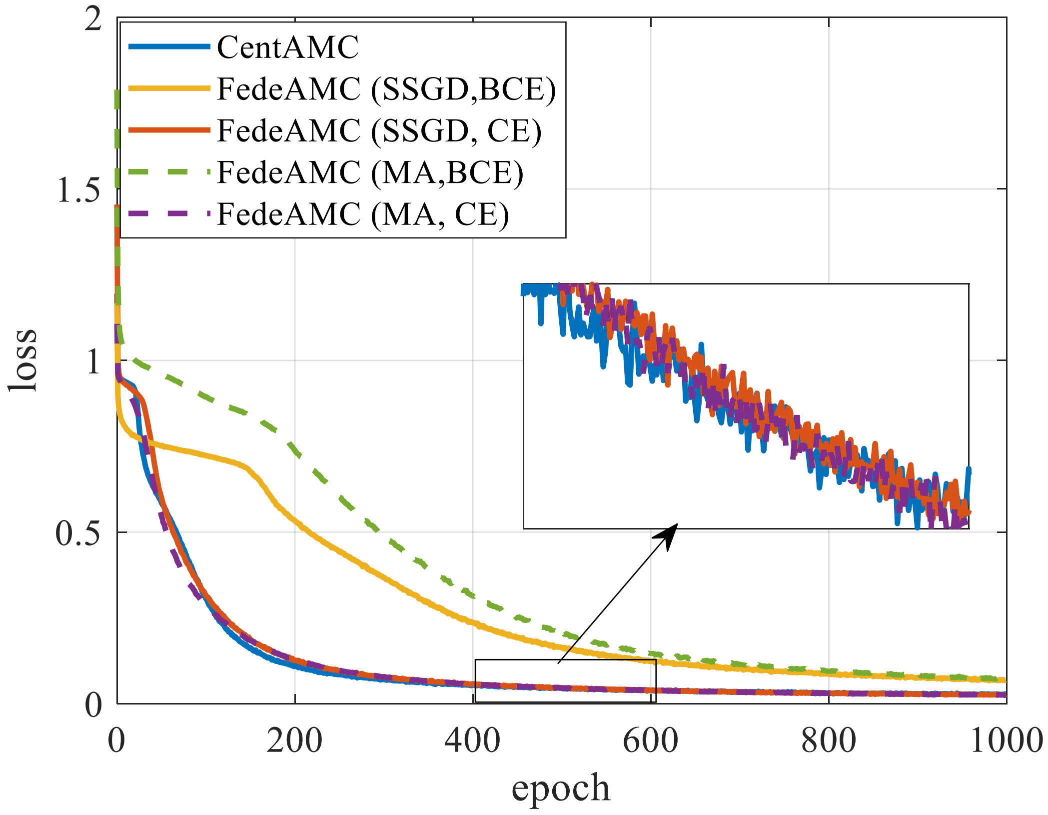


Experimental results on $\bar{P}_{cc}$ (%) with different SNR and AMC methods.

- "CentAMC" is the CNN-based AMC trained on dataset, which contains four sub-datasets, and it has **the best performance**;
- "TradAMC" (High order cumulants with ANN) has almost no work under the condition of training dataset with changing SNRs;
- "SubAMC" is the average performance of the CNN-based AMC trained on the corresponding sub-dataset.
- "FedeAMC" has better performance than "SubAMC", and it still has slight performance gap with "CentAMC" . Specifically, average performance loss is close to **2%**, and the highest performance loss is almost **4%** at 2 dB. In addition, in "FedeAMC"
  - SSGD is slightly beyond MA
  - BCE is slightly beyond CE

II. Deep Learning-based AMC Methods

(8) Experimental results for FedeAMC



Experimental results on loss with different epoch numbers and AMC methods.

- Convergence rate : **CentAMC ≈ FedeAMC (BCE) ≫ FedeAMC (CE)**, and the application of BCE can effectively accelerate the training.
- It is noted that although loss of FedeAMC (SSGD, CE) and FedeAMC (MA, CE) have difference before convergence, but they converge almost at the same epoch.

II. Deep Learning-based AMC Methods

Cooperative Automatic Modulation Classification (Co-AMC)

Y. Wang, J. Wang, W.Zhang, J. Yang, G. Gui, "Deep Learning-Based Cooperative Automatic Modulation Classification Method for MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4575-4579, 2020.

II. Deep Learning-based AMC Methods

(1) AMC for MIMO systems

Signal model $\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{n}_k$

- $\boldsymbol{y}_k = [y_k(1), y_k(2), \dots, y_k(N_r)]^T, k \in [1, N/N_t]$: the received signal in the moment k without considering carrier frequency offset and phase offset.
- $\boldsymbol{x}_k = [x_k(1), x_k(2), \dots, x_k(N_t)]^T, k \in [1, N/N_t]$: the transmitted source signal vector in the moment k, and $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{N/N_t}]$
- $\boldsymbol{n}_k \sim CN(0, \sigma_n^2 \boldsymbol{I}_{N_r})$: additive noise.
- $\boldsymbol{H} \sim CN(0, I_{N_r \times N_t})$: complex-valued MIMO channel

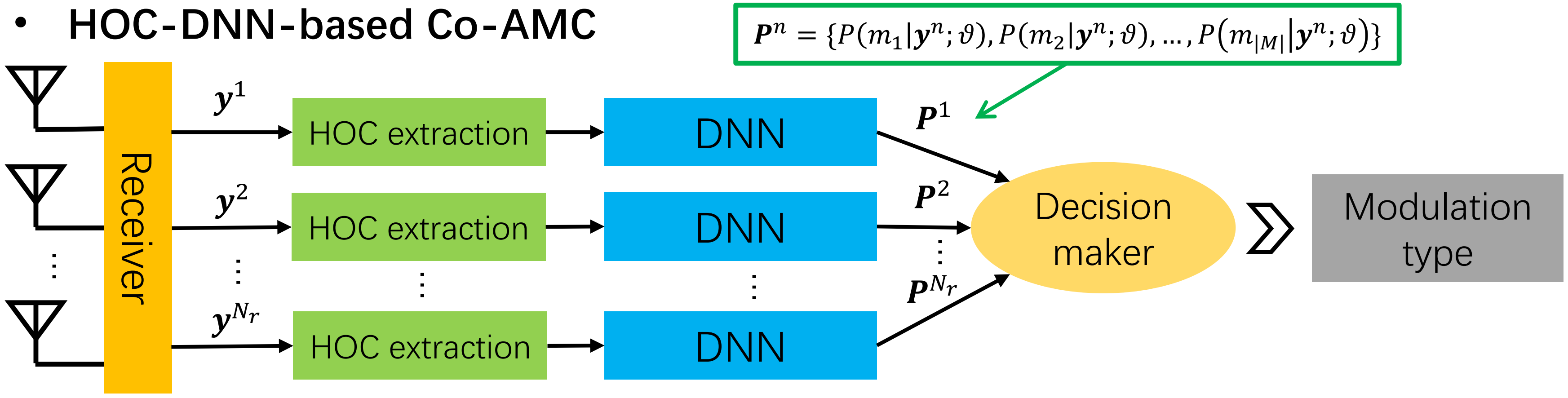
Dataset generation

- $\boldsymbol{x}^n = [x_1(n), x_2(n), \dots, x_{N/N_t}(n)], n \in [1, N_r]$: the transmitted vector in the n-th transmitting antenna, and $\boldsymbol{X} = [\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^{N_r}]^T$ is enforced into the unit power.
- $\boldsymbol{y}^n = [y_1(n), y_2(n), \dots, y_{N/N_t}(n)], n \in [1, N_r]$: the received vector in the n-th receiving antenna, and its real part and imaginary part as a set of training sample of the n-th antenna.

II. Deep Learning-based AMC Methods

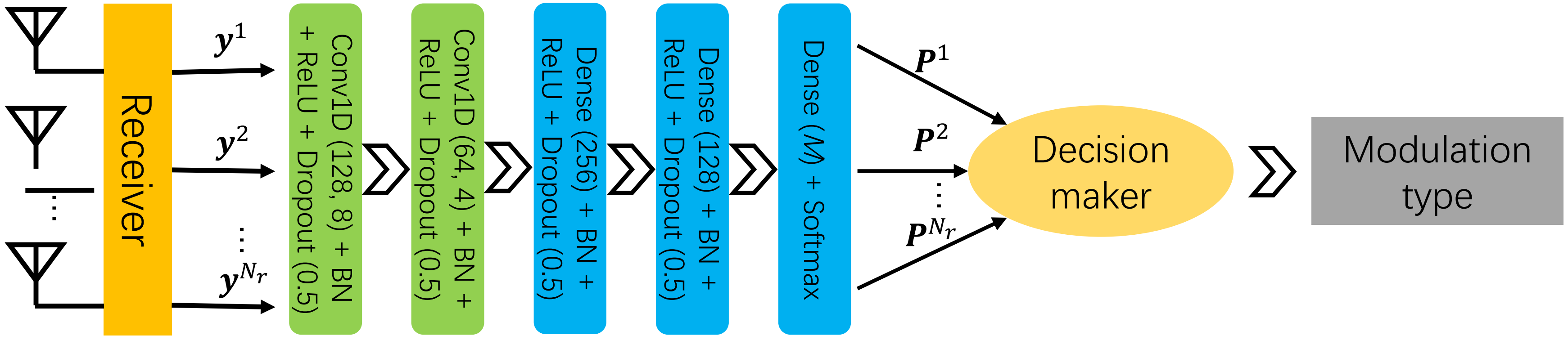
(2) Neural network structures for Co-AMC

- **HOC-DNN-based Co-AMC**



(a) The process of HOC and DNN-based Co-AMC.

- **CNN-based Co-AMC**



(b) The process of CNN-based Co-AMC.

Training Tips

Decision rules:

✓ **Majority voting (MV)**

$$\hat{m}(y^n) = \underset{M}{arg\ max}\ P^n$$

$$\hat{m} = major\{\hat{m}(y^1), \hat{m}(y^2), \dots, \hat{m}(y^{N_r})\}$$

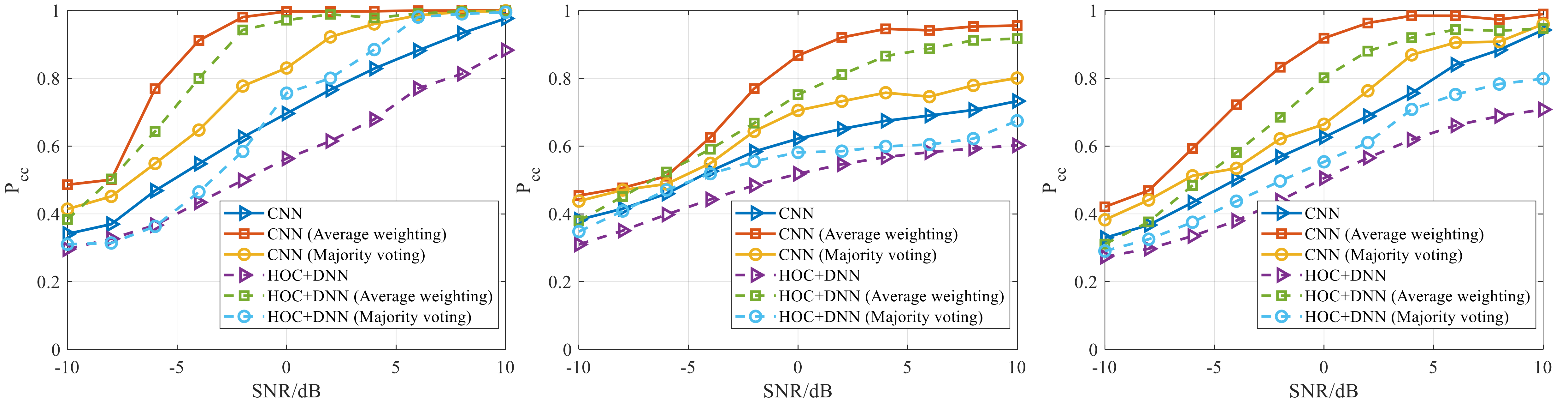
✓ **Average weighting (AW)**

$$\bar{P} = \frac{\sum_{k=1}^{N_r} P^n}{N_r}$$

$$\hat{m} = \underset{M}{arg\ max}\ \bar{P}$$

II. Deep Learning-based AMC Methods

(3) Experimental results for Co-AMC



Simulation results of *Pcc* for different schemes with (a). $N_t = 1, N_r = 4$, (b). $N_t = 2, N_r = 4$, (c). $N_t = 4, N_r = 4$.

- AW method is better than MV method, whether in CNN-based AMC or in HOC-DNN-based AMC;
- CNN-based AMC has better performance than HOC-DNN-based AMC.

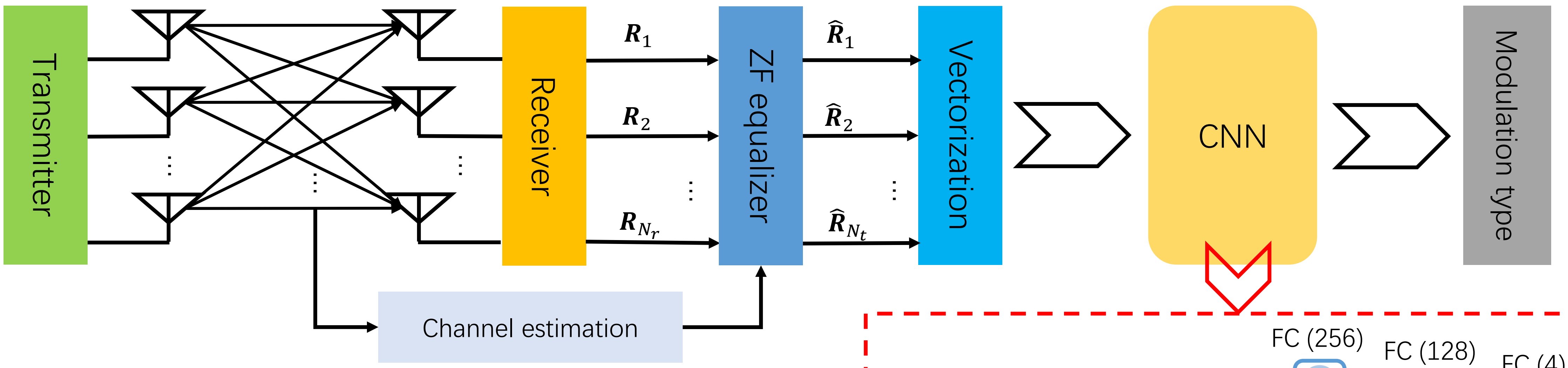
II. Deep Learning-based AMC Methods

Zero Forcing-aided Automatic Modulation Classification (ZF-AMC)

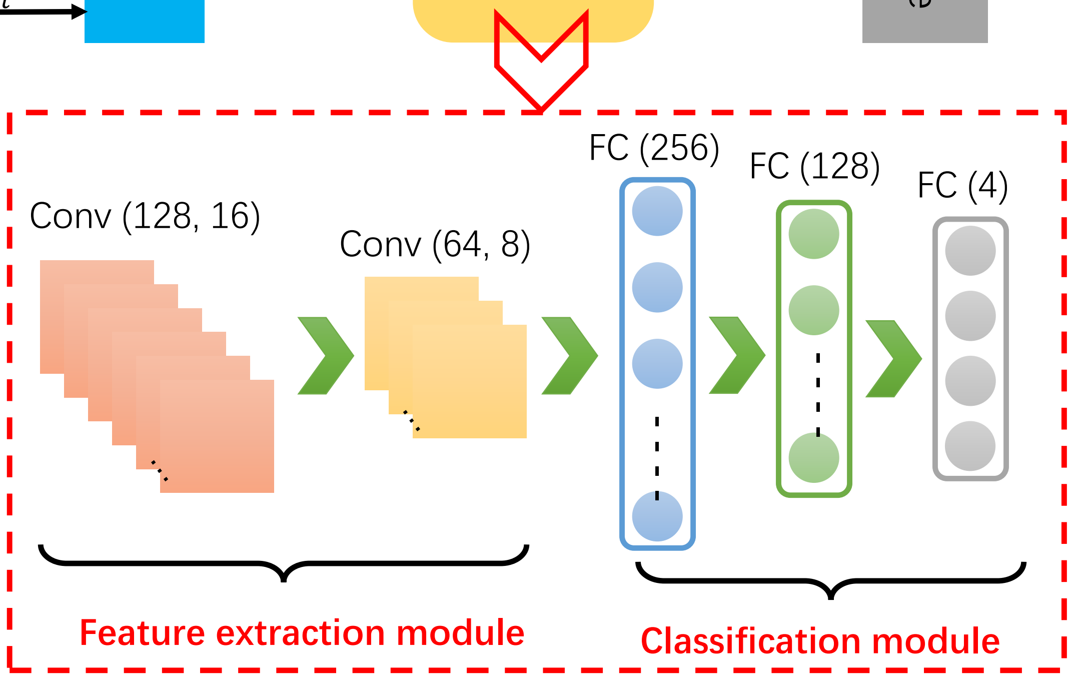
Y. Wang, G. Gui, *et al.*, "Automatic Modulation Classification for MIMO Systems via Deep Learning and Zero-Forcing Equalization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5688-5692, 2020.

II. Deep Learning-based AMC Methods

(1) System model



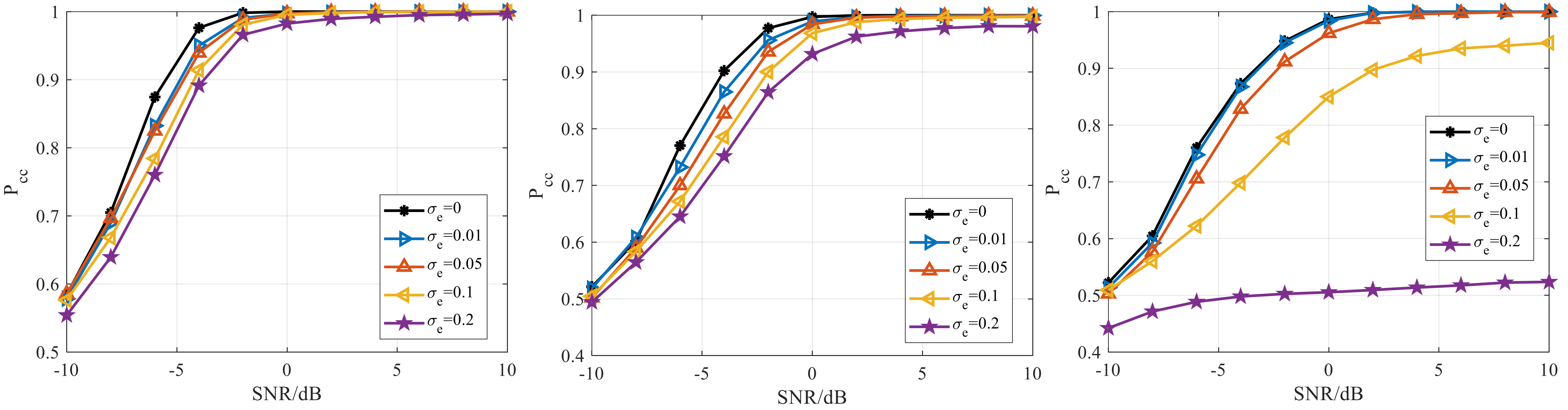
The process of HOC and DNN-based ZF-AMC.



- ZF-AMC uses channel state information (CSI) and equalization algorithm.
- Here, we consider different CSI conditions, including perfect CSI-aided ZF-AMC and imperfect CSI-aided ZF-AMC.

II. Deep Learning-based AMC Methods

(2) Experimental results for ZF-AMC



Simulation results of Pcc for different schemes with (a). $N_t = 1, N_r = 4$, (b). $N_t = 2, N_r = 4$, (c). $N_t = 4, N_r = 4$.

Training Tips:

✓ $\hat{\boldsymbol{x}}_k = ZF(\widehat{\boldsymbol{H}})\boldsymbol{y}_k = \left(\widehat{\boldsymbol{H}}^H\widehat{\boldsymbol{H}}\right)^{-1}\widehat{\boldsymbol{H}}^H(\mathbf{H}\boldsymbol{x}_k + \boldsymbol{n}_k)$

✓ Post-processing SNR: $\tilde{\gamma}_k = \frac{\gamma_k}{\left(1+\frac{\sigma_e}{1-\sigma_e}N_t\gamma_k\right)[(\boldsymbol{H}^H\boldsymbol{H})^{-1}]_{kk}}$

- The larger N_t, the more severe the performance degradation of the imperfect CSI-aided ZF-AMC

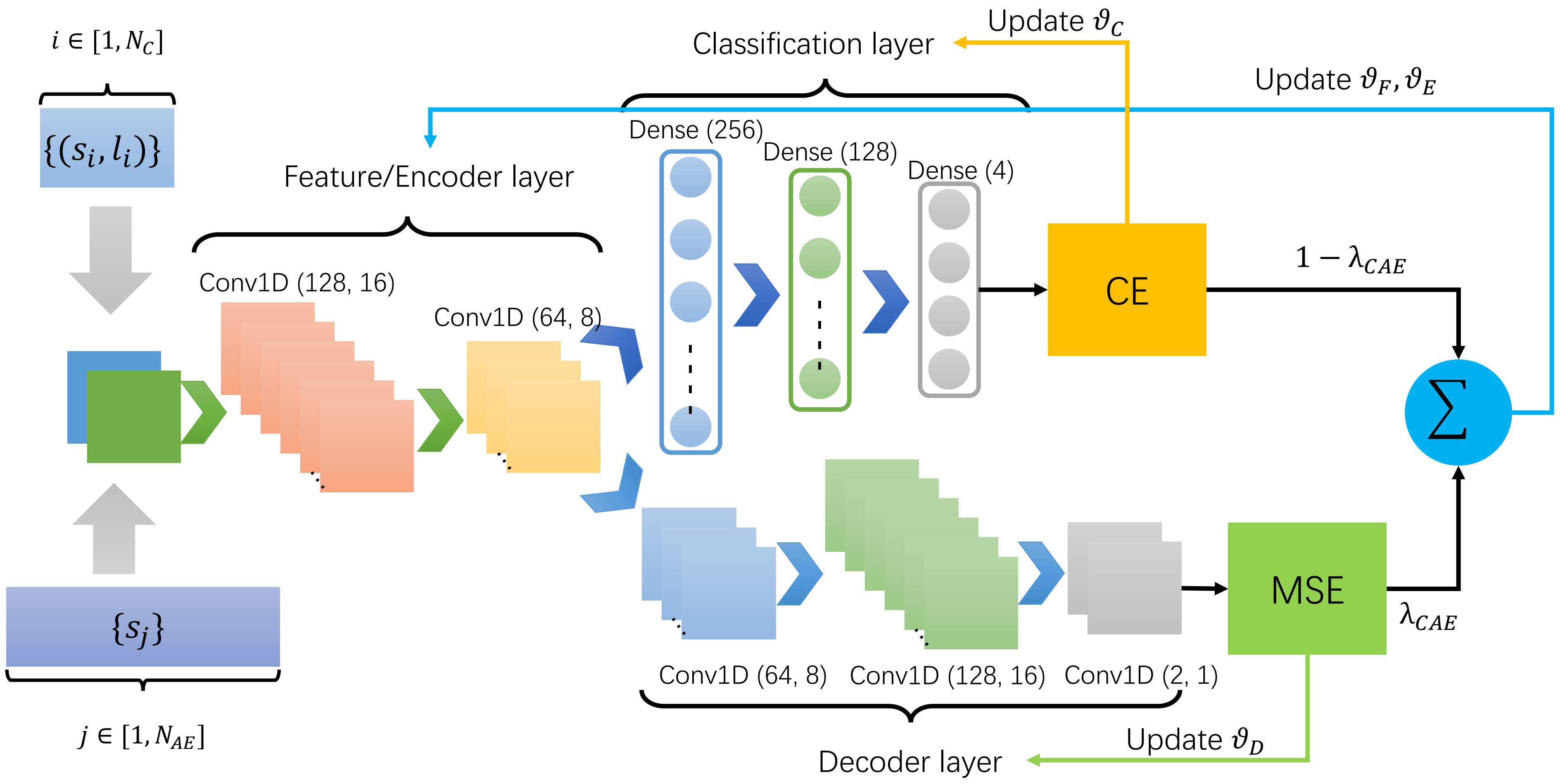
II. Deep Learning-based AMC Methods

Transfer Learning based Automatic Modulation Classification (TL-AMC)

Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, F. Adachi, “Transfer Learning for Semi-Supervised Automatic Modulation Classification in ZF-MIMO Systems,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 10, no. 2, pp. 231-239, 2020.

II. Deep Learning-based AMC Methods

(1) System model



The structure of TL-AMC.

Training Tips:

- ✓ ϑ_F, ϑ_E is the weight for feature and encode layer, and $\vartheta_F \equiv \vartheta_E$;
- ✓ ϑ_C, ϑ_D is the weight for classification and decode layer.
- ✓ $S_C = \{(s_i, l_i)\}_{i=1}^{N_C}$ for classification
- ✓ $S_{AE} = \{s_j\}_{j=1}^{N_{AE}}$ for auto-encoder
- ✓ $\frac{N_C}{N_C + N_{AE}} = 0.95$

II. Deep Learning-based AMC Methods

(2) Training details for TL-AMC

- **Loss function**: categorical cross entropy (CCE) for classification and mean square error (MSE) for auto-encoder:

$$L_{CCE}(\{\vartheta_F, \vartheta_c\}) = -\frac{1}{N_C}\sum_{i=1}^{N_C} l_i \, log(f_C(s_i; \{\vartheta_F, \vartheta_c\}))$$

$$L_{MSE}(\{\vartheta_E, \vartheta_D\}) = \frac{1}{N_{AE}}\sum_{j=1}^{N_{AE}} (f^C(s_j; \{\vartheta_E, \vartheta_D\}) - s_j)^2$$

$$\underset{\vartheta_E, \vartheta_F, \vartheta_c, \vartheta_D}{\arg\min} \ (1-\lambda_{AE-C})L_{CCE}(\{\vartheta_F, \vartheta_c\}) + \lambda_{AE-C}L_{MSE}(\{\vartheta_E, \vartheta_D\}) \, \lambda_{AE-C}$$

- **Optimizer:** stochastic gradient decent (SGD)

For each mini-batch classification data, after a forward pass, update $\{\vartheta_F, \vartheta_c\}$:

$$\{\vartheta_F, \vartheta_c\} \leftarrow \{\vartheta_F, \vartheta_c\} - \eta_1(1-\lambda_{AE-C})\nabla_{\{\vartheta_F, \vartheta_c\}}L_{CCE}(\{\vartheta_F, \vartheta_c\})$$

$$\vartheta_D = \vartheta_F$$

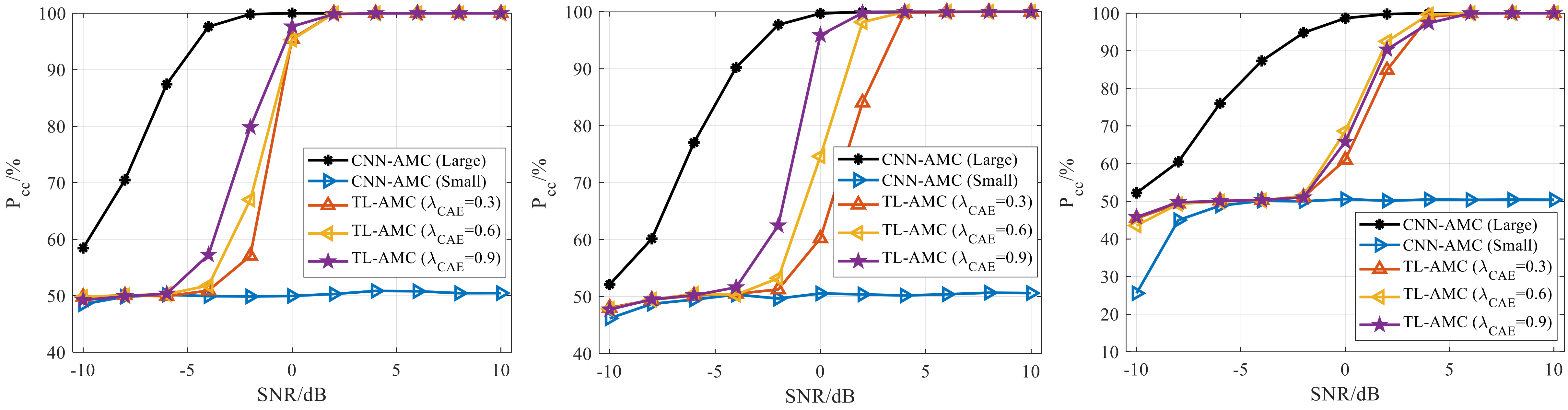
For each mini-batch auto-encoder data, after a forward pass, update $\{\vartheta_E, \vartheta_D\}$:

$$\{\vartheta_E, \vartheta_D\} \leftarrow \{\vartheta_E, \vartheta_D\} - \eta_2\lambda_{AE-C}\nabla_{\{\vartheta_E, \vartheta_D\}}L_{MSE}(\{\vartheta_E, \vartheta_D\})$$

$$\vartheta_F = \vartheta_D$$

II. Deep Learning-based AMC Methods

(3) Experimental results for TL-AMC



Simulation results of *Pcc* for different schemes with(a). $N_t = 1, N_r = 4$, (b). $N_t = 2, N_r = 4$, (c). $N_t = 4, N_r = 4$.

- TL-AMC has the similar performance with CNN, when SNR is higher than 0 dB, but its performance is far below that of CNN.

Our Work Scope in Deep Learning based SEI

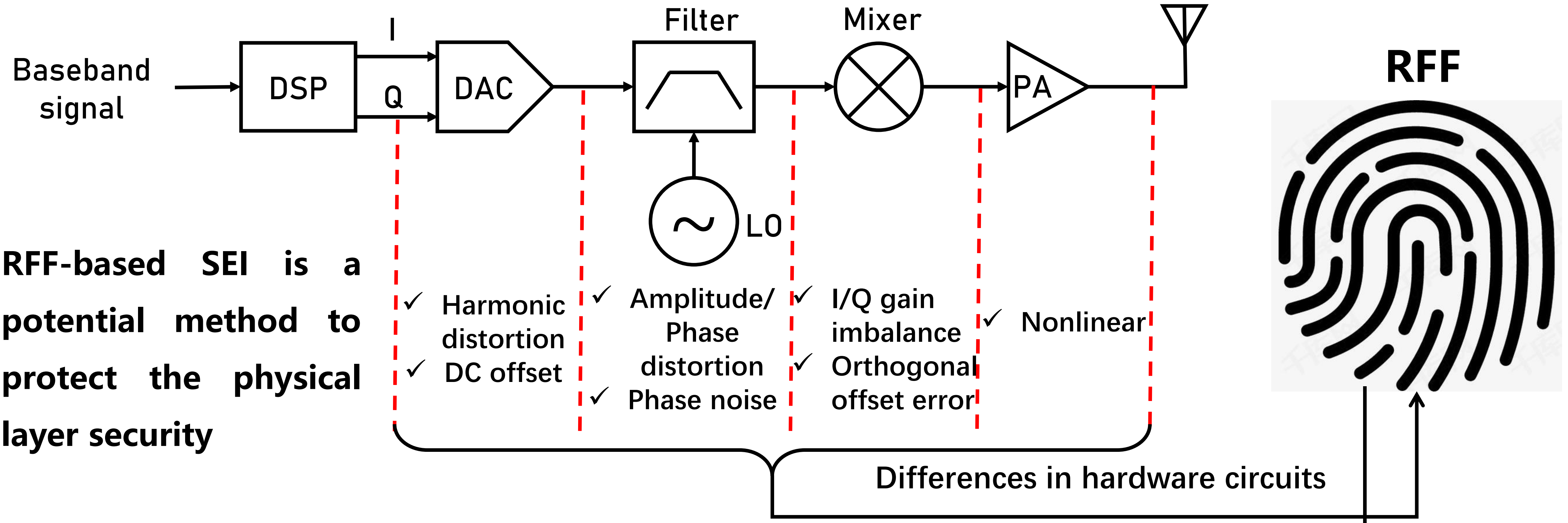
- Background
- System Model and Problem Description
- The Proposed FSL-SEI Method
  - FSL-SEI with Hybrid Metric
  - Benchmarks
- Simulation Results
  - Identification Performances
  - Feature Visualization
- Conclusion

Y. Wang, G. Gui, Y. Lin, H.-C. Wu, C. Yuen, F. Adachi, "Few-Shot Specific Emitter Identification via Deep Metric Ensemble Learning," *IEEE Internet of Things Journal*, early access, 2022.

III. Deep Learning-based SEI Methods

Background: SEI and RFF

SEI based on **radio frequency fingerprinting (RFF)**, which originates from **differences in hardware circuits of wireless devices and is parasitic in the wireless signal [1]**



- **RFF-based SEI is a potential method to protect the physical layer security**

More difficult to tamper with, More difficult to counterfeit!

[1] F. Xie, H. Wen, Y. Li, *et al.*, “Optimized coherent integration-based radio frequency fingerprinting in Internet of Things,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3967-3977, 2018.

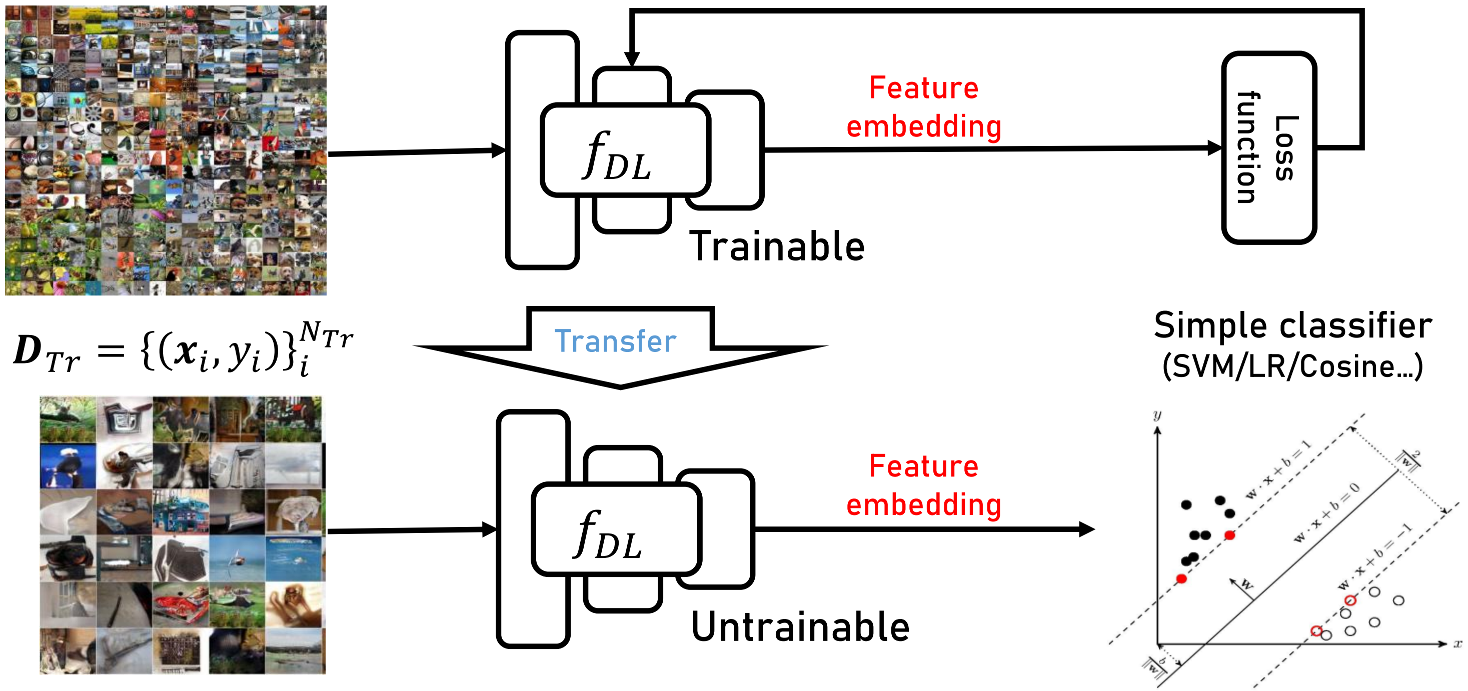
III. Deep Learning-based SEI Methods

Background: Few Shot Learning

Define: Training Dataset $\boldsymbol{D}_{Tr} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_{Tr}}$, Test Dataset $\boldsymbol{D}_{Te} = \{(\boldsymbol{x}_j)\}_{j=1}^{N_{Te}}$, and there are C classes with K samples per classes in $\boldsymbol{D}_{Tr}$, i.e., C-way-K-shot task, and $N_{Tr} = CK$; Assisted Dataset $\boldsymbol{D}_{As} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_{As}}$ is needed, and $\boldsymbol{D}_{As} \cap \boldsymbol{D}_{Tr} = \emptyset, \forall y_n \notin \{y_i\}_i^{N_{Tr}}$

FSL = Feature embedding + Simple classifier [1]

$\boldsymbol{D}_{As} = \{(\boldsymbol{x}_n, y_n)\}_n^{N_{As}}$



- **A good feature embedding**
  - Generative model
  - Metric model (learn to compare)
  - Meta model (learn to learn)
- **Transfer learning (TL) vs. FSL**
  - Existing FSL methods (Pseudo FSL) can be considered as a simple TL
  - FSL focuses on how to construct a good feature embedding rather than how to transfer knowledge

[1] Y. Tian, Y. Wang, D. Krishnan, et al., “Rethinking few-shot image classification: a good embedding is all you need?” in *ECCV 2020: 16th European Conference*, Glasgow, UK, Aug. 23–28, 2020, pp. 266-282.

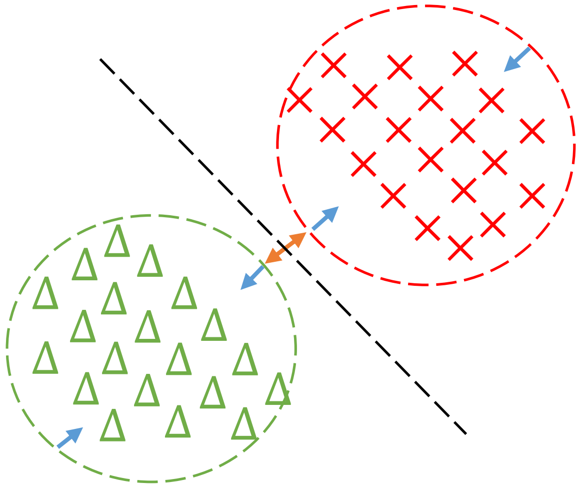
III. Deep Learning-based SEI Methods

Background: Metric Learning

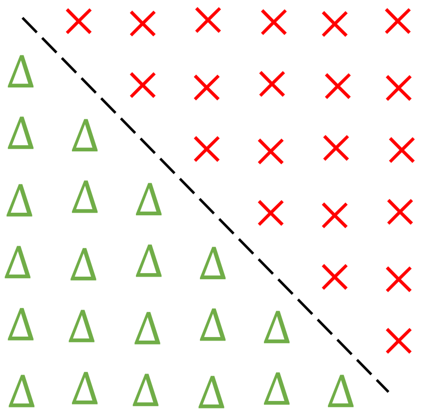
Metric learning: **Distinguish different individuals** rather than **identify their categories**

Discriminative features

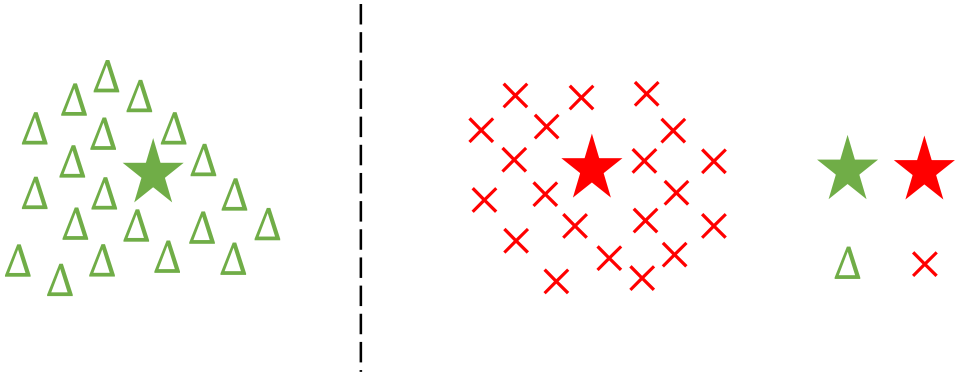
- Enlarge inter-class distance
- Narrow inner-class distance



Separable features



- **Simple classifier:**

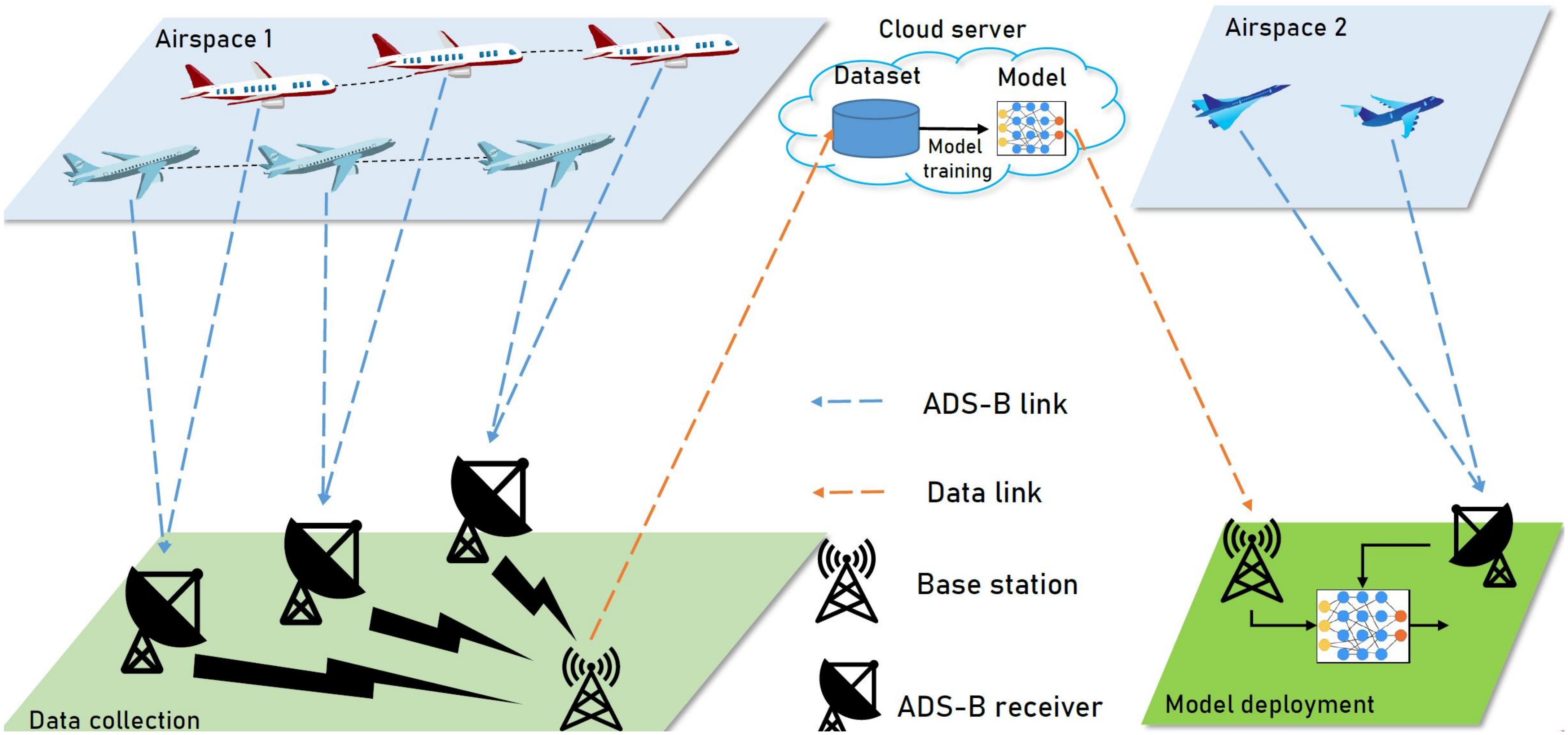


Fearture from few-shot samples

Fearture from test samples

III. Deep Learning-based SEI Methods

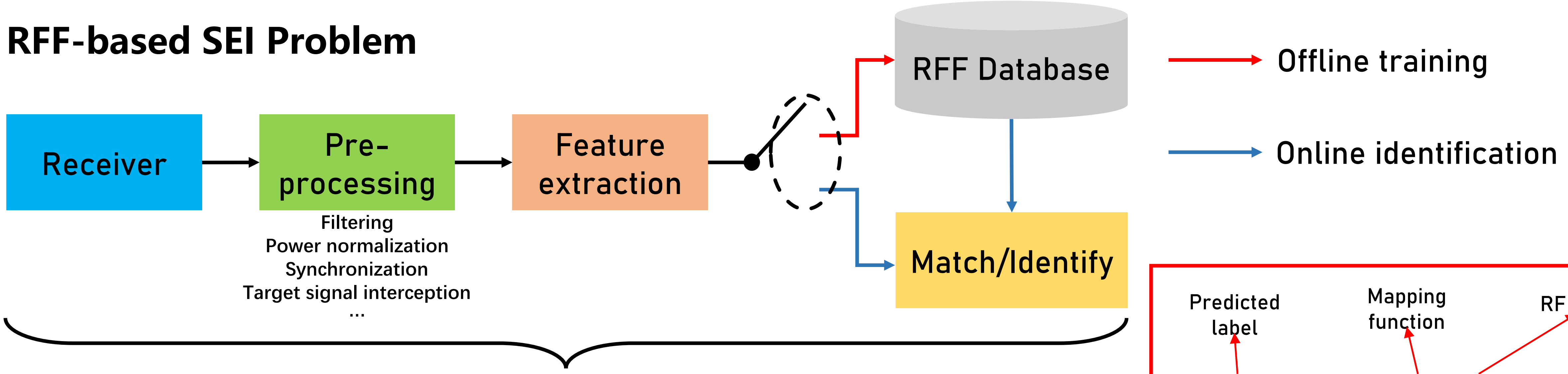
System Model and Problem Formulation



III. Deep Learning-based SEI Methods

System Model and Problem Formulation

- **RFF-based SEI Problem**



SEI is generally defined as **multi-category classification problem**

Predicted label
Mapping function
RF data

$$\hat{y} = \arg\max_{y \in Y} f(y|\boldsymbol{x})$$

Real label
All categories

- **FS-SEI Problem**

Assisted Dataset $\boldsymbol{D}_{As} = \{(\boldsymbol{x}_n, y_n)\}_n^{N_{As}}$: Massive historical ADS-B data containing N classes (>10^5 samples)

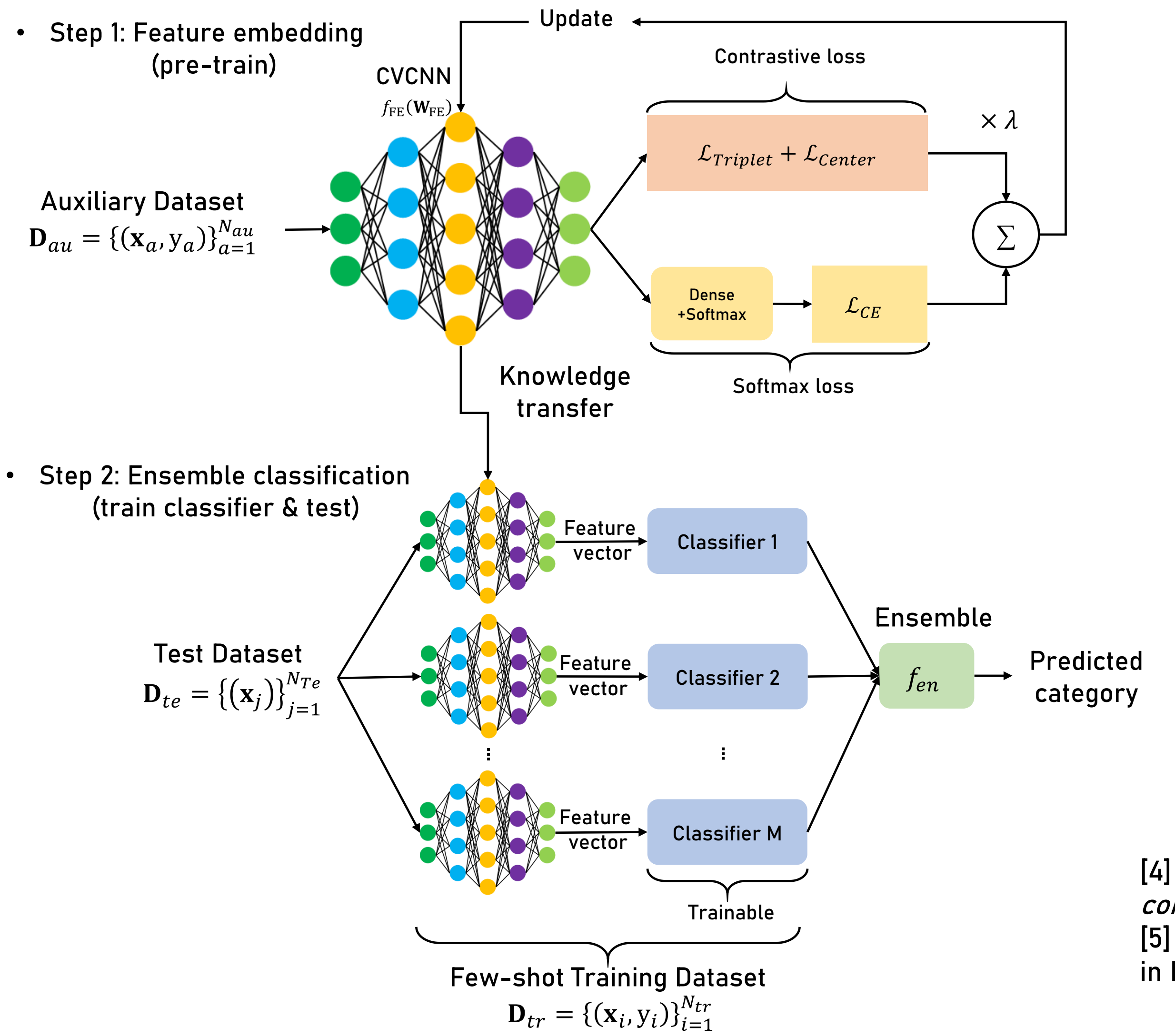
Training Dataset $\boldsymbol{D}_{Tr} = \{(\boldsymbol{x}_i, y_i)\}_i^{N_{Tr}}$: Few ADS-B data from new C classes with K samples per classes

Test Dataset $\boldsymbol{D}_{Te} = \{(\boldsymbol{x}_j)\}_{j=1}^{N_{Te}}$

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathcal{L}[f(\boldsymbol{x}_n; \mathbf{W}), y_n] \quad \sum \quad \mathbf{W}_c^* = \arg\min_{\mathbf{W}_{sc}} \mathcal{L}_c\{f_c[f(\boldsymbol{x}_i; \mathbf{W}^*); \mathbf{W}_c]\} \quad \sum \quad \hat{y}_j = f_c[f(\boldsymbol{x}_j; \mathbf{W}^*); \mathbf{W}_c^*]$$

III. Deep Learning-based SEI Methods

- **Hybrid Metric-based Joint Separable and Discriminative Feature Embedding**

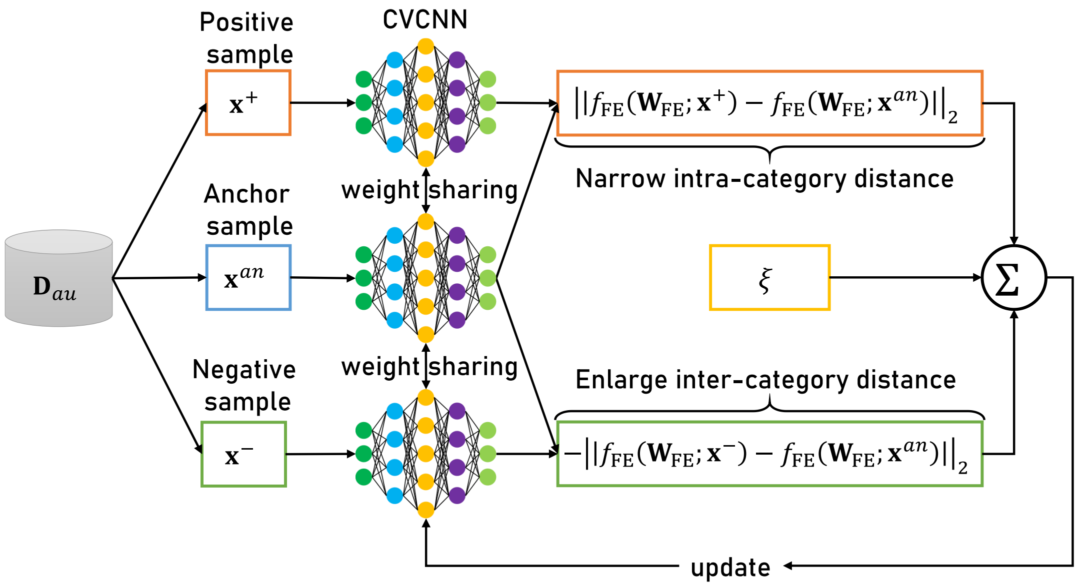


$$\boxed{\mathcal{L}_{HM} = \mathcal{L}_{CE} + \lambda(\mathcal{L}_{Triplet} + \mathcal{L}_{Center})}$$

- ✓ $\mathcal{L}_{CE}$: Cross-entropy (CE) loss function for separable feature representation
- ✓ $\mathcal{L}_{Triplet}$: Triplet loss function for both enlarge inter-class distance and narrowing inner-class distance [4]
- ✓ $\mathcal{L}_{Center}$: Center loss function for narrowing inner-class distance [5]

[4] X. Dong, J. Shen, "Triplet loss in Siamese network for object tracking," in *European conference on computer vision (ECCV)*, pp. 459-474, 2018.
[5] Y. Wen, et al., "A discriminative feature learning approach for deep face recognition," in European conference on computer vision. Springer, Cham, pp. 499-515, 2016.

Triplet network and triplet loss



Loss function:

$$\mathcal{L}_{Triplet} = \mathbb{E}[\|f_{CVCNN}(\boldsymbol{x}^{an};\mathbf{W}) - f_{CVCNN}(\boldsymbol{x}^{+};\mathbf{W})\|_2$$

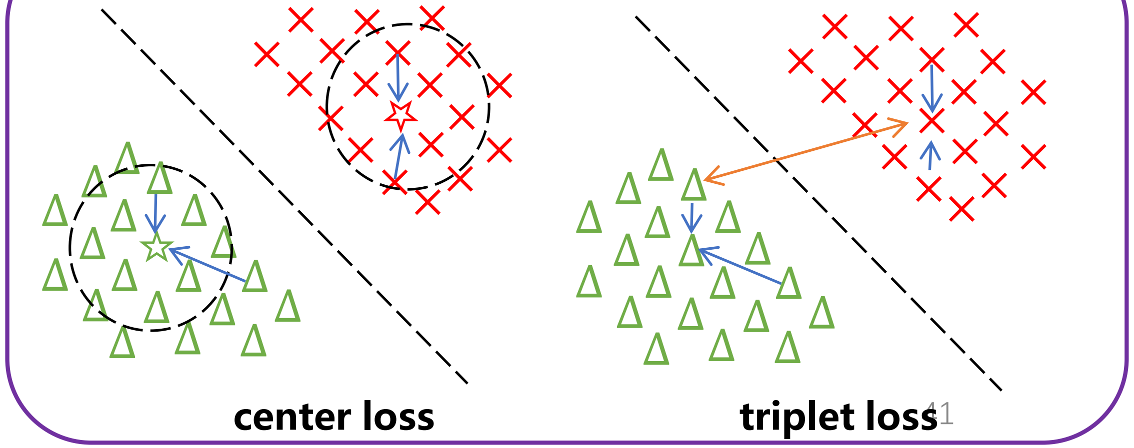
- **Center loss**

$$\mathcal{L}_{\text{center}} = \frac{1}{2}\sum_{i=1}^{M}\left\|f_{CVCNN}(\boldsymbol{x}_i;\mathbf{W}) - \boldsymbol{c}_{\boldsymbol{y}_i}(\mathbf{W}_c)\right\|_2^2$$

$$\frac{\partial \mathcal{L}_{\text{center}}}{\partial \mathbf{W}} = \frac{\partial f_{CVCNN}(\boldsymbol{x}_i;\mathbf{W})}{\partial \mathbf{W}} - \boldsymbol{c}_{\boldsymbol{y}_i}$$

$$\Delta \boldsymbol{c}_{\boldsymbol{y}_i} = \frac{\sum_{i=1}^{M}\delta(\boldsymbol{y}_i = j)\cdot(\boldsymbol{c}_j - \boldsymbol{x}_i)}{1 + \sum_{i=1}^{M}\delta(\boldsymbol{y}_i = j)}$$

Difference between triplet loss and center loss



Benchmarks

- **Instananeous features:** traditional instananeous feature, extrated from few-shot samples [6]
- **CVCNN (few-shot samples):** CVCNN, diretly trained on few-shot samples.
- **CVCNN:** CVCNN, trained on $\boldsymbol{D}_{As} = \{(\boldsymbol{x}_n, y_n)\}_n^{N_{As}}$ with CE loss function, and then applied into few-shot task.
- **Siamese CVCNN:** CVCNN with siamese structure for roubust signal feature extraction [7].
- **SR2CNN:** It was applied into zero-shot signal recognition, which consists of classifier, auto-encoder and center loss. Here, we use its signal feature representation part for comparison rather than zero-shot recognition scheme [8].

[6] W. E. Cobb, E. D. Laspe, R. O. Baldwin, M. A. Temple and Y. C. Kim, "Intrinsic Physical-Layer Authentication of Integrated Circuits," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 14-24, 2012.
[7] Z. Langford, L. Eisenbeiser, M. Vondal, "Robust signal classification using siamese networks," in *ACM Workshop on Wireless Security and Machine Learning*, pp. 1-5, 2019.
[8] Y. Dong, X. Jiang, H. Zhou, et al., "SR2CNN: Zero-Shot Learning for Signal Recognition," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2316-2329, 2021.

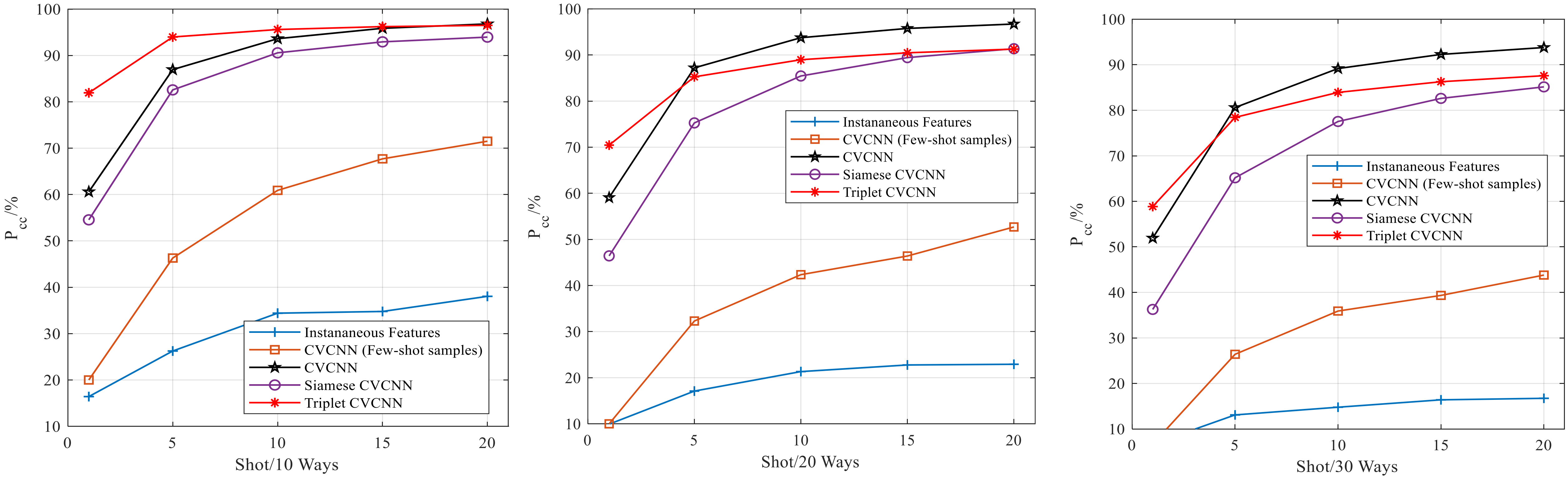
III. Deep Learning-based SEI Methods

Simulation parameters

Sampling rate	50 M Samples/s
Frequency	1090MHz
Bandwidth	10MHz
Gain	30dB
Signal format	IQ
The number of aircrafts in assisted dataset	90
The number of samples in assisted dataset	200~500
Training vs. Validation	7:3
The number of aircrafts in few-shot training/test dataset	10~30
The number of samples in few-shot training dataset	1~20 samples per classes
The number of samples in test dataset	200 samples per classes
Channel	≈LOS
margin m and threshold λ	5, 0.01
Optimizer	ADAM with default parameters

III. Deep Learning-based SEI Methods

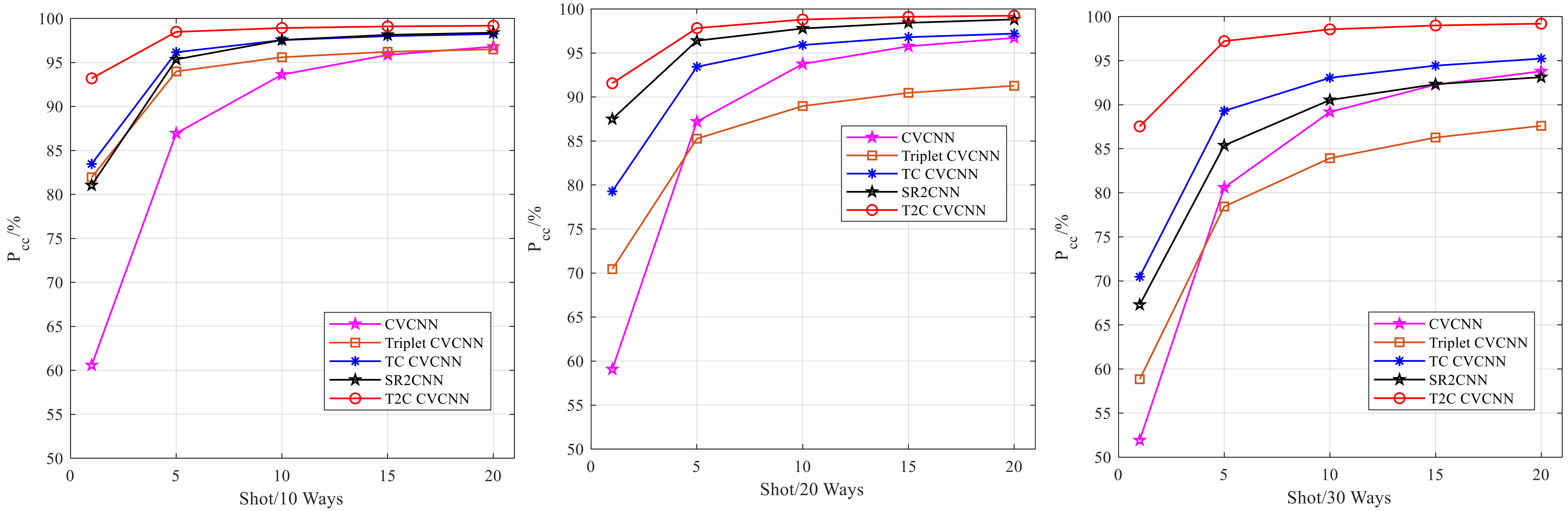
Identification performance comparison of single metric



- CVCNN (separable feature) and Triplet CVCNN (discriminative feature) have their own advantages
- Single separable or discriminative feature can not work well
- Triplet CVCNN has better performance than Siamese CVCNN

III. Deep Learning-based SEI Methods

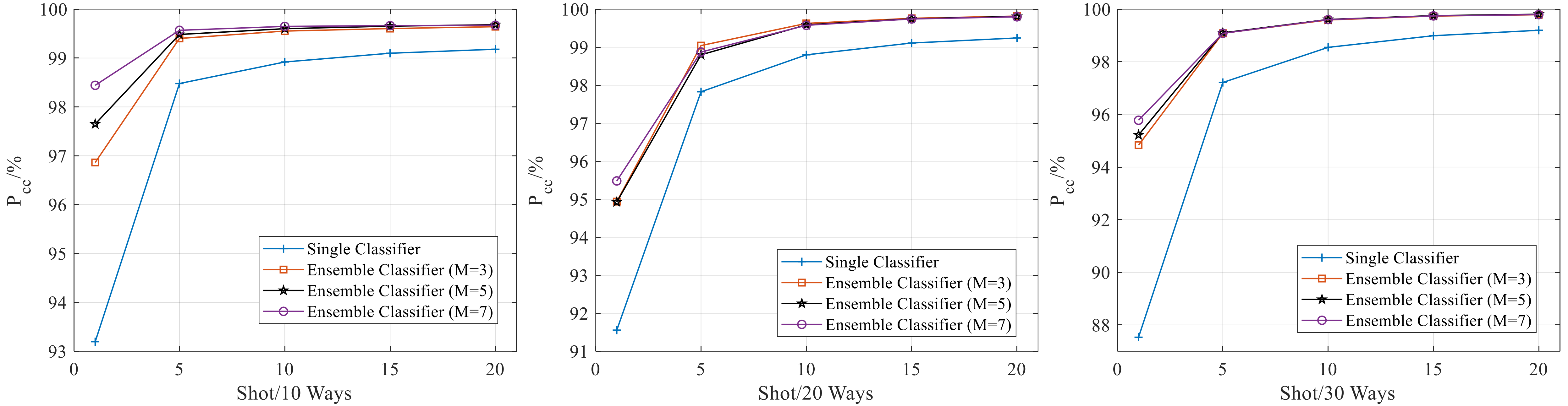
Identification performance comparison of hybrid metric



- TC CVCNN is based on triplet loss and CE loss
- T2C CVCNN is CVCNN based on triplet loss, center loss and CE loss

III. Deep Learning-based SEI Methods

Identification performance comparison: Single Classifier vs. Ensemble Classifier

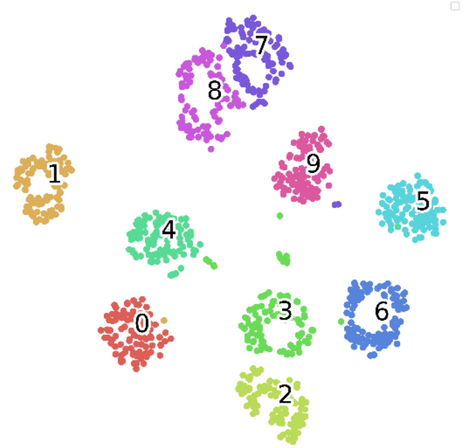


- Ensemble classifier performs better than single classifier
- The more base classifiers in ensemble learning, the better the performance, which is obvious in the one-shot scenario

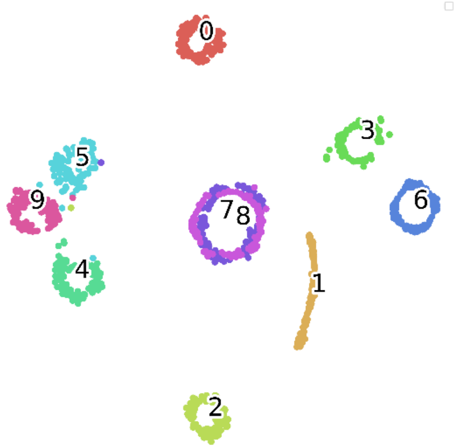
III. Deep Learning based SEI Methods

Feature Visualization

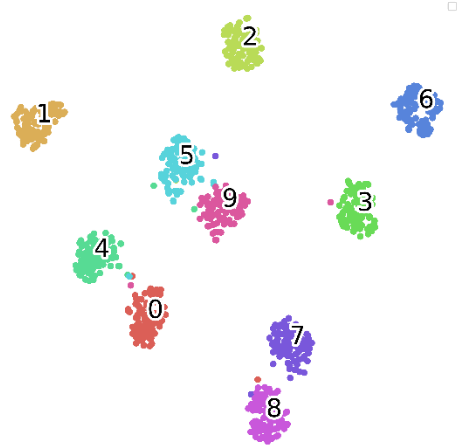
CVCNN



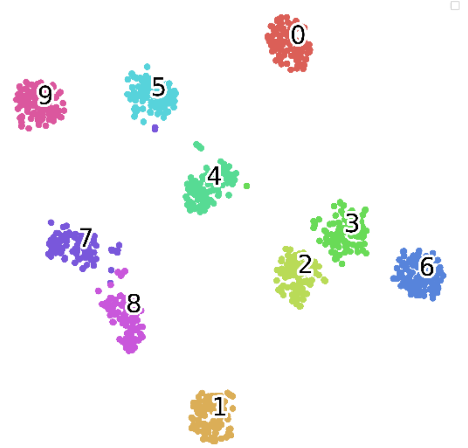
Siamese CVCNN



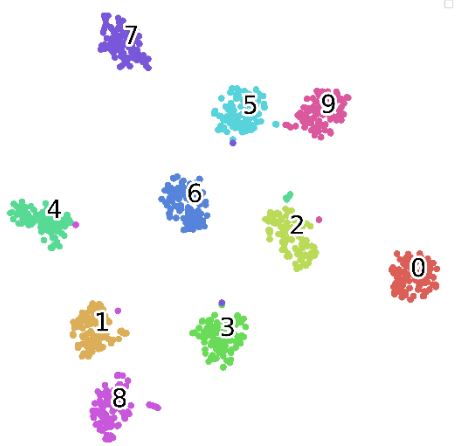
Triplet CVCNN



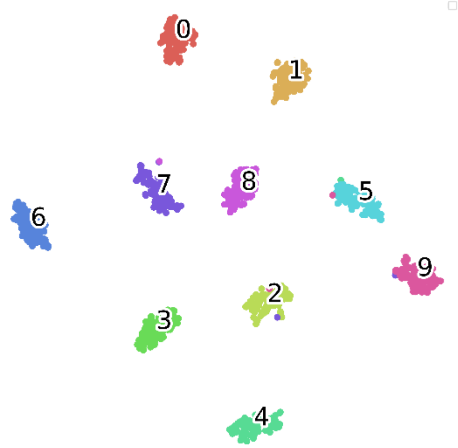
TC CVCNN



SR2CNN



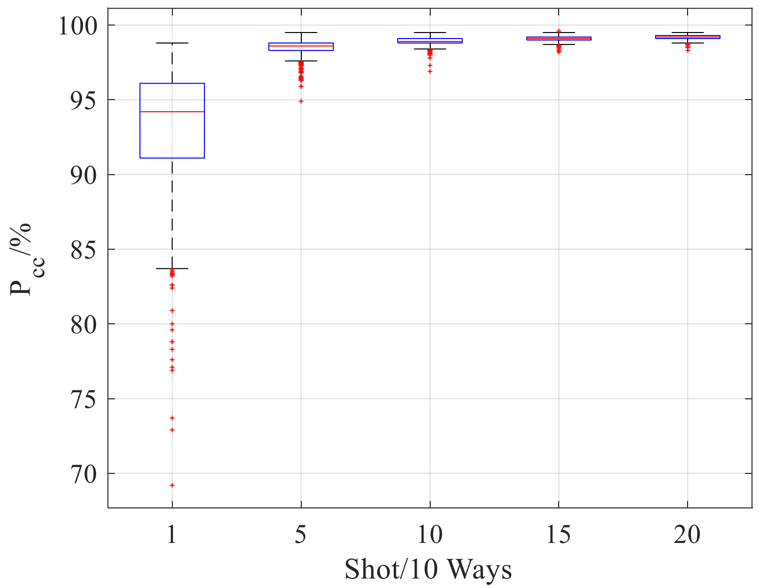
T2C CVCNN

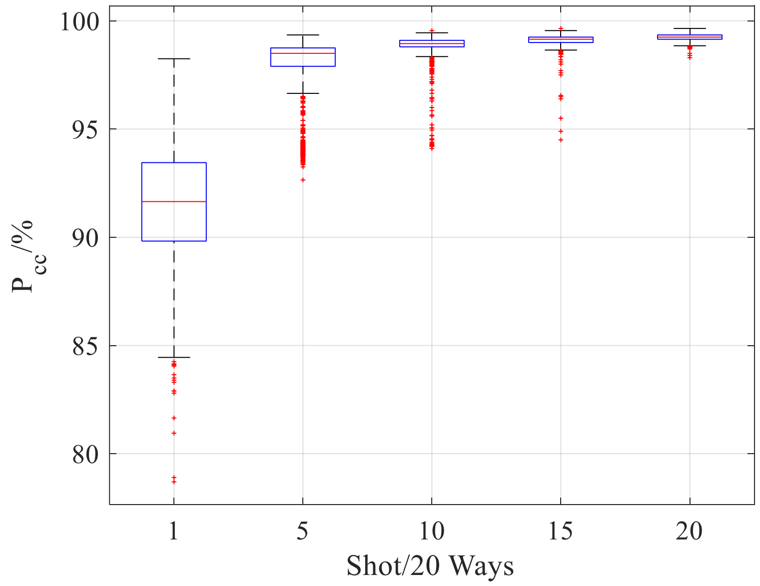


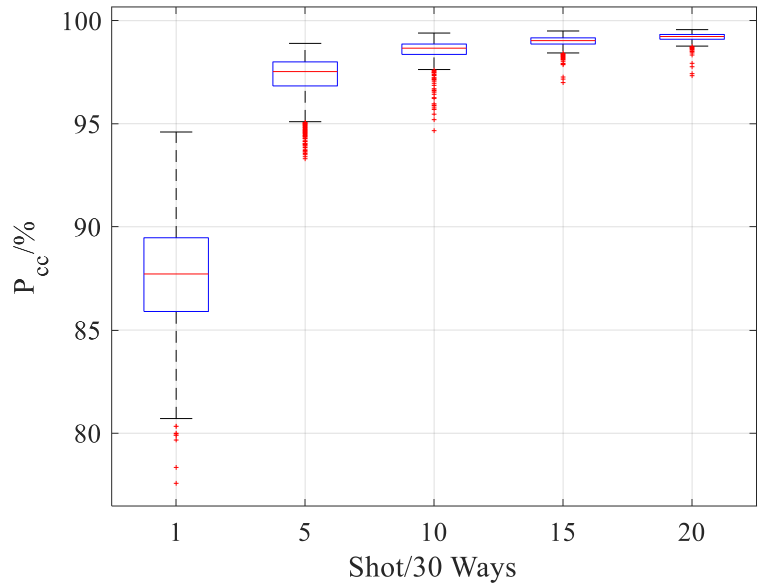
- ✓ The feature distance between emitters of the same category is smaller
- ✓ The feature distance between emitters of different categories is larger

III. Deep Learning-based SEI Methods

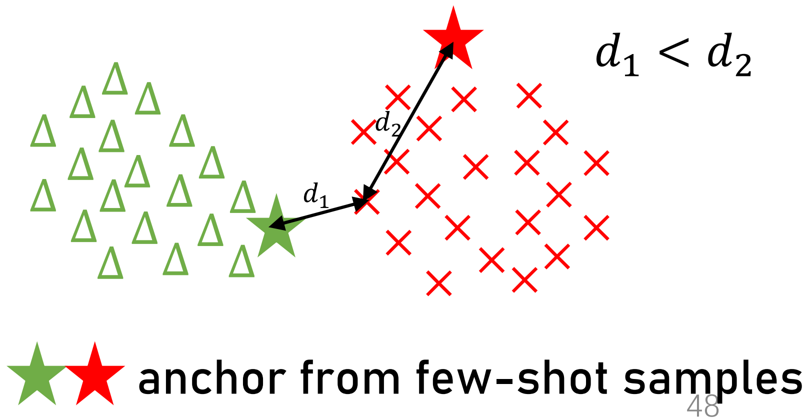
The influence of few-shot sample quality







- Few-shot samples are as anchors of the corresponding classes
- FSL-SEI is to measure the distance between test samples and anchors
- **"Anchor"** is important for FSL-SEI, and a deviated anchor can bring about catastrophic results
- The smaller the number of samples, the more serious the impact



Conclusion

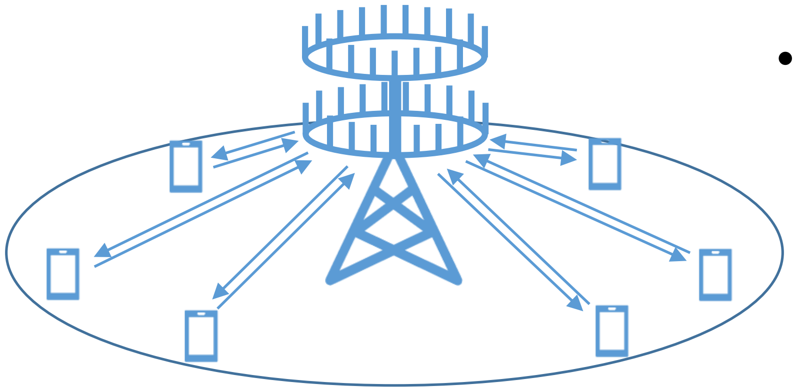
- We proposed an effective FS-SEI method for aircraft identification based on metric learning and ensemble learning. Simulation results demonstrated the effectiveness of our proposed FS-SEI. Feature visualization also showed the compact intra-category distance and separable inter-category distance in the features extracted by our proposed method.
- We also revealed the impact of noisy samples on the stability of the proposed algorithm, and we expect to use some schemes, such as attention mechanism [9], to reduce the impact of sample quality on identification performance in the future works.
- The corresponding codes can be downloaded from GitHub: https://github.com/BeechburgPieStar/Few-Shot-Specific-Emitter-Identification-via-Deep-Metric-Ensemble-Learning

[9] T. Gao, Z. Liu and M. Sun, "Hybrid attention-based prototypical networks for noisy few-shot relation classification", in *AAAI Conference on Artificial Intelligence*, 2019, pp. 6407-6414.

Ⅳ. Deep Learning-based CSI Inferring Methods

Background of CSI inferring technology

- **Challenges in FDD Massive MIMO system** - The acquisition of downlink CSI is a very challenging task for frequency division duplexing (FDD) massive MIMO systems due to high overheads associated with downlink training and uplink feedback.



- **Two observations**:
  (1) A small angular spread (AS) between BS and users;
  (2) There exists angular reciprocity between uplink and downlink.

- **Why deep learning (DL)**
  (1) Inherent characteristics of wireless channels can be captured by DL;
  (2) Deep learning can provide solutions for the problems that have no clear analytical model;
  (3) Efficient parallel computing methods reduce the complexity.

IV. Deep Learning-based CSI Inferring Method

Our Work Scope in Deep Learning-based CSI inferring

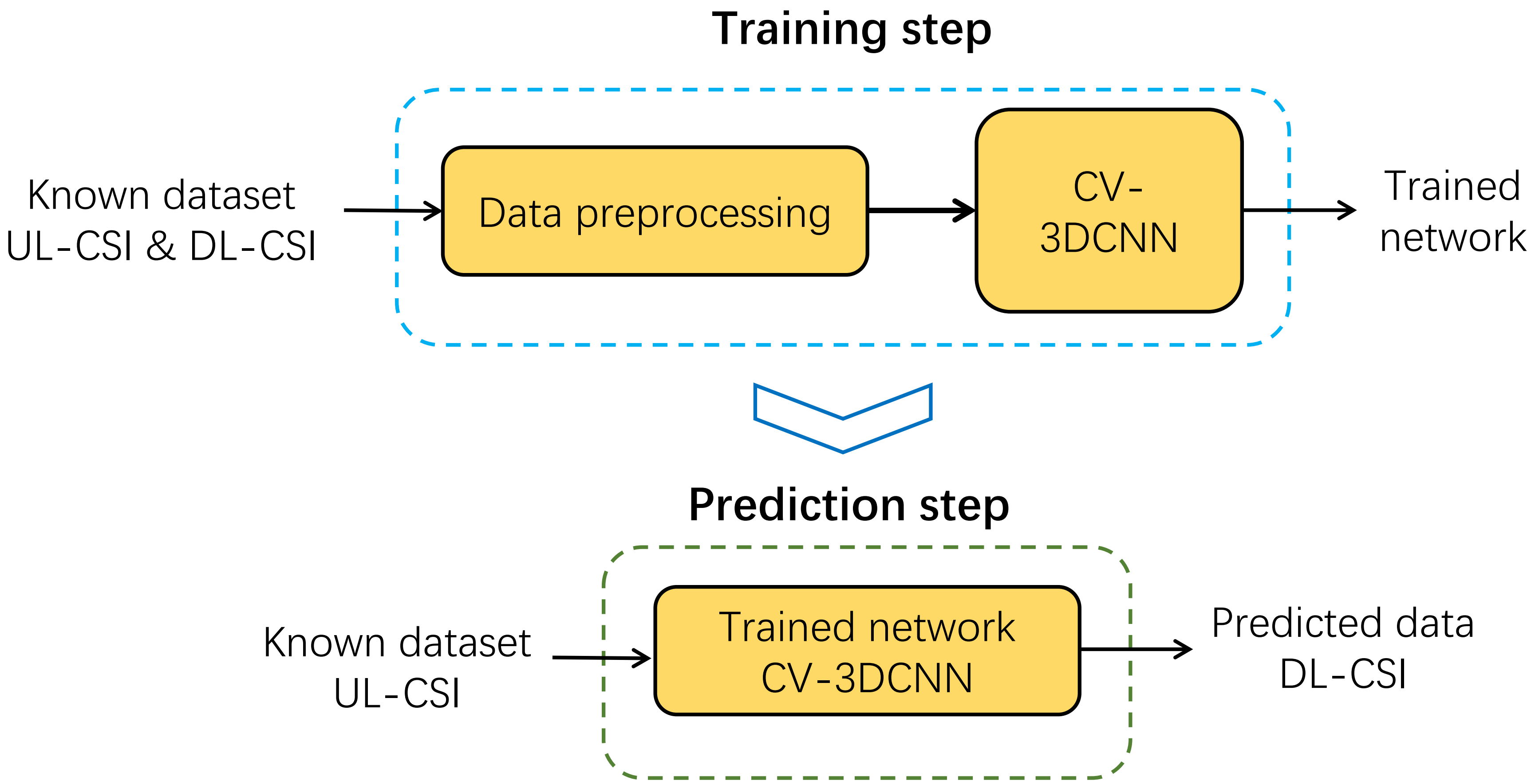
- Complex-valued Deep Learning for CSI prediction in FDD massive MIMO System
- Fully Convolutional Network for CSI limited feedback in FDD massive MIMO System
- Transfer learning for CSI limited feedback in FDD massive MIMO System

CV-3DCNN: Complex-valued Deep Learning for CSI prediction in FDD massive MIMO System

Y. Zhang, J. Wang, J. Sun, B. Adebisi, H. Gacanin, G. Gui, F. Adachi, "CV-3DCNN: Complex-Valued Deep Learning for CSI Prediction in FDD Massive MIMO Systems," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 266-270, 2021.

IV. Deep Learning-based CSI Inferring Methods

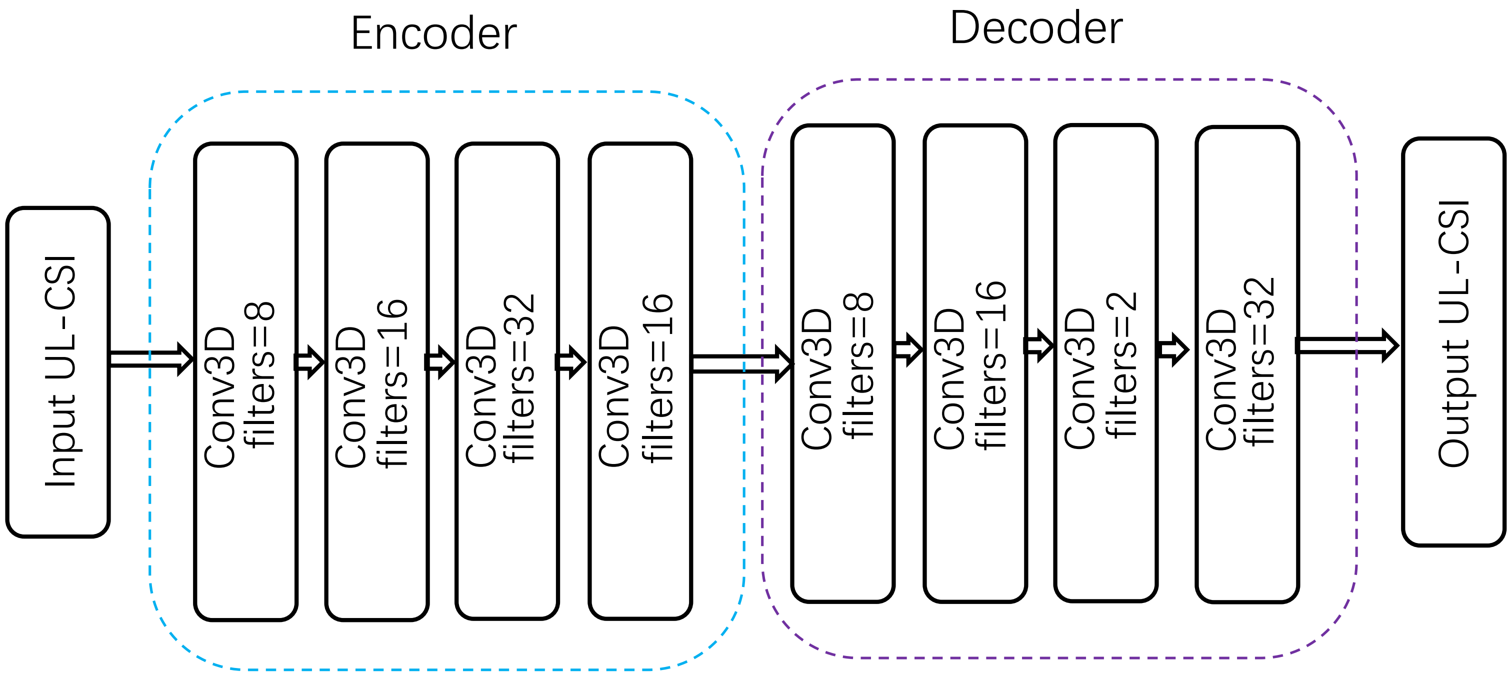
(1) The steps of the prediction scheme



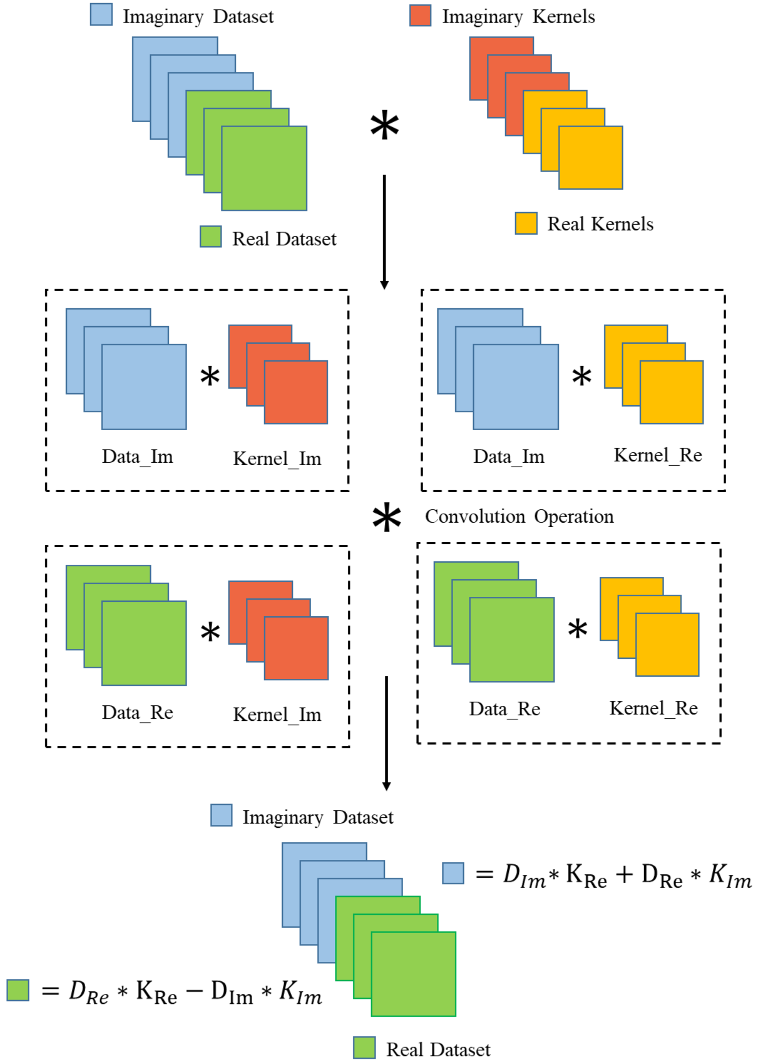
The working mechanism of the proposed CSI prediction scheme

IV. Deep Learning-based CSI Inferring Methods

(2) The structure of our proposed CV-3DCNN



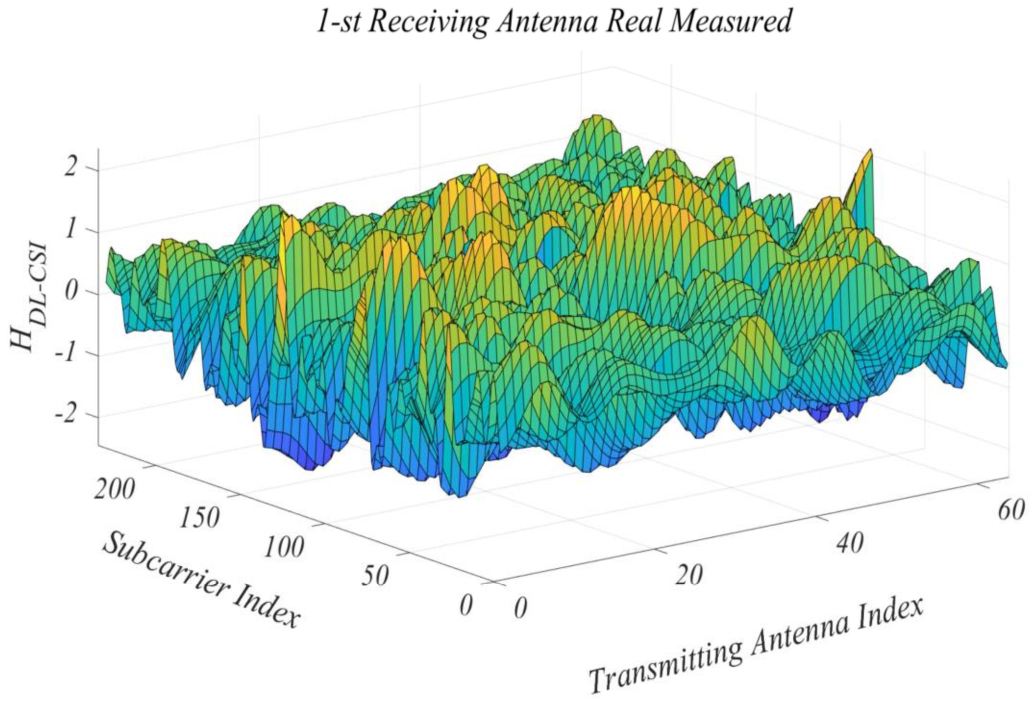
The structure of CV-3DCNN.

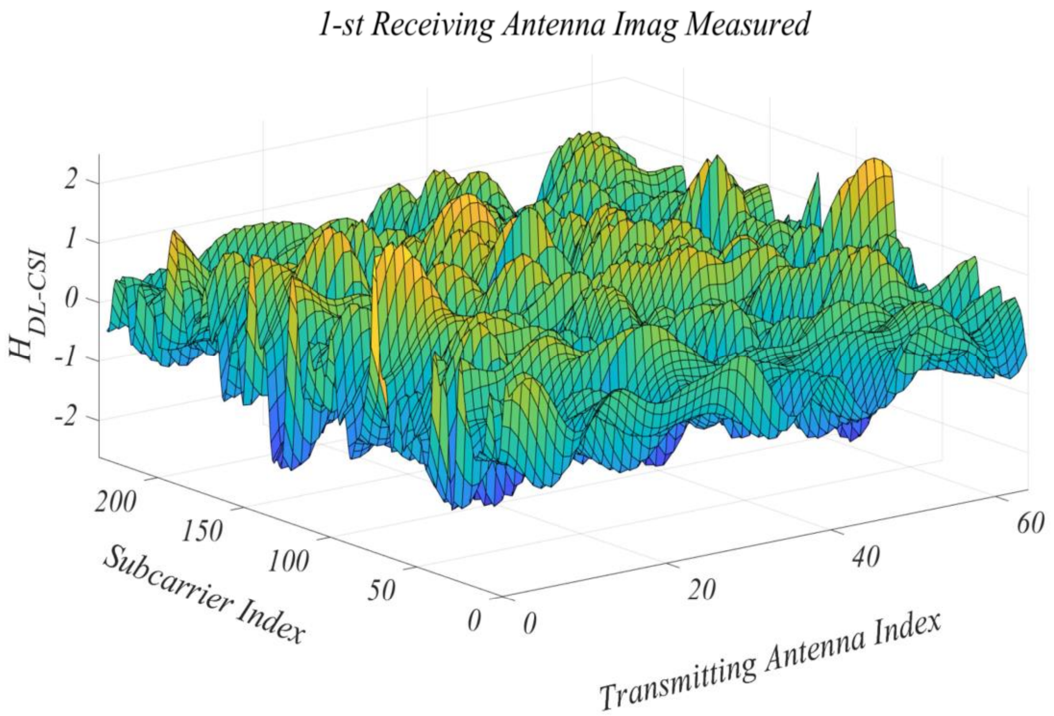


Training Tips: The figure on the right shows the complex-valued (CV) convolution operation.

IV. Deep Learning-based CSI Inferring Methods

(3) Experimental results: Measured and predicted data of first receiver antenna

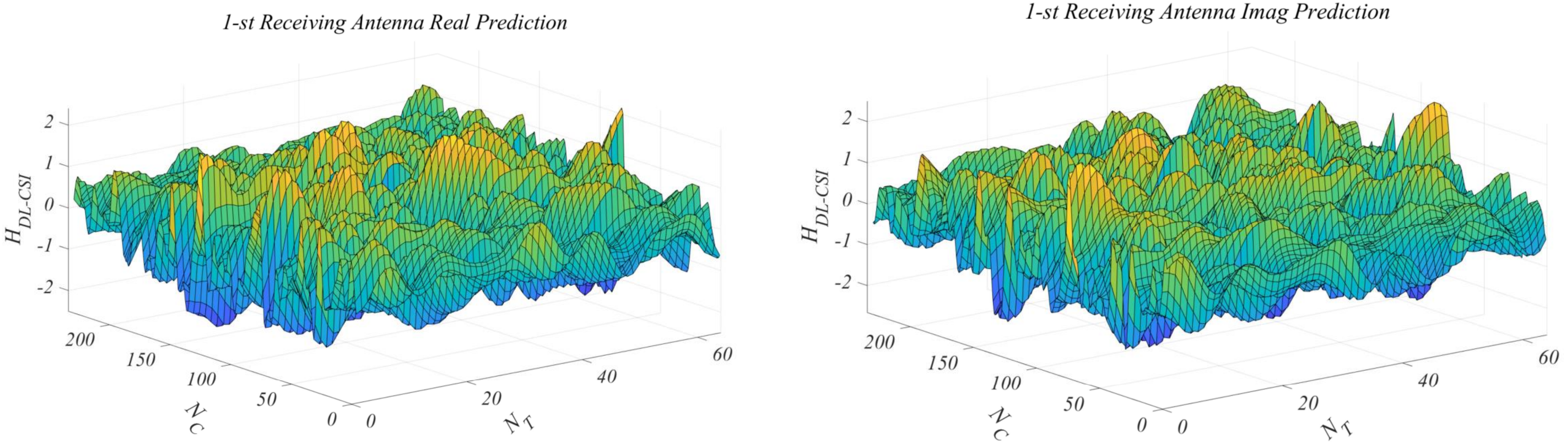




Schematic diagram of measured downlink CSI

IV. Deep Learning-based CSI Inferring Methods

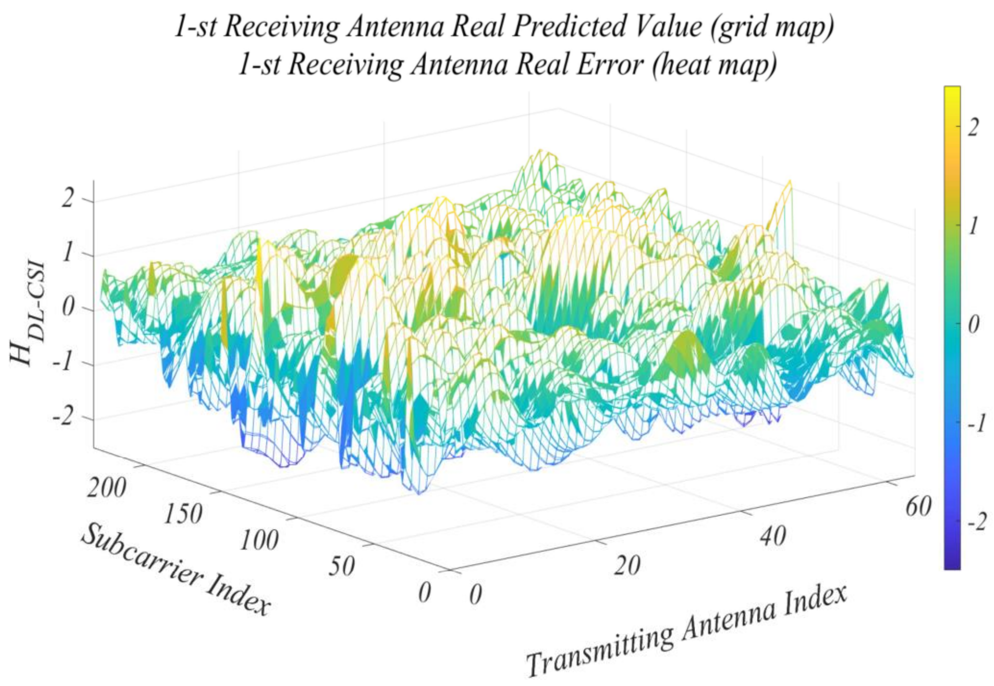
(4) Experimental results: Measured and predicted data of first receiver antenna

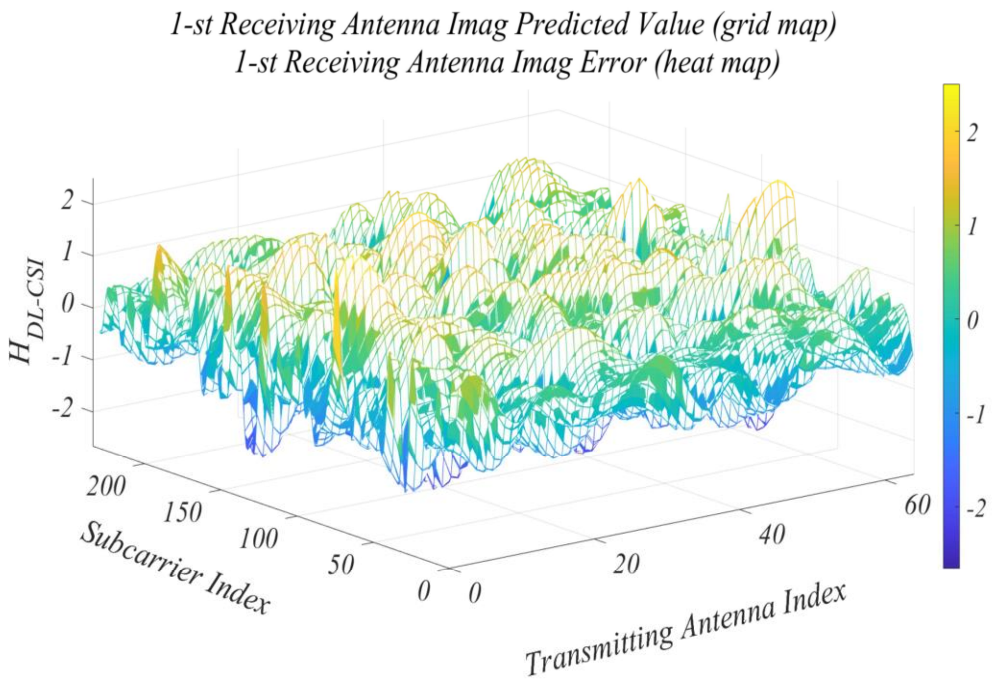


Schematic diagram of predicted downlink CSI method

IV. Deep Learning-based CSI Inferring Methods

(5) Experimental results: Error heat map of first receiver antenna





Schematic diagram of error between measured and predicted downlink CSI method.

(6) Experimental results for CV-3DCNN

	Real-domain Neural Network			CV-3DCN		
No.	NMSE	NMSE(dB)	ρ	NMSE	NMSE(dB)	ρ
1	0.0146	-18.3565	0.9970	0.0051	-22.9243	0.9989
2	0.0138	-18.6012	0.9972	0.0070	-21.5490	0.9985
3	0.0144	-18.4164	0.9970	0.0035	-24.5593	0.9993
4	0.0099	-20.0436	0.9980	0.0041	-23.8722	0.9991
5	0.0122	-19.1364	0.9975	0.0025	-26.0206	0.9995
6	0.0116	-19.3554	0.9976	0.0030	-25.2288	0.9994
Mean	0.0128	-18.9849	0.9974	0.0042	-24.0257	0.9991

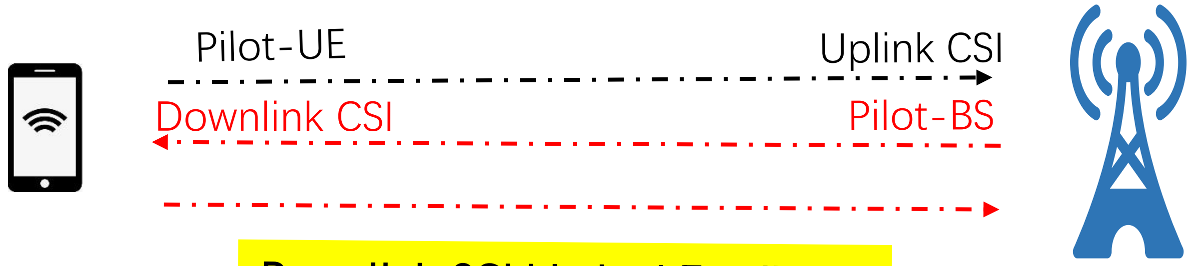
Compared with real-domain neural network, the NMSE performance of our proposed CV-3DCNN improved about 26.55%.

Fully Convolutional Network for CSI limited feedback in FDD massive MIMO System

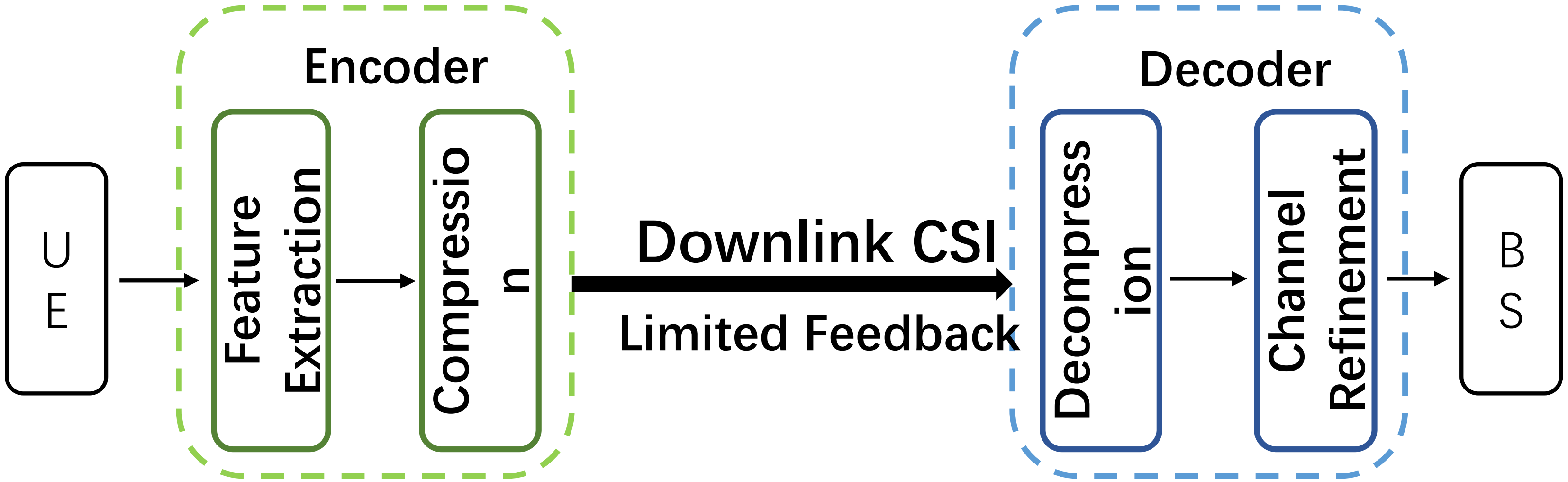
G. Fan, J. Sun, G. Gui, H. Gacanin, B. Adebisi, T. Ohtsuki, "Fully Convolutional Neural Network-Based CSI Limited Feedback for FDD Massive MIMO Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 672-682, 2022.

IV. Deep Learning-based CSI Inferring Methods

(1) The steps of the Downlink CSI Limited Feedback



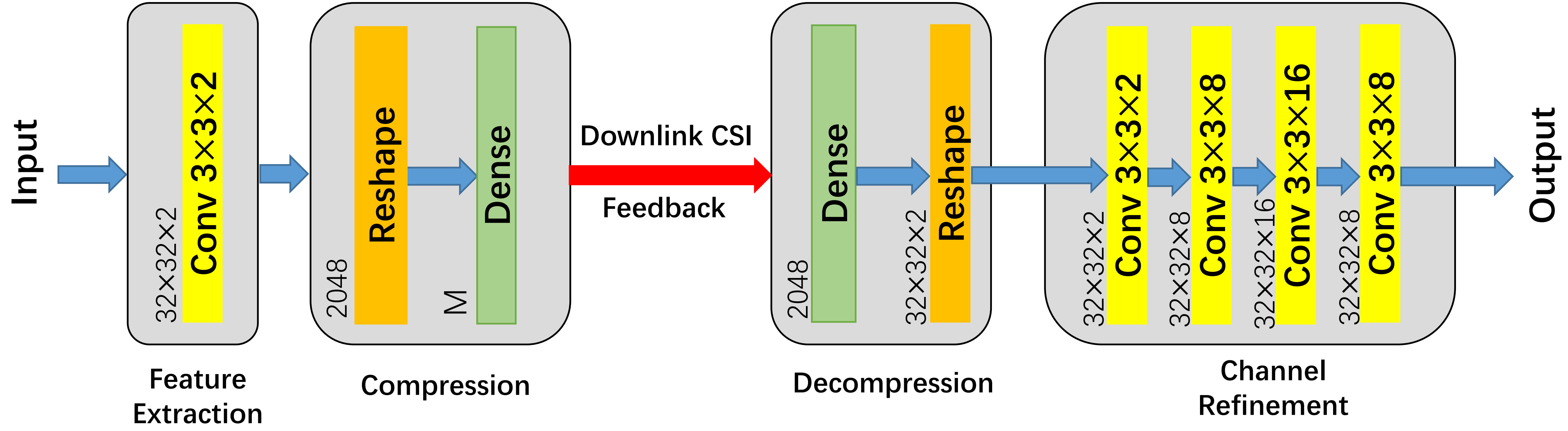
Downlink CSI Limited Feedback.



The working mechanism of downlink CSI limited feedback.

IV. Deep Learning-based CSI Inferring Methods

(2) Existing network structure



Existing network structure: CsiNet [1]

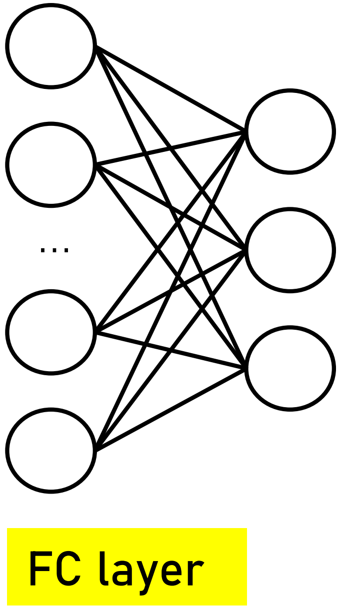
- **Feature Extraction**: Extracting features of Downlink CSI and generate two feature maps.
- **Compression：** Compressing the Downlink CSI and generate the codeword.
- **Decompression：** Mapping the codeword back into the Downlink CSI .
- **Channel Refinement：** Continuously refining the reconstructed Downlink CSI .

[1] C. K. Wen, W. T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

IV. Deep Learning-based CSI Inferring Methods

(3) Exiting problems

- Too many parameters in Fully Connected layer (FC layer)
  → high time complexity and space complexity



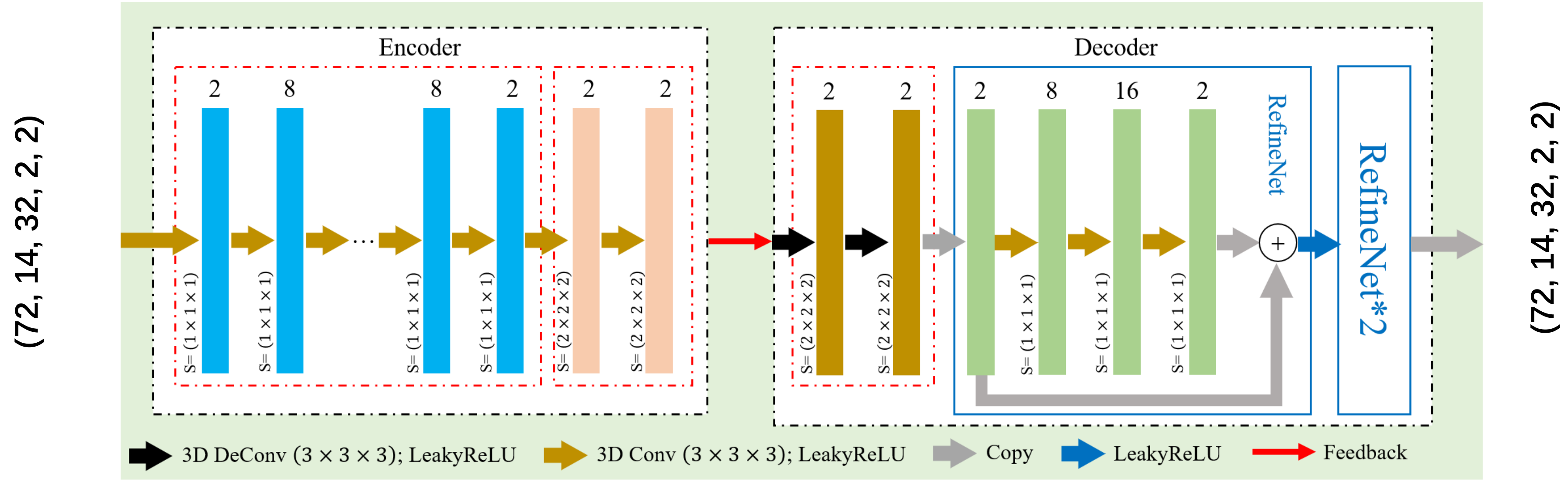
- 4G CSI dimension: $32 \times 32 \times 2$ →2,048 [1]
- If CR=1/4 → 2,048 : 512 → 1,048,576 → 2,097,152
- If CR=1/64 → 2,048 : 32 → 65,536 → 131,072

- 5G CSI dimension : $72 \times 28 \times 32 \times 2$ →129,024
- If CR=1/8 → 129,024 : 16128 → 2,080,899,072 → 4,161,798,144

- The research is carried under 4G channel models, and has not been applied to 5G yet
  → the simple **CsiNet** needs to be modified

IV. Deep Learning-based CSI Inferring Methods

(4) Our proposed FullyConv network for CSI limited feedback



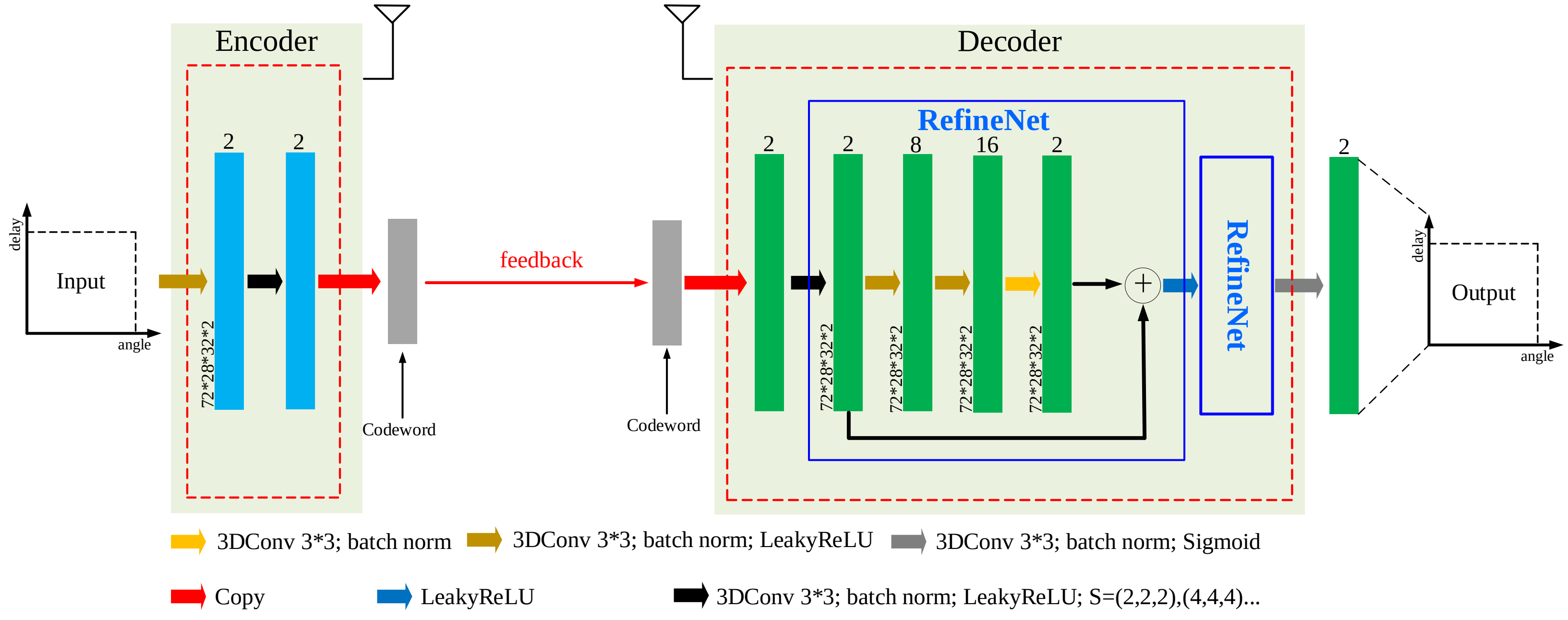
Training Tips:

- ✓ 5G Downlink CSI matrix: (72, 14, 32, 2, 2)

  where 72: Subcarriers, 14: OFDM symbols, 32: Transmitting antennas, 2: receiving antennas, 2: real and imaginary part
- ✓ Feature Extraction module： composed of 7 3Dconv layers → the ability of feature extraction is stronger
- ✓ Compression and Decompression modules ： 3DConv layers and 3DDeConv layers
- ✓ Channel Refinement module ： 2RefineNet blocks → 3RefineNet blocks, refining Downlink CSI

IV. Deep Learning-based CSI Inferring Methods

(5) Baseline model: CsiNet_5G



The structure of CsiNet_5G

- CsiNet cannot be applied to the current 5G Downlink CSI, so we modify CsiNet to CsiNet 5G for 5G downlink CSI.
- The biggest difference between CsiNet_5G and CsiNet is that all convolution operations are 3DConv.
- The dimension of downlink CSI is too high to use FC layers, so the compression and decompression modules of CsiNet_5G use convolutional layers.

IV. Deep Learning-based CSI Inferring Methods

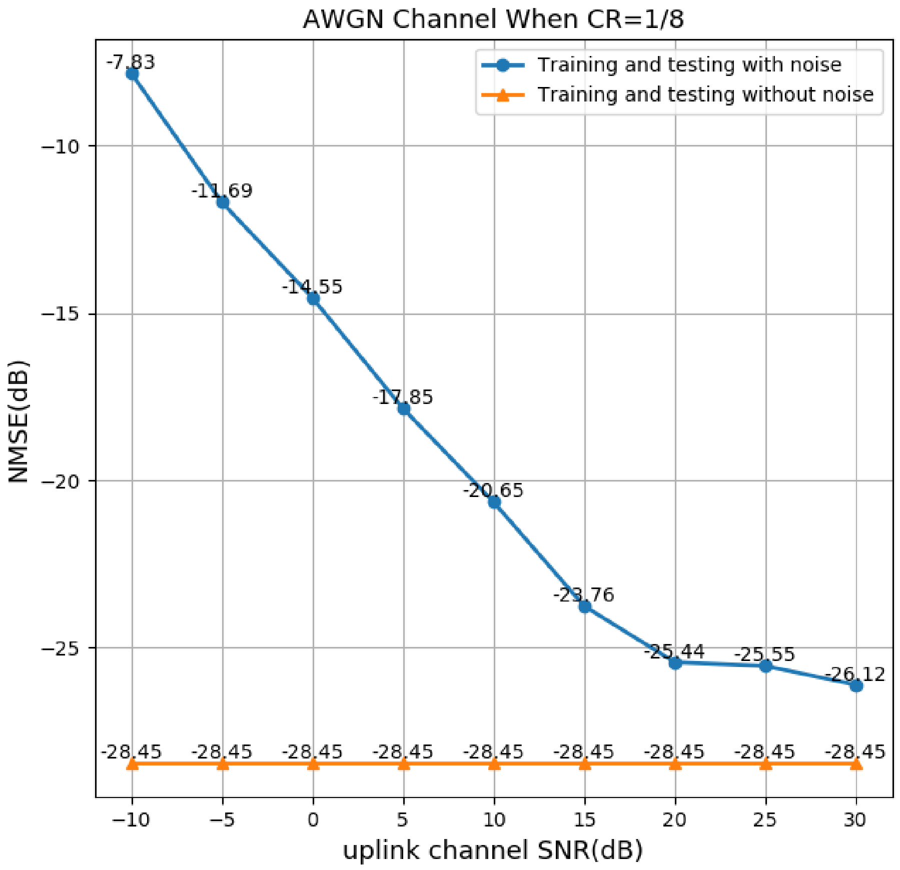
(6) Experimental results of FullyConv compared with baseline

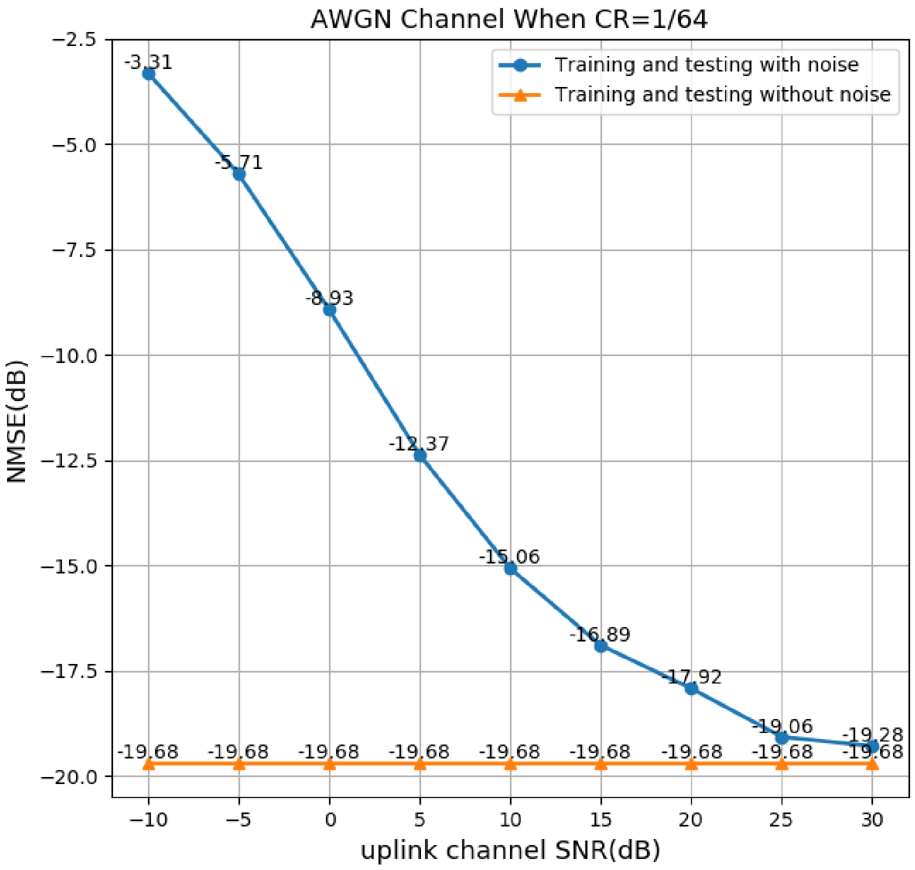
Performances of the CsiNet 5G and the FullyConv.

Ratio	methods	Loss	NMSE (dB)
1/8	CsiNet_5G	2.80×10^{-3}	–22.5777
	FullyConv	5.72×10^{-4}	–28.4488
1/64	CsiNet_5G	1.76×10^{-2}	–13.0397
	FullyConv	4.99×10^{-3}	–18.8472
1/128	CsiNet_5G1	1.87×10^{-2}	–12.8182
	FullyConv	8.82×10^{-3}	–16.1094
1/256	CsiNet_5G	2.30×10^{-2}	–11.9154
	FullyConv	2.05×10^{-2}	–12.4313

IV. Deep Learning-based CSI Inferring Methods

(7) Experimental results of FullyConv compared with baseline





Simulation results in AWGN channel when $CR = \{1/8, 1/64\}$

IV. Deep Learning-based CSI Inferring Methods

(8) Model complexity

- **The model complexity** can be measured by **time complexity** and **space complexity.**
- **Time complexity** refers to the number of floating-point operations (**FLOPs**) in a forward propagation of the model after a single sample is input.
- **Space complexity** refers to the total amount of **memory exchange** in a forward propagation of the model after a single sample is input, which is the memory consumption of the weights of each layer of the model.
- **The Time complexity** defines the training/prediction time of the model.
- **The space complexity** defines the number of parameters of the model.

IV. Deep Learning-based CSI Inferring Methods

(8.1) Space complexity

Number\CR	1/8	1/64	1/256
CsiNet_5G	4,161,953,180	520,365,692	130,195,604
FC layer	4,161,943,296	520,355,808	130,185,720
Proportion	99.9998%	99.9981%	99.9924%
FullyConv	50,170	50,390	50,610

The total weight parameters of all parameterized layers of the models

- CsiNet_5G far exceeds FullyConv in terms of parameters because of the FC layer.
- The FC layer occupies more than 99% of the parameters of CsiNet_5G.
- Due to the use of convolutional layers to compress and decompress downlink CSI, the amount of parameters of FullyConv is much smaller than that of CsiNet_5G.

IV. Deep Learning-based CSI Inferring Methods

(8.2) Time complexity

$$\text{Conv Layer Time Complexity} \sim T\left(\prod_{i=1}^{N} M_i \cdot \prod_{j=1}^{L} K_j \cdot C_{in} \cdot C_{out}\right)$$

$$\text{Dense Layer Time Complexity} \sim T(P_{in} \cdot P_{out})$$

$$\text{Model Time Complexity} \sim T\left(\sum_{l=1}^{C}\left(\prod_{i=1}^{N} M_{li} \cdot \prod_{j=1}^{G} K_{lj} \cdot C_{l-1} \cdot C_l\right) + \sum_{l=1}^{D}(P_{l-1} \cdot P_l)\right)$$

variable notations for computing the time complexity

- M_i: i-th side of convolution kernel
- K_j: j-th side of output feature map
- C_{in}: input channels
- C_{out}: output channels
- P_{in}: input neurons of FC layers
- P_{out}: output neurons of FC layers
- C: number of convolutional layers
- D: number of FC layers

Time complexities of the two models

CR=1/8	CsiNet_5G	FC layer	FullyConv
Flops	7.37 G	4.16 G	3.21 G

Deep Transfer Learning for 5G Massive MIMO Downlink CSI Feedback

J. Zeng, J. Sun, G. Gui, B. Adebisi, T. Ohtsuki, H. Gacanin, H. Sari, “Downlink CSI Feedback Algorithm With Deep Transfer Learning for FDD Massive MIMO Systems,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1253-1265, 2021.

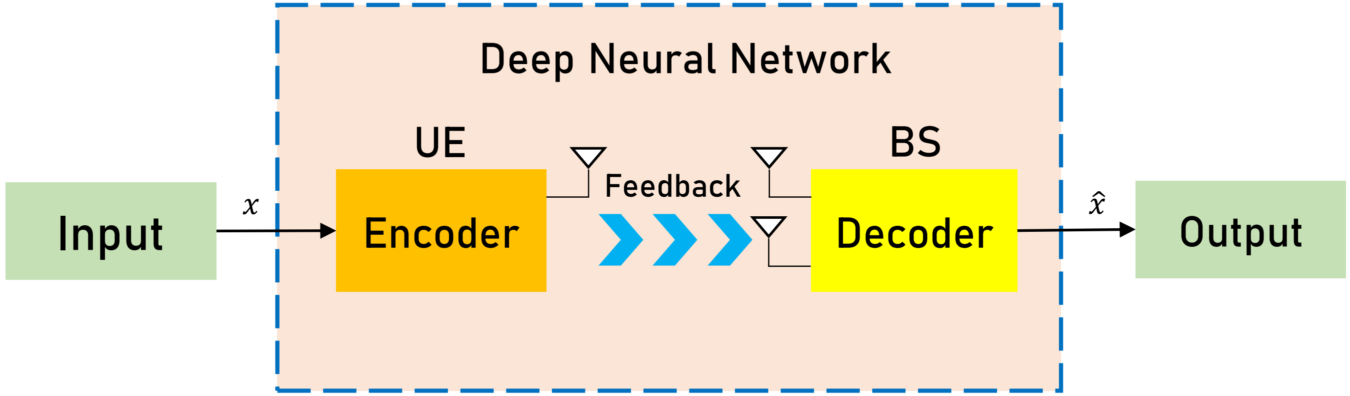
IV. Deep Learning-based CSI Inferring Methods

(1) Background and problem

- Acquisition of downlink channel state information (CSI) is an import procedure at the base station (BS) for high quality wireless transmission in frequency division duplexing (FDD) communication systems.
- Compared with the traditional methods, the deep neural network (DNN) can effectively compress the downlink CSI, thus greatly reducing the feedback overhead. **However, the generalization of DNN is poor**, hence it is necessary to train a DNN from scratch whenever there is a change in the wireless channel environment.
- Training a DNN from scratch requires huge data cost and time cost in 5G massive multiple-input multiple output (MIMO) systems.
- For a similar task, **the deep transfer learning** can obtain a model with excellent performance using a small number of samples based on the pre-trained model.

IV. Deep Learning-based CSI Inferring Methods

(2) System model based on deep transfer learning



The process of downlink CSI feedback using the DNN.

- At the UE (user equipment) side: the downlink CSI is inputted into the encoder of DNN for compression

$$s = f_{en}(H)$$

- At the BS side: the low-dimensional codeword s is inputted into the decoder of DNN for recovering

$$H = f_{de}(s)$$

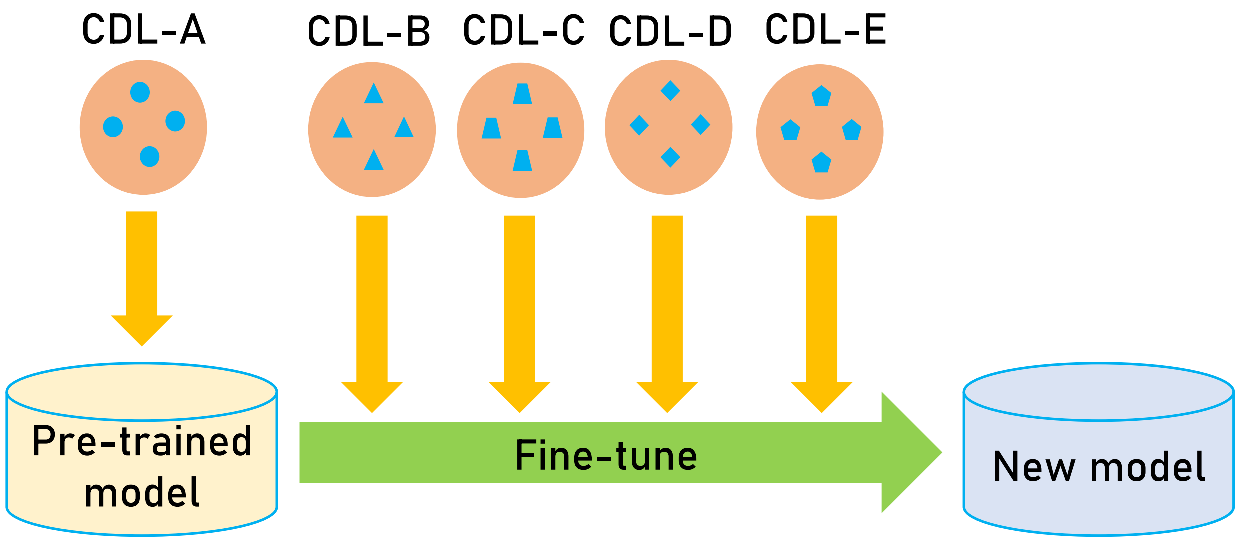
IV. Deep Learning-based CSI Inferring Methods

(3) Deep transfer learning algorithm for CSI limited feedback

- 3GPP R15 defines a new channel model named clustered delay line (CDL) in 38.901, which is divided into CDL-A, CLD-B, CDL-C, CDL-D and CDL-E according to simulated network environments.

- Large number of samples of CDL-A channel are used to train a DNN as the pre-trained model.

- Small number of samples of CDL-B, CDL-C, CDL-D, CDL-E channels are used to fine-tune the pre-trained model, respectively.



The deep transfer learning model for downlink CSI feedback.

IV. Deep Learning-based CSI Inferring Methods

(4) Experiment results

- The performance of the CDL-A model is obtained by training the DNN from scratch with 50000 samples, while the NMSEs of the other channel models are obtained using 4000 samples to fine-tune the CDL-A pre-trained model.

- In different compression ratios γ, the NMSEs of the CDL-B and CDL-C models are similar to that of the CDL-A model, while the NMSEs of the CDL-D and CDL-E models are even better than that of the CDL-A model.

- In four different compression ratios, the training time of the CDL-A model is about 40h using RTX 2080Ti GPU, while the training time of the other channel models is about 4h20min using GTX 1080Ti GPU.

Performance comparison between different models

γ	Channel model	NMSE (dB)	Test loss
1/8	CDL-A	−28.449	5.72×10^{-4}
	CDL-B	−26.934	8.53×10^{-4}
	CDL-C	−29.066	6.14×10^{-4}
	CDL-D	−33.646	3.07×10^{-4}
	CDL-E	−33.532	3.12×10^{-4}
1/64	CDL-A	−16.940	7.62×10^{-3}
	CDL-B	−13.565	1.79×10^{-2}
	CDL-C	−15.553	1.31×10^{-2}
	CDL-D	−23.487	3.14×10^{-3}
	CDL-E	−23.177	3.36×10^{-3}
1/128	CDL-A	−16.109	8.82×10^{-3}
	CDL-B	−12.887	2.10×10^{-2}
	CDL-C	−14.784	1.56×10^{-2}
	CDL-D	−21.993	4.44×10^{-3}
	CDL-E	−22.077	4.34×10^{-3}
1/256	CDL-A	−12.431	2.05×10^{-2}
	CDL-B	−8.454	5.84×10^{-2}
	CDL-C	−9.860	4.81×10^{-2}
	CDL-D	−19.381	8.05×10^{-3}
	CDL-E	−18.299	1.03×10^{-2}

IV. Deep Learning-based CSI Inferring Methods

(4) Experiment results

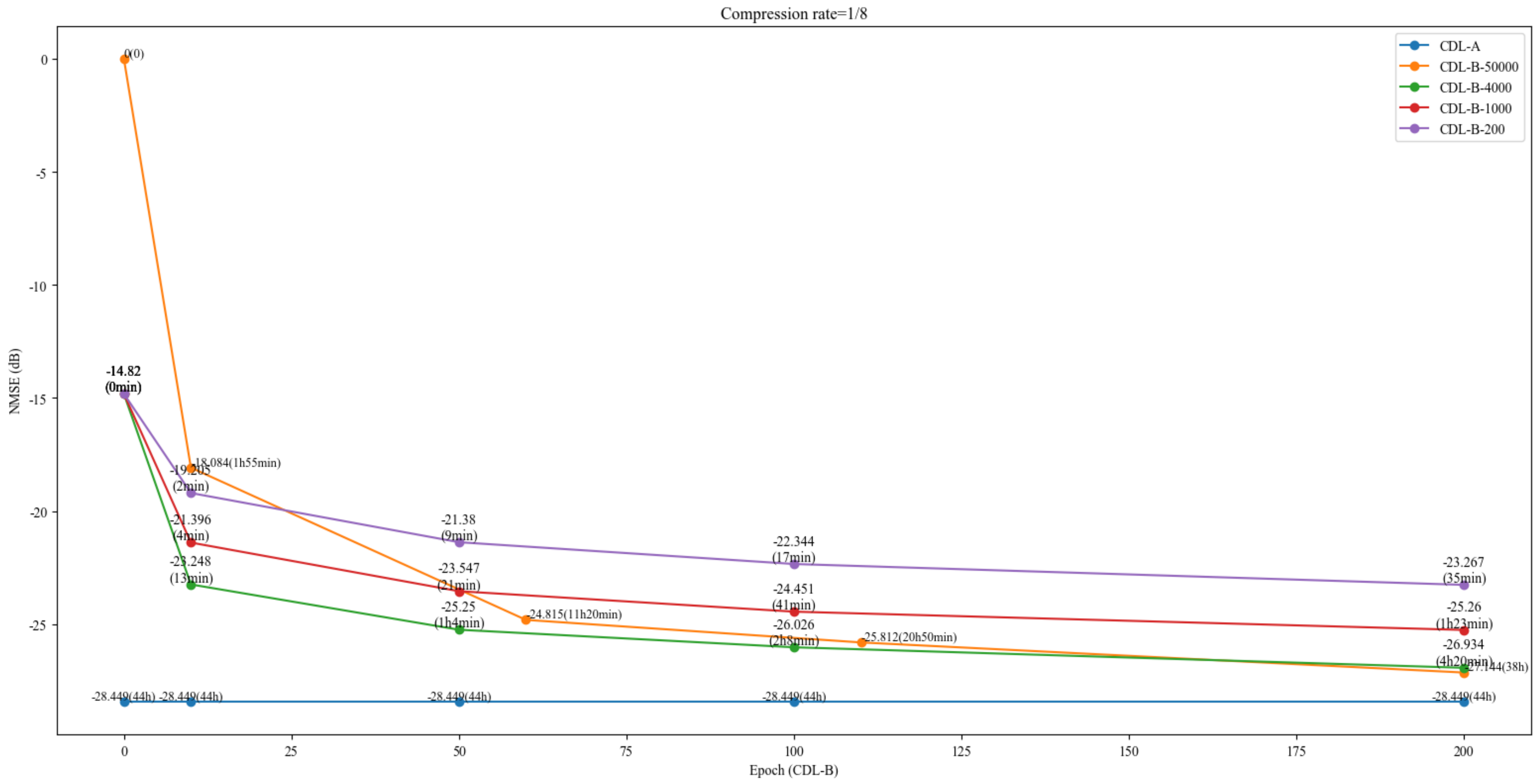
- With the sample size reduces from 4000 to 200, the NMSEs of the CDL-B and CDL-D models also gradually decline.
- With the reduction of the sample size, the training cost is also gradually reduced.
- The reduction of sample size can further reduce the training cost by bearing a small loss of model performance.

Performance comparison between different sample sizes ($\gamma = 1/8$).

Sample size	CDL-B		CDL-D	
	NMSE (dB)	Training time	NMSE (dB)	Training time
50,000	-27.144	38h	-32.212	35h20mim
4,000	-26.934	4h20min	-33.646	4h20min
3,000	-26.570	3h30min	-33.512	3h50min
2,000	-26.070	2h18min	-33.123	2h28min
1,000	-25.260	1h23min	-32.538	1h23min
500	-24.423	1h	-32.100	1h
200	-23.267	35min	-31.392	35min

IV. Deep Learning-based CSI Inferring Methods

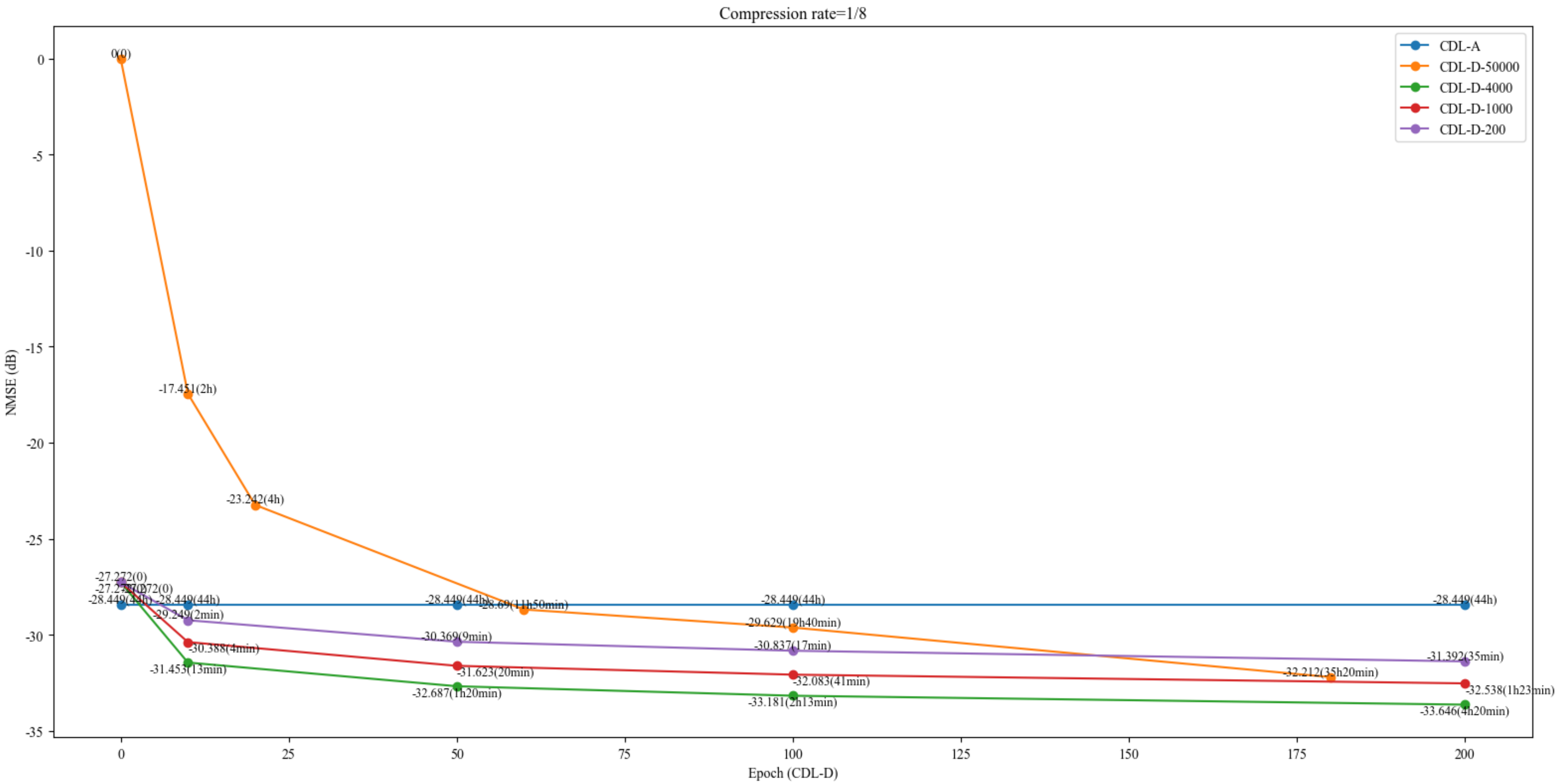
(4) Experiment results



The NMSE of the CDL-B model during training process.

IV. Deep Learning-based CSI Inferring Methods

(4) Experiment results



The NMSE of the CDL-D model during training process.

V. Summary

- Background and Classification of ML for wireless communication
- AMC: LightAMC, Fede-AMC (SISO); ZF-AMC, Co-AMC, TL-AMC (MIMO)
- SEI: Few-Shot SEI via Deep Metric Ensemble Learning
- CSI Inferring: CSI prediction; CSI limited feedback (FCN, TL)

References

- L. Liang, H. Ye, G. Yu, and G. Y. Li, “Deep-Learning-Based Wireless Resource Allocation With Application to Vehicular Networks,” *Proceedings of the IEEE*, vol. 108, no. 2, pp. 341-356, Dec. 2019.
- Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, “Deep Learning in Physical Layer Communications,” *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93-99, Mar. 2019.
- Y. Wang, J. Yang, M. Liu and G. Gui, "LightAMC: Lightweight Automatic Modulation Classification via Deep Learning and Compressive Sensing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3491-3495, Mar. 2020.
- Y. Wang, *et al.*, "Distributed Learning for Automatic Modulation Classification in Edge Devices," *IEEE Wireless Communications Letters*, vol. 9, no. 12, pp. 2177-2181, Dec. 2020.
- Y. Wang, J. Wang, W. Zhang, J. Yang and G. Gui, "Deep Learning-Based Cooperative Automatic Modulation Classification Method for MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4575-4579, Apr. 2020.
- Y. Wang, *et al.*, "Automatic Modulation Classification for MIMO Systems via Deep Learning and Zero-Forcing Equalization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5688-5692, May 2020.
- Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari and F. Adachi, "Transfer Learning for Semi-Supervised Automatic Modulation Classification in ZF-MIMO Systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 2, pp. 231-239, Jun. 2020.
- Y. Wang, G. Gui, Y. Lin, H.-C. Wu, C. Yuen, F. Adachi, “Few-Shot Specific Emitter Identification via Deep Metric Ensemble Learning,” *IEEE Internet of Things Journal*, early access, 2022.
- Y. Zhang, *et al.*, “CV-3DCNN: Complex-valued Deep Learning for CSI Prediction in FDD Massive MIMO Systems,” *IEEE Wireless Communication Letters*, vol. 10, no. 2, pp. 266-270, Feb. 2021.
- G. Fan, J. Sun, G. Gui, H. Gacanin, B. Adebisi, T. Ohtsuki, “Fully Convolutional Neural Network-Based CSI Limited Feedback for FDD Massive MIMO Systems,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 672-682, 2022.
- J. Zeng, J. Sun, G. Gui, B. Adebisi, T. Ohtsuki, H. Gacanin, H. Sari, “Downlink CSI Feedback Algorithm With Deep Transfer Learning for FDD Massive MIMO Systems,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1253-1265, 2021.

Thanks a lot for your attention