Wi-Fi Sensing via Deep Learning

Zijian Zhao
zzhaock@connect.ust.hk
https://zijianzhao.netlify.app

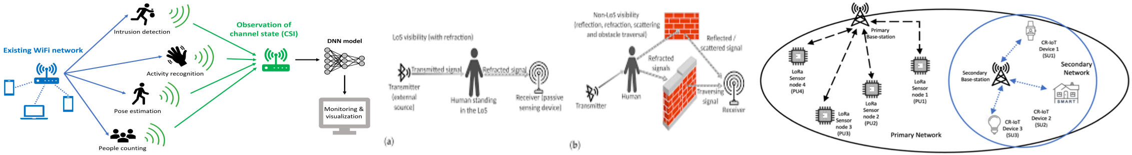
Department of Civil and Environmental Engineering
Hong Kong University of Science and Technology

March 25, 2025

Background: What is Wireless Sensing?

- **Definition**: Wireless sensing is a method for collecting and transmitting data through wireless networks, primarily used to monitor the state of environments, objects, or systems.

- **Typical Wireless Signals**:
  - Wi-Fi (***our research focus**)
  - Bluetooth
  - LoRa



(a) Wi-Fi sensing (b) Bluetooth sensing (c) LoRa sensing

Figure 1: Three typical wireless sensing technologies

Background: What is Wireless Sensing?

- **Definition**: Wireless sensing is a method for collecting and transmitting data through wireless networks, primarily used to monitor the state of environments, objects, or systems.

- **Typical Wireless Signals**:
  - Wi-Fi (***our research focus**)
  - Bluetooth
  - LoRa

- **Sensing Types**:
  - Active sensing: Sensors actively emit signals or stimuli to gather information about the environment or objects.
  - Passive sensing: Sensors passively receive signals or information that naturally exist in the environment without actively emitting signals. (***our research focus**)

Background: Why We Need Wi-Fi Sensing?

❑ **Benefits**

- High privacy
- High penetration
- Extensive coverage: effective even in Non-Line-of-Sight (NLOS) situations
- Low cost

	Function	Accuracy	Coverage	Privacy	Cost
Camera	Comprehensive	High	Low (Affected by occlusion/lighting)	Low	Relatively Low ~￥400
Millimeter Wave Radar	Partially Comprehensive	Moderate	Moderate	Moderate	High ~￥2000
Infrared Sensor	Limited	High	Moderate	High	Low ~￥20
Wi-Fi	Partially Comprehensive	Moderate	High (Non-line-of-sight)	High	Almost Zero

Figure 2: Comparison of Wi-Fi with other sensing technologies

Background: Why We Need Wi-Fi Sensing?

- **Benefits**
  - High privacy
  - High penetration
  - Extensive coverage: effective even in Non-Line-of-Sight (NLOS) situations
  - Low cost

Figure 3: Potential application scenarios of Wi-Fi sensing

Background: How to Realize Wi-Fi Sensing?

Common Wi-Fi Sensing Features

- Received Signal Strength Indicator (RSSI): A measure of the power level that a receiver detects, indicating the strength of the received signal.
- Channel State Information (CSI): A detailed representation of the wireless channel's characteristics, including amplitude and phase information. This allows for advanced signal processing techniques.

CSI Estimation

$$Y = HX + N$$

- Y: received signals; X: transmitted signals
- H: channel matrix; N: noise signals

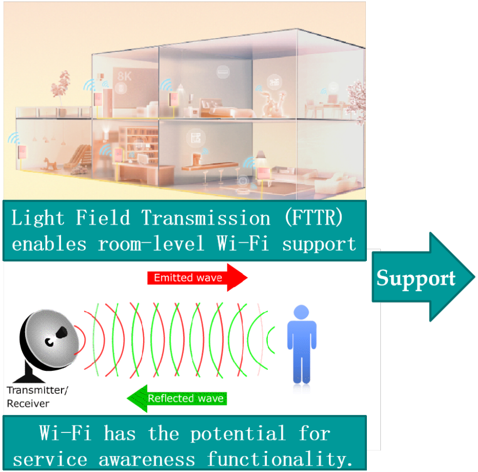
CSI Components

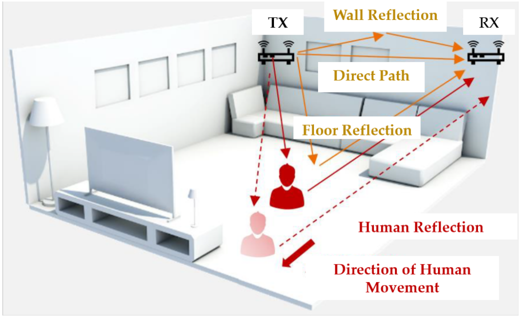
$$H(f,t) = H_s(f,t) + H_d(f,t)$$

- H_s: static component; H_d: dynamic component
- f: subcarrier frequency; t: time-domain sampling point

Background: How to Realize Wi-Fi Sensing?

❑ Wi-Fi Sensing Principle





- **Principle:** Based on **multipath propagation** of wireless signals in the device deployment environment, indoor personnel activities are identified by analyzing **changes in wireless channel parameter characteristics**.

Background: How to Realize Wi-Fi Sensing?

❑ General Wi-Fi Sensing Framework

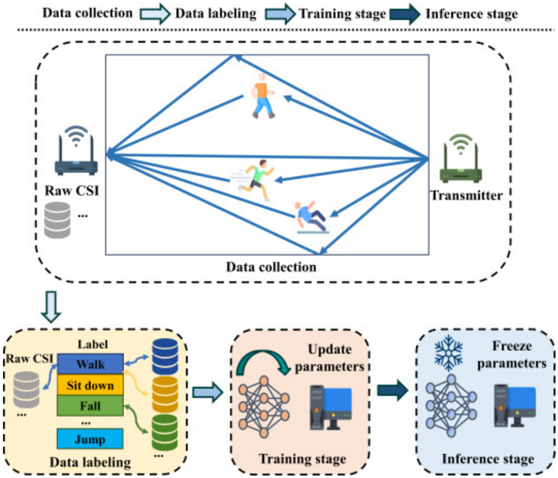


Figure 4: Workflow of learning-based Wi-Fi sensing system (Chen et al. 2024)

Background: Current Situation of Wi-Fi Sensing

- **Wi-Fi Sensing Important Moments**
  - 2008: Research on Wi-Fi sensing began to emerge, based on the 802.11n protocol.
  - 2019: IEEE initiated formal discussions on Wi-Fi sensing.
  - 2025: The first Wi-Fi sensing protocol (802.11bf) is expected to be approved, further advancing Wi-Fi technology.

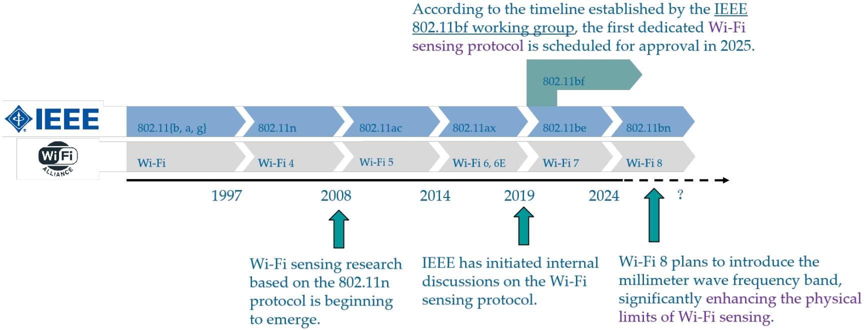
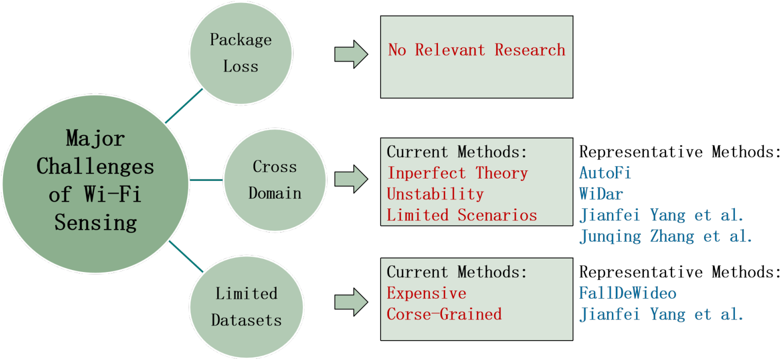


Figure 5: Development of Wi-Fi protocol

Main Challenges & Research Gaps



Three Major Challenges of Wi-Fi Sensing:

1. Signal Characteristics Easily Drowned: Various factors can lead to signal loss, which negatively impacts the performance of sensing models.
2. Weak Generalization of Sensing Models: The accuracy of sensing models is highly dependent on specific environmental conditions.
3. Difficulties in Data Collection: There are numerous diverse scenarios, and the costs associated with data collection are often high.

Research Objectives

Research Objectives

(1) To develop package recovery method tailored to the structure of CSI signals.

(2) To develop method and theoretical framework for general and practical cross-domain Wi-Fi sensing.

(3) To develop easy and cost-effective method for rapid Wi-Fi sensing data.

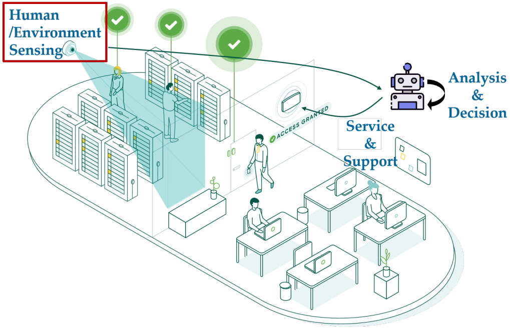


Figure 6: Smart spaces via Wi-Fi sensing

Outline

1. Research Overview
2. CSI-BERT: A Multifunctional Framework for CSI Time Series
3. KNN-MMD: An Effective Framework for Practical Cross-Domain Wi-Fi Sensing
4. CrossFi: A Multi-scenario Framework for Cross-Domain Wi-Fi Sensing
5. LoFi: IoT-Enabled Wi-Fi Sensing Deployment
6. Concluding Remarks

Background: Package Loss

- **Factors Causing Package Loss**
  - Environment noise
  - Frequency interference
  - Hardware errors
  - ...
- **Influence of Package Loss**
  - Incomplete CSI data $\rightarrow$ *Affects model performance!*

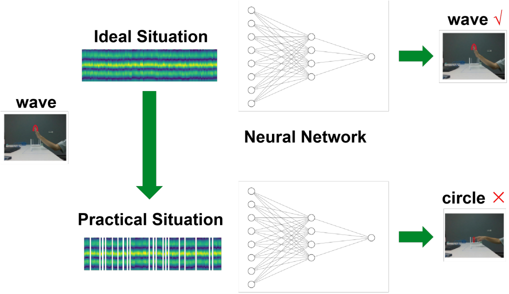


Figure 7: Impact of package loss

Motivation: Mask Language Model (MLM) of BERT

- **Mask Language Model (MLM)**
  - **Original sentence**: “Wi-Fi sensing is one of the important technologies in ISAC.”
  - **Random MASK**: “Wi-Fi [MASK] is one of the [MASK] technologies in ISAC.”
  - **Recovered sentence**: “Wi-Fi sensing is one of the popular technologies in ISAC.”
- **Why Using MLM for CSI Recovery?**
  - The task of recovering lost packets is analogous to MLM.
  - MLM does not require labeled data. $\rightarrow$ *Enable training with unlabeled and incomplete CSI sequences!*

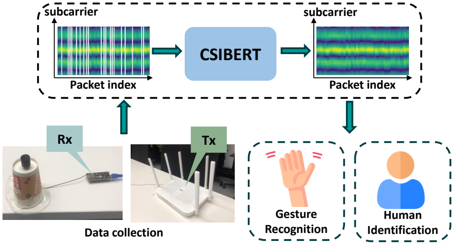
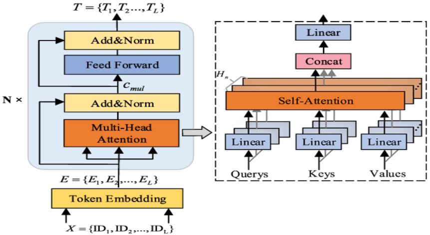


Figure 8: Principle of CSI-BERT

- **Previous Works**
  - BERT for radio map construction (Wang et al. 2023)
  - BERT for indoor localization (Guo et al. 2022; Sun et al. 2021)
- **Shortcomings of Previous Works**
  - Converting continuous signal data into discrete tokens → *information loss.*
  - Applying BERT from NLP directly without any adaptation design → *low performance.*



(a) Model structure

Token	ID
[PAD]	0
[MASK]	1
(MAC1) - (-40dB)	2
(MAC1) - (-41dB)	3
...	...
(MAC1) - (-99dB)	62
(MAC2) - (-40dB)	63
(MAC2) - (-41dB)	64
...	...
(MAC2) - (-99dB)	122
...	...
(MACr) - (-99dB)	1562

One sample data

Token	ID
$[\text{M-R}]_1$	64
$[\text{M-R}]_2$	131
...	...
$[\text{M-R}]_n$	744
$[\text{PAD}]_{n+1}$	0
...	...
$[\text{PAD}]_L$	0

Embedding

E

0	1	...	d
-0.877	0.365	...	3.028
-0.592	0.519	...	-0.361
...	...	...	..
-1.108	-0.236	...	-0.230
0.031	0.129	...	-1.248
...	...	...	...
0.031	0.129	...	-1.248

(b) Tokenization approach

Figure 9: Method proposed in Sun et al. 2021

Model Structure

❑ Main Design

- CSI Embedding Layer with Standardization Mechanism

$$\mathrm{Std}(x_i) = \frac{x_i - \mu_i}{\sigma_i} \quad \mathrm{De\text{-}Std}(y_i) = (y_i + \mu_i) * \sigma_i$$

  - x: input; y: output
  - μ: mean; σ: standard deviation
- Time Embedding Layer: Positional Embedding Style
- Discriminator

$$\min_R \max_D E_x[log(D(x))] + E_x[log(1 - D(R(x)))]$$

  - R: Recoverer; D: Discriminator

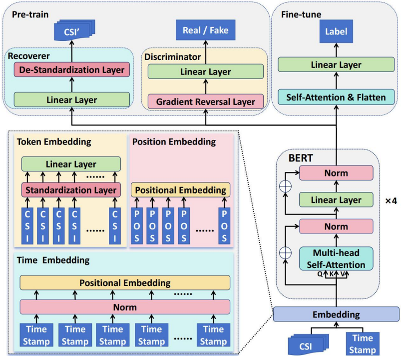


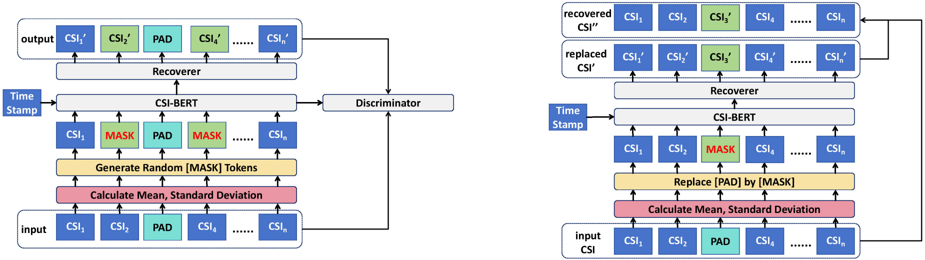
Figure 10: Architecture of CSI-BERT

[1]**Zijian Zhao**, Tingwei Chen, Fanyi Meng, Hang Li, Xiaoyang Li, Guangxu Zhu*, "Finding the Missing Data: A BERT-inspired Approach Against Package Loss in Wireless Sensing" (2024 IEEE International Conference on Computer Communications (INFOCOM) DeepWireless Workshop)

[2]**Zijian Zhao**, Kaifeng Han, Qimei Chen, Guangxu Zhu, Xiaoyang Li, Hang Li, "Channel State Information Recovery Method and Apparatus, Equipment, Storage Medium" (Patent Number: ZL2024102321250, 2024)

Workflow

- **Unsupervised Training**
  - Replace CSI with [MASK] $\sim N(\mu, \sigma^2)$ randomly
  - Train the model to recover the CSI sequence
- **Inference**
  - Fill in the blank positions with [MASK]s and infer them using the trained CSI-BERT
  - Two recovery methods: Recover & Replace

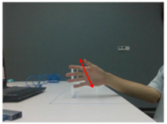


(a) Unsupervised training (b) Inference (recovery task)

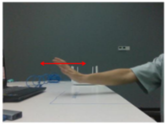
Figure 11: Workflow of CSI-BERT

Experiment Setup

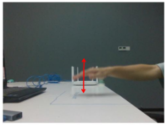
- **Dataset: WiGesture**
  - 60-minute dataset used for gesture recognition and people identification
  - collected using ESP32-S3 (1 antenna, 52 subcarriers)
  - 6 actions & 8 volunteers



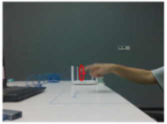
(a) Left-right



(b) Forward-backward

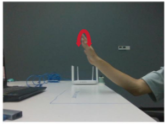


(c) Up-down



(d) Circling

(e) Clapping



(f) Waving

Figure 12: Gesture sketch map of WiGesture dataset

Experiment Result

❑ Experiment Result

- CSI-BERT achieves the lowest recovery error and provides the greatest improvement to classification models.
- However, CSI-BERT performs worse than ResNet in sensing tasks.

Method	MSE ↓	MAE ↓	SMAPE ↓	MAPE ↓	FSS ↑	Time Cost (min) ↓
CSI-BERT	**1.7326**	**0.9413**	**0.0902**	**0.0945**	**0.9999 (replace), 0.9979 (recover)**	**0.03**
Linear Interpolation	2.8294	1.2668	0.1248	0.1344	0.9841	0.64
Ordinary Kringing	3.6067	1.4371	0.1627	0.1395	0.9936	45.15
IDW	2.4306	1.1854	0.1278	0.1167	0.9970	3.30

Table 1: CSI recovery error

Task	Action Classification						People Identification					
Model / Data	MLP 337K	CNN 23K	RNN 33K	LSTM 133K	ResNet 11M	CSI-BERT 2M	MLP 337K	CNN 23K	RNN 33K	LSTM 133K	ResNet 11M	CSI-BERT 2M
Original Data	66.93%	55.72%	39.56%	11.97%	70.31%	76.91%	71.34%	71.14%	66.39%	21.09%	83.76%	93.94%
CSI-BERT recover	74.23%	59.39%	48.96%	22.92%	**92.57%**	71.87%	97.13%	80.60%	80.51%	**35.18%**	94.30%	95.05%
CSI-BERT replace	**86.90%**	**61.51%**	**58.80%**	**52.36%**	84.52%	**79.54%**	**97.65%**	79.18%	**89.24%**	24.22%	97.39%	95.83%
Linear Interpolation	72.91%	58.35%	45.32%	49.09%	80.75%	74.55%	81.84%	70.88%	84.45%	26.83%	86.75%	**97.92%**
Ordinary Kringing	65.62%	57.55%	53.64%	50.00%	88.71%	74.27%	94.76%	**85.38%**	86.42%	21.61%	97.32%	95.83%
IDW	40.17%	56.77%	48.70%	46.88%	80.32%	67.22%	83.22%	74.56%	88.54%	33.91%	94.27%	95.20%

Table 2: CSI sensing classification performance

Second-Generation Model Structure

- **Shortcomings of CSI-BERT1**
  - Limited capacity to capture the relationship between subcarriers
  - Permutation invariance of the positional embedding-based time embedding layer
- **Main Design**
  - Adaptive Re-Weighting Layer (ARL)

$$\mathrm{ARL}(x) = x \cdot \mathrm{MLP}(x)$$

    - $\mathrm{MLP}(x)$: adaptive weight
    - $\cdot$: dot product
  - MLP-based Time Embedding Layer

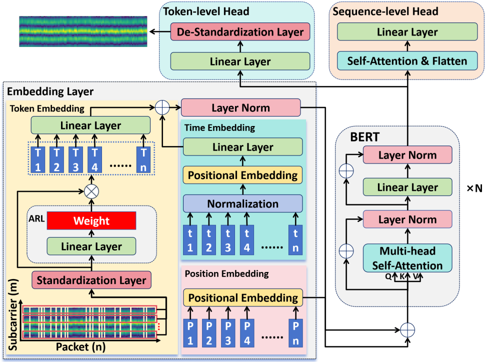


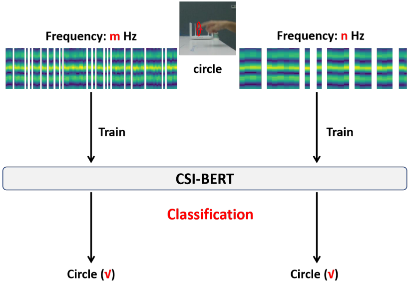
Figure 13: Architecture of CSI-BERT2

[1] **Zijian Zhao**, Fanyi Meng, Hang Li, Xiaoyang Li, Guangxu Zhu*, "CSI-BERT2: A BERT-Inspired Framework for Efficient CSI Prediction and Recognition in Wireless Communication and Sensing" (under review)

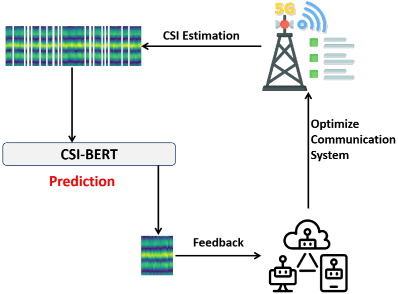
[2] Tingwei Chen, Yantao Wang, Hanzhi Chen, **Zijian Zhao**, Xinhao Li, Nicola Piovesan, Guangxu Zhu*, Qingjiang Shi, "Modelling the 5G Energy Consumption using Real-world Data: Energy Fingerprint is All You Need" (under review, IEEE Wireless Communications Letters (WCL))

Other Application Scenarios

- **CSI Sensing Task under Various Sampling Rates**
  - In practice, data from different users or scenarios may be heterogeneous.
- **CSI Prediction Task**
  - In wireless communication, estimating the CSI matrix is challenging and time-consuming. Longer CSI estimation times lead to reduced valid communication time.



(a) CSI Sensing Task



(b) CSI Prediction Task

Figure 14: Application scenarios of CSI-BERT2

Workflow

- ❑ **Unsupervised Pre-training**
- ❑ **Supervised Fine-tuning**
  - Sensing task: Mask Fine-tuning
  - Prediction task: Mask Prediction Model (MLM)
- ❑ **Inference**

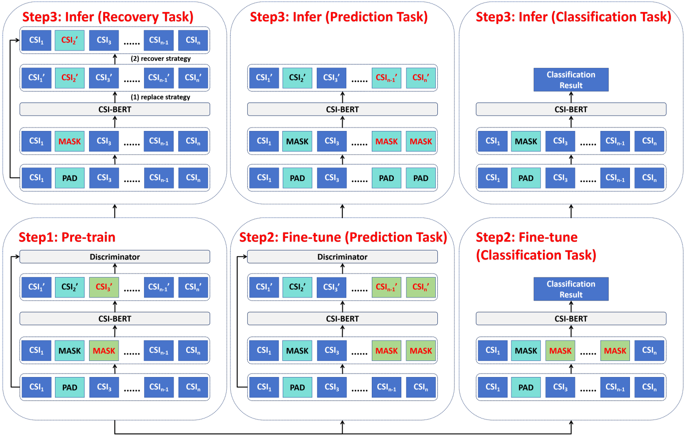
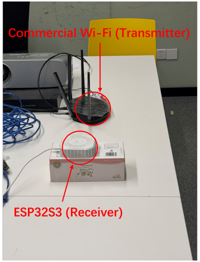


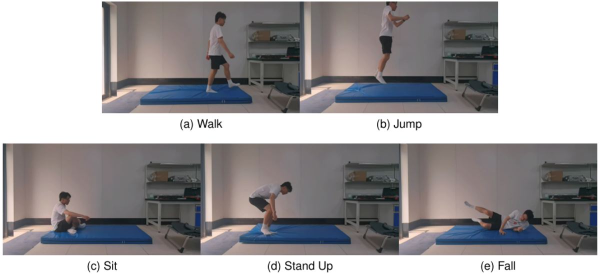
Figure 15: Workflow of CSI-BERT2

Experiment Setup

- **Dataset 1: WiFall**
  - 45-minute dataset used for fall detection, action recognition, and people identification
  - 5 actions & 10 volunteers
- **Dataset 2: WiCount**
  - 15-minute dataset used for estimating the number of people
  - $0 \sim 4$ people



(a) Hardware



(b) Sketch map

Figure 16: WiFall dataset

❑ CSI Recovery Task

- CSI-BERT2 significantly outperforms CSI-BERT1 in both recovery and sensing tasks.

Dataset	WiGesture				WiFall				WiCount			
Method \ Metric	MSE	SMAPE	MAPE	Time(s)	MSE	SMAPE	MAPE	Time(s)	MSE	SMAPE	MAPE	Time(s)
CSI-BERT2	**2.0800**	**0.1153**	**0.1217**	5.53	**4.1463**	**0.1240**	**0.1351**	2.77	2.4531	**0.1092**	0.1189	1.56
CSI-BERT	2.2438	0.1156	0.1244	**1.84**	4.4042	0.1271	0.1373	**1.32**	**2.4471**	**0.1092**	0.1185	**0.67**
Linear Interpolation	2.8642	0.1266	0.1364	38.49	6.4420	0.1461	0.1571	2.81	2.6870	0.1099	**0.1175**	1.20
Ordinary Kringing	3.5090	0.1390	0.1612	2709.43	4.6637	0.1319	0.1462	289.09	4.5964	0.1423	0.1684	109.10
Inverse Distance Weighted (IDW)	2.4726	0.1187	0.1301	19.82	4.4251	0.1276	0.1409	2.45	3.4431	0.1268	0.1483	0.82

Table 3: Recovery error

Task	People Number Estimation (WiCount Dataset)								
Data \ Model	MLP 337K	CNN 23K	RNN 33K	LSTM 133K	Chen et al. 11M	WiGRUNT 11M	CSI-BERT 2M	CSI-BERT2 5M	Average
Original Data	56.77%	69.68%	80.93%	80.72%	48.33%	49.53%	**89.67%**	**<u>94.32%</u>**	71.24%
CSI-BERT2 recover	87.29%	78.75%	83.98%	81.51%	83.32%	**<u>85.42%</u>**	84.06%	<u>91.32%</u>	84.45%
CSI-BERT2 replace	85.62%	78.49%	**88.12%**	86.97%	81.41%	82.34%	81.51%	<u>92.76%</u>	**84.65%**
CSI-BERT recover	**88.98%**	80.83%	86.20%	82.39%	82.22%	82.70%	79.04%	<u>92.70%</u>	84.38%
CSI-BERT replace	81.61%	72.60%	85.67%	84.95%	**85.62%**	82.75%	81.61%	<u>92.86%</u>	83.46%
Linear Interpolation	76.51%	77.73%	85.52%	82.40%	80.17%	83.07%	86.51%	<u>88.64%</u>	82.57%
Ordinary Kringing	87.29%	50.52%	84.84%	85.05%	82.97%	76.72%	85.72%	<u>91.90%</u>	80.63%
IDW	80.72%	**82.29%**	84.17%	**87.00%**	82.10%	81.72%	85.62%	<u>88.54%</u>	84.02%

Table 4: CSI classification performance in WiCount dataset

Experiment Result

❑ CSI Recovery Task

Task	Gesture Recognition (WiGesture Dataset)								
Model / Data	MLP 337K	CNN 23K	RNN 33K	LSTM 133K	Chen et al. 11M	WiGRUNT 11M	CSI-BERT 2M	CSI-BERT2 5M	Average
Original Data	66.93%	55.72%	39.56%	11.97%	70.31%	48.73%	76.91%	**99.48%**	58.70%
CSI-BERT2 recover	72.88%	57.27%	54.34%	48.35%	**92.96%**	**78.97%**	**92.18%**	89.06%	73.25%
CSI-BERT2 replace	73.68%	**62.80%**	55.48%	40.79%	91.92%	74.99%	81.51%	91.95%	71.63%
CSI-BERT recover	74.23%	59.39%	48.96%	22.92%	92.57%	71.87%	71.87%	92.70%	66.81%
CSI-BERT replace	**86.90%**	61.51%	**58.80%**	52.36%	84.52%	78.84%	79.54%	91.41%	**74.24%**
Linear Interpolation	72.91%	58.35%	45.32%	49.09%	80.75%	74.91%	74.55%	88.25%	68.01%
Ordinary Kringing	65.62%	57.55%	53.64%	**50.00%**	88.71%	69.99%	74.27%	85.93%	68.21%
IDW	40.17%	56.77%	48.70%	46.88%	80.32%	71.06%	67.22%	88.28%	62.42%
Task	**People Identification (WiGesture Dataset)**								
Model	**MLP**	**CNN**	**RNN**	**LSTM**	**Chen et al.**	**WiGRUNT**	**CSI-BERT**	**CSI-BERT2**	**Average**
Original Data	71.34%	71.14%	66.39%	21.09%	83.76%	72.07%	93.94%	**99.73%**	72.43%
CSI-BERT2 recover	95.57%	**85.54%**	84.60%	27.98%	93.20%	81.73%	**97.92%**	**99.73%**	83.28%
CSI-BERT2 replace	95.05%	83.07%	84.68%	**54.13%**	95.33%	83.84%	96.35%	94.79%	**85.91%**
CSI-BERT recover	**97.13%**	80.60%	80.51%	35.18%	94.30%	**84.67%**	95.05%	**99.73%**	83.39%
CSI-BERT replace	97.65%	79.18%	**89.24%**	24.22%	**97.39%**	77.77%	95.83%	99.47%	82.59%
Linear Interpolation	81.84%	70.88%	84.45%	26.83%	86.75%	70.28%	**97.92%**	91.67%	76.33%
Ordinary Kringing	94.76%	85.38%	86.42%	21.61%	97.32%	80.84%	95.83%	99.03%	82.64%
IDW	83.22%	74.56%	88.54%	33.91%	94.27%	80.70%	95.20%	99.47%	81.23%

Table 5: CSI classification performance in WiGesture dataset

❑ CSI Recovery Task

Task	Action Recognition (WiFall Dataset)								
Model / Data	MLP 337K	CNN 23K	RNN 33K	LSTM 133K	Chen et al. 11M	WiGRUNT 11M	CSI-BERT 2M	CSI-BERT2 5M	Average
Original Data	47.48%	56.27%	58.61%	52.10%	51.38%	34.44%	**82.43%**	**88.59%**	58.91%
CSI-BERT2 recover	64.97%	**67.18%**	68.48%	62.63%	71.70%	70.96%	67.63%	72.16%	68.21%
CSI-BERT2 replace	69.01%	66.27%	**70.18%**	61.99%	73.96%	69.72%	66.77%	72.70%	**68.82%**
CSI-BERT recover	66.40%	54.94%	68.48%	61.79%	69.66%	70.94%	67.36%	73.69%	66.65%
CSI-BERT replace	**73.05%**	54.97%	66.79%	**66.73%**	72.01%	67.44%	66.61%	73.67%	67.65%
Linear Interpolation	67.44%	64.32%	67.31%	59.78%	**74.22%**	70.57%	64.37%	74.19%	67.77%
Ordinary Kringing	67.96%	65.52%	64.44%	63.88%	70.92%	62.41%	67.36%	71.77%	66.78%
IDW	70.31%	67.08%	69.79%	62.32%	71.09%	**72.39%**	67.22%	70.21%	68.80%
Task	**Fall Detection (WiFall Dataset)**								
Model	**MLP**	**CNN**	**RNN**	**LSTM**	**Chen et al.**	**WiGRUNT**	**CSI-BERT**	**CSI-BERT2**	**Average**
Original Data	78.34%	52.99%	82.29%	80.35%	78.52%	73.69%	**93.28%**	**94.79%**	79.28%
CSI-BERT2 recover	80.79%	**75.95%**	**86.58%**	**86.72%**	**82.42%**	80.90%	82.25%	86.97%	**82.82%**
CSI-BERT2 replace	79.82%	74.89%	83.07%	86.31%	82.16%	78.65%	80.62%	85.38%	81.36%
CSI-BERT recover	80.98%	75.27%	84.37%	80.41%	81.38%	**83.07%**	81.32%	84.92%	81.46%
CSI-BERT replace	80.21%	74.94%	83.46%	84.37%	82.33%	80.79%	83.33%	85.72%	81.89%
Linear Interpolation	81.78%	75.78%	84.50%	84.33%	78.51%	78.77%	81.35%	84.39%	81.17%
Ordinary Kringing	81.64%	75.78%	80.98%	82.29%	82.00%	79.03%	82.31%	84.49%	81.07%
IDW	**82.55%**	54.94%	83.59%	80.59%	78.21%	80.72%	81.72%	84.06%	78.29%

Table 6: CSI classification performance in WiFall dataset

Experiment Result

❑ CSI Prediction Task

- CSI-BERT2 outperforms other CSI prediction models across all datasets.

Dataset	WiGesture				WiFall				WiCount			
Method \ Metric	MSE	SMAPE	MAPE	Time(s)	MSE	SMAPE	MAPE	Time(s)	MSE	SMAPE	MAPE	Time(s)
CSI-BERT2 (5M)	**3.2942**	**0.1583**	**0.1349**	0.46	**4.8598**	**0.1471**	**0.1347**	0.49	**5.3401**	**0.1726**	**0.1590**	0.46
LSTM (133K)	12.3254	0.2397	0.3967	0.05	7.1495	0.1624	0.1882	0.04	32.3377	0.2547	0.3528	0.05
RNN (33K)	19.4708	0.2877	0.4063	**0.04**	16.9083	0.2424	0.2988	0.04	32.3670	0.2548	0.3534	0.03
GRU (100K)	19.7180	0.2922	0.4243	**0.04**	16.5353	0.2395	0.2963	0.07	39.8108	0.2541	0.3556	**0.02**
Mamba (5M)	12.3281	0.2392	0.3277	0.24	6.4666	0.1532	0.1756	0.12	39.9170	0.2566	0.3524	0.11
OCEAN (126K)	19.6257	0.2925	0.4231	0.05	16.8825	0.2423	0.2978	**0.03**	39.7917	0.2542	0.3548	**0.02**
CV-3DCNN (19K)	11.3017	0.2267	0.3044	**0.04**	8.2616	0.1713	0.1981	**0.03**	42.2662	0.2631	0.3560	**0.02**
ConvLSTM (152K)	19.7038	0.2921	0.4242	**0.04**	16.8935	0.2429	0.2983	**0.03**	39.7709	0.2537	0.3552	**0.02**

Table 7: CSI prediction error

❑ CSI Classification Task

- With the aid of time embedding, CSI-BERT2 can simultaneously process CSI data at different sampling rates.

Data	Training Set: 100Hz+50Hz; Testing Set: 100Hz+50Hz							
Model	**MLP**	**CNN**	**RNN**	**LSTM**	**Chen et al.**	**WiGRUNT**	**CSI-BERT**	**CSI-BERT2**
Gesture Recognition	16.51%	14.54%	15.13%	17.37%	16.59%	74.47%	64.61%	**97.04%**
People Identification	13.47%	15.52%	13.41%	13.47%	13.72%	81.25%	70.83%	**99.54%**
Action Recognition	70.97%	67.67%	70.02%	59.63%	75.18%	66.78%	78.18%	**88.35%**
Fall Detection	81.10%	75.80%	83.92%	84.83%	85.98%	80.72%	92.98%	**93.64%**
People Number Estimation	84.03%	46.82%	81.25%	82.00%	80.09%	76.13%	86.93%	**92.54%**
Data	**Training Set: 100Hz; Testing Set: 50Hz**							
Model	**MLP**	**CNN**	**RNN**	**LSTM**	**Chen et al.**	**WiGRUNT**	**CSI-BERT**	**CSI-BERT2**
Gesture Recognition	69.79%	20.38%	36.25%	27.15%	74.89%	71.37%	79.96%	**97.81%**
People Identification	87.29%	11.85%	82.29%	22.32%	87.22%	85.24%	94.44%	**99.38%**
Action Recognition	68.97%	51.96%	68.07%	60.26%	76.56%	73.21%	84.56%	**88.53%**
Fall Detection	80.83%	76.13%	80.17%	84.38%	78.61%	77.08%	94.19%	**94.32%**
People Number Estimation	82.22%	77.03%	84.43%	42.22%	68.89%	71.11%	89.10%	**94.77%**

Table 8: CSI classification performance under different sampling rate

Ablation Study

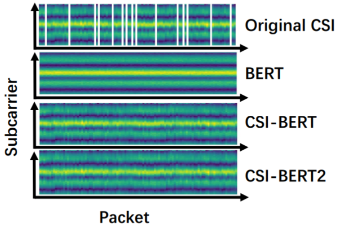
❑ Effect of Pre-training

	w/ Pre-training			w/o Pre-training		
Dataset	**MSE**	**SMAPE**	**MAPE**	**MSE**	**SMAPE**	**MAPE**
WiGesture CSI-BERT	**3.2942**	**0.1583**	**0.1349**	5.3054	0.1962	0.1657
WiFall KNN-MMD	**4.8598**	**0.1471**	**0.1347**	5.0957	0.1595	0.1413
WiCount	**5.4301**	**0.1726**	**0.1590**	6.6868	0.2019	0.1659

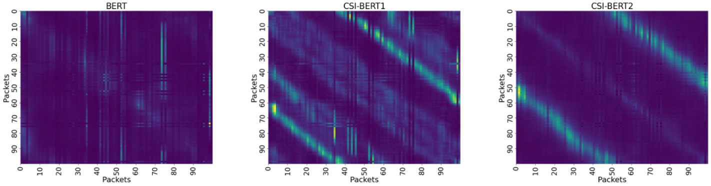
Table 9: CSI-BERT2 performance in CSI prediction task

Figure 17: CSI-BERT2 performance in CSI classification task

- **Effect of Modulation to BERT Structure**
  - The original BERT fails to capture any useful information from CSI, assigning the same value to all positions of the CSI, although it can result in a relatively low loss function value.



(a) CSI recovery result



(a) BERT [20] (sample set 1) (b) CSI-BERT1 [24] (sample set 1) (c) CSI-BERT2 (sample set 1)

(b) Attention map

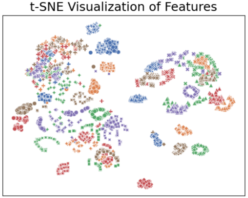
Figure 18: Comparison between CSI-BERT and Original BERT

Outline

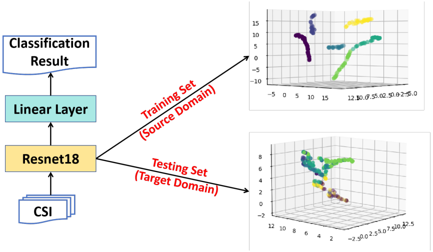
1. Research Overview
2. CSI-BERT: A Multifunctional Framework for CSI Time Series
3. KNN-MMD: An Effective Framework for Practical Cross-Domain Wi-Fi Sensing
4. CrossFi: A Multi-scenario Framework for Cross-Domain Wi-Fi Sensing
5. LoFi: IoT-Enabled Wi-Fi Sensing Deployment
6. Concluding Remarks

Background: Cross-Domain Task

- **Definition**
  - Source Domain: The original domain where the model is trained, containing a significant amount of labeled data that helps the model learn to make predictions.
  - Target Domain: The new domain where the model needs to perform, which may have different characteristics and possibly limited or no labeled data.
- **Influence of Domain Shift**
  - Failure to extract features from the target domain → *Low Model Performance!*



(a) t-SNE of WiGesture dataset



(b) Influence of domain shift

Figure 19: Cross Domain Challenge

Background: Current Methods

Domain Adaptation (DA) Methods

- Metric-based Methods: Utilize distance metrics such as Gaussian distance and cosine similarity.
- Domain Alignment Methods: Focus on aligning the distributions of the source and target domains.
- Learning-based Methods: Include techniques like comparative learning and representation learning.

	Metric-based Method	**Learning-based Method**	**Domain Alignment Method**	**Ours**
Representative Methods	KNN, K-means	Siamese, Triplet Network	MMD, GFK	KNN-MMD
Sensitivity to Quality of Support Set	High	Moderate	**None**	Low
Stability	Low	Low	Low	**High**
Assumption $P_t(y\|x) = P_s(y\|x)$	**No**	Some methods require it	Yes	**No**

Table 10: Comparison of Different DA Methods

Preliminary: Domain Alignment (DAL)

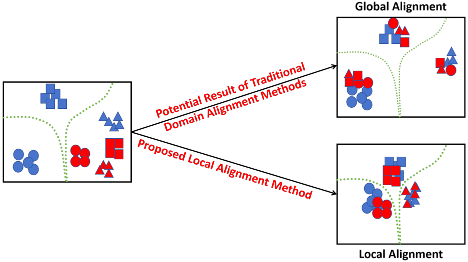
- **Target**: Make the network θ learned in the source domain x_s work in the target domain x_t.

$$\theta = \arg\max_{\theta} P(y_s | x_s; \theta)$$

  - x_s, y_s: input and ground truth from the source domain
  - x_t, y_t: input and ground truth from the target domain
- **Challenge**: Due to the significant domain gap ($P(x_s) \neq P(x_t)$), θ often has low performance in the target domain.
- **DAL Solution**: Find a feature space $F()$ such that $P(F(x_s)) \approx P(F(x_t))$. (i.e., align the distributions of the source and target domains in feature space)

Preliminary: Domain Alignment (DAL)

- **Target**: Make the network θ learned in the source domain x_s work in the target domain x_t.
$$\theta = \arg\max_{\theta} P(y_s|x_s;\theta)$$
  - x_s, y_s: input and ground truth from the source domain
  - x_t, y_t: input and ground truth from the target domain
- **Challenge**: Due to the significant domain gap ($P(x_s) \neq P(x_t)$), θ often has low performance in the target domain.
- **DAL Solution**: Find a feature space $F()$ such that $P(F(x_s)) \approx P(F(x_t))$. (i.e., align the distributions of the source and target domains in feature space)



Preliminary: Domain Alignment (DAL)

❑ **Limitation**: Does DAL really work?

$$
\begin{aligned}
P_s(y|F(x)) &= P_t(y|F(x)) \frac{P_s(F(x)|y)P_s(y)}{P_s(F(x))} \frac{P_t(F(x))}{P_t(F(x)|y)P_t(y)} \\
&= P_t(y|F(x)) \frac{P_t(F(x))}{P_s(F(x))} \frac{P_s(F(x)|y)}{P_t(F(x)|y)} \frac{P_s(y)}{P_t(y)} \\
&= P_t(y|F(x)) \frac{\sum_{y'} P_t(y')P_t(F(x)|y')}{\sum_{y'} P_s(y')P_s(F(x)|y')} \frac{P_s(F(x)|y)}{P_t(F(x)|y)} \frac{P_s(y)}{P_t(y)}
\end{aligned}
$$

The learned θ actually satisfies: $P(y|F(x);\theta) \approx P_s(y|F(x))$. Therefore, if we want θ to work in the target domain, we should ensure that $P_s(y|F(x)) \approx P_t(y|F(x))$.
This implies that what we need to align is $P(F(x)|y)$, not just $P(F(x))$. (the proposed local alignment)

Preliminary: K-Nearest Neighbors (KNN)

- **Basic Idea**: Classify according to the distance to each sample in the Support Set (a very small number of labeled samples in the target domain).
- **Advantage**: Easy & Fast & Interpretability (We can measure confidence based on the distance between the testing sample and the support samples.)
- **Shortcoming**: Accuracy is highly influenced by the quality of the support set.

	KNN			KNN-MMD		
	d=32	**d=64**	**d=128**	**d=32**	**d=64**	**d=128**
n=1, k=1	49%-83%	49%-69%	51%-83%	85%-95%	83%-93%	72%-93%
n=2, k=1	72%-92%	79%-89%	65%-96%	79%-94%	87%-93%	80%-91%
n=2, k=2	65%-76%	58%-82%	49%-83%	88%-95%	84%-92%	88%-91%
n=3, k=1	78%-94%	74%-93%	91%-97%	88%-95%	87%-95%	89%-92%
n=3, k=2	74%-94%	68%-97%	77%-82%	87%-93%	87%-91%	84%-92%
n=3, k=3	64%-93%	68%-94%	74%-91%	91%-96%	86%-90%	90%-94%
n=4, k=1	83%-97%	77%-97%	78%-97%	92%-96%	88%-92%	90%-93%
n=4, k=2	91%-96%	92%-97%	80%-95%	94%-98%	91%-94%	91%-96%
n=4, k=3	77%-97%	73%-97%	82%-92%	85%-93%	89%-93%	90%-94%
n=4, k=4	80%-95%	59%-88%	59%-97%	87%-94%	90%-94%	88%-95%

Table 11: Performance of KNN and KNN-MMD: n denotes the number of shots, k denotes the number of neighbors in KNN, and d denotes the data dimension after reduction using UMAP.

Motivation

The DAL method can achieve high-quality alignment, but we notice that global alignment has low performance guarantees. KNN offers high interpretability but suffers from significant instability. Can we combine the benefits of both?

- **Scenario Setup**:
  - Training Set: Lots of labeled samples from source domain.
  - Support Set: n labeled sample within each category from target domain.
  - Testing Set: Lots of unlabeled samples from target domain. (available during training)
- **Idea**:
  - First, construct a Help Set (samples with pseudo labels in the target domain) using KNN (based on Support Set (labeled samples in the target domain)).
  - Then, achieve local alignment within each category based on the Training Set (source domain) and the Help Set using Multi Kernel Maximum Mean Discrepancy (MK-MMD).

Preliminary: MK-MMD

- **MMD**: A metric to measure the distance between two distributions.
- **MK-MMD**: A practical method to approximate MMD.

$$\mathrm{MMD}[F,p,q] := \sup_{f\in F}\left|\mathrm{E}_p[f(x)] - \mathrm{E}_q[f(x)]\right|$$

$$\begin{aligned}\text{MK-MMD}^2[K,p,q] &:= \sum_{h=1}^{H}\beta_h\left[\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}K_h(x_i^{(p)},x_j^{(p)})\right.\\ &\left.-\frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}K_h(x_i^{(p)},x_j^{(q)}) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}K_h(x_i^{(q)},x_j^{(q)})\right]\end{aligned}$$

- F: the set of all mapping functions in the Reproducing Kernel Hilbert Space (RKHS).
- p,q: two data distributions.
- K: a set of kernel functions.
- n,m: the amounts of data in the two distributions.
- β: a set of weights.

Method Framework

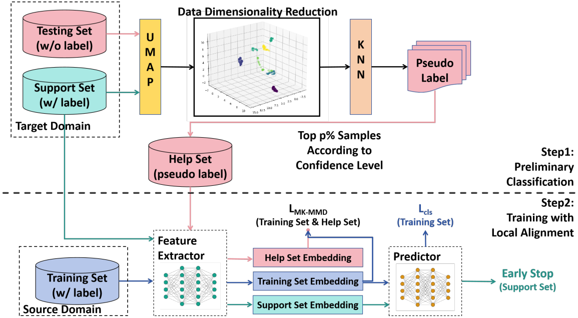


Figure 20: Workflow of KNN-MMD

[1] **Zijian Zhao**, Zhijie Cai, Tingwei Chen, Xiaoyang Li, Hang Li, Qimei Chen, Guangxu Zhu*, "Does MMD Really Align? A Cross Domain Wireless Sensing Method via Local Distribution" (under review, 2025 IEEE/CIC International Conference on Communications in China (ICCC))

[2] **Zijian Zhao**, Zhijie Cai, Tingwei Chen, Xiaoyang Li, Hang Li, Qimei Chen, Guangxu Zhu*, "KNN-MMD: Cross Domain Wireless Sensing via Local Distribution Alignment" (under review, IEEE Transactions on Mobile Computing (TMC))

[3] **Zijian Zhao**, Guangxu Zhu, Qimei Chen, Kaifeng Han, "Method for Object Recognition Using Model Based on Few-Shot Learning and Related Equipment" (Patent Number: ZL202411074110, 2024)

Network Structure

- **Feature Extractor**: ResNet-18
- **Classifier**: MLP

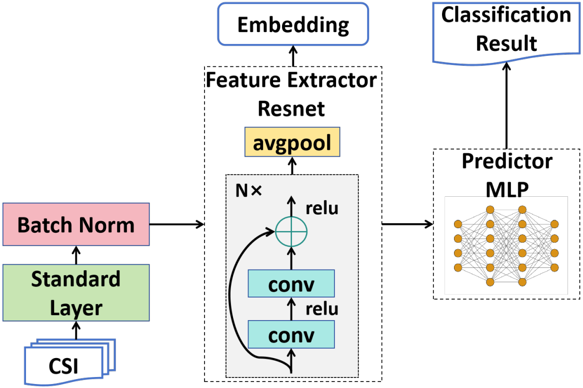


Figure 21: Network architecture

Experiment Setup

❑ **Cross-domain Setting**:

Task	Training Set	Support Set	Testing Set
Gesture Recognition	People ID 1-7	n samples for each gesture in People ID 0	samples excluding support set in People ID 0
People Identification	Action ID 1-5	n samples for each person in Action ID 0	samples excluding support set in Action ID 0
Fall Detection & Action Recognition	People ID 1-9	n samples for each action in People ID 0	samples excluding support set in People ID 0

Table 12: n-shot scenario description

❑ **Ablation Study Setting**: Directly use the help set to fine-tune the trained network on the training set (source domain).

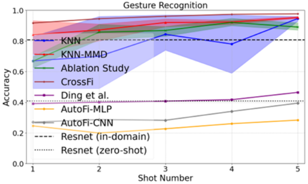
Experiment Result

❑ **One/Zero-shot Comparison**:

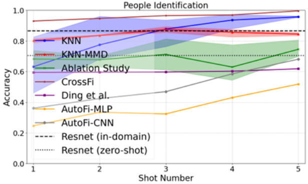
Method	Scenario	Gesture Recognition	People Identification	Fall Detection	Action Recognition	Average Accuracy
Resnet18	in-domain	80.75%	86.75%	91.88%	70.50%	82.47%
	zero-shot	40.84%	70.50%	59.86%	26.00%	49.30%
Siamese	one-shot	70.40%	82.87%	60.62%	38.95%	63.21%
AutoFi (MLP-based)	one-shot	24.62%	24.71%	50.88%	23.59%	30.95%
AutoFi (CNN-based)	one-shot	27.05%	36.14%	48.05%	26.95%	34.55%
Yang et al.	one-shot	67.21%	74.22%	59.75%	48.52%	62.43%
Ding et al.	one-shot	39.14%	70.94%	61.56%	30.37%	50.50%
CrossFi	one-shot	91.72%	**93.01%**	80.93%	49.62%	78.82%
KNN	one-shot	83.02%	82.67%	49.63%	46.87%	65.55%
KNN-MMD (Ours)	one-shot	**93.26%**	81.84%	77.62%	**75.30%**	**82.01%**
Ablation Study	one-shot	69.87%	73.78%	**84.03%**	74.06%	75.44%
MMD	zero-shot	47.92%	67.25%	74.32%	**45.61%**	58.75%
MK-MMD	zero-shot	40.36%	66.47%	72.26%	43.72%	55.70%
DANN	zero-shot	41.41%	67.18%	74.06%	35.99%	54.66%
ADDA	zero-shot	42.71%	65.43%	62.81%	36.08%	51.76%
GFK+KNN	zero-shot	30.79%	51.50%	53.72%	34.17%	42.55%
CrossFi	zero-shot	64.81%	**72.79%**	**74.38%**	40.46%	**63.11%**
Tian et al.	zero-shot	**68.13%**	55.86%	61.72%	42.10%	56.95%
EEG	zero-shot	59.75%	64.63%	69.53%	42.15%	59.02%

Table 13: One-shot experimental results

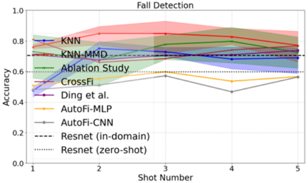
- **Few-shot Comparison**:



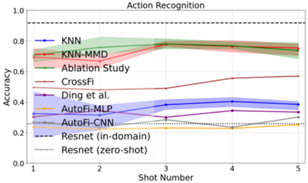
(a) Gesture Recognition



(b) People Identification



(c) Fall Detection

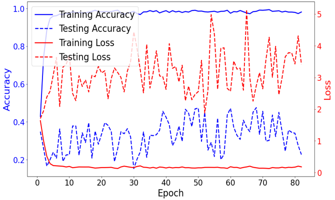


(d) Action Recognition

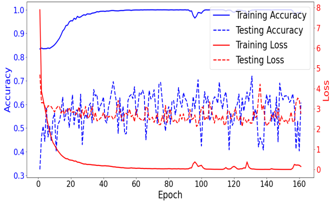
Figure 22: Few-shot experiment result

Experiment Result

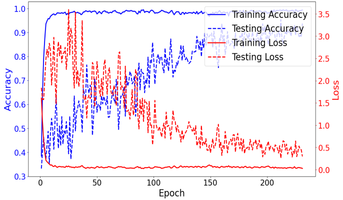
- **Training Process Visualization**:
  - KNN-MMD exhibits the highest accuracy and stability.
  - Even when the training accuracy approaches nearly 100%, the testing accuracy of KNN-MMD continues to increase steadily, which can be attributed to local alignment.



(a) MK-MMD



(b) Siamese

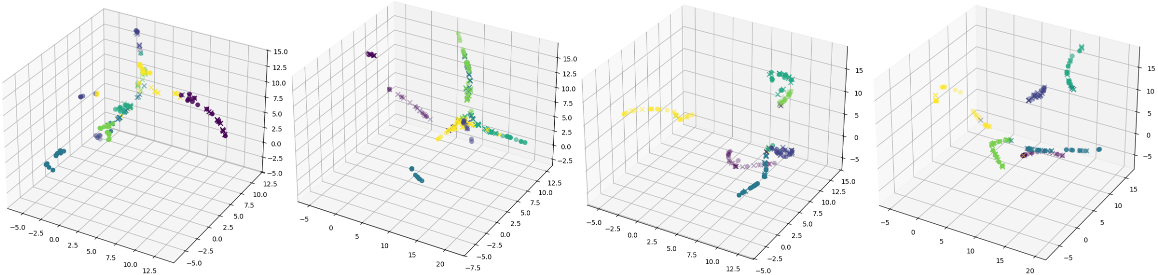


(c) KNN-MMD (shot number=1)

Figure 23: training process of three cross-domain methods

Experiment Result

❑ **Embedding Result Visualization**:



(a) ResNet (no alignment)

(b) MK-MMD (global alignment)

(c) KNN-MMD (shot number=1)

(d) Theoretical Upper Bound (local alignment)

Figure 24: Data dimension reduction results of embedding results from different models: Different colors represent different categories. The circles represent samples from the source domain, and the crosses represent samples from the target domain.

Outline

1 Research Overview

2 CSI-BERT: A Multifunctional Framework for CSI Time Series

3 KNN-MMD: An Effective Framework for Practical Cross-Domain Wi-Fi Sensing

4 CrossFi: A Multi-scenario Framework for Cross-Domain Wi-Fi Sensing

5 LoFi: IoT-Enabled Wi-Fi Sensing Deployment

6 Concluding Remarks

Background: Current Cross-domain Wi-Fi Sensing Methods

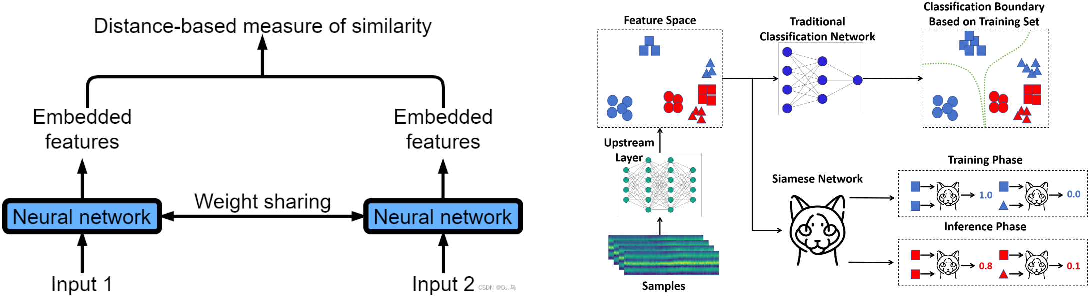
- **Few-shot Methods**: Require some labeled samples from the target domain, which cannot always be satisfied in practice.
- **Zero-shot Domain Generalization (DG) Methods**: Zero-shot methods do not require any labeled data from the target domain. However, DG typically requires multiple different source domains.
- **Zero-shot DAL Methods**: Currently, there are no methods specifically aimed at Wi-Fi sensing. Additionally, methods in machine learning, such as MK-MMD, have been shown to be limitedly efficient in our KNN-MMD work.

$\Rightarrow$ Research Gap: Zero-shot Method for Single Source Domain Scenario

Motivation: Siamese Network

- **Siamese Network**:
  - Train a neural network to extract general features using source domain data.
  - Successfully calculates the similarity of two samples from the target domain.

$\Rightarrow$ Siamese networks demonstrate excellent performance in one-shot tasks. Can we expand this approach to more general scenarios?



(a) Network architecture (b) Working principle

Figure 25: Siamese network

- **Task Scenarios**:
  - In-domain
  - Few-shot
  - One-shot
  - Zero-shot
  - New-class
- **Process**:
  - Template generation
  - Classifying by comparison

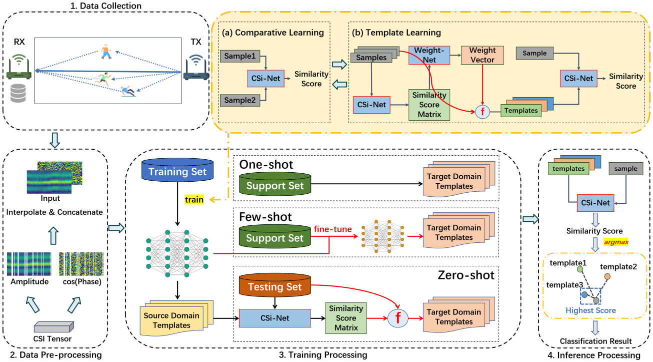


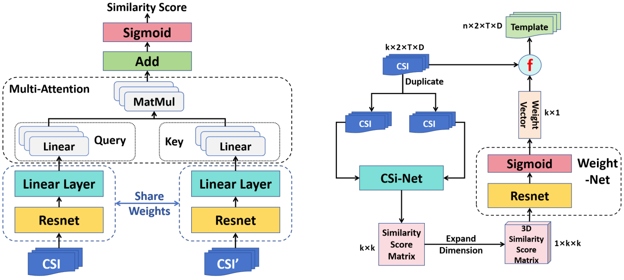
Figure 26: Workflow of CrossFi

[1] **Zijian Zhao**, Tingwei Chen, Zhijie Cai, Xiaoyang Li, Hang Li, Qimei Chen, Guangxu Zhu*, "CrossFi: A Cross Domain Wi-Fi Sensing Framework Based on Siamese Network" (IEEE Internet of Things Journal (IOT-J))

[2] **Zijian Zhao**, Guangxu Zhu, Kaifeng Han, Xiaoyang Li, Hang Li, "Method for Classifying Data Using Model Based on Few-Shot Learning and Related Equipment" (Patent Application Number: 2024108392137, 2024)

Network Structure

- **Cross-domain Siamese Network (CSi-Net)**:
  - Extract features using ResNet.
  - Calculate similarity with QK-attention.
- **Weight-Net**:
  - Calculate sample quality based on the similarity computed by CSi-Net.
  - Generate templates using sample quality as mixing weights.



(a) CSi-Net (b) Weight-Net

Figure 27: Network structure

Zero-shot Template Generation Method

- **Select the samples with the highest similarity to the templates from the source domain as the templates for the target domain.**

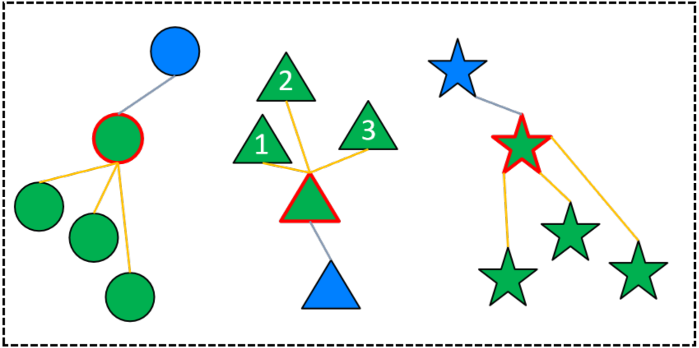


Figure 28: Template generation method of target domain in zero-shot scenario

Experiment Result

- **In-domain Scenario**:

Method	Gesture Recognition	People Identification
ResNet-18	80.75%	86.75%
WiGRUNT	70.46%	97.86%
Zhuravchak et al.	56.93%	88.61%
Yang et al.	43.75%	87.78%
Ding et al.	43.75%	61.72%
AutoFi (MLP-based)	48.22%	89.45%
AutoFi (CNN-based)	89.55%	97.74%
CSI-BERT	74.55%	97.92%
CrossFi	**98.17%**	**99.97%**

Table 14: In-domain experiment

- **One-shot Scenario**:

	Cross Domain		New Class	
Method	Gesture Recognition	People Identification	Gesture Recognition	People Identification
Siamese	70.40%	82.87%	66.41%	80.92%
AutoFi (MLP-based)	24.62%	24.71%	43.82%	81.75%
AutoFi (CNN-based)	27.05%	36.14%	74.13%	**86.58%**
Yang et al.	67.21%	74.22%	58.74%	49.00%
Ding et al.	39.14%	59.50%	–	–
CrossFi w/ MK-MMD	**91.72%**	**93.01%**	80.62%	73.66%
CrossFi w/o MK-MMD	84.47%	87.50%	**84.75%**	81.97%

Table 15: One-shot experiment

Experiment Result

- **Few-shot Experiment**:

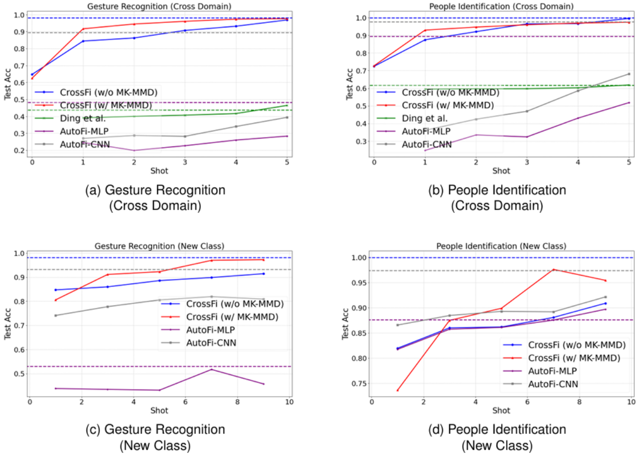


Figure 29: Few-shot experiment

Experiment Result

- **Zero-shot Experiment**:

Method	Gesture Recognition	People Identification
ResNet-18	40.84%	70.50%
ADDA	42.71%	65.43%
DANN	41.41%	67.18%
MMD	47.92%	67.25%
MK-MMD	40.36%	66.47%
GFK+KNN	30.79%	51.05%
CrossFi w/ MK-MMD	62.60%	**72.79%**
CrossFi w/o MK-MMD	**64.81%**	72.46%

Table 13: Zero-shot experiment

- **Expanded Experiment**:

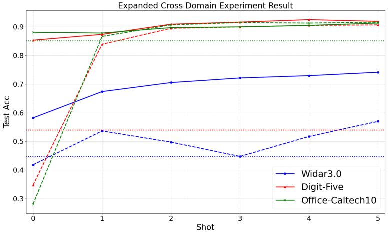


Figure 30: Expanded experiment

Ablation Study

- **Effect of Similarity Calculation Methods**:

Gesture Recognition				
	Full Shot	**One Shot**	**Zero Shot**	**New Class**
Gaussian Distance	95.58%	**84.47%**	20.97%	74.49%
Cosine Similarity	91.64%	77.17%	46.44%	74.36%
Multi-Attention	**98.17%**	62.51%	**64.81%**	**84.75%**

People Identification				
	Full Shot	**One Shot**	**Zero Shot**	**New Class**
Gaussian Distance	99.74%	**87.50%**	38.48%	80.53%
Cosine Similarity	**99.97%**	83.72%	71.16%	74.60%
Multi-Attention	**99.97%**	68.04%	**72.46%**	**81.97%**

Table 16: Ablation study in similarity computation method

- **Effect of Template Generation Methods**:

	Gesture Recognition		**People Identification**	
	Full Shot	**Zero Shot**	**Full Shot**	**Zero Shot**
Random	94.79%	58.83%	98.17%	60.42%
Average	91.90%	56.39%	99.74%	68.95%
Weight-Net	**98.17%**	**64.81%**	**99.97%**	**72.46%**

Table 17: Ablation study in template generation method

- **Attention & Gaussian Distance – Which is better?**
  - The Gaussian distance method performs better when the source domain and target domain have a high similarity.
  - When the domain gap is large, the attention-based method can capture the relationship between the two domains more effectively.

Source Domain: ID 0				
Target Domain ID	**4**	**6**	**7**	**5**
Multi-Attention	94.79%	71.57%	93.29%	65.40%
Gaussian Distance	82.18%	66.94%	96.87%	77.34%
Performance Gap	-12.61%	-4.63%	3.58%	11.94%
Benchmark ResNet	19.51%	25.13%	31.82%	39.92%

Table 18: One-shot experiment

Discussions

❑ Why is Weight-Net useful?

Gaussian Noise Variance	0	2	4	6	8	10
Sample Quality Score	0.4690	0.4673	0.4422	0.4474	0.4456	0.4278

Table 19: Relationship between sample quality score and added Gaussian noise standard deviation

❑ Model Scale vs. Model Performance

Backbone Model	Model Parameter	Model Size	GPU Occupation	WiGesture	Office-Caltech10
ResNet34	4.26M	163.39MB	1.26GB	89.18%	89.05%
ResNet50	4.72M	180.96MB	2.36GB	87.36%	89.91%
ResNet101	8.53M	326.46MB	3.71GB	87.33%	89.07%
ResNet18	2.24M	85.66MB	0.93GB	80.42%	87.78%
Integer Quantization	2.24M	21.62MB	0.63GB	80.36%	84.03%
Pruning (20%)	1.79M	85.66MB	0.93GB	80.22%	84.81%

Table 20: Comparison of complexity and performance in one-Shot cross-domain scenario

Outline

1. Research Overview
2. CSI-BERT: A Multifunctional Framework for CSI Time Series
3. KNN-MMD: An Effective Framework for Practical Cross-Domain Wi-Fi Sensing
4. CrossFi: A Multi-scenario Framework for Cross-Domain Wi-Fi Sensing
5. LoFi: IoT-Enabled Wi-Fi Sensing Deployment
6. Concluding Remarks

Real-time Wi-Fi Sensing System

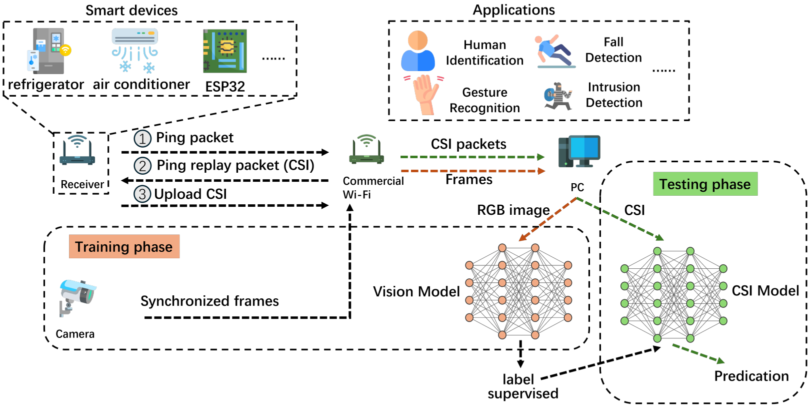


Figure 31: Workflow of real-time Wi-Fi sensing system

[1] **Zijian Zhao**, Guangxu Zhu, Shen Chao, Shi Qingjiang, Han Kaifeng, "Personnel Detection Method, Device, Electronic Equipment, and Storage Medium" (Patent Application Number: 2024105419689, 2024)

Challenge: Lack of Data

- **Existing Wi-Fi Localization Dataset Collection Methods**:
  - LiDAR-based Method: precise but expensive
  - Manual Tagging: coarse-grained
- **Existing Heterogeneous Public Wi-Fi Sensing Datasets**:
  - Different Sampling Rates
  - Different Devices
  - Different Data Formats
  - Different Domains
  - ...

$\Rightarrow$ A cheap and easy method is needed for users to collect their own datasets quickly.

Method: Vision-Aided Wi-Fi Localization Dataset Collection System

- **Workflow of LoFi**:
  - Step 1: Collect CSI and image data simultaneously.
  - Step 2: Localize the person in pixel space.
  - Step 3: Transfer pixel space to physical space.
  - Step 4: Align CSI and image by timestamp.

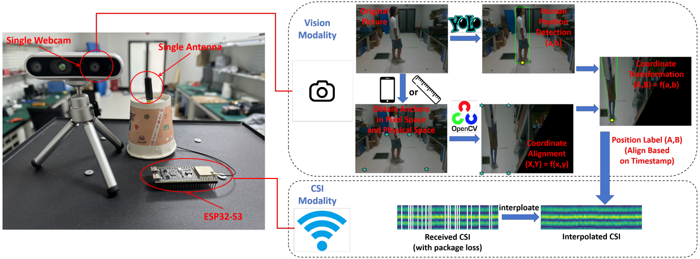
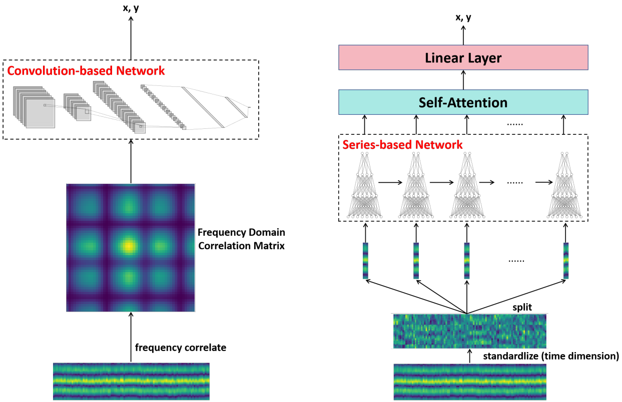


Figure 32: Workflow of LoFi

[1] **Zijian Zhao**, Tingwei Chen, Fanyi Meng, Zhijie Cai, Hang Li, Xiaoyang Li, Guangxu Zhu*, "LoFi: Vision-Aided Label Generator for Wi-Fi Localization and Tracking Sensing" (under review, IEEE Wireless Communications Letters (WCL))

Benchmark Methods in LoFi



(a) Convolution-based methods (b) series-based network

Figure 33: Benchmark methods for LoFi dataset

Benchmark Methods in LoFi

- Our experiment first demonstrates the potential of Wi-Fi localization using a single RX-TX pair with a single antenna.

Methods / Metric	Convolution-based Methods		Series-based Methods			
	CNN	**ResNet**	**RNN**	**GRU**	**LSTM**	**CSI-BERT**
Error Mean	0.8745	**0.5830**	0.8705	0.9413	0.8643	0.6991
Error Standard Deviation	0.3177	0.3475	0.2802	0.3217	**0.2475**	0.3063
Classification Accuracy (6 classes)	31.99%	55.50%	53.29%	49.41%	53.50%	**60.07%**
Classification Accuracy (4 classes)	42.03%	**62.54%**	61.92%	56.00%	62.15%	61.93%
Classification Accuracy (2 classes)	62.84%	82.98%	**84.47%**	73.56%	73.98%	75.63%

Table 21: Experiment result

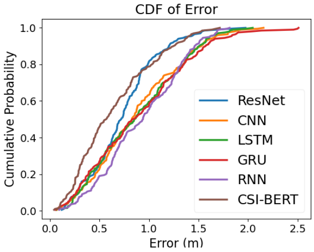


Figure 33: Cumulative Distribution Function (CDF) of the error

Outline

1 Research Overview

2 CSI-BERT: A Multifunctional Framework for CSI Time Series

3 KNN-MMD: An Effective Framework for Practical Cross-Domain Wi-Fi Sensing

4 CrossFi: A Multi-scenario Framework for Cross-Domain Wi-Fi Sensing

5 LoFi: IoT-Enabled Wi-Fi Sensing Deployment

6 Concluding Remarks

Concluding Remarks

- **Project 1**: CSI-BERT: A Multifunctional Framework for CSI Time Series
  - A foundation model for multiple CSI-related tasks, including recovery, prediction, and classification.
  - Proposed CSI-embedding and time-embedding layers improve the model's capacity to capture the inner relationships of CSI sequences.
- **Project 2**: KNN-MMD: An Effective Framework for Practical Cross-Domain Wi-Fi Sensing
  - A few-shot method for cross-domain Wi-Fi sensing.
  - Proves that in domain alignment, what we need is actually local alignment rather than global alignment.

Concluding Remarks

- **Project 3**: CrossFi: A Multi-Scenario Framework for Cross-Domain Wi-Fi Sensing
  - A multi-scenario Wi-Fi sensing method for in-domain, few-shot cross-domain, few-shot new-class, and zero-shot cross-domain scenarios.
  - Improves the performance of the Siamese network using an attention mechanism and expands its application scenarios with Weight-Net.
- **Project 4**: LoFi: IoT-Enabled Wi-Fi Sensing Deployment
  - A vision-aided method for Wi-Fi localization and tracking dataset collection.
  - Reduces the complexity and expense of Wi-Fi sensing dataset collection.

Future Extensions

- **Develop a Large Foundation Model for Wi-Fi Sensing**
  - Develop a heterogeneous large foundation model to make full use of public datasets with different data structures.
  - Explore the scaling law in Wi-Fi sensing (especially the zero-shot ability).
- **Transfer Knowledge from Other Modalities to Wi-Fi**
  - Develop a cross-modal knowledge distillation method to transfer knowledge from strong modalities like images to the CSI modality.
  - Improve the robustness of Wi-Fi sensing by learning from other modalities.

[1] Haolong Chen, Hanzhi Chen, **Zijian Zhao**, Kaifeng Han*, Guangxu Zhu*, Yichen Zhao, Ying Du, Wei Xu, Qingjiang Shi, "An Overview of Domain-specific Foundation Model: Key Technologies, Applications and Challenges" (under review, Science China Information Sciences (SCIS))

References I

Chen, Tingwei et al. (2024). "Deep learning-based fall detection using commodity Wi-Fi". In: *Journal of Information and Intelligence.*

Wang, Zhuang et al. (2023). "Radio map construction based on BERT for fingerprint-based indoor positioning system". In: *EURASIP Journal on Wireless Communications and Networking* 2023.1, pp. 1–18.

Guo, Baoshen et al. (2022). "WEPOS: Weak-supervised indoor positioning with unlabeled Wi-Fi for on-demand delivery". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6.2, pp. 1–25.

Sun, Xu et al. (2021). "BERT-ADLOC: A secure crowdsourced indoor localization system based on BLE fingerprints". In: *Applied Soft Computing* 104, p. 107237.