MuSAC: Toward Communication-Free Sensory Data Acquisition

Sijie Ji, Yiwei Wang, Lixiang Lian

***Abstract*—Sensing and communication are at the core of the Internet of Things, which usually function independently. For example, a smartphone can communicate over Wi-Fi or cellular networks while continuously acquiring sensory data from the environment through various sensors. This paper presents a novel framework, MuSAC (Mutualistic Sensing and Communication), which seamlessly integrates the collection of sensory data with existing communication systems, without adding any additional communication overhead (i.e., communication-free). The framework leverages the mutualistic relationship between specific communication data and sensory data to effectively crowdsource heterogeneous sensory data without harming communication performance in practical distributed systems. To embed massive sensory data into the current transmission of communication data, MuSAC presents novel neural networks to distill universal features from the raw data for compression at the sender side and then extract invariant features on the server side. By doing so, MuSAC eliminates additional communication overhead for sensory data collection while also mitigating privacy concerns and data heterogeneity in crowdsensing. To evaluate the performance of MuSAC, we first conduct system-level simulations in a distributed environment by embedding public human activity sensing datasets into cellular massive MIMO communication processes. The results demonstrate that MuSAC effectively supports heterogeneous sensory data collection over existing wireless communication links at zero additional communication overhead. We further implement a functional prototype of the MuSAC system using off-the-shelf devices, where real-world sensory data are seamlessly integrated into WiFi transmissions. This practical implementation demonstrates the viability of communication-free sensory data acquisition in real-world scenarios, paving the way for broader applications of MuSAC.**

***Index Terms*—Mobile Crowdsensing, CSI Feedback, Communication-Free Sensing, Distributed Systems.**

I. Introduction

With the rapid development of artificial intelligent (AI) technologies, model training for various sensing applications requires large volumes of sensory data from diverse edge devices. The sensory data collection requires the central server and edge devices to establish a separate communication link, resulting in significant communication overhead. For example, as foundation models increasingly drive various intelligent services, their training requires massive and diverse multi-modal sensory data collected from a wide range of edge devices. This data serves as critical foundation for generalization across various downstream sensing and inference tasks. In order to solve the problem of large sensory data collection overhead, a solution like federated learning (FL) protects privacy, yet does not address the communication overhead as it requires multi-round exchanges of data or model parameters between devices and the server, rendering the communication inevitable and significant. Furthermore, in terms of performance, it is more promising to train a model with a large amount of sensory data on a central server, rather than on Internet of Things (IoT) devices with limited computing resources. Especially, the straggler effect of FL makes the performance confined by the computation capabilities of mobile devices [1]–[3]. Therefore, how to crowdsource massive sensory data with little communication overhead and preserved privacy remains a key enabler.

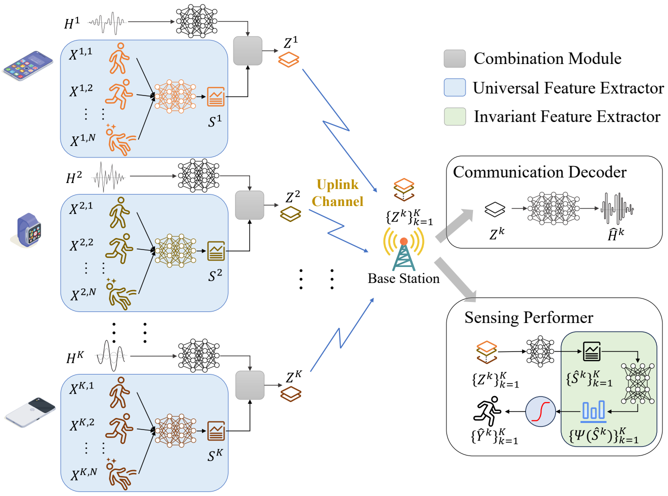


Fig. 1: The design of MuSAC for distributed mobile crowdsensing without additional communication overhead.

Given that edge devices, such as smartphones, smart vehicles, inherently communicate with the central server, e.g., base station (BS), stable communication links already exist with regular data packet transmissions. By embedding sensory data into these existing communication packets, it is possible to achieve communication-free sensory data acquisition and realize free sensing. In this paper, we propose a novel framework, **M**utualistic **s**ensing **a**nd **c**ommunication (MuSAC) for communication-free sensory data acquisition, which aims to efficiently transmit heterogeneous sensory data by reusing existing communication links without incurring additional communication overhead. Our key observation is that sensory

Sijie Ji and Yiwei Wang are co-first authors. Corresponding author: Lixiang Lian.

Sijie Ji is a Schmidt Science Fellow with Division of Engineering and Applied Science, California Institute of Technology, CA 91125, USA (email:sijieji@caltech.edu).

Yiwei Wang and Lixiang Lian are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (email: wangyw12024@shanghaitech.edu.cn; lianlx@shanghaitech.edu.cn).

data can be embedded into the standard and essential data packets designated for sending channel feedback information. Following established communication standards, existing communication systems regularly feedback channel relevant information through uplink transmission to function downlink beamforming. For example, in 5G and beyond cellular networks, distributed mobile devices need to explicitly provide downlink channel state information (CSI) feedback to the BS to enable massive multiple input multiple output (MIMO) techniques [4], [5]. Similarly, from IEEE 802.11ac onwards, WiFi clients transmit compressed beamforming reports (CBR) through action frames to the WiFi Access Point (AP) to enable beamforming [6]. Such channel-relevant information, except for communication, has also been used for wireless sensing for decades, demonstrating an inherent correspondence between such communication data (ComData) and sensory data (SenData). Particularly, we conducted proof-of-concept studies (Section V), and the initial results show that the shared structures or the underlying correlation within SenData and ComData can be automatically discovered, extracted, and utilized to enhance joint compression and reconstruction performance. This mutualistic relationship provides a distinct opportunity for seamless integration in MuSAC, enhancing the overall effectiveness of sensory and communication data transmission.

At a high level, MuSAC seeks this opportunity to ride SenData on top of ComData packets, without impacting the ComData transmission or increasing the packet length. By doing so, MuSAC eliminates additional communication overhead of sensory data transmission, while minimizing the loss of ComData and enhancing the overall transmission efficiency of both SenData and ComData in the entire system. Particularly, MuSAC designs a novel joint compressor to compress the SenData and ComData jointly at the sender side (i.e., edge devices) and a corresponding decompressor to reconstruct both types of data on the server side.

Translating the core idea into a practical MuSAC system faces two major challenges. First, ComData and SenData differ in data volume and data complexity. While ComData enjoys a well-established compression and decompression mechanism, how to embed a large volume of SenData remains open. To overcome this challenge, MuSAC employs contrastive learning (CL) to distill knowledge from the complex SenData at the encoder side. This process helps to reduce the data volume and extract structured universal features, making it easier to compress and facilitating overall compression. Second, ComData can be decoded with minimal distortion thanks to mature mechanisms for removing noise and hardware-induced undesired heterogeneity. Differently, SenData is more sensitive to undesired data heterogeneity caused by the hardware and user placement. To accomplish a crowdsensing task, such undesired heterogeneity needs to be eliminated. To do so, MuSAC formulates a bi-level optimization problem at the server side and develops an invariant feature extractor that can learn invariances across diverse distributions stemming from different devices. Consequently, MuSAC eliminates additional communication overhead, preserves data privacy, and mitigates data heterogeneity, promising massive data collection toward potential IoT applications.

A. Our Contributions

Our key contributions are summarized as follows:

- We first identify the mutualistic relationship between sensory data and specific communication data and leverage this opportunity to integrate SenData into the link where only ComData was previously transmitted, which sheds light on the protocol design of the next-generation communication system.
- We design the MuSAC framework that efficiently crowdsources massive heterogeneous sensory data without additional communication overhead in practical distributed settings.
- We conduct experimental simulations in a cellular massive MIMO communication scenario, where public sensory datasets are embedded into the CSI feedback process of the wireless communication system. The dataset features a highly heterogeneous setting, involving 120 different users and 77 different device models. Simulation results show that MuSAC can improve the data transmission efficiency by 89.47% and 88.89% for ComData and SenData at a compression ratio of 1/4, and by 85.80% and 86.76% at a compression ratio of 1/32, respectively, validating its effectiveness and superiority in supporting large-scale and heterogeneous sensory data collection.
- We implement the MuSAC system using commodity edge devices, a central server, and sensor devices. The SenData is collected using cameras in both indoor and outdoor environments, involving 200 participants and diverse behavior patterns. This data is embedded into WiFi CBR packets via MuSAC framework for concurrent transmission. Practical test results show that MuSAC enhances the data transmission efficiency by 102.46% and 92.91% for ComData and SenData, respectively, at a compression ratio of 1/20, and by 95.94% and 93.07% at a compression ratio of 1/32, averaged across different environments. The successful real-world implementation verifies that MuSAC offers a practical and effective solution for communication-free sensory data collection in AI-driven applications.

B. Enhancements Over the Conference Version

A preliminary version of this work has been published in [7]. This journal paper substantially extends that version with new theoretical insights, real-world system implementation, and expanded evaluation. The major enhancements are summarized below.

- **Real-world system implementation and evaluation:** The conference version validated MuSAC through simulations in a massive MIMO setting. In this paper, we move beyond simulation to a full-scale implementation on commodity hardware, deploying MuSAC in both indoor and outdoor environments. Extensive real-world experiments confirm MuSAC's effectiveness in achieving zero additional communication overhead for sensory data collection.

- **Strengthened theoretical foundation:** Building on the initial concepts of mutualistic relationship between sensing and communication, we provide a deeper theoretical analysis by formulating and proving the *Mutualism-Driven Compression Gain* theorem. This formally proves that sufficient mutual information enables sensory data transmission without additional communication overhead, offering a rigorous theoretical basis for MuSAC's cost-efficiency. A new experiment for human activity recognition using image data is added to validate the proposed theory.
- **Expanded Discussion and Conclusion:** We introduce new use cases of MuSAC, i.e., large language models (LLM)-powered applications, intelligent transportation, and smart healthcare, underscoring the broad applicability and system-level impact of MuSAC in diverse settings. We enrich the discussion and conclusion sections with reflections on deployment challenges, limitations, and future research directions.

II. Related Work

Mobile Crowdsensing: Mobile crowdsensing (MCS) serves a crucial role in building intelligent IoT applications but is hampered by a well-known issue: the sensing devices continuously generate a large amount of data, which consumes many resources (e.g., bandwidth, energy, and storage) [8]. Researchers have proposed plenty of strategies to solve these challenges. For example, many studies have identified significant redundancy in sensory data content, leading to the development of quality-measuring algorithms aimed at selecting data for transmission [9], [10]. However, such algorithms often sacrifice the quality of the final sensing application. A comprehensive survey paper dedicated to cost-effective MCS has provided an extensive review of the current landscape [11]. Different from all the existing MCS systems, MuSAC collects data through existing communication link without imposing additional communication overhead and employs CL to eliminate data redundancy without sacrificing the quality.

Communication Efficiency: Communication poses a known bottleneck in large model training. Many deep learning approaches rely on the centralized data collection, which, however, can induce privacy leakage and significant communication overhead. FL has been widely studied due to its privacy-awareness learning mechanism [12], [13]. However, in FL, the server and clients need to exchange the model parameters intensively during the training, leading to enormous communication overhead. Existing works address communication efficiency in FL from three perspectives: reducing the number of clients [1], [14], decreasing transferred message size [15] and decreasing the number of communication rounds [16]. However, these methods often suffer from significant performance sacrifice, including slow convergence, learning bias, or diminished model accuracy. Instead, the learning paradigm of MuSAC does not require multiple rounds of two-way communication between the mobile devices and server. It leverages the uplink transmission of communication signal, thereby eliminating additional communication overhead.

Sensing with Data Heterogeneity: Data residing across devices is intrinsically heterogeneous (e.g., different data modalities and data distributions). FL is essentially a distributed approximation of centralized learning, which has been proved to suffer poor performance when data is heterogeneous [17]. Many endeavors have been done to address the data heterogeneity in FL, including personalization by fine-tuning the global model to personalize the model on each device using its local data [18], contextualization by adding user context into FL [19], knowledge distillation [20], regularization-based methods [21] and gradient blending-based methods [22], [23], etc. However, these approaches either suffer from overfitting issues (such as personalization) or induce high computation and communication costs (such as contextualization or knowledge distillation). In this paper, MuSAC directly removes the data heterogeneity of SenData caused by different devices and user behaviors by formulating a bi-level optimization problem to extract the invariant features, such that the learned model has good out-of-distribution generalizability.

Semantic Communication: As one of the key technologies in 6G research, semantic communication has been extensively studied across multiple data modalities, such as text [24], video [25], and audio [26], to reduce the communication overhead of transmitting sensory data. However, because semantic communication treats sensory data as standalone payloads to be sent over conventional channels, dedicated links must still be established for their transmission, thereby consuming additional resources. In practical communication systems, various communication tasks occur continuously, such as CSI feedback in traditional wireless systems. Notably, CSI inherently encodes rich environmental information. This natural correlation between sensory data and CSI can be leveraged to embed the sensory information implicitly within existing communication procedures without incurring additional communication overhead. Nevertheless, current semantic communication methods overlook this mutualism and transmit sensory data independently.

Joint sensing and communication: Joint Sensing and Communication (JSAC) provides a unified framework that integrates communication and sensing within a single system by jointly optimizing waveform design, resource allocation, and inference algorithms [27]. Overall, JSAC primarily concentrates on improving the quality of locally perceived sensory data by optimizing sensing performance for specific tasks (e.g., localization, detection) at dual-functional devices, while balancing sensing and communication requirements. However, this line of work largely overlooks the downstream processing of sensed data, including how such data are transmitted, fused, or leveraged across the network for tasks such as large-scale model training. In contrast, the proposed MuSAC framework extends beyond local perception and addresses how distributed sensory data can be efficiently collected and integrated across the network to enable collaborative learning.

Opportunistic integrated sensing and communication: Opportunistic integrated sensing and communication (OISAC) is an emerging paradigm in next-generation wireless networks that enables the dual use of communication signals for both data transmission and environmental sensing. OISAC leverages existing communication signals (e.g. millimeter wave channels, WiFi signals) and infrastructure to perform sensing functions without additional spectrum, power, or hardware overhead. OISAC has now been applied to various fields, including weather monitoring [28]–[30], activity recognition [31], and health monitoring [32]. However, OISAC relies solely on communication signals for environmental sensing, which limits its sensing capability to a narrow range of sensing tasks due to the absence of domain-specific sensory information. In contrast, MuSAC offers a communication-free sensory data collection solution, where the data can span diverse modalities and contain rich semantic content beyond what is available from wireless channels. This design allows MuSAC to support more complex and domain-specific sensing tasks.

III. Applications of MuSAC

MuSAC leverages the existing communication links to embed sensory data into ongoing communication flows, making it particularly suitable for real-world scenarios where both communication and sensing demands naturally coexist. By embedding the sensory data into the communication packets, MuSAC enables free sensing. This is grounded in the mutualistic relationship between SenData and ComData, where both types of data can reinforce each other within the same transmission process. In this section, we provide the use cases of MuSAC in several AI-driven applications, including large language models training [33], intelligent transportation systems [34], [35], and smart healthcare [36].

A. Large Language Models

With the advent of 6G mobile edge intelligence, LLM are being deployed at cellular BSs to support a variety of intelligent applications. Training these LLM requires collecting large-scale, diverse sensory data from many different sensors. Moreover, to provide personalized services, edge devices (e.g., smartphones and smartwatches) frequently send real-time sensor data to cloud servers for model fine-tuning or inference.

Most edge devices already communicate continuously with BSs, necessitating CSI feedback through uplink channel to assist downlink transmission design. MuSAC offers a way to embed LLM-related sensory data within this CSI feedback process. Since sensory data collected by edge devices is highly environment-dependent, it tends to be strongly correlated with the wireless channel state. This inherent correlation makes MuSAC a feasible approach for supporting the training (and inference) of foundation models (FMs) like LLM without requiring separate sensing transmissions. For example, consider an LLM-based fall detection application deployed at the BS. If a user accidentally falls while carrying a mobile device, both the wireless channel conditions and the device's inertial sensor readings (from its Inertial Measurement Unit (IMU)) will change. Using MuSAC, the device can jointly feed back its IMU data and CSI to the BS via the uplink channel. The BS decodes this combined data and feeds the IMU into the LLM for analysis, which promptly issues an accident warning. This enables immediate, continuous, and unobtrusive detection without requiring any additional sensing infrastructure.

B. Intelligent Transportation

In the domain of intelligent transportation systems, contemporary vehicles are equipped with dedicated communication modules that support Vehicle-to-Everything (V2X) services via standards interfaces such as PC5 (for direct V2V/V2I communication) and Uu (for cellular Vehicle-to-Network (V2N) communication). These modules enable the exchange of cooperative awareness messages, traffic updates, and safety-critical information with other vehicles and roadside infrastructure. Meanwhile, vehicles and roadside units (RSUs) are equipped with various sensors (e.g., lidar, radar, and camera) for environmental perception, including pedestrain detection, obstacle recognition, and road condition monitoring. In particular, RSUs commonly communicate with a BS or edge server via the Uu interface of cellular networks (e.g., 4G/5G) to periodically upload the traffic statistics. These statistics, such as vehicle flow, speed profiles, and congestion levels, are essential for dynamic traffic control. To support efficient transmission scheduling, the BS collects CSI feedback from RSUs as part of the standard cellular communication protocol. MuSAC provides an opportunity to embed addtional sensory data, such as detected pedestrain activity or road condition into the CSI feedback process. Note that, both the wireless channel state and sensory data change in response to the environmental changes (e.g., when a vehicle approaches the RSUs), leading to a strong correlation between two types of data. By exploiting such correlations, the Uu transmission of RSUs can achieve concurrent transmission of both traffic statistics through conventional communication procedures and embedded environmental data through MuSAC, without establishing additional sensing link.

C. Smart Healthcare

In smart healthcare application, continuous monitoring of patient activity is essential for timely detection of emergencies or medical diagnoses. Wearable devices, such as smartwatches, fitness bands or health patches, are widely used to collect real-time physiological and motion data. These devices maintain regular wireless communications with a nearby AP, smartphone or home gateway for notification, health data synchronization or firmware update. MuSAC enables the embedding of additional sensory data into existing communication processes without requiring separate transmission channels. For example, when a smartwatch transmits normal communication packets to a home gateway or mobile phone through WiFi or cellular links, MuSAC can encode other information, such as photolithography (PPG) data, into the channel feedback signals (e.g., CSI in cellular or CBR in WiFi). Since the uplink feedback of channel state is continuously happening, embedding sensory data through MuSAC avoids the dedicated sensing transmission. Moreover, due to the high sensitivity of health-related data, transmitting it implicitly within standard

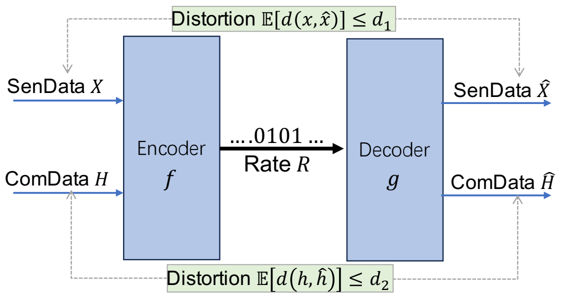


Fig. 2: Rate distortion encoder and decoder.

wireless signaling offers inherent privacy perservation. The fusion of communication data and sensory data also ensures low latency and real-time transmission, which is essential for timely detection of emergencies such as falls or abnormal physiological indicators.

IV. Problem Definition

Assume mobile devices $\{k\}$ and a server B (BS, Wi-Fi AP, work station, etc.) follow specific protocols to enable communication. There is a communication link between device k and server B that transmits ComData H^k regularly, and H^k contains channel-relevant information. At the same time, SenData $\left\{X^k\right\}$ is collected from distributed devices $\{k\}$ and used to train a large model at the server B for specific sensing tasks. For simplicity, we will denote ComData H^k at device k as H and SenData X^k at device k as X in the remaining part of this section to discuss our problem definition. Even though the SenData X and ComData H serve different purposes, they coexist in the same device and both need to be transmitted from mobile devices to the central server. Meanwhile, the two types of data share spatial consistency, encapsulating information about the same environment. This results in shared data structures and intrinsic correlations, establishing a mutualistic relationship.

Definition of Mutualistic Relationship: X and H are mutualistic if they share a nontrivial amount of mutual information, i.e., $I(X;H) \ge \tau$, where $I(X;H)$ is the mutual information and $\tau > 0$ is a task-specific threshold.

In the following, we will employ the classical rate distortion theory [37] in source coding to demonstrate that the mutualistic relationship between X and H can be exploited in joint compression to improve the overall rate-distortion performance compared to separate compression. Rate-distortion theory is a branch of information theory that deals with the trade-off between the amount of information (rate) and the quality of representation (distortion) in a data compression system.

Assume two types of data X and H are all with high dimensions and from different sources of information. We aim to develop a joint encoder f at the IoT device to compress the source (X,H) jointly and a joint decoder g at the server to represent (X,H) by estimates $(\hat{X},\hat{H})$, as illustrated in Fig. 2. We introduce the distortion function $D_1 = \mathbb{E}\left[d(x,\hat{x})\right]$ and $D_2 = \mathbb{E}\left[d(h,\hat{h})\right]$ to measure the cost of representing the signal (X,H) by $(\hat{X},\hat{H})$ and rate R accounting for the communication overhead of transmission (X,H) jointly. The encoder and decoder are jointly designed to minimize each of the distortion $\{D_1,D_2\}$ while reducing the communication cost R. The performance of the joint encoder and decoder design can be characterized by the joint rate distortion function $R_{X,H}(d_1,d_2)$, which is defined as [37]

$$R_{X,H}(d_1,d_2) = \inf_{\substack{p(\hat{x},\hat{h}|x,h):\mathbb{E}[d(x,\hat{x})]\le d_1,\\ \mathbb{E}\left[d(h,\hat{h})\right]\le d_2}} I(X,H;\hat{X},\hat{H}), \quad (1)$$

where I denotes the mutual information function between two sets of random variables, $p(\hat{x},\hat{h}|x,h)$ denotes the transition probability determined by the encoder f and decoder g. Given the source distribution and a distortion measure, $R_{X,H}(d_1,d_2)$ describes the minimum rate required for joint transmission of (X,H) to achieve particular distortion for each data source. A typical example of distortion function for ComData H is squared-error distortion, which is given by

$$d(x,\hat{x}) = (x-\hat{x})^2. \quad (2)$$

The distortion function for SenData X depends on the specific sensing tasks. For example, in classification task, the distortion function can be the inverse of classification accuracy. Building on the conference version [7], we generalize the joint rate-distortion framework from lossless or partially lossy cases to general lossy compression and establish its upper and lower bounds. Consider two random sources X and H, we have the following proposition [38].

Proposition 1 (Mutualism-Driven Compression Gain)**.** *Let $X \sim p_X(x)$ with rate distortion function $R_X(d_1)$ under distortion limit $\mathbb{E}\left[d(x,\hat{x})\right] \le d_1$. Similarly, let $H \sim p_H(h)$ with rate distortion function $R_H(d_2)$ under distortion limit $\mathbb{E}\left[d(h,\hat{h})\right] \le d_2$. Suppose the joint rate distortion function corresponding to joint compression of (X,H) subject to distortion limits d_1 and d_2 is denoted as $R_{X,H}(d_1,d_2)$. We have the following upper bound and lower bound for the joint rate distortion function:*

$$R_{X,H}(d_1,d_2) \le R_X(d_1) + R_H(d_2), \quad (3)$$

$$R_X(d_1) + R_H(d_2) - I(X;H) \le R_{X,H}(d_1,d_2), \quad (4)$$

with equality achieved if and only if X and H are independent, i.e., $I(X;H) = 0$.

Proof. We firstly prove the upper bound. Let $p(\hat{x}|x)$ be a test channel that satisfies the distortion limit d_1, and $p(\hat{h}|h)$ be a test channel that satisfies the distortion limit d_2. Then define the corresponding joint test channel for (X,H) by $p(\hat{x},\hat{h}|x,h) = p(\hat{x}|x)p(\hat{h}|h)$, which also satisfies the distortion limits of d_1 and d_2 for X and H. Denote the corresponding mutual information as $I(X,H;\hat{X},\hat{H})$. By definition (1), we have

$$R_{X,H}(d_1,d_2) \le I(X,H;\hat{X},\hat{H}). \quad (5)$$

Under this joint test channel, we have

$$\begin{aligned} & I(X,H;\hat{X},\hat{H}) - I(X;\hat{X}) - I(H;\hat{H}) \\ & = \mathbb{E}\left[\log\frac{p(\hat{x},\hat{h}|x,h)}{p(\hat{x},\hat{h})} - \log\frac{p(\hat{x}|x)}{p(\hat{x})} - \log\frac{p(\hat{h}|h)}{p(\hat{h})}\right] \\ & = \mathbb{E}\left[\log\frac{p(\hat{x})p(\hat{h})}{p(\hat{x},\hat{h})}\right] = \int p(\hat{x},\hat{h})d(\hat{x},\hat{h})\left[\log\frac{p(\hat{x})p(\hat{h})}{p(\hat{x},\hat{h})}\right] \\ & \leq \int p(\hat{x},\hat{h})\left[\frac{p(\hat{x})p(\hat{h})}{p(\hat{x},\hat{h})} - 1\right] d(\hat{x},\hat{h}) = 0, \end{aligned} \tag{6}$$

where the inequality follows from $\log(x) \leq x - 1$. From (5) and (6), we have

$$R_{X,H}(d_1,d_2) \leq I(X;\hat{X}) + I(H;\hat{H}), \tag{7}$$

which holds for arbitrary test channels $p(\hat{x}|x)$ and $p(\hat{h}|h)$ satisfying the distortion limits. Taking the infimum of (7) with respect to all test channels yields

$$\begin{aligned} R_{X,H}(d_1,d_2) & \leq \inf_{p(\hat{x}|x):D_1\leq d_1} I(X;\hat{X}) + \inf_{p(\hat{h}|h):D_2\leq d_2} I(H;\hat{H}) \\ & = R_X(d_1) + R_H(d_2), \end{aligned}$$

which is the upper bound. Then we prove the lower bound. Based on $\log(x) \leq x - 1$, we have

$$I(X;\hat{X}|H) + I(H;\hat{H}) - I(X,H;\hat{X},\hat{H}) \leq 0. \tag{8}$$

For any test channel $p(\hat{x},\hat{h}|x,h)$ such that $D_1 \leq d_1$ and $D_2 \leq d_2$, we have

$$\begin{aligned} I(X,H;\hat{X},\hat{H}) & \geq I(X;\hat{X}|H) + I(H;\hat{H}) & (9) \\ & \geq R_{X|H}(d_1) + R_H(d_2), & (10) \end{aligned}$$

where $R_{X|H}(d_1)$ is the conditional rate distortion function of X given H. Taking the minimum of $I(X,H;\hat{X},\hat{H})$ over the appropriate set of test channels $p(\hat{x},\hat{h}|x,h)$ yields

$$R_{X,H}(d_1,d_2) \geq R_{X|H}(d_1) + R_H(d_2). \tag{11}$$

We note that $I(X;\hat{X}|H) + I(X;H) - I(X;\hat{X}) = I(X;H|\hat{X}) \geq 0$. For any test channel $p(\hat{x}|x,h)$ such that $D_1 \leq d_1$, we have that $I(X;\hat{X}|H) \geq I(X;\hat{X}) - I(X;H) \geq R_X(d_1) - I(X;H)$. Taking infimum over appropriate test channels $p(\hat{x}|x,h)$ gives

$$R_{X|H}(d_1) \geq R_X(d_1) - I(X;H). \tag{12}$$

Combining with (11), we get the lower bound. When two sources are independent, i.e., $I(X;H) = 0$, we have $R_{X,H}(d_1,d_2) = R_X(d_1) + R_H(d_2)$. □

The upper bound shows that mutualistic relationships between X and H can reduce the total transmission overhead (the rate) required for joint encoding compared to separate transmissions. This supports the premise that mutualism facilitates rate-efficient compression without degrading reconstruction fidelity. Meanwhile, the lower bound indicates that the rate required for joint transmission is no less than the sum of individual rates minus the mutual information between the two sources. The gap between the upper bound and lower bound is governed by the degree of mutualism, characterized by a threshold parameter τ. A large τ corresponds to stronger mutualistic coupling, thereby enlarging the feasible rate reduction region for joint transmission. To achieve zero additional communication overhead for sensory data collection, i.e., $R_{X,H}(d_1,d_2) = R_H(d_2)$, the lower bound in (4) requires $R_X(d_1) \leq I(X;H)$, or $\tau \geq R_X(d_1)$ for some distortion level d_1. In other words, when the shared information between SenData X and ComData H is sufficient to represent X within the target distortion, the joint transmission of (X,H) can be realized without any additional rate cost beyond that for transmitting H alone.

Practical Interpretation: To provide practical insight into the theoretical results, we define the transmission efficiency for ComData and SenData as the ratio between the achieved information quality and the total transmission rate:

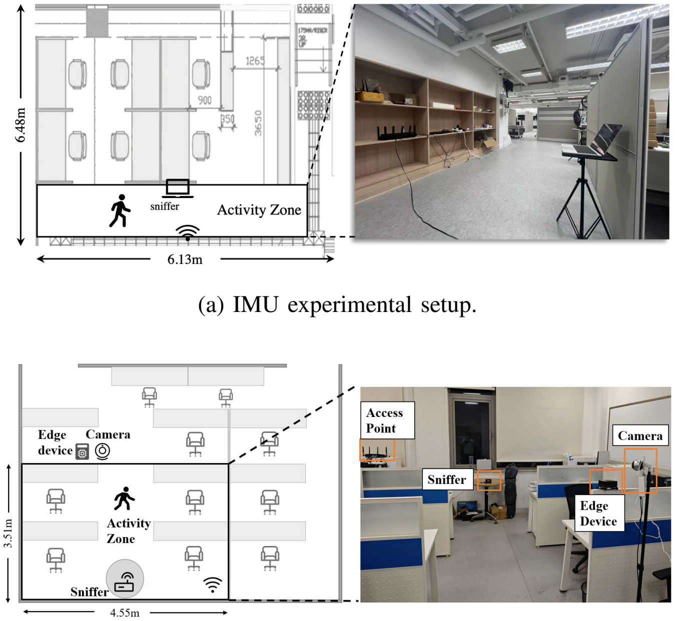
$$\eta_{\text{Sen}} = \frac{Q_X}{R_{\text{total}}}, \quad \eta_{\text{Com}} = \frac{Q_H}{R_{\text{total}}}, \tag{13}$$

where Q_X and Q_H denote the effective information quality (e.g., $1/d_1$ and $1/d_2$) of the SenData X and ComData H, respectively. The total transmission rate R_{total} equals $R_{X,H}(d_1,d_2)$ for joint transmission and $R_X(d_1) + R_H(d_2)$ for separate transmissions. According to (3), the mutualistic relationship ensures that joint transmission always achieves higher efficiency than separate transmission when both attain the same distortion levels. When $\tau > R_X(d_1)$ and $R_X(d_1) = R_H(d_2)$, the proposed MuSAC framework achieves the same distortion with only half the communication rate, resulting in a twofold efficiency improvement. Although this result holds under the strict assumption of identical distortions, in practice, even with slightly higher distortion in joint transmission, the overall efficiency remains superior, as reduced communication overhead often outweighs minor reconstruction losses. Under this relaxed condition, the required mutualism level τ can be significantly lower. This requirement on τ implies that if the communication signal (e.g., CSI in massive MIMO or CBR in WiFi) inherently encodes sufficient environmental information, the receiver can reconstruct the sensory data within a desired distortion level without any additional communication overhead, thereby enabling communication-free sensing.

Summary: Proposition 1 suggests that if two sources are correlated, a well-designed compressor should be able to leverage the inherent correlations to achieve better rate-distortion performance. If two sources are uncorrelated, the optimal rate-distortion performance is the same as that of two sources being compressed independently. In other words, joint compression should not harm the overall rate-distortion. Hence, the mutualistic relationship between SenData and ComData presents a distinct opportunity for seamless integration of mobile crowdsensing into current communication without compromising the effectiveness of either task. In the following section, we will substantiate the above proposition with a proof-of-concept study.

V. Proof-of-concept Study

To validate the feasibility of our insight and substantiate the above proposition, we conducted a series of proof-of-concept



(a) IMU experimental setup.

(b) Human activity recognition experimental setup.

Fig. 3: Experimental setup.

TABLE I: Compression Performance of IMU and CBR Data (NMSE (dB))

	With Correspondence				Without Correspondence			
η	CBR	Combine		IMU	CBR	Combine		IMU
		CBR	IMU			CBR	IMU	
1/10	-8.07	**-9.88**↑	**-5.67**↑	-5.02	-8.07	**-9.12**↑	**-0.68**↑	-0.63
1/20	-5.37	**-6.42**↑	**-4.81**↑	-4.33	-5.37	**-6.05**↑	**-0.62**↑	-0.58

TABLE II: Compression Performance of Image Data for Human Activity Recognition (ACC (%)) and CBR Data (NMSE(dB))

		With Correspondence			
Bandwidth (MHz)	η	CBR	Combine		ACC
			CBR	ACC	
20	1/14	-8.98	**-10.17**↑	**97.96**↑	93.88

studies in typical indoor IoT communication scenarios. We captured the communication data between a mobile phone and Wi-Fi AP while collecting different sensing data in two experiments. These experiments aim to empirically demonstrate that SenData and ComData not only exhibit a meaningful mutualistic relationship, but can also be jointly compressed to improve the overall transmission efficiency without increasing communication payload for communication-free sensing. In the first experiment, the communication data is CBR, and the sensing data comprises raw accelerometer and gyroscope readings from the Inertial Measurement Unit. CBR is a coarse representation of the downlink MIMO channel information sent from the IoT device to the WiFi AP carried in the payload of an action frame, following the standard mechanism in IEEE 802.11 [39]. CBRs are not encrypted and can be captured by a sniffer device. A volunteer performed activities (waving, walking, and falling) in the activity area shown in Fig. 3a. An iPhone 13 equipped with the Phyphox app to record the IMU data was worn on the wrist and kept playing a 1080p video. The AP used was Xiaomi AX6000, and a Macbook Pro installed with Wireshark was used to sniff the wireless channel. The open-source tool Wi-BFI [40] was utilized to obtain the CBR data. The CBR data was around 10 frames per second, so the original IMU data was downsampled from 100Hz to 10Hz. To align the two sets of data, both data types were normalized to [0,1], and the data frames were organized as [1,100,3]. In total, 540 communication samples and 540 activity samples were collected.

We trained a simple autoencoder with 2 convolutional layers with ReLU activation, plus one linear layer to compress the data, and used a symmetric decoder to reconstruct the data from the compressed representation. η is the compression ratio to configure the linear layer. We used the Normalized Mean Squared Error (NMSE) to calculate the data distortion. A smaller NMSE in dB indicates less information loss after compression. Table I (With Correspondence) reports the results of separately compressing communication and sensing data and also the results of jointly compressing them after concatenating the data with input size [2,100,3]. The results show that when data is collected with correspondence, joint compression has enhanced the accuracy of both communication and sensing data, with an average improvement of 21.00% in CBR data and 10.72% in IMU data.

To further substantiate Proposition 1, we sampled the same portion of IMU data from the publicly available Shoaib dataset [41] to replace the IMU data that we collected. The results are shown in Table I (Without Correspondence). When data is collected without correspondence, compressing not explicitly correlated data jointly also helped to reduce distortion and improve the quality of the recovered data. However, the improvement was not as significant as that of explicitly correlated data, with an average improvement of 12.83% in CBR data and 7.42% in IMU data, respectively.

In the second experiment, we deployed edge devices in indoor office environments and collected human activity image sensing datasets for experimentation. Specifically, we used a sniffer to capture the CBR between the edge device and the AP at 20MHz communication band based on the 802.11ac protocol. We chose ASUS RT-AX88U Pro as the AP and NVIDIA Jetson AGX Xavier equipped with Realtek RTL8812BU wireless network interface controller as the edge device and sniffer. The experimental setup is illustrated in Fig. 3b. Simultaneously, we used a spherical camera to capture real-time images of seven common daily indoor activities. There are 35 photos for each action and each picture is uniformly resized to RGB format with input size [48,48,3]. The CBR data were organized as [52,10], which matches those captured by the sniffer in real-world scenarios. Each CBR data is paired with its corresponding image based on synchronized timestamps.

The encoder and decoder were constructed using 3 fully connected layers with ReLU activation functions and Batch normalization, and were jointly trained to minimize the CBR reconstruction error while maximizing the human activity

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3664511

recognition accuracy. The encoder compresses data to size consistent with the CBR data, enabling seamless integration into the existing 802.11ac communication protocol. We use NMSE to evaluate the distortion of the CBR data, and classification accuracy (ACC) to assess the distortion of sensing image data during compression. As shown in Table II, joint compression can reduce communication data loss while improving the accuracy of the classification task, with an improvement of 19.26% in CBR data and 4.08% in sensory data.

Summary: These results substantiate Proposition 1 and validate the feasibility of MuSAC. The proof-of-concept study further provides confirmation of an intrinsic correlation between SenData and ComData, suggesting that compressing them together could enhance overall effectiveness. This justifies the fundamental design principle of MuSAC: embedding sensing into communication without increasing payload size, thereby achieving practical communication-free sensing.

VI. MuSAC Design

In this section, we take into account the distributed nature of mobile crowdsensing and present a practical MuSAC framework. Unlike in Section V, where we keep the communication overhead (the rate) of joint transmission consistent with the individual transmissions, in a practical MuSAC setting, we aim to minimize the distortion in ComData and SenData recovery while ensuring that the communication overhead of joint transmission is the same as that of original ComData's transmission cost (i.e., zero additional communication overhead).

A. Design Consideration

1) SenData is more diverse and complex than ComData, with a higher frame rate and privacy concerns, necessitating pre-compression to preserve its core features and facilitate further co-compression: Although ComData and SenData share mutualistic characteristics and can benefit from co-compression, they are different types of data for different purposes and have inherent differences in their properties. SenData, in particular, is more diverse and complex, with its own distinct features. Another difference is that the frame rate of SenData is typically higher than that of ComData, resulting in a larger volume of data for transmission. Additionally, the direct transmission of raw SenData may raise privacy concerns, making pre-compression necessary before joint compression. Given these considerations, it is important to pre-compress SenData in a way that preserves its core features while also making it easier to compress further. To this end, we design a lightweight on-device universal feature extractor to distill the knowledge from SenData and make it more structured so that it can facilitate further co-compression (Section VI-C).

2) A bi-level optimization-based decoder is developed on the server side to address the heterogeneous distortion in SenData to obtain invariant features: Additionally, unlike the ComData that can be used right after decoding, the SenData is collected for larger model training with the final goal of being capable of solving sensing tasks. The distortion of SenData caused by physical environment, hardware, and user behavior heterogeneous needs to be addressed, otherwise, even large models would not be able to achieve good results [1]. Fortunately, for systems like MuSAC, the server has a clear identification of each device's information, such as user ID. This enables the removal of data heterogeneity caused by different devices and user behaviors by formulating a bi-level optimization problem, ultimately obtaining invariant features (Section VI-D).

B. MuSAC Overview

With the above considerations in mind, Fig. 1 presents an overview of MuSAC, where each mobile device hosts an on-device network, with an encoder to extract universal features and a joint compressor to combine ComData and SenData. The server side hosts a de-mixer first to reconstruct ComData and SenData, and then the reconstructed SenData is fed into the invariant feature extractor module to perform a specific sensing task.

1) On-device Network: According to the first design consideration, MuSAC first pre-compresses the SenData, aiming to preserve the knowledge of the sensing features while making it further compressible. The on-device network $\mathcal{E}_{\boldsymbol{\theta}}$ can be denoted as

$$\mathcal{E}_{\boldsymbol{\theta}} = \Xi \left(\mathcal{CL} \left(\boldsymbol{x} \right), \psi(\boldsymbol{h}) \right). \tag{14}$$

The SenData $\boldsymbol{x}$ is firstly fed into a universal feature encoder (UFE) to distillate representative data knowledge. The UFE at the device k can be represented by the function

$$\mathcal{CL}^k : \boldsymbol{x} \rightarrow \boldsymbol{s}, \tag{15}$$

which is trained locally at the device based on the local sensory data $D^k = (X^k, Y^k)$ and outputs the extracted universal features, denoted as $S^k : \{\boldsymbol{s}\}$. The details of the UFE will be elaborated in the subsequent sections.

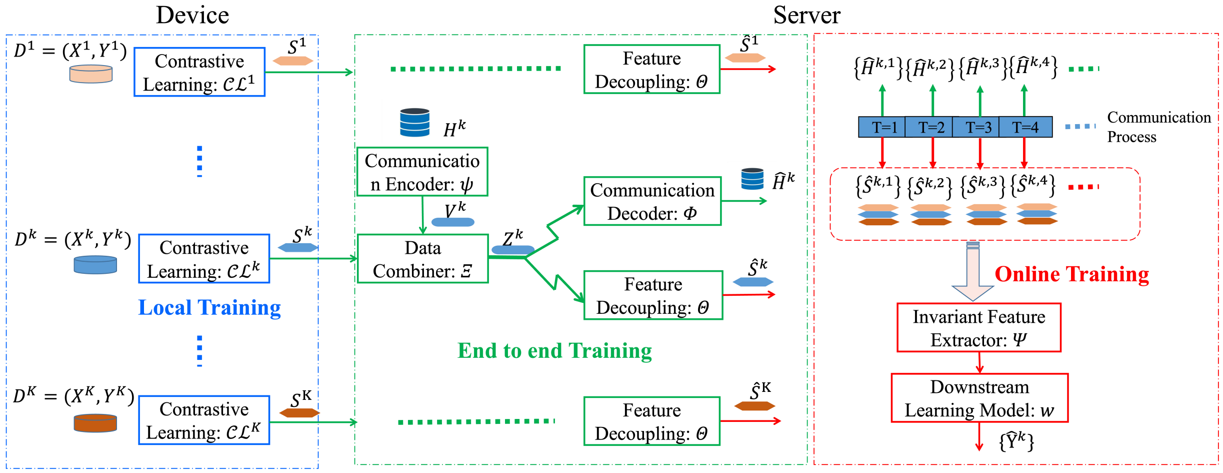
In the meantime, the necessary communication signal $\boldsymbol{h}$ is passed through a predefined encoder $\psi(\cdot)$ to compress the ComData and get $\boldsymbol{v} = \psi(\boldsymbol{h})$. The universal feature $\boldsymbol{s}$ is then being concatenated with the compressed communication signal $\boldsymbol{v}$ through the combiner function Ξ, which is defined as

$$\Xi : (\boldsymbol{s}, \boldsymbol{v}) \rightarrow \boldsymbol{z}.$$

The function of the combiner serves the purpose of leveraging the internal pattern and underlying shared correlation between ComData and SenData to seamlessly integrate SenData feature $\boldsymbol{s}$ into the compressed ComData $\boldsymbol{v}$ without changing its dimension. As a result, the dimension of the composite signal $\boldsymbol{z}$ remains consistent with that of the original ComData feature $\boldsymbol{v}$. The composite signal $\boldsymbol{z}$ is subsequently transmitted to the server through existing communication uplink.

Remark 1. *Regarding to the joint compression process, we would like to make several remarks.*

- *In Section V, we concatenate the original raw data directly as the input for joint compression. In practice, the original ComData undergoes its compression process (i.e.,$\psi(\cdot)$) and the original SenData has its own pre-compression necessity (i.e.,$\mathcal{CL}(\cdot)$). Therefore, in practical*



Device
Server
$D^1 = (X^1, Y^1)$
Contrastive Learning: $\mathcal{CL}^1$
S^1
$D^k = (X^k, Y^k)$
Contrastive Learning: $\mathcal{CL}^k$
S^k
Local Training
$D^K = (X^K, Y^K)$
Contrastive Learning: $\mathcal{CL}^K$
S^K
H^k
Communication Encoder: ψ
V^k
Data Combiner: Ξ
Z^k
Feature Decoupling: Θ
$\hat{S}^1$
Communication Decoder: Φ
$\hat{H}^k$
Feature Decoupling: Θ
$\hat{S}^k$
End to end Training
Feature Decoupling: Θ
$\hat{S}^K$
$\{\hat{H}^{k,1}\}\{\hat{H}^{k,2}\}\{\hat{H}^{k,3}\}\{\hat{H}^{k,4}\}$
T=1
T=2
T=3
T=4
Communication Process
$\{\hat{S}^{k,1}\}\{\hat{S}^{k,2}\}\{\hat{S}^{k,3}\}\{\hat{S}^{k,4}\}$
Online Training
Invariant Feature Extractor: Ψ
Downstream Learning Model: w
$\{\hat{Y}^k\}$

Fig. 4: Overall training of MuSAC framework.

design of MuSAC, the compressed ComData $\boldsymbol{v}$ and the universal feature of SenData $\boldsymbol{s}$ are fed into the combiner Ξ for joint compression.

- *In Section IV, we theoretically demonstrated that the performance of joint compression would surpass that of independent compression. In the practical design of MuSAC, we fix the communication overhead (the rate) the same as the transmission of ComData and evaluate the quality of SenData and ComData respectively. We demonstrate how MuSAC utilizes the mutualism property to transmit SenData and ComData together by measuring the distortion under a given rate i.e., the transmission efficiency, without incurring additional communication overhead. (See Section VII).*

2) Server-side Network: At the server side, the received composite signal $\boldsymbol{z}$ first goes through a de-mixer, given by

$$\Theta : \boldsymbol{z} \rightarrow (\hat{\boldsymbol{s}}, \hat{\boldsymbol{v}}).$$

$\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{v}}$ then fed into $\mathcal{C}_\phi$ and $\mathcal{S}_\varpi$, respectively. Network $\mathcal{C}_\phi$ can be represented as the function $\Phi : \hat{\boldsymbol{v}} \rightarrow \hat{\boldsymbol{h}}$, which reconstructs the ComData $\boldsymbol{h}$. Network $\mathcal{S}_\varpi$ at the server side aims to extract the invariant features from the received signals $Z^k : \{\boldsymbol{z}\}, k \in \mathcal{K}$, such that the optimal classifier w on top of the extracted invariant features matches for all different data distributions. Specifically, the network $\mathcal{S}_\varpi$ consists of three subnetworks, represented by function Θ, Ψ and w, respectively, which can be realized as

$$\mathcal{S}_\varpi = w\left(\Psi\left(\Theta(\boldsymbol{z})\right)\right). \tag{16}$$

Denote the SenData output of Θ as $\hat{S}^k : \{\hat{\boldsymbol{s}}\}$. Based on $\hat{S}^k$, $\Psi(\cdot)$ denotes the invariance feature extraction function, which tries to learn a data representation that elicits an invariant predictor w across different distributions $(\mathcal{X}, \mathcal{Y})^k$ of SenData. w denotes the classifier function, which is guaranteed to be optimal for all data distributions with the invariant feature as the input. The details of the invariant feature learning will be elaborated in the subsequent sections.

C. On Device: Universal Feature Extractor

Pre-compressing SenData to preserve its data knowledge while making it more structured is challenging. Unlike communication data, which has a well-established compression system with encoding and other techniques to ensure that the data can be successfully recovered at the receiving end, SenData requires a different approach due to its varying sensory information. The opportunity lies in the fact that unlike ComData, which needs to be fully recovered at the receiving end, SenData is often used to accomplish specific sensing tasks, and preserving core features related to these tasks is often sufficient. Given that SenData is hosted by mobile devices and labels are accessible, we can leverage these opportunities to design a CL-based lightweight encoder. While using a vanilla encoder with ground truth labels to perform supervised learning can effectively reduce the dimension of the SenData, it cannot ensure that the pre-compressed data becomes more structured and amenable to further compression. Moreover, it may lead the encoder to overfit to the sensing tasks used for its training and filter out some features that are useful for other downstream tasks to be performed on the server. CL is proved to be a powerful method for extracting general features that can be applied to various downstream tasks and has been used for FM training [42], [43]. CL operates by guiding the model to discern between similar and dissimilar pairs of data points. This is achieved by bring positive(similar) pairs closer in the embedding space while pushing negative samples away from the positive samples. Consequently, the model becomes proficient at acquiring significant and semantically rich representations, that can effectively capture important patterns and relationships in the data. These representations can then be used to support a variety of downstream tasks. The core of CL lies in determining the contrastive positive and negative samples. Next, we will provide a detailed explanation of how MuSAC uses CL to train the UFE.

The UFE at device k is represented as $\mathcal{CL}^k$ in (15), which maps the original data from a high dimensional space X^k into a low dimensional feature space S^k. During the training of $\mathcal{CL}^k$, we want the features of two similar datas end up close to each other in the feature space, while the features for two different data to get as far as possible from each other. As the labels of the SenData are known at the device end, we adopt supervised CL [44] in our setting. In supervised CL, positive samples come from the sensory data with the same class label, while negative samples come from the sensory data with different class labels. For example, for a sensory data sample $\boldsymbol{x}_i$ in the dataset X^k, its positive sample is defined as the set $\{\boldsymbol{x}_a\}_{a\in\mathcal{A}_i}$, whose lable is the same as the label of $\boldsymbol{x}_i$, i.e., $\mathcal{A}_i = \{a : y_a = y_i\}$. And its negative sample is defined as the set $\{\boldsymbol{x}_b\}_{b\in\mathcal{B}_i}$, whose label is different from the label of $\boldsymbol{x}_i$, i.e., $\mathcal{B}_i = \{b : y_b \neq y_i\}$. Given the positive and negtive samples, the lightweight network $\mathcal{CL}^k$ is trained to minimize the following contrastive loss:

$$\mathcal{L}_{\mathrm{CL}}^k = \sum_{i=1}^{\left|D^k\right|} \frac{1}{|\mathcal{A}_i|} \left[- \sum_{a\in\mathcal{A}_i} \log \frac{\mathrm{g}\left(\boldsymbol{x}_i, \boldsymbol{x}_a\right)}{\sum_{a\in\mathcal{A}_i}\ \mathrm{g}\left(\boldsymbol{x}_i, \boldsymbol{x}_a\right) + \sum_{b\in\mathcal{B}_i}\ \mathrm{g}\left(\boldsymbol{x}_i, \boldsymbol{x}_b\right)} \right], \tag{17}$$

where $\mathrm{g}(\cdot)$ is used to measure the similarity between two vectors. We defined $\mathrm{g}(\cdot)$ as follows

$$\mathrm{g}(\boldsymbol{x}_i, \boldsymbol{x}_a) = e^{\boldsymbol{x}_i\cdot\boldsymbol{x}_a/\alpha}, \tag{18}$$

where $\alpha \in \mathcal{R}^+$ is a scalar temperature hyperparameter. Unlike common contrastive loss functions, such as InfoNCE loss [45], we use multiple positive samples to extract universal features from the device data. The output data s is now ready for co-compression.

D. On Server: Invariant Feature Extractor

The received signal Z^k at the server can be written as the function of X^k and H^k, that is for any $\boldsymbol{z} \in Z^k$, we have $\boldsymbol{z} = \mathit{\Xi}\left(\mathcal{CL}\left(\boldsymbol{x}\right), \psi(\boldsymbol{h})\right)$, where $\boldsymbol{x} \in X^k$ and $\boldsymbol{h} \in H^k$. Furthermore, the recovered feature $\hat{S}^k$ is a function of Z^k, that is for any $\hat{\boldsymbol{s}} \in \hat{S}^k$, we have $\hat{\boldsymbol{s}} = \mathit{\Theta}(\boldsymbol{z})$, where $\boldsymbol{z} \in Z^k$. Due to the non-IID of X^k, the resulting $\hat{S}^k$ is also non-IID. Assume the sample $\hat{\boldsymbol{s}}$ from device k can be captured by the distribution $\hat{\mathcal{S}}^k$. The goal of the invariant feature extractor is to learn the invariances across different distributions $\{\hat{\mathcal{S}}^k\}_{k=1}^K$. For a classification problem, it means that we want to find a data representation function $\mathit{\Psi}$ from $\{\hat{S}^k\}_{k=1}^K$, such that the optimal classifier, on top of that data representation, is the same for different data distributions. Therefore, during the invariant feature extraction, we have two objectives. Firstly, we want that the data representation can lead to good prediction performance. Secondly, the data representation elicits an invariant predictor across $\{\hat{\mathcal{S}}^k\}_{k=1}^K$. Mathematically, we aim to solve the following bi-level optimization problem:

$$\begin{array}{ll} \min_{\mathit{\Psi},w} & \sum_{k=1}^K R^k(w\cdot\mathit{\Psi}) \\ \text{subject to} & w \in \arg\min_{\bar{w}} R^k(\bar{w}\cdot\mathit{\Psi}), \forall k \in \mathcal{K}, \end{array} \tag{19}$$

where $R^k(w\cdot\mathit{\Psi})$ is the risk for data distribution in device k, given by $R^k(w\cdot\mathit{\Psi}) = \mathbb{E}_{\mathcal{Z}^k}\left[\ell_s\left(w\left(\mathit{\Psi}(\hat{\boldsymbol{s}})\right); \boldsymbol{y}\right)\right]$. ℓ_s denotes the loss function for sensory data-driven learning task. To solve the challenging bi-level optimization problem, we resort to the invariant risk minimization (IRM) framework [46], which translates the original bi-level optimization problem (19) to the following regularized optimization problem:

$$\min_{\mathit{\Psi}} \sum_k \left[R^k(\mathit{\Psi}) + \lambda \cdot \left\|\nabla_{w|w=1.0} R^k(w\cdot\mathit{\Psi})\right\|^2 \right], \tag{20}$$

where $\mathit{\Psi}$ denotes the data representation function and the entire invariant predictor when we assume the scalar classifier $w = 1.0$. The gradient norm penalty measures the optimality of the scalar classifier for each distribution. $\lambda \in [0, \infty)$ is a regularizer balancing between the prediction performance of the representation function $\mathit{\Psi}$ and the invariance of the classifier on top of $\mathit{\Psi}$.

Note that compared to problem (19) , the transformed problem (20) only involves one optimization variable. When optimizing (19) over $(\mathit{\Psi}, w)$, it can be noticed that

$$w\cdot\mathit{\Psi} = \underbrace{\left(w\cdot A^{-1}\right)}_{\tilde{w}} \cdot \underbrace{\left(A\cdot\mathit{\Psi}\right)}_{\tilde{\mathit{\Psi}}}.$$

The re-parameterized invariant predictor allows to design the classifier w to be any non-zero value we choose. Thus the original problem equals to that given the fixed classifier $\tilde{w}$, we search for the data representation $\mathit{\Psi}$, for which all the distribution optimal classifiers are equal to the same classifier $\tilde{w}$. Without loss of generality, we can let $\tilde{w}$ to be a linear classifier with only the first element to be nonzero, i.e., $\tilde{w} = (1, 0, \cdots, 0)$. Then we can obtain more practical optimization problem (20).

To further reduce the impact of data heterogeneity on the final model prediction performance, we introduce a regularization term while extracting the invariant feature, which encourages the features of the same class extracted from different environments to have similar distributions. Specifically, we first group the samples in $(\hat{S}^k, Y^k)$ based on the similarities of labels. For a classification task, we categorize the samples based on the labels. Assume the samples in $(\hat{S}^k, Y^k)$ are clustered into L groups (e.g., there are L classes in the classification task), each group of samples is denoted as $\hat{S}_l^k, l = 1, \cdots, L$. We define the DCOV loss as the distance between the second-order statistics (covariance) of two features $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$:

$$\ell_{DCOV}(\boldsymbol{f}_1, \boldsymbol{f}_2) = \left\|\mathrm{COV}(\boldsymbol{f}_1) - \mathrm{COV}(\boldsymbol{f}_2)\right\|_F^2, \tag{21}$$

where the covariance matrix of $\boldsymbol{f}$ can be calculated as

$$\mathrm{COV}(\boldsymbol{f}) = \frac{1}{n_F - 1}\left(\boldsymbol{F}^T\boldsymbol{F} - \frac{1}{n_F}(\mathbf{1}^T\boldsymbol{F})^T(\mathbf{1}^T\boldsymbol{F})\right), \tag{22}$$

given the data samples of feature $\boldsymbol{f}$ as $\boldsymbol{F}$, the number of data samples as n_F. $\mathbf{1}$ is a column vector with all elements equal to 1. Therefore, based on the group of samples $\hat{S}_l^k$, the covariance of the invariant feature of class l for distribution $\hat{\mathcal{S}}^k$, which is denoted as $\mathrm{COV}(\boldsymbol{f}_l^k)$, can be calculated by replacing $\boldsymbol{F}$ in (22) with $\mathit{\Psi}(\hat{S}_l^k)$, and n_F with the size of group $\hat{S}_l^k$. Finally, we can obtain the following computable loss function for invariant feature extraction:

$$
\begin{aligned}
\bar{\mathcal{L}}_s = & \sum_{k=1}^{K} \left[\ell_s \left(w \left(\Psi(\hat{S}^k) \right); Y^k \right) \right] \\
& + \sum_{k=1}^{K} \sum_{b=1}^{B} \left[\nabla_w \ell_s \left(w \left(\Psi \left(\hat{S}_b^{k,i} \right) \right), Y_b^{k,i} \right) \cdot \right. \\
& \left. \nabla_w \ell_s \left(w \left(\Psi \left(\hat{S}_b^{k,j} \right) \right), Y_b^{k,j} \right) \right] \\
& + \sum_{l=1}^{L} \sum_{k'=k+1}^{K} \sum_{k=1}^{K-1} \left[\ell_{DCOV}(\boldsymbol{f}_l^k, \boldsymbol{f}_l^{k'}) \right],
\end{aligned} \tag{23}
$$

where $(\hat{S}^{k,i}, Y^{k,i})$ and $(\hat{S}^{k,j}, Y^{k,j})$ are two random mini-batches of size B for device k. w is any dummy classifier.

E. MuSAC Training

In the above, we have outlined the proposed modules for MuSAC in practical distributed systems. We will now demonstrate the complete training process of MuSAC, detailing how the modules are developed for the mobile and server sides and how they collaborate to enhance overall performance. This ensures that SenData collection can be achieved through ComData transmission without incurring any additional communication overhead. The training flow is summarized in the Figure 4.

1) Local Training of UFE: Due to the data heterogeneity and the distributed nature of the network, in order to reduce the model downloading overhead and capture the features of local data more effectively, the UFE is trained locally at each device using only local data D^k by minimizing the loss function (17).

2) Warm-up Training of Overall Network: The encoder ψ and decoder Φ for communication are existing autoencoders with fixed weights. Given the per-trained UFEs $\{\mathcal{CL}^k\}_{k=1}^{K}$, encoder ψ and decoder Φ, the combiner Ξ and the feature decoupler Θ are jointly trained in an end-to-end manner by minimizing the following loss function:

$$
\begin{aligned}
\mathcal{L}_e &= \gamma \cdot \mathcal{L}_c + \mathcal{L}_d \\
&= \gamma \cdot \mathbb{E}\left[||\hat{\boldsymbol{v}} - \boldsymbol{v}|| \right] + \mathbb{E}\left[||\hat{\boldsymbol{s}} - \boldsymbol{s}|| \right],
\end{aligned} \tag{24}
$$

where $\mathcal{L}_c$ and $\mathcal{L}_d$ are mean square error of ComData $\boldsymbol{v}$ and SenData $\boldsymbol{s}$, respectively. $\gamma \in (0, \infty)$ balances the de-compression result trade-off between the ComData and SenData. In the evaluation, we set it as 1.2.

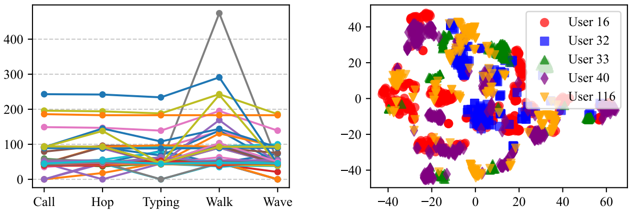
3) Online Training of Invariant Feature Extractor and Downstream Learning Model: After the initial warm-up training, collaborative online training takes place on both the mobile device and server side. On the device side, as SenData is continuously generated, the UFEs are updated locally and produce new distilled SenData $\boldsymbol{s}$. The de-mixer Θ gets the recovered ComData and finishes the communication task. As the communication process proceeds, the server utilizes the collected distilled SenData $\{\hat{S}^k\}_{k=1}^{K}$ from different mobile devices to perform invariant feature extraction and downstream model training. Specifically, as the accumulation of the collected features, the server minimizes the loss function (23) to learn the optimal data representation function Ψ and the invariant predictor w for different data distributions.

VII. Experimetal Simulation

A. Application Scenario and Experimental Setup

MuSAC can be implemented in any systems with established communication links for transmitting mutualistic SenData and ComData to accomplish both communication and sensing tasks. Aside from its use in WiFi communication (Section V), another common application scenario is in the context of massive MIMO technology in cellular systems. Massive MIMO, which forms the foundation of 5G and beyond communication systems, involves the deployment of hundreds or even thousands of antennas in BSs to provide high-throughput connectivity for a large number of IoT devices within a given area. Modern cellular systems operating in frequency-division duplexing (FDD) mode require explicit feedback of downlink CSI from mobile devices to the BS to facilitate beamforming, as the uplink and downlink operate at different frequency bands. Given this, SenData captured by mobile devices, such as IMU data, can be seamlessly transmitted through the uplink of massive MIMO. This SenData can be utilized for localization and tracking, supporting the inference of human activity and enabling mobile health applications. In this section, we will assess the performance of MuSAC in a distributed environment under such scenarios.

The large size of the CSI matrix, which is determined by the number of antennas and sub-carriers, leads to significant overhead in CSI feedback. As a result, it is common practice to compress CSI on the mobile side and then reconstruct it at the BS. Given that the BS can gather CSI information from any point of communication within the area, the current approach involves training a CSI auto-encoder at the BS using a large amount of CSI information and then deploying the encoder to the mobile side [4], [47], [48]. MuSAC follows such a practice. The feedback CSI, which constitutes the communication data, is sourced from the widely-acknowledged COST 2100 cellular channel model, specifically in outdoor rural scenarios operating at the 300MHz communication band. The BS is strategically positioned at the center of a 400 square meter square area, and mobile phones randomly appear within the area. The BS is equipped with 32 uniform linear array (ULA) antennas and has 1024 subcarriers. The purpose of MuSAC is to leverage the existing uplink communication link for both CSI feedback (ComData transmission) and sensing tasks instead of modifying the existing communication system. Therefore, MuSAC directly adopts the existing CSINet [49], an auto-encoder model, for the CSI Feedback part. The CSI samples are preprocessed into $H \in \mathbb{R}^{2\times 32\times 32}$ in the angular-delay domain, totaling 150,000 samples. These samples are divided into 100,000 for training, 30,000 for validation, and 20,000 for testing. We employ compression ratios of 1/4 and 1/32 to compress CSI. The SenData is pre-compressed to match the corresponding sizes after different compression ratios, which is then compressed together with the CSI data for joint transmission with the same dimension as that of compressed CSI. As CSI feedback is fundamental in massive MIMO and is regularly conducted in communication systems, the integrated transmission of SenData and ComData (feedback CSI) will not induce additional communication overhead.



(a) No. of samples for each user. (b) Data distribution of each user.

Fig. 5: Statistics of HARBox Dataset.

TABLE III: Overall Performance of MuSAC.

η	1/32			1/4		
	MuSAC-C	MuSAC	MuSAC-S	MuSAC-C	MuSAC	MuSAC-S
Q.Com	**1.91**	1.77	/	**7.52**	7.14	/
Q.Sen	/	78.31%	**83.41%**	/	81.45%	**85.71%**
Rate	1.23k	1.23k	1.23k	9.84k	9.84k	9.84k
Eff.Com	1.55	**2.88**	/	0.76	**1.44**	
Eff.Sen	/	**1.27**	0.68	/	**0.17**	0.09

SenData is sourced from HARBox [1], a highly diverse dataset collected from 120 different users aged between 17 and 55 years old, using 77 different models of personal smartphones. In contrast to most existing human activity recognition (HAR) IMU data, which are collected in controlled settings where subjects perform prescribed activities in laboratory environments, HARBox is collected from real-world settings. The IMU data is sampled at a rate of 50Hz, using a sliding time window of 2 seconds when users engage in activities. HARBox encompasses five types of daily activities: walking, hopping, phone calls, waving, and typing, comprising a total of 32,935 9-axis IMU samples. It adheres to the real-world non-IID (non-Independently and Identically Distributed) data property found in distributed mobile scenarios, exhibiting variations in data quantities, distribution, and outputs. Figure 5 shows the statistics of the HARBox dataset through data visualization. Figure 5a depicts the quantity of samples per user, categorized by distinct activities within the dataset. The lines represent individual users, highlighting the disparity in data sample distribution—notably, the dataset is *imbalanced* both in terms of activities and users, with certain activities being over-represented for some users and under-represented for others. To visualize the distribution of the dataset, we first randomly select 5 users from the dataset and then utilize t-distributed stochastic neighbor embedding (t-SNE) [50] to reduce the data to 2 dimensions and plot them at Figure 5b. The axes denote the two primary components extracted by the t-SNE, revealing the distribution's heterogeneity. Overlapping data features among users are apparent, yet a clear distinction between different users' data distributions is observable, for example, indicating the *heterogeneity* among different users. We have selected 100 users at random to participate in the training process of MuSAC, and we will test the MuSAC on the remaining data from 20 unseen users. This approach will allow us to assess the overall performance of the model across diverse and unbalanced data, rather than its performance on a specific dataset or environment. By doing so, we can ensure that the model is robust and effective in a real-world distributed setting, highlighting its general applicability.

B. Evaluation Metrics and Configurations

1) Evaluation Metrics:

- Quality of ComData: The ComData needs to be reconstructed to facilitate beamforming, *NMSE* is used to measure the distortion, the quality of ComData is NMSE^{-1}.
- Quality of SenData: The recovered SenData is used to train large models for a specific sensing task, such as activity classification. Therefore, MuSAC evaluates the *ACC* of the sensing task, i.e., the percentage of IMU samples that are accurately classified out of all samples.
- Communication Overhead (Rate): The communication overhead is defined by the volume of transmitted data.
- Data Transmission Efficiency: Since MuSAC transmits both ComData and SenData without additional communication overhead, it's hard to measure the gain. We introduce a metric called data transmission efficiency, which is calculated by quality divided by rate. The data transmission efficiency quantifies the communication cost normalized data recovery quality. A higher value corresponds to better signal recovery quality with the same communication overhead, signifying increased transmission efficiency. Conversely, a smaller value implies the reverse. Correspondingly, the *transmission efficiency* for ComData and SenData are *NMSE*$^{-1}$*/Total Data Volume* and *ACC/Total Data Volume*, respectively. We do not consider the communication overhead caused by retransmissions due to poor network conditions.

2) Configuration: As the dimensions of each ComData is $H \in \mathbb{R}^{2\times32\times32}$, we reorganized each SenData as $X \in \mathbb{R}^{1\times30\times30}$ and then upsampled it to $X \in \mathbb{R}^{1\times32\times32}$ for easy comparison. The encoder and decoder used for CSI Feedback is the existing network, CSINet [49]. The on-device UFE consists of two convolution layers with ReLU activation function. The first layer is a 2D convolution with two [3x3] convolution kernels and the second layer is a 1D convolution layer with a [1x1] convolution kernel to reduce the size of SenData. The parameter size of UFE is 1M and 0.1M respectively when the compression ratio is 1/4 and 1/32. The mobile device is more than enough to run the UFE. The decoder used on the server side is the DCNN [51], a baseline network used for IMU data classification. The joint-compressors and de-compressors are a three-layer ConvNet. We assume that all the compressed values of $\boldsymbol{z}$ for transmission have the same length with $\boldsymbol{v}$, the length of the original compressed ComData. We also evaluate an FL method, FedProx [14], an updated version of the classic FedAvg algorithm that improves generalization performance by adding a regularization term to the local model update to make the local model close to the global model. All the methods are trained using the Adam optimizer with StepLR scheduler and are trained for 1000 epochs. For FedProx, local training epochs are fixed to 100.

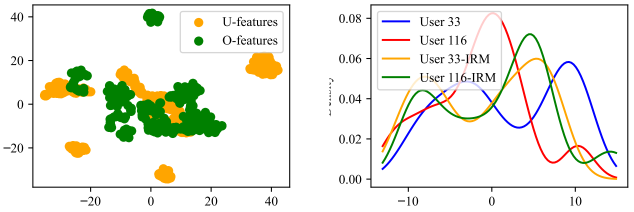
TABLE IV: MuSAC and other learning methods.

	Central	MuSAC	FedProx
Accuracy	84.27%	79.62%	41.07%
Com Overhead	234.24MB	0	2*10*100*0.5MB

C. *Performance Analysis*

1) Overall Performance of MuSAC: To understand the performance of MuSAC. We first run MuSAC and then disable the joint compressor and de-compressor to conduct communication only and sensing only MuSAC. The results are presented in Table III. We noticed that when transmitting an additional type of data under the same communication overhead (rate), it has a negative impact on the performance of both ComData and SenData. Specifically, there is a reduction of 7.33% and 6.11% for a 1/32 compression ratio, and a decrease of 5.05% and 4.97% for a 1/4 compression ratio, for ComData and SenData, respectively. This suggests that if ComData is highly compressed, adding SenData to be transmitted jointly under the existing overhead will affect the performance of ComData. However, when considering the quality of data transfer normalized by communication overhead, i.e., transmission efficiency, a notable improvement is observed. In fact, the transfer efficiency increased by 85.80% and 86.76% for ComData and SenData, respectively, when utilizing a 1/32 compression ratio. This percentage increased further to 89.47% and 88.89% for ComData and SenData, respectively, when using a 1/4 compression ratio. The results from MuSAC demonstrate that through dedicated design, we can seamlessly integrate SenData into existing communication links without incurring additional communication overhead.

2) Comparison between different learning frameworks: Since MuSAC is collecting data from distributed mobile devices and performing learning tasks on the server side, there are other ways to achieve the same goal. One such method is centralized learning, where all the SenData is collected, and then the learning tasks are performed on the server. However, this requires a large communication overhead. Another popular framework is FL, which involves transferring model parameters between the server and the mobile devices. This method has a much lower overhead, but the accuracy may be lower due to data heterogeneity. Different learning frameworks all struggle to balance communication overhead and sensing task accuracy. We compare MuSAC with the other two frameworks. For centralized learning, the network configuration is the same as MuSAC-Sen and we assume that the dataset has been distilled on the mobile device. We calculate the entire distilled training dataset as the communication overhead. The communication overhead of FL is constrained by the number of training rounds and model size. The local training epoch is fixed at 100, so the communication overhead is the multi-run (10*2 runs) model parameter sharing for a single mobile device, in total we have 100 devices for training. Table IV reports the results, the compression ratio for MuSAC is set to 1/16 as MuSAC integrated SenData into ComData transmission, we do not calculate the overhead. As shown in the table, MuSAC achieves good accuracy without any additional communication overhead.



(a) Visualization of original data features and the corresponding extracted universal features.

(b) Probability Density Estimation of User data before and after IRM adoption.

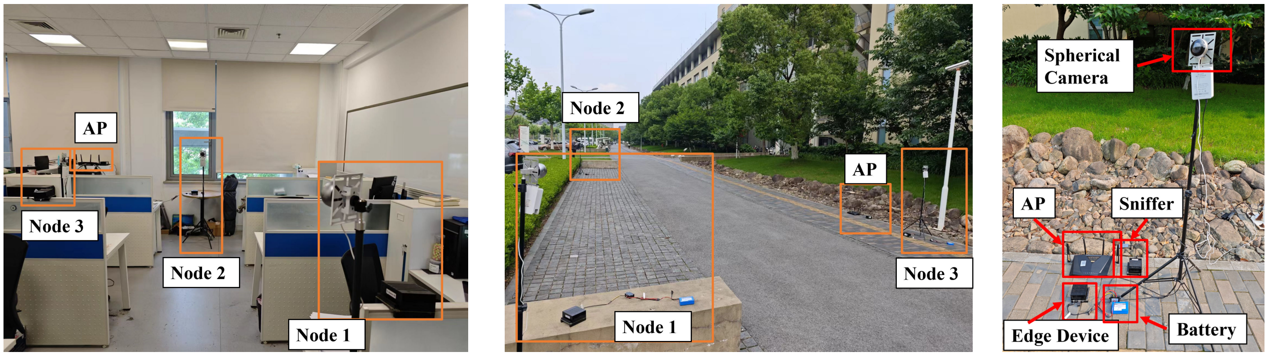
Fig. 6: Evaluation of UFE(a) and IFE(b).

3) Impact of Universal Features Extractor: In order to understand the role of the extractor, we utilize t-SNE to visually represent the two types of features from SenData, universal features (U-features) and original features (O-features), respectively. U-features denote the universal features distilled using CL, and the O-features represent the compressed original features. Figure 6a shows the t-SNE visualization from user 17, which reveals that the U-features are more structured, appearing more separated in feature space compared to the O-features. To validate this observation, we apply the k-means clustering algorithm $(k = 5)$ to both the U-features and the O-features and calculate the clustering accuracy based on the ground truth class label. Since k-means clustering involves randomness, we conducted 100 clustering experiments. On average, the clustering accuracy is 0.10 for the O-features and 0.34 for the extracted U-features. The higher $(\sim 3\times)$ clustering accuracy shows the U-features not only capture the knowledge of the raw data, but also render the data more structured, thereby benefiting subsequent joint compression as stated in proposition.

4) Impact of Invariant Feature Extractor: To verify the effectiveness of the proposed invariant feature extractor based on IRM, during the testing phase, we computed the Jensen-Shannon Divergence (JSD) between pairs of universal features collected from users, as well as the JSD between their corresponding features after passing through the invariant feature extractor. On average, the JSD decreased from 0.025 to 0.017. To visualize, we pick two users and use Kernel Density Estimation (KDE) to estimate the probability density functions of the data. In the testing phase, we plotted the probability density function of the collected universal features of the user data and the probability density function of the features extracted after passing through the invariant feature extractor in Figure 6b. From the figure, it can be observed that the distribution of two users has been significantly brought closer after applying the IRM.

VIII. System Implementation

In this section, we implement a functional prototype of the MuSAC system using off-the-shelf edge devices, a central server, and mobile sensors, where real-world sensory data are seamlessly integrated into WiFi transmissions. This implementation enables practical evaluation of MuSAC's effectiveness



(a) Indoor Deployment Scenario (b) Outdoor Deployment Scenario (c) Hardware Configuration

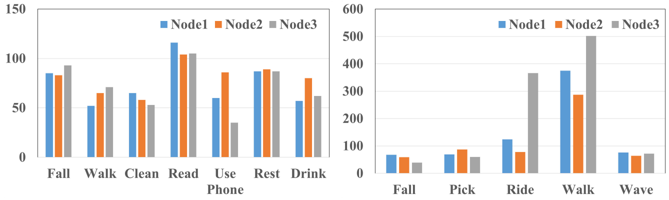
Fig. 7: Experiment Settings.

in supporting communication-free sensory data acquisition in real-world scenarios.

A. *Experimental Methodology*

1) Hardware and Software: As shown in Figure 7, we implement MuSAC system in two environments: an indoor office and an outdoor campus road. In both environments, we deploy three wireless communication nodes, a WiFi packet sniffer and an AP. The three wireless nodes are placed at different locations to capture image data of human activity from various angles in real-world settings. Each node includes a edge device, a Realtek RTL8812BU wireless network interface controller and a spherical camera. The edge device is a mobile computing unit, NVIDIA Jetson AGX Xavier, which is an arch-based device running the Ubuntu 20.04 operating system. It has a shared 16 GB memory (GPU and CPU combined) and is capable of running lightweight models. Its software configuration includes CUDA 11.4, PyTorch 1.13.0, and JetPack 5.1. The spherical camera used in this system is the EZVIZ C6P, which supports panoramic imaging. It is equipped with a 16 GB SD card for local storage of captured human action images along with their corresponding timestamps. The AP is a commerical router, ASUS RT-AX88U Pro, working at 20 MHz comunication band based on the 802.11ac protocol. The central server is connected to AP and it is equipped with an Intel Xeon Gold 6354 CPU and 512GB RAM. Its software configuration includes CUDA 12.6 and PyTorch 2.6.0.

2) Experimet Settings: A small office that accommodates approximately ten people is selected as the indoor scenario. Three nodes are positioned to capture human action images within a rectangular area measuring approximately $4m \times 3m$. Diverse daily behaviors and human movements are captured by each node in the indoor office setting. For the outdoor scenario, we consider the human activity recognition task along a campus road. Three camera nodes are deployed to capture random human activities within a rectangular area of $15m \times 5m$. These nodes are placed near power-supplied roadside infrastructure such as streetlights and surveillance cameras, ensuring stable data transmissions. During pre-training, the edge device connects to the AP via the wireless network interface controller. The sniffer captures CBR data during communication, while the spherical camera records action images. During operation, the edge device inputs the sensing image and CBR data into the encoder for joint compression, embeds the result into an Action No Ack packet, and transmits it to the AP. To address the problem of potential transmission errors, we adopt a queueing mechanism at the edge device, allowing edge devices to temporarily buffer and retransmit packets when network congestion is detected, rather than discarding them directly.



(a) Indoor office room (b) Outdoor campus road

Fig. 8: Dataset distribution across nodes.

3) Dataset: We use a spherical camera to capture a human activity photo every 12 frames in real-world settings, and crop the photo into a [48,48,3] RGB image, referred to as SenData. The sniffer operates on channel 4 with a 20 MHz bandwidth and a center frequency of 2427 MHz under the 802.11ac protocol, collecting all Action No Ack packets addressed to the AP and extracting the CBR information as ComData. In the indoor office scenario, we collected seven types of daily behaviors, including typical body movements such as walking and falling, and activities like drinking, reading, using phone, cleaning, and resting. We recorded about 50 minutes of action video, with a total of 1,593 sets of communication and sensing data. The sample distribution across nodes in the indoor office is illustrated in Figure 8a. In the outdoor campus road scenario, we recorded approximately 90 minutes of action video, yielding 2,326 data sets. We collected five types of common behavioral actions, including walking, falling, riding, picking up, and waving. All video data used in the indoor and outdoor scenario were collected with verbal consent of participants and were used solely for research purposes. Facial features and other identifiable attributes were not used for

TABLE V: Performance of MuSAC in Real-World Environments.

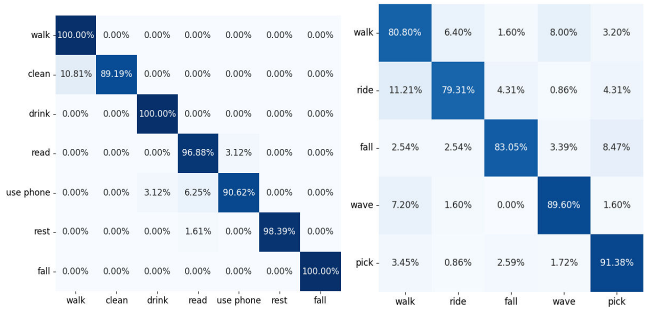
Scenario	Office Room						Campus Road					
η	1/20			1/32			1/20			1/32		
	MuSAC-C	MuSAC	MuSAC-S	MuSAC-C	MuSAC	MuSAC-S	MuSAC-C	MuSAC	MuSAC-S	MuSAC-C	MuSAC	MuSAC-S
Q.Com	56.16	**59.07**	/	**56.02**	54.73	/	**24.23**	23.57	/	**23.36**	22.95	/
Q.Sen	/	**97.50%**	97.19%	/	96.56%	**96.88%**	/	80.33%	**87.00%**	/	80.17%	**86.00%**
Rate	0.37k	0.37k	0.37k	0.23k	0.23k	0.23k	0.37k	0.37k	0.37k	0.23k	0.23k	0.23k
Eff.Com	151.78	**319.29**	/	243.56	**475.91**	/	65.48	**127.40**	/	101.56	**199.56**	/
Eff.Sen	/	**5.27**	2.62	/	**8.39**	4.21	/	**4.34**	2.35	/	**6.97**	3.73

identification and were anonymized before processing.[1] The sample distribution across nodes in the outdoor campus road is shown in Figure 8b. The ratio of the training set to the test set is 4:1.

4) Configuration: Our encoder consists of a three-layer fully connected neural network with ReLU activation functions. The numbers of neurons in the three layers are 4096, 1024, and 520, respectively. The number of neurons in the final layer of the encoder dynamically adjusts according to the compression ratio. By default, it is set to be equal to the CBR dimension. To stabilize training and prevent vanishing gradients, a BatchNorm layer is added to each network layer. We use Adam as the optimizer during network training and set the learning rate parameter to 1e-4. To prevent overfitting during training, we introduce a Dropout layer and set the dropout rate parameter to 0.2. Additionally, the L2 regularization parameter in the loss function is set to 1e-6. During training, image and CBR data are concatenated into a unified vector and fed into the encoder. The encoder outputs a jointly compressed vector, which serves as input to the decoder network deployed at the central server. This output is a one-dimensional vector of 520 elements, matching the CBR shape of [52,10]. According to the image size of [48,48,3], we can conclude that the compression ratio of the MuSAC system is about 1/14 under 20MHz bandwidth. The decoder network also uses a multi-layer fully connected structure, with two output branches: ImageDecoder and CBRDecoder. The CBRDecoder adopts a two-layer fully connected network architecture, similar to the encoder, with each layer also using ReLU activation functions. The numbers of neurons in its two layers are 1024 and 520, respectively. The ImageDecoder employs a single fully connected layer, and the number of output neurons corresponds to the number of activity classes in the classification task. CBRDecoder reconstructs the CBR data, and its loss function L_{cbr} is evaluated using NMSE. ImageDecoder handles image classification tasks, and its loss function L_{img} is the cross-entropy loss. The final loss function L_{total} of the proposed network is defined as

$$L_{total} = \gamma \cdot L_{cbr} + L_{img} \tag{25}$$

where γ is a hyperparameter that controls the balance between two loss terms. Based on experimental results, we set γ to 15 in indoor scenario and 10 in outdoor scenario. Due to the powerful computational capacity of the server, the batch size B during network training is set to the total size of the training dataset. The number of edge devices K is set to 3. The parameter L, representing the number of action classes, is set to 7 in indoor scenarios and 5 in outdoor scenarios. The coefficient λ is set to 0.8, serving as a regularization term that balances the prediction performance of the representation function ψ and the invariance of the classifier built upon ψ.



(a) Indoor office room

(b) Outdoor campus road

Fig. 9: The confusion matrix of utilizing MuSAC for human acticity recognition tasks.

B. Evaluation of MuSAC in Real World Scenarios

1) Overall Performance of MuSAC: In the indoor office scenario, the limited space and number of participants result in activity images that are less affected by lighting and moving object. Similarly, the CBR data is also less influenced by external factors, leading to more stable signals and higher classification accuracy. As shown in Figure 9a, the MuSAC system achieves an overall human activity recognition accuracy of 97.19% in the indoor environment. In the outdoor scenario, the sensing dataset exhibits high heterogeneity. First, the data corresponding to a given class encompasses high variations, as the data contains a diverse population, including various ages, genders and body-type individuals. Second, the number of samples for each activity class is highly imbalanced due to the randomness of capturing images of passersby. Third, external factors such as weather and lighting conditions significantly affect image quality, resulting in varied data quality across samples. The quality of CBR is also susceptible to interference from outdoor environments, resulting in reduced signal reliability. As illustrated in Figure 9b, the MuSAC system achieves an overall accuracy of 85.50% for outdoor human activity recognition, with classification accuracies for riding and walking approximately 5% below average. This is because these two activities are more susceptible to variations among individuals. Nevertheless, the inference accuracy for both activities remains close to 80%, demonstrating strong generalization of the model in real-world settings and its potential for broader applications.

[1] All experiments involving human subjects have been approved by our IRB.

TABLE VI: MuSAC and other learning methods in real-world environments.

	Central	MuSAC	FedAvg
Accuracy	97.50%	97.19%	95.63%
Com Overhead	69.7MB	0	2*20*3*144MB

(a) Office room.

	Central	MuSAC	FedAvg
Accuracy	85.83%	85.50%	82.50%
Com Overhead	365.3MB	0	2*20*3*144MB

(b) Campus road.

The overall performance of MuSAC in both indoor and outdoor scenarios is summarized in Table V. It shows that MuSAC significantly improves the data transmission effeciency for both SenData and ComData across various compression ratios and experimental settings. Specifically, in the indoor scenario, MuSAC improves transmission efficiency by 110.36% and 101.14% for ComData and SenData at a compression ratio of 1/20, and by 95.39% and 99.28% at 1/32 compression. In the outdoor scenario, although overall performance is lower than indoors due to dataset variability, MuSAC's joint compression still yields substantial benefits. At a 1/20 compression ratio, transmission efficiency increases by 94.56% and 84.68% for ComData and SenData, respectively. At a 1/32 compression ratio, the improvements are 96.49% for ComData and 86.86% for SenData. While a slight compromise in transmission quality is observed in most cases due to communication-free embedding, MuSAC achieves both improved transmission quality and zero additional communication overhead in certain scenarios. For instance, at a 1/20 compression ratio in the Office Room, MuSAC achieves slight improvements in both Q.Com and Q.Sen. The training of neural network takes 15 minutes on average. The model size is 138.38 MB. We also record the inference time of MuSAC on edge devices. As shown in Table VII, when deployed and executed on edge devices, MuSAC achieves an average per-sample inference time of 1.4139 ms for different scenarios. Compared with the separate transmission schemes MuSAC-C and MuSAC-S, the inference time of MuSAC increases due to the simultaneous encoding of both ComData and SenData. Nevertheless, when considering the overall processing time of transmitting both ComData and SenData, MuSAC reduces the overall time overhead by 26.61% relative to separate transmission. The low inference latency further indicates that MuSAC is well suited for real-time edge intelligence applications. No data transmission errors are observed in our system evaluation.

2) Comparision with Other Learning Method: In practical settings, we also compare MuSAC with central learning and FL method. In central learning, all nodes transmit image datasets to AP, which then forwards the data to the server for processing. The communication overhead equals the dataset size: 69.7 MB for indoor and 365.3 MB for outdoor data. In FL, we perform data enhancement operations to align the data distribution across three nodes. Then, FedAvg algorithm is utilized to train the model for 2000 epochs, with model parameters shared with the central node every 100 epochs. The model size is approximately 144 MB. For fair comparison, all three methods are evaluated using a compression ratio of 1/14. Both central learning and FL use the same network architecture as MuSAC-Sen, which corresponds to the network components for the sensing task. According to Table VI, compared with the other two methods, in real-world environments, MuSAC achieves accuracy slightly lower than central learning but higher than FL. More importantly, MuSAC incurs zero additional communication overhead. Although FL avoids direct data transmission, it still incurs considerable communication overhead due to 20 rounds of model parameter exchange and aggregation between clients and the server.

TABLE VII: Inference time (ms) on edge devices

Scenario	MuSAC-C	MuSAC-S	MuSAC
Indoor	0.7081	1.2333	1.4239
Outdoor	0.7005	1.2112	1.4040

IX. Conclusion and Discussion

This paper introduces a new paradigm of MuSAC, which leverages the mutualistic relationship between communication data and sensing data to efficiently crowdsource massive heterogeneous sensory data using existing communication links without incurring additional communication overhead. Importantly, MuSAC's architecture is modality-agnostic and generalizable to a variety of sensing tasks. Our evaluations span IMU-based activity classification, image-based behavior recognition, and real-world mobile crowdsensing, demonstrating consistent efficiency gains under fixed communication budgets. These results suggest that MuSAC can serve as a foundational design for scalable, efficient, and privacy-preserving sensory data acquisition in diverse IoT deployments. We believe that MuSAC sheds light on the future protocol design and will play a vital role in shaping future sensing and communication.

However, the current MuSAC framework has certain limitations, our discussions are summarized as follows:

- MuSAC inherently enhances privacy by extracting compact sensing features from raw data and embedding only these features into the communication data. Since much of the original semantic content is removed during this process, an adversary intercepting the MuSAC data faces substantial difficulty reconstructing the raw sensory information, thereby improving privacy protection. However, enhancing privacy involves an inherent trade-off, as obfuscating sensing features to improve privacy inevitably degrades sensing performance. Future work will focus on jointly optimizing the encoder and decoder to develop a MuSAC system with theoretically guaranteed privacy preservation, targeting an optimal balance among privacy protection, communication overhead, and data transmission quality.
- In MuSAC, we assume a perfect communication channel where the server receives exactly what the mobile device transmits. In the future, we can make further improvements by considering the data transmission errors

between the transmitter and receiver, which can be accomplished by incorporating the wireless channel fading and multi-path effect into the training process of the encoding and decoding modules.

- When MuSAC is deployed in real-world sensing tasks, the pre-training of encoders and decoders depends on a large amount of labeled data. One possible solution is to introduce large models (such as CLIP) into edge devices and use zero-shot or few-shot learning methods to automatically realize sensory data distillation and joint data compression.
- Interference between multiple users is currently not considered, which, however, will affect the communication system and thereby impact the ComData. Transmitting SenData and ComData together may further amplify the impact. Our future work will introduce advanced multiple access control (MAC) techniques into the MuSAC system to address this issue.
- MuSAC currently transmits ComData and SenData simultaneously. In practical deployment, we find that due to the occurrence patterns of ComData and SenData differ significantly. While Sendata is continuously generated in large volumes, ComData is transmitted only occasionally, depending on specific communication demands. To address this mismatch, a data-aware MuSAC system can be developed, where the transmission of sensory data is optimally scheduled based on resource constraints and the priority of different SenData, such that the transmission of SenData and ComData can be synchronized.
- Through our experiments, we find that in practical scenarios, the correlation between communication data and sensory data will be weakened by external environmental interference, which is mentioned in Section VIII. In the future, we can expand the definition of MuSAC to include the joint compression of more general sensing-related data and application-layer communication data. For instance, model parameters in FL can be regarded as sensing-related data, as they are derived from training on local sensory data. On the other hand, user-generated content such as voice, text, or videos represent the communication data intended for transmission to a destination. By enabling the joint embedding and transmission of these data types over existing communication links, MuSAC can be envisioned as an effective solution for a broader range of emerging applications, such as real-time FL, intelligent IoT, etc.

References

[1] X. Ouyang, Z. Xie, J. Zhou *et al.*, “Clusterfl: A similarity-aware federated learning system for human activity recognition,” in *MobiSys - Proc. Annu. Int. Conf. Mob. Syst., Appl., Serv.*, 2021, pp. 54–66.

[2] C. Li, X. Zeng, M. Zhang *et al.*, “Pyramidfl: A fine-grained client selection framework for efficient federated learning,” in *Proc Annu Int Conf Mobile Comput Networking*, 2022, pp. 158–171.

[3] A. Li, J. Sun, P. Li *et al.*, “Hermes: An efficient federated learning framework for heterogeneous mobile clients,” in *Proc Annu Int Conf Mobile Comput Networking*, 2021, pp. 420–437.

[4] J. Guo, C.-K. Wen, S. Jin *et al.*, “Overview of deep learning-based CSI feedback in massive MIMO systems,” *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8017–8045, 2022.

[5] S. Ji and M. Li, “Clnet: Complex input lightweight neural network designed for massive MIMO CSI feedback,” *IEEE Wirel. Commun. Lett.*, vol. 10, no. 10, pp. 2318–2322, 2021.

[6] C. Wu, X. Huang, J. Huang *et al.*, “Enabling Ubiquitous WiFi Sensing with Beamforming Reports,” in *SIGCOMM - Proc. ACM SIGCOMM Conf.*, 2023, pp. 20–32.

[7] S. Ji, L. Lian, Y. Zheng *et al.*, “MuSAC: Mutualistic sensing and communication for mobile crowdsensing,” in *Proc Int Conf Distrib Comput Syst*, 2024, pp. 243–254.

[8] A. Capponi, C. Fiandrino, B. Kantarci *et al.*, “A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities,” *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2419–2465, 2019.

[9] Y. Wu, Y. Wang, W. Hu *et al.*, “Resource-aware photo crowdsourcing through disruption tolerant networks,” in *Proc. Int. Conf. Devices, Circuits Syst., ICDCS*, 2016, pp. 374–383.

[10] L. Xu, X. Hao, N. D. Lane *et al.*, “Cost-aware compressive sensing for networked sensing systems,” in *Proc. - ACM/IEEE Int. Conf. Inf. Process. Sens. Networks, IPSN*, 2015, pp. 130–141.

[11] J. Liu, H. Shen, H. S. Narman *et al.*, “A survey of mobile crowdsensing techniques: A critical component for the internet of things,” *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 3, pp. 1–26, 2018.

[12] J. Wang, Z. Charles, Z. Xu *et al.*, “A field guide to federated optimization,” *arXiv:2107.06917*, 2021.

[13] S. Alam, T. Zhang, T. Feng *et al.*, “Fedaiot: A federated learning benchmark for artificial intelligence of things,” *arXiv:2310.00109*, 2023.

[14] T. Li, A. K. Sahu, M. Zaheer *et al.*, “Federated optimization in heterogeneous networks,” *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.

[15] J. Konečný, H. B. McMahan *et al.*, “Federated learning: Strategies for improving communication efficiency,” *arXiv:1610.05492*, 2017.

[16] M. Jaggi, V. Smith, M. Takáč *et al.*, “Communication-efficient distributed dual coordinate ascent,” *Adv. neural inf. proces. syst.*, vol. 2, pp. 3068–3076, 2014.

[17] P. Kairouz, H. B. McMahan *et al.*, “Advances and open problems in federated learning,” *Found. Trends Mach. Learn.*, vol. 14, pp. 1–210, 2019.

[18] H. Wang, M. Yurochkin, Y. Sun *et al.*, “Federated learning with matched averaging,” in *Int. Conf. Learn. Represent., ICLR*, 2019.

[19] Y. Mansour, M. Mohri, J. Ro *et al.*, “Three approaches for personalization with applications to federated learning,” *arXiv:2002.10619*, 2020.

[20] L. Zhang, L. Shen, L. Ding *et al.*, “Fine-tuning global model via data-free knowledge distillation for non-iid federated learning,” in *Proc IEEE Comput Soc Conf Comput Vision Pattern Recognit*, 2022, pp. 10 174–10 183.

[21] D. A. E. Acar, Y. Zhao, R. Matas *et al.*, “Federated learning based on dynamic regularization,” in *Int. Conf. Learn. Represent., ICLR*, 2021.

[22] S. Chen and B. Li, “Towards optimal multi-modal federated learning on non-iid data with hierarchical gradient blending,” in *Proc IEEE INFOCOM*, 2022, pp. 1469–1478.

[23] J. Zhao, L. Qian, and W. Yu, “Human-centric resource allocation in the metaverse over wireless communications,” *IEEE J. Sel. Areas Commun.*, vol. 42, no. 3, pp. 514–537, 2024.

[24] H. Xie, Z. Qin, G. Y. Li *et al.*, “Deep learning enabled semantic communication systems,” *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.

[25] R. Cheng, N. Wu, V. Le *et al.*, “MagicStream: Bandwidth-conserving immersive telepresence via semantic communication,” in *SenSys - Proc. ACM Conf. Embed. Networked Sens. Syst.*, 2024, pp. 365–379.

[26] Z. Weng and Z. Qin, “Semantic communication systems for speech transmission,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, 2021.

[27] J. A. Zhang, M. L. Rahman, K. Wu *et al.*, “Enabling joint communication and radar sensing in mobile networks—a survey,” *IEEE Commun. Surv. Tutor.*, vol. 24, no. 1, pp. 306–345, 2022.

[28] R. Uijlenhoet, A. Overeem, and H. Leijnse, “Opportunistic remote sensing of rainfall using microwave links from cellular communication networks,” *Wiley Interdiscip. Rev.-Water*, vol. 5, 2018.

[29] J. Ostrometzky and H. Messer, “Opportunistic weather sensing by smart city wireless communication networks,” *Sensors*, vol. 24, no. 24, 2024.

[30] C. Han, G. Zhang, B. Ji *et al.*, “On the potential of using emerging microwave links for city rainfall monitoring,” *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 174–180, 2023.

[31] W. Li, M. J. Bocus, C. Tang *et al.*, “On CSI and passive Wi-Fi radar for opportunistic physical activity recognition,” *IEEE Trans. Wirel. Commun.*, vol. 21, no. 1, pp. 607–620, 2022.

[32] W. Li, B. Tan, and R. Piechocki, "Passive radar for opportunistic monitoring in E-Health applications," *IEEE J. Transl. Eng. Health Med.-JTEHM*, vol. 6, pp. 1–10, 2018.

[33] B. Yang, L. He, N. Ling *et al.*, "EdgeFM: Leveraging foundation model for open-set learning on the edge," in *SenSys - Proc. ACM Conf. Embed. Networked Sensors Syst.*, 2024, pp. 111–124.

[34] S. Shi, N. Ling, Z. Jiang *et al.*, "Soar: Design and deployment of a smart roadside infrastructure system for autonomous driving," in *ACM MobiCom - Proc. Int. Conf. Mob. Comput. Netw.*, 2024, pp. 139–154.

[35] T. Zheng, A. Li, Z. Chen *et al.*, "AutoFed: Heterogeneity-aware federated multimodal learning for robust autonomous driving," in *Proc Annu Int Conf Mobile Comput Networking*, no. 15, 2023, pp. 1–15.

[36] X. Ouyang, X. Shuai, Y. Li *et al.*, "ADMarker: A multi-modal federated learning system for monitoring digital biomarkers of Alzheimer's disease," in *ACM MobiCom - Proc. Int. Conf. Mob. Comput. Netw.*, 2024, pp. 404–419.

[37] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[38] A. Wyner and J. Ziv, "Bounds on the rate-distortion function for stationary sources with memory," *IEEE Trans. Inf. Theory*, vol. 17, no. 5, pp. 508–513, 1971.

[39] "IEEE Std 802.11™-2016, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control," 2016.

[40] K. F. Haque, F. Meneghello, and F. Restuccia, "Wi-BFI: Extracting the IEEE 802.11 beamforming feedback information from commercial Wi-Fi devices," in *ACM WiNTECH - Proc. ACM Workshop Wirel. Netw. Testbeds, Exp. Eval. Charact. Part MobiCom*, 2023, pp. 104–111.

[41] M. Shoaib, S. Bosch, O. D. Incel *et al.*, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014.

[42] A. Radford, J. W. Kim, C. Hallacy *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," *arXiv:2103.00020*, 2021.

[43] X. Liu, F. Zhang, Z. Hou *et al.*, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, 2021.

[44] P. Khosla, P. Teterwak, C. Wang *et al.*, "Supervised contrastive learning," *Adv. neural inf. proces. syst.*, vol. 33, pp. 18 661–18 673, 2020.

[45] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748*, 2018.

[46] M. Arjovsky, L. Bottou, I. Gulrajani *et al.*, "Invariant risk minimization," *arXiv:1907.02893*, 2019.

[47] H. C. Qingyong Hu, Hua Kang *et al.*, "CSI-StripeFormer: Exploiting stripe features for CSI compression in massive MIMO system," in *Proc IEEE INFOCOM*, 2023, pp. 1–10.

[48] S. Ji and M. Li, "Enhancing deep learning performance of massive MIMO CSI feedback," in *IEEE Int Conf Commun*, 2023, pp. 4949–4954.

[49] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 5, pp. 748–751, 2018.

[50] L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[51] J. Yang, M. N. Nguyen, P. P. San *et al.*, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *IJCAI Int. Joint Conf. Artif. Intell.*, 2015, pp. 3995–4001.