NEURAL RADIO-FREQUENCY RADIANCE FIELDS: DEVELOPMENT AND APPLICATIONS

XIAOPENG ZHAO

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Computing

Neural Radio-Frequency Radiance fields: Development and Applications

Xiaopeng Zhao

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
August 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Xiaopeng Zhao

Abstract

Recently, the growth of the global Internet of Things (IoT) market has driven the development of numerous wireless systems. To meet the demands of various applications, these systems operate across diverse frequencies and employ different modulation schemes. A standardized wireless channel model is commonly used to describe radio-frequency (RF) signal propagation of these systems. Accurate channel prediction is crucial for optimizing the performance of communication and sensing technologies. However, complex interactions such as reflection and diffraction between RF signals and environmental entities pose challenges. Conventional methods of wireless channel prediction struggle with the complex nature of real-world environments.

To address these limitations, this thesis first introduces the design of Neural Radio-Frequency Radiance Fields (NeRF^2), a model that provides precise predictions of wireless channels at any location in the environment. NeRF^2 begins by measuring a sparse set of signals from the scene of interest and then employs these measurements to train a neural radiance field. This field represents the scene as a continuous volumetric function, detailing the electromagnetic properties of each voxel. Once trained, NeRF^2 can predict the wireless channel at previously unmeasured locations by tracing new paths through the voxel representation. As a model at the physical layer, NeRF^2 has extensive applications, including enhanced channel prediction in Frequency Division Duplex (FDD) systems and improved accuracy in wireless localization.

Moreover, random phase noise from sources like oscillator jitters in real-world wireless

systems impacts channel estimations, especially in low-power backscatter systems. To address this issue, we introduce consistent phase estimation protocols that effectively overcome the challenges of achieving accurate phase estimation across long distances. These protocols resolve the π-ambiguity commonly encountered in commercial RFID readers. Additionally, we further refined these protocols to eliminate flicker noise and neutral white noise, and to correct spatial and temporal imbalances, thereby enhancing the robustness of the system.

Owing to its precise channel prediction, NeRF^2 can generate synthetic training datasets that improve deep-learning-based indoor localization algorithms. However, conventional methods still face challenges from their reliance on high-quality training data and limited adaptability in diverse environments. To overcome these limitations, we introduce the Transformer-based Localization (TBL) model, improving the localization accuracy by its historical position understanding. We also propose a semi-supervised training approach to reduce the need for extensive label collection. To further enhance transferability, we developed LocGPT, a pre-trained version of the TBL model, using 1.3 million data samples. In new environments, this model requires only a minimal dataset for fine-tuning.

Another application of NeRF^2 is in environmental sensing. We introduce Satellitic Radiance Fields (SaRF), an approach that uses crowdsourced smartphone GPS data to construct accurate 3D urban maps. This methodology capitalizes on the observation that materials with higher densities, such as concrete and metal, typically cause greater attenuation of RF signals. By training a neural radiance field, SaRF accurately learns the attenuation properties of each voxel with respect to satellite signals, enabling the detailed reconstruction of 3D voxel maps.

In conclusion, this thesis sheds light on the open challenges in modeling the propagation of RF signals. To address the challenges, we introduce the NeRF^2 and demonstrate its effectiveness in applications such as wireless indoor localization and 3D urban voxel map reconstruction. Looking ahead, we discuss potential future work,

including adapting the neural channel model to dynamic environments. Additionally, we explore transferring methodologies from optical to electromagnetic neural radiance fields, potentially enhancing the implementation of wireless digital twin networks.

Publications Arising from the Thesis

Conference Proceedings

1. Xiaopeng Zhao, Shen Wang, Zhenlin An, Lei Yang, "Crowdsourced Geospatial Intelligence: Constructing 3D Urban Maps with Satellitic Radiance Fields," in *Proc. of ACM IMWUT/UbiComp*, 2024.

2. Xiaopeng Zhao, Guosheng Wang, Zhenlin An, Qingrui Pan, Lei Yang, "Understanding Localization by a Tailored GPT," in *Proc. of ACM MobiSys*, 2024.

3. Xiaopeng Zhao, Zhenlin An, Qingrui Pan, Lei Yang, "NeRF2: Neural Radio-Frequency Radiance Fields," in *Proc. of ACM MobiCom*, 2023. (Best Paper Award Runner-up)

4. Shen Wang, Xiaopeng Zhao, Donghui Dai, Lei Yang, "Mirror Never Lies: Unveiling Reflective Privacy Risks in Glass-laden Short Videos," in *Proc. of ACM MobiCom*, 2024.

5. Qingrui Pan, Zhenlin An, Xiaopeng Zhao, Lei Yang, "Revisiting Backscatter Frequency Drifts for Fingerprinting RFIDs: A Perspective of Frequency Resolution," in *Proc. of IEEE SECON*, 2023. (Best Paper Award)

6. Sicong Liao, Zhenlin An, Qingrui Pan, Xiaopeng Zhao, Jingyu Tong, Lei Yang, "XiTuXi: Sealing the Gaps in Cross-Technology Communication by Neural Machine Transition," in *Proc. of ACM SenSys*, 2023.

7. Xueyuan Yang, Zhenlin An, Xiaopeng Zhao, Lei Yang, "Transfer Beamforming via Beamforming for Transfer," in *Proc. of IEEE INFOCOM*, 2023.

8. Qingrui Pan, Zhenlin An, Xueyuan Yang, Xiaopeng Zhao, Lei Yang, "RF-DNA: Large-Scale Physical-layer Identifications of RFIDs via Dual Natural Attributes," in *Proc. of ACM MobiCom*, 2022.

9. Zhenlin An, Qiongzheng Lin, Xiaopeng Zhao, Lei Yang, Dongjiang Zheng, Guiqing Wu, Shan Chang, "One Tag, Two Codes: Identifying Optical Barcodes with NFC," in *Proc. of ACM MobiCom*, 2021.

Journal Articles

1. Xiaopeng Zhao, Guosheng Wang, Zhenlin An, Qingrui Pan, Qiongzheng Lin, Lei Yang, "Pushing the Boundaries of High-Precision AoA Estimation with Enhanced Phase Estimation Protocol," *IEEE Internet of Things Journal*, 2024.

2. Qingrui Pan, Zhenlin An, Xiaopeng Zhao, Lei Yang, "The Power of Precision: High-Resolution Backscatter Frequency Drift in RFID Identification," *IEEE Transactions on Mobile Computing*, 2023.

3. Xueyuan Yang, Zhenlin An, Xiaopeng Zhao, Lei Yang, "Transfer Beamforming via Beamforming for Transfer," *IEEE Transactions on Mobile Computing*, 2023.

Demos and Posters

1. Jingyu Tong, Zhenlin An, Xiaopeng Zhao, Sicong Liao, Lei Yang, "Demo: Radio Frequency Neural Networks for Wireless Sensing," in *Proc. of ACM MobiCom Demo*, 2023. (Best Graduate Award)

2. Xiaopeng Zhao, Zhenlin An, Qingrui Pan, Lei Yang, "Understanding Wireless Channels through NeRF2," *ACM GetMobile*, 2024. (Invited paper)

Acknowledgments

Four years ago, when I decided to pursue my Ph.D. degree at the Hong Kong Polytechnic University, Prof. Lei Xie told me that it would be an unforgettable journey. At that moment, I could not fully grasp the depth of his words. Now, as I reflect on these four years, I realize that the path has been filled with challenges and growth. However, it was the support and companionship of many wonderful individuals that made this journey not only possible but also meaningful. These collective experiences have indeed shaped an unforgettable chapter in my life.

First and foremost, I would like to express my gratitude to my supervisor, Dr. Lei Yang. His guidance has been crucial in developing my academic knowledge and research skills, enabling me to conduct original, systematic, and comprehensive research. His creative and challenging ideas provided me with a deep understanding of what it means to be a scholar. From him, I have gained an appreciation for research taste, learned how to discover novel research ideas, and honed my academic writing and presentation skills. It is through his mentorship that I have engaged in interesting research, ultimately leading to the completion of this thesis.

I am also grateful to my collaborators and partners throughout my Ph.D. journey. Thanks to Dr. Zhenlin An and Dr. Qingrui Pan for their help in the early stage of my Ph.D. study, which deepened my understanding of the wireless communication field. My appreciation goes to Mr. Jingyu Tong, Mr. Donghui Dai, Mr. Zheng Gong, Mr. Zhimin Mei, Mr. Sicong Liao, Dr. Yuanhao Feng, Ms. Xuanzhi Wang, and

Ms. Xueyuan Yang. Their expertise in various research topics and system design has expanded my academic views, and our discussions have been invaluable to my research. It has been a pleasure to collaborate with Mr. Shen Wang, Mr. Guosheng Wang, and Mr. Kaijia Xu, whose quick thinking and implementation skills have consistently impressed me. I am also thankful to my research group members, Mr. Zhicheng Wang, Mr. Fengrui Zhang, and all other collaborators, for their shared experiences and friendship throughout this journey.

Finally, the thanks deep in my heart go to my beloved parents and my girlfriend, Ms. Jinmei Liu, for their support, encouragement, and love throughout my graduate school career, especially during the period of COVID-19. My Ph.D. journey was marked by numerous setbacks, including receiving 12 rejection letters out of 16 submissions. It was their constant support that gave me the strength to persevere and believe in myself.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In recent years, the Internet of Things (IoT) has emerged as a pivotal area of technology, witnessing unprecedented growth across various applications. According to a recent industry analysis, the global IoT market was valued at USD 406 billion in 2023 and is projected to expand to approximately USD 3,152 billion by 2033 [1]. This rapid expansion is supported by a diverse range of wireless technologies, each designed to meet the specific demands of varied applications. For example, in large-scale agricultural operations, technologies such as LoRa [2] and NB-IoT [3,4] are frequently used in transmitting sensor data, including soil moisture levels [5], temperature readings [6], and humidity metrics [7]. In residential environments, smart technology commonly employs Wi-Fi and Bluetooth for tasks such as indoor localization [8–12] and gesture recognition [13–16]. RFID technology has become synonymous with enhancing efficiency in retail and logistics through automation [17]. Initially, these wireless technologies were primarily designed to support communication across varying ranges and at different data rates. Furthermore, ongoing research has broadened their applications to include sensing capabilities that utilize the wireless signals themselves.

In order to cater to diverse applications, IoT devices typically operate at varying frequencies and utilize different modulation schemes tailored to specific operational environments. Wireless technologies span a broad spectrum, ranging from several MHz, as observed in Near Field Communication (NFC), to the THz frequencies used in advanced communication systems. To standardize the description of signal propagation across these varied frequencies, researchers commonly employ a unified wireless channel model. This model is characterized principally by amplitude attenuation and phase rotation, parameters that are typically quantified using Channel State Information (CSI). Accurate channel estimation plays a key role in optimizing the performance of wireless communications and sensing technologies. For example, the efficiency of massive Multiple Input Multiple Output (MIMO) technology, which is crucial for achieving high data rates and enhanced network capacity, relies heavily on precise channel information. This information is essential for implementing effective precoding strategies. Even slight errors in channel estimation can cause signal interference among users, drastically reducing the Signal-to-Interference-plus-Noise Ratio (SINR) and, consequently, degrading overall data throughput. Similarly, in the realm of wireless sensing, like wireless localization technologies, the accuracy of channel estimation is also critical. Precise channel information is essential for accurately determining metrics such as Time-of-Flight (ToF) and Angle-of-Arrival (AoA), which are important to the functionality of subsequent localization algorithms. Minor deviations in channel estimation can lead to significant errors in ToF and AoA calculations, substantially increasing the localization error and impacting the reliability of the localization systems.

Research into wireless channel acquisition can be roughly categorized into two approaches: precise measurement of the wireless channel and channel prediction without direct measurement. For most systems operating below the GHz range and with reciprocal channels, the acquisition of accurate channel information typically relies on the use of training pilots. These systems transmit predefined signals to facilitate

the estimation of Channel State Information (CSI) between the transmitter and receiver. However, this method encounters several challenges. Factors such as Carrier Frequency Offset (CFO), Carrier Phase Offset (CPO), and jitter from crystal oscillators can introduce substantial errors in channel estimation. While researchers have developed algorithms to compensate for these constant offsets, achieving effective performance, the elimination of random offsets that are introduced by the system still remains a challenge. In systems with massive antenna arrays, the time and computational demands of channel estimation are considerable. Moreover, direct channel estimation, which involves a single measurement, is often impractical. For example, in Frequency Division Duplex (FDD) systems, the uplink and downlink channels operate at different frequencies, precluding the direct use of uplink measurements for downlink beamforming. Similarly, in THz communication systems, the uplink and downlink do not follow the same path, complicating direct measurement further. Consequently, the development of methods for channel prediction without direct measurement, or through side channels, has emerged as an important area of research in recent years.

In this dissertation, we introduce a novel algorithm termed neural radio-frequency radiance fields (NeRF^2), which represents the scene as a continuous volumetric function describing electromagnetic (EM) properties. This model is trained to predict the wireless channel with high precision when the transmitter or receiver is positioned at any chosen location in the environment. Additionally, we develop a robust phase estimation algorithm for backscatter communications, effectively mitigating the random phase noise attributable to oscillator jitters. As an approach in the physical layer of wireless communications, the capabilities of NeRF^2 extend across multiple practical applications. These include enhanced channel estimation in FDD systems, improved accuracy in wireless localization techniques, and the pioneering in 3D voxel mapping for reconstruction. Each application demonstrates the potential of NeRF^2 to revolutionize existing methodologies by providing a more detailed and accurate representation of the electromagnetic environment, thereby facilitating more efficient

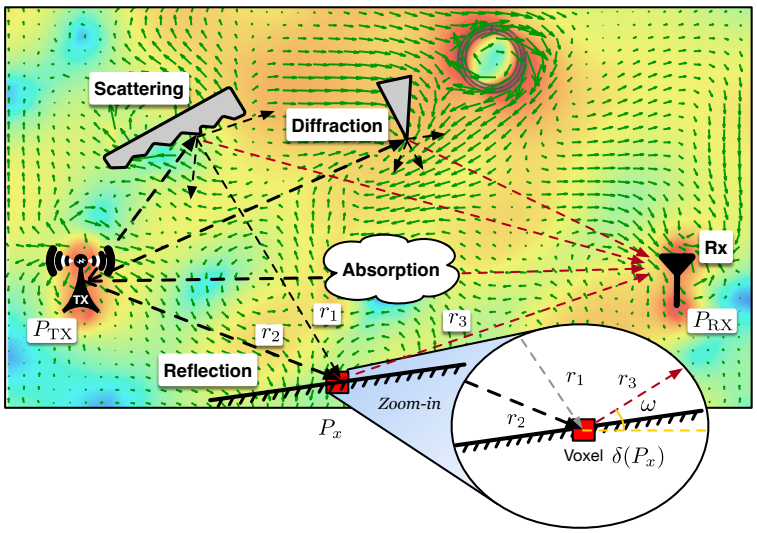


Fig. 1.1: Propagation of RF signal in complicated environment.

and reliable wireless communications and sensing.

1.2 Motivation

The propagation of RF signals, commonly referred to as the wireless channel, can be theoretically described by Maxwell's equations. However, the practical application of these equations is limited by the need for precise environmental parameters, such as the dielectric coefficients of materials, which are often difficult to acquire accurately in real-world scenarios. Furthermore, solving Maxwell's equations is both time-intensive and computationally demanding, exemplified by methods like the Finite-Difference Time-Domain (FDTD) solver. To address these challenges, researchers have developed simpler models to approximate the propagation of RF signals. One such model is the Friis transmission equation, used in free space, where the amplitude of an RF signal decreases inversely with the square of the distance from the source. However, in more complex environments, RF signal interactions become significantly more intricate due to absorption, reflection, diffraction, and scattering effects. As illustrated in Fig. 1.1, these interactions disrupt the entire radiance field:

(1) Reflection: This occurs when EM rays encounter surfaces that do not absorb all the incident energy. According to the law of reflection, the angle at which the ray strikes the surface (angle of incidence) equals the angle at which it departs (angle of reflection), relative to the normal of the surface. This law is for predicting signal paths in environments with many reflective surfaces, such as urban landscapes or indoor settings with metallic or glass structures.

(2) Absorption: While not always included in simplified ray tracing models, absorption is important in accurately predicting RF signal strength across various media. It occurs when the energy of an EM ray is absorbed by the medium, converting to other forms of energy like heat. The degree of absorption depends on the material's properties, such as its permittivity and conductivity, and influences how deeply a signal can penetrate a particular medium.

(3) Refraction: It occurs when an EM ray passes from one medium to another with a different refractive index. According to Snell's Law [18], the ratio of the sine of the angle of incidence to the sine of the angle of refraction is equivalent to the ratio of the speeds of light in the two media, or equivalently, the inverse ratio of the indices of refraction. This principle is essential for modeling signal propagation through different materials, such as air, water, or glass.

(4) Diffraction: This phenomenon describes the bending of EM rays around obstacles and the spreading of waves when they pass through narrow openings. The principles of diffraction are explained by the Huygens-Fresnel Principle, which considers every point on a wavefront as a source of secondary spherical wavelets. The wavefront at any subsequent time is the tangential surface to these wavelets. The Uniform Theory of Diffraction [19] further quantifies diffraction, particularly around edges, by introducing the concept of the diffracted ray, which extends the geometric optics beyond the shadow boundaries.

(5) Scattering: It occurs when the path of an EM ray is randomly redirected as

a result of interactions with small particles or irregularities in a medium. Unlike specular reflection, scattering is a more complex phenomenon that depends on the wavelength of the EM ray and the size and nature of the scattering elements. Rayleigh scattering, for example, occurs when particles are much smaller than the wavelength of the radiation, whereas Mie scattering happens when the particles are about the same size as the wavelength. This scattering is more pronounced with high-frequency RF signals, such as those used in THz communications [20], complicating the prediction of signal paths and intensities due to the random nature of the interactions.

These complexities motivated the development of more adaptable and efficient models for predicting RF signal behavior in varied environments, which is a central focus of this thesis. To address the complexities associated with RF signal propagation, conventional algorithms often employ electromagnetic (EM) ray tracing. This method involves simulating real EM rays to accurately trace the path an RF signal would follow in a real-world environment, thus allowing a more realistic simulation of signal interactions with obstacles. EM ray tracing is based on three core components: the laws of ray propagation, the geometric structure of the environment, and the EM coefficients of the materials involved. The laws of ray propagation mentioned above are fundamental to understanding how RF signals behave in different environments. These laws consist of several key principles and phenomena that dictate the trajectory and interactions of EM rays as they travel through space and interact with various materials and structures. Accurately capturing the environmental geometry is also essential for effective ray tracing. There are two primary approaches to obtaining these structures: manual modeling and automated reconstruction. The first approach uses 3D modeling software to construct the environment manually. Although this method can be time-consuming and often requires many adjustments to capture fine details, it allows for a controlled simulation environment. Some software solutions offer standard components to speed up this process, leading to the loss of intricate details. The second approach uses techniques such as those employed by COLMAP, which

reconstruct structures from images or LiDAR [21, 22] data. While these methods are quicker and can handle large-scale environments, they often only provide a rough approximation of the geometry, which may be sufficient for low-frequency signals but inadequate for high-frequency applications such as millimeter waves. The finer details are important at these frequencies, as even minor inaccuracies in geometry can lead to large errors in signal propagation paths. The EM properties of materials substantially influence ray tracing accuracy. Real-world materials often consist of complex mixtures, and accurately determining their EM coefficients can require costly equipment and may not be feasible on a large scale. Historically, ray tracing algorithms have relied on empirical coefficients, which can lead to discrepancies between simulated results and actual signal behavior.

1.3 Research Scope and Contribution

As shown in Fig. 1.2, the scope of this dissertation is organized into three distinct layers: the hardware layer, the physical layer, and the application layer. At the foundational hardware level, we have developed the Ray dataset, a comprehensive 3D indoor localization dataset that contains 1.6 million samples spread across 50 diverse scenarios. This dataset includes data from RFID, Wi-Fi, and Bluetooth technologies. Previous databases in this field have been limited by their scale, diversity of scenes, precision of labeling, and overall coverage, rendering them inadequate for general-purpose use in benchmarking, training, and transfer learning. To address these deficiencies, we design a real-world system capable of capturing a large-scale dataset that supports subsequent research. In the physical layer, our objective is to achieve precise wireless channel measurement and prediction. To this end, we have introduced a consistent phase estimation algorithm specifically for backscatter phase measurement and developed the Neural Radio-Frequency Radiance Fields (NeRF^2) for predictive modeling of wireless channels. The predictive capability of NeRF^2 will

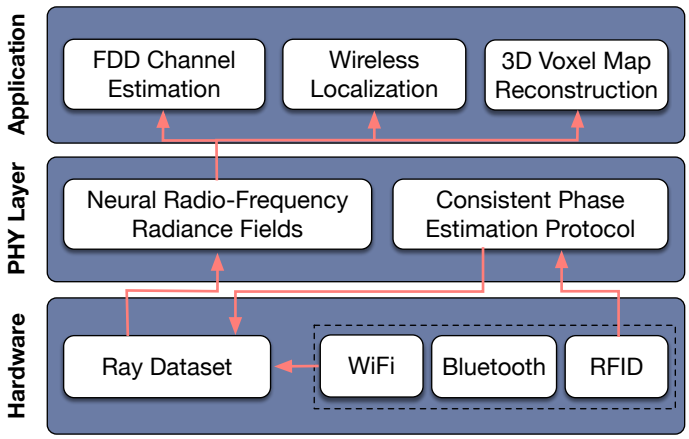


Fig. 1.2: Research Framework of the dissertation.

benefit the tasks in the upstream application layer, such as Frequency Division Duplex (FDD) channel estimation. NeRF^2 can generate synthetic training datasets of high quality, comparable to real datasets. This capability enhances data-driven research in areas like deep-learning-based wireless indoor localization, where extensive wireless channel information is required for training. Furthermore, by integrating wireless channel models with statistical data models, NeRF^2 can learn the physical properties of a scene. This learning facilitates the extraction of specific locations within a scene that influence wireless channel behavior, thereby enabling the reconstruction of detailed 3D voxel maps. The contributions of the dissertation are summarized as follows:

1.3.1 Physical Layer: Channel Measurement & Prediction

- **Neural Radio-Frequency Radiance Fields**: Although Maxwell discovered the physical laws of electromagnetic waves 160 years ago, accurately modeling the propagation of RF signals in large and complex environments remains a long-standing challenge. This difficulty arises from the intricate interactions between the RF signal and various obstacles, such as reflections and diffractions. Drawing inspiration from the successful application of neural networks in modeling optical fields within

computer vision, we introduce a novel concept called the neural radio-frequency radiance field, denoted as NeRF^2. This approach models a continuous volumetric scene function that interprets RF signal propagation effectively. After initial training using a limited set of signal measurements, NeRF^2 is capable of predicting signal reception at any location, given the transmitter's position is known. As a physical-layer neural network, NeRF^2 combines statistical learning with physical ray-tracing models to create synthetic datasets that cater to the specific training requirements of application-layer artificial neural networks (ANNs). By employing what we term "turbo-learning," which integrates real and synthetic datasets, we improve ANNs' training effectiveness. Our experimental results reveal that this method can boost ANN performance by approximately 50%. Furthermore, NeRF^2 demonstrates considerable potential in applications such as indoor localization and 5G MIMO technologies.

- **Consistent Phase Estimation Protocols for Backscatters**: The development of high-precision indoor backscatter tag tracking in environments lacking GPS has many applications, from virtual reality to factory automation. Yet, the effective tracking range of this high-precision technology is constrained to just a few meters, limiting the deployment of backscatters to areas close to checkpoints in warehouses, despite their ability to communicate over 50 meters. This confined localization range primarily stems from the "butterfly effect" in localization systems, where minor phase measurement errors can lead to large localization discrepancies. We introduce two innovative phase estimation protocols aimed at addressing the core challenges of achieving accurate phase estimation over long distances. The first protocol, CPE, effectively resolves the π-ambiguity often found in commercial RFID readers. Expanding on this, CPE+ is designed to eliminate flicker noise and neutral white noise, and to correct spatial and temporal imbalances. Our experimental results show that CPE+ extends the range of precise AoA estimation and centimeter-level localization from 8 meters to 15 meters in stationary settings. It also maintains decimeter-level accuracy across the full 50-meter communication

range when used with two or more gateways.

1.3.2 Application Layer: Localization & Mapping

- **Understanding Localization by a Tailored GPT**: Over the past few decades, the field of indoor localization has evolved from relying primarily on signal processing techniques to adopting deep learning strategies, particularly to address complex issues like the multipath effect. Traditional deep learning methods for this purpose often face challenges due to their dependence on high-quality training samples for supervised learning, along with limited adaptability in different environments. To overcome these limitations, we have developed an innovative hierarchical neural network architecture adapted from the Transformer model, known for its exceptional ability to capture contextual nuances. Additionally, we have introduced two novel loss functions that facilitate semi-supervised training. Our microbenchmark results clearly illustrate the effectiveness of our approach, with performance enhancements ranging from 30% to 70% in a varied set of 50 scenarios, surpassing other contemporary methods. To enhance transferability, we introduce a specialized variant of the Generative Pre-training Transformer (GPT), called LocGPT, which comprises 36 million parameters specifically optimized for transfer learning. By fine-tuning this pre-trained model, we achieve comparable accuracy with only half of the usual dataset size, marking a significant advancement in transfer learning within the indoor localization field.
- **3D Voxel Maps Reconstruction via Satellitic Radiance Fields**: In urban planning and research, 3D city maps play an essential role in supporting various activities, such as cellular network design, urban development, and climate research. Traditional methods of creating these models have involved expensive techniques such as manual 3D mapping, interpreting satellite or aerial images, or utilizing sophisticated depth-sensing equipment. In this work, we introduce a novel method for developing 3D urban maps by analyzing the impact of urban structures on

satellite signals, utilizing GPS data crowdsourced from hundreds of smartphones during routine user movements. We propose the concept of Satellitic Radiance Fields (SaRF), an innovative neural scene representation technique that captures the distribution of GPS signals in urban environments. SaRF uses a sparse voxel octree framework to represent voxel-centric implicit fields, which detail the physical properties, such as the density, of each voxel. This model is progressively refined through a differentiable ray-marching process, which culminates in the accurate reconstruction of 3D urban maps. Our comprehensive experimental evaluation, which includes approximately 27.4 million GPS records, demonstrates an average reconstruction accuracy of 83.1% across six diverse urban environments.

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows:

- In Chapter 2, we present a wide literature review on existing methodologies in the field of wireless communications, with a focus on channel estimation and wireless localization. The review first delves into traditional and neural channel representation approaches for channel estimation. This includes a detailed discussion on the integration of neural networks and physical models, exploring their application in simulating and enhancing RF signal propagation predictions. We also assess various localization algorithms ranging from fingerprint-based methods to angle-based strategies, demonstrating the challenges that have shaped current practices.
- In Chapter 3, we delves into the architectural components of a typical backscatter network. Key discussions cover the modulation of signals by backscatter tags through impedance switching and the triangulation methods used by gateways to calculate tag locations. Additionally, the chapter details the channel estimation processes involved in backscatter systems. Finally, we introduce the Ray dataset

for the following evaluation.

- In Chapter 4, we introduce a deep learning framework for physical layer channel prediction, terms the neural radio-frequency radiance field (NeRF^2), which is a key algorithm in the thesis. NeRF^2 integral the neural networks and physical ray-tracing models. This approach creates a continuous volumetric scene function that effectively predicts RF signal behavior based on a sparse set of initial measurements. By combining real and synthetic datasets through a method we describe as 'turbo-learning,' NeRF^2 enhances the training and performance of application-layer artificial neural networks by up to 50%. This chapter explores NeRF^2's significant implications for indoor localization and advanced 5G MIMO technologies.
- In Chapter 5, we introduce two phase estimation protocols, CPE and CPE+. Despite backscatter tags having a communication range of over 50 meters, their high-precision tracking capabilities are typically limited to just a few meters. This limitation, often a result of the "butterfly effect" where minor phase measurement errors cause significant localization discrepancies, restricts their broader application. These protocols enhance accuracy over long distances, with CPE addressing the π-ambiguity in commercial RFID readers and CPE+, further reducing flicker noise and neutral white noise while correcting spatial and temporal imbalances.
- In Chapter 6, we introduce a novel hierarchical neural network architecture based on the Transformer model. This architecture is enhanced with two innovative loss functions that support semi-supervised training. Our microbenchmark tests demonstrate significant performance improvements, with gains ranging from 30% to 70% across 50 different scenarios, outperforming existing methods. Additionally, we present LocGPT, a specialized variant of the Generative Pre-training Transformer optimized for indoor localization. This model, with 36 million parameters, significantly enhances transfer learning capabilities, achieving high accuracy with only half the typical dataset size.
- In Chapter 7, we propose an approach using GPS data crowdsourced from hundreds of smartphones to analyze the impact of urban structures on satellite signals. This

method involves the novel Satellitic Radiance Fields (SaRF) technique, a neural scene representation that uses a sparse voxel octree framework to detail the density of each voxel in urban environments. Enhanced by a differentiable ray-marching process, this technique enables the precise reconstruction of 3D urban maps. Our extensive experiments, involving approximately 27.4 million GPS records, achieve an average reconstruction accuracy of 83.1% in six diverse urban settings.

- In Chapter 8, we encapsulates the key contributions of the dissertation, including the development of the NeRF^2 framework to enhance wireless channel predictions, a phase estimation protocol for precise localization , and a Transformer-based Localization model in indoor localization across multiple wireless technologies. Additionally, the chapter outlines prospective future directions, emphasizing the need for model adaptations to dynamic environments using explicit neural representations and differentiable ray tracing.

Chapter 2

Literature Review

2.1 Wireless Channel Estimation

In wireless systems, accurately predicting how signals propagate through different environments—such as urban landscapes or indoor scenarios—is crucial for optimizing network performance and reliability. Traditional methods often rely on physical models that, while effective, may not always capture the dynamic and complex nature of real-world environments. Neural channel representations, however, utilize the robust data-processing capabilities of deep learning to model these complexities in a more nuanced and adaptable manner.

2.1.1 Conventional Channel Estimation

Channel estimation plays a pivotal role in the optimization of wireless systems. Historically, various methods have been employed to estimate the wireless channel. These methods include the use of training pilots [23], which provide a known signal that can be used to estimate channel characteristics. Feedback mechanisms [24] involve the receiver sending information back to the transmitter, helping to refine the channel

estimation process. Additionally, parametric or empirical models [25] have been used, which rely on predefined channel models or empirical data to predict channel states. However, as wireless networks evolve with increasing numbers of antennas and higher frequency bands, traditional feedback-based and pilot-based methods face significant challenges. These methods often result in excessive overhead due to the large amount of data required for accurate estimation, making them less feasible in systems with massive MIMO configurations [25]. To address these limitations, recent studies have explored the use of machine learning techniques. For instance, methods like those proposed in FIRE [26] utilize generative models based on Variational Autoencoders (VAE) to learn the downlink channel from uplink feedback. Other approaches, such as NNCONFIG [27], map trained neural networks to intelligent reflective surfaces to enhance channel estimation. R2F2 [28] focuses on improving channel prediction in wireless communication systems by leveraging machine learning. The proposed method reduces the complexity of predicting the downlink channel based on the observed uplink channel using a neural network trained on standard channel models. Despite these advancements, many of the newer, purely data-driven machine learning models depend heavily on large datasets, which may not always be available or practical to obtain. Moreover, these models often lack interpretability because they do not integrate well-understood physical principles governing wave propagation. In contrast, NeRF^2 introduces a novel approach by incorporating a physical model—specifically, the RF radiance field—into the learning process. This integration not only enhances the interpretability of the neural network model but also improves the accuracy of channel learning by utilizing prior knowledge of wave transmission behaviors. This method stands out by bridging the gap between empirical data reliance and physical model precision, paving the way for more reliable and efficient channel estimation in complex wireless environments.

2.1.2 Neural Channel Representation

Neural channel representation is a groundbreaking approach in wireless communications that leverages neural network models to simulate and predict radio frequency signal propagation. Recent research has further extended the utility of neural representation techniques in channel modeling. The applications of neural channel representation range from enhancing the precision of channel state information to improving the design of wireless networks for 5G and beyond. Key contributions in this area include the simulation of ray-surface interactions to better predict signal paths, especially in non-line-of-sight conditions where traditional methods struggle. For instance, Sionna [29] introduces a framework for neural-based simulation of wireless channels, leveraging the high-dimensional data handling capability of NeRF to capture complex wave interactions in urban environments. Additionally, RFGenesis [30] incorporates machine learning with physical models to predict and optimize RF propagation, showcasing substantial improvements in predicting non-line-of-sight (NLoS) paths which are often challenging for traditional channel estimation. Qualcomm's WiNeRT [31] employs neural representation to simulate ray-surface interactions, aiding in the estimation of RF signal propagation along transmit-receive paths. By integrating deep learning with traditional RF propagation models, this approach not only improves accuracy but also enhances the capability of wireless networks to dynamically adjust to their changing environments, paving the way for more robust and efficient communication technologies.

The design of NeRF^2 in this thesis draws inspiration from the development of Neural Radiance Fields (NeRF), initially conceived for generating novel views of complex scenes within the field of computer vision, as pioneered by Mildenhall et al. [32]. NeRF's innovative handling of light and space has been adeptly adapted in NeRF^2 to model the propagation of RF signals, marking a pioneering application in wireless communications—a field where accurate modeling of intricate, dynamic environments is crucial. Notably, NeRF^2 is among the first to adapt NeRF's methodologies to RF

signal modeling, incorporating modifications to accommodate the unique characteristics of RF waves, such as their phase properties, which necessitate a distinct physical tracing model [33–35]. Furthermore, constraints imposed by smaller antenna arrays, as opposed to the larger camera arrays used in visual NeRF, led to the development of two distinct training strategies tailored for NeRF^2. These strategies enhance the model's ability to precisely predict RF signal behavior, thereby benefiting a range of RF applications, including advanced wireless localization and MIMO channel prediction [36, 37]. Through the adaptation of NeRF models to suit electromagnetic wave behaviors, we have substantially improved the accuracy and flexibility of modeling wireless environments, addressing several limitations inherent in conventional channel estimation methods. This breakthrough paves the way for more robust and efficient communication technologies, potentially transforming practices in scene reconstruction, relighting, and view synthesis within the context of RF signal processing [38,39].

2.2 Wireless Indoor Localization System

2.2.1 Wireless Localization

Wireless localization is a long-studied topic with extensive works that investigate the process of locating a device by building the transmission model between the position and various metrics of received RF signals. It is particularly useful in indoor environments where GPS signals may be unreliable or unavailable. These methods contain several techniques, such as Received Signal Strength Indicator (RSSI), phase, Channel State Information (CSI), Time-of-Flight (ToF), and Angle of Arrival (AoA). They are widely applied in Wi-Fi, Bluetooth, RFID, and LoRa technologies. However, traditional geometric or empirical localization models face challenges due to environmental factors such as complex multipath effects and non-line-of-sight conditions. Recent advancements leverage deep learning models to enhance accuracy in complex

indoor scenarios, though they often require extensive training data, which can limit real-world applications. Many deep learning based indoor localization methods can be considered as an enhanced version of the fingerprint-based localization.

Fingerprint-based localization is a technique that involves collecting the distinct "fingerprint" or signal profiles of specific locations within an environment, utilizing metrics such as RSSI, CSI, or other relevant signal properties. These fingerprints are then used to identify or estimate the location of devices or objects based on their real-time signal readings, which are matched against this pre-established database of fingerprints. One of the pioneering implementations of this approach is the LANDMARC system [40]. The system integrates active RFID technology with strategically placed reference tags to refine the accuracy of location estimations. By using these reference tags as known points, the system can better interpolate or calculate the position of target tags based on their signal similarities with the references. This innovative use of reference tags addresses some of the critical challenges in RSSI-based fingerprinting methods, such as the signal attenuation and variability caused by obstacles and multipath effects, which are prevalent in complex indoor environments. Further exploring the limitations of RSSI, Yang et al. [41] highlighted how multipath fading and the dynamic nature of indoor environments can deteriorate the performance of RSSI-based systems. In their research, they proposed the adoption of CSI, which provides a more granular view of the signal environment by capturing the multipath characteristics of wireless signals. The adoption of CSI not only offers a deeper insight into the signal degradation patterns but also enhances the accuracy and reliability of the localization system. The authors demonstrated that leveraging CSI can improve the precision of indoor localization systems, thus providing a robust alternative to traditional RSSI-based approaches. Wang et al. [42] presents a fine-grained RFID positioning system that is robust to multipath and non-line-of-sight conditions, which are common in real-world environments. The system leverages the spatial diversity of multipath signals to create unique fingerprints for each location, allowing for accurate positioning

even in challenging scenarios. [43] proposes a cooperative localization method that utilizes initial fingerprint-based estimates and physical constraints from peer-to-peer interactions to refine localization accuracy for multiple users simultaneously.

Fingerprint-based localization exhibits several inherent limitations that can impact its effectiveness in certain scenarios. One primary challenge is the labor-intensive and time-consuming process of initial fingerprint mapping, which requires comprehensive sampling of signal characteristics throughout the deployment area. This makes it less adaptable to dynamic environments where frequent updates may be necessary to maintain accuracy. Additionally, fingerprint-based systems are highly sensitive to environmental changes, such as alterations in room layout or the presence of new obstacles, which can significantly alter signal patterns and lead to discrepancies in localization accuracy. The reliance on historical signal data also introduces issues of temporal variability, where the aging of infrastructure or changes in ambient conditions can render stored fingerprints obsolete. Moreover, these systems often struggle in densely populated areas where interference from multiple devices can skew the signal data, further complicating the matching process and reducing the overall reliability of the localization.

2.2.2 Long-range Backscatter Localization

Long-range backscatter communication systems push the boundaries of traditional backscatter technologies by overcoming inherent range limitations. These limitations often stem from the inadequacies in energy harvesting capabilities of backscatter tags, which traditionally restricted their functional range. Recent advancements have focused on augmenting the received power at these tags using several innovative methods to extend their operational range. One effective strategy has been the deployment of multiple readers or antenna arrays equipped with beamforming techniques. Notable works such as those by PushID [44] and In-N-Out [24] have shown

substantial improvements in received power through these methods. In particular, PushID proposed a system that utilizes multiple antennas to enhance signal strength and reliability over greater distances. Furthermore, Ma et al. [45] presented the IVN that employs a blind MIMO system. This system transmits different carriers across a narrow bandwidth using multiple RF sources, thus expanding the potential communication range without requiring extensive power consumption. This approach diverges from traditional single-frequency backscatter communications. Tang et al. [46] have explored the utilization of ultralow-power consumption circuits coupled with direct digital frequency synthesis. Their technique supports efficient operation and extends the communication range of backscatter systems. The tunnel emitter proposed by Varshney et al. [47] reduces the power required for carrier signal generation to just tens of microwatts, enabling battery-free operation and enhancing long-range communication capabilities.

While advancements in backscatter communication have notably increased its range, the accuracy of commercial tag position estimation at long distances remains less explored. Techniques that enhance communication range do not inherently improve localization accuracy, especially for cost-effective tags. For instance, Qi et al. [48] achieved RFID localization up to 35 meters using a specialized, but costly, quantum tunneling tag. This highlights a significant challenge: low RSS can sustain communication but may lead to increased localization errors due to phase inaccuracies, marking a critical direction for future research in backscatter systems.

Various metrics of RF signals are widely used for backscatter localization: RSSI [49], carrier phase [50], Channel Impulse Response (CIR) [51], Time-of-Flight (ToF) [9], and AoA [52]. The main challenge that the past AoA algorithms addressed is the multipath effect, with the assumption that the received signals have strong RSS. For example, Arraytrack [53] and SWAN [10] decompose a combined signal into different directions by using an antenna array. The line-of-sight propagation will peak at the spatial spectrum. These algorithms have been demonstrated to be relatively effective

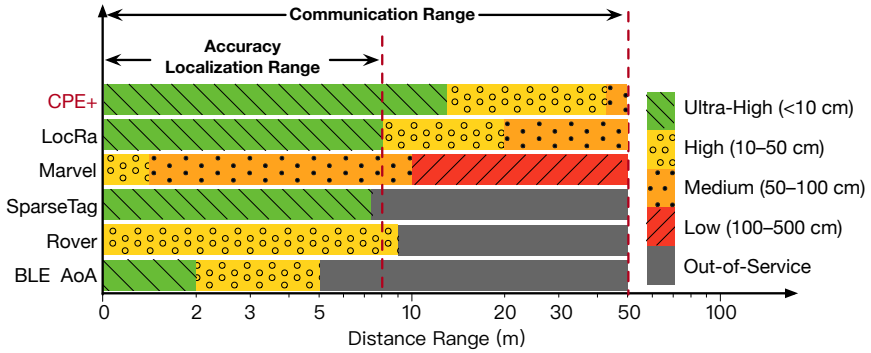


Fig. 2.1: Localization Accuracy. CPE maintains ultra-high precision within 15 m and high precision in the entire communication range.

in dealing with the multipath effect. However, when feeding with ultra-low signals, their performance degrades greatly at a long distance due to phase noise.

In Fig. 2.1, we compare CPE+ with conventional backscatter localization systems used in sub-6 GHz IoT in terms of working range. We categorize accuracy into five levels: ultra-high (<10 cm), high (10–50 cm), medium (50–100 cm), low (100–500 cm), and out-of-service. Specifically, LocRa [51] and Marvel [50] are two representative works of LoRa backscatter systems. LocRa achieves accurate localization with distributed base stations by extracting precise channel information and synchronizing phase among stations, reporting errors as low as 6.8 cm at close ranges and 88 cm at distances up to 50 meters. Marvel, designed for MAVs in indoor environments, supports autonomous navigation and achieves an accuracy of 2.7 m at 50 m without the aid of an Inertial Measurement Unit (IMU). Both systems maintain localization capabilities across their entire communication range of 50 m. However, accuracy decreases as the range extends, due to the diminishing RSS of received backscatter RF signals. SparseTag [52] leverages a sparse RFID tag array for high-precision indoor localization, while Rover [54] is an indoor localization system using a robot with inertial sensors to localize backscatter tags without prior site knowledge. BLE AoA [55] integrates backscatter technology with Bluetooth systems, conducting AoA localization through the advertising channel. It achieves a localization error of less than 10

cm within a 2 m deployment range and an accuracy of 0.31 m at 5 m. Nevertheless, their working ranges are extremely limited. In contrast, our system extends ultra-high precision localization to a range of 1–15 m, and maintains high/medium precision up to 15–50 m.

2.2.3 Phase Estimation

Most current phase-based localization methods rely on COTS RFID readers for phase value acquisition, primarily focusing on short to medium-range distances due to limitations in signal strength and phase accuracy over extended ranges. Studies such as Tao et al. [56] have demonstrated novel approaches combining phase difference with readability and phase-of-arrival methods to achieve accurate localization in sparse tag environments, respectively. Qi et al. [48] introduced a phase-based ranging method using a quantum tunneling tag, reducing the phase noise on dedicated hardware. Our CPE is inspired by [57], which introduced the temporal inconsistency of the phase estimation. Miesen et al. used the predefined patterns of the signals to sort out the correct direction of the complex difference. On the contrary, we directly use the transitional sample pairs for the estimation. Importantly, we maintain consistency in complex division instead of complex differences without any pattern recognition. Our contribution lies in pushing the boundaries of phase estimation for long-distance applications, offering a significant leap over existing COTS RFID reader capabilities, and providing a versatile solution applicable across a wide array of AoA algorithms and practical scenarios.

Chapter 3

Preliminaries

3.1 Backscatter System

Battery-free backscatters provide a promising low-power approach for the Internet of Things (IoT) to achieve truly ubiquitous computing. The research community has proposed various systems by backscattering TV tower [58], RFID reader [59], Wi-Fi [60], Bluetooth [61], ZigBee [62], or LoRa [63] signals. Fig. 3.1 shows the architecture of a typical backscatter network, which contains three main components: an RF source, backscatter tags, and a gateway.

- **RF source**: In the past, public RF stations (e.g., TV tower or RFID reader [59] and FM stations [64]) were preferably reused as the RF source to supply energy to the backscatter sensors. Nowadays, numerous studies tend to establish a dedicated RF source (e.g., RFID reader) or convert an existing RF device (e.g., Wi-Fi APs [65] and Bluetooth devices [66]) into a stable RF source, which transmits a continuous wave (CW) to power up tags.
- **Backscatter tags**: Battery-free backscatter tags harvest energy from the CW. They switch their internal impedance between two states (reflective and non-

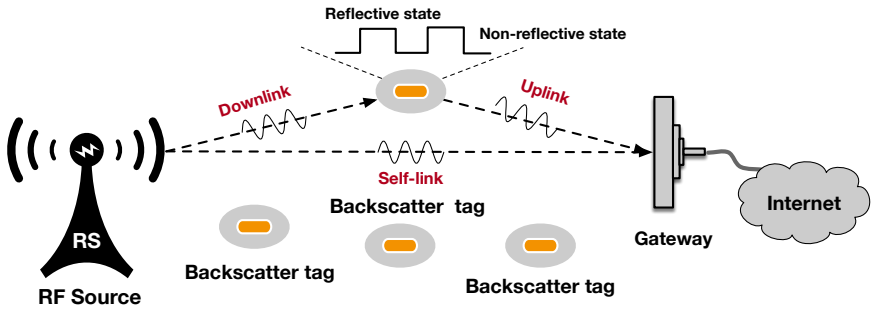


Fig. 3.1: Architecture of a backscatter network. A backscatter network architecture comprising an RF source emitting continuous waves to energize battery-free tags, which modulate the signal through impedance switching, and a gateway receiving the modulated packets.

reflective) to reflect the CW for on-off modulation.

- **Gateway**: The packets transmitted from backscatters are then received by a gateway. In our system, we equip each gateway with a 4×4 antenna array to estimate the direction of tags. Two or more gateways are eventually deployed at different positions to triangulate the locations of tags.

3.1.1 Primer on Backscatter Communication

As shown in Fig. 3.1, three communication links exist among the three components, as described below: **(1) Downlink (RF Source → Tag)**. The RF source transmits a single-tone continuous wave (CW) at some frequency f, which can be expressed as follows:

$$S_{R \to T}(t) = A_{R \to T} e^{\mathbf{J}(2\pi f t + \phi_R)} \tag{3.1}$$

where $A_{R \to T}$ denotes the attenuation from RF source to tag; ϕ_R is a combination of the initial phase offset, including the phase noise caused by the transmitter, and the phase rotation over the distance between the RF source and tag. For clarity, throughout this paper, the notation X → Y is employed to denote the direction from X to Y in mathematical expressions. For example, $S_{R \to T}$ represents the signal transmitted from the RF source (R) to the tag (T). **(2) Uplink (Tag → Gateway)**.

The backscatter tag modules the data by reflecting or not reflecting the incoming CW. Let $S_T(t)$ denote the two states of the tag, reflective and non-reflective state, which is formally defined as follows:

$$S_T(t) = S_{R\to T}(t) \cdot A_T e^{\mathbf{J}\phi_T} \text{ and } A_T \in \{0, 1\} \tag{3.2}$$

where $A_T = 0$ or 1 if the tag is in the non-reflective or reflective state, respectively. ϕ_T is the phase offset caused by the tag (e.g., rotation or posture). A signal transmitted from the tag to the gateway can be given as

$$S_{T\to G}(t) = S_T(t) \cdot A_{T\to G} e^{\mathbf{J}\theta_{T\to G}} \tag{3.3}$$

Similarly, $A_{T\to G}$ and $\theta_{T\to G}$ denote the attenuation and phase rotation over the distance between the tag and gateway, respectively. Substituting Eqn. 3.1 and Eqn. 3.2 into Eqn. 3.3, we can obtain the final signal received by the gateway through the uplink as follows:

$$S_{T\to G}(t) = A_{R\to T} A_T A_{T\to G} e^{\mathbf{J}(2\pi f t + \theta_{T\to G} + \phi_R + \phi_T)} \tag{3.4}$$

Our key task is to measure $\theta_{T\to G}$ because it is the only parameter related to the location of the tag from the view of the gateway. **(3) Self-link (RF Source → Gateway)**. CW is always present regardless of whether the tag backscatters or not. This link is usually called self-interference link because it causes interferences in the gateway. Self-link can be expressed as follows:

$$S_{R\to G} = A_{R\to G} e^{\mathbf{J}(2\pi f t + \theta_{R\to G} + \phi_R)} \tag{3.5}$$

Similarly, $A_{R\to G}$ and $\theta_{R\to G}$ denote the path attenuation and phase rotation over the distance between the RF source and gateway. CW might be reflected from surrounding objects, such as walls, ceilings, or furniture. We can view these reflections and the direct propagation as a single combined signal above.

3.1.2 Channel Estimation for Backscatter

Each antenna at the gateway receives a mixed signal propagated through the uplink and the self-link at the frequency f as shown below:

$$S_G(t) = \begin{cases} S_{T\to G}(t) + S_{R\to G}(t) & A_T = 1 \text{ reflective} \\ S_{R\to G}(t) & A_T = 0 \text{ non-reflective} \end{cases} \tag{3.6}$$

The gateway downconverts the above received signal into a baseband signal by multiplying $S_G(t)$ by $e^{\mathbf{J}-2\pi f' t+\phi_G}$. Let ϕ_G be the phase offset plus the phase noise caused by the gateway's receiver. Eqn. 3.1, Eqn. 3.3 and Eqn. 3.5 are substituted into Eqn. 3.6, and the downconverted baseband signals can be given as follows:

$$\begin{cases} \overline{S}_G(t) = \overline{A} \cdot e^{\mathbf{J}(2\pi\Delta f t+\theta_{R\to G}+\overline{\phi})} \\ \breve{S}_G(t) = \breve{A} \cdot e^{\mathbf{J}(2\pi\Delta f t+\theta_{T\to G}+\breve{\phi})} + \overline{A} \cdot e^{\mathbf{J}(2\pi\Delta f t+\theta_{R\to G}+\overline{\phi})} \end{cases} \tag{3.7}$$

where $\overline{S}_G$ and $\breve{S}_G$ denote the downconverted reflective signal and non-reflective signal, respectively. In the following, we use the notations of $\breve{x}$ and $\overline{x}$ to denote the variables related to the reflective signal and non-reflective signal, respectively. $\Delta f = (f - f')$ is called *carrier frequency offset* (CFO). By employing algorithms in [67], CFO estimation $\Delta\widehat{f}$ is first conducted and then compensated by multiplying the term $e^{-\mathbf{J}2\pi\Delta\widehat{f}t}$ to the baseband signal. This approach effectively eliminates the CFO in $\overline{S}_G(t)$ as follows:

$$\overline{S}_G(t) = \overline{A} \cdot e^{\mathbf{J}(2\pi\Delta f t+\theta_{R\to G}+\overline{\phi})} \cdot e^{-\mathbf{J}2\pi\Delta\widehat{f}t} \approx \overline{A} \cdot e^{\mathbf{J}(\theta_{R\to G}+\overline{\phi})} \tag{3.8}$$

This procedure can also be applied to $\breve{S}_G(t)$. $\breve{\phi}$ and $\overline{\phi}$ are phase offsets caused by hardware:

$$\breve{\phi} = \phi_R + \phi_T + \phi_G \text{ and } \overline{\phi} = \phi_R + \phi_G \tag{3.9}$$

Fig. 3.2 shows a downconverted baseband signal. The samples are clearly distributed in two clusters, which correspond to non-reflective $\overline{S}$ and reflective $\breve{S}$ signals. Eqn. 3.7 presents that only the reflective signal $\breve{S}_G(t)$ contains the desired true phase $\theta_{T\to G}$.

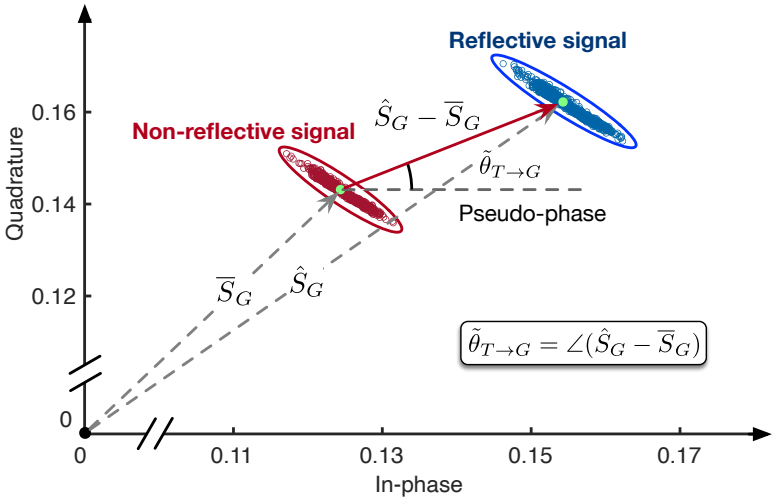


Fig. 3.2: The definition of pseudo-phase. The pseudo-phase is defined as the angle of the vector connecting the centers of the two clusters.

Hence, the naive approach is to subtract $\overline{S}_G(t)$ from $\breve{S}_G(t)$ as follows:

$$\Delta S_G(t) = \breve{S}_G(t) - \overline{S}_G(t) = \breve{A} \cdot e^{\mathbf{J}(\theta_{T\to G} + \breve{\phi})} \tag{3.10}$$

Taking the above *complex difference* eliminates the interference from the self-link but reserves the signal propagated from the uplink. Then, the phase is computed as the angle of the above complex difference as follows:

$$\begin{aligned} \widetilde{\theta}_{T\to G} &= \angle \Delta S_G(t) = \theta_{T\to G} + \breve{\phi} \\ &= \theta_{T\to G} + \phi_R + \phi_T + \phi_G \end{aligned} \tag{3.11}$$

Notably, $\widetilde{\theta}_{T\to G}$ is not the desired phase $\theta_{T\to G}$ but includes $\theta_{T\to G}$. To be distinguishable, we call $\widetilde{\theta}_{T\to G}$ and $\theta_{T\to G}$ *pseudo-phase* and *true phase*, respectively. In Fig. 3.2, the pseudo-phase $\widetilde{\theta}_{T\to G}$ is actually the angle of the vector connecting the centers of the two clusters.

3.1.3 Phase Estimation in Antenna Array

The estimated phase is typically utilized to calculate the angle of arrival (AoA) of the signal for localization purposes, employing an antenna array. An antenna array

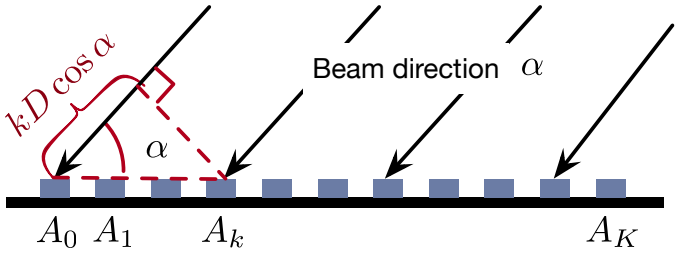


Fig. 3.3: Standard uniform linear array beam steering

(or array antenna) is a set of multiple connected antennas which work together as a single antenna, to transmit or receive RF signals. The individual antennas are called elements, which aim to receive the RF signals in different positions. We utilize an antenna array to disintegrate the received signals to figure out the AoA of the incoming RF signal, i.e., the direction of the target device that transmits the RF signal. Assume that the gateway is equipped with an antenna array consisting of K antennas, which are uniformly and linearly spaced, as depicted in Fig. 3.3. The array can project the received RF signals at all of its antennas in a desired direction α. This projection is conducted by multiplying the received signal at each antenna by a complex weight. The sum of all antennas' multiplication results creates a spatial filter that focuses on direction α while minimizing power in other directions. Specifically, let $S_{G_k} = e^{\mathbf{J}\theta_{\mathbf{T}\rightarrow\mathbf{G_k}}}$ represent the signal from the tag received by the k^{th} antenna in the array, where $\theta_{T\rightarrow G_k}$ denotes the phase rotation attributable to the distance from the tag to the k^{th} antenna, for $k = 0, \ldots, K-1$. Considering the leftmost antenna A_0 as the reference, we can compute the relative power received in a beam in direction α, as follows:

$$P(\alpha) = \left| \sum_{k=0}^{K-1} w(k,\alpha) \cdot \frac{S_{G_k}}{S_{G_0}} \right| = \left| \sum_{k=0}^{K-1} w(k,\alpha) \cdot e^{\mathbf{J}(\Delta\theta_{T\rightarrow G_k})} \right| \tag{3.12}$$

where

$$w(k,\alpha) = e^{-\mathbf{J}2\pi \frac{kD\cos\alpha}{\lambda}} \text{ and } \Delta\theta_{T\rightarrow G_k} = \theta_{T\rightarrow G_k} - \theta_{T\rightarrow G_0} \tag{3.13}$$

$w(k,\alpha)$ is the weight assigned to the k^{th} antenna when steering to direction α. λ is

the wavelength. D is the space between two adjacent antennas, i.e., $D < \lambda/2$. $\theta_{T\to G_k}$ is the phase rotation over the distance between the tag and antenna A_k. The figure shows that the RF signal travels an additional distance of $kD\cos(\alpha)$ to arrive at the antenna A_0 rather than the antenna A_k, which causes the *theoretical phase difference* of $2\pi \cdot kD\cos(\alpha)/\lambda$. $\Delta\theta_{T\to G_k}$ is the actual *measured phase difference* between the two antennas. As a result, if the weights align with the complex divisions (i.e., two phase difference align with each other), then the power will reach the maximum; otherwise, they cancel out each other and lead to a lower power. In this way, we can generate a spatial spectrum by traversing all potential directions. The incoming signal should peak at the direction (aka AoA), leading to the maximum power in the spatial spectrum. We refer the reader to [68] for additional details. Finally, the transmitter can be located at the intersection of two lines along the two AoAs estimated using two antenna arrays via triangulation. We present the relative power for a 1D array here for clarity. In the next section, we will extend to a 2D antenna array. Clearly, if the measured phase difference deviates from the theoretical difference because of phase noise or unknown offsets, then the spatial spectrum will peak in the wrong direction. We compute the relative power using the pseudo-phase. For clarity, we concentrate on the *complex division* of Eqn. 3.12 because the other components are irrelevant to the measured phase. We use ρ_k to denote the complex division for short:

$$\begin{aligned}
\rho_k &= \frac{S_{G_k}}{S_{G_0}} = \frac{\Delta S_{G_k}}{\Delta S_{G_0}} = \frac{\breve{A}_k e^{\mathbf{J}\tilde{\theta}_{T\to G_k}}}{\breve{A}_0 e^{\mathbf{J}\tilde{\theta}_{T\to G_0}}} = \frac{\breve{A}_k}{\breve{A}_0} e^{\mathbf{J}(\tilde{\theta}_{T\to G_k} - \tilde{\theta}_{T\to G_0})} \\
&= \frac{\breve{A}_k}{\breve{A}_0} e^{\mathbf{J}\left[(\theta_{T\to G_k} + \cancel{\phi_R + \phi_T} + \cancel{\phi_{G_k}}) - (\theta_{T\to G_0} + \cancel{\phi_R + \phi_T} + \cancel{\phi_{G_0}})\right]} \\
&= \frac{\breve{A}_k}{\breve{A}_0} e^{\mathbf{J}\Delta\theta_{T\to G_k}}
\end{aligned} \tag{3.14}$$

All antennas in the gateway share a single clock, so the phase offset $\phi_{G_k} = \phi_{G_0}$. The common terms of ϕ_R and ϕ_T are totally cancelled out when the backscatter signal is much stronger than the noise. Compared with Eqn. 3.12, the remainder is exactly as identical as using the true phase. That is, the pseudo-phase is equivalent to the true phase in computing the spatial spectrum. Therefore, the naive phase estimator

(NPE) first determines the two geometric centers in the IQ constellation and then computes the angle of the vector connecting the two centers as the pseudo-phase. NPE has been widely adopted by commercial backscatter systems, such as ImpinJ RFID reader series [69]. The phase estimation approach can serve for many AoA estimation algorithms like RF-IDraw [70], MUSIC [71], ESPIRIT [72] and Tagoram [73], which are based on the complex division or the phase difference as well.

3.2 Wireless Localization

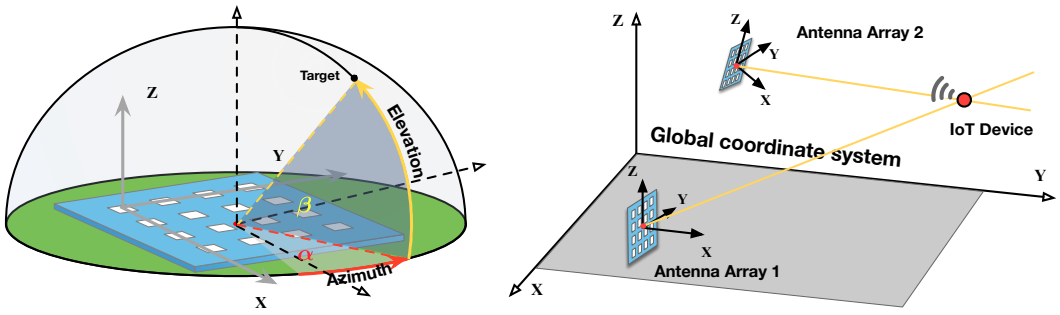
This section introduces the background knowledge about the spatial spectrum and the triangulation-based localization.

3.2.1 Spatial Spectrum

In this section, we will introduce the case of the 2D antenna array. Suppose the antenna array is equipped with $K \times K$ elements uniformly where two adjacent elements are spaced with L. As shown in Fig. 3.4(a), we can set up a local *cartesian coordinate system* for an array by choosing its bottom-left corner as the origin. Then, the coordinate of i^{th} element $E_{i,j}$ is given by:

$$E_{i,j} = (x_i, y_i) = (iL, jL)$$

where $i, j = 0, \ldots, K-1$. Our goal is to compute the direction of the transmitter, which can be represented using a local *horizontal coordinate system* over the array's X-Y plane. As shown in Fig. 3.4(a), an arbitrary direction over X-Y plane can be represented by two angles, azimuthal angle (denoted by α) and elevation angle (denoted by β), where $\alpha \in [0, 360^\circ)$ and $\beta \in [0, 90^\circ]$. The both local coordinate systems are set up relative to the array plane, but target at representing positions



(a) Local Coordinate Systems (b) Triangulation

Fig. 3.4: Illustration of AoA signature.

and angles respectively.

When the backscatter tag is relatively far from the antenna array, the incident signals received by elements are approximated to propagate in parallel, i.e., incident signals arrives at all elements from a same direction. Suppose the RF signal travels along the angle (α, β), the difference of the distances between the target device and $E_{i,j}$ and $E_{0,0}$ is computed as follow:

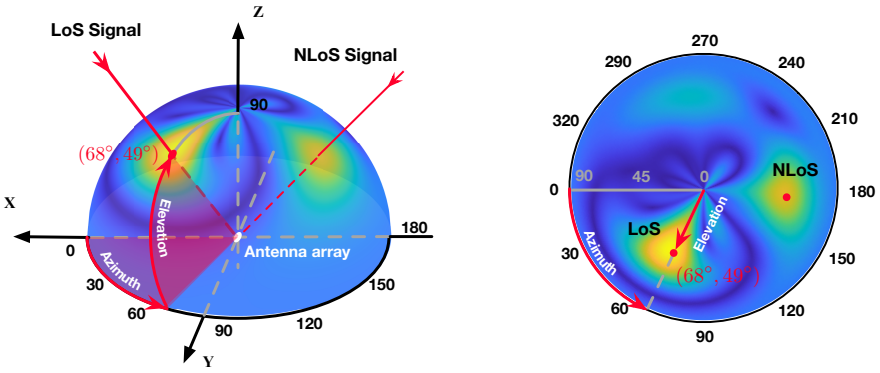
$$\Delta d_{i,j} \approx x_i \cos\alpha \cos\beta + y_j \sin\alpha \cos\beta \tag{3.15}$$

Let $\theta_{i,j}$ denote the phase value of the RF signal received by $E_{i,j}$. The phase difference between the received signals at the two elements, $\Delta\theta_{i,j} = \theta_{i,j} - \theta_{0,0}$, relates to the difference in their distances $\Delta d_{i,j}$ as follows:

$$\begin{aligned} \Delta\theta_{i,j} = \theta_{i,j} - \theta_{0,0} &= -2\pi\Delta d_{i,j}/\lambda \bmod 2\pi \\ &= -2\pi(x_i \cos\alpha \cos\beta + y_j \sin\alpha \cos\beta)/\lambda \bmod 2\pi \end{aligned} \tag{3.16}$$

Relative Power. Choosing the element $E_{0,0}$ as a reference, we can compute the following relative power of projecting the received signal into the direction of (α, β):

$$P(\alpha, \beta) = \frac{1}{K^2} \left| \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} w_{i,j}(\alpha, \beta) \cdot e^{\mathrm{J}\Delta\tilde{\theta}_{\mathrm{i,j}}} \right| \tag{3.17}$$



(a) 3D Spatial Spectrum (b) 2D Spatial Spectrum

Fig. 3.5: Illustration of spatial spectrum

where $w_{i,j}(\alpha, \beta) = e^{-\mathrm{J}\Delta\theta_{\mathrm{i,j}}}$ is the complex weight for steering a beam to a certain angle of (α, β). In the above, $\Delta\widetilde{\theta}_{i,j}$ is the true phase difference computed by using the received signals at $E_{i,j}$ and $E_{0,0}$, whereas $\Delta\theta$ is their theoretical phase difference. The sum aggregates the relative power across the K^2 pairs of elements, i.e., $(E_{0,0}, E_{0,0}), (E_{0,1}, E_{0,0}), \cdots (E_{3,2}, E_{0,0}), (E_{3,3}, E_{0,0})$. When $\Delta\widetilde{\theta}_{i,j}$ aligns with $\Delta\theta_{i,j}$ (i.e., the signal does come from the direction of (α, β)), the normalized relative power $P(\alpha, \beta)$ should achieve the maximum. For clarity, we use ω to denote the tuple of the two angles related to a direction, i.e., $\omega = (\alpha, \beta)$. The relative power at the direction ω is rewritten in the form of the vector as follows:

$$P(\omega) = \Big[w_{0,0}(\omega), w_{0,1}(\omega), \cdots, w_{K-1,K-1}(\omega)\Big] \Big[e^{\mathrm{J}\Delta\tilde{\theta}_{0,0}}, e^{\mathrm{J}\Delta\tilde{\theta}_{0,1}}, \cdots, e^{\mathrm{J}\Delta\tilde{\theta}_{\mathrm{K-1,K-1}}}\Big]^T \tag{3.18}$$

where T denotes the transpose.

Next, a hotmap can be generated to show the relative power at N possible directions that the received RF signal might come from. We call such a hotmap *spatial spectrum*,

which is formalized as follows:

$$\begin{bmatrix} P(\omega_1) \\ P(\omega_2) \\ \vdots \\ P(\omega_N) \end{bmatrix} = \begin{bmatrix} w_{0,0}(\omega_1), w_{0,1}(\omega_1), \cdots, w_{K-1,K-1}(\omega_1) \\ w_{0,0}(\omega_2), w_{0,1}(\omega_2), \cdots, w_{K-1,K-1}(\omega_2) \\ \vdots \\ w_{0,0}(\omega_N), w_{0,1}(\omega_N), \cdots, w_{K-1,K-1}(\omega_N) \end{bmatrix} \begin{bmatrix} e^{\mathrm{J}\Delta\tilde{\theta}_{0,0}} \\ e^{\mathrm{J}\Delta\tilde{\theta}_{0,1}} \\ \vdots \\ e^{\mathrm{J}\Delta\tilde{\theta}_{\mathrm{K-1,K-1}}} \end{bmatrix} + \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \tag{3.19}$$

or

$$\mathbf{\Omega} = \mathbf{WS} + \mathbf{Z} \tag{3.20}$$

where $\mathbf{\Omega}$, $\mathbf{W}$, $\mathbf{S}$, and $\mathbf{Z}$ denote the spatial spectrum, the weight matrix, the received signals, and the noise, respectively. N is a custom parameter depending on the angle resolution. $N = 360 \times 90$ if one degree resolution is accepted. Fig. 6.2 shows an example of the spatial spectrum. Particularly, Fig. 3.5(a) shows the spatial spectrum in 3D where all directions are uniformly distributed; Fig. 3.5(a) shows the 2D spectrum by projecting the 3D onto the X-Y plane, in which the radial distance represents $\cos(\beta)$ so the elevation angle distributes non-uniformly. The spatial spectrum should peak at the direction that the RF signal truly comes from. Formally, the direction of the device is computed by solving the optimization problem as below:

$$\omega^* = \underset{\omega}{\operatorname{argmax}}\ \mathbf{\Omega} = \underset{\omega}{\operatorname{argmax}}\ (\mathbf{WS} + \mathbf{Z}) \tag{3.21}$$

However, due to the presence of multipath effect, the RF signal might arrive from multiple directions. Consequently, there exist multiple peaks in the spectrum, lead to ambiguity. It is worth noting that the community proposed many different types of spatial spectrums by using different weight matrices, such as Bartlett [74], MVDR [75], MUSIC [71], and Tagoram [73]. There, the Bartlett is adopted and others are discussed in the evaluation.

3.2.2 Triangulation

At least two antenna arrays are deployed in the target space to compute the location of the device. Specifically, we can compute a direction using a single antenna array. The device is located at the intersection of two directions or the centroid of the intersected area formed by multiple directions. This localization approach is called *triangulation*. As aforementioned, the direction of the device is estimated relative to the local coordinate system (LCS) of an array. Thus, it requires an extra step to convert all directions computed in local coordinate systems to a global coordinate system (GCS) before taking the triangulation. Formally, suppose the estimated direction $\omega^* = (\alpha^*, \beta^*)$ by using an antenna array, a straight line denoted by l in the LCS relative to the array can be constructed as follows:

$$l(\vec{a}(\omega^*), O) \tag{3.22}$$

where $\vec{a}$ and O are the directional vector of the line and the origin of the LCS that it passes through, namely,

$$\vec{a}(\omega^*) = [1, \tan(\alpha^*), \tan(\beta^*)/\cos\alpha^*]^T \textbf{ and } O = [0, 0, 0]^T \tag{3.23}$$

To compute the intersection, we firstly convert the above line from the LCS to the GCS using a rotation matrix $\mathbf{R} \in \mathbb{R}^3$ and the coordinate of the array $\mathbf{O}$ in GCS. Then, the straight line in the GCS is rewritten as follows:

$$l(\mathbf{R} \cdot \vec{a}(\omega^*), \mathbf{O}) \tag{3.24}$$

Suppose the directions are estimated by two antenna arrays in different positions. Then the location of the device is to solve the following optimization problem:

$$p^* = \underset{p}{\operatorname{argmin}} \sum_{g=1}^{G} \mathbf{d}(p, l(\mathbf{R}_g \cdot \vec{a}(\omega_g^*), \mathbf{O_g})) \tag{3.25}$$

where $\mathbf{d}(\cdot)$ is the Euclidean distance between a point and a line, and G is the number of antenna arrays. Substituting Eqn. 3.21 into Eqn. 3.25, we actually aim to solve

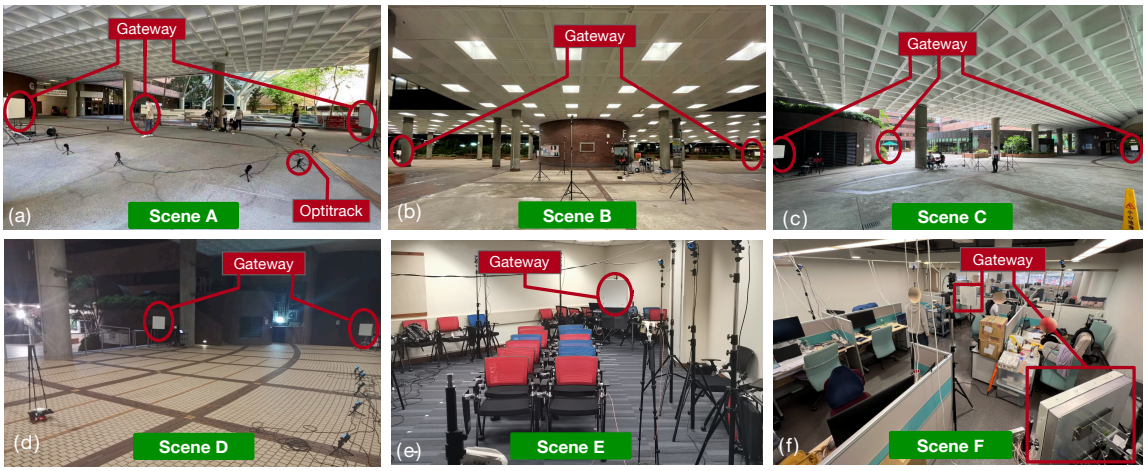


Fig. 3.6: Illustration of RFID example scenes. (a)-(d) shows the semi-indoor environment, which is large-sized and semi-closed halls. (e)-(f) show the full-indoor environment.

the following joint optimization problem:

$$p^* = \underset{p}{\operatorname{argmin}} \sum_{g=1}^{G} \mathbf{d}\left(p, l(\mathbf{R}_g \vec{a}(\underset{\omega}{\operatorname{argmax}}\,(\mathbf{W_g S_g} + \mathbf{Z_g})), \mathbf{O}_g)\right) \tag{3.26}$$

3.3 Ray Dataset

The performance of neural networks highly relies on the quantity and quantity of training datasets. Current datasets have been hindered by various limitations in terms of data scale, scene diversity, label precision, and coverage. To address these limitations, we build a multi-technology, cross-scene, million-scale 3D localization database, named *Ray*, after a three-year effort. A summary of the gathered data is displayed in Table 3.1. We collect data from 17 scenes with 50 different scenarios (i.e., settings). The database contains a total of 1,617,142 records from RFID tags (80.6%), Wi-Fi devices (7.2%), and BLE beacons (12.2%).

(1) RFID: Our design incorporates a dual-channel station with an array composed of 4×4 antennas. We use a USRP X310 SDR from NI [76] to handle the baseband signal processing of a station. Each X310 is equipped with two TwinRX daughterboards. One RX channel is linked to the antenna array, while the other is connected to a

Table 3.1: Summary of Ray Dataset. RSS represents the average signal strength; Total. indicates the total number of samples; Den. provides the sample count per cubic meter; Sta. denotes the count of base stations; Dist. reflects the average distance between sampled points and the base stations; Temp. is the ambient temperature.

Type. (#)	Sc. (#)	St. (#)	RSS (dBm)	Total (#)	Den. (p/m^3)	Sta. (#)	Dist. (m)	Temp. (°C)
RFID	A	S1	−62.5	84,392	3,843	3	5	31.2
		S2	−66.4	57,311	4,689	3	10	30.3
		S3	−66.7	55,527	5,274	3	15	29.9
		S4	−69.4	54,518	3,787	3	20	29.4
		S5	−71.0	50,302	4,336	3	25	27.2
		S6	−75.0	51,241	5,866	3	30	27.4
		S7	−77.4	51,289	5,871	3	35	27.7
		S8	−78.8	74,521	7,834	3	40	28.1
		S9	−79.3	61,909	4,236	3	45	28.3
		S10	−79.1	76,475	5,224	3	50	29
		S11	−88.6	50,186	10,490	3	55	28.7
	B	S12	−71.8	23,028	539	3	25	30.1
		S13	−76.9	21,357	702	3	35	30.4
		S14	−78.1	38,303	1,382	3	40	30.9
	C	S15	−68.3	18,726	6,080	3	20	33.1
	D	S16	−68.9	77,538	4,345	3	13	29.2
		S17	−67.0	40,571	2,546	3	13	28.8
	E	S18	−66.2	160,494	38,213	3	10	18.4
		S19	−65.3	78,635	27,924	2	10	24.9
		S20	−63.7	30,103	10,907	2	10	25.1
		S21	−64.9	26,916	8,901	2	10	27.6
		S22	−65.4	32,042	5,057	2	10	24.8
	F	S23	−65.1	48,467	22,627	3	7	25.8
	G	S24	−61.4	10,521	4,912	3	5	27.5
		S25	−60.2	5,291	938	3	5	30.1
		S26	−61.9	6,723	2,394	3	5	27.3
		S27	−61.9	8,413	1,829	3	5	28.2
		S28	−63.7	8,911	1,600	3	5	27.9
WiFi	H	S29	−58.6	11,288	123	4	7	N/A
		S30	−58.6	14,543	164	4	7	N/A
		S31	−60.0	10,579	106	4	7	N/A
		S32	−60.1	8,287	76	4	7	N/A
		S33	−60.1	5,233	58	4	7	N/A
	I	S34	−48.2	25,976	701	3	4	N/A
		S35	−48.2	23,677	656	3	4	N/A
		S36	−48.3	16,286	497	3	4	N/A
BLE	J	S37	−72.5	8,030	138	4	5	26.4
		S38	−84.5	11,123	199	4	5	26.4
		S39	−89.2	8,967	309	4	5	27.1
	K	S40	−62.4	8,419	1,011	4	5	29.3
		S41	−61.1	8,959	1,134	4	5	28.8
	L	S42	−60.4	6,386	823	4	5	29.9
		S43	−61.7	8,375	607	4	5	28
	M	S44	−60.9	11,235	1,489	4	5	18.5
		S45	−64.9	8,647	1,259	4	5	18.3
	N	S46	−71.0	15,412	1,381	4	20	17.8
		S47	−75.1	19,048	1,311	4	20	17.8
	O	S48	−78.3	32,039	1,059	4	25	17.6
	P	S49	−78.7	31,766	1,251	4	25	17.3
	Q	S50	−69.2	18,975	1073	4	10	16.8

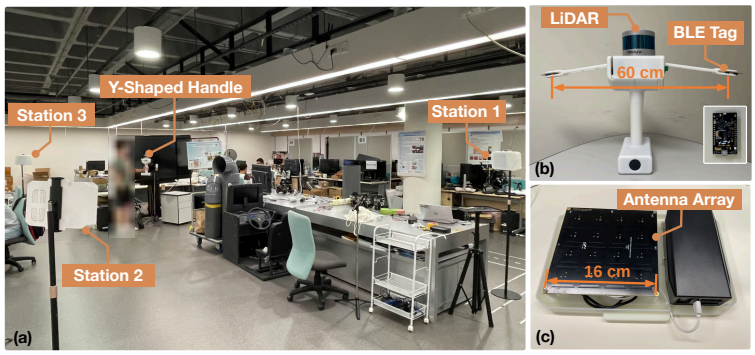


Fig. 3.7: The deployment of BLE localization platform.

circularly polarized patch antenna, serving as the side channel. The RF source is fixed approximately 5 m from the tag; it is used to power the tag and prompt it to continuously send `RN16` packets. The advantage of such a bistatic design is that three antenna arrays can be placed at a considerable distance from the tag (e.g., 50 m). Notably, the focus here is not on ultra-long-range communication of RFID systems. We recommend readers refer to a previous study [77] to further understand how the tag-to-receiver range can exceed 130 m at 1 kbps and 30 dBm transmitting power. Using the experimental platform, we collect data from 7 different types of scenes with 28 distinct settings.

(2) Wi-Fi: We integrate the dataset released by DLoc [9] into the dataset, which aims to position a Wi-Fi receiver using CSI. It covers two scenes and eight settings. Each station is equipped with a four-antenna linear array. For more hardware configuration details, readers are directed to [9]. Unlike narrowband-enabled RFID or BLE, Wi-Fi adopts the wide band, so the CSI contains 64 subcarrier information (i.e., phase and RSSI). To utilize this extra information, we compute a spatial spectrum using Eqn. 3.20 for each subcarrier and simply add the 64 spatial spectra together as the final one.

(3) Bluetooth: The deployment of the BLE localization platform is shown in Fig. 3.7. The platform, developed by Silicon Labs, operates on 2.4 GHz in accordance with BLE V4.2. The platform utilizes a 4×4 patch antenna array (model:

BRD4191A) [78] and has a size of 16×16 cm^2, with each individual patch antenna being 2.42×2.42 cm^2. The average errors in azimuth and elevation are $\pm 3.2^\circ$ and $\pm 4^\circ$, respectively. Each antenna comes with dual-input ports for receiving both horizontally and vertically polarized signals. Built with a JL-2800 laminate type, the antenna array board underwent extensive optimization and testing using IT-180A and IS400 laminates. We made phase measurements using Gecko SDK 4.1 and the RTL library. The Bluetooth tags (model: RD4184A) are also from Silicon Labs.

Chapter 4

Neural Radio-Frequency Radiance Field

4.1 Introduction

Recently, researchers at Google introduced the concept of Neural Radiance Fields (NeRF) to tackle the issue of light ray tracing [32]. Marked as a significant advancement in computer vision, NeRF has showcased remarkable achievements in view synthesis [32, 79, 80], 3D model rendering [36, 81], and creating immersive street views [34, 37]. An array of demonstrations are accessible at [82]. The core principle behind NeRF involves capturing several images from varying angles within a scene, which are then used as input to train a Multilayer Perceptron (MLP) — a particular type of feedforward artificial neural network — to approximate the optical radiance field. NeRF interprets each image pixel as an outcome of a single ray tracing operation, embodying the characteristics of the scene-dependent optical radiance field. Upon successful training with a handful of images, NeRF is capable of accurately forecasting the ray tracing outcome from any other angle, and can further construct a complete image from a specified observation direction.

Building on the notion that light is a form of electromagnetic wave, we put forward the Neural Radio-Frequency Radiance Fields (NeRF^2), extending the domain of neural radiance fields from optics to electromagnetism. In a manner akin to its predecessor, NeRF^2 encapsulates scenes as neural radiance fields by refining an inherent continuous volumetric scene function, leveraging a sparse collection of input signal readings. Specifically, NeRF^2 is adept at foretelling the nature and reception of an RF signal when the transmitter (TX) and/or the receiver (RX) is positioned at a discerned location. To grasp the prowess of NeRF^2 intuitively, we exhibit an illustration in Fig. 4.1. Four distinct algorithms are deployed to formulate (or forecast) the spatial spectrums (i.e., multipath profile) showcasing the manner in which the RX intercepts the signal from varied directions, contingent on the four distinct placements of the TX. Clearly, the prognostication rendered by NeRF^2 closely mirrors the ground truth, which is deduced from the authentic signals intercepted by the antenna array. Additional exemplifications can be viewed in our demonstration video at `https://xpengzhao.github.io/NeRF2`.

As a physical-layer neural network, NeRF^2 is poised to enhance the efficacy of a broad spectrum of pivotal RF applications including indoor localization, channel estimation, wireless power transmission, 5G base station setup, and wireless sensing, among others. Catering to diverse application-layer requisites, we introduce NeRF^2-enabled *turbo learning* (i.e., augmented learning), a methodology that leverages the physical attributes of NeRF^2 to churn out a plethora of synthetic datasets aligned with the physical model. These synthetic datasets are amalgamated with the authentic datasets to amplify the training regimen of application-layer artificial neural networks (ANNs). Turbo-learning not only alleviates the necessity for extensive data collection for ANNs but also ensures a pinnacle of learning accuracy, paving the way for a more refined and effective application-layer performance.

Transitioning NeRF to the RF domain entails navigating a series of hurdles. Firstly, RF signals, especially within the UHF or microwave spectrum (e.g., 800MHz, 2.4GHz,

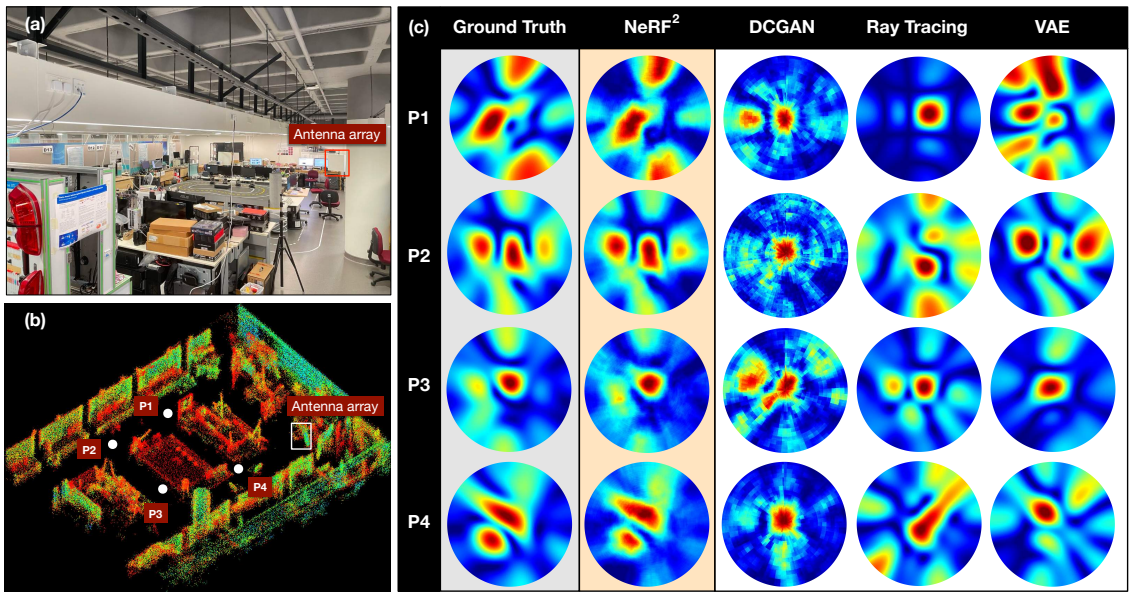


Fig. 4.1: Spatial Spectrum Synthesis. The spatial spectrums, alternatively termed as the multipath profile, illustrate the strength of the RF signal emanating from a specific direction, defined by azimuthal and elevation angles. This concept is formally articulated in Eqn. 3.20. (a) depicts the scene where the TX could be positioned anywhere while the RX, outfitted with a 4×4 antenna array, remains stationary at a corner; (b) displays the point cloud generated by LiDAR, only utilized for the traditional ray-tracing algorithm; (c) contrasts the synthesized spectrums created by diverse algorithms when the TX is situated at four distinct locations. The ground truth is ascertained by utilizing the antenna array.

or 6GHz), are more susceptible to reflection, diffraction, and scattering due to their significantly lower frequencies compared to visible light. Secondly, the optical NeRF predominantly considers the amplitude (i.e., light strength) while overlooking the phase of light, given its repetitive nature every 600–800 nm of propagation. However, in the realm of cm- or mm-wavelength RF signals, phase cannot be sidestepped due to its pivotal role in either constructive or destructive superimposition stemming from multipath effects. Thirdly, while visible light measurements are captured using million-pixel cameras, RF receivers (RX) typically come with either a singular antenna or a modest antenna array, constrained by size limitations (i.e., antenna size correlates with wavelength). Finally, massive MIMO systems can include dozens or even hundreds of subcarriers, each displaying distinct interaction behaviors with the environment. Consequently, a single model may not be suitable for all frequencies.

To mitigate these challenges, we first revise the physical tracing model to better align with the characteristics of RF signals. Subsequently, we integrate both phase and amplitude, implementing a complex-valued Multilayer Perceptron (MLP) for our system. We then propose two distinct training methodologies tailored for single-antenna and array-antenna receivers, establishing a structured approach to adapt NeRF to the complex requirements of the RF domain. Finally, we expand the mode to achieve the frequency-aware NeRF^2 by incorporating an RF-prism module that adapts to various subcarriers.

Summary of Results. We train the NeRF^2 models for each scene with the following results:

- We made a 4×4 antenna array as the RX to forecast the spatial spectrums (i.e., multipath profile) in the micro benchmark. The outcomes indicate that the median similarity between the spatial spectrums generated by NeRF^2 and the ground truth reaches up to 82%, markedly surpassing other synthetic algorithms.
- The frequency-aware NeRF^2 model demonstrated improvements in channel prediction and MU-MIMO performance, achieving an SNR improvement of 4 dB and an SINR improvement of 3.2 dB over the original NeRF^2. Additionally, our optimization techniques resulted in a computational speed-up of 3.5 times.
- In the context of AoA estimation, we gauged the advantages of NeRF^2 enabled turbo-learning. Our extensive experiments, wherein we gather RF signals at 530K locations across 14 scenes, further corroborate the substantial efficacy of turbo-learning. On the whole, the average AoA accuracy experiences a boost of 49.5% across all 14 scenes.
- We present a field study to demonstrate how the NeRF^2 benefits the BLE Localization. Pinpointing an RF device indoors is challenging, particularly when the line-of-sight propagation is blocked. Similar to the problem of ray tracing in graphics, localization accuracy can be improved greatly if the propagation of RF signals is deeply traced using NeRF^2. Our experiment results show that the NeRF^2 enabled

turbo-learning can reduce the median error by 50% and the standard variance by 40%.

Contribution. Our contributions are summarized as follows.

- We translate the NeRF from optics to the RF domain. Specifically, (1) We update the neural networks for complex-valued input parameters; (2) we replace the light propagation model with the Friis equation; (3) we invented the single-antenna and multiple-antenna-based electromagnetic ray-tracing approaches.

- We propose the NeRF^2 enabled turbo-learning. The benefit of turbo-learning is not only limited to the performance enhancement but also, more importantly, addresses the pain point of deep learning – significant reduction of the quantity of training dataset and the corresponding workload on collecting dataset.

- We introduce frequency-aware NeRF^2 as well as the optimization methods to address the frequency selection problem. The real-life trace based case study verfies the efficacy of NeRF^2.

- We conduct real-life field studies in terms of RFID, BLE, and 5G systems. The proposed turbo-learning is evaluated on indoor localization and FDD massive MIMO channel prediction.

4.2 NeRF^2 Design

In line with the conventional practice of NeRF, we uphold similar assumptions: (1) The receivers (e.g., 5G base stations, Bluetooth stations, and RFID readers) are stationed at predetermined locations, while the transmitters (e.g., smartphones, iBeacons, and RFID tags) are permitted to traverse within a confined area. (2) Significant obstructions in each scene (e.g., buildings, walls, and furniture) retain their positions

or structures. (3) The mobility of smaller entities may induce minor transient disruptions in the radiance field, yet such disruptions can be moderated through advanced filtering algorithms like the Kalman filter, hence their impact is not accounted for in this study.

4.2.1 Overview

At the core of NeRF^2 lie two integral components: voxel radiosity and the ray marching:

- **Voxel Radiosity**: For an accurate representation of the radiance field, we divide the designated area into a specified number of small 3D spaces, referred to as voxels. Each voxel encapsulates three attributes: positional attribute, attenuation attribute, and radiation attribute. A neural network consisting of two Multi-Layer Perceptrons (MLPs) is devised to clarify these three attributes.
- **Ray Marching**: After outlining the RF distribution, it is essential to trace signals from all potential directions to predict the signal received at the RX. This process iteratively steps through each voxel on a specific direction until the signals reach the RX or exit the scene.
- **Network Training**: By tracking the signals' paths, ray marching aids in accurately predicting RF signals at the RX, facilitating further training of the neural radiance network.

In the ensuing section, we delve into the nuances of the aforementioned components and unveil the training methodology.

4.2.2 Voxel Radiosity

Radiosity is a technique used in computer graphics to simulate the way light interacts with surfaces to generate realistic images. It calculates the diffuse reflections of light

that contribute to the illumination of surfaces in a scene, considering both direct lighting from light sources and indirect lighting, where light bounces off surfaces to illuminate other parts of the scene. Similarly, we break down a three-dimensional scene into small cube-shaped elements called voxels, each representing a discrete part of the space. In this model, each voxel carries information regarding its position, attenuation, and radiation attributes. The aim is to accurately represent how RF signals or light interact and propagate within a given voxel. To elucidate this principle, an exemplary voxel is illustrated in the magnified section of Fig. 1.1. Let the subscript x represent an arbitrary voxel within the scene. The voxel x located at position P_x intercepts the RF signal via two paths r_1 and r_2, transitioning into a new TX that re-dispatches the RF signal along path r_3 to the RX.

Voxel Attributes: In our framework, each voxel is characterized by three attributes: the position $P_x = (X, Y, Z)$, the attenuation $\delta(P_x) = \Delta a(P_x)e^{\mathbf{J}\Delta\theta(P_x)}$, and the re-transmitted RF signal $S(P_x)$ (aka radiation). The $\delta(P_x)$ is material-dependent, signifying that the amplitude diminishes by $\Delta a(P_x) = |\delta(P_x)|$ and the phase shifts by $\Delta\theta(P_x) = \angle\delta(P_x)$ as an RF signal traverses through the voxel. Acting as a new RF transmitter, the voxel at position P_x emits a new complex-valued signal S_x, i.e., $S(P_x) = a(P_x)e^{\mathbf{J}\theta(P_x)}$ where $\theta(P_x)$ and $a(P_x)$ denote the initial phase and the initial amplitude respectively. A voxel cannot be merely portrayed as an omnidirectional radiance source, but may disperse electromagnetic (EM) waves unevenly across angles. To accommodate this, we introduce an additional variable termed measuring direction $\omega = (\alpha, \beta)$ where α and β define the azimuthal and elevation angles.

Neural Radiance Network. A neural network is utilized to characterize the attributes within each voxel, forming a comprehensive depiction of the radiance field. Through this arrangement, the voxel radiosity approach furnishes a systematic means to simulate and analyze the behavior of RF signals within the delineated space. In

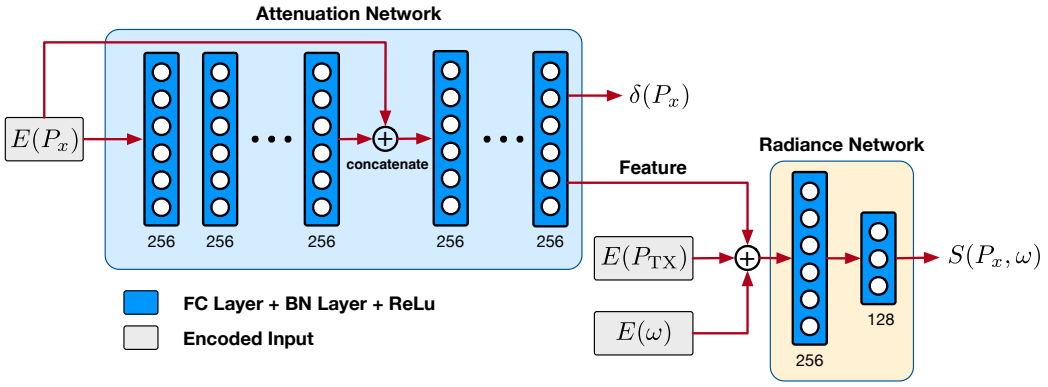


Fig. 4.2: Architecture of the neural network. NeRF^2 consists of two MLPs, the attenuation network, and the radiance network. The attenuation network can predict the attenuation δ of any voxel. Given the TX position and a measuring direction, the radiance network can predict the signal transmitting from an arbitrary voxel.

formal terms, the radiance field $\mathbf{F}$ is articulated as follows:

$$\mathbf{F}_\Theta : (P_{\text{TX}}, P_x, \omega) \rightarrow \Big(\delta(P_x), S(P_x, \omega)\Big) \tag{4.1}$$

where Θ symbolizes the learnable neural network weights and P_{TX} represents the position of the TX. The neural network yields two outputs. One is the attenuation $\delta(P_x)$ of the voxel at P_x, which is closely tied to the voxel's physical traits. The other is the RF signal $S(P_x, \omega)$ retransmitted from the voxel at P_x toward the direction ω when provided with the TX's position P_{TX}. Therefore, the neural network encapsulates both the scene and the RF distribution.

Contrary to the visual NeRF which posits that ambient light remains constant, we incorporate the position of the TX as an added input due to the mobility of our transmitters (e.g., smartphones or IoT devices). This way, we can compile a dataset for a scene by situating the TX at various and ample positions.

Network Architecture: In constructing the neural network, we employ two MLPs: the attenuation network and the radiance network, as depicted in Fig. 4.2. Given that the attenuation property is substantially associated with the materials of the voxel and is independent of incoming signals, we isolate the attenuation network to

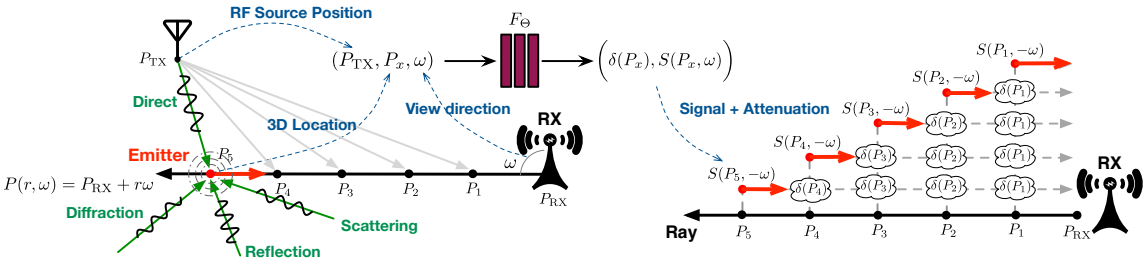


Fig. 4.3: Electromagnetic ray tracing. There are five voxels at $P_1 - P_5$ on the ray. Each voxel becomes a new transmitter that emits the signal along the ray to the RX. Their signals are attenuated by the other voxels between the new transmitters and the RX.

forecast the attenuation $\delta(P_x)$ based on the position P_x. This network comprises eight fully connected layers (with ReLU activations and 256 nodes per layer) and yields $\delta(P_x)$ along with a 256-dimensional feature vector. This vector is subsequently merged with the RX direction ω relative to P_x, and the TX's position P_{TX}. The fusion is relayed to the radiance network, entailing another two fully connected layers (with a ReLU activation and comprising 256 and 128 nodes), which then outputs the direction-dependent RF signal $S(P_x, \omega)$, that is retransmitted from the voxel along the direction ω.

The architecture mirrors that of the optical NeRF but diverges in two key areas. Initially, the visual NeRF presupposes a stationary location of TX (i.e., light source), while in our setup, the TX is mobile. Secondly, our networks are complex-valued, accounting for both magnitude and phase. In addition, the radiance field pertains solely to the scene, encompassing the obstacles and the position of TX, but it does not concern the position of RX. There might be apprehensions regarding handling the multiple reflections of the RF signal. The clever maneuver within NeRF^2 is the consideration of each voxel as a new transmitter, which "retransmits" a unified signal received from all feasible paths. This model streamlines the subsequent computations of ray tracing.

4.2.3 Electromagnetic Ray Tracing

To train the NeRF^2, a naive approach is to probe the RF signals at a vast number of RX positions. Evidently, this approach is unscalable in practice. The visual NeRF views each image of the scene as a result of ray marching [1], where each pixel reflects the intensity of the light propagated from a particular direction due to the pinhole model of cameras. Similarly, the signal received at the RX is a result of electromagnetic ray tracing, where the signal is a combination of signals transmitted from all possible directions. Next, we introduce how we trace the signal from a particular direction.

The propagation of an RF signal S from a transmitter (TX) to a receiver (RX) conforms to the Friis equation as follows:

$$R = H_{\mathrm{TX}\to\mathrm{RX}} S = a_{\mathrm{TX}\to\mathrm{RX}} e^{\mathbf{J}\theta_{\mathrm{TX}\to\mathrm{RX}}} S \tag{4.2}$$

where R is the received signal, $H_{\mathrm{TX}\to\mathrm{RX}}$ is the channel attenuation. Particularly, $a_{\mathrm{TX}\to\mathrm{RX}}$ and $\theta_{\mathrm{TX}\to\mathrm{RX}}$ are the amplitude degradation and the phase rotation caused by the distance from the TX to the RX. Mathematically, a direction ω related to the RX can be modeled as a ray, which starts from the RX and directs toward ω. The points on this ray are correspondingly described as follows:

$$P(r, \omega) = P_{\mathrm{RX}} + r\omega \tag{4.3}$$

where r is the radial distance from the RX to the point on the ray. Note that $P_{\mathrm{RX}} = P(0, \omega)$. The purpose of ray tracing is to accumulate the RF signals emitted from all voxels on this ray. Namely, the received signal at the RX from the direction

[1] Ray tracing and ray marching are two rendering techniques in computer graphics. Ray tracing computes the resulting color by tracing rays and accounting for object interactions, while ray marching estimates the color and opacity of the scene by evaluating a function along a ray.

ω can be expressed as:

$$R(\omega) = \int_0^D H_{P(r,\omega)\to P_{\mathrm{RX}}} S\Big(P(r,\omega), -\omega\Big)\,\mathrm{d}r \tag{4.4}$$

In the above equation, $S(P(r,\omega),-\omega)$ represents the signal transmitted from the voxel at $P(r,\omega)$ to the RX at P_{RX}. Its transmission direction is opposite to the ray's direction, so we take the negative of ω in the equation. D is the maximal distance across the scene. The above equation suggests that the final signal received by the RX from the direction ω is the accumulation of the RF signals transmitted from all voxels on the ray, i.e., from $P(0,\omega)$ to $P(D,\omega)$. The $H_{P(r,\omega)\to P_{\mathrm{RX}}}$ is the attenuation of the signal propagated from the point $P(r,\omega)$ to the RX. It is defined as follows:

$$\begin{aligned} H_{P(r,\omega)\to P_{\mathrm{RX}}} &= \prod_{\tilde{r}=0}^{r} \delta\big(P(\tilde{r},\omega)\big) \\ &= \left(\prod_{\tilde{r}=0}^{r} \Delta a_{P(\tilde{r},\omega)} e^{\mathbf{J}\Delta\theta_{P(\tilde{r},\omega)}}\right) \end{aligned} \tag{4.5}$$

The above equation means that the total attenuation equals the product of all attenuations caused by the voxels between the voxels at $P(r,\omega)$ and at $P(0,\omega)$, i.e., $0 \leqslant \tilde{r} \leqslant r$. To facilitate the calculation, we transform the above equation to an equivalent log-scale form as follows:

$$\begin{aligned} H_{P(r,\omega)\to P_{\mathrm{RX}}} &= \exp\left(\ln\left(\prod_{\tilde{r}=0}^{r}\delta\big(P(\tilde{r},\omega)\big)\right)\right) = \exp\left(\int_0^r \ln\Big(\delta\big(P(\tilde{r},\omega)\big)\Big)\,\mathrm{d}\tilde{r}\right) \\ &= \exp\left(\underbrace{\int_0^r \hat{\delta}\Big(P(\tilde{r},\omega)\Big)\,\mathrm{d}\tilde{r}}_{\text{Sum of attenuations}}\right) \end{aligned} \tag{4.6}$$

where $\hat{\delta}(\cdot)$ denotes the log-scale attenuation of $\delta(\cdot)$, which is defined as follows:

$$\hat{\delta}(P(\tilde{r},\omega)) = \ln \delta(P(\tilde{r},\omega)) \tag{4.7}$$

The log-scale form makes the product become a sum of all attenuations between two voxels, which greatly facilitates the calculation. Substituting Eqn. 4.6 into Eqn. 7.11, the signal coming from the direction ω is given by

$$R(\omega) = \int_0^D \underbrace{\exp\left(\int_0^r \hat{\delta}\big(P(\tilde{r},\omega)\big)\,\mathrm{d}\tilde{r}\right)}_{\text{Attenuation Network}} \overbrace{S(P(r,\omega),-\omega)}^{\text{Radiance Network}}\,\mathrm{d}r \tag{4.8}$$

where the terms engaged in the previous part are predicted by the attenuation network, and the terms engaged in the last part are predicted by the radiance network. Briefly, the result of ray tracing along a direction is to aggregate the signals retransmitted from the voxels on this ray, each of which is regarded as a new source. Meanwhile, each transmission from a voxel must be attenuated by other voxels between the current voxel and the RX. Suppose there are N voxels on the ray, the ray tracing will take $\mathcal{O}(N^2)$ aggregations.

To visually understand the ray tracing algorithm, we show an example in Fig. 7.5. Assuming the horizontal ray is from the RX to the left (i.e., $\omega = 180^\circ$). On the ray, there are five voxels at P_1, P_2, P_3, P_4, and P_5, all of which are considered as new transmitters regardless of how these voxels are lighted up. As a result, the signal received by RX along the opposite direction of the ray (i.e., $-\omega$) is a combination of the five signals retransmitted from these five voxels. Particularly, the signal S_5 retransmitted from the voxel at P_5 is attenuated by the voxels at P_4, P_3, P_2, and P_1 in sequence. The accumulated attenuation equals $(\hat{\delta}_{P_1} + \hat{\delta}_{P_2} + \hat{\delta}_{P_3} + \hat{\delta}_{P_4})$. Similarly, the signals retransmitted from the voxels at P_1, P_2, P_3, and P_4 are attenuated by 0, $\hat{\delta}_{P_1}$, $\hat{\delta}_{P_1} + \hat{\delta}_{P_2}$, and $\hat{\delta}_{P_1} + \hat{\delta}_{P_2} + \hat{\delta}_{P_3}$, respectively.

Summary. The NeRF^2 does not completely depend on the neural network but combines the physical model and the statistic model. Specifically, ray tracing takes a well-known physical model of signal propagation, meanwhile, deep learning offers a statistical model of the complicated interactions between the RF signal and the

surrounding obstacles.

4.2.4 Network Training

The previous describes the ray tracing algorithm, by which we can use the NeRF^2 to predict the signal received by the RX from a particular direction. Regarding which type of antenna is equipped at the RX, we introduce two types of training approaches.

Case I: Single-Antenna RX Model

We consider a simplified case where the RX is equipped with a single omnidirectional or a single directional antenna. Evidently, a single antenna has no discernibility in directions. Thus, the eventually received signal by the RX is a combination of the signals from all potential directions as follows:

$$\begin{aligned} R &= \int_{\Omega} \sqrt{G_{\mathrm{RX}}(\omega)} R(\omega)\, \mathrm{d}\omega \\ &= \int_{\Omega} \int_{0}^{D} \exp\left(\int_{0}^{r} \hat{\delta}(P(\tilde{r}, \omega))\, \mathrm{d}\tilde{r} \right) S(P(r, \omega), -\omega)\, \mathrm{d}\tilde{r} \end{aligned} \tag{4.9}$$

where $G_{\mathrm{RX}}(\omega)$ indicates the antenna directivity (i.e., the gain that the antenna provides in each direction), and Ω denotes the directions that the antenna can cover. Let R and $\widetilde{R}$ denote the predicted signal by NeRF^2 with the ray tracing and the true received signal, respectively. We then can use the following loss function to train NeRF^2:

$$\mathcal{L} = |R - \widetilde{R}|^2 \tag{4.10}$$

The loss function aims to reduce the gap between the true signal and the predicted one.

Case II: Multi-Antenna RX Model

Next, we consider the second case where the RX is equipped with a phased antenna array, which can form a very narrow beam and steer it to receive signals from a particular direction [83]. The RX can then discriminate the signal in directions. Suppose the antenna array is equipped with $K \times K$ elements uniformly. Choosing the element $A_{1,1}$ as a reference, we can compute the following relative power of projecting the received signal into the direction of $\omega = (\alpha, \beta)$:

$$\Psi(\omega) = \frac{1}{(K^2 - 1)} \left| \sum_{i=1}^{K} \sum_{j=1}^{K} w_{i,j}(\omega) \cdot e^{\mathrm{J}\Delta\tilde{\theta}_{\mathrm{i,j}}} \right| \tag{4.11}$$

where $w_{i,j}(\omega) = e^{\mathrm{J}-\Delta\theta_{\mathrm{i,j}}}$ is the complex weight for steering a beam to a certain angle of (α, β). In the above, $\Delta\widetilde{\theta}_{i,j}$ is the phase difference computed by using the received signals at $A_{i,j}$ and $A_{1,1}$, whereas $\Delta\theta$ is their theoretical phase difference [84]. The sum aggregates the relative power across the $(K^2 - 1)$ pairs of elements, i.e., $(A_{1,2}, A_{1,1}), (A_{1,3}, A_{1,1}), \cdots$. When $\Delta\widetilde{\theta}_{i,j}$ aligns with $\Delta\theta_{i,j}$, i.e., the signal comes from the direction of (α, β), the normalized relative power $\Psi(\alpha, \beta)$ should achieve the maximum. A heatmap can then be generated to show the relative power at N possible directions that the received RF signal might come from. We call such a 2D heatmap *spatial spectrum*, denoted by Ψ. The N is a custom parameter depending on the angle resolution. If the one-degree resolution is accepted, $N = 360 \times 90$, and the spatial spectrum is defined as follows:

$$\Psi = \begin{pmatrix} \Psi(0^\circ, 0^\circ) & \Psi(1^\circ, 0^\circ) & \cdots & \Psi(360^\circ, 0^\circ) \\ \Psi(0^\circ, 1^\circ) & \Psi(1^\circ, 1^\circ) & \cdots & \Psi(360^\circ, 1^\circ) \\ \vdots & \vdots & \vdots & \vdots \\ \Psi(0^\circ, 90^\circ) & \Psi(1^\circ, 90^\circ) & \cdots & \Psi(360^\circ, 90^\circ) \end{pmatrix} \tag{4.12}$$

Sometimes, the spatial spectrum is also called multipath profile [42] because it reflects how the signal comes from multiple directions. Fig. 3.5(a) shows the spatial

spectrum in 3D where all directions are uniformly distributed; Fig. 3.5(b) shows the 2D spectrum by projecting the 3D onto the X-Y plane, in which the radial distance represents $\cos(\beta)$ so the elevation angle distributes non-uniformly.

It is spontaneous for NeRF^2 to predict the power of the signal coming from a particular direction and generate a predicted spatial spectrum Ψ' as follows:

$$\Psi'(\omega) = |R(\omega)|^2 \tag{4.13}$$

The relative power is directly proportional to the true power computed above. Even though a constant offset may exist between them, it does not affect the training of the network by using the following loss function:

$$\mathcal{L} = \sum_{\omega \in \Omega} |\Psi(\omega) - \Psi'(\omega)|^2 \tag{4.14}$$

The training aims to reduce the difference in power of the signal received from all possible directions.

Summary

Previous work employing a supervised learning approach generally trained networks to predict signal distribution, where the signal of a target voxel was determined using signals collected from adjacent voxels. As a result, their accuracy was highly dependent on the density of the collected datasets. In contrast, we utilize a semi-supervised learning approach in which the signal at a single voxel involves contributions from almost all voxels due to the nature of ray marching. This approach ensures widespread voxel participation in the training process during each iteration. Consequently, it allows for the collection of signals at fewer positions while still effectively training the network to capture the characteristics of an absolute majority of voxels.

4.3 Frequency-Aware NeRF^2

The architecture of our system currently does not account for wideband RF signals, which can encompass thousands of subcarriers. For example, WiFi CSI provides 52 subcarriers, while the MIMO system developed in [85] even provides the "big CSI", which contains 1024 subcarriers with the ultra-wideband. The interaction between RF signals and physical obstacles is intricately dependent on the frequency of the signals. Frequency-specific characteristics, such as signal attenuation, phase shift, and reflection, are influenced by the material properties of the obstacles and the signal wavelength. Consequently, the lack of consideration for these wideband characteristics in NeRF^2 might limit the accuracy and applicability of the model in environments where wideband signal propagation plays a critical role. To address this, future enhancements of our system could include frequency-aware modeling techniques that dynamically adapt to the varying behaviors of RF signals across different frequencies to improve both the precision and robustness of the model in diverse operational scenarios.

4.3.1 Radio-Frequency Prism

To effectively manage the complexities of wideband RF signals, we have developed the RF prism module, a multi-channel MLP designed to decompose the global features extracted by the attenuation and radiance networks into distinct subcarrier components. Given that the majority of wide-band communication systems utilize Orthogonal Frequency-Division Multiplexing (OFDM), where the frequencies of subcarriers are independently modulated, the RF prism module is specifically tailored to address each subcarrier individually. This module includes two 256-dimensional layers, each equipped with M channels to match the number of subcarriers. This architectural choice ensures precise processing of each subcarrier's signal, maintaining the integrity of modulation characteristics across the frequency spectrum. Moreover,

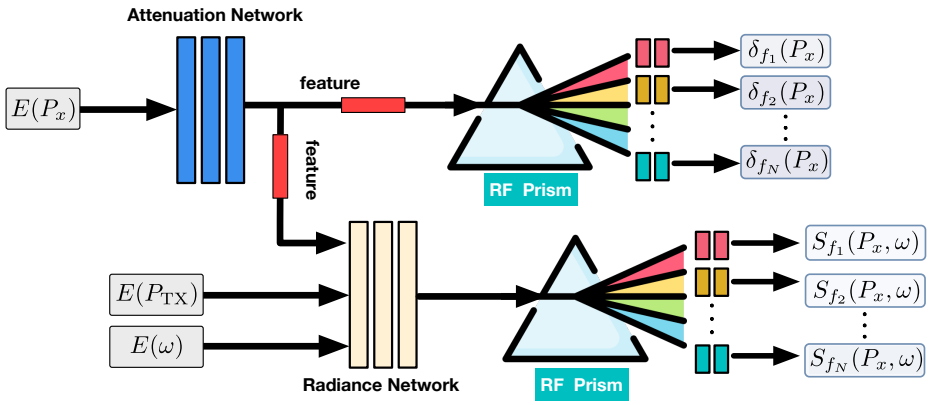


Fig. 4.4: Design of RF Prism. RF prism is appended to the attenuation and radiance networks for deassmbling the two characteriscs of a voxel into different subcarriers.

the RF prism module significantly boosts the system's capacity to accurately discern and adapt to the unique propagation dynamics and interaction effects at different frequencies, which are pivotal for enhancing performance in scenarios characterized by complex multi-path interference. This ability to isolate and process subcarrier-specific information allows for more sophisticated interference management and signal optimization, critical for maintaining high-quality communication in dense RF environments.

4.3.2 Network Training

Ray marching is crucial for training on each subcarrier, especially due to their orthogonality in the frequency domain. Suppose we aim to predict the channel frequency response $\tilde{H} = \{\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_m\}$ across M subcarriers, using the collected ground truth $H = \{h_1, h_2, \ldots, h_m\}$. We can employ the following loss function to optimize NeRF^2:

$$\mathcal{L} = \underbrace{|h_1 - \tilde{h}_1|^2}_{\text{1st channel}} + \underbrace{|h_2 - \tilde{h}_2|^2}_{\text{2nd channel}} + \cdots + \underbrace{|h_M - \tilde{h}_M|^2}_{\text{M-th channel}} \tag{4.15}$$

This loss function is designed to minimize the discrepancies at each of the M subcarriers. Although the loss function aggregates M errors together, it is important to note that the error back-propagation for each channel is independent. This independence

is critical as it allows for tailored adjustments in the model's parameters specific to each subcarrier, thus preserving the orthogonality and unique characteristics of each channel. This method enhances the overall accuracy and efficiency of the system by focusing on the specific error dynamics of each subcarrier, rather than applying a uniform correction across all channels. Additionally, this approach helps in addressing specific propagation challenges unique to each subcarrier, such as multipath effects and frequency-specific fading.

4.3.3 Optimization

As indicated in Eqn. 4.9, the computational complexity of each ray tracing operation is approximately $\mathcal{O}(\Omega \cdot D^2)$, where Ω represents the number of traced directions, and D denotes the number of voxels along a direction. This complexity escalates to $\mathcal{O}(M \cdot \Omega \cdot D^2)$ when accounting for M, the number of subcarriers. Although ray tracing for each path can be executed in parallel, the computational demands remain substantial. To address these challenges, we implement two primary optimization strategies:

(1) Accumulation Network: Considering that the majority of voxels are aerial and therefore minimally impact signal attenuation, traditional brute-force volumetric attenuation integrals are unnecessarily computationally intensive for such regions. To address this inefficiency, we adopt a learned approximation approach. Inspired by the visibility MLP described in [36], we introduce an "accumulation" MLP. This model outputs an estimated attenuation value for any given location along any input direction, as well as an accumulated attenuation estimate for the corresponding ray:

$$F_c : (P_{\text{RX}}, \omega, r) \rightarrow H_{P(r,w) \rightarrow P_{\text{RX}}} \tag{4.16}$$

This accumulation network is jointly trained with NeRF^2, allowing for a more streamlined integration into the overall system architecture. By implementing this approach,

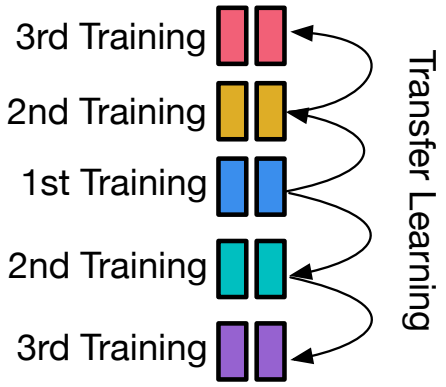


Fig. 4.5: Transfer Learning for RF Prism. The training of the RF prism begins with the middle channel. Subsequently, its parameters are transferred to adjacent channels for fine-tuning, thereby accelerating the overall training process.

we circumvent the need to calculate attenuation for each voxel individually. Consequently, this reduces the computational complexity significantly to $\mathcal{O}(M \cdot \Omega \cdot D)$, where each factor reflects the respective demands of subcarrier processing, directional tracing, and voxel interaction.

(2) Heuristic Training: Although subcarriers operate independently, the propagation characteristics between two adjacent subcarriers are quite similar due to their closely spaced center frequencies. Therefore, it is unnecessary to train the network separately for each subcarrier. Instead, we initially train the network specifically for the m^{th} subcarrier. Once the channel model for this subcarrier has converged, we transfer the learned parameters to the $(m+1)^{\text{th}}$ channel for fine-tuning. This strategy avoids the need to start parameter estimation from scratch for each subcarrier, thereby significantly reducing training time. After successfully fine-tuning the $(m+1)^{\text{th}}$ channel, we then propagate the well-tuned parameters to the $(m+2)^{\text{th}}$ channel, and so forth. Practically, we select the $\lfloor M/2 \rfloor^{\text{th}}$ subcarrier as the starting point and extend the training process bilaterally. The whole procedure is shown in Fig. 4.5. This method leverages the parameter similarity across adjacent subcarriers to expedite convergence and improve overall training efficiency. By applying this heuristic training approach, we enhance the scalability of our system and reduce computa-

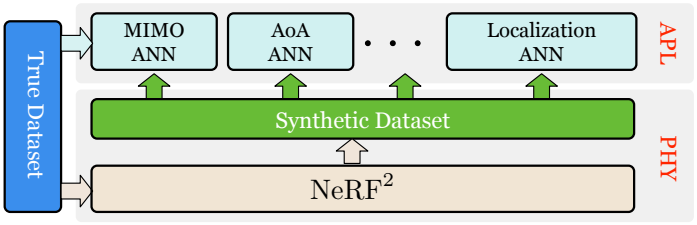


Fig. 4.6: Illustration of turbo-learning

tional overhead, making it feasible to handle larger arrays of subcarriers efficiently while maintaining high accuracy in channel estimation.

4.4 Turbo-Learning

As a physical-layer neural network, NeRF^2 describes the distribution of the radiance field. It cannot directly meet the application-layer demands, such as predicting the location of a receiver or beamforming parameters. Usually, extra neural networks are set up to address the specific application demand (e.g., MIMO ANN, AoA ANN, localization ANN, etc.). Instead, we employ NeRF^2 as a reinforcer to boost the performance of the application-layer ANNs. Fig. 4.6 illustrates this basic idea. First, we train NeRF^2 with the true training dataset. Second, NeRF^2 generates a vast number of synthetic dataset which meets the demand of the application-layer ANNs. Finally, we mix the true dataset and the synthetic dataset together to train the upper-layer ANNs. We call this training approach *turbo-learning*, i.e., applying more additional synthetic data to intensify the learning. Turbo-learning is also termed data augmentation in the field of data science. In the following sections, we will elaborate on turbo-learning case by case.

4.5 Implementation

Each scene trains a distinct NeRF^2, necessitating a dataset of RF signals or spatial spectrums captured within the scene, along with the corresponding TX and RX locations, and scene boundaries (i.e., Ω and D). The location-oriented parameters are gathered via a high-precision infrared positioning system named OptiTrack. During each iteration, the following optimizations are carried out:

(1) Positional Encoding: NeRF^2 takes in two 3D positions and one 2D direction as inputs. Mirroring the optical NeRF's practice of using encoded positions, the dimensions of the inputs are elevated to L using the ensuing encoding function:

$$E(x) = \left(\sin\left(2^0\pi x\right), \cos\left(2^0\pi x\right), \ldots, \sin\left(2^{L-1}\pi x\right), \cos\left(2^{L-1}\pi x\right)\right) \tag{4.17}$$

This function is separately applied to each of the three coordinate values in P_{TX} or P_x, and to the three components of the Cartesian direction unit vector ω. In our experiments, L is set to 10 for P_{TX} and P_x, and to 4 for ω.

(2) Voxel Size: A balance is sought in determining the voxel size. While fine-grained voxels can enhance resolution for NeRF^2 and ray tracing accuracy, the voxel count significantly affects computational complexity. In our trials, the voxel size is set to 1/8 of the wavelength.

(3) Network Configuration: In each dataset, a random 80% of samples are utilized for neural network training, leaving the remaining 20% for testing. A configuration akin to NeRF is employed. Specifically, the batch size is set at 4096. The Adam optimizer is used, with the learning rate starting at $3e{-}4$ and decreasing exponentially to $3e-5$. Other hyper-parameters are kept at default values (e.g., $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-7}$). Training the network for a single scene typically requires around 30k-50k iterations to converge on a single NVIDIA 4090 GPU, roughly amounting to 10 hours. Conversely, testing for a single sample can be done in about 0.2 seconds.

4.6 Microbenchmark

We start with a microbenchmark experiment to provide insights into the working of NeRF^2 in this section.

4.6.1 Experimental Setup

We deploy a USRP-based RX equipped with a 4×4 antenna array. The RX operates at 915 MHz and targets to receive the signal backscattered from a moving RFID tag. The RFID tag is activated by a nearby reader (i.e., 1 m away) and repeatedly transmits `RN16` replies. Figs. 4.1-(a) and (b) show the photo of the scene and the corresponding 3D model (composed of a point cloud) scanned by LiDAR. This is a demo room full of reflectors such as metal desks, shelves, tables, computers, and so on. The dataset is created by placing the tag at random positions. For each position, the antenna array generates a spatial spectrum using Eqn. 4.11, which is represented by 360×90 pixels from viewpoints sampled on the front hemisphere of the antenna array. We collected a total of 10 K data in this scene, where 8 K are used for training and 2 K for testing. We use the approach introduced in [57] to estimate the phase and amplitude of the received backscatter signals and employ Eqn. 4.14 as the loss function to train the neural radiance field.

4.6.2 Spectrum Synthesis

The goal of the original optical NeRF is to synthesize the photo of the scene taken from an arbitrary direction. Similarly, NeRF^2 possesses the ability to synthesize RF spatial spectrums when the TX is located at an arbitrary position. To visually understand such a purpose, we leverage NeRF^2 to synthesize the spatial spectrums that the antenna array receives. The synthesized spatial spectrum helps us intuitively verify whether the neural radiance field can successfully predict the signal propagations in

the scene. We compare NeRF^2 with the other four baseline schemes.

- **Ground truth**: The true spatial spectrums are computed by using the Eqn. 3.20 across the real signals received by the antenna array. The spectrums are desired to peak at the LOS direction. Unfortunately, Fig. 4.1-(c) (1st column) shows possible multiple peaks because of the multipath propagations in such a complex environment.
- **RayTracing**: We employ the RayTracking toolbox in Matlab [86] to generate the spatial spectrums. Particularly, this toolbox requires importing the 3D model of the scene (i.e., Fig. 4.1-(b)). Given the locations of TX, the toolbox can predict the RF signals received by the RX.
- **Deep Convolutional Generative Adversarial Network (DCGAN)**. DCGAN is one of the most popular GANs wherein two models (i.e., generator and discriminator) are trained simultaneously by an adversarial process. The generator model spawns "fake" images that look like the training images. The discriminator model determines whether an image is a real training image or a fake image from the generator. We view the predicted spatial spectrums as images and use DCGAN to learn and generate the spectrums with given TX's locations.
- **Variational Autoencoder (VAE)**. VAE is one of the famous generative models. It is used to resolve similar issues in wireless systems, such as liquid sensing [87] and channel estimation [26]. Adopting the similar architecture in FIRE [26], an encoder network learns the probability distribution of the training set in a lower dimensional latent space. Subsequently, the samples drawn from the decoder network are decoded to generate the data in accord with the learned distribution.

The results are shown in Fig. 4.1-(c), where the spatial spectrums are generated using the above schemes when the TX is located at four positions. Visually, the spatial spectrums generated by NeRF^2 are evidently more similar to the ground truth than other generative models. We further use a common criterion called *structural simi-*

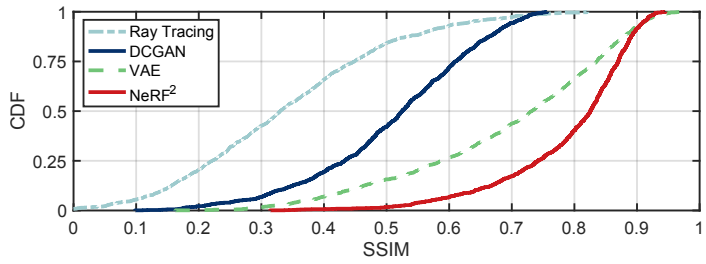


Fig. 4.7: SSIM Comparison

larity index measure (SSIM) to quantify the similarity of two images. The SSIM is employed to assess the prediction accuracy of different beam lobes in spatial spectra, which are critical for beamforming. Owing to the page limit, we omit the definition of SSIM but encourage the reader to refer to [88] for details. A higher SSIM indicates the two images are more similar. We randomly choose 100 positions to synthesize the spatial spectrums using the four algorithms. The CDF of the SSIM between those synthetic spatial spectrums and the ground truth is shown in Fig. 4.7. Particularly, the median SSIM of RayTracking, DCGAN, VAE, and NeRF^2 are 0.33, 0.52, 0.73, and 0.82, respectively, and their 99th percentiles are 0.71, 0.73, 0.91, and 0.92. The RayTracing underperforms because it is short of the material information, even though the geometric model of the scene is provided. DCGAN and VAE view spatial spectrums as a kind of signature related to the TX's location, so they do not really "understand" the rationale behind it. The outperformance of NeRF^2 is in the accurate model of radiance field in accordance with the underlying physical laws.

4.6.3 Performance of Turbo-Learning

To quantify the benefits of NeRF^2, we apply turbo-learning to the AoA estimation, which aims to determine the direction of the line-of-sight propagation. The AoA is desired to be achieved at the peak of the spatial spectrum. Unfortunately, owing to the multipath propagations and the destructive superposition of signals, the peak deviates substantially from the true LOS direction. To address this issue, angular

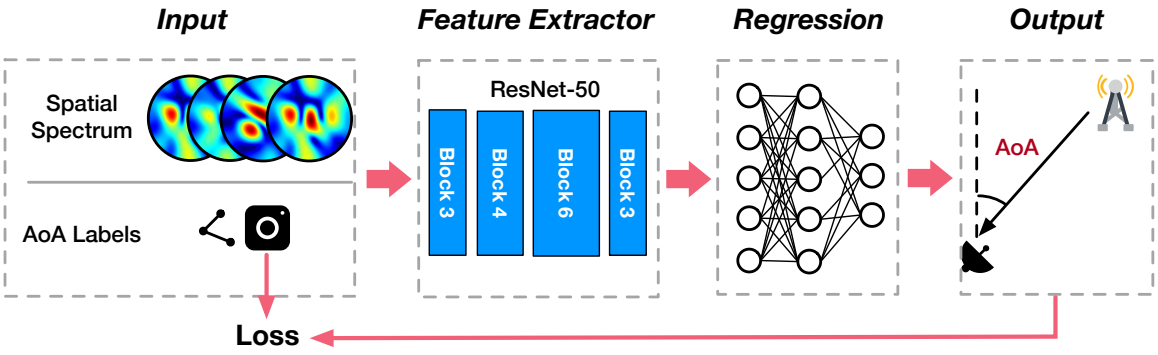
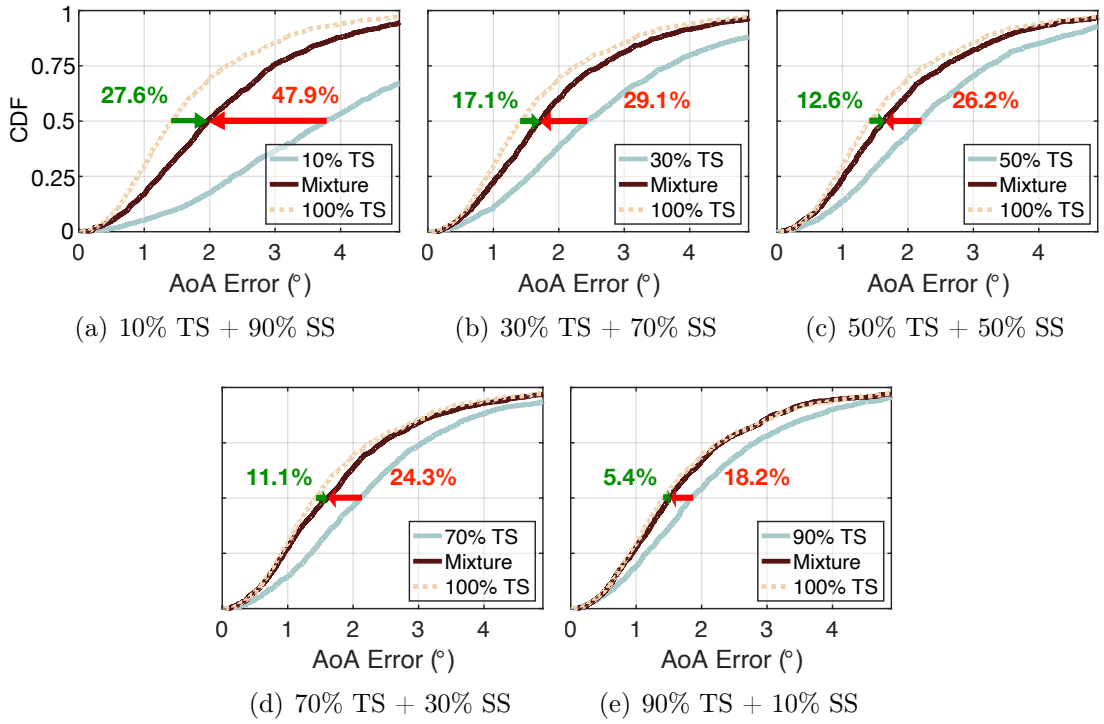


Fig. 4.8: Architecture of Angular Artificial Neural Network

artificial neural networks (AANNs) are resorted to identifying the AoAs [9,84,89,90]. Similar to the iArk [84], we set up an AANN based on the ResNet convolutional network [91], as shown in Fig. 4.8. The AANN accepts spatial spectrums in the image format and outputs the AoAs. In the AANN, a ResNet-50 network is adopted as the feature extractor, which is followed by a fully connected network for regression. We train the AANN using the following two approaches:

- **Naive Learning**. We use 10% of the true training dataset (TS, total 8 K) to train the AANN straightforwardly. In this approach, NeRF^2 is not involved.
- **Turbo-Learning**. We use the same 10% of the true dataset to train the NeRF^2. Then, we use the well-trained NeRF^2 to generate the rest 90% synthetic dataset (SS). Finally, the 10% true dataset and the 90% synthetic dataset (i.e., turbocharger) are mixed to train the AANN.

These two learning approaches fully use the same 10% of the true training dataset for the sake of fairness, i.e., *both hold the same amount of information from the true dataset.* We also use 100% training set to train the AANN as the baseline. The results are shown in Fig. 4.9(a). The median errors of naive learning and turbo-learning are 3.78° and 1.96°, respectively. The result of naive learning is enhanced by NeRF^2 with 47.9%. On the other hand, the accuracy of turbo-learning is extremely approaching the 1.42° error that the baseline achieves. This result demonstrates that the quality



(a) 10% TS + 90% SS (b) 30% TS + 70% SS (c) 50% TS + 50% SS

(d) 70% TS + 30% SS (e) 90% TS + 10% SS

Fig. 4.9: CDFs of AoA error. The ANN is trained by the naive-learning (in light blue) and the turbo-learning (in dark red), respectively. We quantify the benefits of NeRF^2 with different mixture percentages.

of the synthetic dataset generated by NeRF^2 is as good as the true dataset. Clearly, the quantity of true training dataset required by turbo-learning is far less than the baseline, but the accuracy remains at a comparably high level. This feature is useful because collecting a training dataset is an important but cumbersome and painful task for today's deep learning. The power of NeRF^2 is in the significant reduction of the quantity of true training set and the corresponding workload.

We also test other mixture ratios (30% TS+70% SS, 50% TS+50% SS, 70% TS+30% SS, and 90% TS+10% SS) using the same ways. The results are shown in Fig. 4.9(b)-(e). As desired, the error of turbo-learning is reduced from 1.96° to 1.72°, 1.62°, 1.59°, and 1.50°. Evidently, the accuracy is increased with an increasing quantity of true datasets. This is understandable because the accuracies of NeRF^2 and AANN improve as the amount of true information increases. On the other hand, turbo-

learning outperforms naive learning by 47.9% to 29.1%, 26.2%, 24.3%, and 18.2%. This demonstrates that more benefits can be gained when more percent of synthetic data is given. Even if only 10% synthetic data is fed, the median error can be reduced by 18.2% compared with naive learning. One may wonder why do not try the mixture of 0% TS plus 100% SS. It is impossible because the training of NeRF^2 must require a few numbers of true datasets. To achieve the trade-off between the accuracy and the quantity, it is advisable to take the mixture of 30% TS plus 70% SS in practice.

4.6.4 Large-scale Experiments

Regardless of NeRF or NeRF^2, both are scene-dependent because the radiance field is highly related to the scene layout. Whether the outperformance of turbo-learning can still be achieved in different scenes is unclear. Thus, we conduct large-scale experiments. We use the same antenna array to collect a huge dataset from 14 scenes (labeled A∼N). Fig. 3.6 shows six of them. We first collect the data in a large-area semi-indoor environment with the purpose of quantifying the impact of the distance. In such an environment, the antenna array is deployed in four scenes labeled A, B, C, and D, which are large-area and semi-closed halls, as shown in (a)-(d) of Fig. 3.6. In these scenes, the distance varies from 5 to 50 m. The distance is the mean value between the scene center and the antenna array. We then collect the data in the full-indoor environment. We deploy the platform in 10 rooms (i.e., Scenes E-N). Scene E is a warehouse, Scene F is a lab room, and Scenes G-I are classrooms. Scenes J and N are offices, Scene K is the hallway, Scene L is a meeting room, and Scene M is a lift lobby, as shown in (e)-(j) of Fig. 3.6. The coverage of the scene ranges from 5 to 32 m. Particularly, the gateways are deployed behind the wall in Scenes L, M, and N. Majority of these data are collected in the scenes full of people passing by and various reflectors.

Similarly, we choose the 80% dataset for training and the 20% dataset for testing in

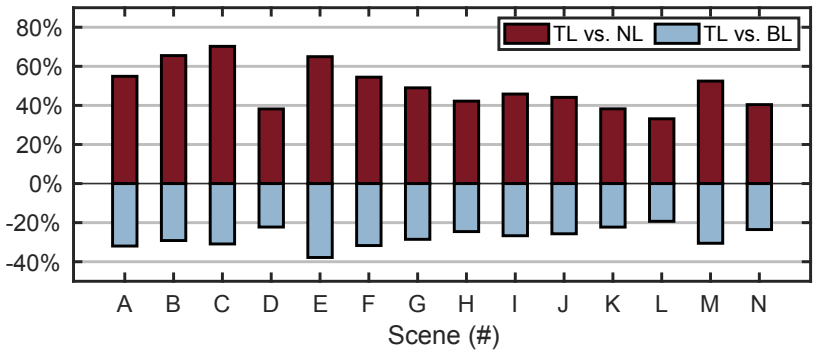


Fig. 4.10: AoA Accuracy vs. Scenes

each scene. Naive learning with the entire 80% true dataset is used for the baseline. Turbo-learning is conducted with 10% (out of the 80%) of the true training dataset plus 90% synthetic dataset. The AoA accuracy results are shown in Fig. 4.10. From the figure, we have the following two findings:

- Compared with naive learning (NL), the turbo-learning (TL) can offer 33%-70% improvement. The average is 49.5%. This shows that the performance enhancement by turbo-learning is a general phenomenon across scenes.
- Compared with the baseline (BL), the turbo-learning can hold -27.5% gap, where the minus sign denotes the "lower accuracy than". However, turbo-learning saves 90% workload for the dataset collection because only 10% training set is used.

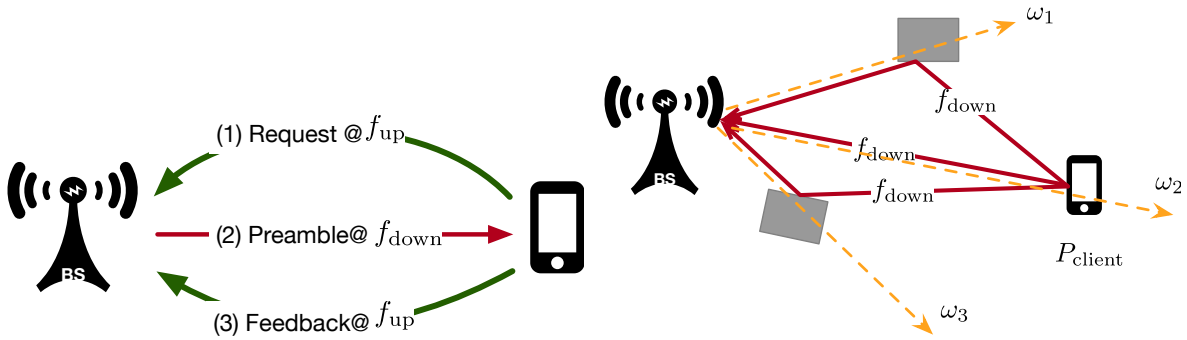
Our experiments reveal two key factors influencing turbo-learning performance: (1) Quantity of the dataset. NeRF^2 has the ability to reduce the requirement for data collection in application-layer NN tasks. However, if adequate data is provided, the application-layer NNs can train the model effectively, thus reducing the benefits of NeRF^2. (2) Quality of the dataset. The performance of NeRF^2 can also be affected by environmental interference, such as passing by people or other signals. Despite this, our results demonstrate that turbo-learning still improves the performance of application-layer NNs by over 30%. In summary, the outperformance of turbo-learning is mainly derived from the physical model provided by NeRF^2. Naive learning models the "signature (feature)-based" relationship between the AoA and the

spectrums, but NeRF^2 learns the physical rationale behind the relationship so it can provide more reasonable samples for the learning.

4.6.5 Evaluation on Frequency-Aware Model

Subsequently, we assess the capabilities of the Frequency-Aware NeRF^2 in FDD OFDM channel prediction tasks. To facilitate MIMO functionalities, base stations require knowledge of the downlink wireless channel from each antenna to all client devices (e.g., smartphones). In TDD systems, achieving this is straightforward due to reciprocity, since both uplink (client to base station) and downlink (base station to client) transmissions occur on the same frequency, allowing base stations to measure the uplink channel from client transmissions and infer the downlink channel. However, in 5G FDD systems, uplink and downlink transmissions occur on different frequencies (denoted by f_{up} and f_{down} respectively), making reciprocity inapplicable [26]. Instead, the client device gauges the wireless channel using additional preamble symbols sent by the base station and relays this data back as feedback to the base station. Fig. 4.11(a) illustrates the whole procedure. This feedback generates overhead that increases linearly with the number of antennas, devices, and the available bandwidth, becoming a significant hindrance for massive MIMO systems.

To simplify the matter at hand, an assumption is made where the client is situated at position P_{client} and possesses the ability to dispatch a packet at f_{down} to the base station. This packet is then leveraged to evaluate the uplink channel state at f_{down}, as illustrated in Fig. 4.11(b). This simplified scenario aligns well with the operational domain of frequency-aware NeRF^2, which encapsulates the scene utilizing an RF radiance field, thereby enabling precise signal prediction at the base station. Specifically, frequency-aware NeRF^2 can accurately predict the channel state by tracing the signal across all conceivable directions from the base station. It's noteworthy that the ray tracing operation doesn't factor in the client's position since it originates from the



(a) Channel Estimation (b) Simplified Case

Fig. 4.11: Channel estimation in 5G system. (a) shows the traditional channel estimation procedure in 5G where the downlink and uplink operate at f_{down} and f_{up}, respectively. (b) shows the simplified case by assuming the client located at a known position P_{client} can broadcast a preamble at f_{down} frequency.

base station, which is assumed to have a fixed position. The client's location is solely utilized to denote how a voxel is lit in the scene, with each voxel being treated as a new radiance source instead of the original client during the ray tracing.

A significant question arises concerning the acquisition of ground truth for training frequency-aware NeRF^2, especially given the client's inability to transmit packets at f_{down} in a real-world scenario, but can collect packets from the base station to estimate the downlink channel states. However, it's acknowledged that the channel showcases reciprocity when operating at identical frequencies. This implies that the downlink channel states, collected by the client at f_{down}, mirror those in the reverse direction due to the same frequency. Hence, the gathered downlink channel states can still be employed to train frequency-aware NeRF^2. The well-trained networks can predict either the uplink or the downlink channel state at f_{down}.

The subsequent query relates to the determination of a client's location. Previous studies underline a notable correlation between CSI and the physical environment, hinting at a unique relationship between the client device's position and its uplink signal's CSI, reminiscent of fingerprint-based localization. As a result, the neural

network is reformulated as:

$$\mathbf{F}_{\Theta} : (I_{\text{uplink}}, \omega, P_x) \rightarrow (\delta(P_x), S(P_x)) \tag{4.18}$$

whereI_{uplink} represents the position indicator (i.e., uplink CSI). The ensuing ray tracing is conducted at downlink frequencies, with the network training leveraging collected downlink CSI, thus disassociating our methodology from the path-sharing assumption. Lastly, 5G technology utilizes a wideband signal that is divided into numerous narrow-band signals operating at various frequencies. Each frequency may interact differently with a voxel, leading to unique attenuation and radiance characteristics. Consequently, the neural network model should be designed to account for these frequency-specific interactions. To address this, we employ the RF prism and heuristic training methods, which effectively predict the behavior of the wideband signal.

Experimental Setup. We opt for the publicly accessible Argos channel dataset [92] for our assessment. The Argos dataset embodies a real-world multi-user MIMO (MU-MIMO) scenario, encompassing 104 antennas at the base station and eight users, inclusive of both mobile and static trace collections. This dataset presents two distinct operational frequency versions, specifically 2.4 GHz and 5 GHz, necessitating the training of two separate NeRF^2 networks. The data compilation occurs on the ArgosV2 platform [93], employing omnidirectional monopole antennas spaced at half a wavelength at 2.4 GHz (i.e., 63.5 mm). The system boasts a bandwidth of up to 20 MHz, facilitated by 64 OFDM subcarriers. For CSI estimation, the system transmits 802.11 Long Training Symbols pilots at the onset of each frame across 52 subcarriers. Aligning with prior work [26], we allocate the initial 26 subcarriers for the uplink channel, reserving the remaining portion for the downlink channel, with guard bands segregating the two channels. Our objective centers on predicting the downlink CSI from the uplink CSI sans feedback. The data, harvested in a complex setting with abundant non-line-of-sight propagations, consists of a total of 100 K en-

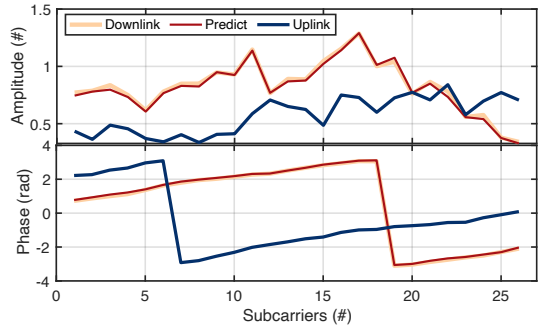


Fig. 4.12: Channel Amplitude & Phase

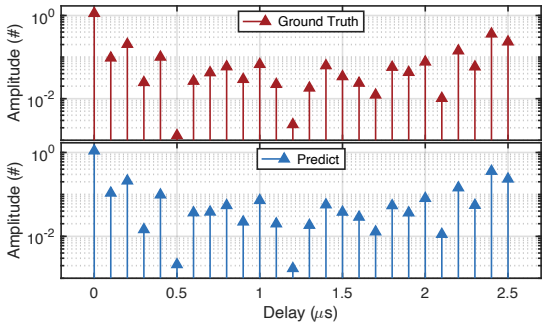


Fig. 4.13: Channel Impulse Response

tries; a 70-30 split is employed for training and testing, respectively. For comparative analysis, we select FIRE [26], R2F2 [25], OptML [28], and FNN [94]. Additionally, we compared our proposed frequency-aware NeRF^2 with the standard NeRF^2. Ensuring a fair comparison, we derive the experimental outcomes of these four algorithms (inclusive of SNR and SINR) from [26], as these metrics are also gauged based on the identical dataset.

Channel Prediction Accuracy. First, we evaluate the channel prediction accuracy of the downlink OFDM channel using frequency-aware NeRF^2. Fig. 4.12 illustrates a prediction result. The uplink CSI depicted in blue comprises amplitude and phase values for 26 subcarriers, which are input into the frequency-aware NeRF^2 as position indicators to construct the radiance field of the environment for the downlink channel. The yellow line represents the ground truth of the downlink CSI. Given that the uplink and downlink channels operate at different frequencies, there is a large disparity between the uplink and downlink CSI, complicating the direct prediction of downlink CSI from uplink CSI. The red line in the figure demonstrates the predicted downlink CSI using frequency-aware NeRF^2, closely aligning with the ground truth and achieving a mean error of 0.0163 in amplitude and 0.043 radians in phase. We also demonstrate an example of channel prediction accuracy in the time domain, described by the Channel Impulse Response (CIR), as illustrated in Fig. 4.13. The CIR characterizes how a channel affects an input signal by showing the dispersion of the signal's energy over time due to multipath propagation. In this environment, each

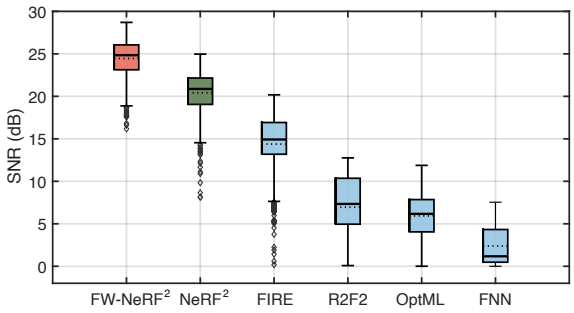


Fig. 4.14: Prediction SNR

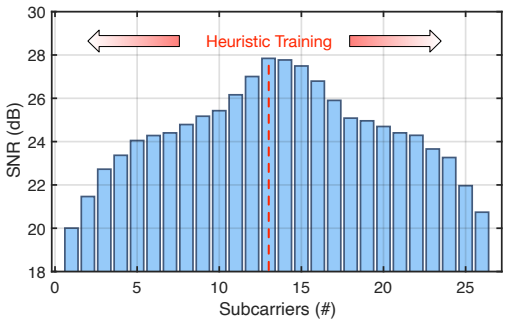


Fig. 4.15: SNR vs. Subcarriers

path contributes a version of the transmitted signal that is both delayed and attenuated, resulting in a composite received signal. The CIR is obtained through the IFFT of the CSI. The red arrows in the figure represent the ground truth impulse response of the downlink channel over time. It is evident that our prediction, indicated by the blue arrows, closely aligns with the ground truth, demonstrating a mean error of 0.0054 in amplitude. These results confirm that frequency-aware NeRF^2 provides accurate channel predictions in both time and frequency domains.

For quantitative accuracy assessment, we employ the prediction SNR metric, as suggested in [26]. This metric compares the predicted channel $\mathbf{H}$ with the ground truth channel $\mathbf{H}_{\text{gt}}$, defined as follows:

$$\text{SNR} = -10 \log_{10} \frac{\|\mathbf{H} - \mathbf{H}_{\text{gt}}\|^2}{\|\mathbf{H}_{\text{gt}}\|^2} \tag{4.19}$$

A higher SNR value indicates a closer alignment between the predicted and the ground truth channels. Fig. 4.14 shows the SNR of prediction results for six prediction algorithms, with frequency-aware NeRF^2 distinctly outperforming the others, achieving a median SNR of 24.84 dB (10^{th} percentile: 21.37 dB, 90^{th} percentile: 26.98 dB). In comparison, the median SNRs for NeRF^2, FIRE, R2F2, OptML, and FNN are 20.87 dB, 14.9 dB, 7.3 dB, 16.1 dB, and 1.2 dB, respectively. Owing to the RF prism module, frequency-aware NeRF^2 effectively represents the radiance fields of different frequencies in the wideband signal, achieving an SNR that is 3.97 dB higher than

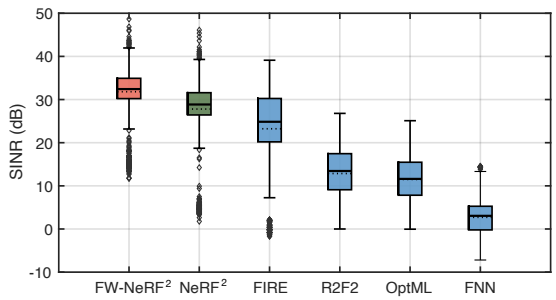


Fig. 4.16: MU-MIMO SINR

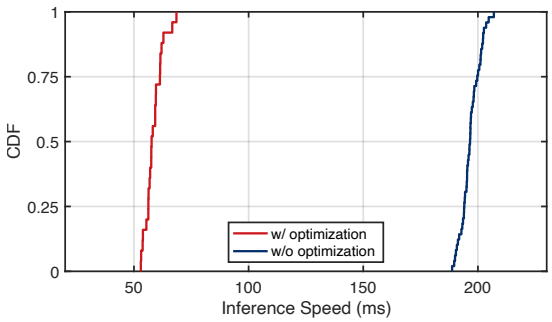


Fig. 4.17: Runtime Evaluation

NeRF^2, 9.94 dB higher than FIRE, 17.54 dB higher than R2F2, 8.74 dB higher than OptML, and 23.64 dB higher than FNN. These results underscore the high-accuracy prediction capabilities of frequency-aware NeRF^2.

Accuracy Across Subcarriers. We then assess the channel prediction accuracy across 26 downlink subcarriers. During the training of frequency-aware NeRF^2, heuristic methods are employed to reduce training times. The training process begins with the 13$^{\text{th}}$ subcarrier, from which well-trained parameters are iteratively deep-copied to adjacent subcarriers. In the fine-tuning stage, only the RF prism MLP heads' parameters are adjusted. Fig. 4.15 presents the results of the channel prediction accuracy across these subcarriers, indicating that the 13$^{\text{th}}$ subcarrier—the starting point—achieves the highest mean SNR of 27.34 dB. As the heuristic training progresses, there is a gradual decrease in accuracy for the MLP head of nearby frequencies. Nevertheless, the heuristic training method still attains high accuracy. After seven iterations, the SNR for subcarriers from the 5$^{\text{th}}$ to the 22$^{\text{th}}$ remains above 24 dB. Following the completion of fine-tuning for all 26 subcarriers over 13 iterations, frequency-aware NeRF^2 achieves an average SNR of 24.53 dB, with all subcarriers' predicted SNRs exceeding 20 dB, sufficient for upstream MU-MIMO or beamforming tasks.

Performance of MU-MIMO. We then delve into assessing the effectiveness of frequency-aware NeRF^2 in MU-MIMO operations, where a base station manages si-

multaneous transmissions to multiple clients through distinct beamforming channels. The key to optimal MU-MIMO performance is accurate channel estimation. Even slight errors in client-specific CSI can lead to significant interference among clients. For further details on data encoding by the base station in MU-MIMO contexts, readers are referred to [26, 95] due to space constraints.

In our analysis, we configure a setup where two clients are paired with a base station equipped with eight antennas, establishing an 8×2 MU-MIMO system. The system's performance is measured by the signal-to-interference-and-noise ratio (SINR). Results, depicted in Fig. 4.16, show that frequency-aware NeRF^2 achieves a median SINR of 32.43 dB, markedly higher than the median SINR of 29.22 dB for NeRF^2, 24.90 dB for FIRE, 13.33 dB for R2F2, and 11.53 dB for OptML. This superior performance of frequency-aware NeRF^2 can be attributed to its precise channel estimation capabilities of wideband RF signal.

Runtime Evaluation. Finally, we assess the inference speeds of frequency-aware NeRF^2 with and without the optimization methods proposed in §4.3.3. To ensure a fair comparison, all models are implemented in the PyTorch framework and evaluated using the same dataset to determine inference times. These evaluations are performed on the identical computational setup described in the experimental setup section, guaranteeing consistency across all tests. Fig. 4.17 presents the CDFs of inference times for the two methods. Notably, the median inference times are 57 ms with optimization and 196 ms without. This marked improvement in efficiency is primarily due to the implementation of accumulation networks, which replace the computationally intensive integral calculations of voxel attenuation along paths. Previously, the network required querying D times to compute the final attenuation and perform backpropagation. Now, with the accumulation networks, this process is streamlined to a single query, reducing the overall complexity to $\mathcal{O}(M \cdot \Omega \cdot D)$.

4.7 Field Study: BLE Localization

In this section, we discuss how NeRF^2 helps indoor localization in the scenario where no antenna array is available at a receiver. We conduct a large-scale experiment with 50 BLE gateways in an elderly nursing home. The project aims to track the potential spread of COVID-19 to protect elderlies from the infection better.

4.7.1 Experiment Setup

Fig. 4.18 shows the floor plan of the facility, which occupies 15,000 ft^2. A total of 50 BLE gateways (red circles) are deployed to collect the ID and RSSI of BLE beacons. Each gateway is $72 \times 7 \times 20\text{mm}^3$ in size, operates at 2.4 GHz and adopts an NRF52832 Bluetooth SoC [96] from Nordic Semiconductor. Redundant gateways are deployed to ensure that each location can be covered by at least 3 gateways. The BLE nodes are embedded into the visitor cards or elderlies' wristbands. They broadcast every 500 ms with 4 dBm transmitting power.

Ground Truth. The Velodyne VLP-16 LiDAR plus a 9-axis IMU are used to serve LIO-SAM (i.e., a publicly available SLAM algorithm [97]) for localization and map construction. The gateways and nodes are located by the LiDAR system as the ground truth. Taking 30 BLE nodes, we randomly walk into the house and totally create a dataset involving 6 K positions in the scene. Each dataset item is a 50-dimensional tuple, including the RSSI values detected by the 50 gateways, plus the position of the BLE node. The RSSI value is set to -100 dB by default if the gateway does not detect any signal from the node. 70% (4.2 K) and 30% (1.8 K) of the dataset are chosen from training and testing datasets.

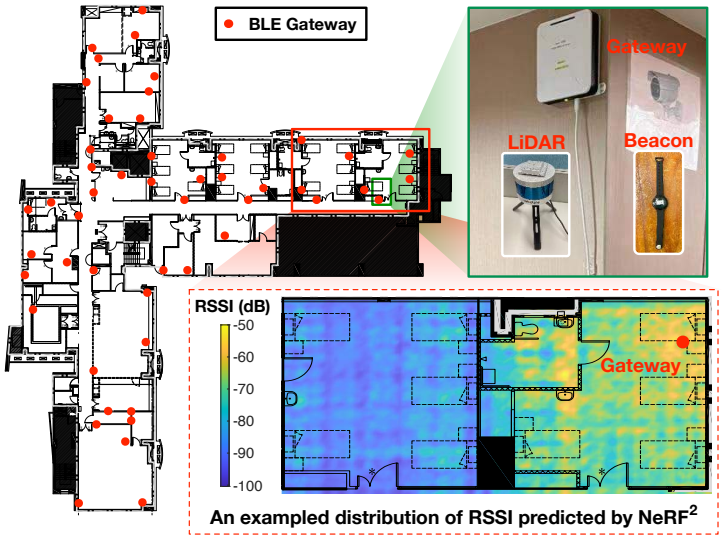


Fig. 4.18: The floor plan of the nursing home and deployment of BLE gateways.

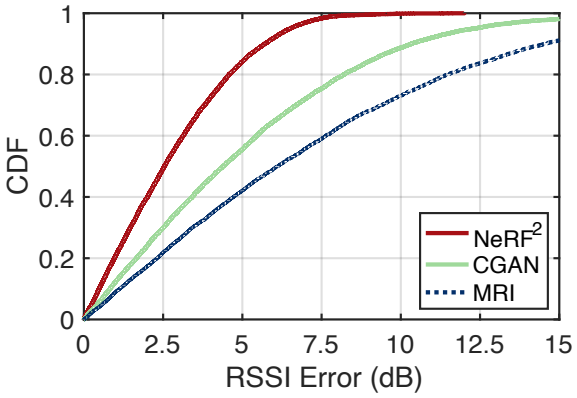


Fig. 4.19: RSSI Prediction

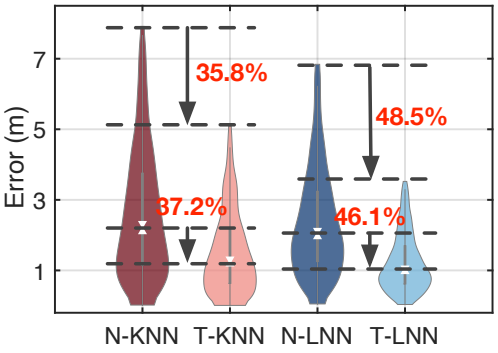


Fig. 4.20: Localization Result

4.7.2 RSSI Prediction

Data-driven approaches are emerging as promising solutions for BLE localization, such as KNN, SVM, and MLP. These methods require an accurate dataset for fingerprint matching or network training. Here, we apply turbo-learning for BLE localization, where the NeRF^2 is trained using the single-antenna RX model (see §4.2.4). The training process is more complicated than in previous cases in that we have 50 RXs here. The beacon of the same BLE node may be received by multiple gateways simultaneously. In this case, we must take ray tracing multiple times, in each of

which the result is an aggregation of signals arriving at the corresponding RX from all possible directions (Eqn. 4.9). Similarly, given a position that a BLE node locates in the scene, we must take the ray tracing to predict the RSSI of the signal received at any gateway with the help of NeRF^2. Fig. 4.18 shows an example distribution of the predicted RSSI across the two rightmost rooms. It can be seen that the coverage of a gateway is not as good as that the manual claims (i.e., 10 m). The signal becomes very weak after walls. Thus, we deployed 3-4 gateways in each room to ensure full coverage. For comparison, we also adopt two other proposed prediction approaches, MRI [98] and CGAN [99]. MRI interpolates the RSSI values at the unsampled location using a basic radio propagation model. CGAN uses the conditional generative adversarial network to predict the RSSI values straightforwardly without regarding any physical model. The prediction error is defined as the difference between the predicted and the collected RSSI values at 1.8 K tested positions. The CDFs of the prediction error are shown in Fig. 4.19. As a result, the median of NeRF^2 is 2.6 dB (10^{th} percentile: 0.5 dB; 90^{th} percentile: 5.7 dB). By contrast, the median errors of CGAN and MRI are 4.5 dB and 6.2 dB, respectively. Evidently, NeRF^2 performs far better than the two others because it combines the advantages of deep learning (e.g., CGAN) and the physical model (e.g., MRI). The physical model provides prior knowledge about signal propagation, while deep learning uses statistical models to depict complicated RF interactions.

4.7.3 Localization Results

We use the well-trained NeRF^2 to generate a 20 K synthetic dataset at random locations and feed them to the following two localization algorithms.

Turbo-KNN (T-KNN): We first evaluate the fingerprint-based localization approach, which assumes the RSSI values are highly related to a node's location. The K-nearest positions are chosen to compute the target node's location, where the RSSI

values collected from these K positions (saved in a database) are most close to the RSSI value collected from the unknown position. The node is located at the weighted average of the K positions [40]. Fig. 4.20 shows the localization accuracy of naive-KNN (N-KNN) and T-KNN. The N-KNN only adopts the 4.2K true dataset only, whereas T-KNN uses the 20 K synthetic dataset. The median error of T-KNN is 1.41 m (10^{th} percentile: 0.27 m; 90^{th} percentile: 3.3 m), whereas that of N-KNN is 2.52 m (10^{th} percentile: 0.61 m; 90^{th} percentile: 5.38 m). Turbo-learning helps the KNN-based localization approach reduce the error by 44%.

Turbo-LNN (T-LNN): We build another neural network to learn the mapping between an RSSI tuple and a position. We call this network *localization neural network* (LNN), which accepts the 50-dimensional RSSI tuple as input and outputs the position. The LNN consists of five-layer fully connected layers with the ReLU activation function. Similarly, the LNN is trained by using the 4.2 K true dataset and 20 K synthetic dataset, respectively. We call them naive-LNN (N-LNN) and T-LNN. Fig. 4.20 shows their results. The median error of T-LNN is 1.11 m (10^{th} percentile: 0.34 m, 90^{th} percentile: 3.46 m), whereas that of N-LNN is 2.26 m (10^{th} percentile: 0.75 m, 90^{th} percentile: 6.78 m). The error is reduced by turbo-learning by 50.8% (i.e., 1.2 m error). Particularly, T-LNN further reduces the 30 cm median error than T-KNN.

In summary, NeRF^2-powered turbo-learning can effectively reduce the localization errors by $\sim 50\%$ regardless of which data-driven approach is used. It also decreases the standard variance by $\sim 40\%$ because the scale of training data is enlarged by $5\times$ and a larger number of samples clearly benefit the convergence.

4.7.4 Impact of Label Errors

Label accuracy critically affects deep learning algorithm performance, so we investigate the impact of beacon location label errors on RSSI prediction accuracy. Fig. 4.21 shows the results, representing median prediction error and quartiles. We introduce

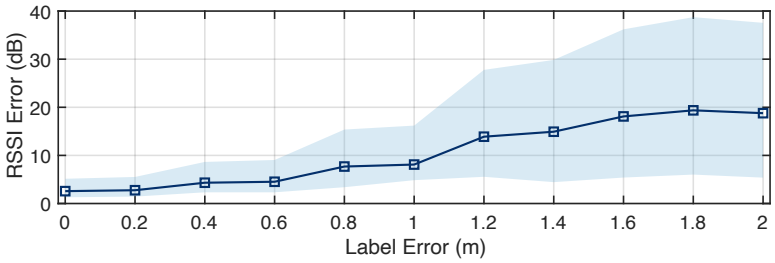


Fig. 4.21: RSSI vs. Label error

uniformly distributed noise to the location label to simulate errors, whereby the label is reported with the circular error of radius r. The error r is increased from 0 m to 2 m in a step of 0.2 m, and the corresponding median RSSI prediction error degrades from 2.6 dB to 18.8 dB, with a near 7× increase. The standard deviation also increases from 1.9 dB to 9 dB. When the label error exceeds 1 m, the prediction accuracy of RSSI decreases severely. This is primarily because when the error surpasses 1 m, the label may be inaccurately situated in another room. On the other hand, when is label error is less than 0.6 m, the RSSI prediction error is below 4.5 dB, which is only 1.9 dB when compared to the absence of error. As such, we adopt the SLAM algorithm, which guarantees a label error of less than 0.2 m, as our preferred method of collecting ground truth data.

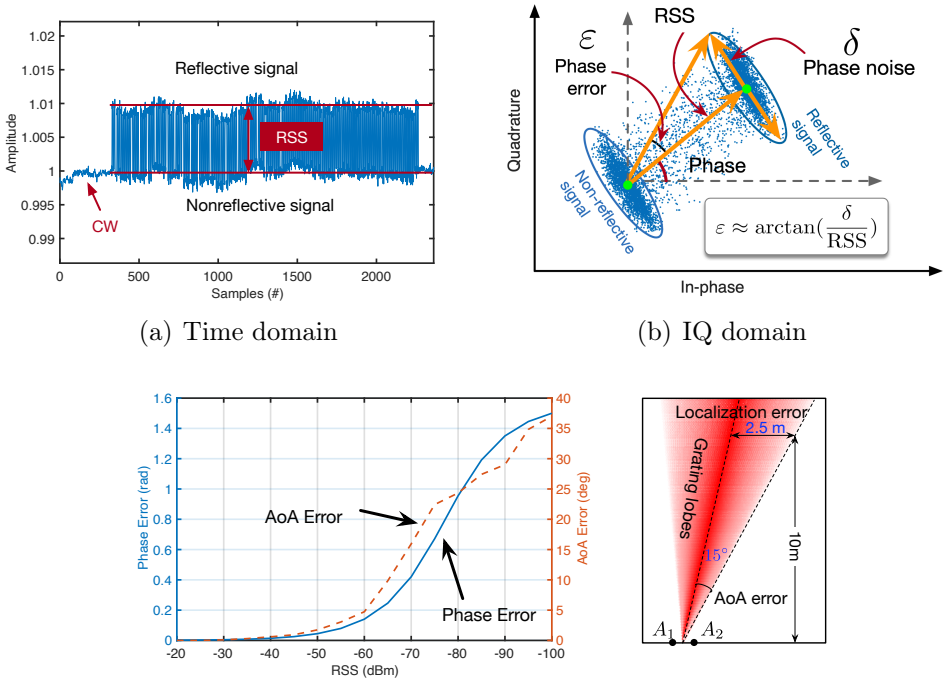
Chapter 5

Consistent Phase Estimation Protocol

5.1 Motivations

We define the working range of a backscatter system as the distance from the RF source → tag → gateway. Commercial backscatter tags (e.g., UHF RFIDs) are capable of maintaining steady communication with gateways over relatively long ranges, provided that the nearby transmitter supplies sufficient energy to the tag. We refer readers to [77], which elaborates how the tag-to-receiver range can exceed 130 meters at 1 kbps with a transmitting power of 30 dBm. However, the high-precision localization range is limited to several meters even when using the most popular and accurate localization algorithms.

Why is it challenging to precisely locate a backscatter tag over long distances? Our discussion focuses on Angle of Arrival (AoA) based localization systems. We find that the accuracy of most AoA estimation algorithms relies heavily on how precisely the RF phase is measured, as this phase acts as a critical indicator of location. A pivotal factor influencing this precision is the received signal strength (RSS), defined as the power of reflected signals relative to continuous waves (CW). We collected backscatter signals from an RFID tag to explore this issue empirically. Our analysis, presented



(a) Time domain (b) IQ domain

(c) Phase Error and AoA Error vs. RSS (d) AoA model

Fig. 5.1: The analysis on localization error. (a) shows a backscatter signal in the time domain; (b) shows the signal in the IQ domain where two clusters are created by the reflective and the non-reflective signals, respectively. The phase is defined as the angle of the vector connecting the centers of the two clusters; (c) shows the phase error and the AoA error as a function of the RSS; (d) shows that a small error (i.e., 15°) in AoA leads to a great discrepancy in the localization result at a distance (i.e., 2.5 m @ 10 m distance).

in Fig. 5.1(a) and 5.1(b), showcases these signals in both time and IQ domains. The IQ diagram, in particular, reveals two distinct clusters indicative of reflective and non-reflective signals. The separation between these clusters represents the RSS, while the RF phase corresponds to the angle of the vector joining the centers of these clusters. Ideally, each cluster should collapse into a point in the IQ diagram. However, due to phase noise, the actual samples form spindle-shaped clusters, resulting in the phase estimation swinging within a small range, termed as phase error. Let ε be the estimated phase error and δ be the deviation of the phase noise. As shown in Fig. 5.1(b), we can simply use a triangle to depict their relationship intuitively as

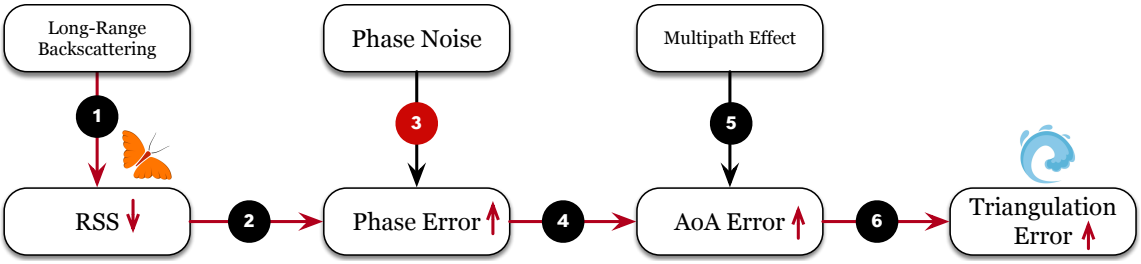


Fig. 5.2: Butterfly effect. Initial small errors amplify through localization stages, significantly affecting accuracy and limiting the precise localization range.

follows (see Eqn. 5.5 for rigorous expression):

$$\varepsilon \approx \arctan(\frac{\delta}{\text{RSS}}) \tag{5.1}$$

To be more specific, we plot the above equation in Fig. 5.1(c), where the δ is fixed to 0.001 radians. When given a -70 dBm RSS backscatter signal, the phase error is raised to 0.4 radians, resulting in an unacceptable 15° error in the angle of arrival (AoA). The greatest known limitation of triangulation is that "a miss is as good as a mile", i.e., a small error (i.e., 15°) in AoA leads to a significant discrepancy in the localization result at a distance (2.5 m error at 10 m), as shown in Fig. 5.1(d). By contrast, the decoding can be successfully conducted once the RSS is greater than -90 dBm.

A small reduction in RSS, particularly over long distances, can set off a cascade of errors throughout the system, much like a butterfly effect, as shown in Fig. 5.2. *In essence, minor errors at the outset are progressively amplified at each stage, ultimately resulting in significant position inaccuracies.* This phenomenon explains why extending the tracking range linearly with the communication range is challenging. Therefore, the key to mitigating this "butterfly effect" lies in addressing the issue at its source – by improving the accuracy of phase measurement at the earliest stage. Moreover, Eqn. 5.1 suggests two effective strategies to enhance phase measurement accuracy: increasing RSS or reducing phase noise. However, boosting RSS is often not viable due to FCC regulations, which cap the power of indoor RF sources at 1 W [100]. While alternative approaches like distributed MIMO [44], beamforming [24],

or constructive power surges [59] exist, they typically require multiple RF sources or a broader bandwidth, leading to increased power consumption and higher infrastructure costs. Consequently, the most practical and effective solution is to minimize phase noise in low-RSS backscatter signals. Recent studies have demonstrated that utilizing receivers with low phase noise, such as quantum receivers [101], can enhance localization accuracy at the hardware level. However, these high-sensitivity, low-noise RF devices come with significant costs. Our paper focuses on improving the phase estimation protocol to minimize phase noise, thereby enhancing the accuracy of AoA estimations.

In this work, we introduce the design and implementation of the first high-precision, long-range backscatter positioning system. This system does not rely on expensive, low-noise RF components. Yet, it works effectively even in low-RSS scenarios that result from far-distance backscatter communication. Unlike past localization works, which primarily focused on designing new localization algorithms, this work improves localization performance by addressing a fundamental problem: how can we enhance the physical-layer measurement quality for localization algorithms? To this end, we propose two protocols for localization-oriented phase estimation, namely CPE and CPE+. To clarify, the "physical layer signal" denotes the IQ baseband backscatter signal captured by the gateway. While "estimation protocol" includes the algorithms employed to process these physical layer signals, aiming to determine the phase value accurately. Firstly, we present a *consistent phase estimator* (CPE), which resolves the longstanding industry issue of temporal inconsistency. Specifically, the phenomenon where the measured phase value jumps between two values differing by π radians, even when the backscatter tags are stationary. Then, we develop CPE+, which enhances the CPE with three denoising algorithms to address the unique challenges in backscatter systems, particularly in bistatic mode where the continuous wave (CW) acts both as a carrier for on-off modulation and a significant source of interference. The enhancements of CPE+ include: (1) a side-channel-aware automatic flicker noise

canceller (AFNC) that utilizes a pure CW signal from the RF source to cancel out flicker noise resulting from clock jitters; (2) a white noise neutralizer that reduces the white Gaussian noise amplified fourfold in the AFNC; (3) restoring spatial and temporal imbalances. Spatial imbalance refers to the diversity in the RSS signals received by different elements in the array, which can lead to power loss in the target direction of the spatial spectrum. We propose using a cyclic redundancy reference to restore spatial imbalance. On the other hand, temporal imbalance is caused by automatic gain control (AGC), which results in regular phase hopping as the gain changes. To address this, we have designed an additional method to restore temporal imbalance.

Summary. We have built five customized gateways based on the proposed phase estimation protocols, which have been extensively evaluated across the commercial off-the-shelf (COTS) RFID tags. Fig. 2.1 summarizes the accuracy of our system as a function of the working range, compared with SOTA typical backscatter positioning systems. Our system extends the range of cm-level localization accuracy from 8 m to 15 m. It also almost maintains dm-level accuracy throughout the entire communication range, by avoiding the butterfly effect from the beginning.

5.2 Overview

Our primary strategy to effectively track backscatter tags involves designing and implementing a gateway equipped with multiple antennas. This setup enables us to estimate the direction of a backscatter tag, known as the AoA. By employing two gateways, we can get the tag's location in 2D space, and with three gateways, in 3D space. Our unique challenge, compared to previous works, lies in determining the direction of a tag that is at a considerable distance (e.g., approximately 50 meters away), where the gateways receive backscatter signals with low RSS (e.g., less than -70 dBm).

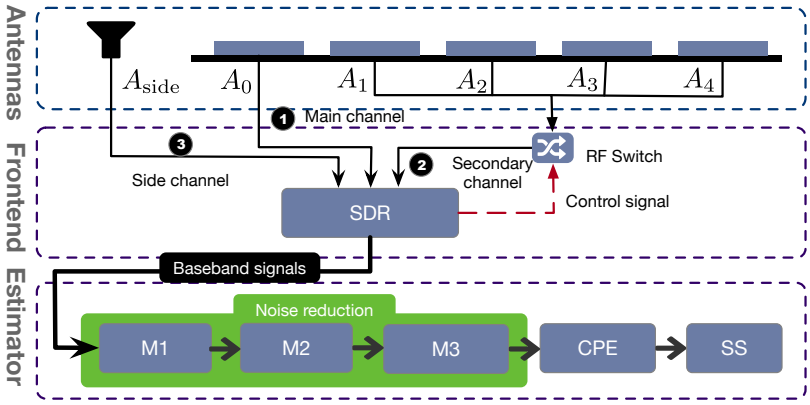


Fig. 5.3: Design of the gateway. The gateway contains a three-layer design: (1) Antenna layer with a 4×4 array and a directional side antenna for signal acquisition. (2) Front-end layer where RF signals are downconverted to baseband by SDR, creating main, secondary, and side channels for decoding and localization. (3) Estimator layer, the core algorithm hub for RF phase estimation and denoising.

5.2.1 System Architecture

The key component of our system is the design and implementation of a customized gateway. The gateway comprises three layers, namely, antenna, front-end, and estimator layers, as shown in Fig. 5.3. **(1) Antenna layer:** The antenna layer includes a uniform 4×4 antenna array and a stand-alone directional antenna called side antenna. The array is used to acquire the backscatter signals, and the side antenna aims to acquire a pure CW signal without backscatter signals. **(2) Front-end layer:** In this layer, RF signals are downconverted into baseband signals by Software Defined Radio (SDR), wherein we establish three channels: main, secondary, and side channels. The *main channel* connects to the antenna A_0 and persistently receives the backscatter signal, which will be used for the decoding. The *secondary channel* connects antennas $A_1 \sim A_{15}$ in a time-sharing manner through RF switches for localization. The *side channel* connects to the side antenna A_{side} for building AFNC. **(3) Estimator layer:** The core of the gateway is the estimator layer. In this layer, we first design the core algorithm (called CPE) to estimate the RF phase (see § 5.3). Then, three main denoising measures (called M1, M2, and M3) are presented to enhance CPE

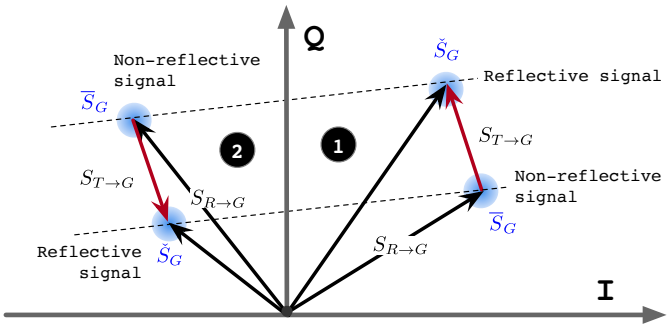
(see § 5.4). The estimated phases are subsequently utilized to generate the spatial spectrum (SS), indicating the direction of the incoming signal. Details regarding the hardware configuration of our system are elaborated in the implementation section (see § 5.5).

5.3 CPE: Consistent Phase Estimation

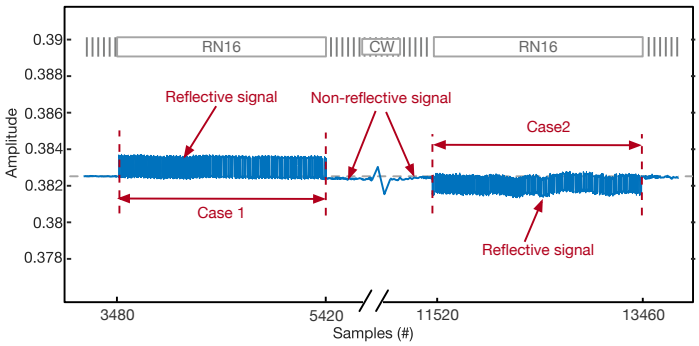
In the section, we first introduce the limitations of NPE introduced in section 3.1 and an algorithm to solve the problem with *relatively high RSS*.

5.3.1 Consistent Phase Estimator

Ideally, the phase of backscatter signals received from a stationary tag should remain unchanged over time. Actually, the phase reported by the commercial devices often jumps between two values, which differ by π-radians, even if the tag remains still. This is the notorious phenomenon called π-*ambiguity*. Reflective signal superposition involves two distinct scenarios: (1) Constructive, where the signal combines in a peak-to-peak manner, enhancing the strength of the reflective signal compared to the non-reflective signal, and (2) Destructive, where the signal combines in a peak-to-valley manner, resulting in a weaker reflective signal. The NPE only considers the constructive, i.e., a reflective signal should be stronger than a non-reflective signal. Thus, NPE always computes the complex difference by subtracting the low-amplitude cluster from the high-amplitude cluster, making the resulting vector always points outward, as shown in case ❶ in Fig. 5.4(a). However, the NPE leads to π-ambiguity in the destructive situation. When the signal is superimposed destructively (case ❷ in Fig. 5.4(a)), the reflective signal is weaker than the non-reflective signal. In this case, if we subtract the weaker signal from the stronger signal, then we actually compute $\overline{S}_G - \breve{S}_G$ rather than the desired difference of $\breve{S}_G - \overline{S}_G$. The two resulting vectors are



(a) Uncertainty in the directivity



(b) Reflective/non-reflective signal vs. high/low voltage

Fig. 5.4: The π-ambiguity in NPE. (a) shows two cases where the uplink and self-link are superimposed constructively and destructively due to the environmental dynamics, making the directions of the complex difference differ by π radians; (b) shows a practical backscatter signal acquired from a stationary RFID tag twice, which demonstrate the π-ambiguity in the time domain.

the same in amplitude but differ by π radians in angle (i.e., pseudo-phase). To verify our hypothesis, we acquire backscatter signals from the same stationary tag twice.

Fig. 5.4(b) shows the acquired signals in the time domain. In the first reply, the high-/low-voltage samples are from the reflective/non-reflective signals, respectively. The pseudo-phase computed using NPE is 3.5 radians. However, the mapping is reversed in the second reply due to environmental changes. The pseudo-phase is flipped to $3.5 + \pi = 6.6$ radians correspondingly.

To disambiguate the π-ambiguity, we develop the consistent phase estimator (CPE), which can hold the temporal consistency. Firstly, CPE finds out all transitional pairs, each of which contains two adjacent samples across a state transition; i.e., one sample

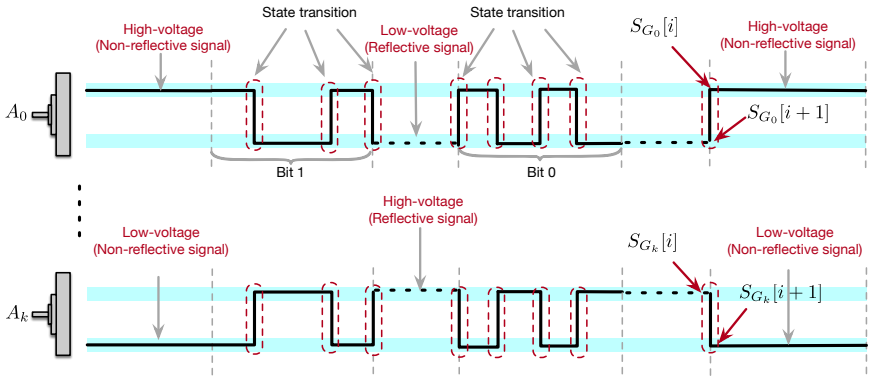


Fig. 5.5: Backscatter signaling. The figure shows the baseband signals received by the antennas A_k and A_0. In the signal S_{G_0}, the high-voltage/low-voltage samples are from the non-reflective/reflective signals, respectively. However, the mapping is reversed in the signal S_{G_k}. CPE adopts a pair of transitional samples across the state transition to compute the pseudo-phase.

has a high voltage, and the other has a low voltage, as shown in Fig. 5.5. Suppose $S_{G_k}[i]$ and $S_{G_k}[i+1]$ are a pair of transitional samples received by antenna A_k. We utilize this pair to compute the pseudo-phase as follows:

$$\angle \Delta S_{G_k}[i] = \angle (S_{G_k}[i] - S_{G_k}[i+1]) \tag{5.2}$$

Notably, *the two samples definitely follow into the two clusters differently. Thus, the resulting angle is one estimation of the pseudo-phase.* Nevertheless, this estimation does not eliminate the π-ambiguity yet, i.e., i$\angle \Delta S_{G_k}[i] = \widetilde{\theta}_{T \to G_k}[i]$ or $\widetilde{\theta}_{T \to G_k}[i] + \pi$. Note that the π-ambiguity may differ between two antennas. Directly computing the relative phase in AoA estimation using the pseudo-phase method does not resolve this issue.

We observe an insight that the state transition must appear at all antennas at the same time because RF signals are received by all antennas simultaneously. The delays caused by distance differences at two antennas are negligible regarding the light speed and MHz-level sampling rate. Therefore, if $S_{G_k}[i]$ and $S_{G_k}[i+1]$ (received by A_k) are a pair of transitional samples, then $S_{G_0}[i]$ and $S_{G_0}[i+1]$ (received by A_0) must be a pair of transitional samples as well. As illustrated in Fig. 5.5, if the i^{th} sample is acquired in a reflective state, then $S_{G_k}[i]$ and $S_{G_0}[i]$ are from the reflective signal or

vice versa. Hence, $\angle(S_{G_k}[i]-S_{G_k}[i+1])$ and $\angle(S_{G_0}[i]-S_{G_0}[i+1])$ either contain the additional π radians or not. We use this new pseudo-phase to compute the complex division as follows:

$$\begin{aligned}\rho_k[i] &= \frac{\breve{A}_k}{\breve{A}_0} e^{\mathbf{J}\left((\tilde{\theta}_{T\to G_k}[i]+\pi)-(\tilde{\theta}_{T\to G_0}[i]+\pi)\right)} \\ &= \frac{\breve{A}_k}{\breve{A}_0} e^{\mathbf{J}\left(\tilde{\theta}_{T\to G_k}[i]-\tilde{\theta}_{T\to G_0}[i]\right)}\end{aligned} \tag{5.3}$$

The additional π radians (if existing) are perfectly removed from the division. Although $\angle\Delta S_{G_k}[i]$ itself does not disambiguate π-ambiguity, it ensures that the result of spatial spectrum is unaffected by the ambiguity.

Suppose M pairs of transitional samples exist in the received backscatter signal. CPE delivers the mean of the M complex divisions to the upper layer as follows:

$$\overline{\rho}_k = \frac{1}{M}\sum_{m=1}^{M}\frac{S_{G_k}[i_m]-S_{G_k}[i_m+1]}{S_{G_0}[i_m]-S_{G_0}[i_m+1]} \tag{5.4}$$

where $m = 1, 2, \ldots, M$. In this way, π-ambiguity is totally eliminated from the complex divisions, holding temporal consistency. Finally, we depict the difference between NPE and CPE in Fig. 5.6. Specifically, NPE uses the angle of the vector connecting the centers of two clusters (highlighted in green) as the pseudo-phase, whereas CPE uses the mean angle of the vectors connecting multiple pairs of transitional samples (highlighted in red) to estimate the pseudo-phase.

5.4 CPE+: Enhanced CPE via Denoising

In this section, we consider the case of far-distance communication in which backscatter signals are received with *low RSS*. As aforementioned, an effective way for reducing the phase error is the denoising except increasing power. To this end, we present three denoising measures (i.e., M1∼M3) progressively. We call the enhanced CPE by the denoising algorithms as CPE+.

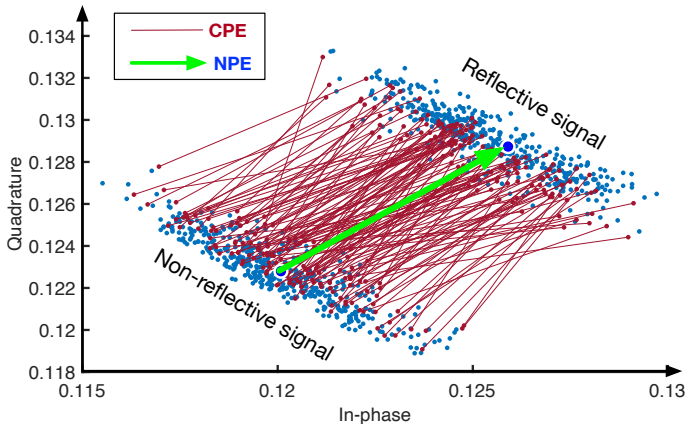


Fig. 5.6: CPE vs. NPE. NPE determines pseudo-phase using the angle between cluster centers (green), while CPE calculates it from the average angle across multiple transitional sample pairs (red).

5.4.1 Revisiting Phase Noise

Phase noise refers to the random fluctuations in the phase of a waveform. An ideal oscillator would generate a pure sine wave, but all real oscillators have phase-modulated noise components. Phase noise often includes low-frequency flicker noise and full-frequency white noise. **(1) Flicker noise**: Flicker noise is caused by the unpredictable jitter of a clock. Once the transmitter and receiver adopt different clocks, the flicker noise cannot be compensated. Flicker noise is a time-varying random variable related to the clock. **(2) White noise**: White noise is a random signal having an equal intensity at different frequencies. It is an internal characteristic of each electronic component and irreverent to the clock. Thus, white noises are different in channels even if they share the same clock.

To better understand phase noise, we use the main and secondary channels to acquire a piece of CW signal concurrently during the same window. The CFO of the two signals has been compensated. Fig. 5.7 compares the phase of the two signals, in which the fluctuations are completely from the two types of noises. The stds of the two signals are 0.011 and 0.010 radians. Given that the two channels share a single clock, they receive the same flicker noise from the RF source but different white noises. Visually, it appears that the two phase sequences are highly coherent, which

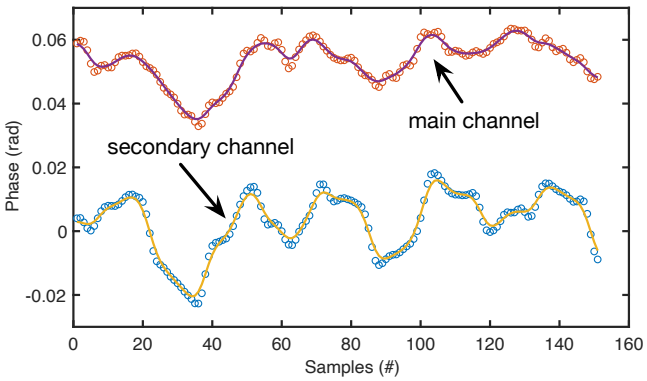


Fig. 5.7: Phase noise in the time domain.

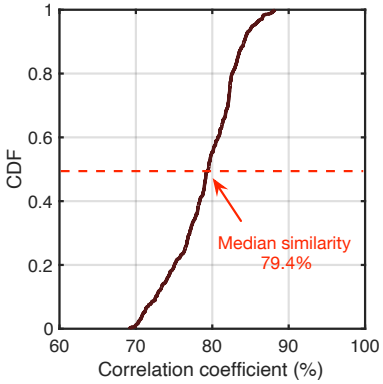


Fig. 5.8: CDF of correlation.

is also confirmed by cross correlation. Their correlation coefficient is up to 80%. This result indicates that the flicker noise dominates the phase noise and it remains consistent across the two channels. The remaining 20% incoherence comes from the white noise. We subtract the two phase sequences to cancel out the flicker noise. As desired, the std of the residual sequence is reduced to 0.006 radians, which is caused by the unavoidable white noise. To quantitively evaluate the consistency of flicker noise across the two channels, we conduct an empirical study. The backscatter tag was positioned in five distinct locations. We collected 100 phase sequences at each location, resulting in a total of 500 phase sequences. Fig. 5.8 shows the CDF of the correlation coefficient between the two channels. The analysis revealed a median similarity of 79.4%, indicating consistency in the flicker noise's contribution to the phase noise across different channels and locations.

5.4.2 M1: Cancelling Flicker Noise

In a high RSS case, the phase noise ($\breve{\phi}$ and $\overline{\phi}$) can be approximated as a constant. In a low RSS case, it should be modeled as a time-varying random variable. Considering a pair of transitional samples $\breve{S}_{G_k}[i]$ and $\overline{S}_{G_k}[i+1]$, the complex difference can be

rewritten as follows:

$$\begin{aligned}\Delta S_{G_k}[i] &= \breve{A}e^{\mathbf{J}(\theta_{T\to G}+\breve{\phi}[i])} + \overline{A}e^{\mathbf{J}(\theta_{R\to G})}(e^{\mathbf{J}\overline{\phi}[i]} - e^{\mathbf{J}\overline{\phi}[i+1]}) \\ &= \overline{A}e^{\mathbf{J}\phi_G}\left(\underbrace{\frac{\breve{A}}{\overline{A}}e^{\mathbf{J}(\theta_{T\to G}+\phi_R[i]+\phi_T)}}_{\text{Phase term}} + \underbrace{e^{\mathbf{J}\theta_{R\to G}}\left(e^{\mathbf{J}\phi_R[i]} - e^{\mathbf{J}\phi_R[i+1]}\right)}_{\text{Noise term}}\right)\end{aligned} \tag{5.5}$$

where the common term of $\overline{A}e^{\mathbf{J}\phi_G}$ will be eliminated from the next step of complex division. Given that ϕ_R is time-varying, $\phi_R[i] \neq \phi_R[i+1]$ and the second nonzero term cannot be removed. Surely, $\breve{A}/\overline{A}$ is proportional to the RSS. When RSS is sufficiently high, the first phase term dominates the sum and the minor phase term can be ignored. However, in the low RSS case, the noise term becomes non-negligible. This qualitative analysis is aligned with previous intuitive analysis shown in Section 5.1.

Inspired by the insight that the synchronized channels at a gateway receive the same flicker noise, we establish a side channel equipped with a directional antenna to cancel out the flicker noise, as shown in Fig. 5.3. Unlike the main and secondary channels, the side channel is designed to receive the pure CW from the RF source instead of from the tags. The baseband signal acquired using the side channel can be given by:

$$S_{\text{side}}[i] = e^{\mathbf{J}(\theta_{\text{side}}+\phi_R[i]+\phi_G[i])} \tag{5.6}$$

where θ_{side} is the phase rotation over the distance between the antennas of the RF source and the side channel. The amplitude is forcedly set to one because we only care about the phase noise. Meanwhile, we add a preprocessing step before phase estimation. Specifically, each sample acquired from the main channel or from the secondary channel is divided by the corresponding sample acquired from the side channel. Formally, the preprocessing is defined as follows:

$$\begin{cases}\overline{S}_{G_k}[i] = \dfrac{\overline{S}_{G_k}[i]}{S_{\text{side}}[i]} = \overline{A}_k e^{\mathbf{J}(\theta_{R\to G}-\theta_{\text{side}})} \\ \breve{S}_{G_k}[i] = \dfrac{\breve{S}_{G_k}[i]}{S_{\text{side}}[i]} = \overline{A}_k e^{\mathbf{J}(\theta_{R\to G}-\theta_{\text{side}})} + \breve{A}_k e^{\mathbf{J}(\theta_{T\to G_k}-\theta_{\text{side}}+\phi_T)}\end{cases} \tag{5.7}$$

The last derivations are obtained by substituting Eqn. 3.7 and Eqn. 5.6 into the division. Then, the complex difference over a pair of transitional samples is redefined

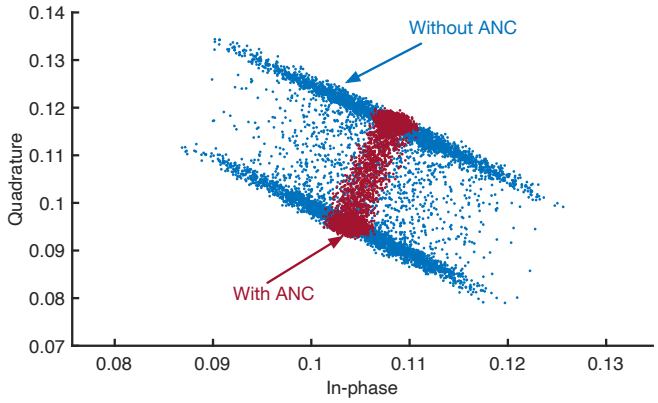


Fig. 5.9: The side-channel aware AFNC.

as follows:

$$\Delta S_{G_k}[i] = S_{G_k}[i] - S_{G_k}[i+1] = \breve{A}_k e^{\mathbf{J}(\theta_{T\to G_k}+\phi_T-\theta_{\text{side}})} \tag{5.8}$$

The phase offset, including the flicker noises, is totally removed from the difference, but the desired true phase of $\theta_{T\to G_k}$ is kept. Correspondingly, the complex division is simplified as follows:

$$\rho_k[i] = \frac{S_{G_k}[i] - S_{G_k}[i+1]}{S_{G_0}[i] - S_{G_0}[i+1]} = \frac{\breve{A}_k}{\breve{A}_0} e^{\mathbf{J}(\theta_{T\to G_k}[i]-\theta_{T\to G_0}[i])} \tag{5.9}$$

which is exactly the same as Eqn. 5.3 that is derived for the high RSS case. This finding demonstrates that the side-channel-aware AFNC can improve the estimated phase quality to the approximate level achieved in the high RSS case. Notice that any additional phase value θ_{side} introduced by the side channel's location is eliminated during the complex division step. This process ensures that the outcomes of phase estimation and tag localization become irrelevant to the side antenna's specific placement.

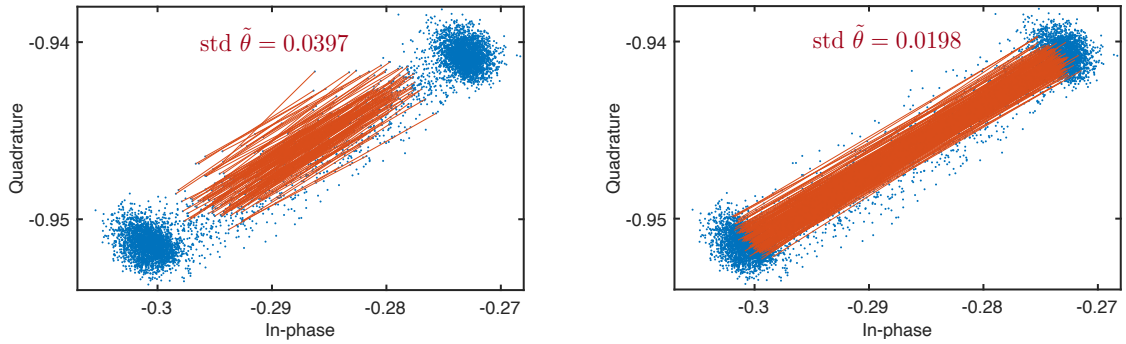
To understand the purpose of the side channel, Fig. 5.9 compares the raw backscatter signals (highlighted in blue) with the same signal after being preprocessed (highlighted in red). The amplitudes of samples are not changed because the preprocessing does not change amplitude. By contrast, the preprocessed clusters become much narrower and more compact than the raw clusters because of denoising. Specifically, the std of

the computed pseudo-phase is decreased from 0.068 rad to 0.038 rad. This result fully demonstrates that the side channel can work as an automatic flicker noise canceller (AFNC) to eliminate the flicker noise.

5.4.3 M2: Neutralizing White Noise

White noises remain in channels even after the flicker noise is canceled using AFNC. They are generally viewed as independent identically distributed Gaussian random variables. We would like to know how the white noise changes during the phase estimation. Let $n_i(t)$ be the white noise generated at the i^{th} antenna, i.e., $n_i(t) \sim \mathcal{N}(0, \delta)$. Suppose the signals acquired by the i^{th} channel are $A_i e^{\mathbf{J}\theta_i + n_i(t)}$. When we conduct the complex division between two signals, the result becomes $A_i/A_j e^{\mathbf{J}(\theta_i - \theta_j + n_i(t) - n_j(t))}$. Suppose both $n_i(t)$ and $n_j(t)$ follow $\mathcal{N}(0, \delta)$. Their difference follows $\mathcal{N}(0, 2\delta)$, namely, *the std of the white noise is doubled when we perform a complex division.* We conduct two complex divisions in total. One division is for the preprocessing in AFNC, and the other is for computing relative power. As a result, *the std of the white noise is amplified by fourfold.* Fig. 5.10(a) shows the results of the complex difference after the preprocessing by using AFNC, where each red vector connects a pair of transitional samples. Ideally, these vectors should remain in parallel because their angles (i.e., pseudo-phase) are the same. Actually, many of the vectors intersect with one another in the figure due to the presence of amplified white noise.

The most efficient way of reducing the influence of white noise is to neutralize the results by using additional samples because they follow the Gaussian model. In fact, multiple "redundant" samples exist on the two sides of a state transition, as shown in Fig. 5.5. They can help in noise neutralization. Suppose there are w samples that exist in each high-voltage and each low-voltage edge (aka window). We update the



(a) Before noise neutralization (b) After noise neutralization

Fig. 5.10: White noise neutralization

complex difference as follows:

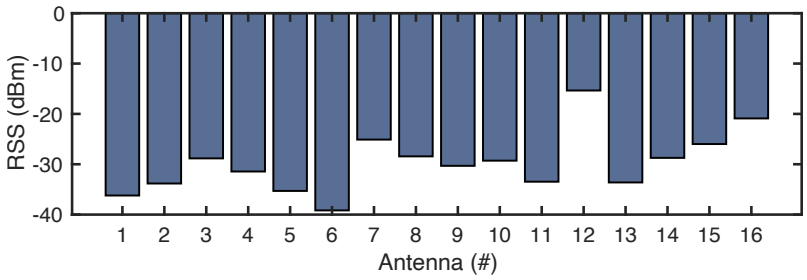
$$\Delta S_{G_k}[i] = \frac{1}{w^2} \sum_{\substack{i-w<a\leqslant i \\ i<b\leqslant i+w}} (S_{G_k}[a] - S_{G_k}[b]) \tag{5.10}$$

The complex difference is neutralized by using the w^2 combined transitional pairs. Fig. 5.10(b) shows the results of complex difference after neutralizing the white noise. Clearly, all vectors are more aligned with one another compared with those in Fig. 5.10(a). It seems that more red lines are in Fig. 5.10(b). Both figures actually show the same number of lines, which is determined by the number of state transitions. The optical illusion results from the misalignment in the first figure. Parameter w is a system configuration related to the sampling rate. Adopting a high-rate sampling can facilitate the smoothing remarkably.

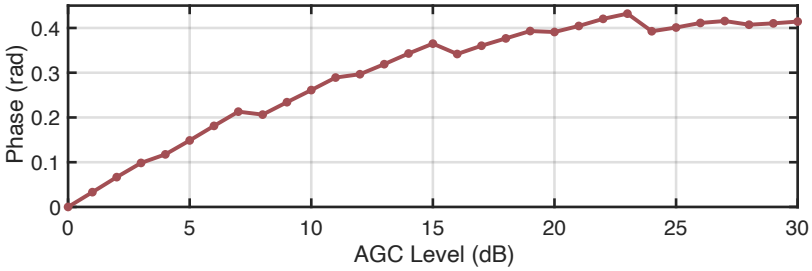
5.4.4 M3: Restore Spatial and Temporal Imbalance

In our analysis of the received backscatter signals, we identify two distinct types of imbalances:

• **Spatial Imbalance**. Conventionally, it is assumed that the RSS of signals received by all antennas in an array is roughly equal. This assumption leads to the selection of any one antenna in the array as a reference point for counteracting constant phase



(a) Spatial disequilibrium



(b) Temporal disequilibrium

Fig. 5.11: The spatial and temporal disequilibria

offsets, typically using the leftmost antenna for complex division (refer to Eqn. 3.12). However, we find that this assumption no longer holds in battery-free backscatter networks. In these networks, the backscatter signals are highly sensitive to both the distance and angle of tags, leading to significant RSS variation across different receiving antennas. Fig. 5.11(a) illustrates the RSS variation for signals received by 16 antennas in an array. The spatial imbalance is evident, with the largest RSS difference among antennas reaching up to 15 dB. Notably, the first antenna records the second lowest RSS. If the weakest antenna is chosen as the reference, there is a risk that the offsets will not be fully counteracted. An extreme scenario is when the chosen reference antenna fails to receive the backscatter signal at all, potentially causing a loss of power in the desired direction of the spatial spectrum. To address this spatial disequilibrium, we propose an updated definition of relative power as follows:

$$P(\alpha) = \left| \sum_{b=0}^{K-1} \sum_{k=0, k\neq b}^{K-1} w(k,\alpha) \cdot \frac{S_{G_k}}{S_{G_b}} \right| \tag{5.11}$$

which traverses all antennas and uses them as references one by one to avoid power

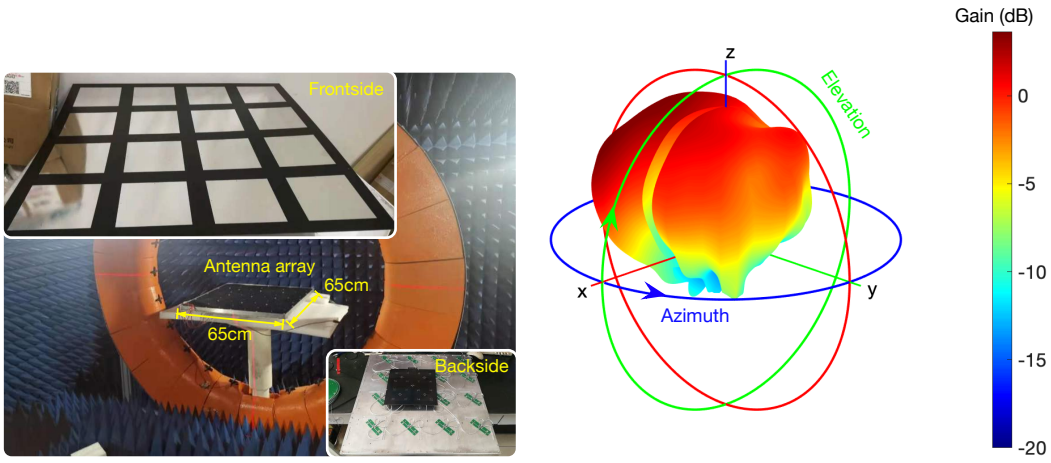
loss. We call this approach *cyclic redundancy reference*, namely, each antenna is selected as the reference in turn.

• **Temporal imbalance**. Additionally, we observe a temporal imbalance in the received signals, which is attributable to the operation of Automatic Gain Control (AGC). AGC is a feature in amplifiers that uses a closed-loop feedback system to maintain a consistent output signal amplitude, despite variations in the input signal amplitude. It dynamically adjusts the amplifiers' gain based on either the average or peak signal level at the output, thereby significantly improving the success ratio of decoding. However, a side effect of AGC is that it alters the phase offsets each time the gain is adjusted, leading to a temporal imbalance in the phase. Fig. 5.11(b) shows how the estimated phase of the received CW signal is influenced when the AGC automatically adjusts the gain at different levels. To address this temporal imbalance, it is necessary to compensate for the phase changes induced by AGC adjustments. Typically, an AGC system operates at several predefined levels, and the chipset includes a feedback loop to detect the current AGC level. By measuring these phase offsets beforehand and establishing a correlation between the offsets and AGC levels, we can create a map for this purpose. Then, during operation, we can use the AGC level to accurately compensate for the phase variation.

5.5 Implementation

The configuration of our gateway prototypes and the experimental system is outlined as follows:

• **Antenna layer**: Our antenna array is composed of 4×4 antennas. We use the microstrip technique to fabricate the antenna array on a printed circuit board (PCB) to avoid signal attenuation on the board. The PCB is composed of a substrate of RT/duroid 5880. The model of high-speed RF switches is HMC241 [102] from Analog



(a) Antenna array (b) Directivity

Fig. 5.12: The design of 4×4 antenna array

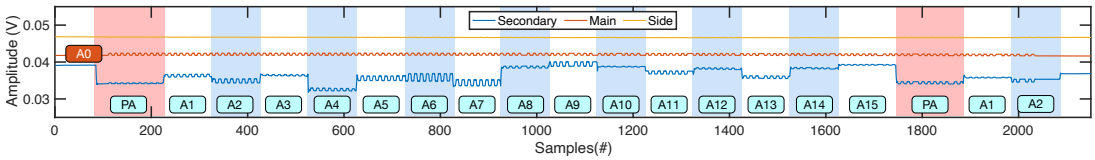
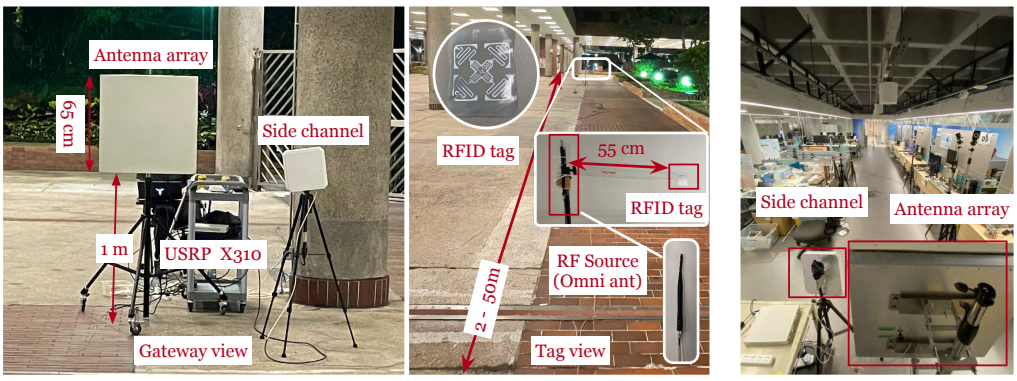


Fig. 5.13: Baseband signals acquired through the three channels. We utilize the preamble (PA) segment for scheduling start identification. Following PA, signals are divided into 15 equal segments, each corresponding to antennas $A_1 \sim A_{15}$.

Devices. The size of the antenna array is 65×65 cm^2, each of which is 12×12 cm^2. Fig. 5.12 shows the antenna array deployed in an RF chamber and its measured gain in all directions. Specifically, each array element has 0.5 dBi gain, 0.5 dB flatness, and $\pm 45^\circ$ dB beamwidth.

• **Front-end layer**: We use a USRP X310 software-defined radio (SDR) from NI [76] to build the front-end. The SDR contains four stand-alone I/O interfaces, which are used to build the main, secondary, and side channels. The main channel connects to the antenna A_0 and continuously receives the backscatter signal for decoding purposes. The side channel equipped with a directional antenna acquires the pure CW signal for building AFNC. In the secondary channel, five single-pole four-throws (SP4T) RF switches bridge antennas $A_1 \sim A_{15}$ to a shared ADC for saving cost. At any moment,



(a) Outdoors (b) Indoors

Fig. 5.14: Experimental scenarios

only a single antenna element can be put through. In this manner, each element can receive a segment of the entire packet. The switching delay is negligible compared with the ms-level packet duration. Fig. 5.13 shows the baseband signals acquired concurrently by the three channels from an RFID tag's `RN16` reply. In the secondary channel, the received signal is sliced into segments through the amplitude changes. The preamble (PA) segment with a larger window is used to determine the beginning of the schedule. After PA, 15 equal segments are received by the antennas $A_1 \sim A_{15}$ respectively. The RF front-end has a bandwidth of 100 MHz, which covers the entire UHF of 860 $\sim$ 960 MHz. The backend runs at a high-performance PC equipped with an Intel CPU Xeon E5-2620.

• **RF source & backscatters**: We use another USRP X310 equipped with an omnidirectional antenna as the RF source, shown in Fig. 5.14. Commercial ImpinJ H47 RFID tags equipped with Monza 4QT chips are employed as the backscatter elements [103]. Specifically, the RF source transmits a single-tone CW to activate tags and broadcasts `Query` commands to bootstrap the communication. To avoid the mutual interaction, we use the `Select` command for acquiring the `RN16` replies only without `ACK`, as used in [104]. In a single-tag scenario, our system is capable of localizing the tag up to 100 times per second. When expanding the setup to include

multiple tags (specifically N tags), the system employs the `SELECT` command to sequentially poll and localize each tag. This approach allows for the localization of each tag at a rate of around $100/N$ times per second.

5.6 Evaluation

We evaluate the performance of the gateway prototype by conducting comprehensive experiments. Due to the indoor space limit, we mainly conduct the benchmark outdoors, as shown in Fig. 5.14(a). Since the power provided by an RF source distributed unevenly in the space, the localization might fail if a tag cannot be activated steadily in some space. In this case, we have no idea about whether the resulted inaccuracy is caused by either the activation failures or the localization errors. Thus, we deploy the tag very close to the RF source (i.e., 55 cm) to ensure that the tag is definitely powered up outdoors. We will discuss the opposite scenario that tag is powered far away at 10 m in the case study. The power of the RF source is fixed at 1 watt. The range is defined as the distance from the RF source to the gateway via the tag. We gather the ground truth by using OptiTrack [105] equipped with 12 infrared cameras.

5.6.1 Performance of Phase Estimators

First, we evaluate performance of CPE and CPE+, which targets at estimating the pseudo-phase. In the experiment, we move the tag away from the antenna array with a distance. We perform 17 experimental trails, in which the distance is set to $2, 3, \cdots, 9, 10, 15, \ldots, 45, 50$ m successively. In each setting, the RFID tag remains stationary and transmits about 3000 `RN16` packets during 30 seconds, by each of which phase estimators output a phase value.

RSS vs. Distance. First of all, we evaluated the RSS of the signal collected through the main channel, focusing on how it varies with distance. The results, presented in

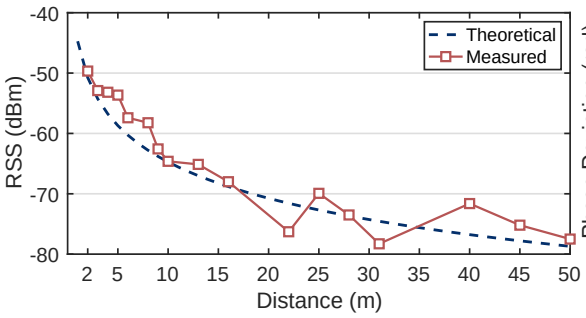


Fig. 5.15: RSS vs Distance

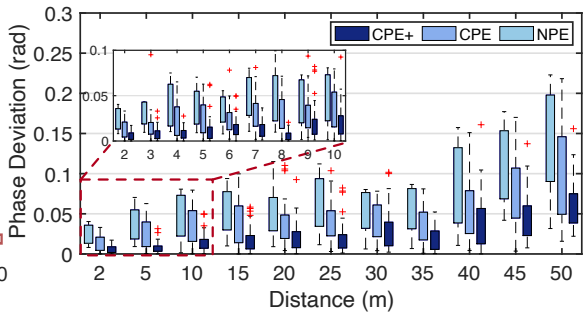


Fig. 5.16: Phase Noise vs. Distance

Fig. 5.15, indicate a decrease in RSS from -40 dBm to -80 dBm as the distance increases. Additionally, the data reveal the shadowing effect of the RF signal at various distances. Despite these observations, the measured signal aligns well with the established models of classical communication channels. Given that the sensitivity threshold of a USRP receiver is -90 dBm, we noted that decoding backscatter signals becomes increasingly challenging beyond 50 meters. Consequently, distances greater than 50 meters were not included in our analysis.

Phase Noise vs. Distance. The primary objective of our research is phase estimation, for which we randomly select an antenna from the array and use CPE, CPE+, and NPE to compute the pseudo-phase. The π-ambiguity is eliminated in NPE by using the method proposed in [57]. CPE+ means CPE+M1+M2, where M3 is not considered here because it is used to equilibrate the power distribution in a spatial spectrum and is invalid for phase stability. The ground truth of the phase value is hardly measured due to the presence of unknown offsets (like ϕ_R and ϕ_G) and multipath effect. Instead, we compute the deviations of the pseudo-phase to evaluate the performance of estimators indirectly. We will use the accuracy of AoA and localization to reflect the phase measurement accuracy indirectly in the following.

Fig. 5.16 plots the deviation distribution of the phase as the function of distance. From the figure, we observe the following findings: (1) The mean deviation (i.e., std) is raised as the increased distance regardless of which estimator is adopted. The phase deviation is caused by the phase noise. The increased distance reduces the

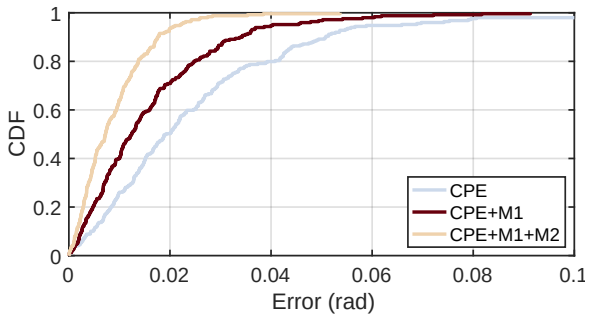


Fig. 5.17: Effect of denoising measures

RSS and conversely enhances the influence of phase noise as we analyzed before; (2) Our practice suggests that a cm-level localization accuracy requires the phase std to be lower than 0.01 radians. The NPE, CPE, and CPE+ meet this criterion when the distance is less than 3 m, 5 m, and 10 m, respectively, which demonstrates that CPE+ can well depress the noise at a further distance than others.

Effect of Denoising Measures. Next, we evaluate the performance two denoising measures introduced in our paper. In the experiment, we fix a tag at a distance of 10 meters. Fig. 5.17 shows the CDFs for phase deviation when employing CPE, CPE with measure M1 (CPE+M1), and CPE with both measures M1 and M2 (CPE+M1+M2). The results showed median phase deviations of 0.021 radians, 0.013 radians, and 0.007 radians for CPE, CPE+M1, and CPE+M1+M2, respectively. The 90th percentile deviations for these setups were 0.052 radians, 0.039 radians, and 0.019 radians, respectively. Overall, measures M1 and M2 contributed to a reduction in phase noise by 57% and 43%, respectively.

Compared with COTS Reader. We also evaluated the phase deviation of CPE+ in comparison with a standard Commercial Off-The-Shelf (COTS) RFID reader, specifically the Impinj Speedway R420, which is widely utilized in RFID localization systems [106]. The RFID tag was positioned at a 5m distance. Fig. 5.18 shows the CDFs of phase deviation for both CPE+ and the Impinj R420. CPE+ achieves an median phase deviation of 0.016 rad (with the 10^{th} percentile at 0.002 rad and the 90^{th} percentile at 0.045 rad). In contrast, the Impinj R420 recorded a median phase

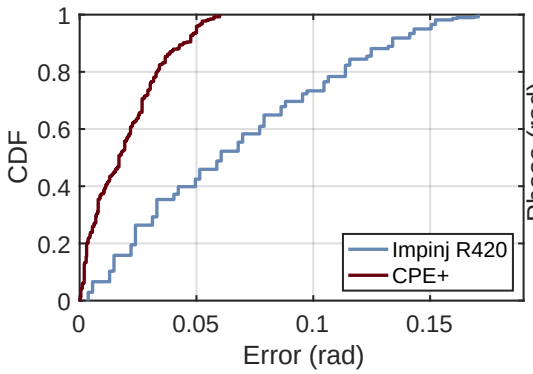


Fig. 5.18: CPE+ vs. COTS reader

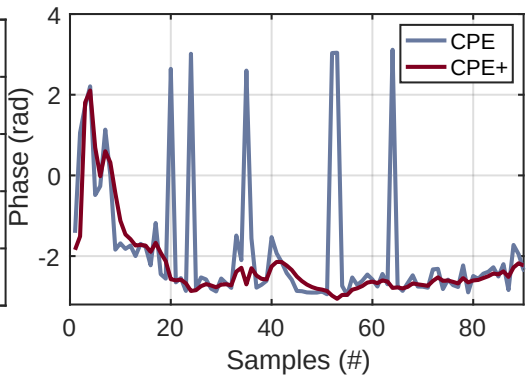


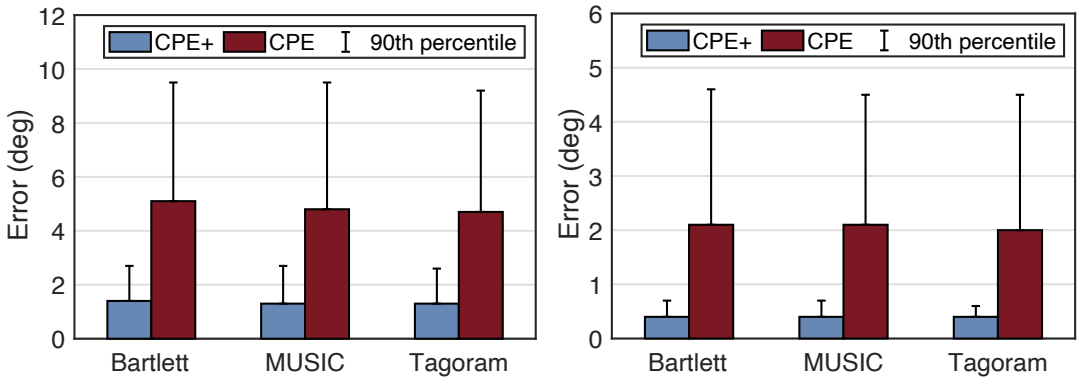
Fig. 5.19: Impact of tag motion

deviation of 0.061 rad (with the 10^{th} percentile at 0.013 rad and the 90^{th} percentile at 0.134 rad). The result indicates that the phase stability of CPE+ is approximately 3.8× greater than that of conventional COTS readers, owing to the implementation of our proposed denoising methods.

Impact of Tag Motion. Finally, we evaluated the stability of phase measurements in CPE+ during RFID tag movement. The robot, with the tag attached, followed a straight path at a speed of 1 m/s. Fig. 5.19 displays the phase samples captured using both CPE and CPE+. It is evident that the phase estimated by CPE+ is notably more stable than that of CPE, a benefit arising from the implemented denoising measures. This enhanced stability in phase estimation is advantageous for tracking tags in motion. In the following section, we conduct a detailed quantitative analysis of how the speed of tag movement impacts localization accuracy.

5.6.2 Accuracy of AoA

Second, we evaluate the accuracy of AoA computed using the estimated phase. In the experiment, the tag is placed on the vertical plane 25 meters away from the gateway's antenna array. We place the tag at different angles and estimate the AoA along the direct path. The ground truth is calculated on the basis of the true physical locations of the tag and the antenna array, which are measured by OptiTrack.



(a) Azimuthal error (b) Elevation error

Fig. 5.20: AoA estimation error vs. Algorithms

Impact of AoA Estimation Algorithms. To assess the generality of CPE+ across different algorithms, we evaluated the accuracy of AoA estimation using both CPE and CPE+. Our experiment involved three main algorithms: Bartlett [107], MUSIC, and Tagoram. We conducted the test in a controlled environment with minimal multipath effects to focus on the influence of CPE+ on these algorithms, thereby eliminating other variables. Fig. 5.20 compares the azimuthal and elevation angle errors across these algorithms. With CPE+, the median azimuthal AoA errors for Bartlett, MUSIC, and Tagoram were 1.4°, 1.3°, and 1.3°, respectively. In contrast, using CPE resulted in median errors of 5.1°, 4.8°, and 4.7°, which are approximately 3.6× larger. A similar trend was observed for elevation angles, where CPE+ achieved a median error of 0.4° for all algorithms, significantly lower than the 2.1° median error with CPE, which is approximately 5.3× larger. These results lead to two conclusions: (1) CPE+ is versatile and effective across different AoA estimation algorithms, and (2) CPE+ significantly enhances AoA estimation accuracy by effectively reducing phase noise.

Impact of Multipath. We also evaluate the performance of CPE+ in environments affected by multipath interference. Fig. 5.21 plots the CDFs of the AoA errors using the pseudo-phase estimated by both CPE and CPE+ (enhanced with measures M1, M2, and M3). From the data, we made the following observations: (1) In the absence

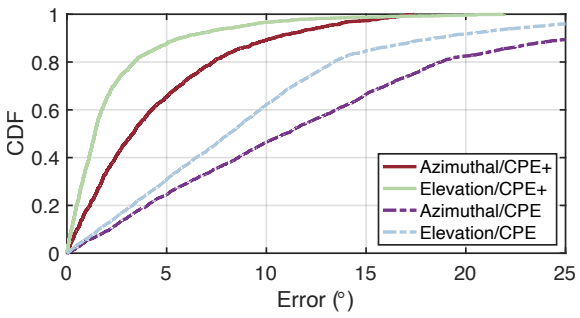


Fig. 5.21: Impact of multipath effect

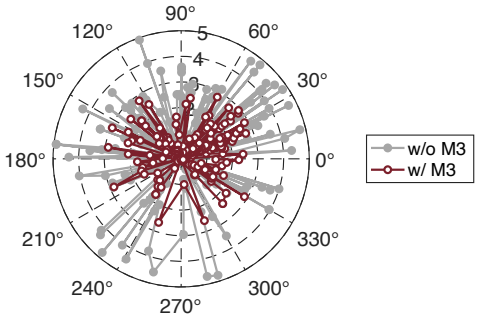


Fig. 5.22: Impact of M3

of denoising measures, the median AoA errors for azimuthal and elevation angles are 11.2° and 8.1°respectively, with the 90th percentile errors reaching 25° and 17.9°. (2) If the three measures are adopted, then the median errors are 3.3° and 1.4°, and the 90 percentile errors are 10.3° and 5.6° in the two angles, respectively. We see 70% and 82% error reduction in the estimation. Clearly, the quality of phase plays a pivotal role in the AoA computation.

Impact of M3. We particularly compare the azimuthal errors with and without M3 in Fig. 5.22. The purpose of M3 is to address the spatial and temporal equilibria. In the figure, the median errors of azimuthal angles with/without M3 are 1.09° and 1.67° respectively. The M3 can help improve the accuracy by 34.7%. In addition, we also observe large fluctuations in the results without M3. For example, the error is raised to 5° at the azimuthal angle of 110° without M3. This is mainly caused by spatial disequilibrium, which can be restored by the technique of cyclic redundancy reference. By contrast, the error is reduced to 2° at that angle of 110°.

5.6.3 Accuracy of Localization

Next, we evaluate the accuracy of localization. The deployment of our system is illustrated in Fig. 5.23, with three gateways positioned equidistantly along a circle's perimeter. The distance is defined as the mean value between the scene center and the

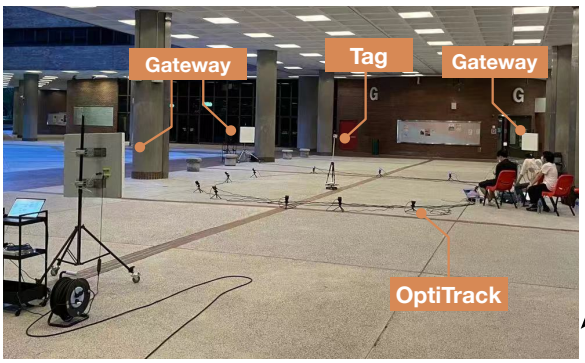


Fig. 5.23: The deployment of system

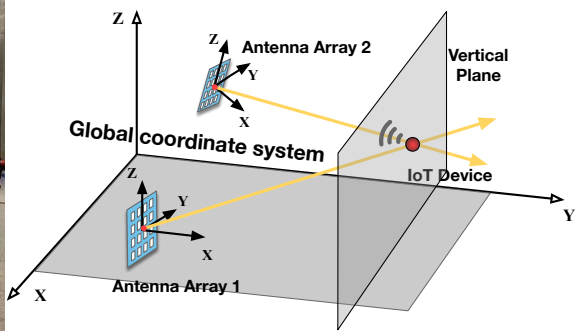


Fig. 5.24: Triangulation algorithm

three gateways. The OpiTrack system is used for ground truth collection. As shown in Fig. 5.24, when a single gateway is used, the tag is localized on the 2D vertical plane at some distance from the antenna array. When employing multiple gateways (2G, 3G), the tag's location is determined as the point that minimally distances itself from the rays, which are directed toward the anticipated AoA from each gateway. A Kalman Filter is employed to enhance result stability during tag tracking.

Impact of Distance. Our initial evaluation focuses on how distance affects localization accuracy. Fig. 5.25 plots the median location errors as a function of the distance. In the figure, 1G, 2G, and 3G are short for 1, 2, and 3 gateways. From the figure, we have the following findings: (1) our system can achieve ultra-high precision lower than 10 cm regardless of how many gateways are used if the distance is less than 10 m. (2) Given one gateway (1G), the error starts to increase rapidly when the distance is over 10 m or equivalently, the RSS reduces to lower than -70 dBm (see Fig. 5.15). Since then, the phase noise dominates the phase error. Due to the "butterfly effect", the location error is eventually amplified to a high level. This effect is more evident at a further distance. Even so, CPE+ prolongs the range of ultra-high precision from 3 m to 15 m compared with CPE. (3) Surely, more gateways can further help reduce the errors. For example, the error at 50 m is reduced from 58 cm to 30 cm by using two additional gateways (3G) compared with a single gateway (1G).

Impact of Velocity. In this experiment, we evaluate how the tag movement speed

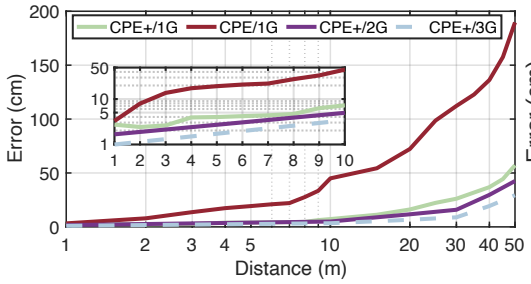


Fig. 5.25: Accuracy in localization

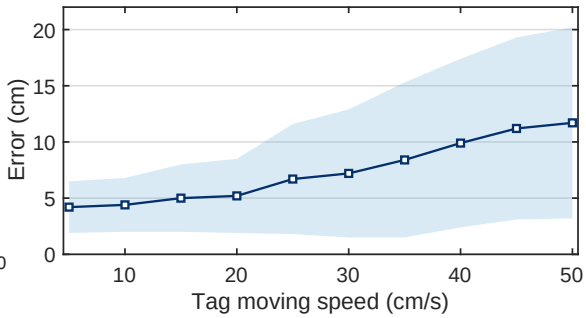


Fig. 5.26: Impact of velocity

influences tracking precision. Utilizing three gateways positioned 5 meters apart, tags were tracked as they moved linearly over a 3-meter distance, with speeds adjusted from 5 to 50 cm/s in 5 cm/s increments. The Optitrack system provided ground truth data. As depicted in Fig. 6.8, localization accuracy remains relatively stable at speeds below 20 cm/s, showcasing a peak median accuracy of 5.2 cm and a standard deviation of 3.3 cm. However, beyond this threshold, we observed a gradual increase in tracking errors, culminating in a median accuracy of 11.7 cm and a standard deviation of 8.5 cm at 50 cm/s. A primary factor for the observed increase in localization error at higher velocities is the Doppler shift. The relative motion between the tag and gateways will change the RF signal frequency, significantly amplifying localization errors at increased velocities.

5.6.4 Case Study

Finally, we would like to quantitatively test the high precision of the gateway in tracking a moving object (i.e., a toy train) in our office. As shown in Fig. 5.14(b), our office is full of different types of indoor reflectors, including tables, chairs, computers, and metal obstacles. In our tracking experiments, a gateway is hung horizontally to the 3 m high ceiling and set up a train track on the ground in a 2×2 m^2 small area. The tag, powered by an RF source located 10 meters away, resulted in a total distance of approximately 13 meters. First, we visually present our tracking results.

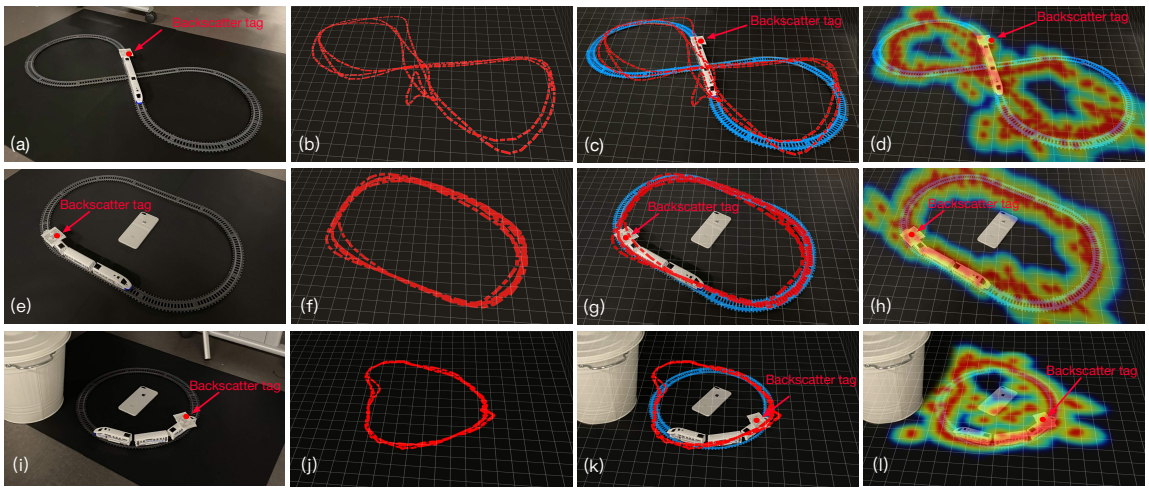


Fig. 5.27: Real-World Applications. The first column shows the real scenarios where an RFID tag is attached to a toy train moving along different tracks; the second column shows the tracked locations of the tag output by our system; the third column compares the output results with the real tracks; the fifth column shows the heat map of trajectory, which fuses all computed spatial spectrums into a single one.

Fig. 5.27 shows the multiple demos for various tracking tasks, organized into three rows representing three distinct scenarios: case I utilizes an "8-shaped" track, case II features a squashed rectangle track, and case III employs a circular track. Particularly, we demonstrate the multipath effect in case III, where a metal object is placed close to the tag on the top left corner. The visual result suggests high positioning accuracy, as indicated by the distinct and unentangled trajectories. Quantitatively, our results show median localization accuracies of 9.7 cm for case I, 8.2 cm for case II, and 6.3 cm for case III, demonstrating the system's efficacy across various configurations.

Chapter 6

Understanding Localization by a Tailored GPT

6.1 Motivation

In recent years, there has been an emerging interest in the construction of precise RF-based indoor localization systems that supplement the final hundred meters where GPS lacks reach. Numerous strides [10, 42, 73, 108–122] have been conducted for this purpose in the past two decades, culminating in a rather promising level of accuracy. High-precision indoor localization has the potential to unlock a plethora of key applications encompassing indoor navigation, augmented reality, location-conscious pervasive computing, advertising, and social networking, among others. As a result, the practice of tracking IoT devices within structural enclosures has emerged as an expanding commercial interest.

However, only a small fraction of the proposed solutions have successfully transitioned to real-world implementations due to unforeseen challenges encountered across different deployment scenarios, some of which are discussed here. (1) *Hardware diversity*: the heterogeneity in hardware arising from circuitry specifications can lead to unde-

sirable errors in RF signal measurements, thereby skewing the input to a localization algorithm with several unknown offsets. (2) *Spatial diversity*: the RF signal experiences spatial fluctuations due to uneven electromagnetic field distribution. (3) *Multipath effect*: amidst an unpredictable, intricate, and dynamic wireless landscape, RF signals suffer reflection from a multitude of static or mobile obstacles. This multipath effect has become a formidable barrier to indoor localization, especially when the localization algorithm is highly reliant on line-of-sight (LoS) propagation.

Indoor localization essentially boils down to solving a non-linear optimization problem, that is, finding the optimal position at which the RF device can generate signals that closely match those received by the base stations (see §3.2). This challenge squarely falls within the realm of machine learning. In the wake of the deep learning (DL) surge, some studies [9, 84, 123–125] have begun to exploit DL advancements to confront these challenges. These works have successfully showcased that DL surpasses traditional localization algorithms in terms of accuracy and stability. The success of previous attempts has inspired our investigation into the potential of DNNs, while addressing the following two main limitations:

- Current methods are largely reliant on supervised learning, in which the efficacy is intimately tied to the volume and quality of training samples (i.e., accurate location labels). To this end, existing methods require another set of high-precision optical localization systems, such as OptiTrack [84] or Lidar [45], to amass the training datasets. This necessity adds significant complexity to deployment. Additionally, the lack of high-quality, large-scale datasets specifically tailored for this purpose further exacerbates the challenges in the community.
- Current methods face the challenge of non-transferability. In particular, the training dataset is intimately tied to the specific layout of a scene, which means that the value of datasets collected in other environments or at earlier times drastically diminishes. This limitation hampers the universal deployment of a trained model across diverse environments, thus impeding scalability. This constrained adapt-

ability represents a significant drawback in existing approaches, highlighting the demand for techniques with enhanced versatility across multiple scenarios.

To address the above limitations, we propose the Transformer-based localization (TBL) model, distinguished by its proficiency in contextual understanding and ability to discern relationships between sequential elements. Our model is distinct from previous works [126, 127], which utilized individual components of the Transformer model (e.g., encoders or decoders). Instead, our study exploits the full potential of the Transformer architecture, aiming to understand both the contextual scenes and the intricate relationships between phase measurements and positions. To unleash the generalizability of the TBL model across scenes, we introduce NeRF^2, a tailored variant of the Generative Pre-training Transformer (GPT) model for localization. NeRF^2, which serves as the pre-training version for the TBL model, is pre-trained on 1.4 million localization data samples. Like linguistic models such as ChatGPT that generate text from provided contexts, NeRF^2 "generates" locations by processing contextual RF signals and previously determined positions. Specifically, we make the following efforts:

• Focusing on antenna array-based triangulation, we present an inventive hierarchical neural network architecture adapted from the Transformer model. The architecture comprises multiple A-Subnetworks and a singular T-Subnetwork. Each A-Subnetwork ingests phase measurements acquired from a specific antenna array as tokens, generating Angle-of-Arrival (AoA) results and directional features. The T-Subnetwork integrates the combined feature vectors along with the historical positions of the target device to generate its current position. The A-Subnetworks and T-Subnetwork are built using Transformer encoders and decoders, respectively. This arrangement harmonizes ideally with the core architecture of the original Transformer.

• Second, we propose a semi-supervised training methodology. Past DL-based localization systems [9, 84, 123–125] require the use of absolute location labels to guide the

neural network's training in a supervised manner. As previously mentioned, this strict requirement for label acquisition necessitates the use of a secondary high-precision localization system, thereby increasing deployment costs and operational complexity. In this work, we introduce two novel loss functions that are designed to minimize the discrepancy between the actual and predicted distances of two sampled locations. Without the need for absolute position labels, the network model can predict two locations whose distances closely align with the ground truth. This subtle modification allows the collection of RF signals using a pair of fixed target devices with a known separation.

• Finally, the micro-benchmark results compellingly demonstrate that our proposed Transformer model consistently outperforms other state-of-the-art solutions, with improvements ranging between 30% and 70%. In light of these promising outcomes, we are excited to release a pre-training version of the Transformer-based localization (TBL) model, LocGPT, which is equipped with 36 million parameters. As a robust initial model for localization, LocGPT can be effectively fine-tuned for a specific scene using less than 50% of the sample data ordinarily required. This design has the potential to significantly lower the barriers to adoption, making high-accuracy indoor localization more accessible across a wide range of applications and environments.

Summary. Our main contributions are fourfold. First, we repurpose the Transformer model to achieve the antenna array-based triangulation. Second, we propose a semi-supervised training methodology to reduce deployment costs and operational complexity. Finally, we release the first pre-training model LocGPT to achieve widespread transfer learning in the indoor localization domain.

6.2 Overview

Antenna arrays, widely utilized for high-capacity communication processes such as MIMO and beamforming, are increasingly being repurposed for indoor localization by many prominent standard organizations. Examples include the enhancements seen in Wi-Fi 802.11 az and Bluetooth 5.1 or above versions. The primary advantage of using antenna arrays lies in their capability to amplify signal strength and enhance localization precision by harnessing spatial diversity and the directionality of signal propagation. Following this trend, our proposed system employs planar antenna arrays to accurately locate wireless terminals, encompassing RFID tags, Bluetooth labels, Wi-Fi devices, and beyond, in 3D space. Our design is a versatile solution that transcends the constraints of specific wireless technologies.

Antenna arrays, commonly used in high-capacity communications, such as MIMO and beamforming, are now being adapted for indoor localization by leading standard organizations, as seen in Wi-Fi 802.11 az and Bluetooth 5.1+. Their strength lies in enhancing signal strength and localization accuracy through spatial diversity and directional signal propagation. In line with this, our system uses antenna arrays to pinpoint various wireless terminals, offering a technology-agnostic solution.

We also harness the power of the Transformer to address the problem of indoor localization. The Transformer, composed of encoders and decoders, is originally designed for natural language processing (NLP) tasks, such as language translation. It has now become the de facto standard model for NLP that is widely adopted for many applications, such as ChatGPT [128] and LLaMA [129]. On a broader scale, our entire system operates analogously to the task of sequence-to-sequence language translation. The Transformer encoders ingest a sequence of phase measurements (recorded by antennas), whereas the Transformer decoders convert this sequence to a sequence of positions. Toward this objective, we first introduce how the Transformer has been repurposed to serve our localization task in §6.3. Then, we demonstrate the feasibility

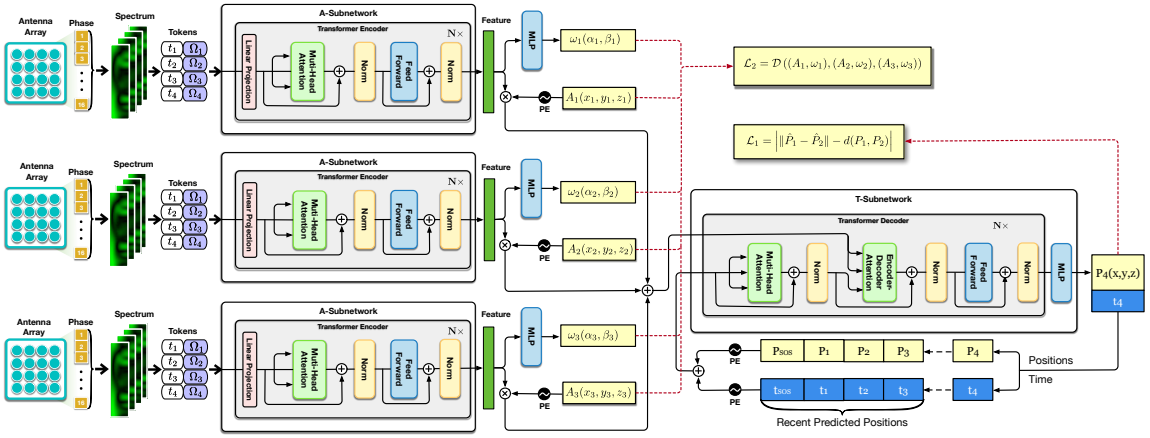


Fig. 6.1: TBL Network Architecture. It consists of two layers of neural networks. The first layer is the A-Subnetwork assigned to each base station with the purpose of estimating the AoA relative to the current antenna array. The second layer is the T-Subnetwork aiming to pinpoint the final position by feeding the AoA feature from three A-Subnetworks.

and effectiveness of the proposed model via a comprehensive micro-benchmark in §6.4 across the high-quality dataset (see §3.3). Finally, we introduce the first pre-training Transformer model, LocGPT, in §6.5 and evaluate it in §6.6.

6.3 Transformer-based Localization

This section provides a detailed account of the repurposed Transformer for the localization task. The approach is termed "Transformer-based Localization" (TBL).

6.3.1 Network Architecture

As illustrated in Fig. 7.5, the architecture of our model is underpinned by the Transformer neural network, which mainly consists of two types of subnetworks: the AoA subnetwork (A-Subnetwork) and the triangulation subnetwork (T-Subnetwork). **(1) A-Subnetwork**: Each A-Subnetwork is dedicated to fit Eqn. 3.21 by accepting the phase values collected by an antenna array. An A-Subnetwork is realized by exploiting the powerful capabilities of the Transformer encoders, which are renowned for

their proficiency in discerning intricate patterns and hierarchically structured representations within the input data. The model architecture accommodates an arbitrary number of A-Subnetworks, each corresponding to a distinct base station. These A-Subnetworks operate independently of one another, transforming their respective inputs into an AoA and corresponding directional features. These outputs then collectively serve as the input to the T-Subnetwork. **(2) T-Subnetwork**: Equipped with Transformer decoders, the T-Subnetwork collates the directional and feature vectors provided by the A-Subnetworks to yield a precise estimate of the target's location. Namely, it is designed to fit Eqn. 3.25 mathematically. Owing to the sequence generation capabilities of the Transformer decoders, the T-Subnetwork also effectively incorporates historical location data, thereby refining the accuracy of location predictions. In the following, we delve into a detailed examination of each subnetwork.

6.3.2 A-Subnetwork

Inspired by the Transformer encoder stack, the A-Subnetwork is designed to estimate the AoA of the RF signals that are transmitted from the target device. More formally, the i^{th} A-Subnetwork denoted by $\mathcal{A}_i$ is defined as follows:

$$\mathcal{A}_i : \left(A_i, \{ \underbrace{(t_j, \Phi_j)}_{\text{Current}}, \underbrace{(t_{j-1}, \Phi_{j-1}), \ldots, (t_{j-M}, \Phi_{j-M})}_{\leqslant M \text{ recent measurements}} \} \right) \rightarrow (\vec{\omega}_i, \mathcal{F}_i) \tag{6.1}$$

where A_i denotes the location of the i^{th} station, and (t_j, Φ_j) represents the phase values $\Phi_j = \{\phi_j^1, \phi_j^2, \cdots, \phi_j^K\}$, which are measured by the station equipped with K antennas at the time t_j. The subnetwork takes the required current phase measurement Φ_j and optional recent historical measurements, namely $\Phi_{j-1}, \Phi_{j-2}, \ldots, \Phi_{j-M}$. The total number of measurements does not exceed $M + 1$. The direction vector $\vec{\omega}_i$ indicates the estimated AoA, while $\mathcal{F}_i$ is the feature vector that is utilized by the subsequent subnetwork. Next, we elaborate on the subnetwork.

Input Representation

Each A-Subnetwork processes a spatial spectrum. We favor the use of spatial spectrum over raw phase values as input due to several key considerations. First, fluctuations in the antenna array size integrated into the base station (e.g., 3×3 or 4×4) can lead to inconsistent input dimensions. Second, the device's operational frequencies may vary (e.g., 800MHz or 2.4GHz), such that even the sample phase value reflects the different distances traversed by the RF signals at varying frequencies. Finally, using raw phase values as inputs fails to consider the structure of the antenna array. For instance, 16 phase values could be measured by arrays of 1×16, 4×4, and 2×8, but this structural information would be lost in the phase sequence. These variations in hardware configurations could undermine the model's universality.

(1) Spectra as Tokens: In contrast, spatial spectra are closely linked with the hardware configurations. Regardless of the hardware setup, the spatial spectra of a consistent size can be generated using Eqn. 3.20. The size of the spatial spectrum depends solely on the granularity with which we traverse the spatial domain. Thus, we partition the space into 36×10 directions with a granularity of 10° to strike a balance between resolution and computational load. Directions with elevation angles less than 10° should be disregarded due to the limited directionality of directional antennas, which is typically less than 80°. Hence, the spatial spectrum size is standardized to 36×9, which we subsequently reshape into a 324-dimension vector that serves as a token. This can be formally expressed as:

$$\Omega = \Big[\mathcal{P}(0^\circ, 0^\circ), \mathcal{P}(0^\circ, 10^\circ), \cdots, \mathcal{P}(350^\circ, 70^\circ), \mathcal{P}(350^\circ, 80^\circ)\Big] \tag{6.2}$$

where $\mathcal{P}(\alpha, \beta)$ denotes the normalized relative power at the direction $\omega(\alpha, \beta)$ (refer to Eqn. 4.11).

Contrary to previous works [84,90,130] that transform spectra into images and employ CNN or AutoEncoder for feature extraction, we opted to preserve a whole spectrum

as a single token form for three primary reasons. First, image recognition is a highly time-intensive process. Second, our experience suggests that the partitioning of images (for instance, ViT patches) causes disorganization within the spectrum, leading to training misconvergence. Finally, an image-centric approach tends to fall into the approaches of signature-based localization, in which an image feature corresponds to a specific location, making it extremely susceptible to environmental factors.

(2) Learning from History: It is important to note that phase measurements from the same object are intricately interconnected over time. Given that historical measurements may help eliminate transient interference, we thus assemble the current measurement and M preceding ones into a token sequence that are fed into the subnetwork. Prior measurements are not mandatory. In cases where past measurements are lacking, their corresponding tokens can simply be set to zero. Nevertheless, the maximum number of tokens fed into the subnetwork is set to M. The measurement time could be seen as the "position" of a measurement within the sequence. To incorporate this aspect, we encode the time using the positional encoding scheme as follows.

$$\mathrm{PE}(x) = \left[\sin\left(2^{0/L}\pi x\right), \cos\left(2^{0/L}\pi x\right), \ldots, \sin\left(2^{(L-1)/L}\pi x\right), \cos\left(2^{(L-1)/L}\pi x\right)\right] \tag{6.3}$$

Originally devised to encode positions within the Transformer model, $\mathrm{PE}(x)$ has evolved to augment the dimension of any number x from one to $2L$. Today, it has been widely used across various DL contexts. In our scenario, we first encode the measurement time using $\mathrm{PE}(t)$ into a 324-dimensional vector, which is then added to the token. Hence, the input for an A-Subnetwork at time t_j can be formally represented as:

$$\mathbf{I}(t_j) = \left[\Omega_j + \mathrm{PE}(t_j - t_j), \Omega_{j-1} + \mathrm{PE}(t_j - t_{j-1}), \ldots, \Omega_{j-M} + \mathrm{PE}(t_j - t_{j-M})\right]$$

where the time is expressed relative to the current time t_j. In addition, $\Omega_j, \ldots, \Omega_{j-M}$ are the spectra obtained at time $t_j, \ldots, t_{j-M}$, i.e., $t_j > t_{j-1} > \cdots > t_{j-M}$. The

input consistently comprises $M + 1$ tokens. Missing tokens are filled with zeros. Our evaluation advises $M = 5$ in practice.

Discussion: In our approach, positional encoding is applied exclusively to the synchronized timestamps of data from each base station. The spatial spectra derived from each base station naturally serve as a form of "spatial encoding" for phase, as depicted in Section 2.1. Therefore, we omit additional positional encoding for spatial spectra.

Network Body

Each A-subnetwork consists of multiple Transformer encoders. The encoder is a key building block in the Transformer architecture, primarily designed for processing sequential data in NLP tasks. This encoder operates through multiple layers, each consisting of two main components: a self-attention mechanism and a position-wise feed-forward neural network. The self-attention mechanism allows the model to weigh the relevance of each element in a sequence relative to all other elements, thus capturing dependencies regardless of their distance in the sequence. The position-wise feed-forward networks, which are applied independently to each position, involve two linear transformations with a GELU activation in between. Layer normalization and residual connections are employed around both the self-attention and feed-forward sub-layers to stabilize the learning process. Then, these layers are stacked to construct the complete Transformer encoder, which transforms input data into a higher-level feature representation that captures complex patterns within the data.

Network Output

Unlike CNNs, the Transformer retains the dimensionality of the input to ensure consistency and compatibility between layers. This means each layer's input and output dimensions remain the same. As we feed in $M + 1$ tokens, the outputs are $M + 1$

feature vectors with 324 dimensions, denoted by $\mathcal{F}_i$. An additional Multilayer Perceptron (MLP, a two-layer fully connected network) is appended to decode the features into a directional vector $\vec{\omega}_i(\alpha, \beta)$, which represents the estimated AoA. Considering the location of the station A_i, the direction can be formulated as a ray:

$$\vec{l_i} = A_i + \vec{\omega_i} \cdot u = A_i + \text{MLP}(\mathcal{F}_i) \times u \tag{6.4}$$

where u is the distance between the A_i and a point on the ray. Given three A-Subnetworks, we finally obtain three rays, namely, $\vec{l_1}$, $\vec{l_2}$, and $\vec{l_3}$.

6.3.3 T-Subnetwork

Our model incorporates a single T-Subnetwork to compute the final position using the resulting AoAs. A T-Subnetwork denoted by $\mathcal{T}$ can be formally expressed as follows:

$$\mathcal{T} : \Big(\big(A_1, \mathcal{F}_1(t_j)\big), \big(A_2, \mathcal{F}_2(t_j)\big), \big(A_3, \mathcal{F}_3(t_j)\big) \Big) \rightarrow P_j(x, y, z) \tag{6.5}$$

where $P_j(x, y, z)$ is the 3D location of the wireless terminal at time t_j. One might question the necessity for the T-Subnetwork, given that the previous three A-Subnetworks have already provided AoA directions. Theoretically, we could compute the intersection of these three rays using a geometric approach, if they intersect at all. However, our observations reveal that these three rays rarely intersect at a single point in 3D space, as shown in Fig. 6.2(a). In such scenarios, the goal is to identify a point closest to the three rays, thereby transforming the problem into another optimization task (see Eqn. 3.25). Notably, recent location estimations can greatly aid in current predictions, especially when tracking a moving target. Therefore, we must devise a strategy to incorporate this motion context into our model. In response to this, employing another neural network, referred to as the T-Subnetwork, has demonstrated its effectiveness in tackling the issues.

Input Representations

As aforementioned, the network should consider two factors: the directional rays and the historically estimated locations. Thus, we have two types of inputs for the T-Subnetwork, which we discuss below.

(1) Input I: Triangulation determines the intersection of the three rays, each originating from a known location and representing a direction. This necessitates the inclusion of both the base station's location $A_i(x_i, y_i, z_i)$ and the directional features $\mathcal{F}_i(t_j)$ into the T-Subnetwork. Specifically, the station's location A_i is first encoded using $\mathrm{PE}(\cdot)$ and then multiplied with the feature $\mathcal{F}_i(t_j)$. Finally, the updated feature vectors from the three A-Subnetworks are combined into a single vector, which is then inputted into the T-Subnetwork. This can be formally represented as:

$$\mathbf{I}_1(t_j) = \mathrm{PE}(A_1) \otimes \mathcal{F}_1(t_j) + \mathrm{PE}(A_2) \otimes \mathcal{F}_2(t_j) + \mathrm{PE}(A_3) \otimes \mathcal{F}_3(t_j) \tag{6.6}$$

where $\mathrm{PE}(A_i) = [\mathrm{PE}(x_i), \mathrm{PE}(y_i), \mathrm{PE}(z_i)]$. Here, x_i, y_i, and z_i are enhanced to a 108D vector using the $\mathrm{PE}(\cdot)$ and subsequently merged into a singular 324D vector. The symbol $\otimes$ stands for the Hadamard product. This approach ensures the input is standardized, regardless of the number of deployed base stations.

(2) Input II: Given that the positions of a target are intrinsically interrelated, the T-Subnetwork input additionally comprises M historical position tokens along with a "start-of-sequence" (SOS) token, specifically employed to initiate the decoding procedure. Similarly, we treat the time as the "position" within the sequence context when the target is found at P_j. Both are encoded using $\mathrm{PE}(x)$ and then combined to form an embedding token. Thus, the second input can be represented as

$$\mathbf{I}_2(t_j) = [\mathrm{PE}(t_j - t_{j-1}) + \mathrm{PE}(P_{j-1}), \cdots, \mathrm{PE}(t_j - t_{\mathrm{SOS}}) + \mathrm{PE}(P_{\mathrm{SOS}})] \tag{6.7}$$

with the phase values measured at time t_j, wherein we are predicting the position of

the device at time t_j. Both the time and location are expanded to a 324D vector. Hence, every element in $\mathbf{I}_2$ is a 324D token, which matches the dimensionality in the A-Subnetwork. The consistent dimensionality is particularly relevant during the cross-attention phase where the decoder attends to the encoder's outputs. The second input contains a maximum of $M + 1$ tokens. Similarly, the history is not mandatory. In cases where we are first localizing the object, the initial context P_{SOS} and t_{SOS} can be set to zero. Even for the stationary object, the historical context could help reduce potential errors.

Network Body

The T-Subnetwork is composed of multiple Transformer decoders, which are instrumental in sequence generation tasks. Each decoder layer comprises three principal components: self-attention, encoder-decoder attention, and a position-wise feed-forward network. (1) The self-attention mechanism allows each token in the input sequence to focus on other tokens within the same sequence, which helps in understanding the sequence's context. As depicted in Fig. 7.5, the input $\mathbf{I}_2(t_j)$ is fed into the self-attention block, which attends to the historical trajectory. Once a position is predicted by the T-Subnetwork, the output position is added to $\mathbf{I}_2(t_{j+1})$, contributing to the tracking context. This mode is often referred to as "autoregressive". (2) The encoder-decoder attention allows every position in the decoder to attend to all features in the input sequence captured by the encoder, helping the decoder concentrate on the input's relevant parts. In the figure, input $\mathbf{I}_1(t_j)$ is integrated into the encoder-decoder attention block. (3) The output of features from the decoder is directed through an MLP for the final positions prediction.

Output

The final output of the T-Subnetwork represents the ultimate spatial location of the wireless terminal at time t_j. This is achieved through the synergistic operations of both the subnetworks, which analyze spatial spectrum data and execute a sequence of historical location estimations to determine the terminal's position in the space. To enable autoregressive learning, the current predicted position P_j will be appended to $\mathbf{I}_2(t_{j+1})$ for the next prediction. If the total number of tokens is greater than $M+1$, the oldest predictions are removed.

6.3.4 Semi-Supervised Training

Next, we introduce a novel semi-supervised training approach along with the loss functions.

(1) Position Loss: All past DL-based indoor solutions, such as DLoc [9] and iArk [84], adopt the supervised training approach, i.e., providing the absolute location labels and using the distance between the predictions and true location labels for the loss function. In particular, their loss functions are defined as follows:

$$\mathcal{L}_0 = \|\widehat{P}_j - P_j\| \tag{6.8}$$

where $\widehat{P}_j$ and P_j are the predicated location and the ground truth, respectively. This approach requires us to acquire the absolute locations using a second high-precision positioning system, such as the OptiTrack or Lidar. However, such a cumbersome deployment mode prevents the spread of DL-driven localization. This limitation motivates us to find new loss functions that do not require absolute position labels.

(2) Distance Loss: Our first loss involves the distance between two sampled locations. We design a Y-shaped handle with two terminals fixed at the two heads at a known distance D, as shown in Fig. 6.2(b). As this handle is maneuvered in space,

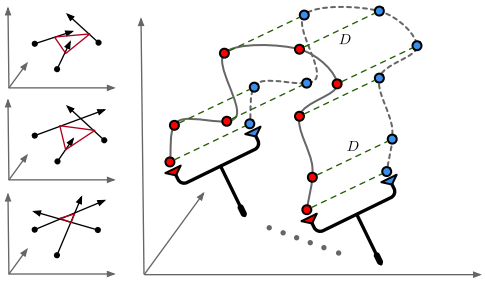


Fig. 6.2: Intersection and Dataset Collection. (a) shows the different intersection cases by three rays. (b) shows our proposed dataset collection approach where two wireless terminals are fixed at the two heads of the Y-shaped handle with a known constant distance.

base stations collect two RF signals, each from a terminal. The data are then converted into two spatial spectra and processed by the model. Consequently, the model can predict two results, $\widehat{P}_j^1$ and $\widehat{P}_j^2$, for the two devices at time t_j, respectively. Then, we can define the distance loss as follows:

$$\mathcal{L}_1 = \left| \|\widehat{P}_j^1 - \widehat{P}_j^2\| - D \right| \tag{6.9}$$

This subtle shift obviates the necessity for recording the absolute locations of the two sampled points. This approach greatly simplifies the process of data collection by eliminating the need for additional positioning systems.

(3) Variance Loss: Prior solutions tend to emphasize minimizing $\mathcal{L}_0$, often overlooking the variability in the resultant errors. When it comes to triangulation, this variability manifests as the area of intersection of the three rays. Ideally, these three rays would converge at a single point. Yet, they often form a triangle, within which the wireless device might be located. Therefore, it is ideal to minimize the area of this error triangle as much as possible. With this perspective in mind, we propose another loss function that is intended to reduce the area of the intersection region. This issue might be amplified in three dimensions as the three rays might not intersect at all, as shown in Fig. 6.2(a). In such instances, computing the area is unfeasible. Thus,

we use the perimeter instead for the loss:

$$\mathcal{L}_2 = \mathcal{D}(\vec{l_1}, \vec{l_2}) + \mathcal{D}(\vec{l_2}, \vec{l_3}) + \mathcal{D}(\vec{l_1}, \vec{l_3}) \tag{6.10}$$

where $\mathcal{D}(\cdot)$ signifies the distance of two rays. The rays of $\vec{l_1}$, $\vec{l_2}$, and $\vec{l_3}$ are the outputs of the A-Subnetwork (see Eqn. 6.4). The minimization of distances can bring these rays closer together in a manner that they are near-intersecting or intersect within a small region, which can be considered an approximation of an intersection point under practical circumstances. It is often sufficient for indoor localization applications.

The $\mathcal{L}_2$ is solely back-propagated into the A-Subnetworks for parameter adjustment without affecting the T-Subnetwork. This characteristic can further instruct the A-subnetworks in the acquisition of AoA knowledge. Previous studies [84, 90, 130] have reported the use of triangulation-based deep learning with the inclusion of CSI images or spatial spectra as inputs. Regrettably, these studies lack a mechanism to verify the model's triangulation fitting. Most of the time, the model functions similarly to a fingerprint-based localization, outputting a location given an image resembling a previously learned one. Our novel loss function, solely back-propagated to the A-subnetwork, guarantees that the produced features are truly relevant to the AoA.

(4) Joint Loss: A joint loss function can be formulated to combine the two loss components, i.e., the deviation between predicted and true distances and the area of intersection region. The overall loss function can be written as follows:

$$\mathcal{L}_3 = \lambda \mathcal{L}_1 + (1 - \lambda)\big(\mathcal{L}_2^1 + \mathcal{L}_2^2\big) \tag{6.11}$$

where $\mathcal{L}_1$ is the distance loss, and $\mathcal{L}_2^{1(2)}$ are the variance loss for the two devices' predicted positions $P^{1(2)}$, respectively. λ is a hyperparameter that determines the relative weighting of the two loss terms and can be tuned for optimal performance.

Discussion: Compared to methods that employed Lidar or OptiTrack, this semi-

supervised approach substantially reduces both the cost and complexity of training dataset collection. Since the model never receives absolute position labels as input, it develops its own global coordinate system. The coordinates generated by the model need to be aligned with the real-world coordinate system through a coordinate transformation, which can be achieved by positioning several anchors at known locations.

6.4 Micro-Benchmark

In this section, we evaluate the TBL solution using the collected datasets, focusing on the errors in individual scenes.

Experimental Setup.

We construct individual models for each distinct setting without addressing the transferability of these models. Thus, a total of 50 TBL models are trained for the corresponding 50 scenarios. In Section 6.4.4, we evaluate various model configurations and ultimately adopt a structure comprising two standard encoders for the A-Subnetwork and two decoders for the T-Subnetwork in micro-benchmark. Within this setup, each attention layer features eight heads, each with 64 dimensions. This architecture results in a total of approximately 12 million parameters for the entire model. In the training process, we split the collected dataset into segments in the time domain, each containing 10 samples to facilitate learning from historical data. The samples in each segment are fed into the model sequentially for training or testing. We allocate 80% of segments for training purposes and reserve the remaining 20% for testing. During the training process, the joint loss with $\lambda = 0.05$ is employed, in which the $\mathcal{L}_2$ is sole-propagated to the encoder. During model training, we maintain a batch size of 4096 and subject each dataset to 5,000–7,000 iterations of training. Our model is trained on a server equipped with an AMD 3955WX processor, 64 GB of RAM, and two NVIDIA 4090 GPUs. The training process in each setting takes approximately

Table 6.1: Accuracy of RFID Localization (mean errors in cm)

#	TBL (w/ history)	TBL (w/o history)	iArk	DLoc
S01	8.2±6.4	10.9 (24.8% ↑)	13.2 (37.9% ↑)	23.4 (65.0% ↑)
S02	9.7±7.1	13.5 (28.1% ↑)	31.7 (69.4% ↑)	32.6 (70.2% ↑)
S03	18.3±10.8	25.1 (27.1% ↑)	46.4 (60.6% ↑)	37.7 (51.5% ↑)
S04	27.4±16.4	33.7 (18.7% ↑)	70.9 (61.4% ↑)	46.1 (40.6% ↑)
S05	33.7±19.8	47.9 (29.6% ↑)	91.2 (63.0% ↑)	56.6 (40.5% ↑)
S06	37.5±34.9	62.5 (40.0% ↑)	99.1 (62.2% ↑)	63.5 (40.9% ↑)
S07	38.1±24.6	51.2 (25.6% ↑)	110.7 (65.6% ↑)	63.2 (39.7% ↑)
S08	55.2±31.3	79.0 (30.1% ↑)	118.2 (53.3% ↑)	85.0 (35.1% ↑)
S09	60.8±36.0	88.2 (31.1% ↑)	133.1 (54.3% ↑)	87.3 (30.4% ↑)
S10	35.8±27.3	57.4 (37.6% ↑)	167.1 (78.6% ↑)	78.1 (54.2% ↑)
S11	41.9±28.4	52.4 (20.0% ↑)	179.0 (76.6% ↑)	72.6 (42.3% ↑)
S12	43.5±29.1	63.7 (31.7% ↑)	91.2 (52.3% ↑)	88.2 (50.7% ↑)
S13	52.9±34.1	65.3 (19.0% ↑)	108.5 (51.2% ↑)	101.2 (47.7% ↑)
S14	39.3±26.6	60.5 (35.0% ↑)	114.0 (65.5% ↑)	76.3 (48.5% ↑)
S15	10.2±6.8	16.4 (37.8% ↑)	60.5 (83.1% ↑)	13.2 (22.7% ↑)
S16	14.2±9.7	23.8 (40.3% ↑)	38.8 (63.4% ↑)	29.6 (52.0% ↑)
S17	14.5±10.1	20.2 (28.2% ↑)	36.0 (59.7% ↑)	41.4 (65.0% ↑)
S18	7.0±3.7	9.9 (29.3% ↑)	29.1 (75.9% ↑)	13.2 (47.0% ↑)
S19	5.1±2.4	9.1 (44.0% ↑)	31.2 (83.7% ↑)	9.7 (47.4% ↑)
S20	5.9±2.7	8.9 (33.7% ↑)	25.3 (76.7% ↑)	11.3 (47.8% ↑)
S21	9.7±5.6	8.9 (9.0% ↓)	19.8 (51.0% ↑)	9.4 (3.2% ↓)
S22	8.1±6.9	13.7 (40.9% ↑)	33.2 (75.6% ↑)	15.3 (47.1% ↑)
S23	4.4±3.1	8.2 (46.3% ↑)	18.6 (76.3% ↑)	9.4 (53.2% ↑)
S24	6.6±4.3	9.9 (33.3% ↑)	16.4 (59.8% ↑)	13.6 (51.5% ↑)
S25	10.8±7.6	13.0 (16.9% ↑)	16.7 (35.3% ↑)	23.7 (54.4% ↑)
S26	8.6±5.0	10.7 (19.6% ↑)	14.2 (39.4% ↑)	13.2 (34.8% ↑)
S27	7.8±5.4	13.1 (40.5% ↑)	20.5 (62.0% ↑)	22.2 (64.9% ↑)
S28	14.7±9.5	16.3 (9.8% ↑)	15.7 (6.4% ↑)	21.5 (31.6% ↑)
Mean	**22.5±14.8**	**31.9 (28.9% ↑)**	**62.5 (60.7% ↑)**	**41.4 (45.5% ↑)**

Table 6.2: Accuracy of Wi-Fi Localization (mean errors in cm).

#	TBL (w/o history)	TBL (w/ history)	iArk	DLoc
S29	23.8±14.2	29.1 (18.2% ↑)	74.8 (68.2% ↑)	63.2 (62.3% ↑)
S30	21.2±12.5	28.1 (24.6% ↑)	73.2 (71.0% ↑)	61.8 (65.7% ↑)
S31	21.8±13.1	30.4 (28.3% ↑)	68.2 (68.0% ↑)	59.5 (63.4% ↑)
S32	20.3±12.9	28.2 (28.0% ↑)	71.5 (71.6% ↑)	58.8 (65.5% ↑)
S33	22.4±14.0	30.5 (26.6% ↑)	73.9 (69.7% ↑)	60.4 (62.9% ↑)
S34	16.4±8.5	19.4 (15.5% ↑)	54.2 (69.7% ↑)	37.2 (55.9% ↑)
S35	11.2±6.3	14.7 (23.8% ↑)	48.7 (77.0% ↑)	32.1 (65.1% ↑)
S36	14.1±7.7	18.0 (21.7% ↑)	45.2 (68.8% ↑)	36.6 (61.5% ↑)
Mean	**18.9±11.1**	**24.8 (23.3% ↑)**	**63.7 (70.5% ↑)**	**51.2 (62.8% ↑)**

Table 6.3: Accuracy of BLE Localization (mean errors in cm)

#	TBL (w/o history)	TBL (w/ history)	iArk	DLoc
S37	19.8±10.4	24.2 (18.2% ↑)	35.9 (44.8% ↑)	30.5 (35.1% ↑)
S38	37.5±25.2	43.2 (13.2% ↑)	58.2 (35.6% ↑)	54.9 (31.7% ↑)
S39	39.2±27.9	47.7 (17.8% ↑)	63.5 (38.3% ↑)	58.0 (32.4% ↑)
S40	7.0±4.8	11.5 (39.1% ↑)	10.3 (32.0% ↑)	27.0 (74.1% ↑)
S41	6.6±2.7	10.3 (35.9% ↑)	9.9 (33.3% ↑)	24.4 (73.0% ↑)
S42	6.6±5.0	8.1 (18.5% ↑)	27.8 (76.3% ↑)	19.9 (66.8% ↑)
S43	5.1±4.7	7.8 (34.6% ↑)	26.2 (80.5% ↑)	21.7 (76.5% ↑)
S44	32.5±22.2	36.5 (11.0% ↑)	65.2 (50.2% ↑)	48.3 (32.7% ↑)
S45	36.2±24.8	40.1 (9.7% ↑)	68.7 (47.3% ↑)	51.6 (29.8% ↑)
S46	50.2±45.8	65.8 (23.7% ↑)	80.0 (37.2% ↑)	103.8 (51.6% ↑)
S47	52.9±47.7	67.3 (21.4% ↑)	84.5 (37.4% ↑)	105.2 (49.7% ↑)
S48	101.1±51.9	107.0 (5.5% ↑)	122.6 (17.5% ↑)	137.8 (26.6% ↑)
S49	72.3±40.5	90.5 (20.1% ↑)	114.6 (36.9% ↑)	97.5 (25.8% ↑)
S50	45.9±34.7	56.2 (18.3% ↑)	82.1 (44.1% ↑)	75.5 (39.2% ↑)
Mean	**36.6±24.9**	**44.0 (20.5% ↑)**	**60.7 (43.7% ↑)**	**61.2 (46.1% ↑)**

4–6 hours each. We accomplished the complete training tasks in 11 days.

6.4.1 Accuracy

The accuracy of our model is gauged by the position error, which is conceptualized as the Euclidean distance between the predicted and actual positions. For a comparative analysis, we adopt the SOTA solutions, namely, iArk [84] and DLoc [9] – as benchmark references. Using spatial spectra in image format, both approaches leverage ResNet coupled with MLP to determine the positions. Particularly, DLoc integrates an additional ResNet-based decoder for consistency verification. In contrast, the Transformer ingests spatial spectra as tokens, which may include or exclude historical data. To ensure unbiased comparisons, input spatial spectra dimensions are consistently set at 36×9. Both iArk and DLoc were trained using the same dataset and under the same train/test sets settings as TBL. The outcomes for the three technologies are tabulated in Table 6.1, Table 6.2, and Table 6.3, respectively. In the table, the red arrow (↑) indicates an increase in percentage error, reflecting a

decrease in performance compared to the TBL with historical context. Conversely, the green arrow (↓) represents a reduction in percentage error, indicating improved performance relative to the TBL with historical context. Overall, the TBL model consistently outperforms the other solutions across all three datasets, regardless of the inclusion of historical data. Given the space constraints, we focus solely on the RFID results for analysis. The findings from the other two datasets indicate similar results. Specifically, we observe the following:

• First, the TBL reduces errors by 60.7% and 45.5% on average when compared with iArk and DLoc, respectively. An outlier (9.4 cm vs. 8.9 cm) is evident in S21. This marginal 3.2% deviation lies within the standard deviation.

• Second, the errors of the TBL without historical context are derived from the localization errors of the first samples in the test segments, where no historical context is available. The TBL results with historical context cover the cases with the context from 1 to 10 samples. The results demonstrate that the inclusion of past data enhances accuracy by an estimated 28.9%.

• Third, a pronounced correlation exists between accuracy and the RSS. This correlation is evident in the results from the S01 to S11 scenarios, all gathered within the same environment but at varying distances. As the range increases (specifically, between 5 and 55 m), the RSS weakens, spanning from -62.5 dBm to -88.6 dBm, which in turn elevates the mean error values, ranging from 8.2 cm to 41.9 cm.

• Fourth, the best outcomes are discerned in S23, S19, and S20. These results are attributable to their augmented densities (e.g., 22,627 samples per m^3) and reduced distances (e.g., 7 m). Conversely, the least accurate outcomes are recorded in segments S8–S11, stemming from notably feeble signals (e.g., -88.6 dBm) and extended distances (e.g., 40 m).

• Finally, comparable configurations within identical scenes (e.g., S18–S22 within Scene E and S24–S28 within Scene G) display closely aligned errors (e.g., deviations

of 22.6% and 29.4%, respectively). This finding suggests that localization precision is predominantly influenced by scene layouts rather than specific settings, such as station placements.

Discussion. The superior performance of TBL when compared with SOTA methods can be attributed to several key factors. Primarily, the self-attention mechanism allows the model to intelligently weigh signals across various time intervals. This is further enhanced by the contextual processing capability of Transformers, which processes data in parallel, fostering a deeper understanding of inter-measurement relationships. Moreover, the model's hierarchical feature abstraction strategy, where encoders in the A-Subnetwork are tasked with AoA estimation and decoders in the T-Subnetwork focus on triangulation, facilitates a refined analysis of environmental dynamics. The integration of historical context aids in mitigating instantaneous interference and noise, ensuring consistent outcomes. Lastly, the introduction of a joint loss function integrates the distinct roles of encoders and decoders, culminating in enhanced localization precision.

6.4.2 Ablation Study

Next, we investigate the influence of various loss functions on localization accuracy. We measure errors using three distinct loss functions: $\mathcal{L}_0$ (Eqn. 6.8), $\mathcal{L}_1$ (Eqn. 6.9), and the joint loss $\mathcal{L}_3 = \mathcal{L}_1 + \mathcal{L}_2$ (Eqn. 6.11). In our subsequent experiments, we primarily utilize the S03 dataset, chosen for its characteristic base station deployment distances typical of indoor localization scenarios, unless specified otherwise. For the ablation study, to ensure fairness, we maintain consistent hyper-parameters across various loss functions. The respective CDFs of these errors are presented in Fig. 7.12. Leveraging absolute location labels results in a reduced mean error of 16 cm (the 90th percentile: 45.7 cm) relative to other setups. The accuracy of joint loss is about 24% lower than $\mathcal{L}_0$. This is anticipated, given that supervised learning typically exhibits

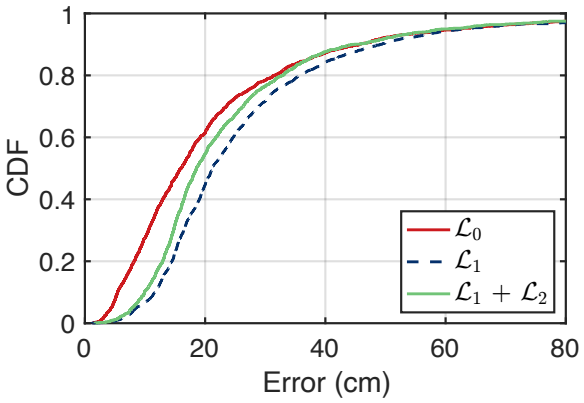


Fig. 6.3: Ablation Study

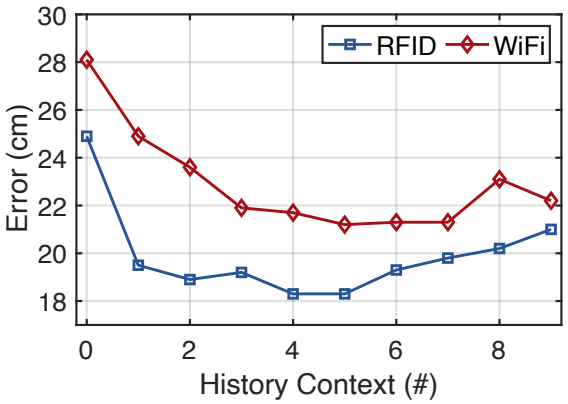


Fig. 6.4: Historical Context

superior performance over semi-supervised techniques. Nevertheless, this comes at the cost of deploying an additional high-precision localization system for acquiring labels. On the other hand, the median errors for $\mathcal{L}_1$ and $\mathcal{L}_1 + \mathcal{L}_2$ stand at 21.2 cm (90th percentile: 49.2 cm) and 18.6 cm (90th percentile: 44.6 cm), respectively, suggesting that incorporating variance loss can reduce errors by approximately 13%.

6.4.3 Impact of the Historical Context

We further analyze the impact of the historical context by focusing on two chosen scenes: S03 (RFID) and S30 (Wi-Fi). During the experiments, we compute the average errors incorporating M historical contexts, i.e., encompassing M past spatial spectra for the A-Subnetwork and M location outcomes for the T-Subnetwork, where M ranges from 0 to 9. As shown in Fig. 6.4, the findings reveal a compelling trend: errors decrease notably when incorporating $5-6$ historical context steps. This finding underscores the value of leveraging historical data, which can help in counteracting the sporadic signal fluctuations instigated by transient interferences, such as moving entities. Interestingly, there is an uptick in errors for $M > 6$, possibly due to the error accumulation. Thus, a setting of $M = 5$ is recommended.

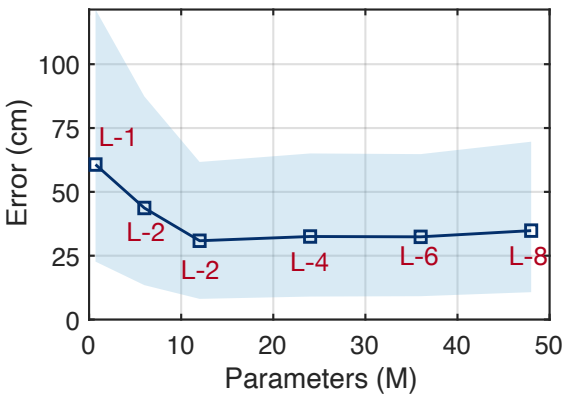


Fig. 6.5: Model Variant

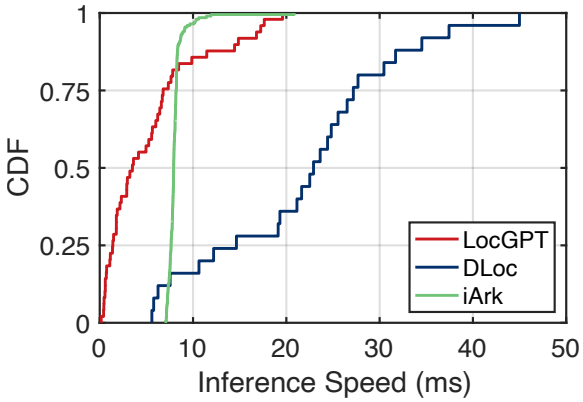


Fig. 6.6: Inference Speed.

6.4.4 Model Variant

In our default configuration, we employ two encoder layers for the A-Subnetwork and two decoder layers for the T-Subnetwork. This experiment aims to assess the performance of various architectural variants. Specifically, we explore configurations with one layer (with two 8D heads each), two layers (with two 8D heads each), two layers (with eight 64D heads each), four layers (with eight 64D heads each), six layers (with eight 64D heads each), and eight layers (with eight 64D heads each). The respective parameter counts for these configurations are 0.8, 6, 12, 24, 36, and 48 M. The results are shown in Fig. 6.5. The six configurations achieve mean errors of 60.7, 43.6, 30.9, 32.5, 32.4, and 34.8 cm, respectively. We also observe a linear reduction in error corresponding to parameter counts less than 10 M. Yet, the error plateaus when the parameter count ranges from 10 to 50 M. This observation is in accordance with the emergence phenomenon or phase transition in large AI models—namely, a pronounced behavioral shift not anticipated from training the system at smaller scales. Without supplementing with additional training data, increasing model size may not necessarily yield enhanced accuracy. Thus, 10 M parameters are advised for the TBL for an individual scenario.

6.4.5 Inference Speed

Finally, we evaluate the inference speeds of TBL, iArk, and DLoc. For a fair comparison, all models are implemented within the PyTorch framework and evaluated using the S03 dataset to measure inference times. These evaluations are conducted on the same computational setup as detailed in the experimental setup section, ensuring consistency across all tests. The CDFs of the inference time for each approach are shown in Fig. 6.6. As can be seen, their median inference time stands at 3.5 ms, 7.9 ms, and 22.9 ms, respectively. Notably, TBL demonstrates superior speed compared with the others. This efficiency can be attributed to TBL's treatment of spatial spectra as tokens, rather than the more resource-intensive image inputs used by iArk and DLoc. Furthermore, the latter two rely on computationally demanding CNNs for processing. Additionally, the inherent parallel processing capabilities of the Transformers contribute to TBL's faster response time.

6.5 LocGPT: Pre-training model

Prior micro-benchmark has demonstrated both the feasibility and efficacy of implementing the Transformer model for localization purposes. In this section, we address the challenge of non-transferability.

6.5.1 Pre-Training

RF signal propagation is highly dependent on environmental specifics, meaning models trained in one setting may not perform well in others, thus limiting their generalizability. This challenge requires the recollection of large volumes of data whenever a new scene is encountered. Inspired by the remarkable success of GPT in the field of NLP, we propose LocGPT to tackle the generalizability issue. LocGPT is a pre-trained

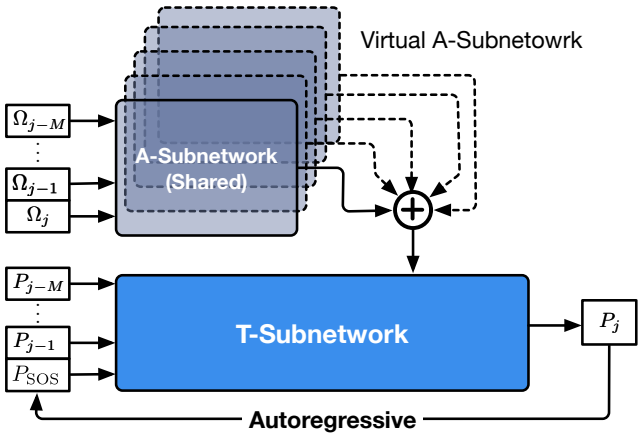


Fig. 6.7: Architecture of LocGPT

TBL model with millions of high-quality data. When encountering a distinct scene, LocGPT is capable of being fine-tuned with a limited amount of data, thereby fitting effectively into the new dynamics.

The network architecture of LocGPT, shown in Fig. 6.7, is almost the same as that of the previously proposed TBL model, except for the number of A-Subnetworks. In the previous architecture, each base station is designated an A-Subnetwork. However, the exact number of base stations deployed can vary depending on the specific setting. To deal with the scalability, we train only a single A-Subnetwork, term as the "parent A-Subnetwork", during the pre-training phase. Doing this allows all trained base stations to share parameters from this singular A-Subnetwork. In contrast to our micro-benchmark setup, for pre-training LocGPT, we employ 1.4 million localization data points. The initial configuration of a two-encoder/decoder model is inadequate for addressing generalization challenges. Through empirical analysis, we determined that a configuration comprising six layers in both the encoder and decoder subnetworks strikes an optimal balance between convergence speed and generalization capabilities, with each layer featuring eight heads of 64 dimensions. Consequently, the total amount of parameters of LocGPT is about 20M, effectively reduced from 36M through parameter sharing of A-Subnetwork. While models with an increased number of encoder/decoder layers exhibit marginally enhanced transferability, they

also require extended training durations. LocGPT is pre-trained using all datasets, excluding S02, S13, S20, S27, S31, S32, S33, S35, S36, S37, S40, S46, and S50. These datasets are reserved for assessing the model's transfer learning performance. This selected dataset enables us to evaluate LocGPT's adaptability across various technologies, including RFID, Wi-Fi, and BLE, and to showcase its transfer learning potential both within identical scenes under varying scenarios and across completely distinct scenes. LocGPT is trained with $\mathcal{L}_0$ loss. The hardware setup for pre-training is the same as that used for the micro-benchmark. The Adam optimizer with a cosine learning rate schedule is employed, ranging from $5e^{-4}$ to e^{-6}. We conducted the pre-training within 5 days.

6.5.2 Fine-Tuning

During the fine-tuning phase, the shared A-Subnetwork is further specialized into distinct A-Subnetworks tailored for individual base stations. The rationale behind this design is to harness the generalized capabilities of the primary A-Subnetwork, which captures universal AoA estimation features and then adapts these features to the specificities of each base station. This model can be fine-tuned through three predominant techniques: Full fine-tuning, LoRA, and Adapter.

(1) Full Fine-tuning: This approach updates all 36M parameters of the pre-trained model based on the target task's dataset. It is a comprehensive method that adjusts the entire model to the new data.

(2) LoRA: Leveraging a recalibration strategy, LoRA modifies the activations of the pre-trained model at each layer. Instead of revising all parameters, LoRA recalibrates existing model knowledge to better fit the new tasks. In this method, about 0.9M new parameters are updated [131].

(3) Adapter: This method embeds small, specialized adapter modules between the model's layers. Rather than adjusting all the parameters, only these adapters, con-

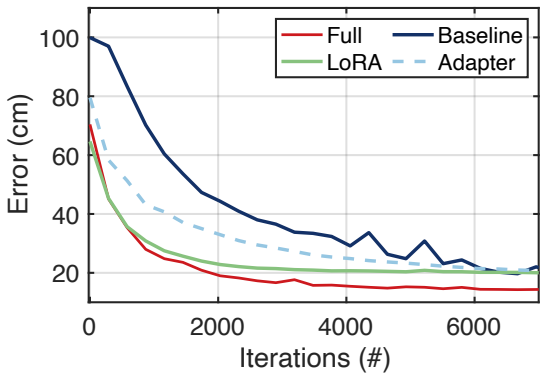


Fig. 6.8: Convergence Efficiency

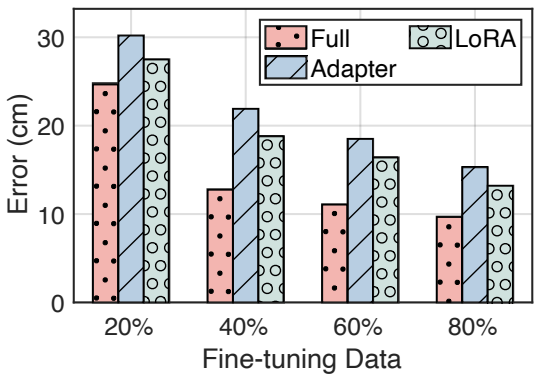


Fig. 6.9: Accuracy

sisting of 0.8M parameters, undergo training, while the primary model parameters stay frozen.

6.6 Evaluation

This section evaluates the advancement of LocGPT and the performance of various fine-tuning techniques.

6.6.1 Convergence Efficiency

We selected the S02 dataset to fine-tune LocGPT, allocating 40% for training and 60% for testing. We measured fine-tuning efficiency by counting iterations (each processing 4096 samples) and tracking mean error. After each iteration, we assess the mean error using the test set. For comparison, we also trained a TBL model from scratch as the baseline, bypassing the benefits of the pre-trained model. The results are depicted in Fig. 6.8. The ideal trajectory for these curves is towards the bottom-left corner, which would indicate achieving lower errors in fewer iterations. The graph clearly shows that leveraging a pre-trained model accelerates this convergence, regardless of the specific fine-tuning approach used. Specifically, to reach a mean error of 25cm, full fine-tuning, LoRA, adapter, and baseline methods took 1450,

2030, 4350, and 5220 iterations, respectively. Thus, in terms of convergence speed, the ranking from slowest to fastest is the baseline, adapter, LoRA, and full fine-tuning. Furthermore, this experiment demonstrates that training deep models from scratch can be computationally intensive and time-consuming. By starting with a pretrained model, one can achieve faster convergence, thus saving computational resources and time.

6.6.2 Accuracy

Next, we employ varying portions of the S02 dataset for fine-tuning, specifically 20%, 40%, 60%, and 80%, while maintaining a consistent 20% for testing. The comparative accuracies of the three fine-tuning techniques are depicted in Fig. 6.9. As can be seen, the trends remain consistent across all four cases, with full fine-tuning consistently achieving the lowest mean errors, followed by the adapter, and then LoRA. For instance, when utilizing 60% of the dataset, the resultant mean errors for the three techniques are 11.1, 18.5, and 16.4 cm, respectively. Evidently, in terms of precision, full fine-tuning stands out as the superior method compared with the others. The advantage of full fine-tuning can be attributed to two main reasons. First, full fine-tuning can provide a good balance between leveraging the pre-trained knowledge and fitting the new data. This can lead to a model that generalizes well to new, unseen data. Second, unlike other methods that add additional components (like adapters) or adjust activations (e.g., LoRA), full fine-tuning does not introduce new architectural components. Instead, it utilizes the existing architecture, which often has been optimized for performance during pre-training.

6.6.3 Transfer Learning

Finally, we assess the transfer learning capabilities of LocGPT in new environments using a full fine-tuning method, termed "FT-40," where it was fine-tuned with 40%

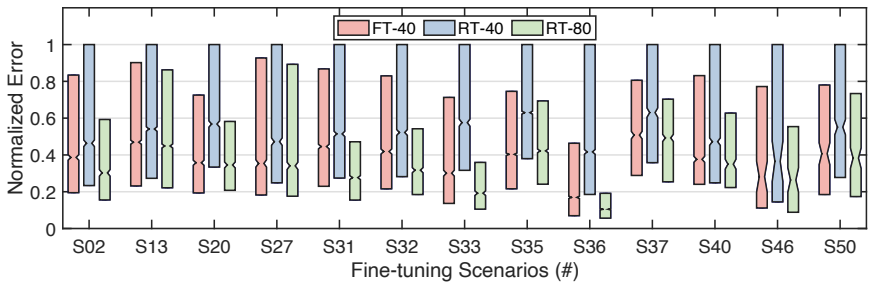


Fig. 6.10: Transfer Learning from LocGPT v1.0

of data from different scenarios. For comparison, we trained separate models from scratch with either 80% or 40% of data, named "RT-80" and "RT-40," respectively, using the same 20% test dataset. To assess transfer learning effectiveness beyond just error rates, we used robust scaling normalization for localization errors, defined as $X_{\text{norm}} = X/\text{IQR}$, with IQR being the interquartile range. The resulting error distributions are shown in Figure 6.10. Our observations highlight two key insights. On one hand, the fine-tuned model consistently surpassed the performance of RT-40, even though both are trained on a similar 40% data subset. For instance, the median errors in the 13 scenarios for RT-40 are reduced by 16.7%, 16.1%, 37.1%, 24.8%, 13.3%, 20.5%, 47.7%, 28.5%, 59.5%, 19.4%, 20.2%, 22.8%, and 23.6% respectively, when compared with FT-40. On the other hand, the outcomes for FT-40 closely mirror the results of RT-80, with median error disparities being 13.7%, 4.5%, 3.2%, 2.8%, 46.0%, 23.8%, 36.1%, 6.5%, 38.2%, 3.1%, 7.0%, 6.2%, and 4.9%, respectively. In short, the results of FT-40 closely align with that of RT-80 (diff: 15%), but are far higher than that of RT-40 (diff: 27%). These comparisons fully demonstrate the efficacy of transferring knowledge from pre-training for specific scenarios, even when faced with constrained data availability.

This efficacy of LocGPT is due to several key factors. First, the pre-trained model's exposure to diverse data enables it to discern various patterns, structures, and data relationships. Second, the pre-trained model's weights offer a robust starting point, encapsulating beneficial features that are typically transferable across tasks, leading

to enhanced performance against models that are arbitrarily initialized. Third, by commencing with a pre-trained model and subsequently fine-tuning with limited data, the risks associated with overfitting are reduced, as the model has already generalized over comprehensive data during its pre-training phase. Finally, the use of spatial spectra-based tokens ensures effective abstraction of hardware diversity, ensuring that LocGPT can be adapted to various localization technologies.

6.7 Limitations and Future works

We discuss the limitations and future improvement of LocGPT as follows:

Zero/Few-shot generalizability: The transfer learning experiments suggest that 40% of new environmental data is essential for effective fine-tuning. This need arises from the intricate and unpredictable nature of indoor environments, amplified by the multipath effect, where signals bounce and create diverse and unpredictable spatial contexts. This complexity demands a deeper learning and adaptation strategy. Integrating additional modalities, like environmental visual data, could offer a viable strategy, enhancing the model's understanding of the context for RF signal propagation. This enhancement has the potential to boost the model's performance in zero/few-shot learning scenarios.

Dynamic adaptation: The performance of the trained model is susceptible to environmental configurations, such as the placement of furniture items like tables and sofas. When primary obstacles or reflectors change their positions, it becomes necessary to fine-tune models using a small number of freshly gathered samples. Regrettably, during the operational phase, we do not have a clear indicator of a drop in prediction accuracy due to the absence of ground truth or a benchmark. Currently, with our model, we can employ the $\mathcal{L}_2$ loss as an indirect measure to monitor performance degradation. A persistently high $\mathcal{L}_2$ loss indicates that the rays estimated by

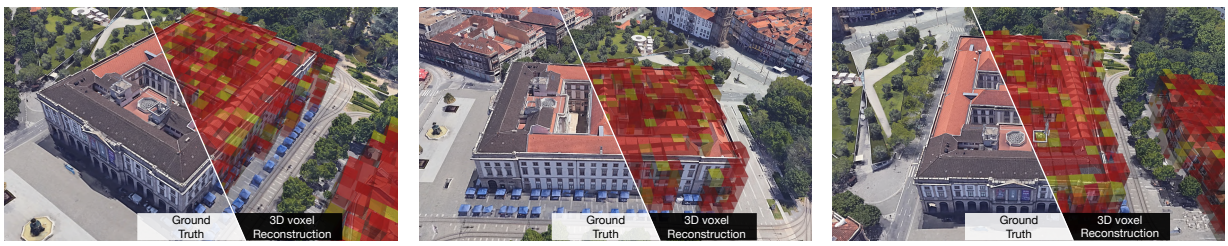
the A-Subnetworks fail to intersect. It suggests that the model might be losing its effectiveness and needs fine-tuning. Future work will explore additional strategies to mitigate the impact of dynamic environmental changes.

Multi-modality scalability: Leveraging the flexibility of the Transformer architecture, LocGPT can easily incorporate multi-modal inputs, such as IMU, visual, and acoustic data, to enhance its localization capabilities. By integrating additional encoders tailored for each modality, the model can perform unified feature extraction, enabling it to adapt to diverse sensor streams. Acoustic data, for example, provides valuable spatial cues, especially in environments where RF signals may be less reliable or visual data is sparse. Time-of-flight measurements or sound patterns could help improve localization by providing insights into object positions or environmental layouts. Additionally, the A-subnetwork, designed for phase information extraction, could be extended to process acoustic signals by modifying the MLP head, while the T-subnetwork could fuse these multi-modal features for enhanced accuracy and robustness. This approach would enable LocGPT to handle complex, multi-sensory environments, opening new possibilities for real-time localization in dynamic, resource-constrained settings.

Chapter 7

Constructing 3D Urban Maps with Satellitic Radiance Fields

7.1 Motivation



(a) 45° Angle View (b) Side View (c) Front View

Fig. 7.1: Illustration of SaRF. Our framework effectively reconstructs 3D urban maps using crowdsourced GNSS data, primarily through a methodical voxelization process. In the figures, medium-sized voxels are used to reduce the overall image size.

3D urban mapping is driven by the need for detailed and accurate representations of urban environments for many reasons, such as better urban planning [132], emergency response [133], environmental monitoring [134], vehicle navigation [135], virtual and augmented reality [136], cultural heritage preservation [137], and so on. Specifically, a comprehensive understanding of the urban landscape is vital for the widespread adop-

tion and success of autonomous vehicles in enabling them to plan routes efficiently, avoid obstacles, and interact with other road users. By creating highly accurate digital representations of real-world environments, these maps allow users enjoying immersive virtual and augmented reality to explore and interact with urban settings in a realistic manner. 3D urban mapping also benefits drone delivery systems by enabling optimized routing, obstacle avoidance, and precise navigation, ensuring safe and efficient transportation of goods in complex urban environments.

Advanced 3D mapping technologies include several key methods: (1) LiDAR, using laser pulses to create detailed point clouds [21, 22]; (2) Photogrammetry, generating 3D data from overlapping 2D images; (3) Synthetic Aperture Radar (SAR) for high-resolution 3D mapping in challenging conditions; (4) Structure from Motion (SfM), reconstructing 3D structures from 2D drone images; and (5) SLAM, used in robotics and autonomous vehicles for real-time mapping and localization. These technologies contribute significantly to precise, high-quality mapping, enhancing our understanding of complex environments. However, they often focus on major structures and require lengthy intervals for updates, with limited access for external use. There remains a gap for cost-effective, regularly updated urban 3D modeling techniques with acceptable accuracy.

Recent studies have delved into the use of crowdsourced geospatial intelligence, leveraging GNSS data casually collected from mobile devices to identify buildings [138] and even construct 3D models of cities [139–141]. Although these methods might not match the accuracy of traditional techniques, they offer the advantages of reduced costs and broader area coverage. The underlying concept is that urban structures like buildings can block, diminish, or reflect GNSS signals. By examining the fluctuations in signal strength and the visibility of satellites, it becomes feasible to deduce the presence and bounding boxes of buildings or other urban elements. This method is economical since it does not necessitate exclusive hardware or specialized data-gathering campaigns; it instead utilizes GNSS data generated incidentally by individ-

uals in their everyday routines. The extensive data coverage is attributed to the vast number of people using GNSS-equipped devices, which facilitates widespread data acquisition across large geographical regions and yields insights into various urban landscapes.

We expand upon the idea of crowdsourced geospatial intelligence by integrating NeRF (Neural Radiance Fields) into the process of 3D urban mapping [32]. NeRF is a pioneering deep learning approach that generates realistic 3D scenes from multiple 2D images. The model is trained to represent a scene as a continuous function, mapping 3D coordinates and viewing directions to colors and densities. Once trained, NeRF can create new views of the scene, consistent with the input images. Recognized for rendering detailed 3D scenes with few inputs, NeRF has led to significant advancements in 3D scene representation [32, 36, 79–81]. Recently, the adaptation of NeRF to work with electromagnetic data (NeRF^2 [142]) has opened opportunities to use crowdsourced GNSS data for fine-grained 3D building reconstruction.

In this work, we introduce the Sattellic Radiance Fields (SaRF), a deep learning method for generating 3D urban maps using GPS data sourced through crowdsourcing. Similar to NeRF^2, SaRF compiles GPS measurements from diverse locations. It employs a sparse voxel octree structure to effectively capture voxel-based implicit fields, highlighting physical characteristics like voxel density. The model is progressively refined via a differentiable ray-marching process, resulting in detailed urban topography maps that accurately reflect variations in voxel densities. This sophisticated modeling and reconstruction approach, facilitated by SaRF, opens up new possibilities in urban planning, simulation, and analysis. Fig. 7.1 presents an example of voxelization applied to a library building, viewed from various angles.

Challenges. Translating the above idea into a practical system presents numerous challenges.

• The first challenge lies in the need for a comprehensive collection of GPS data from

multiple angles to accurately depict a scene, a notably demanding task. To tackle this, we developed an Android app and collaborated with numerous volunteers who collected 617,286 GPS measurements across a specific scene within a year. Furthermore, we leveraged publicly available GPS data from the SenseMyCity project [140, 143], which provided extensive data from five different scenes. This dataset includes over 27.4 million GPS records, gathered by 900 unique users across a span of five years. Collectively, through these crowdsourcing efforts, we accumulated approximately 28 million GPS records.

• Second, traditional approaches often rely on line-of-sight (LOS) attenuation models, like those in [139–141], to infer the presence of buildings along the LOS path. While straightforward, this method falls short in efficiency as it overlooks the impact of reflections from non-line-of-sight (NLOS) propagations. These reflections could be instrumental in deducing the layout of adjacent buildings and in refining the LOS propagation by distinguishing it from other NLOS paths. To address this gap, we introduce SaRF, a model designed to trace potential signal propagations from a comprehensive range of directions. Consequently, every kind of propagation captured by a GPS receiver is effectively utilized in the scene representation, ensuring a more thorough and accurate mapping process.

• Third, in our method, we adopt voxelization to depict a scene. This technique divides the scene into numerous small cubic units, or voxels, which aids in constructing the 3D urban map. Consider a case where each voxel has a volume of 10 cm^3. In a scene spanning 100 m^3, this equates to a total of a billion voxels, creating a substantial computational challenge in the ray marching process. To overcome this, we implement a hierarchical data structure, specifically an octree. Within this framework, each larger voxel can be subdivided into eight smaller sub-voxels during each training iteration, as required, until the smallest voxel size is achieved. This approach is known as progressive training. In the final analysis, non-air voxels whose relative density surpasses a set threshold are identified as part of a building, thereby enhancing

the precision of the resultant 3D urban map.

Contributions. The key contributions of this work are outlined as follows. Firstly, to the best of our knowledge, we are the first to introduce neural radiance fields to address the challenges of a 3D urban map reconstruction using crowdsourced GPS data. Secondly, we introduce SaRF, a novel framework that transforms the training of radiance fields into a problem akin to conventional global illumination. Thirdly, we implement our proposed methodology to construct 3D maps for six diverse scenes, utilizing tens of millions of GPS records available in the public domain.

7.2 Preliminary

In this section, we introduce the background knowledge.

7.2.1 Global Navigation Satellite System

The Global Navigation Satellite System (GNSS) is a collective term for satellite-based navigation systems providing accurate global positioning and timing. Key GNSS systems include GPS, GLONASS, Galileo, and BeiDou. Focusing on GPS, it uses three L-band frequencies, primarily L1 at 1.57542 GHz, as L2 and L3 are less accessible. GPS signals, modulated using BPSK, contain ephemeris data and time stamps. GPS receivers calculate their locations by measuring the time delay of signals from at least four satellites, employing trilateration to determine latitude, longitude, and altitude.

Various factors can impact the precision of GPS trilateration. Common sources of error include inaccuracies in satellite clocks, atmospheric delays, multipath errors, suboptimal satellite geometry, receiver noise, and selective availability. Particularly, urban buildings can significantly impact GPS accuracy and reliability in several ways:

(1) Signal Blockage: High buildings can obstruct the line of sight between the receiver and satellites, particularly in densely built areas, leading to signal loss. **(2) Multipath Errors:** GPS signals reflecting off buildings can take longer or different paths, causing errors where the receiver miscalculates its position. **(3) Signal Attenuation:** Urban structures can attenuate, or weaken, GPS signals before they reach the receiver, which can further degrade positioning accuracy. Consequently, under optimal conditions with clear skies and no enhancements, standard GPS typically achieves a horizontal accuracy of around 5-10 meters [144].

7.2.2 Augmented GPS Accuracy

To mitigate these potential errors, mobile platforms such as iOS and Android incorporate various advanced GPS methods. **(1) Assisted GPS:** It uses an internet connection to assist in the acquisition of satellite data [145]. By accessing information from a network server, mobile devices can reduce the time to find and lock onto satellites, especially in urban settings where GPS signals might be obstructed. **(2) Differential GPS:** The system [146] improves accuracy by using a network of fixed, ground-based reference stations to broadcast the difference between the positions indicated by the satellite systems and the known fixed positions. Mobile devices can use it to correct their own GPS data. **(3) Augmentation Systems:** Systems like WAAS (Wide Area Augmentation System) in North America and EGNOS (European Geostationary Navigation Overlay Service) in Europe [147], provide corrections and additional information to improve GNSS performance in terms of accuracy, integrity, and availability. **(4) Multi-GNSS Support:** Modern mobile devices are increasingly capable of receiving signals from multiple GNSS networks (like GLONASS, Galileo, BeiDou) simultaneously [148], increasing the number of satellites available for positioning and thereby enhancing accuracy. **(5) Integration with Other Sensors:** Modern mobile devices often integrate GPS with other sensors like accelerometers, gyroscopes, and barometers to improve location accuracy [149]. This sensor fusion

approach helps in providing more accurate positioning, especially where GPS signals are weak or unavailable. The combined use of the above methods can significantly enhance GPS accuracy, often achieving cm-level precision even in urban settings.

7.2.3 3D voxel map

SaRF aims to reconstruct the 3D voxel map (i.e., occupancy map), which is a spatial representation technique used primarily in robotics and autonomous vehicle navigation. It involves dividing a three-dimensional space into discrete, uniformly sized cubes known as voxels, each of which can store various types of data about the environment. This method allows for highly detailed and dynamic modeling of physical spaces, facilitating precise object and obstacle detection critical for navigation and decision-making in autonomous systems. For example, companies like Tesla incorporate 3D voxel maps in their self-driving cars [150] to process and interpret vast amounts of sensory data. These maps are instrumental in understanding complex environments, enabling the vehicle to navigate safely by identifying and classifying objects. Key benefits of using 3D voxel maps include improved accuracy in object perception and the ability to perform terrain analysis. This technology not only enhances the safety features of autonomous vehicles but also improves their operational efficiency in diverse driving conditions.

7.2.4 Assumption

Modern mobile devices use advanced localization algorithms that integrate previously mentioned GPS techniques, typically achieving an accuracy of less than 1 meter, as commonly experienced in daily usage. Based on this, we adopt a practical assumption:

Assumption 1 *The positions determined by GPS receivers are deemed relatively accurate, despite raw GPS signals being susceptible to disturbances from urban architec-*

ture, like signal blockage and multipath errors.

Furthermore, satellite positions, known accurately through ephemeris data, enable the precise establishment of both GPS transmitters (satellites) and receivers (devices) locations, despite potential interference from buildings on raw GPS signals. This observation motivates us to discover building structures using raw GPS signals.

7.3 Overview

For the sake of simplicity, we demonstrate our system using GPS signals, despite having collected various GNSS data types. Given that smartphones, commonly equipped with GPS receivers, are widely used, it becomes straightforward to amass substantial data via crowdsourcing. This cost-effective method of data collection encourages more frequent and highly efficient updates of 3D maps. However, our aim is not to supplant existing high-accuracy measurement methods like LiDAR and SAR but to offer an additional methodology for real-time updates. As shown in Fig. 7.2, the workflow of our approach contains four steps:

• **Step 1 - Data Collection:** Gather comprehensive raw GPS data, including raw RF signals, satellite positions, and receiver locations. This data forms the foundation for accurate 3D mapping, providing insights into spatial relationships.

• **Step 2 - Establishing SaRF:** Develop a SaRF model to interpret GPS data and model the urban environment's physical characteristics. SaRF serves as the basis for transforming GPS signals into a structured representation for a 3D scene.

• **Step 3 - Training:** Train the neural network through ray tracing, a technique that simulates GPS signal propagation in the environment. This semi-supervised approach enhances learning from both labeled and unlabeled data, adapting to diverse urban settings.

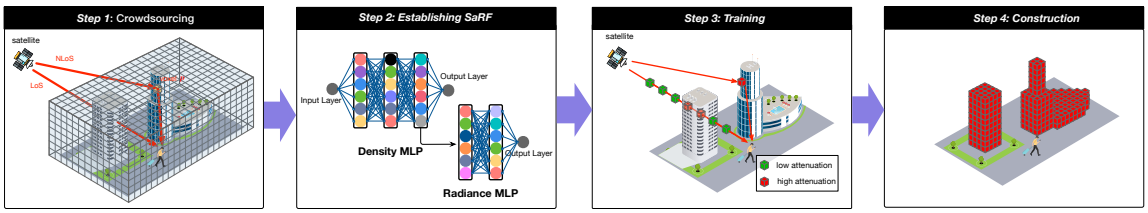


Fig. 7.2: Approach to Building 3D Urban Maps. This involves (1) amassing extensive raw GPS data, encompassing raw RF signals, positions of satellites, and locations of receivers; (2) creating a SaRF, depicted through two Multilayer Perceptrons (MLPs); (3) training the neural network using ray tracing within a semi-supervised learning framework; and (4) forming the 3D map by eliminating superfluous aerial voxels.

• **Step 4 - Reconstruction:** Process and analyze data to create a 3D map by pruning unnecessary aerial voxels. This step focuses the map on critical urban features, resulting in a realistic and functional 3D urban map.

The following subsequent will elaborate on each step.

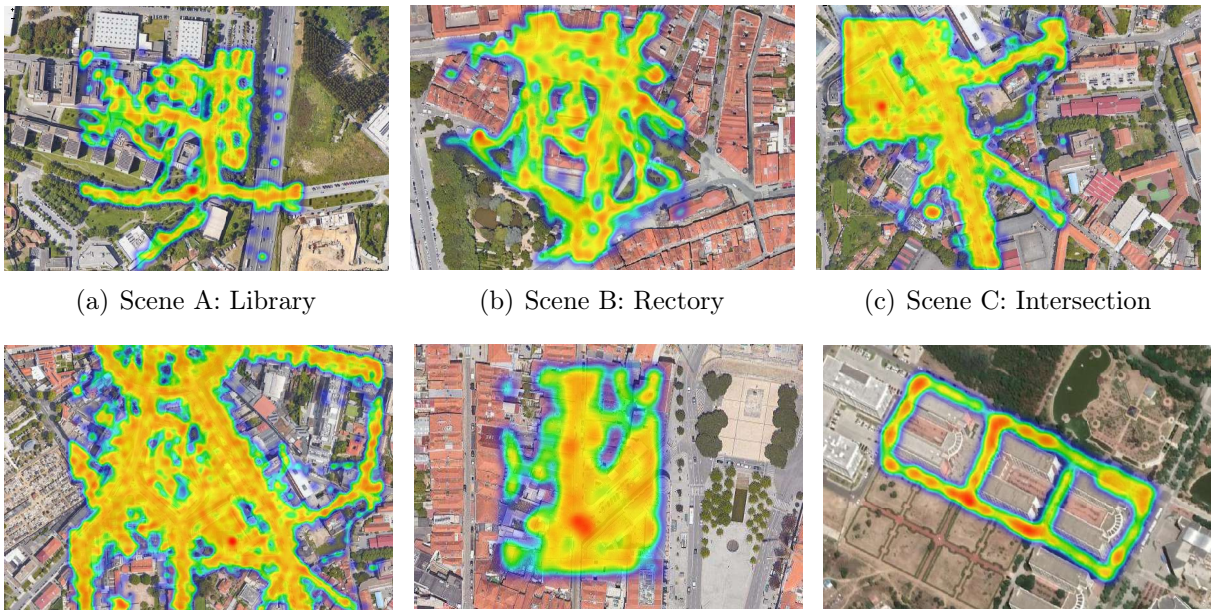
7.4 Data Collection

In this section, we provide information about the crowdsourced data collection.

7.4.1 Methodology

Our team developed a crowdsensing application that autonomously collects GNSS data from users' smartphones during their regular activities. This application has successfully gathered 617,286 million geo-tagged data points from participants' daily movements. Ten volunteers participated in the data collection process, traversing the perimeters of three buildings, each approximately 23 meters in height. We use the Android API on smartphones to report the GNSS data. Each data record consists of data following:

$$[\text{Lon}_p, \text{Lat}_p, \text{Alt}_p, \text{Lon}_s, \text{Lat}_s, \text{Alt}_s, \text{SNR}, \theta, \text{PRN}, \text{Timestamp}]$$



(a) Scene A: Library (b) Scene B: Rectory (c) Scene C: Intersection

(d) Scene D: Avenue (e) Scene E: Canyon (f) Scene F: Campus

Fig. 7.3: Spatial distribution of GNSS data across various scenes. (a)-(b) display the GPS data from five different scenes sourced from the SenseMyCity dataset, while (f) illustrates the GPS records we gathered around the architectural complex of a university campus.

where Lon_p, Lat_p, Alt_p, Lon_s, Lat_s, and Alt_s are the longitude, latitude, and altitude of the smartphone and satellite, respectively. SNR and θ denote the received signal-to-noise ratio and carrier phase of the GNSS signal. PRN code is used to identify the ID of the satellites. Additionally, the use of network-based location data from Wi-Fi and cellular sources by Android devices augments the location accuracy, particularly in areas with poor GPS signal reception.

Moreover, our study incorporated publicly accessible crowdsourced GPS data from SenseMyCity [140, 143], which amassed 27.4 million data points through crowdsourcing involving 900 unique users over a period of five years. The data undergoes the pre-processing [140], leading to the absence of phase and timestamp. The SenseMyCity application employs a low-rate, energy-efficient sensing algorithm to detect user movement, enhancing the frequency of data collection specifically during periods of travel [140].

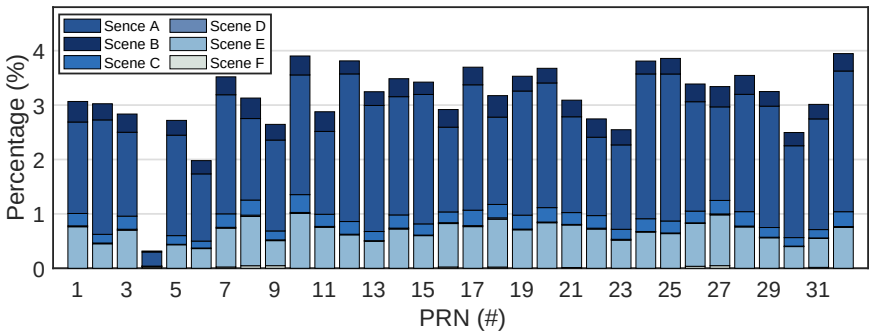


Fig. 7.4: Histogram of GPS records across 32 GPS satellites

In short, an accumulation of approximately 28 million GPS records is collected in a crowdsourcing way, which fuels the next step of the establishment of SaRF.

7.4.2 Data Analysis

Fig. 7.3 illustrates the record distributions for six unique scenes, each serving as a benchmark for our experimental analysis. These scenes, characterized by their distinct layouts, are as follows: **Scene A**: Features a $30 \times 30 \times 34$ m^3 building, surrounded by open areas and a car park, with notably dense data collection in the northern section. **Scene B**: Encompasses a larger, yet shorter building ($60 \times 90 \times 19$ m^3) situated downtown, surrounded by pedestrian zones and a park. **Scene C**: Distinguished by a 5-way junction with buildings 24-35m in height, this scene's data predominantly originates from vehicles. **Scene D**: A busy, 100m segment of a 22 m-wide avenue, lined with buildings 16-28m tall, where data collection is largely vehicle-based. **Scene E**: An urban canyon with a narrow road flanked by buildings 15-20m tall, which likely experiences poor GPS reception. **Scene F**: Represents the architectural layout of a university campus, featuring three main buildings. Each scene presents unique architectural features, with diverse recesses and protrusions typical of urban structures.

Fig.7.4 shows a histogram of the number of satellites visible in the entire dataset, with an average of 7.4 satellites visible at each GNSS location. Notably, satellite

#4, the first GPS satellite launched in 1978 and now decommissioned, still appears occasionally in the data despite its retirement [151]. The distribution indicates that Scene A (the library) forms the largest part of the dataset, while Scene F (the campus) has a smaller representation. The amount of data from each scene is determined by factors like user traffic in the area and the duration of data collection.

7.5 Satellitic Radiance Fields

In this section, we transform the task of estimating the satellite radiance field into a problem akin to global illumination, a concept extensively explored in 3D computer graphics through radiosity.

7.5.1 Radiosity

We segment an urban landscape into a multitude of small, continuous voxels and apply discrete geometry calculations to each. As GPS satellites emit continuous signals from the sky toward the ground, these voxels are illuminated in a manner akin to sunlight, with the satellite serving as the light source. Buildings within the scene either attenuate or reflect these RF signals, which are then captured by GPS receivers. This process resembles global illumination in computer graphics, where the intricate behavior of light in the real world is simulated to create more lifelike images.

The size of each voxel is a customizable parameter, balancing between computational performance and spatial accuracy. In terms of RF signal propagation, each voxel is characterized by specific attenuation and radiation properties.

Attenuation Characteristic

When a GPS signal traverses a voxel, it undergoes attenuation, which is contingent on the voxel's physical properties. For example, a voxel filled with air only causes negligible signal fading. In the standard RF model, the attenuation coefficient of the i^{th} voxel denoted by V_i is represented as a complex number, expressed by the equation:

$$h(V_i) = \Delta a(V_i) e^{\mathbf{J}\Delta\theta(V_i)} \tag{7.1}$$

where $\Delta a(V_i)$ (normalized from 0 to 1) signifies the reduction in amplitude, while $\Delta\theta(V_i)$ (varying from 0 to 2π) indicates the phase shift. For ease of computation in the ray-marching algorithm, we transform this attenuation coefficient into a negative logarithmic form as below:

$$\begin{aligned} \delta(V_i) &= -\ln(h(V_i)) = -\ln(\Delta a(V_i) e^{\mathbf{J}\Delta\theta(V_i)}) \\ &= -\ln \Delta a(V_i) - \mathbf{J}\Delta\theta(V_i) \end{aligned} \tag{7.2}$$

where the real part $-\ln \Delta a(V_i)$ is non-negative since $\Delta a(V_i)$ is less than or equal to 1. Conversely, the imaginary part $-\theta(V_i)$ is negative, reflecting the extent of phase rotation. The preference for using the negative logarithm in this context is due to two ranges being easily confined through the applications of ReLu and Sigmoid activation functions in the subsequent neural network layers.

The attenuation characteristic at a given RF frequency is influenced by physical attributes such as size, density, and the composition of the material, irrespective of whether the voxel is illuminated or not. Typically, a voxel filled with air exhibits a substantially lower attenuation coefficient compared to one made of concrete. This variance in attenuation properties plays a crucial role in accurately defining the contours of buildings in the next step.

Radiance Characteristic

In accordance with the Huygens–Fresnel principle, a voxel acts as a secondary source of radiance upon receiving signals from satellites directly or other voxels attributed to multipath effects. Consequently, we conceptualize each voxel V_i as an emergent RF source originating from the GPS signals transmitted by the GPS satellite O. This voxel then re-emits a new signal along the direction ω, which can be mathematically expressed as:

$$S(V_i, O, \omega) = a(V_i)e^{\mathbf{J}\theta(V_i)} \tag{7.3}$$

where $\theta(V_i)$ represents the initial phase value and $a(V_i)$ the initial amplitude of the signal. Given that a voxel can potentially radiate in any direction, ω symbolizes a prospective 2D direction, denoted as $\omega = (\alpha, \beta)$. The parameters α and β, varying within $(0, 2\pi]$ and $(0, \pi/2]$ respectively, correspond to the azimuthal and elevation angles. This model allows us to account for the diverse and complex ways in which voxels may radiate RF signals, influenced by their interactions with incoming satellite signals and the surrounding environment. Understanding these radiative behaviors is crucial for accurately mapping and interpreting the RF signal propagation within urban landscapes.

7.5.2 Neural Radiance Network

Drawing inspiration from NeRF^2, we employ two MLPs to model two key characteristics: the attenuation and radiance of each voxel. This approach leads us to designate the neural network as *satellitic radiance fields* (SaRF), as shown in Fig. 7.5. Central to SaRF is the concept of representing a scene using two distinct continuous functions. These functions, embodied by the MLPs, are tasked with linking 3D coordinates to the attenuation and radiance properties of a voxel. They process inputs such as the voxel's position V_i, the satellite's position s, and the viewing direction ω, to out-

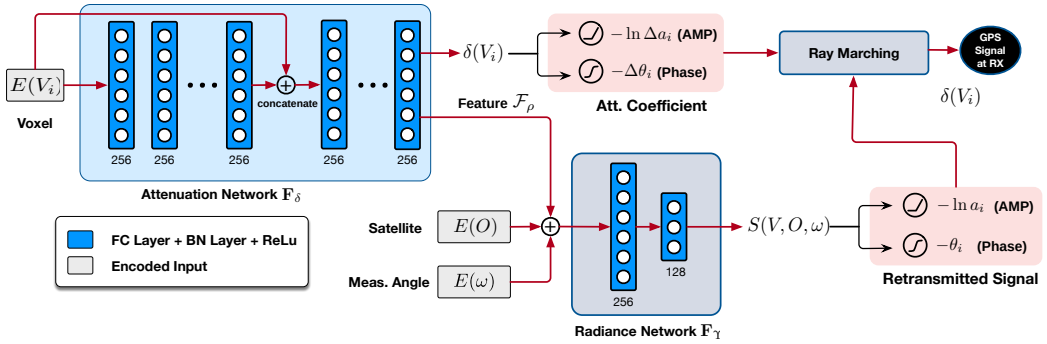


Fig. 7.5: Neural Network Architecture of SaRF

put the attenuation coefficient $\delta(V_i)$ and the reemitted GPS signal $S(V_i, \omega, s)$. The network is fundamentally structured into two segments: the attenuation subnetwork, which focuses on signal weakening, and the radiance subnetwork, which handles signal retransmission.

Attenuation Subnetwork

The first MLP, termed the *attenuation subnetwork*, is designed to correlate a voxel's position with its attenuation coefficient, which is represented by the following equation:

$$\mathbf{F}_\delta : (V_i) \rightarrow (\delta(V_i), \mathcal{F}(V_i)) \tag{7.4}$$

The network processes the 3D position of a voxel and yields two outputs: the attenuation coefficient $\delta(V_i)$ and a 256-dimensional feature vector $\mathcal{F}(V_i)$. The coefficient $\delta(V_i)$ is a complex number. Its real component is modified using a ReLu activation function to guarantee that $-\ln(\Delta a(V_i)) \geqslant 0$ (implying $\Delta a(V_i) \leqslant 1$). The imaginary component, on the other hand, is adjusted with a $2\pi \times$ sigmoid function to constrain the phase shift between 0 and 2π. This feature vector $\mathcal{F}(V_i)$ is then employed as input for the subsequent radiance subnetwork.

Composed of eight fully connected layers, each with ReLU activations and 256 channels, the MLP is structured to process this data effectively. It is important to note

that the attenuation characteristic is solely determined by the voxel's density and the structural composition of the scene, making it independent of the incoming RF signals. As a result, the attenuation subnetwork does not require satellite information as part of its input.

Radiance Subnetwork

The radiance subnetwork, represented by $\mathbf{F}_\Upsilon$, is tasked with predicting the characteristics of the GPS signal that is retransmitted by a voxel. This prediction is based on the voxel's attenuation feature vector $\mathcal{F}(V_i)$, the observation direction ω, and the satellite position O (i.e., orbiter). The functionality of this subnetwork is encapsulated in the equation:

$$\mathbf{F}_\Upsilon : (\mathcal{F}(V_i), O, \omega) \rightarrow (a(V_i), \theta(V_i)) \tag{7.5}$$

It is worth noting that the voxel's position is embedded into the feature vector. Comprising two fully connected layers equipped with ReLU activation functions, the subnetwork features 256 channels in the first layer and 128 in the second. Its output is the direction-dependent retransmitted GPS signal $a(V_i)e^{\mathbf{J}\theta(V_i)}$, where the amplitude and phase components are respectively refined using ReLu and Sigmoid activation functions.

The radiance subnetwork plays a crucial role in modeling how each voxel interacts with the incoming GPS signals, transforming and re-emitting them based on their unique characteristics. The subnetwork's structure allows for the nuanced interpretation of signal behavior, accounting for variations in signal strength (amplitude) and the phase shift by the voxel's material properties and position relative to the satellite. By processing the amplitude with ReLU, the network ensures a non-negative output, while the Sigmoid function applied to the phase ensures it remains within a valid range. This detailed modeling is instrumental in accurately reconstructing the complex signal interactions within an urban environment, facilitating a deeper under-

standing of GPS signal propagation and its interaction with various urban structures.

7.5.3 Summary

Our approach contrasts with NeRF^2, which primarily targets signal prediction, making both of its subnetworks crucial. Different materials, such as concrete or wood, have distinct attenuation characteristics, enabling us to identify voxels composed of non-air materials. Once the attenuation subnetwork is proficiently trained, it becomes possible to reconstruct building structures within the scene. While the radiance subnetwork might appear superfluous for our end goal, it is in fact essential for the training process. Direct collection of ground truth data for attenuation coefficients is impractical, making it challenging to train the attenuation subnetwork in isolation. The training dataset, crowdsourced via smartphones, offers only GPS signal samples captured at various locations within the scene. Consequently, the training approach amalgamates both subnetworks, employing a semi-supervised method to train them jointly. This integrated training methodology ensures a comprehensive learning process, leveraging the strengths of both subnetworks to achieve accurate scene reconstruction.

7.6 Training

In this section, we introduce a self-supervised method for training SaRF using a dataset crowdsourced from samples of GPS signals gathered at various locations.

7.6.1 Divide and Conquer

Self-supervised learning is centered around devising a pretext task, a learning problem that can be resolved using only the input data. By tackling this task, the model learns

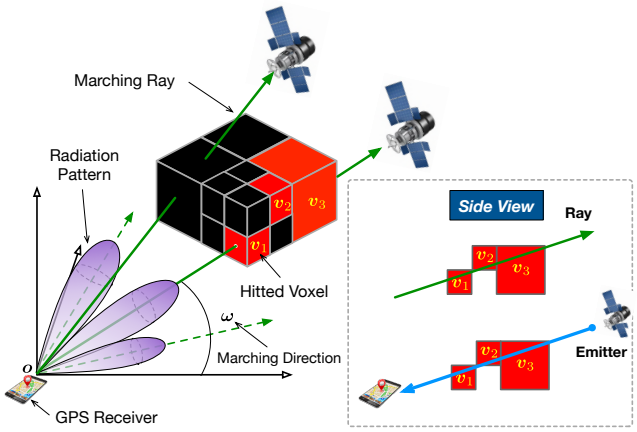


Fig. 7.6: Ray Marching. We progressively trace rays from the smartphone's viewpoint into the scene to predict the GPS signal received by the receiver.

to extract meaningful representations. For SaRF, the task is to accurately predict the GPS signal at a specific location. The objective of the training is to reduce the difference between the real GPS signal received by a smartphone and the model's predicted signal, all without direct supervision. This means that there are no pre-established ground-truth outputs for any of the two subnetworks involved in the process.

To leverage SaRF for predicting the GPS signal received at a specific location, we adopt a divide-and-conquer approach. As shown in Fig. 7.6, the GPS signal a smartphone receives can be methodically broken down. Specifically, ❶ the overall GPS signal is an amalgamation of signals from various satellites → ❷ the signal from each satellite is itself composed of signals from multiple directions → ❸ the signal from a particular direction is an aggregate of signals from all voxels along that path → ❹ the signal retransmitted from a specific voxel. By tracing the signal from individual voxels and cumulatively combining them, we can predict the final received signal. This technique, akin to ray marching in computer graphics, involves progressively tracing rays from the smartphone's viewpoint into the scene. The subsequent sections will delve deeper into utilizing ray marching for predicting GPS signals received by smartphones.

7.6.2 Tracing from a Single Voxel

The RF signal propagation model can generally be depicted using the Friis transmission equation, represented as:

$$S_{\mathrm{RX}} = H_{\mathrm{TX}\to\mathrm{RX}} S_{\mathrm{TX}} \tag{7.6}$$

where S_{TX} and S_{RX} denote the transmitted and received signals, while $H_{\mathrm{TX}\to\mathrm{RX}}$ symbolizes the channel attenuation. Accordingly, the GPS signal emitted from a specific voxel V_i can be expressed as:

$$S_{\mathrm{vox}}(P_0, V_i) = H_{V_i\to P_0} S(V_i, O, \omega) \tag{7.7}$$

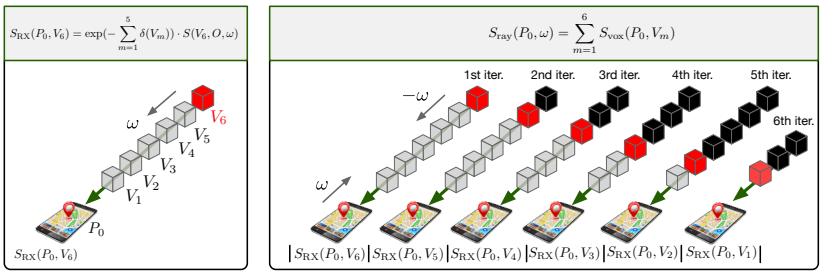
In this scenario, $S(V_i, O, \omega)$ indicates the GPS signal re-emitted from V_i in the direction ω to the receiver located at P_0, originating from the satellite O. If there are M voxels, denoted as $\{V_1, V_2, \ldots, V_M\}$, along the path between V_i and P_0, they collectively attenuate the GPS signal $S(V_i, O, \omega)$. The channel attenuation is thus formulated as:

$$\begin{aligned} H_{V_i\to P_0} &= H(V_1)\cdot H(V_2)\cdot\cdots\cdot H(V_M) = \prod_{m=1}^{M}\left(\Delta a(V_m) e^{\mathbf{J}\Delta\theta(V_m)}\right) \\ &= \exp\left(-\sum_{m=1}^{M} -\ln\left(\Delta a(V_m) e^{\mathbf{J}\Delta\theta(V_m)}\right)\right) \\ &= \exp\left(-\sum_{m=1}^{M}\delta(V_m)\right) \end{aligned} \tag{7.8}$$

where $\delta(V_m)$ represents the negative logarithmic attenuation caused by voxel V_m. The logarithmic form simplifies calculations, transforming multiplications into summations. Integrating this equation into Eqn. 7.7, we derive the GPS signal received at the receiver contributed by V_i as follows:

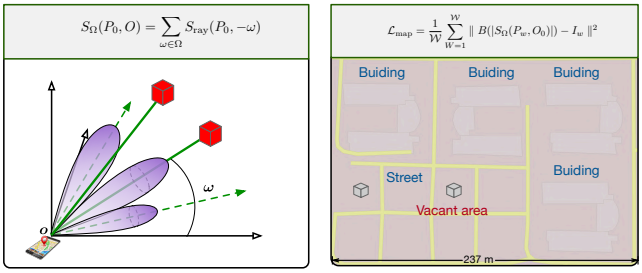
$$S_{\mathrm{vox}}(P_0, V_i) = \exp\left(-\sum_{m=1}^{M}\delta(V_m)\right)\cdot S(V_i, O, \omega) \tag{7.9}$$

where $\delta(V_m)$ and $S(V_i, O, \omega)$ are obtained from Eqn. 7.4 and Eqn. 7.5, respectively. Both are fitted by the two neural subnetworks.



(a) Tracing from a voxel

(b) Tracing from a single direction



(c) Tracing from all directions

(d) Tracing with a 2D map

Fig. 7.7: Divide-and-Conquer Ray Marching Algorithm

Fig. 7.7(a) shows an example. The GPS signal $S(V_6, O, \omega)$, re-emitted by voxel V_6 (depicted as the red voxel), experiences attenuation while passing through voxels $V_1 - V_5$ (shown as gray voxels). The direction ω, relative to V_6, results in an attenuation coefficient that is the sum of $\delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5$, measured in dB units. For signal computation, this coefficient is reconverted to volts, and the received signal at the position P_0 is expressed as $S_{\text{vox}}(P_0, V_6) = \exp(-(\delta_1 + \cdots + \delta_5)) \cdot S(V_6, O, \omega)$. There might be concerns about neglecting the RF signals reflected from adjacent voxels, which originally emanate from V_6. It's crucial to recognize that in this model, each voxel is conceptualized as an RF source that captures signals from all possible directions. Therefore, reflections are treated as an integral part of the re-emission process of the neighboring voxels. This approach ensures that when tracing the signals from nearby voxels, such reflections are automatically counted in the analysis, providing a thorough representation of signal dynamics within the scene.

7.6.3 Tracing from a Single Direction

A given direction ω can be conceptualized as a ray originating from the position P_{RX} and extending towards ω. The points along this ray can be mathematically described as:

$$P(r, \omega) = P_0 + r\omega \tag{7.10}$$

where r represents the radial distance from the RX to the points on the ray, with $P(0, \omega) = P_0$ denoting the starting point. Consequently, the RX is capable of receiving signals that are retransmitted only from the voxels intersecting this particular ray.

Imagine a scenario where a ray intersects with N voxels, labeled as $\{V_1, V_2, \ldots, V_N\}$, with V_1 being the closest to the receiver and V_N the farthest, also acting as the scene's boundary voxel. The signal received from direction ω can be understood as the cumulative effect of the GPS signals re-emitted from these N voxels along the ray. This can be mathematically represented as:

$$\begin{aligned} S_{\text{ray}}(P_0, \omega) &= \sum_{n=1}^{N} S_{\text{vox}}(P_0, V_n) \\ &= \sum_{n=1}^{N} \left(\exp\left(- \sum_{m=1}^{n-1} \delta(V_m) \right) \cdot S(V_n, O, -\omega) \right) \end{aligned} \tag{7.11}$$

This formulation is derived from Eqn. 7.9. It's important to note that while the direction ω is defined with respect to the receiver, the re-emitted GPS signal from a voxel is relative to the voxel itself, effectively 180° opposite to ω. Therefore, $-\omega$ is used in the term $S(\cdot)$. An illustrative example is shown in Fig. 7.7(b), where we execute six iterations of voxel-based tracing. Each iteration calculates the GPS signal re-emitted from voxels V_6 to V_1 sequentially.

7.6.4 Tracing from all Directions

GPS receivers in smartphones typically come with directional antennas, but the orientation of the phones in the crowdsourced dataset is not known. Therefore, it's

necessary to consider all possible directions, represented by Ω. As a result, the GPS signal received by a receiver from a satellite is the cumulative effect of signals from all these directions:

$$\begin{aligned} S_{\Omega}(P_0, O) &= \sum_{\omega \in \Omega} S_{\text{ray}}(P_0, \omega) = \sum_{\omega \in \Omega} \sum_{n=1}^{N} S_{\text{vox}}(P_0, V_n) \\ &= \sum_{\omega \in \Omega} \sum_{n=1}^{N} \left(\exp\left(- \sum_{m=1}^{n-1} \delta(V_m) \right) \cdot S(V_n, O, -\omega) \right) \end{aligned} \tag{7.12}$$

where the voxels of $\{V_1, V_2, \ldots, V_N\}$ are on the direction ω. An example of this is illustrated in Fig. 7.7(c). Typically, Ω represents all possible directions. Yet, when the directionality of the GPS receiver is known, this range can be narrowed down significantly.

7.6.5 Tracing from all Satellites

For a specific detected satellite, we can collect a set of GPS signals at $\mathcal{N}$ distinct locations through crowdsourcing. The training process aims to minimize the following objective function:

$$\mathcal{L}_{\text{gps}} = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \parallel |S_{\Omega}(P_n, O)| - |\tilde{S}_{\Omega}(P_n, O)| \parallel^2 \tag{7.13}$$

where $|\tilde{S}_{\Omega}(P_n, O)|$ denotes the strength of actual raw signal received at position P_n, which is originated from the satellite O (i.e., the location of the satellite at the current time). The goal of this objective function is to reduce the strength discrepancy between the predicted GPS signal, $S_{\Omega}(P_n, O)$, as calculated by our model, and the real-world signal, $\tilde{S}_{\Omega}(P_n, O)$, gathered from the field.

A smartphone typically has the capability to simultaneously receive GPS signals from at least three satellites at any given location. This means that the voxels within a scene are illuminated by multiple satellites. Utilizing the CDMA encoding scheme, the smartphone is able to distinguish and decode each satellite's GPS signals, cap-

turing both amplitude and phase information. Let us assume that the smartphone collects GPS signals from $\mathcal{K}$ satellites, represented as $O_1, O_2, \ldots, O_{\mathcal{K}}$. Under these circumstances, the loss function is modified as follows:

$$\mathcal{L}_{\text{gps}} = \frac{1}{\mathcal{K}\mathcal{N}} \sum_{k=1}^{\mathcal{K}} \sum_{n=1}^{\mathcal{N}} \| \, |S_\Omega(P_n, O_k)| - |\tilde{S}_\Omega(P_n, O_k)| \, \|^2 \tag{7.14}$$

Assuming that the scene remains static, the objective of the training is to minimize the difference between the received and predicted GPS signals at various positions, emanating from different satellites. Typically, smartphones report the estimated amplitude of signals in dBm and their phase. For the purposes of our model, it is necessary to convert the amplitude from dBm to volts. Once converted, these amplitude values, along with the phase information, are combined to form a complex exponential number. This approach allows for a more accurate representation of the GPS signals and facilitates more precise training and prediction of signal behaviors in the model.

7.6.6 Tracing with a Known 2D Map

2D city maps, providing a bird's eye view of urban areas, display streets, landmarks, buildings, and other features, aiding in navigation and urban analysis. These maps are useful for various stakeholders, including tourists, residents, and planners, for city exploration and infrastructure study. An example from Google Maps, shown in Fig. 7.7(d), illustrates buildings in gray, roads in yellow, and vacant areas in pink, effectively portraying the physical layout. Leveraging the accessibility of 2D maps, we use them as a supportive tool in training SaRF. We simulate GPS signals as coming from above to the ground level, where the receiver is conceptually placed corresponding to a pixel on the 2D map. If the receiver is under a building, complete signal attenuation is expected; if in open space, minimal attenuation occurs. The 2D map is thus converted to a binary format, with buildings marked as zero and open

areas as one. Our training focuses on minimizing a specific loss function derived from this binary map representation:

$$\begin{aligned} \mathcal{L}_{\text{map}} &= \frac{1}{\mathcal{W}} \sum_{W=1}^{\mathcal{W}} \parallel B(|S_\Omega(P_w, O_0)|) - I_w \parallel^2 \\ \text{where} \quad B(x) &= \begin{cases} 1 & x > \xi \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{7.15}$$

where P_w refers to the GPS coordinates of the w^{th} pixel in the 2D map, which spans the scene of interest. ξ represents the threshold for minimal detectable GPS signal strength. O_0 is the position of a virtual satellite, strategically placed at the center above the scene at an altitude typical of GPS satellites (20,200 km), allowing us to assume that the GPS signals are parallel with a 90° angle of incidence. For consistency, all coordinates in the earlier equations are converted to GPS-based coordinates, including longitude, latitude, and altitude. I_w is the value of the binarized pixel, set to zero for pixels within building outlines, implying no measurable GPS signal ($|S_\Omega(P_w, O_0)| \leqslant \xi$) underneath a building. The network is expected to recognize these as concrete structures. Conversely, when $I_w = 1$, the actual GPS signals should align with the predicted values. This process essentially projects the predicted 3D map onto a 2D plane, facilitating a comparison with the actual 2D map, particularly regarding building locations.

7.6.7 Summary

Finally, we put the pieces together to get the joint training loss function as follows:

$$\mathcal{L} = (1 - \lambda_1 - \lambda_2)\mathcal{L}_{\text{gps}} + \lambda_1 \mathcal{L}_{\text{map}} + \lambda_2 \cdot \zeta(|S_{\text{RX}}(p_0, s_k)|) \tag{7.16}$$

where ζ is a beta-distribution regularizer introduced by [33,152], λ_1 and λ_2 are hyperparameters. This loss function aims to minimize the difference between the measured and predicted GPS signals at the smartphone's positions. Meanwhile, it obtains hints

about the boundaries and shapes of the building from the 2D map. Certainly, the map loss is optional if the 2D map is not available.

7.7 3D Map Reconstruction

Materials with higher density tend to cause more significant attenuation of RF signals. This implies a direct correlation between a voxel's density and the level of attenuation it imparts on a GPS signal. We thus define the relative density of a voxel as follows:

$$\rho(V_i) = |\exp\big(\delta(V_i)\big)| = \frac{1}{\Delta a(V_i)} \tag{7.17}$$

where $\Delta a(V_i)$ indicates the attenuation of the RF signal that traverses through the voxel V_i. The density of a voxel is inversely proportional to the amount of signal it absorbs. The relative densities of all voxels are acquired from the attenuation subnetwork. By establishing a threshold value, denoted as ρ_{air}, we classify a voxel to be a component of a building when its relative density $\rho(V_i)$ exceeds ρ_{air}. If not, the voxel is identified as air. This approach enables the reconstruction of buildings within the scene by omitting voxels identified as air, thereby effectively differentiating between solid structures and open areas. The final step involves identifying a suitable bounding box that encompasses all the non-air voxels, facilitating the creation of a comprehensive 3D map.

Progressive Training. Employing fine-grained uniform voxelization enhances the model's resolution and accuracy. However, the sheer number of voxels poses a challenge for computational efficiency, especially during ray marching procedures. Consider a scenario where each voxel measures 10 cm^3. In a 100 m^3 scene, this results in a billion voxels. Consequently, a single ray marching along a direction could involve interactions with over a thousand voxels, leading to an excessively high computational burden. Moreover, many areas in a typical scene are 'vacant,' filled predominantly

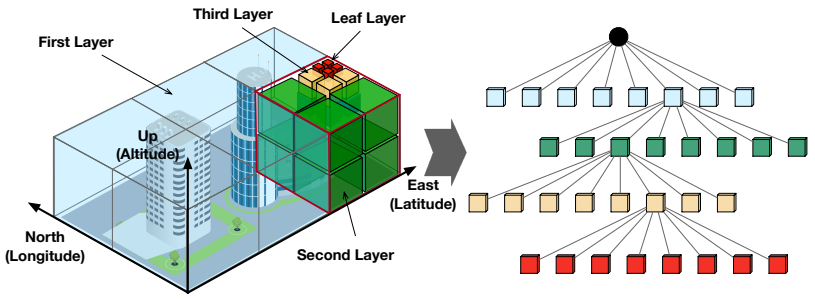


Fig. 7.8: Octree-based Voxelization

with air, contributing minimal attenuation to the ray marching process. To address this, we adopt a hierarchical data structure known as an octree for representing the 3D scene as shown in Fig. 7.8. The fundamental principle of an octree is to recursively subdivide space into eight smaller segments, or 'octants.' Each octant corresponds to a node in the tree structure. The tree is structured such that each node either has eight child nodes or none, with the root node encapsulating the entire space and each subsequent level representing a division into smaller subspaces. Importantly, each node in this structure contains numerous voxels. To build an octree in local coordinate, we transform the coordinate from LLA (Longitude, Latitude, Altitude) to ENU (East, North, Up), in which we select a reference point as the origin, east is the x-axis, north is the y-axis, and up is the z-axis.

7.8 Results

In this section, we evaluate the performance of the 3D map reconstruction using SaRF.

7.8.1 Implementation

In our experiment, we set $\lambda_1 = 0.01$ and $\lambda_2 = 0.001$. We use a batch size of 16 coupled with a cosine learning rate scheduler that varies between 10^{-4} and 10^{-6}. The

Table 7.1: Annotated Voxels Description

Scenes (#)	A	B	C	D	E	F
Records	17.1M	2.6M	1.8M	5.8M	121K	617K
Building	14,012	12,245	13,680	19,097	12,371	7,026
Vacant	23,428	18,811	30,864	22,999	13,069	23,598
Total	37,440	31,056	44,544	42,096	25,440	30,624

loss is computed for each sample. We use Adam to optimize the loss function. The direction space Ω is sampled to $36 \times 9 = 324$ directions. We conducted our training over approximately 500,00 iterations, which took about 10 hours on a single NVIDIA GTX 4090. We start by creating a balanced octree with a root node size of 200 m with coarse leaf nodes of 6.25 m. Nodes can be further subdivided down to the smallest unit, a 0.39 m voxel, if its relative density exceeds $\rho_{\text{air}} = 0.5$. If not, the node undergoes self-pruning. This method not only maintains the model's high resolution where necessary but also significantly reduces computational load in areas with little to no significant content.

Ground Truth. For the ground truth 3D map of the campus, we utilized Google Earth for its high-resolution 3D building models and footprints. Buildings and open spaces were manually annotated with voxels, as shown in Table 7.1, which lists the number of voxels annotated. "Records" means the amount of collected GNSS data in the scenes. Voxels within buildings were labeled as "Building", while those in open spaces near buildings were marked as "Vacant". These labels, numbering over 25,000 per scene, are crucial for assessing the reconstruction accuracy of our model.

7.8.2 Accuracy of Satellite SNR Prediction

Evolving from NeRF^2, which is capable of predicting received signals at any given location, SaRF has also been endowed with the same predictive capability. A higher level of prediction accuracy signifies a more precise fit of the neural radiance fields, as the predictions emerge from ray tracing by using the two MLPs. Therefore, our

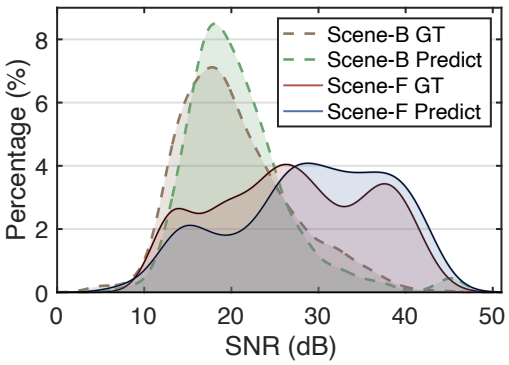


Fig. 7.9: PDFs of SNR Prediction

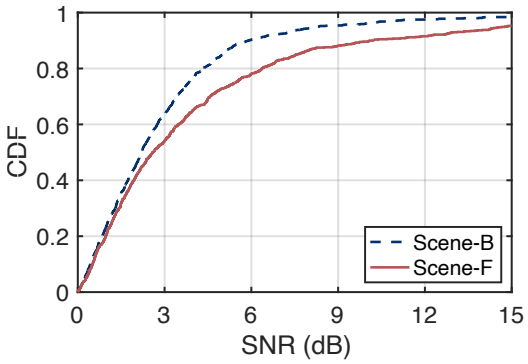


Fig. 7.10: CDFs of SNR Prediction

initial evaluation of SaRF focuses on its accuracy in predicting the GPS SNR. During our experiments, we randomly selected 80% of the datasets from Scenes B and F for training, reserving the remaining 20% for testing purposes.

The results of the predicted SNR, when compared with the ground truth (GT) and represented as a PDF, are depicted in Fig. 7.9. The prediction closely mirrors the actual SNR distribution. Notably, the SNR predominantly ranges between 10 and 30 dB for Scene B, while it spans a broader spectrum of 10 – 40 dB for Scene F, attributable to the latter's more extensive area coverage. Further quantifying the accuracy, we computed the SNR errors, defined as the absolute deviation between the predicted SNR and the ground truth. The CDF of these prediction errors is presented in Fig. 7.10. For Scene B, we attained a median error of 2.3 dB (with the 10^{th} percentile at 0.4 dB and the 90^{th} percentile at 5.9 dB), and for Scene F, the median error was 2.2 dB (with the 10^{th} percentile at 0.6 dB and the 90^{th} percentile at 10.3 dB). These findings affirm that SaRF possesses a high degree of accuracy in modeling GPS signal propagation, which is advantageous for reconstructing 3D maps.

7.8.3 Accuracy of Reconstruction

We extended our evaluation to assess building reconstruction performance using SaRF and the SenseMyCity dataset. To measure the outcomes, we employed balanced

Table 7.2: Accuracy of Reconstruction

	SaRF			SenseMyCity		
Dataset	**Acc. ↑**	**Rec. ↑**	**Pre. ↑**	**Acc. ↑**	**Rec. ↑**	**Pre. ↑**
Scene A	85.8%	77.8%	**92.5%**	78.5%	73.2%	49.5%
Scene B	85.1%	79.4%	89.8%	81.5%	69.5%	76.5%
Scene C	80.9%	92.8%	70.9%	68.0%	38.0%	81.0%
Scene D	73.7%	59.7%	83.0%	63.3%	33.5%	85.3%
Scene E	84.0%	**96.3%**	70.6%	82.5%	93.8%	61.0%
Scene F	**89.2%**	83.9%	86.0%	–	–	–
Avg	83.1%	81.7%	82.1%	74.8%	61.6%	70.7%

accuracy (Acc.), precision (Pre.), and recall (Rec.) as principal metrics. Accuracy is a universal metric reflecting the proportion of voxels accurately classified. This metric treats all voxels with equal weight, which can lead to skewed results in datasets with imbalances, like a predominance of "vacant" voxels over "Building" voxels. Balanced accuracy, defined as $\frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP+FN}} + \frac{\mathrm{TN}}{\mathrm{TN+FP}}\right)$, addresses this by computing the mean of the correct classification rates for each class separately, thereby adjusting for any disparities in class distribution. Precision, denoted as Precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, evaluates the correct identification rate of building voxels within the octree, with TP being true positives and FP false positives. Recall, defined as Recall $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, measures the model's success in correctly identifying actual building voxels, indicative of its comprehensive detection ability. Higher values for these three metrics are preferable.

Partial Analysis. Table 7.2 showcases the reconstruction performance comparison between SaRF and SenseMyCity, with the latter's results sourced from [140] (Scene F is unavailable to SenseMyCity). Our findings indicate that (1) SaRF surpasses SenseMyCity with an average balanced accuracy, recall, and precision of 83.1%, 81.7%, and 82.1%, respectively, while SenseMyCity, utilizing the Random Forest classifier, achieves 74.8%, 61.6%, and 70.7%. This underscores SaRF's superior performance in precise classification and detection of building voxels, along with more effective voxel pruning within the octree framework. (2) SaRF attains peak balanced accuracy of 89.2% in Scene F, a peak recall of 96.3% in Scene E, and peak precision of 92.5% in

Scene A. These scenes, associated with residential and educational environments, are characterized by fewer dynamic objects such as vehicles that could otherwise impact model precision.

Overall Analysis. For a comprehensive evaluation of SaRF's performance, we utilize the F1 score, which represents the harmonic mean of precision and recall and accounts for both metrics concurrently. Higher F1 scores are indicative of more accurate reconstruction. Fig. 7.11 shows a comparison of the F1 scores between SaRF and SenseMyCity across five different scenes. The data reveals that (1) SaRF nearly attains F1 scores exceeding 80%, with specific scores of 84.5%, 84.3%, 80.4%, 69.4% and 81.5% for Scene A, B, C, D and E, respectively. On the other hand, SenseMyCity's F1 scores are comparatively lower, at 50.7%, 68.0%, 47.8%, 32.3%, and 72.3%, respectively. SaRF outperforms SenseMyCity by 25.6%. (2) The F1 scores in Scene D are notably lower, a consequence of particularly poor recall in that scene (59.7% and 33.5%). In Scene D, which features a complex architectural layout with irregular buildings and ambiguous open spaces, numerous voxels are misclassified from open air to building, impacting the accuracy.

Summary. SaRF surpasses existing state-of-the-art methodologies due to two primary factors. Firstly, prior approaches rely solely on line-of-sight (LOS) assumptions to deduce the presence of buildings, neglecting non-line-of-sight (NLOS) signal paths. This simplification is problematic since distinguishing LOS from NLOS components in received signals is nearly impossible, often rendering the assumption inaccurate. In contrast, SaRF fully accounts for all potential signal paths using the ray marching algorithm. Secondly, SaRF enhances prediction accuracy by incorporating the physical properties of objects encountered along signal paths–something not considered in previous models. These properties are finely represented through the dual MLPs within our system.

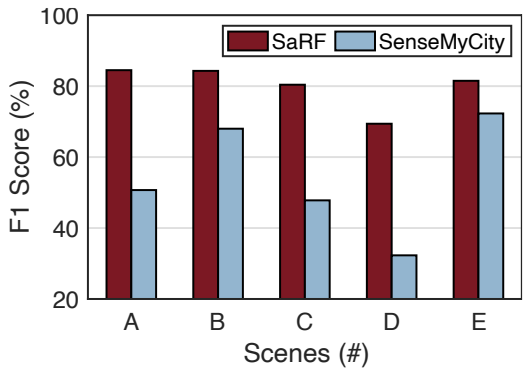


Fig. 7.11: Performance of F1 score

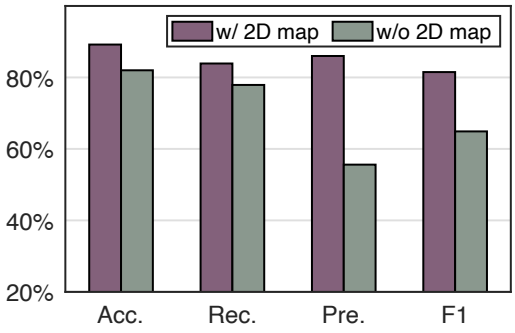


Fig. 7.12: Ablation Study

7.8.4 Ablation Study

An ablation study was performed to assess the contribution of $\mathcal{L}_{\text{map}}$ to the model's performance using Scene F as the test case. Specifically, we evaluated the performance of SaRF with and without the integration of the 2D map in the training process, focusing on balanced accuracy, precision, recall, and F1 score. The findings revealed that without the 2D map, SaRF's performance metrics experienced a drop: accuracy fell to 82.0%, recall to 77.9%, precision to 55.6%, and F1 score to 64.9%. These figures reflect declines of 7.2%, 6.0%, 30.4%, and 16.6%, respectively, when compared to the results achieved with the inclusion of $\mathcal{L}_{\text{map}}$. The most significant reductions were observed in precision and the F1 score, underscoring the 2D map's pivotal role in curbing false positives, where non-building voxels are incorrectly identified as part of a building. Hence, the integration of $\mathcal{L}_{\text{map}}$ into the training markedly curtails false positives, thereby bolstering the precision and overall performance of SaRF.

7.8.5 Impact of Altitude

Then, we assessed the balanced accuracy of SaRF across different voxel altitudes within the reconstructed 3D map. The evaluation was carried out on Scene A (Library) and Scene F (Campus), calculating the average accuracy for reconstructed voxels at different height levels. The campus scene featured building heights around

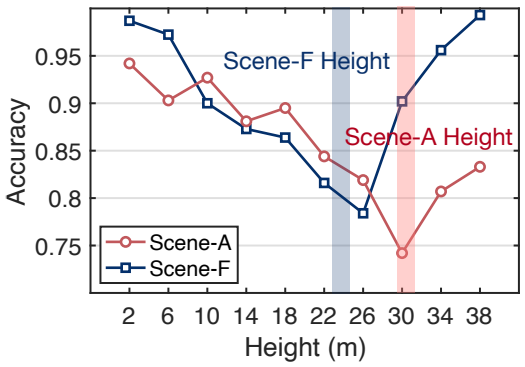


Fig. 7.13: Impact of Altitude

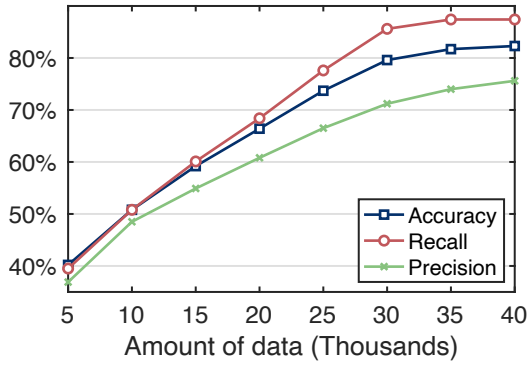


Fig. 7.14: Impact of Data Amount

23 m, while the library scene had buildings approximately 31 m tall. We analyzed accuracy for voxel heights from 2 m to 38 m, in increments of 4 m, as depicted in Fig. 7.13. Our results indicated that accuracy fluctuated across different altitudes. Typically, accuracy peaked at ground level, then gradually diminished as height increased, up to a point close to the building's height, before rebounding. For example, accuracy was (98.1%, 94.3%) at 2m, dipped to (78.4%, 74.2%) at 23m and 30m, and then increased to (99.8%, 83.6%) at 38m. The average accuracy across all layers was 89.2% for Scene A and 85.8% for Scene F.

The pattern of varying accuracy across different voxel altitudes can be explained by the limitations of the ray tracing technique. GPS signals are predominantly captured at ground level, resulting in ray tracing predominantly occurring in upward directions from the ground. Particularly during directional tracing, voxels nearer to the ground are more frequently traced, while those at higher altitudes may be traced less often. This occurs because the separation between two distinct directional paths increases with altitude, leading to less effective tracing of higher voxels. Additionally, the standard practice of assuming voxels to be vacant by default contributes to improved accuracy at levels above the actual height of buildings, thereby enhancing the accuracy observed in the upper layers of the reconstruction.

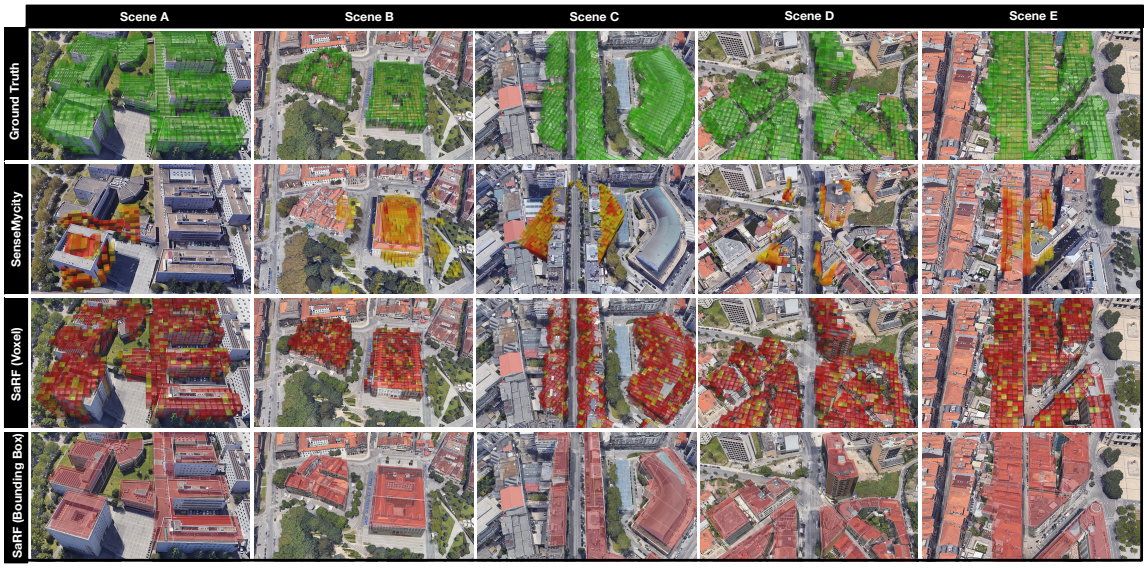


Fig. 7.15: Illustration of 3D Urban Constructions. The first row displays the ground truth for five scenes, with 3D building models sourced from Google Earth and voxels labeled manually. The second row illustrates the reconstruction outcomes from the related project, SenseMyCity. The third row presents the 3D voxels reconstructed using SaRF. The fourth row depicts the bounding boxes of the reconstructed buildings, where each box circumscribes relevant voxels.

7.8.6 Impact of Training Scale

Next, we assessed how varying amounts of data affect the performance of 3D building reconstruction in SaRF. For this evaluation, we selected a building in the Scene B. Fig.7.14 illustrates the changes in accuracy, recall, and precision as the volume of data increments from 5K to 40K in steps of 5K. Initial results with 5K data show SaRF achieving 40.2% accuracy, 36.9% recall, and 39.5% precision. As the data volume increases, these metrics progressively improve. However, beyond 30K data, the rate of performance enhancement slows down. At 40K data, SaRF reaches 82.3% accuracy, 75.6% recall, and 87.4% precision. Compared to the full dataset training results (approximately 120K), detailed in Table7.2, there is a performance decrease of 4.1%, 4.8%, and 2.8% in accuracy, recall, and precision, respectively. Considering the balance between data collection efforts and performance efficiency, we recommend 40K data as an optimal amount for effective building reconstruction.

7.8.7 Visualization

Finally, we visualize the reconstruction results in Fig. 7.15. We employ the City3D toolbox [153] to draw the recognized building voxels and bounding boxes on Google Earth. In the model, red and yellow represent the higher and lower-density building voxels, respectively. The visualization distinctly shows that our recognition results are more detailed and precise compared to those achieved by SenseMyCity. Regardless of the coverage or accuracy, SaRF outperforms SenseMyCity.

7.9 Related Work

Our work is related to the following three categories:

Crowdsensing. Crowdsensing has become a popular method for conducting large-scale, diverse data across various domains. It's used to many applications from location and driving information [154–157], spectrum monitoring with smartphones [158, 159], construct Wi-Fi maps [160]. However, these systems often face challenges related to the quality of data, especially when the collection process is opportunistic and lacks controlled environments [160]. In addressing these challenges, recent research has proposed various methodologies. For instance, Fang et al. explore pervasive vehicular sensing to refine urban map inference, which can potentially enhance the accuracy and reliability of location-based data [135]. Uvlens [161] integrates crowdsourced data with open government datasets to identify urban village boundaries and estimate populations, which showcases the potential of crowdsourcing in urban planning. Additionally, MultiCell develops a model for urban population dynamics using multiple cellphone networks, offering a novel approach to demographic analytics [162]. Wang et al. focus on constructing a cellular signal map, which serves to improve network coverage analysis through mobile crowdsensing [163]. SaRF employs crowdsourced GNSS data from mobile devices to construct a 3D occupancy map.

GNSS-based 3D Mapping. RF signals have demonstrated the capacity to perceive the environment beyond their original applications, extending to areas such as occupancy detection [164–166], human activity recognition [167, 168], localization [169] and imaging [170]. 3D city maps, crucial for applications ranging from network planning to climate studies [171–173], have traditionally relied on expensive methods such as manual annotation and depth-sensing technologies [21, 22, 174, 175]. However, GNSS signals, commonly used for positioning, have transformed urban mapping. They provide a globally accessible, economical option for 3D mapping, enhanced by smartphone-based passive data collection [176, 177]. GNSS-based 3D mapping, employing building signal obstruction and GPS for urban reconstruction, shows significant potential [139, 178]. The integration of crowdsourced data and probabilistic methods using GNSS SNR measurements further demonstrates the adaptability of GNSS for urban modeling [138, 140, 154, 179], mitigating the inaccuracies inherent in crowdsourced data, thereby improving the reliability and precision of the urban models generated. While GNSS mapping holds promise, its accuracy faces challenges from signal strength classification and processing, especially in complex urban settings. Unlike prior methods using SNR-based classifiers for identifying building voxels, SaRF leverages an RF model-informed neural radiance field to discern voxel attenuation properties, thereby improving accuracy.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

Wireless technologies have attracted enormous attention from around the world with the development of related technologies. In wireless systems, the wireless channel is crucial for both the communication and sensing tasks. As the increment of massive antenna arrays and frequency bands for higher capacity communications, the intriguing interaction between the RF signals and environments makes the wireless channel modeling and prediction becomes more challenges. Moreover, in a practical environment, the non-negligible noise and frequency offset will make it struggle in the mud. This dissertation analyzes the development and applications of neural radio-frequency radiance fields, which is a novel concept for wireless channel modeling and prediction. Specifically, we make the following contributions:

- *Neural Radio-Frequency Radiance Field:* We introduce NeRF^2, a pioneering deep learning framework specifically developed for wireless channel understanding. This framework incorporates a sophisticated physical model of electromagnetic wave transmission within its learning algorithm, enabling NeRF^2 to achieve a detailed un-

derstanding of wireless communication channels. To accurately predict the channel characteristics of wideband RF signals, we have developed frequency-aware NeRF^2, which includes an RF prism module along with specialized optimization techniques. Our extensive experiments demonstrate that NeRF^2 substantially enhances performance in critical deep learning-based application-layer tasks, particularly in indoor localization and massive MIMO communication systems.

- *Consistent Phase Estimation Protocol:* This work presents a long-range, high-accurate tracking system for commercial backscatter tags. It develops a phase estimation protocol to solve the problem of π-ambiguity reported by the commercial devices. In addition, we also proposed three denoising measures for antenna array equipped backscatter gateways. This innovative combination enables precise AoA estimation and localization across the entire communication range, a significant advancement in the field of backscatter tag tracking. A detailed prototype evaluation reveals a 3× coverage improvement in high-accurate tracking compared to the state-of-the-art works.
- *Understanding Localization by a Tailored GPT:* This paper presents a Transformer-based Localization (TBL) model designed for wireless indoor localization encompassing RFID, Wi-Fi, and BLE technologies. The hierarchical structure of TBL consists of multiple A-Subnetworks, each tasked with determining the AoA from different base stations. These are then collectively processed by the T-Subnetwork to predict the localization results along with historical positions. TBL outperforms existing methods in 50 different scenarios. To enhance the generalizability of the TBL model across scenarios, we introduce LocGPT, which is pre-trained on 1.4 million data samples. It demonstrates the capability to maintain near-optimal accuracy even with considerably reduced datasets.
- We introduce SaRF, a framework designed for 3D urban mapping utilizing crowd-sourced GNSS data. The key of SaRF lies in its innovative voxel-based representation of urban structures, coupled with the use of radio frequency neural radiance fields for learning the attenuation properties of these voxels. This unique combina-

tion allows our framework to efficiently and accurately reconstruct urban building structures based on GNSS data, representing a significant advancement in the field of urban mapping.

8.2 Future Works

8.2.1 Adaptation for Dynamics Environments

NeRF^2 employs an implicit model to encapsulate the radiance fields of RF signals within a specified scene. The primary advantage of utilizing an implicit model lies in its independence from a rigid structural model for ray tracing, thereby enabling the integration of voxel radiance data through numerical integration to achieve precise outcomes. Nevertheless, a limitation of this approach is its inflexibility in adapting to dynamic environmental changes. Once the RF radiance fields are established, altering the physical setup—such as relocating a table—poses a challenge, as the implicit model struggles to identify and adjust the affected voxels within the radiance model.

Potential Direction I: Adaptation via Explicit Neural Representations

To address these limitations, the adoption of explicit representations such as neural signed distance functions, neural voxel grids, or 3D Gaussian splatting offers a more structured and adaptable approach to environmental modeling.

- **Neural Signed Distance Functions (SDFs):** Neural signed distance functions provide a powerful method for representing complex geometries and scenes through continuous scalar fields. In a neural SDF, each point in space is associated with the shortest distance to the surface boundary, with the sign indicating whether the point is inside or outside the object. This representation allows for precise

boundary delineation and robust handling of topological transformations, which are invaluable for applications involving dynamic environments. The utilization of neural SDFs in RF signal modeling can lead to more accurate predictions of wave interactions with the environment, as the model dynamically adjusts to changes such as moving objects or altering structural layouts. Furthermore, the differentiability of SDFs facilitates the integration with gradient-based learning algorithms, enhancing the adaptability and speed of training processes in response to environmental changes.

- **Neural Voxel Grids**: Neural voxel grids represent environments using a discretized three-dimensional grid where each voxel stores learned features or parameters that describe the local electromagnetic properties. The representation is similar to the SaRF. This approach allows for a straightforward integration of spatial data with deep learning architectures, enabling efficient large-scale scene representations. Voxel-based models excel in scenarios where high-resolution data is available, and precise control over local interactions is necessary. They are particularly adept at handling indoor environments with multiple obstructions and varying material properties. Additionally, the modularity of voxel grids makes them highly adaptable to changes within the scene, as updates can be localized to specific voxels affected by physical modifications, thereby preserving the integrity and accuracy of the overall model.
- **3D Gaussian Splatting (3DGS):** 3D Gaussian splatting extends the concept of voxelization by incorporating Gaussian functions to model the influence of each data point over its neighborhood, resulting in a smoother and more continuous representation of the scene. This technique is particularly useful for RF modeling as it naturally accommodates the diffuse and scattered nature of RF signals. The use of Gaussian functions helps in approximating the gradual changes in signal strength and properties across space, providing a more nuanced understanding of signal behavior in complex environments. The continuous nature of the Gaussian splat model also means that updates to the environment, such as the movement of

objects or changes in layout, can be smoothly integrated into the existing model without the need for extensive retraining or recalibration.

Each of these explicit representations offers unique advantages for modeling dynamic environments in RF signal propagation. Their integration into the initialization process of environmental models not only enhances the precision and adaptability of the system but also ensures that the models remain relevant and accurate even as the physical setting evolves. These advancements represent significant strides towards creating more responsive and robust systems for real-time RF-based applications, such as autonomous navigation, interactive simulations, and smart infrastructure management.

Potential Direction II: Adaptation via Differentiable Ray Tracing

Exploring the integration of differentiable ray tracing technologies, such as NV Sionna, presents an innovative method for simulating the intricate interactions of electromagnetic waves within complex environments. Differentiable ray tracing (DRT) stands out by enabling the computation of gradients for both the geometry and material properties of a scene relative to the path of electromagnetic signals. This capability not only facilitates more accurate simulations but also enhances the adaptability of the models to dynamic changes within the environment.

- **Initial Geometry Acquisition through Inverse Rendering.** The first step in employing DRT effectively involves using inverse rendering techniques to ascertain the geometric structure of the environment. Inverse rendering works by reconstructing the three-dimensional geometry of the scene from observed data, typically captured through various sensors or imaging devices. This process essentially inverts the traditional rendering pipeline, utilizing observed measurements to deduce physical properties of the scene. By accurately reconstructing the geometry,

the foundation is laid for subsequent detailed analyses and simulations, providing a robust framework upon which further properties can be built and refined.

- **Material Property Learning via Ray Tracing.** Once the geometry is established, DRT can be utilized to learn the material properties of entities within the scene. This involves tracing simulated rays through the environment and adjusting material properties based on how these rays interact with different surfaces. The differentiation aspect of DRT allows for the adjustment of these properties by backpropagating errors from observed versus simulated interactions, optimizing material characteristics to reflect real-world behaviors. This step is crucial for understanding and predicting complex interactions such as reflection, absorption, and scattering, which are vital for accurate channel modeling in RF applications.
- **Dynamic Adaptation and Path Tracing.** After the electromagnetic (EM) properties of the scene's materials have been learned, the system is equipped to handle dynamic changes efficiently. In dynamic environments, where objects may move or where new structures might be introduced, DRT can dynamically trace new paths for the electromagnetic waves, recalculating interactions based on the updated scene configuration. This ability to adapt in real-time is pivotal for applications such as real-time gaming, autonomous vehicle navigation, and urban planning, where environments are continually changing.

The development of algorithms that integrate DRT with existing neural representations marks a significant advancement in predictive performance of channel models. These algorithms would leverage the learned EM properties and geometrical data to predict signal behavior with high accuracy, even in environments characterized by complex physical interactions. The validation of these algorithms involves rigorous testing across a variety of scenarios to ensure they not only meet theoretical expectations but also perform robustly in practical applications. By refining the interaction models between electromagnetic waves and environmental features, DRT can significantly enhance the predictive accuracy of channel models. This improvement is

particularly beneficial in scenarios involving complex geometries and diverse material types, where traditional models may fail to capture the nuanced behaviors of EM waves.

Potential Direction III: Adaptation via Temporal Model

To further enhance the adaptability of RF signal models in dynamic environments, incorporating time as an additional input can provide crucial insights into the temporal dynamics of electromagnetic wave interactions. A temporal model could be used to track and model the evolution of RF signal behaviors over time. This would allow the system to not only account for spatial changes but also adapt in real-time as the environment evolves.

By extending the concept of NeRF^2 to include time as a parameter, dynamic NeRF^2 can be used to model the continuous evolution of the scene's electromagnetic properties over time. Each voxel or radiance field element is parameterized by both spatial coordinates and temporal information, allowing the model to account for dynamic changes such as moving objects, fluctuating environmental conditions, or time-of-day effects. This extension makes the model responsive not only to static changes but also to periodic or transient environmental transformations. The temporal component enables more accurate modeling of time-varying RF behaviors, such as signal fading, interference patterns, or the movement of obstacles that alter wave propagation. Environmental elements and their material properties can change over time—due to factors such as temperature variations, humidity, or motion. A temporal model can incorporate these time-dependent variations into the RF propagation model. This adaptation is crucial for applications that require real-time updates to signal propagation predictions, such as autonomous vehicle navigation or dynamic wireless communication networks. By dynamically adjusting material properties, the system can more accurately reflect the changing physical and electromagnetic characteristics of

the environment.

8.2.2 Time Consumption Reduction

Potential Direction I: Compression for Lightweight Models

Recent advancements in model compression techniques have revealed effective strategies for accelerating the training and deployment of NeRF^2 models, which is essential for applications requiring fast updates and quick deployment. Two of the most impactful methods in this area are pruning and quantization, both of which aim to improve the computational efficiency of these complex neural networks.

Pruning is a technique that involves systematically removing less important weights from a neural network, thereby reducing its overall complexity. By focusing computational resources on the most influential parts of the model, pruning not only accelerates training but also simplifies the architecture. This results in a model with fewer parameters to manage, which in turn leads to faster load times and reduced latency during inference. Pruning is especially valuable in scenarios where models need to be updated incrementally, as it allows systems to adapt more swiftly to changes in the environment or data. This agility enhances both the efficiency and effectiveness of NeRF^2 applications in dynamic settings.

Quantization can address the computational demands of NeRF^2 models by converting high-precision parameters (e.g., 32-bit floating point) to lower precision formats, such as 16-bit or 8-bit integers. This reduction in precision lowers the computational and memory overhead, making the model lighter and more efficient. Additionally, quantization enables the use of specialized hardware accelerators like FPGAs or AI chips, which are optimized for handling lower-precision calculations. The reduced computational load speeds up training and inference, while also decreasing energy consumption, making the system more sustainable and cost-effective.

Integrating these compression techniques into the NeRF^2 training pipeline could drastically improve the deployment of these models in dynamic, real-time environments. By reducing downtime and enhancing responsiveness, pruning and quantization make NeRF^2 technologies more viable for applications where time, efficiency, and resource management are critical. As a result, these techniques not only make NeRF^2 models more accessible but also help optimize their performance for real-world applications.

Potential Direction II: Effective Rendering and Representation Strategies

Beyond model compression, targeted algorithmic enhancements significantly streamline NeRF^2 training processes, crucial for rapid and efficient model development. Subsampling techniques such as importance sampling are instrumental in this regard. By prioritizing the processing of data points that contribute most significantly to the model's learning, importance sampling effectively minimizes the computations on less critical data. This strategic data handling not only accelerates convergence but also conserves computational resources, crucial for training complex models like NeRF^2. The selective processing ensures that the training focus is maintained on data that enhance model accuracy and learning speed, optimizing the overall training cycle.

Further enhancing NeRF^2's efficiency, advanced data structures such as hash tables and octrees play a pivotal role in optimizing memory usage and accelerating data access. For instance, multi-resolution hash tables adaptively manage the spatial hierarchy of data, allowing for quick adjustments to the model parameters throughout the training process. This method leverages the parallel processing capabilities of GPUs, facilitating a significant reduction in computation time while handling the extensive data volumes typical in NeRF^2 training. These data structures not only improve the speed but also enhance the spatial accuracy of the rendered models, crucial for applications requiring high fidelity and precision.

These algorithmic improvements extend beyond mere acceleration. They also bolster

the robustness and adaptability of NeRF^2 models, making them suitable for dynamic and real-time applications. Looking forward, the integration of these strategies with real-time processing requirements could be transformative, particularly in fields such as autonomous vehicle navigation, augmented reality, and dynamic scene reconstruction. By marrying effective rendering techniques with efficient data representation, future research could unlock new potentials for NeRF^2, enabling it to operate seamlessly in real-world scenarios that demand both speed and accuracy. This evolution in NeRF^2 training methodologies could potentially set new standards for performance in complex, interactive environments.

8.2.3 Scaling for Broader Applications

Potential Direction I: City-scale 5G Digital Twin Network

A promising direction for extending the capabilities of NeRF^2 is the development of city-scale 5G digital twin networks. By combining detailed 3D models of urban environments, we can simulate RF signal propagation across entire cities in real-time. These digital twins would offer a highly accurate, dynamic, and scalable representation of how 5G signals interact with the complex geometries of urban infrastructure, including buildings, roads, vegetation, and moving objects. With these models, we could simulate various propagation effects such as reflections, diffractions, and scattering, leading to more precise predictions of signal strength, coverage, and interference in different parts of the city. This would significantly aid in network planning, optimization, and troubleshooting, allowing for proactive management of urban 5G networks and ensuring reliable service in high-demand areas. One possible solution is to leverage Google Street Map data to train a visual NeRF model and then transfer this model to NeRF^2 model for 5G signal prediction.

Potential Direction II: mmWave and THz bands Adaptation

Adapting NeRF^2 models to mmWave and THz frequency bands presents challenges due to their increased signal attenuation, susceptibility to absorption, and sensitivity to environmental obstacles like wall. These higher frequency bands also experience more pronounced diffraction and scattering effects, requiring high spatial resolution to accurately model signal propagation. To address these challenges, solutions include enhancing the NeRF^2 framework with finer spatial resolution and incorporating more granular material properties to capture the complex interactions at these frequencies. Additionally, leveraging visual data could improve model accuracy and adaptability. Advanced machine learning techniques, including transfer learning and domain adaptation, can also help mitigate the limited availability of large datasets for these bands, enabling the model to better generalize to diverse urban environments and dynamic propagation conditions. These strategies would improve the accuracy of mmWave and THz signal predictions, aiding in more efficient network design and optimization in challenging environments.

References

[1] Precedence Research, "Internet of things market size, share, growth analysis report," https://www.precedenceresearch.com/internet-of-things-market, 2023.

[2] K. N. Choi, H. Kolamunna, A. Uyanwatta, K. Thilakarathna, S. Seneviratne, R. Holz, M. Hassan, and A. Y. Zomaya, "Loradar: Lora sensor network monitoring through passive packet sniffing," *ACM SIGCOMM Computer Communication Review*, vol. 50, no. 4, pp. 10–24, 2020.

[3] G. Valecce, P. Petruzzi, S. Strazzella, and L. A. Grieco, "Nb-iot for smart agriculture: Experiments from the field," in *Proc. of IEEE CoDIT*, vol. 1. IEEE, 2020, pp. 71–75.

[4] D. Yang, X. Huang, J. Huang, X. Chang, G. Xing, and Y. Yang, "A first look at energy consumption of nb-iot in the wild: Tools and large-scale measurement," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2616–2631, 2021.

[5] Z. Chang, F. Zhang, J. Xiong, J. Ma, B. Jin, and D. Zhang, "Sensor-free soil moisture sensing using lora signals," *Proc. of ACM IMWUT*, vol. 6, no. 2, pp. 1–27, 2022.

[6] R. K. Kodali, S. Yerroju, and S. Sahu, "Smart farm monitoring using lora enabled iot," in *Proc. of IEEE ICGCIoT*. IEEE, 2018, pp. 391–394.

[7] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A survey on lora for smart agriculture: Current trends and future perspectives," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3664–3679, 2022.

[8] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proc. of ACM SIGCOMM*, 2015, pp. 269–282.

[9] R. Ayyalasomayajula, A. Arun, C. Wu, S. Sharma, A. R. Sethi, D. Vasisht, and D. Bharadia, "Deep learning based wireless localization for indoor navigation," in *Proc. of ACM MobiCom*, 2020, pp. 1–14.

[10] Y. Xie, Y. Zhang, J. C. Liando, and M. Li, "Swan: Stitched wi-fi antennas," in *Proc. of ACM MobiCom*, 2018.

[11] R. Ayyalasomayajula, D. Vasisht, and D. Bharadia, "Bloc: Csi-based accurate localization for ble tags," in *Proc. of ACM CoNEXT*, 2018, pp. 126–138.

[12] M. Cominelli, P. Patras, and F. Gringoli, "Dead on arrival: An empirical study of the bluetooth 5.1 positioning system," in *Proc. of WiNTECH*, 2019, pp. 13–20.

[13] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Proc. of ACM MobiSys*, 2019, pp. 313–325.

[14] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *Proc. of IEEE INFOCOM*. IEEE, 2015, pp. 1472–1480.

[15] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," in *Proc. of ACM MobiSys*, 2018, pp. 401–413.

[16] R. Gao, W. Li, Y. Xie, E. Yi, L. Wang, D. Wu, and D. Zhang, "Towards robust gesture recognition by characterizing the sensing quality of wifi signals," *Proc. of ACM IMWUT*, vol. 6, no. 1, pp. 1–26, 2022.

[17] Q. Pan, Z. An, X. Yang, X. Zhao, and L. Yang, "Rf-dna: Large-scale physical-layer identifications of rfids via dual natural attributes," in *Proc. of ACM MobiCom*, 2022, pp. 419–431.

[18] Wikipedia, "Snell's law," https://en.wikipedia.org/wiki/Snell%27s_law, 2024.

[19] R. G. Kouyoumjian and P. H. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *Proceedings of the IEEE*, vol. 62, no. 11, pp. 1448–1461, 1974.

[20] R. Shen and Y. Ghasempour, "Scattering from rough surfaces in 100+ ghz wireless mobile networks: From theory to experiments," in *Proc. of ACM MobiCom*, 2023, pp. 1–15.

[21] "Electromagnetic Simulation Software," https://www.remcom.com/.

[22] E. Egea-Lopez, J. M. Molina-Garcia-Pardo, M. Lienard, and P. Degauque, "Opal: An open source ray-tracing propagation simulator for electromagnetic characterization," *Plos one*, vol. 16, no. 11, p. e0260060, 2021.

[23] T. L. Marzetta and B. M. Hochwald, "Fast transfer of channel state information in wireless systems," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1268–1278, 2006.

[24] X. Fan, L. Shangguan, R. Howard, Y. Zhang, Y. Peng, J. Xiong, Y. Ma, and X.-Y. Li, "Towards flexible wireless charging for medical implants using distributed antenna system," in *Proc. of ACM MobiCom*, 2020, pp. 1–15.

[25] D. Vasisht, S. Kumar, H. Rahul, and D. Katabi, "Eliminating channel feedback in next-generation cellular networks," in *Proc of ACM SIGCOMM*, 2016, pp. 398–411.

[26] Z. Liu, G. Singh, C. Xu, and D. Vasisht, "Fire: enabling reciprocity for fdd mimo systems," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 628–641.

[27] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "End-to-end wireless path deployment with intelligent surfaces using interpretable neural networks," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6792–6806, 2020.

[28] A. Bakshi, Y. Mao, K. Srinivasan, and S. Parthasarathy, "Fast and efficient cross band channel prediction using machine learning," in *Proc. of ACM MobiCom*, 2019, pp. 1–16.

[29] J. Hoydis, F. A. Aoudia, S. Cammerer, M. Nimier-David, N. Binder, G. Marcus, and A. Keller, "Sionna rt: Differentiable ray tracing for radio propagation modeling," *arXiv preprint arXiv:2303.11103*, 2023.

[30] X. Chen and X. Zhang, "Rf genesis: Zero-shot generalization of mmwave sensing through simulation-based data synthesis and generative diffusion models," in *Proc. of ACM SenSys*. Istanbul, Turkiye: ACM, New York, NY, USA, 2023, pp. 1–14. [Online]. Available: https://doi.org/10.1145/3625687.3625798

[31] T. Orekondy, P. Kumar, S. Kadambi, H. Ye, J. Soriaga, and A. Behboodi, "Winert: Towards neural ray tracing for wireless channel modelling and differentiable simulations," in *The Eleventh International Conference on Learning Representations*, 2022.

[32] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Proc. of ECCV*. Springer, 2020, pp. 405–421.

[33] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[34] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proc. of IEEE/CVF CVPR*, 2022, pp. 8248–8258.

[35] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *Proc. of IEEE/CVF CVPR*, 2022, pp. 12 932–12 942.

[36] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *Proc. of IEEE/CVF CVPR*, 2021, pp. 7495–7504.

[37] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proc. of IEEE/CVF CVPR*, 2021, pp. 7210–7219.

[38] C. Chen, A. Richard, R. Shapovalov, V. K. Ithapu, N. Neverova, K. Grauman, and A. Vedaldi, "Novel-view acoustic synthesis," in *Proc. of the IEEE/CVF CVPR*, 2023, pp. 6409–6419.

[39] B. Attal, E. Laidlaw, A. Gokaslan, C. Kim, C. Richardt, J. Tompkin, and M. O'Toole, "Törf: Time-of-flight radiance fields for dynamic scene view synthesis," *Advances in neural information processing systems*, vol. 34, pp. 26 289–26 301, 2021.

[40] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: Indoor location sensing using active rfid," in *Proc. of IEEE PerCom*. IEEE, 2003, pp. 407–415.

[41] Z. Yang, Z. Zhou, and Y. Liu, "From rssi to csi: Indoor localization via channel response," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1–32, 2013.

[42] J. Wang and D. Katabi, "Dude, where's my card? rfid positioning that works with multipath and non-line of sight," in *Proc. of ACM SIGCOMM*, 2013, pp. 51–62.

[43] L. Chen, K. Yang, and X. Wang, "Robust cooperative wi-fi fingerprint-based indoor localization," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1406–1417, 2016.

[44] J. Wang, J. Zhang, R. Saha, H. Jin, and S. Kumar, "Pushing the range limits of commercial passive rfids," in *Proc. of USENIX NSDI*, 2019, pp. 301–316.

[45] Y. Ma, Z. Luo, C. Steiger, G. Traverso, and F. Adib, "Enabling deep-tissue networking for miniature medical devices," in *Proc. of ACM SIGCOMM*. ACM, 2018, pp. 417–431.

[46] X. Tang, G. Xie, and Y. Cui, "Self-sustainable long-range backscattering communication using rf energy harvesting," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 737–13 749, 2021.

[47] A. Varshney and L. Corneo, "Tunnel emitter: Tunnel diode based low-power carrier emitters for backscatter tags," in *Proc. of ACM Mobicom*, 2020, pp. 1–14.

[48] C. Qi, F. Amato, M. Alhassoun, and G. D. Durgin, "A phase-based ranging method for long-range rfid positioning with quantum tunneling tags," *IEEE Journal of Radio Frequency Identification*, vol. 5, no. 2, pp. 163–173, 2020.

[49] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate rssi indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, 2019.

[50] S. Zhang, W. Wang, N. Zhang, and T. Jiang, "Rf backscatter-based state estimation for micro aerial vehicles," in *Proc. of IEEE INFOCOM*, 2020, pp. 209–217.

[51] J. Jiang, J. Wang, Y. Chen, Y. Liu, and Y. Liu, "Locra: Enable practical long-range backscatter localization for low-cost tags," in *Proc. of ACM MobiSys*, 2023, pp. 317–329.

[52] C. Yang, X. Wang, and S. Mao, "Rfid tag localization with a sparse tag array," *IEEE internet of things journal*, vol. 9, no. 18, pp. 16 976–16 989, 2021.

[53] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proc. of USENIX NSDI*, 2013, pp. 71–84.

[54] S. Zhang, W. Wang, S. Tang, S. Jin, and T. Jiang, "Robot-assisted backscatter localization for iot applications," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5807–5818, 2020.

[55] Y. Fu, H. Zhang, and H. Zhang, "Bluetooth aoa positioning based on backscatter technology," in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*. IEEE, 2020, pp. 559–565.

[56] B. Tao, H. Wu, Z. Gong, Z. Yin, and H. Ding, "An rfid-based mobile robot localization method combining phase difference and readability," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1406–1416, 2020.

[57] R. Miesen, A. Parr, J. Schleu, and M. Vossiek, "360 carrier phase measurement for uhf rfid local positioning," in *Proc. of IEEE RFID-TA*. IEEE, 2013, pp. 1–6.

[58] D.-T. Phan-Huy, D. Barthel, P. Ratajczak, R. Fara, M. Di Renzo, and J. De Rosny, "Ambient backscatter communications in mobile networks: Crowd-detectable zero-energy-devices," *IEEE Journal of Radio Frequency Identification*, vol. 6, pp. 660–670, 2022.

[59] Z. An, Q. Lin, Q. Pan, and L. Yang, "Turbocharging deep backscatter through constructive power surges with a single rf source," in *Proc. of IEEE INFOCOM*, 2021, pp. 1–10.

[60] A. Abedi, F. Dehbashi, M. H. Mazaheri, O. Abari, and T. Brecht, "Witag: Seamless wifi backscatter communication," in *Proc. of ACM SIGCOMM*, 2020, pp. 240–252.

[61] S. Chen, M. Zhang, J. Zhao, W. Gong, and J. Liu, "Reliable and practical bluetooth backscatter with commodity devices," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1717–1729, 2021.

[62] Z. Xu and W. Gong, "Bumblebee: Enabling the vision of pervasive zigbee backscatter communication," in *In Proc. of IEEE PerCom*, 2023, pp. 252–261.

[63] M. Hessar, A. Najafi, and S. Gollakota, "Netscatter: Enabling large-scale backscatter networks," in *Proc. of USENIX NSDI*, 2019, pp. 271–284.

[64] K. Xu, W. Gong, Y. Li, J. M. Purushothama, G. Goussetis, S. McLaughlin, J. S. Thompson, C. Song, and Y. Ding, "Fm rider: Two-fsk modulation-based ambient fm backscatter over 100 m distance," *IEEE Transactions on Microwave Theory and Techniques*, 2024.

[65] F. Dehbashi, A. Abedi, T. Brecht, and O. Abari, "Verification: can wifi backscatter replace rfid?" in *Proc. of ACM Mobicom*, 2021, pp. 97–107.

[66] M. Zhang, S. Chen, A. Nayak, and W. Gong, "Enabling multi-channel backscatter communication for bluetooth low energy," in *In Proc. of IEEE ICC*, 2020, pp. 1–6.

[67] P. Li, Z. An, L. Yang, and P. Yang, "Towards physical-layer vibration sensing with rfids," in *Proc. of IEEE INFOCOM*, 2019, pp. 892–900.

[68] S. J. Orfanidis, "Electromagnetic waves and antennas," 2002.

[69] ImpinJ, "Speedway revolution reader application note: Low level user data support," in *Speedway Revolution Reader Application Note*, 2010.

[70] J. Wang, D. Vasisht, and D. Katabi, "Rf-idraw: Virtual touch screen in the air using rf signals," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 235–246, 2014.

[71] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[72] T. B. Lavate, V. Kokate, and A. Sapkal, "Performance analysis of music and esprit doa estimation algorithms for adaptive array smart antenna in mobile communication," in *Proc. of IEEE ICCNT*, 2010, pp. 308–311.

[73] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices," in *Proc. of ACM MobiCom*, 2014, pp. 237–248.

[74] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE signal processing magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[75] S. A. Vorobyov, "Principles of minimum variance robust adaptive beamforming design," *Signal Processing*, vol. 93, no. 12, pp. 3264–3277, 2013.

[76] "USRP X310," https://www.ettus.com/all-products/x310-kit/, 2020.

[77] J. Kimionis, A. Bletsas, and J. N. Sahalos, "Increased range bistatic scatter radio," *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 1091–1104, 2014.

[78] S. Labs, "Direction finding using bluetooth low energy," Application Note, 2021. [Online]. Available: https://www.silabs.com/documents/public/application-notes/an1298-direction-finding-using-bluetooth-low-energy.pdf

[79] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[80] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," *ICCV*, 2021.

[81] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[82] "NeRF Demonstration," https://www.matthewtancik.com/nerf.

[83] P. Stoica, R. L. Moses *et al.*, *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ, 2005, vol. 452.

[84] Z. An, Q. Lin, P. Li, and L. Yang, "General-purpose deep tracking platform across protocols for the internet of things," in *Proc. of ACM MobiSys*, 2020, pp. 94–106.

[85] F. Euchner, M. Gauger, S. Dörner, and S. ten Brink, "A Distributed Massive MIMO Channel Sounder for "Big CSI Data"-driven Machine Learning," in *WSA 2021; 25th International ITG Workshop on Smart Antennas*, 2021.

[86] "RayTracing Toolbox," https://www.mathworks.com/help/antenna/ref/rfprop.raytracing.html.

[87] U. Ha, J. Leng, A. Khaddaj, and F. Adib, "Food and liquid sensing in practical environments using rfids," in *Proc. of USENIX NSDI*, 2020.

[88] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[89] P. Babakhani, T. Merk, M. Mahlig, I. Sarris, D. Kalogiros, and P. Karlsson, "Bluetooth direction finding using recurrent neural network," in *Proc. of IEEE IPIN*. IEEE, 2021, pp. 1–7.

[90] M. Comiter and H. Kung, "Localization convolutional neural networks using angle of arrival images," in *Proc. of IEEE GLOBECOM*. IEEE, 2018, pp. 1–7.

[91] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE/CVF CVPR*, 2016, pp. 770–778.

[92] C. Shepard, J. Ding, R. E. Guerra, and L. Zhong, "Understanding real many-antenna mu-mimo channels," in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 461–467.

[93] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: Practical many-antenna base stations," in *Proc. of ACM MobiCom*. ACM, 2012, pp. 53–64.

[94] C. Huang, G. C. Alexandropoulos, A. Zappone, C. Yuen, and M. Debbah, "Deep learning for ul/dl channel calibration in generic massive mimo systems," in *Proc. of IEEE ICC*. IEEE, 2019, pp. 1–6.

[95] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser mimo channels," *IEEE transactions on signal processing*, vol. 52, no. 2, pp. 461–471, 2004.

[96] "NRF52832," https://www.nordicsemi.com/products/nrf52832.

[97] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. of IEEE/RSJ IROS*. IEEE, 2020, pp. 5135–5142.

[98] H. Shin, Y. Chon, Y. Kim, and H. Cha, "Mri: Model-based radio interpolation for indoor war-walking," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1231–1244, 2014.

[99] F. Parralejo, F. J. Aranda, J. A. Paredes, F. J. Álvarez, and J. Morera, "Comparative study of different ble fingerprint reconstruction techniques," in *Proc. of IEEE IPIN*. IEEE, 2021, pp. 1–8.

[100] EPCGlobal, "Epcglobal uhf gen2 protocol standard," https://www.gs1.org/standards/rfid/uhf-air-interface-protocol, 2024.

[101] F. Zhang, B. Jin, Z. Lan, Z. Chang, D. Zhang, Y. Jiao, M. Shi, and J. Xiong, "Quantum wireless sensing: Principle, design and implementation," in *Proc. of ACM Mobicom*, 2023, pp. 1–15.

[102] "HMC241," https://www.analog.com/en/products/hmc241.html, 2021.

[103] "RFID Tag," https://support.impinj.com/hc/en-us/articles/202756908-Monza-4-RFID-Tag-Chip-Datasheet, 2019.

[104] C. Bocanegra, M. A. Khojastepour, M. Y. Arslan, E. Chai, S. Rangarajan, and K. R. Chowdhury, "Rfgo: a seamless self-checkout system for apparel stores using rfid," in *Proc. of ACM Mobicom*, 2020, pp. 1–14.

[105] "OptiTrack," https://optitrack.com/, 2021.

[106] Z. Wang, J. A. Zhang, F. Xiao, and M. Xu, "Accurate aoa estimation for rfid tag array with mutual coupling," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12 954–12 972, 2022.

[107] A. Badawy, T. Khattab, D. Trinchero, T. ElFouly, and A. Mohamed, "A simple angle of arrival estimation system," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[108] Y. Ma, N. Selby, and F. Adib, "Minding the billions: Ultra-wideband localization for deployed rfid tags," in *Proc. of ACM MobiCom*, 2017, pp. 248–260.

[109] Y. Xie, J. Xiong, M. Li, and K. Jamieson, "md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking," in *Proc. of ACM MobiCom*, 2019, pp. 1–16.

[110] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person motion tracking via rf body reflections," in *Proc. of USENIX NSDI*, 2015.

[111] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba, "Rf-based 3d skeletons," in *Proc. of ACM SIGCOMM*, 2018, pp. 267–281.

[112] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *Proc. of IEEE/CVF CVPR*, 2018, pp. 7356–7365.

[113] Y. Ma and E. C. Kan, "Accurate indoor ranging by broadband harmonic generation in passive nltl backscatter tags," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 5, pp. 1249–1261, 2014.

[114] X. Hui and E. C. Kan, "Radio ranging with ultrahigh resolution using a harmonic radio-frequency identification system," *Nature Electronics*, vol. 2, no. 3, p. 125, 2019.

[115] A. Haniz, G. K. Tran, K. Saito, K. Sakaguchi, J.-i. Takada, D. Hayashi, T. Yamaguchi, and S. Arata, "A novel phase-difference fingerprinting technique for localization of unknown emitters," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8445–8457, 2017.

[116] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proc. of ACM MobiSys*, 2005, pp. 205–218.

[117] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona lisa: Spot localization using phy layer information," in *Proc. of ACM MobiSys*, 2012, pp. 183–196.

[118] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proc. of ACM MobiCom*, 2012, pp. 269–280.

[119] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Proc. of ACM MobiCom*, 2012, pp. 305–316.

[120] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. of ACM MobiSys*, 2012, pp. 197–210.

[121] L. Ni, Y. Liu, Y. Lau, and A. Patil, "Landmarc: Indoor location sensing using active rfid," *Wireless networks*, 2004.

[122] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for wifi-based indoor localization," in *Proc. of ACM AAAI workshop*, vol. 6, 2008.

[123] C. Li, Z. Cao, and Y. Liu, "Deep ai enabled ubiquitous wireless sensing: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–35, 2021.

[124] W. Qian, F. Lauri, and F. Gechter, "Supervised and semi-supervised deep probabilistic models for indoor positioning problems," *Neurocomputing*, vol. 435, pp. 228–238, 2021.

[125] C. Zhan, M. Ghaderibaneh, P. Sahu, and H. Gupta, "Deepmtl: Deep learning based multiple transmitter localization," in *Proc. of IEEE WoWMoM*, 2021, pp. 41–50.

[126] S. M. Nguyen, D. V. Le, and P. J. Havinga, "Learning the world from its words: Anchor-agnostic transformers for fingerprint-based indoor localization," in *Proc. of IEEE PerCom*, 2023, pp. 150–159.

[127] X. Wang, J. Zhang, S. Mao, S. C. Periaswamy, and J. Patton, "Locating multiple rfid tags with swin transformer-based rf hologram tensor filtering," in *Proc. of IEEE VTC*, 2022, pp. 1–2.

[128] OpenAI, "Chatgpt," https://openai.com/chatgpt, 2023.

[129] Meta, "Large language model (llama) at meta ai," https://ai.meta.com/blog/large-language-model-llama-meta-ai/, 2023.

[130] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with csi images," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 316–327, 2018.

[131] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[132] L. K. Goyal, R. Chauhan, R. Kumar, and H. S. Rai, "Use of bim in development of smart cities: A review," in *IOP Conference Series: Materials Science and Engineering*, vol. 955, no. 1. IOP Publishing, 2020, p. 012010.

[133] T. H. Kolbe, G. Gröger, and L. Plümer, "Citygml–3d city models and their potential for emergency response," in *Geospatial information technology for emergency response*. CRC Press, 2008, pp. 273–290.

[134] M. Breunig and S. Zlatanova, "3d geo-database research: Retrospective and future directions," *Computers & Geosciences*, vol. 37, no. 7, pp. 791–803, 2011.

[135] Z. Fang, G. Wang, X. Xie, F. Zhang, and D. Zhang, "Urban map inference by pervasive vehicular sensing systems with complementary mobility," *Proc. of the ACM IMWUT*, vol. 5, no. 1, pp. 1–24, 2021.

[136] A. Cirulis and K. B. Brigmanis, "3d outdoor augmented reality for architecture and urban planning," *Procedia Computer Science*, vol. 25, pp. 71–79, 2013.

[137] D. Suwardhi, S. W. Trisyanti, R. Virtriana, A. A. Syamsu, S. Jannati, and R. S. Halim, "Heritage smart city mapping, planning and land administration (hestya)," *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, p. 107, 2022.

[138] A. T. Irish, J. T. Isaacs, F. Quitin, J. P. Hespanha, and U. Madhow, "Probabilistic 3d mapping based on gnss snr measurements," in *Proc. of IEEE ICASSP*. IEEE, 2014, pp. 2390–2394.

[139] K. Kim, J. Summet, T. Starner, D. Ashbrook, M. Kapade, and I. Essa, "Localization and 3d reconstruction of urban scenes using gps," in *2008 12th IEEE international symposium on wearable computers*. IEEE, 2008, pp. 11–14.

[140] J. G. Rodrigues and A. Aguiar, "Extracting 3d maps from crowdsourced gnss skyview data," in *Proc. of ACM MobiCom*, 2019, pp. 1–15.

[141] Y. Wada, L.-T. Hsu, Y. Gu, and S. Kamijo, "Optimization of 3d building models by gps measurements," *GPS solutions*, vol. 21, pp. 65–78, 2017.

[142] X. Zhao, Z. An, Q. Pan, and L. Yang, "Nerf2: Neural radio-frequency radiance fields," in *Proc. of ACM MobiCom*, 2023, pp. 1–15.

[143] "Sense my city," Available online: https://sensemycity.pt/.

[144] M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-grade global positioning system (gps) accuracy and reliability," *Journal of forestry*, vol. 103, no. 4, pp. 169–173, 2005.

[145] N. Vallina-Rodriguez, J. Crowcroft, A. Finamore, Y. Grunenberger, and K. Papagiannaki, "When assistance becomes dependence: characterizing the costs and inefficiencies of a-gps," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 17, no. 4, pp. 3–14, 2013.

[146] A. Minetto, M. C. Bello, and F. Dovis, "Dgnss cooperative positioning in mobile smart devices: A proof of concept," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3480–3494, 2022.

[147] J. Ventura-Traveset, C. López de Echazarreta, J.-P. Lam, and D. Flament, "An introduction to egnos: The european geostationary navigation overlay system," *GALILEO Positioning Technology*, pp. 323–358, 2015.

[148] J. Paziewski, M. Fortunato, A. Mazzoni, and R. Odolinski, "An analysis of multi-gnss observations tracked by recent android smartphones and smartphone-only relative positioning results," *Measurement*, vol. 175, p. 109162, 2021.

[149] F. Zangenehnejad and Y. Gao, "Gnss smartphones positioning: Advances, challenges, opportunities, and future perspectives," *Satellite navigation*, vol. 2, pp. 1–23, 2021.

[150] A. Elluswamy, "CVPR 2022 Workshop on Autonomous Driving," https://www.youtube.com/watch?v=jPCV4GKX9Dw&ab_channel=WADatCVPR, 2022.

[151] J. L. Sloan., "Gps satellite prn 4," https://coverclock.blogspot.com/2018/11/gps-satellite-prn-4.html, 2018.

[152] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *arXiv preprint arXiv:1906.07751*, 2019.

[153] J. Huang, J. Stoter, R. Peters, and L. Nan, "City3d: Large-scale building reconstruction from airborne lidar point clouds," *Remote Sensing*, vol. 14, no. 9, 2022.

[154] V. Coric and M. Gruteser, "Crowdsensing maps of on-street parking spaces," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*. IEEE, 2013, pp. 115–122.

[155] Q. Jiang, Y. Ma, K. Liu, and Z. Dou, "A probabilistic radio map construction scheme for crowdsourcing-based fingerprinting localization," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3764–3774, 2016.

[156] D. Chen and K. G. Shin, "Turnsmap: enhancing driving safety at intersections with mobile crowdsensing and deep learning," *Proc. of the ACM IMWUT*, vol. 3, no. 3, pp. 1–22, 2019.

[157] C. Cao, Z. Liu, M. Li, W. Wang, and Z. Qin, "Walkway discovery from large scale crowdsensing," in *Proc. of IEEE IPSN*. IEEE, 2018, pp. 13–24.

[158] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Y. Zhao, X. Zhou, and H. Zheng, "Empirical validation of commodity spectrum monitoring," in *Proc. of ACM Sensys*, 2016, pp. 96–108.

[159] J. Shi, Z. Guan, C. Qiao, T. Melodia, D. Koutsonikolas, and G. Challen, "Crowdsourcing access network spectrum allocation using smartphones," in *Proc of ACM workshop on HotNets*, 2014, pp. 1–7.

[160] X. Wu, P. Yang, S. Tang, X. Zheng, and Y. Xiong, "Privacy preserving rss map generation for a crowdsensing network," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 42–48, 2015.

[161] L. Chen, C. Lu, F. Yuan, Z. Jiang, L. Wang, D. Zhang, R. Luo, X. Fan, and C. Wang, "Uvlens: urban village boundary identification and population estimation leveraging open government data," *Proc. of the ACM IMWUT*, vol. 5, no. 2, pp. 1–26, 2021.

[162] Z. Fang, F. Zhang, L. Yin, and D. Zhang, "Multicell: Urban population modeling based on multiple cellphone networks," *Proc. of the ACM IMWUT*, vol. 2, no. 3, pp. 1–25, 2018.

[163] H. Wang, B. Guo, S. Wang, T. He, and D. Zhang, "Csmc: Cellular signal map construction via mobile crowdsensing," *Proc. of the ACM IMWUT*, vol. 5, no. 4, pp. 1–22, 2021.

[164] M. F. R. M. Billah, N. Saoda, J. Gao, and B. Campbell, "Ble can see: a reinforcement learning approach for rf-based indoor occupancy detection," in *Proc. of ACM IPSN*, 2021, pp. 132–147.

[165] A. Kalyanaraman, E. Soltanaghaei, and K. Whitehouse, "Doorpler: A radar-based system for real-time, low power zone occupancy sensing," in *Proc. of IEEE RTAS*. IEEE, 2019, pp. 42–53.

[166] X. Wang, K. Niu, A. Yu, J. Xiong, Z. Yao, J. Wang, W. Li, and D. Zhang, "Wimeasure: Millimeter-level object size measurement with commodity wifi devices," *Proc. of the ACM IMWUT*, vol. 7, no. 2, pp. 1–26, 2023.

[167] D. Wu, D. Zhang, C. Xu, Y. Wang, and H. Wang, "Widir: Walking direction estimation using wireless signals," in *Proc. of ACM UbiComp*, 2016, pp. 351–362.

[168] C. Wu, X. Huang, J. Huang, and G. Xing, "Enabling ubiquitous wifi sensing with beamforming reports," in *Proc. of ACM SIGCOMM*, 2023, pp. 20–32.

[169] P. Corbalán, G. P. Picco, and S. Palipana, "Chorus: Uwb concurrent transmissions for gps-like passive localization of countless targets," in *Proceedings of the*

18th International Conference on Information Processing in Sensor Networks, 2019, pp. 133–144.

[170] J. Guan, S. Madani, S. Jog, S. Gupta, and H. Hassanieh, "Through fog high-resolution imaging using millimeter wave radar," in *In Proc. of IEEE/CVF CVPR*, June 2020.

[171] M. Bevis, S. Businger, T. A. Herring, C. Rocken, R. A. Anthes, and R. H. Ware, "Gps meteorology: Remote sensing of atmospheric water vapor using the global positioning system," *Journal of Geophysical Research: Atmospheres*, vol. 97, no. D14, pp. 15 787–15 801, 1992.

[172] O. Esrafilian, R. Gangula, and D. Gesbert, "Three-dimensional-map-based trajectory design in uav-aided wireless localization systems," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9894–9904, 2020.

[173] Y. Corre and Y. Lostanlen, "Three-dimensional urban em wave propagation model for radio network planning and optimization over large areas," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3112–3123, 2009.

[174] H. Masaharu and H. Hasegawa, "Three-dimensional city modeling from laser scanner data by extracting building polygons using region segmentation method," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, no. B3/1; PART 3, pp. 556–562, 2000.

[175] F. Alidoost, H. Arefi, and F. Tombari, "2d image-to-3d model: Knowledge-based 3d building reconstruction (3dbr) using single aerial images and convolutional neural networks (cnns)," *Remote Sensing*, vol. 11, no. 19, p. 2219, 2019.

[176] S. Banville and F. van Diggelen, "Precision gnss for everyone," *GPS World*, vol. 27, no. 11, pp. 43–48, 2016.

[177] P. D. Groves, "Shadow matching: A new gnss positioning technique for urban canyons," *The journal of Navigation*, vol. 64, no. 3, pp. 417–430, 2011.

[178] R. Swinford, "Building on-the-fly world models for pervasive gaming and other ubicomp applications using gps availability data," in *The IEEE International Workshop on Intelligent Environments*. IET, 2005, pp. 133–142.

[179] J. G. Rodrigues, A. Aguiar, and C. Queirós, "Opportunistic mobile crowdsensing for gathering mobility information: Lessons learned," in *Proc. of IEEE ITSC*. IEEE, 2016, pp. 1654–1660.