RESILIENT UAV POSITIONING SYSTEM FOR GNSS-DENIED ENVIRONMENTS

by

Kazuya Hirotsu

A Master Thesis

Submitted to
the Graduate School of the University of Tokyo
on February XX, 2025
in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in the Field of
Systems Innovation

Thesis Supervisor: Nakao Akihiro
Professor of Engineering

SUMMARY

1. Introduction

Unmanned Aerial Vehicles (UAVs) are now integral to sectors such as agriculture, infrastructure inspection, and disaster response. However, the operational reliability of these autonomous systems remains heavily dependent on Global Navigation Satellite Systems (GNSS) like GPS. This dependence creates a critical vulnerability: GNSS signals suffer obstruction in urban canyons and mountainous terrain, and face growing threats from intentional jamming and spoofing. Alternative navigation technologies, including Visual SLAM (Simultaneous Localization and Mapping) and LiDAR, impose heavy computational loads, require expensive hardware, and break down in feature-poor environments such as deserts, snow fields, or open water. A resilient, low-cost, and rapidly deployable backup positioning system that operates independently of GNSS remains an open challenge.

Low-Power Wide-Area (LPWA) networks, specifically LoRa (Long Range), offer kilometer-range communication, low power consumption, and minimal hardware cost, making them attractive as positioning infrastructure. In such a system, ground-based beacons broadcast their known coordinates, and a receiver estimates its position by measuring the Received Signal Strength Indicator (RSSI) from multiple beacons and applying multilateration.

Problem Statement: Repurposing LoRa for localization raises two challenges: (1) On the client side, RSSI-based localization suffers from high variance due to multipath fading and environmental shadowing, directly degrading positioning accuracy. (2) On the infrastructure side, beacon positions must be known before they can serve as reference points, yet in GNSS-denied zones, conventional surveying or GNSS-based calibration may be unavailable.

This thesis investigates the feasibility of using LoRa networks as a standalone positioning infrastructure for GNSS-denied environments, addressing both the client-side localization challenge and the infrastructure initialization problem.

We propose a two-layer architecture that addresses both challenges: (1) Robust Client-Side Localization, which enables UAV positioning given known beacon locations, and (2) Infrastructure Self-Organization, which enables the beacon network to localize itself in unknown environments. Together, these layers form a survivable positioning system that supports UAV operations in GNSS-denied regions with operationally useful accuracy.

Our contributions are threefold:

1. We introduce a robust multilateration algorithm that combines altitude constraints with the Huber loss function, suppressing the influence of RSSI outliers on UAV position estimates.
2. We develop a Physics-Aware Graph Convolutional Network (GCN) that enables sparse beacon networks to self-localize in complex terrain by learning a terrain-aware path-loss model jointly with the graph structure.
3. We present a feasibility analysis of the integrated system, showing that despite RSSI-induced localization errors on the order of 150 to 200 meters, point-to-point navigation and wide-area search missions remain achievable through geometric stability and strategic flight path planning.

Our work differs from prior research by unifying infrastructure self-organization and robust client positioning into a single resilient system for GNSS-denied environments. Existing approaches treat these problems separately: UAV localization studies typically assume a pre-surveyed, high-precision anchor network, while sensor localization methods rely on high beacon densities or labor-intensive fingerprinting. We instead propose an integrated architecture that builds positioning capability from scratch. At the infrastructure layer, our physics-aware graph learning model embeds radio propagation physics, unlike standard black-box neural networks, enabling sparse beacon networks to self-localize in complex terrain without GNSS. At the client layer, we avoid hardware-intensive timing or angle-based techniques and instead develop a robust multilateration

algorithm that exploits altitude constraints and the Huber loss to handle the volatility of low-power signals. This integrated approach removes the dependence on external positioning at both layers, yielding a standalone solution for unknown environments.

This thesis demonstrates that while LoRa-based positioning cannot match GNSS precision, it provides a viable backup system for emergency operations, enabling autonomous missions to continue when primary navigation fails.

2. Related Work

Wireless Positioning Fundamentals: Time-Difference-of-Arrival (TDoA) and Angle-of-Arrival (AoA) methods yield higher accuracy than Received Signal Strength Indicator (RSSI), but impose strict hardware constraints. TDoA demands nanosecond-level time synchronization, typically derived from GPS, which defeats the purpose of a GNSS-independent backup. AoA requires complex antenna arrays. We therefore focus on RSSI-based methods, which work with standard, low-cost LoRa modules.

Client-Side: UAV Localization Approaches: Conventional UAV navigation relies on GNSS or visual odometry (SLAM). SLAM performs well in structured environments but is computationally heavy and degrades in feature-poor settings such as deserts or open water. In the RF domain, prior work targets high-density indoor scenarios (WiFi or Bluetooth grids) or depends on hardware-intensive TDoA/AoA systems. These solutions do not transfer to sparse, large-scale outdoor deployments where hardware simplicity is critical. Our approach achieves robustness through algorithmic improvements, specifically altitude constraints and Huber loss, rather than specialized hardware or high beacon density, and operates with simple, single-antenna beacons.

Infrastructure-Side: Network Self-Localization: For self-localizing the beacon network, standard multilateration is insufficient as it cannot localize nodes lacking direct connection to at least three anchors. Iterative Multilateration extends this by promoting newly localized nodes to anchor status, but it suffers from severe error propagation in sparse, non-line-of-sight (NLOS) conditions. Fingerprinting methods achieve high accuracy but require labor-intensive site surveys. Graph Neural Networks (GNNs) have been applied to sensor localization, yet prior studies typically treat the network as a black box or focus on dense indoor environments. Our work applies a *Physics-Aware* GCN designed for sparse, outdoor LPWA networks, embedding path-loss physics directly into the learning process.

3. System Architecture

To clarify the operational concept, we define the system architecture into two distinct components: the **Infrastructure** (Beacons) and the **Client** (UAV).

- **Infrastructure (Beacons):** The ground segment consists of low-cost LoRa beacons. In scenarios where beacon positions are unknown (e.g., aerial deployment), the system initiates a **Self-Organization Process**. This process uses a Physics-Aware GCN model, pre-trained on synthetic data derived from digital elevation models to capture site-specific propagation characteristics. Beacons measure peer-to-peer RSSI, and this data is aggregated at a central hub where the model estimates their positions. The hub then transmits the estimated coordinates back to each beacon. Once localized (or if positions were already known), the beacons enter a power-saving state, activating **Broadcast Mode** only upon receiving a wake-up signal.
- **Client (UAV):** The UAV primarily operates in a passive, receive-only mode. It listens to the global broadcasts from the localized beacons. Using the received RSSI and the broadcasted coordinates, the UAV performs onboard multilateration to estimate its own position. To conserve infrastructure power, the UAV transmits a wake-up signal only when broadcast signals are unavailable, maintaining system scalability.

4. Robust UAV Localization

This section addresses the client-side localization problem: estimating the UAV's position given beacons with known locations. The core technical challenge is the high variance in RSSI data caused by multipath propagation and shadowing.

To address this, we improve the standard multilateration algorithm with two key modifications:

1. **Altitude Decoupling:** Modern UAVs carry reliable barometers or laser altimeters. We exploit this by decoupling vertical (Z-axis) estimation from horizontal (X/Y) positioning, reducing the optimization from three dimensions to two and preventing vertical errors from corrupting the horizontal solution.
2. **Huber Loss:** We replace the standard Least Squares (L2) objective with the Huber loss. Least Squares penalizes large errors quadratically and is therefore sensitive to outliers; Huber loss applies a linear penalty beyond a threshold, preventing occasional extreme RSSI values from distorting the position estimate.

We validate this approach through simulation and field experiments:

- **Simulation:** In static hover tests, the proposed method achieves a median error of **4.5 m**, compared to 29.7 m for the standard method. In dynamic flight at 10 m/s, the median error rises to **62.9 m**, still a **25%** reduction relative to the standard approach (84.1 m).
- **Field Experiments:** Tests in Italy using a commercial UAV and four low-cost LoRa beacons confirm the simulation trends, though improvements are more modest with fewer beacons. The experiments also reveal a hardware limitation: at higher altitudes, the antenna radiation pattern attenuates signals directly above the beacon (a "cone of silence"), indicating the need for altitude-aware path-loss models or alternative antenna configurations.

5. Infrastructure Self-Localization

This section addresses the infrastructure-side problem: self-localization of the beacon network in a GNSS-denied environment where manual surveying is impractical. The deployment scenario assumes a subset of nodes serve as anchors with known positions (e.g., established at the perimeter of the denied zone or obtained during brief windows of GNSS availability), while the remaining beacons must determine their coordinates from peer-to-peer RSSI measurements alone.

To solve this ill-posed inverse problem, we employ a **Graph Convolutional Network (GCN)** pre-trained on synthetic terrain data. Unlike standard data-driven models that treat the network as a black box, our architecture is physics-aware: it learns the parameters of a log-distance path-loss model (n and o) jointly with the graph structure, allowing the model to adapt to local radio propagation characteristics.

We evaluate the method on a terrain-aware simulation of a 4 km $\times$ 4 km mountainous region using the Longley-Rice propagation model, with 25% of nodes designated as anchors (16 out of 64).

- **Results:** At a beacon density of 4 per km^2, iterative multilateration fails with a mean error exceeding **1974 m**. A standard GCN reduces this to **750 m**, but still performs poorly. The proposed Physics-Aware GCN achieves a mean error of **206 m**, a **90% improvement** over the baselines.
- **Implication:** Although this accuracy marks an advance for RSSI-only self-localization in complex terrain, the residual error of roughly 200 m raises a practical question: is this accuracy sufficient for UAV operations?

6. Integrated Feasibility and Deployment Strategy

This section combines the infrastructure self-localization and robust UAV localization results. We analyze the full deployment lifecycle, from air-dropping beacons to executing a mission, and simulate a UAV navigating with beacon positions estimated by the GCN.

Sensitivity Analysis: We first quantify how infrastructure errors propagate to the UAV through grid-wide localization analysis at 2,000 sampled positions. Adding beacon position error of 200 m mean 2D error (matching the GCN's self-localization performance of 206 m) increases the UAV's localization error from a baseline of 122 m to 157 m, a degradation of only 29%. This sub-linear degradation confirms system stability: errors in the infrastructure do not cause catastrophic failure on the client side.

Navigation Feasibility: We then run closed-loop flight simulations across three mission profiles: diagonal transit (4 km point-to-point), circular patrol (1 km radius), and area sweep (systematic coverage pattern). This analysis provides three insights into operational feasibility:

1. **Geometric Stability:** When the UAV is more than 1 km from its target, a 200 m position error translates to a worst-case heading error of only **11 degrees**. This geometric dilution keeps the flight path stable and nearly straight toward the target, despite high positional uncertainty.
2. **Turning Point Problem:** As the UAV closes to within 200 m of a waypoint, the ratio of distance to error shrinks, causing heading error to spike and inducing erratic behavior. We mitigate this by setting a **200 m arrival threshold**, favoring mission continuity over pinpoint waypoint accuracy.
3. **Mission Success:** All three mission profiles achieve 100% waypoint success at beacon error levels up to 200 m. For area coverage missions, planning flight paths with overlapping tracks compensates for navigation uncertainty.

7. Conclusion

This thesis shows that LoRa technology, while unsuitable for precision navigation, can serve as an effective backup for emergency positioning. By coupling a self-organizing ground network with robust UAV localization algorithms, we build a functional positioning infrastructure in environments with limited prior infrastructure. The resulting system supports disaster-response and remote sensing operations, allowing autonomous missions to continue when primary navigation fails.

Acknowledgements

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Unmanned Aerial Vehicles (UAVs) have transformed numerous sectors, from commercial delivery and infrastructure inspection to environmental monitoring and disaster response [**?**, **?**]. The global UAV market continues to expand as autonomous flight capabilities enable applications that were previously impractical or impossible. At the core of these autonomous operations lies accurate positioning, which is the ability to determine the vehicle's location in real-time with sufficient precision to navigate safely and complete assigned missions.

For the past several decades, Global Navigation Satellite Systems (GNSS), exemplified by the U.S. Global Positioning System (GPS), have served as the dominant positioning technology for outdoor applications [**?**]. GNSS provides meter-level accuracy with global coverage, low receiver cost, and minimal infrastructure requirements. These characteristics have made GNSS the de facto standard for UAV navigation, with virtually all commercial UAVs relying on satellite signals for position estimation.

However, GNSS dependence introduces a critical single point of failure into UAV operations. Three categories of vulnerabilities threaten GNSS reliability:

1. **Environmental Obstruction:** In urban canyons surrounded by tall buildings, dense forests, mountainous terrain, and indoor or semi-indoor environments such as warehouses and stadiums, GNSS signals become weak, intermittent, or entirely unavailable due to physical obstruction and multipath propagation [**?**].

2. **Intentional Interference:** GNSS jamming and spoofing have emerged as significant threats. Jamming devices that overwhelm receivers with noise are inexpensive and widely available, while sophisticated spoofing attacks can feed false position data to receivers [**?**]. Reports indicate increasing incidents of deliberate GNSS interference in various regions worldwide.

3. **Extraterrestrial Environments:** Future UAV and robotic operations on the Moon, Mars, or other celestial bodies cannot rely on Earth-based GNSS infrastructure, necessitating alternative positioning approaches.

Alternative navigation technologies exist but impose substantial constraints. Visual Simultaneous Localization and Mapping (SLAM) uses camera imagery to build environmental maps while tracking position, achieving high accuracy in structured environments [**?**]. However, SLAM requires significant computational resources, struggles in feature-poor environments such as deserts, snow fields, and open water, and fails entirely in low-visibility conditions including fog,

smoke, and darkness. Light Detection and Ranging (LiDAR) systems offer robust performance across lighting conditions but add considerable weight, power consumption, and cost to the platform. Inertial Navigation Systems (INS) provide short-term position estimates through accelerometer and gyroscope integration but accumulate drift errors rapidly, limiting their standalone utility to seconds or minutes.

These limitations motivate research into radio-frequency (RF) positioning systems that can complement or replace GNSS when satellite signals are unavailable. Among RF technologies, Low-Power Wide-Area Network (LPWAN) protocols have attracted attention for their combination of long communication range, low power consumption, and minimal hardware cost [**?**]. LoRa (Long Range), a leading LPWAN technology, achieves communication distances of 2–15 km in line-of-sight conditions while consuming milliwatts of power and requiring only low-cost transceiver modules [**?**]. These characteristics make LoRa attractive as the foundation for a backup positioning infrastructure that can be rapidly deployed in GNSS-denied regions.

1.2 Problem Statement

Repurposing LoRa networks for localization introduces two fundamental challenges that this thesis addresses:

Challenge 1: Client-Side Localization Accuracy The most accessible approach to RF-based localization uses Received Signal Strength Indicator (RSSI) measurements. When a receiver detects signals from multiple beacons at known positions, it can estimate distances based on signal attenuation and compute its position through multilateration. However, RSSI exhibits high variance due to multipath fading, environmental shadowing, and antenna orientation effects [**?**]. In practical deployments, the relationship between RSSI and distance becomes noisy and non-linear, causing standard multilateration algorithms to produce large position errors.

Alternative RF metrics offer higher accuracy but impose prohibitive constraints for low-cost backup systems. Time-of-Flight (ToF) and Time-Difference-of-Arrival (TDoA) methods measure signal propagation time to estimate distance, achieving meter-level accuracy [**?**]. However, these techniques require nanosecond-level clock synchronization across all beacons, which is typically achieved using GNSS timing signals; this defeats the purpose of a GNSS-independent backup. Angle-of-Arrival (AoA) methods determine bearing to signal sources but require complex antenna arrays at each beacon [**?**].

The client-side challenge is therefore: *How can we achieve robust UAV localization using only RSSI measurements from commodity LoRa hardware, without requiring clock synchronization or specialized antennas?*

Challenge 2: Infrastructure Self-Localization Multilateration requires beacon positions to be known precisely before they can serve as reference points. In conventional deployments, beacon coordinates are obtained through GNSS measurements during installation or professional surveying. However, in GNSS-denied environments, the very scenarios motivating this research, these calibration methods become unavailable.

Consider an emergency response scenario where a beacon network must be established rapidly in a disaster zone or remote area. Beacons might be air-

dropped from aircraft or deployed by ground teams without access to surveying equipment. In such cases, the beacons themselves must determine their positions through mutual communication before they can provide localization services to UAV clients.

This infrastructure self-localization problem is fundamentally ill-posed when using RSSI alone. A small subset of nodes may have known positions (anchors), perhaps established at the perimeter of the denied zone or recorded during brief windows of GNSS availability, but the majority must estimate their coordinates from peer-to-peer RSSI measurements. Classical geometric solvers such as iterative multilateration suffer from severe error propagation in sparse networks with complex terrain, where early estimation errors compound catastrophically through subsequent iterations [**?**].

The infrastructure-side challenge is therefore: *How can a sparse beacon network self-localize in complex outdoor terrain using only RSSI measurements, given that only a minority of nodes have known positions?*

1.3 Contributions

In this thesis, we demonstrate that a carefully designed system architecture, combined with algorithmic innovations at both the client and infrastructure layers, can provide operationally useful positioning in GNSS-denied environments despite the inherent limitations of RSSI-based localization.

Our contributions are threefold:

1. **Robust Client-Side Localization Algorithm:** We introduce a multilateration algorithm that combines altitude constraints with the Huber loss function to suppress the influence of RSSI outliers on UAV position estimates. By exploiting reliable altitude data from barometers or laser altimeters, we decouple vertical estimation from horizontal positioning, reducing the optimization from three dimensions to two. The Huber loss function provides robustness against the large RSSI variance inherent in outdoor LoRa deployments. Through comprehensive simulations and field experiments, we demonstrate median localization errors of 4.5 m in static conditions and 62.9 m during dynamic flight at 10 m/s, representing a 25% improvement over conventional least-squares multilateration.

2. **Physics-Aware Graph Convolutional Network for Infrastructure Self-Localization:** We develop a Graph Convolutional Network (GCN) architecture that embeds radio propagation physics directly into the learning process. Unlike black-box neural networks, our model jointly learns the parameters of a log-distance path-loss model alongside the graph message-passing weights. This physics-aware design enables the network to adapt to local propagation characteristics while maintaining interpretability. In terrain-aware simulations of a 4 km $\times$ 4 km mountainous region, our method achieves a mean localization error of 206 m, representing a 90% improvement over iterative multilateration (1974 m) and 72% improvement over a standard GCN without the physics-informed module (750 m).

3. **Integrated System Feasibility Analysis:** We present a comprehensive analysis of the complete system, from infrastructure self-organization through UAV mission execution. Through closed-loop navigation simulations, we demonstrate that despite localization errors on the order of 150–200 m, the system maintains operational utility through geometric stability:

at 1 km from target, even 200 m position error produces at most 11 degrees of heading error. We identify the "turning point problem" where accuracy degrades near waypoints and propose mitigation strategies including relaxed arrival thresholds and overlapping flight paths for area coverage missions.

Our work differs from prior research by unifying infrastructure self-organization and robust client positioning into a single resilient system for GNSS-denied environments. Existing approaches treat these problems separately: UAV localization studies typically assume a pre-surveyed, high-precision anchor network, while sensor network localization methods rely on high beacon densities or labor-intensive fingerprinting. We instead propose an integrated architecture that builds positioning capability from scratch using only commodity LoRa hardware and algorithmic innovations.

Scope and Evaluation Methodology The end-to-end integrated system (infrastructure self-localization combined with UAV navigation) is evaluated through simulation using terrain-aware propagation models. Field experiments with real UAV hardware validate only the client-side localization algorithm (Chapter **??**), conducted under GNSS-available conditions to provide ground truth for accuracy assessment. Bridging the simulation-to-reality gap for the complete system remains an important direction for future work.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter ??: Related Work reviews the landscape of wireless positioning technologies, comparing RSSI, TDoA, and AoA methods. We examine prior work on UAV localization and network self-localization, positioning our contributions relative to existing approaches.

Chapter ??: System Architecture presents the overall design of the proposed positioning system, defining the roles and interactions between infrastructure beacons, client UAVs, and the central processing hub.

Chapter ??: Robust UAV Localization addresses the client-side localization problem. We detail the proposed multilateration algorithm with altitude constraints and Huber loss, present simulation results using OMNeT++, and validate the approach through field experiments in Italy.

Chapter ??: Infrastructure Self-Localization addresses the infrastructure-side problem. We formulate the beacon self-localization task as a graph learning problem, present the physics-aware GCN architecture, and evaluate performance against baselines using terrain-aware Longley-Rice simulations.

Chapter ??: Integrated System Analysis combines results from the previous chapters to assess overall system feasibility. We analyze sensitivity to infrastructure errors, demonstrate geometric stability through closed-loop simulations, and propose deployment strategies for practical applications.

Chapter ??: Conclusion summarizes contributions, discusses limitations, and outlines directions for future work.

Chapter 2

Related Work

This chapter reviews the landscape of wireless positioning technologies and prior work relevant to the two core problems addressed in this thesis: client-side UAV localization and infrastructure-side network self-localization. We examine the trade-offs among different signal metrics, survey existing approaches in each problem domain, and position our contributions relative to the state of the art.

2.1 Wireless Positioning Fundamentals

Wireless positioning systems estimate location by analyzing signals exchanged between a target device and reference nodes at known positions. The fundamental approaches differ in which signal characteristic they measure and how they convert measurements to position estimates.

2.1.1 Signal Metrics for Distance Estimation

Three primary signal metrics are used for wireless localization, each offering different trade-offs between accuracy, hardware complexity, and deployment constraints.

Received Signal Strength Indicator (RSSI) RSSI measures the power level of a received signal in dBm, which attenuates with distance according to the log-distance path-loss model [**?**]:

$$P_r = P_t + o - 10n \log_{10}(d) + X_\sigma \tag{2.1}$$

where:

- P_r is the received power (RSSI) in dBm
- P_t is the transmitted power in dBm
- o is an offset capturing antenna gains and reference distance effects (dB)
- n is the path-loss exponent (typically 2 in free space, 2.5–4 in outdoor environments)
- d is the distance in meters
- X_σ is a zero-mean Gaussian random variable representing shadow fading (typically 4–12 dB standard deviation)

To estimate distance from an RSSI measurement, we invert Equation **??**, setting aside the stochastic term:

$$\hat{d} = 10^{(P_t + o - P_r)/(10n)} \tag{2.2}$$

The parameters n and o must be determined through calibration in the deployment environment or learned from data.

RSSI is the most widely available metric on commodity radios, requiring no specialized hardware beyond a standard transceiver. However, RSSI exhibits high variance due to multipath propagation, shadowing from obstacles, antenna orientation, and environmental factors [**?**]. In outdoor environments with complex terrain, the noise term X_σ can be large and non-Gaussian, causing a single RSSI value to correspond to multiple possible distances.

Time-Based Methods (ToF/TDoA) Time-of-Flight (ToF) measures the propagation delay of a signal traveling from transmitter to receiver, converting time to distance using the speed of light. Time-Difference-of-Arrival (TDoA) uses the difference in arrival times at multiple receivers to hyperbolic position fixing, eliminating the need for transmitter-receiver synchronization [**?**].

Time-based methods offer superior accuracy compared to RSSI because electromagnetic waves propagate at a constant speed regardless of environmental conditions. Fargas and Petersen [**?**] demonstrate GPS-free geolocation using LoRa TDoA with promising results. However, achieving the nanosecond-level time synchronization required for meter-level accuracy typically relies on GNSS timing signals, which defeats the purpose of a GNSS-independent backup system. Alternative synchronization methods using atomic clocks or network protocols add significant cost and complexity.

Angle-of-Arrival (AoA) AoA methods determine the direction from which a signal arrives using antenna arrays, enabling triangulation-based positioning. Baik et al. [**?**] demonstrate a hybrid RSSI-AoA system achieving approximately 3 m accuracy in a 55 m $\times$ 101 m field. However, AoA requires phased antenna arrays or time-modulated array receivers at each reference node, significantly increasing hardware cost and complexity compared to single-antenna RSSI systems.

Channel State Information (CSI) CSI provides fine-grained physical-layer information about the wireless channel, capturing amplitude and phase across multiple subcarriers. Fan and Yan [**?**] demonstrate CSI-based indoor localization using graph structure fusion. While CSI enables high accuracy in controlled environments, the specialized hardware required to extract CSI is not commonly available on low-cost LPWA devices designed for outdoor deployment.

2.1.2 Why LoRa and RSSI

Given the constraints of GNSS-denied environments and the goal of rapid, low-cost deployment, this thesis focuses on RSSI-based localization using LoRa technology. This choice reflects several practical considerations:

- **Hardware Availability:** RSSI is a standard output of all LoRa transceivers, requiring no modifications or additional equipment. Commodity LoRa modules cost under $10 and are widely available.

- **GNSS Independence:** Unlike TDoA, RSSI-based methods do not require clock synchronization, enabling truly GNSS-independent operation.
- **Communication Range:** LoRa achieves 2–15 km range in line-of-sight conditions, significantly reducing the number of beacons required compared to WiFi (tens of meters) or Bluetooth (meters) [**?**].
- **Power Efficiency:** LoRa's low power consumption enables battery-powered or solar-powered beacons suitable for remote deployment.

The challenge, addressed in subsequent chapters, is to achieve useful positioning accuracy despite RSSI's inherent variance through algorithmic improvements rather than hardware enhancements.

2.2 UAV Localization Approaches

This section reviews methods for localizing a mobile UAV relative to reference infrastructure, corresponding to the client-side problem addressed in Chapter **??**.

2.2.1 Non-RF Methods

Visual SLAM Simultaneous Localization and Mapping (SLAM) uses camera imagery to construct an environmental map while simultaneously tracking the camera's position within that map [**?**]. Visual SLAM achieves centimeter-level accuracy in structured indoor environments with distinctive visual features. However, SLAM requires substantial computational resources, making it challenging for resource-constrained UAV platforms. More critically, SLAM degrades significantly in feature-poor environments such as deserts, snow fields, and open water, and fails entirely in low-visibility conditions including fog, smoke, and darkness.

LiDAR-Based Navigation Light Detection and Ranging (LiDAR) systems emit laser pulses and measure return times to build 3D environmental models. LiDAR is robust to lighting conditions and provides accurate range measurements. However, LiDAR sensors add considerable weight, power consumption, and cost to UAV platforms, making them impractical for small, low-cost vehicles.

Inertial Navigation Inertial Navigation Systems (INS) integrate accelerometer and gyroscope measurements to estimate position changes from a known starting point. While INS provides continuous position updates at high rates, integration errors accumulate rapidly, causing position drift that renders standalone INS useful only for short durations (seconds to minutes) without external corrections.

2.2.2 RF-Based UAV Localization

TDoA-Based Systems Several research efforts have applied TDoA to LoRa-based UAV localization. Fargas and Petersen [**?**] present a GPS-free geolocation system using LoRa TDoA, demonstrating the potential of time-based methods. Chen et al. [**?**] investigate multi-scene LoRa positioning using Kalman filtering with TDoA inputs. While these systems achieve higher accuracy than RSSI-based approaches, they typically require GPS-derived timing for beacon synchronization, creating a dependency that contradicts the goal of GNSS-independent backup.

RSSI-Based Systems Vazquez-Rodas et al. [**?**] conduct experimental evaluation of RSSI-based positioning with low-cost LoRa devices, characterizing the relationship between RSSI variance and positioning accuracy. Müller et al. [**?**] examine outdoor ranging and positioning based on LoRa modulation, demonstrating feasibility but noting accuracy limitations. Ingabire et al. [**?**] apply random neural networks to LoRa RSSI-based outdoor localization in urban areas. Yucer et al. report RSSI-based outdoor localization with a single UAV, demonstrating the concept but with limited accuracy.

Hybrid Approaches Baik et al. [**?**] combine RSSI and AoA measurements using a time-modulated array receiver, achieving approximately 3 m accuracy. However, this approach requires custom dual-antenna configurations and phase modulators at each beacon, significantly increasing deployment complexity and cost.

2.2.3 Positioning of This Work

Existing RF-based UAV localization methods either achieve high accuracy through hardware-intensive TDoA or AoA techniques, or accept the limitations of standard RSSI multilateration. Our approach differs by achieving robustness through algorithmic improvements alone:

- **Altitude Constraints:** We exploit reliable altitude data from barometers or laser altimeters to reduce the optimization problem from 3D to 2D, preventing vertical errors from corrupting horizontal estimates.
- **Huber Loss:** We replace the standard least-squares objective with the Huber loss function, which applies quadratic penalty for small errors and linear penalty for large errors, suppressing the influence of RSSI outliers.

This algorithmic approach requires only commodity single-antenna LoRa modules, enabling practical deployment without the cost and complexity of timing synchronization or antenna arrays.

2.3 Network Self-Localization

This section reviews methods for localizing nodes within a wireless network given limited anchor information, corresponding to the infrastructure-side problem addressed in Chapter **??**.

2.3.1 Classical Geometric Methods

Multilateration Standard multilateration estimates a node's position by finding the point that best satisfies distance constraints to multiple anchors with known positions. Given distance estimates $\hat{d}_i$ to anchors at positions $\mathbf{p}_i$, the target position $\mathbf{x}$ is found by minimizing:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_i \left(\|\mathbf{x} - \mathbf{p}_i\| - \hat{d}_i \right)^2 \tag{2.3}$$

Multilateration requires at least three anchors (in 2D) with known positions and accurate distance estimates. When RSSI is used for distance estimation, the inherent noise leads to inconsistent constraint systems and degraded accuracy.

Iterative Multilateration Iterative multilateration extends the basic approach by promoting newly localized nodes to anchor status, enabling localization of nodes that lack direct connectivity to the original anchors [**?**]. The algorithm proceeds in rounds: nodes with sufficient anchor neighbors are localized, then temporarily promoted to anchors for subsequent rounds.

However, as noted by Hada and Srivastava [**?**], iterative multilateration suffers from severe error propagation in sparse networks with complex terrain. Early estimation errors in promoted nodes compound through subsequent iterations, potentially causing catastrophic failure where final position estimates diverge from true locations by kilometers. This vulnerability makes iterative multilateration unsuitable for the sparse, terrain-rich outdoor networks considered in this thesis.

2.3.2 Data-Driven Methods

Fingerprinting Fingerprinting methods learn a direct mapping from observed signal patterns to coordinates using a pre-collected database of location-tagged measurements [**?**]. Purohit et al. [**?**] demonstrate fingerprinting-based indoor and outdoor localization with LoRa and deep learning, achieving high accuracy in environments where training data is available.

However, fingerprinting requires extensive, laborious site surveys to build the signal database, and the learned mapping is sensitive to environmental changes. These characteristics limit fingerprinting's applicability to scenarios requiring rapid deployment in unknown environments, such as disaster response or remote monitoring.

Machine Learning for RSSI-Distance Mapping Various machine learning approaches have been applied to improve RSSI-to-distance conversion, including neural networks and random forests. These methods can capture non-linear relationships that simple path-loss models miss. However, they typically require labeled training data (RSSI measurements with known distances), which may be unavailable in new deployment environments.

2.3.3 Graph Neural Networks for Localization

Graph Neural Networks (GNNs) have emerged as a powerful framework for localization by explicitly modeling the wireless network as a graph, where nodes represent devices and edges represent communication links [**?**].

Indoor Localization Kang et al. [**?**] propose an indoor localization algorithm based on high-order GNNs, capturing multi-hop connectivity patterns. Fan and Yan [**?**] combine CSI fingerprinting with GCN-based graph structure fusion for indoor localization. Vishwakarma et al. [**?**] develop IndoorGNN, a graph neural network approach for WiFi RSSI-based indoor positioning.

Large-Scale Network Localization Yan et al. [**?**] apply GNNs to large-scale network localization, demonstrating the ability to leverage unknown-to-unknown node connectivity, a source of information that geometric solvers cannot exploit.

Edge-Conditioned Convolutions Simonovsky and Komodakis [**?**] introduce dynamic edge-conditioned filters for graph convolutions, where edge attributes shape the message-passing weights. This architecture enables the network to

learn distance-aware aggregation functions when edge features encode proximity information.

2.3.4 Positioning of This Work

Existing GNN-based localization methods have primarily focused on indoor environments with high node density, or treat the network as a black box without incorporating physical propagation models. Our approach differs in several key aspects:

- **Physics-Aware Design:** We integrate a trainable log-distance path-loss model directly into the GCN architecture. The path-loss parameters are learned jointly with the graph network weights, providing an interpretable physical prior while adapting to local propagation characteristics.
- **Outdoor LPWA Focus:** We specifically target sparse outdoor networks using LoRa, where node density is low (4 nodes/km^2) and terrain effects dominate signal propagation.
- **Terrain-Aware Evaluation:** We evaluate using a Longley-Rice propagation model that captures terrain effects including diffraction and shadowing, rather than idealized free-space assumptions that can produce misleading performance rankings.

2.4 Summary

Table **??** summarizes the positioning of this work relative to prior approaches.

Table 2.1: Comparison of localization approaches

Method	**Hardware**	**GNSS-Free**	**Outdoor**	**Sparse**
TDoA [**?**]	Synced clocks	No*	Yes	Yes
AoA [**?**]	Antenna array	Yes	Yes	Yes
Fingerprinting [**?**]	Standard	Yes	Limited	No
Indoor GNN [**?**]	Standard	Yes	No	No
Iter. Multilat. [**?**]	Standard	Yes	Yes	No
This work (Ch.4)	Standard	Yes	Yes	Yes
This work (Ch.5)	Standard	Yes	Yes	Yes

*Typically requires GNSS for clock synchronization

The key gap addressed by this thesis is the lack of methods that simultaneously: (1) work with commodity hardware, (2) operate independently of GNSS, (3) are designed for outdoor environments with complex terrain, and (4) function in sparse network deployments. Our contributions at both the client layer (Chapter **??**) and infrastructure layer (Chapter **??**) target this gap through algorithmic innovations rather than hardware enhancements.

Chapter 3

System Architecture

This chapter presents the overall architecture of the proposed positioning system for GNSS-denied environments. The design separates the system into three functional segments: the infrastructure segment consisting of fixed beacons, the client segment consisting of mobile UAVs, and the hub segment responsible for centralized computation. This separation enables scalability, privacy preservation, and efficient resource utilization.

3.1 Design Goals and Principles

The system architecture is driven by several key design goals that reflect the operational requirements of GNSS-denied scenarios:

Rapid Deployment The system must be deployable within hours rather than days, enabling effective response to emergent situations such as natural disasters or remote expeditions. This requirement precludes approaches that depend on extensive site surveys, pre-existing infrastructure, or complex calibration procedures.

GNSS Independence The system must operate without any reliance on GNSS signals, either directly for positioning or indirectly for time synchronization. This independence ensures that the system functions as a true backup when satellite signals are unavailable due to jamming, spoofing, or environmental obstruction.

Scalability The architecture must support an arbitrary number of client UAVs without requiring proportional increases in communication bandwidth or infrastructure complexity. As UAV swarm operations become more common, the system should accommodate dozens or hundreds of simultaneous users.

Privacy Preservation Client UAVs should be able to determine their positions without revealing their locations to external observers or even to the infrastructure itself. This property is essential for privacy-sensitive applications where UAV positions may be confidential.

Cost Effectiveness The system should utilize commodity hardware, specifically single-antenna LoRa transceivers, to minimize deployment costs and enable rapid procurement. Algorithmic improvements should compensate for hardware limitations rather than requiring specialized equipment.

3.1.1 Asymmetric Communication Model

A central architectural decision is the adoption of an asymmetric communication model where information flows primarily in one direction: from infrastructure to clients. This contrasts with symmetric models where clients and infrastructure exchange bidirectional messages.

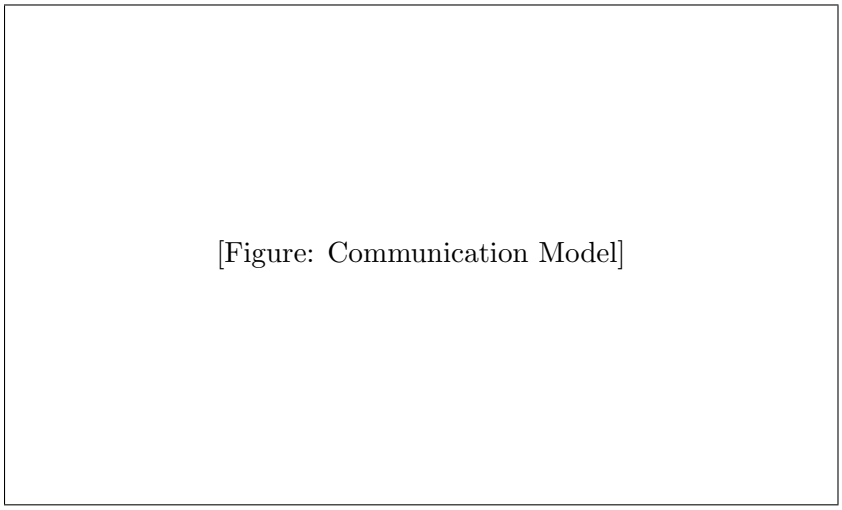

Figure 3.1: Asymmetric communication model. Solid arrows indicate primary data flow; dashed arrows indicate optional messages.

As illustrated in Figure **??**, beacons broadcast their positions, and UAVs passively receive these broadcasts without transmitting. This model provides several advantages:

- **Unlimited Clients:** Since UAVs do not transmit during normal operation, any number of UAVs can simultaneously use the system without increasing channel congestion.
- **Position Privacy:** UAVs determine their positions locally without revealing them to the infrastructure or other parties.
- **Simplified Protocol:** No handshaking, acknowledgment, or traffic management is required, reducing system complexity.

The trade-off is that time-based ranging (ToF/TDoA) becomes impractical without bidirectional communication, necessitating the RSSI-based approach developed in this thesis.

3.2 Infrastructure Segment

The infrastructure segment consists of fixed beacons deployed throughout the operational area. These beacons serve as position references for client UAVs, analogous to the role of satellites in GNSS but at much shorter ranges.

3.2.1 Beacon Hardware

Each beacon comprises:

- A LoRa transceiver module (e.g., SX1276/SX1278) operating in unlicensed ISM bands

- A microcontroller for packet generation and timing
- A power supply (battery, solar panel, or mains connection)
- An omnidirectional antenna, typically a quarter-wave monopole or dipole

The total hardware cost per beacon is under $50 using commodity components, enabling cost-effective deployment of dozens of beacons.

3.2.2 Operating Modes

Beacons operate in two distinct modes depending on system phase:

Mode A: Self-Organization During initial deployment or periodic recalibration, beacons measure RSSI values from neighboring beacons and report these measurements to the central hub. The hub aggregates measurements from all beacons to construct a network-wide connectivity graph with edge weights derived from RSSI. Using this graph, the hub executes the infrastructure self-localization algorithm (Chapter **??**) to estimate beacon positions.

In this mode, beacons transmit identification packets at randomized intervals to enable neighbor discovery. Randomization prevents systematic collisions that would occur with synchronized transmissions. The self-organization process typically completes within minutes, after which estimated positions are distributed back to beacons.

Mode B: Service Operation During normal operation, beacons periodically broadcast packets containing:

- Beacon identifier (unique ID)
- Estimated position (latitude, longitude, altitude)
- Transmission power level
- Optional: uncertainty estimate or quality indicator

Broadcast intervals are randomized within a configurable window (e.g., 1–5 seconds) to avoid collisions while ensuring that UAVs receive sufficient measurements for localization within acceptable latency bounds.

3.2.3 Energy Management

For battery-powered deployments, energy conservation is critical for extended operation. We incorporate a wake-on-demand mechanism:

1. **Sleep State:** By default, beacons enter deep sleep mode with only a low-power wake-up receiver active, consuming microamps of current.
2. **Wake-up Signal:** When a UAV enters an area and requires positioning service, it broadcasts a single wake-up signal. This is the only transmission required from UAVs.
3. **Active Period:** Upon receiving the wake-up signal, beacons transition to Mode B for a predetermined duration (e.g., 5–10 minutes), providing positioning service.

4. **Return to Sleep:** After the active period expires without additional wake-up signals, beacons return to sleep mode.

This mechanism enables month-long battery operation while providing on-demand service. Multiple UAVs benefit from a single wake-up signal, and subsequent UAVs entering the area can extend the active period with additional wake-up transmissions.

3.3 Client Segment

The client segment consists of mobile UAVs that utilize beacon signals for positioning. The design emphasizes passive operation to maximize scalability and preserve position privacy.

3.3.1 UAV Hardware

Each UAV requires:

- A LoRa receiver module (transmitter capability optional, needed only for wake-up)
- A microcontroller or flight computer for signal processing and localization computation
- An altitude sensor (barometer or laser altimeter) for vertical position reference
- Integration with the flight control system for navigation

The LoRa receiver can be the same module used for telemetry or command links in many UAV platforms, requiring minimal additional hardware.

3.3.2 Positioning Process

The UAV positioning process operates as follows:

1. **Signal Reception:** The UAV continuously monitors the LoRa channel, receiving beacon broadcasts and extracting RSSI values along with beacon position data.
2. **Measurement Caching:** Received measurements are stored in a sliding window cache, typically retaining measurements from the past 5–30 seconds depending on UAV speed and beacon density.
3. **Altitude Determination:** The UAV determines its altitude independently using barometric pressure (calibrated to local sea level pressure) or a downward-facing laser altimeter measuring height above ground.
4. **Position Estimation:** When sufficient measurements are available (minimum 3 beacons for 2D, 4 for 3D without altitude constraint), the UAV executes the multilateration algorithm described in Chapter **??**.
5. **Filtering:** Position estimates are optionally filtered using a Kalman filter or similar state estimator to smooth trajectory and reject outliers.

3.3.3 Scalability Analysis

The passive reception model provides theoretical unlimited scalability. Consider a scenario with B beacons each transmitting at average interval T seconds with packet duration τ:

$$\text{Channel utilization} = \frac{B \cdot \tau}{T} \tag{3.1}$$

For $B = 20$ beacons, $\tau = 100$ ms packets, and $T = 3$ s average interval, channel utilization is approximately 0.67%, leaving ample capacity for the channel regardless of how many UAVs are receiving.

In contrast, a symmetric model where each of N UAVs transmits would have utilization proportional to N, creating congestion as the UAV count increases.

3.4 Hub Segment

The hub segment provides centralized computation for infrastructure self-localization, which requires global knowledge of the beacon network topology.

3.4.1 Hub Deployment Options

The hub can be implemented in several configurations depending on operational requirements:

Fixed Base Station For permanent or semi-permanent deployments, the hub is a ground-based computing platform with reliable power and network connectivity. This configuration provides maximum computational resources and can support large beacon networks.

Mobile Command Post For deployable scenarios, the hub is integrated into a vehicle-mounted command post that can relocate as operations evolve.

Aerial Hub (Mother UAV) For fully mobile operations, a larger UAV serves as the hub, communicating with beacons via LoRa and performing on-board computation. This configuration enables truly infrastructure-free deployment where all system components are transportable by air.

3.4.2 Hub Functions

The hub performs the following functions:

Data Aggregation During self-organization (Mode A), the hub collects RSSI measurements from all beacons, constructing the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices represent beacons and edges represent measured links with RSSI-derived weights.

Self-Localization Computation The hub executes the physics-aware GCN algorithm (Chapter **??**) to estimate beacon positions from the network graph. This computation is performed once during initial deployment and can be repeated periodically to account for environmental changes or beacon drift.

Position Distribution After computing beacon positions, the hub transmits position updates to each beacon. Beacons store their assigned positions and include them in subsequent broadcasts.

Anchor Management A small subset of beacons may have known positions from survey or GPS fixes obtained before GNSS denial. The hub manages anchor designations, ensuring the self-localization algorithm is constrained by available ground truth.

3.5 Pre-Deployment Planning and Model Training

Before physical deployment, a planning phase prepares the system for the target operational area. This phase serves two purposes: assessing deployment feasibility and training the machine learning model that will be used for infrastructure self-localization.

3.5.1 Terrain-Aware Simulation

The planning phase begins with acquisition of terrain data for the target area, such as digital elevation models from satellite-derived datasets. Using this terrain information, the system generates synthetic training data by simulating radio propagation between hypothetical beacon placements.

The simulation employs a terrain-aware propagation model, specifically the Longley-Rice Irregular Terrain Model, which accounts for:

- Path loss due to distance and frequency
- Diffraction over terrain obstacles
- Tropospheric scatter effects
- Surface reflections based on ground conductivity

For each simulated scenario, beacons are placed at random locations within the operational area, and pairwise RSSI measurements are computed based on the propagation model with added noise to simulate real-world variability. Ground truth positions are retained for supervised training.

3.5.2 Feasibility Assessment

The simulation framework enables feasibility assessment before committing resources to physical deployment. Key questions addressed include:

Minimum Beacon Count Given the terrain complexity and target accuracy requirements, how many beacons are needed? Simulations with varying beacon densities reveal the accuracy-cost trade-off for the specific area.

Placement Recommendations While exact beacon placement need not be surveyed, simulations can identify regions where beacon placement would be particularly beneficial (e.g., hilltops with good line-of-sight) or problematic (e.g., deep valleys with limited connectivity).

Expected Accuracy Running the trained model on held-out simulated scenarios provides accuracy estimates for the deployment. If expected accuracy falls below mission requirements, the deployment plan can be revised before field operations begin.

3.5.3 Model Training

The physics-aware GCN (detailed in Chapter **??**) is trained on the synthetic dataset generated through terrain-aware simulation. Training occurs on standard computing infrastructure with GPU acceleration, producing a model that has learned:

- The relationship between RSSI measurements and distances in terrain-rich environments
- How to aggregate information across the network graph to resolve position ambiguities
- Robustness to the noise characteristics of the target propagation environment

The trained model weights are then loaded onto the hub for deployment. During field operations, the hub performs inference only, requiring modest computational resources compared to training.

3.5.4 Simulation-to-Real Transfer

A limitation of the current approach is that both training and the evaluations presented in Chapter **??** use simulated data. While the Longley-Rice model provides realistic terrain effects, discrepancies between simulation and real-world propagation are inevitable.

Future work could address this gap through fine-tuning: after collecting real RSSI measurements during self-organization, the pre-trained model could be adapted using a small number of gradient updates on real data. This simulation-to-real transfer would improve accuracy while retaining the benefits of terrain-aware pre-training.

3.6 System Integration

Figure **??** illustrates the complete system architecture and data flows between segments.

3.6.1 Operational Timeline

A typical deployment follows this timeline:

1. **Pre-Deployment Planning** (days to weeks before): Terrain data is acquired, feasibility assessment is conducted through simulation, and the GCN model is trained. The trained model is loaded onto the hub.
2. **Physical Deployment** (0–2 hours): Beacons are placed at suitable locations throughout the operational area. Placement can be approximate; exact surveying is not required.

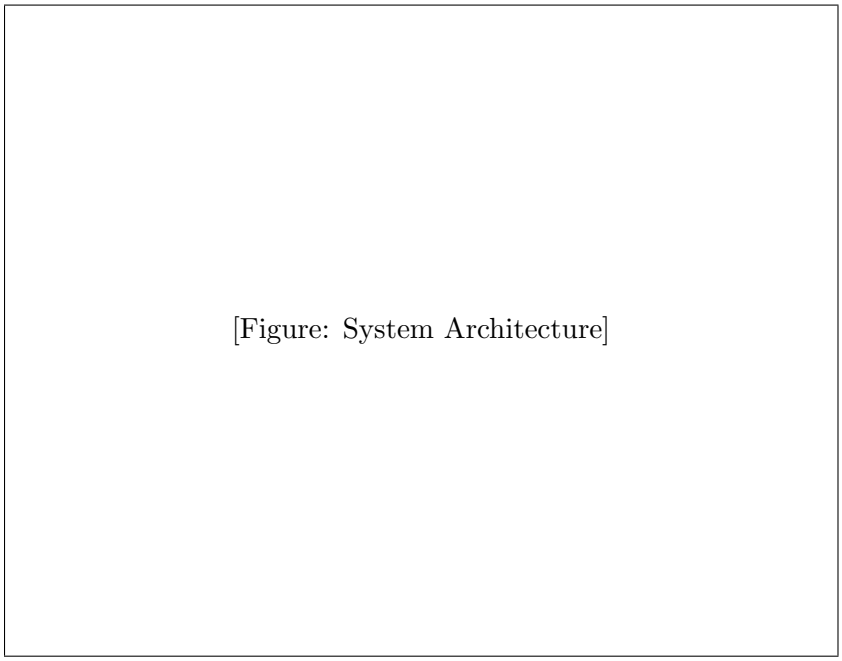

Figure 3.2: System architecture showing beacons (circles), UAVs (diamonds), and hub. Dashed lines indicate beacon-to-beacon measurements during self-organization; solid blue arrows show beacon broadcasts received by UAVs; orange arrows show hub-to-beacon communication.

3. **Self-Organization** (15–30 minutes): Beacons enter Mode A, exchanging signals and reporting measurements to the hub. The hub runs inference using the pre-trained GCN to estimate beacon positions from the network graph.

4. **Position Distribution** (5 minutes): The hub transmits estimated positions to each beacon, which store them for subsequent broadcasts.

5. **Service Operation** (indefinite): Beacons enter Mode B, broadcasting positions for UAV consumption. The system is now operational for client positioning.

6. **Periodic Recalibration** (optional): Self-organization can be repeated periodically (e.g., daily) to account for environmental changes affecting propagation characteristics.

3.6.2 Failure Modes and Recovery

The system is designed for graceful degradation under component failures:

Beacon Failure If individual beacons fail, UAVs can still localize using remaining beacons, with accuracy degrading proportionally to coverage reduction. During the next self-organization cycle, the hub detects missing beacons and recomputes positions for the reduced network.

Hub Failure Hub failure does not immediately affect UAV positioning since beacons continue broadcasting with their last known positions. Service continues until positions become stale due to environmental changes or beacon drift.

Partial Network Partition If a subset of beacons loses connectivity to the hub, that subset continues operating with last-known positions while connected beacons can be recalibrated.

3.7 Summary

This chapter has presented the system architecture for GNSS-denied positioning:

- The **infrastructure segment** provides position references through beacon broadcasts, with self-organization capability for deployment without surveying.
- The **client segment** enables scalable, privacy-preserving positioning through passive reception and local computation.
- The **hub segment** performs centralized self-localization inference using a pre-trained model, requiring global network knowledge but operating only during deployment and recalibration.
- The **pre-deployment phase** conducts feasibility assessment through terrain-aware simulation and trains the GCN model before field operations.

The following chapters detail the algorithms that enable this architecture: Chapter **??** addresses client-side positioning (UAV localization given beacon positions), and Chapter **??** addresses infrastructure-side positioning (beacon self-localization from mutual measurements).

Chapter 4

Robust UAV Localization with Altitude Constraints

This chapter addresses the client-side localization problem: determining the position of a mobile UAV using RSSI measurements from beacons at known locations. We present an enhanced multilateration algorithm that combines the Huber loss function with altitude constraints to achieve robust positioning despite the high variance inherent in RSSI measurements. The approach is validated through both network simulations and real-world flight experiments.

4.1 Problem Formulation

4.1.1 Scenario

Consider a UAV operating in an environment where GNSS signals are unavailable or unreliable. The operational area is equipped with N beacons at known positions $\mathbf{p}_i = (X_i, Y_i, Z_i)$ for $i = 1, \ldots, N$, expressed in a local Cartesian coordinate system. Each beacon periodically broadcasts LoRa packets containing its identifier and position. The UAV receives these broadcasts and measures the Received Signal Strength Indicator (RSSI) for each packet.

The localization task is to estimate the UAV position $\hat{\mathbf{p}} = (\hat{X}, \hat{Y}, \hat{Z})$ from the set of beacon positions and corresponding RSSI measurements $\{(\mathbf{p}_i, \mathrm{RSSI}_i)\}_{i=1}^{N}$.

4.1.2 Challenges

RSSI-based localization faces several challenges that degrade accuracy:

Signal Variance RSSI values exhibit large variance due to multipath propagation, shadowing from obstacles, and environmental factors such as humidity and temperature. A single distance may produce RSSI values spanning 10–20 dB, making point estimates unreliable.

Non-Line-of-Sight (NLOS) When direct paths between beacons and UAVs are obstructed, signals arrive via reflections and diffractions, causing systematic overestimation of distances.

Antenna Effects Practical antennas have non-uniform radiation patterns. Monopole antennas commonly used in LoRa modules exhibit reduced gain directly above the antenna (the "cone of silence"), causing signal strength variations that depend on geometry rather than distance alone.

Three-Dimensional Geometry Unlike ground-based localization where altitude is approximately constant, UAV localization is inherently three-dimensional. However, the vertical dimension is poorly constrained by horizontal beacon arrangements, leading to large altitude errors that can corrupt horizontal position estimates.

4.1.3 Objectives

The goal is to develop a localization algorithm that:

1. Achieves robust accuracy despite RSSI variance, without requiring additional hardware such as synchronized clocks or antenna arrays
2. Scales effectively with increasing beacon density
3. Provides a mechanism to detect and reject unreliable estimates

4.2 Proposed Method

4.2.1 System Overview

The proposed system places multiple beacons at elevated positions (rooftops, towers, or poles) throughout the operational area. Each beacon transmits LoRa packets at randomized intervals, with each packet containing:

- Beacon identifier
- Beacon position (latitude, longitude, altitude)
- Transmission power level

The UAV operates as a passive receiver, extracting RSSI values from received packets without transmitting. This unidirectional communication model eliminates the need for traffic management and enables unlimited simultaneous UAV users, as discussed in Chapter **??**.

Received measurements are cached in a sliding window, and localization is performed when sufficient measurements from distinct beacons are available (minimum 3 for 2D with altitude constraint, 4 for unconstrained 3D).

4.2.2 RSSI-to-Distance Conversion

The first step in multilateration is converting RSSI measurements to distance estimates. Following the log-distance path-loss model introduced in Chapter **??**, we invert Equation **??** to obtain the distance estimate $\hat{d}_i$ from RSSI measurement $P_{r,i}$:

$$\hat{d}_i = 10^{(P_t + o - P_{r,i})/(10n)} \tag{4.1}$$

where P_t is the transmitted power, o is the offset, and n is the path-loss exponent.

The parameters n and o are determined through least-squares regression on calibration measurements collected in controlled conditions (see Section **??** for calibration procedure). This model captures the fundamental log-distance relationship while remaining simple enough to calibrate with limited data. More sophisticated models incorporating altitude or terrain effects are discussed in Section **??**.

4.2.3 Multilateration Formulation

Given distance estimates $\hat{d}_i$ to beacons at known positions $\mathbf{p}_i$, multilateration finds the position $\hat{\mathbf{p}}$ that best satisfies the distance constraints. The geometric constraint for beacon i is:

$$\|\hat{\mathbf{p}} - \mathbf{p}_i\| = \hat{d}_i \tag{4.2}$$

or equivalently:

$$(\hat{X} - X_i)^2 + (\hat{Y} - Y_i)^2 + (\hat{Z} - Z_i)^2 = \hat{d}_i^2 \tag{4.3}$$

Since RSSI-derived distances contain errors, the constraints cannot be satisfied exactly. The localization problem becomes an optimization:

$$\hat{\mathbf{p}}^* = \arg\min_{\hat{\mathbf{p}}} \sum_{i=1}^{N} \mathcal{L}(r_i) \tag{4.4}$$

where $r_i = \|\hat{\mathbf{p}} - \mathbf{p}_i\| - \hat{d}_i$ is the residual for beacon i, and $\mathcal{L}(\cdot)$ is a loss function.

4.2.4 Huber Loss Function

The conventional approach uses squared loss:

$$\mathcal{L}_{\text{L2}}(r) = r^2 \tag{4.5}$$

Squared loss is optimal when errors follow a Gaussian distribution, but performs poorly when outliers are present. Large errors receive disproportionate weight, potentially pulling the position estimate far from the true location.

We propose using the Huber loss function [**?**]:

$$\mathcal{L}_{\text{Huber}}(r) = \begin{cases} \frac{1}{2}r^2 & \text{if } |r| \leq \delta \\ \delta|r| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{4.6}$$

where δ is a threshold parameter.

The Huber loss combines the advantages of both squared and absolute loss:

- For small residuals ($|r| \leq \delta$), it behaves like squared loss, providing smooth optimization and sensitivity to small errors
- For large residuals ($|r| > \delta$), it behaves like absolute loss, limiting the influence of outliers to linear rather than quadratic growth

This property is particularly valuable for RSSI-based localization, where occasional large measurement errors are common due to multipath and NLOS conditions. The threshold δ controls the transition point; we use $\delta = 100$ m in our experiments, though the algorithm is not highly sensitive to this choice.

4.2.5 Altitude Constraints

A key insight is that UAVs can obtain reliable altitude information from onboard sensors independent of the RF positioning system:

- **Barometric altimeters** measure atmospheric pressure, providing altitude relative to a calibrated reference (typically sea level or takeoff location) with accuracy of 1–5 m
- **Laser altimeters** measure height above ground using time-of-flight of laser pulses, providing centimeter-level accuracy over smooth terrain

We incorporate altitude as a hard constraint in the optimization. Let Z_{sensor} denote the altitude obtained from onboard sensors. The constrained optimization becomes:

$$\hat{\mathbf{p}}^* = \arg\min_{\hat{X},\hat{Y}} \sum_{i=1}^{N} \mathcal{L}_{\text{Huber}}(r_i) \quad \text{subject to} \quad \hat{Z} = Z_{\text{sensor}} \tag{4.7}$$

This constraint provides two benefits:

Dimensionality Reduction The optimization reduces from 3D to 2D, decreasing computational complexity and improving convergence.

Error Decoupling Without altitude constraints, vertical position errors can corrupt horizontal estimates due to the coupled nature of the spherical distance constraints. By fixing altitude to a reliable sensor value, we prevent this error propagation.

4.2.6 Implementation

The optimization in Equation **??** is solved using iterative nonlinear least squares. The algorithm proceeds as follows:

1. Initialize $(\hat{X}, \hat{Y})$ at the centroid of beacon positions
2. Set $\hat{Z} = Z_{\text{sensor}}$
3. Compute residuals $r_i = \sqrt{(\hat{X} - X_i)^2 + (\hat{Y} - Y_i)^2 + (\hat{Z} - Z_i)^2} - \hat{d}_i$
4. Compute Huber weights w_i based on residual magnitudes
5. Update $(\hat{X}, \hat{Y})$ using weighted least squares
6. Repeat steps 3–5 until convergence

The algorithm typically converges within 10–20 iterations. The final minimization error $\sum_i \mathcal{L}_{\text{Huber}}(r_i)$ serves as a quality indicator, as discussed in Section **??**.

4.3 Simulation Study

We evaluate the proposed method through network simulations that model realistic urban propagation conditions.

4.3.1 Simulation Setup

Simulation Framework The simulation is built on OMNeT++, a discrete event simulation framework for network systems. We use the INET framework for wireless network modeling, the FLoRa extension [**?**] for LoRa-specific parameters, and OSM Buildings for 3D urban geometry rendering.

Environment The simulation environment models an urban area in Berlin using a 3D map derived from OpenStreetMap. Specifically, we use one tile at zoom level 15, corresponding to an area of 0.5537 km^2. The 3D building models provide realistic signal obstruction and reflection characteristics for urban propagation. Ten beacons are placed on randomly selected building rooftops, ensuring elevated positions with good line-of-sight to the sky. The UAV operates at a fixed altitude of 100 m, above most buildings but within range of rooftop beacons.

Radio Parameters Table **??** summarizes the LoRa radio parameters used in the simulation. These parameters represent typical LoRa configurations for long-range, low-power communication.

Table 4.1: Simulation parameters

Parameter	**Value**
Spreading Factor (SF)	7
Transmission Power	14 dBm
Bandwidth	500 kHz
Path Loss Model	Nakagami Fading
Beacon Transmission Interval	1 s
Map Area	0.5537 km^2 (Berlin)
Number of Beacons	10
UAV Altitude	100 m

RSSI-to-Distance Model Calibration The conversion model (Equation **??**) requires calibration of path-loss parameters n and o. We conduct preliminary simulations in an obstacle-free environment to collect RSSI measurements at known distances. Least-squares regression on this data determines the optimal parameter values. This controlled calibration setting allows accurate characterization of the RSSI-distance relationship without interference from building-induced effects. The resulting parameters are then applied during localization in the urban environment, where the algorithm must handle building effects as additional noise. This approach mirrors practical deployment, where calibration would be performed in open areas before operating in complex environments.

4.3.2 Compared Methods

We compare four methods representing all combinations of loss functions (squared vs. Huber) and altitude constraints (with vs. without). Table **??** summarizes the method configurations.

Table 4.2: Compared localization methods

Method	**Description**	**Loss**	**Alt. Constraint**
1	Conventional baseline	Squared (L2)	No
2	Altitude-constrained	Squared (L2)	Yes
3	Robust loss	Huber	No
4	Proposed method	Huber	Yes

- **Method 1 (Squared Loss):** The conventional multilateration baseline using squared error loss $\mathcal{L} = \sum_i (d_i - \hat{d}_i)^2$, optimizing over all three coordinates (X, Y, Z).
- **Method 2 (Squared Loss + Altitude):** Squared error loss with the altitude constraint, fixing Z to the sensor-derived value and optimizing only (X, Y).
- **Method 3 (Huber Loss):** Huber loss function without altitude constraints, providing outlier robustness but optimizing over all three coordinates.

- **Method 4 (Huber Loss + Altitude):** The proposed method combining Huber loss for outlier robustness with altitude constraints for error decoupling.

This factorial design allows us to isolate the individual contributions of each improvement (Huber loss and altitude constraints) as well as their combined effect.

4.3.3 Evaluation Scenarios

Static Scenario The UAV hovers at a fixed position while receiving beacon signals for 500 seconds. RSSI values are averaged over this period before localization, reducing noise to evaluate the fundamental accuracy of each method. This scenario isolates algorithmic differences from temporal sampling effects.

Dynamic Scenario The UAV flies in a circular path (200 m radius) at 10 m/s, representing typical flight operations. Localization is performed at 1 Hz using a 5-second sliding window of measurements, matching the update rate of typical GNSS receivers.

4.3.4 Results

Static Analysis

Figures **??**–**??** show boxplots of localization error for each method across varying numbers of reference beacons. These results are obtained by generating all possible combinations of beacon subsets and computing localization error for each combination.

Table **??** presents the static localization results with all 10 beacons. The proposed method (Method 4) achieves a median error of 4.50 m in both 2D and 3D, representing an 85% improvement over the conventional baseline (Method 1) which achieves 29.70 m in 2D.

Table 4.3: Static simulation results (10 beacons)

Method	**2D Error (m)**	**3D Error (m)**
Method 1 (L2)	29.70	44.34
Method 2 (L2 + Alt)	30.27	30.27
Method 3 (Huber)	5.72	158.05
Method 4 (Huber + Alt)	4.50	4.50

Several observations emerge from these results:

Altitude Constraints are Essential for 3D Comparing the 3D results in Figures **??** and **??**, methods without altitude constraints exhibit significantly larger vertical errors. Method 3 (Huber without altitude) achieves good 2D accuracy (5.72 m) but catastrophic 3D accuracy (158.05 m). The unconstrained vertical dimension absorbs optimization residuals, producing large altitude errors that would be problematic for actual flight. Altitude constraints eliminate this failure mode entirely, as seen in Figures **??** and **??**.

(a) 2D error

(b) 3D error

Figure 4.1: Static localization error for Method 1 (squared loss, no altitude constraint). Large 3D errors result from unconstrained vertical estimation.

Figure 4.2: Static localization error for Method 2 (squared loss with altitude constraint). 2D and 3D errors are identical due to the altitude constraint.

Huber Loss Improves Robustness Comparing Methods 1 and 3 (both without altitude constraints), Huber loss reduces 2D error from 29.70 m to 5.72 m, demonstrating its effectiveness at suppressing outlier influence from noisy RSSI measurements.

Combined Approach is Optimal Method 4 combines both improvements, achieving the best performance in all metrics. Note that Figure **??**(a) shows occasional large errors unique to Method 4; these outliers are addressed through error monitoring as discussed below.

Scalability Analysis

Figure **??** shows how localization accuracy varies with the number of available beacons. A striking difference emerges: methods using Huber loss (3, 4) improve monotonically with additional beacons, while methods using squared loss (1, 2) degrade.

This counterintuitive behavior for squared loss methods occurs because additional beacons increase the probability of encountering outlier measurements. Under squared loss, these outliers receive disproportionate weight, corrupting the overall solution. Huber loss limits outlier influence, allowing additional measurements to improve accuracy through averaging.

This scalability property is crucial for practical deployment, where denser beacon networks should yield better performance.

(a) 2D error

(b) 3D error

Figure 4.3: Static localization error for Method 3 (Huber loss, no altitude constraint). Good 2D accuracy but catastrophic 3D errors due to unconstrained vertical dimension.

(a) Full scale

(b) Cropped view

Figure 4.4: Static localization error for Method 4 (proposed: Huber loss with altitude constraint). (a) shows occasional large errors that can be detected via minimization error monitoring; (b) shows the typical error distribution achieving 4.5 m median.

Figure 4.5: Localization error versus number of beacons. Huber loss methods (3, 4) improve with additional beacons while squared loss methods (1, 2) degrade.

Error Monitoring

Figure **??** shows the relationship between the final minimization error (sum of Huber losses) and the true localization error for Method 4. A strong linear correlation ($R^2 = 0.99$) indicates that large localization errors produce correspondingly large minimization errors.

This correlation enables a practical error detection mechanism: by monitoring the minimization error and comparing against a threshold, the system can flag potentially unreliable position estimates. Such estimates can be rejected or weighted lower in downstream navigation filters.

Dynamic Analysis

Table **??** presents the dynamic simulation results. As expected, accuracy degrades compared to the static scenario due to reduced averaging time and position changes during the measurement window.

Method 4 achieves median error of 62.98 m with standard deviation of 43.05 m, representing a 25% improvement in median error and 59% reduction in standard deviation compared to Method 1. The reduced variance is particularly important for navigation applications, where consistent accuracy is often more valuable than occasional high accuracy.

Figure 4.6: Correlation between minimization error and localization error for Method 4. The strong correlation ($R^2 = 0.99$) enables detection of unreliable estimates.

Table 4.4: Dynamic simulation results (10 beacons, 10 m/s)

Method	**2D Med.**	**2D Std**	**3D Med.**	**3D Std**
Method 1	84.16	105.16	145.98	114.63
Method 2	87.73	109.76	87.73	109.76
Method 3	67.96	43.48	89.88	48.88
Method 4	62.98	43.05	62.98	43.05

Speed-Accuracy Trade-off Additional simulations at reduced speed (1 m/s) yield median error of 33.22 m and standard deviation of 14.36 m with Method 4. This suggests that slowing the UAV during critical phases (takeoff, landing, waypoint arrival) can significantly improve positioning accuracy.

4.4 Experimental Study

To validate the simulation results and assess practical feasibility, we conducted field experiments with real hardware.

4.4.1 Hardware

Table **??** lists the hardware components used in the experiments. All components are commercially available, demonstrating the approach's compatibility with commodity equipment.

Table 4.5: Experimental hardware

Component	**Model**
Computer	Raspberry Pi 3B
LoRa Module	RFM95W
Antenna	VERT900
UAV Platform	Holybro X500 V2
Flight Controller	Pixhawk 6C
GPS Module	G8N (for ground truth)

Each beacon consists of a Raspberry Pi connected to an RFM95W LoRa module with a VERT900 omnidirectional antenna. The UAV carries an identical receiver configuration, with the flight controller providing altitude data and GPS providing ground truth for evaluation.

4.4.2 Experimental Setup

Figure 4.7: Beacon placement for field experiments in Pergine, Italy. Four beacons are arranged in a rectangular configuration spanning 610 m^2.

Experiments were conducted in a parking area in Pergine, Italy. Four beacons were deployed in a rectangular configuration, as shown in Figure **??**, covering an area of 610 m^2. Beacons transmitted at intervals drawn from a normal distribution with mean 0.5 seconds.

Four flight experiments were conducted with varying control modes and trajectories:

- **Experiment 1:** Manual control, oval trajectory, 5–10 m altitude

(a) (b)

Figure 4.8: Experimental hardware: (a) UAV platform with LoRa receiver, (b) beacon with Raspberry Pi and antenna.

- **Experiment 2:** Manual control, figure-eight (pylon eights) trajectory, 5–10 m altitude
- **Experiment 3:** Autonomous control, oval trajectory, 15 m altitude
- **Experiment 4:** Autonomous control, figure-eight trajectory, 15 m altitude

Ground Truth Evaluation Evaluating localization accuracy requires comparing estimated positions against ground truth at the same time instants. However, beacon signals arrive more frequently than GPS updates, creating a temporal mismatch: LoRa packets may be received at times between GPS fixes. We address this by employing linear interpolation on the GPS trajectory data, estimating the UAV's GPS position at the precise moment of each LoRa signal reception. This interpolation approach ensures accurate alignment between the localization estimate (computed from the received LoRa packet) and the corresponding ground truth position.

4.4.3 Results

Table **??** presents the experimental results. The format shows median error and standard deviation for 2D/3D measurements; methods with altitude constraints show identical 2D and 3D values.

Table 4.6: Experimental results (median error / standard deviation in meters)

Exp.	**Method 1**	**Method 2**	**Method 3**	**Method 4**
1 (Manual, Oval)	7.41/9.78	7.69	7.38/12.60	**7.04**
2 (Manual, Fig-8)	8.65/10.53	8.70	8.71/10.96	8.59
3 (Auto, Oval)	26.93/29.73	25.86	26.80/31.04	**25.91**
4 (Auto, Fig-8)	16.01/19.27	14.47	16.01/20.79	**14.38**

The proposed method (Method 4) achieves the best or near-best accuracy in all experiments, with median errors of 7.04 m in the best case (Experiment 1) and 25.91 m in the most challenging case (Experiment 3).

Comparison with Simulation The experimental results show smaller improvements over the baseline compared to simulations. This is attributed to the limited number of beacons (4 versus 10 in simulation); with fewer beacons, the probability of outliers is lower, reducing the advantage of Huber loss. Additionally, the small test area (610 m^2) limits distance diversity, reducing the dynamic range of RSSI values.

Altitude Effects Experiments 3 and 4 (15 m altitude) show degraded accuracy compared to Experiments 1 and 2 (5–10 m altitude). This is attributed to antenna directivity effects, discussed in Section **??**.

4.5 Discussion

4.5.1 Antenna Directivity Effects

Figure **??** shows the RSSI-to-distance relationship derived from experimental data at different altitudes. At low altitude, the expected inverse relationship holds: RSSI decreases with distance. At high altitude, this relationship inverts near the beacons due to antenna directivity.

The VERT900 antenna is a quarter-wave monopole with a radiation pattern that has a null directly overhead. When the UAV is positioned above a beacon at high altitude, it falls within this "cone of silence," receiving weaker signals despite the short distance. This effect violates the assumptions of the simple log-distance model.

Mitigation Approaches Two approaches can address antenna directivity effects:

1. **Altitude-Aware RSSI Model:** Extend the RSSI-to-distance model to include altitude as a parameter:

$$d_i = f(\text{RSSI}_i, Z_{\text{UAV}}, Z_{\text{beacon}}) \tag{4.8}$$

   Figure **??** illustrates an altitude-parameterized model fitted to experimental data from Experiments 1 and 2. The 3D surface captures how signal strength varies with both horizontal distance and vertical separation, more accurately representing the antenna's radiation pattern. This approach requires more extensive calibration data collection across multiple altitudes but can substantially improve accuracy in scenarios where UAV altitude varies.

2. **Antenna Selection:** Use antennas with more uniform radiation patterns, such as circularly polarized patch antennas designed for aerial applications.

4.5.2 Sampling Time Optimization

Figure **??** shows how localization accuracy varies with sampling time. Longer sampling windows reduce noise through averaging, improving accuracy in static or slow-moving scenarios. However, in dynamic scenarios, extended sampling introduces position lag and averages measurements taken at different locations.

Adaptive sampling strategies could optimize this trade-off: using long windows during hover or slow transit, and short windows during rapid maneuvers. Integration with the flight controller could enable automatic adjustment based on planned or observed UAV dynamics.

(a) Low altitude (Experiments 1–2)

(b) High altitude (Experiments 3–4)

Figure 4.9: RSSI-to-distance relationships at different altitudes. The inverted relationship at high altitude results from antenna directivity.

Figure 4.10: Signal strength variation with UAV altitude over a beacon. Higher altitude brings the UAV into the antenna's null region.

Figure 4.11: Altitude-parameterized RSSI-to-distance model. The 3D surface shows how signal strength depends on both horizontal distance and altitude difference, capturing antenna directivity effects.

(a) Median error

(b) Standard deviation

Figure 4.12: Effect of sampling time on localization accuracy. Longer sampling improves static accuracy but may introduce lag in dynamic scenarios.

4.5.3 Scalability Considerations

The system scales favorably in several dimensions:

Number of UAVs The unidirectional communication model (beacons broadcast, UAVs receive) supports unlimited simultaneous UAVs without increasing channel load. Each additional UAV performs localization independently using the same beacon signals.

Coverage Area LoRa's 2–15 km range enables sparse beacon deployment. Our simulations suggest that 18 beacons/km^2 achieves 11.6 m median static accuracy, while 7.2 beacons/km^2 achieves 4.5 m accuracy. For a 10 km^2 operational area, this translates to 72–180 beacons, a manageable infrastructure investment.

Beacon Density Unlike squared loss methods that degrade with additional beacons, the Huber loss approach improves monotonically. This enables incremental deployment: starting with minimal beacons and adding more as accuracy requirements increase.

4.5.4 Comparison with GPS

From the user perspective, both GPS and the proposed system provide position fixes from commodity receivers with similar form factors and power consumption. LoRa receivers actually consume less power than GPS receivers, offering potential battery life advantages.

From the operator perspective, the proposed system requires ground-based infrastructure (beacons) rather than space-based infrastructure (satellites). While satellites provide global coverage, beacons provide local coverage that can be rapidly deployed and reconfigured. For applications requiring GNSS backup in specific areas, beacon-based systems offer a practical and cost-effective alternative.

4.6 Summary

This chapter has presented a robust UAV localization method using RSSI measurements from LoRa beacons. The key contributions are:

1. **Huber Loss for Outlier Robustness:** Replacing squared loss with Huber loss suppresses the influence of RSSI outliers, improving accuracy from 29.7 m to 5.7 m in static conditions.

2. **Altitude Constraints for Error Decoupling:** Incorporating barometric or laser altimeter data prevents vertical errors from corrupting horizontal estimates, achieving consistent 2D/3D accuracy.

3. **Scalability:** Unlike conventional methods that degrade with additional beacons, the proposed approach improves monotonically, enabling effective use of dense beacon networks.

4. **Error Detection:** The strong correlation between minimization error and localization error enables detection of unreliable estimates.

Simulation results demonstrate median error of 4.5 m in static conditions and 63 m in dynamic conditions (10 m/s flight). Field experiments validate the approach with real hardware, achieving 7 m median error in favorable conditions.

The method assumes beacon positions are known, either from prior survey or from the infrastructure self-localization process described in the following chapter.

Chapter 5

Infrastructure Self-Localization via Graph Learning

This chapter addresses the infrastructure-side localization problem: determining the positions of beacons themselves when only a subset have known coordinates. Unlike the client-side problem of Chapter **??**, where beacon positions are given, here we must infer beacon positions from pairwise RSSI measurements within the beacon network. We propose a physics-aware Graph Convolutional Network (GCN) that learns to localize beacons by combining a trainable RSSI-to-distance module with graph-based message passing.

5.1 Problem Formulation

5.1.1 Scenario

Consider a network of N static beacons deployed in an outdoor area where GNSS is unavailable or unreliable. A subset of beacons, called anchors, have known positions obtained through prior survey, brief GNSS access before denial, or placement at pre-determined coordinates (e.g., airdrop with GPS logging). The remaining beacons, called unknown nodes, must determine their positions using only radio measurements exchanged with neighboring nodes.

This scenario arises in rapid deployment situations where comprehensive surveying is impractical: disaster response, scientific expeditions, or remote monitoring in GNSS-challenged environments.

5.1.2 Graph Formulation

We model the beacon network as a directed graph $G = (\mathcal{V}, \mathcal{E})$ where:

- $\mathcal{V}$ is the set of all nodes with $|\mathcal{V}| = N$
- $\mathcal{A} \subset \mathcal{V}$ is the set of anchor nodes with known positions
- $\mathcal{U} = \mathcal{V} \setminus \mathcal{A}$ is the set of unknown nodes to be localized
- $\mathcal{E}$ is the set of directed edges representing communication links

Each node i has an absolute 2D position $\mathbf{p}_i = (x_i, y_i) \in \mathbb{R}^2$. For anchors ($i \in \mathcal{A}$), this position is known; for unknown nodes ($i \in \mathcal{U}$), it must be inferred.

For each directed edge $(i, j) \in \mathcal{E}$, we observe a time series of K RSSI samples:

$$\{r_{ij}^{(k)}\}_{k=1}^{K} \tag{5.1}$$

representing signal strength measurements from node i to node j. The task is to predict the positions $\hat{\mathbf{p}}_i$ for all unknown nodes $i \in \mathcal{U}$.

5.1.3 Path-Loss Model

The relationship between RSSI and distance is governed by the log-distance path-loss model:

$$r_{ij}^{(k)} = P_t - 10n \log_{10}(d_{ij}) + o + \varepsilon_{ij}^{(k)} \tag{5.2}$$

where:

- $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|_2$ is the Euclidean distance
- P_t is the effective transmit power (dBm)
- n is the path-loss exponent (typically 2–4)
- o is an offset capturing antenna gains and other constants
- $\varepsilon_{ij}^{(k)}$ represents noise from multipath, shadowing, and measurement error

Inverting this model, the distance estimate from mean RSSI $\bar{r}_{ij}$ is:

$$\hat{d}_{ij} = 10^{(P_t + o - \bar{r}_{ij})/(10n)} \tag{5.3}$$

In terrain-rich outdoor environments, the noise term ε can be large and non-Gaussian due to diffraction, shadowing, and multipath. A single RSSI value may correspond to multiple distances, making direct inversion unreliable. This ill-posed nature of RSSI-to-distance conversion motivates our learning-based approach.

5.1.4 Challenges

Infrastructure self-localization faces several challenges beyond those of client-side localization:

Sparse Anchor Information With limited anchors (e.g., anchor-to-unknown ratio of 1:3), many unknown nodes have no direct anchor connectivity. Information must propagate through unknown-to-unknown links, but these links carry measurement noise.

Error Propagation Classical iterative methods that promote newly localized nodes to anchor status suffer from error propagation: small errors in early localizations compound through subsequent iterations, potentially causing catastrophic failure.

Terrain Effects Outdoor deployment in mountainous or urban terrain introduces severe signal attenuation that violates simple path-loss models. The RSSI-distance relationship becomes noisy and multi-modal.

No External Reference Unlike client-side localization where UAVs can use altimeters for altitude constraints, infrastructure localization has no independent position reference for unknown nodes.

5.2 Proposed Method

We propose a physics-aware, edge-conditioned Graph Convolutional Network that addresses these challenges through three key innovations: a trainable RSSI-to-distance module that provides physical priors, edge-conditioned convolutions that enable distance-aware message passing, and anchor-aware training that prevents geometric drift.

5.2.1 Architecture Overview

Figure 5.1: Architecture of the proposed edge-conditioned GCN. The RSSI time series and learned distance estimate form edge features that condition the message-passing weights.

Figure **??** illustrates the model architecture. The network takes as input the graph structure, node features, and edge features, and outputs predicted 2D coordinates for all nodes.

Node Features The initial node features are:

$$\mathbf{x}_i^{(0)} = [x_i, y_i, \text{is\_anchor}] \tag{5.4}$$

where (x_i, y_i) are the known coordinates for anchors and are initialized at the anchor centroid for unknown nodes. The binary indicator is_anchor allows the network to distinguish anchor nodes during message passing.

Edge Features The edge attributes for edge (i, j) are:

$$\mathbf{e}_{ij} = [r_{ij}^{(1)}, \ldots, r_{ij}^{(K)}, \hat{d}_{ij}] \tag{5.5}$$

consisting of the raw RSSI time series concatenated with the learned distance estimate $\hat{d}_{ij}$ from Equation **??**.

The inclusion of $\hat{d}_{ij}$ provides the network with a physically-grounded prior on distance, giving it a sense of geometric scale that pure signal features lack. Critically, $\hat{d}_{ij}$ is computed using learnable path-loss parameters, allowing the distance estimate to adapt to the propagation environment.

5.2.2 Edge-Conditioned Graph Convolution

Standard graph convolutions apply identical transformation weights to all edges, treating messages from all neighbors equally. This is problematic for localization, where the reliability and geometric significance of a neighbor's message depends critically on the edge characteristics (signal strength, estimated distance).

We employ edge-conditioned convolutions (NNConv) where a small neural network, called the EdgeNet, dynamically generates a unique weight matrix for each edge based on its attributes. This allows the network to learn edge-specific transformations that account for varying link qualities and distances.

The layer computation is:

$$\mathbf{x}_i^{(\ell+1)} = \sigma\left(\text{mean}_{j \in \mathcal{N}(i)}\left(\mathbf{W}_{ij}^{(\ell)} \mathbf{x}_j^{(\ell)}\right)\right) \tag{5.6}$$

where:

- $\mathcal{N}(i)$ denotes the neighbors of node i
- $\mathbf{W}_{ij}^{(\ell)} = \text{EdgeNet}(\mathbf{e}_{ij})$ is the edge-specific weight matrix generated from edge features
- σ is the ReLU activation function
- Mean aggregation is used to combine messages from all neighbors

EdgeNet Architecture The EdgeNet is a 2-layer MLP that maps the $(K+1)$-dimensional edge feature vector $\mathbf{e}_{ij}$ (10 RSSI samples plus 1 distance estimate) to a flattened weight matrix of size $H_{\text{in}} \times H_{\text{out}}$:

$$\text{EdgeNet} : \mathbb{R}^{K+1} \to \mathbb{R}^{H_{\text{in}} \times H_{\text{out}}} \tag{5.7}$$

Each EdgeNet contains 64 hidden units with ReLU activation. The output is reshaped into the weight matrix $\mathbf{W}_{ij}$.

This architecture enables distance-aware aggregation: the network can learn to weight messages from nearby nodes (small $\hat{d}_{ij}$, strong RSSI) differently from messages from distant or obstructed nodes (large $\hat{d}_{ij}$, weak RSSI). For instance, the network might learn to trust geometric information from close neighbors while being more skeptical of distant measurements that may be corrupted by terrain effects.

Network Structure The full network consists of:

1. Input layer: 3-dimensional node features $(x, y, \text{is\_anchor})$
2. First NNConv layer: $3 \to 64$ dimensions with ReLU activation
3. Second NNConv layer: $64 \to 64$ dimensions with ReLU activation
4. Output layer: Linear mapping from $64 \to 2$ (predicted x, y coordinates)

Each NNConv layer has its own EdgeNet with the same architecture but separate learned parameters.

5.2.3 Trainable Path-Loss Module

Rather than using fixed path-loss parameters from literature or manual calibration, we learn the global parameters n (path-loss exponent) and o (offset) as trainable scalar parameters optimized jointly with the GCN weights. The transmit power P_t is fixed at a known value (13 dBm in our experiments), as it is typically specified by the radio hardware.

The distance estimate from mean RSSI $\bar{r}_{ij}$ is computed as:

$$\hat{d}_{ij} = 10^{(P_t + o - \bar{r}_{ij})/(10n)} \tag{5.8}$$

Numerical Stability Constraints To ensure numerical stability during training, we apply several constraints that prevent invalid or extreme distance estimates:

- **Denominator clamping:** The term $10n$ is clamped to $|10n| \geq \epsilon_n$ to prevent division by values near zero when the path-loss exponent is small
- **Exponent clamping:** The exponent $(P_t + o - \bar{r}_{ij})/(10n)$ is clamped to the range $[-2, 5]$, corresponding to plausible distances from 0.01 m to 100 km given typical RSSI ranges
- **Distance lower bound:** The final distance estimate $\hat{d}_{ij}$ is lower-bounded by a small $\epsilon > 0$ to prevent zero or negative distances

These constraints are essential because outlier RSSI values (from severe multipath or interference) or transient parameter states during early training can produce invalid distance estimates that destabilize optimization. The constraint ranges are chosen to be permissive enough not to affect valid links within the 4 km evaluation area while preventing numerical issues.

Benefits of Learned Parameters Learning path-loss parameters jointly with the GCN provides two key benefits:

1. **Environment Adaptation:** The model automatically adapts to the propagation characteristics of the training environment, learning appropriate values of n and o for the terrain and conditions
2. **Interpretability:** After training, the learned parameters can be examined to understand the effective propagation model, providing insight into the environment (e.g., higher n values indicate more severe path loss due to terrain)

5.2.4 Anchor Handling

Anchor nodes serve as fixed reference points that prevent the solution from drifting or scaling arbitrarily. We implement anchor handling through three mechanisms:

Fixed Coordinates Anchor coordinates are treated as ground truth throughout training and inference. While messages are passed through all nodes (including anchors), the predicted coordinates for anchors are overwritten with their known values after each forward pass.

Loss Exclusion Anchors are excluded from the loss calculation, ensuring gradients flow only from unknown node predictions.

Message Passing Participation Despite having fixed outputs, anchors participate fully in message passing. This allows anchor position information to propagate to neighboring unknown nodes, providing geometric constraints throughout the graph.

5.2.5 Training Objective

The network is trained to minimize the Smooth L1 (Huber) loss between predicted and true coordinates for unknown nodes:

$$\mathcal{L} = \sum_{i \in \mathcal{U}} \rho(\hat{\mathbf{p}}_i - \mathbf{p}_i) \tag{5.9}$$

where ρ is the Smooth L1 function:

$$\rho(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{5.10}$$

The Smooth L1 loss combines the benefits of L1 and L2 losses: it is less sensitive to outliers than L2 (preventing large errors from dominating gradients) while being smooth near zero (enabling stable optimization).

5.3 Evaluation Setup

We evaluate the proposed method through comprehensive simulations using a terrain-aware propagation model that captures the challenging conditions of real outdoor deployments.

5.3.1 Dataset and Environment

Geographic Area We simulate a 4 km × 4 km outdoor area in Çorum Province, Turkey. This region is selected because public reports indicate frequent GPS interference in the area, motivating the need for GNSS-independent localization solutions.

Network Configuration Each simulated network instance contains 64 nodes:

- 16 anchor nodes (25%) placed in a grid-constrained layout providing good geometric coverage
- 48 unknown nodes (75%) sampled uniformly at random within the area
- All nodes at height 1.0 m above ground

RSSI Measurements For each directed pair of nodes, we simulate $K = 10$ RSSI samples to capture signal variability. The limited sample budget reflects practical constraints on measurement time in rapid deployment scenarios.

Dataset Split We generate 1000 unique graph instances using a MATLAB-based data generator. The dataset is split into 800 training graphs and 200 held-out test graphs. All metrics are computed on the test set.

Table **??** summarizes the core experimental parameters.

Table 5.1: Core experimental parameters

Parameter	**Value**
Deployment area	4 km × 4 km
Propagation model	Longley-Rice (terrain) / Free-space
Carrier frequency	915 MHz
Total nodes	64
Anchor nodes	16 (25%)
Unknown nodes	48 (75%)
Node height	1.0 m above ground
RSSI samples per link (K)	10
Transmit power (P_t)	13 dBm
Coordinate frame	Local Cartesian
Origin	40.466198°N, 33.898610°E

5.3.2 Propagation Regimes

We evaluate under two propagation regimes to understand how environmental realism affects method performance:

Terrain-Aware Regime Our primary evaluation setting uses the Longley-Rice Irregular Terrain Model, which accounts for:

- Path loss due to distance and frequency
- Diffraction over terrain obstacles based on actual elevation data
- Tropospheric scatter effects
- Surface reflections based on ground conductivity

Figure **??** shows the evaluation area. The mountainous terrain introduces significant signal attenuation and shadowing that challenges localization algorithms.

Free-Space Regime For comparison, we also evaluate under idealized free-space propagation with no terrain effects. This regime reveals how environmental factors affect the relative performance of different methods.

Figure **??** illustrates the stark difference between regimes. In terrain-aware propagation, a single RSSI value can correspond to multiple distances due to terrain effects, making direct RSSI-to-distance conversion unreliable.

Figure 5.2: Evaluation area: (a) Satellite view showing terrain complexity; (b) Elevation map with 4 km × 4 km extent.

5.3.3 Baseline Methods

We compare against two baselines representing classical geometric and ablated deep learning approaches. These baselines are chosen to isolate the contributions of different components of our method.

Iterative Multilateration

This classical geometric method represents the standard approach to network localization without machine learning. It follows a two-stage iterative process that alternates between path-loss calibration and position estimation, repeated for 10 iterations:

1. **Path-Loss Estimation:** A simple log-distance model of the form $\text{RSSI}(d) = A - 10n \log_{10}(d)$ is fit to anchor-anchor links, where true distances are known from anchor coordinates. Least-squares regression determines the optimal parameters (A, n) for the current graph instance. This per-graph calibration allows the model to adapt to varying propagation conditions across different instances.

Figure 5.3: RSSI versus distance relationship: (a) Terrain-aware regime showing high variance and non-linearity; (b) Free-space regime showing clean log-distance relationship.

2. **Distance Conversion:** The fitted path-loss model is applied to convert the mean RSSI of all other links (anchor-unknown and unknown-unknown) to distance estimates $\hat{d}$.

3. **Position Solving:** For each unknown node that has at least 3 neighbors with known positions (anchors or previously localized nodes), the 2D coordinates are estimated using bounded nonlinear least-squares optimization. The objective minimizes the sum of squared differences between estimated distances and geometric distances implied by the candidate position. Bounds constrain solutions to lie within the deployment area.

4. **Anchor Promotion:** Nodes that were successfully localized with residual error below a threshold are temporarily promoted to anchor status. Their estimated positions are used as references in subsequent iterations, extending the reach of localization to nodes that lack direct anchor connectivity.

This method requires no training data and can be applied directly to new environments. However, anchor promotion is vulnerable to error propagation: if an early iteration produces an inaccurate position estimate, that error propagates

to dependent nodes in later iterations, potentially causing cascading failures.

Plain GCN (Ablation)

This ablation study isolates the contribution of the physics-aware RSSI-to-distance module by removing it from the proposed architecture. The plain GCN uses:

- **Identical architecture:** Same number of NNConv layers, hidden dimensions, and EdgeNet structure
- **Identical training procedure:** Same optimizer, learning rate, epochs, and loss function
- **Modified edge features:** Only the raw RSSI time series $[r_{ij}^{(1)}, \ldots, r_{ij}^{(K)}]$, without the learned distance estimate $\hat{d}_{ij}$

By comparing the proposed method against this ablation, we can quantify the benefit of incorporating physics-aware distance estimation into the edge features. The plain GCN must learn the relationship between raw RSSI values and geometric distances entirely from data, without the inductive bias provided by the path-loss model.

5.3.4 Training Details

All GCN models are implemented in PyTorch Geometric. Table **??** summarizes the training hyperparameters.

Table 5.2: GCN training hyperparameters

Hyperparameter	**Value**
Optimizer	Adam
Learning rate	10^{-4}
Weight decay	10^{-5}
Epochs	50
Batch size	1
Hidden dimension	64
Convolution layers	2 $\times$ NNConv (mean aggregation)
Loss function	Smooth L1 on unknown nodes
Gradient clipping	1.0

All node features (initial coordinates) and target coordinates are standardized using scalers fit on the training data. This normalization ensures that the network operates on zero-mean, unit-variance inputs, improving optimization stability. During inference, predictions are inverse-transformed using the same scalers to recover coordinates in the original metric space.

Gradient clipping at norm 1.0 prevents exploding gradients that can occur when the path-loss module produces extreme distance estimates during early training. Weight decay is applied to both GCN parameters and the path-loss module parameters to prevent overfitting.

5.4 Results

5.4.1 Terrain-Aware Results

Under the primary terrain-aware evaluation setting, the proposed method demonstrates substantial improvement over baselines. Table **??** summarizes the localization error statistics.

Table 5.3: Terrain-aware localization error (meters)

Method	Mean	Median	P90	P95
Proposed GCN	**206.74**	**182.42**	**381.01**	**455.37**
Plain GCN	750.87	662.12	1330.32	1571.84
Iter. Multilateration	1974.40	1747.99	3581.93	4489.04

The proposed method achieves mean error of 206.74 m, representing:

- 72% reduction compared to the plain GCN (750.87 m)
- 90% reduction compared to iterative multilateration (1974.40 m)

Figures **??** and **??** show the error distributions. The proposed method achieves a clear left-shift in the CDF, indicating consistently lower errors across the entire distribution. Critically, the method also dramatically reduces the tail of large errors: the 90th percentile (P90) error is 381 m compared to over 1330 m and 3580 m for the baselines.

Figures **??**, **??**, and **??** provide qualitative comparison of all three methods. The proposed method produces localization results that closely track true positions with small, consistent errors. The plain GCN ablation shows larger and more variable errors, demonstrating the importance of the physics-aware distance features. Iterative multilateration exhibits catastrophic errors scattered throughout the network, illustrating the vulnerability of anchor promotion to error propagation in terrain-rich environments.

5.4.2 Free-Space Results

To understand the impact of propagation environment, we repeat the evaluation under idealized free-space conditions. Table **??** shows a striking inversion of performance rankings.

Table 5.4: Free-space localization error (meters)

Method	Mean	Median	P90	P95
Proposed GCN	51.15	51.53	68.92	74.24
Plain GCN	257.80	210.44	494.98	632.21
Iter. Multilateration	**6.28**	**5.80**	**10.87**	**12.65**

In free-space conditions, iterative multilateration achieves mean error of only 6.28 m, substantially outperforming both GCN methods. The proposed GCN achieves 51.15 m, while the plain GCN still struggles at 257.80 m.

This result has important implications:

Figure 5.4: Cumulative distribution function of localization error in terrain-aware setting. The proposed method shows consistently lower errors across the entire distribution.

Model Specification When the geometric model is correctly specified (free-space path loss matches reality), direct optimization outperforms learning. The advantage of our method emerges precisely when model-reality mismatch is severe.

Evaluation Realism Free-space evaluation can produce misleading performance rankings. Methods that excel in idealized conditions may fail dramatically in realistic terrain-aware settings, and vice versa. This underscores the necessity of terrain-aware evaluation for outdoor localization.

Figure 5.5: Histogram of localization error in terrain-aware setting. The proposed method eliminates the heavy tail of large errors present in baselines.

Figure 5.6: Qualitative localization result (terrain-aware): proposed GCN. Triangles indicate anchors; circles indicate unknown nodes. Lines connect predicted (hollow) to true (filled) positions.

Figure 5.7: Qualitative localization result (terrain-aware): plain GCN ablation (without learned distance). Errors are larger and more scattered than the proposed method.

Figure 5.8: Qualitative localization result (terrain-aware): iterative multilateration. Catastrophic errors are visible due to error propagation through anchor promotion.

Figure 5.9: CDF of localization error in free-space setting. Iterative multilateration achieves best performance when the path-loss model is perfectly specified.

5.5 Discussion

5.5.1 Performance Drivers

The substantial performance gap between methods originates from how they handle the ill-posed RSSI-to-distance conversion in terrain-rich environments.

Why Multilateration Fails As shown in Figure **??**, terrain effects create a noisy, multi-modal RSSI distribution where a single RSSI value can correspond to multiple distances. Iterative multilateration commits to hard distance estimates for each link, producing a geometrically inconsistent constraint system. Early localization errors in promoted nodes compound through subsequent iterations, explaining the catastrophic outliers in its error distribution.

Why the Proposed Method Succeeds Our method succeeds by reframing the problem. Instead of committing to hard distance conversion, the trainable path-loss module generates distance estimates that serve as soft, physically-grounded priors. The edge-conditioned convolution learns to weigh this prior against the raw RSSI time series on a per-edge basis, effectively learning when to trust the distance estimate and when to distrust it.

Furthermore, the GCN architecture provides powerful regularization through message passing. Whereas multilateration computes each node's position independently, the GCN performs collective, simultaneous inference across the entire graph. Each node's position is constrained by its neighbors, smoothing predictions and preventing isolated large errors. This graph-based reasoning is responsible for the dramatic reduction in the error tail.

Role of Physics-Aware Features The comparison with plain GCN (72% error reduction) demonstrates the importance of the learned distance estimate. Without physical priors, the network must learn the relationship between raw RSSI values and distances purely from data, which is difficult given the complex, terrain-dependent mapping. The physics-aware module provides a structured inductive bias that accelerates learning and improves generalization.

5.5.2 Limitations

While results are promising, several limitations present opportunities for future work:

Absolute Accuracy The proposed method does not achieve GNSS-level accuracy. Mean error of 207 m is sufficient for many applications (regional navigation, search area narrowing) but insufficient for precision tasks. Future work should investigate the trade-off between beacon density and achievable accuracy.

Anchor Geometry Our simulations use grid-constrained anchor placement providing good geometric coverage. Real deployments may be constrained by terrain accessibility, potentially resulting in suboptimal anchor geometry (e.g., all anchors on periphery). Robustness to poor anchor placement requires further study.

Simulation-to-Reality Gap Both training and evaluation use simulated data. While Longley-Rice provides realistic terrain effects, it does not capture all real-world complexities: device heterogeneity, antenna placement variations, temporal environmental changes, and interference. Bridging this gap through domain adaptation or field experiments is a critical next step.

Global Path-Loss Parameters The model learns global path-loss parameters applied uniformly across the network. Real deployments may exhibit regional variations in propagation characteristics. Spatially-aware or per-region parameterization could improve accuracy in heterogeneous environments.

5.5.3 Deployment Considerations

Based on our results and analysis, we outline a practical deployment workflow:

Pre-Deployment Training Before physical deployment, obtain terrain data (digital elevation model) for the target area and generate a synthetic training dataset using terrain-aware simulation (e.g., Longley-Rice). Pre-train the GCN to create a terrain-matched prior that captures the expected propagation characteristics.

Post-Installation Calibration After deploying beacons, collect anchor-to-anchor RSSI measurements where true distances are known. Perform lightweight calibration by fine-tuning only the path-loss module parameters (n, o) while keeping GCN weights frozen. This aligns the physical model with actual hardware and local conditions without requiring full retraining.

Inference With anchors fixed and the model calibrated, run inference to localize unknown nodes. The hub aggregates RSSI measurements from the self-organization phase, constructs the network graph, and applies the trained model to predict beacon positions.

5.6 Summary

This chapter has presented a physics-aware Graph Convolutional Network for infrastructure self-localization in sparse outdoor LPWA networks. The key contributions are:

1. **Trainable RSSI-to-Distance Module:** A physically-interpretable component that learns path-loss parameters jointly with the graph network, providing distance-aware edge features that adapt to the propagation environment.

2. **Edge-Conditioned Architecture:** NNConv layers that generate per-edge weight matrices from edge features, enabling distance-aware message passing that weighs nearby and distant neighbors appropriately.

3. **Anchor-Aware Training:** A scheme that fixes anchor positions throughout training and inference, leveraging known coordinates to prevent geometric drift while allowing information to flow through the entire graph.

4. **Terrain-Aware Evaluation:** Demonstration that evaluation regime dramatically affects method rankings, with our approach achieving 90% error reduction over multilateration in realistic conditions while multilateration wins in idealized free-space.

The proposed method achieves mean localization error of 206.74 m in terrain-aware simulations with 64 nodes (16 anchors, 48 unknowns) over a 4 km $\times$ 4 km area. This represents substantial improvement over both classical multilateration and plain GCN approaches, establishing that physics-grounded graph learning provides a viable foundation for infrastructure localization when GNSS is unavailable.

Combined with the UAV localization method of Chapter **??**, this completes the core algorithmic contributions of the thesis. The following chapter analyzes the integrated system performance and navigation feasibility.

Chapter 6

Integrated System Analysis and Navigation Feasibility

The preceding chapters have addressed the two core algorithmic problems independently: Chapter **??** developed robust UAV localization given known beacon positions, while Chapter **??** developed infrastructure self-localization using graph learning. This chapter integrates these components to assess the feasibility of the complete system, analyzing how infrastructure errors propagate to client-side localization and evaluating navigation performance under realistic conditions.

The central question motivating this analysis is: *Given that the physics-aware GCN achieves mean localization error of approximately 206 m for beacon self-localization, can the resulting beacon network still support useful UAV navigation?* We address this question through sensitivity analysis, closed-loop navigation simulations, and evaluation across multiple mission profiles.

6.1 Simulation Framework

To evaluate integrated system performance, we construct a simulation environment that combines the RSSI-based localization algorithms from Chapter **??** with beacon position uncertainty representative of the self-localization errors from Chapter **??**.

6.1.1 Environment Setup

The simulation uses RSSI data generated from a 4 km $\times$ 4 km deployment area with 64 beacons. The beacons are positioned according to the same distribution used in Chapter **??**, providing consistent geometry between infrastructure and client-side evaluations.

RSSI Model A grid-based RSSI interpolation model is constructed from simulated measurements, enabling evaluation of UAV localization at any position within the deployment area. The UAV operates at a fixed altitude of 100 m, and the multilateration algorithm from Chapter **??** (Huber loss with altitude constraint) is applied using the $k = 10$ nearest beacons.

Beacon Position Uncertainty To model the effect of infrastructure self-localization errors, we add Gaussian noise to beacon coordinates. The noise is characterized by mean 2D error rather than per-axis standard deviation, providing direct comparability with localization error metrics. For a target mean

2D error of μ, the per-axis standard deviation is $\sigma = \mu/\sqrt{\pi/2} \approx 0.798\mu$, derived from the Rayleigh distribution of 2D distances from independent Gaussian components.

6.1.2 Evaluation Metrics

We evaluate system performance using:

- **Localization Error:** Euclidean distance between estimated and true UAV positions, reported as mean, median, and 90th percentile (P90)
- **Waypoint Success Rate:** Percentage of waypoints successfully reached during navigation missions
- **Heading Error:** Angular deviation between the commanded heading (based on estimated position) and the true heading to target

6.2 Sensitivity Analysis

We first analyze how beacon position errors propagate to UAV localization accuracy, establishing the relationship between infrastructure quality and client-side performance.

6.2.1 Grid-Wide Localization Analysis

To obtain statistically robust results, we sample 2,000 random positions within the deployment area and compute localization error at each position for varying levels of beacon position uncertainty. This grid-wide analysis captures the full distribution of errors across the operational area, avoiding bias from specific flight paths.

Table **??** presents the localization error statistics for beacon position errors ranging from 0 m (perfect knowledge) to 300 m (beyond expected GCN performance).

Table 6.1: UAV localization error versus beacon position error

Beacon Error (m)	**Mean (m)**	**Median (m)**	**P90 (m)**
0 (baseline)	122	115	182
50	119	110	193
100	122	110	210
150	125	112	220
200	157	134	287
250	181	156	341
300	207	189	372

Figure **??** visualizes the relationship between beacon and UAV errors. Several observations emerge:

Sub-Linear Degradation UAV localization error increases sub-linearly with beacon position error. At 200 m beacon error (approximately matching GCN performance), UAV error increases from 122 m to 157 m, a degradation of only 29%. This sub-linear relationship indicates that the multilateration algorithm provides inherent robustness to beacon position uncertainty through geometric averaging across multiple beacons.

Figure 6.1: UAV localization error as a function of beacon position error. The vertical dashed line indicates the GCN self-localization error (206 m). Error bars show standard deviation.

Graceful Degradation The system does not exhibit catastrophic failure modes. Even at 300 m beacon error, significantly worse than expected from the GCN, UAV localization remains functional with mean error of 207 m. This graceful degradation is essential for operational reliability.

Tail Behavior Figure **??** shows the 90th percentile (P90) error, representing worst-case performance. At 200 m beacon error, P90 reaches 287 m, indicating that 90% of positions achieve better than this accuracy. The tail grows faster than the mean, suggesting that beacon errors more severely impact geometrically unfavorable positions.

6.2.2 Implications for GCN-Based Infrastructure

The physics-aware GCN from Chapter **??** achieves mean beacon localization error of 206 m. Interpolating from Table **??**, this corresponds to expected UAV localization error of approximately 160 m mean and 290 m P90.

While this accuracy falls short of GNSS standards (typically 2–5 m), it remains useful for many operational scenarios. The following sections evaluate whether this accuracy supports practical navigation missions.

Figure 6.2: 90th percentile UAV localization error versus beacon position error, representing worst-case performance across the operational area.

6.3 Navigation Simulations

To validate the theoretical stability analysis, we conduct closed-loop navigation simulations across three mission profiles representative of common UAV operations.

6.3.1 Simulation Setup

Control Model The UAV is modeled with simple proportional control: at each time step, it computes the heading to the current waypoint based on its estimated position and adjusts velocity accordingly. The UAV flies at 15 m/s with position updates at 1 Hz. This simplified model captures the essential feedback dynamics without unnecessary complexity.

Waypoint Threshold Waypoints are considered reached when the UAV's true position is within 200 m, consistent with the stability analysis. This threshold prevents the UAV from circling indefinitely near waypoints due to position uncertainty.

Mission Profiles We evaluate three mission profiles:

- **Diagonal Transit:** Point-to-point navigation across 4 km, representing long-range transit missions
- **Circular Patrol:** Circular path with 1 km radius, representing perimeter surveillance

- **Area Sweep:** Systematic coverage pattern, representing search-and-rescue or mapping missions

6.3.2 Diagonal Transit Results

The diagonal transit mission tests long-range navigation capability, flying from coordinates (600, 600) to (3400, 3400) meters, a distance of approximately 4 km.

Figure 6.3: Diagonal transit with perfect beacon positions. The true path (blue) closely follows the direct route while the estimated path (red dashed) shows localization noise.

Figures **??** and **??** compare flight paths with perfect and degraded beacon positions. Figure **??** provides striking visual confirmation of the geometric stability analysis: during the long transit phase, the true path (blue line) remains nearly straight despite the noisy estimated path (red dashed line). However, near the destination waypoint, the path shows increased deviation as the geometric stability breaks down at close range.

Figure **??** shows the localization error throughout the flight, while Figure **??** shows the error distribution. The mission achieves 100% waypoint success with mean localization error of 152 m.

6.3.3 Circular Patrol Results

The circular patrol mission tests continuous navigation around a 1 km radius circle centered at (2000, 2000) meters, with 8 waypoints spaced evenly around the perimeter.

Figure 6.4: Diagonal transit with 200 m beacon position error. The true path remains stable during the long transit but shows increased deviation near the destination, illustrating the geometric stability principle.

Figures **??** and **??** show the patrol paths. The circular pattern provides consistently moderate distances between waypoints (approximately 780 m), keeping the UAV in the stable navigation regime throughout the mission. With 200 m beacon error, all 8 waypoints are reached with mean localization error of 137 m.

6.3.4 Area Sweep Results

The area sweep mission tests systematic coverage, following a zigzag pattern across a 2.8 km $\times$ 2.8 km region with 5 rows. This profile is representative of search-and-rescue operations where complete area coverage is essential.

Figures **??** and **??** show the coverage patterns. With 200 m beacon error, the UAV successfully completes all waypoints with mean localization error of 145 m. The zigzag pattern is maintained, though with increased deviation from the planned tracks.

Coverage Implications For area coverage missions, the track deviation caused by beacon errors can be compensated by planning overlapping flight paths. With expected lateral deviation of approximately 150 m at 200 m beacon error, planning overlapping tracks (e.g., 100 m spacing for a 200 m sensor swath, yielding 50% overlap) ensures complete ground coverage despite navigation uncertainty. This strategy trades flight efficiency for guaranteed coverage.

Figure 6.5: Localization error during diagonal transit with 200 m beacon error. Error varies along the path but remains bounded.

6.3.5 Comparative Analysis

Table **??** summarizes navigation performance across all mission profiles and beacon error levels.

Table 6.2: Navigation simulation results

Mission	**Beacon Error**	**Success**	**Mean Error**
Diagonal	0 m	100%	112 m
	150 m	100%	144 m
	200 m	100%	152 m
Circular	0 m	100%	124 m
	150 m	100%	118 m
	200 m	100%	137 m
Area Sweep	0 m	100%	115 m
	150 m	100%	127 m
	200 m	100%	145 m

All three mission profiles achieve 100% waypoint success at beacon error levels up to 200 m, corresponding to expected GCN self-localization performance. This demonstrates that despite the substantial beacon position uncertainty in-

Figure 6.6: Distribution of localization errors during diagonal transit with 200 m beacon error.

troduced by RSSI-only self-localization, the integrated system supports practical UAV navigation.

Figure 6.7: Circular patrol with perfect beacon positions. The UAV successfully navigates around all waypoints.

Figure 6.8: Circular patrol with 200 m beacon position error. Despite increased path deviation, the UAV completes all waypoints.

Figure 6.9: Localization error during circular patrol with 200 m beacon error.

Figure 6.10: Distribution of localization errors during circular patrol with 200 m beacon error.

Figure 6.11: Area sweep with perfect beacon positions. The UAV follows the planned zigzag pattern.

Figure 6.12: Area sweep with 200 m beacon position error. The pattern is maintained with increased track deviation.

Figure 6.13: Localization error during area sweep with 200 m beacon error.

Figure 6.14: Distribution of localization errors during area sweep with 200 m beacon error.

6.4 Discussion

6.4.1 Geometric Stability

A key insight enabling navigation despite high localization error is *geometric stability*: the relationship between position error and navigation error depends strongly on distance to target.

Consider a UAV at true position $\mathbf{p}_{\text{true}}$ with estimated position $\mathbf{p}_{\text{est}}$ attempting to navigate toward target $\mathbf{t}$. The worst-case heading error $\Delta\theta$ occurs when the position error is perpendicular to the true heading direction, and depends on both the position error magnitude ϵ and the distance to target d:

$$\Delta\theta_{\max} \approx \arctan\left(\frac{\epsilon}{d}\right) \tag{6.1}$$

When the position error is parallel to the heading direction (toward or away from the target), the heading error approaches zero.

Figure **??** illustrates this worst-case relationship for various position error magnitudes. Heading error decreases rapidly with distance: at 1 km distance, even 200 m position error produces at most 11 degrees of heading error. This explains why the diagonal transit in Figure **??** maintains a stable, nearly straight path during the long transit phase, but shows increased deviation near the destination where the geometric stability breaks down.

Figure 6.15: Geometric stability: worst-case heading error (position error perpendicular to heading) decreases rapidly with distance to target. At 1 km distance, 200 m position error produces at most 11 degrees of heading error.

This analysis motivates the 200 m waypoint threshold used in the simulations: by declaring arrival before entering the unstable close-approach regime, the system avoids erratic circling behavior while maintaining mission reliability.

6.4.2 Operational Envelope

The analysis establishes the operational envelope for the integrated system:

Supported Operations The system reliably supports:

- Long-range transit between waypoints separated by more than 1 km
- Patrol missions with moderate waypoint spacing (500–1000 m)
- Area coverage with planned overlap to compensate for navigation uncertainty
- Search operations where approximate position is sufficient

Unsupported Operations The system is unsuitable for:

- Precision landing requiring meter-level accuracy
- Dense waypoint sequences with spacing below 300 m
- Obstacle avoidance requiring precise position knowledge
- Formation flight requiring relative positioning accuracy

6.4.3 Design Trade-offs

The 200 m waypoint threshold represents a fundamental trade-off between mission completion reliability and waypoint accuracy. Alternative threshold values would shift this trade-off:

- **Smaller threshold (100 m):** Higher accuracy when successful, but increased risk of failure due to instability in the close-approach regime
- **Larger threshold (300 m):** More reliable completion, but reduced effective accuracy and potential for missing narrow targets

The 200 m value is chosen to provide reliable operation at the expected GCN error level (206 m) while maintaining useful accuracy for most operational scenarios.

6.4.4 Comparison with GNSS

Table **??** compares the proposed system with GNSS-based navigation.

Table 6.3: Comparison with GNSS navigation

Characteristic	**GNSS**	**Proposed**
Position accuracy	2–5 m	120–160 m
Global availability	Yes	No (local)
Jamming resistance	Low	High
Infrastructure required	Satellites	Beacons
Deployment time	N/A	Hours
Cost per coverage area	Low	Moderate

The proposed system does not compete with GNSS for precision navigation. Instead, it provides a complementary capability for GNSS-denied environments,

enabling continued operations when primary navigation fails. The key advantage is resilience: while GNSS can be denied by jamming or environmental obstruction, the proposed system operates independently of external infrastructure.

6.4.5 Limitations

Several limitations should be acknowledged:

Simulation Fidelity The navigation simulations use simplified dynamics and idealized sensor models. Real-world flight would introduce additional factors including wind, control latency, and sensor noise beyond RSSI variance.

Static Environment The analysis assumes a static environment with fixed beacon positions. In practice, beacon positions might drift due to equipment settling or environmental changes, requiring periodic recalibration.

Single Altitude All simulations assume fixed UAV altitude of 100 m. Varying altitude would affect RSSI measurements due to antenna directivity effects noted in Chapter **??**.

6.5 Summary

This chapter has analyzed the integrated performance of the proposed positioning system, combining infrastructure self-localization (Chapter **??**) with UAV multilateration (Chapter **??**).

The key findings are:

1. **Sub-Linear Error Propagation:** Beacon position errors propagate sublinearly to UAV localization errors. At 200 m beacon error (matching expected GCN performance), UAV error increases by only 29% from baseline, demonstrating inherent robustness.

2. **Geometric Stability:** Navigation remains stable when targets are distant. For targets beyond 1 km, position errors up to 200 m produce heading errors below 11 degrees, enabling reliable waypoint navigation.

3. **Mission Feasibility:** Closed-loop simulations demonstrate 100% waypoint success across diagonal transit, circular patrol, and area sweep missions at beacon error levels up to 200 m.

4. **Operational Trade-offs:** A 200 m waypoint threshold balances reliability against accuracy, and area coverage missions can compensate for track deviation through planned overlap.

These results establish that despite localization errors substantially larger than GNSS, the integrated system provides operationally useful positioning for emergency and backup navigation scenarios. The system enables mission continuation when primary navigation fails, fulfilling its role as a resilient alternative for GNSS-denied environments.

Chapter 7

Conclusion

This thesis has investigated the feasibility of using Low-Power Wide-Area (LPWA) networks, specifically LoRa, as a standalone positioning infrastructure for GNSS-denied environments. We addressed two fundamental challenges: client-side UAV localization given beacon positions, and infrastructure self-localization when beacon positions are unknown. This chapter summarizes the contributions, discusses limitations, and outlines directions for future work.

7.1 Summary of Contributions

The thesis makes three primary contributions to the field of GNSS-independent positioning:

Contribution 1: Robust Client-Side Localization Chapter **??** presented an enhanced multilateration algorithm that combines the Huber loss function with altitude constraints. The Huber loss suppresses the influence of RSSI outliers that arise from multipath propagation and environmental shadowing, while altitude constraints from barometric or laser altimeters decouple vertical estimation from horizontal positioning. This combination achieves:

- Median localization error of 4.5 m in static conditions (85% improvement over conventional least-squares)
- Median error of 63 m during dynamic flight at 10 m/s (25% improvement)
- Monotonic improvement with additional beacons, unlike squared-loss methods that degrade
- Strong correlation between minimization error and localization error, enabling detection of unreliable estimates

Field experiments with real hardware validated these improvements, achieving 7 m median error in favorable conditions.

Contribution 2: Physics-Aware Infrastructure Self-Localization Chapter **??** developed a Graph Convolutional Network (GCN) architecture that embeds radio propagation physics directly into the learning process. The key innovation is a trainable RSSI-to-distance module that learns path-loss parameters jointly with the graph message-passing weights. This physics-aware design enables the network to adapt to local propagation characteristics while maintaining interpretability. In terrain-aware simulations using the Longley-Rice propagation model:

- The proposed method achieves mean localization error of 206 m
- This represents 90% improvement over iterative multilateration (1974 m)
- And 72% improvement over a plain GCN without the physics module (750 m)
- The method demonstrates robustness to the severe signal attenuation caused by mountainous terrain

Contribution 3: Integrated System Feasibility Analysis Chapter **??** analyzed the complete system from infrastructure self-organization through UAV mission execution. The key finding is that geometric stability enables useful navigation despite substantial localization errors:

- Beacon position errors propagate sub-linearly to UAV localization (29% degradation at 200 m beacon error)
- For targets beyond 1 km, position errors up to 200 m produce heading errors below 11 degrees
- Closed-loop simulations demonstrate 100% waypoint success across diverse mission profiles
- Area coverage missions can compensate for navigation uncertainty through planned overlap

Overall Achievement Together, these contributions demonstrate that while LoRa-based positioning cannot match GNSS precision, it provides a viable backup system for emergency operations. The integrated architecture builds positioning capability from scratch, requiring only commodity LoRa hardware and predeployment terrain simulation for model training. This enables autonomous missions to continue when primary navigation fails due to jamming, spoofing, or environmental obstruction.

7.2 Limitations

Several limitations of this work should be acknowledged:

Simulation-Based Evaluation The infrastructure self-localization (Chapter **??**) and integrated system analysis (Chapter **??**) rely on simulated data. While the Longley-Rice propagation model provides realistic terrain effects, it does not capture all real-world complexities including device heterogeneity, temporal environmental changes, and interference from other radio sources. The gap between simulation and reality remains to be quantified through field validation.

Accuracy Limitations The achieved localization accuracy, approximately 120–160 m for UAVs with GCN-estimated beacon positions, falls far short of GNSS standards (2–5 m). This limits applicability to scenarios where approximate positioning is sufficient, excluding precision tasks such as landing, obstacle avoidance, and formation flight.

Anchor Requirements The infrastructure self-localization assumes 25% of nodes have known positions (anchors). In truly unknown environments, establishing even this anchor set may be challenging. The sensitivity of localization accuracy to anchor quantity and geometry was not comprehensively studied.

Static Environment Assumption Both the beacon network and the propagation environment are assumed static. In practice, beacon positions might drift, and environmental changes (vegetation growth, construction, weather) could alter propagation characteristics over time.

Single Altitude Operation All UAV simulations assume fixed altitude of 100 m. The antenna directivity effects observed in Chapter **??** suggest that varying altitude would require altitude-aware RSSI models not fully developed in this work.

7.3 Future Work

Several directions for future research emerge from this work:

Field Validation The most critical next step is field validation of the complete system. This would involve deploying a beacon network in a GNSS-denied or GNSS-challenged environment, collecting real RSSI measurements, and evaluating both infrastructure self-localization and UAV navigation accuracy. Such validation would quantify the simulation-to-reality gap and identify practical deployment challenges.

Simulation-to-Real Transfer The physics-aware GCN is pre-trained on synthetic data generated from terrain simulations. Fine-tuning the model using real RSSI measurements collected during deployment could improve accuracy by adapting to actual propagation conditions. Techniques from domain adaptation and transfer learning could facilitate this simulation-to-real transfer with limited real data.

Dynamic Beacon Networks Extending the system to handle dynamic beacon networks, where beacons may be added, removed, or relocated, would increase operational flexibility. This could involve online learning approaches that incrementally update position estimates as the network evolves.

Multi-UAV Coordination The current system treats each UAV independently. In multi-UAV operations, vehicles could share position estimates or RSSI observations to improve collective localization accuracy. Cooperative localization algorithms could exploit inter-UAV measurements when vehicles are within communication range.

Hybrid Systems Integration with other positioning modalities could improve accuracy and reliability. Candidate technologies include:

- Inertial Navigation Systems (INS) for short-term position propagation between RSSI updates
- Visual odometry for relative motion estimation in feature-rich environments

- Ultra-Wideband (UWB) ranging for high-accuracy local positioning near critical waypoints

Such hybrid systems could leverage LoRa for coarse global positioning while using complementary technologies for local refinement.

Altitude-Aware Models The antenna directivity effects observed in field experiments motivate development of altitude-aware RSSI-to-distance models. Such models could explicitly account for the antenna radiation pattern, improving accuracy for UAVs operating at varying altitudes.

7.4 Concluding Remarks

The increasing reliance on GNSS for autonomous systems creates a critical vulnerability that this thesis has sought to address. By demonstrating that LPWA networks can provide emergency positioning capability, we offer a path toward more resilient autonomous operations.

The key insight from this work is that useful navigation does not require precise localization. Through geometric stability, a UAV can navigate effectively toward distant targets even with position errors of hundreds of meters. This observation shifts the design target from achieving high accuracy to ensuring reliable, bounded errors that support mission completion.

While the proposed system cannot replace GNSS for precision applications, it fills an important gap: providing backup positioning that enables mission continuation when primary navigation fails. For disaster response in areas with damaged infrastructure, for scientific expeditions in remote regions, and for operations beyond terrestrial GNSS coverage, such backup capability can make the difference between mission success and failure.

We conclude that LoRa-based positioning, while imprecise, represents a practical and cost-effective approach to GNSS-denied navigation. The combination of self-organizing infrastructure and robust client-side algorithms creates a system that can be rapidly deployed and operated with minimal prior knowledge of the environment. This capability advances the resilience of autonomous systems in an era of increasing GNSS vulnerability.

References

[1] K.-J. Baik, S. Lee, and B.-J. Jang. Hybrid RSSI-AoA Positioning System with Single Time-Modulated Array Receiver for LoRa IoT. In *Proc. EuMC*, pages 1133–1136, Madrid, Spain, 2018.

[2] H. Durrant-Whyte and T. Bailey. *Simultaneous Localization and Mapping: Part I*, volume 13. 2006.

[3] S. Fan and J. Yan. CSI Fingerprint and GCN-based Indoor Localization using Graph Structures Fusion. In *Proc. 23rd IEEE Int. Conf. Commun. Technol. (ICCT)*, pages 111–116, Wuxi, China, 2023.

[4] B. C. Fargas and M. N. Petersen. GPS-free Geolocation using LoRa in Low-power WANs. In *Proc. Global Internet Things Summit (GIoTS)*, pages 1–6, Geneva, Switzerland, June 2017.

[5] GPSJAM. Daily Maps of GPS Interference. Online. Available: `https://gpsjam.org/`, 2025. Accessed Oct. 28, 2025. Data provided by ADSBexchange.

[6] P. D. Groves. Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. *Journal of Navigation*, 64(3):417–430, 2011.

[7] R. P. S. Hada and A. Srivastava. A Hybrid Approach for Localisation of Sensor Nodes in Remote Locations. *ACM Trans. Sensor Netw.*, 21(2):23, March 2025.

[8] R. P. S. Hada and A. Srivastava. A Hybrid Approach for Localisation of Sensor Nodes in Remote Locations. *ACM Trans. Sensor Netw.*, 21(2):23, March 2025.

[9] P. J. Huber. Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[10] W. Ingabire, H. Larijani, and R. M. Gibson. LoRa RSSI based Outdoor Localization in an Urban Area using Random Neural Networks. 284:1032–1043, 2021.

[11] J.-H. Kang, K.-J. Park, and H. Kim. Analysis of Localization for Drone-fleet. In *Proc. ICTC*, pages 533–538, Jeju, Korea (South), 2015.

[12] X. Kang, X. Liang, and Q. Liang. Indoor Localization Algorithm based on a High-order Graph Neural Network. *Sensors*, 23(19):8221, 2023.

[13] P. Müller, H. Stoll, L. Sarperi, and C. Schüpbach. Outdoor Ranging and Positioning based on LoRa Modulation. In *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, pages 1–6, Tampere, Finland, June 2021.

[14] J. Pospisil, R. Fujdiak, and K. Mikhaylov. Investigation of the Performance of TDoA-based Localization over LoRaWAN in Theory and Practice. *Sensors*, 20(19):5464, 2020.

[15] J. Pospisil, R. Fujdiak, and K. Mikhaylov. Investigation of the Performance of TDoA-based Localization over LoRaWAN in Theory and Practice. *Sensors*, 20(19):5464, 2020.

[16] J. Purohit, X. Wang, S. Mao, X. Sun, and C. Yang. Fingerprinting-based Indoor and Outdoor Localization with LoRa and Deep Learning. In *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pages 1–6, Taipei, Taiwan, December 2020.

[17] S. Richhariya, K. Wanaskar, S. Shrivastava, and J. Gao. Surveillance Drone Cloud and Intelligence Service. In *Proc. MobileCloud*, pages 1–10, Athens, Greece, 2023.

[18] D. Schneider. The Delivery Drones are Coming. *IEEE Spectrum*, 57(1):28–29, January 2020.

[19] M. Simonovsky and N. Komodakis. Dynamic Edge-conditioned Filters in Convolutional Neural Networks on Graphs. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 29–38, Honolulu, HI, USA, July 2017.

[20] R. S. Sinha, Y. Wei, and S.-H. Hwang. A Survey on LPWA Technology: LoRa and NB-IoT. *ICT Express*, 3(1):14–21, 2017.

[21] M. Slabicki, G. Premsankar, and M. Di Francesco. Adaptive Configuration of LoRa Networks for Dense IoT Deployments. In *Proc. NOMS*, pages 1–9, Taipei, Taiwan, 2018.

[22] J. P. Shanmuga Sundaram, W. Du, and Z. Zhao. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. *IEEE Commun. Surv. Tutorials*, 22(1):371–388, 2020.

[23] A. Vazquez-Rodas, F. Astudillo-Salinas, C. Sanchez, B. Arpi, and L. I. Minchala. Experimental Evaluation of RSSI-based Positioning System with Low-cost LoRa Devices. *Ad Hoc Networks*, 105:102168, August 2020.

[24] R. Vishwakarma, R. B. Joshi, and S. Mishra. IndoorGNN: A Graph Neural Network-based Approach for Indoor Localization using WiFi RSSI. In V. Goyal et al., editors, *Big Data and Artificial Intelligence*, volume 14418, pages 134–144, Cham, Switzerland, 2023. Springer.

[25] J. Xiong, Q. Qin, and K. Zeng. A Distance Measurement Wireless Localization Correction Algorithm Based on RSSI. In *Proc. ISCID*, pages 276–278, Hangzhou, China, 2014.

[26] W. Yan, D. Jin, Z. Lin, and F. Yin. Graph Neural Network for Large-scale Network Localization. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 5250–5254, Toronto, ON, Canada, June 2021.