UNIVERSITY OF CALIFORNIA, MERCED

Towards Practical AI for Networked Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering and Computer Science

by

Yuning Chen

Committee in charge:

Professor Wan Du, UC Merced, Chair
Professor Shawn Newsam, UC Merced
Professor Hua Huang, UC Merced

May 2025

The dissertation of Yuning Chen is approved, and
it is acceptable in quality and form for publication
on microfilm and electronically:

Professor Shawn Newsam

Professor Hua Huang

Professor Wan Du, Chair

University of California, Merced

May 2025

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Wan Du, for his unwavering guidance throughout my graduate studies. His support has been instrumental in shaping my vision and driving my progress. I will always cherish our shared passion for research, the countless papers and innovations we explored and discussed, and the many late nights spent tackling challenging research questions and refining our writing. I will continue to pursue on-the-ground research and development throughout my life, as he has profoundly inspired and empowered me to do so.

I am also sincerely thankful to my committee members, Prof. Shawn Newsam and Prof. Hua Huang, for their valuable insights and support throughout my research journey. Their thoughtful questions and constructive feedback during my qualifying exam were particularly inspiring, shaping the direction of my final-year research and empowering me throughout the job search process.

Additionally, I would like to acknowledge the support of my current and former colleagues, including Kang Yang, Sikai Yang, Zhiyu An, Khaled M. Bali, Brady E. Holder, Luke T. Paloutzian, Xianzhong Ding, Miaomiao Liu, Gang Yan, Yuanlin Yang, Fan Zhao, Zhibo Hou, as well as the master and undergraduate students who contributed along the way. Their collaboration and encouragement have made this journey both productive and memorable.

Finally, I would like to thank my family, especially my wife, Yuxin, for her unwavering trust, love, and support. I am deeply grateful for the passion we share and the journey weve taken together toward the future we envision.

VITA

2017 - 2021	B. E. in Software Engineering, Tsinghua University
2021 - Now	Ph. D. in Electrical Engineering and Computer Science, University of California, Merced

PUBLICATIONS

Conference and Journal Papers

Kang Yang*, **Yuning Chen***, and Wan Du, "FLog: Automated Modeling of Link Quality for LoRa Networks in Orchards", *ACM Transactions on Sensor Networks 21.2 (2025): 1-28.(TOSN)*.

Yuanlin Yang*, **Yuning Chen***, Kang Yang*, Sikai Yang and Wan Du, "Demo Abstract: Comprehensive Wireless Soil Component Sensing via VNIR and LoRa", *The ACM Conference on Embedded Networked Sensor Systems (SenSys Demo)*, Irvine, May 2025.

Yifei Xu, Xumiao Zhang, **Yuning Chen**, Pan Hu, Xuan Zeng, Zhilong Zheng, Xianshang Lin, Yanmei Liu, Songwu Lu, Z. Morley Mao, Wan Du, Dennis Cai, Ennan Zhai, and Yunfei Ma, "Roaming Free in the VR World with MP2", 2025 USENIX Annual Technical Conference (ATC), Boston, July 2025.

Yuning Chen, Kang Yang, Zhiyu An, Brady Holder, Luke Paloutzian, Khaled Bali, and Wan Du, "MARLP: Time-series Forecasting Control for Agricultural Managed Aquifer Recharge", *The ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Barcelona, August, 2024.

Yifei Xu*, **Yuning Chen***, Xumiao Zhang*, Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu, Wan Du, Z. Morley Mao, Dennis Cai, and Ennan Zhai, CloudEval-YAML: A Practical Benchmark for Cloud Configuration Generation, *The Annual Conference on Machine Learning and Systems (MLSys)*, Santa Clara, May, 2024.

Kang Yang, **Yuning Chen**, and Wan Du, OrchLoc: In-Orchard Localization via a Single LoRa Gateway and Generative Diffusion Model-based Fingerprinting, *The ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Tokyo, June, 2024.

Kang Yang*, **Yuning Chen***, Tingruixiang Su, and Wan Du, "Link Quality Modeling for LoRa Networks in Orchards", *The International Conference on Information Processing in Sensor Networks (IPSN)*, San Antonio, May 2023.

Yifei Xu*, **Yuning Chen***, Xumiao Zhang*, Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu, Wan Du, Z. Morley Mao, Dennis Cai, and Ennan Zhai, CloudEval-YAML: A Realistic and Scalable Benchmark for Cloud Configuration Generation, *Conference on Neural Information Processing Systems Workshop on ML for Systems (NeurIPS workshop)*, New Orleans, Dec, 2023.

Zhizhang Hu*, Shangjie Du*, **Yuning Chen**, Xuan Zhang, Wan Du, Asa Bradman, and Shijia Pan, Poster Abstract: Enhancing Fault Resilience of Air Quality Monitoring in San Joaquin Valley: A Data Equity Analysis, *The ACM Conference on Embedded Networked Sensor Systems (SenSys Poster)*, Istanbul, November, 2023.

Under Submission

Yuning Chen, Kang Yang, Yuanlin Yang, and Wan Du, GenRTC: Enabling Generative Visual Captions in Live Video Streaming, 2025, Under Submission.

Yuning Chen, Zhiyu An, and Wan Du, GrangerNet: Multi-variate Time Series Modeling with Structural Granger Causality, 2025, Under Submission.

Yuanlin Yang*, **Yuning Chen***, Kang Yang*, Sikai Yang and Wan Du, SoilX: Calibration-Free Comprehensive Soil Sensing Through Contrastive Cross-Component Learning, 2025, Under Submission.

Minyuan Zhou*, **Yuning Chen***, Yifei Xu, Jiaqi Zheng, Guihai Chen, Wanchun Dou, Pan Hu, Yongping Tang, Wendong Yin, Jie Lin, Qingyan Yu, Yuanchao Su, Songwu Lu, Wan Du, "AnyPro: Preference-Preserving Anycast Optimization based on Strategic AS-Path Prepending", 2025, Under Submission.

Kang Yang, **Yuning Chen**, Wan Du, "GWRF: A Generalizable Wireless Radiance Field for Wireless Signal Propagation Modeling", 2025, Under Submission.

ABSTRACT OF THE DISSERTATION

Towards Practical AI for Networked Systems

by

Yuning Chen
Doctor of Philosophy in Electrical Engineering and Computer Science
University of California Merced, May 2025
Professor Wan Du, UC Merced, Chair

Artificial intelligence has made remarkable progress in areas such as vision, language, and robotics, yet its deployment in real-world networked systems remains limited. These systems, ranging from wireless sensor networks to live-streaming platforms, present unique challenges such as unreliable data transmission, complex environmental dynamics, and strict system-wise constraints.

This dissertation presents a suite of learning-enabled systems designed to tackle the unique challenges of real-world networked environments. It begins with a physically grounded propagation model for LoRa-based communication in orchards, enabling accurate signal estimation under dense canopy and terrain constraints. Building on this foundation, a model predictive control framework is developed for groundwater recharge optimization in agriculture, leveraging long-term forecasting and causal learning to ensure both water sustainability and crop safety. Inspired by the need for universal temporal reasoning, GrangerNet advances time series modeling by unifying Granger causality and delay differential equations to enhance interpretability and predictive accuracy. Extending this line of work into interactive networked applications, GenRTC addresses the real-time demands of generative AI by introducing a scalable system for low-latency visual captioning in live-streamed content. Together, these systems demonstrate how co-designing machine learning with domain and system knowledge enables robust, interpretable, and deployable intelligence in diverse networking scenarios.

Chapter 1

Introduction

The rapid progress of Artificial Intelligence (AI) has revolutionized a wide array of domains, enabling breakthroughs in areas such as image generation, predictive analytics, and human-computer interaction. While many of these advancements have been successfully applied in domains like finance, healthcare, and conversation, their practical adoption in networked systems remains underexplored.

Networked systems form the foundation of modern infrastructure and are crucial to our daily lives. They encompass low-power and long-range data communication, edge and in-network signal processing, control decision making, and a suite of downstream applications including video streaming and augmented/virtual reality [3]. These applications span industries such as smart agriculture, manufacturing, and remote work. The deployment of AI in these settings presents unique challenges, such as noisy environmental interactions, stringent system-wise constraints, and complex context for AI to understand and support. To address these challenges, this dissertation takes a system-first approach to embedding intelligence in networked systems. Through a series of works, it explores how to develop practical, interpretable, and domain-adaptive machine learning systems for real-world networking scenarios.

The journey in filling this gap starts in Chapter 2 with a real-world deployment of a LoRa sensor network in California's almond orchards. Despite the promise of LoRas long-range communication, empirical observations reveal significant performance degradation in dense orchards due to the complex propagation environment.

To address this, the dissertation presents FLog, a novel signal propagation model tailored for structured orchards [4, 5]. FLog captures signal shadowing through a 3D Fresnel Zone-based model and incorporates both tree canopies and ground effects. It offers interpretable and accurate signal strength estimation, which enables better planning of gateway deployment and network coverage. This work demonstrates how domain structure (regular tree patterns) and physical-layer insights can be embedded into data-driven modeling to provide practical and efficient solutions for real-world sensor networks.

Building on the success of the LoRa network deployment, Chapter 3 extends the AI-embedded system into a critical water management practice in California: Agricultural Managed Aquifer Recharge (Ag-MAR). The objective is to recharge groundwater aquifers by intentionally flooding farmland in the rain season. However, this process must be carefully managed to avoid crop damage due to root oxygen deficit in the soil. This chapter introduces MARLP, a Model Predictive Control (MPC) system that leverages long-term forecasting of soil oxygen levels under various flooding schedules. It integrates multi-periodicity time series learning, causal-aware calibration with weather forecasts, and domain-specific planning heuristics to deliver real-time, optimized flooding strategies. MARLP explores how temporal reasoning and causality can be harnessed for predictive control in complex agricultural ecosystems.

Chapter 4 further abstracts from the Ag-MAR control scenario to propose a general-purpose causal time series model called GrangerNet. GrangerNet builds on the classical vector autoregressive (VAR) modeling in Granger causality by integrating it with delay differential equations (DDE) theory and a learned Granger matrix, which encodes variable interactions across time lags and derivative orders. Unlike prior work that only infers causal structures, GrangerNet uses this structure to directly improve forecasting and imputation tasks in multivariate time series. Extensive evaluation demonstrates that GrangerNet not only produces interpretable causal graphs, but also achieves state-of-the-art performance on real-world datasets. This work illustrates the power of interpretable inductive biases in improving the generalization and transparency of time series models.

To extend the AI adoption in various network problems, Chapter 5 shifts the focus from sensor networks to interactive video systems, addressing the problem of integrating AI-powered visual captioning into live-streamed media. As real-time generative models grow in popularity, deploying them under the strict latency and bandwidth constraints of live systems becomes increasingly difficult. This chapter introduces GenRTC, a practical visual captioning system for live streaming that predicts speaker intent and proactively generates relevant images. GenRTC includes a cross-layer rendering pipeline that synchronizes generative outputs with WebRTC video frames, and an MPC-based resource scheduler that maintains QoE across concurrent sessions. Together, these components deliver an end-to-end low-latency experience for viewers without overloading the media server, further pushing the AI adoption in human-computer iteration via multimedia networked systems.

In Chapter 6, we describe the future directions of what we plan to study based on the scaling law in multi-modal large language models. Different from images, the modality in wireless and network has unique challenges: (1) Different input format caused by different frequency bands and hardware specifications; (2) Sparse information in the spectrum; (3) Key insights are embedded in both temporal domain and frequency domain. Preliminary data collection and grounding has been done to provide a high-quality pre-training datatset. Future works will explore the scaling law in pretraining, post-training, and inference-time optimizations. The vision is that AI can utilize the networked system as a window to understand the physical world and make trustworthy interpretations and control decisions.

In summary, this dissertation contributes a suite of real-world task-driven algorithms and systems for networked applications. It demonstrates how domain knowledge, time series modeling, causal reasoning, and system co-design can be woven into practical AI tools. By addressing the distinct challenges of sensor reliability, predictive control, causal modeling, and real-time streaming, this dissertation advances the frontier of applied AI for networked systems. Its findings have the potential to enable more resilient, interpretable, and intelligent infrastructure in a broad range of application domains.

Chapter 2

Automated Modeling of Link Quality for LoRa Networks in Orchards

LoRa networks have been deployed in many orchards for environmental monitoring and crop management. An accurate propagation model is essential for efficiently deploying a LoRa network in orchards, *e.g.*, determining gateway coverage and sensor placement. Although some propagation models have been studied for LoRa networks, they are not suitable for orchard environments, because they do not consider the shadowing effect on wireless propagation caused by the ground and tree canopies. This paper presents *FLog*, a propagation model for LoRa signals in orchard environments. *FLog* leverages a unique feature of orchards, *i.e.*, all trees have similar shapes and are planted regularly in space. We develop a 3D model of orchards. Once we have the location of a sensor and a gateway, we know the media that the wireless signal traverses. Based on this knowledge, we generate the First Fresnel Zone (FFZ) between the sender and the receiver. The intrinsic path loss exponents (PLE) of all media can be combined into a classic Log-Normal Shadowing model in the FFZ. Extensive experiments in almond orchards show that *FLog* reduces the link quality estimation error by 42.7% and improves gateway coverage estimation accuracy by 70.3%, compared with a widely used propagation model. The source codes and dataset are released at https://github.com/ycucm/Flog.

2.1 Introduction

The orchard crop is a significant contributor to the economy in California, generating over 8.96 billion dollars in profit in 2021 [6]. Smart orchard management, such as smart irrigation [7, 8], tree health monitoring [9], air quality monitoring [10] and pest control [11], is crucial for improving yield and minimizing cost [12, 13, 14]. All these applications need to deploy a large number of sensors across the orchards, *e.g.*, tens or even hundreds of acres. LoRa is a promising solution with a communication range of several miles, allowing gateways to receive sensor data in a large field [15, 16, 17, 18, 19, 20, 21, 22]. However, this range is only achievable in free space [23, 24], which is rarely found in orchards.

Our preliminary experimental results reveal that in an almond orchard, LoRa signals can only be reliably transmitted up to 140 m with a Spreading Factor (SF) of 10; whereas LoRa signals with the same hardware and transmission parameters can reach a communication distance of 2 km in free space. In orchards, LoRa sensor nodes are typically installed close to the ground, under the tree canopy. Meanwhile, LoRa gateways are often placed on towers of 6 - 10 m because there are no high buildings on farms and it is expensive to construct taller towers. As a result, the energy of wireless signals may be absorbed and reflected by the ground, which can affect the LoRa signal propagation. In addition, the wireless signal from a sensor node must penetrate the canopies of multiple trees to reach a gateway.

In this paper, we study the propagation modeling [25] of LoRa signals in orchard environments, which requires us to estimate the attenuation of wireless signals as they propagate through all media between the transmitter and the receiver, such as air and the canopy of trees. Although two propagation models [26, 27] for LoRa have been developed for large-scale urban scenarios, but they cannot be used in orchards. For example, for a sensor node in orchards, its signal propagation varies significantly across different directions, since the signal traverse different distances in canopies. In addition, several foliage propagation models [28, 29, 30] have been studied for wireless signals passing through trees. They are developed for high-frequency channels (*e.g.*, 5 GHz in satellite communications), but not applicable to LoRa links. The propagation characteristics of low-frequency LoRa signals are

different from high-frequency signals, and these foliage propagation models do not consider the impact of the ground on LoRa signal propagation.

To study the characteristics of LoRa signal propagation in orchards, we conduct a series of experiments using a gateway and a LoRa sensor node. The latter is deployed at different locations for experiments (Section 2.4.1 for details). Based on our experimental results, we made three observations. 1) If the sender and the receiver are both under trees, the link quality of the visual line-of-sight (LoS) path is worse than that of non-visual LoS paths. When a wireless signal passes through a tree, it may penetrate, scatter (*i.e.*, reflected or refracted), or diffract around the tree. This power loss is known as the shadowing effect, which needs to be analyzed from a 3D perspective. 2) Due to the long wavelength of LoRa signals, wireless signals can easily diffract from tree trunks and branches. 3) The low installation height of LoRa sensor nodes means that the ground has an apparent shadowing impact on signal propagation.

Based on the observations above, this paper presents *FLog*, a wireless signal propagation model designed for LoRa links in orchards. It is inspired by the regular layout of orchards, where all trees are aligned and have a similar shape and canopy density, as they are planted and pruned at the same time. The shape of a tree can be modeled by a few parameters, including tree height, trunk height, and canopy width. The layout of an orchard can be modeled by the shape of its trees and the spacing distances between rows and columns. Given the locations of a sensor node and a gateway, we can leverage our 3D orchard model to study the complex shadowing effect that the wireless signal will experience from the sensor node to the gateway. Based on the shadowing effect analysis, our wireless propagation model can calculate the attenuation that the signal will have as it travels through the air and tree canopies. Combining the predefined transmission settings, including antenna gains and transmission power, we can calculate the strength of the received LoRa signal.

FLog adopts the First Fresnel Zone (FFZ) to capture the complex shadowing effect caused by tree canopies and ground in orchards. FFZ is a 3D ellipsoid region with two focus points located at the node and the gateway. This zone carries most

of the signal energy received by the receiver. Different links have their unique portions of the wireless transmission media in their FFZs. To calculate the portion of each medium for a transmission pair, we perform numerical sampling in the FFZ. Each sampling point may encounter free space, trees, or the ground. The portions of sampling points in each medium over the total number of points are used as weights for profiling the shadowing effect. These weights are used to combine the intrinsic path loss exponent (PLE) of each transmission medium, resulting in the final PLE of the classic Log-Normal Shadowing model in the FFZ.

Beside the weights, we also need to determine the intrinsic PLEs for each medium in *FLog*. We obtain these parameters by fitting the collected packets' signal strength using the nonlinear least square algorithm [31]. Furthermore, we adapt these parameters to environmental variation based on a few recently received packets.

To demonstrate the applications of our propagation model, we use *FLog* to determine gateway coverage. A LoRa node is considered to be covered by a gateway if the Packet Delivery Ratio (PDR) exceeds 80% [32, 33]. The PDR is computed using a BER model [34], with inputs of Signal-to-Noise Ratio (SNR) and SF. We use *FLog* to estimate the SNR for any LoRa nodes and a gateway, and thus the coverage of that gateway under different transmission settings.

Extensive experiments have been conducted in two almond orchards and one walnut orchard. The results show that *FLog* can reduce path loss estimation errors by up to 42.7% compared to the Log-Normal Shadowing model.

In summary, this paper makes three major contributions:

- We study the propagation modeling problem of wireless LoRa signals in orchards and explain why path loss models only considering the direct ray blockage are unsuitable for the orchard scenario.

- We propose *FLog*, a novel interpretable propagation model for LoRa networks in orchards. It leverages the first Fresnel zone theory and the regular tree layout of orchards to model the complex shadowing effect caused by tree canopies and the ground with minimal overhead.

- Extensive experiments demonstrate the effectiveness of *FLog*. We release *FLog* source code on GitHub [35].

2.2 background and Motivation

After a brief introduction to the basic LoRa concepts and the Log-Normal Shadowing model, we conduct experiments to study LoRa links in free space and orchards.

2.2.1 LoRa Primer

The LoRa adopts Chirp Spread Spectrum (CSS) to facilitate long-distance and low-power communication [36, 37, 38].

The sensor data undergoes a series of encoding operations at the sender side to improve its over-the-air resilience. These operations consist of Forward Error Correction (FEC) encoding, whitening, diagonal interleaving, and gray mapping, and are followed by CSS modulation of the encoded data into multiple symbols. Each symbol represents an integer from 0 to $2^{SF} - 1$, where SF determines the number of bits in each symbol. To modulate a symbol, a base chirp with an initial frequency is shifted by a step of $BW/2^{SF}$, where BW is the frequency channel bandwidth.

At the receiver side, LoRa performs demodulation and decoding. Demodulation identifies a symbol's value by measuring its chirp's initial frequency, and the resulting signal is subjected to FFT to obtain the amplitude spectrum of each received symbol. Each frequency bin corresponds to a possible symbol value, and the symbol value with the highest amplitude is identified by the frequency bin. The recognized symbols are then concatenated into a binary sequence that undergoes gray demapping, deinterleaving, and dewhitening in sequence. The original sensor data can be obtained by FEC decoding.

Frequency Band: LoRa operates on license-free radio frequency bands, such as 915 MHz in North America. It can transmit over long distances of several miles or more in rural areas. In contrast, high-frequency bands such as 2.4 GHz or 5

GHz used by WiFi have a short communication range, typically limited up to 45 m indoors [39, 40]. This short communication distance makes it difficult to provide coverage for orchards that span several acres.

Expected Signal Power (ESP): LoRa defines the ESP to quantify the received signal strength [26, 41, 27], which is derived from the measured RSSI (Received Signal Strength Indicator) and SNR:

$$\text{ESP} = \text{RSSI} + \text{SNR} - 10 \cdot \log_{10}\left(1 + 10^{0.1 \cdot \text{SNR}}\right) \tag{2.1}$$

where the unit of ESP and RSSI is dBm and the unit of SNR is dB. The RSSI and SNR are reported from the LoRa gateway for each received packet.

2.2.2 Log-Normal Shadowing Model

The Log-Normal Shadowing model is widely used to predict the received signal power [42]. The ESP of a received packet, P_{rx}, is calculated as follows [42]:

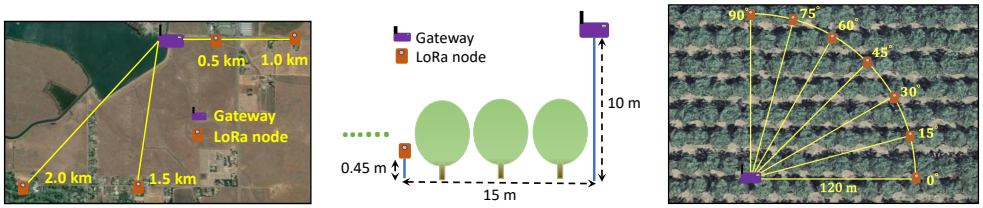
$$P_{rx} = P_{tx} + G_{tx} + G_{rx} - PL \tag{2.2}$$

where P_{tx} represents the transmission power in dBm, G_{tx} and G_{rx} denote the transmitting and receiving antenna gains in dBi, respectively. The last term PL is the path loss [43] in dB and can be calculated via the Log-Normal Shadowing model [42]:

$$PL\left(d\right) = \overline{PL}\left(d_0\right) + 10 \cdot n \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma \tag{2.3}$$

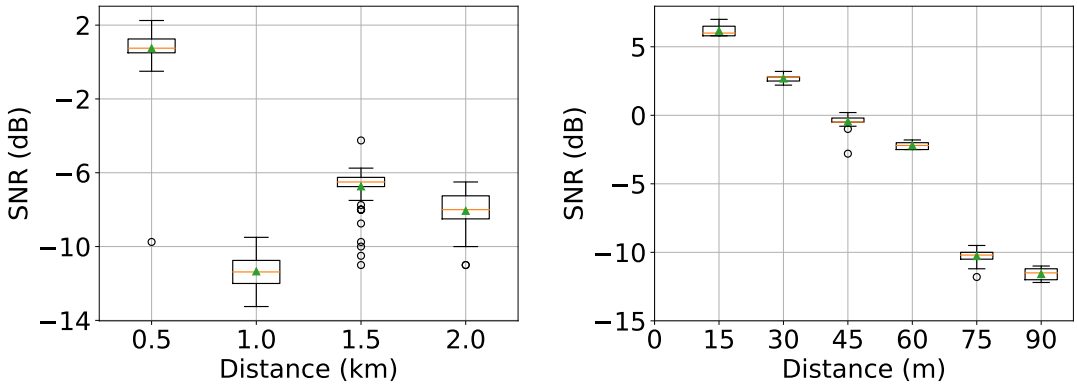
where the distance between a LoRa node and a gateway is denoted as d in meters, n is the PLE, and X_σ is a zero-mean Gaussian distribution with a standard deviation of σ. The reference path loss $\overline{PL}\left(d_0\right)$ is obtained from field measurements at a reference distance of d_0, where d_0 is normally set to 1 m [26]. Based on our collected packets, we obtained an average $\overline{PL}\left(d_0\right)$ of 78.59 dBm.

When considering a specific distance, the model represents the path loss as a Gaussian distribution with a mean of $\overline{PL}(d_0) + 10n\log\left(\frac{d}{d_0}\right)$ and a standard deviation of σ. As the model produces a distribution, its accuracy cannot be assessed against a single measurement. Instead, we use the mean as the model's predictive result, enabling the calculation of the absolute error relative to the



(a) Free space scenario (b) A typical scenario in orchards (c) Nodes at seven directions

Figure 2.1: The deployment layout of the almond orchard and illustration for different directions.



(a) Free space scenario (b) The orchard scenario

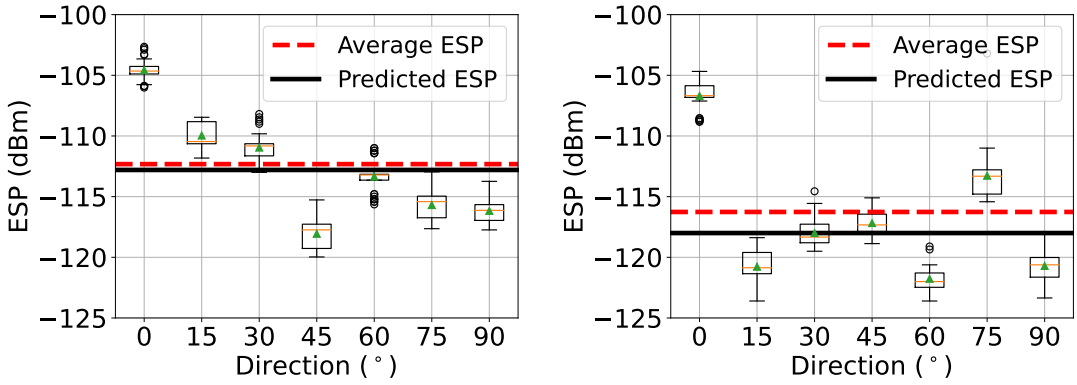
Figure 2.2: The measured link quality (SNR) in the free space and almond orchard.

measured signal strength. In the following, unless specified otherwise, the output of the Log-Normal Shadowing model is assumed to be the mean value.

2.2.3 Free Space Scenario

We conducted an experiment to measure the LoRa link quality in the free space. Figure 2.1(a) shows that a LoRa node is placed at four locations with distances of 0.5, 1.0, 1.5, and 2 km. The nodes and gateway are installed on two poles with heights of 6 m and 10 m, respectively. The nodes transmitted packets periodically with SF10, $P_{tx} = 14$ dBm, a bandwidth of 125 kHz, and a coding rate of 4/5.

Figure 2.2(a) depicts the experimental results. Even at a distance of 1.5 km, the link's SNR remains above the receiving sensitivity threshold (*i.e.*,-7.5 dB for



(a) Distance = 80 m

(b) Distance = 120 m

Figure 2.3: The measured link quality (ESP) at different directions for two communication distances.

SF7). This indicates that LoRa nodes can transmit packets using SF7 at distances of up to 1.5 km in free space. However, the SNR at a distance of 1.0 km is lower than that at 1.5 and 2.0 km, primarily because buildings and trees obstruct the propagation path, as illustrated in Figure 2.1(a). This highlights the significant impact of blockages on link quality.

2.2.4 Orchard Scenario

Figure 2.1(c) depicts an almond orchard where trees are organized tidily to facilitate uniform allocation of sunlight, water, and soil nutrients. LoRa nodes and the gateway are deployed in orchards as shown in Figure 2.1(b). The nodes are typically positioned on the ground or attached to the main branches, measuring data related to soil and tree health. The installation height is usually between the ground and the canopy, less than 1.2 m (*e.g.*, 0.45 m in our implementation). Meanwhile, the gateway is mounted on top of a pole or tower (*e.g.*, 10 m in our implementation), providing long communication coverage, as shown in Figure 2.11. Therefore, the signals from sensor nodes normally pass through three media: free space, the ground, and trees in orchards.

Short Communication Distance

We conducted an investigation of link quality in an almond orchard. The position of the gateway is fixed in the center between two adjacent almond trees in a row. We then move the locations of LoRa nodes in the row, the communication distance between the node and gateway ranges from 15 m to 90 m with a step of 15 m. The transmission settings were identical to those outlined in Section 2.2.3. From Figure 2.2(b), we observed a significant reduction in communication: at only 90 m, the SNR was -12.0 dB. As a result, SF10 was necessary for a reliable communication distance of 90 m. Conversely, SF7 was suitable for distances of 1.5 km in free space. By fitting our collected data, we found that the PLE in the almond orchard was 2.95. This value is larger than that of free space, indicating a shorter communication distance, and reflects the high path loss in the orchard caused by tree blockage.

Large Deviation at Different Directions

We then measured the received signal strength in seven directions, ranging from 0° to 90° in 15° increments for a fixed distance, as depicted in Figure 2.1(c). Figure 2.3 shows the ESP of the received signal at different directions for communication distances of 80 m and 120 m. The dotted red line represents the average ESP in seven directions for one distance. The results indicate a significant variation in the signal ESP across different directions, with discrepancies of up to 13 dBm, such as -105 dBm *vs.*-118 dBm at 0 degrees and 45 degrees.

According to Equation 2.3, the Log-Normal Shadowing model predicts a mean ESP value for a given distance, as shown by the black line in Figure 2.3. Although the predicted ESP with the Log-Normal Shadowing model is close to the average ESP, with errors of less than 1 dBm, the overall ESP error across all the collected data can exceed 4 dBm due to the significant deviation in signal strength across different directions. This finding motivates us to develop a new propagation path loss model that can account for the variation in signal strength across different directions.

Table 2.1: Notations used in this paper.

Notations	Meaning
ESP	Expected signal power.
PLE	The path loss exponent (PLE) of a LoRa link.
α	The PLE when the space is filled with air.
β	The PLE when the space is filled with foliage.
γ	The PLE when the space is filled with the ground.
P_{open}	The portion of free space within the propagation path.
$P_{foliage}$	The portion of the foliage within the propagation path.
P_{ground}	The portion of the ground within the propagation path.
p_{gap}	The distance between two adjacent sampling planes.
N_{ref}	The number of sampling points of the first sampling plane.

2.3 Strawman Solution

This section develops a strawman solution to model link quality in orchards. We first build an orchard 3D profile. Using this profile, we specify the PLE of a Log-Normal Shadowing model. Table 2.1 summarizes the notations used in this paper.

2.3.1 3D Modeling of Orchards

Abstracting One Tree: Botanists have developed several detailed tree models based on physiological knowledge [44]. However, these models require precise measurements of the shapes of trunks, branches, and leaves for all trees in an orchard, which is a labor-intensive process. We adopt a simple solution proposed by Torrico *et al.* [45]. It profiles the trunk as a cylinder and abstracts the crown as an ellipsoid with varying horizontal and vertical radii, as shown in Figure 2.1(b). To create a profile of a tree, we need to measure its height, trunk height, canopy width, and trunk radius.

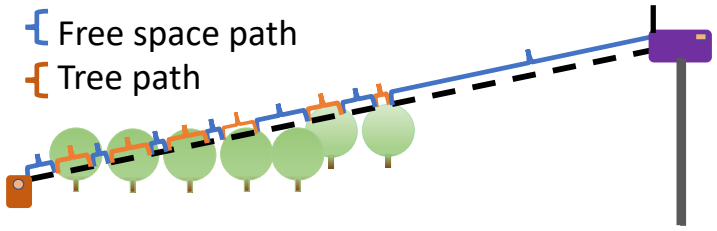


Figure 2.4: The illustration of the strawman solution.

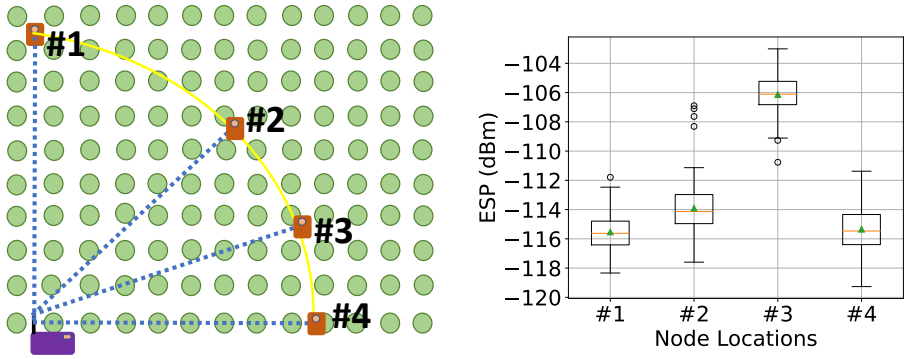
Abstracting an Orchard: Orchards typically have a uniform layout, with trees being planted at the same time of year and exhibiting similar shapes. To analyze an orchard, we adopt a Cartesian coordinate system with the x-axis and y-axis representing the directions along and across rows, respectively. We also measure the distances between adjacent rows and adjacent trees in one row, which determine the positions of all trees in the Cartesian coordinate system.

Therefore, by adding two parameters, we can extend tree modeling to orchard modeling. With all of these parameters as input, we can reconstruct the orchard as 3D shapes, which can be translated into point clouds. Specifically, given the position of any point, we can determine whether it falls within the foliage or not.

2.3.2 Adapting PLE with Line Shadowing

Figure 2.4 depicts the signal propagation path between a LoRa node and a gateway, where the path is viewed as a direct line. The path includes both free space and foliage portions, which can have different shadowing effects on the signals. Thus, it is intuitive to consider these two parts separately.

To achieve this, we calculate the free space and foliage portions along the propagation path's direct line, denoted as P_{open} and $P_{foliage}$, respectively. To calculate the foliage portion $P_{foliage}$, we equally split the direct line into numerous points. If a point is in the tree, then the foliage points increase by 1; otherwise, the free space points increase by 1. Finally, the foliage portion $P_{foliage}$ is obtained by dividing the foliage points by the total number of points, which is similar to the free space portion calculation. The sum of these portions should be equal to one.



(a) Experimental Settings (b) ESP for four locations

Figure 2.5: Comparison between visual and non-visual LoS paths.

As free space and foliage have different shadowing effects, they have different PLEs in the Log-Normal Shadowing model. To account for this heterogeneity, we propose an adaptive PLE calculation approach. This involves separating the path into two types of media (free space and foliage) and then recombining them together.

$$PLE_{comb} = P_{open} \cdot \alpha + P_{foliage} \cdot \beta \tag{2.4}$$

where α denotes the free space PLE and β is the PLE when the space is filled with foliage. To determine the values of α and β, we perform a least square fitting on a collected dataset. The final PLE_{comb} is a compromise between the two values. For example, if $P_{open} = 1$, it means that no trees are present and the path can be treated as free space. Once we have determined the PLE_{comb}, we can calculate the total path loss using the Log-Normal Shadowing model.

From experimental results in Figure 2.20(c), the strawman solution (marked as "LLog") can provide different estimations in different directions, but this model underestimates or overestimates the path loss in some directions. The possible reason is that we modeled the propagation path as a direct line. But the surrounding area also plays a role for the signal propagation.

2.4 The Design of *FLog*

This section presents tree validations via the empirical analysis of the LoRa signal propagation in orchards. We then introduce a path loss model based on the FFZ.

2.4.1 Empirical Analysis of LoRa Signal Propagation

To design a path loss model for orchards, it is crucial to understand how the signal propagates within this environment. Therefore, we conducted three preliminary experiments to investigate the LoRa signal propagation in orchards.

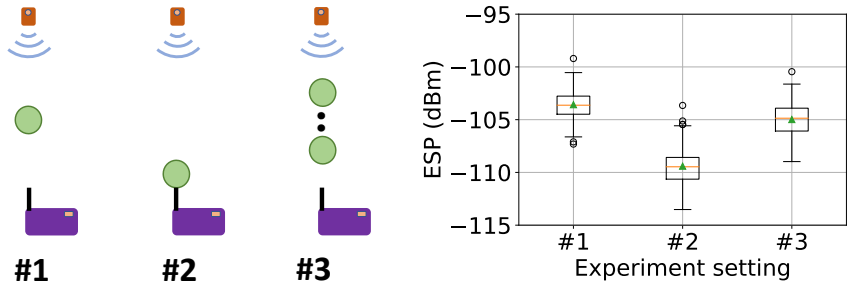
Visual Line-of-Sight Path

At frequencies around 900 MHz, the wavelength facilitates diffraction around obstacles. This implies that radio waves can 'bend' and may not necessarily demand a direct line of sight [46]. We have conducted experiments to validate this assertion.

Figure 2.5(a) shows the experimental setup where a gateway is positioned at the lower-left corner, and four different locations are selected to place LoRa nodes in varying directions but with the same communication distance of 60 m between the gateway and nodes. The gateway and nodes are placed at a height of 0.45 m. Locations #1 and #2 have the visual LoS propagation path, while locations #3 and #4 have the non-visual LoS path that passes through two and ten trunks, respectively. At each location, a LoRa node transmits packets periodically for three minutes with an interval of 1.5 s.

According to Figure 2.5(b), location #3 with the non-visual LoS path achieves the highest ESP, which is even higher than the visual LoS path (*i.e.*, location #1 and #2). Furthermore, despite having ten trunks in location #4, this non-visual LoS path still attains a comparable ESP to the visual LoS path at locations #1 and #2. In summary, we conclude that:

Observation 1: *In orchards, the surrounding environment around the visual LoS path plays a pivotal role in determining the strength of LoRa signal.*



(a) Experimental Settings (b) ESP for three settings

Figure 2.6: The diffracted signal strength.

Diffracted Signal Strength

Diffraction refers to the propagation of waves behind an obstruction.

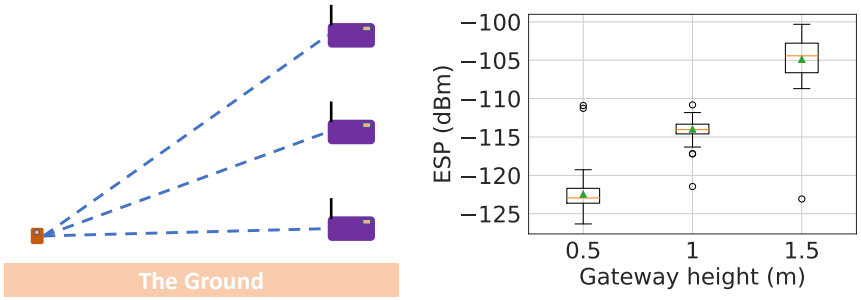
Figure 2.6(a) shows the setup of experiments, where the green circles represent tree trunks. In both settings #1 and #2, there is one trunk located between the nodes and the gateway. The difference is that in setting #2, the trunk is positioned very close to the gateway antenna, resulting in significantly weaker diffracted waves [47, 48].

Although setting #3 has five trunks between the gateway and node, resulting in weaker penetrating signal power, Figure 2.6(b) shows that its signal power is similar to that of setting #1. Furthermore, we can see that setting #2 exhibits the lowest signal strength, with a difference of more than 6 dBm compared to setting #1 and 4 dBm compared to setting #3. This leads us to the second validation:

Observation 2: *Comparing setting #1 with #2, diffraction plays a crucial role in LoRa signal propagation. Comparing setting #1 with #3, the signal power from diffraction outweighs the power transmitted in the direct line.*

Ground Absorption

Signal propagation in orchards can be impacted by the low height of nodes, resulting in the absorption of signal energy by the ground. To investigate this effect on signal power, we conducted experiments in a free sand region without any obstacles. The experiments involved fixing the horizontal distance at 100 m and maintaining the sensor height at 0.45 m while varying the gateway height



(a) Experimental Settings (b) ESP for three settings

Figure 2.7: The effect of ground absorption.

to 0.5 m, 1 m, and 1.5 m, as illustrated in Figure 2.7(a). The results, shown in Figure 2.7(b), reveal that increasing the gateway height leads to an increase in the received signal power.

Observation 3: *When the sender and receiver are in close proximity to the ground, a significant portion of the signal power can be absorbed by the ground.*

All of these validations demonstrate that the link quality can be influenced by the surrounding objects, and the "path" cannot be considered as simply the visual LoS path between the sender and receiver. Instead, more attention should be given to the surrounding regions along the visual LoS path when estimating path loss. To address this issue, we propose the adoption of FFZ in path loss calculations, as it takes into account the surrounding environment.

2.4.2 First Fresnel Zone for LoRa Signal

The FFZ refers to the 3D ellipsoid region around the direct path through which the received signal passes. Figure 2.8 depicts that a point P lies on the surface of the FFZ if and only if:

$$\sqrt{d_1^2 + r^2} + \sqrt{d_2^2 + r^2} = (d_1 + d_2) + \frac{\lambda}{2} \tag{2.5}$$

where λ refers to the wavelength. This equation quantifies confocal prolate ellipsoidal-shaped regions with the sender and receiver located at two focal points. This theory suggests that all media within the FFZ shadows the signal, not just media along the direct path.

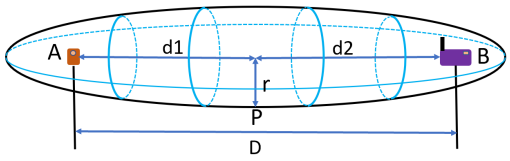


Figure 2.8: The illustration of FFZ.

Based on Equation 2.5, the volume of the FFZ increases proportionally with wavelength. For instance, at a distance of 100 m, the maximum radius of the FFZ could be 2.8 m. This highlights the need to quantify the shadowing of the surrounding area over the direct path.

2.4.3 Design of *FLog*

This section provides the design details: including the overview, portion quantification, intrinsic PLE fitting, and parameter adaptation mechanism.

Overview

Except for the foliage and free space, we also need to consider the ground shadowing effect. Because the FFZ of the LoRa signal has intersections with the ground due to the low height of sensor nodes. Therefore, we update the final PLE calculation Equation 2.4 as follows:

$$PLE_{comb} = P_{open} \cdot \alpha + P_{foliage} \cdot \beta + P_{ground} \cdot \gamma \tag{2.6}$$

where two new variables, P_{ground} and γ, are introduced to represent the ground shadowing effect. The values of α, β, and γ correspond to the intrinsic PLEs when the wireless signal propagates through free space, foliage, and ground, respectively. The portion of free space, foliage, and ground in the FFZ is denoted by P_{open}, $P_{foliage}$, and P_{ground}, respectively. The final PLE_{comb} in the FFZ is obtained as a weighted combination of α, β, and γ, where the corresponding weights are P_{open}, $P_{foliage}$, and P_{ground}. In an orchard, all the pairs of senders and receivers share the same α, β, and γ, but have different final PLE_{comb} due to the varying weights.

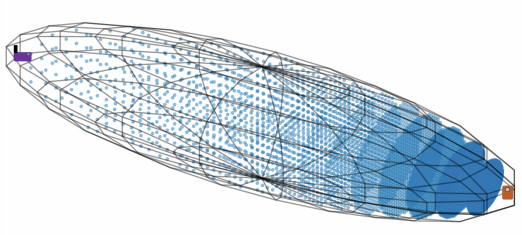


Figure 2.9: The illustration of sampling.

The next two subsections explain how to determine their values.

Portion Quantification

The weights of each media in an FFZ are calculated by computing the volume of each media and dividing it by the total volume of the FFZ. However, computing this volume requires solving a triple integral in a constrained 3D space, which is computationally intensive and time-consuming. In this paper, as shown in Figure 2.9, we used a sampling approach by dividing the FFZ into multiple planes with equal spacing. Each plane was then further sampled into numerous points.

Specifically, we incremented the number of corresponding media points by one if a sampling point was located within that media. For example, if a sampling point was in free space, we incremented the free space points by one. The portion of each media was then calculated by dividing the number of points contained in that media by the total number of points in the FFZ. Therefore, the sum of the portions for all three media types should be one.

Sampling Planes: The first sampling plane is located at $d_0 = 1\ m$, which is the reference distance in the Log-Normal Shadowing model. The remaining sampling planes are equally spaced at an interval of p_gap. We change the value of p_gap from 0.15 m to 0.35 m to calculate the portions of different mediums. Table 2.2 shows that there is no significant impact on sampling results for different p_gap values less than 0.5 m. Considering the computational burden and sampling effectiveness, we empirically set $p\_gap = 0.25\ m$.

The number of sampling points in each plane varies and is proportional to the signal energy in the plane, which is calculated by multiplying the power flux

Table 2.2: Average sampling results at 60 m *vs.* *p_gap*.

p_gap (m)	0.25	0.3	0.35	0.5	1.0
P_{open}	0.686	0.686	0.687	0.690	0.702
$P_{foliage}$	0.313	0.313	0.312	0.309	0.297

Table 2.3: Average sampling results at 60 m *vs.* N_{ref}.

N_{ref}	2360 k	590 k	148 k	74 k	18 k
P_{open}	0.686	0.686	0.686	0.686	0.687
$P_{foliage}$	0.313	0.313	0.313	0.313	0.313

density by the area of the plane. The radius of the plane can be determined via Equation 2.5. The power flux density of a sampling plane decreases proportionally to the square of the distance [42], *i.e.*, $4\pi d^2$, indicating that the number of sampling points in each plane should similarly decrease with distance. Therefore, the total number of sampling points on a plane i is given by $N^i_{plane} = N_{ref} \cdot S_i \ / \ 4\pi d_i^2$, where N_{ref} is a constant that can be adjusted to control the total number of sampling points, S_i is the area of plane i, and d_i is the distance between the sender and the plane i. Table 2.3 indicates that N_{ref} has no impact on sampling results if the sampling points are adequate. We set N_{ref} as 590 k in the evaluations.

Sampling Points in a Plane: A sampling plane i contains N^i_{plane} sampling points. To determine the locations of these points in the plane, we first locate the central point of the plane, which serves as the first sampling point. We then divide the plane into multiple circles with varying radii, each centered at the central point. At each circle, we sample N_{sample} points with equal arc length. Due to the diffraction loss, the power flux density of the received signal is greater at the center of the plane than at the edges [47, 48]. we accordingly set the radii of the circles to increase quadratically from the central point to the edge of the plane.

Intrinsic PLE Fitting

To calculate the final PLE_{comb} in Equation 2.6, we also need to determine the values of α, β, and γ. The process for obtaining these values is outlined below.

To determine the value of γ, we utilized non-foliage data from Figure 2.7 and applied the least square algorithm to Equation (2.6), where $P_{foliage} = 0$ and $\alpha = 2$. This yielded a value of $\gamma = 5.39$. The ground's dynamic nature poses challenges to algorithmic stability due to its high PLE, which significantly affects signal strength. Incorporating its variability can skew PLE values for air and trees, potentially impeding algorithm convergence. To maintain stability, we keep the ground's PLE constant and adjust the PLEs of air and trees to offset any ground variations. Hence, the 5.39 will serve as the default for the remainder of the paper unless otherwise noted.

To determine the values of parameters α and β, we collected packets at multiple pairs of transceivers in our testbed, as described in Section 2.5.1. For each pair, we calculated three portions: P_{open}, $P_{foliage}$, and P_{ground}. We then combined Equations 2.1 - 2.3 to obtain the final PLE_{comb}, using the RSSI and SNR values obtained after receiving a packet. Thus, the only unknown variables in Equation 2.6 are α and β. We then employed a least square error fitting algorithm to determine the optimal values of α and β.

Parameter Adaptation Mechanism

The parameters in Equation 2.6 could be impacted by different environmental dynamics, which can be categorized into four scenarios:

1. Short-term environmental noise variation.
2. **Transient weather changes**, *e.g.*, temperature.
3. **Foliage density changes** on a yearly cycle due to the growth and loss of leaves and fruits.
4. **Long-term foliage shape changes:** Shape changes as trunk and branch grow over years.

To mitigate the impact of dynamic (1), we collect multiple packets to compute the average signal power as the reference signal power at a given location. Environmental dynamics (2) and (3) can affect α and β in Equation 2.6. To recalibrate

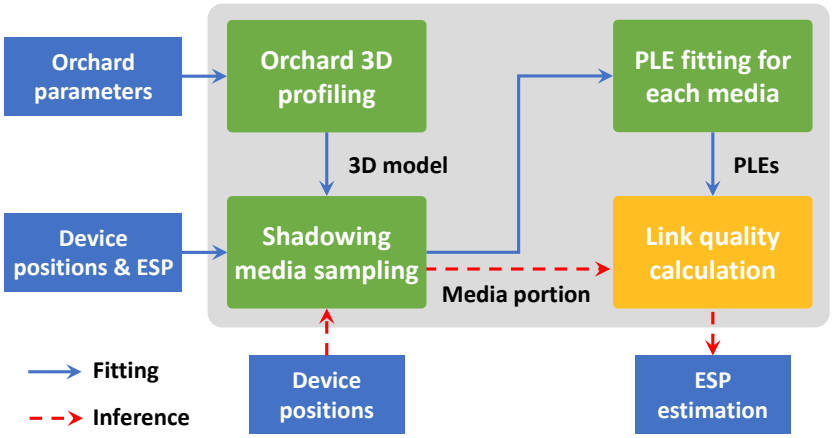


Figure 2.10: Workflow of the *FLog*, where the term *device* refers to both the LoRa node and the gateway.

these two parameters, we use the N most recently received packets from all sensor nodes. When a new packet is received, we obtain its measured ESP. Thus, the ESP of the past $N \times M$ packets from M nodes will form $N \times M$ equations to recalibrate α and β, as shown in the example equations below (where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$):

$$\begin{bmatrix} PLE_1^1 \\ \vdots \\ PLE_j^i \\ \vdots \\ PLE_M^N \end{bmatrix} = \begin{bmatrix} P_{\text{open}_1} & P_{\text{foliage}_1} & P_{\text{ground}_1} \\ \vdots & \vdots & \vdots \\ P_{\text{open}_j} & P_{\text{foliage}_j} & P_{\text{ground}_j} \\ \vdots & \vdots & \vdots \\ P_{\text{open}_M} & P_{\text{foliage}_M} & P_{\text{ground}_M} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{2.7}$$

The new α and β could be fitted by employing the non-linear least square algorithm on the set of equations shown above. We empirically set $N = 5$. Note that all links in an orchard at the same cycle share the same α, and β.

The dynamic (4) refers to changes in the 3D profile of the orchard, which affect P_{open} and $P_{foliage}$ in Equation 2.6 for all nodes, but not α and β. To handle this dynamic, it is necessary to remeasure the tree height, trunk height, and canopy width to rebuild the 3D profiles. This is expected to be done every year or even longer.

2.4.4 Workflow of *FLog*

Figure 2.10 summarizes how *FLog* works, including all modules mentioned in the previous section and their input and output. The farmer provides input parameters for generating the 3D orchard, and *FLog* estimates the link quality between any two locations in the orchard based on these parameters. The user can choose a configuration that they think is best suited for their orchard. Specifically, *FLog* requires the following parameters from the farmer: tree profile, layout, and deployment parameters.

1. Tree Parameters: Tree height, trunk height, and canopy width: They are used to build the 3D profile of a single tree. As the growth accumulates, the user may measure these parameters every year or longer.

2. Layout Parameters: The distances between adjacent rows and adjacent trees in one row are required for the layout configuration. These factors will never change. There might be several blank positions for removed dead trees. *FLog* also accepts such input or updates.

3. Sensor and gateway position (optional): If the application requirements dictate the placement of the sensor and gateway, the user should input their designated positions. Or we can determine the optimal position based on *FLog*.

Using the parameters above, *FLog* creates an FFZ between any two locations. It then calculates the portion of foliage, free space, and the ground within this zone and estimates the link quality via Equations 2.2, 2.3, and 2.6. Additionally, *FLog* utilizes a parameter resetting mechanism to tune our model's parameters, which adapts to different environmental dynamics.

We made the assumption that all trees share the same shape. While this generalization might seem oversimplified, it is a necessary approach considering the impracticality of measuring every individual tree's shape. To account for this simplification, we fit each medium's PLE to ensure our model aligns well with the measured signal strength. This helps mitigate the potential impact of our tree shape simplification on system performance.

(a) LoRa node (b) LoRa gateway

Figure 2.11: The hardware implementation of experiments.

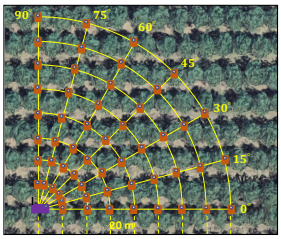


Figure 2.12: The testbed layout in an almond orchard.

2.5 Evaluation

We evaluate the overall performance of *FLog* in Section 2.5.2, Then, we study the performance under different factors in Section 2.5.3.

2.5.1 Experiment Setting

Hardware Implementation

Figure 2.11 depicts the used hardware of LoRa node and gateway. LoRa nodes are hand-crafted with SX1276 Radio [49] on the Arduino Uno host boards [50]. They are equipped with a 3,000 mAh power bank. They work in the frequency band 904.3 MHz. The RAK831 Pilot Gateway [51] is used as the LoRa receiver to receive LoRa packets. It consists of a Raspberry-Pi 3, an RAK831 LoRa Concentrator and a converter board with GPS for routing the signals between the Raspberry and the RAK831. Both nodes and gateways are using omni-directional antennas, with antenna gains of 5 dBi and 3 dBi, respectively. The gateway executes a thread of LoRa Packet Logger [52] that demodulates packets and stores them as comma-separated values (CSV) files. Our model is implemented with Python script on a

PC with an Intel(R) Core (TM) i9-11900KF @ 3.50 GHz CPU with 16 cores.

Benchmarks

We compare the performance of *FLog* with the following multiple baselines.

- *Log-Normal Shadowing Model [42]:* It is calculated by Equation 2.3. We use the collected data to fit PLE in the orchard scenario. It is referred to as "Log".
- *Line-Based Log-Normal Shadowing Model:* It is our strawman solution, which is introduced in Section 2.3. We call it "LLog".
- *Empirical Foliage Loss Models:* They include Weissberger [28], ITU-R [29], COST235 [30], Two-ray ground-reflection model [53] and Okumura-Hata models [26]. All these models that are parameterized or have multiple variants have been carefully evaluated and reported as their best performance.

Performance Criteria.

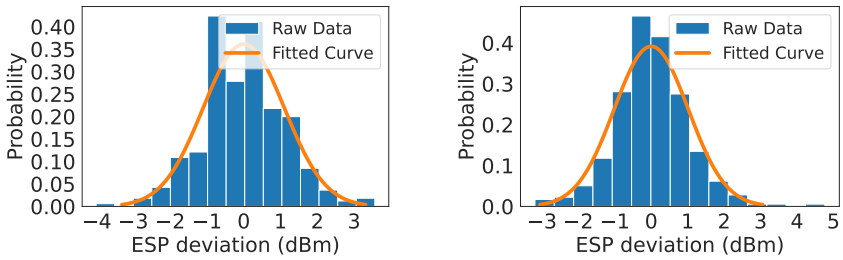
The estimation accuracy is quantified by errors between the predicted ESP and the corresponding actual value for each packet. The unit of errors is dBm.

$$\text{ESP error} = (y_i - \hat{y}_i) \tag{2.8}$$

where y_i is the measured ESP for the i-th received packet and $\hat{y}_i$ is the predicted ESP value.

Experiments in an Almond Orchard

We chose almond orchards as the main focus of our evaluation since they are a critical agriculture industry in the US. In 2021, the US produced 2.5 billion pounds of almonds, with a value of over 7 billion dollars [54]. Moreover, the estimated almond acreage in California was 1.64 million acres, making it the largest producer of almonds in the world [54]. We also tested *FLog* in a walnut orchard, which is another dominant arbor crop.



(a) Environmental noise (60 m) (b) Environmental noise (100 m)

Figure 2.13: The ESP deviation distribution for two different communication distances.

Figure 2.12 shows locations of LoRa nodes and a gateway in an almond orchard with a size of 200 × 200 m^2. In this orchard, trees are planted in a line with 4.88 m distances, and lines are separated by 6.66 m. The almond trees are 6.1 m in height and 2.8m in width. The heights of the LoRa node and the gateway are set as 0.45 m and 10 m. The transmission power is 14 dBm, SF is 10, bandwidth is 125 kHz, and coding rate is 4/5. We collect packets from the spatial and temporal dimensions to evaluate our model.

We do not rely on the GPS to determine the physical locations of LoRa nodes, due to its large localization error in orchards [55]. Instead, we take advantage of the uniform arrangement of trees in orchards. Trees are consistently spaced both along the rows and across columns. By counting the number of trees, we can accurately compute the distance between any two positions. Specifically, we use a 3D Cartesian coordinate system to represent these positions: the x and y axes correspond to the directions along and across the orchard rows, respectively, while the z axis points upwards. The origin of this system is aligned with the gateway's position on the ground. We determine the distance between any nodes and the gateway using the Euclidean distance formula.

Datasets

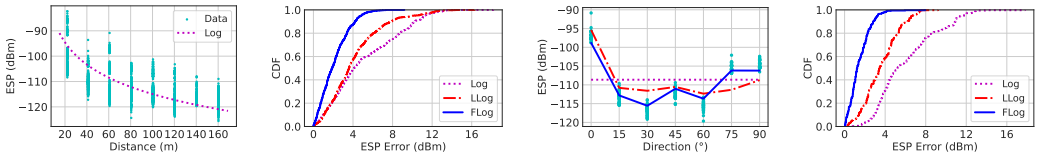
We measure the received packets at a large number of locations in an orchard. We also conduct experiments at some locations for months.

Spatial Dimension Dataset: As shown in Figure 2.12, the gateway is

located in the lower-left corner of one of the 90-degree fan-shaped areas of the orchard. We have deployed LoRa nodes at 56 locations across this fan-shaped area to collect packets. For each received packet, we calculate the ESP by using its RSSI and SNR. The nodes are placed at communication distances ranging from 20 m to 160 m, with a step size of 20 m. We measured seven directions ranging from 0 to 90 degrees, with a step size of 15 degrees. At each location, the nodes transmit an 8-byte packet to the gateway every 1.5 seconds for a duration of 1.5 minutes. The experiment was conducted for a duration of 10 hours in the autumn of 2022, and 2900 packets were collected in total. The ESP ranges from -125 to -85 dBm, corresponding to an SNR range of $[-15, 7]$ dB. The collected data is unavoidably affected by the environmental noise at each location. We compute the average ESP at each location as the reference ESP. The ESP deviation is obtained by comparing the average ESP and the collected ESP of each packet. Figure 2.13 shows that the ESP deviation at two distances generally follows a Gaussian distribution with a deviation of 1.1 and 1.0 dBm.

Given the consistent planting patterns, we deduce that patterns, signal strengths, and propagation characteristics observed in one quadrant will reflect those in the remaining three. Consequently, by strategically positioning our gateway in a corner, we can capture representative data without the need for exhaustive measurements across the entire orchard.

Temporal Dimension Dataset: We collected another dataset that covers four environmental dynamics, as described in Section 2.4.3. Four LoRa nodes are deployed at four locations randomly selected from Figure 2.12 to collect long-term data over a period of four weeks in the spring (January and February 2023). The transmission settings used for collecting the data are the same as those used for the spatial dimension data. We collected data continuously for 24 hours each week, resulting in a dataset of 69,173 packets. The dataset contains three environmental dynamics: (1) Short-term environment variations. (2) Transient weather changes, including temperatures ranging from 26.2 to 65.5°F and precipitation ranging from 0 to 0.15 inch/hour. (3) Foliage density changes. In the final week of the four-week period, almond trees started to bloom with flowers. By combining this with the



(a) ESP *vs.* distance (b) Distances and directions (c) ESP *vs.* direction (60 m) (d) All directions at 60 m

Figure 2.14: The overall performance of three models for different communication distances and directions. Figure (a) does not draw the fitted curve of the LLog and *FLog*, it is because they have different estimated values at different directions for one communication distance. Figure (c) plots the estimated value in different directions when the horizontal distance between LoRa nodes and the gateway is 60 m. The curve of the Log model is a horizontal line, which could not handle the direction.

spatial dimension data, foliage density can be classified into three categories: trees with dense leaves, trees without leaves, and trees with flowers.

Finally, dynamic (4), the long-term foliage shape changing, was emulated by collecting data in another almond orchard, where the almond trees have a height of 4.5 m and width of 2.6 m, which corresponds to changing trunks and branches, compared to the almond trees shown in Figure 2.12.

Fitting and Testing Data: In the spatial dimension of the data, we used measurements collected from four randomly selected distances to fit the parameters in our path loss models, such as α and β. These measurements are referred to as fitting data. The data from the remaining four communication distances were used as testing data. To obtain the optimal parameter values on the fitting data, we employed the least square approximation method. These parameter values were then used to estimate the ESP on the testing dataset.

2.5.2 Overall Performance

We evaluated the performance of all modes on both spatial and temporal dimension data. In order to measure the impact of different distances on estimation error, we combined all directions' data at one distance. For a specific distance, we

studied the performance of all models in different directions with an interval of 15°.

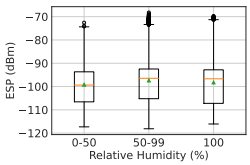
Spatial Dimension

In the fitting and testing data, four communication distances were randomly selected from eight distances in our testbed. Therefore, there are $\binom{8}{4} = 70$ combinations to select the fitting and testing data. For each set of fitting data, we obtained corresponding fitted values of α and β. After analyzing all 70 combinations, we found that the mean value of α was 1.98 with a standard deviation of 0.11, while the mean value of β was 5.07 with a standard deviation of 0.17. We report all the estimation errors for these 70 combinations in Figure 2.14(b).

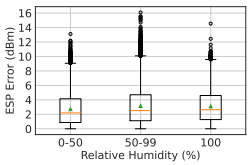
Communication Distance: Figure 2.14(a) presents the estimated curves of Log with changing communication distances. We do not draw the curve of the LLog and *FLog* because they have different estimated values in different directions for a given distance, while the Log model has one average value for each distance. We can observe a trend wherein the measured ESP correlates linearly with the logarithm of the distance. Figure 2.14(b) quantifies the estimation error for different models in the CDF curve (cumulative distribution function). The average error of *FLog* is only 2.85 dBm, showing great performance improvement over other models in terms of link quality estimation. In particular, *FLog* decreases the ESP estimation error of Log and LLog by 42.7% and 35.2%, respectively. This is because Log does not consider the influence of direction on the received signal power and can only estimate the signal power based on the communication distance in Equation 2.3.

Although LLog is aware of the ESP variation in different directions, it only considers the obstacle of lines between the node and gateway. However, the signal is concentrated in the FFZ based on the diffraction theory [42] and our preliminary experiments in Section 2.4. Therefore, *FLog* developed a more reasonable model to consider the shadowing effects in the FFZ.

Direction: Figures 2.14(c-d) present the fitted curves of all models and the estimation errors with different directions at a horizontal distance of 60 m. It is observed that Log is unaware of directions and thus provides a horizontal line as

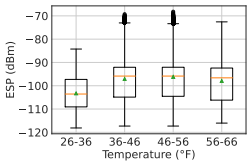


(a) ESP *vs.* humidity

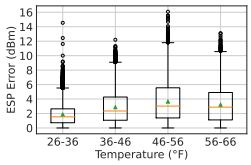


(b) ESP error *vs.* humidity

Figure 2.15: Impact of the dynamic (2): humidity.



(a) ESP *vs.* temperature



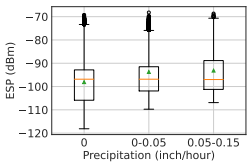
(b) ESP error *vs.* temperature

Figure 2.16: Impact of the dynamic (2): temperature.

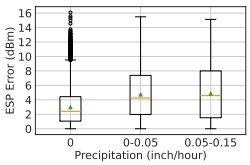
the estimated curve, which remains unchanged for different directions. Although LLog considers the influence of direction, it produces unstable performance. At direction 75°, LLog produces the largest error, that is, 5.93 dBm. This is due to the fact that LLog considers only the shadowing in the line of the propagation path, which results in overestimating or underestimating the ESP of the received signal.

Empirical Foliage Loss Models: Table 2.4 presents the average estimation ESP error on the test data for five different foliage loss models. These foliage models provide worse prediction accuracy compared with the Log, LLog, and *FLog* models. There are three reasons for the poor performance.

Firstly, these models are developed using different frequency channels and have different attenuation in the same situation since the wavelength is different. Thus, the empirical models cannot be directly applied to our scenarios. Secondly, these loss models have no adjustable parameters to adapt to different scenarios. However, the Log-Normal Shadowing model can adapt to various situations by fitting its parameters, such as PLE. Hence, it can perform well if we fit it with the collected data on the specific case, compared with those path loss models with fixed parameters. Thirdly, those models cannot handle the ESP deviation in different directions. This is similar to the Log-Normal Shadowing model. They can only consider the effect of the communication distance on the signal strength, but not on other factors such as directions.

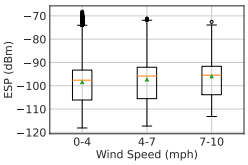


(a) ESP *vs.* precipitation

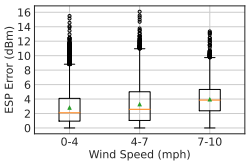


(b) ESP error *vs.* precipitation

Figure 2.17: Impact of the dynamic (2): precipitation.



(a) ESP *vs.* wind speed



(b) ESP error *vs.* wind speed

Figure 2.18: Impact of the dynamic (2): wind speed.

Table 2.4: The ESP estimation error for other models.

Models	Weis.	ITU-R	COST235	O-H	Two-ray
ESP Error (dBm)	41.6	111.2	87.4	12.7	72.3

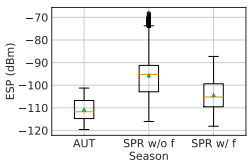
Temporal Dimension

We use temporal dimension data to test the models' performance on the three different environmental dynamics. We apply our parameter resetting mechanism to all path loss models, which uses the most recently received packets to calibrate models' parameters. Note that the environmental dynamic (1) was addressed by averaging the received ESP of multiple packets.

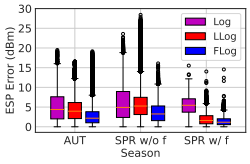
Environmental Dynamic (2) - Transient Weather Changes: We first obtain the temperature, precipitation, wind speed, and humidity from a weather station [56]. Next, we group the data based on the value range of corresponding weather factors at the time of collection. Figures 2.15 2.16, 2.17, and 2.18 illustrate the ESP values and ESP estimation error of *FLog* under different weather conditions.

Figure 2.16 depicts that ESP at 36 °F and 66 °F is higher than that at 26 °F and 36 °F. Figure 2.17 shows the discrepancy in different precipitation levels. They are caused by waterproof cover deformation. There is no explicit difference if the data trace of that node (the blue line in Figure 2.26) is removed.

As shown in these four figures, models achieve no increased ESP prediction error for different temperatures, humidity, wind speed, and precipitation levels. *FLog* always outperforms Log and LLog in any weather condition groups, with

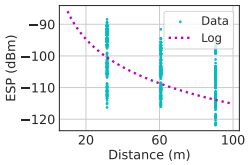


(a) ESP *vs.* season

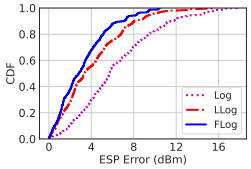


(b) ESP error *vs.* season

Figure 2.19: Impact of the dynamic (3): foliage density.



(a) ESP *vs.* distance



(b) ESP error distribution

Figure 2.20: Impact of the dynamic (4): foliage shape.

an average improvement of 39.8% and 41.5%, respectively. This indicates the effectiveness of our parameter resetting mechanism in adapting to dynamic (2). It is reasonable because the proposed mechanism can adjust the values of parameters α and β based on the recently received packets. We analyzed the received ESP of LoRa signals under different temperature and precipitation conditions. Our findings indicated that the difference in signal power at different temperatures or precipitations was small. This small difference may be attributed to the effects of atmospheric and vapor molecules on electromagnetic waves, which become more apparent at frequencies exceeding 10 GHz. However, the absorbance for 900 MHz bands at short distances ($\leq$ 50 km) is negligible [57, 58].

Environmental Dynamic (3) - Foliage Density Changes: In the spatial and temporal dimension dataset, foliage density has three statuses: trees with dense leaves, trees without leaves, and trees with flowers, which are referred to as "AUT", "SPR w/o f", and "SPR w/ f", respectively. We use our proposed parameter resetting mechanism to adapt to this dynamic. Figure 2.19 illustrates the received signal power and ESP estimation error for the three foliage density statuses. Figure 2.19(a) indicates that foliage density significantly impacts the received signal power. *FLog* handles foliage density variations by continuous parameter refitting scheme. In particular, Figure 2.19(b) shows that *FLog* is capable of adapting well to this dynamic, with estimation errors of 2.85, 3.61, and 1.42 dBm across three foliage density states.

Environmental Dynamics (4) - Long-Term Foliage Shape Changes: We collected data in another almond orchard to evaluate the impact of growing

stages on our proposed method, which we refer to as environmental dynamic (4). The transmission settings of the nodes were the same as those used in the testbed, and we collected data at three communication distances with seven different directions (0, 15, 30, 45, 60, 75, and 90). We used the parameters fitted from the testbed data to estimate the link quality in the new orchard. To reconstruct the 3D structure, *FLog* requires measuring one tree, as described in Section 2.4.4. The new 3D structure will be used to calculate P_{open}, $P_{foliage}$ in Equation 2.6.

The results, shown in Figure 2.20, indicate that *FLog* achieved an average estimation error that was reduced by 48.3% and 16.2%, compared to Log and LLog, respectively. Figure 2.20(a) shows that the Log model could roughly fit the measured signal over the communication distance. However, *FLog*'s performance was comparable to the overall performance reported in Section 2.5.2, with an average ESP of 2.51 dBm compared to 3.42 dBm for Log. This confirms that *FLog* is insensitive to different almond fields and growing conditions.

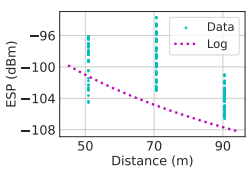
Figure 2.20(b) illustrates that LLog achieved a similar ESP error to *FLog*. This could be attributed to the smaller shape of trees in this orchard, and LLog generates a smaller shadowing media portion, closer to that of *FLog*. The statistics indicate that the two methods had comparable estimations of $P_{foliage}$, averaging 0.17 and 0.24. Hence, LLog delivers a performance comparable to *FLog*.

2.5.3 Parameter Study

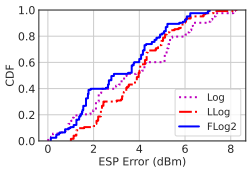
The results of the previous experiments demonstrate the effectiveness of our model in almond orchards. We also conducted further studies to evaluate the performance of our system under various experimental settings. It is worth noting that the collected data was solely used to test all models with parameters that were fitted from previous fitting data.

Impact of Orchard Species

Different orchard types can introduce varying shadowing effects, making it logical to expect distinct β values for each species. Recognizing the distinct foliage

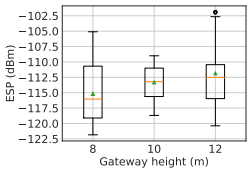


(a) ESP *vs.* distance

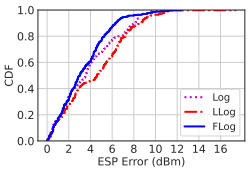


(b) ESP error distribution

Figure 2.21: The impact of the orchard species.



(a) ESP *vs.* height



(b) ESP error distribution

Figure 2.22: The impact of the gateway height.

shape and density differences between walnut and almond trees, we selected the walnut orchard to ascertain the adaptability of *FLog* in such varied settings.

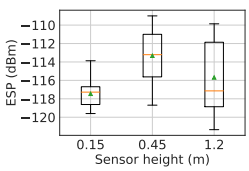
To fit parameters in a walnut orchard, we collected data at a horizontal distance of 50 m and 90 m in three directions (30°, 60° and 90°). We then utilized three path loss models with newly fitted parameters to estimate the ESP in the same walnut orchard, the evaluation is conducted on the data collected at 70 m in three different directions (30°, 60°, and 90°). All other transmission settings remained the same as the testbed.

As shown in Figure 2.21(a), similar trends were observed in the walnut orchard as in the almond orchard, *i.e.*, the ESP of the received signal decreased with increasing communication distance. Figure 2.21(b) reports the CDF of the ESP estimation error for different models, which shows that *FLog* provides the highest estimation accuracy. The *FLog* reduced the error by 15.4% and 18.0% for the Log and LLog models, respectively.

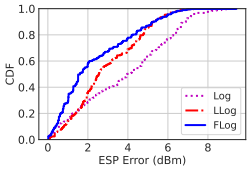
Impact of Gateway Height

We investigated the effect of gateway height on performance gain by conducting experiments at varying heights. The maximum height of our tripod was 12 m, and we set the gateway height at 8 m, 10 m, and 12 m, with a fixed horizontal distance of 80 m between the sensor and the gateway. We collected data in four different directions at each gateway height (0°, 30°, 60°, and 90°).

The results, shown in Figure 2.22(a), demonstrate that the ESP of the received signal increases as the gateway height increases from 8 m to 12 m, which matches

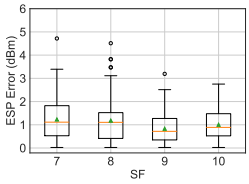


(a) ESP *vs.* height

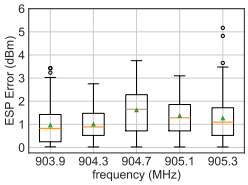


(b) ESP error distribution

Figure 2.23: The impact of the sensor height.



(a) ESP error *vs.* SF



(b) ESP error *vs.* frequency

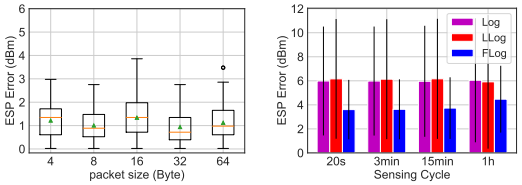
Figure 2.24: Impact of the transmission parameters.

the expectation. Figure 2.22(b) shows that *FLog* could fit the measured ESP well at all gateway heights, compared to the other two methods. Besides, LLog performed worse than Log because $P_{foliage}$ in LLog overreact to the change of gateway height *e.g.*, $P_{foliage}$ was incorrectly estimated at an average of 0.541. Therefore, LLog performed worse than the other two models.

Impact of Sensor Height

We also investigated the effect of sensor height on performance gain. Placing the sensor on the ground can significantly affect signal propagation, so we varied the sensor height to observe the impact. We tested heights of 0.15 m, 0.45 m, and 1.2m, with a fixed horizontal distance of 80 m between the sensor and the gateway. We collected data in four directions at each sensor height, *i.e.*,, 0°, 30°, 60°, and 90°.

Figure 2.23(a) shows that increasing the sensor height from 0.15 m to 0.45 m improved the received signal's ESP. However, the signal strength decreased when the sensor height was 1.2m, likely due to the almond tree's leaves blocking the signals at the LoRa nodes. Therefore, we set the sensor height to 0.45 m. Figure 2.23(b) displays the estimated ESP error for the three methods, with *FLog* performing the best compared to Log and LLog models. Notably, *FLog* reduced the average ESP error by 27.1% and 12.0%, respectively.



(a) ESP error *vs.* packet size (b) ESP error *vs.* sensing cycle

Figure 2.25: Impact of packet size and sensing cycle.

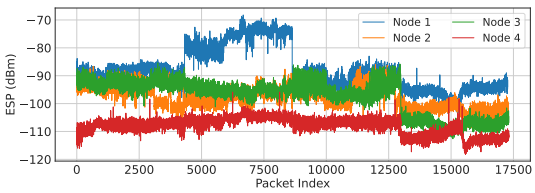


Figure 2.26: The received signal power at four nodes over a period of four weeks.

Impact of Transmission Parameters

We conducted experiments to study the effects of various parameters on our model. Theoretically, transmission power, antenna gain, frequency, and SF do not have a direct impact on the path loss exponent, which is what the link quality models aim to estimate. The packet size may slightly affect the calculation of SNR on the hardware and thus lead to a variation in ESP. Figure 2.24 and Figure 2.25(a) show the estimation error for different SFs, frequency bands, and packet sizes, and the results demonstrate that none of these parameters have a significant effect on the ESP.

Impact of Sensing Cycle

We utilize the most recently received N packets to perform the parameter resetting mechanism, where N is set to 5. The performance of this mechanism may be influenced by the sensing cycle, which is the time between two adjacent transmitted packets. In our testbed, each node sends 45 packets every 15 minutes with a 20-second interval. To simulate a 15-minute sensing cycle, we use the first, 46th, 91st, 136th, and 181st packets to recalibrate our model's parameters, which are then used to predict the link quality of the 226th packet. This allows us to employ our parameter resetting mechanism at different sensing cycles. Figure 2.25(b) shows the error bars with different sensing cycles for three models, depicting the standard deviation from the average. We can find that the duration of the sensing cycle has a minimal impact on the performance of our model. Specifically, compared to a

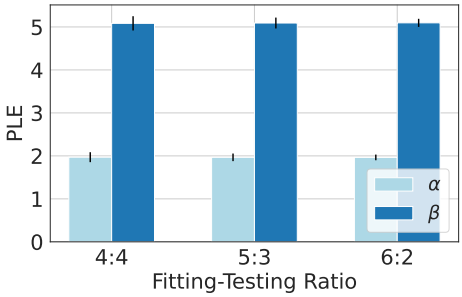


Figure 2.27: The values of parameter of α and β under different ratios of the fitting and testing data.

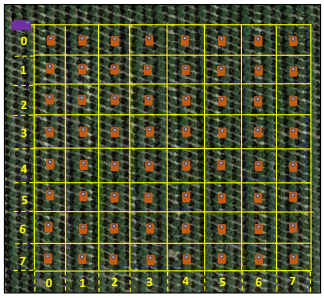


Figure 2.28: Illustration of collecting data in a new almond orchard for gateway coverage estimation.

sensing cycle of 20 s, the estimation errors of *FLog* increase by 0.87%, 4.03%, and 9.56% with sensing cycles of 3 min, 15 min, and 1 hour. Figure 2.26 illustrates the received signal power over a period of four weeks. It can be observed that the link quality at a particular location in the orchards remains relatively consistent, indicating that changing the sensing cycle has minimal impact on the estimation error.

It should be noted that for node #1, there were two significant changes in ESP at packet indexes 4,316 and 8,633, respectively. The first change occurred due to a gusty wind that caused the waterproof cover behind the antenna to lift up like a reflection mirror, resulting in a significant increase in ESP. The cover was manually recovered at the second change point, which caused the ESP to return to normal levels. Although this incident caused an increase in errors, its impact on the comparison between different sensing cycles is negligible. This is because two outliers would only affect five estimations in our parameter resetting mechanism. Secondly, during the last week of the four-week period, almond trees began to bloom with flowers, resulting in a significant decrease in the received LoRa signal power for all four nodes.

Impact of Ratio Between Fitting and Testing Dataset

In the previous sections, the default ratio is 4:4. We evaluated our model with ratios of 5:3 and 6:2. The settings of 5:3 and 6:2 had a similar performance to the ratio of 4:4, with a negligible difference in average estimation error of 0.09 dBm

and 0.04 dBm. We also reported the obtained values of the α and β in Figure 2.27. The error bar represents one standard deviation from the average. We find that the fitted values of α and β under different ratios have no differences. Therefore, we only report the results for the ratio of 4:4.

2.6 Related Work

Modeing LoRa Link Quality: There have been several empirical studies conducted on the LoRa link quality [26, 59, 60, 22, 23, 61, 62, 27, 63, 64]. For example, Adrian et al. [65] select locations of sensors and gateways to provide a LoS signal propagation path in FFZ. Demetri et al. [26] utilize remote sensing to quantitatively analyze the composition of land covers along LoRa links. Based on the dominant land-cover type along the link, they decide the right version of the Okumura-Hata model from two variants [66, 67, 42]. However, their method is not applicable in orchard scenarios. 1) Since orchards typically only have one type of land cover (*i.e.*, trees), their method will ignore local spatial shadowing features such as the amount of space blocked by trees between the sensor nodes and the gateway. 2) The Urban model has deterministic parameters that are empirically fitted for cellular signals, which are typically received by base stations with high antenna heights, unlike LoRa gateways used in orchards.

Extensive measurements [60, 59, 61, 23, 68] have been conducted in various environments, such as indoor, urban, rural, and multi-floor buildings. Based on these measurements, empirical path loss models have been derived. Although these models perform well on their collected data, they do not consider the unique features of orchards, *e.g.*, large deviations of the received signal powers in different directions that cannot be modeled via satellite images. *FLog* is a model based on FFZ theory, which can capture the wireless signal propagation mediums in orchards through 3D modeling.

Foliage Effect on Wireless Signals: The foliage has been observed to have a pronounced influence on link quality [69]. Numerous empirical foliage loss models have been introduced, including the Weissberger model [28], the ITU

Recommendation (ITU-R) model [29], and the COST235 model [30]. However, implementing these models in orchards poses challenges for several reasons. Firstly, they operate over different frequency bands, resulting in varying attenuation levels. Secondly, a majority of these models are deterministic, lacking flexible parameters to suit diverse orchard scenarios. Thirdly, their representations of foliage are rather simplistic (*e.g.*, assumed to be uniformly distributed in space), making them ill-suited for handling significant signal power variations across different directions.

The two-ray ground-reflection model [53] is tailored for open environments with a direct LOS path between the transmitter and receiver. However, introducing foliage complicates the signal propagation paths, rendering this model unsuitable for the orchard context. Contrarily, *FLog* gauges link quality by taking into account both the surrounding trees and the open spaces within FFZ.

LaPS [70] leverages remote sensing and LiDAR to locate trunks in forests and profile the canopy size of each tree. Then it uses multi-regression to get the shadowing parameters of each tree in the rectangular area along the path. *FLog* advances by utilizing the FFZ to judge the involvement of each canopy and mitigating the LiDAR scanning by leveraging the tidy layout of orchards.

Empirical Studies on LoRa Links: Bor *et al.* [23] discovered that in rural areas, communication distance could extend up to 3 km if the gateway is placed on a 100 m tall building. Centenario *et al.* [59] quantified the number of LoRa gateways required for city-wide communication coverage and found that LoRa coverage could span up to 2 km when the gateway is placed in a high-building area. This paper presents experimental results that demonstrate a significant reduction in communication distances in orchards due to the large attenuation of signal strength when gateways are deployed at limited heights. Cattani *et al.* [16] conducted experiments in indoor, outdoor, and underground settings and found that link reliability varied with environmental changes such as temperature. Voigt *et al.* [71] conducts simulations to analyze the influence of the in-network interference [72] from the other LoRa networks working over the same deployment area. The authors conclude that we can deploy additional gateways to make sure that all LoRa nodes are within the communication distance of one of them. Orestis

et al. [24] constructs the coverage probability by considering the ALOHA protocol, where the collisions happen if two packets with the same frequency channel and SF arrive at the gateway simultaneously. *FLog* develops a path loss propagation model for the orchard, paralleling research in these models.

Recent LoRa Advances: Recent LoRa research improves reliability with mobile beamsteering antennas and diversity techniques that salvage weak packets [73, 74, 75, 76], scales network capacity by reexamining logicalchannel orthogonality, concurrent multichannel gateways, hybrid CSS/DSSS modulation, fullduplex links, and interferencetolerant coding [77, 78, 79, 80, 81, 82], and optimizes performance and security through oneshot compressed aggregation plus antijamming and covertchannel defenses [83, 37, 84]. LoRa physicallayer innovation has recently focused on advanced signal designs, from ChirpTransformers versatile, learnable encoding for mainstream IoT radios [85] to Prisms nonlinearchirp backscatter that boosts raw throughput [86]. The community has standardised evaluation with NELoRaBench, a public dataset and metrics suite that stresstests neural demodulators across channel conditions [87]. Neural weaksignal recovery has progressed via SRLoRa, which fuses multigateway receptions to achieve superresolution demodulation [88]. Complementarily, LoRaTrimmer condenses symbol energy through adaptive chirp trimming, further improving robustness at low SNR [89]. Another key trend is enabling ground-aerial and ground-space communication [90, 91], which further push the boundary of LoRa and LoRaWAN. Therefore, the technical stack of LoRa is more and more comprehensive to be widely adopted.

Wireless Signal for Ubiquitous Sensing: Motivated by cross-media communication where signal attenuation/polarization is the major challenge [92, 5, 4], media sensing also becomes practical when researchers finds that the amplitude, delay and phase can tell the properties of the media itself. For instance, soil moisture, texture and organic carbon ratio can be told by combining LoRa and visible and near-infrared (VNIR) spectrums [93], while mmWave enbales fine-grained leafwetness detection in the field [94, 95, 96, 97]. Fullbody posture can be reconstructed with passive PIR arrays [98]. Devicefree sensing for indoor localization and activity recognition has progressed with SCALINGs plugandplay tracking

framework [99] and M4Xs syntheticdataaugmented metric learning for crossview RF HAR [100]. Noncontact vitalsign monitoring is advanced by practical signalquality detection [101, 102, 103], RFQs unsupervised assessment scheme [104], DeepVSs deeplearning pipeline [105], an FMCWradar configuration study [106], robust multiuser VitalHub system [107], and piezoaided speaker authentication against spoofing [108, 109].

2.7 Conclusion

This paper presents *FLog*, a propagation model for LoRa signals in orchards. We first investigated the propagation characteristics of LoRa signals in orchards, revealing strong diffraction caused by trees and the ground. To capture these features, *FLog* estimates link quality by calculating the PLE in the Log-Normal Shadowing model for any pair of nodes and gateway using the FFZ theory. Extensive experiments demonstrate the effectiveness of our model.

Chapter 3

Time-series Forecasting Control for Agricultural Managed Aquifer Recharge

The rapid decline in groundwater around the world poses a significant challenge to sustainable agriculture. To address this issue, agricultural managed aquifer recharge (Ag-MAR) is proposed to recharge the aquifer by artificially flooding agricultural lands using surface water. Ag-MAR requires a carefully selected flooding schedule to avoid affecting the oxygen absorption of crop roots. However, current Ag-MAR scheduling does not take into account complex environmental factors such as weather and soil oxygen, resulting in crop damage and insufficient recharging amounts.

This paper proposes *MARLP*, the first end-to-end data-driven control system for Ag-MAR. We first formulate Ag-MAR as an optimization problem. To that end, we analyze four-year in-field datasets, which reveal the multi-periodicity feature of the soil oxygen level trends and the opportunity to use external weather forecasts and flooding proposals as exogenous clues for soil oxygen prediction. Then, we design a two-stage forecasting framework. In the first stage, it extracts both the cross-variate dependency and the periodic patterns from historical data to conduct preliminary forecasting. In the second stage, it uses weather-soil and flooding-soil causality to facilitate an accurate prediction of soil oxygen levels. Finally, we

conduct model predictive control (MPC) for Ag-MAR flooding. To address the challenge of large action spaces, we devise a heuristic planning module to reduce the number of flooding proposals to enable the search for optimal solutions. Real-world experiments show that *MARLP* reduces the oxygen deficit ratio by 86.8% while improving the recharging amount in unit time by 35.8%, compared with the previous four years.

3.1 Introduction

Groundwater is the water found underground in the spaces of soil, sand, and rock. It is an important resource for agricultural stability, for example, providing up to 60% of the water supply in dry years for California [110]. Due to recurring droughts in recent years, groundwater pumping has increased significantly, exceeding the natural recharge rate and leading to insufficient water supply in underground aquifers [111]. This poses a threat to the food security of many regions [112, 113]. Therefore, careful management and conservation of groundwater resources are highlighted in many regions globally. In California, the Sustainable Groundwater Management Act (SGMA) has been introduced, aiming to achieve a balance between extraction and recharge within the next 20 years [114].

Managed Aquifer Recharge (MAR) is a technique used to redirect excess surface water into underground aquifers during raining seasons, helping to replenish groundwater sources and reduce the impacts of excessive water withdrawal. This method is particularly adopted in areas close to rivers where the land is primarily used for farming, known as Agricultural Managed Aquifer Recharge (Ag-MAR). Ag-MAR has been recognized as an effective method for restoring water levels in depleted aquifers, as well as for other advantages such as conditioning the soil before planting seasons and enhancing habitats for bird populations [112, 115]. Its process is illustrated in Figure 3.1.

The effective implementation of Ag-MAR in general remains a difficult problem. When farmland is flooded, the water reduces the amount of oxygen dissolving into the soil and therefore reduces the amount of oxygen accessible to the roots of the

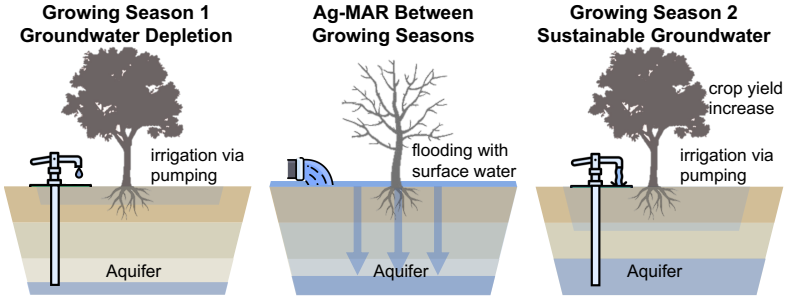


Figure 3.1: The benefits of applying Ag-MAR.

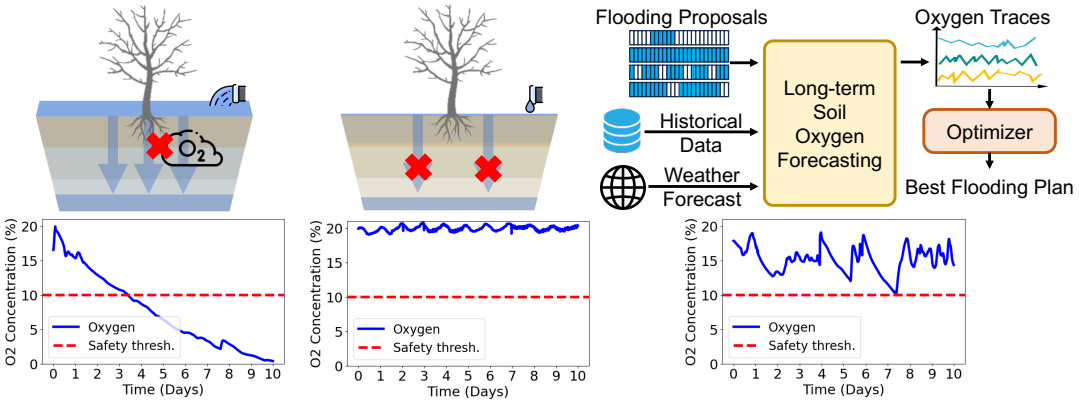


Figure 3.2: The illustration of oxygen fluctuation in continuous flooding, regular irrigation and intermittent flooding.

crops. Crops have specific tolerance thresholds for the soil oxygen level; if the level drops below the threshold, the crop root starts to decay, significantly damaging the crops. Consequently, Ag-MAR needs to optimize two objectives at the same time, i.e., maximizing the amount of water recharged to the underground aquifer while keeping the soil oxygen level above a predefined threshold. We show examples of the soil oxygen level trends under three different scenarios in Figure 3.2. From left to right of the figure: first, if the flooding continues for an excessive period, the soil oxygen level continuously drops over the tolerance threshold for the plant root, resulting in root rot and future yield reduction. Second, if the water amount is slight, like moderate rain and regular irrigation, it may only achieve a balance with evapotranspiration (water lost to air due to evaporation), resulting in insufficient aquifer recharge. Finally, the optimized solution is to flood on an intermittent cycle, alternating between substantial flooding and drying periods so that the oxygen can diffuse into the soil.

Ideally, such a schedule needs to take multiple pieces of information, e.g., current soil oxygen level, future weather, and soil type, into consideration. Because water takes a certain time to sink through different soils, the effect of flooding will not be detected immediately but will result in a delayed, long-term effect. Consequently, it is essential to model and predict the effect of each flooding in the long term in order to choose the optimal flooding schedule. However, an accurate prediction and control method that considers all the above information remained out-of-reach.

Analyzing our four-year real-world dataset in alfalfa fields[1] revealed two key observations that motivate our design, which we will describe in detail in Section 3.2. In short, we found out that:

- Soil oxygen exhibits a multi-periodicity pattern, modulated by environmental conditions and flooding actions.
- Due to the strong causality between weather, soil water content and soil

[1]Alfalfa is an classical crop for Ag-MAR as it does not require any nitrogen fertilizer after establishment, gaining all N from biological N_2 fixation and root uptake. It has minimal nitrate leakage risks. Its safety threshold of oxygen concentration is 10%.

oxygen, external weather forecast can boost the oxygen prediction.

In light of these observations, we devise *MARLP*, a Model Predictive Control (MPC) system for Ag-MAR. The core is a Transformer-based long-term forecasting model that features a two-stage learning scheme. First, it integrates cross-variate and multi-periodic learning, enabling the preliminary self-consistent multi-variate forecasting. Then, the weather forecast data is utilized as an external clue to calibrate the oxygen prediction via a causal-aware module. By combining them, the final oxygen prediction adapts well to the fluctuating rhythms of environmental factors.

This predictive capability sets the stage for the subsequent MPC workflow, which provides the predictive outputs to any flooding proposals for days in the future. The optimizer can accordingly choose the best flooding strategy that both mitigates oxygen-deficit risks and maximizes the recharging to the underground aquifer. However, due to the long forecasting window, the total number of flooding proposals are 2^{720}, which can not be brute-force searched and is too large to apply stochastic optimizing techniques. To this end, we propose a domain-specific heuristic planning module. The number of proposals has been reduced to hundreds, making it practical for the system loop to infer the oxygen consequences of them in real-time.

To demonstrate the effectiveness of *MARLP* to predict oxygen variations and schedule flooding actions, we conduct statistical comparisons on past datasets, real-world control trials, and large-scale simulations. The experimental results show that the proposed algorithm and scheme outperform the state-of-the-art models and can provide practical and trustworthy decisions.

Our contributions are summarized as follows:

- We formulate the Ag-MAR optimization as a long-term model predictive control problem and design a customized time-series forecasting model that utilizes the external predictive input and extracts the periodicity features of data traces.

- We propose *MARLP*, an MPC workflow based on the model. To handle

the large action space, we devise a heuristic planning scheme, making the forecasting-based search practical.

- We conduct extensive experiments to demonstrate the effectiveness of *MARLP* on both recharging amounts and safety warranty.

3.2 Background

3.2.1 Ag-MAR Optimization

Ag-MAR is the most applied MAR scheme for two primary reasons: 1) The nearby river areas are usually fully plotted with fields, and applying water to these fields enables minimal water transportation; 2) The riverside fields typically represent the lowest points in the area, ensuring minimal lift work. For the resource-demand mismatching area, Ag-MAR is conducted during the off-season when the surface water flow is adequate [116], and crops are not in their active growth phase.

Ideally, we should flood as much as possible into the field, but the excessive ponding will cause oxygen-deficit and root rot[117]. To coordinate these contradicting objectives, the water usage must be intermittent, so that the soil can dry off and the oxygen level can take time to recover, as shown in Figure 3.2. The control objective is to maximize the amount of water recharged while maintaining a healthy oxygen level for the root zone. So this problem could be formulated as an optimization problem while the output is a control decision series, indicating whether and how much water to apply at each timestamp:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in T} (f_i F + p_i - ET_i - \Delta S_i) \\ \text{subject to} \quad & O_i > O_{safe}, && \forall i \in T, \\ & f_i \in \{0, 1\}, && \forall i \in T, \end{aligned} \tag{3.1}$$

where f_i is a boolean variable denoting if the flooding is conducted in the i-th time step, F is the flooding gain(mm) per time step, p_i is the precipitation gain(mm), and ET_i is the evapotranspiration loss (ET) in the unit time. ΔS_i is

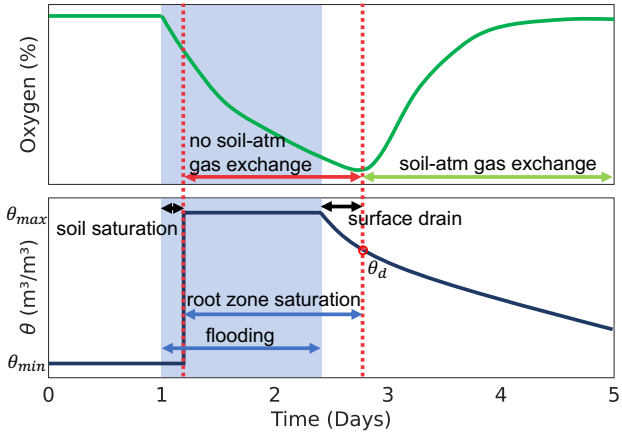


Figure 3.3: The rationale of soil oxygen and water content variation during flooding.

the change in soil storage (mm) (dependent on the available water capacity (AWC) of the soil). Surface runoff is not considered due to the flat field. Note that ET_i is the combination of surface evaporation and plant transpiration. The oxygen level at any time is the accumulated results of past environmental factors:

$$O_j = \mathcal{F}(f_i, p_i, ET_i, \Delta S_i, \text{ for } i \leq j) \tag{3.2}$$

Figure 3.3 shows the principle of how the soil water content ($\theta(m^3/m^3)$) and oxygen percentage change within a flooding event. In the beginning, the water saturates the soil and exceeds the surface, the oxygen level drops because water has squeezed the gas in the soil (t_1 to t_2). This period continues after the valve is turned off. t_3 represents the turning point of the oxygen level, which

In our experimental field, after 10 minutes of turning on the valve, the whole field surface would be flooded, i.e., the soil water content reaches the peak value, and the oxygen starts to drop since no gas exchange may happen. Then the flooding lasts for t_s duration, which can be controlled by our strategy.

Why predicting for a long-term? To choose the best flooding strategy that maximizes the water amount while reducing the risks of oxygen-deficit situations, the consequences of applying water should be accurately predicted, until the next time that the oxygen level recovers to the dry level. This recovery process may be interrupted by precipitation and drop.

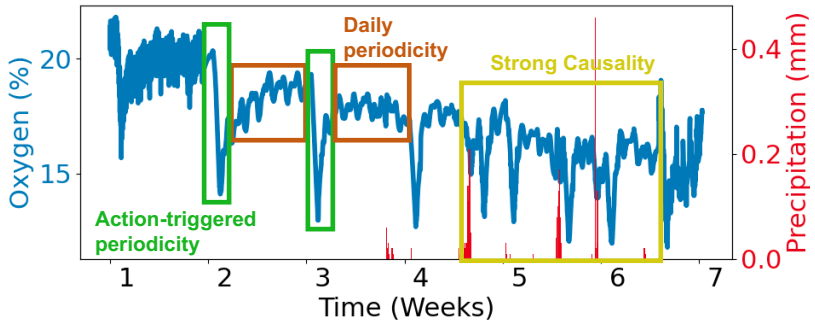


Figure 3.4: Multi-periodicity and causal relationship pattern.

3.2.2 Key Observations

Throughout our five-year in-field Ag-MAR dataset, exampled in Figure 3.4, we've revealed two key observations for helping long-term prediction.

Observation 3.2.1 *Soil oxygen exhibits a multi-periodicity pattern.*

Daily periodicity: This dynamic is subject to the impact of micro-biofunctions, whose behavior is modulated by environmental conditions and our strategic flooding interventions, e.g., elevated temperatures invigorate microbial activities. This leads to a more rapid consumption of oxygen, especially during the day when the temperature is at its zenith and microbial metabolic activities peak. Overall, the multi-periodicity pattern making the prediction task far from interpretable and straightforward.

Action-triggered periodicity: Post saturation, the oxygen level forms periodical v-curves after saturations, first drops and then gradually recovers, as shown in Figure 3.3.

Observation 3.2.2 *Strong causality lies between flooding, weather, and soil oxygen.*

Precipitation impacts soil oxygen levels by saturating the soil, displacing air from pore spaces, and reducing aeration, leading to anaerobic conditions that affect plant and microbial respiration. Thus, predicting soil oxygen levels benefits from considering the causality between weather and soil oxygen levels. Luckily, modern weather forecasting reports that synthesize global atmospheric modeling are

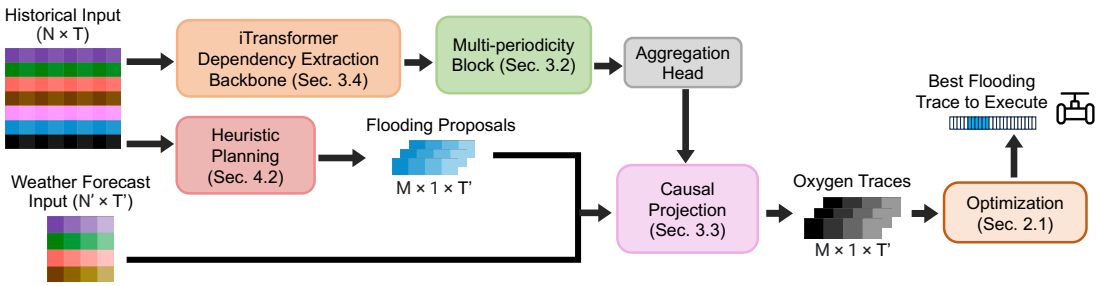


Figure 3.5: The illustration of long-term soil oxygen prediction architecture and how it facilitates control.

becoming more and more reliable, especially in scenarios with sudden and severe rainfall events[118, 119]. They can be used as external clues to facilitate oxygen level inference.

3.3 Long-term Soil Oxygen Prediction

In this section, we introduce a long-term time-series forecasting model customized for soil oxygen prediction.

3.3.1 Model Overview

The model architecture is shown in Figure 3.5, consisting of three major components: dependency extraction backbone, multi-periodicity block, and causal projection. The iTransformer dependency learning backbone [120, 121] extracts the cross-variate and temporal dependencies. The latent space representation is then passed towards the multi-periodicity block to learn the inter- and intra-periodicity features. To this end, a forecasting result would be produced, which is self-consistent among all variates. Then the partially observable future clues, i.e., future flooding and external weather forecasts, are utilized to calibrate the oxygen prediction.

The historical records can be represented as $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\} \in \mathbb{R}^{T \times N}$, with T time steps and N variates. They are soil oxygen concentration, soil water content, flooding history, and weather records, i.e., air temperature, precipitation, humidity, and wind speed. Besides, some partially observable future clues are included,

specifically the flooding plan and weather forecasts for future S time steps $\mathbf{Y} = \{\boldsymbol{x}'_{T+1}, \ldots, \boldsymbol{x}'_{T+S}\} \in \mathbb{R}^{S \times N'}$. With both inputs, we predict the oxygen in future S time steps $\mathbf{z} = \{x_{T+1}, \ldots, x_{T+S}\} \in \mathbb{R}^{S \times N}$.

3.3.2 Multi-Periodicity Extraction

To harness the full potential of periodicity within micro-biofunction data, we adopt a structured approach to first conduct data segmentation and re-organization [122]. The data is segmented according to the first 2 frequency bin of FFT results to isolate periodic components, which is then reorganized into a 2D format that aligns their intra-period indexes,

In this re-organized structure, each column of the tensor represents a discrete time point within a single period, and each row correlates to the same phase across different periods. This configuration allows our system, *MARLP*, to differentiate and learn from both intra-period and inter-period variations. Through this transformation, we overcome the inherent limitations of 1D time series data representation, paving the way for learning the temporal patterns in microbial activities. Inception blocks are implemented for each periodicity pattern, dedicated to extracting and learning the periodicity from a specific time segment period/frequency. The outputs of each block are adaptively aggregated to output the final forecasting.

3.3.3 Causal Projection with Future Clues

Trustworthy weather forecasts and flooding plans are partially observable factors of the future. To combine their insights with history-inferred results, we compare them with preliminary weather and flooding outputs from history-based prediction modules. Considering the temporal self-consistency of $\alpha_{T+n} \in \boldsymbol{x}'_{T+n}$ with all other $\beta_{T+n} \in \boldsymbol{x}'_{T+n}$, if an external clue $\hat{\alpha}_{T+n}$ is trustworthy, $\Delta\alpha_{T+n} = \hat{\alpha}_{T+n} - \alpha_{T+n}$ can be leveraged to fill the confidence gap between β_{T+n} and the groundtruth if causality holds between them.

The causality between temperature, precipitation, soil water, and oxygen holds

intuitively and empirically. Discovering causal relationships within dozens of sequential variates is relatively easy, but it remains an open problem how to leverage the causality to facilitate external clue-aware time-series forecasting.

We divide the variates into three layers. Variates at the upper layer may Granger-cause the lower-layer ones, but it can't be reversed, due to physical rules. After setting the causal layers between variates, we apply Granger causality learning for each pair of variates to learn the parameters:

$$\Delta\beta(t) = \sum_{j=1}^{p} A_j \Delta\alpha(t-j) + \sum_{j=1}^{p} A'_j \frac{d\Delta\alpha(t-j)}{dt} + E(t) \tag{3.3}$$

where $E(t)$ is the residual, A_j and A'_j are linear parameters, we only model the first-derivative causal since it's intuitive for rationales like increased soil water to reduce the oxygen diffusion speed, higher temperature leads to faster soil water emission, etc. During inference, we iterate through layers, up to bottom, to calculate $\Delta\beta$ for all β without external clues, i.e., soil water and soil oxygen. These values are utilized to calibrate raw outputs to achieve a new consistency between soil oxygen forecasting and external clues.

3.3.4 Dependency Extraction among Variates

Understanding dependencies between soil oxygen levels and other variables is crucial for predicting soil oxygen level. While transformer-based methods excel at uncovering dependencies in time series [123], they are less effective at identifying relationships across different variables. To this end, inversed transformer (iTransformer) [120] is adopted as the backbone to learn the dependencies among variables via inversed layer normalization and attention mechanism.

The layer normalization is applied across time steps rather than features, which preserves the distinct temporal dynamics of each variable, ensuring the learning of the patterns inherent to the data. The feed-forward networks serve to distill complex temporal features from each variate, allowing the attention module to work effectively. The attention mechanism of iTransformer is carefully calibrated to work with the tokenized series of variates. It avoids the traditional composite

token format to enhance the model's ability to map out the dependencies among multiple variables.

In-situ Model Update. Considering distribution shifts between regions, fields, and other environmental dynamics, we use the newly collected data to adapt the forecasting model, which enhances the scalability for wide adoption.

3.4 Model Predictive Control

In this section, we integrate the prediction of soil oxygen into our MPC design to guide the recharging actions.

3.4.1 Workflow

The workflow of *MARLP* is shown in Figure 3.2. The heuristic planning generates flooding traces with predefined rules and constraints to propose potential flooding schedules. The long-term oxygen forecasting module then incorporates historical data and weather forecasts to simulate the consequential oxygen trace of all flooding traces. These flooding proposals and the oxygen level predictions are then analyzed by the optimizer according to the optimization objective specified by Eq.3.1. Once the best flooding plan is identified, the optimizer sends the actions to the actuator. The whole process can be conducted again anytime, not necessarily after the plan is finished. We set the control scheduling interval to be 10 minutes in our experiment to balance the timeliness and the computation overhead. Note that agile re-scheduling doesn't conflict with the necessity of long-term forecasting because wrong flooding actions can never be revoked.

3.4.2 Heuristic Planning

The essential part of MPC is to estimate the optimal flooding trace among all flooding proposals, while not consuming an intolerable amount of computation. Traditionally, this is done by adopting stochastic searching methods such as shooting-based methods[124] or cross-entropy methods[125]. Considering a

Figure 3.6: The alfalfa field of 2023 experiments.

planning period that may extend beyond 120 hours, with a decision required every 10 minutes, the number of potential action sequences can reach 2^{720}, which is too large for a stochastic searching method to be effective. To address this, we propose schemes based on an understanding of saturating actions in the system. These schemes are derived from two key principles:

(1) **Flooding Duration Constraint:** To ensure the effectiveness of groundwater recharging, once flooding begins, it must continue for at least a minimum duration before ceasing. This constraint can be represented as:

$$\forall t \in T, \quad (F(t-1) = 0 \wedge F(t) = 1) \Rightarrow (\forall \tau \in [t, t + \Delta t_{\text{min\_flood}}), F(\tau) = 1) \quad (3.4)$$

$$\forall t \in T, \quad (F(t) = 1) \Rightarrow (\exists \tau \in [t, t + \Delta t_{\text{max\_flood}}), F(\tau) = 0) \quad (3.5)$$

Here, $F(t)$ is a binary indicator function where $F(t) = 1$, if flooding is occurring at time t, T, is the total observation time period, and $\Delta t_{\text{min\_flood}}$ is the minimum required duration for continuous flooding.

(2) **Idle Period Duration Constraint:** Between two floodings, the idle interval must be long enough that the oxygen can effectively diffuse, otherwise it should not take the interval, but should grasp the chance to flood more. This constraint ensures an adequate drying period for the soil oxygen recovery:

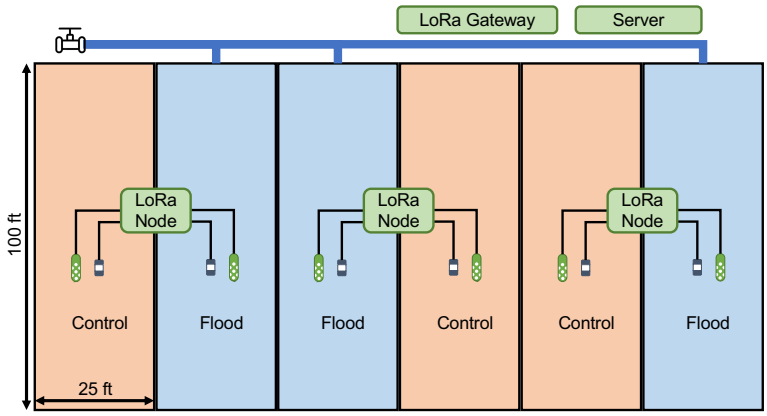


Figure 3.7: The illustration of in-field deployment.

$$\forall t \in T, \quad (F(t-1) = 1 \wedge F(t) = 0) \Rightarrow$$
$$(\forall \tau \in [t, t + \Delta t_{\text{min\_idle}}), F(\tau) = 0) \quad (3.6)$$

In this case, $F(t) = 0$ indicates an idle (non-flooding) period at time t, and $\Delta t_{\text{min\_idle}}$ represents the minimum required duration for the idle period to allow for soil aeration.

By implementing these constraints, we can mathematically define and enforce the necessary spacing between flooding and idle periods within the MPC framework. By filtering out suboptimal action traces, we significantly reduced the size of the search space to several hundred traces, which can be effectively brute-forced to find the optimal trace.

3.5 In-Field Experiment Setup

As shown in Figures 3.6 and 3.13, we build a real experimental testbed to verify the effectiveness of *MARLP* in practical scenarios. It includes sensory data collection, transmission and decision making modules. These components function as a comprehensive in-field experiment system that enables online evaluation of *MARLP*, offering more precise and realistic insights than those obtained from simulations or offline evaluations.

Sensory Data Collection: The oxygen and moisture sensors are placed at critical points throughout the facility. These sensors continuously measure real-time oxygen and moisture data, enabling prompt adjustments to maintain optimal oxygen levels.

Sensory Data Transmission: The collected oxygen and moisture data are transmitted to a central server through LoRa networks, ensuring long-range, low-power wireless communication. We configured the transmission parameters, specifically setting the spreading factor (SF) to 8, to ensure that all measured sensory data are accurately received by our central server.

Inference Server. This server runs the neural networks and MPC algorithm based on continuously received sensory data and crawled weather forecast data from websites for prediction.

System Overhead. The power consumption of each sensor node for sensing and communicating is 64 mW, which can be easily covered by a solar panel, mitigating the need to change batteries. The gateway and server could be purchased as a service during flooding seasons. Overall, the system requires less than $400 to implement and $50/year to maintain, ensuring easy adoption, even for small farms.

3.6 Evaluation

We ask the following questions to evaluate *MARLP* through in-field experiments and large-scale simulations:

RQ1 How effective is *MARLP* in predicting oxygen curves?

RQ2 How effective is *MARLP* in control performance?

RQ3 Can *MARLP* be effectively generalized across different soil types, plant species, and weather patterns?

RQ4 How effective is each design component of *MARLP*?

To answer these questions, we evaluate the predictive capacity on datasets of past years in Section 3.6.1 and assess the control performance of *MARLP* during the

Table 3.1: The collection settings of Ag-MAR datasets.

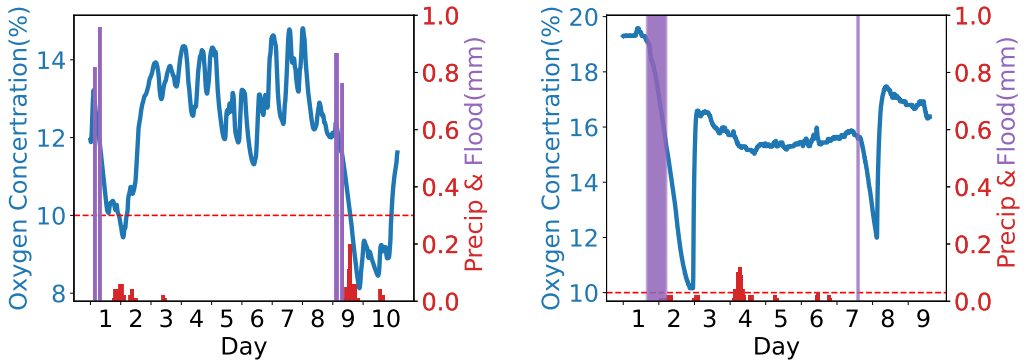
Year	2020	2021	2022	2023	2024
Area(ft^2)	590*280	590*280	590*280	284*132	150*100
Flooding	Const.	Const.	Const.	Const.	MARLP
Duration	2/20-4/2	2/12-3/31	1/19-4/8	2/28-4/6	1/19-4/4
Sequence	6086	6902	11455	5389	11001

in-field deployment in Section 3.6.2. Then, we investigate the performance under different factors in Section 3.6.3, followed by the ablation study in Section 3.6.4.

3.6.1 Predictive Capability

We choose four recent, representative and highly performed time-series forecasting models as baselines: TimesNet [122], PatchTST [126], DLinear [127], and iTransformer [120], covering convolutional, linear, and transformer-based methodologies. We evaluate all baseline forecasting models on five datasets from five years within three fields. The prediction sequence length is set as 720, which represents five days. Given the sensor reading interval of 10 minutes, the forecasting window is 120 hours, which can fully observe oxygen recovery in most cases. Table 3.1 shows dataset statistics, including the collection area, period, flooding strategy, and sequence length. Ag-MAR actions lie between growing seasons, which is roughly from January to April for alfalfa in California, US. From 2020 to 2023, the field is flooded with constant intervals, e.g., once a week. Trials in 2024 are controlled by *MARLP*.

Table **??** reports the MSE, MAE, as well as the mean absolute error of the peak time (PTE) and peak value (PVE) among all forecasting models. The unit of peak time error is an hour. In Table **??**, it is evident that *MARLP* consistently achieves the highest performance (highlighted with red values) when incorporating weather forecast data. This underscores the effectiveness of our long-term soil oxygen prediction algorithm, which is vital for the MPC.



(a) Weekly flooding scheme fails to adapt to the weather conditions.

(b) *MARLP* can avoid potential oxygen deficits while grasping flooding chances.

Figure 3.8: An example of how *MARLP* handles precipitation.

Table 3.2: Control performance comparison.

	Const.	*MARLP*
Oxygen Deficit Ratio	2.72%	0.36%
Recharging Amount (inch per week)	7.640	10.371

3.6.2 In-field Control Experiments

We perform the first in-field deployment of the Ag-MAR control scheme, as illustrated in Section 3.5. The quality of control is evaluated under two factors: **oxygen deficit ratio (ODR)** and **recharging amount**. ODR calculates the ratio of time that the soil oxygen level is below the safety threshold, which should be zero in ideal cases. The best recharging amount may vary according to weather conditions, so the optimal recharging amount can not be asserted, instead, we can only compare the schemes with each other. All in-field experiments are conducted in alfalfa fields, with the safety threshold of oxygen concentration as 10%.

Table 3.2 compares the control performance of *MARLP* with the weekly flooded scheme in 2020-2023 as the baseline. The weekly flooded scheme resulted in an average oxygen deficit ratio of 2.72%, while controlling with *MARLP* yields an ODR of 0.36%. At the same time, *MARLP* increased the recharging amount (inch per week) from 7.64 to 10.371, with a 35.8% improvement.

Figure 3.8 provides an example of comparison between the weekly flooding scheme and *MARLP* to show the reason behind the high effectiveness of *MARLP*. The red and purple bars represent the amount of precipitation and flooding, respectively. Each bar is 10 minutes wide, so the visual areas indicate the total amount of water input. In Figure 3.8 (a), the weekly-based approach ignores the heavy rain forecast after flooding, resulting in the unexpected oxygen deficit below 10% on day 1 and day 8. At the same time, it misses the opportunity to flood during days 3-6, when there is no rainfall. Figure 3.8 (b) shows the performance of *MARLP*. On day 1 and day 2, when the rainfall is negligible, it conducts a 14-hour recharge for 3.1 mm. The oxygen low peak is 10.16%, without exceeding the safety threshold, as depicted by the red dashed line. Knowing the heavy rain on day 4 of the forecast, it discarded aggressive proposals and stopped flooding for 5 days to avoid danger. This shows that *MARLP* can foresee the consequences of each proposal and avoid risky flooding while making full use of flooding opportunities.

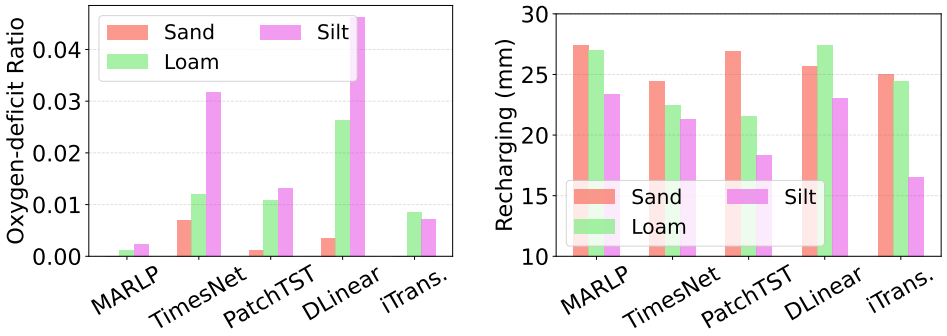
3.6.3 Large-scale Simulation

To evaluate the generalization capability of all prediction models, we utilize a simulator to replicate a diverse array of factors, including soil types, crop species, and regional climate.[2] Specifically, we simulate the water content transition in the soil using HYDRUS [128], a classical simulator for soil flux [129]. Then we model the oxygen diffusion process, root respiration, and microbial respiration based on empirical equations [130]. The evaluation targets the oxygen deficit ratio and recharging amount in the simulation.

Impact of Soil Types

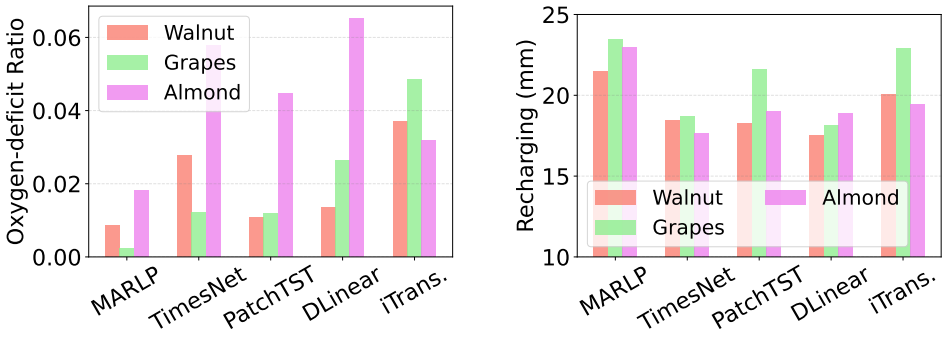
To check the generalizability of *MARLP* on different soil, we mimic three representative soil types: sand, loam, and silt according to the soil texture triangle defined by USDA [131]. We simulate each soil texture as a testbed, where all models are evaluated with MPC architecture the same as *MARLP*. The crops and weather remain the same as in the 2023 in-field experiment. Figure 3.9(a) shows

[2]We didn't conduct in-field A/B tests due to the limits of field resources.



(a) The oxygen deficit ratio. (b) The recharging amount.

Figure 3.9: Control performance for soil types.



(a) The oxygen deficit ratio. (b) The recharging amount.

Figure 3.10: Control performance for crop species.

that *MARLP* achieves the lowest oxygen deficit ratio than the baselines for all three soil textures, and Figure 3.9(b) shows that *MARLP* achieves a high recharging amount at the same time. Although DLinear achieves a higher recharging amount on loam and slit, it performs significantly worse than *MARLP* in terms of oxygen deficit ratio. The superior robustness based on the causality-aware forecasting of *MARLP* holds on other soil types. Note that the soil texture has a significant impact on the general trend of the potential recharging amount because soil with relatively high percolation rates can accelerate the atmosphere exchange process.

Impact of Crop Species

We choose walnut trees, grapes and almond trees, with distinct root densities, depths and oxygen-deficit tolerances [132], to assess how the crop diversity

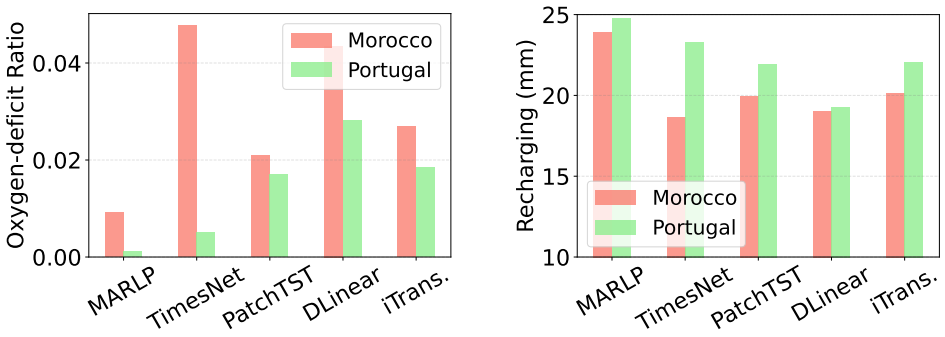
influences the control performance. Walnuts and almonds are less tolerant to flooding than alfalfa, hence they are more susceptible to high oxygen-deficit ratio. Figure 3.10(a) shows that *MARLP* achieves the lowest oxygen-deficit ratio than the baselines for all three crop species, and Figure 3.10(b) shows that *MARLP* achieves a high recharging amount at the same time. Therefore, *MARLP* has the best generalization capability among all methods.

Impact of Regional Climatic

Although California is at the forefront of Ag-MAR adoption, this practice is becoming increasingly popular in other parts of the world that face similar hydrological challenges, particularly those with a Mediterranean climate. We broaden its scope to include the High-Atlas region in Morocco [133] and the Algarve region in Portugal [134], incorporating soil and weather data from these diverse global regions that are actively exploring Ag-MAR. The experimental periods for the simulations are both Feb. 28th - Apr. 6th, 2023, aligning with our real-world experiments in California. We use HYDRUS [128] to simulate the soil dynamics of these two sites and use five models to conduct flooding control separately, with the same goal settings. The results are shown in Figure 3.11. The control performance of all models exhibits drops since the High-Atlas encounters a few sharp precipitations due to high altitude, while Algarve is located by the sea, with stronger daily periodicity. *MARLP* keeps achieving the best control performance. It underscores the generalizability of *MARLP* and its potential to be adapted worldwide.

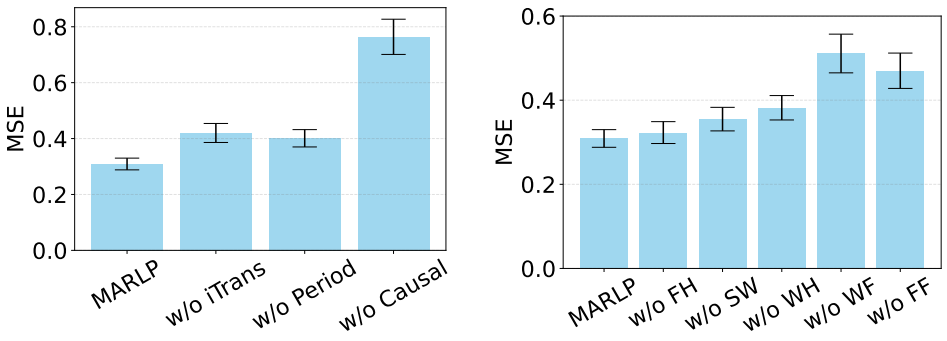
3.6.4 Ablation Study

In this section, we systematically tested the performance of *MARLP* by excluding each design module to evaluate their individual effectiveness. Additionally, we examined how each input variable contributes to the system. These tests revealed the necessity of involving these clues and casual strength between the soil oxygen and each of them. We plot the mean and standard deviation of MSE across all datasets.



(a) The oxygen deficit ratio. (b) The recharging amount.

Figure 3.11: Control performance for climatic features in different regions.



(a) Ablation study for modules. (b) Ablation study for input variables.

Figure 3.12: Ablation study for modules and input variables.

Design module ablations. We construct ablated versions of the model by substituting the iTransformer with a vanilla Transformer, removing the periodicity block and the causal module. Figure 3.12(a) illustrates that the performance of each ablated version diminishes, with the version lacking the causal module experiencing the most significant reduction. This decline is because of the pronounced causality inherent in Ag-MAR data.

Impact of Input Features. Figure 3.12(b) shows the influence of various inputs on the MSE of oxygen forecasting. The abbreviations FH, SW, WH, WF, and FF represent flooding history, soil water content, weather history, weather forecast, and future flooding, respectively. All inputs play a pivotal role in enhancing the final performance, with weather forecasts and future flooding making the most significant contributions. Consequently, optimal forecasting necessitates a synergistic integration of historical data.

Overall, this ablation study affirms the importance of each factor in improving oxygen forecasting and underscores the significance of causal-aware forecasting that incorporates external indicators.

3.7 Related Works and Discussions

Ag-MAR studies. Ag-MAR has been studied in terms of environmental benefits and potential risks[129, 113, 135], the fitness of different soil types, crops, and geological regions. The preliminary results are positive and bring it towards global adoption [136]. However, its current implementation is empirical [137], lacking a standardized and automatic workflow to achieve the best flooding. Based on the oxygen pattern analysis, *MARLP* provides a systematical solution that can work seamlessly with sensor systems, without the need for expert knowledge.

Multi-variate time-series prediction algorithms. The evolution of multivariate time-series prediction algorithms has been marked by significant milestones, starting from the development of the Autoregressive Integrated Moving Average (ARIMA) model [138], advancing through the use of Recurrent Neural Networks (RNNs) [139, 140], and further evolving with the introduction of Transformer [141].

NS-Transformer [142] proposes series stationarization and de-stationary attention to handle the distribution shift and boost the performance of the Transformer on time-series prediction. iTransformer [120] optimizes temporal and cross-variate dependency extraction by swapping 2 modules. Instead of conducting universal forecasting, other works focus on different objectives: TSMixer [143] utilizes the MLP architecture to reduce computing resources. *MARLP* distinguishes itself with a two-stage method that first predicts based on historical patterns, then utilizes external clues for causality-aware calibration.

Causal discovery and inference for time-series data. Causal relationship widely exists in sequence data, e.g., the classical causality between milk price and butter price[144]. Granger causality is proposed to handle the delay of the causal impact[145]. [146] proposes Bayesian forecasting with time-varying causal models, but it only works for short terms. CASTOR [147] introduces time-lagged links into GNN to enhance Granger causality modeling. REASON [148] utilizes graph neural networks to extract layered causal relationships. Instead of discovering and quantifying causality, our solution focuses on applying causality to long-term time-series forecasting with external clues.

Time series modeling for real world applications. Time series modeling is not only a critical question in agricultural practices but also vital in other industries like electricity, weather, traffic, etc [149, 150]. RAPT[151] conducts prediction on medical data for pregnancy complication diagnosis. ClimODE [152] uses physics-informed neural ODE to simulate global atmosphere dynamics for climate forecasting. We will try to enhance the sensor systems to enable neural ODE on Ag-MAR oxygen modeling in future works. For dynamic systems, manifold-aware Gaussian processes infer smoothly changing parameters, while O-MAGIC flags structural change-points online with theoretical guarantees [153, 154]. Large language models (LLM) are also capable of processing spatial temporal data, but so far cannot achieve trustworthiness due to hallucination [155].

MPC based on long-term forecasting. MPC has been used for application scenarios from the microsecond-level horizon (e.g., embedded system voltage control [156]) to the hour-level horizon (e.g., building control [124, 157],

grid control [158]). Existing works utilize MPC for irrigation, but the planning horizon and forecast window are less than one day [7, 159]. Reinforcement learning [160, 161, 162] based control can achieve effective control, but lacks reliability and requires a large amount of data to converge, making it unsuitable for Ag-MAR. To the best of our knowledge, this work is the first MPC work to consider a planning horizon of several days.

Robustness and Efficiency. Robustness is a major concern for applied machine learning [163, 164, 165]. The predicted state trajectory generated as part of the MPC planning process allows the safety check of the trajectory, which offers a higher level of reliability [166, 167]. In future works, we may apply safe guarantee mechanisms like a Gaussian-based uncertainty check[5], to achieve better tradeoffs between reliability and efficiency. Furthermore, given the typically extensive search spaces involved in real-world, long-term forecasting control, efficiency becomes a pivotal aspect. Unlike RL, which can separate agent models from environmental models to improve efficiency [168, 169, 170], time series modeling offers limited scope to balance performance with efficiency. However, recent advances in state space models (SSM) have significantly mitigated computational burdens [171]. Future research could explore the application of SSMs in scenarios where real-time requirements are critical.

3.8 Sensor node deployment

The sensor node is powered by solar energy. As shown in Figure 3.13, it consists of several key components. The Arduino Uno serves as the central controller, managing the operations of sensor readings and LoRa signal modulation. It is connected to KE-25 oxygen sensors and IRROMETER Watermark 200SS soil moisture sensors, which are deployed in the soil. The InAir9B LoRa radio is connected to the Arduino Uno via a relay board, which enables low-power long-range communication. The rechargeable battery along with the solar charger ensures minimum maintenance efforts. Key components are hosted in the waterproof box to protect them from damage and fast aging caused by environmental factors. The

Figure 3.13: The illustration of the solar-powered sensor node.

spreading factor (SF) [172] of the LoRa transmission is 8.

3.9 Conclusion

This paper introduces *MARLP*, a model predictive control system for Ag-MAR, employing heuristic planning and a long-term oxygen forecasting module to optimize groundwater recharge and soil oxygen levels. Through forecasting, *MARLP* effectively manages environmental variables in real-time, enhancing water use efficiency in agriculture. The successful deployment of *MARLP*, which reduces the oxygen-deficit rate and improves the total amount of applied water, showcases the system's potential in precision agriculture and sustainable resource management.

Chapter 4

Task-driven Multi-variate Time Series Modeling with Structural Granger Causality

Granger causality is a fundamental concept in multivariate time series analysis, proven to be valuable in numerous real-world applications. However, most existing research focuses on causal discovery, neglecting the potential of causal insights to improve downstream tasks such as forecasting and imputation.

In this work, we introduce GrangerNet, a novel framework designed to harness Granger causality for multivariate time series modeling. It advances traditional linear approaches by introducing the structural Granger matrix, which captures delayed temporal correlations across multiple orders of derivatives and delay periods for each variable pair. GrangerNet employs an end-to-end training process to learn the optimal mixtures of causal components, tailored to specific tasks. To facilitate model training efficiency, we leverage a causality-augmented regularizer to enforce the sparse nature of the causal structure, as well as the directional and acyclic feature of the causal graphs. We demonstrate that GrangerNet achieves competitive performance with state-of-the-art algorithms, on downstream tasks and causal discovery simultaneously, especially when the causal relationship is explicit.

4.1 Introduction

Granger causality, originally developed in econometrics [173], is a statistical concept that identifies temporal dependencies between time series. It is based on the idea that one time series is said to Granger cause another if its past values can improve the prediction of future values of the latter [174]. This delayed temporal correlation concept has become fundamental across diverse fields such as finance, neuroscience, and biology. Understanding the causal relationships between dynamic variables is essential for tasks like forecasting, imputation, classification, and decision-making.

Despite the clear potential of Granger causality, many contemporary models purely focus on inferring the causal structure [175, 176, 177, 178] but overlook the potential in its applied values. For example, MCD [175] discover causal structures in heterogeneous time series using variational inference, while JRNGC [177] improves Granger causality estimation by integrating a Jacobian regularizer. However, is remains unclear how these causal structures can facilitate practical problem solving for time series, which mainly lies in forecasting, imputation, classification, as well as the subsequent control tasks.

On the other hand, numerous studies have advanced time series modeling and its applications, especially with deep learning techniques [126, 122, 171, 179]. But they mainly concentrate on latent space representation learning, missing the opportunities of intrinsic causality that is easy to interpret and quantify. Timer [180] is a representative recent study that models and extracts causal temporal dependencies. They devise a mechanism called *TimeAttention*, which mask out non-beneficial dependencies during modeling. However, it can not interpret the learnt causality as structural causal models or directed acyclic graphs (DAG). This gap leaves a need for models that not only uncover causal structures but also integrate them into downstream tasks for better performance in real-world scenarios.

In this paper, we propose **GrangerNet**, a novel multivariate time series modeling framework that integrates Granger causality and Delay Differential Equations (DDE) to enhance the performance of time series forecasting and imputation tasks. GrangerNet is designed to learn the Granger causality between one variable and

the others in a multivariate setting, and integrate this information to improve the accuracy of downstream tasks. At the core of GrangerNet is the Granger matrix, an efficient mathematical structure that extracts both the delay and causal effects in the system. Specifically, we model the dynamics of each variable as a function of the past values of all other time series in the system. Considering that each past state may have an influence to the current state, we extend the possible delays as one dimension of the Granger matrix. Besides, the causal relationships may be direct or indirect, motivating the usage of various orders of derivatives as another dimension. In this way, each pair of variables has a sub-matrix that lists the most possible causal relationship between them, with different combinations of delays and derivative orders.

To quantify the importance of each component in the Granger matrix, we devise a weight matrix that matches the elements in the Granger matrix accordingly. Each weight item inside it indicates the strength of a certain relationship, which can provide analytical insights and facilitate applications simultaneously. The two matrixes together form GrangerNet.

As a modeling method, GrangerNet can address various downstream tasks when equipped with specialized output interfaces. The forecasting head utilizes the Granger matrix to predict the fluctuations of future time steps and then makes updates in a step-wise manner. The prediction error can inversely be used as the loss function for learning the weight matrix. The imputation head uses a similar methodology to infer missing values, but with more reference timestamps available, particularly in partially observed scenarios. Benefit from these applications, GrangerNet can be easily trained with flexible task-specific loss functions.

However, a challenge lies in the weight matrix. Ideally, the trained weights can translate to the causal structure, i.e., causal graph, but the learned weights are dense and easy to be filled with noise. To solve this problem and embrace training efficiency, GrangerNet further adopts a causal augmented regularizer. Firstly, it enforces the sparsity of the weight matrix by applying the Frobienus norm in the regularization. Then it leverages the acyclic pattern of the causal graph representation to suppress noise weights.

We evaluate the effectiveness of GrangerNet through extensive experiments on 5 real-world datasets from diverse domains, including agriculture, weather, electricity and medical information. GrangerNet is compared against two categories of state-of-the-art baseline models, for both causal discovery and downstream tasks. Specifically, the baselines include 14 causal discovery models, 4 forecasting models, and 4 imputation models in different categories. The results show that GrangerNet consistently achieves high performance for applications like exogenous clue-aware forecasting and imputation, and can achieve SOTA for scenarios with strong and explicit delayed causal patterns. We also assess the interpretability of the learned Granger matrix, confirming that the discovered causal relationships align with known dependencies in the datasets. Additionally, we explore the impact of parameter choices, including delay window size and orders of derivatives in the modeling, providing insights into the models adaptability.

The contributions of this paper are as follows:

- We introduce the GrangerNet framework, which uses a Granger matrix and a weight matrix to model delayed temporal correlations in multivariate time series. It can serve downstream applications and provide analytical insights simultaneously.

- GrangerNet employs end-to-end training with downstream tasks and adopts a causal augmented regularizer to facilitate the effectiveness and efficiency of the modeling.

- We conduct extensive evaluations on various datasets, demonstrating the competitive performance of GrangerNet in both forecasting and imputation tasks. Post-hoc analysis of the weight matrix reveals that GrangerNet is effective in causal discovery.

4.2 Background and Related Work

4.2.1 Granger Causality Modeling.

Granger causality. Granger causality is a predictive notion of causality, where a time series variable $\boldsymbol{x}_i$ Granger causes another one $\boldsymbol{x}_j$ if the past values of $\boldsymbol{x}_i$ improve the prediction of future values of $\boldsymbol{x}_j$. Given a multivariate time series $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\} \in \mathbb{R}^{T \times N}$ with N variates and T time steps, the system dynamics for each variate $\boldsymbol{x}_j(t)$ can be modeled as:

$$\boldsymbol{x}_j(t) = f_j(\boldsymbol{x}_1(< t), \ldots, \boldsymbol{x}_N(< t)) + \varepsilon_j, \tag{4.1}$$

where ε_j is the independent noise, and each function f_j models the causal relationship among different time series.

Granger causality is usually modeled in a vector autoregressive (VAR) pattern [181, 182, 183]. For instance, NTS-NOTEARS [181] extends the NOTEARS acyclicity framework to nonparametric time-series by adopting 1D convolutional networks for both lagged (inter-slice) and instantaneous (intra-slice) dependencies. Apart from these works, MCD [175] discovers multiple causal models in heterogeneous time series by using variational inference to infer both causal structures and mixture memberships, outperforming state-of-the-art methods in accuracy and clustering. JRNGC [177] improves the estimation of Granger causality by incorporating a Jacobian matrix regularizer into a unified neural network model, improving scalability and accuracy. However, these works do not explore how the parameterized Granger causality can facilitate downstream tasks, and also lack a derivative-based modeling that directly matches the causal nature in real world applications.

Delay differential equations (DDE). DDEs provide a framework to model systems where the rate of change of a state variable depends not only on the current state but also on past states [184]. The general form of a DDE for a single time series $\boldsymbol{x}(t)$ is:

$$\frac{d}{dt}\boldsymbol{x}(t) = f(t, \boldsymbol{x}(t), \boldsymbol{x}_t) \tag{4.2}$$

where $\boldsymbol{x}_t = \{\boldsymbol{x}(\tau) : \tau \leq t\}$ represents the solution trajectory in the past. For a

multivariate system, we extend this to:

$$\frac{d}{dt}\boldsymbol{x}_j(t) = f_j(t, \boldsymbol{x}(t), \boldsymbol{x}_t) \tag{4.3}$$

where $\boldsymbol{x}_t = \{\boldsymbol{x}_1(< t), \ldots, \boldsymbol{x}_N(< t)\}$ captures the past values of all variables and ε_j is the residual of the first derivative.

4.2.2 Causal reasoning in Time Series.

For time series with regular intervals, deep learning based models have successful experience in extracting causal relationships. ORL[185] estimates long-term causal effects of continuous treatments from short-term experiments by leveraging reinforcement learning techniques, offering a robust alternative to surrogate methods. CUTS+[186] improves high-dimensional causal discovery in irregular time series by integrating coarse-to-fine discovery and message-passing graph neural networks. LipCDE[187] improves treatment effect estimation from irregular time series with hidden confounders by using Lipschitz-bounded neural controlled differential equations, reducing bias and enhancing robustness. NCTRL[188] improves latent causal variable identification in nonstationary time series by recovering time-delayed causal relations without observing auxiliary variables or domain indices. On the other hand, Multi-variate Hawkes Process and Neural Point Process focus on event sequence data sampled from irregular time intervals instead of regular-interval time series. For instance, ISAHP[189] discovers instance-level causal structures in an unsupervised manner. These works focus solely on outputting the causal structures and miss the opportunity to utilize them.

4.2.3 Time Series Forecasting with Exogenous Variables.

Incorporating auxiliary exogenous variables has been widely shown to be effective for forecasting, as temporal dynamics are often shaped by external factors [179, 190]. CATS[149] enhances forecasting quality by constructing auxiliary series to enhance representing inter-series relationships. Manifold[153] utilizes the Gaussian process to forecast multivariate time series with nonlinear dynamic structure. WEITS[191] improves interpretable time series forecasting by incorporating

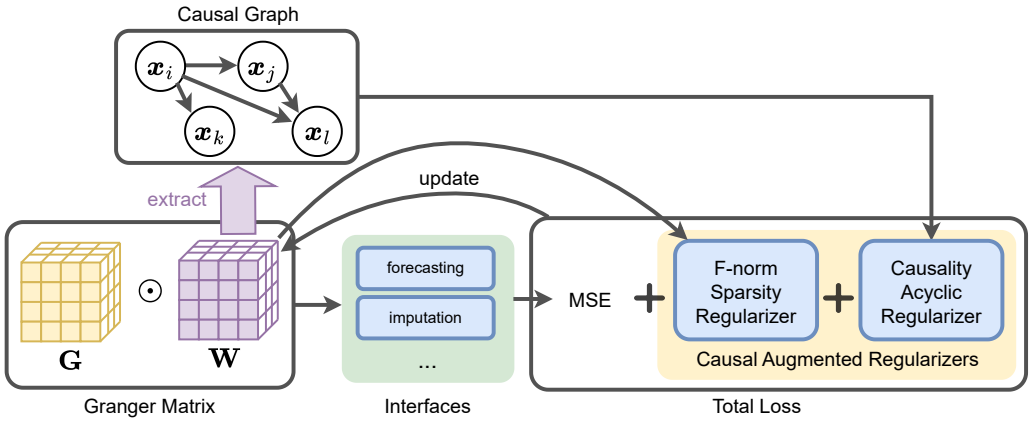


Figure 4.1: The overview of GrangerNet, including two matrixes and the causal augmented regularizer.

wavelet-enhanced residual networks for accuracy and efficiency. [150] proposes an in-context LLM structure and reformulated forecasting tasks as input tokens to LLM structure, alleviating overfitting issues. CauDiTS[192] enhances unsupervised domain adaptation for multivariate time series by disentangling domain-invariant causal rationales from domain-specific correlations, improving cross-domain consistency and robustness. MARLP[193] is the first work that utilizes external time series as clues and has proven its potential in boosting control performance on a real agricultural deployment. However, these works do not explicitly learn the Granger causality. To the best of our knowledge, GrangerNet is the first framework that models the multivariant time series as Granger representations and leverages it for downstream tasks.

4.3 GrangerNet

In this section, we describe the mathematical formation of the Granger matrix and its interfaces for forecasting and imputation.

4.3.1 Modeling Granger Causality with DDE

We formulate the prediction function f_j to predict the change in time series of interest $\Delta \boldsymbol{x}_j$, such that

$$\begin{aligned} \boldsymbol{x}_j(t+1) &= \boldsymbol{x}_j(t) + \Delta \boldsymbol{x}_j(t) \\ &= \boldsymbol{x}_j(t) + f_j(\boldsymbol{x}_1(<t), \ldots, \boldsymbol{x}_N(<t)) + \varepsilon_j \end{aligned} \tag{4.4}$$

The output of f_j can be decomposed into a linear combination of impacts predicted from all time series data

$$f_j(\cdot) = \sum_{i=1}^{N} w_{ij} h_{ij}(\boldsymbol{x}_i(<t)) \tag{4.5}$$

where h_{ij} is a function that predicts the impact on $\Delta \boldsymbol{x}_j(t)$ using a single time series $\boldsymbol{x}_i$ and w_{ij} is a trainable weight. For each time series data $\boldsymbol{x}_i$, we want to measure the causality between the ground truth $\bar{h_{ij}}$ and the combination of two attributes:

1. The rate of change $[n]\boldsymbol{x}_i t(t)$ for various orders of derivative $n \in [0, M]$,
2. The delayed effect $\boldsymbol{x}_i(t-\tau)$ for various delaying time steps $\tau \in [0, T]$.

Because we are dealing with discrete sequences time series data, the delay steps τ are integers. Hence, at each time step t, the combinations of the above two attributes generate MT distinct values. We assign a trainable weight to each of these values and define the output of the prediction function:

$$h_{ij}(\boldsymbol{x}_i(<t); M, T) = \sum_{n=0}^{M} \sum_{\tau=0}^{T} w_{n\tau} \left([n]\boldsymbol{x}_i t \Big|_{t-\tau} \right) \tag{4.6}$$

Where $[n]\boldsymbol{x}_i t|_{t-\tau}$ is computed by recursively calculating $([n-1]\boldsymbol{x}_i t|_{t-\tau+1} - [n-1]\boldsymbol{x}_i t|_{t-\tau-1})/2$, with $\boldsymbol{x}_i t|_{t-\tau} = (\boldsymbol{x}_i|_{t-\tau+1} - \boldsymbol{x}_i|_{t-\tau-1})/2$.

4.3.2 Granger Matrix

The h_{ij} defined above predicts the change in time series $\Delta \boldsymbol{x}_j$ using a single time series $\boldsymbol{x}_i$. We now define the process of predicting $\Delta \boldsymbol{x}_j$ using all time series. First, we define a matrix $H \in \mathbb{R}^{M \times T}$ composed of the values in h_{ij}:

$$\mathbf{H}_{ij}(\cdot) = \begin{bmatrix} [0]\boldsymbol{x}_i t|_{t-0} & \cdots & [0]\boldsymbol{x}_i t|_{t-T} \\ \vdots & \ddots & \vdots \\ [M]\boldsymbol{x}_i t|_{t-0} & \cdots & [M]\boldsymbol{x}_i t|_{t-T} \end{bmatrix} \tag{4.7}$$

We isolate the trainable weights to a separate matrix $\mathbf{W}$ so that the grand sum of the Hadamard of the weight matrix $\mathbf{W}_{\mathbf{H}_{ij}} \odot H_{ij}$ is equal to Eq.4.6. Next, we define Granger matrix $\mathbf{G} \in \mathbb{R}^{N^2 \times M \times T}$ as the correlation matrix between N $\mathbf{H}$ matrices, and weights $\mathbf{W}$, such that

$$\mathbf{W} \odot \mathbf{G}(N, M, T) \tag{4.8}$$

$$= \mathbf{W} \odot \begin{bmatrix} \mathbf{H}_{11} & \cdots & \mathbf{H}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{N1} & \cdots & \mathbf{H}_{NN} \end{bmatrix} \tag{4.9}$$

$$= \begin{bmatrix} \mathbf{W}_{\mathbf{H}_{11}} \odot \mathbf{H}_{11} & \cdots & \mathbf{W}_{\mathbf{H}_{1N}} \odot \mathbf{H}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{\mathbf{H}_{N1}} \odot \mathbf{H}_{N1} & \cdots & \mathbf{W}_{\mathbf{H}_{NN}} \odot \mathbf{H}_{NN} \end{bmatrix} \tag{4.10}$$

$\mathbf{W}$ and $\mathbf{G}$ together form the **GrangerNet**. Given t and $\mathbf{x}_i (< t)$ for $i \in [1, N]$, $\mathbf{G}$ can be viewed as a matrix of constant values. When inferencing with GrangerNet, at each time step t, $\mathbf{G}$ is computed in closed-form, while the final result is calculated by weighting each element in G using the trainable parameters $\mathbf{W}$.

The weight matrix training is driven by downstream tasks, which provide ground-truth labels for loss calculation and learning through backpropagation. For example, in single-step forecasting, we use $\boldsymbol{x}j|t$ as the ground truth and $\boldsymbol{x}j|t-\tau, \tau \in [1, \tau_{max}]$ as the input. The loss is then computed as the MSE of the prediction for $\boldsymbol{x}(t)$.

Ideally, the $\mathbf{W}$ after training translates into a causal DAG, where a higher weight indicates a higher probability of causal relationship presence. This offers interpretable insights into cross-variable correlations, such as the potential rationale between two variables or the presence of mediators. However, the resulting weight matrix is often dense with many conflicting causal relationships. For example, both $max(\mathbf{W}_{ij})$ and $max(\mathbf{W}_{ji})$ can be large, indicating a bidirectional causal

relationship which is not possible. Furthermore, self-causality, represented by $\mathbf{W}_{ii}$, can also be significant. These pseudo-causal pairs, which contradict the acyclic nature of causal structures, result in a confusing causal pattern and cause degraded and unstable performance on downstream tasks.

4.3.3 Causality Augmented Regularizers

To steer the convergence of the weights $\mathbf{W}$ towards not only excellent downstream task performances, but also revealing an interpretable causal structure of the underlying data, we introduce two causality-augmented regularizers to the objective function. These three regularizers are tailored to make the resulting $\mathbf{W}$ to represent a valid causal graph between time series. We now introduce the two regularizers.

Frobienus-norm as a sparsity regularizer. When training the weight matrix $\mathbf{W}$, for elements in $\mathbf{G}$ that is always close to 0, their corresponding weight can explode to an arbitrarily large value. Similarly, the smaller elements in $\mathbf{G}$ can attain large weights. These weights does not significantly affect forecasting or imputation accuracy (since their multipliers are small), but they severely impact our interpretation of the causal graph from $\mathbf{W}$. To penalize redundant and exploding weights, we minimize the norm of $\mathbf{W}$. While the norm choice depends on the nature of the causal structure we aim to impose, to optimize the efficiency, we choose to use the Frobienus norm (F-norm) [194, 177]:

$$\mathcal{F}_{\text{causal}} = \|\mathbf{W}\|_F^2 = Tr(\mathbf{W}\mathbf{W}^T) \tag{4.11}$$

This formulation penalizes large weights and pushes redundant weights to zero, which improves sparsity. This makes sure the model is focused on extracting the most significant causal weights. The resulting higher sparsity in $\mathbf{W}$ enables us to extract a more accurate causal graph.

Causality acyclic regularizer. Cyclic causal relationship happens when the causal graph extracted from $\mathbf{W}$ contains two nodes i, j, such that $i \to j$ and $j \to i$ exists at the same time. This representation indicates that the two variables cause one another. However, this is not possible because an event happened later

in time cannot cause an earlier event. When the two time series elements i and j happens at the same time step and have strong correlation, $i \to j$ or $j \to i$ does not affect the performance of the downstream tasks. Hence, we introduce an acyclic regularization term $\mathcal{L}_{\text{acyclic}}$. This term penalizes deviations from the expected causal relationships, as encoded in the weight matrix $\mathbf{W}$. Specifically,

$$\mathcal{L}_{\text{acyclic}} = \sum_{i,j} \left(\mathbf{W}_{ij} \cdot \frac{1}{1 + \exp(-\mathbf{W}_{ji})} \right) \tag{4.12}$$

$\mathcal{L}_{\text{acyclic}}$ is formulated to encourage sparsity in the Granger matrix, pushing the model to identify a small set of strong causal relationships rather than diffuse, weak connections across all variables.

For cases that $i = j$, $\mathbf{W}_{ii}$ denotes a causal loop, with potential latent mediators and back to self. This case still conflicts with the acyclic consideration. Therefore we apply the same suppression calculation.

Total loss definition. The total loss is the combination of causality suppression and self-causality regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_1 \mathcal{L}_{\text{acyclic}} + \lambda_2 \mathcal{F}_{\text{causal}} \tag{4.13}$$

Here, $\mathcal{L}_{\text{task}}$ is the downstream task-specific loss function. For our forecasting and imputation tasks, it is mean squared error (MSE). λ_1 and λ_2 controls the balance between suppressing minor causality and allowing self-causality.

By integrating the causality regularization into the overall loss function, GrangerNet is encouraged to dismiss the noise and reveal true causal relationships during training, enhancing the training efficiency.

4.3.4 Output Interfaces for Downstream Tasks

To enable GrangerNet to handle specific downstream tasks, we design separate output heads for forecasting and imputation, which leverage the Granger matrix to produce inference outputs.

Table 4.1: Benchmarking causal discovery performance on the CausalTime benchmark [2]. The best and the second-best performance is in bold and underlining, respectively.

Methods	AUROC			AUPRC		
	AQI	Traffic	Medical	AQI	Traffic	Medical
GC [173]	0.4538	0.4191	0.5737	0.6347	0.2789	0.4213
SVAR [195]	0.6225	0.6329	0.7130	0.7903	0.5845	0.6774
N.NTS [181]	0.5729	0.6329	0.5019	0.7100	0.5770	0.4567
PCMCI [196]	0.5272	0.5422	0.6991	0.6734	0.3474	0.5082
Rhino [197]	0.6700	0.6274	0.6520	0.7593	0.3772	0.4897
CUTS [198]	0.6013	0.6238	0.3739	0.5096	0.1525	0.1537
CUTS+ [186]	<u>0.8928</u>	0.6175	<u>0.8202</u>	<u>0.7983</u>	**0.6367**	0.5481
NGC [199]	0.7172	0.6032	0.5744	0.7177	0.3583	0.4637
NGM [200]	0.6728	0.4660	0.5551	0.4786	0.2826	0.4697
LCCM [201]	0.8565	0.5545	0.8013	**0.9260**	0.5907	<u>0.7554</u>
eSRU [202]	0.8229	0.5987	0.7559	0.7223	0.4886	0.7352
SCGL [203]	0.4915	0.5927	0.5019	0.3584	0.4544	0.4833
TCDF [204]	0.4148	0.5029	0.6329	0.6527	0.3637	0.5544
JRNGC [177]	**0.9279**	<u>0.7294</u>	0.7540	0.7828	0.5940	0.7261
GrangerNet (Ours)	0.8601	**0.7319**	**0.8225**	0.7441	<u>0.6155</u>	**0.7608**

Forecasting Head

The forecasting head operates by predicting the change in a time series variable, $\Delta x_j(t)$, using the Granger matrix. For each time step, the model calculates $\Delta x_j(t)$ based on the historical data from a window of size T, leveraging the delayed effects of all input variables. The forecasted change is then added to the current value of the time series to produce the prediction for the next time step.

Given a time series $\boldsymbol{x}(t)$, the forecasting function g_{forecast} is defined as:

$$\Delta \hat{x}_j(t) = g_{\text{forecast}}\left(\mathbf{x}_1(< t), \ldots, \mathbf{x}_N(< t), \mathbf{W}\right) \tag{4.14}$$

The forecasted value at time $t+1$ is obtained as:

$$\hat{x}_j(t+1) = x_j(t) + \Delta \hat{x}_j(t) \tag{4.15}$$

This process is repeated recursively for multi-step forecasting, using the newly predicted values as the basis for subsequent steps. The Granger matrix ensures that each Δx_j incorporates temporal and causal dependencies from the other variables.

Imputation Head

The imputation head operates in a similar fashion to the forecasting head but is designed to infer missing values in the time series. For each missing value at time t, the model calculates $\Delta x_j(t)$ using both the available past observations and the Granger matrix, with a larger time window available compared to long-term forecasting. This increased information allows for more accurate imputation, particularly in scenarios where only a subset of time steps is missing.

The imputation function g_{impute} is defined as:

$$\Delta \hat{x}_j(t) = g_{\text{impute}}\left(\mathbf{X}_{\text{observed}}, \mathbf{G}\right) \tag{4.16}$$

where $\mathbf{X}_{\text{observed}}$ represents the available observed values in the time series, and $\mathbf{G}$ again captures the causal relationships between variables. The missing value is then imputed as:

$$\hat{x}_j(t) = x_j(t-1) + \Delta \hat{x}_j(t) \tag{4.17}$$

Thus, the imputation process leverages both existing data and the Granger matrix to fill in missing gaps in time series, ensuring that the causal structure of the data is respected and used to improve the accuracy of the imputation.

4.3.5 Model Workflow

GrangerNet's training process optimizes the Granger matrix through end-to-end backpropagation, combining a task-specific loss (e.g., forecasting or imputation) with the sparsity regularizer. The task loss drives the model to learn meaningful causal relationships, while the regularizer enforces sparsity, highlighting the most important connections. This approach ensures that GrangerNet learns an efficient and interpretable representation that generalizes to new data.

Post-hoc causal importance analysis. The causal importance can be extracted from the output. Specifically, values greater than an empirical threshold indicate strong weight items that significantly facilitate downstream tasks. This result can be represented as causal DAG, providing interpretable clues as side products.

4.4 Experiments

In this section, we apply GrangerNet on two downstream tasks, forecasting with exogenous clues and imputation. The weight matrix learned as a side output serves to infer the causal structure.

4.4.1 Experimental Setup

For both causal discovery and downstream tasks, we train on the 80% data and test on the remaining 20%. The maximum order of derivatives is set as 3 while the number of delay timestamps is 72. More settings can be found in Section 4.4.4.

Baselines: We choose three categories of models as baselines for forecasting and imputation:

- Multi-Layer perceptron (MLP) architecture. Dlinear [127].

Table 4.2: Comparison of forecasting performance between GrangerNet and baselines, both the input and output sequence length is 720. The five subdatasets are collected in 2020-2024. The best and the second-best performance is in bold and underlining, respectively.

Model	Metric	AgMAR1	AgMAR2	AgMAR3	AgMAR4	AgMAR5
GrangerNet	MSE	**0.392**	**0.324**	**0.790**	<u>0.371</u>	<u>1.261</u>
	MAE	<u>0.552</u>	<u>0.479</u>	<u>0.752</u>	0.510	0.965
TimesNet[122]	MSE	0.576	0.527	0.883	1.130	1.295
	MAE	0.603	0.581	**0.746**	0.924	0.935
PatchTST[126]	MSE	<u>0.465</u>	<u>0.337</u>	1.545	0.342	1.311
	MAE	**0.536**	**0.456**	0.988	**0.431**	<u>0.913</u>
DLinear[127]	MSE	2.370	1.814	4.079	**0.322**	1.598
	MAE	1.391	1.180	1.828	<u>0.446</u>	1.074
iTransformer[120]	MSE	0.537	0.387	<u>0.864</u>	0.477	**1.151**
	MAE	0.581	0.490	0.753	0.527	**0.853**

- Convolutional neural network (CNN) architecture. TimesNet [122].
- Transformer architecture. Two different categories of modeling architecture: patch-wise iTransformer [120] and variate-wise PatchTST [126].

For causal discovery, 14 models are selected as the baselines, as shown in Table 4.1.

Datasets: We use two categories of datasets:

- Causality-labeled datasets for causal discovery. CausalTime [2] benchmarks the causal discovery performance with diverse datasets from representative domains. It has three sub-datasets. The AQI dataset represents air quality metrics collected from multiple monitoring stations, and causal relationships in the data are influenced by the spatial proximity of stations. The Traffic dataset contains real-world traffic flow data with ground truth causal graphs derived from temporal

and spatial dynamics. The Medical dataset is based on electronic health records, where causal relationships are extracted using DeepSHAP, capturing feature importance from patient variables to ensure realistic yet interpretable causal structures. We use all three datasets and directly reproduce key baselines from the original paper.

- Unlabeled datasets for downstream tasks. Agriculture [193], Electricity transformer temperature (ETT) [205], Electricity [206], and Weather [207].

4.4.2 Causal Discovery

To assure the causal DAG, we identify strong correlations from the weight matrix as the causal relationship quantification. This process leverages the interpretability of GrangerNet, as the identified causal relationships correspond to specific orders of derivative and time shifts in the data. In this experiment, AUROC (Area Under the Receiver Operator Curve) and AUPRC (Area Under the Precision-Recall Curve) are adopted as the evaluation metrics.

As shown in Table 4.1, GrangerNet achieves state-of-the-art in the medical and traffic dataset, which achieves AUROC of 0.7319 and 0.8225, respectively. This showcases the capability of accurately grasping the correct causal pairs. For AQI dataset, the performance is relatively low. It is mainly because the air quality at different locations are not strongly temporal-correlated with each other, instead it's more like spatial correlations. Besides, their causal DAG is not fully connected, with lots of connected components and separated variables [2]. This intrinsic shortage of causal correlations leads to a low degree of DAG, which contradicts GrangerNet's strong causal mixture intuition. To summarize, GrangerNet is capable of detecting physically related causal relationships, while relatively weak at other spatial-temporal correlations.

Quantifying causality shifts. Recent studies have observed that the gap between training and testing periods can result in distribution shifts [208], and have proposed methods to quantify and adapt to these shifts over time. In our work, we also analyze the learned weight matrix across different periods [1] and find

[1]Most datasets in our evaluation span at least 2 years.

that the causal structure is relatively consistent. For the ETT and Agriculture datasets, the causal DAG remains stable over 2 and 5 years, respectively. This suggests that Granger correlations are consistent over the long term, as long as the underlying physical processes remain unchanged, affirming the effectiveness and reliability of Granger representation modeling.

Table 4.3: Comparison of imputation performance between GrangerNet and baselines. The best and the second-best performance is in bold and underlining, respectively.

Model	Metric	Weather	Electricity	ETTm1	ETTm2	ETTh1	ETTh2
GrangerNet	MSE	0.030	0.143	**0.022**	<u>0.023</u>	<u>0.076</u>	0.053
	MAE	0.061	0.285	<u>0.117</u>	<u>0.093</u>	0.191	0.166
TimesNet[122]	MSE	**0.029**	**0.089**	<u>0.023</u>	**0.020**	**0.069**	**0.046**
	MAE	**0.052**	**0.206**	**0.101**	**0.085**	**0.178**	**0.141**
FedFormer[209]	MSE	0.064	0.120	0.052	0.080	0.106	0.137
	MAE	0.163	0.251	0.166	0.195	0.236	0.258
DLinear[127]	MSE	0.048	0.118	0.080	0.085	0.180	0.127
	MAE	0.103	0.247	0.193	0.196	0.292	0.247
NS-Trans.[142]	MSE	<u>0.029</u>	<u>0.097</u>	0.032	0.024	0.080	<u>0.049</u>
	MAE	<u>0.056</u>	<u>0.214</u>	0.119	0.096	<u>0.189</u>	<u>0.147</u>

4.4.3 Downstream Tasks

Forecasting with Exogenous Clues

Exogenous variables that have causal relationship with target variables can facilitate the forecasting performance [193, 190]. This enlightens several applications including agriculture and energy, exogenous variables like weather forecasts and temperature have a significant impact on the future soil oxygen and energy.

In this experiment, we use the Algriculture datasets. To forecast one target variable for T'-length output, we use the historical T-length inputs of all vari-

ables and the auxiliary values (the weather forecasting and action series) of other variables for future T' timestamps as the exogenous clues. Specifically, we use a patch length of 16 consistently and fix the lookback series length T to 96. The prediction length T' varies among the following values: 96, 192, 336, and 720. The auxiliary variables of the same length as the prediction length are used separately as exogenous clues. They are input into GrangerNet during the training stage, but are not used as the outputs.

Results. As shown in Table 4.2, GrangerNet achieves consistently high accuracy and outperforms all the other baselines in MSE on three datasets. Notably, while GrangerNet tends to have a larger MAE, its superior MSE suggests that it captures overall trends effectively but may face challenges with consistent small errors.

Imputation

Real-world systems operate continuously and are monitored by automated observation equipment. However, malfunctions can result in partial gaps in the collected time series, making downstream analysis challenging. Imputation is, therefore, widely used in practical applications, e.g., computer networks [210]. We choose datasets from electricity and weather domains as benchmarks, including ETT [205], Electricity [206], and Weather [207], where missing data is a common issue. To assess the models performance under varying degrees of missing data, we mask 25% of time points. Additionally, since iTransformer and PatchTST are designated for forecasting tasks, we include two other models, FedFormer [209] and Non-stationary Transformer [142] as baselines.

The imputation task, driven by missing time points, requires the model to identify underlying temporal patterns from irregular and partially observed time series. It has an intrinsic feature that the gaps between missing values are small. Consequently, compared with forecasting tasks, the cumulative error and distribution shift along the temporal domain are negligible. Therefore, the MSE of imputation is notably lower than forecasting.

Results. Table 4.3 shows the MSE and MAE of imputation. GrangerNet

achieves consistently high performance and exceeds TimesNet on an ETT dataset, while on Electricity dataset, the accuracy degrades. This is because each variable in the dataset is from one independent client, which reveals weak causal correlations between each other. For weather and ETT datasets, the causality is intuitive and significant, e.g., the oil temperature and the amounts of different types of workloads. In such datasets, GrangerNet achieves SOTA or close-to-SOTA performance.

4.4.4 Parameter Study

Table 4.4: The forecasting performance of Agriculture dataset on different values of M and T.

M	1	2	3	4	5	6
MSE	1.015	0.772	0.628	0.619	0.622	0.637
T	36	48	60	72	84	96
MSE	0.897	0.720	0.712	0.628	0.621	0.617

The model complexity is $O(N^2MT)$, which depends on how many steps are set for the order of derivatives and delays. To test how these settings influence the modeling capacity and performance, we analyze various different settings of M and T to see the trade-off between the forecasting performance and computing time. Only one parameter is changed for each test.

Table 4.4 shows the forecast performance in the agriculture dataset under different settings of derivative orders and maximum window of delay modeling. It demonstrates that the first three orders of derivatives offer significant contributions to the performance, which supports the derivative calculation. The 4th and 5th orders offer negligible impact while the 6th order brings in extra noise that degrades the forecast accuracy. On the other hand, increasing the delay modeling window brings consistent gain, but it becomes marginal for delay more than 72 timestamps. Considering the sampling rate of the dataset is 10 minutes, it implies that most

Granger causality lies in 12 hours for environment-soil dynamics. The optimal value of these two parameters depends on the dataset specifications, e.g., the meaning of each variable, the presence of latent mediators, and sampling rates.

4.5 Conclusion

In this paper, we introduced GrangerNet, a framework that leverages Granger causality to enhance multivariate time series modeling for tasks like forecasting and imputation. By incorporating the structural Granger matrix and a causality-augmented regularizer, GrangerNet effectively captures temporal dependencies and improves the interpretability of the model. Our experimental results demonstrated that GrangerNet performs competitively with state-of-the-art methods, particularly in datasets with clear causal relationships. The learned Granger matrix provides valuable insights into the underlying data structure, reinforcing the benefits of causality-aware modeling. Future work can explore extending GrangerNet to continuous representations of derivatives and delays.

Chapter 5

Enabling Generative Visual Captions in Live Video Streaming

Visual captions enhance the understanding of complex or unfamiliar concepts in live video streaming, such as TV broadcasts or video conferences. The process typically involves two steps: proposing an appropriate image based on user speech and seamlessly integrating it into the video. For live streaming, visual captioning must operate in real-time, completing both steps for one concept before the next arises. However, these steps require intense computational resources. This paper proposes GenRTC, a real-time visual captioning system comprising three key components. First, we leverage LLMs to predict user-referenced concepts and pre-generate corresponding visual captions. Second, a codec-aware scheduler integrates captions into the video stream adhering to the WebRTC protocol. Third, to support multiple simultaneous video streams, GenRTC incorporates a MPC-based scalable computational resource management scheme, optimizing the execution of visual captioning tasks across all streams. Data-driven simulations and real-world experiments demonstrate GenRTC delivers high prediction accuracy and maintains excellent video quality, ensuring a seamless live-streaming experience.

Figure 5.1: One use cases of GenRTC for live streaming. Visual captions on robots and remote-controlled driving are generated when the speaker mentioned them. With GenRTC, everyone can stream with visual demos like CEOs.

5.1 Introduction

Live video streaming plays a vital role in sectors such as web conferencing [211, 212], live TV streaming [213], cloud gaming [214, 215], and interactive virtual reality (VR) [216]. Visual captioning dynamically generates images to illustrate some complex or unusual concepts in conversations, improving viewers' understanding. While promising, preparing visual materials ahead is labor-intensive, while automating visual captions(e.g., retrieves images from the Internet with speech keywords [217]) struggles with intent-image alignment, especially for complex or uncommon scenes where it fails to deliver relevant images. This paper presents GenRTC, a visual captioning system that leverages advanced text-to-image generative models [218, 219, 220, 221, 222] to generate images from users' words, enabling the visualization of unusual or complex scenes. Figure 5.1 illustrates a live streaming example where the speaker introduces a computing platform. Generated visual captions make it easier for viewers unfamiliar with these domains to comprehend the specific scenarios being discussed.

To integrate text-to-image generative models into live streaming systems, we need to tackle three key challenges. **First,** the generation process for visual captioning is time-consuming, making real-time visualization difficult; and the latency of rendering the images and the original video frames is long too, which makes the visual captions appear with a large delay after corresponding concepts men-

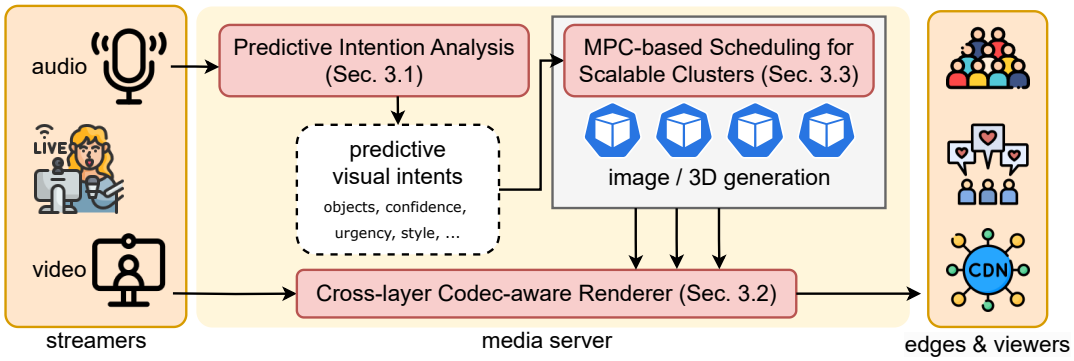


Figure 5.2: Workflow of GenRTC, including three components: intent analysis, intent rendering, and scheduling for scalable streamers. Intent refers to the visual content (e.g., images) that a speaker wishes to display.

tioned, significantly disrupting the user experience. **Second,** integrating generated images into the default video streaming workflow introduces complications. The generation pipeline may interfere with video encoding, increasing frame delays and reducing video quality. In systems like WebRTC that use frame-dropping strategies to maintain synchronization, the computational load from visual generation can limit the framerate, affecting video fluidity and overall viewing experience. **Third,** the resource-intensive nature of real-time generative models often creates bottlenecks, particularly when server capacity is stretched by concurrent live streaming sessions. Managing computational resources (GPU Flops) on a server to optimize the global Quality of Experience (QoE) for all users is critical. This requires advanced scheduling algorithms and resource allocation strategies to ensure QoE across all sessions without compromising visual captioning performance.

GenRTC is designed with three novel technical components to overcome the challenges, as shown in Figure 5.2. First, we introduce a **predictive intention analysis** module to perform audio-to-text conversion and fine-tune Large Language Models (LLMs) to discern current focal points or predict future intents in the dialogue. An intent is the visual content a speaker wishes to display on the streaming video, capturing her implicit desire to explain her spoken words. This proactive approach generates visual captions before a certain focal point is mentioned, mitigating the impact on user experience from the delays caused by

time-consuming LLM inference. Next, we develop a **cross-layer codec-aware renderer** that integrates the generated images into the original video frames with minimum processing latency and video quality degradation. This module synchronizes the generative and rendering-encoding workflows, optimizing parallel processing to minimize the impact on user experience, effectively balancing the load between visual generation and video streaming tasks, and mitigating the interference of generation workload to the streaming workflow. Lastly, we design a **MPC-based scalable scheduling** strategy to ensure server-side scalability across multiple live-streaming sessions. Specifically, we formulate the scheduling as an online bin-packing problem and solve it with Model Predictive Control (MPC) that enables efficient and responsive resource management by considering a set of factors, such as the urgency of visual intents, quality requirements, and network conditions.

We implemented GenRTC with a widely used open-source real-time communication framework (WebRTC [214]). We tested different computation partitioning strategies and found the most efficient one, i.e., all video processing operations are handled on the media server. To validate GenRTC's effectiveness, we conducted a comprehensive evaluation via trace-driven simulations and real-world experiments. We also assessed the impact of generative visual captions on system performance and user experience. The results show that GenRTC achieves an average perceived delay of 5.08 s, with a deduction of 2.13 s over the strawman implementation. 71.2% among all intents are rendered in time, which is 4.62× over strawman. The overhead on frame latency and PSNR drop is 14.6ms and 0.19 dB, which are negligible. Case studies on live streaming and collaborative VR further validate the user experience provided by GenRTC.

To summarize, this paper has the following contributions:

- Proposing GenRTC, leveraging generative AI to perform visual captioning for live video streaming.
- Designing three system designs that reduce the perceived latency introduced by generation workloads, transforming GenRTC into a practical real-time visual captioning system.

- Evaluation indicates that the end-to-end latency, scalability, fairness, and user study are significantly improved and fulfill the user requirements.

5.2 Adopting Generative AI for Live Visual Captions: Benefits and Challenges

5.2.1 Benefits

Advanced text-to-image generative models [219, 220, 221, 222], such as Stable Diffusion [223], generate images from text descriptions, enabling visualization of complex or rare scenes that are often unavailable on the Web. To validate this proposal, we compared two approaches: retrieving images from the Internet and generating images using Stable Diffusion [223]. We use two categories of text prompts: text from TV clips where people are unaware of generative functionality and prompts collected from user studies where volunteers know about that their intents will be visualized. User scores are used to evaluate how well each image matches its corresponding text description, from 5 perspectives: semantic match, detail completeness, visual quality, content focus, and creativity, with each on a 5-point scale `(1=very bad, 2=bad, 3=fair, 4=good, 5=very good)`.

As illustrated in Figure 5.3, a user study with 20 participants revealed significant improvements when using generative captions compared to retrieval-based results: Increasing semantic match improved by 5.1%, detail completeness by 17.9%, visual quality by 1.5%, content focus by 58.5%, and creativity by 18.6%. These results highlight the ability of generative models to provide clearer, more focused, and imaginative visual representations. By addressing the limitations of retrieval-based approaches, generative models enable transformative applications in education, creative design, and personalized content generation.

5.2.2 Challenges

While integrating generated visual captions into live video streaming offers significant benefits, it also introduces unique challenges. Existing works on live

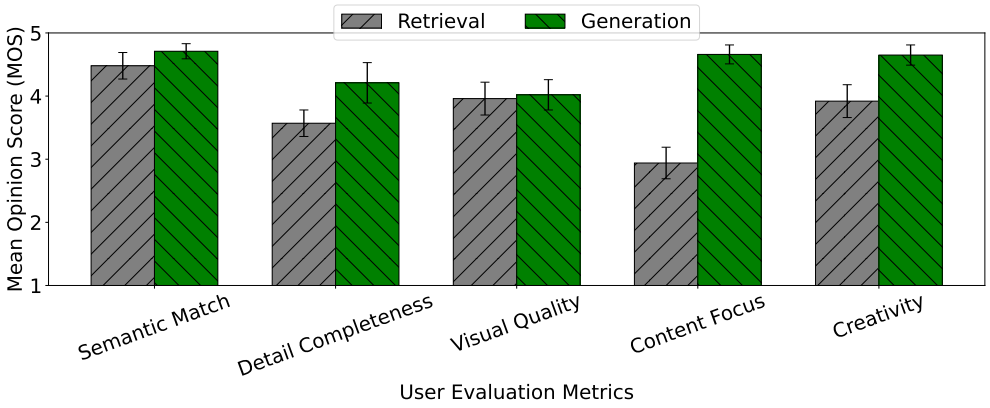


Figure 5.3: Generated visual captions improve user intent satisfaction significantly over retrieval.

streaming has explored different aspects of system optimization, e.g., neural compression [219, 224, 225], adaptive frame rate [226] and redundancy reduction [227], which mainly focuses on reduce the overall latency. However, these approaches can not enable live visual captions due to the following challenges we recognized during the implementation:

Long Generation Time vs. Strict Requirments

When using generative models for visual captions, the process begins by extracting text descriptions as prompts from the speech. These prompts guide the models in generating corresponding images. However, the dynamic nature of topic migration presents alignment challenges.

Resource-Intensive Generative Models. Although text-to-image models such as DALL-E 2 [228], DALL-E 3 [222] and Stable Diffusion [218] produce high-quality images, inference is significantly time-consuming. Among these advanced models in our test on an A100 GPU, the fastest average inference record is 3.81s by SD. This record does not include network delays or the time required for intent recognition.

Frequent Intent Migration in Verbal Communication. Human speech typically occurs at a pace of 100-150 words per minute in English [229], which allows speakers to shift rapidly across topics [230]. In those cases, each visual intent is set a strict deadline for the generated visuals to be rendered, which is the

end of this topic or the start of the new one. Otherwise, they will be outdated and confuse the audience with misaligned captions. In this case, they should only be discarded and computing resources are wasted. In our dataset, the average intent duration is 5.17 seconds, with 21.9% of intents lasting less than 3.81 seconds.

This misalignment makes it impractical to incorporate generative functionalities for commercial applications.

Video Quality Degradation

During implementation, we observed that integrating visual captions into live video streams could build up the overall latency and degrade the overall quality of the original video.

For local rendering, the delay does not impact the original video, as it can be deferred, similar to auto-captions on commercial platforms like YouTube [231]. However, server-side real-time rendering is different because the entire video must be synchronized before transmission. The reasons for latency buildup are two-fold. First, the video must be decoded for rendering and then re-encoded, introducing additional latency. Second, the generation and rendering processes increase CPU and memory usage, further delaying video processing [232]. In our measurement, the median RTT increased to 265 ms. However, for interactive video streaming systems, end-to-end latency is expected to remain below 130 ms [233, 234, 227].

Moreover, the additional jitter caused by these delays leads to uneven video delivery, which can cause the transport layer protocol to overestimate queueing delays and, as a result, reduce the bitrate selection. This, combined with fluctuating network conditions between participants, exacerbates performance issues and impacts the consistency of video quality. Quantitative measures in PSNR also show significant drops, as illustrated in Figure 5.9(c), highlighting the need to mitigate the impact of additional computation workloads.

The generation pipeline introduces frame drop and quality degradation. Specifically, the rendering process causes frame drops in WebRTC's frame logic. In VoD scenarios with a large buffer, rendering does not affect quality. However, in real-time streaming, without an explicit buffer, Google Congestion Control (GCC) [235]

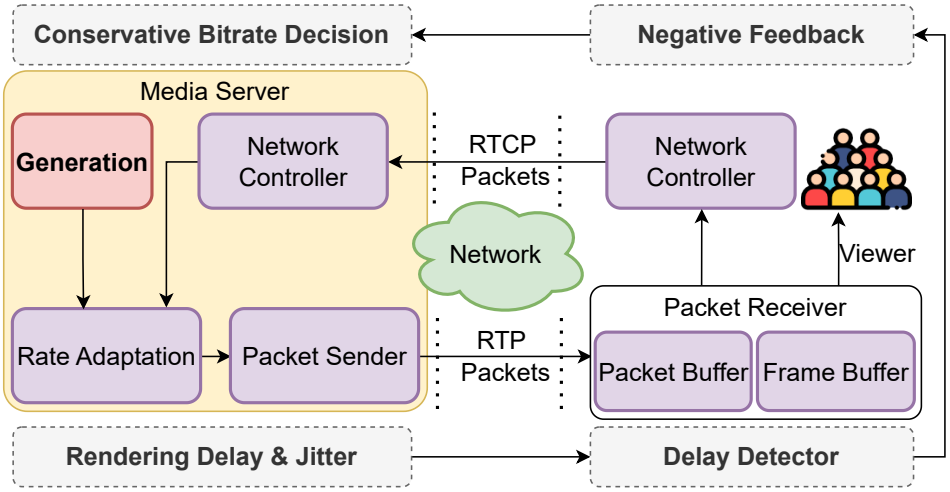


Figure 5.4: Illustration of how the generation process causes frame delay during rate adaptations.

perceives any frame delay as network congestion, leading to a reduction in bitrate. As illustrated in Figure 5.4, the jitter caused by the rendering delay will cause the receiver side to send negative feedback through RTCP packets. The sender side mistakenly takes it as network flunctuations, resulting in conservative bitrate adjustments.

Scaling up to Multi-session

Commercial streaming providers commonly use CDNs to manage downstream traffic [236, 237, 238, 239], while central media servers handle millions of simultaneous streams using advanced signaling techniques [240]. This infrastructure is designed to deliver high-quality streaming experiences to large user bases.

Scaling the generation of visual captions across multiple sessions, however, presents challenges. Each session varies in network conditions, object distributions, and quality or urgency requirements, complicating scheduling and resource allocation. To address these challenges, the system must dynamically adapt to real-time demands, ensuring efficient resource usage while maintaining overall system performance.

The strawman's task scheduling is managed using a first-fit (FF) algorithm [241], where the available GPU pods are assigned tasks immediately. Additional tasks

are placed in a queue until resources become available. Missed intents, whether from pods or the queue, are discarded to prevent outdated requests from consuming resources, ensuring efficient processing and timely visual captions.

Throughout our evaluation, the strawman solution achieves an intent rendering rate of only 15.1%, highlighting a significant under-utilization of computational resources. For the 84.9% of missed intents, many of them have halfway inference but have become outdated. This inefficiency is primarily due to rapid shifts in user intent and varying intent requirements across sessions. These factors contribute to delays in visual caption generation, often resulting in missed objects and suboptimal user experiences.

5.3 Design of GenRTC

In this section, we detail the design of GenRTC describing the three key modules shown in Figure 5.2. The intention analysis and object planning module continuously parses speech and outputs the objects the speaker intends to visualize. The cross-layer codec-aware renderer ensures optimal video quality and latency during rendering. Finally, the scalable clusters and scheduler maximize GPU resource utilization when multiple sessions are hosted on a single media server.

5.3.1 Predictive Intention Analysis

GenRTC employs audio-to-text models and fine-tuned LLMs to perform real-time speech analysis during video streams.

Figure 5.5 illustrates this process, where each gray circle represents audio parsing triggered at regular intervals. The speech history and recent speech are organized into a few-shot prompt, allowing fine-tuned LLMs to identify and describe the three intents most likely needed by the speaker. Specifically, the prompt template guides the LLM to output in a JSON format, including the description and confidence level. In some instances, if no new intent is detected in the recent speech, the output visual caption prompts remain unchanged.

Additionally, to ensure visual consistency, we maintain a subject memory that

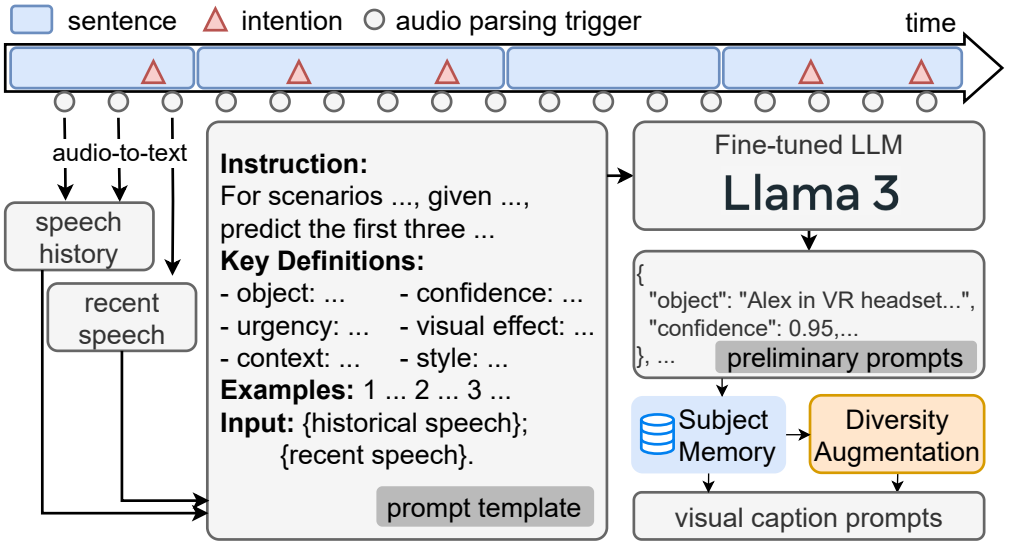


Figure 5.5: Framework for analyzing user intents.

stores visual feature descriptions of intents. After generating the initial prompts, the subject memory complements them and updates itself based on new information. This approach guarantees visual consistency [242, 243]; for example, the visual representation of 'Alex' remains consistent across multiple samples. For human subjects without prior memory, we apply diversity augmentation [244] to ensure diversity in gender, age, and other traits.

LLM Fine-Tuning. We fine-tune LLMs with labeled datasets to improve accuracy, using frameworks from Together.ai for seamless integration of custom data into pre-trained models.

The dataset consists of 263 manually labeled `(<speech text>, <intent>)` pairs, mapping speech segments to corresponding intents. LLaMA-3 70B [245] serves as base models, fine-tuned through multiple iterations to better align with the dataset. Fine-tuned LLaMA-3 improves intent analysis accuracy by 21.4% and Top-3 accuracy by 17.1%. These results highlight an enhanced understanding of speech-intent relationships, leading to more accurate, contextually relevant predictions in live video streaming.

5.3.2 Cross-Layer Codec-Aware Renderer

Traditional media servers simply forward media content without modification. However, to enhance user experience, we enrich media streams by rendering visual captions into video frames. To avoid heavy computational load of decoding and encoding in the strawman solution described in Section 5.2.2, we design a codec-aware, cross-layer renderer that efficiently integrates visual captions into WebRTC streams. Additionally, integrating visual captions into video frames requires modifying the GCC transport layer algorithm to ensure rate adaptation accounts for changes in frame size.

Codec-Aware Rendering Algorithm

Efficient live video streaming relies on video compression techniques, where the video is encoded on the streamer side and decoded on the viewer side. On the media server, while it is possible to decode, modify each frame, and re-encode it, this process introduces significant additional time consumption due to the overhead of these steps. Therefore, we introduce a codec-aware rendering algorithm to bypass this overhead by directly processing encoded frames.

Video codecs leverage inter-frame relationships, such as those between I-frames and predicted frames (P-frames). I-frames are full-size, self-contained frames that can be decoded independently and serve as key reference points. P-frames, in contrast, store only the differences from preceding frames, significantly reducing file sizes while maintaining continuity. P-frames are dynamically selected by the encoder based on video dynamics and application latency requirements. In live streaming encoding, the percentage of P-frames is high to minimize overall video file size [225], making P-frames optimization essential.

Figure 5.6 illustrates our algorithm for modifying encoded I-frames and P-frames to maintain high-quality visual communication under varying network conditions.

For I-frames, they directly render the visual captions on the default mask area. P-frames, on the other hand, are carefully processed, which first pass the mask detection, adaptive rendering and redundancy removal to optimize the efficiency.

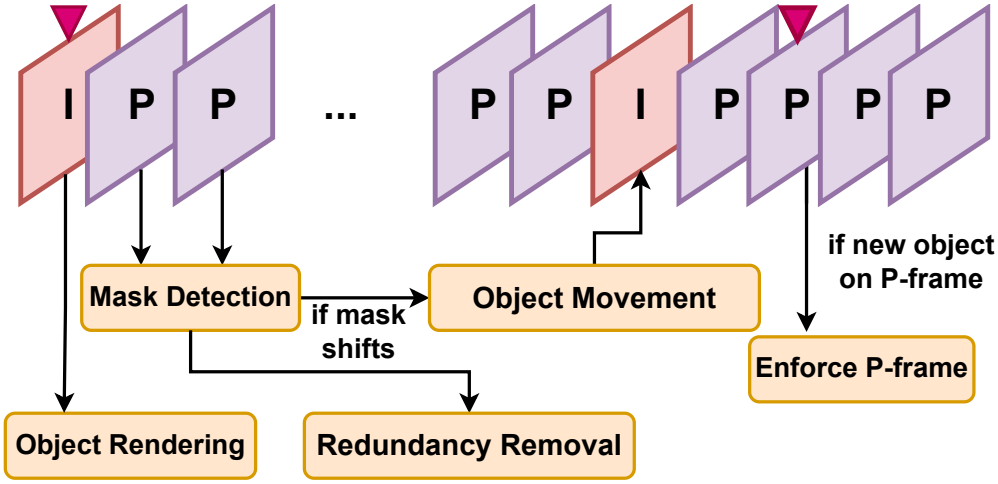


Figure 5.6: Codec-aware rendering, with the red triangle representing the rendering operation.

Mask Area Detection. Mask detection identifies and updates the relatively static regions of the video frame, referred to as the mask area. This ensures that captions do not obstruct dynamic visual content, which is highly possible to be the viewer's interests [246]. While computer vision algorithms could analyze each frame for static regions, they require significant computational resources [247]. Instead, we adopt a more efficient method by leveraging the intrinsic information in P-frames. Since P-frames encode only the differences from preceding I-frames or P-frames, they naturally highlight areas of motion or change, eliminating the need for full-frame analysis. By analyzing the motion vectors encoded in P-frames, which represent the degree of change in different regions, static areas with low or zero motion can be identified. Every timestamp, we sample the next available P-frame and heuristically search for low-dynamic regions, designating them as the mask area for rendering visual captions without disrupting the scene's flow. This approach reduces computational overhead and ensures captions are placed in unobtrusive areas, preserving both video quality and user experience.

If the mask area maintains, the visual caption should remain static. But if the mask area shifts due to dynamic content in subsequent frames, it will be calculated using the pixel density statistics of all P-frames between two I-frames and enforce the area change at the next I-frame.

Adaptive rendering. When a new visual caption comes on the timestamp

of a P-frame, the module modifies this P-frame directly, ensuring visual captions remain aligned and relevant. The other P-frames will remain unchanged.

Redundancy Removal. For all P-frames, redundancy removal is performed at the end to avoid duplicating visual information. The intuition is that the original pixels covered by the visual captions no longer need to be transmitted. In the workflow, we directly filter out the redundant pixels without transmitting them.

This combination of mask detection, object rendering, and redundancy removal allows for smooth visual integration across both I-frames and P-frames, adapting dynamically to changes in the video stream.

Incorporating with Rate Adaptation

Rate adaptation adjusts the video bitrate based on available bandwidth, regulating I-frames and P-frames to maintain smooth streaming, even with limited bandwidth, by compressing frames and minimizing interruptions. This design is compatible with both H.264 and VP8 codecs.

Bitrate Adaptation. Algorithm 1 outlines the bitrate adaptation process. When modifying encoded frames, GenRTC employs a model predictive control (MPC) approach to manage frame rate preferences and P-frame queue sizes. This predictive model tests and adjusts frame-dropping ratios for the upcoming seconds of the stream, optimizing for: *1) Ensuring the tunnel is not idle.* By dynamically adjusting the frame rate and size, the system ensures efficient use of the network tunnel, avoiding under-utilization that could cause perceived lag. *2) Maximizing expected QoE.* The goal is to maximize user experience by balancing frame rate, resolution, and quality with available network bandwidth and the client device's computational power. Resolution and frame rate are increased as much as possible, while reserving bandwidth to handle network fluctuations and variations in video frame size, ensuring a smoother experience for users.

Frame Rate Adaptation. To handle outdated frames, WebRTC provides the `DropTail` [248] functionality, which drops frames in the receiver queue and requests a new keyframe if overflow occurs. We use the same `DropTail` setting as a backup when jitter causes overflow.

Algorithm 1: Rate Adaptation MPC

Input: Past Latency, Network conditions, QoE metrics

Output: Bitrate calibration for the current state

```
Function AdaptationMPC(Latency, Network, QoE):
    FrameRatePrefs ← initializePreferences()
    PFrameQueue ← initializePFrameQueue()
    while streaming do
        DropRatio ← predictRatio(Latency, Network)
        adjustFrameDrop(DropRatio)
        if tunnel is idle then
            ensureNonIdleTunnel()
        MaxQoE ← maximizeQoE(QoE, Network)
    return calibrated bitrate
```

5.3.3 MPC-Based Scalable Scheduling

For large-scale streaming services, ensuring scalability is crucial, as the media server must support multiple streaming sessions. In our system, GPU resources are utilized for generative model inference to create visuals, such as images. The GPU scheduling problem is formulated as an online knapsack problem [249, 240]. Specifically, intent-rendering tasks are queued and face two options: execute immediately or wait. The optimization objective is to maximize task completion (task satisfaction) within the available GPU capacity, while considering the urgency and confidence of each task.

1) Scheduling Input - Sequential Scheduling Task. We denote $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ as the set of intent-rendering tasks that arrive over time. Each task T_i has the following attributes:

- Priority $v_i \in \mathbb{R}^+$: representing the priority of the task.
- Confidence $c_i \in (0, 1]$: representing the resources required (e.g., GPU time) to complete the task.

- Intent Duration $d_i \in [0, 30000)$ (ms): representing the time period during which rendering is considered effective.
- Urgency $u_i \in [0, 30000)$ (ms): the estimated time until the task is needed, with 0 indicating immediate need.
- Roundtrip Latency $rtt_i \in \mathbb{R}^+$ (ms): measured by the WebRTC API and stored on the media server.

2) Scheduling Output - Decision Variable. A binary variable where $x_i = 1$ indicates that task T_i is selected for execution. Conversely, $x_i = 0$ indicates that task T_i is deferred, meaning it will not be executed at this time and may be revisited later depending on resources and scheduling priorities.

3) Optimization Objective. Maximize the total value of the executed tasks, subject to the C1 and C2 constraints:

$$\max \sum_{i=1}^{n} x_i \cdot v_i, \tag{5.1}$$

- C1: Higher-priority and higher-confidence tasks are favored, so the selection should consider urgency scores u_i, potentially integrated into the value term:

$$v_i = c_i \times \max\left(0, 1 - \frac{rtt_i + T_G}{d_i + u_i}\right) \tag{5.2}$$

  where T_G is the mean generation time, and $1 - \frac{rtt_i + T_G}{d_i}$ represents the likelihood of meeting the deadline if the task is executed immediately.

- C2: Capacity Constraint: The total resources used by the selected tasks should not exceed the GPU's capacity C_{gpu}:

$$\sum_{i=1}^{n} x_i \cdot w_i \leq C_{gpu} \tag{5.3}$$

4) MPC-Based Solution. We propose an MPC-based scheduling algorithm to dynamically manage GPU resources. By predicting future task arrivals and estimating intent duration using the harmonic mean of the last 10 intents from the same streamer, the model adapts to the streamer's habits. The MPC algorithm

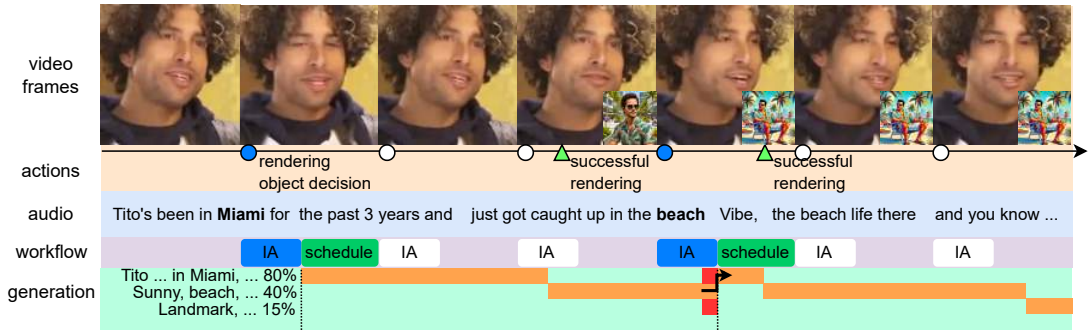


Figure 5.7: An example of control workflow of GenRTC. Each line has different representations. "IA" stands for intention analysis. The blue circles indicate IA events that obtain updated results, while the white circles remain the same output as the previous one. The green triangle means successful rendering.

evaluates tasks in real-time, considering urgency, priority, confidence, and GPU capacity to determine whether tasks should be executed immediately or deferred. It continuously re-evaluates decisions as new tasks enter the queue, ensuring efficient resource allocation and prioritization of high-priority tasks, while minimizing latency and adhering to GPU capacity constraints.

5.3.4 GenRTC: Put It Together

Figure 5.7 illustrates the GenRTC workflow for the generation and rendering during a video segment. The process begins with periodic intention analysis (IA), where the system listens to the speaker and predicts multiple possible intents based on the speech content. In this example, the speaker first mentions "Miami," triggering three intent predictions with varying confidence levels. GenRTC generates the first intent, which is rendered as an image and added to the video stream, indicated by the green triangle. The process continues with the second intent inferred and scheduled until the next IA event when predictions are updated. This design enables GenRTC to achieve fast and efficient rendering by processing multiple intents in parallel, minimizing latency, and ensuring that visual content remains synchronized with the speaker's dialogue.

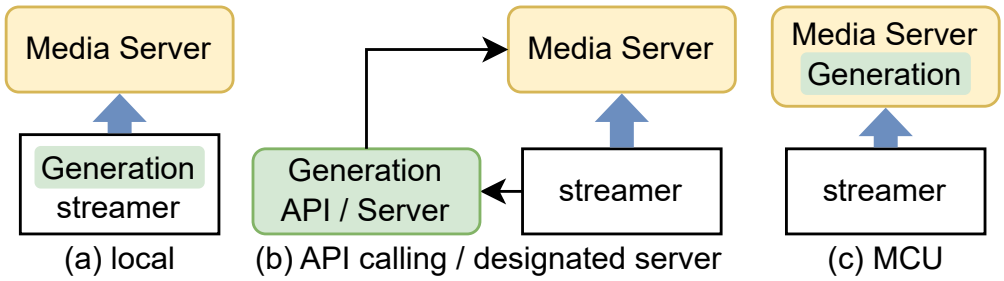


Figure 5.8: The major architectures of generative visual captioning in live video streaming. Blue arrows indicate heavy video streams while black arrows mean intermittent requests.

5.4 System Implementation

GenRTC is built on WebRTC, enabling real-time communication capabilities essential for live video streaming. We choose aiortc [250] for easier interfaces with Pytorch.

5.4.1 Client-Server Architecture

Figure 5.8 displays three major architectures, each differing in where the generative model is executed.

- (a) Local generation: Content generation is co-located with the streamer, enabling immediate processing and integration with streamed media. While this aligns well with the widely adopted SFU architecture, limited GPU capacity on streamers leads to high latency and overwhelmed CPUs [251], affecting video encoding and transmission. Experiments on an RTX 4090-equipped laptop show significant latency buildup and frame rate degradation, confirming its unsuitability for edge deployment.
- (b) API calling / Designated server: This scheme utilizes a separate server for generation tasks, interacting with the media server via an API, which helps distribute processing loads. However, the reliance on network connectivity between generation server and media server introduces latency issues, complicating real-time streaming performance.
- (c) Multipoint Control Unit (MCU): The MCU architecture involves putting the

generation workload with the media server, similar to the traditional transcoding tasks. It is often selected for its capability to handle complex, stream processing tasks without the drawbacks mentioned above.

We measure the end-to-end latency of each scheme with the same network traces (Round-Trip Time (RTT) = 102 ms). The local machine is a laptop with an RTX 4090 GPU. For scheme (b), we use the API of calling stable diffusion instead of using closed-source models. The settings remain the same for the generation server and media server, but it also serves another streaming to maintain fairness. The average time gaps are 19,207 ms, 11,383 ms, and 9,604 ms, respectively. Therefore, the MCU architecture is the optimal solution for the best latency performance.

Note that we excluded placing generation workloads on the edges or viewers due to redundancy, that is, each device would need to generate its own visual captions, multiplying inference by the number of edges / clients.

Media Server. The server infrastructure of GenRTC is powered by high performance hardware, with NVIDIA A100 GPUs, designed to handle the intensive computational demands of real-time video processing and the generation of complex visual representations. The network interface card (NIC) specifications are tailored to support high-speed data transfer rates, essential for maintaining the quality and responsiveness of the video conferencing experience.

Streaming Client. Clients interact with the GenRTC system comprising a diverse range of devices, including laptops and Android phones, each accessing the service through web pages. This cross-platform compatibility ensures wide accessibility, catering to users in various scenarios. According to our measurements, the choice of end devices does not impact the QoE measurements. However, we include different types of devices in the case study.

GPU Scheduling over Container. The rendering process is encapsulated within Kubernetes containers, offering a modular organization that facilitates flexible GPU resource allocations and updates, which closely aligns with industrial practices. Within each worker node, Redis is used to manage the information of the media file, speech history and subject memory, and system status, ensuring efficient data retrieval and real-time updates.

Network Condition. To realistically simulate and optimize performance under typical usage scenarios, we evaluate both real network environments and trace-driven simulations. The real-world tests include both cellular and Wi-Fi connections at the edge. This approach addresses challenges posed by varying network speeds and stability, ensuring that GenRTC consistently delivers high-quality visualizations across different environments and network conditions.

5.5 Evaluation

In this section, we address the following questions through two open video datasets and two case studies:

RQ1 How effective is GenRTC in analysing and predicting intents from the speech?

RQ2 What's the QoE compared with vanilla WebRTC and the strawman implementation?

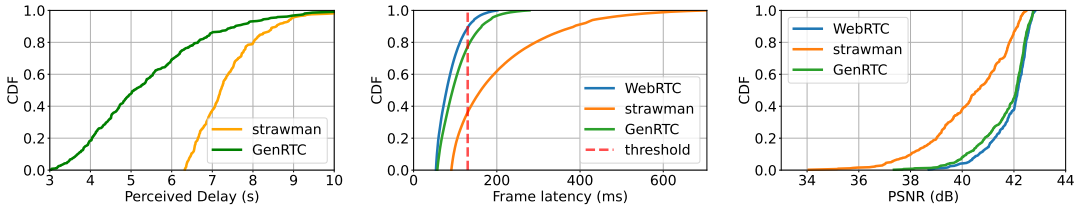
RQ3 Can GenRTC guarantee the performance across various video sources, network conditions, and devices?

RQ4 Would the streamers and audiences prefer vanilla WebRTC or GenRTC?

5.5.1 Experimental Setting

We deploy a server with high-performance network interfaces, achieving an average upload speed of 920.79 Mbps. In real-world tests, clients use cellular and Wi-Fi, while for simulation purposes, we generate synthetic network traces to emulate the clients' downlink connections. For generative models, we utilize Stable Diffusion v1.5 without further modifications.

Video Datasets. We use two video datasets featuring clear and continuous human speech in the audio. The first is VoxCeleb [252], a large-scale audio-video dataset of human speech. The second consists of live TV clips from NBC, CBS, ABC *etc.*, collected from the Puffer [213]. These datasets provide a variety of real-world communication scenarios. All datasets were labeled with intents and timestamps



(a) Perceived delay. (b) Frame latency. (c) Video quality.

Figure 5.9: Overall performance on two combined video datasets for different methods with ten streamers.

by human-computer interaction experts. In total, 127 videos were labeled, with an average duration of 25 minutes and 11 seconds [1].

Baselines. To the best of our knowledge, GenRTC is the first work that incorporates generative visual captions in live streaming. To demonstrate the superior performance, we use the two baselines [2]:

- *Vanilla WebRTC without Visual Captions.* Vanilla WebRTC provides ideally uninterrupted streaming experience since no generation functionalities are applied. This serves as a skyline to demonstrate the sacrifice involved in incorporating visual caption generation. However, without visual captions, the overall user experience may lack the enhanced clarity and engagement that visual elements bring.
- *Strawman: An Intergrated SOTA.* To validate the performance improvement over the latest industry practices, we integrate VisualCaption [217], Kubernetes containers [249], and PIFO [255], contributing optimizations in pipeline efficiency, multi-session handling, and scheduling respectively.

Metrics. The following metrics assess system efficiency:

- *Intent accuracy.* Predictions from the fine-tuned LLM are compared with expert-labeled intents as the ground truth to evaluate intent prediction accuracy.

- *Perceived delay.* The delay is the combination of network latency and generation

[1] We prefer long videos as they fully test the intent prediction functionality.

[2] New algorithms and models (e.g., ADD [253] and SANA-Sprint [254]) are not included as baselines since we focus on system optimization. On the other hand, any improvements in model speed or quality can be seamlessly integrated into GenRTC.

	Resolution	GPUs	Network
#1	1280x720	A100	Ethernet
#2	854x480	2xA6000	Wi-Fi
#3	640x360	A6000	Cellular [257]
#4	426x240	3080 Ti	

Table 5.1: Settings in the generalizability study.

runtime. Therefore, the measured delay is $T_{up} + T_{gen} + T_{down}$, where T_{gen} represents delay that occurred on the media server.

- *Frame latency.* We measure the end-to-end delay of successfully received frames. As different applications pose varied deadline requirements, from 50 ms to 150ms. we choose 130 ms as a reference [234].

- *Video quality.* The peak signal-to-noise ratio (PSNR) is used to quantify the received video quality.

- *Intent Rendering Rate.* It measures the ratio of rendered captions to recognized intents. The next intent serves as the deadline for the current one. Failures include unprocessed intents or those that become outdated during generation, while successes are counted only when the intended prompts are successfully rendered. A higher ratio indicates that the system effectively generates and displays the correct visual content in real time across multiple streams.

These metrics exclude cases where GenRTC accurately predicts intents, but the rendered image does not align with the intent. This exclusion is justified for two reasons: First, modern text-to-image models generate high-quality images due to training on large-scale datasets [256]. These models consistently produce relevant, high-fidelity images, minimizing dissatisfaction related to intent alignment. Second, in our two case studies, participants were instructed to evaluate GenRTC, resulting in high satisfaction with the system.

5.5.2 Overall Performance

All videos in the dataset are randomly sampled for the experiments in this section. Ten virtual streamers upload live video at the same time. An A100 GPU serves generation.

Perceived Delay. Figure 5.9(a) illustrates the perceived delay for both the *strawman* and GenRTC. Since the visual caption generation process takes about 4 seconds, the strawman implementation shows a median perceived delay of 7.21 seconds, which is due to its lack of optimization in predictive intent analysis and GPU resource scheduling. In contrast, GenRTC achieves a much lower median perceived delay of 5.08 seconds, demonstrating its more efficient handling of real-time caption generation and transmission. Moreover, GenRTC occasionally obtains perceived delay shorter than the inference time, thanks to the predictive intent analysis module.

Frame Latency. We compare the frame latency of GenRTC with two baselines. In Figure 5.9(b), the median frame latency for *strawman GenRTC* is 257.91 ms, which is 2.67 times higher than the 79.22 ms of *Vanilla WebRTC*. Additionally, during continuous GPU full-duty periods, a noticeable long-tail distribution occurs, with 95-th and 99-th percentile latencies reaching 551.01 ms and 815.89 ms, respectively. However, the optimizations in GenRTC reduce the median latency to 93.78 ms, with the 95-th and 99-th percentile latencies dropping to 187.72 ms and 238.44 ms, respectively. This results in only an 18% increase compared to *Vanilla WebRTC*.

Video Quality. In Figure 5.9(c), the video quality of three different methods is compared. *Vanilla WebRTC* provides the best video quality, serving as the benchmark with a mean PSNR of 41.87 dB. In contrast, the *strawman* implementation performs significantly worse, with a mean of 40.31 dB, primarily due to the lack of optimization in handling visual tasks. However, *GenRTC* demonstrates much better performance, closely approaching the baseline with a mean of 41.68 dB. This improvement in GenRTC highlights the benefits of optimizations that mitigate the impact on video quality.

Intent Accuracy. Apart from LLMs, intent analysis can be performed using

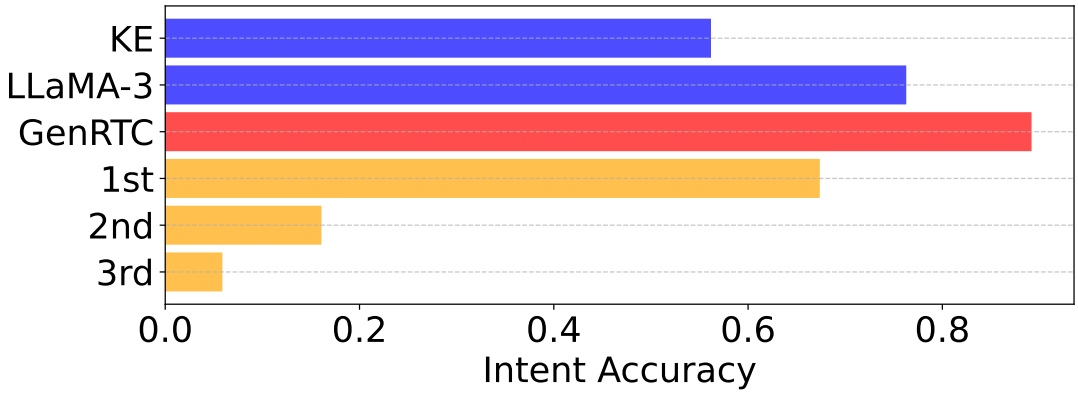


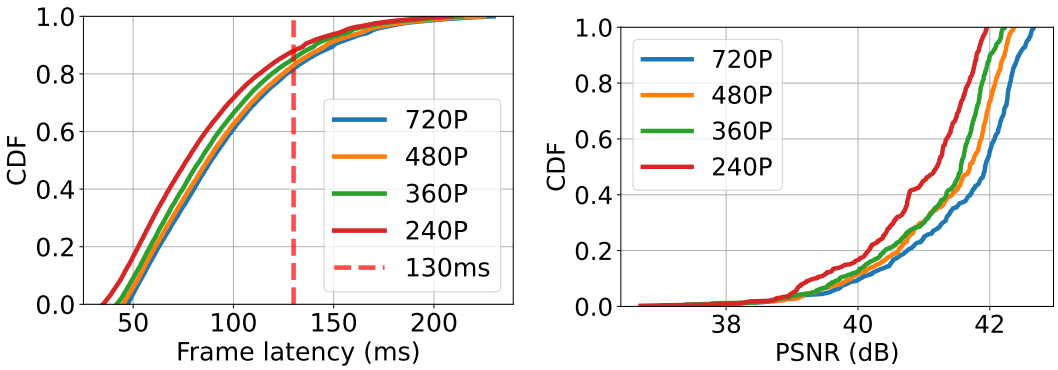
Figure 5.10: Overall performance of intention analysis.

traditional NLP algorithms. For example, the keyword extraction algorithm [258] generates similar text that can be used as prompts. We use this as a baseline, referred to as *KE*. Figure 5.10 compares the Top-3 intent accuracy of *KE* (56.1%), vanilla LLaMA-3 (76.2%), and GenRTC (89.1%). For GenRTC, the accuracy breakdown across each output rank is 67.3%, 16.0%, and 5.8%, respectively. This result highlights the effectiveness of our fine-tuned LLMs in better predicting the speaker's intent.

Intent Rendering Rate. We focus solely on whether the intent is generated and displayed on time, without considering correctness, as these depend on the performance of generative models. *GenRTC* achieves an intent rendering rate of 71.2%, while *vanilla WebRTC* reaches only 15.4%. This demonstrates that *GenRTC* significantly improves real-time intent rendering, due to its optimizations for handling visual captioning tasks. The detailed contribution of various factors to this difference is discussed in the ablation study in Sec 5.5.4.

Although the current intent rendering rate is optimized, a gap remains toward achieving full satisfaction. This is primarily due to sudden, short-lasting intents, such as a quick, novel topic that lasts for less than 3.81 seconds, which cannot be rendered in time. We attribute this limitation to current commodity hardware and believe that advances in MLPerf [259][3] will further enhance the integration of GenAI into live video streaming. For the 21.9% of intents that last less than 3.81 seconds, GenRTC still managed to render 14.0%, demonstrating the benefit

[3]NVIDIA reports 235.36 ms for SD image generation on H200 datacenters.



(a) Frame latency. (b) PSNR.

Figure 5.11: Impact of video resolutions.

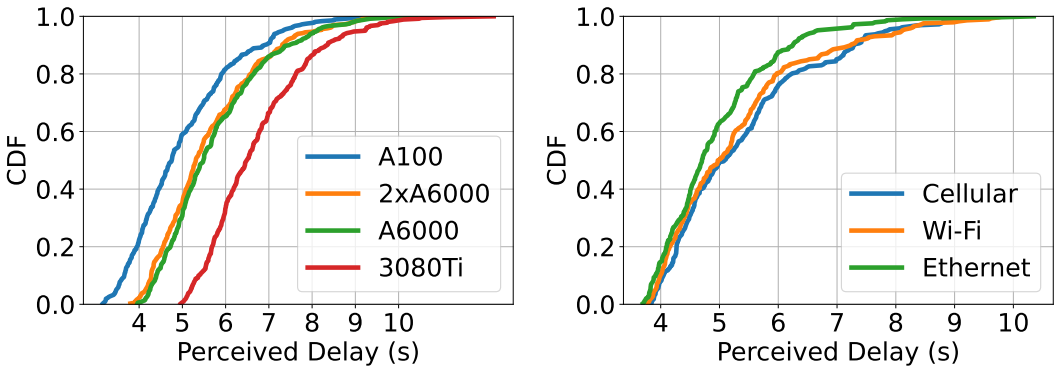
of predictive intention analytics.

5.5.3 Generalizability Study

We evaluate the robustness and adaptability of the system under varying audio and video quality conditions. Table 5.1 details the different settings. This evaluation is crucial to ensure that GenRTC performs effectively in real-world scenarios with diverse streaming quality requirements.

Video Resolution. Most videos in Voxceleb dataset have a limited resolution of 256×256. To rigorously test GenRTC's capability to handle various video qualities, we utilize pre-recorded TV clips from Puffer [213]. The audio specifications do not affect QoE due to the minimal bitrate. To evaluate the generalizability of GenRTC with different video resolutions, we tested four levels: 1280x720, 854x480, 640x360, and 426x240. The results indicate that the impact on perceived delay is within 1%, which is negligible. As shown in Figures 5.11(a) and 5.11(b), the frame latency and PSNR remain consistently high at all resolution levels.

GPU Resources. To assess the impact of GPU resources on QoE, we conduct comparison experiments using 1xA100, 2xA6000, 1xA6000, and 1xRTX 3080 Ti. The number of virtual streamers is still 10. The number of virtual streamers is set to 5 for this study. As shown in Figure 5.12(a), the perceived delay generally follows the generation time. For the 2xA6000 setup, 14.5% of GPU time remains unused since we did not double the number of streamers. Additionally, we record



(a) GPU resources.

(b) Network conditions.

Figure 5.12: Impact of GPU resources and network conditions.

the intent rendering rate for each setting, with the A100 being the highest because the minimum inference time is the lowest. Given that the rental cost of 1xA100 is similar to that of 2xA6000 [260, 261], we recommend prioritizing performance by using 1xA100 to enhance user experience and maximize GPU utilization. We observe that lower generation times lead to higher GPU utilization.

Network Condition. To evaluate whether GenRTC adapts to different network environments, we gathered two network traces using Mahimahi [262] for Ethernet and WiFi. FCC cellular traces [257] were also included. These traces capture variations in bandwidth, latency, and packet loss to simulate real-world scenarios. Performance is measured in terms of perceived delays. Figure 5.12(b) shows that network conditions have a negligible impact on perceived delay. GenRTC maintains high performance, with average perceived delays of 4948.6 ms on Ethernet, 5280.8 ms on WiFi, and 5385.0 ms on cellular, demonstrating robustness across various network conditions.

Number of Streamers. Figure 5.13 illustrates the scalability of GenRTC under varying loads. We set the number of streamers to 5, 10, 15, and 20. Each virtual streamer uploads a video file with a maximum resolution of 720p. The hardware setup serves 1, 2, 3, and 4 A100s GPUs for fairness. Note that with 5 users, the GPUs were not fully utilized, with 16.3% of duty time remaining idle.

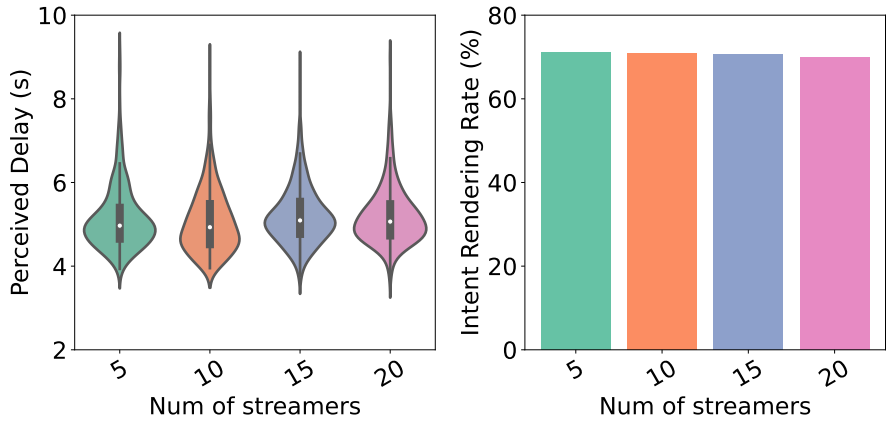


Figure 5.13: Perceived delay and intent rendering rate of GenRTC across varying numbers of streamers.

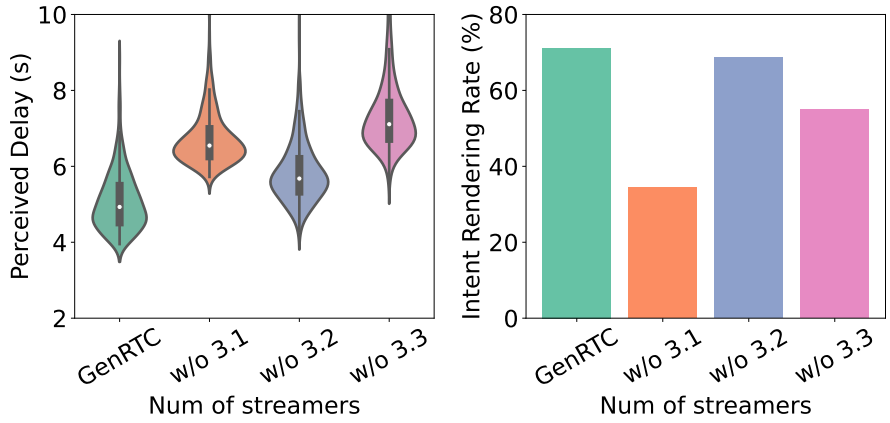


Figure 5.14: The perceived delay and GPU utility rate of GenRTC when each design module is removed.

5.5.4 Ablation Study

To validate the contribution of each component, we deactivate the following design components individually: continuous intention analysis and planning, codec-aware rendering scheduler, and dynamic resource allocation mechanism. This allows us to evaluate their respective contributions to overall system performance. We measure system latency from audio input to visual output, computational overhead through GPU and CPU usage, and visual fidelity based on the clarity and relevance of generated visuals in controlled user tests

Figure 5.14 presents the results of each ablated version. Replacing the contin-

uous intention analysis module with a naive LLM increases the average perceived delay from 5.06s to 6.65s and drastically reduces intent rendering rate from 71.2% to 34.5%. Disabling the codec-aware rendering scheduler results in an slight increase in perceived delays and a 2.6% decrease in the intent rendering rate. Lastly, removing the dynamic resource allocation mechanism leads to suboptimal resource management during peak demands, causing perceived delay to rise by 46.2% and obtaining intent rendering delay as 55%. These findings affirm the critical role of each proposed module in GenRTC architecture.

5.5.5 Case Study: Live Video Streaming

To gain qualitative insights into user experiences in live video streaming, we conducted a user study involving 20 volunteers from diverse backgrounds. Participants interacted with GenRTC through standard streaming scenarios, with 5 of them also participating as streamers, giving free talks and watching their own echo video in real time, which is common practice for streamers. Volunteers used three types of devices including laptops, desktops, and smartphones, to access GenRTC via the Web. For the audience, we played 5 randomly selected videos from the dataset, covering three categories: sports events, talk shows, and variety shows. Volunteers experienced all three schemes in random order and rated each on a 5-point scale `(1=very bad, 2=bad, 3=fair, 4=good, 5=very good)`.

Figure 5.15 presents the results. Their feedback highlights the practical impact of our technological enhancements and helps gauge user satisfaction in real-world conditions, making it a crucial part of our system evaluation. An additional discovery is that, from a linguistic perspective, visual captions influence the speaking patterns of streamers. For example, streamers may reference the visual captions to continue their speech. This represents a key distinction between trace-driven simulations and real user studies.

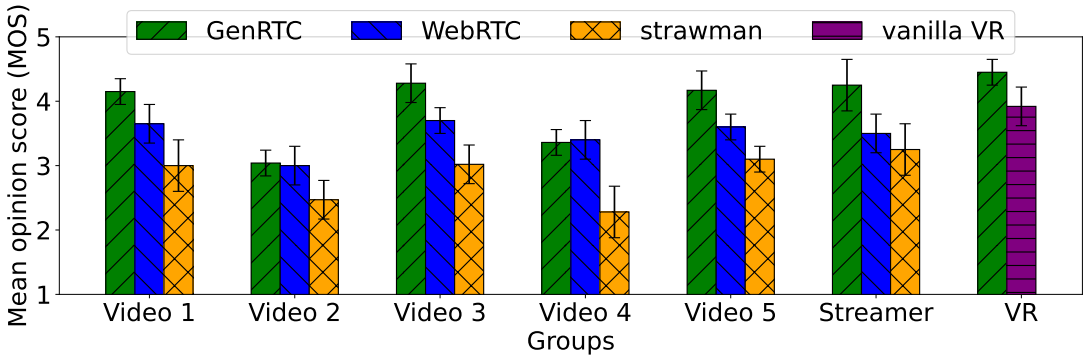


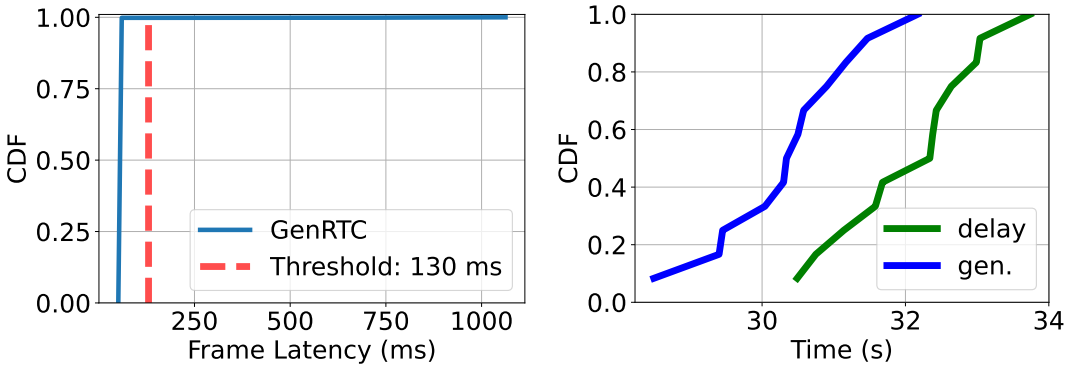
Figure 5.15: The user preference in MOS for live streaming audiences, streamers making free speech, and VR game players.

Figure 5.16: The 3D generation effect in the VR collaborative game [1]. The generated object is highlighted.

5.5.6 Case Study: Collaborative VR Scenario

Collaborative VR offers unique opportunities for GenRTC, especially due to its immersive nature and the crucial need for 3D innovations and creations [263, 264]. However, the generated objects need to be 3D to be campatible with the environments, which poses a more harsh challenge compared to images. This case study explores the application of GenRTC in VR scenarios, evaluating its performance in terms of frame latency and perceived delay.

Setup. The evaluation takes place in a controlled VR environment where two participants engage in an interactive session. Users operate Meta Oculus 2 [265] headsets, connected to the server via USB, with Netcode [266] applied to simulate a one-way latency of 50 ms. Since there are no specific frame definitions in VR



(a) Frame latency: 99.18% of measurements are less than 65ms.

(b) Perceived delay and generation inference time.

Figure 5.17: The frame latency and perceived latency of 3D generation with Rodin in VR collaboration game.

volumetric streaming, we measure perceived delay and frame latency using the Unity API [266]. The two users form a 1-on-1 conferencing structure, where both act as streamers and audience members. However, the upstream consists of only sparse command flow within the game. Additionally, the object is separated from the VR stream, so the renderer component is disabled.

Model Selection. Text-to-3D models are computationally intensive for both training and inference. We test the pre-trained models Prolificdreamer [267] and DreamGaussian [268] on our server, as well as the closed-source model Rodin-Gen-1 [269] via API calls. The first two models take over 2 minutes to run, while Rodin completes the generation in approximately 30 seconds. Due to its superior performance, we selected Rodin for the following experiments. The rendered mesh and texture are stored in the Unity project and automatically activated for display. As shown in Figure 5.16, user 1 expresses the need for a bowl of sausage slices, triggering the GenRTC workflow with the Rodin-Gen-1 model.

Through the user study, 4 users form 2 pairs, with 24 objects generated. Figure 5.17 shows that 99.18% of frame delay is less than 65ms, because only the packets for sparse human movements and audio are transmitted. However, when the generated object first occurs, there is a traffic burst due to bulk transfer, in which the average frame latency is 271.4ms. The perceived delay for this case is 32.1

seconds, with 30.4 seconds spent on generation inference. This result demonstrates that, in 3D object generation, the majority of the delay is due to the generation process rather than to system flaws.

This case study underscores the potential of GenRTC to enhance interactive real-time applications in VR, suggesting avenues for further research and development in this rapidly evolving field. This structured assessment allows us to validate the effectiveness of GenRTC under various quality metrics, ensuring that the system's performance is consistent and reliable regardless of the input quality. Future works can trivially extend GenRTC to Augmented Reality (AR) [270] as well.

5.6 Related Work

Latency Optimization for Live Video Streaming. Predictive content can boost timeliness. ZGaming [215] reduces the latency of cloud gaming by predicting the foreground of the game scenes. Farfetchfusion [271] uses a disentangled fusion approach and topology anchoring to overcome the challenges of 3D reconstruction latency and computational complexity on mobile devices. Super-resolution is another popular machine learning pipeline for multimedia transport systems [272]. However, most of them are performed on the edges and therefore do not raise latency issues [273]. The CV domain also highlights generation speed and efficiency to approach the real-world live streaming needs [274], but the generation time is far from acceptable. Nevertheless, GenRTC will benefit from any future optimizations on generative models.

Generative Models for Visual Object. The generation can be categorized into three major kinds: object, scene, and human [275]. Rodin [269] and HeadNeRF [276] generate human head avatars. DreamFusion [277] utilizes a pre-trained 2D text-to-image diffusion model to perform general text-to-3D synthesis. Prolificdreamer [267] refines score-distillation sampling to generate high-resolution 3D objects. Other works explore the systematical challenges of adopting 3D generation, especially in edges and mobile devices. Zhang et al. [278] explores how

to enable text-3D generation on resource-constrained mobile devices. CoDi [279] and GaussianDreamer [268] improve generation speed, but it still takes more than 2 minutes. DiffAgent [280] selects the best text-to-image API in seconds to better match needs, but this selection latency is unacceptable. Face-GPS quantify facial muscle dynamics in videos for medical applications [281, 282] Besides, music is used as another clue to adjust video contents. Music-driven image animation synchronizes visual content with audio cues [283], while MuseChat demonstrates the power of conversational music recommendation for videos [284]. Despite being promising, they also need extra inference time and can't achieve real-time. Our work pushes a steady step towards its application.

GPU Resource Scheduling. The efficient scheduling of limited GPU resources for multiple sessions is critical [285, 286, 287, 288]. For example, Gravel [286] assigns jobs to heterogeneous GPUs. Usher [288] enhances GPU utilization for ML inference by optimizing resource allocation and reducing inter-model interference. CacheSack [287] improves flash cache efficiency by optimizing admission policies, reducing disk reads, flash writes, and overall costs. Zhao et al. [289] train job placement by optimizing resource allocation with statistical INA, improving efficiency and job completion times. Following the optimization pattern of these works, GenRTC formulates GPU scheduling as an online knapsack problem and applies the heuristics from LLMs to form an MPC-based solution to maximize the expectation of intent rendering rate.

Compatibility with Recent Live Streaming Advances. GenRTC is compatible with recent advances in neural compression and loss recovery frameworks, as mentioned below: Gemino [219] applies neural compression for video conferencing, reducing bandwidth usage while maintaining high-quality video at low bitrates. Tambur [224] improves videoconferencing by using streaming codes for efficient loss recovery, reducing freezes and bandwidth overhead. Grace [225] optimizes video transmission by optimizing neural codecs for loss recovery, reducing undecodable frames and bandwidth usage. For these works, GenRTC may transmit the visual captions in a side datachannel and combine with the frames after decoding. Hairpin [227] enhances interactive video streaming by optimizing retransmissions

and redundancy, reducing bandwidth costs and deadline misses. The frame rate adaptation is proposed by AFR [226] to optimize tail latency by introducing the frame rate adaptation mechanism. Other tracedriven and environmentaware bitrate adaptation schemes advance mobile video streaming by dynamically aligning encoding rates with cellular conditions [290, 291]. Yuzu [292] improves volumetric video streaming with optimized 3D super-resolution. MuV2 [293] enhances multi-user volumetric video streaming by optimizing resource use through content hybridization, view sharing, and encoder multiplexing. GenRTC can compatibly benefit from these innovations to further reduce the latency. Overall, there is a huge systematical gap between these novel generating functionalities and product-level adoption, which this work fills.

Future Directions in Multi-modal Perspective. Future works can be extended for multiple aspects. For instance, eye gaze tracking [294, 270] can be adopted to conduct region-aware augmentation inside one frame. Retrieval-augmented generation(RAG) benefits certain tasks with external documents [295], which could be leveraged after we expand the dataset. The cross-modal workflow can be enhanced with adaptive feature aggregation for imagetext retrieval[296] and lexical grounding via definition-aware LLMs[297]. Vision-language models can be adopted to fully understand video frames and and make visual captions more user-friendly [298]. The follow-up effects [299] can also be adopted for generated objects to maintain positional consistency during interactions, of which the orientation of objects can be accurately tracked [300, 301, 302]. On the audio side, audio-visual models can analyze background music to provide more context [303]. We can also embed age-aware and health-aware voice analytics, including healthy-aging assessment, prosody profiling, and vocal age inference [304, 305, 306], to better register speaker profiles [307] in multi-user scenarios.

Automating System Configurations and Workflows. The recent surge in applying large language models to automate system configuration and optimization suggests several concrete directions. For instance, integrating LLMs into HPC workflows [308], like HPC-GPT [309], LM4HPC [310], and OMPGPT [311], offers the potential to automate parallelization directives and optimize code generation.

HumanAI interaction studies span evaluation of chatbotsuggestion interfaces [312] and RLTHFs targeted feedback for largelanguagemodel alignment [313]. Besides, LLM-driven DevOps [314, 315] can automate Kubernetes configuration at scale. Moreover, automated translation of legacy code [316], and data-race detection using LLMs [317] point toward robust debugging and upgrading pipelines. Enhancing the program parallelism can further improve the architecture-level efficiency [318, 319]. Federated learning [320, 321, 322, 323] further facilitates geodistributed system patterns. Tailored LLMs can facilitate automating diverse AIoT applications, as illustrated in DomAIn, AutoIOT and GPIoT [324, 325, 326].

System Performance Optimization. Efficient networkdatamining work uses unbiased sketch structures to pinpoint topk items in highspeed streams, like WavingSketch [327], PiSketch [328], and a realtime traffic sketch [329]. For cellular networks, applicationlayer latency trimming [330], simulatorbased failure recovery with Seed [331], and the rootfree diagnostic framework LDRP [332] drive 5G performance gains. Deep reinforcement learning optimizes cyberphysical systems, covering VM rescheduling [333], interpretable HVAC control [334, 335, 336], adaptive multimodal computation [337], and mobile analytics for map matching [338], appusage prediction [339], and temporalpointprocess forecasting [340]. More works spans kernel storage acceleration, highperformance compilation, garbagecollection, and nextgen distributed storage. XRP moves key storage functions into eBPF to cut kernel crossings and boost I/O throughput [341], while ROLLER streamlines tensoroperator codegen for deeplearning backends [342]. A line of Java research brews NVMaware heaps (Espresso [343]), trims GC pause tails for interactive services (Platinum [344]), and devises compaction heuristics and analyses for fullheap collection (Analysis [345], ScissorGC [346]). At the storage layer, RubbleDB shows CPUefficient NVMeoF replication with lightweight userspace processing [347], and a followup enables strongly consistent writeback page caching for distributed FUSE file systems [348].

5.7 Conclusion

This paper proposes GenRTC, a system that advances real-time video streaming experience by seamlessly integrating visual captions into live video streams. By continuously analyzing the speech intents, GenRTC translates speech into visual representations in real-time, enhancing both user engagement and comprehension. The system's robust handling of latency, scalability, and resource management ensures it meets the stringent demands of live streaming environments. Extensive evaluation on dataset-driven experiments, user studies and VR case studies shows that GenRTC can achieve a high level of efficiency and responsiveness, providing the captions without compromising latency and video quality.

Chapter 6

Future Work: Building Multi-modal Foundation Models for Wireless Spectrum Signals

Looking ahead, this dissertation opens several promising directions for future exploration at the intersection of artificial intelligence and networked systems. In particular, we aim to push the boundary further by investigating the scaling behavior of multi-modal large language models (MLLMs) when extended to wireless modalities.

6.1 Challenges

Existing works apply diffusion and radiance-field techniques to wireless signal generation and corresponding tasks like localization [349, 350, 351, 352]. However, these works are case-by-case designs. Unlike traditional modalities such as images and text [353, 354, 355], building a foundation model for wireless signals present unique modeling challenges due to their inherent physical and hardware constraints. These challenges include:

- **Heterogeneous Input Formats**: Data collected across different frequency bands, hardware platforms, and communication standards often vary in resolu-

tion, structure, and encoding schemes. This heterogeneity introduces non-trivial modality alignment issues that are not encountered in image or text data.

- **Sparsity in the Spectrum**: In many wireless communication scenarios, particularly in wideband and low-activity channels, the informative content is sparsely distributed across time and frequency. Learning meaningful representations under such sparsity requires the model to capture subtle temporal and spectral correlations.

- **Dual-Domain Insight Requirement**: Key information is embedded across both the time domain (e.g., signal waveform dynamics) and the frequency domain (e.g., channel and interference patterns). Effectively learning from and reasoning about these domains simultaneously calls for novel architectural innovations.

6.2 Solution Preview

6.2.1 Dataset Construction

The training and evaluation of large models both pose significant challenges for dataset size and quality. We aim to collect, curate, and ground a high-quality multi-modal dataset, representing diverse wireless communication patterns across realistic settings, similar to Nie et al. [356], but covering diversity in more applications and technical aspects, including diverse frequency bands, dimension definitions, modulation technologies and hardware specifications. In addition, data augmentation can further improve the quantity of the dataset [357, 358], but the quality drop should be carefully monitored.

6.2.2 Model Training and Optimization

Future research will explore the scaling laws and emergent capabilities of such models in three complementary stages:

- **Pretraining**: Investigating how model size, data diversity, and task mixtures affect generalization in spectrum-aware MLLMs. We plan to benchmark whether performance improvements follow logarithmic, power-law, or other scaling behaviors observed in vision and language models.

- **Post-training**: Exploring methods such as instruction tuning, multi-modal alignment, and domain-specific adaptation to enhance model utility in specialized wireless applications (e.g., interference mitigation, anomaly detection, or protocol classification).

- **Inference-Time Optimization**: Developing efficient inference strategies that enable real-time and on-device deployment, such as model pruning, sparsity-aware inference, and low-rank adaptation under hardware constraints.

Sub-model Proposals: The features in different context can be processed by various sub-models [359]. For instance, we can incorporate time-domain expert (T-Agent, implemented as an i-Transformer backbone [120]), a pixel-level expert (P-Agent, realized by a ViT-Window backbone [360, 361, 362] with 2-D RoPE), and a frequency-domain expert (F-Agent, using a patch-wise transformer with an FFT-based periodicity module [122]). Diffusion models

We will also broaden our future agenda along three LLM-centric directions. First, we aim to apply dynamic soft prompting and domain-specific pruning to tailor lightweight language models for edge and networked deployments, building on recent advances in memorization control and structural compression [363, 364]. Second, the knowledge-injected graph neural networks [365] can be adopted to fuse domain ontologies with spectrum data to improve anomaly attribution in large-scale systems. Third, we will investigate synthetic-data generation and intermittent inference to support low-power, self-sustaining devices: synthetic visual corpora for representation learning [366] and energy-harvesting-aware network design for on-device reasoning [367]. Together, these threads will equip our next-generation models with better efficiency, adaptability, and resilience in wireless spectrum signal understanding.

To summarize, the long-term vision is to empower AI systems to use networked

infrastructure as a perceptual window into the physical world, not only sensing but also interpreting, reasoning, and controlling. By equipping AI with the ability to reason over multi-modal signals from networked systems, we take a step toward making machines that are not just connected, but contextually aware, trustworthy, and intelligent collaborators in real-world environments.

Chapter 7

Conclusions

This dissertation presents a unified exploration of how artificial intelligence can be practically and effectively embedded into networked systems to enable intelligent sensing, modeling, control, and interaction. Through four core case studies: LoRa link quality modeling in orchards, groundwater recharge optimization, causality-aware time series modeling, and real-time generative visual captioning, it demonstrates how AI can be adapted to meet the physical, temporal, and computational constraints of real-world environments. Each chapter addresses a distinct challenge: signal degradation in complex propagation environments, long-term decision-making via effective forecasting, learning interpretable causal structures from time series, and real-time intent analysis and demo generation inference under resource limitations. Across these scenarios, the dissertation shows that domain knowledge, physical constraints, and systems-level considerations must be tightly coupled with machine learning design to achieve deployable and trustworthy solutions.

Collectively, the findings contribute toward a broader vision where AI does not operate in isolation, but instead serves as an intelligent layer over networked infrastructure: perceiving, interpreting, and interacting with the physical world. This work lays the groundwork for a new class of AI systems that are resource-aware, domain-aligned, and capable of operating at the edge or within the loop. The dissertation also highlights the importance of interpretability and robustness when deploying AI in critical applications like agriculture and communication. As

networked systems become increasingly intelligent and multimodal, future efforts will build on these foundations to explore scalability, generalization, and collaboration across modalities. Ultimately, this dissertation argues for a co-design approach where systems and AI evolve together to unlock new levels of responsiveness, efficiency, and understanding in real-world applications.

Bibliography

[1] Javad Sameri, Sam Van Damme, Susanna Schwarzmann, Qing Wei, Riccardo Trivisonno, Filip De Turck, and Maria Torres Vega. Collaborative Cooking in VR: Effects of Network Distortion in Multi-User Virtual Environments. In *ACM MMSYS*, 2024.

[2] Yuxiao Cheng, Ziqian Wang, Tingxiong Xiao, Qin Zhong, Jinli Suo, and Kunlun He. Causaltime: Realistically generated time-series for benchmarking of causal discovery. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] Yifei Xu, Xumiao Zhang, Yuning Chen, Pan Hu, Xuan Zeng, Zhilong Zheng, Xianshang Lin, Yanmei Liu, Songwu Lu, Z. Morley Mao, Wan Du, Dennis Cai, Ennan Zhai, and Yunfei Ma. Roaming free in the vr world with mp2. In *2025 USENIX Annual Technical Conference (USENIX ATC 25)*, 2025.

[4] Kang Yang, Yuning Chen, and Wan Du. Flog: Automated modeling of link quality for lora networks in orchards. *ACM Transactions on Sensor Networks*, 21(2):1–28, 2025.

[5] Kang Yang, Yuning Chen, Xuanren Chen, and Wan Du. Link quality modeling for lora networks in orchards. In *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, pages 27–39, 2023.

[6] United States Department of Agriculture, National Agricultural Statistics Service. California agricultural production statistics. `https://www.cdfa.ca.gov/Statistics/`.

[7] Xianzhong Ding and Wan Du. DRLIC: Deep Reinforcement Learning for Irrigation Control. In *ACM/IEEE IPSN*, 2022.

[8] Younsuk Dong, Benjamin Werling, Zhichao Cao, and Gen Li. Implementation of an in-field iot system for precision irrigation management. *Frontiers in Water*, 6:1353597, 2024.

[9] Tuan Dang, Trung Tran, Khang Nguyen, Tien Pham, Nhat Pham, Tam Vu, and Phuc Nguyen. iotree: a battery-free wearable system with biocompatible sensors for continuous tree health monitoring. In *ACM MobiCom*, 2022.

[10] Zhizhang Hu, Shangjie Du, Yuning Chen, Xuan Zhang, Wan Du, Asa Bradman, and Shijia Pan. Enhancing fault resilience of air quality monitoring in san joaquin valley: A data equity analysis. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, pages 514–515, 2023.

[11] Michele Preti, François Verheggen, and Sergio Angeli. Insect pest monitoring with camera-equipped traps: strengths and limitations. *Journal of pest science*, 94(2):203–217, 2021.

[12] Abhishek Khanna and Sanmeet Kaur. Evolution of internet of things (iot) and its significant impact in the field of precision agriculture. *Computers and electronics in agriculture*, 157:218–231, 2019.

[13] Yas Hosseini Tehrani, Arash Amini, and Seyed Mojtaba Atarodi. A tree-structured LoRa network for energy efficiency. *IEEE Internet of Things Journal*, 8(7):6002–6011, 2020.

[14] Fudong Lin, Kaleb Guillot, Summer Crawford, Yihe Zhang, Xu Yuan, and Nian-Feng Tzeng. An open and large-scale dataset for multi-modal climate change-aware crop yield predictions. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 5375–5386, 2024.

[15] Kang Yang. *Advanced LoRa Networking and Sensing for IoT Applications*. PhD thesis, University of California, Merced, 2024.

[16] Marco Cattani, Carlo Alberto Boano, and Kay Römer. An experimental evaluation of the reliability of LoRa long-range low-power wireless communication. *Journal of Sensor and Actuator Networks*, 6(2):7, 2017.

[17] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. PLoRa: A passive long-range data network from ambient LoRa transmissions. In *ACM SIGCOMM*, 2018.

[18] Xiuzhen Guo, Longfei Shangguan, Yuan He, Nan Jing, Jiacheng Zhang, Haotian Jiang, and Yunhao Liu. Saiyan: Design and Implementation of a Low-power Demodulator for {LoRa} Backscatter Systems. In *USENIX NSDI*, 2022.

[19] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. NELoRa: Towards Ultra-low SNR LoRa Communication with Neural-enhanced Demodulation. In *ACM SenSys*, 2021.

[20] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. LMAC: Efficient carrier-sense multiple access for LoRa. In *ACM MobiCom*, 2020.

[21] Xianjin Xia, Yuanqing Zheng, and Tao Gu. FTrack: Parallel decoding for LoRa transmissions. In *ACM SenSys*, 2019.

[22] Jothi Prasanna Shanmuga Sundaram, Wan Du, and Zhiwei Zhao. A survey on LoRa networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials*, 22(1):371–388, 2019.

[23] Martin C. Bor, Utz Roedig, Thiemo Voigt, and Juan M. Alonso. Do LoRa Low-Power Wide-Area Networks Scale? In *ACM MSWiM*, 2016.

[24] Orestis Georgiou and Usman Raza. Low power wide area network analysis: Can LoRa scale? *IEEE Wireless Communications Letters*, 6(2):162–165, 2017.

[25] Manuel Eugenio Morocho-Cayamcela, Martin Maier, and Wansu Lim. Breaking wireless propagation environmental uncertainty with deep learning. *IEEE Transactions on Wireless Communications*, 19(8):5075–5087, 2020.

[26] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. Automated estimation of link quality for LoRa: A remote sensing approach. In *ACM/IEEE IPSN*, 2019.

[27] Li Liu, Yuguang Yao, Zhichao Cao, and Mi Zhang. DeepLoRa: Learning accurate path loss model for long distance links in LPWAN. In *IEEE INFOCOM*, 2021.

[28] Mark A Weissberger. An initial critical summary of models for predicting the attenuation of radio waves by trees. Technical report, Electromagnetic Compatibility Analysis Center Annapolis MD, 1982.

[29] International Telecommunication Union Radiocommunication. Attenuation in vegetation. 2001.

[30] MPM Hall. COST project 235 activities on radiowave propagation effects on next-generation fixed-service terrestrial telecommunication systems. In *IET ICAP*, 1993.

[31] Scipy. Non-linear Least-Square Algorithm. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html`.

[32] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *ACM SenSys*, 2003.

[33] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[34] Tallal Elshabrawy and Joerg Robert. Closed-form approximation of LoRa modulation BER performance. *IEEE Communications Letters*, 22(9):1778–1781, 2018.

[35] Artifacts of FLog. `https://github.com/ycucm/Flog`, 2023.

[36] Kang Yang and Wan Du. LLDPC: A Low-Density Parity-Check Coding Scheme for LoRa Networks. In *ACM SenSys*, 2022.

[37] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. Jamming of lora phy and countermeasure. *ACM Transactions on Sensor Networks (TOSN)*, 19(4):1–27, 2023.

[38] Luoyu Mei, Zhimeng Yin, Shuai Wang, Xiaolei Zhou, Taiwei Ling, and Tian He. Ecrlora: Lora packet recovery under low snr via edge–cloud collaboration. *ACM Transactions on Sensor Networks (TOSN)*, 20(2):1–25, 2024.

[39] Wan Du, Jansen Christian Liando, Huanle Zhang, and Mo Li. When pipelines meet fountain: Fast data dissemination in wireless sensor networks. In *ACM SenSys*, 2015.

[40] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *ACM MobiCom*, 2015.

[41] Yuxiang Lin, Wei Dong, Yi Gao, and Tao Gu. Sateloc: A virtual fingerprinting approach to outdoor LoRa localization using satellite images. *ACM Transactions on Sensor Networks (TOSN)*, 17(4):1–28, 2021.

[42] Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey, 1996.

[43] Wankai Tang, Ming Zheng Chen, Xiangyu Chen, Jun Yan Dai, Yu Han, Marco Di Renzo, Yong Zeng, Shi Jin, Qiang Cheng, and Tie Jun Cui. Wireless communications with reconfigurable intelligent surface: Path loss modeling and experimental measurement. *IEEE Transactions on Wireless Communications*, 20(1):421–439, 2020.

[44] Tossaporn Srisooksai, Kamol Kaemarungsi, Junichi Takada, and Kentaro Saito. Path loss measurement and prediction in outdoor fruit orchard for

wireless sensor network at 2.4 ghz band. *Progress In Electromagnetics Research C*, 90:237–252, 2019.

[45] Saul A Torrico, Henry L Bertoni, and Roger H Lang. Modeling tree effects on path loss in a residential environment. *IEEE Transactions on Antennas and Propagation*, 46(6):872–880, 1998.

[46] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[47] Han Na and Thomas F Eibert. A huygens' principle based ray tracing method for diffraction calculation. In *IEEE EuCAP*, 2022.

[48] Kun Yang, Andreas F Molisch, Torbjörn Ekman, Terje Røste, and Marion Berbineau. A round earth loss model and small-scale channel properties for open-sea radio propagation. *IEEE Transactions on Vehicular Technology*, 68(9):8449–8460, 2019.

[49] Semtech SX1276 datasheet. `https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276`, 2020.

[50] Arduino Uno Rev3. `https://store-usa.arduino.cc/products/arduino-uno-rev3/?selectedStore=us`, 2021.

[51] RAKwireless Technology. Rak831 pilot gateway product specification v1.3. `https://www.thethingsnetwork.org/docs/gateways/rak831/`.

[52] LoRa Packet Logger. `https://github.com/Lora-net/lora_gateway`, 2017.

[53] Erich Zöchmann, Ke Guan, and Markus Rupp. Two-ray models in mmWave communications. In *2017 IEEE SPAWC*, 2017.

[54] United States Department of Agriculture National Agricultural Statistics Service. 2021 California Almond Acreage Report. `https://www.almonds.com/sites/default/files/2022-04/2021_NASS_Acreage_Report.pdf`.

[55] Pedro M Santos, Traian E Abrudan, Ana Aguiar, and Joao Barros. Impact of position errors on path loss model estimation for device-to-device channels. *IEEE transactions on wireless communications*, 13(5):2353–2361, 2014.

[56] Richard L Snyder. California irrigation management information system. *American potato journal*, 61:229–234, 1984.

[57] Michael Marcus and Bruno Pattan. Millimeter wave propagation: spectrum management implications. *IEEE Microwave Magazine*, 6(2):54–62, 2005.

[58] Yilin Liu, Shijia Zhang, Mahanth Gowda, and Srihari Nelakuditi. Leveraging the properties of mmwave signals for 3d finger motion tracking for interactive iot applications. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–28, 2022.

[59] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67, 2016.

[60] Yidong Ren, Li Liu, Chenning Li, Zhichao Cao, and Shigang Chen. Is LoRaWAN really wide? fine-grained LoRa link-level measurement in an urban environment. In *IEEE ICNP*, 2022.

[61] Weitao Xu, Jun Young Kim, Walter Huang, Salil S Kanhere, Sanjay K Jha, and Wen Hu. Measurement, characterization, and modeling of lora technology in multifloor buildings. *IEEE Internet of Things Journal*, 7(1):298–310, 2019.

[62] Zongxing Xie and Fan Ye. Self-calibrating indoor trajectory tracking system using distributed monostatic radars for large scale deployment. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 188–197, 2022.

[63] Giulio Maria Bianco, Romeo Giuliano, Gaetano Marrocco, Franco Mazzenga, and Abraham Mejia-Aguilar. LoRa system for search and rescue: Path-loss

models and procedures in mountain scenarios. *IEEE Internet of Things Journal*, 8(3):1985–1999, 2020.

[64] Iury da S Batalha, Andréia Vanessa Rodrigues Lopes, Wirlan Gomes Lima, Yuri HS Barbosa, Miércio Cardoso De Alcântara Neto, Fabricio JB Barros, and Gervásio PS Cavalcante. Large-Scale Modeling and Analysis of Uplink and Downlink Channels for LoRa Technology in Suburban Environments. *IEEE Internet of Things Journal*, 9(23):24477–24491, 2022.

[65] Adrian Zarnescu, Razvan Ungurelu, Mihai Secere, Gaudentiu Varzaru, and Bogdan Mihailescu. Implementing a large LoRa network for an agricultural application. In *IEEE EE&AE*, 2020.

[66] Masaharu Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE transactions on Vehicular Technology*, 29(3):317–325, 1980.

[67] Yoshihisa Okumura. Field strength and its variability in vhf and uhf land-mobile radio service. *Rev. Electr. Commun. Lab.*, 16:825–873, 1968.

[68] Jane Yen, Jianfeng Wang, Sucha Supittayapornpong, Marcos AM Vieira, Ramesh Govindan, and Barath Raghavan. Meeting slos in cross-platform nfv. In *ACM CoNEXT*, 2020.

[69] MS Karaliopoulos and F-N Pavlidou. Modelling the land mobile satellite channel: A review. *Electronics & communication engineering journal*, 11(5):235–248, 1999.

[70] Silvia Demetri, Gian Pietro Picco, and Lorenzo Bruzzone. LaPS: LiDAR-assisted placement of wireless sensor networks in forests. *ACM Transactions on Sensor Networks (TOSN)*, 15(2):1–40, 2019.

[71] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. Mitigating inter-network interference in LoRa networks. *arXiv preprint arXiv:1611.00688*, 2016.

[72] Clement Demeslay, Philippe Rostaing, and Roland Gautier. Theoretical performance of LoRa system in multipath and interference channels. *IEEE Internet of Things Journal*, 9(9):6830–6843, 2021.

[73] Ningning Hou, Yifeng Wang, Xianjin Xia, Shiming Yu, Yuanqing Zheng, and Tao Gu. Molora: Intelligent mobile antenna system for enhanced lora reception in urban environments. In *ACM SenSys*, 2025.

[74] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. Dont miss weak packets: Boosting lora reception with antenna diversities. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022.

[75] Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. Pcube: Scaling lora concurrent transmissions with reception diversities. *ACM Transactions on Sensor Networks*, 18(4):1–25, 2023.

[76] Xianjin Xia, Qianwu Chen, Ningning Hou, Yuanqing Zheng, and Mo Li. Xcopy: Boosting weak links for reliable lora communication. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023.

[77] Shiming Yu, Ziyue Zhang, Xianjin Xia, Yuanqing Zheng, and Jiliang Wang. Are lora logical channels really orthogonal? practically orthogonalizing massive logical channels. In *ACM MobiSys*, 2025.

[78] Shiming Yu, Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. Revolutionizing lora gateway with xgate: Scalable concurrent transmission across massive logical channels. In *ACM MobiCom*, 2024.

[79] Xianjin Xia, Qianwu Chen, Ningning Hou, Yuanqing Zheng, and Tao Gu. Hylink: Toward high throughput lpwans with lora compatible communication. *IEEE/ACM Transactions on Networking*, 32(4):33153330, April 2024.

[80] Xianjin Xia, Qianwu Chen, Ningning Hou, and Yuanqing Zheng. Hylink: Towards high throughput lpwans with lora compatible communication. In

Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems, 2022.

[81] Shiming Yu, Xianjin Xia, Ziyue Zhang, Ningning Hou, and Yuanqing Zheng. Fdlora: Tackling downlink-uplink asymmetry with full-duplex lora gateways. In *ACM SenSys*, 2024.

[82] Ruonan Li, Ziyue Zhang, Xianjin Xia, Ningning Hou, Wenchang Chai, Shiming Yu, Yuanqing Zheng, and Tao Gu. From interference mitigation to toleration: Pathway to practical spatial reuse in lpwans. In *ACM MobiCom 2025*, 2025.

[83] Ningning Hou, Xianjin Xia, Yifeng Wang, and Yuanqing Zheng. One shot for all: Quick and accurate data aggregation for lpwans. *IEEE/ACM Transaction on Networking*, 32(3):22852298, January 2024.

[84] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. Cloaklora: A covert channel over lora phy. *IEEE/ACM Transactions on Networking*, 31(3):1159–1172, 2022.

[85] Chenning Li, Yidong Ren, Shuai Tong, Shakhrul Iman Siam, Mi Zhang, Jiliang Wang, Yunhao Liu, and Zhichao Cao. Chirptransformer: Versatile lora encoding for low-power wide-area iot. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, pages 479–491, 2024.

[86] Yidong Ren, Puyu Cai, Jinyan Jiang, Jialuo Du, and Zhichao Cao. Prism: High-throughput lora backscatter with non-linear chirps. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[87] Jialuo Du, Yidong Ren, Mi Zhang, Yunhao Liu, and Zhichao Cao. Nelora-bench: A benchmark for neural-enhanced lora demodulation. *arXiv preprint arXiv:2305.01573*, 2023.

[88] Jialuo Du, Yidong Ren, Zhui Zhu, Chenning Li, Zhichao Cao, Qiang Ma, and Yunhao Liu. Srlora: Neural-enhanced lora weak signal decoding with multi-gateway super resolution. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 270–279, 2023.

[89] Jialuo Du, Yunhao Liu, Yidong Ren, Li Liu, and Zhichao Cao. Loratrimmer: Optimal energy condensation with chirp trimming for lora weak signal decoding. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 1104–1118, 2024.

[90] Yidong Ren, Gen Li, Yimeng Liu, Younsuk Dong, and Zhichao Cao. Aeroecho: Towards agricultural low-power wide-area backscatter with aerial excitation source. In *Proceedings of IEEE INFOCOM*, 2025.

[91] Yidong Ren, Amalinda Gamage, Li Liu, Mo Li, Shigang Chen, Younsuk Dong, and Zhichao Cao. Sateriot: High-performance ground-space networking for rural iot. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024.

[92] Yidong Ren, Wei Sun, Jialuo Du, Huaili Zeng, Younsuk Dong, Mi Zhang, Shigang Chen, Yunhao Liu, Tianxing Li, and Zhichao Cao. Demeter: Reliable cross-soil lpwan with low-cost signal polarization alignment. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024.

[93] Yuanlin Yang, Yuning Chen, Kang Yang, Sikai Yang, and Wan Du. Demo abstract: Comprehensive wireless soil component sensing via vnir and lora. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, pages 722–723, 2025.

[94] Yimeng Liu, Maolin Gan, Huaili Zeng, Yidong Ren, Gen Li, Jingkai Lin, Younsuk Dong, Xiaobo Tan, and Zhichao Cao. Proteus: Enhanced mmwave leaf wetness detection with cross-modality knowledge transfer. In *Proceedings*

of the 23rd ACM Conference on Embedded Networked Sensor Systems, pages 43–57, 2025.

[95] Yimeng Liu, Maolin Gan, Gen Li, Younsuk Dong, and Zhichao Cao. Adonis: Neural-enhanced fine-grained leaf wetness sensing with efficient mmwave imaging. In *Proceedings of IEEE INFOCOM*, 2025.

[96] Maolin Gan, Yimeng Liu, Li Liu, Chenshu Wu, Younsuk Dong, Huacheng Zeng, and Zhichao Cao. Poster: mmleaf: Versatile leaf wetness detection via mmwave sensing. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 563–564, 2023.

[97] Yimeng Liu, Maolin Gan, Huaili Zeng, Li Liu, Younsuk Dong, and Zhichao Cao. Hydra: Accurate multi-modal leaf wetness sensing with mm-wave and camera fusion. In *Proceedings of ACM MobiCom*, 2024.

[98] Huaili Zeng, Gen Li, and Tianxing Li. Pyrosense: 3d posture reconstruction using pyroelectric infrared sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–32, 2024.

[99] Zongxing Xie and Fan Ye. Scaling: plug-n-play device-free indoor tracking. *Scientific reports*, 14(1):2913, 2024.

[100] Mengjing Liu, Zongxing Xie, and Fan Ye. M4x: Enhancing cross-view generalizability in rf-based human activity recognition by exploiting synthetic data in metric learning. In *2024 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 49–60. IEEE, 2024.

[101] Zongxing Xie, Bing Zhou, and Fan Ye. Signal quality detection towards practical non-touch vital sign monitoring. In *Proceedings of the 12th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–9, 2021.

[102] Zongxing Xie, Bing Zhou, Xi Cheng, Elinor Schoenfeld, and Fan Ye. Passive and context-aware in-home vital signs monitoring using co-located uwb-

depth sensor fusion. *ACM transactions on computing for healthcare*, 3(4):1–31, 2022.

[103] Patrick Wu, Yiwen Dong, Chenhan Xu, Mark Geil, Modupe Akintomide, Xinyue Zhang, and Zongxing Xie. Wireless sensing of gait for neurodegenerative disease assessment: A scoping review. In *Proceedings of the 3rd International Workshop on Human-Centered Sensing, Modeling, and Intelligent Systems*, pages 58–63, 2025.

[104] Zongxing Xie, Ava Nederlander, Isac Park, and Fan Ye. Short: Rf-q: Unsupervised signal quality assessment for robust rf-based respiration monitoring. In *Proceedings of the 8th ACM/IEEE International Conference on Connected Health: Applications, Systems and Engineering Technologies*, pages 158–162, 2023.

[105] Zongxing Xie, Hanrui Wang, Song Han, Elinor Schoenfeld, and Fan Ye. Deepvs: A deep learning approach for rf-based vital signs sensing. In *Proceedings of the 13th ACM international conference on bioinformatics, computational biology and health informatics*, pages 1–5, 2022.

[106] Zongxing Xie, Yindong Hua, and Fan Ye. A measurement study of fmcw radar configurations for non-contact vital signs monitoring. In *2022 IEEE Radar Conference (RadarConf22)*, pages 1–6. IEEE, 2022.

[107] Zongxing Xie, Bing Zhou, Xi Cheng, Elinor Schoenfeld, and Fan Ye. Vitalhub: Robust, non-touch multi-user vital signs monitoring using depth camera-aided uwb. In *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pages 320–329. IEEE, 2021.

[108] Gen Li, Zhichao Cao, and Tianxing Li. Echoattack: Practical inaudible attacks to smart earbuds. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 383–396, 2023.

[109] Gen Li, Huaili Zeng, Hanqing Guo, Yidong Ren, Aiden Dixon, Zhichao Cao, and Tianxing Li. Piezobud: A piezo-aided secure earbud with practical speaker authentication. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, pages 564–577, 2024.

[110] Alvar Escriva-Bou, Brian Gray, Sarge Green, Thomas Harter, Richard Howitt, Duncan MacEwan, and N Seavy. Water stress and a changing san joaquin valley. *Public Policy Institute of California. https://www. ppic. org/content/pubs/report/R_0317EHR. pdf*, 2017.

[111] Scott Jasechko, Hansjörg Seybold, Debra Perrone, Ying Fan, Mohammad Shamsudduha, Richard G Taylor, Othman Fallatah, and James W Kirchner. Rapid groundwater decline and some cases of recovery in aquifers globally. *Nature*, 625(7996):715–721, 2024.

[112] Helen E Dahlke, Andrew G Brown, Steve Orloff, Daniel H Putnam, and Toby O'Geen. Managed winter flooding of alfalfa recharges groundwater with minimal crop damage. *California Agriculture*, 72(1), 2018.

[113] Yonatan Ganot and Helen E Dahlke. Natural and forced soil aeration during agricultural managed aquifer recharge. *Vadose Zone Journal*, 20(3):e20128, 2021.

[114] California Department of Water Resources. Sustainable Groundwater Management Act (SGMA). `https://water.ca.gov/programs/groundwater-management/sgma-groundwater-management`, 2014.

[115] Elad Levintal, Maribeth L Kniffin, Yonatan Ganot, Nisha Marwaha, Nicholas P Murphy, and Helen E Dahlke. Agricultural managed aquifer recharge (ag-mar)a method for sustainable groundwater management: A review. *Critical Reviews in Environmental Science and Technology*, 53(3):291–314, 2023.

[116] Richard G Niswonger, Eric D Morway, Enrique Triana, and Justin L

Huntington. Managed aquifer recharge through off-season irrigation in agricultural regions. *Water Resources Research*, 53(8):6970–6992, 2017.

[117] Jiangxiao Qiu, Samuel C Zipper, Melissa Motew, Eric G Booth, Christopher J Kucharik, and Steven P Loheide. Nonlinear groundwater influence on biophysical indicators of ecosystem services. *Nature Sustainability*, 2(6):475–483, 2019.

[118] Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 619(7970):526–532, 2023.

[119] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, page eadi2336, 2023.

[120] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[121] Siqiao Zhao, Zhikang Dong, Zeyu Cao, and Raphael Douady. Hedge fund portfolio construction using polymodel theory and itransformer. *arXiv preprint arXiv:2408.03320*, 2024.

[122] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations(ICLR)*, 2023.

[123] Fudong Lin, Xu Yuan, Yihe Zhang, Purushottam Sigdel, Li Chen, Lu Peng, and Nian-Feng Tzeng. Comprehensive transformer-based model architecture for real-world storm prediction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 54–71, 2023.

[124] Zhiyu An, Xianzhong Ding, Arya Rathee, and Wan Du. Clue: Safe model-based rl hvac control using epistemic uncertainty estimation. In *ACM BuildSys)*, 2023.

[125] Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J Zico Kolter. Differentiable mpc for end-to-end planning and control. *Advances in neural information processing systems*, 31, 2018.

[126] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[127] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[128] J Simnek, M Sejna, and M Th Van Genuchten. *The HYDRUS-2D software package for simulating the two-dimensional movement of water, heat, and multiple solutes in variably-saturated media: Version 2.0.* US Salinity Laboratory, Agricultural Research Service, 1999.

[129] Khaled M Bali, Abdelmoneim Zakaria Mohamed, Sultan Begna, Dong Wang, Daniel Putnam, Helen E Dahlke, and Mohamed Galal Eltarabily. The use of hydrus-2d to simulate intermittent agricultural managed aquifer recharge (ag-mar) in alfalfa in the san joaquin valley. *Agricultural Water Management*, 282:108296, 2023.

[130] FJ Cook and JH Knight. Oxygen transport to plant roots: modeling for physical understanding of soil aeration. *Soil Science Society of America Journal*, 67(1):20–31, 2003.

[131] NRCS USDA. United states department of agriculture. *Natural Resources Conservation Service. Plants Database. http://plants. usda. gov (accessed in 2000)*, 1999.

[132] AT O'Geen, Matthew BB Saal, Helen E Dahlke, David A Doll, Rachel B Elkins, Allan Fulton, Graham E Fogg, Thomas Harter, Jan W Hopmans, Chuck Ingels, et al. Soil suitability index identifies potential areas for groundwater banking on agricultural lands. *California Agriculture*, 69(2), 2015.

[133] Houssne Bouimouass, Sarah Tweed, Vincent Marc, Younes Fakir, Hamza Sahraoui, and Marc Leblanc. The importance of mountain-block recharge in semiarid basins: An insight from the high-atlas, morocco. *Journal of Hydrology*, page 130818, 2024.

[134] Kath Standen, Luís Costa, Rui Hugman, and José Paulo Monteiro. Integration of managed aquifer recharge into the water supply system in the algarve region, portugal. *Water*, 15(12):2286, 2023.

[135] George Kourakos, Helen E Dahlke, and Thomas Harter. Increasing groundwater availability and seasonal base flow through agricultural managed aquifer recharge in an irrigated basin. *Water Resources Research*, 55(9):7464–7492, 2019.

[136] Nisha Marwaha, George Kourakos, Elad Levintal, and Helen E Dahlke. Identifying agricultural managed aquifer recharge locations to benefit drinking water supply in rural communities. *Water Resources Research*, 57(3):e2020WR028811, 2021.

[137] Yonatan Ganot and Helen E Dahlke. A model for estimating ag-mar flooding duration based on crop tolerance, root depth, and soil texture data. *Agricultural Water Management*, 255:107031, 2021.

[138] George EP Box and Gwilym M Jenkins. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109, 1968.

[139] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[140] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[141] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[142] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. 2022.

[143] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 459469, 2023.

[144] Titus O Awokuse and Xiaohong Wang. Threshold effects and asymmetric price adjustments in us dairy markets. *Canadian Journal of Agricultural Economics/Revue canadienne d'agroeconomie*, 57(2):269–286, 2009.

[145] Anil Seth. Granger causality. *Scholarpedia*, 2(7):1667, 2007.

[146] Biwei Huang, Kun Zhang, Mingming Gong, and Clark Glymour. Causal discovery and forecasting in nonstationary environments with state-space models. In *International conference on machine learning*, pages 2901–2910. PMLR, 2019.

[147] Abdellah Rahmani and Pascal Frossard. Castor: Causal temporal regime structure learning. *arXiv preprint arXiv:2311.01412*, 2023.

[148] Dongjie Wang, Zhengzhang Chen, Jingchao Ni, Liang Tong, Zheng Wang, Yanjie Fu, and Haifeng Chen. Interdependent causal networks for root cause localization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5051–5060, 2023.

[149] Jiecheng Lu, Xu Han, Yan Sun, and Shihao Yang. Cats: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. In *Forty-first International Conference on Machine Learning*, 2024.

[150] Jiecheng Lu, Yan Sun, and Shihao Yang. In-context time series predictor. *arXiv preprint arXiv:2405.14982*, 2024.

[151] Houxing Ren, Jingyuan Wang, Wayne Xin Zhao, and Ning Wu. Rapt: Pre-training of time-aware transformer for learning robust healthcare representation. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 3503–3511, 2021.

[152] Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate forecasting with physics-informed neural odes. In *The Twelfth International Conference on Learning Representations*, 2023.

[153] Yan Sun and Shihao Yang. Manifold-constrained gaussian process inference for time-varying parameters in dynamic systems. *Statistics and Computing*, 33(6):142, 2023.

[154] Yan Sun, Yeping Wang, Zhaohui Li, and Shihao Yang. O-magic: Online change-point detection for dynamic systems. *arXiv preprint arXiv:2411.12277*, 2024.

[155] Qiang Li, Mingkun Tan, Xun Zhao, Dan Zhang, Daoan Zhang, Shengzhao Lei, Anderson S Chu, Lujun Li, and Porawit Kamnoedboon. How llms react to industrial spatio-temporal data? assessing hallucination with a novel traffic incident benchmark dataset. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 36–53, 2025.

[156] Eyke Liegmann, Petros Karamanakos, and Ralph Kennel. Real-time imple-

mentation of long-horizon direct model predictive control on an embedded system. *IEEE Open Journal of Industry Applications*, 3:1–12, 2021.

[157] Zhiyu An, Xianzhong Ding, and Wan Du. Go beyond black-box policies: Rethinking the design of learning agent for interpretable and verifiable hvac control. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 2024.

[158] James R Nelson and Nathan G Johnson. Model predictive control of microgrids for real-time ancillary service market participation. *Applied Energy*, 269:114963, 2020.

[159] Xianzhong Ding and Wan Du. Optimizing irrigation efficiency using deep reinforcement learning in the field. *ACM Transactions on Sensor Networks*, 2023.

[160] Xianzhong Ding, Alberto Cerpa, and Wan Du. Exploring deep reinforcement learning for holistic smart building control. *ACM Transactions on Sensor Networks*, 2023.

[161] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, and Jianhua Zou. DeepAPP: A Deep Reinforcement Learning Framework for Mobile Application Usage Prediction. In *ACM SenSys*, 2019.

[162] Xianzhong Ding, Alberto Cerpa, and Wan Du. Multi-zone hvac control with model-based deep reinforcement learning. *arXiv preprint arXiv:2302.00725*, 2023.

[163] Abhijeet Phatak, Sharath Raghvendra, Chittaranjan Tripathy, and Kaiyi Zhang. Computing all optimal partial transports. In *International Conference on Learning Representations*, 2023.

[164] Sharath Raghvendra, Pouyan Shirzadian, and Kaiyi Zhang. A new robust partial p-wasserstein-based metric for comparing distributions. *arXiv preprint arXiv:2405.03664*, 2024.

[165] Nathaniel Lahn, Sharath Raghvendra, and Kaiyi Zhang. A combinatorial algorithm for approximating the optimal transport in the parallel and mpc settings. *Advances in Neural Information Processing Systems*, 36, 2024.

[166] Zhiyu An, Xianzhong Ding, and Wan Du. Reward bound for behavioral guarantee of model-based planning agents. *arXiv preprint arXiv:2402.13419*, 2024.

[167] Zhiyu An, Zhibo Hou, and Wan Du. Disentangling uncertainties by learning compressed data representation. *arXiv preprint arXiv:2503.15801*, 2025.

[168] Kevin Gmelin, Shikhar Bahl, Russell Mendonca, and Deepak Pathak. Efficient RL via disentangled environment and agent representations. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[169] Guangchen Lan, Han Wang, James Anderson, Christopher Brinton, and Vaneet Aggarwal. Improved communication efficiency in federated natural policy gradient via admm-based gradient updates. *Advances in Neural Information Processing Systems*, 36, 2024.

[170] Guangchen Lan, Dong-Jun Han, Abolfazl Hashemi, Vaneet Aggarwal, and Christopher G Brinton. Asynchronous federated reinforcement learning with policy gradient updates: Algorithm design and convergence analysis. *arXiv preprint arXiv:2404.08003*, 2024.

[171] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[172] Xianjin Xia, Qianwu Chen, Ningning Hou, Yuanqing Zheng, and Mo Li. Xcopy: Boosting weak links for reliable lora communication. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[173] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[174] Norbert Wiener. The theory of prediction. *Modern mathematics for engineers*, 1956.

[175] Sumanth Varambally, Yi-An Ma, and Rose Yu. Discovering mixtures of structural causal models from time series data. *arXiv preprint arXiv:2310.06312*, 2023.

[176] Shanyun Gao, Raghavendra Addanki, Tong Yu, Ryan Rossi, and Murat Kocaoglu. Causal discovery in semi-stationary time series. *Advances in Neural Information Processing Systems*, 36:46624–46657, 2023.

[177] Wanqi Zhou, Shuanghao Bai, Shujian Yu, Qibin Zhao, and Badong Chen. Jacobian regularizer-based neural granger causality. *arXiv preprint arXiv:2405.08779*, 2024.

[178] He Zhao and Edwin V Bonilla. Bayesian factorised granger-causal graphs for multivariate time-series data. *arXiv preprint arXiv:2402.03614*, 2024.

[179] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. 2024.

[180] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. *ICLR*, 2025.

[181] Xiangyu Sun, Oliver Schulte, Guiliang Liu, and Pascal Poupart. Nts-notears: Learning nonparametric dbns with prior knowledge. In *International Conference on Artificial Intelligence and Statistics*, pages 1942–1964. PMLR, 2023.

[182] Ričards Marcinkevičs and Julia E Vogt. Interpretable models for granger causality using self-explaining neural networks. In *International Conference on Learning Representations*, 2021.

[183] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605. Pmlr, 2020.

[184] Xunbi A Ji, Tamás G Molnár, Sergei S Avedisov, and Gábor Orosz. Learning the dynamics of time delay systems with trainable delays. In *Learning for Dynamics and Control*, pages 930–942. PMLR, 2021.

[185] Allen Tran, Aurélien Bibaut, and Nathan Kallus. Inferring the long-term causal effects of long-term treatments from short-term experiments. *arXiv preprint arXiv:2311.08527*, 2023.

[186] Yuxiao Cheng, Lianglong Li, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts+: High-dimensional causal discovery from irregular time-series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11525–11533, 2024.

[187] Defu Cao, James Enouen, Yujing Wang, Xiangchen Song, Chuizheng Meng, Hao Niu, and Yan Liu. Estimating treatment effects from irregular time series observations with hidden confounders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[188] Xiangchen Song, Weiran Yao, Yewen Fan, Xinshuai Dong, Guangyi Chen, Juan Carlos Niebles, Eric Xing, and Kun Zhang. Temporally disentangled representation learning under unknown nonstationarity. *Advances in Neural Information Processing Systems*, 2024.

[189] Dongxia Wu, Tsuyoshi Idé, Georgios Kollias, Jiri Navratil, Aurelie Lozano, Naoki Abe, Yian Ma, and Rose Yu. Learning granger causality from instance-wise self-attentive hawkes processes. In *International Conference on Artificial Intelligence and Statistics*, 2024.

[190] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Yunzhong Qiu, Haoran Zhang, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *arXiv preprint arXiv:2402.19072*, 2024.

[191] Ziyou Guo, Yan Sun, and Tieru Wu. Weits: A wavelet-enhanced residual framework for interpretable time series forecasting. *arXiv preprint arXiv:2405.10877*, 2024.

[192] Shiliang Sun et al. Caudits: Causal disentangled domain adaptation of multivariate time series. In *Forty-first International Conference on Machine Learning*, 2024.

[193] Yuning Chen, Kang Yang, Zhiyu An, Brady Holder, Luke Paloutzian, Khaled Bali, and Wan Du. Marlp: Time-series forecasting control for agricultural managed aquifer recharge. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'24)*, 2024.

[194] Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

[195] Luca Gambetti et al. Structural vector autoregressive models. In *Oxford Research Encyclopedia of Economics and Finance*, pages 1–73. Oxford University Press, 2020.

[196] Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances*, 5(11):eaau4996, 2019.

[197] Wenbo Gong, Joel Jennings, Cheng Zhang, and Nick Pawlowski. Rhino: Deep causal temporal relationship learning with history-dependent noise. In *The Eleventh International Conference on Learning Representations*, 2023.

[198] Yuxiao Cheng, Runzhao Yang, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts: Neural causal discovery from irregular time-series data. *arXiv preprint arXiv:2302.07458*, 2023.

[199] Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2021.

[200] Alexis Bellot, Kim Branson, and Mihaela van der Schaar. Neural graphical modelling in continuous-time: consistency guarantees and algorithms. In *International Conference on Learning Representations*, 2022.

[201] Edward De Brouwer, Adam Arany, Jaak Simm, and Yves Moreau. Latent convergent cross mapping. In *International Conference on Learning Representations*, 2020.

[202] Saurabh Khanna and Vincent YF Tan. Economy statistical recurrent units for inferring nonlinear granger causality. In *International Conference on Learning Representations*, 2019.

[203] Chenxiao Xu, Hao Huang, and Shinjae Yoo. Scalable causal graph learning through a deep neural network. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1853–1862, 2019.

[204] Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019.

[205] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

[206] UCI. ElectricityLoadDiagrams20112014. `https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014`, 2015.

[207] Wetterstation. Weather Data. `https://www.bgc-jena.mpg.de/wetter/`, 2024.

[208] Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. Calibration of time-series forecasting: Detecting and adapting context-driven distribution shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

[209] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, 2022.

[210] Fengchen Gong, Divya Raghunathan, Aarti Gupta, and Maria Apostolaki. Zoom2net: Constrained network telemetry imputation. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 764–777, 2024.

[211] Zoom. Bringing Zoom's end-to-end optimizations to WebRTC. `https://blog.livekit.io/livekit-one-dot-zero/`.

[212] Aaron Ullal. What powers Google Meet and Microsoft Teams? WebRTC Demystified. `https://levelup.gitconnected.com/what-powers-google-meet-and-microsoft-teams-webrtc-demystified-step-by-step-tutorial-e0cb422010f7`.

[213] Francis Y Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, and Keith Winstein. Learning in situ: a randomized experiment in video streaming. In *USENIX NSDI 20*, 2020.

[214] WebRTC. WebRTC Cloud Gaming: Unboxing Stadia. . `https://webrtc.ventures/2021/02/webrtc-cloud-gaming-unboxing-stadia/`.

[215] Jiangkai Wu, Yu Guan, Qi Mao, Yong Cui, Zongming Guo, and Xinggong Zhang. ZGaming: Zero-latency 3D cloud gaming by image prediction. In *ACM SIGCOMM 2023*, 2023.

[216] Meta. Livestream from Meta Quest to YouTube using OBS. `https://www.meta.com/help/quest/articles/in-vr-experiences/social-features-and-sharing/livestream-to-youtube-obs/`.

[217] Xingyu Liu, Vladimir Kirilyuk, Xiuxiu Yuan, Peggy Chi, Xiang Chen, Alex Olwal, and Ruofei Du. Visual Captions: Augmenting Verbal Communication with On-the-Fly Visuals. In *ACM CHI*, 2023.

[218] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF CVPR*, 2022.

[219] Vibhaalakshmi Sivaraman, Pantea Karimi, Vedantha Venkatapathy, Mehrdad Khani, Sadjad Fouladi, Mohammad Alizadeh, Frédo Durand, and Vivienne Sze. Gemino: Practical and Robust Neural Compression for Video Conferencing. *arXiv preprint arXiv:2209.10507*, 2022.

[220] Sheng-Yang Chiu, Yu-Ting Huang, Chieh-Ting Lin, Yu-Chee Tseng, Jen-Jee Chen, Meng-Hsuan Tu, Bo-Chen Tung, and YuJou Nieh. Privacy-preserving video conferencing via thermal-generative images. In *IEEE ICRA*. IEEE, 2023.

[221] Devar uses generative AI to create 3D AR images from text prompts. `https://venturebeat.com/games/devar-uses-generative-ai-to-create-3d-ar-images-from-text-prompts`.

[222] OpenAI. DALL-E 3. `https://openai.com/index/dall-e-3/`.

[223] Stability AI Runway, CompVis. Stability AI. `https://stability.ai/stable-image`.

[224] Michael Rudow, Francis Y Yan, Abhishek Kumar, Ganesh Ananthanarayanan, Martin Ellis, and KV Rashmi. Tambur: Efficient loss recovery for videoconferencing via streaming codes. In *USENIX NSDI*, 2023.

[225] Yihua Cheng, Ziyi Zhang, Hanchen Li, Anton Arapin, Yue Zhang, Qizheng Zhang, Yuhan Liu, Kuntai Du, Xu Zhang, Francis Y Yan, et al. GRACE:Loss-Resilient Real-Time Video through Neural Codecs. In *USENIX NSDI*, 2024.

[226] Zili Meng, Tingfeng Wang, Yixin Shen, Bo Wang, Mingwei Xu, Rui Han, Honghao Liu, Venkat Arun, Hongxin Hu, and Xue Wei. Enabling high quality Real-Time communications with adaptive Frame-Rate. In *USENIX NSDI*, 2023.

[227] Zili Meng, Xiao Kong, Jing Chen, Bo Wang, Mingwei Xu, Rui Han, Honghao Liu, Venkat Arun, Hongxin Hu, and Xue Wei. Hairpin: Rethinking packet loss recovery in edge-based interactive video streaming. In *USENIX NSDI*, 2024.

[228] OpenAI. DALL-E 2. `https://openai.com/index/dall-e-2/`.

[229] Sandra Götz. Fluency in native and nonnative English speech. 2013.

[230] Douglas W Maynard. Placement of topic changes in conversation. *Semiotica*, 30(3-4):263–290, 1980.

[231] YouTube. Use automatic captioning. `https://support.google.com/youtube/answer/6373554`.

[232] Talha Waheed, Ihsan Ayyub Qazi, Zahaib Akhtar, and Zafar Ayyub Qazi. Coal not diamonds: how memory pressure falters mobile video QOE. In *ACM CONEXT*, 2022.

[233] Alan Jones, Peter Sevcik, and Rebecca Wetzel. Internet connection requirements for effective video conferencing to support work from home and eLearning. *NetForecast Report*, 2021.

[234] Zili Meng, Yaning Guo, Chen Sun, Bo Wang, Justine Sherry, Hongqiang Harry Liu, and Mingwei Xu. Achieving consistent low latency for wireless real-time communications with the shortest control loop. In *ACM SIGCOMM*, 2022.

[235] Gaetano Carlucci, Luca De Cicco, Stefan Holmer, and Saverio Mascolo. Analysis and design of the google congestion control for web real-time communication (webrtc). In *Proceedings of the 7th International Conference on Multimedia Systems*, pages 1–12, 2016.

[236] Jiayi Chen, Nihal Sharma, Tarannum Khan, Shu Liu, Brian Chang, Aditya Akella, Sanjay Shakkottai, and Ramesh K Sitaraman. Darwin: Flexible learning-based cdn caching. In *ACM SIGCOMM*, 2023.

[237] Gang Yan, Jian Li, and Don Towsley. Learning from optimal caching for content delivery. In *CoNEXT*, 2021.

[238] Gang Yan and Jian Li. Rl-bélády: A unified learning framework for content caching. In *ACM Multimedia*, 2020.

[239] Gang Yan and Jian Li. Towards latency awareness for content delivery network caching. In *USENIX ATC*, 2022.

[240] Bingyang Wu, Kun Qian, Bo Li, Yunfei Ma, Qi Zhang, Zhigang Jiang, Jiayu Zhao, Dennis Cai, Ennan Zhai, Xuanzhe Liu, et al. Xron: A hybrid elastic cloud overlay network for video conferencing at planetary scale. In *ACM SIGCOMM*, 2023.

[241] Chan C Lee and Der-Tsai Lee. A simple on-line bin-packing algorithm. *Journal of the ACM (JACM)*, 32(3):562–572, 1985.

[242] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. MirrorGAN: Learning Text-To-Image Generation by Redescription. In *IEEE/CVF CVPR*, June 2019.

[243] Jisu Nam, Heesu Kim, DongJae Lee, Siyoon Jin, Seungryong Kim, and Seunggyu Chang. Dreammatcher: Appearance matching self-attention for semantically-consistent text-to-image personalization. In *IEEE/CVF CVPR*, 2024.

[244] Xinyu Li, Chuang Zhao, Hongke Zhao, Likang Wu, and Ming HE. GAN-Prompt: Enhancing Robustness in LLM-Based Recommendations with GAN-Enhanced Diversity Prompts. *arXiv preprint arXiv:2408.09671*, 2024.

[245] Meta. Meet Llama 3.1. `https://www.llama.com/`.

[246] Xu Zhang, Yiyang Ou, Siddhartha Sen, and Junchen Jiang. {SENSEI}: Aligning video streaming quality with dynamic user sensitivity. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 303–320, 2021.

[247] Shuyuan Xu, Jun Wang, Wenchi Shou, Tuan Ngo, Abdul-Manan Sadick, and Xiangyu Wang. Computer vision techniques in construction: a critical review. *Archives of Computational Methods in Engineering*, 28:3383–3397, 2021.

[248] Ilya Nikolaevskiy. Refactor FrameBuffer to allow for new decode scheduling strategy. `https://issues.webrtc.org/issues/42223541`.

[249] kubernets. Production-Grade Container Orchestration. `https://kubernetes.io/`.

[250] aiortc. aiortc. `https://github.com/aiortc/aiortc`.

[251] Eduardo Cuervo, Alec Wolman, Landon P Cox, Kiron Lebeck, Ali Razeen, Stefan Saroiu, and Madanlal Musuvathi. Kahawai: High-quality mobile gaming using gpu offload. In *Proceedings of the 13th annual international conference on mobile systems, applications, and services*, pages 121–135, 2015.

[252] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[253] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, 2025.

[254] Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Enze Xie, and Song Han. Sana-sprint: One-step diffusion with continuous-time consistency distillation. *arXiv preprint arXiv:2503.09641*, 2025.

[255] Anirudh Sivaraman, Suvinay Subramanian, Mohammad Alizadeh, Sharad Chole, Shang-Tse Chuang, Anurag Agrawal, Hari Balakrishnan, Tom Edsall, Sachin Katti, and Nick McKeown. Programmable packet scheduling at line rate. In *SIGCOMM*, 2016.

[256] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[257] Federal Communication Commission. Measuring Broadband America - July 2012. `https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-july-2012`.

[258] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *ACM NAACL*, 2009.

[259] NVIDIA. Inference Performance of NVIDIA Data Center Products. `https://developer.nvidia.com/deep-learning-performance-training-inference/ai-inference`.

[260] Lambda. Introducing NVIDIA RTX A6000 GPU Instances on Lambda Cloud. `https://lambdalabs.com`.

[261] GpuMart. Dedicated A100 GPU Hosting, NVIDIA A100 Rental. `https://www.gpu-mart.com/nvidia-a100-rental#plans`.

[262] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. Mahimahi: accurate {Record-and-Replay} for {HTTP}. In *2015 USENIX Annual Technical Conference (USENIX ATC 15)*, pages 417–429, 2015.

[263] Gustav Bøg Petersen, Valdemar Stenberdt, Richard E Mayer, and Guido Makransky. Collaborative generative learning activities in immersive virtual reality increase learning. *Computers & Education*, 207:104931, 2023.

[264] Mallesham Dasari, Edward Lu, Michael W Farb, Nuno Pereira, Ivan Liang, and Anthony Rowe. Scaling VR Video Conferencing. In *IEEE VR*. IEEE, 2023.

[265] Meta. Meta Quest 2. `https://www.meta.com/quest/products/quest-2/`.

[266] Unity. Managing latency with prediction. `https://docs.unity3d.com/Packages/com.unity.netcode@1.3/manual/prediction-n4e.html`.

[267] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[268] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models. In *IEEE/CVF CVPR*, 2024.

[269] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *IEEE/CVF CVPRr*, 2023.

[270] Sikai Yang, Kang Yang, Yuning Chen, Fan Zhao, and Wan Du. See behind walls in real-time using aerial drones and augmented reality. *arXiv preprint arXiv:2410.13139*, 2024.

[271] Kyungjin Lee, Juheon Yi, and Youngki Lee. FarfetchFusion: Towards Fully Mobile Live 3D Telepresence Platform. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023.

[272] Shuowei Jin, Ruiyang Zhu, Ahmad Hassan, Xiao Zhu, Xumiao Zhang, Z Morley Mao, Feng Qian, and Zhi-Li Zhang. OASIS: Collaborative Neural-Enhanced Mobile Video Streaming. In *ACM MMSYS*, 2024.

[273] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[274] Ozgur Kara, Bariscan Kurtkaya, Hidir Yesiltepe, James M. Rehg, and Pinar Yanardag. RAVE: Randomized Noise Shuffling for Fast and Consistent Video Editing with Diffusion Models. In *IEEE/CVF CVPR*, 2024.

[275] Jian Liu, Xiaoshui Huang, Tianyu Huang, Lu Chen, Yuenan Hou, Shixiang Tang, Ziwei Liu, Wanli Ouyang, Wangmeng Zuo, Junjun Jiang, et al. A Comprehensive Survey on 3D Content Generation. *arXiv preprint arXiv:2402.01166*, 2024.

[276] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *IEEE/CVF CVPR*, 2022.

[277] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. In *ICLR*.

[278] Xuechen Zhang, Zheng Li, Samet Oymak, and Jiasi Chen. Text-to-3D Generative AI on Mobile Devices: Measurements and Optimizations. In *ACM SIGCOMM*, 2023.

[279] Kangfu Mei, Mauricio Delbracio, Hossein Talebi, Zhengzhong Tu, Vishal M Patel, and Peyman Milanfar. CoDi: Conditional Diffusion Distillation for Higher-Fidelity and Faster Image Generation. In *IEEE/CVF CVPR*, 2024.

[280] Lirui Zhao, Yue Yang, Kaipeng Zhang, Wenqi Shao, Yuxin Zhang, Yu Qiao, Ping Luo, and Rongrong Ji. DiffAgent: Fast and Accurate Text-to-Image API Selection with Large Language Model. In *IEEE/CVF CVPR*, 2024.

[281] Juni Kim, Zhikang Dong, and Pawel Polak. Face-gps: A comprehensive technique for quantifying facial muscle dynamics in videos. In *Medical Imaging Meets NeurIPS: An official NeurIPS Workshop*, 2023.

[282] Zhikang Dong, Juni Kim, and Paweł Polak. Mapping the invisible: Face-gps for facial muscle dynamics in videos. In *2024 IEEE First International Conference on Artificial Intelligence for Medicine, Health and Care (AIMHC)*, pages 209–213. IEEE, 2024.

[283] Zhikang Dong, Weituo Hao, Ju-Chiang Wang, Peng Zhang, and Pawel Polak. Every image listens, every image dances: Music-driven image animation. *arXiv preprint arXiv:2501.18801*, 2025.

[284] Zhikang Dong, Xiulong Liu, Bin Chen, Pawel Polak, and Peng Zhang. Musechat: A conversational music recommendation system for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12775–12785, 2024.

[285] Zibo Wang, Pinghe Li, Chieh-Jan Mike Liang, Feng Wu, and Francis Y Yan. Autothrottle: A Practical {Bi-Level} Approach to Resource Management for {SLO-Targeted} Microservices. In *USENIX NSDI 24*, 2024.

[286] Deepak Narayanan, Keshav Santhanam, Fiodar Kazhamiaka, Amar Phanishayee, and Matei Zaharia. Heterogeneity-Aware cluster scheduling policies for deep learning workloads. In *USENIX OSDI*, 2020.

[287] Tzu-Wei Yang, Seth Pollen, Mustafa Uysal, Arif Merchant, and Homer Wolfmeister. CacheSack: Admission Optimization for Google Datacenter Flash Caches. In *USENIX ATC*, 2022.

[288] Sudipta Saha Shubha, Haiying Shen, and Anand Iyer. USHER: Holistic Interference Avoidance for Resource Optimized ML Inference. In *USENIX OSDI*, 2024.

[289] Bohan Zhao, Wei Xu, Shuo Liu, Yang Tian, Qiaoling Wang, and Wenfei Wu. Training Job Placement in Clusters with Statistical In-Network Aggregation. In *ACM ASPLOS*, 2024.

[290] Chunyu Qiao, Gen Li, Qiang Ma, Jiliang Wang, and Yunhao Liu. Trace-driven optimization on bitrate adaptation for mobile video streaming. *IEEE Transactions on Mobile Computing*, 21(6):2243–2256, 2020.

[291] Chunyu Qiao, Gen Li, Jiliang Wang, and Yunhao Liu. Neiva: environment identification based video bitrate adaption in cellular networks. In *Proceedings of the International Symposium on Quality of Service*, pages 1–10, 2019.

[292] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. YuZu: Neural-enhanced Volumetric Video Streaming. In *USENIX NSDI*, 2022.

[293] Yu Liu, Puqi Zhou, Zejun Zhang, Anlan Zhang, Bo Han, Zhenhua Li, and Feng Qian. MuV2: Scaling up Multi-user Mobile Volumetric Video Streaming via Content Hybridization and Sharing. In *ACM MOBICOM*, 2024.

[294] Sikai Yang and Wan Du. Tri-cam: Practical eye gaze tracking via camera network. *arXiv preprint arXiv:2409.19554*, 2024.

[295] Zhiyu An, Xianzhong Ding, Yen-Chun Fu, Cheng-Chung Chu, Yan Li, and Wan Du. Golden-retriever: High-fidelity agentic retrieval augmented generation for industrial knowledge base. *arXiv preprint arXiv:2408.00798*, 2024.

[296] Zuhui Wang, Yunting Yin, and I.V. Ramakrishnan. Enhancing image-text matching with adaptive feature aggregation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[297] Bach Pham, JuiHsuan Wong, Samuel Kim, Yunting Yin, and Steven Skiena. Word definitions from large language models, 2025.

[298] Zhikang Dong, Apoorva Beedu, Jason Sheinkopf, and Irfan Essa. Mamba fusion: Learning actions through questioning. *arXiv preprint arXiv:2409.11513*, 2024.

[299] Jingao Xu, Guoxuan Chi, Zheng Yang, Danyang Li, Qian Zhang, Qiang Ma, and Xin Miao. Followupar: Enabling follow-up effects in mobile ar applications. In *ACM MOBISYS*, 2021.

[300] Miaomiao Liu, Sikai Yang, Wyssanie Chomsin, and Wan Du. Real-time tracking of smartwatch orientation and location by multitask learning. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 120–133, 2022.

[301] Sikai Yang, Miaomiao Liu, and Wan Du. Magnetic distortion resistant orientation estimation. *arXiv preprint arXiv:2410.12304*, 2024.

[302] Miaomiao Liu, Sikai Yang, Arya Rathee, and Wan Du. Orientation estimation piloted by deep reinforcement learning. In *2024 IEEE/ACM Ninth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 134–145. IEEE, 2024.

[303] Xiulong Liu, Zhikang Dong, and Peng Zhang. Tackling data bias in music-avqa: Crafting a balanced dataset for unbiased question-answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4478–4487, 2024.

[304] Yunting Yin, Douglas William Hanes, Steven Skiena, and Sean A P Clouston. Quantifying healthy aging in older veterans using computational audio analysis. *The Journals of Gerontology: Series A*, 79(1):glad154, 06 2023.

[305] Yunting Yin and Steven Skiena. Inferring age from linguistic and verbal cues in celebrity interviews. In *Proceedings of the 2023 International Conference on Frontiers of Artificial Intelligence and Machine Learning*, FAIML '23, page 17, New York, NY, USA, 2024. Association for Computing Machinery.

[306] Charuta Pethe, Bach Pham, Felix D Childress, Yunting Yin, and Steven Skiena. Prosody analysis of audiobooks, 2025.

[307] Yunting Yin. *A Study of Aging Through Speech and Language Analysis*. PhD thesis, State University of New York at Stony Brook, 2024.

[308] Le Chen, Nesreen K Ahmed, Akash Dutta, Arijit Bhattacharjee, Sixing Yu, Quazi Ishtiaque Mahmud, Waqwoya Abebe, Hung Phan, Aishwarya Sarkar, Branden Butler, et al. The landscape and challenges of hpc research and llms. *arXiv preprint arXiv:2402.02018*, 2024.

[309] Xianzhong Ding, Le Chen, Murali Emani, Chunhua Liao, Pei-Hung Lin, Tristan Vanderbruggen, Zhen Xie, Alberto Cerpa, and Wan Du. Hpc-gpt: Integrating large language model for high-performance computing. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 951–960, 2023.

[310] Le Chen, Pei-Hung Lin, Tristan Vanderbruggen, Chunhua Liao, Murali Emani, and Bronis De Supinski. Lm4hpc: Towards effective language model application in high-performance computing. In *International Workshop on OpenMP*, pages 18–33. Springer, 2023.

[311] Le Chen, Arijit Bhattacharjee, Nesreen Ahmed, Niranjan Hasabnis, Gal Oren, Vy Vo, and Ali Jannesari. Ompgpt: A generative pre-trained transformer model for openmp. In *European Conference on Parallel Processing*, pages 121–134. Springer, 2024.

[312] Zihan Gao and Jiepu Jiang. Evaluating human-ai hybrid conversational systems with chatbot message suggestions. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 534–544, 2021.

[313] Yifei Xu, Tusher Chakraborty, Emre Kıcıman, Bibek Aryal, Eduardo Rodrigues, Srinagesh Sharma, Roberto Estevao, Maria Angels de Luis Balaguer, Jessica Wolk, Rafael Padilha, et al. Rlthf: Targeted human feedback for llm alignment. *arXiv preprint arXiv:2502.13417*, 2025.

[314] Yifei Xu, Yuning Chen, Xumiao Zhang, Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu, Wan Du, Zhuoqing Mao, Ennan Zhai, et al. Cloudeval-yaml:

A practical benchmark for cloud configuration generation. *Proceedings of Machine Learning and Systems*, 6:173–195, 2024.

[315] Yifei Xu, Yuning Chen, Xumiao Zhang, Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu, Wan Du, Z Morley Mao, Ennan Zhai, et al. Cloudeval-yaml: A realistic and scalable benchmark for cloud configuration generation. 2023.

[316] Le Chen, Bin Lei, Dunzhi Zhou, Pei-Hung Lin, Chunhua Liao, Caiwen Ding, and Ali Jannesari. Fortran2cpp: Automating fortran-to-c++ migration using llms via multi-turn dialogue and dual-agent integration. *arXiv preprint arXiv:2412.19770*, 2024.

[317] Le Chen, Xianzhong Ding, Murali Emani, Tristan Vanderbruggen, Pei-Hung Lin, and Chunhua Liao. Data race detection using large language models. In *Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 215–223, 2023.

[318] Le Chen, Quazi Ishtiaque Mahmud, Hung Phan, Nesreen Ahmed, and Ali Jannesari. Learning to parallelize with openmp by augmented heterogeneous ast representation. *Proceedings of Machine Learning and Systems*, 5:442–456, 2023.

[319] Le Chen, Quazi Ishtiaque Mahmud, and Ali Jannesari. Multi-view learning for parallelism discovery of sequential programs. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2022.

[320] Gang Yan, Hao Wang, Xu Yuan, and Jian Li. Fedrola: Robust federated learning against model poisoning via layer-based aggregation. In *ACM SIGKDD*, 2024.

[321] Gang Yan, Hao Wang, Xu Yuan, and Jian Li. Criticalfl: A critical learning periods augmented client selection framework for efficient federated learning. In *ACM SIGKDD*, 2023.

[322] Leming Shen, Qiang Yang, Kaiyan Cui, Yuanqing Zheng, Xiao-Yong Wei, Jianwei Liu, and Jinsong Han. Fedconv: A learning-on-model paradigm for heterogeneous federated clients. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 2024.

[323] Maolin Gan, Lanpeng Li, Samiul Alam, Li Liu, Luyang Liu, Mi Zhang, and Zhichao Cao. Geofl: A framework for efficient geo-distributed cross-device federated learning. In *Proceedings of IEEE INFOCOM*, 2025.

[324] Yueyuan Sui, Yiting Zhang, Yanchen Liu, Minghui Zhao, Kaiyuan Hou, Jingping Nie, Xiaofan Jiang, and Stephen Xia. Domain: Towards programless smart homes. In *Proceedings of the 3rd International Workshop on Human-Centered Sensing, Modeling, and Intelligent Systems*, pages 7–12, 2025.

[325] Leming Shen, Qiang Yang, Yuanqing Zheng, and Mo Li. Autoiot: Llm-driven automated natural language programming for aiot applications. *arXiv preprint arXiv:2503.05346*, 2025.

[326] Leming Shen, Qiang Yang, Xinyu Huang, Zijing Ma, and Yuanqing Zheng. Gpiot: Tailoring small language models for iot program synthesis and development. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, 2025.

[327] Jizhou Li, Zikun Li, Yifei Xu, Shiqi Jiang, Tong Yang, Bin Cui, Yafei Dai, and Gong Zhang. Wavingsketch: An unbiased and generic sketch for finding top-k items in data streams. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1574–1584, 2020.

[328] Zhuochen Fan, Zhoujing Hu, Yuhan Wu, Jiarui Guo, Wenrui Liu, Tong Yang, Hengrui Wang, Yifei Xu, Steve Uhlig, and Yaofeng Tu. Pisketch: finding persistent and infrequent flows. In *Proceedings of the ACM SIGCOMM Workshop on Formal Foundations and Security of Programmable Network Infrastructures*, pages 8–14, 2022.

[329] Yuhan Wu, Shiqi Jiang, Yifei Xu, Siyuan Dong, Kaicheng Yang, Peiqing Chen, and Tong Yang. Unbiased real-time traffic sketching. *IEEE Transactions on Network Science and Engineering*, 11(3):2371–2383, 2023.

[330] Zhaowei Tan, Jinghao Zhao, Yuanjie Li, Yifei Xu, and Songwu Lu. {Device-Based}{LTE} latency reduction at the application layer. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 471–486, 2021.

[331] Jinghao Zhao, Zhaowei Tan, Yifei Xu, Zhehui Zhang, and Songwu Lu. Seed: a sim-based solution to 5g failures. In *Proceedings of the ACM SIGCOMM 2022 Conference*, pages 129–142, 2022.

[332] Zhaowei Tan, Jinghao Zhao, Yuanjie Li, Yifei Xu, Yunqi Guo, and Songwu Lu. Ldrp: Device-centric latency diagnostic and reduction for cellular networks without root. *IEEE Transactions on Mobile Computing*, 23(4):2748–2764, 2023.

[333] Xianzhong Ding, Yunkai Zhang, Binbin Chen, Donghao Ying, Tieying Zhang, Jianjun Chen, Lei Zhang, Alberto Cerpa, and Wan Du. Towards vm rescheduling optimization through deep reinforcement learning. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025.

[334] Xianzhong Ding, Wan Du, and Alberto Cerpa. Octopus: Deep reinforcement learning for holistic smart building control. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*, pages 326–335, 2019.

[335] Xianzhong Ding, Wan Du, and Alberto E Cerpa. Mb2c: Model-based deep reinforcement learning for multi-zone building control. In *Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation*, pages 50–59, 2020.

[336] Xianzhong Ding, Zhiyu An, Arya Rathee, and Wan Du. A safe and data-efficient model-based reinforcement learning system for hvac control. *IEEE Internet of Things Journal*, 2025.

[337] Jason Wu, Kang Yang, Lance Kaplan, and Mani Srivastava. Admn: A layer-wise adaptive multimodal network for dynamic input noise and compute resources. *arXiv preprint arXiv:2502.07862*, 2025.

[338] Zhihao Shen, Kang Yang, Xi Zhao, Jianhua Zou, Wan Du, and Junjie Wu. Dmm: A deep reinforcement learning based map matching framework for cellular data. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5120–5137, 2024.

[339] Zhihao Shen, Kang Yang, Xi Zhao, Jianhua Zou, and Wan Du. DeepAPP: A Deep Reinforcement Learning Framework for Mobile Application Usage Prediction. *IEEE Transactions on Mobile Computing*, 22(2):824–840, 2023.

[340] Kang Yang, Xi Zhao, Jianhua Zou, and Wan Du. Atpp: A mobile app prediction system based on deep marked temporal point processes. *ACM Transactions on Sensor Networks*, 19(3):1–24, 2023.

[341] Yuhong Zhong, Haoyu Li, Yu Jian Wu, Ioannis Zarkadas, Jeffrey Tao, Evan Mesterhazy, Michael Makris, Junfeng Yang, Amy Tai, Ryan Stutsman, et al. {XRP}:{In-Kernel} storage functions with {eBPF}. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 375–393, 2022.

[342] Hongyu Zhu, Ruofan Wu, Yijia Diao, Shanbin Ke, Haoyu Li, Chen Zhang, Jilong Xue, Lingxiao Ma, Yuqing Xia, Wei Cui, et al. {ROLLER}: Fast and efficient tensor compilation for deep learning. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 233–248, 2022.

[343] Mingyu Wu, Ziming Zhao, Haoyu Li, Heting Li, Haibo Chen, Binyu Zang, and Haibing Guan. Espresso: Brewing java for more non-volatility with

non-volatile memory. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 70–83, 2018.

[344] Mingyu Wu, Ziming Zhao, Yanfei Yang, Haoyu Li, Haibo Chen, Binyu Zang, Haibing Guan, Sanhong Li, Chuansheng Lu, and Tongbao Zhang. Platinum: A {CPU-Efficient} concurrent garbage collector for {Tail-Reduction} of interactive services. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 159–172, 2020.

[345] Haoyu Li, Mingyu Wu, and Haibo Chen. Analysis and optimizations of java full garbage collection. In *Proceedings of the 9th Asia-Pacific Workshop on Systems*, pages 1–7, 2018.

[346] Haoyu Li, Mingyu Wu, Binyu Zang, and Haibo Chen. Scissorgc: scalable and efficient compaction for java full garbage collection. In *Proceedings of the 15th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pages 108–121, 2019.

[347] Haoyu Li, Sheng Jiang, Chen Chen, Ashwini Raina, Xingyu Zhu, Changxu Luo, and Asaf Cidon. {RubbleDB}:{CPU-Efficient} replication with {NVMe-oF}. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 689–703, 2023.

[348] Haoyu Li, Jingkai Fu, Qing Li, Windsor Hsu, and Asaf Cidon. Enabling the write-back page cache with strong consistency in distributed userspace file systems. *arXiv preprint arXiv:2503.18191*, 2025.

[349] Guoxuan Chi, Zheng Yang, Chenshu Wu, Jingao Xu, Yuchong Gao, Yunhao Liu, and Tony Xiao Han. Rf-diffusion: Radio signal generation via time-frequency diffusion. In *Proceedings of the ACM MobiCom*, 2024.

[350] Kang Yang, Yuning Chen, and Wan Du. OrchLoc: In-Orchard Localization via a Single LoRa Gateway and Generative Diffusion Model-based Fingerprinting. In *ACM MobiSys*, 2024.

[351] Kang Yang, Yuning Chen, and Wan Du. Gwrf: A generalizable wireless radiance field for wireless signal propagation modeling. *arXiv preprint arXiv:2502.05708*, 2025.

[352] Kang Yang, Gaofeng Dong, Sijie Ji, Wan Du, and Mani Srivastava. Scalable 3d gaussian splatting-based rf signal spatial propagation modeling. *arXiv preprint arXiv:2502.01826*, 2025.

[353] Xianzhong Ding, Wanshi Hong, Zhiyu An, Bin Wang, and Wan Du. Deepot: Parking lot identification using low-resolution satellite imagery. 2025.

[354] Fudong Lin, Xu Yuan, Lu Peng, and Nian-Feng Tzeng. Cascade variational auto-encoder for hierarchical disentanglement. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, pages 1248–1257, 2022.

[355] Yi He, Fudong Lin, Xu Yuan, and Nian-Feng Tzeng. Interpretable minority synthesis for imbalanced classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2542–2548, 2021.

[356] Jingping Nie, Yuang Fan, Minghui Zhao, Runxi Wan, Ziyi Xuan, Matthias Preindl, and Xiaofan Jiang. Multi-modal dataset across exertion levels: Capturing post-exercise speech, breathing, and phonocardiogram. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, pages 297–304, 2025.

[357] Mingkun Tan, Daniel Langenkämper, and Tim W Nattkemper. The impact of data augmentations on deep learning-based marine object classification in benthic image transects. *Sensors*, 22(14):5383, 2022.

[358] Mingkun Tan, Daniel Langenkämper, Michael Kloster, and Tim W Nattkemper. Scaling down annotation needs: The capacity of self-supervised learning on diatom classification. *iScience*, 28(4), 2025.

[359] Jingping Nie, Ran Liu, Behrooz Mahasseni, and Vikramjit Mitra. Model-driven heart rate estimation and heart murmur detection based on phonocardiogram. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2024.

[360] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[361] Fudong Lin, Jiadong Lou, Xu Yuan, and Nian-Feng Tzeng. Towards robust vision transformer via masked adaptive ensemble. *arXiv preprint arXiv:2407.15385*, 2024.

[362] Fudong Lin, Summer Crawford, Kaleb Guillot, Yihe Zhang, Yan Chen, Xu Yuan, et al. Mmst-vit: Climate change-aware crop yield prediction via multi-modal spatial-temporal vision transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5751–5761, 2023.

[363] Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang, Shangqian Gao, and Yanfu Zhang. Unlocking memorization in large language models with dynamic soft prompting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9782–9796, 2024.

[364] Lei Lu, Zhepeng Wang, Runxue Bao, Mengbing Wang, Fangyi Li, Yawen Wu, Weiwen Jiang, Jie Xu, Yanzhi Wang, and Shangqian Gao. All-in-one tuning and structural pruning for domain-specific llms. *arXiv preprint arXiv:2412.14426*, 2024.

[365] Zhepeng Wang, Runxue Bao, Yawen Wu, Guodong Liu, Lei Yang, Liang Zhan, Feng Zheng, Weiwen Jiang, and Yanfu Zhang. Self-guided knowledge-injected graph neural network for alzheimers diseases. In *International Con-*

ference on Medical Image Computing and Computer-Assisted Intervention, pages 378–388. Springer, 2024.

[366] Yawen Wu, Zhepeng Wang, Dewen Zeng, Yiyu Shi, and Jingtong Hu. Synthetic data can also teach: Synthesizing effective data for unsupervised visual representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2866–2874, 2023.

[367] Yawen Wu, Zhepeng Wang, Zhenge Jia, Yiyu Shi, and Jingtong Hu. Intermittent inference with nonuniformly compressed multi-exit neural network for energy harvesting powered devices. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.