ABSTRACT

Title of Dissertation: PHOTONIC RESERVOIR COMPUTING
BASED ON OPTOELECTRONIC OSCILLATORS

Haoying Dai
Doctor of Philosophy, 2025

Dissertation Directed by: Professor Yanne Kouomou Chembo
Department of Electrical and Computer Engineering

Optoelectronic Oscillator (OEO) is a device widely used in modern electronic and photonic systems. Its applications include but not limited to chaos communication systems, random number generation, chaotic radar and lidar, ultra-pure microwave generation, and sensing systems. Moreover, it has rich nonlinear dynamics suitable for studying phenomena like period-one (P1) oscillation, period doubling (P2), and chaos.

Reservoir computing (RC) is a machine learning (ML) algorithm that breaks the traditional von Neumann computational paradigm. It is able to learn directly from data and perform both the regression and the classification tasks. Comparing to other machine learning algorithms like recurrent neural network (RNN), it has a much simpler structure that allows it to avoid computationally expensive back-propagation (BP) algorithm while maintains a competitive performance.

Recently, as an intrinsic time-delay system, optoelectronic oscillator has been introduced as a machine learning platform to perform reservoir computing. Its ability to accept both the radio-frequency (RF) signal and the optical signal make it an ideal machine learning platform for optical fiber communication systems and radio-frequency communication systems.

In Ch. 1, we start the thesis with a general review of the history of artificial intelligence (AI) and the recent development of microwave photonics. Then, we discuss the most recent advances that use photonic devices for machine learning (ML) - one of the most important branches in the field of photonic AI research.

In Ch. 2, we introduce different types of OEOs, namely the broadband OEO and narrowband OEO, and how to mathematically model them. Specifically, we will derive the time-delayed equations that govern their behaviors. For the narrowband OEO, we introduce the derivation of an envelop equation which is suitable for the numerical simulations.

Following the introduction of mathematical models of OEOs, we delve into our research work of the narrowband OEO-based RC, which is the first time a narrowband OEO is introduced into the field of machine learning for a time-delay reservoir computer implementation in Ch. 3. In this chapter, we develop the mathematical model required for studying the narrowband OEO-based RCs and introduce the concepts of OEO-based RCs. We also numerically simulate this narrowband OEO-based RC and demonstrate its suitability for processing IQ-modulated signals. Lastly, we train and test the narrowband OEO-based on IQ-modulation classification tasks, which reaches state-of-the-art performance with a reduced training set size.

In Ch. 4, we further extend the narrowband OEO-based RC to the field of RF fingerprinting - a technology that is widely used to identify RF transmitters. In this chapter, we thoroughly evaluate the performance of the narrowband OEO-based RC across a wide range of benchmark datasets. Our simulation results demonstrate the suitability of the narrowband OEO-based RC for RF fingerprinting. Moreover, we once again show that the narrowband OEO not only requires significantly less training data for the IQ modulation classification task, as presented in Ch. 3, but also maintains excellent performance under limited-resource conditions, extending its effectiveness to RF fingerprinting.

In Ch. 5, we further push the narrowband OEO-based RC to an extreme by further limiting its computational source and training data size. We experimentally test the narrowband OEO-based RC's ability in such scenarios. Meanwhile, we propose a new metric named NET to measure ML platforms' performances across different algorithms that takes into account the complexity of the algorithm, the performance, and data size required for training. Lastly, we also show that this metric is useful to quantitatively analyze the diminished returns of scaling the number of parameters in machine learning algorithms.

It is known that nearly all OEO-based RCs require an expensive and bulky external modulator, which poses a significant obstacle to their practical applications. To resolve this issue, in Ch. 6, we, for the first time, introduce the concept of the directly laser-modulated OEOs for RCs (DL-OEO based RC). In our proposed scheme, we cancel the external modulator and replace it by directly modulating the laser, which not only significantly reduces the system size, but also the total budget for implementation an OEO-based RCs. This implementation makes our DL-OEO based RC one of the simplest OEO-based RCs ever known. First, we start by deriving mathematical models for this proposed scheme. Then, we show our numerical results of using the DL-OEOs for reservoir computing. Lastly, we discuss the usefulness of the nonlinearity introduced by the DL-OEO and how it could contribute the a significant improvement in performance in comparison with traditional OEO-based RCs.

In Appx. I, we revisit the concepts of time-division multiplexing, time-delay differential equation, and how to use them to numerically simulate the OEO-based RC.

In Appx. II, for the first time, we show the relationship between the number of cavity modes N_{CM} supported by OEO and the total number of virtual nodes N required by OEO-based RCs. We conduct extensive simulations of both types of OEO-based RCs (including narrowband and wideband configurations) across a diverse set of datasets. Our simulation shows that the total number of virtual nodes N required for sufficiently good performance of OEO-based RC can be as less as the total number of cavity modes N_{CM}.

PHOTONIC RESERVOIR COMPUTING
BASE ON OPTOELECTRONIC OSCILLATOR

by

Haoying Dai

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor Yanne. Kouomou Chembo
Professor Daniel Lathrop
Professor Thomus Murphy
Professor Kevin Daniels
Professor Gong Cheng

Acknowledgments

I owe my gratitude to many people I met before and during my graduate school journey - without whom, this thesis would not be possible.

First and foremost, I would like to thank my advisor, Professor Yanne, who took me into his group years ago. Without him, I would never be able to start this journal. In the past years, with his profound knowledge, he patiently taught me and guided me through countless challenges in the projects. I owe my deepest gratitude to him not only for the wealth of knowledge he has generously taught me, but also for the warmth and kindness. His influence has inspired me to grow, not just intellectually but also personally. Indeed, he is such an excellent exemplar that makes me not only a better researcher but also a better human being.

My heartfelt gratitude extends to Professor Daniel Lathrop, Professor Thomus Murphy, Professor Kevin Daniels, and Professor Gong Cheng for their graciously agreeing to serve on my dissertation committee and for their invaluable feedback on the manuscript.

I am deeply grateful to my colleagues at the Photonics Systems Lab for Aerospace and Communication Engineering (PSACE Lab): Dr.Fengyu Liu and Dr. Benjamin Hudson Klimko, for their collaboration and insightful discussions, which significantly enrich my knowledge and inspire my research.

I am also grateful to the whole IREAP and ECE department at the University of Maryland, College Park, which provides an unparalleled environment for both my research and study. This gratitude extends to all the professors and lecturers who taught me in both the ECE and CS departments.

Furthermore, I wish to express my appreciation to the Astronomy Department, which offers me a Teaching Assistantship and the opportunity to teach undergraduate students and improve my teaching skills, the Minta Martin Fellowship, the Northrop Grumman - University of Maryland Seed Grant Program, and the Office of Naval Research for their financial support.

As always, I want to thank my friends Xindong He and Xin Bao. I have known you since six, and after all of this many years, you are still and always there.

Here, I also want to express my special thanks to Mr. Huan Tang, who taught me math in middle school, and Mr. Changyu Yang, who taught me English in high school, for their lifelong impacts on me as teachers.

In our Chinese culture, there is a phrase that can be translated into English as " greatest gratitude cannot be expressed in words". This is the reason that I would prefer not to say thanks to my parents and my family here. For all that they have done for me, I have nothing to return but remain silent and remember well.

Lastly, thank God!

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Analog-To-Digital Conversion
AI	Artificial Intelligence
AM-DSB	Amplitude Modulation - Double Sideband
AM-SSB	Amplitude Modulation - Single Sideband
AMP	Amplifier
ANN	Artificial Neural Networks
ASIC	Application-Specific Integrated Circuits
AWG	Arbitrary Waveform Generator
BPSK	Binary Phase-Shift Keying
BFSK	Binary Frequency-Shift Keying
BP	Back-Propagation
BRNN	Bidirectional Recurrent Neural Networks
CNN	Convolutional Neural Network
CM	Cavity Modes
CPFSK	Continuous Phase-Frequency-Shift Keying
CPU	Central Processing Unit
CW	Continuous-Wave
DAG	Directed Acyclic Graph
DARPA	Defense Advanced Research Projects Agency
DDEs	Delayed Differential Equations
DFB	Distributed Feedback Laser
DFB-LDs	Distributed Feedback Laser Diodes
DL	Delay Line
DNNs	Deep Neural Networks
DSP	Digital Signal Processing
EO	External Modulator
ELM	Extreme Learning Machine
ESNs	Echo-State Networks
E/O	Electrical-To-Optical
FBGs	Fiber Bragg Gratings
FC-NNs	Fully Connected Neural Networks
FC-ONN	Fully Connected Optical Neural Network
FDL	Fiber Delay Line
FPGA	Field Programmable Gate Array
FIR	Finite Impulse Response
FSR	Free-Spectral Range
FWM	Four-Wave Mixing
GHz	Gigahertz
GPUs	Graphics Processing Units

GRU	Gated Recurrent Unit
High-Q	High-Quality
IoT	Internet Of Things
InP	Indium Phosphide
IQ	In-Phase/Quadrature Components
I	In-Phase Component
JLT	Jet Propulsion Laboratory
LD	Laser Diode
LLMs	Large Language Models
LMC	Linear Memory Capacity
LSMs	Liquid State Machines
LSTM	Long Short-Term Memory
MAP	Maximum A Posteriori
MC	Memory Capacity
MCtot	Total Memory Capacity
MHz	Megahertz
ML	Machine Learning
MRR	Microring Resonator
MVM	Microwave Coupler
MZI	Mach-Zehnder Interferometer
MZM	Mach-Zehnder Modulator
NARMA10	Tenth-Order Nonlinear Autoregressive-Moving-Average
NCE	Nonlinear-Channel Equalization
NMT	Neural Machine Translation
NMSE	Normalized Mean Squared Error
O/E	Optical-To-Electrical
OEO	Optoelectronic Oscillator
OIUs	Optical Interference Units
ONNs	Optical Nonlinear Units
ONR	Office Of Naval Research
ONN	Optical Neural Network
OSI	Open Systems Interconnection
PCA	Principal Component Analysis
PCs	Principal Components
PC	Polarization Controller
PCMs	Phase Change Materials
PD	Photo Detector
P1	Period-One
P2	Period Doubling
PAM4	Pulse-Amplitude Modulation - 4 Level
PID	Proportional–Integral–Derivative
PICs	Photonic Integrated Circuits
PSD	Power Spectral Density
Q	Quadrature Component
QPSK	Quadrature Phase-Shift Keying
RC	Reservoir Computing

ReLU	Rectified Linear Unit
RF	Radio Frequency
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RoF	Radio-Over-Fiber
RBS	Rule Based System
SAI	Symbolic Artificial Intelligence
SC	Signal Combiner
Si3N4	Silicon Nitride
SNR	Signal-To-Noise Ratio
SOA	Semiconductor Optical Amplifier
SOI	Silicon-On-Insulator
SVM	Support Vector Machine
TPUs	Tensor Processing Units
TS	Training Samples
UAV	Unmanned Aerial Vehicle
VNs	Virtual Nodes
VVA	Voltage Variable Attenuator
WB-FM	Wideband Frequency Modulation
WDM	Wavelength Division Multiplexing

Disclaim

Chapters 3 - 6 are based on a published or submitted article and have been adapted to conform to the formatting and structural requirements of this thesis.

The relevant publications are:

- Chapter 3 is adapted from: ***Haoying Dai***, and Yanne K. Chembo. ”Classification of IQ-modulated signals based on reservoir computing with narrowband optoelectronic oscillators.” *IEEE Journal of Quantum Electronics*, 57(3):1–8, 2021.
- Chapter 4 is adapted from: ***Haoying Dai***, and Yanne K. Chembo. ”RF fingerprinting based on reservoir computing using narrowband optoelectronic oscillators.” *Journal of Lightwave Technology*, 40(21):7060–7071, 2022.
- Chapter 5 is adapted from: Benjamin H. Klimko, ***Haoying Dai***, and Yanne K. Chembo. ”Resource-constrained narrowband optoelectronic oscillator-based reservoir computing for classification of modulated signals.” *Optics Letters*, 49(13):3608–3611, 2024.
- Chapter 6 is adapted from: ***Haoying Dai***, and Yanne K. Chembo. ”Photonic Reservoir Computing Using Optoelectronic Oscillators With Direct Laser Modulation.” *Journal of Lightwave Technology*, 2024.

All chapters have been revised where necessary to maintain consistency of style, formatting, and notation throughout the thesis.

If you look at this universe, you will find a small blue planet located in the third place near the sun in the solar system in the Oriohkn arm inside the Milky Way galaxy - one of the many barred spiral galaxies in the Local Group that belongs to the Virgo Supercluster. This small blue planet is called Earth, where about 15 million species dwell. Among these species, more than 90% remain unknown. For these known, about 300 thousand are plants, 610 thousand are fungi, and 8 million are animals. There is one species among all the animals named human. It has a population of 8.1 billion, and they deem themselves as the only one that is "intelligent". They feel lonely and desire for equals. Thus, they eagerly looked at the starry sky above, setting up astronomical telescopes one after another, enthusiastically sending countless probes from Voyager I to new Horizons, and continuously monitoring signals from the deep universe day by day without a moment of rest – yet, they sadly found none. Thus, they dreamed (decided) to create one (by their own): artificial intelligence. That is the beginning of the story.

Chapter 1: General Introduction: Where Artificial Intelligence and Microwave Photonics Cross

1.1 Artificial Intelligence (AI)

When searching for the terminology “Artificial Intelligence (AI)” in the scientific literature, one might be surprised to find the first usage of the word was quite “recent”. The first formal usage of AI as a scientific term was in 1956, at a conference in Dartmouth organized by John McCarthy. This conference is recognized as the “official birth of AI as a field”. But the concept of creating a machine that shares human intelligence could be dated back way earlier. In the year of 1920, the Czech writer Karel Čapek first created the term “robot” in his most famous science fiction Rossum’s Universal Robots where a company named Rossum created robots that are organic and resemble humans’ intelligence. Since then, the term robots is widely accepted as a type of automatons that have a similar appearance and soul to humans. However, in modern industry, the term robots often refers to mechanic systems that are programmed and designed to execute a set of complicated actions. Before the invention of term robots, a close term android was created by the French novelist Eugène Fromentin in his 1880 novel “L’Inhumain”. The term itself derives from the ancient Greek word “aner” (man) and “eides” (form or shape) – which has the meaning of “man-like” or “man-form”. Meanwhile, the idea of human-crafted automatons with human shape could be traced back to the early history of human beings. For example, during the Warring States Period (about 475 BC -221 BC, also known as Zhanguo), the ancient Chinese sage philosopher Liezi wrote a chapter entitled Tangwen to record folk legends, fables, and mythological stories in the work named after himself, Liezi. In this chapter, Liezi recorded a very interesting story named Carpenter Creates Humans - a famous carpenter presented King Mu of Zhou a human-made automaton as a gift. It vividly described how the automaton behaves like the beings, even the King himself could not tell the difference between the automaton and a real human being. Only by removing the wooden head, the automaton could not talk, by removing the man-made kidney, the automaton could not walk, by removing the liver, the automaton could not see; did the King Mu of Zhou realize how delicate the design was? Meanwhile, in ancient Greek (around 700BC), a myth described a giant automaton named Talos was built by the god Hephaestus to protect the island of Crete. However, there is a subtle difference between Talos and the automaton mentioned by Liezi. Talos is a giant bronze man made by god – which makes himself closer to a species that is a counterpart of human-beings. Meanwhile, before the year of 1956, there are a few early important theories in the scientific field that deserve attention. In the year of 1950, the British mathematician and logician Alan Turing introduced the concept of Turing Completeness and the Universal Turing Machine. It is at the same year, Turing proposed the imitation game to test the machine's "intelligence" level to that of human beings. A machine is regarded as equivalent, or indistinguishable from human beings, if humans could not tell a machine from a human in a text-based conversation without knowing the identities they have dialogue with at first. This was later known as the Turing test. Before we delve into scientific advances in "artificial intelligence", a question must be asked or answered first - what is "intelligence"?

1.1.1 What is Intelligence?

The separation from philosophy marks the formation of a discipline as a distinct field. The study of human minds, and behavior and examining their mental process became an independent discipline in 1879 when Wilhelm Wundt (1832-1920) established the first experimental psychology laboratory at the University of Leipzig, Germany. As a pioneer of psychoanalysis, Sigmund Freud (1856-1939) developed a theory about the structure of the human mind, which is intertwined with his understanding of intelligence.

Freud's theory on the structure of the mind provides a framework for understanding human behavior, thought processes, and personality. In this theory, human minds are composed of three parts: Id, Ego, and Superego. The id represents the unconscious, instinctual drives and desires. It is governed by the pleasure principle, seeking immediate gratification of basic needs and urges, such as hunger, thirst, and sexual desires. The id operates irrationally and is not concerned with reality, morality, or social norms. Meanwhile, the ego develops to mediate between the id's demands and the realities of the external world. It operates on the reality principle, seeking to satisfy the id's desires in a realistic and socially appropriate manner. The ego is responsible for higher cognitive functions, such as reasoning, planning, and problem-solving. Finally, the superego represents internalized societal and parental standards of morality. It acts as a conscience, guiding the ego toward ethical behavior. Freud believed that much of human behavior and thought, including aspects of intelligence, are influenced by unconscious processes. According to Freud, the unconscious mind stores repressed memories, desires, and experiences that continue to affect conscious thoughts and behaviors. Intelligent behavior, therefore, is not only the product of conscious reasoning but is also shaped by unconscious motivations. However, the unconscious mind could also hinder intelligent behavior. Repressed conflicts or unresolved psychological issues could lead to irrational actions or cognitive distortions. Another trend in early psychology studying human intelligence is known as psychometrics - a field that is devoted to testing, measuring, and assessing mental abilities.

Modern psychology views intelligence as a complex, multifaceted construct, moving beyond earlier definitions that emphasized a single, fixed ability. It encompasses various theories and models that recognize different types of intelligence, the influence of genetic and environmental factors, and the importance of cultural and contextual variables. In 1983, Howard Gardner proposed that intelligence is not a single, unified construct but rather a collection of multiple, distinct intelligences. Each individual has a unique combination of these intelligences. In his initial list of intelligences, he included a total of seven types of intelligence: linguistic intelligence, logical-mathematical intelligence, musical intelligence, bodilykinesthetic intelligence, spatial intelligence, interpersonal intelligence, and intrapersonal intelligence. This list was further expanded by him later. Similarly, in 1985, Robert J. Sternberg argued that intelligence is not limited to analytical abilities measured by traditional IQ tests. Instead, it encompasses three components: analytical Intelligence, creative intelligence, and practical intelligence. In 1991, Daniel Goleman proposed the concept of Emotional intelligence, which refers to the ability to perceive, understand, manage, and regulate emotions in oneself and others.

In recent years, with the advent of information theory and complexity science, a new perspective has emerged: intelligence is perceived as a property of systems that can manage and process information. Informatics refers to the science of information processing, including the collection, classification, storage, retrieval, and dissemination of recorded knowledge. In the context of intelligence, informatics focuses on how systems handle information to adapt, learn, and solve problems. Intelligence can be conceptualized as the capacity to process information in a way that enhances an organism's or system's ability to adapt to its environment. Complexity theory [9] [10] studies how interactions between components of a system give rise to emergent properties that cannot be understood by examining the components individually. It focuses on systems that exhibit non-linearity, feedback loops, adaptability, and emergence. Intelligence can be seen as an emergent property that arises from the complex interactions of simpler components [11]. For example, individual neurons interact in networks that give rise to consciousness, thought, and intelligent behavior [12]. Self-organization is a key concept in complexity theory and is crucial for understanding intelligence. In self-organizing systems, order and complexity arise from local interactions without a central controller. Humans' neural networks self-organize based on experience, leading to learning and adaptation. In biological systems, intelligence is a product of both informatics and complexity. The neuronal system processes vast amounts of information using a complex network of neurons. This information processing is highly efficient, allowing for quick adaptation to changing environments. Complexity theory helps explain how the neuron structure and function result in emergent properties like consciousness and cognition [10].

1.1.2 What is Artificial Intelligence?

In the previous section, we have already seen the controversies on the definitions of intelligence. These ongoing debates remain heated to this day. For example, Noam Chomsky, one of the founders of modern linguistics, proposed the theory of universal grammar, suggesting that all human language shares a common underlying structure. He argued that humans are born with an innate ability to learn language, indicating a universal set of grammatical principles hardwired into the mind. Any single language's grammar is just a subset of the universal grammar. His argument follows the traditions of the rationalists. Meanwhile, Geoffrey Hinton, a British-Canadian computer scientist widely recognized as one of the founders of deep learning is strongly against the idea of Chomsky. In his opinion, languages are acquired through learning - denying the existence of innate language abilities, which aligns with the tradition of the empiricists.

As an extension of the intelligence, definitions of artificial intelligence can be tricky, circular, and even controversial. In the literal sense, artificial intelligence simply means intelligence made by humans. For example, John McCarthy [13] defines AI as a science and engineering that aims to simulate human intelligence processes by machines, particularly computer systems. With that definition, one could naturally raise a question: are software programs AI? For example, in 1997, Deep Blue - a program invented by IBM successfully defeated the world champion Garry Kasparov in a chess six-game match. This was considered a milestone in the development of AI at that time. However, from a modern perspective, Deep Blue is a software program that exploits tree-trimming algorithms to decide the best moves - all of its decisions are hard-programmed and deterministic. Moreover, it can not learn from the data, which simply means it is not different from a calculator that can perform addition, subtraction, multiplication, and division much faster and more accurately than human beings. Thus, one might draw the strange conclusion that a calculator also has intelligence if a computer program can be accepted as AI. This could lead to an even “weirder” conclusion that a hard drive storing voice information and a video recorder also have intelligence since they can recite a poem like humans.

Meanwhile, researchers like Tom Stonier will argue that intelligence should not be a single phenomenon, but a wide spectrum of phenomena [14]. However, even if Stonier’s argument avoided the awkwardness of regarding a video recorder as intelligence, it raises a new paradox: how wide should the spectrum of phenomena should be that is enough to be considered as “intelligence”? In practice, the requirement for human equivalent intelligence will result in the requirement for general AI - which is still impossible for humans now. What’s worse, if we push that conclusion a little longer, one might even supersede the scope of anthropocentric intelligence and conclude that there is no such thing as intelligence. Therefore, recent researchers started to define artificial intelligence as a program that can learn and improve its performance from the inputs - a key characteristic of philosophers and psychologists deem as intelligent, leading to a new field known as machine learning. However, we think that definition is incomplete for AI. First, learning or adapting is only a part of intelligence. Second, learning might not be the most difficult part of AI [15], even if it is essential to build AI. Lastly, there is also an argument that distinguishes learning from intelligence, i.e. learning is the process of acquiring intelligence - its outcome rather than the process itself is the AI.

Therefore, it becomes essential to define artificial intelligence with greater nuance and precision. Currently, in both industry and academia, one way to navigate the ongoing debates is to distinguish between two categories: weak AI and generational AI. Weak AI is also known as Narrow AI. It refers to artificial intelligence systems designed and trained to handle specific tasks or applications. Typically, a weak AI is task-oriented, which means that it is programmed to perform specific tasks or solve particular problems and excels in its designated areas but lack the ability to perform tasks outside of its programming. Another important characteristic of weak AI is that it does not possess self-awareness or consciousness. It operates based on algorithms and data without any understanding or awareness of its actions.

In contrast, general AI, also known as Strong AI, refers to hypothetical AI systems with the ability to understand, learn, and apply intelligence across a wide range of tasks, similar to human cognitive abilities. Except for the general cognitive abilities, Strong AI is also envisioned to have self-awareness, consciousness, and an understanding of its own existence, similar to human intelligence.

At present, due to technological limitations and our incomplete understanding of the true nature of intelligence, all existing AI systems remain within the realm of weak AI. As a rapidly advancing field in both research and industry, weak AI has experienced unprecedented growth and has become increasingly integrated into everyday life. To distinguish AI from software, software is often defined as a program that performs specific, predefined tasks that follow explicit and static instructions coded by developers and can not learn from the data or adapt itself. Meanwhile, AI often involves machine learning, where the system improves its performance over time as it is exposed to more data.

1.1.3 Classical Techniques of AI

Currently, the most widely accepted and utilized approach to achieving AI in both academia and research is through machine learning (ML). Before exploring machine learning in detail, it is helpful to briefly review several historically significant techniques that were foundational in the development of AI.

The 1960s marked the first golden era of artificial intelligence, a period during which many foundational concepts were introduced and AI research attracted significant funding, particularly from governmental organizations such as the Defense Advanced Research Projects Agency (DARPA). In 1958, John McCarthy introduced LISP, the first programming language specifically designed for AI. A year earlier, in 1957, Frank Rosenblatt developed the perceptron, a model now recognized as a single-layer neural network [16].

However, the enthusiasm surrounding AI at the time led to inflated expectations and overly optimistic predictions about the capabilities of these early systems. In 1969, Marvin Minsky and Seymour Papert published a critical analysis showing that perceptrons were incapable of solving certain basic problems, such as computing the XOR function [17]. This revelation led to a significant decline in interest and investment in neural network research, ultimately triggering a broader retreat in AI development known as AI's first winter.

While connectionism encountered a cold, **Symbolic Artificial Intelligence** (SAI) became the dominant research field on AI from the 1950s to the 1990s. This method, often referred to as classical AI now, was considered the ultimate solution for AI during that period. Its most successful applications are the rule-based systems or the expert systems.

The **Rule Based System** (RBS) was initially proposed as a concept in the early 1960s and became mainstream during the revival of AI in the 1970s. The philosophy behind this approach is to list a set of rules and program the computer to reason based on these predefined logics, mimicking the human decision-making process. It typically includes a knowledge base and a general-purpose reasoning process to solve problems in a specific domain. For example, DENDRAL [18] developed at Standford University in the 1960s is one of the earliest rule-based systems. Its major purpose was to assist scientists in determining the structure of organic molecules from their mass spectrometry data. Its success has inspired the MYCIN system - one of the most successful rule-based systems for blood infection diagnosis in the mid-1970s. MYCIN, a medical expert system developed in the 1970s, was designed to identify bacteria responsible for sepsis and recommend appropriate antibiotic dosages. Remarkably, its diagnostic performance was comparable to that of human medical experts.

The development of Symbolic AI relies on a range of tools and algorithms, including but not limited to, logic programming, production rules, and semantic networks. Meanwhile, it is also worth mentioning that since the era of machine learning, especially the surge of neural networks, many hybrid algorithms that use machine learning algorithms and neural networks together with symbolic AI have been developed. For example, in order to incorporate uncertainty reasoning into symbolic AI systems, hidden Markov models and Bayesian networks have been popularized in this field, which led to the development of Markov Logic Networks or Probabilistic Soft Logic. In recent years, there has been a resurgence of interest in Neuro-symbolic AI—a field that seeks to combine the strengths of both neural networks and symbolic AI to enhance the robustness, reasoning, learning capabilities, and cognitive functions of artificial intelligence systems.

1.1.4 Machine Learning for AI

Throughout history, humanity has progressed through three major stages of language: natural language, mathematical language, and programming language. Today, machine learning emerges as the fourth and most recent language - a powerful tool for understanding, modeling, and describing the world around us. While mathematics provides an axiomatic language that enables a more precise and quantitative description of the world than natural language, programming languages take this a step further by translating these descriptions into sets of executable, predefined processes or actions. Machine learning, as a new language built upon the foundations of mathematics and programming, takes a fundamentally critical step forward - it learns executable processes directly from data, rather than relying on explicitly programmed instructions. Originally developed as a statistical tool for data analysis, machine learning evolved rapidly, soon surpassing the boundaries of traditional statistics and emerging as an independent research field in its own right.

Based on the different types of data used to train the algorithms, machine learning can be classified into the following categories. Supervised learning refers to a type of machine learning where the data used to train the algorithm are labeled. The machine learning algorithm learns a mapping from the input to the output so that the model can predict the correct label for new, unseen data. Meanwhile, unsupervised learning involves training a machine learning algorithm on data without labels. The goal is to discover patterns, structures, or relationships within the data. It is often used for clustering or dimensionality reduction. Semi-supervised learning falls between supervised learning and unsupervised learning. It uses a dataset where only a small portion is labeled. The model is trained on both the labeled data and the unlabeled data, leveraging the structure in the unlabeled data to improve learning. Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent receives feedback in terms of rewards or penalties and aims to maximize the total rewards. RL differs from other forms of machine learning because it focuses on learning a strategy and is well-suited to problems when a long sequential decision-making process is involved.

In the past decades, the field of machine learning, particularly techniques based on artificial neural networks (ANNs), has experienced unprecedented growth and transformation. The pace of development has been so rapid that new frameworks are introduced and state-of-the-art performance benchmarks are surpassed every few weeks. To better understand this evolution from a technical perspective, we categorize the history of machine learning into three distinct stages: the era of classical machine learning, the rise of deep neural networks, and the dominance of large language models (LLMs).

1.1.4.1 Classical Machine Learning

Classical machine learning refers to the techniques developed prior to, or alongside, the emergence of deep neural networks, but notably, excluding neural network-based approaches. It dominated the field before the rise of deep learning and is characterized by algorithms with well-established mathematical and statistical foundations. Unlike deep learning, classical methods are typically more interpretable and grounded in mature theoretical frameworks. These algorithms have proven effective across a range of tasks in supervised, unsupervised, and semi-supervised learning. However, one of their key limitations lies in their dependence on manual data preprocessing and feature engineering—the process of extracting relevant features from raw data to achieve high performance. This step often requires significant domain expertise and effort.

In the research field of supervised learning, labeled data - i.e., data with known outcomes - can be categorized based on the type of predicted value: discrete or continuous. When the label is a discrete value, the task is known as a classification problem, where the model learns to assign inputs to predefined categories or classes. In contrast, when the label is a continuous value, the task becomes a regression problem, where the model is trained to predict a range of numeric values, often forming a continuous output or trend over time. Here, we provide an overview of some of the most important and widely used classical machine learning algorithms.

Linear regression algorithm is considered one of the most straightforward and simplest regression algorithms. Given an output variable y and input variable vector $\mathbf{x}$, a linear regression model assumes that the relationship between output variable y and input variable vector $\mathbf{x}$is linear. It is a good method when the relationship between input and output is known to be linear. Moreover, the linear regression can be extended to multivariate linear regression, where the output variable y is assumed to be linearly related to multiple variables. However, it is not recommended in most application scenarios since it makes an oversimplified assumption about the data being processed in the real world. The meaning of this algorithm is that it paved the way for the gap between statistics and machine learning.

Despite its name, logistic regression is actually a classification algorithm, not a regression method. It is one of the simplest and most commonly used techniques for binary classification, predicting the probability that an input belongs to one of two categories. Much like linear regression, logistic regression can be generalized to multiclass classification problems, where the model outputs the probability distribution across multiple categories. Logistic regression is particularly effective for linearly separable problems and demonstrates a certain robustness to noise. However, it shares a key limitation with linear regression: its decision boundary is linear, making it poorly suited for complex or nonlinear relationships in the data. Additionally, logistic regression is prone to overfitting, especially when irrelevant or redundant features are present. Its performance heavily depends on effective feature selection and preprocessing, as failing to identify and include the correct independent variables can significantly degrade model accuracy.

The decision tree is a supervised machine learning approach that can solve both regression and classification problems. The operating principle of this algorithm is that it continuously divides the data according to predefined parameters stored in leaves. Using a tree data structure, the input data passes through nodes at different layers in the tree and finally falls into a bottom node where the output is specified for a certain category (classification problem) or a value (regression problem). The merits of the decision tree algorithm are obvious. First, it can solve both classification and regression problems much more explainably compared with other machine learning algorithms. Moreover, the tree traversal algorithms allow it to operate at a fast speed. Lastly, the data required to train the decision tree is small. However, it is known that the decision tree could easily overfit the data. The tree size grows faster when increasing the training dataset size, making the tree data structure unmanageable. Thus, the decision tree is only suitable for low-resource machine learning problems. Another issue is that the decision tree can only be trained to a local optimal solution, which further limits its performance.

Support Vector Machine (SVM) [19], introduced by Vladimir Vapnik at Bell Labs in 1995, was considered one of the most powerful and versatile supervised machine learning algorithms before the era of deep learning. Taking advantage of the concept of kernels, it optimizes the margins of each class to the decision hyperplanes. The kernel trick offers the possibility to map input data into an arbitrarily high-dimensional feature space. As a max-margin machine learning algorithm, SVM is resilient to noise and guaranteed to reach the global optimal solution. The algorithm can also be easily extended from classification to regression problems. As one of the most heavily studied machine learning algorithms, the mathematical theories behind SVM are clear and elegant. SVM can handle structured and semi-structured high-dimensional data that require complex decision boundaries. Given proper kernel functions, SVM, in theory [20], has the capacity to universally approximate arbitrary functions. Lastly, SVM is also known for not getting stuck in local optima. The major critiques of SVM are as follows: first, its performance deteriorates significantly as the dataset size increases. Moreover, finding suitable kernel functions can be difficult. Lastly, as a max-margin classification algorithm, its final output is not a probability, which makes the results harder to interpret.

Naïve Bayes is one of the simplest probabilistic supervised machine learning algorithms. It assumes conditional independence between all pairs of feature variables given the output variable and applies Bayes' theorem to simplify the probability distribution equations. Then, Maximum A Posteriori (MAP) estimation is applied to estimate the probability of a certain input. Naïve Bayes has the following advantages: it works well on small training datasets and is easy to implement. It can estimate the probability of binary or multi-classification problems. The major disadvantage of this algorithm is that the Naïve Bayes assumptions can sometimes be too simple for real-world applications, since the features can be highly correlated.

Compared to Naïve Bayes, the Bayesian network is a more complicated probabilistic machine learning model that involves the graphical representation of the conditional probabilities or dependencies between different variables. The graph used to represent the probabilistic dependencies between the variables is a directed acyclic graph (DAG). Each edge inside the graph represents a direct conditional dependency, while each node represents a variable with an underlying Bayesian assumption: it might be an observable quantity, a latent variable, or some unknown parameters.

1.1.4.2 Artificial Neural Networks

Artificial neural networks have revolutionized the whole machine learning field by their unprecedented performance in complex tasks. The concept of the first artificial neural network is not recent—it can be dated back to the era of the perceptron. Despite the proof that the perceptron can't solve the XOR problem, which caused the first AI winter and the transition of mainstream AI technologies to symbolic AI, the impact of deep neural networks (DNNs) is significant.

The idea of getting “deep” - which simply means adding an extra layer into the fully connected perceptron model to solve the XOR problem—has a far-reaching impact: it indicates the revival of fully connected neural networks, a predecessor of a type of AI technology that has come to dominate—and still dominates—the AI field today. Dating back to 1989, Cybenko [21] proved that fully connected neural networks (FC-NNs) with three or more layers could universally approximate arbitrary functions, which saved the endangered connectionism in AI. Even though the deep neural network was proposed very early—the first fully connected neural networks trained by the backpropagation algorithm can be traced back to 1967 in an article published by Amari, *et al.* [22]. However, it was not until the last two decades that neural networks became the dominant machine learning algorithm. This is related to the fact that the training and testing of neural networks not only require extremely large datasets but also have strict requirements for computational power. It was not until 2003 that major research interest in deep learning was brought back by a publication by Yoshua Bengio, *et al*, which gained huge success in language modeling using a deep, fully connected neural network [23].

Except for the fully connected neural networks (multi-layer perceptrons), many deep neural network structures have been proposed and utilized in many other fields. The Convolutional Neural Network (CNN) is one of the most important deep learning techniques in the computer vision and image processing research fields. The first CNN in the modern sense that specializes in processing image data was proposed in 1989 by Yann LeCun [24], *et al.*, to recognize ZIP codes. The fifth version of this CNN, proposed in 1998 - the LeNet-5, is the most well-known of this series of works. The LeNet-5 features 3 convolutional layers with 2 subsampling layers and a fully connected layer. With a total of 60,000 parameters, it was designed to identify the images of handwritten digits in the MNIST dataset and reached state-of-the-art performance at that time. The invention of LeNet-5 convinced the industry and the research community that CNN is a promising technique for image-related tasks. Moreover, convolutional layers and pooling layers introduced by LeNet have become standard and have been widely used in modern CNNs. Lastly, its structure has inspired nearly all the CNNs afterward. For example, the ReLU activation function—the most commonly adopted activation function in modern CNNs - was hugely inspired by the sigmoid function in the LeNet.

The emergence of AlexNet in 2012 [25], one of the most famous CNN architectures, completely revolutionized the field and marked a new era of computer vision research. Before AlexNet, CNN-based techniques were still in intense competition with traditional machine learning techniques like SVMs. The outperformance of AlexNet by such a large margin compared to traditional machine learning algorithms set a completely new standard for computer vision performance. What's more, it adapted LeNet-5 to a structure with deeper layers—8 layers—and overcame the vanishing gradients problem to achieve unprecedented high performance. Its success drew the attention of scientists to the importance of depth in the design of deep neural networks and inspired the invention of nearly all larger and deeper neural network structures thereafter, including but not limited to VGG, GoogLeNet, and ResNet. Since then, making neural networks deeper has been a trend and a golden rule in designing DNNs. During its development, it catalyzed the usage of graphics processing units (GPUs) for fast training and testing of such neural networks. It also revealed the potential of exploiting super-large datasets with deeper layers to gain higher performance.

Simultaneously, the recurrent neural network (RNN) and its variants changed the research field of processing sequential data, including but not limited to time series, natural language, and audio. One of the most important differences between recurrent neural networks and other feedforward neural networks is that RNNs feature feedback loops inside their structure. These feedback loops, often referred to as recurrent links, offer RNNs the flexibility to track long-temporal information. In the research field of artificial neural networks, the first RNN can be traced back to Rosenblatt [26], Frank's discussion of a three-layer perceptron model, where the middle layer was connected back to the first layer in 1962.

Then the research on RNNs fell into a quiet period until the resurgence brought by the Jordan Network [27] and Elman Network [28] in the 1980s. Even though these two neural networks only have three layers - not deep enough in the modern sense of deep learning, they feature the most important characteristics, as well as the basic building blocks that are required for modern RNNs in deep learning. These simple yet elegant structures pioneered most RNNs that are still used today.

Meanwhile, in the deep learning era, RNNs are often based on two main structures that are suitable for building deep-layered models: long short-term memory (LSTM) [29] networks, which were invented by Hochreiter and Schmidhuber in 1995, and bidirectional recurrent neural networks (BRNN) [30], invented by Schuster in 1997. Since then, LSTM has become the default setting for RNNs in deep learning, as it mitigates the exploding and vanishing gradient problems that RNNs often encounter. In 2006, the BRNN based on LSTM — or bidirectional LSTM — began transforming speech recognition by surpassing traditional models in specific applications. It enhanced large-vocabulary speech recognition [31], text-to-speech synthesis [32], and was integrated into Google Voice Search and Android dictation. Additionally, LSTM set new benchmarks in machine translation, language modeling, and multilingual language processing. When combined with convolutional neural networks (CNNs), it also contributed to significant advancements in automatic image captioning [33].

Compared to traditional RNNs, BRNNs feature two hidden layers that process the input information in opposite directions, increasing their ability to memorize temporal information. It is worth mentioning that the Gated Recurrent Unit (GRU) [34], originally introduced in 2014 and often serving as a simplified version of the LSTM, has a gating mechanism to forget certain features of the inputs. This feature allows it to maintain the same level of performance as LSTM while greatly reducing the total number of parameters. Meanwhile, as a powerful tool that excels at processing sequential data, RNNs have also evolved to process sequential data with more complicated structures. For example, a video is often considered a sequence of 2-dimensional data (frames). Thus, a combination of CNN and RNN, also known as CNN-RNN [35] structures, has been proposed. Compared to 3D CNNs, this type of structure has the benefit of remembering long temporal information and is more suitable for extracting logical connections and

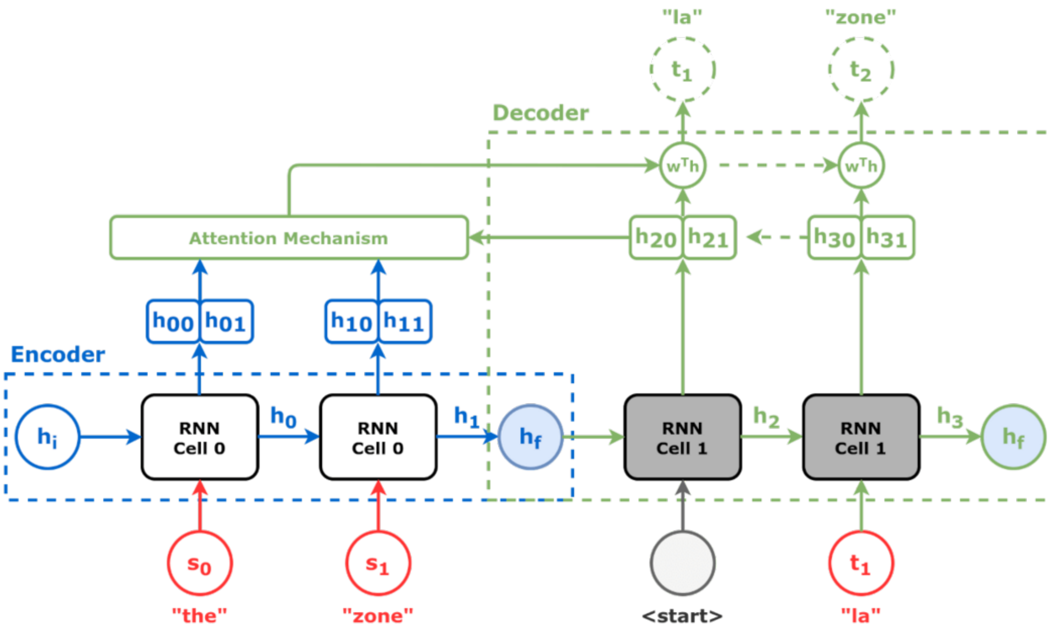


Figure 1.1: Encoder-Decoder structure implemented on RNN with attention mechanism. This figure is contributed by Daniel Voigt Godoy from `https://github.com/dvgodoy/dl-visuals/`.

potential dependencies between a frame and its long-past predecessors. PixelRNN[36] often serves as a generative model to predict the nearby pixels in both directions.

Among all the variants of the RNN, one of the most important structures that later revolutionized the whole machine learning field is the encoder–decoder model. The concept of the encoder and decoder originates from digital logic circuit design, where the encoder takes 2^N inputs and converts them into N outputs, while the decoder does exactly the opposite. The encoder–decoder in RNNs is similar to that of the digital logic circuit. As shown in Figure 1.1, it uses two RNNs connected front to back - one RNN serves as the “encoder”, and the other serves as the “decoder”. The idea behind the encoder–decoder structure is inspired by the application of neural machine translation (NMT): the first RNN is used to encode the source language into a sequence of hidden vectors that serve as an intermediate representation, and then the second RNN acts as the “decoder” that decodes these intermediate representations into the translation of the second language. With a proper attention mechanism, this model achieved state-of-the-art performance in neural translation from 2014 to 2017 and also inspired one of the most important inventions of a type of super scalable artificial neural networks - large language models (LLMs).

1.1.4.3 Large Language Models

In 2017, Vaswani Ashish, *et al.*, published an article titled “Attention Is All You Need” [37], which introduced the Transformer – a model that later became the foundation of modern neural network research and the cornerstone of famous AI applications, including but not limited to ChatGPT and Gemini. In this article, they proposed self–attention mechanisms to replace the RNN. This mechanism is so significant that it has become the default setting for nearly all large language models.

There are many advantages of Transformers over RNNs. First, instead of processing one word at a time, the self–attention mechanism allows Transformers to process all the words in parallel. Moreover, the self–attention mechanism ensures that, even though Transformers handle sequential data in parallel, the model still retains long-term dependencies without losing information. In addition, the self-attention module is more scalable compared with traditional RNNs – in theory, its number of parameters can be infinite. This feature is crucial because it paves the way for industrial applications – all you actually need are data and computational resources! Since then, large companies like Apple, OpenAI, and Google have started building LLMs upon Transformers and entered the field of deep neural networks with parameter counts in the billions. Lastly, it is worth mentioning that LLMs support pre–training followed by fine-tuning, enabling task-specific adaptation at a reduced cost.

1.1.5 Reservoir Computing

In contrast to all the previously mentioned ANNs that try to reduce errors by increasing the total number of parameters, reservoir computing is an ANN with minimal design. As a variant of recurrent neural network (RNN), it only has three layers inside: the input mask, the reservoir layer and the read-out

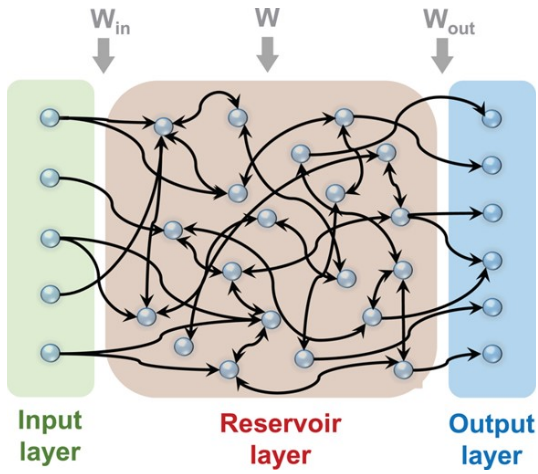


Figure 1.2: The reservoir computing forms a minimal style recurrent neural network. Only weights W_out in the read-out layer are trained, while all other weights are randomly initialized. This figure is reproduced from ref. [1].

layer. This structure greatly simplifies training by keeping the internal network - called the reservoir - fixed and only trains the read-out layer. The reservoir is a dynamic system that transforms input signals into a rich set of spatiotemporal patterns, enabling the system to capture complex temporal relationships.

1.1.6 Reflections: The More, The Better?

Historically, we have observed a trend in the development of the semiconductor industry - both in hardware and machine learning – toward the evolution from complex modules with limited scalability to simpler modules that are highly scalable. With the rise of parallel processing units, GPUs have become the primary computational resource, replacing the earlier role of complex, sequential processing units inside CPUs. Similarly, in machine learning, the complex and less scalable modules found in CNNs or RNNs have been replaced by the uniform and simple self–attention mechanism used in Transformers. It seems we have entered an era where nearly all available linguistic resources of humanity are being used to train large language models. Major companies have been drawn into a path of investing billions of dollars in GPUs and building AI infrastructure to enhance their capabilities. Even the surprisingly low–cost large language model DeepSeek, recently developed in China, required millions of dollars for training. Many researchers have fallen into a myth: that the only way to build better, more intelligent AI is to spend more money. Some even believe that LLMs and scale are the gold standard, and that other machine learning algorithms are obsolete.

However, diminishing returns from increasing the size of LLMs have become apparent. Scaling up computation resources by a factor of 10 no longer yields equivalent performance gains. DeepSeek has demonstrated that with less than 1% of the expenditure, a model can still achieve performance close to the most advanced systems like GPT–4. Moreover, in many research fields, it remains extremely costly and difficult to gather enough data to train these artificial neural networks – especially large models.

1.2 Microwave Photonics

Microwave photonics is a multidisciplinary field that investigates the interaction between optical and microwave signals. Specifically, it involves the use of photonic devices to generate, process, transmit, and detect microwave and millimeter-wave signals. The field has demonstrated significant potential in a variety of applications, including broadband wireless access networks, sensor networks, radar systems, and satellite communications. Over the past two decades, numerous research topics have emerged, with several gaining substantial traction and adoption in both academia and industry. These include photonic generation of microwave and mm-wave signals, radio-over-fiber (RoF) systems, analog-to-digital conversion (ADC) enabled by photonic architectures, and optically controlled phased array antennas. In recent years, machine learning (ML) techniques have been increasingly integrated into microwave photonics research. Broadly, this integration follows two principal directions. The first focuses on employing machine learning algorithms to assist in the modeling, analysis, and optimization of microwave photonic systems, with the goal of enhancing system performance, robustness, and adaptability. This includes applications such as parameter tuning, fault diagnosis, and behavioral prediction of complex photonic subsystems. The second direction explores the use of photonic systems themselves as computational platforms for implementing machine learning algorithms. Due to the inherent high-speed and parallel processing capabilities of photonic hardware, this approach presents a promising pathway for developing ultra-fast, low-latency machine learning accelerators. Such systems could be especially advantageous for real-time, high-throughput applications in signal processing and intelligent communication systems.

1.2.1 Generation of Microwave and mm-wave Using Photonic Devices

It is highly desirable to design microwave or millimeter-wave (mm-wave) sources with ultra-low phase noise and frequency tunability. These characteristics are critical for applications in radar, satellite systems, and sensing technologies. Traditionally, high-frequency microwave sources are implemented using electronic circuits that rely on multiple stages of frequency doubling or up-conversion. However, this process inevitably introduces phase noise and signal degradation. Moreover, when the generated microwave signal must be delivered to a remote location, transmitting it via free space or electrical cables (e.g., coaxial lines) becomes impractical due to high transmission losses. To address this limitation, radio-over-fiber (RoF) technology—where microwave or mm-wave signals are transmitted through optical fibers via modulated optical carriers—has been proposed and is now widely adopted in modern communication systems. As a result, microwave or mm-wave sources that are inherently compatible with RoF systems can significantly simplify system architecture and reduce hardware complexity. Photonic generation of microwave and mm-wave signals has emerged as a promising solution. This approach not only integrates seamlessly with RoF systems but also offers significant advantages over traditional analog electronic circuits. In particular, the generated microwave or mm-wave signal is the beat note of two optical carriers, eliminating the need for electronic frequency scaling or multiplication. Finally, photonic-based microwave and mm-wave sources are intrinsically less vulnerable to electromagnetic interference, further enhancing their reliability and robustness in complex environments. In general, assumes that we have two optical signals at two different frequencies:

$$E_1(t) = E_{01}\cos(\omega_1 t + \phi_1) \tag{1.1}$$

$$E_2(t) = E_{02}\cos(\omega_2 t + \phi_2) \tag{1.2}$$

where E_{01} and E_{02} are amplitudes, ω_1 and ω_2 are angular frequencies, and ϕ_1 and ϕ_2 are phases. When these two optical signals form a beat node at a photo detector (PD), a radio frequency signal can be generated with following equation:

$$I_{RF} = A\cos[(\omega_1 - \omega_2)t + (\phi_1 - \phi_2)] \tag{1.3}$$

where A is the new amplitude of the RF signals.

1.2.2 Radio-over-Fiber (RoF) systems

An optical fiber offers the advantages of ultra-low loss (typically around $0.2\ dB/km$), extremely high bandwidth, and resistance to electromagnetic interference. These qualities make it the foundation of modern communication systems. The transmission of radio waves through optical fiber has been extensively studied and widely applied in industrial settings. The concept of the first radio-over-fiber (RoF) system dates back to as early as 1990 [38], when a four-channel 2G telephone signal was successfully transmitted through a single-mode fiber using subcarrier multiplexing. The baseband signal, operating in the frequency range of 864 MHz to 868 MHz, directly modulated a carrier signal generated by a multi-longitudinal mode laser operating at 1300 nm. To increase system capacity, various multiplexing techniques such as wavelength division multiplexing (WDM) and network topologies including tree [39] and ring [40] architectures [2] were proposed and investigated. In recent years, with the rapid expansion of 5G cellular networks and Internet of Things (IoT) applications, the demand for advanced RoF systems has significantly increased. To meet these demands, RoF systems operating at millimeter-wave and even terahertz frequencies have been explored [41]. To address the power consumption challenges of such high-frequency systems, new approaches like optically powered RoF systems [42] have also been developed. Notably, due to the extremely low loss and ultra-wide bandwidth of optical fibers, no signs of saturation or fundamental limitations have yet emerged for RoF systems. It is therefore reasonable to conclude that RoF, as a mature and robust technology, will continue to serve as a core component of modern communication systems in the foreseeable future.

1.2.3 All Optical Microwave Signal Processing

In recent years, a noticeable shift has occurred in the research community—from merely transmitting microwave signals through radio-over-fiber (RoF) systems to processing microwave signals using all-optical methods. This trend is driven by several key factors. First, RoF is a mature and well-established technology, extensively studied from 1990 to 2015. It remains highly effective in supporting modern communication systems, including fifth-generation (5G) networks, with minimal modifications. As a result, its research momentum has naturally slowed. Second, signal processing is inherently more complex than signal transmission. Photonic devices operate at much higher frequencies—typically in the terahertz range—compared to traditional electronic circuits, which function in the megahertz (MHz) to gigahertz (GHz) range. Since a system's processing capability is generally tied to its clock rate, photonic devices have a natural advantage in terms of speed and bandwidth. Lastly, electronic circuits capable of operating at high frequencies tend to be costly and often incompatible with standard communication infrastructure. This makes the idea of processing microwave signals entirely in the optical domain particularly attractive to both research and industry, offering a promising pathway to achieving higher performance with lower cost and greater scalability.

1.2.3.1 Using Optical Fiber as Delay-Line Filters

It can be dated back as early as 1976 that optical fibers can serve as a delay line [43] to process signals, as they have low loss and high bandwidth. The key idea behind this scheme is to take advantage of a group of parallel delay lines with delays equal to $T, 2T, ..., nT$, using a finite impulse response (FIR). When these responses are combined, they form all-optical filters. In real-world implementations, these parallel optical fiber delay lines can be realized with optical couplers [44, 45], Fiber Bragg Gratings (FBGs) [46], or arrayed waveguides [47]. Other implementations of parallel optical fiber delay line arrays exploit negative refractive index [48] or complex refractive index [49], offering a wider dynamic range and tunability."

1.2.3.2 Optical Implementation of Logical Gates

One of the most direct ways to boost the operating frequency of circuits is to implement the logic gates with all-optical devices, which are not only compatible with communication systems but also exploit the ultra-fast processing speed of optical devices. However, there is an underlying challenge to such implementations: electron-electron interactions can be easily manipulated in the presence of an electromagnetic field, whereas controlling photon-photon interactions on a large scale and maintaining precision under noise remains difficult. Moreover, compared to their counterpart, CMOS devices in electronic circuits which can be integrated at the million or even billion-device level per chip, the integration of photonic devices is still limited to at a small scale.

The earliest idea of implementing all-optical flip-flops can be traced back to 1975, when optical bistability [50]was observed in a cavity with nonlinear effects. Since then, this property has been exploited to fabricate all-optical flip-flops, known as cavity-based all-optical logic gates. In 1980, the first experiment demonstrating the feasibility of this concept was conducted. Using Fabry-Pérot etalons [51], a logic gate was constructed by coupling two optical triode switches. However, this experiment used large, bulky, and discrete photonic devices, which required high energy consumption to produce a strong nonlinear effect, making it unsuitable for large-scale implementations. Following this, research has focused on microcavities with the potential for integration while maintaining low power consumption. In the early 2000s, micro-ring laser-based cavities were proposed as bistable switches with trigger energies as low as femtojoules (fJ) per gate [52]. Meanwhile, other passive implementations of all-optical logic gates were also proposed. For example, 2D photonic crystal nanocavity-based flip-flops were theoretically demonstrated [53] and experimentally examined [54]. In 2014, leveraging large-scale photonic crystal nanocavity arrays built on silicon photonic devices, researchers achieved 105-bit operations [55]. More recently, all-optical and all-in-one systems using liquid metal plasmon nonlinearity have also been proposed and tested [56]. Nonetheless, progress in cavity-based or cavity-array-based all-optical logic gates remains in the laboratory stage. The technology is still far from industrial applicability due to limitations in total operating bits.

Aside from cavity-based all-optical logic gates, semiconductor optical amplifier (SOA)-based all-optical gates have also garnered significant attention. Compared to cavity-based implementations, SOA-based gates have an intrinsic advantage in scalability, as manufacturing an SOA on a wafer is much easier than fabricating a cavity. Additionally, SOA-based gates have lower power consumption. Similar to their cavity-based counterparts, SOA-based all-optical gates rely on the nonlinear effects within the optical medium. Initially, for a single SOA device, only one logic function could be implemented using a specific nonlinear effect, such as four-wave mixing (FWM). However, it was soon discovered that multiple logic functions could be implemented in a single SOA, processing signals in parallel by simultaneously exploiting all available nonlinear effects [57]. This discovery greatly enhances the potential of SOA-based logic gates, as there are various nonlinear effects within the SOA, including, but not limited to, cross-gain modulation (XGM), cross-phase modulation (XPM), and cross-polarization modulation.

Research on all-optical logic gates is still facing challenges in on-chip integration, although current progress in all-optical logic devices using photonic crystals, silicon ring resonators, and plasmonic microstructures [56] has already significantly reduced the feature size to the level of hundreds of μm.

1.2.3.3 Integrated Microwave Photonics

In recent years, microwave photonics has seen a research trend towards integration, that is, the integration of all photonic and electronic components into a single chip. Integrating all components onto a single chip not only significantly reduces the overall size of the microwave photonic system but also promises lower costs and greater compatibility with other components, such as RF circuits in communication networks. The advancements in the fabrication processes of indium phosphide (InP), silicon-on-insulator (SOI), and silicon nitride (Si_3N_4) have enabled many impressive applications. For example, Ref. [58] demonstrates a plasmonic modulator with an ultra-small size that achieves a bandwidth of 170 GHz. Moreover, researchers have also demonstrated the possibility of a programmable signal processor by integrating the building blocks of the Mach-Zehnder interferometer (MZI) into two-dimensional (2D) meshes, which can then be programmed to synthesize a variety of optical functionalities, including filters, delay lines, and unitary $N \times N$ transformations [59].

1.3 Photonic Machine Learning

In previous sections, we have briefly described the history of AI technologies and recent advances in microwave photonics. It can be seen that, in recent years, connectionism has become the dominant philosophy in AI research. Various variants of artificial neural networks (ANNs) have emerged and become the leading approaches in different research areas and industrial applications. Traditionally, most of these artificial neural networks run on digital circuits—ranging from central processing units (CPUs) and graphics processing units (GPUs) to tensor processing units (TPUs) and application-specific integrated circuits (ASICs), targeting the acceleration of specific machine learning algorithms. Meanwhile, modern communication frameworks heavily rely on fiber-based photonic systems that offer low loss and ultrawide capacity. Moreover, photonic devices operate at much higher frequencies compared to electronic devices, presenting the potential to significantly improve signal processing capabilities. Thus, it is natural to wonder whether it is possible to develop machine learning systems that operate on photonic devices. Over the past decades, there has been a surge in research interest in developing such photonic machine learning systems, accompanied by various research trajectories and diverse platform architectures being proposed.

1.3.1 Photonic Accelerator for Traditional Machine Learning Algorithms

While artificial neural networks dominate contemporary AI research, traditional machine learning algorithms, such as principal component analysis (PCA), remain widely used in applications that require low computational resources and work with limited data. In addition to using photonic universal computation platforms or photonic logic gates to implement and accelerate these algorithms, it is worth mentioning other research that takes advantage of nonlinear effects within photonic devices and exploits these nonlinearities for superfast computing. For example, in digital circuits, a Fourier transform often involves multiple stages of digitization and multiplication. Meanwhile, light propagating through a concave lens undergoes a Fourier transform. These properties were originally exploited for computation, as mentioned in previous sections, and have also been demonstrated to be suitable for machine learning applications. For instance, in Ref. [60], a microring resonator (MRR) weight bank is used as a photonic machine learning platform to perform PCA analysis. The authors experimentally demonstrate a two-channel photonic PCA system, showing strong agreement between the resulting principal components (PCs) and those derived from traditional software-based PCA. Ref. [61] proposes the first application of four wave mixing (FWM)-based nonlinear signal processing within the context of machine learning and neuromorphic computing. Through numerical simulations, they demonstrate that FWM can serve as a flexible and effective nonlinear activation function, with the potential to enhance dimensionality reduction in extreme learning machine

(ELM) architectures.

1.3.2 Photonic Artificial Neural Networks

The increasing demand for machine learning and artificial intelligence in fields such as telecommunications and high-performance scientific computing has spurred interest in new hardware solutions. At the same time, the development of digital electronics for artificial neural networks has revealed key limitations in this domain, most notably, issues related to processing latency. To address this challenge, photonic computing has emerged as a promising direction, driven by advancements in optoelectronic components and photonic integration platforms. Photonic integrated circuits (PICs) have enabled ultrafast artificial neural networks, forming the foundation for a new generation of information processing systems.

In modern artificial neural networks, the computation paradigm often differs significantly from traditional computer algorithms supported by the Von-Neumann architecture. ANNs typically require large-scale parallel processing of data rather than sequential computing, meaning that highly distributed, low-latency computations are more important and favored than ever. The computations within an ANN can generally be divided into two types of highly parallel operations: linear operations and nonlinear operations. Additionally, the communication time between processing units must be minimized. These requirements make photonic circuits particularly well-suited for implementing high-performance processing units for artificial neural networks, due to two key advantages: their ability to support dense interconnectivity at superfast speeds and their efficiency in performing both linear and nonlinear operations. It is known that the most frequently computed operations in artificial neural networks, such as CNNs, RNNs, or LLMs, are matrix multiplication (dot product), addition, and nonlinear operations (e.g., sigmoid or ReLU). In photonic circuits, optical signals can be efficiently multiplied by transmitting them through tunable waveguide elements and then added using wavelength-division multiplexing (WDM) to accumulate carriers in a semiconductor [62, 63] or through electronic currents [64, 65]. Moreover, nonlinear transformations

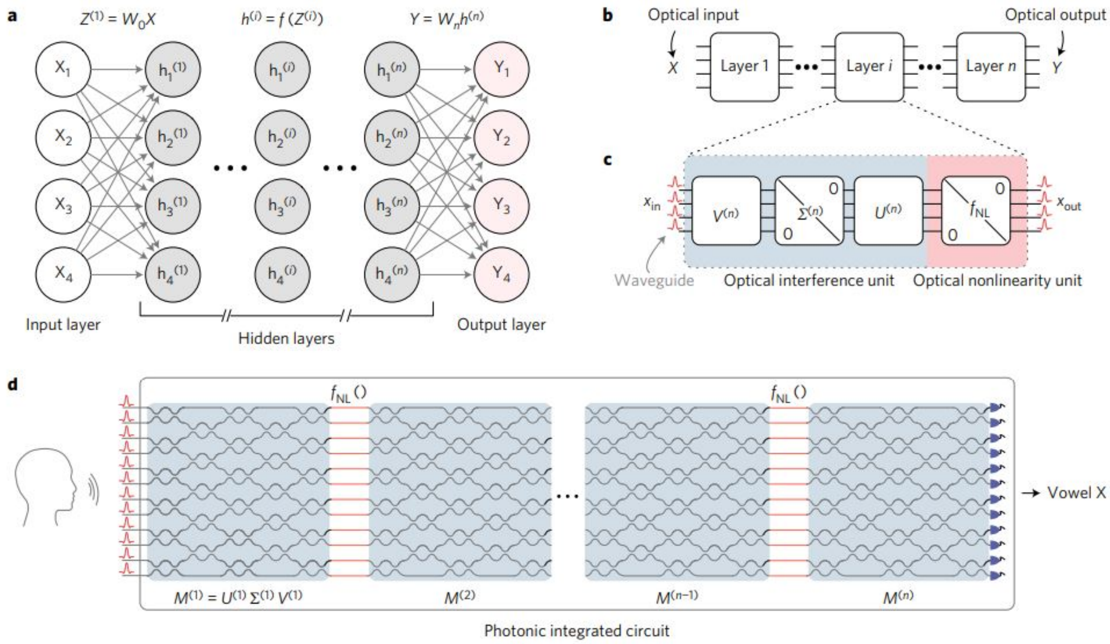


Figure 1.3: Fully connected optical neural networks implemented by MZI to perform weights multiplication. This figure is reproduced from ref. [2].

can be efficiently implemented using the intrinsic nonlinear effects within photonic devices. Compared to the digital implementations of data centers built with distributed computing clouds, photonic data centers offer significant advantages in interconnecting each computation unit using optical waveguides or optical fibers, which have much lower loss, higher bandwidth, and generate less heat than metal waveguides. Finally, one major advantage of photonic devices is their potential for parallel computations, further enhancing their suitability for high-performance applications.

In recent years, with the rapid development of silicon photonics, the integration of photonic devices has experienced a large-scale increase, making it possible to create a scalable photonic computational platform for implementing artificial neural networks. For example, Ref. [2] proposes a fully connected optical neural network (FC-ONN) using an array of Mach-Zehnder interferometers (MZIs) in silicon photonics for programmable weight multiplications, and optical interference units (OIUs) to implement optical nonlinear units (ONUs) that perform activation functions. The proposed scheme is shown in

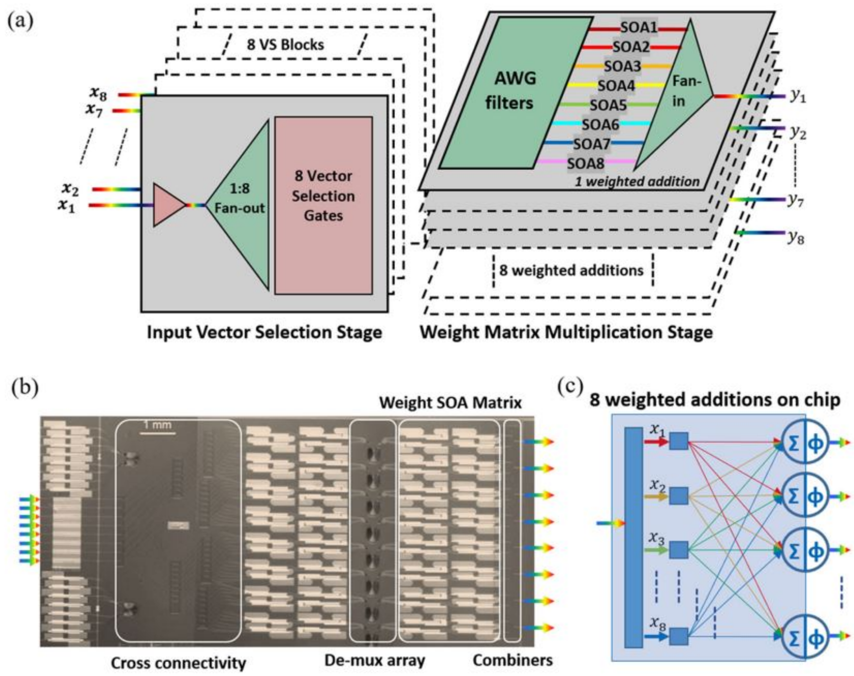


Figure 1.4: Fully connected optical neural networks implemented by array of semiconductor optical amplifiers(SOAs). This figure is reproduced from ref. [3].

Figure 1.3. This implementation of the FC-ONN has improved latency by two orders of magnitude compared to fully connected neural networks based on digital circuits. For the implementation of optical neural networks, two key factors must be considered: first, how to leverage photonic devices for matrix multiplication, and second, how to exploit nonlinear effects to function as activation functions. This opens up the possibility of using many different candidates, rather than relying solely on MZIs. For example, the microring weight bank mentioned in Ref. [66] could also serve as the weight multiplication mechanism for an optical neural network (ONN). Similarly, as shown in Figure 1.4, an array of InP-based semiconductor optical amplifiers (SOAs) is proposed for weight multiplication in an ONN [3]. Additionally, as shown in Figure 1.5, a FC-ONN utilizing phase change materials (PCMs) for matrix multiplication and wavelength-division multiplexing (WDM) multiplexers as nonlinear activation functions has been proposed [67]. Apart from fully connected neural networks, other types of photonic deep neural networks, such as

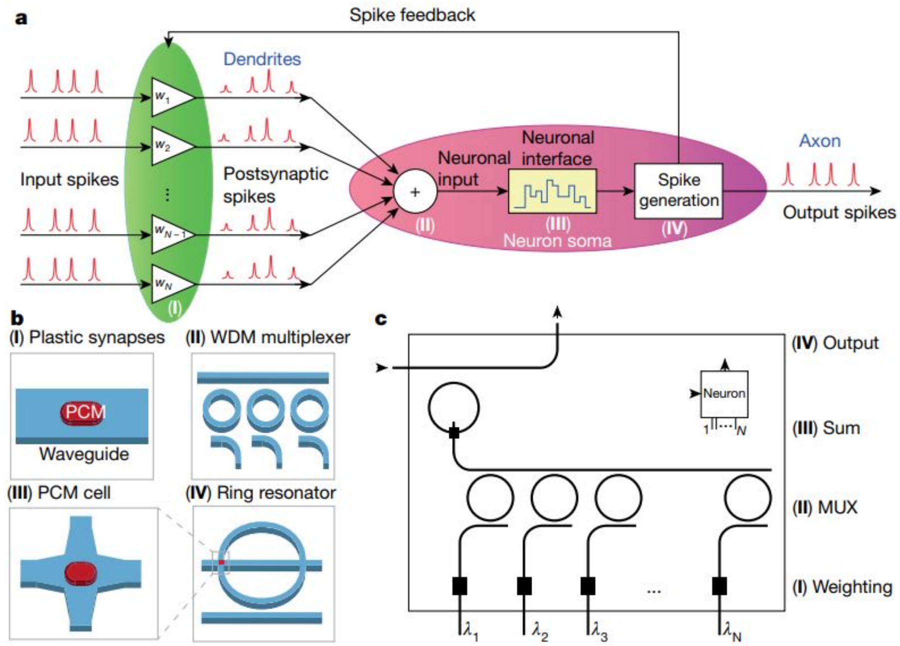


Figure 1.5: Fully connected optical neural networks implemented by phase change materials. It uses WDM multiplexer to perform nonlinear activation functions for neural networks.This figure is reproduced from ref. [3].

photonic convolutional neural networks (CNNs) and photonic recurrent neural networks (RNNs), have also been proposed. For example, Ref. [68] proposes and examines a large-scale photonic CNN based on distributed feedback laser diodes (DFB-LDs) that leverages biological time-to-first-spike coding for performing convolutions. Similarly, as in Ref. [3], Ref. [69] also exploits SOAs for weight multiplication and incorporates WDM multiplexers for nonlinear activation. However, instead of FC-ONNs, they experimentally demonstrated the first all-optical RNN with a gating mechanism. The list of technologies and research trajectories for all-optical neural networks using integrated photonic devices is extensive and continues to grow.

Parallel computation plays a crucial role in enabling the high-speed implementation of artificial neural networks. In integrated photonic systems, individual components within the photonic circuit, each performing fundamental operations such as multiplication or addition in parallel, constitute what is commonly referred to as device-level parallelism. However, the inherent capability of photonic devices

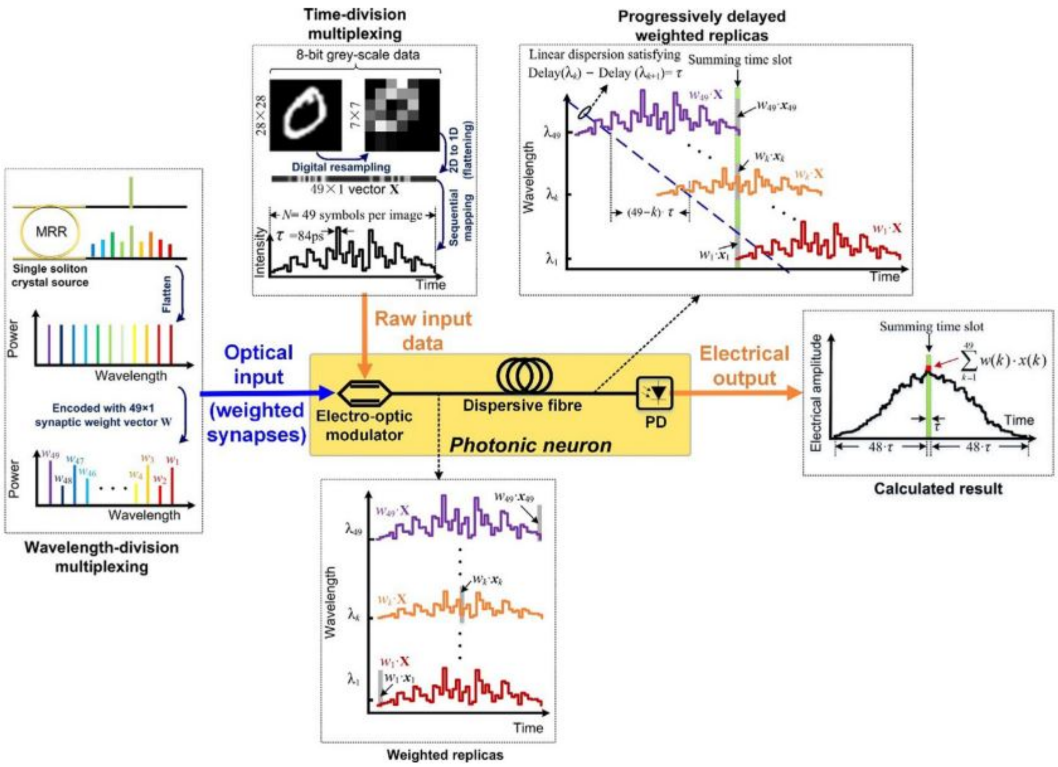


Figure 1.6: Fully connected optical neural networks implemented by taking advantage of Kerr-combs. It forms a frequency level parallelism for fully-connected artificial neural network implementations. This figure is reproduced from ref. [4].

to facilitate parallel computation extends far beyond this device-level scope, offering a broader and more profound potential for computational acceleration. It is important to note that there are various forms of parallelism inherent in photonic systems. These parallelisms have already been widely accepted and utilized in optical communication research, typically under the concept of “multiplexing”. If leveraged by photonic computation techniques, these forms of parallelism could unlock significantly enhanced computational capabilities. Moreover, these parallelisms are compatible with the device scaling-law of photonic integrations, making them even more promising for large-scale applications. For example, Ref. [4] studies a fully connected optical neural network (FC-ONN) implementation based on a microring resonator (MRR). As shown in Figure 1.6, an MRR generates an optical Kerr-comb with frequency components evenly distributed in the frequency domain. Each comb is then modulated by the weights of a layer within

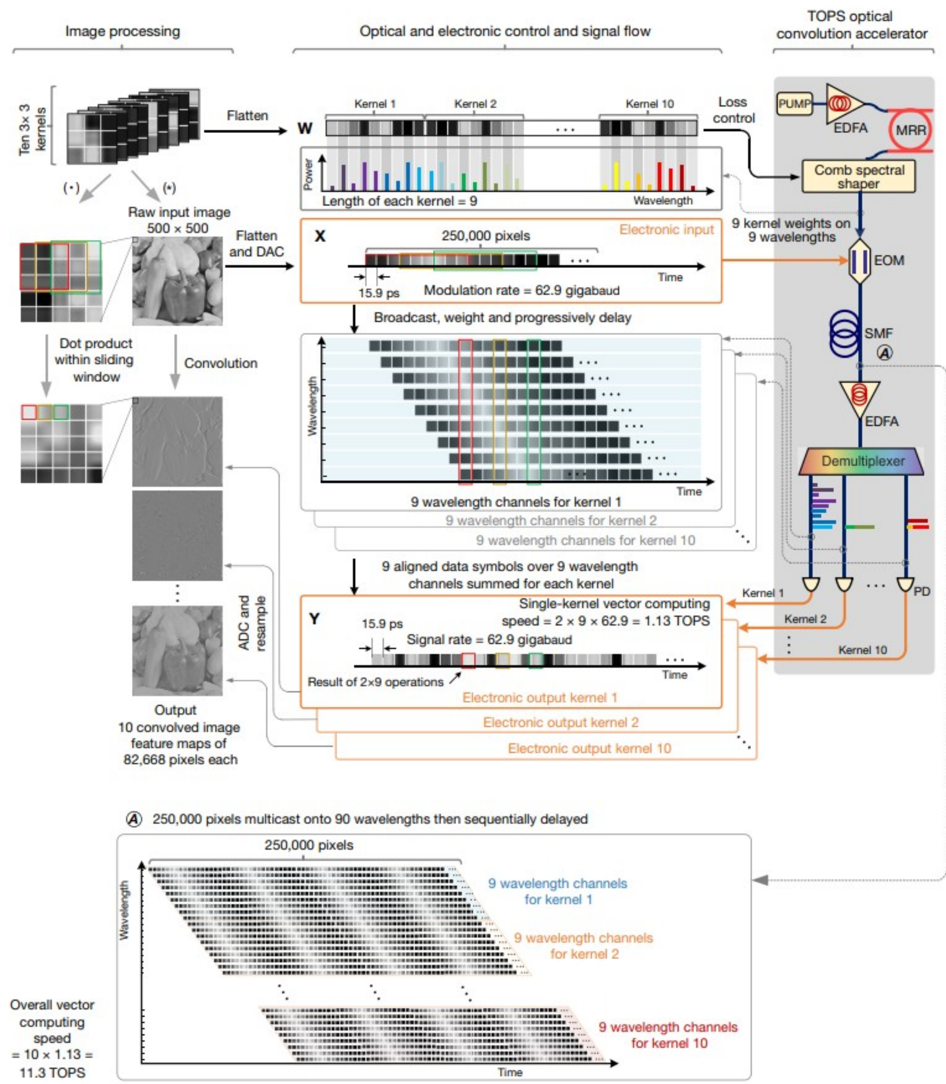


Figure 1.7: Photonic CNNs implemented by Kerr-combs. This figure is reproduced from ref. [5].

the deep neural network in the temporal domain. A photodiode (PD) is used to sum the weights of the network. These Kerr-combs together form a fully connected neural network. Compared to previously discussed methods, this approach exploits parallelism in both the frequency and temporal domains. It is important to note that this method is not limited to FC-ONNs. Figure1.7 illustrates how it can be adapted to implement a convolutional neural network (CNN) [5]for image classification, achieving a processing speed of 11 Teraflops.

1.3.3 Photonic Delay Line based Reservoir Computer

As mentioned in Sec. 1.2.3.1, a delay line is often used to process signals as filters in microwave photonics. In 2011, a key property of delay lines - its infinite dimensionality in the temporal domain,was found to be extremely useful for the implementation of a machine learning algorithm known as reservoir computing. In that framework, expanding information in the temporal domain via a delay line was proposed for constructing a reservoir computer. Since then, this approach has become a hot research topic and numerous schemes have been proposed and studied. Since the 1980s, recurrent neural networks (RNNs) have been widely used to process temporal or sequential signals. The primary training method for RNNs is the gradient-based backpropagation algorithm, which is computationally expensive and does not always guarantee a globally optimal solution. In the early 2000s, echo-state networks (ESNs) [70–72] and liquid state machines (LSMs) [73] were independently proposed and studied. These two types of artificial neural networks share similar concepts with reservoir computing but differ significantly from traditional RNNs. One of the most notable differences is that these networks avoid the use of backpropagation algorithms.

Reservoir computing (RC), a variant of recurrent neural networks, has a much simpler structure. It typically consists of three layers: the input layer, the reservoir layer, and the output layer. The inputs are projected into a much higher-dimensional space after being multiplied by the input weights, making the data more separable. The reservoir layer performs similarly to the recurrent layers in an RNN, giving RC the ability to retain long-term information. Unlike most artificial neural network models widely used today, RC avoids the time-consuming backpropagation algorithm. One of RC's most notable advantages over other types of ANNs is that it can be trained with significantly less data, making it particularly suitable for low-resource machine learning scenarios.

The idea of implementing a reservoir computer using a delay line was initially proposed in 2011. As

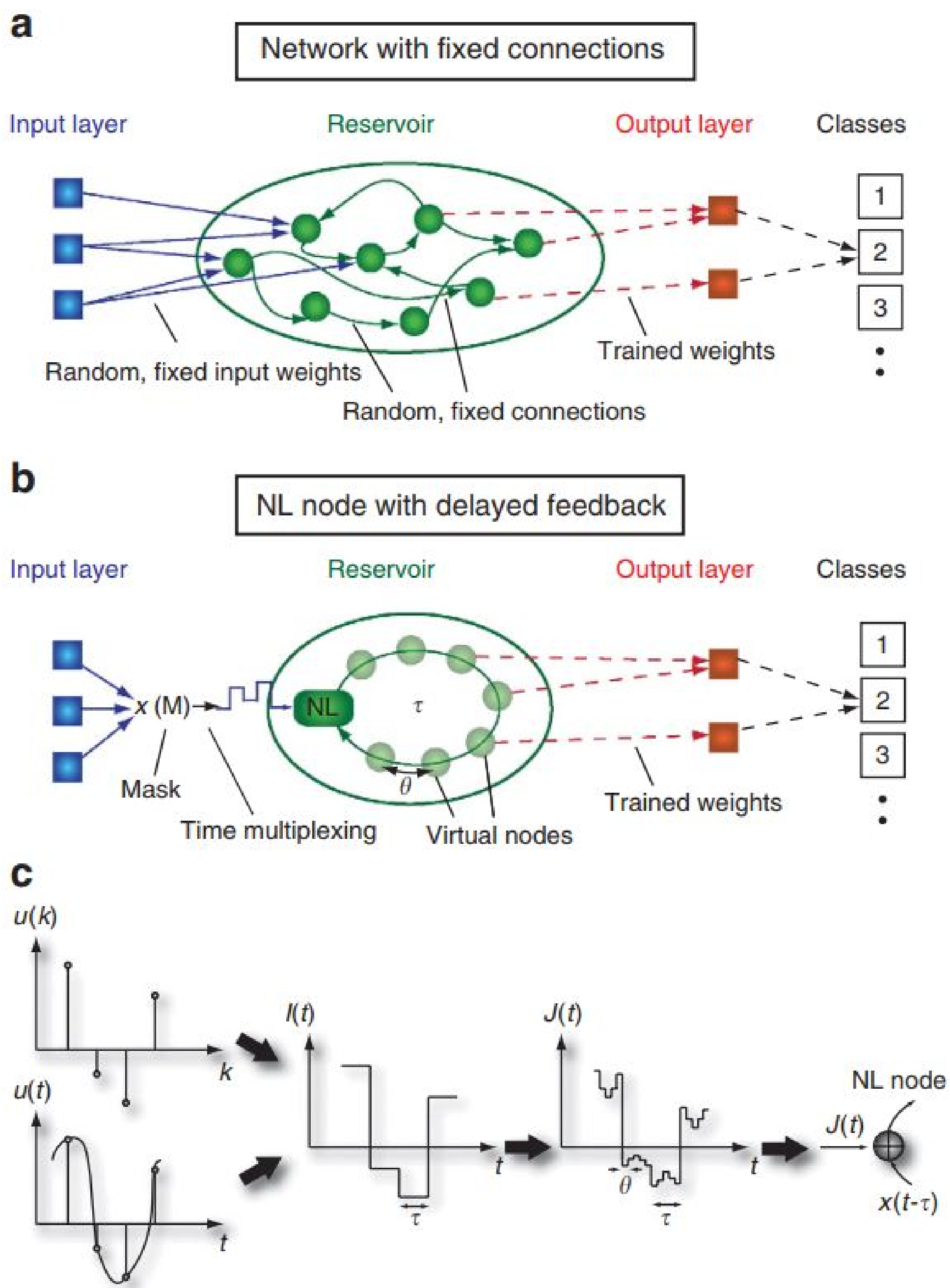


Figure 1.8: Photonic CNNs implemented by Kerr-combs. This figure is reproduced from ref. [6].

shown in Figure 1.8, Ref. [6] introduces the concept of temporal multiplexing to project the input signal into a higher-dimensional space for the reservoir layer. In such implementations, the temporal information in the delay line is considered as neurons equally distributed along the delay line. One key advantage is that this idea can be widely applied to systems with a delay line and nonlinearity. For example, Ref. [74] proposes using a digital Proportional–Integral–Derivative (PID) controller with a feedback delay line to implement delay line based RC, Ref. [75] utilizes an active-ring resonator comprising a magnetic thin-film delay line for better integration, Ref. [76] uses a three-stage monolithic integrated amplified feedback laser that concatenates a distributed feedback laser (DFB) laser, a passive phase section, and an amplifier section to form a feedback loop, and Ref. [77] uses Mackey-Glass time-delay system for implementing an RC. Currently, the number of time-delay-based RC implementations with different structures and dynamic systems is still skyrocketing and the rate of increase in related publications is expected to continue growing rapidly in the future.

Among all of these time-delay based reservoir computing implementations, the implementation of reservoir computers based on optoelectronic oscillator (OEO) has been extensively studied and is the most widely accepted. It was not accidental that the first proposed time-delay photonic reservoir computer (RC) was based on an optoelectronic oscillator (OEO). The suitability of the OEO for photonic RC implementations can be justified by the following reasons. First, the optoelectronic oscillator (OEO) is a key device widely used in optoelectronics, offering exceptional compatibility with modern communication systems and infrastructure. It not only contains a photonic component that can be seamlessly integrated into optical fiber communication systems, but also includes a radio frequency (RF) component capable of directly handling and processing RF signals generated by technologies such as Wi-Fi, Bluetooth, and fifth-generation (5G) networks, which are widely used in today's wireless communication. Second, it features a delay line, a necessary condition for the time delay RCs. Lastly, it has rich nonlinear dynamics inside.

Chapter 2: Optoelectronic Oscillator and Its Mathematical Modeling

The optoelectronic oscillator (OEO) is an autonomous, nonlinear, and dissipative optoelectronic device that features a delay line within its closed loop. Originally invented to generate ultra-pure microwave signals with extremely low phase noise, it has now become a versatile and essential device widely used in various modern communication system applications. As mentioned in [7], the earliest prototype of a modern OEO can be traced back to 1968 [78], where an optical path was concatenated back into the electrical path to stabilize mode locking between the longitudinal modes of a He-Ne laser. The first optoelectronic oscillator in the modern sense was a breakthrough achieved in 1994 at the Jet Propulsion Laboratory (JLT) at California Institute of Technology [79,80], offering a novel method for generating low phase-noise microwave signals by combining photonic and electronic components in a closed feedback loop. The idea was to introduce the optical fiber as a high-quality factor (high-Q) component to store energy. Compared to the traditional oscillators that often exploit RF components, for example, a Fabry-Perot resonator, to store energy, the idea of incorporating an optical fiber for an oscillator was revolutionary. Due to its ultra-low loss and material stability of optical fibers at room temperature, it became possible to generate ultrapure microwave with extremely low phase noise.

In the past 30 years, due to its rich nonlinear dynamics and widespread applications in microwave photonics, the number of publications related to OEO has grown at a steady rate. It has already become one of the most studied optoelectronic devices [81]. Its applications have evolved beyond ultra-pure microwave generation and now include, but are not limited to, chaos synchronization and communications, OEO-based sensing technologies, and reservoir computing. For fundamental research, the OEO also serves as a good exemplar for studying the dynamics of time-delayed systems, which the following equation can characterize:

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{x_t}) \tag{2.1}$$

where $\mathbf{x}$ is N-dimensional flow with $\mathbf{x} = (x_1, x_2, \cdots, x_l)$ and $\mathbf{x_T}$ is a time delayed part of $\mathbf{x}$ with $\mathbf{x_T} = \mathbf{x}(t - T)$.

Compared to the regular differential equations, it is not only required to have initial conditions with a finite set of values at $t = 0$ as regular differential equations to solve Eq. 2.1, but it is also mandatory to have the initial conditions for the time delayed part defined in the interval $[-T, 0]$ with a specific function $g(t)$. Thus, a set of infinite values have to be known to fully characterize a solution for such systems. This is in full agreement with mathematical theories that time delayed differential equations (DDEs) are inherently infinite-dimensional [82]. It is known that timescales and delay time inside the OEO can have 10 orders of magnitude difference, making it an ideal system to investigate the interactions between infinite-dimensionality and nonlinearity, the complex dynamical states such as slow-fast oscillation, chaos, and even superchaos.

2.1 Mathematical Modeling of a Time Delayed System

To date, aside from the original OEO implementation proposed by Yao and Maleki [80], which utilized a narrowband filter and a single optical feedback loop, numerous variants and implementations have been proposed and studied. However, the essence of all these variants remains the same and can still be characterized by a generic representation, as shown in Figure 2.1: an OEO is an optoelectronic system that features both an electrical and an optical path with a time delay line. Energy inside the system flows through the loop, converting from electrical form to optical form via an electrical-to-optical (E/O) converter, and then back from optical to electrical form through an optical-to-electrical (O/E) converter, finally feeding back into the input via the electrical path. With a loop structure featuring both an optical

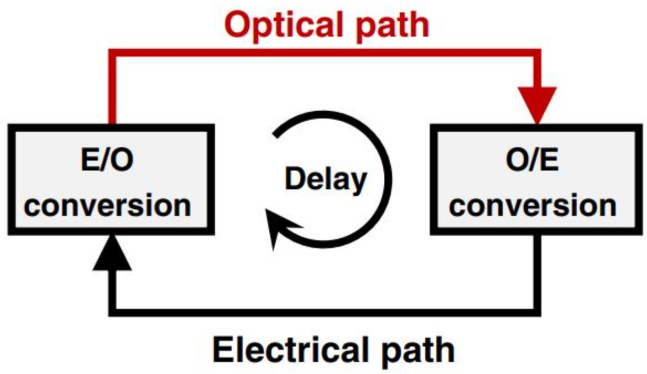


Figure 2.1: A general representation of an optoelectronic oscillator with simplified structures. Energy flows through the loop and converts from electrical form to optical form with an electrical-optical (E/O) conversion, and then converts back from optical form to the electrical form and feeds back to the input with an electrical path following the optical-electrical (E-O) converter. This figure is reproduced from ref. [7].

path and an electrical path, modern implementations of OEOs can simultaneously generate and output an optical signal ($\sim$ 50 THz to 500 THz) and a microwave signal ($\sim$ 30 kHz to 200 GHz) by inserting an optical coupler and an RF coupler into the optical and electrical paths, respectively. It is also worth noting that, despite their structural differences, most of these implementations can be conveniently realized using off-the-shelf components. For example, optical-to-electrical conversion can be easily achieved using a phase and polarization insensitive device (e.g., a photodiode), which makes the experiment more robust and controllable by relaxing the requirement for light coherence.

In general, the study of mathematical modeling for OEOs is closely related to a type of time-delayed equations originally proposed by Ikeda [83]. The Ikeda delay differential equation (Ikeda-DDE) has the following form:

$$\phi + \tau\dot{\phi} == A^2\{1 + 2B\sin[\phi(t-T)] + C\} \tag{2.2}$$

This equation was originally used to describe the dynamics and distribution of the laser field in a four-mirror ring optical cavity when laser light was propagating freely in a small nonlinear medium. Inside the equation, ϕ represents the optical phase shift due to the interference of the ring feedback in the nonlinear

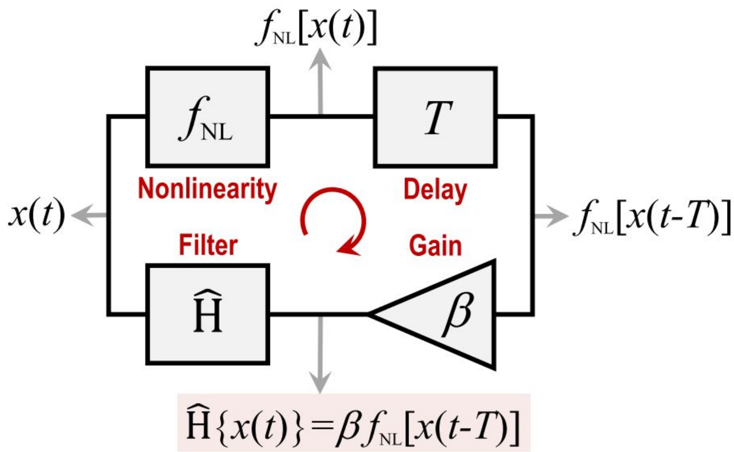


Figure 2.2: An Ikeda-like OEO can be characterized by four key components: a nonlinear function f_{NL}, a linear gain β, a spectrum filter $\hat{H}$, and a time delay T. This figure is reproduced from ref. [7].

medium. τ is the response time of the low pass filter inside the feedback loop. T counts the total delay inside the loop. Eq. 2.2 represents a class of optical systems introduced by Bonifacio, *et al.* [84, 85]. One of the major impacts of this system is that it illustrates that a simple model with nonlinearity and time delay can evolve to complex dynamical states that include chaos. Moreover, the study of these equations provides insights into a particular type of OEOs that we will discuss in the following section - the Ikeda-like OEOs.

2.2 Mathematical Model of Ikeda-like OEOs

As shown in Figure. 2.2, a typical optoelectronic oscillator (OEO) can be further divided into four parts: a nonlinear function f_{NL}, a linear gain β, a spectrum filter $\hat{H}$, and a time delay T. To study its behavior, equations similar to Eq. 2.2 can be of great importance.

Assuming that a scalar dynamical variable $x(t)$ is used to characterize the system's dynamics in the time domain with its Fourier transform counterpart written as $x(t) \xrightarrow{\mathscr{F}} X(i\omega)$, the equation that governs the behavior of an Ikeda-like OEO in the frequency domain can be express as follows:

$$H^{-1}(iw)X(i\omega) = \beta F_{NL}(i\omega)e^{-i\omega T} \tag{2.3}$$

where $F_{NL}(i\omega)$, and $H(i\omega)$ are the Fourier transform of f_{NL}, and $\hat{H}$, respectively. If the filter is passive (a condition that often holds for most implementations), the filter can be characterized by an impulse function $h(t)$ in the time domain, which is the inverse Fourier transform of $\hat{H}(i\omega)$, with $h(t) = 0$ for $t < 0$, and $\int_0^{+\infty} |h(s)| ds < +\infty$. Thus, performing an inverse Fourier transform on both sides of Eq. 2.3 yields an equation in the time domain that is suitable for all types of Ikeda-like OEOs:

$$\hat{H}\{x(t)\} = \beta f_{NL}(x(t-T)) \tag{2.4}$$

where $\hat{H}\{x(t)\}$ is a linear integro-differential operator with its output equal to the input: $\hat{H}(x_{out}) = x_{in}$. By definition, Eq. 2.4 is an Ikeda-like equation. Inside the equation, the four key components - a nonlinear function f_{NL}, a linear gain β, a spectrum filter $\hat{H}$, and a time delay T inside the OEO loop together determine the dynamical behaviors of an OEO.

2.3 Mathematical Model of Low-pass Filter and Broad Bandpass Filter OEOs

Currently, there are many variants of OEO implementations existing in the literature. For example, increasing the number of loops within an OEO has been proven to be an effective method for improving output stability and reducing phase noise. Thus, in Ref. [86], a dual-loop OEO has been proposed and examined. Following this trend, theories and experiments have been extended to multi-loop OEOs that include a triple-loop OEO in Ref. [87]. However, these structures often require more optical and electrical components and often involve more complex mechanisms that are beyond the scope of our OEO-based RC. Thus, in this thesis, we only consider the OEO structure that has a single loop. Figure. 2.3 shows the

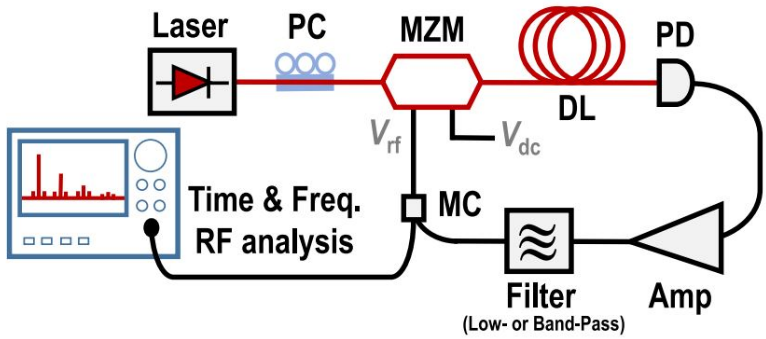


Figure 2.3: The OEO implementations that involve an external Mach-Zehnder Modulator (MZM). PC: Polarization Controller; DL: Delay Line; AMP: Amplifier; MC: Microwave Coupler. This figure is reproduced from Ref. [7].

configuration of single-loop OEOs that involve an external Mach-Zehnder Modulator (MZM).

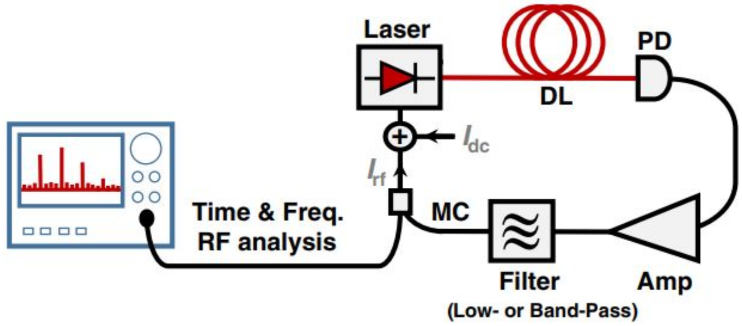


Figure 2.4: The OEO implementation that uses direct modulation from the laser. PC: Polarization Controller; DL: Delay Line; AMP: Amplifier; MC: Microwave Coupler. This figure is reproduced from Ref. [7].

Different from the OEOs that originally proposed to generate ultrapure microwave in Ref. [80], this configuration of OEO is known as the wideband OEO - as it features a linear low-pass RF filter or a Bandpass RF filter. Inside the OEO, the MZM brings the nonlinearity into the loop. Compared to this implementation, there is a similar but simplified wideband OEO implementation of OEO that takes advantage of directly modulating a laser as shown in Figure. 2.4. Both of these implementations of OEOs can be characterized as broadband OEOs that are governed by similar equations.

First, let's consider the low-pass filter case and derive the equations. Since inside the equation, only the first order of the low-pass filter with a high cut-off frequency should be considered, we can easily replace the $\hat{H}\{x(t)\}$ as:

$$\hat{H}\{x\} = x + \tau\dot{x} \tag{2.5}$$

where $\tau = 1/2\pi f_H$ is the time constant introduced by the filter. Combining Eq. 2.4 and Eq. 2.5, we can easily obtain a generic form of equations that rule both these types of OEOs with a low-pass filter:

$$x + \tau\dot{x} = \beta f_{NL}(x(t - T)) \tag{2.6}$$

This equation can be further simplified as a first-order differential equation as follows:

$$\dot{x} = -\frac{1}{\tau}x + \frac{\beta}{\tau}f_{NL}(x_T) \tag{2.7}$$

where x_T is a time-delayed counterpart of the x and the f_{NL} is a nonlinear function solely determined by the modulator (an external modulator or directly modulating the laser).

Then, let's consider the case of a broadband bandpass filter. One of the most straightforward ways to derive this equation is by modeling the bandpass filter as a combination of a low-pass filter and a high-pass filter, with a high cut-off frequency f_H and a low cut-off frequency f_L. In the Fourier domain, the introduction of these two linear filters modifies the operator $\hat{H}$, which can be mapped back into the time domain as an integro-differential equation. Recall the Eq. 2.4 and the four key components in the configuration of OEOs, the only change is to replace a low-pass filter with a bandpass filter. Thus, it can be easily concluded that the new OEO equation can be derived by replacing the nonlinear operator $\hat{H}$ in Eq. 2.5 with:

$$\hat{H}\{x\} = (1 + \frac{\tau}{\theta})x + \tau\dot{x} + \frac{1}{\theta}\int_{t_0}^{t} x(s)ds \tag{2.8}$$

Similarly, if we combined the above equation with Eq. 2.4, an equation can be conveniently derived to describe the broad bandpass filter OEOs as follows:

$$\tau\dot{x} + (1 + \frac{\tau}{\theta})x + \frac{1}{\theta}\int_{t_0}^{t} x(s)ds = \beta f_{NL}(x_T) \tag{2.9}$$

where $\tau = 1/2\pi f_H$ and $\theta = 1/2\pi f_L$. In general, for a broadband OEO, it always has $f_L \ll f_H$, leading to $\tau/\theta \to 0$ and $(1 + \frac{\tau}{\theta})x \simeq x$. Thus, Eq. 2.9 can be further simplified as follows:

$$\dot{x} + x + \frac{1}{\theta\tau}\int_{t_0}^{t} x(s)ds = \frac{\beta}{\tau} f_{NL}(x_T) \tag{2.10}$$

It can be seen that Eq. 2.10 has a similar form to Eq. 2.7 - with a nonlinear function f_{NL} solely determined by the modulator.

2.4 Mathematical Model of Narrowband OEOs

A single-loop narrowband OEO can be implemented by simply replacing the low-pass or bandpass filter to a narrowband filter as shown in Figure. 2.3. Since it has almost the identical configuration as a bandpass filter OEO, the steps to derive the equations ruling its behavior is almost exactly the same. The key is to determine the behavior of the filter and study the narrowban OEO under the framework of Eq. 2.4. Different from broad bandpass OEOs inside which the filters can be viewed as two cascaded first-order filters, the filter used to contruct narrowband OEOs should be characterized as a single and resonator filter. If we model this filter by linearly approximating it with a central angular frequency Ω_0 and a bandwidth of $\Delta\Omega$, the nonlinear operator $\hat{H}$ inside Eq. 2.4 can be expressed as:

$$\hat{H}\{x\} = x + \frac{1}{\Delta\Omega}\dot{x} + \frac{\Omega_0^2}{\Delta\Omega}\int_{t_0}^{t} x(s)ds \tag{2.11}$$

Following the same steps in Section. 2.3, the equation that rules narrowband OEO can be written as:

$$x + \frac{1}{\Delta\Omega}\dot{x} + \frac{\Omega_0^2}{\Delta\Omega}\int_{t_0}^{t} x(s)ds = \beta f_{NL}(x_T) \tag{2.12}$$

where the f_{NL} is determined by the MZM.

Interestingly, due to the large ratio between the fastest and slowest dynamical timescales in the above equation ($\Omega_0^2/\Delta\Omega \sim 10^7$), it is actually hard to directly study the dynamics of narrowband OEO by numerically simulating Eq. 2.12. Instead, an slowly varying envelop equation with a smaller timescale difference introduce by Ref. [88] becomes foundations for time-domain analysis of such system and our study of narrowband OEO-based RCs. Thus, we re-derive a few key steps here following [88]. Due to the narrowband filter residing in the OEO loop, only microwave that within its band could pass through. Thus, the microwave variable $x(t)$ can be written in complex form as follows:

$$x(t) = \frac{1}{2}\mathcal{A}(t)e^{i\Omega_0 t} + \frac{1}{2}\mathcal{A}^*(t)e^{-i\Omega_0 t}, \tag{2.13}$$

where $\mathcal{A} = |\mathcal{A}|e^{i\phi}$ is the complex envelop, and ϕ is phase of the envelop. Considering that fact that envelop is varying slowly relative to the central frequency of the filter, we have:

$$|\dot{\mathcal{A}}| \ll \Omega_0|\mathcal{A}| \tag{2.14}$$

In the case for narrowband OEO with an external MZM, the nonlinear function can be written as follows:

$$f_{NL}(x_T) = \cos^2(x_T + \phi) \tag{2.15}$$

Taking advantage of the Jacobi-Anger expansion:

$$e^{iz\cos\alpha} = \sum_{n=-\infty}^{+\infty} i^n J_n(z) e^{in\alpha}, \tag{2.16}$$

where J_n is the n-th order of first-kind Bessel function. Given that $\cos^2 a = [2\cos(2a) + 1]/2$ and $x(t)$ is nearly sinusoidal around Ω_0, Fourier spectrum of $\cos^2(x_T + \phi)$ will be a series of frequency components distributed around harmonics of Ω_0. With the narrowband filter centered near Ω_0 in the OEO loop, a equation can be derived by combining Eq. 2.12, Eq. 2.13 and Eq. 2.15:

$$x(t) + \tau\frac{dx}{dt} + \frac{1}{\theta}\int_{t_0}^{t} x(s)ds = -\beta\sin 2\phi \times J_1(2|\mathcal{A}(t-T))\cos(\phi(t-T)), \tag{2.17}$$

where $\tau = 1/\Delta\Omega$ and $\theta = \frac{\Delta\Omega}{\Omega_0^2}$. To cancel to integral term $\frac{1}{\theta}\int_{t_0}^{t} x(s)ds$ in above equation, we can use an intermediate variable $u(t)$:

$$u(t) = \int_{t_0}^{t} x(s)ds. \tag{2.18}$$

Assuming that:

$$u(t) = \frac{1}{2}\mathcal{B}(t)e^{i\Omega_0 t} + \frac{1}{2}\mathcal{B}^*(t)e^{-i\Omega_0 t}. \tag{2.19}$$

If we insert Eq. 2.19 into Eq. 2.17, and assuming that the slowly-varying condition also holds for $\mathcal{B}$ with $|\ddot{\mathcal{B}}| \ll \Omega_0|\dot{\mathcal{B}}| \ll \Omega_0^2|\mathcal{B}|$, following equation can be obtained:

$$i\Omega_0\dot{\mathcal{B}}(t) + i\mu\Omega_0\mathcal{B} = -\mu\gamma J_1(2\Omega_0|\mathcal{B}_T|)e^{-i\Omega_0 T}e^{i\psi_T}, \tag{2.20}$$

where $\psi_T = \mathcal{A}_T/|\mathcal{A}_T|$.

It is worth noting that $\mathcal{A} \simeq i\Omega_0\mathcal{B}$. Inserting this condition into Eq. 2.20, an equation in terms of the complex envelop $\mathcal{A}$ of the microwave variable $x(t)$ can be derived as:

$$\dot{\mathcal{A}}(t) = -\mu\mathcal{A} - 2\mu\gamma J_{c_1}(2|\mathcal{A}(t-T)|)\mathcal{A}(t-T)e^{-i\sigma} \tag{2.21}$$

where $\sigma = \Omega_0 T$, and J_{c_1} is the Bessel-Cardinal function with a definition of $J_{c_1}(x) = J_1(x)/x$.

2.5 Conclusion

In this chapter, we derived the time-delayed differential equations that govern the behaviors of both narrowband OEO and broadband OEO in the same framework. For numerical simulation, we also derived the envelop equation for narrowband OEO.

Chapter 3: Classification of IQ-Modulated Signals based on Reservoir Computing with Narrowband Optoelectronic Oscillators

In general, the OEOs used for reservoir computing are wideband and are processing analog signals in the baseband. However, their hardware architecture is inherently inadequate to directly process radiotelecom or radar signals, which are modulated carriers. On the other hand, the high-Q OEOs that have been developed for ultra-low phase noise microwave generation have the adequate hardware architecture to process such multi-GHz modulated signals, but they have never been investigated as possible reservoir computing platforms. In this chapter, we show that these high-Q OEOs are indeed suitable for reservoir computing with modulated carriers. The dataset (DeepSig RadioML) is composed with 11 analog and digital formats of IQ-modulated radio signals (BPSK, QAM64, WBFM, etc.), and the task of the high-Q OEO reservoir computer is to recognize and classify them. We show that with a simpler architecture, a smaller training set, fewer nodes and fewer layers than their neural network counterparts, high-Q OEO-based reservoir computers perform this classification task with an accuracy better than the state-of-the-art, for a wide range of parameters. We also investigate in detail the effects of reducing the size of the training sets on the classification performance.

3.1 Machine Learning for Modulation Classification

The impact of machine learning is expected to be transformational in wireless communications networks, which are submitted to a (genuinely) exponential increase of traffic, en route to reach a zettabyte per year within 3 years. Therefore, the radiofrequency (RF) spectrum is becoming increasingly congested with signals sharing the same frequency band. Processing this amount of information is an unprecedented and difficult challenge, which could be in part addressed by machine learning [89].

Effective adoption of ML for wireless networks is nevertheless facing several challenges. The first one is that it is quite common for ML approaches to be solely based on software, that is, the input signal needs to be converted to a computer-compatible format before ML-software processing. Along the same line, in most instances, the machine learning platform requires transduction or conversion at both the input and output stages. A second challenge is that in the spectral domain, wireless signals are generally narrowly centered around a carrier frequency – up to 70 GHz for the next generation of wireless networks. An optimal ML platform should be able to process the modulated signal, without demodulation to the baseband. A third challenge is that high throughput wireless networks need to be interconnected with optical fiber networks both at the backhaul and fronthaul level, since optical fiber is the optimal technological solution to ensure high-bandwidth connection between the network nodes. Consequently, from the hardware perspective, ML solutions should ideally have the capability to handle both microwave and lightwave signals, in order to interact seamlessly with both types of signals in these heterogeneous wireless-fibered networks.

Reservoir computing (RC) based on optoelectronic oscillators is a machine learning approach that can address the three aforementioned challenges. OEOs have a hybrid architecture that concatenates a laser-driven optical fiber line and an electronic branch, and they have been thoroughly investigated from both the fundamental and applied points for view (see review article [7] for a survey of OEO science and technology).

A few years ago, it was shown that wideband OEOs have the capability to perform machine learning based on reservoir computing (or RC) with RF and/or optical input signals [90,91]. Subsequently, wideband OEOs have been proven to be a robust RC-compatible platform by numerous studies, achieving state-of-the-art performances with several benchmarks such as spoken digit recognition, nonlinear time-series prediction, packet header recognition, signal radar forecasting, and even cancer data classification (see for example refs. [1, 90–100]). This analog computing has also investigated for all-optical systems as well (see for example refs. [101–116]). This computational power is partially rooted in the fact that RC only features 3 layers (input, reservoir and output, as explained in detail in refs. [72, 73, 117]), and requires to train only the output layer while the reservoir (or network) layer is fixed: this distinctive property results in a drastic reduction of processing latency, ideal for large throughput.

An important property of OEOs for wireless networks is their versatility, in the sense that they can process input information either in the microwave domain (RF signals) or lightwave domain (optical fiber data, etc.). Noteworthy is the fact that OEOs can operate virtually at any frequency up to 100 GHz, and have the intrinsic capability handle high-bandwidth signals and perform ultrafast processing, such as in the experiment reported in ref. [96] where OEO-based RC permitted to achieve a record of one million words per second.

In this chapter, we will extend the concept of OEO-based RC for the classification of RF modulated signals. Wideband OEOs are not suitable for this task, as they essentially operate in the baseband. However, narrowband OEOs, initially developed for the purpose of ultralow phase noise microwave generation, are capable of processing RF modulated signals despite the fact that they have never been considered for the purpose of RC or ML in general. Our main objective here is to therefore to show, for the first time to the best of our knowledge, that narrowband OEOs can be used for RC in wireless communications. Another objective is to show that even with a reduced training set size, the reservoir computer can still provide a high level of performance – sometimes better than the state-of-the-art – with a faster computing speed in comparison to other ML alternatives. We consider a benchmark where an RF signal is IQ-modulated using 11 different formats, and with different levels of signal-to-noise ratio. The task of the RC computer will be to classify these incoming signals with the highest accuracy but smaller training set sizes.

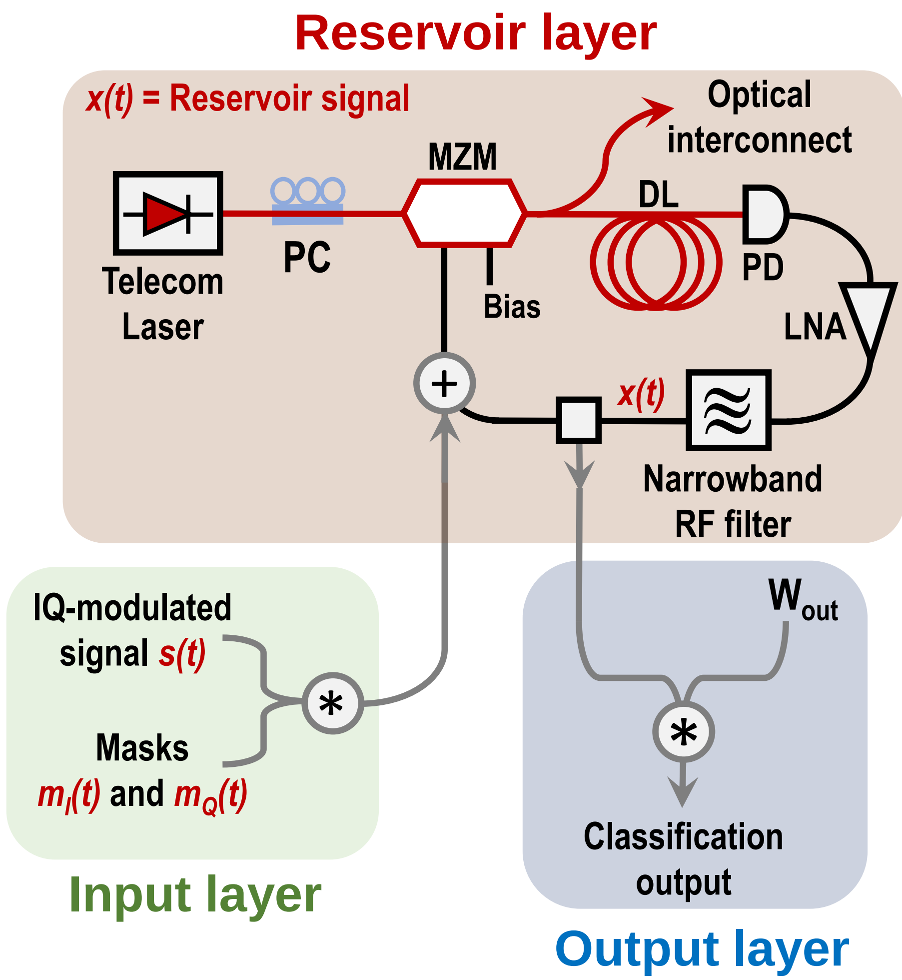


Figure 3.1: Schematical representation of the reservoir computer based on a narrowband OEO. DL: Delay line; LNA: Low-noise amplifier. MZM: Mach-Zehnder modulator; PD: Photodiode.

3.2 The Narrowband OEO System

The system under study is a narrowband single-loop OEO with an input port in the RF branch for the signal to be processed. The major difference of narrowband OEO and wideband OEO is that narrowband OEO often features a much higher quality factor Q (typically at several hundred) while the wideband OEO often has a quality factor $Q \approx 1$. A schematic representation of this oscillator is displayed in Figure. 3.1, and it consists of the following elements in a closed feedback loop: (i) A wideband integrated Mach-Zehnder modulator, seeded by a continuous-wave (CW) semiconductor laser of optical power P;

the modulator is characterized by a DC half-wave voltage $V_{\pi_{DC}}$ and an RF half-wave voltage $V_{\pi_{RF}}$, which are both of the order of few volts. (ii) The modulated laser beam at the output of the modulator is launched into an optical fiber delay line of length $L = 4$ km and group velocity index $n_{\rm g}$, that creates a time delay of $T = n_{\rm g}L/c = 20$ μs, where c is the velocity of light in vacuum; This delay line also induces microwave ring-cavity modes with a free-spectral range (FSR) equal to $\Omega_T/2\pi = 1/T = 50$ kHz. Note that for our purpose, a physical optical fiber delay line is not an absolute necessity; the time delay can be implemented using a digital signal processing (DSP) of field programmable gate array (FPGA) board. (iii) The time-delayed optical signal is converted into an electrical signal by a photodiode with a conversion factor S, in units of V/W. (iv) The electrical signal is then sent to narrowband microwave RF filter, that has the role to select the carrier frequency $\Omega_0/2\pi$ and the allowed bandwidth $\Delta\Omega/2\pi$ around it. This central frequency can be anywere in the 1-100 GHz range, while the bandwidth is typically few tens of MHz wide. In all cases, the RF quality factor $Q_{\rm RF} = \Omega_0/\Delta\Omega$ of the narrowband RF filter is generally in excess of 100. (v) The filtered electrical is subsequently amplified with a gain factor G, before being connected back as a voltage $V(t)$ to the RF electrode of the Mach-Zehnder modulator. All optical and electrical losses are accounted for with a single attenuation factor κ. (vi) The IQ-modulated RF signal to be processed is eventually inserted in the feedback loop via a microwave coupler, placed just before the RF electrode of the modulator.

It is important to emphasize here the essential differences between narrowband and wideband OEOs. As discussed in the Ch. 2, both systems are Ikeda-like oscillators characterized by the very same four elements (linear gain, filter, delay and nonlinearity) and feature similar bandwidth-delay products ($\Delta\Omega \times T \sim 100$), but the difference in parameters $\Delta\Omega$ and T leads to substantially different dynamical properties.

Indeed, wideband OEOs feature short time delay and a large bandwidth filter, while conversely, narrowband OEO feature a long time delay and a narrow bandwidth filter. The wideband OEOs generally output spectrally flat and broad signals, including hyperchaotic ones. As a consequence, it is intuitively

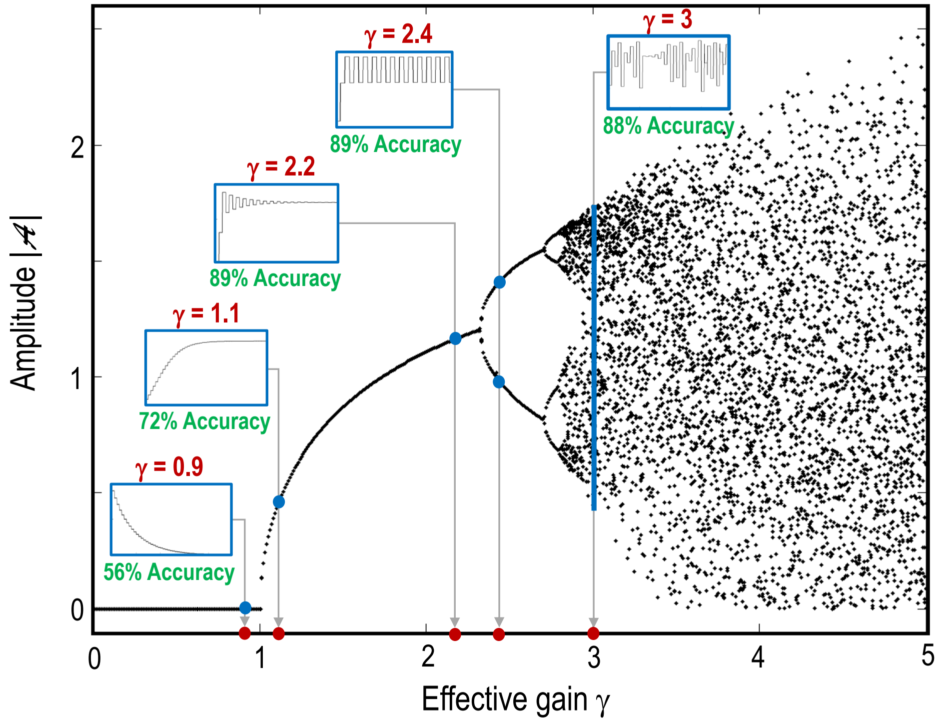


Figure 3.2: Bifurcation diagram of the narrowband OEO as the effective gain γ is increased. The system undergoes a Hopf bifurcation at $\gamma = 1$, and a Neimark-Sacker bifurcation at $\gamma = 2.31$. The timetraces in blue inserts display the transient dynamics of the non-driven OEO to the final state for 5 gains values of interest. The best classification accuracy achieved with a training set of $230 \times 11 = 2530$ samples and $\mathrm{SNR} = 18$ dB is indicated as well for these 5 gain values (ρ is scanned from 1.5 to 2.5). The highest classification accuracy in our RC simulations (88.94% for $\gamma = 2.2$ and $\rho = 1.63$) is obtained with only 1 trained layer: It outperforms the state-of-the-art accuracy of 87% which is achieved using neural networks with 5 trained layers and ~ 4800 (i.e twice as many) training samples [8]. Interestingly, our work shows that optimal performance is achieved around the Neimark-Sacker bifurcation, and *not* around the primary Hopf bifurcation like in RC using broadband OEOs.

logical to use them to process baseband electrical signals through RC. However, narrowband OEOs were intended to generate a single tone RF signals, so that it is less intuitive to foresee how they can be useful for RC. Nevertheless, as it will be shown in this article, RF modulated signals have a spectral spread that falls within these narrowband OEO bandwidths, and can therefore be processed.

3.3 System and Model

3.3.1 The Narrowband OEO Model

There are several models proposed to study to dynamics of OEOs [80, 88, 118, 119]. Following [88, 118], we utilize the delay-differential model to study dynamics of narrowband OEO-based RC, which has already discussed in Sec. 2.4, where a nonlinear time-delay differential equation Eq. 2.12 to describe its behavior has been developed. The integro-differential equation that narrowband OEO obeys can be further written under the simplified and dimensionless form as follows:

$$x + \tau\frac{dx}{dt} + \frac{1}{\theta}\int_{t_0}^{t} x(\xi)d\xi = \beta\cos^2[x_T + \phi]\,, \tag{3.1}$$

where $x(t) = \pi V(t)/2V_{\pi_{RF}}$ is the dimensionless voltage at the input of the Mach-Zehnder modulator, $x_T \equiv x(t-T)$ is its time-delayed counterpart, $\beta = \pi\kappa GSP/2V_{\pi_{RF}}$ is the normalized feedback gain, $\phi = \pi V_B/2V_{\pi_{DC}}$ is the Mach-Zehnder off-set phase. In wideband OEOs, the parameters $\tau = 1/2\pi f_{\mathrm{H}}$ and $\theta = 1/2\pi f_{\mathrm{L}}$ are the typical timescales corresponding to the low- and high-frequency cutoffs f_{L} and f_{H} of two cascaded first-order filters. However, in narrowband OEOs, they are written instead as $\tau = 1/\Delta\Omega$ and $\theta = \Delta\Omega/\Omega_0^2$ and jointly describe the characteristic timescales associated with the resonant bandpass filter.

However, in Sec. 2.4, it has been shown that Eq. (3.1) is not suitable for the study of the narrowband OEO because of the very large splitting between the various timescales [88]. This problem is circumvented by acknowledging that the narrowband filter forces the oscillations to be quasi-sinusoidal with frequency Ω_0, so that their complex envelope carries all the relevant dynamical information. Therefore, we can write the solution of Eq. (3.1) as signal with a carrier of frequency Ω_0, which is slowly modulated by a complex-valued amplitude $\mathcal{A}(t)$ following

$$x(t) = \frac{1}{2}\mathcal{A}(t)e^{i\Omega_0 t} + \frac{1}{2}\mathcal{A}^*(t)e^{-i\Omega_0 t} \tag{3.2}$$

where the envelope $\mathcal{A}(t) = A(t)e^{i\psi(t)}$ varies slowly relatively to the carrier of frequency Ω_0 – a condition that mathematically translates to $|\dot{\mathcal{A}}(t)| \ll \Omega_0 |\mathcal{A}(t)|$.

Following [88, 118, 120], we have already shown in 2.11 that the envelop of the modulated signals follows Eq. 2.21. For demonstration purposes, we rewrite this equation here as Eq. 3.3:

$$\dot{\mathcal{A}} = -\mu\mathcal{A} - 2\mu\gamma e^{-i\sigma}\,\mathrm{Jc}_1[2|\mathcal{A}_T|]\mathcal{A}_T\,, \tag{3.3}$$

where $\mu = \Delta\Omega/2$ is the half-bandwidth of the radio-frequency filter, $\gamma = \beta \sin 2\phi$ is the effective gain of the feedback loop, $\sigma = \Omega_0 T$ is the round-trip phase shift of the microwave, Jc_1 is the Bessel-cardinal function defined as $\mathrm{Jc}_1(x) = \mathrm{J}_1(x)/x$ [88, 118, 120]., and $\mathcal{A}_T \equiv \mathcal{A}(t-T)$. The phase condition of the oscillation is $\gamma e^{-i\sigma} = -1$, and it is convenient to set $\sigma = \pi$ (modulo 2π), so that $\gamma \geq 0$.

Equation (3.3) typically has four dynamical regimes as a function of γ [118]. For $\gamma < 1$, the system does not oscillate and the trivial fixed point is stable ($\mathcal{A} = 0$). At $\gamma = 1$, the system undergoes a primary Hopf bifurcation and the system oscillates with a constant complex-valued envelope , i.e. $x(t)$ is a sinusoidal signal corresponding to a stable limit-cycle (this regime is the one that is useful for ultrapure microwave generation). As the gain is increased and reaches the critical value $\gamma_{\mathrm{cr}} = \{2\mathrm{Jc}_1[\mathrm{J}_0^{-1}(0)]\}^{-1} \simeq 2.31$, the oscillator undergoes a secondary Hopf bifurcation, also known as a Neimark-Sacker bifurcation, where the amplitude of the sinusoidal signal becomes periodically modulated $x(t)$ – in other words, envelope undergoes a primary Hopf bifucation and the limit-cycle bifurcates to a torus. This (multi-) periodic regime bifurcates to chaos when the gain is increased beyond a value around 2.8.

3.3.2 The Narrowband OEO Model with IQ-modulated Signal Injection

In order to perform RC on the narrowband OEO for RF fingerprinting, we need to inject the I/Q modulated RF signals into the electrical feedback loop of OEO. Before delving into working principles of narrowband OEO-based RC, we need to derive an dynamical model for the I/Q signal driven narrowband

OEO first.

The modulated I/Q signal could be expressed as:

$$s(t) = I(t)\cos\Omega_0 t + Q(t)\sin\Omega_0 t \tag{3.4}$$

where "I" stands for the *in-phase* component and "Q" stands for the *quadrature* component of the baseband signal. With I/Q modulation scheme, they are encoded onto a RF carrier Ω_0 that could be up a few hundreds GHz. Now, taking this I/Q modulated signals back into the narrowband OEO equations Eq. (3.1), we can get following equation for an OEO with I/Q modulated signal injection:

$$x + \tau\frac{dx}{dt} + \frac{1}{\theta}\int_{t_0}^{t} x(\xi)d\xi = \beta\cos^2[x_T + \rho s(t) + \phi]\,, \tag{3.5}$$

where ρ is a dimensionless amplifying factor of the signal $s(t)$ in the feedback loop. Due to similar reasons for Eq. (3.1), Equation (3.5) is also not suitable for study the I/Q signals driven narrowband OEO. Thus, an envelope equation for the I/Q signal driven narrowband OEO needs to be derived. First, we can rewrite the modulated term inside the nonlinear $\cos^2$ as

$$x_T + \rho s(t) = q(t)\cos\Omega_0 t + r(t)\sin\Omega_0 t \tag{3.6}$$

where

$$q(t) = \rho I(t) + A_T\cos(\psi_T - \sigma) \tag{3.7}$$

$$r(t) = \rho Q(t) - A_T\sin(\psi_T - \sigma) \tag{3.8}$$

with

$$A_T \equiv |\mathcal{A}(t-T)| \;\text{ and }\; \psi_T \equiv \arg \mathcal{A}(t-T)\,. \tag{3.9}$$

Then, we can rewrite the signal injected into RF electrode of the Mach-Zehnder modulator as

$$x_T + \rho s(t) = \frac{1}{2}\mathcal{Z}(t)e^{i\Omega_0 t} + \frac{1}{2}\mathcal{Z}^*(t)e^{-i\Omega_0 t} \tag{3.10}$$

with $\mathcal{Z}(t)$ being its complex-valued envelope explicitly defined through

$$|\mathcal{Z}(t)| = \sqrt{q^2(t) + r^2(t)} \tag{3.11}$$

$$\arg \mathcal{Z}(t) = \arctan \frac{q(t)}{r(t)} - \frac{\pi}{2}\,. \tag{3.12}$$

Following the same procedure as the one leading to Eq. (3.3), we now obtain

$$x(t) + \tau\frac{dx}{dt} + \frac{1}{\theta}\int_{t_0}^{t} x(s)ds = -\beta \sin 2\phi J_1(2|\mathcal{Z}|)\cos(\Omega_o t + \arg \mathcal{Z}) \tag{3.13}$$

Finally, we can get

$$\dot{\mathcal{A}} = -\mu\mathcal{A} - 2\mu\gamma \mathrm{Jc}_1[2|\mathcal{Z}|]\mathcal{Z} \tag{3.14}$$

which is the envelope equation that governs the dynamics of the narrowband OEO envelope when driven by an I/Q modulated signal.

3.4 IQ-modulation Classification Dataset

The data set we have chosen to evaluate the performance of narrowband OEOs is made of radiofrequency signals with IQ modulation (“I” for *in-phase*, and “Q” for *quadrature*). The waveforms $I(t)$ and $Q(t)$ are

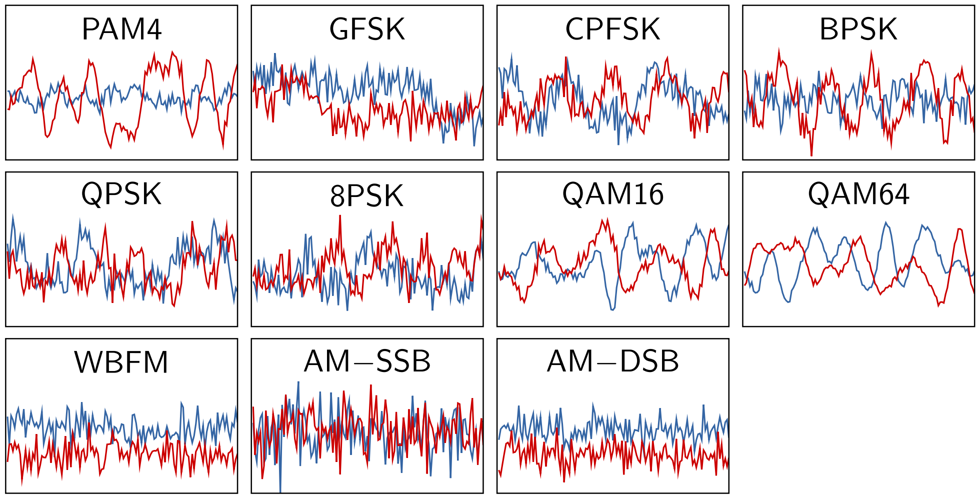


Figure 3.3: Examples of normalized timetraces of the I (blue) and Q (red) signals for our 11 modulation formats, when the $\mathrm{SNR} = 0$ dB. Each timetrace has 128 sample points.

baseband signals, and the IQ-modulation scheme encodes them onto a radio carrier whose frequency can be as high as few hundred GHz. This modulation scheme is popular in modern architectures of wireless networks because it allows to transmit high-capacity modulation formats (constellations with more bits per symbol), and therefore permits to optimize the bandwidth of the transmission link.

As the airspace becomes crowded by many RF signals sharing the same frequency band allocation, it becomes critical to have the capacity to identify, and the large throughput of data favors hardware analog recognition in order to avoid latency and network congestion. Typical tests involve for example emitted modulation formats or RF fingerprinting hardware imperfections. In this chapter, we are considering the former case, which is particularly important in many instances such as for dynamic spectrum access protocols, which require the optimization of spectrum allocation. The efficiency of this optimization procedure greatly relies on the efficient radio recognition, and while prior works are mostly based on feature engineering from temporal signals, analog computing techniques are promising for directly processing these signals in the physical layer.

From the perspective of machine learning, the radio modulation recognition task could be characterized as a classification problem that requires correct prediction of the modulation scheme directly from the radio signals. In order to train and test our proposed narrowband OEO-based RC, we will exploit the IQ radiofrequency modulation dataset proposed by O'Shea, Corgan and Clancy (Deepsig dataset RADIOML 2016.04C – see refs. [8, 121]). This dataset consists of 11 modulation signals generated by GNU radio, as displayed in Figure. 3.3. Eight of these modulations are digital modulations, namely BPSK (binary phase-shift keying), BFSK (binary frequency-shift keying), QPSK (quadrature phase-shift keying), CPFSK (continuous phase-frequency-shift Keying), PAM4 (4-level pulse-amplitude modulation), 8PSK (8 phase-shift keying), 16QAM (16 quadrature amplitude modulation), and 64QAM (64 quadrature amplitude modulation), while the remaining three are analog modulations schemes, namely WB-FM (wideband frequency modulation), AM-SSB (single-sideband amplitude modulation), and AM-DSB (double-sideband amplitude modulation).

Each example in the dataset contains 128 sampled points for both in-phase and quadrature components. For each type of modulation, signals are collected with different signal-to-noise ratio (SNR) ranging from -20 dB to 18 dB with a step of 2 dB. Hence, the in-phase and quadrature components of one example could be expressed by two vectors of length 128 written as $\{X_I^i(1), X_I^i(2), \ldots, X_I^i(128)\}$ and $\{X_Q^i(1), X_Q^i(2), \ldots, X_Q^i(128)\}$, where subscripts I and Q indicates the signal is in-phase or quadrature component and the superscript indicates that it corresponds to the i^{th} example in the dataset.

3.5 Conventional Reservoir Computer

Reservoir computing is a particular subset of supervised machine learning algorithm that arose as a one of the leading paradigms in the area of neuromorphic computing. This approach, which is conceptually close to the ones of echo state network (ESN) or a liquid-state machine (LSM), is a neuromorphic framework that can perform efficient and real-time processing for temporal information [72, 73, 117, 122]). Unlike the usual neural networks where the full connectivity matrix is updated via a backpropagation algorithm (for example), reservoir computing is based on the optimization of "read-out" coefficients only. The reservoir computer therefore encodes the information signal to be processed into the highly dimensional dynamical system.

There are two main advantages justifying our choice of reservoir computing for our project: *(i)* Only the output layer needs to be trained, while the reservoir remains static. This feature greatly simplifies the determination of the optimal weights. Once the optimal read-out coefficients are obtained, they are used to project the nonlinear transient dynamics of the reservoir onto an identifiable target state. *(ii)* The second advantage is that this simpler architecture permits faster training and testing than in conventional neural networks. Since, high-throughput inherently imposes a stringent constraint on computation time, the possibility provided by reservoir computing to expedite the training appears as a critical advantage for our purpose.

The signals at the input layer are fed into the RC, such that the reservoir and output states $\mathbf{x}$ and $\mathbf{y}$ are sequentially updated at discrete time-steps $n \in \mathbb{N}$ following

$$\mathbf{x}(n+1) = \mathbf{f}_{\mathrm{NL}}[\mathbf{W}_{\mathrm{in}} \cdot \mathbf{u}(n) + \mathbf{W} \cdot \mathbf{x}(n)] \tag{3.15}$$

$$\mathbf{y}(n+1) = \mathbf{W}_{\mathrm{out}} \cdot \mathbf{x}(n+1)\,, \tag{3.16}$$

where $\mathbf{u}(n)$ is the input signal while $\mathbf{f}_{\mathrm{NL}}$ is a nonlinear vector function. The optimal readout connectivity matrix $\mathbf{W}_{\mathrm{out}}$ is obtained via ridge regression through

$$\mathbf{W}_{\mathrm{out}}^{\mathrm{opt}} = \mathbf{M}_{\mathbf{y}} \cdot \mathbf{M}_{\mathbf{x}}^{\mathsf{T}} \cdot [\mathbf{M}_{\mathbf{x}} \cdot \mathbf{M}_{\mathbf{x}}^{\mathsf{T}} + \lambda \cdot \mathbf{I}]^{-1}\,, \tag{3.17}$$

where $\mathbf{M}_{\mathbf{x}}$ is a matrix that suitably concatenates the internal state $\mathbf{x}$ generated with some specific training input vectors $\mathbf{u}$, while $\mathbf{M}_{\mathbf{y}}$ is the target (or *teaching*) matrix yielding the desired classification outcome, $\mathbf{I}$ is the identity matrix and $\lambda \ll 1$ is a small regularization factor for to avoid the overfitting of the inversion problem. The prediction task becomes a regression problem if the teaching matrix is high-dimensional

Table 3.1: Classification accuracy as a function of gain and SNR (with $\rho = 1.61$ and 660 training samples)

SNR	-6 dB	0 dB	6 dB	8 dB	10 dB	18 dB
$\gamma = 0.9$	19%	44%	63%	65%	50%	32%
$\gamma = 1.1$	19%	46%	67%	67%	60%	59%
$\gamma = 1.6$	22%	47%	71%	73%	68%	79%
$\gamma = 2.2$	22%	47%	76%	77%	69%	82%
$\gamma = 2.4$	22%	47%	76%	77%	69%	83%
$\gamma = 3.0$	22%	48%	73%	78%	70%	75%

(quasi-continuous approximation), and a classification problem when it is low-dimensional. Despite its simplicity, RC provides a powerful analog computing algorithm with performances comparable to or even better than RNNs on several benchmarks [123]. Interested readers could refer to [124] for a more general and detailed review.

3.5.1 Narrowband OEO-based RC implementation

Theoretically, any time-delay system is infinite-dimensional, which indicates that any input data with finite dimensions could be mapped into arbitrary high dimension space within a single delay line. This property is exploited by OEO-based RC to excite a large number of virtual nodes in temporal domain as traditional RC does in spatial domain. In the narrowband OEO-based RC, each sample point of the example is mapped into a N dimensional space after multiplying with a mask signal with N random values. The expansion of the original sample at the temporal domain is known as time-division multiplexing. The reservoir layer is obtained by dividing the delay line into a total of N intervals using time-division multiplexing and the values of each $\Delta T = T/N$ inside the delay line are sampled to represent the value of N virtual nodes. Unlike digital RC that needs an explicit implementation of the non-linear function, the non-linearity of RC based on time-delayed system is introduced implicitly directly from nonlinear property of the system itself. The first successful implementation of RC with a time-delayed system was introduced in 2012 using an electronic Mackey-Glass oscillator [6]. We refer the reader to that reference, as well as to [90, 91] for an in-depth presentation of RC using time-delayed systems.

With a simple structure of three computational layers inside, wideband OEO-based RC has achieved state-of-the-art performance on several benchmarks: spoken-digit recognition, time series prediction, radar signal regression at super-fast speed [90–96]. In ref. [96], a wideband OEO-based RC achieved a record of one million words per second speed for spoken-digit recognition. The surprisingly simple yet powerful structure of RC, combining with ultra-fast OEO device, leads to its unbeatable low processing latency together with a faster training procedure and makes it a potentially ideal ML platform for IQ-modulated signal related tasks. However, the wideband OEO-based RC mentioned above is not suitable for this task as it operates at baseband.

To deal with IQ-modulated signals, an ideal ML platform should be able to directly process the modulated signals. Meanwhile, the narrowband OEO, which was originally invented to generate ultra-low phase noise microwave is capable of processing such RF modulated signal. Therefore, in this paper, our main objective is to demonstrate the suitability of a narrowband OEO-based RC to perform RF fingerprinting. Another objective is to show that the OEO-based RC can still provide high level performance with greatly reduced training set size. Similar to its wideband counterpart, narrowband OEO-based RC also requires the time-division multiplexing procedures to map input signal into reservoir layer. The core difference is that narrowband OEO-based RC needs to calculate the complex-valued envelope equation Eq. (3.14) while the wideband is driven by the original Ikeda-like integro-differential equation Eq. (3.1). In order simulate the envelope equation Eq. (3.14), the first step is to determine $r(t)$ and $q(t)$ from the input data. As shown in next section, the original dataset containing RF fingerprinting signals are discrete sampled in-phase and quadrature components. In order to simulate the continuous time-delayed envelope equation, another two continuous time series are generated by holding each sampled data point in-phase and quadrature component $X_I^i(k)$ and $X_Q^i(k)$ for system delay time T, where the superscript i indicates it's the i^{th} example in the training set, subscript I and Q indicates it's the in-phase or quadrature components of the I/Q modulated signals, and k indicates is the k^{th} data point of the i^{th} training example. In this way, two continuous time series are generated from discrete sample points of one training example which satisfies:

$$I(t) \equiv X_I^i(k), \quad (k-1)T \leq t \leq kT \tag{3.18}$$

$$Q(t) \equiv X_Q^i(k), \quad (k-1)T \leq t \leq kT\,. \tag{3.19}$$

Comparing with most digital implementation of RC with computer-hardware that maps all the signals of one training example into high dimensional space in parallel, OEO-based RC projects input data sequentially into temporal domain with a fixed separation between virtual nodes. Exploiting the concept of time-division multiplexing, the narrowband OEO-based RC only feed one data point into RC layer during one delay time T. In each duration of delay time T, a data point is coupled to the virtual nodes after multiplication with the input mask. To stimulate a total number of N equally distant virtual nodes in the RC layer, the weight values of the mask must have the same separation time between two consecutive weight with $\Delta T = T/N$ as two virtual nodes (for our case $N = 400$). As masks contain fixed weights for all the data point, it's produced as a continuous signals with period equals to the delay-time T. Following the approach proposed in [91], the weights of masks are uniformly sample from six values. As the input data contains two streams corresponding to in-phase and quadrature components, in this article, we introduce two masks $m_I(t)$ and $m_Q(t)$ to couple them into RC layer following:

$$J_I(t) = I(t) \cdot m_I(t) \ \text{ and } \ J_Q(t) = Q(t) \cdot m_Q(t)\,. \tag{3.20}$$

In order to train the RC, the values of envelope $\mathcal{A}(t)$ from the middle point of each interval ΔT are sampled to represent the reservoir states for each virtual nodes. This leads to the output of RC layer value corresponding to j^{th} virtual node of i^{th} example of input data as:

$$\mathsf{A}_{ij} = \mathcal{A}[i \cdot T + (N - j) \cdot \Delta T]\,. \tag{3.21}$$

Concatenating all the virtual nodes' states in the RC layer of the whole training set for A_{ij} gives the suitably designed matrix $\mathbf{A}$, which is then used to optimize the read-out layer. The process of obtaining the optimal read-out layer weights $\mathbf{W}^{\mathrm{RO}}$ leads to an optimization problem of searching for:

$$\mathbf{W}^{\mathrm{RO}}_{\mathrm{opt}} = \underset{\mathbf{W}^{\mathrm{RO}}}{\mathrm{argmin}} \sum_{i=1}^{N_{\mathrm{Train}}} \|\mathbf{W}^{\mathrm{RO}}\mathbf{A}_{i\cdot} - \mathbf{Y}_i\|^2 + \lambda\|\mathbf{W}^{\mathrm{RO}}\|^2 \tag{3.22}$$

where $\mathbf{A}_{i\cdot}$ is the RC states of i^{th} training example, and $\mathbf{Y}_i$ is the corresponding label. This optimization problem could be solved by ridge regression algorithm [123]:

$$\mathbf{W}^{\mathrm{RO}}_{\mathrm{opt}} = \mathbf{A}(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{Y} \tag{3.23}$$

where $\mathbf{Y}$ stands for the labels of training set and λ is a small regulation term to prevent overfitting.

3.6 Simulation Results and Discussion

We consider the following parameters for our numerical simulations: The delay time is $T = 20\ \mu$s, the narrowband filter bandwidth is $\Delta\Omega/2\pi = 20$ MHz, the roundtrip phase shift is $\sigma = \Omega_0 T = \pi$, the insertion gain for the input data is $\rho = 5$, and the regularization term for ridge regression algorithm is $\lambda = 0.001$.

Here we use the dataset mentioned in Section 3.4 as benchmarks to test our narrow band OEO-based RC. Unless otherwise stated, we extract 60 training samples and 60 test samples for every type of modulation and for each SNR, which leads to a training set and a test set with equal size of 660. Before

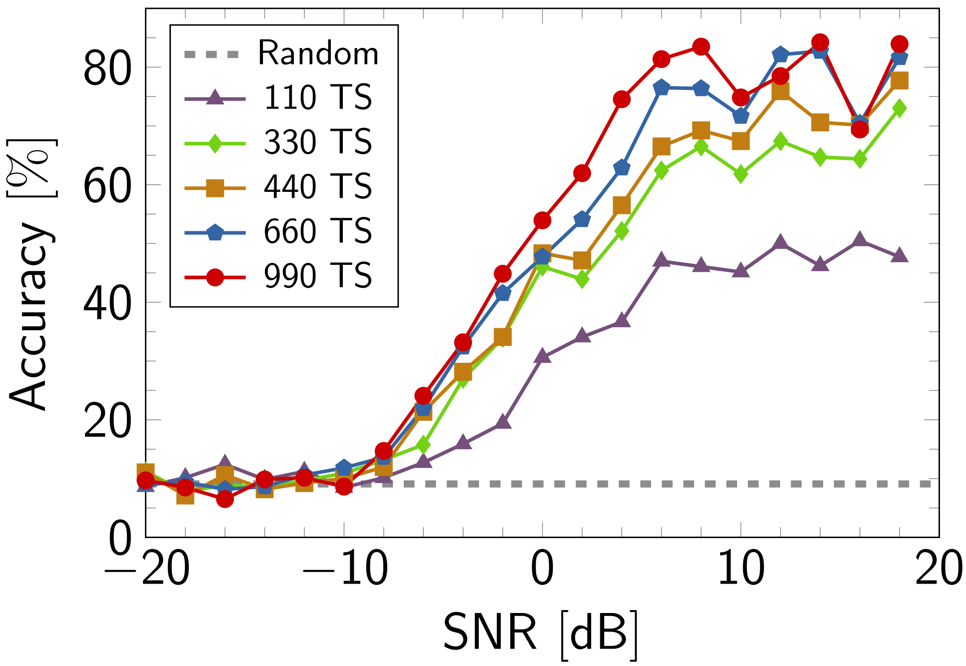


Figure 3.4: Variation of the classification accuracy of the narrowband reservoir computer with the SNR, for $\gamma = 2.2$. We have considered various set sizes for the training samples (TS), ranging from ~ 100 to ~ 1000 (note that the typical set size for training in neuronal-network-based ML is ~ 5000 – see [8]). The purple line with triangle symbols corresponds to the accuracy when the training set has 110 patterns (10 for each of the 11 modulation formats); The green line with diamond symbols corresponds to the accuracy when the training set involves 220 patterns (20 for each of the 11 modulation formats); etc. The dashed line indicates the performance of a random classifier ($1/11 \simeq 9.09\%$) For very low SNR (i.e., very high levels of noise), the reservoir computer does not performs better than a random classifier. However, as the SNR is increased, the reservoir computer accuracy significantly improves and achieves a performance approaching 90%.

multiplying with input mask, we normalized these signals first so that each training example's power sums to one, following:

$$X^{i\prime}_{I,Q}[j] = \sum_{k=1}^{128} \frac{X^i_{I,Q}[j]}{\sqrt{X^i_I[k]^2 + X^i_Q[k]^2}} \tag{3.24}$$

As a classification task, the performance metrics is the classification accuracy, which is defined as the percentage of correctly predicted modulation samples with regard to all predicted samples. We also introduce the confusion matrix as another pertinent indicator that helps understand where are the sources of error in our RC classifier.

As explained in Figure. 3.2, our best accuracy performance is 88.94% (with a training set size of

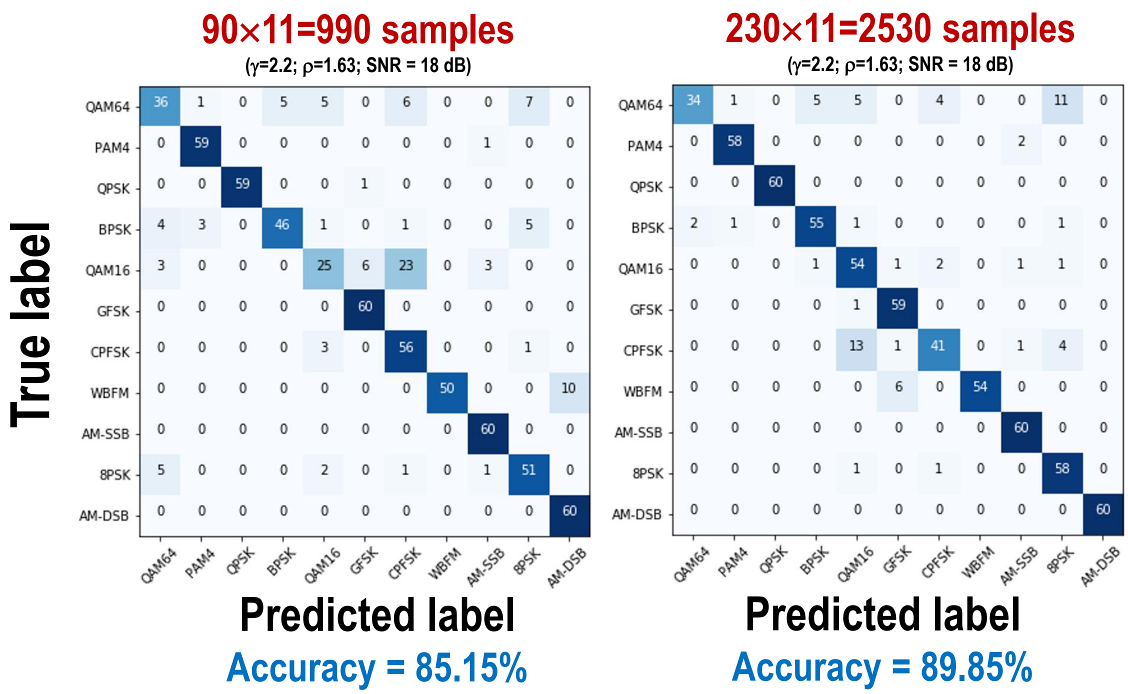


Figure 3.5: Confusion matrices for $\gamma = 2.2$, and for two different training sample sizes. The testing set size is contains 660 items, so that the sum for each line must is always 60 (the number of testing samples). One can observe that most errors mainly come from a confusion between WBFM and AM-DSB analog formats, and from a confusion between the digital formats QAM64, QAM16, and the PSK (8, Q, and B).

2530 samples and $\rho = 1.62$), which is better than the state-of-the-art at 87% (with a training set size of 4800 samples). The OEO-based RC is already demonstrating its potential with this performance, but in this work, one of our key objectives is to investigate how the RC performs with even *smaller* training set sizes.

Our results are presented in Table 3.1, which displays the accuracy of our classifier as a function of the gain γ and the signal-to-noise ratio, when the training set size is reduced to 660 samples. As shown in Figure. 3.2, the parameter γ is directly related to the dynamical regime of narrowband OEO. We can see from the table, the RC performs poorest when $\gamma = 0.9$, which corresponds to the oscillator being in the trivial equilibrium when not driven by the IQ-modulated signal [$x(t) \equiv 0$ and $\mathcal{A}(t) \equiv 0$]. This result seems to be in contradiction with the conventional wisdom gained from RC with broadband OEOs, where normalized gain values below unity provide optimal performance. In the case of narrowband OEOs, the dynamical regime where RC is the most efficient is located around the secondary bifurcation, i.e., the one where the envelope (and not the carrier) undergoes a Hopf bifurcation. This situation actually makes sense when one realizes that the I and Q signals only contribute to the envelope dynamics, and not to the one of the carrier: as a consequence, only the Hopf instability related to the envelope might contribute to the RC performance. This is exactly what is observed as the value of the gain is increased: The RC efficiency is ameliorated when γ is in the area surrounding the Neimark-Sacker bifurcation at $\gamma_{\rm cr} = 2.31$. It is indeed unconventional to operate a reservoir computer beyond a fixed point but previous work had already shown that this approach is valid under certain conditions for RNNs [125]. To the best of our knowledge, RC has never been operated in a chaotic regime, but in our case, the performance at $\gamma = 3$ remains comparable to the maximal efficiency of our classifier. This is explained by the fact that chaos in for this gain parameter remains low-dimensional, in many points similar to the laminar chaos recently discussed in the literature [126]. However, our simulations show that if the gain is increased any further, the system becomes more chaotic and the performance steadily decreases: For an $\mathrm{SNR} = 0$ dB, the accuracy drops to $\simeq 50\%$ for $\gamma = 3.5$, and to $\sim 40\%$ for $\gamma = 4.2$.

As shown in Figure. 3.4, we have also investigated the effect of SNR on the RC performance, for the fixed gain value $\gamma = 2.2$, but different training size sets sizes. We can first note that as expected, the RC efficiency is poor when the SNR is low (typically, $\mathrm{SNR} < -10$ dB). In that case, the reservoir computer basically operates as a random classifier, allocating the 11 labels following a uniform probability distribution with accuracy $1/11 \simeq 9.09\%$. As the noise is reduced, the reservoir computer progressively begins to outperform the random classifier, and it is noteworthy that even when the noise is 4 times more powerful than the signal ($\mathrm{SNR} = -6$ dB), RC can still distinguish features and perform much better than a random guess. Higher SNR values lead to a significant increase of performance, but quite counter-intuitively, higher SNR does not systematically lead to better accuracy. These anomalies (also present in Table 3.1) are be explained by the variability of the dataset.

The confusion matrices of Figure. 3.5 give us an insight in relation to the origin of misclassifications. We have reported such matrices for two sets of parameters yielding good performance. It appears that there two confusion clusters generate most of the errors. The first one involves the analog formats WBFM and AM-DSB, and it can be seen that when the reservoir computer classifies one of them with high accuracy, it performs very poorly for the other. This is consistent with the results from [8], where the classifications is performed with neural networks. The second confusion cluster involves the 5 digital formats QAM64, QAM16, 8PSK, QPSK and BPSK, which are sometimes shuffled by the classifier.

3.7 Conclusion

In this chapter, we have investigated the performance of high-Q OEO-based RC for the classification of IQ-modulated signals. We have shown that they have an performance better than the state-of-the-art, while featuring a simpler architecture, which leads to a faster computing capability. We have also explored the effect of training set size reduction, and shown using about ten times less training samples that the state-of-the-art neural networks leads to only a marginal reduction of accuracy for our RC. This robustness property gives to OEO-based RC an edge for real-time computing or ML with nonstationary datasets, and positions them as promising ML systems for the processing of radio signals. Our results also indicate that since OEOs can process both lightwave and microwave signals, they are suitable as real-time analog computing nodes in the physical layer of the next generation of networks, where wireless and fiber communications links are interconnected.

Chapter 4: RF Fingerprinting based on Reservoir Computing using Narrowband Optoelectronic Oscillators

Radiofrequency (RF) fingerprinting refers to a range of technologies that recognize transmitters by their intrinsic hardware-level characteristics. These characteristics are often introduced during the fabrication process and form a unique fingerprint of the transmitter that is very hard to counterfeit. RF fingerprinting often serves as a security measure at the physical-layer of communication networks against potentials attacks. In recent years, neuromorphic computing techniques such as convolutional neural networks (CNNs) have been explored as classifiers for RF fingerprinting. However, in radiofrequency communication networks, the transmitted signals are I/Q modulated on multi-GHz carriers while most conventional machine learning algorithms operate at the baseband. Therefore, the I/Q modulated signals have to be demodulated and converted into compatible formats before applying to these platforms – a procedure that that inevitably slows down the processing speed. Moreover, the deep learning technologies often require a large amount of data to train the artificial neural networks (ANNs) while in practice, the available amount of data for a new transmitter is limited. Reservoir computing (RC) provides a relatively simple yet powerful structure that is capable of reaching state-of-the-art performance on several benchmarks. However, traditional digital RC also operates at baseband, which is not suitable for directly processing the I/Q modulated signals. In Ch. 3, we have illustrated that the narrowband OEO-based RC can be used for IQ Modulation classification task. In this chapter, we further extend the narrowband OEO-based RC to the field of RF fingerprinting. We successfully train and test our narrowband OEO-based RC on three publicly available benchmarks, namely the FIT/CorteXlab RF fingerprinting dataset, the ORACLE RF fingerprinting dataset, and the AirID RF fingerprinting dataset. We show that for all three datasets, the narrowband OEO-based RC demonstrates competing accuracy with much less training data comparing to CNNs, and achieves an accuracy as high as 97%.

4.1 Machine Learning for RF Fingerprinting

Wireless communication systems have evolved to the era of 5G, and the next generation is just underway. The unprecedented bandwidth and low latency granted by these technologies have reshaped the communication industry and incubated huge amounts of new applications with a remarkable growth of wireless devices connected to the network. The undergoing Internet of Things (IoT) technology is even more ambitious, as it aims at constructing a network that physically connects nearly everything into cyberspace [127]. The smooth operation of all these devices heavily relies on the successful switching of small formatted units of data known as *packets*. Unlike other layers in the Open Systems Interconnection (OSI) model, signals transmitting at physical layers are not under protection of any cryptographic protocol but only use headers to provide guiding information and prevent transmitter identification error. Inside the IoT systems, the portion of headers might even outweight the payloads. As security is always the top priority and the most fundamental requirement of any communication systems, it becomes critical to design certain transmitter identification technologies at this layer to authenticate prior connected transmitters and prevent misuse or penetration from malicious transmitters [128].

As a type of transmitter identification technology, radiofrequency (RF) fingerprinting refers to a method that recognizes transmitters with their hardware level characteristics. Indeed, because of manufacture imperfections during the fabrication process, even two RF frontends with exactly the same configuration and specification feature slight difference introduced by imperfect components like power amplifiers and local oscillators. Unlike software-level signatures [129], these hardware-level differences form a unique fingerprint of one specific RF transmitter and are very hard to forge [130]. On top of these differences, RF fingerprinting builds up a classification system for the purpose of disambiguating legitimate from malicious transmitters.

The concept of RF fingerprinting is not recent. For example, the idea of extracting features from the transient signals [131, 132] when the transmitters' status changes and using these features for RF fingerprinting had already been proposed. The major difficulty of this type of RF fingerprinting is the requirement of precisely locating the start point of the transient signal from channel noise [132,133]. Later on, researchers switched to methods that directly perform recognition on steady-state signals. Before the era of artificial neural networks (ANNs), machine learning algorithms applied to RF fingerprinting mostly required complex feature engineering to achieve good performance [134–136]. In the last decades, deep neural networks (DNN) have revolutionized the field of pattern recognition and achieved state-of-the-art performance in many tasks like video, image, voice and speech recognition. In the last 5 years, this technique has been introduced into the field of processing RF signals and has gained wide acceptance [89, 137].

Compared to traditional machine learning algorithms, CNN provide an end-to-end approach which captures essential features directly from original signals. However, effective adoption of neural networks (NNs) into processing RF signals still faces several challenges. The first challenge arises from the coordination between hardware and software. In the spectral domain, RF signals are narrowly centered on a carrier frequency – up to 70 GHz – while most implementations of NNs are based on GPU, a piece of computer hardware designed for high-throughput parallel digital baseband signal processing operating at only a few GHz. This means that the modulated RF signals must be demodulated first before being processed into computer compatible format. The demodulation and format conversion inevitably slow down the processing speed. The second challenge is the requirement of large dataset and time exhausting procedures to train NNs. For application scenarios that need frequent updates of transmitter list with newly connected transmitters, it becomes unrealistic to collect a large amount of training examples for unseen transmitters and re-train the whole CNN.

Therefore, an ideal machine learning (ML) platform for RF fingerprinting should have following properties. First, it should be configured fast enough to process RF signals. Ideally, it should be a hardware platform with integration of software. Second, it should be able to directly perform classification on the modulated signals without the need of demodulation. Finally, it should require the smallest possible number of training examples to be collected when being trained for new transmitters.

Unlike other deep neural networks that all the weights in all layer are trained via computational expensive back-propagation algorithms, the RC randomly initializes the weights of all its three layers and only optimizes the read-out layer, leading to a super-fast re-training when receiving new data. Moreover, the reservoir layer maps input data into a higher dimensional space, making the data more linearly separable. A rigorous theoretical proof has shown that RC satisfies three critical properties, namely separation property, fading-memory property and approximation property [73], which makes it an universal approximation to arbitrary function [138].

Comparing with deep neural network, there are several advantages for the choice RC for RF fingerprinting. *(i)* Only the read-out layer needs to be trained while the other two layers are randomly initialized [73]. This greatly simplifies the optimization for training. For RF fingerprinting with frequent update of transmitters, the training time is greatly reduced when new transmitters are connected. Unlike DNN optimization which often inevitably gets stuck at local optima, the global optimal solution of RC optimization could be easily achieved. Once optimal read-out layer is obtained, it could be immediately applied to project the I/Q modulated signals into an identifiable target state. As shall be shown in numerical simulation part later, the training set size needed for RC is also reduced significantly, which further reduces time for sampling data. *(ii)* The simpler structure of RC permits faster training and testing and allows its faster deployment, where high-throughput and real-time processing of RF modulated signals is in demand. *(iii)* The RF signals are timeseries that could be regarded as a sequential data. Unlike CNNs that are most often used as a tool

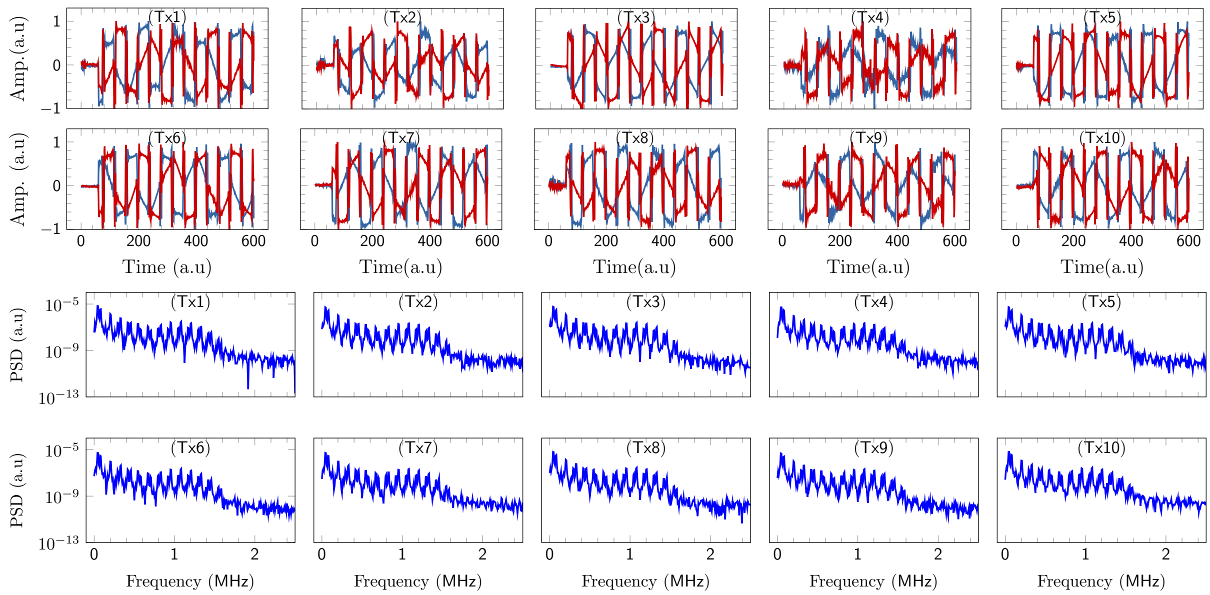


Figure 4.1: Top rows: Ten transmitters (from Tx1 to Tx10) in the FIT/CorteXlab are set to transmitting static bit sequence with QPSK modulation. This figure includes a full packet of data (600 points) from 10 different transmitters. The orange line and black line are corresponding to the in-phase and in-quadrature components. Bottom rows: The power spectral density (PSD) of 10 different transmitters from FIT/CorteXlab are shown here. The signals across different transmitters have very similar PSD, making it hard even for a human expert to recognize a transmitter merely from the PSD.

to process 2D image data, as a special type of RNNs, RCs are powerful tools with excellent ability to memorize temporal information and process sequential data.

In Ch. 3, we have demonstrated that narrowband OEO-based RC can efficiently process IQ-Modulation Classification tasks. Now, we will further examine the possibility of introducing this platform for RF fingerprinting applications.

4.2 RF Fingerprint Dataset

RF fingerprinting exploits hardware level characteristics to recognize transmitters. The usefulness of this technology lies in two aspects: communication security and spectrum efficiency. Communication security is the most basic and critical requirement for communication systems. Nowadays, the signals transmitting through communication channels are divided into small packets. In the OSI model, due to

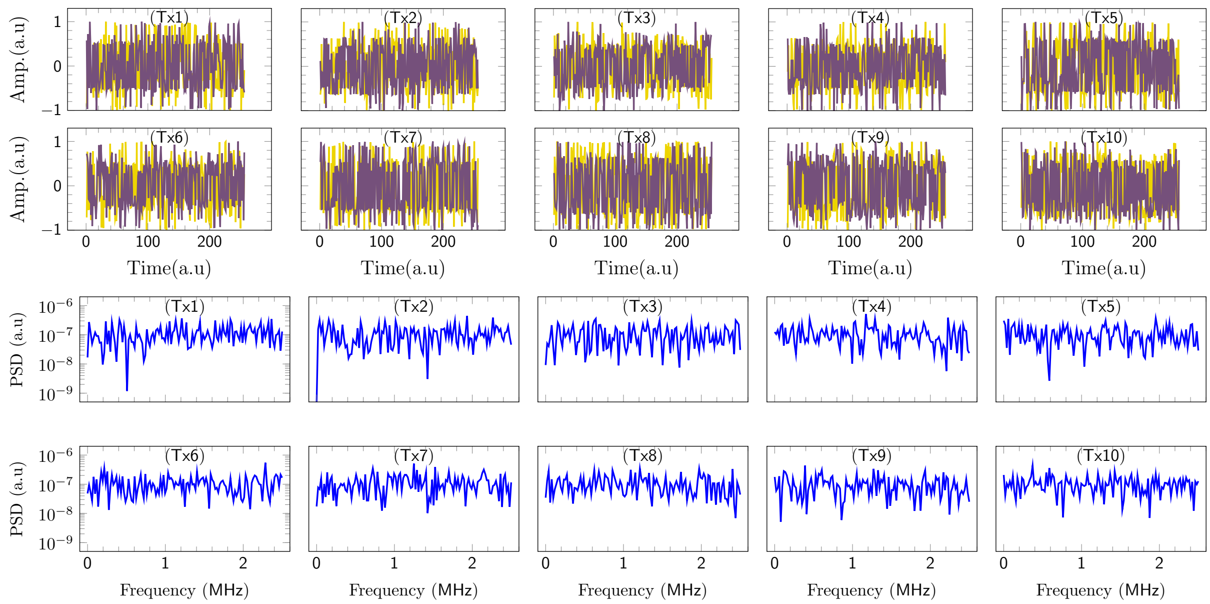


Figure 4.2: Top rows: Ten transmitters (from Tx1 to Tx10) are considered. Over 200 million samples are collected for each transmitter in the ORACLE RF fingerprinting dataset. The samples extracted from each transmitters are pre-processed into training and test set first. Each example in the training set and test set contains 256 data points. This figure shows representative time series from 10 transmitters. The orange line and black line are corresponding to the in-phase and in-quadrature components, respectively. Bottom rows: The transmitters in the ORACLE RF fingerprinting have similar PSD.

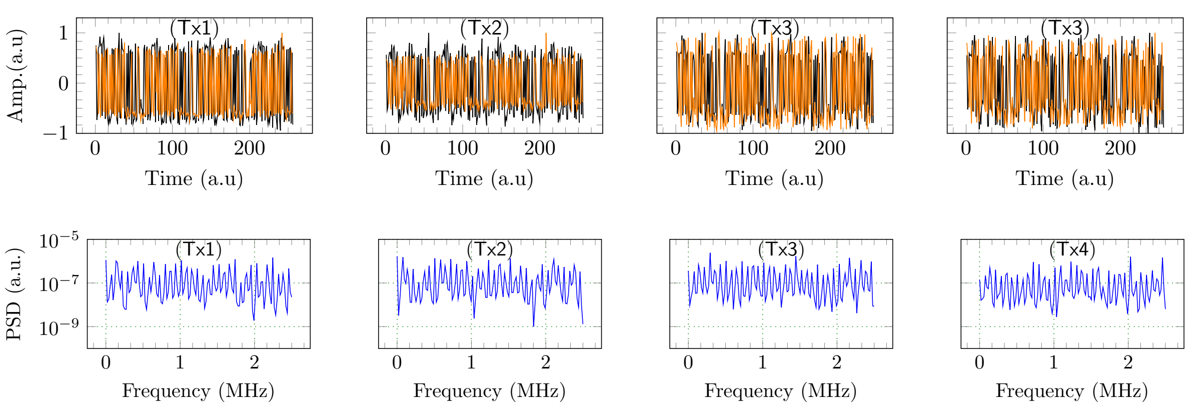


Figure 4.3: Top row: In AirID RF fingerprinting dataset, the transmitters are installed in UAVs. Samples of 4 transmitters inside the AirID are pre-processed in examples each contains 256 data points. This figure shows representative time series from these 4 transmitters. The orange line and black line are corresponding to in-phase and in-quadrature components. Bottow row: he corresponding power spectral density (PSD) of 4 different transmitters from AirID RF fingerprinting dataset.

the bandwidth limit, these small packets transmitting at the physical layer are the only exceptions to the protection of cryptographic protocols. Thus, it becomes urgent to add a physical layer protection that could prevent malicious transmitters. Meanwhile, today's airspace are congested by RF signals within the same frequency band allocation. The identifier of different transmitters are convoyed through small packets known as headers. For applications like IoT, there always exists a huge number of different nodes that are transmitting these packet simultaneously. Inevitably, lots of the packets transmitting through channels are headers. Effective and fast identification of transmitters through their hardware level characteristics could not only prevent the packets from malicious transmitters but also greatly reduce the cost of identifier transmission.

The hardware characteristics that differs one transmitter from the other could be divided into three categories. The first category of difference is the channel effect, which arises from the impact of channels between the transmitter and receiver. The loss caused by path length varies the power at the receiver end and the relative distance between two transmitter and receiver could be used to recognize transmitters. The second type of difference is known as I/Q imbalance that are caused by the power amplifier imperfections. The third kind of difference is the frequency shift at the transmitter end due to local oscillator imperfections.

In this thesis, we will utilize several publicly available RF fingerprint datasets to train and test the narrowband OEO-based reservoir computing (RC) system, demonstrating its broad applicability in RF fingerprinting.

4.2.1 FIT/CorteXlab RF Fingerprinting Dataset

The first RF fingerpriting dataset we used to train and test our narrowband RC is FIT/CorteXlab [139]. This dataset contains I/Q signals generated from 21 different transmitters under the guidance of GNU Radio. Each of the transmitters is generated by a National Instruments USRP N2932 software defined radio (SDR) and collected under different scenarios. There are mainly two different scenarios.

1) Type of Transmitting Signals

- *Static packet*: all emitters are sending exactly the same bit sequence of the 802.15.4 preamble with QPSK modulations.

- *Random packet*: all emitters are sending randomly generated bit sequence with same modulation schemes.

- *Noise packet*: all emitters are sending payloads that are randomly generated from a uniform noise source. The difference between noise packet and random packet is that the signals are not undergoing through any modulation schemes.

2) Transmission Complexity

- *Plain*: all emitters are sending signals with the same amplitude. Nothing moves on the transmission path between RF transmitters and receivers.

- *Varying Amplitude*: the amplitude of each transmitter is varying its amplitude to emulate the different path loss of each transmitters without moving them. Inside the experimental room, nothing moves so that the multi-path parameter is still static.

- *Robots*: all emitters are sending payloads with varying amplitudes as previous settings. Besides that, a robot with metallic sheet cover is introduced, and randomly moves inside the room, which leads to a constantly changing reflection to randomize multi-path parameters.

In this chapter, we mainly focus on using I/Q imbalance, the most difficult characteristics for classifiers to learn for the RF fingerprinting. So we extract the training and testing data from the static packet that transmitting with plain amplitude, which allows the RC to recognize transmitters solely by I/Q imbalance.

Each of example in the data set contains a complete small packet sent by a transmitter with both I/Q components ("I" for *in-phase*, and "Q" for *quadrature*). A full packet of one example contains 600 sampled points of the I/Q modulated signals. The total number of data collected for one transmitter contains about 35000 packets.

Figure 4.1 shows representative time series of a packet sent from 10 different transmitters. It could be seen that these signals are similar in amplitude and only under distortion of I/Q imbalance. It can also be seen that that all of these 10 transmitters have similar power spectrum, making it not possible to recognize the source by frequency shiftting.

4.2.2 ORACLE RF Fingerprinting Dataset

The ORACLE RF fingerprinting dataset contains I/Q modulated signals generated from 16 different USRP X310 transmitters. In this dataset, all 16 USRP X310 transmitters are set to generate bit-similar signals under the guidance of it IEEE 802.1a standards. All the data frames transmitted through transmitters contain a random payload with the same address fields. For each of the transmitter, over 20 million of data points are sampled. The whole dataset contains signals transmitted at different distance between the transmitter and receiver. In the paper, we select a subset of the dataset that stress the I/Q imbalance.

Figure 4.2 shows representative time series and power spectrum density (PSD) of 10 transmitters. It can be seen that these 10 transmitters are transmitting very similar signals and have similar PSD.

4.2.3 AirID RF Fingerprinting Dataset

Comparing with previous two datasets that focus on using I/Q imbalance for RF fingerprinting, the AirID dataset has the most complicated channel conditions. It contains raw I/Q signals that transit over-the-air of 4 USRP B200mini radios, which are installed as transmitters on DJI M100 unmanned aerial vehicle (UAV). As the UAVs are hovering with varying distance from the receiver, the channel condition varies greatly. At the transmitter side, specially designed I/Q imbalance is purposely introduced such that the RF fingerprint of transmitter could be exploited for identification of UAVs.

Figure 4.3 displays some representative time series and power spectrum density (PSD) of 4 transmitters. It can be seen that these 4 transmitters are transmitting very similar signals and have similar PSD.

4.3 Simulation Settings and Results

Unless otherwise specified, for our numerical simulations the narrow OEO parameters are set as follows : The delay time is $T = 20$ μs, the narrowband filter bandwidth is $\Delta\Omega/2\pi = 20$ MHz, the roundtrip phase shift is $\sigma = \Omega_0 T = \pi$, and the regularization term for ridge regression algorithm is $\lambda = 0.0001$. For the RC, the total number of virtual nodes inside the RC layer is $N = 400$. If not otherwise specified, all results are obtained with $\gamma = 2.4$.

4.3.1 Numerical Results on FIT/CorteXlab Dataset

4.3.1.1 Latency and accuracy trade-off

Latency is defined as the processing time of one example. It is corresponding to the narrowband OEO-based RC's speed. In this section, we discuss over the trade-off between data truncate length, a parameter that directly proportional to latency, and the narrowband OEO-based RC's performance.

In principle, to excite virtual nodes in the narrowband OEO-based RC system, each data sample in an example is fed into the delay line and being held for the narrowband OEO's delay time T. Thus, the latency for processing one example is directly proportional the total number of samples points one example contains. As real-time processing speed is highly desired for RF fingerprinting applications, it is critical to reduce the total number of samples of each example while maintain a high level of accuracy. It is worth mentioning that truncating the examples into shorter length also has the benefits of reducing the total number of samples need to be collected of each transmitter, which is critical to application scenarios that frequent update of transmitters and re-train the RC is in demand.

Following settings for training and testing a CNN in [139], we start our simulations by training and testing the narrowband OEO-based RC without any truncation of example's length at all. Each of these full length examples has 600 samples inside. To reduce latency, the example size is then truncated into

128 and 256, which simply means that only the first 128 or 256 samples in each example will be kept to train and test the narrowband OEO-based RC.

In this simulation, the training set size is limited to 200 per each transmitter. For full length examples, the total number of samples needed to train the narrowband OEO-based RC is only 0.51% of that being needed to train a CNN in [139]. When further truncating the example size to 256 or 128, the total number of samples needed to train to RC is further reduced to 0.21% or 0.11% needed by training a CNN.

As shown in Figure 4.4, in the task of classification of 10 different transmitters, the best accuracy achieved is 93.17% for full packet size (with $\rho = 10.61$) and 92.67% for truncate length of 256 (with $\rho = 11.61$). This simulation result indicates the latency for classification of a transmitter could be reduced to less half being needed using full packet data (5.12 ms v.s. 12 ms) by truncating the data length to 256 with a slight trade-off of 0.5% accuracy. When example size is further reduced to 128, the time needed to process one example could be reduced to as low as 2.56 ms while maintaining the accuracy at a high level of 88.33% (with $\rho = 10.61$). The reduction of accuracy when data length is truncating into 128 is not to our surprise since the first 128 samples contain lots of steady state signals.

4.3.1.2 Role of injection amplifying factor ρ and stability analysis

The injection amplifying factor ρ is a parameter corresponding to the amplification of the I/Q injected signals, which has significant impact on the narrowband OEO-based RC's performance. To further explore the RC's stability under different injection amplifying factor, we scanned through different ρ to train and test the narrowband OEO-based RC with three data truncate length. As shown in Figure 4.4, for the same example truncate length, the RC's performance is sensitive to the injection amplifying factor ρ with an accuracy variation of 7%, making ρ an important hyper-parameter that should be fine-tuned during the training process. Moreover, it can been seen that the accuracy of data truncating at 128 with

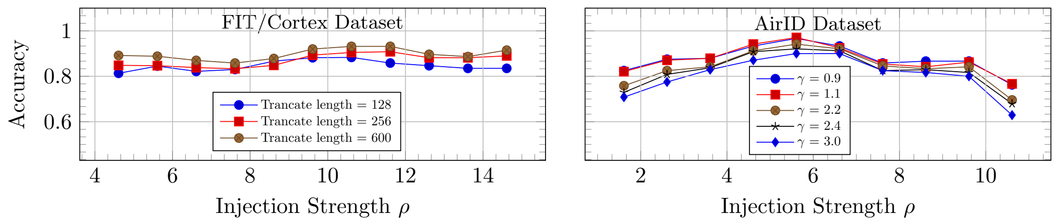


Figure 4.4: The left figure shows the scanning across different injection amplifying factor ρ for different data truncate length on the FIT/Cortex dataset. This classification task involves 10 different transmitters. For full data length (600 points), the highest classification accuracy 93% at $\rho = 10.61$. With a small trade-off of performance, truncating the data length to 256 reduces the narrowband OEO-based RC's processing time to less than half required by using full packet data. The right figure shows a narrowband OEO-based RC that is trained and test on classification of 4 transmitters using AirID dataset with $\gamma = 2.4$, $\rho = 9.61$.

$\rho = 10.61$ is even 2% higher than full packet data at $\rho = 7.61$. It can also be seen that under the same injection amplifying factor ρ, increasing the data length always improves the classification accuracy of a narrowband OEO-based RC. Thus, determining injection amplifying factor ρ becomes a critical step in training a narrowband OEO-based RC system. Finally, for region of the injection amplifying factor $\rho < 2$, the RC's performance degraded to $\sim 10\%$, which could be interpreted as that RC is making random prediction. This can be explained by Eq. 3.6 to Eq. 3.10 that the influence to the system's dynamics of weakly injected I/Q signals is very limited. The weakly injected signals makes it hard to pass the information in the I/Q signals to virtual nodes.

4.3.1.3 Training set size optimization

The benefit of reducing the training set size while maintaining the performance at a certain level is obvious. In application scenarios where new transmitters often connect in and training examples of these transmitters need to be collected for re-training the classifier, real-time processing becomes an impossible task if the classifier requires a training set size at the level of tens of thousands for each transmitter. CNN is known for its demanding requirements for huge training data. It's also known that CNN doesn't perform well on low resource settings when training set size is limited to hundred level [140]. Even with sufficient data, the time-consuming back-propagation algorithms to re-train CNN for new transmitters make it inappropriate for real-time applications. In contrast, as shown in previous section, the narrowband OEO-based RC could still achieve accuracy $\sim 90\%$ for a training set size of 200 even when each training example length is truncated to 256. With the urgent needs of reducing the number of sample points for re-training the narrowband OEO-based RC, we further explore the relationship between the performance and the number of training examples of each transmitter in this section. Figure 4.5 shows a curve of accuracy on different number of training examples. It can be seen that the performance has been improved significantly when the number of training examples for each transmitter increases from 10 to 100 and then saturated at $\sim$ 90% afterwards. This suggests that for a new transmitter, it is sufficient to collect 100 training examples with a truncate length of 256 which further reduce the number of sample points to 0.055% required by CNN. This property of narrowband OEO-based RC could also be especially useful for RF fingerprinting with low-resource settings or pre-selection before high precision classification. Lastly, for FIT/CorteXlab dataset, it can be seen that a further truncation of data length to 128 and limit training set size to 100 reduces number of sample points for training narrowband OEO-based RC to 0.027% of that required by CNN and halves the latency with a sacrifice of 3% accuracy, making it a tempting option for radical optimization on processing speed.

4.3.1.4 Extending to more transmitters

In previous section, we limited our simulations to a task of classification of 10 transmitters. In this section, we investigate the narrowband OEO-based RC's performance when extending the classification task to include different number of transmitters.

As can be seen from the Figure 4.6, the narrowband OEO-based RC maintains a high level of accuracy ($\sim 90\%$) when the number of transmitters increases to 14. But when the classes of transmitters further increases to 21, the accuracy decreases to 72.28%. This can be explained by the very fact that the narrowband OEO-based RC only uses 0.11% sample points to be trained comparing with state-of-the-art

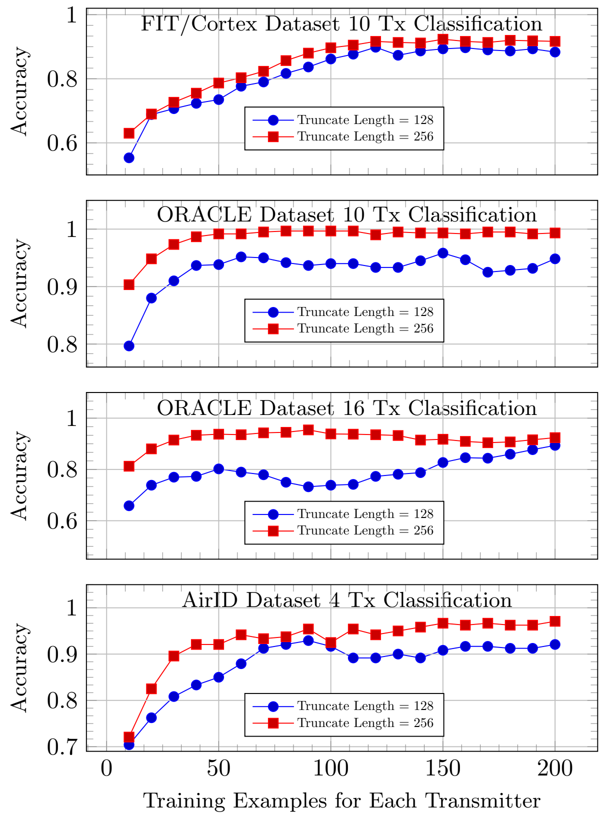


Figure 4.5: For the FIT/Cortex dataset with a classification task that involves 10 different transmitters, the narrowband OEO-based RC's performance increases rapidly for both data truncate length when the number of training example per transmitter is increased from 10 to 120 and saturated afterwards. Narrowband OEO-based RC always has a slight loss of accuracy around 3% on data truncate length 128 but has double processing speed. For ORACLE dataset, the narrowband OEO-based RC is first trained and test on classification of 10 transmitters with $\gamma = 2.4$, $\rho = 9.61$. The narrowband OEO's performance is saturated for both truncate length when number of training examples for each transmitter reaches 50. It can be seen that the narrowband-OEO based RC reach above 99% accuracy with only 50 training examples for truncate length of 256. When further increase the number of different transmitters being classified to 16, the narrowband OEO's performance is saturated at 50 training examples per transmitter for truncate length of 256 but keeps increasing for data truncate length of 128 until training examples per transmitter reaches 200. The overall accuracy for classification all the transmitters in the dataset is still above 95% only requiring 50 training examples. For AirID dataset, the narrowband OEO-based RC is trained and test on classification of 4 transmitters with $\gamma = 2.4$, $\rho = 5.61$. The performance curves follows similar trend when applying narrowband OEO-based RC to AirID dataset. Its performance is saturated after training set size exceeds 100 per transmitter. The best accuracy obtained is $\sim 97\%$ with 200 training examples for data truncate length of 256.

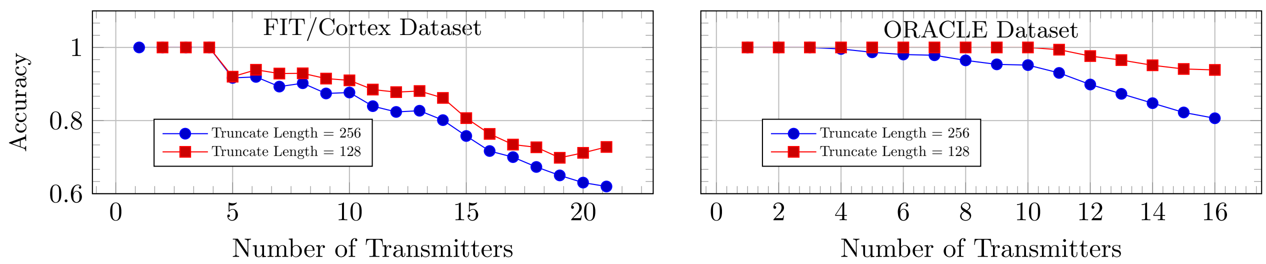


Figure 4.6: The classification becomes harder when increasing the number of transmitters. The left figure shows the trend of RC's performance when more transmitters are involved for FIT/Cortex dataset at two different truncate length. For a total of 21 transmitters, the performance decreases to 73% and 63% for data truncate length 256 and 128, respectively, which is not able to compete with CNN. But the classification accuracy remains competitive for classification of less than 14 transmitters. It's also worth noting that the narrowband OEO-based RC only uses 0.11% or 0.05% of sampled points for training compared with that of CNNs when truncating the data length from 600 to 256 or even 128. The right figure shows the trend of RC's performance when more transmitters are involved for ORACLE dataset. Here we examine two settings. For truncate length of 256, the narrowband OEO is trained by 100 training examples per transmitter, which aims at higher prediction accuracy. For truncate length of 128, the narrowband OEO is trained by 60 training examples per transmitter, which aims at higher training and processing speed.

CNNs. Moreover, the simplified network structure with only the read-out layer being trained limited its ability to be extended to classification of more transmitters.

However, we argue that the narrowband OEO-based RC has its own unique advantages over CNNs in following aspects. First, the simplified network structure, combined with ultra-fast photonic hardware platform, grants it ultra-high processing speed without the need to demodulate the I/Q signals. The deep layers of CNNs itself is a double-edged sword. Although it introduces more parameters and allows the network be more easily adaptable for classification of more transmitters at higher accuracy, the deep network structure enforces the input data to be propagating forward the deep networks layer by layer, which inevitably slows down the processing speed. For applications like RF fingerprinting where real-time processing is in demand, this drawback could be fetal. The complicate CNN structure also means that more parameters should be trained, which brings another problem that a large training set size is needed to avoid CNN's poor performance in low-resource settings. The time-consuming back-propagation algorithms for training a CNN further deteriorate the situation when new transmitters are connected in and the whole CNN has to be re-trained. Narrowband OEO-based RC tackles the problem that CNNs encountered and provide an alternative solutions for RF fingerprinting of less transmitters in low-resource settings. Its performance at fewer number of transmitters ≤ 14 remains competitive with CNN while the total number of samples for training could be reduced to only 0.11% or even 0.027% required by CNN. What's more, the shallow network and simple Ridge Regression algorithm avoid time-consuming back propagation algorithms, making it a great candidate for application scenarios where real-time processing speed and frequent update of transmitters are needed. Nonetheless, it is more than complementary solution to RF fingerprinting with application scenarios that not suitable for CNNs. Its ultra-fast processing speed also make it a great candidate for communication and security system where a pre-selection and blocking of malicious transmitters is in need.

4.3.1.5 Different operation region γ

In this part, we study the dynamic regime of narrowband OEO operates on and its impact on the RC's performance. Here, the narrowband OEO-based RC is trained and tested with the same injection amplifying factor ρ but varying the parameter γ so that the narrowband OEO-based RC is operating on different regime. The γ is set to be equals to $0.9, 1.1, 2.2, 2.4, 3$, respectively. These numbers are corresponding to the dynamic regime before the Hopf's bifurcation, after Hopf's bifurcation, before the second bifurcation, slightly after the second bifurcation and chaotic regime. The best accuracy obtained for test set is $61\%, 90\%, 93\%, 92\%, 90\%$, respectively. It seems a little bit strange that we can obtain 90% accuracy at $\gamma = 1.1$, slightly after the Hopf's bifurcation, which was proven not working well in previous modulation recognition paper.

4.3.2 Numerical Results on ORACLE Dataset

4.3.2.1 Latency and training set size optimization

Data truncate length and training set size are two parameters that most closely related to the narrowband OEO-based RC's processing speed and time needed for training. In this section, we discuss the trade-off between latency and accuracy, as well as the training set size optimization. We first limited the class of transmitters to 10 and explore its performance with different number of training set size at two different data truncate length. Then, the research is extended to a setting that all 16 different transmitters in the dataset are involved.

Following previous section, the original samples of I/Q signal are first pre-processed into training set and testing set where each example contains 256 or 128 samples. The hyper-parameters of narrowband OEO-based RC is then fine-tuned by scanning through different ρ and γ. The optimized values of ρ and γ (where $\gamma = 2.4$, $\rho = 9.61$) is then fixed when scanning through different number of training set size. Figure 4.5 shows the narrowband OEO-based RC's performance with different number of training example for each transmitter. A similar trend of performance to FIT/CorteXlab dataset can be observed here: for both data truncate length, the performance first increases when more training examples are used to train the RC. Then, its performance saturated when number of training example per transmitter reaches 50. With a truncate length of 256, the narrowband OEO-based RC achieved nearly perfect classification accuracy of $\sim 99.9\%$ and only needs 50 training examples for each transmitter. For truncate length of 128, the classification accuracy is $\sim 95\%$ for 60 training examples per transmitter.

With the same simulation settings of classification 10 transmitters, the narrowband OEO-based RC is then trained and tested extending to all 16 different transmitters. As shown in Figure 4.5, it requires only 50 training examples per transmitter for data length truncated at 256 to reach the best performance $\sim 94\%$. Interestingly, instead of saturating at 50, the narrowband OEO-based RC's performance keeps growing when the number of training examples per transmitter increases to 200. This can be explained by that narrowband OEO-based RC is still under-fitted until the number of training example per transmitter finally reaches 200 since example length is reduced.

4.3.2.2 Extending to more transmitters

In this section, we examine the narrowband OEO-based RC's ability to classify different number of transmitters. More training examples and longer data truncate length generally endorse a better performance but inevitable increases the training and processing time. A careful trade-off should always be made between the accuracy and speed. Here, we check the narrowband OEO-based RC's performance for the classification task from 2 to 16 transmitters. One setting involves more training examples (100 per transmitter) and each example contains more samples (256 samples per example), aiming at better performance. The other system settings emphasizes the processing speed with limited training examples (60 per transmitter) and each example contains less samples (128 per transmitter). Figure 4.6 shows the prediction accuracy with transmitters from 2 to 16. For simpler classification tasks that only involves 2 to 4 transmitters, both settings achieve perfect classification. The performance of narrowband OEO-based RC with less training examples and less samples started to degrade when the 5 or more transmitters are involved. It maintains an accuracy above $\sim 90\%$ for the classification of 12 transmitters but the prediction accuracy severely degrades to $\sim 80\%$ when further increase number of transmitters from 12 to 16. Meanwhile, the narrowband OEO-based RC trained with more training examples and more samples experiences less reduction in performance. The curve remains flat with a perfect classification accuracy for classification of 2 to 10 transmitters. The prediction accuracy reduced slightly when increasing the total number of transmitters from 11 to 16. It is able to main a very high prediction accuracy of $\sim 94\%$ even for the most complicated classification of 16 different transmitters. It's worth mentioning that even with first configuration that involves more training examples and large truncate length, the total number of sample points needs to train narrowband OEO-based RC is only $25,600$ per transmitter, which is only $\sim 0.2\%$ that used to train a CNN.

4.3.2.3 Different operation region γ

Our first simulations is to train and test the narrowband OEO-based RC with a fine-tuned $\rho = 9.61$ by 100 training examples per transmitter. The RC is performing a classification task of 10 transmitters with truncate length of 256. We obtained a nearly perfect accuracy above 99% for all γ value of $0.9, 1.1, 2.2, 2.4, 3$. In order to further check if narrowband OEO-based RC stills performs similarly across different operation regime for this dataset, we purposely make it hard for the narrowband OEO-based RC by performing a more complicated classification task of 16 transmitters with further limiting the training set size to 60 per transmitter. Moreover, the truncate length is also set to the shorter 128. For γ value equals to $0.9, 1.1, 2.2, 2.4, 3$, the best accuracy is $77\%, 78\%, 78\%, 80\%, 79\%$, respectively. This result suggests that for the ORACLE RF fingerprinting dataset, there exists a wide range of operation regime that narrowband OEO-based RC performs well.

4.3.3 Numerical Results on AirID Dataset

4.3.3.1 Latency and training set size optimization

Similar to previous ORACLE dataset, we start our analysis by examining two parameters that are related to processing speed: data truncate length and training set size. Before fed into narrowband OEO-based RC, the dataset is first pre-processed into training set and testing set where each example contains 256 or 128 samples. The optimal value of ρ is obtained via a scanning and remain fixed to 5.61 when the narrowband OEO-based RC is trained with different training set size. As shown in Figure 4.5, the performance curves follows similar trend when applying narrowband OEO-based RC to previous two datasets. The narrowband OEO-based RC starts learning information with a limited number of 10 training examples per transmitter and the performance steadily increases when the number of training examples is further increase to 90. Then, its performance saturated after training set size exceeds 100 per transmitter. The best prediction accuracy obtained here is $\sim 97\%$ with 200 training examples for data truncate length of 256. For data truncate length of 128, the performance has a gap of $\sim 5\%$ to that of 256.

4.3.3.2 The role of effective gain γ and injection amplifying factor ρ

Effective gain γ of the feed-back loop directly determines the operation regime of narrowband OEOs. Meanwhile, the injection amplifying factor ρ indicates the signal strength being injected into the OEO feedback loop. In this part, we examine the effects of these two parameters toward OEO-based RC's performance. Figure 4.4 shows the OEO-based RC's performance with different injection amplifying factor ρ in 5 typical effective gain γ values. As can be seen, the OEO-based RC performs best when $\gamma = 0.9$ and decreases when increase the γ from 0.9 to 3.0, which means that narrowband OEO-based RC performs best near the first bifurcation and the performance constantly decreases when the operation regime is moved the second bifurcation. With the same injection amplifying factor ρ, this trend always holds. This suggests that for training a narrowband OEO-based RC on a dataset, these two hyper-parameters could be fine-tuned independently during the training. In general, a γ value having better performance under one injection amplifying factor ρ value indicates it has better performance regardless the ρ value.

4.4 Conclusion

In this chapter, we have proposed a scheme that utilizes narrowband OEO as ML platform to deal with RF fingerprint task - a task that high processing speed for I/Q modulated signal is desired. We have numerically simulated the narrowband-OEO based RC and tested its performance with three different datasets: the FIT/Cortexlab RF fingerprinting dataset, ORACLE RF fingerprint dataset and AirID RF fingerprinting dataset. We have shown that with a greatly reduced training set size that is at the level of 0.1% to 1% to that of CNN, the narrowband OEO-based RC maintains a high classification accuracy that is comparable to state-of-the-art. This greatly reduced requirement for training data of narrowband OEO-base RC gives an edge for application scenarios where real-time classification of unseen transmitters is in demand. The simpler structure and fast training algorithm, combining with less hunger for training data make it easy and fast to re-train the narrowband OEO-based RC when deployed. Our results also shows that with the ability to directly process I/Q modulated signals, narrowband OEO-based RC is suitable to be directly integrated into radio-over-fiber systems as a ML platform for next-generation of smart communications systems.

Chapter 5: Experimental Results on Narrowband OEO with Limited Resource and a New Metric for Measuring the Performance

In the previous two chapters, we presented numerical simulation results of the narrowband OEO-based RCs for IQ-modulation classification and RF fingerprinting. However, it is still challenging to compare the performance of different ML learning algorithms that take into account the processing speed, resources required to train, and the performance. In this chapter, we further push the resource to train the narrowband OEO-based RC to a lower limit to test its performance. Moreover, we propose a new metric that is able to compare performance across different machine learning algorithms.

5.1 Experimental Setup

[1] In our experimental setup, we employed an MBX bias and power controller in conjunction with an FPC562 fiber polarization controller (both from Thorlabs) to actively stabilize and manage the optical output of a Thorlabs FPL1009S laser diode. The optical subsystem was further configured with an EOSpace lithium niobate Mach-Zehnder intensity modulator (10 GHz bandwidth), a Coherent XPDV2120R photodiode (50 GHz bandwidth), and approximately 1 km of Corning SMF-28 single-mode fiber to complete the optical path. On the RF side, signal amplification and filtering were achieved using a cascade of two Mini-Circuits ZFL-500LN+ low-noise amplifiers, combined with a Mini-Circuits SBP-21.4+ bandpass filter (centered at 21.4 MHz, with a 19.2–23.6 MHz passband). A Mini-Circuits ZMDC-10-1+ directional coupler was used to tap a portion of the RF signal for real-time monitoring and data

[1] The experiments are performed by Benjamin Klimko

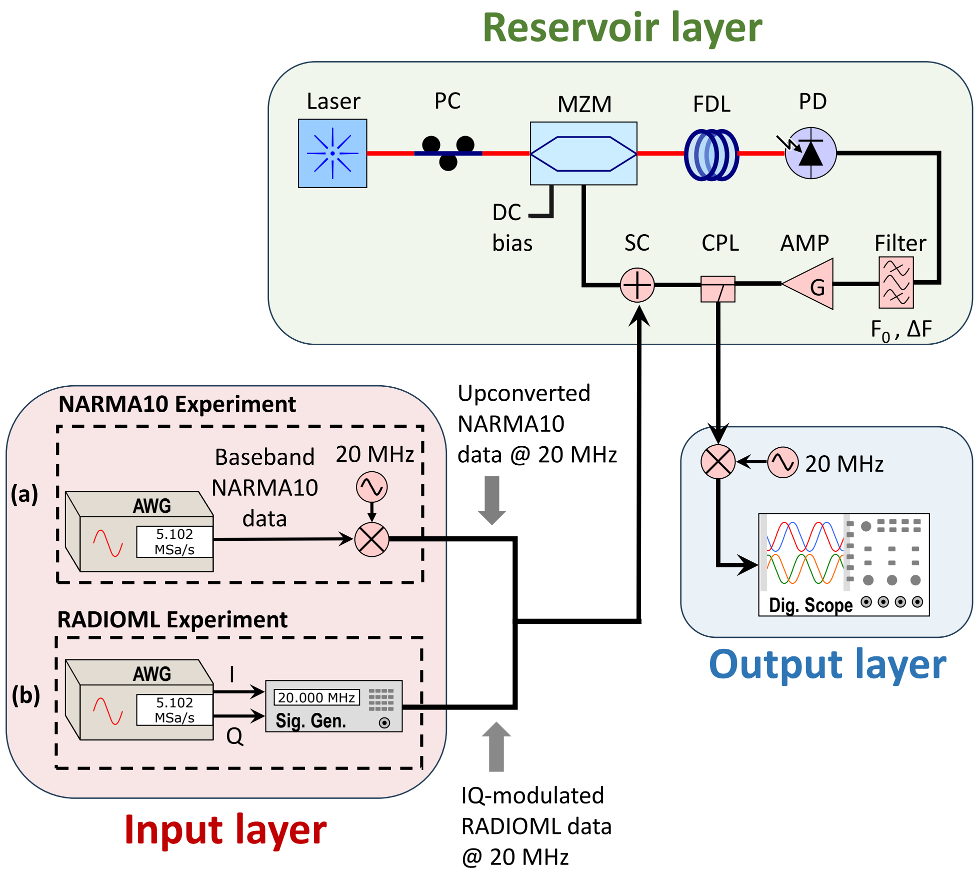


Figure 5.1: Schematic for the experiments of the narrowband OEO-based RC. PC: polarization controller; MZM: Mach-Zehnder modulator; FDL: fiber delay line; PD: photodiode; AMP: amplifier; CPL: coupler; SC: signal combiner; AWG: Arbitrary waveform generator. (a) NARMA10 baseband signal is output from AWG and mixed up to the passband; (b) IQ data is output from the AWG and modulated by a signal generator. Either case (a) or (b) is connected to the OEO via the SC but not both simultaneously. The optical path is shown in red, and the RF path is in black.

acquisition. To facilitate the injection of the external reference signal from the arbitrary waveform generator (AWG) into the optoelectronic oscillator (OEO) feedback loop, a Mini-Circuits ZMSC-2-2 two-way power splitter was integrated into the RF path. All RF components were sourced from Mini-Circuits. A detailed schematic of the complete experimental configuration is presented in Figure 5.1.

To mitigate systematic errors during data collection, a shared local oscillator was employed for both upconversion and downconversion in the benchmarking experiments. This configuration ensured that frequency drift and phase noise were common-mode, thereby minimizing the introduction of artifacts during post-processing. In the RADIOML experiments, the local oscillator was phase-locked to the signal generator via its 10 MHz reference output, establishing a shared frequency standard and reducing timing and synthesis errors during independent generation and acquisition of the 20 MHz carrier.

The masked baseband signal is generated by a Siglent SDG6022X arbitrary waveform generator (AWG) and subsequently upconverted to a 20 MHz intermediate frequency using a Mini-Circuits ZLW-1-1+ mixer, with a Siglent SSG3021X-IQE signal generator providing the local oscillator (LO). The circulating signal within the optoelectronic oscillator (OEO) loop was tapped via a Mini-Circuits ZMDC-10-1+ directional coupler and digitized at a sampling rate of 12.5 MSa/s using a Siglent SDS5034X oscilloscope. Downconversion to baseband was performed using a second Mini-Circuits ZLW-1-1+ mixer driven by the same SSG3021X-IQE LO source, thereby maintaining phase coherence between upconversion and downconversion stages. The acquired baseband signals were saved to disk and subsequently transferred to a computer for offline analysis. Each experimental run was repeated ten times with the external modulation signal applied to the Mach-Zehnder modulator (MZM) at an amplitude of 2 V_{pp}. Upon transfer to the analysis environment (MATLAB), the recorded data underwent a standardized preprocessing pipeline. A conversion factor was computed to account for the discrepancy between the input data sample rate and the oscilloscope acquisition rate, enabling reconstruction of the full-length example from the saved data. Specifically, if the input signal was generated at 1 MSa/s with a sequence length of 1000 samples, but recorded at 10 MSa/s by the oscilloscope, the resulting captured waveform would consist of 10,000 samples corresponding to the original example duration.

Minor modifications to the experimental setup are necessary to transition from the benchmarking task to the modulation recognition task. Masked in-phase and quadrature (IQ) signals are generated by the AWG and applied to the signal generator to produce a 2 V_{pp} modulated carrier at 20 MHz, which is subsequently injected into the RF port of the Mach-Zehnder modulator (MZM).

5.2 Proposed NET Metric and Dataset

5.2.1 NET Metric

The reservoir computing features a much simpler structure with less training data required. However, much of the current research within the machine learning (ML) community continues to focus on enhancing accuracy, often through the acquisition of larger neural network and datasets. On the other hand, the resources required to train these algorithms have been widely overlooked and there often lacks quantitative analysis on it. Therefore, it becomes necessary to have a metric that is able take all of these factors into consideration. Naturally, a new metric that takes into account the performance of the algorithm, the size of the neural network, and the resources required for training can be formalized intrisically as

$$\eta = N \times E \times T. \tag{5.1}$$

where N is total number of nodes to perform the computation , the error rate E normalized to random accuracy (so that we should typically expect $0 \leq E \leq 1$), and training set size T measured in thousands of examples (e.g. for 5000 training examples, $T = 5$).

The idea behind this metric is that an ideal system should simultaneously minimize N, E, and T, leading to a minimum value of the NET product. The metric thus incorporates the system overhead N and evaluates the amount of training data T required to achieve a given error rate E. The number of nodes N serves as a proxy for computation time, which is inherently hardware-dependent, often not reported in the literature, and may be less meaningful in experimental systems not designed for real-time, low-latency processing. Although the exact data processing method from the reservoir may vary across different experiments, it is reasonable to assume that a larger system producing more raw data (i.e., more virtual nodes) will require more computation time than a smaller system that outputs less raw data.

However, there is an issue with treating N, E, and T equally in the metric, as an unweighted product $\eta = NET$, since that the primary goal of reservoir computing is to minimize error, or equivalently, to maximize accuracy. Consequently, it is necessary to heavily penalize poor error performance. This can be accomplished by replacing E with $\operatorname{atanh}(E)$, resulting in the redefined metric:

$$\eta = NT \operatorname{atanh}(E) \tag{5.2}$$

The advantage of this formulation is that for low error rates ($E \to 0$), we have $\operatorname{atanh}(E) \approx E$, as initially intended. However, since $\operatorname{atanh}(1) = +\infty$, the metric imposes a disproportionately large penalty for high error rates. This ensures that the metric cannot be circumvented by arbitrarily reducing N and T to offset an unacceptably high error rate as $E \to 1$.

5.2.2 Dataset

The tenth-order nonlinear autoregressive-moving-average (NARMA10) task is a regression task with the goal of predicting one step into the future given the current value of the time series. The time series $y(n)$ is defined by the recurrence

$$\begin{aligned} y(n+1) = 0.3y(n) + 0.05y(n)\sum_{i=0}^{9} y(n-i) \\ + 1.5u(n-9)u(n) + 0.1 \end{aligned} \tag{5.3}$$

where $u(n)$ is a random variable drawn from a uniform distribution on the range [0, 0.5].

We used normalized mean squared error (NMSE) as our performance metric, defined by

$$NMSE = \frac{\langle (\mathbf{Y} - \hat{\mathbf{Y}})^2 \rangle}{\langle (\mathbf{Y} - \langle \mathbf{Y} \rangle)^2 \rangle} \tag{5.4}$$

The other dataset is the IQ-Modulated signal dataset RADIOML 2016.04C classification mentioned in Sec. 3.4. Due to the complexity of the dataset, a subset of the modulations with: 64-QAM, QPSK, GFSK, 8-PSK, and AM-DSB is chosen. To further test its ability with limited resources and less number of virtual nodes, in the experiment, we set the total number of virtual nodes to 25. (VNs =25)

5.3 Results and Discussion

The numerical simulation results and experimental results of NARMA10 task is shown in Table 5.1. It can be seen that the narrowband OEO-based RC has similar NMSE, as well as NET score η when compared to other RC implementations with a similar number of virtual Nodes. These results prove that the narrowband OEO-based RC has similar computational power for processing data compared to the wideband OEO-based RCs.

Table 5.1: NET Metric for Optoelectronic Reservoir Computing NARMA10 Results

Reference	Nodes (N)	Training Examples (k=1000)	NMSE (Lowest reported)	NET score η
[141]	50	3k	0.168	25.44
[142]	47	1k	0.23	11.01
[94]	80	20k	0.185	29.95
[91]	50	1k	0.168	13.57
[113]	50	1k	0.103	5.17
Experiment	25	2k	0.251	12.82

Meanwhile, the numerical results and experimental results for the IQ-modulation classification are shown in Table 5.2. We list η we get from simulations reported in Ch. 3 when the NMSE is optimized at 0.111 as well as results for reduced training set sizes. In comparison with the CNN or DNN-based IQ-modulation classification, the narrowband OEO-based RC has achieved two orders of magnitude better performance regarding η while maintaining a similar level of error rate (E). If willing to sacrifice the error

Table 5.2: NET Metric for RADIOML Classification Results, 18 dB SNR input

Ref.	Nodes/ CNN Filter Values	Training Examples (k=1000)	Min E	Min η	Max η
CNN [8]	427	96k	0.07	3162.6	4071.5
CNN2 [8]	1515	96k	0.04	6403.5	8007.3
DNN [8]	907	96k	0.11	10 587.6	13 516.6
OEO RC simulation [143]	400	2.53k	0.111	124.2	–
OEO RC simulation [143]	400	0.11k	0.46	21.89	–
OEO RC simulation [143]	400	0.33k	0.27	36.54	–
Experiment	25	0.1k	0.57	2.23	3.11
Experiment	25	0.6k	0.383	17.33	25.59

rate (E) to the level of around 0.3, with a total number of $N = 400$, the η can be further optimized by one order of magnitude lower. It is also worth noting that for an experiment that aims at optimizing the η, a best η of 2.23 can be achieved. However, this η is achieved by limiting both the total number of virtual nodes to 25 and with only 0.1k of data for training. For such a scenario, the error rate E is relatively large at 0.57. One might argue that such a large E might not be practical for applications. Indeed, this result is achieved with many IQ-modulation formats. The error rate can and will be further reduced if only 2 types of modulations are involved - the exact reason that we push to an extreme for optimizing η. Because in the context of a "friend or foe" identification task, there will be only two types of signals that will be involved. Such an application scenario implies a more urgent demand to recognize the target with as little as possible data (T) and ultrafast processing speed (N).

Another interesting result from the Table 5.2 is a question we have already raised in the general discussion of modern AI techniques in Sec. 1.1.6: the more, the better? In the deep neural network (DNN) era, a clear trend has emerged: increasing computational resources and model complexity, typically through a higher number of parameters at million level, has consistently led to performance improvements. This trend has only further intensified with the advent of large language models (LLMs), where scaling up data, architecture, and training infrastructure has become a defining characteristic of state-of-the-art systems. The number of parameters can easily surpass a billion. With this trend, we have seen an inevitable exponential growth of the associated computational resources, data, and energy costs required for these AI technologies. However, the researchers in both industry and research are still following the trend - optimizing the performance by investing more resources and incorporating more parameters for ML algorithms, regardless of the existing diminished returns. Currently, this phenomenon, especially the diminished return seems to be a blind spot for the whole field, and there is an obvious lacks of a metric for quantification. Therefore, the NET we proposed here offers insight and an opportunity to see these long-existing but overlooked diminished returns in the machine learning field since the NE product inside the η can also be considered as a metric that involves the "cost" of a machine learning algorithm into consideration.

It can be seen that "there is no free lunch" in the machine learning field. A better algorithm can prevail, for example, the DNN mentioned in [8] has a similar error rate E of 0.11 comparing to our models in [143], but our model prevails by reducing η to 2 orders lower, which implies a 2 orders lower cost. Meanwhile, the best model reported in [8] is referred to as CNN2, with error rate $E = 0.04$. However, to achieve this E, their η is about ~50 times to ours, which indicates an overall 141 times higher cost inside NE product!

Meanwhile, for the same machine learning algorithm (in our case, the reservoir computing), there is always a trade-off. While maintaining the same level of η, one might have to sacrifice the performance in order to reduce the total number of virtual nodes or the number of training examples required.

5.4 Conclusion

In this chapter, we introduce a new metric η that gives insight into the computational resource, training set size, and performance that is useful to directly compare performance across different machine learning algorithms and quantitatively analyze the diminished returns of these machine learning algorithms.

Typically, it can be seen that an increment of an order of performance in E does not reduce the η, but, by contrast, increases the η by several orders. Secondly, our results in the NARMA10 dataset show that the computational power of a narrowband OEO-based RC is similar to that of wideband OEO-based RCs. Lastly, we push the narrowband OEO-based RC to its limit by experimentally reducing the training examples and total number of virtual node while maintain an acceptable performance, which paves the road to certain application scenarios that the computational resources and data are highly limited.

Chapter 6: Photonic Reservoir Computing using Optoelectronic Oscillators with Direct Laser Modulation

Reservoir computing (RC) is a simple yet efficient machine learning algorithm that has reached state-of-the-art performance on several benchmarks. Incorporating ultra-fast photonic devices, implementations of RC based on optoelectronic oscillator (OEO) with a fiber delay-line has shown the ability to process both the optical and RF signals at an unprecedented high speed. However, previously reported OEOs used for reservoir computing require an external modulator, typically a bulky $LiNbO_3$ Mach-Zehnder Modulator (MZM), not only making it hard for such systems to be integrated onto a silicon chip but also increasing the overall cost. In this article, we present a simpler implementation of OEO-based RC without the need to use external modulators. Contrary to previous implementations of OEO-based RCs that inject the signal into the delay line via external modulators to excite VNs, we show that the input signal can be injected into the delay line by internally modulating the laser. The intensity modulation of the laser in this architecture introduces a piecewise-linear power intensity transfer function, leading to a rectified linear unit (ReLU) non-linearity inside the reservoir layer. We first derive the equations that govern the dynamics of such an OEO-based RC. Then, we numerically simulate its dynamics and successfully train and test this direct laser modulated OEO-based RC on several datasets, namely the Nonlinear Auto Regressive Moving Average (NARMA10) dataset, Nonlinear-Channel Equalization (NCE) dataset, Radar Signal Forecasting dataset, and Santa Fe Laser Series Prediction dataset, reaching a performance that is equivalent to or better than the traditional OEO-based RCs with external modulators.

6.1 Simplest OEO for RC

In previous chapters, we have demonstrated that a narrowband OEO-based RC could be applied to perform classification tasks on I/Q modulated radio-over-fiber signals without the requirement of demodulating and processing the signals into computer-compatible format first. Such a system has been proven to be valid and powerful in performing modulation format recognition [143] and RF fingerprinting [144] with state-of-the-art performance on several benchmarks.

However, all of the aforementioned OEO-based RCs are implemented using bulky discrete photonic elements. The requirement of an external modulator (EO), typically a electrooptical $LiNbO_3$ Mach-Zehnder Modulator (MZM), makes it hard for them to be integrated. Nonetheless, several applications that emerged since the era of 5G have generated unprecedented demands for ML platforms with a compact size, low energy consumption, and compatibility for integration. Meanwhile, several miniature OEO structures have already been proposed and studied for ultra-pure RF signal generation but ignored by the photonic RC research communities. Reference [145] serves as a good example. In order to generate ultra-pure RF signals, an autonomous OEO that cancels the external intensity or phase modulator and replaces them by directly performing intensity modulation through the laser pump current has been proposed and studied. With the simplest structure ever known, this direct laser modulated OEO (DL-OEO) provides the possibility for a cost-effective, energy-efficient, and compact alternative to the traditional implementation of OEO-based RCs. Therefore, in this paper, we introduce this simplest structured OEO for machine learning purposes and demonstrate its suitability to be applied for photonic implementations of RC. Moreover, we will show that even with the simplest structure, this OEO-based RC could still reach a competitive or better performance compared to traditional OEO-based RCs.

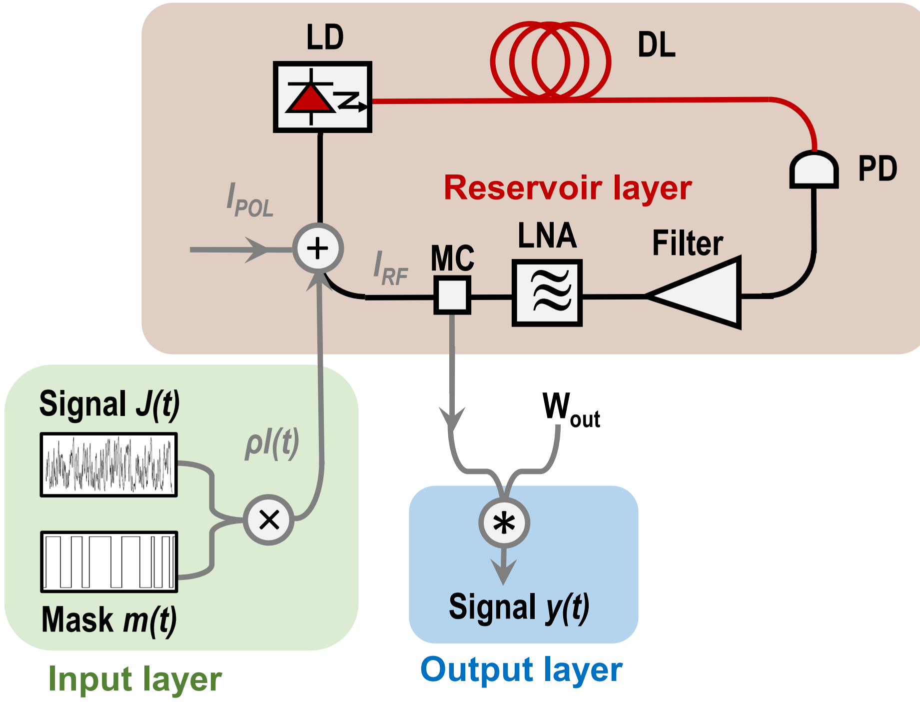


Figure 6.1: Schematical representation of the reservoir computer based on a DL-OEO. DL: Delay line; LNA: Low-noise amplifier. PD: Photodiode. MC: Microwave Coupler. The RF filter in this diagram is introduced by the response time of the system, not an actual RF filter.

6.2 System and Model

Typically, an OEO could be characterized by four key components: linear gain, filter, delay line, and nonlinearity. Figure 6.1 shows the configuration of the DL-OEO system under study. A continuous wave laser diode (LD) is operating at the telecom wavelength of $\lambda_{\mathrm{L}} = 1550$ nm. It is driven by a feedback current $I_{\mathrm{RF}}(t)$ from the RF output branch of the OEO, a polarization current I_{pol} and an externally injected signal $I(t)$ that is used to couple the data point from the dataset. The threshold current of LD is set to $I_{\mathrm{th}} = 16$ mA. The output of the LD is then coupled into an optical fiber with a length of $L = 40$ m, yielding a delay time of $T = nL/c = 0.2\ \mu$s, where n is the refractive index of the optical fiber with $n = 1.5$ and c is the velocity of light in vacuum. The optical output from the optical fiber is then converted into a photocurrent by a photodetector (PD) with responsivity $S = 4.75$ V/mW and a detection bandwidth of 500 MHz. The PD converts the input power $P(t)$ into a voltage $V(t)$, which is then coupled to a voltage variable attenuator (VVA) that has a gain equals to G. It is worth noting that the bandpass filter of the DL-OEO originates from the cascaded elements inside the loop, which further simplifies the implementation. For numerical simulation purposes, this bandpass filter could be characterized and modeled by a frequency band limited by a low pass frequency $f_\mathrm{L} = 531.5$ kHz and a high pass frequency $f_\mathrm{H} = 6.4$ MHz, leading to two different time scale $\theta = 1/2\pi f_\mathrm{L} = 0.3$ μs and $\tau = 1/2\pi f_\mathrm{H} = 25$ ns.

The LD plays a central role in converting the electrical injected signal to the optical domain which is then used to excite VNs in the reservoir layer. The power-intensity (PI) transfer functioncould be modified for our RC with an injection as follows [145]:

$$P(t) = \begin{cases} 0, & \text{if } I_\mathrm{RF} + \rho I(t) \leq I_\mathrm{o} \\ \mu[I_\mathrm{RF} + \rho I(t) - I_\mathrm{o}], & \text{if } I_\mathrm{RF} + \rho I(t) > I_\mathrm{o}. \end{cases} \tag{6.1}$$

Here, $\mu = \eta_\mathrm{d}\ \hbar\nu$ represents laser conversion slope, where η_d is the quantum efficiency, $\hbar$ is the Planck constant, and $\nu = c/\lambda_\mathrm{L}$ is the laser carrier frequency. Additionally, we also introduce the new threshold current $I_\mathrm{o} = I_\mathrm{th} - I_\mathrm{pol}$, while ρ is a dimensionless amplifying factor for the injected signal that directly modulates the laser diode (LD). Following a few steps of derivation from ref. [145], we can obtain a simplified and dimensionless equation that rules the behavior of DL-OEO with an RF internally injected signal $\rho I(t)$ as follows:

$$x(t) + \tau\frac{dx(t)}{dt} + \frac{1}{\theta}\int_{t_0}^{t} x(\xi)d\xi = \beta D[x_T + \rho u_I(t) - \alpha], \tag{6.2}$$

where $x(t)$ is a normalized dimensionless variable with $x(t) = V_\mathrm{RF}(t)/V_\mathrm{ref}$ and $u_I(t) = I(t)R_\mathrm{Z}/V_\mathrm{ref}$. The variables $V_\mathrm{RF}(t)$ and $u_I(t)$ are converted through the characteristic impedance of $R_\mathrm{Z} = 50\ \Omega$ after being multiplied with the feedback loop current $I_\mathrm{RF}(t)$ and the injection current $I(t)$, respectively. Thus, we have $V_{\mathrm{RF}} = I_{\mathrm{RF}} R_{\mathrm{Z}}$ where $V_{\mathrm{ref}} = 1$ V is a conveniently assigned reference voltage, while $x_T = x(t-T)$ is the time-delayed part of $x(t)$ induced by the time-delay line. The parameter α is a dimesionless voltage offset with $\alpha = I_{\mathrm{o}} R_{\mathrm{z}}/V_{\mathrm{ref}}$, and $\beta = \kappa S \mu G / R_{\mathrm{z}}$ is the feedback loop gain that takes into account the total loss κ inside the OEO loop.

It is worth noting that the nonlinearity of function Eq. (6.1) inside the DL-OEO-based RC system introduced by the PI nonlinear transfer function of the LD as

$$D(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ x, & \text{if } x > 0. \end{cases} \tag{6.3}$$

It has the same format as the ReLU function extensively used by convolutional neural networks (CNNs) in the field of machine learning for image and video related tasks.

6.2.1 Time-division Multiplexing

In the temporal domain, all time-delay systems are theoretically infinite-dimensional, leading to a property that within a single delay line, any input data with a finite dimension could be expanded to arbitrarily high dimensions sequentially. In practice, limited by the response time of the system, sample rate, and modulation bandwidth, the input data could only be mapped into a finite dimensional space, albeit higher than the original. For an RC based on the time-delayed system, the input data is expanded temporally into a high dimensional space by multiplying each data point with a randomly generated N-valued time series with a period equals to delay time T. This N-valued time series functions as the input mask of regular reservoir layer $\mathbf{W}_{\mathrm{in}}$. The only difference between a regular digital RC and a photonic RC based on a time-delay line is that the former maps input data to the reservoir layer in the spatial domain while the latter maps the data in the temporal domain. The procedure of expanding one data point into an N-dimensional space is repeated for all data points within each example and is known as time-division multiplexing [6]. The states of the N values multiplied by a data point that later go through a non-linear transform will be extracted by a sampling process to represent the reservoir states of the RC implementation in the temporal domain and are therefore known as VNs. Therefore, for the purpose of sampling, it becomes critical to maintain an equal time separation between the N-valued time series fed into the delay line, leading to a time separation between each value with $\Delta T = T/N$. Unlike its digital counterpart which requires an explicit implementation of the non-linear function, the non-linearity of RC based on a time-delay system is introduced by the non-linearity of the system itself.

6.2.2 Implementation of RC based on the DL-OEO

In order to implement the RC based on the DL-OEO, a continuous injection signal $u_I(t)$ in Eq. (6.2) must be generated from all the examples in the dataset accordingly. The first step is to generate a continuous wave from the discrete data points of each example in the dataset. To achieve this, each sample corresponding to one data point inside all examples is extracted and held for a time duration equivalent to the DL-OEO's delay time T [143], i.e.,

$$J(t) \equiv X^i(k), (k-1)T \leq t \leq kT, \tag{6.4}$$

where the superscript i denotes it is the i^{th} example in the training set and k signifies that it is the k^{th} sample of the i^{th} training example. Then, in order to maintain the mask weights being fixed for each example, the mask signal $m(t)$ must be generated as a continuous periodical signal and aligns its period to the system's delay T. For all four datasets we used to train and test our DL-OEO-based RC, each example contains only one sample, leading to $k \equiv 1$. To ensure that each sample is coupled into the delay line and excites exact N equally distant VNs, each period of mask signal must contain a total of N weight values and any two consecutive weights must be kept with the same separation $\Delta T = T/N$. Following [91], we adopt a mask $m(t)$ with its weights uniformly sampled from six values. After obtaining mask $m(t)$ and a continuous wave $J(t)$, the continuous injection signal $u_I(t)$ can be obtained by multiplying them together:

$$u_I(t) = J(t) \cdot m(t). \tag{6.5}$$

The next step is to sample the states from the reservoir layer to train the read-out layer of RC. To achieve this goal, the middle points of each interval ΔT from the continuous time series $x(t)$ are sampled to represent the states of the VNs. These samples could be organized into a matrix $\mathbf{X}$ with X_{ij} being the reservoir state of i^{th} training example's j^{th} VN state:

$$\mathsf{X}_{ij} = x[i \cdot T + (N - j) \cdot \Delta T]. \tag{6.6}$$

After collecting all the reservoir states for the whole training set and organizing them in an order specified by Eq. (6.6), we now get the matrix $\mathbf{X}$ that is equivalent to $\mathbf{M_x}$ in the digital implementation of RC in Eq. (3.17). Identically as for conventional digital RC, we can apply the Ridge regression algorithm [123] to obtain optimal read-out layer weights:

$$\mathbf{W}_{\text{opt}}^{\text{RO}} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{Y}, \tag{6.7}$$

where $\mathbf{Y}$ are the labels of training set and λ being the regularization factor to prevent overfitting.

Different from the narrowband OEO-based RCs, this DL-OEO is a broadband OEO-based RC. This means that there are no timescale issues in numerical simulations and we can directly simulate Eq. 6.2 using Runge-Kutta method.

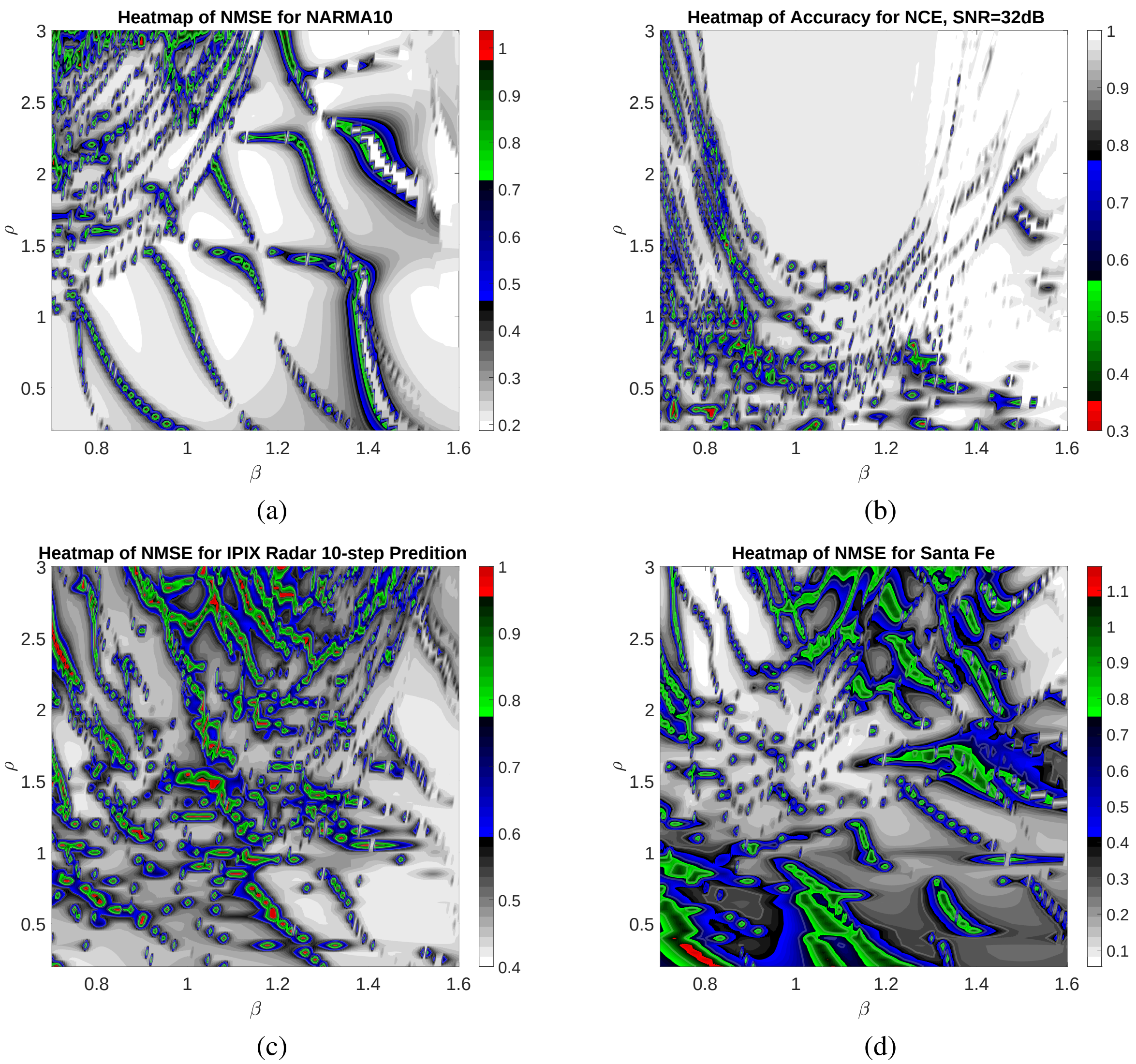
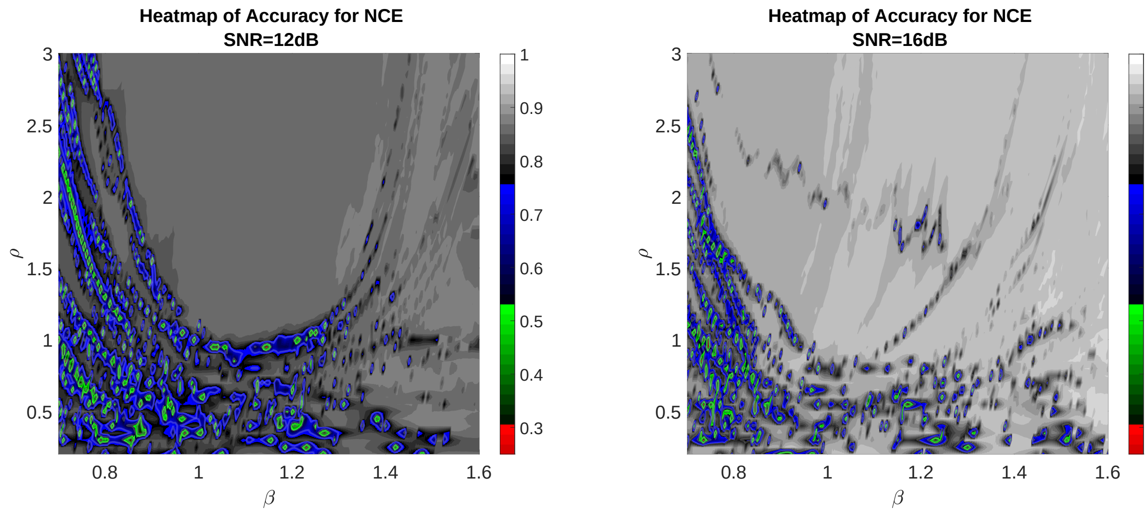


Figure 6.2: This figure shows the simulation results of RC based on the DL-OEO's performance on four datasets. The heatmaps are scanned through system gain β and injection amplification factor ρ. It is worth noting that the metrics for NARMA10, Radar Prediction, and Santa Fe Laser prediction tasks are NMSE - a smaller number corresponding to better performance while the metrics for the NCE dataset is accuracy - a larger number corresponding to better results. Here, for the Radar prediction task, the NMSE is measured on a 10-step prediction - the most difficult test for this dataset. For NCE, the accuracy is measured on a dataset with $\mathrm{SNR} = 32$ dB. The total number of VNs excited at the reservoir layer is $N = 50$ with a modulation frequency of $f_\mathrm{m} = 250$ MHz. For the NARMA10 dataset, Radar dataset, and Santa Fe laser prediction task, the best NMSE obtained is 0.2008, 0.4106, and 0.0563, respectively. For the NCE dataset, the best accuracy obtained is 99.42%.



(a) (b)

Figure 6.3: The left figure is a heatmap of accuracy for the NCE dataset with $\mathrm{SNR} = 12$ dB and the right figure is the accuracy for the same dataset with $\mathrm{SNR} = 16$ dB. It can be seen that the noise degrades the system's performance but for the same system operation point (β, γ), the dynamics and performance are similar. That is, for the same dataset, if a parameter set has been proven to be able to work efficiently at a one noise level, it in general works well at the same dataset with a different SNR.

6.3 Datasets and Simulation Parameters

6.3.1 Introduction to Dataset and Preprocessing

In this article, we will use four datasets widely used by the research community, namely Nonlinear Auto Regressive Moving Average (NARMA10) dataset, Nonlinear-Channel Equalization (NCE) dataset, Radar Signal Forecasting dataset, and Santa Fe Laser Series Prediction dataset to train and test the DL-OEO-based reservoir computers.

6.3.1.1 NARMA10 Dataset

Nonlinear Auto Regressive Moving Average (NARMA10) [146] task is widely used to test the performance of RC. As a regression task, it specifically tests the RC's ability to memorize long-term information. The recurrence of trait for NARMA10 is given by following equation:

$$\begin{aligned} y(n+1) = 0.3y(n) + 0.05y(n)\sum_{i=1}^{9} y(n-i) \\ + 1.5u(n-9)u(n) + 0.1, \end{aligned} \tag{6.8}$$

where $u(n)$ is a random variable (independent and identically distributed, or i.i.d) drawn from a uniform distribution in $[0, 0.5]$. The task is to predict $y(n+1)$ given previous $y(n)$. Performance metrics of this task is characterized by Normalised Mean Square Error (NMSE) defined as follows:

$$NMSE = \frac{\langle (\mathbf{Y} - \hat{\mathbf{Y}})^2 \rangle}{\langle (\mathbf{Y} - \langle \mathbf{Y} \rangle)^2 \rangle} \tag{6.9}$$

where $\langle\ \rangle$ is the expectation operator, $\mathbf{Y}$ is the time series of the ground truth the $\hat{\mathbf{Y}}$ is the predicted time series from the machine learning algorithm.

6.3.1.2 Non-linear Channel Equalization dataset

The nonlinear channel equalization task was first introduced in [147]. Random i.i.d. token sequence d(n) drawn from $\{-3, -1, 1, 3\}$ first goes through a linear channel, which yields:

$$\begin{aligned} q(n) = 0.08d(n+2) - 0.12d(n+1) + d(n) + 0.18d(n-1) \\ - 0.1d(n-2) + 0.91d(n-3) - 0.05d(n-4) \\ + 0.04d(n-5) + 0.03d(n-6) + 0.01d(n-7), \end{aligned} \tag{6.10}$$

Then, the signal goes through a noisy nonlinear channel, which gives the distorted output sequence as follows:

$$u(n) = q(n) + 0.036q(n)^2 - 0.011q(n)^3 + \sigma(n)^2, \tag{6.11}$$

where $\sigma(n)$ is an i.i.d. Gaussian noise with zero mean and signal-to-noise ratio (SNR) ranges from 12 dB to 32 dB with a step of 4 dB. The task is to predict the original input signal $d(n)$ from the output $u(n)$ at the noisy non-linear channel. This task is defined as a classification task, and its performance is assessed by the prediction accuracy, which is the ratio of correctly predicted tokens to the total number of tokens being predicted.

6.3.1.3 Radar Signal Forecasting Dataset

This task is to predict a radar signal from one to ten steps in the future. The radar signal is collected by the McMaster University IPIX [148] radar from the ocean surface under a low sea state with an average wave height of 0.8 m and a maximum 1.3 m. The dataset itself contains in-phase and in-quadrature of radar output after demodulation. This task could also be categorized into regression problems and the metric is NMSE.

6.3.1.4 Santa Fe Laser Series Prediction Dataset

This task is one of the 6 datasets utilized in the "Time Series Prediction Task" sponsored by Santa Fe Institute. The signal is extracted from a chaotic laser. The training set contains 3000 samples and the test set contains 1000 samples. The task is to predict the future intensity of the chaotic laser given previous n values of the laser output. Similar to NARMA10, this is also a regression problem and the metric used to measure RC's performance is NMSE.

In order to train and test our DL-OEO-based RC, the signals are extracted from above four mentioned datasets and represented as discrete points of sampled time series. Following Sec. 6.2.2, the discrete data is then processed into a continuous signal $J(t)$ and multiplied by a mask signal $m(t)$ to generate $u_I(t)$ to simulate Eq. (6.2). It is worth mentioning that Eq. (6.2) is normalized with a reference voltage $V_{\mathrm{RF}} = 1$ V, leading a normalized variable $x(t)$ has the same numerical value of voltage $V(t)$. For this reason, the numerically simulated injected signal value $\rho u_I(t)$ could be easily converted into the real injection current $I(t)$ coupled to modulated the LD by dividing the characteristic impedance $R_z = 50\ \Omega$. For our case, the corresponding current $I(t)$ used to internally modulate the laser for an injection amplification factor $\rho \approx 1$ is in an energy-efficient range of several mAs.

6.3.2 Simulation Parameters

For the numerical simulation of the DL-OEO-based RC using Eq. (6.2), the following parameters are applied: the optical fiber has a length of 40 m, introducing into the DL-OEO system a delay time of $T = 0.2\ \mu$s. The number of VNs is set as $N = 50$, leading to a time separation between VNs with $\Delta T = T/N = 4$ ns and corresponding to an internal modulation frequency of LD equals to $1/\Delta T = 250$ MHz. The regulation term λ for the Ridge regression algorithm is set as $\lambda = 0.0001$. The two timescales of high cut-off frequency and low cut-off frequency are set as $\tau = 25$ ns and $\theta = 0.3\ \mu$s, respectively. The offset normalized voltage is set as $\alpha = -0.5$. The choice of parameters β and ρ are studied in Section 6.4.

6.3.2.1 NARMA10

The training set of NARMA10 contains a total of 3000 training examples and the test set contains a total of 6000 test examples. The DL-OEO-based RC is trained and tested to predict the time sequence from 1 to 10 steps ahead, respectively.

6.3.2.2 Non-linear Channel Equalization

In this task, we trained our RC and tested it on different SNRs from 12 dB to 32 dB with a separation of 4 dB. The training set contains a NCE time series of 3000 training examples and the test set contains a total of 6000 test examples.

6.3.2.3 Radar Signal Forecasting

For this dataset, the training set contains 3000 training examples and the test set contains 6000 test examples.

6.3.2.4 Santa Fe Laser Series Prediction

The training set of this dataset contains 3000 training examples and the test set contains 1000 examples.

6.4 Simulation Results

6.4.1 System Gain and Injection Amplification factor

The DL-OEO under study has been proven in [145] being able to operate in different dynamic regions like fast scale oscillation, slow scale oscillation, and even chaos. Its bifurcation can be controlled by tuning the parameter β. Another parameter that is closely related to the DL-OEO based RC's performance is the amplification factor of injection signal ρ - a parameter that determines the strength of the signal being injected to excite the VNs inside the reservoir layer. We scan through the system gain β from 0.7 to 1.6 together with injection amplification factor ρ from 0.2 to 3. The RC based on the DL-OEO is trained and tested on the four datasets mentioned in Section 6.3, respectively.

It can be seen from Fig. 6.2 that the DL-OEO-based RC's performances are sensitive to the injection amplification factor ρ. For all four datasets, the DL-OEO-based RC performs poorly when the injection amplification factor ρ is too low ($\rho = 0.2, 0.3$). This can be explained by the fact that a weak injection of signal can hardly have any impact on the OEO's internal dynamics, making the information injected into the OEO too weak for the VNs to actually contain much information from the dataset. Meanwhile, a larger injection amplification factor – corresponding to a stronger injection current could also deteriorate the DL-OEO-based RC's performance. This could be seen in the region for smaller system gain $\beta < 1.2$

for all four datasets when injection amplification factor ρ is increased to the level above 2. This can be explained by that the DL-OEO-based RC is operating before the first bifurcation point in this region – a region that has been proven not to work well for the purpose of reservoir computing in traditional OEO-based RC [149]. From the perspective of dynamics, our result shows a similar trend of performance of the DL-OEO-based RC compared to the traditional OEO-based RC despite the fact that it has a completely different nonlinear transfer function and dynamical parameters. It can also be observed that both types of OEO-based RC get their optimal performance after the first bifurcation. For our simulations, the DL-OEO-based obtains the best performance for all four datasets with a system gain $1.2 < \beta < 1.6$, that is, $\beta = 1.24$ for NARMA10 and $\beta = 1.26$ for NCE and $\beta = 1.42$ for both Radar prediction and Santa Fe laser prediction tasks, where $\beta = 1.2$ and $\beta = 1.6$ are corresponding to first and second bifurcation points of the system.

6.4.2 Noise Level of the Injection Signal

It is important to assess the resilience of the DL-OEO-based RC against noise. Figure 6.3 shows two heatmaps of accuracy scanned across different injection amplification factor ρ and system gain β on the nonlinear channel equalization dataset. Figure 6.3(a) and Fig. 6.3(b) are corresponding to SNR of 12 dB and 16 dB, respectively. Together with Fig. 6.2(b) that has a SNR of 32 dB, it can be seen that the increment of noise level will deteriorate the DL-OEO-based RC's performance – the best accuracy achieved by the DL-OEO for $\mathrm{SNR} = 12$ dB is 87.82% and 94.72% for $\mathrm{SNR} = 16$ dB. For $\mathrm{SNR} = 32$ dB, the accuracy could achieve a 99.52% accuracy.

It is worth noting that all of these three heatmaps show a similar performance trend regarding the dynamic regions. In other words, regardless of noise level, the system gain β that has a direct impact on the dynamics regime, together with injection amplification factor ρ, determine the DL-OEO-based RC's learning capability on a specific dataset. For the nonlinear channel equalization, if one parameter set (β, γ)

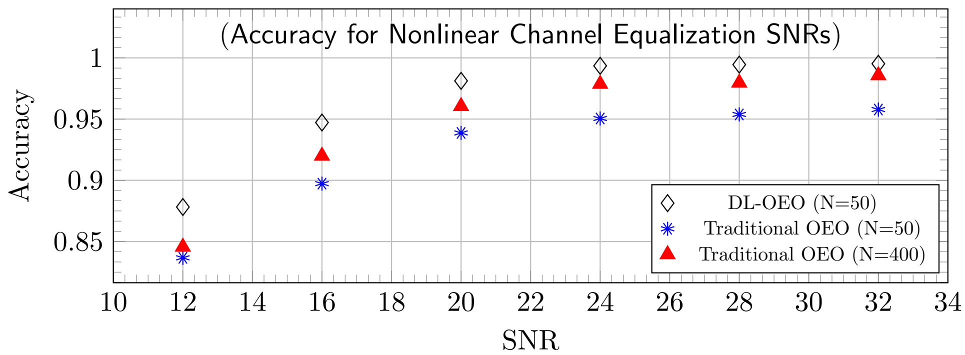


Figure 6.4: Accuracy for the Nonlinear Channel Equalization dataset across different SNRs. The DL-OEO-based RC has shown a better performance even when increasing the baseline model to a total of 400 VNs.

of the DL-OEO-based RC system's operation point has been proven to work efficiently at one noise level, it tends to work efficiently for the same dataset even when the channel condition varies the SNR. Moreover, it can be seen from Fig. 6.4 that before the SNR reaches a threshold of 20 dB, the DL-OEO-based RC maintains a high level of accuracy above 99%. This could be interpreted as that for low-level noise under a power that is less than 1% of signal, the noise has nearly no negative impact on the DL-OEO-based RC's performance. Considering the fact that a threshold of noise level with $\mathrm{SNR} = 20$ dB is not difficult to realize by modern communication systems, which typically have much better SNR, around 30 dB or even above 40 dB. Therefore, we can safely conclude that the DL-OEO-based RC's performance is sufficiently resilient to noise and its performance under noise is pretty reliable.

6.4.3 Comparing the DL-OEO-based RC's Performance with a Baseline Model

In this section, we will compare our DL-OEO-based RC's performance with an optimized traditional broadband OEO-based RC using an external MZM modulator. Following physical parameters in [90], we numerically simulate a traditional broadband OEO-based RC with a delay time of 20.82 μs and use it as the baseline model. First, to gain a better understanding of the differences between the two types of OEO-based RCs, we test their memory capacity (MC) with 50 VNs. Then, we optimize the performance of the

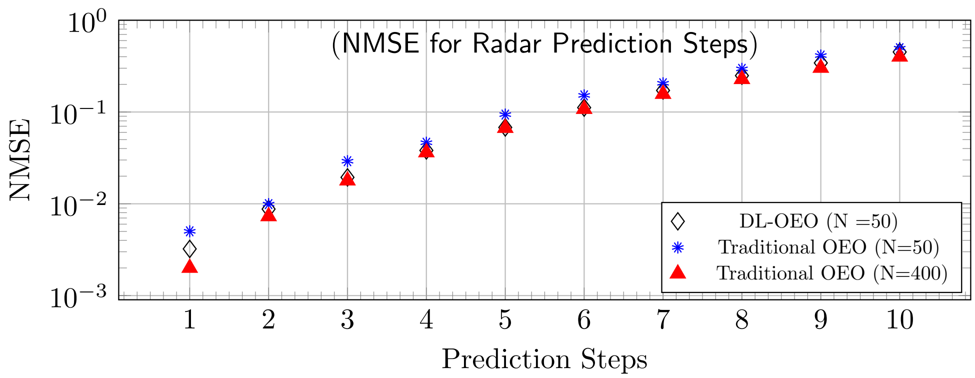


Figure 6.5: Accuracy for the Radar Prediction dataset. The DL-OEO-based RC performs better than the baseline model with 50 VNs in all prediction steps. It remains competitive when increasing the baseline model to 400 VNs.

traditional broadband OEO-based RC over all four datasets and compare it with our DL-OEO-based RC. It can be seen that with an equal number of VNs ($N = 50$), the DL-OEO-based RC excels the baseline model on all four datasets. Lastly, we further increase the total number of VNs of the baseline to 400 and show that DL-OEO-based RC still maintains a better or comparable performance with fewer VNs.

6.4.3.1 Memory Capacity

Memory Capacity (MC) is a task-independent metric widely used to test the RC's ability to memorize temporal information. Initially proposed in [71], linear memory capacity (LMC, often written as MC_1) measures the RC's ability to reconstruct previous inputs k steps ahead based on the current input. Specifically, the task tests RC's ability to reconstruct previous k steps of an input $y_k(n) = u(n-k)$ using current input $y_0(n) = u(n)$, where $u(i)$ is random variable i.i.d drawn from a continuous uniform distribution in the interval of -11. The capacity for a certain step k is defined as:

$$C[y_k] = 1 - NMSE[y_k]. \tag{6.12}$$

It has been demonstrated in [150] that the maximum memory capacity of an RC to track temporal information is limited by its total number of VNs. Thus, LMC is defined as the sum of the capacity of all the previous steps of k from 1 to N:

$$MC_1 = \sum_{k=1}^{N} C[y_k]. \tag{6.13}$$

To generalize LMC to higher-order, the RC is required to reconstruct higher-order Legendre polynomial of previous k inputs defined as:

$$y_k^i = P_i(y_k), \tag{6.14}$$

where $P_i(y_k)$ is i^{th} order Legendre polynomial of y_k. For example, when $i = 2$, $P_i(y_k)$ is calculated as $P_2(y_k) = 3y_k^2 - 1$. Similarly, the i^{th} order of MC is defined as the sum of capacity to reconstruct y_k^i, i.e.,

$$MC_i = \sum_{k=1}^{N} C[y_k^i]. \tag{6.15}$$

The total memory capacity (MC_{tot}) is defined as the sum of MC from 1^{st} to the highest order considered:

$$MC_{tot} = \sum_{i=1}^{M} MC_i. \tag{6.16}$$

Following [151], we collect 1000 samples from a continuous uniform distribution and use 800 samples and 200 samples to train and test the OEO-based RCs. We measure the linear memory capacity MC_1 and second-order memory capacity MC_2 for both types of OEO-based RCs and use them to determine the total memory capacity MC_{tot}. The non-linearity of the OEO-based RCs is decided by the operating dynamic regions of the OEOs as well as their non-linear transfer functions. To reduce the difference caused by dynamic regions, we set both types of OEOs operating at and after the Hopf-bifurcation point, which have been proven to work efficiently for OEO-based RCs. This leads to a choice of feedback loop gain β as shown in TABLE 6.1.

Table 6.1: Memory Capacity of DL-OEO-based RC and traditional broadband OEO-based RC ($N = 50$)

Metrics	**DL-OEO**		**traditional OEO**	
	$\beta = 1.18$	$\beta = 1.32$	$\beta = 1$	$\beta = 1.32$
MC_1	8.37	6.32	7.89	5.77
MC_2	2.17	1.19	1.67	0.69
MC_{tot}	10.54	7.51	9.56	6.46

It can be seen that DL-OEO-based RC has a slightly better MC in all orders compared with traditional OEO-based RC operating at or after the Hopf-bifurcation point, respectively. It has been proven in [152] that there is a universal trade-off between non-linearity and memory capacity for RCs. So our finding here is expected since the DL-OEO-based RC has a much simpler non-linear transfer function.

6.4.3.2 NARMA10 Dataset

The best performance obtained for NARMA10 on the baseline model with a VN number of 50 is $\mathrm{NMSE} = 0.3107$. By increasing the total number of VNs to 400, the performance of the traditional OEO-based RC improves to $\mathrm{NMSE} = 0.2703$. Meanwhile, the proposed DL-OEO-based RC performs better than the baseline model with both VN settings with a $\mathrm{NMSE} = 0.2003$. As shown in [153], the NARMA10 task often demands an RC operating near the linear regime with a higher MC. Despite the different dynamics inside the systems, our results confirm their findings as the DL-OEO-based RC obtains a better performance with a more linear transfer function and a better MC. This result is technologically important since the (ReLU) [154] function has been widely accepted by the deep learning community as a better activation function for convolutional neural networks (CNNs) [25] than other nonlinear functions including the sigmoid function or higher order trigonometric functions $\cos^2$ - the typical nonlinear transfer function introduced into traditional OEOs by external modulators [143]. Except for being more linear-like, another benefit of the ReLu function is that it does not saturate at large input values and provides sparsity for training - effectively reducing the risk of overfitting the data. The wide acceptance of the ReLu function is also tested through years of research and empirical applications [155, 156] while other more complex nonlinear functions that have been proposed only have limited gain on certain dataset [157].

6.4.3.3 Nonlinear Channel Equalization Dataset

Figure 6.4 shows the accuracy curve across different SNRs for the NCE dataset with SNRs from 12 dB to 32 dB with a step of 4 dB. The diamond marks, star marks, and triangle marks correspond to the DL-OEO-based RC with 50 VNs, and the traditional OEO-based RC with 50 and 400 VNs, respectively. It can be seen that the DL-OEO-based RC has higher classification accuracy over the traditional broadband OEO-based RCs across all the different SNRs regardless of the number of VNs. Meanwhile, it can be seen that both types of OEO-based RCs' performance saturate at $\mathrm{SNR} = 24$ dB or greater while it still improves significantly from $\mathrm{SNR} = 12$ dB to $\mathrm{SNR} = 20$ dB. The better performance of DL-OEO-based RC also suggests that it has improved its resilience to noise compared with traditional OEO-based RC by using the ReLU nonlinear activation.

6.4.3.4 Radar Prediction Dataset

Figure 6.5 shows the measured NMSE for the IPIX radar time series prediction task with low state settings for both the DL-OEO-based RC and the baseline model. It can be seen that DL-OEO-based RC performs better than traditional OEO-based RC at all different prediction steps when they share an identical number of VNs with $N = 50$. By further increasing the total number of VNs of the baseline model to 400, the traditional OEO-based RC demonstrates a slightly smaller NMSE compared to the DL-OEO-based RC for the one step ahead prediction task. Meanwhile, for the prediction that involves 2 to 9 steps, the DL-OEO-based RC demonstrates equivalent performance to traditional OEO-based RC with a much less number of VNs. The fact that DL-OEO-based RC excels the traditional OEO-based RC with an equal number of VNs but could no longer beat the traditional OEO-based RC with 400 nodes like the rest three benchmarks agrees with analysis in [153] that the IPIX radar task's performance is less relevant to MC.

6.4.3.5 Santa Fe Laser Prediction Dataset

For this dataset, the best NMSE obtained by the baseline model with 50 nodes and 400 nodes are 0.188 and 0.095, while the best NMSE achieved by the DL-OEO-based RC is 0.0563. Here again, we note that the DL-OEO-based RC with piece-wise nonlinear transfer function has shown superior performance to the traditional broadband OEO-based RC. Since the predicted time series is generated by a laser operating at a weakly chaotic region near oscillation, our results suggest that memory capacity is more important for this task.

6.5 Conclusion

In this chapter, we have proposed a scheme of photonic implementation of reservoir computers using the DL-OEO. We proved that this photonic machine learning platform, with one of the simplest architectures possible for an OEO using a piece-wise nonlinear transfer function, is a performant configuration for RC. We have shown that it not only greatly reduces the overall cost, but also maintains a high level of performance, comparable to or even better than traditional broadband OEO-based RCs on the most widely used four benchmarks for RC. By removing the bulky external $\mathrm{LiNbO_3}$ MZM modulator, and simultaneously introducing a short delay line, this scheme has the benefit of a much compact size and greatly reduced cost. It also opens the gate for the next generation of OEO-based RC which has the advantage of being able to perform machine learning at ultra-fast speed using photonic devices. It also offers silicon compatibility for potential integration. Its compact size, reduced cost and the potential for silicon integration make it a budget option for the implementation of chip-size photonic ML hardware at the physical level. Finally, the DL-OEO ability to process both RF and optical signals makes it an ideal ML platform with great compatibility for radio-over-fiber communication systems to build large-scale smart communication networks and applications, as a cost-effective and energy-efficient solution.

Chapter 7: Conclusion

The Optoelectronic Oscillator is a versatile device that is widely in modern communication systems - featuring as a distinctive component that is able to handle both optical and RF signals with wide applications, including but not limited to generating ultra-low phase noise signals, chaos communication, and sensing technologies. In recent years, OEO-based RCs have been proven to be a simple and efficient ML platform with ultra-low latency and are suitable for applications in communication networks. However, all previously reported OEO-based RCs are based on wideband OEOs that are not suitable for directly processing I-Q modulated signals. Meanwhile, the narrowband OEO, which was used to generate ultra-pure microwave, has shown the capacity to process modulated signals, but has been ignored by the photonic machine learning community.

In this thesis, for the first time, we proposed a narrowband OEO-based RC in the world to process I-Q modulated signals. Firstly, we derived the envelope equation that is needed to numerically simulate such a system with an injection signal. Then, we demonstrate its ability to reach state-of-the-art performance on I-Q modulation classification tasks. We show that with a reduced training set size, this narrowband OEO-based RC serves as an efficient ML platform for the classification of I-Q modulated signals that does not require demodulating and pre-processing of the signal, which paves the road for low-latency and real-time processing of the I-Q modulated signals. Following the I-Q modulation tasks, we then move to another application field of such systems - RF fingerprinting. Here we tested the narrowband OEO-based RC on several datasets. We find that with only about 0.1 % to 1% of the data amount that is required to train a CNN, our narrowband OEO-based RC can still reach a similar performance to that of the CNN's.

In the experiment, we further push the narrowband OEO-based RC to operate with the lowest possible virtual nodes and training dataset size. Meanwhile, in order to compare the performance across different machine learning algorithms, we propose a new metric η that takes into account the total number of nodes, training set size, and accuracy. This metric helps us to quantitatively analyze the diminishing returns of the machine learning algorithms of increasing dataset size and computational resources.

In Ch. 6, we have proposed a scheme for the photonic implementation of reservoir computer that first uses the concept of directly modulated OEOs. We demonstrated that this photonic machine learning platform - based on one of the simplest possible OEO architectures with a piecewise nonlinear transfer function, offers strong performance for reservoir computing. Our results show that it not only significantly reduces overall cost but also delivers high performance, comparable to or even surpassing that of traditional broadband OEO-based RC systems on the four most widely used benchmark tasks.

Currently, the total number of virtual nodes required for time-delay OEO-based RC has remained unclear. In Appendix. II, we performed numerical simulations of both types of narrowband OEO-based RC and wideband OEO-based RC across a wide range of datasets. For the first time, we find that the total number of virtual nodes N required for reservoir computing can be as little as the total number of cavity modes $N_{CM} = \frac{f_H - f_L}{f_{FSR}}$ inside the cavity. This finding offers insight into the choices of N that has been ignored by the research community for a long time.

Appendix I: Time-Division Multiplexing and Numerical Simulation Methods

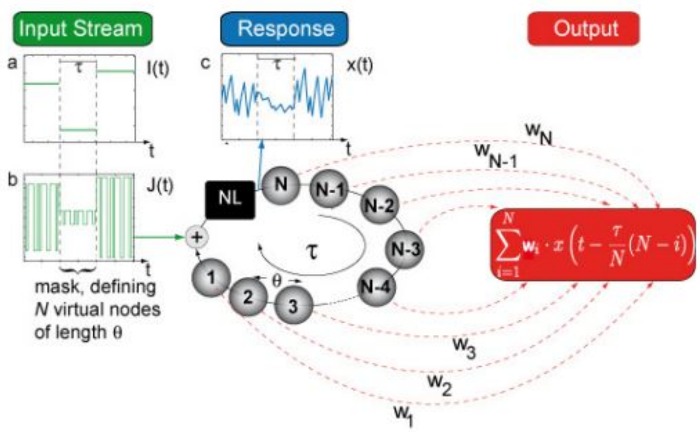


Figure I.1: The Concept of time-division multiplexing. This figure is reproduced from ref. [6].

We have already introduced the concepts of time-division multiplexing in Ch. 3. Here, we introduce the numerical simulation methods for implementation of delay line based RC. Following [6], we write a time delay system with a delay time as τ (which corresponds to T in our OEO-based systems). It can be seen that each sample in the input data is produced as stream $I(t)$ with a holding time of τ. The mask values are defined as a time series where each value will hold $\theta = \tau/N$. This interval of θ also corresponds to a virtual nodes in the temporal domain.

To numerical simulate the time-delayed differential equation for RCs, we can use any classical numerical methods like Euler's method or Runge-Kutta method.

Given an initial value problem of the form:

$$\frac{dy}{dt} = f(t,y), \quad y(t_0) = y_0 \tag{I.1}$$

The formula of Euler's method is given as follows

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{I.2}$$

where h is time step for updating equations. Meanwhile, we can also write 4^{th} order Runge-Kutta method as follow

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right), \\ k_4 &= f(t_n + h, y_n + hk_3), \\ y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{I.3}$$

It can be seen that both these two numerical methods can be readily adapted to simulate the narrowband OEO-based RC with an envelop equation

$$\dot{\mathcal{A}} = -\mu\mathcal{A} - 2\mu\gamma\mathrm{Jc}_1[2|\mathcal{Z}|]\mathcal{Z} \tag{I.4}$$

since Eq. I.4 has already been in a differential format.

Meanwhile, it is worth noting that time-delay differential equations for a broadband OEO-based RC has the following format

$$x(t) + \tau\frac{dx(t)}{dt} + \frac{1}{\theta}\int_{t_0}^{t} x(\xi)d\xi = \beta D[x_T + \rho u_I(t) - \alpha], \tag{I.5}$$

which has an integral part inside, making direct simulation use these two methods impossible.

One trick to solve this issue is by transforming the integral part of Eq. I.5 into differential format is to set $y(t) = \int_{t_0}^{t} x(\xi)d\xi$ to transform it into a second-order time-delayed differential equations. Then, we can introduce a new variable $z(t) = \frac{dy}{dt}$ to reduce the second-order time-delayed differential equation into two coupled first-order time-delayed differential equations that can be directly simulated using Eq. I.3.

In our research, we often use the Euler's method for a fast test of an new idea, but always report our final results with 4^{th} order Runge-Kutta Method for better simulation accuracy.

Another issue is related to the time scale. It is always important to determine the timestep h inside the numerical simulation that maintain a level of accuracy within a reasonable running time. During the simulations, we must ensure that the timestep is much smaller than the minimum timescale inside the equations. For simulations with large size of virtual nodes N, we can choose h as θ divided by an odd number (for example 5, so that the middle point can be easily sampled to represent the virtual nodes). In this case, since N is large, θ often serves as the minimum timescale in the equation and the computation is safe. But if N is not large, the timestep h must be chosen carefully to avoid mistakes.

Meanwhile, due to property of time-delayed differential equations, the numerical simulation must contain all the information from the delay-line in the previous delay time T which are used as initial conditions for current round of numerical simulation. This can make numerical simulation be RAM expensive since all the values of T/h has to stored.

Another caveat is about the initial conditions of the delay line x_T in simulations. To start the simulation, a small value will work fine. In general, for most of the datasets, in order to avoid transients in the OEO, we will injected random signal into the OEO-based RCs that typically last for several delay time T and do not record any information. This process often serves as a preheat step for the OEO-based RCs that ensures when the OEO-based RC is in working conditions when real data points are injected.

Appendix II: Exploring the Relationship between Cavity Modes and Number of Virtual Nodes for OEO-based Time-Delay Reservoir Computer

In previous chapters, we have discussed the choice of parameters like feedback loop gains β (for DL-OEO RC), γ (for narrowband OEO-based RC), and injection strength ρ and their impacts to the performances of OEO-based RCs. We show the process of optimizing them to improve OEO-based RCs' performance. We have also proposed a new metric η to illustrate the possible diminished returns of performance regarding the number of virtual nodes N and the training set size T. However, there is another important question regarding the number of virtual nodes N that has not been answered: how many nodes N will be sufficient for OEO-based RC?

This question has been overlooked by the research community of photonic reservoir computing using a time-delay system. The reason for such an oversight is quite natural: for a digital implementation of RCs with circuits, an increment in the total number of virtual nodes N typically would lead to more computational power. The theory for the three fundamental properties of reservoir computers has already been formalized in ref. [73]. Therefore, researchers in the time-delayed RC have taken this conclusion as granted while ignore the intrinsic physical properties of the new systems. In most of the publications regarding the choice of N, we have seen several default settings of N. For example, one might choose $N = 400$ following most of the publications. Sometimes, N is increased to a few thousand for better performance, while it might be reduced to 50 for simplicity. In all previous publications, we have not seen a thorough analysis of the best number of N for OEO-based RCs.

In this chapter, we numerically simulate both types of OEO-based RCs, i.e., the broadband OEO-based RCs and narrowband OEO-based RCs, across different datasets to show that the total number of virtual nodes required by OEO-based RCs is directly related to the total number of RF cavity modes inside the OEO. Specifically, we will show that for both types of OEO-based RCs, the number of cavity modes inside the OEO will serve as a good cut-off point for the choices of N. Beyond this point, the further increments of N has very limited gain to the performance of OEO-based RCs. Due to the nature of this thesis, we will only present numerical simulation results here. Our thorough examination and development of the theory that fully explains this phenomenon will be published in the future.

II.1 Numerical Results on Broadband OEO-based RCs

The broadband OEO-based RC under examination here relies on a broadband OEO introduced in Ch. 2 with a configuration shown in figure 2.4. The equations that govern the behavior of this type of OEO-based RCs is given by the following equation

$$x(t) + \tau \frac{dx(t)}{dt} + \frac{1}{\theta} \int_{t_0}^{t} x(\xi) d\xi = \beta \cos^2[x_T + \rho u_I(t) + \phi], \tag{II.1}$$

where $\theta = 1/2\pi f_{\mathrm{L}}$ and $\tau = 1/2\pi f_{\mathrm{H}}$, f_L and f_H are corresponding to the low-cut off frequency and high cut-off frequency of the RF filter, respectively. The free spectral range (FSR) of the delay line is given by the following equation:

$$f_{FSR} = 1/T, \tag{II.2}$$

where T is the delay time of the optical fiber. The total number of cavity modes can be determined by the following equation:

$$N_{CM} = \frac{f_H - f_L}{f_{FSR}} \tag{II.3}$$

In the numerical simulation, we choose a fixed delay line that leads to a delay time of $T = 20.82\mu s$. In order to adjust the cavity modes supported within the bands of OEO, we change the filter frequencies to alter $f_H - f_L$, which leads to the choices of a set of cavity modes with $N_{CM} = 14, 35, 50, 70$. The other parameters are setting as $\phi = -0.65\pi$ and $\rho = 0.45$. We train and test this broadband OEO-based RC with the most commonly used four benchmarks for broadband OEO-based RC mentioned in Ch. 6, with the number of training examples and test examples being the exactly the same as specified in Sec. 6.3.2. We train and test the broadband OEO-based RCs by scanning through a different number of virtual nodes N. Then, we normalize the N by dividing the number of cavity modes to get the normalized virtual nodes defined as

$$R = \frac{N}{N_{CM}}, \tag{II.4}$$

with $R = 1$ corresponding to a total number of virtual nodes equal to the supported cavity modes $N = N_{CM}$.

Here, we use the error rate E rather than accuracy to measure the performance of nonlinear channel equalization tasks with

$$E = 1 - Accuracy. \tag{II.5}$$

The performance of broadband OEO-based RC is shown in figure II.1. It can seen that for cavity modes equal to 14, the NMSE (Error Rate) reduced significantly when increasing the number of virtual nodes N before $R = 1$ for all four datasets. Meanwhile, further increments of the virtual nodes' number beyond $R = 1$ contribute little to improving broadband OEO-based RCs' performance. This conclusion holds valid when we change the number of cavity modes N_{CM} from 14 to 35, 50, and 70.

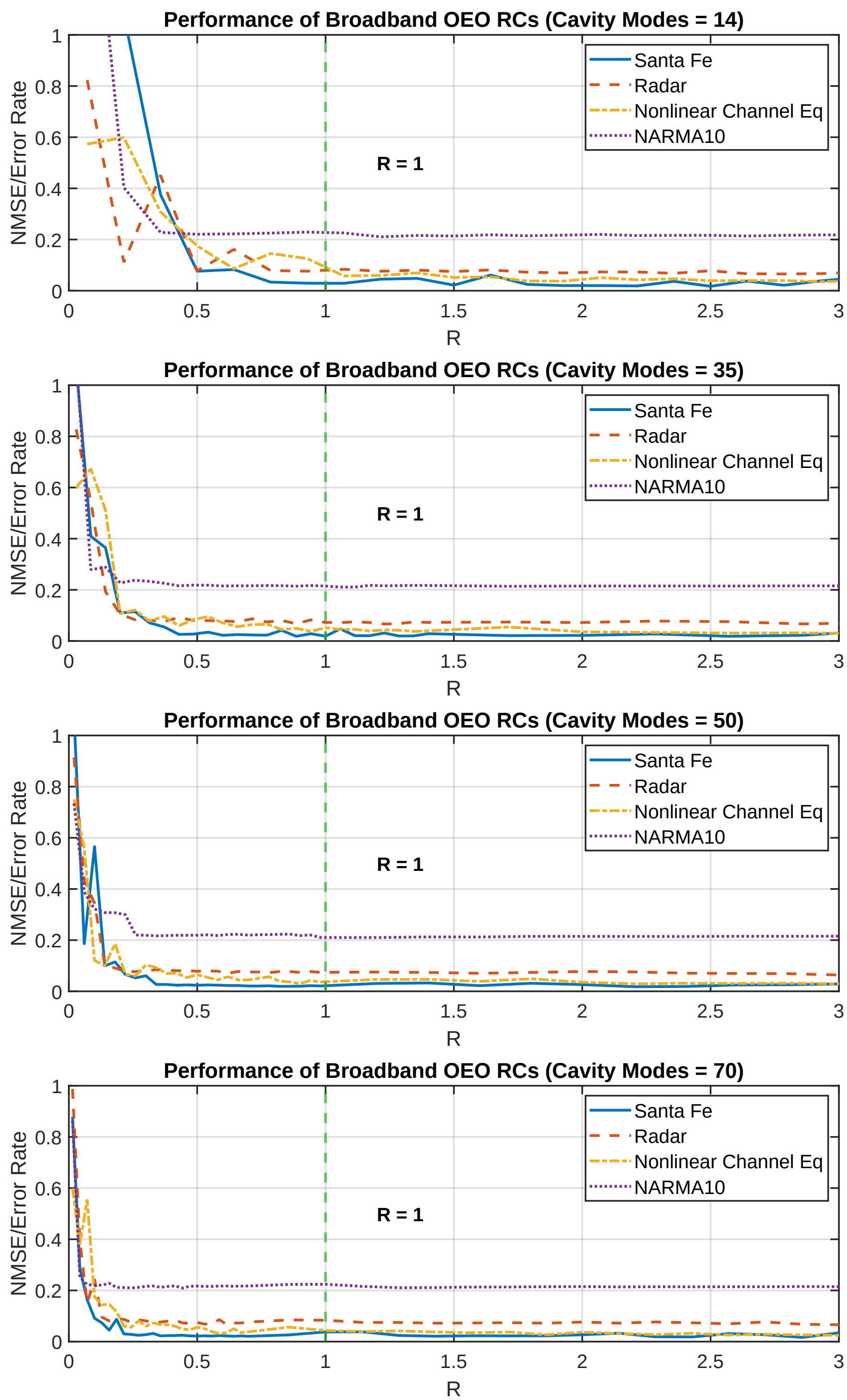


Figure II.1: Simulation results of broadband OEO-based RCs' performance across different cavity modes $N_{CM} = 14, 35, 50, 70$ in all four different datasets: the Santa Fe Laser Prediction, IPIX Radar prediction, Nonlinear Channel Equalization, and NARMA10. R is the normalized number of virtual nodes. It can be seen that the performance of broadband OEO-based RC increases fast before $R = 1$ and saturates at $R = 1$. Further increments of the virtual nodes' number beyond $R = 1$ contribute little to improving broadband OEO-based RCs' performance.

The implication of this study illustrates that for broadband OEO-based RCs, it might not actually need that much of virtual nodes N as suggested by most publications. Instead, a total number of virtual nodes N that surpasses the supported cavity modes N_{CM} inside broadband OEO is sufficient for the purpose of reservoir computing.

II.2 Numerical Results on Narrowband OEO-based RCs

Here, we use the IQ-modulation classification as a benchmark to see if the boundary condition $R = 1$ to choose the number of virtual nodes N still holds for a narrowband OEO-based RC. Following all the parameters in Sec. 3.6 with the number of training examples being 660. Similarly, we choose the number of cavity modes to be $N_{CM} = 100, 150, 200, 300$ and scan through different numbers of virtual nodes. It is worth noting that here we also use the error rate $E = 1 - Accuracy$ instead of $Accuracy$ for illustration purposes.

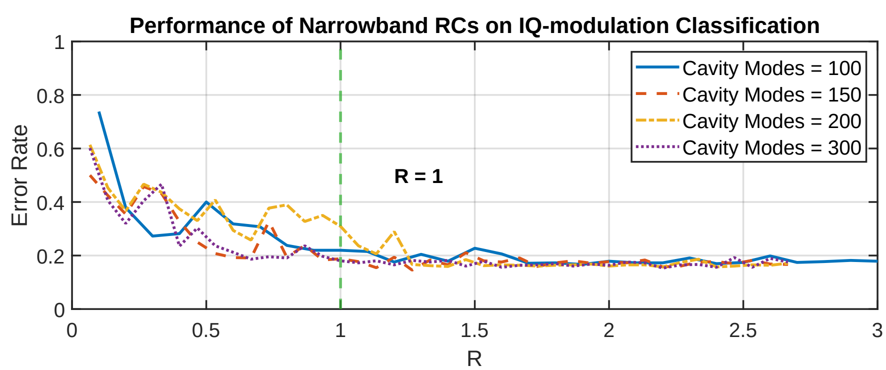


Figure II.2: Simulation results of narrowband OEO-based RCs' performance across different cavity modes $N_{CM} = 100, 150, 200, 300$ for the RADIOML 2016.04C dataset. R is the normalized number of virtual nodes. It can be seen that the error rate E of narrowband OEO-based RC decreases significantly when increasing the number of virtual nodes N before $R = 1$ and saturates at $R = 1$. Further increments of the virtual nodes' number beyond $R = 1$ only slightly reduce the classification error rate of narrowband OEO-based RC.

It can be seen from figure II.2 that $R = 1$ also serves as a critical condition for the total number of virtual nodes N. The error rate E of narrowband OEO-based RC decreases significantly when increasing the number of virtual nodes N before $R = 1$ and saturates at $R = 1$. Further increments of the virtual nodes' number beyond $R = 1$ only slightly reduce the classification error rate of narrowband OEO-based RC. This also concludes that for narrowband OEO-based RC, the total number of virtual nodes N will suffice when reaching the number of supported cavity modes N_{CM} inside the narrowband OEO.

Appendix III: Published Works

Journal Articles

- **Haoying Dai**, and Yanne K. Chembo. Classification of IQ-modulated signals based on reservoir computing with narrowband optoelectronic oscillators. *IEEE Journal of Quantum Electronics*, 57(3):1–8, 2021.
- **Haoying Dai**, and Yanne K. Chembo. RF fingerprinting based on reservoir computing using narrowband optoelectronic oscillators. *Journal of Lightwave Technology*, 40(21):7060–7071, 2022.
- Benjamin H. Klimko, **Haoying Dai**, and Yanne K. Chembo. Resource-constained narrowband optoelectronic oscillator-based reservoir computing for classification of modulated signals. *Optics Letters*, 49(13):3608–3611, 2024.
- **Haoying Dai**, and Yanne K. Chembo. Photonic Reservoir Computing Using Optoelectronic Oscillators With Direct Laser Modulation. *Journal of Lightwave Technology*, 2024.

Submitted Articles

- **Haoying Dai**, Benjamin H. Klimko, and Yanne K. Chembo, "Reservoir Computing with Nonlinear Delayed Photonic Systems: Opening the black box with nonlinear dynamics tools"

Bibliography

[1] Y. K. Chembo. Machine learning based on reservoir computing with time-delayed optoelectronic and photonic systems. *Chaos*, 30:013111, 2019.

[2] Yichen Shen, Nicholas C Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, et al. Deep learning with coherent nanophotonic circuits. *Nature photonics*, 11(7):441–446, 2017.

[3] Bin Shi, Nicola Calabretta, and Ripalta Stabile. Deep neural network through an inp soa-based photonic integrated cross-connect. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(1):1–11, 2019.

[4] Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Thach G Nguyen, Andreas Boes, Sai T Chu, Brent E Little, Roberto Morandotti, Arnan Mitchell, et al. Photonic perceptron based on a kerr microcomb for high-speed, scalable, optical neural networks. *Laser & Photonics Reviews*, 14(10):2000070, 2020.

[5] Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Andreas Boes, Thach G Nguyen, Sai T Chu, Brent E Little, Damien G Hicks, Roberto Morandotti, et al. 11 tops photonic convolutional accelerator for optical neural networks. *Nature*, 589(7840):44–51, 2021.

[6] Lennert Appeltant, Miguel Cornelles Soriano, Guy Van der Sande, Jan Danckaert, Serge Massar, Joni Dambre, Benjamin Schrauwen, Claudio R Mirasso, and Ingo Fischer. Information processing using a single dynamical node as complex system. *Nature communications*, 2(1):1–6, 2011.

[7] Yanne K Chembo, Daniel Brunner, Maxime Jacquot, and Laurent Larger. Optoelectronic oscillators with time-delayed feedback. *Reviews of Modern Physics*, 91(3):035006, 2019.

[8] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. Convolutional radio modulation recognition networks. In *International conference on engineering applications of neural networks*, pages 213–226. Springer, 2016.

[9] Yaneer Bar-Yam. *Dynamics of complex systems*. crc Press, 2019.

[10] Melanie Mitchell. *Complexity: A guided tour*. Oxford university press, 2009.

[11] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[12] Roger Penrose. *The emperor's new mind: Concerning computers, minds, and the laws of physics*. oxford university press, 2016.

[13] John McCarthy. What is artificial intelligence?, 2004. `https://cse.unl.edu/~choueiry/S09-476-876/Documents/whatisai.pdf`.

[14] Tom Stonier. *Information and the internal structure of the universe: An exploration into information physics*. Springer Science & Business Media, 2012.

[15] John Haugeland. *Artificial intelligence: The very idea*. MIT press, 1989.

[16] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[17] Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 479(480):104, 1969.

[18] Edward A Feigenbaum, Bruce G Buchanan, and Joshua Lederberg. On generality and problem solving: A case study using the dendral program. Technical report, 1970.

[19] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[20] Barbara Hammer and Kai Gersmann. A note on the universal approximation capability of support vector machines. *neural processing letters*, 17:43–53, 2003.

[21] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[22] Shunichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, (3):299–307, 1967.

[23] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[24] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[26] Frank Rosenblatt. Principles of neurodynamics. *Perceptrons and the theory of mechanisms*, 1962.

[27] Michael I Jordan. Serial order: A parallel distrmuted processing approach. 1986.

[28] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

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

[30] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[31] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[32] Hasim Sak, Andrew W Senior, Françoise Beaufays, et al. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, volume 2014, pages 338–342, 2014.

[33] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[34] 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.

[35] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[36] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PmLR, 2016.

[37] 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.

[38] AJ Cooper. 'fibre/radio'for the provision of cordless/mobile telephony services in the access network. *Electronics letters*, 26(24):2054–2056, 1990.

[39] Andreas Stohr, Ken-ichi Kitayama, and Dieter Jager. Full-duplex fiber-optic rf subcarrier transmission using a dual-function modulator/photodetector. *IEEE Transactions on Microwave Theory and Techniques*, 47(7):1338–1341, 1999.

[40] Anthony Nkansah, Anjali Das, Nathan J Gomes, Pengbo Shen, and David Wake. Vcsel-based single-mode and multimode fiber star/tree distribution network for millimeter-wave wireless systems. In *2006 International Topical Meeting on Microwave Photonics*, pages 1–4. IEEE, 2006.

[41] Pham Tien Dat, Atsushi Kanno, Toshimasa Umezawa, Naokatsu Yamamoto, and Tetsuya Kawanishi. Millimeter-and terahertz-wave radio-over-fiber for 5g and beyond. In *2017 IEEE Photonics Society Summer Topical Meeting Series (SUM)*, pages 165–166. IEEE, 2017.

[42] Fahad MA Al-Zubaidi, Juan Dayron López Cardona, D Sanchez Montero, and Carmen Vazquez. Optically powered radio-over-fiber systems in support of 5g cellular networks and iot. *Journal of Lightwave Technology*, 39(13):4262–4269, 2021.

[43] Kalman Wilner and Anthony P Van Den Heuvel. Fiber-optic delay lines for microwave signal processing. *Proceedings of the IEEE*, 64(5):805–807, 1976.

[44] S Sales, J Capmany, J Martí, and D Pastor. Solutions to the synthesis problem of optical delay line filters. *Optics letters*, 20(23):2438–2440, 1995.

[45] Jose Capmany, Joaquin Cascon, Jose Luis Martin, Salvador Sales, Daniel Pastor, and Javier Marti. Synthesis of fiber-optic delay line filters. *Journal of Lightwave Technology*, 13(10):2003–2012, 2002.

[46] W Zhang, JAR Williams, and I Bennion. Polarization synthesized optical transversal filter employing high birefringence fiber gratings. *IEEE Photonics Technology Letters*, 13(5):523–525, 2001.

[47] V Polo, B Vidal, JL Corral, and J Marti. Novel tunable photonic microwave filter based on laser arrays and n/spl times/n awg-based delay lines. *IEEE Photonics Technology Letters*, 15(4):584–586, 2003.

[48] Jianping Yao and Qing Wang. Photonic microwave bandpass filter with negative coefficients using a polarization modulator. *IEEE Photonics Technology Letters*, 19(9):644–646, 2007.

[49] Ningsi You and Robert A Minasian. A novel tunable microwave optical notch filter. *IEEE Transactions on Microwave Theory and Techniques*, 49(10):2002–2005, 2001.

[50] SL McCall, HM Gibbs, and TNC Venkatesan. Optical transistor and bistability (a). *Journal of the Optical Society of America (1917-1983)*, 65:1184, 1975.

[51] Hiroyuki Tsuda and Takashi Kurokawa. Construction of an all-optical flip-flop by combination of two optical triodes. *Applied physics letters*, 57(17):1724–1726, 1990.

[52] Martin T Hill, Harmen JS Dorren, Tjibbe De Vries, Xaveer JM Leijtens, Jan Hendrik Den Besten, Barry Smalbrugge, Yok-Siang Oei, Hans Binsma, Giok-Djan Khoe, and Meint K Smit. A fast low-power optical memory based on coupled micro-ring lasers. *nature*, 432(7014):206–209, 2004.

[53] Marin Soljačić and John D Joannopoulos. Enhancement of nonlinear effects using photonic crystals. *Nature materials*, 3(4):211–219, 2004.

[54] Takasumi Tanabe, Masaya Notomi, Satoshi Mitsugi, Akihiko Shinya, and Eiichi Kuramochi. Fast bistable all-optical switch and memory on a silicon photonic crystal on-chip. *Optics letters*, 30(19):2575–2577, 2005.

[55] Eiichi Kuramochi, Kengo Nozaki, Akihiko Shinya, Koji Takeda, Tomonari Sato, Shinji Matsuo, Hideaki Taniyama, Hisashi Sumikura, and Masaya Notomi. Large-scale integration of wavelength-addressable all-optical memories on a photonic crystal chip. *Nature Photonics*, 8(6):474–481, 2014.

[56] Jinlong Xu, Chi Zhang, Yulin Wang, Mudong Wang, Yanming Xu, Tianqi Wei, Zhenda Xie, Shiqiang Liu, Chao-Kuei Lee, Xiaopeng Hu, et al. All-in-one, all-optical logic gates using liquid metal plasmon nonlinearity. *Nature Communications*, 15(1):1726, 2024.

[57] Jianji Dong, Xinliang Zhang, and Dexiu Huang. A proposal for two-input arbitrary boolean logic gates using single semiconductor optical amplifier by picosecond pulse injection. *Optics Express*, 17(10):7725–7730, 2009.

[58] Claudia Hössbacher, Arne Josten, Benedikt Baeuerle, Yuriy Fedoryshyn, Horst Hettrich, Yannick Salamin, Wolfgang Heni, Christian Haffner, Christoph Kaiser, Rolf Schmid, et al. Plasmonic modulator with¿ 170 ghz bandwidth demonstrated at 100 gbd nrz. *Optics Express*, 25(3):1762–1768, 2017.

[59] Daniel Pérez, Ivana Gasulla, Lee Crudgington, David J Thomson, Ali Z Khokhar, Ke Li, Wei Cao, Goran Z Mashanovich, and José Capmany. Multipurpose silicon photonics signal processor core. *Nature communications*, 8(1):636, 2017.

[60] Philip Y Ma, Alexander N Tait, Thomas Ferreira de Lima, Chaoran Huang, Bhavin J Shastri, and Paul R Prucnal. Photonic independent component analysis using an on-chip microring weight bank. *Optics express*, 28(2):1827–1844, 2020.

[61] Kostas Sozos, Stavros Deligiannidis, Charis Mesaritakis, and Adonis Bogris. Unconventional computing based on four wave mixing in highly nonlinear waveguides. *IEEE Journal of Quantum Electronics*, 2024.

[62] Konstantin Kravtsov, Mable P Fok, David Rosenbluth, and Paul R Prucnal. Ultrafast all-optical implementation of a leaky integrate-and-fire neuron. *Optics express*, 19(3):2133–2147, 2011.

[63] Alexander N Tait, Mitchell A Nahmias, Bhavin J Shastri, and Paul R Prucnal. Broadcast and weight: an integrated network for scalable photonic spike processing. *Journal of Lightwave Technology*, 32(21):3427–3439, 2014.

[64] Jeffrey M Shainline, Sonia M Buckley, Richard P Mirin, and Sae Woo Nam. Superconducting optoelectronic circuits for neuromorphic computing. *Physical Review Applied*, 7(3):034013, 2017.

[65] Viraj Bangari, Bicky A Marquez, Heidi Miller, Alexander N Tait, Mitchell A Nahmias, Thomas Ferreira De Lima, Hsuan-Tung Peng, Paul R Prucnal, and Bhavin J Shastri. Digital electronics and analog photonics for convolutional neural networks (deap-cnns). *IEEE journal of selected topics in quantum electronics*, 26(1):1–13, 2019.

[66] Alexander N Tait, Allie X Wu, Thomas Ferreira De Lima, Ellen Zhou, Bhavin J Shastri, Mitchell A Nahmias, and Paul R Prucnal. Microring weight banks. *IEEE Journal of Selected Topics in Quantum Electronics*, 22(6):312–325, 2016.

[67] Johannes Feldmann, Nathan Youngblood, C David Wright, Harish Bhaskaran, and Wolfram HP Pernice. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature*, 569(7755):208–214, 2019.

[68] Junfeng Zhang, Bowen Ma, Yang Zhao, and Weiwen Zou. A large-scale photonic cnn based on spike coding and temporal integration. *IEEE Journal of Selected Topics in Quantum Electronics*, 29(6: Photonic Signal Processing):1–10, 2023.

[69] George Mourgias-Alexandris, George Dabos, Nikolaos Passalis, Angelina Totović, Anastasios Tefas, and Nikos Pleros. All-optical wdm recurrent neural networks with gating. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(5):1–7, 2020.

[70] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German national research center for information technology gmd technical report*, 148(34):13, 2001.

[71] Herbert Jaeger. Short term memory in echo state networks. 2001.

[72] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.

[73] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[74] Tian Gan, Susan Stepney, and Martin A Trefzer. Combining multiple inputs to a delay-line reservoir computer: Control of a forced van der pol oscillator system. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2023.

[75] Stuart Watt and Mikhail Kostylev. Reservoir computing using a spin-wave delay-line active-ring resonator based on yttrium-iron-garnet film. *Physical Review Applied*, 13(3):034057, 2020.

[76] Xulin Gao, Qiupin Wang, Yanting Liu, Shan Hu, Heman Zhang, Zhengmao Wu, Dan Lu, and Guangqiong Xia. Time-delayed reservoir computing system based on a three-stage monolithic integrated amplified feedback laser with electrical information injection. *Journal of Lightwave Technology*, 43(6):2587–2591, 2025.

[77] Dagobert Wenkack Liedji, Godpromesse Kenné, and Jimmi Hervé Talla Mbé. Delay-based reservoir computing using mackey–glass oscillator and arduino board for edge intelligence applications. *AIP Advances*, 13(12), 2023.

[78] GR Huggett. Mode-locking of cw lasers by regenerative rf feedback. *Applied Physics Letters*, 13(5):186–187, 1968.

[79] X Steve Yao and Lute Maleki. High frequency optical subcarrier generator. *Electronics Letters*, 30(18):1525–1526, 1994.

[80] X Steve Yao and Lute Maleki. Optoelectronic microwave oscillator. *JOSA B*, 13(8):1725–1735, 1996.

[81] Laurent Larger and John M Dudley. Optoelectronic chaos. *Nature*, 465(7294):41–42, 2010.

[82] Serhiy Yanchuk and Giovanni Giacomelli. Spatio-temporal phenomena in complex systems with time delays. *Journal of Physics A: Mathematical and Theoretical*, 50(10):103001, 2017.

[83] Kensuke Ikeda. Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics communications*, 30(2):257–261, 1979.

[84] R Bonifacio and LA Lugiato. Bistable absorption in a ring cavity. *Nuovo Cimento Lettere*, 21:505–510, 1976.

[85] R Bonifacio and LA Lugiato. Instabilities for a coherently driven absorber in a ring cavity. *Nuovo Cimento Lettere*, 21:510–516, 1976.

[86] X Steve Yao, Lute Maleki, Yu Ji, George Lutes, and Meirong Tu. Dual-loop opto-electronic oscillator. In *Proceedings of the 1998 IEEE International Frequency Control Symposium (Cat. No. 98CH36165)*, pages 545–549. IEEE, 1998.

[87] X Steve Yao and Lute Maleki. Multiloop optoelectronic oscillator. *IEEE Journal of Quantum Electronics*, 36(1):79–84, 2000.

[88] Y Kouomou Chembo, Laurent Larger, Hervé Tavernier, Ryad Bendoula, Enrico Rubiola, and Pere Colet. Dynamic instabilities of microwaves generated with optoelectronic oscillators. *Optics letters*, 32(17):2571–2573, 2007.

[89] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep learning in mobile and wireless networking: A survey. *IEEE Communications surveys & tutorials*, 21(3):2224–2287, 2019.

[90] Laurent Larger, Miguel C Soriano, Daniel Brunner, Lennert Appeltant, Jose M Gutiérrez, Luis Pesquera, Claudio R Mirasso, and Ingo Fischer. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Optics express*, 20(3):3241–3249, 2012.

[91] Yvan Paquot, Francois Duport, Antoneo Smerieri, Joni Dambre, Benjamin Schrauwen, Marc Haelterman, and Serge Massar. Optoelectronic reservoir computing. *Scientific reports*, 2(1):1–6, 2012.

[92] S Ortín, Miguel C Soriano, L Pesquera, Daniel Brunner, D San-Martín, Ingo Fischer, CR Mirasso, and JM Gutiérrez. A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron. *Scientific reports*, 5(1):1–11, 2015.

[93] Romain Martinenghi, Sergei Rybalko, Maxime Jacquot, Yanne K Chembo, and Laurent Larger. Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Physical review letters*, 108(24):244101, 2012.

[94] Michiel Hermans, Piotr Antonik, Marc Haelterman, and Serge Massar. Embodiment of learning in electro-optical signal processors. *Physical review letters*, 117(12):128301, 2016.

[95] Guy Van der Sande, Daniel Brunner, and Miguel C Soriano. Advances in photonic reservoir computing. *Nanophotonics*, 6(3):561–576, 2017.

[96] Laurent Larger, Antonio Baylón-Fuentes, Romain Martinenghi, Vladimir S Udaltsov, Yanne K Chembo, and Maxime Jacquot. High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. *Physical Review X*, 7(1):011015, 2017.

[97] P. Antonik, M. Haelterman, and S. Massar. Bain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems. *Phys. Rev. Applied*, 7:054014, May 2017.

[98] Daniel Brunner, Bogdan Penkovsky, Bicky A Marquez, Maxime Jacquot, Ingo Fischer, and Laurent Larger. Tutorial: Photonic neural networks in delay systems. *Journal of Applied Physics*, 124(15):152004, 2018.

[99] J. D. Hart, L. Larger, T. E. Murphy, and R. Roy. Delayed dynamical systems: networks, chimeras and reservoir computing. *Phil. Trans. R. Soc. A*, 377:20180123, 2019.

[100] F. Bohm, G. Verschaffelt, and G. Van der Sande. A poor man's coherent ising machine based on opto-electronic feedback systems for solving optimization problems. *Nat. Comm.*, 10:3538, 2019.

[101] K. Hicke, M. A. Escalona-Moran, D. Brunner, M. C. Soriano, I. Fischer, and C. R. Mirasso. Information processing using transient dynamics of semiconductor lasers subject to delayed feedback. *IEEE J. Sel.Top. Quantum Electron.*, 19:1501610, 2013.

[102] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Commun.*, 4:1364, Jan 2013.

[103] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature Commun.*, 5:3541, Mar 2014.

[104] Charis Mesaritakis, Adonis Bogris, Alexandros Kapsalis, and Dimitris Syvridis. High-speed all-optical pattern recognition of dispersive Fourier images through a photonic reservoir computing subsystem. *Opt. Lett.*, 40(14):3416–3419, Jul 2015.

[105] Quentin Vinckier, François Duport, Anteo Smerieri, Kristof Vandoorne, Peter Bienstman, Marc Haelterman, and Serge Massar. High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica*, 2:438–446, May 2015.

[106] Julián Bueno, Daniel Brunner, Miguel C. Soriano, and Ingo Fischer. Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback. *Opt. Express*, 25:2401–2412, Feb 2017.

[107] R. M. Nguimdo, E. Lacot, O. Jacquin, O. Hugon, G. Van der Sande, and H. Guillet de Chatellus. Prediction performance of reservoir computing systems based on a diode-pumped erbium-doped microchip laser subject to optical feedback. *Opt. Lett.*, 42(3):375–378, Feb 2017.

[108] F. Denis-Le Coarer, M. Sciamanna, A. Katumba, M. Freiberger, J. Dambre, P. Bienstman, and D. Rontani. All-optical reservoir computing on a photonic chip using silicon-based ring resonators. *IEEE J. Sel.Top. Quantum Electron.*, 24:7600108, 2018.

[109] YuShuang Hou, GuangQiong Xia, WenYan Yang, Dan Wang, Elumalai Jayaprasath, ZaiFu Jiang, ChunXia Hu, and ZhengMao Wu. Prediction performance of reservoir computing system based on a semiconductor laser subject to double optical feedback and optical injection. *Opt. Express*, 26(8):10211–10219, Apr 2018.

[110] A. Katumba, M. Freiberger, F. Laporte, A. Lugnan, S. Sackesyn, C. Ma, J. Dambre, and P. Bienstman. Neuromorphic computing based on silicon photonics and reservoir computing. *IEEE J. Sel.Top. Quantum Electron.*, 24:8300310, 2018.

[111] Jeremy Vatin, Damien Rontani, and Marc Sciamanna. Experimental reservoir computing using VCSEL polarization dynamics. *Opt. Express*, 27(13):18579–18584, Jun 2019.

[112] DianZuo Yue, ZhengMao Wu, YuShuang Hou, Bing Cui, YanHong Jin, Min Dai, and GuangQiong Xia. Performance optimization research of reservoir computing system based on an optical feedback semiconductor laser under electrical information injection. *Opt. Express*, 27(14):19931–19939, Jul 2019.

[113] Yaping Chen, Lilin Yi, Junxiang Ke, Zhao Yang, Yunpeng Yang, Luyao Huang, Qunbi Zhuge, and Weisheng Hu. Reservoir computing system with double optoelectronic feedback loops. *Optics Express*, 27(20):27431–27440, 2019.

[114] Xing Xing Guo, Shui Ying Xiang, Ya Hui Zhang, Lin Lin, Ai Jun Wen, and Yue Hao. Polarization multiplexing reservoir computing based on a VCSEL with polarized optical feedback. *IEEE J. Sel.Top. Quantum Electron.*, 26:1700109, 2020.

[115] A. Argyris, J. Cantero, M. Galletero, E. Pereda, C. R. Mirasso, I. Fischer, and M. C. Soriano. Comparison of photonic reservoir computing systems for fiber transmission equalization. *IEEE J. Sel.Top. Quantum Electron.*, 26:5100309, 2020.

[116] C. Sugano, K. Kanno, and A. Uchida. Reservoir computing using multiple lasers with feedback on a photonic integrated circuit. *IEEE J. Sel.Top. Quantum Electron.*, 26:1500409, 2020.

[117] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20:391–403, 2007.

[118] Yanne Kouomou Chembo, Laurent Larger, and Pere Colet. Nonlinear dynamics and spectral stability of optoelectronic microwave oscillators. *IEEE journal of quantum electronics*, 44(9):858–866, 2008.

[119] Ye Xiao, Yitang Dai, Nuannuan Shi, Wei Li, Ninghua Zhu, and Ming Li. Microwave-photonics iterative nonlinear gain model for optoelectronic oscillators. *Optics Express*, 30(7):12131–12149, 2022.

[120] Y Kouomou Chembo, Laurent Larger, Ryad Bendoula, and Pere Colet. Effects of gain and bandwidth on the multimode behavior of optoelectronic microwave oscillators. *Optics express*, 16(12):9067–9072, 2008.

[121] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. Unsupervised representation learning of structured radio communication signals. In *First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*. IEEE, 2016.

[122] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose. Recent advances in physical reservoir computing: A review. *Neural Netw.*, 115:100–123, 2019.

[123] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[124] Mantas Lukoševičius, Herbert Jaeger, and Benjamin Schrauwen. Reservoir computing trends. *KI-Künstliche Intelligenz*, 26(4):365–371, 2012.

[125] B. A. Marquez, L. Larger, M. Jacquot, Y. K. Chembo, and D. Brunner. Dynamical complexity and computation in recurrent neural networks beyond their fixed point. *Sci. Rep.*, 8:3319, 2018.

[126] Joseph D. Hart, Rajarshi Roy, David Müller-Bender, Andreas Otto, and Günter Radons. Laminar chaos in experiments: Nonlinear systems with time-varying delays and noise. *Phys. Rev. Lett.*, 123:154101, Oct 2019.

[127] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, 2015.

[128] Yongqiang Jia, Shengli Zhu, and Lu Gan. Specific emitter identification based on the natural measure. *Entropy*, 19(3):117, 2017.

[129] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. *Communications, internet, and information technology*, 1, 2004.

[130] Davide Zanetti, Boris Danev, and Srdjan apkun. Physical-layer identification of uhf rfid tags. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 353–364, 2010.

[131] Oktay Ureten and Nur Serinken. Wireless security through rf fingerprinting. *Canadian Journal of Electrical and Computer Engineering*, 32(1):27–33, 2007.

[132] Boris Danev, Davide Zanetti, and Srdjan Capkun. On physical-layer identification of wireless devices. *ACM Computing Surveys (CSUR)*, 45(1):1–29, 2012.

[133] Boris Danev and Srdjan Capkun. Transient-based identification of wireless sensor nodes. In *2009 International Conference on Information Processing in Sensor Networks*, pages 25–36. IEEE, 2009.

[134] Barnard Kroon, Susan Bergin, Irwin O Kennedy, and Georgina O'Mahony Zamora. Steady state rf fingerprinting for identity verification: one class classifier versus customized ensemble. In *Irish Conference on Artificial Intelligence and Cognitive Science*, pages 198–206. Springer, 2009.

[135] Irwin O Kennedy, Patricia Scanlon, Francis J Mullany, Milind M Buddhikot, Keith E Nolan, and Thomas W Rondeau. Radio transmitter fingerprinting: A steady state frequency domain approach. In *2008 IEEE 68th Vehicular Technology Conference*, pages 1–5. IEEE, 2008.

[136] Patricia Scanlon, Irwin O Kennedy, and Yongheng Liu. Feature extraction approaches to rf fingerprinting for device identification in femtocells. *Bell Labs Technical Journal*, 15(3):141–151, 2010.

[137] Tong Jian, Bruno Costa Rendon, Emmanuel Ojuba, Nasim Soltani, Zifeng Wang, Kunal Sankhe, Andrey Gritsenko, Jennifer Dy, Kaushik Chowdhury, and Stratis Ioannidis. Deep learning for rf fingerprinting: A massive experimental study. *IEEE Internet of Things Magazine*, 3(1):50–57, 2020.

[138] Lukas Gonon and Juan-Pablo Ortega. Reservoir computing universality with stochastic inputs. *IEEE transactions on neural networks and learning systems*, 31(1):100–112, 2019.

[139] Cyrille Morin, Leonardo S Cardoso, Jakob Hoydis, Jean-Marie Gorce, and Thibaud Vial. Transmitter classification with supervised deep learning. In *International Conference on Cognitive Radio Oriented Wireless Networks*, pages 73–86. Springer, 2019.

[140] Kangjun Bai, Lingjia Liu, and Yang Yi. Spatial-temporal hybrid neural network with computing-in-memory architecture. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(7):2850–2862, 2021.

[141] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar. Virtualization of a photonic reservoir computer. *IEEE/OSA J. Lightw. Technol.*, 34:2085–2091, Feb 2016.

[142] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar. Fully analogue photonic reservoir computer. *Sci. Rep.*, 6:22381, Mar 2016.

[143] Haoying Dai and Yanne K Chembo. Classification of iq-modulated signals based on reservoir computing with narrowband optoelectronic oscillators. *IEEE Journal of Quantum Electronics*, 57(3):1–8, 2021.

[144] Haoying Dai and Yanne K Chembo. Rf fingerprinting based on reservoir computing using narrowband optoelectronic oscillators. *Journal of Lightwave Technology*, 2022.

[145] Géraud Russel Goune Chengui, Paul Woafo, and Yanne K Chembo. The simplest laser-based optoelectronic oscillator: An experimental and theoretical study. *Journal of Lightwave Technology*, 34(3):873–878, 2016.

[146] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[147] Ali Rodan and Peter Tiňo. Simple deterministically constructed recurrent neural networks. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 267–274. Springer, 2010.

[148] A Farina, Fulvio Gini, MV Greco, and L Verrazzani. High resolution sea clutter data: statistical analysis of recorded live data. *IEE Proceedings-Radar, Sonar and Navigation*, 144(3):121–130, 1997.

[149] François Duport, Bendix Schneider, Anteo Smerieri, Marc Haelterman, and Serge Massar. All-optical reservoir computing. *Optics express*, 20(20):22783–22795, 2012.

[150] Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Serge Massar. Information processing capacity of dynamical systems. *Scientific reports*, 2(1):514, 2012.

[151] Xin Yuan, Lin Jiang, Lianshan Yan, Songsui Li, Liyue Zhang, Anlin Yi, Wei Pan, and Bin Luo. The optoelectronic reservoir computing system based on parallel multi-time-delay feedback loops for time-series prediction and optical performance monitoring. *Chaos, Solitons & Fractals*, 186:115306, 2024.

[152] Masanobu Inubushi and Kazuyuki Yoshimura. Reservoir computing beyond memory-nonlinearity trade-off. *Scientific reports*, 7(1):10199, 2017.

[153] Bernard J Giron Castro, Christophe Peucheret, and Francesco Da Ros. Memory capacity analysis of time-delay reservoir computing based on silicon microring resonator nonlinearities. In *Machine Learning in Photonics*, volume 13017, pages 115–126. SPIE, 2024.

[154] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[155] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021.

[156] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.

[157] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.