TCN-DPD

Parameter-Efficient Temporal Convolutional Networks for Wideband Digital Predistortion

Huanqiang Duan

Delft University of Technology

TU Delft

TCN-DPD

Parameter-Efficient Temporal Convolutional Networks for Wideband Digital Predistortion

by

Huanqiang Duan

Student Name:	Huanqiang Duan	
Student Number:	5399750	
Project Duration:	January 2023 - January 2025	
Thesis committee:	Prof. Dr. Leo de Vreede,	TU Delft, chairman
	Dr. Chang Gao,	TU Delft, supervisor
	Dr. Sijun Du,	TU Delft
Faculty:	EEMCS, Delft	

TU Delft

Acknowledgments

It is time to bid farewell to my master's study journey at TU Delft. In these three years of study life in Delft, I experienced a professional academic learning environment that also contained emotions with pressure, difficulties, happiness, and relief. I am fairly grateful for all the people and things I met in Delft.

First of all and foremost, I am truly grateful for my supervisor in the whole thesis, Professor Chang Gao. After the first contact in 2022 and starting the thesis, he provided very professional guidance throughout the whole project. From teaching me how to read academic papers to hands-on coding instructions, his dedication and patience have fairly exceeded my expectations. Especially when I encountered a challenge that I could not solve along the way, he always gave me detailed ideas and instructions. When the project is about to end, there are some innovative results, he reviewed them and offered quite a lot of guidance to teach me how to write a paper in academics. I can only say I am lucky to have such a perfect mentor. The only regrettable thing is that I did not have the opportunity to learn more hardware under his guidance.

I would also like to thank the members of my thesis committee, Professor Leo de Vreede and Professor Sijun Du, for attending my thesis defence and providing valuable feedback on my work. Their insights made my ideas and thesis much stronger. Additionally, I want to thank Professor Qinyu Chen, who provided significant support at the beginning of my thesis, helping me to clarify my thoughts and gain a comprehensive understanding of the project. This laid a solid foundation for the subsequent experiments.

I am also deeply grateful to my friends Jing Wen, Yuhao Chen, and Qing Zhang. Whether it was relaxing me during difficult times or offering suggestions, your support and encouragement have been truly invaluable.

Finally, for my family, your love and support are hard to express with words. I only can say I love you all.

Huanqiang Duan
Delft, January 2025

Summary

As a crucial component, one common challenge of power amplifiers (PA) is the nonlinearity in the wireless communication system. Digital predistortion (DPD) is essential for mitigating nonlinearity in radio frequency (RF) power amplifiers, particularly for wideband applications.

This thesis work aims to present the TCN-DPD model, a novel parameter-efficient architecture based on temporal convolutional networks (TCNs) to enhance the performance of PA. Therefore, the main problem should focused on the TCN-DPD model implementation with parameter efficiency.

When TCN architecture was designed with several noncausal and dilated depthwise convolution layers and 1*1 convolution layers, optimized activation functions should be explored to complete the TCN architecture. By evaluating on the `OpenDPD` framework with the `DPA_200MHz` dataset, Hardswish, Tanh, SiLU, and GELU were considered by benchmarking different activations of TCN-DPD based on SIM-NMSE and SIM-ACLR metrics on average. Hardswish was confirmed in the later experiments as the optimized activation function in TCN architecture based on simulated results ACLR of -51.54 dBc and NMSE of -44.61 dB.

Since the TCN-DPD architecture was completed, this proposed model's performance in PA and DPD benchmarks is desirable to be tested, and later experiments will use the same dataset and framework as the benchmark of activation function did. In PA benchmarking, the TCN model achieves SIM-NMSE -34.99 dB on average compared to other models, LSTM, GRU, RVTDCNN, VDLSTM, PNTDNN, and DGRU. This achievement shows the TCN architecture has a high potential to handle a range of dependencies efficiently in the DPD application system.

Furthermore, DPD benchmarking is the main experiment in this thesis. Two architectures were selected as the pre-trained PA model: the DGRU and TCN models. When the pre-trained PA is fixed as the DGRU model, TCN-DPD demonstrates superior linearization performance with only 500 real-valued parameters, achieving averaged and simulated ACPRs of -51.58/-49.26 dBc (L/R), EVM of -47.52 dB, and NMSE of -44.61dB. The results are simulated ACPRs of -50.39/-50.01 dBc (L/R), EVM of -47.88 dB, and NMSE of -45.51 dB in average when the pre-trained PA model is TCN. Both DPD benchmarks include different DNN-DPD models, and TCN-DPD has superior performance in the comparison, especially the SIM-NMSE and SIM-EVM performance is significantly higher than other models when the pre-trained model is TCN. These results establish TCN-DPD as a promising solution for efficient wideband PA linearization.

Moreover, the evaluation extended to DNN-DPD performance with various numbers of parameters ranging from 200 to 1000, where the TCN-200 model highlighted its effectiveness by showing impressive results in SIM-NMSE -41.27dB/-43.51dB(DGRU/TCN PA), achieving superior linearization performance while using significantly fewer parameters than existing deep neural network solutions, proving the TCN-DPD model's parameters efficiency.

The research in this thesis conclusively demonstrates that TCNs can be implemented in DPD applications, providing more parameters efficiency, better performance, and robust PA linearization solutions, potentially setting a new alternative in DPD technology.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
PA	Power Amplifier
DPD	Digital Predistortion
MP	Memory Polynomial
GMP	Generalized Memory Polynomial
TDNN	Time Delay Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
DGRU	Dense Gated Recurrent Unit
RVTDCNN	Real-Valued Time-Delay Convolutional Neural Network
VDLSTM	Vector Decomposed Long Short-Term Memory
TCN	Temporal Convolutional Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
PN-TDNN	Phase-Normalized Real-valued Time Delay Neural Network
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature Amplitude Modulation
AM-AM	Amplitude-to-Amplitude
AM-PM	Amplitude-to-Phase
NMSE	Normalized Mean square Error
ACPR	Adjacent Channel Power Ratio
EVM	Error vector Magnitude
Act. Funcs.	Activation Functions
FEx	Feature Extraction Layer
FC	Fully Connected
TWTAs	Travelling Wave Tube Amplifiers
RLS	Recursive Least Squares
ILC	Iterative Learning Control
DLA	Direct Learning Architecture
ILA	Indirect Learning Architecture

1 Introduction

1.1. Motivation

The increasing demand for high-efficiency, wideband communication systems has intensified the need for effective power amplifier (PA) linearization techniques. As modern wireless systems push toward broader bandwidths and higher efficiency requirements, digital predistortion (DPD) has become indispensable for maintaining signal quality while allowing PAs to operate in their efficient nonlinear regions [1, 2].

The historical evolution of DPD techniques shows the efforts to address PA linearization's evolving challenges in the past few decades. In the 1980s, memoryless polynomial models were proposed to provide the initial methods for modeling static PA nonlinearities; this model stated the primary chapter of DPD technology in the early stage [3]. In the 1990s, the Volterra series model was proposed, which can capture the memory effects, but the computation load is heavy [4, 5]. As entered the 21st century, Memory Polynomial(MP) models were introduced to offer a low complexity with memory effects to linearize the PAs, but limited with the bandwidth [6, 7].

In the mid-2000s, the Generalized Memory Polynomial (GMP) model was proposed, a significant milestone in DPD's evolution. The GMP model has long been the industry standard for PA linearization [8]. However, as communication bandwidths expand and modulation schemes grow more complex, GMP's limitations in modeling sophisticated memory effects have become increasingly apparent [9, 10]. This challenge has driven the exploration of more advanced modeling approaches capable of capturing intricate PA behavior across wider bandwidths.

Deep learning has emerged as a promising solution for next-generation DPD systems. The evolution began with Time Delay Neural Networks (TDNNs), which demonstrated remarkable effectiveness in modeling the sequential memory characteristics inherent in PAs [11]. This success led to enhanced variants such as the Phase-Normalized Real-valued Time Delay (PN-TDNN) architecture [12]. The field subsequently witnessed the implementation of various recurrent neural architectures, including Long Short-Term Memory (LSTM) [13], Vector Decomposed Long Short-Term Memory (VDLSTM) [14], and Gated Recurrent Unit (GRU) [15] networks. These architectures have shown exceptional capabilities in capturing temporal dependencies within RF signals. Recent innovations have also incorporated convolutional neural networks (CNNs), exemplified by the RVTDCNN model, which leverages convolutional layers to model both spatial and temporal signal characteristics simultaneously. The RVTDCNN models use the 2D convolutional layers to operate the formatted features from in-phase(I) and quadrature(Q) of the signals [16]. In contrast, TCNs employ 1D convolutions directly on the input time series, combined with dilated convolutional layers that efficiently capture long-term memory effects. This architectural difference probably enables TCNs to achieve more parameter-efficient wideband DPD while maintaining or improving modeling accuracy. However, despite these advanced deep neural network(DNN) based digital predistortion (DPD) models, Temporal Convolutional Networks (TCNs) remain largely unexplored in DPD applications, even though they have demonstrated significant advantages in capturing long-range temporal dependencies with reduced computational complexity in time series prediction tasks [17, 18].

Given the demonstrated advantages in various time series prediction tasks, TCNs represent a promising frontier in the evolution of DPD technology. Their potential for improving the linearization of PAs in complex, high-efficiency communication systems positions TCNs as a critical area of research. By leveraging the historical insights and technological advancements outlined, exploring TCNs in DPD could lead to substantial improvements in the performance and efficiency of future wireless communication systems.

1.2. Problem Statement and Objectives

Digital predistortion (DPD) is essential for mitigating nonlinearity in radio frequency (RF) power amplifiers, particularly for wideband applications. Temporal Convolutional Networks (TCNs) are widely used for sequential tasks with high efficiency and effectiveness, TCNs' characteristics are fairly affable for the needs of DPD techniques by including the sequential data of inn-phase(I) and quadrature(Q) components. The integration of TCNs into the DPD techniques shows a promising opportunity due to the TCNs' advantages, such as the parameter efficiency, computational complexity, and the capabilities to handle long-range dependencies in the signal data.

Despite the theoretical alignment of TCNs' capabilities with DPD requirements being agreeable, practical implementation of a TCN-based DPD system remains largely unexplored. This gap presents a critical problem as Fig 1.2.1 show:

- TCNs own numerous advantages, such as their parameter efficiency, stability in handling long sequences, and low computational complexity, which can be effectively harnessed to DPD techniques for improving PA linearization.

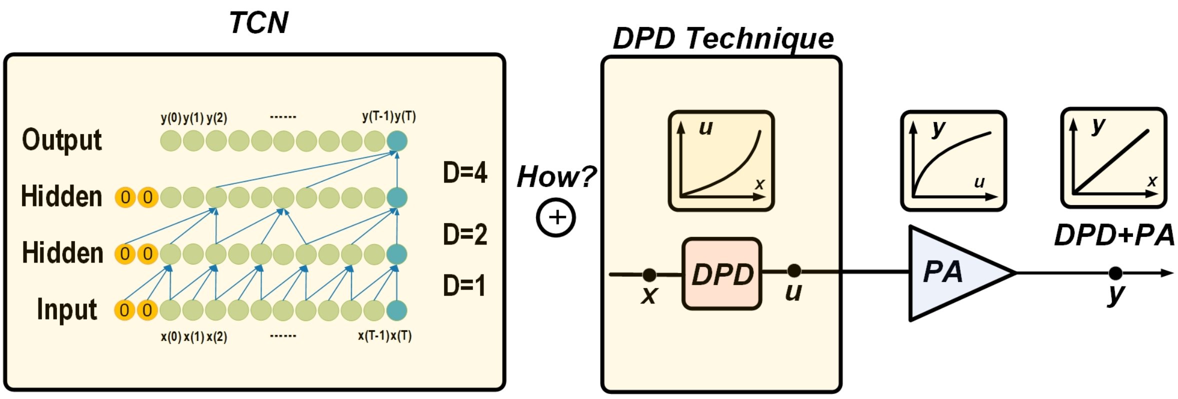

Figure 1.2.1: Problem Statement of TCN-DPD

This research aims to address this problem by designing and testing a TCN-based DPD architecture. The specific objectives are to:

- **Design and Implementation:**

  Design a specific TCN architecture tailored to the requirements of the DPD system by successfully handling the sequential I/Q data in the signal.

- **Computational Efficiency:**

  Decrease the computational efficiency of TCNs in the DPD applications with similar performance when compared with DNN-based DPD models in communication systems.

- **Parameter Optimization:**

  Explore the parameters efficiency of TCN DPD for linearizing the PAs to achieve comparable or superior performance with lower parameters.

- **Performance Comparison:**

  With the same parameters and computational complexity, this work aims to validate the improvement of accuracy and efficiency of the TCN architecture in DPD by achieving superior performance when compared with the traditional DNN-based DPD models.

The successful integration of TCNs into DPD systems could be an evolution of PA linearization techniques, offering a more robust, efficient, and scalable solution for modern and future communication systems. This research will not only contribute to the theoretical understanding of TCN applications in DPD but also aims to provide a practical solution that could be adopted in wireless systems.

1.3. Research Questions and Scope

Research Questions for Integrating Temporal Convolutional Networks (TCNs) into Digital Predistortion (DPD) Systems

- How can the advantages of TCNs—such as their parameter efficiency, stability in processing long sequences, and computational advantages—be effectively harnessed to DPD techniques for improving PA linearization?
- How can a Temporal Convolutional Network be specifically designed to meet the demands of Digital Predistortion for handling the sequential in-phase (I) and quadrature (Q) components of signals in communication systems?
- What are the computational efficiencies of TCNs when applied to Digital Predistortion systems?
- How does the parameter efficiency of TCNs affect their performance in modeling the nonlinearities of power amplifiers compared to traditional DNN-based DPD models?
- How does a TCN-based DPD model perform in terms of linearization accuracy and system efficiency when compared to traditional DNN-based DPD models?

Scope

This thesis is dedicated to exploring the potential of Temporal Convolutional Networks (TCNs) in Digital Predistortion (DPD) systems, emphasizing simulation-based research within the OpenDPD framework. The scope of this study is defined by several key limitations and focus areas that outline the boundaries within which the research is indicated:

- Simulation-Based Approach: The research presented in this thesis relies exclusively on simulations performed within the OpenDPD framework. No real-world measurements with physical hardware setups are included. This approach allows for a controlled analysis of the TCN's theoretical capabilities and performance in a standardized environment, facilitating comparisons with existing DNN-based DPD models.
- Exclusion of Power Consumption Analysis: This study does not consider the specific power consumption of the TCN-based DPD model. The focus remains primarily on the linearization performance and computational aspects of the model rather than its energy efficiency. Future studies may need to explore the power consumption metrics to provide a more comprehensive evaluation of the model's practical capability.
- Single Dataset for Stability: To ensure the stability and reproducibility of the simulation results, all experiments tested as part of this thesis utilize only one dataset. This controlled variable approach helps maintain consistency across tests but limits the generalizability of the findings. The dataset represents typical scenarios encountered in DPD applications, but variations in different dataset conditions could lead to different outcomes.
- Exploratory Nature and Reference Use: The findings and conclusions drawn from this research are intended for reference and as a preliminary exploration into the application of TCNs in DPD systems. While the results provide valuable insights into the potential benefits and limitations of using TCNs for PA linearization, they are indicative rather than definitive. The actual performance of the TCN-based DPD model should ideally be validated through real-world measurements and a broader set of experiments to confirm their efficacy outside of the simulated environment.

1.4. Thesis novitiates

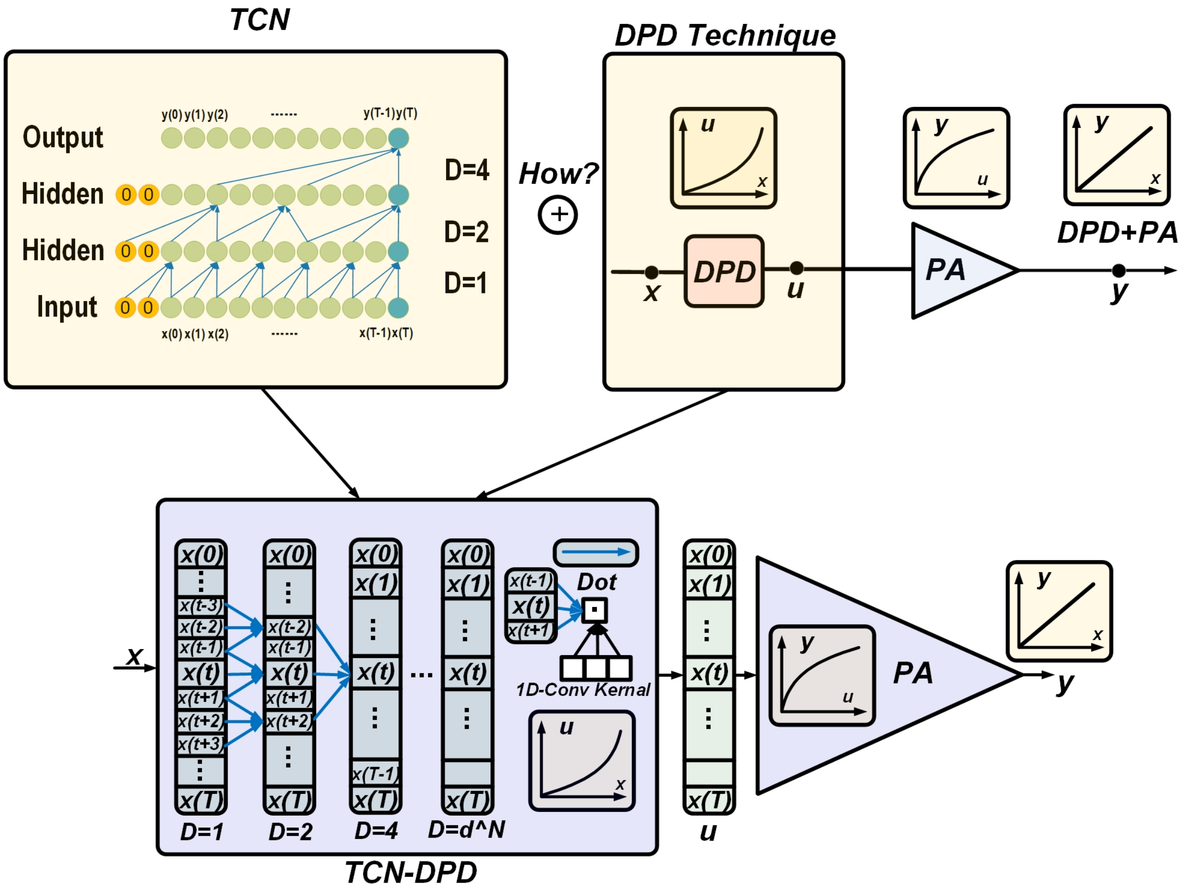

Figure 1.4.1: Basic concept of TCN-DPD with pre-trained PA model for linearization; example with noncausal 1D convolution layers with dilation sizes $D = 1, 2, 4$ up to d^N in the last layer, and kernel size of 3

1. **A Novel TCN-based DPD Model proposed:** As Fig 1.4.1 shows, this thesis's work proposes a novel approach by implementing the temporal convolutional network(TCN) into the digital pre-distortion(DPD) of power amplifiers(PAs). This model architecture design can effectively process the sequential data in-phase(I) and quadrature(Q) components of radio frequency(RF) signals by using noncausal and dilated convolutional layers to capture the long-range temporal dependencies.
2. **Computational complexity reduced:** Another novelty of this thesis's work is the reduction in the computational complexity. The TCN DPD model achieved a similar performance with lower parameters, showing the computational load reduced without sacrificing the performance. The reduced computational complexity enhances the ability to adapt the TCN DPD model in real-world communication systems.
3. **Parameters Efficiency Achieved:** One of the significant novelties of the thesis's work is the achievement of remarkable parameter efficiency in the DPD field. The TCN-based DPD model can achieve excellent performance with fewer parameters, which fairly improves the sustainability of DPD systems.
4. **Superior Performance Compared to Other DNN-based DPD Models:** The most impactful novelty is the benchmark results provided by the thesis, which show that the TCN-based DPD model achieves superior performance and outperforms all the existing deep neural network(DNN) based DPD models. This finding shows the potential of TCN architecture in the DPD field and offers an efficient approach and alternative for linearizing the PAs in the real-world communication system.

1.5. Outline

Chapter 2: Literature Review

The second chapter gives a comprehensive literature review, including the traditional and deep neural network-based DPD technique approaches and the general temporal neural networks(TCNs) architecture design. In the first subsection, the evolution of DPD techniques is introduced and summarized to have a basic concept of DPD. In the subsequent subsection, the DNN-based DPD are explained with the classic architecture like RNNs, TDNNs and CNNs, these DNN-based DPD models are also the compared model in the thesis. In the final subsection, the general TCN architecture will be analyzed, especially the utilization of causal and dilated convolutions and the advantages and disadvantages of residual block.

Chapter 3: TCN Architecture Design

The third chapter introduces the TCN DPD architecture design. It includes various innovations like the noncausal convolution for capturing the future elements in the signals, dilated convolutions for expanding the receptive fields, and the core depth separable convolutional block instead of the residual block for achieving high performance with a low number of parameters. This architecture is tailored for the DPD techniques and tests on the OpenDPD framework.

Chapter 4: Experimental Results

In the experimental results chapter, the experimental setup and the results will be provided based on the TCN-based DPD model test on the DPA_200MHz dataset. These experiments include the benchmarks of different activation functions, the power amplifier's behavioral evaluation, and the digital predistortion performance based on the different pre-trained PA models in the final. The results show that TCN's performance comparison with other DNN-based DPD models proves this work supports the thesis's objectives and innovations.

Chapter 5: Conclusion and Recommendation

The final chapter shows the main findings of this thesis's work, answers the research questions, and proves the objectives that have been supported. The main contributions also reflect the innovation of DPD systems in this thesis. The limitations have also been indicated in the recommendation, which will be helpful for further research. The recommendation introduces the subsequent actions that should be taken to improve the TCN DPD model's capability in the future.

2

Literature Review

2.1. Digital Predistortion (DPD)

Power amplifiers(PAs) are crucial components in modern communication systems. By amplifying the signal power, PAs cost great quality power in the communication infrastructure. Power consumer's roles make the PAs significant in the transmission system. However, one of the biggest challenges is the nonlinearities in PAs, which can fairly degrade the signal quality and introduce interference. This problem makes the reliability of data transmission unstable in the real world; addressing these nonlinear distortions is imperative [1].

Higher data rates and wide bandwidth are necessary and should be adopted in advance by the evolution of wireless communication, such as the modulation techniques, which include Orthogonal Frequency Division Multiplexing (OFDM) and Quadrature Amplitude Modulation (QAM). These ever-changing methods' complex signal structures are quite sensitive to nonlinear distortions. This characteristic also will make PAs' challenges greater, like communication errors and interference with nearby channels, and developing linearization techniques is essential for eliminating the distortions in the data transmission.

Early research on linearization techniques can be introduced in the 20th century. In 1920, feedforward and feedback techniques were proposed to meet the demands of the telecommunications industry and radio broadcasting. Feedback and feedforward techniques were effective but limited by the complexity and power consumption. These two early approaches have the foundation for subsequent studies in PA linearization [2].

Predistortion is an alternative to feedback and feedforward techniques to linearize the PA. In the early implementation, analog was always considered with the development of digital signal processing, digital restoration techniques surfaced in this field. DPD technique offers greater flexibility and adaptability compared to analog predecessors, making it the preferred choice in the modern communication system. Fig 2.1.1 shows the basic concept of the DPD technique to linearize the power amplifiers.

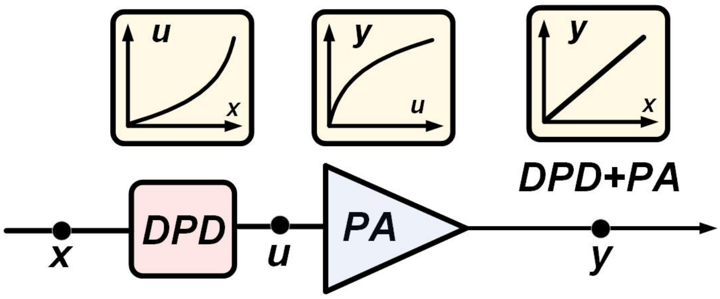

Figure 2.1.1: Basic Concept of DPD For PA Linearization

As a crucial technology in modern wireless communication systems, Digital Predistortion (DPD) stands out in improving the performance and efficiency of Radio Frequency Power Amplifiers(RFPAs) while reducing costs and computational complexity. Especially when the modern communication system needs comparable efficiency with wide bandwidth, the role of DPD becomes more important, which is why the DPD technique enables maximum power efficiency and maintains signal integrity in PAs.

There are several metrics, Normalized Square Error(NMSE), Error Vector Magnitude (EVM), and Adjacent Channel Power Ratio(ACPR), are used to evaluate the linearity and signal quality of DPD systems as evidence. By enabling PAs to function more efficiently, DPD development is fairly historical due to the aims of reducing the power consumption in wireless transmission and promoting the economic and environmental sustainability of communication networks.

This section introduces a historical evolution of DPD techniques as a foundation since the thesis's main problem is implementing the TCN architecture into DPD systems. By researching DPD's development from the early stage to the used and complicated DPD systems today, we can understand the main innovative ideas in this field. Table 2.1.1 shows the milestones in the DPD development history.

Decade	**Technique**	**Key Paper**	**Highlights**
1980s	Memoryless Polynomial	[3]	Early modeling for static PA linearization.
1990s	Volterra Series	[4, 5]	Captured memory effects but computationally heavy.
2000s	Memory Polynomial	[6, 7]	Low complexity but limited bandwidth.
2000s	Generalized Memory Polynomial	[8]	Trade-off between complexity and accuracy, but overfitting with wider bandwidth in modern communication system.
2010s	TDNN	[19, 11, 12]	Early DNN based DPD model with good performance
2010s	RNN	[14, 15]	Learn complex memory effects.
2010s	CNN	[16]	First CNN DPD approach with comparable efficiency
2020s	OpenDPD Framework	[20]	Benchmarking framework for DPD research.

Table 2.1.1: Evolution of Digital Predistortion Techniques for PA Linearization

From the earliest development of DPD techniques, which are the memoryless polynomial models, this memoryless model has been the foundational research in DPD development until now. The main idea is only to consider the static nonlinearities in the memoryless polynomial model, like the amplitude to amplitude(AM/AM) and amplitude to phase(AM/PM) distortions in PAs. Without considering the memory effects, which are the past values in signal processing, the output value at time t will only be influenced by the current value. This assumption will make the equation simple as the output is equal to a function of the input amplitude and will make this model computation and parameters efficient in the DPD application. However, it is easy to know the accuracy will not be excellent in the real world; acceptable linearization should be achieved only when the frequency range is narrow and memory effects are negligible. The earliest application of the memoryless model is used in the satellite systems to linearize the Travelling Wave Tube Amplifiers (TWTAs) with sufficient accuracy in data transmission [3]. This model shows the early approach to linear Power Amplifiers (PAs) as a foundation for more complicated studies in the future. The limitation of bandwidth and memory effects to motivate the development of memory effects is considered a model.

Volterra series model, as a significant advancement in Digital Predistortion(DPD) system, since the biggest difference is considered the memory effects after memoryless polynomial model [4]. The specific characteristic is to capture both static and dynamic effects in linearization processing, which is crucial to increasing the signal quality. Since the traditional methods can not address the management of high-order nonlinearities, [5] proposed an approach by using the Indirect Learning architecture (ILA). This architecture applied a Recursive Least Squares (RLS) algorithm in the signal processing, which can adapt the pre-distorter coefficients iteratively to increase the efficiency and applicability of linearization. The Volterra series model was applied to satellite communications practically, with the memory effects considered; the complex signal modulations also could be handled with superior linearization performance in the 1990s. With these advantages, the Volterra series model's limitation is the computational demands, especially when the complex signal modulation with extended memory. In general, the Volterra model with Indirect Learning Architecture advances the evolution of DPD technology.

After a few years, the Memory Polynomial (MP) model was proposed for linearizing Power Amplifiers(PAs). This model provided another solution for PAs linearization with memory effects, especially for wideband systems like OFDM signals. The difference with the general Volterra model is that the MP model only considers the captured memory terms and nonlinearity interactions. This will reduce the computational complexity and make this model computationally feasible. Therefore, the MP model offers a way that can trade off the computational complexity and accuracy in the DPD system, as a specific case of the Volterra model with Indirect Learning Architecture(ILA) [6, 7]. The MP model addressed the problem of memoryless approaches and the Volterra series models, like the memory effects and intensive computation. However, the limitation of the MP model is the wideband applications with higher-order nonlinearities and long memory effects. Overall, the ability to balance the complexity and accuracy makes the MP model an alternative to the DPD system for the real world.

The generalized memory polynomial (GMP) was proposed in 2006 as a significant milestone over the past conventional DPD model. The main contribution is that it provides a solution that can linearize the PAs accurately with memory effects in wideband communication systems, addressing the common challenge of the memoryless model, Volterra series model, and memory polynomial (MP) model. The working principle of the GMP model is similar to the MP model, but considering the cross term between the delayed samples and using the least square techniques will fairly improve the performance of PAs linearization with wideband signals. Like the MP model, the GMP model also uses the Indirect Learning Architecture. This approach can also give the GMP model a trade-off between computation and performance to become a practical deployment in the real world. Since the GMP model takes a significant step forward in developing DPD research, the GMP model has long been the industry standard for PA linearization [8]. However, as communication bandwidths expand and modulation schemes grow more complex, GMP's limitations in modeling sophisticated memory effects have become increasingly apparent [9, 10]. This challenge has driven the exploration of more advanced modeling approaches capable of capturing intricate PA behavior across wider bandwidths.

To address the challenges of RF Power Amplifiers with complex nonlinearities and memory effects in wideband signals, deep neural network-based digital predistortion (DNN-based DPD) techniques have been developed and applied with different architectures. Time delay neural networks (TDNNs) were proposed for phoneme recognition [19] by Waibel for the first time and were widely explored to the DPD system by extracting the time delayed in-phase and quadrature components from the input signals, which is the I and Q to the network's architecture [11]. In the early stage, TDNNs DPD have been treated as an efficient approach for linearizing the PAs, recent research like phase normalized TDNNs, which was proposed to achieve high accuracy with reduced complexity in wideband system by preprocessing the input with phase normalization [12]. Based on TDNNs' advancement in the DPD systems, Recurrent Neural Networks(RNNS) provide a huge potential to capture the long-memory effects of their mechanisms. Specially, Long short-term (LSTM) and Gated Recurrent Units (GRU) are special RNNs that avoid the gradients vanishing and exploding problem, which have been applied to the DPD systems [14, 15, 21]. These models are fairly suited for modern RF power amplifiers (PAs) since the DPD techniques can be simplified into a special sequential task. At the same time, convolutional neural networks (CNNs) are a very powerful architecture for image processing. Therefore, a desirable expectation from the scientist is that their advantages could be connected to the DPD systems. CNN-based DPD approach, real-valued time delay convolution neural network (RVTDCNN), was proposed to linearize the PAs by extracting the input features into the structured layers. This model can reduce the computation load significantly compared to other traditional neural networks in the communication system with high bandwidth signals [16]. Overall, these deep neural networks(DNNs) provide a robust evolution in DPD techniques, integrating adaptability and computational efficiency to meet the demands of modern communication systems. Otherwise, these DNN-based DPD models will be introduced in detail in the next section as the compared model in the thesis.

Besides the DPD's historical evolution, several architectures have always been used for the various DPD models, such as Iterative Learning Control (ILC), Direct Learning Architecture (DLA), and Indirect Learning Architecture(ILA). As Table. 2.1.2 shows, different architectures have strengths and limitations.

Feature	**DLA**	**ILA**	**ILC**
Adaptation Method	Direct feedback from output error	Inverse modeling	Iterative error minimization
Convergence Speed	Depends on complexity of error	Faster, based on inverse model	Slower, needs multiple iterations
Computation Load	High (depending on complexity)	Lower	Higher, especially for high-order models
Robustness to Errors	Can be sensitive to initial conditions	Robust (if inverse model is accurate)	More robust, adjusts iteratively
Handling Memory Effects	Effective with complex models	Can handle memory effects indirectly	Naturally compensates memory effects

Table 2.1.2: Comparison of DLA, ILA, and ILC Architecture in DPD

ILA is one of the most widely used architectures for DPD techniques since its straightforward implementation analyzes the output of its input as an inverse modeling. This simplicity makes the computational load low with acceptable militarization but is limited by the difference between the training conditions and real-world situations [6, 8]. DLA was designed to address the limitations of ILA by directly receiving feedback from the output error. This architecture can be more effective with complex models but limited with high computation load and sensitivity of initial conditions [22]. ILC was proposed and became the most flexible architecture compared with ILA and DLA. ILC utilizes iterative optimization to adapt the PA input based on the real output and the desired linear output. This method makes the model achieve high adaptability with strong memory effects. However, the limitation is the convergence speed will be slow when the communication is at a high speed [23]. The architectures could be selected based on the different needs of DPD models. ILA will be attractive when the requirements are low computation and high-speed processing; DLA could be used when the DPD models need to handle complex signals, and ILC could be applied when the computation load could be ignored but high performance is necessary.

In summary of this section, the DPD technique is a cornerstone of RF power amplifier linearization and vital for the wireless communication system. The continuous evolution of DPD techniques will be crucial to achieving greater performance and capabilities in the future. There are still plenty of challenges, such as the complexity of DPD modeling, parameter efficiency, multiband signals, etc. This thesis will address some of the challenges of DPD techniques by proposing a TCN DPD model with efficient parameters.

2.2. Deep Neural Network Based Digital Predistortion (DPD)

2.2.1. Time Delay Neural Network Based DPD

Time-delay neural Networks (TDNNs) were proposed to execute the sequential data with temporal properties and can be treated as a type of feedforward neural network. The main characteristic of TDNN is that the neurons can handle the data from different points in time instead of relying only on the current moment. This working principle can make TDNNs capture more temporal connections in data. TDNNs have been widely used in different applications like speech recognition, gesture recognition, and also DPD techniques [11].

In detail, and different from other neural network architecture, TDNNS can make a moving window of the input features like the I and Q in DPD systems by adding a series of delays to the input for recognizing the patterns across the different time steps. The main core of TDNN architecture is similar to a standard feedforward neural network with the addition of time delays between the connected layers. These delays will create memory effects by using the information from various time points.

In DPD systems, the TDNN model is often used to correct the nonlinearities of RF power amplifiers, which are easily affected by past signal states. TDNNs DPD models generally learn to predict the distortion by a predistortion function, which can produce the output signal by taking the transmission signal as input and transforming it into the predistorted signal. Recently, Phase Normalized Real-Valued Time-Delay Convolutional Neural Network (PN-TDNN) architecture was proposed and achieved great results in DPD application [12], also will be treated as the compared model in this thesis. As Fig 2.2.1 shows, the core architecture of the PN-TDNN DPD model and components are listed here:

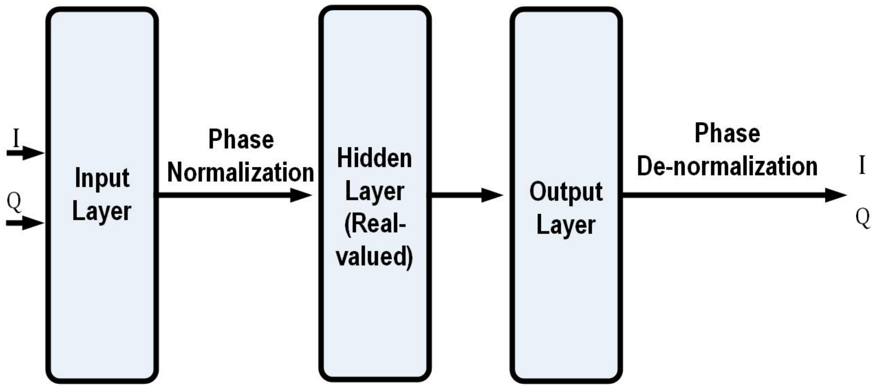

Figure 2.2.1: Block Diagram of Phase Normalized Real-Valued Time-Delay Convolutional Neural Network DPD Architecture

- **Input Layer:** The network receives complex I/Q inputs, where 'I' is the in-phase and 'Q' is the quadrature component. Each I/Q sample is normalized by its phase, using the phase normalization factor to focus the network's learning on amplitude and relative phase changes rather than absolute phase values. This normalization involves dividing the complex I/Q vector by its magnitude and adjusting its phase angle towards zero, making the input purely real for the phase of the current sample. The input vector is augmented with past values(memory effect) and possibly other derived features, such as powers of the baseband amplitude, to capture the dynamics and memory effects of the amplifier more effectively.
- **Hidden layer:** These layers are composed of real-valued neurons and capture the nonlinear relationships between the normalized inputs and the desired output. These layers can use various activation functions to introduce non-linearity, such as ReLU.
- **Output Layer:** At the output, the learned real-valued signals are converted back into complex I/Q format by re-applying the phase information normalized at the input stage. This ensures that the output retains the correct phase characteristics necessary for accurate RF performance, The final output is reconstructed by re-integrating the phase information through a complex multiplication, which aligns the output back with the original I/Q phase, ensuring that the predistorted signal correctly compensates for the amplifier's nonlinearities.

This architecture is powerful for real-time DPD applications, but it is sensitive to the overfitting problem. Therefore, the setup of hyperparameters, such as the depth of the network, number of layers, and learning rate, is fairly important. As a compared model with TCN, parameter tuning is necessary before the experiments.

2.2.2. Recurrent Neural Network Based DPD

Recurrent Neural Networks(RNNs), as a classic neural network, are proposed specifically for sequential tasks. RNNs are highly different from the standard feedforward neural networks, like fully connected networks. The core working principle is that RNNS maintains the memory effects by utilizing the output as part of the input. This method will make the RNNs effective for the historical data, which are significant for applications like language processing, time series analysis, and signal processing. However, the advantages of RNNs are impressive but limited by vanishing and exploding gradient problems due to the long-term dependencies. Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have been developed to solve this limitation and are introduced below:

LSTM

To address the vanishing or exploding problem in traditional Recurrent Neural Networks(RNNs), Long Short-Term Memory (LSTM) networks are designed. The working principle is mainly dependent on the designed gates in the architecture. With input time series x_t, LSTMs include a bunch of gates, which are input gate i, output gate o, cell gate $\tilde{C}$ and forget gate f to process the information in the memory cell state C. At time step t, output gate o will decide what cell memory should be transferred to the hidden state h. These gates decide which information to maintain or eliminate, enabling the network to capture long-term dependencies [13]. The formulations are:

$$
\begin{gathered}
f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\
o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t = o_t * \tanh(C_t)
\end{gathered}
$$

which W and b are the weight matrices and bias vectors, and σ is the logistic sigmoid function.

GRU

Since the LSTMs still suffer from complicated structures, Gated Recurrent Units (GRUs) are designed to simplify the LSTM structure by combining the input and forget gates into a single gate, which is named the update gate. The reset gate r and new hidden state $\tilde{h}_t$ are also proposed, resulting in a more efficient learning process [15]. The formulations are:

$$
\begin{gathered}
z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\
r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\
\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b) \\
h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t
\end{gathered}
$$

where z_t is the update gate, r_t is the reset gate, $\tilde{h}_t$ is the candidate for the new hidden state, and other symbols are the same as the LSTM.

RNNs apply to Digital Predistortion (DPD) by modeling the dynamic behivior of RF power amplifiers, exhibiting non-linear responses and memory effects. In DPD, an RNN can learn the complex relationships between the input signals and the distorted outputs produced by the amplifiers. By training on historical data of input-output pairs, the RNN effectively predicts and compensates for these distortions, adjusting the input signal to minimize errors in the output [14, 21].

[20] offers a concrete example of how Recurrent Neural Networks (RNNs), as Fig 2.2.2 shows, specifically through a Dense Gated Recurrent Unit (DGRU) architecture, can be effectively applied in the field

of Digital Predistortion (DPD) techniques for power amplifiers. This novel DGRU-DPD architecture incorporates a Gated Recurrent Unit (GRU) with a dense skip path that includes a feature extraction layer (FEx), input layer, GRU layer, hidden fully connected (FC) layer, and an output FC layer.

The feature extraction layer is instrumental as it processes the input signal components, the in-phase (I) and quadrature (Q) components, which represent the real and imaginary parts of the signal in the time domain. It computes the absolute value, cubic value, sine, and cosine of the angle between the I and Q components, which are then forwarded to subsequent layers. This preprocessing is critical for enhancing the model's ability to capture the nonlinear characteristics and dynamic behivior of the RF amplifier being modelled, and this model will be compared with the thesis's work TCN-DPD model.

By integrating these extracted features back into the input of the output FC layer through concatenation, this architecture aims to combat the vanishing gradient problem commonly happening in training deep neural networks. Moreover, this strategy leverages the inherent temporal modeling capabilities of the GRU, making it exceptionally suited for the dynamic nature of DPD, where the amplifier's characteristics may change over time due to thermal effects and ageing. This approach shows how RNN enhances the linearization efficiency of the model in the field of RF communications.

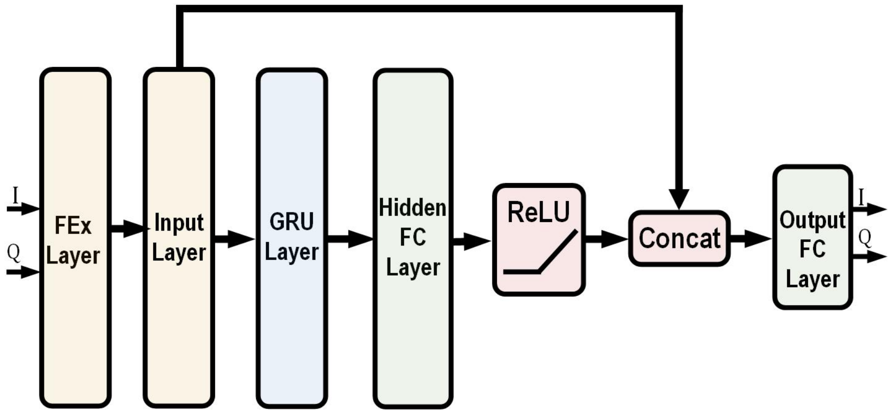

Figure 2.2.2: Dense Gated Recurrent Unit DPD Architecture

2.2.3. Convolutional Neural Network Based DPD

Convolutional Neural Networks(CNNs) are also very classic and powerful neural networks designed especially for image processing. The general CNN architecture includes a series of layers, which are convolutional, pooling layers, and fully connected layers. This architecture is fairly effective for capturing the spatial features and can be applied in the real world popularly, like the complex image recognition tasks and autonomous driving [24, 25].

Convolutional Neural Networks (CNNs) provide numerous attractive advantages in feature extraction and make CNNs a potential candidate for Digital Predistortion implementation. Until now, there are only a few research to explore the CNNs DPD application; a novel CNNs DPD model was proposed by [16] and also the compared model in our thesis.

Real-valued time-delay Convolutional Neural Networks (RVTDCNN) are proposed to combine the CNNs' advantages to the DPD systems and aim to achieve efficient linearization and memory characteristics in wideband PAs. The objective of the RVTDCNN model design is also to address the high signal bandwidth limitations that most conventional DPD models face as significant obstacles.

The RVTDCNN model successfully implemented a convolutional neural network architecture to the DPD systems by handing the one-dimensional time-varying elements of signal data like the in-phase and quadrature (I/Q) components differently. These one-dimensional inputs will be formatted into a two-dimensional structure and still include the original information. The formatted input features will be captured by using the convolutional layers to make complex interactions through a predefined filter and be processed by the subsequent layers in the network architecture. The RVTDCNN structure is illustrated as Fig 2.2.3:

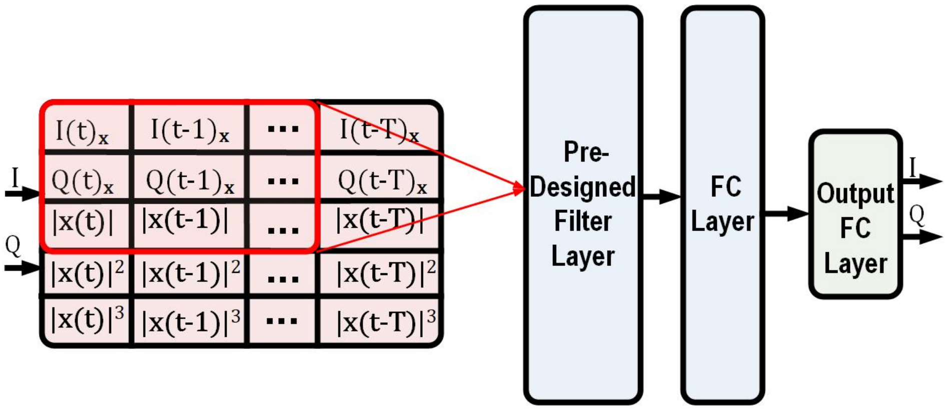

Figure 2.2.3: Real-Valued Time-Delay Convolutional Neural Network DPD Architecture

- **Input Layer:** The input data I and Q will be formatted into a two-dimensional pattern and processed by the convolutional layers.
- **Convolutional Layer (Predesigned Filter)**: The formatted features from the input will be convoluted by multiple convolutional layers with kernels; the function of these convolution layers is to reduce the model complexity by sharing the weight in CNN.
- **Fully Connected Layer:** Integrates extracted features to model the PA behivior from the convolutional layers' output.
- **Output Layer:** Produces the final output corresponding to the I/Q components of the PA output signal.

The obvious innovation of the RVTDCNN model is the use of the convolutional approach in DPD techniques. After the one-dimensional features from the input signal are formatted into two-dimensional features, the convolution layers use several predesigned filters to extract the input features from the signal; this can significantly enhance the ability to linearize the PAs by capturing the memory effects without increasing the computation load. A fully connected layer will process the convolutional layers' output by integrating the features to predict the final output in the output layer, providing a detailed representation of the PAs' behivior. During the whole process, the activation functions were used to connect each layer and achieve high performance by enhancing the model's ability to handle nonlinear dynamics.

The RVTDCNN model has also been validated in the real world and achieves a comparable or superior performance to other traditional neural network DPD models. Besides that, the RVTDCNN model is fairly suitable for handling higher bandwidth signals, which achieve comparable results with fewer coefficients and lower computation load by processing the convolution. [16] shows that RVTDCNN can effectively improve signal quality by evaluating the normalized mean square error (NMSE) and adjacent channel power ratio (ACPR).

Overall, the RVTDCNN model shows a significant advancement in DPD evolution, especially for wideband applications. RVTDCNN's ability can make it an efficient solution for DPD techniques in modern wireless communication systems. However, RVTDCNN's performance is highly dependent on the convolution process, and this process should be redesigned based on the complex signals in the modern communication system; the memory depth and computation load will be highly increased when the convolution process can not be suited to DPD system well.

2.3. Temporal Convolutional Network (TCN)

When there is a sequential task that needs to be handled by deep learning methods in the real world, Recurrent Neural Networks (RNNs) have always been the preferred choice because of their ability to model time dependencies in a standard way. The real applications, including language translation, time series forecasting, and speech recognition, have benefited by utilizing the sequential processing capabilities of RNNs. With these numerous advantages of RNNs, there are still limitations, like struggling with long-range dependencies and gradients vanishing or exploding problems during the training. These limitations motivate the exploration of more efficient and robust models.

Temporal Convolutional Networks (TCNs) were proposed to address the limitations of RNNs as a fairly powerful alternative. The characteristics of TCNs and RNNs are highly different; TCNs apply a convolutional architecture to parallel the sequence processing and eliminate the gradient problems of RNNs. The core architecture of TCNs incorporates the one-dimensional dilated convolutional layers to expand the receptive field, this method will make the network architecture get longer sequences without losing the resolution or coverage. The causal technique will make the TCNs predict the output based on the past and current values. Therefore, TCN architecture has been considered as an alternative model that can capture long-range dependencies in sequential tasks.

TCNs have been applied to various applications and achieve the state of the art performance, especially outperforming the classical RNNs [17, 18, 26, 27]. Besides that, TCNs provide a more straightforward architecture by using the one-dimensional convolution layers, which also makes the TCNs more scalable and adaptable to have a huge potential in academia and industry applications.

Based on these advancements, this thesis will explore this architecture's core components, the application field, and the advantages, aiming to take their advantages to the DPD application as an efficient and robust solution. Therefore, the systematic analysis of TCNs is necessary, and to know the main working principles, this section will dive into the general TCN architecture.

2.3.1. Sequential modeling

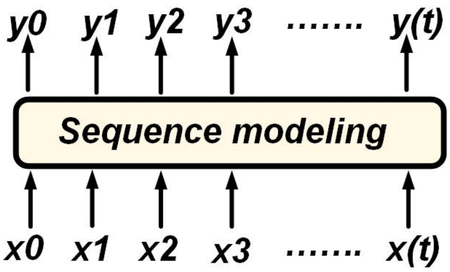

Figure 2.3.1: Basic Concept of Sequence modeling

Sequence modeling is a crucial technique in machine learning that involves predicting future elements in a sequence based on observed elements, considering the order of data points essential. This approach is foundational in domains where temporal dynamics and sequential dependencies govern the system's behavior. There are many applications of sequence modeling in the real world, like Natural Language Processing (NLP) that aims to predict the next word based on the provided previous words, time series forecasting for weather or finance, audio processing that predicts the next note depends on the previous notes in speech system, etc.

There is a main constraint of sequence modeling as Fig 2.3.1 shows: the output sequence length should be the same as the input sequence length, which can make a direct correspondence between the elements of input and output to preserve the temporal structure of the sequence.

2.3.2. Causal and 1D Dilated Convolution

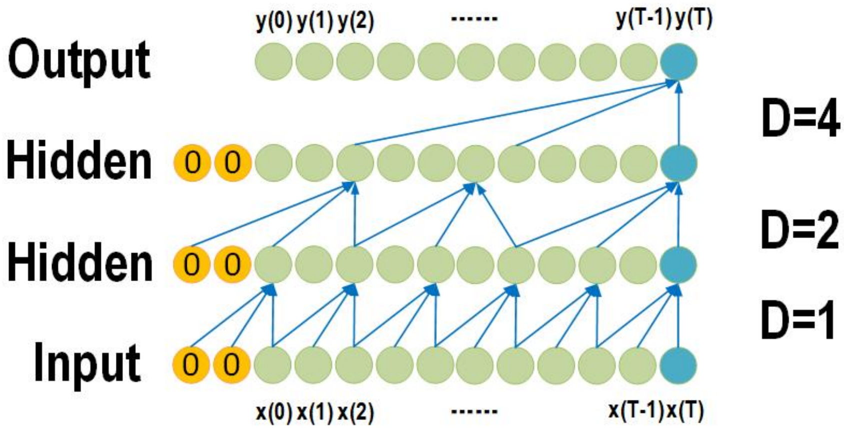

Figure 2.3.2: Causal and 1D Dilated Convolution with dilation factors d=2 and kernel size=3, D represents the dilation size in each layer

Causal convolutions are a type of convolutional operation where each output at time t is computed solely from elements from time t and earlier in the input sequence. This ensures that the model does not inadvertently use future input information to predict the present or past values, adhering to the principle of causality in sequence modeling. This design is crucial in applications like real-time series forecasting where future data points are naturally unavailable [17, 18, 28]. In practical terms, TCNs achieve this causal architecture by padding the input sequence appropriately before each convolutional layer. This padding strategy ensures that the convolution operation does not extend beyond the current time step into the future, maintaining the output length equal to the input length and thereby satisfying both the causality and length constraints fundamental to sequence modeling [29].

Dilated convolutions are employed to enhance the capability of TCNs to capture long-range dependencies within a sequence. A dilated convolution involves skipping input values at regular intervals during the convolution process, effectively enlarging the receptive field of the convolutional filters without increasing their size. This allows the network to integrate information over larger spans of the input sequence with fewer layers, enhancing both the efficiency and scalability of the model [30, 31].

As Fig. 2.3.2 shows, the length of the receptive field is dependent on the dilation size D and kernel size in each layer. The dilation size D is based on the dilation factor, denied as d, which is the stride in the convolutional filer over the sequence. When a dilation factor d=1 means the layer includes a standard convolution, as d increases to 2 like in Fig. 2.3.2, the filter will skip more input elements in the sequence, to increase the receptive field. Therefore, careful consideration is necessary when implementing the causal and dilated convolutions into the TCN architecture. In the first, a larger kernel size can increase the receptive field in each layer but obtain a higher computational complexity. Secondly, the dilation factor can increase the receptive field quickly but has a high potential for overfitting problems with big dilated factors.

In general TCN architecture, the layers are stacked and the dilation size increases with the subsequent layer. Each layer includes both one-dimensional causal and dilated convolution, to obtain the ability to process information over long time steps by making the network depth grow. This exponential growth in the receptive field will make the network cover a very long input sequence to handle more model dependencies, which is crucial for effective learning in sequential tasks which require the information of long-term dependencies like audio processing, natural language processing tasks and weather forecasting.

2.3.3. Residual Block

The core component of TCNs architecture, residual block was proposed to be used in Residual Networks(ResNets) by [32]. With the development of TCNs, the residual block was widely used and helped the TCNs to handle very deep network training effectively. These residual blocks can make the network handle more complex sequential tasks through its ability to access a long history of prior inputs.

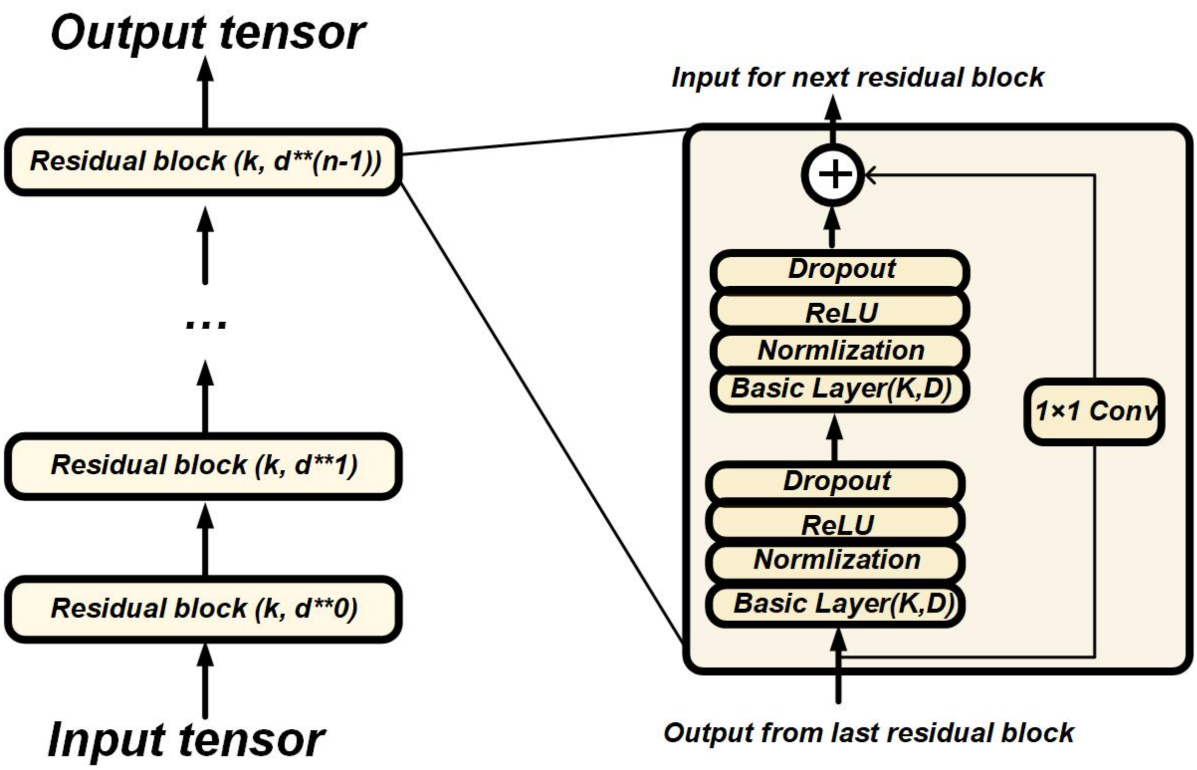

Figure 2.3.3: Residual Block Diagram

The TCNs normally contain a series of components, which are causal dilated convolutional layers, activation functions, normalizations, and shortcut connections by adding the block input to its output. Fig 2.3.3 shows the general formulation of a residual block in a TCN, which also can be expressed as:

- **Dilated Causal Convolutional Layers:**

  In the residual block of TCNs, normally, two dilated causal convolutional layers will be used. These dilated causal convolutional layers will increase the receptive field without losing the resolution or coverage to make the TCNs effectively handle the information from a longer historical context. Since the causal convolution with zero padding will make the input length longer, the dropout technique will keep the same length between the output from the last residual block and input for the next residual block to promote a robust learning process.

- **Activation Function:**

  The Rectified Linear Unit (ReLU) activation function is always used in the residual block. ReLU function can introduce the nonlinearity into the block and capture the complex patterns in the data. However, there are not many explorations that benchmark that different activation functions will affect the TCN's performance. Therefore, the following section results will study the effects of TCN models when the activation functions are different.

- **Normalization:**

  Weight or batch normalizations are applied in the residual block after convolutional layers, and this method will help the training process stale by controlling the weights not to become too large or too small. However, this technique should be dependent on the requirements of the specific sequential tasks.

- **Shortcut Connections:**

  Shortcut connections are always applied in TCNs' architecture to help mitigate the gradient problem by directly adding the input to the output. This skip connection will make it feasible for the TCNs to train more effectively.

Residual block's advantages and disadvantages in TCNs can give a comprehensive view of its ability and adaptability to help the author to design the TCN DPD architecture and list here:

Advantages:

- **High Performance:**

  The residual block in TCNs can handle complex sequential tasks with a deeper network, especially tasks requiring long sequence dependencies. This powerful method will make the prediction more accurate by adding several residual blocks.

- **Avoid the Gradients Problem:**

  By using the skip connection in the residual block, the gradients vanishing or exploding problem will be eliminated.

- **Flexibility and Scalability:**

  The requirements of the receptive field could decide the number of residual blocks in the TCNs' architecture. Different numbers will affect the size of the model, which is dependent on the complexity of the tasks.

Disadvantages:

- **Increased Complexity:**

  The complexity will be increased when the residual blocks are incorporated into TCNs, which can handle the complex sequential tasks. The increased complexity will make the network design complicated and also need more parameters in the learning process. Especially the number of parameters should be limited in the DPD system, which is a big challenge when the author designs the TCN DPD architecture.

- **Potential for Overfitting:** When the sequential tasks are not complex and can not provide large input data to the TCNs, applying residual blocks possibly leads to overfitting. With so many deep learning techniques in the residual blocks, adding more residual blocks will make the TCNs' architecture so deep. Deep architecture probably does not contribute to performance improvements, and it even decreases performance because of complexity. This limitation will also be a key consideration when designing the TCN architecture since the DPD systems do not need a large network architecture.

- **Dependency on Skip Connections:** The TCNs' performance improvements mainly rely on the skip connections in the residual block. In some tasks, the skip connections may not contribute the benefits too much.

TCNs with residual blocks achieve superior performance in sequential tasks in the real world, outperforming recurrent neural networks(RNNs) like LSTMs and GRUs. It shows effectiveness by adding the residual learning techniques into the convolutional architectures. In summary, the integration of residual blocks into the TCN architectures makes a significant advancement in neural network design for sequential tasks. TCNs with residual blocks provide a very good architecture design for large sequential tasks, but limitations should also be considered when designing the TCN DPD model.

3

Temporal Convolution Network Architecture Design

3.1. Noncausal and 1D Dilated Convolution

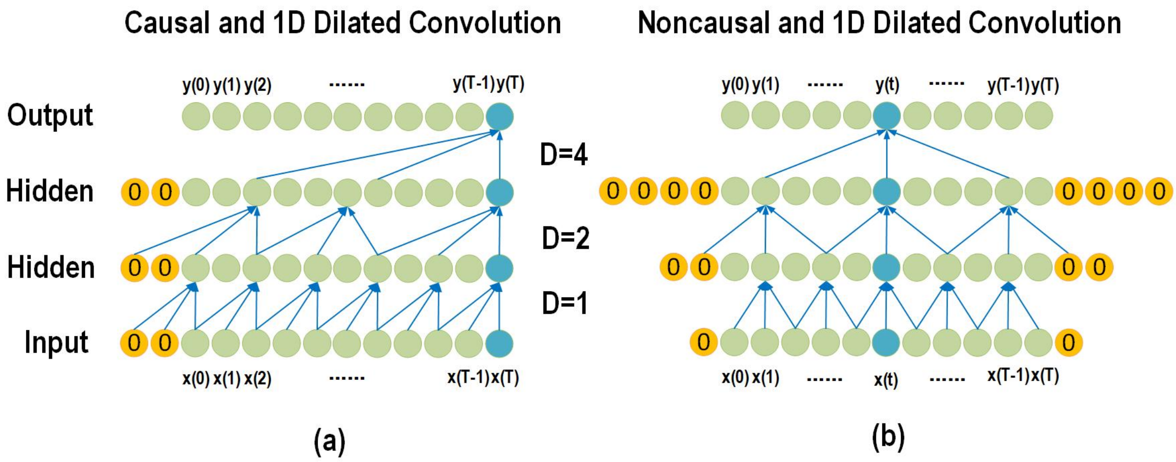

Figure 3.1.1: Causal and Noncausal Convolution

Traditional TCN implementations rely on causal convolution, where network predictions are based exclusively on past data [17, 18, 28]. This design philosophy, inherited from TDNNs, ensures that outputs at time step t depend solely on data from t and earlier timesteps as Fig 3.1.1 shows. Such causal systems require leading zero-padding along the time dimension at each layer of TCN to maintain temporal alignment, as defined by Eq. 3.1:

$$\text{Causal Padding Size} = K - 1 \tag{3.1}$$

where K represents the convolutional kernel size. While one-dimensional fully-convolutional networks (FCN) can partially address this challenge by maintaining consistent input-output sequence lengths [29], they still demand significant computational resources. Even the causal way needs more computation resources and compensates for the accuracy, but it is still used popularly. One of the most significant reasons that most TCN architectures use the causal convolution consistently is that it can adapted to real-world applications, like speech recognition and weather forecasting, etc. These applications are all real-time scenarios and only capture current and past information; future information cannot be accessed.

Since the dataset 200_MHZ from the OpenDPD framework has already recorded all the signal information [20]. From this perspective, the signal information from the future can be captured during the training. Therefore, the approach implements noncausal convolution since the simulation results are more focused in this thesis, which reduces the required padding size according to Eq. 3.2:

$$\text{Noncausal Padding Size} = \frac{K-1}{2} \tag{3.2}$$

This noncausal design enables the TCN to capture both past and future temporal dependencies within each I/Q data context window of the convolutional kernel size, thereby improving DPD modeling accuracy. Also, it decreases the computation resources with different padding ways compared to causal convolution.

Sequential tasks requiring extensive historical context often challenge standard convolutional architectures, which struggle to capture long-range temporal dependencies [28]. This TCN architecture addresses this limitation by incorporating dilated convolutions, building upon established approaches [30, 31]. These dilated convolutions expand the network's receptive field exponentially without requiring additional layers or larger kernels. By progressively increasing dilation sizes across layers, the network efficiently processes extended temporal sequences while maintaining parameter efficiency. Each input within the effective history receives processing from at least one kernel, ensuring comprehensive modeling of long-term dependencies. Therefore, the dilated convolution will still be used in this work. Fig. 1.4.1 demonstrates how this dilated structure systematically expands the receptive field across network layers.

3.2. Proposed Architecture

The TCN-DPD architecture, illustrated in Fig. 3.2.2, comprises three main components: input and output 1×1 convolution layers sandwiching a core Depthwise Separable Convolution Block, with activation functions (Act. Funcs.) connecting each layer to enable nonlinear modeling.

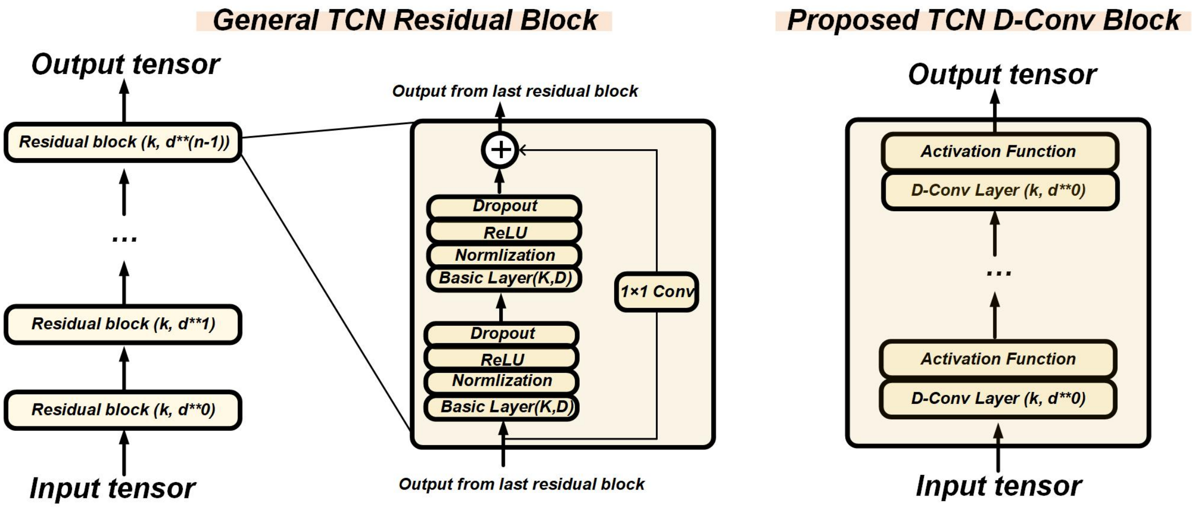

Figure 3.2.1: General TCN Residual Block and Proposed D-Conv Block Comparison

3.2.1. 1*1 Convolution Layer

The network employs 1×1 convolution layers at its input and output boundaries. These layers serve as dimensional adaptation interfaces, efficiently transforming feature representations between the network's internal processing stages. Not only does the feature integration function in the proposed architecture, but it also keeps the computation efficient with less multiplication during the training, which is helpful for the proposed model parameter efficiency.

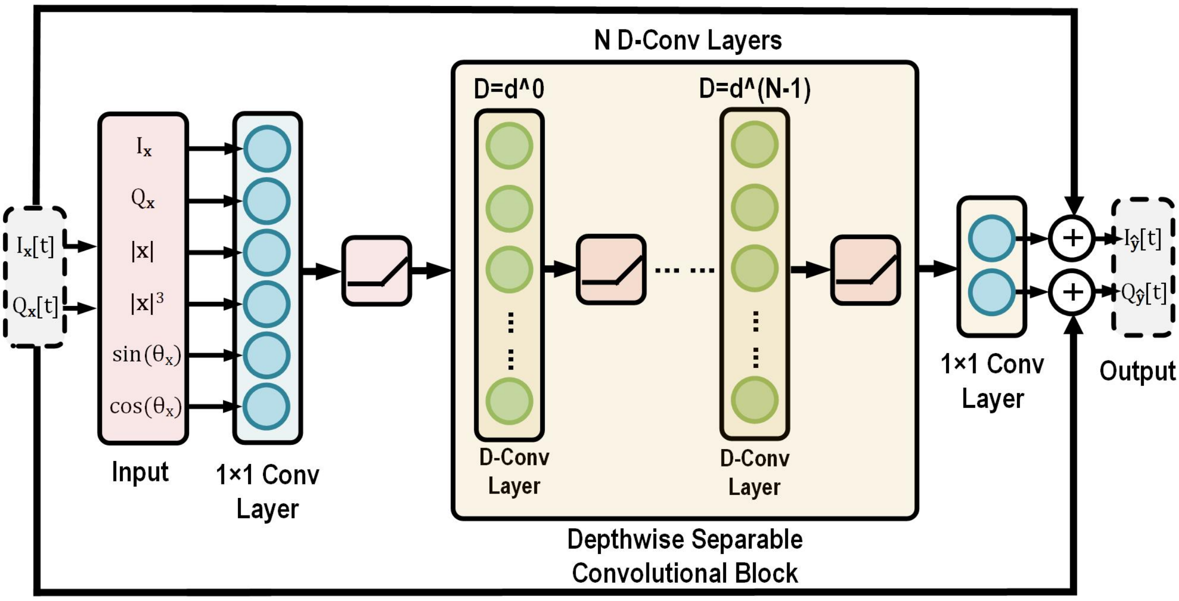

Figure 3.2.2: The residual TCN architecture with depthwise separable convolutional (D-Conv) layers. $D = d^{N-1}$, where D is the dilation size of each D-Conv layer, d is the dilation base, and N is the number of D-Conv layers

3.2.2. Depthwise Separable Convolution Block

When the traditional TCNs' architecture usually uses the residual block, as section 2.3.3 explained, it can achieve superior performance in many sequential tasks like speech recognition, etc. However, the DPD technique is highly affected by parameter efficiency, but the usual residual block will lead to an increase in depth in the architecture with more parameters. Therefore, the residual block will be decomposed into one depthwise separable convolution block layer with activation function as Fig 3.2.1 show. In this work, the architecture's core lies a Depthwise Separable Convolution Block featuring multiple depthwise separable convolution layers [33]. Each layer implements progressively larger dilation sizes, enabling efficient capture of temporal information through an expanding receptive field as Fig. 3.2.2 presents. The padding size for these dilated layers follows Eq. 3.3:

$$\text{Dilated Noncausal Padding Size} = \frac{K-1}{2} \times D \tag{3.3}$$

where D represents the dilation size and K denotes the kernel size. Each layer incorporates an activation function to model complex nonlinear signal patterns.

3.2.3. Input Features Extracted from I/Q

The network processes both I and Q, along with their derived features, as illustrated in the input layer shown in Fig. 3.2.2. This enriched input representation provides comprehensive amplitude and phase information as sections 2.2.2 and 2.2.3 mentioned when introducing the DGRU and RVTDCNN DPD model.

3.2.4. Residual Connections

The architecture implements residual connections directly linking I and Q inputs to the output, enhancing the model stability in training. During the training, early layers will update the weight difficulty since the gradients are small when propagating backwards. These connections preserve critical input information while addressing the vanishing gradient challenges common in deep architectures [32].

4

Experimental Results

4.1. Experimental Setup

There are three main benchmarks in this section, which are the benchmark of Activation Functions, PA benchmarking, and DPD benchmarking. And there are some common experimental setups could be indicated in the beginning:

- **Simulation Framework and Data Collection:**

  All experiments were conducted using the Open-DPD framework, which holds a status and function analogous to that of the MNIST [34] dataset in the field of deep learning for DPD applications. It was also conducted using the `DPA_200MHz` dataset, which contains 200 MHz bandwidth signals (10 channels × 20 MHz) with 64-QAM OFDM modulation, measured from a 40 nm digital transmitter [35]. While simulated results may differ from measured performance, previous research has demonstrated that the relative performance rankings among different architectures remain consistent between simulation and measurement, validating our simulation-based evaluation approach [20].

- **Training and Validation:**

  The DPA_200MHz dataset will be divided into three parts, including the training, validation, and testing sets. The training set is designed to train the models, the validation set can be used to evaluate the models initially, and the testing set is served to evaluate the models robustly as a thorough assessment.

- **Random Seeds:**

  The random seeds will be applied to initialize the weights of the neural networks. This method in the benchmarks is crucial to assess the variability in model performance and ensure that the results are not biased by specific conditions. In all benchmarks, the random seeds will be set with 5.

- **Performance Evaluation:**

  Some metrics will be used to evaluate the performance of different DNN-based DPD models, including the adjacent channel power ratio (ACPR), error vector magnitude (EVM), and normalized mean square error (NMSE). NMSE metric is used to evaluate the accuracy of PA behaviours. ACPR metric is crucial for evaluating the interference caused in the different DNN-based DPD models. EVM, which is used to measure the constellation error in signal transmission. Lower EVM shows a clearer and more accurate signal representation, this metric is important for evaluating the high quality of signals.

Benchmark of Activation Functions

After establishing the core TCN architecture, a systematic investigation of activation functions will be conducted to optimize network performance. Traditional TCN implementations often employ ReLU or its variants [17, 28]. However, these functions present two significant limitations for DPD applications: their inability to process negative values effectively and their non-smooth characteristics, which can introduce spectral leakage. These limitations typically require additional network parameters to model the target function accurately.

To identify the most effective activation function, 22 different options will be evaluated by using a baseline TCN configuration with 4 depthwise convolution layers, kernel size $K = 5$, and dilation base $d = 2$. Each activation function was tested across 5 random seeds, with results presented in Fig. 4.2.1. To ensure reliability, the averaged performance metrics across all seeds, as detailed in Table 4.2.1.

Power Amplifier Benchmarking

When the selection of activation functions was decided and the TCN architecture was finalized, the next benchmark was planned to use this configuration to evaluate the behavior of power amplifiers (PAs). This power amplifier benchmark is crucial since it can evaluate the efficacy of the TCN-based DPD approach in the simulation and check whether the implementation was successful. The main metric will be the Normalized Mean Square Error (NMSE), which shows the model's accuracy in correcting the power amplifier distortions. The compared models include the LSTM, GRU, DGRU, VDLSTM, PN-TDNN and RVTDCNN, these established models will be compared with the TCN model by evaluating the NMSE results. By analyzing the NMSE simulation results across the different models, the superior capability of the TCN DPD model in capturing the memory effects and mitigating the non-linearities will be assessed.

Digital Predistortion Benchmarking

After the benchmarking of various PA models, the DGRU and the designed TCN models are the pre-trained PA models to evaluate the DPD performance further. There are several reasons for choosing these two architectures. Like the choice of DGRU architecture, there are two purposes. The first is to compare the results directly with previous research in existing literature. Secondly, with the same experiment setup condition, offer a benchmark to ensure the findings are reliable and robust. The choice of the TCN PA model is mainly dependent on its demonstrated potential to simulate superior performance in modeling PA behavior. Using these two pre-trained PA models as references, DPD benchmarking can provide a comprehensive understanding of the TCN-based DPD approach compared to traditional DNN-based DPD models. The strengths and limitations of the TCN DPD approach can be analyzed based on the results to explore its potential to linearize the PA in the real world.

When the pre-trained PA models are selected, the evaluated configurations range from 200 to 1000 parameters against other state-of-the-art DNN-based DPD models, the same as the PA benchmarking. Each configuration underwent testing across the setup of random seeds to ensure statistical significance and choose the best model over 5 random seeds in the validation set. To consolidate the results from the validation set, choose the configuration of around 500 parameters and average the results on the test set over all random seeds for a robust analysis.

4.2. Benchmark of Activation Functions

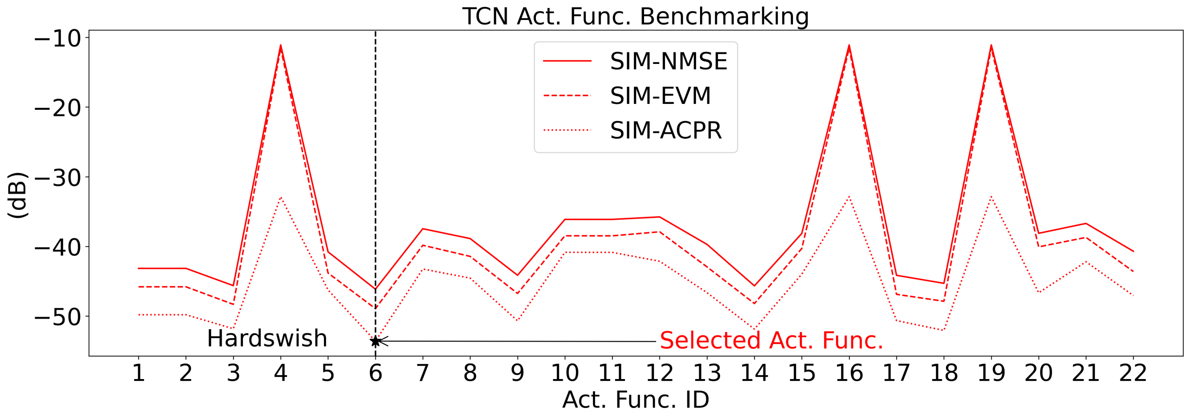

Figure 4.2.1: TCN Act. Func. benchmarking with a fixed DGRU PA pre-trained Model on the dataset of DPA_200MHz validation, each curve stands the optimal performance of different Act. Func. across 5 random seeds.

Table 4.2.1: DPD Performance Comparison of 22 Act. Funcs. Based on a Fixed DGRU PA Pre-trained Model on the dataset of DPA_200MHz Validation, Averaged across 5 Random Seeds

ID	1	2	3	4	5	**6**	7	8	9	10	11
Act. Func.	CELU	ELU	GELU	Hardshrink	Hardtanh	**Hardswish**	LeakyReLU	LogSigmoid	Mish	ReLU	ReLU6
SIM-NMSE (dB)	-43.62	-43.62	-44.79	-11.10	-42.00	**-44.89**	-38.20	-35.62	-44.33	-36.74	-36.77
SIM-ACLR (dBc)	-50.21	-50.20	-51.52	-32.92	-48.00	**-51.54**	-44.16	-40.88	-51.17	-42.53	-42.53
ID	12	13	14	15	16	17	18	19	20	21	22
Act. Func.	RReLU	SELU	SiLU	Softplus	Softshrink	Softsign	Tanh	Tanhshrink	Hardsigmoid	Sigmoid	PReLU
SIM-NMSE (dB)	-37.25	-41.08	-44.70	-38.59	-11.10	-43.65	-44.58	-11.10	-38.05	-35.62	-41.70
SIM-ACLR (dBc)	-43.06	-47.53	-51.54	-44.02	-32.92	-50.14	-51.37	-32.92	-44.88	-40.88	-47.98

[a] TCN architecture set up with 4 depthwise convolution layers, kernel size $K = 5$, and a predefined dilation base $d = 2$.

Fig. 4.2.1 and Tab. 4.2.1 reveal significant insight into the model performance with different activation functions. The initial comparative evaluation of activation functions was presented in Fig. 4.2.1 by choosing the best model of different architecture, demonstrating that the Hardswish activation function achieves the best performance across all the metrics, including NMSE, ACPR, and EVM. To further validate the results from Fig. 4.2.1, Tab. 4.2.1 shows the averaged results on the test set and establishes the Hardswish activation function's robust performance by achieving SIM-NMSE -44.89 dB and SIM-ACLR -51.54 dBc. Even the optimized activation function was selected, there are three activation functions GELU, SiLU, and Tanh, also demonstrate strong performance with averaged results SIM-NMSE -44.79 dB and SIM-ACPR -51.52 dBc, SIM-NMSE -44.70 dB and SIM-ACPR -51.54 dBc, SIM-NMSE -44.58 dB and SIM-ACPR -51.37 dBc.

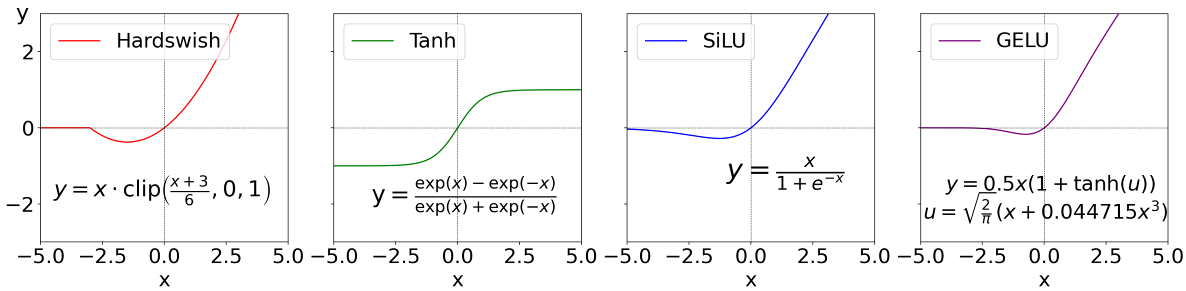

Figure 4.2.2: The Top-4 performing activation functions in Table 4.2.1

Fig. 4.2.2 shows four activation functions, which achieved good simulated results in the benchmark. When looking at the schematic diagram of these activation functions, there are some important common characteristics. First, these functions are differentiable across their domains. Secondly, the range of the domain accepts both positive and negative input values and has smooth gradient transitions. These properties make these four activation functions specifically well-suited for the DPD techniques, offering viable alternatives to the commonly used ReLU variants in deep learning architectures.

4.3. Power Amplifier (PA) Modelling

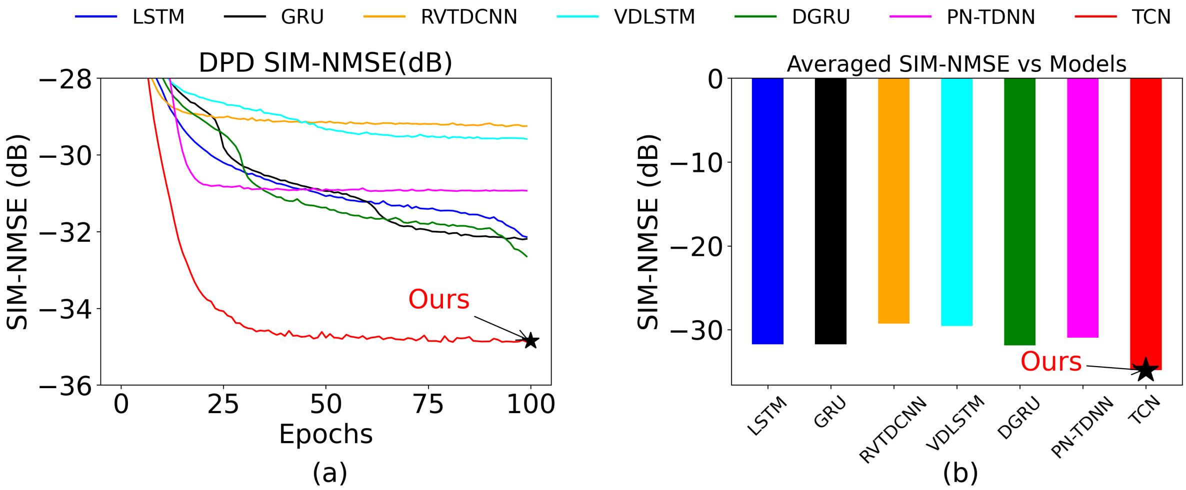

Figure 4.3.1: The 500-parameter PA modeling NMSE over training epochs simulation results on OFDM signals at 200 MHz, 10 channels by 20 MHz from the dataset of DPA_200MHz validation and test with 5 different random seeds (a) Optimal PA model selected from 5 random seeds on the validation set over 100 training epochs. (b) Averaged NMSE comparison with different models on the test set.

The benchmark of PA behavioral modeling mainly aims to find whether performance enhancement exists when using the Temporal Convolutional Network (TCN) architecture. As illustrated in Fig. 4.3.1(a), various DNN-based models' performance of PA were compared across 100 training epochs over 5 random seeds. The TCN model, also labeled as "Ours," shows superior performance by using the NMSE metric. This first benchmark of PA modeling on the validation set provides evidence that TCN architecture could be implemented to linearize the PAs successfully, and it shows great improvement.

Fig. 4.3.1(b) and Tab. 4.3.1 further consolidate these findings by comparing the averaged NMSE results across different models on a test dataset. The bar diagram illustrates the different models' performance and the TCN model as a leader, surpassing all the traditional DNN models by evaluating the NMSE metrics. Table 4.2.2 provides a comprehensive comparison insight of Normalized Mean Square Error (NMSE) and Error Vector Magnitude (EVM) across models, where the TCN-500 model again stands out with an NMSE of -34.99 dB and an EVM of -39.57 dB. These results are not only better than those achieved by the LSTM and GRU models, which recorded SIM-NMSE of -31.85 dB and -31.84 dB, respectively, but also superior to the other specialized DNN architectures like RVTDCNN and PN-TDNN with the simulated results SIM-NMSE -28.98 dB and -29.39 dB.

Table 4.3.1: Performance Comparison of PA Modelling with estimated 500 Real-Valued Parameters on `DPA_200MHz` Test Set, Averaged Over 5 Random Seeds ± Standard Deviations

Classes	PA Models	SIM-NMSE (dB)	SIM-EVM (dB)
Base RNN	LSTM	-31.85±0.21	-33.97±0.84
	GRU	-31.84±0.54	-34.28±1.07
Prior Model	RVTDCNN [16]	-28.98±0.07	-30.98±0.07
	VDLSTM [14]	-29.39±0.09	-31.39±0.11
	PN-TDNN [12]	-30.98±0.02	-33.01±0.01
	DGRU [20]	-31.94±0.78	-34.02±1.43
This Work	**TCN-500**	**-34.99±0.06**	**-39.57±0.11**

This significant improvement in NMSE metrics of PA behavioral modeling highlights the TCN architecture's capability to handle complex non-linearities and memory effects more efficiently. Overall, the TCN-based model shows a distinctive advantage in PA modeling, providing substantial improvements over the traditional DNN models. These results prove the hypothesis that TCNs provide a robust and effective solution for linearizing the Power Amplifiers(PAs) in communication systems.

4.4. Digital Predistortion (DPD) Benchmarking

4.4.1. DPD benchmarking based on DGRU PA model

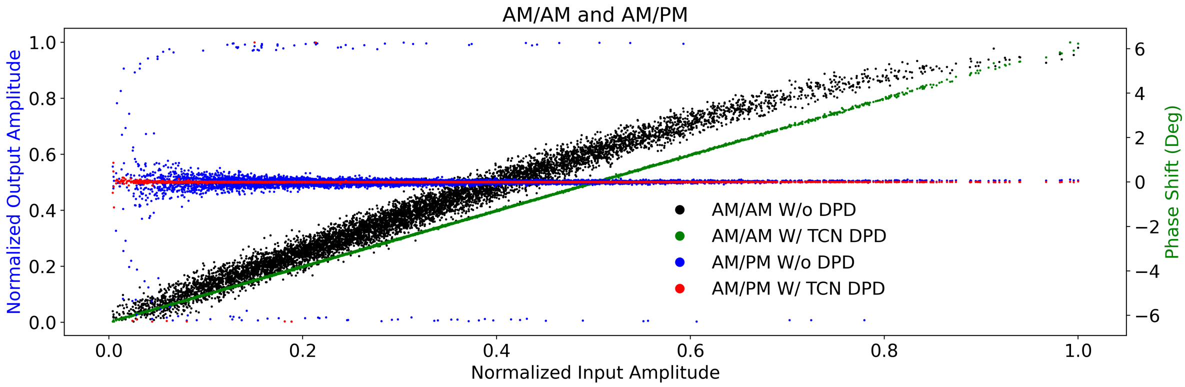

Figure 4.4.1: The 500-parameter DPD modeling AM/AM and AM/PM plot comparison between TCN DPD and Without DPD, TCN training simulation results on OFDM signals at 200 MHz, 10 channels by 20 MHz from the dataset of DPA_200MHz validation, the optimal performance TCN model was selected across 5 random seeds with a fixed DGRU PA model.

Fig. 4.4.1 illustrates the amplitude-to-amplitude (AM/AM) and amplitude-to-phase (AM/PM) plots with and without the DPD technique. The TCN DPD shows a remarkable improvement in linearity, indicating the effective correction of nonlinear distortions in power amplifiers (PAs). This AM/AM and AM/PM plot also confirm that implementing the TCN DPD model successfully addresses the inherent nonlinear characteristics of the Power Amplifiers (PAs), enhancing the signal integrity.

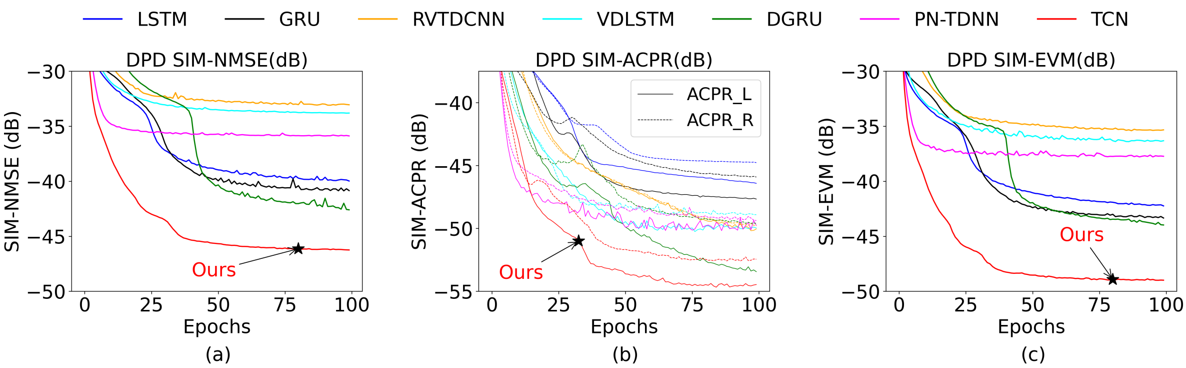

Figure 4.4.2: Training simulation results on OFDM signals at 200 MHz, 10 channels by 20 MHz from the dataset of DPA_200MHz validation. Each curve stands the optimal performance of each architecture across 5 random seeds with a fixed DGRU PA model.
(a) DPD NMSE comparison (b) DPD ACPR comparison (c) DPD EVM comparison

When the reliability of the TCN-based DPD model was initially established, subsequent experiments in the DPD benchmarking should be the DPD performance comparison by evaluating different metrics, including the Normalized Mean Square Error (NMSE), Adjacent Channel Power Ratio (ACPR), and Error Vector Magnitude (EVM) across different DNN based DPD models, like LSTM, GRU, RVTDCNN, VDLSTM, DGRU, PN-TDNN, and this work TCN. Fig. 4.4.2(a), (b), and (c) highlight the TCN's superior performance over the 100 epochs in all metrics, as the label "Ours." Each curve presents the best architecture performance with around 500 real-valued parameters over 5 random seeds. In this experiment, the TCN model outperforms all standard DPD metrics, affirming TCN's potential as a leading approach in DPD techniques for PA linearization.

Table 4.4.1: Performance Comparison of DPD Models Based on a Fixed DGRU PA Model with estimated 500 Real-Valued Parameters on the dataset of DPA_200MHz Test, Averaged across 5 Random Seeds ± Standard Deviations

Classes	**DPD Models**	**SIM-NMSE (dB)**	**SIM-ACPR (dBc, L/R)**	**SIM-EVM (dB)**
W/o DPD[a]	-	-	-31.90±0.24 / -30.45±0.31	-34.02±1.42
Base RNN	LSTM	-35.22±3.86	-43.60±1.14 / -42.68±0.36	-37.52±5.11
	GRU	-40.01±1.68	-44.95±0.65 / -43.76±1.60	-42.70±2.23
Prior DPD	RVTDCNN [16]	-32.03±1.19	-48.04±0.71 / -46.26±1.27	-34.61±1.81
	VDLSTM [14]	-32.50±0.71	-47.04±1.44 / -45.85±1.32	-34.94±1.54
	PN-TDNN [12]	-35.49±0.47	-49.25±0.64 / -48.43±1.11	-37.70±0.89
	DGRU [20]	-41.82±2.87	-50.57±1.82 / -49.16±1.67	-44.04±2.38
This Work	**TCN-500**	**-44.61±1.37**	**-51.58±2.84 / -49.26±2.04**	**-47.52±1.49**
	TCN-200	-41.27±1.55	-45.83±2.39 / -46.76±1.23	-43.81±1.67
	TCN-1000	-46.37±1.13	-52.58±2.43 / -50.84±1.44	-49.40±1.90

The following benchmarking experiment averages the results on the test set to prove the robustness of the TCN-based DPD model. This experiment will confirm that the TCN DPD model's superior performance is not the result of overfitting or a particular seed but is consistent when various conditions change.

Table 4.4.1 presents a comprehensive DPD performance comparison of different DNN-based DPD models, including the traditional, prior and TCN DPD models across all the metrics. The Tab. 4.4.1 also shows TCN model shows remarkable performance, such as the TCN-500 model with achievement SIM-NMSE of -44.61 dB, SIM-ACPR (L/R) of -51.58/-49.26 dBc, and SIM-EVM of -47.52 dB. These averaged results not only demonstrate the TCN DPD model's capability in linearizing the PA but also its efficiency in signal integrity. Regardless of the base DPD model or the advanced models like PN-TDNN and DGRU, the TCN model significantly improves by capturing the memory effects more efficiently.

These results robustly present that the integration of TCN into the DPD systems significantly improves PA performance, providing substantial enhancements in linearity and signal quality. In summary, the TCN model's capability outperforms established DPD models consistently across various test conditions, establishing the TCN model as a promising solution for PA linearization in the real world.

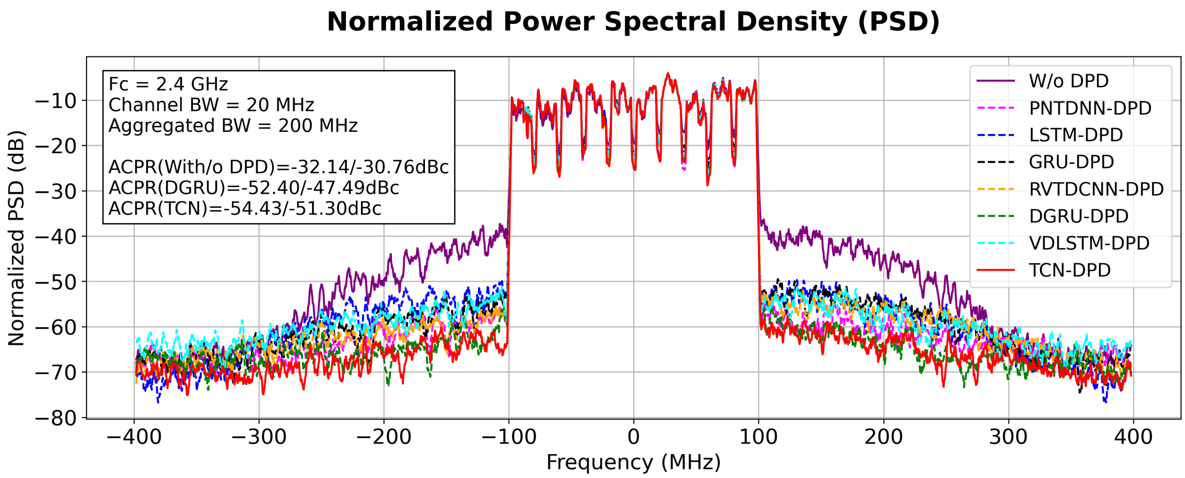

Figure 4.4.3: PSD comparison of the output signal with different 500 real-valued parameter DNN-DPD models on a 200 MHz signal from the dataset of DPA_200MHz test. Each curve stands the best DPD performance of each model across 5 random seeds based on a fixed DGRU PA model.

Fig 4.4.3 illustrates the Power Spectral Density (PSD) comparison of the output signal across different DNN-based Digital Predistortion (DPD) models implemented on a 200 MHz signal from the DPA_200MHz test set. Each curve in the graph represents the best performance achieved by each model over five random seeds based on a fixed DGRU PA model.

Fig. 4.4.3 depicts how different DNN-based DPD models manage spectrum utilization and interference. This plot clearly shows the TCN DPD model achieves superior performance by reducing the out-of-band leakage compared to models without DPD and other DNN-based DPD. The PSD plot demonstrates that the TCN DPD can suppress the spectral regrowth while maintaining a clearer spectral output. The SIM-ACPR improvements are most obvious with the TCN's result SIM-ACPR -54.43/-51.30 dBc, significantly better than other DPD models, highlighting its capability to reduce interference.

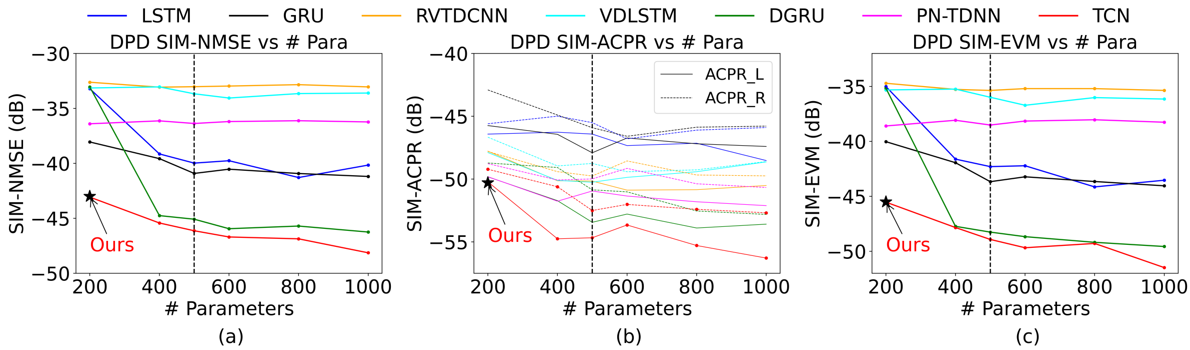

Figure 4.4.4: DPD performance comparison across different parameter configurations based on a fixed DGRU PA model (a) DPD NMSE comparison (b) DPD ACPR comparison (c) DPD EVM comparison

Fig. 4.4.4 illustrates the comprehensive DPD performance comparison across different parameter configurations based on the DGRU pre-trained PA model. The SIM-NMSE, SIM-ACPR, and SIM-EVM are still the standard metrics in the evaluation. Fig. 4.4.4 shows that this thesis's work TCN-based DPD model consistently outperforms other previous architecture across parameter ranges from 200 to 1000. With the real-valued parameters 500, Fig. 4.4.4 and Tab. 4.4.1 show that the TCN DPD model performs best with the same number of parameters and conditions. This means that the performance could be sacrificed to reduce the computation load in the TCN DPD model and provide a low-complexity model with comparable performance. With just 200 parameters, Table 4.4.1 and Fig 4.4.4 show that TCN-DPD achieves remarkable performance with average ACPR (L/R) of -45.83 and -46.76 dBc, and an average EVM of -43.81 dB. This achievement is particularly significant as competing architectures struggle to maintain comparable EVM performance at such low parameter counts, the lower parameters highlighting the TCN's potential as an efficient DPD solution while maintaining parameter efficiency.

4.4.2. DPD based on TCN PA model

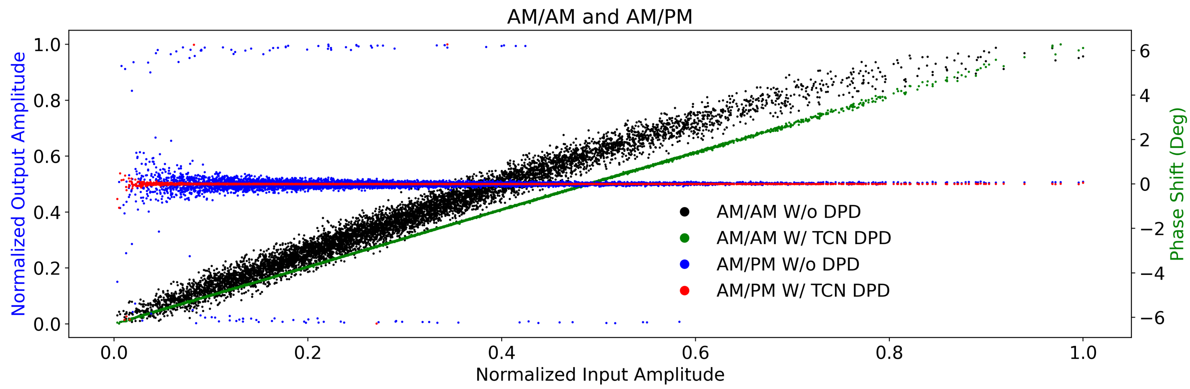

Figure 4.4.5: The 500-parameter DPD modeling AM/AM and AM/PM plot comparison between TCN DPD and Without DPD, TCN training simulation results on OFDM signals at 200 MHz, 10 channels by 20 MHz from the dataset of DPA_200MHz validation, the optimal performance TCN model was selected over 5 random seeds with a fixed TCN PA model.

Fig 4.4.5 also shows the AM/AM and AM/PM conversion characteristics with and without DPD, the benchmark step is the same as the section 4.4.1 but with a different pre-trained PA model. The TCN

DPD also shows a significant improvement in linearity, indicating effective correction of nonlinear distortions of power amplifiers(PAs).

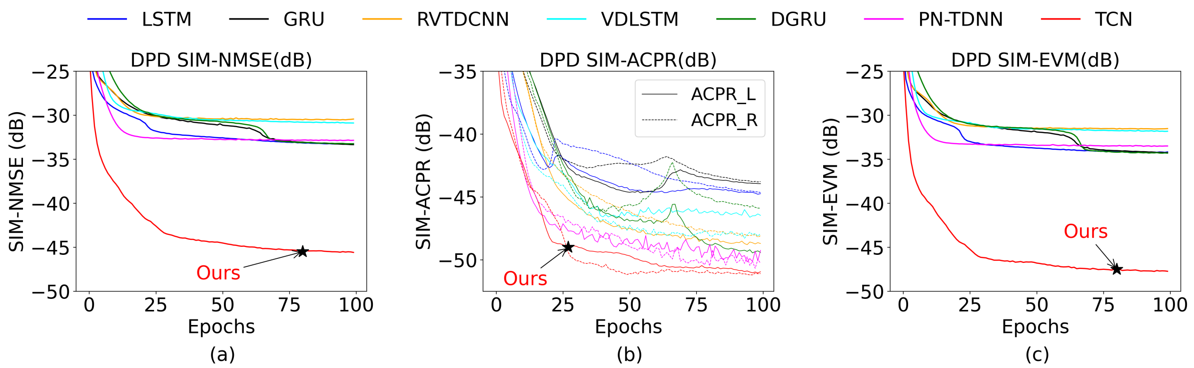

Figure 4.4.6: Training simulation results on OFDM signals at 200 MHz, 10 channels by 20 MHz from the dataset of DPA_200MHz validation. Each curve stands the optimal performance of each architecture across 5 random seeds with a fixed TCN PA model.
(a) DPD NMSE comparison (b) DPD ACPR comparison (c) DPD EVM comparison

This benchmark is the same as section 4.4.1 and different from using a TCN pre-trained PA model; the results show that TCN also outperforms the previous models. Specifically, the TCN architecture is beneficial when the PA and DPD are both based on TCN. This result shows that the DPD performance will be affected when the PA behavioral modeling achieves better or worse performance. From Fig. 4.4.6, we know that the DPD will become hard to train when the pre-trained PA model performs better.

The big difference is the results of SIM-NMSE and SIM-EVM when using a TCN pre-trained PA model instead of the DGRU model. The SIM-NMSE and SIM-EVM results of the TCN DPD model are fairly higher than those of other traditional DNN-based DPD models. It shows that the choice of PA modeling significantly influences the performance of DPD, as Fig. 4.4.6 shows. As a result, the benchmarking demonstrates that the TCN base DPD model significantly improves signal integrity. This robust and improved performance based on the TCN-based PA model establishes the TCN architecture as a powerful approach for future DPD techniques to meet the requirements of advanced communication systems.

Table 4.4.2: Performance Comparison of DPD Models Based on TCN PA Model with Approximately 500 Real-Valued Parameters on the dataset of `DPA_200MHz` Test Averaged on 5 Random Seeds ± Standard Deviations.

Classes	DPD Models	SIM-NMSE (dB)	SIM-ACPR (dBc, L/R)	SIM-EVM (dB)
Without DPD[a]	-	-	-31.90±0.06/-30.44±0.27	-39.57±0.12
Base RNN	LSTM	-32.77±0.68	-43.60±0.27/-42.48±0.67	-33.77±0.72
	GRU	-33.82±1.72	-43.65±0.99/-42.99±0.8	-34.84±2.16
Prior DPD	RVTDCNN [16]	-31.18±0.07	-48.65±0.49/-47.22±0.38	-31.55±0.08
	VDLSTM [14]	-30.58±0.17	-46.93±0.83/-46.31±1.31	-31.85±0.16
	PN-TDNN [12]	-32.57±0.32	-48.70±0.91/-49.43±1.06	-33.47±0.54
	DGRU [20]	-32.16±1.35	-47.90±1.03/-45.91±0.57	-33.31±1.32
This Work	**TCN-500[d]**	**-45.51±0.80**	**-50.39±1.19/-50.01±0.87**	**-47.88±0.73**
	TCN-200[d]	-43.51±0.57	-47.46±0.63/-48.12±1.29	-46.11±0.22
	TCN-1000[d]	-46.90±0.29	-52.35±0.71/-51.96±0.60	-49.19±0.33

Table 4.4.2 shows the DPD performance comparison based on the TCN PA model on average for a robust evaluation as in section 4.4.1. Not surprisingly, the TCN-DPD model still achieves the best performance based on the SIM-NMSE -45.51 dB, SIM-ACPR -50.39/-50.01 dBc (L/R), and SIM-EVM -47.88 dB when compared with other traditional DNN-based DPD models. When looking at Tab 4.4.2, it is easy to find out that TCN-DPD has a big improvement in SIM-NMSE and SIM-EVM metrics; other models normally achieve the SIM-NMSE and SIM-EVM around -35 dB; this number is fairly close to the real-world measurement in previous research [20]. These results probably show that the simula-

tion results are getting close to the measurement when the PA behavioral modeling achieves better performance.

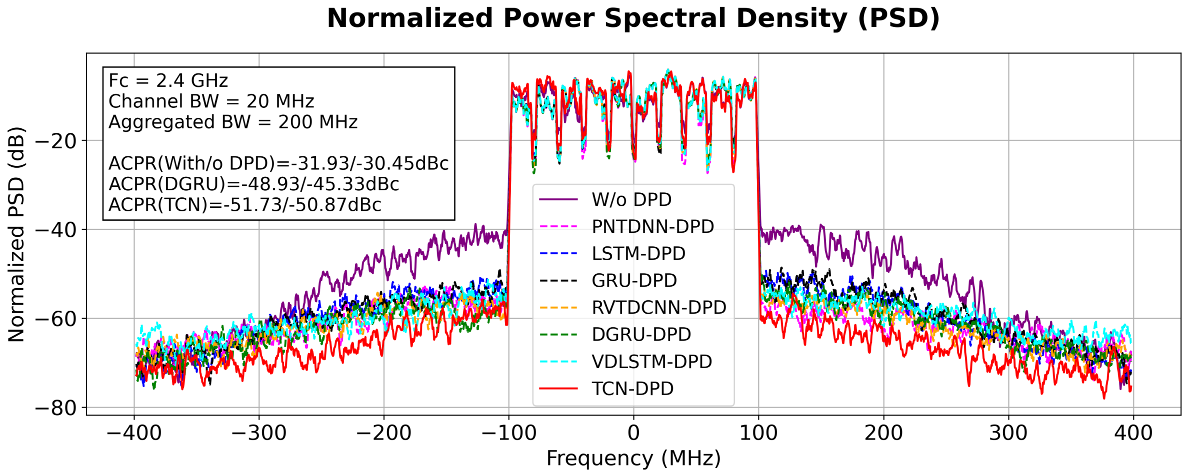

Figure 4.4.7: PSD comparison of the output signal with different 500 real-valued parameter DNN-DPD models on a 200 MHz signal from the dataset of `DPA_200MHz` test. Each curve stands the optimal DPD performance of each model across 5 random seeds based on a fixed TCN PA model.

Like the last subsection, Fig 4.4.7 illustrates the Power Spectral Density(PSD) plot. The graph demonstrates that the TCN-DPD model, indicated by the distinctive curve, consistently achieves superior spectral containment compared to traditional and other neural network-based DPD models. Specifically, the SIM-ACPR values for the TCN model with -51.73/-50.87 dBc further confirm its efficiency in maintaining a cleaner spectrum compared to other models where values, such as -31.97/-30.45 dBc without DPD and -48.93/-45.33 dBc for DGRU, shows higher spectral leakage. This analysis proves the robustness of the TCN-based DPD model in providing effective linearization with improved spectral efficiency, making it a highly agreeable solution for modern communication systems that need stringent spectral compliance.

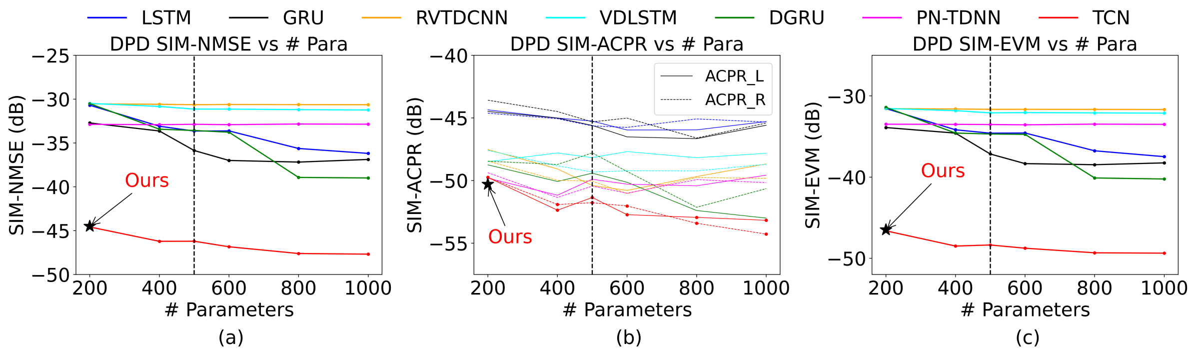

Figure 4.4.8: DPD performance comparison across different parameter configurations based on a fixed TCN PA model

Fig 4.4.8 have the same benchmark in section 4.4.1 but a different PA model, which is TCN. The results are also quite similar to Fig 4.4.4, the TCN-DPD model still achieves the best performance from arranging 200 to 1000 parameters. The only difference is that SIM-NMSE and SIM-EVM results are fairly improved. When the number of parameters is small, like 200 and 400, it can still be approved that the TCN-DPD model is parameters efficient and reduces computation load.

5 Conclusion and Recommendation

5.1. Conclusion

This chapter draws the conclusion of the whole thesis based on the contribution and recommendation for future study.

To address the gap between the TCN-based DPD model, this thesis proposed a novel idea by implementing the TCN with noncausal dilated convolution into the DPD system. Following the whole process in the thesis, the major conclusions are:

1. This thesis has successfully demonstrated the effective integration of Temporal Convolutional Networks (TCNs) into Digital Predistortion (DPD) systems for linearizing power amplifiers (PAs), addressing the complexities of non-linearities and memory effects prevalent in wideband applications. The research presented here not only validates the potential of TCNs in enhancing PA performance but also underscores the model's superior parameter efficiency compared to previous methods.

2. Throughout this study, an exploration was made to discover the effects of the activation function on DPD performance. Hardswish was identified as particularly effective, enhancing the model's performance across various benchmarks. Additionally, Hardswish, Tanh, SiLU, and GELU were considered to be affable for the DPD techniques based on evaluating the averaged SIM-NMSE and SIM-ACPLR on the benchmark in section 4.2. These four activation functions show similar characteristics, such as input acceptation range and smooth curves. These results prove that the activation functions with these characteristics are helpful for DNN-DPD design.

3. The thesis also provides the main experiment, DPD benchmarking. Two architectures were selected for the pre-trained PA model, which is the DGRU and TCN model. When the pre-trained PA is fixed as the DGRU model, TCN-DPD demonstrates superior linearization performance with only 500 real-valued parameters, achieving averaged and simulated ACPR of -51.58/-49.26 dBc (L/R), EVM of -47.52 dB, and NMSE of -44.61dB. The results are simulated ACPR of -50.39/-50.01 dBc (L/R), EVM of -47.88 dB, and NMSE of -45.51 dB in average when the pre-trained PA model is TCN. Both DPD benchmarks include different DNN-DPD models. TCN-DPD has superior performance in the comparison, especially the SIM-NMSE and SIM-EVM performance is significantly higher than other models when the pre-trained PA model is TCN. These results indicate that the TCN-DPD model is fairly outstanding compared to previous DNN-based DPD models.

4. In the benchmark, the evaluation was extended to DNN-DPD performance with various parameters ranging from 200 to 1000; the TCN-DPD model achieved impressive results with all metrics and surpassed all previous DNN-DPD models with different parameters. With the same number of real-valued parameters in the benchmark, the performance of TCN DPD is higher than that of other models. This is a pivotal finding as it suggests that TCNs can achieve comparable performance in PA linearization with fewer parameters, thus reducing the computational load of the DPD system. where the TCN-200 model highlighted its effectiveness by showing impressive results in SIM-NMSE -41.27dB/-43.51dB

(DGRU/TCN PA), achieves superior linearization performance while using significantly fewer parameters than existing deep neural network solutions, proving the TCN-DPD model's parameters efficiency.

Finally, this thesis not only advances the author's understanding of TCNs in DPD applications but also opens new avenues for their potential integration into broader PA systems, promising substantial improvements in efficiency and performance in the telecommunications industry.

5.2. Recommendation

The thesis implemented the TCNs architecture into the DPD system, this novel approach provides a robust foundation for future research. Future studies could be continued to explore, and the author's suggestions are listed below:

1. **Real-World Measurements:**

Since all results in this thesis are simulations of the OpenDPD framework, one of the future studies should aim to have real-world measurements to validate the simulation results. While the TCN-based DPD models have shown promising results in the thesis's research, testing them under actual operating conditions can provide insights into their practical viability and robustness against nonlinearities and disturbances. Besides that, the proposed TCN-DPD model uses noncausal convolution, improving the simulation accuracy significantly. Previous research shows the consistency of simulation and measurement results. However, the measurements are real-time processing and the future elements of signals can not be provided, probably the TCN-DPD efficiency will be highly affected. Therefore, the causal and noncausal TCN-DPD models are both suggested to be measured in the real world.

2. **Incorporate Power Consumption Metrics:** Based on the thesis's results, the proposed TCN-DPD model is fairly parameters efficient. When achieving a similar performance with lower parameters, power consumption will also decrease in the whole system. However, the thesis did not include a specific analysis of the power consumption metrics. As energy efficiency is crucial for mobile and embedded systems, subsequent studies should include power analysis. This will allow for a comprehensive evaluation of the model's efficiency, not just its performance in linearizing PAs but also its impact on the overall power budget of the communication system.

3. **Enhanced Dataset Diversity:**

Since the thesis's benchmarks were all tested on the same dataset, the performance evaluation was still limited. To overcome the limitations imposed by using a single dataset, future research should incorporate multiple datasets that reflect a wider range of scenarios and conditions typically encountered in DPD applications. This will enhance the robustness and adaptability of the TCN-based DPD models. Based on the needs of communication systems at present, the dataset could include different types of signals and bandwidths. These various datasets can be used to test whether any specific conditions negatively impact their performance and how well the TCN DPD model adapts to real-world conditions.

4. **Refinement of TCN Architectures:**

Combining TCNs with other neural network architectures or traditional signal processing methods could potentially lead to new findings in performance and capabilities, such as phase normalization, etc.

5. **Optimization of TCN Architectures:**

Even though this thesis has already explored the TCN architecture design, there are probably still some components or tricks that have been ignored to optimize the proposed architecture to achieve better performance and ability.

References

[1] Lei Guan and Anding Zhu. "Green Communications: Digital Predistortion for Wideband RF Power Amplifiers". In: *IEEE Microwave Magazine* 15.7 (2014), pp. 84–99. DOI: 10.1109/MMM.2014.2356037.

[2] Allen Katz, John Wood, and Daniel Chokola. "The Evolution of PA Linearization: From Classic Feedforward and Feedback Through Analog and Digital Predistortion". In: *IEEE Microwave Magazine* 17.2 (2016), pp. 32–40. DOI: 10.1109/MMM.2015.2498079.

[3] A.A.M. Saleh. "Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers". In: *IEEE Transactions on Communications* 29.11 (1981), pp. 1715–1720. DOI: 10.1109/TCOM.1981.1094911.

[4] C. J. Harris. "Book Review: The Volterra and Wiener Theories of Nonlinear Systems". In: *International Journal of Electrical Engineering & Education* 18.2 (1981), pp. 187–187. DOI: 10.1177/002072098101800238. eprint: https://doi.org/10.1177/002072098101800238. URL: https://doi.org/10.1177/002072098101800238.

[5] Changsoo Eun and Edward J Powers. "A new Volterra predistorter based on the indirect learning architecture". In: *IEEE transactions on signal processing* 45.1 (1997), pp. 223–227.

[6] Lei Ding et al. "Memory polynomial predistorter based on the indirect learning architecture". In: *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*. Vol. 1. 2002, 967–971 vol.1. DOI: 10.1109/GLOCOM.2002.1188221.

[7] Lei Ding et al. "A robust digital baseband predistorter constructed using memory polynomials". In: *IEEE Transactions on Communications* 52.1 (2004), pp. 159–165. DOI: 10.1109/TCOMM.2003.822188.

[8] Dennis R Morgan et al. "A generalized memory polynomial model for digital predistortion of RF power amplifiers". In: *IEEE Transactions on signal processing* 54.10 (2006), pp. 3852–3860.

[9] Reina Hongyo et al. "Deep neural network-based digital predistorter for Doherty power amplifiers". In: *IEEE Microwave and Wireless Components Letters* 29.2 (2019), pp. 146–148.

[10] Xu Lu et al. "A Low-Computational-Complexity Digital Predistortion Model for Wideband Power Amplifier". In: *Sensors (Basel, Switzerland)* 24.21 (2024), p. 6941.

[11] Meenakshi Rawat, Karun Rawat, and Fadhel M. Ghannouchi. "Adaptive Digital Predistortion of Wireless Power Amplifiers/Transmitters Using Dynamic Real-Valued Focused Time-Delay Line Neural Networks". In: *IEEE Transactions on Microwave Theory and Techniques* 58.1 (2010), pp. 95–104. DOI: 10.1109/TMTT.2009.2036334.

[12] Arne Fischer-Bühner et al. "Phase-normalized neural network for linearization of RF power amplifiers". In: *IEEE Microwave and Wireless Technology Letters* (2023).

[13] S Hochreiter. "Long Short-term Memory". In: *Neural Computation MIT-Press* (1997).

[14] Hongmin Li et al. "Vector decomposed long short-term memory model for behavioral modeling and digital predistortion for wideband RF power amplifiers". In: *IEEE Access* 8 (2020), pp. 63780–63789.

[15] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[16] Xin Hu et al. "Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.8 (2021), pp. 3923–3937.

[17] Ashutosh Pandey and DeLiang Wang. "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 6875–6879.

[18] Yitian Chen et al. "Probabilistic forecasting with temporal convolutional neural network". In: *Neurocomputing* 399 (2020), pp. 491–501.

[19] Alexander H. Waibel et al. "Phoneme recognition using time-delay neural networks". In: *IEEE Trans. Acoust. Speech Signal Process.* 37 (1989), pp. 328–339. URL: `https://api.semanticscholar.org/CorpusID:9563026`.

[20] Yizhuo Wu et al. "OpenDPD: An open-source end-to-end learning & benchmarking framework for wideband power amplifier modeling and digital pre-distortion". In: *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2024, pp. 1–5.

[21] Tugce Kobal et al. "Digital predistortion of RF power amplifiers with phase-gated recurrent neural networks". In: *IEEE Transactions on Microwave Theory and Techniques* 70.6 (2022), pp. 3291–3299.

[22] Henna Paaso and Aarne Mammela. "Comparison of direct learning and indirect learning pre-distortion architectures". In: *2008 IEEE International Symposium on Wireless Communication Systems*. IEEE. 2008, pp. 309–313.

[23] Siqi Wang, Wenhui Cao, and Thomas Eriksson. "Identification methods with different digital pre-distortion models for power amplifiers with strong nonlinearity and memory effects". In: *2020 IEEE MTT-S International Wireless Symposium (IWS)*. IEEE. 2020, pp. 1–3.

[24] Yann LeCun et al. "Comparison of learning algorithms for handwritten digit recognition". In: *International conference on artificial neural networks*. Vol. 60. 1. Perth, Australia. 1995, pp. 53–60.

[25] K O'Shea. "An introduction to convolutional neural networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[26] Charlotte Pelletier, Geoffrey I Webb, and François Petitjean. "Temporal convolutional neural network for the classification of satellite image time series". In: *Remote Sensing* 11.5 (2019), p. 523.

[27] Colin Lea et al. "Temporal convolutional networks: A unified approach to action segmentation". In: *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer. 2016, pp. 47–54.

[28] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271* (2018).

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[30] Aaron Van Den Oord et al. "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499* 12 (2016).

[31] Fisher Yu and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions". In: *International Conference on Learning Representations*. 2016.

[32] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[33] François Chollet. "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.

[34] Corinna Cortes Yann LeCun and Christopher J.C Bureges. "THE MNIST DATABASE of handwritten digits". In: Retrieved from https://yann.lecun.com/exdb/mnist/. 2010.

[35] Mohammadreza Beikmirza et al. "A Wideband Energy-Efficient Multi-Mode CMOS Digital Transmitter". In: *IEEE Journal of Solid-State Circuits* 58.3 (2023), pp. 677–690. DOI: `10.1109/JSSC.2022.3222028`.

A

Source Code

```
"""
__author__ = "Huanqiang Duan, Chang Gao"
__email__ = "Huanqiang , chang.g" """

import torch
import torch.nn.functional as F
import torch.nn as nn

class TCN(nn.Module):

    def __init__(self, hidden_size, output_size, num_layers):
        super(TCN, self).__init__()
        self.in_channels = 6
        self.hidden_channels = hidden_size
        self.out_channels = output_size
        self.kernel_size = 5
        self.padding = (self.kernel_size-1) // 2
        self.dilation_1 = 1
        self.dilation_2 = 2
        self.dilation_3 = 4
        self.dilation_4 = 8
        self.stride = 1

        self.network = nn.Sequential(
            nn.Conv1d(in_channels=self.in_channels, out_channels=self.hidden_channels,
                kernel_size=1),
            nn.Hardswish(),

            nn.Conv1d(self.hidden_channels, self.hidden_channels, self.kernel_size,stride=
                self.stride,
                      padding=self.padding*self.dilation_1, dilation=self.dilation_1, groups=
                          self.hidden_channels, bias=False),
            nn.Hardswish(),
            nn.Conv1d(self.hidden_channels, self.hidden_channels, self.kernel_size,stride=
                self.stride,
                      padding=self.padding *self.dilation_2, dilation=self.dilation_2, groups
                          =self.hidden_channels, bias=False),
            nn.Hardswish(),
            nn.Conv1d(self.hidden_channels, self.hidden_channels, self.kernel_size,stride=
                self.stride,
                      padding=self.padding*self.dilation_3, dilation=self.dilation_3, groups=
                          self.hidden_channels, bias=False),
            nn.Hardswish(),

            nn.Conv1d(self.hidden_channels, self.hidden_channels, self.kernel_size, stride=
                self.stride,
                      padding=self.padding*self.dilation_4, dilation=self.dilation_4, groups=
                          self.hidden_channels, bias=False),
            nn.Hardswish(),
```

```
            #nn.Conv1d(self.hidden_channels2, self.hidden_channels2, self.kernel_size, stride=self.stride,
                      #padding=(self.kernel_size-3)*self.dilation*16, dilation=self.dilation*16, groups=self.hidden_channels2, bias=False),
           # nn.Hardswish(),
            nn.Conv1d(self.hidden_channels, self.out_channels, kernel_size=1, bias=False),
            )

    def forward(self, x, h_0):
        # Feature Extraction
        i_x = torch.unsqueeze(x[..., 0], dim=-1)
        q_x = torch.unsqueeze(x[..., 1], dim=-1)
        amp2 = torch.pow(i_x, 2) + torch.pow(q_x, 2)
        amp = torch.sqrt(amp2)
        amp3 = torch.pow(amp, 3)
        cos = i_x / amp
        sin = q_x / amp
        input = torch.cat((i_x, q_x), dim=-1)
        x = torch.cat((i_x, q_x, amp, amp3, sin, cos), dim=-1)
        x_1 = x.transpose(1,2)
        out = self.network(x_1)
        out = out.transpose(1,2)
        return out + input
```