Linearization of Quadrature Digital Power Amplifiers by Neural Network of ULR_LSTM: Unsupervised Learning Residual LSTM

Jiayu Yang, Luyi Guo, Yicheng Li, Wang Wang, Zixu Li, Manni Li, Zijian Huang
Yinyin Lin*[1], Yun Yin*[3], and Hongtao Xu*[2]
State Key Laboratory of Integrated Chips and Systems, School of Microelectronics, Fudan University, Shanghai, China
Email: *[1]yylin@fudan.edu.cn; *[2]hongtao@fudan.edu.cn; *[3]yiny@fudan.edu.cn

***Abstract*—For the first time, this paper presents an unsupervised learning residual long short-term memory (ULR_LSTM) neural network to develop a digital predistortion (DPD) method for the linearization of digital power amplifiers (DPAs). Our method eliminates the need for iterative learning control (ILC) to obtain the ideal input of the DPA required by state-of-the-arts (SOTAs), which leads to high computational complexity and extensive training time. We perform behavioral modeling of the DPA using the R_LSTM network. After determining the optimal behavioral model architecture, the corresponding DPD model is obtained through an inverse training process. A 15-bit transformer-based quadrature DPA chip incorporating Class-G and IQ-cell-sharing techniques was implemented in a 28nm CMOS process to validate our proposed method. Experimental results demonstrate outstanding linearization performance comparing to prior arts, achieving an error vector magnitude (EVM) of -40.4dB for the 802.11ax 40MHz 64QAM signal.**

***Index Terms*—digital power amplifier (DPA), digital predistortion (DPD), unsupervised learning, residual LSTM (R_LSTM), deep neural network (DNN)**

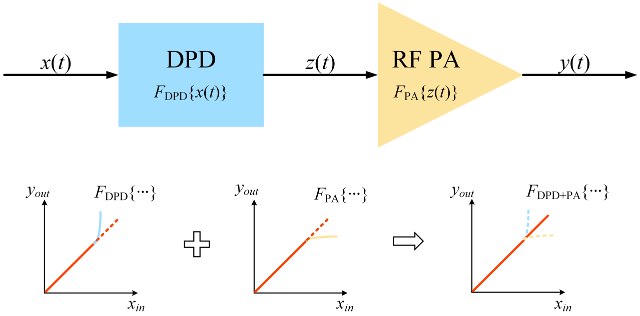

$x(t)$
DPD
$F_{\mathrm{DPD}}\{x(t)\}$
$z(t)$
RF PA
$F_{\mathrm{PA}}\{z(t)\}$
$y(t)$
y_{out}
$F_{\mathrm{DPD}}\{\cdots\}$
x_{in}
y_{out}
$F_{\mathrm{PA}}\{\cdots\}$
x_{in}
y_{out}
$F_{\mathrm{DPD+PA}}\{\cdots\}$
x_{in}

Fig. 1. Schematic of the digital predistortion.

I. Introduction

For modern spectrum-intensive communication standards like WLAN, LTE, and the upcoming sixth-generation (6G) communication systems [1], the increasing demand for higher data rates in wireless communications necessitates wider signal bandwidths and higher-order modulation schemes (e.g., 802.11ax). These advancements, coupled with the high peak-to-average power ratio (PAPR) inherent in such schemes, impose stricter requirements on the linearity and efficiency of wireless transmitters. Radio frequency (RF) power amplifiers (PAs), a critical component of wireless communication systems, suffer from nonlinearities that degrade signal quality. To address the trade-off between efficiency and linearity, digital predistortion (DPD) has become a widely used calibration technique in modern communication systems. The fundamental concept of DPD is to use a baseband equivalent inverse model of the RF PA to counteract the nonlinearity, as shown in Fig. 1.

Recently, the switched-capacitor power amplifier (SCPA) has garnered significant attention in wireless applications as a prevalent implementation of the digital power amplifier (DPA) [2], [3], [4]. This approach benefits greatly from CMOS scaling and demonstrates good compatibility with digital systems [5], [6], [7]. Compared to outphasing and polar architectures, quadrature DPA (QDPA) eliminates the need for complex design elements like CORDICs, wideband phase modulators, and delay calibration circuits. However, it introduces nonlinearity issues due to the I/Q combination, transistor time-variant resistance, supply network, etc., resulting in amplitude-to-amplitude (AM/AM) and amplitude-to-phase (AM/PM) distortions as well as the discernible memory effects. Such complexities make the nonlinear analysis and predistortion of the QDPA particularly challenging [4].

Up to now, limited prior arts of DPD for QDPAs are mainly based on two-dimensional (2-D) look-up table (LUT) methods. Static nonlinearities arising from rotation, LUT size, and interpolation have been explored in [8]. A 2D-LUT predistortion scheme was proposed in [9], but its performance was limited by interpolation and iterative gradient descent training. To accelerate training, [10] introduced coefficient inversion with 2D-LUT and least mean square filters, albeit at the cost of reduced accuracy. Memory effects in QDPA DPD were first addressed in [11], [12] using a two-stage approach combining 2D-LUT and generalized memory polynomials (GMP) [13]. All of the above-mentioned DPDs fundamentally require prior knowledge of the pre-distorted output signal (i.e., the ideal input of the DPA). Therefore, they first need to obtain the pre-distorted output signal through various inversion methods, such as iterative learning control (ILC) [14]. However, this step involves complex mathematical calculations, which typically require substantial computational resources and time.

978-3-9826741-0-0/DATE25/© 2025 EDAA

Deep learning in the AI field has shown excellent performance for the linearization of conventional analog PAs (APAs). Some notable architectures proposed include the real-valued time-delay neural network (RVTDNN) [15], the augmented RVTDNN (ARVTDNN) [16], the vector-decomposed time delay neural network (VDTDNN) [17], the block-oriented time-delay neural network (BOTDNN) [18], the real-valued time-delay convolutional neural network (RVTDCNN) [19], and the block-oriented Just Another Network (BO-JANET) [20]. Nevertheless, the majority of NN-based DPD solutions for APAs employ a supervised learning paradigm [21], which necessitates obtaining pre-distorted output signals through inversion methods like ILC to serve as training labels. These methods essentially do not address the aforementioned issue, as they still rely on ILC for inverse operations to obtain the ideal input for the PA. Despite the superior performance of NN models compared to traditional DPD algorithms like LUT and GMP, the underlying problem remains unresolved. The few NNs that employ unsupervised learning [22] generally exhibit poor performance due to the lack of carefully designed model architectures and inverse operation implementations.

To the best of our knowledge, there are no publications that utilize deep learning to address DPD for DPAs without requiring prior knowledge of the pre-distorted output signal.

For the first time, we propose an unsupervised learning residual long short-term memory (ULR_LSTM) neural network to develop a DPD method for linearizing DPAs. We directly utilize the original input and output signals of the DPA, rather than the label signals obtained through ILC, to train the DPD model. Specifically, we begin by conducting behavioral modeling of the DPA using the R_LSTM network. Once the optimal behavioral model architecture is identified, we derive the corresponding DPD model through an inverse training procedure. This method has been validated on our QDPA platform, demonstrating exceptional linearization performance compared to state-of-the-arts (SOTAs).

II. Prior Arts And Insights Motivating ULR_LSTM

A. Review of State-of-the-Art Methods

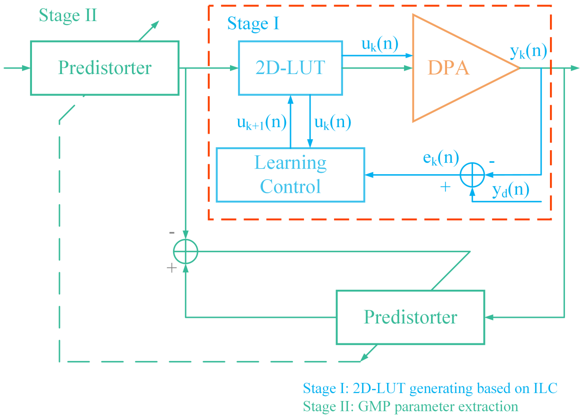


Fig. 2. The block diagram of the two-stage DPD in SOTAs.

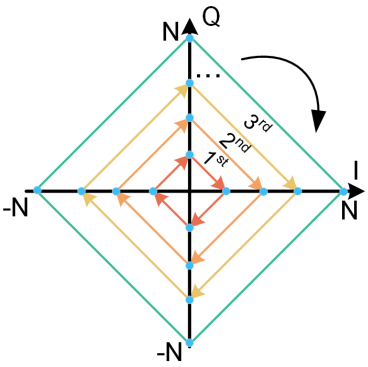


Fig. 3. Training sequence for 2D-LUT predistortion.

The SOTA works [11], [12] adopt a two-stage approach combining 2D-LUT and GMP to implement DPD. In the first stage, the 2D-LUT addresses static nonlinearity, while in the second stage, GMP addresses dynamic memory effects, as illustrated in Fig. 2.

Due to the decoding scheme internally in the IQ-cell-sharing QDPA [23], the complex domain only allows input codes within $|I| + |Q| \leq 2N$ to go through while others outside this boundary will be clipped, thus resulting in a diamond region for input signals. A diamond cycle is continuously swept from the smallest cycle to the largest and vice versa, as shown in Fig. 3. Each point in the sequence remains for 64 clock cycles to minimize fluctuations caused by memory effects.

To ensure the performance of the DPD, they adopt ILC algorithm [14], which is a well-established control theory technique that can obtain the inverse of a system, to identify the optimal input signal that drives the DPA to the desired linear output response. Although ILC algorithms offer high precision and robustness against noise, they impose stringent stability requirements on the system. In cases of system instability, convergence may be slow or even unattainable, necessitating multiple iterative steps, which leads to high computational complexity and prolonged training time.

B. Insights Motivating ULR_LSTM

The pain of the SOTA methods stems from the efforts to separate the static nonlinearity and dynamic memory effects, of which the two effects mix together in reality. The SOTA methods employ a two-stage approach to separate the static nonlinearity of the DPA from the dynamic memory effects. The first step involves using a special diamond training sequence (as shown in Fig. 3) to construct a 2D-LUT, rather than using actual modulated signals. To minimize fluctuations caused by memory effects, each point in the sequence remains for 64 clock cycles. This step attempts to isolate memory effects, addressing the static nonlinearity. In the second step, the actual modulated signal is passed through the 2D-LUT constructed in the first step to filter out the static nonlinearity, leaving only the memory effects. Finally, the GMP is used to model the memory effects, resulting in the complete DPD model.

The aforementioned separation of static nonlinearity and memory effects is costly, especially in the first step of constructing the 2D-LUT, which requires ILC. This approach is computationally expensive, and the entire model implementation process is quite cumbersome.

The two-stage approach is necessitated by the fact that traditional DPD algorithms are incapable of simultaneously mitigating the severe device nonlinearity inherent in QDPAs. However, deep neural networks (DNNs) possess powerful nonlinear fitting capabilities [24]. This motivates us to use DNNs to model DPA nonlinearities without artificially and mechanically separating the static nonlinearity and dynamic memory effects. Before constructing the DPD model, it is crucial to establish an accurate nonlinear behavioral model. Regarding the choice of behavioral model architecture, since both memory effects and static nonlinearity need to be handled simultaneously, we adopt the LSTM network as our baseline due to its ability to effectively capture long-term dependencies within input sequences through its internal gating mechanisms [25], making it well-suited for modeling nonlinear systems with memory effects. Thus, we first use an NN to perform behavioral modeling of the DPA. The behavioral model can be expressed as:

$$\mathbf{y} = f_{NN}(\mathbf{x}, \mathbf{w}) \tag{1}$$

where $\mathbf{x}$ represents the inputs of the behavioral model; $\mathbf{y}$ represents the corresponding outputs; f_{NN} represents NN functions for the behavioral model; $\mathbf{w}$ represents the NN weights.

As for the realization of unsupervised learning, we adopt a data-driven approach to constructing the DPD model. By reversing the input-output relationship in the training dataset, we can directly learn the inverse mapping, thereby avoiding the high cost of ILC. Thus, the DPD model can be expressed as:

$$\bar{\mathbf{y}} = f_{NN}^{-1}(\bar{\mathbf{x}}, \bar{\mathbf{w}}) \tag{2}$$

where $\bar{\mathbf{x}}$ represents the inputs of the DPD model; $\bar{\mathbf{y}}$ represents the corresponding outputs; f_{NN}^{-1} represents NN functions for the DPD model; $\bar{\mathbf{w}}$ represents the NN weights.

Our method does not require pre-distorted output signals as labels, making it an unsupervised learning approach. The next section will present the implementation of our ULR_LSTM-based DPD algorithm for DPAs.

III. Proposed ULR_LSTM-Based DPD

A. Overview of Our Method

Fig. 4 illustrates an overview of our method, which consists of two steps.

Step ①: By capturing raw input and output signals through practical DPA measurements, we construct a dataset to train a NN for behavioral modeling of DPAs. The optimal architecture for the behavioral model is not an arbitrary choice; on the contrary, it embodies the fruits of meticulous design and empirical validation.

Step ②: After identifying the optimal architecture for the behavioral model, we implement the reverse modeling technique described in Section II-B, interchanging the input and output of the behavioral model. This indicates that our DPD model is not an analytical inverse of the obtained behavioral model, but rather a shared structure requiring an inverse training process.

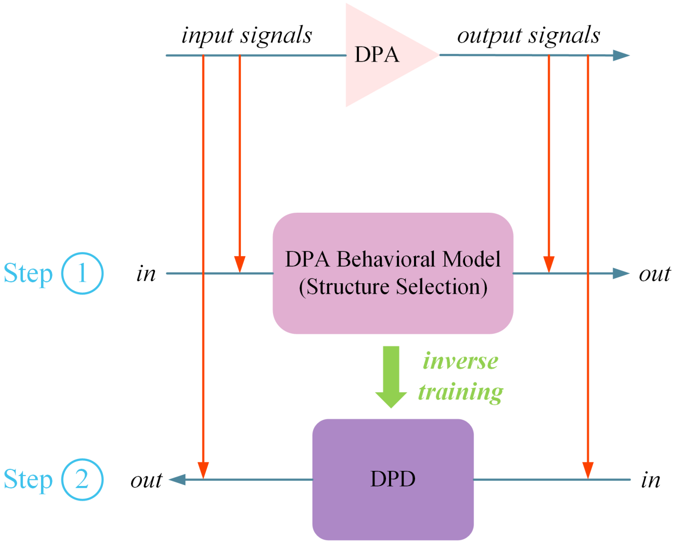


Fig. 4. Overview of our method.

Through Step ①, the behavioral modeling accurately captures the nonlinear characteristics of the DPA, effectively constructing an NN-based simulation framework for the DPA. Subsequently, in Step ②, by swapping the input and output signals, we perform inverse training on the established NN-based simulation framework in Step ① to obtain the equivalent inverse model of the DPA, which is the corresponding DPD model. Throughout the entire process, we only require the original input and output signals of the DPA, which are easily obtainable at virtually no cost. There is no need to obtain pre-distorted output signals through the expensive ILC method as training labels, making this an unsupervised learning approach.

B. Implementation of ULR_LSTM-Based DPD

The first step is to perform behavioral modeling of the DPA. The behavior of the DPA can be represented as

$$\mathbf{y}_t = \mathcal{F}(\mathbf{x}_t, \mathbf{w}_{PA}) \tag{3}$$

where $\mathbf{x}_t$, $\mathbf{y}_t$ are the baseband input and output signals of the DPA, respectively. $\mathcal{F}$ represents the behavioral model and $\mathbf{w}_{PA}$ is the corresponding parameters.

The training and testing datasets are constructed using a sequence-to-one approach to account for memory effects simultaneously. This approach differs from the one-to-one mode by incorporating a sequence of past input samples to capture the temporal dynamics and memory behavior of the DPA. By leveraging a combination of experimental insights and test results, we established a memory depth range of [1, 20] and identified that a memory depth of 14 most efficiently accounts for memory effects. The datasets are formed using digital baseband samples obtained from practical DPA measurements. Additionally, to prevent the exploding gradient problem during training, we normalize the input and output I/Q elements.

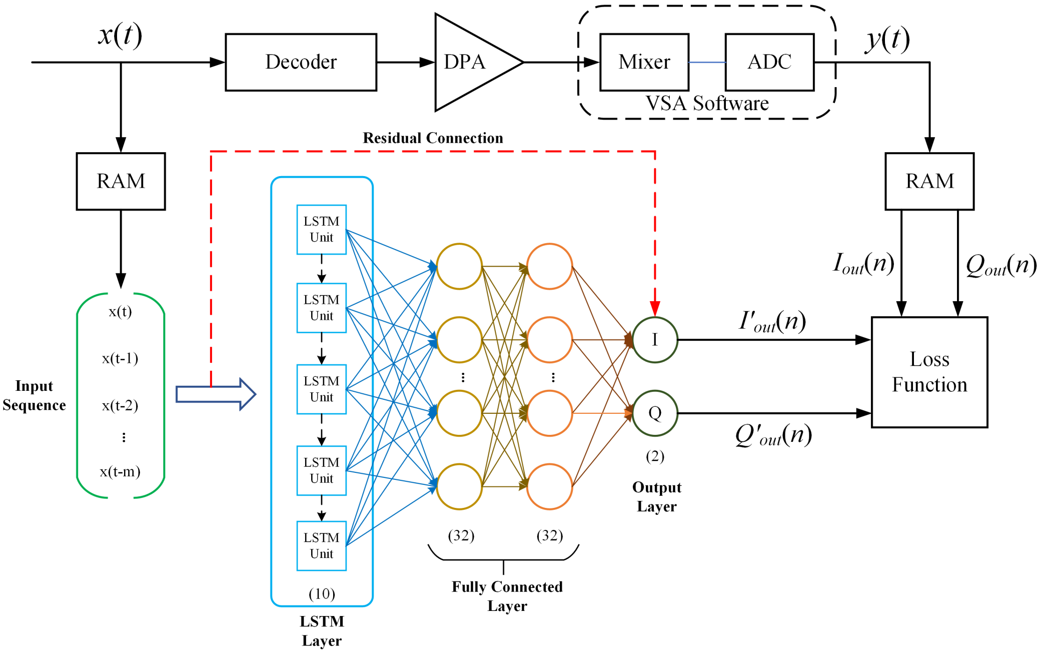


Fig. 5. Block diagram of the proposed ULR_LSTM model.

The proposed ULR_LSTM architecture, depicted in Fig. 5, comprises an input layer, an LSTM layer, two fully connected (FC) layers, and an output layer. A residual connection, indicated by the red dashed line in Fig. 5, directly links the input layer to the output layer, enabling faster convergence with improved accuracy [26].

The effectiveness of the model is assessed using Normalized Mean Squared Error (NMSE) as the evaluation metric throughout the entire construction process.

$$\begin{aligned} &NMSE \\ &= 10\times\lg\frac{\frac{1}{N}\sum_{i=1}^{N}[(I(y(i))-I(\hat{y}(i)))^2+(Q(y(i))-Q(\hat{y}(i)))^2]}{\frac{1}{N}\sum_{i=1}^{N}(I(y(i))^2+Q(y(i))^2)} \end{aligned} \tag{4}$$

where $I(y(i))$ and $Q(y(i))$ represent the measured output, respectively; $I(\hat{y}(i))$ and $Q(\hat{y}(i))$ represent the output of the ULR_LSTM model, respectively; and N is the length of the training data. The lower the NMSE, the better the performance.

Fig. 6 illustrates the impact of varying unit counts across different network layers on model performance. Each subplot displays NMSE on the left vertical axis, parameter count on the right vertical axis, and units per layer on the horizontal axis. The target NMSE is set at -36dB. We began by establishing a baseline network architecture consisting of a single LSTM layer, which was designed to address memory effects as well as partial nonlinearity (the LSTM layer internally contains nonlinear functions). We observed a positive correlation between the number of LSTM units and the network's regression accuracy. However, this improvement plateaued when the number of LSTM units reached 10. Next, we introduced an FC1 layer with a nonlinear activation function to address the remaining nonlinearities. After observing diminishing returns from increasing the number of neurons in the FC1 layer, we opted to add an FC2 layer. This additional layer facilitated further improvement in regression accuracy. To balance performance and computational cost, we settled on a configuration with 32 neurons in both the first and second FC layers. Regarding the choice of activation function, we opted for the ReLU function due to its advantages in simplicity, efficiency, sparsity, and effective gradient propagation [27]. The final output layer consists of two neurons, corresponding to the I and Q elements.

$$ReLU(x) = max(0, x) \tag{5}$$

We implemented the model using the PyTorch framework and trained it with the Adam optimizer [28]. A dynamic learning rate strategy was employed, where the initial learning rate of 0.001 decayed by a factor of 10 if the training loss stagnated for a predefined number of epochs. Training utilized a batch size of 32 and the NMSE loss function to evaluate model performance. To mitigate overfitting, early stopping was implemented, terminating training if the validation loss did not improve for a set number of epochs. Algorithm 1 outlines the training procedure, where model parameters are initialized with a uniform distribution.

Hence, the proposed behavioral model can be expressed as the following

$$\begin{aligned} \mathbf{y}_t &= \mathcal{F}(\mathbf{x}_t, \mathbf{w}_{PA}) \\ &= O(ReLU[FC_2(ReLU[FC_1(LSTM(I(\mathbf{x}_t)))])])+O(I(\mathbf{x}_t)) \end{aligned} \tag{6}$$

where I represents the input layer, $LSTM$ represents the LSTM layer, FC_1, FC_2 represent the two FC layers, $ReLU$

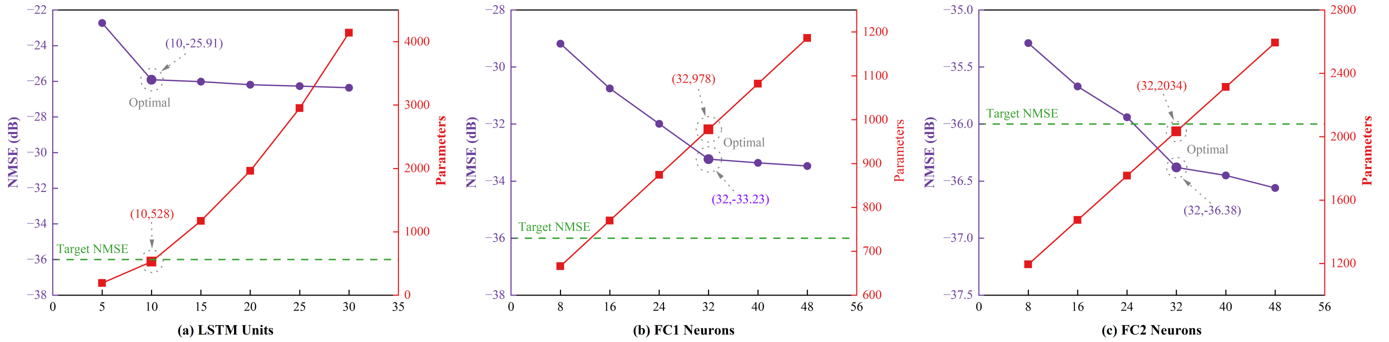


Fig. 6. Performance variation of the proposed model with different unit counts in network layers. (a) A single LSTM layer's performance is evaluated against the number of LSTM units, with an optimal count of 10 identified. (b) Building upon the optimal LSTM configuration, an FC1 layer is added, and its unit count is varied. The optimal number of FC1 neurons is determined to be 32. (c) With both LSTM and FC1 layers optimized, an FC2 layer is introduced, demonstrating an optimal unit count of 32, which also meets the target performance metric.

Algorithm 1 Training procedure of the behavioral model.

```
Input: $\mathbf{x}_t, \mathbf{y}_t$
Output: Trained model
1: Data acquisition and preprocessing;
2: Constructing the training and testing dataset;
3: Set hyperparameters and start training with a dynamically adjusted learning rate strategy;
4: $\mathcal{L} \leftarrow NMSE(\mathbf{y}_t, \hat{\mathbf{y}}_t)$;
5: Optimize $\mathcal{L}$ with Adam;
6: Early termination if fixed-epoch validation set loss is not reduced;
7: return Trained model
```

represents the nonlinear activation function and O represents the output layer.

The second step is inverse modeling, which means swapping the input and output signals of the original behavioral model. Our deep learning-based approach leverages the selected architecture depicted in Fig. 5 to simultaneously model and linearize the DPA. This indicates that our proposed DPD model shares the same structure as the behavioral model, albeit with distinct model parameters.

Therefore, there is no need to select a new network architecture. The proposed ULR_LSTM-based DPD model can be expressed as follows:

$$\begin{aligned}\mathbf{z}_t &= \mathcal{G}(\mathbf{x}_t, \mathbf{w}_{PD}) \\ &= O(ReLU[FC_2(ReLU[FC_1(LSTM(I(\mathbf{x}_t)))])]) + O(I(\mathbf{x}_t))\end{aligned} \tag{7}$$

where $\mathbf{z}_t$ represents the pre-distorted signals. $\mathcal{G}$ represents the DPD model and $\mathbf{w}_{PD}$ is the parameters of the DPD model.

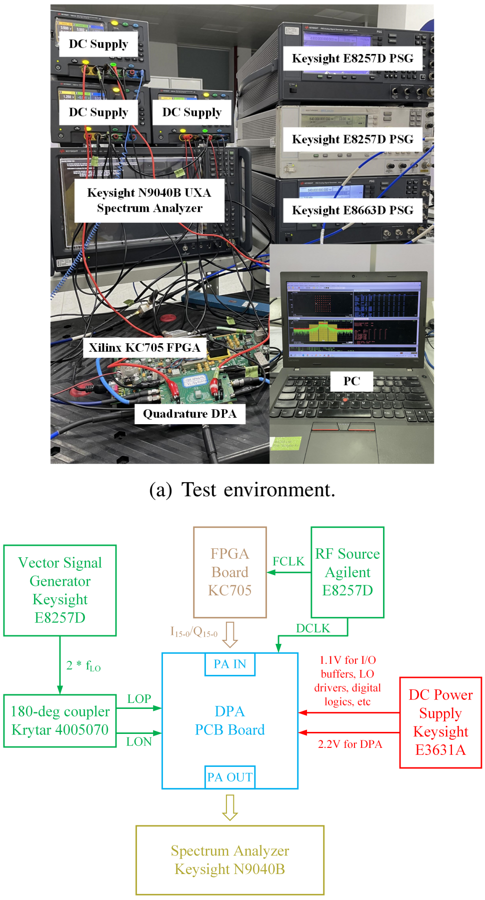
Therefore, the cascade system of DPD and DPA can be expressed as

$$\mathbf{y}_t = \mathcal{F}(\mathcal{G}(\mathbf{x}_t, \mathbf{w}_{PD}), \mathbf{w}_{PA}) \tag{8}$$

With the emergence of the DPD model, output signals $\mathbf{y}_t$ would be linear with respect to input signals $\mathbf{x}_t$.

IV. Results Based On Measurements

A. Measurement Setup



DC Supply
Keysight E8257D PSG
DC Supply
DC Supply
Keysight E8257D PSG
Keysight N9040B UXA Spectrum Analyzer
Keysight E8663D PSG
Xilinx KC705 FPGA
PC
Quadrature DPA
Vector Signal Generator Keysight E8257D
FPGA Board KC705
FCLK
RF Source Agilent E8257D
$I_{15\text{-}0}/Q_{15\text{-}0}$
DCLK
$2 * f_{LO}$
PA IN
1.1V for I/O buffers, LO drivers, digital logics, etc
DC Power Supply Keysight E3631A
180-deg coupler Krytar 4005070
LOP
LON
DPA PCB Board
2.2V for DPA
PA OUT
Spectrum Analyzer Keysight N9040B

(a) Test environment.

(b) Block diagram of connectivity.

Fig. 7. Measurement setup.

A 15-bit transformer-based QDPA chip [23], integrating Class-G and IQ-cell-sharing techniques, was fabricated in a 28nm CMOS process and operates at 2.4 GHz. Fig. 7 shows the measurement setup, which includes RF signal generators for clock and LO signal generation, 1.1V and 2.2V DC power supplies, an FPGA for parallel transmission of input bitstreams to the chip, a spectrum analyzer with vector signal analysis (VSA) for demodulating the RF output signal, and a PC for offline DPD training and processing.

The used signals were 802.11ax 40MHz bandwidth orthogonal frequency division multiplexing (OFDM) with about 9.7dB PAPR using crest factor reduction. The sampling rates are 320Msps.

B. Predistortion Results

TABLE I
MEASURED RESULTS IN 802.11AX 40MHZ 64QAM.

Signal Type	802.11ax 40MHz 64QAM		
	NMSE (dB)	EVM (dB)	Pavg (dBm)
w/o DPD	-19.4	-29.1	17.1
w/ LUT [12]	-26.5	-33.5	16.6
w/ LUT+GMP [12]	-32.8	-36.2	16.6
w/ ULR_LSTM	**-37.6**	**-40.4**	**16.6**

Table I presents the measured NMSE, error vector magnitude (EVM), and average power (Pavg) for the 802.11ax 40MHz 64QAM signal using different DPD methods. Experimental results show that compared to the SOTA DPD methods [12], our proposed ULR_LSTM model achieves better performance across all metrics. At maximum output power, the EVM is reduced from -29.1dB to -40.4dB, an improvement of over 11dB.

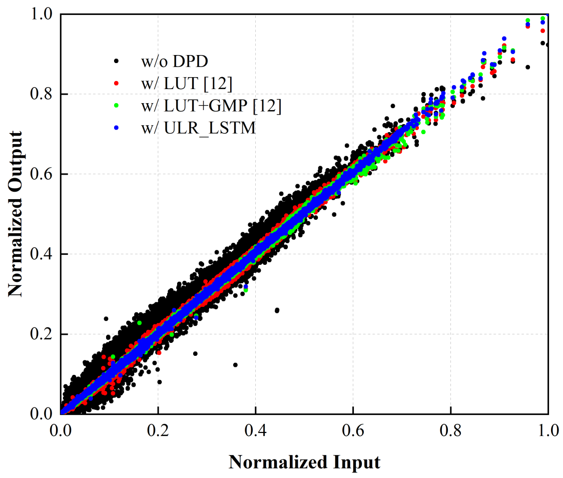


Fig. 8. Measured AM/AM characteristics of 802.11ax 40MHz 64QAM signal using different DPD methods.

The AM/AM and AM/PM characteristic curves without DPD, with SOTA DPD, and with our ULR_LSTM-based DPD are shown in Fig. 8 and Fig. 9, respectively. It can be noted that our model significantly mitigates nonlinearity and memory effects.

Fig. 10 shows the measured power spectrum density (PSD) using different DPD methods. It is evident that our ULR_LSTM model achieves the best out-of-band spectrum suppression ability and effectively improves adjacent channel power ratio (ACPR) performance.

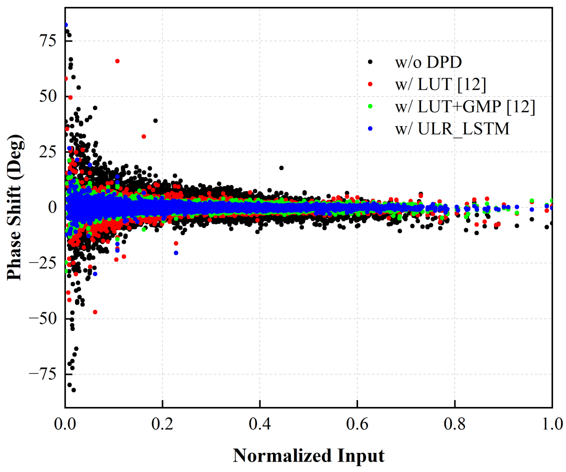


Fig. 9. Measured AM/PM characteristics of 802.11ax 40MHz 64QAM signal using different DPD methods.

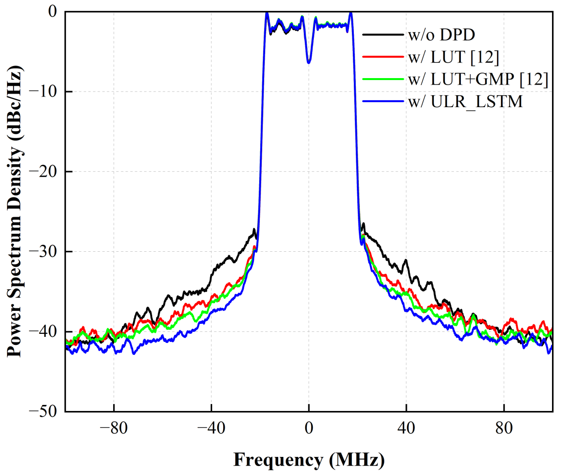


Fig. 10. Measured power spectrum density of 802.11ax 40MHz 64QAM signal using different DPD methods.

V. CONCLUSION

This paper presents a ULR_LSTM neural network to implement DPD for DPAs, eliminating the need for ILC in SOTA methods. Our proposed deep learning-based DPD method shares the same structure as the behavioral model but with distinct model parameters. Compared to SOTA DPD methods, experimental results demonstrate that our proposed ULR_LSTM model achieves excellent linearization performance, with an EVM of -40.4dB for the 802.11ax 40MHz 64QAM signal.

References

[1] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, “The roadmap to 6g: Ai empowered wireless networks,” *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[2] S.-M. Yoo, J. S. Walling, E. C. Woo, B. Jann, and D. J. Allstot, “A switched-capacitor rf power amplifier,” *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2977–2987, 2011.

[3] S.-M. Yoo, J. S. Walling, O. Degani, B. Jann, R. Sadhwani, J. C. Rudell, and D. J. Allstot, “A class-g switched-capacitor rf power amplifier,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 5, pp. 1212–1224, 2013.

[4] W. Yuan, V. Aparin, J. Dunworth, L. Seward, and J. S. Walling, “A quadrature switched capacitor power amplifier,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 5, pp. 1200–1209, 2016.

[5] D. Zheng, Y. Yin, Y. Zhu, L. Xiong, Y. Li, N. Yan, and H. Xu, “24.5 a 15b quadrature digital power amplifier with transformer-based complex-domain power-efficiency enhancement,” in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 370–372.

[6] Y. Li, Y. Yin, D. Zheng, X. Jia, J. Lin, F. Gao, Y. Zhu, L. Xiong, N. Yan, Y. Lu *et al.*, “A 15-bit quadrature digital power amplifier with transformer-based complex-domain efficiency enhancement,” *IEEE Journal of Solid-State Circuits*, vol. 57, no. 6, pp. 1610–1622, 2021.

[7] H. J. Qian, B. Yang, J. Zhou, H. Xu, and X. Luo, “A quadrature digital power amplifier with hybrid doherty and impedance boosting for complex domain power back-off efficiency enhancement,” *IEEE Journal of Solid-State Circuits*, vol. 56, no. 5, pp. 1487–1501, 2021.

[8] R. Bhat and H. Krishnaswamy, “Design tradeoffs and predistortion of digital cartesian rf-power-dac transmitters,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 11, pp. 1039–1043, 2016.

[9] C. Lu, H. Wang, C. Peng, A. Goel, S. Son, P. Liang, A. Niknejad, H. Hwang, and G. Chien, “A 24.7 dbm all-digital rf transmitter for multi-mode broadband applications in 40nm cmos,” in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*. IEEE, 2013, pp. 332–333.

[10] Y. Hu, T. Wang, and Z. Hong, “The digital front end with dual-box digital pre-distortion in all-digital quadrature transmitter,” in *2019 IEEE 13th International Conference on ASIC (ASICON)*. IEEE, 2019, pp. 1–4.

[11] F. Gao, Y. Yin, Y. Li, J. Lin, and H. Xu, “A two-stage digital predistortion method for quadrature digital power amplifiers,” in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2022, pp. 2491–2495.

[12] L. Guo, Y. Yin, F. Gao, Y. Li, J. Lin, and H. Xu, “Automatic dual-box digital predistortion calibration with residual elimination for quadrature digital transmitters,” in *2023 IEEE International Symposium on Radio-Frequency Integration Technology (RFIT)*. IEEE, 2023, pp. 44–46.

[13] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, “A generalized memory polynomial model for digital predistortion of rf power amplifiers,” *IEEE Transactions on signal processing*, vol. 54, no. 10, pp. 3852–3860, 2006.

[14] J. Chani-Cahuana, P. N. Landin, C. Fager, and T. Eriksson, “Iterative learning control for rf power amplifier linearization,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 9, pp. 2778–2789, 2016.

[15] T. Liu, S. Boumaiza, and F. M. Ghannouchi, “Dynamic behavioral modeling of 3g power amplifiers using real-valued time-delay neural networks,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 3, pp. 1025–1033, 2004.

[16] D. Wang, M. Aziz, M. Helaoui, and F. M. Ghannouchi, “Augmented real-valued time-delay neural network for compensation of distortions and impairments in wireless transmitters,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, pp. 242–254, 2018.

[17] Y. Zhang, Y. Li, F. Liu, and A. Zhu, “Vector decomposition based time-delay neural network behavioral model for digital predistortion of rf power amplifiers,” *IEEE Access*, vol. 7, pp. 91 559–91 568, 2019.

[18] C. Jiang, H. Li, W. Qiao, G. Yang, Q. Liu, G. Wang, and F. Liu, “Block-oriented time-delay neural network behavioral model for digital predistortion of rf power amplifiers,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 3, pp. 1461–1473, 2021.

[19] X. Hu, Z. Liu, X. Yu, Y. Zhao, W. Chen, B. Hu, X. Du, X. Li, M. Helaoui, W. Wang *et al.*, “Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3923–3937, 2021.

[20] Q. Zhang, C. Jiang, G. Yang, R. Han, and F. Liu, “Block-oriented recurrent neural network for digital predistortion of rf power amplifiers,” *IEEE Transactions on Microwave Theory and Techniques*, 2023.

[21] P. Cunningham, M. Cord, and S. J. Delany, “Supervised learning,” in *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer, 2008, pp. 21–49.

[22] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani, and J. Friedman, “Unsupervised learning,” *The elements of statistical learning: Data mining, inference, and prediction*, pp. 485–585, 2009.

[23] Y. Li, Y. Yin, D. Zheng, F. Gao, J. Lin, Z. Hu, Y. Lu, and H. Xu, “A quadrature digital power amplifier with wide efficiency enhancement coverage and high dynamic power range,” *IEEE Journal of Solid-State Circuits*, 2024.

[24] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, “Explaining deep neural networks and beyond: A review of methods and applications,” *Proceedings of the IEEE*, vol. 109, no. 3, pp. 247–278, 2021.

[25] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: Lstm cells and network architectures,” *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[27] J. He, L. Li, J. Xu, and C. Zheng, “Relu deep neural networks and linear finite elements,” *arXiv preprint arXiv:1807.03973*, 2018.

[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.