DeltaDPD

Exploiting Dynamic Temporal Sparsity in Recurrent Neural Networks for Energy-Efficient Wideband Digital Predistortion

Wu, Yizhuo; Zhu, Yi; Qian, Kun; Chen, Qinyu; Zhu, Anding; Gajadharsing, John; de Vreede, Leo C.N.; Gao, Chang

DOI
10.1109/LMWT.2025.3565004

Publication date
2025

Document Version
Final published version

Published in
IEEE Microwave and Wireless Technology Letters

DeltaDPD: Exploiting Dynamic Temporal Sparsity in Recurrent Neural Networks for Energy-Efficient Wideband Digital Predistortion

Yizhuo Wu, *Graduate Student Member, IEEE*, Yi Zhu, Kun Qian, Qinyu Chen, Anding Zhu, *Fellow, IEEE*, John Gajadharsing, *Senior Member, IEEE*, Leo C. N. de Vreede, *Senior Member, IEEE*, and Chang Gao, *Member, IEEE*

***Abstract*—Digital predistortion (DPD) is a popular technique to enhance signal quality in wideband radio frequency (RF) power amplifiers (PAs). With increasing bandwidth and data rates, DPD faces significant energy consumption challenges during deployment, contrasting with its efficiency goals. State-of-the-art DPD models rely on recurrent neural networks (RNNs), whose computational complexity hinders system efficiency. This letter introduces DeltaDPD, exploring the dynamic temporal sparsity of input signals and neuronal hidden states in RNNs for energy-efficient DPD, reducing arithmetic operations and memory accesses while preserving satisfactory linearization performance. Applying a TM3.1a 200 MHz-BW 256-QAM OFDM signal to a 3.5-GHz GaN Doherty RF PA, DeltaDPD achieves −50.03 dBc in adjacent channel power ratio (ACPR), −37.22 dB in normalized mean square error (NMSE) and −38.52 dB in error vector magnitude (EVM) with 52% temporal sparsity, leading to a 1.8× reduction in estimated inference power. The DeltaDPD code is available at https://www.opendpd.com.**

***Index Terms*—Digital predistortion (DPD), digital signal processing (DSP), power amplifier (PA), recurrent neural network (RNN), temporal sparsity.**

I. Introduction

DIGITAL predistortion (DPD) is a popular method to linearize wideband radio frequency (RF) power amplifiers (PAs). Nevertheless, in modern radio digital backends, DPD consumes a substantial portion of power [1]. This issue could be further intensified by the potential incorporation of machine learning (ML) techniques, such as recurrent neural networks (RNNs), which, despite their promising capabilities, increase power requirements.

Recent progress in ML-based long-term RNN-based DPD for wideband PAs is detailed in [2], [3], [4], and [5]. However, the considerable computational complexity and memory needs of RNN-based DPD systems present major challenges to their efficient implementation in digital signal processing (DSP) processors for wideband transmitters. This is especially relevant for upcoming 5.5G/6G base stations or Wi-Fi 7 routers, where limited power resources restrict real-time DPD model computation.

Previous methods to tackle DPD energy consumption include reducing the sample rate [6], utilizing a sub-Nyquist feedback receiver in the observation path [7], dynamically modifying model cross-terms based on input signal properties [8], creating simplified computational pathways for DPD algorithms [9], pruning the unimportant weight of fully connected layer (FC) to induce static spatial weight sparsity [10] and reducing the precision of DPD models [11].

This letter proposes a novel power-saving approach for wideband DPD by inducing and exploiting dynamic temporal sparsity [12] in RNN inputs and hidden states using the delta network algorithm [13]. The proposed algorithm decreases both memory access and arithmetic operations by deactivating part of multiplication-and-accumulation (MAC) operations. It facilitates the design of power-area-efficient RNN computing hardware suitable for DPD deployment in resource-constrained environments. The proposed method can potentially be applied to various RNN-based DPDs.

II. DeltaDPD Algorithm

In this work, we use JANET [14] and GRU [15] cells, as shown in Fig. 1(a), which were adopted in recent DPD studies [3], [4], [11], to verify the effectiveness of the DeltaDPD in reducing power without significant linearization loss and its adaptability to different RNN architectures. Both the JANET and GRU cells are cascaded with an FC layer with two output neurons as the output layer.

A. Delta Updating Rule

Neural networks (NNs) use dense-matrix-dense-vector multiplication ($\mathbf{M} \times \mathbf{V}$) as illustrated in Fig. 1(b). When processing continuous sequential signals using NNs, input data samples ϕ and hidden states h of the network could have high autocorrelation, causing small changes (Δ) between neighboring time steps at durations when the derivative of data is low, leading to temporal sparsity in delta input $\Delta\phi$

Received 27 February 2025; revised 23 April 2025; accepted 24 April 2025. Date of publication 16 May 2025; date of current version 9 June 2025. This work was supported in part by the European Research Executive Agency (REA) under the Marie Skłodowska-Curie Actions (MSCA) Postdoctoral Fellowship Program, under Grant 101107534 (AIRHAR). *(Corresponding author: Chang Gao.)*

Yizhuo Wu, Kun Qian, Leo C. N. de Vreede, and Chang Gao are with the Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: chang.gao@tudelft.nl).

Yi Zhu and John Gajadharsing are with Ampleon Netherlands B.V., 6534 AV Nijmegen, The Netherlands.

Qinyu Chen is with Leiden Institute of Advanced Computer Science (LIACS), Leiden University, 2300 RA Leiden, The Netherlands.

Anding Zhu is with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, D04 V1W8 Ireland.

This article was presented at the IEEE MTT-S International Microwave Symposium (IMS 2025), San Francisco, CA, USA, June 15–20, 2025.

Digital Object Identifier 10.1109/LMWT.2025.3565004

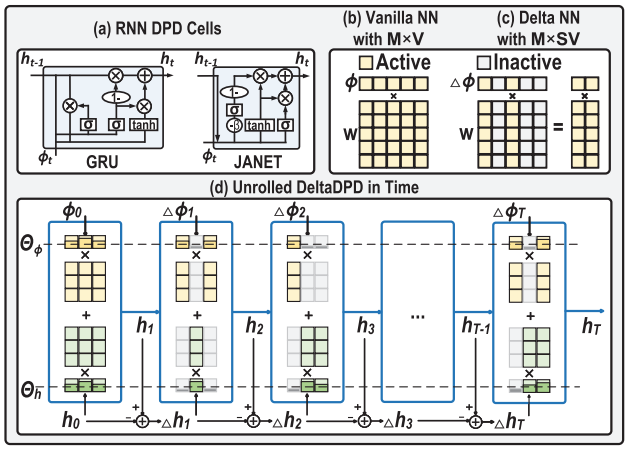

Fig. 1. (a) GRU and JANET cell structure with inputs ϕ and hidden states h. (b) Vanilla network and (c) delta network M × V. (d) Unrolled DeltaDPD network M × V in time.

and delta hidden state vectors Δh, which can be used to convert M × V into dense-matrix-sparse-vector multiplication (**M** × **SV**). As depicted in Fig. 1(d), by defining thresholds Θ_ϕ and Θ_h, DeltaDPD skips MAC operations and memory access involving below-threshold Δ vector elements and their corresponding weight columns, where all gray elements are skipped.

A sequential delta M × V can be derived by

$$\mathbf{y}_t = \mathbf{W}\mathbf{x}_t \tag{1}$$

$$\mathbf{y}_t = \mathbf{W}\Delta\mathbf{x}_t + \mathbf{y}_{t-1} = \mathbf{W}(\mathbf{x}_t - \mathbf{x}_{t-1}) + \mathbf{y}_{t-1} \tag{2}$$

where x can be either the RNN input $\boldsymbol{\phi}_t$ or hidden state $\mathbf{h}_t$ vector at time t, $\mathbf{W}$ represents the weight matrices, and $\mathbf{y}_{t-1}$ is M × V result from the previous time step $t-1$. In (1), $\mathbf{W}\Delta\mathbf{x}_t$ becomes **M** × **SV** if only computations corresponding to above-threshold $\Delta\mathbf{x}_t$ elements are kept, as given by

$$\Delta\mathbf{x}_t = \begin{cases} \mathbf{x}_t - \tilde{\mathbf{x}}_{t-1}, & |\mathbf{x}_t - \tilde{\mathbf{x}}_{t-1}| > \Theta_x \\ 0, & |\mathbf{x}_t - \tilde{\mathbf{x}}_{t-1}| \le \Theta_x \end{cases} \tag{3}$$

where a piece of memory $\tilde{\mathbf{x}}$ is used to buffer the previous state. To prevent error accumulation over time by memorizing only the last significant change, each kth scalar element $\tilde{x}^k$ of vector $\tilde{\mathbf{x}}$ only gets updated when the corresponding Δx^k exceeds the threshold. This updating rule is defined by

$$\tilde{x}^k_{t-1} = \begin{cases} x^k_{t-1}, & \left|x^k_t - \tilde{x}^k_{t-1}\right| > \Theta_x \\ \tilde{x}^k_{t-2}, & \left|x^k_t - \tilde{x}^k_{t-1}\right| \le \Theta_x. \end{cases} \tag{4}$$

B. Definition of DeltaDPD

Taking the classic GRU-RNN as an example, the preactivation accumulation of DeltaGRU with input feature $\boldsymbol{\phi}_t = \left[I_t,\ Q_t,\ |x_t|,\ |x_t|^3, \sin\theta_t, \cos\theta_t\right]$ can be derived by transforming the original GRU equations into their delta forms by following (1) and (2):

$$\mathbf{M}_{r,t} = \mathbf{W}_{ir}\Delta\boldsymbol{\phi}_t + \mathbf{W}_{hr}\Delta\mathbf{h}_{t-1} + \mathbf{M}_{r,t-1} \tag{5}$$

$$\mathbf{M}_{z,t} = \mathbf{W}_{iz}\Delta\boldsymbol{\phi}_t + \mathbf{W}_{hz}\Delta\mathbf{h}_{t-1} + \mathbf{M}_{z,t-1} \tag{6}$$

$$\mathbf{M}_{n\phi,t} = \mathbf{W}_{in}\Delta\boldsymbol{\phi}_t + \mathbf{M}_{n\phi,t-1} \tag{7}$$

$$\mathbf{M}_{nh,t} = \mathbf{W}_{hn}\Delta\mathbf{h}_{t-1} + \mathbf{M}_{nh,t-1}. \tag{8}$$

The terms M_r, M_z, and M_n are the preactivation accumulation of DeltaGRU's reset gate r, update gate z, and new gate n, initialized by the biases of gates $\mathbf{M}_{r,0} = \mathbf{b}_{ir}$, $\mathbf{M}_{z,0} = \mathbf{b}_{iz}$, $\mathbf{M}_{n\phi,0} = \mathbf{b}_{in}$, and $\mathbf{M}_{nh,0} = \mathbf{b}_{hn}$. Therefore, the DeltaGRU-based DPD is defined as

$$\mathbf{r}_t = \sigma\left(\mathbf{M}_{r,t}\right) \tag{9}$$

$$\mathbf{z}_t = \sigma\left(\mathbf{M}_{z,t}\right) \tag{10}$$

$$\mathbf{n}_t = \tanh\left(\mathbf{M}_{n\phi,t} + \mathbf{r}_t \odot \mathbf{M}_{nh,t}\right) \tag{11}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \mathbf{n}_t. \tag{12}$$

The predicted DPD outputs are generated by a final FC layer

$$\left[\hat{I}_t,\ \hat{Q}_t\right] = \hat{\mathbf{y}}_t = \mathbf{W}_{\hat{y}}\boldsymbol{h}_t + \mathbf{b}_{\hat{y}}. \tag{13}$$

The same process can be used to convert the JANET algorithm into a DeltaJANET-based DPD. The delta updating rules of DeltaGRU and DeltaJANET both follow (3) and (4).

C. Theoretical Operation and Memory Access Savings

In DeltaGRU DPD, the arithmetic operations and memory accesses are dominated by the M × V in (5)–(8). By further considering the overhead in (3) and (4) an assuming all vectors have length n and the weight matrices have dimensions $n \times n$, the dense/sparse computational cost C_{comp} and memorial cost C_{mem} for calculating M × V and M × SV are given as

$$C_{\text{comp,dense}} = n^2 \tag{14}$$

$$C_{\text{comp,sparse}} = (1-\Gamma)n^2 + 2n \tag{15}$$

$$C_{\text{mem,dense}} = n^2 + n \tag{16}$$

$$C_{\text{mem,sparse}} = (1-\Gamma)n^2 + 4n \tag{17}$$

where Γ is the overall temporal sparsity by considering zeros in both $\Delta\boldsymbol{\phi}$ and $\Delta\mathbf{h}$. Therefore, the theoretical computation speedup and memory access reduction of a DeltaDPD are approximated as

$$\text{Speedup} \approx \frac{n}{(1-\Gamma)n + 2} \tag{18}$$

$$\text{Memory Access Reduction} \approx \frac{n+1}{(1-\Gamma)n + 4}. \tag{19}$$

In RNN-based DPD tasks with 500–1000 parameters, the value of n for an RNN structure typically ranges from 8 to 20. For example, in DeltaGRU-1067, n equals 15. Considering the overhead terms in (15) and (17), only sparsity greater than 27% can lead to useful memory access reduction larger than 1 [see (19)]. Although we give the complete (18) and (19), for easy comparison and presentation of results, we estimate the number of active parameters during DeltaDPD inference by

$$\begin{aligned}\#\text{Active Params} = {} & \#\text{DeltaGRU Params} \times \Gamma \\ & + \#\text{FC Params}.\end{aligned} \tag{20}$$

III. Experimental Results

A. Experimental Setup

Fig. 2 illustrates the experimental setup. The `TM3.1a` 5 × 40 MHz (200-MHz) 256-QAM OFDM baseband I/Q signal with 10.01 dB peak-to-average power ratio (PAPR)

TABLE I

LINEARIZATION PERFORMANCE OF DIFFERENT DPD MODELS EVALUATED WITH TM3.1A 200-MHZ FIVE-CHANNEL ×40-MHZ 256-QAM OFDM SIGNALS SAMPLED AT 983.04 MHZ ALONGSIDE THEIR ESTIMATED DYNAMIC POWER CONSUMPTION IN 7 NM WITH FP32 PARAMETER PRECISION [16]

Class	DPD Models	Θ_h	Temporal Sparsity	#Active Params	NMSE (dB)	EVM[a] (dB)	ACPR (dBc)	Number of MUL/ADD/MEM	Energy/Inference (nJ)	Energy Reduction
Prior DPD	RVTDCNN [17]	-	-	1007	-31.64	-32.43	-51.75	1063/1975/1019	9.35	-
	PG-JANET [3]			1130	-39.77	-39.94	-52.91	1144/3397/1133	10.54	
	DVR-JANET [4]			1097	-38.02	-38.24	-53.79	1111/2464/1100	10.10	
This Work[b]	DeltaGRU-1067	0	0%	1067	-40.01	-42.23	-54.02	1083/2499/1204	10.85	1x
	DeltaGRU-889	0.008	20%	889	-39.36	-38.95	-52.50	905/2321/1026	9.25	1.2x
	DeltaGRU-766	0.016	31%	766	-38.73	-38.58	-52.01	782/2198/903	8.15	1.3x
	DeltaGRU-573	**0.05**	**52%**	**573**	**-37.22**	**-38.52**	**-50.03**	**589/2005/710**	**6.41**	**1.7x**
	DeltaGRU-504	0.1	60%	504	-36.67	-37.83	-49.22	520/1936/641	5.80	1.9x
	DeltaGRU-391	0.4	71%	391	-34.26	-35.14	-48.20	407/1823/528	4.78	2.1x
	DeltaJANET-1062	0	0%	1062	-38.50	-40.29	-52.45	1078/2494/1198	10.80	1x
	DeltaJANET-845	0.004	22%	845	-38.66	-39.42	-51.73	861/2277/981	8.85	1.2x
	DeltaJANET-725	0.008	33%	725	-38.40	-39.37	-51.40	741/2157/861	7.78	1.4x
	DeltaJANET-593	**0.012**	**45%**	**593**	**-38.31**	**-39.14**	**-50.20**	**609/2025/729**	**6.59**	**1.6x**
	DeltaJANET-449	0.03	60%	449	-36.78	-36.72	-49.05	465/1881/585	5.30	2.0x
	DeltaJANET-377	0.05	66%	377	-35.33	-35.06	-48.54	393/1809/513	4.65	2.3x

[a] Due to limitations in the experimental setup, the EVM is calculated based on the input signal and the measured output signal rather than the reference grid and the measured output signal. Additionally, the mild CFR applied to the input signal may cause a degradation in the EVM.
[b] We use $\Theta_\phi = 0$ for all DeltaDPDs in this table.

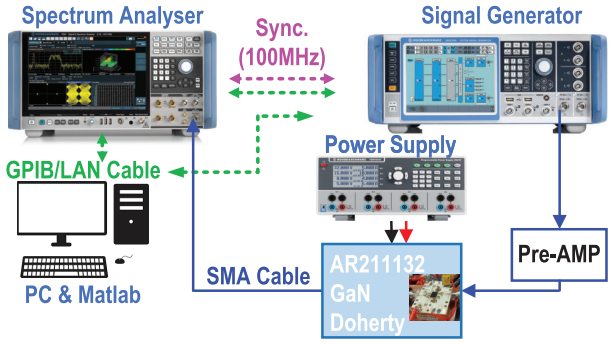

Fig. 2. Setup for dataset acquisition and DPD performance measurement.

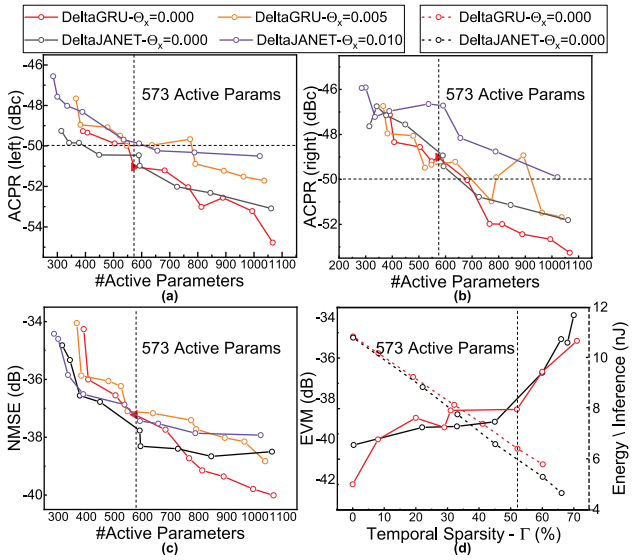

Fig. 3. Activated parameter scan of DPD models versus (a) ACPR (left) and (b) ACPR (right). (c) NMSE. (d) Sparsity of 1067-parameter-GRU versus EVM (left Y-axis) and estimated dynamic power (right Y-axis).

was emitted by R&S-SMW200A and amplified by a 3.5-GHz GaN Doherty PA at 41.5-dBm average output power with and without DPD. The output signal was digitized using an R&S-FSW43 analyzer. Since this spectrum analyzer lacks error vector magnitude (EVM) calculation capability, the EVM was determined by comparing the input signal with the digitized output signal instead of using the reference grid. The dataset, comprising 98304 samples, was divided into 60%, 20%, and 20% for training, validation, and testing.

The end-to-end DPD learning process involves back-propagation through a pretrained 2751-parameter −40.04-dB normalized mean square error (NMSE) DGRU PA behavioral model [18] with the newly measured PA dataset. The models were trained for 200 epochs using the ADAMW optimizer with an initial learning rate of $5E^{-3}$ with ReduceOnPlateau decay and a batch size of 64.

B. Results and Discussion

Table I compares the NMSE, ACPR, and EVM results for different DPD models alongside the number of MUL, ADD operations, and 8-kB SRAM accesses. The estimation method follows [11]. The DeltaGRU-573 DPD model with Θ_ϕ of 0, Θ_h of 0.05 achieves an ACPR of −50.03 dBc, an NMSE of −37.22 dB, and an EVM of −38.52 dB while estimated to consume 6.41 nJ per inference in 7-nm technology. The DeltaGRU-573 demonstrates the most considerable power reduction while maintaining the ACPR better than −50 dBc, as highlighted by the horizontal dashed lines in Fig. 3.

Fig. 3(a)–(c) shows the correlation between ACPR/NMSE and estimated energy/inference against #active parameters of DeltaGRU/DeltaJANET covering 300–1100 active parameters. Even at a high temporal sparsity of around 70% with around 400 active parameters, DeltaGRU and DeltaJANET still maintain ACPR values better than −48 dB. Comparing the performance of various Θ_ϕ, utilizing temporal sparsity of input feature even close to 0 in the DPD task degrades the linearization performance by 1.57 dB because the DPD performance is highly sensitive to the I/Q sampling rate. Fig. 3(d) presents the estimated energy per inference in 7 nm

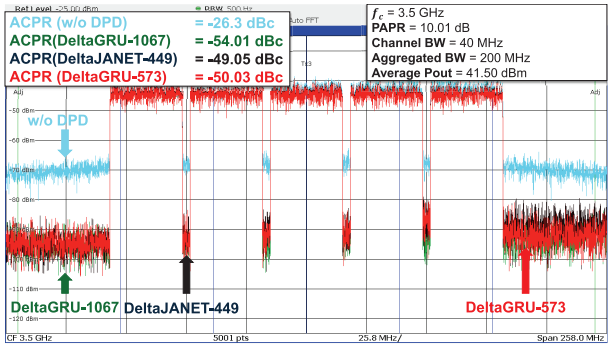

Fig. 4. Measured spectrum on the 200-MHz TM3.1a signal.

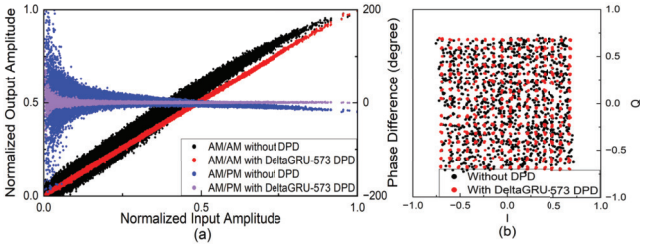

Fig. 5. (a) AM/AM and AM/PM characteristics. (b) Constellation map with and without DPD for the 200-MHz OFDM signal.

of DeltaDPDs. The DeltaGRU-573 model realizes a 1.7× power reduction over the DeltaGRU-1067 network.

Fig. 4 displays the measured spectrum with and without DPDs, which confirms that the DeltaGRU-573 model achieves ACPR of −50 dBc. Fig. 5 exhibits the AM/AM, AM/PM characteristics and constellation map with and without DPDs.

C. Comparison to Prior Works

Due to power constraints in DPD applications, state-of-the-art models are typically limited to around 1000 parameters [5], making NN performance particularly susceptible to compression and sparsity of input compared to delta networks with parameters more than 160 000 in other domains [12], [13]. The previous approaches of lightening the DPD model have primarily relied on static spatial weight pruning of NN weights [10], [19]. Using a 100-MHz OFDM signal, Liu et al. [10] achieved an ACPR of −45.5 dBc with a pruned convolutional NN-based DPD model containing 106 parameters, reduced from 158. Li et al. [19] demonstrated an ACPR of −45.1 dBc at 200 MHz using a pruned phase-normalized time-delay NN with 909 parameters. However, these unstructured pruning methods create irregular distributions of nonzero values in weight matrices, causing unbalanced workloads among hardware arithmetic units and limiting real speedup or efficiency gains in actual hardware implementations. In contrast, our proposed DeltaDPD achieves a superior ACPR of −50.03 dBc at 200 MHz with only 573 parameters while maintaining structure.

IV. Conclusion

This work introduces DeltaDPD, a novel method for energy-efficient RF PA linearization that leverages dynamic temporal sparsity. By reducing computational complexity and memory access compared to conventional approaches, DeltaDPD achieves power savings while maintaining robust linearization performance.

References

[1] S. Wesemann, J. Du, and H. Viswanathan, "Energy efficient extreme MIMO: Design goals and directions," *IEEE Commun. Mag.*, vol. 61, no. 10, pp. 132–138, Jul. 2023.

[2] H. Li, Y. Zhang, G. Li, and F. Liu, "Vector decomposed long short-term memory model for behavioral modeling and digital predistortion for wideband RF power amplifiers," *IEEE Access*, vol. 8, pp. 63780–63789, 2020.

[3] T. Kobal, Y. Li, X. Wang, and A. Zhu, "Digital predistortion of RF power amplifiers with phase-gated recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 6, pp. 3291–3299, Jun. 2022.

[4] T. Kobal and A. Zhu, "Digital predistortion of RF power amplifiers with decomposed vector rotation-based recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4900–4909, Nov. 2022.

[5] A. Fischer-Bühner, L. Anttila, M. Dev Gomony, and M. Valkama, "Recursive neural network with phase-normalization for modeling and linearization of RF power amplifiers," *IEEE Microw. Wireless Technol. Lett.*, vol. 34, no. 6, pp. 809–812, Jun. 2024.

[6] Y. Li, X. Wang, and A. Zhu, "Sampling rate reduction for digital predistortion of broadband RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 3, pp. 1054–1064, Mar. 2020.

[7] N. Hammler, A. Cathelin, P. Cathelin, and B. Murmann, "A spectrum-sensing DPD feedback receiver with *30*× reduction in ADC acquisition bandwidth and sample rate," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 9, pp. 3340–3351, Sep. 2019.

[8] Y. Li, X. Wang, and A. Zhu, "Reducing power consumption of digital predistortion for RF power amplifiers using real-time model switching," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 3, pp. 1500–1508, Mar. 2022.

[9] M. Beikmirza, L. C. N. de Vreede, and M. S. Alavi, "A low-complexity digital predistortion technique for digital I/Q transmitters," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2023, pp. 787–790.

[10] Z. Liu, X. Hu, L. Xu, W. Wang, and F. M. Ghannouchi, "Low computational complexity digital predistortion based on convolutional neural network for wideband power amplifiers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1702–1706, Mar. 2022.

[11] Y. Wu et al., "MP-DPD: Low-complexity mixed-precision neural networks for energy-efficient digital predistortion of wideband power amplifiers," *IEEE Microw. Wireless Technol. Lett.*, vol. 34, no. 6, pp. 817–820, Jun. 2024.

[12] S. Liu, S. Zhou, Z. Li, C. Gao, K. Kim, and T. Delbrück, "Bringing dynamic sparsity to the forefront for low-power audio edge computing: Brain-inspired approach for sparsifying network updates," *IEEE Solid StateCircuits Mag.*, vol. 16, no. 4, pp. 62–69, Jan. 2024.

[13] D. Neil, J. H. Lee, T. Delbruck, and S.-C. Liu, "Delta networks for optimized recurrent network computation," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2584–2593.

[14] J. van der Westhuizen and J. Lasenby, "The unreasonable effectiveness of the forget gate," 2018, *arXiv:1804.04849*.

[15] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734. [Online]. Available: http://www.aclweb.org/anthology/D14-1179

[16] N. P. Jouppi et al., "Ten lessons from three generations shaped Google's TPUv4i: Industrial product," in *Proc. ACM/IEEE 48th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2021, pp. 1–14.

[17] X. Hu et al., "Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3923–3937, Aug. 2022.

[18] Y. Wu, G. Deep Singh, M. Beikmirza, L. C. N. de Vreede, M. Alavi, and C. Gao, "OpenDPD: An open-source end-to-end learning & benchmarking framework for wideband power amplifier modeling and digital pre-distortion," 2024, *arXiv:2401.08318*.

[19] W. Li, R. Criado, W. H. Thompson, K. Chuang, G. Montoro, and P. L. Gilabert, "GPU-based implementation of pruned artificial neural networks for digital predistortion linearization of wideband power amplfiers," *TechRxiv*, Jul. 2024.