ISSCC 2026 Forums

Open-Source AI for Analog Correction: From RF Power Amplifiers to Energy-Efficient Silicon

Chang Gao
www.tudemi.com

Department of Microelectronics
Delft University of Technology

Linearity is Now an Energy Problem

Extreme mMIMO

Wi-Fi 7/8

NTN/Direct-to-Cell

Common Challenges:

1. **Wider bandwidth** → harder to stay linear across frequency
2. **Higher-order QAM** → tiny error budget (EVM)
3. **Many antennas + coupling** → distortion becomes multi-dimensional

- Linearity is not "just RF quality". It sets **efficiency** and thus **heat dissipation**.
- AI-based calibration could help, but **its computation must be tiny**.

[1] Nokia Bell Labs, White Paper, 2022.

Where the Energy Goes: The Radio Unit (RU)

Industry Reality: Electricity Consumption

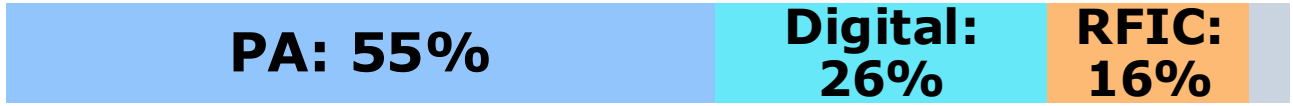
Base Station Power Breakdown

- The Radio Unit (RU) dominates

Control

Inside a mMIMO RU (@100% Load)

RU Power Breakdown



- DPD/CFR live in the Digital/RFIC blocks

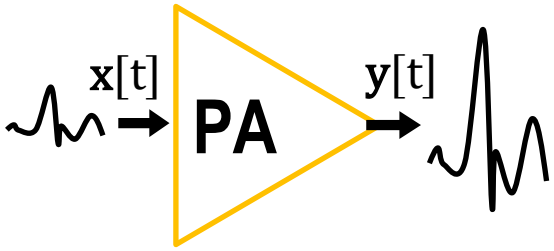
PA PSU

- RAN dominates at scale:
  - RU is the single largest HW
- **Power Amplifier (PA) is the largest term:**
  - ~55% of RU energy at full load goes to the PA

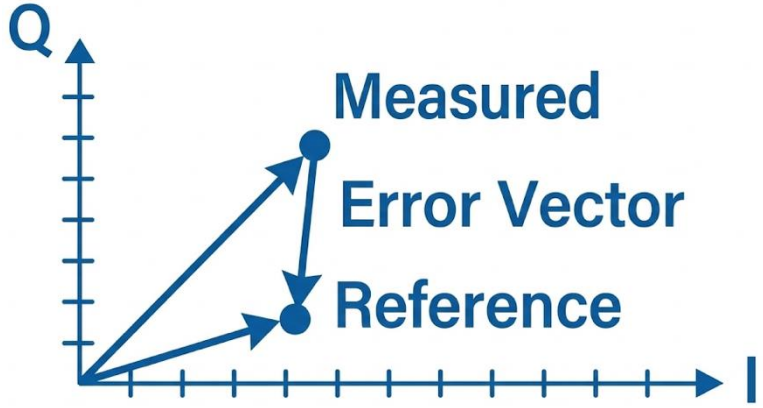
[2] Ericsson, White Paper, 2024.

PA's Linearity Metrics

- The last and most power-hungry block in the TX chain
- PA boosts RF power, but becomes nonlinear near saturation
  - **In-band distortion** → constellation gets "blurry"
  - **Out-of-band regrowth** → Spurious leakage to adj. channels

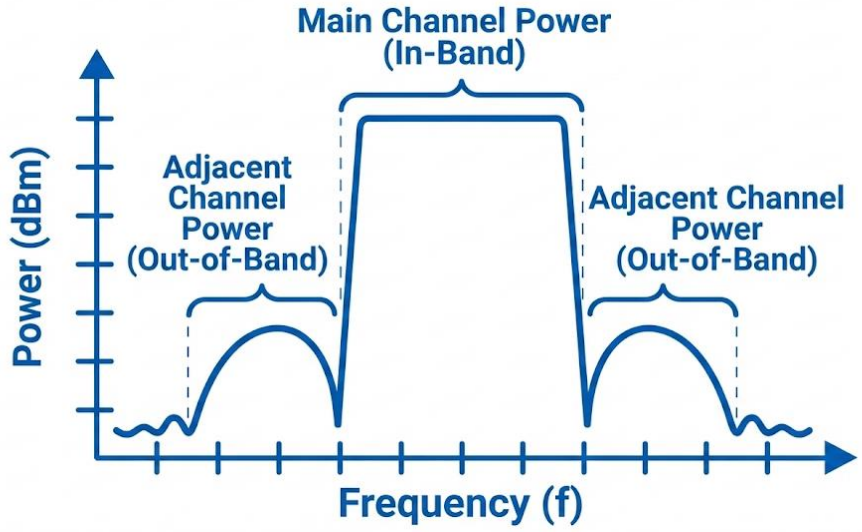


Metrics of Linearity:



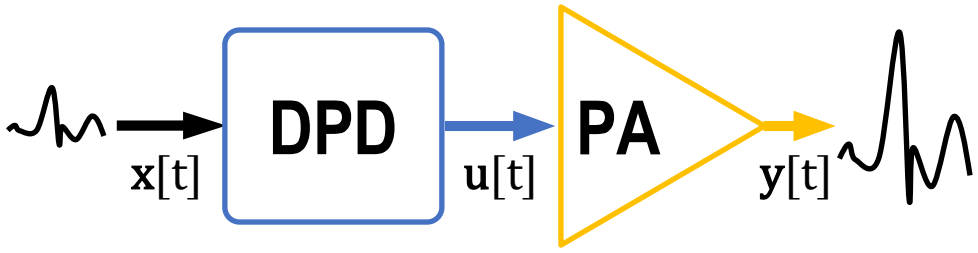
$$\mathbf{EVM}_{\%} = \frac{\mathrm{RMS}(|(I_m + jQ_m) - (I_r + jQ_r)|)}{\mathrm{RMS}(|(I_r + jQ_r)|)}$$

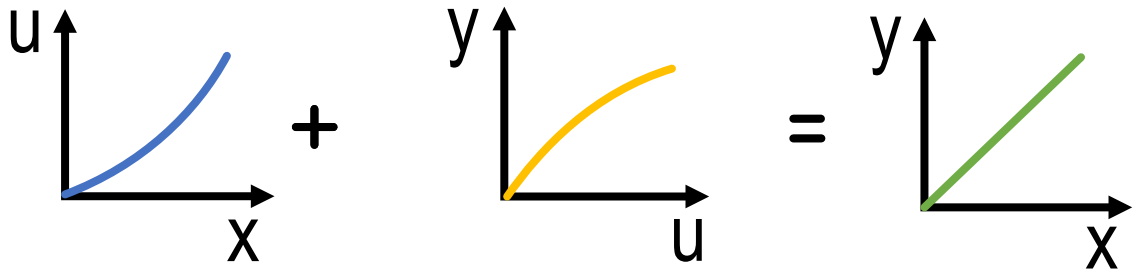
$$\mathbf{EVM}_{\mathbf{dB}} = 20\log_{10}\mathrm{EVM}_{\%}$$



$$\mathbf{ACPR} = 10\log_{10}\frac{\text{Adjacent Channel Power}}{\text{Main Channel Power}}$$

Digital Predistortion (DPD)



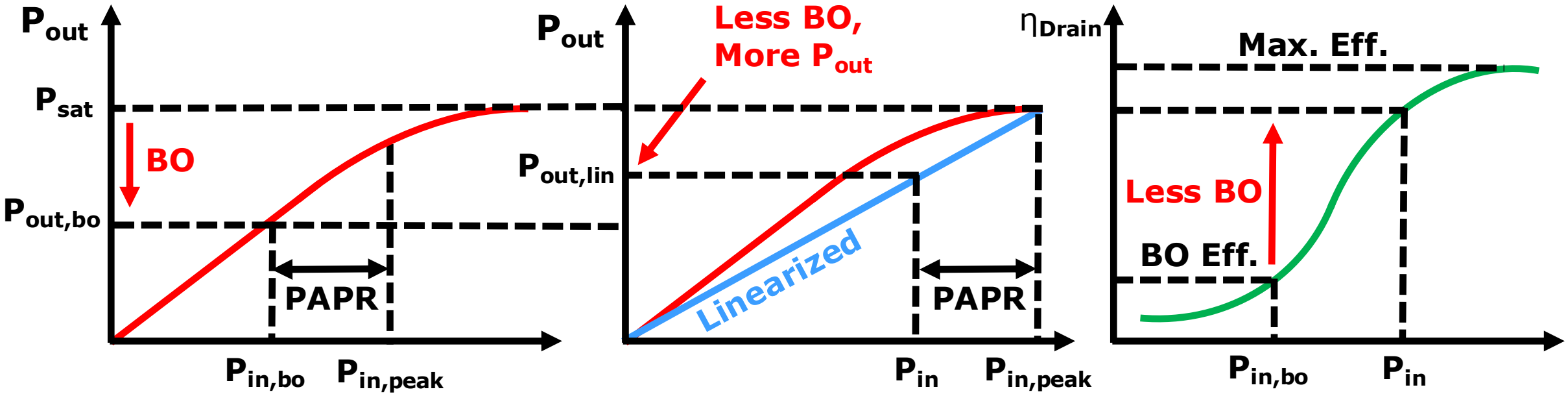


A DPD generates u[t] so that PA output becomes linear: y[t] ≈ G·x[t]

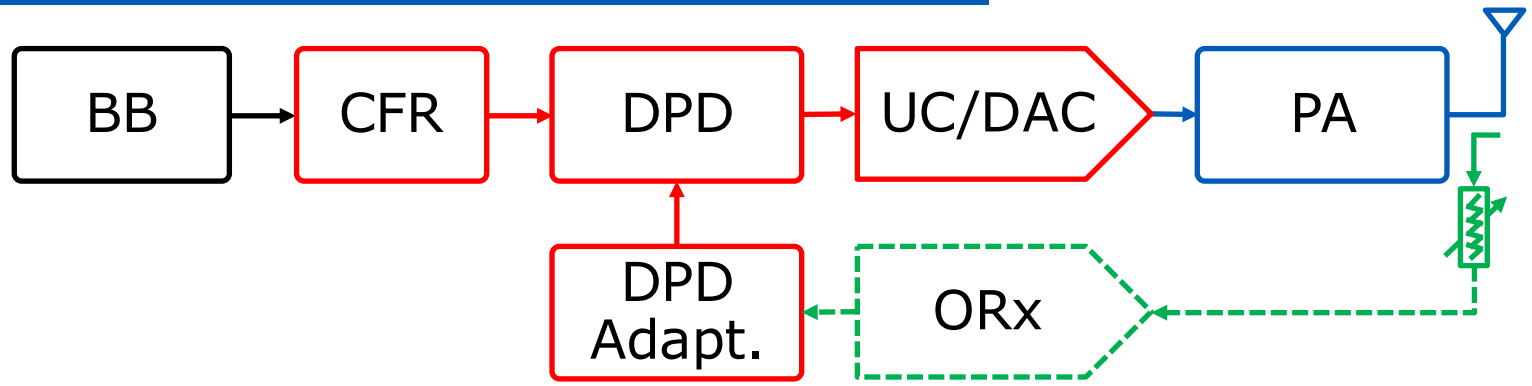
- **In-band linearity:** lower EVM → tighter constellation
- **Spectral compliance:** lower ACLR/ACPR (less spectral regrowth) → meet SEM mask
- **Efficiency at spec:** same EVM/ACLR with less back-off → higher efficiency η

PA Linearization Improves Efficiency

- High-order modulated signals (e.g., 5G-NR OFDM) have high PAPR (~10 dB)
- We must backoff (BO) from P_{sat} to meet spec. (e.g., <-45 dBc ACLR, <3.5% EVM for 256-QAM) but sacrifice efficiency.
- With linearization, we can run closer to saturation→ higher P_{out} and higher efficiency



When is DPD Worthy?



- DPD delivers linearity without the energy inefficiency of deep back-off.
  - DPD → less back-off → higher efficiency @ spec
- Robust DPD needs online adaptation (duty-cycled every 1~10 ms or longer)
- Net Power Gain Condition:
  - $P_{total}=P_{PA}+P_{DPD}+P_{ORx}$
  - Worth it if: $\Delta P_{PA} >> P_{DPD}+P_{Orx}$
- DPD's overhead must be **minimized** to yield a net power reduction
- Typical DPD Processor Power is **< 2 Watts** [3]

[3] Texas Instruments., GC5325 datasheet, 2009.

Wideband Waveforms Push DPD Complexity

System	Channel BW
LTE/4G	20 MHz
5G NR	100 MHz
5G-Advanced	200 MHz
Wi-Fi 7/8	160~320 MHz
6G Macro	100~400 MHz

- Wideband OFDM + high-order QAM amplifies the **memory effects** of PAs.
  - Bigger DPD model size
- DPD runs typ. @ 3x~5x BW
  - More DPD computations
- We are facing **linearity-per-watt** pressure

[4] ETSI, ES 203 700 V1.1.1, 2020.
[5] Huawei, 5G Power White Paper, 2019

Scaling Problem of Classical Volterra DPDs

General Form of Volterra DPDs

$$\hat{y} = \sum_{i=0}^{I-1} \alpha_i \, \beta_i(x)$$

Learnable Coefficient

Basis Function

Rule: Reduce I ⇒ Implementable

Challenge: Model Requirement↑

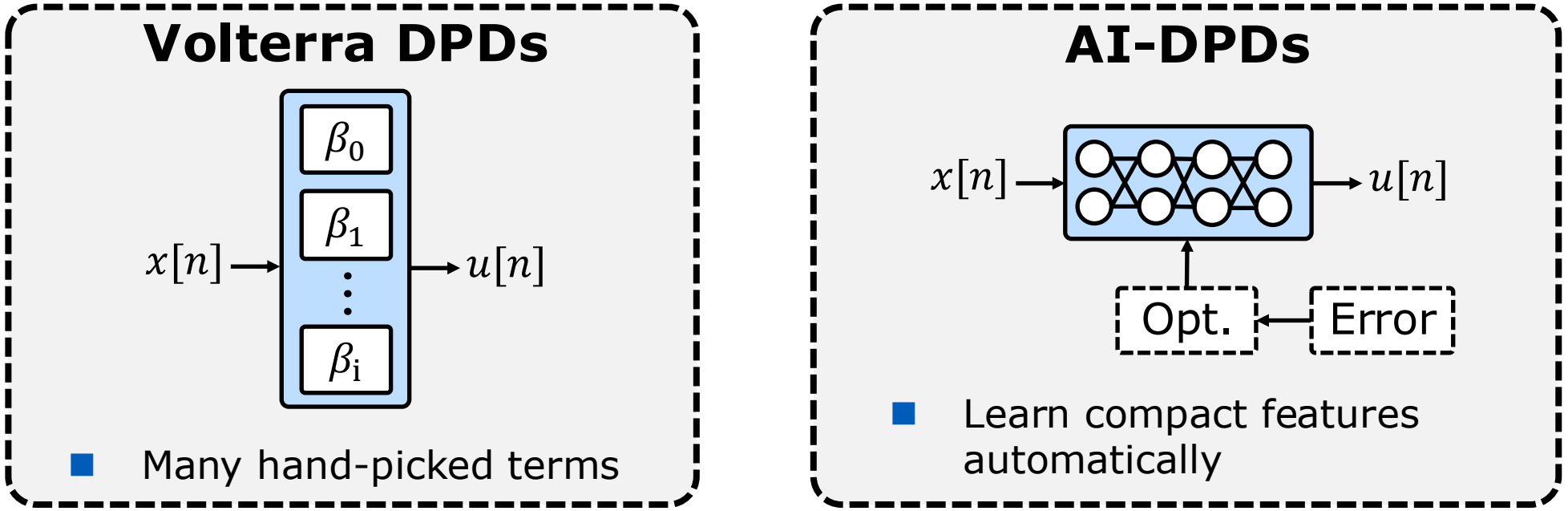
- BW↑ ⇒ Memory Effects↑
- Arrays↑⇒ #PAs↑ ⇒ #DPDs↑

Dilemma:

- Grow I ⇒ Explode Comp./Mem.
- Small I ⇒ Bad Linearization

- Classical DPD depends on a **hand-designed** basis set $\{\beta_i\}$.
- **Memory Polynomial (MP)**
  - $i \mapsto (K, M)$ for diagonal terms
  - $\beta_{k,m}(x) = \sum_m^M \sum_k^K \alpha_{m,k} \; x(n-m)|x(n-m)|^k$
- **Generalized Memory Polynomial (GMP)**
  - $i \mapsto (K, M, Q)$ for selected cross-terms
  - $\beta_{k,m,q}(x) = \sum_m^M \sum_k^K \sum_q^Q \alpha_{m,k,q} \; x(n-q)|x(n-m)|^k$
- **Compute and model size scale with I**
- **Problems:**
  - Wideband signals requires more terms
  - **DPD model size (#parameter) explodes**

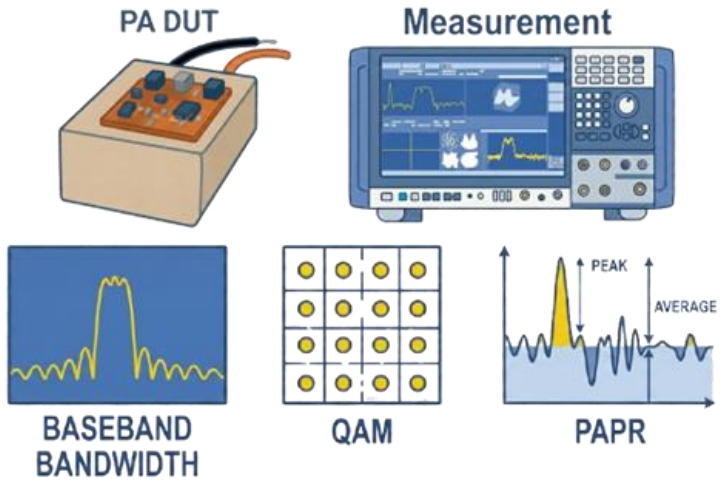
AI-DPD: Learn a Compact Basis from Data



- **MP/GMP = hand-crafted sparsity pattern**
  - We choose the terms; the model only learns coefficients.
  - More and more terms needed when BW/memory effect grows.
- **AI-DPD = learned compact basis**
  - The model learns **which features matter**, and can **reuse** them across memory and nonlinear orders.

Why Do We Need Open-Source AI-DPD?

- DPDs remain hard to compare:
  - People use different measurement devices
  - Different test signal standards
  - An online measurement setup like RF-WebLab [6] is expensive and hard to scale
- Open-source DPD training and testing code is rarely seen



Benchmarking

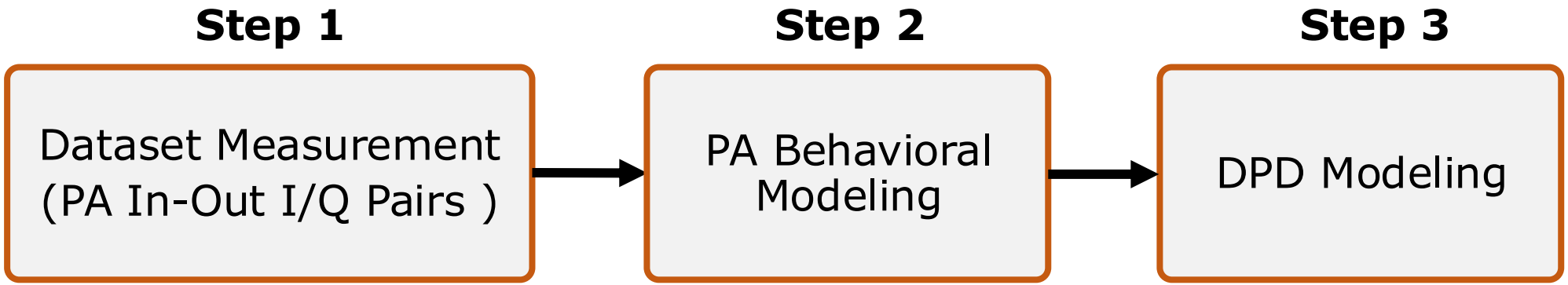
Common Baseline Missing

Reproducibility

Papers rarely Release Code

[6] P. N. Landin et al., IEEE Microw. Mag, 2024.

OpenDPD: End-to-End DPD Learning

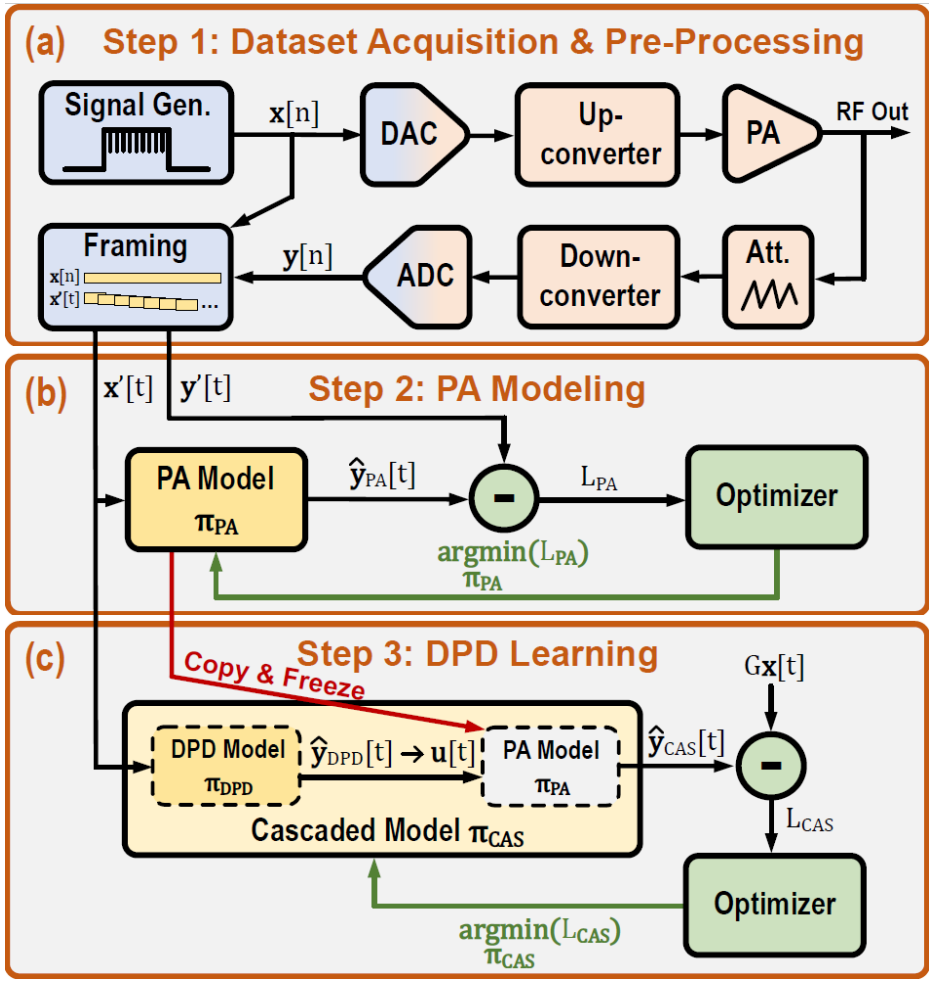


- Fully implemented in **PyTorch** and can be interfaced with **MATLAB**
- Public code, datasets, and pre-trained PA & DPD models
- More PA datasets with various test signals can be plugged in over time
- Same learning & benchmarking pipeline for **fair algorithm modeling capability comparison**
- Available at **www.opendpd.com**

[7] Y. Wu et al., ISCAS 2024.

DPD Learning via a Differentiable PA Model

- **Step 1:** Measure PA in-out I/Q pairs $(\mathrm{x}[\mathrm{n}], \mathrm{y}[\mathrm{n}])$
- **Step 2:** learn π_{PA} from frames $(\mathrm{x}'[\mathrm{t}], \mathrm{y}'[\mathrm{t}])$
- **Step 3:**
  - Freeze π_{PA} and insert π_{DPD} before it to form a cascaded model π_{CAS}
  - Backpropagate through π_{CAS}
  - Update only π_{DPD} coefficient/parameters
- **Gradient Descent Optimization Target:**
  - $\pi^*_{\mathrm{DPD}} = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}\left[\left\|\pi_{\mathrm{PA}}\left(\pi_{\mathrm{DPD}}(x;\theta)\right) - Gx\right\|^2\right]$
  - The loss is the expectation $\mathbb{E}$ approximated by the empirical mean over the training dataset
  - G is the tunable target gain of PA after linearization



[7] Y. Wu et al., ISCAS 2024.

AI-DPD Building Blocks

Fully-Connected Layers

FCN/MLP

- Good for feature mixing
- Easy to parallelize
- Bad for modeling mem. eff.
- Memory-bounded

Recurrent Layers

RNN:GRU/LSTM/SNN

- Longer memory
- Streaming-friendly
- Hard to parallelize
- Mem. Bounded

Convolutional Layers

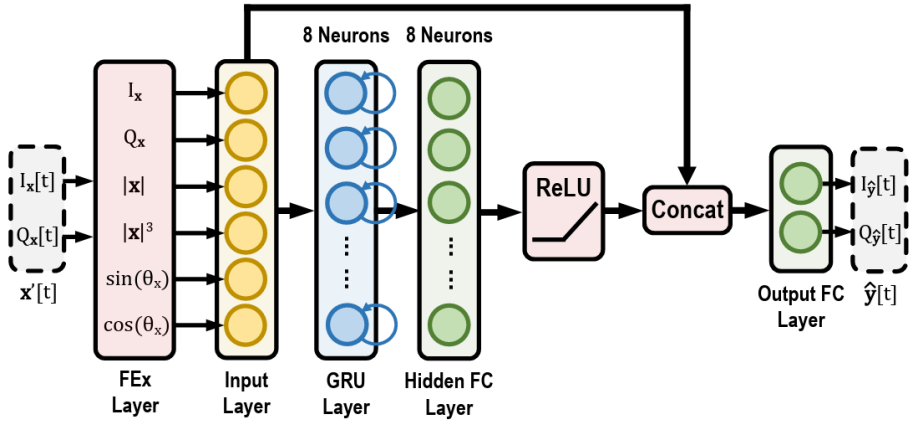
CNN: Conv1D/2D

- Parameter sharing
- Easy to parallelize
- Bad for long mem. effect
- Arithmetically-intensive

- **DPDs Using Each Block Mainly**
  - FCN: [8], [9], [10], [11]
  - RNN: [7], [12], [13], [14], [15], [16]
  - CNN: [18], [19], [20], [21], [22]
- **AI-DPDs are not necessarily only built by neural network (NN) layers**
  - Feature extraction (amplitude, amp^n, phase), phase normalization
- **Hybrid models can be built by mixing diff. NN layer types.**

Benchmarking: AI-DPD vs. Volterra (GMP)

DenseGRU DPD Network



- All with ~500 full-precision parameters
- 40 nm CMOS-based DTX @ 13.4 dBm
- 200 MHz, 10×20 MHz OFDM, 64-QAM @ PAPR = 9.6 dB (38k samples)
- Better measured ACPR/EVM among similar-size models
- **AI-based DPDs outperform GMP** on 200 MHz wideband test signal

DPD Linearization Performance @ ~500 parameters

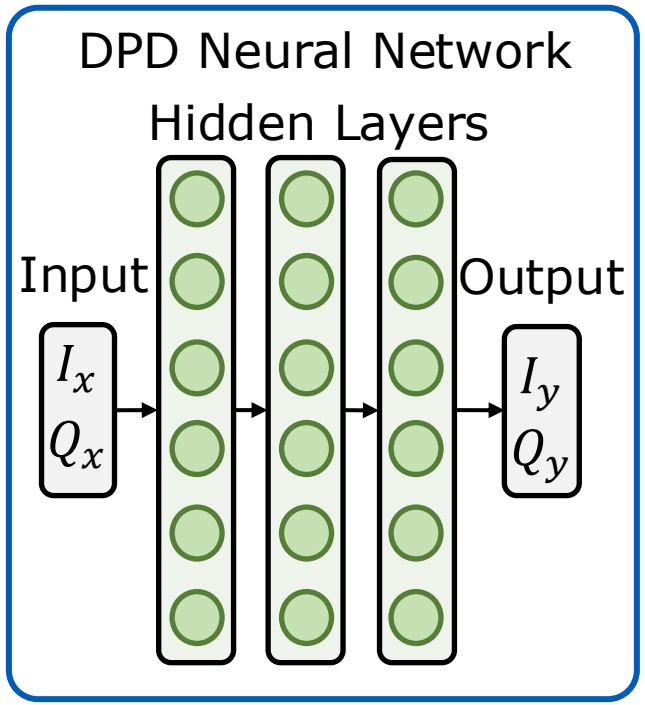
Model	ACPR (dBc)	EVM
GMP	-41.0	2.6%
LSTM	-41.7	1.9%
GRU	-42.4	1.8%
RVTDCNN [18]	-43.8	2.0%
VDLSTM [12]	-44.0	1.9%
DenseGRU [7]	-44.6	1.7%

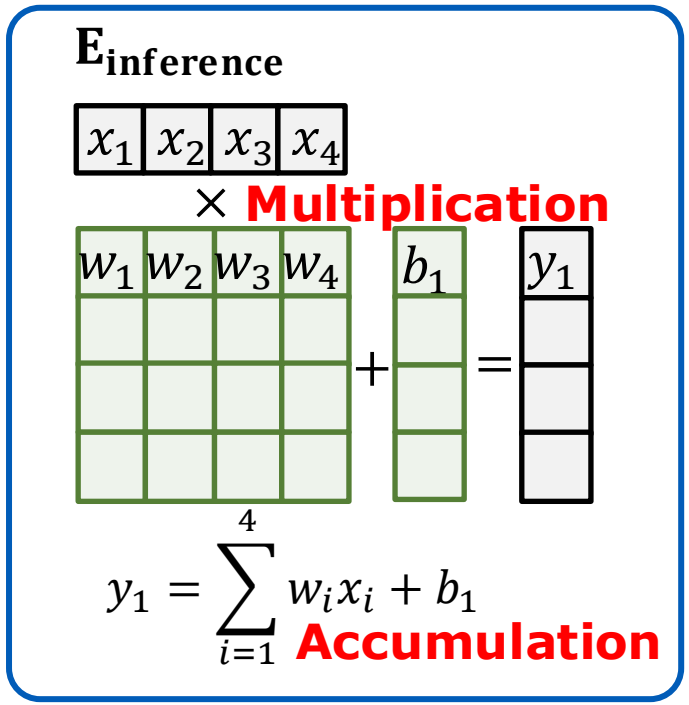
[7] Y. Wu et al., ISCAS 2024.

Power Issue of DPD

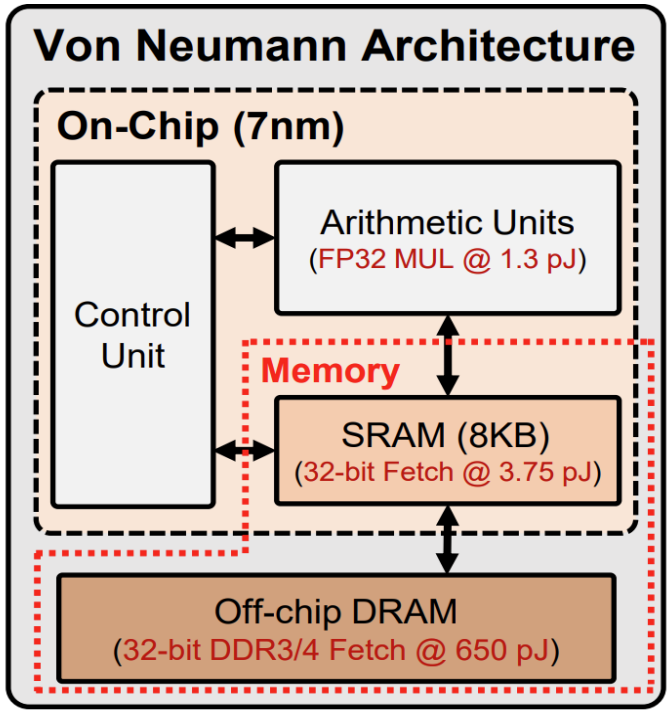
$$P_{inference} = f_s \times E_{inference} = f_s \times (E_{Multiplications} + E_{Additions} + E_{memory})$$

- **6G/WiFi-8: BW >> 200MHz**
- **Typ. DPD f_s : 3.5x to 5x bandwidth**
- **DPD having more parameters**
- **Increasing #MUL, #ADD, #MEM access**

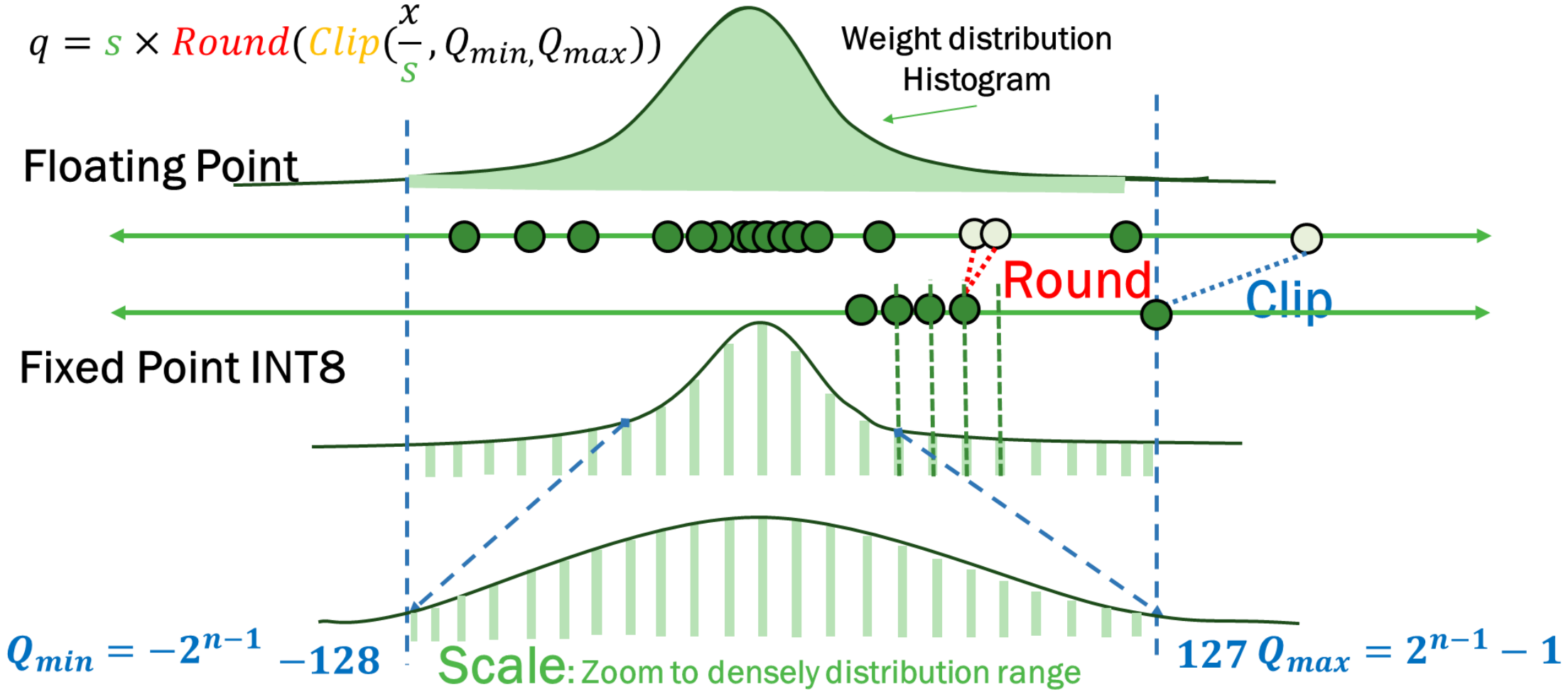




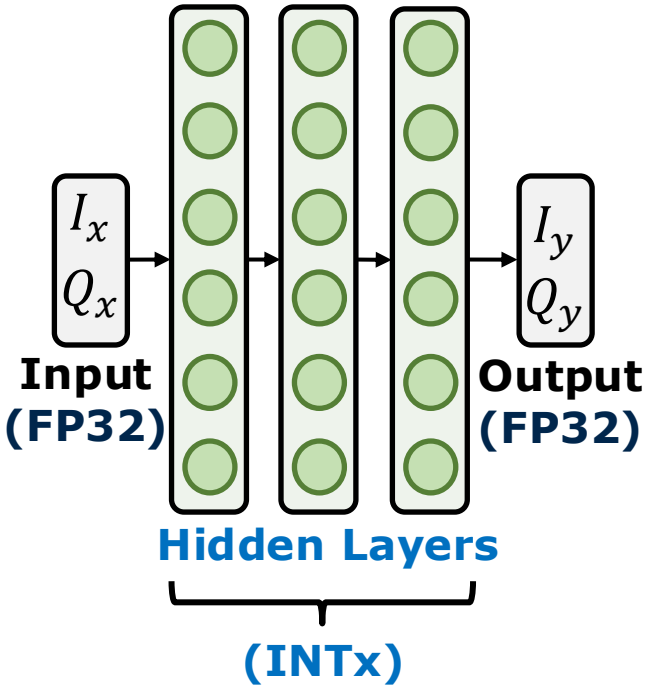
Mem. Wall

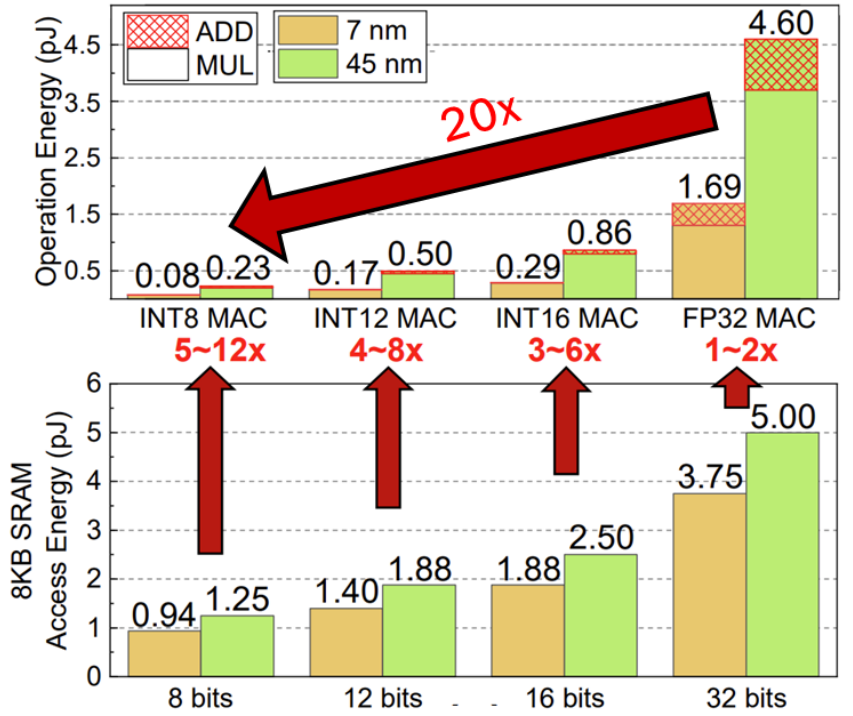


A Simple Mitigation: Quantization



Mixed-Precision Neural Networks for DPD





- Arithmetic Operation Energy Save:
  - FP32 -> INT8 = 20x
- Memory Access Energy Save:
  - FP32 -> INT8 = 4x

Sign-1 bit	Exponent-8 bit	Fraction-23 bit

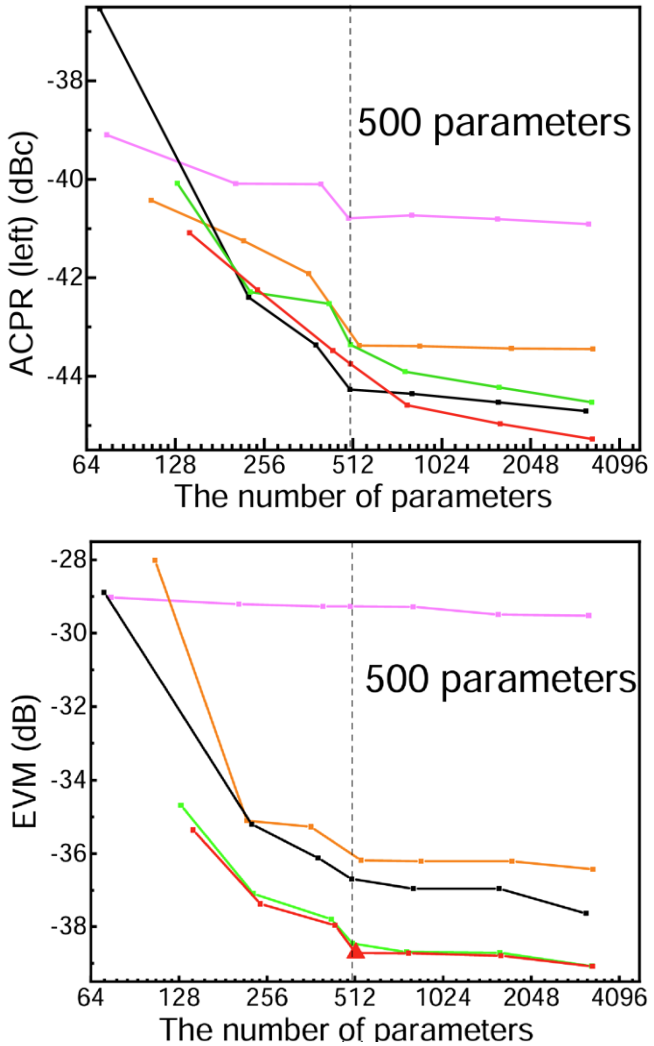
$$FP32 = -1^{S} \times (1 + F) \times 2^{E}$$

Sign-1 bit	Interger-4 bit	Fraction-4 bit

$$INT8 = -1^{S} + I + F$$

[16] Y. Wu et al., IMS, 2024.

Results: MP-DPD w/o Losing Linearization

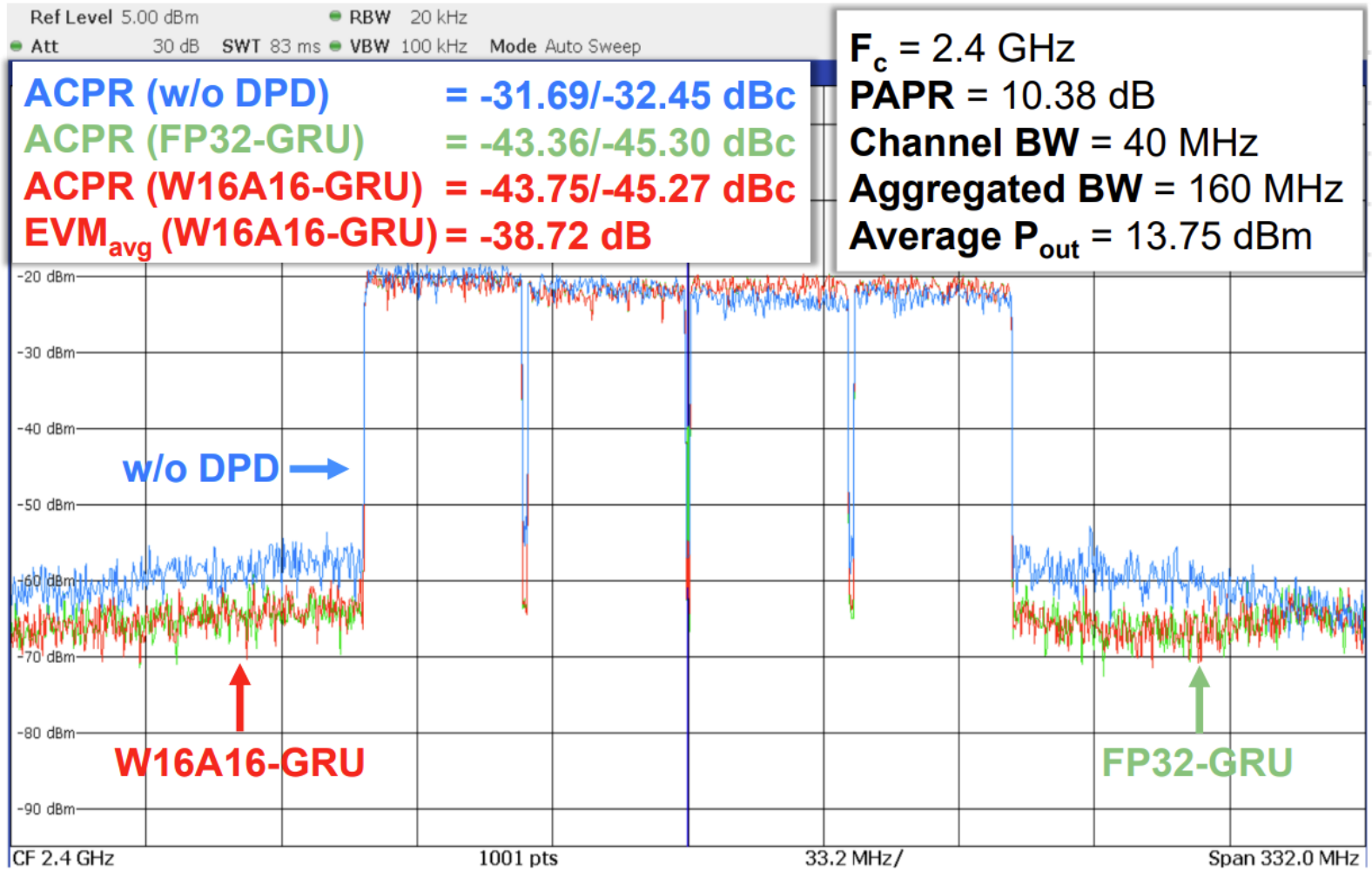


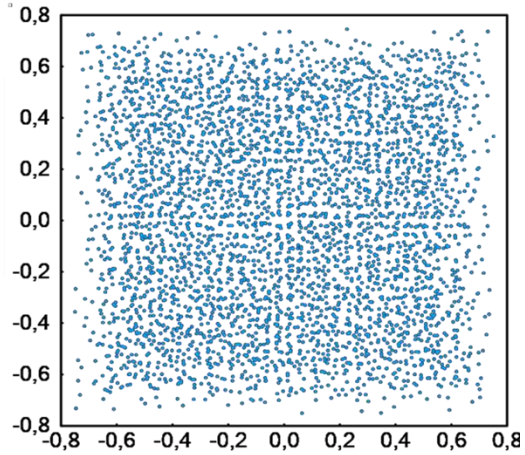
Model Type	EVM	ACPR (dBc)	Energy Reduction (7nm)
GRU (Ours)	1.2%	-44.2	1X
W16A16-GRU	**1.2%**	**-44.5**	**2.8X**
W12A16-GRU	1.3%	-43.9	3.7X
W12A12-GRU	1.3%	-43.1	3.8X
W8A16	1.5%	-42.2	4.2X
W8A12	1.5%	-42.3	4.3X
W8A8	4.5%	-35.8	4.5X

- 16-bit quantization = Deployment Sweet Spot
  - FP32 -> INT16 = **2.8X saving w/o perf. loss**
- Save more if you can tolerate worse ACPR
  - FP32 -> INT12 = 3.8X w/ 1.1 dBc loss

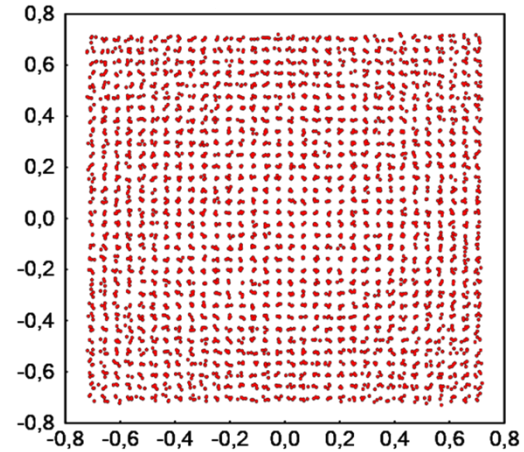
[16] Y. Wu et al., IMS, 2024.

Results: MP-DPD w/o Losing Linearization





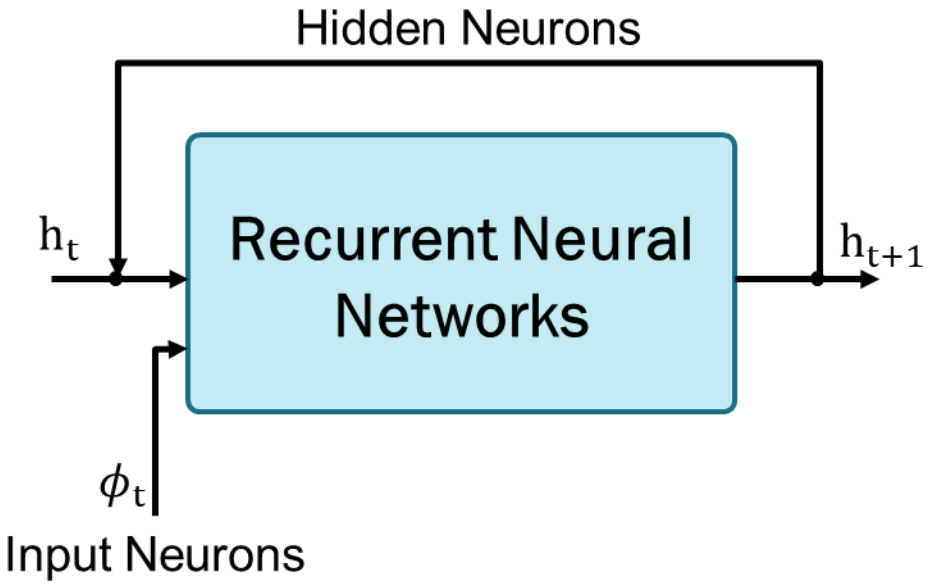
Without DPD

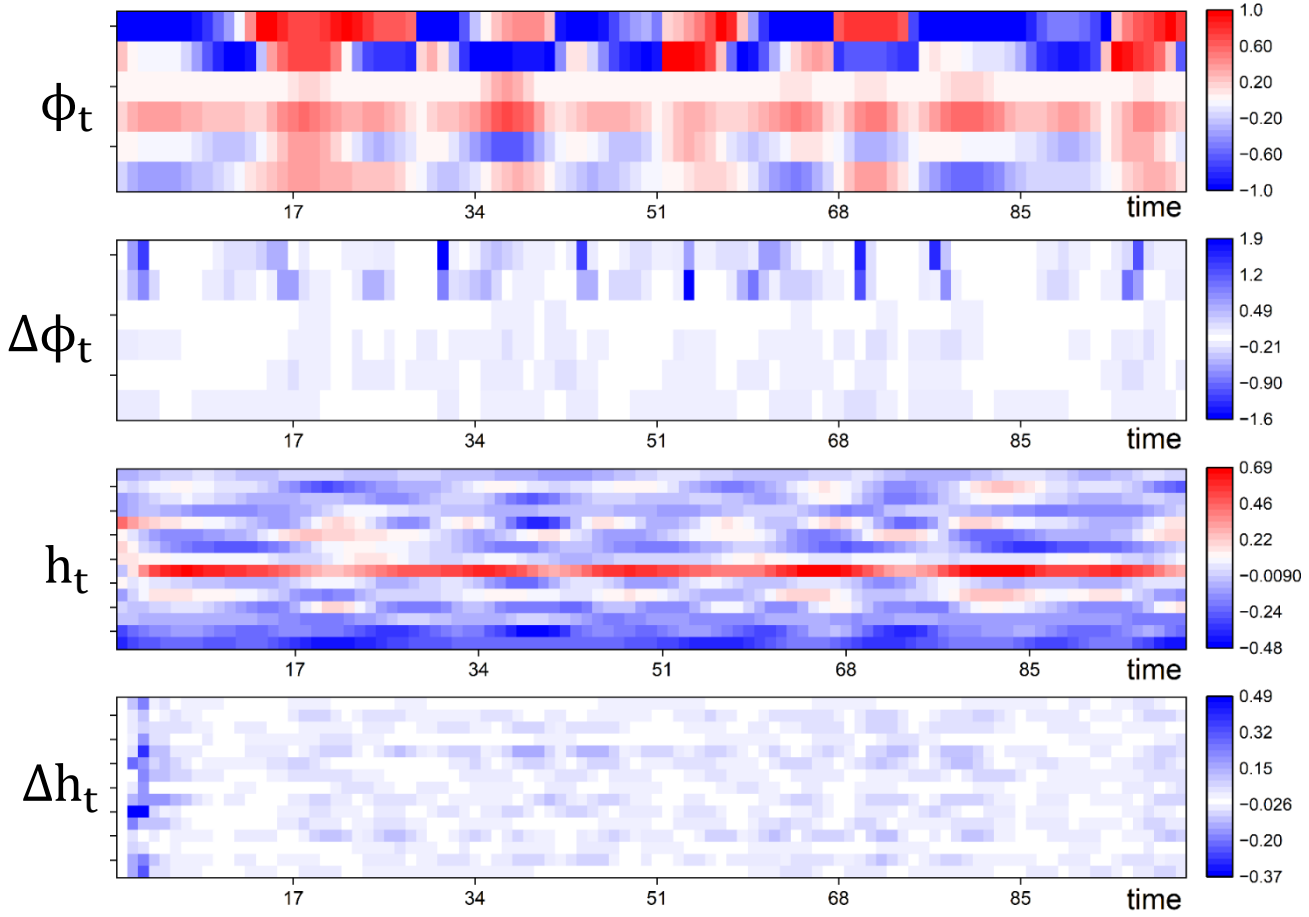


With DPD (W16A16)

[16] Y. Wu et al., IMS, 2024.

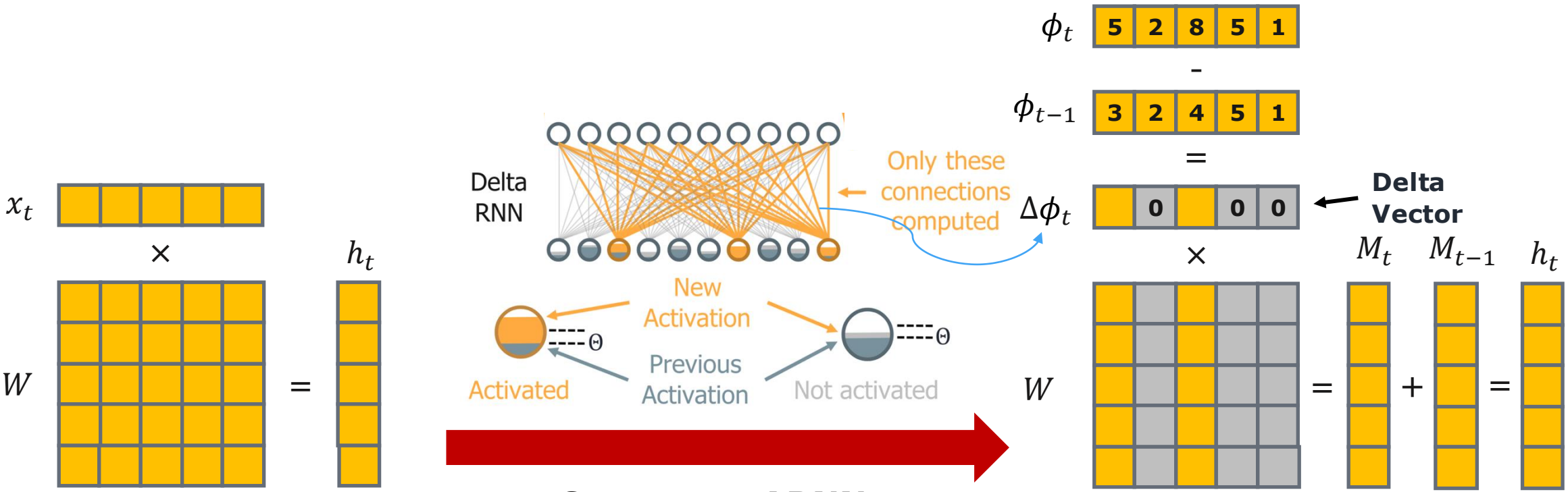
How to Save More? Use Sparsity.





- Classical RNN:
  - $h_t = \sigma\left(\mathrm{W_{\phi h}}\phi_t + \mathrm{W_{hh}}h_{t-1} + \mathrm{b_h}\right)$
- Delta RNN:
  - $h_t = \sigma\left(\mathrm{W_{\phi h}}\Delta\phi_t + \mathrm{W_{hh}}\Delta h_{t-1} + \mathrm{M_{t-1}}\right)$
  - High sparsity in $\Delta\phi_t$, Δh_{t-1}

Dynamic Temporal Sparsity in RNNs



Convert to ΔRNN

Vanilla RNN DPD

Dense Matrix Dense Vector Multiplication

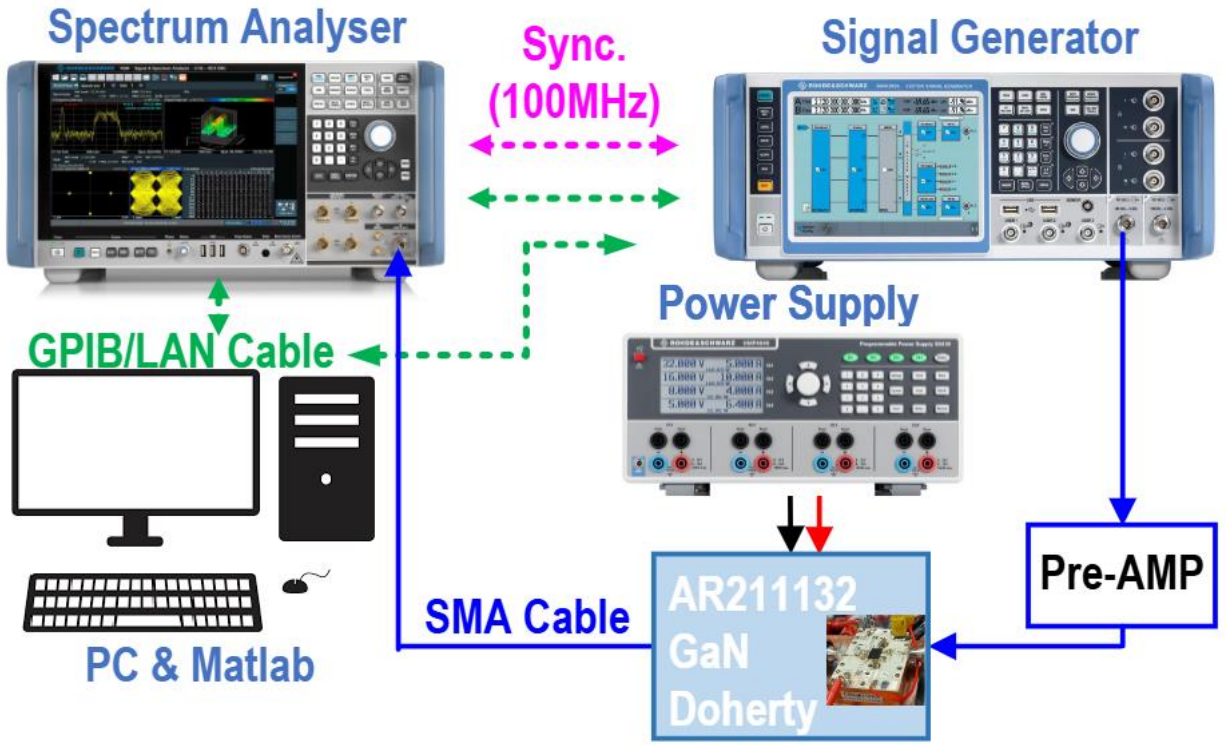
$$\Delta\phi_t = \begin{cases} \phi_t - \phi_{t-1}, & |\phi_t - \phi_{t-1}| \leq \Theta_\phi \\ 0, & |\phi_t - \phi_{t-1}| \leq \Theta_\phi \end{cases}$$

DeltaDPD

Dense Matrix Sparse Vector Multiplication

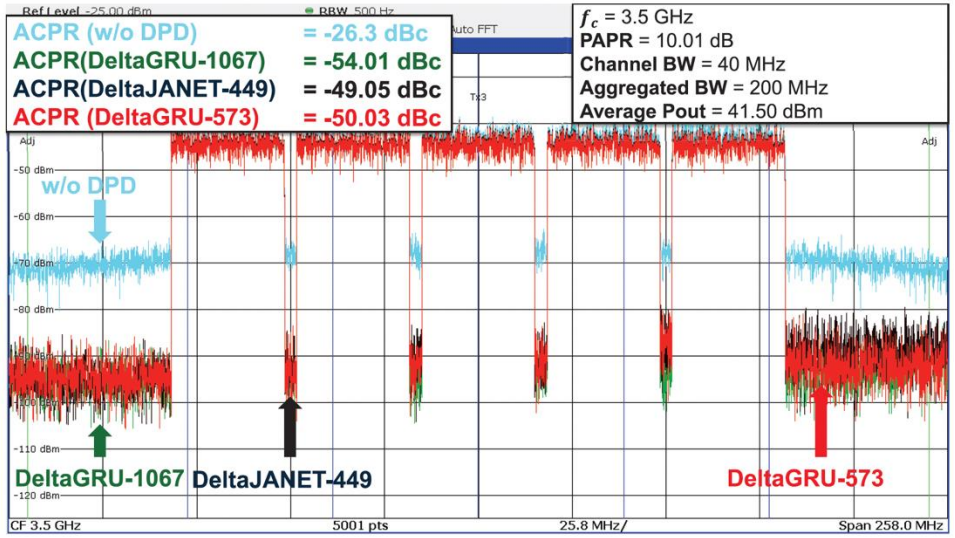
Experimental Setup of DeltaDPD

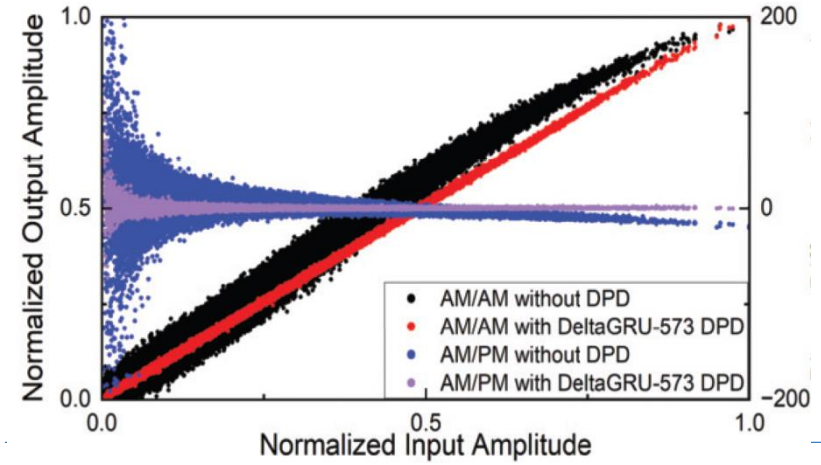
- TM3.1a 200-MHz 256-QAM OFDM @ PAPR = 10 dB
- Ampleon 3.5-GHz GaN Doherty PA @ P_{avg} = 41.5 dBm
- 60-20-20% train/val/test split with 38k samples
- 200 Epochs (rounds of training)



[17] Y. Wu et al., IMS, 2025.

DeltaDPD Results





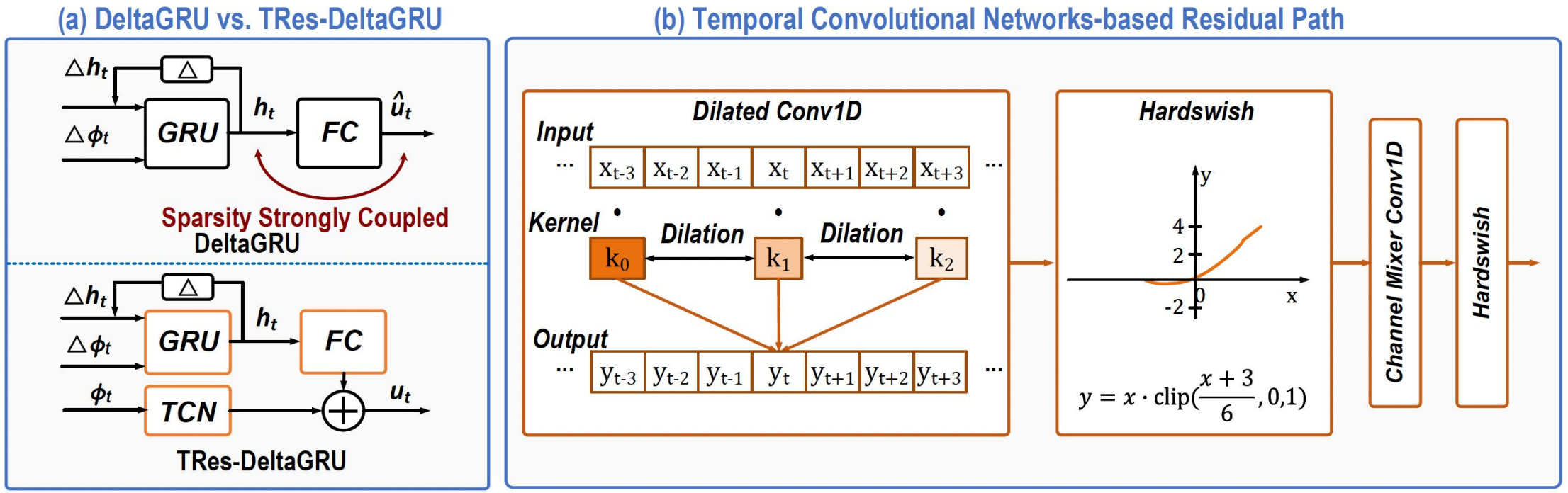
DeltaGRU Linearization Performance

Temporal Sparsity (%)	Active Parameter	ACPR (dBc)	EVM (dB)
0	1067	-54.0	0.8%
20	889	-52.5	1.1%
31	766	-52.0	1.2%
52	573	-50.0	1.2%
71	391	-48.2	1.7%

- -54 dBc ACPR @ 1067 parameters
- -50 dBc ACPR @ 573 active parameters
  - **~2X compute saving**
- Can we improve ACPR @ high sparsity? Yes!

[17] Y. Wu et al., IMS, 2025.

TRes-DeltaGRU: Hybrid-NN for Performance



- **Problem:** DeltaGRU makes DPD outputs too sparse -> worse linearization
- **Solution:** Use temporal conv. net. (TCN) skip path to mix dense input with sparse output
- **Outcome:** Better linearization @ the same cost

[22] Y. Wu et al., arXiv, 2025.

Ablation Study of Design Choices

- A conventional FCN-based residual path doesn't work the best
- TCN-based residual path provides improvements vs. DeltaGRU:
  - 4.4 dBc ACPR
  - 0.1% dB EVM
  - w/ the same number of active parameters

DeltaGRU Linearization Performance

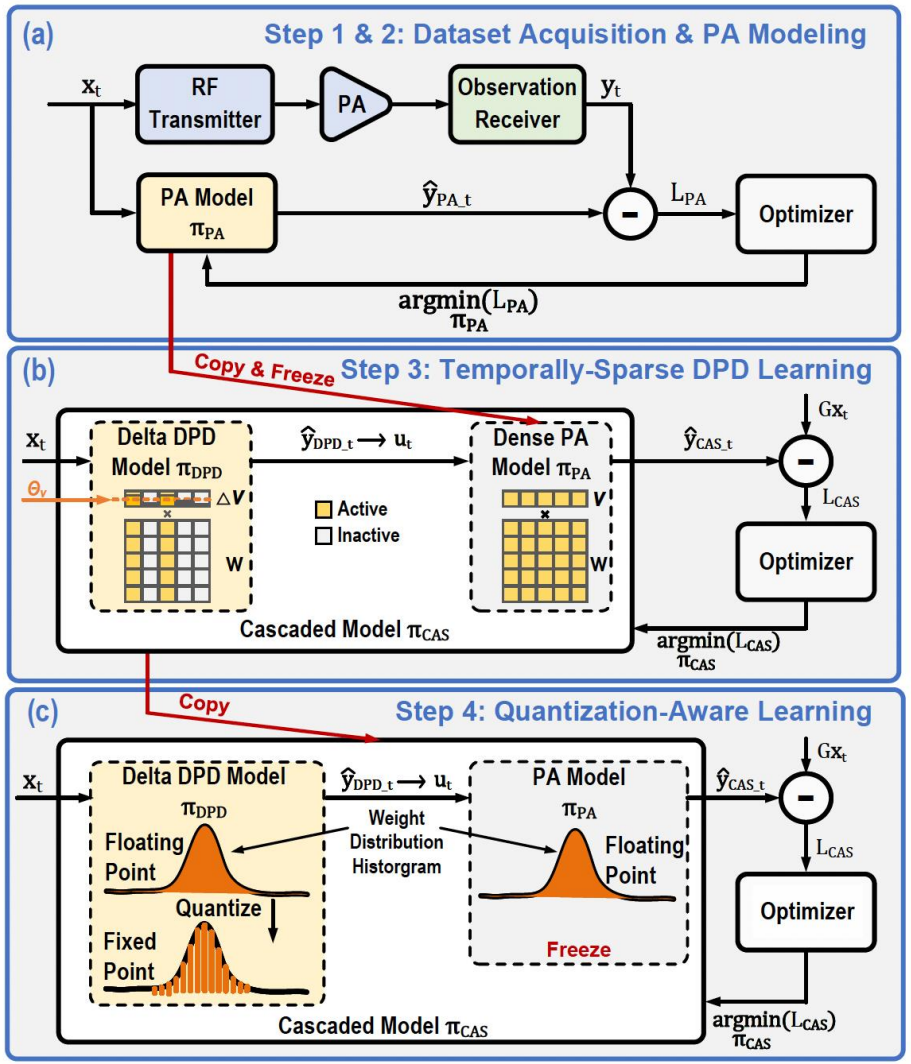
Model	Temporal Sparsity (%)	Active Param.	ACPR (dBc)	EVM
DeltaGRU	0	1084	-59.4	1.2%
	56	498	-52.8	1.4%
	76	290	-48.4	1.5%
FCRes-DeltaGRU	0	1090	-58.0	1.2%
	56	501	-51.4	1.2%
	76	290	-50.1	1.4%
TRes-DeltaGRU	0	996	-59.9	0.8%
	56	450	-52.9	0.9%
	72	288	-52.0	1.4%

[22] Y. Wu et al., arXiv, 2025.

OpenDPDv2: Sparsity + Quantization

TRes-DeltaGRU	Active Param.	Precision	ACPR (dBc)	EVM
Dense	1084	FP32	-59.9	0.8%
Dense	311	FP32	-52.3	2.0%
Sparse	288	FP32	-52.0	1.4%
		W16A16	-48.2	2.0%
		W12A12	-45.2	2.8%

- The sparse model with similar #active param. gives better EVM
- Adding quantization further reduces memory footprint
- 12-bit Weight + Activation keeps ACPR better than -45 dBc



OpenDPDv2 Linearization Performance

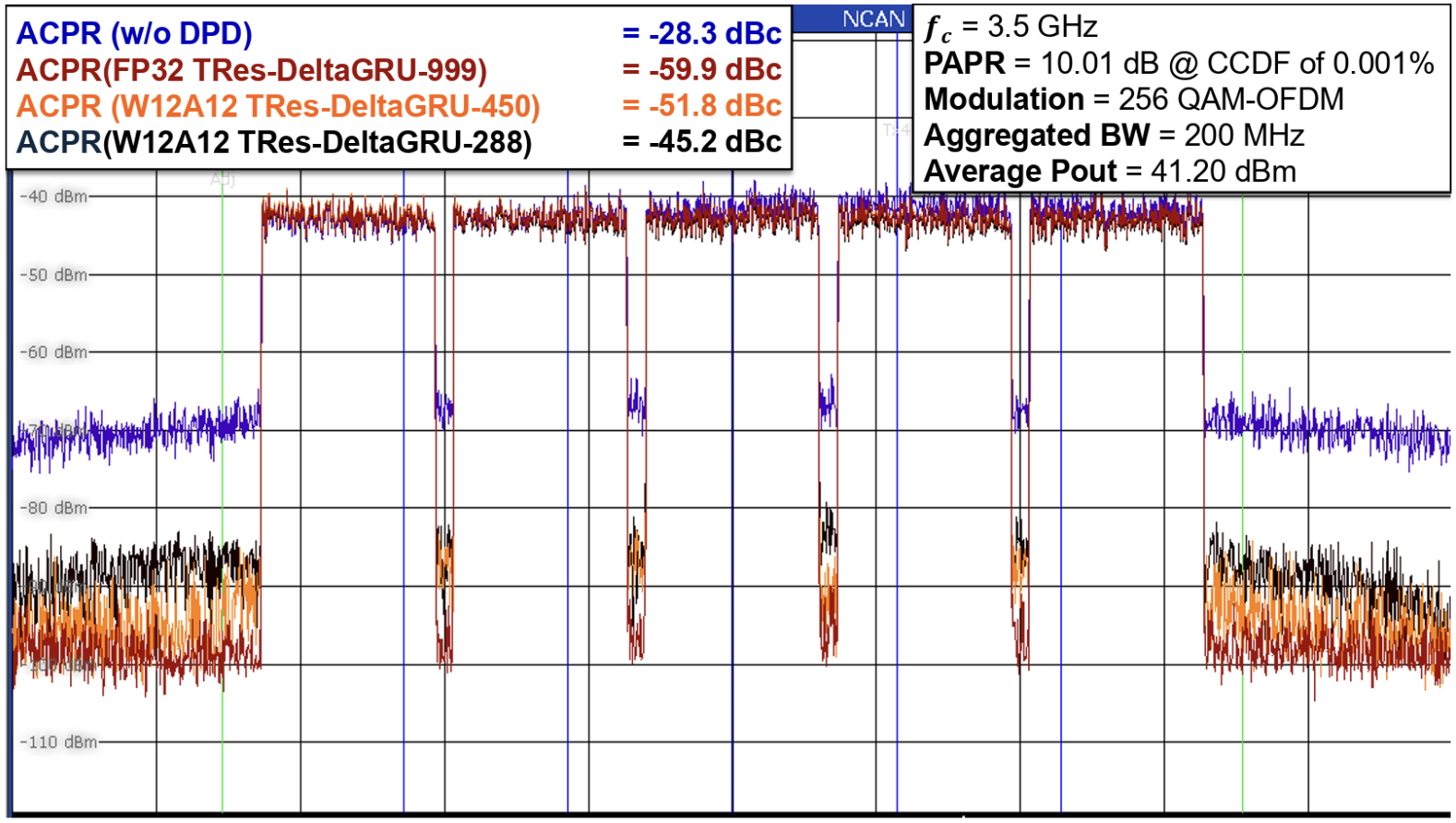
- TM3.1a 200-MHz 256-QAM OFDM @ PAPR = 10 dB
- Ampleon 3.5-GHz GaN Doherty PA @ P_{avg} = 41.2 dBm
- P1dB @ 46.5 dBm
- P3dB @ 50 dBm
- 60-20-20% train/val/test split with 98k samples
- 240 epochs

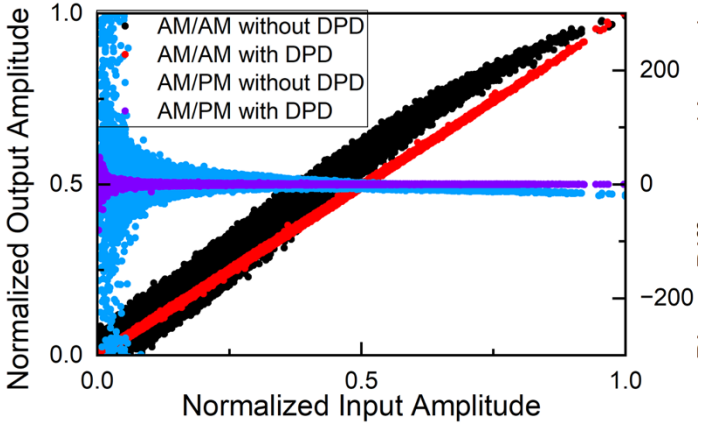
AI-DPD Models Trained by OpenDPDv2

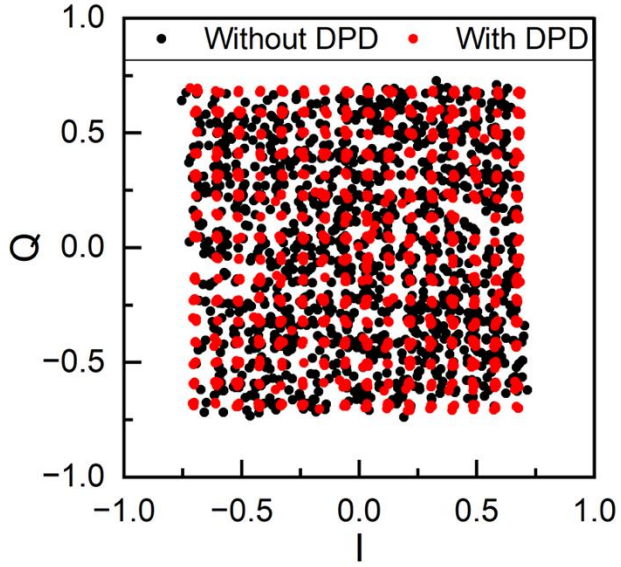
DPD Models	#Params	FLOPs	ACPR (dBc)	EVM
W/o DPD	-	-	-28.3	5.8%
RVTDCNN [18]	1007	1587	-50.8	2.0%
PG-JANET [13]	1130	1507	-58.2	1.1%
DVR-JANET [14]	1097	1370	-59.4	1.2%
BO-JANET [15]	1064	1535	-58.7	0.7%
APNRRU [23]	1043	1328	-58.6	1.3%
TRes-DeltaGRU [22]	999	1324	-59.9	0.8%

[22] Y. Wu et al., arXiv, 2025.

OpenDPDv2 Linearization Performance

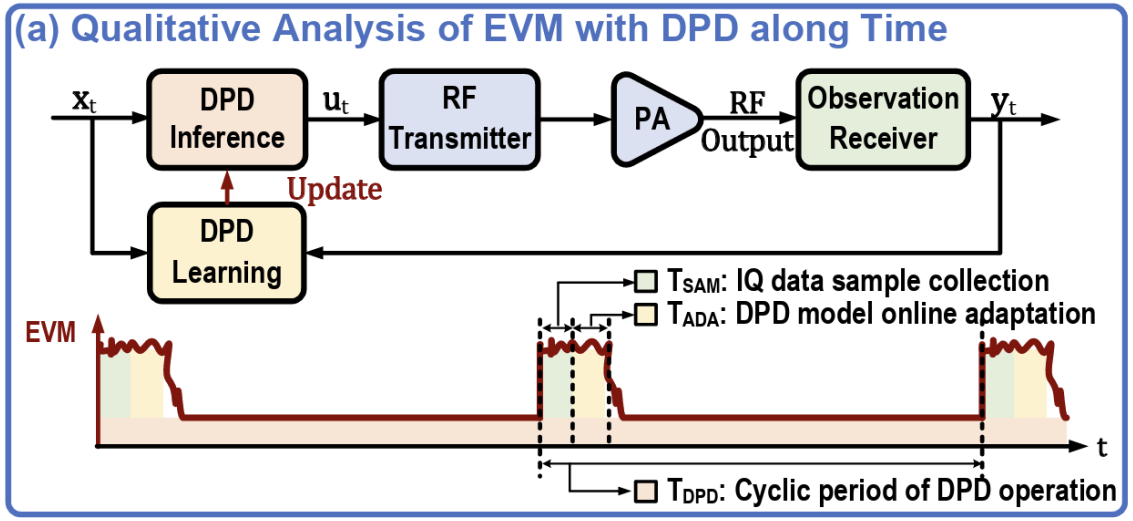


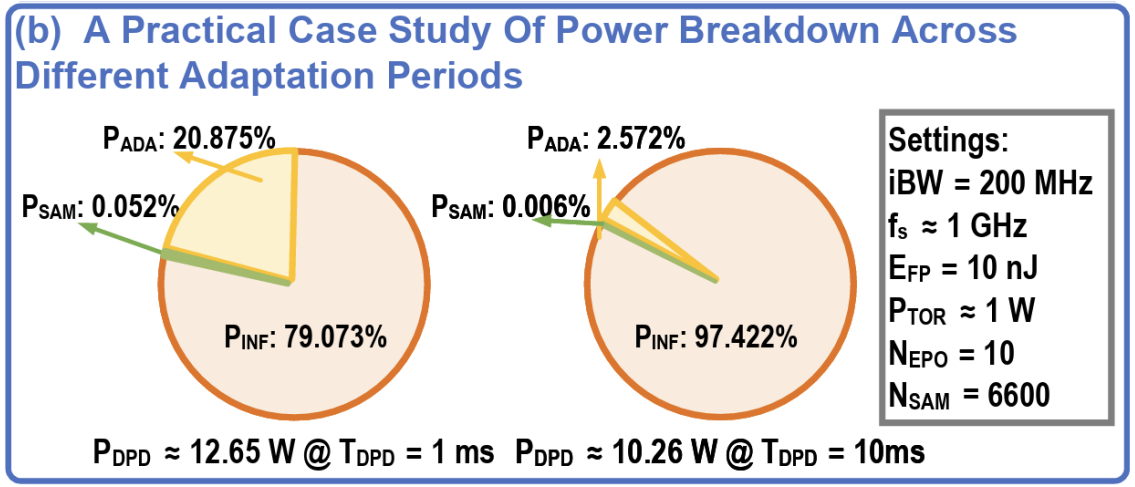




AI-DPD HW Design Challenge: Power

- AI-DPD power includes:
  - Inference Power: P_{INF}
  - Adaptation Power: P_{ADA}
- If including ORx Power:
  - $P_{DPD} = P_{INF} + P_{ADA} + P_{ORx}$
- Assumptions for power analysis:
  - E_{INF} = 10 nJ for 1000-param. GRU [17];
  - P_{ORx} ≈ 1W (12-bit, 1GSps) [24];
  - #epochs = 10, #samples = 6600 [25]
  - Adapt. Period: T_{DPD} from 1 ms to 1 s;
- **Good news:** ORx is not always-on
- **Takeaways:**
  - P_{INF} dominates when $T_{DPD} > 1$ ms
  - Start from inference opt. first





[24] Y. Wu et al., arXiv, 2025.

AI-DPD Energy Model

- Inference power is (derivation in [22]):
  - $P_{INF} = E_{INF} f_s$
  - f_s: Sample rate/Op. freq. of DPD
  - E_{INF}: Energy per fwd. pass of AI-DPD
- $E_{INF} = E_{\text{MUL}} + E_{\text{ADD}} + E_{\text{MEM}}$
  - Computations are cheap
  - Memory access is the highest wall
  - 12-bit access to 8 KB SRAM is 4~8x more expensive vs. 12-bit MAC [16].
- $P_{\text{HBM}} = 10\ \text{param.} \times 250\ \text{pJ} \times 1\ \text{GSps} = 2.5\ \text{W}$
- Takeways:
  - DRAM access is not affordable
  - DPD parameters should be buffered fully on-chip

Energy/Operation. 64-bit Mem. [26]

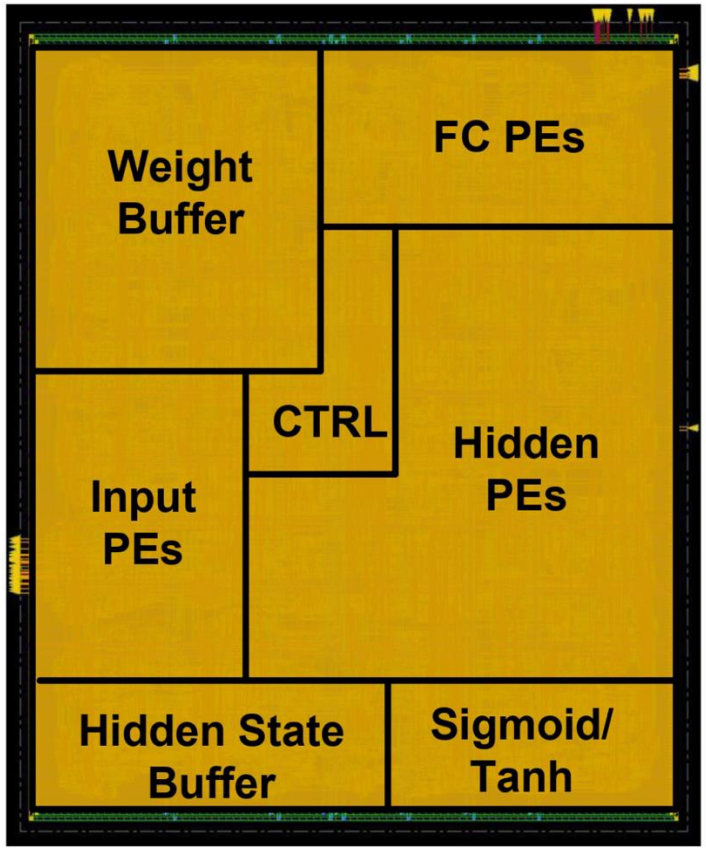
Operation		Picojoules per Operation		
		45 nm	7 nm	45 / 7
+	Int 8	0.03	0.007	4.3
	Int 32	0.1	0.03	3.3
	BFloat 16	--	0.11	--
	IEEE FP 16	0.4	0.16	2.5
	IEEE FP 32	0.9	0.38	2.4
×	Int 8	0.2	0.07	2.9
	Int 32	3.1	1.48	2.1
	BFloat 16	--	0.21	--
	IEEE FP 16	1.1	0.34	3.2
	IEEE FP 32	3.7	1.31	2.8
SRAM	8 KB SRAM	1	7	8
	32 KB SRAM	2	8	2.4
	1 MB SRAM[1]	10	14	7.1
GeoMean[1]		--	--	2.6
DRAM		Circa 45 nm	Circa 7 nm	
	DDR3/4	1300[2]	1300	1.0
	HBM2	--	250-450[2]	--
	GDDR6	--	350-480[2]	--

120×

3500×

An AI-DPD ASIC: DPD-NeuralEngine

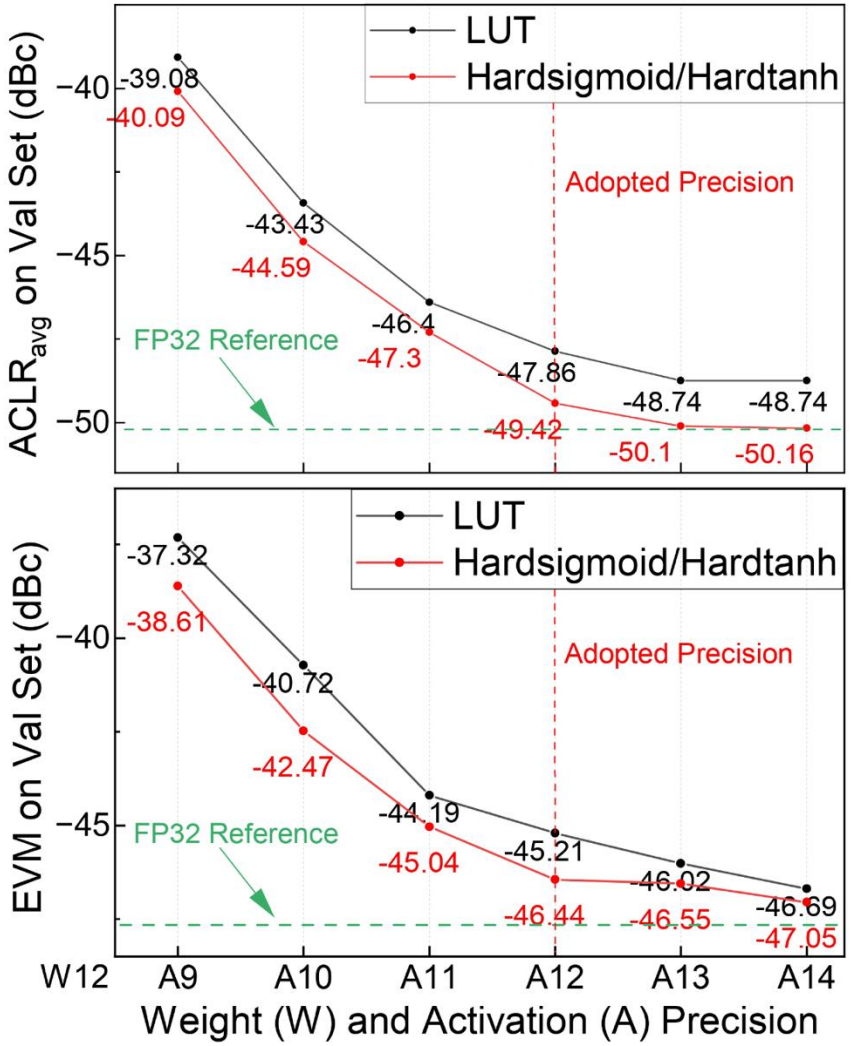
- GF 22FDX-PLUS PDK
- 500-param. On-chip GRU DPD model
- Capable of real-time DPD for up to 60 MHz I/Q @ 240 MHz sample rate



	Specifications		
Technology	GlobalFoundries 22nm (22FDX-PLUS)		
Die Area	0.4mm x 0.5mm		
Supply Voltage	0.9V		
Latency	7.5ns		
Clock Frequency	2GHz		
Power (TT, 25°C)	Dynamic	Leakage	Total
	186.57mW	8.15mW	194.72mW
Power-Area Efficiency	6.6 TOPS/W/mm²		

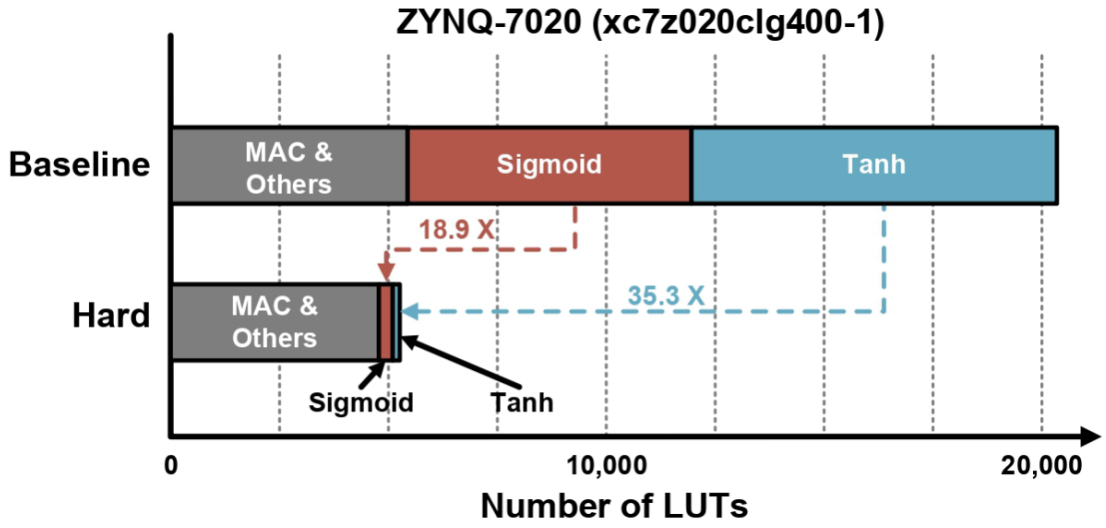
[27] A. Li et al., ISCAS, 2024.

Nonlinear Function Implementation



FPGA Emulation-based Resource Estimation

	LUT	FF	DSP	BRAM
Available	53200	106400	220	140
Used (LUT-Sig./Tanh)	20522 (38.7%)	3969 (3.7%)	85 (38.6%)	0 (0%)
Used (Hard-Sig./Tanh)	5439 (10.2%)	3156 (3.0%)	95 (43.2%)	0 (0%)



Towards a General AI Correction Engine

RF Transmitter
- IQ modulator, upconversion mixer, driver, DAC, PA
- LO leakage, DC offsets, I/Q imbalance, memory effect

Clocking & Synthesis
- Fractional-N PLL, DTC
- Spur mitigation, phase interpolator nonlinearity, timing errors

RF Receiver
- LNA/mixer/baseband chain
- Compressive distortion, bias-dependent gain, I/Q imbalance, memory effects

DAC
- Static/dynamic nonlinearity, bandwidth-dependent distortion

AI-based Correction Engine
- Data-driven Analog Modelling
- AI Calibration Model Training
- Cost model + HW implementation

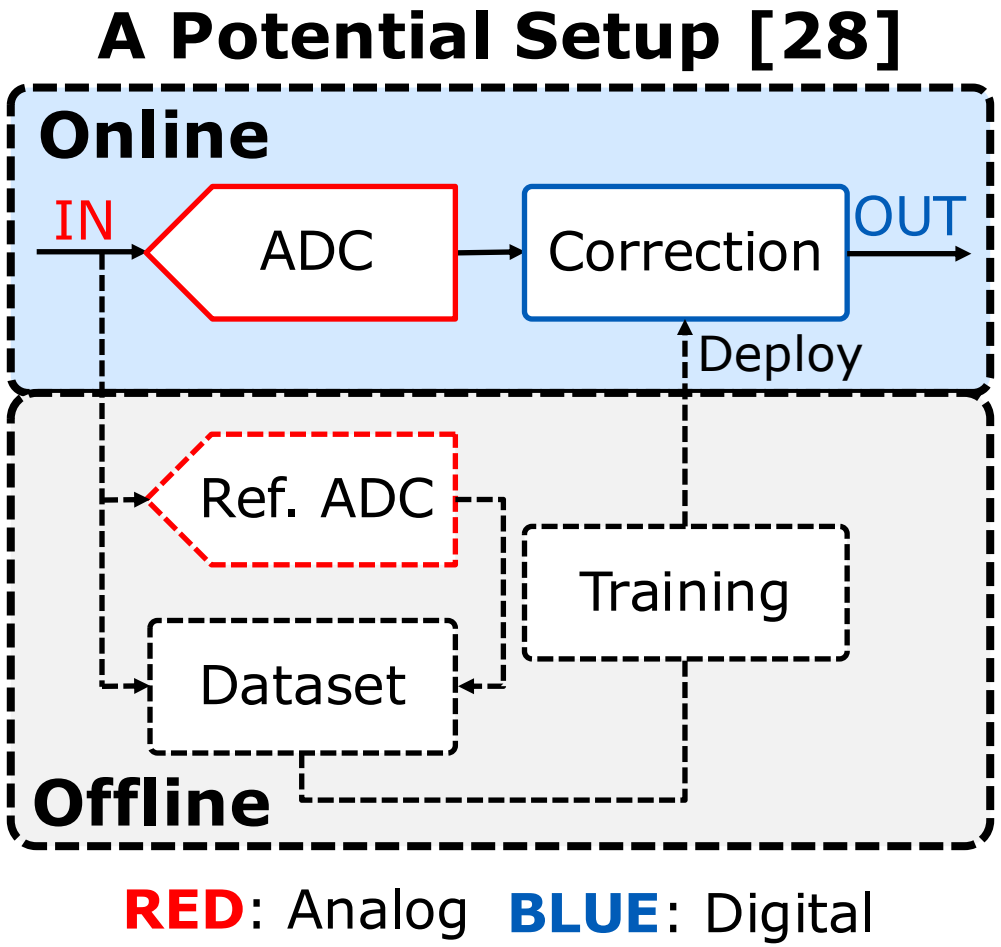
ADC
- NL/DNL, dynamic distortion, TI-ADC timing mismatch, spur cleanup

Power Management
- Digitally-assisted Energy harvesters, DC-DC, LDOs
- MPPT, dead-time, nonlinearity, operation point

Analog AI Computing
- ReRAM/PCM crossbars, neuromorphic cores
- device mismatch, drift, nonlinear I–V, peripheral non-idealities

Sensor/Analog Feat. Ext.
- Audio, biomed, optical AFEs
- Drift, mismatch, nonlinearity, memory effect

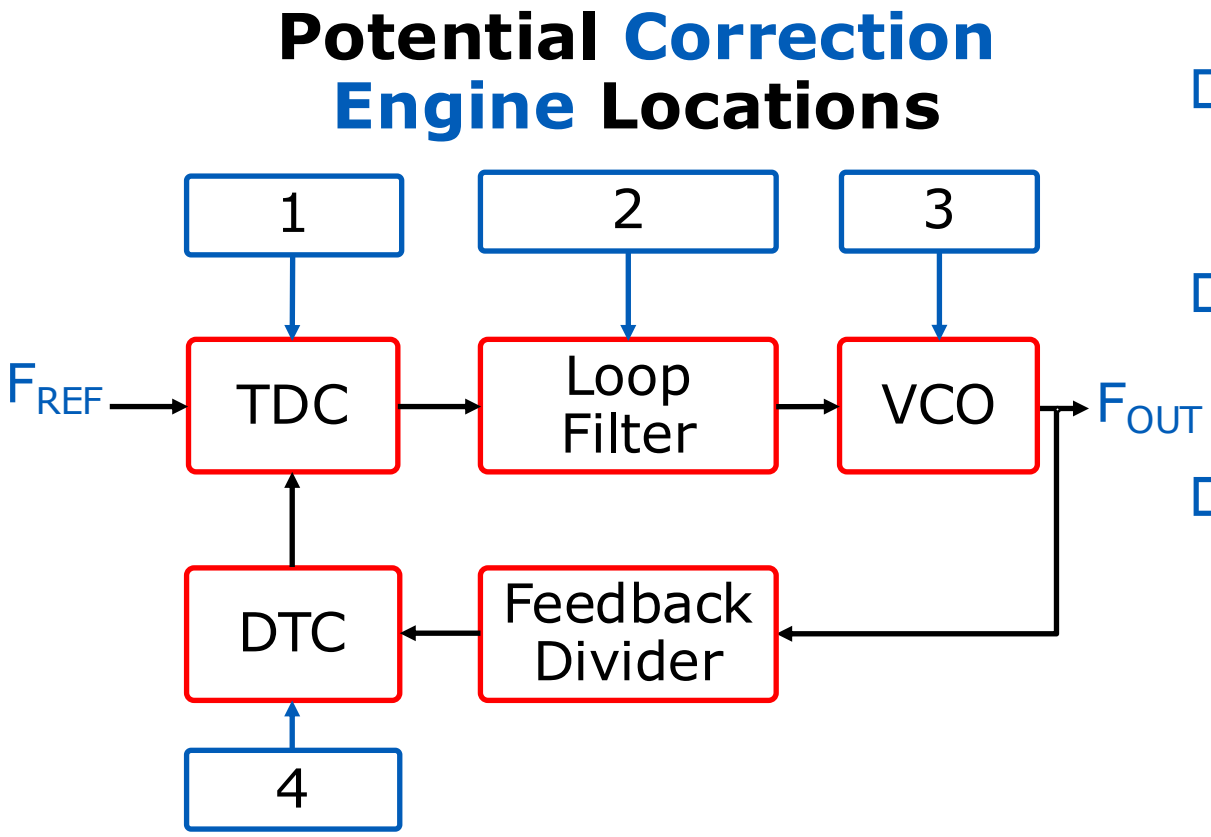
Potential 1: ADC Post-Correction



- **To fix:** INL/DNL, dynamic distortion, bandwidth limits, and TI-ADC skew/mismatch, etc.
- **Open Questions:**
  - How to guarantee the Ref. ADC is always better across PVT, aging?
  - What if signal changes (bandwidth, modulation, PAPR)?
  - What #compute and size is allowed for the calibration engine?
  - Is online adaptation even possible?

[28] R. Van Veldhoven, U.S. Patent, 2023.

Example 2: DPLL Calibration



- **To fix:** fractional spurs & jitter from TDC/DTC nonlinearity, PI nonlinearity, and drift (PVT/aging).
- **Correction Locations:**
  - Different insertion points trade observability, stability, and compute.
- **Open Questions:**
  - What can we measure on-chip to learn the correction?
  - Which insertion point (or hybrid points) offers the best performance?
  - How tiny can the correction engine be?

Conclusion & Outlook

- **AI-DPDs achieve state-of-the-art wideband PA linearization @ 200 MHz**
  - More parameter efficient than classic Volterra PAs.
  - Parameter-efficiency: Linearization performance per model parameter
- **AI model compression methods also work for AI-DPDs**:
  - **Quantization:** INT16 gives ~2.8× power reduction without losing linearization.
  - **Temporal sparsity:** ~52% fewer active parameters while keeping strong ACPR in measurement.
- **Hardware takeaway**: Memory access dominates, so keep DPD parameters on-chip and use model compression to shrink the memory footprint.
- **Reproducibility Gap:** Being open-source will facilitate reproduction & fairer benchmarking
- **Next Step:**
  - Concrete efficiency gain and power saving measurement.
  - Scale the framework beyond PAs to a general AI correction engine.

References

- [1] Nokia Bell Labs, Extreme massive MIMO for macro cell capacity boost in 5G-Advanced and 6G, 2022.
- [2] Ericsson, Energy Performance of 6G Radio Access Networks: A once in a decade opportunity, 2024.
- [3] Texas Instruments, GC5325 Wideband Digital Predistortion Transmit Processor, Datasheet SLWS215, Jan. 2009.
- [4] ESTI, Sustainable power feeding solutions for 5G network, ETSI ES 203 700 V1.1.1, Dec. 2020.
- [5] Huawei, 5G Power White Paper, 2019.
- [6] P. N. Landin, S. Gustafsson, C. Fager, and T. Eriksson, " Weblab: A webbased setup for PA digital predistortion and characterization [applicationnotes], " IEEE Microw. Mag., vol. 16, no. 1, pp. 138–140, Feb. 2015.
- [7] Y. Wu, G. D. 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," ISCAS, 2024, pp. 1-5
- [8] Taijun Liu, S. Boumaiza and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," in IEEE TMTT, vol. 52, no. 3, pp. 1025-1033, March 2004
- [9] R. Hongyo, Y. Egashira, T. M. Hone and K. Yamaguchi, "Deep Neural Network-Based Digital Predistorter for Doherty Power Amplifiers," in IEEE MWCL, vol. 29, no. 2, pp. 146-148, Feb. 2019
- [10] C. Jiang et al., "Block-Oriented Time-Delay Neural Network Behavioral Model for Digital Predistortion of RF Power Amplifiers," in IEEE TMTT, vol. 70, no. 3, pp. 1461-1473, March 2022
- [11] Y. Wu, U. Gustavsson, A. G. i. Amat and H. Wymeersch, "Residual Neural Networks for Digital Predistortion," IEEE GlobeCom, 2020, pp. 01-06

References

- [12] 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," in IEEE Access, vol. 8, pp. 63780-63789, 2020
- [13] T. Kobal, Y. Li, X. Wang and A. Zhu, "Digital Predistortion of RF Power Amplifiers With Phase-Gated Recurrent Neural Networks," in IEEE TMTT, vol. 70, no. 6, pp. 3291-3299, June 2022
- [14] T. Kobal and A. Zhu, "Digital Predistortion of RF Power Amplifiers With Decomposed Vector Rotation-Based Recurrent Neural Networks," in IEEE Transactions on Microwave Theory and Techniques, vol. 70, no. 11, pp. 4900-4909, Nov. 2022
- [15] Q. Zhang, C. Jiang, G. Yang, R. Han and F. Liu, "Block-Oriented Recurrent Neural Network for Digital Predistortion of RF Power Amplifiers," IEEE TMTT, vol. 72, no. 7, pp. 3875-3885, July 2024
- [16] Y. Wu et al., "MP-DPD: Low-Complexity Mixed-Precision Neural Networks for Energy-Efficient Digital Predistortion of Wideband Power Amplifiers," in IEEE MWTL, vol. 34, no. 6, pp. 817-820, June 2024
- [17] Y. Wu et al., "DeltaDPD: Exploiting Dynamic Temporal Sparsity in Recurrent Neural Networks for Energy-Efficient Wideband Digital Predistortion," in IEEE MWTL, vol. 35, no. 6, pp. 772-775, June 2025
- [18] X. Hu et al., "Convolutional Neural Network for Behavioral Modeling and Predistortion of Wideband Power Amplifiers," in IEEE TNNLS, vol. 33, no. 8, pp. 3923-3937, Aug. 2022
- [19] P. Jaraut et al., "Augmented Convolutional Neural Network for Behavioral Modeling and Digital Predistortion of Concurrent Multiband Power Amplifiers," in IEEE TMTT, vol. 69, no. 9, pp. 4142-4156, Sept. 2021
- [20] M. H. Khazani, A. Motaqi, A. Dalbah, M. Helaoui, W. Chen and F. M. Ghannouchi, "A CNN-Based Digital Predistortion Method for Multi-Beam Phased-Array Transmitters," in IEEE Wireless Communications Letters, 2026

References

- [21] H. Duan, M. Versluis, Q. Chen, L. C. N. de Vreede and C. Gao, "TCN-DPD: Parameter-Efficient Temporal Convolutional Networks for Wideband Digital Predistortion," 2025 IEEE/MTT-S IMS, 2025, pp. 1103-1106
- [22] Yang et al., "Digital Predistortion of Quadrature Digital Power Amplifiers Using RVRTCNN: Real-Valued Residual Temporal Convolutional Neural Network," in IEEE Communications Letters, vol. 29, no. 9, pp. 2028-2032, Sept. 2025
- [22] Y. Wu, A. Li, and C. Gao, 'OpenDPDv2: A Unified Learning and Optimization Framework for Neural Network Digital Predistortion', Dec. 16, 2025, arXiv: arXiv:2507.06849
- [23] A. Fischer-Bühner, L. Anttila, M. Turunen, M. Dev Gomony and M. Valkama, "Augmented Phase-Normalized Recurrent Neural Network for RF Power Amplifier Linearization," in IEEE TMTT, vol. 73, no. 1, pp. 412-422, Jan. 2025
- [24] Texas Instruments, "ADC12SJ1600 12-Bit, 1.6-GSPS, RF-Sampling Analog-to-Digital Converter (ADC)," https://www.ti.com/lit/ds/symlink/adc12sj1600.pdf?ts=1751638711567&ref_url=https%253A%252F%252Fwww.ti.com%252Fdata-converters%252Fadc-circuit%252Fproducts.html, accessed: Jul. 4, 2025.
- [25] Y. Li, X. Wang and A. Zhu, "Complexity-Reduced Model Adaptation for Digital Predistortion of RF Power Amplifiers With Pretraining-Based Feature Extraction," in IEEE TMTT, vol. 69, no. 3, pp. 1780-1790, March 2021
- [26] N. P. Jouppi et al., "Ten Lessons From Three Generations Shaped Google's TPUv4i: Industrial Product," ISCA, 2021
- [27] A. Li, H. Wu, Y. Wu, Q. Chen, L. C. N. de Vreede and C. Gao, "DPD-NeuralEngine: A 22-nm 6.6-TOPS/W/mm2 Recurrent Neural Network Accelerator for Wideband Power Amplifier Digital Pre-Distortion," ISCAS, 2025
- [28] R. Van Veldhoven, "Correction of sigma-delta analog-to-digital converters (ADCs) using neural networks," U.S. Patent 11,722,146 B1, Aug. 8, 2023.