TEXAS A&M UNIVERSITY
Engineering

A Neural Network Approach for Calibrating Memristors Crossbars

Mohammad Rezaeifar, He Chaoyi, Kamran Entesari, Linda Katehi

MemSys2025-11th International Symposium on Memory Systems
October 7-8,2025
Washington D.C. , USA

TEXAS A&M UNIVERSITY
Department of Electrical
& Computer Engineering

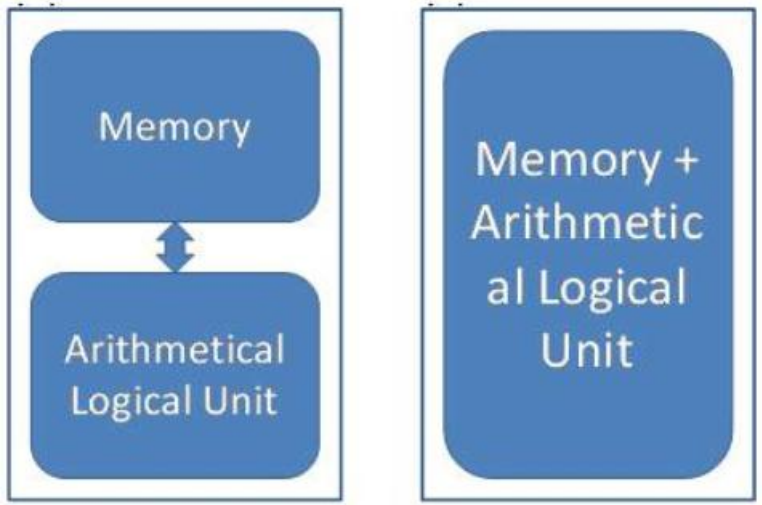
iEMSL
INTELLIGENT ELECTROMAGNETIC
SENSOR LABORATORIES

Title

- **A Neural Network Approach for Calibrating Memristor Crossbars**
- Mohammad Rezaeifar, He Chaoyi, Kamran Entesari, Linda Katehi
- Texas A&M University
- **MemSys 2025 – Washington, VA**

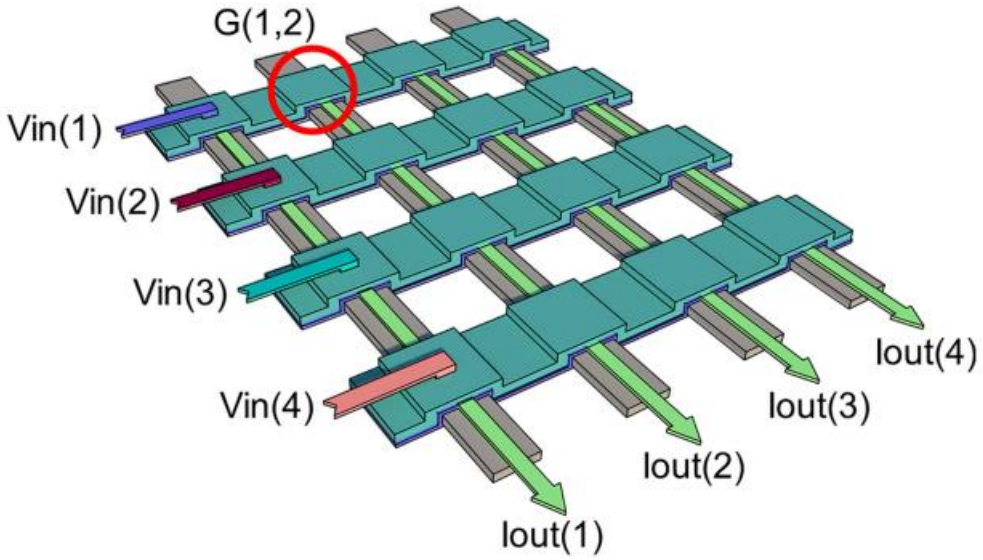
Motivation

- Von Neumann bottleneck → costly data movement
- In-Memory Computing (IMC) with memristors = promising solution
- BUT: crossbars suffer IR-drop, parasitic effects, non-linearities
- Accuracy degradation in analog MVM

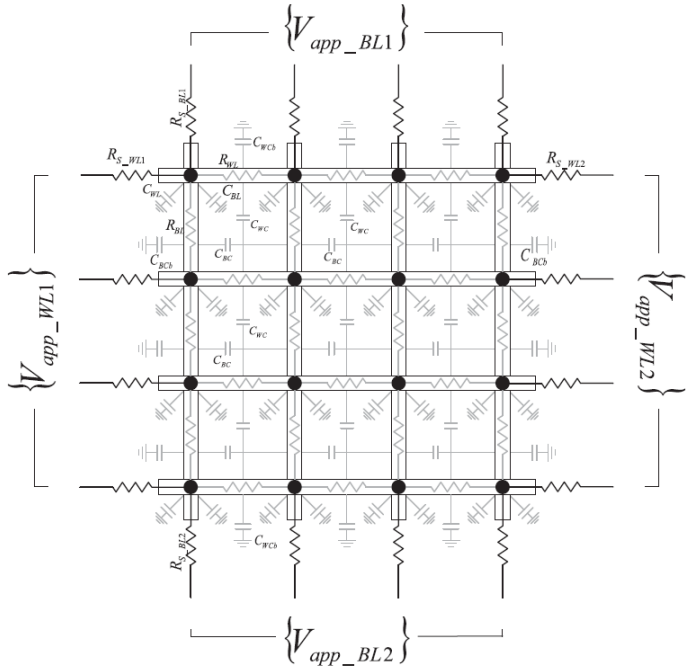


In memory & Von-Neumann computing

Problem Statement



Crossbar Memristor



Model of a Crossbar Memristor

Contributions

- **Post-fabrication calibration** (no hardware change)
- **Neural network framework**
- **accuracy improvement** (MSE reduction)

Crossbar Basics

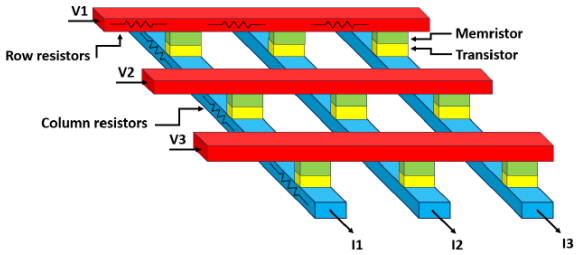


Figure 1: A memristor crossbar array structure.

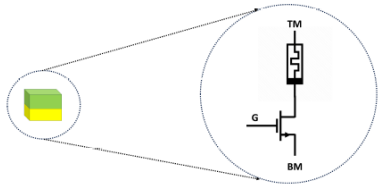


Figure 2: A memristor cell architecture integrating o sistor (1T) and one resistor (1R).

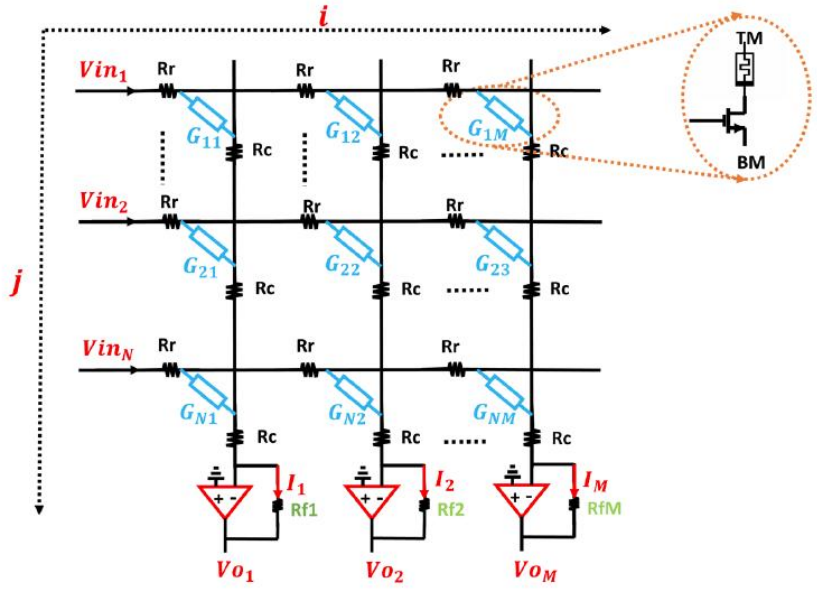
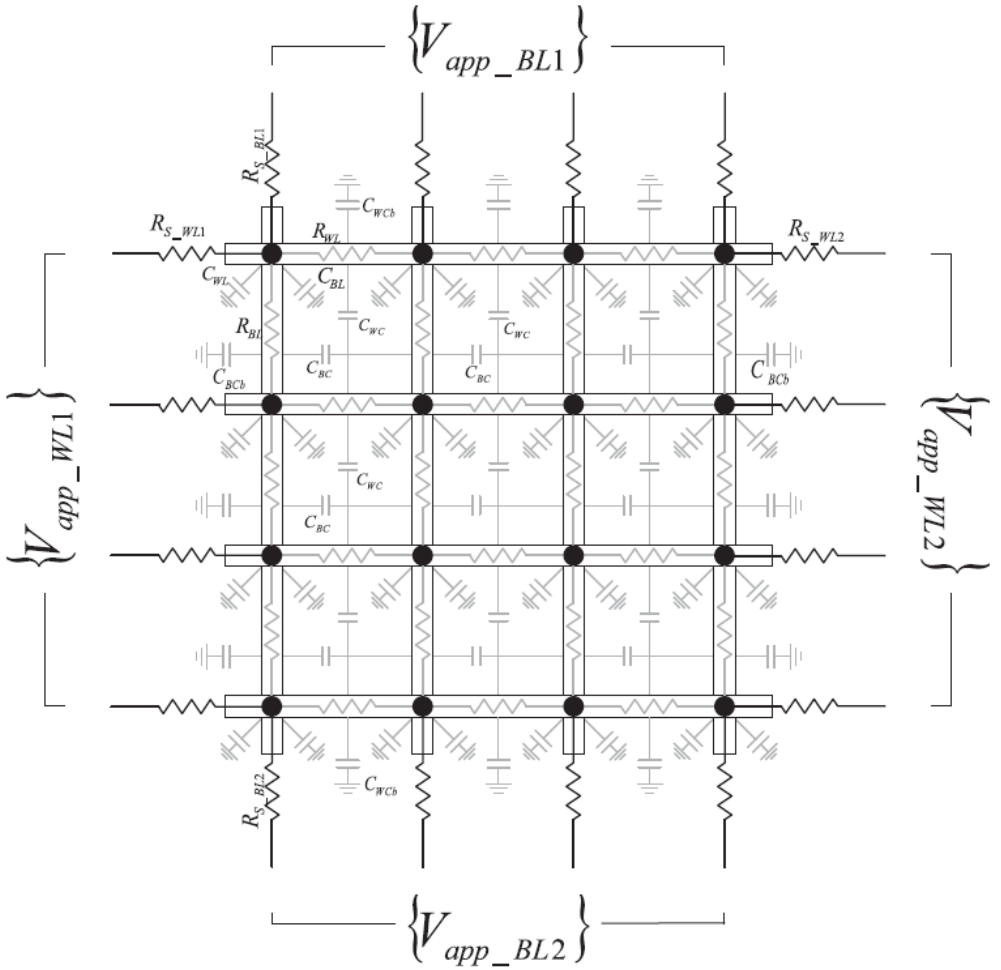


Figure 3: Memristor crossbar performing MVM. Conductance matrix G (size $N \times M$) is multiplied by voltage input vector V_{in} (size $1 \times N$).

Sources of Errors

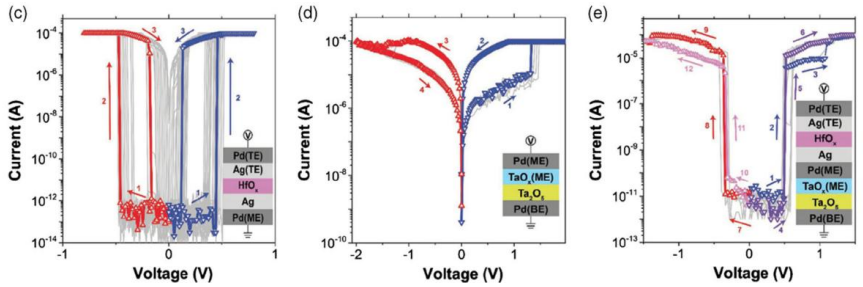
- **Wire resistance** → IR-drop
- **Capacitance/inductance** → distortions
- **Device variability** → drift, noise
- Example: outputs deviate significantly from ideal MVM



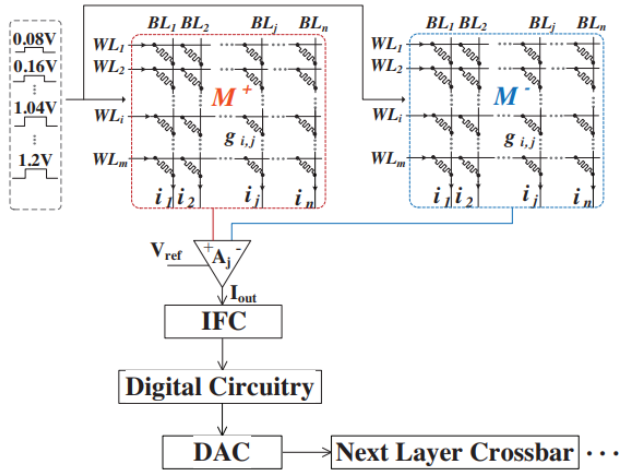
Model of a Crossbar Memristor

Related Work

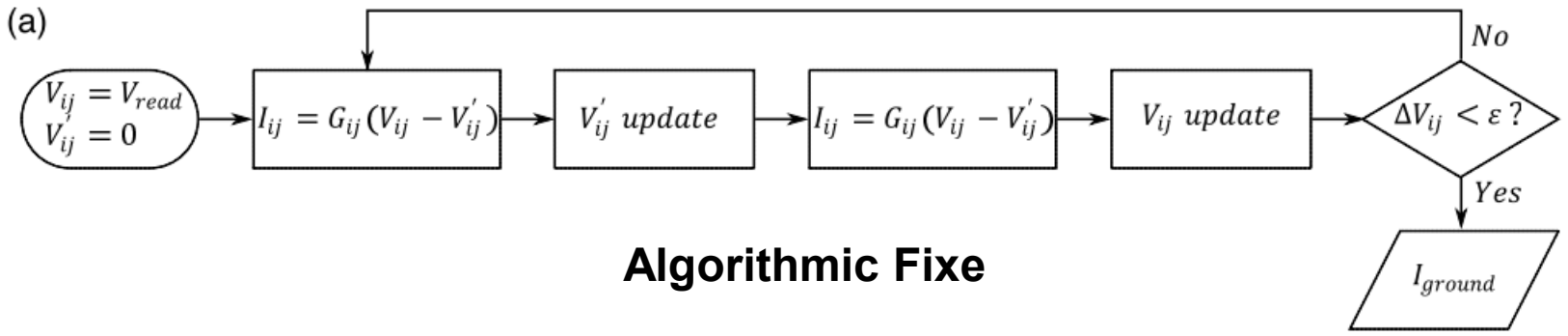
- **Hardware fixes** → bigger area, higher complexity
- **Circuit-level** → error correction, compensation
- **Algorithmic** → retraining with hardware-aware models
- Still: incomplete + trade-offs



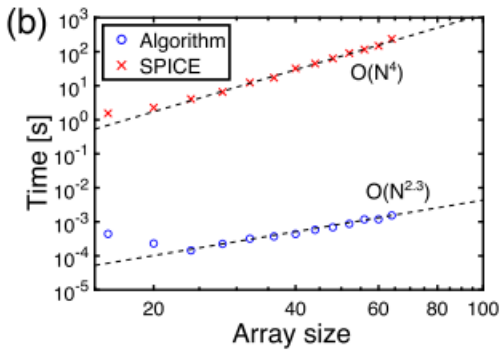
Hardware Fixe



Circuit-level Fixe



Algorithmic Fixe



Our Approach

- Learn **systematic distortions** with NN
- **Input**: Ideal conductance matrix (encode latent vector) , Real measured currents
- **Output**: corrected, near-ideal currents

$$I_{\text{real}} = \mathcal{F}(M, V_{\text{in}}) + \varepsilon \tag{2}$$

$$\begin{aligned} \hat{I}_{\text{out}} &= \mathcal{G}(I_{\text{real}}, M) \equiv \mathcal{G}_\theta(I_{\text{real}}, M) = \tilde{I}_{\text{out}} \\ &\rightarrow \max_\theta p_\theta(\hat{I}_{\text{out}} | I_{\text{real}}, M) \\ &= \max_\theta p_\theta(\hat{I}_{\text{out}} | \mathcal{F}(M, V_{\text{in}}) + \epsilon, M) \end{aligned} \tag{3}$$

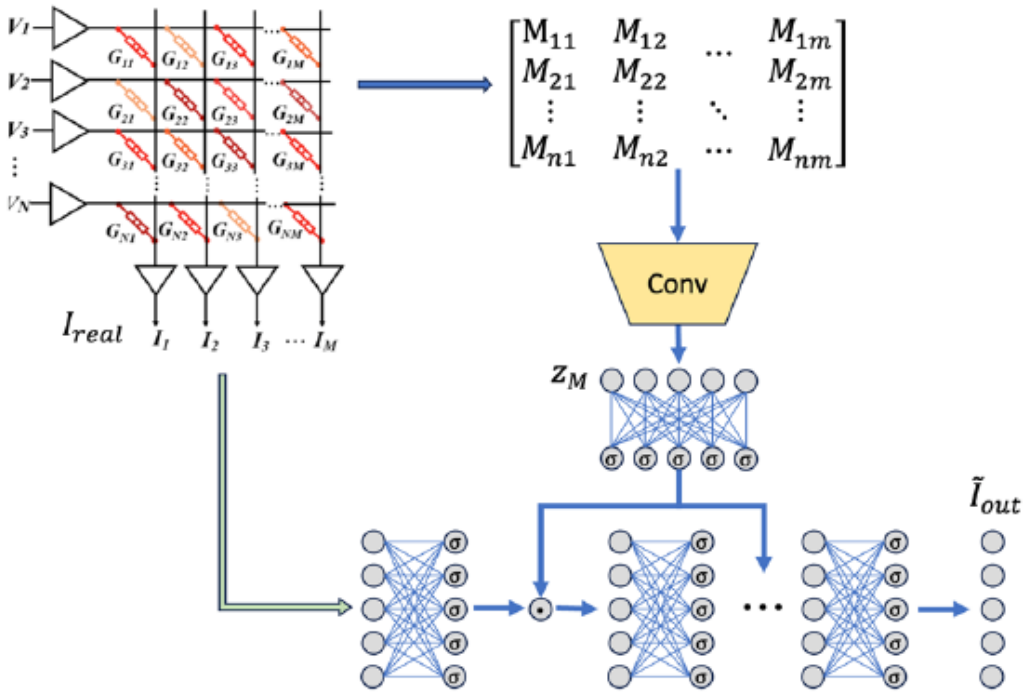


Figure 4: Model Structure

$$\begin{aligned} O_{i+1} &= CS(O_i, z_m) \\ &= (W_1 O_i + b_1) \odot \sigma(W_2 z_m + b_2) + W_3 z_m, \\ i &= 0, 1, 2, \ldots, N-1, \end{aligned} \tag{4}$$

Framework

- Step 1: Encode conductance matrix → latent vector Z_m
- Step 2: Used with measured output I_{real}
- Step 3: Neural net calibration layers → I_{out}
- Activation = sigmoid

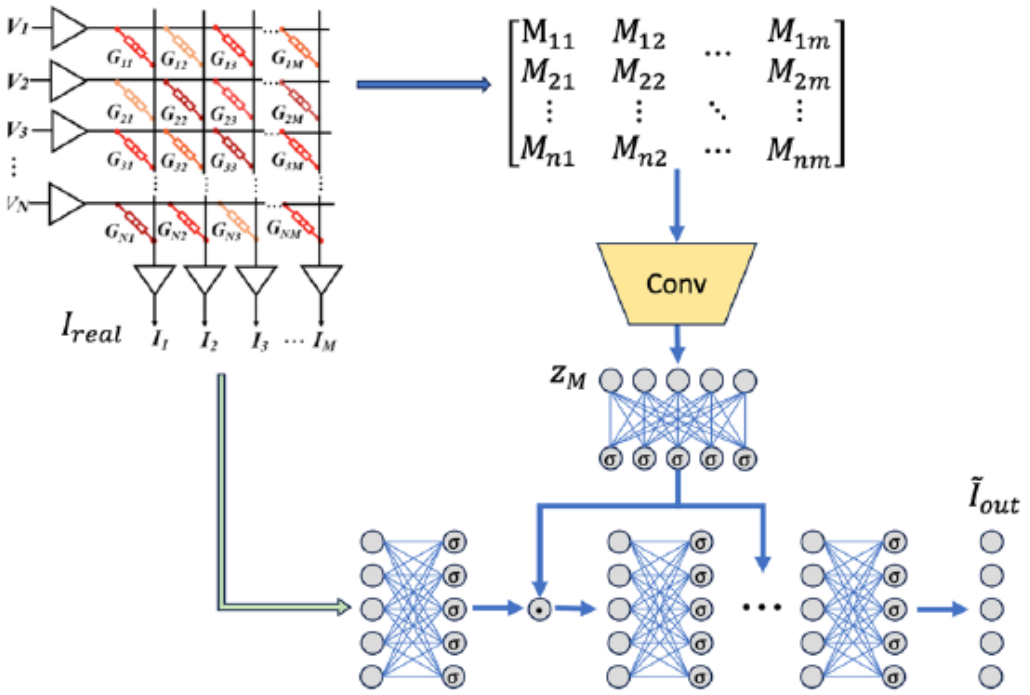
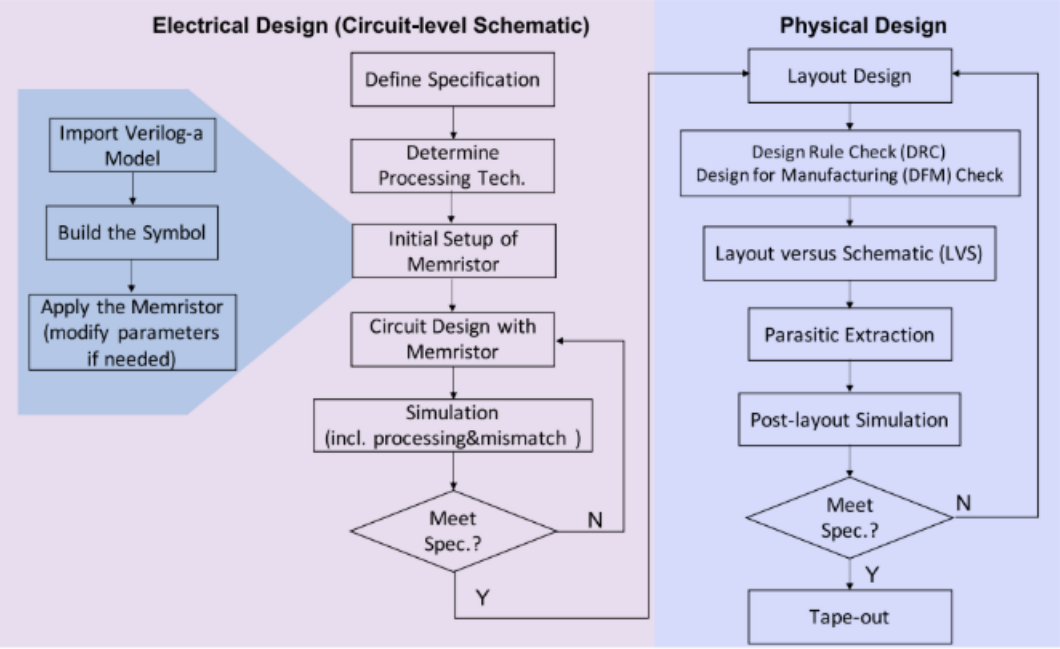


Figure 4: Model Structure

Data Collection

- Custom Verilog-A memristor model in Cadence Virtuoso
- Integrated with **22nm FD-SOI CMOS**
- Dataset: 1,000 conductance matrices (16.6μS–100μS), 100 input vectors each (0.1–0.5V), → **100,000 VMM outputs**




```
module analytical (p, n);
        inout p, n;
electrical  p, n;
        parameter real Ap = 0.12340, An = -0.33000;
        parameter real tp = 2.74111, tn = 2.59685;
        parameter real rp0 = -40928.13784, rp1 = 55117.97865;
        parameter real rn0 = 41366.35820, rn1 = 7789.66771;
        parameter real Rinit = 40000;
        parameter real eta = 1;
        parameter real ap=0.225, bp=4.12;
        parameter real an=0.2801, bn=4.10;
        real Rmp, Rmn, svp, svn, vin, RS, IVp, IVn, IV;
real first_iteration,  R0_last, dt, it;
analog function integer stp;
        real arg; input arg;
        stp = (arg >= 0 ? 1 : 0 );
endfunction
analog begin
        if (first_iteration==0)  begin
        it=0;          R0_last=Rinit;
        end
        dt=$abstime-it;
        vin=V(p,n);
        Rmp=rp0+rp1*vin;     Rmn=rn0+rn1*vin;
        if (vin>0) begin
        svp=Ap*(-1+exp(abs(vin)/tp));
        RS=(R0_last+svp*Rmp*(Rmp-R0_last)*dt)/(1+svp*(Rmp-
        R0_last)*dt);
        end
        else begin
        svn=An*(-1+exp(abs(vin)/tn));
        RS=(R0_last+svn*Rmn*(Rmn-   R0_last)*dt)/(1+svn*(Rmn-
        R0_last)*dt);
        end
        if (RS>=Rmp && vin>0)                    RS=R0_last;
        if (RS<=Rmn && vin<0)                    RS=R0_last;
        if (abs(vin)<=0.5)                       RS=R0_last;
        IVp=ap*(1/RS)*sinh(bp*vin);
        IVn=an*(1/RS)*sinh(bn*vin);
        IV=IVp*stp(vin)+IVn*stp(-vin);
        I(p, n)<+ IV;
        R0_last=RS;
        first_iteration=1;
        it=$abstime;
end
endmodule
```

Neural Network Model

- Non-linear inverse operatorG_θ
- Compensate: IR-drop, noise,device non- linearity
- Lightweight inference (offline training, fast runtime)
- Adaptable to drift/ageing with small calibration set

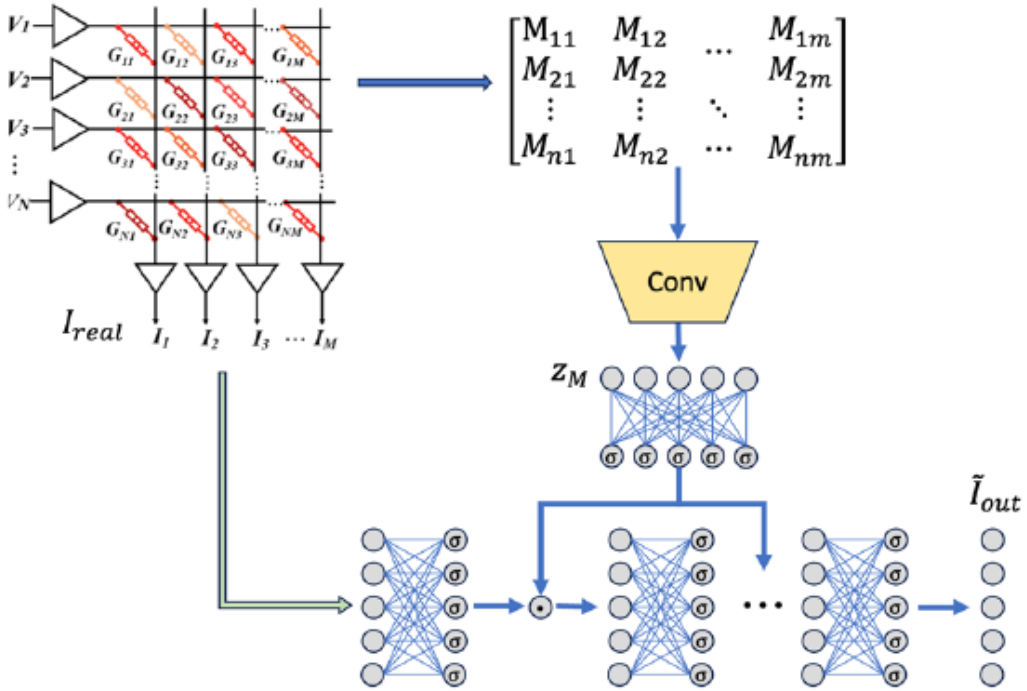


Figure 4: Model Structure

Training Setup

- Training: random input vectors + random matrices
- Validation: sinusoidal input, FIR filter config

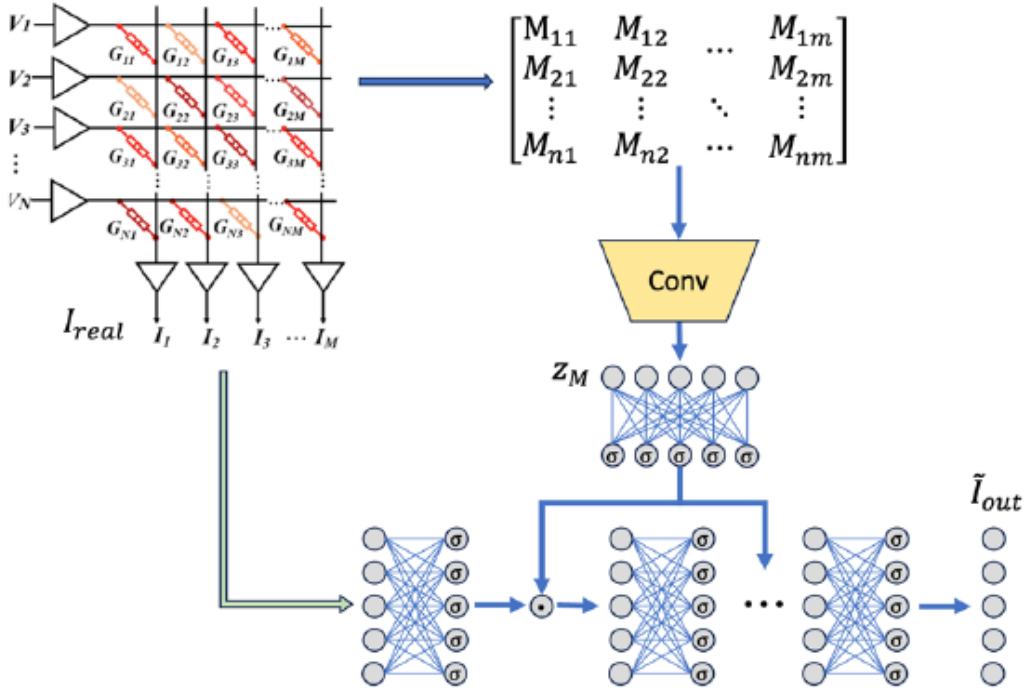


Figure 4: Model Structure

Results (Training/Validation)

- MSE reduction ~1000× : Before calibration: 0.13 , After calibration: 1.4e-4
- Strong generalization to unseen signals
- Works with sinusoidal + DC inputs

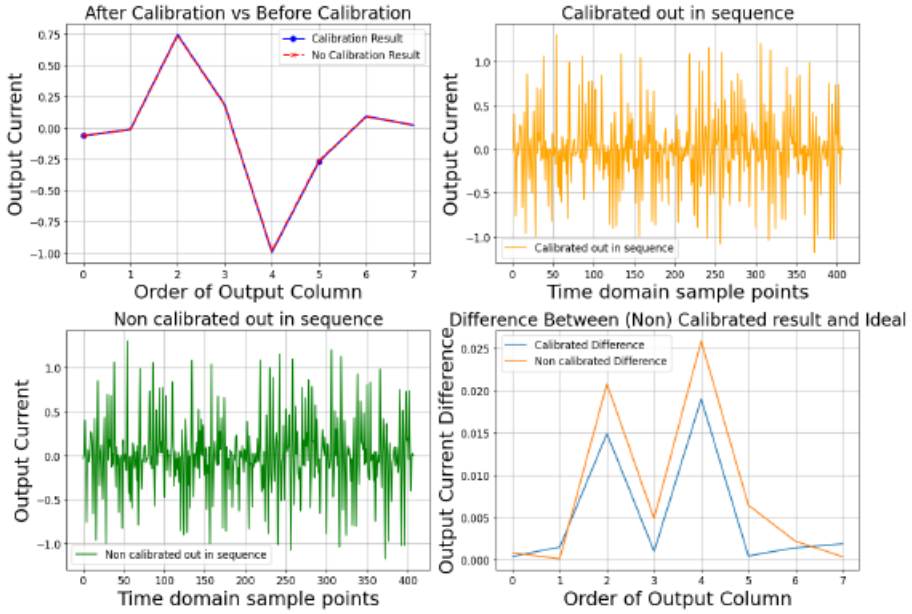


Figure 6: Visualization of the calibration results compared with crossbar original output.

Table 1: Training and Testing MSE Performance for Different Input Type

Input Data Type	$(\tilde{I}_{out} - I_{ideal})^2)$	$(I_{real} - I_{ideal})^2)$
Training with Random	0.000139	0.131878
Validating with Random	0.000149	0.122114
Testing with Sinusoidal	0.003398	0.032687
Testing with DC Input	0.003185	0.031794

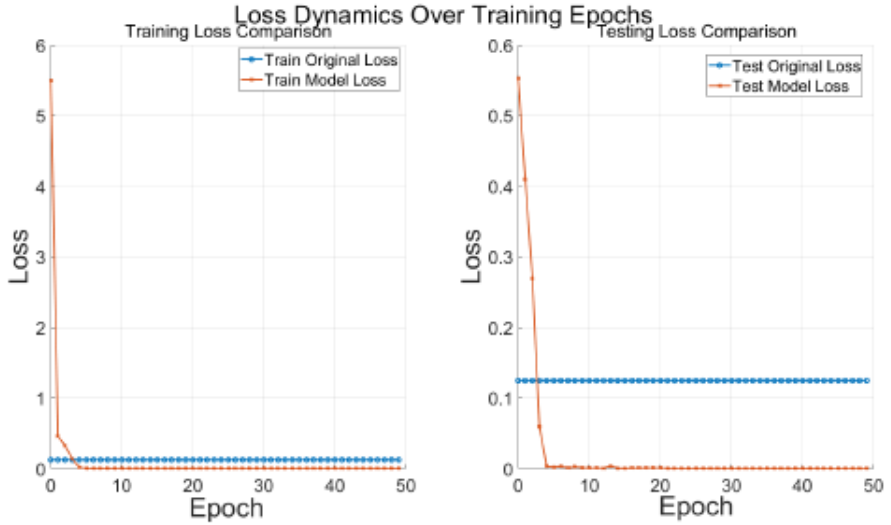


Figure 5: Training and Validating results

Robustness

- Handles parasitic distortions + noise
- No frequent retraining needed
- Low overhead → only inference at runtime

Comparison

- Device-level: area + complexity cost
- Circuit-level: partial corrections only
- Our method: post-fab, lightweight, scalable

Future Work

- Fabrication + Silicon validation Under noise, drift, temp.
- Extend to **larger/tiled arrays** (hierarchical calibration)
- Conditional fine-tuning for ageing effects
- Public release: Verilog-A

Conclusion

- Neural network framework → restores memristor accuracy
- Corrects distortions without hardware changes
- Enables practical, high-precision IMC systems
- Key step toward reliable neuromorphic computing

Acknowledgments / Q&A

- Intelligent Electromagnetic Sensor Lab (IEMSL), Texas A&M
- Collaborators & advisors
- Thank you