KEYSIGHT

# Unlocking the Potential of AI/ML in EDA: Recent Use-Cases and Techniques

Majid Ahadi Dolatsara, Zijian Song, Alexander Petr
Mohit Khanna, Jianjun Xu, Marijana Krivic, Fei-Jiang Hu

Keysight Technologies

October 5th, 2023

# Agenda

## Subtitle

- Introduction
- Background Review
  - Model complexity
  - Neural networks
  - Bayesian optimization
- Use-cases
  - DC and RF semiconductor device modeling and parameterization
  - Worst-case eye diagram analysis
  - Application of PINN for EM solvers
  - AI/ML for 5G and 6G Networks

# Introduction

# Why AI/ML for EDA?

- What is the *non-artificial* intelligence?
  - The ability to speak?
    - We are the only animal that can “speak”.
    - We are also the only “intelligent” one.
    - Natural language processing (NLP)?
    - Large language models (LLMs)?
- What does it have to do with EDA?
  - Go read books and learn on your own?
  - Code-free solution assisting the engineers?
- AI/ML needs to be applied at different stages of design and test.

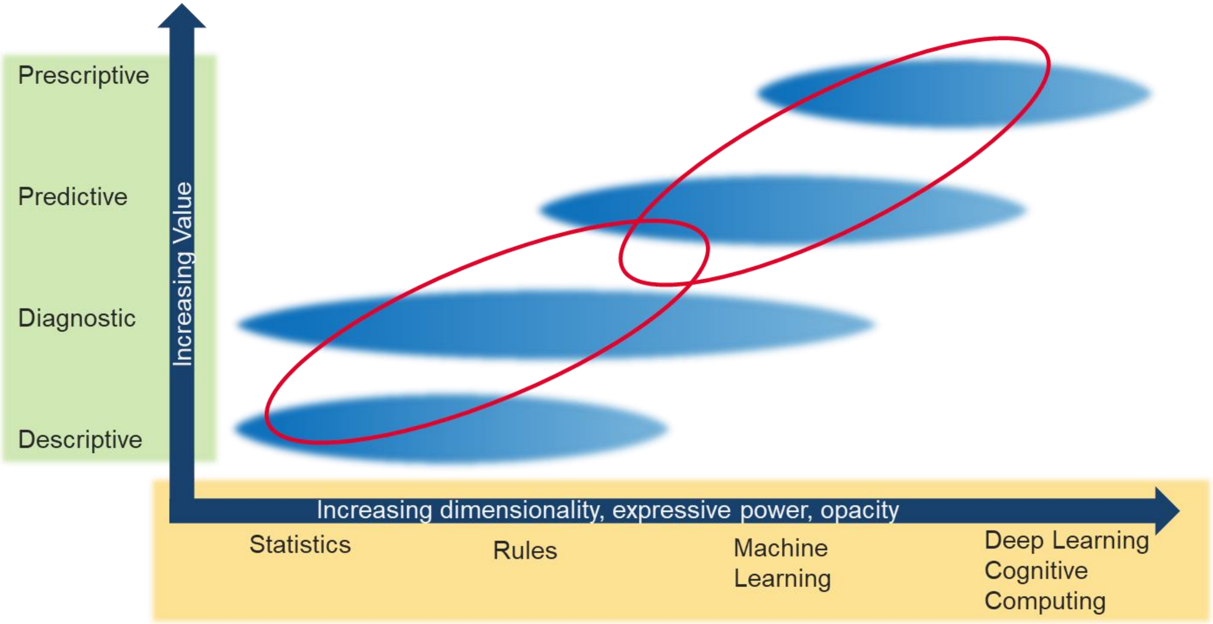
“*The ability to speak does not make you intelligent.*”
- Qui-Gon Jinn
Star Wars: Episode I - The Phantom Menace

Keysight PathWave System Design (SystemVue)


Keysight 5G Over-The-Air (OTA) chambers

# Revolutionizing EDA

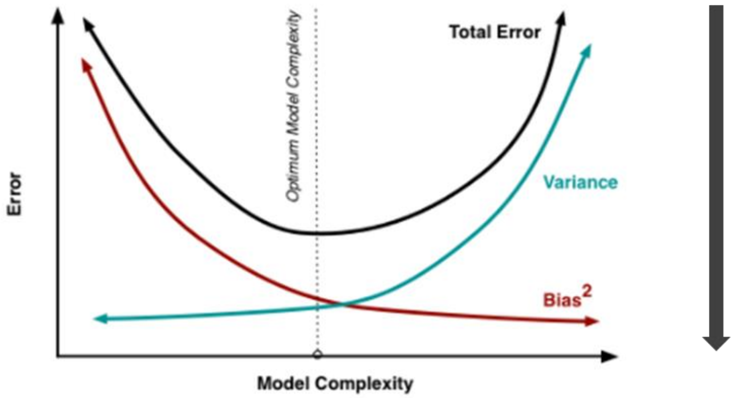
- Design complexity
  - Has been increasing exponentially.
  - Workforce is scaling at a slower rate.
  - E.g., 6G, Quantum, Automotive, Defense, etc.
- AL/ML opportunities
  - Capturing new business
  - Avoiding loss of current business
- Solution complexity
  - Should be appropriate to the problem scale, complexity, and data volume.
    - i.e., do not use AI/ML if easier solutions exist
  - Customized and modified for EDA



# Background Review

# Model complexity

- Bias and variance in linear regression $h(\boldsymbol{x}) = \beta^T \boldsymbol{x} + \beta_0$



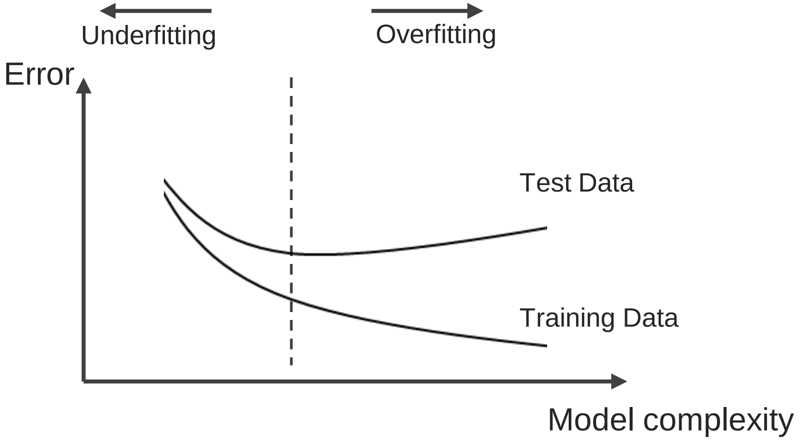

Less parameters, Simpler model
High bias, low variance
Model is underfit

More parameters, Complex model
Low bias, High variance
Model is Overfit

- Add features (parameters) to achieve Low bias
- How to control variance (avoid Overfitting)
  1. Reduce the model complexity
  2. Regularization
- Reduce model complexity will increase the bias.
  So, our choice is <u>regularization</u>
- *If you are using regression without regularization, you have to be very special!'*

  – **Owen Zhang**, Chief Product Office, DataRobot

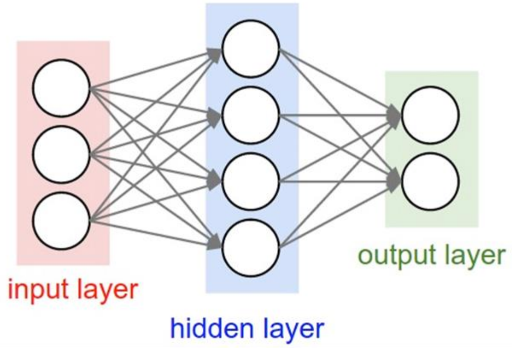
- Generalized model complexity tradeoff



# Artificial Neural Networks (ANN)

- 2-layer neural network:

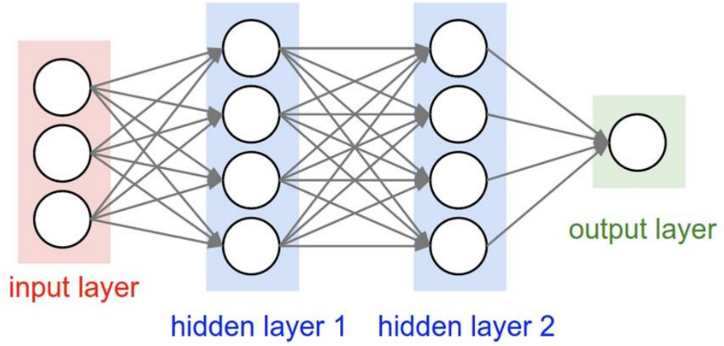
$$y = W_2 f(W_1 . x)$$




Fully connected neural networks

- 3-layer neural network:

$$y = W_3 f(W_2 f(W_1 . x)\,)$$




- Increasing number of layers
  → Deep Neural Network
  - Why do not we always use a deep neural network?

AlexNet

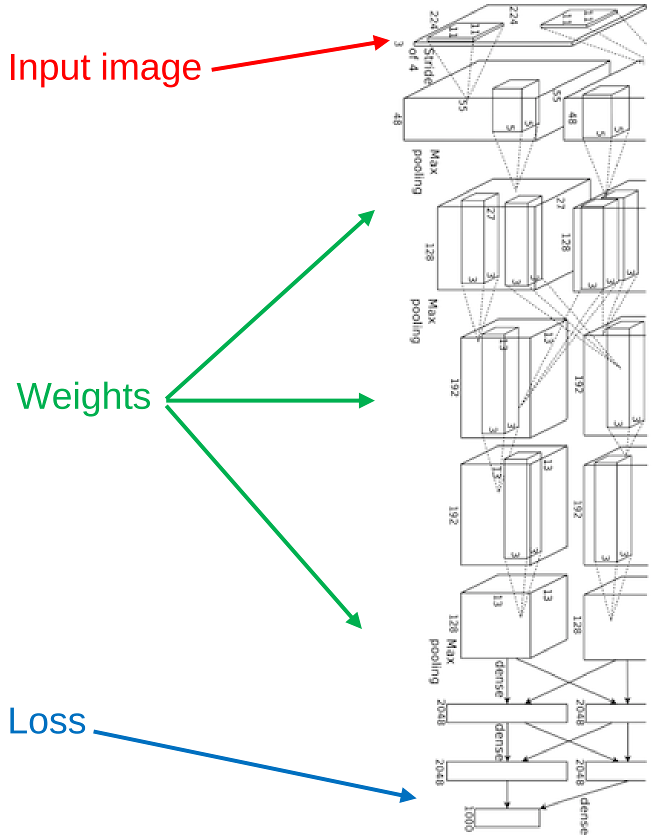
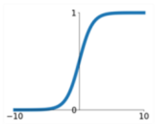


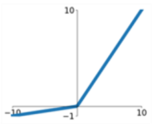

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012.

- Activation functions:

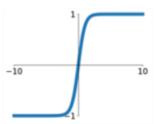
Sigmoid $\frac{1}{1+e^{-x}}$



Leaky ReLU $\max(0.1x, x)$

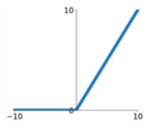


tanh $\tanh(x)$

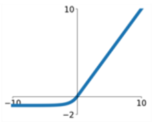


Maxout $max(w_1^T x + b_1, w_2^T x + b_2)$

ReLU $\max(0, x)$



ELU $\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

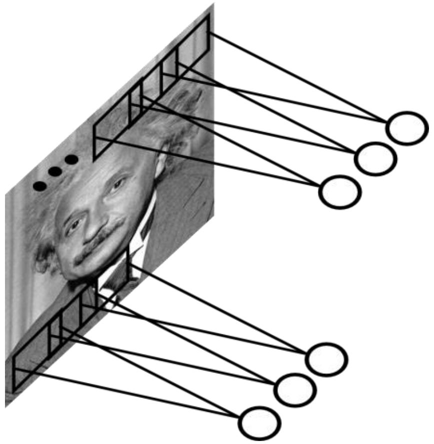


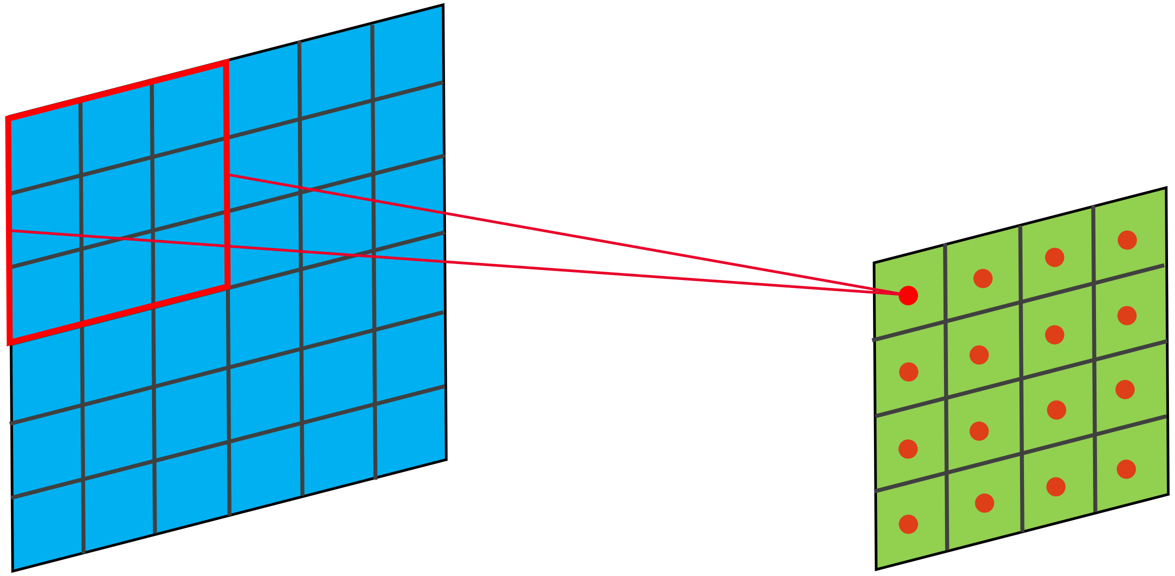


Images credit: Georgia Tech CS 7643

# Convolutional Neural Network

- Convolutional layer:
  - Spatial correlation is local
  - Share the same parameters across different locations (convolution with learned kernels)
  - Significantly decreases number of parameters





Sharing the parameters and moving the filter

From Georgia Tech CS 7643 which credits Marc'Aurelio Ranzato

- Example:

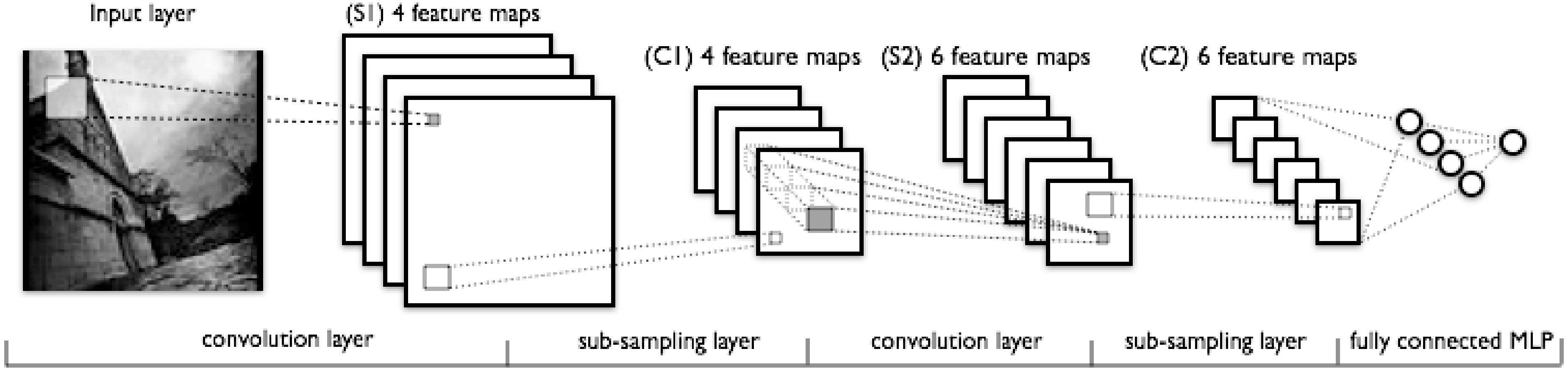



Image from Georgia Tech CS 7643 which credits Yann LeCun, Kevin Murphy

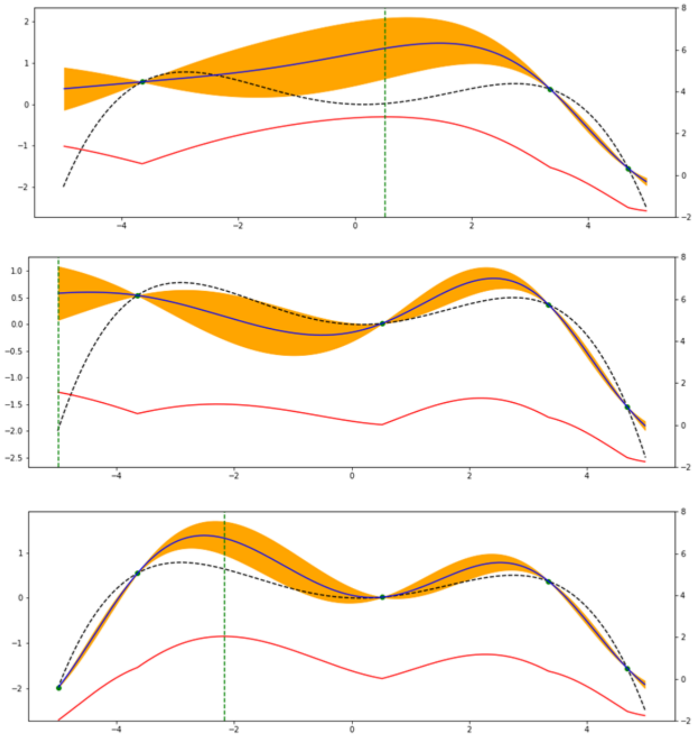
# Bayesian Optimization (BO)

Prior Distribution → Next Point → Posterior Distribution

- BO algorithm
  - Fast convergence and applicable to nonlinear and expensive functions.
- Based on the Bayes' theorem:

$$P(f(x)|D_{1:t}) \propto P\big(D_{1:t}|f(x)\big)P\big(f(x)\big)$$

  - $D_{1:t} = \{x_{1:t}, f_{1:t}\}$ is the set of observations.
  - $P\big(f(x)\big)$ prior and $P(f(x)|D_{1:t})$ posterior distributions.
  - $P\big(D_{1:t}|f(x)\big)$: likelihood of observing $D_{1:t}$ given prior $P\big(f(x)\big)$.
- How to find the posterior distribution? → **Gaussian process** $f_t \sim \mathcal{N}(\mu, \mathbf{K})$
  - Tradeoff between **Exploration** and **Exploitation**
- How to choose the next sample point? → **Acquisition function** A function of $\mu_{t+1}$ and $\sigma_{t+1}^2$



First 3 iterations of BO on a test function, acquisition function is shown in red

# Use-Cases

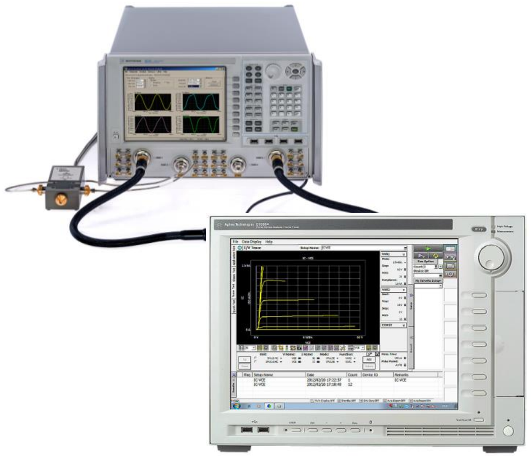
# Use-cases

| | Use Case | Technology | Field | Benefits |
|---|---|---|---|---|
| 1.a | DC and RF semiconductor device model generation | Neural Networks | Device Modeling | Automate and speedup the device Verilog-A modeling procedure |
| 1.b | DC and RF semiconductor device modeling parameter extraction | Neural Networks | Device Modeling | Automate and speedup the device modeling parameter extraction |
| 2 | Worst-case eye diagram analysis | Bayesian optimization | High-speed Digital | Faster channel analysis for non-LTI systems |
| 3 | Application of PINN for EM solvers | Physics Informed Neural Network (PINN) | RFMW(EM) | Reducing the time of MoM matrix generation and its size |
| 4 | AI/ML for 5G and 6G Networks | Convolutional Neural Network (CNN) | Communication | Improving classical channel estimation |

# 1- DC and RF semiconductor device modeling and parameterization

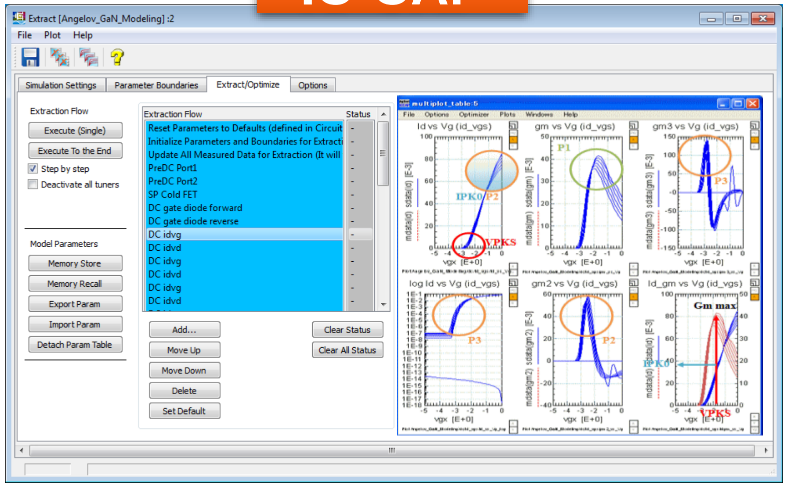
- Conventional modeling process:
  - Based on physical behavior
    - Need to find hundreds of parameters to match with measurements
  - Can take weeks to months.

```
* --------- Parameters for Transistor Model ---
.PARAM
+ BULKMOD = 0                    TNOM = 27
+ GEOMOD = 0                     RDSMOD = 0
+ ASYMMOD = 0                    IGCMOD = 0
+ GIDLMOD = 0                    IIMOD = 0
+ SHMOD = 0                      RGATEMOD = 0
+ CGEOMOD = 0                    CGEO1SW = 0
+ LINT = -1.874341E-9            XL = 0
+ LLN = 1                        DLC = 0
+ DLBIN = 0                      LLC = 0
+ EOT = 1E-9                     TOXP = 1.2E-9
+ HFIN = 2.5E-8                  FECH = 1
+ FECHCV = 1                     DELTAWCV = 0
```

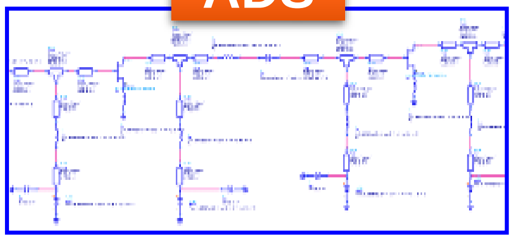


DC, CV, S-par, Pulsed IV

IC-CAP



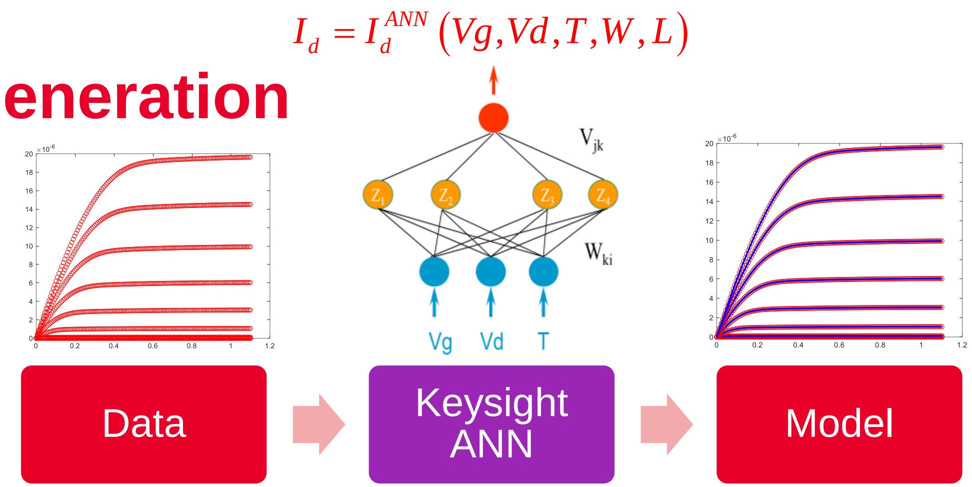

ADS



IC design

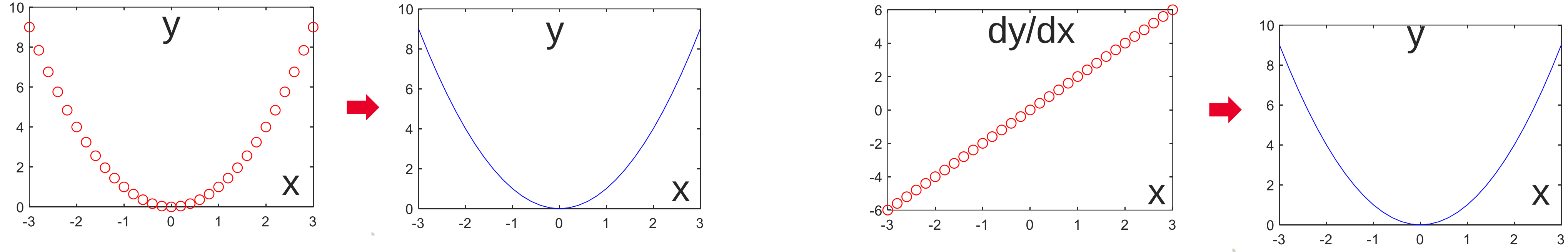
**Measurements** → **Modeling** → **Design**

# 1.a- DC and RF semiconductor device model generation

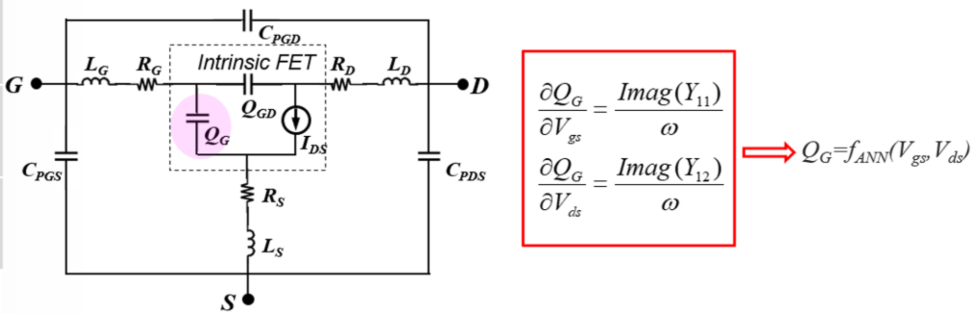



- Proposed solution:
  - Use neural networks to generate device models.
  - New *Keysight ANN* architecture to train using derivative of parameters.
    - *Keysight ANN* can extract charge models from capacitive information.

$y = x^2$

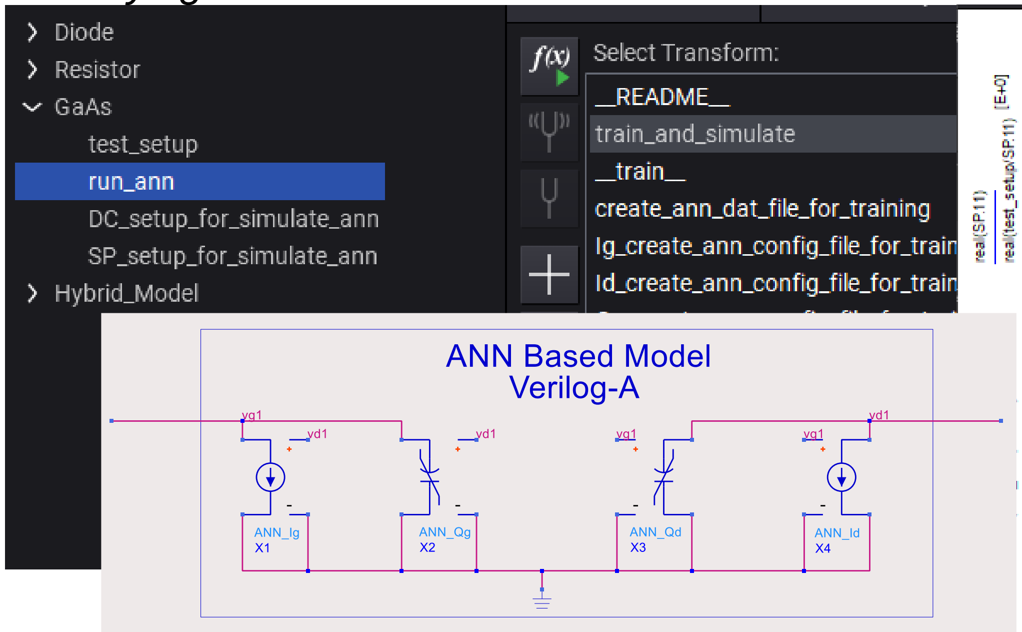


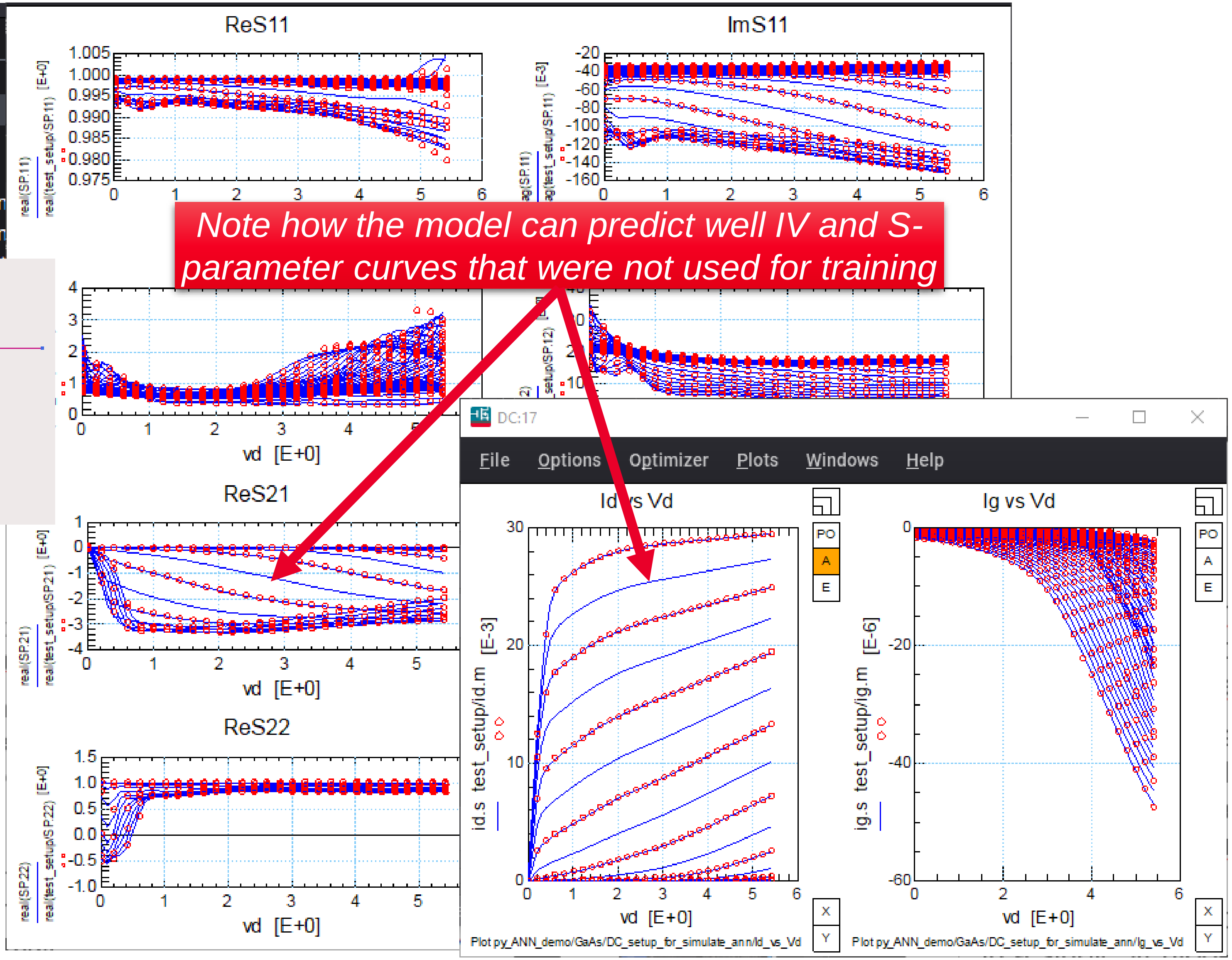

| y = $f_{ANN}$(x) | Train $f_{ANN}$ from (x, y) | Train $f_{ANN}$ from (x, dy/dx) | Export $f_{ANN}$ in Verilog-A or Text formula format |
|---|---|---|---|
| Keysight ANN | ✓ | ✓ | ✓ |
| TensorFlow (Google) | ✓ | | |
| Scikit-Learn | ✓ | | |
| Matlab ANN Toolbox | ✓ | | |



# 1.a- DC and RF semiconductor device model generation

- *Keysight ANN* EXAMPLE: GaAs HEMT



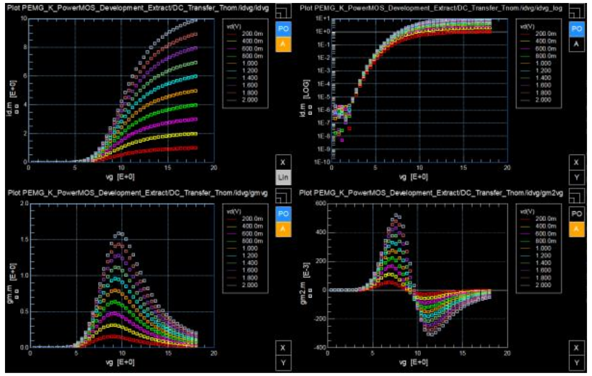




- 4x core nonlinear model functions generated in parallel from data and validated within minutes

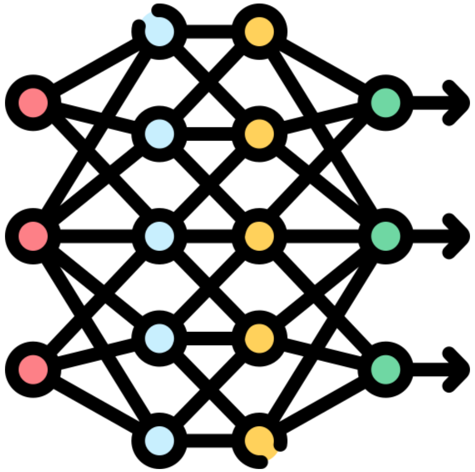
- Both currents and charges are well modeled as shown in the IV and S-parameters measured vs. simulated plots

# 1.b- DC and RF semiconductor device modeling parameter extraction

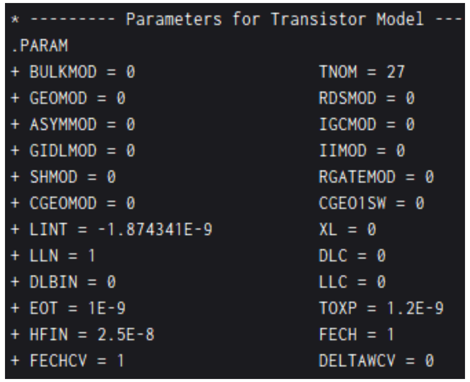
- Neural networks are non-physical
- Physical model extrapolation is **limited by physics**
- *Keysight ANN* model extrapolation is **limited by data**
- Neural networks models are not transparent
  - Lack of physical description
  - People are afraid to use them
- <u>Can we create the physical models using AI/ML</u>?
  - Find hundreds of characteristic parameters required for the physical equations



Given: Measurement Data (in many plots)



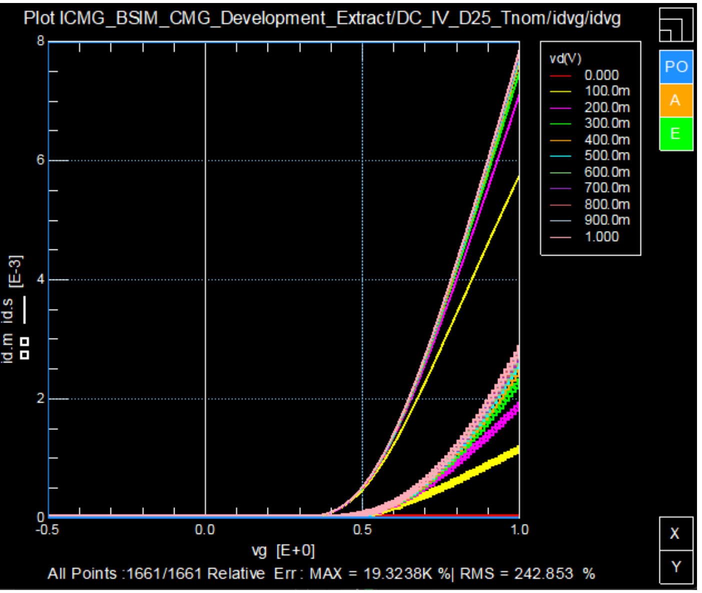
$f(x)$

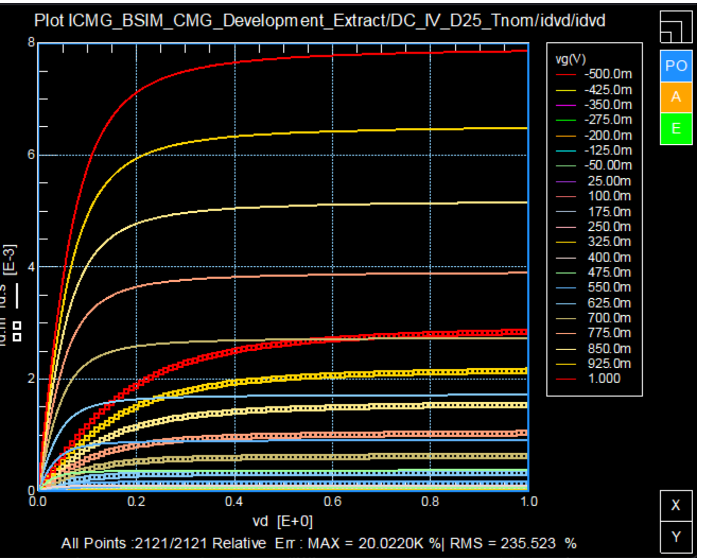


```
* --------- Parameters for Transistor Model ---
.PARAM
+ BULKMOD = 0                       TNOM = 27
+ GEOMOD = 0                        RDSMOD = 0
+ ASYMMOD = 0                       IGCMOD = 0
+ GIDLMOD = 0                       IIMOD = 0
+ SHMOD = 0                         RGATEMOD = 0
+ CGEOMOD = 0                       CGEO1SW = 0
+ LINT = -1.874341E-9               XL = 0
+ LLN = 1                           DLC = 0
+ DLBIN = 0                         LLC = 0
+ EOT = 1E-9                        TOXP = 1.2E-9
+ HFIN = 2.5E-8                     FECH = 1
+ FECHCV = 1                        DELTAWCV = 0
```

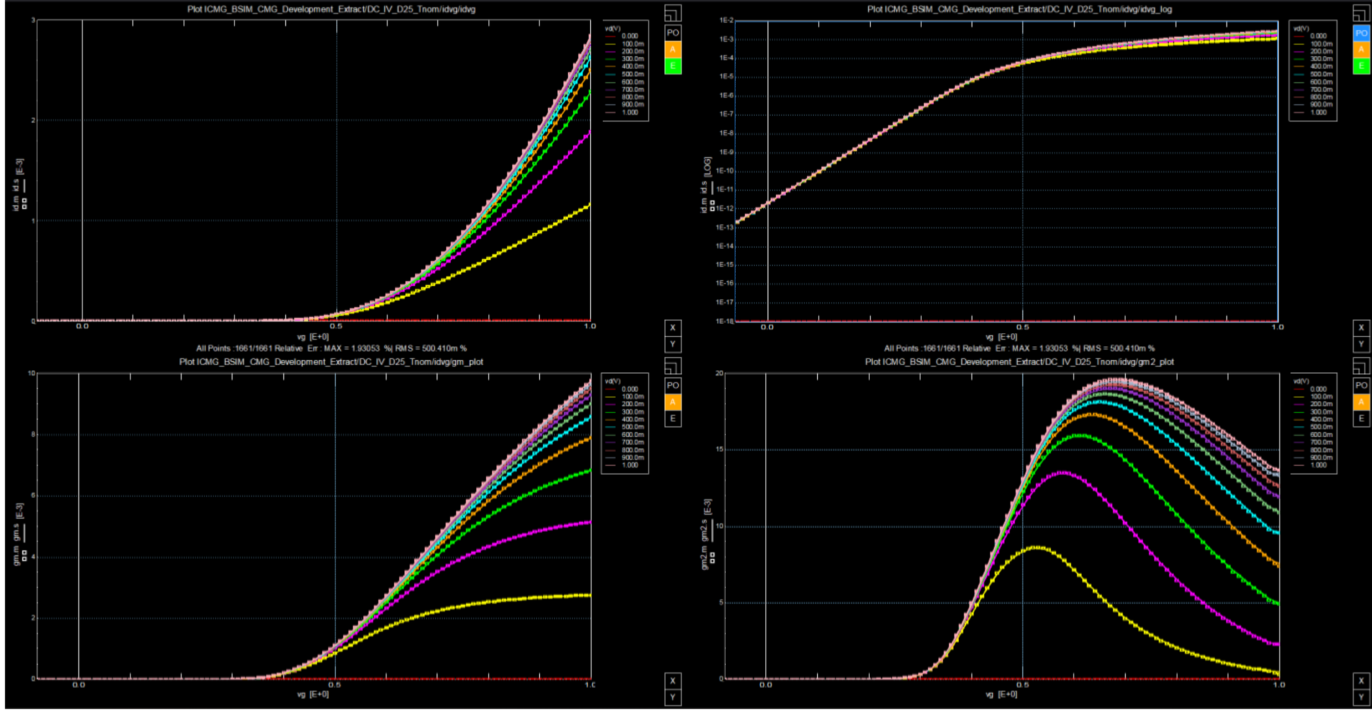
Result: Complete netlist (an array of parameters)

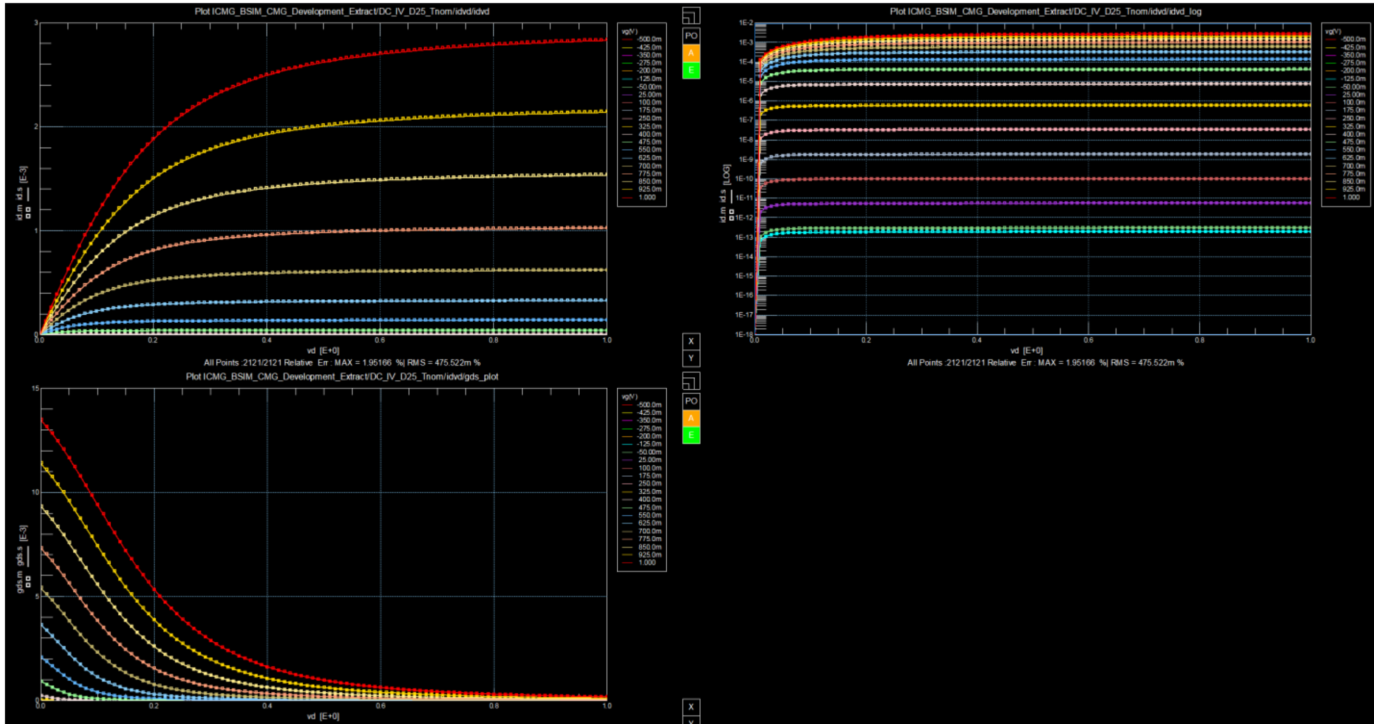
# 1.b- DC and RF semiconductor device modeling parameter extraction




- Extraction demo:
  - BSIM CMG
  - One geometry
  - Three temperatures

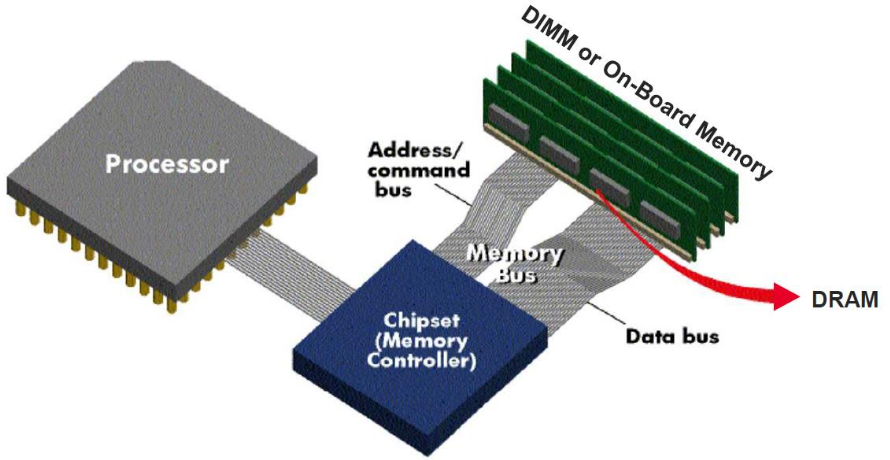


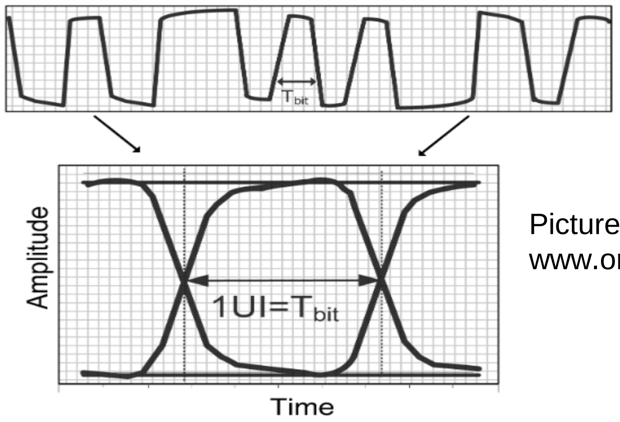


- Extraction Time:
  2 seconds
- Devices:
  DC_IV_D25
- Temperatures:
  Tnom, T2, T3
- 10 output parameters
  (40 with K_PowerMOS)

# 2- Worst-case eye diagram analysis

- High-speed serial channels
  - Chip-to-chip communication for transmitting digital signals
  - Noise and jitter from loss, reflection, crosstalk, discontinuities
  - Data rate has been increasing exponentially
- Performance of the channel is evaluated with eye diagram
  - Traditionally done by very long random transient simulation
- Intersymbol interference (ISI)
  - Tail of a single bit response interfering with next bits
  - Requires transient simulation of millions of  random bits
- Statistical eye solutions
  - Only applicable for linear time-invariant (LTI) systems
- Goal:
  - Use machine learning to find the bitt-pattern resulting in the worst-case eye

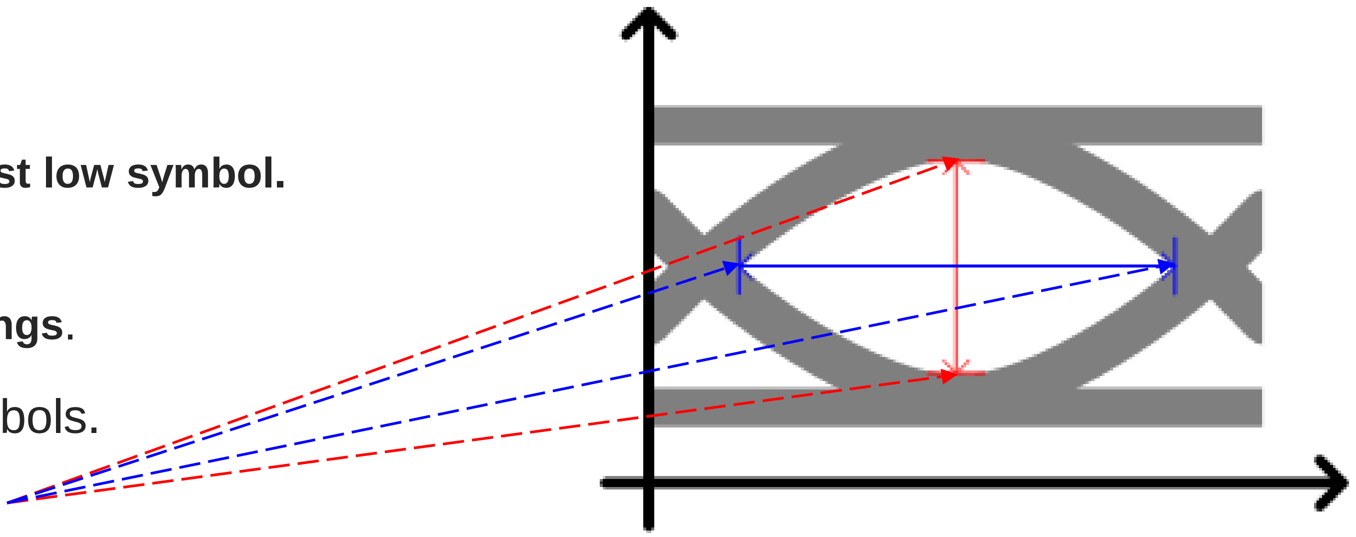


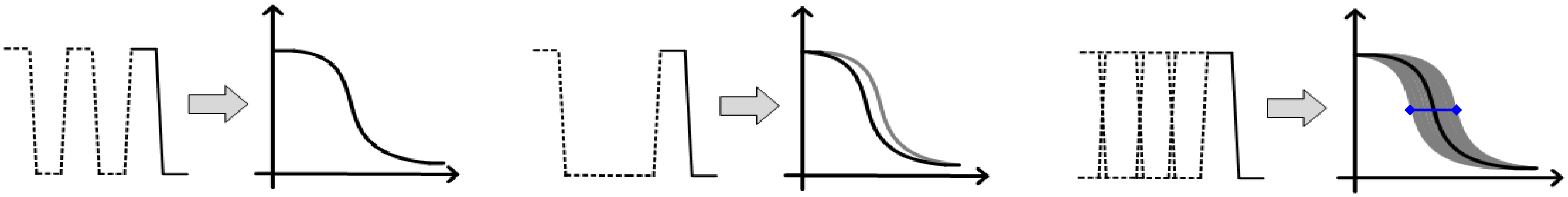




Picture from
www.onsemi.com

# 2- Worst-case eye diagram analysis

- Eye height (EH):
  - Difference of **lowest high symbol** and **highest low symbol.**
- Eye width (EW):
  - Difference of the **two innermost zero crossings**.
- EH and EW are a function of previous symbols.
- Find the bit patterns causing these points
  - Using Bayesian optimization. → Overlay the obtained waveforms → **worst-case eye**.

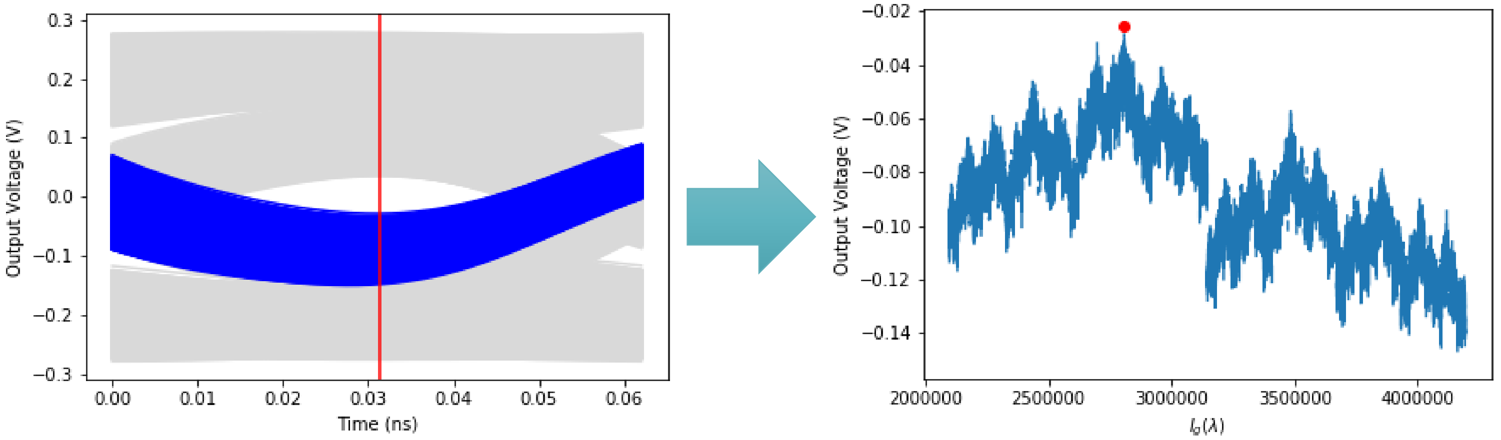




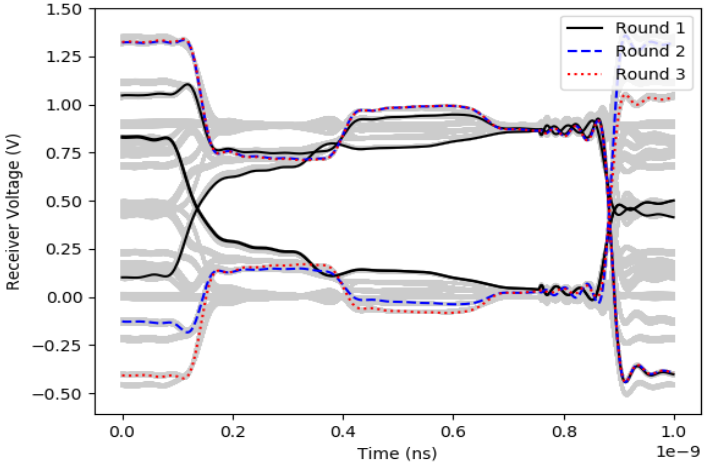
Reference: M. Ahadi, et. al. "Worst-case eye analysis of high-speed channels based on Bayesian optimization", IEEE TEMC, V. 63. N. 1, 2020. (Best TEMC Transaction Award)

# 2- Worst-case eye diagram analysis

- Give an index number to each bit pattern corresponding to the value that it shows.




- Example:



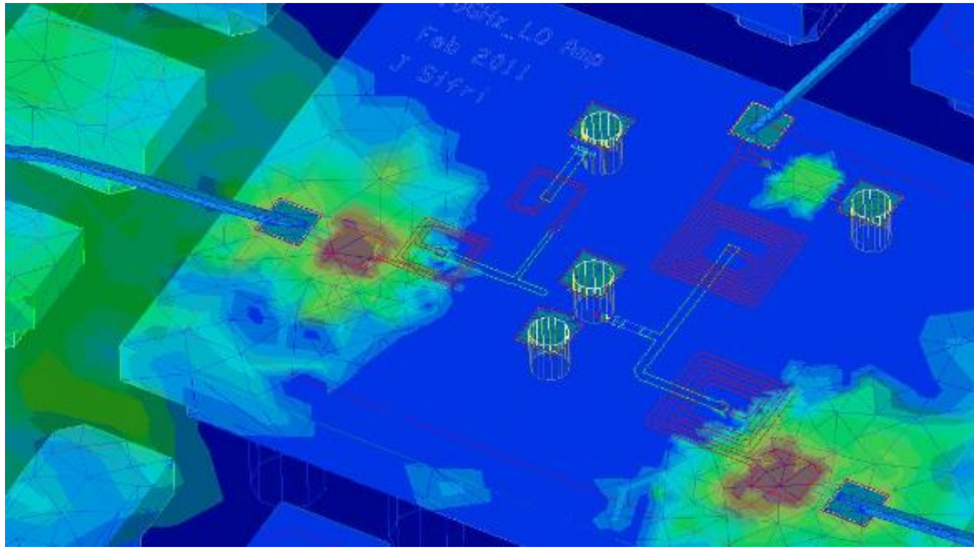

| | Eye height (mV) | Eye width (ps) | Number of bits |
|---|---|---|---|
| Worst-eye (1st round) | 639 | 745 | 5,760 |
| Transient Eye | 639 | 744 | 1,000,000 |

**47X** speedup

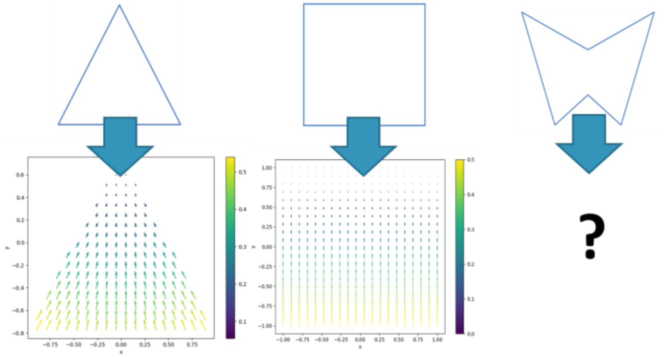
- Other applications:
  - Worst-case bit pattern generator

# 3- PINN Polygonal Vector Basis Function Model

- With increasing complexity of high frequency electronic devices
  → EM simulation performance is becoming more critical
- Planar method of moments (MOM)
  - One of the most common EM simulation methods
  - Complexity of $O(n^3)$
    - $n$: Number of unknowns on a meshed conductor
    - Meshes are comprised of triangular and rectangular simplices
      → Only vector basis functions (VBFs) for which there is a straightforward analytical solution → increasing $n$
- Recently proposed *Generalized Poisson-Neumann Basis Functions*
  - Partial differential equation (PDE) for generating VBFs on an arbitrary polygonal simplex
  - Significant improvement in simulation performance
  - Requiring solving a complex numerical problem
- Proposed ML approach
  - Utilizing Physics Informed Neural Network (PINN)
  - Solving the PDE problem
  - Building vector basis functions on a generalized polygonal domain



Keysight Pathwave Momentum
Advanced Method of Moments (MoM) 3D planar EM simulator

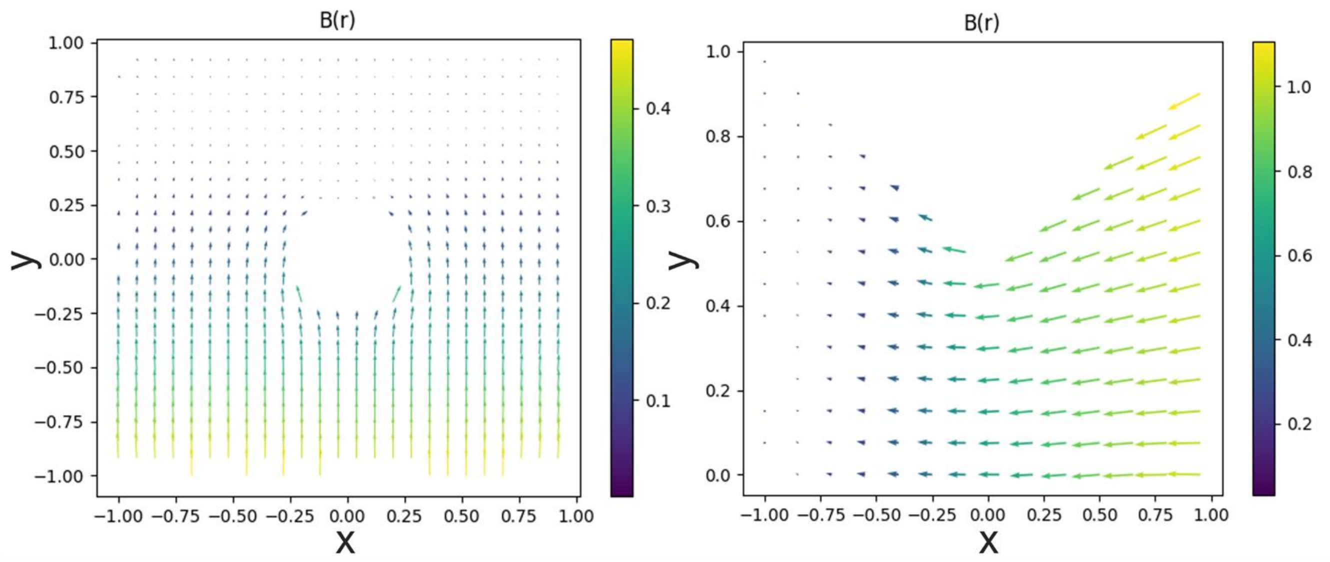



While there are straightforward analytic solutions for triangular and rectangular VBSs, there is none for general polygons.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860023.*

# 3- PINN Generalized Polygonal Vector Basis Function Model

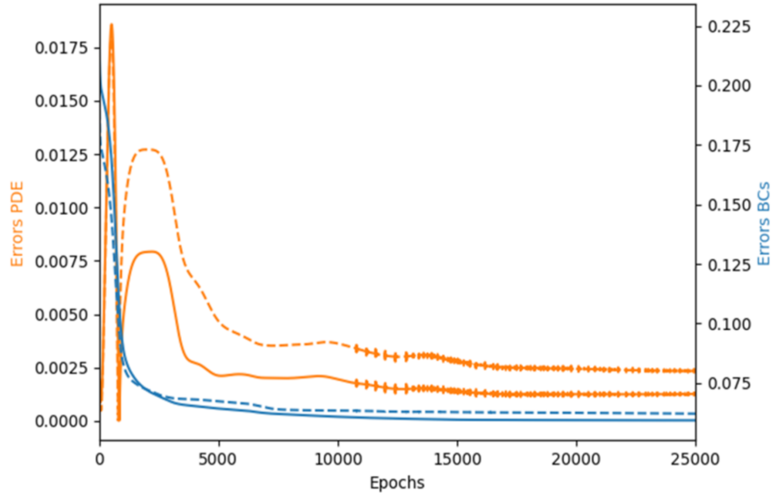
- Physics informed neural networks (PINNs)
  - Able to solve PDE problems by encoding the differential equation and boundary conditions into the objective function used for training.
- Proposed approach:
  - Training a PINN model to store general VBF properties
- Advantages:
  - Fast *ad hoc* query of the PINN model during the MoM solution building
    - Reduces time and memory requirements
  - PINNs are not limited by the convexity and simple connectedness solution requirement
  - Inherit transfer learning property of NNs



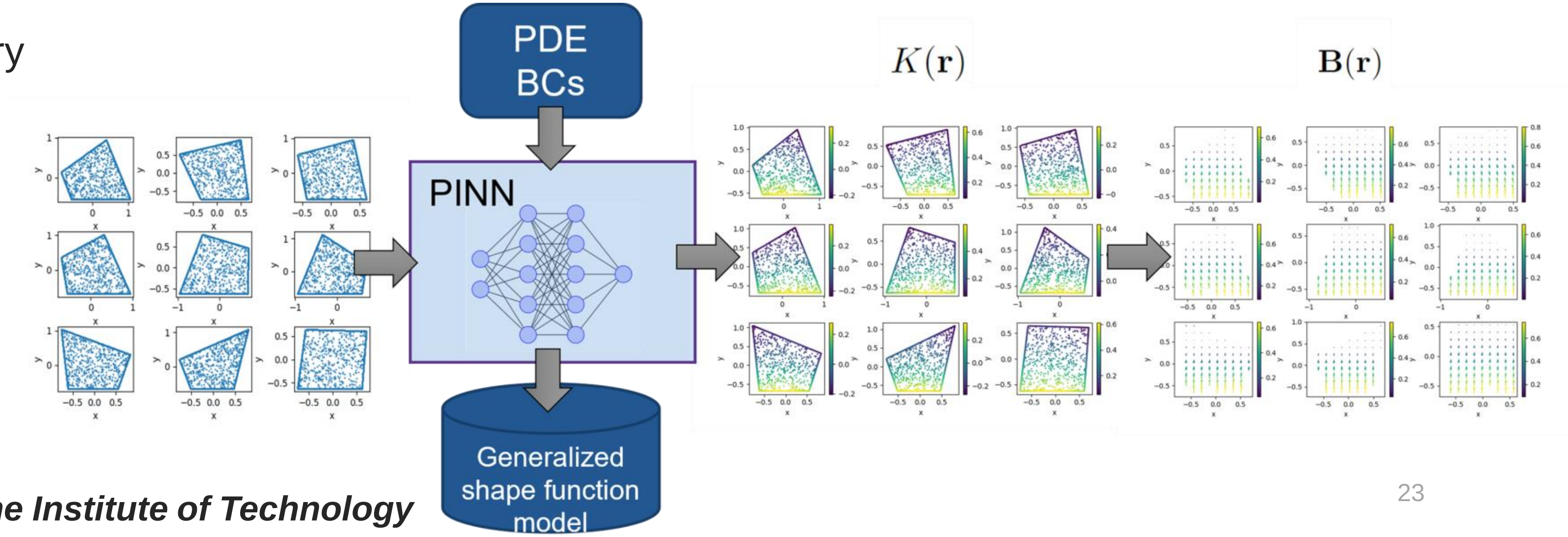

VBFs on a non simply connected (left) and non concave (right)

# 3- PINN Generalized Polygonal Vector Basis Function Model

- Example:
  - Training the PINN on a set of quadrilaterals
  - Using PINN Python library (deepXDE)
  - Randomly generated quadrilaterals
    - 40 for training
    - 10 for validation
  - Shallow NN: 2 layers with 30 neurons each
    - To enable fast ad hoc queries
  - Sampling: 500 points within the domain of the shapes (trained on PDE) and 500 sampling points on the boundary (trained on BCs)

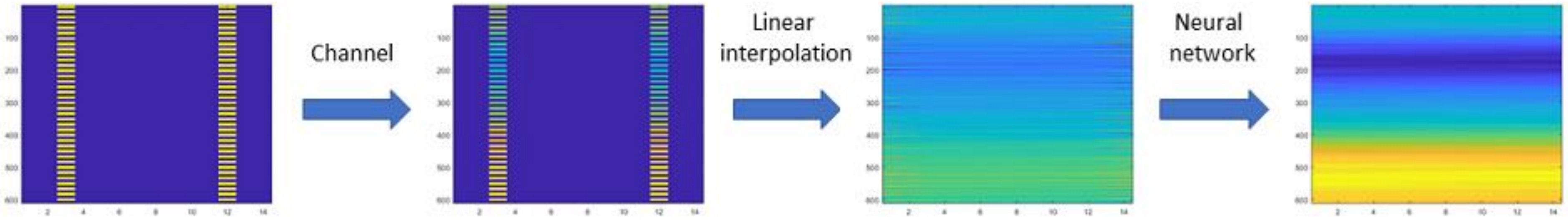



Average error of the training set (full line) and the validation set (dashed line) throughout training

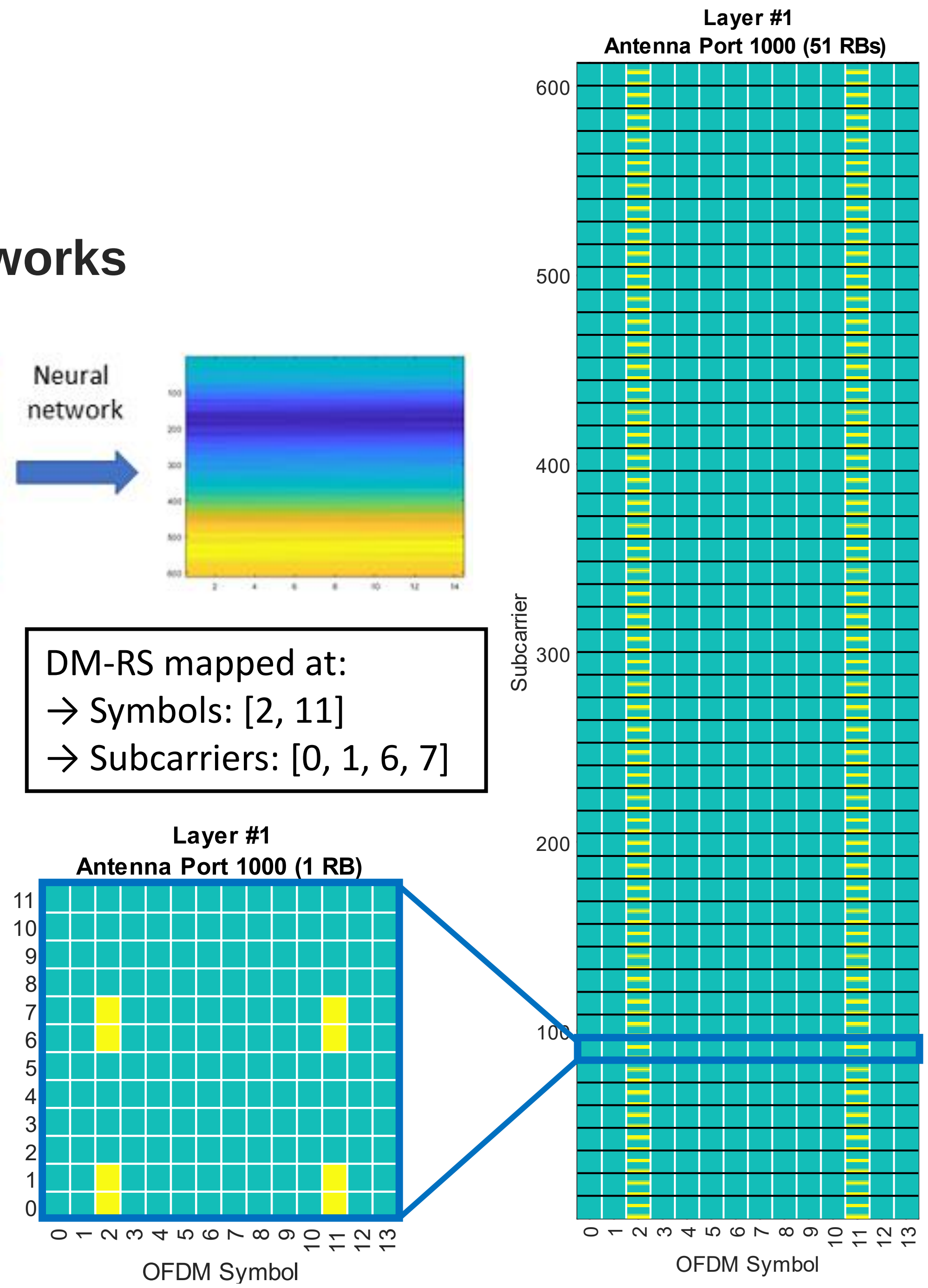


# 4- AI/ML for 5G and 6G Networks

- **Channel estimation using Convolutional Neural Networks**

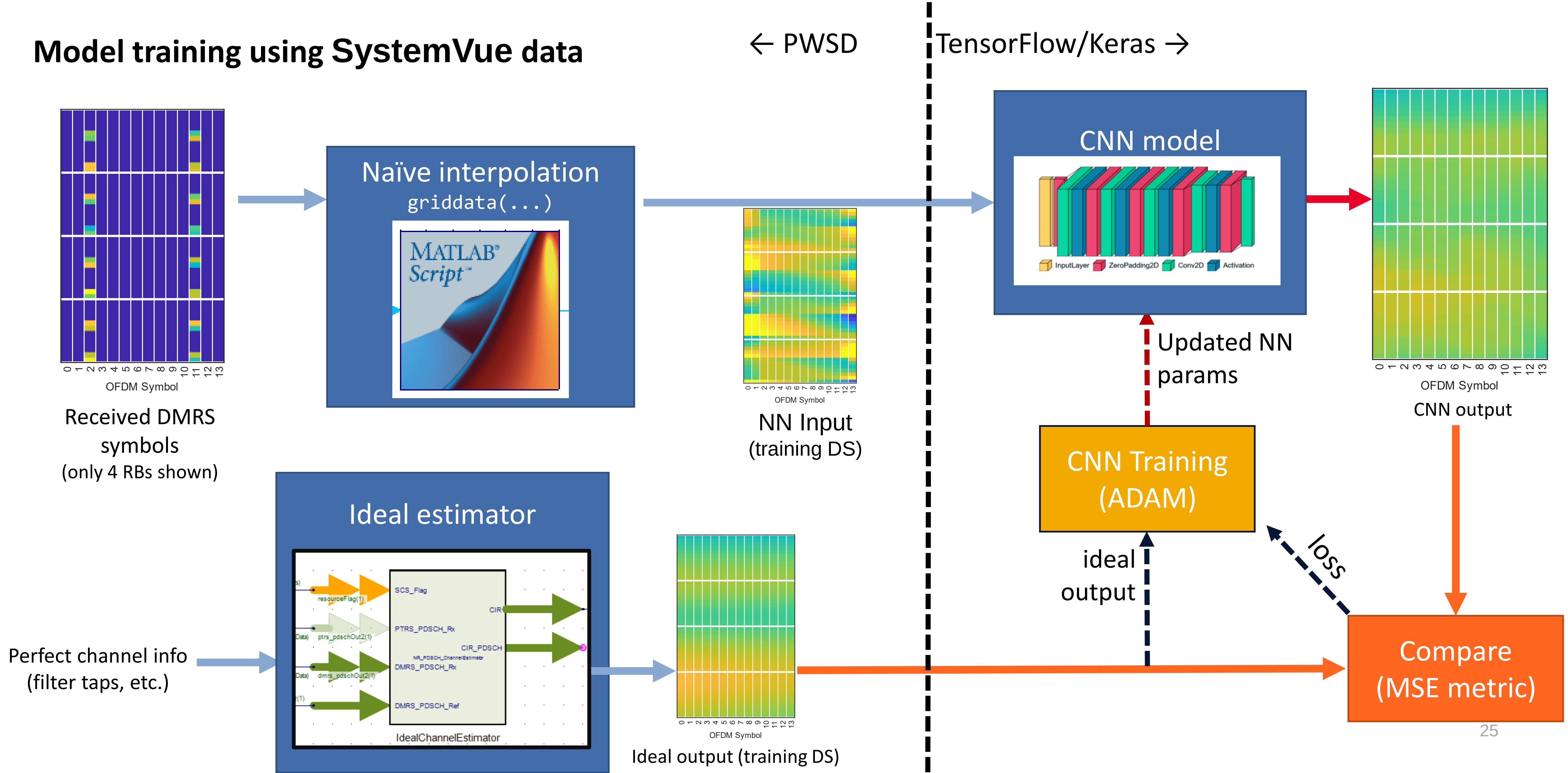



- PathWave System Design (SystemVue) workflow PoC
- Transmission: 5G DM-RS pilots over CDL channel
- **Goal**: Estimate the channel from the pilots, using NNs interpolation techniques improving what can be done with classical methods (MMSE)



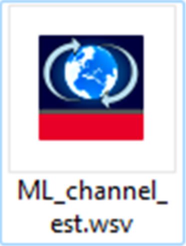
# 4- AI/ML for 5G and 6G Networks

## Model training using SystemVue data



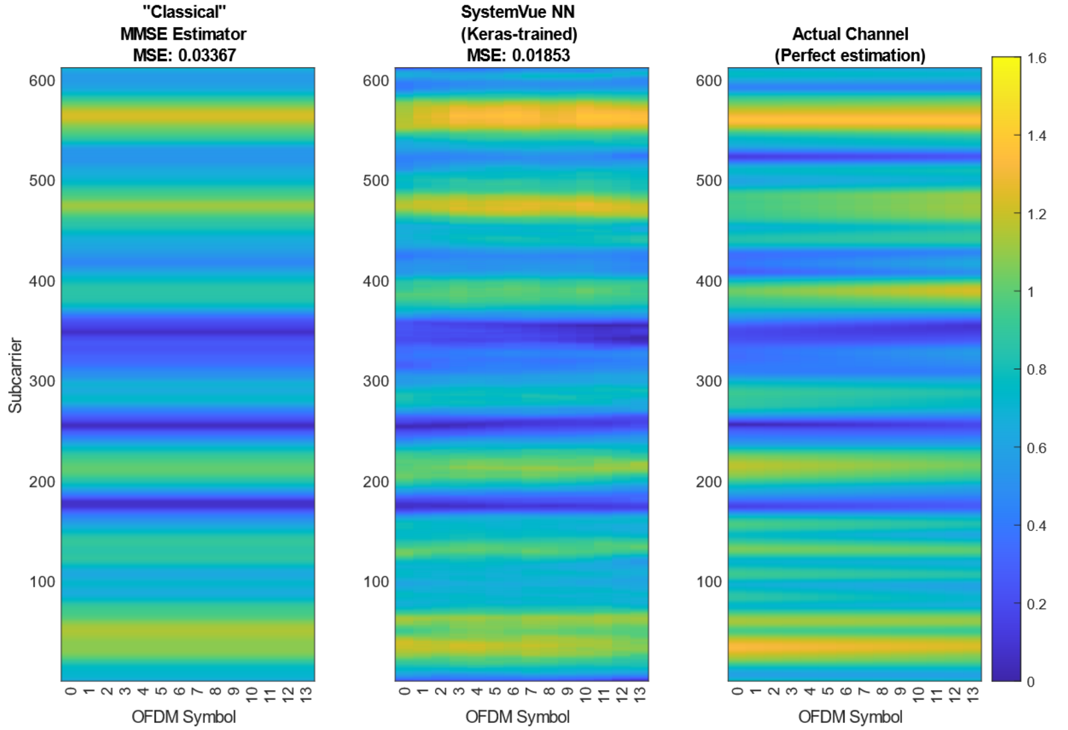
# 4- AI/ML for 5G and 6G Networks

- Performance review:

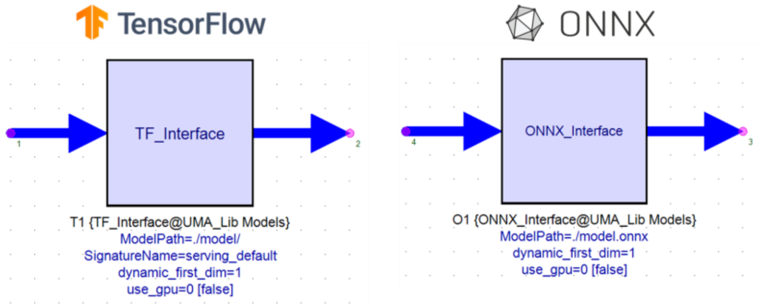



+

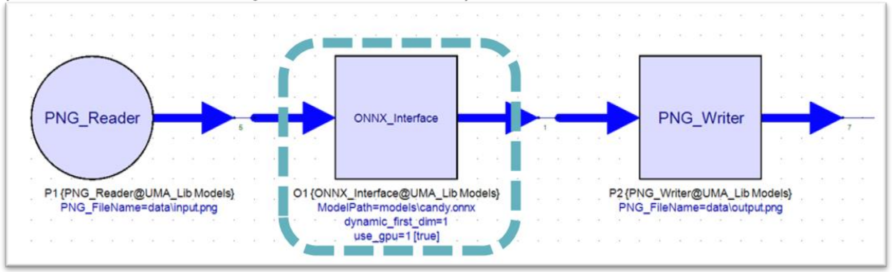
Test Dataset
.mat file
(NN input + label)




- Machine learning model inference in SystemVue:




Artistic style transfer Neural Network example in PWSD
*(Based on the work by J. Johnson et al.)*




`input.png` → AI/ML processing → `output.png`

UNIVERSIDAD DE MÁLAGA *This project is in collaboration with **Universidad de Málaga***

# Thank you

Questions?
Majid.ahadi@keysight.com