KEYSIGHT

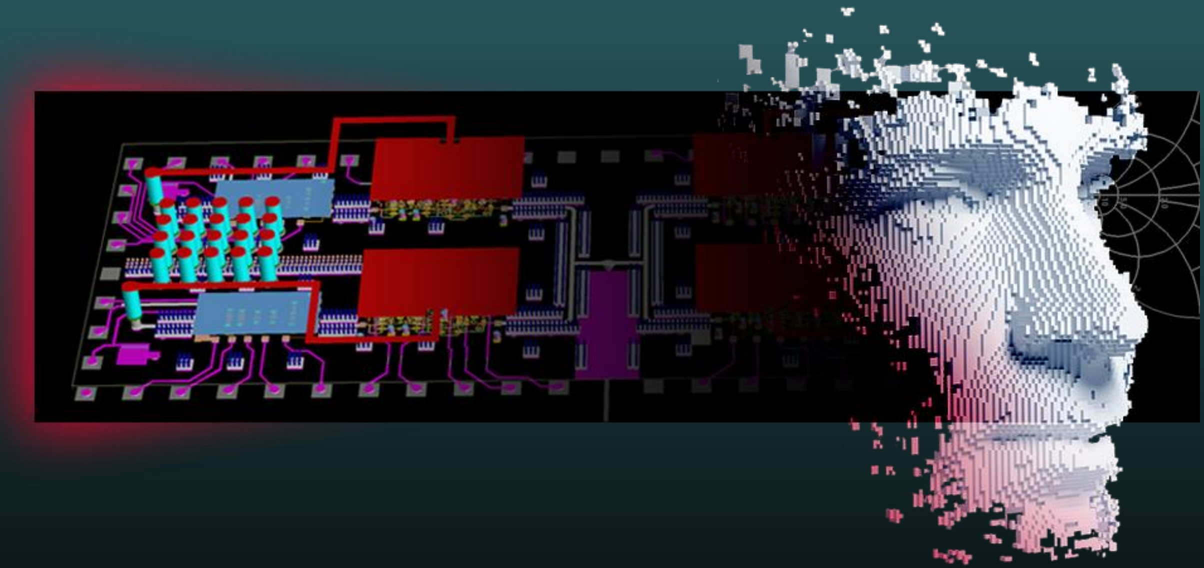
Re-imagining RF Design Using ADS Python APIs & AI/ML

Daren McClearnon
AI/ML Prod Mgr, Keysight EDA

Re-imagining RF Design Using ADS Python APIs & AI/ML

Agenda

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
   (Video from Baylor Univ)
3. Case 2: “QR Code” Filters with RFPro
   (Video from Chalmers Univ)
4. AI/ML Technologies in ADS



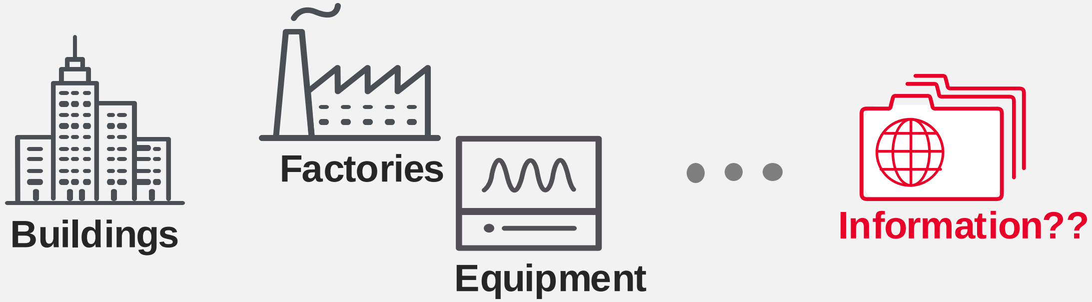
Objectives of this module

- Awareness of ADS Python APIs, emerging applications
- Learn how people are putting Data into motion using Automation and AI/ML
- See resources to help you get started

Customer/Industry Challenges

EDA Industry & Customer Challenges

- Shorter market windows, greater complexity
- Complex tool flows
- Skilled Engineers are at a premium
- Scalable, but…
  - ….needs to be secure, traceable, sustainable



Automation

- Enables **information** to become more like a tangible asset
- Engineering organizations who fully leverage information assets will be more successful than those who do not or cannot.

→ Put Your Data in Motion

Keysight Solution Approach – Move Toward An Application Layer

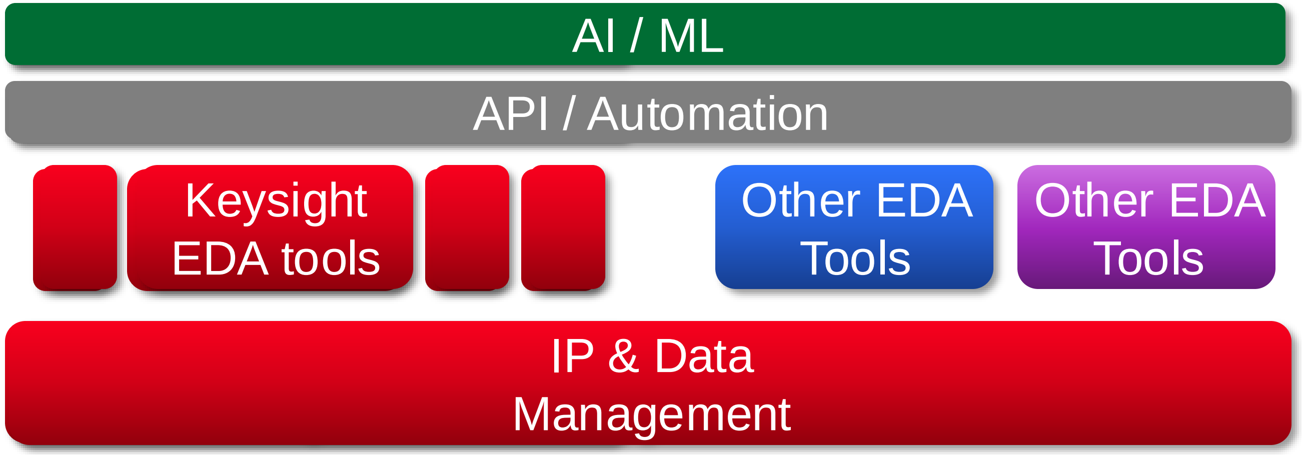
Electronic product lifecycles and workflows usually span many functional boundaries

Development of an Application Layer

- Efficiency: Streamlined Design personalities
- Agility: react quickly to changing opportunities
- Innovation: keep pace at the speed of the industry, create proprietary advantages

Scalability

- Connect to HPC, parallelization, GPU accel
- Enterprise system integration
- Unlock potential for AI/ML and machine-driven automation



Keysight Solution Approach – Modularity and APIs

Create tailored, more efficient Workflows, Personas, and Use Cases, while enabling a future "Agentic" Layer

Overcoming Rigid, Complex Tools

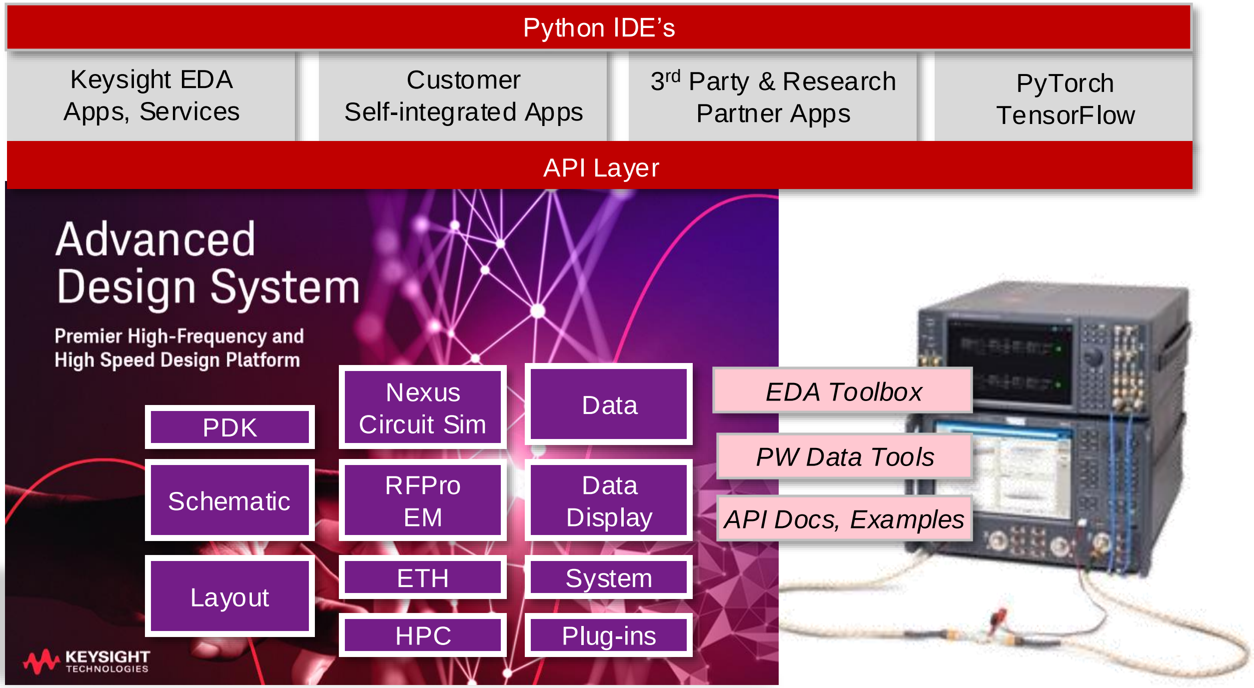
- More Modularity, APIs, SDKs/Toolkits
- Move from "Tasks" → "Workflows"

Adapt faster with intelligent Automation and accelerate Value Generation

Cross traditional boundaries between functional blocks and vendors

Enable New Use Cases

Quickly address opportunities

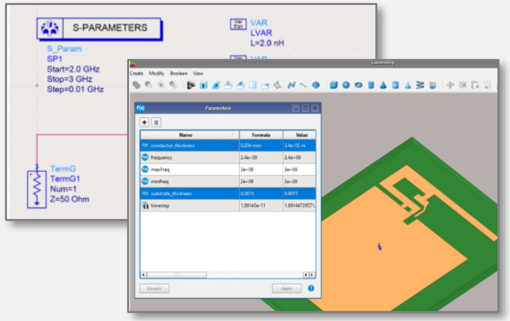


- ADS has improved its internal Modularity
- Python is the popular language choice
- Documented APIs make integrations easy

Broad Python tool support for ADS 2025 U2

EDA Toolbox: “Wrapper API’s” to drive & connect tools

- Simulate a circuit using Python (ADS)
- Co-optimize antenna+ matching network (ADS+ EM Pro)
- Driving SystemVue using Python API (SV)
- Measurement demo (VSA)

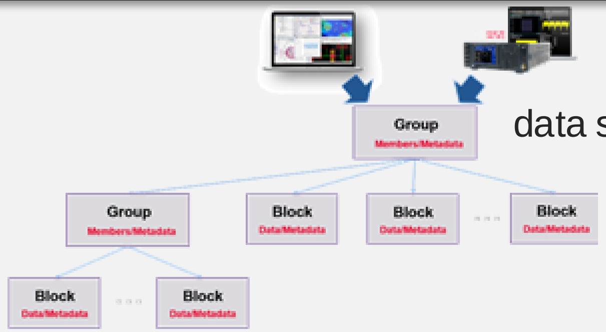



```
from keysight.edatoolbox import ads, circuit, util
```

PW Data Tools: Data Interoperability

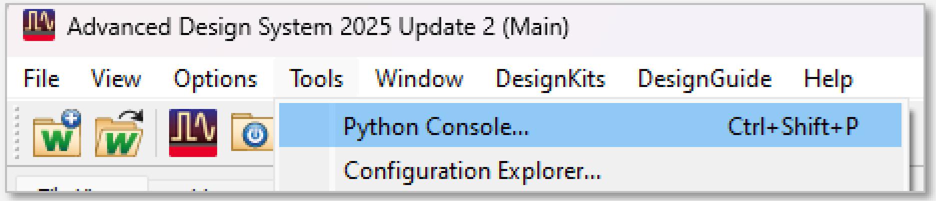
- Universal Data Structures + Easy and Consistent File I/O
- ADS, SV, HFSS, IC-Cap, Touchstone, Citi, **Load Pull** + more!

```
from keysight import pwdatatools as pwdt
```



data structure

ADS Python Console: ADS Main window, Data Display



Standard IDE’s: Code, Debug, Run with ADS

KEYSIGHT

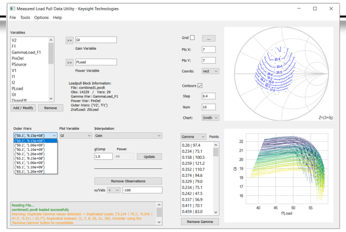
Use Cases for Python Automation

Users have Different Motivations and Use Models

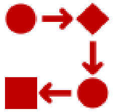
Personal Productivity

For End-users and Designers
who need to design hardware products more efficiently

New Use Models Enabled
- Menu-based Task automation
- Interactive Jupyter-style command line interface
- Standalone apps

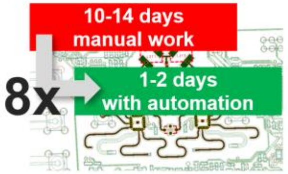


Enterprise Automation



For EDA Workflow Developers
who write software that streamlines workflows for a team

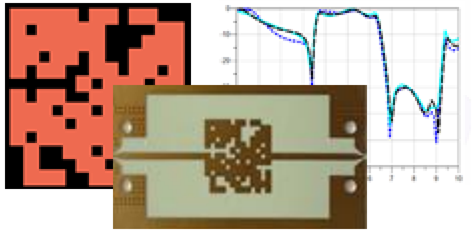
New Use Models Enabled
- Scalable verifications
- Multi-tool workflows
- Enterprise orchestration
- Data and IP Management and Engineering Lifecycle



Exploring AI/ML

For Data Scientists
who transform insights with applied Machine Learning and AI

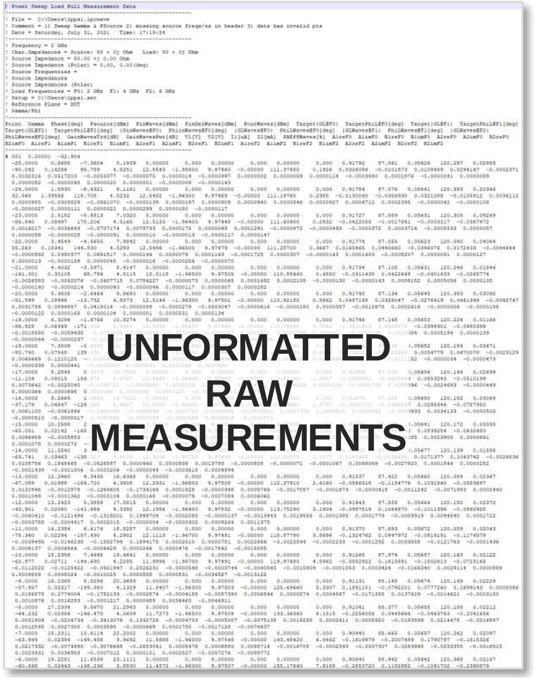
New Use Models Enabled
- Trusted Training Data generation
- Multi-domain Optimization
- Domain awareness for AI/ML development



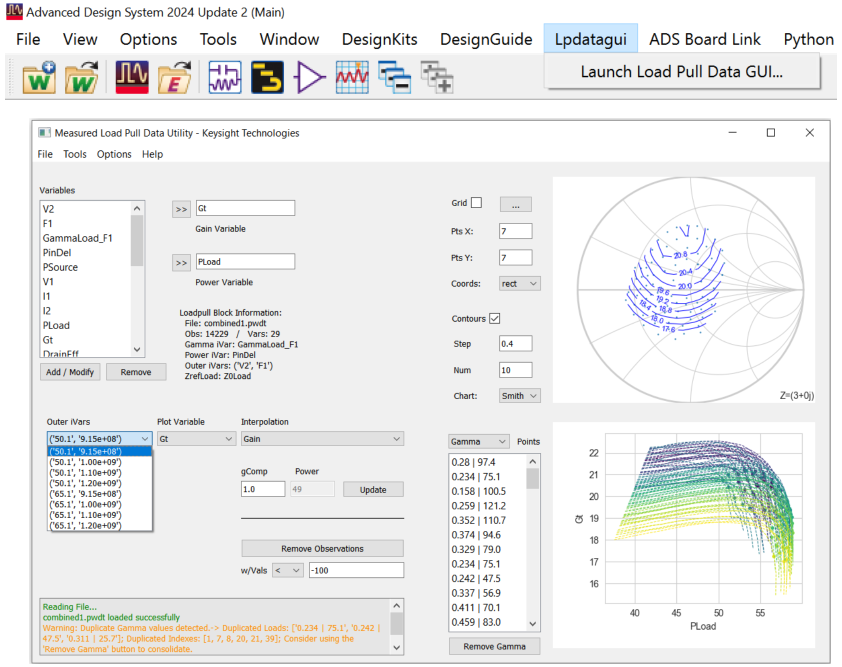
Direct link

Use Case 1 (Productivity) : Measured load pull data importer

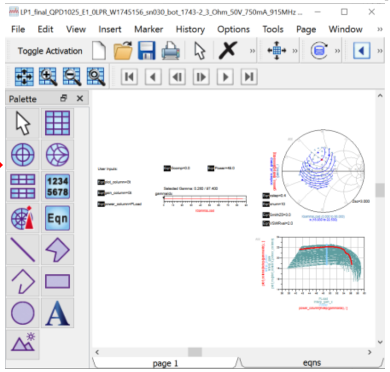
LOADPULL DATA IMPORT



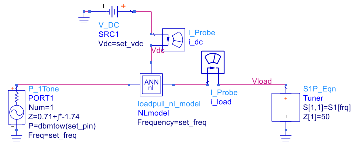
Custom Python GUI to import and combine measured data, generating datasets, data displays, testbenches and models.



Datasets and Data Displays



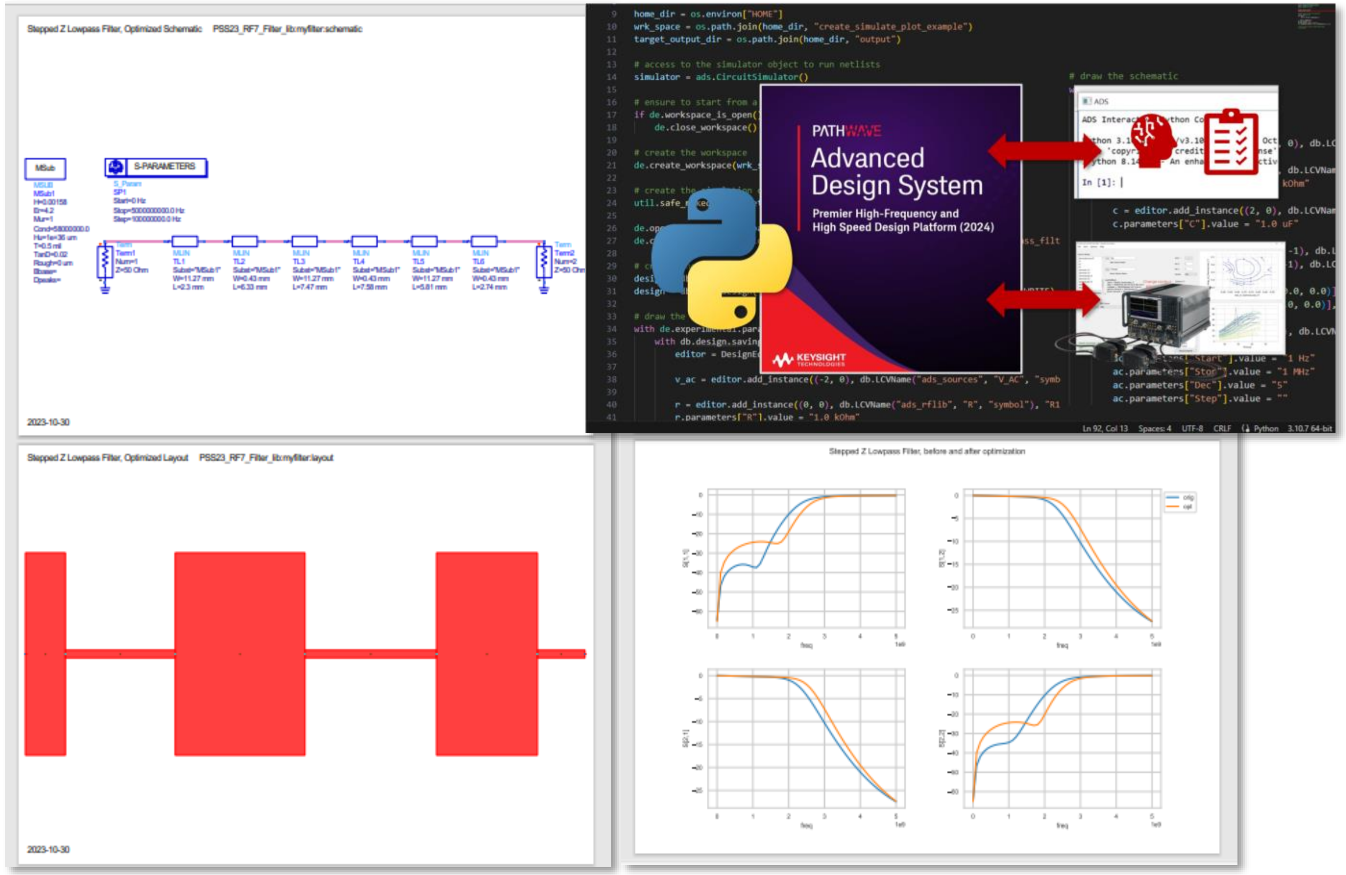
Artificial Neural Network Simulation Models and testbenches

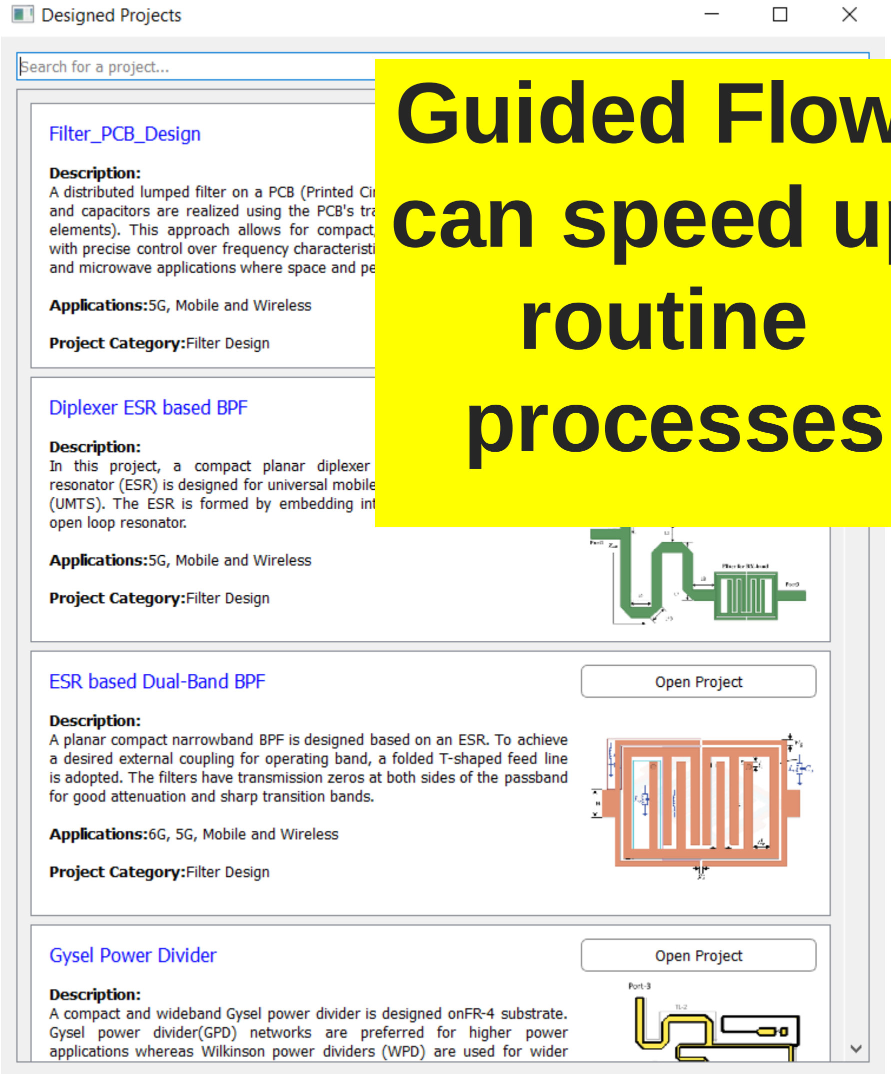


Watch now:
https://www.keysight.com/content/dam/keysight/en/vid/ungate/demos/Load-Pull-Modeling-with-Python-s-AI-Power.mp4

Use Case 1 (Productivity) : Scripted design personalities

Filters created and optimized in a headless, automated method in Keysight ADS – an on-ramp to AI/ML



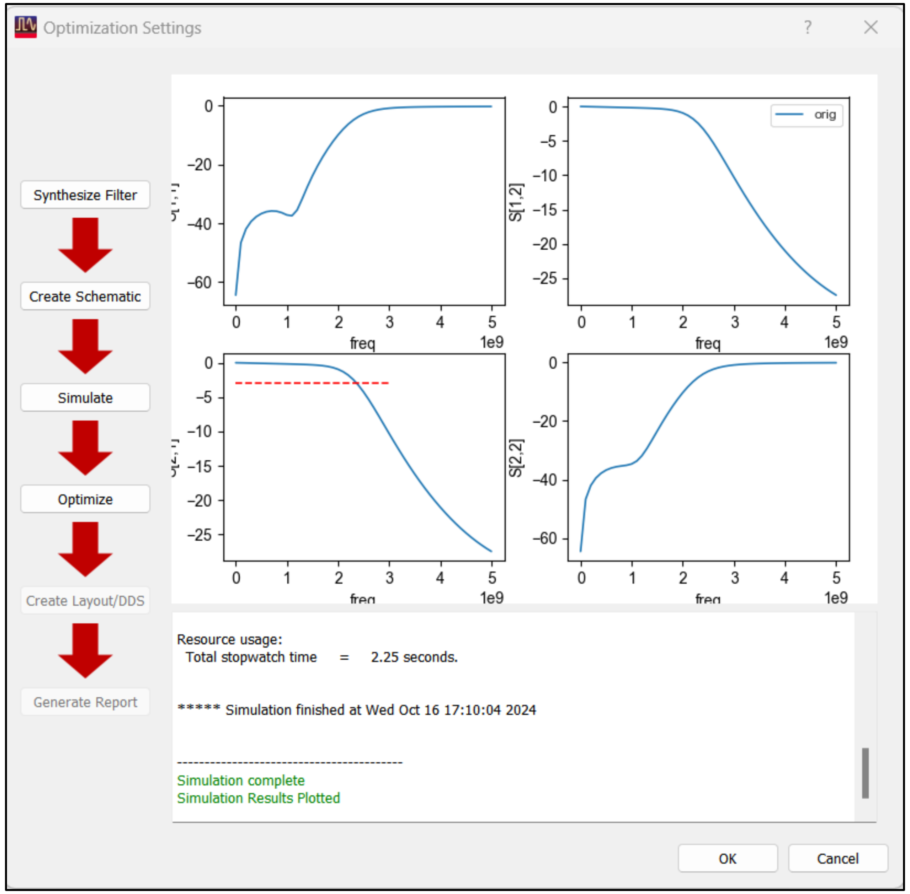


Guided Flows can speed up routine processes

Personal Productivity Demo: ADS Filter Workflow

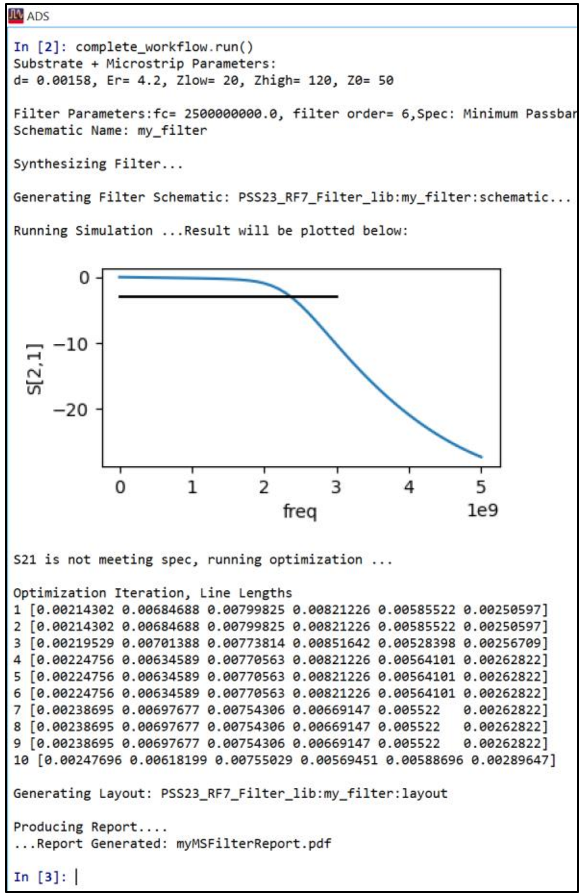
3 ways to run interactive applications

Custom Menu and User Interface

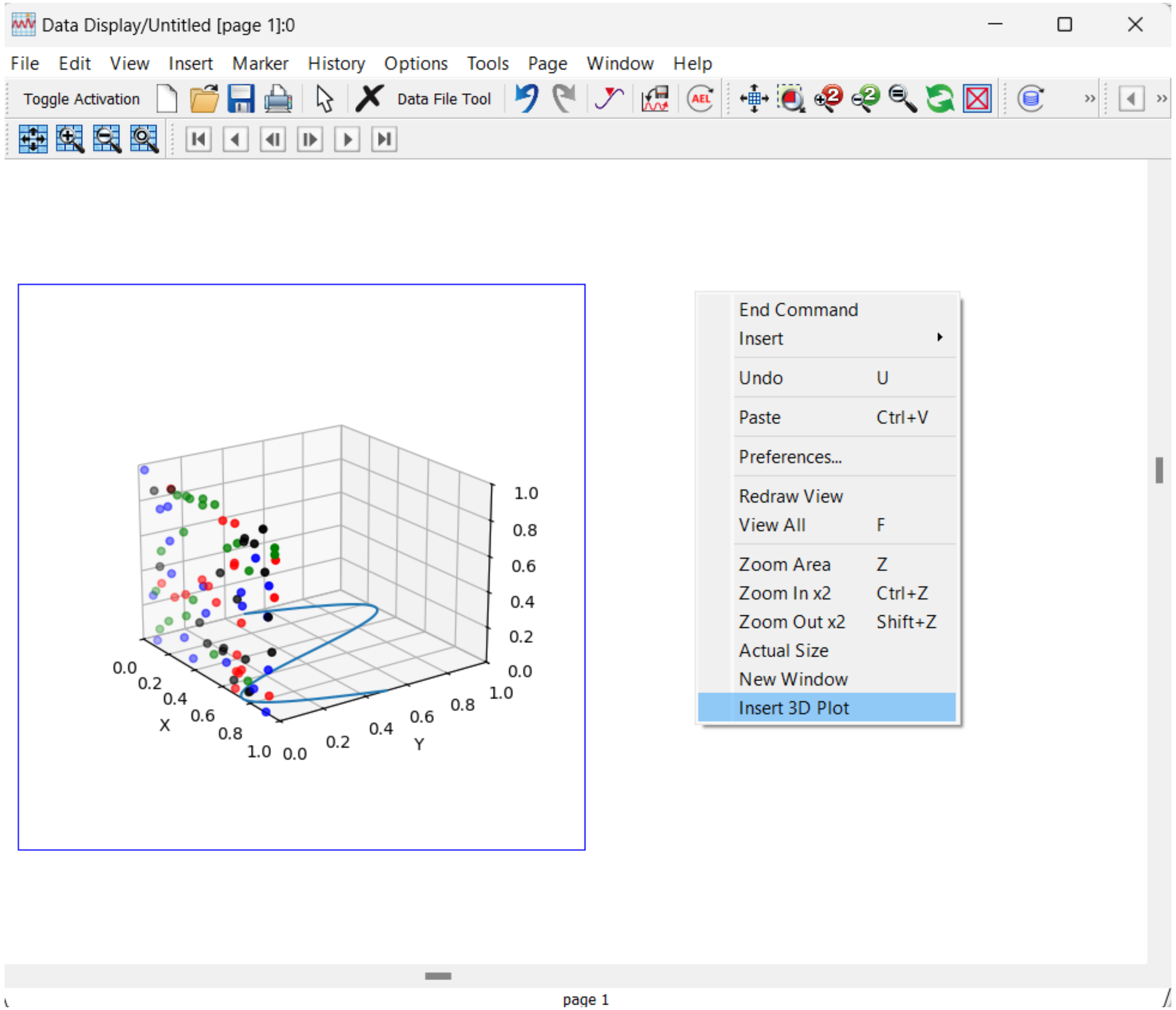


Interactive Jupyter Notebook

ADS Jupyter Python Console



Use Case 1 (Productivity) : Embedding an Interactive in Data Display

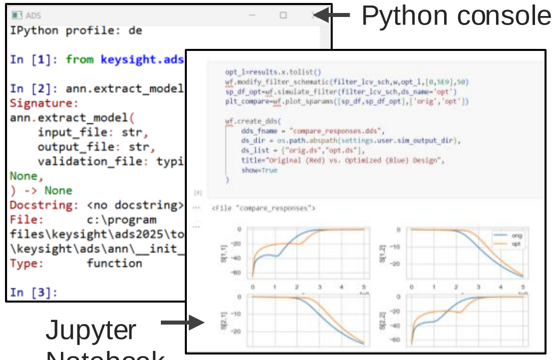


- Python callbacks allow QT objects (widgets, lists, buttons, forms) to insert into Data Display Pages
- Callback is invoked from a Right Click menu, but you can control that
- This code was implemented using the new Python Add on capability in Data Display, where you can add or modify Menus and even pop up a UI
- This example ships with ADS 2025, under: $HPEESOF_DIR/dds/python/app/examples/ex_addon_3d_plot
- Note: Beta Feature

Use Case 1 (Productivity): Small customizations

ADS 2025 Update 1 Python: Automate / Customize ADS using native API's

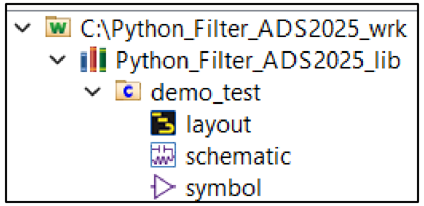
Run inside ADS, or external "headless"



Python console

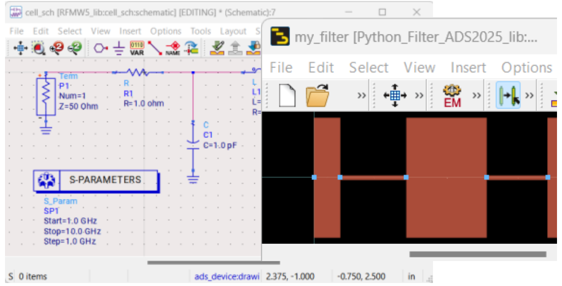
Jupyter Notebook

Create workspace, libraries, schematics, layouts, symbols



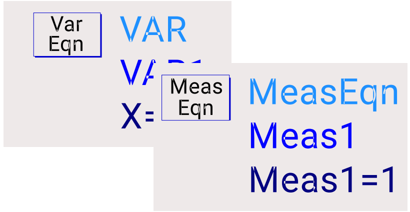

```
de.create_workspace(path)
de.create_new_library(name,path)
db_uu.create_schematic(LCVname)
db_uu.create_layout(LCVname)
```

Modify & Edit Existing Designs, Set or change instance parameters



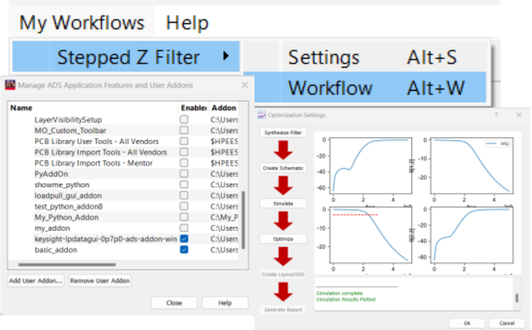

```
C1["C"].value="1.0 pF"
```

Work with Vars + MeasEqns



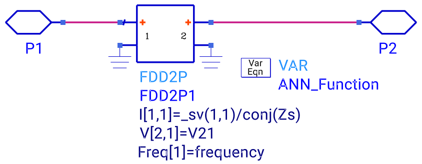

```
dsn.add_var_instance((x,y), name)
dsn.add_instance(LCVme,(x,y),name,ang)
```

Create Custom Menus, Add-Ons, GUIs



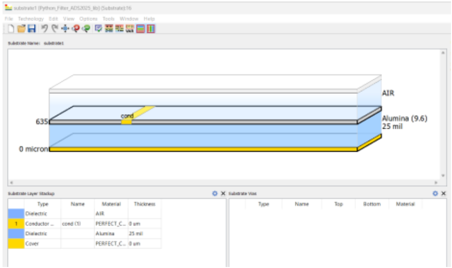

```
menu_bar.add_menu(my_menu)
```

Extract Advanced ANN Models



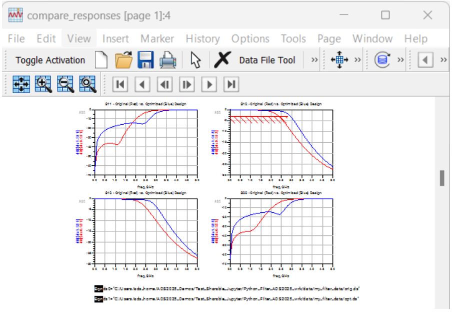

```
ann.extract_model("in.txt","out")
```

Create + Modify Substrates



```
sub=subst.create_substrate(lib,name)
material=subst.materials[ix]
```

Create + Modify Data Displays

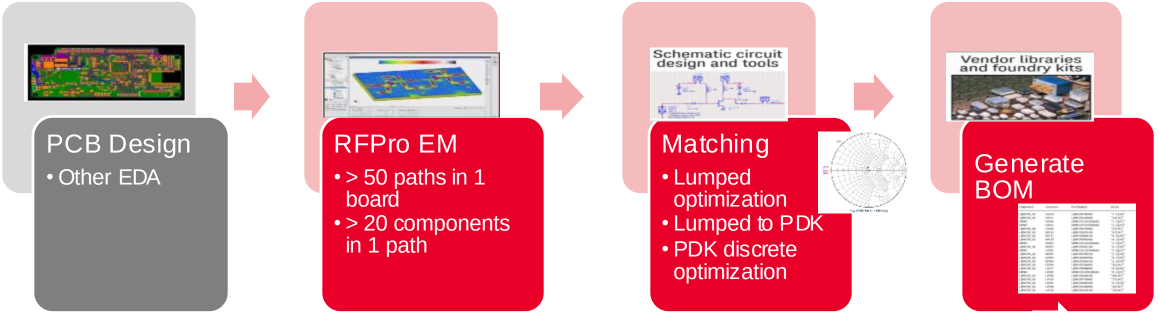



```
plot=page.add_plot()
  »traces=plot.add_traces(list[str])
```

Use Case 2 (Enterprise) : Automated Workflows (digital transformation)

Accelerate a Mobile Handset D&V workflow using ADS automation

Workflow



Time per RF Path	RFPro EM, with setup	Matching Design	Bill of Materials
Traditional workflow (manual)	60 mins	40 mins	20 mins
Automated workflow	5 mins	10 mins	10 secs
Solution approach	• Python for manual steps to import & prep for RFPro	• Faster optimization with better initial constraints • 1-button replace ideal → vendor parts	• 1-button BOM generation

Results

Results	Total Time 1 RF Path	Total Time 50 RF Paths
Traditional workflow (manual)	120 min	6000 min (100 hrs)
Automated workflow	15 min	750 min (12.5 hrs)
Solution approach	**Use automation to improve each Task within the Workflow**	

"Accelerate RF Board BOM Simulation with ADS Design Automation", DAC '61

Use Case 3 (AI/ML Exploration)

How will industry incorporate AIML into everyday design?

1. Engineering workflows are streamlined through **Automation**
2. **APIs** enable synthetic training data generation, even beyond T&M capability
3. Then, **AI Models** are developed, or AI Agents are integrated to manage.

AI Model Training Workflow

DUT
Set up the tech files
Parameterize design

GENERATE
Setup sweeps
Run simulations
Post-process to FOM's

TRAIN
Fit AI/ML models
Validate AI/ML models

DEPLOY
Metadata, versioning
Package/Encrypt model
Installer

Filter Workflow w/ AI Agents

SYNTHESIZE
Synthesize filter
AI Agent

MEASURE
Set up schematic
Run Simulation
Generate/Update Layout
Run EM Simulation
AI Agent

OPTIMIZE
Change Parameters
Iterate
AI Agent

No

DECIDE
Does the filter meet
Target spec?

Yes

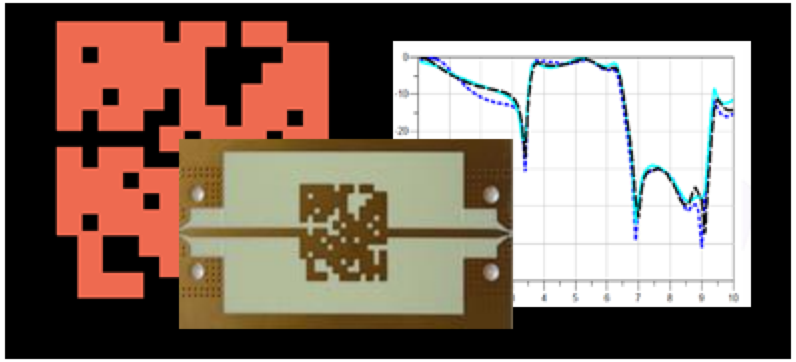
Yes

ACT
Assess Sensitivity
Generate Report
and compliance table

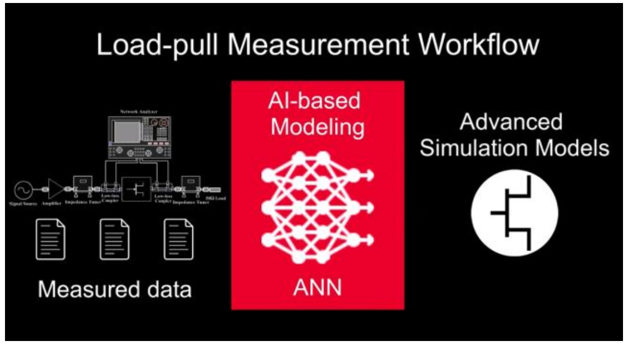
Examples of recent research

Published AI research already accumulating

Use Case 3 (A/ML Exploration) : ML enabled by ADS APIs & scalability



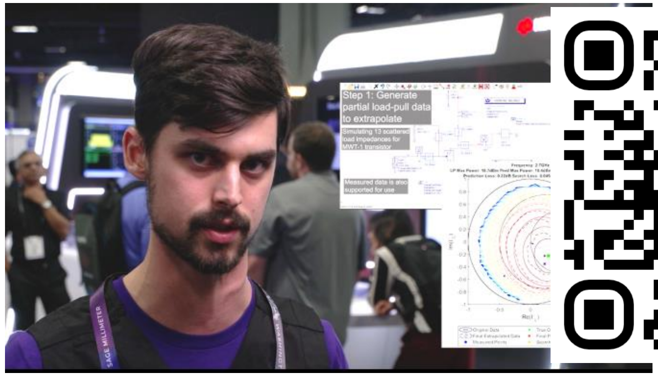
University of Chalmers automates machine learning
"Unthinkable Circuits"



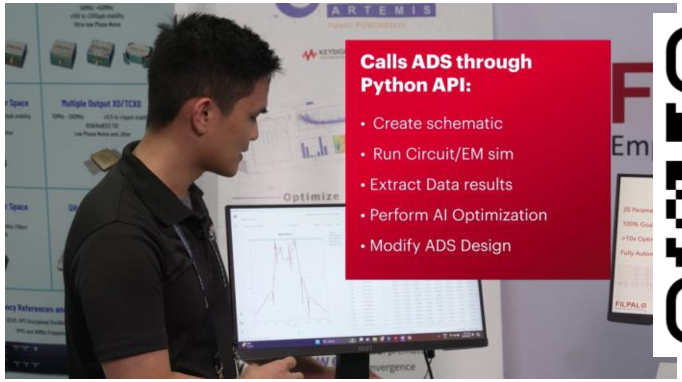
ANN for measurement-based PA workflow

17-page app note (new!)

(https://www.keysight.com/content/dam/keysight/en/vid/ungate/demos/ADS-Python-Automat



Baylor University (US) succeeds with ADS Python APIs for AI-based Loadpull research
https://www.youtube.com/watch?v=jvRouRay1jU

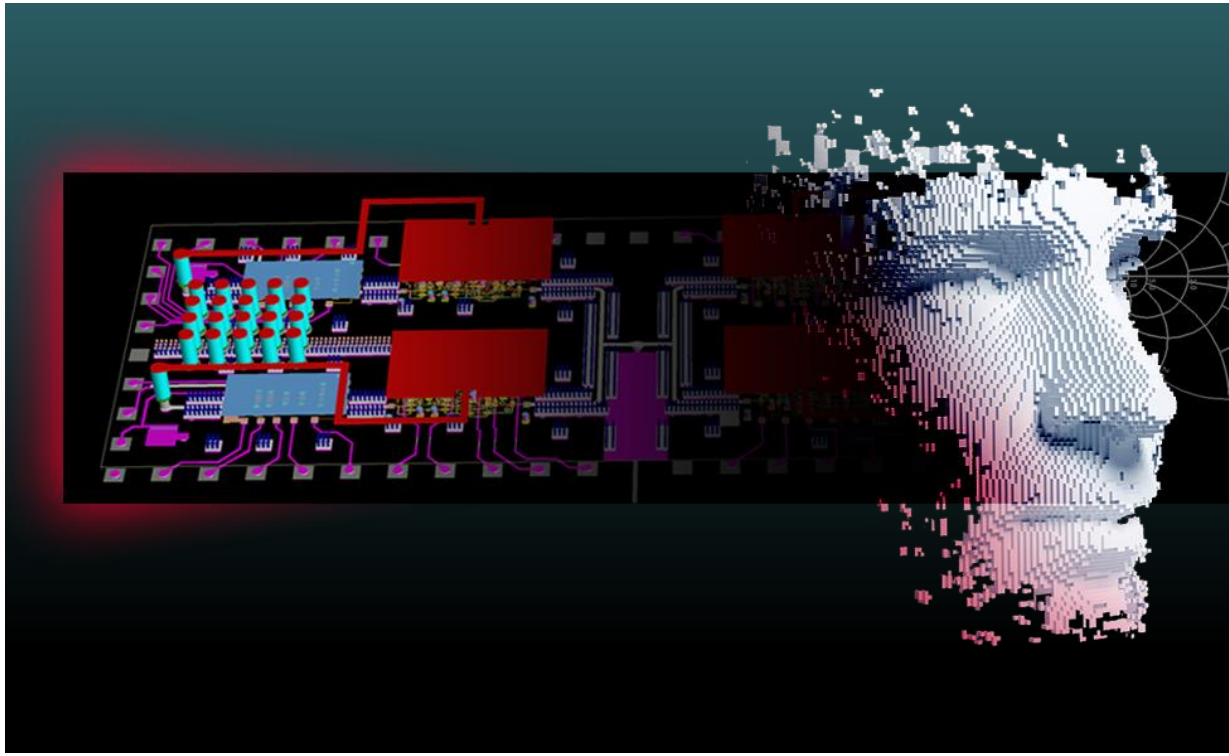


Filpal (Malaysia) startup succeeds with ADS Python APIs for AI-based Optimization research
https://www.youtube.com/watch?v=zVDZgyXg-6w

Re-imagining RF Design Using ADS Python APIs & AI/ML

Agenda

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
   (Video from Baylor Univ, 11:00)
3. Case 2: “QR Code” Filters with RFPro
   (Video from Chalmers Univ, 9:00)
4. AI/ML Technologies in ADS



KEYSIGHT

AI-Powered Multidimensional Load-Pull Extrapolation Plugin for Accelerated Computer-Aided Design (CAD) Simulations in Keysight's ADS

Jonathan E. Swindell[1], Adam C. Goad[1], Austin Egbert[1], Casey Latham[2], Matthew Ozalas[2], Andy Howard[2], Daren McClearnon[2], Charles Baylis[1], Robert J. Marks II[1]

[1]Baylor University
[2]Keysight Technologies
Charles_Baylis@baylor.edu

ADS Python Success Story: Baylor University

“Multidimensional Load-Pull Extrapolation Plugin for Accelerated Computer-Aided Design (CAD) Simulations in Keysight’s ADS

Application Summary

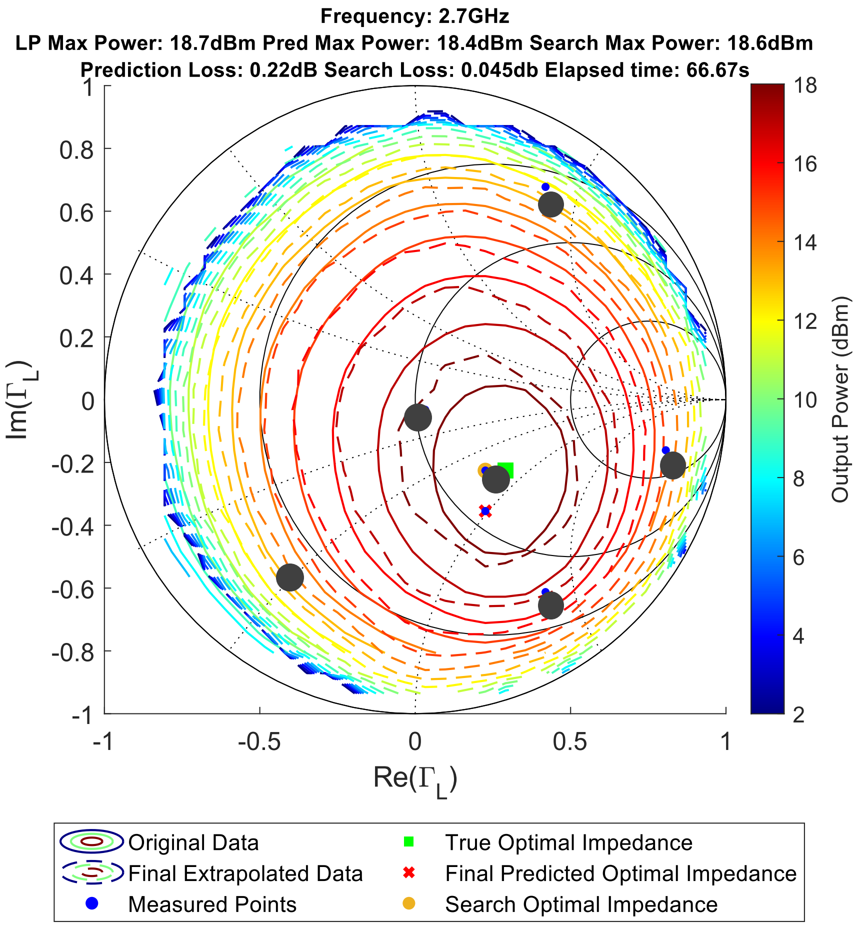
- Based on half-dozen observed Loadpull points…
- Use AI image generation techniques to estimate the rest of the curved surface, draw contours, and quickly estimate what is possible with this device

How?

- ADS is used to generate synthetic training data
- 130k Harmonic Balances of GaN device(s) vs. freq, bias, input power, and more

Conclusion

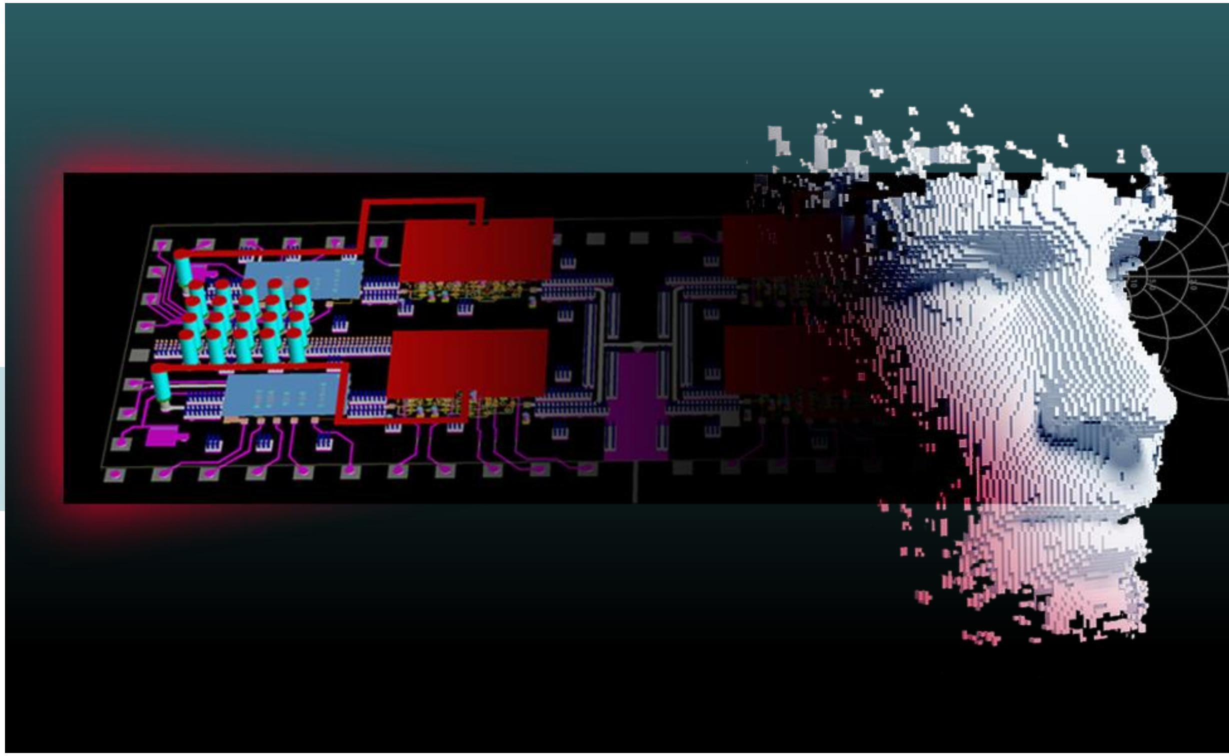
- ADS 2025 APIs were essential and easy
- AI can estimate points that are difficult/expensive to measure in real life and find solutions to multi-dimensional challenges



Re-imagining RF Design Using ADS Python APIs & AI/ML

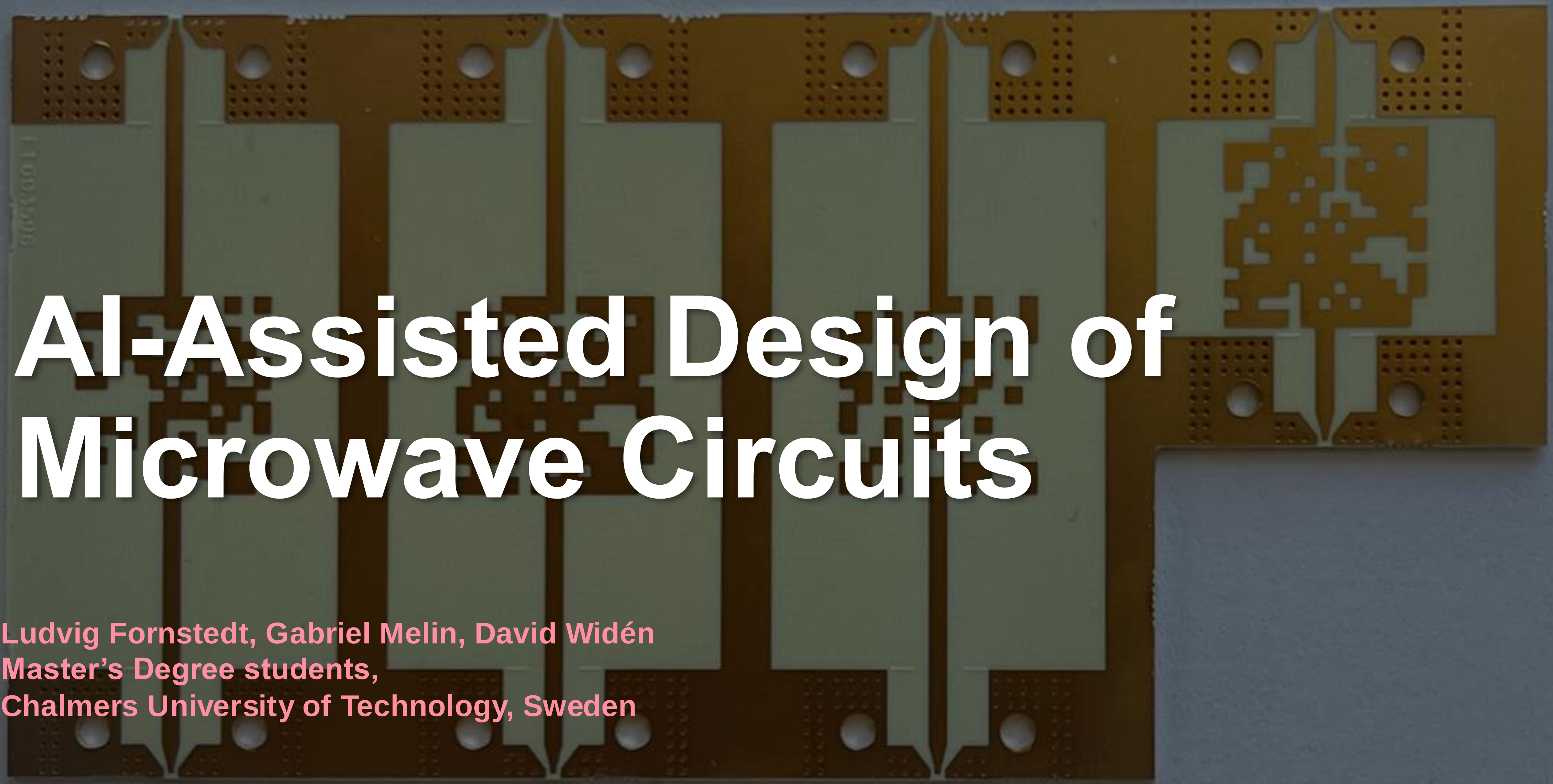
Agenda

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
   (Video from Baylor Univ, 11:00)
3. Case 2: “QR Code” Filters with RFPro
   (Video from Chalmers Univ, 9:00)
4. AI/ML Technologies in ADS



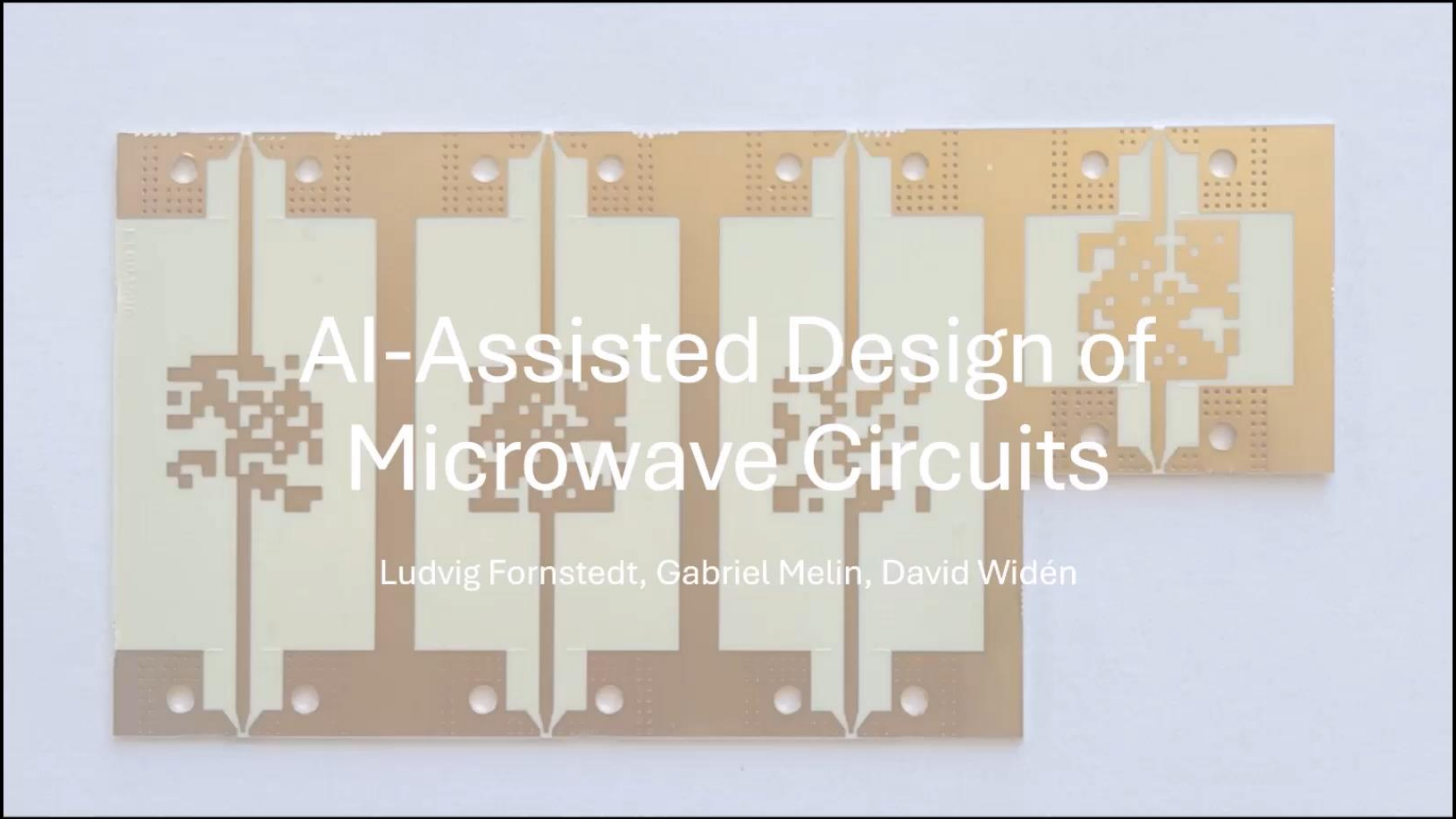
KEYSIGHT

PLAY VIDEO NEXT SLIDE



AI-Assisted Design of Microwave Circuits

Ludvig Fornstedt, Gabriel Melin, David Widén
Master’s Degree students,
Chalmers University of Technology, Sweden



Gabriel Melin

David Widén

ADS Python Success Story: Chalmers University

“AI-Assisted Design of Microwave Circuits”

Application Summary

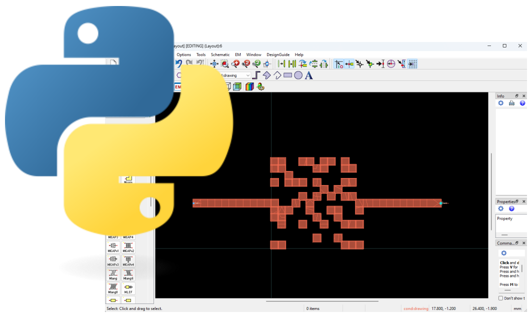
- Goal : find “Unthinkable Circuits” that humans would overlook or never consider
- Use AI/ML techniques to do inverse design of microwave filters using pixelated layouts

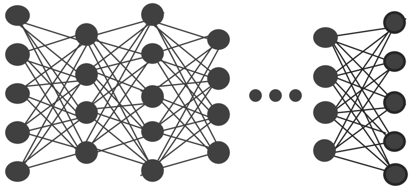
How?

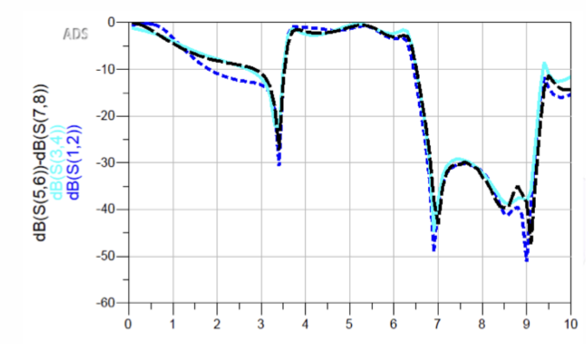
- ADS used to generate synthetic training data
- 100k EM simulations vs. freq, raster metal patterns, trained a complex neural net
- Genetic optimizer chooses closest response, using ANN to evaluate quickly

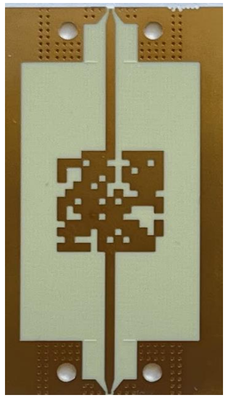
Conclusion

- ADS 2025 APIs were essential and easy
- Fast AI surrogate models with EM-level accuracy can transform design practices





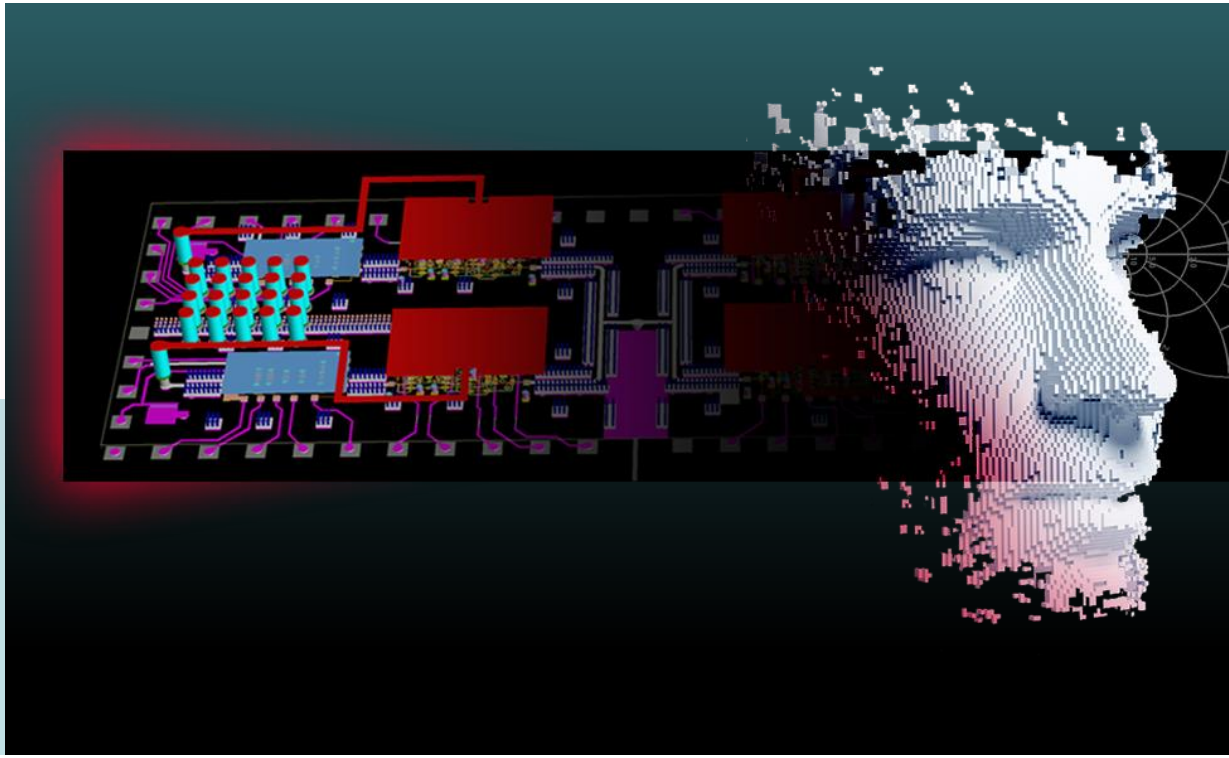




Re-imagining RF Design Using ADS Python APIs & AI/ML

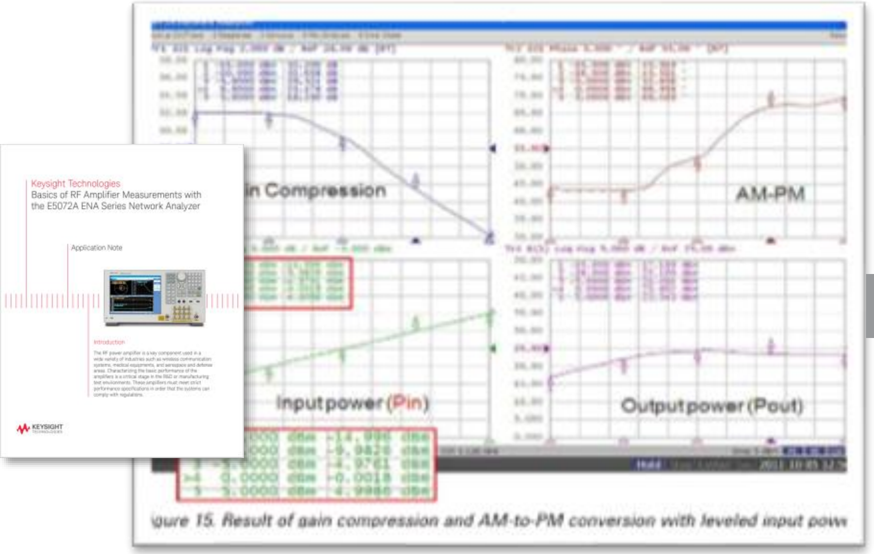
Agenda

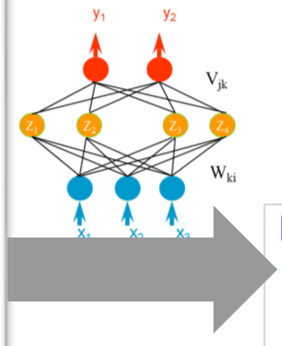
1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
3. Case 2: “QR Code” Filters with RFPro
4. **AI/ML Technologies in ADS**
   - Artificial Neural Net modeling (ANN)
     Putting static data into motion
   - ML Optimization
     Letting machines do the hard work
   - Chatbots & Generative AI
     Intelligent assistants arrive in ADS



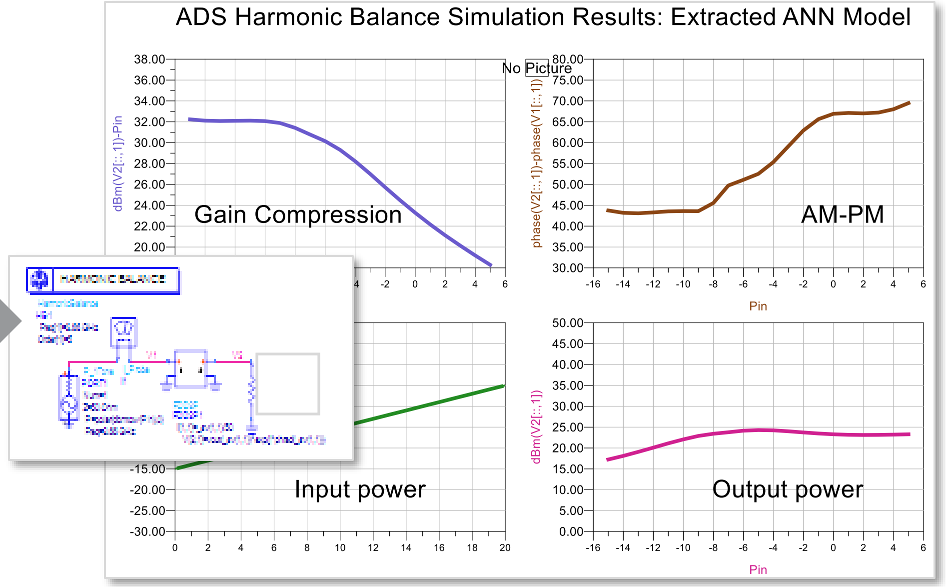
Neural Nets – ANN’s can turn static data into executable models

From pictures on a datasheet…





…to nonlinear simulation!



Turn data into executable models

- Smoother, continuous interpolation from sparse, irregular, or multi-dimensional data
- Faster execution of behavioral mdoel
- Protects circuit and measurement IP
- Built into ADS 2025

General-purpose video tutorial series:

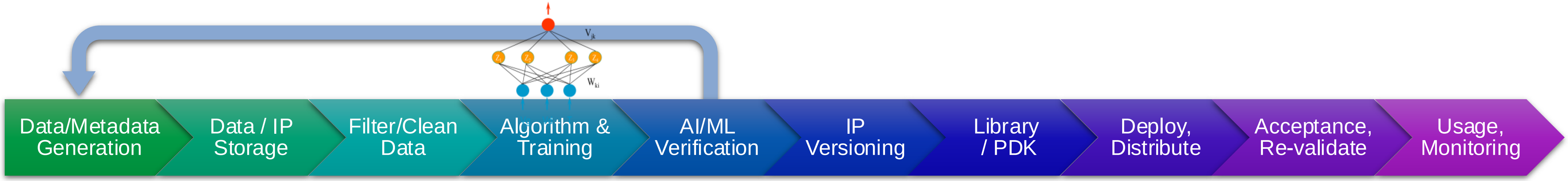
“Using AI to Create a Simulation Model”

https://www.youtube.com/playlist?list=PLtq84kH8xZ9G8PkPdGMAAwxM1Y6dZ6oIC

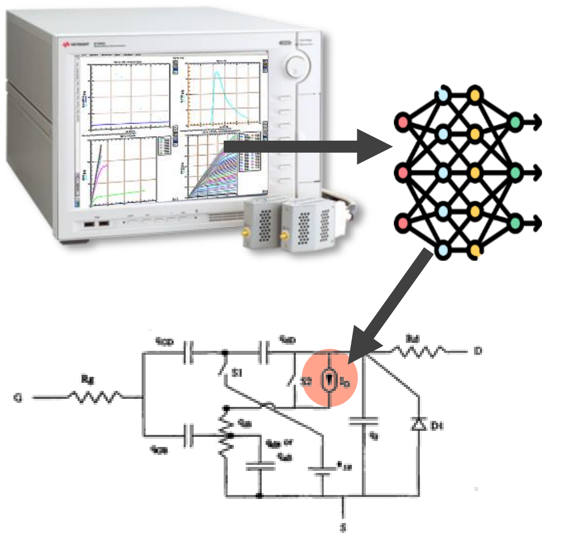
- Part 1: https://www.youtube.com/watch?v=IYQ3b4YUQFM
- Part 2: https://www.youtube.com/watch?v=mWkHV1wfzFs
- Part 3: https://www.youtube.com/watch?v=ylrwYq5b_r8

Neural Net Modeling Lifecycle

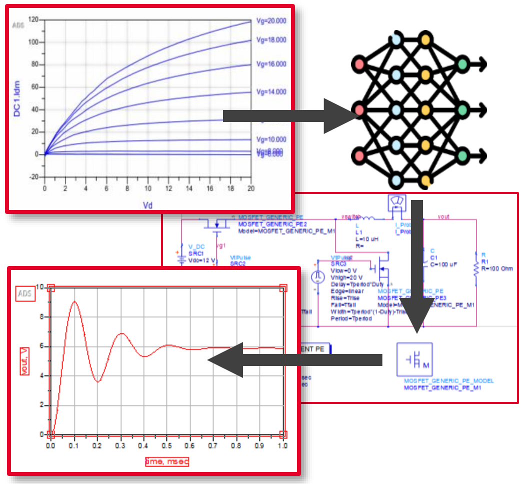
ANN modeling can be convenient, and yet strategically important at the department level



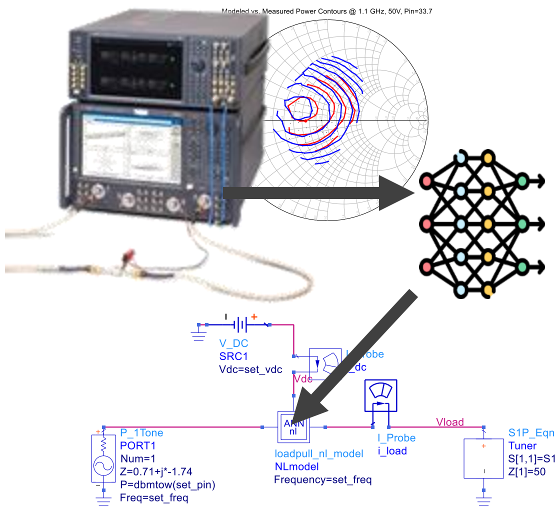
ICCAP ANN



Power Electronics ANN

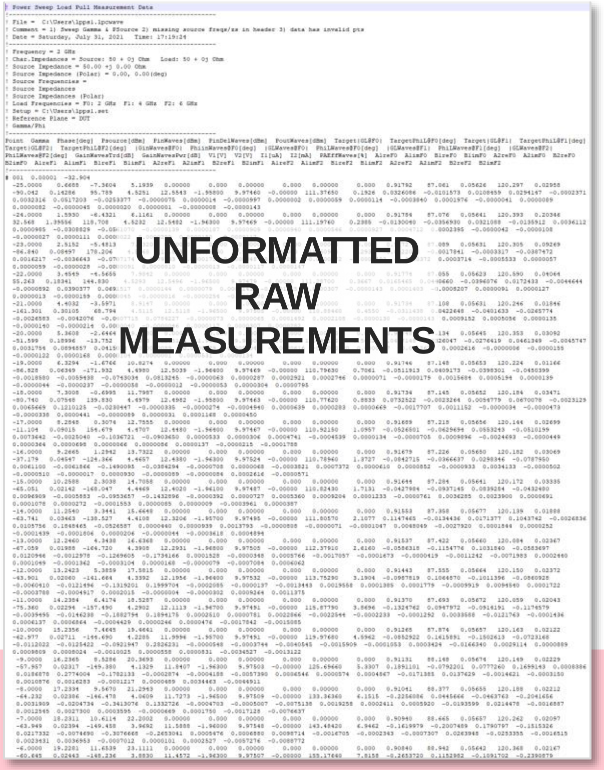


5G PA Load Pull ANN



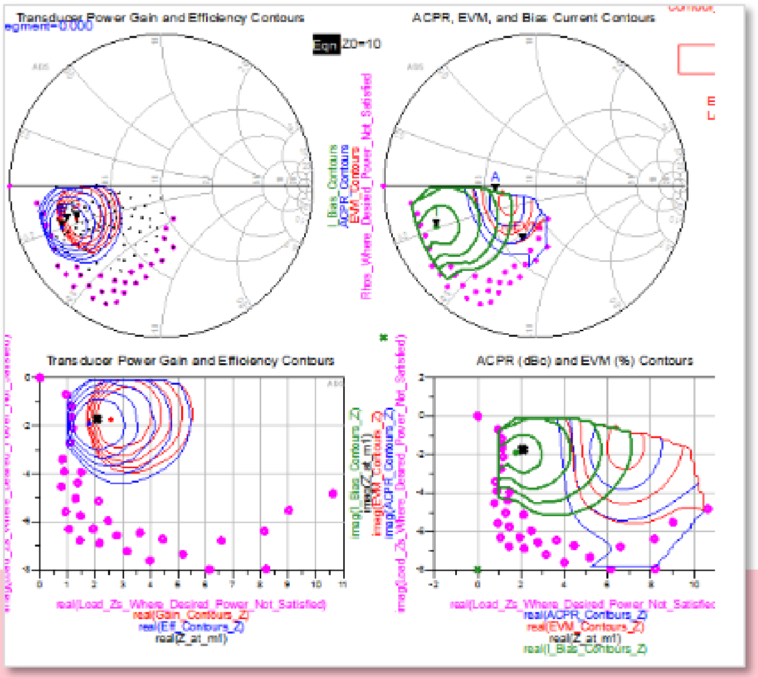
Let’s Apply ANNs to Loadpull PA Design – Problem Statement

Original Loadpull dataset. How to use it?

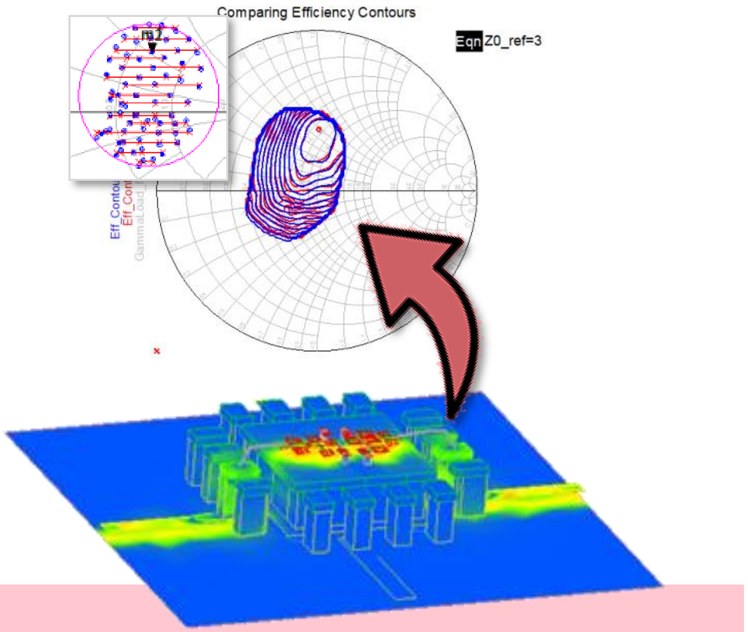


1 Plot Contours

2 Pick the “Best” Complex Load



3 Finish the design in ADS

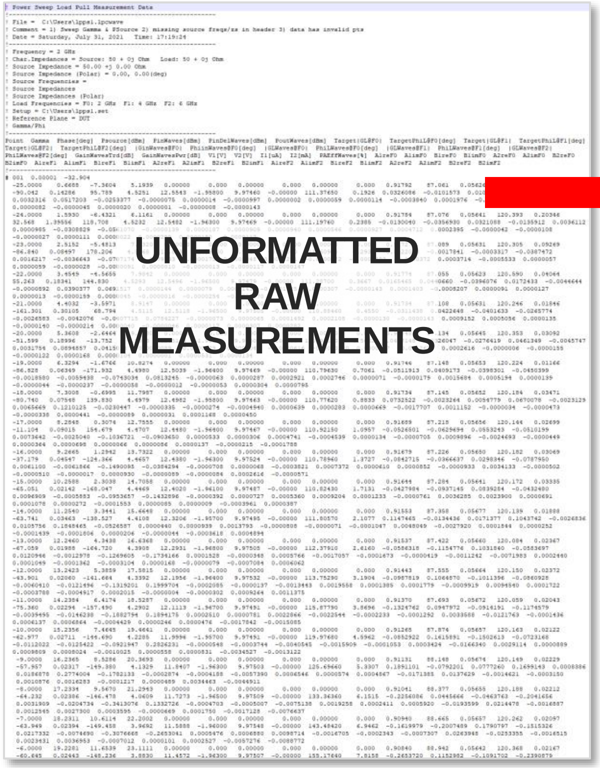


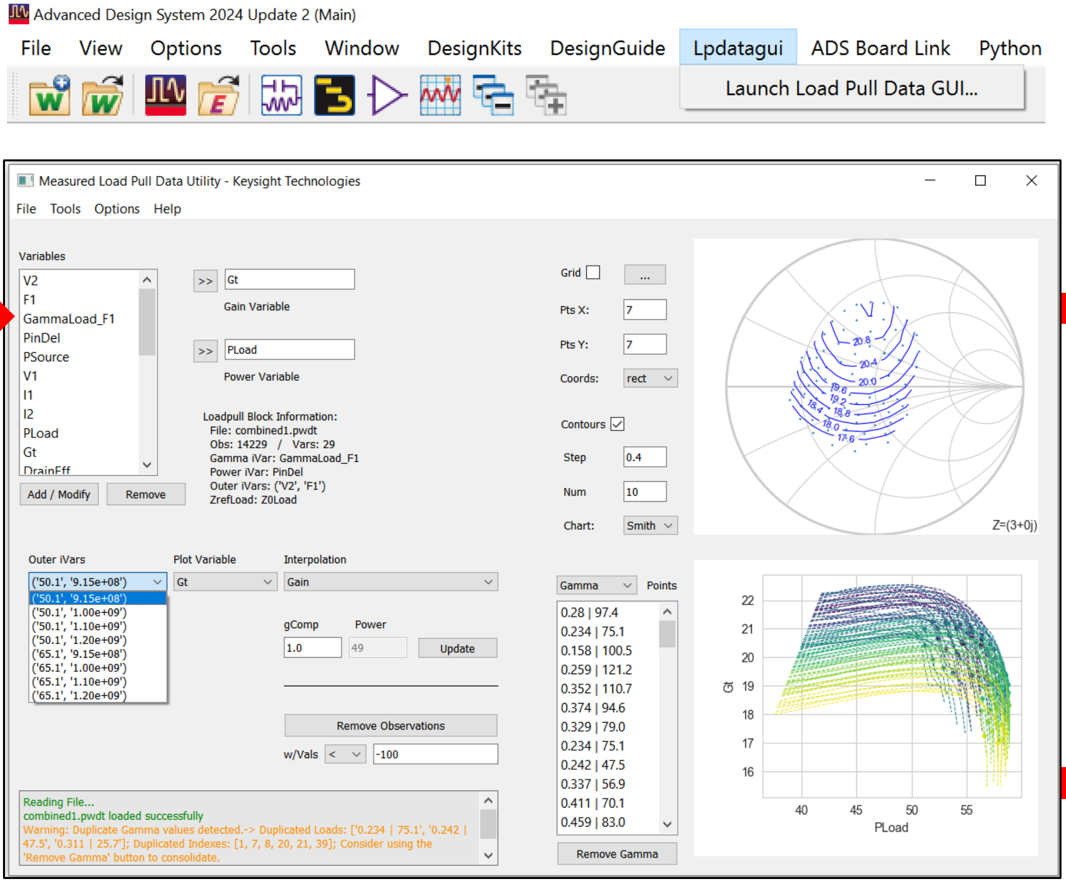
Challenges:

“How to clean up the measured data?” → Python Automation

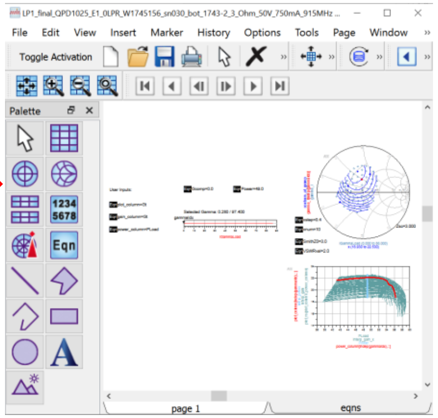
“How to turn the data into an executable model?” → ANNs

Let’s Apply ANNs to Load Pull PA Design – Solution

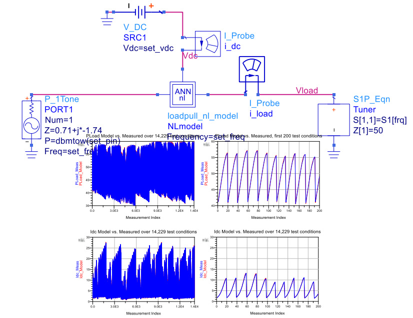




ADS Datasets and Data Displays



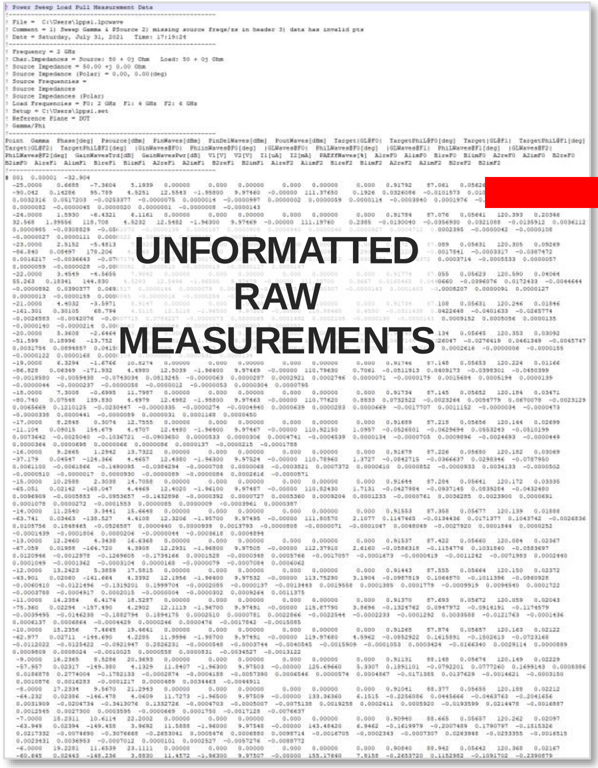
ADS Models and Testbenches

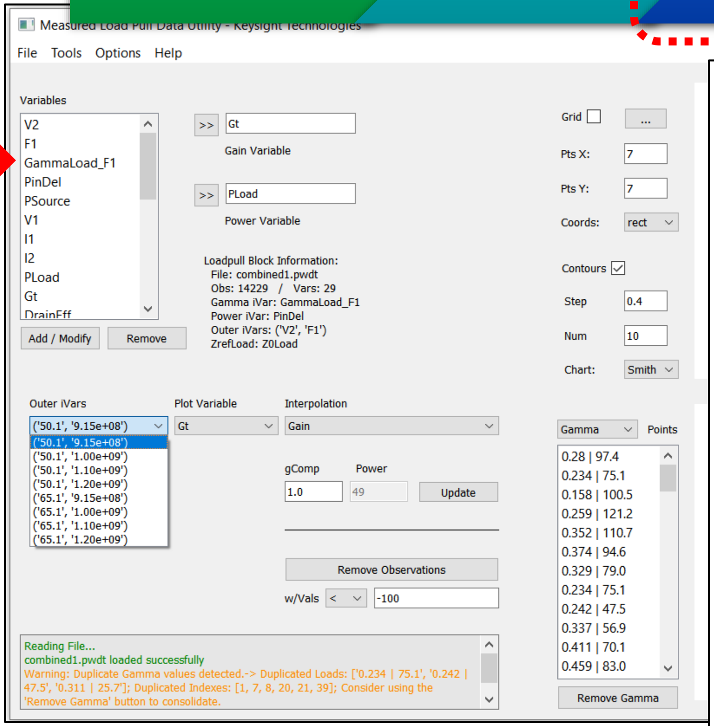


IMPORT DATA
CLEAN IT UP
EXPORT IT

https://docs.keysight.com/pages/viewpage.action?spaceKey=eesofkcads&title=Load+Pull+Data+GUI+for+ADS

Let’s Apply ANNs to Load Pull PA Design – Solution



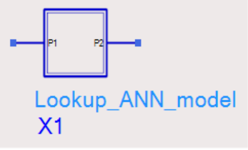


3 LEVEL OF ANN MODELS

“Lookup” model

Optimize

Train on raw data

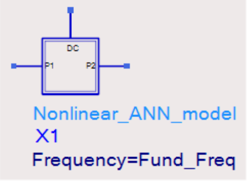


Fast, efficient matching network optimization, great at interpolation.

“Tone” model

Emulate

Train on AMAM / AMPM

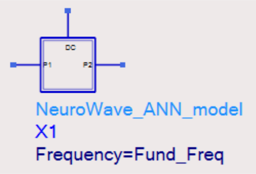


Simple, tone based nonlinear model for basic circuit and system analysis.

“Wave” model

Design

Train on A / B wave data

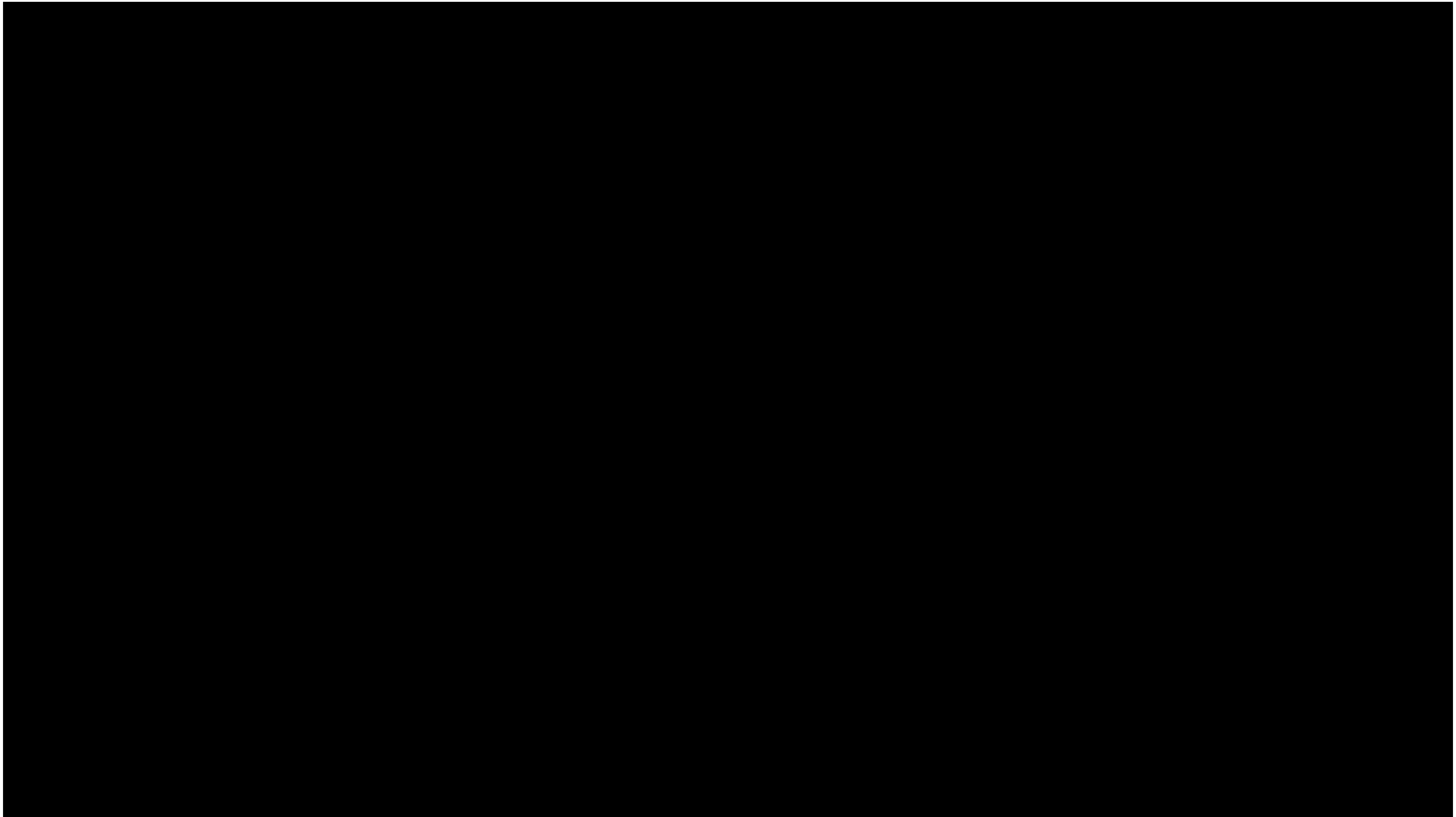


Design-capable nonlinear model; can use to build a full PA (within meas. limits).

Watch now:

https://www.keysight.com/content/dam/keysight/en/vid/ungate/demos/Load-Pull-Modeling-with-Python-s-AI-Power.mp4

OPTIONAL : Play Video Demo of Loadpull DataGUI (5:12)



For More Information on Loadpull and to download binaries

Application note & video on this Loadpull topic

- https://www.keysight.com/us/en/assets/3124-1550/application-notes/Measurement-Based-Artificial-Neural-Network-Simulation-Models-for-RF-Power-Amplifiers.pdf
- https://www.keysight.com/content/dam/keysight/en/vid/ungate/demos/Load-Pull-Modeling-with-Python-s-AI-Power.mp4

To access the ADS workspace and Python scripts from this Application Note, plus additional Jupyter Notebooks and schematic creation scripts:

- https://docs.keysight.com/display/eesofkcads/Measurement-based+Artificial+Neural+Network+Simulation+Models+for+RF+Power+Amplifiers
  (Keysight Knowledge Center)

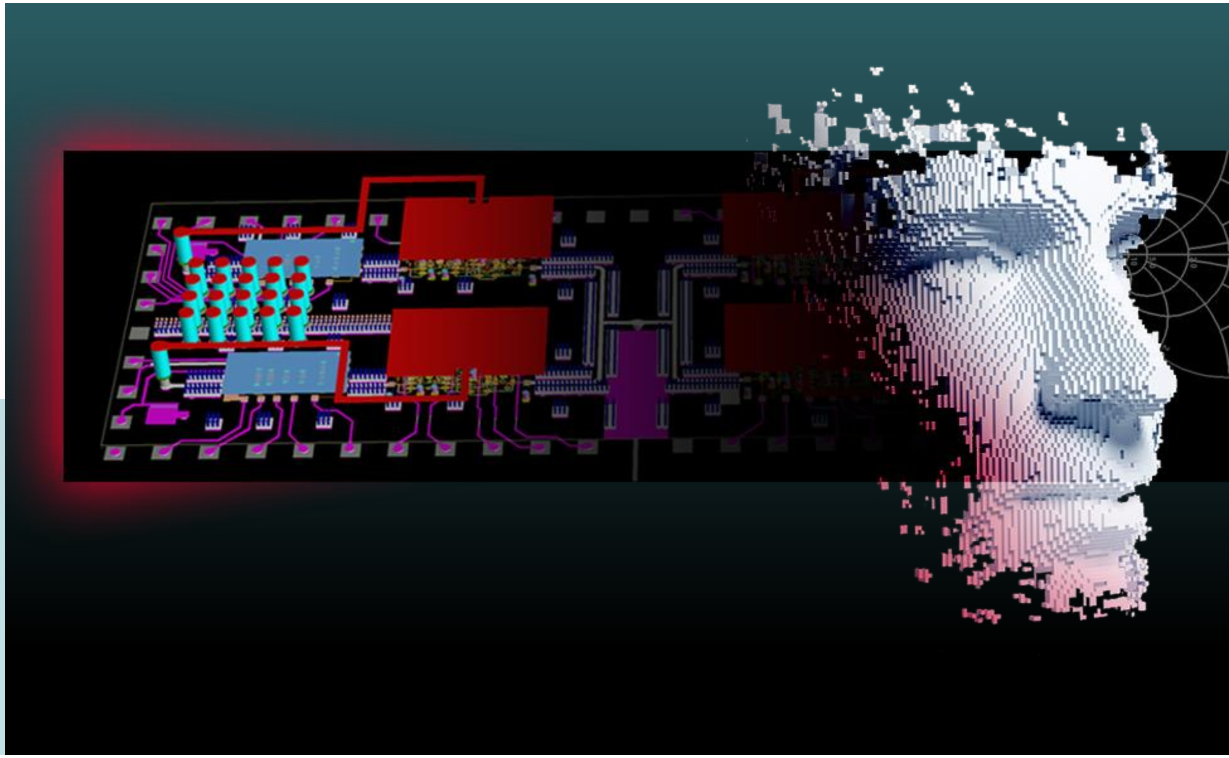
To download a Python based add-on for ADS which reads measured load pull data and generates models similar to the ones described in this work:

- https://docs.keysight.com/display/eesofkcads/Load+Pull+Data+GUI+for+ADS
  (Keysight Knowledge Center)

Re-imagining RF Design Using ADS Python APIs & AI/ML

Agenda

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
3. Case 2: “QR Code” Filters with RFPro
4. **AI/ML Technologies in ADS**
   - Artificial Neural Net modeling (ANN)
     Putting static data into motion
   - ML Optimization
     Letting machines do the hard work
   - Chatbots & Generative AI
     Intelligent assistants arrive in ADS



ML Optimization – a new generation of algorithms

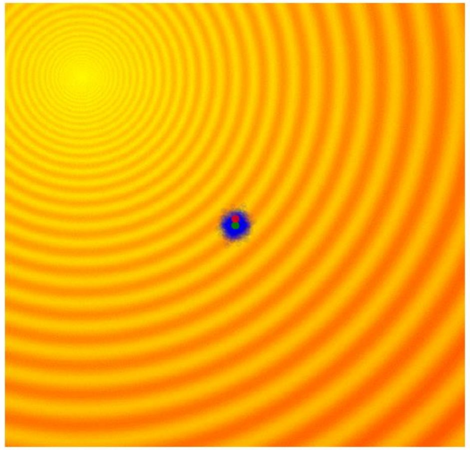
Breakthrough techniques unlock a proven concept

Challenge

- Conventional optimizers struggle when handling more than 20 variables and/or objectives/goals
- Result: Users need to work around limitations.
  → Manually break into many smaller problems

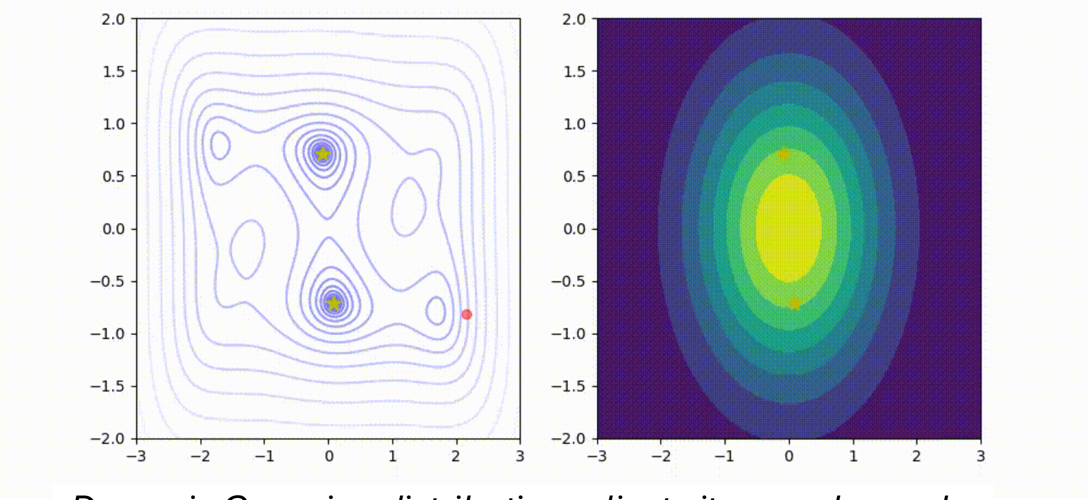
Solution

- Use new AI/ML-driven optimization techniques:
- Expanded capacity to handle 40+ parameters, multi-objectives
- Able to automate complex processes, accelerating workflow



Older fixed Gaussian distributions try within a circle of predictions

AI/ML approaches are smart and adaptive, supporting higher optimization dimensions



Dynamic Gaussian distribution adjusts its search area by learning from the data, giving oval shaped predictions

ML Optimization – a new generation of algorithms

Abilities of Modern Optimizers:

✓ Large number of input Variables, and output Goals

✓ Fewer iterations needed to converge, given complex problems

Optimizers especially shine when:

- There is no historical data —– cannot apply supervised/unsupervised Machine Learning algorithms
- Limited number of iterations (less than 10K) —– cannot apply Reinforcement learning

Recently developed optimization algorithms:

- CMA-ES, LSHADE, LSHADE-SPACMA, TSA, TuRBO
- Particle Swarm, Corana Simulated Annealing, Non Dominated Sorted Genetic Algorithm II (NSGA-II), more
- Ability to integrate your own optimizer from PyTorch, Pymoo, etc

IC-CAP 2025 ML Optimizer

< 10 Parameters → with ML → 40 + Parameters

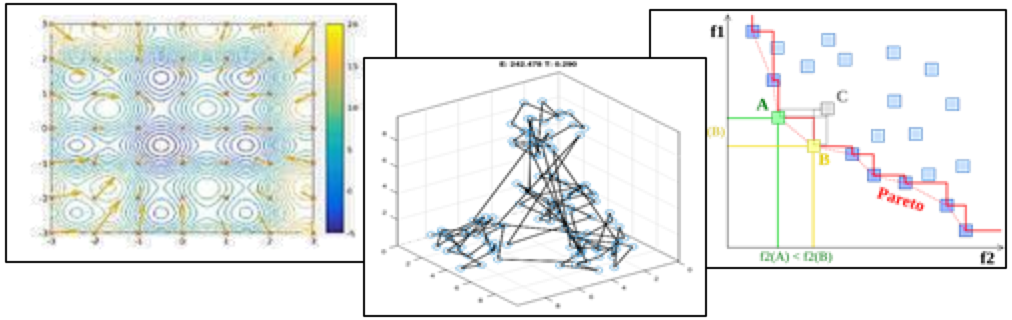
Zijian Song et al.,
"Automating Compact Model Parameter Extraction: Industry-Grade Implementation and Benchmarking of Machine Learning-Driven Derivative-Free Algorithms", 2024 IEEE NAFEMS

ADS 2024 Worst-Case Bit-Pattern Generator for Eye Diagram

14x Simulation Speed Up

Majid Ahadi Dolatsara,
"Worst-Case Bit-Pattern Generator for Eye Diagram Analysis of Non-LTI High-Speed Channels", 2024 Signal Integrity Journal

Corana
PSO
StoGO
CRS
LBFGS
COBYLA
SLSQP
TNewton
BOBYQA
SBPLX
Python
Random



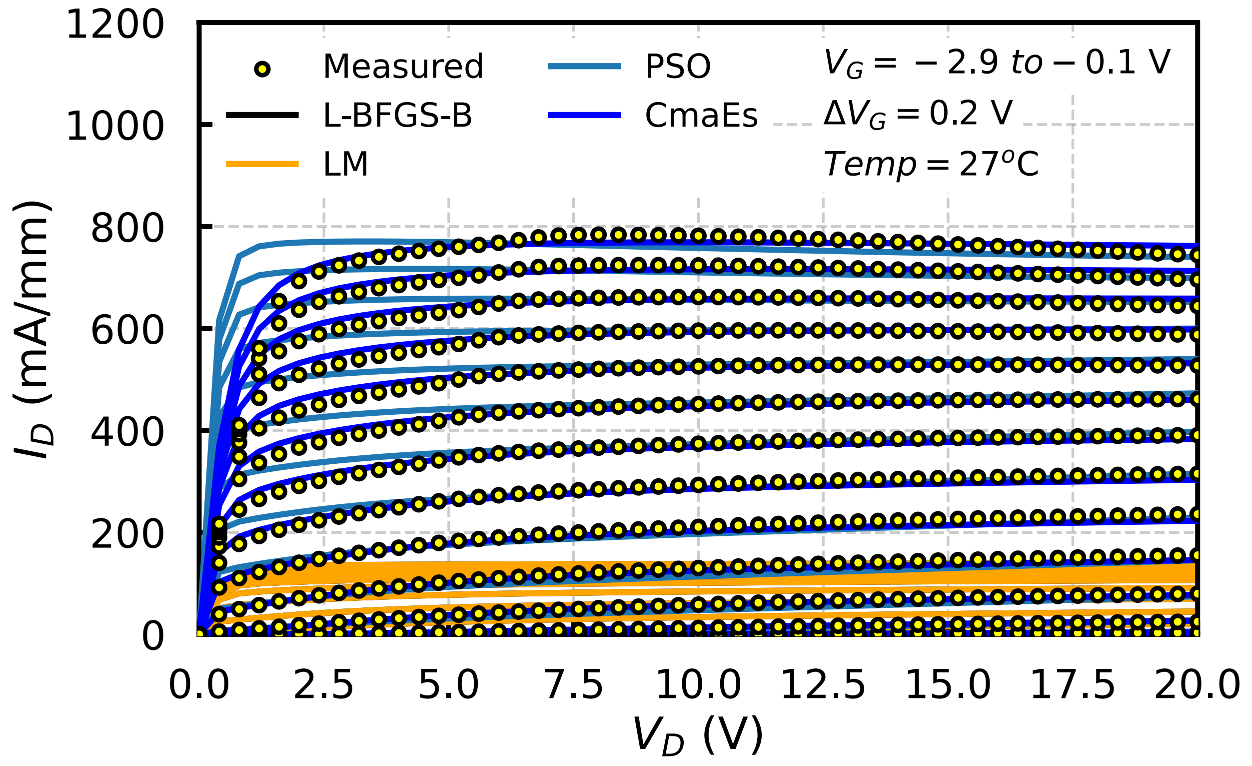
Graphics credit: Wikipedia

ML Optimization results : Adjusting 27 Parameters in ICCAP

Improved workflow for complex task

Analyzed and compared simulated vs. measured GaN-on-SiC HEMT $I_D - V_D$ characteristics

- CMA-ES achieved the best fitting performance
- L-BFGS-B and LM results overlapped due to similar fitting performances

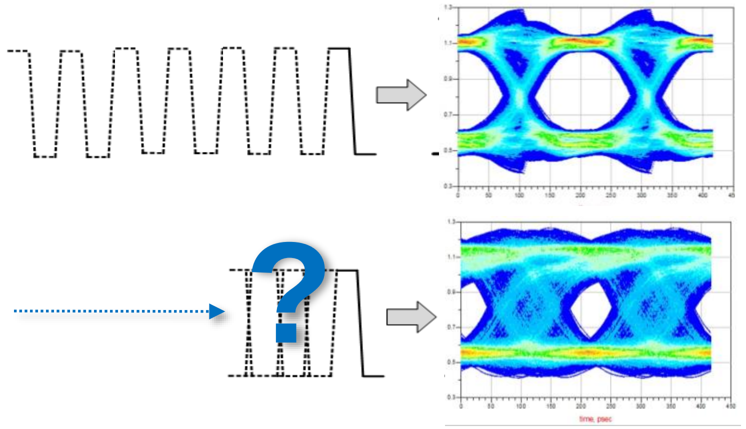


Implication for RF Circuit design:

- faster Optimizer convergence is coming soon on realistic circuits with many parameters

ML Optimization – Applied to Complex Verifications

Worst-case Eye Pattern for HSD/Signal Integrity applications

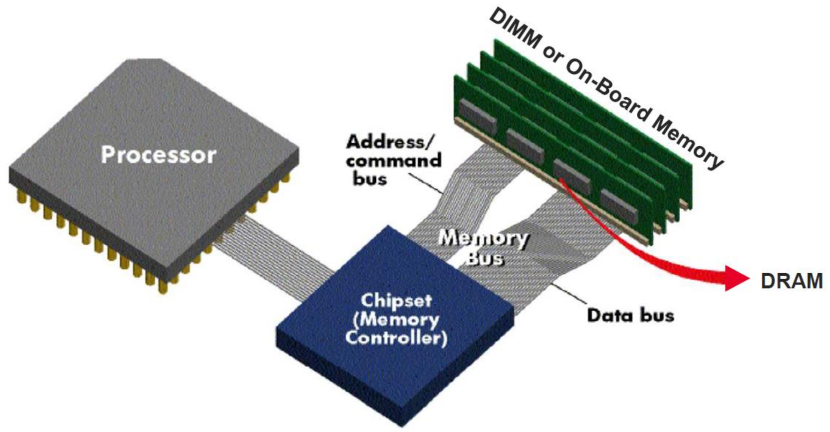


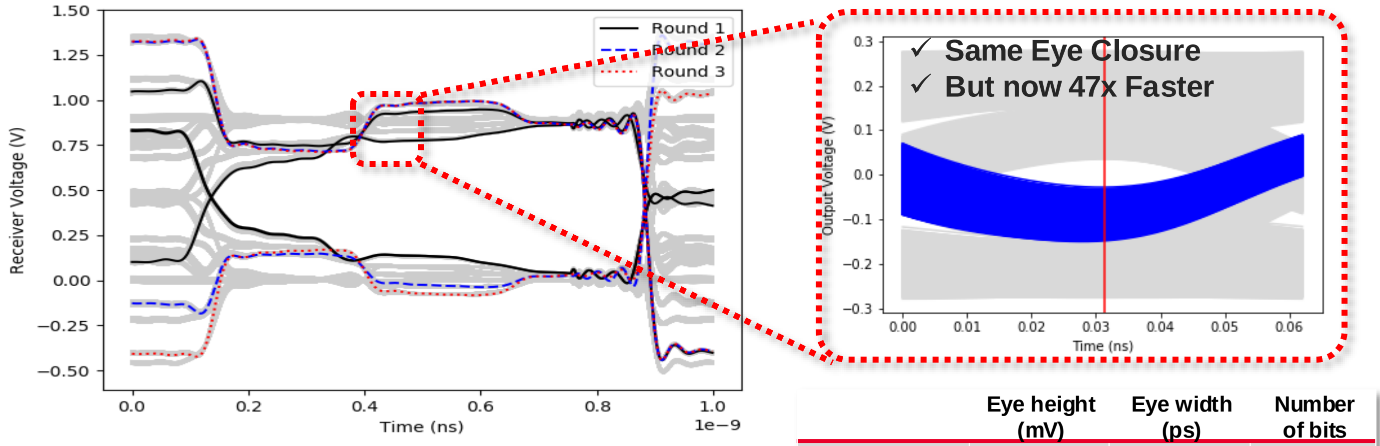
Bayesian Optimization toward a maximal aggressor waveform

- Finds the worst-case Eye Diagram, using << shorter bit sequence, in the fewest # of Transient search simulations
- Result: Dramatic simulation speed-ups with optimized waveform

Bayesian optimizer now added to ADS

Bayesian optimization for the Equalizers to clean up Eye Closure



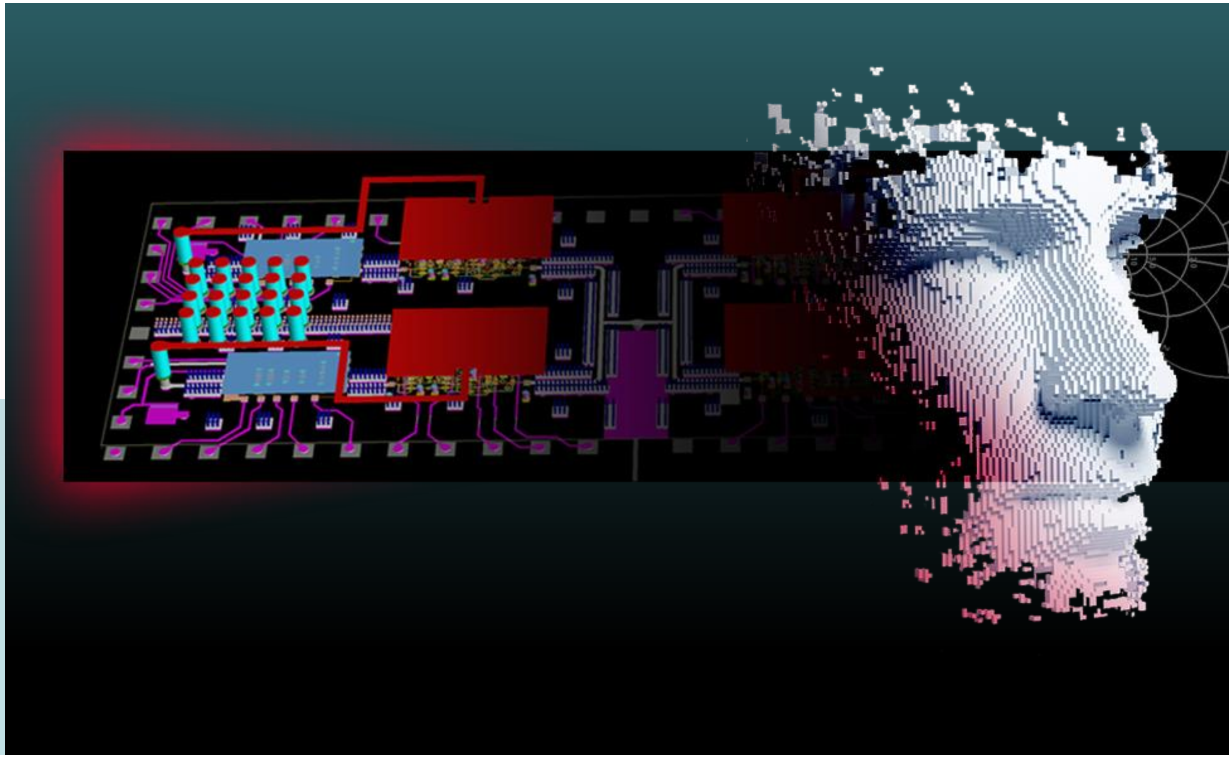


	Eye height (mV)	Eye width (ps)	Number of bits
Worst-eye	639	745	5,760
Transient Eye	639	744	1,000,000

Re-imagining RF Design Using ADS Python APIs & AI/ML

Agenda

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
3. Case 2: “QR Code” Filters with RFPro
4. **AI/ML Technologies in ADS**
   - Artificial Neural Net modeling (ANN)
     Putting static data into motion
   - ML Optimization
     Letting machines do the hard work
   - Chatbots & Generative AI
     Intelligent assistants arrive in ADS



Introducing Keysight’s EDA Chat

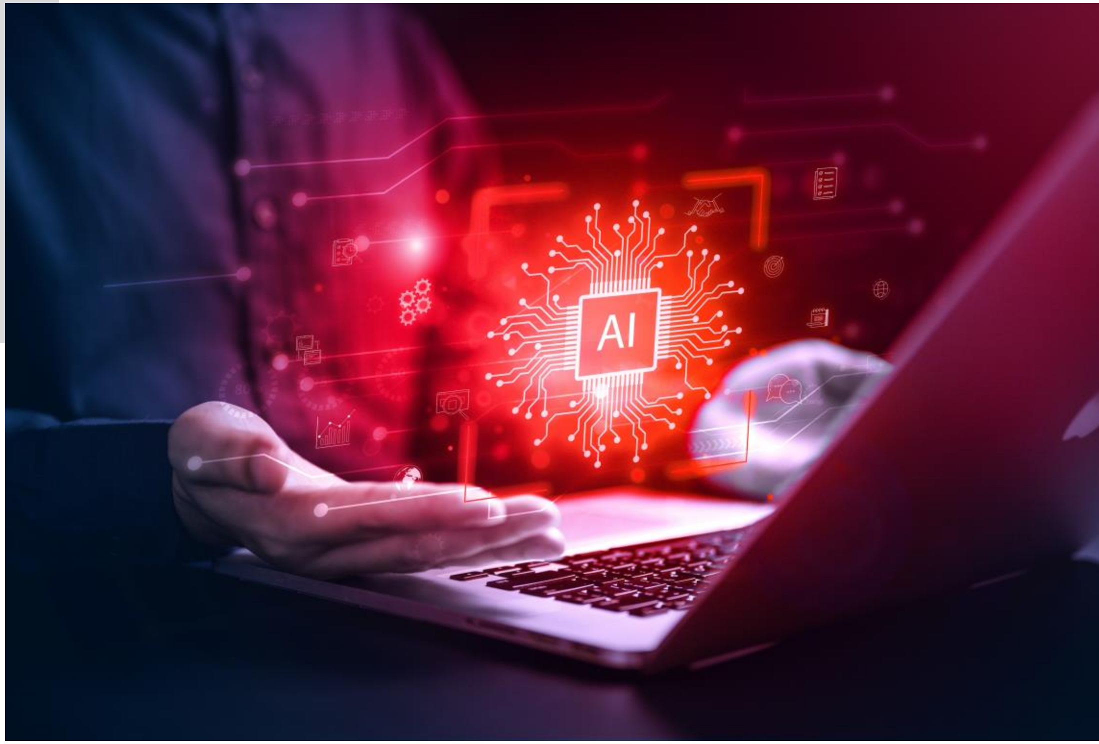
1000’s of pages of Documentation and KC documents at your fingertips

NEW ADS USER
Ask chat question

EDA CHATBOT
Learned best tricks
from 100s of Keysight Experts,
Documents, resources

NEW ADS USER
Faster success
on real design tasks

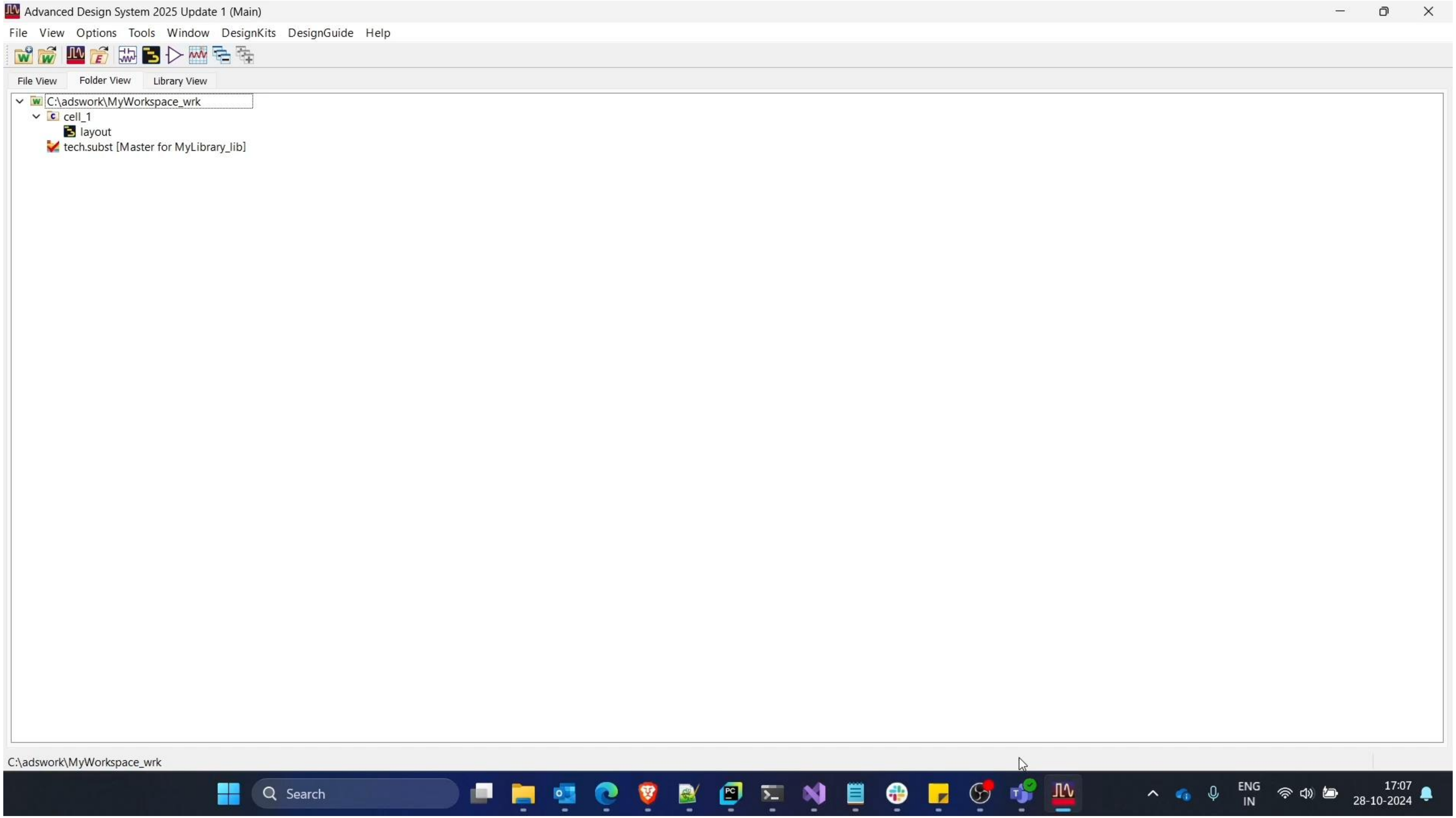
Succeed faster,
beginning with ADS 2025 Update 2



Benefits
- Natural language responses
- Hyperlinks to original ADS Doc & KC references
- Avoids obsolete and conflicting product info
- ADS only : Not a general engineering reference or design agent

Productivity
- <u>New users</u>: onboard quickly with ADS
- <u>Experienced users</u>: find detailed references easily

Keysight EDA Chat – demo preview

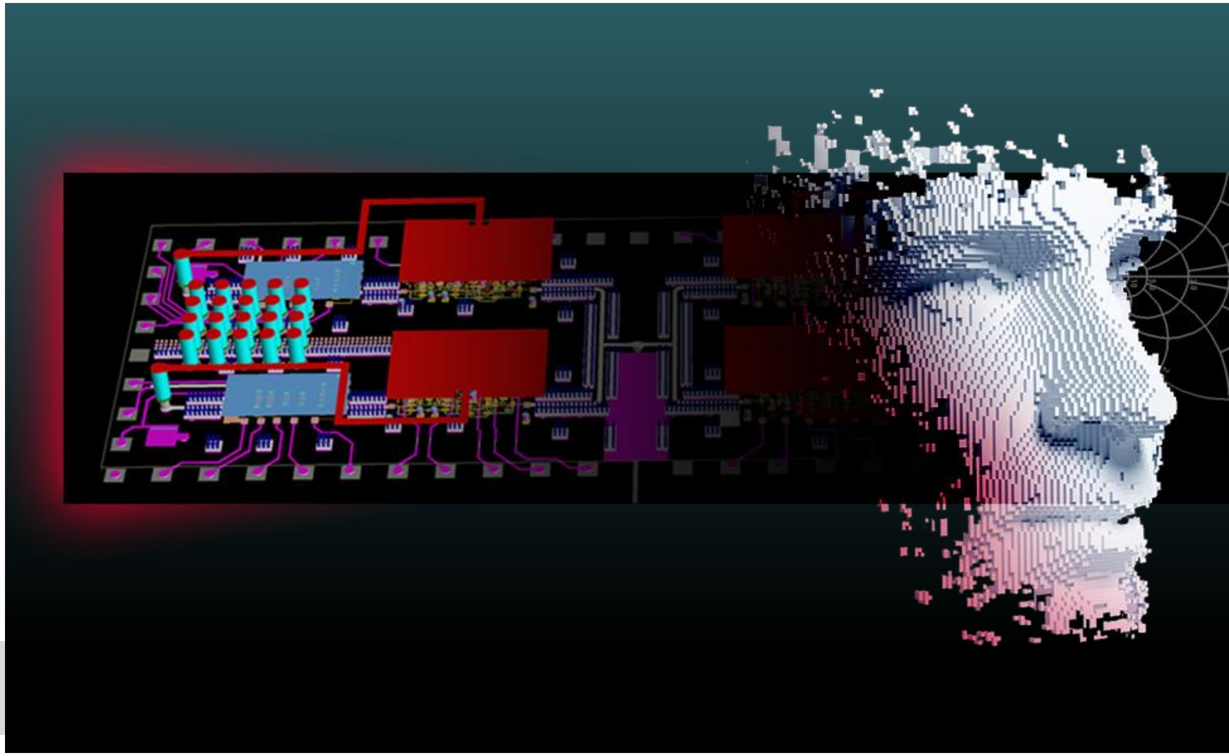


Re-imagining RF Design Using ADS Python APIs & AI/ML

Conclusion

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
   (Video from Baylor Univ)
3. Case 2: "QR Code" Filters with RFPro
   (Video from Chalmers Univ)
4. AI/ML Technologies in ADS

Conclusion



Conclusion

✓You can achieve more through Automation

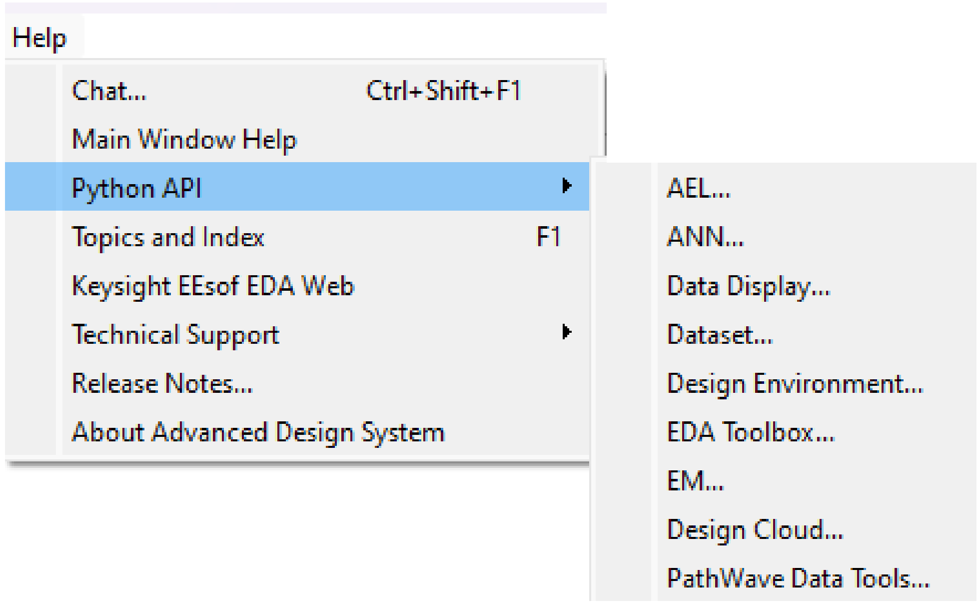
- Python is an easy, universal language, and ADS APIs make automation easier than ever.
- Multiple interfaces : Command line inside ADS, headless from outside, or custom Apps
- Microsoft VS Code and Copilot are remarkably helpful
- Anyone can do it!

✓Automation is the on-ramp to AI/ML

- A new wave of technology is coming to assist designers
- Designers who use AI will out-produce Designers who don’t
- Invest now to enhance your skills

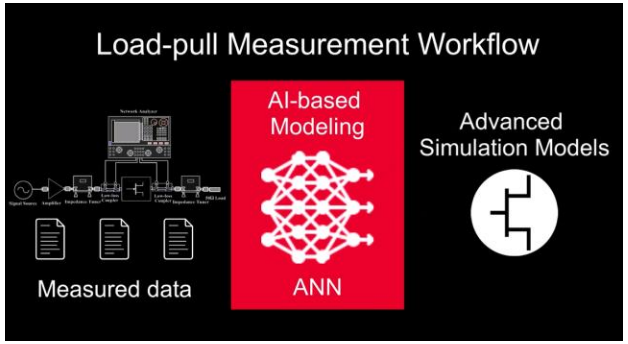
✓Open, standards-based workflows

- AI technology is moving quickly. Open standards reduce risk.
- Keysight is working toward Open, Secure, and Scalable workflows



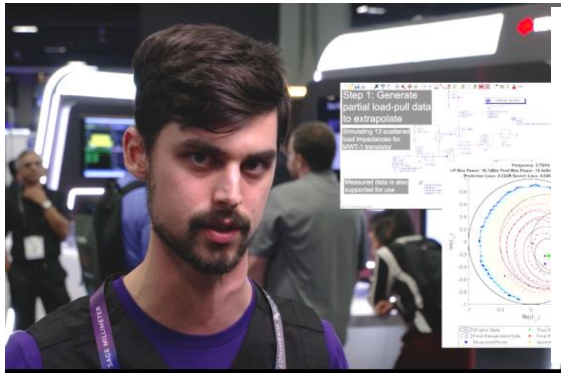
Learn more about ADS APIs & Python Automation

Whitepaper outlining what’s possible with ADS python

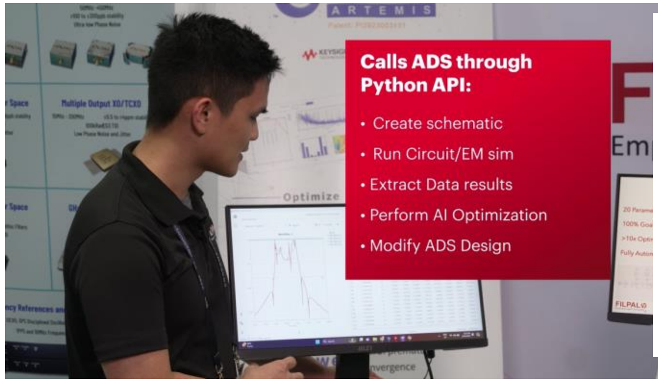


ANN for measurement-based PA workflow
(Video coming soon)

17-page app note (new!)



Baylor University (US) succeeds with ADS Python APIs for AI-based Loadpull research
https://www.youtube.com/watch?v=jvRouRay1jU



Filpal (Malaysia) startup succeeds with ADS Python APIs for AI-based Optimization research
https://www.youtube.com/watch?v=zVDZgyXg-6w

Learn more about ADS APIs & Python Automation

Python Tutorials and resources

ADS 2025 U1 Python API documentation

- https://docs.keysight.com/ads2025update1/ads-python-api-documentation

ADS Python Getting Started (resource page)

- https://docs.keysight.com/display/eesofkcads/ADS+Good+Reference+Starts+for+AEL+and+Python+Scripting+or+Command+Line

To learn more about PathWave Data Tools Python Library (v. 0.11.0):

- https://docs.keysight.com/pwdt0x11x0 (keysight.com)

To learn more about Python automation in Keysight Advanced Design System:

- Python APIs for EDA Workflows (keysight.com)

To learn about **Keysight EDA Chat**

- https://docs.keysight.com/display/keysightedachat/About+Keysight+EDA+Chat

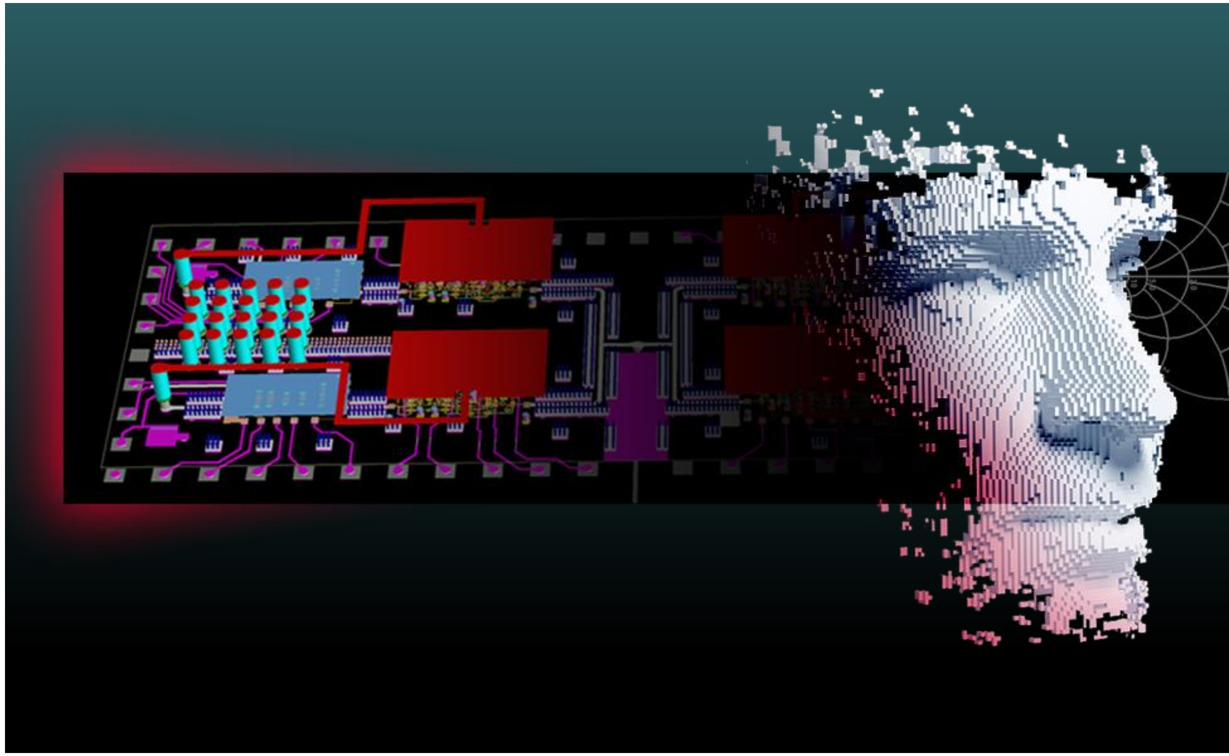
Questions?

KEYSIGHT

Re-imagining RF Design Using ADS Python APIs & AI/ML

Agenda

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
   (Video from Baylor Univ)
3. Case 2: "QR Code" Filters with RFPro
   (Video from Chalmers Univ)
4. AI/ML Technologies in ADS



KEYSIGHT

PLAY VIDEO

Multidimensional Load-Pull Extrapolation Plugin for Accelerated Computer-Aided Design (CAD) Simulations in Keysight's ADS

Jonathan E. Swindell[1], Adam C. Goad[1], Austin Egbert[1], Casey Latham[2], Matthew Ozalas[2], Andy Howard[2], Daren McClearnon[2], Charles Baylis[1], Robert J. Marks II[1]

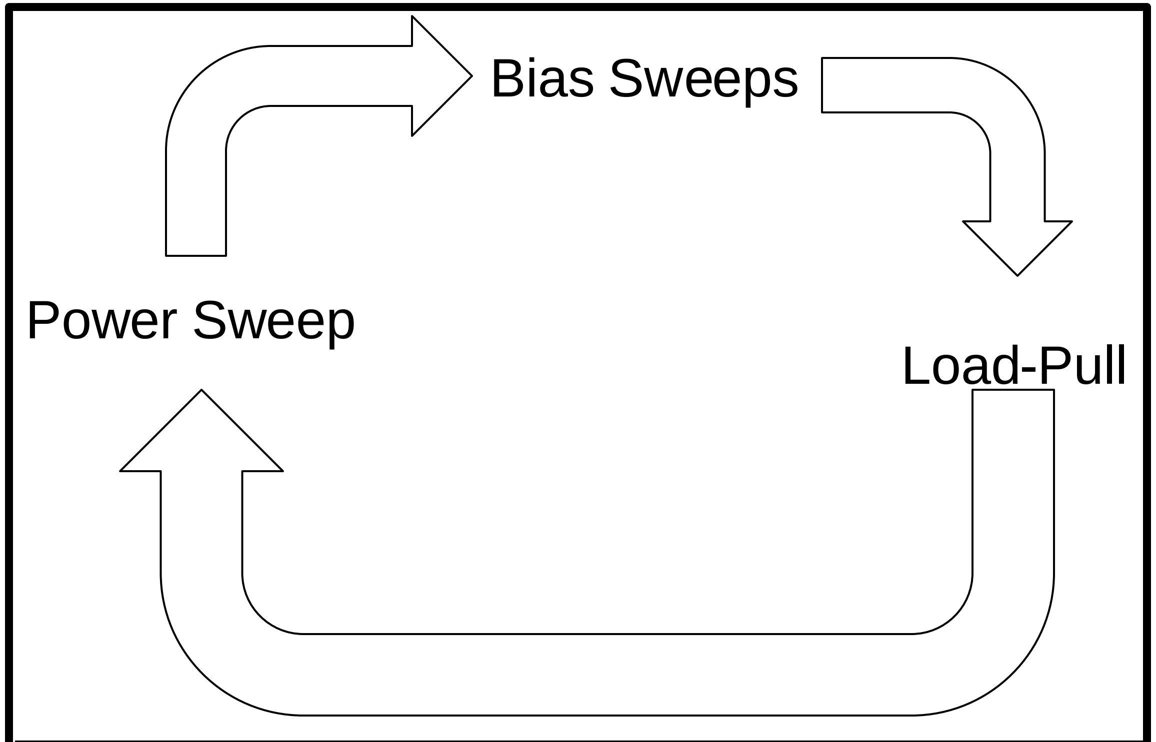
[1]Baylor University
[2]Keysight Technologies
Jonathan_Swindell1@baylor.edu

Outline

- Motivation
- WGAN Smith Tube Image Completion
- Model Architecture
- Error Metrics
- AI-powered Accelerated Computer-Aided Design via ADS Plugin
- Results
- Next Steps and Conclusions

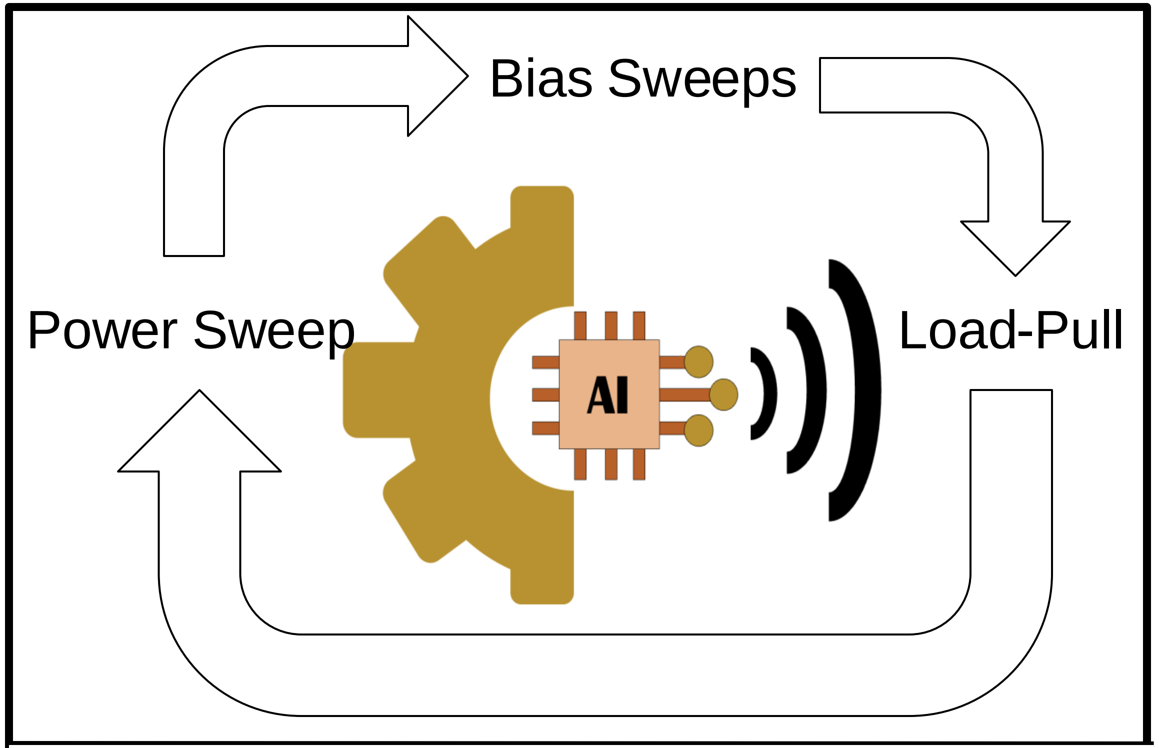
Nonlinear Power Amplifier Design

Traditional RF Design



- Requires many measurements and many design iterations
- Requires years of domain expertise to complete many RF designs
- Requires slow acquisition and expensive specialty equipment for measurements at high frequencies and high powers

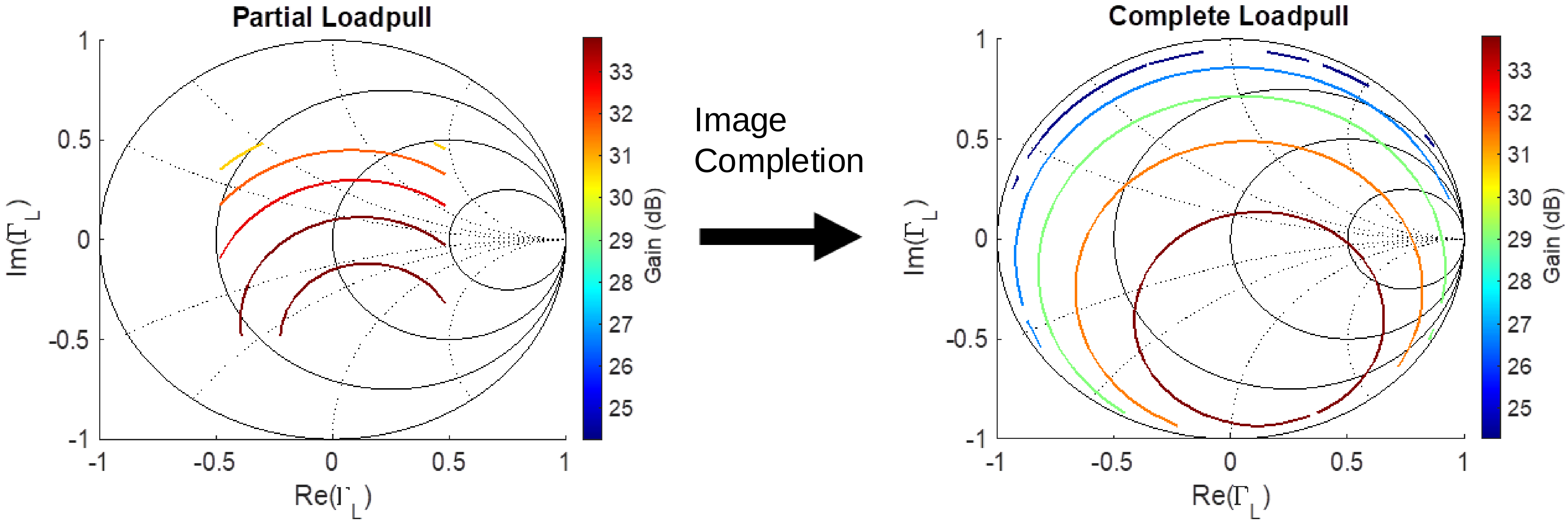
AI-Powered RF Design and Automation



- AI-powered image completion enables a reduction in the number of required simulations or measurements.
- AI-powered design and automation tools enable less experience to be required to complete an RF design.
- AI-powered RF tools can accelerate measurements and simulation actions required by many applications.

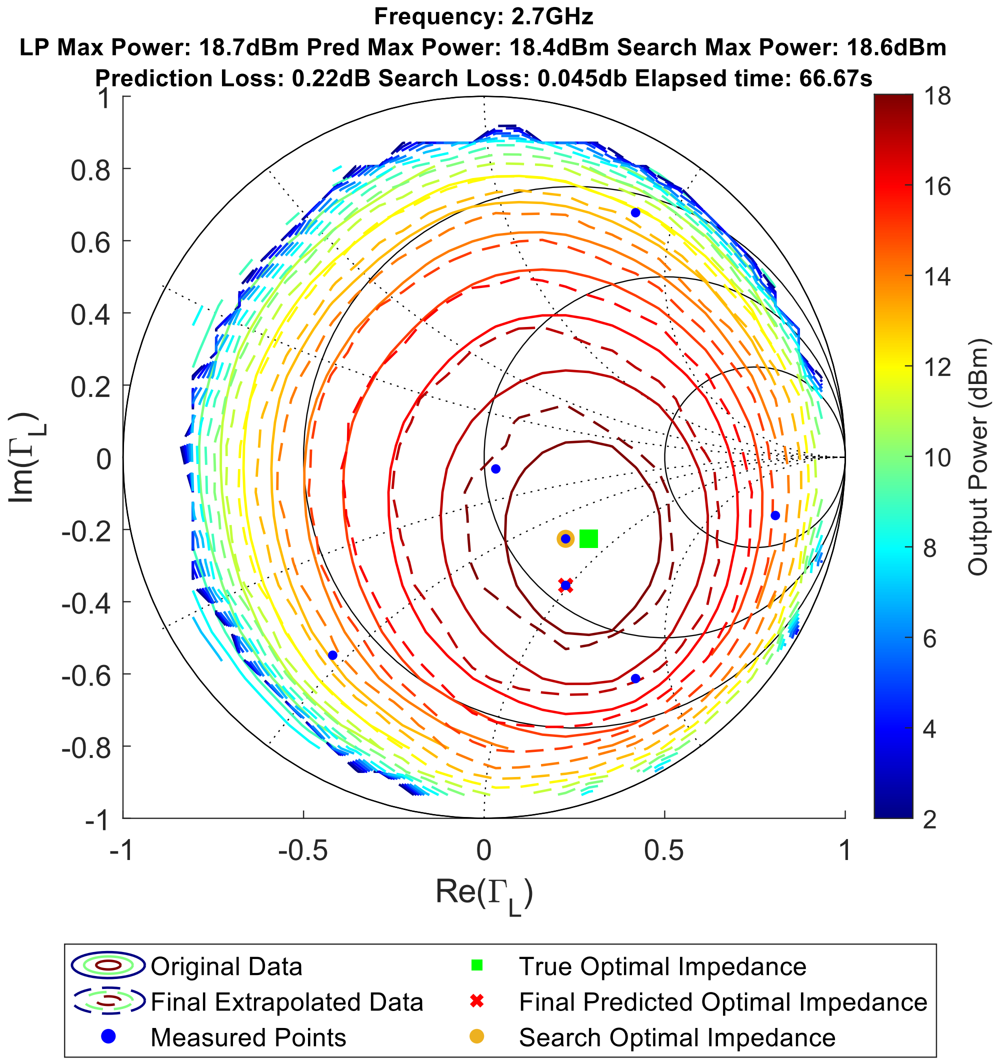
Partial Load-Pull Extrapolation

Deep learning image completion methods can be leveraged to extrapolate an entire set of load-pull performance contours from a small set of measured impedances

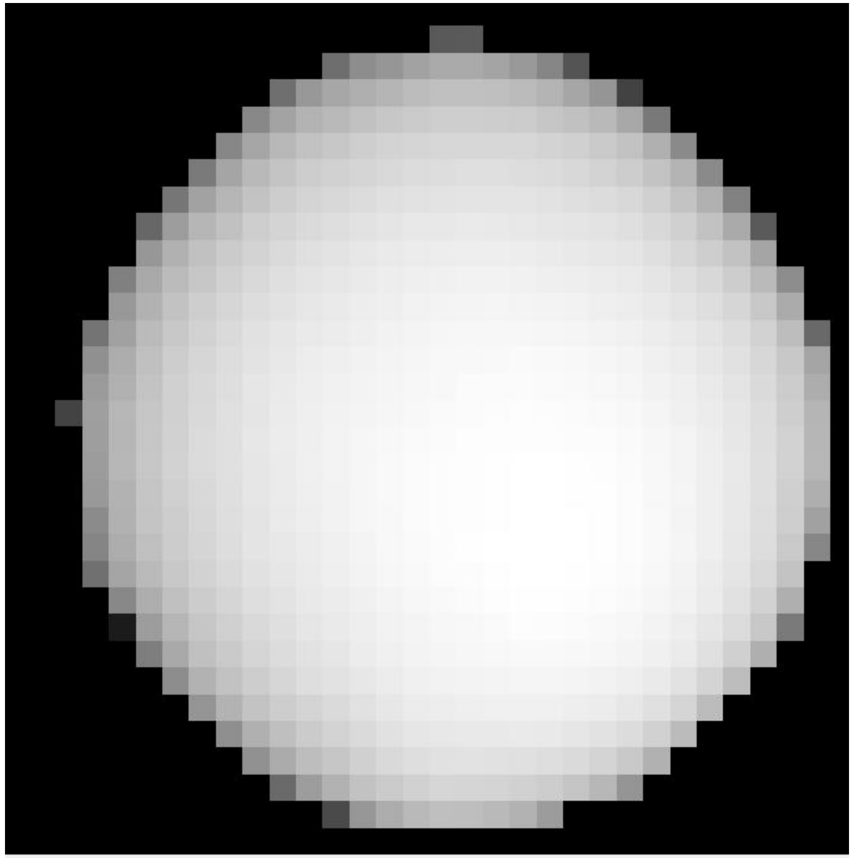


[1] A. Egbert, C. Baylis and R. J. Marks, "Extrapolation of Load-Pull Data: A Novel Use of GAN Artificial Intelligence Image Completion," in IEEE Trans. Microw. Theory Techn., vol. 70, no. 11, pp. 4849-4856, Nov. 2022, doi: 10.1109/TMTT.2022.3209700.

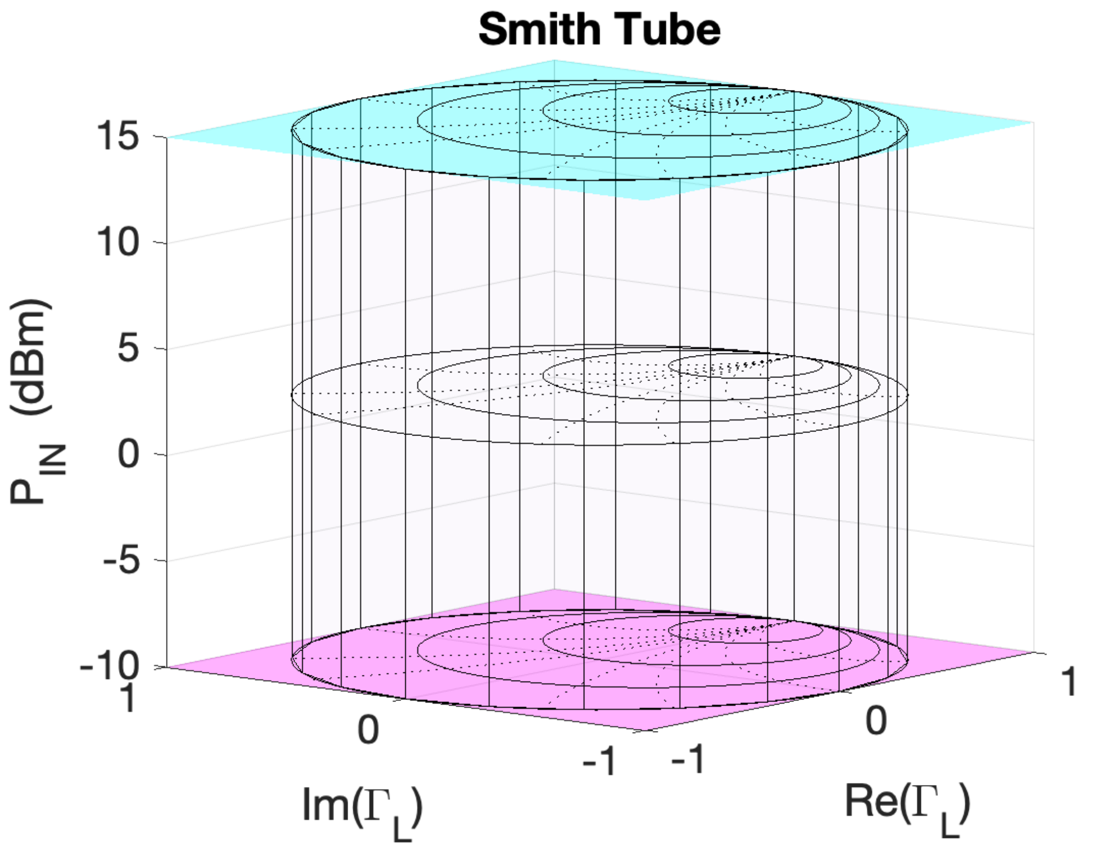
Smith Chart Data as Grayscale Image



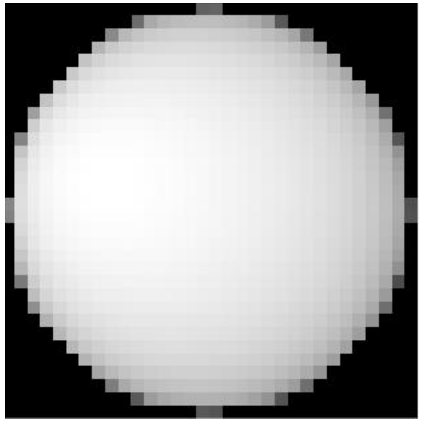
Output power contours as gray scale pixel values



Smith Tube Data as an Image in 3 Dimensions

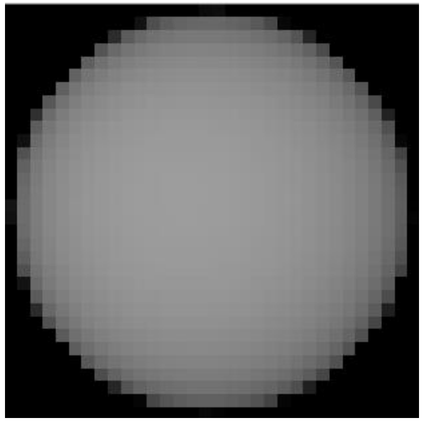


Top Slice



⋮

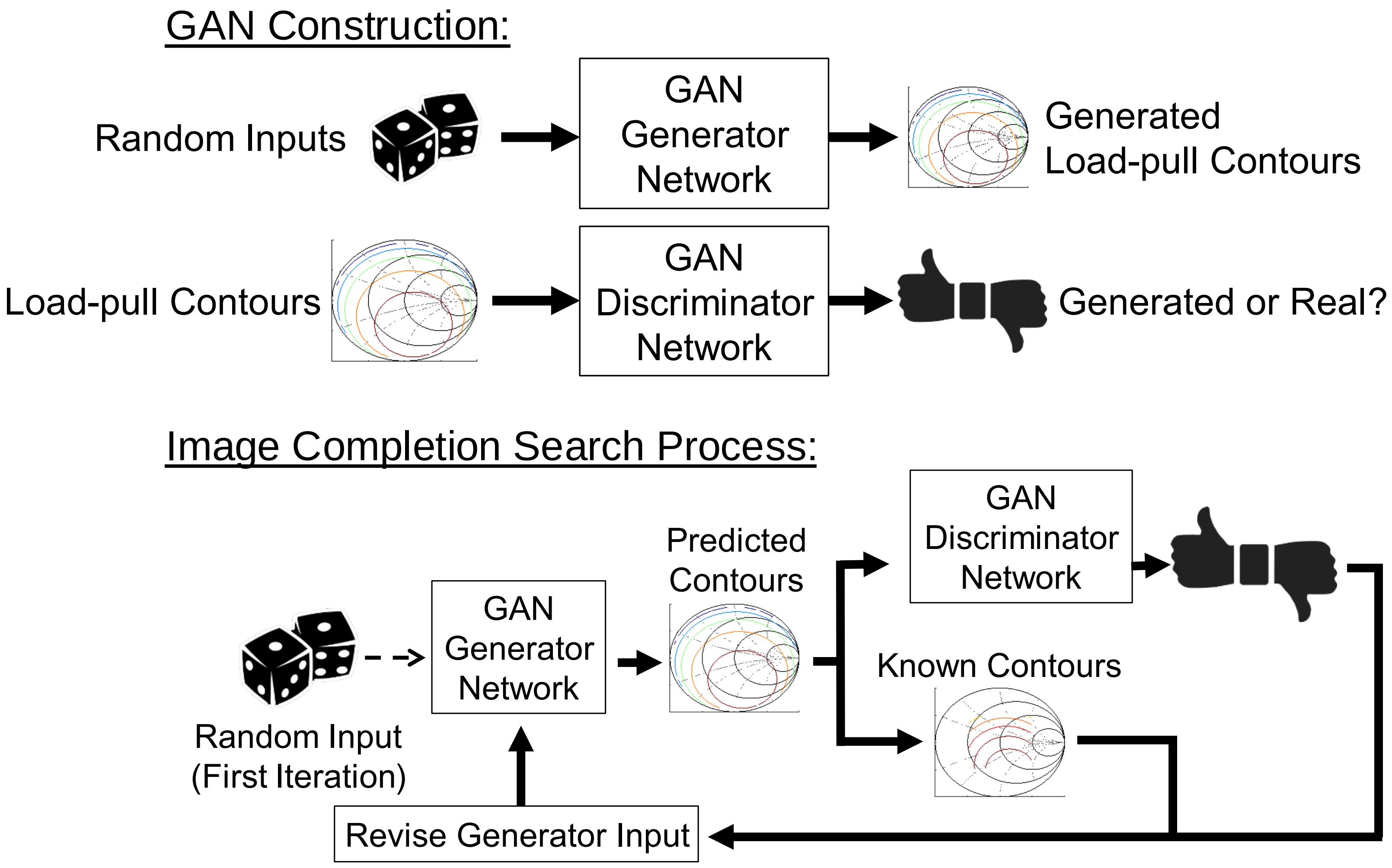
Bottom Slice



[2] J. E. Swindell, A. C. Goad, A. Egbert, C. Baylis, R. J. Marks, "Multidimensional Load-Pull Extrapolation using Generative Adversarial Networks" in *IEEE MTT-S Latin America Microwave Conference*, San Juan, Puerto Rico, Jan. 2025.

Generative Adversarial Networks (GAN) Image Completion

- Two neural networks trained iteratively
- Generator: provides reliable datasets meeting certain requirements
- Discriminator (Critic): trained to recognize datasets meeting requirements.
- Wasserstein GAN was used in initial work



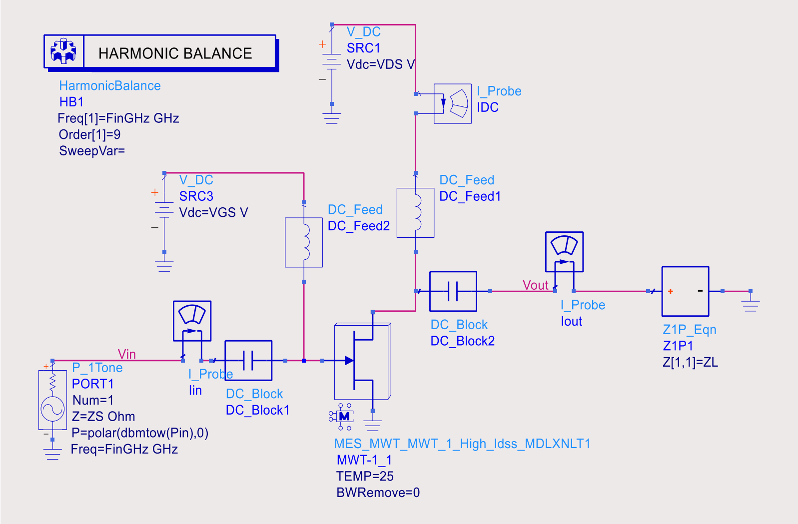
Data Collection

ADS Python API Enables:

- Programmatic design of schematics and layouts
- Run simulations with changing variables
- Easy access and integration with other systems

Modelithics MWT-1 Nonlinear Device Transistor Model:

ADS Simulation:



Variable Sweeps:

Variable	Start	End	Steps
Z_{Load}	476 radially distributed points in the Smith Chart		560
Z_{Source}	Select Points with $\lvert\Gamma\rvert < 1$ $-1 < \Gamma_r < 1$ $-1 < \Gamma_i < 1$		69
Gate Voltage	-2V	0V	11
Drain Voltage	0.1V	6V	11
Power In	-10dBm	15 dBm	28
Frequency	3.00 GHz	3.00 GHz	1

Training data collected using ADS 2024 Update 2 with custom Python scripts used to enable data collection parallelization across two desktop computers.

Completed 130,912,320 simulations in ~21 hours.

Model Training and Compute Resources

Compute Resources:

CPU: AMD Ryzen 5800X3D

RAM: 32 GB DDR4

GPU: NVIDIA RTX 4070Ti Super (16GB VRAM)

Training Time: 3.25 minutes per 30k images per epoch

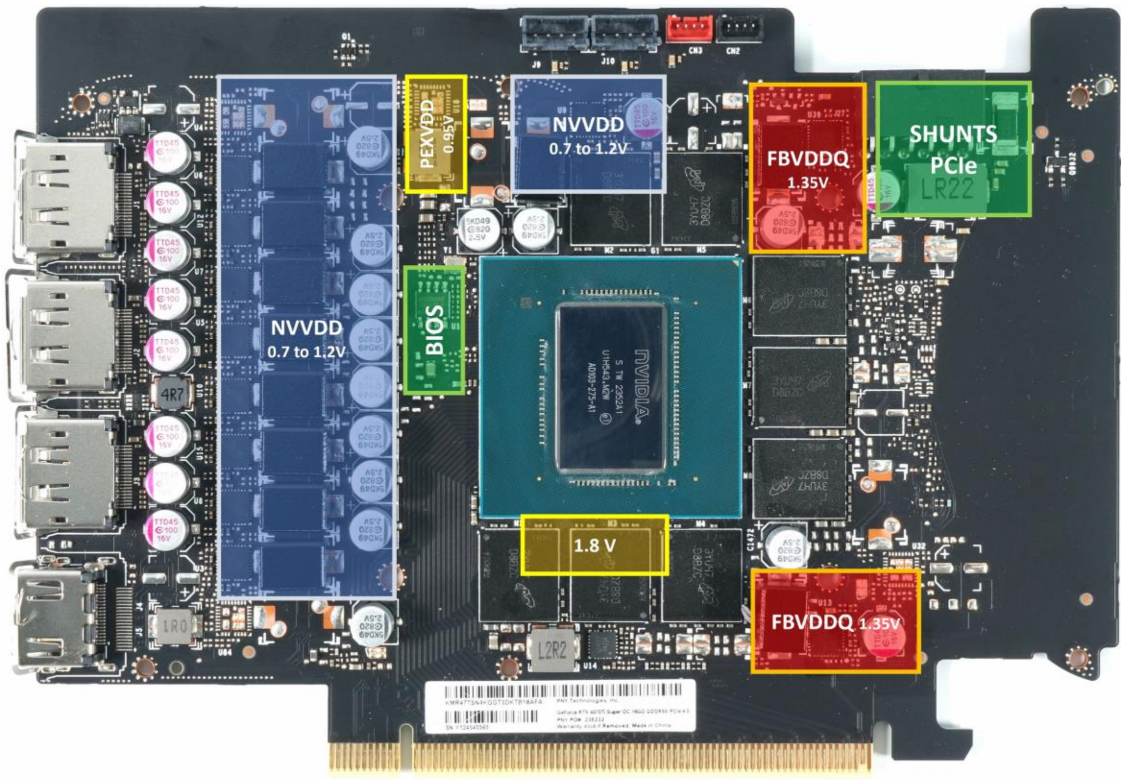


Image credit: Igorslab.de: https://www.igorslab.de/en/pny-geforce-rtx-4070-ti-super-xlr8-gaming-verto-epic-x-16-gb-review-more-clock-speed-through-better-cooling/4/

Types of Error Metrics

Distance Metrics

- Distance between the position of the true optimum value on a contour and the predicted position of that true optimum.

Value Metrics

- The difference between the value predicted at a contour position and the true value at that contour position.

Distance Error Metrics

- **Slice Distance:** distance between predicted and true optimum location on a single power slice in units of pixels

$$D_{slice}=\sqrt{(x_r - xp)^2 + (y_r - y_p)^2}$$

- **Solid Distance:** distance between predicted and measured optimum location on the Smith Tube in units of voxels

$$D_{solid}=\sqrt{(x_r - xp)^2 + (y_r - y_p)^2 + (z_r - z_p)^2}$$

Value Error Metrics

- **Peak Level Error:** the difference between the maximum output power at the measured optimum location in the Smith Tube of the measured and predicted images in units of dB

$$P = \max(R) - \max(P)$$

- **Opportunity Error:** the difference between the measured behavior at the measured and predicted optimum locations in units of dB

$$O_{err} = R(ropt) - R(popt)$$

- **Expectation Error:** the difference between the measured and predicted performance at the predicted optimum in units of dB

$$E_{err} = R(popt) - P(popt)$$

Results

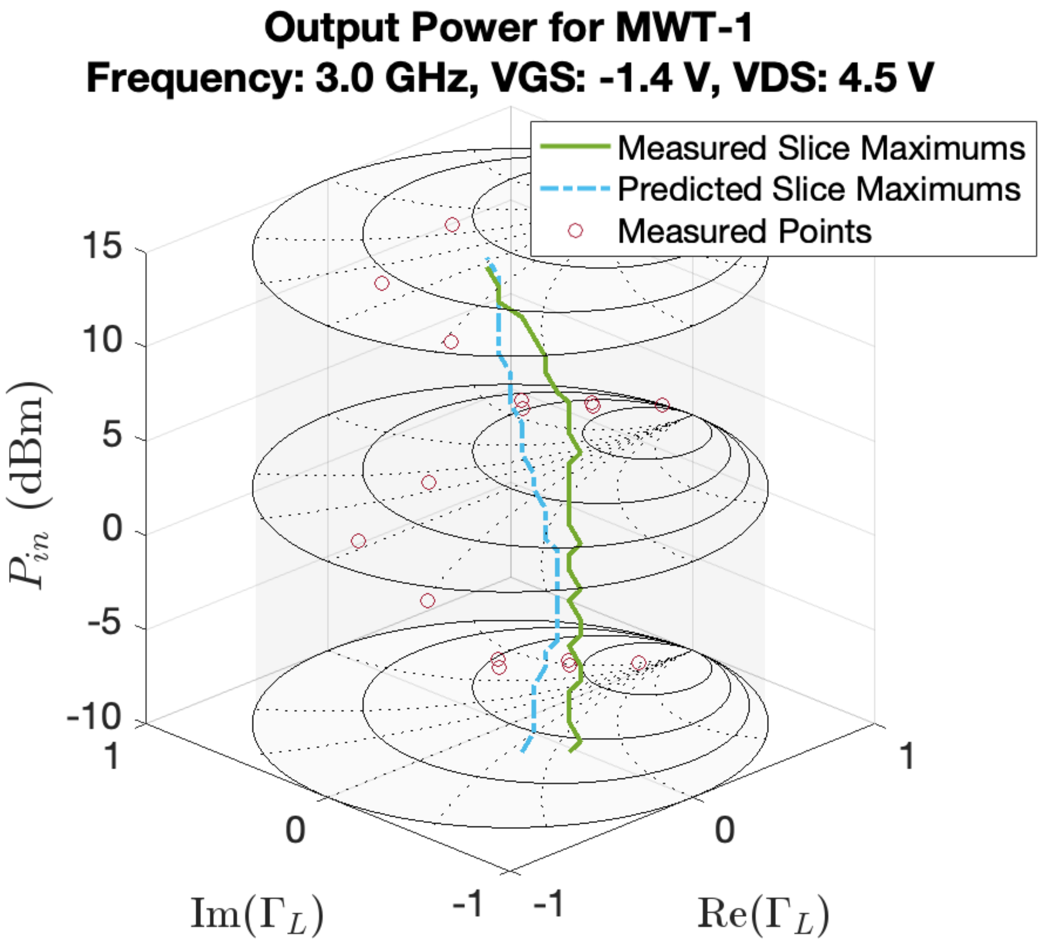


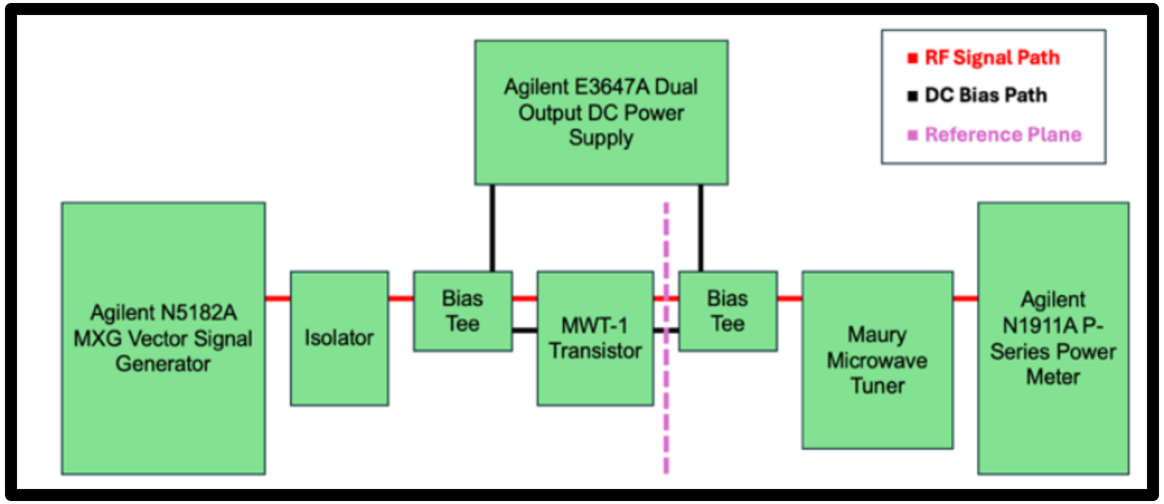
Table I: Measurement Extrapolation Error

Error Metric	Mean	Median
Slice Distance (Reflection Coefficient)	0.147	0.129
Solid Distance (voxels)	2.028	1.414
Peak Level Error (dB)	-0.1524	-0.1435
Opportunity Error (dB)	0.1883	0.06501
Expectation Error (dB)	-0.3407	-0.277

This accuracy was achieved with a **1,500 times reduction** in the required number of simulated load-pull points!

[2] J. E. Swindell, A. C. Goad, A. Egbert, C. Baylis, R. J. Marks, “Multidimensional Load-Pull Extrapolation using Generative Adversarial Networks” in *IEEE MTT-S Latin America Microwave Conference*, San Juan, Puerto Rico, Jan. 2025.

Multidimensional Load Pull Extrapolation on Measured Data



- Extrapolating from 16 points measured on the MWT 1 Transistor can complete a Smith Tube with comparable errors to the simulation case.
- This method can be used to enable a 1,500 - times reduction in requirement load-pull measurement!

[3] [Submitted] J. E. Swindell, A. C. Goad, A. Egbert, Et. al., "Measurement Validation of AI-Powered Multidimensional Nonlinear Load-Pull Extrapolation ", IEEE International Microwave Symposium, San, Francisco CA, 2025.

Comparison between Simulation and Measured Results

Table 1. Simulation Extrapolation Error

Error Metric	Mean	Median
Slice Distance (Reflection Coefficient)	1.907	1.414
Solid Distance (voxels)	2.402	2.236
Peak Level Error (dB)	-0.7133	-0.5935
Opportunity Error (dB)	0.2595	0.1657
Expectation Error (dB)	-0.9728	-0.7984

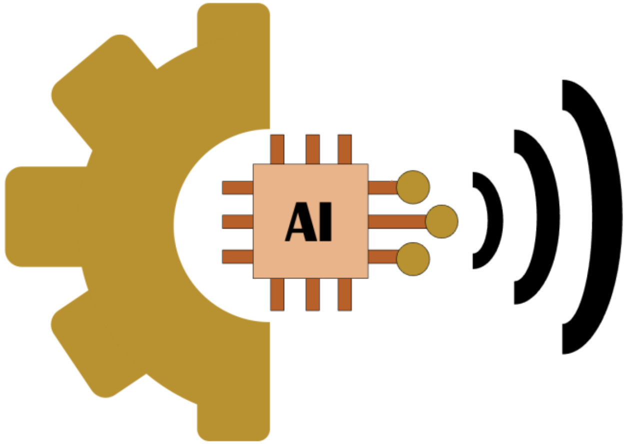
Table 2. Measurement Extrapolation Error

Error Metric	Mean	Median
Slice Distance (Reflection Coefficient)	2.271	2.000
Solid Distance (voxels)	2.028	1.414
Peak Level Error (dB)	-0.1524	-0.1435
Opportunity Error (dB)	0.1883	0.06501
Expectation Error (dB)	-0.3407	-0.277

AIML can Accelerate Computer-Aided Design

These results are awesome, but remember our frustrated engineer:

- to make use of these AIML advancements, we need to deploy these tools in RF CAD software.



AIML-powered RF circuit design and optimization requires CAD tools with APIs that enable design automation. Baylor has used Keysight ADS's Python Integration API to develop an ADS plugin that can reduce the required number of load-pull simulations by 1500 times.

AI-powered accelerated Computer-Aided Design via RF Design Automation

SparseIdx	Pout	(ZL-50)/(ZL+50)	Pin
1.000	-8.180	0.548 / 139.764	-10.000
2.000	-3.583	0.355 / 95.194	-8.387
3.000	-0.086	0.500 / 45.000	-6.774
4.000	0.606	0.355 / -5.194	-5.161
5.000	-0.339	0.548 / -49.764	-3.548
6.000	1.001	0.419 / -94.399	-1.935
7.000	0.827	0.591 / -135.000	-0.323
8.000	4.737	0.419 / -175.601	1.290
9.000	8.235	0.548 / 130.236	3.710
10.000	12.589	0.419 / 85.601	5.323
11.000	15.302	0.591 / 45.000	6.935
12.000	16.887	0.419 / 4.399	8.548
13.000	16.666	0.548 / -40.236	10.161
14.000	19.022	0.355 / -84.806	11.774
15.000	19.471	0.500 / -135.000	13.387
16.000	21.981	0.355 / 174.806	15.000

**Video removed processing time. Ported MATLAB processing to Python which is currently unoptimized.*

Partial_Loadpull_3d [page 1]:1

File Edit View Insert Marker History Options Tools Page Window Python Help

Partial_Loadpull_3D

Palette

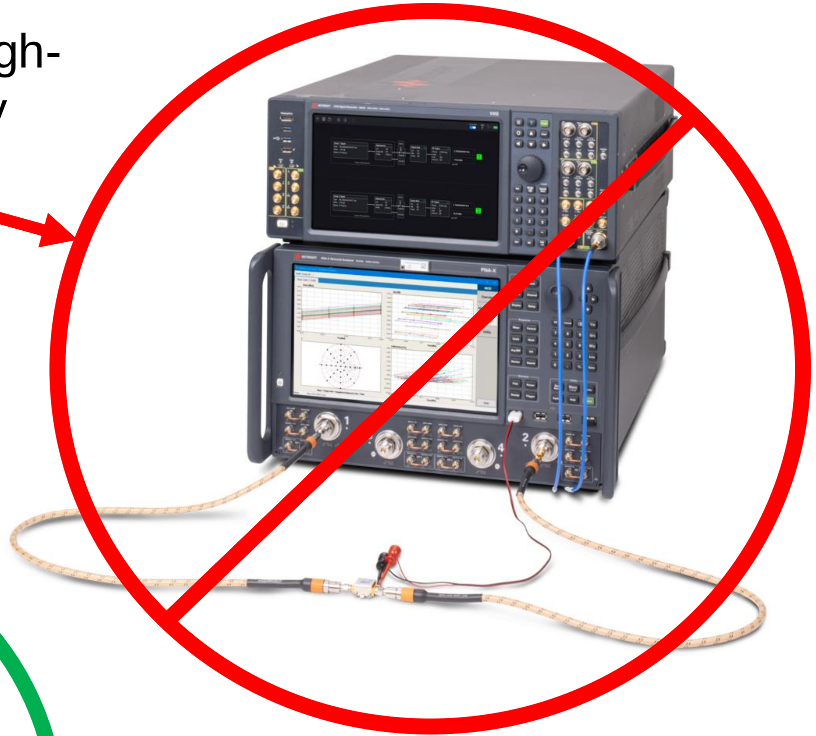
SparseIdx	Pout	(ZL-50)/(ZL+50)	Pin
1.000	-8.180	0.548 / 139.764	-10.000
2.000	-3.583	0.355 / 95.194	-8.387
3.000	-0.086	0.500 / 45.000	-6.774
4.000	0.606	0.355 / -5.194	-5.161
5.000	-0.339	0.548 / -49.764	-3.548
6.000	1.001	0.419 / -94.399	-1.935
7.000	0.827	0.591 / -135.000	-0.323
8.000	4.737	0.419 / -175.601	1.290
9.000	8.235	0.548 / 130.236	3.710
10.000	12.589	0.419 / 85.601	5.323
11.000	15.302	0.591 / 45.000	6.935
12.000	16.887	0.419 / 4.399	8.548
13.000	16.666	0.548 / -40.236	10.161
14.000	19.022	0.355 / -84.806	11.774
15.000	19.471	0.500 / -135.000	13.387
16.000	21.981	0.355 / 174.806	15.000

Applications

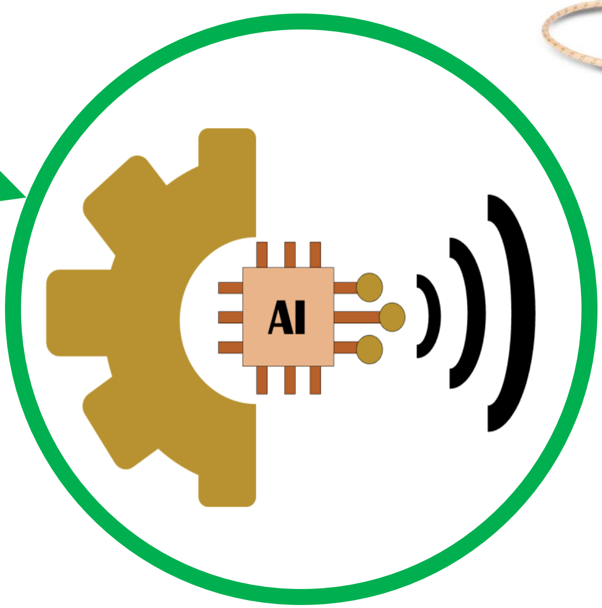
High-Power, High-Frequency Load Pull Approximation

While active load-pull measurement devices can take load-pull measurements quickly, these devices are expensive and may not be available for **high-power** or **high-frequency** devices.

Not available for very high-power or high-frequency applications



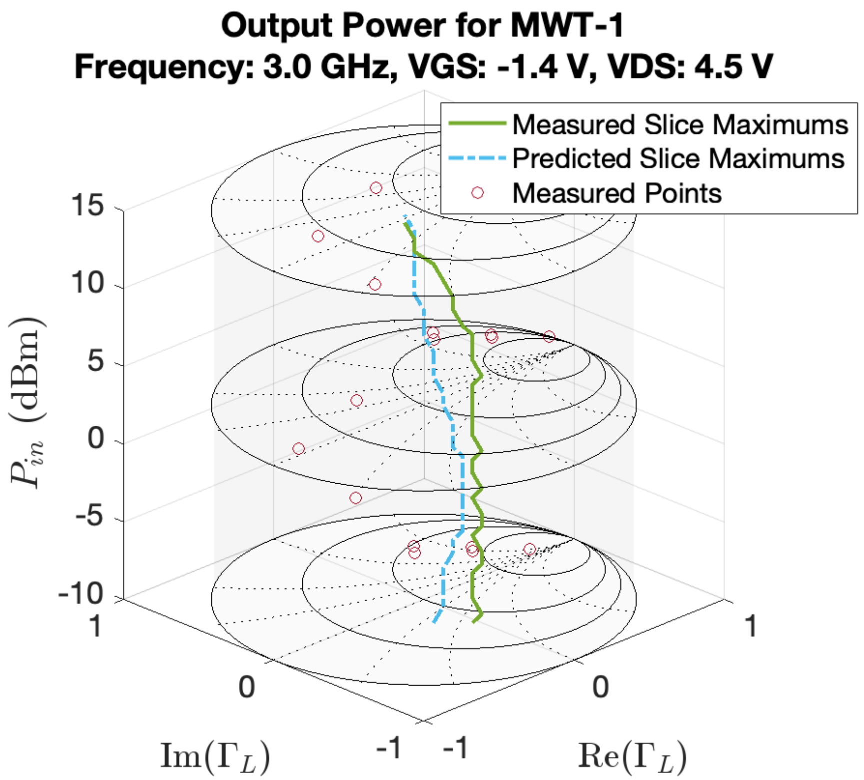
In these cases, multidimensional load-pull extrapolation can approximate performance at those locations.



Accelerated Simulations can speed up large scale simulations

- Multidimensional Load-Pull extrapolation can be used to accelerate large-scale simulations.

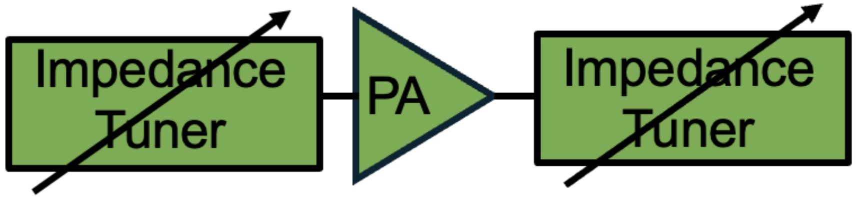
- Measurements or simulations can be taken at strategic points in the search space and then passed to the ML network for extrapolation.

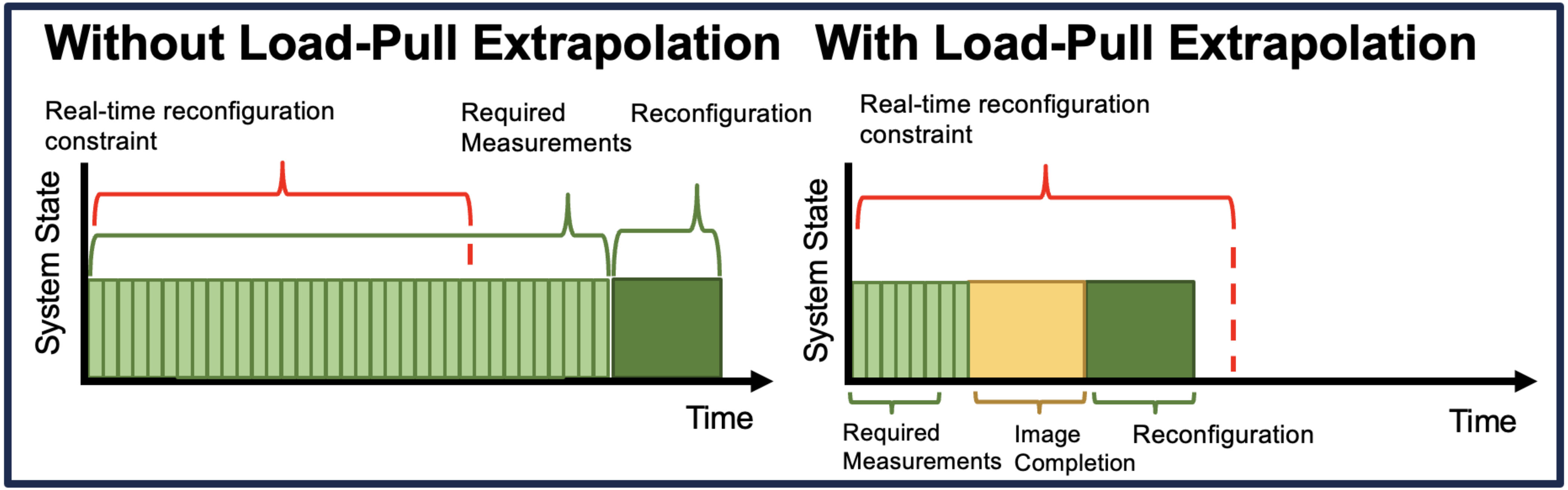


This accuracy was achieved with a **1,500 times reduction** in the required number of simulated load-pull points!

Accelerated Measurements Enable Real-Time Reconfiguration

Reconfigurable circuits break the coherence between the input and output of a transmit chain.



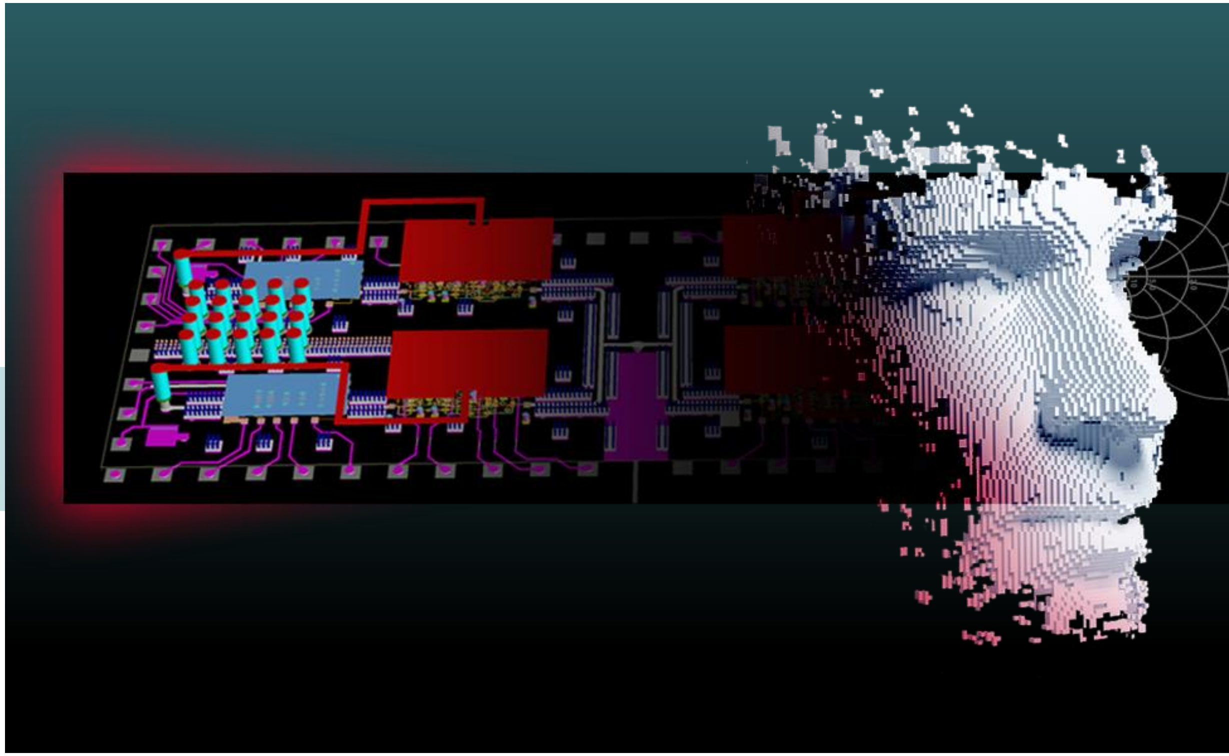


Thank you

Re-imagining RF Design Using ADS Python APIs & AI/ML

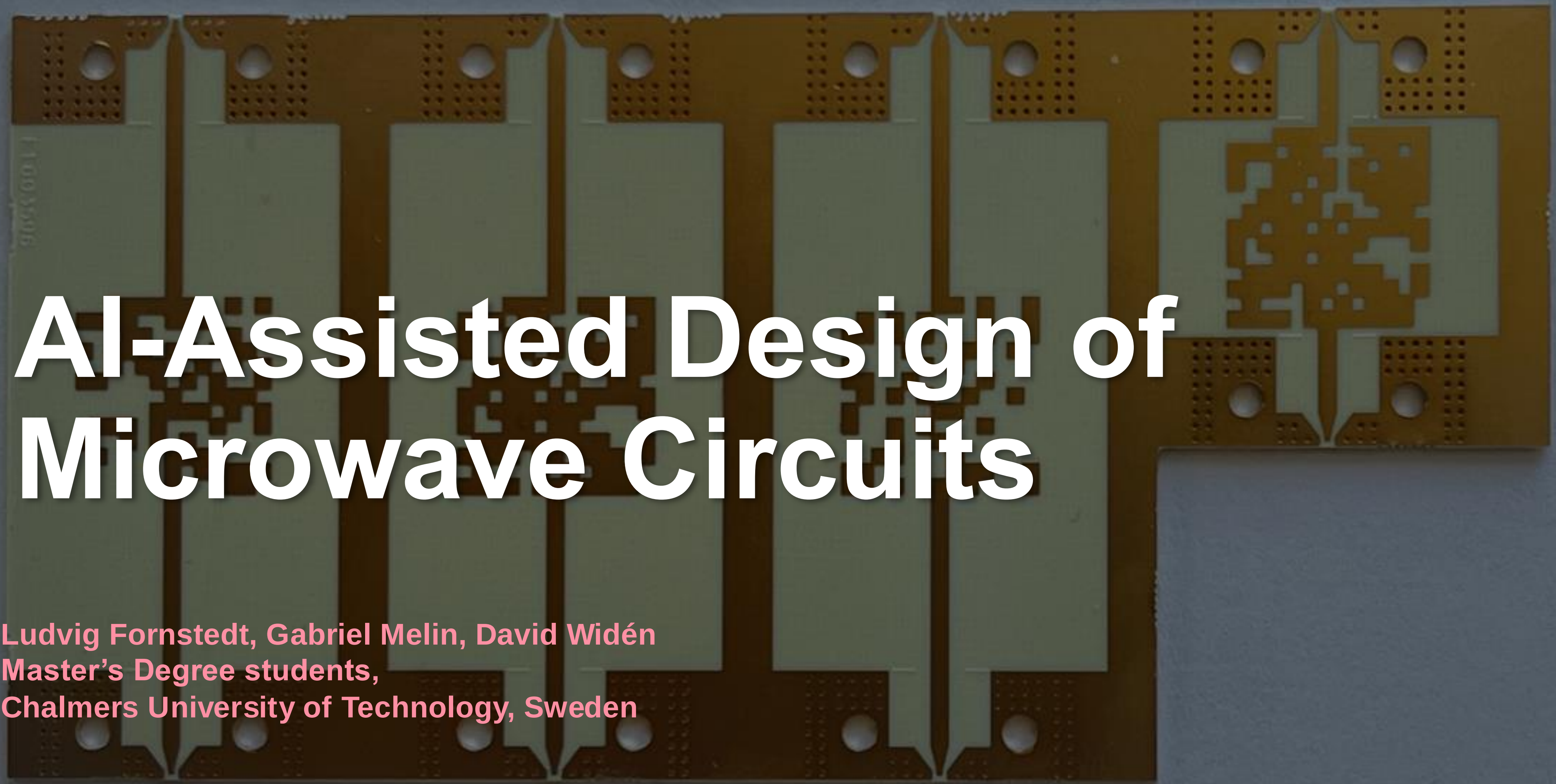
Agenda

1. Python APIs in ADS
2. Case 1: Generative AI for Loadpull
   (Video from Baylor Univ)
3. Case 2: “QR Code” Filters with RFPro
   (Video from Chalmers Univ)
4. AI/ML Technologies in ADS



KEYSIGHT

PLAY VIDEO



AI-Assisted Design of Microwave Circuits

Ludvig Fornstedt, Gabriel Melin, David Widén
Master's Degree students,
Chalmers University of Technology, Sweden

Who are we?

Master's Students at Chalmers University of Technology, Sweden:

- **Ludvig Fornstedt** – Complex Adaptive Systems Program
- **Gabriel Melin** – Complex Adaptive Systems Program
- **David Widén** – Wireless, Photonics, and Space Engineering Program

- Worked on AI-Assisted Design of Microwave Circuits as part of our Bachelor's Thesis.


CHALMERS

Maskininlärning för design av mikrovåg-skretsar

En maskininlärningsassisterad designmetod för pixelerade anpassningsnät i 1-10 GHz-området

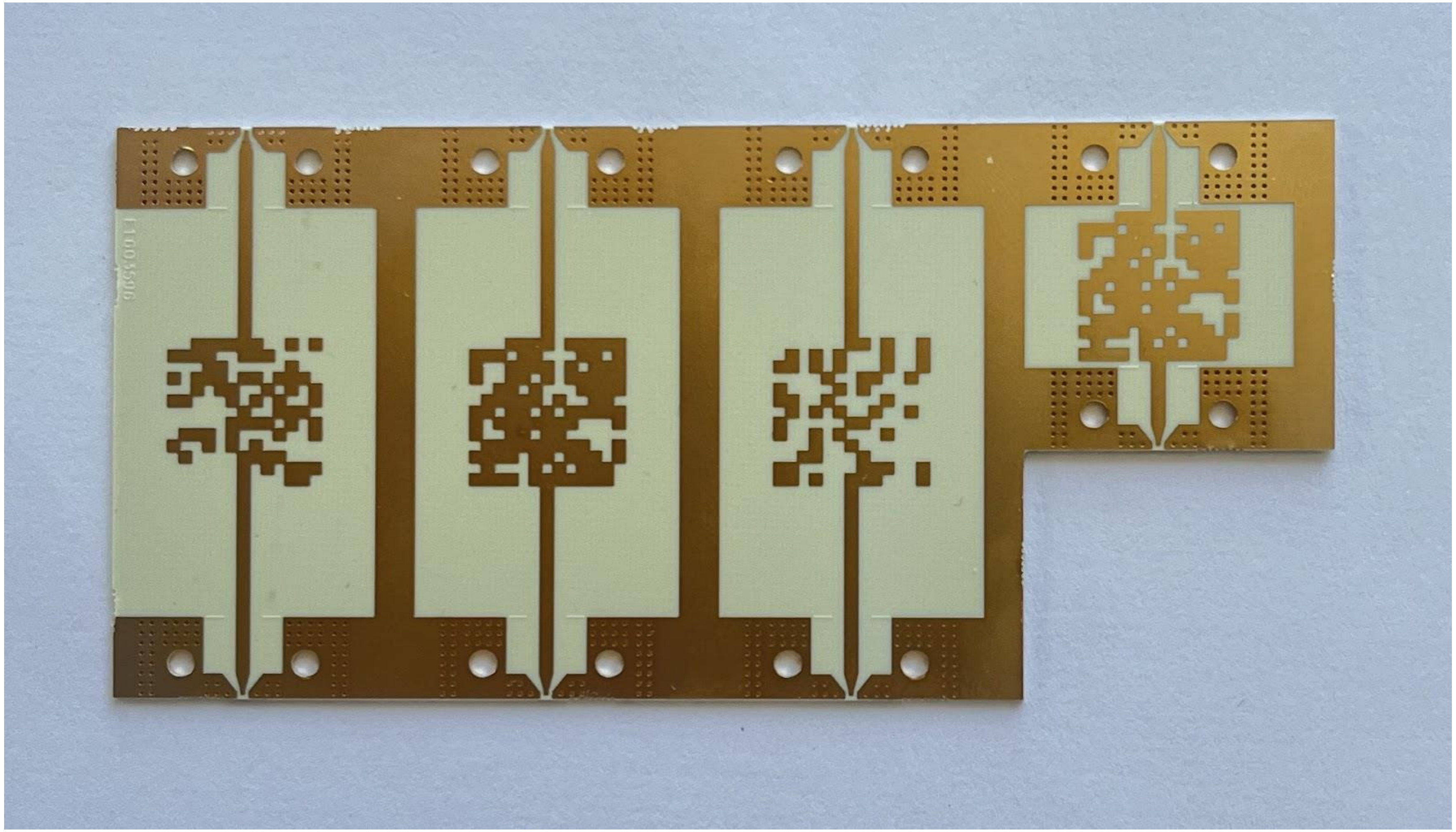
Kandidatarbete i Teknisk Fysik

Alexander Bohlin, Ludvig Fornstedt, Pontus Lindeberg Fredriksson, Gabriel Melin & David Widén

Institutionen för mikroteknologi och nanovetenskap
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2025
www.chalmers.se

AI-Assisted Design for "QR-code" filters

Quick and opens up new, previously unthinkable design solutions.

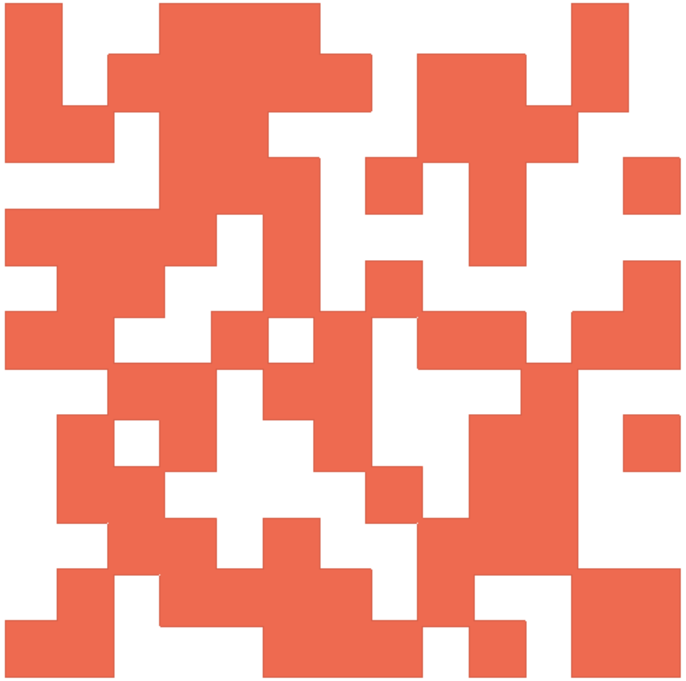


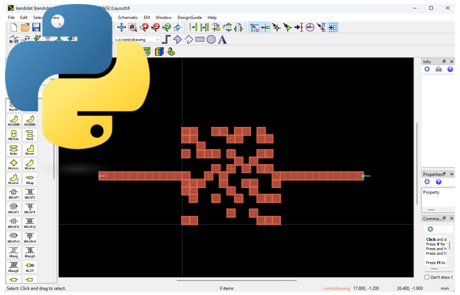
Overview

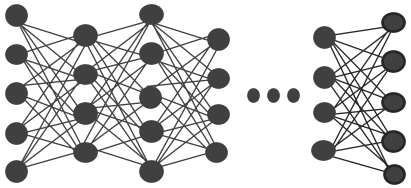
Simulate ~100 000 Circuits in ADS

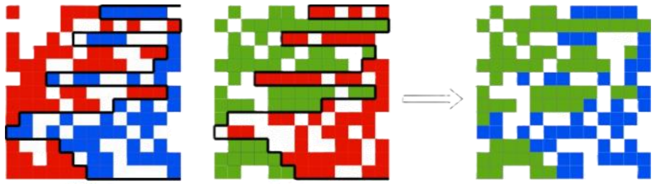
Train a CNN to predict s-parameters

Use the CNN in a Genetic Algorithm to produce circuit







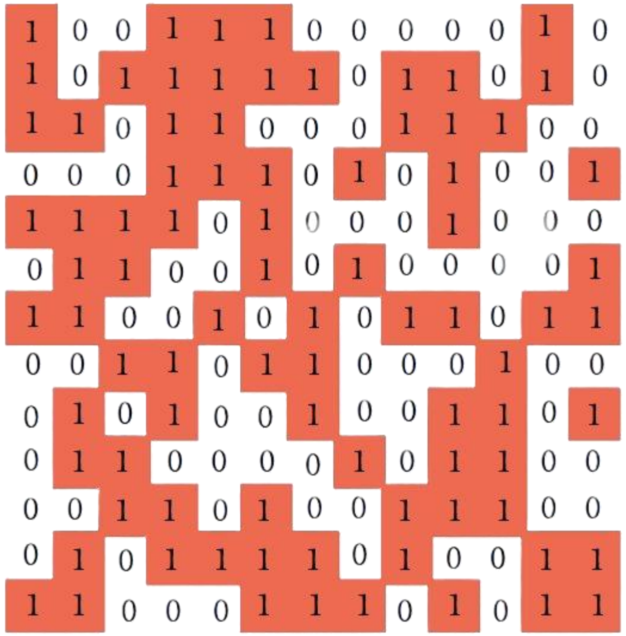


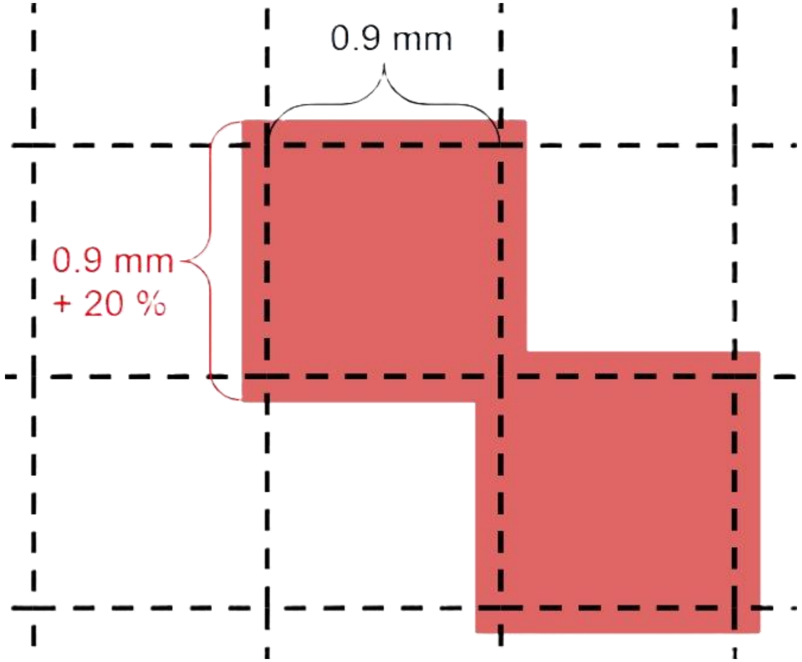
Simulation

- Represent circuit as a binary matrix
  1=metal
  0= no metal

- Each square is 0.9 mm in length
  L ≈ 120° @ 5 GHz

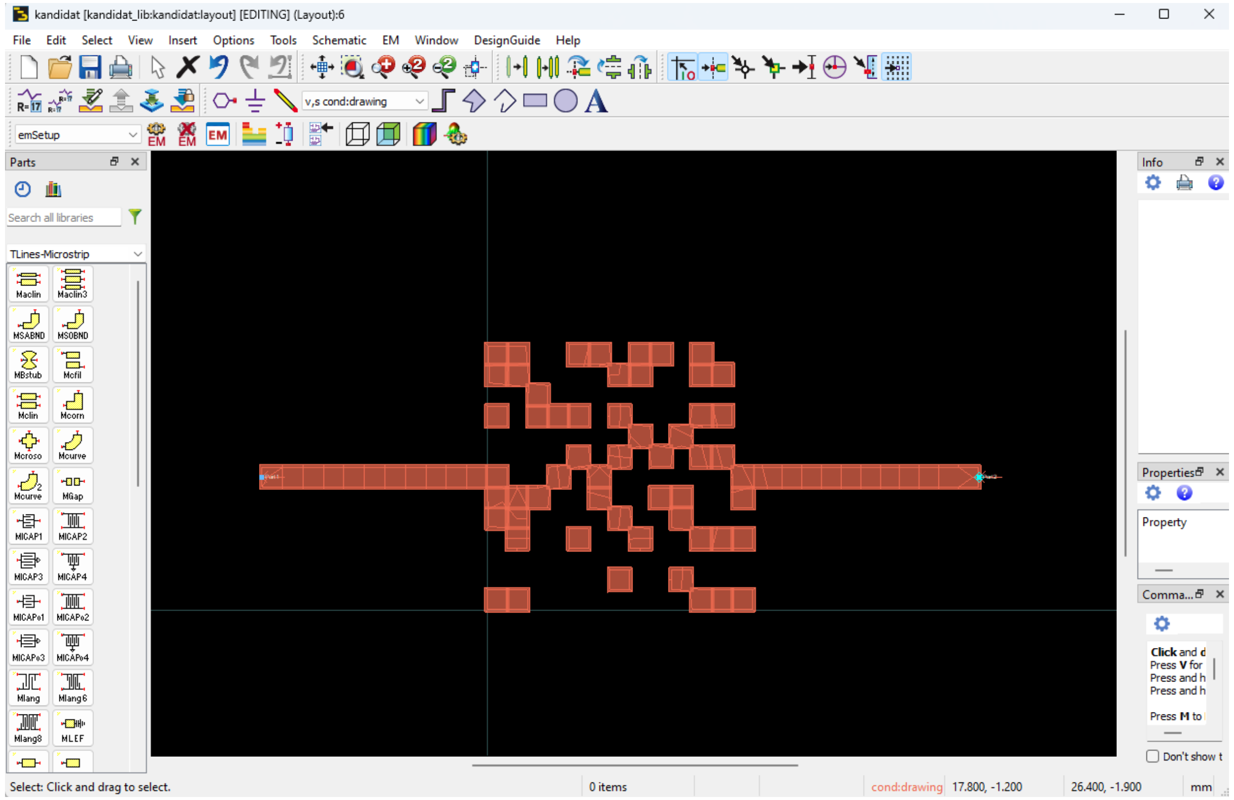
- Randomly generate and select 13 x 13 matricies which satisfy certain conditions





Simulation

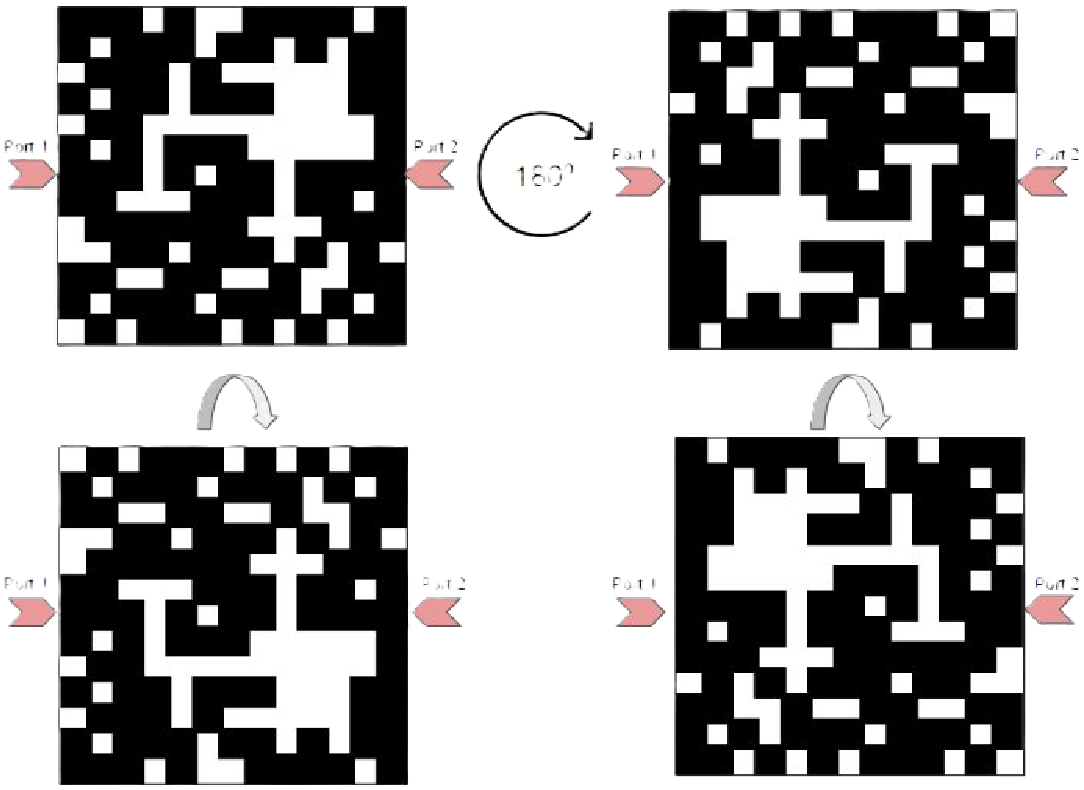
- Convert binary matrices into layouts in ADS using python



- Simulate S-parameters for 19 points between 1-10 GHz

Simulation

- Start simulation and extract data using python
- Process all data using python

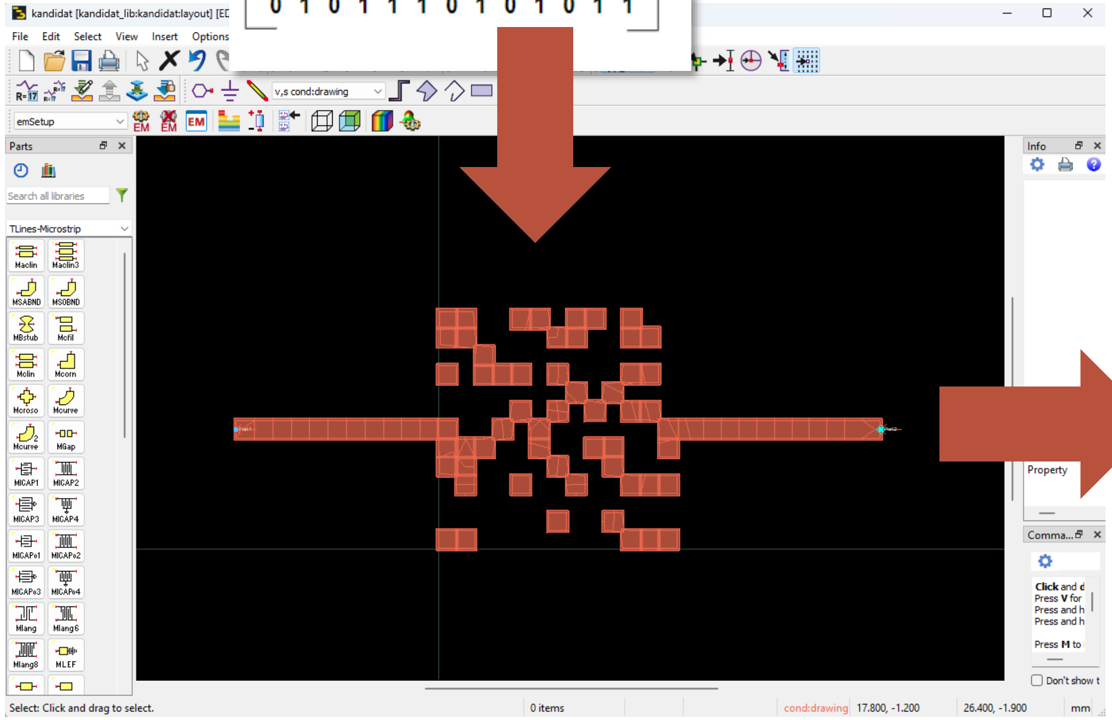


$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\Rightarrow
$$

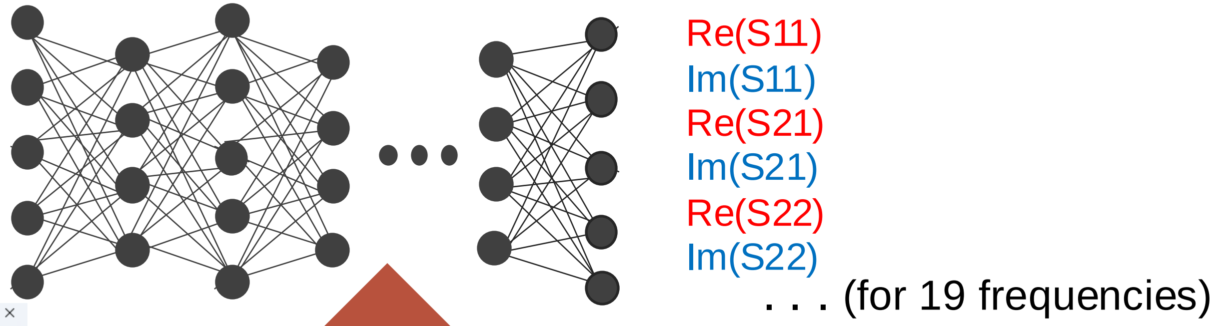
Re(S11)
Im(S11)
Re(S21)
Im(S21)
Re(S22) . . . (for 19 frequencies)
Im(S22)

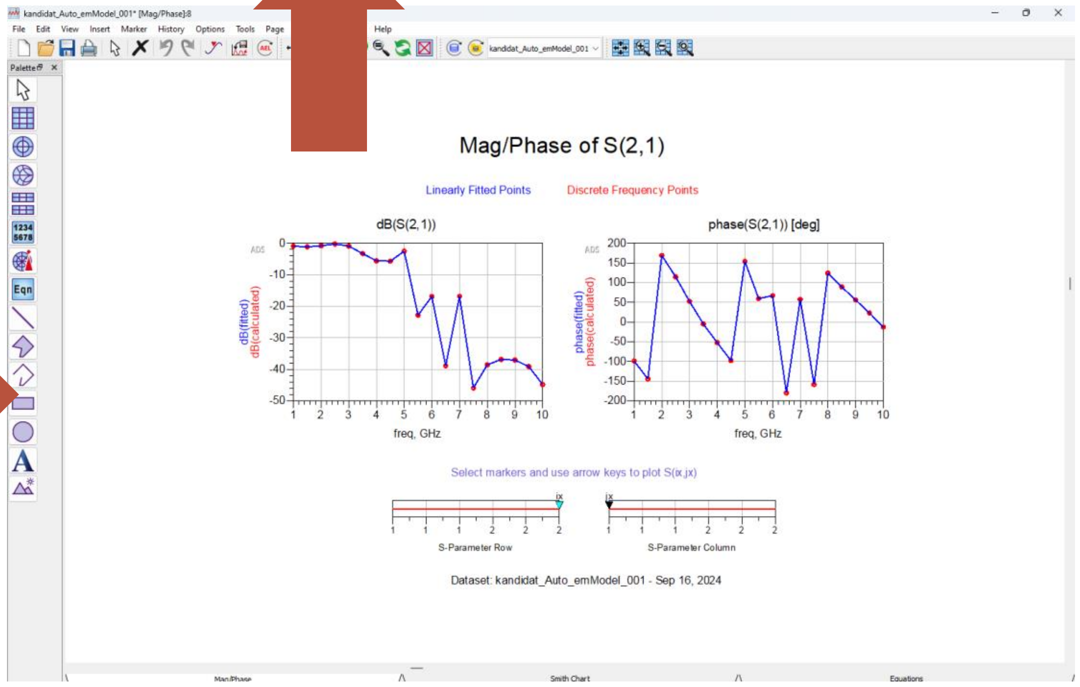
Neural-Network

$$\begin{bmatrix} 1&1&1&0&1&0&0&1&1&1&1&0&1 \\ 1&0&1&1&1&0&1&1&0&1&0&1&1 \\ 0&1&1&1&0&1&0&0&0&0&0&1&1 \\ 1&0&1&1&0&1&1&1&0&0&0&1&1 \\ 0&1&1&0&0&0&0&0&0&0&0&0&1 \\ 1&0&1&0&1&1&1&0&0&0&0&0&1 \\ 1&1&1&0&1&0&1&1&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1&0&1&1&0&1 \\ 0&1&1&1&1&1&1&0&0&0&1&1&1 \\ 0&0&1&1&0&1&1&1&0&1&0&1&0 \\ 1&1&0&0&1&1&0&0&1&0&0&1&1 \\ 1&0&1&1&1&0&1&1&1&0&1&0&1 \\ 0&1&0&1&1&1&0&1&0&1&0&1&1 \end{bmatrix}$$

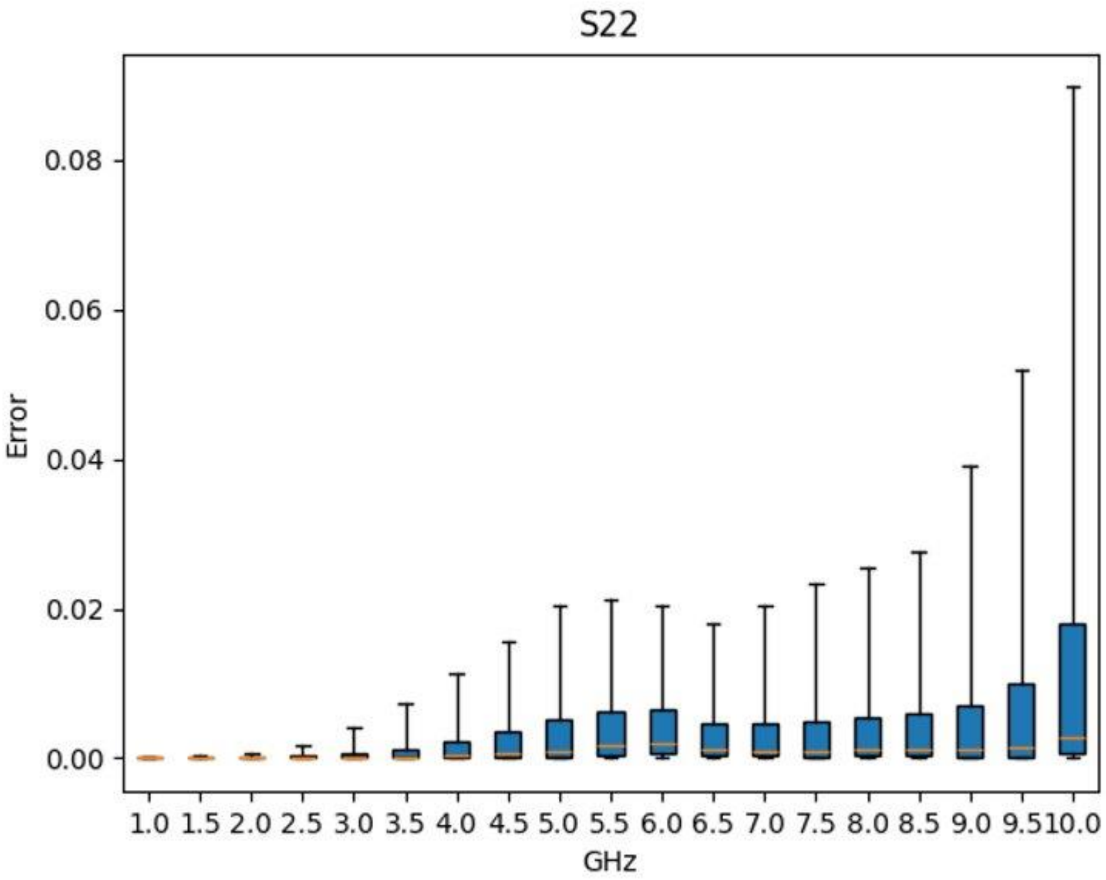


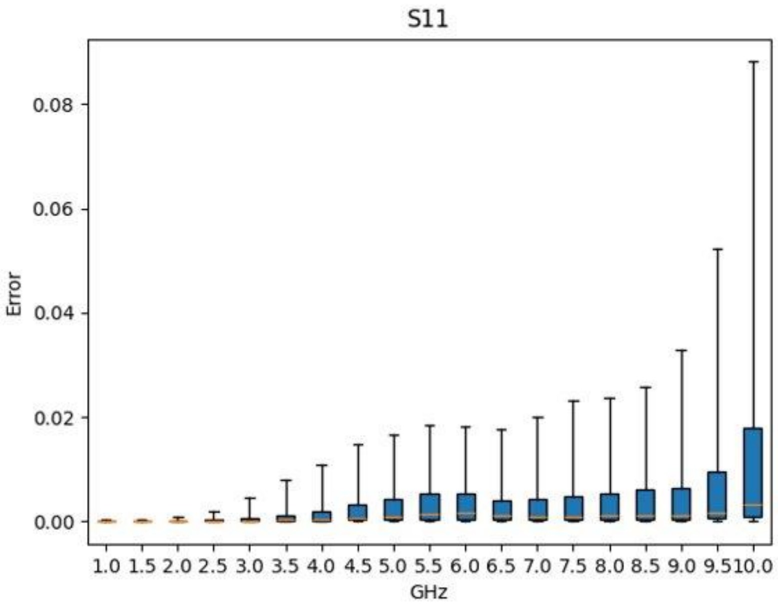
- 14 convolutional layers with residual connections
- 5 fully connected layers

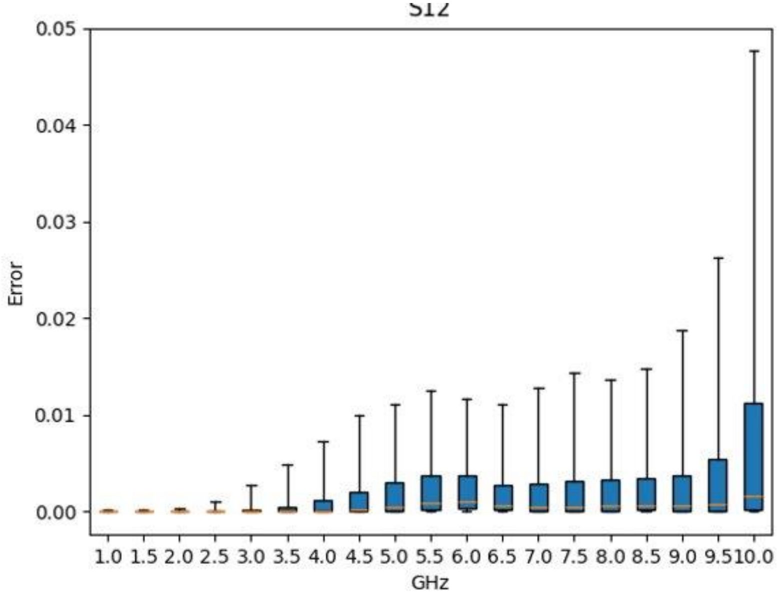




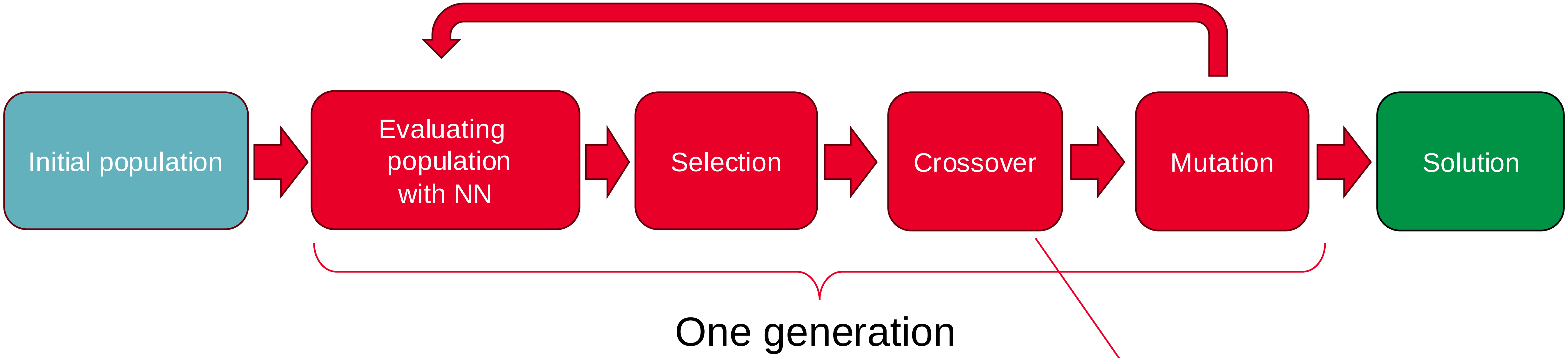
Neural-Network







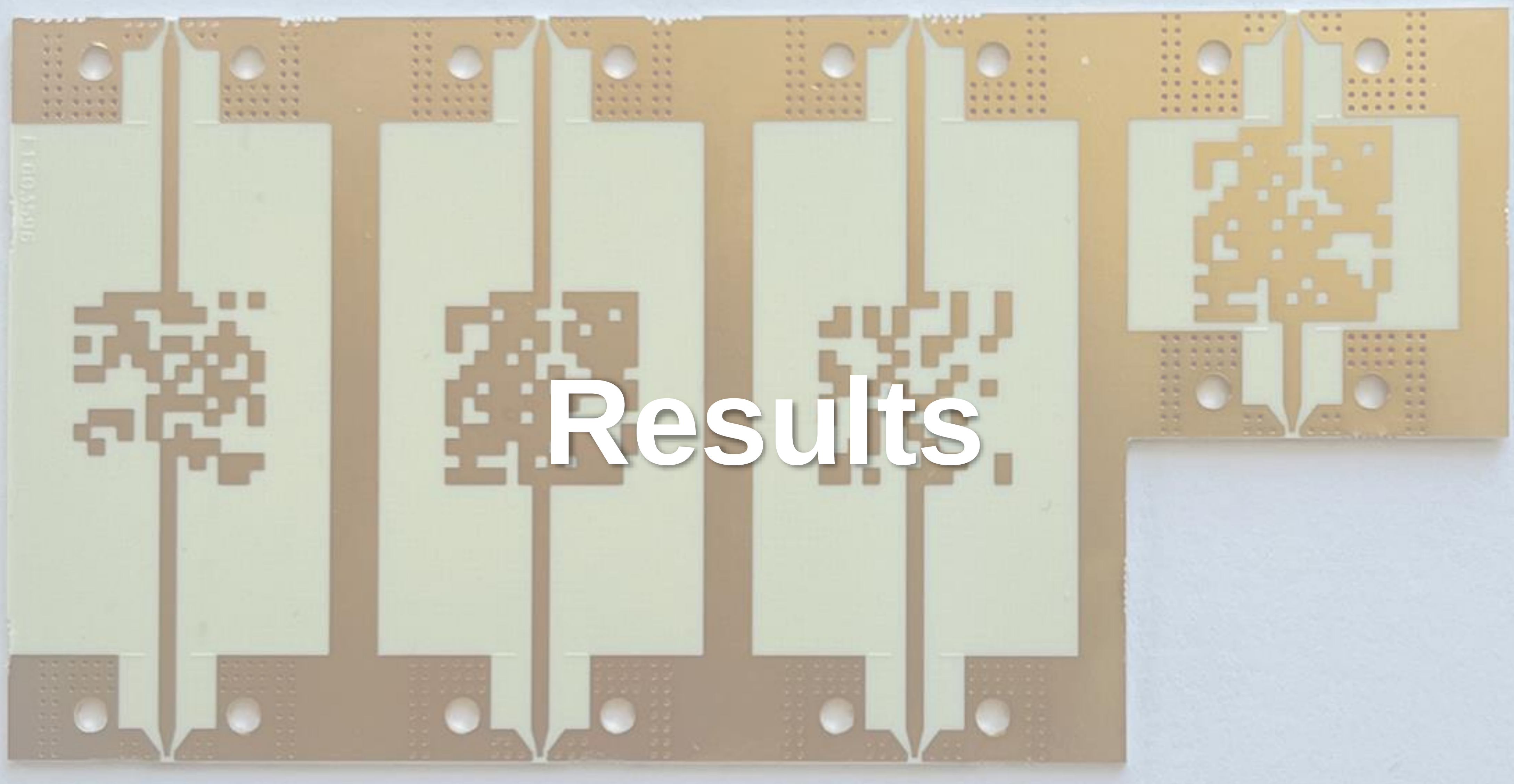
Genetic Optimization



Evaluation:

- Using the NN to predict the S-parameters
- Compare them to the chosen goal parameters
- Calculating a fitness score for each individual

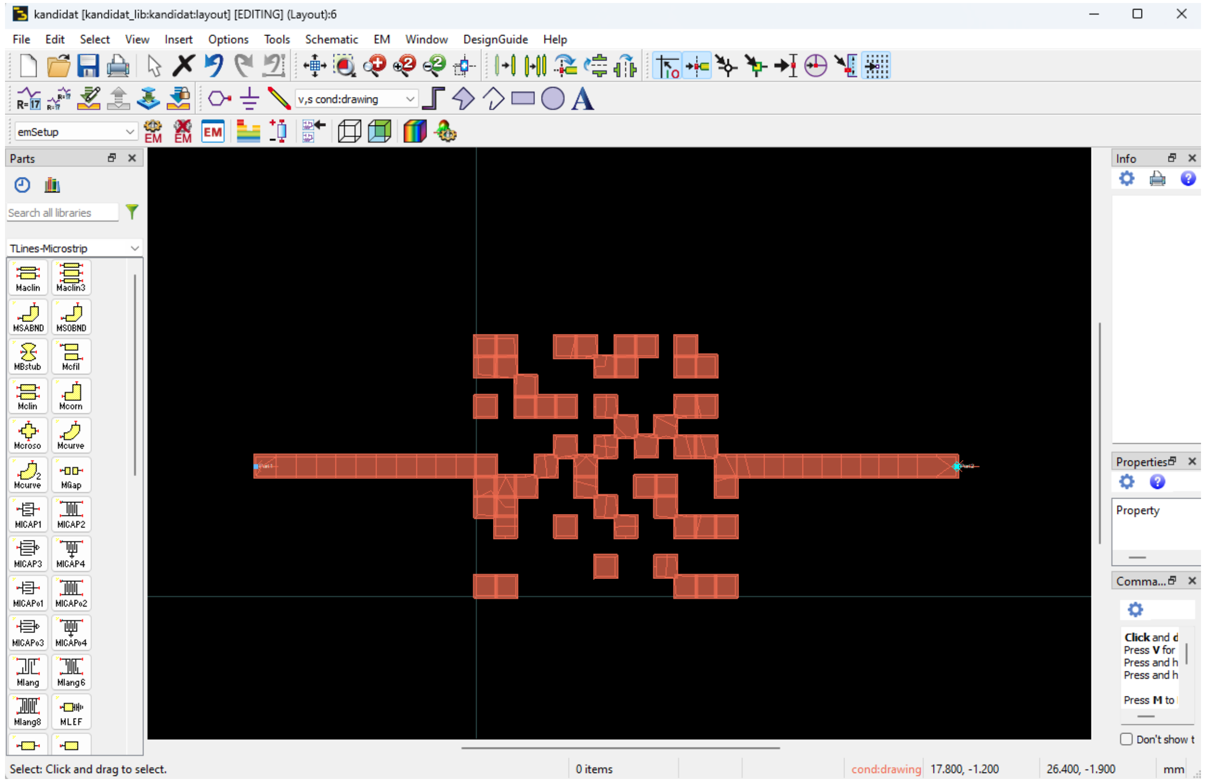
Results

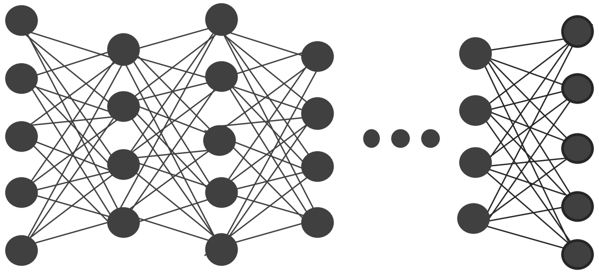


Results

Time requirements:

- Simulate 100 000 circuits ≈ 111 hours
- Train CNN ≈ 8 hours
- Generate one circuit ≈ 50 min



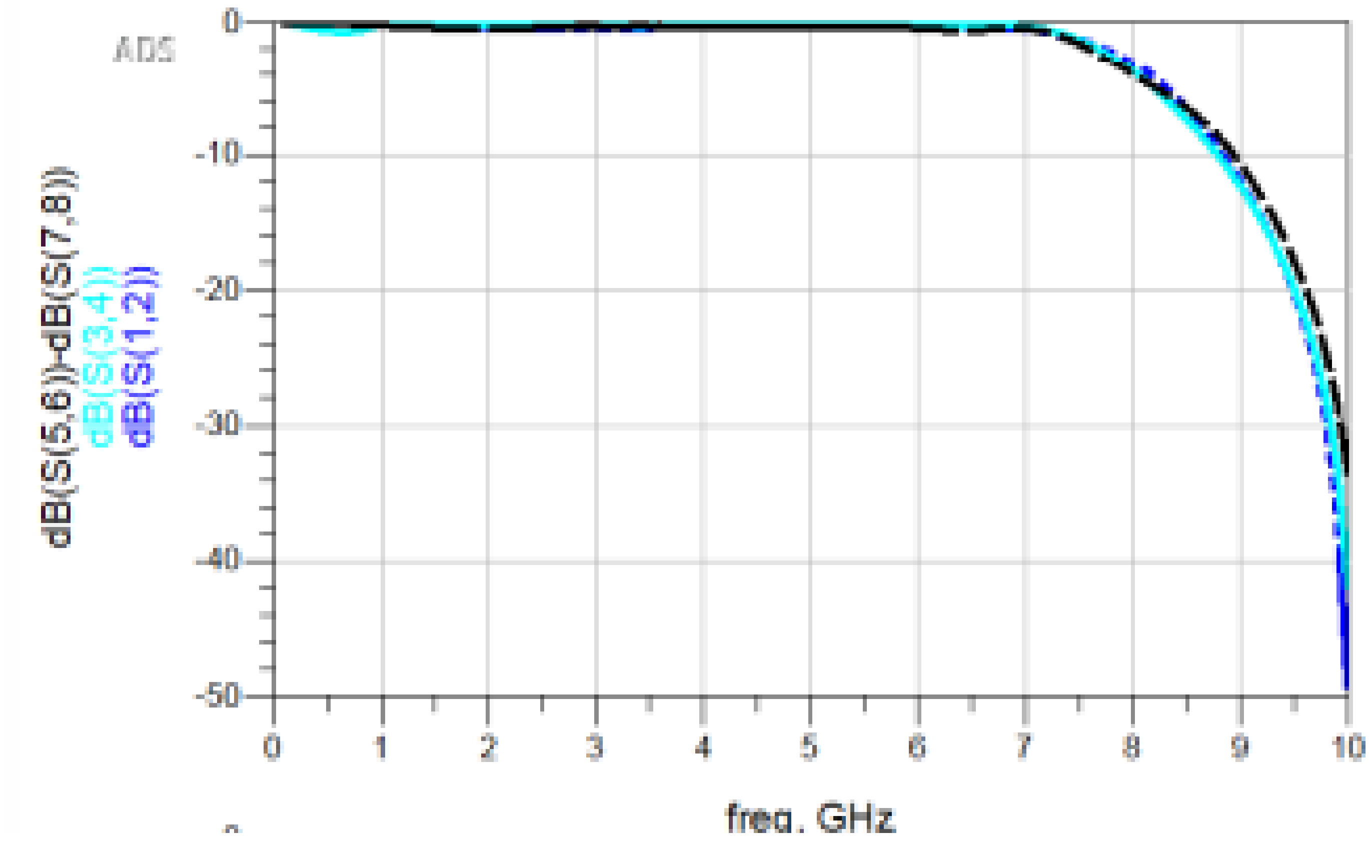


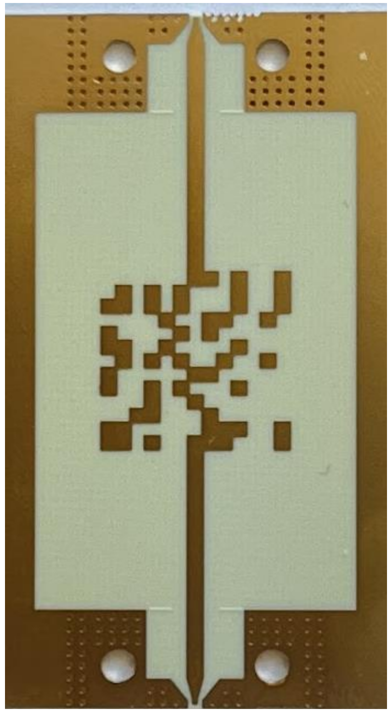
Lowpass filter

ADS MoM MW

FEM (3rd party)

Measurement





Design Goals:

|S21| =1 (f ≤ 7 GHz)

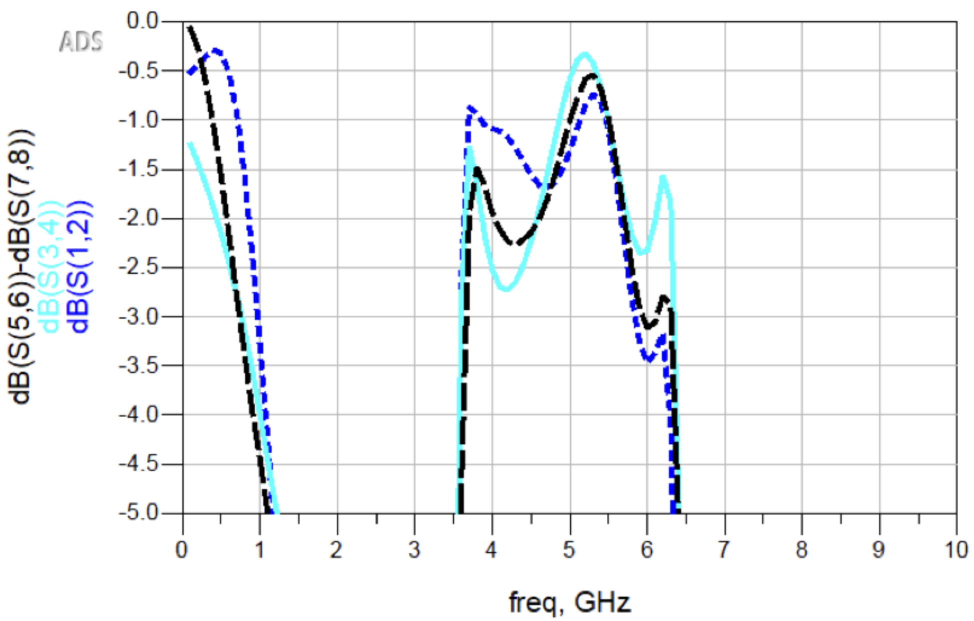
S21=0 (f > 7 GHz)

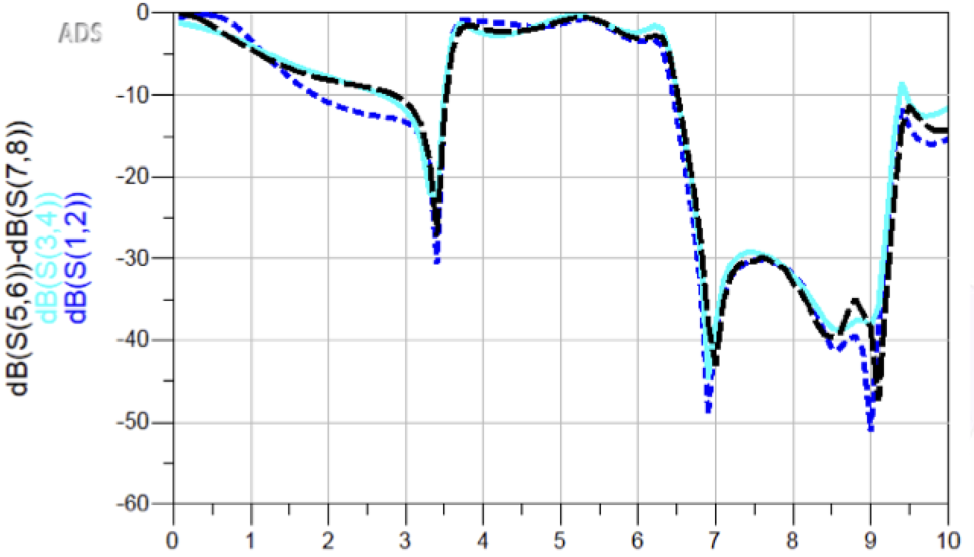
Bandpass filter

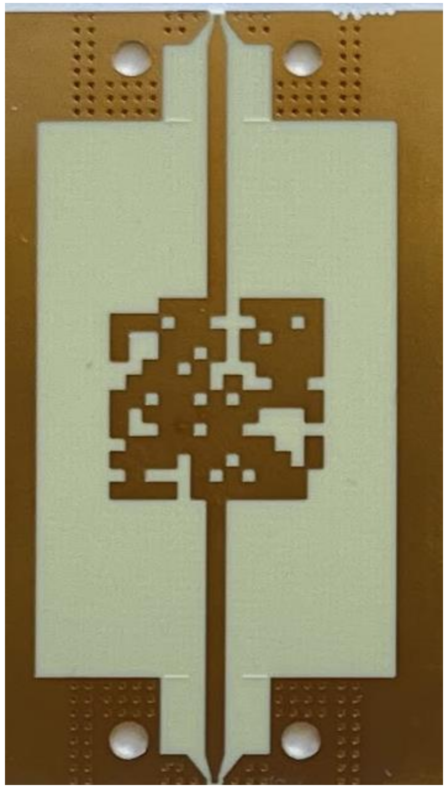
ADS MoM MW

FEM (3rd party)

Measurement





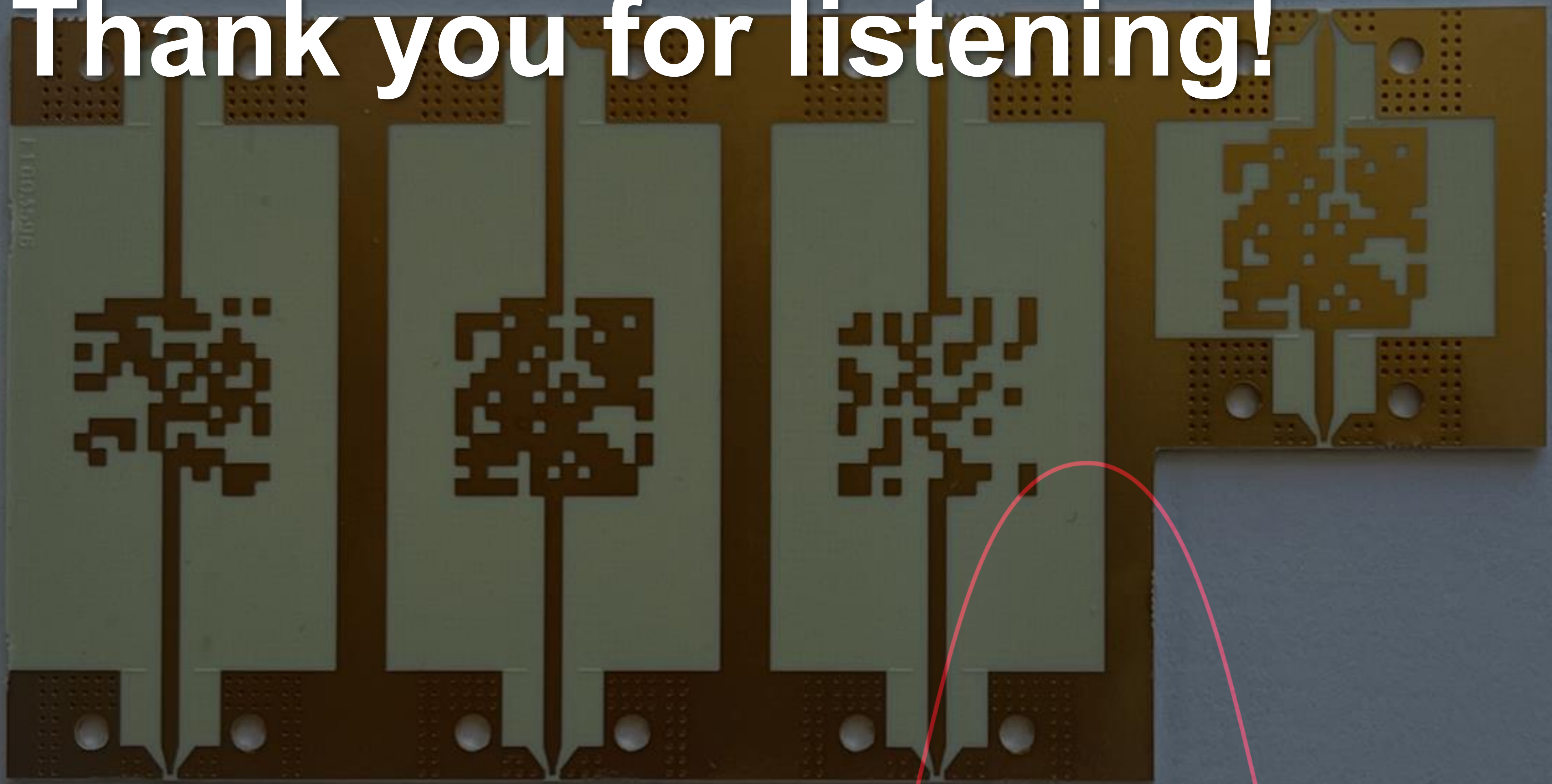


Design Goals:

|S21| =1 (3.5 ≤ f ≤ 6.5 GHz)

S21=0 (otherwise)

Thank you for listening!



Ludvig Fornstedt, Gabriel Melin, David Widén