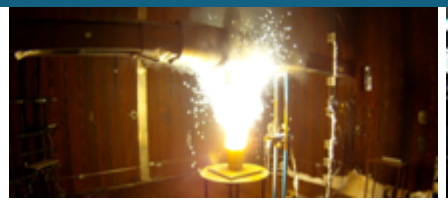
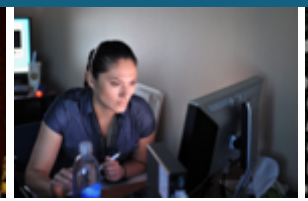
SAND2022-6999C

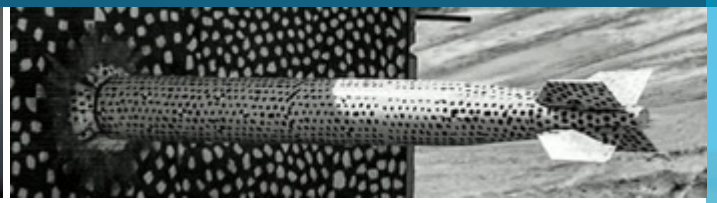
Sandia National Laboratories

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Python Data-Driven Model Integration for the Xyce Circuit Simulator







May 24th, 2021

Paul Kuberry (SNL, 1442)
Biliana Paskaleva (SNL, 1356)
Ting Mei (SNL, 1355)
Eric Keiter (SNL, 1355)
Pavel Bochev (SNL, 1400)
Thomas Buchheit (SNL, 1356)
Matthew Glickman (SNL, 1462)

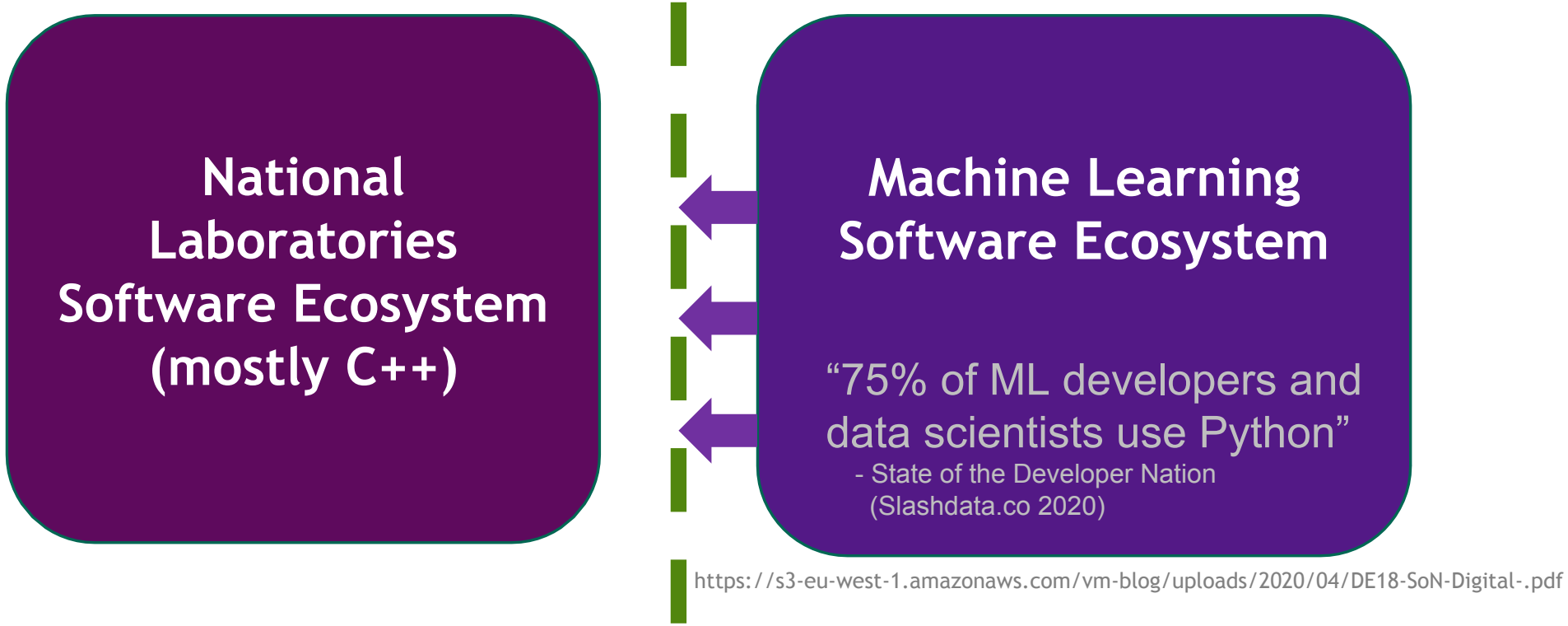
U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY | NNSA

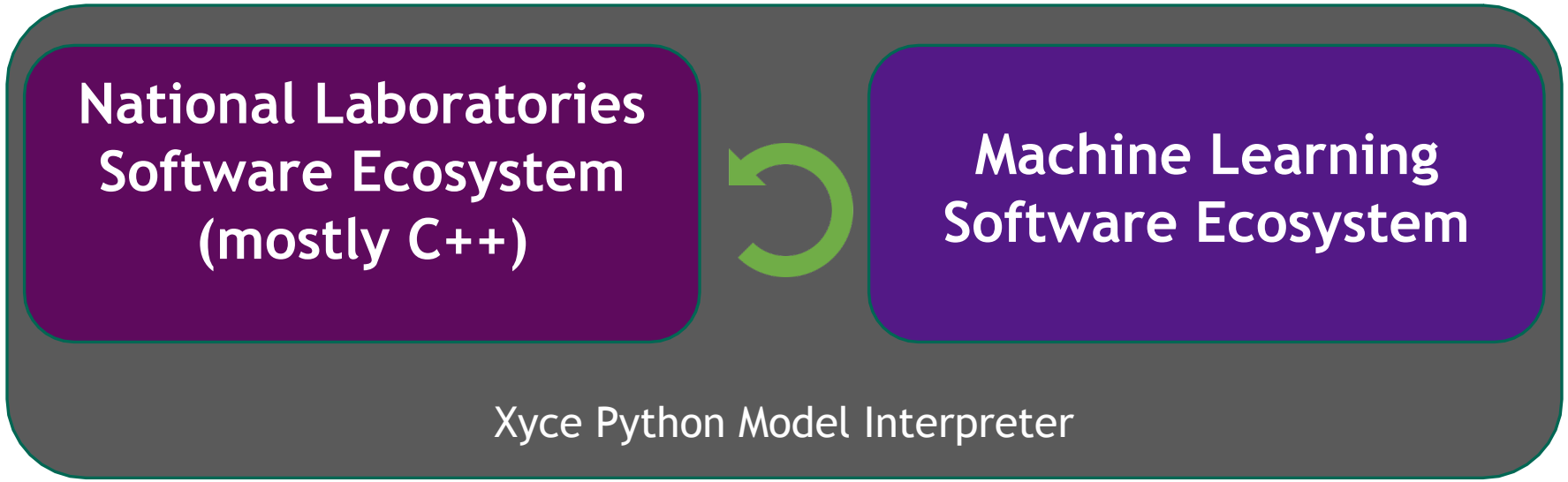
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The need for Python-defined models



Our goal is to enable ML advancements to impact design phases of nuclear deterrence electrical systems via data-driven compact device models

What is Xyce-PyMi?



- Leverages General External interface (C++) from Xyce, by Tom Russo
- Easily installable and callable capability for executing Python ML models in a production circuit simulation software (Xyce ←→ PyBind11)
- Provide data-driven compact device modeling approaches (GMLS, splines, deep neural networks) from ongoing LDRDs at Sandia such as *Data-Driven, Radiation-Aware, Agile Modeling Approach for Rapid Nuclear Deterrence Design Assessment*

Provide a tool that...

- **rapidly assesses impact of device behavior on circuit level performance**
  - Device changes may not be representable by underlying model parameters
- **performs this type of evaluation efficiently**
  - Calibration is not required
  - Process is automated
- **can use experimental or high-fidelity simulation data as an input**
  - Example: Device behavior extracted from a TCAD simulation

How to call and specify device/subcircuit behavior

```
>> Xyce-PyMi example.cir
```

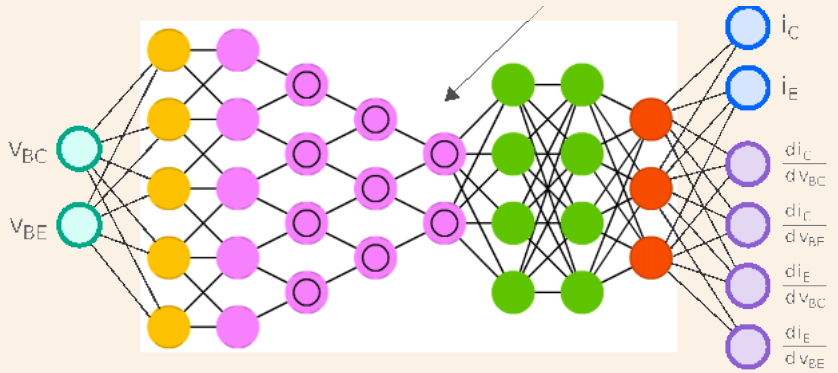
Xyce-PyMi
(Xyce + **P**ython **M**odel **I**nterpreter)

- Full Xyce functionality + devices / subcircuits / circuits defined in Python
- Python class defines how F, Q, B, dF/dX, and dQ/dX vectors/matrices are populated for Xyce DAE equation:
  - residual = f(x,t) + dq(x,t)/dt - b(t)
- Supports all popular machine learning frameworks such as TensorFlow, Keras, PyTorch, Jax, Numba, Numpy, etc…

```
* example.cir (Example of easy inclusion in a netlist)
YGENEXT devicename terminal1 terminal2
+ SPARAMS={NAME=MODULENAME VALUE=PythonDevice.py}
```

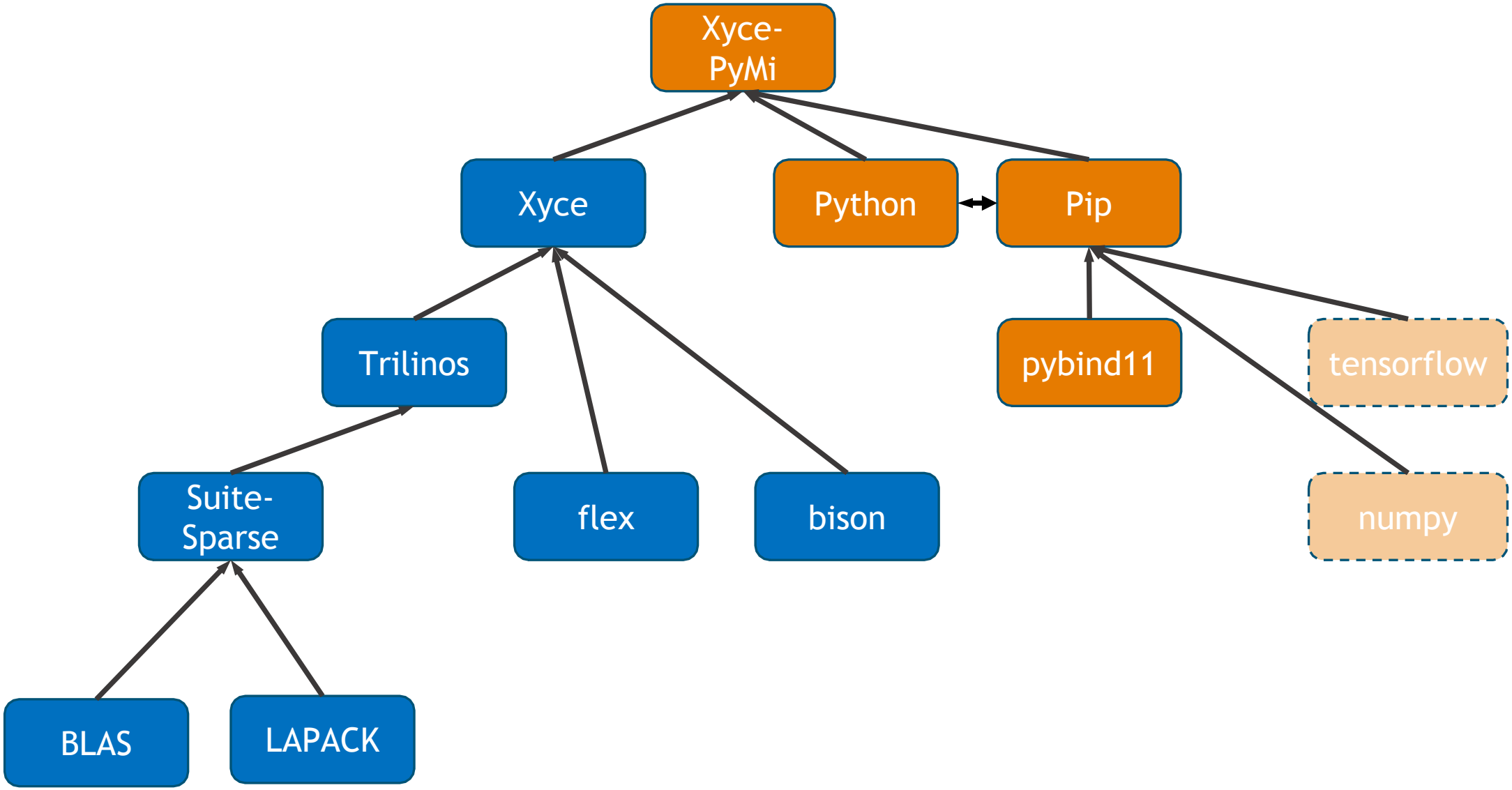
```
# PythonDevice.py
import numpy as np
from BaseDevice import BaseDevice

class Device(BaseDevice):
    def computeXyceVectors(…):
        # definition of device in Python goes here
```



BJT DNN Compact Model

Dependencies of Xyce-PyMi

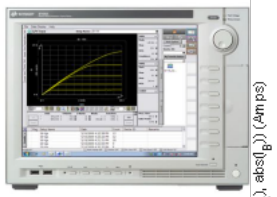


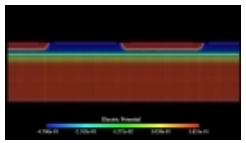
>> spack spec xyce+pymi

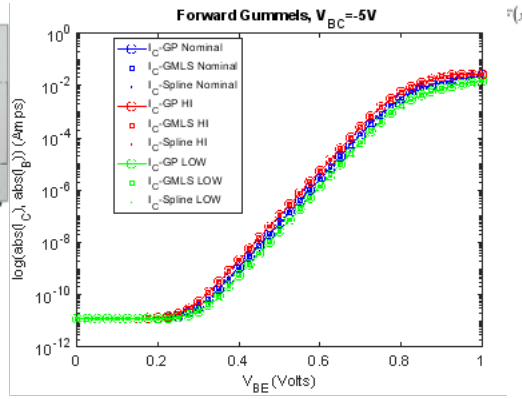
```
xyce@github.master%apple-clang@12.0.0~ipo+mpi~plugin+pymi~shared build_type=Release cxxstd=11 arch=darwin-catalina-skylake
    ^bison@3.8.2%apple-clang@12.0.0 arch=darwin-catalina-skylake
        ^diffutils@3.8%apple-clang@12.0.0 arch=darwin-catalina-skylake
            ^libiconv@1.16%apple-clang@12.0.0 libs=shared,static arch=darwin-catalina-skylake
        ^m4@1.4.19%apple-clang@12.0.0+sigsegv patches=9dc5fbd,bfdffa7 arch=darwin-catalina-skylake
            ^libsigsegv@2.13%apple-clang@12.0.0 arch=darwin-catalina-skylake
        ^perl@5.34.0%apple-clang@12.0.0+cpanm+shared+threads arch=darwin-catalina-skylake
            ^berkeley-db@18.1.40%apple-clang@12.0.0+cxx~docs+stl patches=b231fcc arch=darwin-catalina-skylake
            ^bzip2@1.0.8%apple-clang@12.0.0~debug~pic+shared arch=darwin-catalina-skylake
            ^gdbm@1.19%apple-clang@12.0.0 arch=darwin-catalina-skylake
                ^readline@8.1%apple-clang@12.0.0 arch=darwin-catalina-skylake
                    ^ncurses@6.2%apple-clang@12.0.0~symlinks+termlib abi=none arch=darwin-catalina-skylake
                        ^pkgconf@1.8.0%apple-clang@12.0.0 arch=darwin-catalina-skylake
            ^zlib@1.2.11%apple-clang@12.0.0+optimize+pic+shared arch=darwin-catalina-skylake
    ^cmake@3.22.3%apple-clang@12.0.0~doc+ncurses+openssl+ownlibs~qt build_type=Release arch=darwin-catalina-skylake
        ^openssl@1.1.1n%apple-clang@12.0.0~docs~shared certs=system arch=darwin-catalina-skylake
    ^flex@2.6.3%apple-clang@12.0.0+lex~nls arch=darwin-catalina-skylake
        ^findutils@4.8.0%apple-clang@12.0.0 patches=440b954 arch=darwin-catalina-skylake
    ^openmpi@4.1.2%apple-clang@12.0.0~atomics~cuda~cxx~cxx_exceptions~gpfs~internal-hwloc~java~legacylaunchers~lustre~memchecker~pmi~pmix+romio~singularity~sqlite3+static~thread_multiple+vt+wrapper-rpath fabrics=n
schedulers=none arch=darwin-catalina-skylake
        ^hwloc@2.7.0%apple-clang@12.0.0~cairo~cuda~gl~libudev+libxml2~netloc~nvml~opencl~pci~rocm+shared arch=darwin-catalina-skylake
            ^libxml2@2.9.12%apple-clang@12.0.0~python arch=darwin-catalina-skylake
                ^xz@5.2.5%apple-clang@12.0.0~pic libs=shared,static arch=darwin-catalina-skylake
        ^libevent@2.1.12%apple-clang@12.0.0+openssl arch=darwin-catalina-skylake
        ^openssh@8.9p1%apple-clang@12.0.0 arch=darwin-catalina-skylake
            ^libedit@3.1-20210216%apple-clang@12.0.0 arch=darwin-catalina-skylake
    ^py-pybind11@2.9.1%apple-clang@12.0.0~ipo build_type=RelWithDebInfo arch=darwin-catalina-skylake
    ^py-pycompadre@master%apple-clang@12.0.0+trilinos debug=0 arch=darwin-catalina-skylake
        ^trilinos@13.2.0%apple-
clang@12.0.0~adios2+amesos+amesos2+anasazi+aztec+basker+belos~boost~chaco+complex~cuda~cuda_rdc~debug~dtk+epetra+epetraext+epetraextbtf+epetraextexperimental+epetraextgraphreorderings~exodus+explicit_templa
oat+fortran~gtest+hdf5~hypre+ifpack~ifpack2~intrepid~intrepid2~ipo+isorropia+kokkos~mesquite~minitensor~ml+mpi~muelu~mumps+nox~openmp~panzer~phalanx~piro~python~rocm~rocm_rdc~rol~rythmos+sacado~scorec~sharc
stk+stokhos~stratimikos~strumpack+suite-sparse~superlu~superlu-dist~teko~tempus~thyra+tpetra+trilinoscouplings~wrapper~x11+zoltan~zoltan2 build_type=RelWithDebInfo cxxstd=14 gotype=long_long patches=b9614a5 arch=da
skylake
            ^hdf5@1.12.1%apple-clang@12.0.0~cxx~fortran+hl~ipo~java+mpi+shared~szip~threadsafe+tools api=default build_type=RelWithDebInfo patches=ee351eb arch=darwin-catalina-skylake
            ^metis@5.1.0%apple-clang@12.0.0~gdb~int64~real64+shared build_type=Release patches=4991da9 arch=darwin-catalina-skylake
            ^parmetis@4.0.3%apple-clang@12.0.0~gdb~int64~ipo+shared build_type=RelWithDebInfo patches=4f89253,50ed208,704b84f arch=darwin-catalina-skylake
            ^suite-sparse@5.10.1%apple-clang@12.0.0~cuda~graphblas~openmp+pic~tbb arch=darwin-catalina-skylake
                ^gmp@6.2.1%apple-clang@12.0.0 arch=darwin-catalina-skylake
                    ^autoconf@2.69%apple-clang@12.0.0 patches=35c4492,7793209,a49dd5b arch=darwin-catalina-skylake
                    ^automake@1.16.5%apple-clang@12.0.0 arch=darwin-catalina-skylake
                    ^libtool@2.4.6%apple-clang@12.0.0 arch=darwin-catalina-skylake
                ^mpfr@4.1.0%apple-clang@12.0.0 arch=darwin-catalina-skylake
                    ^autoconf-archive@2019.01.06%apple-clang@12.0.0 arch=darwin-catalina-skylake
                    ^texinfo@6.5%apple-clang@12.0.0 patches=12f6edb,1732115 arch=darwin-catalina-skylake
```

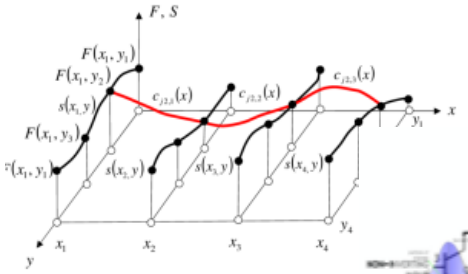
Installation using Spack

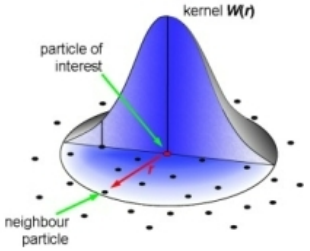
Our “data-to-simulation” pipeline in Xyce-PyMi

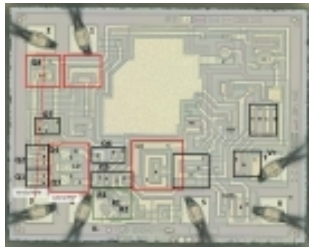


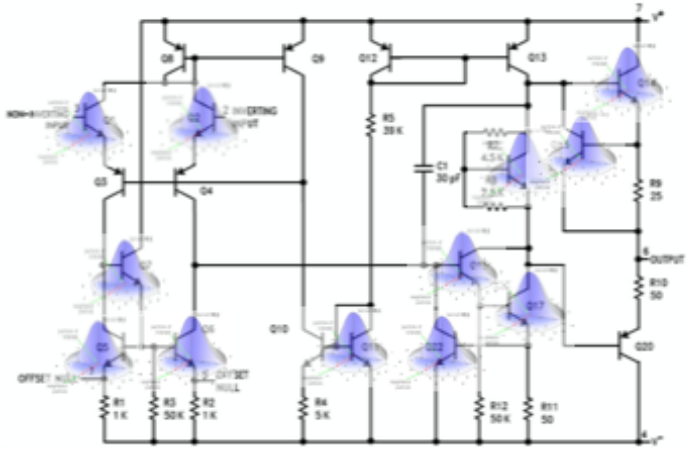


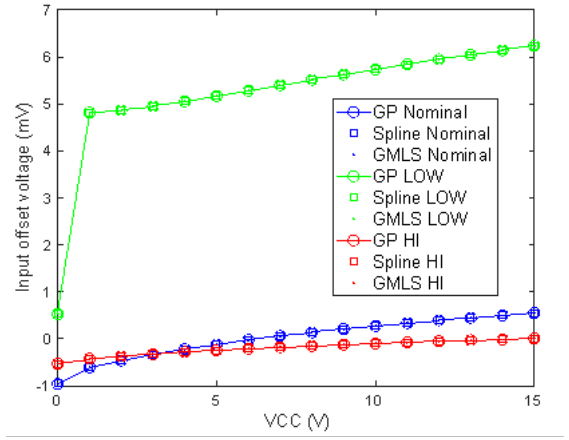












Xyce-PyMi

Measured or simulated (TCAD)

Data-driven models

Xyce-PyMi LM148 circuit model

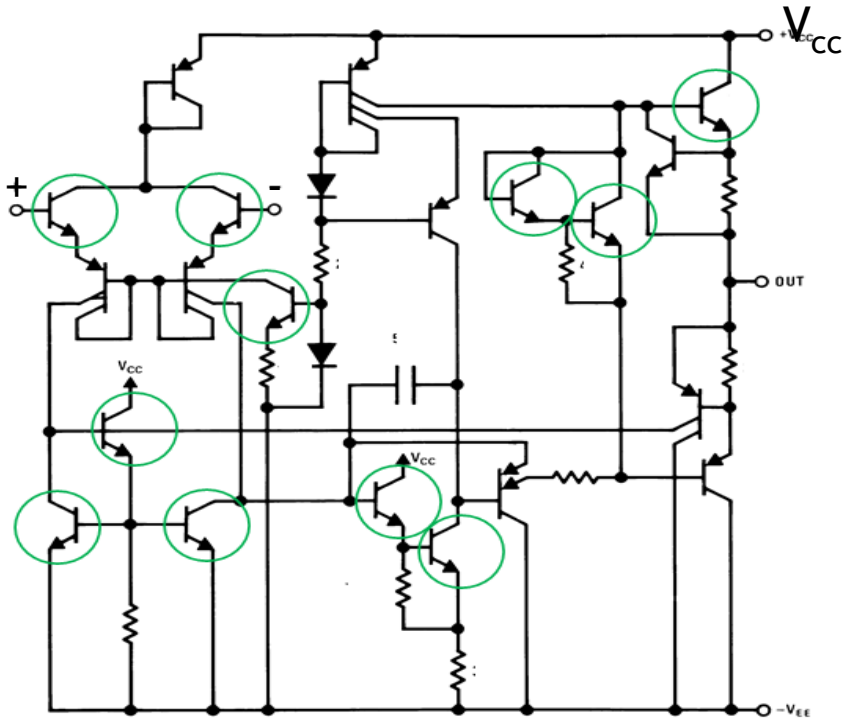
Data-driven LM148 simulations

Given data sets, time to execute: minutes to few days

Demonstration Case: LM148 Op-Amp

Vendor Supplied Model- Validated with Xyce

LM148 Op-Amp Schematic

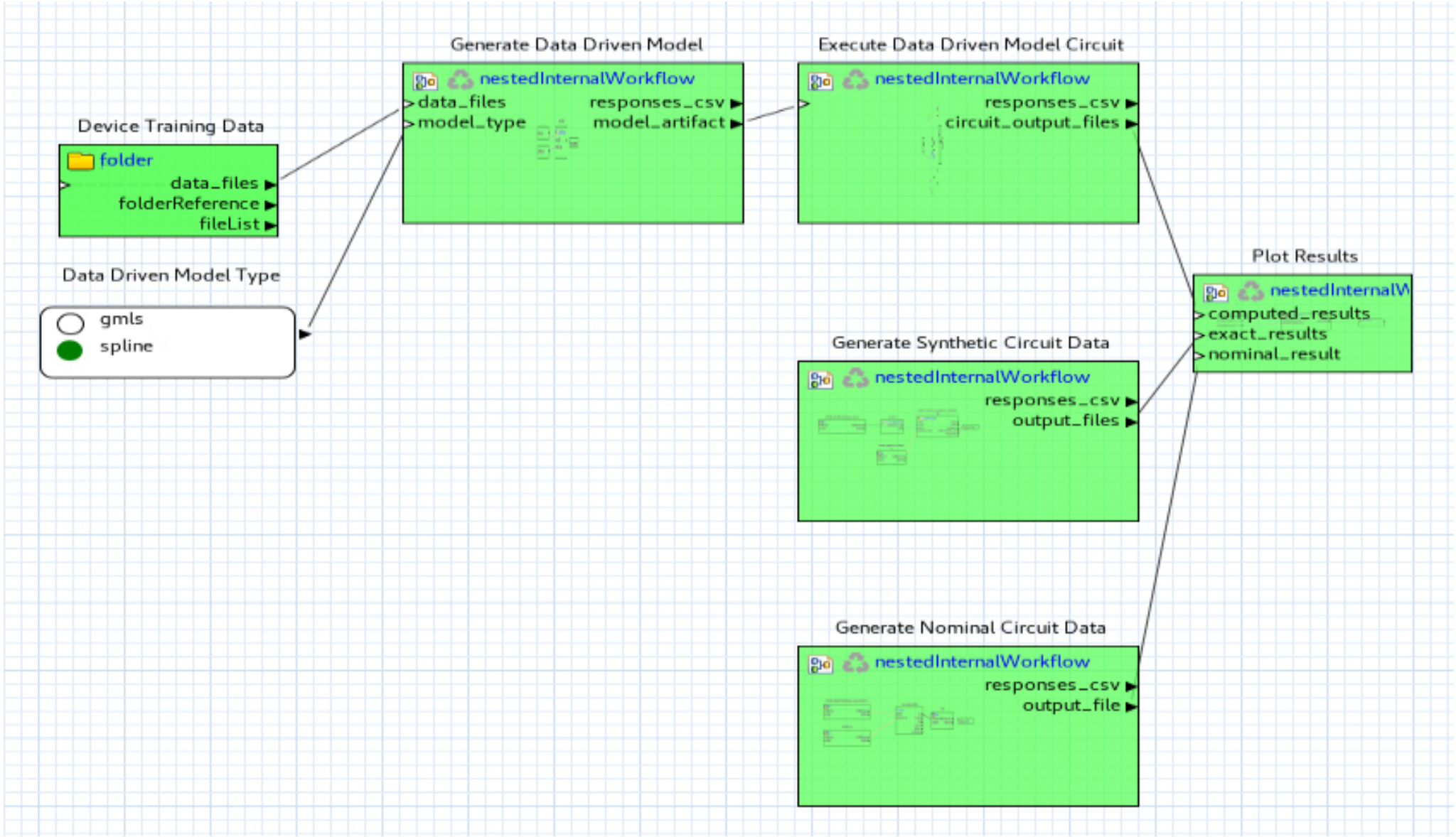


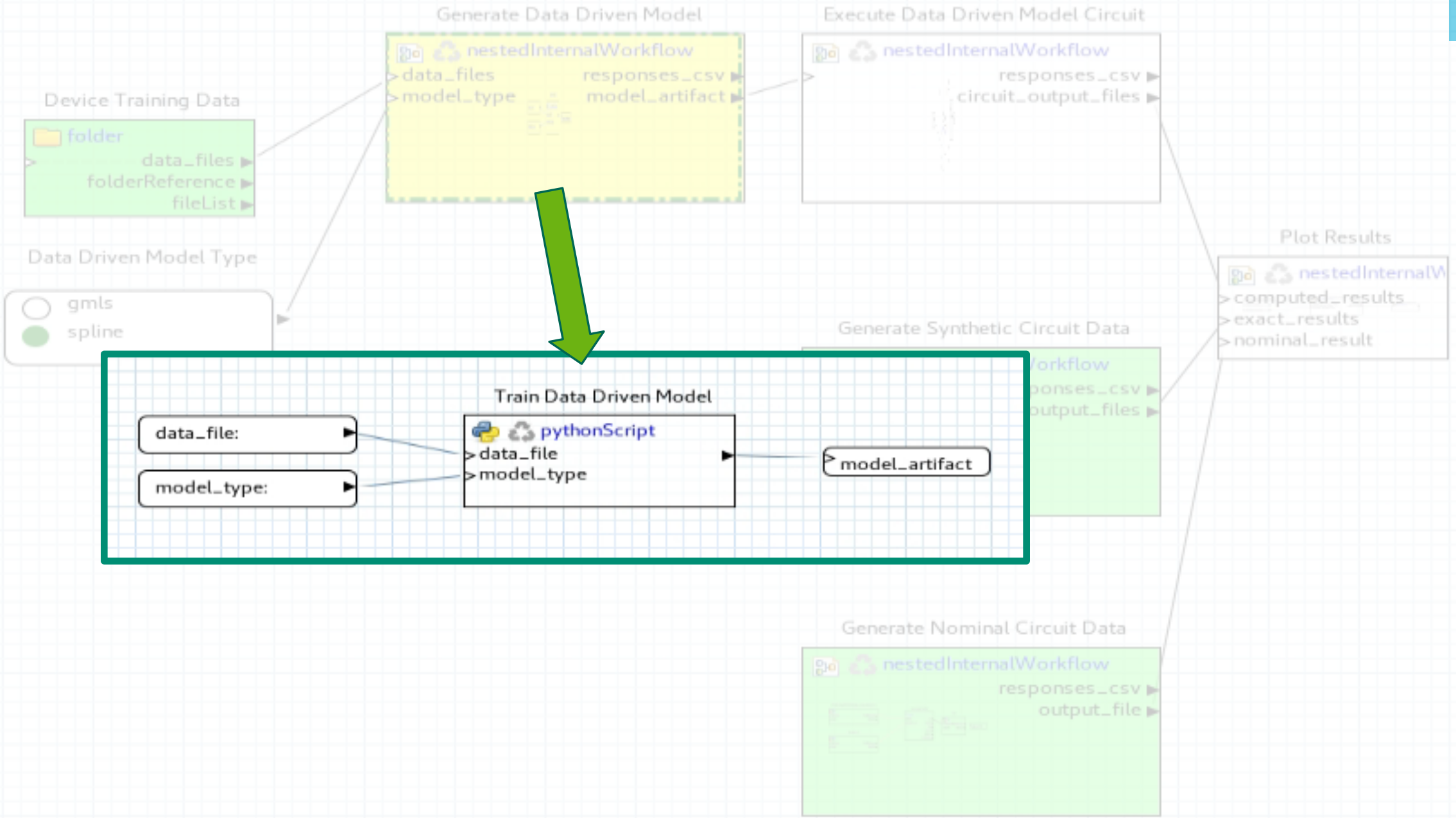
NPN transistors identified

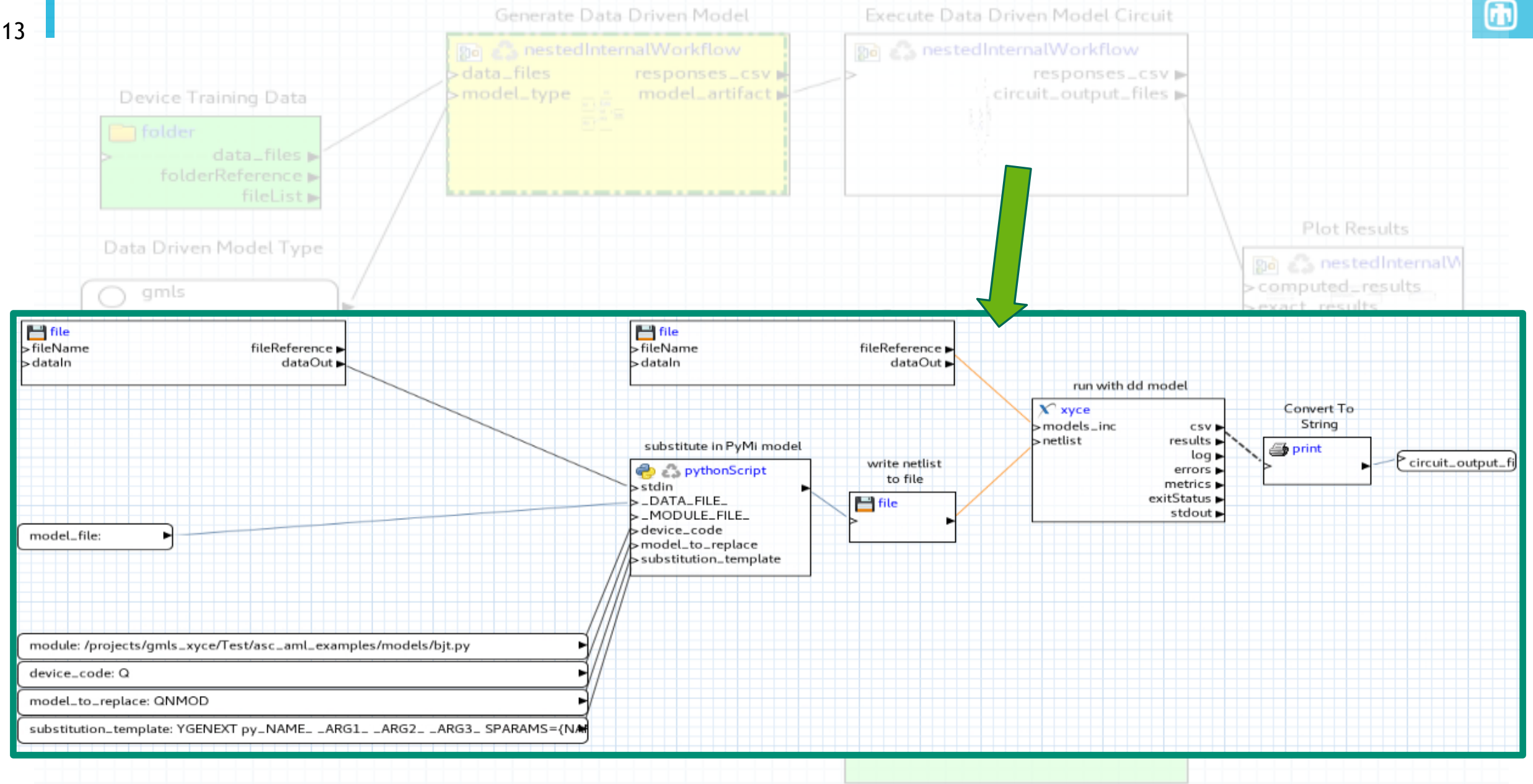
- Parameters thought to affect the forward characteristics of a transistor due to aging:
  Gain, Emitter Resistance, and Transport Saturation Current
- Generate 30 transistor datasets by perturbing the aging related parameters
- Extract 30 data driven (DD) models
- Execute 30 circuit simulations based on the 30 DD models, substituting all NPN transistors in the circuit with the same DD model instance

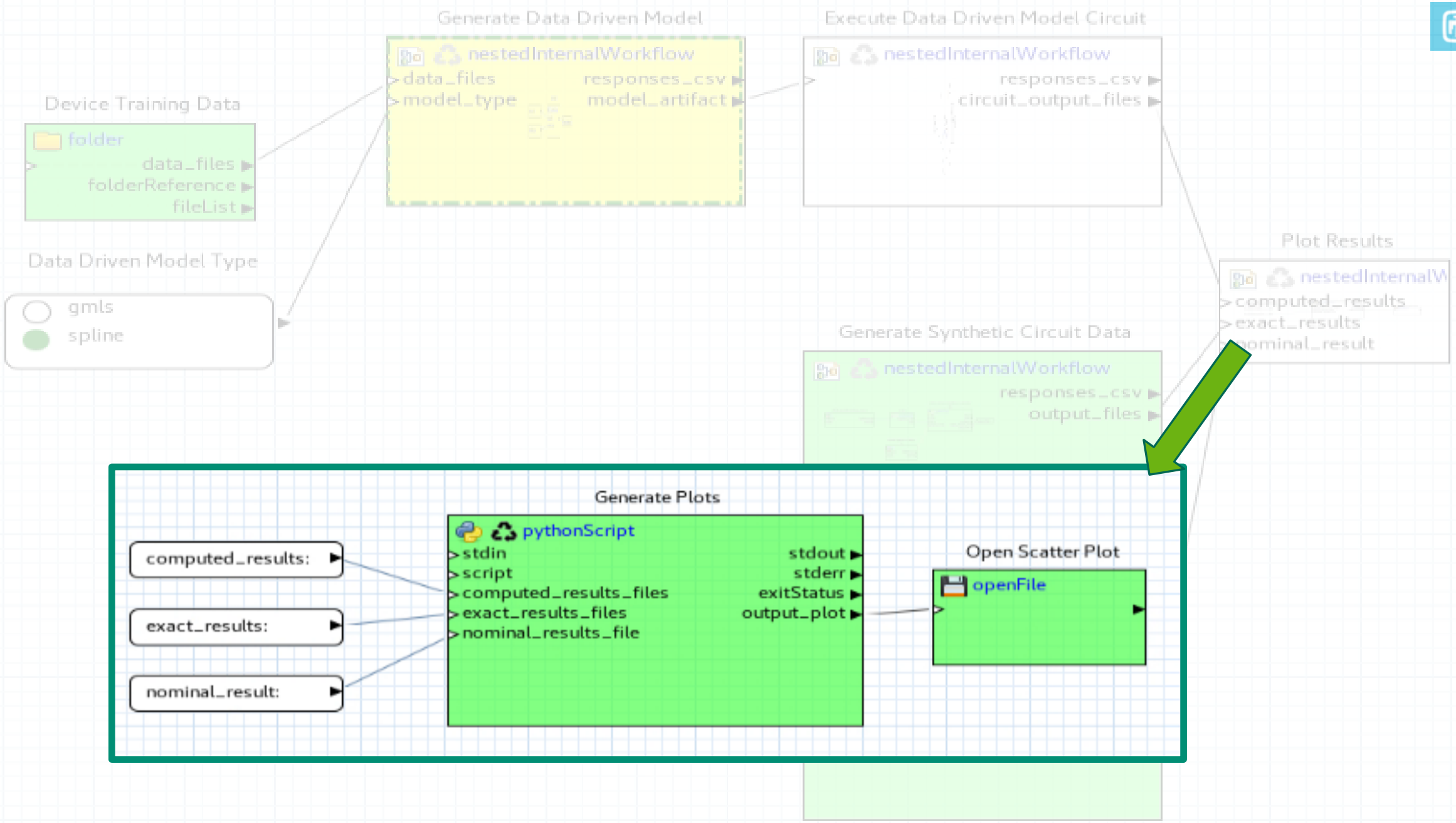
- Generate a statistical sampling of datasets for data-driven models derived from a gaussian distribution of three parameters used in the compact models for the NPN Transistors in the LM148.

Sandia Analysis Workbench (SAW) workflow

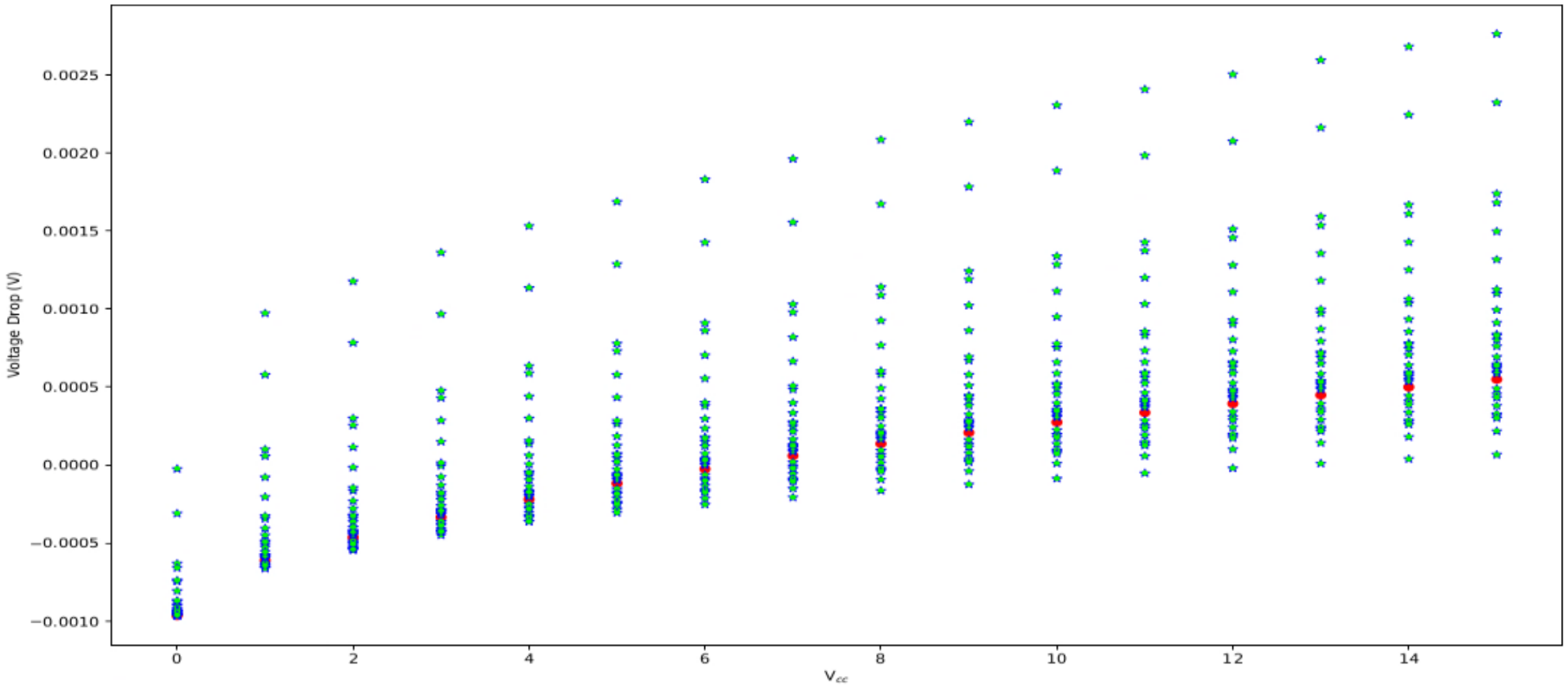



Generate Data Driven Model
nestedInternalWorkflow
data_files
model_type
responses_csv
model_artifact
Execute Data Driven Model Circuit
nestedInternalWorkflow
responses_csv
circuit_output_files
Device Training Data
folder
data_files
folderReference
fileList
Data Driven Model Type
gmls
spline
Plot Results
computed_results
exact_results
nominal_result
Generate Synthetic Circuit Data
Train Data Driven Model
data_file:
model_type:
pythonScript
data_file
model_type
model_artifact
Generate Nominal Circuit Data
nestedInternalWorkflow
responses_csv
output_file

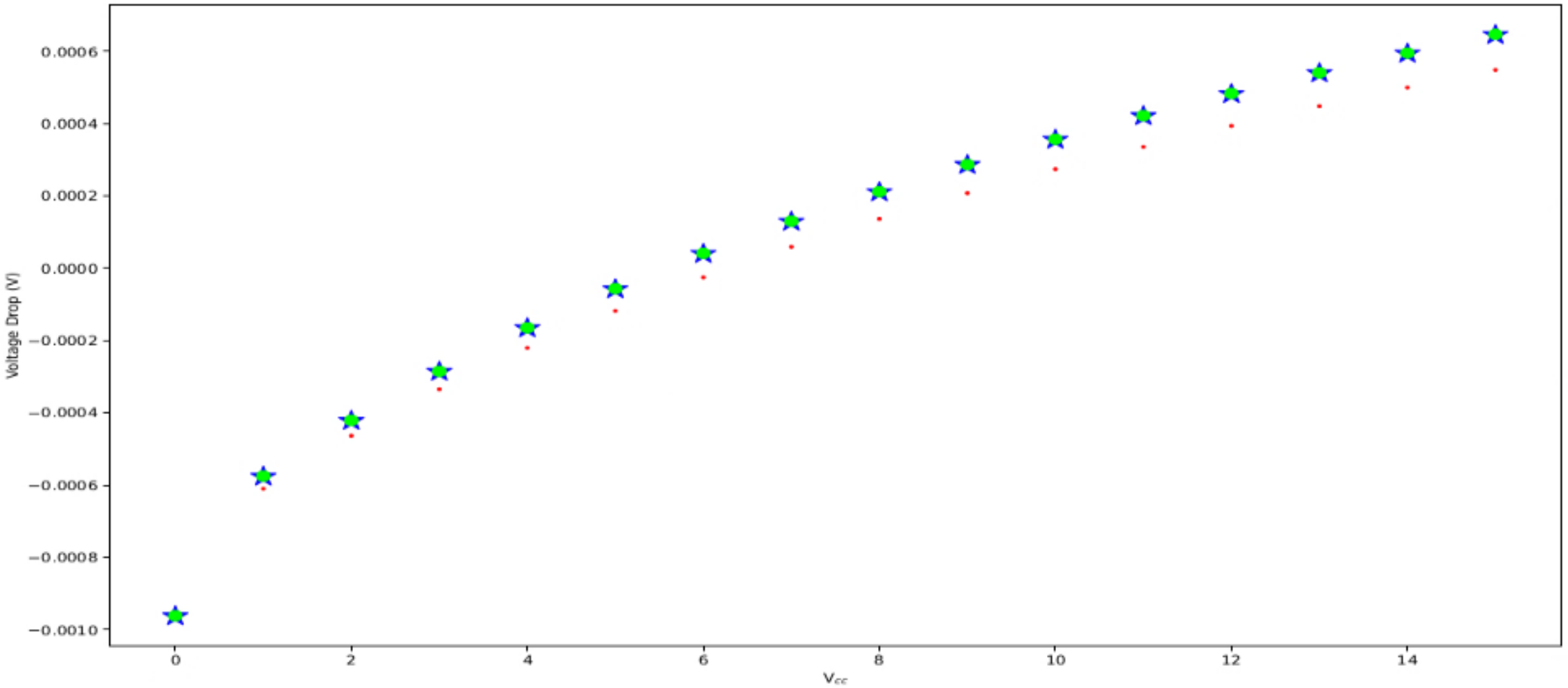

Generate Data Driven Model
nestedInternalWorkflow
data_files
model_type
responses_csv
model_artifact
Execute Data Driven Model Circuit
nestedInternalWorkflow
responses_csv
circuit_output_files
Device Training Data
folder
data_files
folderReference
fileList
Data Driven Model Type
gmls
Plot Results
nestedInternalW
computed_results
file
fileName
dataIn
fileReference
dataOut
model_file:
substitute in PyMi model
pythonScript
stdin
_DATA_FILE_
_MODULE_FILE_
device_code
model_to_replace
substitution_template
write netlist
to file
run with dd model
xyce
models_inc
netlist
csv
results
log
errors
metrics
exitStatus
stdout
Convert To
String
print
circuit_output_fi
module: /projects/gmls_xyce/Test/asc_aml_examples/models/bjt.py
device_code: Q
model_to_replace: QNMOD
substitution_template: YGENEXT py_NAME_ _ARG1_ _ARG2_ _ARG3_ SPARAMS={NA


Generate Data Driven Model
nestedInternalWorkflow
data_files
model_type
responses_csv
model_artifact
Execute Data Driven Model Circuit
nestedInternalWorkflow
responses_csv
circuit_output_files
Device Training Data
folder
data_files
folderReference
fileList
Data Driven Model Type
gmls
spline
Plot Results
computed_results
exact_results
Generate Synthetic Circuit Data
nestedInternalWorkflow
responses_csv
output_files
Generate Plots
computed_results:
exact_results:
nominal_result:
pythonScript
stdin
script
computed_results_files
exact_results_files
nominal_results_file
stdout
stderr
exitStatus
output_plot
Open Scatter Plot
openFile

Comparison of 30 data-driven circuit results



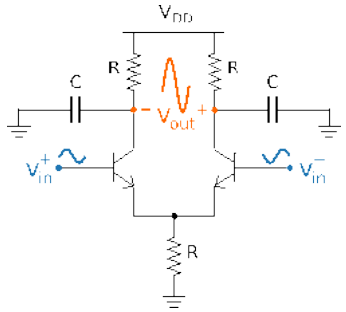
Single comparison of data-driven vs. synthetic circuit

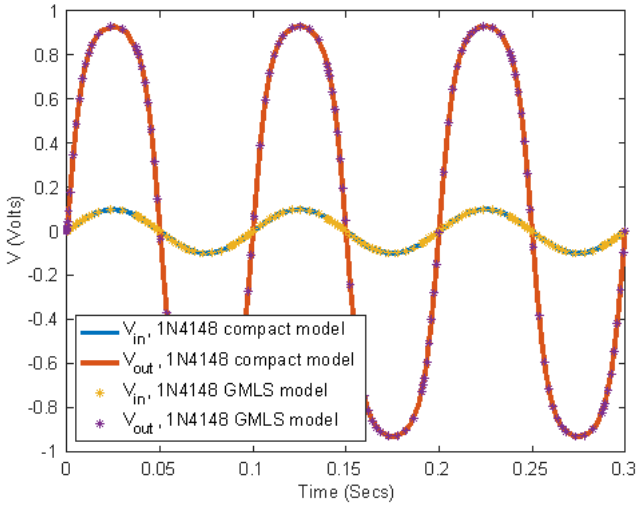


Examples

Operational Amplifier with BJTs

```
**************
* Netlist for Operational Amplifier
**************
VDD    1   0  DC   2.5
R1 1 4 1e4
R2 1 5 1e4
R3 6 0 5e3
C1 4 0 5e-12
C2 5 0 5e-12
YGENEXT pyQ1 4 7 6
+ SPARAMS={NAME=MODULENAME,DATAFILE
        VALUE=../models/gmls_bjt_2N2222.py,../data/2N2222_alan.01.dat}
RQ1 7 2 50
YGENEXT pyQ2 5 8  6
+ SPARAMS={NAME=MODULENAME,DATAFILE
        VALUE=../models/gmls_bjt_2N2222.py,../data/2N2222_alan.01.dat}
RQ2 8 3 50
Em_plus 2 0 VALUE={1+50e-3*sin(2*pi*10*time)}
Em_minus 3 0 VALUE={1-50e-3*sin(2*pi*10*time)}
```

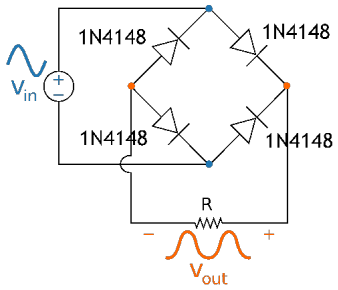


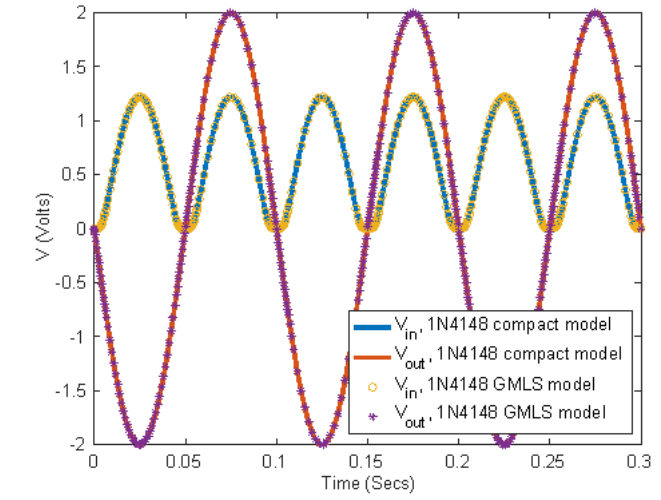


* Runs GMLS on data generated from synthetic MMBT2222, NPN, Fairchild

Fast switching 1N4148 diode in bridge rectifier

```
**************
* Netlist for Bridge Rectifier
**************
V3 1 2 SIN (0 2 10)
R3 3 0 10M
R4 3 4 100K
YGENEXT pyd3 1 4
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py,../data/1N4148_synthetic.dat}
YGENEXT pyd1 3 1
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py,../data/1N4148_synthetic.dat}
YGENEXT pyd4 3 2
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py,../data/1N4148_synthetic.dat}
YGENEXT pyd2 2 4
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py,../data/1N4148_synthetic.dat}
.TRAN 0 0.3s
.options timeint reltol=1.0e-4
.PRINT TRAN V(1) V(2) V(3) V(4) V(2,1) V(4,3)
.END
```

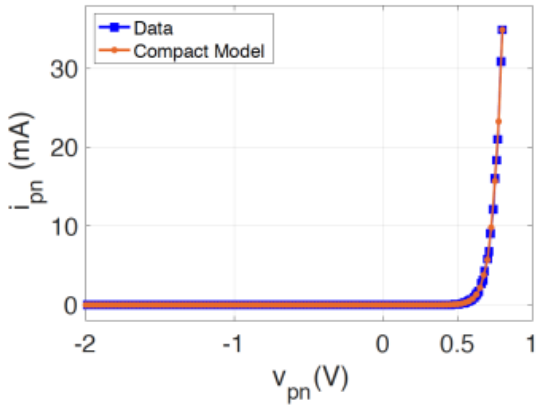


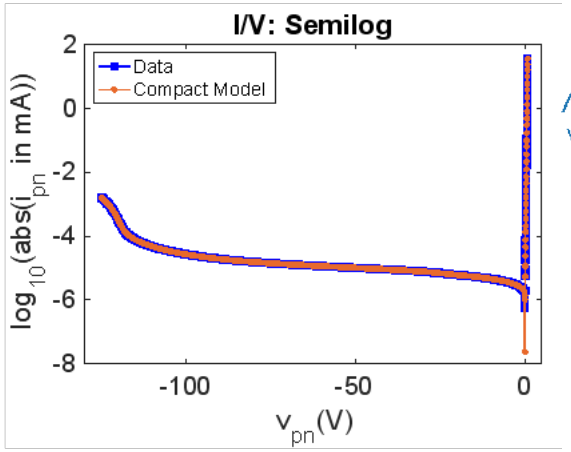


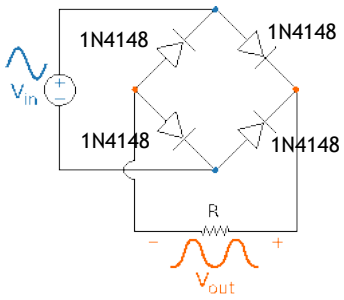
* Runs GMLS on data generated from synthetic 1N4148

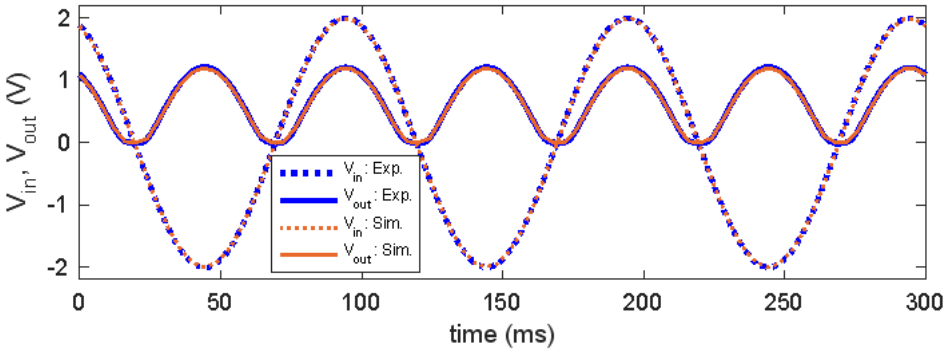
Example results using experiments

Fast switching diode 1N4148 spline model

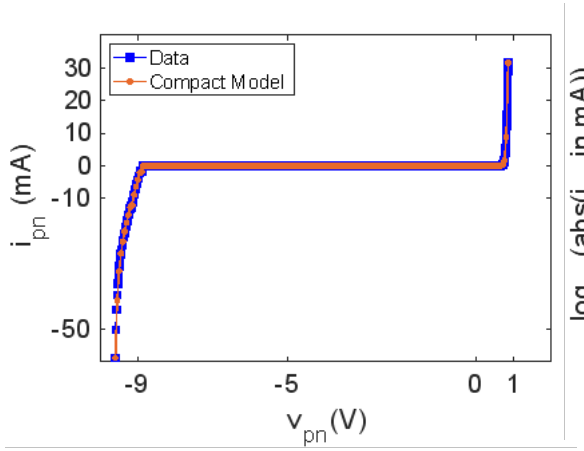


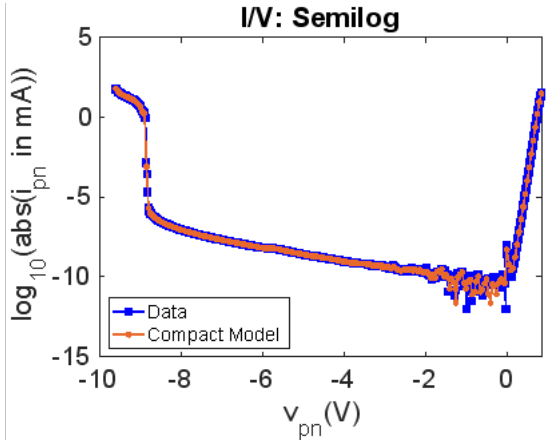


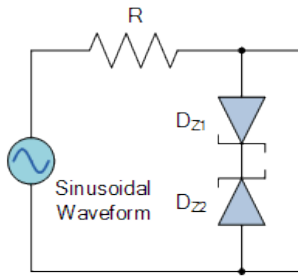


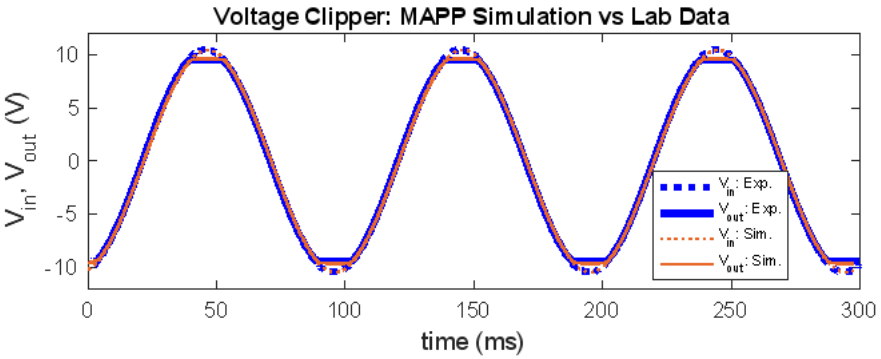


Zener diode MMSZ5239 (9.1 V, 500 mW)









Questions?

Installation / usage details:

P. Kuberry, E. Keiter, **An embedded Python model interpreter for Xyce (Xyce-PyMi)**, *Sandia Report SAND2021-9504*, 2021. doi:10.2172/1813650.

https://spack.readthedocs.io