SAND2019-1451C

Image Analysis, Automation, and Machine Learning Techniques Applied to Is MOS Quantum Dot Tune-Up

PRESENTED BY

Andy Mounce

Phillip J. Lewis, Cara P. Monical, N. Tobias Jacboson, Martin Rudolph, John Anderson, Joel R. Wendt, Tammy Pluym, Dan R. Ward, Kurt

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.

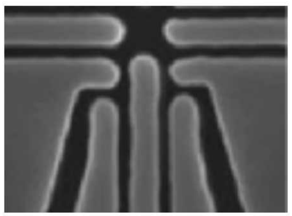
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Outline

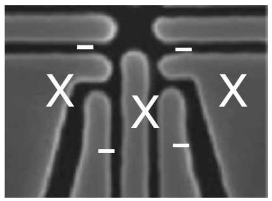
1. Zero to charge sensor automation
2. Ingredients for n-quantum dot auto-tune
3. Machine learning integration
4. Image analysis techniques
5. Device simulator
6. Auto-tune n-quantum dots

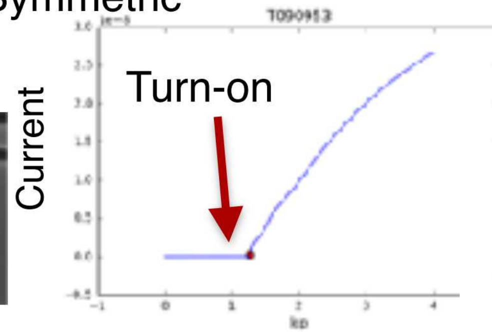
Previous work, presented at QCPR 2018 "From zero to charge sensor"

1. Start (unknown device configuration)

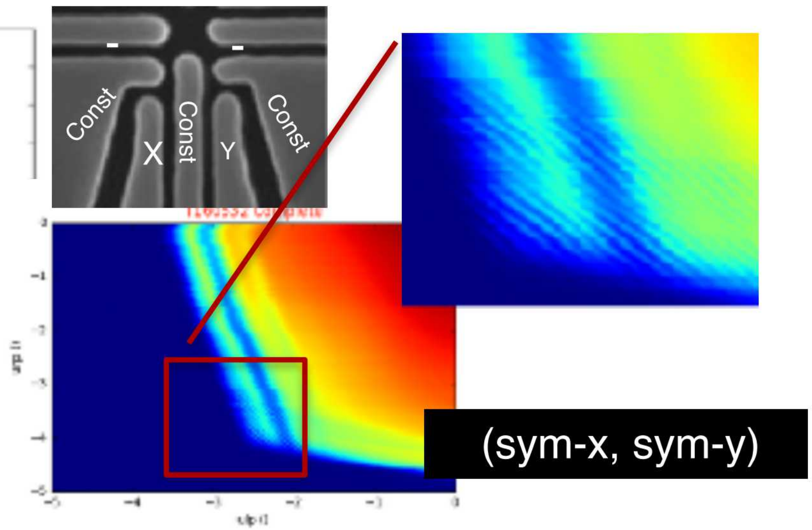


2. Find Symmetric Turn-on

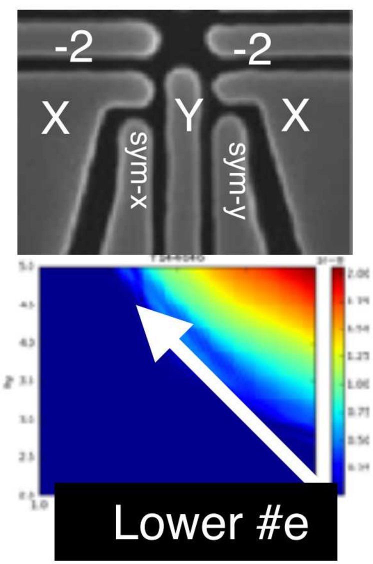




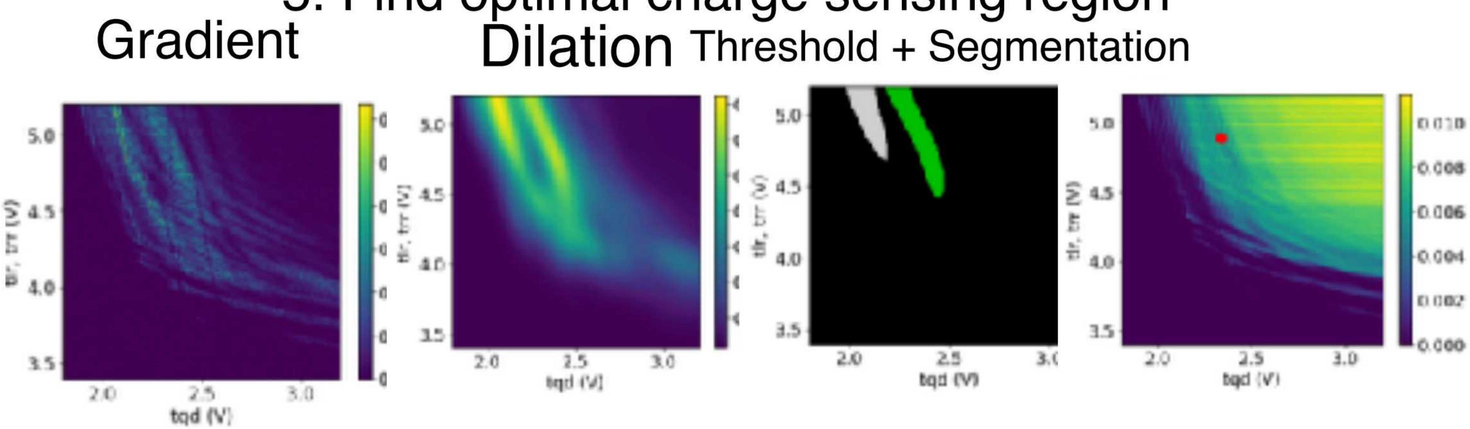
3. Symmetrize left and right sides



4. Symmetrically increase reservoirs while depleting QD



5. Find optimal charge sensing region

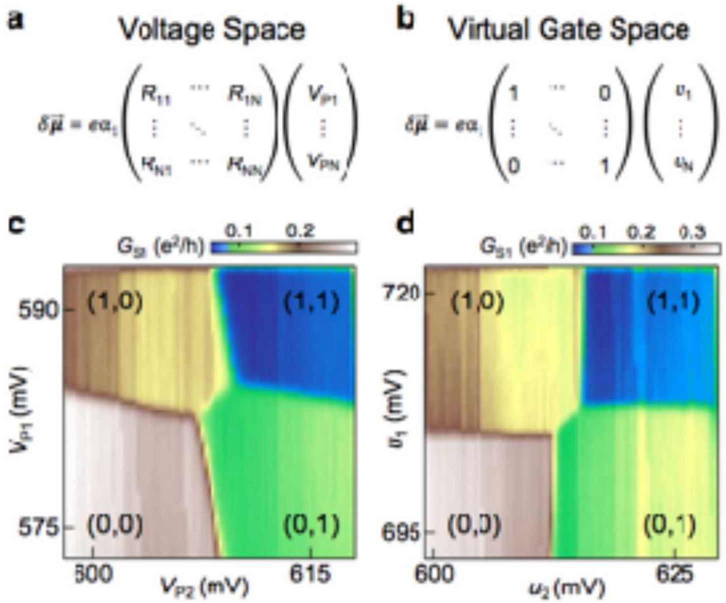


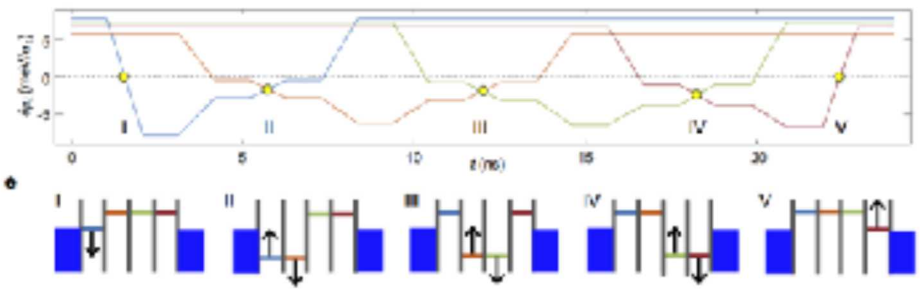
Choose center of largest region

Goals for analysis, ML, and automation

Highest level goal:

- Find the capacitance matrix for gate to dot
- Find the dot-dot mutual capacitance matrix
  - Create virtual gates automatically
  - Navigate charge stability space by indicating spin filling/chemical potential, rather than voltages



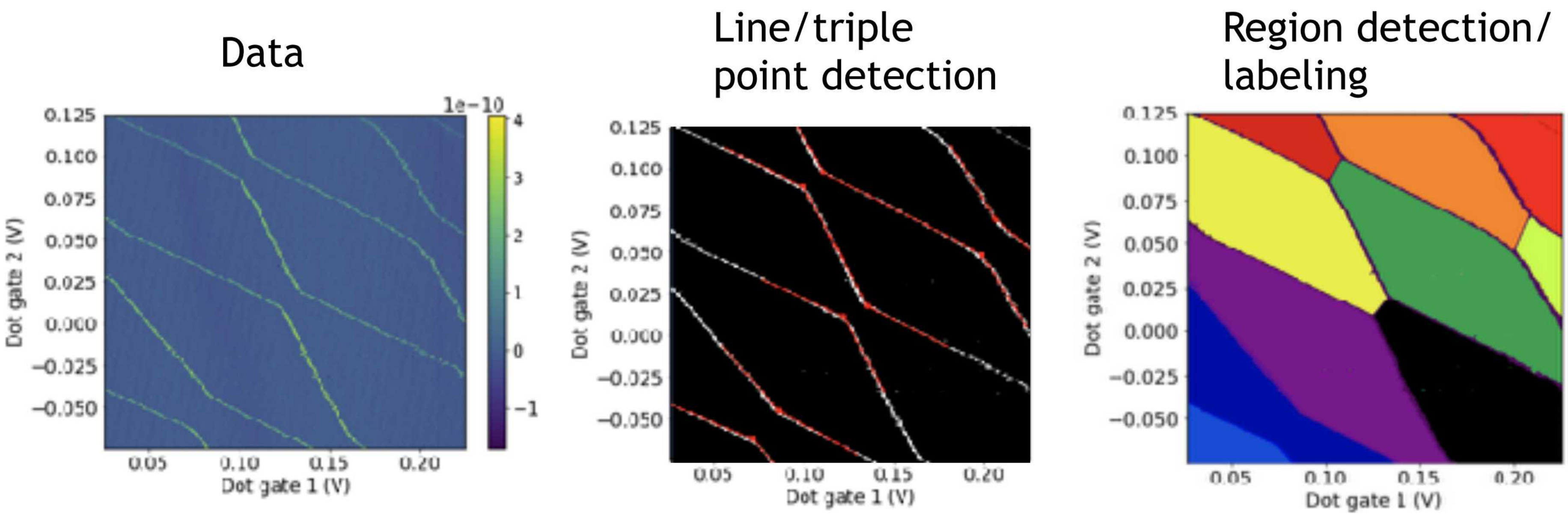


A.R. Mills *et al, arXiv1809.03976v1 (2018)*

Goals for analysis, ML, and automation

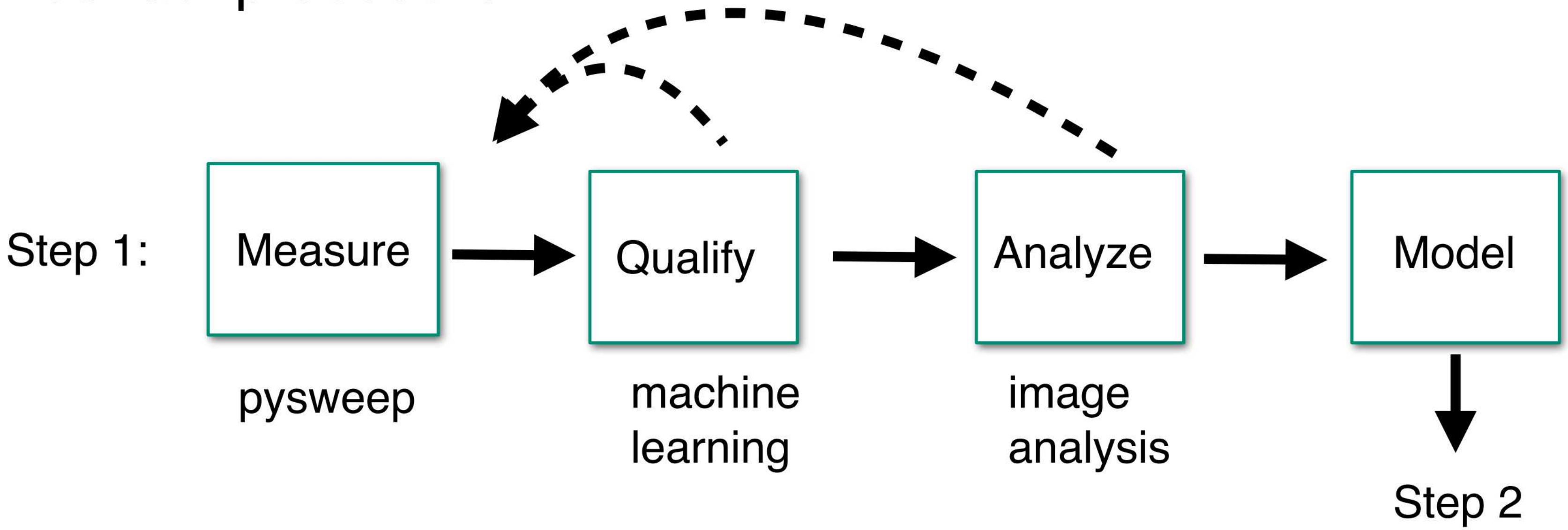
Per-scan goals:

- What features are in this image?
  - Charge offsets, anti-crossings, latching
- What are the parameters that define the features?
  - Slopes, triple points, charge occupation regions



Auto-tune n-quantum dots

Abstract procedure:

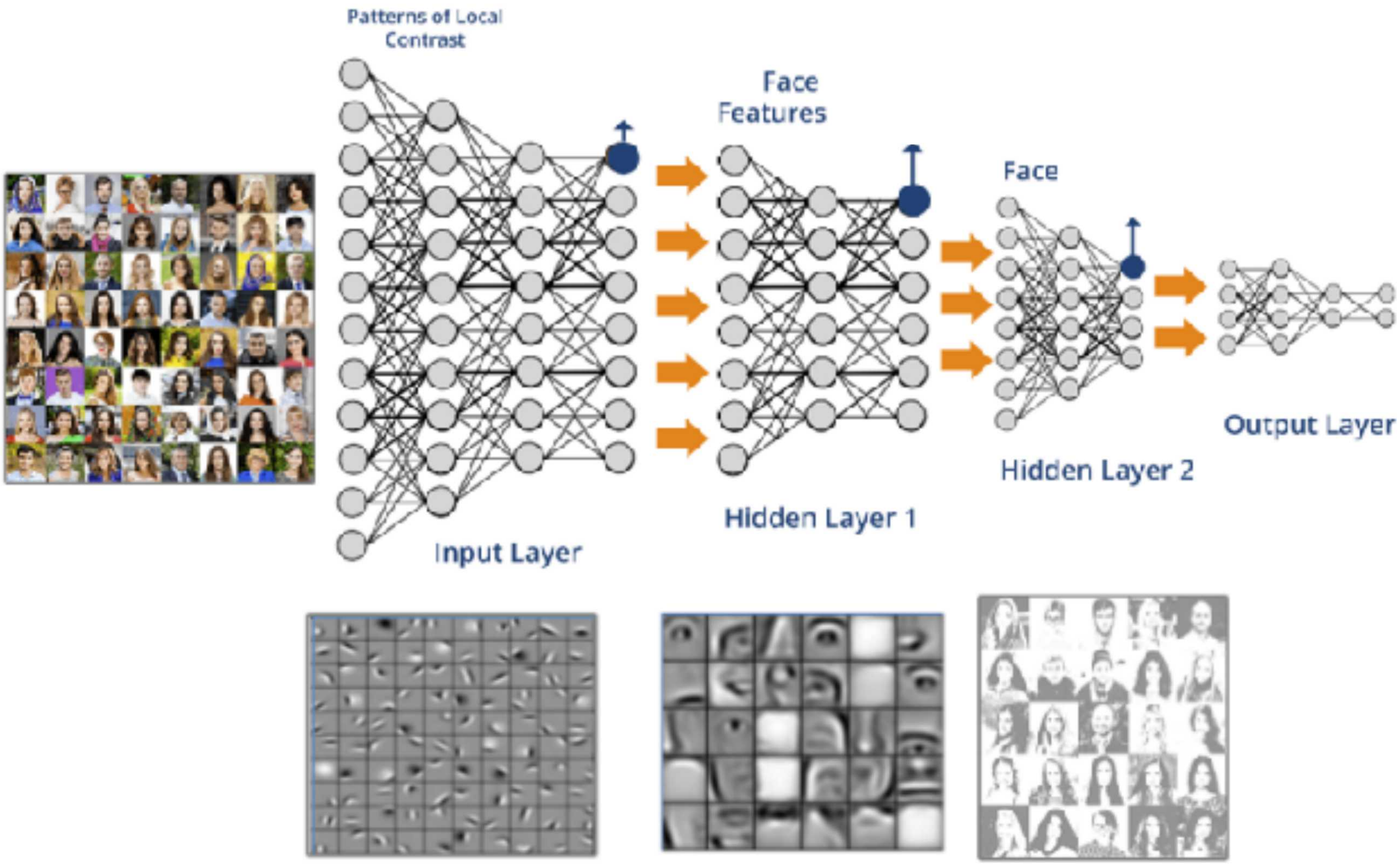


Goals:

- Start from zero knowledge of device, find establish virtual gates for device operation.
- Note: Stochastic donors make this problem somewhat harder

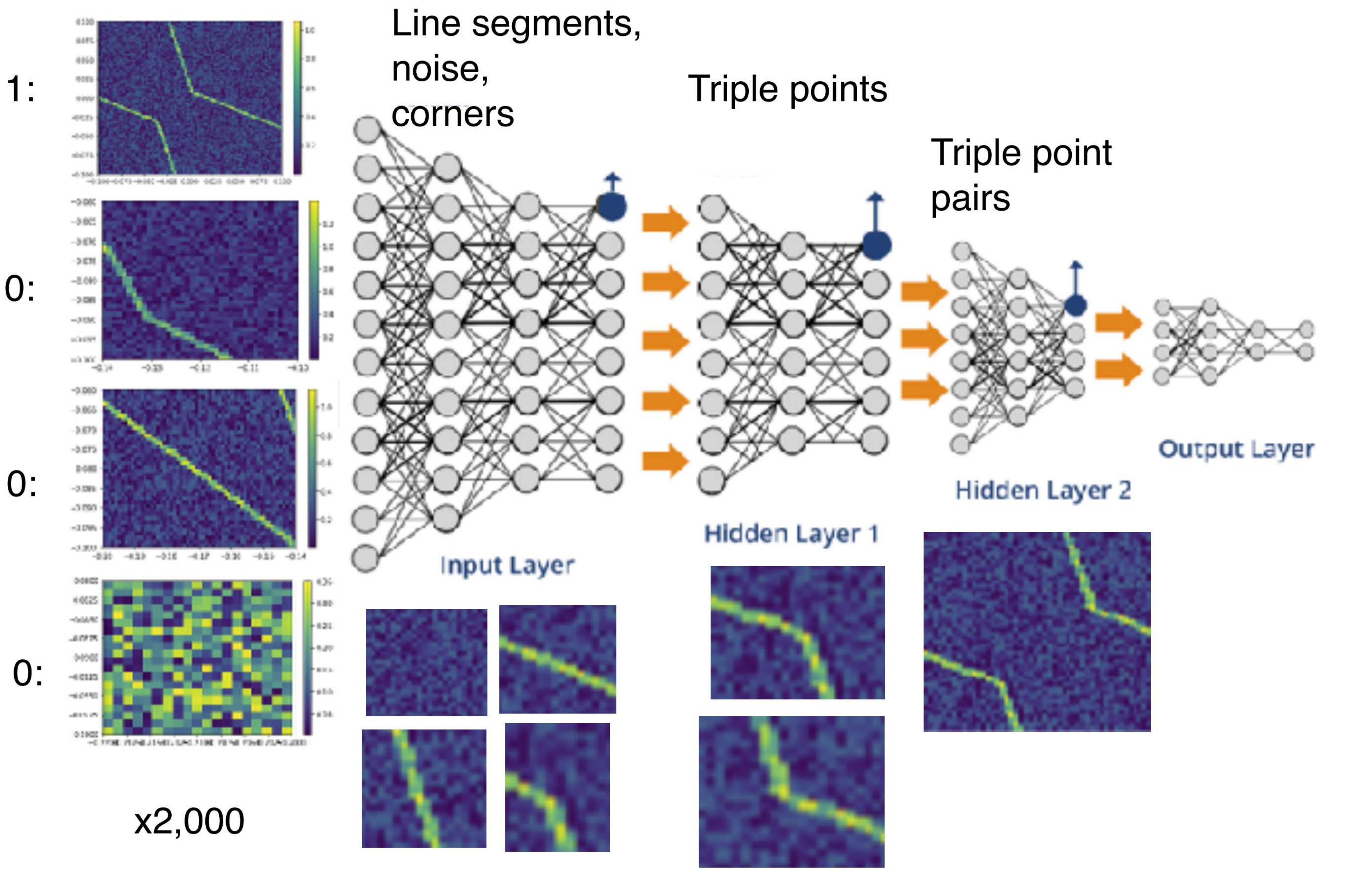
Machine learning - Convolutional Neural Net

Convolutional Neural Network - How does it work?



Machine learning - CNN

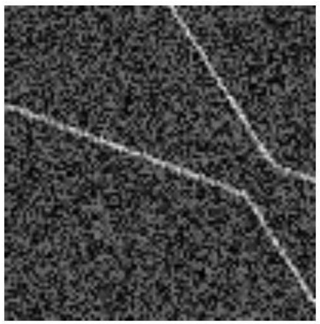
How might a CNN train to identify anticrossing: Binary classification



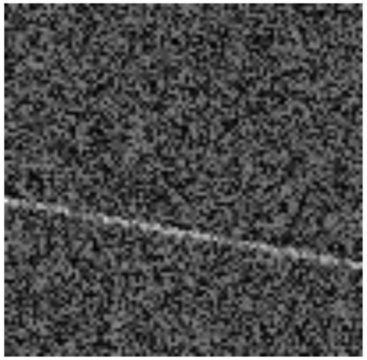
Machine learning - CNN

Trained CNN 1800 plots with offsets, anitcrossings or nothing

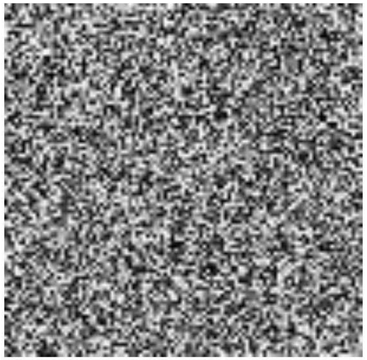
Labeled by (has offset, has anticrossing)



(True, True)



(True, False)



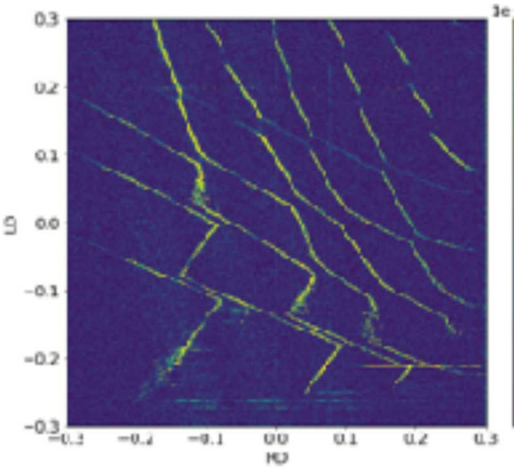
(False, False)

Each set of two layers has a pooling and a convolutional layer

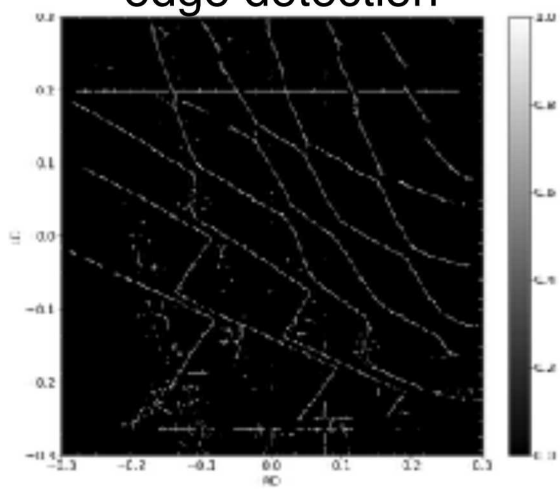
NN layers	Charge offset accuracy	Anticrossing Accuracy
4	99.1%	89.9%
6	100%	98.2%

Image analysis techniques

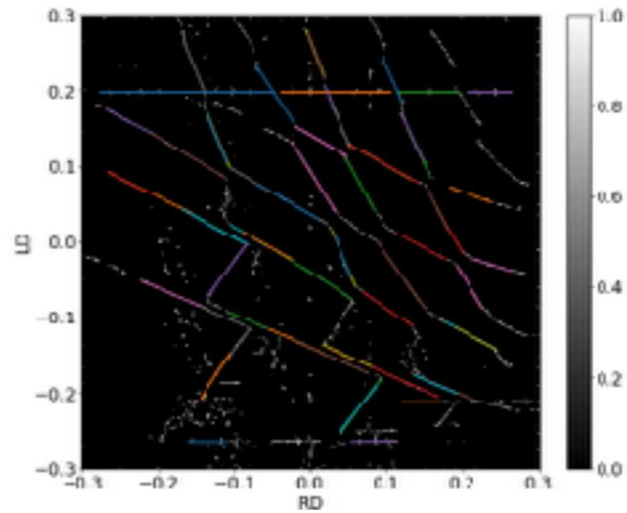
Data



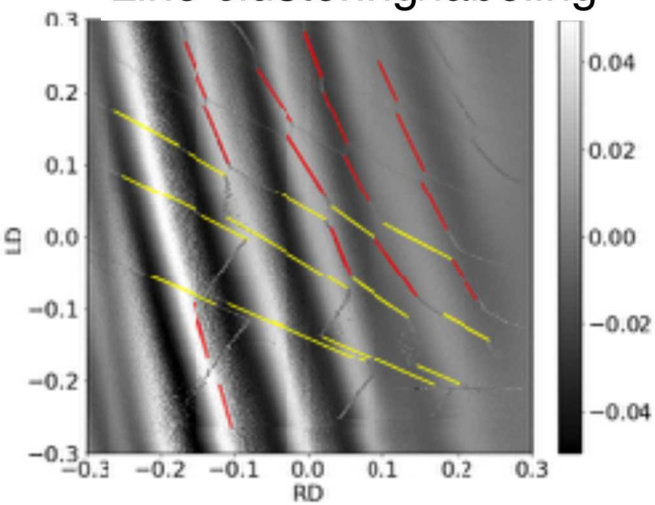
Peak finding/ edge detection



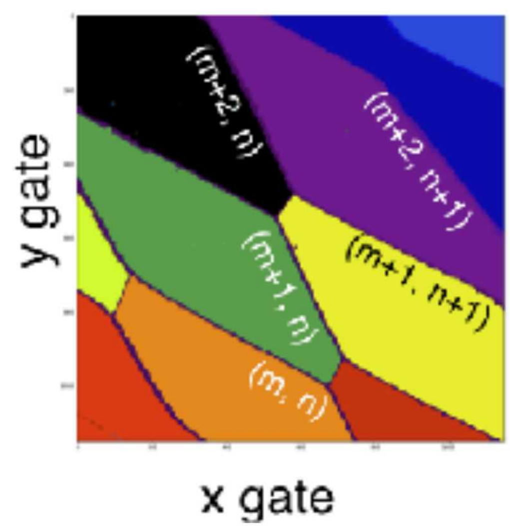
Line/shape detection



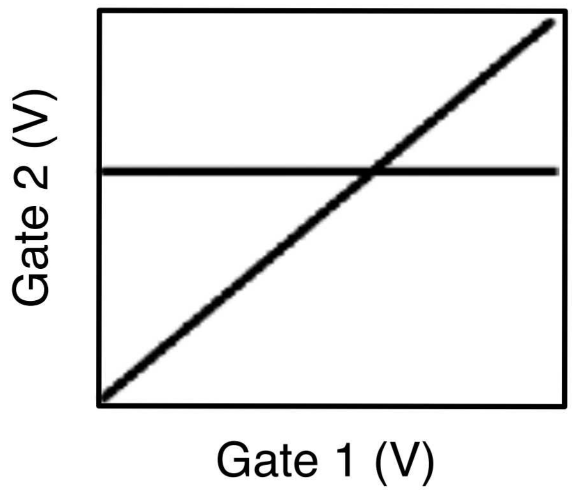
Line clustering/labeling

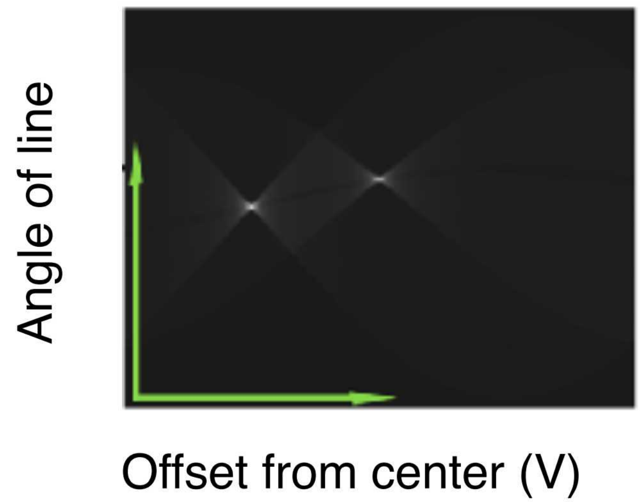


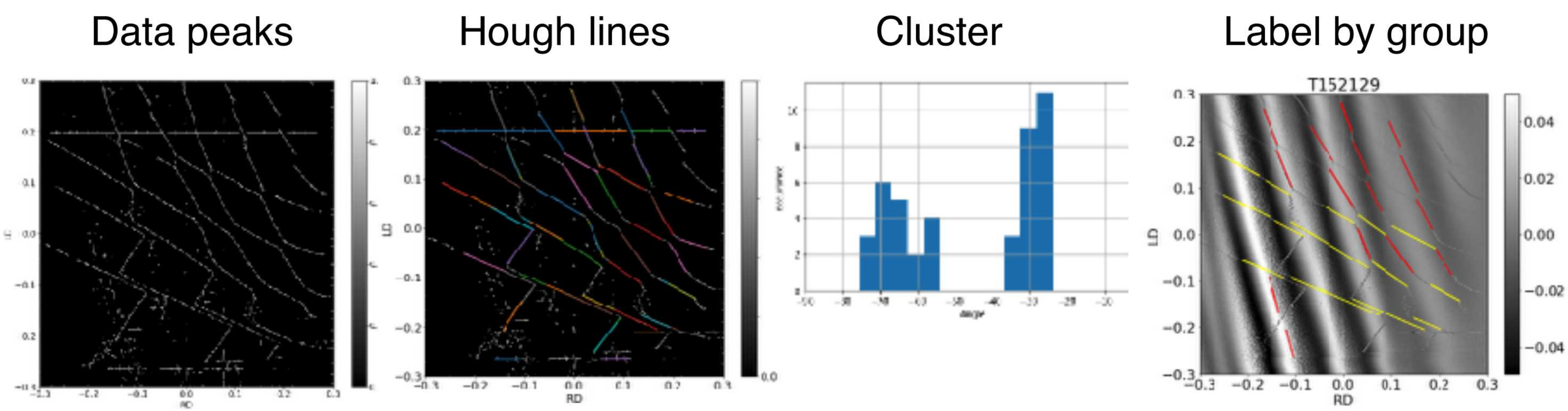
Region detection/labeling



Finding line segments: Hough transform





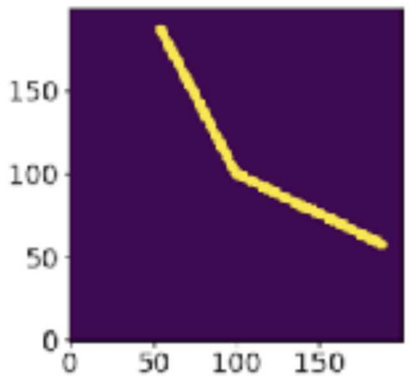


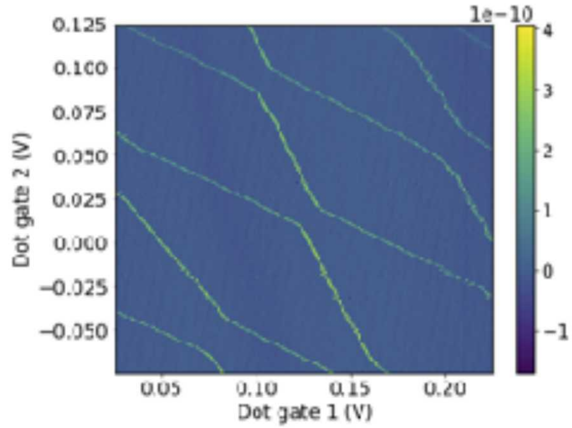
Results: Get line equation of each segment, i.e capacitance ratios

*Similar line recognition techniques developed at U. Sherbrooke Lapointe-Major, Thesis (2017)

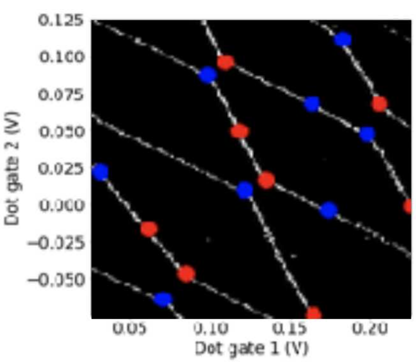
Finding AC: Generalized Hough transform

Instead of using a line, use templates.
We chose ray templates defined by two angles

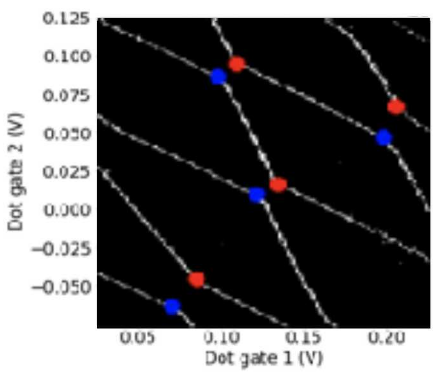




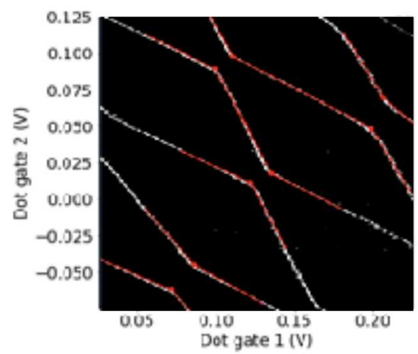
Template matches



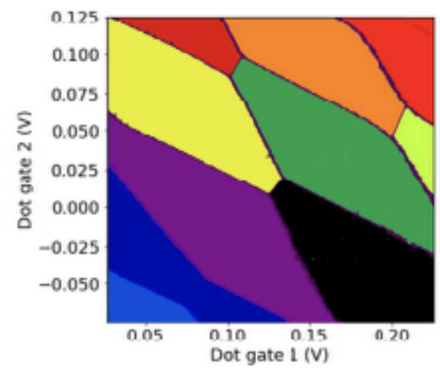
Cluster pairs



Line segments



Closed regions

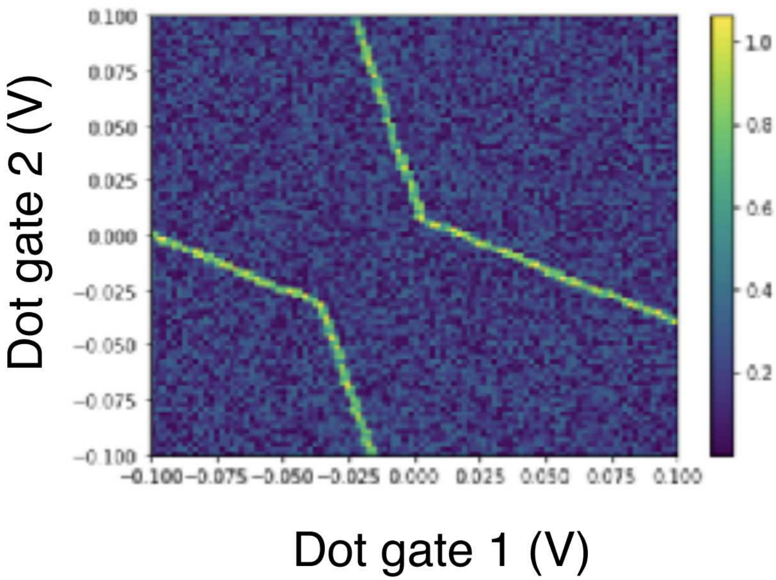


Results: C_m and capacitance ratio for each AC

Device simulator example ‘measurement’

cb object is a n-quantum dot object that has capacitance matrix information, custom or randomly generated

```
# Setup
runinfo = ps.RunInfo('fast')
runinfo.sweep  = {'dot_gate_1': ps.drange(-0.1, 0.002, 0.1)}
runinfo.step  = {'dot_gate_2': ps.drange(-0.1, 0.002, 0.1)}
# Measure
expt = get_stability_diagram(runinfo, cb)
# Plot
qplt.charge_sense_plot(expt)
```



Simplifications

- Tunnel rates are just right
- Fixed tunnel coupling
- No stochastic donors
- No charge sensor background

Auto tune procedure for n-quantum dots

0. Fix voltages for 0 electron occupation in each dot

1.1 Increase dot gate 1 until charge transition is found (CNN)

1.2 Get capacitance matrix for dot 1 vs all other gates (IA)

1.3 Find next charge transition for dot 1 (CNN)

1.4 Get charging voltage for dot 1 (IA)

1.5 Virtualize all gates relative to dot 1

1.6 Fix chemical potential of dot 1, so dot 2 can be filled

IA - image analysis
CNN - Convolutional Neural Network

2.1 Increase dot gate 2 until charge transition is found (CNN)

2.2 Cap matrix (IA)

2.3 Next transition (CNN) 2.4 Charging V (IA)

2.4 Virtualize relative to dot 1 and dot 2

2.5 Go to (1, 0) <-> (0, 1) crossing and get C_m

2.6 Fix chemical potential of dot 1 and 2 so that dot 3 can be filled

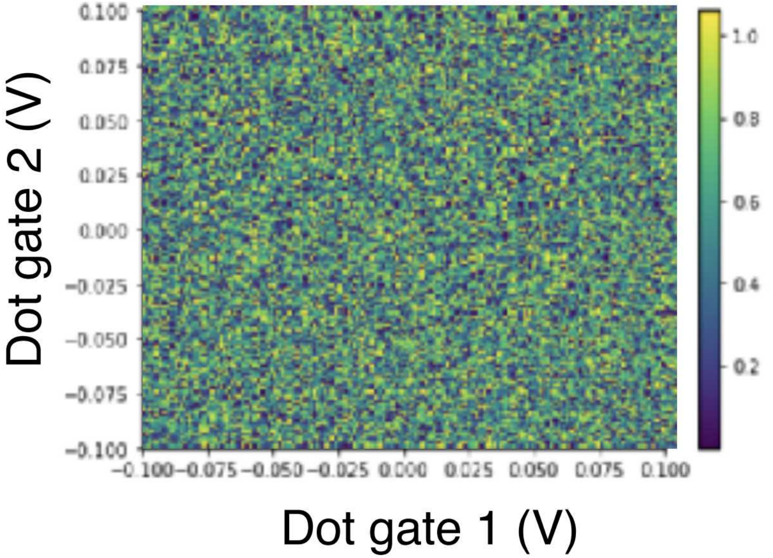
3. to n. Repeat all steps in 2 for all other quantum dots

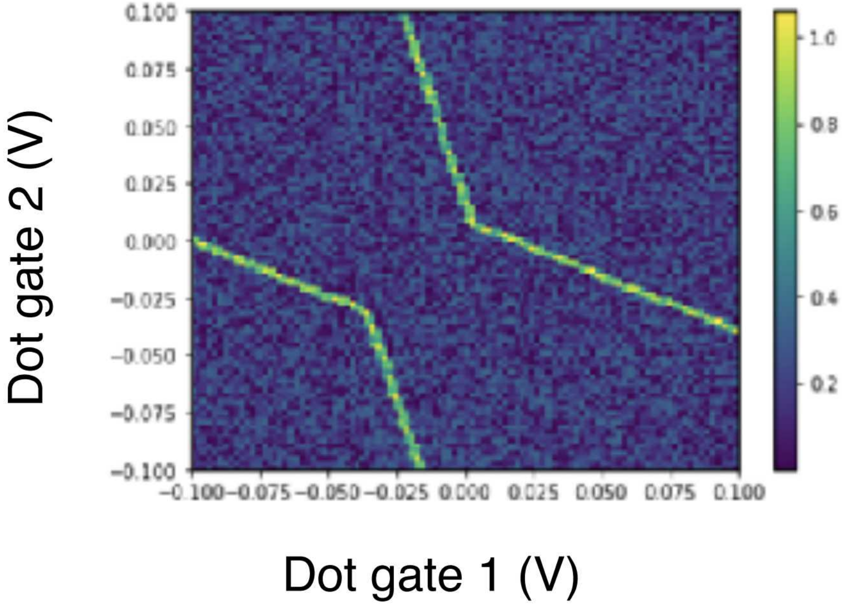
Auto tune procedure for n-quantum dots

- Successfully implemented on simulated device with two quantum dots.

Automation, CNN, & IA

0 - electrons in QDs → Desired Configuration





- Generalizing for step 3 and beyond
- Can likely restrict virtualization to NN and NNN

Conclusions

- Image analysis, using hough transforms give quantitative information about stability diagrams
- Machine learning, (convolutional neural network) gives qualitative information
- Combination of techniques builds into automation routine for auto-tuning n-quantum dots.

Future steps:

- Add dynamic component binary classifiers, i.e. latching and tunnel broadening
- Sprinkle random donors
- Add model learning routines for self correcting device model, corrections of drift or miss-predictions
- Use model information for pulse design