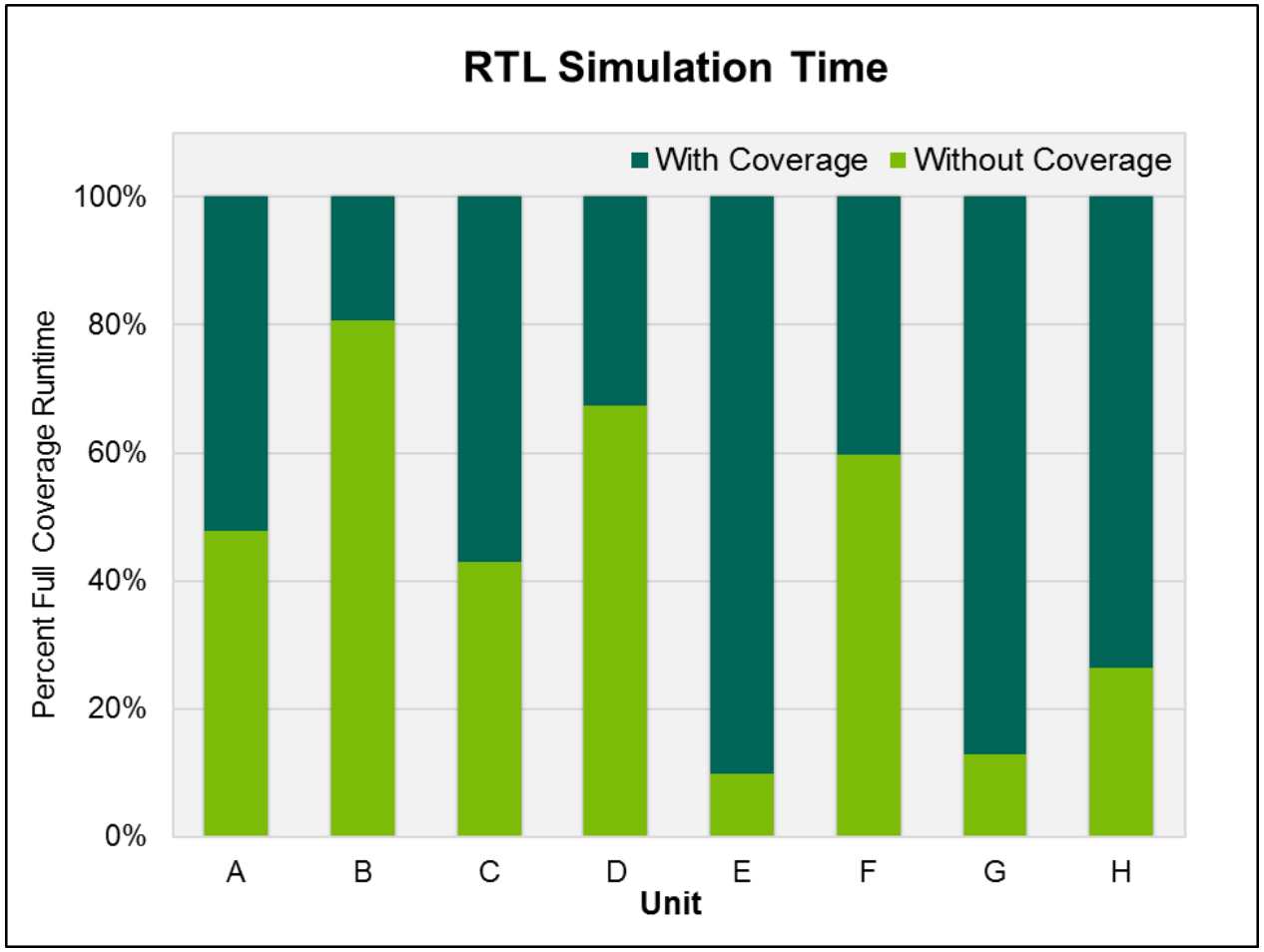
2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

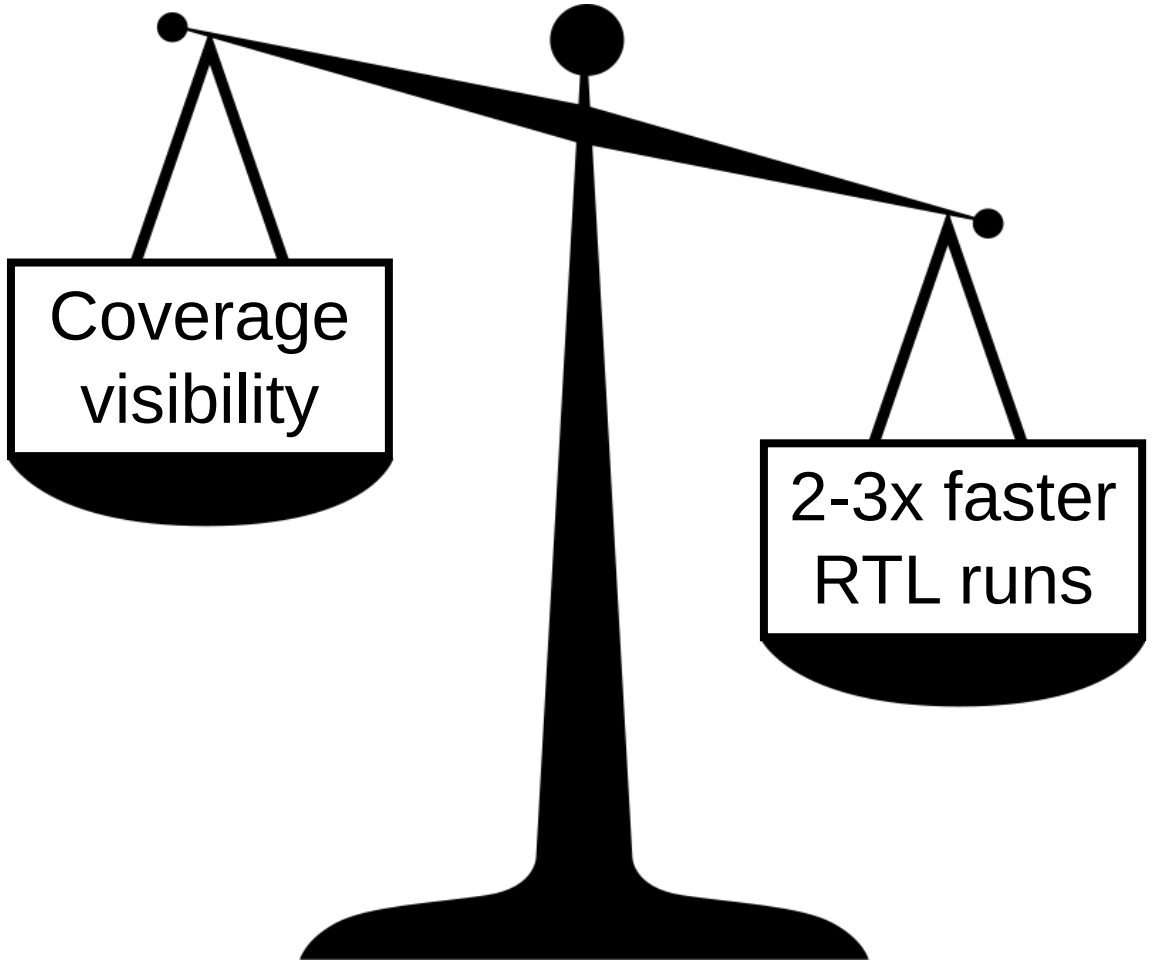
Deep Predictive Coverage Collection

Rajarshi Roy - NVIDIA Corp.
Chinmay Duvedi - NVIDIA Corp.
Saad Godil - NVIDIA Corp.
Mark Williams - NVIDIA Corp.

NVIDIA VOLTA: 21B Transistors

Coverage isn't free

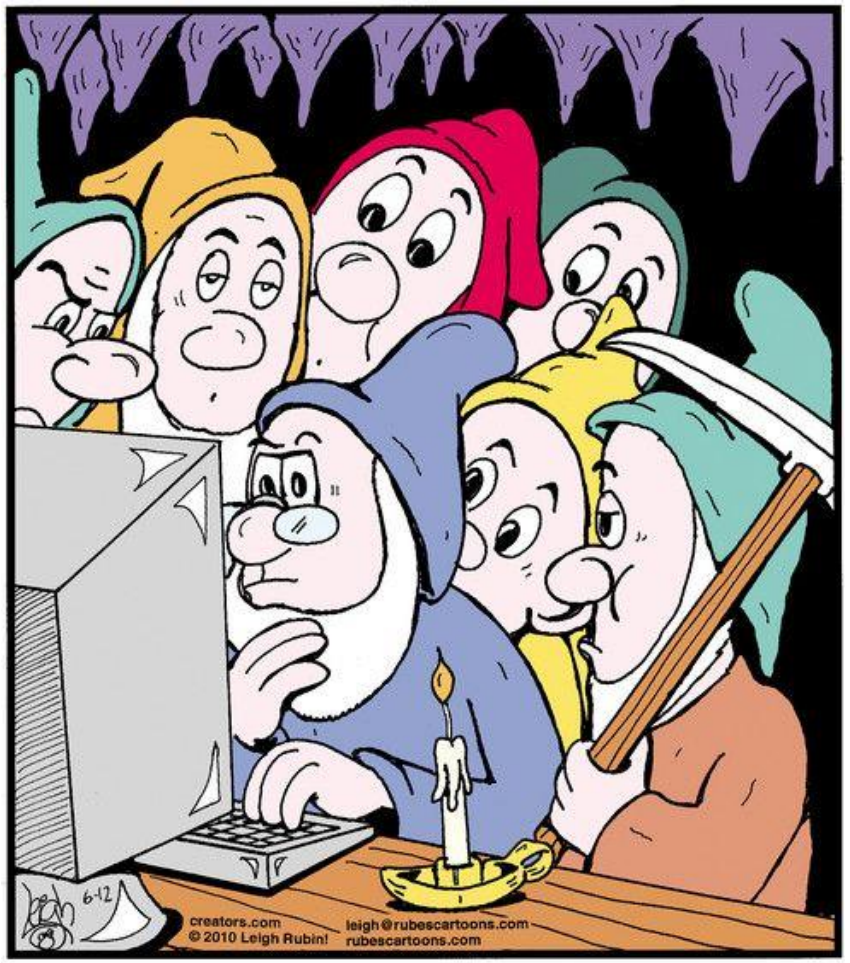




Looking closer at the data

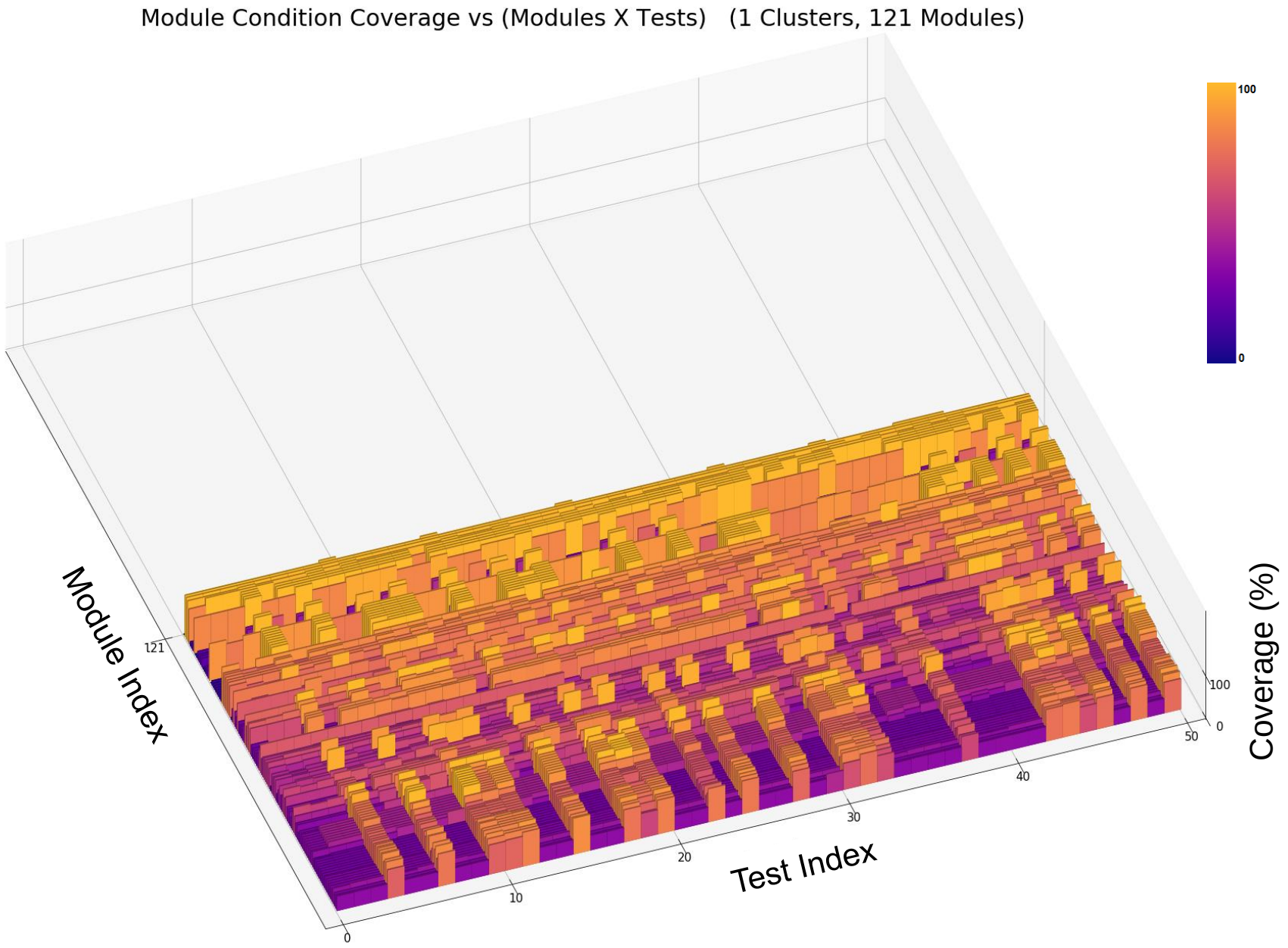
- Weekly:
  - ~100,000 tests across 8 units
  - TBs of raw coverage data collected
- Usual flow:
  - Compile overall reports
  - Use compiled reports for coverage feedback
  - Throw away raw coverage data

Missed Opportunity?

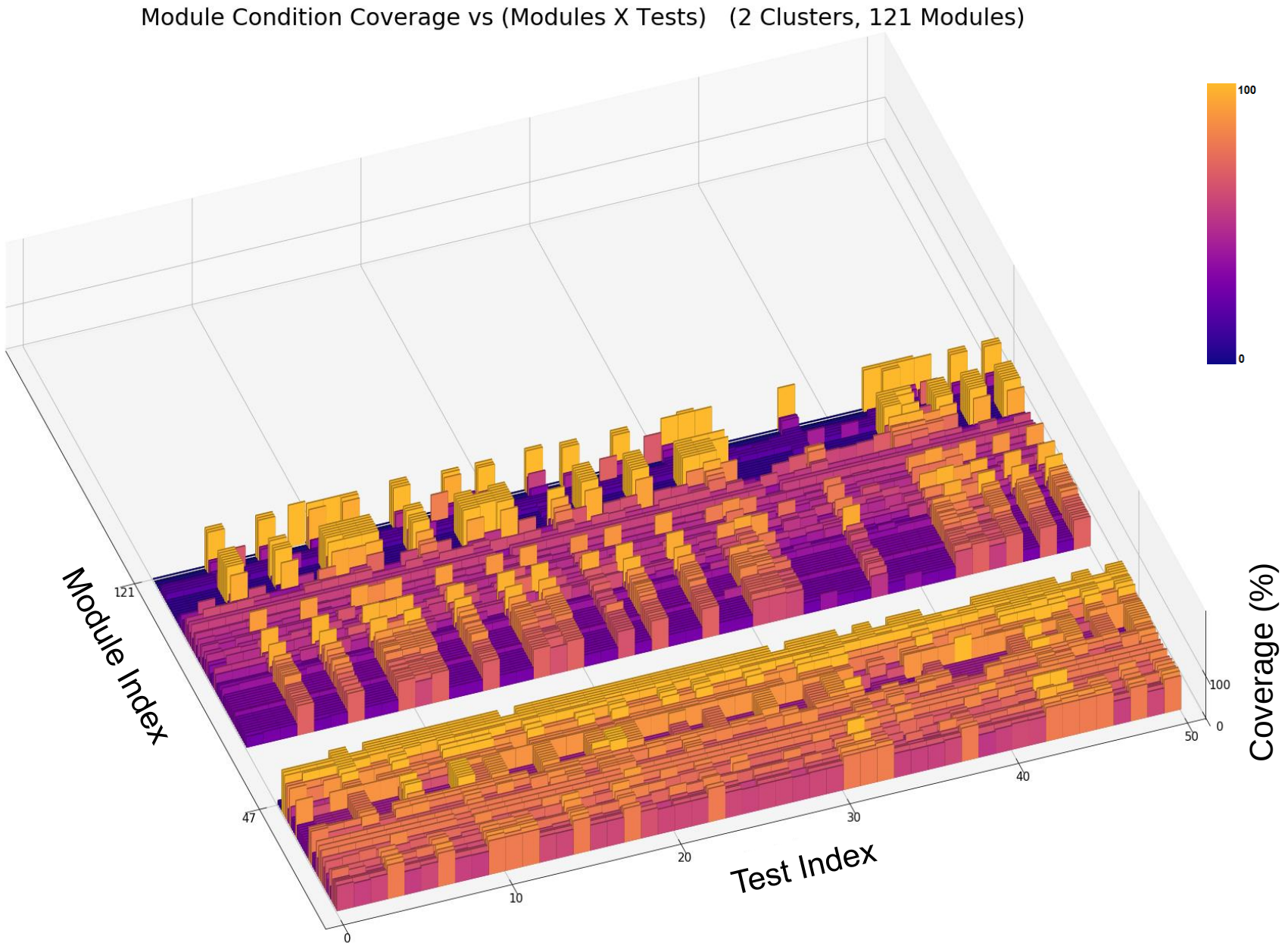


Data mining

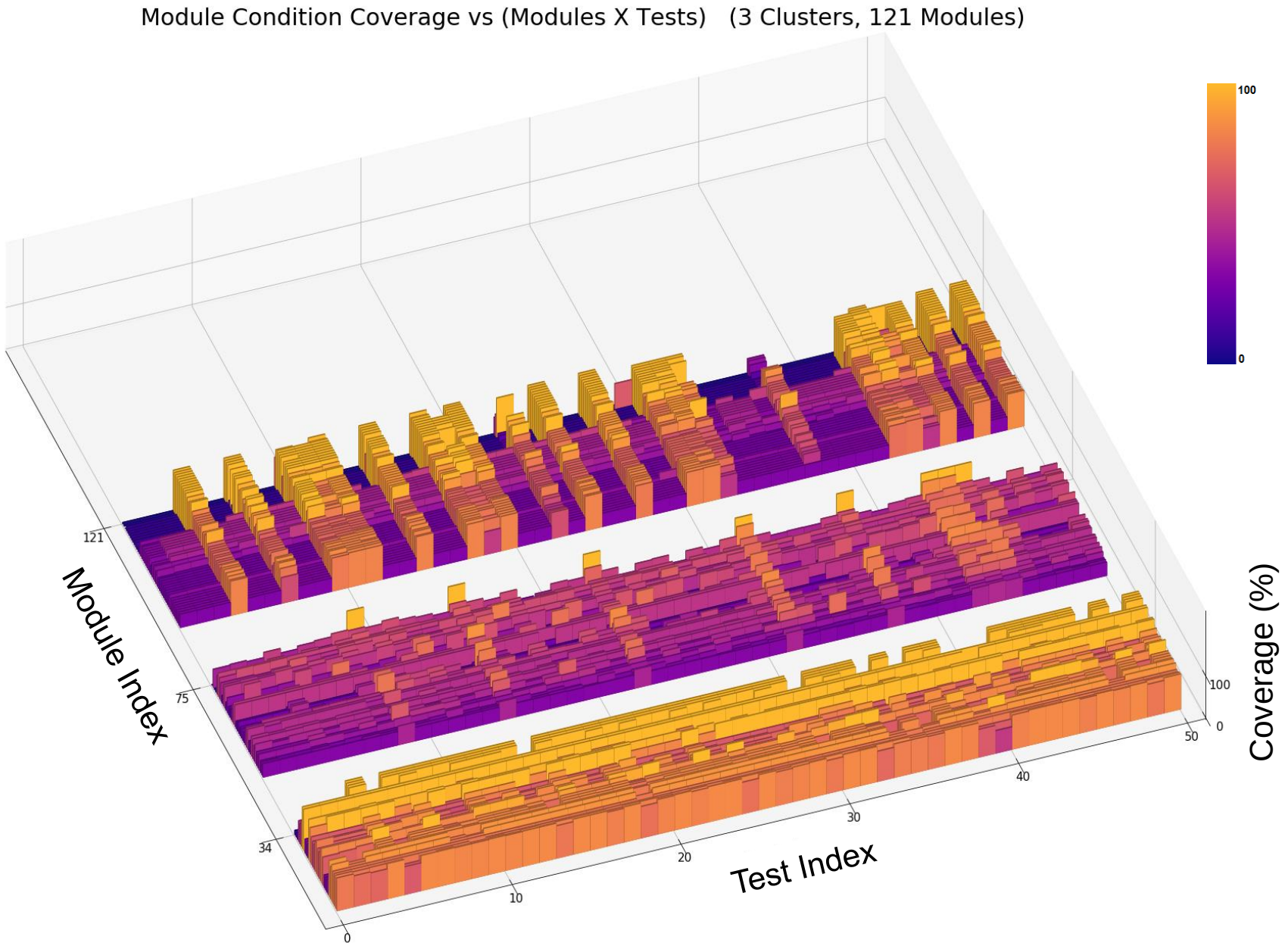
- Example:
  - Small unit in NVIDIA GPU
  - 121 modules
  - Module condition coverage
- Height/heat map for 50 tests shown



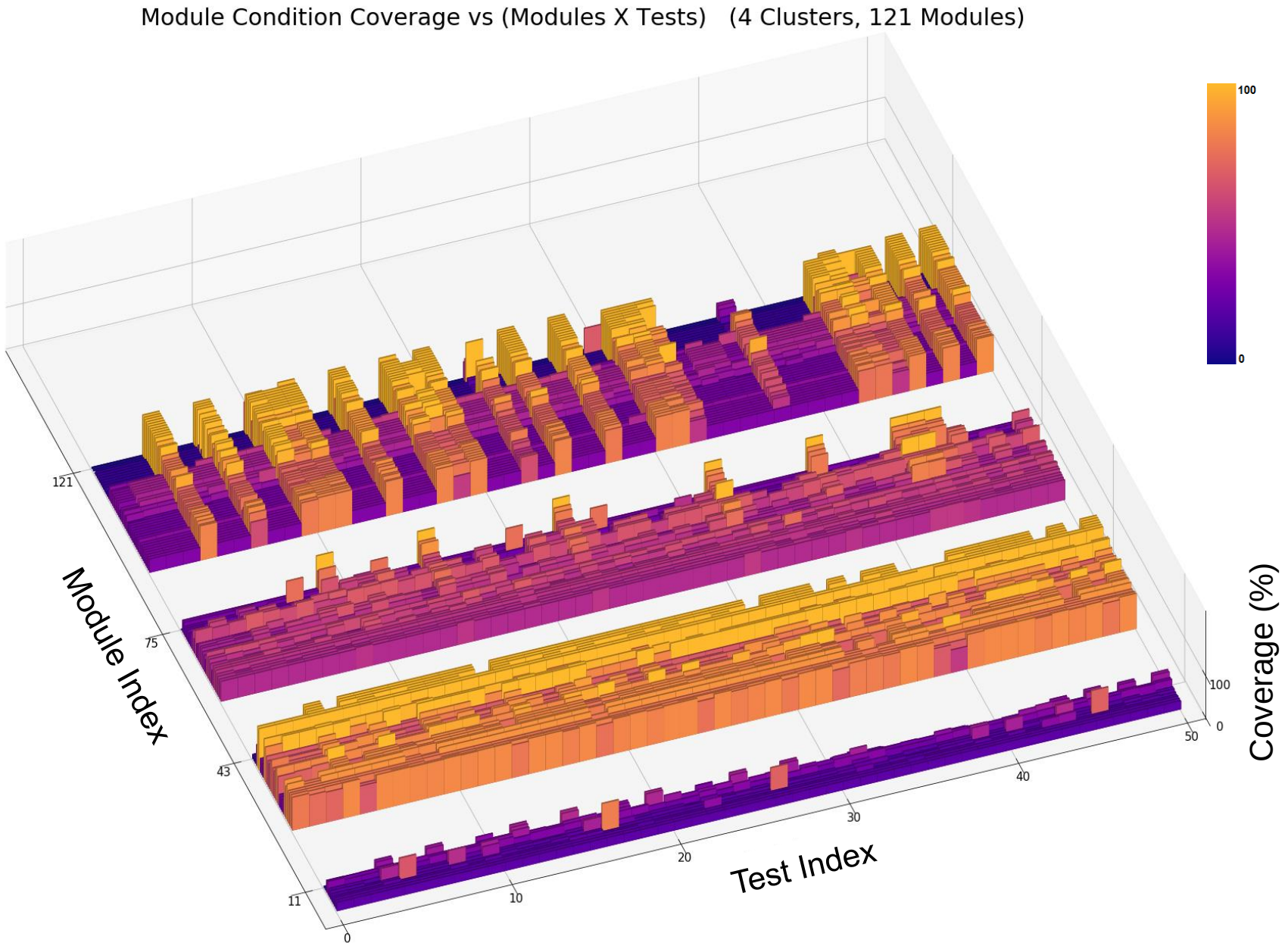
- 2 clusters


Module Condition Coverage vs (Modules X Tests) (2 Clusters, 121 Modules)
100
0
Module Index
121
47
0
Test Index
10
20
30
40
50
100
0
Coverage (%)

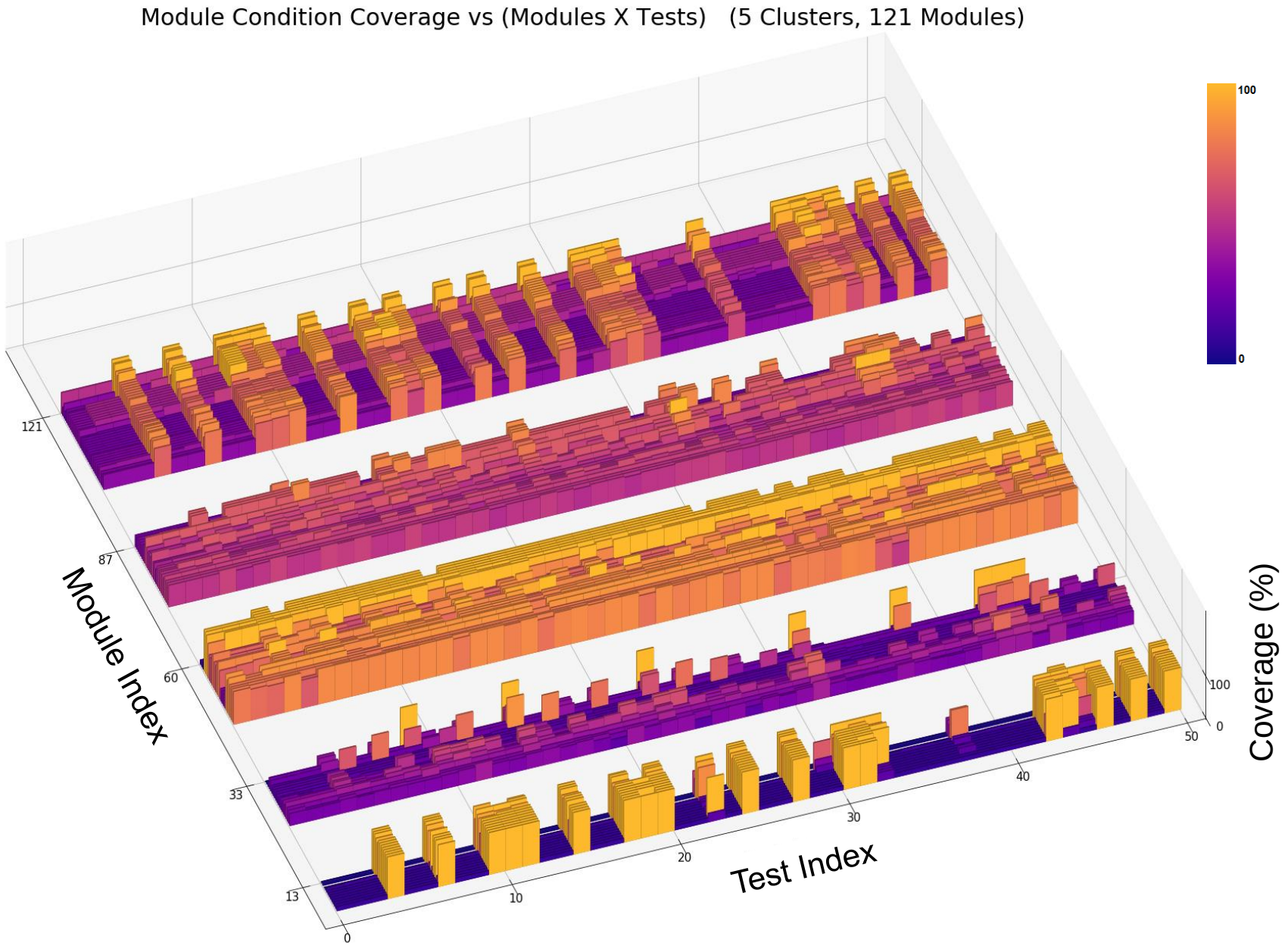
- 3 clusters


Module Condition Coverage vs (Modules X Tests) (3 Clusters, 121 Modules)
Module Index
121
75
34
0
Test Index
0
10
20
30
40
50
Coverage (%)
100
0

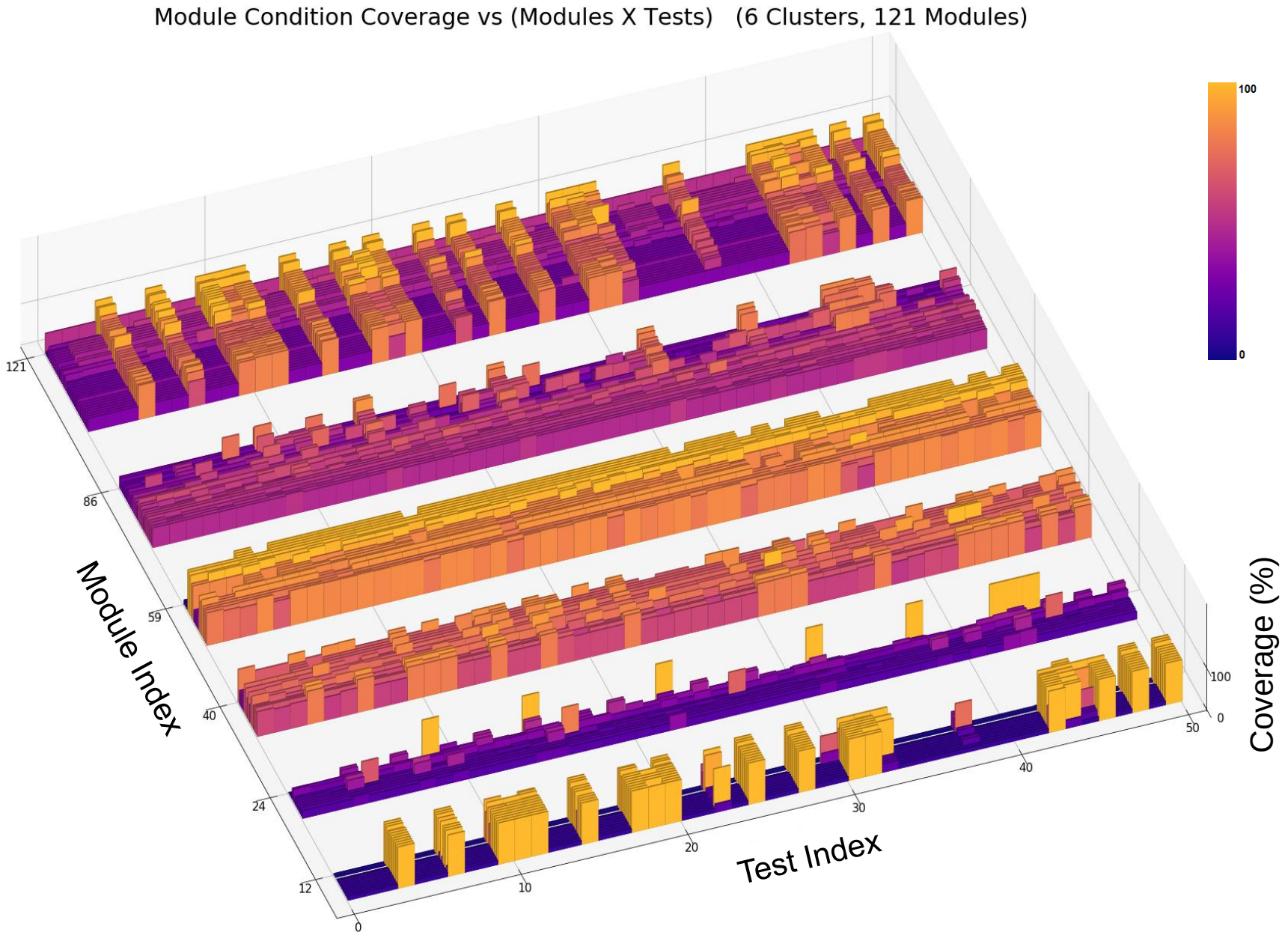
- 4 clusters


Module Condition Coverage vs (Modules X Tests) (4 Clusters, 121 Modules)
Module Index
121
75
43
11
0
Test Index
0
10
20
30
40
50
Coverage (%)
100
0

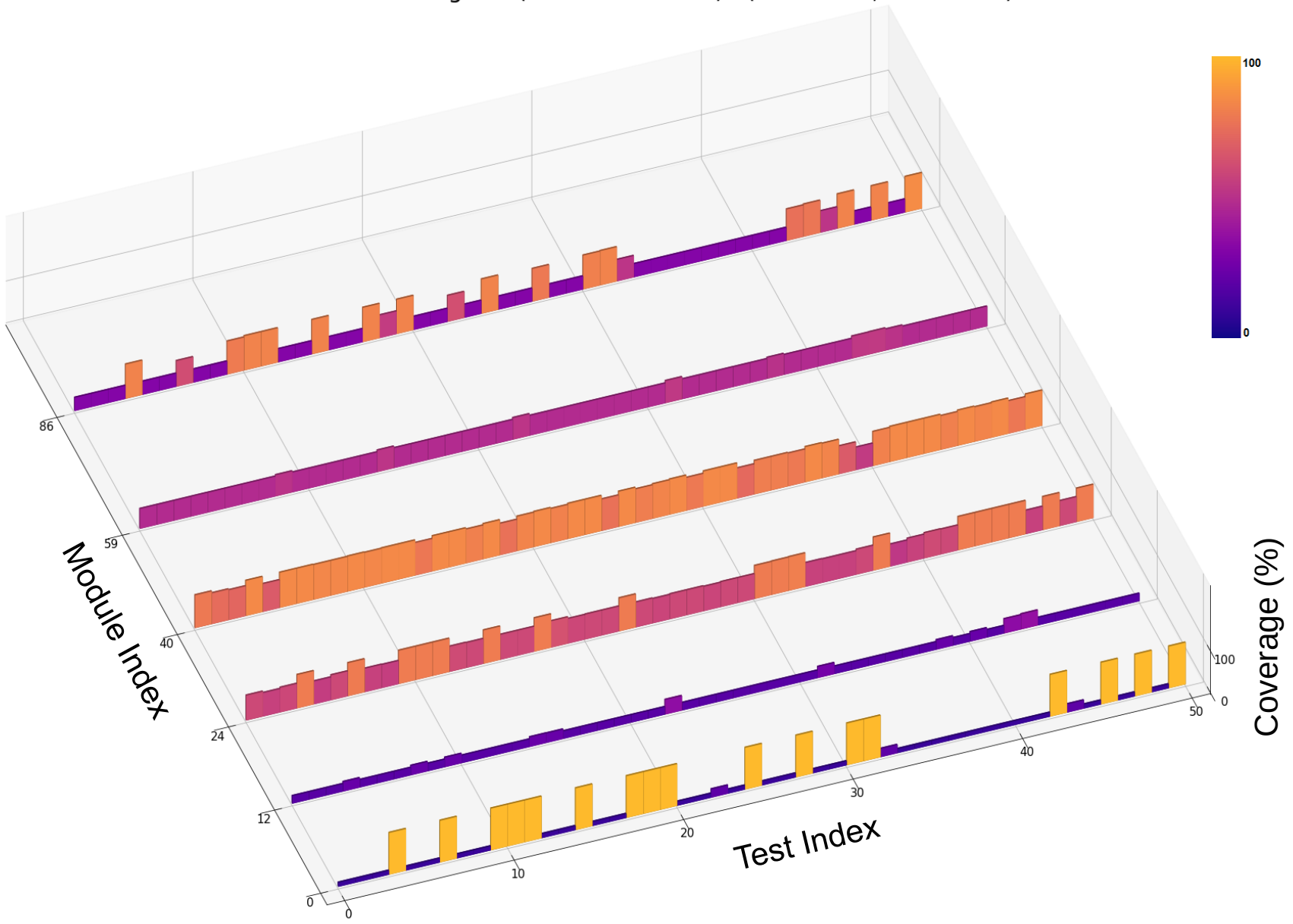
- 5 clusters


Module Condition Coverage vs (Modules X Tests) (5 Clusters, 121 Modules)
Module Index
121
87
60
33
13
0
Test Index
0
10
20
30
40
50
Coverage (%)
100
0

- 6 clusters
- Modules in a cluster have very similar coverage behavior!
- Collect coverage for one module per cluster…

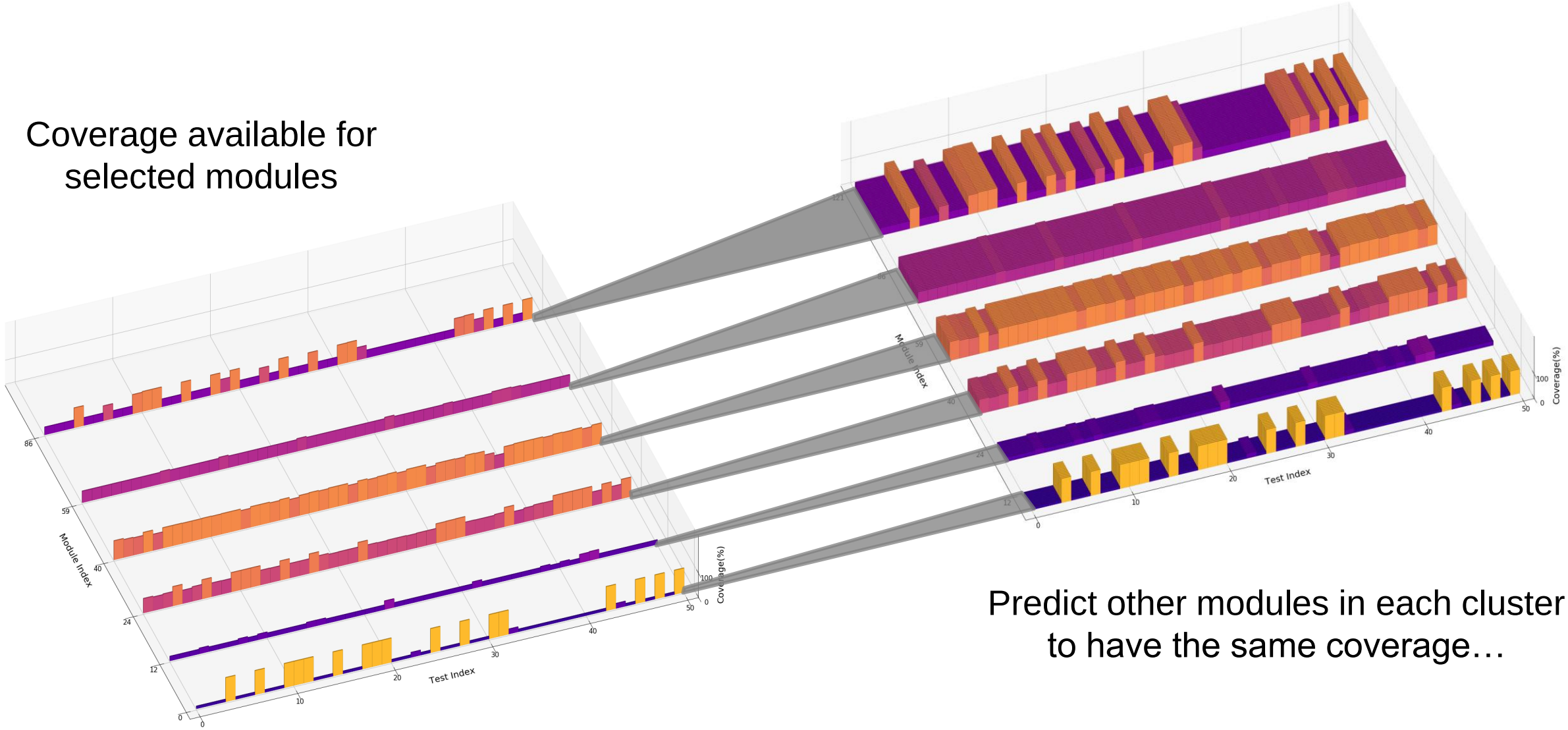


- 6 modules selected for coverage collection
- Out of 121 modules
- ~5% of design



Given a new test…

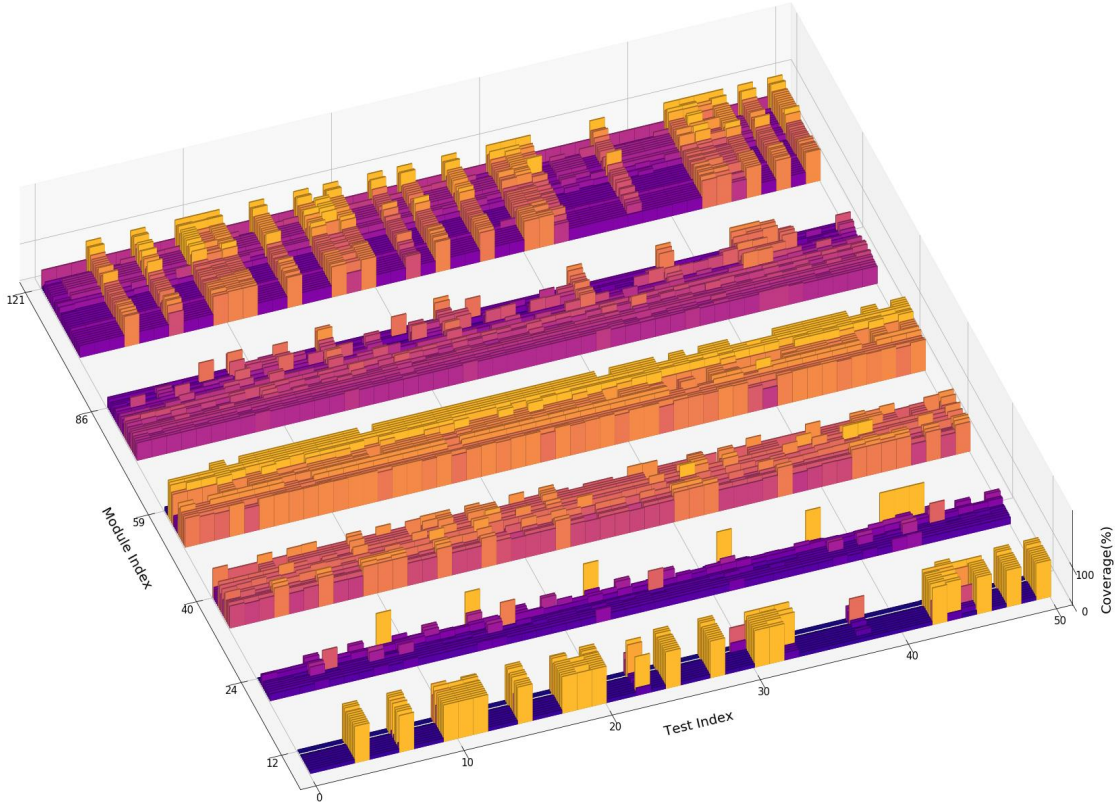
Coverage available for selected modules



Predict other modules in each cluster to have the same coverage…

Given a new test…

- A decent guess…
- We can do a lot better!

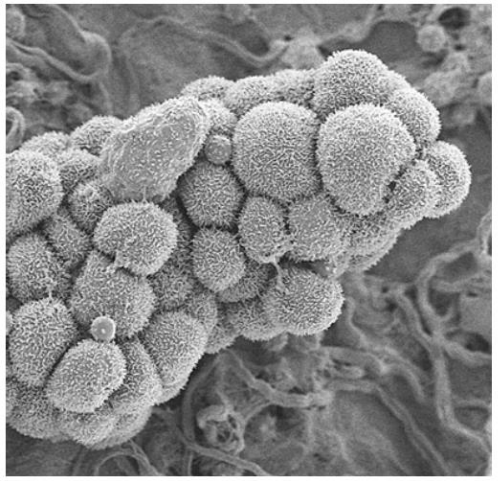


2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Deep Learning

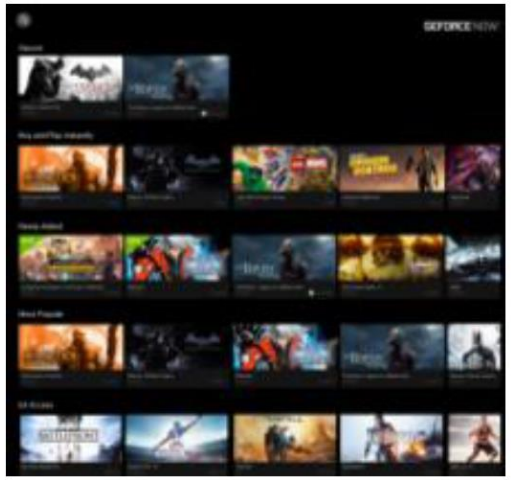
INTERNET & CLOUD

Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation



MEDICINE & BIOLOGY

Cancer Cell Detection
Diabetic Grading
Drug Discovery

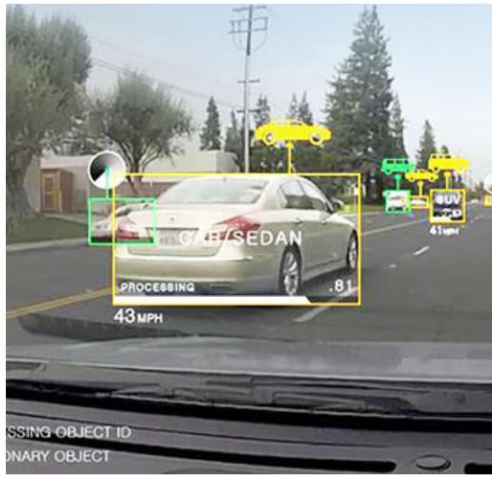


MEDIA & ENTERTAINMENT

Video Captioning
Video Search
Real Time Translation

SECURITY & DEFENSE

Face Detection
Video Surveillance
Satellite Imagery

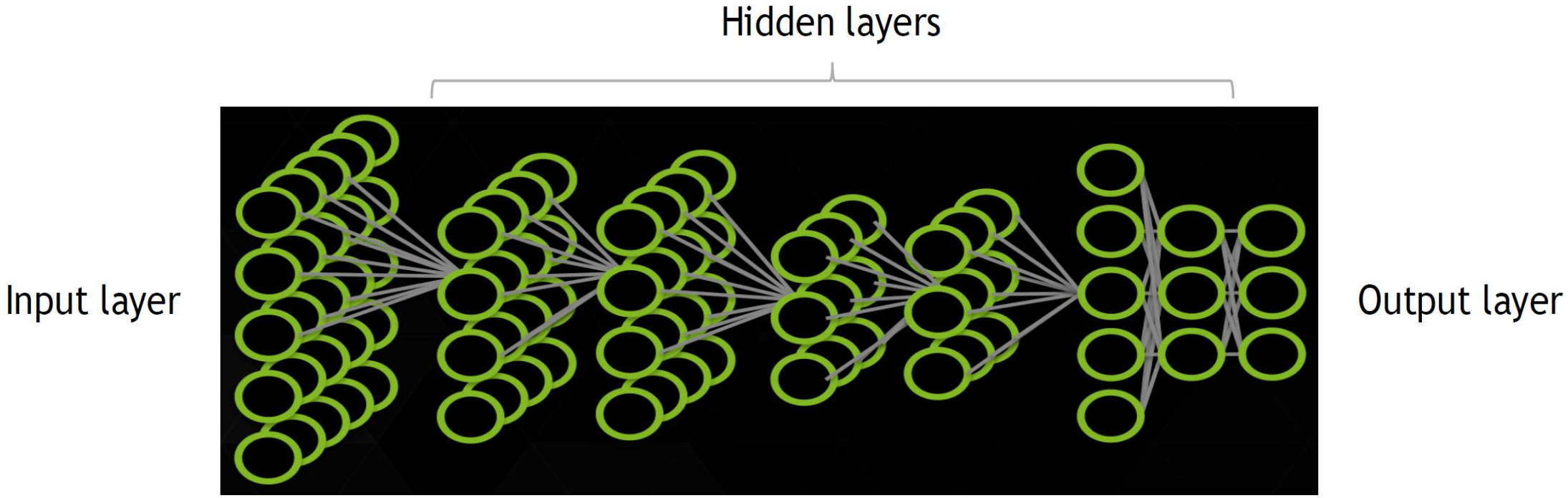


AUTONOMOUS MACHINES

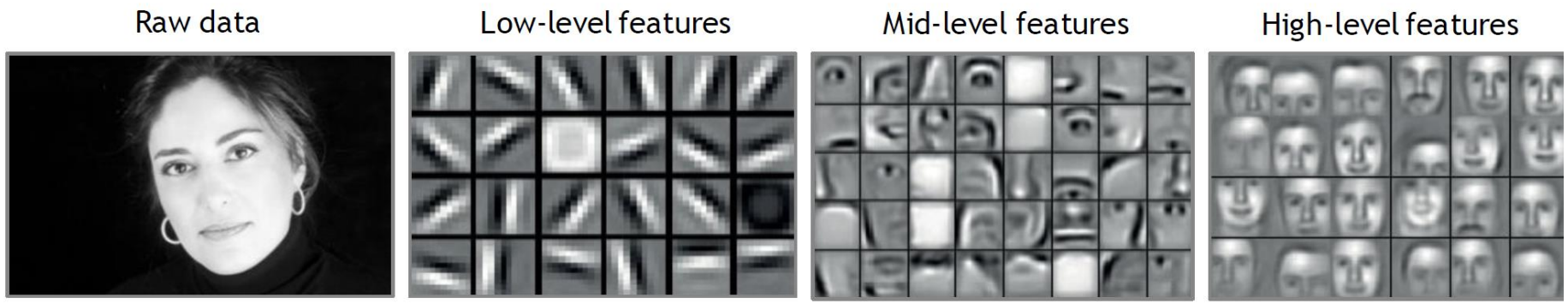
Pedestrian Detection
Lane Tracking
Recognize Traffic Sign

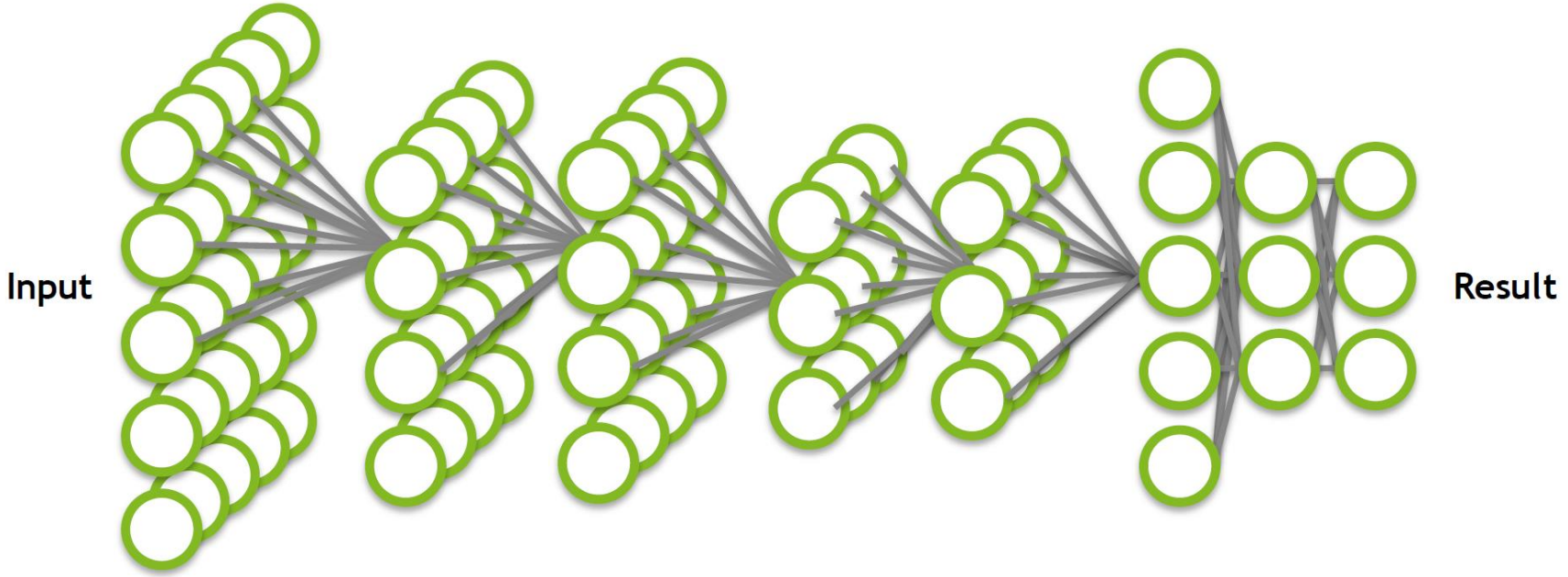
Deep Neural Networks (DNNs)

- Collection of simple, trainable mathematical units
- Collectively learn complex functions



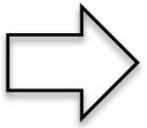
DNN Layers

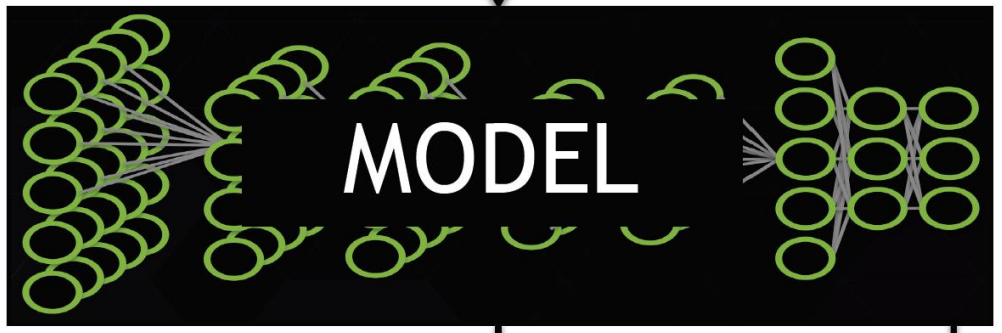

Raw data
Low-level features
Mid-level features
High-level features


Input
Result

Training DNNs

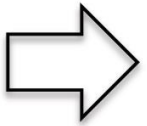
Train:

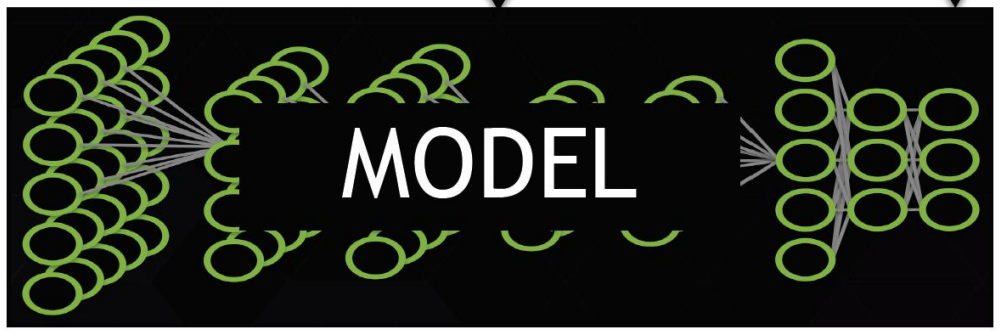




Errors

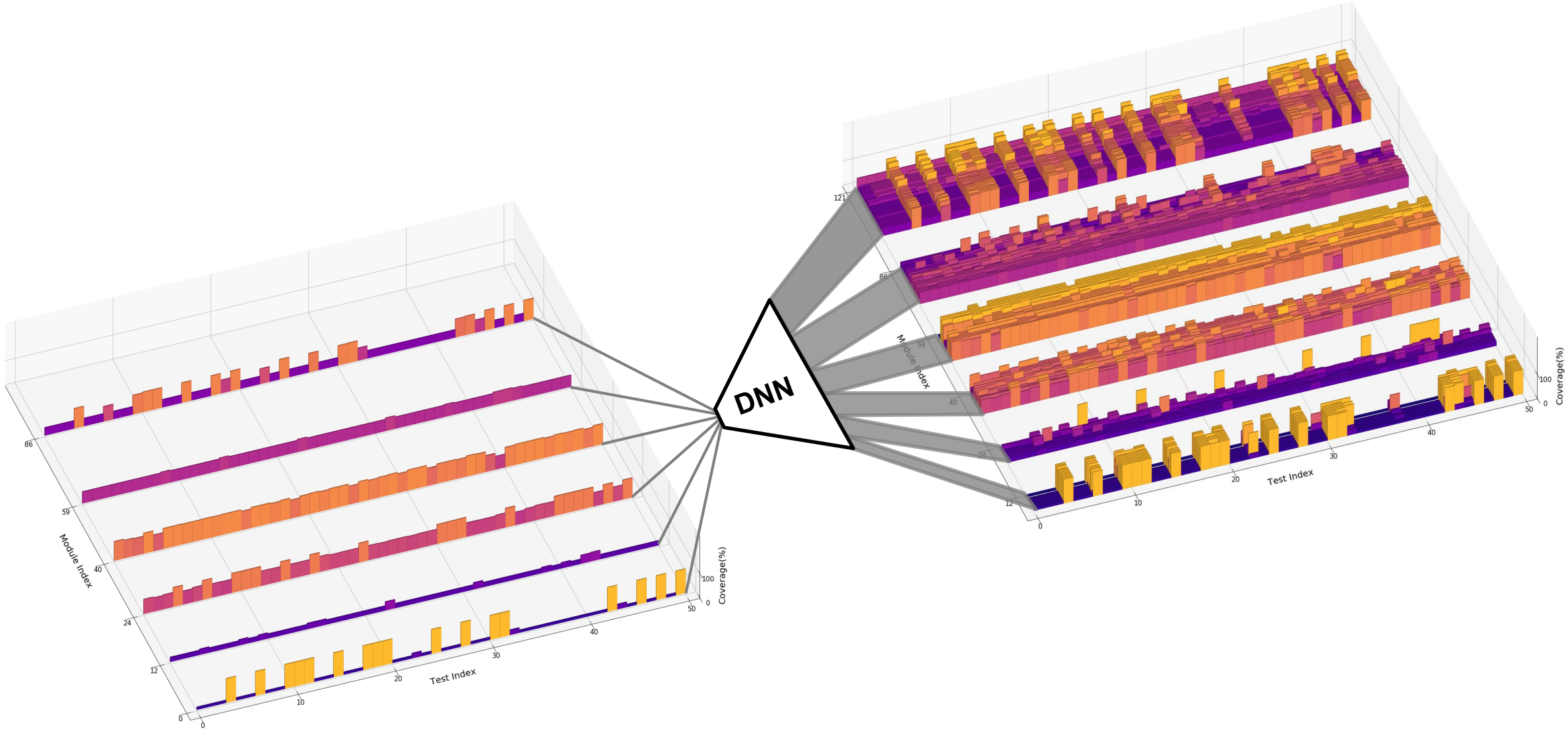
Deploy:



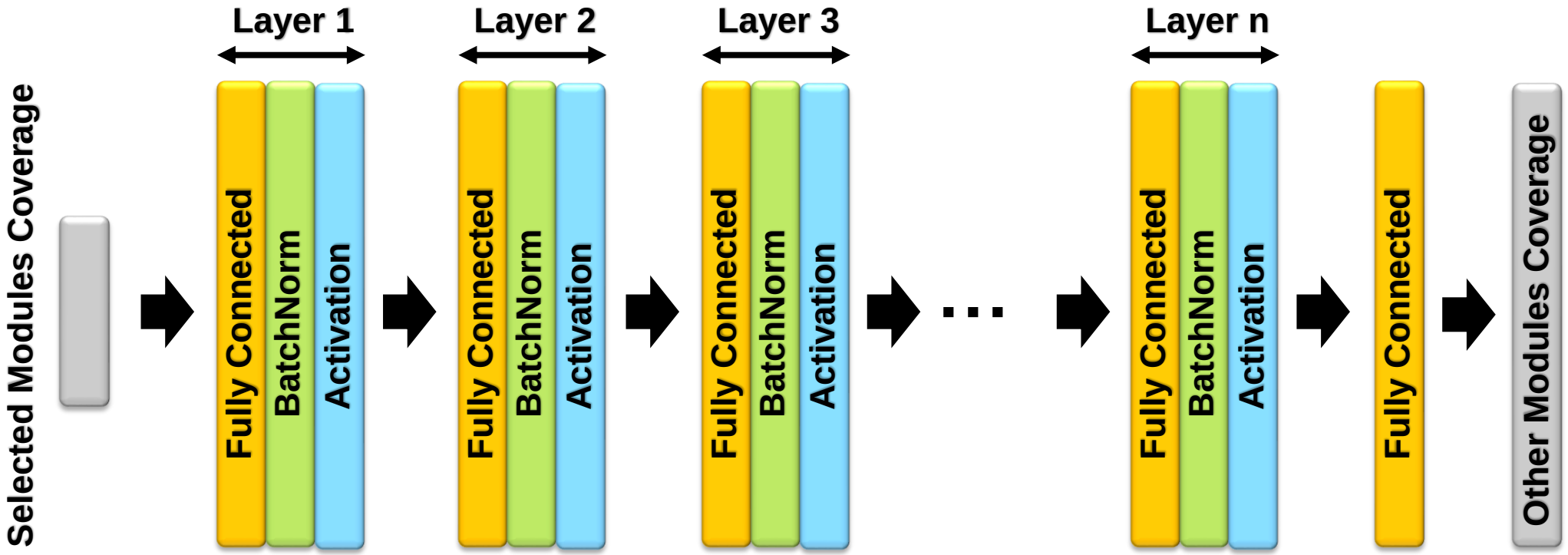


2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Predict full coverage with DNN



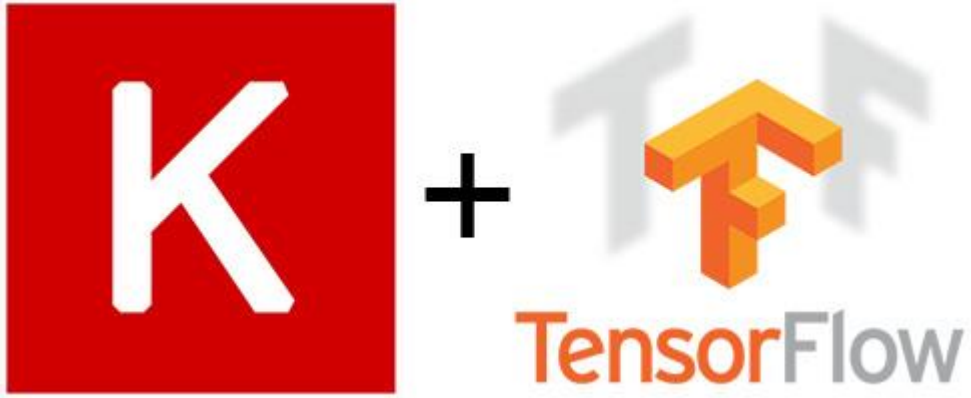
Deep Neural Net Architecture



Deep Neural Net Architecture

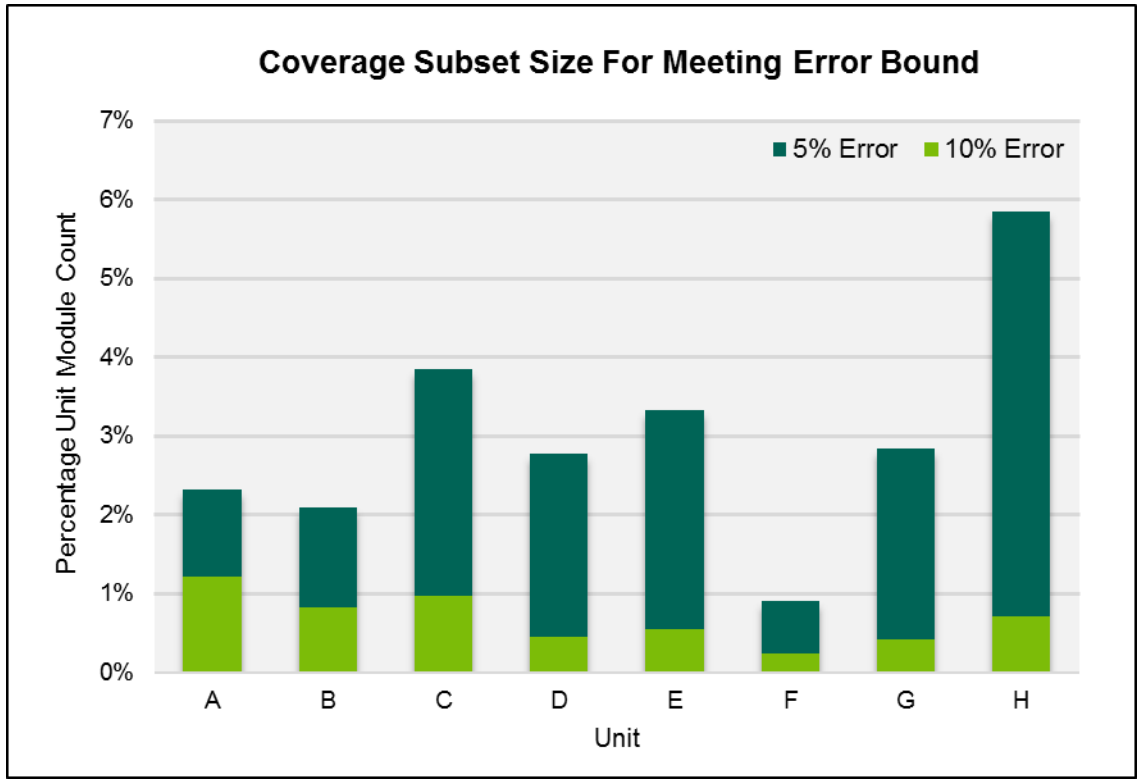
```
model = Sequential()
model.add(Dense(y_train.shape[1], input_shape=(x_train.shape[1],), W_regularizer=regularizers.l2(l2_reg)))

for l in range(layers-1):
  model.add(BatchNormalization())
  model.add(Activation(activation))
  model.add(Dense(y_train.shape[1], W_regularizer=regularizers.l2(l2_reg)))

model.compile(optimizer=SGD(lr=lr), loss='mse', metrics=['mae'])
```

- 7 lines of code!
- GPU accelerated learning!

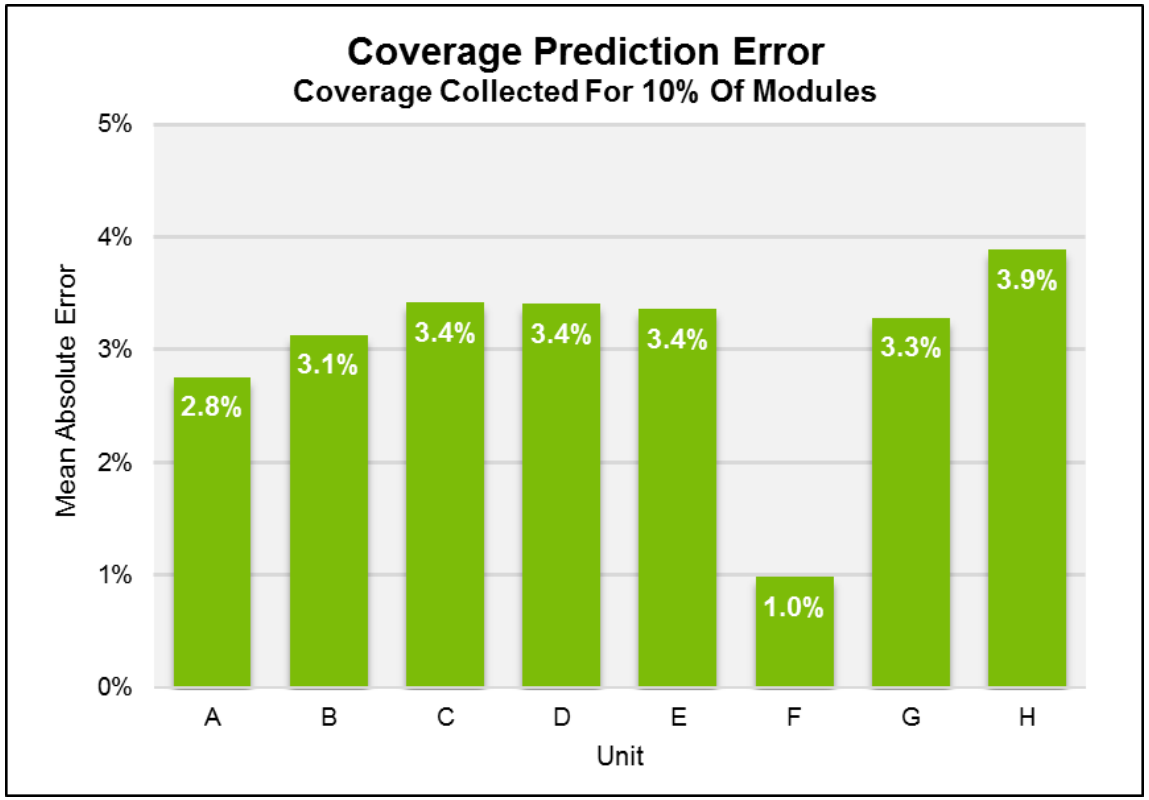


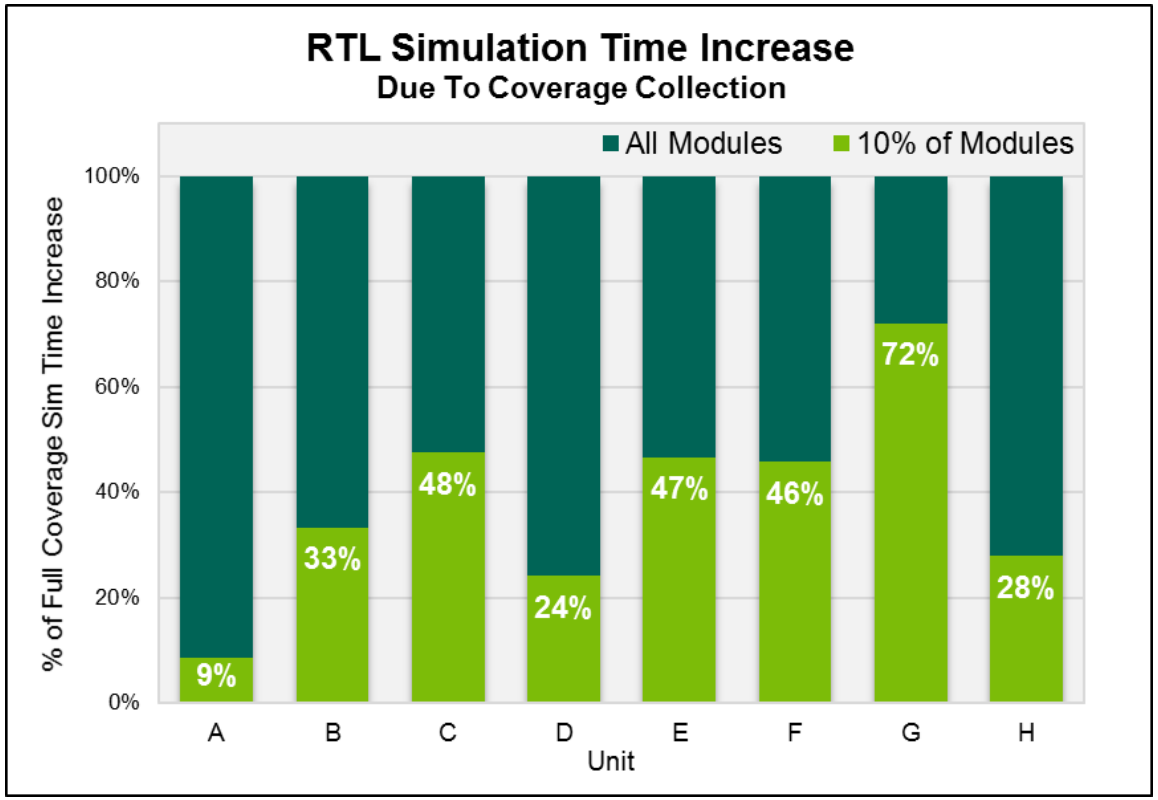
Results: Accuracy

- % modules selected for <5% error
  - **0.9% - 5.9%**
- % modules selected for <10% error
  - **0.2% - 1.2%**



Results: Speedup

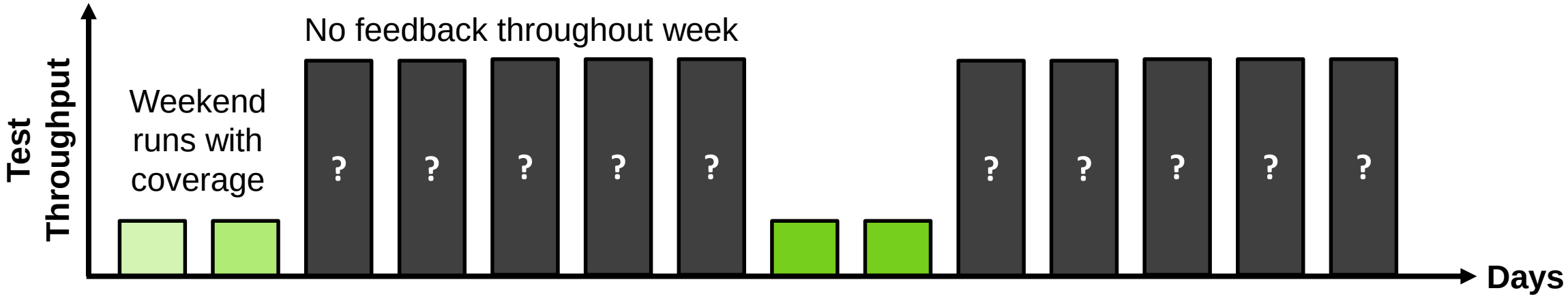




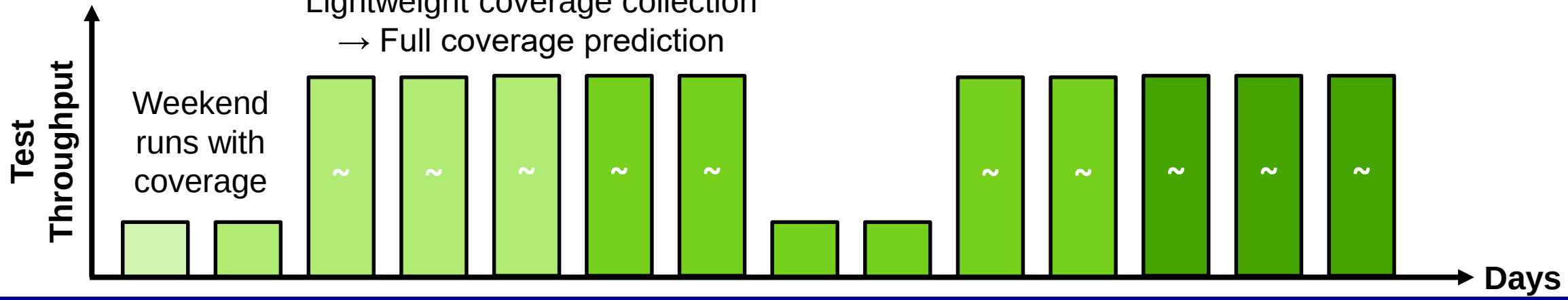
Unit A: Full coverage reports with **only 2.8% error** at **11.53x overhead reduction**

Always-On Coverage Collection

- Earlier

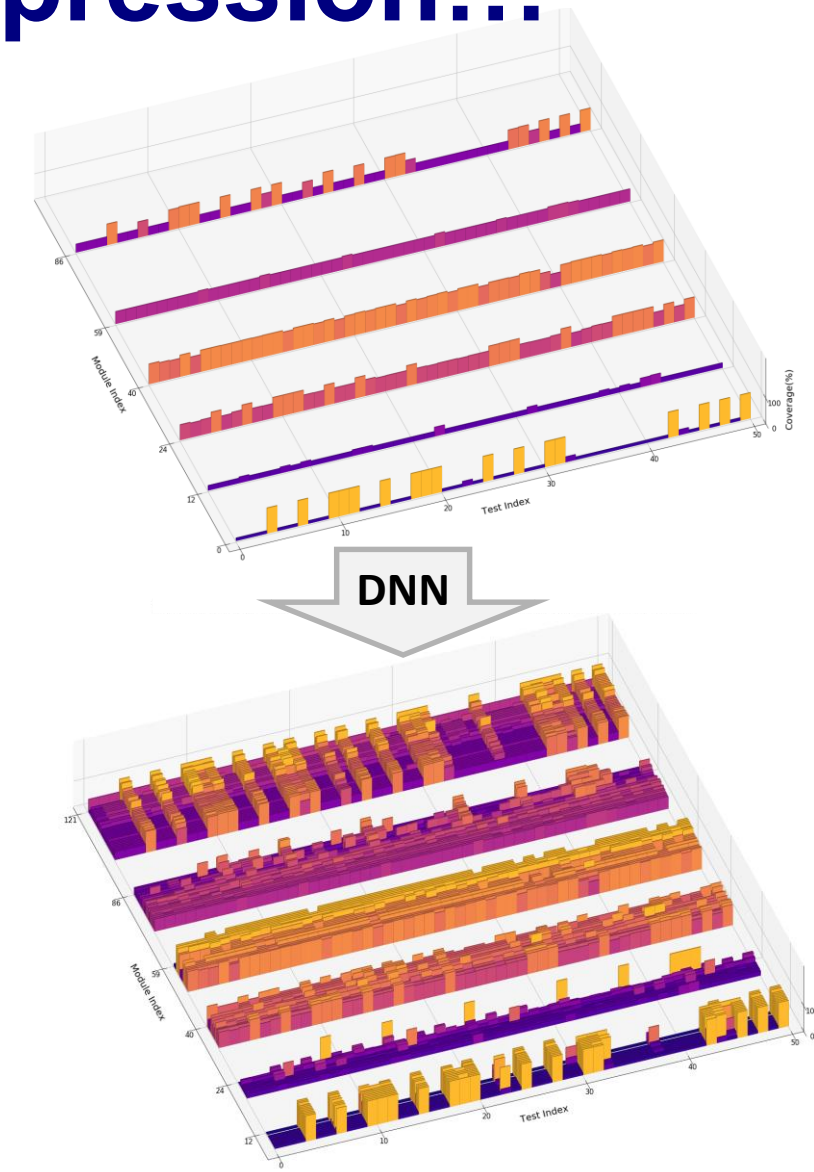


- Now

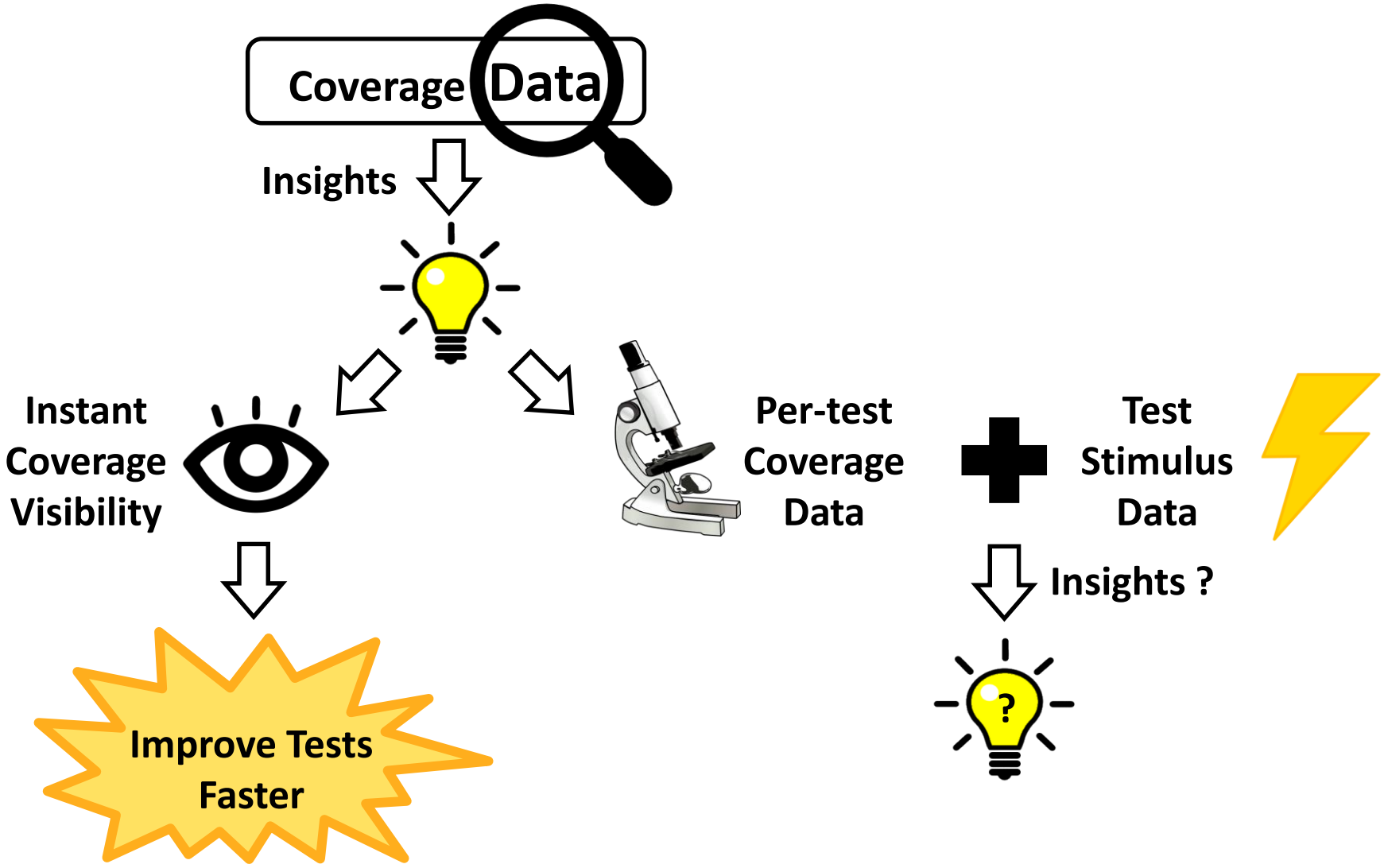


And coverage data compression…

- Store:
  - Per-test coverage data
  - For selected (10%) modules
  - 1/10th storage needed
- Predict “decompress”:
  - Per-test coverage data
  - For all modules



A data driven path forward…



2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Questions?

References:

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition" in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[2] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks" arXiv preprint arXiv:1608.06993, 2016.

[3] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks" arXiv preprint arXiv:1505.00387, 2015.

[4] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps" arXiv preprint arXiv:1312.6034, 2013.

[5] R. K. Brayton, G. D. Hachtel, C. McMullen and A. Sangiovanni-Vincentelli, "Logic minimization algorithms for VLSI synthesis. Vol. 2" Springer Science & Business Media, 1984.

[6] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1" arXiv preprint arXiv:1602.02830, 2016.

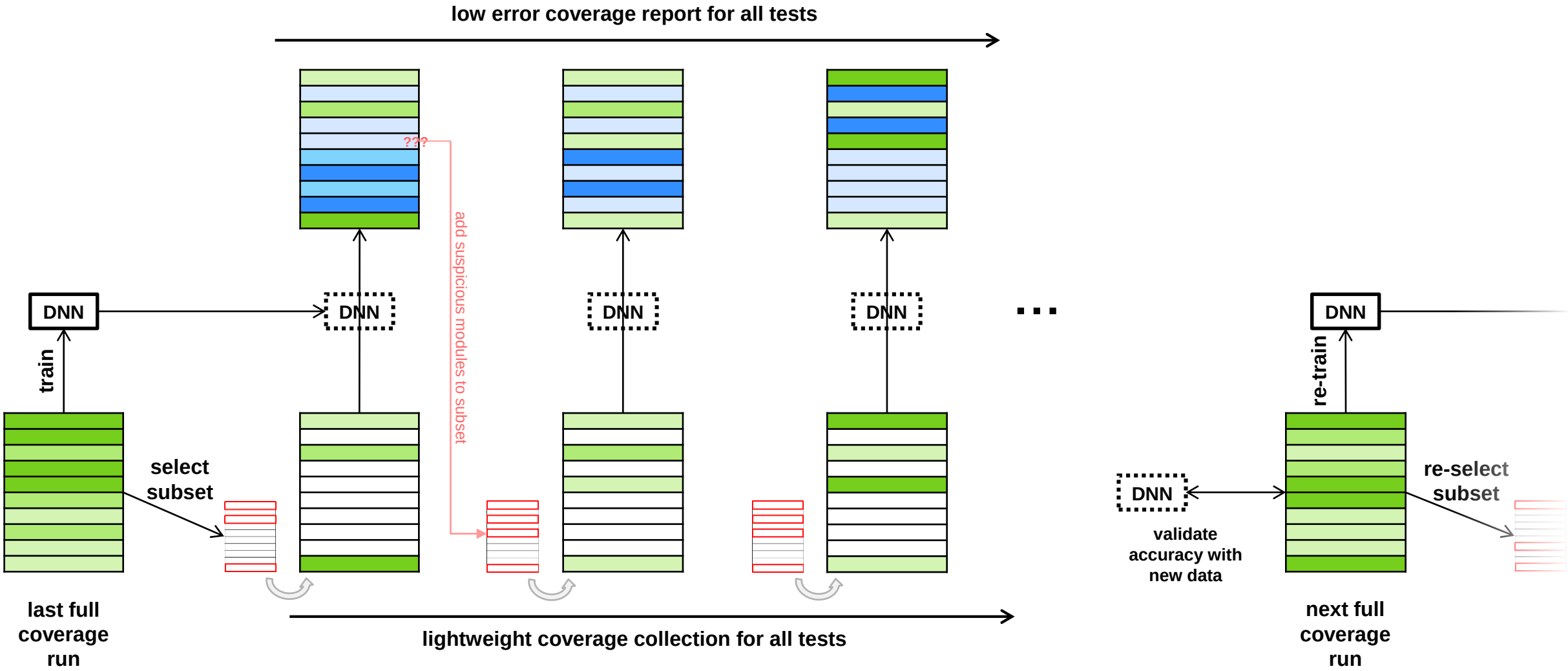
Supplements

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Next Steps

- Improve on current results
  - Better neural net architecture (ResNets[1], DenseNets[2], HighwayNets[3])
  - Let DNN select modules instead of k-means (saliency maps[4])
  - Optimize module subset selection for simulation time reduction
  - Evaluate robustness:
    - Extent/nature of change to design/stimulus before DNN accuracy drops
- Scale up to per-line/per-condition inference granularity
  - Binary coverage values and Boolean relations
  - Logic minimization[5] instead of k-means?
  - Binarized neural networks [6]

Overall flow



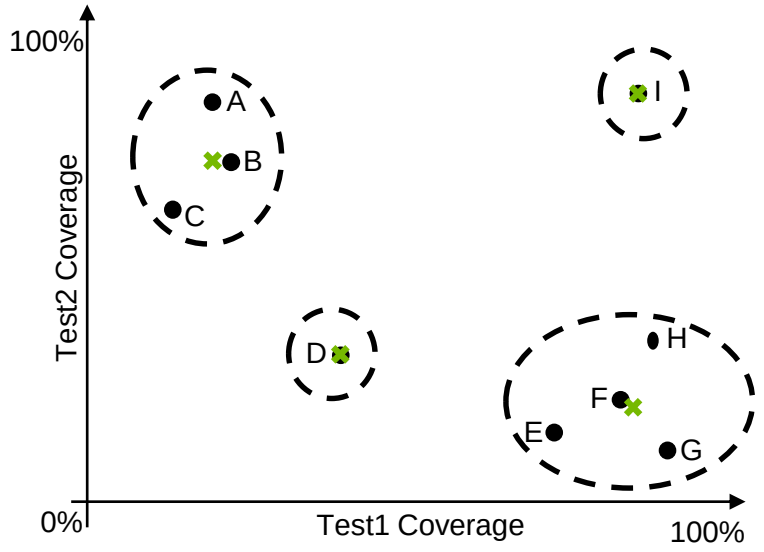
K-means clustering

- Given a dataset of module coverage across multiple tests
  - MxT matrix (M modules X T tests)
  - Each module represented as a vector of length T (it's coverage across tests)
- K-means clustering clusters the modules into K clusters
  - Modules in a cluster have similar test coverage vectors

Right:

4-means clustering clusters 9 modules into 4 clusters based on coverage from 2 tests.

Each clusters' center (mean) is shown with ✖



Evaluation: Metrics

- Revisiting Goal: High quality coverage feedback with low overhead
- Low overhead:
  - % of full design sampled (smaller the better)
  - Simulation time overhead of coverage collection (lower the better)
- High quality:
  - Error in inferring full coverage (lower the better)
  - Error defined as: Mean Absolute Error

Actual Coverage	Test 1	Test 2
Module A	10%	20%
Module B	80%	50%

Inferred Coverage	Test 1	Test 2
Module A	10%	25%
Module B	75%	60%

Absolute Error	Test 1	Test 2
Module A	0%	5%
Module B	5%	10%

Mean Absolute Error
5%

Evaluation: Dataset

- Module Condition Coverage (modules == module instances)
- 8 NVIDIA GPU units of various sizes
- Various test suites split into:
  - Training sets (clustering, subset selection, DNN training)
  - Validation sets (validating inference against actual coverage)

	# modules	# training tests (40%)	# validation tests (60%)
Unit A	82	13416	20124
Unit B	121	5864	8796
Unit C	176	8612	12918
Unit D	224	20676	31014
Unit E	268	3964	5946
Unit F	410	3772	5658
Unit G	500	12752	19128
Unit H	574	13224	19836