SAND2017-11820C

Sandia National Laboratories

Learning Noise in Quantum Information Processors

Travis L Scholten
@Travis_Sch

Center for Quantum Information and Control, University of New Mexico, Albuquerque, USA

Center for Computing Research, Sandia National Laboratories, Albuquerque, USA

QTML 2017

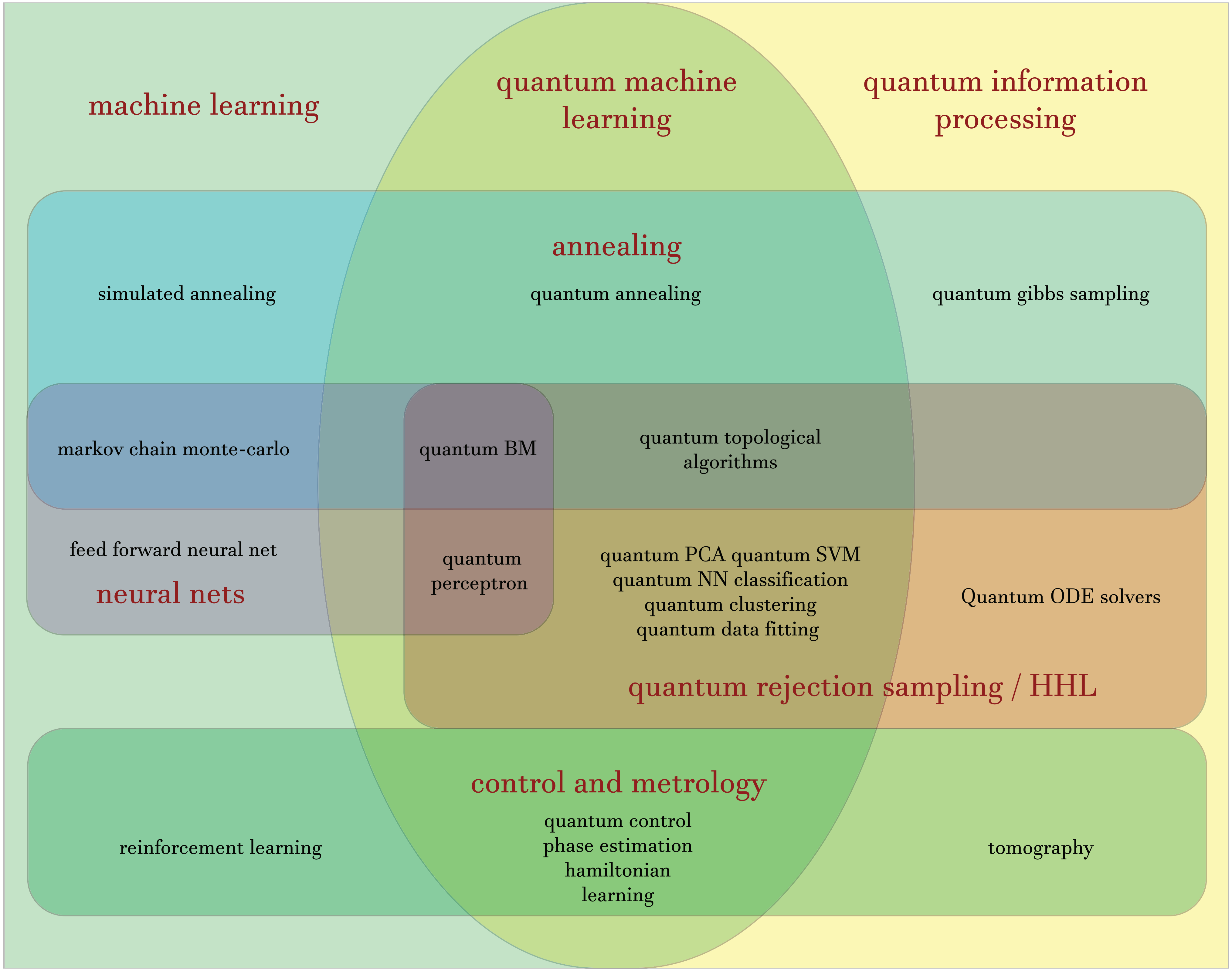
U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

CCR Center for Computing Research

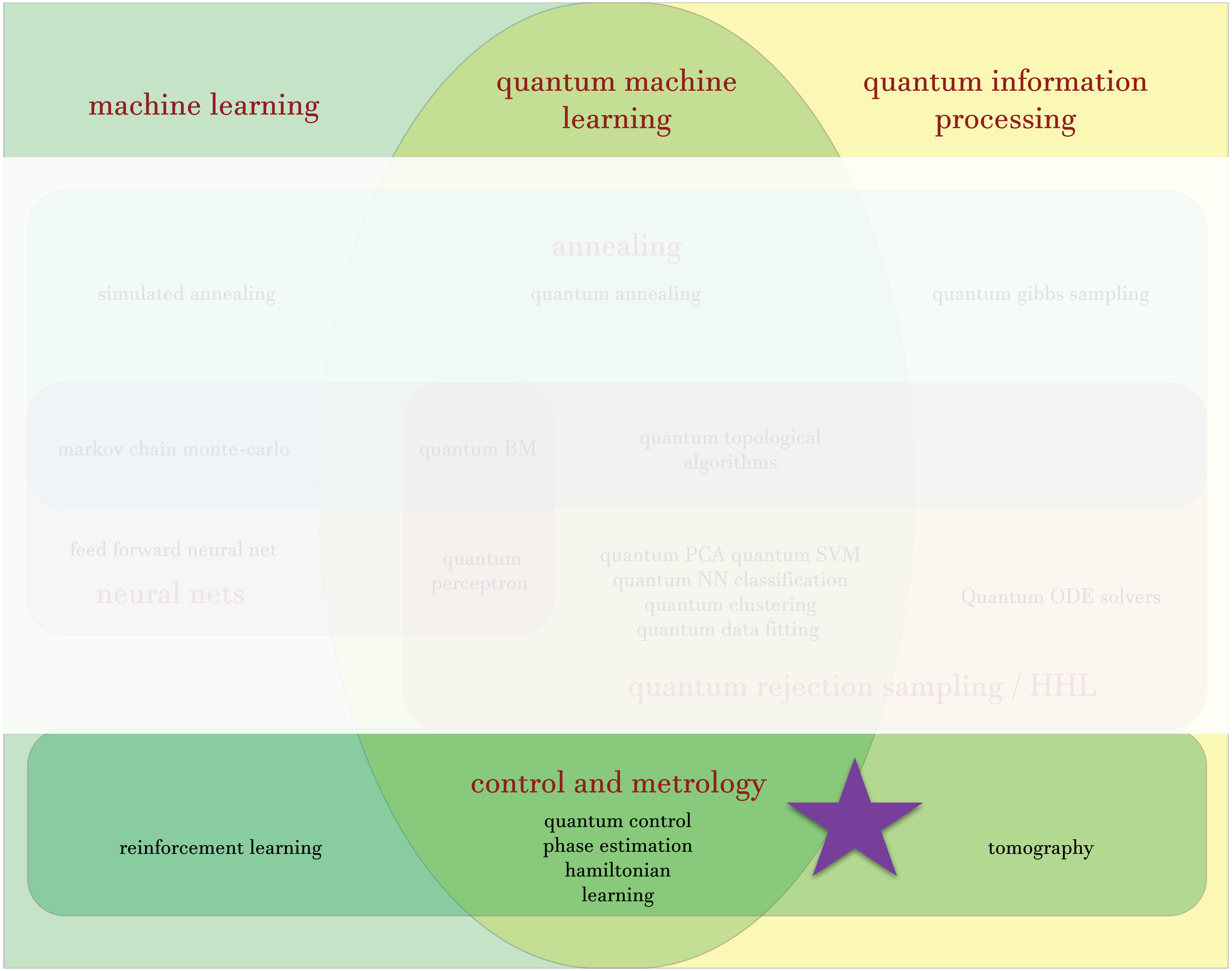
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

There are lots of applications at the intersection of QI/QC and ML…



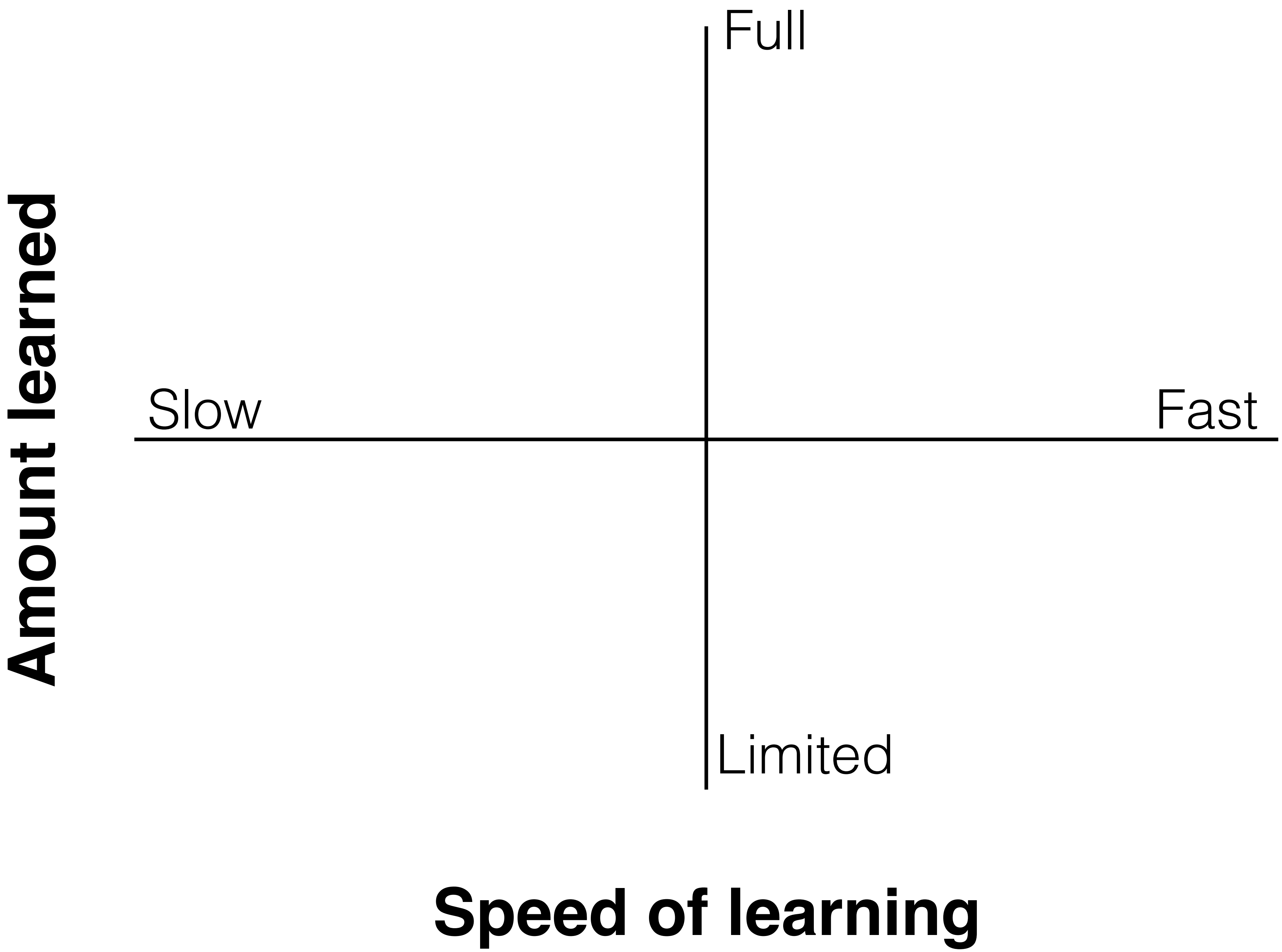
Biamonte, et. al, arXiv: 1611.09347

…I want to focus on how ML can improve *characterization of quantum hardware*.

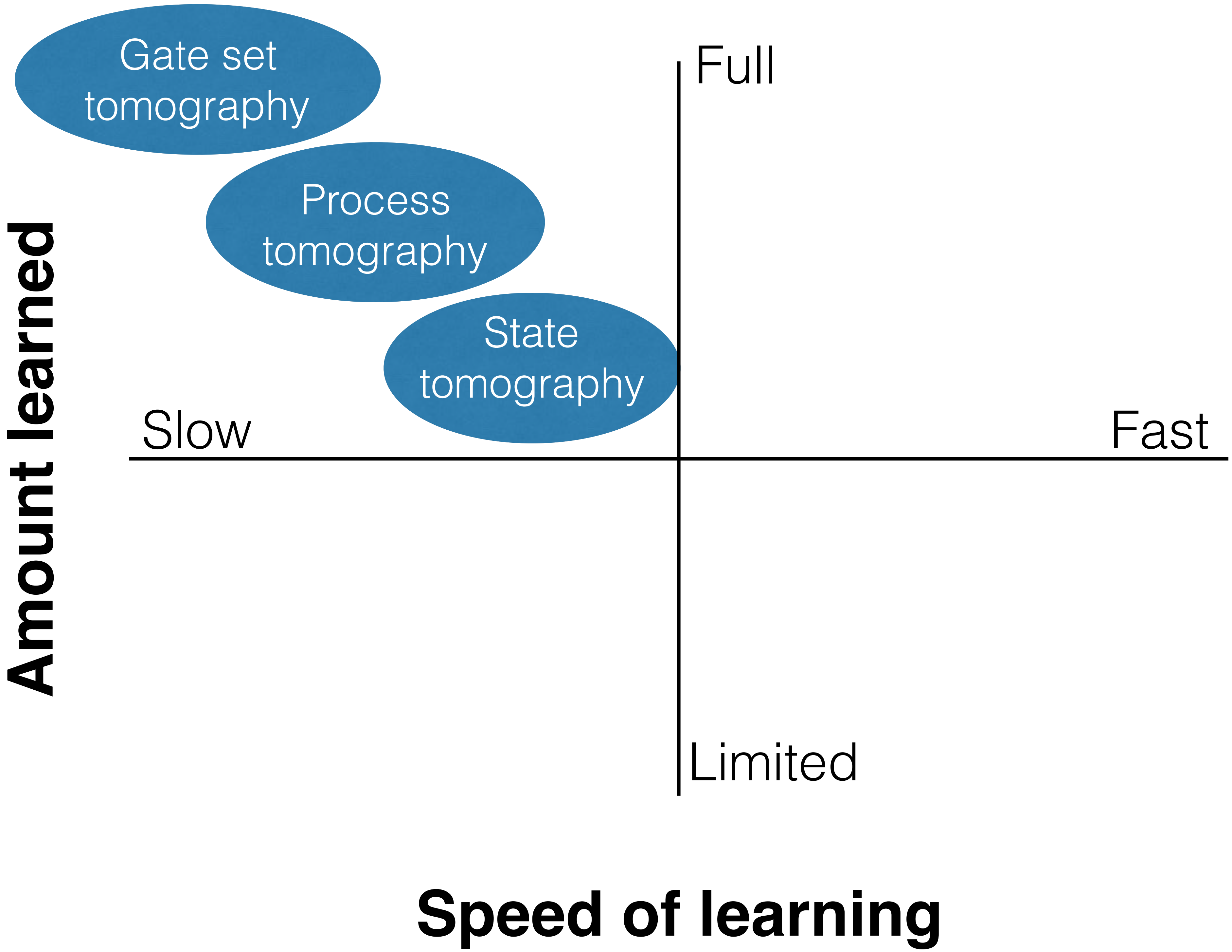


Biamonte, et. al, arXiv: 1611.09347

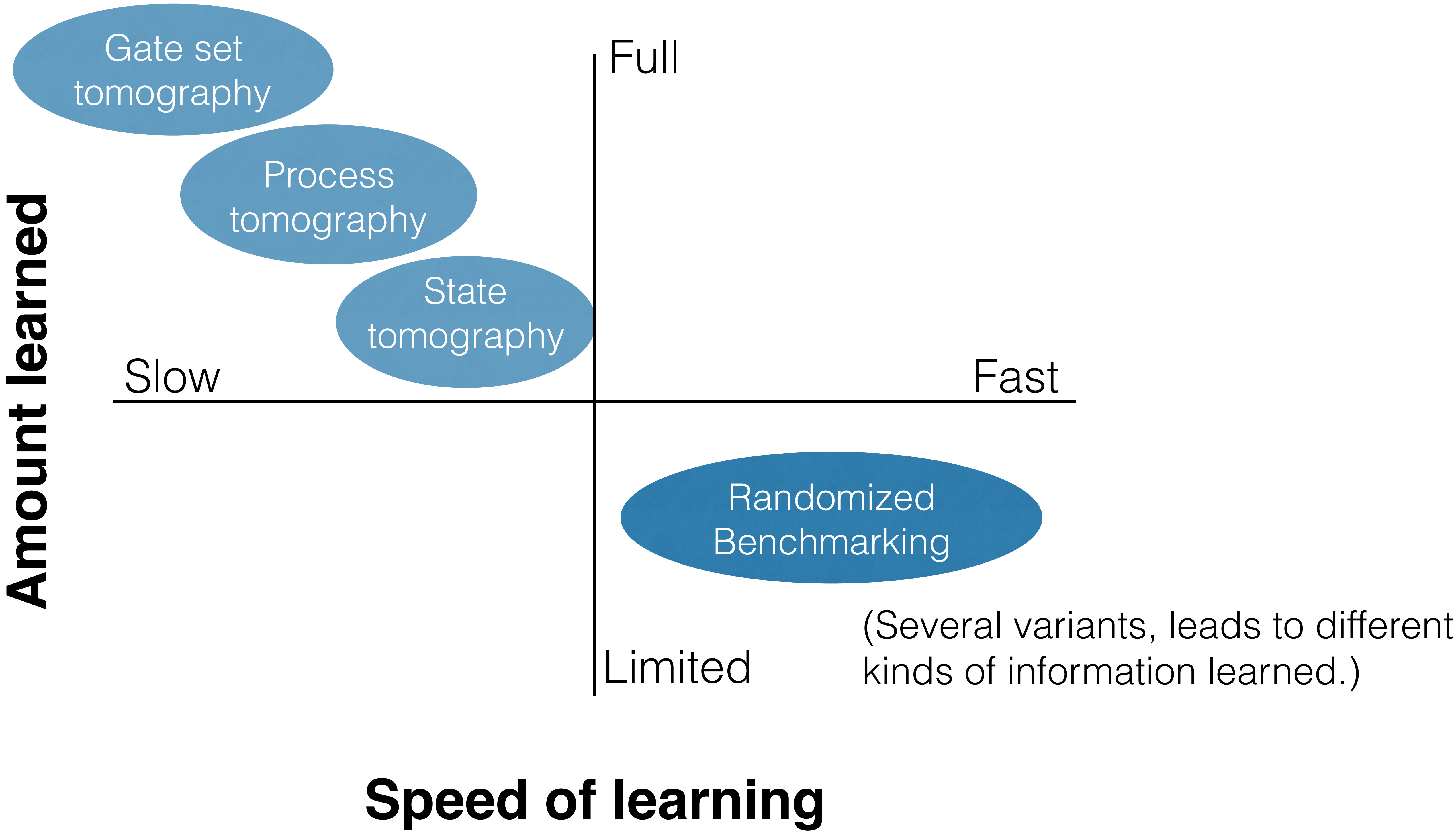
Quantum device characterization (QCVV) techniques arranged by amount learned and time required.



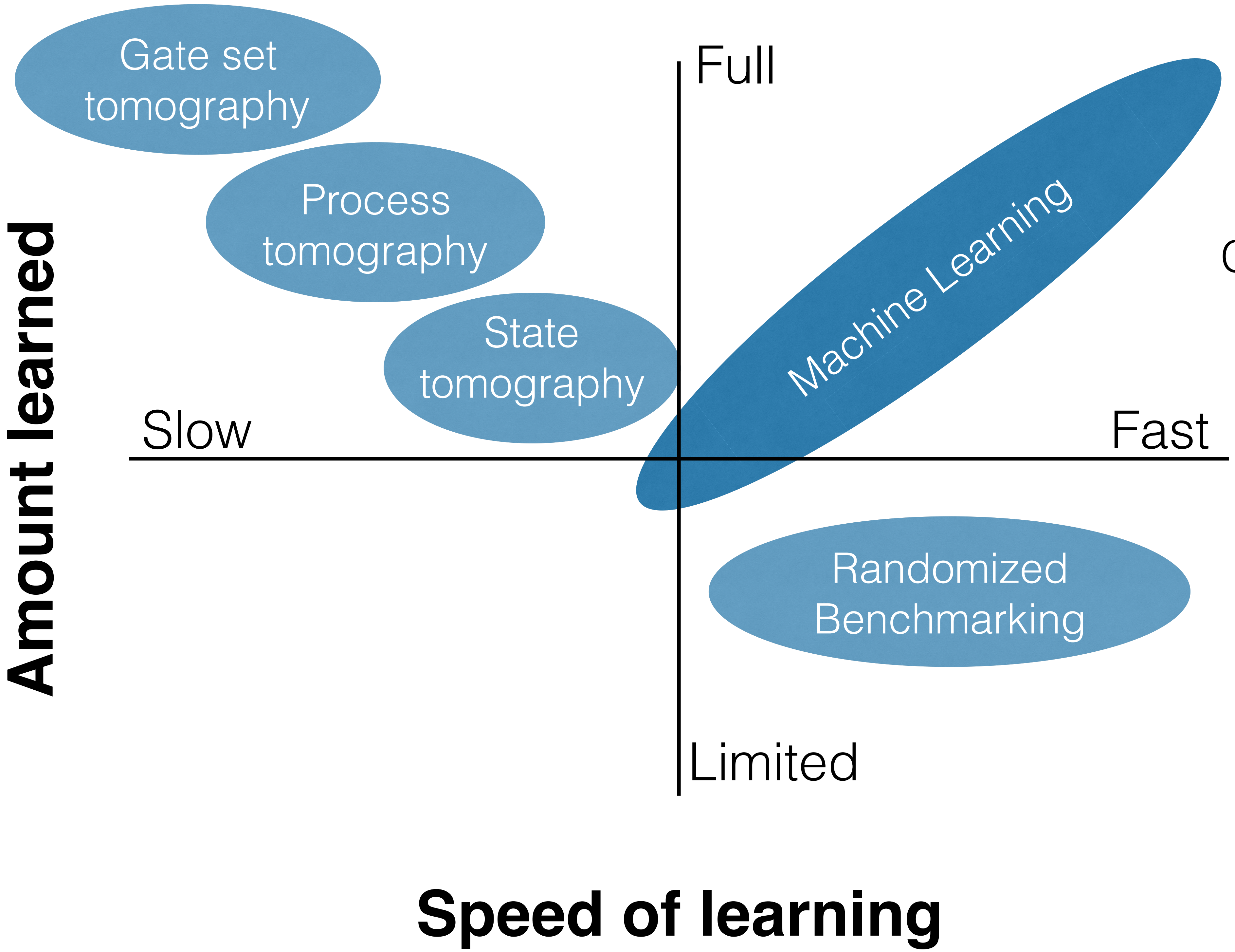
Tomography is *very informative*, but *time-consuming*!



Randomized benchmarking is *fast*,
but yields *limited information*.

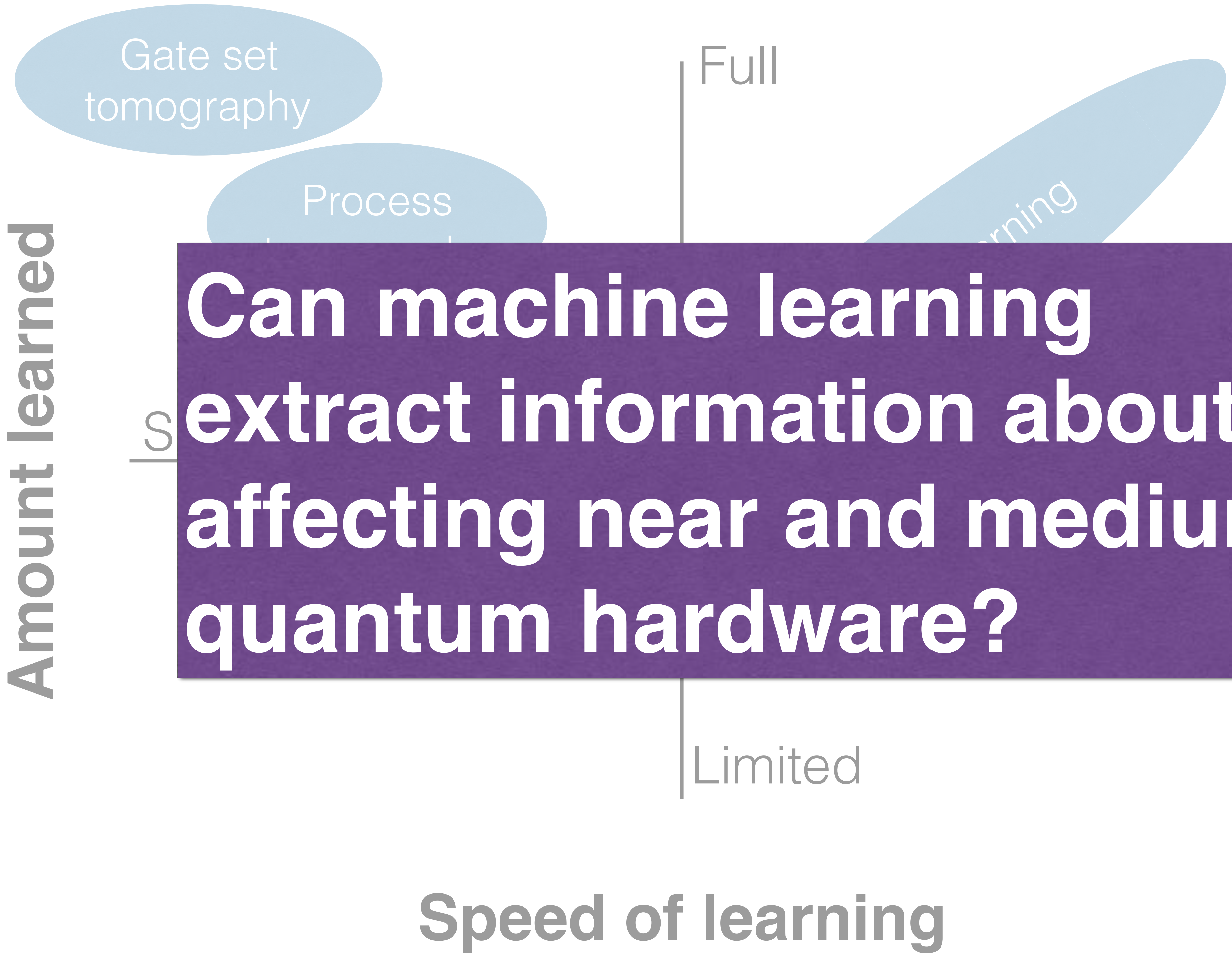


Depending on how much we want to learn, and how quickly, machine learning could be useful.



Caveat: "Speed" doesn't include *training time*

Depending on how much we want to learn, and how quickly, machine learning could be useful.



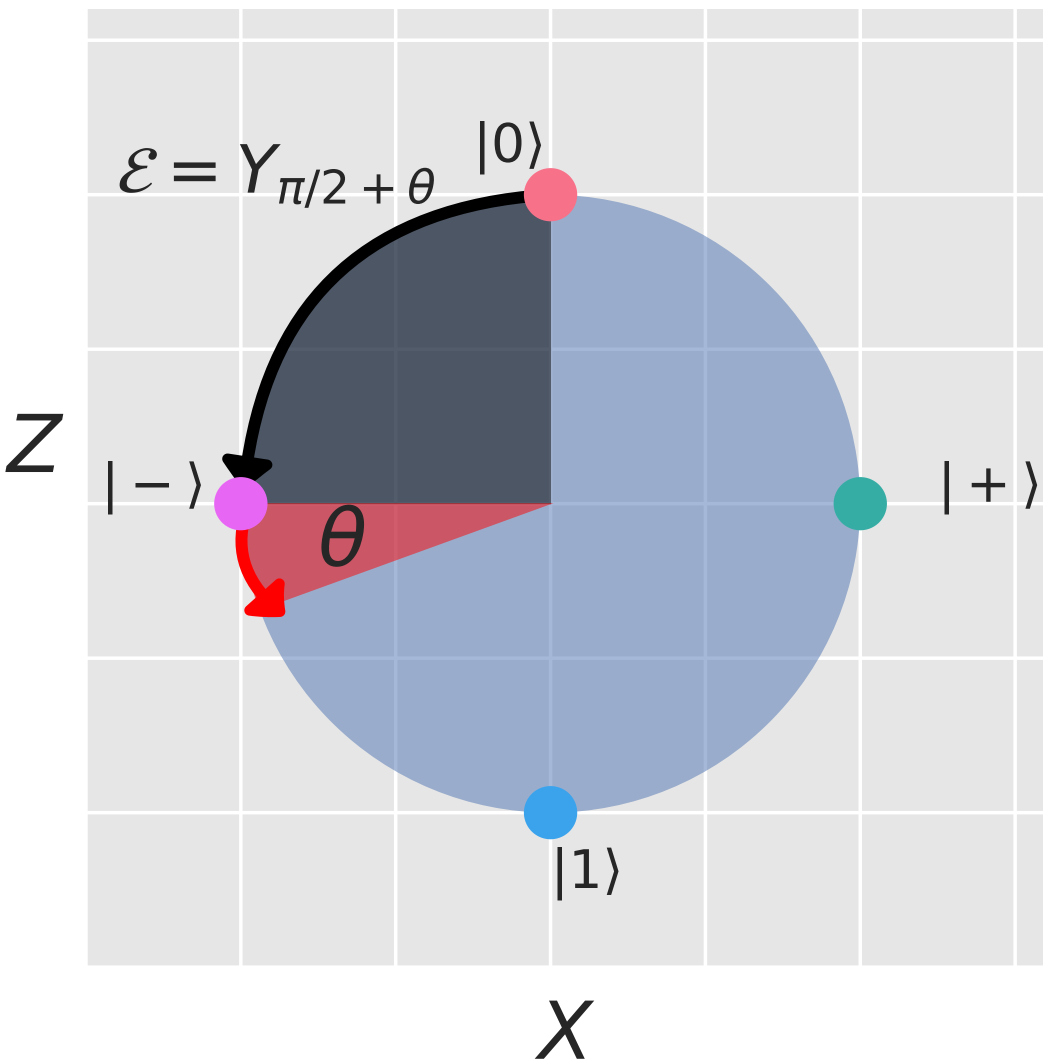
Can machine learning extract information about noise affecting near and medium-term quantum hardware?

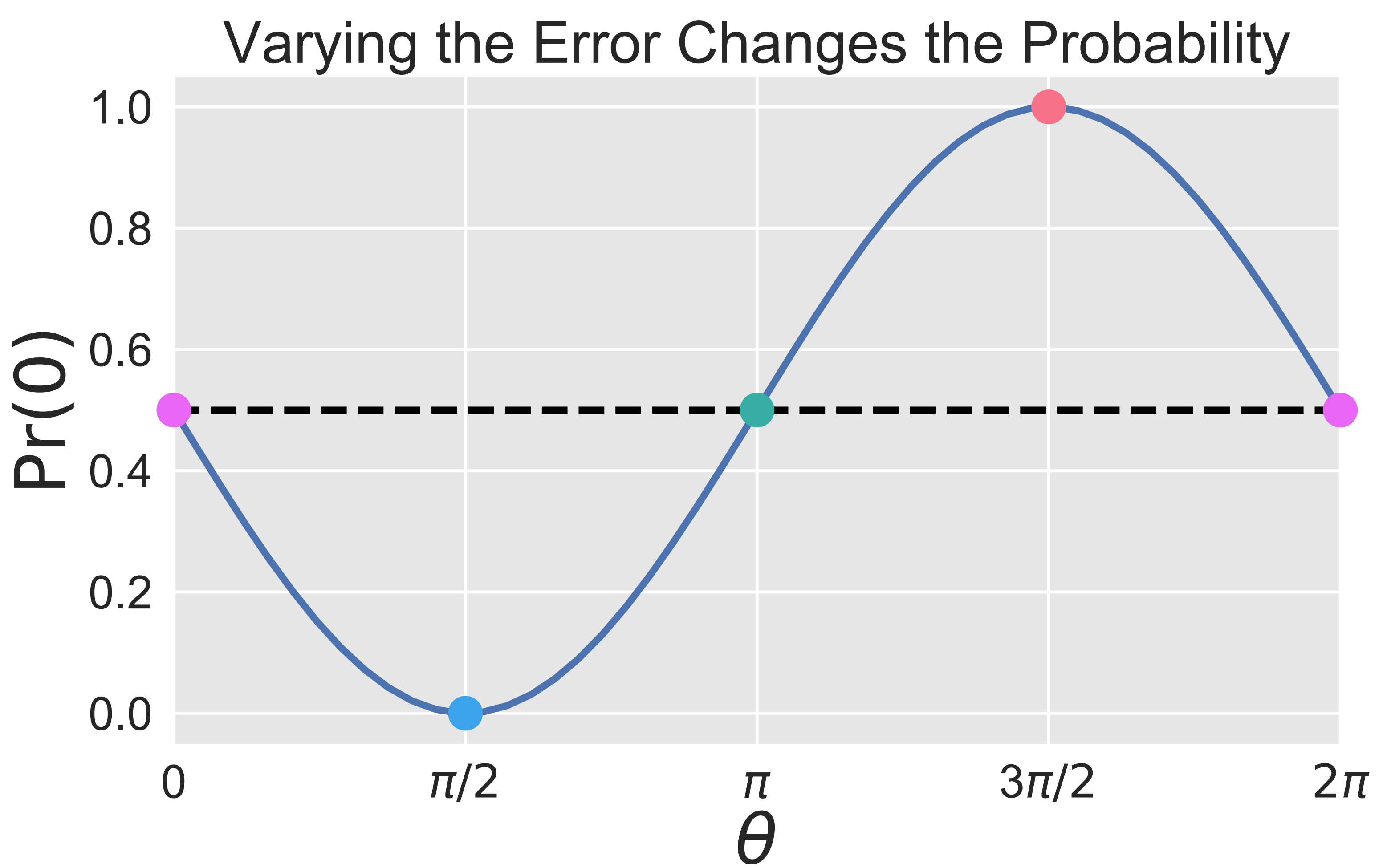
Noise affects the
outcome probabilities of
quantum circuits.

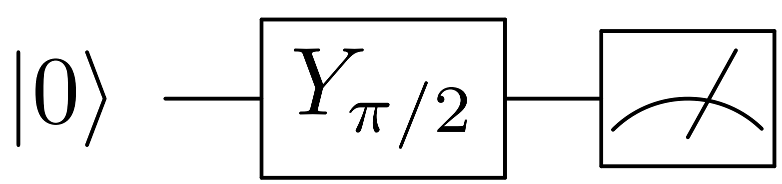
How can we **learn** about **noise** using the **data** we get from running **quantum circuits**?

Noise in quantum hardware affects the outcome probabilities of circuits.

Example: over-rotation error of a single-qubit gate







(The circuit we write down)

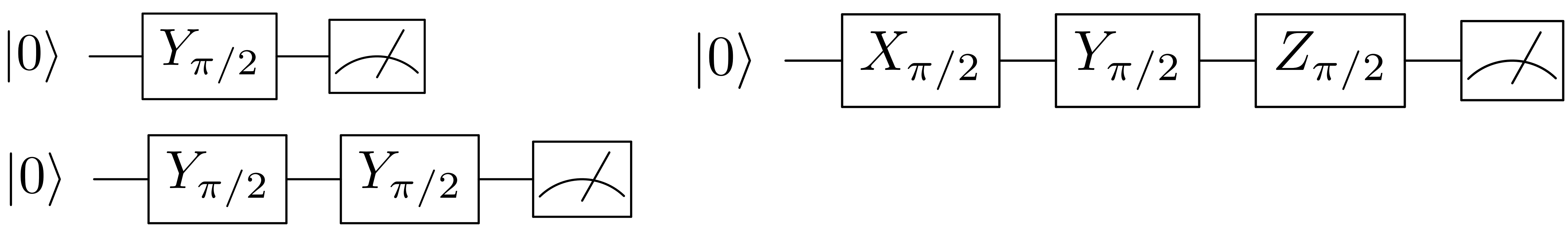
$$\Pr(0) = \mathrm{Tr}(|0\rangle\langle 0|\mathcal{E}(|0\rangle\langle 0|)) = \frac{1}{2}(1 - \sin\theta)$$

(Noise affects outcome probability)

Gate set tomography (GST) provides a set of structured circuits we can use for learning.

GST assumes the device is a black box, described by a *gate set.*

GST prescribes certain circuits to run that collectively *amplify all types of noise*.

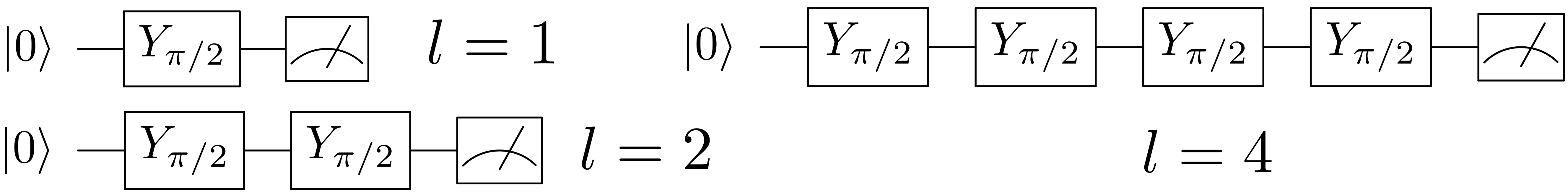


Standard use: Outcome probabilities are analyzed by pyGSTi software to estimate the noisy gates

Blume-Kohout, et. al, arXiv 1605.07674

Gate set tomography (GST) provides a set of structured circuits we can use for learning.

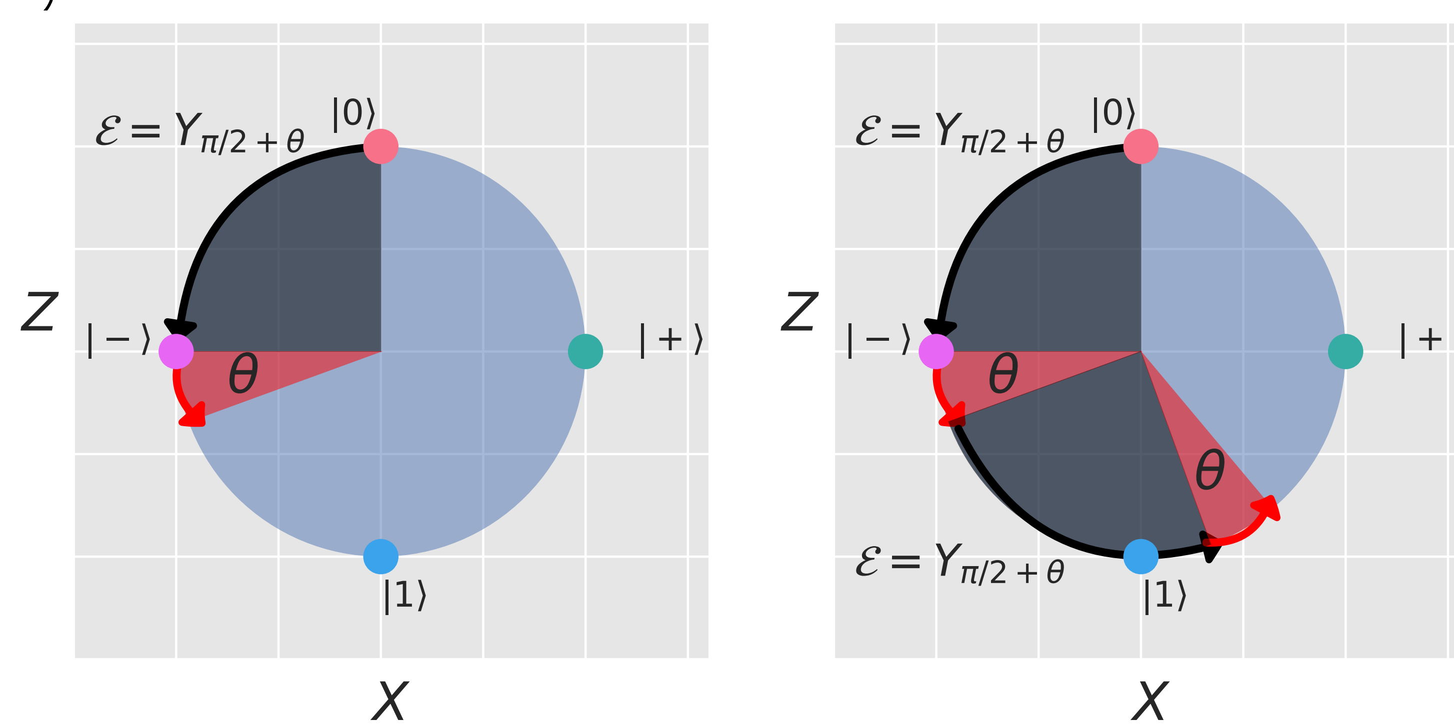
GST prescribes certain circuits to run that collectively *amplify all types of noise*.



Circuits have varying length, up to some maximum length L.

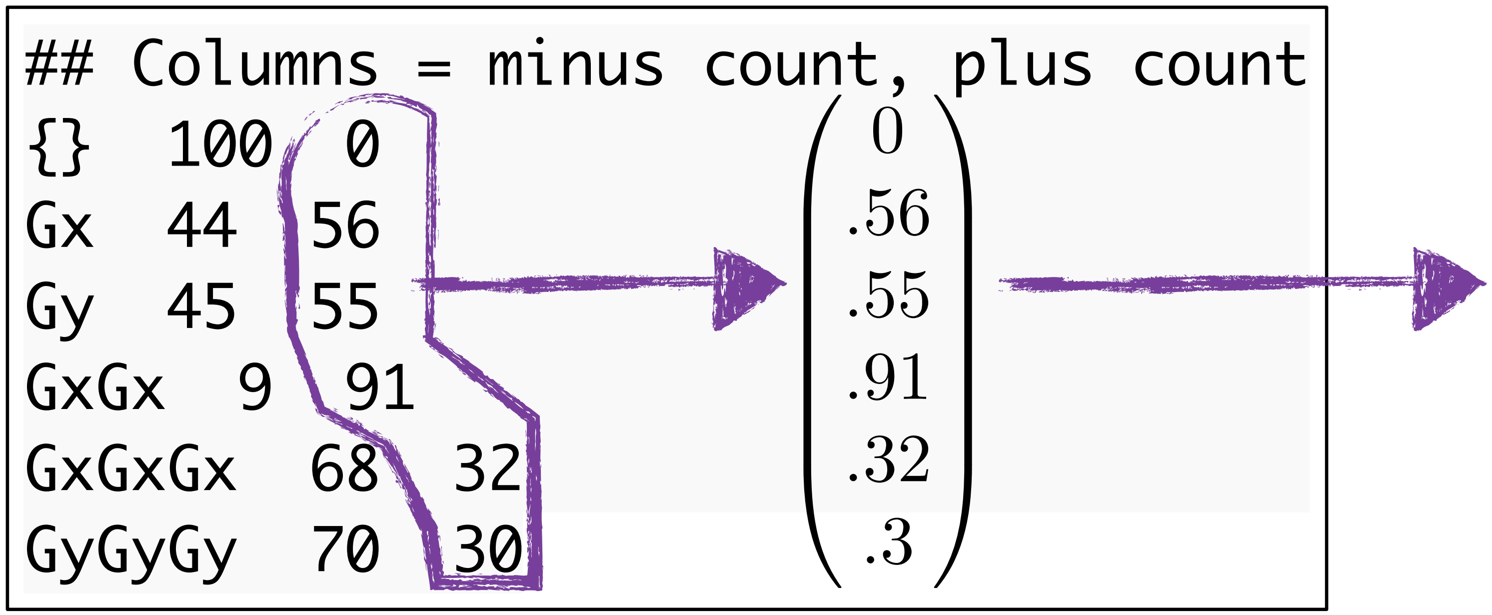
$$l = 1, 2, 4, \cdots, L$$

Why? Because longer circuits are more sensitive to noise.



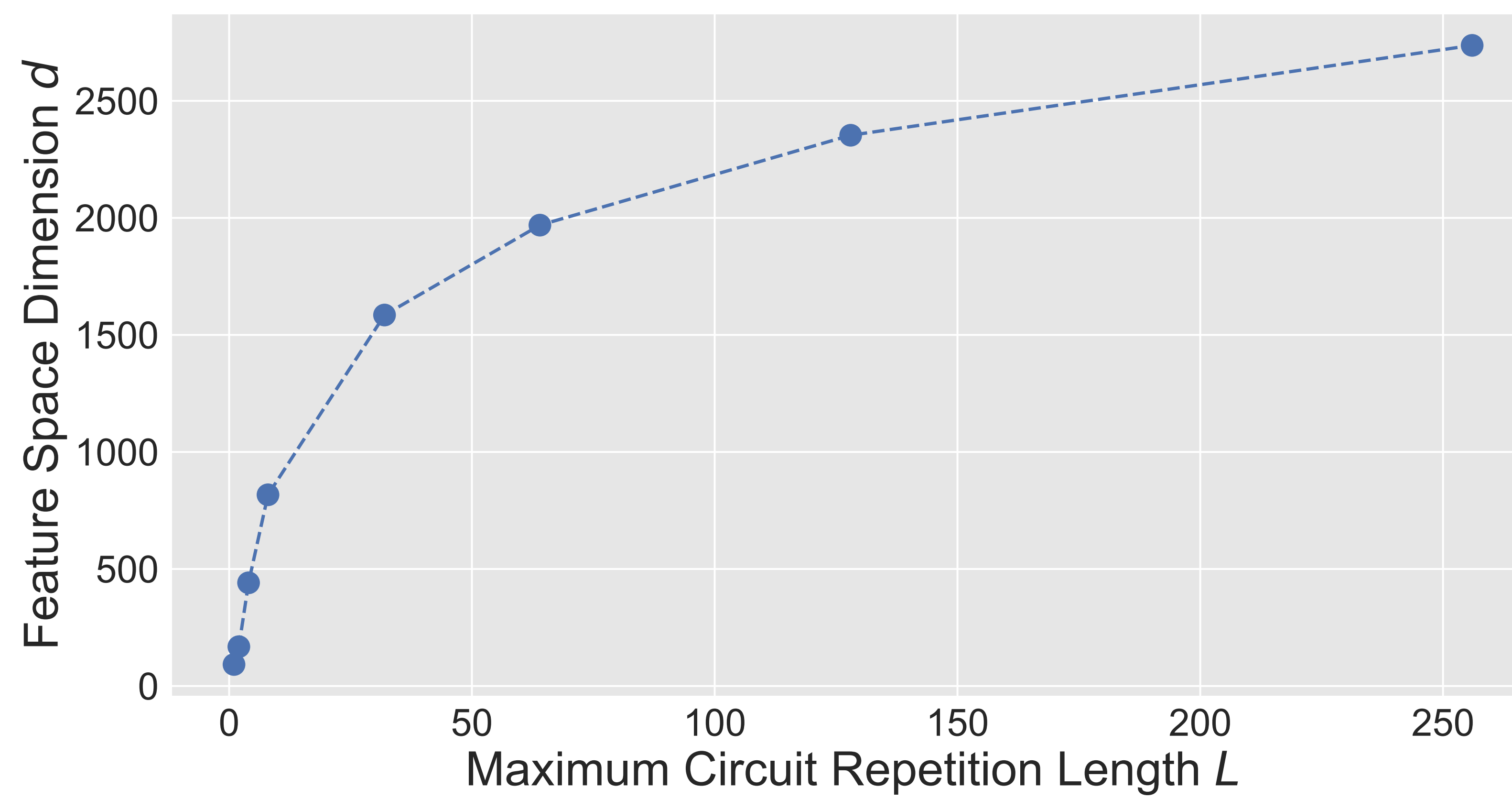
To do machine learning on GST data sets, embed them in a feature space.

(GST data set)

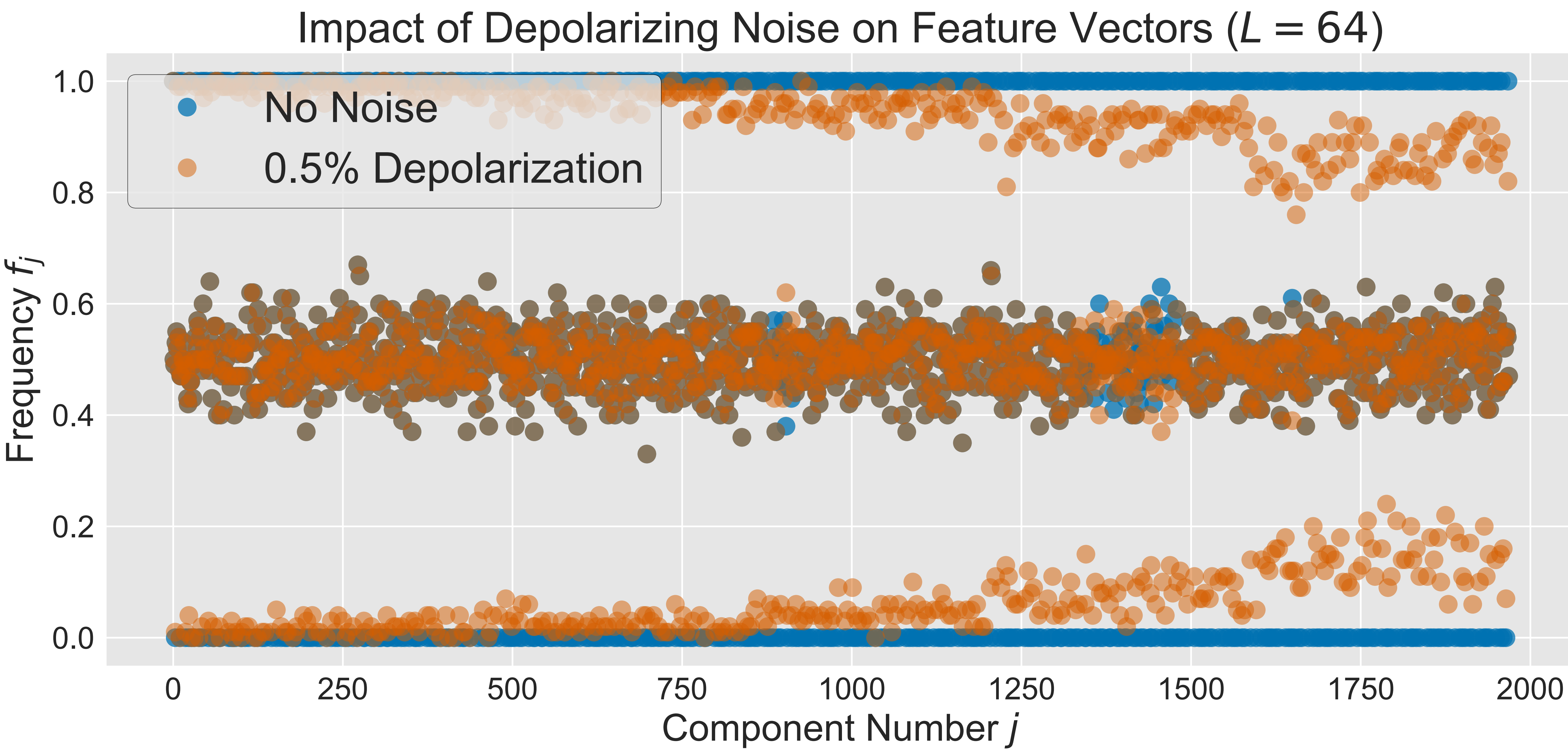


$$\mathbf{f} = (f_1, f_2, \cdots) \in R^d$$

The dimension of the feature space grows with L because more circuits are added.



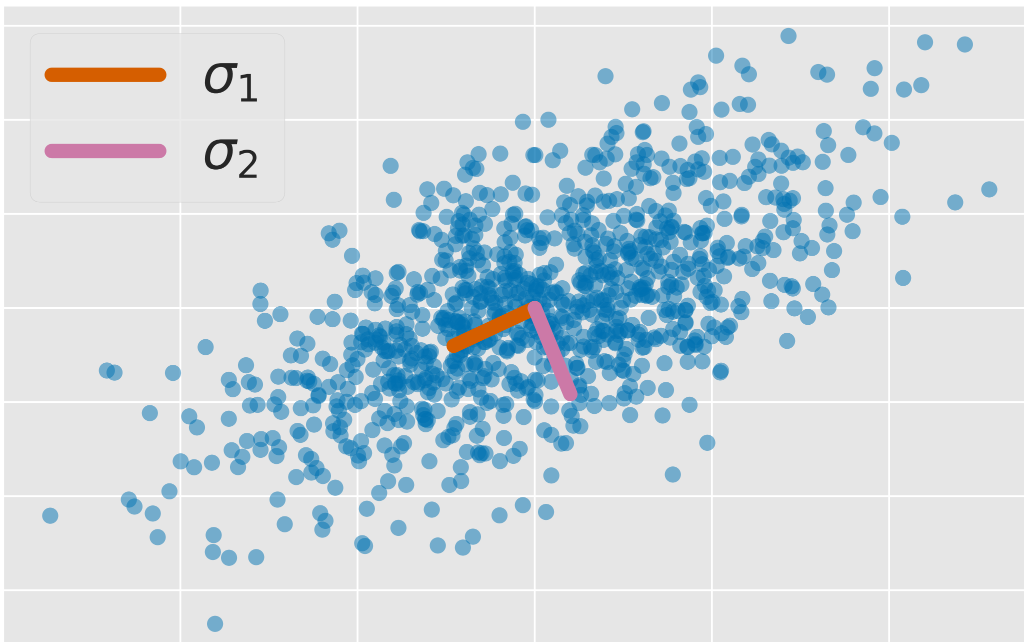
Noise changes some components of the feature vectors.



How can we identify the “signature” of a noise process using GST feature vectors?

Principal component analysis (PCA) reveals a structure to GST feature vectors.

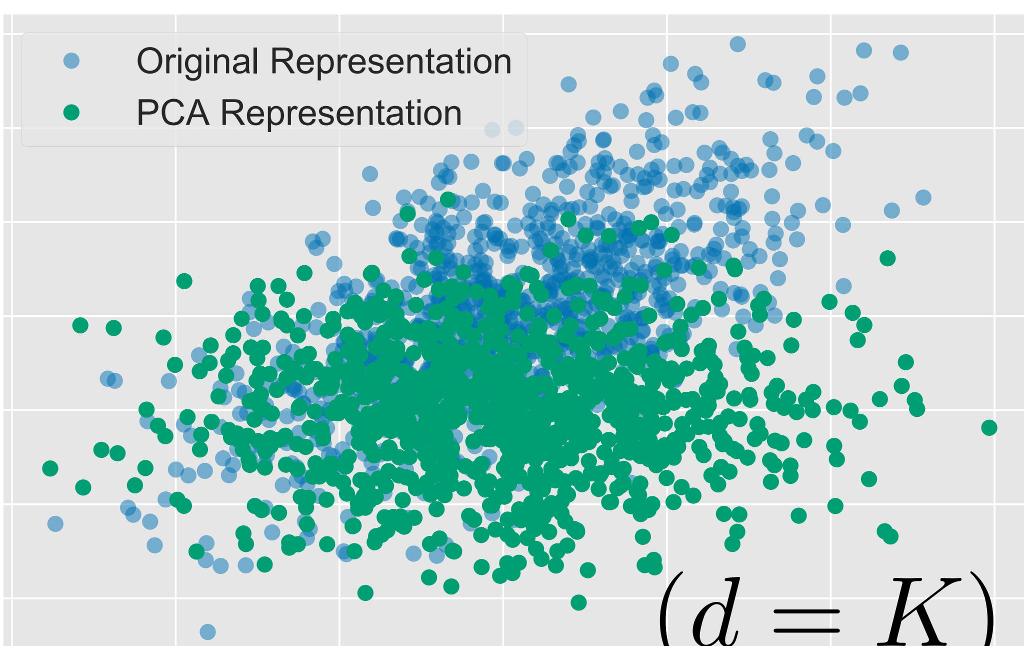
PCA finds a low-dimensional representation of data by looking for directions of *maximum variance*.



Compute covariance matrix & diagonalize

$$C = \sum_{j=1}^{K} \sigma_j \boldsymbol{\sigma}_j \boldsymbol{\sigma}_j^T$$

$$\sigma_1 \geq \sigma_2 \cdots \geq \sigma_K$$

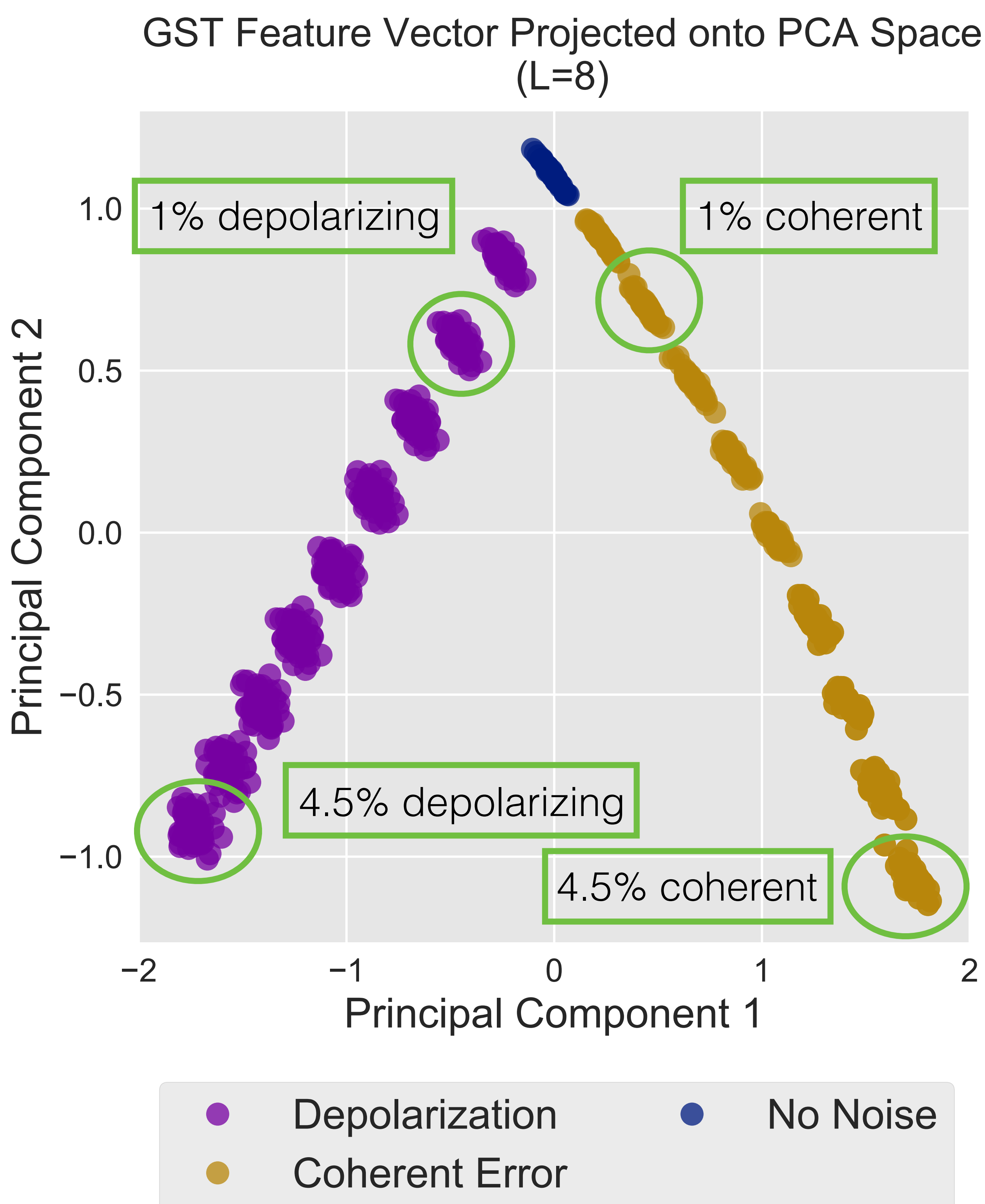


$$(d = K)$$

Defines a map:

$$\mathbf{f} \to \sum_{j=1}^{K} (\mathbf{f} \cdot \boldsymbol{\sigma}_j) \boldsymbol{\sigma}_j$$

Projection onto a 2-dimensional PCA subspace reveals a *structure to GST feature vectors*.

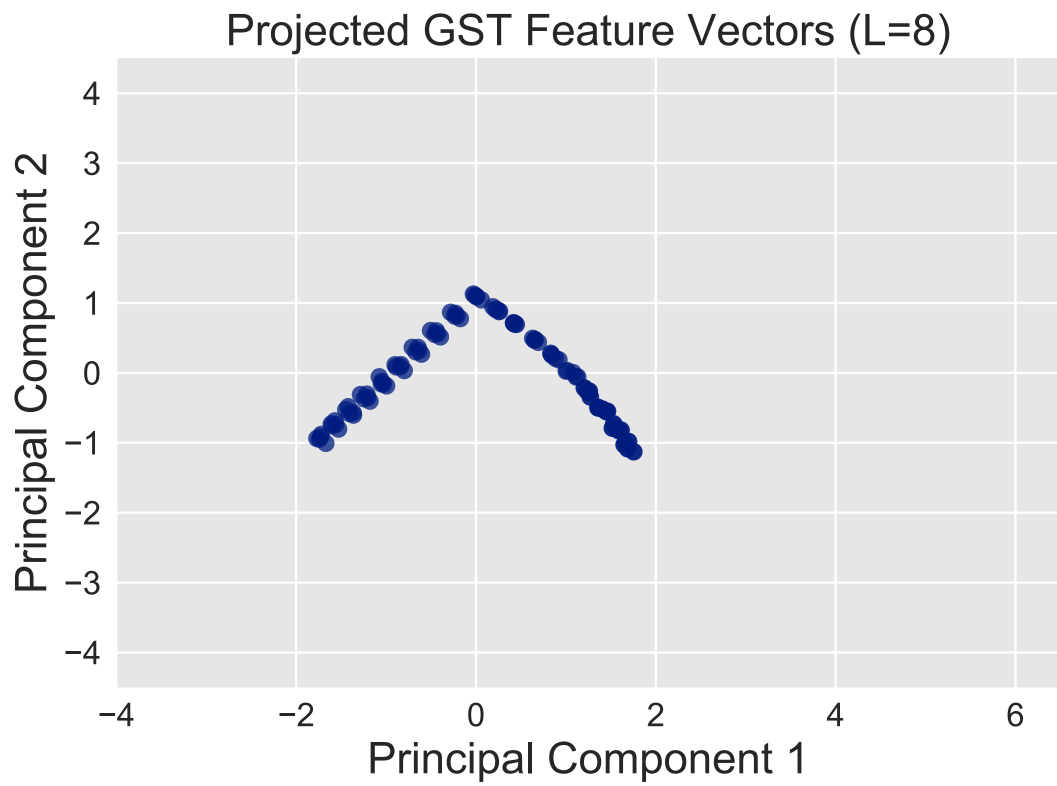


Different noise types and noise strengths tend to **cluster**!

(PCA performed on entire dataset, then individual feature vectors transformed.)

Noise Type	Number of Feature Vectors	Number of Noise Strengths
Coherent Error	450	9
Depolarization	450	9
No Noise	50	1

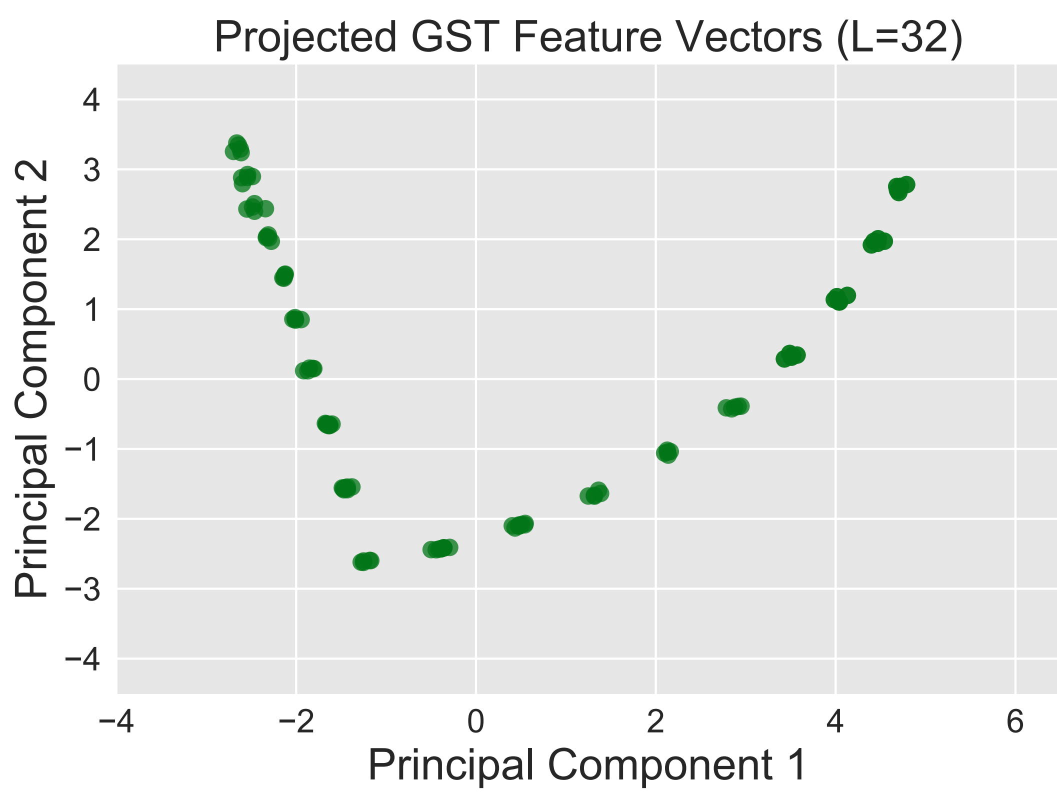
Adding longer circuits makes the clusters more distinguishable.

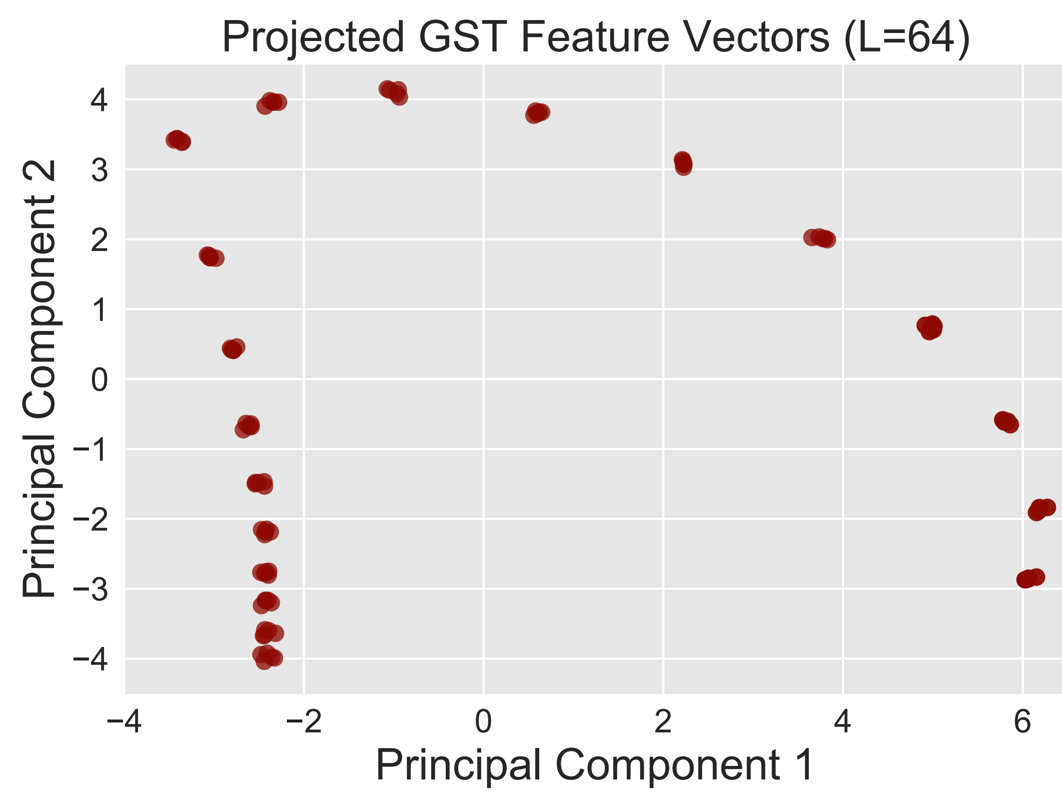


Longer GST circuits amplify noise, making the clusters more distinguishable.

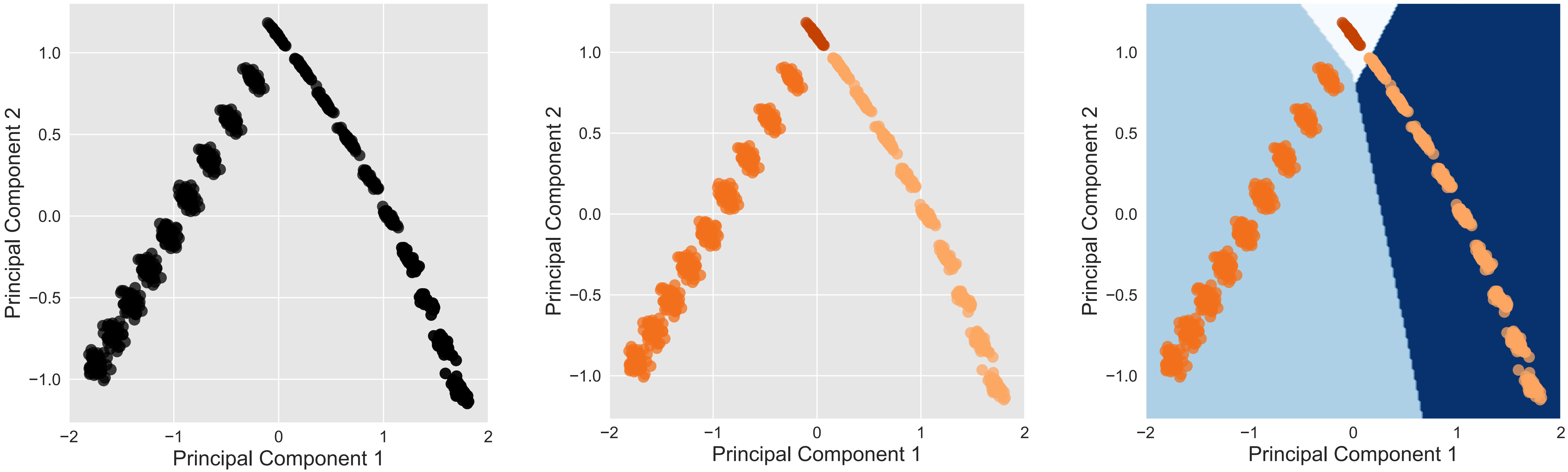
We can use this structure to do **classification**!

(An independent PCA was done for each L.)





Classification is possible because the data sets cluster based on noise type and strength!

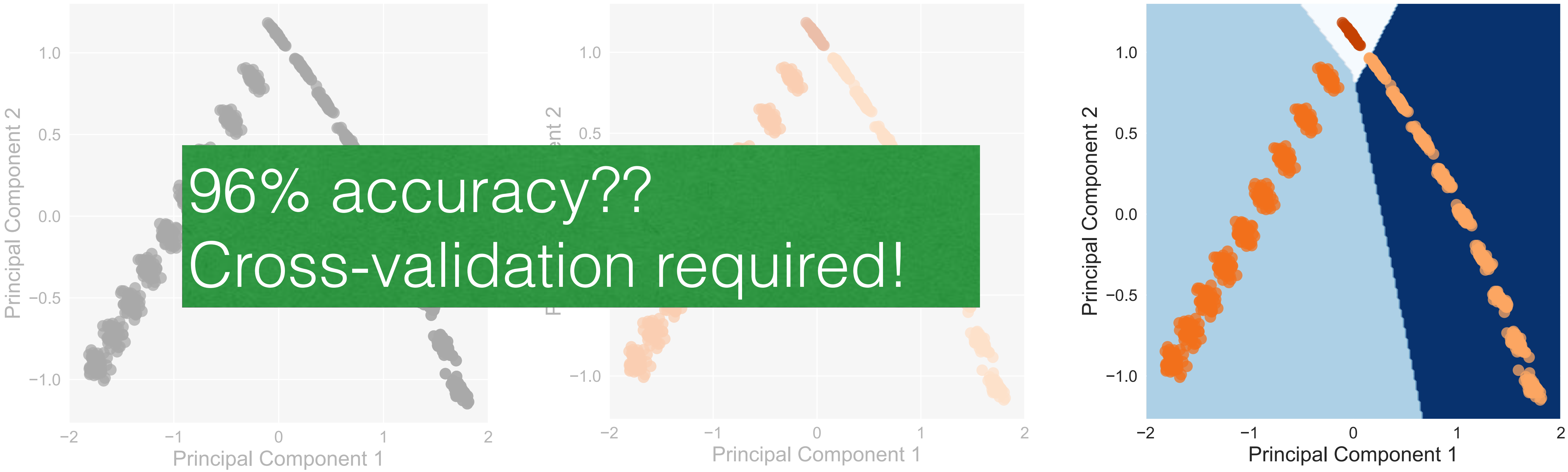


Project feature vectors based on **PCA**

Label feature vectors based on **noise**

Train a **soft-margin**, **linear** support vector machine (**SVM**)

Classification is possible because the data sets cluster based on noise type and strength!



96% accuracy??
Cross-validation required!

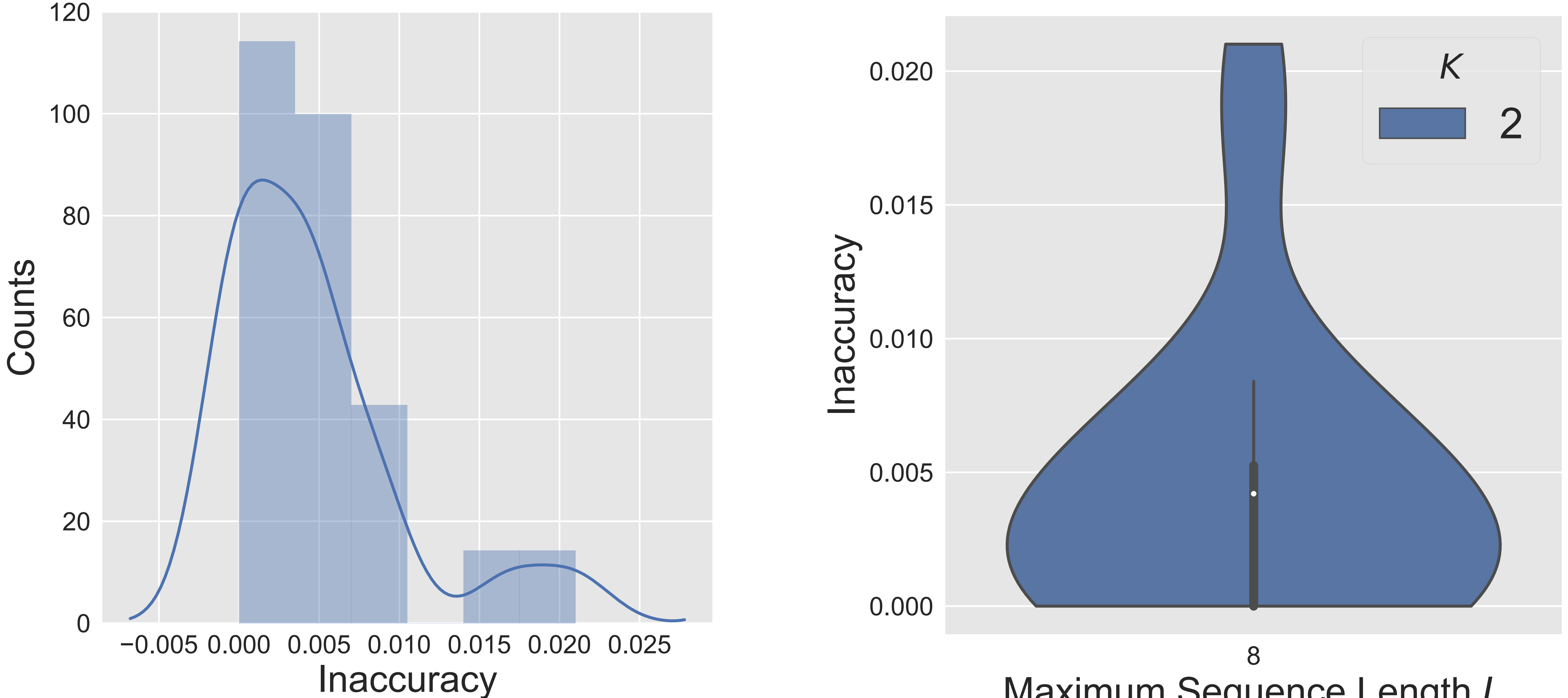
Project feature vectors based on **PCA**

Label feature vectors based on **noise**

Train a **soft-margin**, **linear** support vector machine (**SVM**)

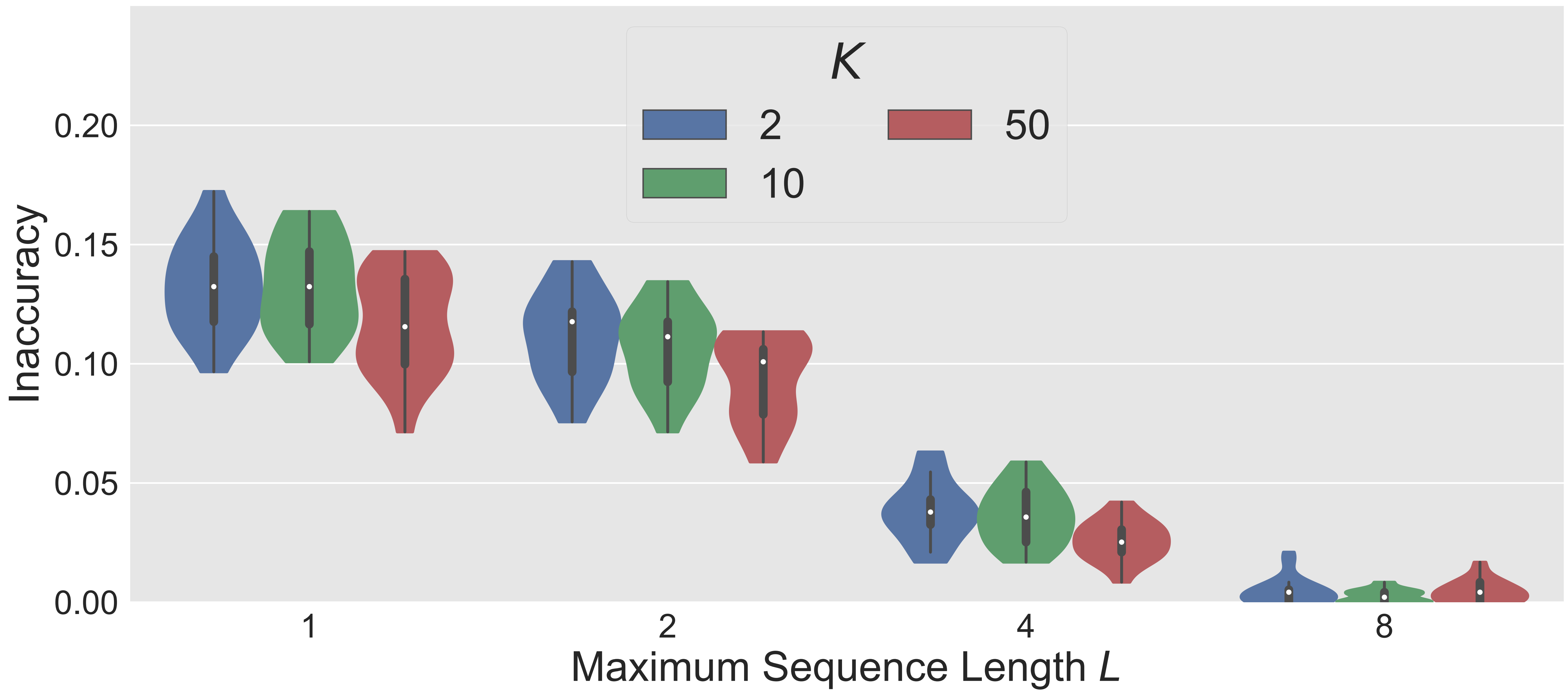
Using cross-validation, we find the SVM has reasonably high accuracy.

SVM is fairly accurate -
lowest accuracy is ~98%



20-fold shuffle-split cross-validation
(25% withheld for testing)

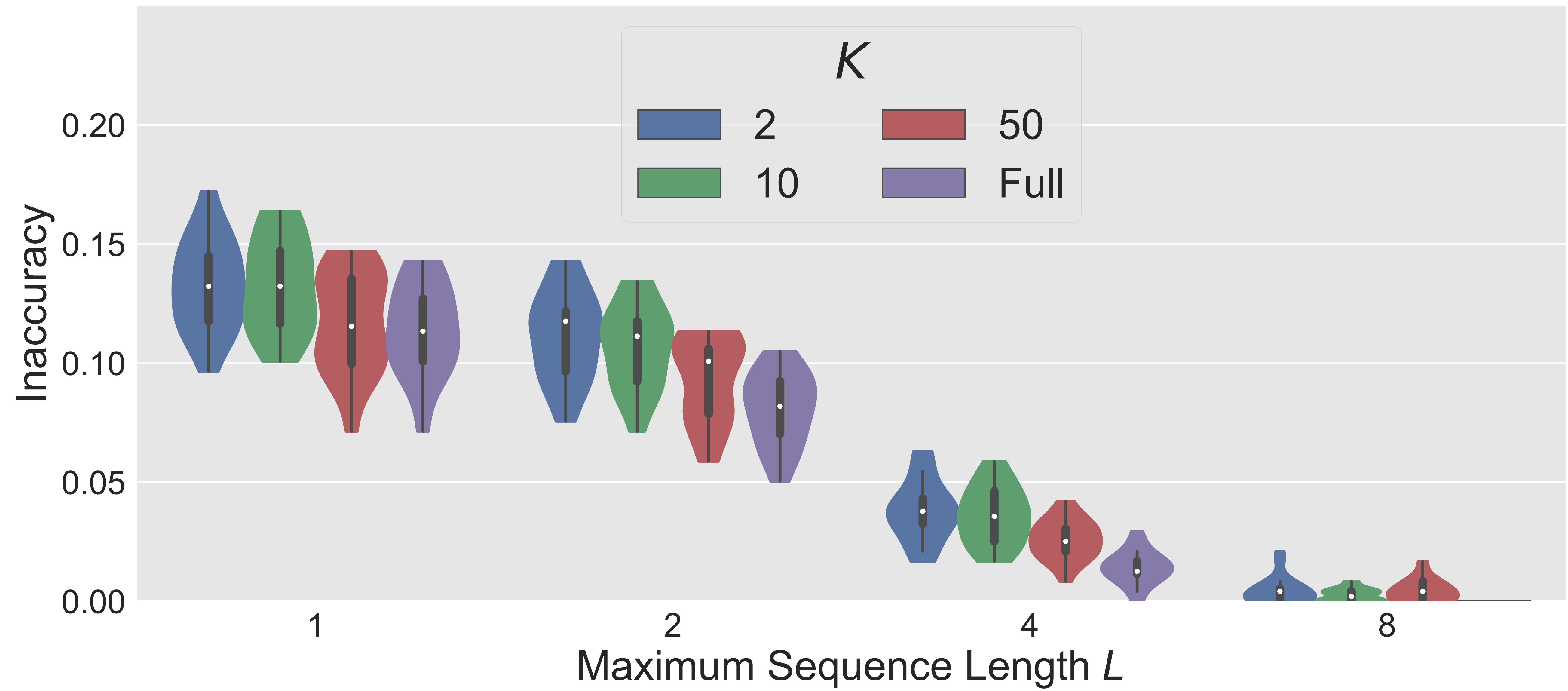
The accuracy of the SVM is affected by the number of components and maximum sequence length.



Noise Type	Number of Feature Vectors	Number of Noise Strengths
Coherent Error	450	9
Depolarization	450	9
No Noise	50	1

20-fold shuffle-split cross-validation scheme used, with 25% of the data withheld for testing on each split. A “one-versus-one” multi-class classification scheme was used.

Accuracies obtained on PCA-projected data are comparable to accuracies on the full feature space.

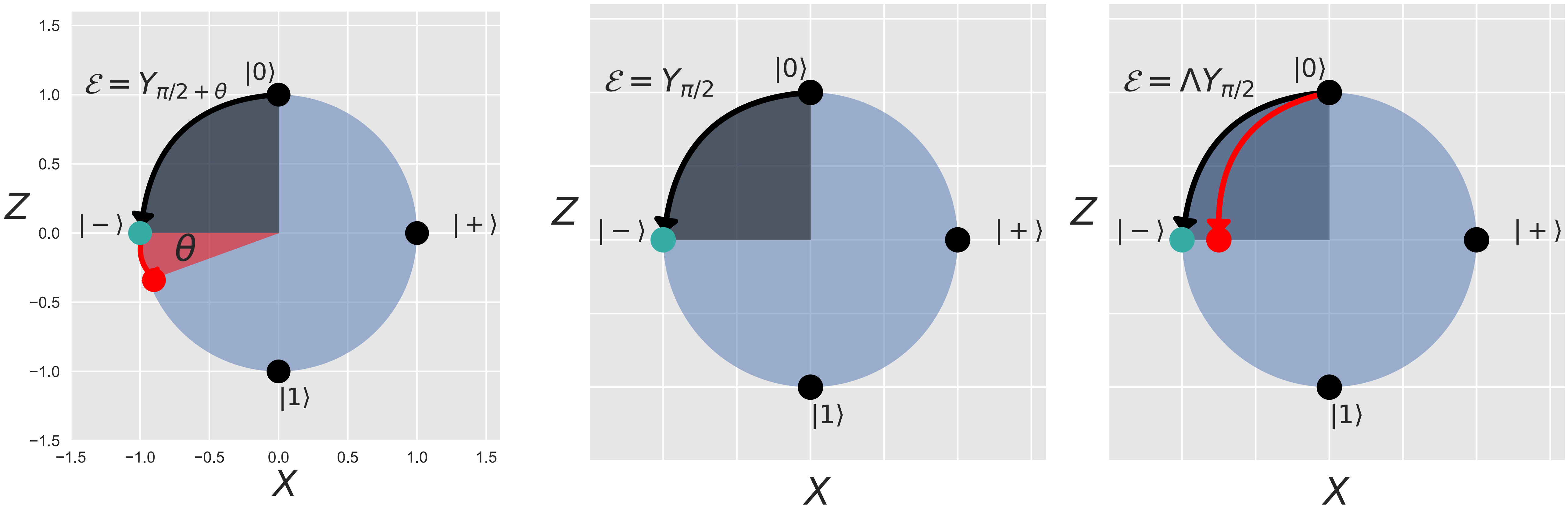


Noise Type	Number of Feature Vectors	Number of Noise Strengths
Coherent Error	450	9
Depolarization	450	9
No Noise	50	1

20-fold shuffle-split cross-validation scheme used, with 25% of the data withheld for testing on each split. A “one-versus-one” multi-class classification scheme was used.

Can a classifier learn the difference between arbitrary *stochastic* and arbitrary *coherent* noise?

Coherent Noise | **Ideal** | **Stochastic Noise**



Coherent Noise:

$$\dot{\rho} = -i[H_0, \rho] - i[e, \rho]$$

Ideal:

$$\dot{\rho} = -i[H_0, \rho]$$

Stochastic Noise:

$$\dot{\rho} = -i[H_0, \rho] + A\rho A^\dagger - \frac{1}{2}\{A^\dagger A, \rho\}$$

$$\mathcal{E} = V \circ G_0$$

$$VV^T = I$$

$$\mathcal{E} = G_0$$

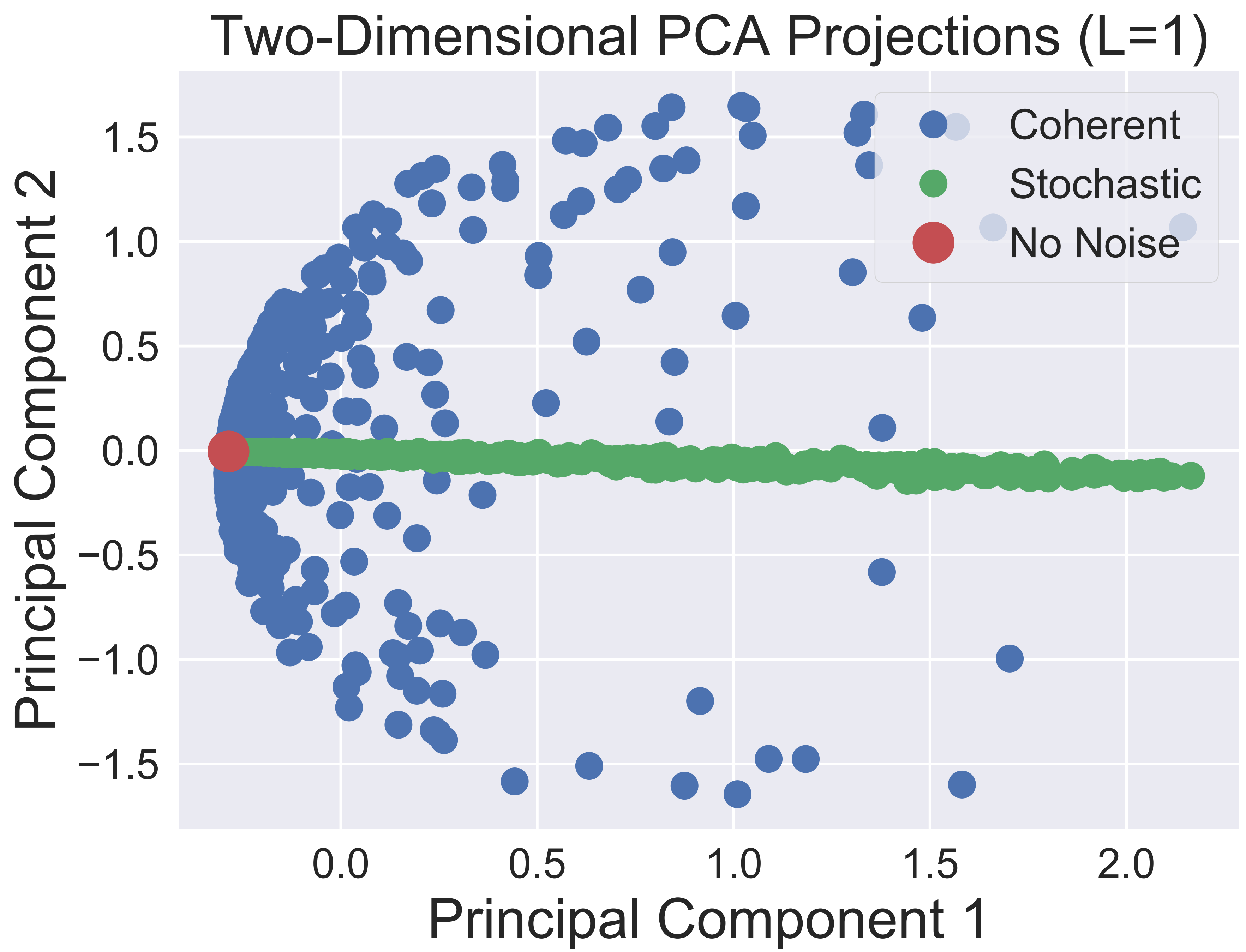
$$\mathcal{E} = \Lambda \circ G_0$$

$$\Lambda\Lambda^T \neq I$$

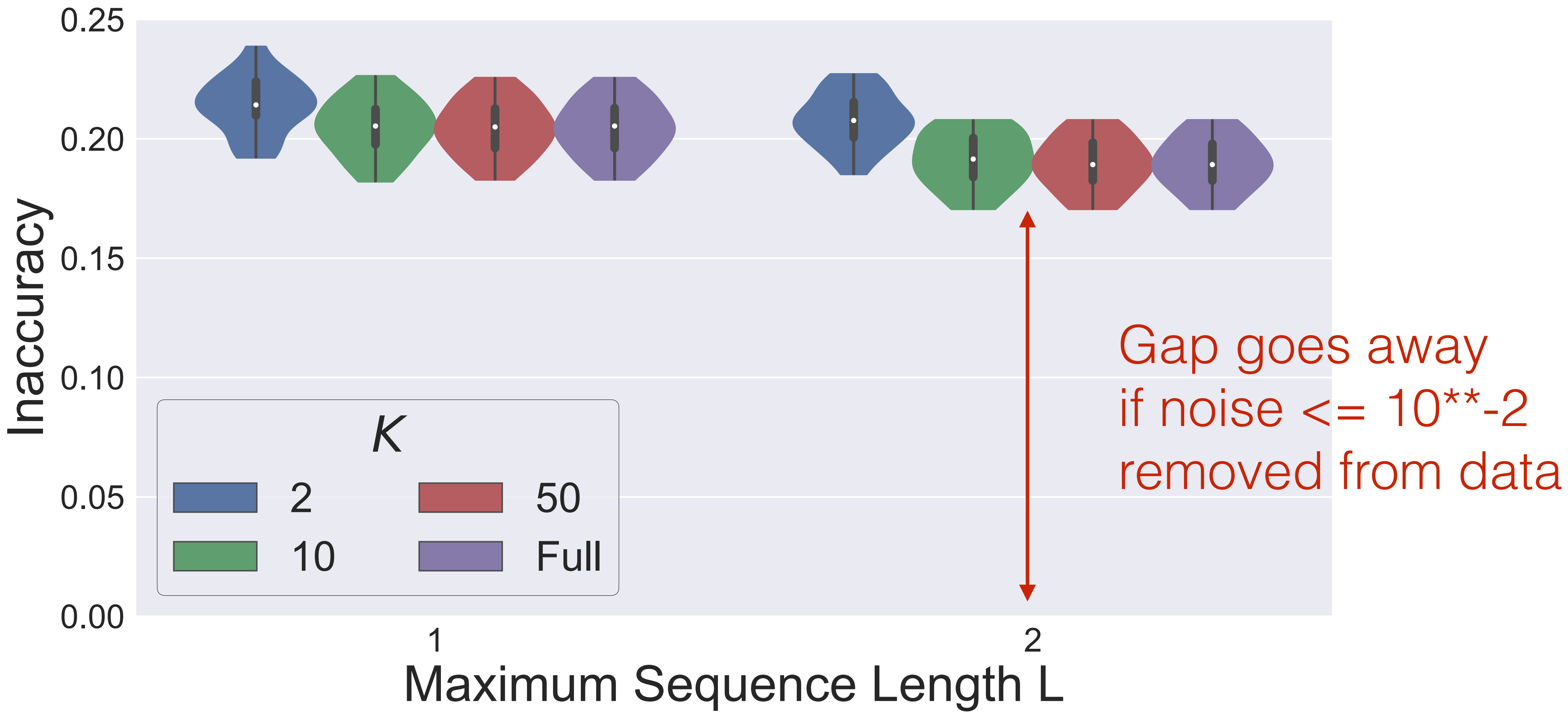
Classification in a 2-dimensional subspace is harder, due to structure of PCA-projected feature vectors.

"Radio dish" type structure

Linear classifier infeasible with only 2 PCA components



Preliminary results indicate a linear, soft-margin SVM can classify these two noise types in higher dimensions.

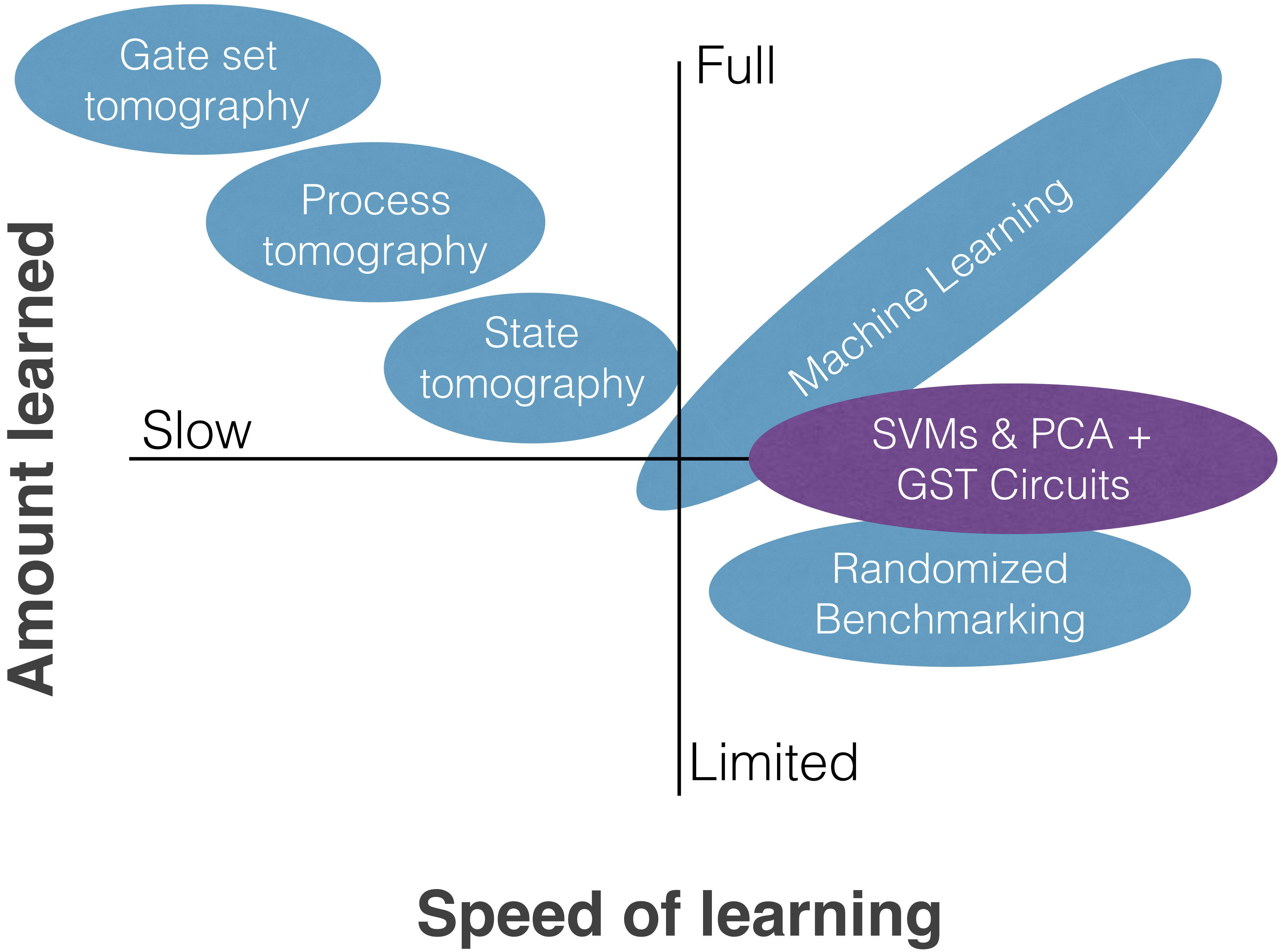


For each L:

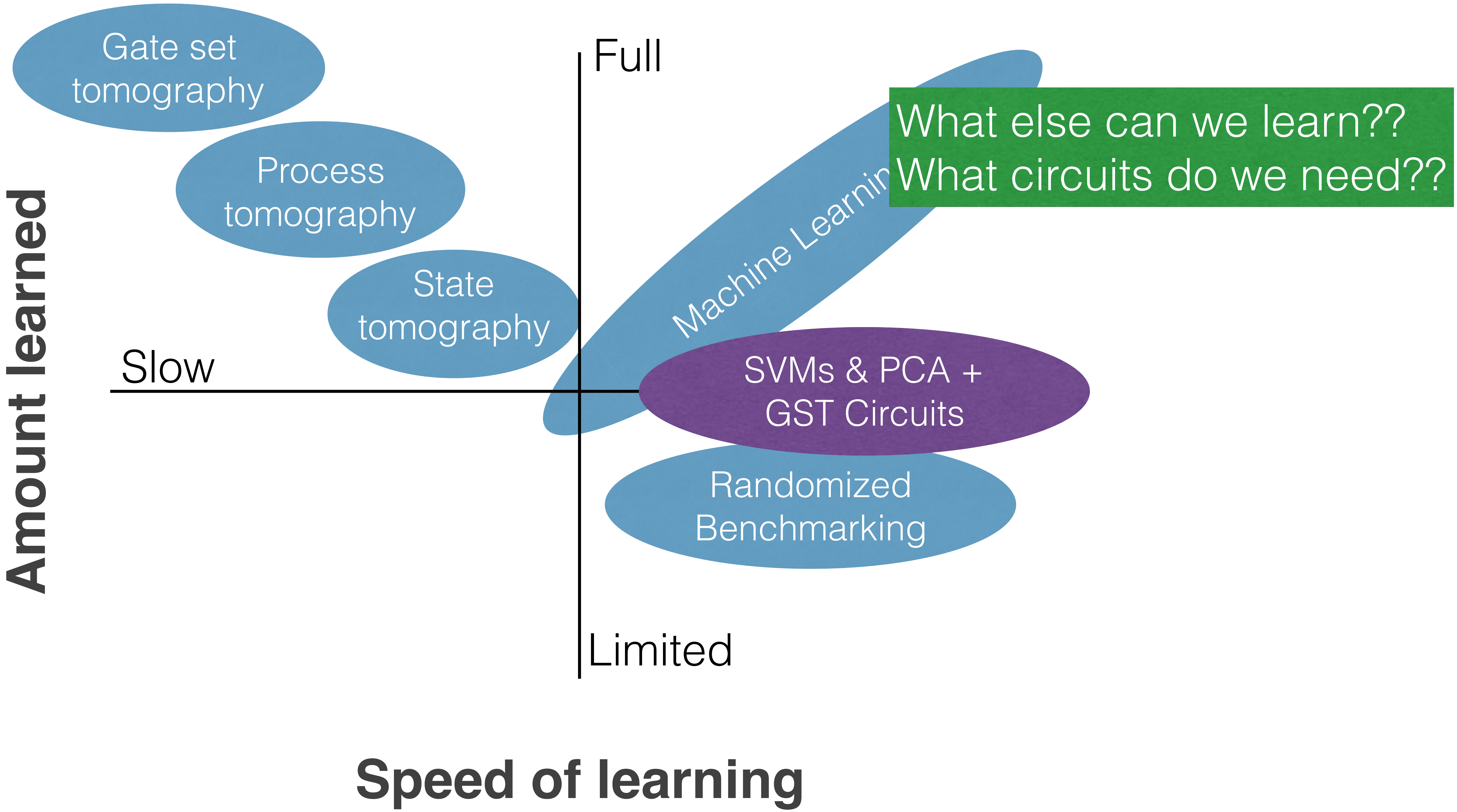
- 10 values of noise strength in [10**-4, 10**-1]
- 260 random instances

20-fold shuffle-split cross-validation scheme used,
with 25% of the data withheld for testing on each split.
A “one-verus-one” multi-class classification scheme was used.

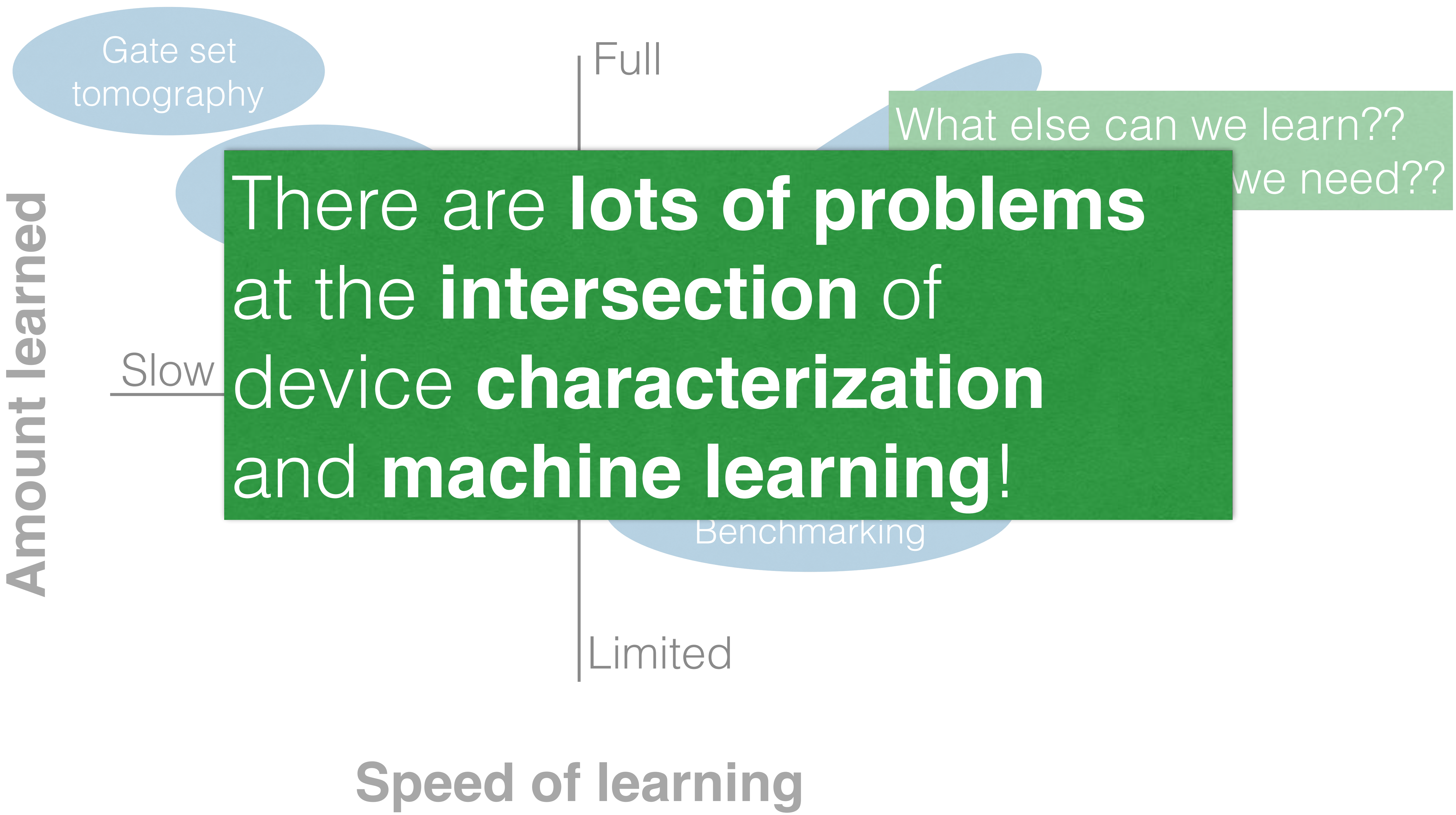
Support vector machines and PCA can analyze GST circuits and learn about noise with high accuracy.



Support vector machines and PCA can analyze GST circuits and learn about noise with high accuracy.

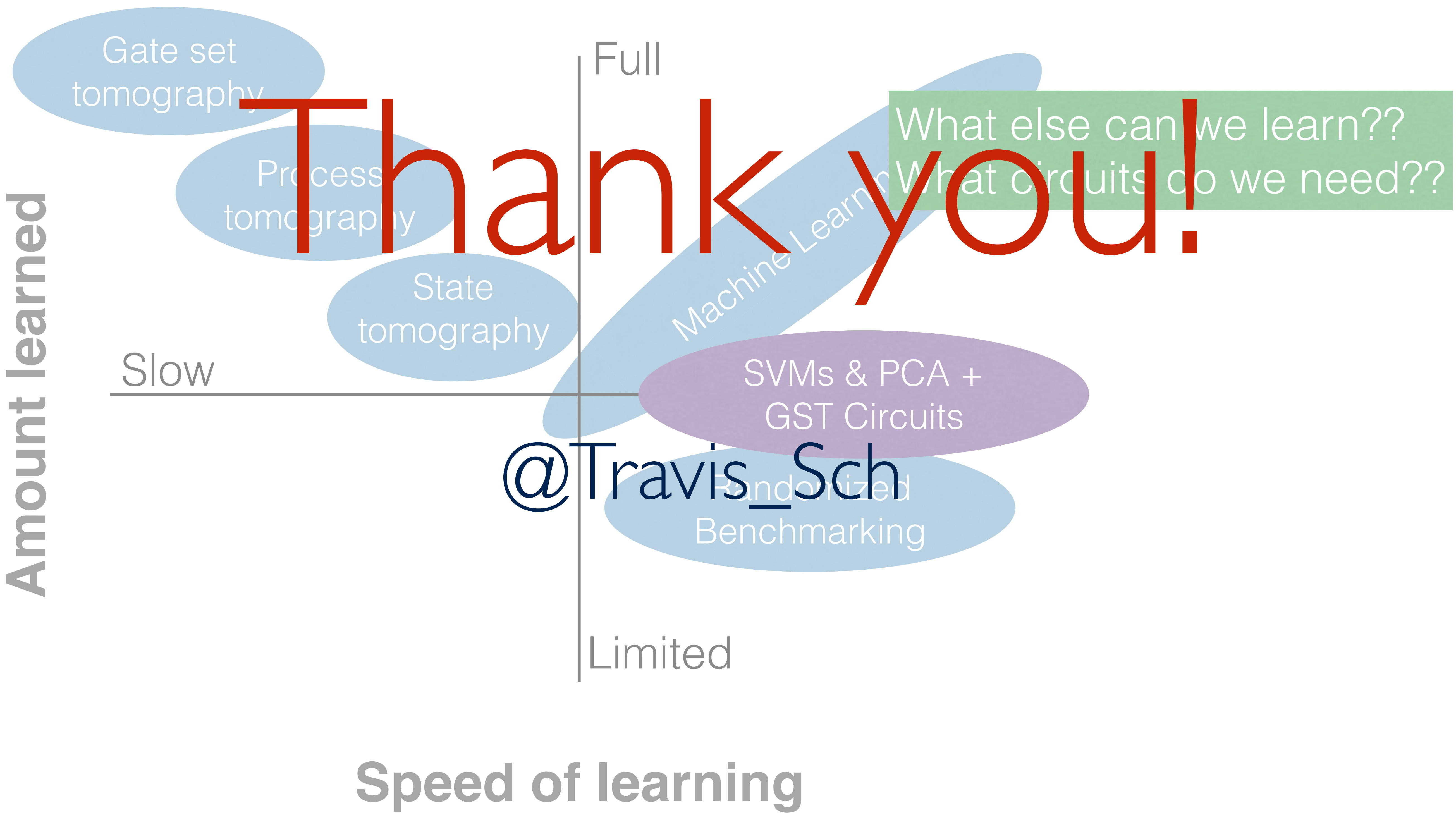


Support vector machines and PCA can analyze GST circuits and learn about noise with high accuracy.



There are **lots of problems** at the **intersection** of device **characterization** and **machine learning**!

Support vector machines and PCA can analyze GST circuits and learn about noise with high accuracy.



Thank you!

@Travis_Sch