Machine learning with PMU data

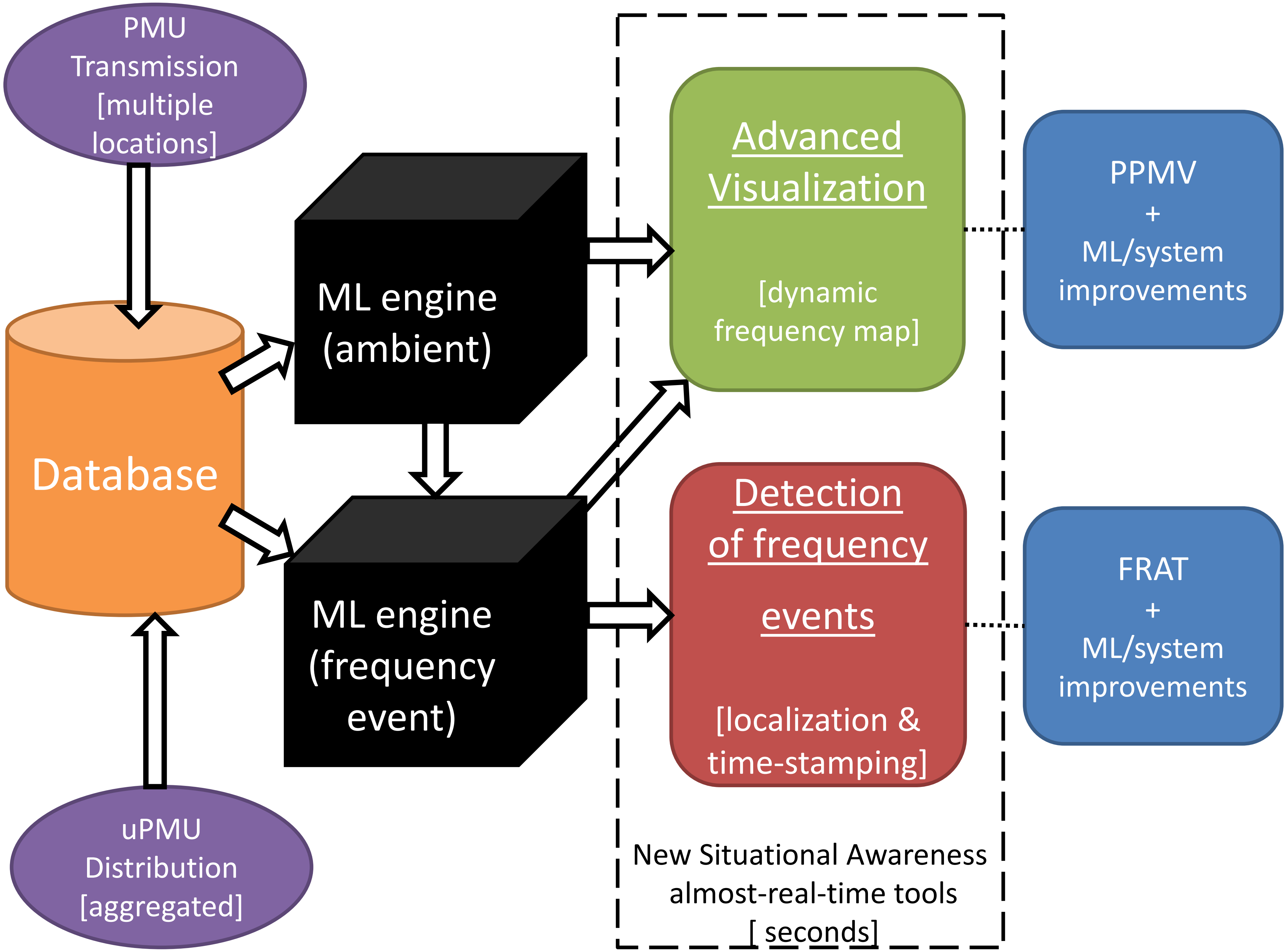
Columbia University:
Daniel Bienstock, Apurv Shukla, Mauro Escobar, Shuoguang Yang

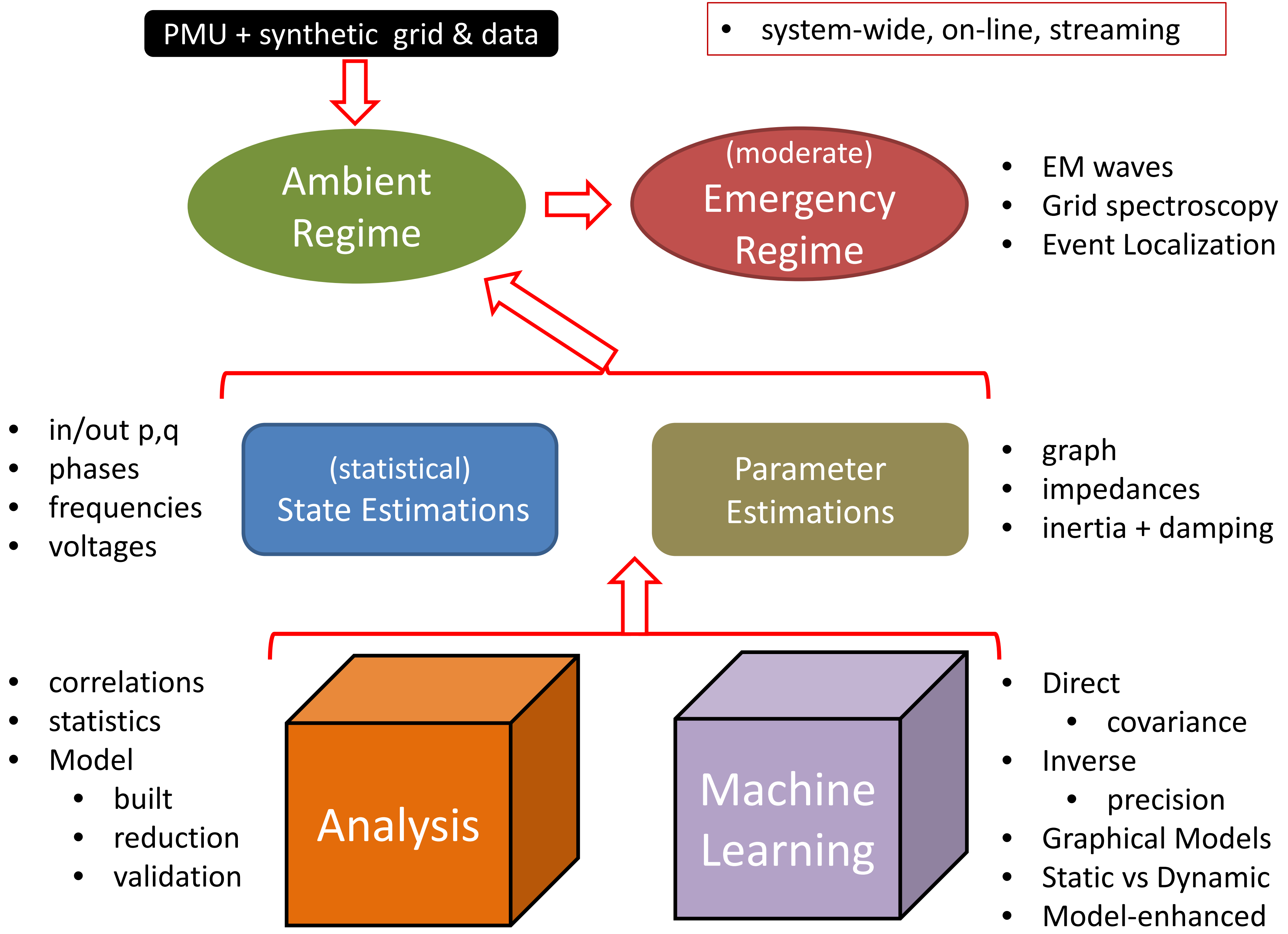
LANL:
Seyoung Yun, Andrey Lokhov, Deepjyoti Deka, Sidhant Misra, Marc Vuffray, Michael Chertkov

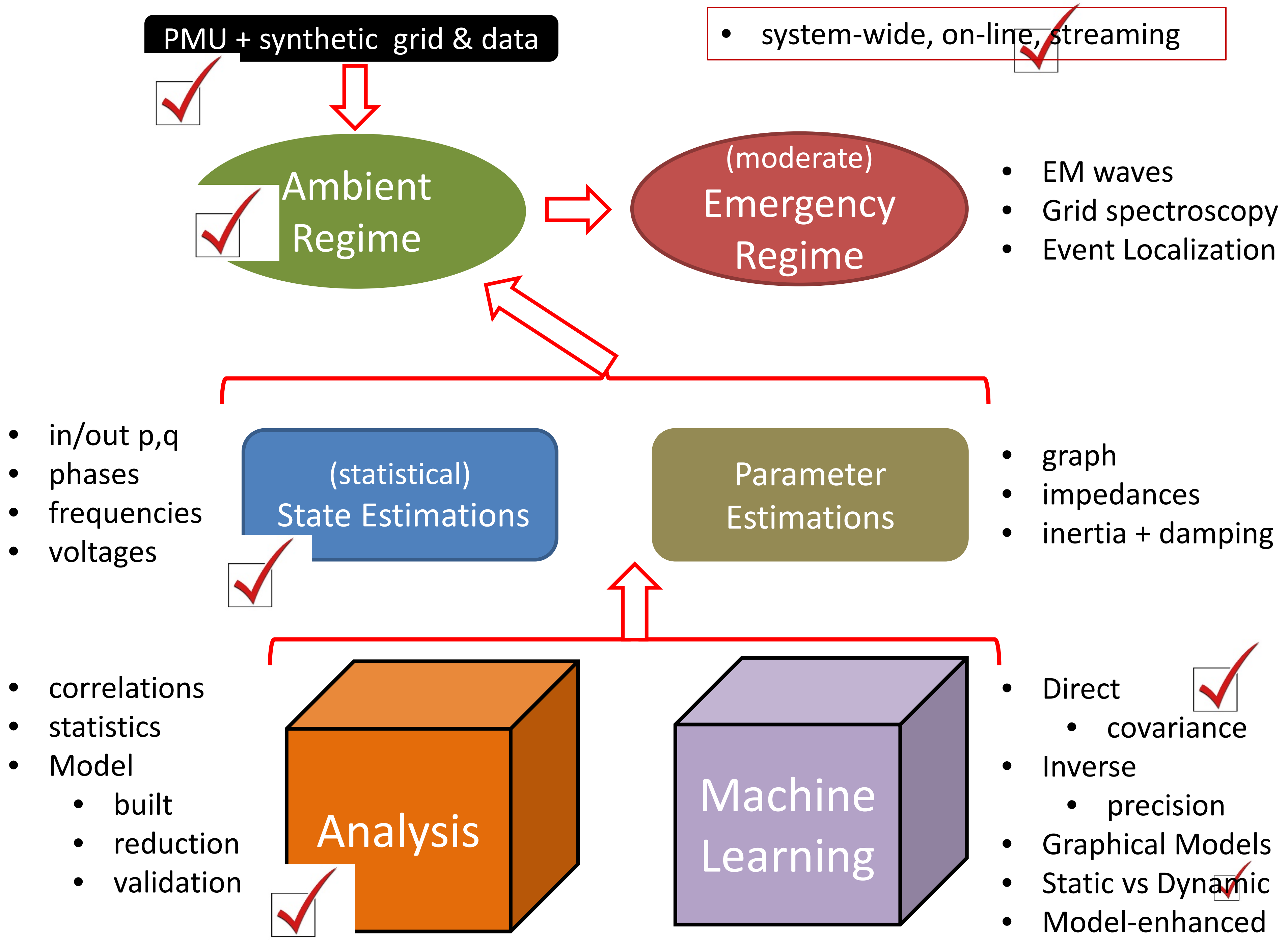
GMLC, Cat 2

Advanced Machine Learning for Synchro-phasor Technology

- LANL
- PNNL
- LBNL
- Columbia U




PMU + synthetic grid & data
• system-wide, on-line, streaming
Ambient Regime
(moderate) Emergency Regime
• EM waves
• Grid spectroscopy
• Event Localization
• in/out p,q
• phases
• frequencies
• voltages
(statistical) State Estimations
Parameter Estimations
• graph
• impedances
• inertia + damping
• correlations
• statistics
• Model
• built
• reduction
• validation
Analysis
Machine Learning
• Direct
• covariance
• Inverse
• precision
• Graphical Models
• Static vs Dynamic
• Model-enhanced


PMU + synthetic grid & data
• system-wide, on-line, streaming
Ambient Regime
(moderate) Emergency Regime
• EM waves
• Grid spectroscopy
• Event Localization
• in/out p,q
• phases
• frequencies
• voltages
(statistical) State Estimations
Parameter Estimations
• graph
• impedances
• inertia + damping
• correlations
• statistics
• Model
• built
• reduction
• validation
Analysis
Machine Learning
• Direct
• covariance
• Inverse
• precision
• Graphical Models
• Static vs Dynamic
• Model-enhanced

Focus: learning structure from PMU data

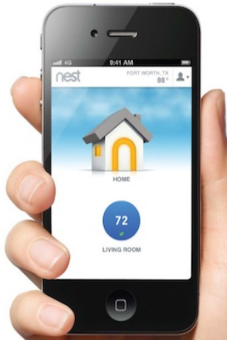
- Normal regime
- Adversarial regime

Power Grid: Structure Learning

- Uses:
  - Real time control
  - Failure Identification
  - Optimizing flows

- Challenge:
  - Limited real-time breakers
  - Brute Force inefficient

- Solution
  - Smart meters: PMUs, micro-PMUs, IoT
  - Big Data: High fidelity measurements

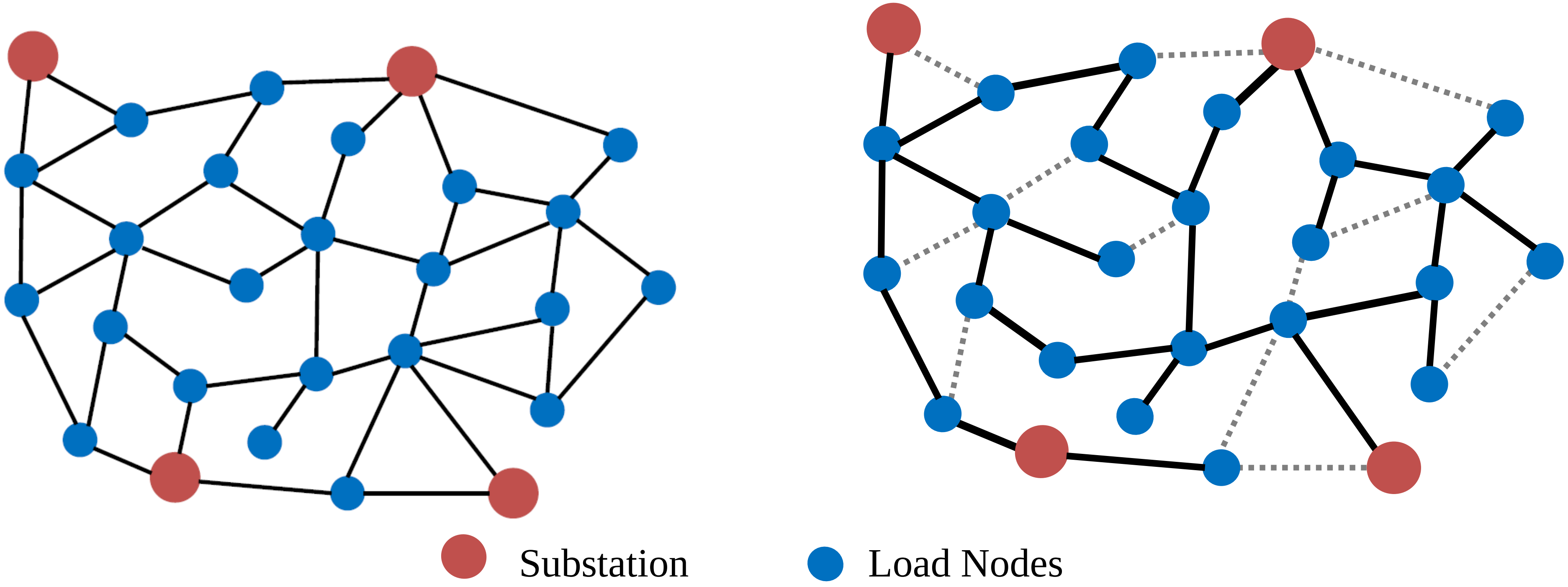


Grid ‘*Operational*’ Structure

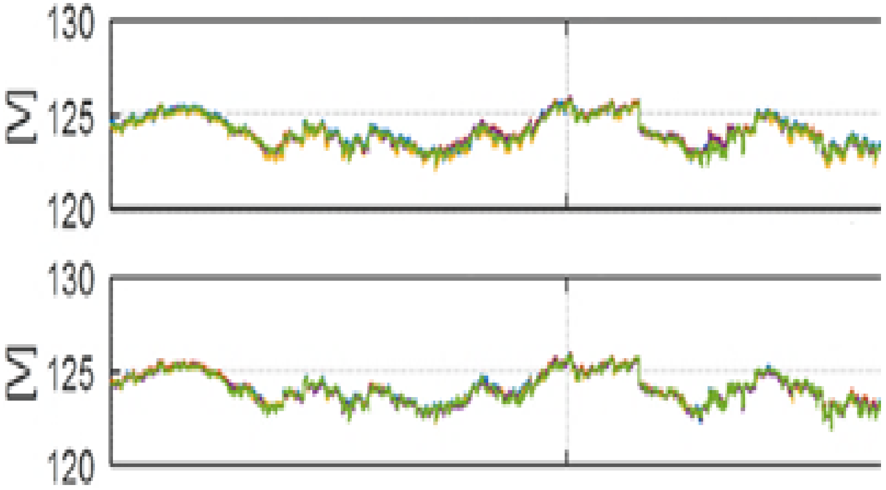
- Underlying Loopy network
- Switches/Relays decide structure

Learning Problem:

- Estimate Configuration of Switches/Relays



Learning Methodology



Probabilistic Graphical Model *for the nodal voltages*

- Static Regime: *Power Flow Equations*
- Dynamic Regime: *Swing Equations*

Adversarial regime (ongoing work)

- A fictional setting: an adversary controls some (few) grid resources
- Some overt physical changes have been made by the adversary
- Additionally the adversary alters some sensor data
- Adversary is intelligent and has computed these actions in advance

1. Can we even tell that something has happened?
2. And quickly?
3. Do we want to use a lot of data?
4. Or maybe not (data could be corrupted)?
5. Methodology: robust inverse optimization

Focus: learning covariance matrices for PMU data

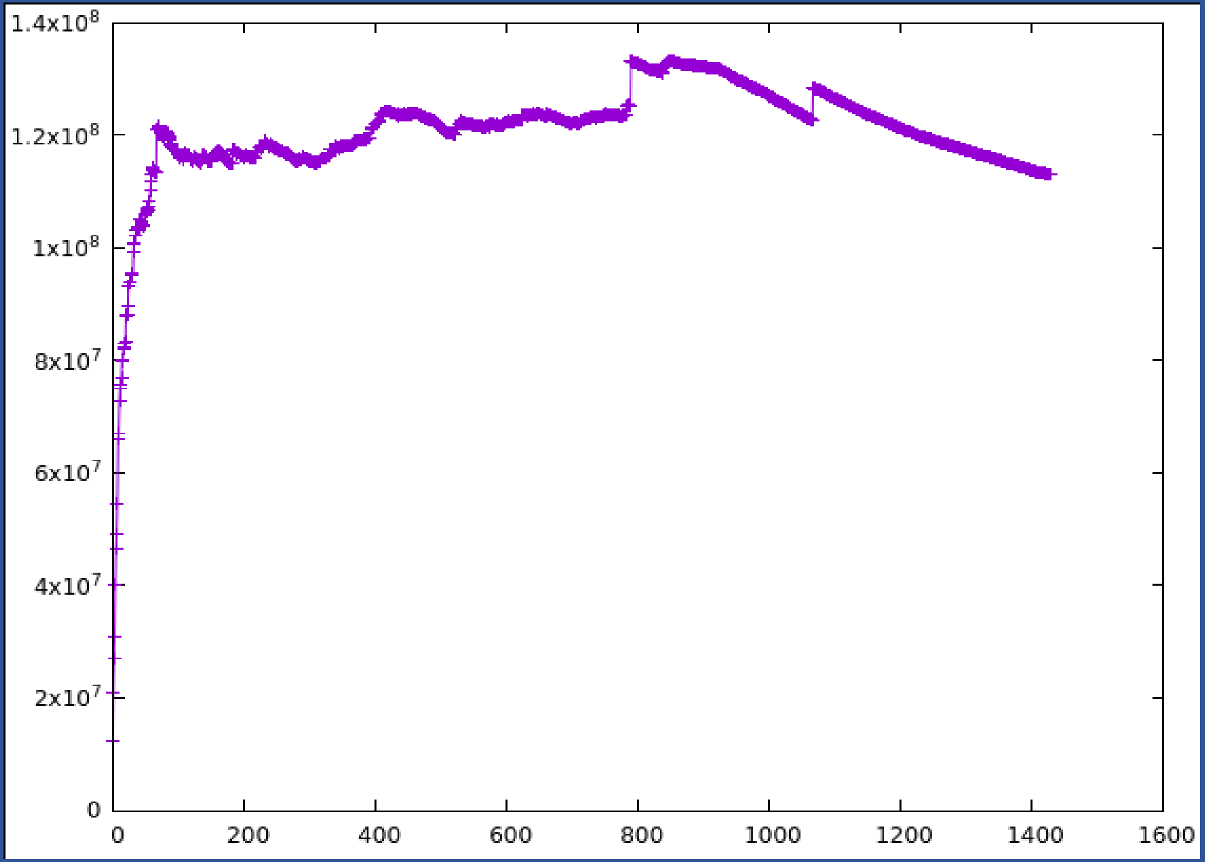
- Covariance analysis is a major tool in risk planning
- Correlation patterns important: what is revealed?
- Counterpoint: principal components analysis (PCA) is an important perspective
- A classical outlook: the factors revealed by PCA can be used in control algorithms, even though there is no overt "understanding" of the factors

Experiment using data from an industry partner

- Approx. 200 PMUs
- Consider voltage magnitude output for one day
- Compute covariance matrix since start of day until each minute of the day

First day: “Frobenius norm” = sum of eigenvalues

A measure of total volatility

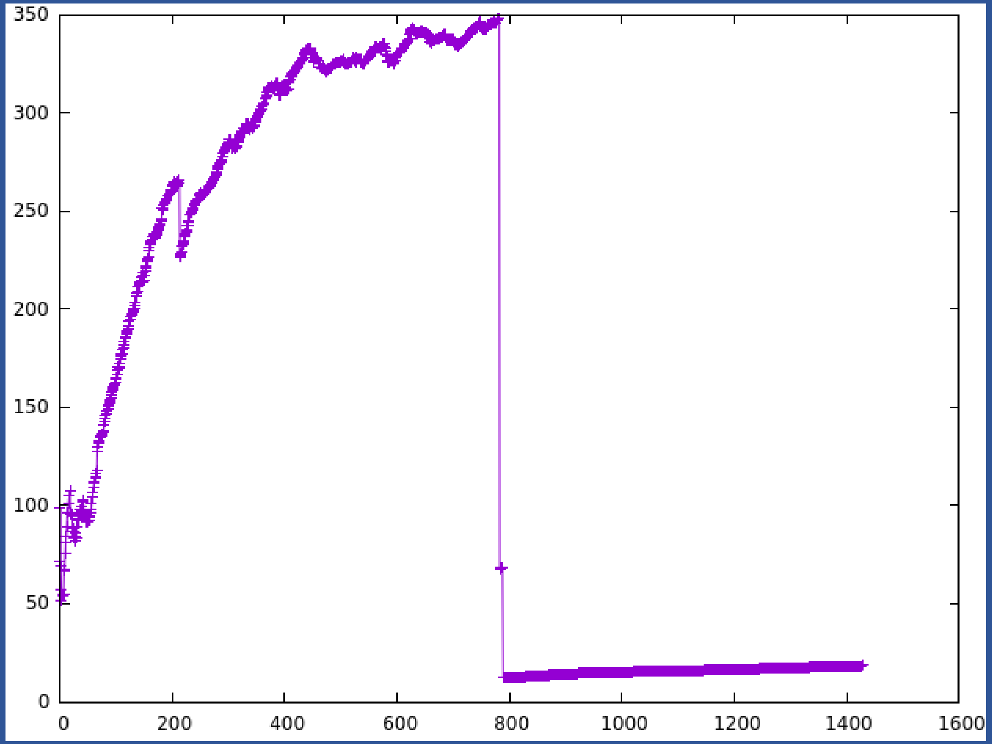


Volatility rises rapidly in the early morning and then stabilizes

What is that jump around minute 800 ~ 1:45 PM ?

Ratio of largest to second largest eigenvalue

A measure of ***lack*** of diversity in volatility



Ratio rises in the morning, stabilizes and then …

What is that jump around minute 800 ~ 1:45 PM ?

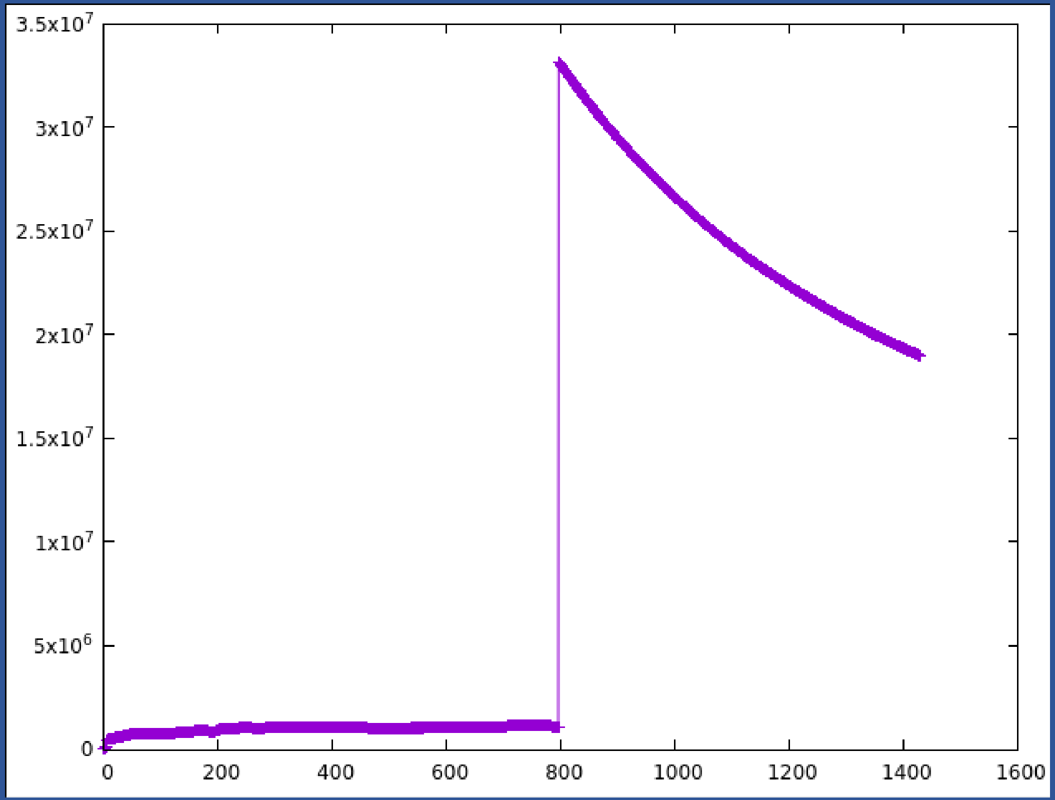
LET’S REPEAT THE EXPERIMENT

But using a different day

A day with some voltage oscillations

Second day: “Frobenius norm” = sum of eigenvalues

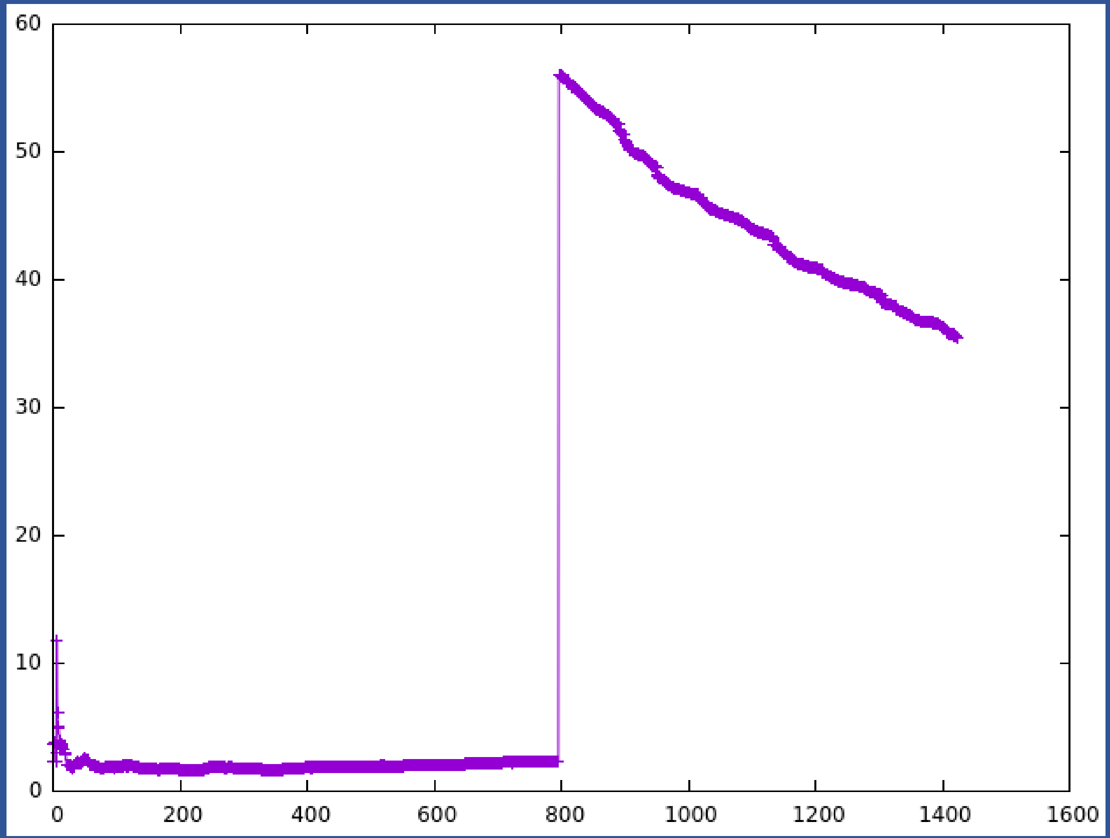
A measure of total volatility



Volatility low until minute 800, then a large jump

Ratio of largest to second largest eigenvalue

A measure of ***lack*** of diversity in volatility



Ratio close to 1.0 until minute 800 …

Jump suggests a decrease in volatility diversity

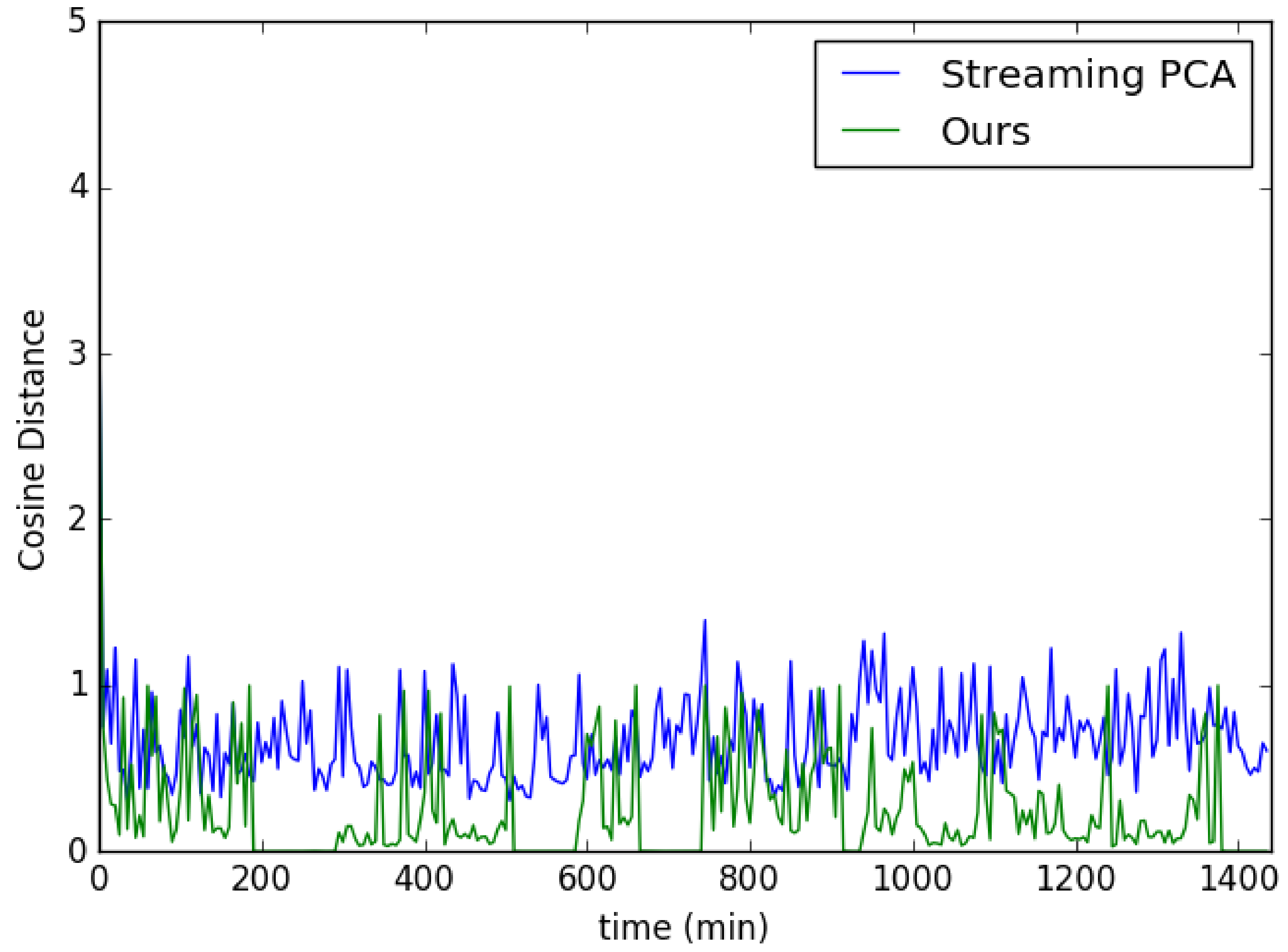
FOCUS: fast, online estimation of covariance matrices from streaming data

- Exact covariance computation is slow and requires many samples to converge

- Can we quickly obtain an approximate estimate of the covariance matrix – actually what we care about is PCA

- Methodology: “sketching” algorithms

- Goal is to estimate the top eigenvalues/eigenvectors quickly



Streaming PCA
Ours
Cosine Distance
time (min)
0
1
2
3
4
5
0
200
400
600
800
1000
1200
1400