machine learning conference

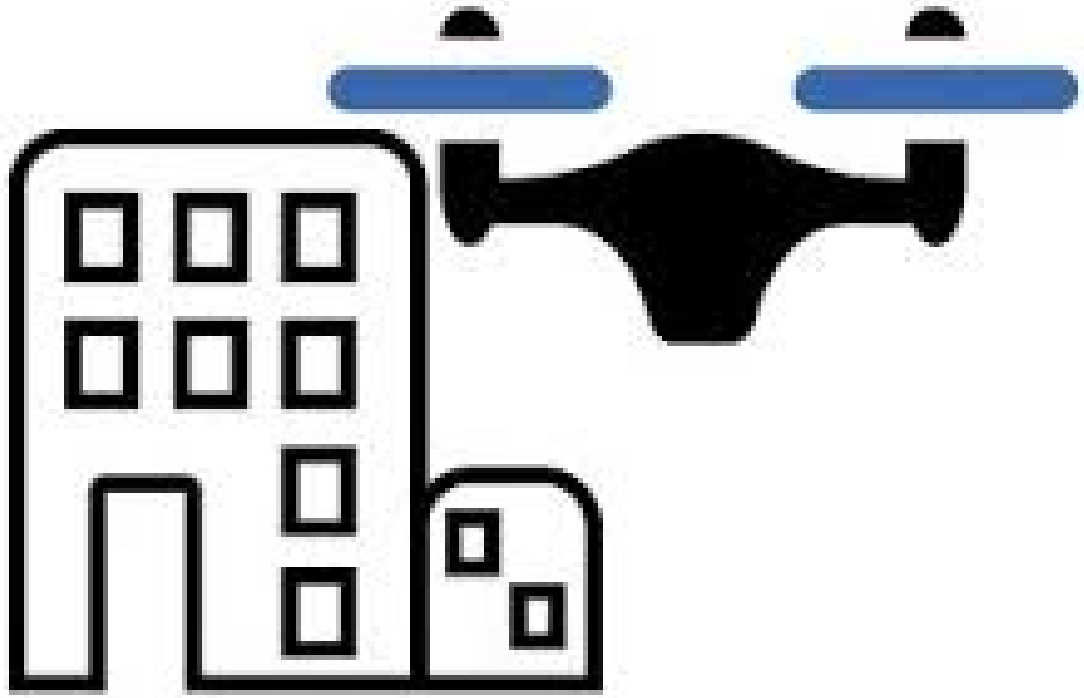
Semi-supervised Learning on Data Streams

Shiva Kasiviswanathan, Amazon AI Lab

Joint work with Tal Wagner (MIT), Sudipto Guha (Amazon), Nina Mishra (Amazon)

Autonomous Drones

Autonomous Drones



Input: Video (sequence of images)

Input: Video (sequence of images)

Input: Video (sequence of images)

Input: Video (sequence of images)

Hot air balloon | Hot air balloon | Hot air balloon | Bridge | Bridge | UFO | UFO | UFO

Goal:

To build a streaming, real-time classification system that identifies obstacles and works with little labeled data and much unlabeled data

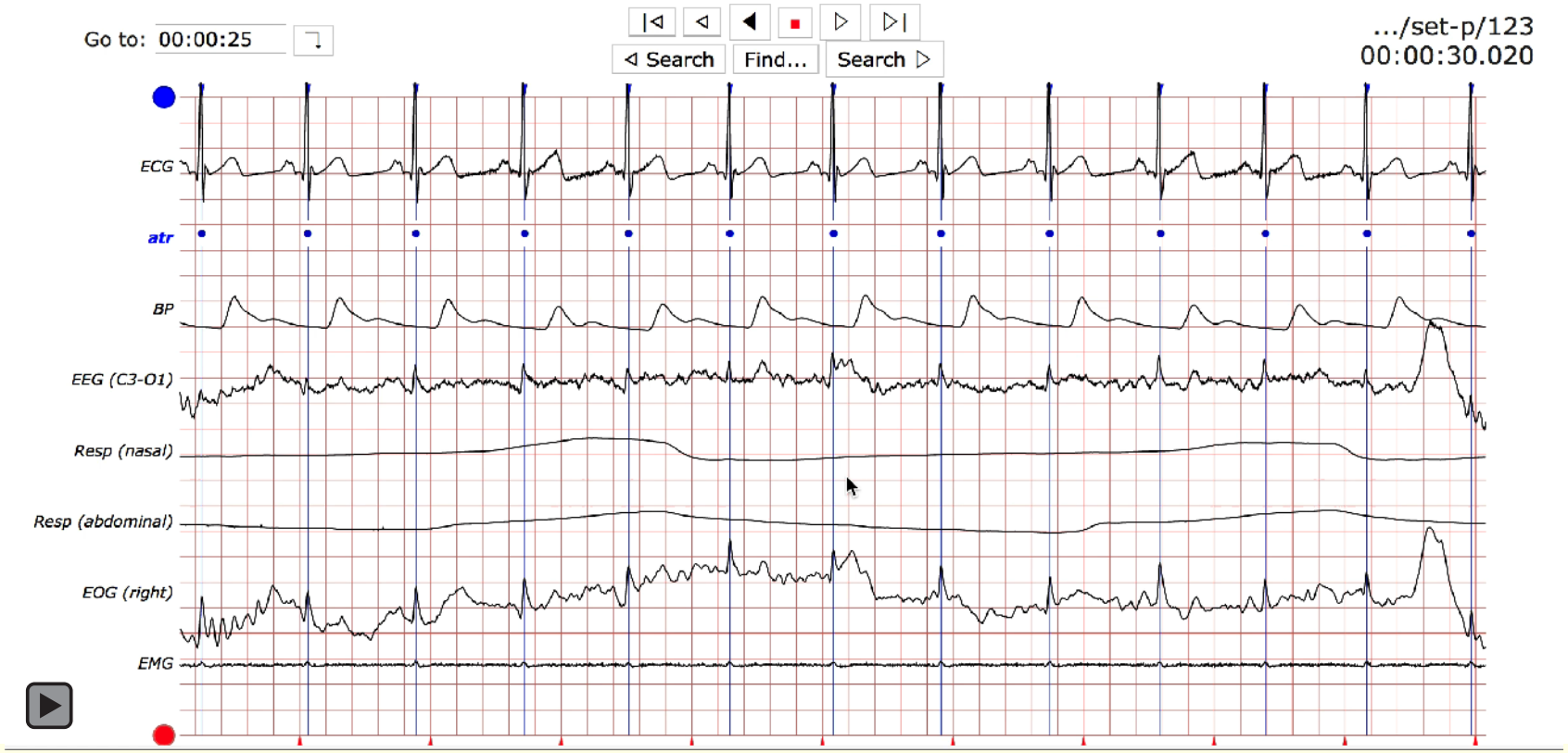
Input: Video (sequence of images)

Hot air balloon | Hot air balloon | Hot air balloon | Bridge | Bridge | UFO | UFO | UFO

Goal:

To build a streaming, real-time classification system that identifies obstacles and works with little labeled data and much unlabeled data

Input: ~~Video (sequence of images)~~ Multi-dimensional timeseries



General Statement:

To build a streaming, real-time classification system that works with little labeled data and much unlabeled data

Getting Labeled Data is “Hard”

Medical Domain
(wearable sensors)

Security Domain

(Semi)-autonomous Cars

Data Center

Getting Labeled Data is “Hard”

	Customer Usecases	Labeled data
Medical Domain (wearable sensors)	EEG, ECG analysis, Fall detection ….	Expensive
Security Domain	Denial of service attack, Phishing …..	Rare
(Semi)-autonomous Cars	Detecting construction zones, tunnels …….	Laborious
Data Center	Data center outages Power failures ….	Digital Exhaust

Getting Labeled Data

Hard for Amazon and its customers

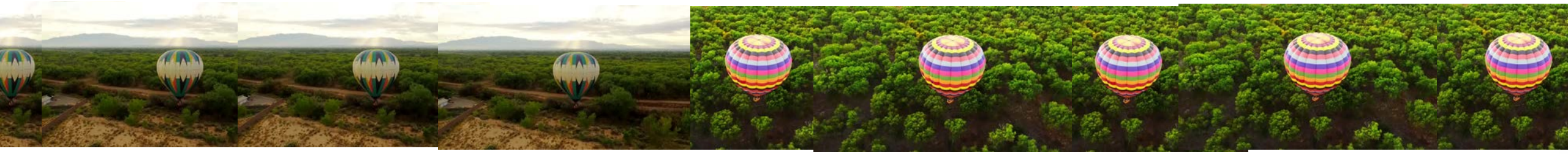
Getting Unlabeled Data

Significantly easier for Amazon and its customers

Label = hot air balloon

Semi-Supervised learning

A good classifier



Unlabeled data – Drone Footage

Real-Time Aspect

	Customer Usecases	Labeled data
Medical Domain (wearable sensors)	EEG, ECG analysis, Fall detection ….	Expensive
Security Domain	Denial of service attack, Phishing …..	Rare
(Semi)-autonomous Cars	Detecting construction zones, tunnels …….	Laborious
Data Center	Data center outages Power failures ….	Digital Exhaust

Our Solution:

Semi-supervised algorithm that operates on stream

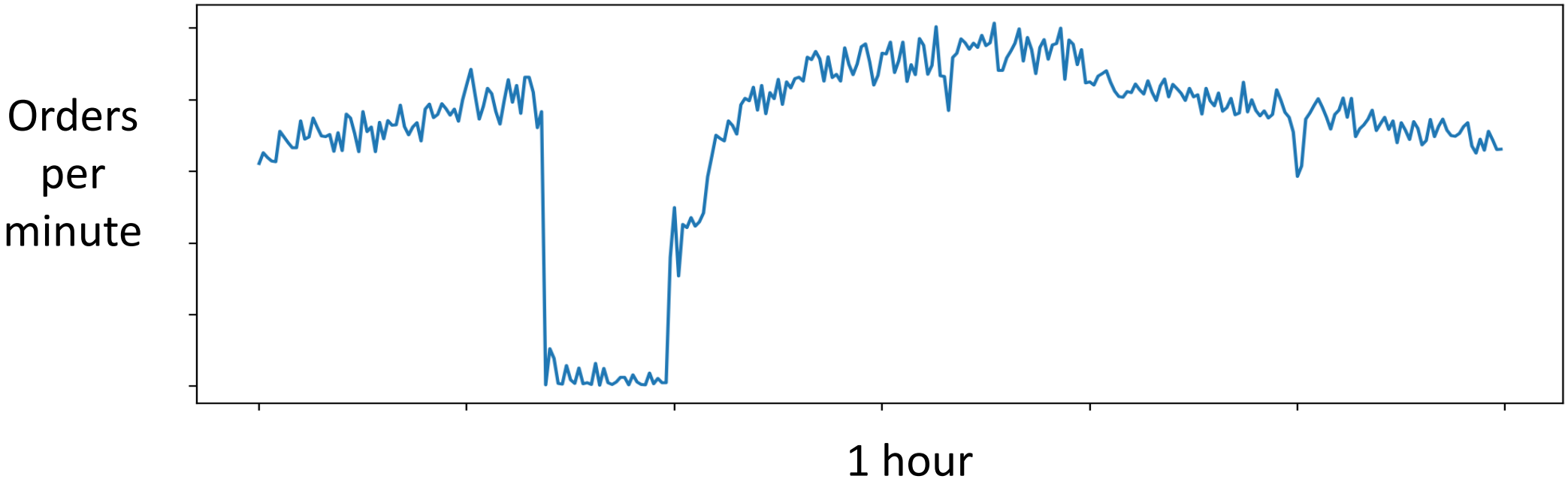
I. Works with little labeled data ✓

II. Real-time classification ✓

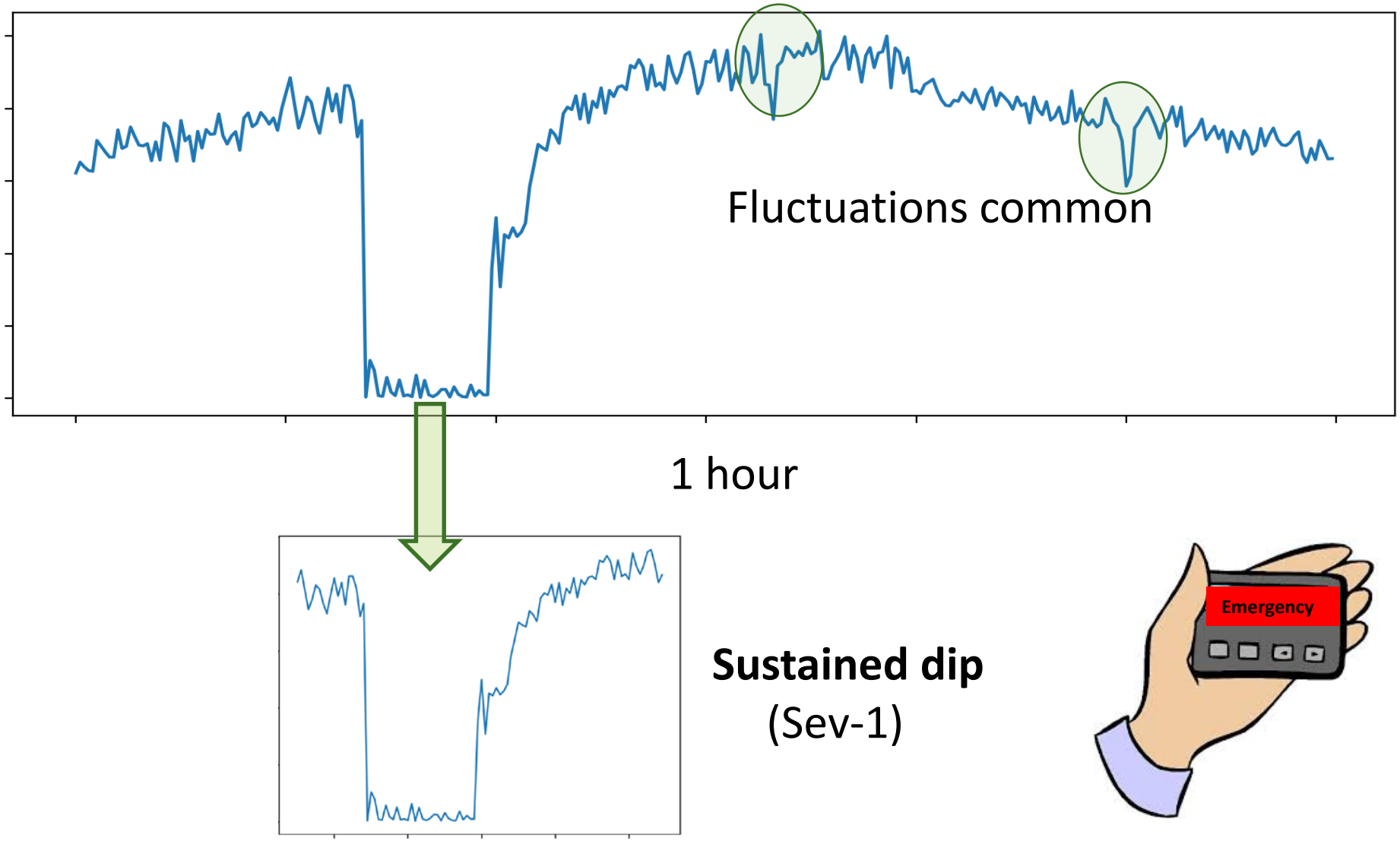
III. Computationally efficient ✓

IV. Adapts to new data patterns ✓

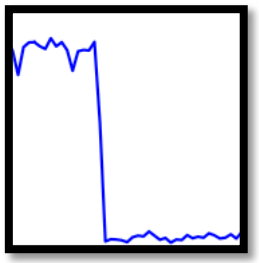
Amazon Orders



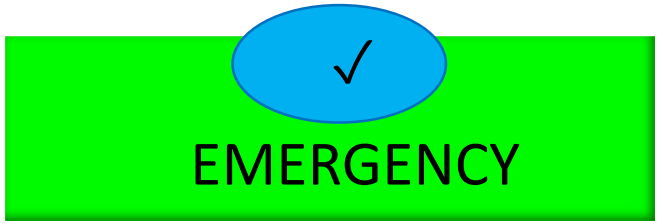
Amazon Orders

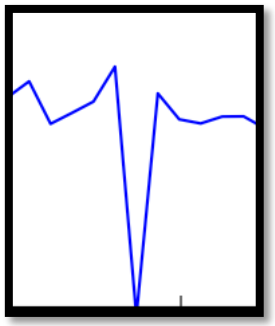


Sustained dip
(Sev-1)

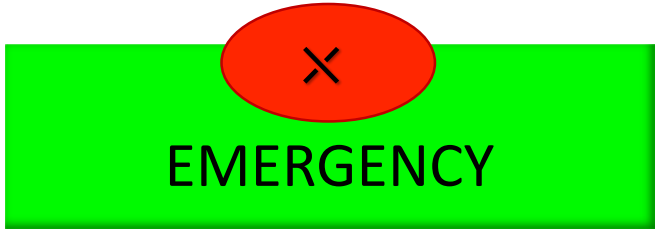


sustained dip



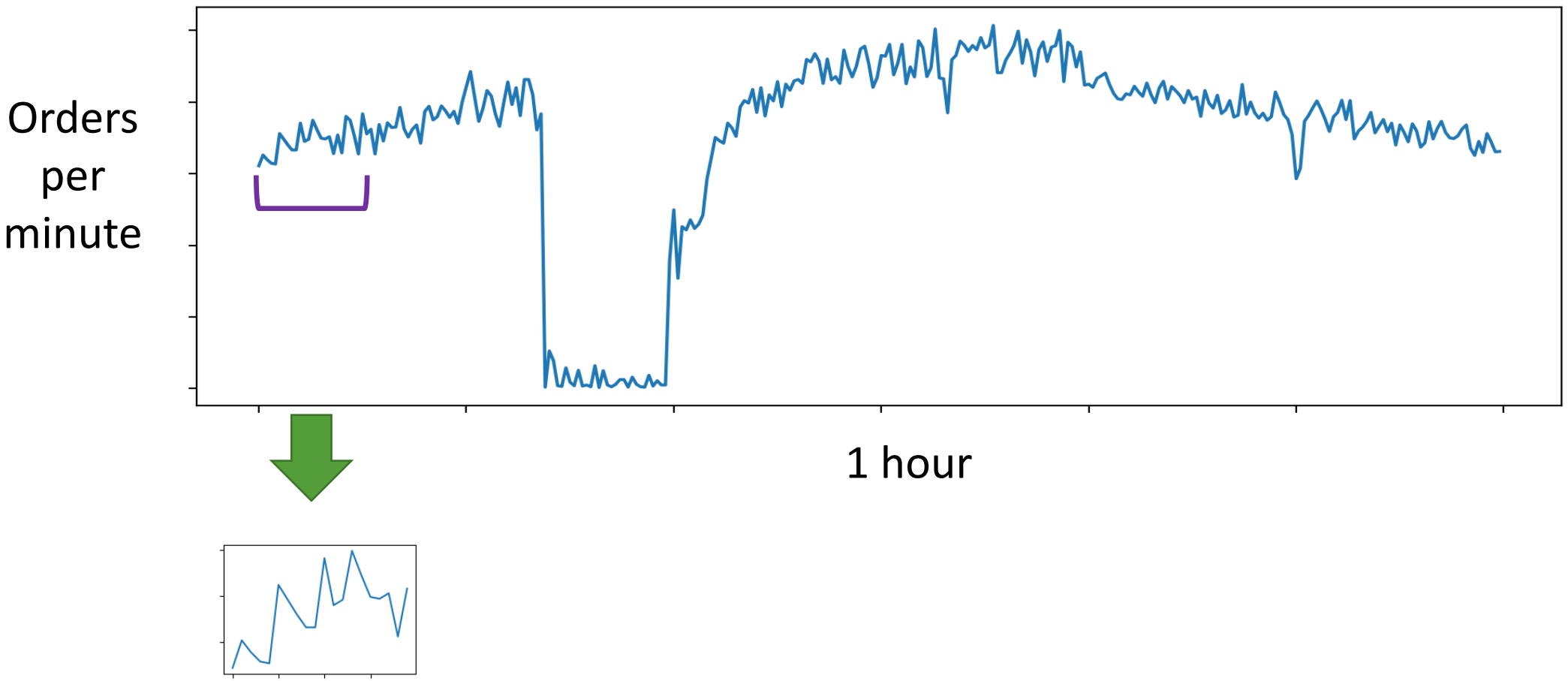


don’t bother

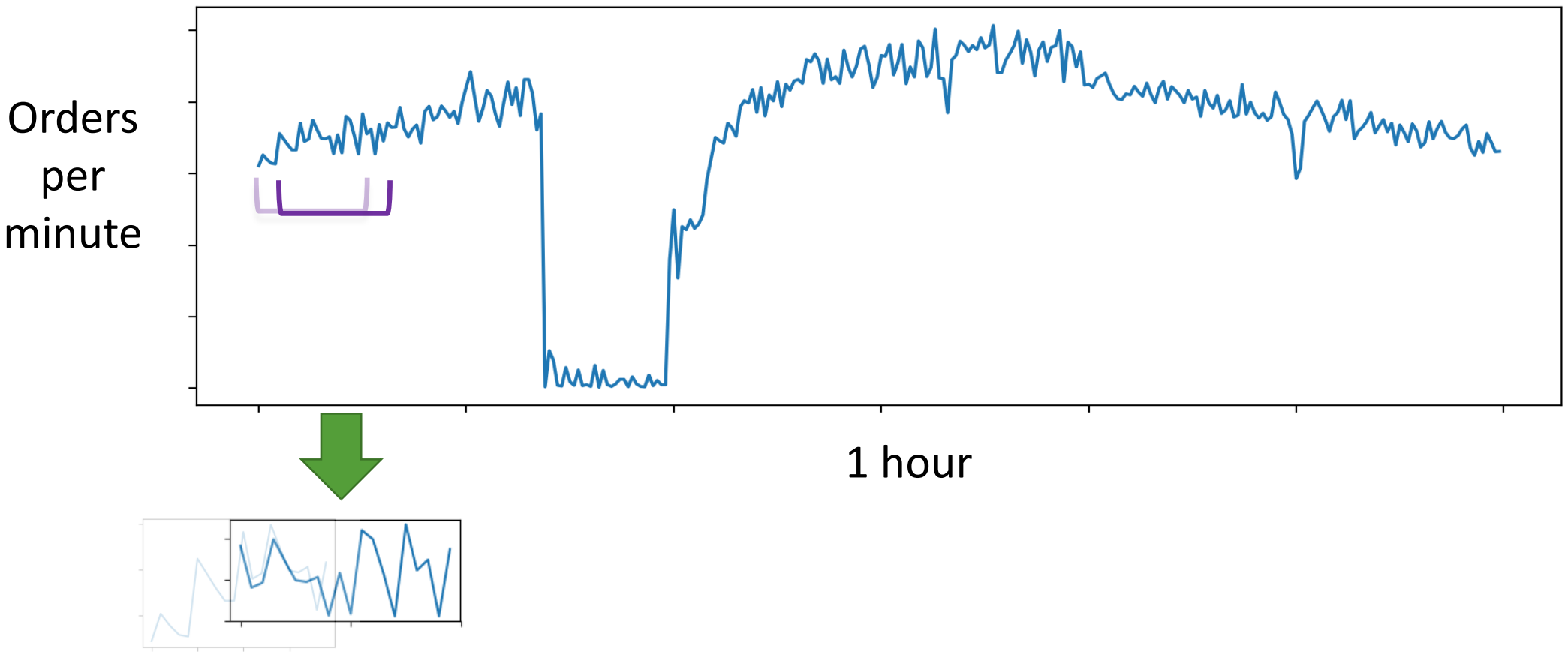


Can we incorporate user feedback
to make the system smarter over time?

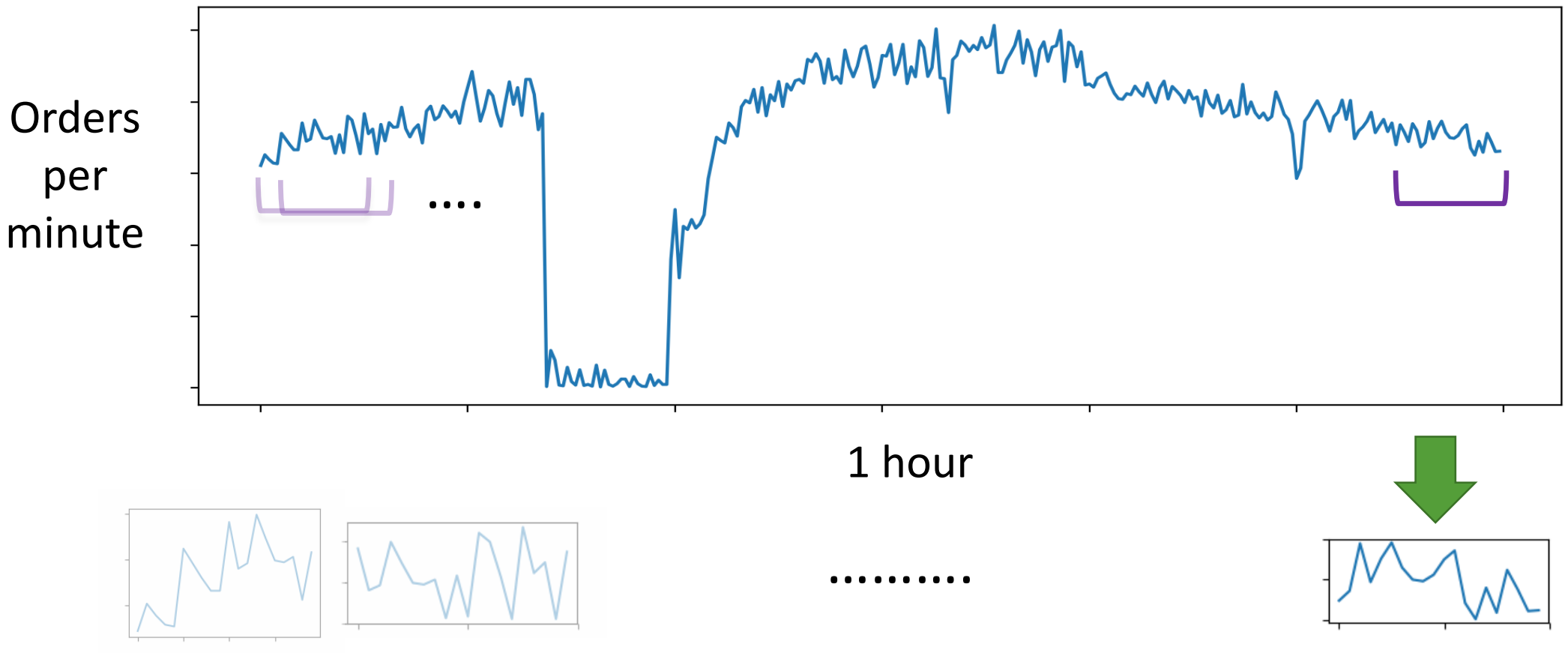
How to use semi-supervised learning to solve this problem?



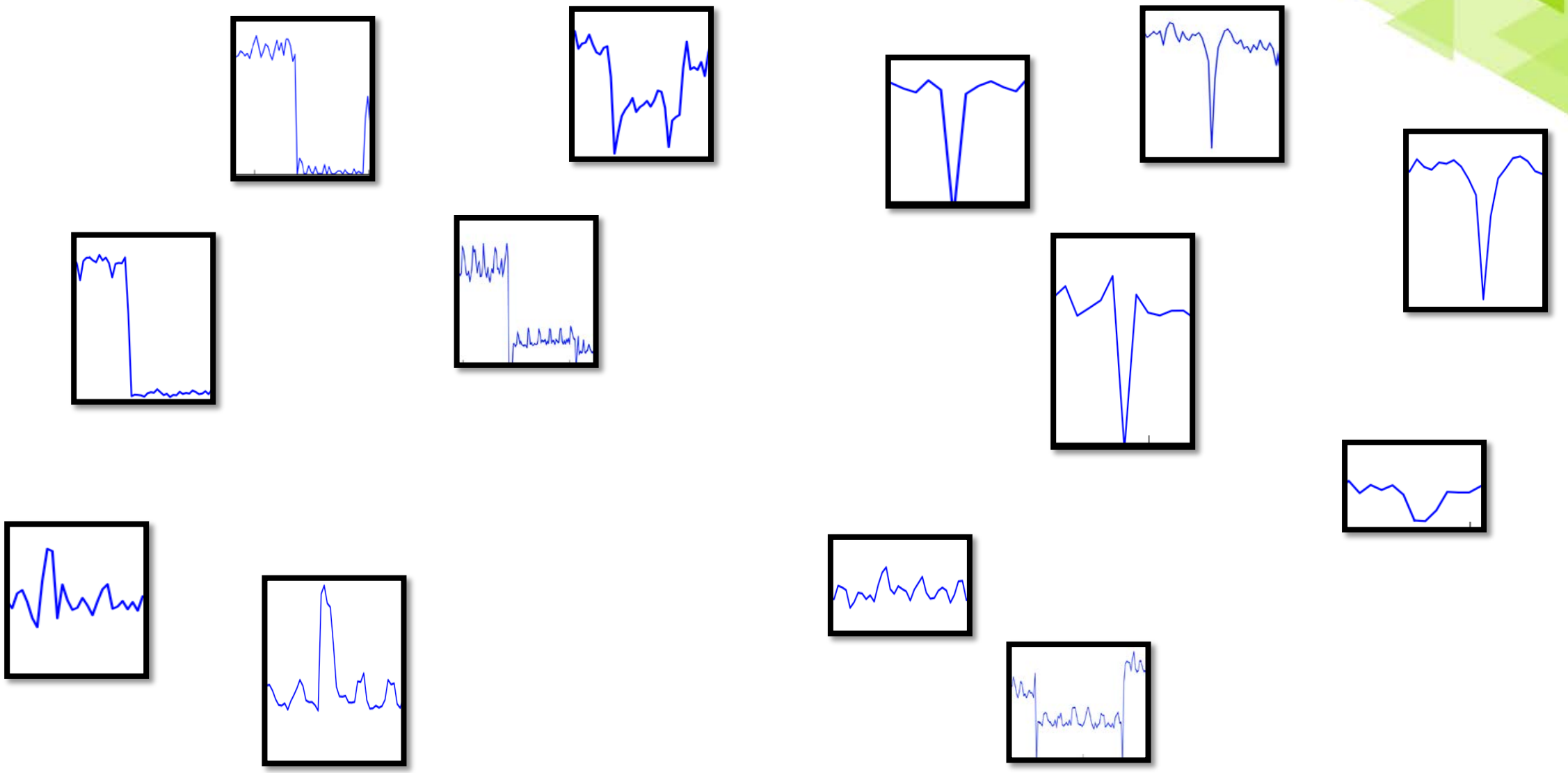
Step 1: (Pre-process) Fragment the data (shingling)



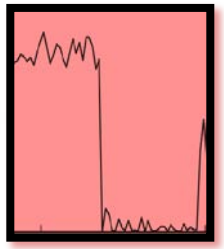
Step 1: (Pre-process) Fragment the data (shingling)



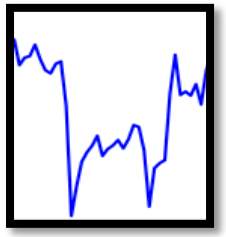
Step 1: (Pre-process) Fragment the data (shingling)

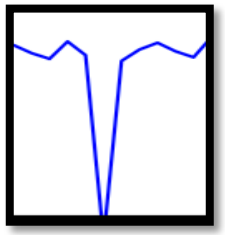


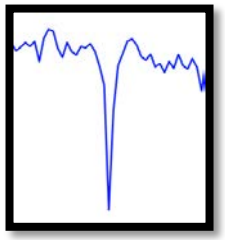
Step 2: Embed these fragments into a metric space

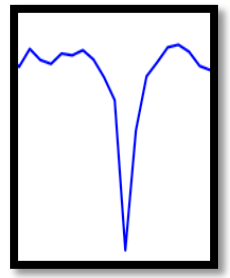


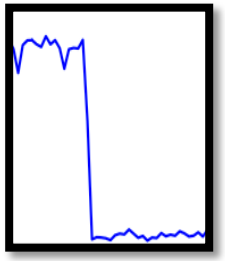
sustained dip (Sev-1)

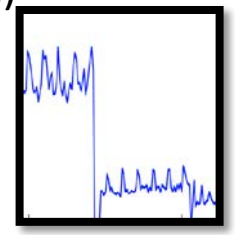


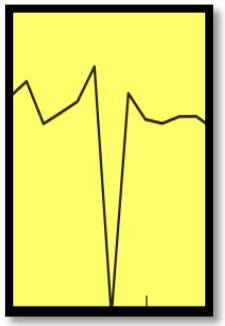




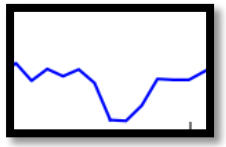


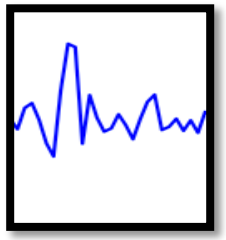


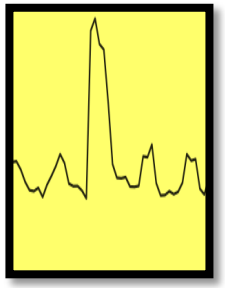




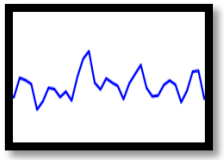
don’t bother

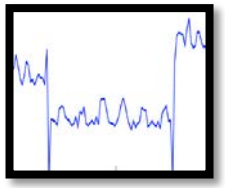




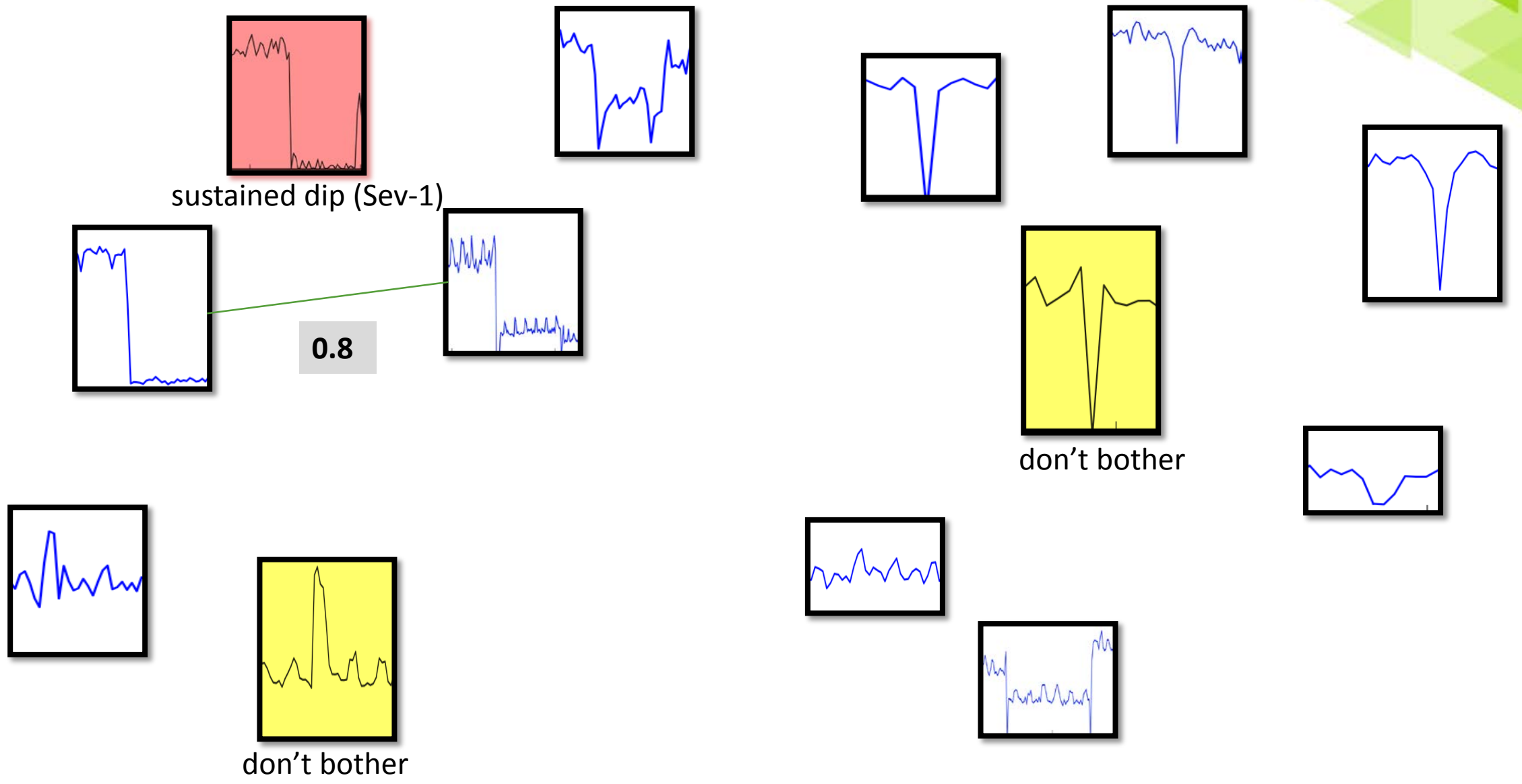


don’t bother

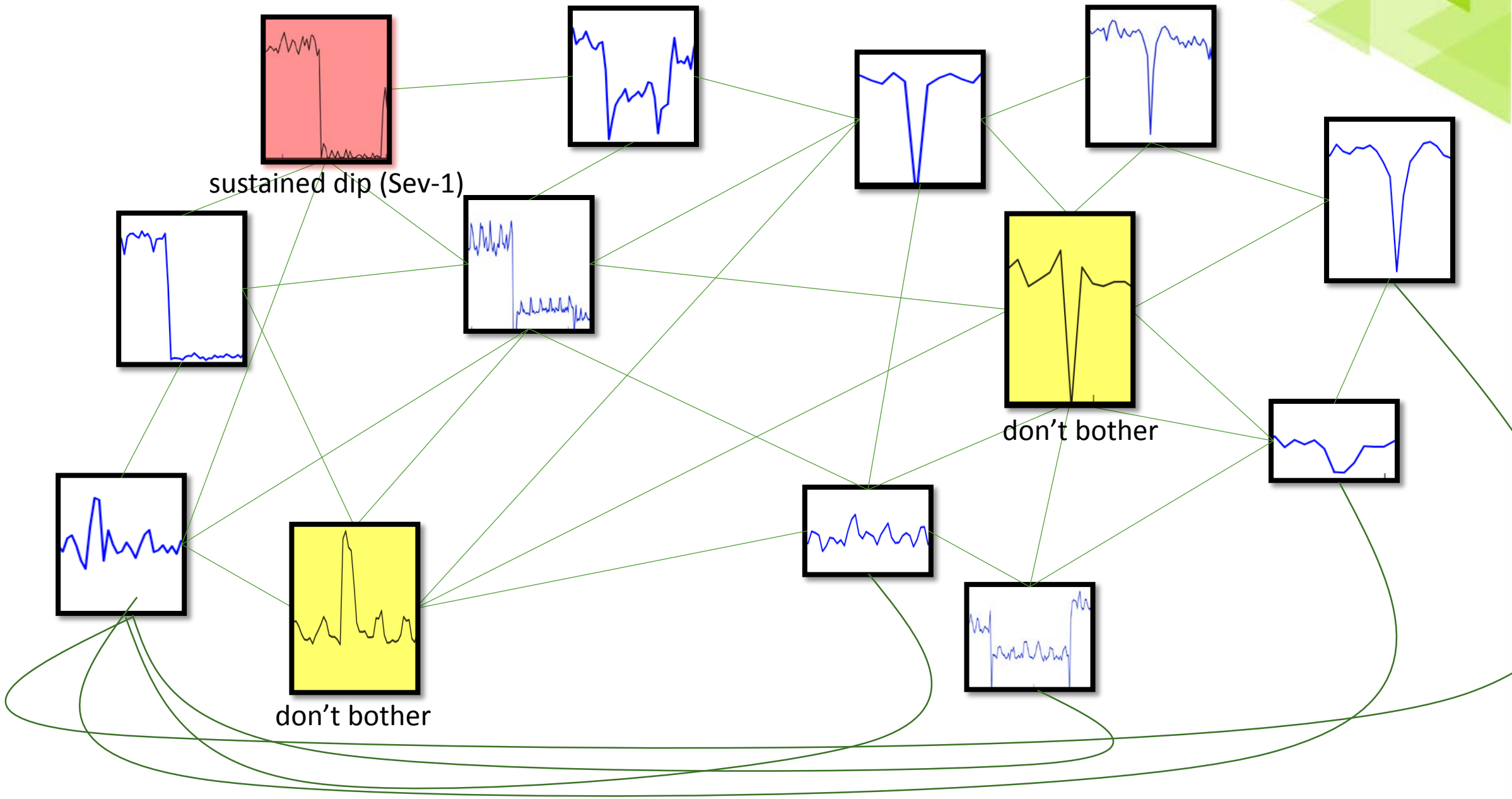




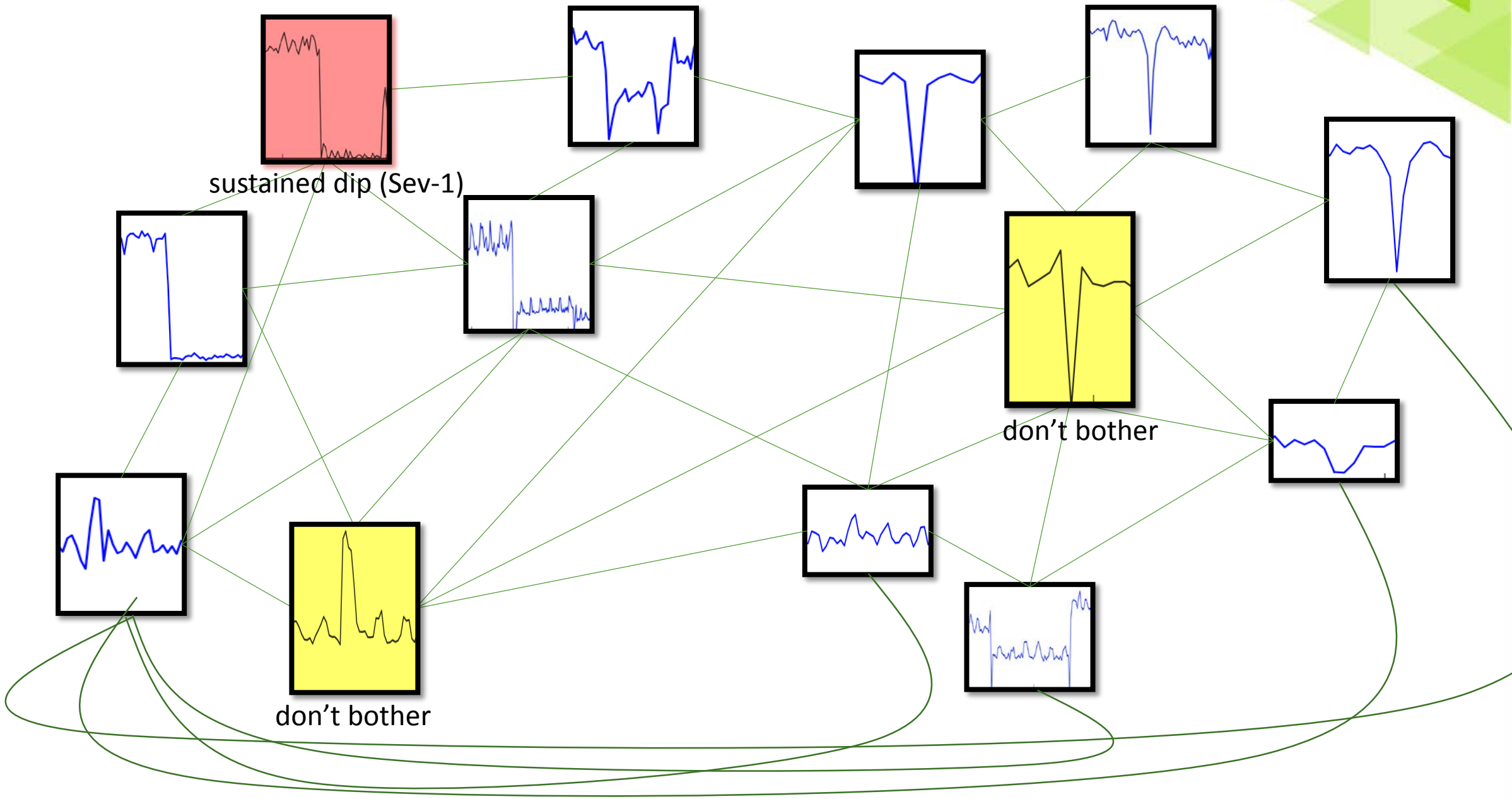
Expert provides few labels. Goal to “spread” this label to the remaining data.



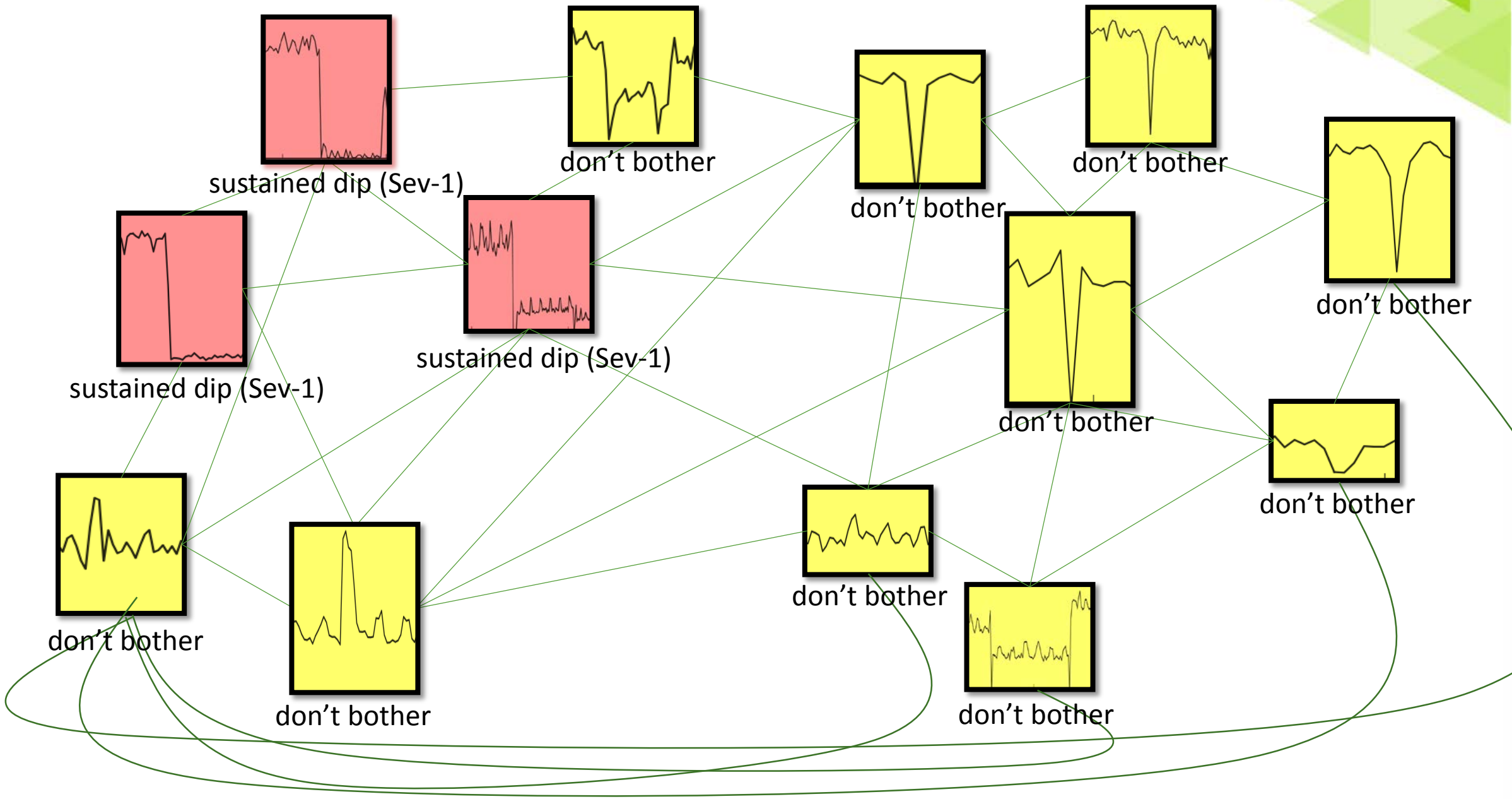
Step 3: Compute distances between fragments



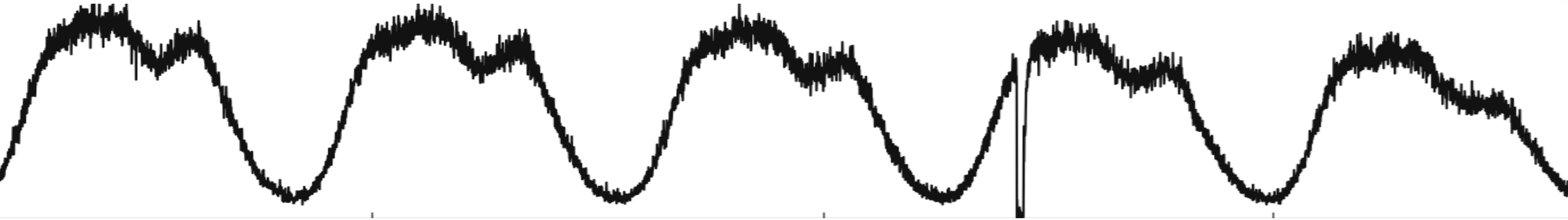
Step 4: Construct a “similarity graph”



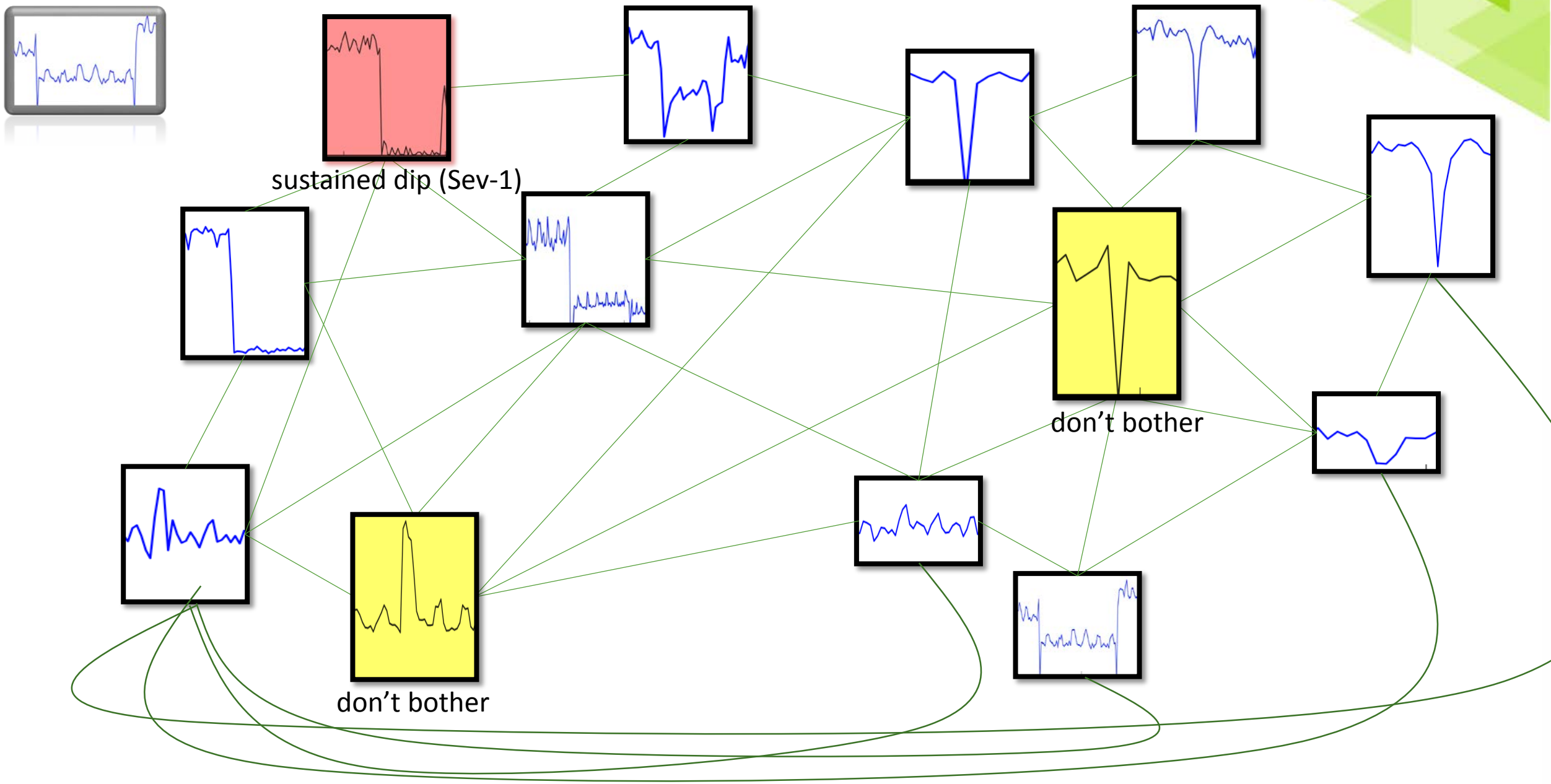
Step 5: "Label propagation" on this graph [Zhu Ghahramani Lafferty ICML '03]



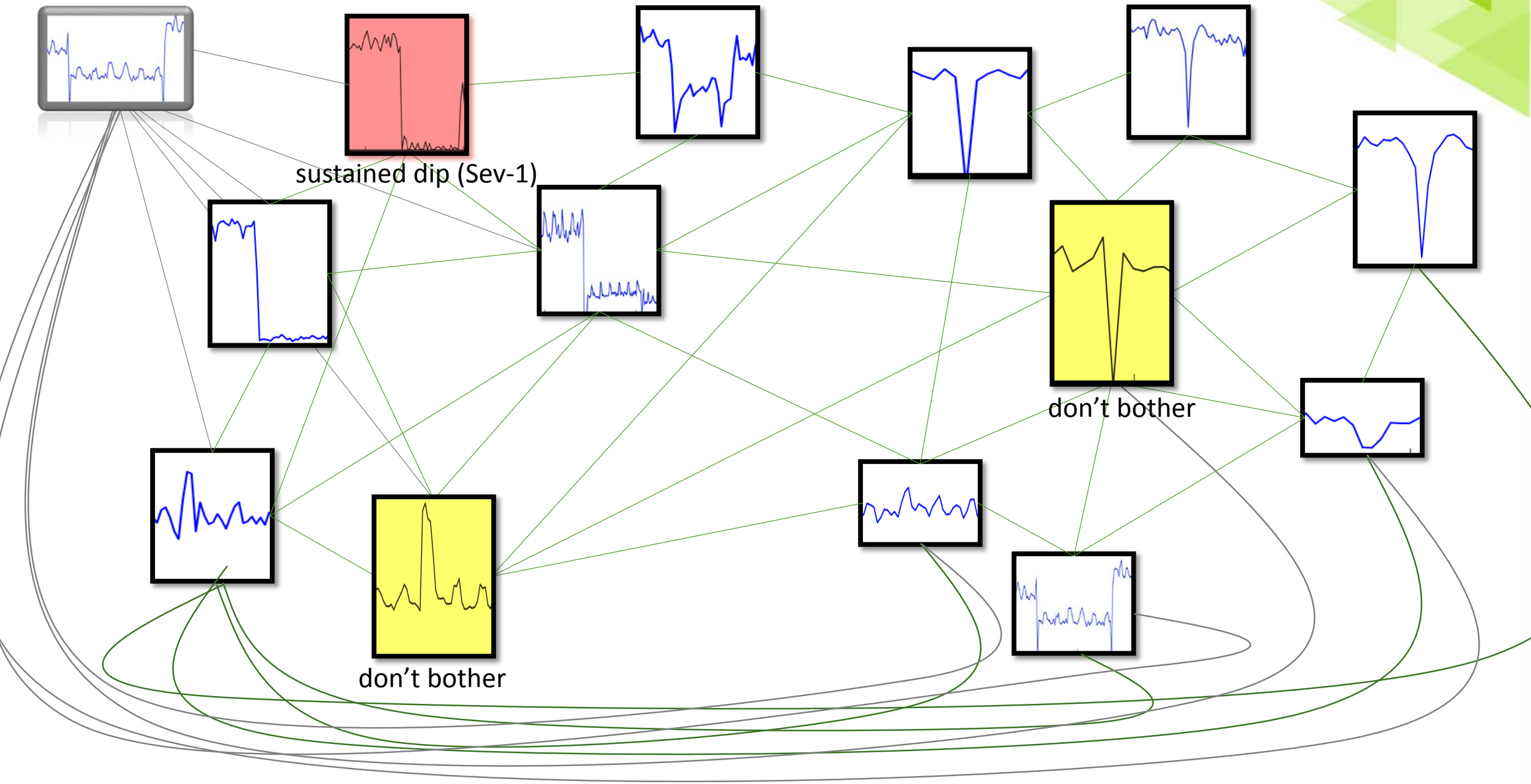
Output: **Label on each unlabeled fragment**



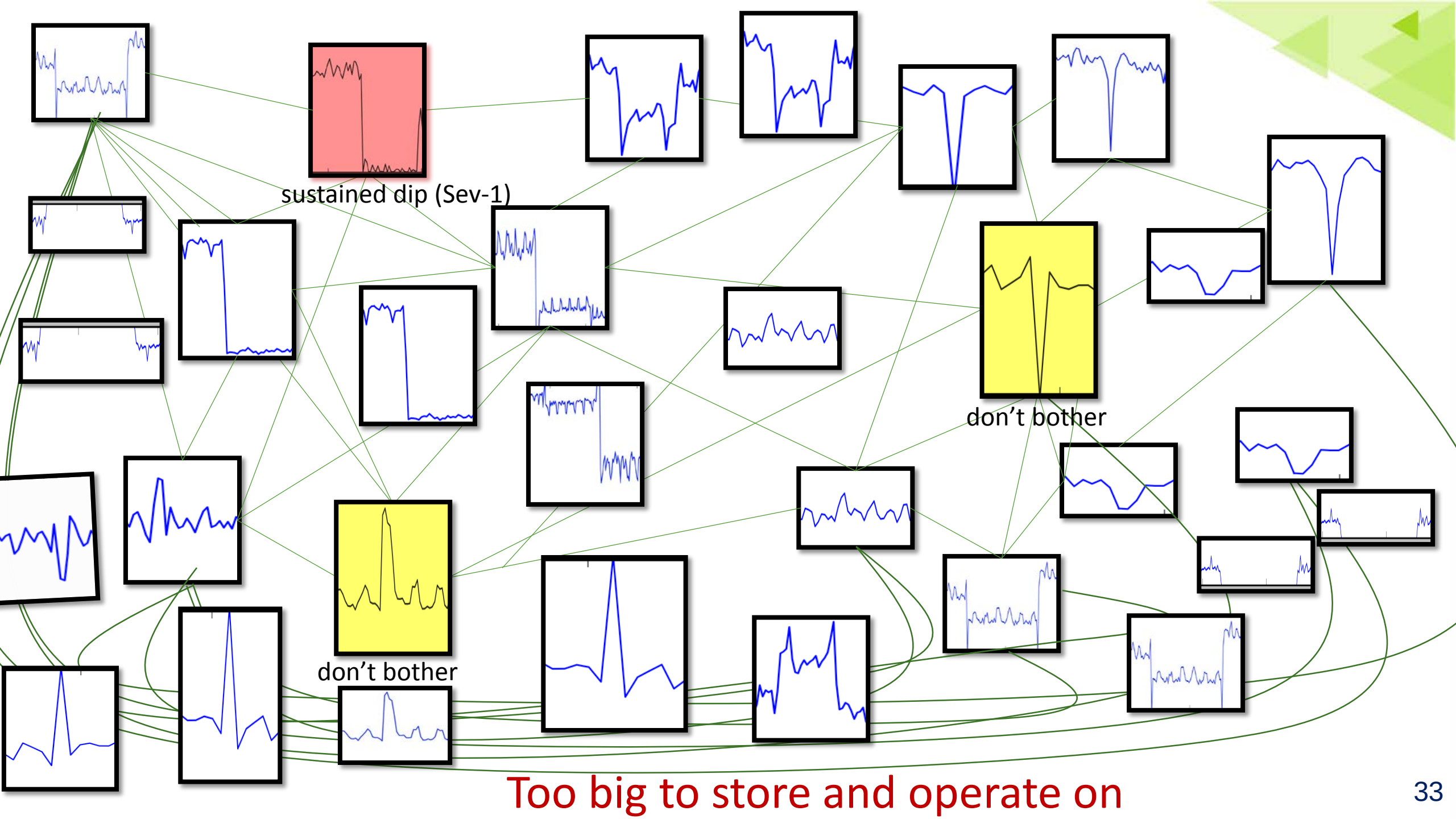
In a long stream…..



Fragments keep coming in



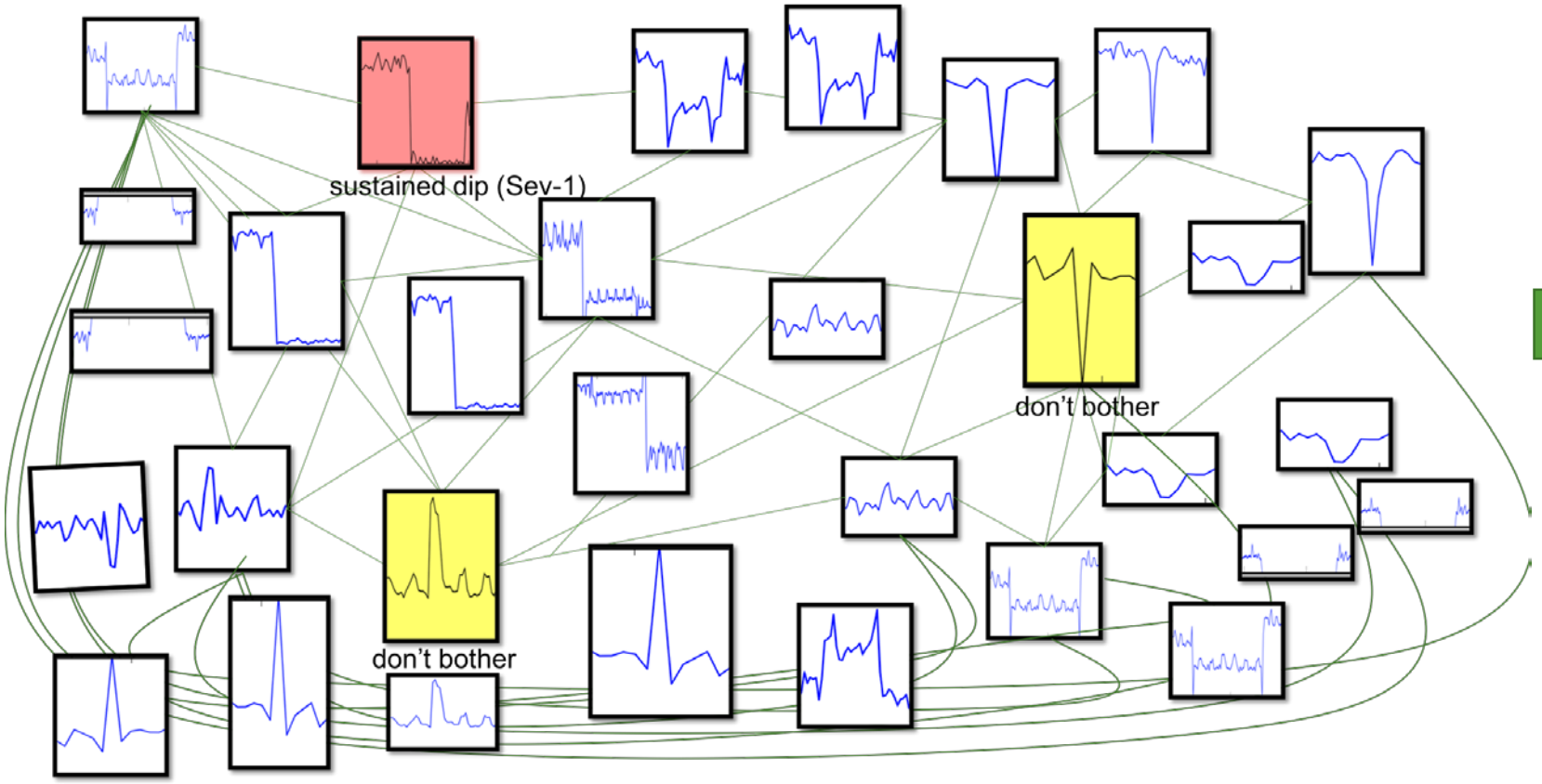
The graph keeps growing….



Too big to store and operate on

Our Algorithm in Pictures

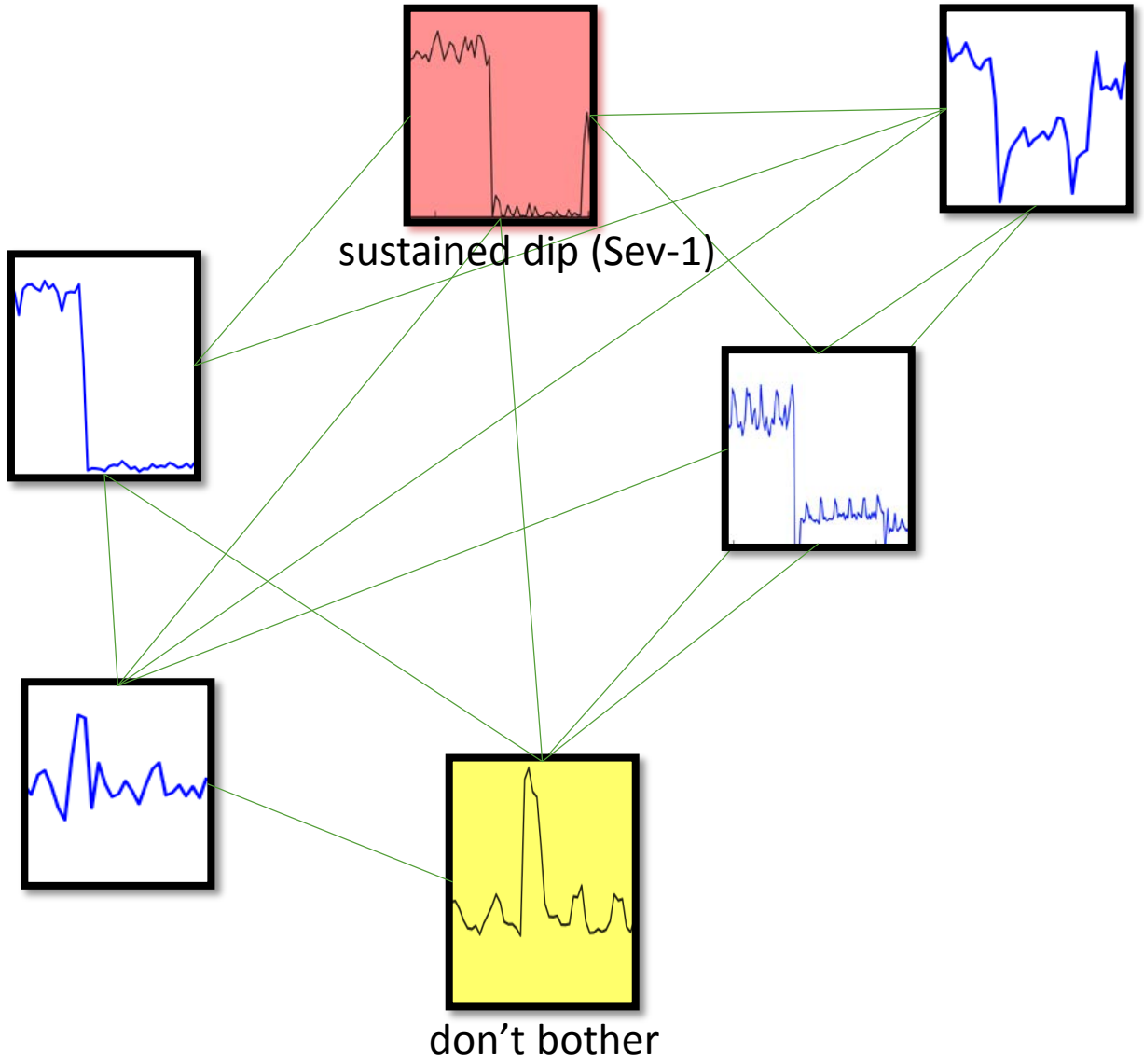
Constructs a compressed temporal graph



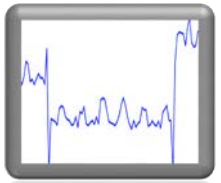
Compressed Graph

Our Algorithm in Pictures

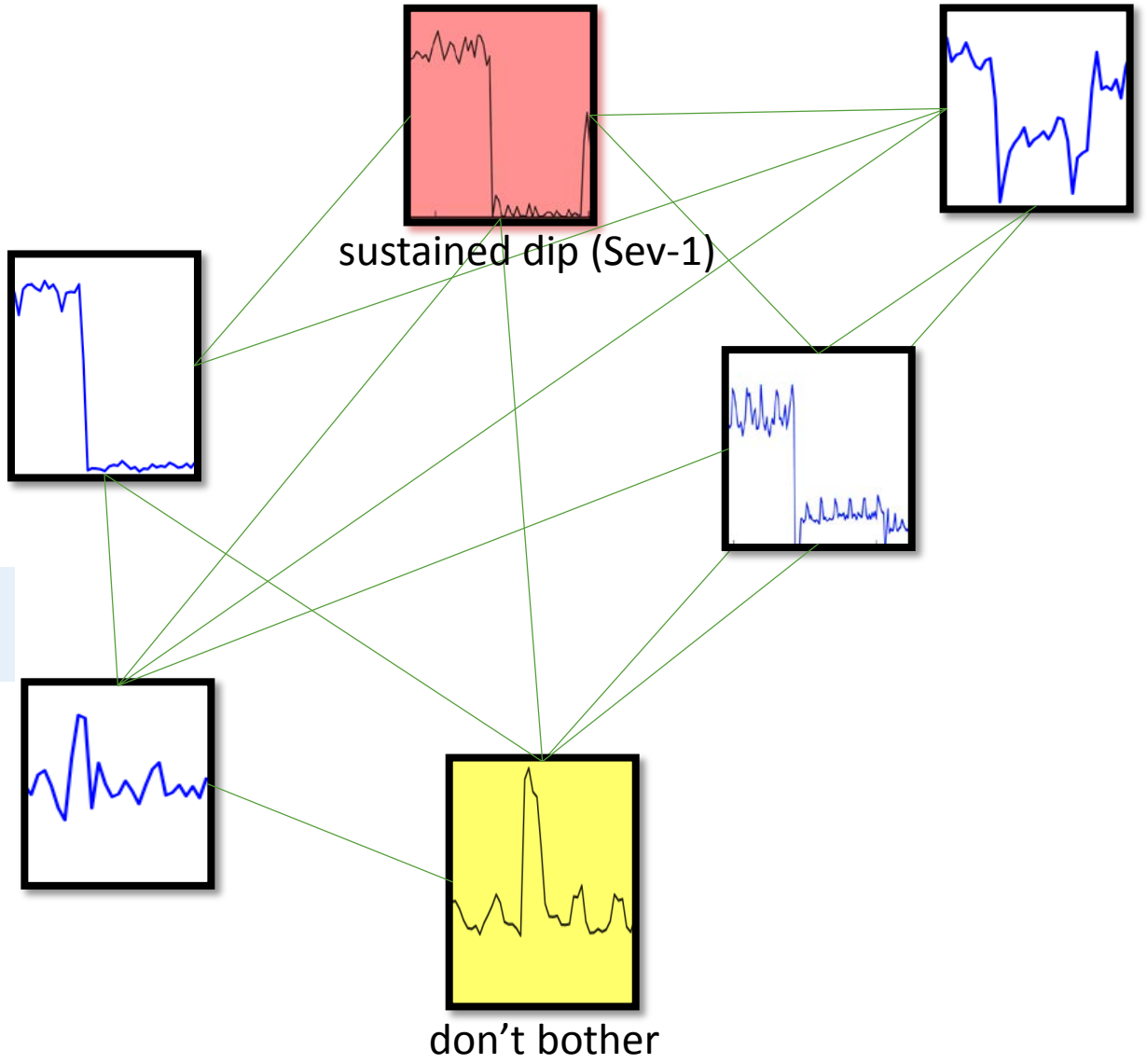
Maintains the graph over a sliding window



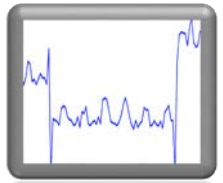
Our Algorithm in Pictures



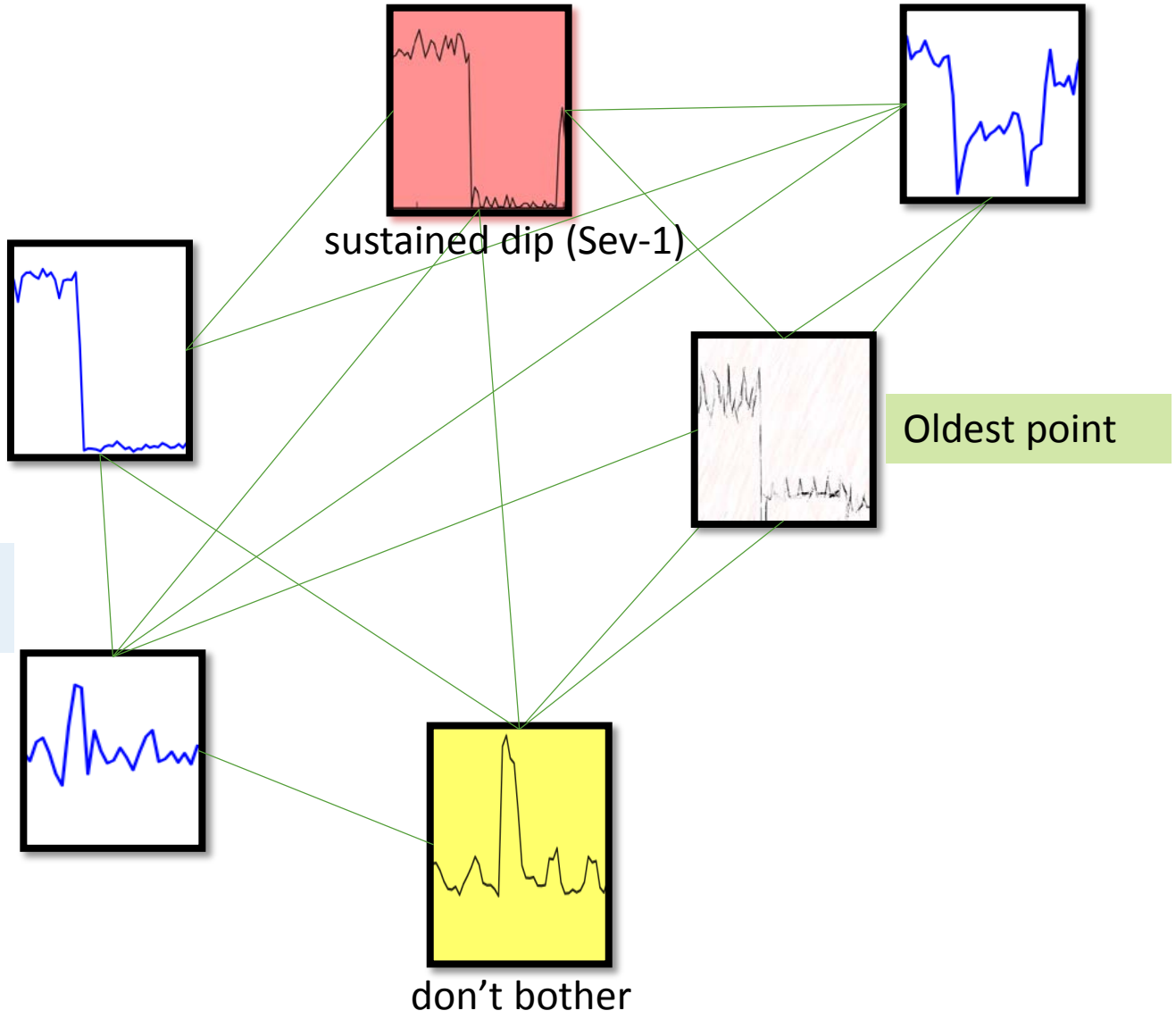
When a new point arrives



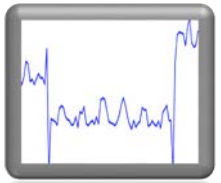
Our Algorithm in Pictures



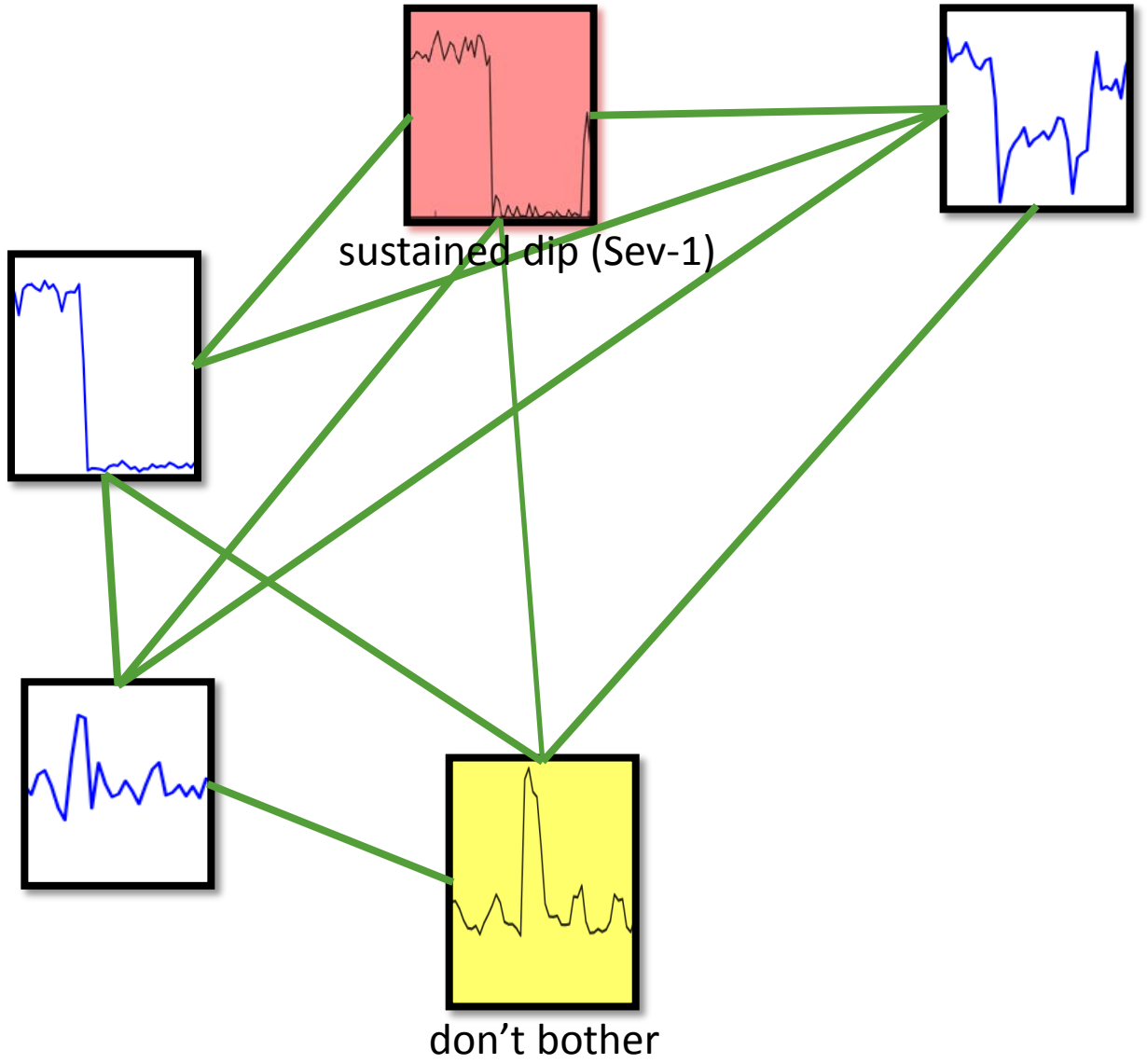
Delete oldest point



Our Algorithm in Pictures

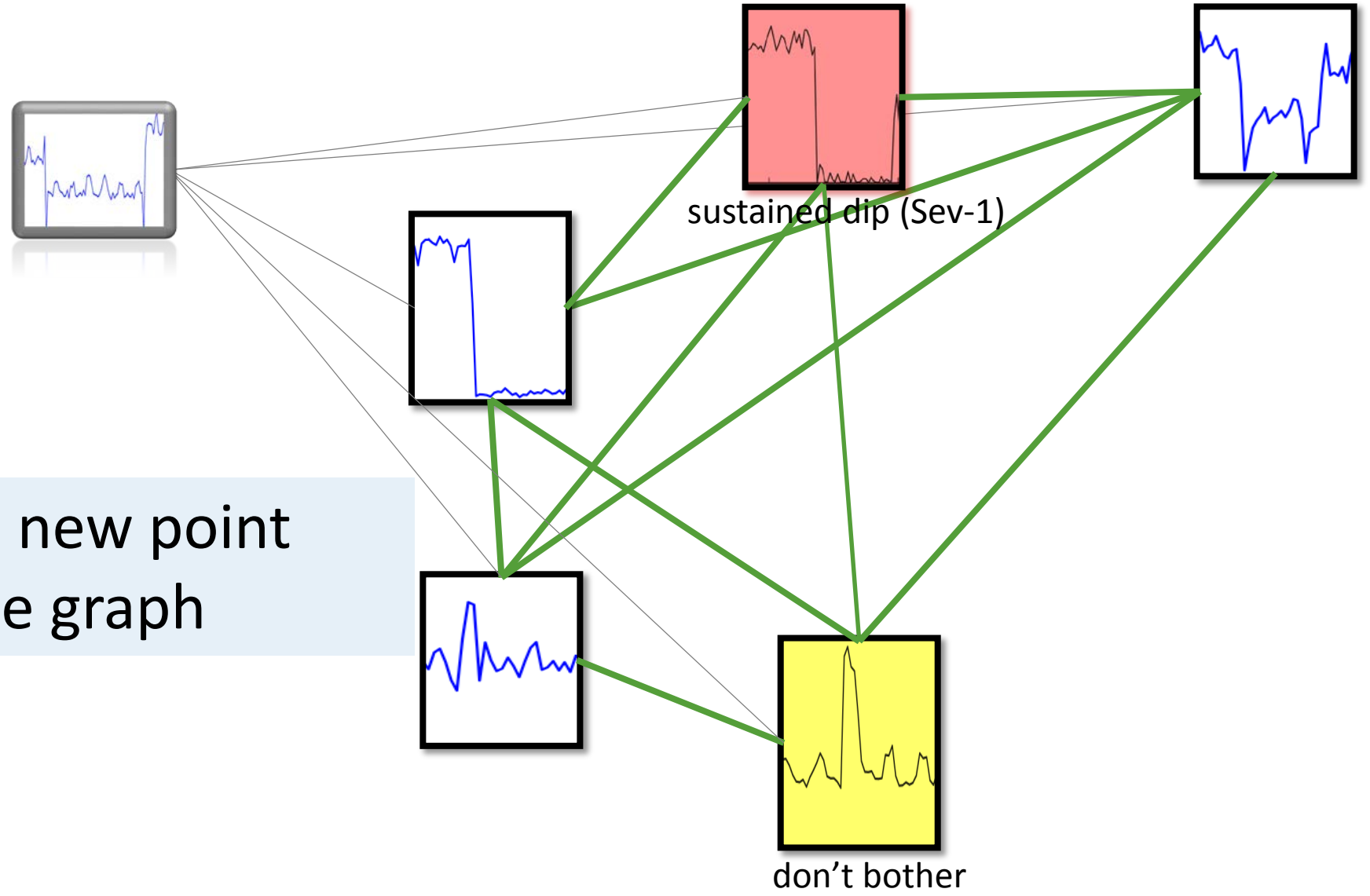


Update weights
on the edges
by **star-mesh** transform
(a concept from electric circuits)*



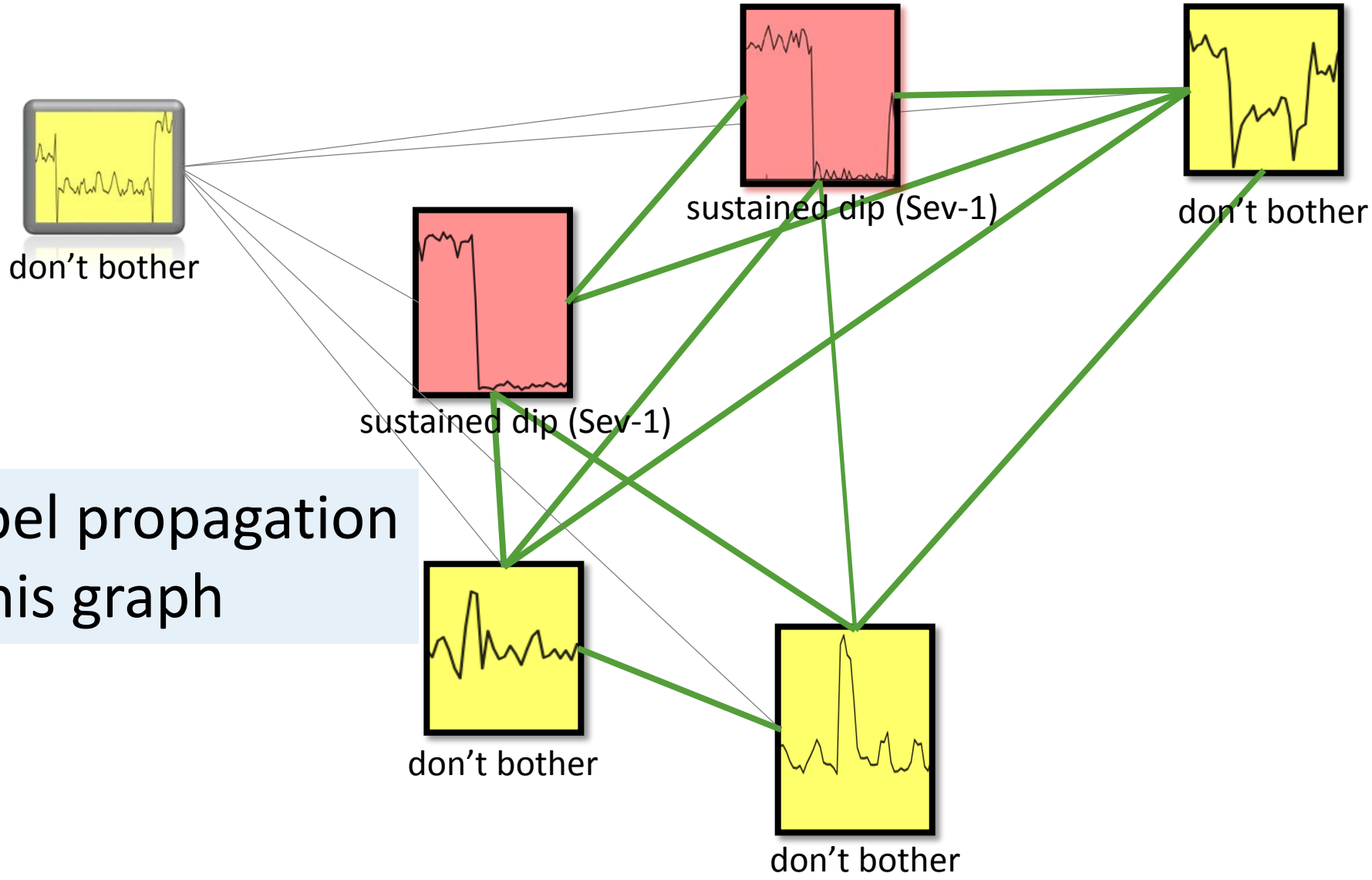
* Kennelly 1889

Our Algorithm in Pictures



Add the new point to the graph

Our Algorithm in Pictures



Perform label propagation on this graph

Our Theoretical Result

Temporal graph

(A graph where we only have edges from a point to its recent past)

Our Theoretical Result

Theorem:

Equivalent for the purpose of label propagation

Temporal graph

(A graph where we only have edges from a point to its recent past)

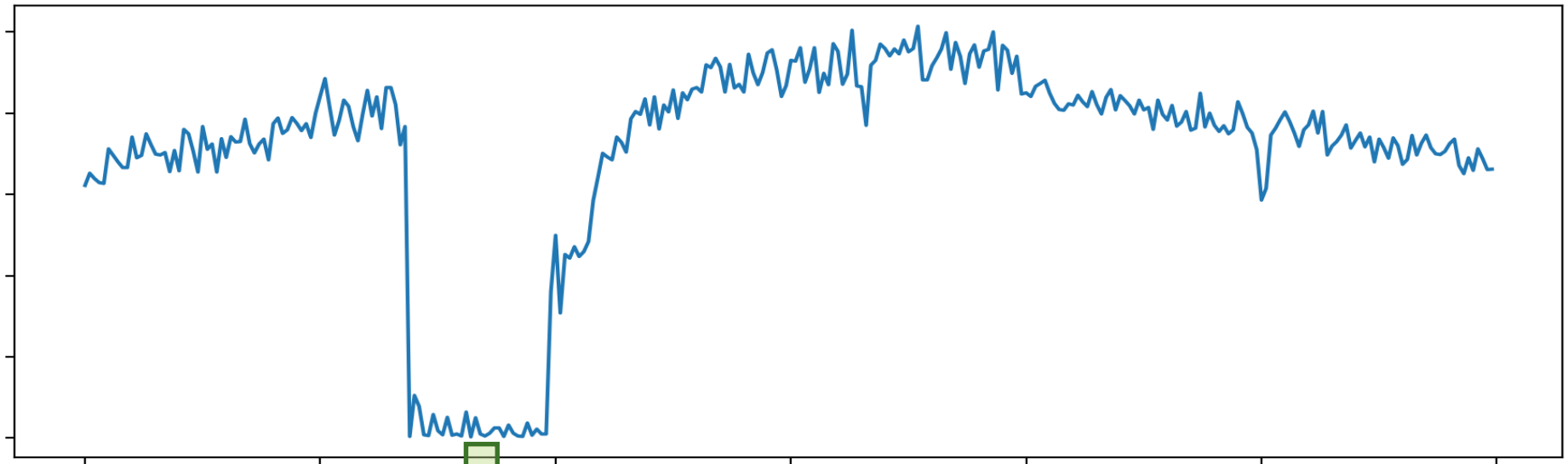
≡

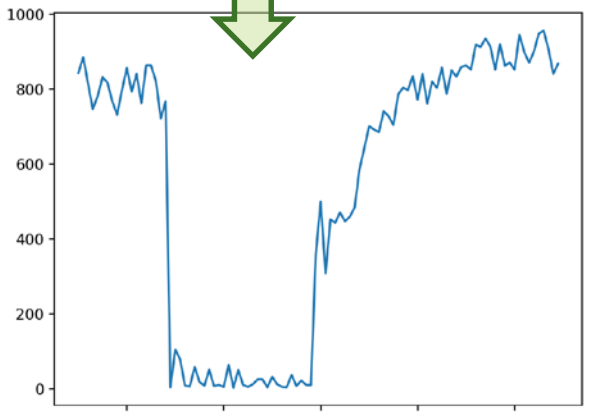
Compressed graph

Experimental Results

Streaming Timeseries Classification

Orders per minute





sustained dips

Amazon orders dataset

For 1 year of data: Accuracy > 95%
(just one labeled instance)
(memory consumed = 0.4 MB)

Drone Video – Identifying Balloons

Labeled data used for training

Correctly labeled frames

Accuracy > 77%*

(with only 5 labeled frames)

(memory consumed = 2 MB)

(response time = 0.4 sec)

*Without relying on modern CV techniques

How do I start using this?

AWS Kinesis Data Analytics

Stay tuned, launch in AWS Kinesis Data Analytics by Q2-2018

Conclusion

An ML algorithm that

- Operates on a stream
- Performs real-time classification
- Requires little labeled data
- Learns from both labeled and unlabeled data
- Works in both theory and practice

Conclusion

An ML algorithm that

- Operates on a stream
- Performs real-time classification
- Requires little labeled data
- Learns from both labeled and unlabeled data
- Works in both theory and practice

Big thanks to AWS Kinesis Analytics team

Appendix