

# Learning Time-Varying Graphs from Data

Alberto Natali

Graphs&Data Seminar // 07 December 2023

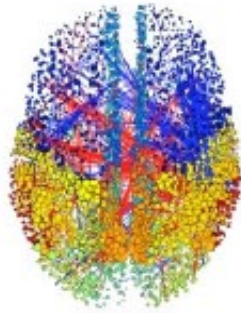


# In the previous talk ...

- What is the relationship between classical signal processing and graph signal processing?
- What are graph signals?
- Which architectures can be used to process graph signals?
- Applications?

# In this talk...

In many scenarios the structure of the network is *unknown*



Brain connections



Social interactions



Stock correlations

However, we can measure *data/signals* on these networks:

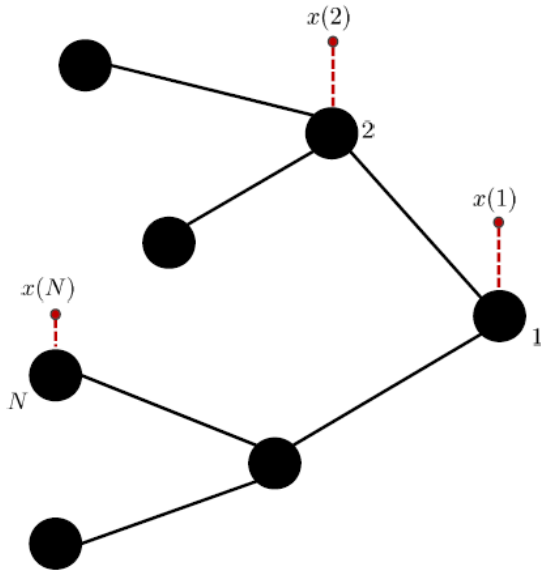
- fMRI, ultrasound, ECoG signals, ...
- User profiling, social networks data, ...
- Daily stock volumes, earnings reports, ...

In this talk...

Learn the **topology** of the network from  
the available **data**

# GSP 101: notation

We consider data element  $\mathbf{x}$  to reside over a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{S})$



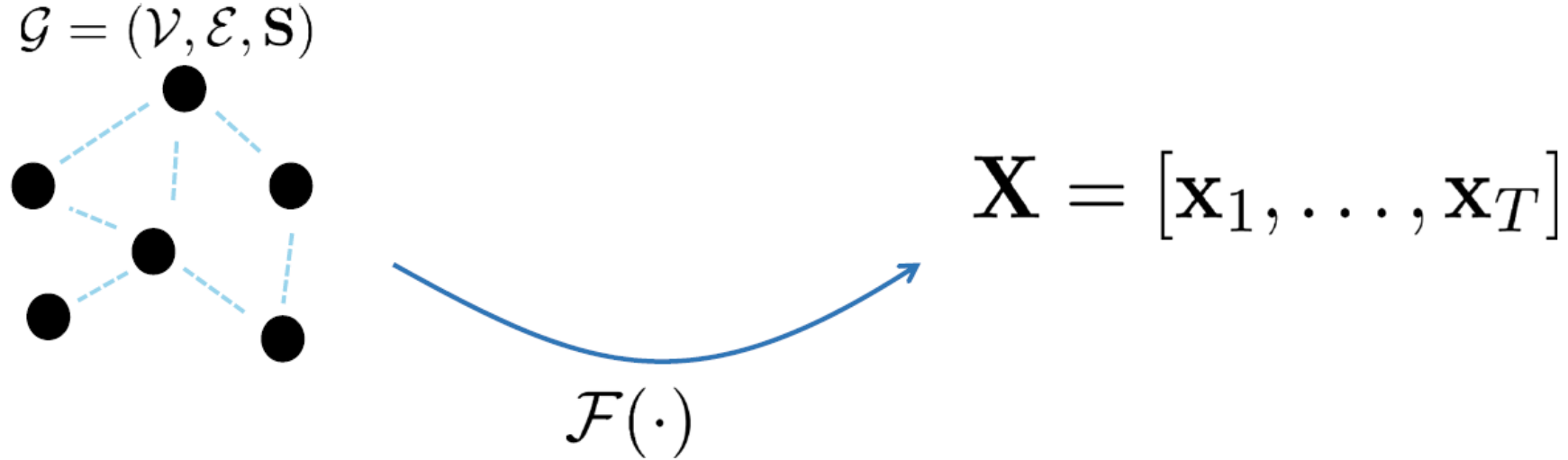
- $\mathcal{V} = \{1, \dots, N\}$  node set
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  edge set
- $\mathbf{S} \in \mathbb{R}^{N \times N}$  is the shift operator

- $\mathbf{x}$  is the **graph signal**
- $x(i)$  is the value at node  $i \in \mathcal{V}$
- $\mathbf{S}$  is typically the adjacency matrix or Laplacian  $\leftarrow$  graph structure

# Static Topology Inference

(graph unknown but fixed over time)

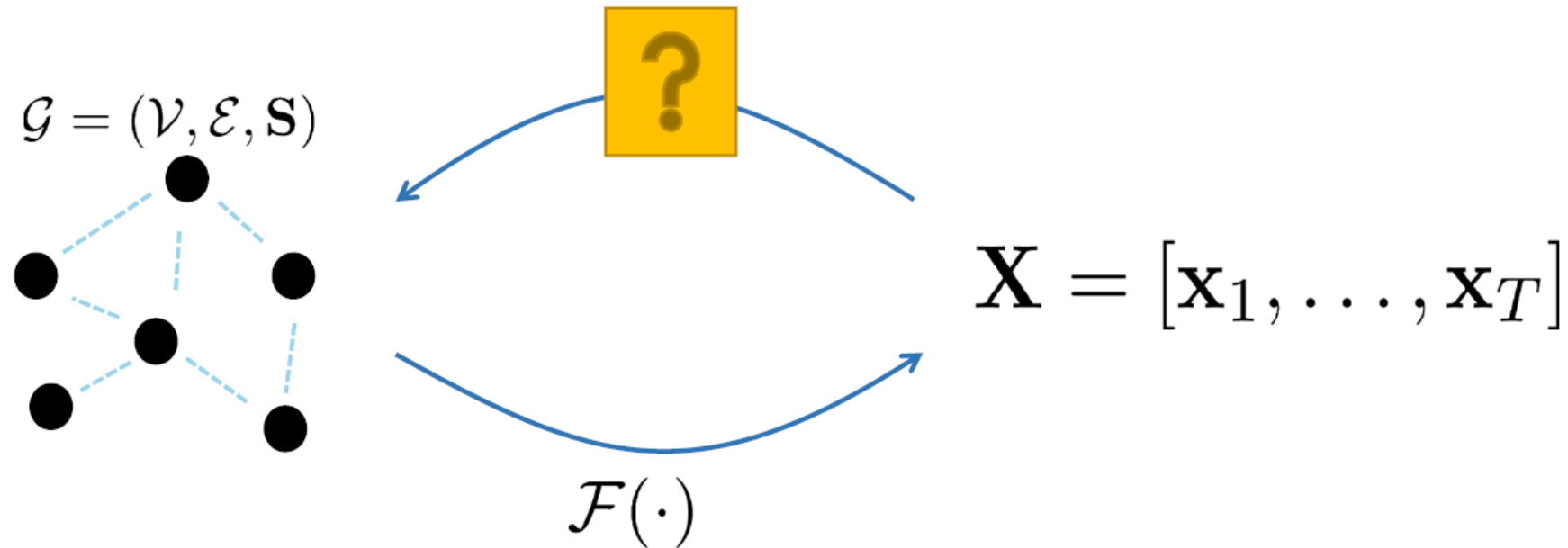
Consider matrix  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$





# Static Topology Inference (graph unknown but fixed over time)

Consider matrix  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$



How to infer graph  $\mathcal{G}$  from  $\mathbf{X}$ ? (How to infer  $\mathbf{S}$ ?)

# Graph Topology Inference

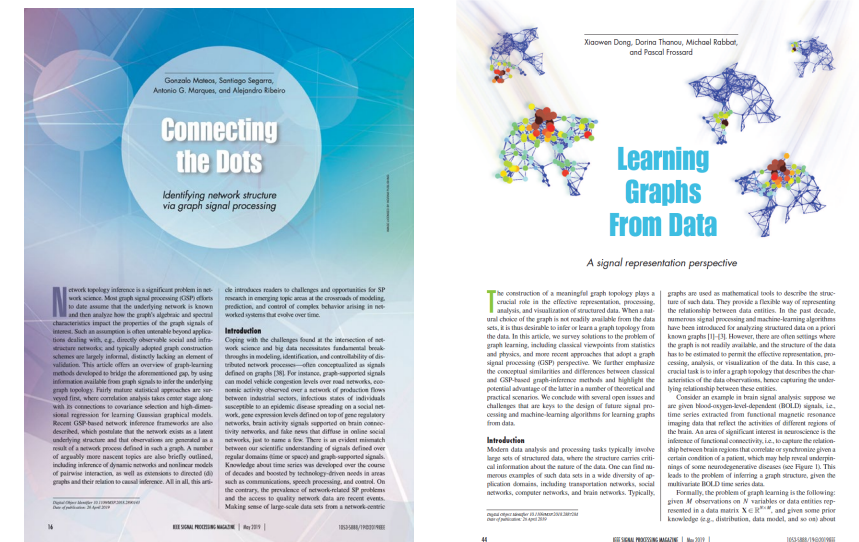
- **Statistical Methods**

- ⇒ Covariance Selection (GMRF) [Dempster '72]
- ⇒ Neighbourhood-based Regression [Meinshausen '06]
- ⇒ Graphical Lasso [Friedman '08]
- ⇒ Laplacian-constrained GMRF [Lake '10] [Egilmez '17]

- **Graph Signal Processing**

- ⇒ Smoothness-based models [Dong '16] [Kalofolias '16]
- ⇒ Edge subset selection [Chepuri '17] [Romero '17]
- ⇒ Spectral Templates [Pasdeloup '17] [Segarra '17] [Shafipour '18]
- ⇒ Heat-diffusion [Thanou '17]
- ⇒ Dynamics and non linearity [Giannakis '18]

- Machine Learning on Graphs [Scarselli '08] [Defferdard '16]

 INVITED  
PAPER

# Topology Identification and Learning Over Graphs: Accounting for Nonlinearities and Dynamics

*This article focuses on the problem of learning graphs from data, in particular, to capture the nonlinear and dynamic dependencies.*

By GEORGIOZ B. GIANNAKIS, Fellow IEEE, YANNING SHEN, Student Member IEEE,  
AND GEORGIOZ VASILEIOS KARANIKOLAS, Student Member IEEE

[illegible]

and lends itself to batch and computationally affordable online learning algorithms, which include novel Kalman filters over graphs. Real data experiments highlight the impact of the nonlinear and dynamic models on consumer and financial networks, as well as gene-regulatory and functional connectivity brain networks, where connectivity patterns revealed exhibit discernible differences relative to existing approaches.

**KEYWORDS** | Kernel-based models; network topology; inference; nonlinear modeling; time-varying networks

## I. INTRODUCTION

The science of networks and networked interactions has recently emerged as a major catalyst for understanding the behavior of complex systems [26], [67], [93], [109]. Such systems are typically described by graphs, and can be found in many domains. For example, human interaction over the web commonly occurs on social networks such as Facebook and Twitter, while sophisticated brain functions are the result of complex physical interactions among neurons; see, e.g., [95] and references therein. Other complex networks show up in diverse fields including financial markets, genomics, proteomics, power grids, and transportation systems, to name a few.

Despite their popularity, single-layer networks may fall short in describing complex systems. For instance, modeling interactions between two individuals using a single edge weight can be an oversimplification of reality. Generalizing their single-layer counterparts, multilayer networks allow nodes to belong to different groups, termed layers [10], [66].

0010-0101/0/2010 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See [http://www.ieee.org/publications\\_standards/publications/rights/index.html](http://www.ieee.org/publications_standards/publications/rights/index.html) for more information.



# Graph-Data Model

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \boldsymbol{\Sigma} \in \mathbb{S}_{++}^N$$

sparsity pattern precision matrix  
=  
conditional independence



## Gaussian Graphical Model (GGM)

$$\begin{aligned} &\underset{\mathbf{S}}{\text{minimize}} \quad -\log \det(\mathbf{S}) + \text{tr}(\mathbf{S}\hat{\boldsymbol{\Sigma}}) \\ &\text{s. t.} \quad \mathbf{S} \in \mathbb{S}_{++}^N \end{aligned}$$

$$x_t(i) = \sum_{j \neq i} S(i, j)x_t(j) + e_t(i),$$

Sort of neighbourhood regression



## Structural Equation Model (SEM)

$$\begin{aligned} &\underset{\mathbf{S}}{\text{minimize}} \quad \frac{1}{2T} \|\mathbf{X} - \mathbf{SX}\|_F^2 + g(\mathbf{S}), \\ &\text{s. t.} \quad \mathbf{S} \in \mathcal{S} \end{aligned}$$

$$\text{LQ}_G(\mathbf{x}_t) := \mathbf{x}_t^\top \mathbf{L} \mathbf{x}_t = \sum_{i \neq j} W(i, j)(x_t(i) - x_t(j))^2$$

Signals are smooth on the graph



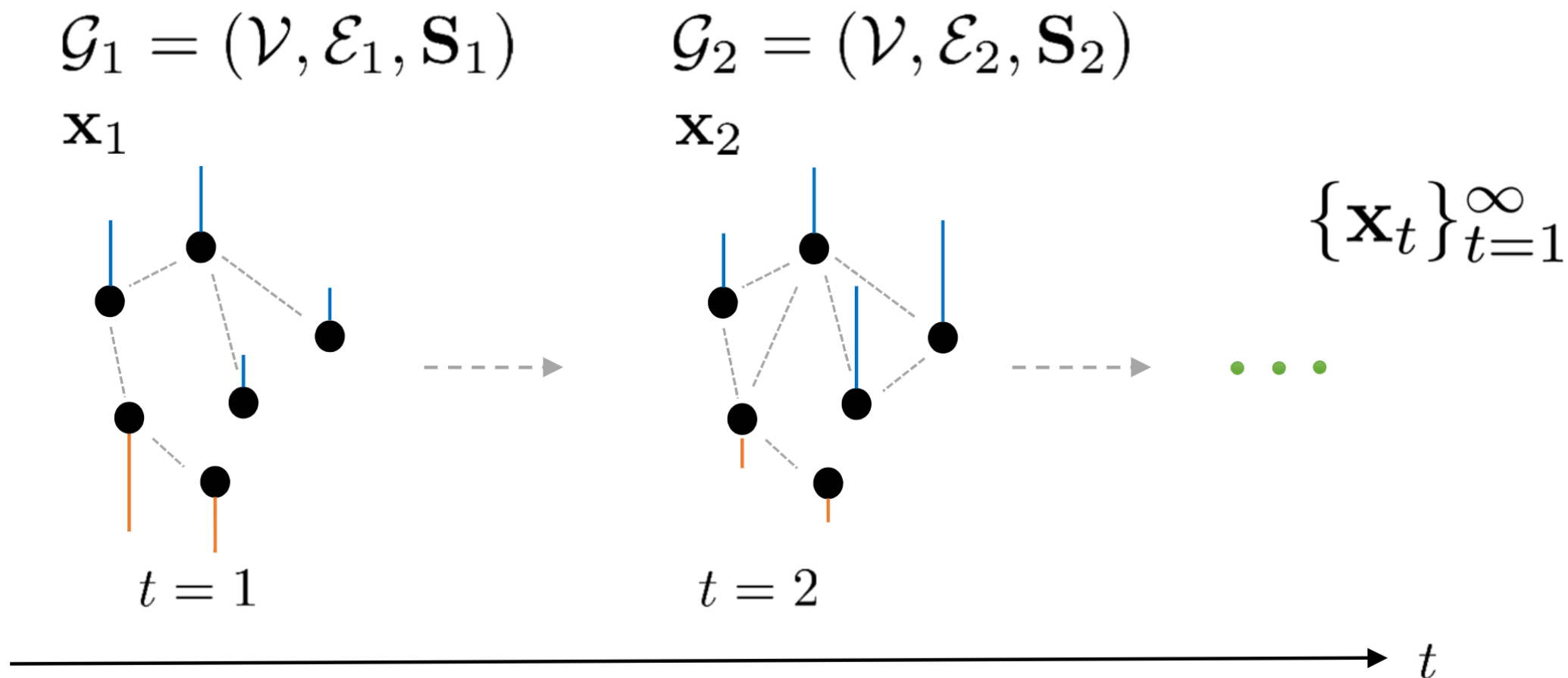
## Smoothness-Based Model (SBM)

$$\begin{aligned} &\underset{\mathbf{S}}{\text{minimize}} \quad \frac{1}{T} \text{tr}(\mathbf{X}^\top \mathbf{S} \mathbf{X}) + g(\mathbf{S}), \\ &\text{s. t.} \quad \mathbf{S} \in \mathcal{S} \end{aligned}$$

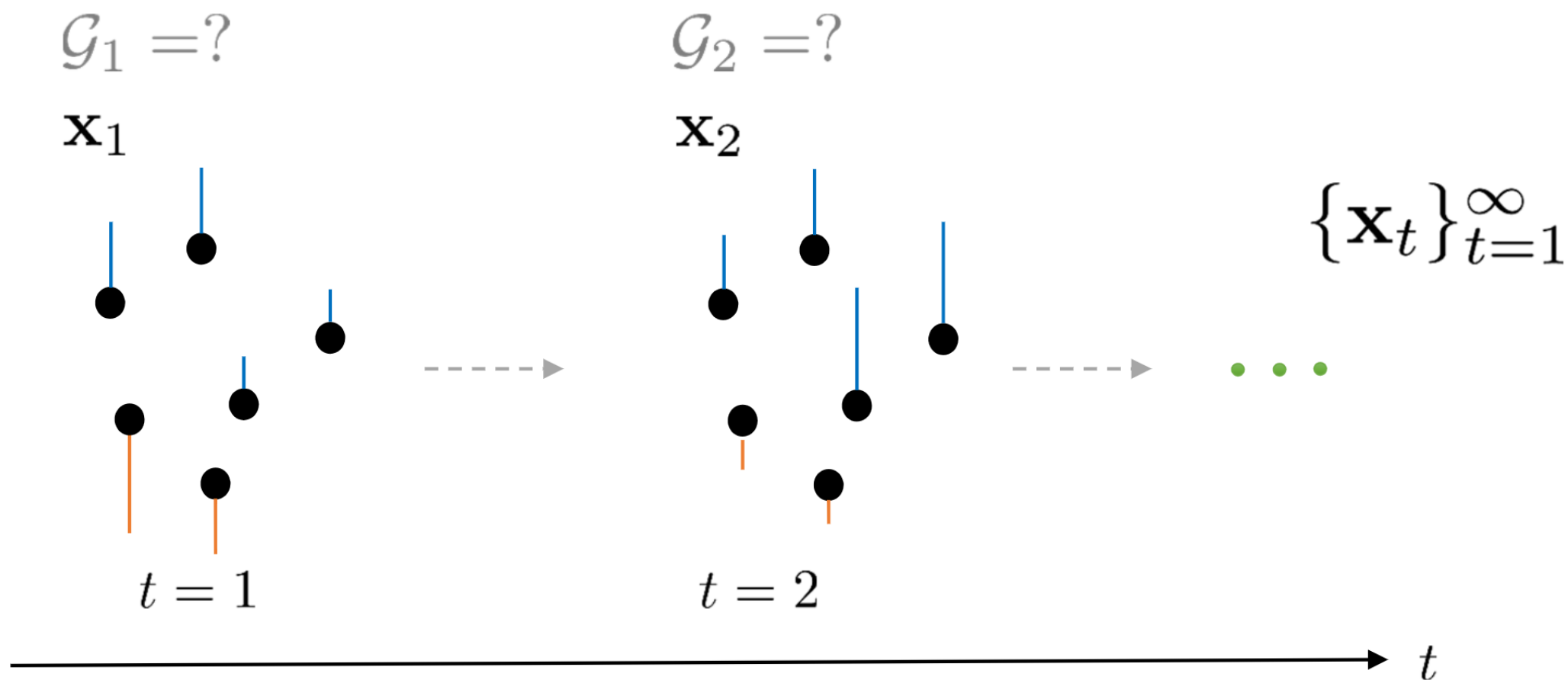
However...

Often the underlying **network** changes over **time**, and **data** arrive in a **streaming** fashion

# Scenario

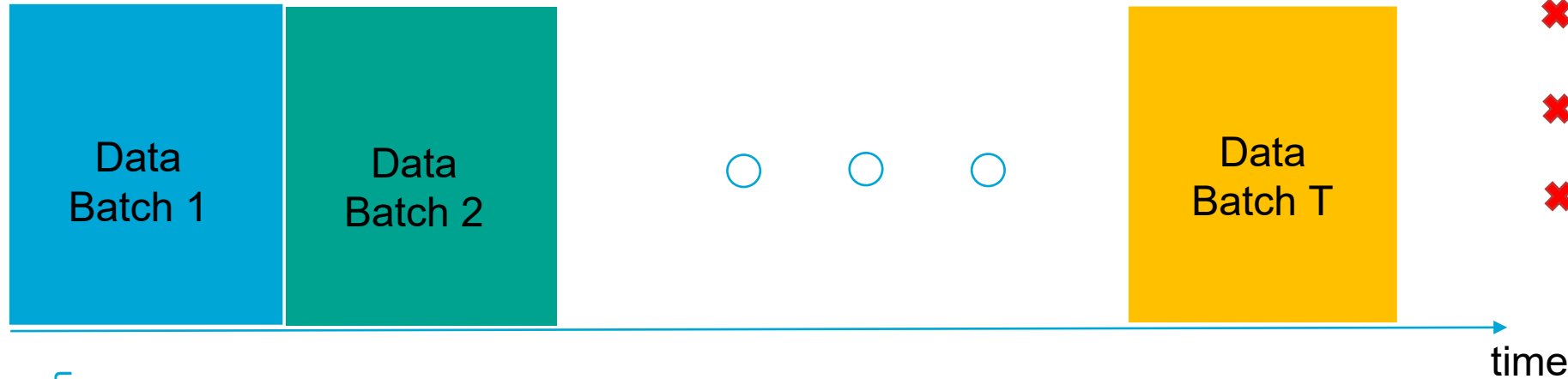


# Scenario



# Standard Approach

for instance EEG rec.

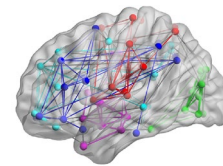
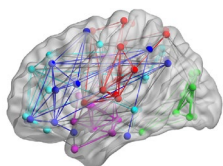
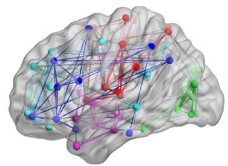


✗ Computationally expensive

✗ Not Real Time

✗ Where to window?

optimization



Solutions over time

# Our work

- A model-agnostic framework for learning time-varying graphs from streaming data
  - An umbrella for all the methods we have seen so far
  - Theoretical analysis on the abstract formulation
  - Operates in non-stationary environments
  - Accelerated through a prediction-correction scheme



# Our work

For every new graph signal  $\mathbf{x}_t$  we have a new **time-varying function**  $F(\mathbf{S}; t) := \underbrace{f(\mathbf{S}; t)}_{\text{Data fidelity}} + \underbrace{\lambda g(\mathbf{S}; t)}_{\text{Regularization and constraints}}$

$$\mathbf{S}_t^* := \operatorname{argmin}_{\mathbf{S}} f(\mathbf{S}; t) + \lambda g(\mathbf{S}; t) \quad \text{for } t = 1, 2, \dots$$

## TV-GGM

$$\begin{aligned} f(\mathbf{S}; t) &= -\log \det(\mathbf{S}) + \operatorname{tr}(\mathbf{S} \hat{\Sigma}_t) \\ g(\mathbf{S}; t) &= \iota_{\mathcal{S}}(\mathbf{S}) \end{aligned}$$

## TV-SEM

$$\begin{aligned} f(\mathbf{S}; t) &= \frac{1}{2} [\operatorname{tr}(\mathbf{S}^2 \hat{\Sigma}_t) - 2 \operatorname{tr}(\mathbf{S} \hat{\Sigma}_t) + \operatorname{tr}(\hat{\Sigma}_t)] \\ g(\mathbf{S}; t) &= \|\mathbf{S}\|_1 + \iota_{\mathcal{S}}(\mathbf{S}) \end{aligned}$$

## TV-SBM

$$\begin{aligned} f(\mathbf{S}; t) &= \operatorname{tr}(\operatorname{Diag}(\mathbf{S} \mathbf{1}) \hat{\Sigma}_t) - \operatorname{tr}(\mathbf{S} \hat{\Sigma}_t) \\ g(\mathbf{S}; t) &= \frac{\lambda_1}{4} \|\mathbf{S}\|_F^2 - \lambda_2 \mathbf{1}^\top \log(\mathbf{S} \mathbf{1}) + \iota_{\mathcal{S}}(\mathbf{S}) \end{aligned}$$

Dependence of the data through covariance matrix  $\hat{\Sigma}_t \longrightarrow \hat{\Sigma}_t = \gamma \hat{\Sigma}_{t-1} + (1 - \gamma) \mathbf{x}_t \mathbf{x}_t^\top$  EWMA

# Our work

For every new graph signal  $\mathbf{x}_t$  we have a new **time-varying function**  $F(\mathbf{S}; t) := \underbrace{f(\mathbf{S}; t)}_{\text{Data fidelity}} + \underbrace{\lambda g(\mathbf{S}; t)}_{\text{Regularization and constraints}}$

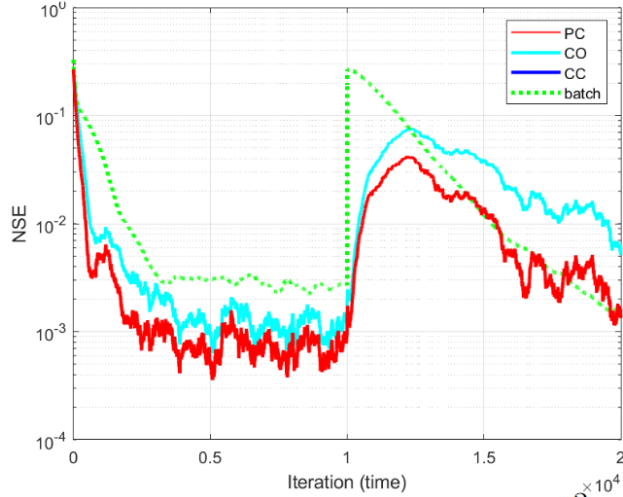
$$\mathbf{S}_t^* := \operatorname{argmin}_{\mathbf{S}} f(\mathbf{S}; t) + \lambda g(\mathbf{S}; t) \quad \text{for } t = 1, 2, \dots$$

Solved through an iterative routine relying on a **prediction-correction** scheme (onto the half-vec space)

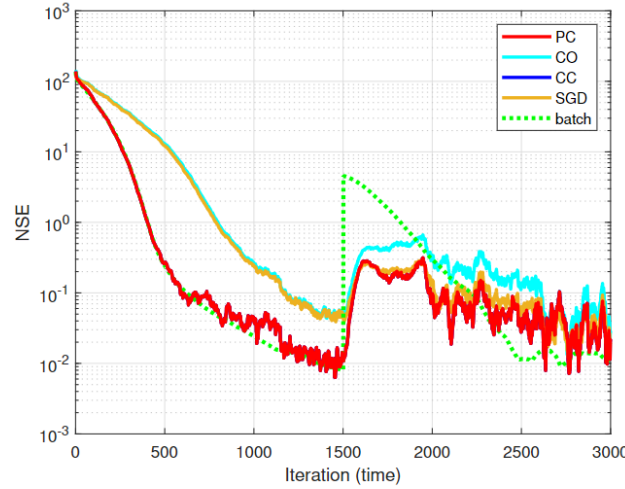
(at time t)	next time-step function prediction			
<b>Prediction</b>	$\mathbf{s}_{t+1 t}^* := \operatorname{argmin}_{\mathbf{s}} \hat{F}(\mathbf{s}; t+1).$	$\longrightarrow$	$\hat{\mathbf{s}}^{p+1} = \hat{\mathcal{T}}\hat{\mathbf{s}}^p, \quad p = 0, 1, \dots, P-1$	$\longrightarrow \hat{\mathbf{s}}_{t+1 t} = \hat{\mathbf{s}}^P$
(at time t+1)				
<b>Correction</b>	$\mathbf{s}_{t+1}^* := \operatorname{argmin}_{\mathbf{s}} F(\mathbf{s}; t+1).$	$\longrightarrow$	$\hat{\mathbf{s}}^{c+1} = \mathcal{T}\hat{\mathbf{s}}^c, \quad c = 0, 1, \dots, C-1$	$\longrightarrow \hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}^C$

# Synthetic Experiment

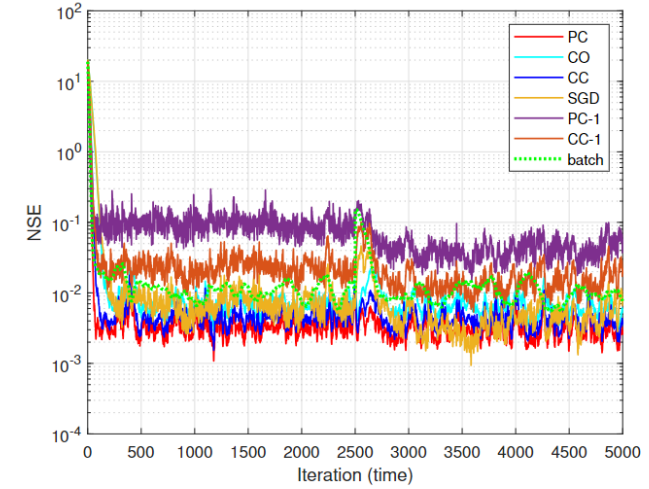
$$\text{NSE}(\hat{\mathbf{s}}_t, \mathbf{s}_t^\star) = \frac{\|\hat{\mathbf{s}}_t - \mathbf{s}_t^\star\|_2^2}{\|\mathbf{s}_t^\star\|_2^2}.$$



(a) **TV-GGM.**  $N=18$ ,  $\alpha = \beta = 10^{-2}$ ,  
 $\gamma = 99.9 \times 10^{-2}$



(b) **TV-SEM.**  $N=28$ ,  $\alpha = \beta = 0.1 \times 10^{-2}$ ,  
 $\lambda = 0.5$ ,  $\gamma = 99 \times 10^{-2}$



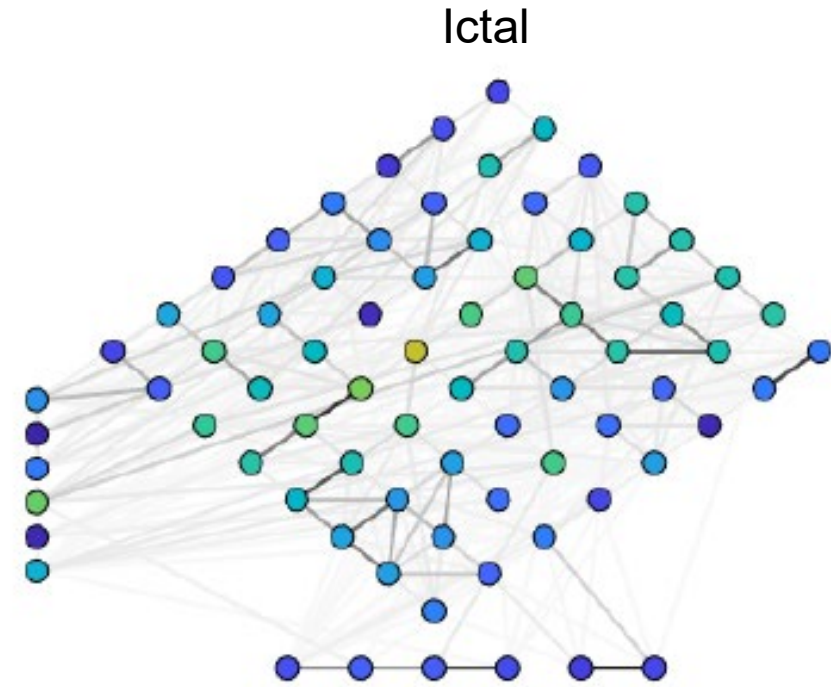
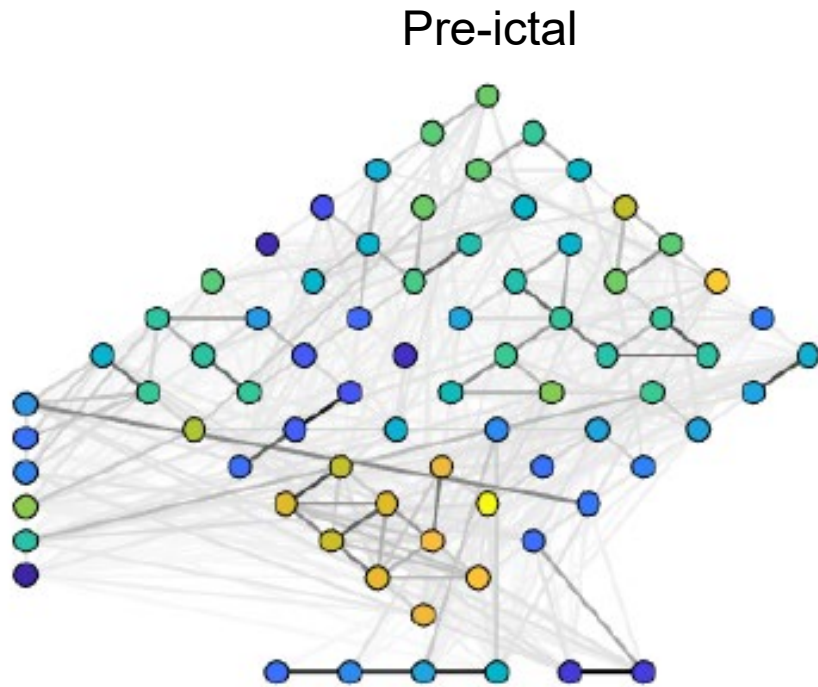
(c) **TV-SBM.**  $N=28$ ,  $\alpha = \beta = 0.1 \times 10^{-2}$ ,  
 $\lambda_1 = 10$ ,  $\lambda_2 = 10$ ,  $\gamma = 99 \times 10^{-2}$

TABLE I: Average time (expressed in seconds) required to compute the PC and the CVX solution at each time instant.

	PC	CVX
TV-GGM	$0.110 \times 10^{-2}$	3.6
TV-SEM	$0.824 \times 10^{-2}$	2.0
TV-SBM	$0.023 \times 10^{-2}$	3.6

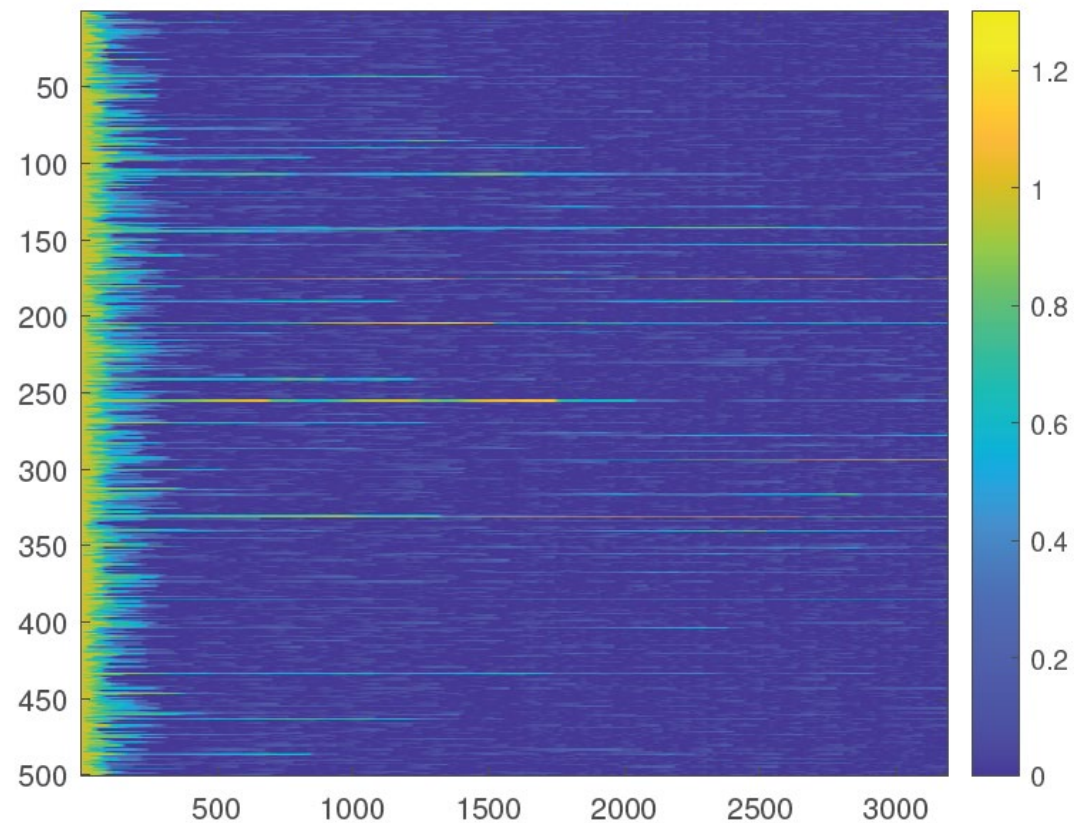
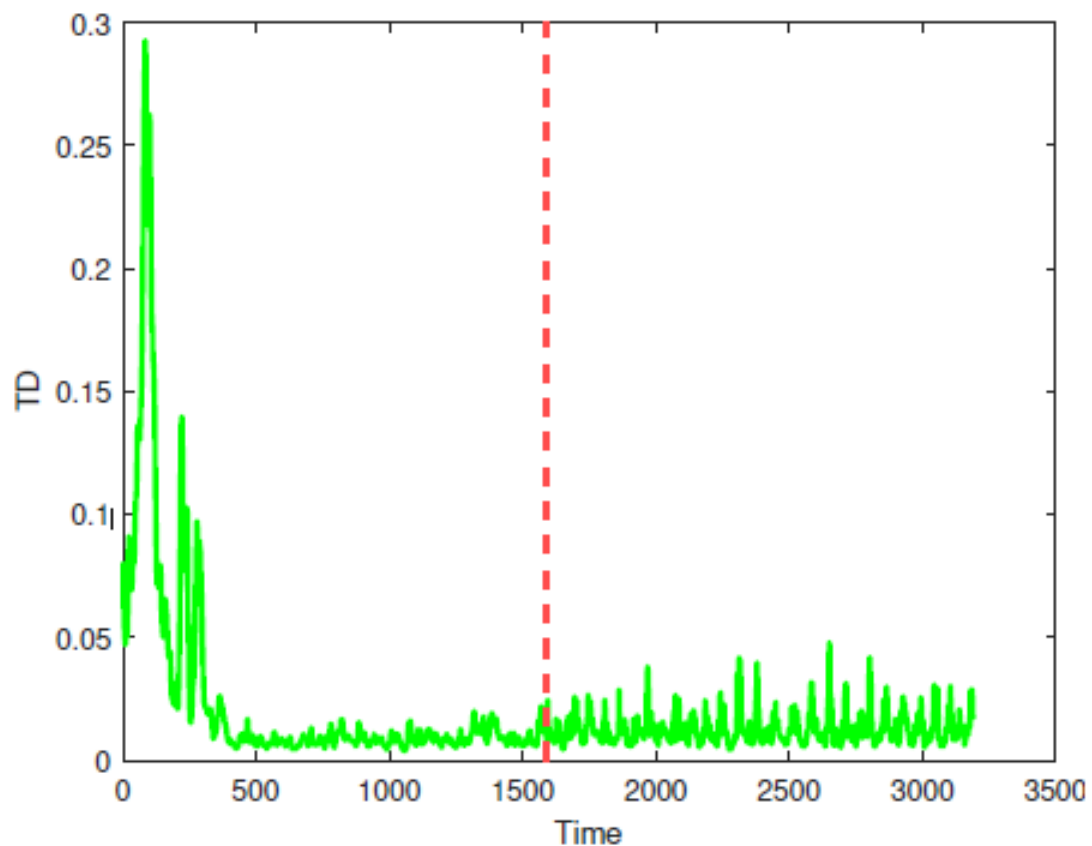
# Epilepsy Study

$N = 76$  electrodes



# Epilepsy Study

$$TD(t) := \|\hat{\mathbf{s}}_t - \hat{\mathbf{s}}_{t-1}\|_2$$



# Q&A

Alberto Natali  
[a.natali@tudelft.nl](mailto:a.natali@tudelft.nl)