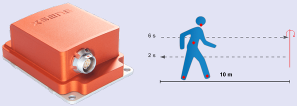
# Event-based representations for Electromagnetic Brain Signals

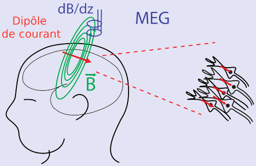
Thomas Moreau
INRIA Saclay - MIND Team

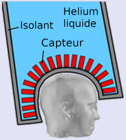
**Physiological signals:** Measurements of the body's functions and processes using physical sensors.




Gait Analysis

Occulography





General Anesthesia (GA)

Smart Watch


Magneto/Electro EncephaloGram (M/EEG)

# Physiological signals: applications

**Use cases:**

- Healthcare – early diagnosis, monitoring, treatment
- Interfaces – Brain-Computer Interface (BCI), prosthetics, ...
- Neurosciences – understanding the brain functions

**Corresponding ML tasks:**

- Full signal – classification, regression, clustering
- Sequence to sequence – non-invasive monitoring, forecasting
- Event-based – event/anomaly/change point detection and prediction

# Physiological signals: challenges

- **Machine Learning:**
  Multiple sources of variability, low-labeled data regime, complex evaluation
- **Signal Processing:**
  High-dimensional data, underlying topology, require domain expertise
- **Medical:**
  Complex evaluation, unclear labels, ethical and societal impact
- **Software:**
  Many standards/formats, demanding computational resources

# Physiological signals: challenges

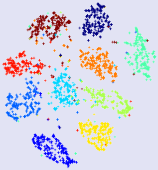
- **Machine Learning:**
  Multiple sources of variability, low-labeled data regime, complex evaluation
- **Signal Processing:**
  High-dimensional data, underlying topology, require domain expertise
- **Medical:**
  Complex evaluation, unclear labels, ethical and societal impact
- **Software:**
  Many standards/formats, demanding computational resources

$\Rightarrow$ This calls for unsupervised learning methods
to characterize the signals and events distribution

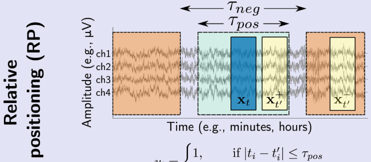
# Unsupervised and self-supervised learning

Learning from data without explicit supervision

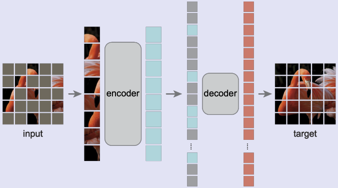
[Macqueen 1967]

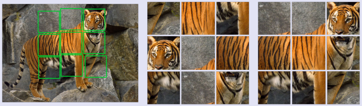


[Banville et al. 2019]




$$y_i = \begin{cases} 1, & \text{if } |t_i - t'_i| \leq \tau_{pos} \\ \text{-1}, & \text{if } |t_i - t'_i| > \tau_{neg} \end{cases}$$



[He et al. 2022]

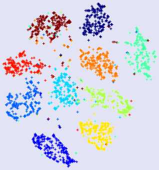
[Noroozi and Favaro 2017]

. . .

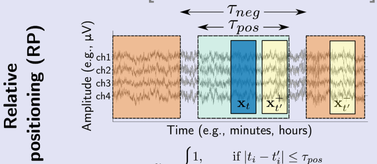
# Unsupervised and self-supervised learning

Learning from data without explicit supervision

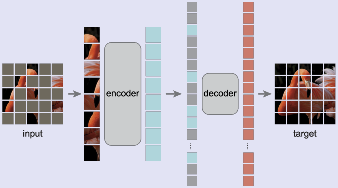
[Macqueen 1967]



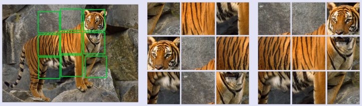
[Banville et al. 2019]




$$y_i = \begin{cases} 1, & \text{if } |t_i - t'_i| \leq \tau_{pos} \\ \text{-}1, & \text{if } |t_i - t'_i| > \tau_{neg} \end{cases}$$




[He et al. 2022]

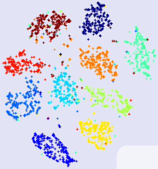


[Noroozi and Favaro 2017]

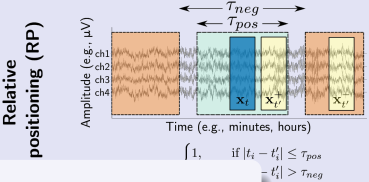
⇒ Despise successes for Image and Text, limited success for time-series

Learning from data without explicit supervision

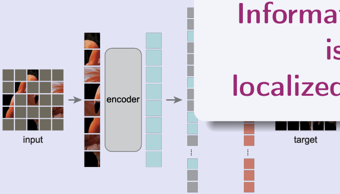
[Macqueen 1967]

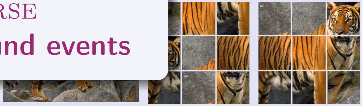


[Banville et al. 2019]




**Information of interest**
**is** SPARSE
**localized around events**



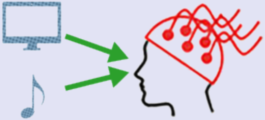



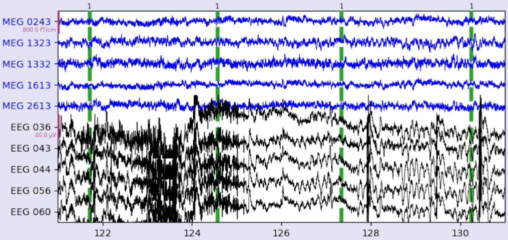
[He et al. 2022]

[Noroozi and Favaro 2017]

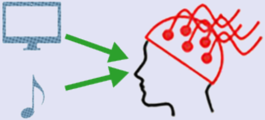
$\Rightarrow$ Despise successes for Image and Text, limited success for time-series

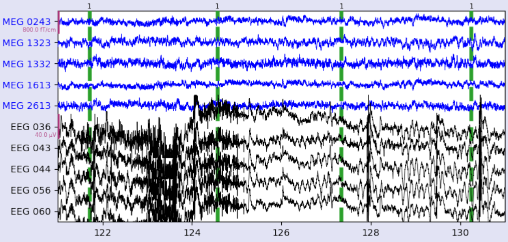
# Event-based processing: the case of M/EEG signals

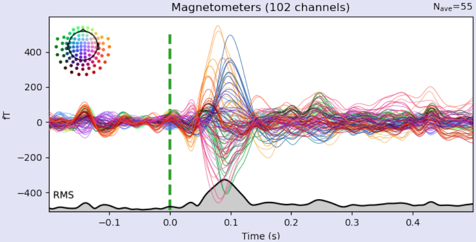




# Event-based processing: the case of M/EEG signals

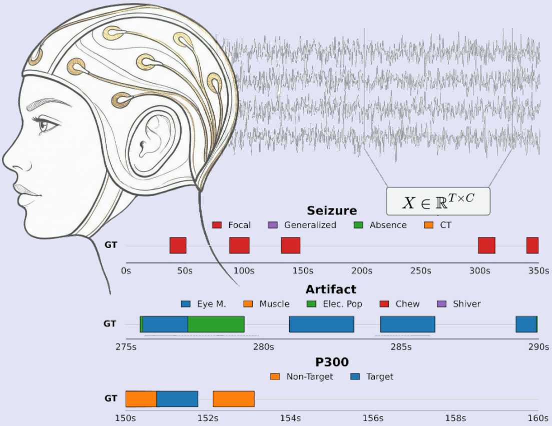




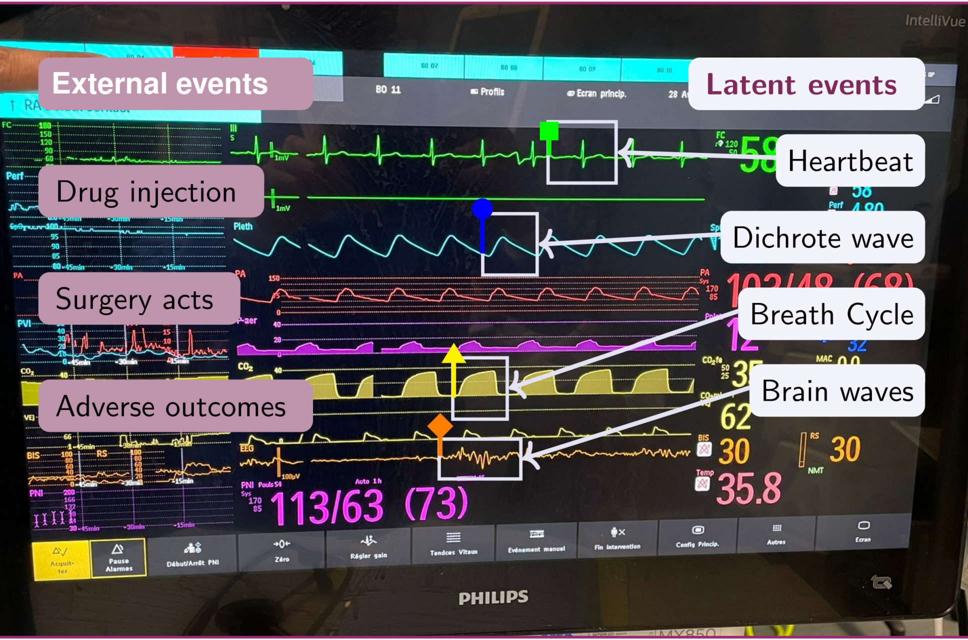

$\Rightarrow$ *Analysis:* average effect after a stimuli (external event)

# Event-based processing: M/EEG decoding [Levy et al. (upcoming)]




$\Rightarrow$ *Decoding:* the signal building blocks are events.

# Event-based processing: the case of General Anesthesia

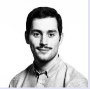
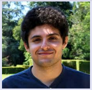


# Event-based processing

**Focus of my research for a few years:**

- Unsupervised event detection in signals
- Modeling events distributions
- End-to-end frameworks for event-based learning

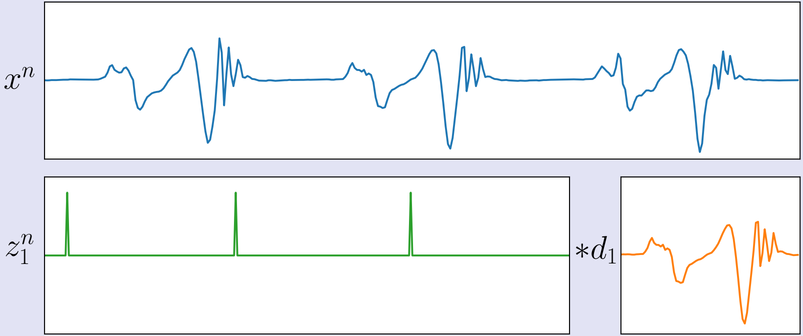
# Convolutional Dictionary Learning
# for unsupervised event detection



## References


- Dupré la Tour, T., **TM**, Jas, M., and Gramfort, A. (2018). Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals. In *NeurIPS*
- Yehya, J., Benbakoura, M., Allain, C., Malézieux, B., Kowalski, M., and **TM** (2025). RoseCDL: Robust and scalable convolutional dictionary learning for rare-event detection. Preprint

**Key idea**: find recurrent patterns and their localization




$$\boxed{x^n}[t] = \sum_{k=1}^{K} (\boxed{z_k^n} * \boxed{d_k})[t] + \varepsilon[t]$$

For a set of $N$ univariate signals $\boldsymbol{x}$, solve

$$\min_{\boldsymbol{D};\|\boldsymbol{D}_k\|_2\leq 1} \frac{1}{N}\sum_{\boldsymbol{x}}\left[\min_{\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{x} - \sum_{k=1}^{K} \boldsymbol{z}_k * \boldsymbol{D}_k\|_2^2 + \lambda\|\boldsymbol{z}\|_1\right]$$

This problem a problem solved with alternate minimization (AM)

1. Solve for each $\boldsymbol{z}$ with a fixed $\boldsymbol{D}$
2. Update $\boldsymbol{D}$ with fixed $\boldsymbol{z}$ for the $\boldsymbol{x}$ considered

# Convolutional Dictionary Learning [Grosse et al. 2007]

For a set of $N$ univariate signals $\boldsymbol{x}$, solve

$$\min_{\boldsymbol{D};\|\boldsymbol{D}_k\|_2\le 1} \frac{1}{N}\sum_{\boldsymbol{x}}\left[\min_{\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{x}-\sum_{k=1}^{K}\boldsymbol{z}_k * \boldsymbol{D}_k\|_2^2 + \lambda\|\boldsymbol{z}\|_1\right]$$

This problem a problem solved with alternate minimization (AM)

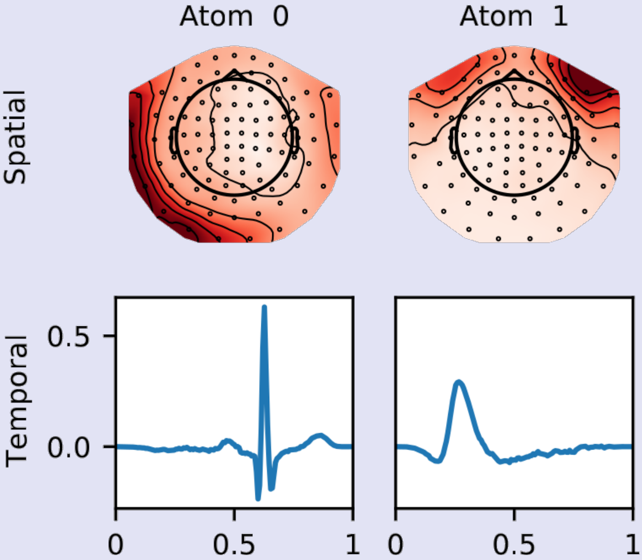
1. Solve for each $\boldsymbol{z}$ with a fixed $\boldsymbol{D}$
2. Update $\boldsymbol{D}$ with fixed $\boldsymbol{z}$ for the $\boldsymbol{x}$ considered

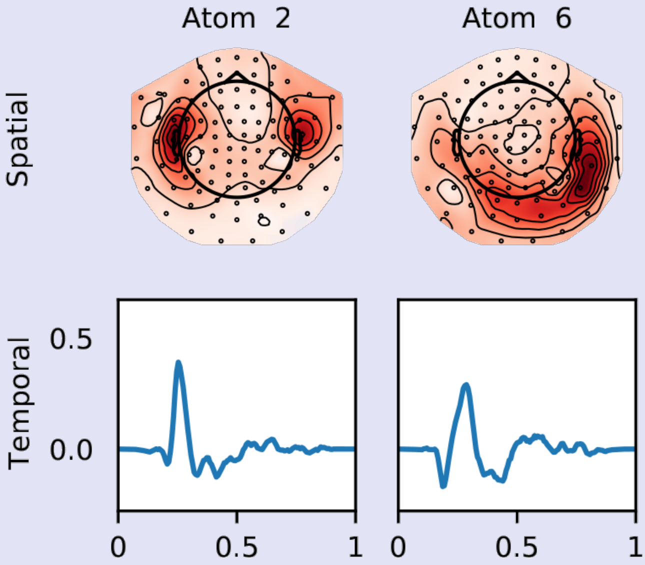
We proposed rank-1 extension [Dupré la Tour et al. 2018]

$$\boldsymbol{D}_k = u_k v_k^\top$$

where $u_k$ captures the temporal pattern and $v_k$ the spatial pattern.

# Learned atoms – Artefacts [Dupré la Tour et al. 2018]




Atom 2
Atom 6
Spatial
Temporal
0.5
0.0
0
0.5
1
0
0.5
1

# Convolutional Dictionary Learning for event detection

CDL is a powerful tool for unsupervised event detection with clear patterns

# Convolutional Dictionary Learning for event detection

CDL is a powerful tool for unsupervised event detection with clear patterns

**Challenges:**

- Scalability issues with long signals and large datasets
- Sensitive to artifacts and outliers
- Analysis of the learned atoms requires expert knowledge
- Some events have unclear patterns/low occurrence

For a set of $N$ signals $\boldsymbol{x}$, solve

$$\min_{\boldsymbol{D};\|\boldsymbol{D}_k\|_2 \leq 1} \frac{1}{N} \sum_{\boldsymbol{x}} \left[ \min_{\boldsymbol{z}} \frac{1}{2} \|\boldsymbol{x} - \sum_{k=1}^{K} \boldsymbol{z}_k * \boldsymbol{D}_k\|_2^2 + \lambda \|\boldsymbol{z}\|_1 \right]$$

For a **population** of signals $\boldsymbol{x}$, solve

$$\min_{D; \|D_k\|_2 \leq 1} \mathbb{E}_{\boldsymbol{x}} \sum_{\boldsymbol{x}} \left[ \min_{\boldsymbol{z}} \frac{1}{2} \|\boldsymbol{x} - \sum_{k=1}^{K} \boldsymbol{z}_k * D_k\|_2^2 + \lambda \|\boldsymbol{z}\|_1 \right]$$

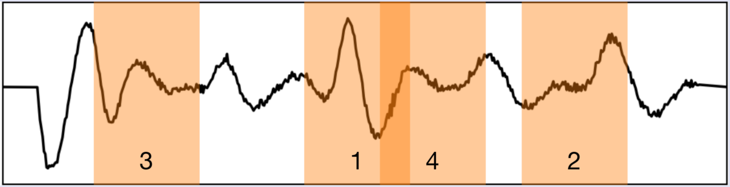
$\Rightarrow$ Shift to a population point of view

For a **population** of signals $\boldsymbol{x}$, solve

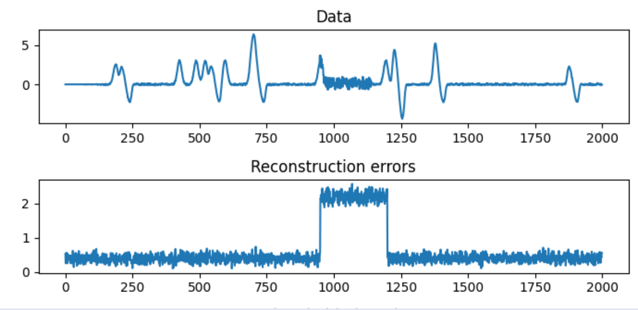
$$\min_{D;\|D_k\|_2\leq 1} \mathbb{E}_{\boldsymbol{x}} \sum_{\boldsymbol{x}} \left[ \min_{\boldsymbol{z}} \frac{1}{2} \|\boldsymbol{x} - \sum_{k=1}^{K} \boldsymbol{z}_k * D_k\|_2^2 + \lambda \|\boldsymbol{z}\|_1 \right]$$

$\Rightarrow$ Shift to a population point of view

**1. Stochastic optimization for scalability:**

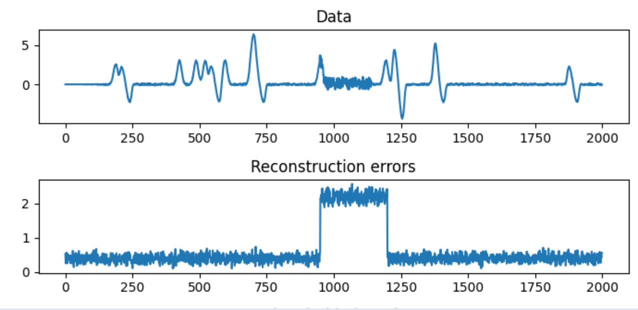


## 2. CDL for outlier detection:




Outlier if its reconstruction error is high compared to the usual one.

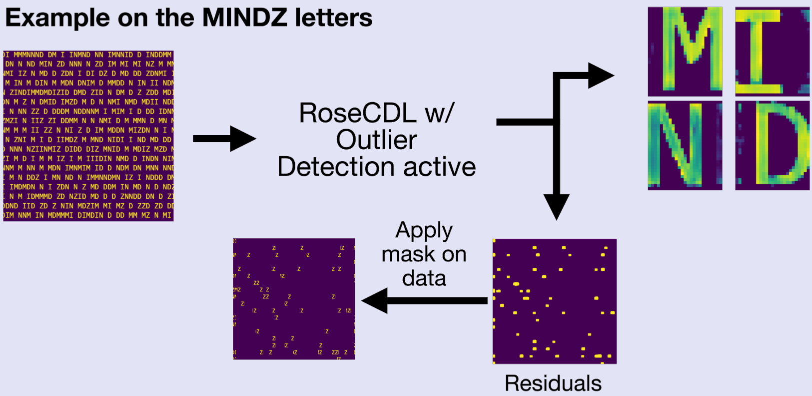
2. **CDL for outlier detection:**

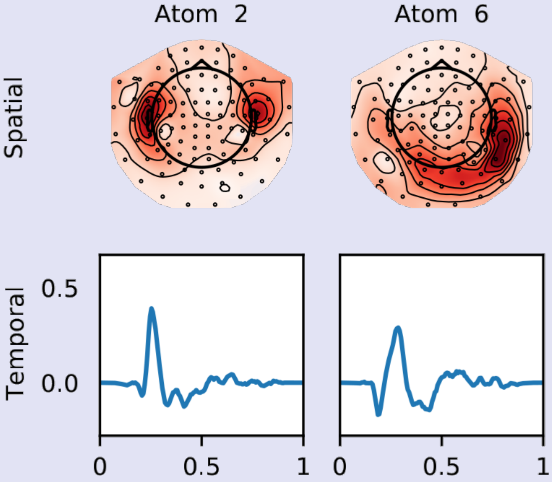



Outlier if its reconstruction error is high compared to the usual one.

3. **Inline outlier detection for robust CDL:** Use this mechanism *inside* the CDL algorithm to improve the learned atoms.

4. CDL for rare event detection:






$\Rightarrow$ Faster and with less preprocessing!

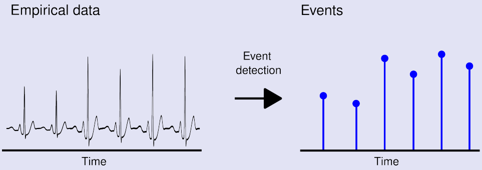
# Modeling event distribution with Point Processes




## References

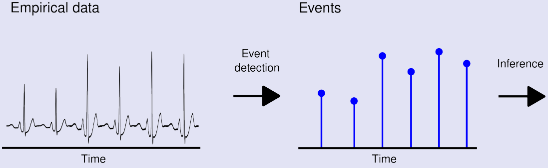

- Staerman, G., Allain, C., Gramfort, A., and **TM** (2023). FaDIn: Fast Discretized Inference for Hawkes Processes with General Parametric Kernels. In *ICML*
- Loison, V., Staerman, G., and **TM** (2025). Unmixing Noise from Hawkes Process to Model Learned Physiological Events. In *AISTATS*

# Modeling event distribution




Event detection discretizes the signal into a stream of events.

# Modeling event distribution




Intensity function:
$\lambda(t; \mathcal{F}_t, \theta)$

Event detection discretizes the signal into a stream of events.

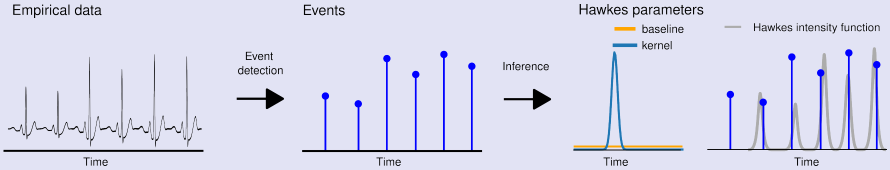
Models for the distribution of events in time: **Point processes**

Characterized by the intensity function $\lambda(t; \mathcal{F}_t, \theta)$

$$\lambda(t; \mathcal{F}_t, \theta) = \lim_{dt \to 0} \frac{P(N(t + dt) - N(t) = 1 | \mathcal{F}_t)}{dt}$$

instant rate of events at time $t$.

# Adapting Hawkes processes for physiological events




Modeling interactions between events: **Hawkes processes** [Hawkes 1971]

$$\lambda(t; \mathcal{F}_t, \theta) = \mu(t) + \sum_{t_k < t} \kappa_\theta(t - t_k)$$

where $\kappa_\theta$ model the interaction between events

# Inference for Hawkes processes

Hawkes process inference consists in minimizing NLL or $\ell_2$ loss.

$$\mathcal{L}(\theta) = \int_0^T \phi(\lambda(t;\theta))dt - \sum_{k=1}^{K} \psi(\lambda(t_k;\theta))$$

with $\phi, \psi$ simple functions. Computational bottleneck is to evaluate $\lambda(t_k;\theta)$

# Inference for Hawkes processes

Hawkes process inference consists in minimizing NLL or $\ell_2$ loss.

$$\mathcal{L}(\theta) = \int_0^T \phi(\lambda(t;\theta))dt - \sum_{k=1}^{K} \psi(\lambda(t_k;\theta))$$

with $\phi, \psi$ simple functions. Computational bottleneck is to evaluate $\lambda(t_k;\theta)$

- Usually efficient due to markovian properties of the exponential kernel:
  *"Easy to compute $\lambda(t+\Delta t;\theta)$ from $\lambda(t;\theta)$"*

$\Rightarrow$ *Complexity to compute $\lambda(t_k;\theta)$ linear $\mathcal{O}(K)$ in the number of events $K$*

Hawkes process inference consists in minimizing NLL or $\ell_2$ loss.

$$\mathcal{L}(\theta) = \int_0^T \phi(\lambda(t;\theta))dt - \sum_{k=1}^{K} \psi(\lambda(t_k;\theta))$$

with $\phi, \psi$ simple functions. Computational bottleneck is to evaluate $\lambda(t_k;\theta)$

- Usually efficient due to markovian properties of the exponential kernel:
  *"Easy to compute $\lambda(t+\Delta t;\theta)$ from $\lambda(t;\theta)$"*

$\Rightarrow$ *Complexity to compute $\lambda(t_k;\theta)$ linear $\mathcal{O}(K)$ in the number of events $K$*

- With general kernels, naive computations of $\lambda(t_k;\theta)$ are in $\mathcal{O}\left(K^2\right)$

If we consider the $\ell_2$ loss and discretize the time, we can rewrite the inference loss as:

$$L(\theta) = \sum_{t=0}^{T} \frac{1}{2} \| z[t] - (z * \kappa_\theta)[t] - \mu \|_2^2$$

where $z$ is a *sparse activation vector*, with $z[t] = 1$ if $t \in \{t_k\}$, 0 otherwise.

# Efficient inference with FaDIn

If we consider the $\ell_2$ loss and discretize the time, we can rewrite the inference loss as:

$$L(\theta) = \sum_{t=0}^{T} \frac{1}{2} \|z[t] - (z * \kappa_\theta)[t] - \mu\|_2^2$$

where $z$ is a *sparse activation vector*, with $z[t] = 1$ if $t \in \{t_k\}$, 0 otherwise.

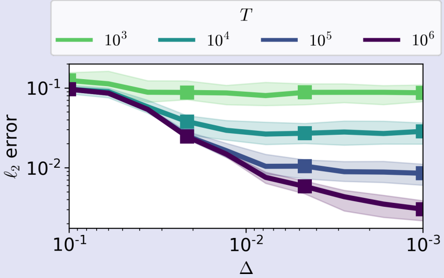
$\Rightarrow$ This loss is similar to dictionary updates in CDL

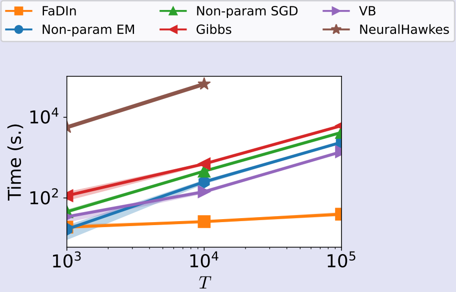
Can be minimized efficiently when the kernel has a finite support using precomputations

# Efficient inference with FaDIn

**FaDIn:** Fast Discretized Inference for finite support kernels

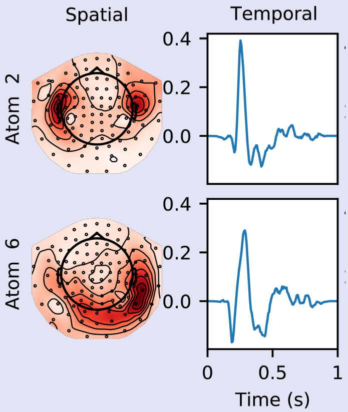
- Bias due to discretization goes to 0 as the discretization is refined



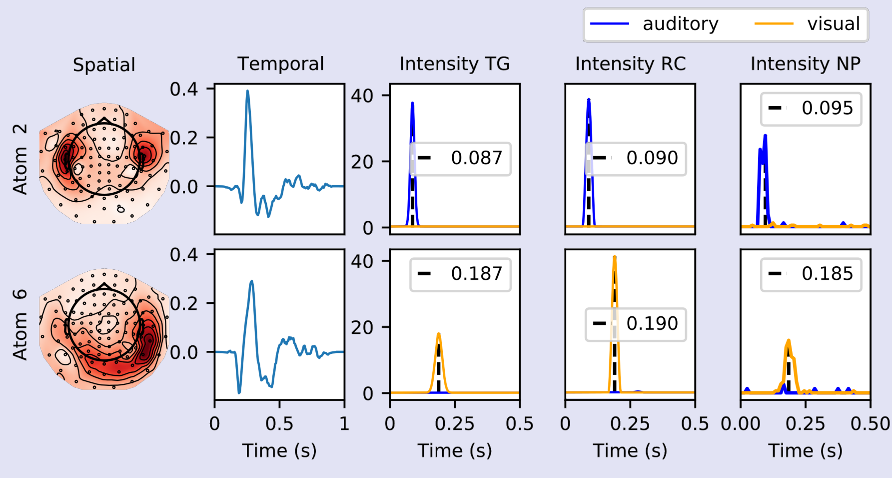

- Optimization complexity is independent of the number of events $K$

# Characterizing evoked responses in M/EEG signals

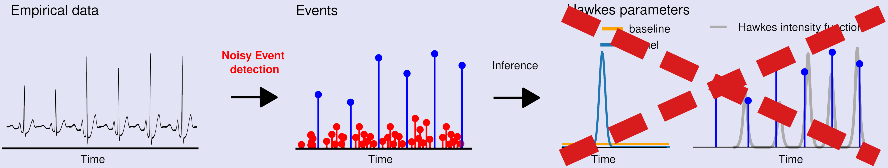



CDL

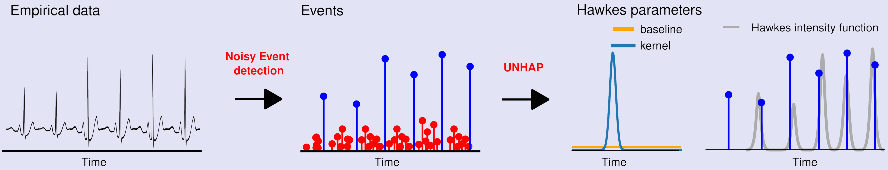
# Characterizing evoked responses in M/EEG signals




**CDL** **FaDIn**




In most cases, the events we are looking for are structured, while the spurious detections are random.




In most cases, the events we are looking for are structured, while the spurious detections are random.

**UNHaP Goal:** Classify events based on their temporal distribution.

- The noisy detections are modeled as a Poisson process $\mathcal{F}_T^0$
- The structured events are modeled as a Hawkes process $\mathcal{F}_T^1$

We observe the mixture $\mathcal{F}_T = \mathcal{F}_T^0 \cup \mathcal{F}_T^1$

# Deriving the mixture model

If we are given $Y_k \in \{0, 1\}$ encoding whether the event $t_k \in \mathcal{F}_T^0$ or $\mathcal{F}_T^1$, we have:

$$\lambda^1(t) = \mu^1 + \sum_{t_k \in \mathcal{F}_T} Y_k \phi(t - t_k; \theta^1)$$

and we can rewrite the $\ell_2$-risk of the model as:

$$\begin{aligned}\mathcal{L}(\theta, \mathrm{Y}; \mathcal{F}_T) = \int_0^T & \lambda^0(t;\theta)^2 + \lambda^1(t;\theta)^2 dt \\ & - 2 \sum_{t_k \in \mathcal{F}_T} (1 - Y_k)\lambda^0(t_k;\theta) + Y_k \lambda^1(t_k;\theta)\end{aligned}$$

# Deriving the mixture model

If we are given $Y_k \in \{0,1\}$ encoding whether the event $t_k \in \mathcal{F}_T^0$ or $\mathcal{F}_T^1$, we have:

$$\lambda^1(t) = \mu^1 + \sum_{t_k \in \mathcal{F}_T} Y_k \phi(t - t_k; \theta^1)$$

and we can rewrite the $\ell_2$-risk of the model as:

$$\begin{aligned}\mathcal{L}(\theta, \mathrm{Y}; \mathcal{F}_T) = \int_0^T \lambda^0(t;\theta)^2 + \lambda^1(t;\theta)^2 dt \\ - 2 \sum_{t_k \in \mathcal{F}_T} (1 - Y_k)\lambda^0(t_k;\theta) + Y_k \lambda^1(t_k;\theta)\end{aligned}$$

$\Rightarrow$ Direct resolution with the EM algorithm is not possible

# Classification EM

We use a mean-field approximation to the posterior distribution of $Y_k$ where we compute $\rho_k$, the probability that event $k$ is from $\mathcal{F}^1_T$:

$$p(\mathrm{Y}; \mathcal{F}_T) \approx \prod_{k=1}^{K} q(Y_k; \rho_k)$$

# Classification EM

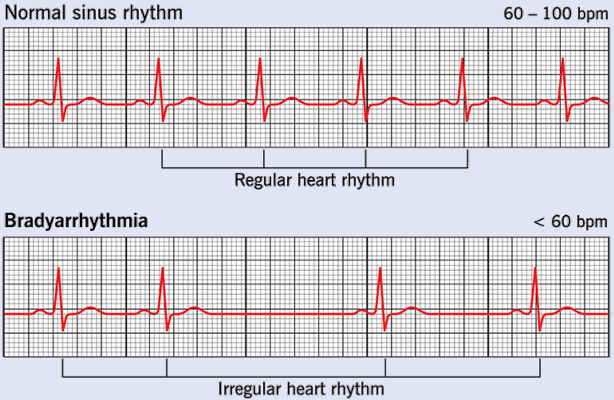
We use a mean-field approximation to the posterior distribution of $Y_k$ where we compute $\rho_k$, the probability that event $k$ is from $\mathcal{F}_T^1$:

$$p(\mathrm{Y}; \mathcal{F}_T) \approx \prod_{k=1}^{K} q(Y_k; \rho_k)$$

**Classification EM:**

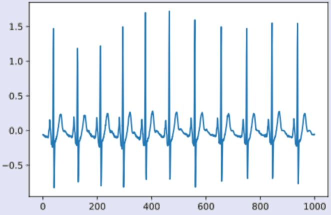
1. **E-step:** minimize $\mathcal{L}(\theta, \rho; \mathcal{F}_T)$ w.r.t $\rho$
2. **C-step:** for each event, assign $Y_k$ based on the value of $\rho_k$
3. **M-step:** minimize $\mathcal{L}(\theta, \mathrm{Y}; \mathcal{F}_T)$ with respect to $\theta$

# Event-based processing: the case of ECG signals

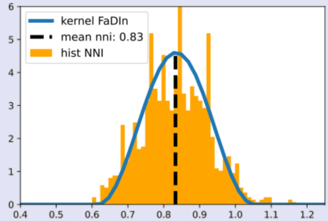



$\Rightarrow$ *Diagnosis:* Characterizing heartbeat shape and regularity

# Characterizing heart rate from ECG




CDL + UNHaP




| **Model** | CDL + FaDIn | **CDL + UNHaP** | pyHRV | Neurokit |
|---|---|---|---|---|
| **error** | 2.57 (0.26-40.4) | **0.27** (0.14-0.84) | 0.81 (0.16-2.08) | 0.54 (0.51-0.61) |

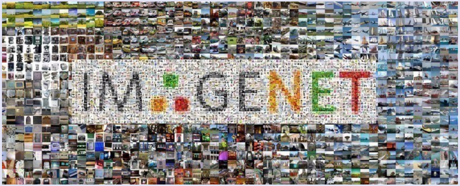
# Reproducible method comparison with Benchopt

 . . .

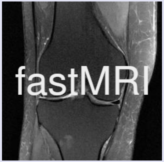
## References


- **TM**, Massias, M., Gramfort, A., Ablin, P., Bannier, P.-A., Charlier, B., Dagréou, M., la Tour, T. D., Durif, G., Dantas, C. F., Klopfenstein, Q., Larsson, J., Lai, E., Lefort, T., Malézieux, B., Moufad, B., Nguyen, B. T., Rakotomamonjy, A., Ramzi, Z., Salmon, J., and Vaiter, S. (2022). Benchopt: Reproducible, efficient and collaborative optimization benchmarks. In *NeurIPS*

## Benchmarks fueled AI progress

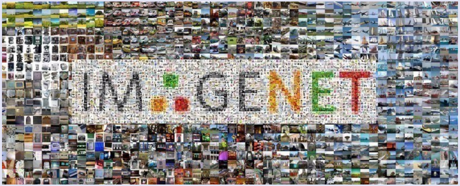


GLUE

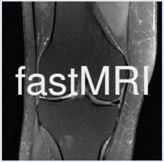


# Benchmarks and reproducibility

## Benchmarks fueled AI progress



GLUE




**Do we really need Foundation Models for multi-step-ahead Epidemic Forecasting?**

**Position: Quo Vadis, Unsupervised Time Series Anomaly Detection?**

M. Saquib Sarfraz[1,2] Mei-Yen Chen[1] Lukas Layer[1] Kunyu Peng[2] Marios Koulakis[2]

PNAS RESEARCH ARTICLE | COMPUTER SCIENCES OPEN ACCESS

**Implicit data crimes: Machine learning bias arising from misuse of public data**

Efrat Shimron[a,1], Jonathan I. Tamir[b,c,d], Ke Wang[a], and Michael Lustig[a]

**Descending through a Crowded Valley — Benchmarking Deep Learning Optimizers**

Robin M. Schmidt[*1] Frank Schneider[*1] Philipp Hennig[1,2]

## Many fields lack reproducible reference benchmarks!

$\Rightarrow$ Many novel methods but unclear improvements

# Making runnable benchmarks with `benchopt`

`benchopt` provides a framework to organize and run benchmarks

Examples of existing benchmarks:

- **Image Classification (resnet)**
- **Logistic regression**
- **Lasso**
- **ICA**
- **Unsup. Domain Adaptation**
- **Bilevel Optimization**
- **Brain Computer Interface**
- **Many others ...**

# Conclusion

- Event-based processing is a promising approach for M/EEG signals
- Convolutional Dictionary Learning (CDL) is a powerful tool for unsupervised events detection
- PP can be used to model the events distribution and detect anomalies

Some code available online:

**alphacsc** : `https://alphacsc.github.io`

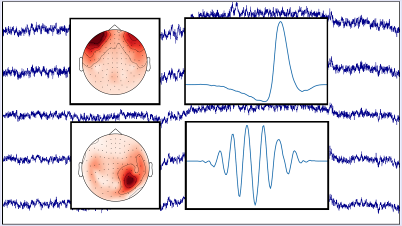
**FaDIn** : `https://mind-inria.github.io/FaDIn/`

**benchopt** : `https://benchopt.github.io`

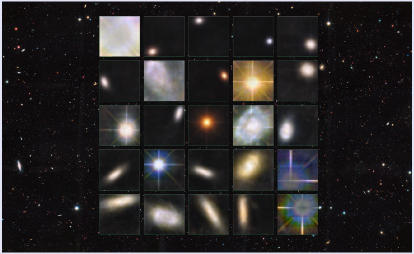
tommoral.github.io @tommoral

# Finding Events in Physical Signals

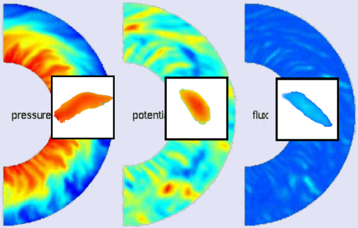

General Anesthesia



Neuroscience (MEG)



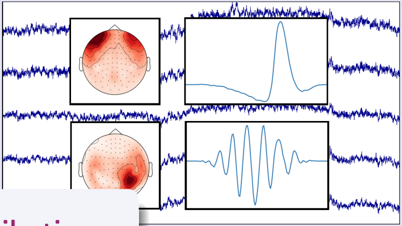
Astronomy




Physics Simulation

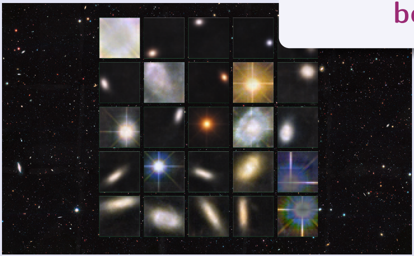
# Finding Events in Physical Signals

General Anesth

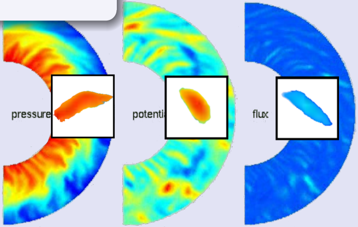


science (MEG)

**Events distribution characterize the signal behavior**



Astronomy




Physics Simulation