Reconstruction of neural activity using kinetic Ising model

Michele Avella (m 2024548), **Elena Leonelli** (m 2028635)
Marika Sartore (m 2017916), **Filippo Ziliotto** (m 2017425)

UNIVERSITÀ DEGLI STUDI DI PADOVA

Table of contents

Introduction

Università degli Studi di Padova

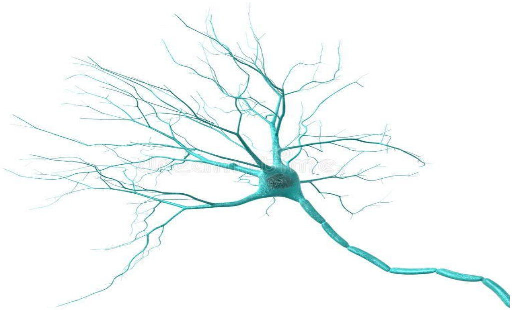
What's the goal?

- Reconstruction of the neuron response given a periodic visual stimulus
- Scalability of the project for thousand of neurons

↓

Is it possible to replicate the neurons signals and categorize their importance?

Zebra fish



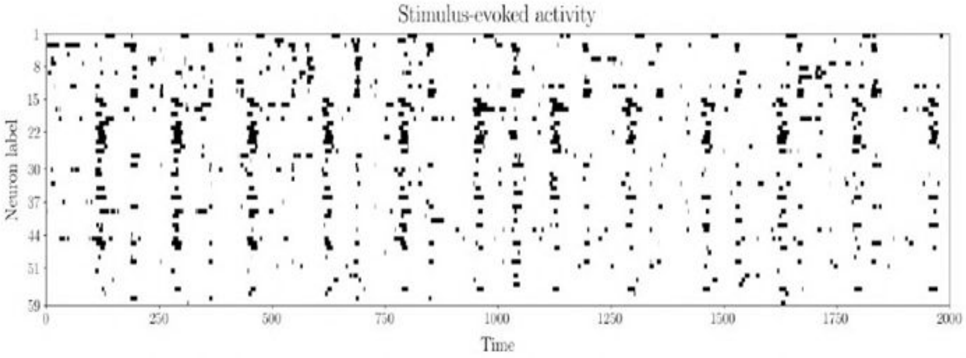
Introduction

UNIVERSITÀ DEGLI STUDI DI PADOVA

Data?

- Real neurons activity over time

- Synthetic generated data



Probability of flipping

$$\boxed{p_{flip}} = \gamma \delta t \frac{1}{2}[1 - s_i(t) tanh H_i(t)]$$

Ising Model

$$H_i(t) = h_i + \sum_{j \neq i} J_{ij} s_j(t)$$

External field

Coupling matrix

To reconstruct the model we need to discretize data

Theoretical Foundations

UNIVERSITÀ DEGLI STUDI DI PADOVA

Theoretical base

Infer the parameters of the (asynchronous) Ising model

$$L = \sum_i \frac{1}{N_{\tau_i}} \sum_{\tau_i} [\, s_i(\tau_i + \delta t)\theta_i(\tau_i) - logcosh\theta_i(\tau_i)]$$

- Spin history ✔
- Updated times ✘

Pseudo-Likelihood

(Unconnected) covariance matrix

$$\begin{cases} C_{ij} = \langle s_i(t) s_j(t) \rangle_t \\ \dot{C}_{ij} = \delta t^{-1} [\langle s_i(t+\delta t) s_j(t) \rangle_t - C_{ij}] \\ L = \sum_i \frac{1}{N_t} \sum_t [h_i s_i(t) + \sum_{i \neq j} J_{ij} \left(\frac{\dot{C}_{ij}}{\gamma} + C_{ij} \right) - logcoshH_i(t)] \end{cases}$$

ref. [2]

Theoretical Foundations

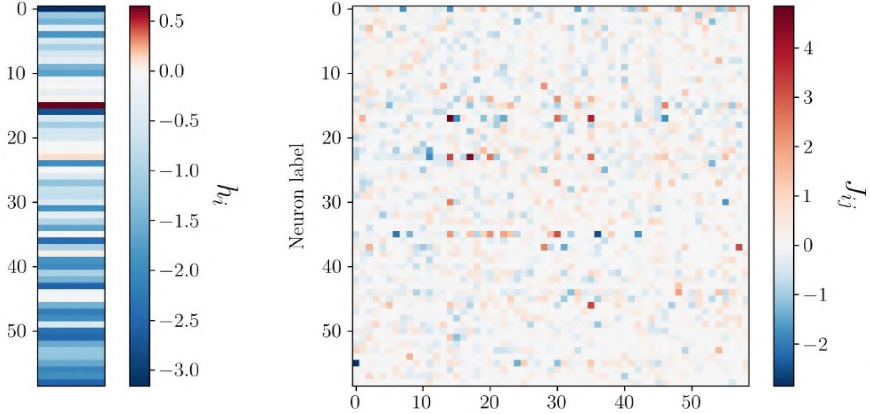
Given the Pseudo-likelihood → ∇L

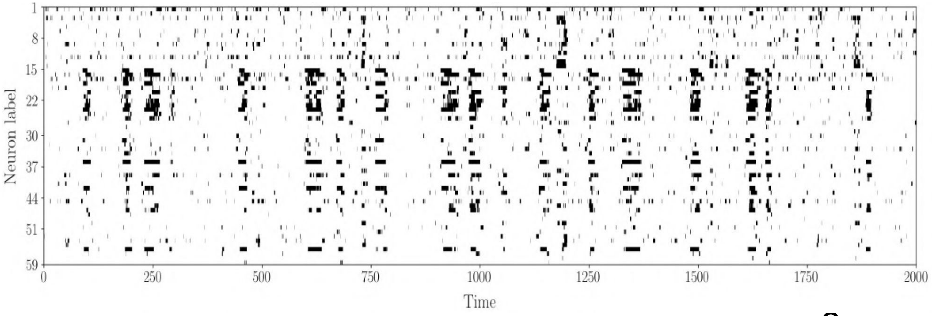
$$\begin{cases} \dfrac{\partial L}{\partial h_i} = \langle s_i(t) - tanh H_i(t) \rangle_t \\ \dfrac{\partial L}{\partial J_{ij}} = \gamma^{-1} \dot{C}_{ij} + C_{ij} - \langle [tanh H_i(t)] s_i(t) \rangle_t \end{cases}$$

Algorithms implemented for maximization:

- GA (mom)
- ADAM
- RMS prop
- NAG

Detailed results later...





Simulation

UNIVERSITÀ DEGLI STUDI DI PADOVA

$$p_{flip} = \gamma \delta t \frac{1}{2}[1 - s_i(t) \tanh H_i(t)]$$

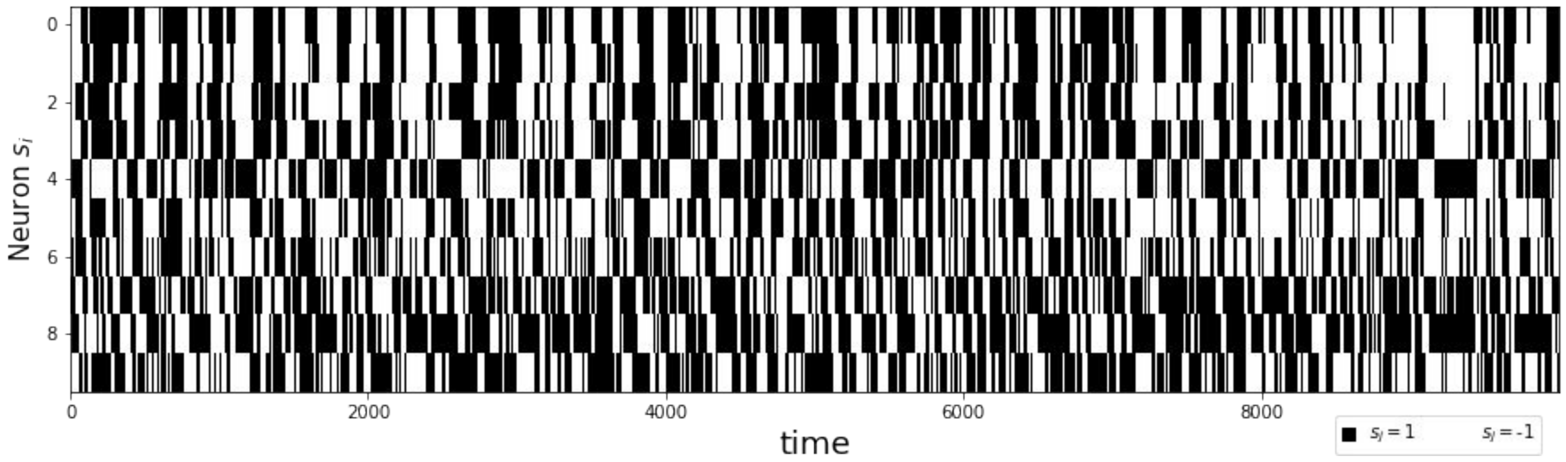
how to infer real-data parameters?

SIMULATION

- ➔ few neurons
- ➔ 100/1000 neurons

```
def simulation(J,h,gamma,steps,dt):

    S = [-np.ones(N)]

    for i in tqdm(range(1,steps)):
        s = np.copy(S[-1])
        H = h + np.dot(J,s)
        p_flip = gamma*dt*0.5*(1-s*np.tanh(H))
        p = np.random.rand(N)
        s[p<p_flip]*=-1
        S.append(s)

    S = np.array(S).T

    return S
```

10 neurons: simulation




```python
J = np.random.normal(0,1,(N,N))
np.fill_diagonal(J,0)
h = np.random.normal(0,1,N)
```

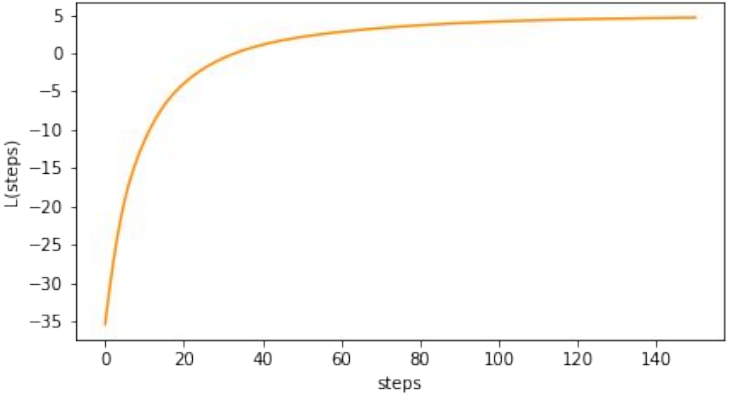
δt = 0.1

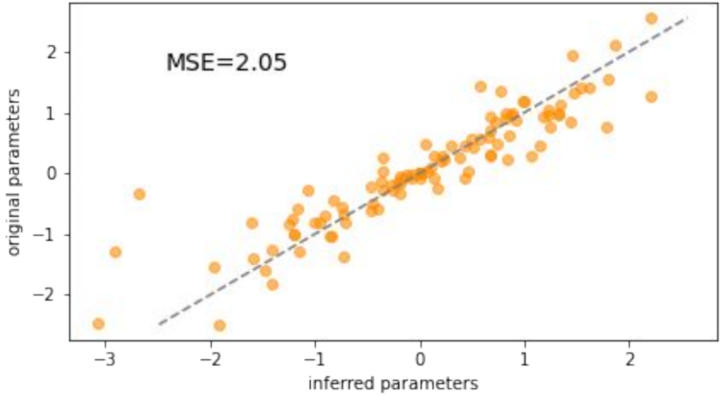
10^6 time steps

gamma = 1

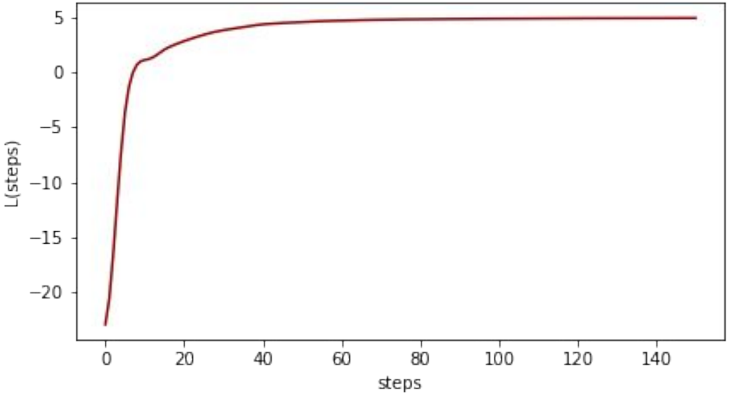
10 neurons: inference

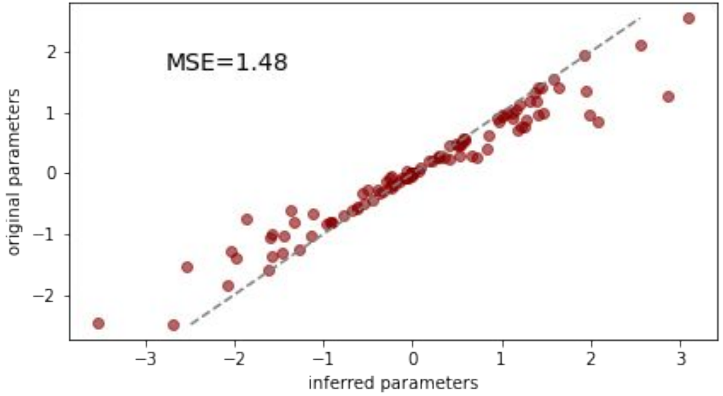
Gradient Ascend



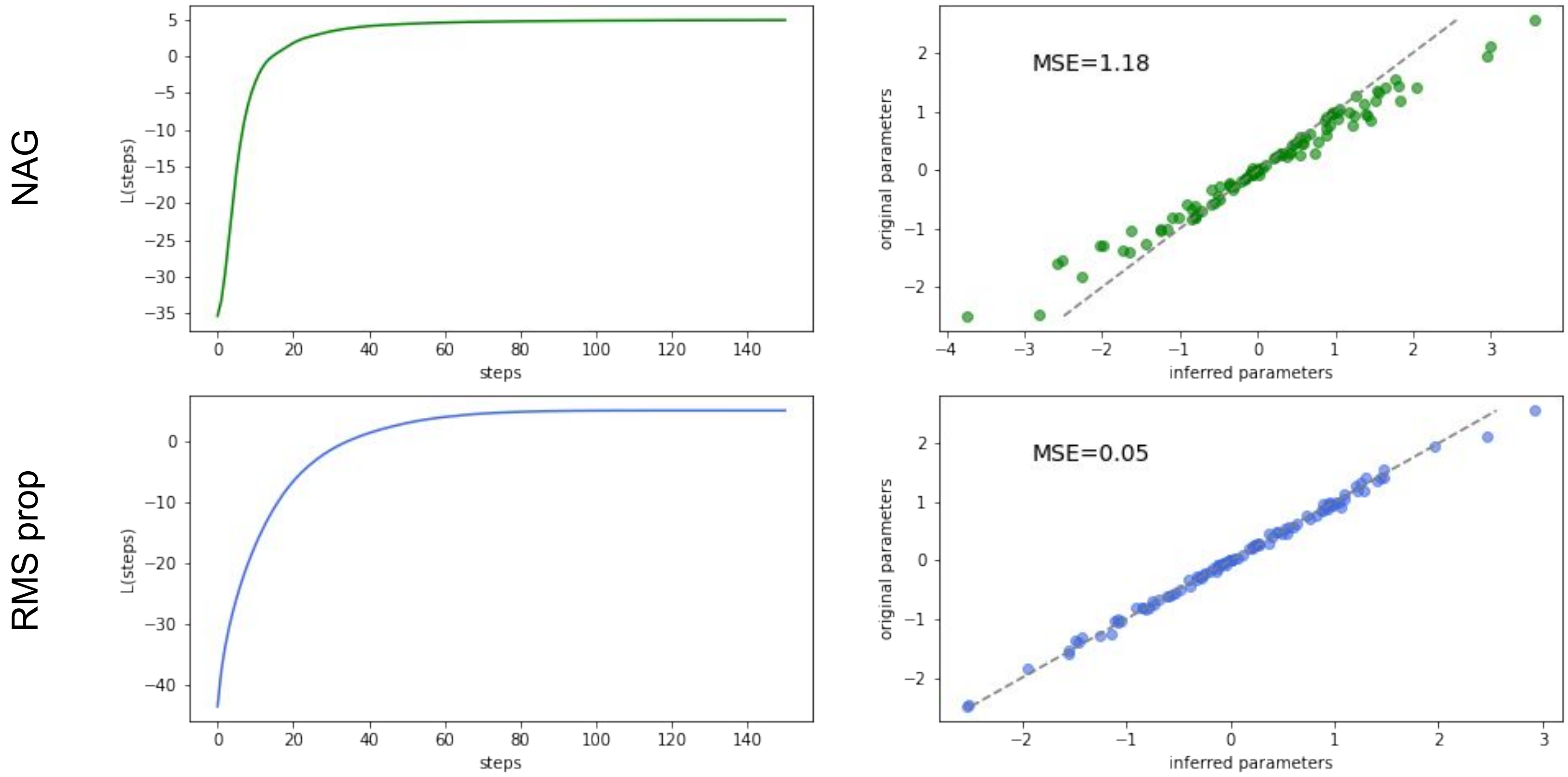


Gradient Ascend with momentum





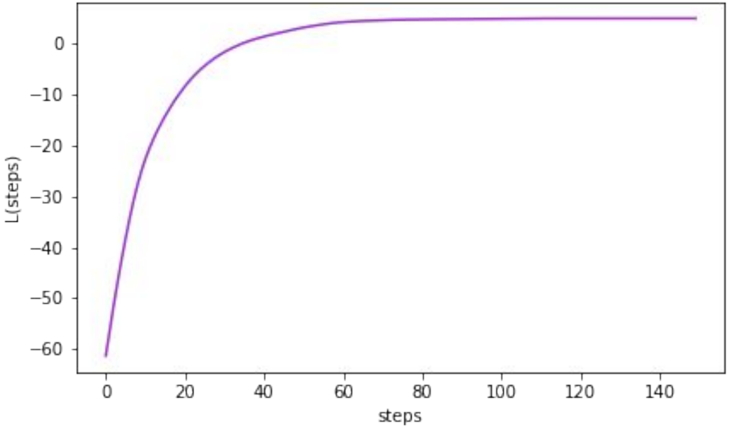
10 neurons: inference

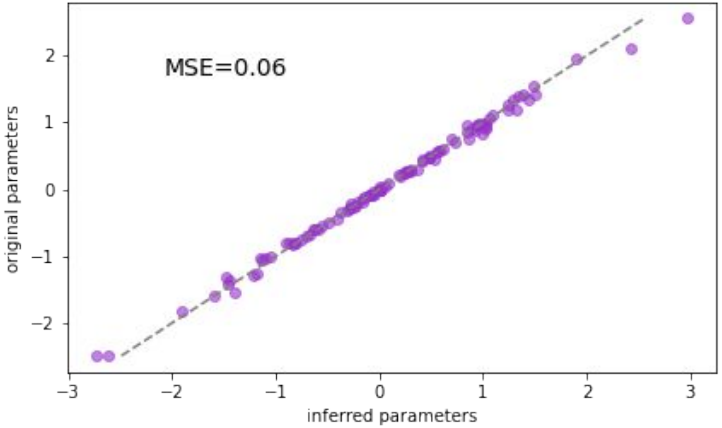


10 neurons: inference

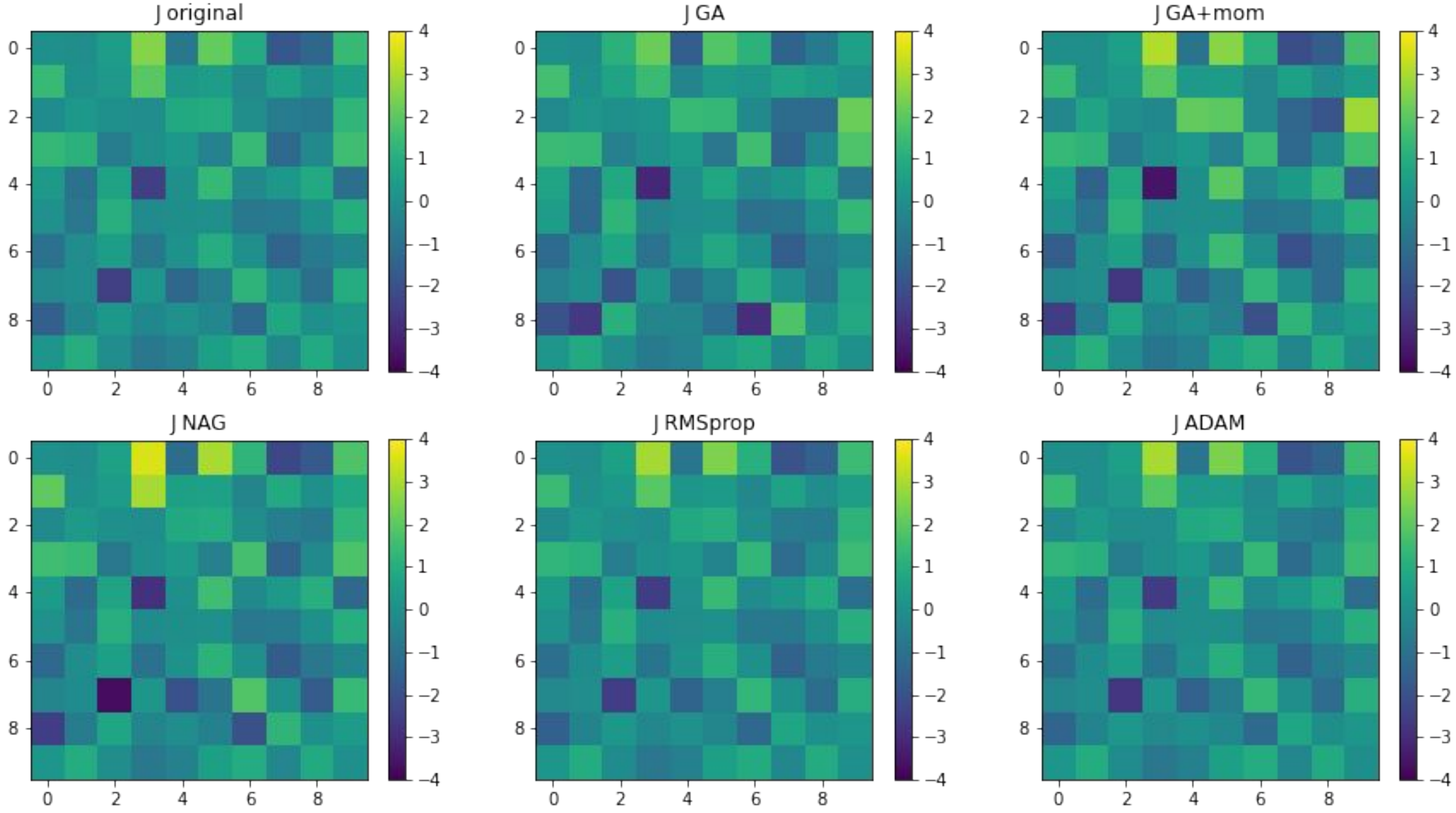
UNIVERSITÀ DEGLI STUDI DI PADOVA

ADAM

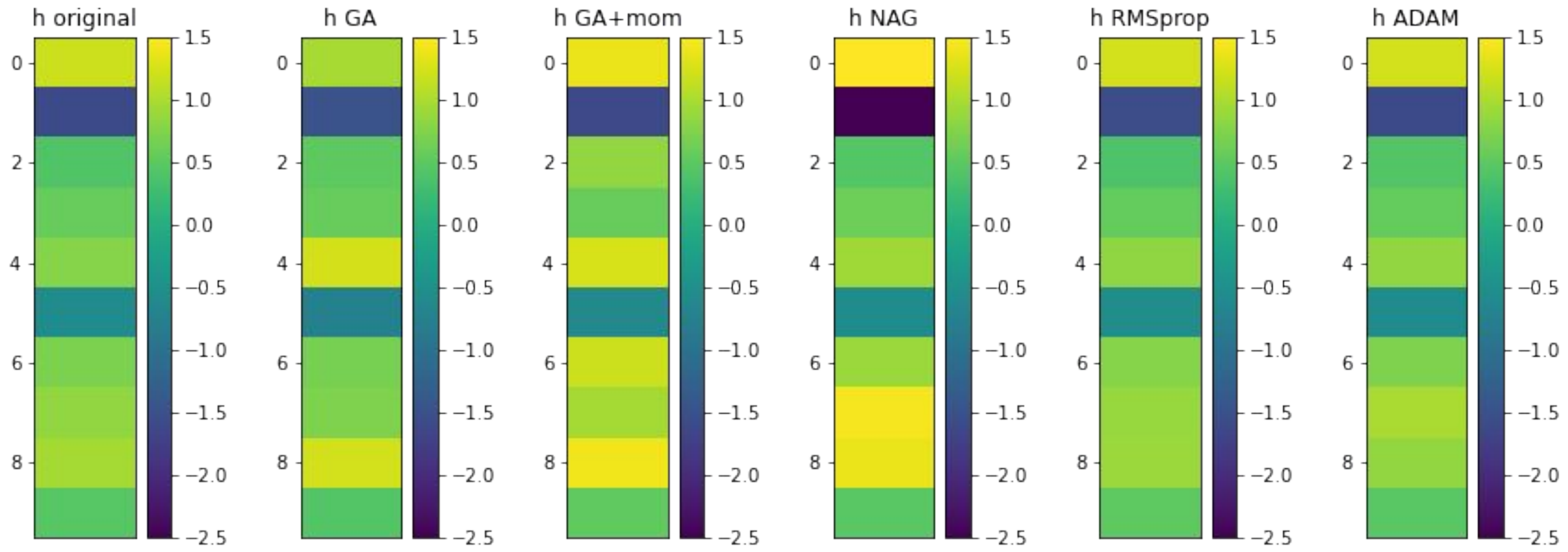




10 neurons: couplings J

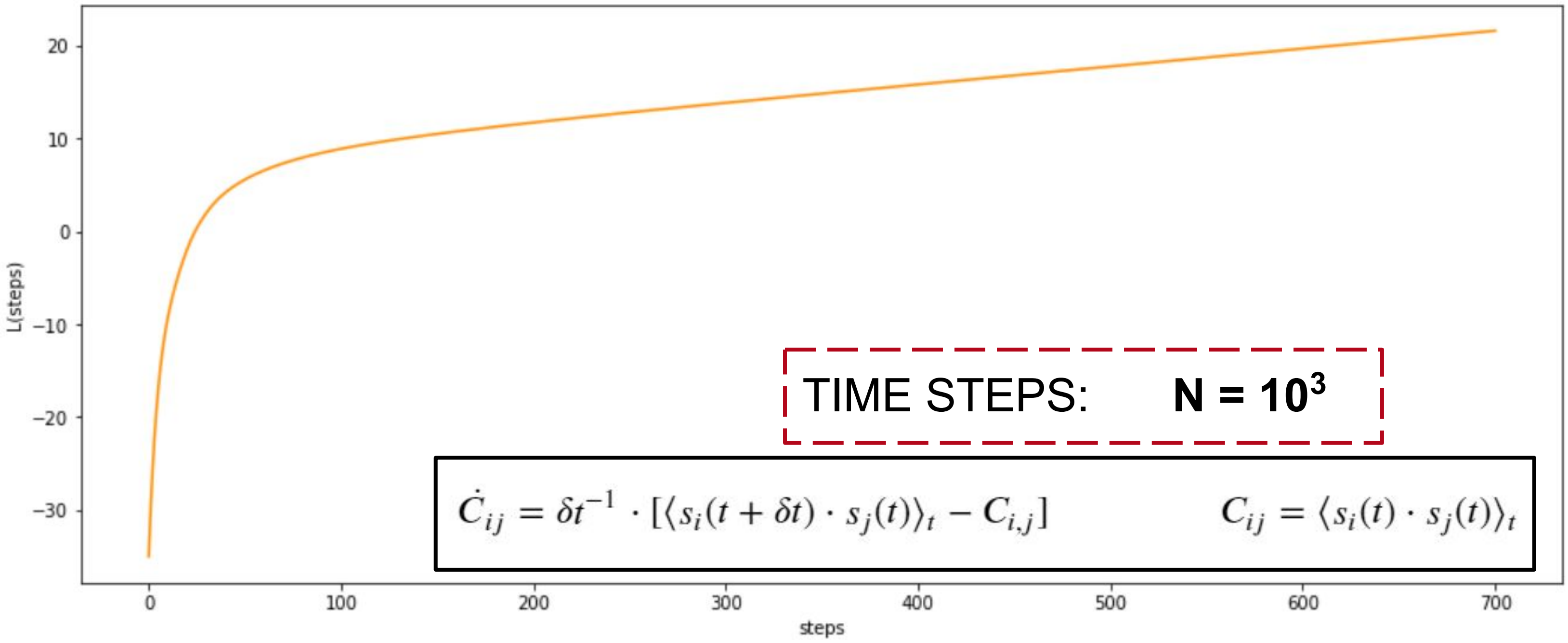


10 neurons: external field h



Convergence of the algorithms

GA convergence profile



TIME STEPS: **N = 10^3**

$$\dot{C}_{ij} = \delta t^{-1} \cdot [\langle s_i(t+\delta t) \cdot s_j(t) \rangle_t - C_{i,j}] \qquad C_{ij} = \langle s_i(t) \cdot s_j(t) \rangle_t$$

Optimization

UNIVERSITÀ DEGLI STUDI DI PADOVA

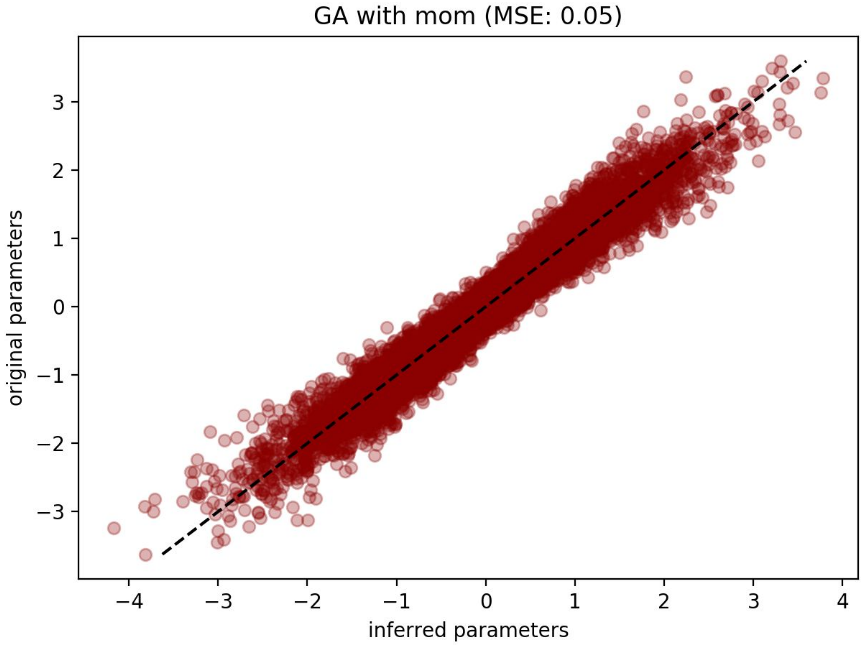
Numpy matrices

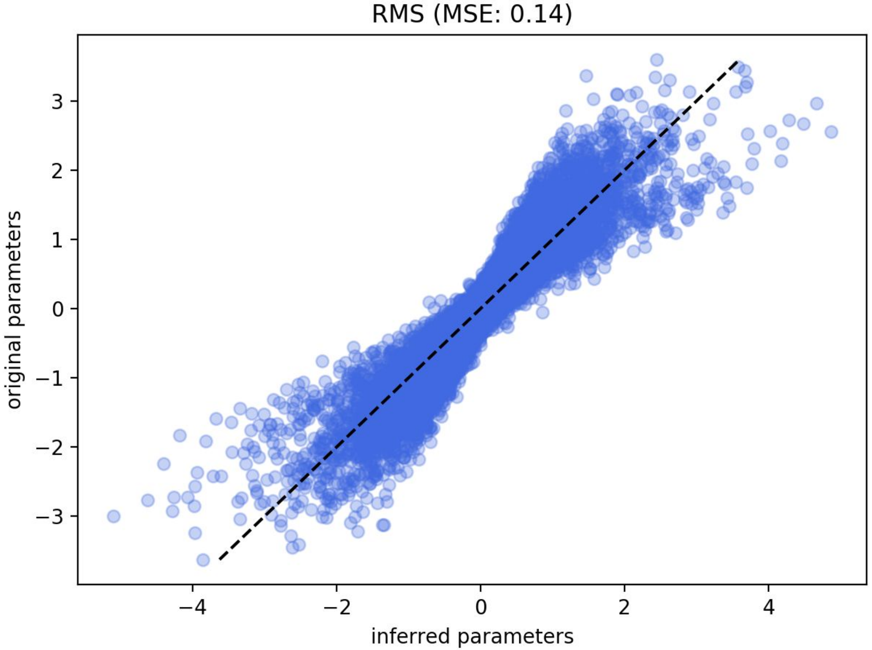
```
def grad_L(p,S,C,C_dot,gamma):
    N_t = S.shape[1]
    N = S.shape[0]

    # p are the params, S is the spins history
    J = p[:N*N].reshape((N,N))      # NxN matrix
    np.fill_diagonal(J,0)           #removing the diagonal
    h = p[N*N:]                     # N vector

    #grad wrt h
    H = (h + np.dot(J,S).T).T
    G_1 = np.mean(S-np.tanh(H),axis=1)

    #grad wrt J
    G_0 = C_dot/gamma + C
    G_0 -= (np.dot(np.tanh(H),S.T))/N_t

    #new
    np.fill_diagonal(G_0,0)
    return np.concatenate((G_0.flatten(),G_1))
```

Dask parallelization on Cloud Veneto to compute the gradient faster

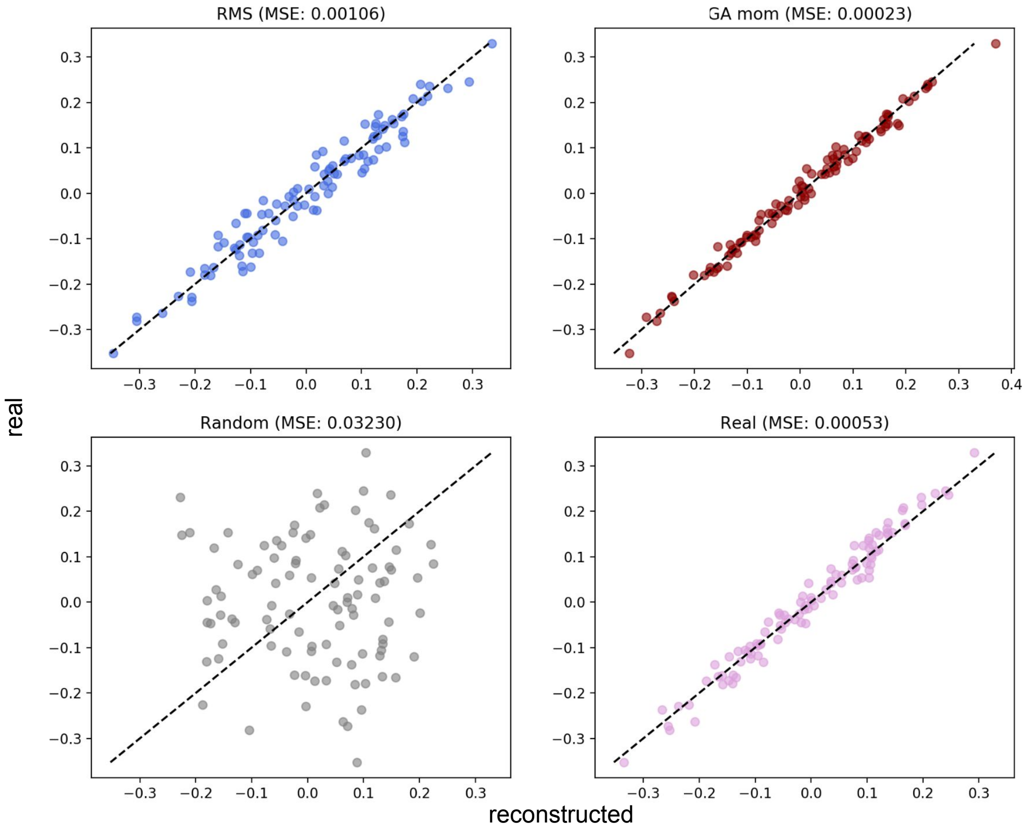
```
def parralel_grad(p,S_split,C,C_dot,gamma):
    ris = []
    for S in S_split:
        ris.append(delayed(grad_L)(p,S,C,C_dot,gamma))
    grad = delayed(sum)(ris).compute()/len(S_split)
    return grad
```

100 neurons: inference

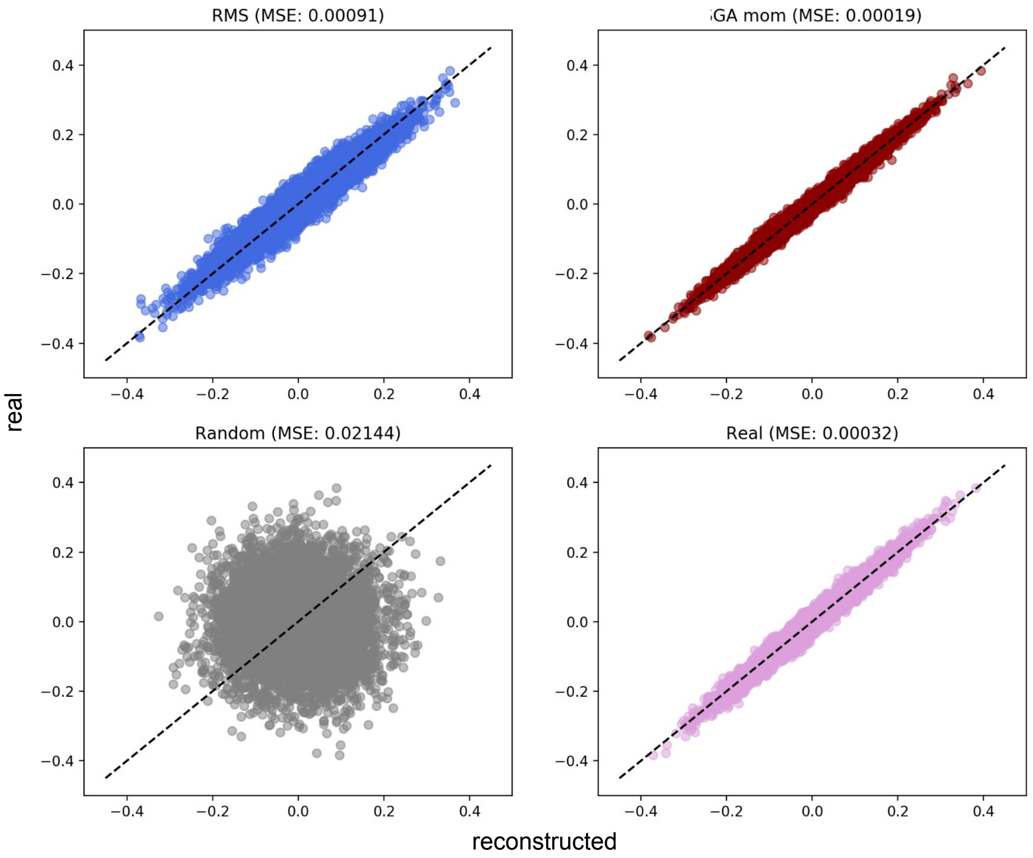




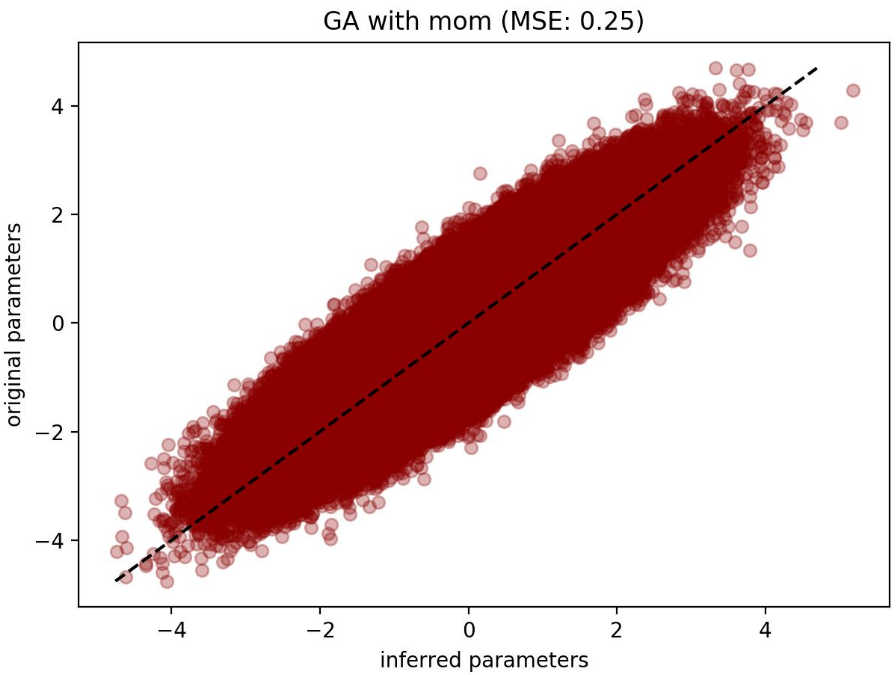
100 neurons: average magnetization per site

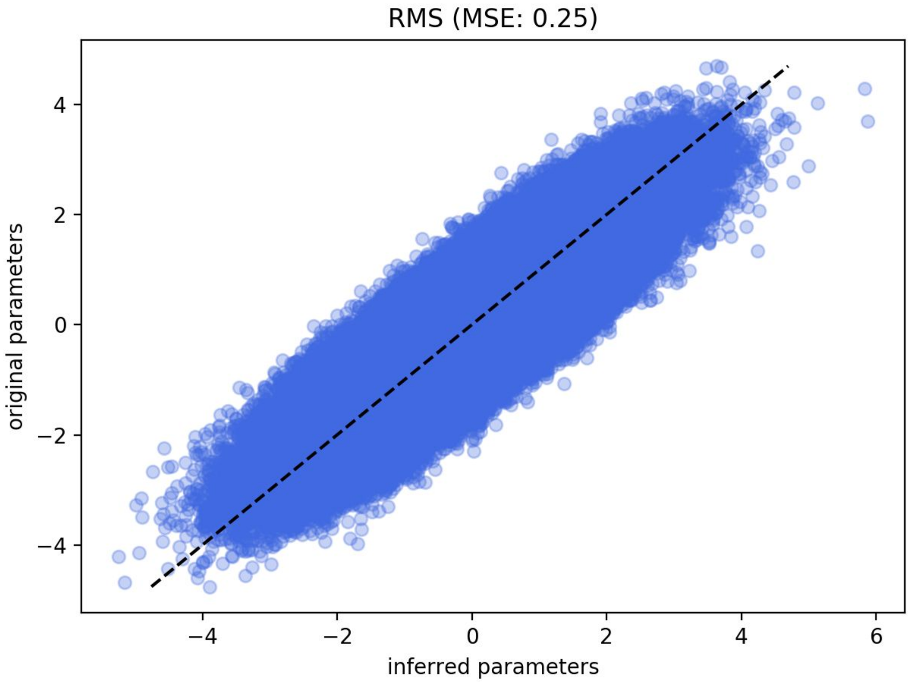


100 neurons: correlation matrix

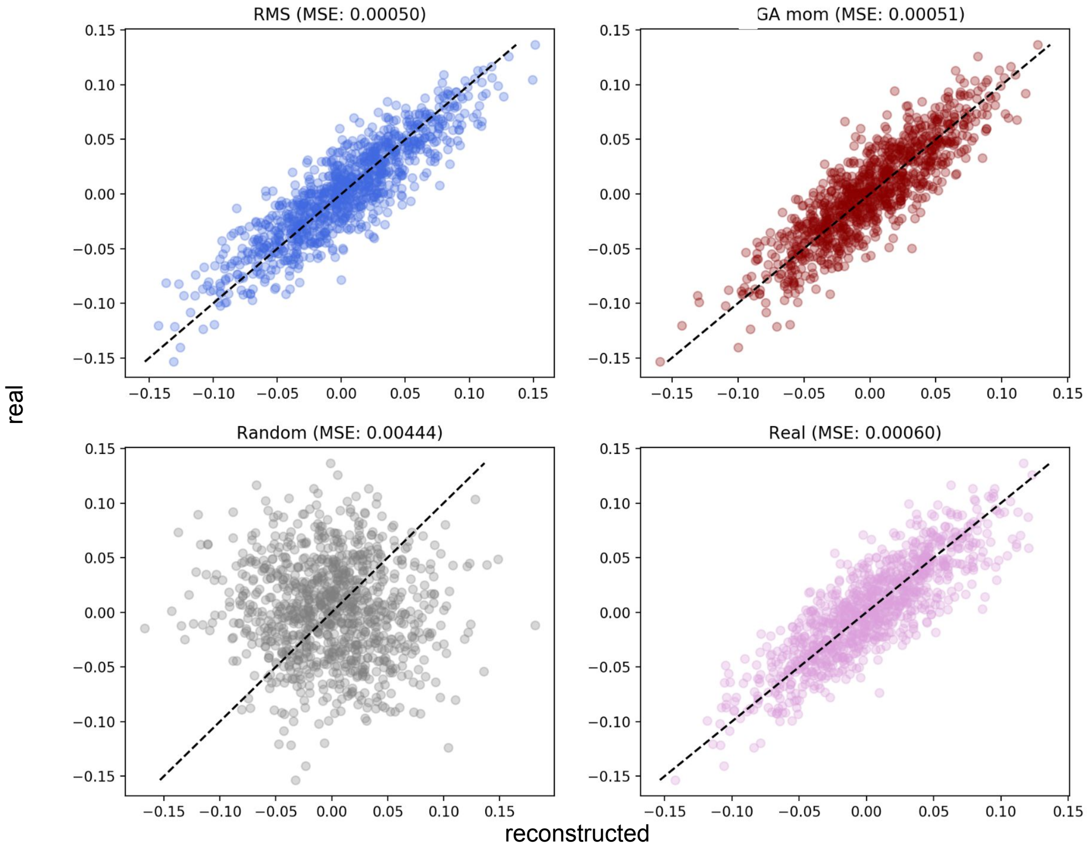


1000 neurons: inference

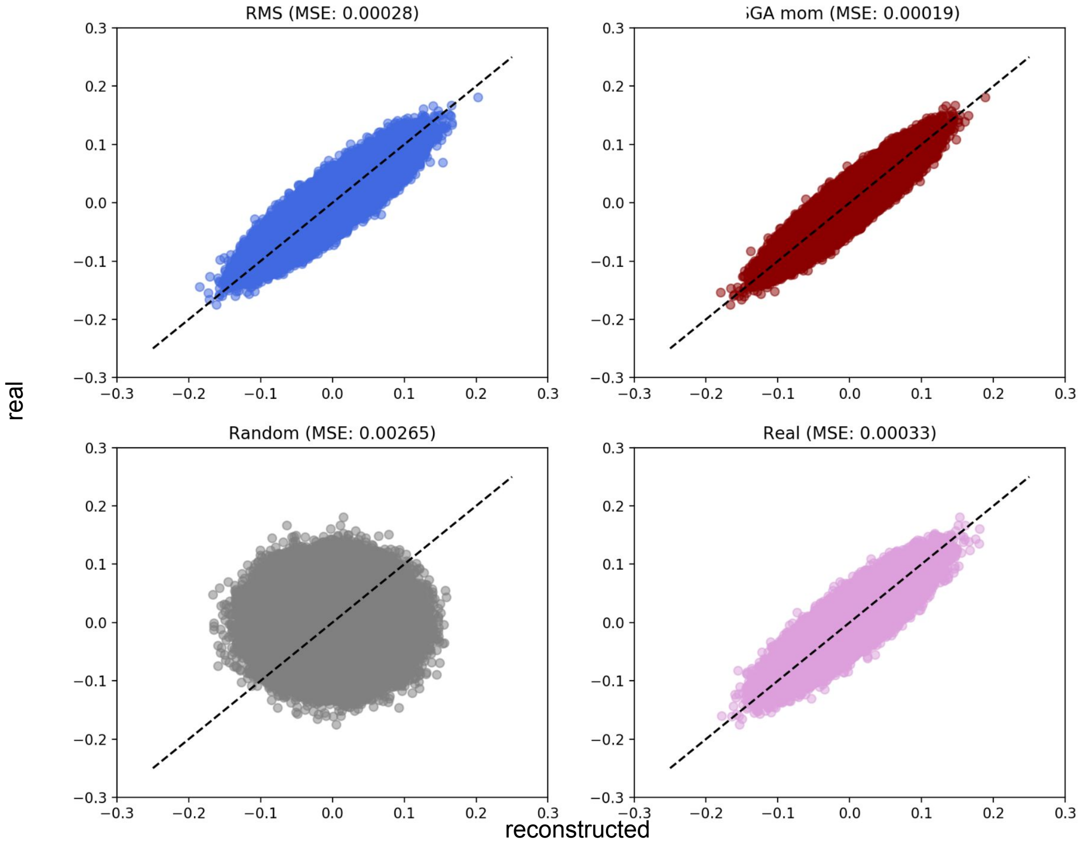




1000 neurons: average magnetization per site

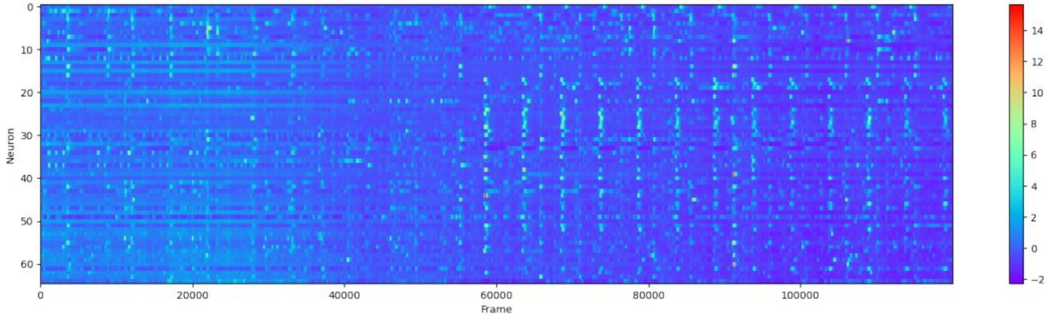


1000 neurons: correlation matrix

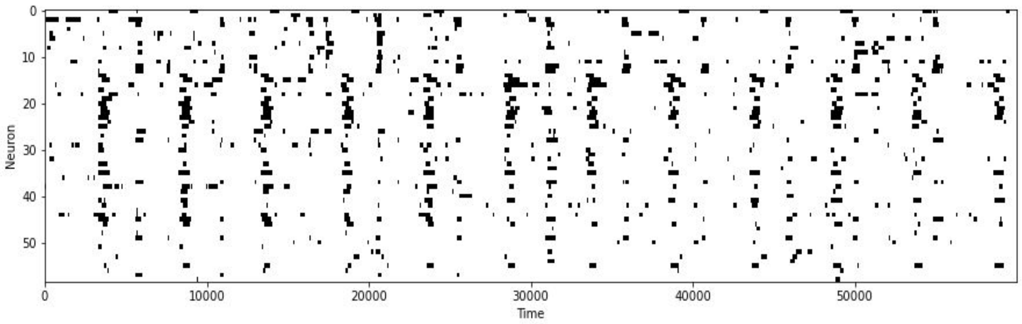


Real Dataset

from Raw Dataset: [3]

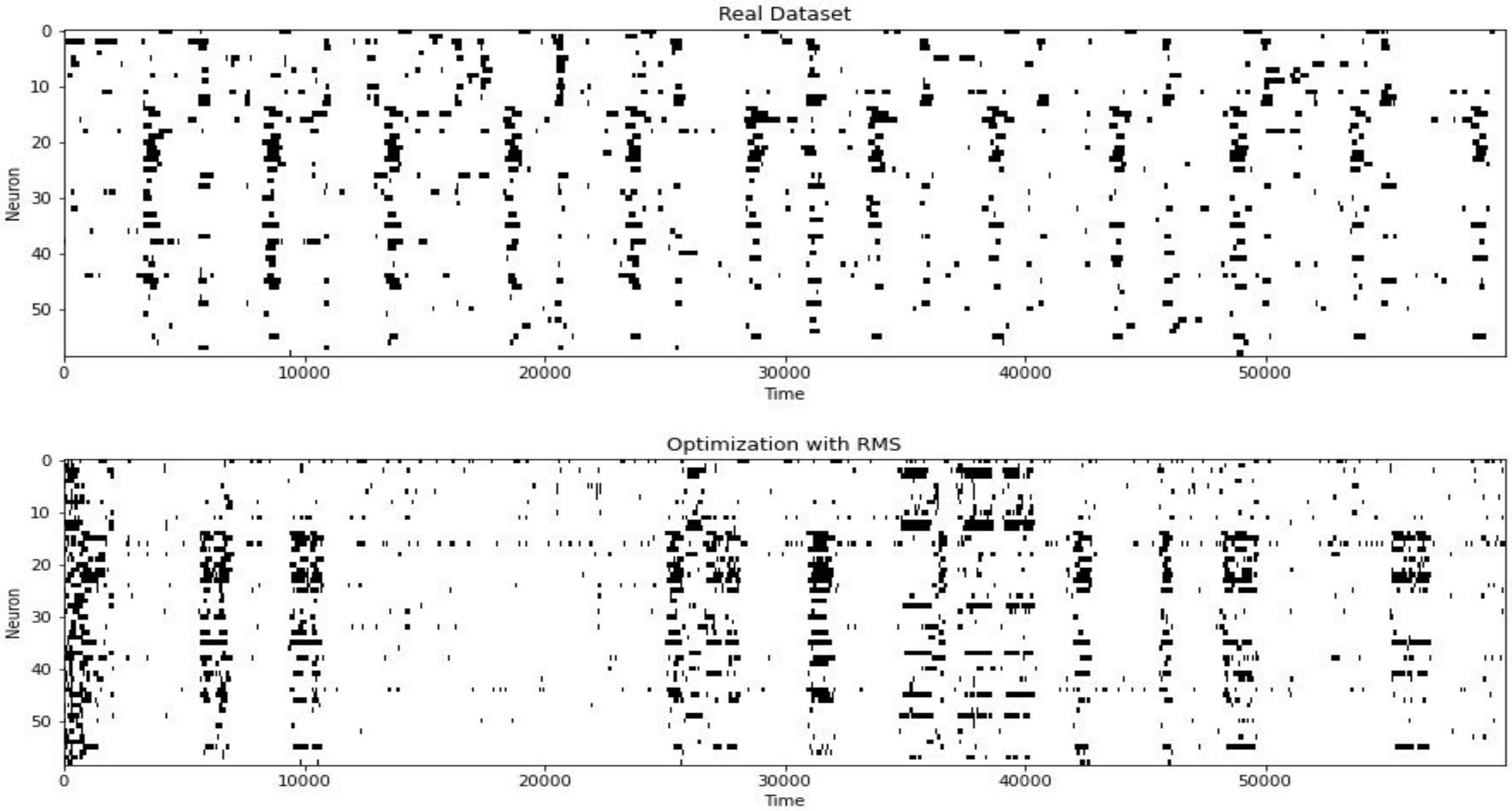


to Pre-Processed Dataset:



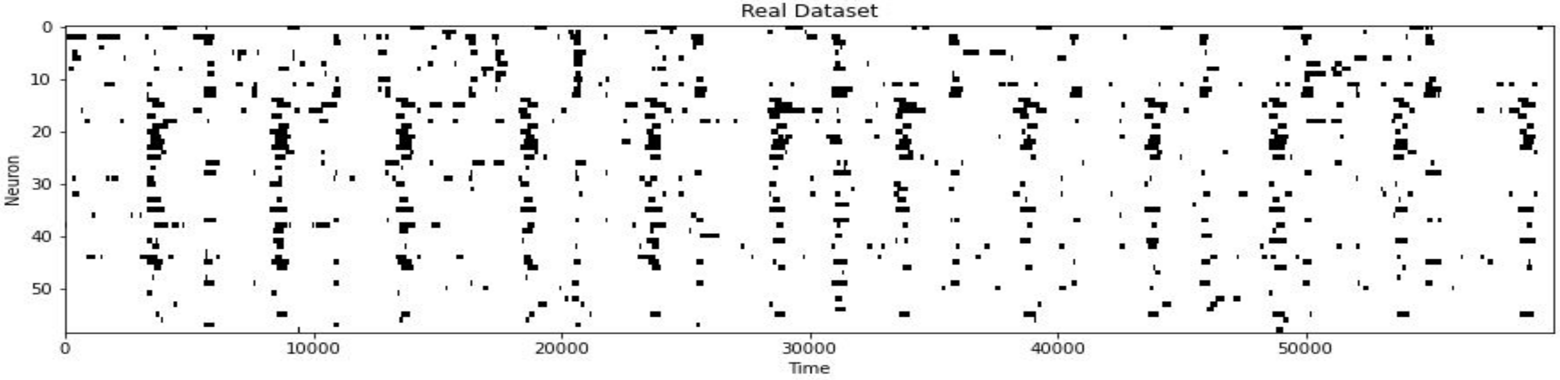
➜ discretizing data
➜ removing the first half of the frames without stimuli

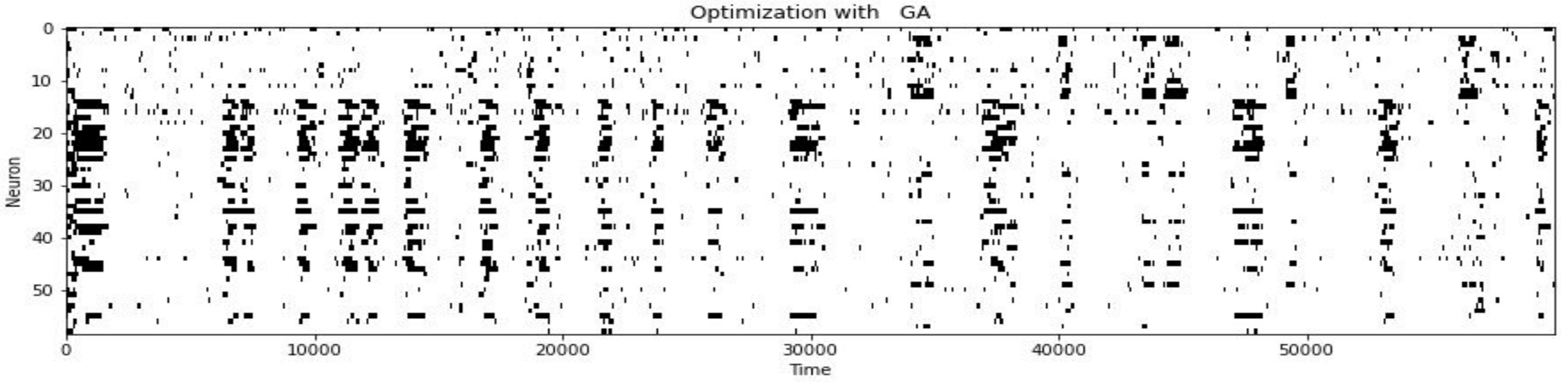
Real Dataset: reconstructed dynamic



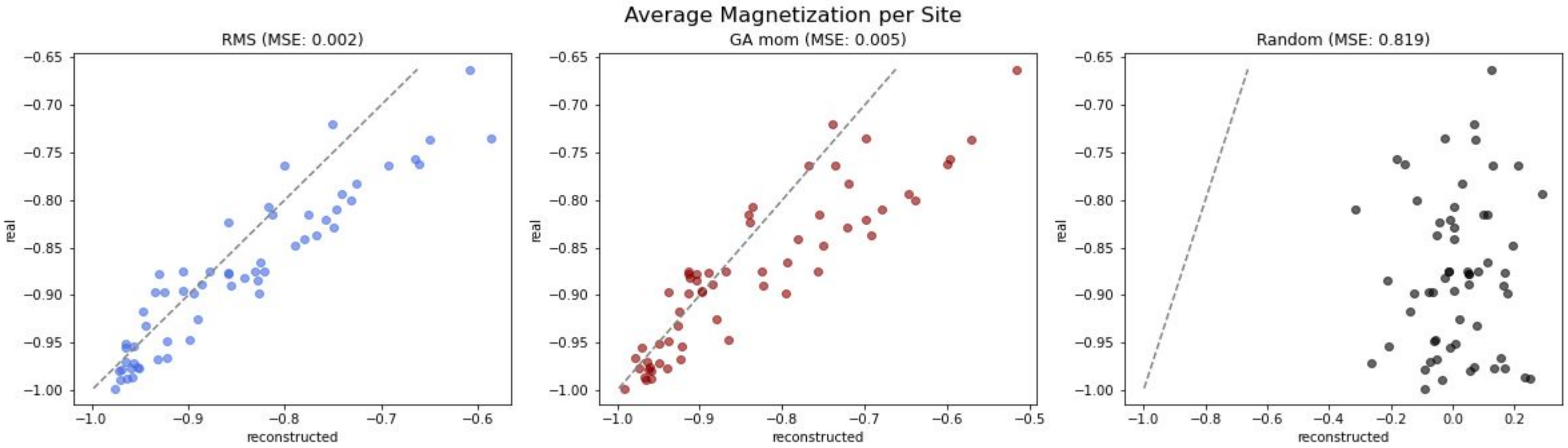
Real Dataset: reconstructed dynamic

Università degli Studi di Padova

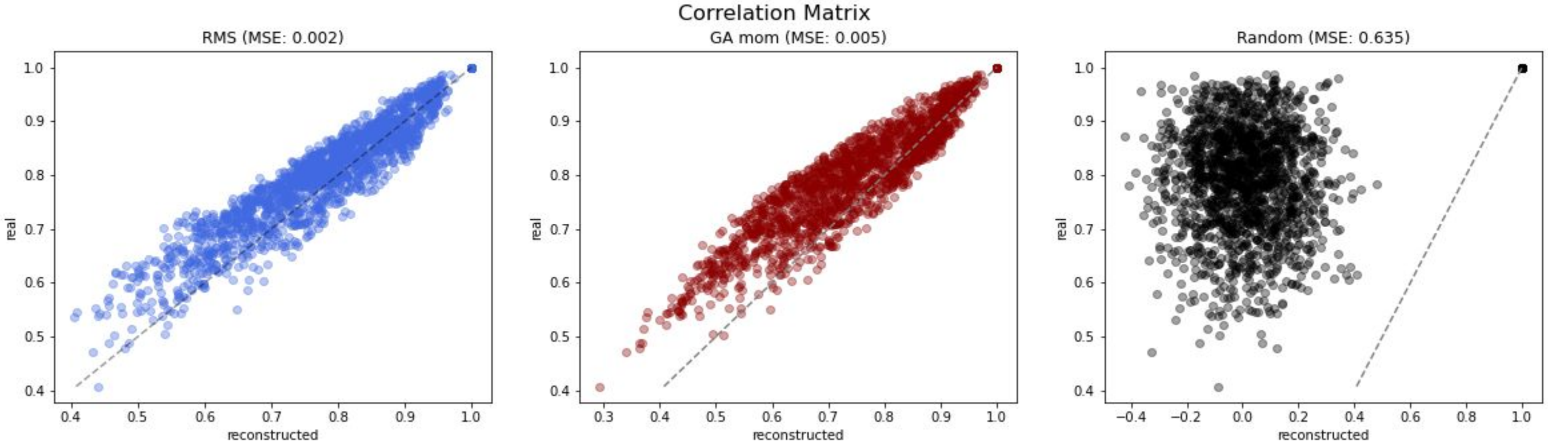




Average Magnetization per site



Correlation Matrix



Conclusions

The reconstruction model works well on synthetic dataset, while on the real dataset the performances get a little worse.

We have found that to work properly, the model needs a big number of time steps ($>10^5$)
➔ a future development is to use real datasets with high number of neurons and time steps

↓

experimental problem:
it's not possible yet to take measurements from an high number of neurons for a long time ➜ trade-off between the two

Possible solution: repeat more times and concatenate together the same timeserie.
But in real life neurons tend to adapt to visual stimuli ➜ this trend should be taken into account on the couplings that now are constant over the time.

Thank you for you attention

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

References

UNIVERSITÀ DEGLI STUDI DI PADOVA

[1] Hong-Li Zeng, Erik Aurell, Mikko Alava, and Hamed Mahmoudi.
Network inference using asynchronously updated kinetic Ising model
Phys. Rev. E **83**, 041135 – Published 29 April 2011

[2] Hong-Li Zeng, Mikko Alava, Erik Aurell, John Hertz, and Yasser Roudi.
Maximum Likelihood Reconstruction for Ising Models with Asynchronous Updates
Phys. Rev. Lett. **110**, 210601 – Published 20 May 2013

[3] Notes provided by the group LIPh.