JSM 2018

DYNAMIC BAYESIAN MODELS FOR NEURAL DATA ANALYSIS

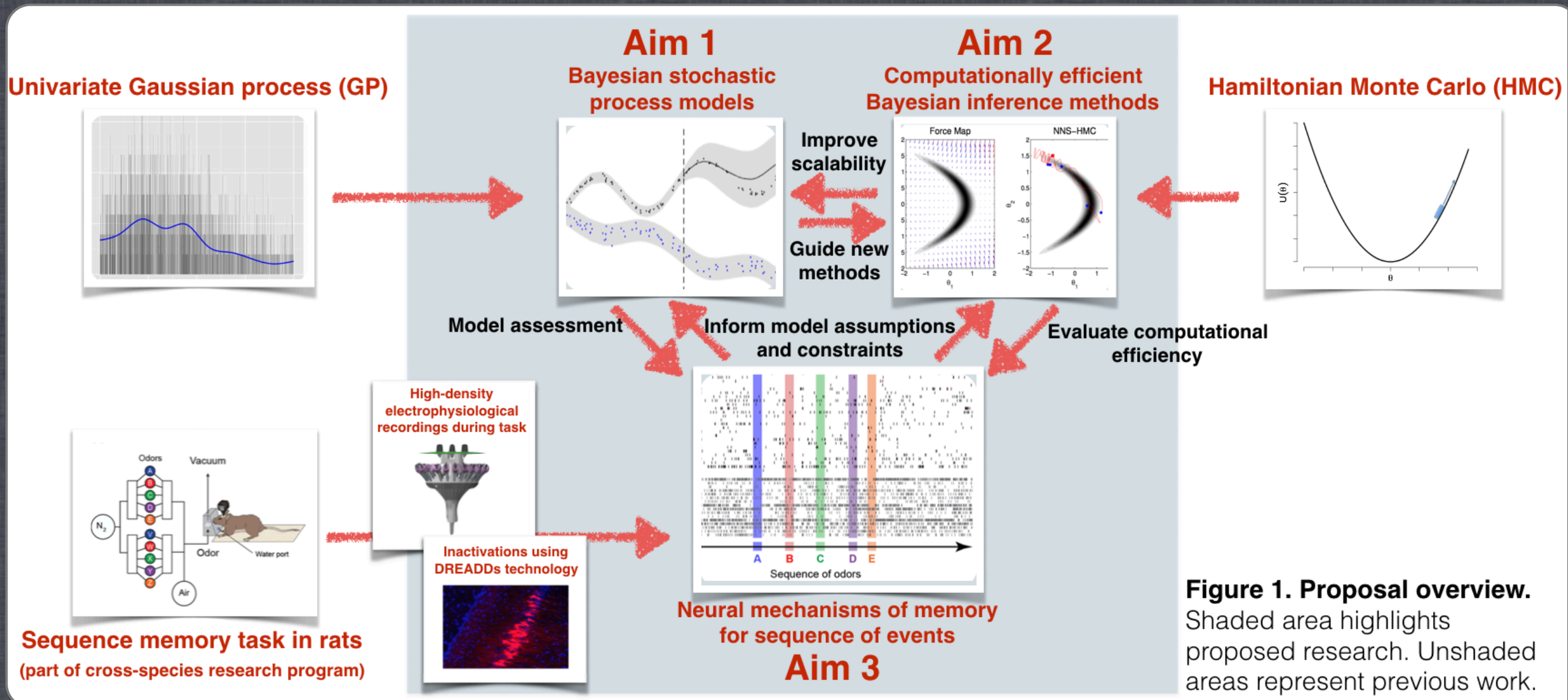


Figure 1. Proposal overview. Shaded area highlights proposed research. Unshaded areas represent previous work.

Babak Shahbaba, PhD
Departments of Statistics and Computer Science
UC Irvine

DMS 1622490

R01MH115697

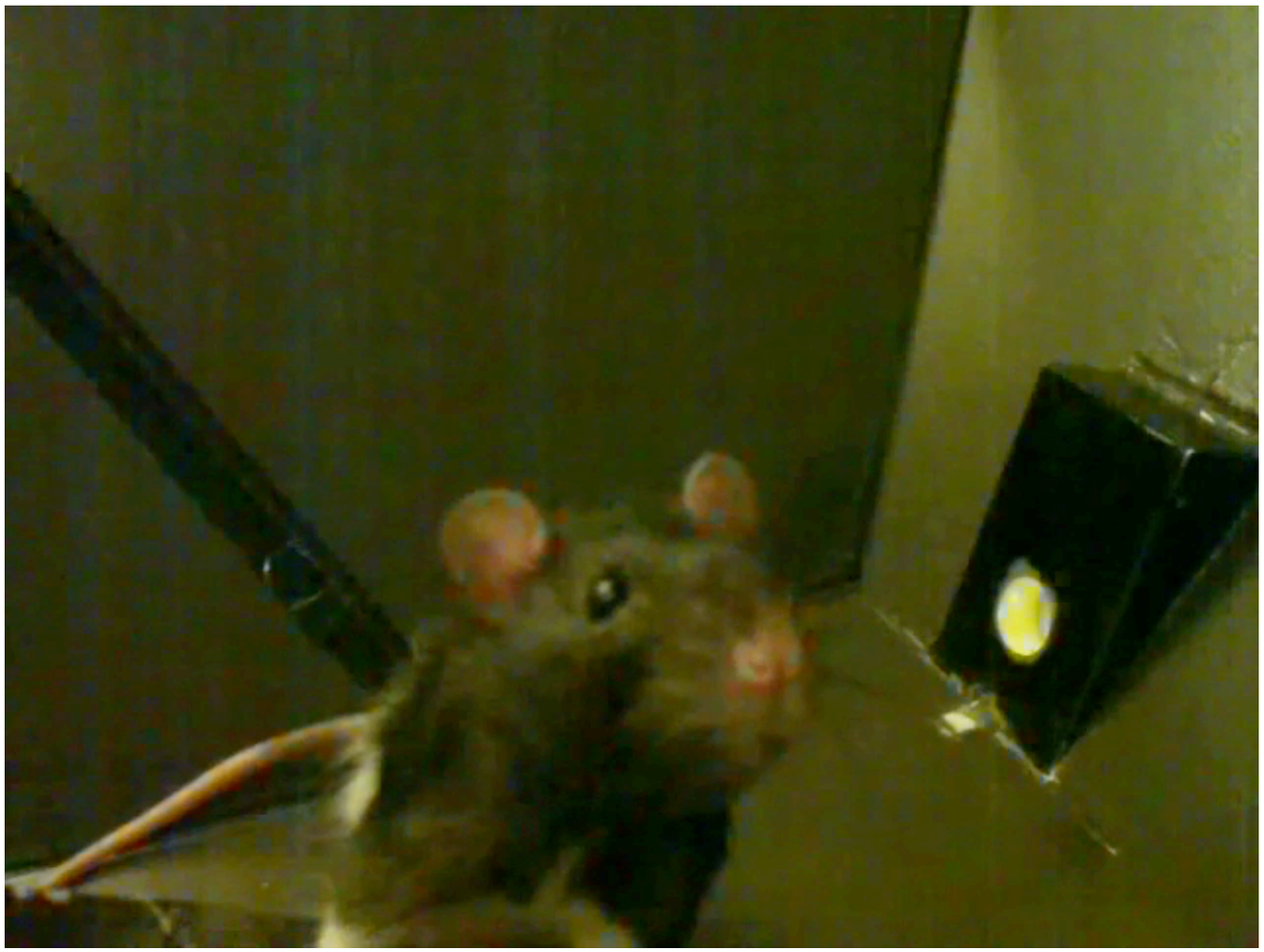
Outline

1. Experiment
2. Dynamic model for time series analysis
3. Computation
4. Results

Experiment

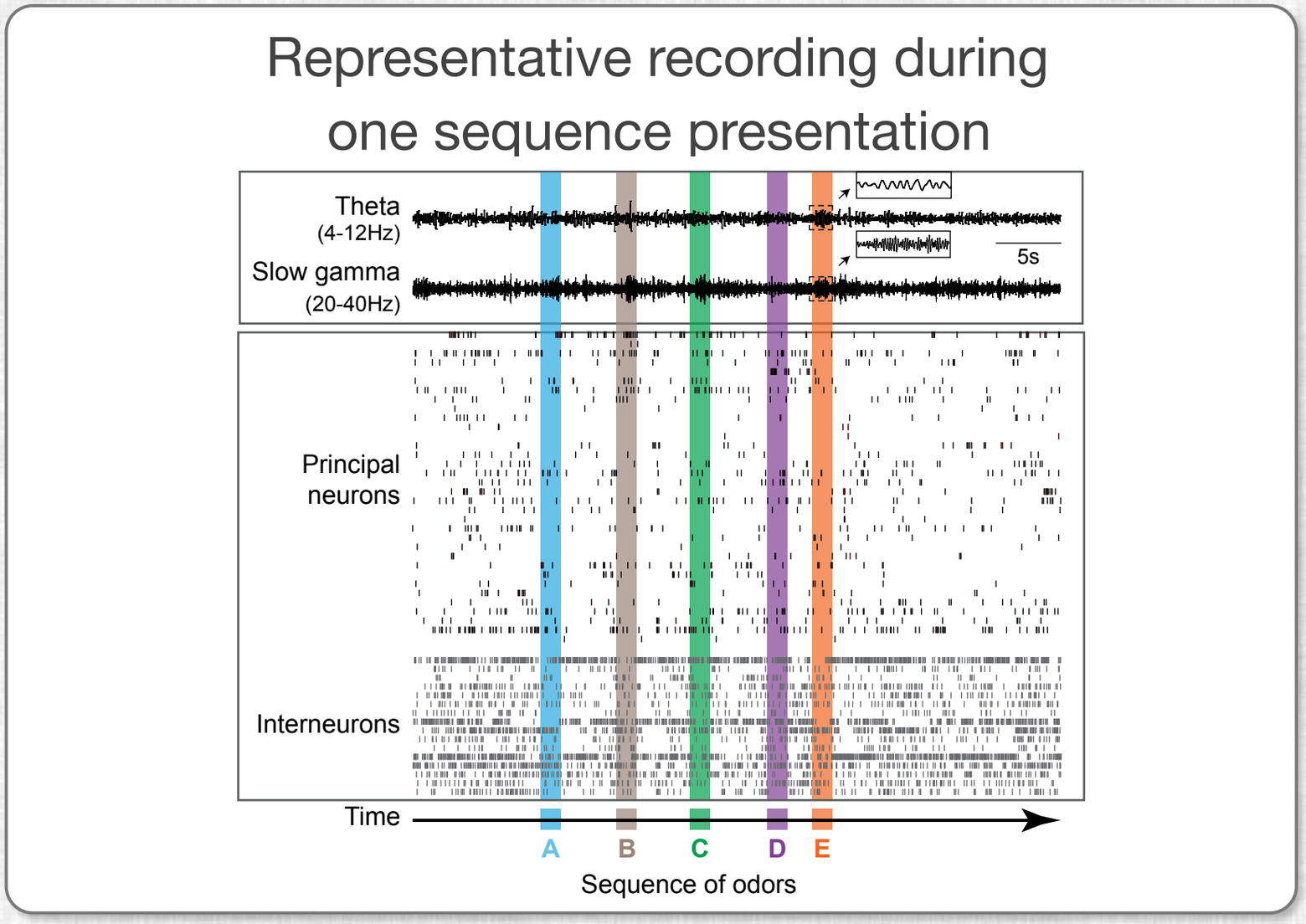
Behavioral Procedures in Rats

- Rats are trained on a particular correct sequence (A,B,C,D, E) of odors
- Each trial involves the rat smelling an odor through a port
- If the odor is out-of-sequence, the rat should withdraw its nose from the port quickly (before 1 second)



Hippocampal Activity

- We focus on a session consisted of **218** trials lasting anywhere from 0.48 to 1.74 seconds each.
- For each trial the data include LFP signals from **12** tetrodes and spike counts from **52** neurons.

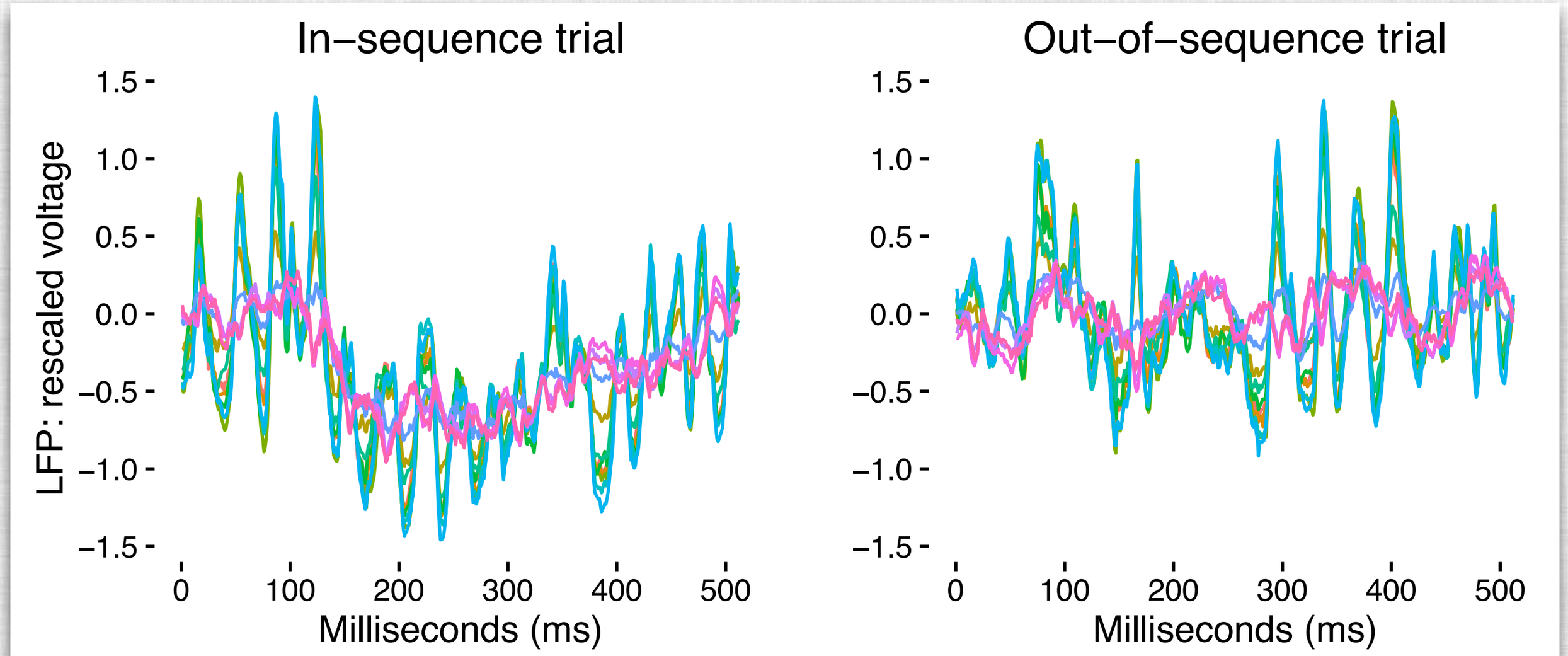


Scientific Questions

- We want to answer the following scientific questions:
  - What is the underlying neuronal mechanism for sequence memory?
  - Can we decode the rat's response from the neuronal data?
  - Are the two data modalities (spikes & LFP) complementary with respect to the outcome?
  - Using our models, can we demonstrate nonspatial forms of sequence reactivation (Replay)?

Dynamic Model

Dynamic Covariance Modeling of LFP



Stationary Model

- For multiple time series, we have

$$\mathbf{y}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma} = [\sigma_{ij}]_{D\times D} > 0$$
$$\mu_i(t) \sim \mathcal{GP}(0, \mathcal{C}), \quad i = 1, \cdots, D$$

- Spatial dependence is coded in $\sum$
- Temporal evolution is modeled by the GP model
- We need to ensure the positive-definiteness of the covariance matrix over time

Stationary Model

- We can use Cholesky decomposition of covariance matrix:

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^{\mathsf{T}}, \quad \sigma_{ij} = \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk}, \quad \mathbf{L} = \begin{bmatrix} * & & \\ ** & & \\ * * * & & \end{bmatrix}$$

$$\sigma_i^2 := \sigma_{ii} = \sum_{k=1}^{i} l_{ik}^2 = \|\mathbf{l}_i\|^2, \quad \mathbf{L} = \begin{bmatrix} \mathbf{l}_1 \\ \vdots \\ \mathbf{l}_D \end{bmatrix}$$

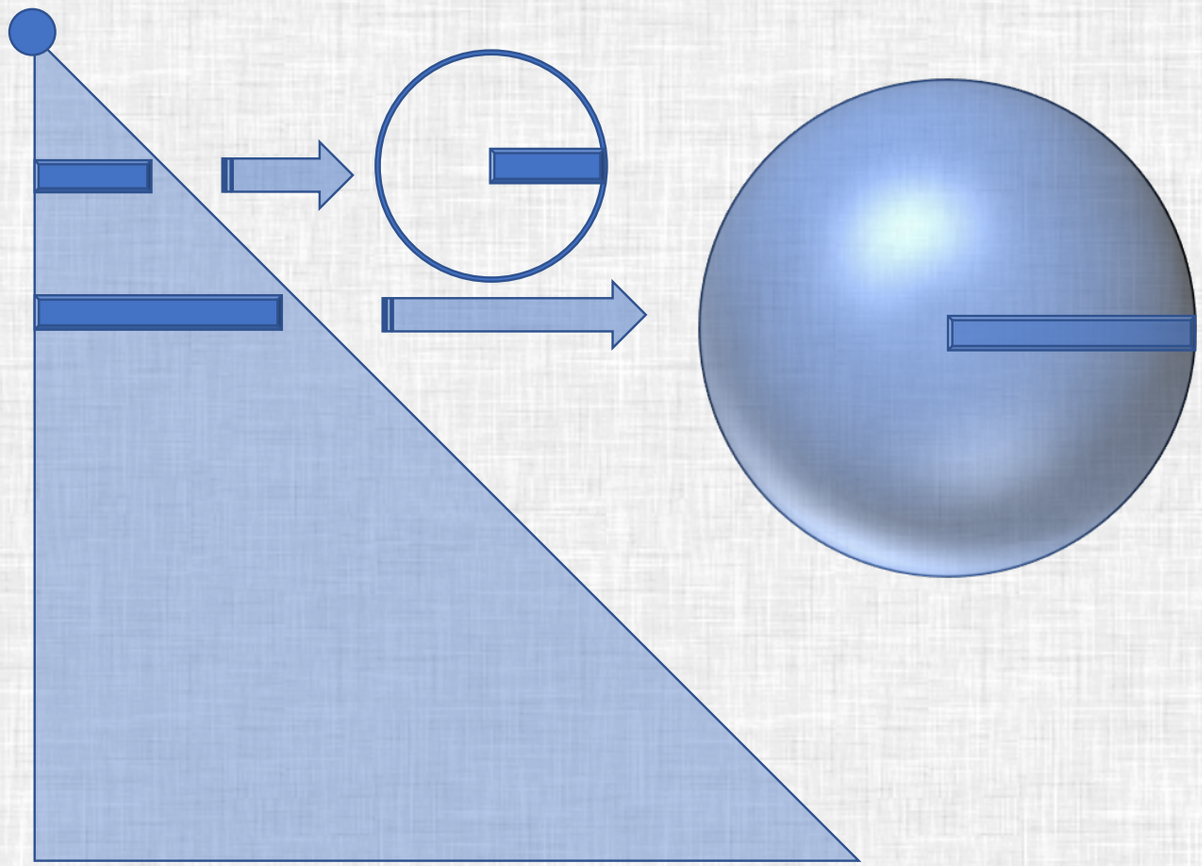
$$(\mathbf{l}_1, \mathbf{l}_2, \cdots, \mathbf{l}_D) \in \mathcal{S}_0^0(\sigma_1) \times \mathcal{S}_0^1(\sigma_2) \cdots \times \mathcal{S}_0^{D-1}(\sigma_D)$$

- For correlation matrix:

$$\mathbf{P} := \mathrm{diag}(\boldsymbol{\Sigma})^{-\frac{1}{2}} \boldsymbol{\Sigma}\, \mathrm{diag}(\boldsymbol{\Sigma})^{-\frac{1}{2}} = \mathbf{L}^* (\mathbf{L}^*)^{\mathsf{T}}, \; \rho_{ij} = \sum_{k=1}^{\min\{i,j\}} l_{ik}^* l_{jk}^*$$

$$(\mathbf{l}_1^*, \mathbf{l}_2^*, \cdots, \mathbf{l}_D^*) \in \mathcal{S}_0^0 \times \mathcal{S}_0^1 \cdots \times \mathcal{S}_0^{D-1}$$

Geometric View of Covariance Matrix



$$(\mathbf{l}_1, \mathbf{l}_2, \cdots, \mathbf{l}_D) \in \mathcal{S}_0^0(\sigma_1) \times \mathcal{S}_0^1(\sigma_2) \cdots \times \mathcal{S}_0^{D-1}(\sigma_D)$$

$$(\mathbf{l}_1^*, \mathbf{l}_2^*, \cdots, \mathbf{l}_D^*) \in \mathcal{S}_0^0 \times \mathcal{S}_0^1 \cdots \times \mathcal{S}_0^{D-1}$$

- We have previously developed a sampling method for distribution defined on spheres (discussed later)

Priors

- We can show that the conjugate Inverse-Wihsart prior can be presented in this framework
- However, our proposed approach allows for specifying more flexible priors
- If two variables $\boldsymbol{y_i}$ and are $\boldsymbol{y_j}$ known to be uncorrelated a priori, then we can choose a prior that encourages $\boldsymbol{l_i}$ and $\boldsymbol{l_j}$ to be perpendicular
- For example, we can specify priors $\boldsymbol{p(l_i)}$ that concentrate on the poles of $\boldsymbol{S_0^{j-1}}$,
- This leads to fewer non-zero off-diagonal elements, which in turn leads to a larger number of uncorrelated variables

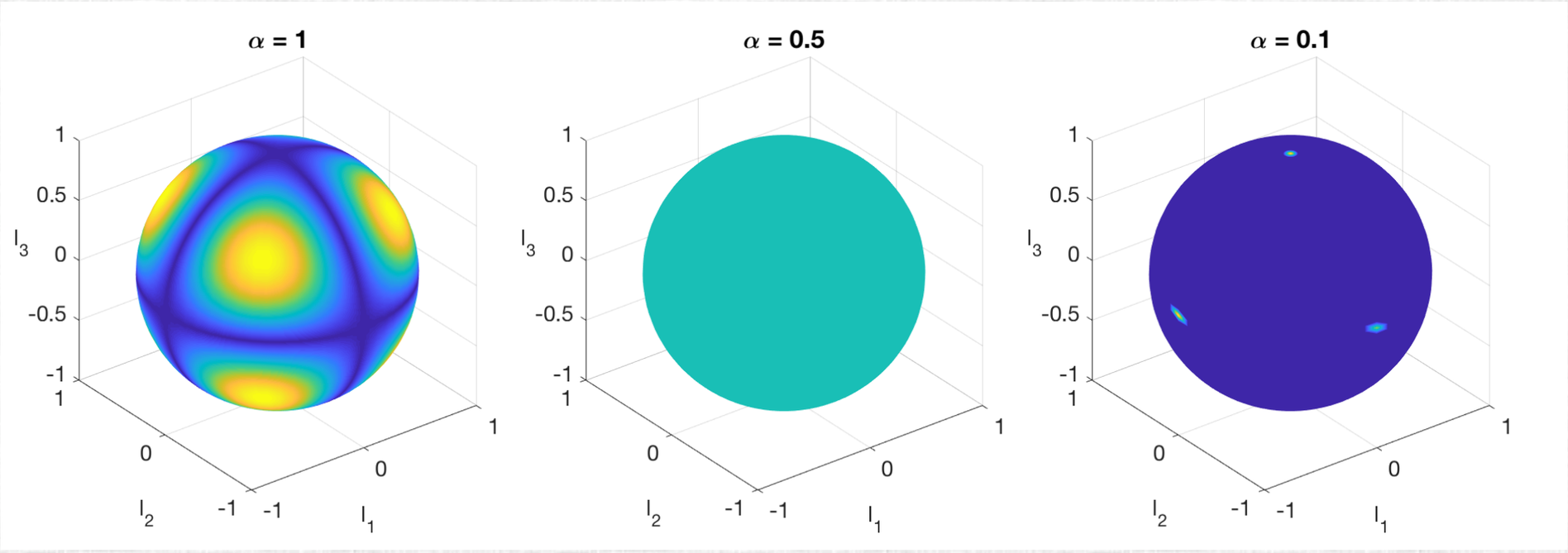
Squared-Dirichlet Prior

- In general, we can define priors on $\boldsymbol{L}$ by mapping a probability distribution defined on the simplex onto the sphere
- Squared-Dirichlet distribution

$$\mathbf{l}_i^2 := (l_{i1}^2, l_{i2}^2, \cdots, l_{ii}^2) \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)$$

$$\mathbf{l}_i \sim \mathrm{Dir}^2(\boldsymbol{\alpha}_i)$$

$$p(\mathbf{l}_i) = p(\mathbf{l}_i^2)|2\mathbf{l}_i| \propto (\mathbf{l}_i^2)^{\boldsymbol{\alpha}_i - 1}|\mathbf{l}_i| = |\mathbf{l}_i|^{2\boldsymbol{\alpha}_i - 1} := \prod_{k=1}^{i} |l_{ik}|^{2\alpha_{ik}-1}$$



Squared-Dirichlet Prior

- Setting $\boldsymbol{\alpha}_i = (\frac{1}{2}\mathbf{1}_{i-1}^\mathsf{T}, \alpha_{ii}),\ \alpha_{ii} = \frac{(i-2)D-1}{2}$ leads to a marginally uniform prior

$$\rho_{ij} \sim \text{Unif}(-1, 1), \quad i \neq j$$

- Setting $\boldsymbol{\alpha}_i = (\frac{1}{2}\mathbf{1}_{i-1}^\mathsf{T}, \alpha_{ii}),\ \alpha_{ii} = \frac{D-i}{2} + 1$ leads a joint uniform prior

$$p(\mathbf{P}) \propto 1$$

Unit Vector Gaussian Prior

- Another natural spherical prior can be obtained by constraining a multivariate Gaussian random vector to have unit norm.
- Unit-vector Gaussian distribution:

$$p(\mathbf{l}_i | \, \|\mathbf{l}_i\|_2 = 1) = \frac{1}{(2\pi)^{\frac{i}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} (\mathbf{l}_i - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{l}_i - \boldsymbol{\mu}) \right\}, \; \|\mathbf{l}_i\|_2 = 1$$

- The conditional density essentially defines Fisher-Bingham distribution and von Mises-Fisher distribution.

Dynamic Covariance Modeling

- To model the covariance (or correlation) matrix dynamically, we use a time-varying Cholesky matrix, $\boldsymbol{L}_t$
- Since each row of $\boldsymbol{L}_t$ has to be on a sphere of certain dimension, we consider a multivariate process, called unit-vector process (uvP), satisfying the unit-norm requirement

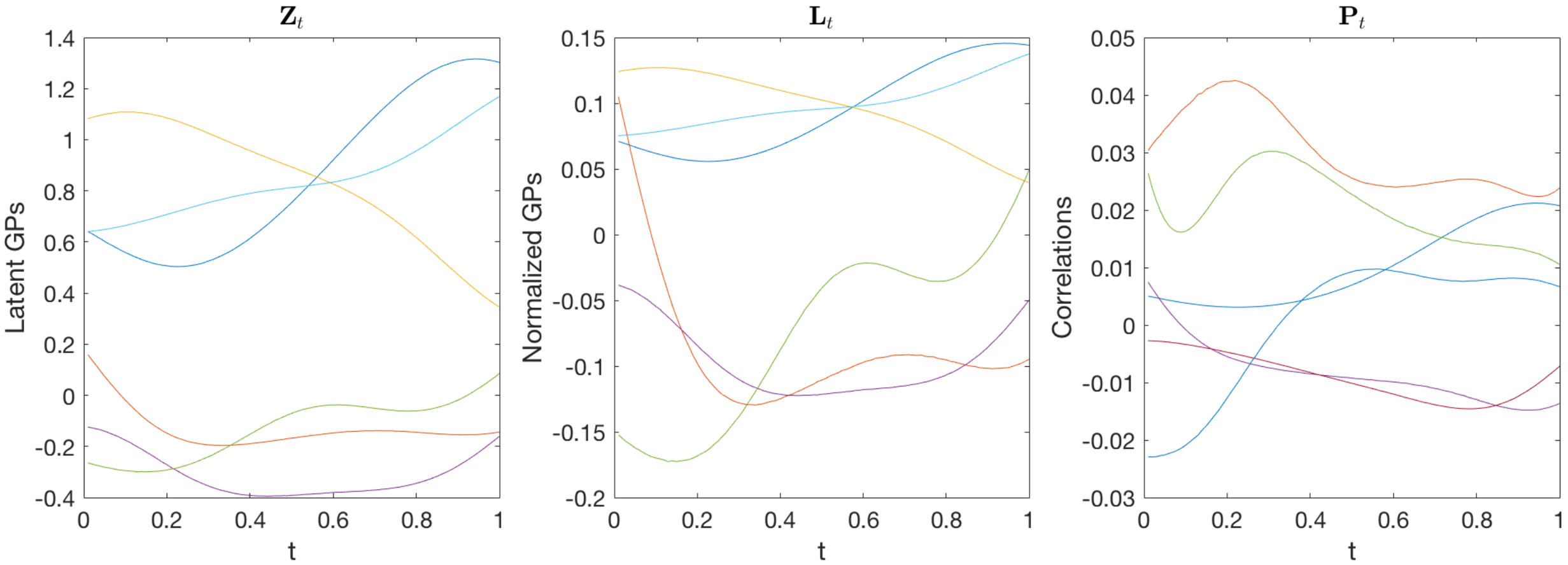
$$\|\mathbf{l}_i(t)\| \equiv 1, \quad \forall t \in \mathcal{T}$$

- More specifically, we are using A D-dimensional vector Gaussian process

$$\mathbf{Z}(x) \sim \mathcal{GP}_D(\boldsymbol{\mu}, \mathcal{C}, \mathbf{V}_{D\times D})$$

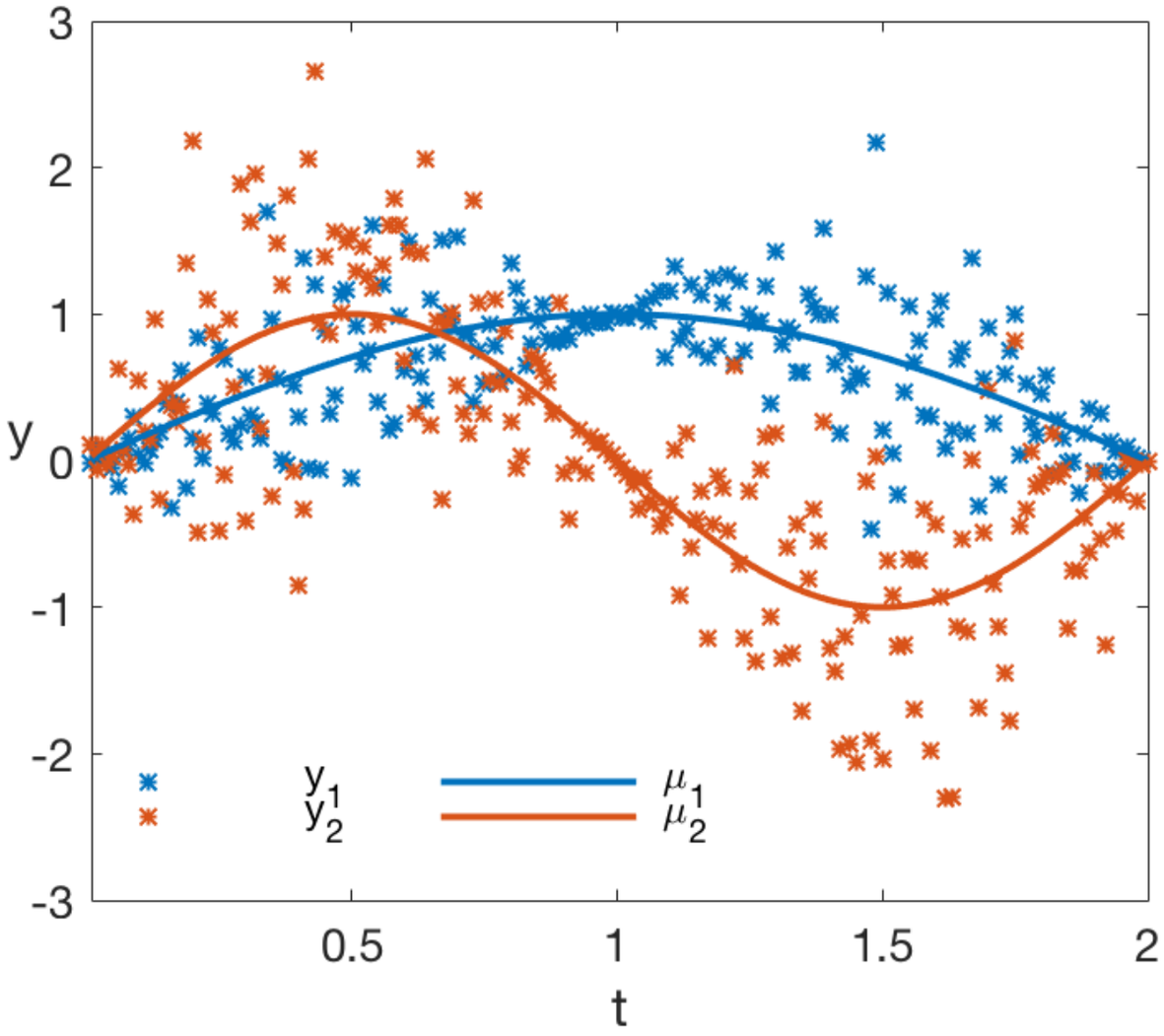
which is constrained to unit-sphere to obtain a unit-vector Gaussian process (uvGP)

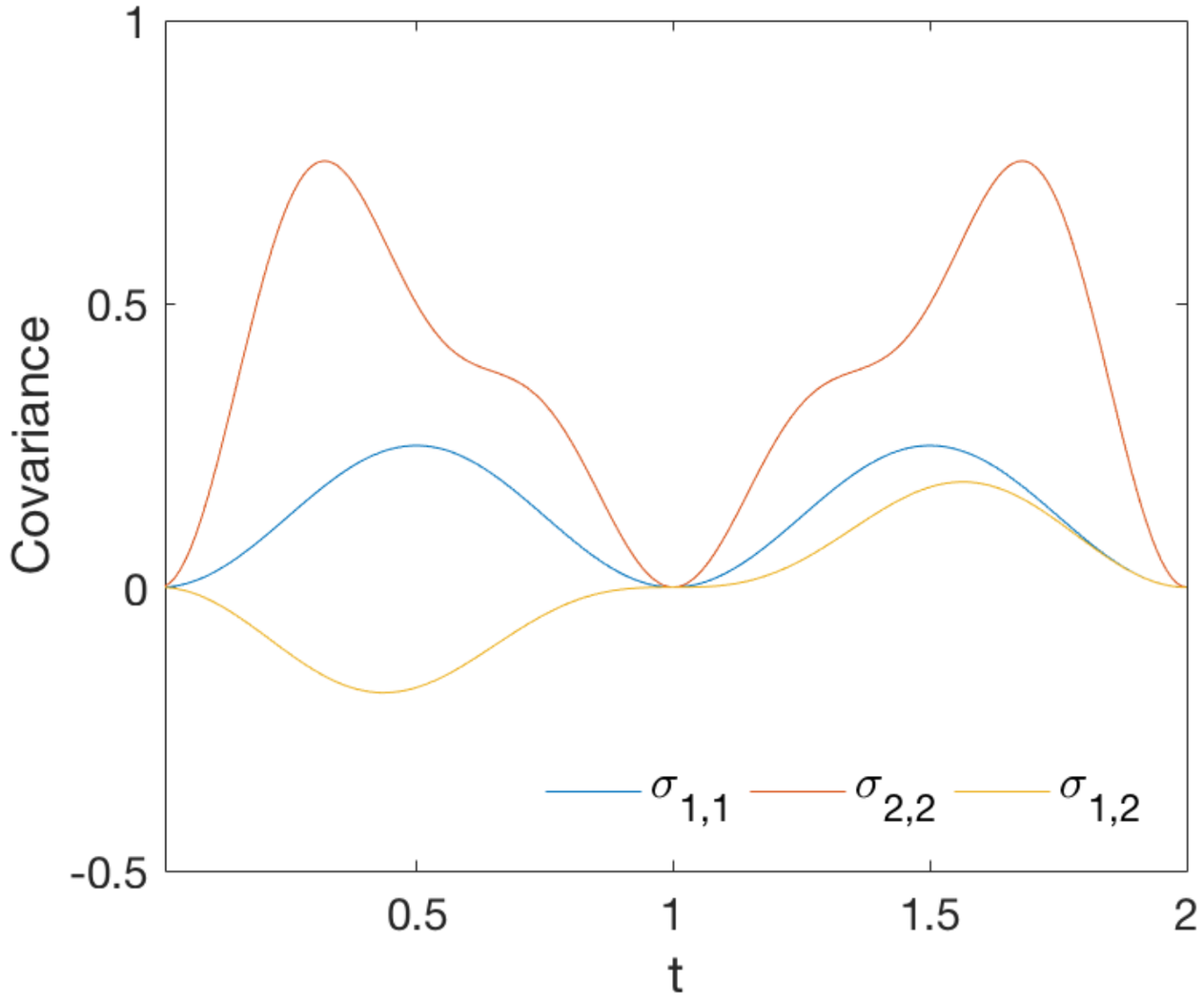
Dynamic Covariance Modeling



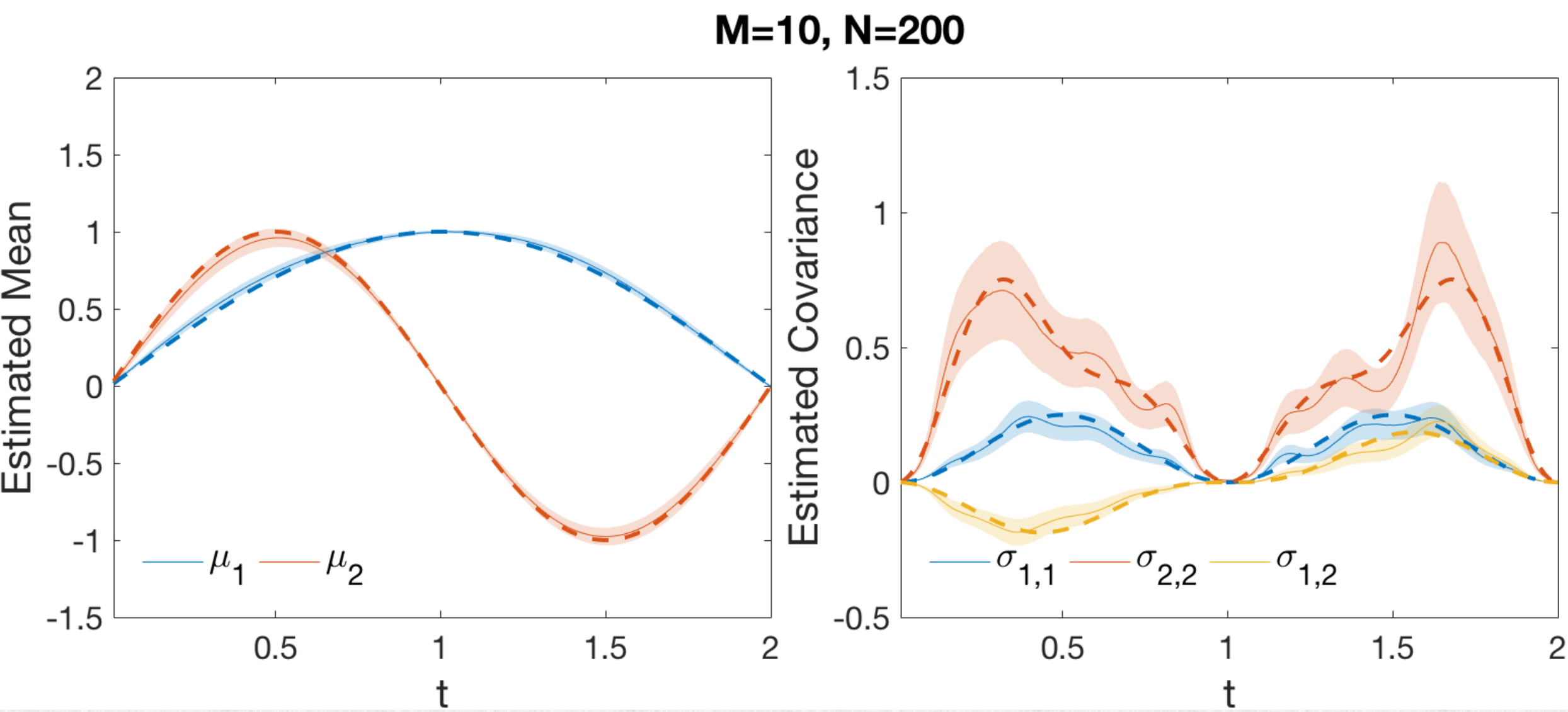
A realization of vector GP (left), its normalization (forming rows of) $\boldsymbol{L_t}$ (middle) and the induced correlation process.

Illustrative Example

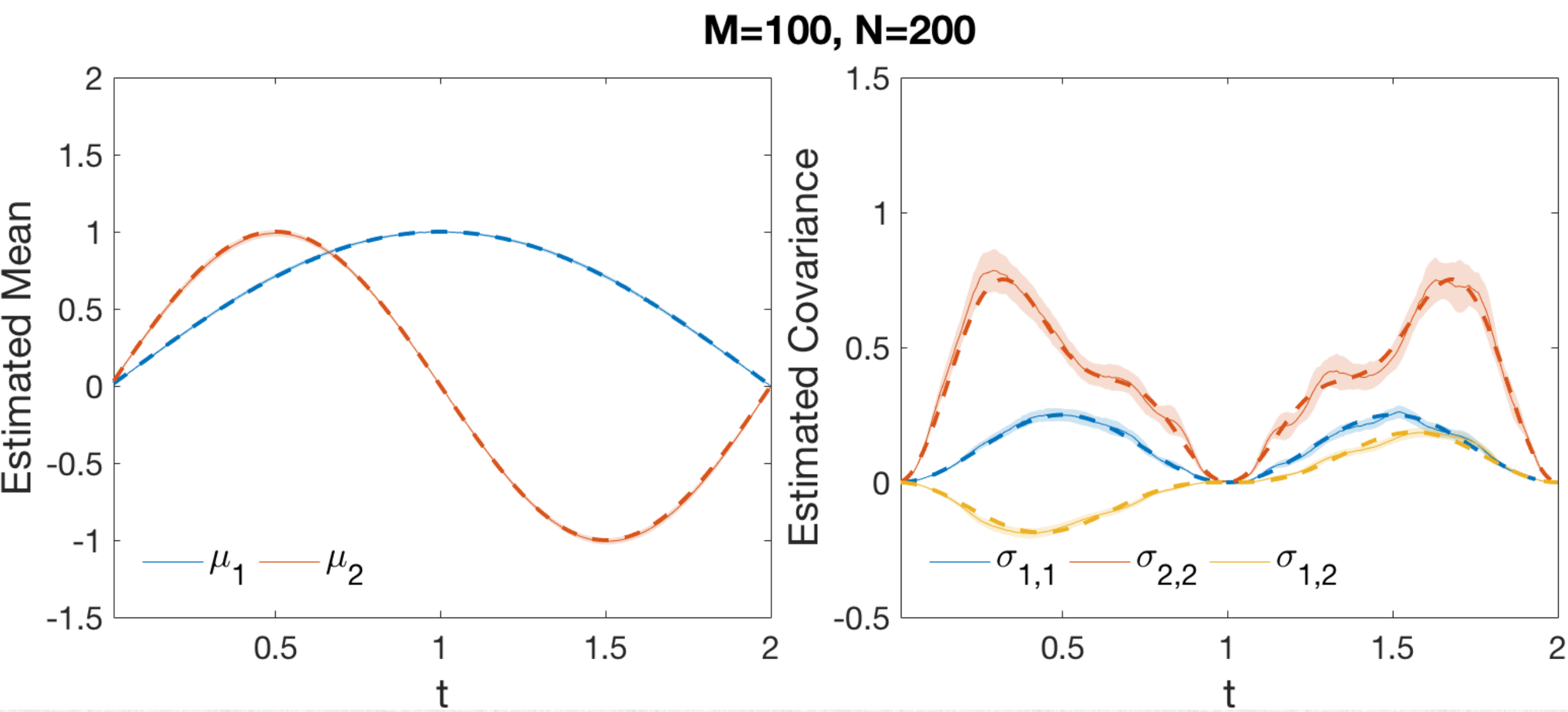




Illustrative Example

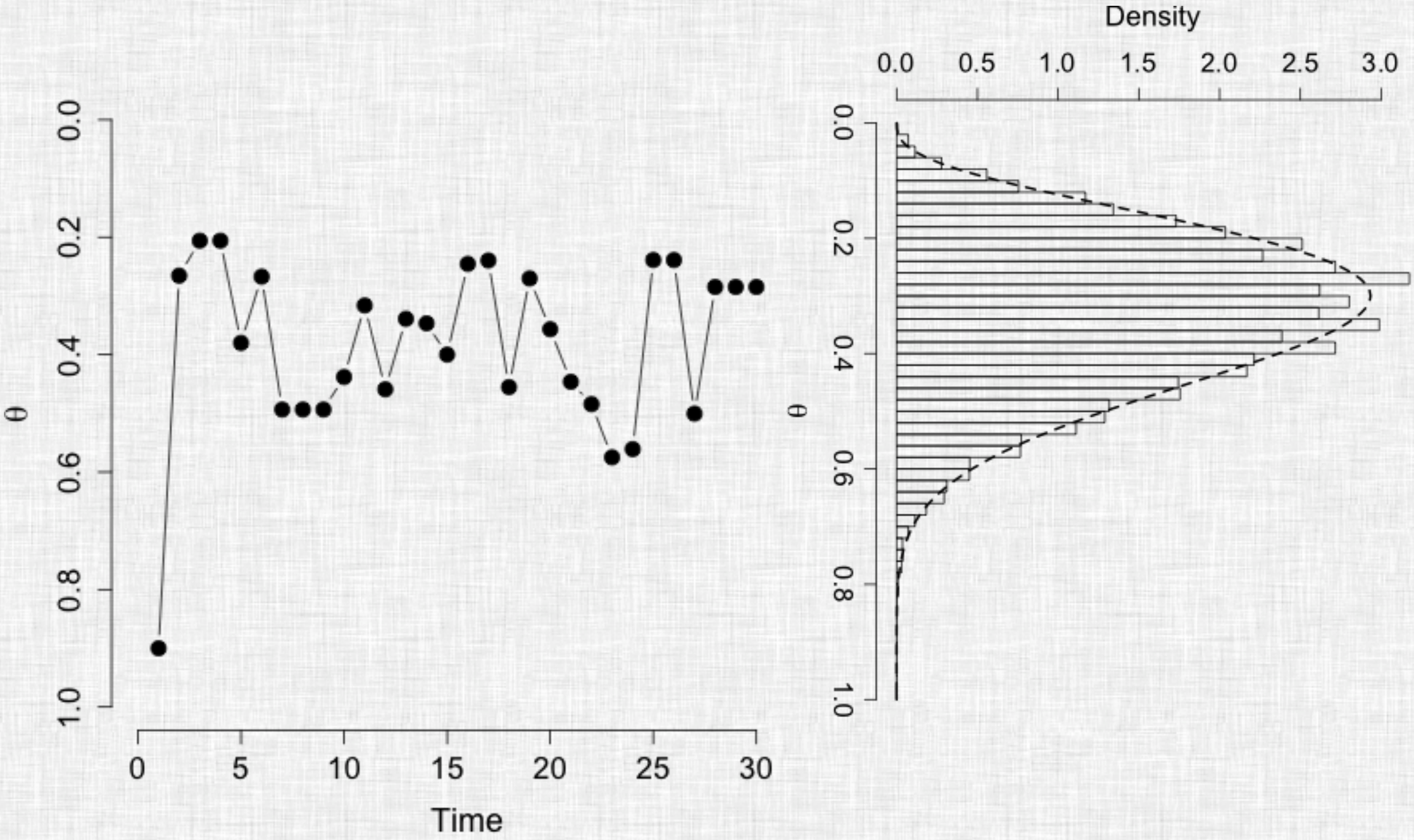


Illustrative Example

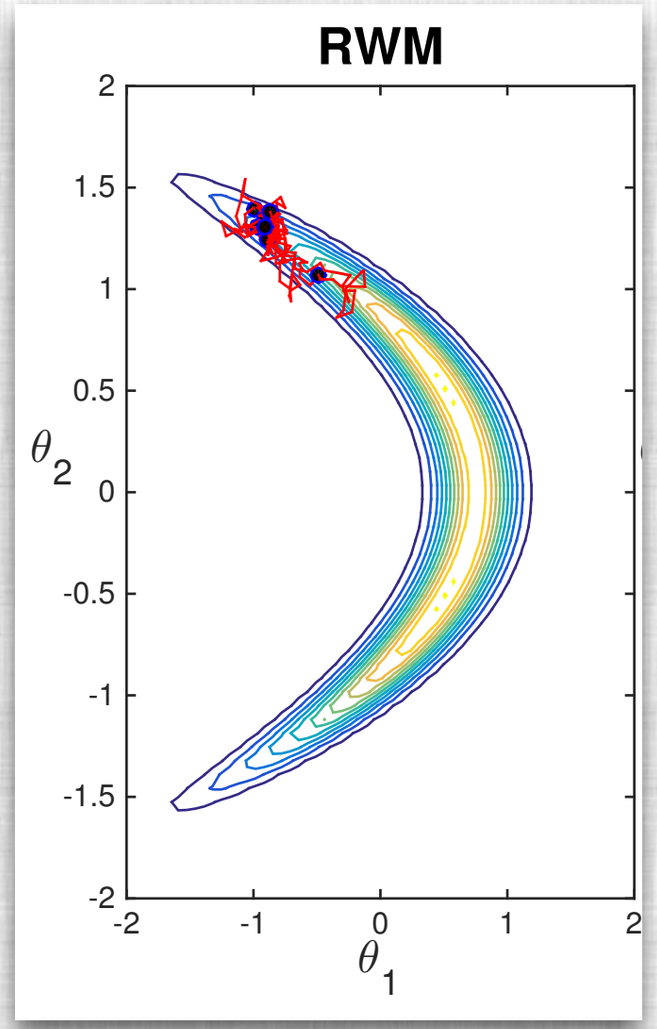


Computation

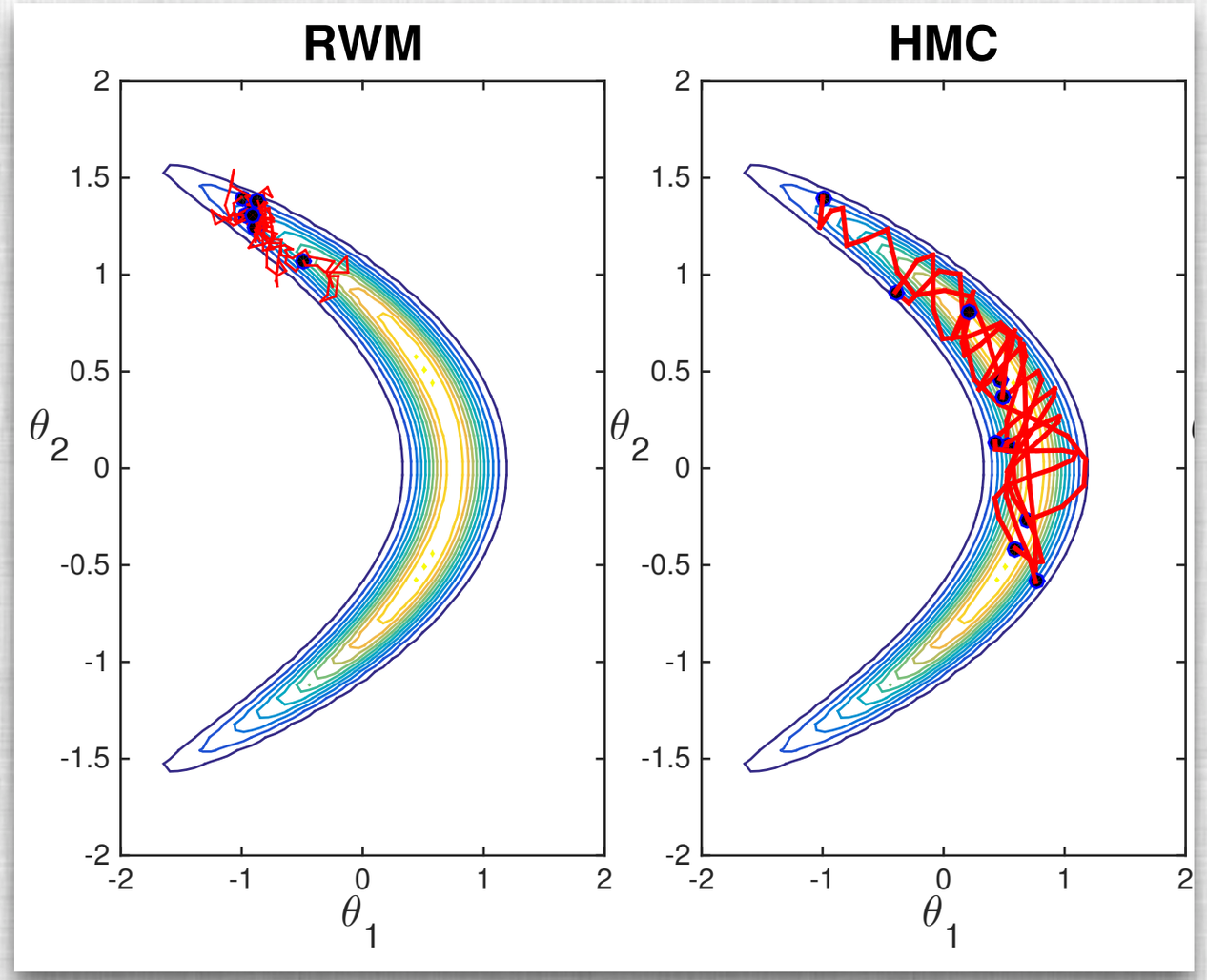
Markov Chain Monte Carlo



HMC and Its Variations



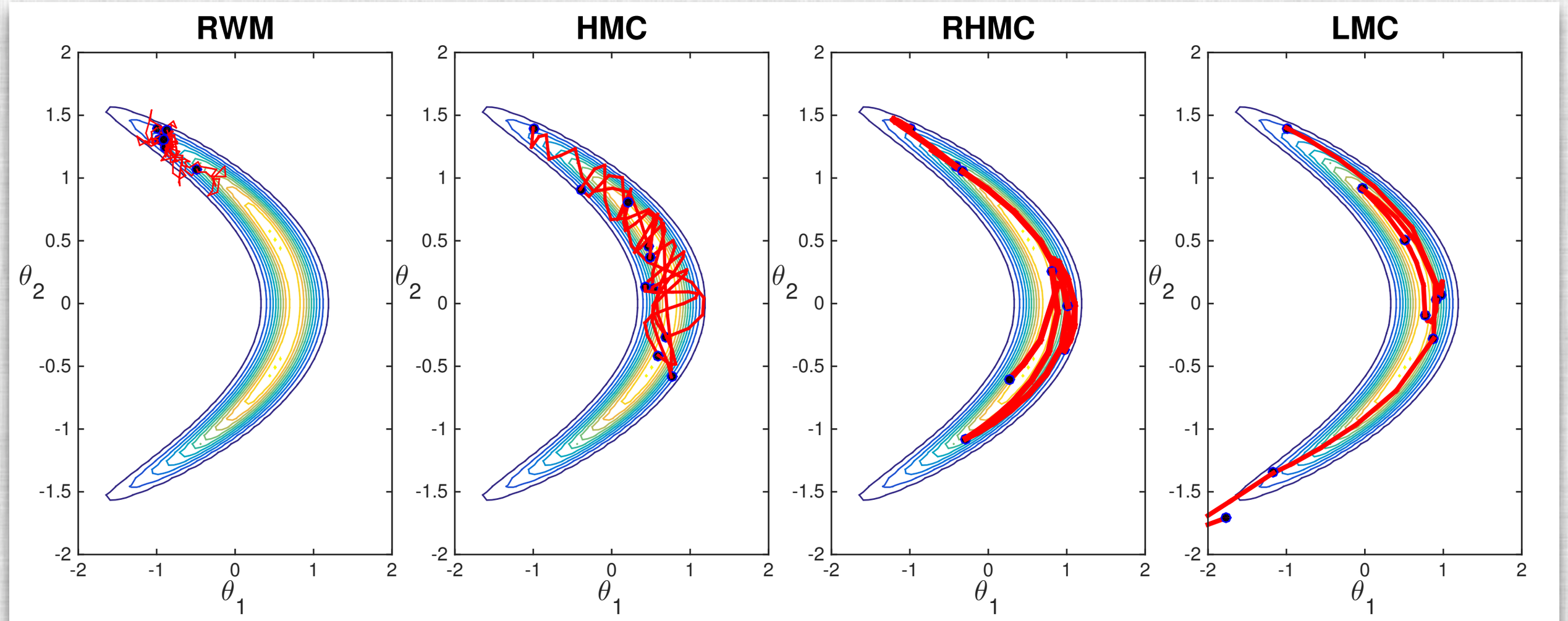
HMC and Its Variations



$$
\begin{aligned}
U(q) &= -\sum_{i=1}^{N} \log P(y_i|q) - \log P(q) \\
K(p) &= \frac{1}{2} p^T M^{-1} p \\
H(q,p) &= U(q) + \frac{1}{2} p^T M^{-1} p \\
P(q,p) &\propto \exp\left(-U(q) - \frac{1}{2} p^T M^{-1} p\right)
\end{aligned}
$$

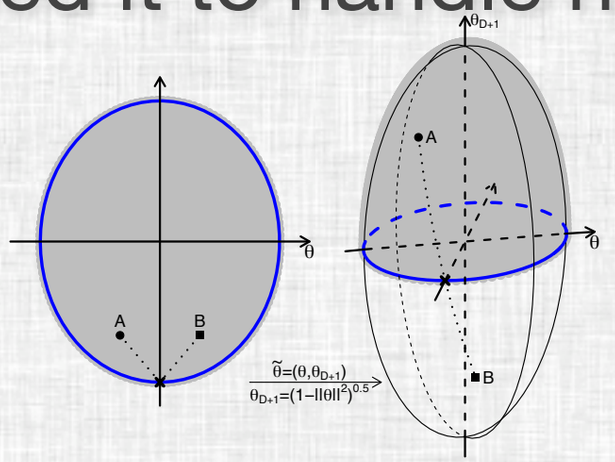
$$
\begin{aligned}
\frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} \\
\frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i}
\end{aligned}
$$

HMC and Its Variations



Spherical HMC

- Spherical Hamiltonian Monte Carlo is a Hamiltonian Monte Carlo algorithm on spheres
- It can be viewed as a special case of geodesic Monte Carlo (Byrne and Girolami, 2013), or manifold Monte Carlo methods (Girolami and Calderhead, 2011)
- We originally proposed it to handle norm constraints in sampling



- To sample $\boldsymbol{L}$ (or $\boldsymbol{L_t}$), we run a multi-spherical HMC on the product of spheres in parallel, $\prod_{i=1}^{D} \mathcal{S}_0^{i-1}$

Scalability

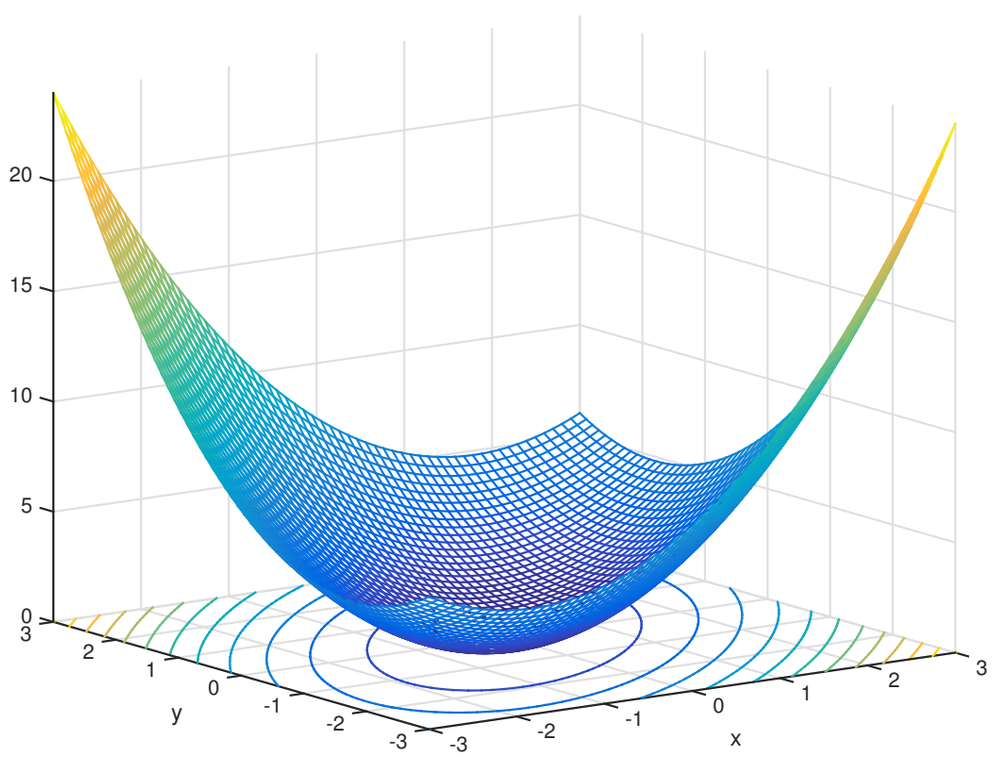
Neural Network Surrogate Hamiltonian Monte Carlo

- While HMC and its variants are more effective at exploring the parameter space, they require costly gradient evaluations at each iteration
- We have recently developed several methods for fast approximation of the gradient function using neural networks
- Our first strategy was to approximate the energy function, *U(q)* using neural network surrogate:

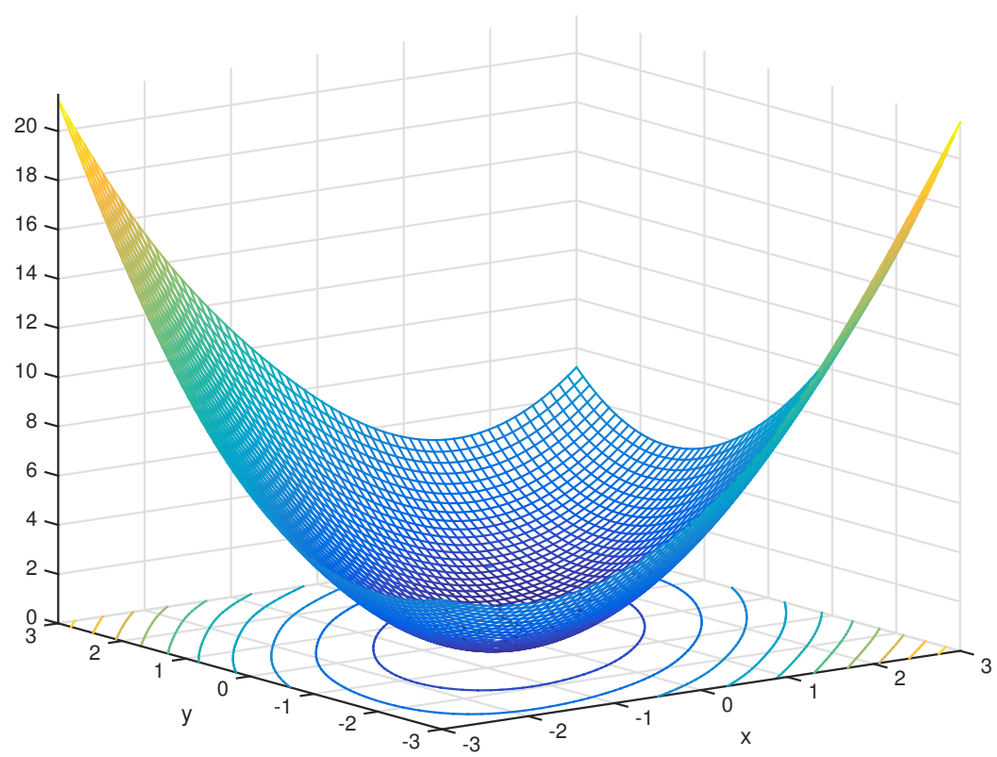
$$\tilde{H}(q,p) = \tilde{U}(q) + \frac{1}{2} p^T M^{-1} p$$

Neural Network Surrogate Hamiltonian Monte Carlo

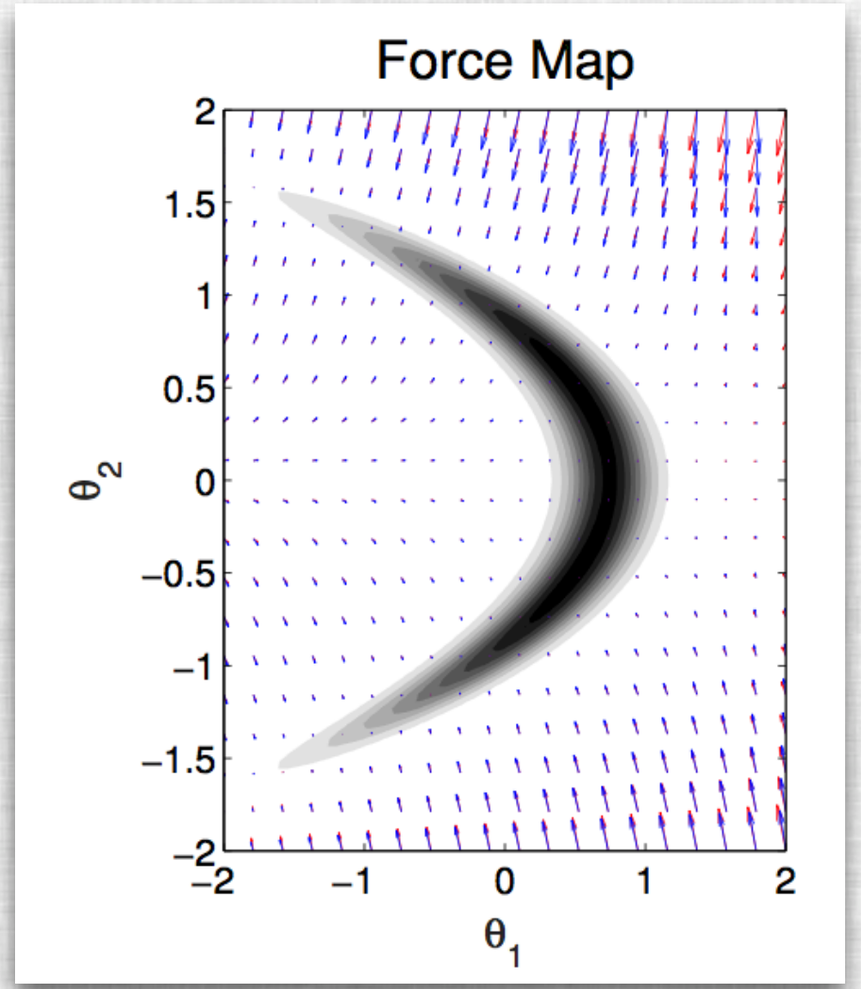
$U(q)$



$\tilde{U}(q)$



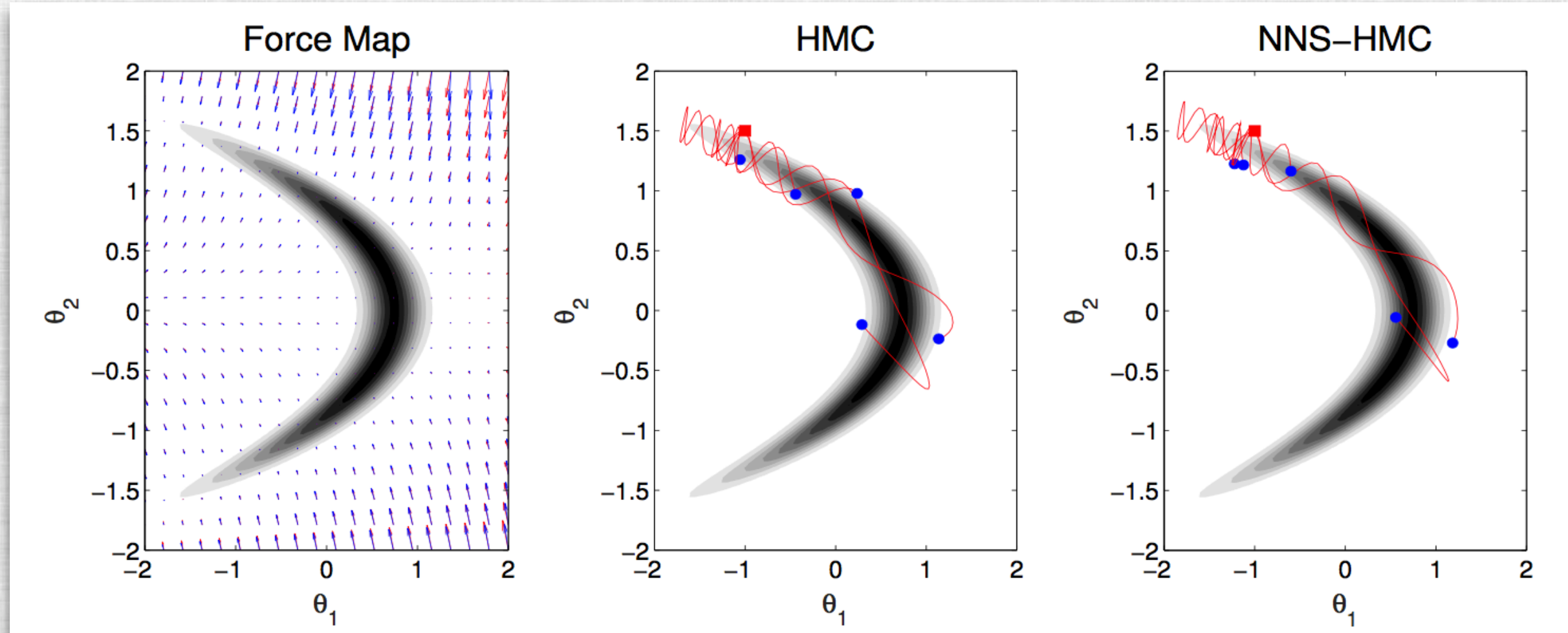
Neural Network Surrogate Hamiltonian Monte Carlo



We can approximate the gradient of the energy function (i.e., force) by the gradient of the neural network surrogate function

$$\nabla U \approx \nabla \tilde{U}$$

Neural Network Surrogate Hamiltonian Monte Carlo



Zhang et. al. (2016), *S&C*

Illustrative Examples

Experiment	Method	AP	s/Iter	min(ESS)/s	Spped-up
LR (Simulation)	HMC	0.6656	3.573E-01	1.45	1
	RMHMC	0.8032	3.794	0.06	0.04
	NNS-HMC	0.6726	1.364E-02	37.83	**26.09**
	NNS-RMHMC	0.8162	1.027E-01	2.17	1.50
LR (Bank Marketing)	HMC	0.8038	7.400E-02	6.51	1
	RMHMC	0.9210	6.305E-01	0.56	0.08
	NNS-HMC	0.7944	7.508E-03	58.22	**8.94**
	NNS-RMHMC	0.9064	2.741E-02	14.41	2.21
LR (Adult Data)	HMC	0.8300	7.898E-02	0.21	1
	RMHMC	0.8526	5.842E-01	1.06	4.81
	NNS-HMC	0.8096	9.914E-03	2.66	12.09
	NNS-RMHMC	0.8400	3.300E-02	18.68	**84.90**
Elliptic PDE	HMC	0.7077	1.568	0.061	1
	RMHMC	0.8014	4.388	0.228	3.74
	NNS-HMC	0.7138	7.419E-02	1.410	23.11
	NNS-RMHMC	0.6584	9.720E-02	4.375	**71.72**

Neural Network Gradient Hamiltonian Monte Carlo

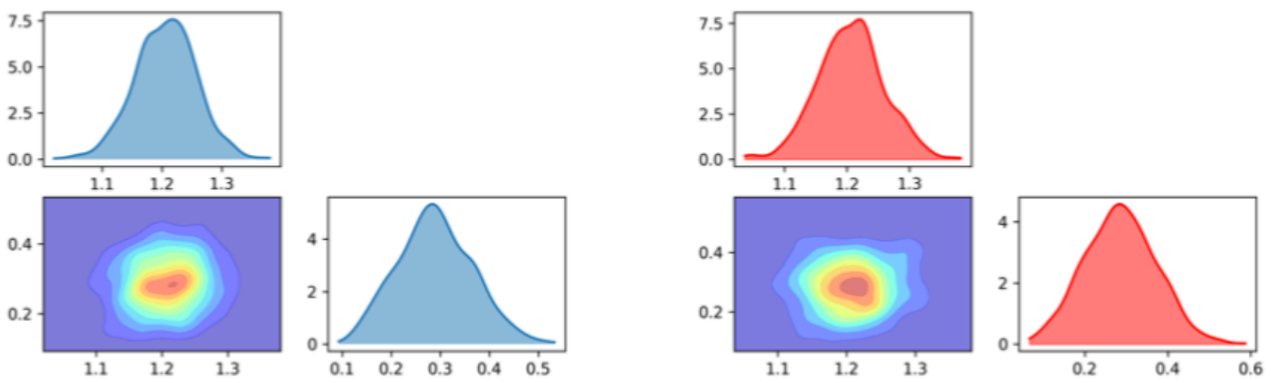
- In contrast to our previous work, we now fit a neural network to approximate directly ∇U using the training data $(q, \nabla U(q))$ collected from early period of HMC
- Once the approximate gradient is learned, the algorithm is exactly the same as classical HMC, but with neural network gradient $\widehat{\nabla U}$ replacing ∇U
- One benefit of this method is that we include all the data points over each leapfrog trajectory in our training sample

Illustrative Example

- Gaussian process regression model with two hyperparameters
- Gradient collected during the first 1000 draws is then used to train a neural network

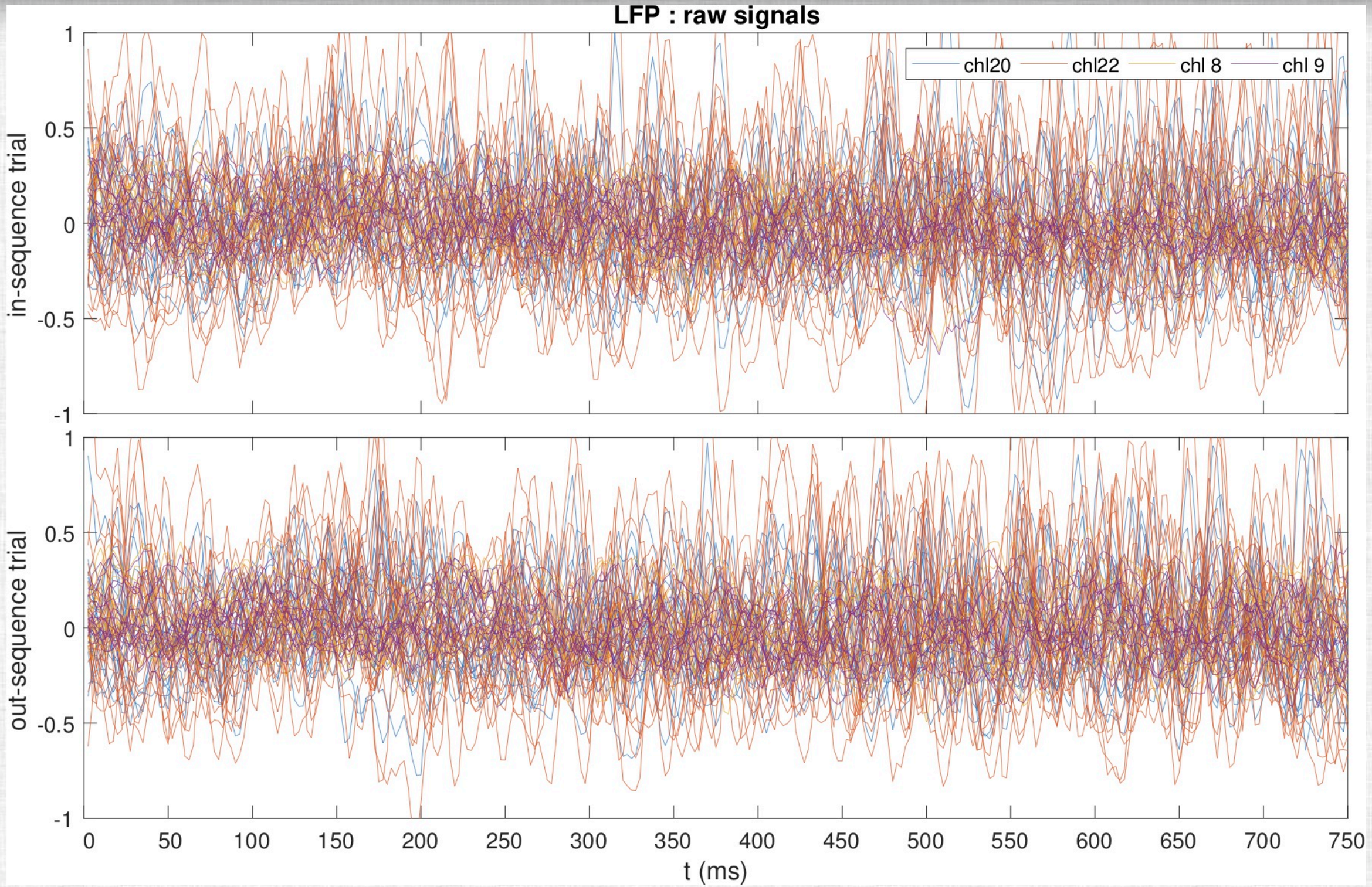
Method	AP	ESS	CPU time	Median ESS/s	Speed-up
Standard	0.83	(5135, 5754, 7635)	1834s	3.14	1
NNg	0.84	(4606, 6172, 7741)	50s	123.4	**39.3**

AP: acceptance probability
ESS: effective sample size (min, median, max) after removing 10% burn-in

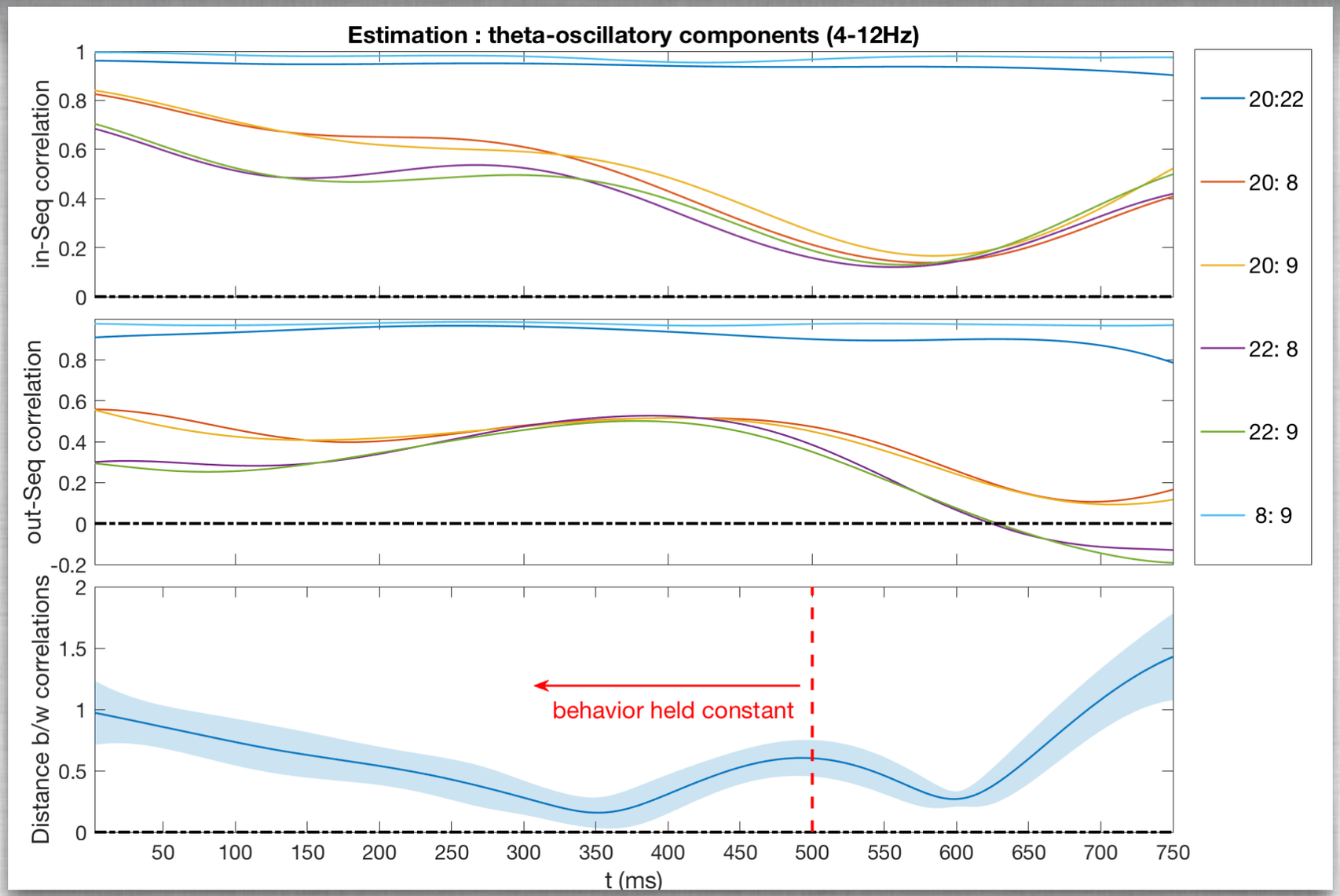


Results

Raw LFP Signals



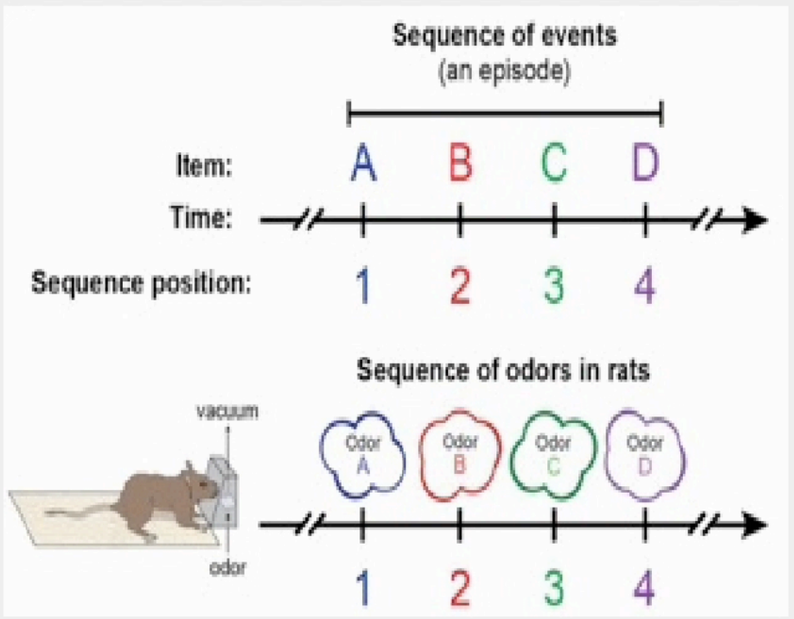
Dynamic Covariance Modeling for LFP

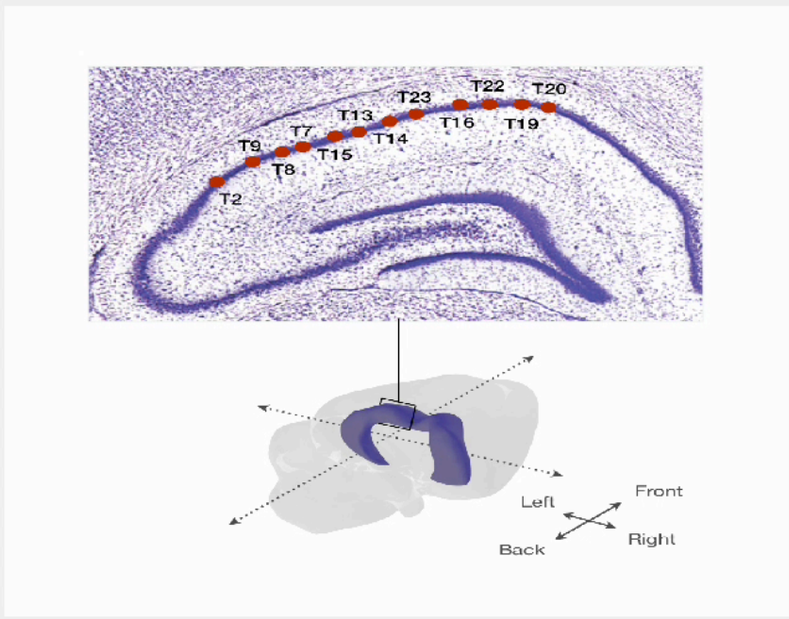


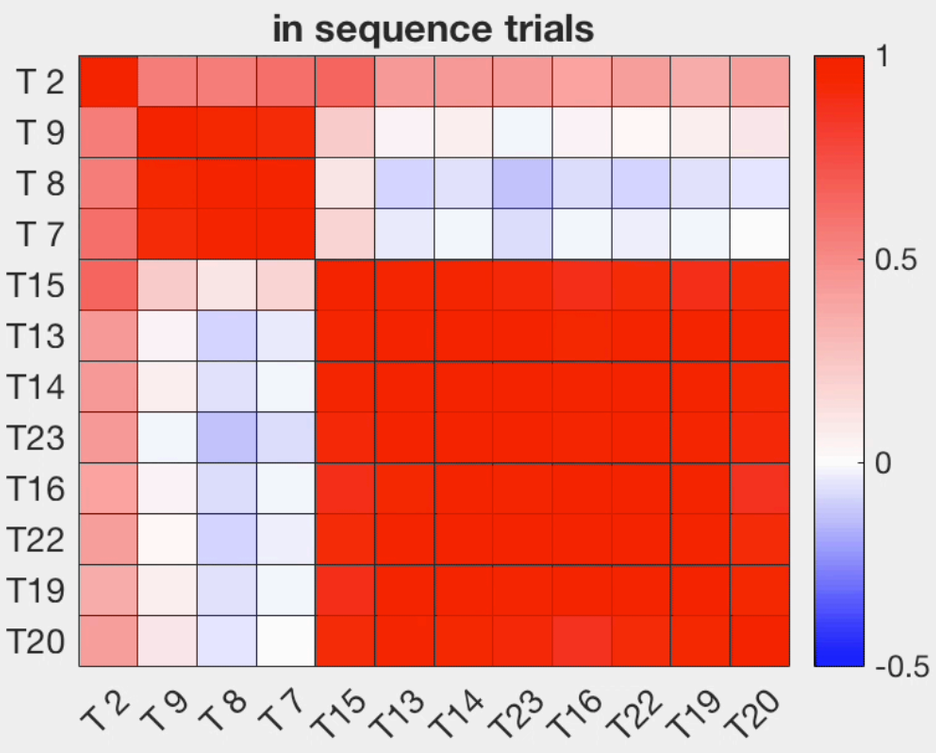
- Nearby electrodes (20:22 and 8:9) displayed remarkably higher correlations in LFP compared to distant pairs (20:8, 20:9, 22:8, and 22:9).
- InSeq and OutSeq activity was very similar at the beginning (e.g., before 350ms) but maximally different at the end.

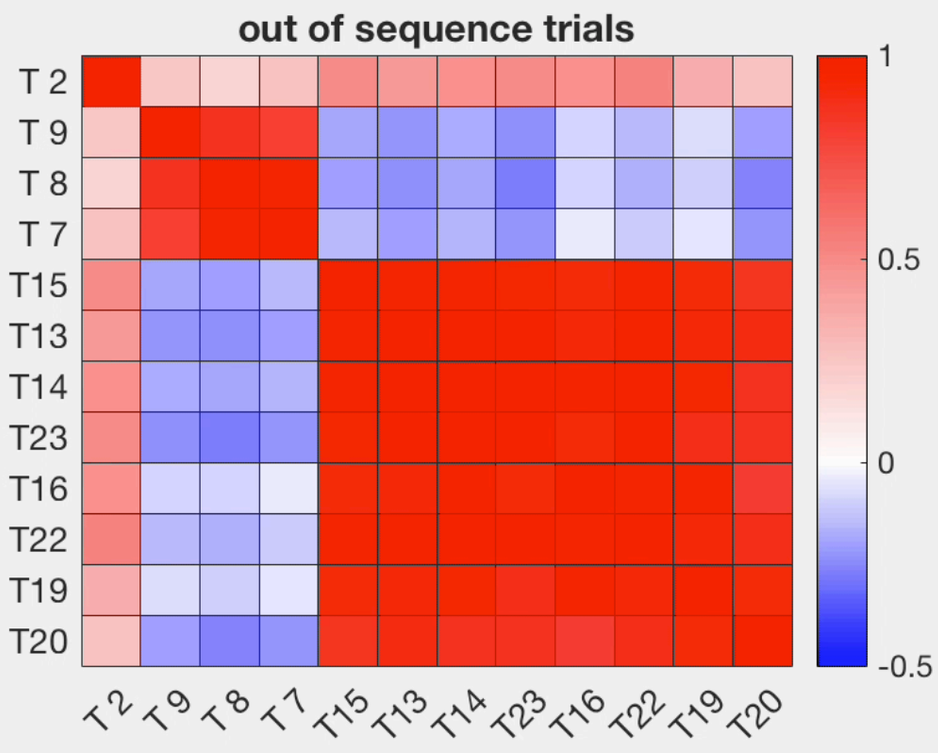
Results for All 12 Channels

t = 2.5ms









ACKNOWLEDGEMENTS

- Collaborators

Norbert Fortin, PhD

Shiwei Lan, PhD

Hernando Ombao, PhD

Pierre Baldi, PhD

- Postdocs and Students

Gabriel Elias, Ph.D.

Lingge Li

Tian Chen

Andrew Holbrook

Forest Agostinelli

- Funding sources

DMS 1622490

R01MH115697

Thank You!