Likelihood-free Posterior Density Learning for Uncertainty Quantification in Inference Problems

Rui Zhang, Oksana Chkrebtii, Dongbin Xu

30th April, 2026

The Ohio State University

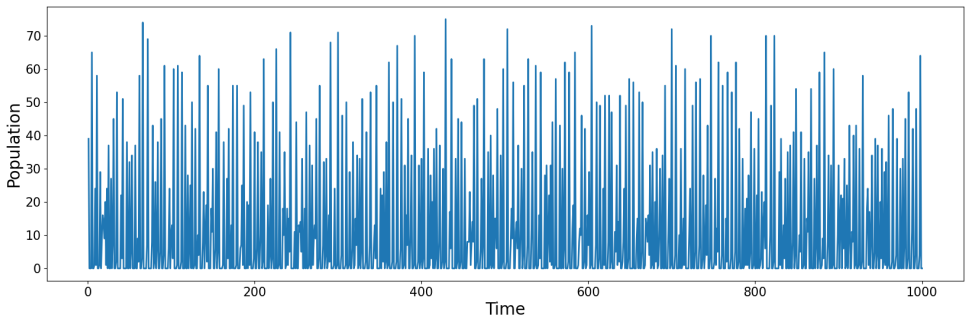
Introduction

Motivation: Intractable Likelihoods

- Generative models in biology, ecology, and engineering often lack closed-form likelihoods
- When are likelihoods intractable?
  - Consist of combinatorially large number of components (e.g., lattice models)
  - High-dimensional integrals over latent variables (e.g., state-space models, agent-based models)
  - Include intractable normalizing constant (e.g., models on manifolds)
- **Goal:** A fast, accurate approach for Bayesian inference without explicit likelihood evaluation

A Simple Example

- Ricker model: ecological model that describes the density-dependent dynamics of an animal population
- Population density $N(t+1) = N(t)e^{-N(t)+\epsilon(t)+\eta}$, where $\epsilon(t) \overset{\text{ind}}{\sim} \mathcal{N}(0, \sigma^2)$ is process noise
- Observations of population $y(t) \overset{\text{i.i.d.}}{\sim} \text{Poisson}(\delta N(t))$



- Used by Wood (2010) to model the data from Nicholson's blowfly experiment
- Likelihood evaluation requires high-dimensional integration over the latent population trajectories

Simulation-based inference methods

- Approximate Bayesian computation (ABC, Tavaré et al. 1997)
  - MCMC methods that target an approximate posterior distribution
  - Discrepancy between simulated summarized and observed data replaces likelihood evaluation in sampling algorithm
  - Trade-off between accuracy and Monte Carlo errors (Fearnhead and Prangle, 2012)
  - Requires non-trivial tuning
- Synthetic likelihood estimation (SLE) (Wood, 2010)
  - Replace likelihood with a multivariate normal density of summary statistics computed from simulated data
  - Model misspecification can lead to estimation bias

Deep learning for likelihood-free inference

Neural network (NN) learning for likelihood-free estimation

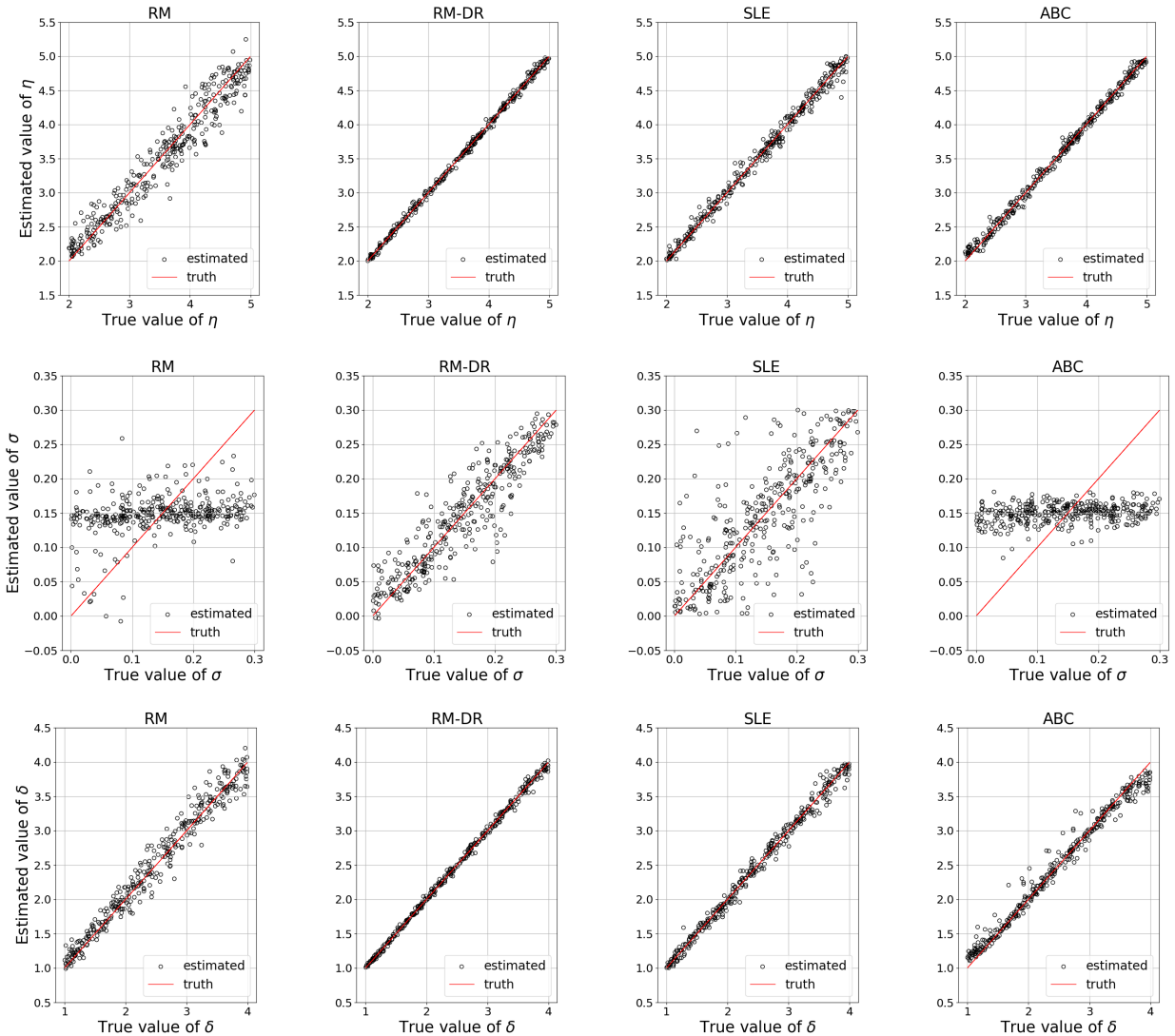
- Reconstruction-map estimation
  - Estimator obtained by learning mapping from the data space to the parameter space by training a NN model on a large number of simulated data-parameter pairs (Lenzi et al., 2023; Sainsbury-Dale et al., 2024)
  - Zhang et al. (2025a) formalizes the method, provides a connection to Bayes estimation, and introduces dimension reduction (RM-DR)
  - Trade-off between accuracy and Monte Carlo errors
  - Bootstrap estimates of sampling distribution are computationally expensive

Example: Ricker model

Parameters of interest $\theta = (\eta, \sigma, \delta)^\top$. RM-DR, SLE, and ABC are implemented using the summary statistics in Wood (2010)

- sample mean and sample autocovariance with lag h from $0 - 5$ —summaries of time series data
- number of zeros observed —insights into distribution of Poisson data
- coefficients of cubic regression of ordered differences on ordered observed values —summarize shape of marginal distribution
- coefficients of autoregression $(y(t+1))^{0.3}$ on $y(t)^{0.3}$ and $y(t)^{0.6}$ —information about dynamic structure

Example: Ricker model



Estimates versus simulation values of η, σ, and δ (Zhang et al., 2025a)

Deep learning for likelihood-free inference

- Mixture density networks (MDN, Bishop, 1994)
  - Train NN to define Gaussian mixture that approximates the posterior distribution
  - Bias when prior is diffuse and training data are sparse
  - Variant by Papamakarios and Murray (2016); Lueckmann et al. (2017) suffer from numerical instability, lack of generality, requiring multiple rounds of neural network training and significant computational cost

Deep learning for likelihood-free inference

- Normalizing flows (NF)
  - NF (Rezende and Mohamed, 2015; Papamakarios et al., 2021) transform a simple base distribution into a target via sequence of invertible NN-parametrized transformations
  - Likelihood-free conditional invertible NNs (Winkler et al., 2019; Ardizzone et al., 2019) require global invertibility $\Rightarrow$ sensitive to initialization and challenging to tune
  - Automatic posterior transformation Greenberg et al. (2019): inference as a density ratio estimation problem, requiring multiple rounds of training

Our contribution (Zhang et al., 2025b)

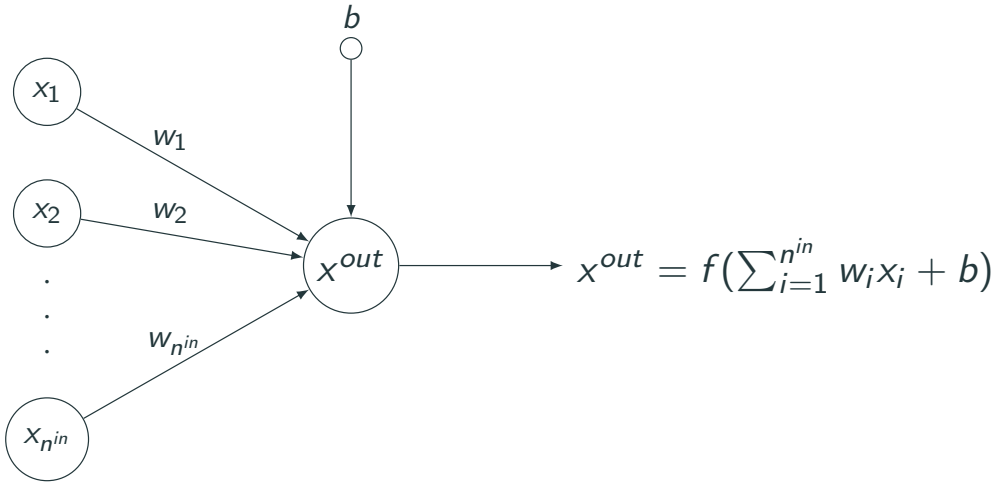
- **Kernel-adaptive synthetic posterior estimation (KASPE)**
  - Learns auxiliary parameters defining a parametric approximation to the exact posterior
  - Adaptive sampling used to generate synthetic training data
  - Built-in uncertainty quantification
- Show that KASPE is consistent under some conditions
- Connect KASPE to expectation propagation (EP, Minka, 2001) and show MDN estimation as special case

Background and Notation

Neural network (NN) model

- Computational model that uses interconnected *neurons* in a layered structure; mimic the way human brain works
- x^{in}: input variables, w: weights

$$x^{out} = f(w^\top x^{in} + b).$$



A neuron and its components; $x_1, \ldots, x_{n^{in}}$ are input nodes/variables with weights $w_1, \ldots, w_{n^{in}}$; b is the bias parameter; f is the activation function; and x^{out} is the output.

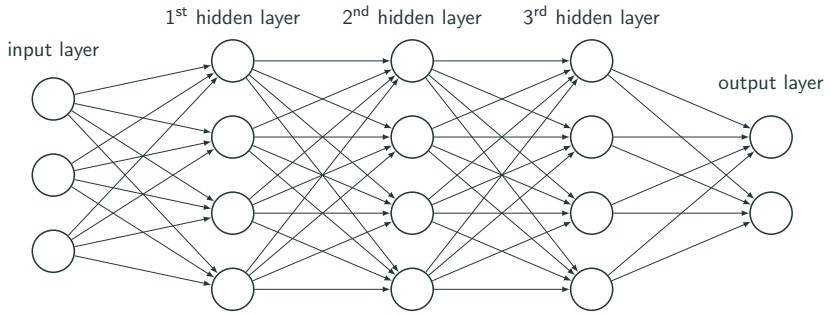
Multi-layer NN

- $\omega := (A_1, \ldots, A_L)$ are parameters of the neural network (NN)
- $\mathbf{N}(\cdot, \omega) : \mathbb{R}^{n^{in}} \to \mathbb{R}^{n^{out}}$ is the vector-valued NN function

$$\mathbf{N}(\cdot, \omega) = f_L \circ h_{A_L} \circ \cdots \circ f_1 \circ h_{A_1},$$

  composition of alternating linear and activation functions
- Output for this L-layer NN is

$$x^{out} = \mathbf{N}(x^{in}, \omega).$$



An example of a 4-layer neural network with 3 input nodes, 4 neurons per hidden layer, and 2 output neurons.

Expectation propagation (EP)

Inference as an optimization problem over a class of densities targeting posterior approximation (Minka, 2001)

- Parametric family of densities $\mathcal{Q} = \{q(\cdot \mid \eta) : \eta \in \mathcal{E}\}$ over θ
- Want $\eta(y_0) \in \mathcal{E}$ that produces q closest to $\pi(\cdot \mid y_0)$

$$\widehat{\pi}_{EP}(\cdot \mid y_0) = q(\cdot \mid \widehat{\eta}(y_0))$$
$$\widehat{\eta}(y_0) = \arg\min_{\eta \in \mathcal{E}} \mathrm{KL}\left(\pi(\cdot \mid y_0) \mid q(\cdot \mid \eta)\right)$$

- Closeness measured by KL divergence where $\mathrm{KL}\left(\pi(\cdot \mid y_0) \mid q(\cdot \mid \eta)\right) := \mathrm{E}^{\theta \sim \pi(\theta|y_0)}\left(\log \frac{\pi(\theta|y_0)}{q(\theta|\eta)}\right)$, and $\eta(\cdot)$ is called EP parametrization function

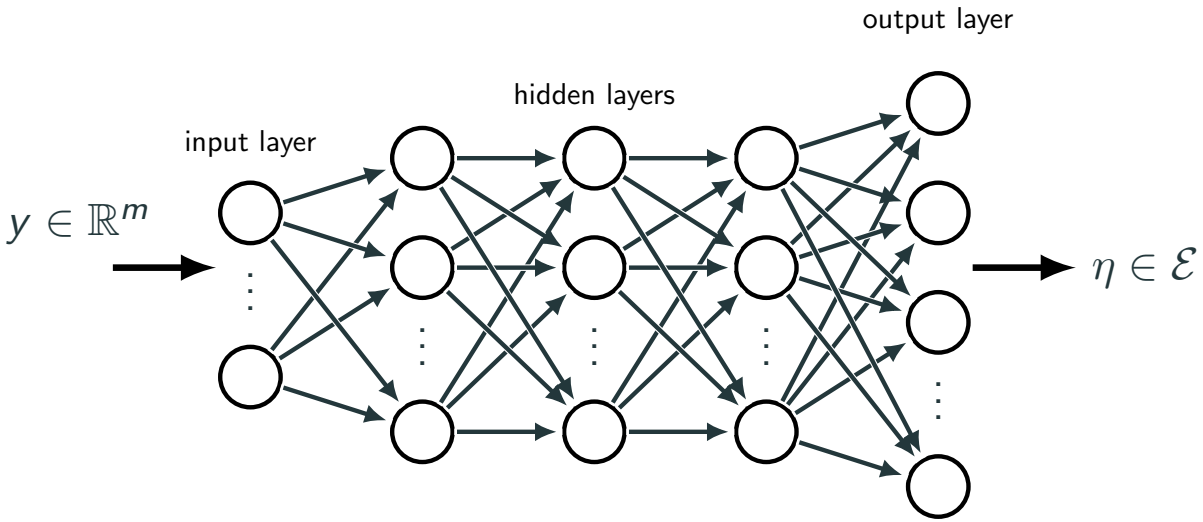
Expectation propagation (EP)

- Mean-field assumption: posterior can be approximated as a product of independent factors
- Iteratively update these factors (pass messages between the factors) to minimize the KL divergence
- If posterior density is contained in the family of densities and no mean field assumption is used, EP can recover true posterior density
- Limitation: tractable likelihood, factorization assumption, non-trivial analytical work to derive factor updates, convergence and scalability issues, etc

Kernel-adaptive synthetic posterior estimation (KASPE)

Main idea

- Choose parametric family of densities $\mathcal{Q} = \{q(\cdot \mid \eta) : \eta \in \mathcal{E}\}$
- Draw synthetic data $(\theta_i, y_i) \overset{\text{ind}}{\sim} \pi(\theta)p(y \mid \theta)$ offline and retain with probability $K(\frac{y_i - y_0}{h})$ until effective sample size n
- Define the vector-valued *NN function* $\mathbf{N}(\cdot, \omega) : \mathbb{R}^m \to \mathcal{E}$, where vector $\omega \in \Omega$ denotes the NN weight parameters



Main idea

- Train a deep neural network on synthetic data, by minimizing loss function

$$Q_n(\omega) = -\frac{1}{n}\sum_{i=1}^{n} w_i \log q(\theta_i \mid \mathbf{N}(y_i, \omega))$$

(penalizes NN weights that lead to small average training data densities given the simulation parameters)

- KASPE is the parametric function q indexed by parameters $\widehat{\eta}$ obtained by mapping observed data y_0 to $\mathcal{E}$ through a NN trained on the synthetic data, i.e.

$$\widehat{\pi}_n(\cdot \mid y_0) = q(\cdot \mid \mathbf{N}(y_0, \widehat{\omega}_n)),$$

$$\text{where } \widehat{\omega}_n \in \arg\min_{\omega \in \Omega} \; -\frac{1}{n}\sum_{i=1}^{n} w_i \log q(\theta_i \mid \mathbf{N}(y_i, \omega)).$$

Comments

- Must be able to sample a large number of input-output pairs from the model $p(y \mid \theta)$
- Parametric family of densities is application specific
- As in ABC, data dimension m impacts the quality of estimation, so summarization $S(y)$ is done in practice
- Tuning parameter h - should be as small as possible, balancing with the need for a suitable acceptance rate for synthetic training samples

KASPE Algorithm

Algorithm 2 Algorithm for KASPE-DR

Input: observed data y_0, likelihood function $p(\cdot \mid \cdot)$, prior density $\pi(\cdot)$, summary function $S(\cdot)$, family of distributions $q(\cdot \mid \cdot)$, NN function $\mathbf{N}(\cdot, \cdot)$, bandwidth $h > 0$, integer $n > 0$

Output: $\widehat{\pi}_n(\cdot \mid y_0)$

1: **for** $i = 1$ to n **do**
2: sample $\theta_i \sim \pi(\cdot)$
3: sample $y_i \mid \theta_i \sim p(\cdot \mid \theta_i)$
4: let $s_0 = S(y_0)$ and $s_i = S(y_i)$ for $i = 1, \ldots, n$
5: with probability $K(\frac{s_i - s_0}{h})$, set $w_i = 1$; otherwise, we set $w_i = 0$
6: **end for**
7: use numerical optimization to solve

$$\widehat{\omega}_n \in \arg\min_{\omega \in \Omega} \; -\frac{1}{n} \sum_{i=1}^{n} w_i \log q(\theta_i \mid \mathbf{N}(s_i, \omega))$$

8: set posterior density estimate $\widehat{\pi}_n^{DR}(\cdot \mid s_0) = q(\cdot \mid \mathbf{N}(s_0, \widehat{\omega}_n))$

Properties and Connection with Existing Approaches

Assumptions

We make the following assumptions

- The parameter space Ω of NN weights is compact
- $\mathbf{N}(y, \omega)$ is continuous in ω for any fixed $y \in \mathcal{Y}$
- For any $\omega \in \Omega$, finite expected training loss function $Q_0(\omega) = \mathbb{E}_{(\theta,y,w)}\left[Q_n(\omega)\right] = -\mathbb{E}_{(\theta,y,w)}\left[w \log q(\theta \mid \mathbf{N}(y, \omega))\right]$ has a set of minimizers $\Omega_0 = \underset{\omega\in\Omega}{\operatorname{argmin}}\, Q_0(\omega)$ that satisfies, for any $\omega_a, \omega_b \in \Omega_0$, $\mathbf{N}(\cdot, \omega_a) = \mathbf{N}(\cdot, \omega_b)$
- Unique induced NN at the minimizers denoted by $\mathbf{N}_0(\cdot)$
- $\sup_{\omega\in\Omega} |Q_n(\omega) - Q_0(\omega)| \xrightarrow{p} 0$ as $n \to \infty$.

Convergence and connection to EP

Theorem 1: KASPE parameterization estimator $\widehat{\eta}_n(\cdot)$ converges pointwise in probability to the function $\mathbf{N}_0(\cdot)$ as the number of synthetic weighted training data-parameter pairs $n \to \infty$. That is, for each fixed $y \in \mathcal{Y}$:

$$\widehat{\eta}_n(y) \xrightarrow{p} \mathbf{N}_0(y), \text{ as } n \to \infty.$$

Theorem 2: Suppose there exists an EP parameterization estimator $\eta(\cdot)$ within $\mathcal{M} = \{\mathbf{N}(\cdot, \omega) : \omega \in \Omega\}$. Then the KASPE parameterization estimator $\widehat{\eta}_n(\cdot)$ converges pointwise in probability to the EP parameterization estimator $\eta(\cdot)$ as the number of synthetic weighted training data-parameter pairs $n \to \infty$. That is, for each fixed $y \in \mathcal{Y}$:

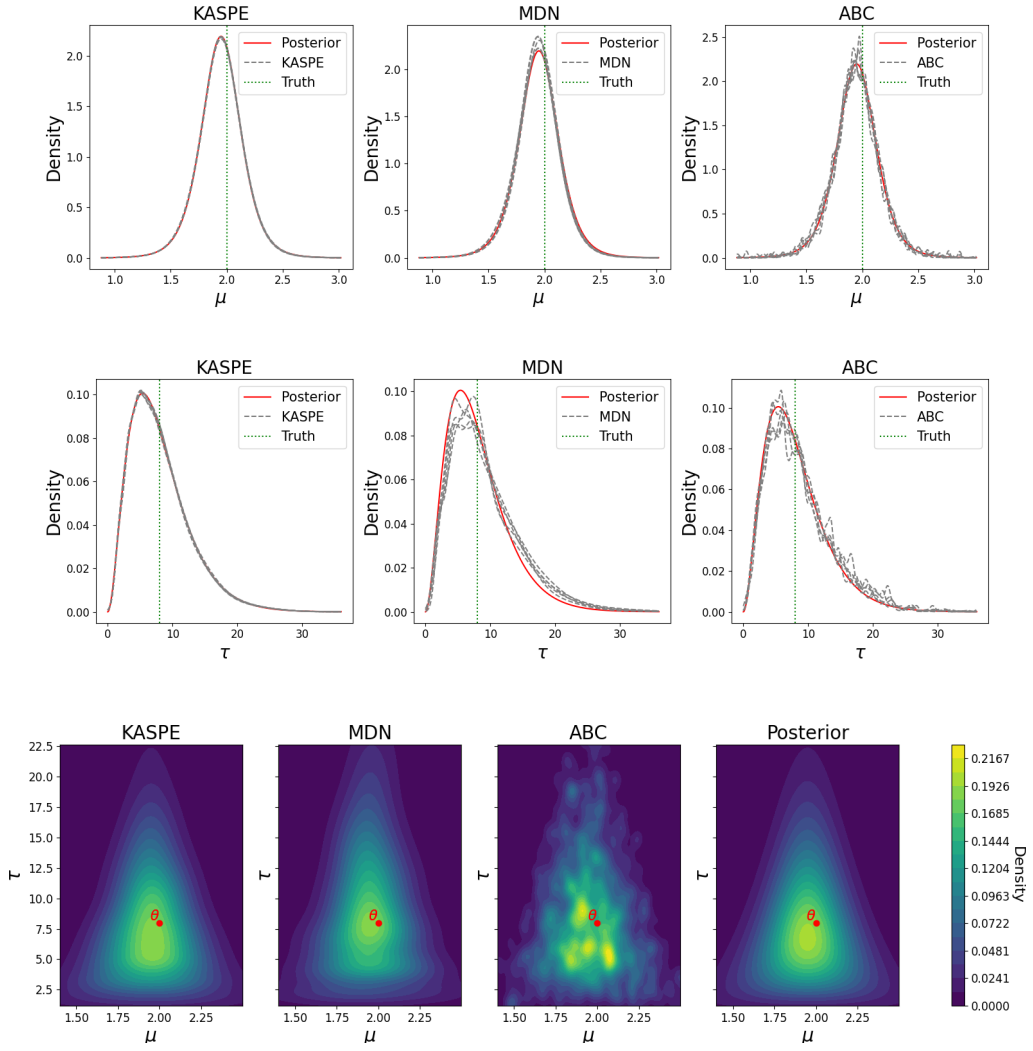
$$\widehat{\eta}_n(y) \xrightarrow{p} \eta(y), \text{ as } n \to \infty.$$

Numerical Experiments

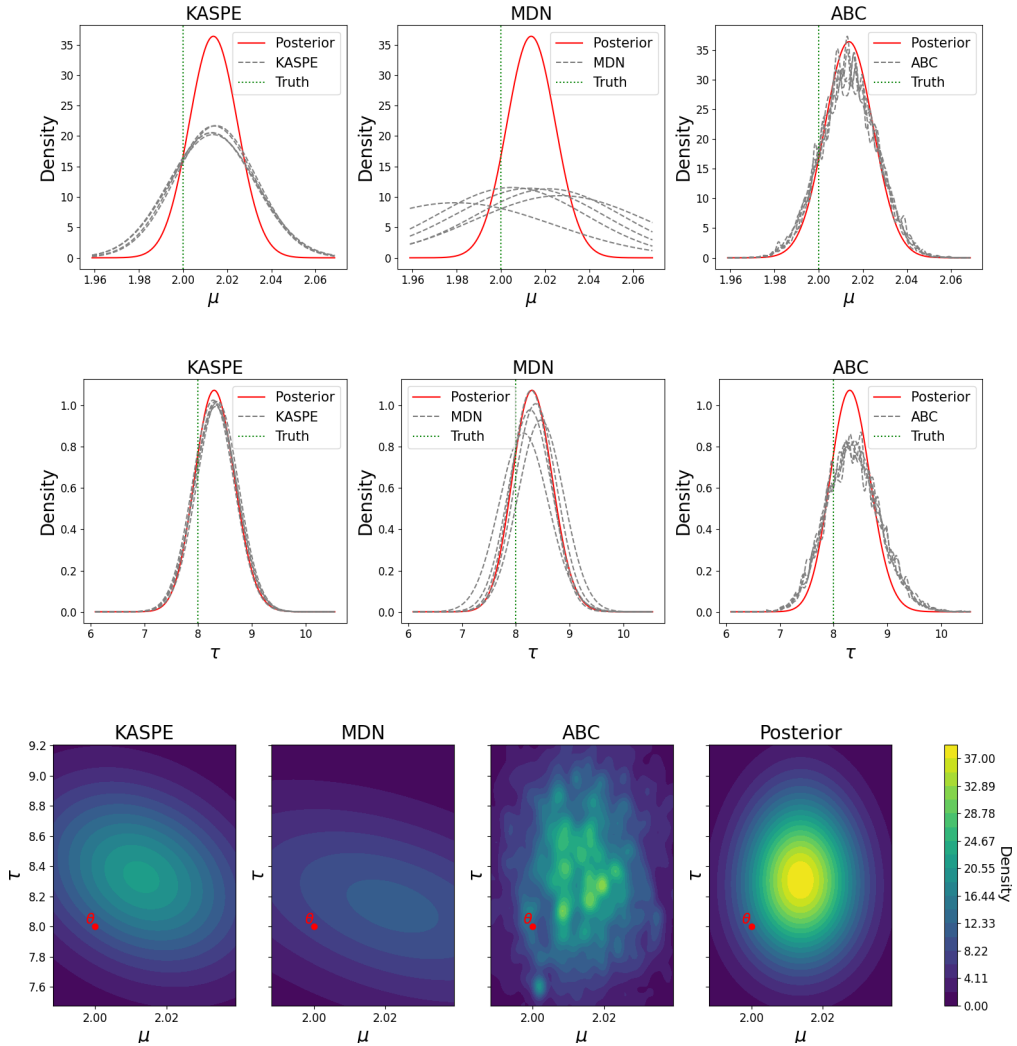
Heavy-tailed posterior density

- Observed data $y = (y(1), \ldots, y(m))^\top$, where $y(i) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \tau^{-1})$
- Conjugate normal-gamma prior $(\mu, \tau) \sim \text{NG}(\eta, \lambda, \alpha, \beta)$, defined as: $\tau \sim \text{Gamma}(\alpha, \beta)$, and $\mu \mid \tau \sim \mathcal{N}(\eta, (\lambda\tau)^{-1})$
- Marginal distribution of μ follows a non-standardized Student's t-distribution
- Posterior will also follow a normal-gamma distribution
- Sample mean and standard deviation used as summaries

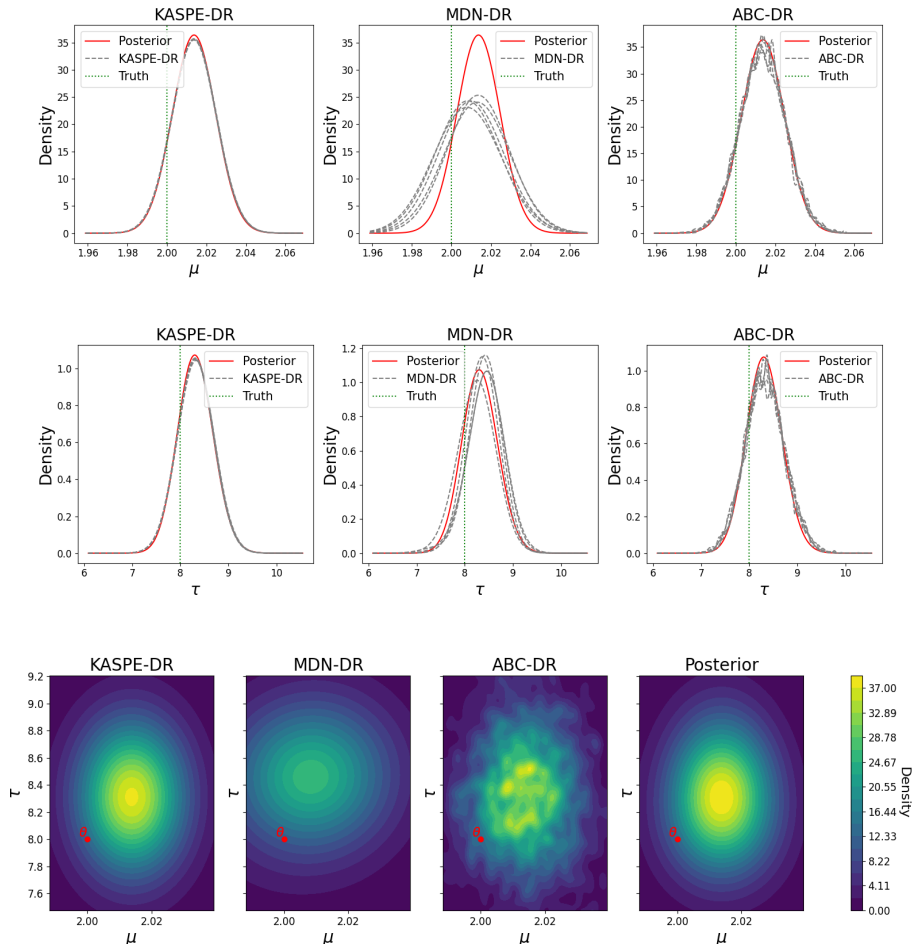
Results, $m = 4$


KASPE
MDN
ABC
Posterior
KASPE
MDN
ABC
Truth
Density
μ
τ
θ
Posterior
0.2167
0.1926
0.1685
0.1444
0.1204
0.0963
0.0722
0.0481
0.0241
0.0000
Density

Results, $m = 1000$



Results, $m = 1000$, dimension reduction



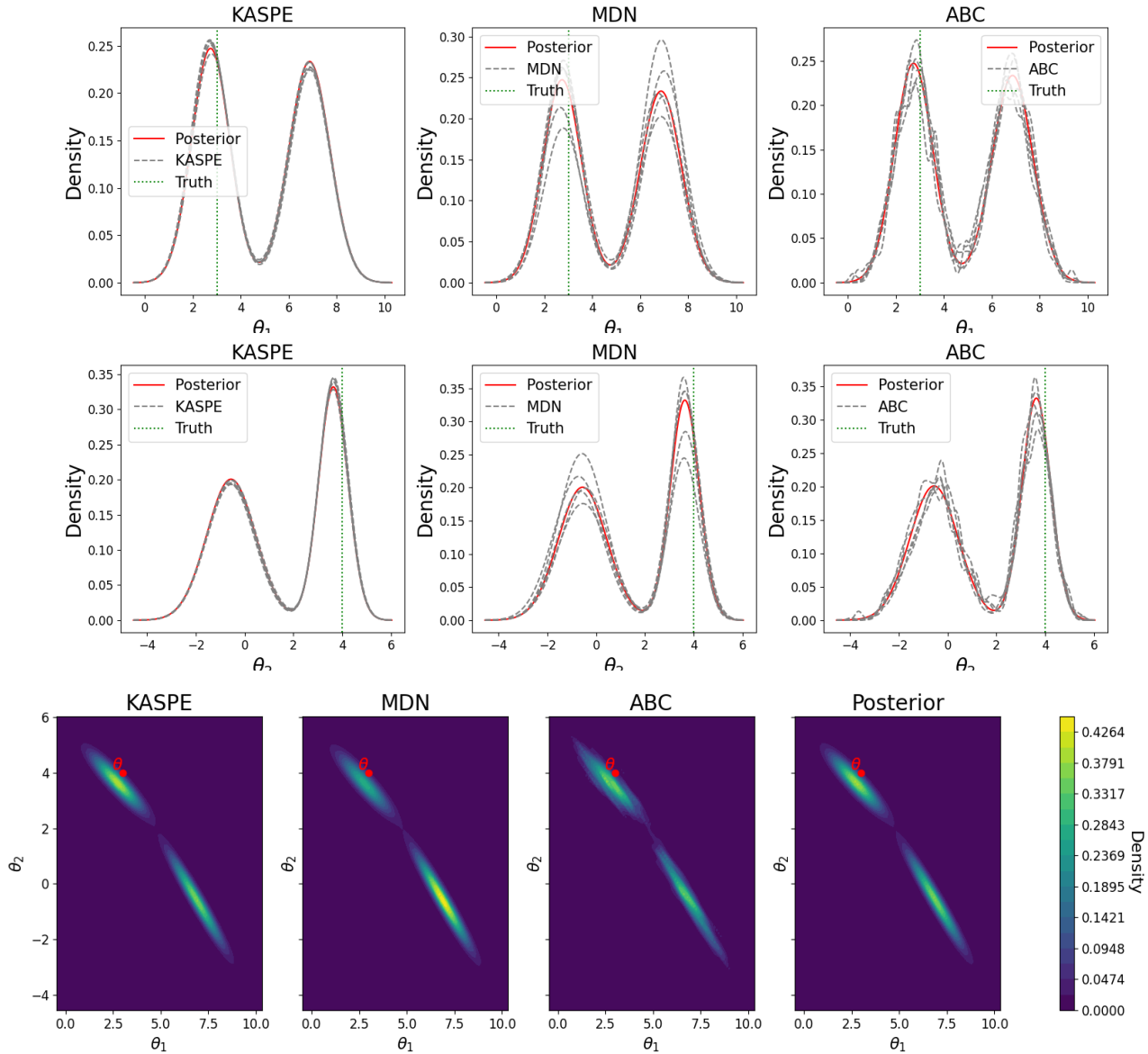
Multi-modal posterior distribution

- Gaussian mixture model with unknown mean components $\theta = (\theta_1, \theta_2)^\top$
- Time-dependent covariate vectors $v(t) = (t, t^2)^\top$ and $r(t) = (t^2, \sqrt{t})^\top$
- Data is a time series $y = (y(t_1), y(t_2), \ldots, y(t_m))^\top \in \mathbb{R}^m$ and the likelihood for y given θ follows a Gaussian mixture distribution:

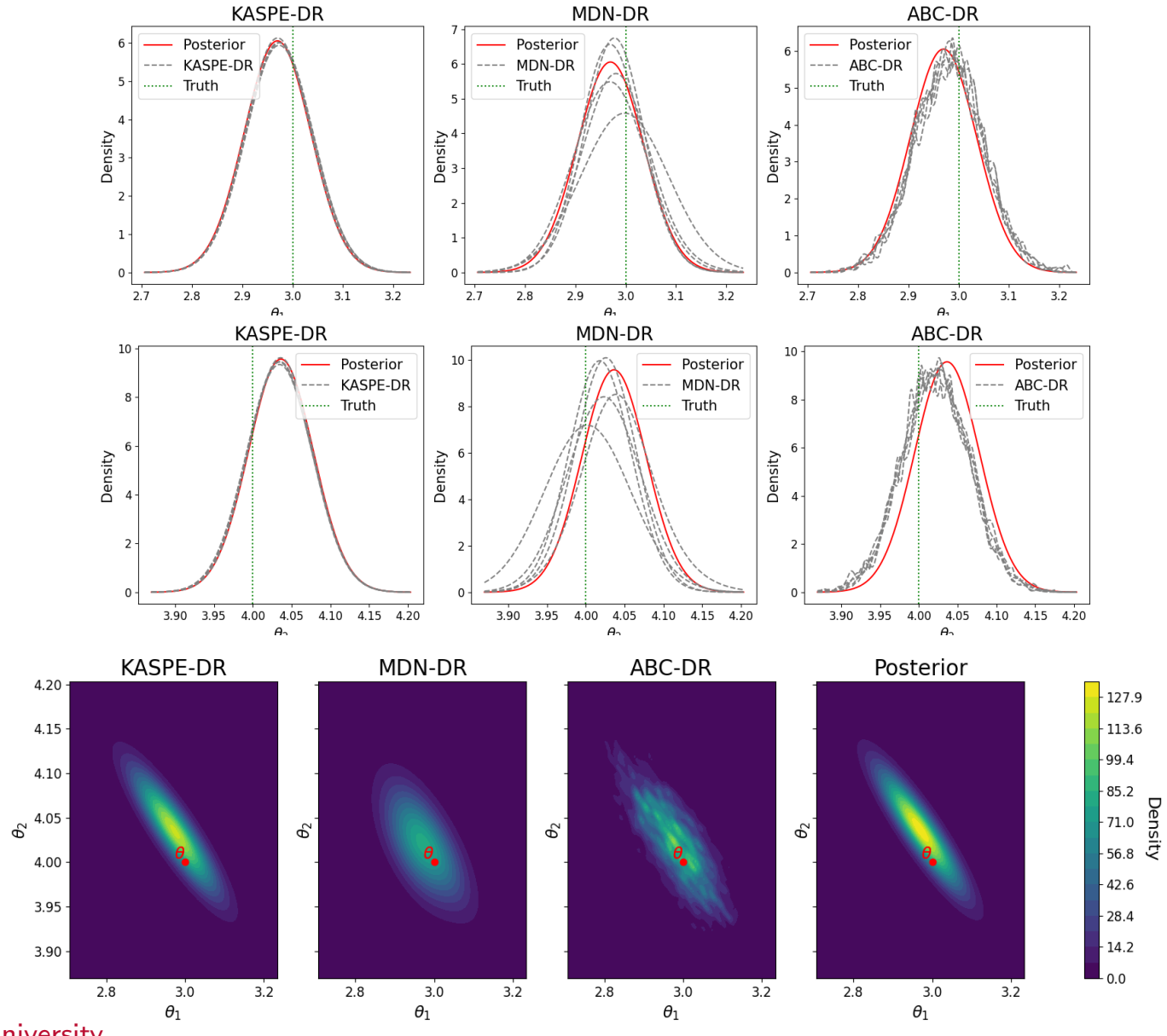
$$y \mid \theta \sim p_1 \mathcal{N}(v\theta, \Sigma_1) + p_2 \mathcal{N}(r\theta, \Sigma_2), \tag{1}$$

- We use least square estimates in each component as summary statistics

Results, $m = 4$


KASPE
MDN
ABC
Posterior
KASPE
MDN
ABC
Truth
Density
θ_1
θ_2
Posterior
0.4264
0.3791
0.3317
0.2843
0.2369
0.1895
0.1421
0.0948
0.0474
0.0000

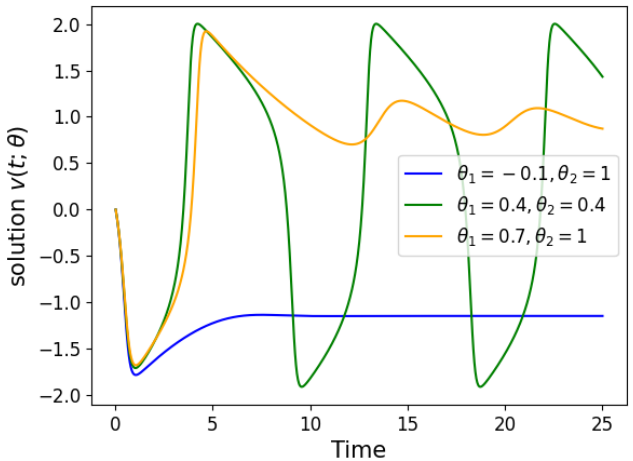
Results, $m = 1000$, dimension reduction



FitzHugh–Nagumo model

A nonlinear system defined via coupled ODEs; model dynamically spiking membrane potential of a biological neuron

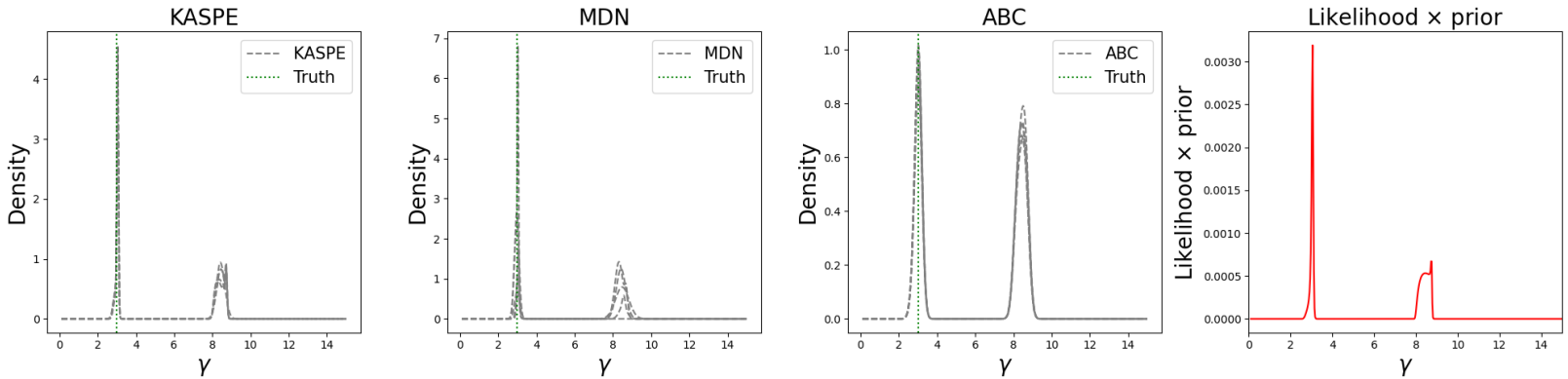
$$\begin{cases} \dfrac{\mathrm{d}v}{\mathrm{d}t} = \gamma\left(v - \dfrac{v^3}{3} + r + \zeta\right) \\ \dfrac{\mathrm{d}r}{\mathrm{d}t} = -\dfrac{1}{\gamma}\left(v - \theta_1 + \theta_2 r\right) \end{cases}$$



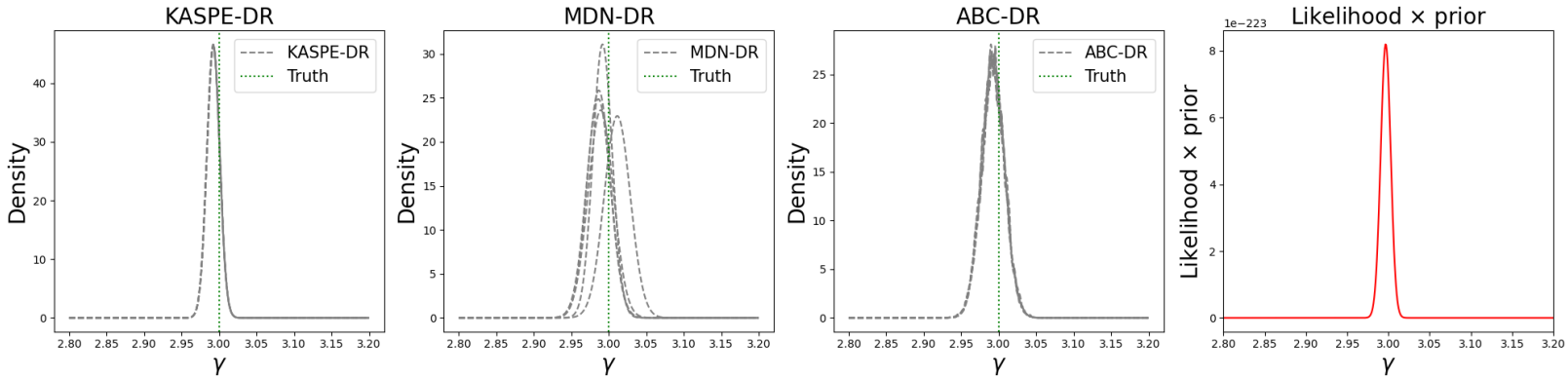
- $v(0) = -1$, $r(0) = 1$; $\theta = (\theta_1, \theta_2)^\top$ fixed, γ unknown
- Voltage observed with additive noise via $y(t) = v(t) + \epsilon$ at $t_i = 0.025i$ for $i = 1, \ldots, 1000$, and $\epsilon \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.06^2)$
- Summaries are coefficients of a nonlinear regression on K Fourier basis functions

FitzHugh–Nagumo Model

Uniform prior, $\gamma \sim \text{unif}(0, 15)$



$m = 6$, full data



$m = 1000$, dimension reduction

References

All references cited in the slides are listed here: