Detecting Model Misspecification in Amortized Bayesian Inference with Neural Networks

Marvin Schmitt[1], Paul-Christian Bürkner[1,2], Ullrich Köthe[3], & Stefan T. Radev[3]

German Conference on Pattern Recognition 2023

[1] University of Stuttgart
[2] TU Dortmund University
[3] Heidelberg University

For Jovita

Do you have a moment to talk about our

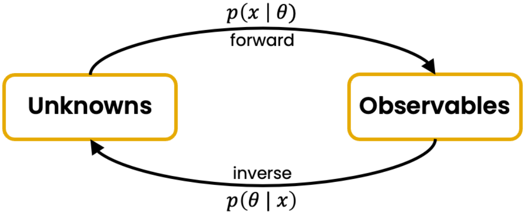
lord and savior

Do you have a moment to talk about our

lord and savior

Sir Thomas Bayes?

Inverse problems



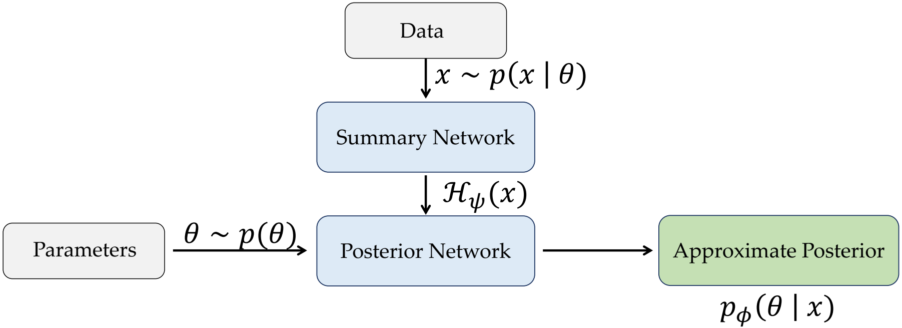
Statistical modeling:	**Parameters θ**	**Data x**
Epidemiology:	Virus attributes	Infection curve (time series)
Image processing:	Crisp image	Blurry image
Physics:	Physical attributes	Graviational wave measurements

Bayesian inference

$$\overbrace{p(\boldsymbol{\theta} \mid \boldsymbol{x})}^{\text{inverse}} = \frac{\overbrace{p(\boldsymbol{x} \mid \boldsymbol{\theta})}^{\text{forward}} \, p(\boldsymbol{\theta})}{\underbrace{\int p(\boldsymbol{x}, \boldsymbol{\theta}) \mathrm{d}\theta}_{= p(\boldsymbol{x})}}$$

Figure 1: Maybe Thomas Bayes

Neural posterior estimation (NPE)



NPE: Optimization objective

The analytic posterior $p(\boldsymbol{\theta} \,|\, \boldsymbol{x})$ and the approximated posterior $p_{\boldsymbol{\phi}}(\boldsymbol{\theta} \,|\, \mathcal{H}_{\boldsymbol{\psi}}(\boldsymbol{x}))$ on learned summary statistics $\mathcal{H}_{\boldsymbol{\psi}}(\boldsymbol{x})$ shall match:

$$
\begin{aligned}
(\boldsymbol{\phi}^*, \boldsymbol{\psi}^*) &= \underset{\boldsymbol{\phi},\boldsymbol{\psi}}{\operatorname{argmin}}\, \mathbb{E}_{p^*(\boldsymbol{x})}\Big[\mathbb{KL}\Big(p\left(\boldsymbol{\theta} \,|\, \boldsymbol{x}\right) \Big|\Big| p_{\boldsymbol{\phi}}\left(\boldsymbol{\theta} \,|\, \mathcal{H}_{\boldsymbol{\psi}}\left(\boldsymbol{x}\right)\right)\Big)\Big] \\
&= \underset{\boldsymbol{\phi},\boldsymbol{\psi}}{\operatorname{argmin}}\, \mathbb{E}_{p^*(\boldsymbol{x})}\Big[\mathbb{E}_{p(\boldsymbol{\theta} \,|\, \boldsymbol{x})}\big[-\log p_{\boldsymbol{\phi}}(\boldsymbol{\theta} \,|\, \mathcal{H}_{\boldsymbol{\psi}}(\boldsymbol{x}))\big]\Big]
\end{aligned}
$$

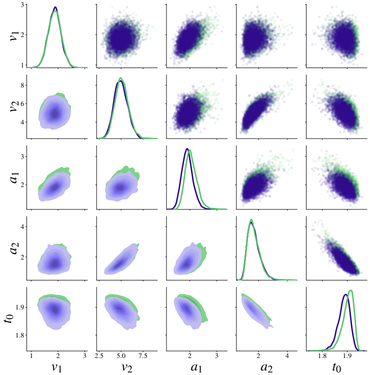
Out-of-distribution in NPE

Assume that the true data distribution $p^*(\boldsymbol{x})$ equals the simulated $p(\boldsymbol{x})$:

$$(\boldsymbol{\phi}^*, \boldsymbol{\psi}^*) = \underset{\boldsymbol{\phi},\boldsymbol{\psi}}{\operatorname{argmin}} \, \mathbb{E}_{p(\boldsymbol{x})} \left[\mathbb{E}_{p(\boldsymbol{\theta} \,|\, \boldsymbol{x})} \Big[-\log p_{\boldsymbol{\phi}}(\boldsymbol{\theta} \,|\, \mathcal{H}_{\boldsymbol{\psi}}(\boldsymbol{x})) \Big] \right]$$
$$= \underset{\boldsymbol{\phi},\boldsymbol{\psi}}{\operatorname{argmin}} \, \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})} \Big[-\log p_{\boldsymbol{\phi}}(\boldsymbol{\theta} \,|\, \mathcal{H}_{\boldsymbol{\psi}}(\boldsymbol{x})) \Big]$$

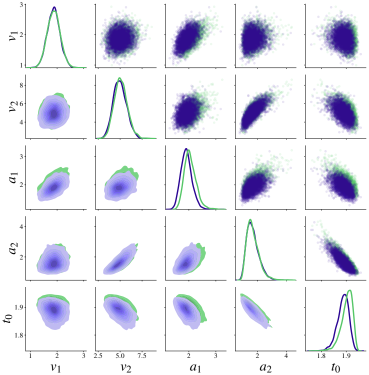
If $p^*(\boldsymbol{x}) \neq p(\boldsymbol{x})$, we optimize with respect to the wrong distribution.

What happens when MCMC and NPE encounter OOD data?

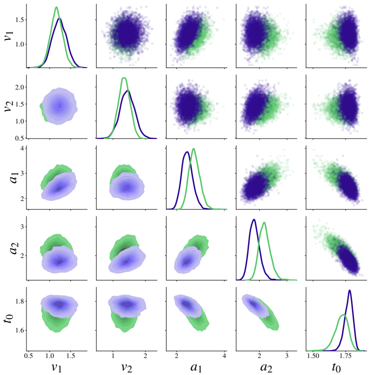


(a) well-specified: MCMC $\approx$ NPE

What happens when MCMC and NPE encounter OOD data?

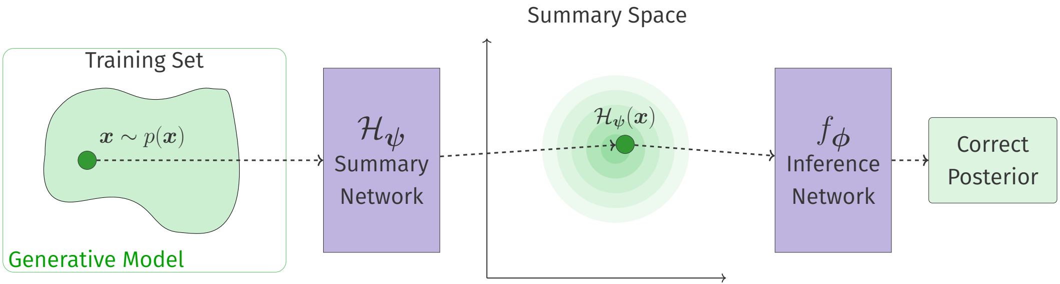


(a) well-specified: MCMC $\approx$ NPE



(b) misspecified: MCMC $\neq$ NPE

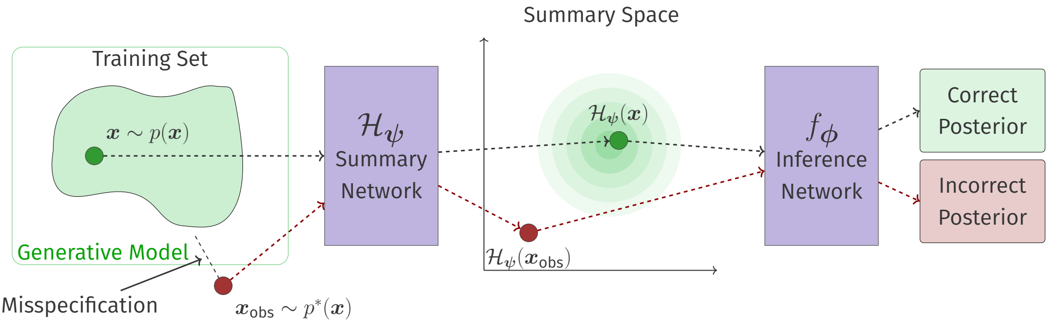
Structured summary statistics



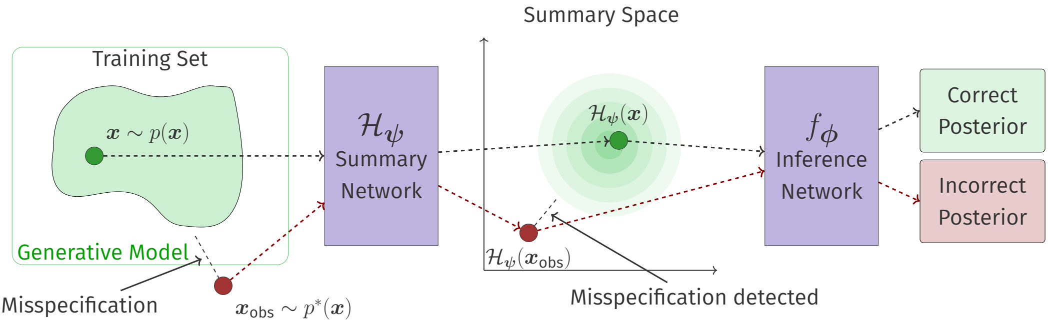
Optimize the summary network's output $\mathcal{H}_\psi(x)$ towards a unit Gaussian:

$$p\big(\mathcal{H}_\psi(x)\big) \approx \mathcal{N}(z \mid 0, \mathbb{I})$$

Detecting out-of-distribution data



Detecting out-of-distribution data



Detect via MMD between simulated $\mathcal{H}_\psi(\boldsymbol{x})$ and observed $\mathcal{H}_\psi(\boldsymbol{x}_{\text{obs}})$

Experiment 1: Gaussian toy model

Gaussian: Setup

Recover mean vector $\boldsymbol{\mu}$ of a $2-$dimensional spherical Gaussian:

$$\begin{aligned} \boldsymbol{\mu} &\sim \mathcal{N}(\boldsymbol{\mu} \,|\, \boldsymbol{\mu}_0, \tau_0 \mathbb{I}) \\ \boldsymbol{x}_k &\sim \mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \tau \mathbb{I}) \quad \text{for } k = 1, \ldots, K. \end{aligned} \tag{1}$$

Potential misspecifications:

- Prior location $\boldsymbol{\mu}_0$ and scale τ_0
- Likelihood scale τ
- Unmodeled noise

Gaussian: Perfect performance for well-specified model

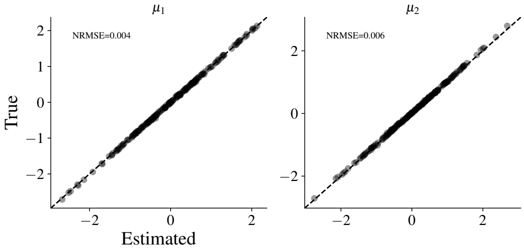


Figure 3: Well-specified case

Gaussian: Perfect performance for well-specified model

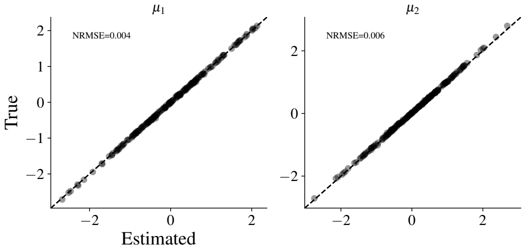


Figure 3: Well-specified case

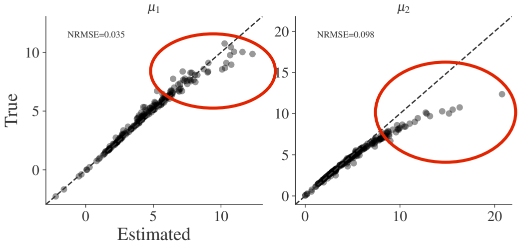
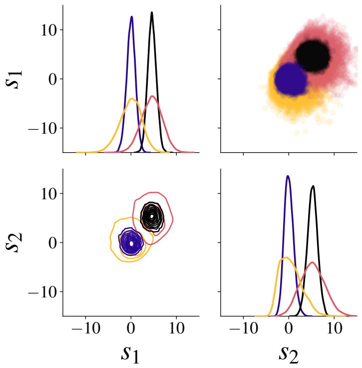
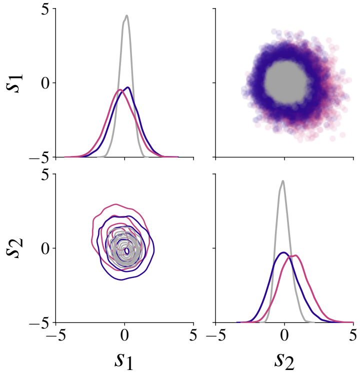


Figure 4: Prior misspecification: $\mu_0 = 2.5$

Gaussian: Inspecting the summary space





Prior location: $\mu_0 = 5$

Prior scale: $\tau_0 = 2.5$

Prior location and scale: $\mu_0 = 5, \tau_0 = 2.5$

No MMS

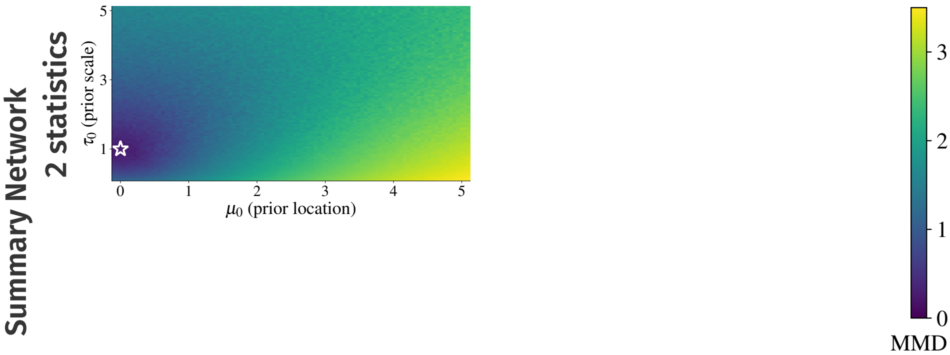
Simulator scale: $\tau = 10$

Noise fraction: $\lambda = 0.5$

Gaussian: How many summary statistics?

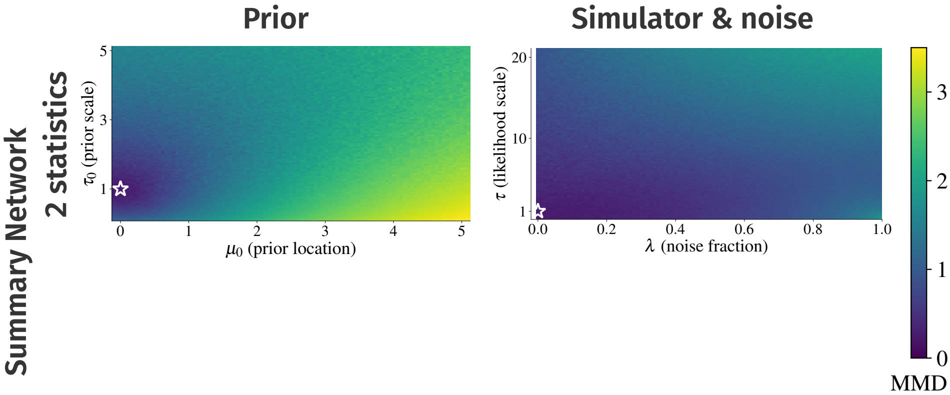
Model Misspecification

Prior



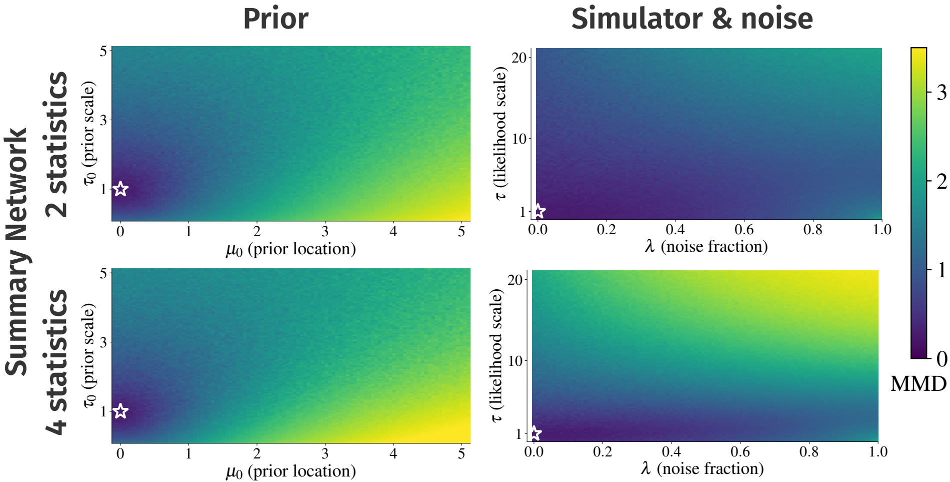
Gaussian: How many summary statistics?

Model Misspecification



Gaussian: How many summary statistics?

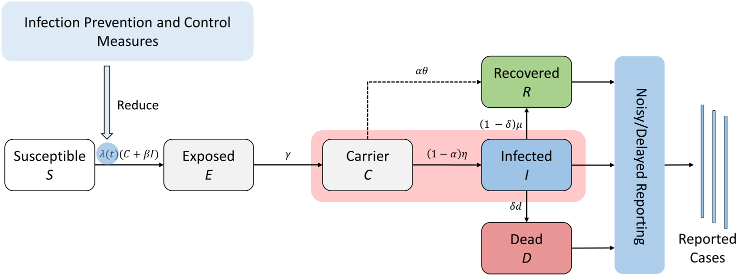
Model Misspecification



Experiment 3: COVID-19 modeling

Compartmental Models for disease outbreaks (Radev et al., 2021)

1. Inference is based on posteriors $\rightarrow$ must be trustworthy
2. Are initially well-specified models misspecified at some point?

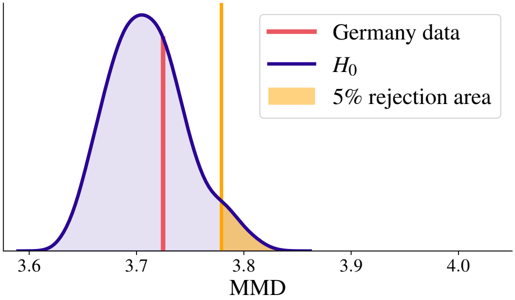


COVID-19: Setup

- Train the network on data from the full model $\mathcal{M}^*$
- Simulate 1000 time series each from
  - $\mathcal{M}^*$: full model
  - $\mathcal{M}_1$: no intervention sub-model
  - $\mathcal{M}_2$: no observation sub-model
- Find discrepancies in the latent summary space

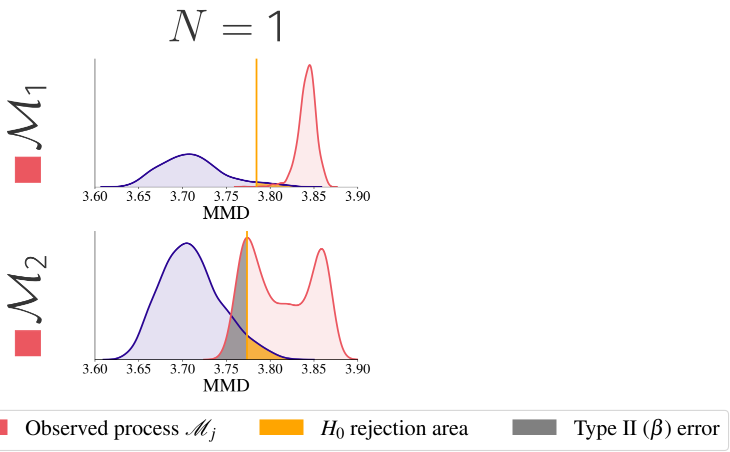
COVID-19: Is the model well-specified for German data?

Frequentist hypothesis test: $H_0 : p^*(\boldsymbol{x}) = p(\boldsymbol{x}) \quad H_1 : p^*(\boldsymbol{x}) \neq p(\boldsymbol{x})$

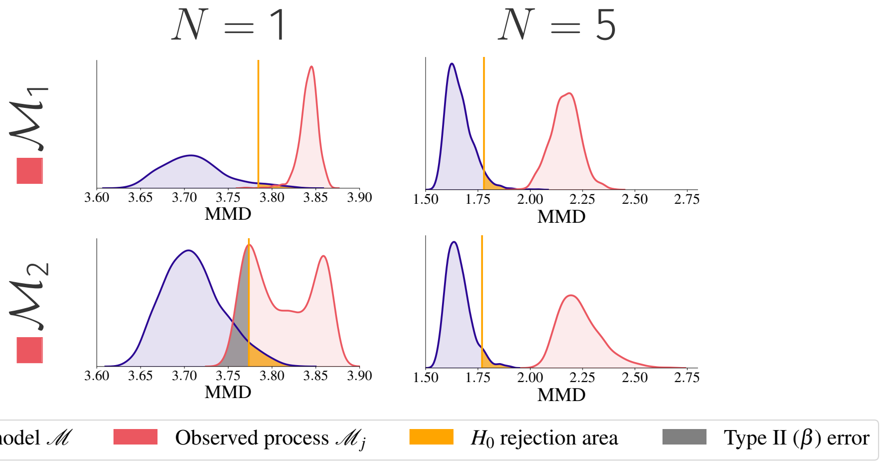


Conclusion: Don't reject the null hypothesis → model is well-specified.

Power of a frequentist hypothesis test on summary space MMD



Power of a frequentist hypothesis test on summary space MMD



Summary

- Neural posterior estimation hinges on a valid simulator

Summary

- Neural posterior estimation hinges on a valid simulator
- OOD detection is difficult in data space (entire ML field)

Summary

- Neural posterior estimation hinges on a valid simulator
- OOD detection is difficult in data space (entire ML field)
- OOD detection is easier in a structured summary space
- Imposing structure is easy in NPE with learned summaries

Summary

- Neural posterior estimation hinges on a valid simulator
- OOD detection is difficult in data space (entire ML field)
- OOD detection is easier in a structured summary space
- Imposing structure is easy in NPE with learned summaries
- Unhappy with MMD or the frequentist hypothesis test?
    - Bring your own distance metric
    - Bring your own test

Summary

- Neural posterior estimation hinges on a valid simulator
- OOD detection is difficult in data space (entire ML field)
- OOD detection is easier in a structured summary space
- Imposing structure is easy in NPE with learned summaries
- Unhappy with MMD or the frequentist hypothesis test?
  - Bring your own distance metric
  - Bring your own test
- All implementations in the `BayesFlow` library: bayesflow.org

Contact

Marvin Schmitt
@MarvinSchmittML

Paul Bürkner
@paulbuerkner

Ullrich Köthe
Ullrich Köthe

Stefan Radev
@stefanradev13

Partially funded by

Cyber Valley

SimTech Cluster of Excellence

Informatics for Life

STRUCTURES CLUSTER OF EXCELLENCE