Learning summary statistics for Bayesian inference with Autoencoders

Carlo Albert

OWABC, September 28 2023

Problem Statement

Likelihood functions involving high-dimensional integration/summation over latent variables, because there is either

- a large number of unobserved latent variables:

$$f\left(\mathbf{y}^{(\mathrm{obs})} \mid \boldsymbol{\theta}\right) = \int f\left(\mathbf{y}^{(\mathrm{obs})} \mid \mathbf{x}, \boldsymbol{\theta}\right) f\left(\mathbf{x} \mid \boldsymbol{\theta}\right) d\mathbf{x}$$

- or an unknown partition function (e.g. Ising-type model):

$$f\left(\mathbf{y}^{(\mathrm{obs})} \mid \boldsymbol{\theta}\right) = Z^{-1}(\boldsymbol{\theta}) \exp[-H_{\boldsymbol{\theta}}(\mathbf{y}^{(\mathrm{obs})})]$$

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{y}} \exp[-H_{\boldsymbol{\theta}}(\mathbf{y})]$$

Approximate Bayesian Computation (ABC)

- Simulate model outputs $\mathbf{y}$, for many parameter sets $\boldsymbol{\theta}$, from $f(\mathbf{y} \mid \boldsymbol{\theta})$.
- Accept/reject the corresponding parameter sets depending on whether those outputs agree with the observations in terms of
  - Certain **features** (**summary statistics**) $\mathbf{s}(\mathbf{y})$
  - A certain **tolerance**
- E.g. Ising model

$$f\left(\mathbf{y} \mid \beta, h\right) = Z^{-1}(\beta, h)\exp\left[\beta\left(\sum_{<i,j>} y_i y_j + h\sum_i y_i\right)\right]$$

$$= Z^{-1}(\beta, h)\exp\left[\beta E(\mathbf{y}) + \beta h M(\mathbf{y})\right]$$

$$\mathbf{s}(\mathbf{y}) := (E(\mathbf{y}), M(\mathbf{y}))^T$$

sufficient statistics!

Tuning the tolerance: Simulated Annealing ABC (SABC)

Albert et al., Stat. Comput. 2014

- Use metric $\rho(\mathbf{s}, \mathbf{s}^{(obs)})$ -> interpret as **energy**.
- Simulate from $f\left(\mathbf{s}, \boldsymbol{\theta}\right) e^{-\rho(\mathbf{s},\mathbf{s}^{(obs)})/T}$ while continuously lowering **temperature** T.
- For $T \to 0$, $f\left(\mathbf{s}, \boldsymbol{\theta}\right) e^{-\rho(\mathbf{s},\mathbf{s}^{(obs)})/T} \to f\left(\boldsymbol{\theta} \;\; \mathbf{s}^{(obs)}\right)$.

Initialisation: sample an ensemble of particles $\{\mathbf{s}_j, \boldsymbol{\theta}_j\}_{j=1}^{N}$ from the prior $f\left(\mathbf{s}, \boldsymbol{\theta}\right)$

1. Draw a random particle $(\mathbf{s}_j, \boldsymbol{\theta}_j)$ from the ensemble
2. Make a jump in parameter space $\boldsymbol{\theta}_j \to \boldsymbol{\theta}_j^*$
3. Simulate a data set $\mathbf{y}_j^*$ from the model $f\left(\mathbf{y} \mid \boldsymbol{\theta}\right)$ and calculate $\mathbf{s}_j^* = \mathbf{s}(\mathbf{y}_j^*)$
4. Accept the move with probability $\min\left(1, \frac{f(\boldsymbol{\theta}_j^*)}{f(\boldsymbol{\theta}_j)} \exp\left[-\frac{\rho(\mathbf{s}_j^*, \mathbf{s}^{(\mathrm{obs})}) - \rho(\mathbf{s}_j, \mathbf{s}^{(\mathrm{obs})})}{T}\right]\right)$
5. Lower the temperature T adaptively so as to minimise entropy production.

Data compression - fighting the curse of dimensionality

- The computational price to pay for decreasing the tolerance grows exponentially with the output dimension.
- Therefore, we must compress the data to a handful of summary statistics.
- Generally, this leads to a loss of information relevant to constrain the parameters.

Summary Statistics

Albert et al., SciPost Physics 2022

Consider a generic stochastic model:

$$f(\mathbf{y}, \boldsymbol{\theta}) = f(\mathbf{y} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta}) \quad \boldsymbol{\theta} \in \mathbb{R}^p, \quad \mathbf{y} \in \mathbb{R}^N,$$

and a map of summary statistics

$$\mathbf{s} : \mathbb{R}^N \longrightarrow \mathbb{R}^q .$$

We want them to be

asymptotically sufficient: $I(\mathbf{s}(\mathbf{Y}), \boldsymbol{\Theta}) = I(\mathbf{Y}, \boldsymbol{\Theta}) + \mathcal{O}(N^{-1})$

(Accuracy)

Asymptotically concentrated: $H(\mathbf{s}(\mathbf{Y}) \mid \boldsymbol{\Theta}) \sim -\ln N$

(Efficiency)

-> Thermodynamic state variables

Summary Statistics

Albert et al., SciPost Physics 2022

- If sufficient statistics cannot easily be found, a reasonable choice is to use parameter estimators, e.g.

$$\mathbf{s}(\mathbf{y}) = \hat{\boldsymbol{\theta}}(\mathbf{y}) = \int \boldsymbol{\theta} f(\boldsymbol{\theta} \mid \mathbf{y}) d\boldsymbol{\theta}$$

- Fearnhead and Prangle (2012) use a linear regression (built-into SABC software packages)
- Machine Learning lends itself to this task as well! (e.g. Jiang 2017)

Summary Statistics

Albert et al., SciPost Physics 2022

- For large *i.i.d.* datasets, parameter estimators capture most of the $\boldsymbol{\theta}$-related information in the data.
- If the data is *correlated*, this is no longer the case and we might need more summary statistics than parameters
- This is due to entropic effects: For members of the exponential family

$$f(\mathbf{y} \;\; \boldsymbol{\theta}) = Z^{-1}(\boldsymbol{\theta}) c(\mathbf{y}) \exp\left(\sum_{\alpha=1}^{q} s_\alpha(\mathbf{y}) g^\alpha(\boldsymbol{\theta}) \right)$$

the free energy splits into en energy and an entropy term

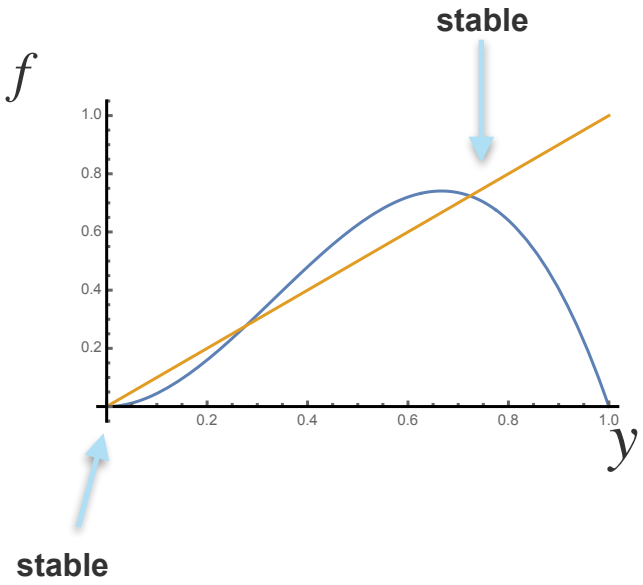
$$F_{\boldsymbol{\theta}}(\mathbf{s}) := -\ln f(\mathbf{s} \;\; \boldsymbol{\theta}) = -\ln \int f(\mathbf{y} \;\; \boldsymbol{\theta}) d\Omega_{\mathbf{s}}(\mathbf{y}) = U_{\boldsymbol{\theta}}(\mathbf{s}) - S(\mathbf{s})$$

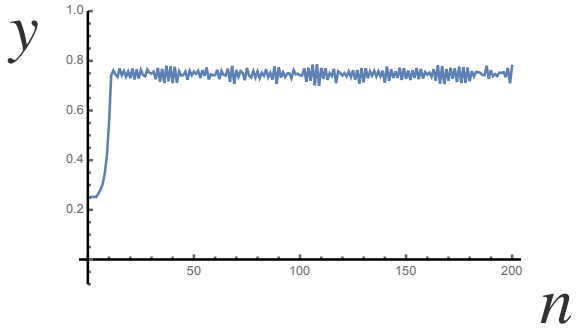
and is not necessarily concentrated around a single point, for fixed $\boldsymbol{\theta}$, even if N is large; possibly even when $N \to \infty$ (*phases!)*.

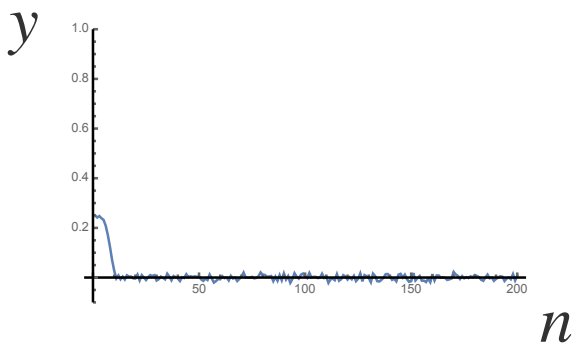
Summary Statistics

Albert et al., SciPost Physics 2022

Toy: $y_{n+1} = \alpha f(y_n) + \sigma\epsilon_n\,, \quad \epsilon_n \sim \mathcal{N}(0,1) \quad \text{i.i.d.}\,, \quad f(y) = y^2(y-1)$







Summary Statistics

Albert et al., SciPost Physics 2022

Toy: $y_{n+1} = \alpha f(y_n) + \sigma\epsilon_n\,, \quad \epsilon_n \sim \mathcal{N}(0,1) \quad \text{i.i.d.}\,, \quad f(y) = y^2(y-1)$

$$f(\mathbf{y}\ \boldsymbol{\theta}) \propto \sigma^{-N} \exp\left[-\sum_n \frac{(y_{n+1} - \alpha f(y_n))^2}{2\sigma^2}\right], \quad \boldsymbol{\theta} = \begin{pmatrix}\alpha \\ \sigma\end{pmatrix}$$

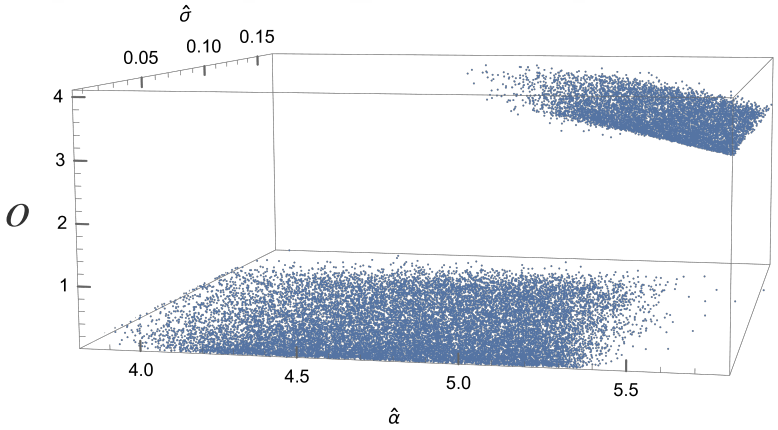
Sufficient statistics:

$$\left.\begin{aligned} \hat{\alpha}(\mathbf{y}) &= \frac{\sum_{n=1}^{N} y_n f(y_{n-1})}{\sum_{n=1}^{N} (f(y_{n-1}))^2}\,, \\ \hat{\sigma}^2(\mathbf{y}) &= \frac{1}{N}\sum_{n=1}^{N} (y_n - \hat{\alpha}(\mathbf{y}) f(y_{n-1}))^2\,, \end{aligned}\right\}$$ **Parameter estimators**

$$o(\mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N} (f(y_{n-1}))^2\,.$$ **Order parameter**

Summary Statistics

Albert et al., SciPost Physics 2022



Sufficient statistics:

$$\hat{\alpha}(\mathbf{y}) = \frac{\sum_{n=1}^{N} y_n f(y_{n-1})}{\sum_{n=1}^{N} (f(y_{n-1}))^2},$$

$$\hat{\sigma}(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{\alpha}(\mathbf{y}) f(y_{n-1}))^2,$$

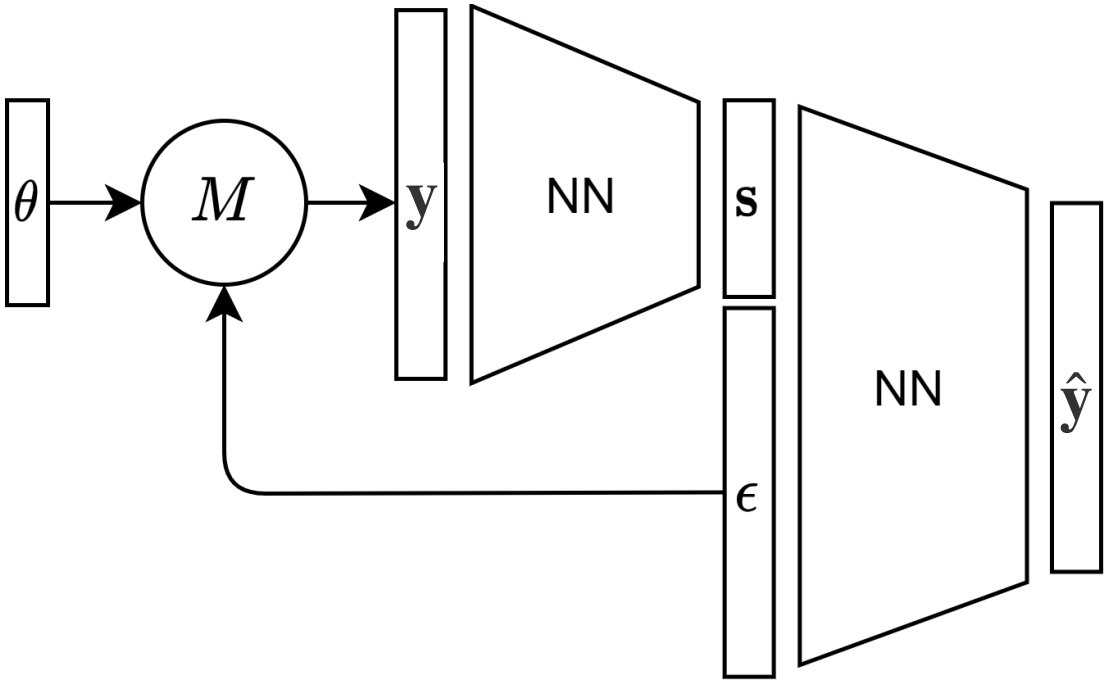
} **Parameter estimators**

$$o(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} (f(y_{n-1}))^2.$$

Order parameter

Summary Statistics

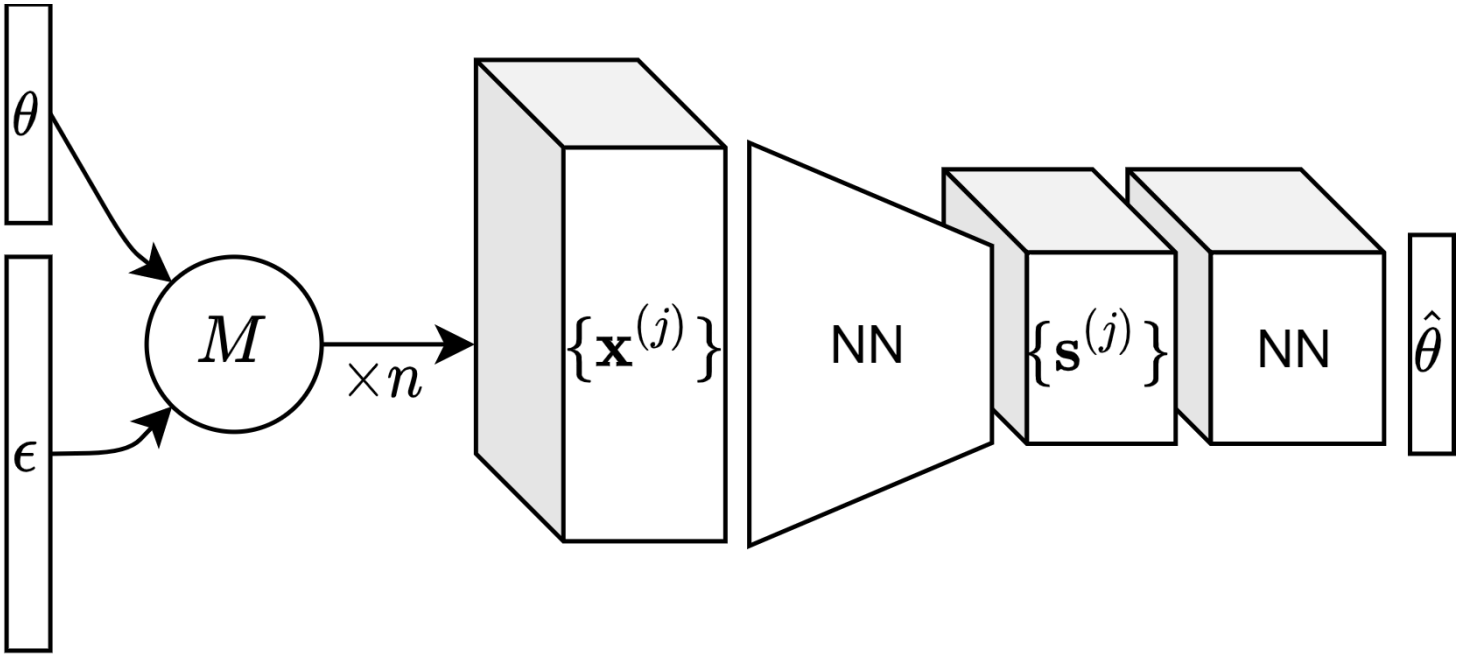
Albert et al., SciPost Physics 2022



$$\mathscr{L} = \frac{1}{p}\sum_{\alpha=1}^{p}\left(\frac{s_\alpha - \theta_\alpha}{\theta_\alpha}\right)^2 + \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2 .$$

Explicit Noise Conditional Autoencoder
ENCA

Summary Statistics



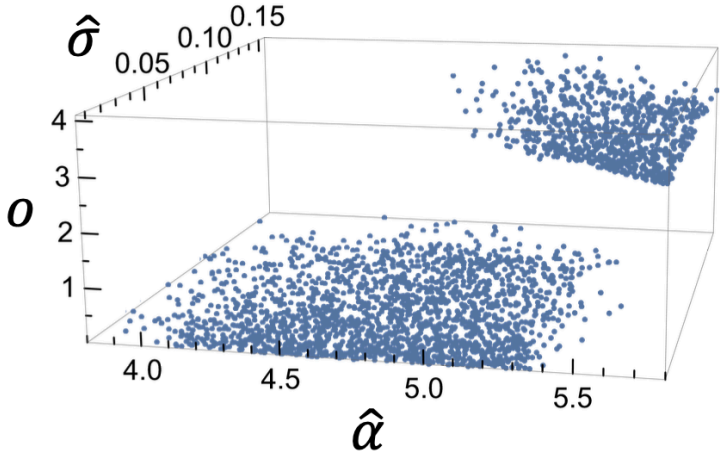
$$\mathscr{L} = \sum_{j,\alpha} \left(\frac{s_\alpha^{(j)} - \theta_\alpha}{\theta_\alpha} \right)^2 + \sum_{\alpha} \left(\frac{\hat{\theta}_\alpha - \theta_\alpha}{\theta_\alpha} \right)^2$$

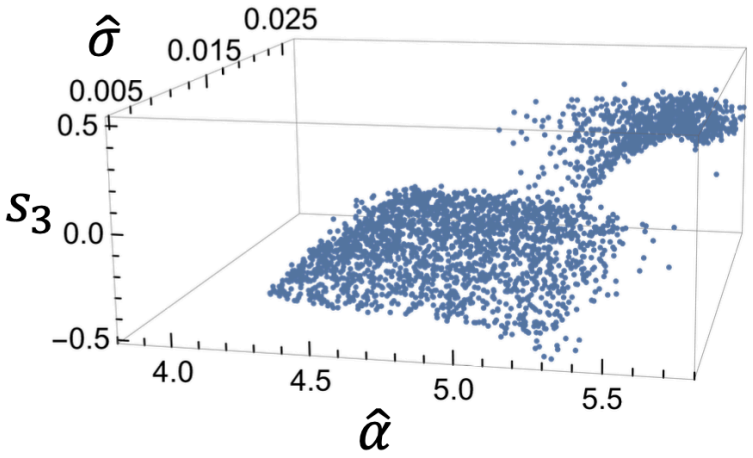
Explicit Noise Conditional Autoencoder
ENCA

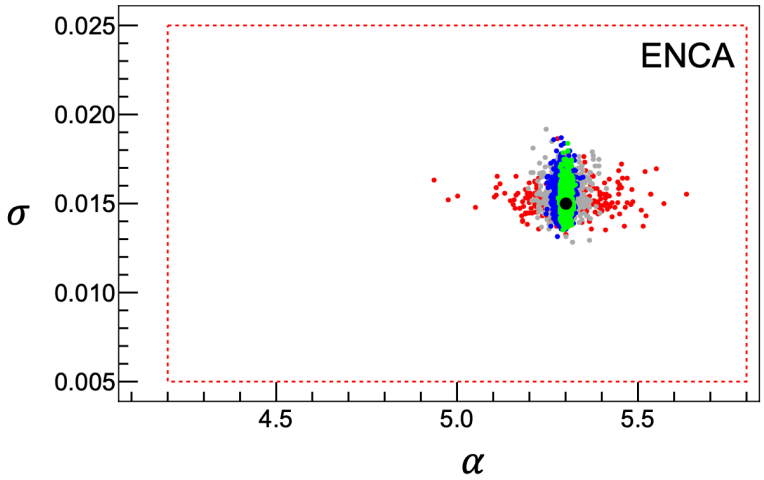
Summary Statistics

Albert et al., SciPost Physics 2022

Toy: $y_{n+1} = \alpha f(y_n) + \sigma\epsilon_n\,, \quad \epsilon_n \sim \mathcal{N}(0,1) \quad \text{i.i.d.}\,, \quad f(y) = y^2(y-1)$





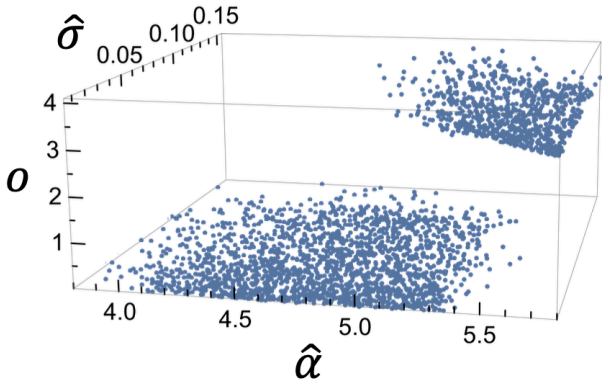


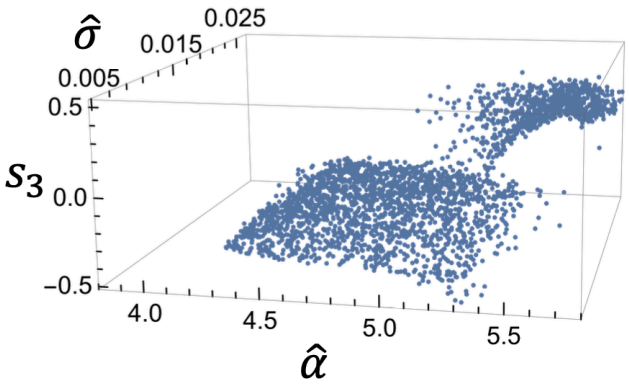
- True posterior
- Only MLEs
- 3 Stats form ENCA
- 2 Stats from ENCA

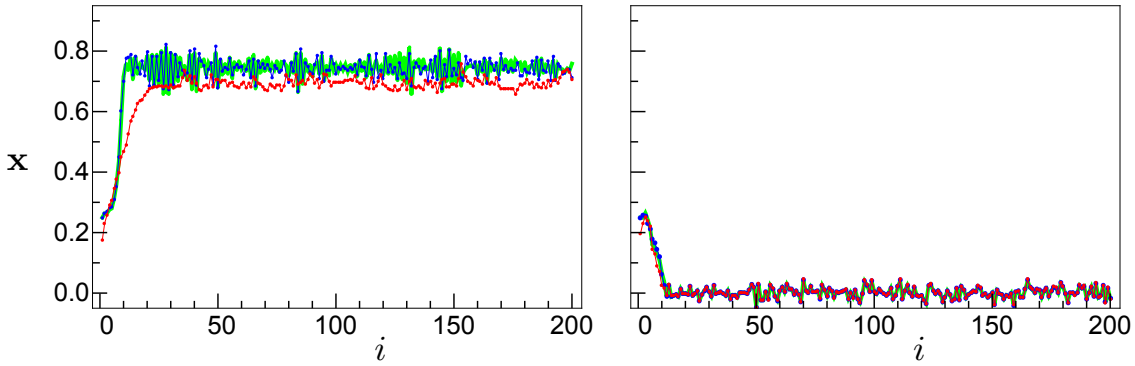
Summary Statistics

Albert et al., SciPost Physics 2022

Toy: $y_{n+1} = \alpha f(y_n) + \sigma\epsilon_n\,, \quad \epsilon_n \sim \mathcal{N}(0,1) \quad \text{i.i.d.}\,, \quad f(y) = y^2(y-1)$

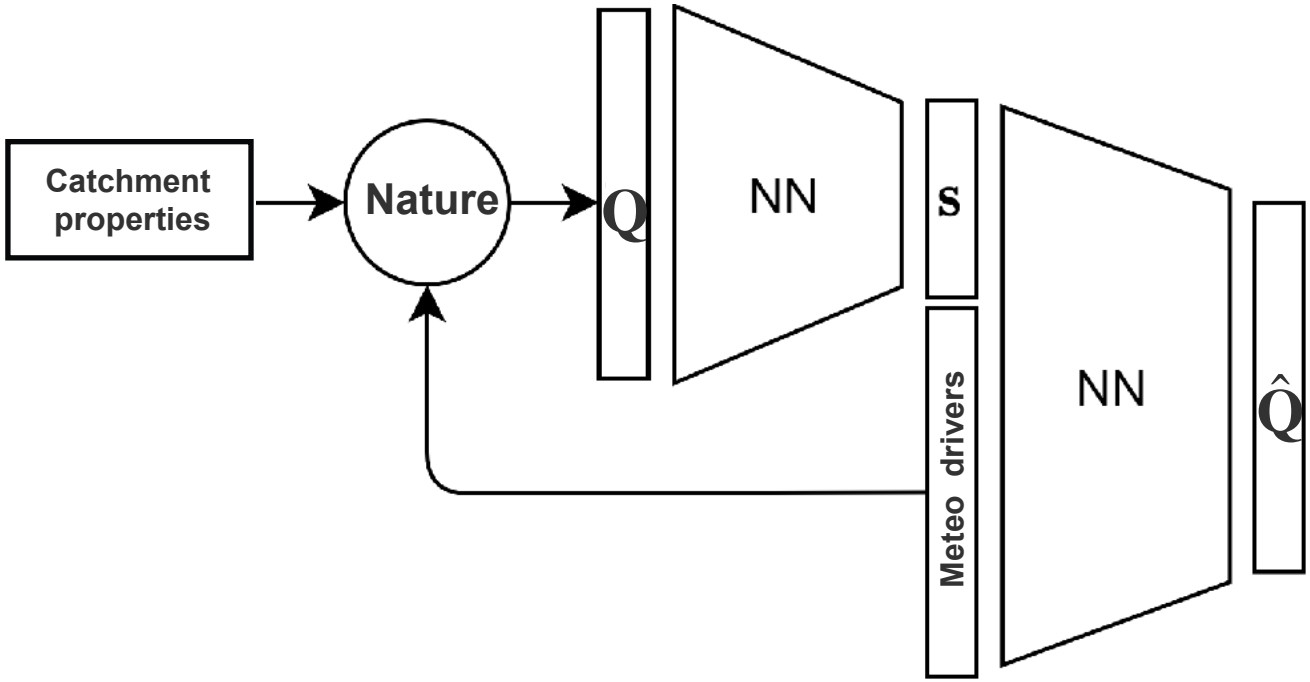






- M-realizations
- 2 Stats
- 3 Stats

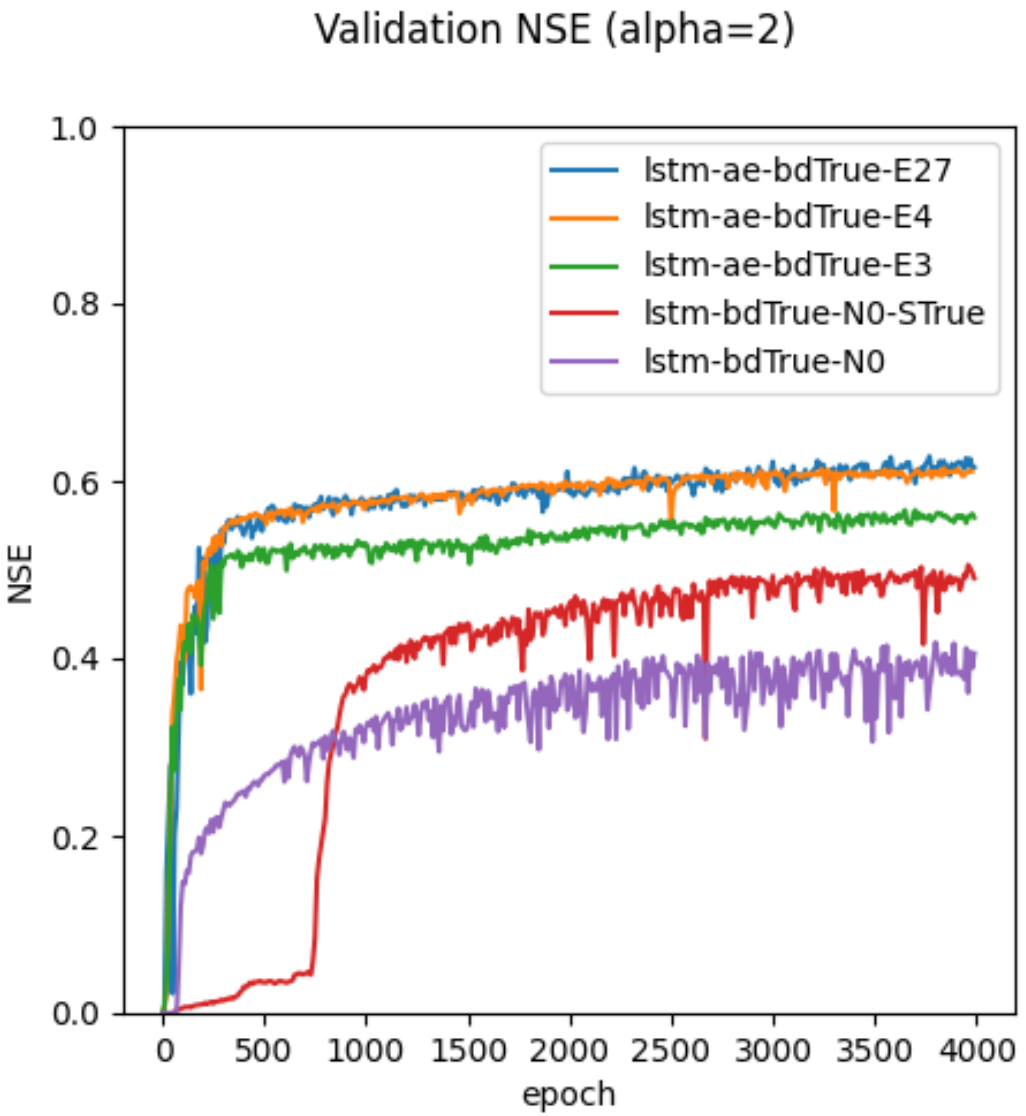
Finding low-dimensional features in streamflow data



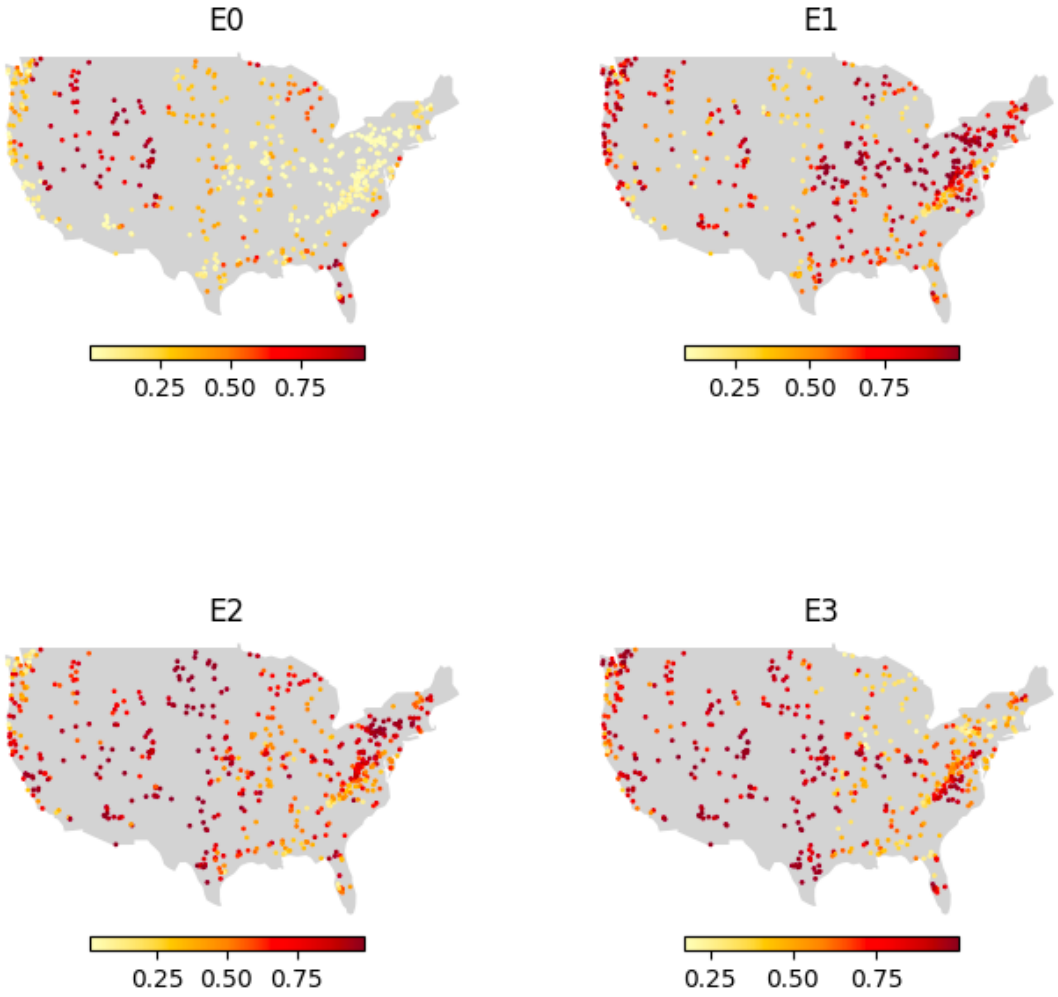
$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{Q}_i - Q_i}{Q_i}\right)^2 .$$

Finding low-dimensional features in streamflow data

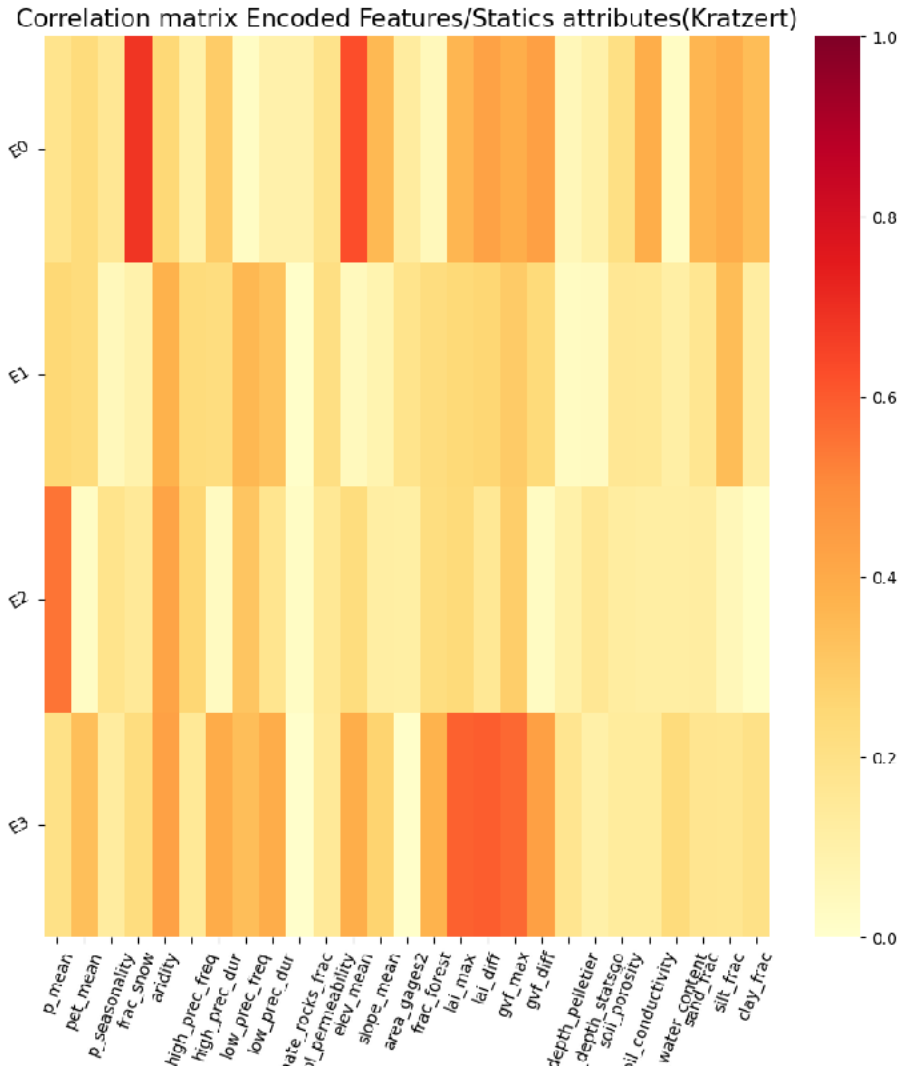
- **4 features** seem to be sufficient to predict streamflow.
- There seems to be **information missing** in the known static catchment attributes.



Finding low-dimensional features in streamflow data



Finding low-dimensional features in streamflow data



Conclusions

- The **ENCA architecture** can be used to disentangle high-dimensional irrelevant from low-dimensional relevant information.
- It can be used to find near-sufficient and concentrated summary statistics in **simulated** data from stochastic models.
- It can be used to find catchment features in **observed** streamflow data.

C.A. al., SciPost Physics 2022

Thanks to my collaborators: Simone Ulzega, Antonietta Mira, Firat Oezdemir, Fernando Perez-Cruz, Alberto Bassi.