University of Stuttgart
Cluster of Excellence in Data-integrated Simulation Science

Toward an Information-Theoretic Diagnostic Evaluation Framework for Hybrid Models

Manuel Álvarez Chaves, Uwe Ehret, Hoshin Gupta and Anneli Guthke

Non-parametric estimation in Information Theory

Most approaches typically involve an initial step of density estimation before computing the desired quantities from information theory. Density estimation remains a challenge specially in higher dimensions. k-NN based methods skip this initial step.

$$
\begin{aligned}
H(X) &= -\int_{\mathcal{X}} p(x)\log p(x)\,\mathrm{d}x \\
&= -\sum \Delta \hat{f}(x_i)\log \hat{f}(x_i) - \underbrace{\sum \hat{f}(x_i)\Delta \log \Delta}_{}
\end{aligned}
$$

correction factor due to binning

What is your Δ ?

$$
\begin{aligned}
D_{KL}(p\,||\,q) &= \int_{\mathcal{X}} p(x)\log\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x \\
&= \frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{\hat{p}(x_i)}{\hat{q}(x_i)}\right)
\end{aligned}
$$

Kernel representation of p(x)

What is your Kernel ?

$$
\begin{aligned}
I(X;Y) &= \int_{\mathcal{Y}}\int_{\mathcal{X}} p(x,y)\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)\mathrm{d}x\,\mathrm{d}y \\
&= \psi(k) - \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\psi(n_{i,x}+1)+\psi(n_{i,y}+1)\right]+\psi(N)
\end{aligned}
$$

number of neighbours

What is your k ?

Gamma-Exponential dist.

$$p(x_1, x_2) = \frac{x_1^{\theta} e^{-x_1 - x_1 \cdot x_2}}{\Gamma(\theta)}$$

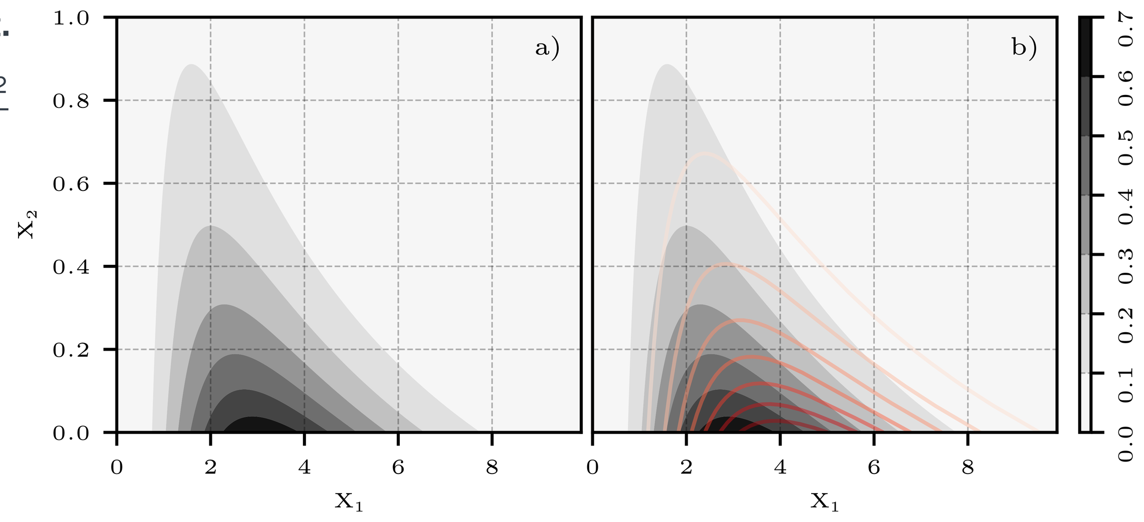


Figure 1: a) Density of the Gamma-Exponential distribution with $\theta = 3$ **b)** Same as a) with an approximating Gamma-Exponential distribution with $\theta = 4$ in red.

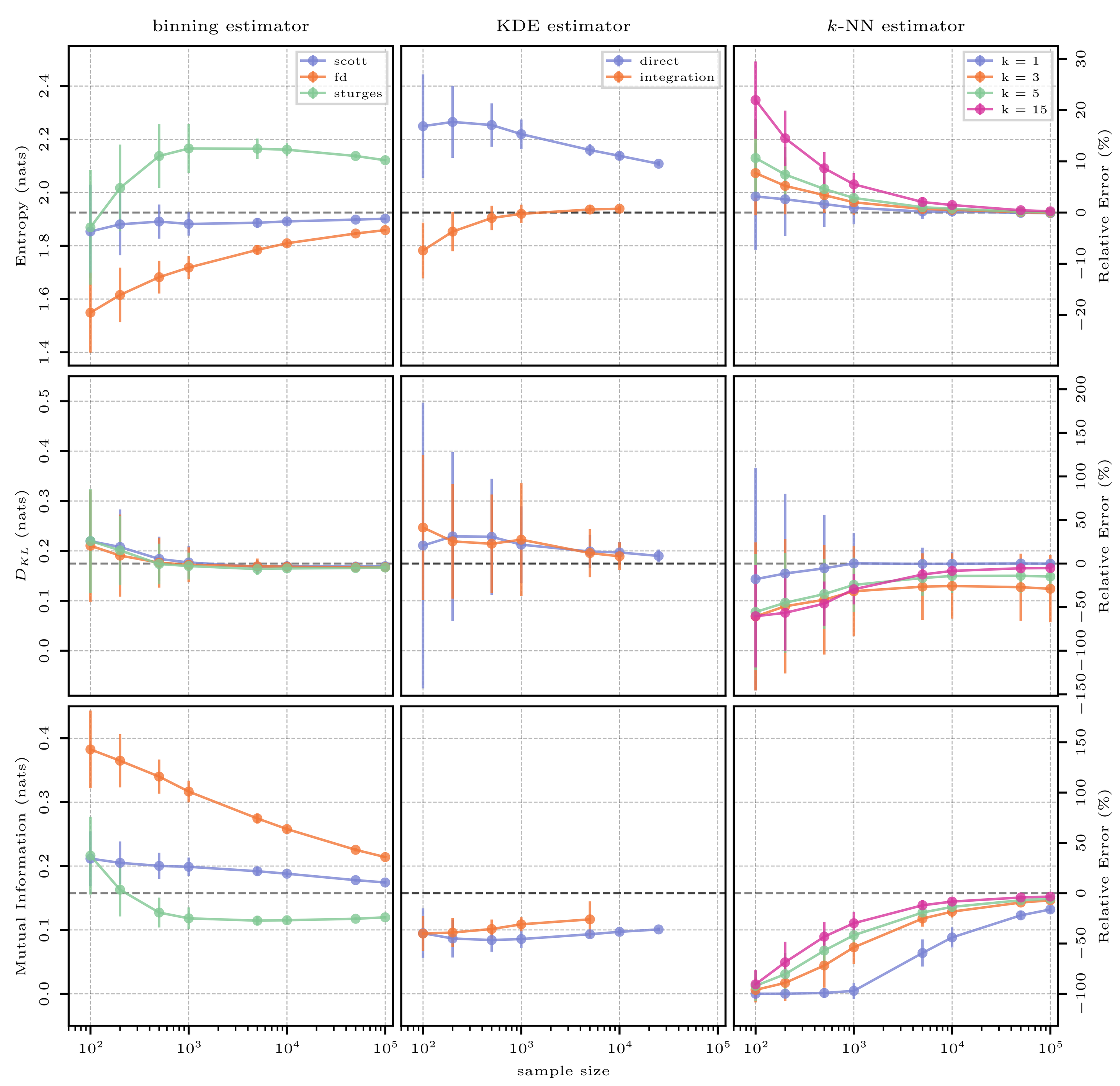
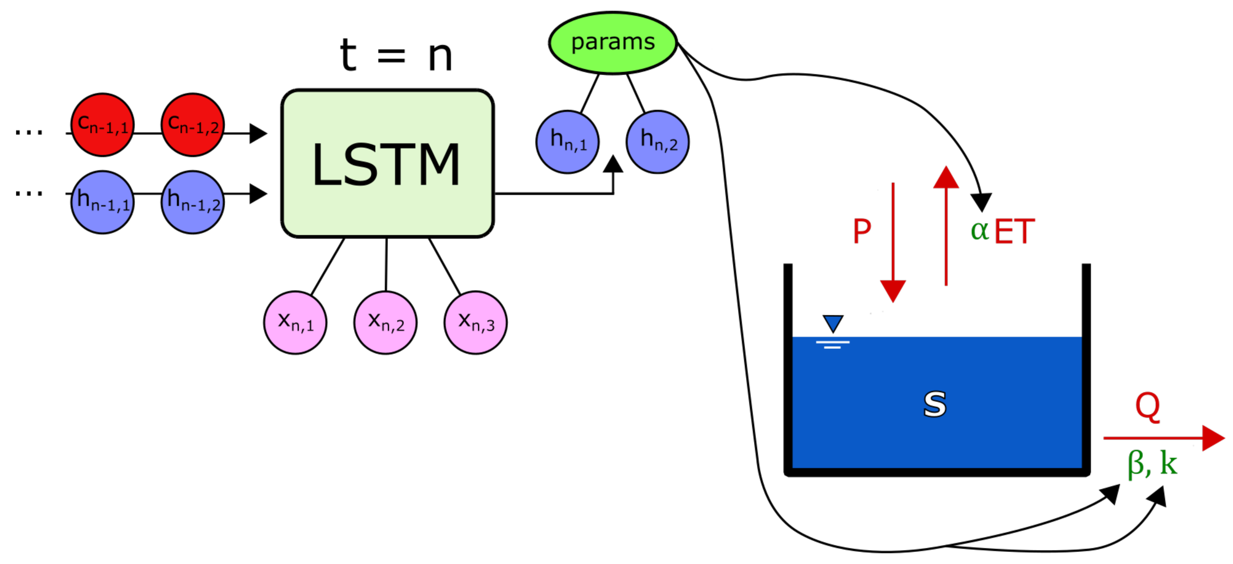


Figure 2: Evaluation of all estimators in the test case of the Gamma-Exponential distribution

```
In [2]: from scipy import stats
        from unite_toolbox import knn_estimators

        dist = stats.norm(loc=0, scale=0.6577)
        samples = dist.rvs(size=(10_000, 1))

        est_h = knn_estimators.calc_knn_entropy(samples)

        print(f"True entropy: {dist.entropy():.3f} nats")
        print(f"Est. entropy: {est_h:.3f} nats")

        True entropy: 1.000 nats
        Est. entropy: 1.008 nats
```

Applications

Hybrid Models



LSTMs for Model Diagnostics

→ Through the power of **Information Theory**!

We can understand how much an LSTM has to overcompensate for a poorly specified model by measuring the entropy of the predictions for the parameters.

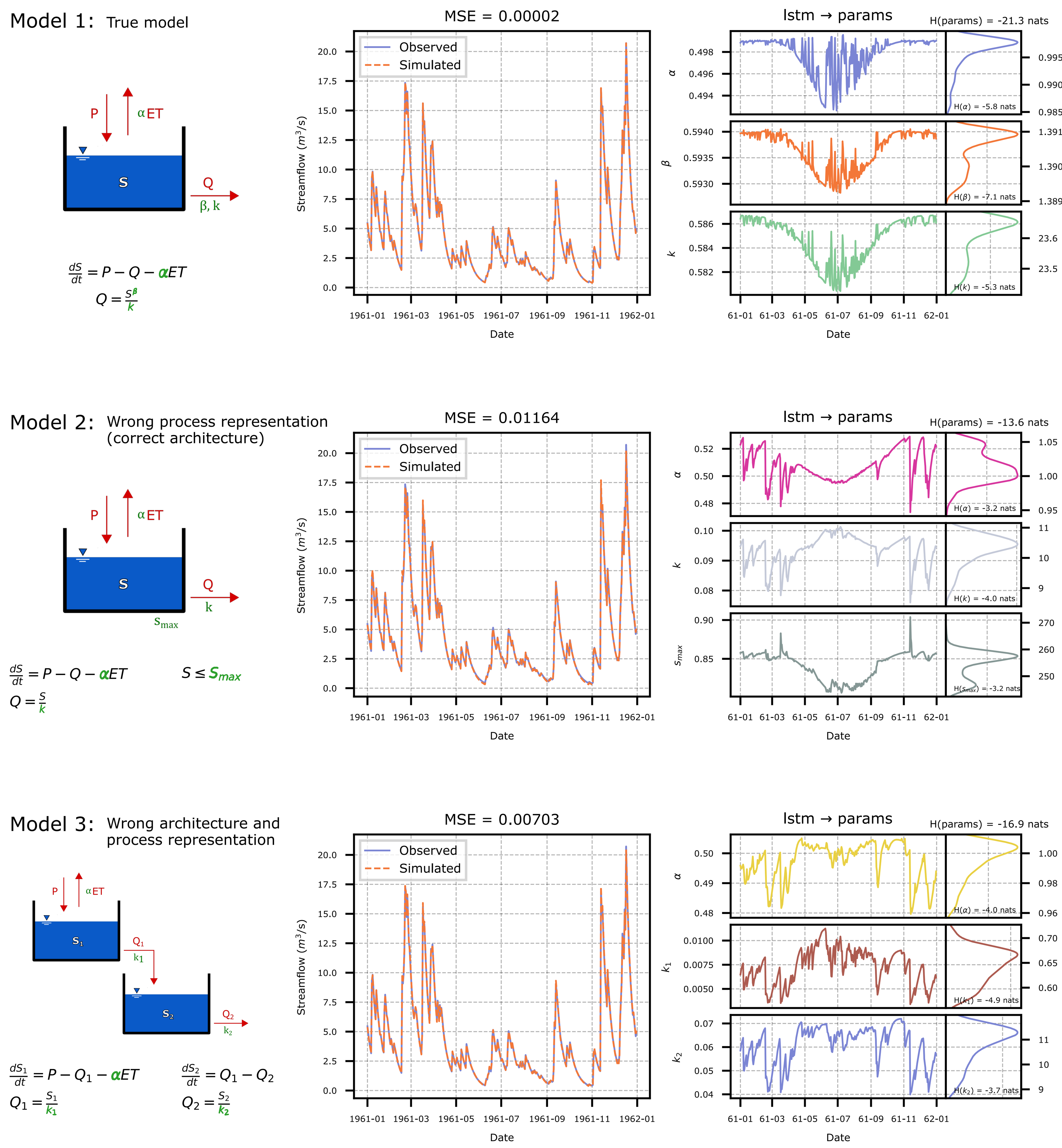


Figure 3: Model diagnostics

Because $H(params)$ is larger for Model 2 than for Model 3, in Model 2 the LSTM is overcompensating for a worse choice of model. Model 3 is then a better representation of Model 1.

CAMELS-GB (thanks Eduardo!)

MODEL	Entropy H(X) [in nats]
LSTM + Bucket	-0.428
LSTM + SHM	-1.926
LSTM + SHM**	-5.921

Table 1: Evaluating predicted recession constants (k_i) or (k_f, k_i, k_b)**

MODEL	Entropy H(X) [in nats]
LSTM + Bucket	-109.94
LSTM + SHM	-92.75
LSTM + Nonsense	-108.36

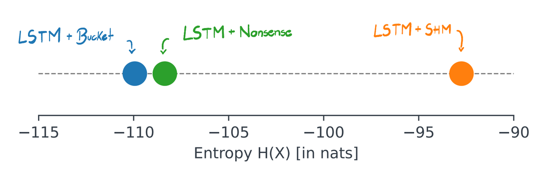


Table 2: Evaluating LSTM hidden states (h_s)