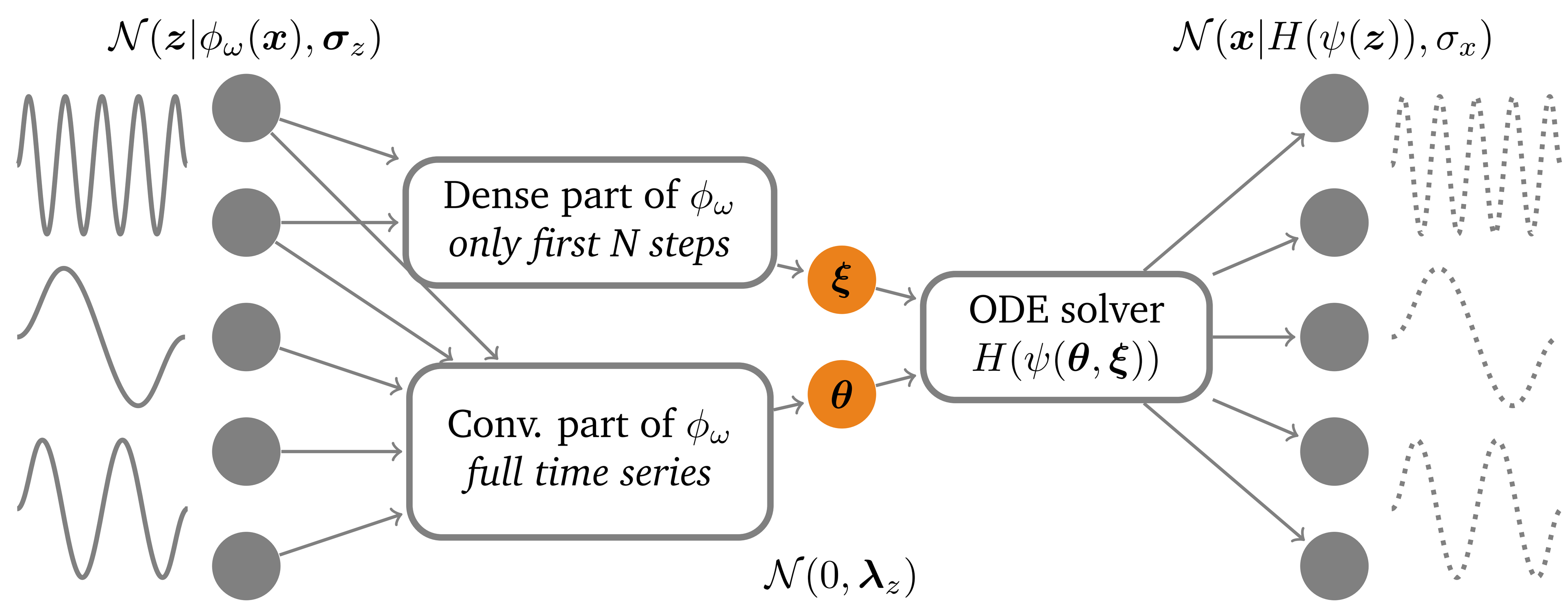
Learning **physical concepts** by **relevance determination**: Identifying **manifolds** of **differential equations.**

Rodent: Relevance determination in ODE

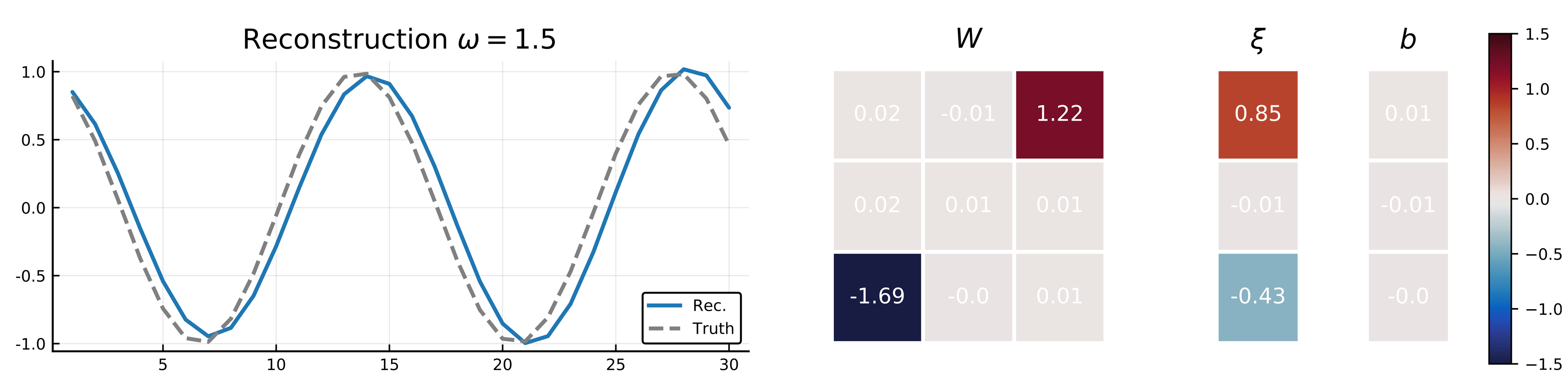
Niklas Heim, Václav Šmídl, Tomáš Pevný — Czech Technical University, Prague

Learning differential equations

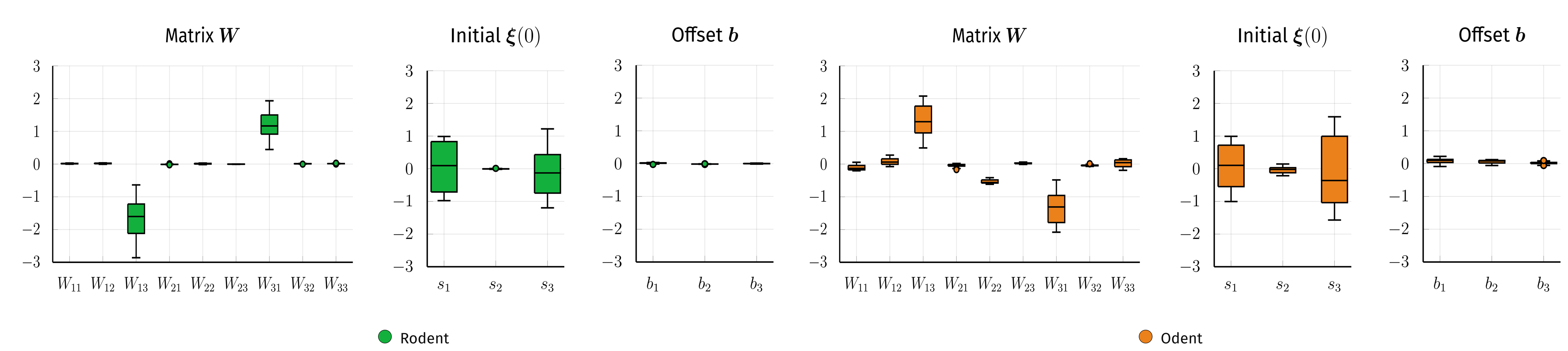
- We want to find the simplest ODE that describes a dynamical system
- By simple we mean: minimal order of ODE and minimum number of non-zero parameters.
- Discover physically meaningful equations that might help understand the underlying process.
- We can learn manifolds of generating models not only a single process



Manifold learning & Reidentification



Rodent reconstructions of a harmonic signal on the left. The heatmaps on the right show the corresponding encodings for the weights $\boldsymbol{W}$, biases $\boldsymbol{b}$, and initial conditions $\boldsymbol{\xi}$. The Rodent reduced the latent space to the four truly relevant parameters.



Latent codes of the Rodent compared to Odent. All redundant parameters are pushed to zero by Rodent while Odent keeps one irrelevant parameter (W_{22}). Additionally we can see that the Rodent squeezes irrelevant parameters closer to zero than Odent.

Advantages of the relevant ODE identifier

- **Explainability.** The parameters of $\boldsymbol{z}$ are decoded through an ODE solver, which gives them physical meaning.
- **Sparsity.** The automatic relevance determination prior on $\boldsymbol{z}$ encourages the simplest solution with fewest non-zero parameters.
- **Partial observations.** Rodent allows learning of an ODE without knowledge of all state trajectories. The observation operator is assumed to be known.
- **Time series.** The convolutional part of the encoder is agnostic to different time series lengths.

Check out the full paper at `http://tiny.cc/f6x6gz` or scan the QR code on the left!
Email `niklas.heim@aic.fel.cvut.cz`

Rodent in depth

Problem definition: We are concerned with time series $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_K]$ with $\boldsymbol{x}_i \in \mathbb{R}^d$ generated from discrete-time, noisy observations

$$\boldsymbol{x}_k = H(\boldsymbol{\xi}(\Delta t k)) + e_k, \tag{1}$$

where $k = 1 \dots K$, noise $e_k \sim \mathcal{N}(0, \sigma_e^2 \mathbf{I})$, and partial observation operator H. The evolution of $\boldsymbol{\xi}(t) \in \mathbb{R}^N$ is governed by an ODE:

$$\frac{\partial \boldsymbol{\xi}}{\partial t} = f(\boldsymbol{\theta}, t) \approx \boldsymbol{W}\boldsymbol{\xi} + \boldsymbol{b}. \tag{2}$$

We aim to learn structure and order of the ODE from a set of trajectories $\{\boldsymbol{X}_i\}_{i=1}^L$ generated by the same generative process but with different $\boldsymbol{\theta}_i$ and different $\boldsymbol{\xi}_i(0)$, for each trajectory, i.e.

$$\boldsymbol{X}_i = H(\psi(\boldsymbol{\theta}_i, \boldsymbol{\xi}_i(0), \boldsymbol{t})) + \boldsymbol{e}, \tag{3}$$

where $\boldsymbol{t} = [0, \Delta t, \dots, K\Delta t]$ and ODE solver ψ. Assuming we observe a system with expected order M, we choose $N \geq M$.

Rodent: By combining the ODE state and its parameters in $\boldsymbol{z} = [\boldsymbol{\theta}, \boldsymbol{\xi}(0)]$ we can define the data likelihood as

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|H(\psi(\boldsymbol{z})), \sigma_x^2). \tag{4}$$

To determine the structure of the ODE, we employ the ARD prior:

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|0, \mathrm{diag}(\boldsymbol{\lambda}_z^2)) \quad p(\boldsymbol{\lambda}_z) = 1/\boldsymbol{\lambda}_z. \tag{5}$$

The posterior of $\boldsymbol{z}$ is prescribed by

$$p(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\phi_\omega(\boldsymbol{x}), \boldsymbol{\sigma}_z^2) \tag{6}$$

where mean $\boldsymbol{\mu}_z = \phi_\omega(\boldsymbol{x})$ is a neural net with parameters ω. The resulting ELBO:

$$\mathcal{L} = \sum_{i=1}^{n} \mathsf{E}_{p(z|x)} \left[\frac{(\boldsymbol{x}_i - \psi(\phi_\omega(\boldsymbol{x}_i) + \boldsymbol{\sigma}_z \odot \boldsymbol{\epsilon}))^2}{2\sigma_e^2} \right] + \frac{nd}{2} \log(\sigma_e) + \sum_{i=1}^{n} \left(\log\left(\frac{\boldsymbol{\lambda}_z^2}{\boldsymbol{\sigma}_z^2}\right) - m + \frac{\boldsymbol{\sigma}_z^2}{\boldsymbol{\lambda}_z^2} + \frac{\phi_\omega(\boldsymbol{x}_i)^2}{\boldsymbol{\lambda}_z^2} \right), \tag{7}$$

with Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, decoder $H(\psi(\boldsymbol{\theta}, \boldsymbol{\xi}(0))) \equiv \phi(\boldsymbol{z})$, and $\dim(\boldsymbol{z}) = m$.

Encoder: The encoder ϕ_ω consists of two parts: (i) A dense net that receives only a few steps of the beginning of the time series, responsible for predicting $\boldsymbol{\xi}(0)$. (ii) A CNN that predicts $\boldsymbol{\theta}$. The CNN averages over the time dimension after the convolutions, which makes it possible to use samples of different length.

Reidentification: We sample a batch of latent codes from the encoder for each input. The samples are used as starting points for another optimization of the reconstruction error R

$$R = \mathrm{MSE}(\phi(\boldsymbol{z}) - \boldsymbol{x}), \tag{8}$$

while keeping all irrelevant parameters fixed. On the left, four parameters, namely W_{13}, W_{31}, ξ_1, and ξ_3 were found to be relevant, so only those are changing during the optimization with respect to R. This means that we stay in the identified model manifold, but are able to extrapolate far beyond the training range.