Effective Latent Differential Equation Models via Attention and Multiple Shooting

Anonymous Author(s)
Affiliation
Address
email

Abstract

The GOKU-net is a continuous-time generative model that allows leveraging prior knowledge in the form of differential equations. We present GOKU-UI, an evolution of the GOKU-nets, which integrates attention mechanisms and a novel multiple shooting training strategy in the latent space. On simulated data, GOKU-UI significantly improves performance in reconstruction and forecasting, outperforming baselines even with 16 times less training data. Applied to empirical human brain data, using stochastic Stuart-Landau oscillators, it is able to effectively capture complex brain dynamics, surpassing baselines in reconstruction and better predicting future brain activity up to 15 seconds ahead. Ultimately, our research provides further evidence on the fruitful symbiosis given by the combination of established scientific insights and modern machine learning.

1 Introduction

Scientific Machine Learning (SciML) is an emerging field combining scientific models with modern data-driven techniques, often yielding increased interpretability, generalizability, and data efficiency. (Baker et al., 2019; von Rueden et al., 2023; Shen et al., 2023). Latent Ordinary Differential Equations (Latent ODEs) (Chen et al., 2018; Rubanova et al., 2019) are VAE-like generative models that encode time series data into a latent space ruled by a differential equation which is parametrized by a neural network. Building on Latent ODEs, Linial et al. (2021) introduced GOKU-nets (Generative ODE Modeling with Known Unknowns), which fundamental difference with the former is the inclusion of a predefined differential equation structure as a prior for the latent dynamics. Compared to LSTM and Latent-ODE on pendulum videos and cardiovascular system modeling, GOKU-net excelled in reconstruction, forecasting, reduced training data needs, and offered better interpretability.

We propose an enhancement to the original GOKU-net architecture which adds attention mechanisms to the main part of the model that infers the parameters of the differential equations. Moreover, to overcome the inherent difficulties of training, we developed a novel strategy to train the GOKU-net based on the multiple shooting technique (Bock & Plitt, 1984; Ribeiro et al., 2020; Turan & Jäschke, 2021) in the latent space. Testing on simulated stochastic oscillators and empirical brain data derived from resting state human functional Magnetic Resonance Imaging (fMRI), our *GOKU-nets with Ubiquitous Inference* (GOKU-UI) surpassed both the original GOKU-net and baselines in accuracy and data efficiency. GOKU-UI exemplifies the potential of melding traditional scientific insights with modern machine learning.

2 Methods

2.1 Basic GOKU-nets

A general model class that we denominate Latent Differential Equation model (Latent DE), illustrated in Figure 1, begins by independently processing each temporal frame x_i with a *Feature Ex-*

Submitted to the DLDE-III Workshop in the 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

tractor. The sequence then passes through a *Pattern Extractor* which aims to learn the distribution of the initial conditions and possibly of the parameters for the DE that will be subsequently integrated. Lastly, a *Reconstructor* transforms the solution back to the input space. Training follows the standard VAE approach, optimizing the evidence lower bound (ELBO) (Kingma & Welling, 2013).

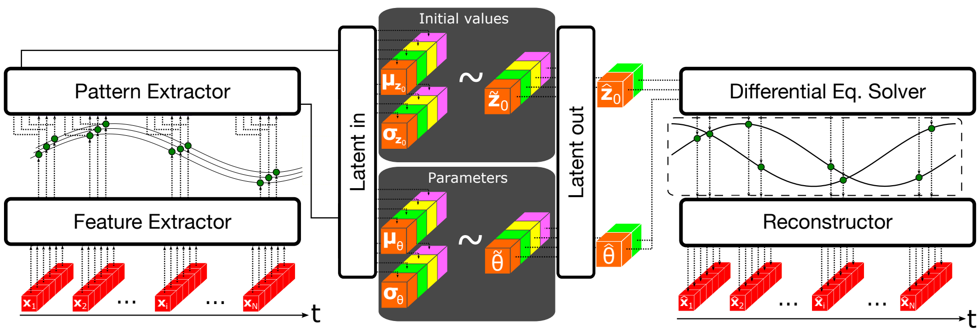

Figure 1: Diagram of a Latent Differential Equation model.

The original GOKU-net by Linial et al. (2021), understood as a Latent DE model, uses a ResNet with dense NNs as the Feature Extractor. Its Pattern Extractor employs an RNN for the initial conditions and a bidirectional LSTM for the ODE parameters. Unlike Latent ODEs, which parameterize the differential equation using another NN, in this case, the differential equation is predefined, incorporating prior system knowledge. GOKU-net employs fully connected layers for the Latent in/out, and a ResNet similar to the Feature Extractor for the Reconstructor.

2.2 GOKU-UI

Attention mechanism In our GOKU-UI model, we incorporate a basic attention mechanism (Vaswani et al., 2017) into the Pattern Extractor, specifically when learning the differential equation parameters. Namely, instead of keeping the last element of the bidirectional LSTM (BiLSTM) used in the original GOKU-net model, all of its sequential outputs pass through a dense layer with softmax activation to calculate the attentional scores that weight the sum of all the BiLSTM outputs in order to obtain its final output.

Multiple Shooting Gradients calculations through differential equations often lead to highly complex loss landscapes (Ribeiro et al., 2020; Metz et al., 2021). Turan & Jäschke (2021) showed that training Neural ODEs even on basic oscillatory data could be problematic, resulting in trajectories resembling moving averages. To address this, they utilized *multiple shooting* techniques (Bock & Plitt, 1984; Diehl et al., 2006; Baake et al., 1992; Ribeiro et al., 2020). This method divides the differential equation's time span into segments, independently inferring each segment's initial conditions. These segments are then merged, enforcing continuity during the optimization.

However, applying the multiple shooting method to GOKU-nets is not straightforward. Firstly, in most cases that use this method, such as in Turan & Jäschke (2021), the differential equations are typically directly modeling the observable data, having direct access to the true initial conditions for each window. In the case of GOKU-nets, the dynamics modeled by differential equations occur in the latent space, which is being learned simultaneously; as a result, such true initial conditions are not available. Secondly, it is necessary to determine how the method will behave in relation to the parameters of the differential equation, which in the case of Neural ODEs are implicitly learned as part of their parameterization through the neural network.

Our proposal for extending the multiple shooting method to GOKU-nets is as follows. After passing through the Feature Extractor, we divide the temporal interval in the latent space in such a way that the Pattern Extractor generates in parallel different initial conditions for each temporal window, but provides a single set of parameters for the differential equations that will be shared by all windows. By this strategy, we maintain the potential benefits inherent to the multiple shooting method while leveraging the information available in a wider temporal range for the task of parameter inference, which is generally more challenging than estimating initial conditions. As mentioned before, we do not have access to target true initial conditions, however, what we can strive to achieve is the continuity of trajectories across different windows. To this end, these intervals are defined by overlapping the last temporal point of each window with the first one of the following and the goal is to minimize the distance between these points. Specifically, we employ regularization in the cost function when training the model, quadratically penalizing the discrepancy in the latent space of the overlapping points, that is, between the initial condition of each window and the end point of its preceding segment.

Non-variational GOKU-nets outperformed their variational counterparts in our experiments (see Supplementary Information B.3). Consequently, we utilized non-variational GOKU-nets for subsequent results. Rather than sampling from normal distributions in the latent space (Figure 1), we used mean values μ_{z_0} and μ_θ. The resulting model's cost function excludes the KL divergence term from the ELBO but includes the reconstruction term, computed as the normalized mean squared error between the model's outputs and inputs. Continuity regularization from multiple shooting training is also incorporated.

2.3 Experiments

We assessed our attention and multiple shooting enhancements on synthetic data based stochastic Stuart-Landau oscillators and empirical human brain data. We compared various GOKU-model variations (basic, attention-enhanced) using either the original single shooting or the new multiple shooting method. Baseline models included LSTM, Latent ODE, and a naïve model. For fairness, the LSTM and Latent ODE maintained the GOKU-net's architecture, differing only at the differential equation layer. Here, the Neural ODE replaced the differential equation layer for the Latent ODE model, while an LSTM did for the other. The Latent ODE and LSTM sizes were adjusted to match GOKU-UI's total number parameters. Naïve predictors utilized time-averaged input values for constant predictions. Detailed training procedures, models architectures, and hyperparameters are available in the Supplementary Information.

Simulated data Our simulated data derives from networks of stochastic Stuart-Landau (SL) oscillators, a standard model for resting state fMRI brain dynamics (Jobst et al., 2017; Deco et al., 2017). The dynamics for an oscillator node in an N node network is:

$$\begin{aligned}
\dot{x}_j &= Re(\dot{z}_j) = [a_j - x_j^2 - y_j^2]x_j - \omega_j y_j + G\sum_{i=1}^{N} C_{ij}(x_i - x_j) + \beta\eta_j(t) \\
\dot{y}_j &= Im(\dot{z}_j) = [a_j - x_j^2 - y_j^2]y_j + \omega_j x_j + G\sum_{i=1}^{N} C_{ij}(y_i - y_j) + \beta\eta_j(t)
\end{aligned} \tag{1}$$

where C_{ij} is the network's connectivity matrix, G is a global coupling factor, and η_j is Gaussian noise. Each node has distinct bifurcation parameters a_j and frequencies ω_j. During the construction of our dataset, we perform a dimensionality augmentation on the network of oscillators, which are utilized as latent dynamics. Specifically, we apply a fixed random linear transformation, $f : \mathbb{R}^{2N} \to \mathbb{R}^D$, to the latent trajectories of each sample, with $D = 784$. Each sample corresponds to a unique random set of initial conditions and parameters for the $N = 3$ coupled oscillators.

Empiric data We assessed our models on resting-state fMRI data from the human brain, sourced from 153 subjects in the Track-On HD study (Klöppel et al., 2015). After preprocessing as outlined in Polosecki et al. (2020), we applied a 20-component Canonical ICA (Varoquaux et al., 2010), retaining 11 components post artifact removal. This resulted in 306 samples, each with 160 time points collected every 3 seconds. We reserved 20% of the data (n=60) for testing, ensuring balanced representation, and used the remaining (n = 246) for training and validation. The first 114 time points from each of these samples were used for model training, with the remainder reserved for validation. The GOKU-UI model employed 20 Stuart-Landau oscillators (Eq. 1) in its latent space.

3 Results

We assessed four GOKU-net variants for both reconstruction and forecast tasks, considering single/multiple shooting methods and the presence or absence of attention. We compared them to three baselines: LSTM, Latent ODEs, and a naïve predictor. Although models were exclusively trained for reconstruction, we evaluated their forecasting during testing. The normalized root mean square error (NRMSE) measures the prediction error against the target ground truth.

Simulated data Figure 2a depicts GOKU-net variants with multiple shooting having significantly reduced errors on a synthetic dataset of three stochastic Stuart-Landau oscillators. Attention mechanism enhanced performance, especially in single shooting. GOKU-UI, combining attention and multiple shooting, remained the top performer, with Wilcoxon tests confirming significance at p-values < 0.02 after Holm correction. Latent ODEs underperformed, resembling the naïve predictor, while LSTMs surpassed basic GOKU-nets. Notably, when trained on just 150 unique samples, GOKU-UI outperformed all other single shooting models, even those trained on datasets 32 times larger. In forecasting, as shown in Figure 2b, attention-aided GOKU-nets excelled over LSTMs, with GOKU-UI standing out up to 150 samples (p-values < 0.02, Wilcoxon signed-rank, Holm corrected), beyond which its performance was statistically indistinguishable from that of the basic GOKU model with multiple shooting (p-values > 0.05, Wilcoxon signed-rank tests, Holm corrected).

Empirical data Figure 2c demonstrates that the attention mechanism didn't boost single shooting GOKU-net performance. Yet, multiple shooting training significantly improved results, with the GOKU-UI model, merging both techniques, reducing the median reconstruction error by five times compared to single shooting baselines. Furthermore, GOKU-UI had a significantly lower reconstruction NRMSE than the multiple shooting GOKU basic model (p < 0.04, Wilcoxon signed-rank test). In forecasting, GOKU-UI outperformed other models, achieving lower forecast errors for up to 15 seconds of brain activity.

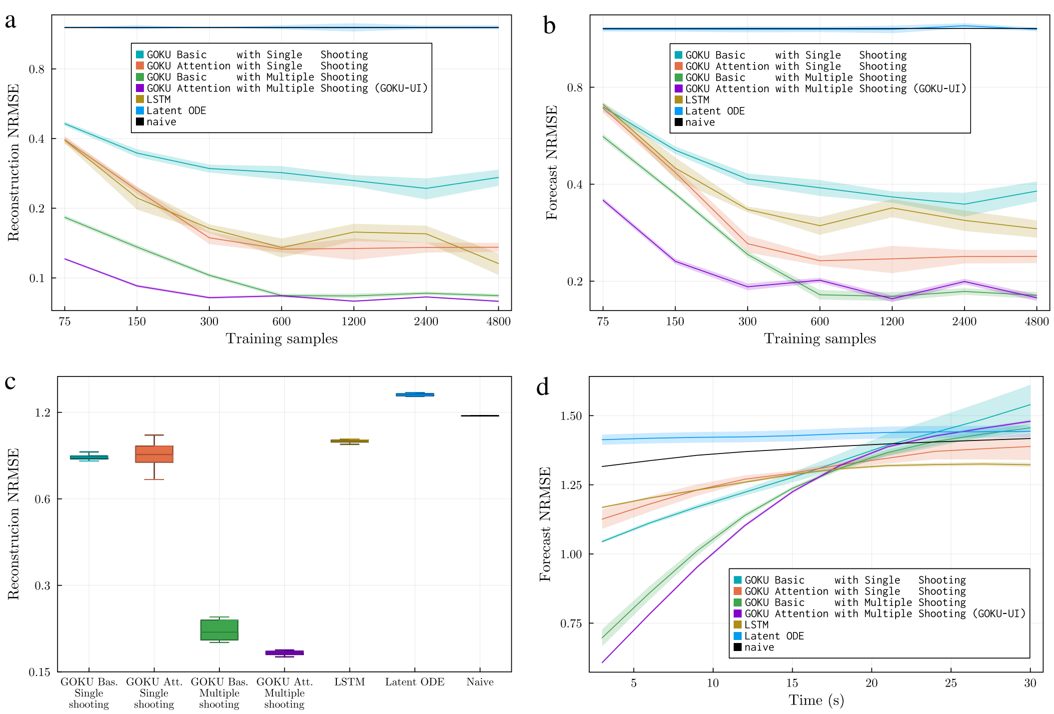

Figure 2: Comparison of reconstruction (left panels) and forecast (right panels) performances on the synthetic Stuart-Landau (top panels) and fMRI (bottom panels) test data sets, using normalized RMSE. Averages are taken across input dimensions and time span, with shaded areas indicating standard errors from multiple training runs with varied random seeds. Forecasts in panel b are assessed over a 20 time-step horizon.

4 Conclusion

We enhanced the GOKU-nets with attention and multiple shooting, with the latter yielding the most impact. The resulting model, GOKU-UI, showed improved performance and data efficiency on both synthetic and empirical brain data. By leveraging established scientific insights into modern machine learning, GOKU-UI was able to encode whole-brain dynamics into a latent representation, learning a low-dimensional interpretable dynamical system model that could offer insights into brain functionality and open avenues for multiple practical applications.

Applying GOKU-UI to new problems might not be as straightforward as a general-purpose black-box neural network model due to the need for a specific differential equation. Still, with guidance from dynamical systems theory, it is not only feasible but also beneficial.

References

Baake, E., Baake, M., Bock, H., and Briggs, K. Fitting ordinary differential equations to chaotic data. *Physical Review A*, 45(8):5524, 1992.

Baker, N., Alexander, F., Bremer, T., Hagberg, A., Kevrekidis, Y., Najm, H., Parashar, M., Patra, A., Sethian, J., Wild, S., Willcox, K., and Lee, S. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence, 2 2019. URL `https://www.osti.gov/biblio/1478744`.

Bock, H. G. and Plitt, K.-J. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations, 2018. URL `https://arxiv.org/abs/1806.07366`.

Datseris, G., Isensee, J., Pech, S., and Gál, T. Drwatson: the perfect sidekick for your scientific inquiries. *Journal of Open Source Software*, 5(54):2673, 2020. doi: 10.21105/joss.02673. URL `https://doi.org/10.21105/joss.02673`.

Deco, G., Kringelbach, M. L., Jirsa, V. K., and Ritter, P. The dynamics of resting fluctuations in the brain: metastability and its dynamical cortical core. *Scientific reports*, 7(1):3095, 2017.

Diehl, M., Bock, H. G., Diedam, H., and Wieber, P.-B. Fast direct multiple shooting algorithms for optimal robot control. *Fast motions in biomechanics and robotics: optimization and feedback control*, pp. 65–93, 2006.

Innes, M., Saba, E., Fischer, K., Gandhi, D., Rudilosso, M. C., Joy, N. M., Karmali, T., Pal, A., and Shah, V. Fashionable modelling with flux. *CoRR*, abs/1811.01457, 2018. URL `https://arxiv.org/abs/1811.01457`.

Jobst, B. M., Hindriks, R., Laufs, H., Tagliazucchi, E., Hahn, G., Ponce-Alvarez, A., Stevner, A., Kringelbach, M. L., and Deco, G. Increased stability and breakdown of brain effective connectivity during slow-wave sleep: mechanistic insights from whole-brain computational modelling. *Scientific reports*, 7(1):1–16, 2017.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Klöppel, S., Gregory, S., Scheller, E., Minkova, L., Razi, A., Durr, A., Roos, R. A., Leavitt, B. R., Papoutsi, M., Landwehrmeyer, G. B., et al. Compensation in preclinical huntington's disease: evidence from the track-on hd study. *EBioMedicine*, 2(10):1420–1429, 2015.

Linial, O., Ravid, N., Eytan, D., and Shalit, U. Generative ODE modeling with known unknowns. In *Proceedings of the Conference on Health, Inference, and Learning*. ACM, apr 2021. doi: 10.1145/3450439.3451866. URL `https://doi.org/10.1145%2F3450439.3451866`.

Ma, Y., Gowda, S., Anantharaman, R., Laughman, C., Shah, V., and Rackauckas, C. Modelingtoolkit: A composable graph transformation system for equation-based modeling, 2021.

Metz, L., Freeman, C. D., Schoenholz, S. S., and Kachman, T. Gradients are not all you need. *arXiv preprint arXiv:2111.05803*, 2021.

Misra, D. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

Polosecki, P., Castro, E., Rish, I., Pustina, D., Warner, J. H., Wood, A., Sampaio, C., and Cecchi, G. A. Resting-state connectivity stratifies premanifest huntingtons disease by longitudinal cognitive decline rate. *Scientific reports*, 10(1):1–15, 2020.

Rackauckas, C. and Nie, Q. Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1):15, 2017.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. Universal differential equations for scientific machine learning, 2020. URL `https://arxiv.org/abs/2001.04385`.

Ribeiro, A. H., Tiels, K., Umenberger, J., Schön, T. B., and Aguirre, L. A. On the smoothness of nonlinear system identification. *Automatica*, 121:109158, 2020.

Rubanova, Y., Chen, R. T., and Duvenaud, D. K. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fenicia, F., Kifer, D., Li, L., et al. Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, pp. 1–16, 2023.

Turan, E. M. and Jäschke, J. Multiple shooting for training neural differential equations on time series. *IEEE Control Systems Letters*, 6:1897–1902, 2021.

Varoquaux, G., Sadaghiani, S., Pinel, P., Kleinschmidt, A., Poline, J.-B., and Thirion, B. A group model for stable multi-subject ica on fmri datasets. *Neuroimage*, 51(1):288–299, 2010.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., and Schuecker, J. Informed machine learning a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2023. doi: 10.1109/TKDE.2021.3079836.

Supplementary Information

A Models architectures

Referring to the diagram in Figure 1, the specific architecture used for the different models, for both simulated and empirical data experiments, is as follows:

A.1 Basic GOKU-nets

Feature Extractor

ResNet with 4 fully-connected layers, each with 200 neurons and using mish activation functions (Misra, 2019). Input dimension = number of dimensions in the input data. Output dimension = 128.

Pattern Extractor

Initial values path: an RNN with 2 layers and 64 neurons in each with ReLU activations. Input dim = 128. Output dim = 64.

Parameters path: Bidirectional LSTM with 2 layers and 64 neurons in each. Input dimension = 128. Output dimension = 128. Note that the dimension of the output of the forward LSTM and the backward LSTM are 64 but when concatenating them, the resulting output dimension is the given one.

Latent in

Initial values path: single-layered fully connected NN. Input dim = 64. Output dim = 64.

Parameters path: fully connected NN with 1 layer. Input dim = 128. Output dim = 128.

Latent out

Initial values path: fully connected NN with 2 layers and 200 neurons in the hidden layer, using no activation function (identity). Input dim = 64. Output dim = number of state variables of the differential equation.

Parameters path: fully connected NN with 2 layers and 200 neurons in the hidden layer, using sigmoid activation function. The parameters are projected from the interval [0, 1] to the desired range when integrating the differential equation. Input dim = 128. Output dim = number of parameters of the differential equation.

Differential Equation layer

The predefined differential equation is solved numerically for each of the sets of parameters and initial conditions provided by the previous layer. The output is the trajectories at time points equivalent to the input data.

Reconstructor

ResNet is similar to the one in the Feature Extractor, except that in this case the input dimension is the number of state variables of the differential equation and the output dimension is the one corresponding to the input data.

A.2 GOKU-nets with attention

With the exception of the Pattern Extractor, the rest of the layers in the GOKU-nets with attention model remain identical to those in the basic GOKU-nets.

Pattern Extractor

Initial values path: LSTM with 1 layer. Input dimension = 128. Output dimension = 128.

Parameters path: Bidirectional LSTM (BiLSTM) with 1 layer. Input dim = 128. Output dim = 128. A fully connected NN with input and output dimensions of 128 is used for the attention mechanism. This attention NN processes all the output sequences of the BiLSTM, after which a softmax is applied across the time dimension in order to obtain the attentional scores that will be used in the weighted sum of all the time steps returned by the BiLSTM.

A.3 LSTM baseline model

The whole architecture is the same as in the basic GOKU-net, except for the Differential Equation layer, which is replaced by an LSTM:

LSTM layer

We used a single-layered LSTM with input and output dimensions set to z_dim. This value is determined in each experiment to ensure that the total number of parameters in the LSTM model closely matches that of the corresponding GOKU-UI. For the simulated dataset experiments, we set $z\_dim = 42$. In the case of the empirical dataset experiments, $z\_dim = 105$. The LSTM operates recursively. It takes as its first input the value equivalent to the initial condition in differential equations. Subsequently, the model feeds back its last output as the new input, continuing this process until the number of time steps matches that of the model's input.

A.4 Latent ODE baseline model

The whole architecture is the same as in the basic GOKU-net, except for the Differential Equation layer, which is replaced by a Neural ODE:

Neural ODE layer

Neural ODE is parametrized by a fully connected NN with 3 layers and $node\_hidden\_dim$ neurons in each. The input and output dimensions are given by z_dim, which is the number of state variables. In the case of the simulated dataset experiments, the number of state variables was selected to match the true latent dimension $z\_dim = 6$ and the number of neurons in each layer was adjusted so that the total number of parameters in the model matched as closely as possible that of the corresponding GOKU-UI, resulting in $node\_hidden\_dim = 137$. On the other hand, in the case of the fMRI experiments, the number of state variables was set to $z\_dim = 20$ and $node\_hidden\_dim = 317$, also matching the total number of parameters of the corresponding GOKU-UI model.

B Comprehensive description of experiments

B.1 Simulated dataset generation

The high-dimensional simulated dataset used for training the model was constructed based on the simulations of 3 coupled Stuart-Landau oscillators (Eqs. 1) with different random sets of parameters. Each set of parameters corresponds to a different training sample. Whenever we used the Stuart-Landau model in our experiments (both when generating the dataset and when using it inside the GOKU-nets), the time was rescaled by multiplying the right-hand side of Eqs. 1 by 20. Thus, when integrating the equations with the used $dt = 0.05$, the input sequences of length 46 time steps contain a few oscillations. The parameters a, ω and C were sampled from uniform distributions within the following ranges

$$a \in [-0.2, 0.2]; \quad \omega \in [0.08\pi, 0.14\pi]; \quad C \in [0, 0.2]$$

while $G = 0.1$ and $\eta = 0.02$. On the other hand, the initial conditions for the six state variables were sampled from uniform distributions within the ranges $[0.3, 0.4]$. For each set of parameters and initial conditions, the system is integrated with the `SOSRI` solver, a Stability-optimized adaptive strong order 1.5 and weak order 2.0 for diagonal/scalar Ito SDEs, from the DifferentialEquations.jl Julia package (Rackauckas & Nie, 2017). The complete time span of the integration is 35 units of time and the trajectories are saved every 0.05, resulting in 700 time points. The first 100 time steps are trimmed, in order to remove possible initial transients. Afterwards, a random linear transformation is independently applied to each of the 600 remaining time steps, in order to obtain 784 dimensions. In other words, every state vector of length 6 from each sample is multiplied by the same 784Œ6 matrix, initialized randomly sampling from a uniform distribution in the range [-1, 1]. A training dataset was created with 5000 samples, which serves as the source for the different training instances using different sizes of training sets (see Figure 2a). A different test set with 900 samples was created for the posterior evaluations of the model.

B.2 Empirical dataset generation

We used resting state fMRI data from 153 participants, obtained from the Track-On HD study (Klöppel et al., 2015). The data underwent pre-processing, as described in Polosecki et al. (2020), and a 20-component Canonical ICA (Varoquaux et al., 2010) was performed. Upon inspecting the resulting 20 components, 9 were identified as artifacts and thus discarded, leaving 11 components for further analysis in our experiments. Each subject contributed data from two visits, accumulating a total of 306 data samples. Each sample comprised 160 time points, obtained at a temporal resolution of 3 seconds.

For our investigation, we set aside approximately 20% of the data samples (n=60) for testing, while ensuring balanced representation from sex, condition, and measurement site. The remaining data samples (n=246) were allocated for training and validation. Specifically, the first 114 time points from each of these samples were utilized for model training, with the remainder reserved for validation and early training termination. Finally, the training, validation, and test splits were all normalized by the standard deviation of the training set.

B.3 Training settings

All the experiments underwent the same training procedure with identical hyperparameters, which will be described here.

The input sequence length for all the models was 46 time steps, and the batch size was set at 64. As described above, the full length of each sample in the training sets was 600 time steps for the synthetic dataset and 114 for the fMRI dataset. The procedure for generating a batch of training data is as follows: First, 64 samples that have not been used previously in the current training epoch are randomly selected. Then, for each sample, a 46 time-step-long interval is randomly chosen within the 600 or 114 time steps available in the full sample length.

The GOKU-net based models, contain the same Stuart-Landau differential equations as described above, however, the allowed ranges of parameters differ from the ones used during the generation of the synthetic dataset. In order to be closer to a real world use-case we allow for a wider range of parameters than those actually used for generating the data, since in principle one would not know the true range:

$a \in [-1, 1]; \quad \omega \in [0, 1]$

while keeping, the other parameters the same except of the connectivity in the empirical fMRI training, in which case it was allowed to be negative: $C \in [-0.2, 0.2]$. The differential equations definitions were optimized for higher computational performance with the help of ModelingToolkit.jl (Ma et al., 2021). During training, they were solved with the `SOSRI` solver, a Stability-optimized adaptive strong order 1.5 and weak order 2.0 for diagonal/scalar Ito SDEs, from the DifferentialEquations.jl Julia package (Rackauckas & Nie, 2017). The sensitivity algorithm used was `ForwardDiffSensitivity` from the SciMLSensitivity.jl package (Rackauckas et al., 2020). The models were defined and trained within the deep learning framework of the Flux.jl package (Innes et al., 2018). The experiments were managed using DrWatson.jl package (Datseris et al., 2020).

The model was trained with Adam with a weight decay of 10^{-10}, and the learning rate was dynamically determined by the following schedule. The learning rate begins with a linear growth (also referred to as *learning rate warm-up*) from 10^{-7}, escalating up to 0.005251 across 20 epochs. Afterwards, it maintains that value until the validation loss stagnates (has not achieved a lower value for 50 epochs), at which point it starts a sinusoidal schedule with an exponentially decreasing amplitude.

For the multiple shooting training, all the presented experiments used a time window length of 10, therefore partitioning 46-time-steps-long sequences into 5 windows with their endpoints overlapping. The regularization coefficient in the loss function for the continuity constraint had a value of 2.

Since we found that models with variational versions of the GOKU-nets underperformed their non-variational versions, all the results presented in this work were obtained using non-variational GOKU-nets. This is, instead of sampling from normal distributions in the latent space as depicted in Figure 1, we pass forward the mean values μ_{z_0} and μ_θ. Thus, the associated loss function does not have the KL divergence term associated with the ELBO but retains the reconstruction loss given by the mean squared error between the output of the model and the input, normalized by the mean absolute value of the input. In addition, when multiple shooting training is employed, the extra term regarding the continuity constraint is included in the loss function. This extra term consists of the mean squared differences between the last point of a window and the initial from the next one, divided by the number of junctions and multiplied by a regularization coefficient. Please, note that this continuity regularization is performed in the state space of the differential equation and not in the input space.

C Reconstruction plots

To provide a visual representation of the model's performance, this section presents trajectories from both the synthetic and empirical fMRI test sets, along with their corresponding reconstructions by GOKU-UI and the original GOKU-nets (lacking attention mechanisms and trained with single shooting). The x-axis represents time steps in all cases. To display representative cases, samples were selected based on their mean reconstruction RMSE being closest to the median error across all samples. For the synthetic data, 11 components were randomly selected for display in Figures 3 and 4, due to the impracticality of displaying all 784 components. Each figure displays results from different instances of models, all trained with 4800 samples but each initialized with a unique random seed. For the fMRI data, all 11 ICA components are displayed in Figures 5 and 6.

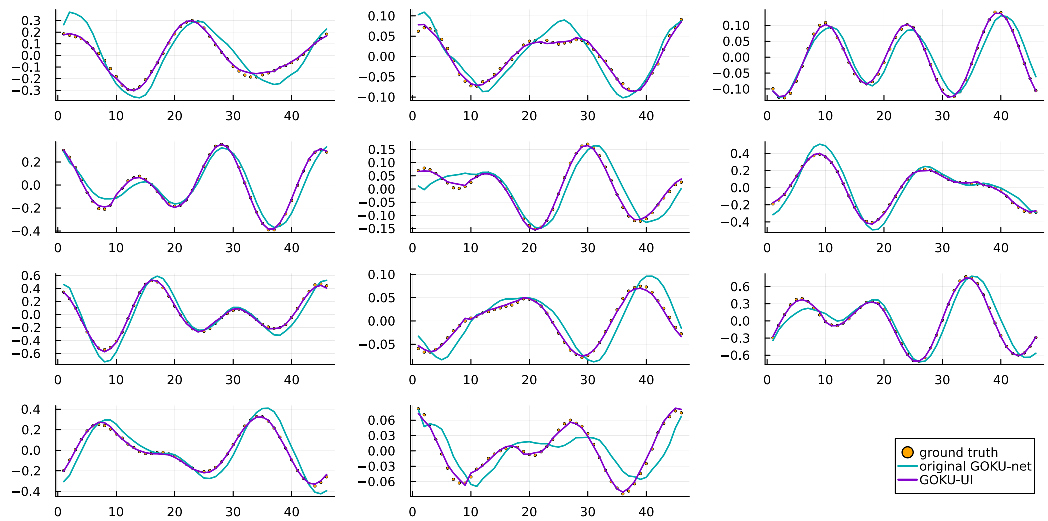

Figure 3: Representative example of a 46-time-step input sequence from the synthetic test set, accompanied by its reconstructions from both GOKU-UI and the original GOKU-nets (lacking attention mechanisms and trained with single shooting). The sample was selected so that its RMSE was the closest to the median error across all samples. 11 randomly selected components out of the 784 are displayed.

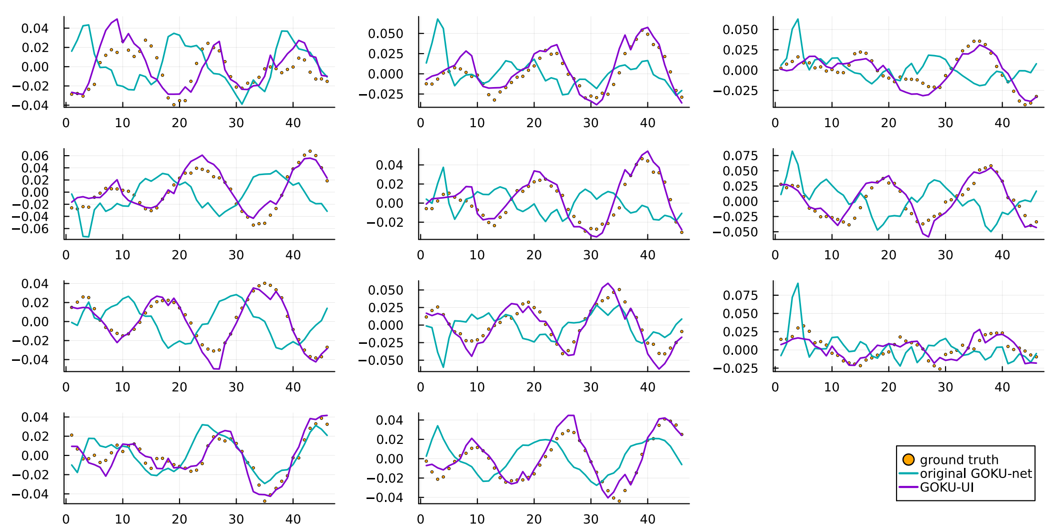

Figure 4: Representative example of a 46-time-step input sequence from the synthetic test set, accompanied by its reconstructions from both GOKU-UI and the original GOKU-nets (lacking attention mechanisms and trained with single shooting). The sample was selected so that its RMSE was the closest to the median error across all samples. 11 randomly selected components out of the 784 are displayed. This figure is similar to the previous one but presents results from different instances of the trained models, each initialized with a unique random seed.

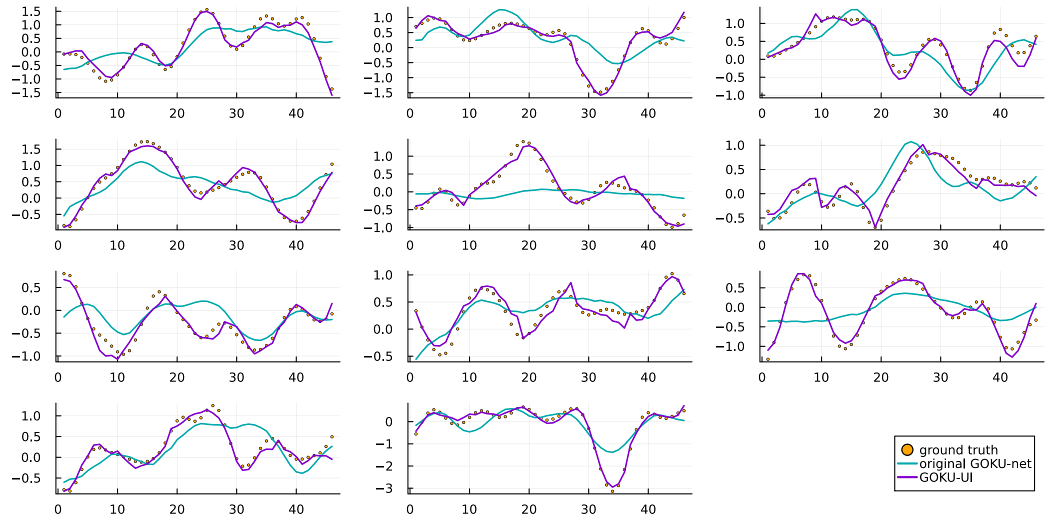

Figure 5: Representative example of a 46-time-steps input sequence for all considered ICA components from the empirical fMRI test set, accompanied by its reconstructions from both GOKU-UI and the original GOKU-nets (lacking attention mechanisms and trained with single shooting). The sample was selected so that its RMSE was closest to the median error across all samples. The x-axis represents time steps, each corresponding to 3 seconds.

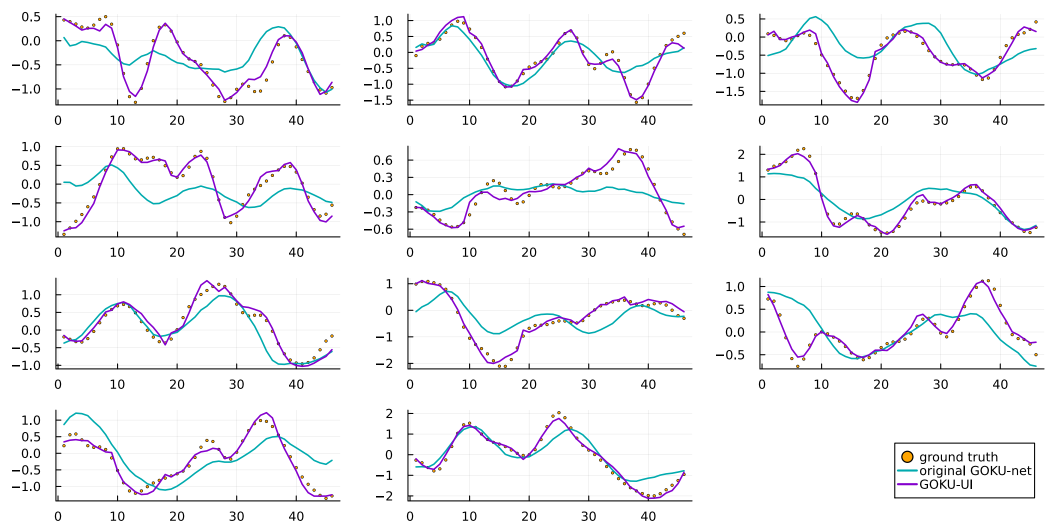

Figure 6: Representative example of a 46-time-steps input sequence for all considered ICA components from the empirical fMRI test set, accompanied by its reconstructions from both GOKU-UI and the original GOKU-nets (lacking attention mechanisms and trained with single shooting). The sample was selected so that its RMSE was closest to the median error across all samples. This figure is similar to the previous one but presents results from different instances of the trained models, each initialized with a unique random seed. The x-axis represents time steps, each corresponding to 3 seconds.