SIMBAML: CONNECTING MECHANISTIC MODELS AND MACHINE LEARNING WITH AUGMENTED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Training sophisticated machine learning (ML) models requires large datasets that are difficult or expensive to collect for many applications. If prior knowledge about system dynamics is available, mechanistic representations can be used to supplement real-world data. We present SimbaML (Simulation-Based ML), an open-source tool that unifies realistic synthetic dataset generation from ordinary differential equation-based models and the direct analysis and inclusion in ML pipelines. SimbaML conveniently enables investigating transfer learning from synthetic to real-world data, data augmentation, identifying needs for data collection, and benchmarking physics-informed ML approaches. SimbaML is available from `https://anonymous.4open.science/r/Simba_ML-4884`.

1 INTRODUCTION AND RELATED WORK

The success of machine learning (ML) models highly depends on the quality and quantity of available data. However, collecting real-world data is costly and time-consuming. Recent advances in generative models that produce synthetic data, such as generative adversarial networks (Goodfellow et al., 2020), variational autoencoders (Wan et al., 2017), and diffusion models (Sohl-Dickstein et al., 2015), do not fully alleviate this issue as these models need to be trained on large data corpora themselves and cannot extrapolate out-of-distribution. Scientific communities have commonly developed a wealth of domain knowledge that should be leveraged (Baker et al., 2018; Alber et al., 2019). Detailed prior knowledge of interactions between modeled entities can be represented by mechanistic models, such as ordinary differential equations (ODEs). They have been used to simulate the dynamical behavior of various systems (Herty et al., 2007; Harush & Barzel, 2017; Hass et al., 2019; Baum et al., 2019). Consequently, they allow physics-informed learning via observational bias (Karniadakis et al., 2021) and generating datasets for ML benchmarks. Multiple frameworks combining data generation from mechanistic models with ML have recently been proposed (Otness et al., 2021; Takamoto et al., 2022; Hoffmann et al., 2021), see Table 1 in the supplement for details. However, they do not allow for generating realistic data by simulating measurement errors or missing data. In addition, they are not designed for easy extension to other mechanistic models, mainly focus on benchmarking ML models, and commonly do not provide transfer learning functionalities.

We introduce SimbaML, an all-in-one framework for integrating prior knowledge of ODE models into the ML process by synthetic data augmentation. SimbaML allows for the convenient generation of realistic synthetic data by sparsifying and adding noise. Furthermore, our framework provides customizable pipelines for various ML experiments, such as identifying needs for data collection and transfer learning.

2 SIMBAML: FEATURES AND USE CASES

SimbaML provides diverse functionalities to generate synthetic data to improve ML tasks (see Figure 1 A). As for the simulation part, we obtain time-series data by solving user-defined ODE systems. SimbaML can add different types of noise and sparsify data by randomly removing time points to make time-series data more realistic. Noise, kinetic parameters, as well as initial condition values, can be easily configured by the user.

Furthermore, SimbaML offers multiple pipelines covering data pre-processing, training, and evaluation of ML models. In addition to supporting standard ML approaches, SimbaML allows for the

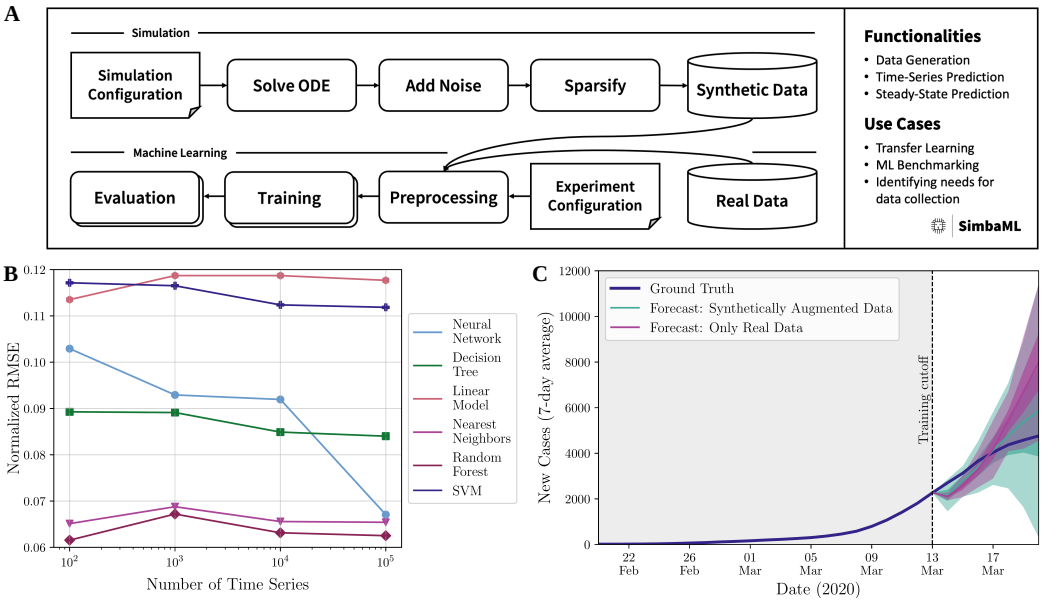

Figure 1: (A) Overview of the SimbaML framework. (B) Performance of various ML models for different synthetic dataset sizes simulated using a biochemical pathway model. SimbaML conveniently allows end-to-end evaluation of required dataset properties, given prior knowledge of the system dynamics. (C) Comparison of two Covid-19 7-day forecasts from March 13, 2020, using a probabilistic neural network augmented with synthetic data generated by SimbaML and real-world data (we show 50% and 85% prediction intervals). This suggests that including prior knowledge using SimbaML can improve the quality of forecasts.

effortless integration of any other model, for example, from Keras, PyTorch Lightning, and scikit-learn. As all pipelines can be customized based on configuration files, SimbaML enables highly automated ML experiments. Overall, SimbaML provides well-tested functionalities (100% test coverage) for various use cases, ranging from transfer learning to benchmarking and identifying needs for data collection.

We illustrate the versatility of SimbaML in two use cases. First, we use SimbaML to identify needs for data collection. We generate multiple time-series datasets with noise for a complex biochemical model of a signaling pathway (Huang & Ferrell, 1996). Based on the performance comparison for different ML models (see Figure 1 B), we can make an informed decision to use the random forest or nearest neighbor approach for smaller dataset sizes. The neural network becomes a viable candidate if sufficient data is available.

Second, we use SimbaML to augment scarce real-world data (Robert Koch-Institut, 2022) for a COVID-19 time series forecasting task (see Figure 1 C). We generate noisy synthetic time series using the Susceptible-Infected-Recovered (SIR) epidemiological model (Kermack & McKendrick, 1927) with randomly sampled kinetic parameter values. Exemplary predictions on synthetically augmented training data improve compared to predictions based solely on observed data.

3 Conclusion

We present SimbaML, an all-in-one solution for generating realistic data from ODE models and leveraging them for improved ML experiments. For two exemplary use cases, we show how SimbaML effectively utilizes prior knowledge via ODEs for ML tasks in settings with limited data availability. As an open-source Python package, we plan to extend SimbaML with additional functionalities, for example, out-of-the-box physics-informed and graph neural networks, to maximize the potential of mechanistic models for ML.

URM STATEMENT

The authors acknowledge that all key authors of this work meet the URM criteria of ICLR 2023 Tiny Papers Track.

REFERENCES

Mark Alber, Adrian Buganza Tepole, William R. Cannon, Suvranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W. Lytton, Paris Perdikaris, Linda Petzold, and Ellen Kuhl. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *npj Digital Medicine*, 2(1):115, 2019. ISSN 2398-6352. doi: 10.1038/s41746-019-0193-y. URL https://doi.org/10.1038/s41746-019-0193-y.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020. URL http://jmlr.org/papers/v21/19-820.html.

Ruth E. Baker, Jose-Maria Peña, Jayaratnam Jayamohan, and Antoine Jérusalem. Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biology Letters*, 14:20170660, 05 2018. doi: 10.1098/rsbl.2017.0660.

Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116:15344–15349, 07 2019. doi: 10.1073/pnas.1814058116.

Katharina Baum, Johannes Schuchhardt, Jana Wolf, and Dorothea Busse. Of gene expression and cell division time: A mathematical framework for advanced differential gene expression and data analysis. *Cell Systems*, 9(6):569–579.e7, 2019. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2019.07.009. URL http://www.sciencedirect.com/science/article/pii/S2405471219302662.

Zak Costello and Hector Garcia Martin. A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data. *npj Systems Biology and Applications*, 4, 05 2018. doi: 10.1038/s41540-018-0054-3.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63:139–144, 10 2020. doi: 10.1145/3422622.

Uzi Harush and Baruch Barzel. Dynamic patterns of information flow in complex networks. *Nature Communications*, 8(1):2181, 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-01916-3. URL https://doi.org/10.1038/s41467-017-01916-3.

Helge Hass, Carolin Loos, Elba Raimúndez-Álvarez, Jens Timmer, Jan Hasenauer, and Clemens Kreutz. Benchmark problems for dynamic modeling of intracellular processes. *Bioinformatics*, 35(17):3073–3082, 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz020. URL https://doi.org/10.1093/bioinformatics/btz020.

M. Herty, A. Klar, and A. K. Singh. An ode traffic network model. *Journal of Computational and Applied Mathematics*, 203(2):419–436, 2007. ISSN 0377-0427. doi: https://doi.org/10.1016/j.cam.2006.04.007. URL https://www.sciencedirect.com/science/article/pii/S0377042706002159.

A C Hindmarsh. Odepack, a systemized collection of ode solvers. *Sci. Comput. World*, 1983.

Moritz Hoffmann, Martin Scherer, Tim Hempel, Andreas Mardt, Brian de Silva, Brooke E Husic, Stefan Klus, Hao Wu, Nathan Kutz, Steven L Brunton, and Frank Noé. Deeptime: a python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*, 3:015009, 12 2021. doi: 10.1088/2632-2153/ac3de0.

C. Y. Huang and J. E. Ferrell. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proceedings of the National Academy of Sciences*, 93:10078–10083, 09 1996. doi: 10.1073/pnas.93.19.10078. URL https://www.pnas.org/doi/epdf/10.1073/pnas.93.19.10078.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 06 2021. doi: 10.1038/s42254-021-00314-5. URL https://www.nature.com/articles/s42254-021-00314-5.

W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 115:700–721, 08 1927. doi: 10.1098/rspa.1927.0118. URL https://doi.org/10.1098/rspa.1927.0118.

Karl Otness, Arvi Gjoka, Joan Bruna, Daniele Panozzo, Benjamin Peherstorfer, Teseo Schneider, and Denis Zorin. An extensible benchmark suite for learning to simulate physical systems. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=pY9MHwmrymR.

Linda Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM Journal on Scientific and Statistical Computing*, 4(1):136–148, 1983. doi: 10.1137/0904010. URL https://doi.org/10.1137/0904010.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 02 2019. doi: 10.1016/j.jcp.2018.10.045.

Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020. doi: 10.1126/science.aaw4741. URL https://www.science.org/doi/abs/10.1126/science.aaw4741.

Robert Koch-Institut. Sars-cov-2 infektionen in deutschland. 08 2022. doi: 10.5281/zenodo.6994808. URL https://doi.org/10.5281/zenodo.6994808.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 2256–2265. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/sohl-dickstein15.html.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=dh_MkX0QfrK.

Z. Wan, Y. Zhang, and H. He. Variational autoencoder based synthetic data generation for imbalanced learning, 11 2017. URL https://ieeexplore.ieee.org/abstract/document/8285168.

A APPENDIX

A.1 EXPERIMENTAL SETUP

We describe two example use cases that leverage SimbaML's capabilities. The code for both experiments is available at `https://anonymous.4open.science/r/Simba_ML_Experiments-54F6`.

A.1.1 FORECASTING IN A COMPLEX BIOCHEMICAL PATHWAY: DATA NEEDS AND DECIDING FOR AN APPROPRIATE ML MODEL

For this use case (shown in Figure 1 B), we forecast time series for a selected readout of a biochemical signaling pathway. We use SimbaML to identify needs for data collection, i.e., which amount of data is needed for training an ML model for this prediction task, and which ML model to apply.

We assume the expected measurement data to consist of time series of five time steps of a single modeled quantity. We want the ML models to be capable of predicting the subsequent three time steps of that quantity. To infuse prior knowledge on the underlying system with SimbaML, we assume the dynamics to be governed by an ODE model of the underlying biological pathway, the MAPK signaling pathway (Huang & Ferrell, 1996). The ODE system represents the dynamics of 20 interacting species. The readout quantity is the relative amount of the total maximally activated signaling molecule (this is derived as a combination of two of the modeled species and an initial condition: free plus phosphatase-bound doubly phosphorylated MAPK by total MAPK). We restrict the kinetic parameters and initial conditions to relevant ranges and sample from them to generate between 10^2 and 10^5 distinct time series for ML model training. We use the ODE solver LSODA (Petzold, 1983; Hindmarsh, 1983) with SimbaML to simulate 20 time steps for each time series. We render data more realistic by adding lognormal noise with SimbaML to mimic measurement noise on the simulated time series.

We use SimbaML to train and evaluate the following ML models: Neural network, decision tree, linear model, nearest neighbor, random forest, and support vector machine. We use the normalized root mean squared error (RMSE) as the performance metric.

A.1.2 SIR-INFORMED TIME SERIES FORECASTING IN A SMALL DATA SETTING

We obtain daily reports on new cases of COVID-19 in Germany during the initial wave of the pandemic in early 2020 from Robert Koch-Institut (2022) and preprocess the data using a 7-day-moving average.

Using SimbaML, we generate 100 synthetic pandemic incidence time series with an SIR - ODE compartment model (Kermack & McKendrick, 1927). We initialize the susceptible compartment S with the estimated population of Germany N at the end of 2019 (83,166,711) subtracted by 100 initially infected individuals I, and zero recovered individuals. Within SimbaML we sample the transmission rate (β) and recovery rate (γ) parameters from continuous uniform distributions with support $(0.32, 0.35)$ and $(0.123, 0.125)$, respectively. We use SimbaML's additive Gaussian noise to mimic measurement errors synthetically. Alongside the standard SIR model, we simulate the cumulative new cases C_Σ with its derivative:

$$\frac{dC_\Sigma}{dt} = \frac{\beta SI}{N}$$

and calculate the finite difference to attain the new cases for each timepoint t:

$$C(t) = C_\Sigma(t) - C_\Sigma(t-1).$$

We train a neural network that predicts the parameters of a Student's t-distribution for the new case number using seven days as input and forecasting horizon, respectively. We use a feed-forward architecture as implemented in the Python library GluonTS (Alexandrov et al., 2020) with two layers of 20 neurons each. We define a training cut-off in the early stage of the pandemic, before the peak of the first wave, and compare the prediction of a model trained solely on the observed time series to a model trained additionally on all synthetically generated time series (see Figure 1 C).

A.2 RELATED WORK

We reviewed relevant publications that use or enable data generation with mechanistic modeling and time series prediction. We highlight differences from SimbaML in Table 1.

Table 1: Comparison of SimbaML to related work in five areas. First, on the data generation capabilities, adding noise and sparsification of the generated data for realistic real-world data. Second, the supported prediction tasks of time series as well as steady states. Third, on the use cases, analyzing for both benchmark and transfer learning capabilities. Fourth, on the software aspect of extendability, the scope of the use-case, and software licensing. Fifth, we analyzed the functionalities not focused on limited data scenarios, but that are likewise related to the combination of machine learning with data generated from mechanical models: learning of ODEs as functions from data and solving of computationally expensive mechanistic models. 1: Takamoto et al. (2022) (PDEBench), 2: Hoffmann et al. (2021) (Deeptime), 3: Otness et al. (2021), 4: Costello & Martin (2018), 5: Raissi et al. (2020) (HFM), 6: Bar-Sinai et al. (2019), 7: Raissi et al. (2019)

	SimbaML	**Frameworks**			**Use Cases**		**Related**	
Related Work		1	2	3	4	5	6	7
Generate Data	x	x	x	x	x	x	x	x
Noise and Sparsification	x							
Predict Time Series	x	x	x	x	x	x	x	x
Predict Steady State	x	x						
Benchmarking of ML	x	x	x	x	x			x
Transfer Learning	x				x	x		
Extendable	x		x	x				
Generic Use-Case	x	x	x	x				
Open Source	x	x	x	x		x	x	x
Learning ODE/PDE							x	x
Solving ODE/PDE With ML				x		x		x