The Journal of Open Source Software

kooplearn: A Scikit-Learn Compatible Library of Algorithms for Evolution Operator Learning

Giacomo Turri [1], **Grégoire Pacreau** [2], **Giacomo Meanti** [3], **Timothée Devergne** [4,1], **Daniel Ordoñez** [1], **Erfan Mirzaei** [1], **Bruno Belucci** [5], **Karim Lounici** [2], **Vladimir Kostic** [1,6], **Massimiliano Pontil** [1,7], **and Pietro Novelli** [1]

1 Computational Statistics and Machine Learning, Italian Institute of Technology, Genoa, Italy **2** Centre de Mathématiques Appliquées, École Polytechnique, Palaiseau, France **3** Centre Inria de l'Université Grenoble Alpes, Montbonnot, France **4** Atomistic Simulations, Italian Institute of Technology, Genoa, Italy **5** Paris Dauphine University, Paris, France **6** Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Novi Sad, Serbia **7** AI Centre, Department of Computer Science, University College London, London, UK

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 13 January 2026
Published: unpublished

Summary

`kooplearn` is a machine learning library that implements linear, kernel, and deep learning estimators of *dynamical operators* and their spectral decompositions. `kooplearn` can model both discrete-time evolution operators (Koopman/Transfer) and continuous-time infinitesimal generators. By learning these operators, users can analyze dynamical systems via spectral methods, derive data-driven reduced-order models, and forecast future states and observables. `kooplearn`'s interface is compliant with the `scikit-learn` API (Pedregosa et al., 2011), facilitating its integration into existing machine learning and data science workflows. Additionally, `kooplearn` includes curated benchmark datasets to support experimentation, reproducibility, and the fair comparison of learning algorithms. The software is available at https://github.com/Machine-Learning-Dynamical-Systems/kooplearn.

Statement of Need

From fluid flows down to atomistic motions, dynamical systems permeate every scientific discipline. Among the data-driven frameworks for modeling dynamical systems, evolution operator learning (Kostic et al., 2022) is both general and principled, and is especially well suited for interpretability (Mezić, 2005; Schütte et al., 2001) and dimensionality reduction (Klus et al., 2018). An evolution operator E characterizes dynamical systems, either stochastic $x_{t+1} \sim p(\cdot|x_t)$, or deterministic $x_{t+1} \sim \delta(\cdot - F(x_t))$, as follows: for every function f of the state of the system, $(\mathsf{E}f)(x_t)$ is the expected value of f one step ahead in the future, given that at time t the system was found in x_t

$$(\mathsf{E}f)(x_t) = \int p(dy|x_t) f(y) = \mathbb{E}_{y \sim X_{t+1}|X_t}[f(y)|x_t].$$

Notice that E is an operator because it maps any function f to another function, $x_t \mapsto (\mathsf{E}f)(x_t)$, and is *linear* because $\mathsf{E}(f + \alpha g) = \mathsf{E}f + \alpha \mathsf{E}g$. When the dynamics is deterministic, E is known as the *Koopman operator* (Koopman, 1931), while in the stochastic case it is known as the *transfer operator* (Applebaum, 2009). Arguably, the most important feature of evolution operators is their spectral decomposition (Mezić, 2005), which can be used to express the dynamics as a linear superposition of *modes*. These ideas lie at the core of the celebrated

Time-lagged Independent Component Analysis (Molgedey & Schuster, 1994), and Dynamical Mode Decomposition (DMD) (Kutz et al., 2016; Schimid, 2010).

Evolution operator learning is best understood from the perspective of *latent linear dynamical models*, which is schematically depicted in Figure 1. In this framework, the dynamical state x_t is first mapped into a latent space defined by a (fixed or learned) representation φ. Then, a *linear evolution* map E is learned to approximate the dynamics of the latents. The pair (φ, E) provides an approximation of E restricted to the d-dimensional subspace spanned by the components of φ, given the data. `kooplearn` implements state-of-the-art methods to learn φ, E, and the associated spectral decomposition of E.

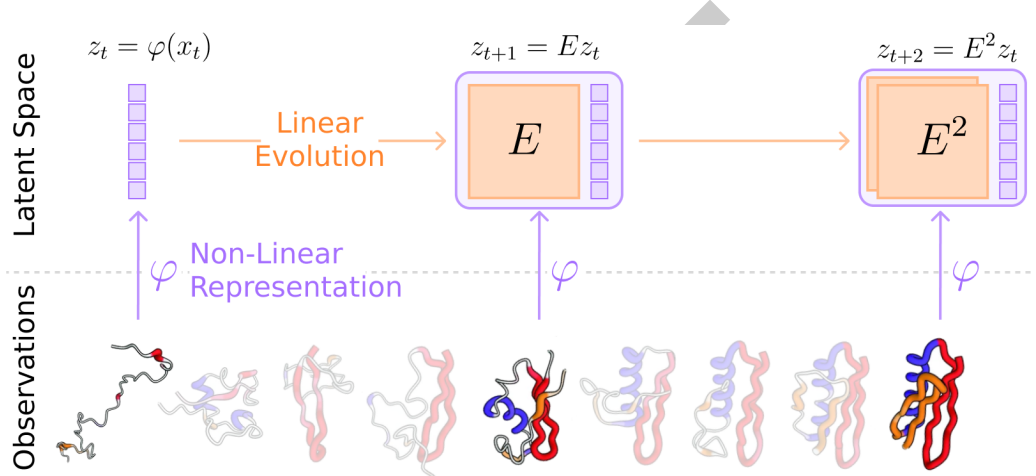

Figure 1: Sketch of the action of an evolution operator on a protein folding trajectory. The dynamics of the protein is linearized by means of a nonlinear representation φ and subsequently evolved by means of the linear map E.

The ecosystem of Python libraries that support operator-based modeling has grown considerably in recent years, with a predominant focus on the DMD family of methods. `PyDMD` (Ichinaga et al., 2024) emphasizes classical and kernel DMD variants; `pykoopman` (Pan et al., 2024) implements classical DMD methods with dictionary-based feature maps; `pykoop` (Dahdah & Forbes, 2025) offers a modular framework for lifting-function construction with a focus on system identification and control; `DLKoopman` (Dey & Davis, 2023) focuses on autoencoder approaches, while `KoopmanLab` (Xiong et al., 2023) targets Koopman neural operators. `kooplearn` addresses the general problem of learning evolution operators, and it is the result of a multi-year research effort in innovative operator learning algorithms. While it provides standard prediction and spectral decomposition utilities, it extends the state of the art in evolution operator learning codes by implementing fast kernel estimators (Meanti et al., 2023; Turri et al., 2023), infinitesimal generator models for SDEs (Devergne et al., 2024; Kostic, Halconruy, et al., 2024), and specialized losses for deep representation learning (Kostic, Novelli, et al., 2024; Kostic, Pacreau, et al., 2024; Mardt et al., 2018; Turri et al., 2025). We now provide a concise overview of the functionality of `kooplearn`.

Learning Linear Evolution Maps E

`kooplearn` implements state-of-the-art algorithms for learning evolution operators when the representation φ is fixed. The library offers estimators in both their linear and kernel formulations (see the `Ridge` and `KernelRidge` classes), which bridge the gap between recent theoretical advances (Kostic et al., 2022, 2023; Kostic, Lounici, et al., 2024; Kostic, Novelli, et al., 2024) and practical code implementations. A key model in `kooplearn` is the kernel-based *Reduced Rank Regression* (Kostic et al., 2022). This estimator provably outperforms traditional methods (Williams et al., 2015) in approximating the operator's spectrum (Kostic et al.,

2023), as illustrated in Figure 2. To our knowledge, kooplearn provides the only open-source implementation of this algorithm. To handle large datasets, kooplearn also includes randomized (Turri et al., 2023) and Nyström-based (Meanti et al., 2023) kernel estimators, which significantly speed up the fitting process, making it one of the fastest libraries for kernel-based operator learning, as shown in Figure 3.

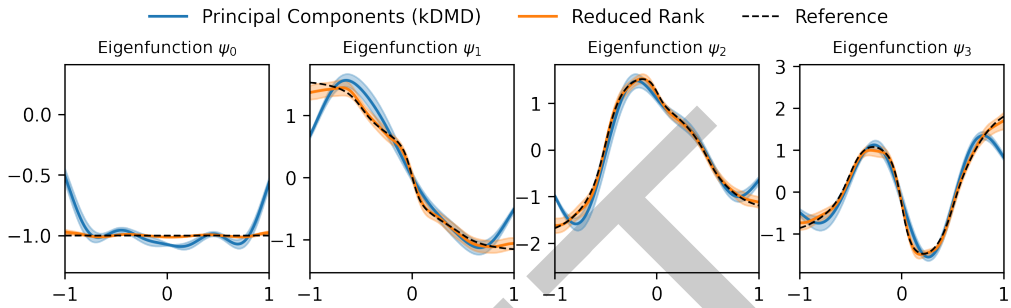

Figure 2: Comparison between kernel DMD (kDMD) and Reduced Rank estimators. The Reduced Rank estimator provides a more accurate approximation of the leading eigenfunctions of the transfer operator for the overdamped Langevin dynamics.

Figure 3: Fit time of a Kernel model (Gaussian kernel) on a dataset of 5000 observations from the Lorenz-63 dynamical system. The results are the median of three independent runs on a system equipped with an Intel Core i9-9900X CPU (3.50GHz) and 48GB of RAM memory.

Learning the Representation φ

kooplearn also exposes theoretically-grounded loss functions — implemented in both PyTorch (Paszke et al., 2019) and JAX (Bradbury et al., 2018) — suited for learning the representation φ with neural network models. This allows the incorporation of structural priors, such as graph-based encoders. Within this deep learning approach, two main families are supported: (i) *encoder-decoder* schemes with the loss proposed in Lusch et al. (2018), and (ii) *encoder-only* schemes, for which kooplearn implements the VAMP loss (Mardt et al., 2018) and the spectral contrastive loss (Turri et al., 2025).

Learning the Infinitesimal Generator of Diffusion Processes

In continuous-time dynamics, the system's evolution operator can be expressed as the exponential of the *infinitesimal generator* L, a differential operator defined by the equations of motion (Applebaum, 2009) (Chapter 3). Formally, for time-homogeneous dynamics, the generator relates to the evolution operator via $\mathrm{E} = e^{\mathrm{L}}$, and consequently $\mathbb{E}[f(X_t)|x_0] = (e^{t\mathrm{L}}f)(x_0)$. Since the exponential of an operator preserves its eigenfunctions, one can use knowledge of L (or its properties) to learn dynamical behavior without requiring lag-time data. In other words, it becomes possible to construct a physics-informed kinetic model E solely from static (equilibrium) data. To this end, kooplearn provides implementations of recent kernel-based

algorithms for diffusion processes with Dirichlet boundary conditions from (Kostic, Halconruy, et al., 2024), as well as neural representations as proposed in (Devergne et al., 2024). As demonstrated in (Devergne et al., 2025), these approaches improve sample complexity compared to estimators that rely solely on lag-time trajectory data.

Datasets

To foster reproducibility and rigorous benchmarking, `kooplearn` includes the `kooplearn.datasets` module, containing utilities to easily generate trajectories for systems that range from deterministic chaos (e.g., *Lorenz-63*, *Duffing oscillator*, *Logistic Map*) to stochastic and metastable dynamics (e.g., *stochastic linear systems*, *regime-switching models*, *Langevin dynamics*). A distinguishing feature of the library is the inclusion of benchmarks with accessible ground-truth spectral decompositions—such as the *Noisy Logistic Map* (Ostruszka et al., 2000) and *Overdamped Langevin Dynamics* in a quadruple-well potential (Prinz et al., 2011). These allow users to quantify the accuracy of learned eigenvalues and eigenfunctions directly (as demonstrated in Figure 2). Finally, the suite includes the *Ordered MNIST* from (Kostic et al., 2022) to evaluate performance on high-dimensional structured data. Examples of trajectories generated using the `kooplearn.datasets` module are illustrated in Figure 4.

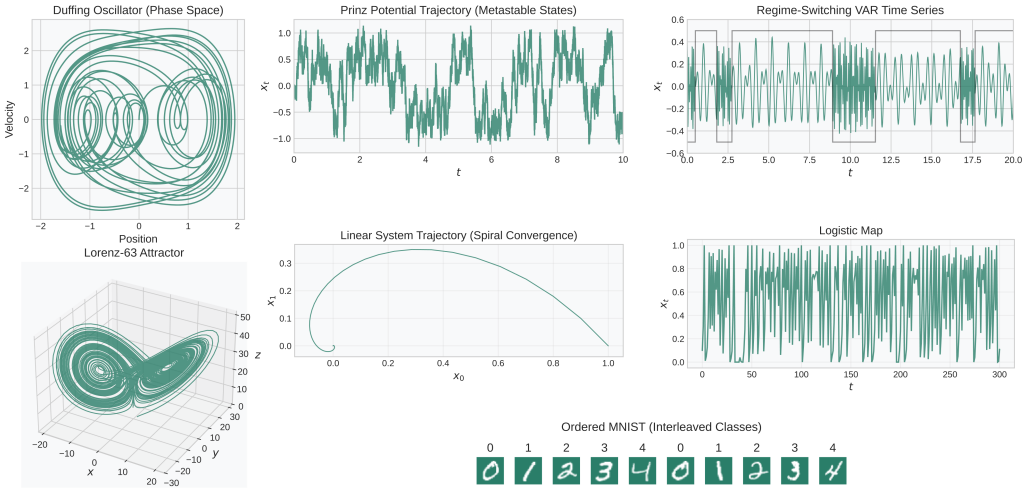

Figure 4: Samples from the datasets included in `kooplearn`.

Conclusion

`kooplearn` closely follows the `scikit-learn` API (Pedregosa et al., 2011) and strives to lower the technical barrier to experimenting with evolution operators. At the same time, it provides optimized implementations of state-of-the-art algorithms for evolution operator learning, making it valuable for research, education, rapid prototyping, and exploratory analysis of dynamical systems. As of today, `kooplearn` has been employed in a variety of studies (Bevanda et al., 2023, 2025; Kostic et al., 2022, 2023; Kostic, Lounici, et al., 2024; Kostic, Novelli, et al., 2024; Turri et al., 2025). It can be installed using the command `pip install kooplearn`. Its documentation, alongside many worked-out examples, is available on the webpage https://kooplearn.readthedocs.io/.

AI usage disclosure

Generative AI tools were used only for minor auxiliary tasks, such as code refactoring and formatting, assistance with documentation and unit tests, and proofreading for typographical

or grammatical errors in this manuscript. All scientific ideas, software design decisions, experiments, and interpretations were developed entirely by the authors, who verified all AI-generated content for accuracy.

Acknowledgements

This work is partially funded by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), through the PNRR MUR Project PE000013 CUP J53C22003010006 "Future Artificial Intelligence Research (FAIR)" and EU Project ELIAS under grant agreement No. 101120237.

References

Applebaum, D. (2009). *Lévy processes and stochastic calculus*. Cambridge University Press. https://doi.org/10.1017/cbo9780511809781

Bevanda, P., Beier, M., Lederer, A., Capone, A., Sosnowski, S., & Hirche, S. (2025). Koopman-equivariant gaussian processes. *arXiv Preprint arXiv:2502.06645*.

Bevanda, P., Beier, M., Lederer, A., Sosnowski, S., Hüllermeier, E., & Hirche, S. (2023). Koopman kernel regression. *Advances in Neural Information Processing Systems*, *36*, 16207–16221.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13). http://github.com/jax-ml/jax

Dahdah, S., & Forbes, J. R. (2025). Pykoop: A python library for koopman operator approximation. *Journal of Open Source Software*, *10*(114), 7947.

Devergne, T., Kostic, V. R., Parrinello, M., & Pontil, M. (2024). From biased to unbiased dynamics: An infinitesimal generator approach. *Advances in Neural Information Processing Systems*, *37*, 75495–75521.

Devergne, T., Kostic, V. R., Pontil, M., & Parrinello, M. (2025). Slow dynamical modes from static averages. *The Journal of Chemical Physics*, *162*(12).

Dey, S., & Davis, E. W. (2023). DLKoopman: A deep learning software package for koopman theory. *Learning for Dynamics and Control Conference*, 1467–1479.

Ichinaga, S. M., Andreuzzi, F., Demo, N., Tezzele, M., Lapo, K., Rozza, G., Brunton, S. L., & Kutz, J. N. (2024). PyDMD: A python package for robust dynamic mode decomposition. *Journal of Machine Learning Research*, *25*(417), 1–9.

Klus, S., Nüske, F., Koltai, P., Wu, H., Kevrekidis, I., Schütte, C., & Noé, F. (2018). Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, *28*(3), 985–1010.

Koopman, B. O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, *17*(5), 315–318.

Kostic, V. R., Halconruy, H., Devergne, T., Lounici, K., & Pontil, M. (2024). Learning the infinitesimal generator of stochastic diffusion processes. *Advances in Neural Information Processing Systems*, *37*, 137806–137846.

Kostic, V. R., Lounici, K., Inzerilli, P., Novelli, P., & Pontil, M. (2024). Consistent long-term forecasting of ergodic dynamical systems. *Forty-First International Conference on Machine Learning*.

Kostic, V. R., Lounici, K., Novelli, P., & Pontil, M. (2023). Sharp spectral rates for koopman operator learning. *Advances in Neural Information Processing Systems*, *36*, 32328–32339.

Kostic, V. R., Novelli, P., Grazzi, R., Lounici, K., & Pontil, M. (2024). Learning invariant representations of time-homogeneous stochastic dynamical systems. *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=twSnZwiOIm

Kostic, V. R., Novelli, P., Maurer, A., Ciliberto, C., Rosasco, L., & Pontil, M. (2022). Learning dynamical systems via koopman operator regression in reproducing kernel hilbert spaces. *Advances in Neural Information Processing Systems*, *35*, 4017–4031.

Kostic, V. R., Pacreau, G., Turri, G., Novelli, P., Lounici, K., & Pontil, M. (2024). Neural conditional probability for uncertainty quantification. *Advances in Neural Information Processing Systems*, *37*, 60999–61039.

Kutz, J. N., Brunton, S. L., Brunton, B. W., & Proctor, J. L. (2016). *Dynamic mode decomposition: Data-driven modeling of complex systems*. SIAM.

Lusch, B., Kutz, J. N., & Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, *9*(1). https://doi.org/10.1038/s41467-018-07210-0

Mardt, A., Pasquali, L., Wu, H., & Noé, F. (2018). VAMPnets for deep learning of molecular kinetics. *Nature Communications*, *9*(1). https://doi.org/10.1038/s41467-017-02388-1

Meanti, G., Chatalic, A., Kostic, V. R., Novelli, P., Pontil, M., & Rosasco, L. (2023). Estimating koopman operators with sketching to provably learn large scale dynamical systems. *Advances in Neural Information Processing Systems*, *36*, 77242–77276.

Mezić, I. (2005). Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, *41*(1), 309–325.

Molgedey, L., & Schuster, H. G. (1994). Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, *72*(23), 3634–3637. https://doi.org/10.1103/physrevlett.72.3634

Ostruszka, A., Pakoński, P., Słomczyński, W., & Życzkowski, K. (2000). Dynamical entropy for systems with stochastic perturbation. *Physical Review E*, *62*(2), 2018.

Pan, S., Kaiser, E., Silva, B. M. de, Kutz, J. N., & Brunton, S. L. (2024). PyKoopman: A python package for data-driven approximation of the koopman operator. *Journal of Open Source Software*, *9*(94), 5881. https://doi.org/10.21105/joss.05881

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & others. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Prinz, J.-H., Wu, H., Sarich, M., Keller, B., Senne, M., Held, M., Chodera, J. D., Schütte, C., & Noé, F. (2011). Markov models of molecular kinetics: Generation and validation. *The Journal of Chemical Physics*, *134*(17). https://doi.org/10.1063/1.3565032

Schimid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5–28. https://doi.org/10.1017/S0022112010001217

Schütte, C., Huisinga, W., & Deuflhard, P. (2001). *Transfer operator approach to conformational dynamics in biomolecular systems*. Springer.

Turri, G., Bonati, L., Zhu, K., Pontil, M., & Novelli, P. (2025). Self-supervised evolution

operator learning for high-dimensional dynamical systems. *arXiv Preprint arXiv:2505.18671*.

Turri, G., Kostic, V. R., Novelli, P., & Pontil, M. (2023). A randomized algorithm to solve reduced rank operator regression. *arXiv Preprint arXiv:2312.17348*.

Williams, M. O., Rowley, C. W., & Kevrekidis, I. G. (2015). A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics*, *2*(2), 247–265. https://doi.org/10.3934/jcd.2015005

Xiong, W., Ma, M., Huang, X., Zhang, Z., Sun, P., & Tian, Y. (2023). KoopmanLab: Machine learning for solving complex physics equations. *APL Machine Learning*, *1*(3).