Uncertainty Quantification for Reduced-Order Surrogate Models Applied to Cloud Microphysics

Jonas E. Katona
Department of Applied & Computational Mathematics
Yale University
New Haven, CT 06511
jonas.katona@yale.edu

Emily K. de Jong
Atmospheric, Earth, & Energy Division
Lawrence Livermore National Laboratory
Livermore, CA 94550
dejong5@llnl.gov

Nipun Gunawardena
Atmospheric, Earth, & Energy Division
Lawrence Livermore National Laboratory
Livermore, CA 94550
gunawardena1@llnl.gov

Abstract

Reduced-order models (ROMs) can efficiently simulate high-dimensional physical systems but lack robust uncertainty quantification methods. Existing approaches are frequently architecture- or training-specific, which limits flexibility and generalization. We introduce a post hoc, model-agnostic framework for predictive uncertainty quantification in latent-space ROMs that requires no modification to the underlying architecture or training procedure. Using conformal prediction, our approach estimates statistical prediction intervals for multiple components of the ROM pipeline: latent dynamics, reconstruction, and end-to-end predictions. We demonstrate the method on a latent-space dynamical model for cloud microphysics, where it accurately predicts the evolution of droplet-size distributions and quantifies uncertainty across the ROM pipeline.

1 Introduction

Latent-space reduced-order modeling learns a compact representation of high-dimensional physical dynamics in a lower-dimensional *latent space*. These models are valuable for scientific applications where the governing physics are partially known or computationally prohibitive to resolve. For example, accurately resolving clouds and precipitation in an atmospheric simulation would require tracking high-dimensional droplet-size distributions (DSDs), a longstanding parametric challenge in climate and weather modeling known as "cloud microphysics." Error and uncertainty in microphysics parameterizations is typically not quantified, but is believed to be a dominant source of uncertainty in future climate projections [26].

Latent-space ROMs have proven effective in efficiently simulating related complex fluid mechanical systems (e.g. [10, 13]), yet convincing practitioners of their reliability is challenging due to the lack of unified and robust uncertainty quantification (UQ) frameworks. Existing UQ methods for latent-space dynamical models are often tied to specific architectures [5, 9, 42], require expensive training [33], or make parametric assumptions [17, 21].

We present a model-agnostic, post hoc framework for predictive UQ in latent-space ROMs that quantifies uncertainty on *reconstruction*, *latent dynamics*, and *end-to-end predictions* without altering the base architecture or training procedure. Our approach utilizes conformal prediction (CP), a

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

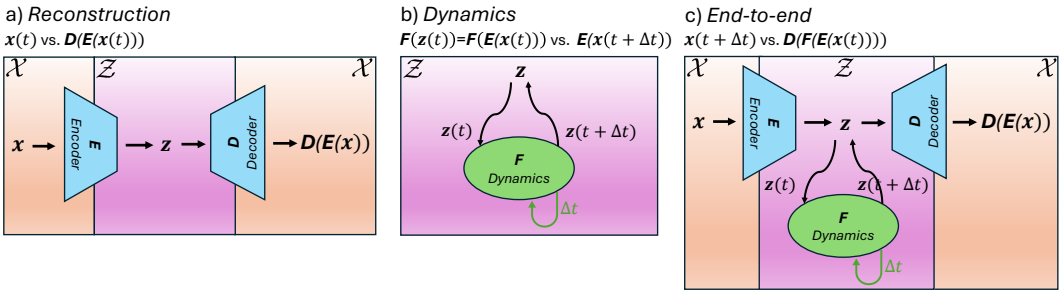

Figure 1: A generic latent-space dynamical model with fixed dynamical time-step Δt. Figures 1a and 1b show the reconstruction and dynamics sub-models, respectively, that comprise the end-to-end model architecture shown in Figure 1c.

distribution-free method that produces statistically valid prediction intervals—a first for latent-space ROMs. We demonstrate this UQ pipeline on a cloud microphysics ROM trained to predict the evolution of cloud DSDs during coalescence and the formation of precipitation [12], producing reliable UQ estimates that allow practitioners to rigorously evaluate individual components of the ROM architecture.

2 Proposed UQ framework

A latent-space dynamical ROM consists of a data space, $\mathcal{X} \subseteq \mathbb{R}^d$, and a latent space, $\mathcal{Z} \subseteq \mathbb{R}^m$, where $m \ll d$; an encoder $E : \mathcal{X} \to \mathcal{Z}$ and a decoder $D : \mathcal{Z} \to \mathcal{X}$; and a dynamical system model $F : \mathcal{T} \times \mathcal{Z} \to \mathcal{Z}$ defined on the latent space, where $\mathcal{T} \subseteq [0, T]$ for some final time $T > 0$. (See Figure 1.)

We consider the setting where we observe n independent time-evolving realizations of a physical system in $\mathcal{X}$: $\{x_t^{(j)}\}_{t\in\mathcal{T}}$ for $j = 1, \ldots, n$. The proposed pipeline applies regardless of whether model components are trained separately or end-to-end. Hence, at *each* fixed time $t \in \mathcal{T}$ and across samples $j = 1, \ldots, n$, we compute predictive uncertainties in the components of a latent-space dynamical model: 1) reconstruction from the latent space, 2) dynamics in the latent space, and 3) the entire pipeline combined (end-to-end), as shown in Figure 1.

Conformal predictions Conformal prediction (CP) provides statistical prediction intervals by computing nonconformity scores on a held-out calibration dataset such that the true outcome Y is contained in the prediction set $\Gamma(X)$ with probability at least $1 - \alpha$ [2, 3, 34, 38, 39], i.e., $\mathbb{P}(Y \subseteq \Gamma(X)) \geq 1 - \alpha$. This guarantee holds under the assumption of exchangeability of the calibration and test data, satisfied in the case of our DSD dataset due to independence of the sampled initial conditions—see Appendix C.1 for details. Because CP requires no changes to model architecture or changing or parametric assumptions on the data distribution, it can be applied either to full outputs or component-wise in multi-dimensional settings to obtain reliable, distribution-free guarantees on predictive coverage [2]. We illustrate three common variants of conformal predictions: **Vanilla conformal** (train–test split, using training data for scoring discrepancies), **split conformal** (train–validation–test split, scoring on validation set), and **CV+ conformal** (k-fold train–validation splits with aggregated residuals) [2, 3, 35]. CV+ generally yields tighter intervals while maintaining coverage guarantees, with the choice of folds k balancing statistical efficiency and computational overhead [3].

For DSD-valued predictions, we construct two-sided conformal prediction intervals using the $\alpha/2$ and $1 - \alpha/2$ empirical quantiles of the *signed* residuals [3]. This approach permits asymmetric upper and lower bounds when the residual distribution is skewed—a crucial feature for DSDs, as these are non-negative and often attain values near zero, allowing prediction intervals to reflect physical uncertainty better. Meanwhile, for latent-space predictions, whose outputs are multivariate and correlated, we instead use a scalar conformity score based on the Mahalanobis distance between predicted and true latent variables, providing a covariance-aware measure of predictive error. In either case, conformal sets are computed at each timestep separately, so that uncertainty can be tracked in time alongside latent-space dynamics and DSD evolution. See Appendix B for further details.

3 Application to Cloud Microphysics ROM

We demonstrate the UQ pipeline with a cloud microphysics ROM, trained on DSDs from particle-based simulations of warm-rain coalescence, and developed in detail in [12]. Coalescence is the process of small cloud droplets colliding and merging to form larger ones, eventually leading to large rain drops that precipitate [19]. As shown in Figure 2, droplet mass shifts from smaller to larger size bins under coalescence, and the strong nonlinearity of the process drives the emergence and disappearance of multiple modes.

New parameterizations of cloud microphysics must not only outperform traditional schemes in accuracy and efficiency, but also quantify structural errors and parametric uncertainties that currently hinder the accuracy of large-scale models. Nonlinear ROMs offer a more flexible alternative to bulk parameterizations, which impose restrictive modeling assumptions [18, 20, 30], and to linear latent-space ROMs, which could require inefficiently high-dimensional representations to capture DSD coalescence dynamics accurately [24, 29, 40]. Hence, to model droplet coalescence, we combine an autoencoder (AE) for nonlinear dimensionality reduction with parsimonious latent-space ODEs, based on the Sparse Identification of Nonlinear Dynamics (SINDy) technique [6, 8, 12].

For details on the AE–SINDy architecture, training, and accuracy, see Appendix C, the associated code repository, and the companion paper by De Jong et al. [12], which develops, trains, and evaluates the AE-SINDy surrogate in depth. The companion paper also applies the UQ framework introduced here to two additional latent-space ROMs—each using a similar autoencoder architecture but different dynamical models in the latent space—thereby highlighting broader context and applicability. Here, we focus specifically on the UQ pipeline and its expanded application to the AE–SINDy surrogate.

3.1 Dataset from LES with superdroplet method

We represent each DSD as a mass-density function $\frac{dm}{d\ln r}$, where r is the droplet radius, so that the DSD is defined per logarithmic radius interval. To train and evaluate the AE-SINDy surrogate, we use PSD trajectories generated from large eddy simulations (LES) of warm-rain coalescence with a high-fidelity Lagrangian particle representation of cloud droplets [12, 32, 36]. DSDs are averaged over a $(200\text{m})^3$ cubic domain, filtered for the presence of cloud condensate, and discretized into $N_{\text{bins}} = 64$ bins uniformly spaced in $\ln r$. For each simulation grid cell, a 600s coalescence-only forward integration produces the evolution of its PSD from $t = 0$ to $t = 600$s at $\Delta t = 10$s intervals. These trajectories provide both the instantaneous DSDs and the associated time derivatives used for training the AE-SINDy model and for assessing predictive uncertainty across different initial conditions.

3.2 Computational efficiency and UQ

The LES with Lagrangian microphysics used to generate the dataset are computationally intensive, using 3×10^6 grid cells, each with 128 particles for a total of $\sim 10^8$ Lagrangian particles per simulation. Collisional-coalescence is computed via linear stochastic pairwise sampling within each grid cell [32], which scales linearly with the number of particles. Combined with Lagrangian advection and Eulerian-Lagrangian coupling, these simulations tend to require hundreds of CPU-hours for a single multi-hour LES. Traditional Eulerian binned microphysics can be far less expensive, but still evolves 30–100 prognostic DSD bins per cell with quadratically scaling computations for collisional coalescence [19].

In contrast, the AE-SINDy surrogate used here compresses each 64-bin PSD into a 4-dimensional latent state governed by an ODE system whose evaluation is $\mathcal{O}(1)$ per grid cell per timestep. This yields reductions of several orders of magnitude in computational cost relative to SDM and at least an order of magnitude reduction relative to bin microphysics.

These reductions motivate the need for rigorous UQ. The UQ pipeline introduced here identifies when the surrogate faithfully reproduces the high-fidelity Lagrangian microphysics and when surrogate uncertainty becomes dominant. Furthermore, this pipeline allows us to measure how both structural uncertainty in the AE-based compression of DSDs and parametric uncertainty in the SINDy-identified latent dynamics propagate through DSD coalescence predictions. The examples in Figure 2 illustrate prediction intervals for three representative DSDs at fixed times, while Figure 3 summarizes the

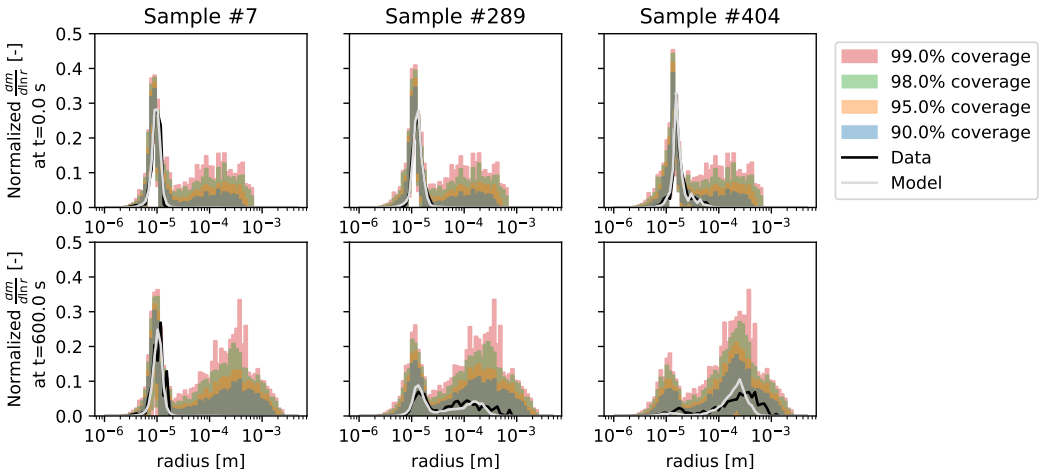

Figure 2: Initial and final states for three sample DSD trajectories from the dataset, predicted using the trained AE-SINDy architecture ("Model") with empirical prediction intervals provided via CV+ conformal predictions (using $k = 20$ folds) at varying nominal coverage levels. These predictions are also compared with the actual DSD final states from the dataset ("Data").

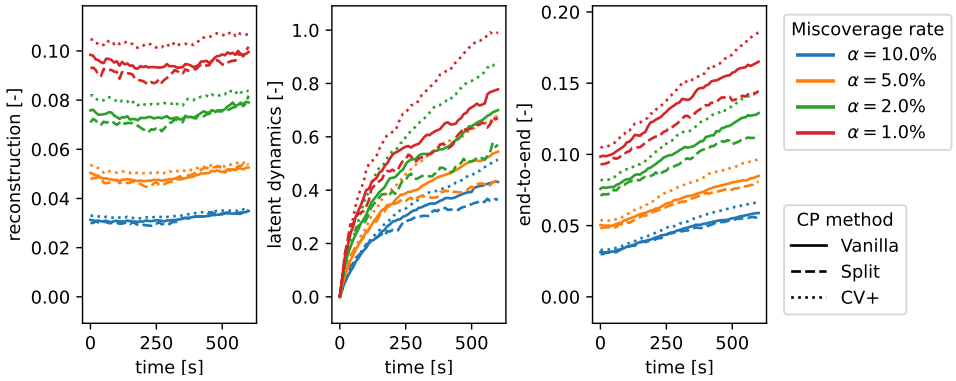

Figure 3: Average prediction interval width vs. time, computed by applying vanilla CP, split CP (using a 60-20-20 train-validation-test split), and CV+ conformal predictions (using $k = 20$ folds), respectively, to the indicated components of the AE-SINDy pipeline at varying nominal coverage levels $1 - \alpha$. Interval widths are computed using the mass-normalized integral of the area between prediction bands across bins for DSD-valued predictions and the volume of the Mahalanobis distance-derived prediction ellipsoids for latent-space predictions.

uncertainty propagated across different components of the ROM as a function of time and nominal miscoverage level α.

In Figure 3, the y-values report an average prediction interval width for each indicated ROM component. For DSD-valued predictions, this is computed as a normalized, total mass-weighted integral of prediction interval widths across bins, while for latent-space predictions, it is computed via the Mahalanobis distance-derived prediction ellipsoids in latent space, whose geometry is normalized by the residual covariance. Our analysis highlights the ability of the presented UQ pipeline to characterize data-driven ROMs by identifying the specific scales and processes where model improvements are most necessary.

4 Results & Discussion

Figure 2 shows how predictive uncertainty, as estimated on the testing data using CP, evolves across droplet-size bins during warm-rain coalescence. While end-to-end predictive uncertainty increases with time (cf. Figure 3), the uncertainty systematically shifts from smaller to larger droplet sizes: the

prediction interval "peak" at the sub-50μm cloud-droplet scale tends to contract while the prediction intervals expand markedly at larger rain-droplet scales as coalescence proceeds. This trend even holds for unimodal cloud droplet populations with negligible collisional growth (e.g., sample 7)—nevertheless, uncertainty in the larger rain bins grows with time. Physically, this reflects the inherent difficulty of predicting the onset of rain formation (i.e., the emergence of a secondary right-hand peak), compared to the more stationary evolution of smaller cloud droplets. The result underscores both the interpretability of uncertainty estimates in this framework and a key limitation of this particular ROM: intervals remain widest where precipitation processes begin, highlighting a persistent challenge in modeling warm-rain initiation [26].

Figure 3 illustrates not only predictive accuracy on unseen data but also qualitative patterns of uncertainty propagation across different components of the AE-SINDy architecture. As anticipated, reconstruction uncertainty intervals, which characterize the autoencoder alone, are consistent across time. The latent dynamics exhibit rapid incipient growth in uncertainty that slows over time, reflecting the challenging cloud-to-rain transition before droplets settle into a rain-dominant coalesced state. By contrast, the full ROM produces nearly linear growth in the end-to-end predictive errors. This highlights a key advantage of component-wise uncertainty analysis in ROMs: we detected how latent errors are effectively "smoothed out" by the autoencoder, yielding linear error growth in the final predictions.

Although a dynamical system may evolve on a lower-dimensional manifold, an inaccurate mapping between physical and latent space hinders a faithful, parsimonious latent representation, complicating the modeling of latent dynamics— particularly in the context of SINDy [6, 8]. Figure 3 shows that predicted reconstruction errors remain consistent over time for most of the data, regardless of α or the CP method. Even though predicted errors in the latent dynamics increasingly affect the end-to-end model output accuracy as time progresses, the propagation of these dynamics errors is ultimately mediated by reconstruction back to physical space. Thus, for this cloud microphysics ROM, future work to reduce structural uncertainty in the autoencoder will likely have a greater impact on overall model performance than refining the dynamical model.

While all conformal prediction methods achieved near-nominal coverage (see Appendix A), CV+ yielded wider average intervals to produce more reliable coverage, but at a higher computational cost. Although this can be done in parallel, CV+ requires retraining the surrogate model k times, compared to just once for vanilla or split conformal. The cost of training the surrogate model will therefore determine practical choices for CP techniques in future applications.

5 Limitations & Future Directions

The agreement across CP methods and variability in empirical coverages was notably better for the reconstruction and end-to-end network predictions than for latent dynamical predictions (cf. Figure 3). This is likely due to greater *variance* in prediction fidelity across times and variables—as well as a wider range of magnitudes overall—in latent predictions compared to normalized DSD predictions. Increasing training data—for instance, by altering the initial conditions or dynamical driver of the cloud LES (see Appendix C.1 or [12])—could reduce this variability, yielding more consistent CP intervals and improving coverage accuracy on the test set.

While this work demonstrates the flexibility of conformal prediction for uncertainty quantification in general black-box architectures, a key limitation of standard CP methods is that prediction intervals are scaled only *relative* to the input variables; the *width* of a given interval at a particular output and time remains fixed across the dataset. Although adaptive variants can adjust interval widths to reflect varying uncertainty [2, 4, 11, 15, 23, 28], we do not explore these extensions in this study.

That being said, the post hoc UQ approach introduced in this work is not limited to conformal prediction and could also extend to other interval- and set-valued UQ methods—e.g., parametric prediction intervals, confidence intervals, and Bayesian credible intervals [7, 14, 16, 25, 41]. Exploring these extensions, especially on other ROMs, could show the usefulness of this approach for quantifying other types of uncertainty in surrogate modeling pipelines.

Taken together with the companion study De Jong et al. [12], which develops the AE-SINDy surrogate itself, this work lays the foundation for a unified framework for both rigorous UQ of latent-space ROMs and its relevance in the efficient modeling of warm-rain microphysics.

Acknowledgments and Disclosure of Funding

This work was performed in part under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory (LLNL) under Contract DE-AC52-07NA27344 and supported by the Laboratory Directed Research and Development Program (LDRD), project number 25-ERD-045. J. Katona was partially supported by a stipend and teaching fellowship from the Yale Graduate School of Arts and Sciences. The authors have declared that none of them have any competing interests. Released under IM number LLNL-CONF-2010541.

The authors would like to thank LLNL—particularly Livermore Computing—for their support and for providing high-performance computing resources that enabled the simulations, data analyses, and figure generation in this study. The authors also appreciate the support and valuable advice of Peter Caldwell, Hassan Beydoun, Aaron Donahue, Chris Golaz, Youngsoo Choi, and many others in AEED and CASC at LLNL.

References

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330701. URL https://doi.org/10.1145/3292500.3330701.

[2] A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511, 2021. URL https://people.eecs.berkeley.edu/~angelopoulos/publications/downloads/gentle_intro_conformal_dfuq.pdf.

[3] R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021. ISSN 0090-5364, 2168-8966. doi: 10.1214/20-AOS1965. URL https://projecteuclid.org/journals/annals-of-statistics/volume-49/issue-1/Predictive-inference-with-the-jackknife/10.1214/20-AOS1965.full. Publisher: Institute of Mathematical Statistics.

[4] R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. The limits of distribution-free conditional predictive inference. *Proceedings of the National Academy of Sciences*, 119(34): e2204569119, 2022. doi: 10.1073/pnas.2204569119. URL https://www.pnas.org/doi/10.1073/pnas.2204569119.

[5] D. Bonnet, T. Hirtzlin, A. Majumdar, T. Dalgaty, E. Esmanhotto, V. Meli, N. Castellani, S. Martin, J.-F. Nodin, G. Bourgeois, J.-M. Portal, D. Querlioz, and E. Vianello. Bringing uncertainty quantification to the extreme-edge with memristor-based bayesian neural networks. *Nature Communications*, 14(1), Nov. 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-43317-9. URL http://dx.doi.org/10.1038/s41467-023-43317-9.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113. URL https://www.pnas.org/doi/10.1073/pnas.1517384113. Publisher: Proceedings of the National Academy of Sciences.

[7] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, Pacific Grove, CA, 2 edition, 2002. ISBN 9780534243128.

[8] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. doi: 10.1073/pnas.1906995116. URL https://www.pnas.org/doi/10.1073/pnas.1906995116.

[9] S. Cheng, D. Kong, J. Xie, K. Lee, Y. N. Wu, and Y. Yang. Latent space energy-based neural odes, 2025. URL https://arxiv.org/abs/2409.03845.

[10] Y. Choi, D. Coombs, and R. Anderson. SNS: A solution-based nonlinear subspace method for time-dependent model order reduction, 2020. URL http://arxiv.org/abs/1809.04064.

[11] N. Colombo, V. Vovk, R. Guo, J. Lei, and I. Nouretdinov. On training locally adaptive conformal predictors. In *Proceedings of the 40th International Conference on Machine Learning*, volume 204 of *Proceedings of Machine Learning Research*, pages 6235–6258, 2023. URL https://proceedings.mlr.press/v204/colombo23a/colombo23a.pdf.

[12] E. K. De Jong, N. Gunawardena, J. E. Katona, H. Beydoun, D. Ghosh, and P. M. Caldwell. Data-driven reduced ordering modeling for warm rain microphysics. *Authorea*, Nov. 2025. doi: 10.22541/au.176220214.46858428/v1. URL https://www.authorea.com/doi/full/10.22541/au.176220214.46858428/v1.

[13] W. D. Fries, X. He, and Y. Choi. LaSDI: Parametric latent space dynamics identification. *Computer Methods in Applied Mechanics and Engineering*, 399:115436, 2022. ISSN 0045-7825. doi: 10.1016/j.cma.2022.115436. URL https://www.sciencedirect.com/science/article/pii/S0045782522004807.

[14] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. CRC Press, Boca Raton, FL, 3 edition, 2013. ISBN 9781439840955. doi: 10.1201/b16018.

[15] L. Guan and J. Lei. Localized conformal prediction: A generalized framework. *Biometrika*, 110 (1):33–50, 2023. doi: 10.1093/biomet/asac042. URL https://academic.oup.com/biomet/article/110/1/33/6647831.

[16] R. J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 3 edition, 2021. Available online at https://otexts.com/fpp3/.

[17] V. Iakovlev, C. Yildiz, M. Heinonen, and H. Lähdesmäki. Latent neural odes with sparse bayesian multiple shooting, 2023. URL https://arxiv.org/abs/2210.03466.

[18] E. Kessler. On the distribution and continuity of water substance in atmospheric circulations. *On the Distribution and Continuity of Water Substance in Atmospheric Circulations*, pages 1–84, 1969. doi: 10.1007/978-1-935704-36-2_1.

[19] A. P. Khain, K. D. Beheng, A. Heymsfield, A. Korolev, S. O. Krichak, Z. Levin, M. Pinsky, V. Phillips, T. Prabhakaran, A. Teller, S. C. van den Heever, and J.-I. Yano. Representation of microphysical processes in cloud-resolving models: Spectral (bin) microphysics versus bulk parameterization. *Reviews of Geophysics*, 53(2):247–322, 2015. ISSN 1944-9208. doi: 10.1002/2014RG000468. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/2014RG000468. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/2014RG000468.

[20] M. Khairoutdinov and Y. Kogan. A new cloud physics parameterization in a large-eddy simulation model of marine stratocumulus. *Monthly Weather Review*, 128(1):229–243, 2000. ISSN 1520-0493, 0027-0644. doi: 10.1175/1520-0493(2000)128<0229:ANCPPI>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/128/1/1520-0493_2000_128_0229_ancppi_2.0.co_2.xml. Publisher: American Meteorological Society Section: Monthly Weather Review.

[21] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

[22] O. Ledoit and M. Wolf. Nonlinear shrinkage estimation of large-dimensional covariance matrices. *Annals of Statistics*, 40(2):1024–1060, 2012. doi: 10.1214/12-AOS989.

[23] J. Lei, A. Rinaldo, and L. Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018. doi: 10.1080/01621459.2017.1307116.

[24] Y. Lin and Z. Gao. An enhanced reduced-order model based on dynamic mode decomposition for advection-dominated problems. *J. Sci. Comput.*, 102(3), 2025. ISSN 0885-7474. doi: 10.1007/s10915-025-02820-5. URL https://doi.org/10.1007/s10915-025-02820-5.

[25] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. Wiley, Hoboken, NJ, 6 edition, 2014. ISBN 9781118539715. doi: 10.1002/9781118762885.

[26] H. Morrison, M. v. Lier-Walqui, A. M. Fridlind, W. W. Grabowski, J. Y. Harrington, C. Hoose, A. Korolev, M. R. Kumjian, J. A. Milbrandt, H. Pawlowska, D. J. Posselt, O. P. Prat, K. J. Reimel, S.-I. Shima, B. v. Diedenhoven, and L. Xue. Confronting the challenge of modeling cloud and precipitation microphysics. *Journal of Advances in Modeling Earth Systems*, 12(8), 2020. ISSN 1942-2466. doi: 10.1029/2019MS001689.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] Y. Romano, E. Patterson, and E. J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/hash/5103c3584b063c431bd1268e9b5e76fb-Abstract.html.

[29] F. Romor, G. Stabile, and G. Rozza. Non-linear manifold reduced-order models with convolutional autoencoders and reduced over-collocation method | journal of scientific computing. *Journal of Scientific Computing*, 94(74), 2023. doi: 10.1007/s10915-023-02128-2. URL https://link.springer.com/article/10.1007/s10915-023-02128-2.

[30] A. Seifert and K. D. Beheng. A double-moment parameterization for simulating autoconversion, accretion and selfcollection. *Atmospheric Research*, 59-60:265–281, 2001. ISSN 0169-8095. doi: 10.1016/S0169-8095(01)00126-0. URL https://www.sciencedirect.com/science/article/pii/S0169809501001260.

[31] R. J. Serfling. Chapter 2. In *Approximation Theorems of Mathematical Statistics*, pages 74–107. Wiley, 1980. doi: 10.1002/9780470316481.

[32] S. Shima, K. Kusano, A. Kawano, T. Sugiyama, and S. Kawahara. The super-droplet method for the numerical simulation of clouds and precipitation: a particle-based and probabilistic microphysics model coupled with a non-hydrostatic model. *Quarterly Journal of the Royal Meteorological Society*, 135(642):1307–1320, 2009. ISSN 1477-870X. doi: 10.1002/qj.441.

[33] T. Simpson, K. Vlachas, A. Garland, N. Dervilis, and E. Chatzi. VpROM: a novel variational autoencoder-boosted reduced order model for the treatment of parametric dependencies in nonlinear systems. *Sci. Rep.*, 14(1):6091, Mar. 2024. doi: 10.1038/s41598-024-56118-x.

[34] L. Steinberger and H. Leeb. Leave-one-out prediction intervals in linear regression models with many variables, 2016. URL http://arxiv.org/abs/1602.05801.

[35] S. Thirumuruganathan, S. Shetiya, N. Koudas, and G. Das. Prediction intervals for learned cardinality estimation: An experimental evaluation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 3051–3064, 2022. doi: 10.1109/ICDE53745.2022.00274.

[36] S. Unterstrasser, F. Hoffmann, and M. Lerch. Collisional growth in a particle-based cloud microphysical model: insights from column model simulations using LCM1D (v1.0). *Geoscientific Model Development*, 13(11):5119–5145, Oct. 2020. ISSN 1991-959X. doi: 10.5194/gmd-13-5119-2020. URL https://gmd.copernicus.org/articles/13/5119/2020/. Publisher: Copernicus GmbH.

[37] A. W. van der Vaart. Chapter 21. In *Asymptotic Statistics*, pages 304–315. Cambridge University Press, 1998. doi: 10.1017/CBO9780511802256.

[38] V. Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74(1): 9–28, 2015. ISSN 1573-7470. doi: 10.1007/s10472-013-9368-4. URL https://doi.org/10.1007/s10472-013-9368-4.

[39] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer International Publishing, 2022. ISBN 978-3-031-06648-1 978-3-031-06649-8. doi: 10.1007/978-3-031-06649-8. URL https://link.springer.com/10.1007/978-3-031-06649-8.

[40] Z. Y. Wan, L. Zepeda-Núñez, A. Boral, and F. Sha. Evolve smoothly, fit consistently: Learning smooth latent dynamics for advection-dominated systems, 2023. URL http://arxiv.org/abs/2301.10391.

[41] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, New York, 2004. ISBN 9780387402727. doi: 10.1007/978-0-387-21736-9.

[42] J. Y. Yong, R. Geelen, and J. Guilleminot. Learning latent space dynamics with model-form uncertainties: A stochastic reduced-order modeling approach. *Computer Methods in Applied Mechanics and Engineering*, 435:117638, 2025. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2024.117638. URL https://www.sciencedirect.com/science/article/pii/S0045782524008922.

A Validation of empirical prediction intervals

Tables 1 and 2 give summary statistics—means and standard deviations, and medians, respectively—for the nominal coverages in the outputs for different subsets of the network: autoencoder/reconstruction alone, latent dynamical model alone, and the entire end-to-end network predictions. These statistics are averaged over all output variables—64 bins for DSD data or 4 latent variables—and times—61 timesteps at 10-second intervals.

Table 1: Empirical coverage (in %, given as **empirical mean** $\pm$ **empirical standard deviation** across all times and output coordinates) for prediction intervals at marginal coverage levels 90%, 95%, 98%, and 99%. Split conformal was applied on a 60-20-20 train-validation-test split, and CV+ conformal was applied with $k = 20$ folds. Empirical medians appear in Table 2.

Sub-model	CP Method	$1-\alpha = 90\%$	$1-\alpha = 95\%$	$1-\alpha = 98\%$	$1-\alpha = 99\%$
Reconstruction	Vanilla	88.56 ± 3.16	93.86 ± 2.30	96.96 ± 1.65	98.00 ± 1.38
Reconstruction	Split	87.70 ± 4.31	92.87 ± 3.62	96.10 ± 2.57	97.34 ± 2.17
Reconstruction	CV+	89.04 ± 3.09	94.36 ± 2.21	97.34 ± 1.60	98.28 ± 1.35
Latent dynamics	Vanilla	89.38 ± 5.41	95.16 ± 3.36	97.88 ± 1.64	98.98 ± 1.04
Latent dynamics	Split	88.22 ± 4.33	94.30 ± 1.87	97.32 ± 1.11	98.44 ± 1.26
Latent dynamics	CV+	95.23 ± 3.45	98.36 ± 1.93	99.44 ± 1.19	99.83 ± 0.53
End-to-end	Vanilla	88.65 ± 3.35	93.73 ± 2.55	96.79 ± 1.96	97.94 ± 1.54
End-to-end	Split	87.36 ± 4.57	92.79 ± 3.55	96.25 ± 2.54	97.45 ± 1.96
End-to-end	CV+	90.56 ± 3.57	95.20 ± 2.43	97.64 ± 1.68	98.46 ± 1.37

Table 2: Empirical coverage (in %, given as the **empirical median** across all times and output coordinates) for prediction intervals at marginal coverage levels $1-\alpha$ =90%, 95%, 98%, and 99%. Split conformal was applied on a 60-20-20 train-validation-test split, and CV+ conformal was applied with $k = 20$ folds. Corresponding means and standard deviations appear in Table 1.

Sub-model	CP Method	$1-\alpha = 90\%$	$1-\alpha = 95\%$	$1-\alpha = 98\%$	$1-\alpha = 99\%$
Reconstruction	Vanilla	88.71	93.55	96.77	98.39
Reconstruction	Split	87.10	93.55	95.97	97.58
Reconstruction	CV+	89.52	94.35	97.58	98.39
Latent dynamics	Vanilla	91.13	95.97	97.58	99.19
Latent dynamics	Split	88.71	94.35	97.58	98.39
Latent dynamics	CV+	95.97	99.19	100.00	100.00
End-to-end	Vanilla	88.71	92.74	96.77	98.39
End-to-end	Split	87.10	92.74	96.77	97.58
End-to-end	CV+	91.13	95.16	97.58	99.19

Across all CP methods, the empirical coverage was generally close to nominal levels, indicating good calibration. While split conformal is theoretically more accurate than vanilla conformal [2, 23], for both reconstruction and end-to-end outputs, vanilla and split conformal performed similarly, with

mean coverages typically within 1%–2% of the target and relatively low variability and standard deviations typically under ∼4% (aside from when $1 - \alpha = 90\%$). By contrast, CV+ consistently achieved slightly higher accuracy, producing empirical coverages that were closer to the nominal rates in both mean and median, especially at higher confidence levels. This improvement was most apparent in the end-to-end model, where CV+ coverage levels tracked the nominal ones more tightly than vanilla or split conformal did.

The latent dynamical model displays somewhat different coverage behavior from what was observed for the reconstruction and end-to-end components. For vanilla and split CP, latent-space coverages are still near the nominal levels—typically within 1-2%—but with slightly larger variability across samples than for reconstructions and end-to-end outputs. In contrast, CV+ produces systematically conservative intervals for the latent dynamics, with mean coverages around 95% vs. 90% at the 90% nominal level and mean coverages exceeding 98% at the 95–99% nominal levels; in fact, the corresponding medians saturated at 100% for the two highest nominal coverages. These results suggest that CP methods measure reconstruction and end-to-end prediction uncertainties reasonably well, while for latent dynamics, the coverage tends to be more conservative, especially for CV+, which provides the most stable yet also most pessimistic intervals.

To test more specifically for consistency in the predictive intervals across different CP methods, we also refer the reader to Figure 3. For all three subsets of the network, the predictive errors become less consistent as $\alpha \to 0$. This is because the convergence of empirical quantiles to their true values depends strongly on the miscoverage rate α [31, 37]. In particular, for CP, predictive intervals at smaller α require larger calibration sets to stabilize because extreme quantiles converge more slowly, whereas more central quantiles yield more stable intervals with fewer samples [23, 39].

B Nonconformity scores

In the more basic forms of conformal prediction, uncertainty intervals are constructed symmetrically—predictive errors above and below are assumed to have the same distribution. Concretely, for a model $f : X \to Y$, if we denote the **signed residual**

$$R := y - f(x)$$

between the model prediction $f(x)$ for an input $x \in X$ and a true outcome $y \in Y$, then the *absolute* residuals $|R|$ are often used to calibrate a single quantile. Thus, the prediction interval is symmetric and can be written as

$$[f(x) - Q_{1-\alpha}(|R|), f(x) + Q_{1-\alpha}(|R|)],$$

where $Q_{1-\alpha}(|R|)$ is the $(1-\alpha)$-quantile of $|R|$ as computed empirically over the dataset.

The aforementioned construction is simple and guarantees the desired coverage over the calibration data, but it forces the lower and upper bounds to be equally wide. Hence, following the tailwise quantile approach first introduced in [3], we use tailwise (one-sided) quantiles for the **DSD-valued** predictions. For a desired miscoverage α, we split the miscoverage evenly between the tails on both sides, with $\alpha/2$ mass for each. We then compute the lower and upper quantiles for the *signed* residuals: $Q_{\alpha/2}(R)$ and $Q_{1-\alpha/2}(R)$. This defines the prediction intervals used in this study:

$$\left[f(x) + Q_{\alpha/2}(R), f(x) + Q_{1-\alpha/2}(R)\right],$$

which still ensures $1 - \alpha$ coverage over the validation set but allows the lower and upper margins to differ whenever the residual distribution is asymmetric.

For **latent-space predictions**, the outputs are multivariate and typically exhibit correlations across dimensions. Hence, in this case, coordinate-wise absolute or signed residuals may not accurately capture the joint error structure. Accordingly, for a predicted latent vector $\hat{z} = f(x)$ and true latent vector z, we define the residual $r = z - \hat{z}$ and use a scalar nonconformity score based on the (squared) **Mahalanobis distance**:

$$S(z, \hat{z}) = r^\top \Sigma_r^{-1} r,$$

where Σ_r is the empirical covariance matrix of latent-space residuals estimated from the calibration set (in practice, using a Ledoit–Wolf shrinkage estimator [22, 27]). In particular, the scalar score $S(z, \hat{z})$ is used to compute the empirical $(1 - \alpha)$-quantile on the calibration set, which then defines the size of the latent-space prediction ellipsoid. Using the *residual* covariance rather than the latent

covariance yields a conformity score that is properly normalized with respect to the model's error geometry and captures correlated uncertainty through multivariate prediction ellipsoids.

The prediction intervals and nonconformity scores described above are computed independently at each timestep, using instantaneous residuals and covariance estimates. Hence, the prediction sets evolve in time, therby reflecting the temporal evolution of model uncertainty.

C Reproducibility Details for AE-SINDy Model Training

This appendix provides the essential information required to reproduce the autoencoder-SINDy (AE-SINDy) model setup and training procedure used in this study. The description covers data acquisition and preprocessing, model architecture and hyperparameters, and the training workflow, including loss function specification. The source code and data are further included in a linked repository. Further details on the AE-SINDy architecture, the polynomial SINDy library, and the loss structure used during training are provided in the companion study De Jong et al. [12], which develops the surrogate model in full.

C.1 Data Source and Preprocessing

The AE-SINDy model is trained using binned particle size distribution (PSD) data generated from a large-eddy simulation (LES) employing the superdroplet method. We simulate the evolution of a warm liquid-phase cloud that forms from a Gaussian surface moisture and heat flux, growing in altitude before precipitating. The primary datasets are accessed in NetCDF format and contain variables for binned droplet mass distributions over time and space, with droplet coalescence active as the only enabled droplet dynamic. For model input, only samples with sufficient liquid water content (e.g., exceeding 10^{-5} kg/kg) are included. Each PSD is normalized by its total liquid mass to ensure scale invariance during encoding.

The dataset is partitioned into training (80%, 494 samples) and testing (20%, 124 samples) sets. For each sample, the normalized PSD and its time derivative (computed via finite differences) are paired with the corresponding total liquid mass. The input tensors are shaped as $(N_{\text{batch}}, N_{\text{bins}})$ for both the PSD and its time derivative, where $N_{\text{bins}} = 64$. The total mass is provided as an additional input feature that bypasses the encoder to become the final latent variable. Total mass is further rescaled during training and testing by the maximum value of total mass contained in the training dataset.

For a full description of the large eddy simulations used to generate the PSD training data—including the simulation setups, temporal sampling, grid resolution, and the physics assumed in the forward simulation—see De Jong et al. [12], which details the simulation design and post-processing used to produce the binned PSD datasets.

C.2 Model Architecture and Hyperparameters

The AE-SINDy architecture used here follows the structure introduced in [12], which employs four fully-connected encoder and decoder layers that halve (or double, respectively) the dimension at each layer, using ReLU activations in the hidden layers and a softmax output to ensure normalized reconstructions, and employs a SINDy model for the dynamics in the latent space. More explicitly, the model consists of three components:

1. **Encoder:** A feed-forward neural network (FFNN) with four fully connected layers that sequentially reduce the input dimension from N_{bins} to the latent dimension (excluding the total-mass variable), yielding $N_{\text{latent}} - 1 = 3$ latent variables. Hidden layers use ReLU activations, and the final layer maps to the latent space without a nonlinearity.
2. **Decoder:** A FFNN mirroring the encoder structure, with four fully connected layers expanding from $N_{\text{latent}} - 1$ back to N_{bins}. A softmax activation on the output layer enforces that reconstructed PSDs remain normalized.
3. **SINDy Dynamics Module:** A bias-free, single-layer neural network that implements the SINDy formulation by outputting a linear combination of latent-space time derivatives using a polynomial feature library of terms up to second order.

Key hyperparameters for the model are as follows:

- `latent_dim`: Number of latent variables (4 total: 3 from the PSD encoding, plus 1 for the total liquid mass)
- `poly_order`: Maximum polynomial order in SINDy library (2)
- `batch_size`: Training batch size (25)
- `learning_rate`: Initial learning rate for AdamW optimizer (e.g., 0.0042)
- `patience`: Early stopping patience (50 epochs)
- `weight_decay`: L2 regularization coefficient (10^{-3})
- `tol`: Numerical tolerance for loss calculations (10^{-8})
- `loss_weights`: Relative weights for loss terms, determined via Champion et al.'s recommended scaling (see code for details)

All network weights are initialized using Xavier or Kaiming normal initialization, with zero bias.

C.3 Training Procedure and Loss Function

Training procedures for the AE-SINDy architecture follow the approach detailed in [12] and the associated code repository, including early stopping, learning-rate scheduling, and a composite loss combining reconstruction, PSD-derivative, and latent-derivative terms.

In summary, training is performed using the AdamW optimizer with learning rate scheduling and early stopping based on validation loss. The model is trained for up to 1000 epochs, with the option to halt training if no improvement is observed over a specified patience interval.

The total loss function L is a weighted sum of three components:

$$L = L_{\text{recon}} + w_{dx} L_{dx} + w_{dz} L_{dz} \tag{1}$$

where:

- L_{recon} is the Kullback-Leibler divergence between the normalized input PSD and its reconstruction.
- L_{dx} is the mean squared error between the predicted and actual time derivative of the PSD, projected via the decoder.
- L_{dz} is the mean squared error between the predicted and actual time derivative in the latent space, as computed by the SINDy module.

Loss weights are chosen to balance the reconstruction and dynamics learning, following a scaling based on the relative magnitudes of the PSD and its time derivative in the training data as in Champion et al. [8]. Other parameters, including the batch size, initial learning rate, and a multiplicative factor of w_{dx}, were determined using hyperparameter optimization with Optuna [1].

C.4 Code Availability

All code used for data processing, model definition, and training is written in Python using PyTorch and is available at `https://github.com/jonaskat87/UQ_AE-SINDy`. The scripts include utilities for loading NetCDF datasets, constructing PyTorch DataLoaders, defining the AE-SINDy architecture, executing the training loop, and running and visualizing the uncertainty quantification pipelines.

Additional scripts for data processing, model specification, and training of the AE-SINDy surrogate are available in the code repository accompanying the companion paper De Jong et al. [12].