Forecasting N-Body Dynamics: A Comparative Study of Neural Ordinary Differential Equations and Universal Differential Equations

Suriya R S, Prathamesh Dinesh Joshi, Rajat Dandekar, Raj Dandekar, Sreedath Panat

Vizuara AI

Abstract

The n-body problem, fundamental to astrophysics, simulates the motion of n bodies acting under the effect of their own mutual gravitational interactions. Traditional machine learning models that are used for predicting and forecasting trajectories are often data-intensive "black box" models, which ignore the physical laws, thereby lacking interpretability. Whereas Scientific Machine Learning (Scientific ML) directly embeds the known physical laws into the machine learning framework. Through robust modelling in the Julia programming language, our method uses the Scientific ML frameworks: Neural ordinary differential equations (NODEs) and Universal differential equations (UDEs) to predict and forecast the system's dynamics. In addition, an essential component of our analysis involves determining the "forecasting breakdown point", which is the smallest possible amount of training data our models need to predict future, unseen data accurately. We employ synthetically created noisy data to simulate real-world observational limitations. Our findings indicate that the UDE model is much more data efficient, needing only 20% of data for a correct forecast, whereas the Neural ODE requires 90%.

1 Introduction

Scientific Machine Learning (Scientific ML) has emerged as a powerful paradigm where we shift our objective from just simulating a known physical model to discovering or correcting the governing equations directly from observational data. Scientific ML combines the expressive power of neural networks with the interpretability of differential equations. This approach has been successfully implemented in various scientific disciplines like fluid mechanics, circuit modelling, optics, gene expression, quantum circuits, and epidemiology Baker et al. [2019], Dandekar et al. [2020b,a], Abhijit Dandekar [2022], Ji et al. [2022], Bills et al. [2020], Lai et al. [2021], Nieves et al. [2024], Wang et al. [2023], Ramadhan [2024], Rackauckas et al., Aboelyazeed et al. [2023].

Primarily, the progress in Scientific ML is driven by the following two frameworks: **Neural Ordinary Differential Equations** (NeuralODEs) Chen et al. [2018], Dupont et al. [2019], Massaroli et al. [2020], Yan et al. [2019], which learns the entire system dynamics through Neural Networks from data, and **Universal Differential Equations** (UDEs)Rackauckas et al. [2020], Bolibar et al. [2023], Teshima et al. [2020], Bournez and Pouly [2020], which blends in the known physical laws with neural networks to learn only the unknown/unmodelled dynamics from data. While these frameworks are being used in astrophysics Gupta et al. [2022], Branca and Pallottini [2023], Origer and Izzo [2024], a thorough comparative analysis of their effectiveness in solving problems is yet to be determined. In this study, we try to understand the effectiveness and limitations of these two Scientific ML frameworks.

This study aims to answer several key questions: whether the UDE framework can be used to learn and recover the pairwise gravitational interaction term by replacing it with a neural network; how the predictive accuracy of NeuralODEs compares to that of UDEs when modelling the trajectories;

2nd Conference on Applied AI and Scientific Machine Learning (CASML 2025).

whether both NeuralODEs and UDEs can be used to forecast the system's trajectories in the long term; and finally, if UDEs, by incorporating known physics, offer superior performance in forecasting over purely data-driven NeuralODEs.

2 Methodology

The foundation of our study uses the classical Newtonian formulation of the gravitational N-body problem. For a system of n-bodies, the state evolution of the system is described by their positions $r_i(t) \in \mathbb{R}^3$ and velocities $v_i(t) \in \mathbb{R}^3$ of each body i. The state evolution form a system of ordinary differential equations (ODEs) given by, $\frac{dr_i}{dt} = v_i \qquad \frac{dv_i}{dt} = G\sum_{j\neq i} m_j \frac{r_j - r_i}{\|r_j - r_i\|^3}$. Where G is the gravitational constant and m_j is the mass of body j. This system of equations is used for generating ground truth data and as the structural prior for the UDE model.

Dataset generation A 3-body system (random initial condition) was simulated with Runge–Kutta by Tsitouras' numerical integrator in the Julia Programming language for no-noise dataset generation. For the purpose of this simulation, we set the gravitational constant $G = 1$ and the masses of the three bodies to $m_1 = m_2 = m_3 = 1$. The simulation was run for 7 seconds, the domain $t \in [0, 7]$ was discretised into 70 equally spaced time points. Additionally, 2 more datasets were generated from no-noise dataset to simulate real-world observational dataset: moderate noise (7% Gaussian std. dev.), and high noise (35% Gaussian std. dev.)

2.1 Neural Ordinary Differential Equation (NODE)

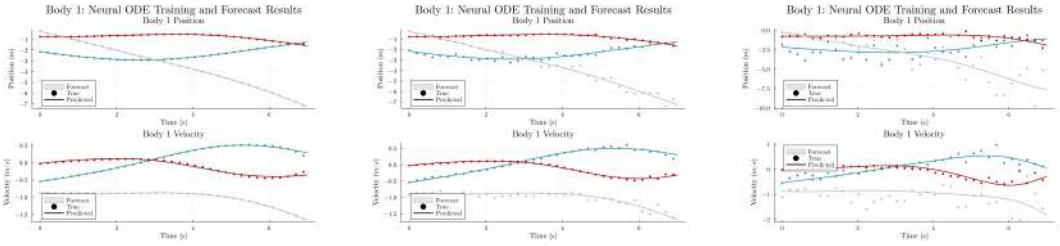
In this approach, we define the dynamics of the system's hidden state vector $h(t)$ with the help of an ordinary differential equation, $\frac{dh}{dt} = f(h(t), t, \theta)$, where the function describing the change is a neural network f parameterised by θ. Here, we perform backpropagation through the neural network augmented ODE. In doing so, we find the optimal values of the neural network parameters. In our application to the n-body problem, we consider the state vector $h(t) \in \mathbb{R}^{3n}$ which is just the concatenation of the position vectors and the velocity vectors of all the n-bodies $h(t) = [r_1, v_1, r_2, v_2, \ldots, r_n, v_n]$. Here, the Neural ODE framework replaces all the dynamics (Equations (**??**)) of the n-body system with a neural network.

2.2 Universal Differential Equations (UDEs)

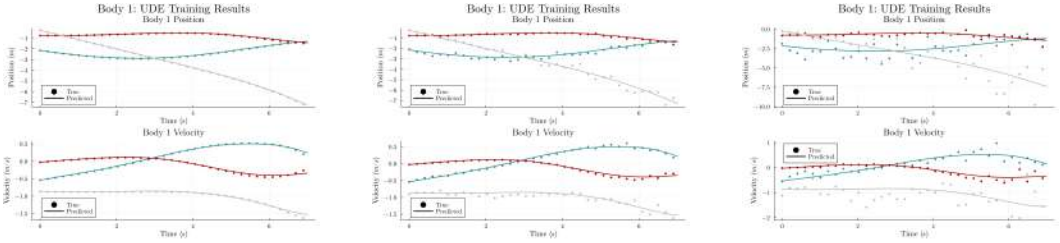
In the context of the n-body problem, a UDE is formed by retaining the knowledge that the total acceleration of a body is the sum of pairwise interactions with all other bodies. In addition, we replace the interaction term itself with a neural network. This allows the neural network to discover the underlying gravitational interaction between the objects from the data. The UDE is therefore defined as $\frac{dr_i}{dt} = v_i \qquad \frac{dv_i}{dt} = \sum_{j=1, j\neq i}^{n} \mathrm{NN}(r_i, r_j, m_i, m_j, \theta)$. Where θ denotes the parameters of the Neural Network (NN), the input to the network is the states (positions r_i, r_j, and masses m_i, m_j) of the two interacting bodies i and j. In this configuration, we assume the kinematic relationship and summation structure are already known physical laws.

3 Results

We have considered a total of 5 cases with different dataset percentages, evaluating each of them under the following three noise levels: no noise, moderate noise(7% standard deviation), and high noise (35% standard deviation). The main paper presents the results from training the Neural ODE and UDE models on the complete dataset (Case 1) and on the datasubsets of 80% (Case 3), 20% (Case 5).For brevity, only the results for body 1 are presented in the main text; the corresponding figures for body 2 and body 3 are available in the Appendix. Here, we also include the analysis of the forecasting breakdown point. Results for the remaining three cases, which utilized 90% (Case 2), and 40% (Case 4) of the data for training, are located in Appendix.



(a) NODE (No noise) (b) NODE (Moderate noise) (c) NODE (High noise)



(d) UDE (No noise) (e) UDE (Moderate noise) (f) UDE (High noise)

Figure 1: Combined results for Case 1 (100% training). Top row (a-c) shows Neural ODE results; bottom row (d-f) shows UDE results across different noise levels for body 1.

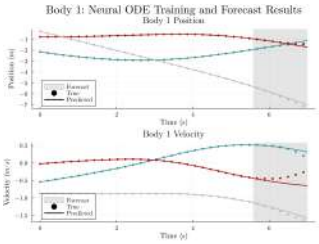
3.1 Case 1: Training on complete dataset

From Figure 1, it is evident that the Neural ODE effectively learns the n-body dynamics throughout the entire time span and across different noise conditions. When trained on noise-free data, its forecasts for the position and velocity of each body closely align with the actual trajectories. As noise is introduced, the model persists in generating smooth and physically plausible trajectories, successfully filtering out a majority of the random fluctuations inherent in the training data. Notably, this level of performance is maintained even as noise intensifies, with the model reliably producing accurate and stable trajectories. **These findings underscore the NeuralODE's considerable robustness to noise, ensuring long-term accuracy even in scenarios with significant data corruption.**

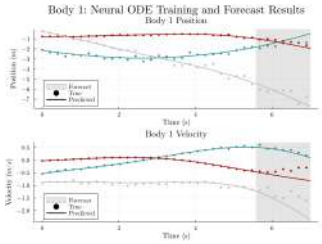
From Figure 1, the UDE, which is trained on the entire dataset, performs remarkably well in predicting the 3-body trajectories in all the noise levels. When the dataset is noise-free, throughout the duration of the simulation, the model's trajectory perfectly aligns with the ground truth data. Even under the moderate noise, the UDE produces clean and accurate predictions following the underlying dynamics. Even though under high noise, the data points have become significantly scattered, the model's predictions remain smooth and physically plausible. While minor deviations from the true path may appear, the overall shape and evolution of the trajectories are preserved. **These results demonstrate that when given access to the full dataset, the UDE is highly effective at learning the correct system dynamics and robust to substantial noise.**

3.2 Case 3: Training on 80% of the dataset and forecasting

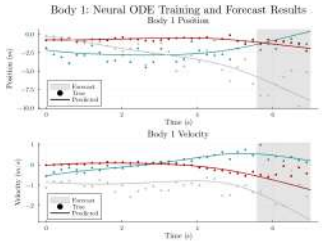
As depicted in Figure 2, wherein the Neural ODE is trained on 80% of the temporal domain and subsequently predicts the remaining 20%, a notable disparity in forecasting accuracy is observed between position and velocity. In the absence of noise, the predicted position trajectories align closely with the ground truth within both training and forecasting domains; conversely, the velocity predictions exhibit early signs of divergence. Under moderate noise conditions, the model continues to produce smooth and physically plausible position trajectories that adhere to the central trends of the noisy data; however, the accuracy of the velocity forecast deteriorates significantly, revealing considerable error. In scenarios with high noise levels, the training data becomes considerably more dispersed, and while the position forecasts from the Neural ODE maintain a degree of smoothness, the velocity forecasts completely fail. **Ultimately, these findings highlight that the NeuralODE's ability to generate reliable forecasts is compromised when it is trained using only 80% of the available data.**



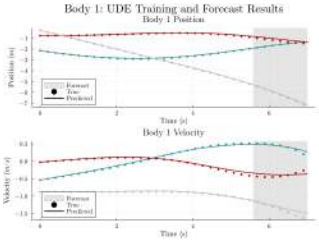
(a) NODE (No noise)



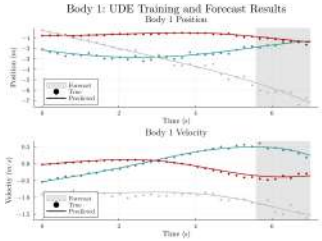
(b) NODE (Moderate noise)



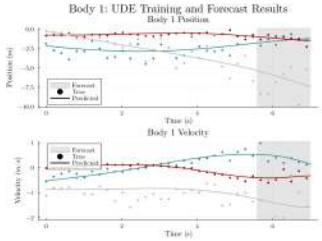
(c) NODE (High noise)



(d) UDE (No noise)



(e) UDE (Moderate noise)

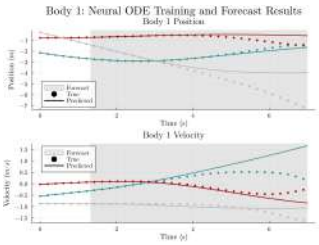


(f) UDE (High noise)

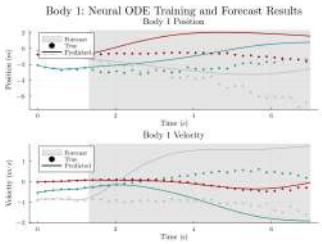
Figure 2: Combined results for Case 3 (80% training). Top row (a-c) shows Neural ODE results; bottom row (d-f) shows UDE results for body 1. The grey shaded region denotes the forecast domain.

In Figure 2, we have UDE trained on 80% of the dataset and forecasted on the rest. We can see it performs strongly across all noise levels. When the data is noise-free, its predictions for each body's trajectory stay very close to the actual path, both during training and the short forecast interval, indicating excellent generalization. Under moderate noise, where the training points are a bit dispersed, the UDEs prediction remains smooth and follows the underlying dynamics very well. Under high noise, the UDEs still produce a clean forecast that stays close to the true trajectory.**Threfore, these findings hihglight that UDE's ability to forecast is intact when trained with 80% of the dataset**

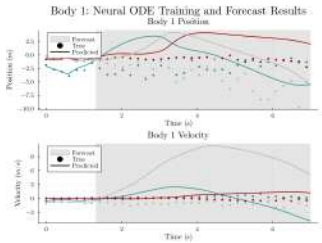
3.3 Case 5: Training on 20% of the dataset and forecasting



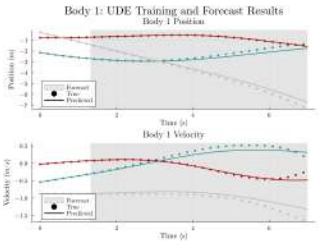
(a) NODE (No noise)



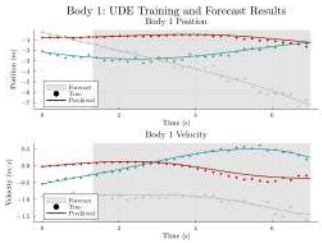
(b) NODE (Moderate noise)



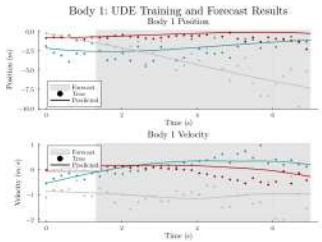
(c) NODE (High noise)



(d) UDE (No noise)



(e) UDE (Moderate noise)



(f) UDE (High noise)

Figure 3: Combined results for Case 5 (20% training). Top row (a-c) shows Neural ODE results; bottom row (d-f) shows UDE results for body 1. The grey shaded region denotes the forecast domain.

From Figure 3, when the Neural ODE is trained on only 20% of the domain, its ability to forecast the remaining 80% is significantly challenged. In the no-noise case, the predictions match the true paths well within the small training area, but the forecasted trajectories show clear and growing deviations over time. With moderate noise, the model captures the general trends within the limited training region; however, forecasting errors increase substantially, leading to predicted paths that diverge significantly from the ground truth. Under high noise, the model's forecasted trajectories lose coherence and show poor long-term predictive accuracy, failing to generalize from the sparse, noisy data. **These findings about NODE's inablility to forecast stays consistent with the result that we have got from case 3.**

In Figure 3, the UDE is trained on 20% of the dataset and forecasted on the rest. Under a noise-free dataset, the UDE's prediction for each body's trajectory follows the true path almost exactly, not only within the limited training region but also far into the extended forecast area, demonstrating robust generalization from a small data subset.Even With moderate & high noise, the model still produces a reliable trajectory. **These findings highlight UDE's ability to perform exceptionally well in data scrace and noisy environment.**

3.4 Forecasting Breakdown Point Analysis

To further explore the model's long-range forecasting capabilities, we progressively reduced the amount of training data to identify their forecasting breakdown points. This point is defined as the smallest percentage of training data below which the model fails to produce a physically plausible forecast of the unseen trajectory. For the no-noise dataset, the Neural ODE required at least 90% of the data, failing to forecast the future trajectory when trained on smaller subsets. In contrast, the UDE model demonstrated superior data efficiency, providing a reliable forecast even when trained with as little as 20% of the available data. However, the UDE also failed when the training data was reduced to just 10%. It is important to note that for the noisy datasets (moderate and high), UDE requried 40% of the dataset for reliable forecast, meanwhile, Neural ODE requried 95% of the dataset.

4 Discussion and Conclusion

The paper here offers a comparative analysis between Neural Ordinary Differential Equations (Neural ODEs) and Universal Differential Equations (UDEs) for forecasting trajectories related to the gravity n-body problem under various data and noise setting. Neural ODEs had a strong ability to understand the system's dynamics when trained on the complete dataset. However, their forecasting success was highly dependent on data availability with more than 90% dataset. Compared to this, UDEs were much more data-efficient, with low forecasting errors for models trained on as few as 20% of the dataset. This data-robustness underscores the benefit of hardcoding known physical laws—the form of gravitational interactions in this example—into models. With only unknown or unmodeled factors to learn, the UDE formulation provides a more trustworthy route to generalization with few-data sets.

That being said, a few of its limitations hold for both models as well. For the 7-second simulation window, this work's concern lies with showing proof-of-concept recovery and short-term forecasting accuracy, not long-horizon stability. Numerous UDE and Neural ODE benchmarks employ shorter time intervals initially to guarantee local dynamics accuracy before moving to larger trajectories. Furthermore, this study was conducted using a single set of initial conditions, and the models' ability to learn the dynamics from different set of initial conditions haven't been explored. Finally, the analysis is confined to a 3-body system, leaving its scalability to systems with more bodies as an open question.

These results are in agreement with the large-scale experience in the SciML community. For noisy, data-lean sitations, physics-informed models such as UDEs exhibit stronger generalizability compared to black-box models such as Neural ODEs. With the preserved structure of governing equations, UDEs exhibit increased interpretability, since learned neural components may be investigated for explaining discrepancies or unmodeled effects—an important benefit for scientific exploration. Future work will shift towards the long-term forecasting as well as generalizing the framework towards more complex gravity systems, for instance, with non-gravity forces or relativistic effects. The ultimate goal is to apply these Scientific ML models to real observational data, perhaps revealing new information about celestial mechanics and expanding beyond the reaches of our current physical models.

References

Raj Abhijit Dandekar. *A new way to do epidemic modeling*. PhD thesis, Massachusetts Institute of Technology, 2022.

Doaa Aboelyazeed, Chonggang Xu, Forrest M Hoffman, Jiangtao Liu, Alex W Jones, Chris Rackauckas, Kathryn Lawson, and Chaopeng Shen. A differentiable, physics-informed ecosystem modeling and learning framework for large-scale inverse problems: demonstration with photosynthesis simulations. *Biogeosciences*, 20(13):2671–2692, 2023.

Nathan Baker, Frank Alexander, Timo Bremer, Aric Hagberg, Yannis Kevrekidis, Habib Najm, Manish Parashar, Abani Patra, James Sethian, Stefan Wild, et al. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. Technical report, USDOE Office of Science (SC), Washington, DC (United States), 2019.

Alexander Bills, Shashank Sripad, William L Fredericks, Matthew Guttenberg, Devin Charles, Evan Frank, and Venkatasubramanian Viswanathan. Universal battery performance and degradation model for electric aircraft. *arXiv preprint arXiv:2008.01527*, 2020.

Jordi Bolibar, Facundo Sapienza, Fabien Maussion, Redouane Lguensat, Bert Wouters, and Fernando Pérez. Universal differential equations for glacier ice flow modelling. *Geoscientific Model Development Discussions*, 2023:1–26, 2023.

Olivier Bournez and Amaury Pouly. A universal ordinary differential equation. *Logical Methods in Computer Science*, 16, 2020.

Lorenzo Branca and Andrea Pallottini. Neural networks: solving the chemistry of the interstellar medium. *Monthly Notices of the Royal Astronomical Society*, 518(4):5718–5733, 2023.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Raj Dandekar, Shane G Henderson, Marijn Jansen, Sarat Moka, Yoni Nazarathy, Christopher Rackauckas, Peter G Taylor, and Aapeli Vuorinen. Safe blues: A method for estimation and control in the fight against covid-19. *medRxiv*, pages 2020–05, 2020a.

Raj Dandekar, Chris Rackauckas, and George Barbastathis. A machine learning-aided global diagnostic and comparative tool to assess effect of quarantine control in covid-19 spread. *Patterns*, 1(9), 2020b.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in neural information processing systems*, 32, 2019.

Raghav Gupta, PK Srijith, and Shantanu Desai. Galaxy morphology classification using neural ordinary differential equations. *Astronomy and Computing*, 38:100543, 2022.

Weiqi Ji, Franz Richter, Michael J Gollner, and Sili Deng. Autonomous kinetic modeling of biomass pyrolysis using chemical reaction neural networks. *Combustion and Flame*, 240:111992, 2022.

Zhilu Lai, Charilaos Mylonas, Satish Nagarajaiah, and Eleni Chatzi. Structural identification with physics-informed neural ordinary differential equations. *Journal of Sound and Vibration*, 508: 116196, 2021.

Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes. *Advances in neural information processing systems*, 33:3952–3963, 2020.

Emily Nieves, Raj Dandekar, and Chris Rackauckas. Uncertainty quantified discovery of chemical reaction systems via bayesian scientific machine learning. *Frontiers in Systems Biology*, 4:1338518, 2024.

Sebastien Origer and Dario Izzo. Closing the gap: Optimizing guidance and control networks through neural odes. *arXiv preprint arXiv:2404.16908*, 2024.

Chris Hill Rackauckas, Jean-Michel Campin, and Raffaele Ferrari. Capturing missing physics in climate model.

Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

Ali Ramadhan. *Data-driven ocean modeling using neural diferential equations*. PhD thesis, Massachusetts Institute of Technology, 2024.

Takeshi Teshima, Koichi Tojo, Masahiro Ikeda, Isao Ishikawa, and Kenta Oono. Universal approximation property of neural ordinary differential equations. *arXiv preprint arXiv:2012.02414*, 2020.

Allen M Wang, Darren T Garnier, and Cristina Rea. Hybridizing physics and neural odes for predicting plasma inductance dynamics in tokamak fusion reactors. *arXiv preprint arXiv:2310.20079*, 2023.

Hanshu Yan, Jiawei Du, Vincent YF Tan, and Jiashi Feng. On robustness of neural ordinary differential equations. *arXiv preprint arXiv:1910.05513*, 2019.

A Appendix

A Hyperparameter Details

This appendix provides the detailed hyperparameters used for training the Neural ODE and UDE models, corresponding to the initial no-noise, full-dataset training runs.

Table 1: Neural ODE range of hyperparameters on training data (no-noise).

Hyperparameter	**Values**	**Search Range**
Time Span	(0.0, 7.0)	(0, 7.0) - (0, 10.0)
Activation Function	tanh	ReLU, tanh, swish
Optimization Solver	Adam & BFGS	Adam, AdamW, BFGS
Learning Rate	Adam: 0.001	1e-4, 1e-3, 1e-2
Hidden Layers/Units	3 hidden layers, 64 units	16, 32, 64, 128 units
Number of Epochs	Adam: 200, BFGS: 200	100 - 1000

Table 2: UDE range of hyperparameters on training data (no-noise).

Hyperparameter	**Values**	**Search Range**
Time Span	(0.0, 7.0)	(0, 7.0) - (0, 10.0)
Activation Function	swish	ReLU, tanh, swish
Optimization Solver	AdamW	Adam, AdamW, BFGS
Learning Rate	0.001	1e-4, 1e-3, 1e-2
Hidden Layers/Units	1 hidden layer, 32 units	16, 32, 64, 128 units
Number of Epochs	700	100 - 1000

B Supplementary Material

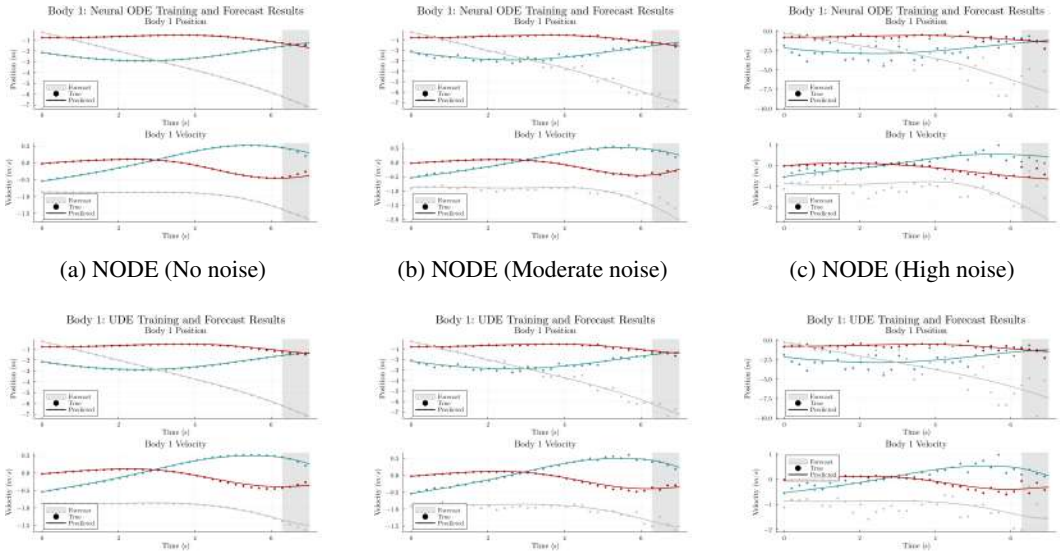
This appendix presents the two additional cases (90% and 40% training coverage) omitted from the main text for brevity. Both follow the same experimental setup and evaluation protocol described in Section 2.

B.1 Case 2: Training on 90% of dataset and forecasting

From Figure 4, the Neural ODE's predictions align well with the true trajectories in the no-noise scenario for both the training and the short 10% forecast window. When noise is introduced, the

model effectively filters it within the training region. However, minor deviations begin to appear in the forecast region, especially under high noise.

From Figure 4, the UDE performs exceptionally well, as expected. With 90% of the data, the model's predictions are virtually indistinguishable from the ground truth across all noise levels. It produces a clean, accurate forecast that perfectly captures the system's dynamics, showcasing its superior robustness and reliability when given ample data.



(a) NODE (No noise) (b) NODE (Moderate noise) (c) NODE (High noise)

(d) UDE (No noise) (e) UDE (Moderate noise) (f) UDE (High noise)

Figure 4: Combined results for Case 2 (90% training). Top row (a-c) shows Neural ODE results; bottom row (d-f) shows UDE results across different noise levels for body 1. The grey shaded region denotes the forecast domain.

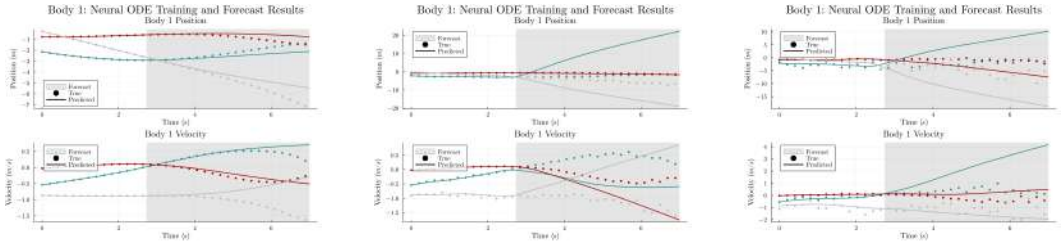
B.2 Case 4: Training on 40% of dataset and forecasting

As shown in Figure 5, the Neural ODE fails to generalize. While it fits the 40% training data, its forecasted trajectory diverges significantly from the ground truth almost immediately, regardless of the noise level. The predicted paths become physically implausible, confirming that 40% data coverage is insufficient for the purely data-driven model.

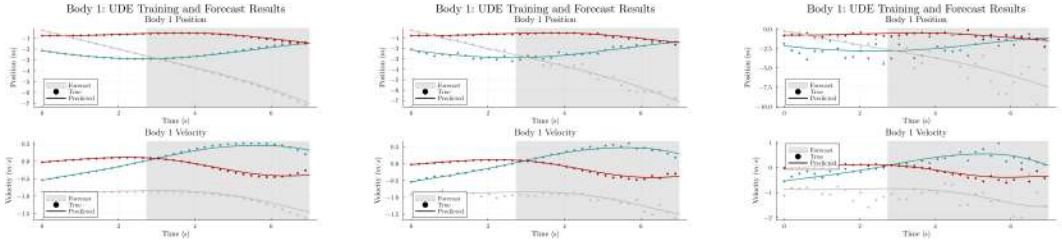
In contrast, Figure 5 demonstrates the UDE's continued robustness. The model provides a stable and accurate long-range forecast in the no-noise and moderate-noise conditions. Under high noise, some minor errors accumulate over the 60% forecast window, but the overall trajectory remains physically plausible and closely follows the underlying dynamics. This starkly contrasts with the Neural ODE's failure, highlighting the critical advantage of incorporating physical dynamics.

C Additional Figures

This section provides the full 3-body trajectory plots for each case, comparing the Neural ODE and UDE models. Dotted lines represent the training data, while the dashed line represents the model's prediction (in the training region) or forecast (in the grey shaded forecast region).



(a) NODE (No noise) (b) NODE (Moderate noise) (c) NODE (High noise)



(d) UDE (No noise) (e) UDE (Moderate noise) (f) UDE (High noise)

Figure 5: Combined results for Case 4 (40% training). Top row (a-c) shows Neural ODE results; bottom row (d-f) shows UDE results across different noise levels for body 1. The grey shaded region denotes the forecast domain.

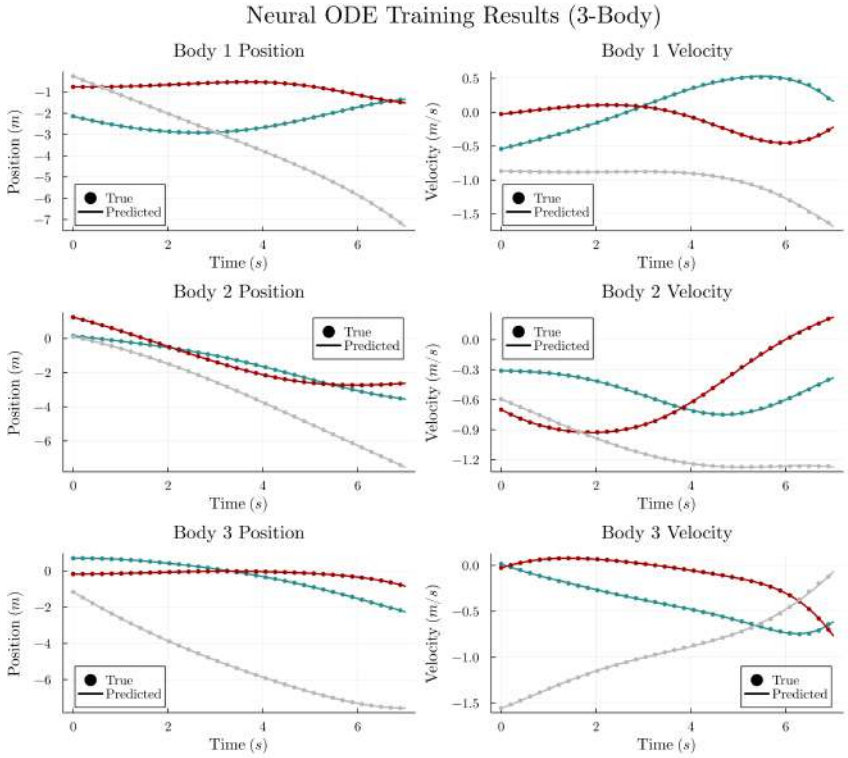


Figure 6: Neural ODE full trajectory plot for 100% training.

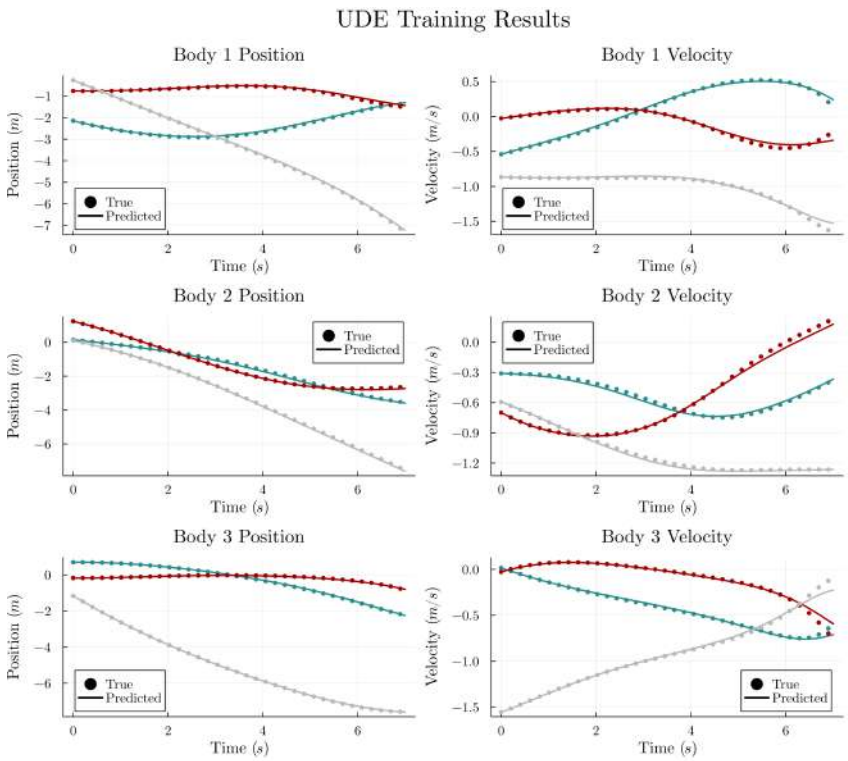


Figure 7: UDE full trajectory plot for 100% training.

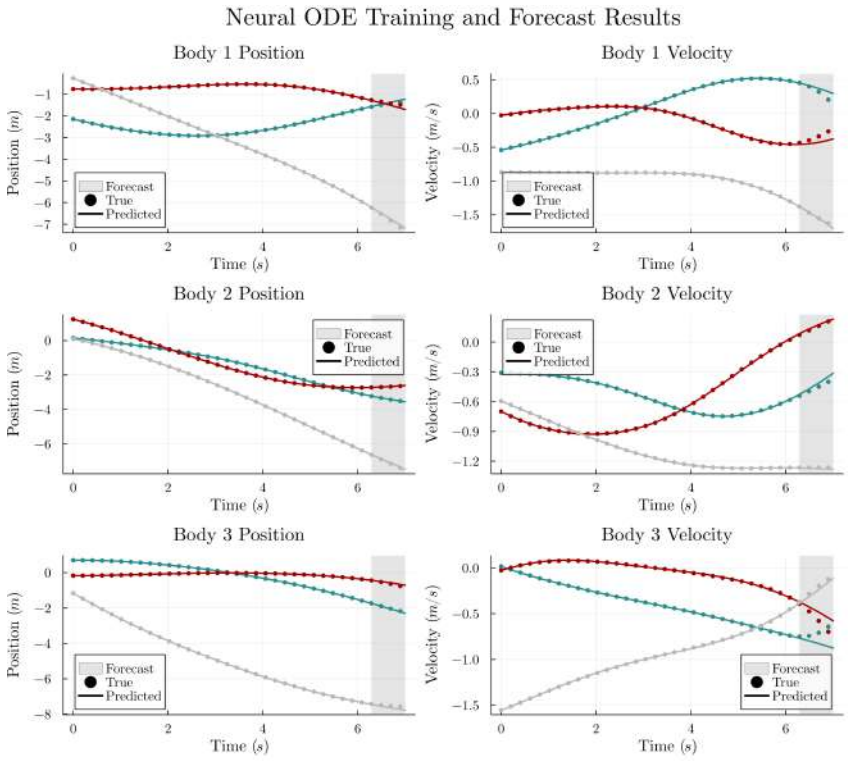


Figure 8: Neural ODE full trajectory plot for 90% training and 10% forecasting. The grey region denotes the forecast domain.

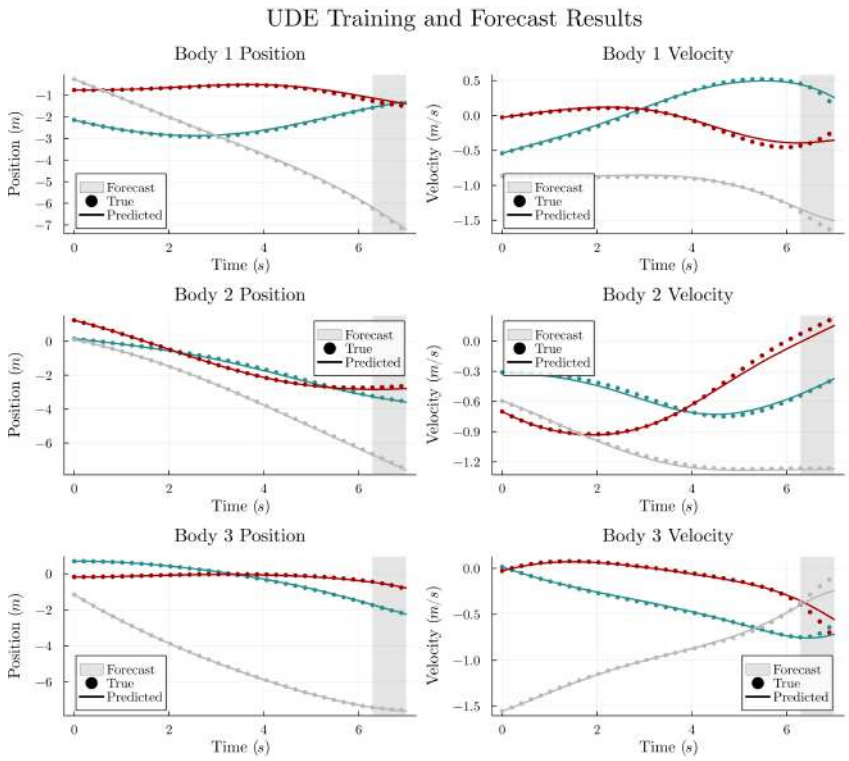


Figure 9: UDE full trajectory plot for 90% training and 10% forecasting. The grey region denotes the forecast domain.

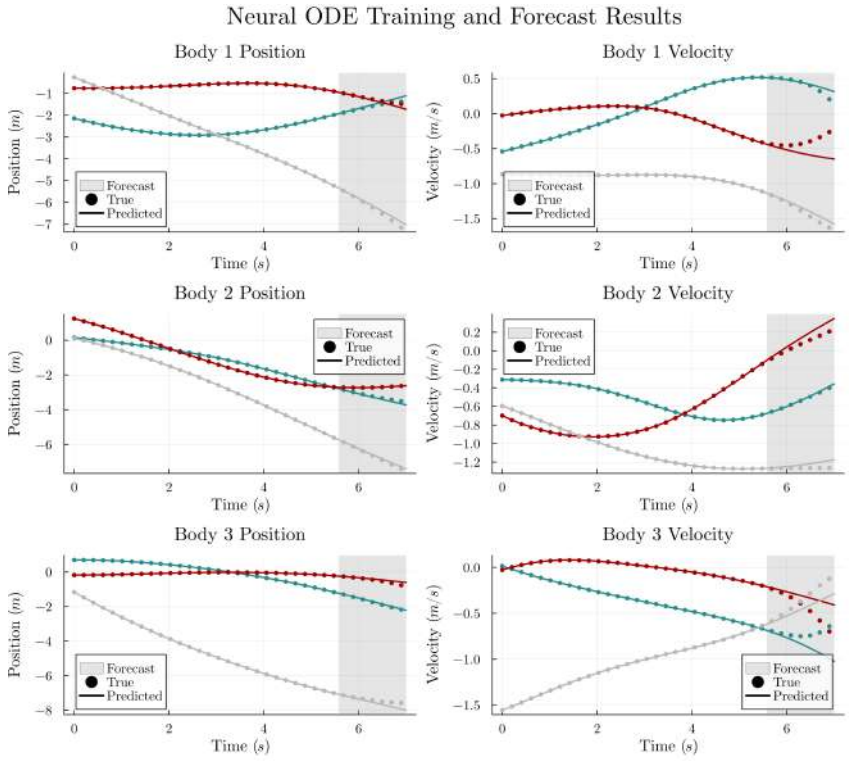


Figure 10: Neural ODE full trajectory plot for 80% training and 20% forecasting. The grey region denotes the forecast domain.

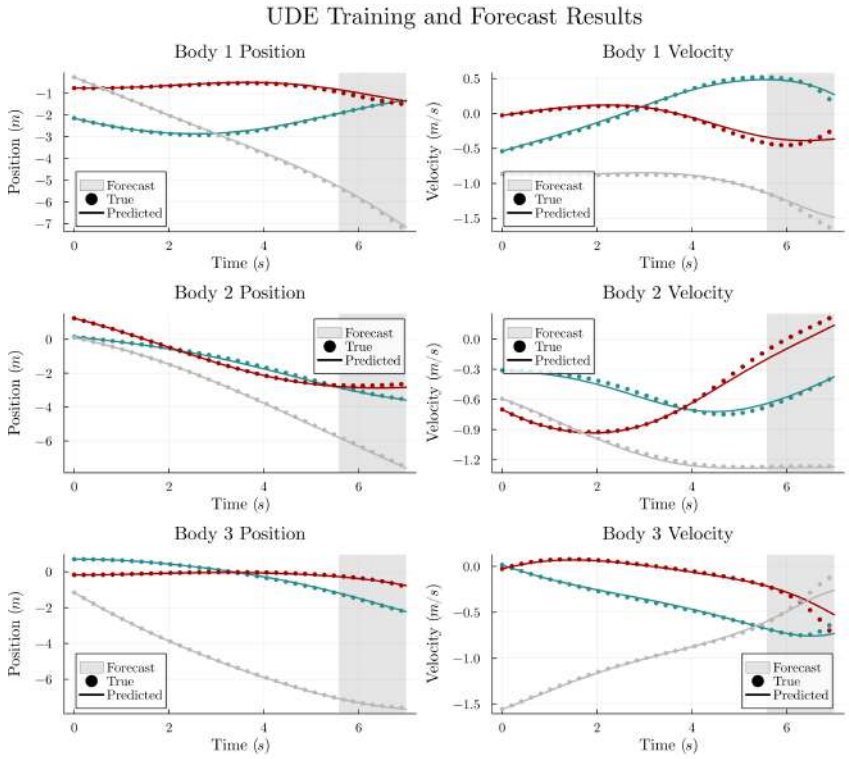


Figure 11: UDE full trajectory plot for 80% training and 20% forecasting. The grey region denotes the forecast domain.

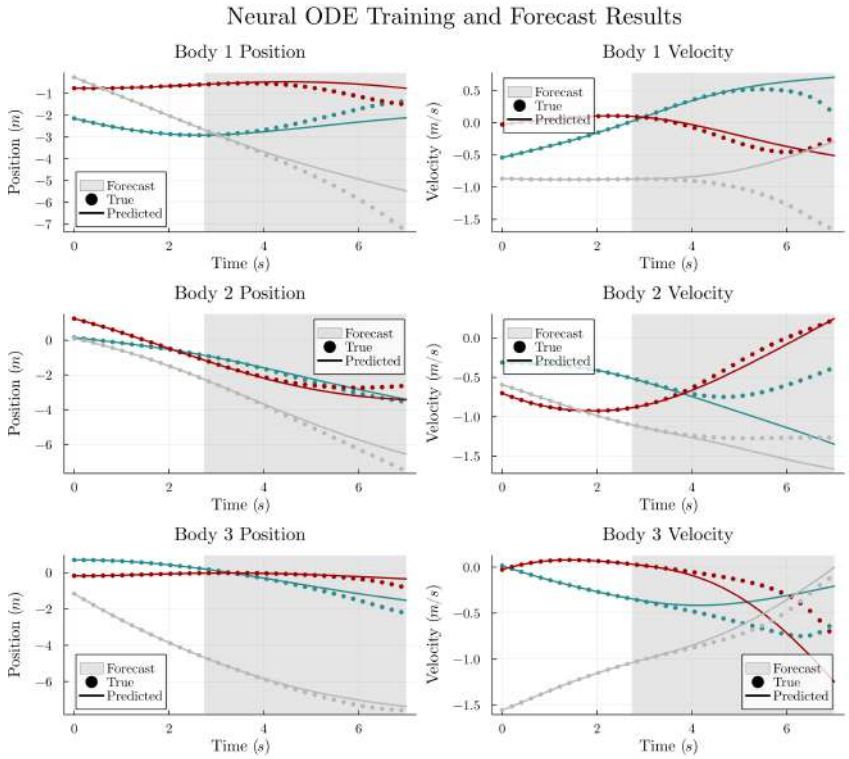


Figure 12: Neural ODE full trajectory plot for 40% training and 60% forecasting. The grey region denotes the forecast domain.

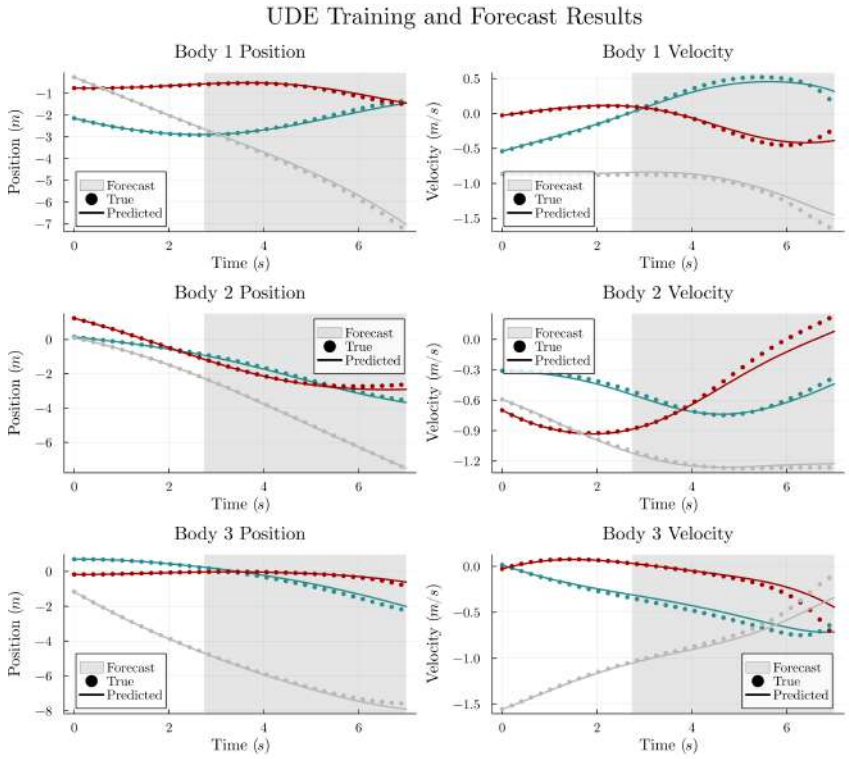


Figure 13: UDE full trajectory plot for 40% training and 60% forecasting. The grey region denotes the forecast domain.

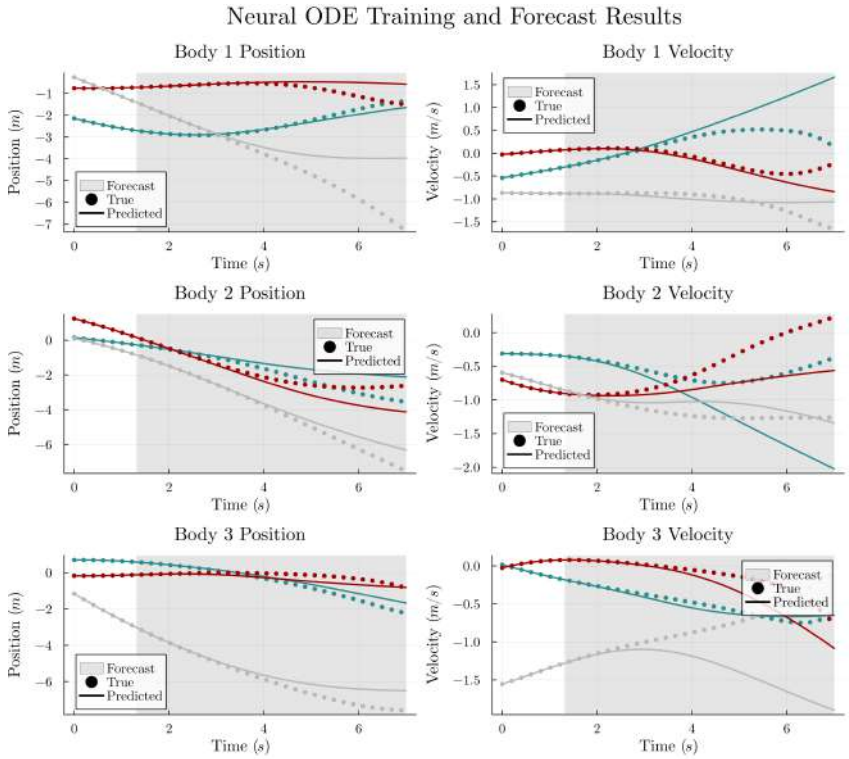


Figure 14: Neural ODE full trajectory plot for 20% training and 80% forecasting. The grey region denotes the forecast domain.

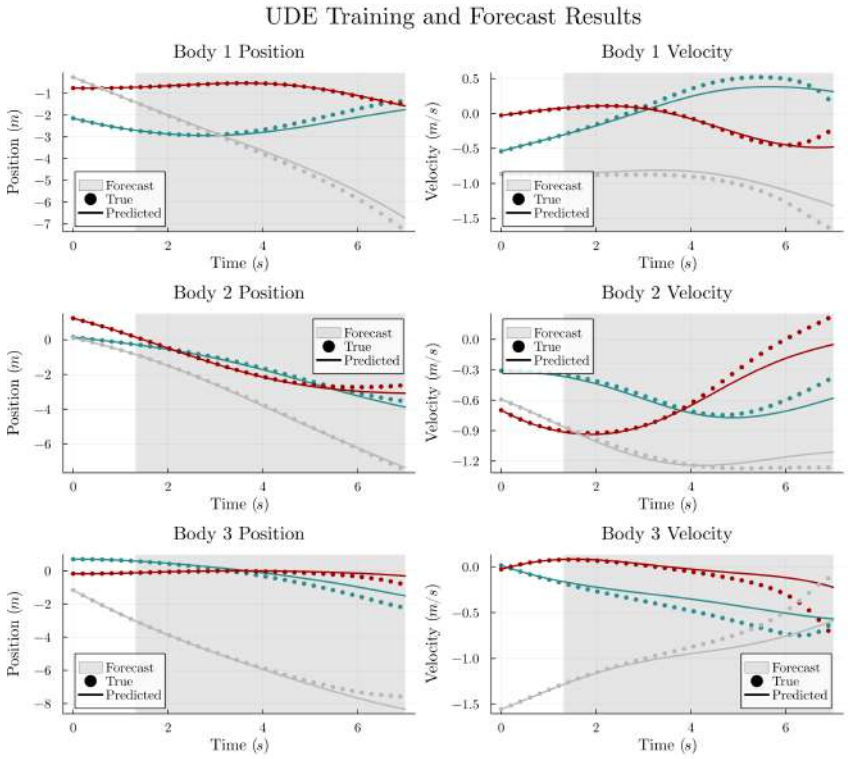


Figure 15: UDE full trajectory plot for 20% training and 80% forecasting. The grey region denotes the forecast domain.