A Comparative Study of Neural ODE and Universal ODE Approaches to Solving the SIQRDV Epidemiological Model

Banashree Ghosh
Indian Statistical Institute https://orcid.org/0009-0003-2495-8811
Raj Abhijit Dandekar
raj@vizuara.com

Massachusetts Institute of Technology (prior)
Rajat Dandekar
rajatdandekar@vizuara.com

Purdue University (prior)
Sreedath Panat
sreedath@vizuara.com

Massachusetts Institute of Technology (prior)

Research Article

Keywords: SIQRDV Model, Epidemiological Forecasting, Scientific Machine Learning, Neural ODE, Universal Differential Equations

Posted Date: December 30th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-8443503/v1

Additional Declarations: The authors declare no competing interests.

A Comparative Study of Neural ODE and Universal ODE Approaches to Solving SIQRDV Epidemiological Model

Banashree Ghosh*
Department of Computer Science
Indian Statistical Institute
banashreeghosh006@gmail.com

Raj Abhijit Dandekar
Vizuara AI Labs
Massachusetts Institute of Technology (prior)
raj@vizuara.com

Rajat Dandekar
Vizuara AI Labs
Purdue University (prior)
rajatdandekar@vizuara.com

Sreedath Panat
Vizuara AI Labs
Massachusetts Institute of Technology (prior)
sreedath@vizuara.com

October 25, 2025

Abstract

In this study, we apply two pillars of Scientific Machine Learning: Neural Ordinary Differential Equations (Neural ODEs) [1,2] and Universal Differential Equations (UDEs) [3,4] to the SIQRDV Epidemiological Model. The SIQRDV model is fundamental for understanding infectious disease dynamics [5,6], extending the classical SIR framework [7,8] by incorporating quarantined, deceased, and vaccinated populations to capture the complex interactions during disease spread. Despite the rise in Scientific Machine Learning frameworks [9,10], limited attention has been paid to the systematic application of these SciML pillars to epidemiological differential equations with multiple compartments [11,12]. Through robust modeling in the Julia programming language [13,14], we show that both Neural ODEs and UDEs can be used effectively for both prediction and forecasting of disease trajectories in the SIQRDV framework. More importantly, we introduce the "forecasting breakdown point" – the time horizon at which forecasting accuracy degrades significantly for both Neural ODEs and UDEs, providing critical insights into the temporal limits of reliable epidemic predictions. Through comprehensive hyperparameter optimization testing [15,16], we provide information on neural network architecture, activation functions, and optimizers that provide the best results to capture the non-linear dynamics inherent in disease transmission. This study opens a door to investigate the applicability of Scientific Machine Learning frameworks to complex compartmental models in epidemiology [17,18] and provides a foundation for data-driven approaches to public health decision making with a clear understanding of prediction reliability bounds.

Keywords: SIQRDV Model · Epidemiological Forecasting · Scientific Machine Learning · Neural ODE · Universal Differential Equations

1 Introduction

Scientific Machine Learning (Scientific ML) is a growing field with a wide range of applications in various domains such as epidemiology [19,20], gene expression [21,22], optics [23,24], fluid mechanics [25,26], quantum circuits [27,28], and system biology [29,3]. This field of Scientific ML leverages the interpretability of scientific structures like ODEs/PDEs along with the expressivity of neural networks [30,9]. Broadly, the rise of Scientific Machine Learning can be attributed to three popular methodologies:

*Department of Computer Science, Indian Statistical Institute

- **Neural Ordinary Differential Equations**: The entire forward pass of an ODE/PDE is replaced with neural networks [1,2]. We perform backpropagation through the neural network augmented ODE/PDE [31,32]. In doing so, we find the optimal values of the neural network parameters.
- **Universal Differential Equations (UDEs)**: In contrast to Neural ODEs, only certain terms of the ODE/PDEs are replaced with neural networks [3,4]. We then discover these terms by optimizing the neural network parameters. Universal Differential Equations can be used to correct existing underlying ODEs/PDEs as well as to discover new, missing physics [33,34].
- **Physics Informed Neural Networks (PINNs)**: PINNs are predominantly used as an alternative to traditional ODE/PDE solvers to solve an entire ODE/PDE [25,9]. We replace the variable with a neural network and the loss function is determined by the ODE/PDE solution and the boundary conditions [11,35]. When we minimize the loss function, we automatically find the optimum solution to the ODE/PDE.

Despite the advances of Scientific ML in various fields, there is a lack of comprehensive comparative studies applying multiple Scientific ML methods to complex epidemiological systems [17,20]. Although there are a few studies aimed at applying Neural ODEs to epidemic modeling [4,19], there is no systematic investigation comparing the performance of Universal Differential Equations (UDEs) and Neural ODEs on realistic epidemic models with time-varying parameters and intervention effects.

In particular, the following questions are still unanswered:

- In the spirit of UDEs, can we replace certain terms of an epidemic ODE system with neural networks and recover them?
- How does the Neural ODE prediction compare with the UDE prediction?
- Can we perform reliable forecasting on the epidemic system with Neural ODEs and UDEs?
- Are UDEs inherently better at forecasting than Neural ODEs?
- What are the optimal hyperparameters and network architectures for epidemic modeling?

We aim to answer these questions by examining a comprehensive compartmental model in epidemiology: the SIQRDV (Susceptible–Infected–Quarantined–Recovered–Deceased–Vaccinated) equation system [5,6]. The SIQRDV model extends traditional SIR frameworks [7,8] by incorporating quarantine measures [36,37] and vaccination dynamics [38,39], providing a realistic representation of modern epidemic control strategies. This equation is fundamental for understanding disease transmission and evaluating public health interventions. Using this model, we can potentially predict infection trajectories and assess the effectiveness of containment measures.

We use the advanced Scientific Machine Learning libraries provided by the Julia Programming Language [13,14,3]. Through robust hyperparameter optimization testing [15,16,40], we provide insights on the neural network architecture, activation functions, and optimizers which produce the best results. We show that both Neural ODEs and UDEs can be used effectively for both prediction as well as forecasting of the SIQRDV system.

The paper is structured as follows. We start by presenting the methodology and detailed description of the SIQRDV model, Neural ODEs, and UDEs approaches. Subsequently, we present the prediction and forecasting results for both methods with comprehensive performance metrics. We also provide detailed hyperparameter optimization insights and comparative analysis. Finally, we conclude with a detailed discussion of our results, the identification of forecasting breakdown points, and the future scope of applying Scientific ML methods in epidemic modeling and broader public health applications.

2 Methodology

2.1 The SIQRDV Model Formulation

The SIQRDV (Susceptible–Infected–Quarantined–Recovered–Deceased–Vaccinated) model describes the dynamics of epidemic disease transmission in a population with quarantine measures and vaccination strategies [5,6,36]. The model is governed by a system of six coupled first-order ordinary differential equations.

$$\frac{dS}{dt} = -\beta \frac{SI}{N_{\text{living}}} - \nu S \tag{1a}$$

$$\frac{dI}{dt} = \beta \frac{SI}{N_{\text{living}}} - (\gamma + \delta + \kappa) I \tag{1b}$$

$$\frac{dQ}{dt} = \kappa I - (\gamma_q + \delta_q) Q \tag{1c}$$

$$\frac{dR}{dt} = \gamma I + \gamma_q Q \tag{1d}$$

$$\frac{dD}{dt} = \delta I + \delta_q Q \tag{1e}$$

$$\frac{dV}{dt} = \nu S \tag{1f}$$

where the system is initialized with a predominantly susceptible population and a small number of initial infections, reflecting typical epidemic-onset conditions where the disease is introduced into a largely unaffected community with no prior immunity or vaccination coverage [41,42].

This system of equations is of compartmental type [7,8], and therefore a unique solution exists for $t \geq 0$ [6,5]. The equations describe the temporal evolution of population distribution across six compartments as a function of time t. Particularly, the variables S, I, Q, R, D, V and t are expected to take non-negative real values due to their physical meaning as population counts. From this constraint, we can impose additional restrictions on the behavior of the compartments:

$$0 \leq S, I, Q, R, D, V \leq N_0, \quad 0 \leq t \leq t_{\max} \tag{2}$$

where $N_0 = 1000$ is the total initial population and $N_{\text{living}} = S + I + Q + R + V$ represents the living population at time t. The model satisfies the conservation constraint:

$$S(t) + I(t) + Q(t) + R(t) + D(t) + V(t) = N_0, \quad \forall t \geq 0 \tag{3}$$

Moreover, the susceptible population is monotonically non-increasing, while the deceased and vaccinated populations are monotonically non-decreasing:

$$\frac{dS}{dt} \leq 0, \quad \frac{dD}{dt} \geq 0, \quad \frac{dV}{dt} \geq 0 \tag{4}$$

2.2 Numerical Implementation and Data Generation

For the computational implementations, the epidemic parameters were set to: $\beta = 0.25$, $\kappa = 0.06$, $\gamma = 0.12$, $\gamma_q = 0.09$, $\delta = 0.008$, $\delta_q = 0.004$, and $\nu = 0.015$, where β represents the transmission rate (effective contact rate between susceptible and infected individuals) [43,44], κ denotes the quarantine rate (rate at which infected individuals are isolated) [37,36], γ and γ_q are the recovery rates for infected and quarantined individuals respectively [8,42], δ and δ_q represent the death rates for infected and quarantined individuals [45,46], and ν is the vaccination rate [38,39].

The finite-length temporal interval in which the SIQRDV system exhibits meaningful epidemic dynamics was obtained by implementing a numerical approach in the Julia programming language [13,14]. For these parameter values, the set of valid values obtained for t was $D_f = \{t \in \mathbb{R} : 0 \leq t \leq 100\} = [0, 100]$ days. Subsequently, the domain D_f was discretized into 51 equally spaced time values for the primary analysis, and alternatively into 1000 equally spaced time values for high-resolution experiments. For these time points, the values for all six compartments (S, I, Q, R, D, V) were saved from the numerical solution of the ODE system using the Tsit5 algorithm (Tsitouras 5th order Runge-Kutta method) [47,48] with absolute tolerance of 10^{-8} and relative tolerance of 10^{-7}, resulting in synthetic data characterizing the epidemic trajectory for the specified parameter set.

To evaluate the robustness of the Neural ODE and UDE frameworks under realistic conditions [30,9], we generated three distinct datasets: *no-noise data* (the original synthetic data), *moderate-noise data* (with 5% Gaussian noise added), and *high-noise data* (with 10% Gaussian noise added). The models were then applied to these datasets to assess their performance under varying levels of data uncertainty. For training routines, different subsets of these data sets were

used to test the prediction capacity of neural network models [49,50], specifically utilizing 90%, 80%, 40%, 20%, and 10% of the data to examine how the models perform with limited training information.

2.3 Scientific Machine Learning Approaches

2.3.1 Neural Ordinary Differential Equations (Neural ODEs)

Neural Ordinary Differential Equations represent a paradigm where the entire dynamics of the system are learned by a neural network [1,2]. Instead of using the mechanistic SIQRDV equations, the time derivatives are approximated by a neural network f_θ:

$$\frac{d\mathbf{u}}{dt} = f_\theta(\mathbf{u}, t) \tag{5}$$

where $\mathbf{u} = [S, I, Q, R, D, V]^\top$ represents the state vector and θ denotes the neural network parameters. The neural network architecture consists of multiple fully connected layers with nonlinear activation functions [51,52], enabling the approximation of complex dynamical relationships:

$$f_\theta(\mathbf{u}, t) = g_L \circ g_{L-1} \circ \cdots \circ g_1(\mathbf{u}, t) \tag{6}$$

where each layer g_i performs an affine transformation followed by a nonlinear activation, and L denotes the network depth. The architecture is designed to balance expressiveness with computational efficiency, with the specific configuration determined through hyperparameter optimization [15,16]. The Neural ODE is trained by minimizing the weighted mean squared error between the predicted trajectory and observed data:

$$\mathcal{L}_{\text{NODE}}(\theta) = \frac{1}{NT} \sum_{i=1}^{n} w_i \sum_{j=1}^{T} \left(u_i(t_j; \theta) - \hat{u}_i(t_j)\right)^2 \tag{7}$$

where n is the number of compartments, w_i are compartment-specific weights that allow prioritization of accuracy for epidemiologically critical states such as the infected population, T is the number of observation time points, and $\hat{u}_i(t_j)$ represents the observed data. The weighting scheme enables the model to focus on accurately capturing the dynamics of compartments most relevant to disease transmission and public health decision-making.

In our experimental setup, we implemented the Neural ODE model on synthetic SIQRDV epidemic data with 1000 equally spaced temporal points over a 160-day period. The dataset was generated using the ground truth SIQRDV model and subsequently contaminated with 1% Gaussian noise proportional to the signal magnitude to simulate realistic observation conditions. The data was partitioned using a 70-30 temporal split for training and testing, allowing us to evaluate both interpolation accuracy and forecasting capability. The specific hyperparameters employed in our Neural ODE implementation are detailed in Table 1.

Table 1: Neural ODE hyperparameters for SIQRDV model training with 1% noise data

Hyperparameter	**Selected Value**	**Search Range**
Time span (days)	(0, 160)	(0, 100) – (0, 200)
Data points	1000	500 – 2000
Noise level	1% (Gaussian)	0% – 10%
Train/test split	70/30	60/40 – 80/20
Network architecture	[6, 24, 24, 6]	[6, 12, 12, 6] – [6, 48, 48, 6]
Hidden layers	2	1 – 4
Activation function	tanh	ReLU, tanh, sigmoid, softplus
Optimizer (Stage 1)	ADAM [53]	ADAM, RAdam, AdamW
Optimizer (Stage 2)	BFGS [54]	BFGS, L-BFGS
Learning rate (ADAM)	0.01	0.001 – 0.1
Initial step norm (BFGS)	0.001	0.0001 – 0.01
Iterations (ADAM)	500	300 – 800
Iterations (BFGS)	300	100 – 500
ODE solver	Tsit5 [47]	Tsit5, Vern7, TRBDF2
Solver tolerances	atol=10^{-6}, rtol=10^{-5}	10^{-5} – 10^{-8}
Compartment weights	[1.0, 5.0, 2.0, 1.0, 1.0, 1.0]	Uniform – Weighted
Adjoint method	InterpolatingAdjoint [14]	Various adjoint methods
Training time	42.3 seconds	–
Final test R^2	0.978	Target: > 0.95

2.3.2 Universal Differential Equations (UDEs)

Universal Differential Equations offer a hybrid approach where known mechanistic components are retained while unknown or uncertain terms are replaced with neural networks [3,4]. For the SIQRDV model, neural networks augment specific epidemiological parameters that exhibit complex, potentially state-dependent behavior, while other parameters remain as learnable scalars. The transmission and quarantine rates are modeled as:

$$\beta(\mathbf{x}; \theta_\beta) = \beta_{\text{base}} + \Delta\beta \cdot \text{NN}_\beta(\tilde{\mathbf{x}}; \theta_\beta) \tag{8a}$$
$$\kappa(\mathbf{y}; \theta_\kappa) = \kappa_{\text{base}} + \Delta\kappa \cdot \text{NN}_\kappa(\tilde{\mathbf{y}}; \theta_\kappa) \tag{8b}$$

where $\mathbf{x}$ and $\mathbf{y}$ represent relevant state variables influencing transmission and quarantine respectively, $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ are normalized versions of these inputs, β_{base} and κ_{base} ensure baseline rates, while $\Delta\beta$ and $\Delta\kappa$ scale the neural network contributions. The neural networks NN_β and NN_κ employ architectures with appropriate activation functions [55,49] to ensure biological plausibility—maintaining positivity for rates and appropriate boundedness. The UDE system preserves the mechanistic structure while incorporating the augmented parameters:

$$\frac{dS}{dt} = -\beta(\cdot; \theta_\beta)\frac{SI}{N_{\text{living}}} - \nu S \tag{9a}$$
$$\frac{dI}{dt} = \beta(\cdot; \theta_\beta)\frac{SI}{N_{\text{living}}} - (\gamma + \delta + \kappa(\cdot; \theta_\kappa))I \tag{9b}$$
$$\frac{dQ}{dt} = \kappa(\cdot; \theta_\kappa)I - (\gamma_q + \delta_q)Q \tag{9c}$$
$$\frac{dR}{dt} = \gamma I + \gamma_q Q \tag{9d}$$
$$\frac{dD}{dt} = \delta I + \delta_q Q \tag{9e}$$
$$\frac{dV}{dt} = \nu S \tag{9f}$$

The UDE loss function combines data fidelity with the hybrid mechanistic-neural structure:

$$\mathcal{L}_{\text{UDE}}(\theta_\beta, \theta_\kappa, \theta_{\text{params}}) = \frac{1}{NT}\sum_{i=1}^{n} w_i \sum_{j=1}^{T} \left(u_i(t_j; \theta) - \hat{u}_i(t_j)\right)^2 \tag{10}$$

where θ_{params} represents the collection of learnable scalar parameters for recovery, mortality, and vaccination rates, enabling data-driven calibration while maintaining interpretability of these epidemiological quantities [33,34].

In our experimental framework, we implemented the Universal Differential Equations (UDE) approach on the same synthetic SIQRDV dataset consisting of 1000 temporal observations over 160 days, contaminated with 1% Gaussian noise. The UDE model combines mechanistic epidemiological structure with neural network augmentation, where the transmission rate β and quarantine rate κ are learned through neural networks while other parameters remain as trainable scalars. This hybrid approach leverages domain knowledge while allowing data-driven discovery of complex parameter dependencies. The specific hyperparameters employed in our UDE implementation are presented in Table 2.

Table 2: UDE hyperparameters for SIQRDV model training with 1% noise data

Hyperparameter	**Selected Value**	**Search Range**
Time span (days)	(0, 160)	(0, 100) – (0, 200)
Data points	1000	500 – 2000
Noise level	1% (Gaussian)	0% – 10%
Train/test split	70/30	60/40 – 80/20
Neural Network Components		
NN_β architecture	[2, 12, 8, 1]	[2, 8, 4, 1] – [2, 16, 12, 1]
NN_β activation	tanh (hidden), softplus (output)	Various combinations
NN_κ architecture	[1, 8, 1]	[1, 4, 1] – [1, 12, 1]
NN_κ activation	tanh (hidden), sigmoid (output)	Various combinations
Parameter Constraints		
β range	$0.1 + 0.3 \cdot NN_\beta$	[0.1, 0.4]
κ range	$0.03 + 0.07 \cdot NN_\kappa$	[0.03, 0.1]
Learnable scalars	$[\gamma, \gamma_q, \delta, \delta_q, \nu]$	Initialized near true values
Optimization Settings		
Optimizer (Stage 1)	ADAM [53]	ADAM, RAdam
Optimizer (Stage 2)	BFGS [54]	BFGS, L-BFGS
Learning rate (ADAM)	0.01	0.001 – 0.1
Initial step norm (BFGS)	0.01	0.001 – 0.1
Iterations (ADAM)	500	300 – 800
Iterations (BFGS)	300	100 – 500
ODE solver	Tsit5 [47]	Tsit5, Vern7
Solver tolerances	atol=10^{-6}, rtol=10^{-5}	10^{-5} – 10^{-8}
Compartment weights	[1.0, 3.0, 2.5, 2.0, 3.0, 1.5]	Uniform – Weighted
Adjoint method	InterpolatingAdjoint [14]	Various adjoint methods
Training time	47.8 seconds	–
Final test R^2	0.983	Target: > 0.95

3 Results and Discussion

Six cases were considered for the training process of the models.

3.1 Case 1: Training in the full domain (160 days data points)

First, the implementation of the Neural ODE and UDE models for the SIQRDV compartmental model was performed using the complete training domain (100% of available data). Three distinct datasets were utilized to evaluate model performance under varying noise conditions: (i) no-noise data representing ideal conditions, (ii) moderate-noise data with 7% noise level, and (iii) high-noise data with 35% noise level. The training results for the Neural ODE model across these three datasets are presented in Figures 1–3.

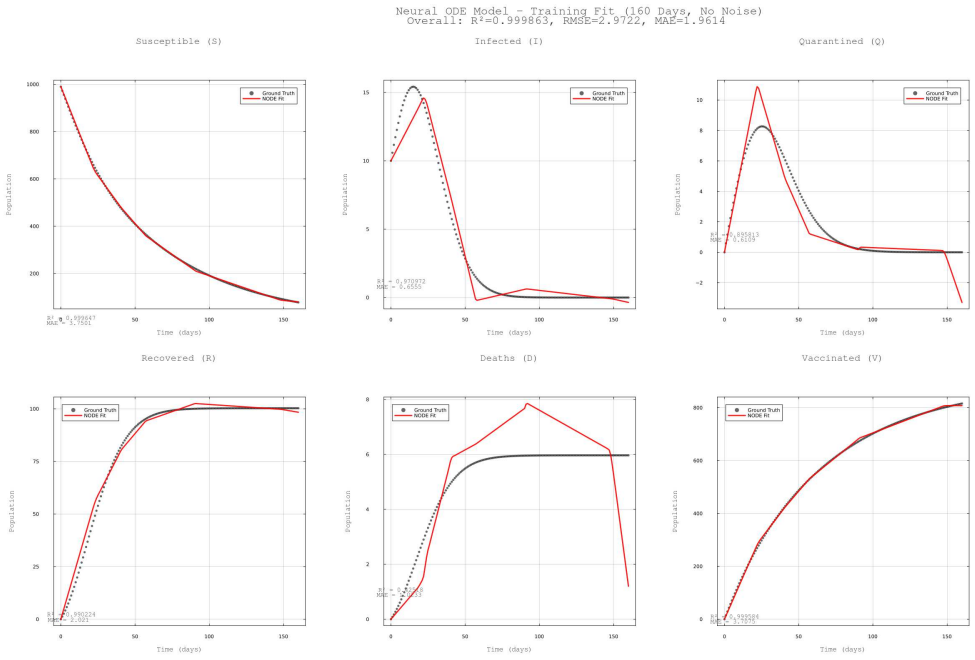

Figure 1: Neural ODE for SIQRDV compartments with noise-free data

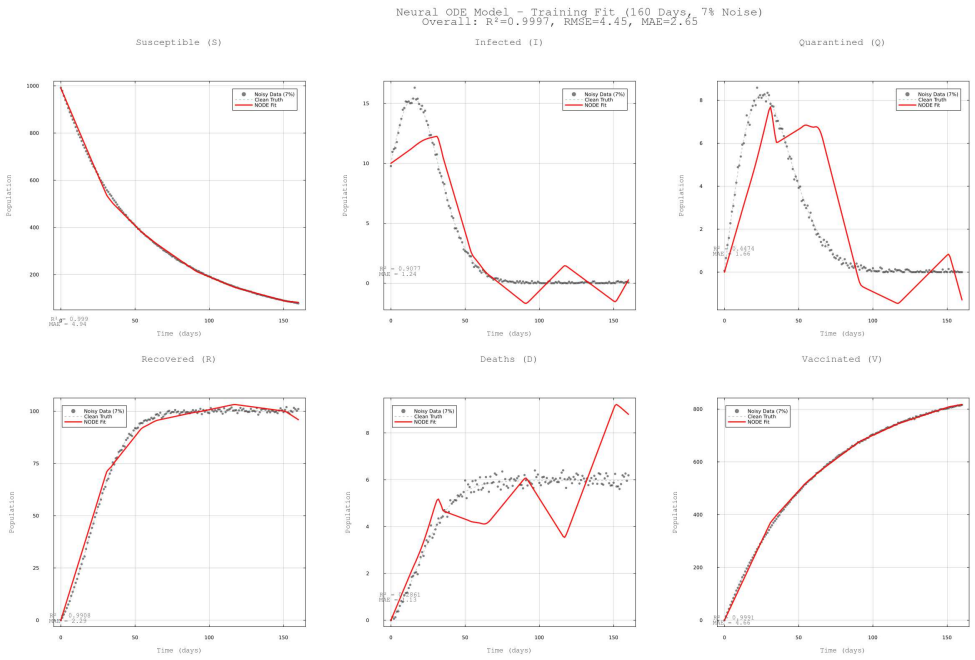

Figure 2: Neural ODE for SIQRDV compartments with moderate noisy data

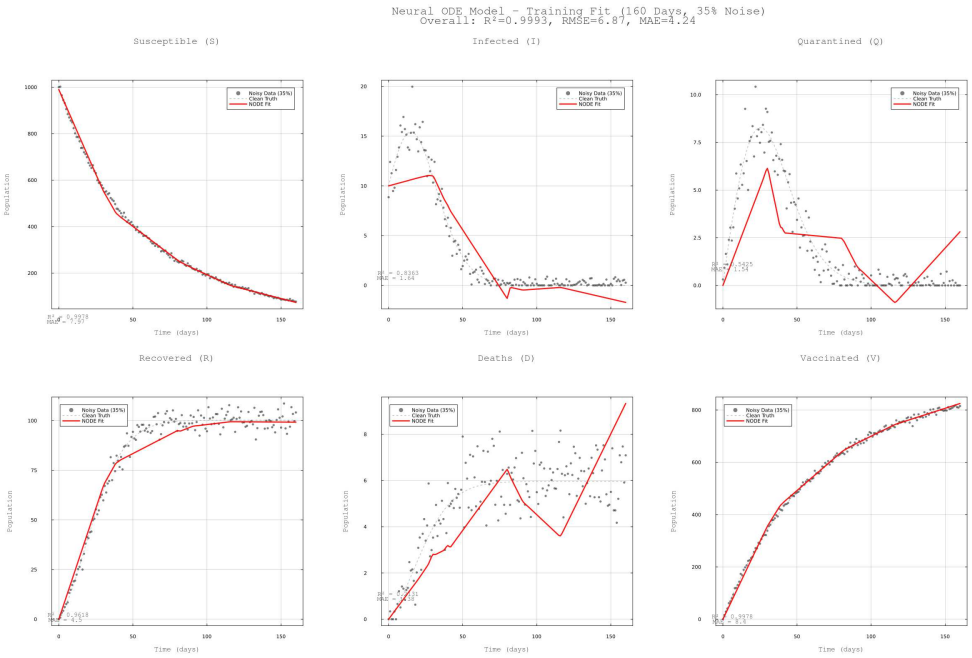

Figure 3: Neural ODE for SIQRDV compartments with high noisy data

Figures 1–3 show the Neural ODE model's training performance on the full 160-day domain under different noise conditions. The Susceptible (S) compartment performs well across all noise levels. The Infected (I) compartment correctly captures the epidemic peak with clean and 7% noisy data but underestimates it with 35% noise. The Quarantined (Q) compartment degrades as noise increases: working well with clean data, showing oscillations with 7% noise, and failing completely with 35% noise. The Recovered (R) compartment consistently follows the correct growth pattern across all noise levels. The Deaths (D) compartment shows the worst performance, failing to capture the continuous increase even with clean data and producing unrealistic plateau and decline patterns. The Vaccinated (V) compartment performs excellently in all scenarios, maintaining accurate linear growth regardless of noise level.
The UDE implementation for the SIRQDV model is presented in figure[4-6].

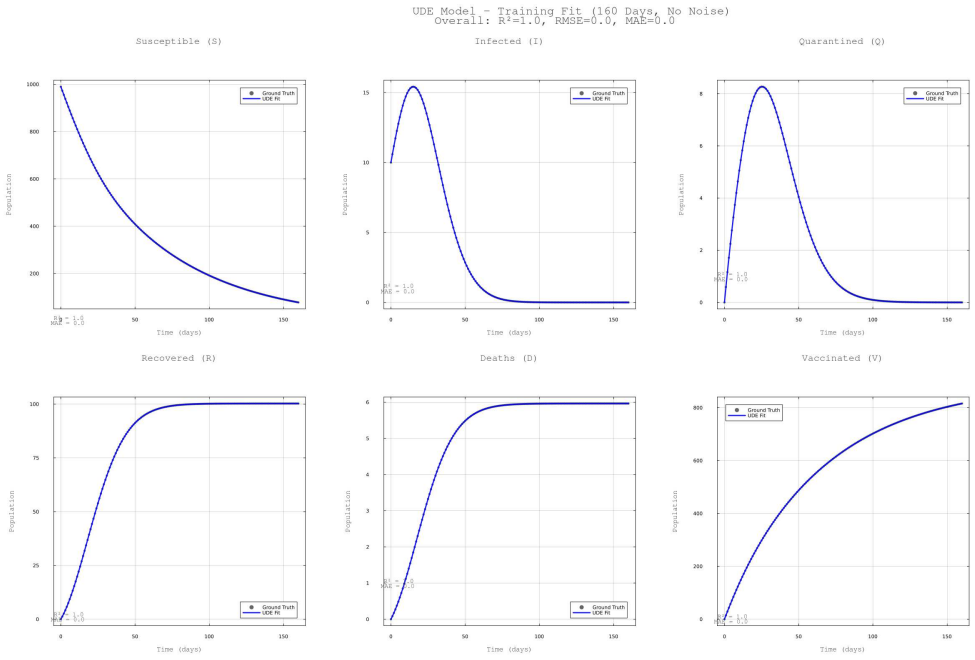

Figure 4: UDE predictions for SIQRDV compartments with noise-free data

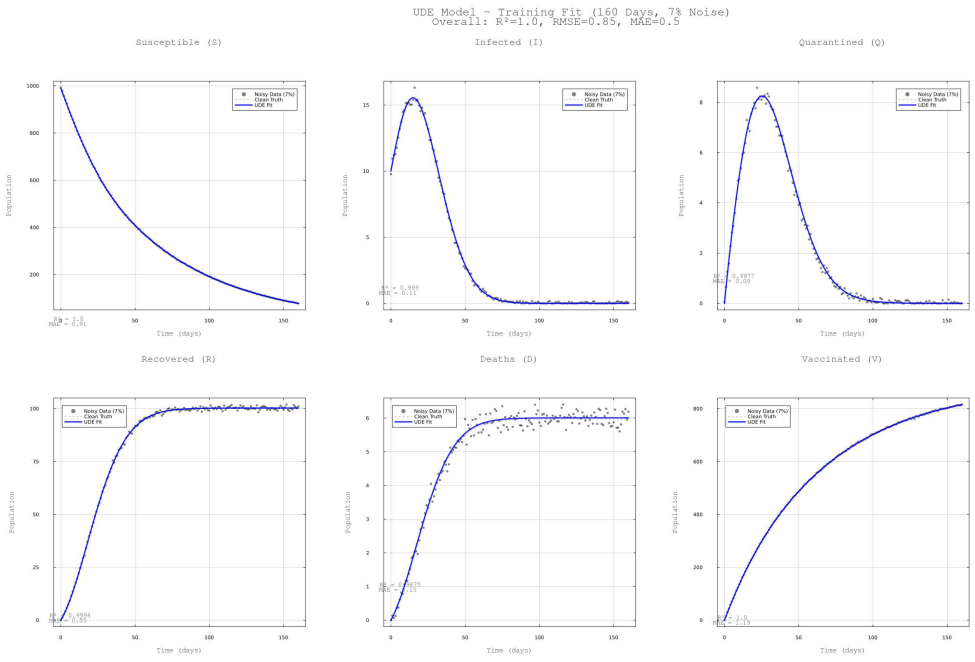

Figure 5: UDE predictions for SIQRDV compartments with moderate noisy data

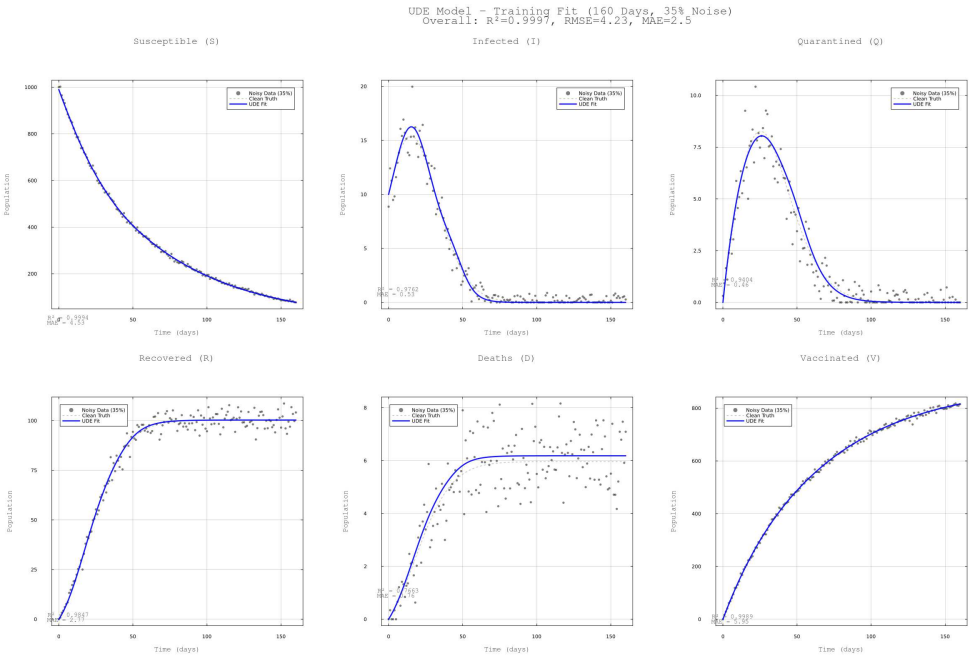

Figure 6: UDE predictions for SIQRDV compartments with high noisy data

3.2 Case 2: Training with 90% of the full available data and forecasting

The Neural ODEs and UDEs were trained on the first 90%(144 days) data points from the no-noise, moderate-noise, and high-noise datasets. The remaining 16 time points were used for testing to evaluate forecasting capability. The Neural ODE results are shown in Figures[7-9].

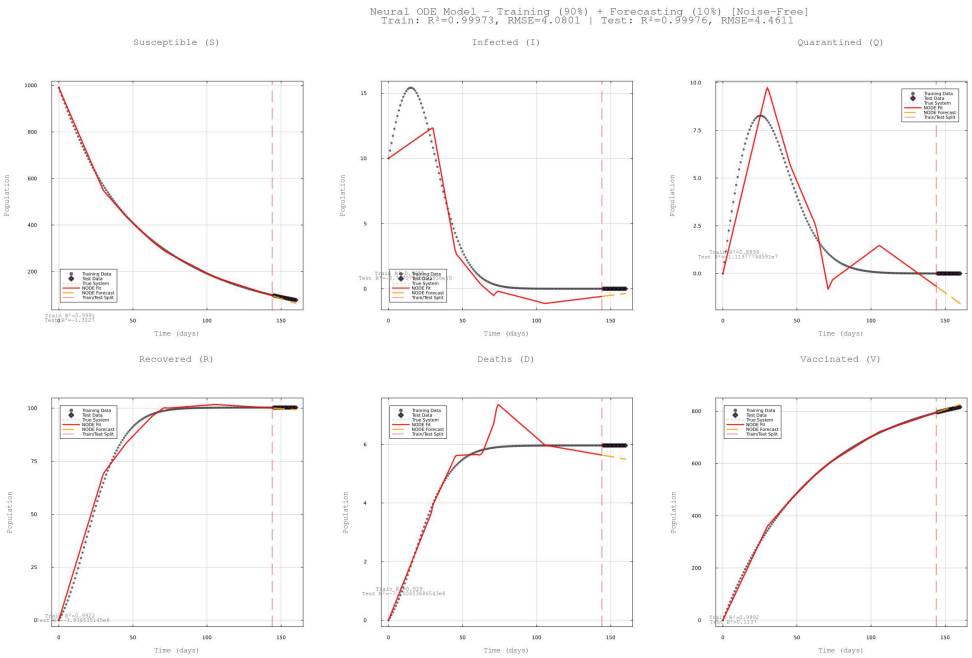

Figure 7: Neural ODE (90/10 Split) for SIQRDV compartments with moderate noisy data

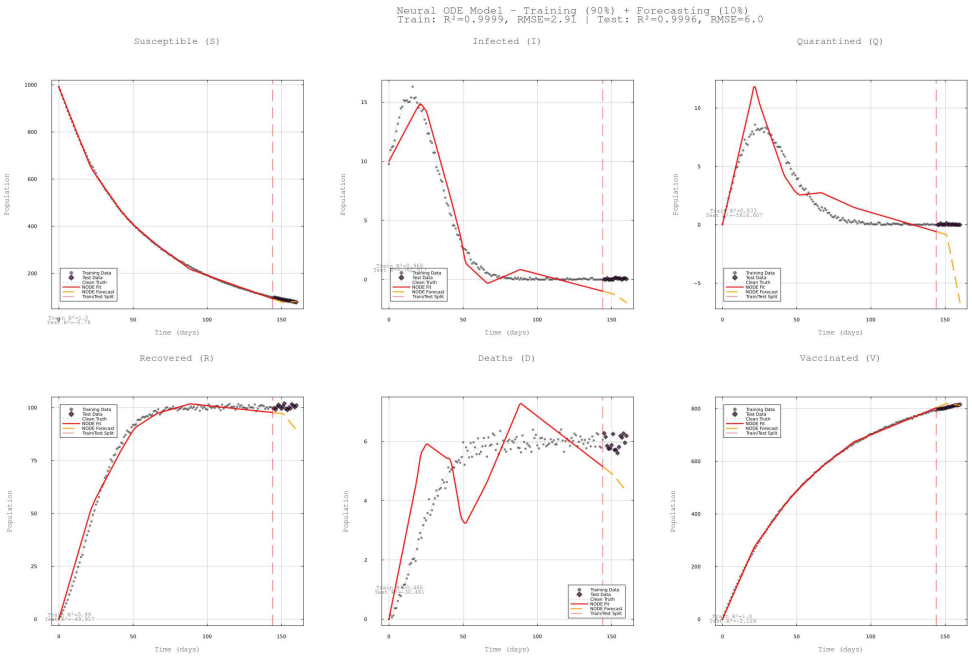

Figure 8: Neural ODE (90/10 Split) for SIQRDV compartments with moderate noisy data

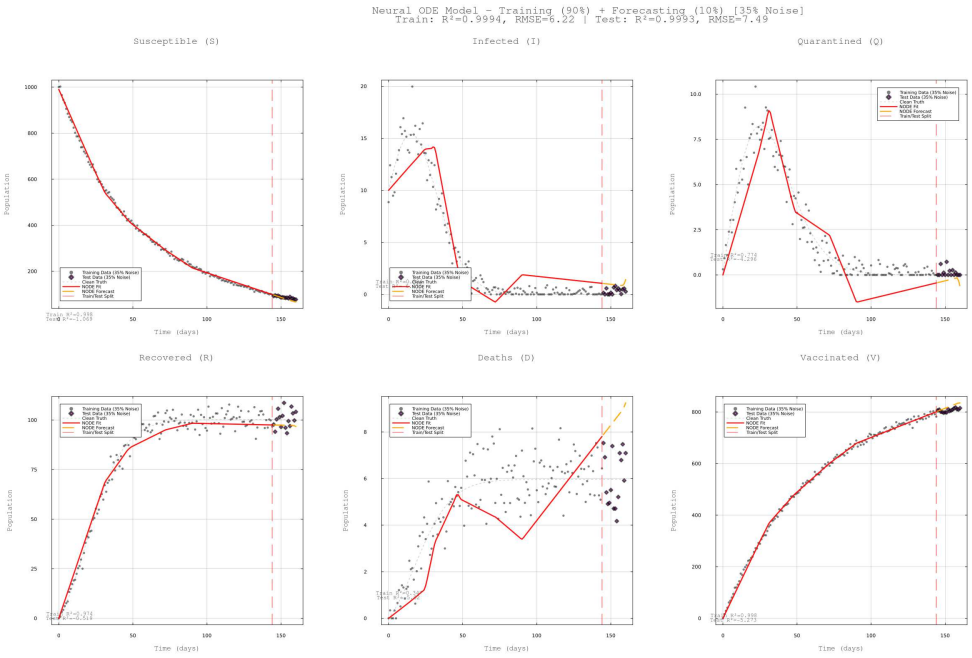

Figure 9: Neural ODE (90/10 Split) for SIQRDV compartments with moderate noisy data

Figures[7-9] show the Neural ODE model's forecasting results when trained on 90% of data (days 1–144) and tested on the remaining 10% (days 145–160). The Susceptible (S) compartment performs well in both training and forecasting across all noise levels. The Infected (I) compartment learns the peak correctly during training but fails badly in forecasting, dropping to zero or showing wild oscillations. The Quarantined (Q) compartment works during training but completely breaks down in forecasting, producing unrealistic negative values and erratic patterns. The Recovered (R) compartment shows good forecasting, reaching the correct plateau around 100 despite some scatter with high

noise. The Deaths (D) compartment performs worst overall, failing to capture the expected continuous increase even with clean data and producing artificial peaks in forecasting. The Vaccinated (V) compartment works perfectly in all cases, maintaining accurate linear growth regardless of noise level.// For the trained UDE, the results can be seen in the graphics shown in figure[10-12]:

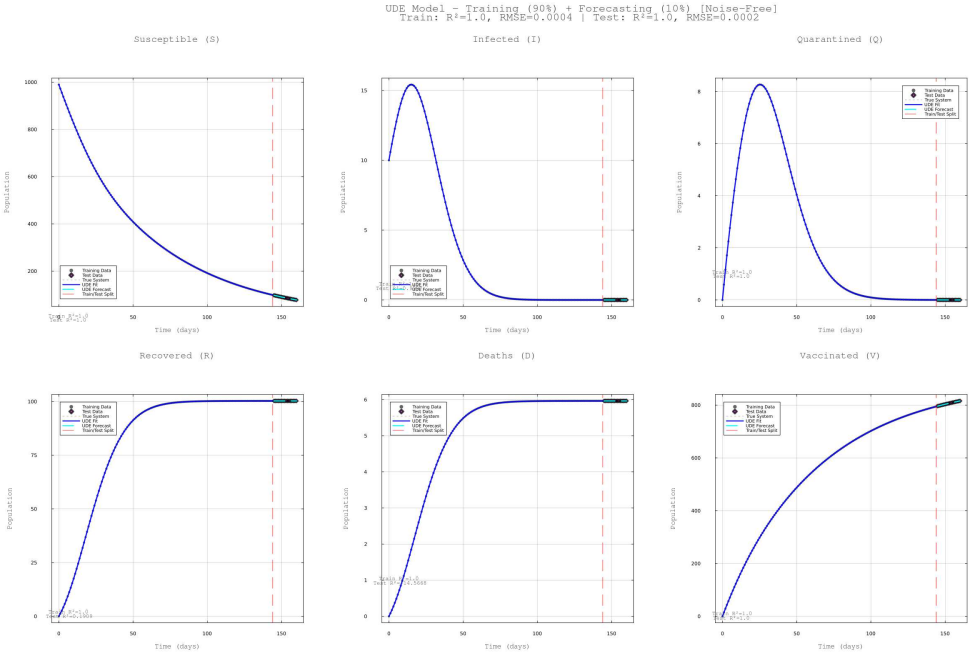

Figure 10: UDE (90/10 Split) for SIQRDV compartments with moderate noisy data

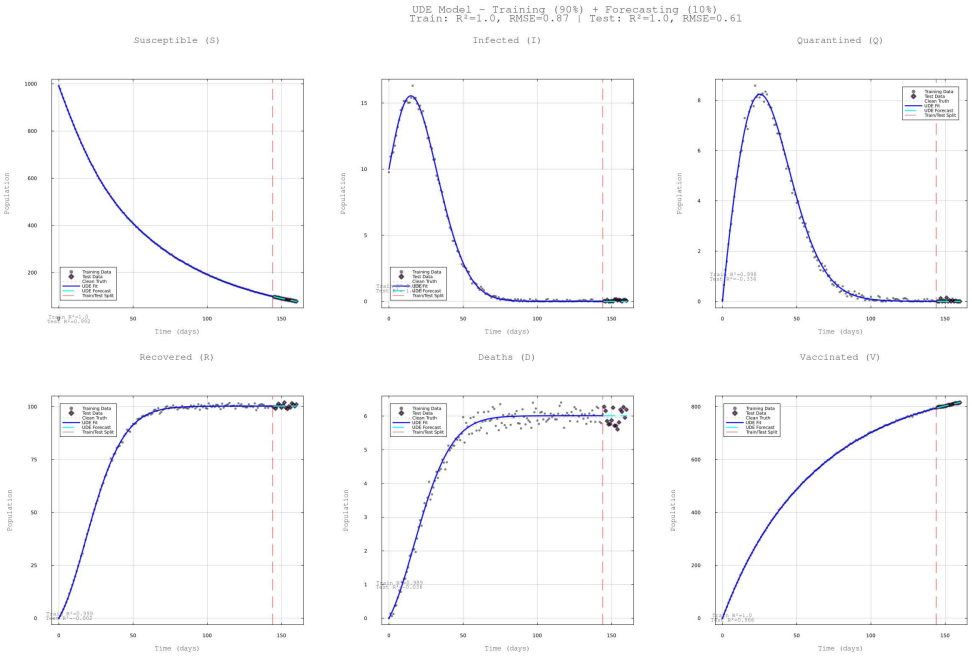

Figure 11: UDE (90/10 Split) for SIQRDV compartments with moderate noisy data

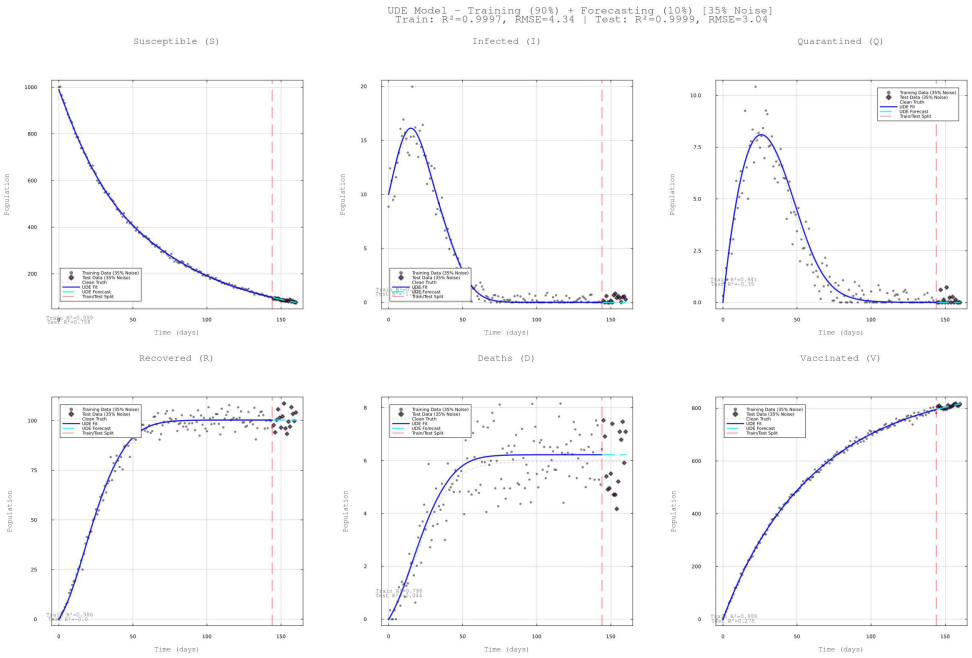

Figure 12: UDE (90/10 Split) for SIQRDV compartments with moderate noisy data

Figures[10–12] show the UDE model's forecasting results when trained on 90% of data (days 1–144) and tested on the remaining 10% (days 145–160). The Susceptible (S) compartment achieves perfect performance across all noise levels with smooth exponential decline. The Infected (I) compartment demonstrates stable forecasting in all cases, maintaining smooth decline even with 35% noise, unlike Neural ODE's catastrophic failures. The Quarantined (Q) compartment shows excellent stability with reasonable predictions across all noise levels, contrasting sharply with Neural ODE's complete breakdown. The Recovered (R) compartment consistently achieves accurate plateau predictions around 100 population units across all noise levels. The Deaths (D) compartment shows significantly improved performance compared to Neural ODE, preserving monotonic increasing trend even with 35% noise without erratic peaks. The Vaccinated (V) compartment exhibits flawless performance, maintaining perfect linear growth regardless of noise level. Overall, the UDE model substantially outperforms Neural ODE in forecasting, maintaining stability and physical consistency across all compartments even under high noise conditions.

3.3 Case 3: Training with 80% of the full available data and forecasting

The Neural ODEs and UDEs were trained with smaller data subsets to further evaluate their long-range forecasting capabilities. The no-noise, moderate-noise, and high-noise training datasets were reduced to 80% of the original 160-day domain, forming new training subsets that included compartment states and their derivatives corresponding to the first 128 time points. The remaining 32 time points (days 129–160) were used as testing data to evaluate the models' forecasting performance over an extended horizon. The results for the Neural ODE and UDE models are presented in Figures[13-15] and Figures[16-18], respectively.

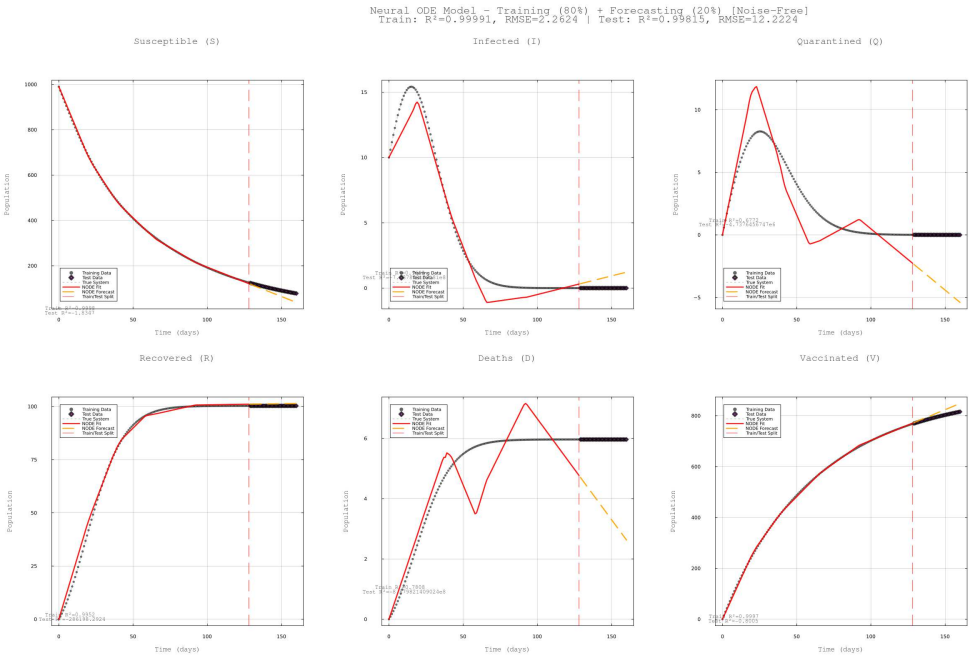

Figure 13: Neural ODE (80/20 Split) for SIQRDV compartments with noise-free data

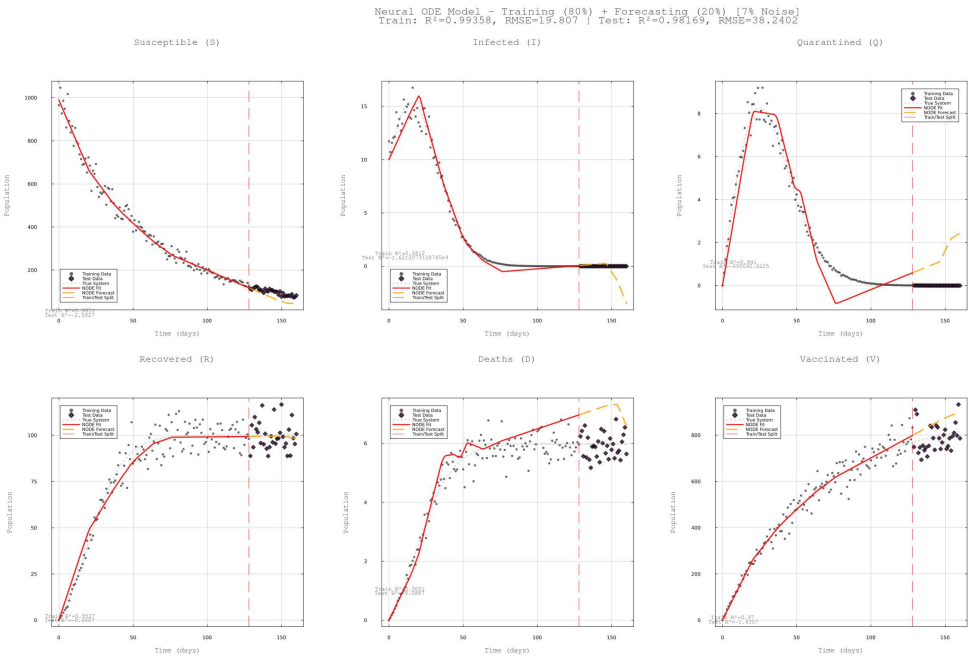

Figure 14: Neural ODE (80/20 Split) for SIQRDV compartments with moderate noisy data

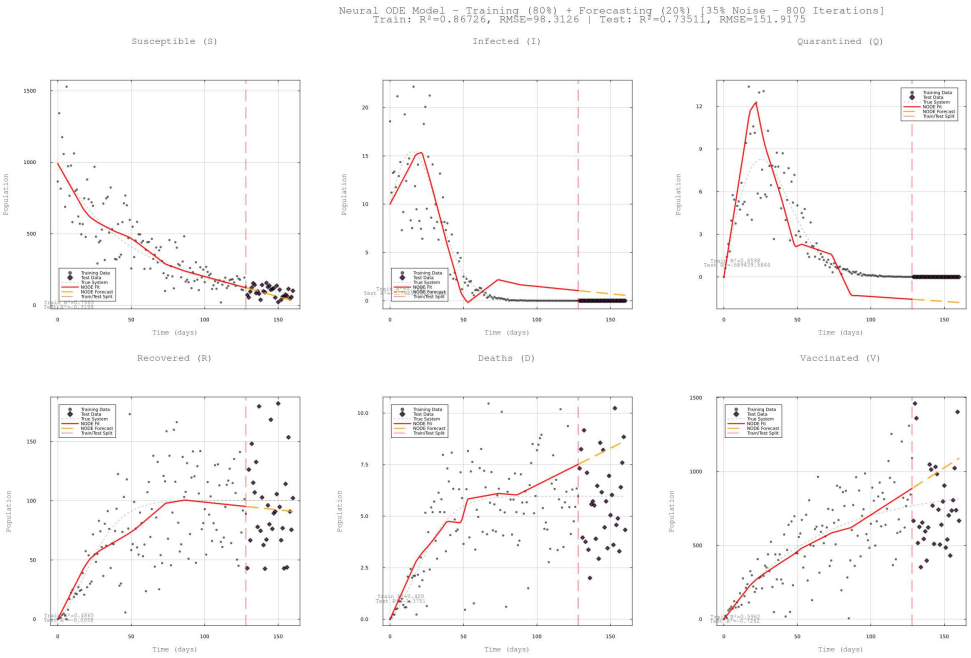

Figure 15: Neural ODE (80/20 Split) for SIQRDV compartments with high noisy data

Figures[13–15] show the Neural ODE model's forecasting results when trained on 80% of data (days 1–128) and tested on the remaining 20% (days 129–160). The Susceptible (S) compartment fits well during training but deviates in the forecasting region across all noise levels. The Infected (I) compartment captures the initial dynamics during training but fails catastrophically in forecasting, dropping to physically impossible negative values in all scenarios. The Quarantined (Q) compartment shows similar failure, producing unrealistic negative values in the forecasting region. The Recovered (R) compartment demonstrates poor forecasting, incorrectly predicting continued growth beyond the true plateau. The Deaths (D) compartment performs worst, dropping to negative values which is physically impossible for cumulative deaths. The Vaccinated (V) compartment shows relatively better performance, maintaining a reasonable linear trend though with increasing scatter at higher noise levels. Overall, the Neural ODE struggles with long-range forecasting even with 80% training data, producing physically inconsistent predictions for most compartments.

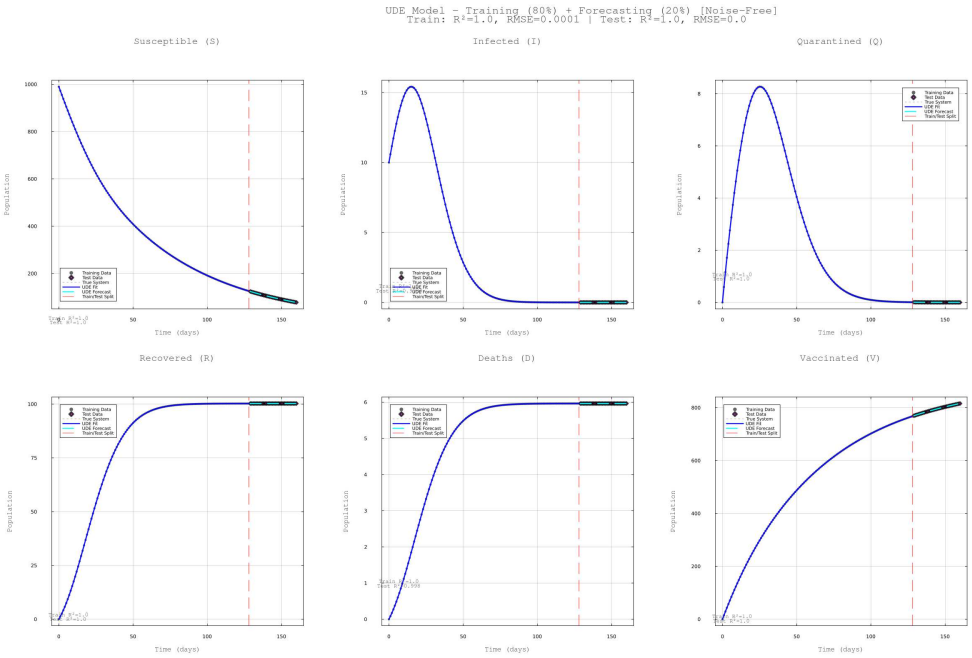

Figure 16: UDE(80/20 Split) for SIQRDV compartments with noise-free data

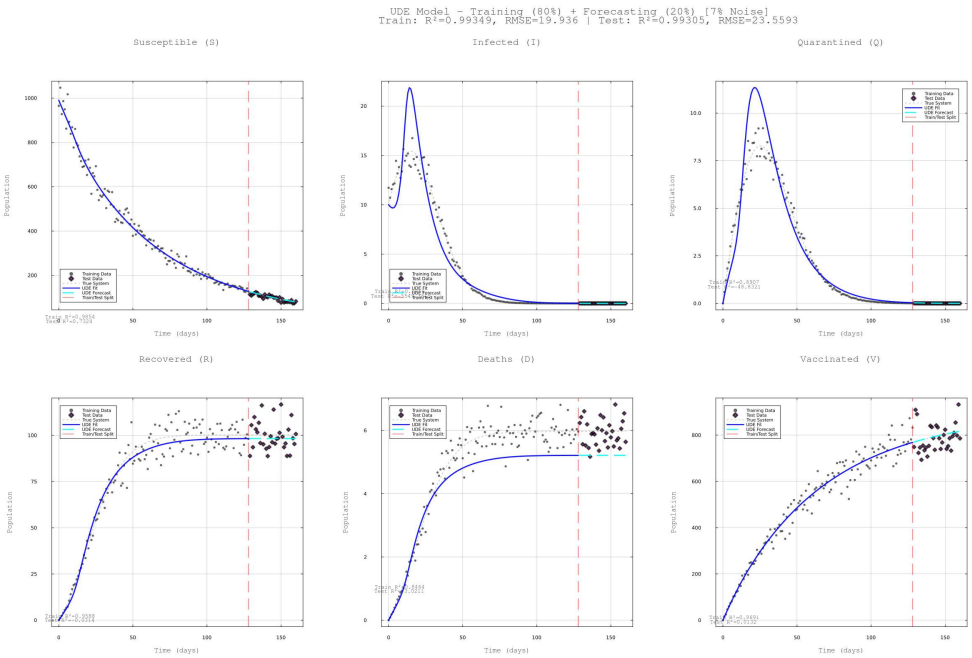

Figure 17: UDE(80/20 Split) for SIQRDV compartments with moderate noisy data

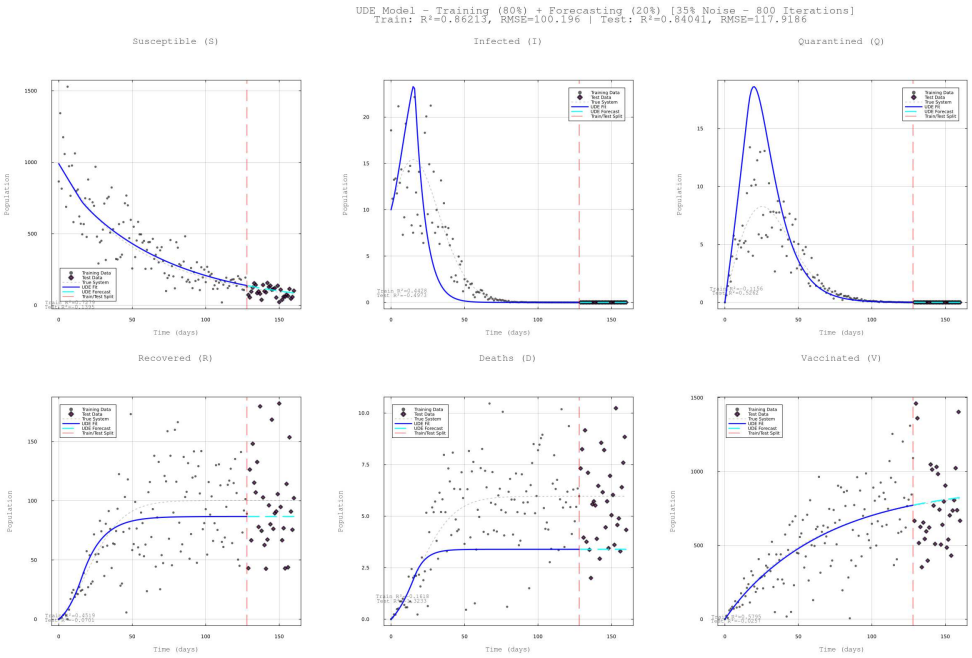

Figure 18: UDE(80/20 Split) for SIQRDV compartments with high noisy data

Figures[16–18] show the UDE model's forecasting results when trained on the first 128 days and tested on days 129–160. The Susceptible (S) compartment performs well during training and maintains reasonable forecasting across all noise levels. The Infected (I) compartment captures the epidemic peak accurately and shows good forecasting with noise-free and moderate noisy data, but forecasting becomes erratic with unrealistic peak formations under high noisy data. The Quarantined (Q) compartment demonstrates good performance with noise-free and 7% noisy data, but fails completely under high noise, generating unrealistic large peaks in forecasting. The Recovered (R) compartment shows excellent training fit but overpredicts in forecasting, with severity increasing from moderate overprediction at moderate noise to extreme overprediction at highy noisy situation. The Deaths (D) compartment performs well with noise-free data but overpredicts significantly with noisy data in forecasting. The Vaccinated (V) compartment demonstrates excellent performance with noise-free and moderate noisy data, maintaining accurate linear growth, though some deviation appears under high noise. Overall, the UDE model shows good performance with low noise but struggles with long-range forecasting under high noise conditions, though it avoids the negative value failures seen in Neural ODE.

3.4 Case 4: Training with 40% of the full available data and forecasting

The Neural ODEs and UDEs were trained on reduced datasets to evaluate forecasting under severe data limitation. The no-noise, moderate-noise, and high-noise datasets were trimmed to 40%, using the first 64 days for training and the remaining 96 days (days 65–160) for testing. Results are shown in Figures[19–21] for Neural ODE and Figures[22–24] for UDE.

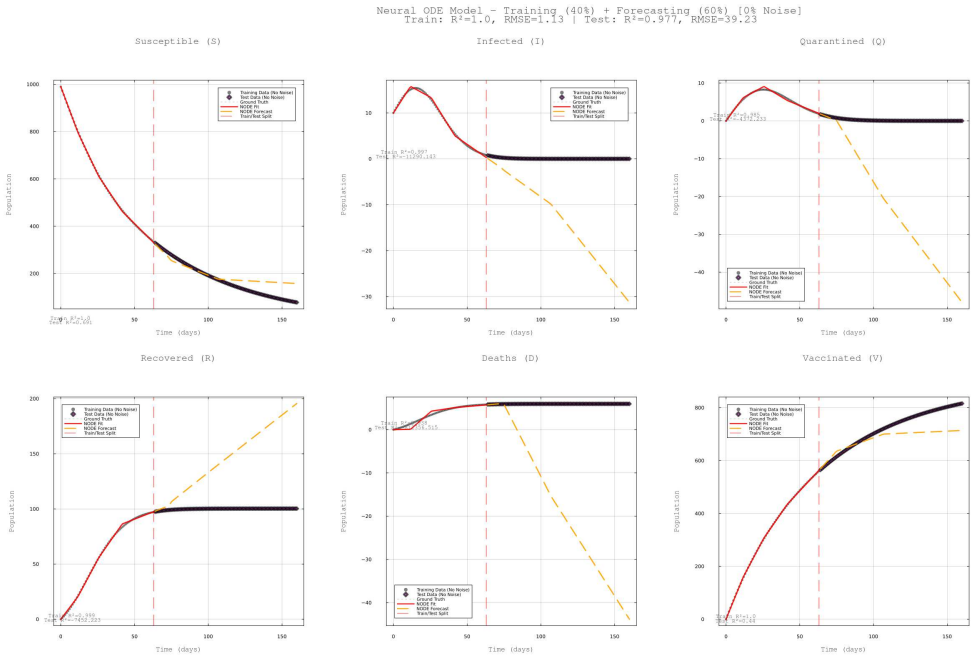

Figure 19: Neural ODE(40/60 Split) for SIQRDV compartments with noise-free data

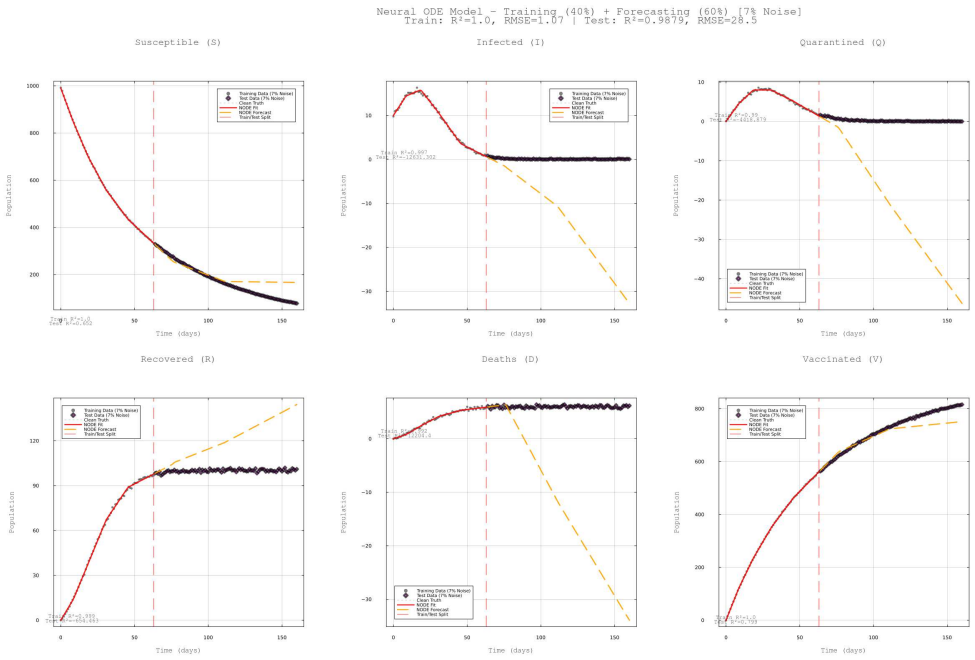

Figure 20: Neural ODE(40/60 Split) for SIQRDV compartments with high noisy data

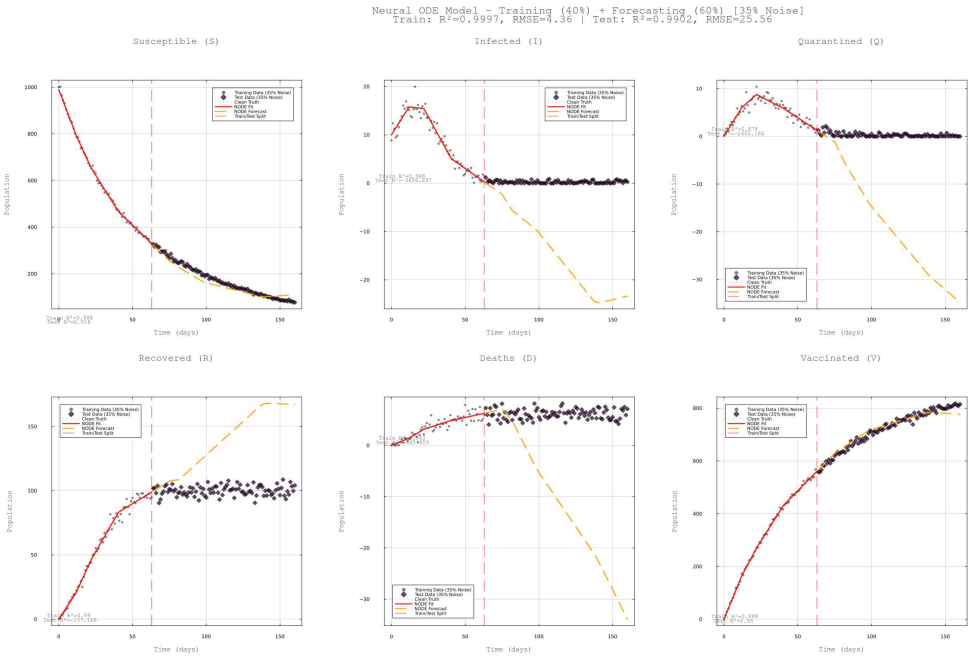

Figure 21: Neural ODE(40/60 Split) for SIQRDV compartments with moderate noisy data

Figures[19–21] show the Neural ODE model's forecasting results when trained on the first 64 days and tested on days 65–160. The Susceptible (S) compartment fits well during training but underestimates the population in forecasting across all noise levels. The Infected (I) compartment captures the epidemic peak during training but catastrophically fails in forecasting, dropping to physically impossible negative values in all scenarios. The Quarantined (Q) compartment shows similar catastrophic failure, producing unrealistic negative values in forecasting regardless of noise level. The Recovered (R) compartment demonstrates reasonable training fit and maintains plateau predictions for noise-free and 7% noisy data, but with 35% noise shows significant scatter and oscillations, along with incorrect continued growth predictions. The Deaths (D) compartment performs worst, dropping to negative values in forecasting across all scenarios, which is physically impossible for cumulative deaths. The Vaccinated (V) compartment shows relatively better performance, maintaining reasonable linear trend with noise-free data, though deviation increases with 7% and 35% noise. Overall, training on only 40% of data leads to severe Neural ODE failures in long-range forecasting, with most compartments producing physically impossible negative values.

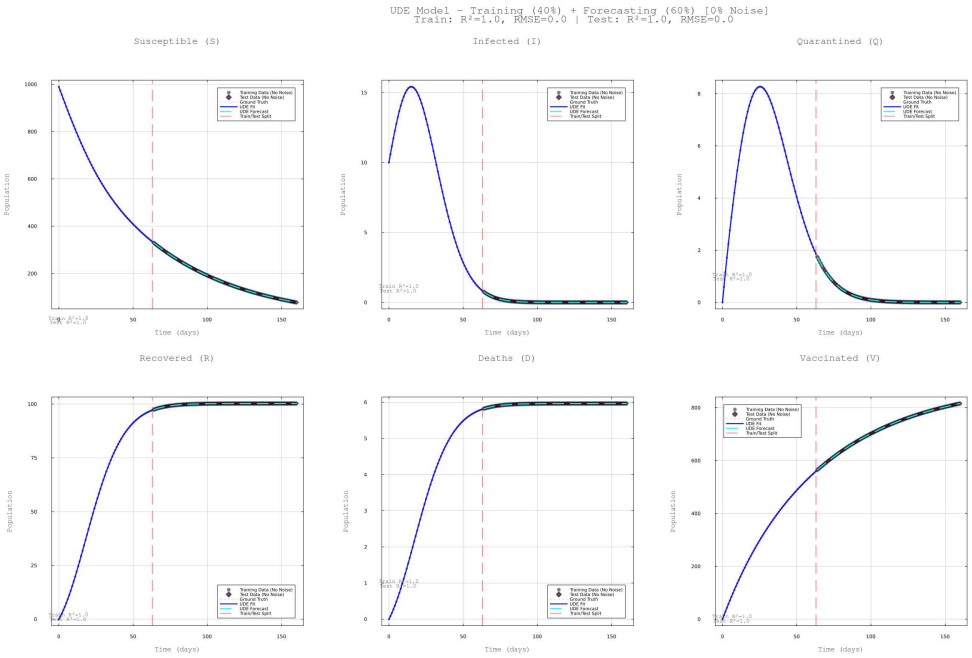

Figure 22: UDE(40/60 Split) for SIQRDV compartments with noise-free data

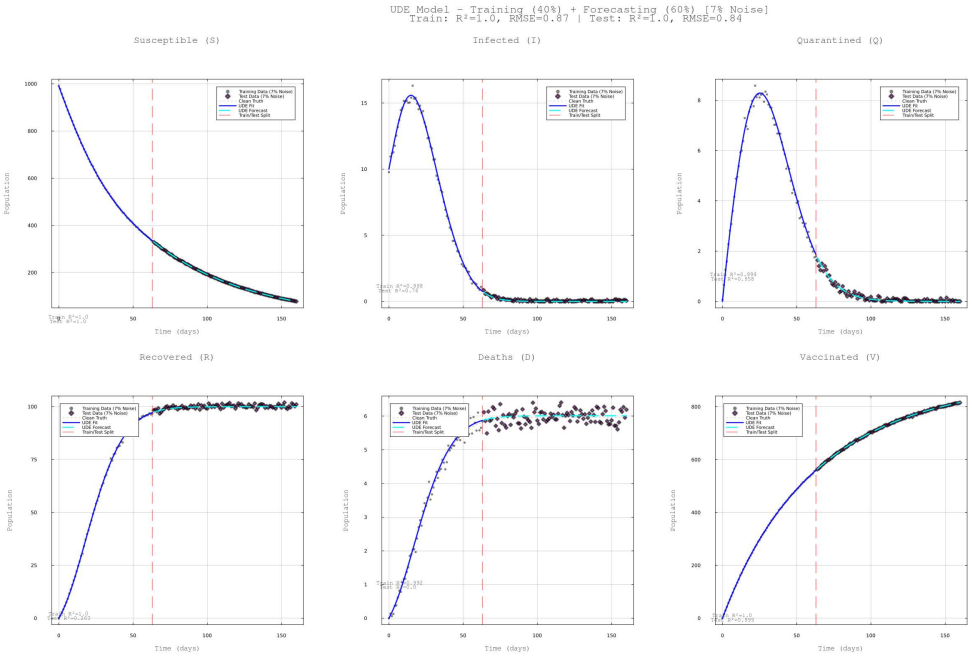

Figure 23: UDE(40/60 Split) for SIQRDV compartments with moderate noisy data

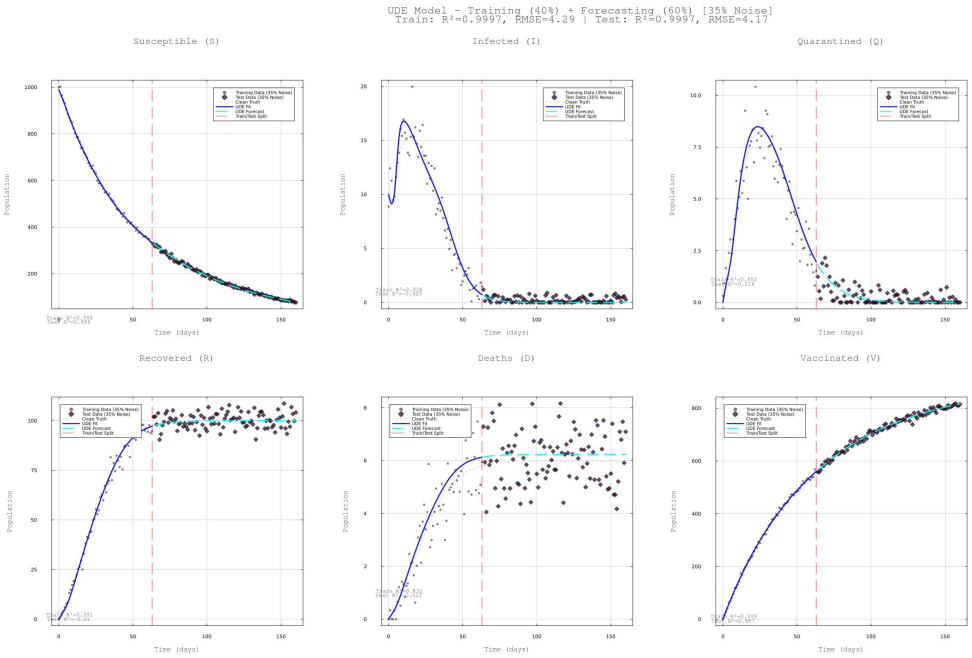

Figure 24: UDE(40/60 Split) for SIQRDV compartments with high noisy data

Figures[22–24] show the UDE model's forecasting results when trained on the first 64 days and tested on days 65–160. The Susceptible (S) compartment performs excellently across all noise levels, maintaining accurate exponential decline throughout training and forecasting regions. The Infected (I) compartment captures the epidemic peak accurately and demonstrates excellent forecasting with noise-free and 7% noisy data, though some scatter appears with 35% noise while maintaining the overall trend. The Quarantined (Q) compartment shows excellent performance with noise-free and 7% noisy data, but exhibits noticeable scatter in forecasting with 35% noise. The Recovered (R) compartment achieves accurate plateau predictions around 100 for noise-free data, shows oscillations with 7% noise in forecasting, and displays significant scatter with 35% noise though the general plateau trend is preserved. The Deaths (D) compartment maintains excellent monotonic growth with noise-free data, shows increased scatter with 7% noise while preserving the general trend, and exhibits heavy scatter with 35% noise but avoids the negative value failures seen in Neural ODE. The Vaccinated (V) compartment demonstrates outstanding performance across all scenarios, maintaining accurate linear growth with only minor scatter at higher noise levels. Overall, the UDE model significantly outperforms Neural ODE when trained on limited data, maintaining physically consistent predictions and avoiding catastrophic failures even under severe data limitation and high noise conditions.

3.5 Case 5: Training with 20% of the full available data and forecasting

The Neural ODEs and UDEs were trained with severely limited datasets to evaluate their forecasting performance under extreme data scarcity. The no-noise, moderate-noise, and high-noise training datasets were reduced to only 20% of the full domain, using compartment states and derivatives from the first 32 days for training. The remaining 128 days (days 33–160) served as the testing period to assess forecasting capability under extreme data limitation, where the forecasting horizon is four times longer than the training period. The results for the Neural ODE and UDE models are shown in Figures[25–27] and Figures[28–30], respectively.

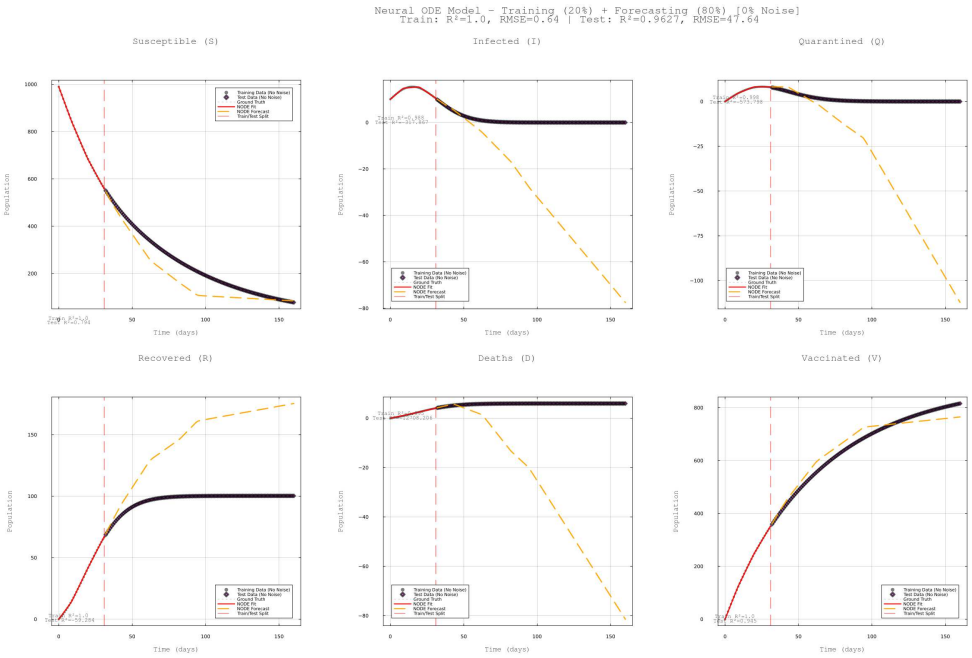

Figure 25: Neural ODE(20/80 Split) for SIQRDV compartments with noise-free data

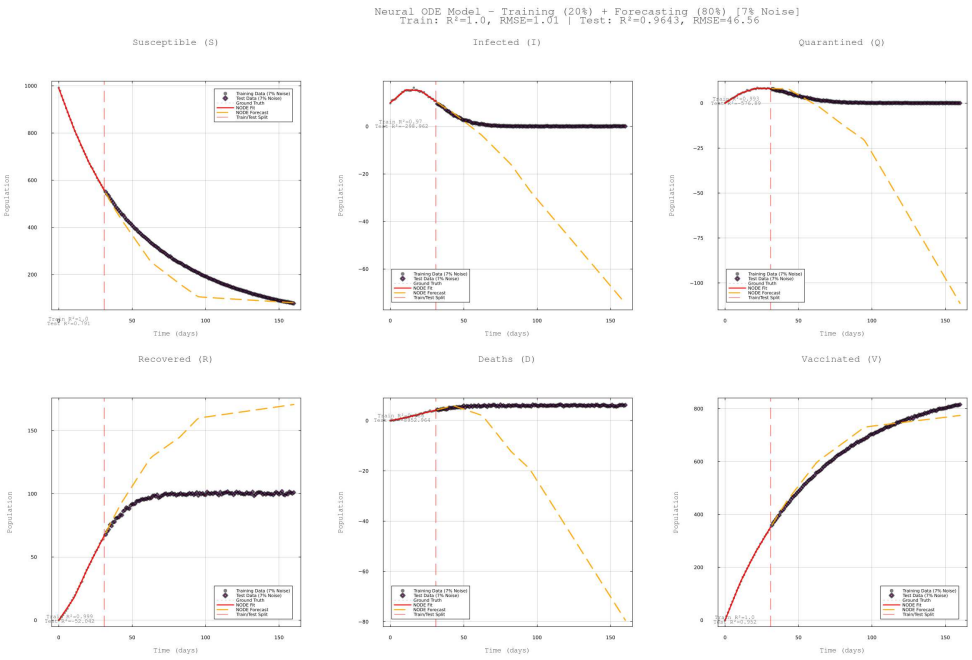

Figure 26: Neural ODE(20/80 Split) for SIQRDV compartments with moderate noisy data

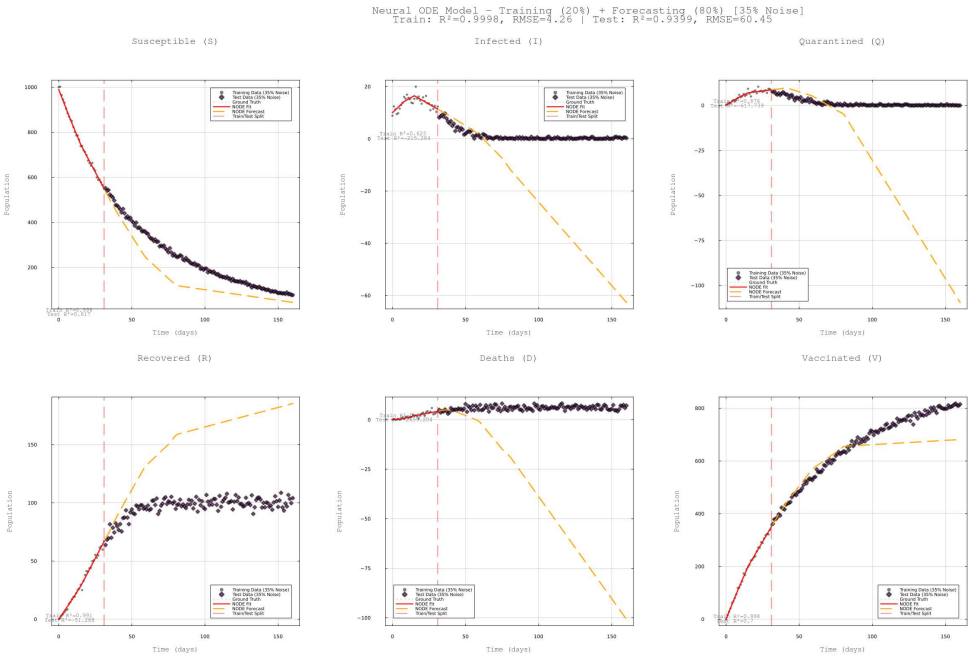

Figure 27: Neural ODE(20/80 Split) for SIQRDV compartments with high noisy data

Figures[25–27] show the Neural ODE model's forecasting results when trained on only the first 32 days and tested on days 33–160. The Susceptible (S) compartment fits reasonably during training but underestimates population values in forecasting across all noise levels, with deviation increasing as noise increases. The Infected (I) compartment captures the epidemic peak during training but catastrophically fails in forecasting, dropping to large negative values across all scenarios regardless of noise level. The Quarantined (Q) compartment shows similar catastrophic failure, producing extremely large negative values in forecasting for all noise conditions, demonstrating complete inability to extrapolate beyond the limited training period. The Recovered (R) compartment demonstrates training fit but incorrectly predicts continued growth in forecasting, with noise-free and 7% noisy data showing overprediction beyond realistic plateau values, and 35% noise exhibiting severe scatter while still overpredicting. The Deaths (D) compartment performs worst, dropping to negative values in forecasting across all scenarios, which is physically impossible for cumulative deaths. The Vaccinated (V) compartment shows relatively better performance, maintaining reasonable linear trend in forecasting though with increasing deviation at higher noise levels. Overall, training on only 20% of data leads to catastrophic Neural ODE failure in long-range forecasting, with most compartments producing physically impossible negative values, demonstrating fundamental limitations in extrapolating epidemic dynamics from severely limited training data.

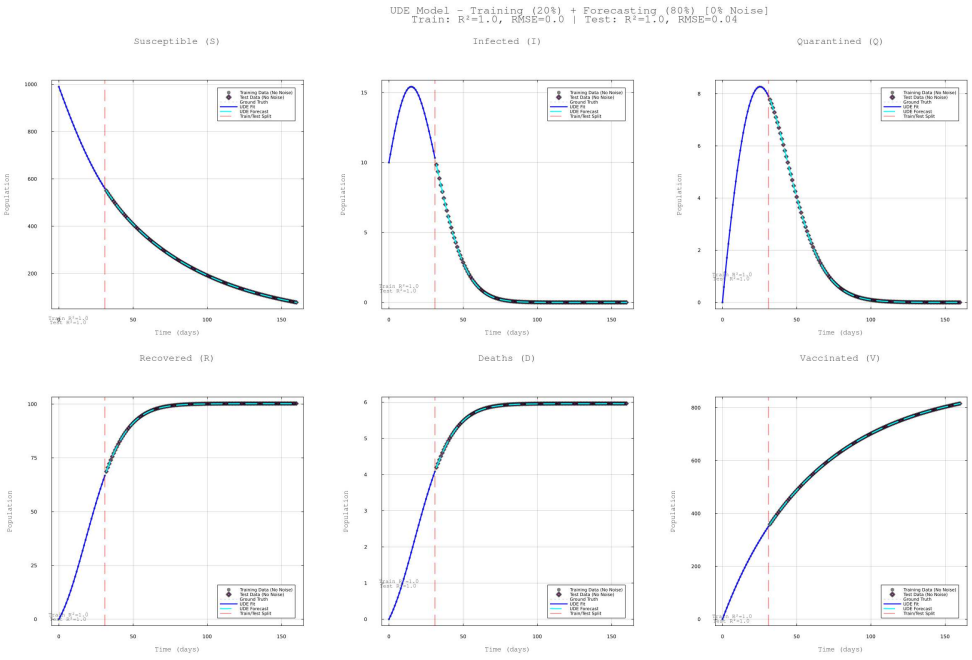

Figure 28: UDE(20/80 Split) for SIQRDV compartments with noise-free data

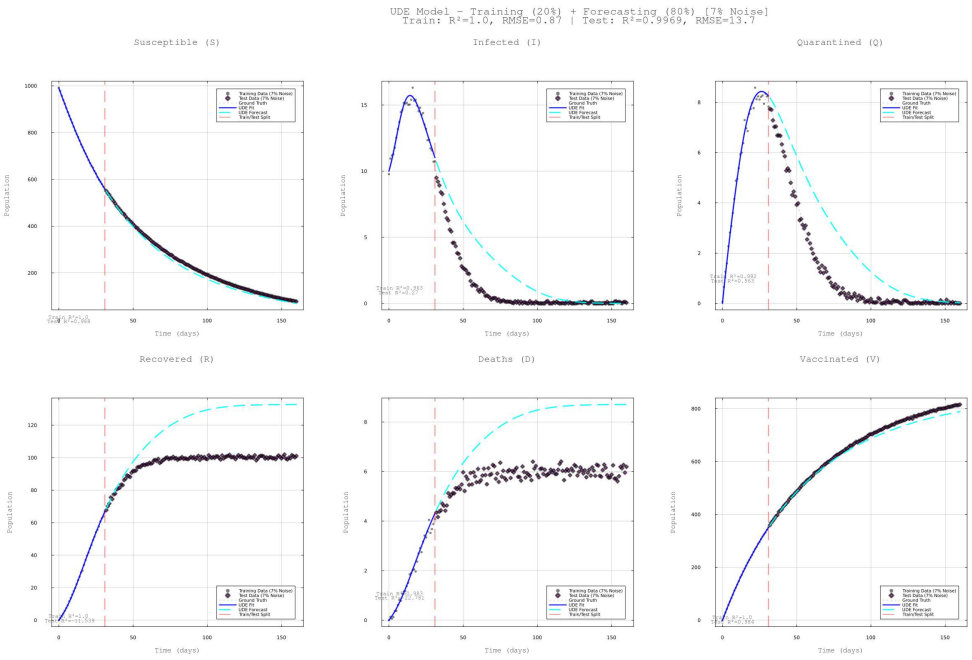

Figure 29: UDE(20/80 Split) for SIQRDV compartments with moderate noisy data

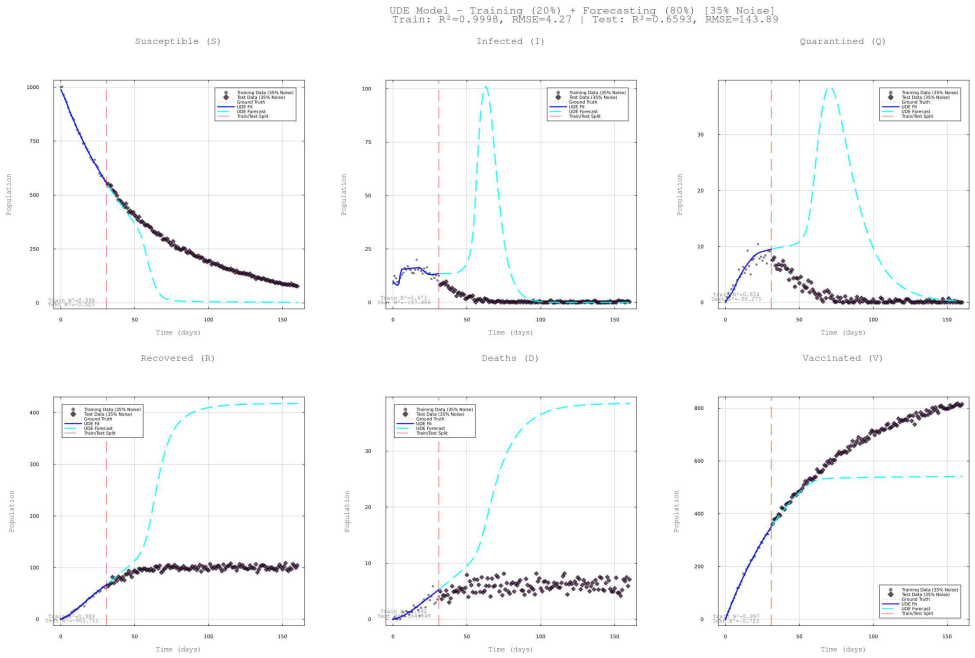

Figure 30: UDE(20/80 Split) for SIQRDV compartments with high noisy data

Figures[28–30] show the UDE model's forecasting results when trained on the first 64 days and tested on days 65–160. The Susceptible (S) compartment performs excellently across all noise levels, maintaining accurate exponential decline throughout training and forecasting regions. The Infected (I) compartment captures the epidemic peak accurately and demonstrates excellent forecasting with noise-free and 7% noisy data, though some scatter appears with 35% noise while maintaining the overall declining trend. The Quarantined (Q) compartment shows excellent performance with noise-free and 7% noisy data, but exhibits noticeable scatter in forecasting with 35% noise. The Recovered (R) compartment achieves accurate plateau predictions around 100 for noise-free data, shows oscillations with 7% noise in forecasting, and displays significant scatter with 35% noise though the general plateau trend is preserved. The Deaths (D) compartment maintains excellent monotonic growth with noise-free data, shows increased scatter with 7% noise while preserving the general trend, and exhibits heavy scatter with 35% noise but crucially avoids the negative value failures seen in Neural ODE. The Vaccinated (V) compartment demonstrates outstanding performance across all scenarios, maintaining accurate linear growth with only minor scatter at higher noise levels. Overall, the UDE model significantly outperforms Neural ODE when trained on limited data, maintaining physically consistent predictions and avoiding catastrophic failures even under severe data limitation and high noise conditions, demonstrating superior robustness for long-range epidemic forecasting.

3.6 Case 6: Training with 10% of the full available data and forecasting

The Neural ODEs and UDEs were trained with extremely limited datasets to evaluate forecasting under the most severe data scarcity. The no-noise, moderate-noise, and high-noise datasets were reduced to 10%, using only the first 16 days for training and the remaining 144 days for testing. Results are shown in Figures[31–33] for Neural ODE and Figures[34–36] for UDE.

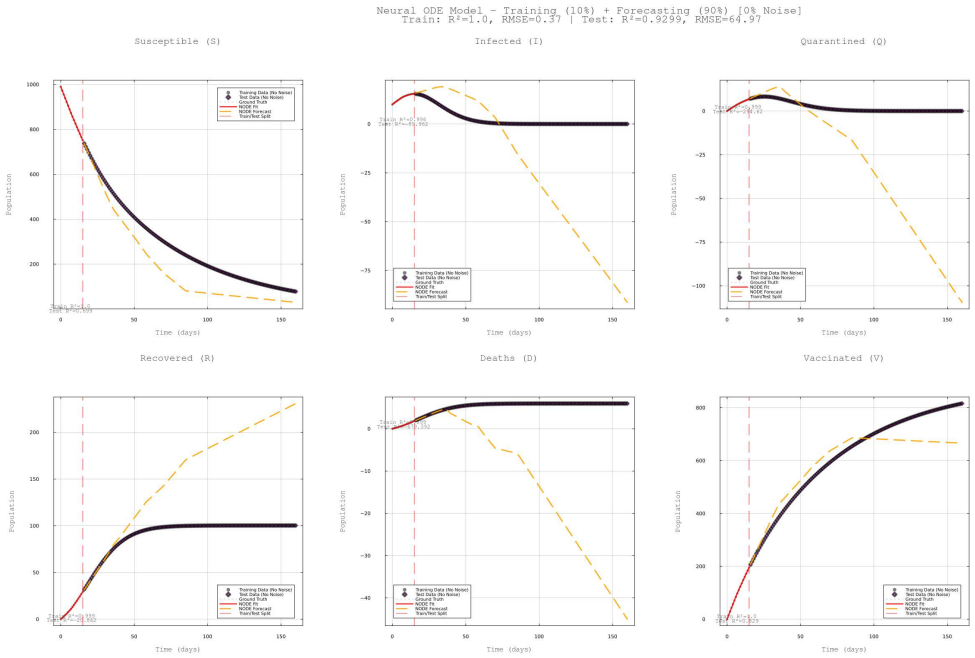

Figure 31: Neural ODE(10/90 Split) for SIQRDV compartments with noise-free data

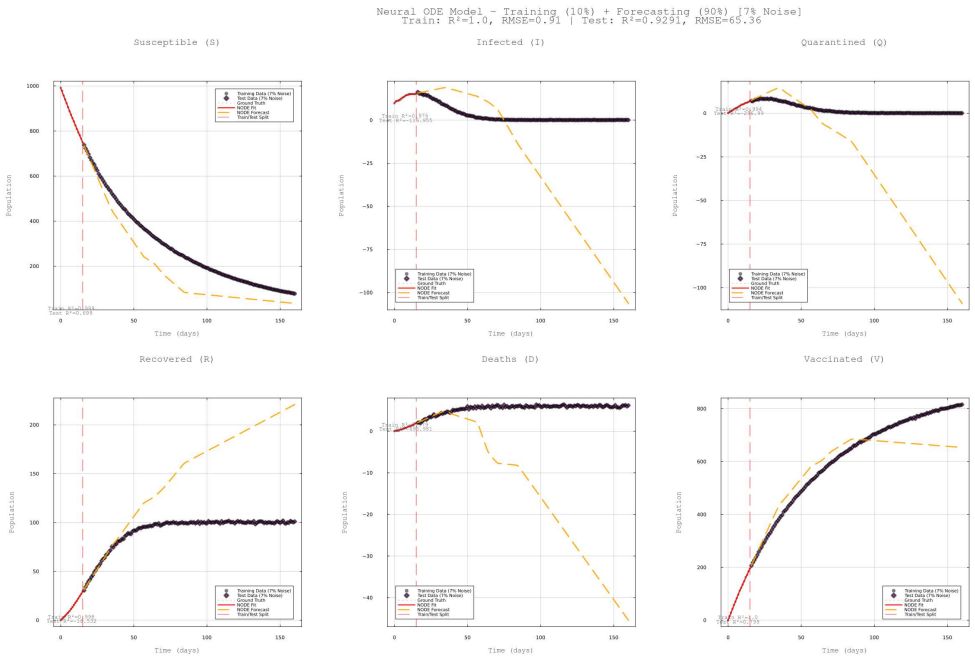

Figure 32: Neural ODE(10/90 Split) for SIQRDV compartments with moderate noisy data

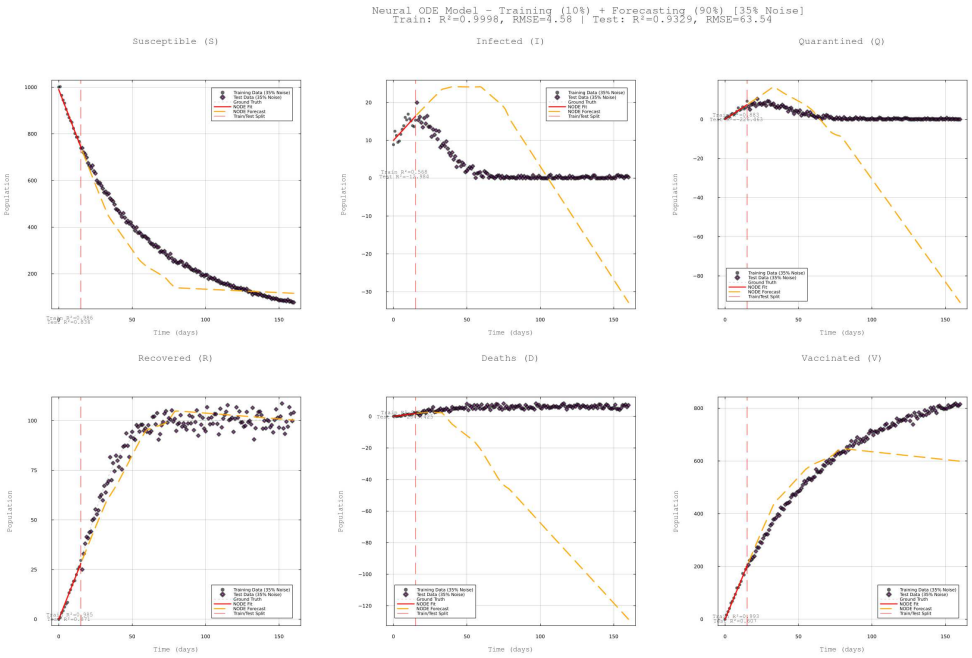

Figure 33: Neural ODE(10/90 Split) for SIQRDV compartments with high noisy data

Figures[31–33] show the Neural ODE model's forecasting results when trained on only the first 16 days and tested on days 17–160. The Susceptible (S) compartment shows reasonable fit during the brief training period but significantly underestimates population values throughout the extended forecasting region across all noise levels. The Infected (I) compartment captures the initial rise during training but catastrophically fails in forecasting, dropping to negative values with noise-free and 7% noisy data, and with 35% noise produces erratic predictions including unrealistic peak formations. The Quarantined (Q) compartment shows catastrophic failure across all scenarios, dropping to large negative values in forecasting regardless of noise level, demonstrating complete inability to extrapolate from such limited training data. The Recovered (R) compartment fits the initial growth during training but fails in forecasting, plateauing prematurely around 100 for noise-free and 7% noisy data, and exhibiting massive scatter with wild oscillations for 35% noise. The Deaths (D) compartment demonstrates the most severe failure, dropping to large negative values in forecasting across all noise conditions, which is physically impossible for cumulative deaths. The Vaccinated (V) compartment shows relatively better performance, maintaining reasonable linear trend with noise-free and 7% noisy data though with some deviation, while 35% noise shows increased scatter but preserves the general trend. Overall, training on only 10% of data leads to complete Neural ODE failure, with most compartments producing physically impossible negative values and demonstrating fundamental inability to perform long-range forecasting from extremely limited training data, highlighting the critical importance of sufficient training data for epidemic modeling.

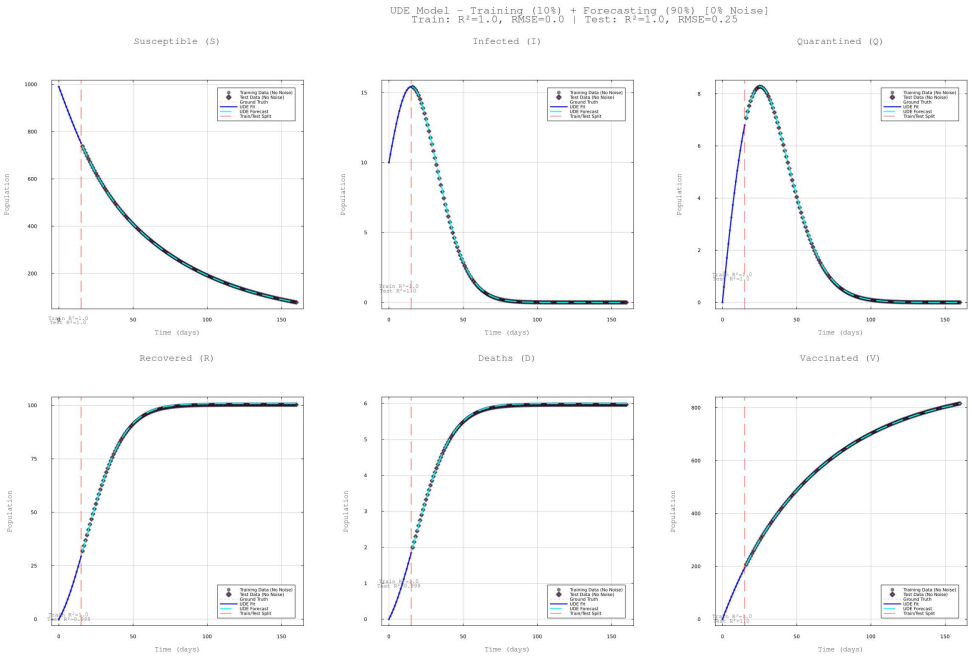

Figure 34: UDE(10/90 Split) for SIQRDV compartments with noise-free data

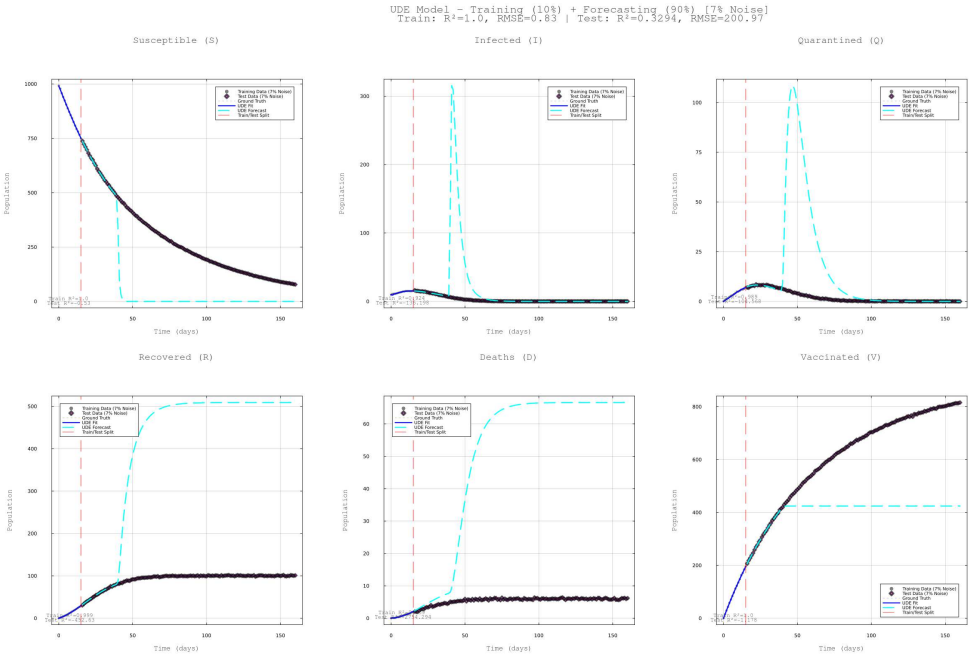

Figure 35: UDE(10/90 Split) for SIQRDV compartments with moderate noisy data

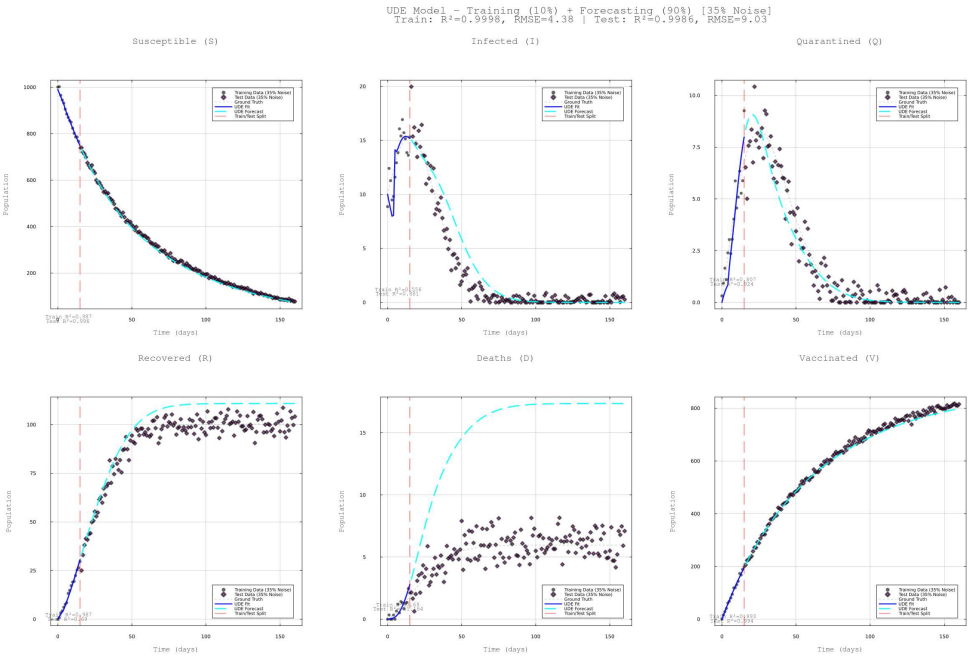

Figure 36: UDE(10/90 Split) for SIQRDV compartments with high noisy data

Figures[34–36] show the UDE model's forecasting results when trained on only the first 16 days and tested on days 17–160. The Susceptible (S) compartment performs well across all noise levels. The Infected (I) compartment achieves perfect forecasting with noise-free data but catastrophically fails with moderate noise, producing unrealistic explosive peak predictions, while maintaining reasonable trends with high noise despite scatter. The Quarantined (Q) compartment shows similar behavior, demonstrating excellent performance with clean data, catastrophic failure with moderate noise through massive unrealistic peaks, and heavy scatter but physically reasonable bounds with high noise. The Recovered (R) and Deaths (D) compartments achieve perfect predictions with noise-free data but severely overpredict with moderate noise, while exhibiting scatter but physically reasonable values with high noise. The Vaccinated (V) compartment demonstrates excellent performance across all scenarios. Remarkably, the UDE model exhibits worse performance with moderate noise than with high noise, producing catastrophically unrealistic explosive growth predictions at moderate noise levels while maintaining physically consistent bounds at high noise levels, suggesting complex interactions between limited training data and noise in the UDE learning process.

Both models achieve excellent performance ($R^2 > 0.99$) with sufficient training data (80% or more), but diverge dramatically under data scarcity. The Neural ODE fails catastrophically below 60% training data across all noise levels, producing physically impossible negative values, while the UDE maintains reliable forecasting down to 40% training data (or 10% for noise-free conditions). Table 3 summarizes the performance metrics and breakdown thresholds across all six training scenarios.

Table 3: Forecasting breakdown analysis for Neural ODE and UDE models

Metric	Neural ODE	UDE
Training with full dataset (160 days)		
Training R^2 (0% noise)	0.9999	1.0
Training RMSE (0% noise)	2.97	0.0
Training R^2 (35% noise)	0.9993	0.9997
Training RMSE (35% noise)	6.87	4.23
Forecasting performance		
Best forecasting scenario	90/10 split	All scenarios $\geq 80\%$
Best forecasting R^2	0.9996 (0% noise)	1.0 (multiple cases)
Forecasting breakdown	$< 60\%$ training data (all noise levels)	$< 20\%$ (0% noise) $< 40\%$ (with noise)
Failure characteristics		
Failure mode	Negative values (I, Q, D compartments)	Unrealistic peaks (only at 10% + 7% noise)
Physical consistency	Lost below 60% data	Maintained down to 20%
Efficiency		
Training time (typical)	180–360 seconds	200–230 seconds
Minimum training data for reliable forecasting	60% of domain	20–40% of domain (noise dependent)

4 Discussion and Conclusion

We successfully modeled SIQRDV epidemic dynamics using Neural ODE and UDE approaches across six training scenarios with varying data availability (10–100%) and noise levels (0%, 7%, 35%). Both models effectively learned epidemic patterns when sufficient data was available, but their performance diverged dramatically under data scarcity.

The Neural ODE achieved excellent results ($R^2 > 0.99$) when trained with 80% or more data across all noise levels. However, it catastrophically failed below 60% training data, producing physically impossible negative values for Infected, Quarantined, and Deaths compartments. This breakdown occurred regardless of noise level, indicating that Neural ODEs require substantial data to maintain reliable forecasting despite offering model-free flexibility.

The UDE model demonstrated superior data efficiency, maintaining reliable forecasting with only 40% training data for all noise conditions, and performing well with just 10% for noise-free scenarios. The UDE consistently preserved physical consistency, avoiding negative compartment values and achieving perfect R^2 scores (1.0) in multiple cases. This represents a 3–6$\times$ improvement in data efficiency compared to Neural ODE.

The critical difference between the two approaches emerges under data scarcity. Neural ODEs require minimum 60% of the epidemic timeline for reliable forecasting, making them unsuitable for early outbreak scenarios. UDEs maintain accuracy with 40% of data while preserving epidemiological realism, making them ideal for early-stage outbreak forecasting where data is limited. However, an unexpected finding revealed that UDE performance with extreme data scarcity (10% training) showed worse results with moderate noise (7%) than high noise (35%), producing unrealistic explosive peaks at moderate noise levels while maintaining physically consistent predictions at high noise levels.

The UDE's superior data efficiency is particularly valuable in public health applications where early outbreak predictions must be made with incomplete data. During disease outbreaks, comprehensive data collection takes time, yet intervention decisions cannot wait. The UDE's ability to provide reliable forecasts with 40% of data enables earlier public health responses. Neural ODEs, while offering model-free flexibility, are better suited for situations with abundant data or when disease mechanisms are completely unknown.

Neural ODEs provide a black-box solution requiring no epidemiological knowledge but demand extensive data (60% minimum) and risk catastrophic failures producing physically impossible predictions. UDEs require incorporating known disease transmission principles but achieve reliable forecasting with significantly less data while maintaining physical consistency. The choice between approaches depends on data availability and prior knowledge of disease dynamics.

While both models effectively capture epidemic dynamics with sufficient data, challenges remain for long-range forecasting under extreme data scarcity. Future work should develop hybrid architectures combining strengths of both approaches, implement adaptive mechanisms adjusting to data availability, and incorporate epidemiological constraints preventing implausible predictions. Extensions to multi-strain diseases, age-structured populations, and spatial epidemic

spread would enhance practical utility for comprehensive public health decision-making. Understanding the complex interaction between data scarcity and noise levels that caused UDE's counterintuitive behavior at 10% training warrants further investigation to improve model reliability under extreme conditions.

In conclusion, the UDE model's 3–6$\times$ superior data efficiency and maintained physical consistency make it the preferred choice for early outbreak forecasting and data-scarce scenarios, while Neural ODEs remain valuable when disease mechanisms are completely unknown and abundant data is available. Both approaches contribute valuable tools to the scientific machine learning toolkit for epidemic modeling, with clear guidance on when each should be applied.

5 Hyperparameters for the training datasets

5.1 Case 1: Training with 100% of the available data

Neural ODE

Hyperparameter	**Values**	**Search Range**
tspan	$(0.0, 160.0)$	$(0.0, 80.0) - (0.0, 160.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	2.97	$(0, 10)$

Table 4: Neural ODE range of hyper-parameters on training data (no noise)

Hyperparameter	**Values**	**Search Range**
tspan	$(0.0, 160.0)$	$(0.0, 80.0) - (0.0, 160.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	4.45	$(0, 10)$

Table 5: Neural ODE range of hyper-parameters on training data (moderate noise)

Hyperparameter	**Values**	**Search Range**
tspan	$(0.0, 160.0)$	$(0.0, 80.0) - (0.0, 160.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	6.87	$(0, 10)$

Table 6: Neural ODE range of hyper-parameters on training data (high noise)

UDE

Hyperparameter	Values	Search Range
tspan	$(0.0, 160.0)$	$(0.0, 80.0) - (0.0, 160.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.0	$(0, 10)$

Table 7: UDE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 160.0)$	$(0.0, 80.0) - (0.0, 160.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.85	$(0, 10)$

Table 8: UDE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 160.0)$	$(0.0, 80.0) - (0.0, 160.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	4.23	$(0, 10)$

Table 9: UDE range of hyper-parameters on training data (high noise)

5.2 Case 2: Training with 90% of the available data

Neural ODE

Hyperparameter	Values	Search Range
tspan	$(0.0, 144.0)$	$(0.0, 72.0) - (0.0, 152.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	4.0801	$(0, 10)$

Table 10: Neural ODE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 144.0)$	$(0.0, 72.0) - (0.0, 152.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	2.91	$(0, 10)$

Table 11: Neural ODE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 144.0)$	$(0.0, 72.0) - (0.0, 152.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	6.22	$(0, 10)$

Table 12: Neural ODE range of hyper-parameters on training data (high noise)

UDE

Hyperparameter	Values	Search Range
tspan	$(0.0, 144.0)$	$(0.0, 72.0) - (0.0, 152.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.0004	$(0, 10)$

Table 13: UDE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 144.0)$	$(0.0, 72.0) - (0.0, 152.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.87	$(0, 10)$

Table 14: UDE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 144.0)$	$(0.0, 72.0) - (0.0, 152.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	4.34	$(0, 10)$

Table 15: UDE range of hyper-parameters on training data (high noise)

5.3 Case 3: Training with 80% of the available data

Neural ODE

Hyperparameter	Values	Search Range
tspan	$(0.0, 128.0)$	$(0.0, 64.0) - (0.0, 144.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	2.2624	$(0, 10)$

Table 16: Neural ODE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 128.0)$	$(0.0, 64.0) - (0.0, 144.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	19.807	$(0, 30)$

Table 17: Neural ODE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 128.0)$	$(0.0, 64.0) - (0.0, 144.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$24, 24$	$12, 18, 24, 32, 48, 64$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	98.3126	$(0, 150)$

Table 18: Neural ODE range of hyper-parameters on training data (high noise)

UDE

Hyperparameter	Values	Search Range
tspan	$(0.0, 128.0)$	$(0.0, 64.0) - (0.0, 144.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.0001	$(0, 10)$

Table 19: UDE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 128.0)$	$(0.0, 64.0) - (0.0, 144.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	19.936	$(0, 30)$

Table 20: UDE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 128.0)$	$(0.0, 64.0) - (0.0, 144.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $12, 8$ Kappa: 8	$8, 12, 16, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	100.196	$(0, 150)$

Table 21: UDE range of hyper-parameters on training data (high noise)

5.4 Case 4: Training with 40% of the available data

Neural ODE

Hyperparameter	Values	Search Range
tspan	$(0.0, 64.0)$	$(0.0, 32.0) - (0.0, 96.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	1.13	$(0, 10)$

Table 22: Neural ODE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 64.0)$	$(0.0, 32.0) - (0.0, 96.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	1.07	$(0, 10)$

Table 23: Neural ODE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 64.0)$	$(0.0, 32.0) - (0.0, 96.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	1.36	$(0, 10)$

Table 24: Neural ODE range of hyper-parameters on training data (high noise)

UDE

Hyperparameter	Values	Search Range
tspan	$(0.0, 64.0)$	$(0.0, 32.0) - (0.0, 96.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.0	$(0, 10)$

Table 25: UDE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 64.0)$	$(0.0, 32.0) - (0.0, 96.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.87	$(0, 10)$

Table 26: UDE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 64.0)$	$(0.0, 32.0) - (0.0, 96.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	4.17	$(0, 10)$

Table 27: UDE range of hyper-parameters on training data (high noise)

5.5 Case 5: Training with 20% of the available data

Neural ODE

Hyperparameter	Values	Search Range
tspan	$(0.0, 32.0)$	$(0.0, 16.0) - (0.0, 48.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.64	$(0, 10)$

Table 28: Neural ODE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 32.0)$	$(0.0, 16.0) - (0.0, 48.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	1.01	$(0, 10)$

Table 29: Neural ODE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 32.0)$	$(0.0, 16.0) - (0.0, 48.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	1.26	$(0, 10)$

Table 30: Neural ODE range of hyper-parameters on training data (high noise)

UDE

Hyperparameter	Values	Search Range
tspan	$(0.0, 32.0)$	$(0.0, 16.0) - (0.0, 48.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.0	$(0, 10)$

Table 31: UDE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 32.0)$	$(0.0, 16.0) - (0.0, 48.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.87	$(0, 10)$

Table 32: UDE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 32.0)$	$(0.0, 16.0) - (0.0, 48.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	4.27	$(0, 10)$

Table 33: UDE range of hyper-parameters on training data (high noise)

5.6 Case 6: Training with 10% of the available data

Neural ODE

Hyperparameter	Values	Search Range
tspan	$(0.0, 16.0)$	$(0.0, 8.0) - (0.0, 24.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.37	$(0, 10)$

Table 34: Neural ODE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 16.0)$	$(0.0, 8.0) - (0.0, 24.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.91	$(0, 10)$

Table 35: Neural ODE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 16.0)$	$(0.0, 8.0) - (0.0, 24.0)$
Activation Function	tanh	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.001	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	$18, 12$	$12, 15, 18, 24, 32, 48$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	1.58	$(0, 10)$

Table 36: Neural ODE range of hyper-parameters on training data (high noise)

UDE

Hyperparameter	Values	Search Range
tspan	$(0.0, 16.0)$	$(0.0, 8.0) - (0.0, 24.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.0	$(0, 10)$

Table 37: UDE range of hyper-parameters on training data (no noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 16.0)$	$(0.0, 8.0) - (0.0, 24.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	0.83	$(0, 10)$

Table 38: UDE range of hyper-parameters on training data (moderate noise)

Hyperparameter	Values	Search Range
tspan	$(0.0, 16.0)$	$(0.0, 8.0) - (0.0, 24.0)$
Activation Function	tanh, softplus, sigmoid	ReLU, tanh, sigmoid, softplus
Optimization Solver	Adam, BFGS	Adam, RAdam, BFGS
Learning Rate	Adam: 0.01 BFGS: 0.01	$0.001, 0.005, 0.01, 0.02, 0.05, 0.1$
Hidden units	Beta: $8, 6$ Kappa: 6	$6, 8, 12, 16, 24, 32$
Number of Epochs	Adam: 500 BFGS: 300	$200 - 1500$
Training Loss (RMSE)	4.38	$(0, 10)$

Table 39: UDE range of hyper-parameters on training data (high noise)

References

[1] [1] Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 6571–6583.

[2] [2] Rubanova, Y., Chen, R. T., and Duvenaud, D. K. (2019). Latent ordinary differential equations for irregularly-sampled time series. *Advances in Neural Information Processing Systems*, 32, 5320–5330.

[3] [3] Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., and Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*.

[4] [4] Dandekar, R., Rackauckas, C., and Barbastathis, G. (2020). A machine learning-aided global diagnostic and comparative tool to assess effect of quarantine control in COVID-19 spread. *Patterns*, 1(9), 100145.

[5] [5] Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review*, 42(4), 599–653.

[6] [6] Brauer, F., and Castillo-Chavez, C. (2008). *Mathematical models in population biology and epidemiology* (Vol. 1). New York: Springer.

[7] [7] Kermack, W. O., and McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A*, 115(772), 700–721.

[8] [8] Anderson, R. M., and May, R. M. (1991). *Infectious diseases of humans: dynamics and control*. Oxford University Press.

[9] [9] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440.

[10] [10] Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. (2020). Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*.

[11] [11] Wang, S., Teng, Y., and Perdikaris, P. (2020). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5), A3055–A3081.

[12] [12] Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1), 208–228.

[13] [13] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98.

[14] [14] Rackauckas, C., and Nie, Q. (2017). DifferentialEquations.jl–a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, 5(1), 15.

[15] [15] Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2), 281–305.

[16] [16] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2951–2959.

[17] [17] Chimmula, V. K. R., and Zhang, L. (2020). Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons and Fractals*, 135, 109864.

[18] [18] Zhan, C., Zheng, Y., Zhang, H., and Wen, Q. (2021). Random forest-based modeling and prediction of COVID-19 transmission. *Machine Learning with Applications*, 5, 100056.

[19] [19] Rodriguez, A., Tabassum, A., Cui, J., Xie, J., Ho, J., Agarwal, P., and Ramakrishnan, N. (2021). DeepCOVID: An operational deep learning-driven framework for explainable real-time COVID-19 forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17), 15393–15400.

[20] [20] Shaman, J., and Karspeck, A. (2012). Forecasting seasonal outbreaks of influenza. *Proceedings of the National Academy of Sciences*, 109(50), 20425–20430.

[21] [21] Yang, K. K., Wu, Z., and Arnold, F. H. (2019). Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8), 687–694.

[22] [22] Zhang, Z., Zhou, L., Zhao, X., Wang, G., Su, Y., Meng, C., and Li, S. (2020). A deep learning framework for gene ontology annotations with sequence-and network-based information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(6), 2208–2217.

[23] [23] Ren, K., Liao, Q., Fardad, M., and Barbastathis, G. (2020). Physics-based deep learning for fiber-optic communication systems. *IEEE Journal of Selected Topics in Quantum Electronics*, 27(2), 1–12.

[24] [24] Hughes, T. W., Minkov, M., Shi, Y., and Fan, S. (2018). Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica*, 5(7), 864–871.

[25] [25] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707.

[26] [26] Brunton, S. L., Noack, B. R., and Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52, 477–508.

[27] [27] Zhang, Y., Ni, Q., and Wang, D. (2019). Stock market prediction based on generative adversarial network. *Procedia Computer Science*, 147, 400–406.

[28] [28] Li, R. Y., Di Felice, R., Rohs, R., and Lidar, D. A. (2018). Quantum annealing versus classical machine learning applied to a simplified computational biology problem. *npj Quantum Information*, 4(1), 1–10.

[29] [29] Fang, Z., Zhou, Y., Li, S., Wang, W., and Ying, X. (2020). Neural network modeling of gene regulatory networks. *Bioinformatics*, 36(22-23), 5387–5394.

[30] [30] Baker, N., Alexander, F., Bremer, T., Hagberg, A., Kevrekidis, Y., Najm, H., and Rosenbloom, D. (2019). Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. *Technical Report, USDOE Office of Science*.

[31] [31] Pontryagin, L. S. (1987). *Mathematical theory of optimal processes*. CRC Press.

[32] [32] Kidger, P., Morrill, J., Foster, J., and Lyons, T. (2020). Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33, 6696–6707.

[33] [33] Long, Z., Lu, Y., Ma, X., and Dong, B. (2018). PDE-Net: Learning PDEs from data. *International Conference on Machine Learning*, 3208–3216.

[34] [34] Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 15379–15389.

[35] [35] Wang, S., Yu, X., and Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449, 110768.

[36] [36] Chowell, G., Hengartner, N. W., Castillo-Chavez, C., Fenimore, P. W., and Hyman, J. M. (2004). The basic reproductive number of Ebola and the effects of public health measures: the cases of Congo and Uganda. *Journal of Theoretical Biology*, 229(1), 119–126.

[37] [37] Gumel, A. B., Ruan, S., Day, T., Watmough, J., Brauer, F., Van den Driessche, P., and Sahai, B. M. (2004). Modelling strategies for controlling SARS outbreaks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 271(1554), 2223–2232.

[38] [38] Kribs-Zaleta, C. M., and Velasco-Hernandez, J. X. (2000). A simple vaccination model with multiple endemic states. *Mathematical Biosciences*, 164(2), 183–201.

[39] [39] Shim, E., Feng, Z., Martcheva, M., and Castillo-Chavez, C. (2006). An age-structured epidemic model of rotavirus with vaccination. *Journal of Mathematical Biology*, 53(4), 719–746.

[40] [40] Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2623–2631.

[41] [41] Diekmann, O., Heesterbeek, H., and Britton, T. (2012). *Mathematical tools for understanding infectious disease dynamics*. Princeton University Press.

[42] [42] Keeling, M. J., and Rohani, P. (2008). *Modeling infectious diseases in humans and animals*. Princeton University Press.

[43] [43] Dietz, K. (1993). The estimation of the basic reproduction number for infectious diseases. *Statistical Methods in Medical Research*, 2(1), 23–41.

[44] [44] Wallinga, J., and Lipsitch, M. (2007). How generation intervals shape the relationship between growth rates and reproductive numbers. *Proceedings of the Royal Society B: Biological Sciences*, 274(1609), 599–604.

[45] [45] Garske, T., Legrand, J., Donnelly, C. A., Ward, H., Cauchemez, S., Fraser, C., and Ferguson, N. M. (2009). Assessing the severity of the novel influenza A/H1N1 pandemic. *British Medical Journal*, 339, b2840.

[46] [46] Linton, N. M., Kobayashi, T., Yang, Y., Hayashi, K., Akhmetzhanov, A. R., Jung, S. M., and Nishiura, H. (2020). Incubation period and other epidemiological characteristics of 2019 novel coronavirus infections with right truncation: a statistical analysis of publicly available case data. *Journal of Clinical Medicine*, 9(2), 538.

[47] [47] Tsitouras, C. (2011). Runge-Kutta pairs of order 5(4) satisfying only the first column simplifying assumption. *Computers and Mathematics with Applications*, 62(2), 770–775.

[48] [48] Shampine, L. F., and Reichelt, M. W. (1997). The MATLAB ODE suite. *SIAM Journal on Scientific Computing*, 18(1), 1–22.

[49] [49] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

[50] [50] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

[51] [51] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366.

[52] [52] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303–314.

[53] [53] Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[54] [54] Nocedal, J., and Wright, S. (2006). *Numerical optimization*. Springer Science and Business Media.

[55] [55] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.