A Details for Neural Manifold Operator

A.1 Model Details

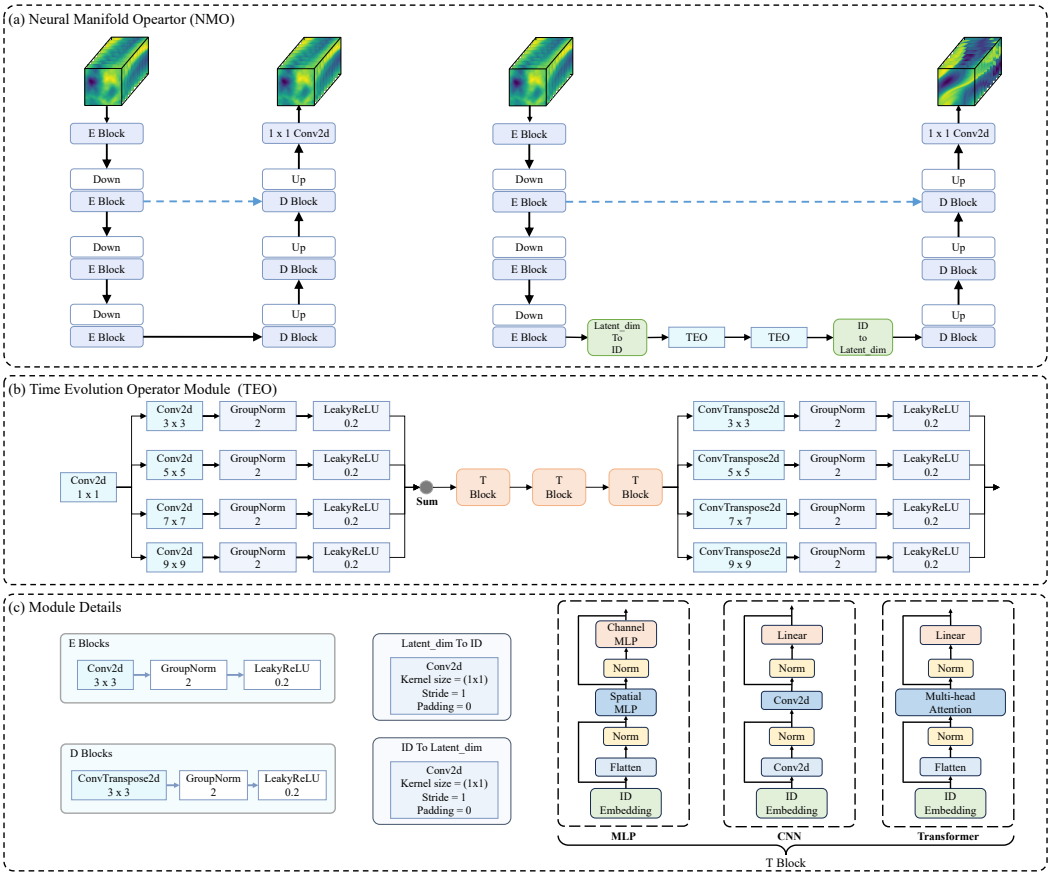

Figure 6: The architectural details of NMO.

Figure 6 shows the detailed structure of NMO. Figure 6(a) shows the overall network structure of NMO in the pretraining and training process. In the pretraining process, the model structure consists of an encoder and a decoder. In the training process, the model structure consists of an encoder, a decoder, and a time evolution operator. Figure 6(c) depicts the design of the E Block, which primarily consists of 3x3 Conv2d convolution kernels, complemented by GroupNorm for normalization and LeakyReLU as the activation function. The core function of the E Block is to downsample while elevating the feature dimension of the observed data. In contrast, the details of the D Block, also showcased in Figure 6(c), focus on upsampling and restoring latent features to the target dimension. The D Block is mainly composed of 3x3 ConvTranspose2d, GroupNorm, and LeakyReLU.

There are two linear projecting layers in the training process of the model, which aim to project the latent space into the intrinsic dimension or from the intrinsic dimension into the original latent space. The aim of the design for linear projecting layers is to learn the underlying operators by the Time Evolution Operator (TEO) module in the intrinsic dimension.

Figure 6(b) illustrates the design of the TEO, which is similar to U-Net. Initially, downsampling is performed using multiscale convolutions with kernel sizes of $\{3, 5, 7, 11\}$. Subsequently, the data passes through the T Block and is then upsampled using multiscale transpose convolution layers. The details of the three implementations based on MLP, CNN, and Transformer for T Block are shown in Figure 6(c).

A.2 ALGORITHM DETAILS

Algorithm 1 Neural Manifold Operator

Problem Formulation:
Require: Physics variable X such that $X \in D$ where $D \subset \mathbb{R}^d$
Ensure: The physical dynamics represent the temporal evolution of X
Let the dynamical system be represented as $\frac{dX}{dt} = f(X)$
The discrete forms is $X_{t+\varepsilon} = \mathbf{F}(X_t, t)$
Aim: Create a parameterized structure G_θ, to approximate the flow mapping of X based on finite data.
Architecture Formulation:
Components:
- Encoder $\mathcal{P}$: Maps from $\mathbb{R}^d$ to $\mathbb{R}^{d_L}$
- Decoder $\mathcal{Q}$: Maps from $\mathbb{R}^{d_L}$ to $\mathbb{R}^d$
- Time Evolution Operator $\mathcal{K}$: Acts on v_t in $\mathbb{R}^{d_L}$ producing an evolved state
In the pretraining process: Encoder and Decoder are self-supervised trained by reconstruction constraint L_{pred}
When the reconstruction training reaches convergence, projecting X to a latent variable V in a higher-dimensional latent space $\mathbb{R}^{d_L}$
Computing the minimum dimensional submanifold representation of the latent variable V and its dimension m is considered as the intrinsic dimension d_{id}
Linear projecting $\mathcal{L}$ from the latent space to the compact space
Compose the architecture as: $G_\theta = \mathcal{Q} \circ \mathcal{L}^{-1} \circ \mathcal{K}_{\text{id}} \circ \mathcal{L} \circ \mathcal{P}$
Projection Network in Latent Space:
Objective: Facilitate mappings between the physical and latent spaces
Components: Encoder $\mathcal{P}$ and Decoder $\mathcal{Q}$
Optimize by minimizing the reconstruction loss: $L_{rec} = ||\mathcal{Q} \circ \mathcal{P}(X_t) - X_t||_2^2$
Aim: Determine the minimal dimension representation of the manifold $\mathcal{M}$
Calculate the intrinsic dimension d_{id} by maximum likelihood estimation.
Formulation of the Time Evolution Operator:
Given the intrinsic dimension d_{id}, linearly project $\mathcal{V}_t$ to $\mathcal{W}_t$
Decompose $\mathcal{K}_{\text{id}}$ into: $\mathcal{K}_u \circ \mathcal{K}_e \circ \mathcal{K}_d$
Implement the evolution capture operator $\mathcal{K}_e$ using MLP, CNN, or Transformer
Objective: Minimize prediction loss $L_{pred} = ||G_\theta(X_t) - X_{t+\varepsilon}||_2^2$

Algorithm 2 Intrinsic Dimension Calculation

Aims: Calculating the minimal manifold representation of the latent variable $\mathcal{V}_t$ to approximately calculate the intrinsic dimension d_{id}.
Nearest Neighbor Computation:
function KNN($\mathcal{V}_t$, n_neighbors, n_jobs)
    Convert $\mathcal{V}_t$ to a Numpy array and move to CPU
    Initialize a nearest neighbor object with given neighbors and jobs
    return distances and indices of nearest neighbors
end function
Calculate Intrinsic Dimension by Maximum Likelihood Estimate:
function MLE($\mathcal{V}_t$, dists, k)
    Compute matrix A using distances
    Compute intrinsic dimension from A
    return the expected value of dimension m
end function
Algorithm Flow:
function ESTIMATE_DIMENSION(latent_embedding, k)
    Extract shape of latent_embedding
    Reshape and rearrange latent_embedding for KNN
    Get distances using KNN
    Maximum Likelihood Estimate for calculating intrinsic dimension
    return estimated dimension m
end function

B Details for Benchmarks and Experiment

We have summarized benchmark configurations in Table 3. All experiments are conducted on a single NVIDIA A100 40GB GPU.

Table 3: Detailed information for benchmarks. N_{tr}, N_{ev} and N_{te} represent the number of instances in the training, evaluation, and test sets, while I_l and O_l denote the lengths of the input and prediction sequences, respectively. N_v denotes the number of variables.

Dataset	N_{tr}	N_{ev}	N_{te}	N_v	Resolution	I_l	O_l	Interval
SEVIR	35,718	4465	4465	1	(192, 192)	13	12	5 mins
Kuroshio	1660	208	208	2	(128, 128)	5	15	1 day
Typhoon	4158	445	500	3	(512, 512)	6	36	1 hour
Navier-Stokes equation	1000	100	100	1	(64, 64)	10	10	1 step
Shallow-Water equations	1000	100	100	1	(128, 128)	10	90	1 step
Rayleigh-Bénard convection	1544	193	193	2	(64, 448)	10	60	1 step
Diffusion-Reaction equation	1000	100	100	2	(128, 128)	50	50	1 step

B.1 Real-World Scenarios

B.1.1 SEVIR

Data Description. SEVIR dataset is a standard benchmark that includes various types of temporally and spatially aligned image sequences for weather radar and satellite. The publicly available dataset has attracted widespread attention from the weather and climate research community. We choose Infrared Satellite imagery by the sensor named GOES-16 C09 with 6.9 μm infrared channels. The physical variable inverted by satellites in the particular wavelength range is mid-level water vapor, which is highly correlated with precipitation.

Experiment Settings. In the SEVIR scenario, the configuration of NMO consists of a 4-layer encoder, a 4-layer decoder, and a 6-layer time evolution operator. In the pretraining process, we set 256 as the dimension of latent space and employ an early stopping strategy. The pretraining process stops when the reconstruction error converges to 0.002786. The intrinsic dimension is 6. In the training process, the training epoch is 100. The parameters of the encoder and decoder are frozen. We set MSE Loss and Adam optimizer with a learning rate of 0.001 in the pretraining process and training process. The time steps for the input and output tensors are 13 and 12, respectively. Each time step interval is 5 minutes. The maximum prediction length through a single forward process of the model is 1 hour.

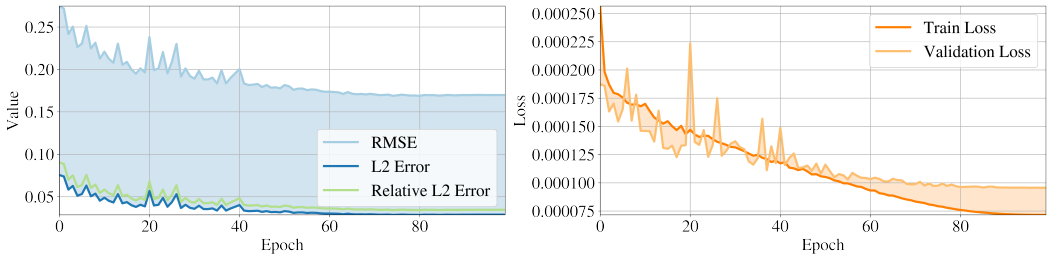

Figure 7: Training details for the SEVIR scenario. The left picture shows the variation of RMSE, L2 error and relative L2 error with iterations. The right picture shows the change in MSE loss for both the training and validation dataset with respect to iterations.

Experiment Result. In the SEVIR scenario, the training details of the NMO are shown in Figure. 7. In the training process, NMO typically converges after approximately 40 epochs, reaching an L2 error of 0.02885460 and a Relative L2 error of 0.03457794. Following the settings mentioned above, we utilize the officially recommended visualization library for visual presentation. As depicted in Figure 8, the first row displays the initial conditions, comprising the 65-minute duration of historical

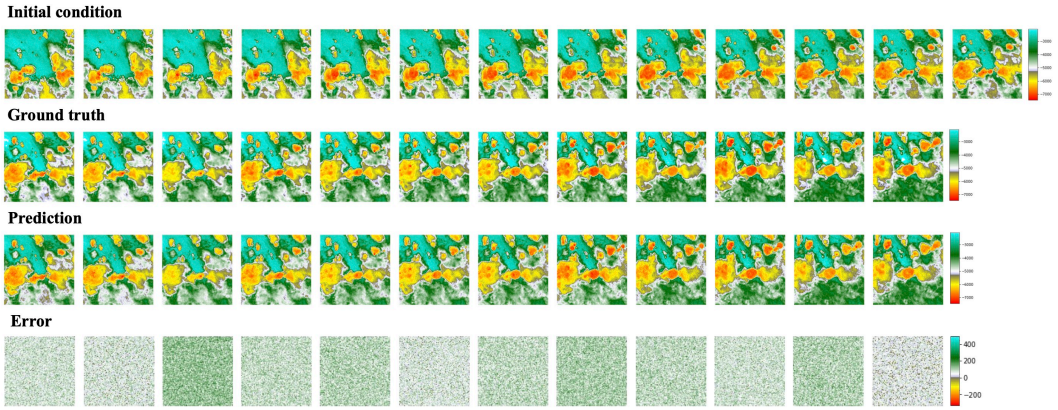

Figure 8: Visualization of the SEVIR scenario.

observation data. The second row showcases the 60-minute duration of ground truth data, while the third row illustrates the prediction results of NMO for the future 60 minutes. The fourth row visualizes the prediction error between the predictions and the ground truth.

B.1.2 KUROSHIO

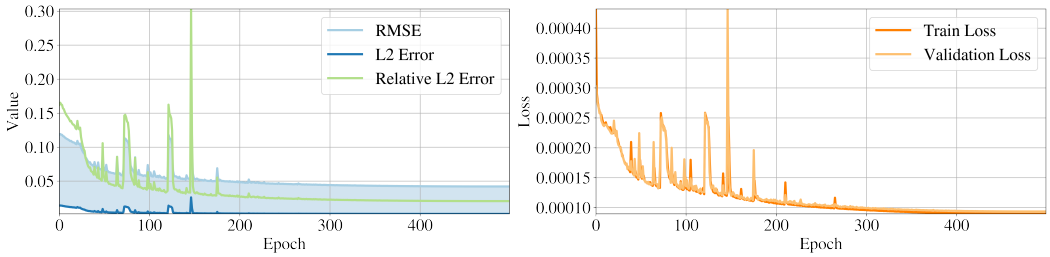

Figure 9: Training details for the Kuroshio scenario. The left picture shows the variation of RMSE, L2 error, and relative L2 error with iterations. The right picture shows the change in MSE loss for both the training and validation dataset with respect to iterations.

Data Description. We utilize the sea surface stream velocity data from the Copernicus Marine Environment Monitoring Service (CMEMS), which is a daily global satellite sea level product with a resolution of 0.25×0.25 degrees. Specifically, we select data from the Kuroshio region (10-42°N, 123-155°E) and the Gulf Stream region (20-52°N, 33-65°W).

Experiment Settings. In the Kuroshio scenario, the configuration of NMO consists of a 4-layer encoder, a 4-layer decoder, and an 8-layer time evolution operator. During the pretraining process, we set the dimension of the latent space to 256 and adopt an early stopping strategy. The pretraining process halts when the reconstruction error converges to 0.0000032. The intrinsic dimension is 6. During the training process, we set the batch size to 15 and the total epochs to 500. The parameters of the encoder and decoder are frozen. In both the pretraining and training processes, we employ a mean squared error (MSE) loss and an Adam optimizer with a learning rate of 0.001. The time steps for the input and output tensors are 5 and 15, respectively. Each time step interval is 1 day. The model's maximum prediction length through a single forward process is 15 days.

Experiment Result. In the Kuroshio scenario, the training details of the NMO are shown in Figure. 9. In the training process, NMO typically converges after approximately 200 epochs, reaching an L2 error of 0.00177859 and a Relative L2 error of 0.02066821.

In Figure 10, we use the historical 5-day ocean current velocity field as input and forecast the future state for the next 15 days. Surprisingly, even on the 15th day, the prediction aligns closely with the

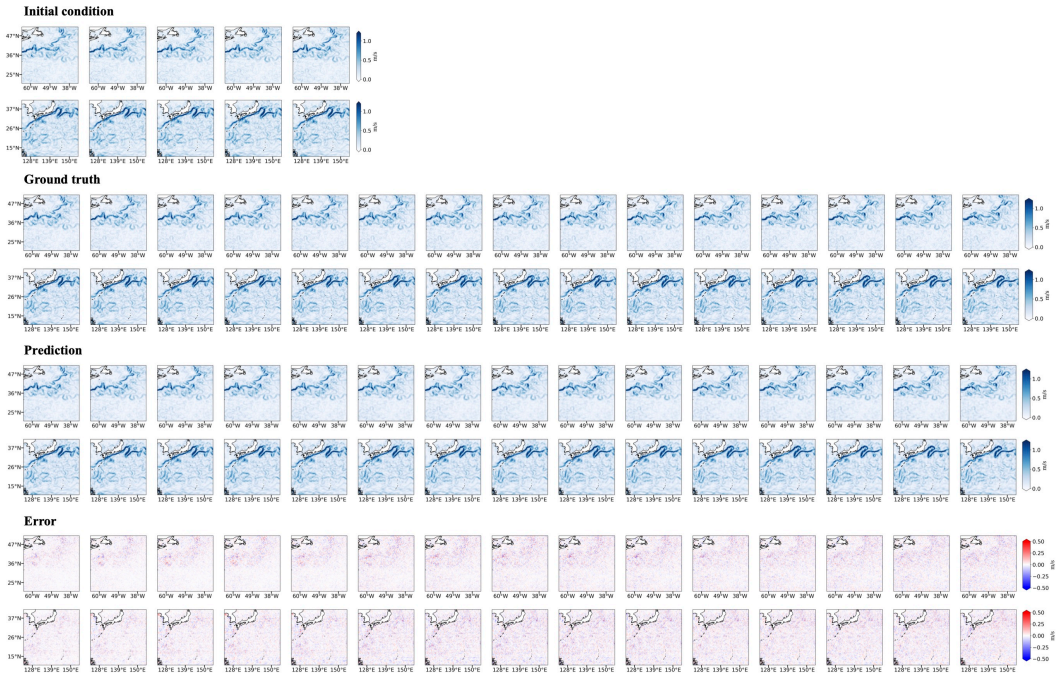

Figure 10: Visualization of the Kuroshio scenario.

ground truth in spatial patterns. Furthermore, the forecast error exhibits a subtle increase in the lead time.

B.1.3 TYPHOON

Data Description. Typhoon data is obtained by the Japanese Himawari-8 Geostationary Satellite Data including three water vapor channels at high, mid, and low altitudes in the East Asia to Southeast Asia Pacific coastal region. The dataset captures the development and growth stages of typhoons through water vapor information. The time series of past meteorological satellite data contained in the dataset can be used to train the model to predict the details of typhoon development in the next 36 hours, including its position, intensity, and water vapor distribution.

Experiment Settings. In the Typhoon scenario, the configuration of NMO consists of a 6-layer encoder, a 6-layer decoder, and a 16-layer time evolution operator. During the pretraining process, we set the dimension of the latent space to 768 and adopt an early stopping strategy. The pretraining process halts when the reconstruction error converges to 0.0001232. The intrinsic dimension is 8. During the training process, there are 300 epochs, and we set the batch size to 1. The parameters of

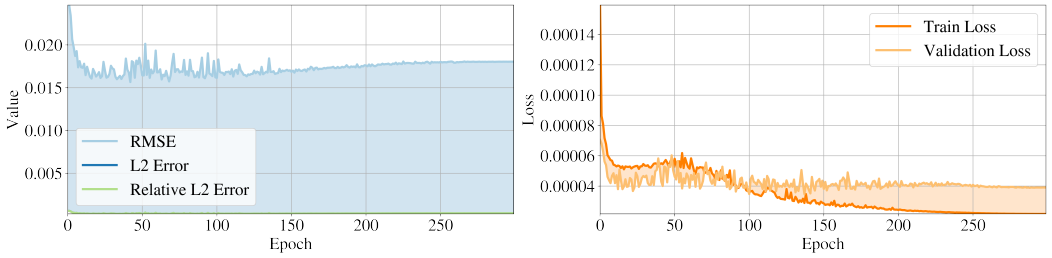

Figure 11: Training details for the Typhoon scenario. The left picture shows the variation of RMSE, L2 error, and relative L2 error with iterations. The right picture shows the change in MSE loss for both the training and validation dataset with respect to iterations.

the encoder and decoder are frozen. In both the pretraining and training processes, we employ a mean squared error (MSE) loss and an Adam optimizer with a learning rate of 0.01. The time steps for the input and output tensors are 6 and 36, respectively. Each time step interval is 1 hour. The model's maximum prediction length through a single forward process is 36 hours.

Experiment Result. In the context of the Typhoon scenario, the training details of NMO are depicted in Figure 11. Throughout the training, NMO typically converges after approximately 150 epochs, achieving an L2 error of 0.00032553 and a relative L2 error of 0.00032245. The visualization results are shown in Figure 12. The first section of the visualization presents the initial condition. The three rows therein represent the data from the high, medium, and low water vapor channels respectively. The subsequent 6 images depict input data over a continuous 6-hour period. In the second section, we showcase the ground reality data for the upcoming 36-hour. However, for the sake of brevity, we chose to display one image every six hours. The third section reveals NMO's predictions for the next 36 hours. The fourth row contrasts the predicted data with the actual results, providing an intuitive assessment of the prediction error.

B.2 Equation-governed Scenarios

B.2.1 Navier-Stokes Equation

Data Description. The dataset is calculated through a pseudospectral method to solve a viscous, incompressible 2-d Navier-Stokes equation with vorticity form expressed as

$$\begin{aligned} \partial_t w(x,t) + u(x,t)\cdot\nabla w(x,t) &= \nu\Delta w(x,t) + f(x), && x\in(0,1)^2, t\in(0,T] \\ \nabla\cdot u(x,t) &= 0, && x\in(0,1)^2, t\in[0,T] \\ w(x,0) &= w_0(x), && x\in(0,1)^2. \end{aligned}$$

The forcing term is fixed as $f(x) = 0.1\left(\sin\left(2\pi\left(x_1+x_2\right)\right)+\cos\left(2\pi\left(x_1+x_2\right)\right)\right)$. In order to obtain a set of different solutions for training the mapping between Banach space, the initial condition is generated by the $w_0 \sim \mu$ with $\mu = \mathcal{N}\left(0, 7^{3/2}(-\Delta+49I)^{-2.5}\right)$ and the boundary condition is periodic boundary conditions. We set the viscosity coefficient as $\nu = 10^{-5}$ to make the solutions become chaotic enough with time evolution.

Experiment Settings. In the Navier-Stokes equation scenario, the configuration of NMO consists of a 4-layer encoder, a 4-layer decoder, and an 8-layer time evolution operator. During the pretraining process, we set the dimension of the latent space to 256 and adopt an early stopping strategy. The pretraining process halts when the reconstruction error converges to 0.0015347. The intrinsic dimension is 4. During the training process, there are 100 epochs, and we set the batch size to 20. The parameters of the encoder and decoder are frozen. In both the pretraining and training processes, we employ a mean squared error (MSE) loss and an Adam optimizer with a learning rate of 0.01. The time steps for the input and output tensors are 10 and 10, respectively. The model's maximum prediction length through a single forward process is 10 time steps.

Experiment Result. In the context of the Navier-Stokes equation scenario, the training details of NMO are outlined in Figure 13. Throughout its training, NMO typically converges after roughly 70 epochs, achieving an L2 error of 0.05058351 and a relative L2 error of 0.03410483. The visualization results are presented in Figure 14. The first row depicts the initial conditions with 10 time steps. The second row showcases the ground truth data with 10-time steps, while the third row illustrates NMO's predictions for the subsequent 10 time steps. The fourth row offers a visual comparison between the predicted results and the actual data, highlighting the prediction error.

B.2.2 Shallow-Water Equations

Data Description. The dataset is calculated by a comprehensive finite volume solver to solve 2-d Shallow-Water equations for free-surface flow, expressed as

$$\partial_t h + \partial_x h u_x + \partial_y h u_y = 0$$
$$\partial_t h u_x + \partial_x \left(u_x^2 h + \frac{1}{2} g_r h^2 \right) = -g_r h \partial_x b$$
$$\partial_t h u_y + \partial_y \left(u_y^2 h + \frac{1}{2} g_r h^2 \right) = -g_r h \partial_y b$$

where h is the depth of the water column, u_x and u_y is the component of velocity with respect to the x-axis and y-axis, b is the bathymetry variation and g_r is the gravitational acceleration.

Experiment Settings. In the Shallow-Water equations scenario, the configuration of NMO consists of a 3-layer encoder, a 3-layer decoder, and a 6-layer time evolution operator. During the pretraining process, we set the dimension of the latent space to 256 and adopt an early stopping strategy. The pretraining process halts when the reconstruction error converges to 0.0000312. The intrinsic dimension is 6. During the training process, there are 100 epochs, and we set the batch size to 10. The parameters of the encoder and decoder are frozen. In both the pretraining and training processes, we employ a mean squared error (MSE) loss and an Adam optimizer with a learning rate of 0.01. The time steps for the input and output tensors are 10 and 90, respectively. The model's maximum prediction length through a single forward process is 90 time steps.

Experiment Result. In the context of the Shallow-Water equations scenario, the training details of NMO are outlined in Figure 15. Throughout its training, NMO typically converges after roughly 25 epochs, achieving an L2 error of 0.0000079 and a relative L2 error of 0.00270999. The visualization results are presented in Figure 16. The first row depicts the initial conditions with an input of 10-time steps. The second row showcases the ground truth data for those 90-time steps, while the third row illustrates NMO's predictions for the subsequent 90-time steps. The fourth row offers a visual comparison between the predicted results and the actual data, highlighting the prediction error.

B.2.3 Rayleigh-Bénard Convection

Data Description. The dataset is calculated by the Lattice Boltzmann Method to solve the 2-d fluid thermodynamics equations for two-dimensional turbulent flow, and its general form is expressed as

$$\nabla \cdot \mathbf{u} = 0$$
$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho_0} \nabla p + v \triangle \mathbf{u} + \left[1 - \alpha \left(\mathrm{T} - \mathrm{T}_0\right)\right] \mathbf{X}$$
$$\frac{\partial \mathrm{T}}{\partial \mathrm{t}} + (\mathbf{u} \cdot \nabla)\mathrm{T} = \kappa \triangle \mathrm{T}$$

where g is the gravitational acceleration, $\mathbf{X}$ is the acceleration due to the body-force of the fluid parcel, ρ_0 is the relative density, T denotes temperature, T_0 is the average temperature, α denotes the coefficient of thermal expansion and κ denotes the coefficient of thermal conductivity. The simulation parameters of the data respectively are Prandtl number = 0.71, Rayleigh number = 2.5 × 108, and the maximum Mach number = 0.1.

Experiment Settings. In the Rayleigh-Bénard convection scenario, the configuration of NMO consists of a 2-layer encoder, a 2-layer decoder, and a 4-layer time evolution operator. During the pretraining process, we set the dimension of the latent space to 512 and adopt an early stopping strategy. The pretraining process halts when the reconstruction error converges to 0.0129318. The intrinsic dimension is 4. During the training process, there are 300 epochs, and we set the batch size to 1. The parameters of the encoder and decoder are frozen. In both the pretraining and training processes, we employ a mean squared error (MSE) loss and an Adam optimizer with a learning rate of 0.01. The time steps for the input and output tensors are 10 and 60, respectively. The model's maximum prediction length through a single forward process is 60 time steps.

Experiment Result. In the Rayleigh-Bénard convection scenario, Figure 17 depicts the training details of the NMO. Throughout the training process, the NMO typically converges after about 150 cycles and reaches an L2 error of 0.0201180, while the relative L2 error is 0.0053570. As Fig.

18 shown, the first column displays the 10 time-steps for the initial condition; the second column illustrates the ground truth, which only shows time-step indexes of 6, 12, 18, ..., 54, 60; the third column shows the NMO prediction results; and the fourth column presents the error visualization.

B.2.4 DIFFUSION-REACTION EQUATION

Data Description. The dataset is calculated by the finite volume method for spatial discretization and the fourth-order Runge-Kutta method for time integration to solve a 2-d diffusion-reaction equation expressed as

$$\partial_t u = D_u \partial_{xx} u + D_u \partial_{yy} u + R_u,$$
$$\partial_t v = D_v \partial_{xx} v + D_v \partial_{yy} v + R_v,$$

where D_u, D_v, R_u, and R_v are the diffusion coefficient and reaction function for the activator and inhibitor, respectively. The reaction functions are defined as

$$R_u(u, v) = u - u^3 - k - v$$
$$R_v(u, v) = u - v$$

where constant number $k = 5 \times 10^{-3}$, the diffusion coefficients $D_u = 1 \times 10^{-3}$ and $D_v = 5 \times 10^{-3}$.

Experiment Settings. In the Diffusion-Reaction equation scenario, the configuration of NMO consists of a 4-layer encoder, a 4-layer decoder, and an 8-layer time evolution operator. During the pretraining process, we set the dimension of the latent space to 256 and adopt an early stopping strategy. The pretraining process halts when the reconstruction error converges to 0.00000292. The intrinsic dimension is 6. During the training process, there are 100 epochs, and we set the batch size to 2. The parameters of the encoder and decoder are frozen. In both the pretraining and training processes, we employ a mean squared error (MSE) loss and an Adam optimizer with a learning rate of 0.01. The time steps for the input and output tensors are 50 and 50, respectively. The model's maximum prediction length through a single forward process is 50 time steps.

Experiment Result. In the context of the Diffusion-Reaction equation, the training process of NMO is depicted in Figure 19. Throughout the training, NMO typically converges around 400 epochs, achieving an L2 error of 0.00000063 and a relative L2 error of 0.00005421. The associated visualization results are shown in Figure 20. Given that the equation involves two variables, the visualization is split into two sections, left and right. On the left, the initial conditions are presented, followed by the ground truth values. Subsequently, the NMO's predictions are displayed, and the last section presents the prediction error.

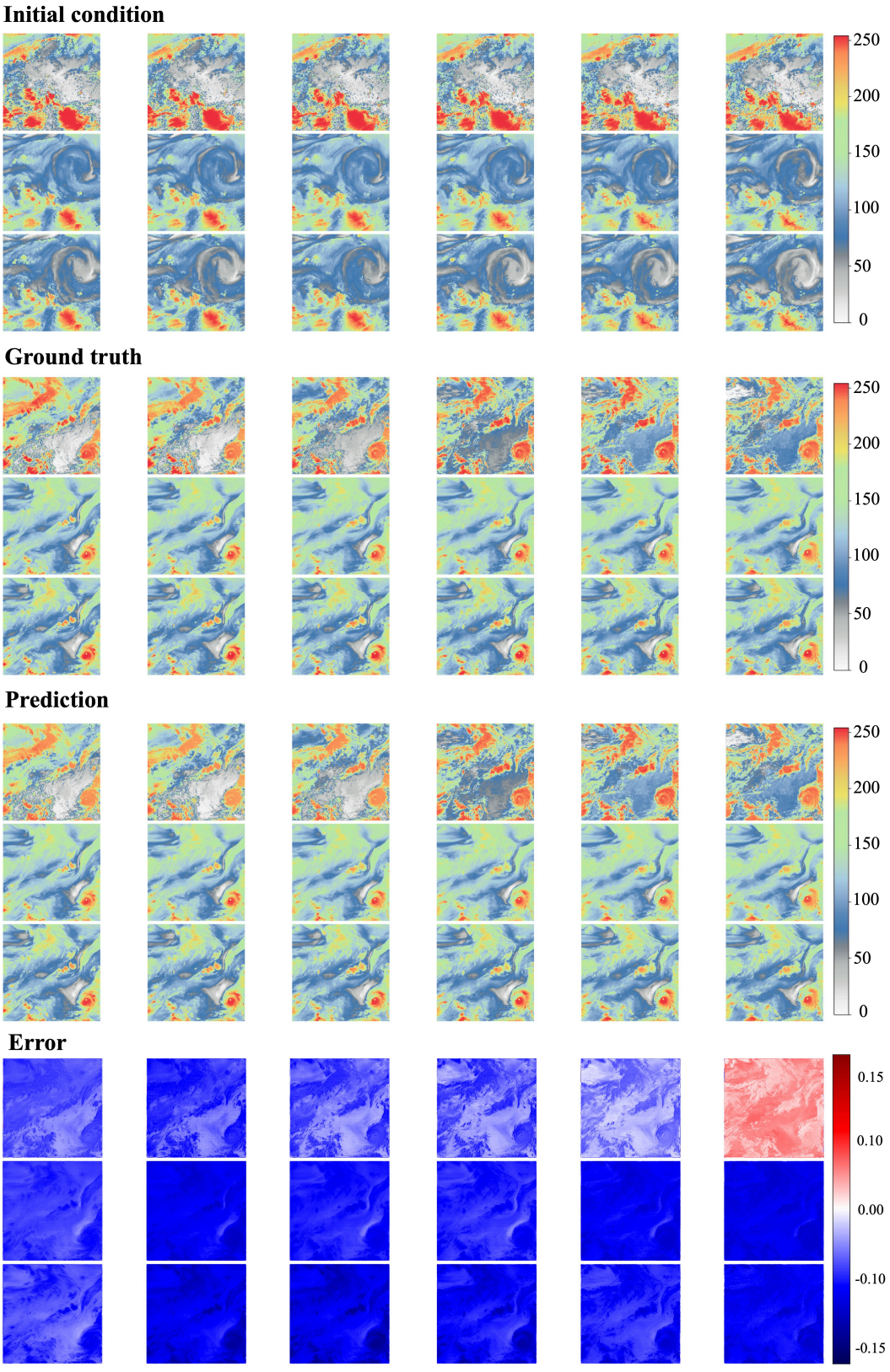

Figure 12: Visualization of the Typhoon scenario.

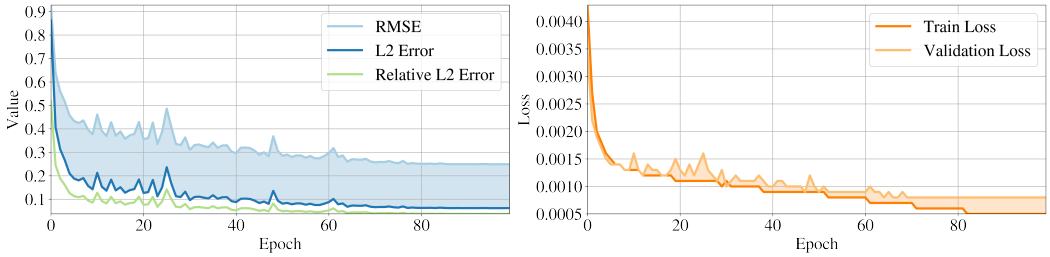

Figure 13: Training details for the Navier-Stokes equation scenario. The left picture shows the variation of RMSE, L2 error and relative L2 error with iterations. The right picture shows the change in MSE loss for both the training and validation dataset with respect to iterations.

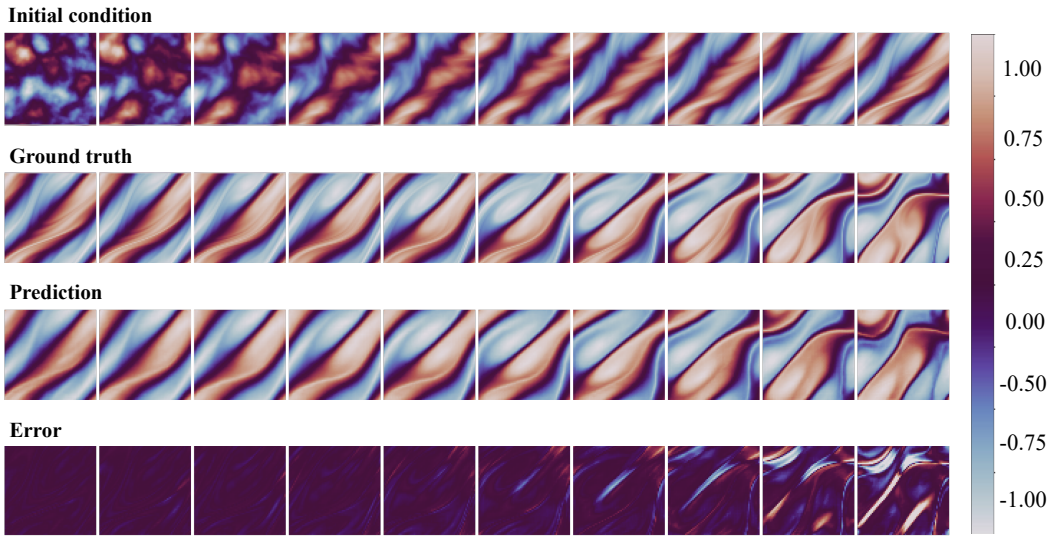

Figure 14: Visualization of the Navier-Stokes equation scenario.

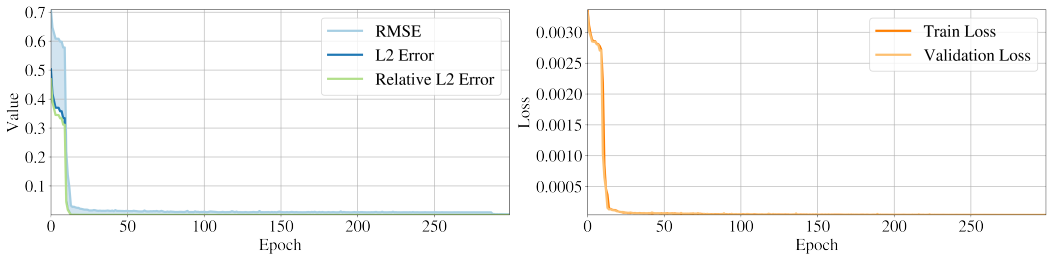

Figure 15: Training details for the Shallow-Water equations scenario. The left picture shows the variation of RMSE, L2 error and relative L2 error with iterations. The right picture shows the change in MSE loss for both the training and validation dataset with respect to iterations.

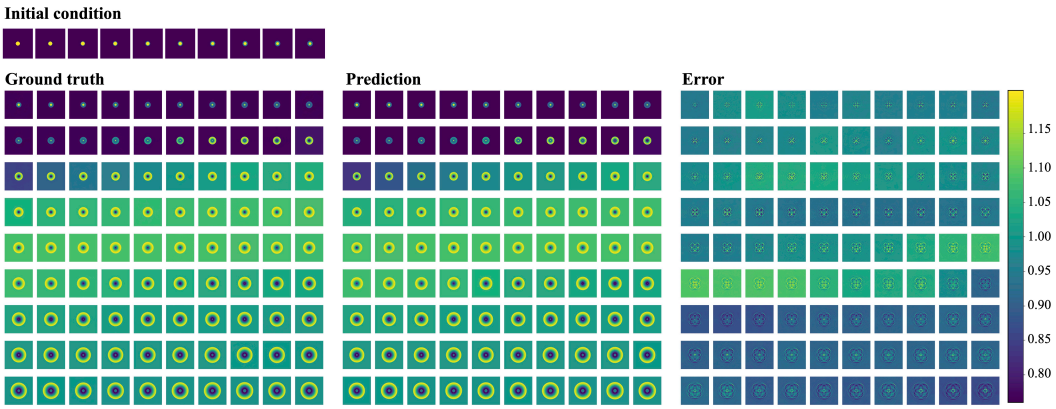

Figure 16: Visualization of the Shallow-Water equations scenario.

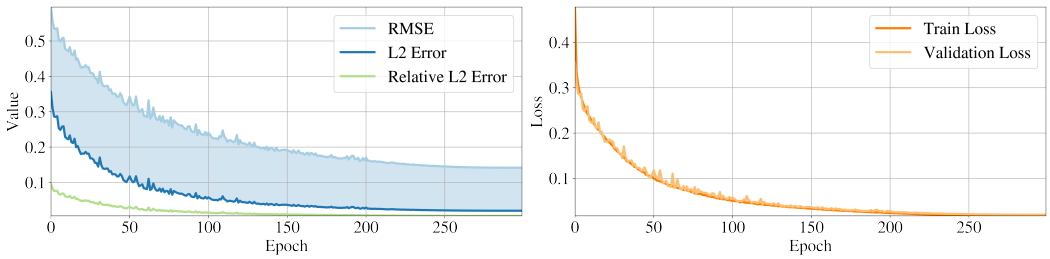

Figure 17: Training details for the Rayleigh-Bénard convection scenario. The left picture shows the variation of RMSE, L2 error and relative L2 error with iterations. The right picture shows the change in MSE loss for both the training and validation dataset with respect to iterations.

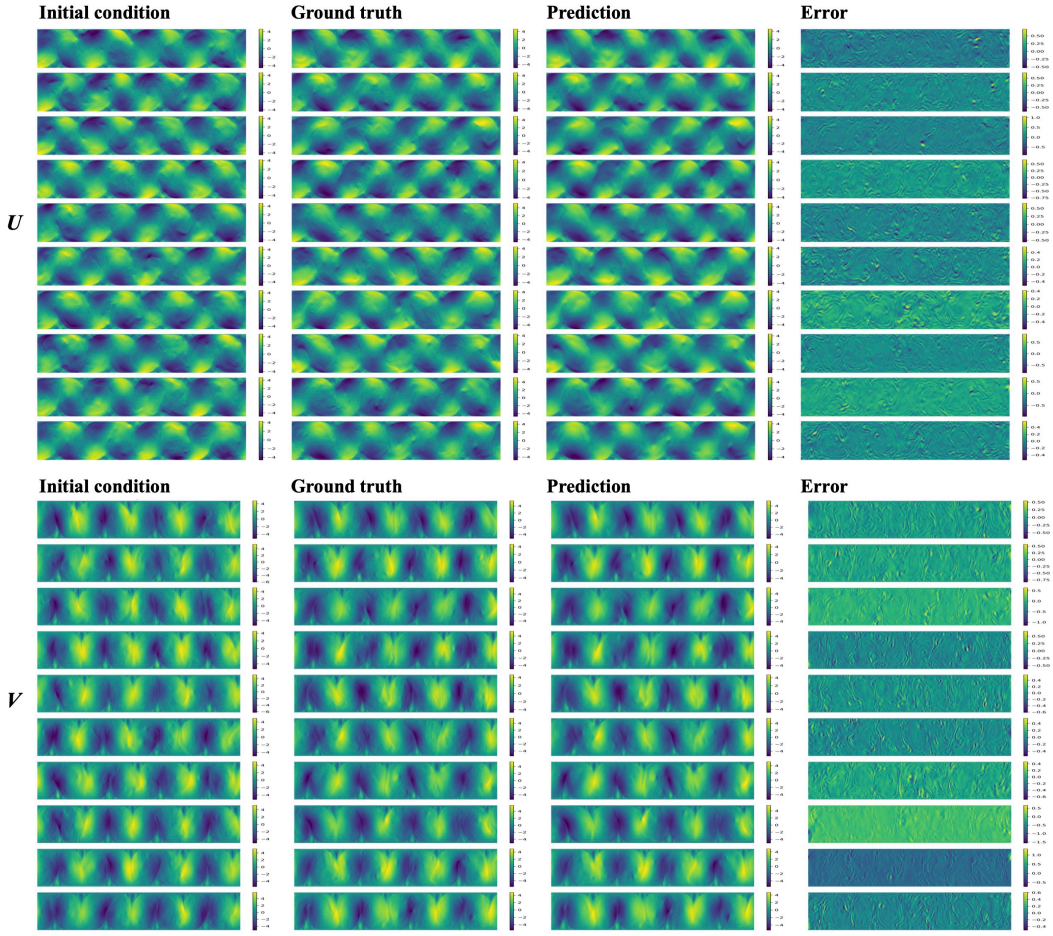

Figure 18: Visualization of the Rayleigh-Bénard convection scenario.

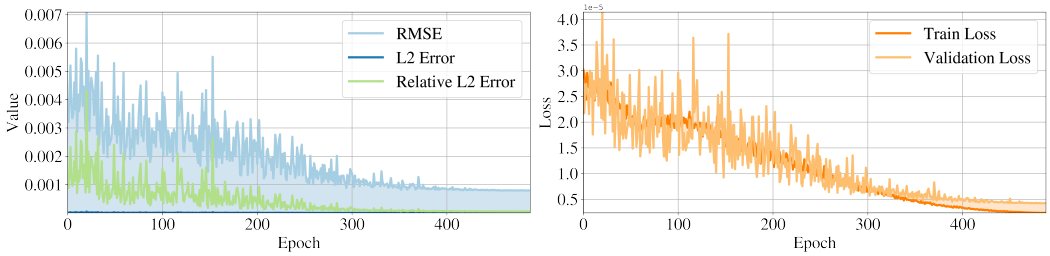

Figure 19: Training details for the Diffusion-Reaction equation scenario. The left picture shows the variation of RMSE, L2 error and relative L2 error with iterations. The right picture shows the change in MSE loss for both the training and validation dataset with respect to iterations.

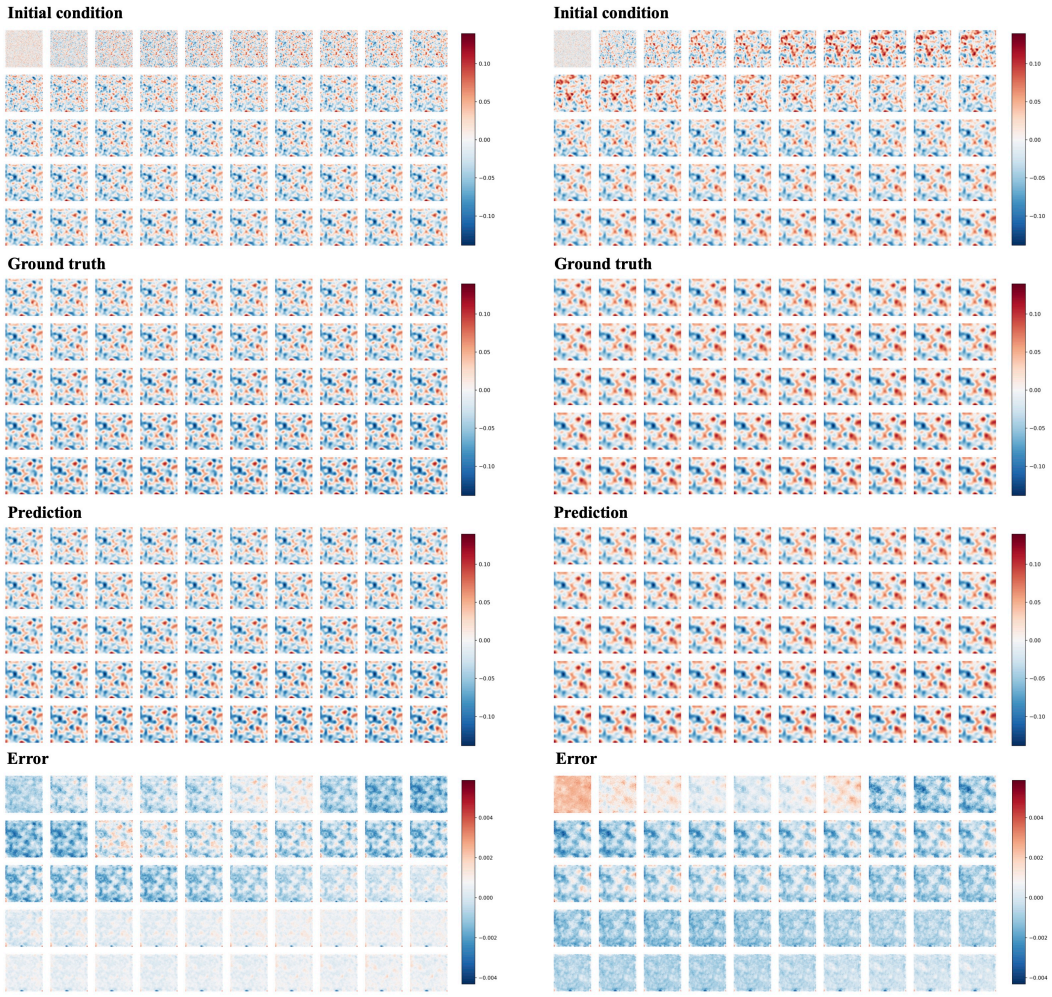

Figure 20: Visualization of the Diffusion-Reaction equation scenario