LA-UR-24-26289

Approved for public release; distribution is unlimited.

Title:	Physics-assisted Latent Space Dynamics Learning for Stiff Collisional-radiative Models
Author(s):	Xie, Xuping Tang, Qi Tang, Xianzhu
Intended for:	Report
Issued:	2024-06-26

Physics-assisted Latent Space Dynamics Learning for Stiff Collisional-radiative Models

Xuping Xie,[1] Qi Tang,[1] and Xianzhu Tang[1]

Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

(*Electronic mail: xxie@lanl.gov)

(Dated: 17 June 2024)

Collisional-radiative (CR) models describe the atomic processes in a plasma by tracking the population density in the ground and excited states for each charge state of the atom/ion. These models predict important plasma properties such as charge state distributions and radiative emissivity and opacity. Accurate descriptions of the CR balance of the plasma are essential in fusion whole device modeling, especially when significant impurities are introduced into the plasmas. In an integrated fusion plasma and CR simulation, a CR model, which is a high-dimensional stiff ODE, needs to be solved on each grid point in the configuration space, and can overwhelm the overall computational cost. In this work, we propose a machine-learning-based method that discovers a latent space and learns its corresponding latent dynamics, which can capture the essential physics to make accurate predictions at much lower online computational cost. The proposed approach is physics-assisted, due to its combination of a physical latent space and a data-driven latent space. It has been demonstrated that the proposed architecture can predict both the full-order CR dynamics and a physical quantity of interest accurately.

I. INTRODUCTION

Fusion reactors, such as tokamaks, confine plasma using magnetic fields to sustain nuclear fusion reactions. However, maintaining stable plasma confinement is challenging due to instabilities that can lead to disruptions. Plasma disruptions involve a sudden loss of confinement, causing the plasma to interact with the reactor walls, leading to potential damage and reduced reactor performance. To prevent or mitigate these disruptions, understanding the underlying physical processes, including radiative and collisional interactions within the plasma, is essential. In particular, collisional-radiative (CR) modeling is a critical component in the study of plasma physics, especially in the context of plasma disruption mitigation in fusion reactors [1].

The CR model represents these interactions and is mathematically described by a high-dimensional nonlinear stiff dynamical system. This model considers various processes such as ionization, recombination, excitation, and de-excitation, which occur simultaneously within the plasma. In practical plasma disruption simulations, high-fidelity simulations of the CR model are extremely computationally expensive, making the use of the original CR system impractical for real-time applications. As a result, there is an increasing interest and demand for efficient and accurate surrogate models for the CR system in plasma disruption simulations [2,3].

The objective of this work is to develop a data-driven surrogate method for the CR model using deep neural networks. Specifically, we first use a physics-assisted autoencoder on the CR data to find a low-dimensional latent representation of the original CR system. Then, we use a flow map neural network to learn the latent dynamics. Once our reduced surrogate model is trained, we can predict the whole latent dynamics given only the initial condition, through iteratively applying the flow map neural network, and then reconstruct its radiative power loss via a decoder. By leveraging deep learning techniques, our proposed surrogate model can provide a computationally efficient and accurate representation of the CR dynamics, facilitating better prediction and mitigation of plasma disruptions in fusion reactors.

A. Collisional-radiative Modeling

CR modeling deals with the complex interactions between electrons, ions, and neutral particles in a plasma. These interactions include collisional excitation, de-excitation, ionization, and recombination processes, as well as radiative emissions, Fig. 1 illustrates the whole CR process.

The goal of CR modeling is to calculate the population densities of various charge states and energy levels within the plasma, accounting for both collisional and radiative processes. To manage the wide range of possible ion states that can occur in a plasma, we order the set of all ion population density vector in a given plasma as,

$$\mathbf{N} = \{N^j_{\alpha,Z}\},$$

where $N^j_{\alpha,Z}$ denotes the population density of ion level j for atomic species α and charge state Z, which ranges from 0 to A_α, the atomic number of species α. The index j labels each of the ground and excited states. Typical gas species used in impurity plasma mitigation include helium ($A_\alpha = 2$), lithium ($A_\alpha = 3$), nitrogen ($A_\alpha = 7$), neon ($A_\alpha = 10$), and argon ($A_\alpha = 18$). The high fidelity solution of a CR model can accurately provide plasma properties as input to subsequent coupled stages of plasma simulation code. These properties include the species ion charge state

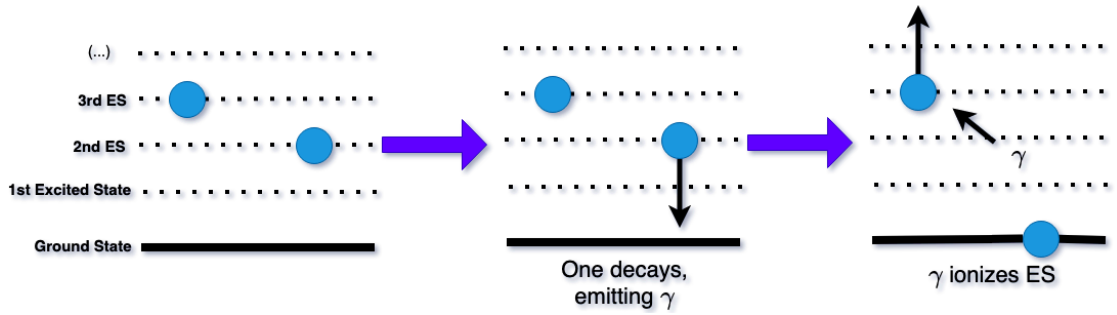

FIG. 1. Ion particles ionization and de-excitation in a CR model.

distribution $n_{\alpha,Z}$ and the radiative cooling rate $\mathscr{RL}$. These quantities are crucial for radiative energy transport in fusion plasma simulation and help design experiments at ITER. The species charge state density $n_{\alpha,Z}$ is simply the sum over all ground and excited states of a given charge state,

$$n_{\alpha,Z} = \sum_j N^j_{\alpha,Z}.$$

The radiative power loss (RL) is a crucial quantity representing the energy radiated away by the plasma due to various radiative processes. The radiative power loss can be calculated by summing up the contributions from all radiative transitions, including line emissions, recombination radiation, and bremsstrahlung. The general form of the radiative loss equation is given by:

$$\mathscr{RL} = \sum_{\alpha,Z,j,k} N^j_{\alpha,Z} A^j_{\alpha,Z\to k} E^j_{\alpha,Z\to k},$$

where $A^j_{\alpha,Z\to k}$ is the coefficient for spontaneous emission from energy level j to level k, and $E^j_{\alpha,Z\to k}$ is the the energy of the photon emitted in the transition from level j to level k.

The CR model is mathematically represented as a parameterized ODE:

$$\frac{d\mathbf{N}}{dt} = R(\mathbf{N}, n_A, T_e)\,\mathbf{N}, \tag{1}$$

with $\mathbf{N}$ the population density of ions in various charge states including both ground and excites states, T_e is the temperature, $n_{A,\alpha} = \sum_Z n_{\alpha,Z}$ is the total density for species α. The dimension of $\mathbf{N}$ can vary enormously depending on whether one would want to resolve the fine and super-fine structures in the excited states. In LANL ATOMIC model [4], it can have $N \approx 10^6$ for argon species. The rate matrix $R(\mathbf{N}, n_A, T_e)$ is a square matrix, which includes a number of atomic processes, broadly grouped in up-transition and down-transition. The up-transition includes collisional ionization/excitation, photo-ionization and excitation. The down-transition accounts for various channels of recombination and de-excitation; see Fig. 1 for instance. Collisional charge exchange can provide both down- and up-transition depending on the specific ion/atom involved. The overall rate matrix R has explicit dependence on the electron distribution. The normal assumption is to approximate the electron distribution as a Maxwellian with temperature T_e and total density $n_{A,\alpha}$ for species α.

The CR model is a parameterized stiff ODE as many of the transition rates are very fast on timescale of interest. Numerical solutions of the full system (1) require implicit methods and are usually performed with a standard linear algebra package. The high-dimensional dynamics of a CR model can be a computational burden to obtain high-fidelity solution with accurate prediction of the ion charge population distribution. The state-of-the-art CR models contain orders of millions degrees of freedom can take hours to obtain a solution. It is simply impossible to directly couple the CR physics module in its original form to 3D plasma simulations, since for each spatial grid point in the plasma simulation, one would need to evolve a coupled ODE with 10^{3-6} degrees of freedom, which would overwhelm the plasma simulation cost by many orders of magnitude. The goal of this work is to introduce a reduced-order model that can efficiently and accurately predict the fast scale dynamics transition of the charge state, $n_{\alpha,Z}$, and the radiative cooling rate, $\mathscr{R}_L$.

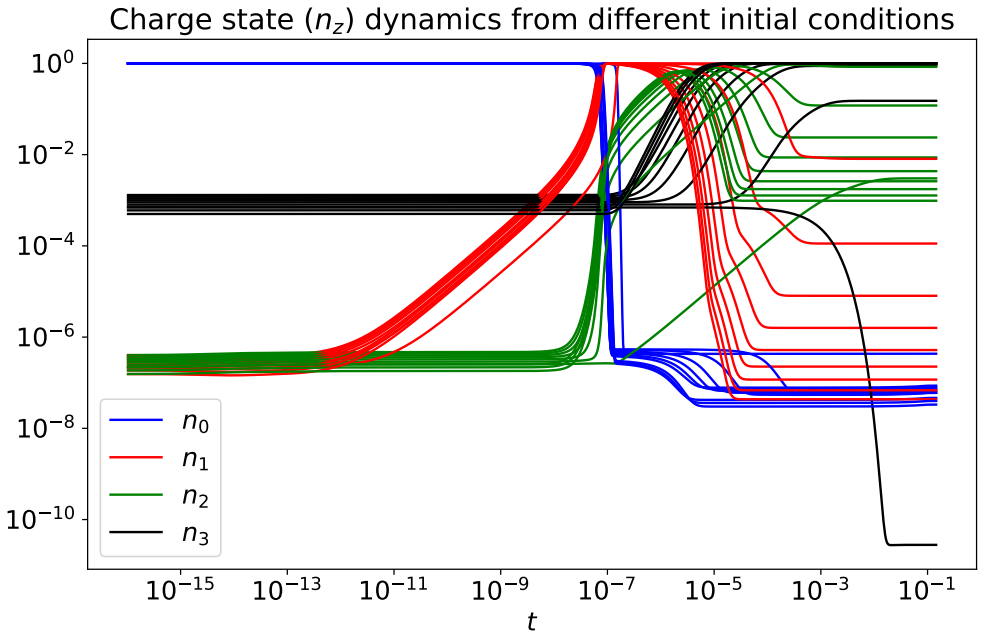

FIG. 2. Trajectory of the charge state from the high-fidelity numerical simulation of the CR model for different initial conditions. The solution is for a single species, lithium, where the atomic number $A_\alpha = 3$. Four colored lines represent four charge states (they are partial sums of the solution state vector; details will be defined later), n_0, n_1, n_2, n_3.

B. Related Work

Reduced Order Modeling (ROM) seeks low-dimensional approximations of high-dimensional systems, significantly reducing computational costs. In the framework of data-driven ROM, machine learning (ML) techniques and data are used to distill the essential features of complex systems into more manageable representations. This is especially relevant in fields such as fluid dynamics, material science, and climate modeling, where solving full-scale problems involves significant computational challenges due to high dimensionality. Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD) are two of the most popular approaches to extract dominant features and dynamic structures from data. These methods have been successfully applied in fluid mechanics, control, and biomechanical problems [5–8]. With the development of deep learning, autoencoder neural networks have gained significant attention in ROM due to their ability to efficiently compress high-dimensional data into a lower-dimensional latent space while preserving essential features. Recent works have demonstrated the use of autoencoders for learning low-dimensional representations of fluid dynamics [9,10]. Ref. [11] combined autoencoders

with DMD to learn the governing equations of dynamical systems directly from data, allowing for efficient predictions of future states.

Data-driven discovery of dynamics has emerged as a powerful tool for modeling and predicting the behavior of complex physical systems. This approach leverages ML techniques to extract patterns and underlying dynamics from data, facilitating the development of reduced-order models that are both efficient and accurate. Ref. [12] introduced the Sparse Identification of Nonlinear Dynamical Systems (SINDy) method, which aims to discover governing equations from data. By representing the dynamics in a library of candidate functions, SINDy selects the most relevant terms to construct a parsimonious model. This data-driven methodology has proven effective in capturing the underlying physics of complex systems with minimal assumptions, offering a powerful tool for system identification and model reduction [13,14]. In [15], they introduced Neural Ordinary Differential Equations (NODE), which parametrize the time derivative of the hidden state with a neural network, allowing the model to learn complex dynamics directly from data. This method can also be applied for identifying latent dynamics [16,17]. Ref. [18] utilized autoencoders to identify a latent space where the nonlinear dynamics are approximated by linear models based on linear Koopman operator theory, enabling efficient and accurate predictions of system behavior. Ref. [19] introduced physics- and data-assisted ROM based on approximate inertial manifolds theory using deep neural networks. Their approach is successfully demonstrated through dissipative PDEs. Ref. [19] discussed the "physics-assisted" latent space learning where the latent space is a "gray-box" approach including the known physics latent space and data-driven latent space. A few recent works on learning flow maps using structure-preserving neural networks can be found in [20,21]. These studies underscore the potential of integrating ROM approach with latent dynamics learning to address the computational challenges in simulating high-dimensional dynamical systems.

C. Our contribution

The main challenge in finding the reduced CR model lies in two parts. The first is to efficiently identify the low-dimensional representation from the data. The second is to learn latent dynamics that can accurately predict the trajectories of the reduced system using only the initial conditions. This allows for an approximation of the full-order dynamics and radiative cooling rate with high accuracy. Fig. 2 plots the trajectories of different charge states from the high-fidelity numerical

simulation. It is evident that the sharp transitions over very short time scales present a significant challenge in accurately modeling the reduced CR dynamics. In this work, we introduce a novel physics-assisted latent space dynamics learning approach based on deep neural networks, which is partially related to [19]. First, we use an autoencoder neural network to identify the low-dimensional representation of the original high-dimensional CR system. This representation is denoted as the black space. We then incorporate the known physical information, specifically the charge states, denoted as the white space, into the entire latent space. This gray-box approach ensures that both data-driven and physics-based aspects are integrated into the model. Then, we use a flow map neural network to learn the dynamics within the latent space, similar with the approaches in [20,21]. To the best of our knowledge, this is the first effort to apply the integrated data-driven and physics-based latent dynamics learning for constructing a reduced-order model of a parameterized CR system. For the purpose of demonstration, our approach will be excessively tested by the CR data from a single species, lithium, under different parameters and initial conditions in Section III.

II. DATA-DRIVEN MODEL REDUCTION FOR CR MODELING

In this section, we present a physics-assisted model reduction framework for CR modeling employing deep learning techniques. The methodology involves two crucial steps: latent space reduction and latent dynamics learning, both of which are facilitated by designing an appropriate neural network architecture. The network comprises an encoder to reduce the input dimension to a lower latent space, followed by a neural network to learn the flow map of the latent space. Subsequently, a decoder is employed to reconstruct the full ion density space. Fig. 3 illustrates the architecture of our data-driven ML-based surrogate for the CR system.

A. Physics-assisted Latent Space via Autoencoder

In the latent space discovery, we integrate known physics information by adding the charge state as **white space** into the latent space discovered by the autoencoder. The latent space is thus composed of both the white space and the black space. The white space is derived from the charge state of the corona equilibrium model of CR modeling. In the corona equilibrium model, the rate equations for the charge states simplify significantly. The assumption is that the ion population

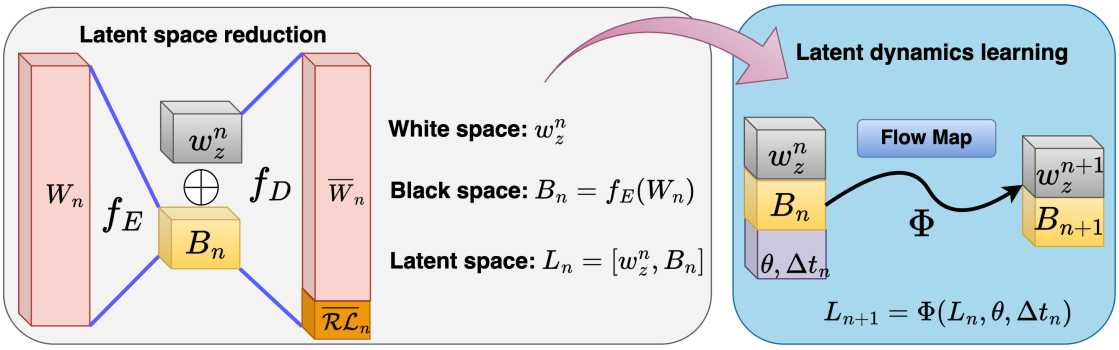

FIG. 3. Autoencoder architecture for physics-assisted latent space identification (left). Flow map for latent dynamics learning (right). θ denotes the parameter space in our CR system which are total density, n_A, and temperature, T_e.

is primarily determined by ionization from and recombination to the ground state, rather than by transitions between excited states.

$$\frac{dn_z}{dt} = R \cdot n_z$$

The corona equilibrium model leads to a significant simplification of the CR model by reducing the complexity of the rate equations. Instead of considering detailed level populations, the model focuses on the balance between ionization and recombination rates for each charge state. This makes it easier to calculate the charge state distribution and related properties such as radiative losses and spectral line intensities. The charge state from the corona equilibrium model provides a simplified yet accurate representation of the ionization balance in a high-temperature plasma. This model captures the essential physics of ionization and recombination processes without the complexity of detailed level populations. By focusing on the charge state, we reduce the dimensionality of the problem. This allows us to incorporate critical physical insights without overwhelming the autoencoder with excessive complexity. The autoencoder can then efficiently learn the latent dynamics of the remaining variables (black space). Incorporating the charge state as white space guides the autoencoder towards physically meaningful representations. This helps the model to learn latent dynamics that are consistent with known physical laws. The inclusion of known physics information in the latent space enhances the interpretability of the model.

Autoencoders consist of two main components: an encoder function f_E, which maps the input data to a lower-dimensional latent space, and a decoder function f_D, which reconstructs the original data from its latent representation. During training, the autoencoder minimizes the reconstruction error between the input and the reconstructed output, thereby capturing the most

significant features of the data. In the first part of our data-driven framework, an autoencoder is utilized to reduce the full ion population space into a **black space**, B_n, at a given time step t_n. This black space, in combination with known physics information such as the charge state (i.e., a white space, w_z^n), forms our latent space, L_n. The full ion population can then be reconstructed through a decoder f_D from this latent space. Note that when setting the dimension of the black space to zero, we solely rely on the charge state as our latent space, which is referred to as coronal equilibrium, though impractical. Additionally, the radiation power loss, $\mathscr{RL}$, can be retrieved from the decoder. In our case, the encoder part transforms the input, i.e., a normalized ion population $W_n \in \mathbb{R}^n$, to a black space $B_n \in \mathbb{R}^r$ with $r \ll N$:

$$B_n = f_E(W_n).$$

The decoder part tries to reconstruct the input W_n and predict the radiative loss from the latent space $L_n = [w_z^n, B_n]$:

$$[\overline{W_n}, \overline{\mathscr{R}_{L,n}}] = f_D(L_n).$$

The objective is to minimize the reconstruction loss $\mathscr{L}_{recons}$, which measures the difference between the input W_n and the reconstructed output $\overline{W_n}$ and $\overline{\mathscr{R}_{L,n}}$:

$$\mathscr{L}_{recons} = \frac{1}{S}\sum_{n=1}^{S}\left[\|W_n - \overline{W_n}\|^2 + \|\mathscr{R}_{L,n} - \overline{\mathscr{R}_{L,n}}\|^2\right],$$

where S is the total number of time steps in our training dataset.

B. Latent Dynamics Learning via Flow Map

In dynamical systems theory, a flow map describes the evolution of states in the system over time. Specifically, it is a mathematical function that maps an initial state to its future state at any given time instant. The flow map serves as a bridge between the system's underlying equations and its qualitative behavior. For continuous-time dynamical systems governed by ordinary differential equations (ODEs), the flow map elucidates how an initial point in the system's phase space evolves along a trajectory determined by the ODEs. In discrete-time systems, the flow map is often a straightforward function illustrating transitions between states from one time step to the next. Analyzing the flow map offers insights into critical properties of the system, such as stability, convergence, and the existence of attractors or repellors, which are crucial in comprehending the

long-term behavior of the system. Capturing the essence of how states transition over time, the flow map in the CR model represents the latent dynamics, which can be leveraged to derive the quantity of interest with minimal computational cost—a significant advantage in fusion simulation for disruption mitigation. In continuous-time dynamical systems described by ordinary differential equations (ODEs), the latent system's behavior can be expressed as:

$$\frac{d\mathbf{L}}{dt} = \mathbf{F}(\mathbf{L};\theta),$$

Here, $\mathbf{L} \in \mathbb{R}^r$ is the reduced latent state vector, and $\mathbf{F}(\mathbf{L};\theta)$ is the vector field that describes the dynamics of the system parameterized by θ. The flow map $\Phi(\Delta t, \mathbf{L}_0)$ is a function that maps the initial state $\mathbf{L}_0$ at $t = t_0$ to its future state $\mathbf{L}(t)$ at a time interval Δt, such that:

$$\mathbf{L}(t_0 + \Delta t) = \Phi(\Delta t, \mathbf{L}(t_0))$$

In general, for discrete-time dynamical systems with a uniform time step, the system can often be described by:

$$\mathbf{L}_{n+1} = \mathbf{G}(\mathbf{L}_n)$$

Since our CR model has an exponential time steps and also parameterized by the total density n_A and temperature T_e, our flow map Φ acts at discrete steps, mapping $\mathbf{L}_n$ to $\mathbf{L}_{n+1}$ and can be denoted as:

$$\mathbf{L}_{n+1} = \Phi(\Delta t_n, \mathbf{L}_n; n_A, T_e),$$

see the Fig. 3 for the model architecture. In this case θ includes n_A and T_e. Thus, the flow map can be approximated by a neural network with the loss function defined as the following,

$$\mathscr{L} = \frac{1}{S}\sum_{n=0}^{S}\left[\|\mathbf{L}_{n+1} - \Phi(\Delta t_n, \mathbf{L}_n;\theta)\|^2 + \|n_A - \sum_{i=0}^{Z} n_i^n\|^2\right],$$

where $\Phi(\Delta t_n, \mathbf{L}_n;\theta)$ is the neural network prediction of the trajectory at next time step, and $\|n_A - \sum_{i=0}^{Z} n_i^n\|^2$ is the additional physical constraint, i.e., the atomic conservation law.

III. NUMERICAL EXPERIMENT

In this section, we present detailed numerical experiments for the proposed surrogate model in Section II. As increasing the number of species would significantly amplify the dataset size, computational requirements, and training costs, we utilize Lithium as our representative species

in our high-fidelity simulation of the CR model in this study to obtain the necessary data. The CR solution of Lithium has $\dim \mathbf{N} = 94$ states corresponding to different energy levels. Since the atomic number $A_\alpha = 3$, we have 4 charge states defined as the following,

$$n_0 := \sum_{i=0}^{31} N_i, \qquad n_1 := \sum_{i=32}^{62} N_i, \qquad n_2 := \sum_{i=63}^{93} N_i, \qquad n_3 := N_{94}.$$

Fig. 2 shows the charge state trajectories from different initial conditions. We can clearly see that there is sharp transition at very tiny time scale in the CR dynamics which makes it one of challenging part in the reduced latent dynamics learning.

A. Data processing

The success of the ML training heavily relies on the proper data processing. Here we describe two critical components to rescale the data to ease the ML training. A proper sampling for stiff dynamics is also an important step to guarantee a good training result.

a. ***Ion Density Normalization.*** The magnitude of ion population $\mathbf{N}$ from the numerical solution varies largely between 1e13 to 1e-11. Since the total density n_A should be preserved for different parameter setting, we first normalize the population using its total density n_A as the follows,

$$\tilde{\mathbf{N}} = \frac{\mathbf{N}}{n_A}, \qquad n_A = \sum_{z=0}^{Z} n_z$$

After the total density scaling, the range of ion population density $\tilde{\mathbf{N}}$ lies in $(1\text{e-}27, 1)$. Many of the ion densities are still very small, with magnitudes less than 1e-10 making it challenging for the neural network to train on such values of tiny magnitude. To accurately resolve the high-fidelity dynamics of the CR model, it is important to resolve small values of the states, Fig. (6) shows the high-fidelity solution of the charge state dynamics from different temperature (Te). The raw data however is dominated by one or two states that have much larger magnitudes.

To accommodate that, we first apply the following change of variable to transform the data into a more suitable range:

$$\tilde{W} = 1 - \log(\tilde{\mathbf{N}}),$$

then using $(0,1)$ min-max scaler on $\tilde{W}$ to obtain the properly scaled training data W:

$$W = \frac{\tilde{W} - \tilde{W}_{min}}{\tilde{W}_{max} - \tilde{W}_{min}}.$$

The transformed variable W lies in $(0,1)$ and is properly scaled, making it easier for the neural network to train effectively. Fig. 4 shows the charge state, n_z, both in its original scale and its normalized scale (between 0 and 1), as used in practical neural network training. Note that we use the notation w_z (w0, w1, w2, w3) to represent the transformed charge state of n_z (n0, n1, n2, n3).

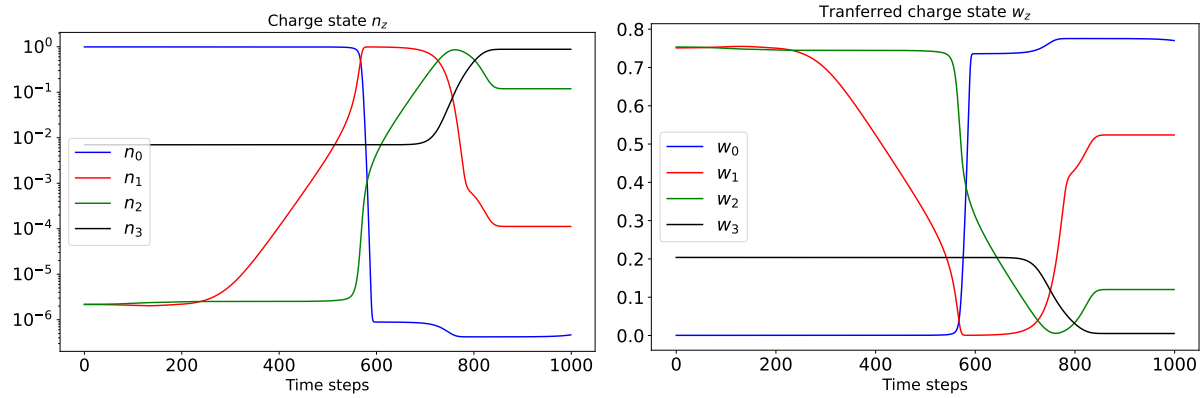

FIG. 4. Charge states in the original scale, n_z, and in the normalized scale, w_z. Latter is used in the neural network training.

b. ***Time Step Scaling.*** Time step size scaling is another crucial aspect for a successful training. For stiff equations or rapidly changing dynamics, a fixed time step size is often inadequate for capturing the system's behavior accurately. In such cases, adaptive time stepping methods, based on the local behavior of the system, are employed. This adaptive approach often results in more accurate and computationally efficient simulations. In the high-fidelity numerical simulation of the CR model, we used prescribed adaptive time steps in the numerical integration. The dataset was collected from non-uniform time step solutions with Δt_n ranging from 10^{-16} to 10^{-1}. The tiny scale of the time step size makes it challenging to train the neural network, as we input Δt_n into the flow map to learn the latent dynamics evolution from current step $L(t_n)$ to the next $L(t_{n+1})$. To make the neural network training more efficient, we use the following transfer formula to scale Δt_n into a proper range of $(0,1)$,

$$\tilde{\Delta t} = -\frac{1}{\log(\Delta t)}. \tag{2}$$

See Fig. 5 for a demonstration.

c. ***Coarse Sampling.*** We simulate the CR equation (1) with 37 temperatures T_e from 5eV to 95eV with a step size of 2.5, 10 different total densities with n_A =[1e14, 2e14, 3e14, 4e14, 5e14, 6e14, 7e14, 8e14, 9e14, 1e15], and 40 different initial conditions. This results in a total of 14,800 trajectories. The dataset was collected with 1,000 time steps for each trajectory, leading to 14.8

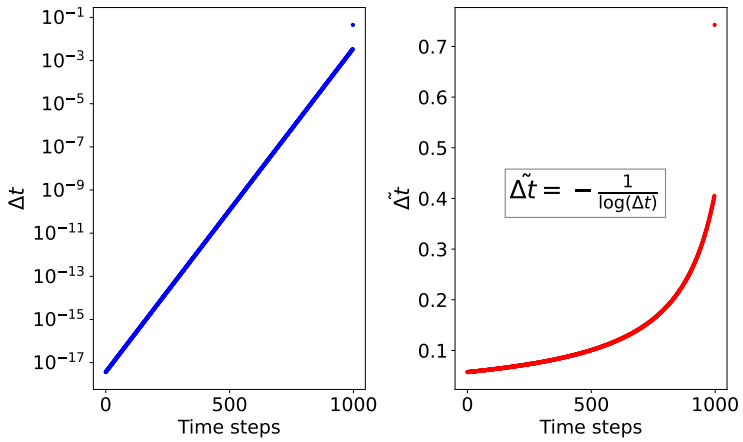

FIG. 5. Scaling of the time steps.

million pairs of (L_n, L_{n+1}) in the flow map neural network training to learn latent dynamics. This large dataset is very expensive to train in practice. Additionally, there are a significant number of time steps that are essentially identity mappings (time steps are less than 500). This poses two challenges: first, the dataset size is substantial, and second, the presence of many identity mappings makes the training process difficult. To address these issues, we performed coarse sampling of the data. The purposes of this approach were to reduce the dataset size and to avoid the identity mappings. Instead of using all 1,000 time step data points, we performed coarse sampling to retain 161 points; see Fig. 7. Note that time steps are selected to fully resolve the sharp transition of the dynamics.

B. Autoencoder Training

The entire model is trained in two steps. First, we train the autoencoder to identify the latent space. Next, we use a flow map neural network (FMNet) to learn the latent dynamics. In the autoencoder architecture, the encoder and decoder both have two hidden layers. The input is the normalized ion population density $\mathbf{W}$ with an input dimension of 94. The output of the decoder is the reconstruction of the ion population density and radiative loss rate $\mathscr{R}_L$, with a dimension of 95. Increasing the number of hidden neurons and the latent space size increases the training cost. For the fully connected layers, sigmoid activations are used. The learning rate has constant decay after every 1000 epochs starting with 0.001. We trained the model for 10,000 epochs, experimenting with different numbers of hidden units and latent space dimensions. To balance accuracy and

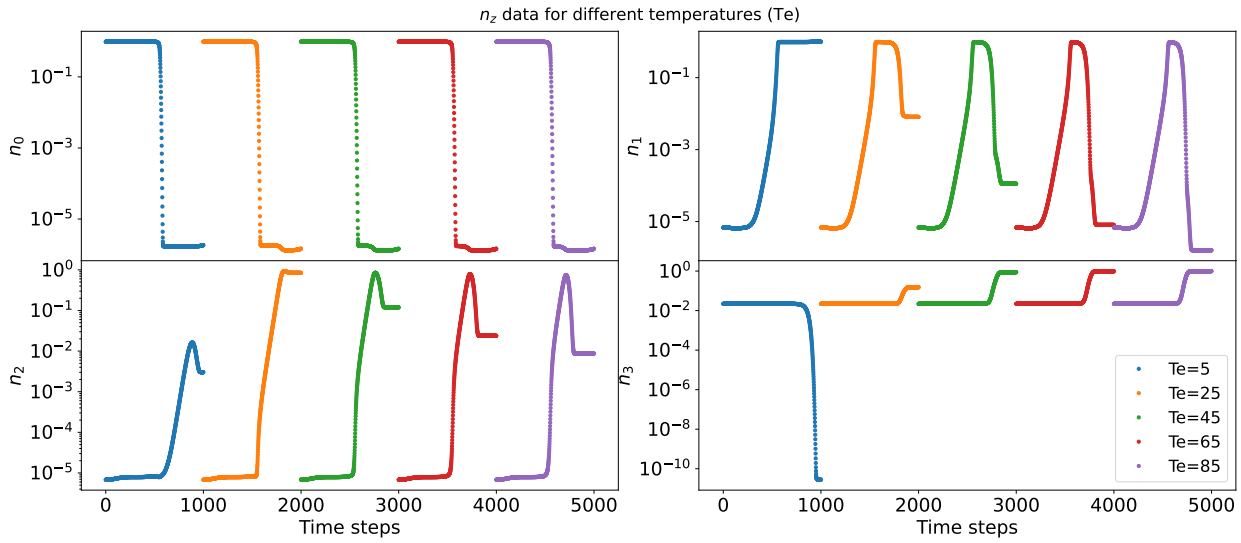

FIG. 6. High fidelity CR solution of charge state n_z. Trajectories from 10 different temperatures T_e are plotted.

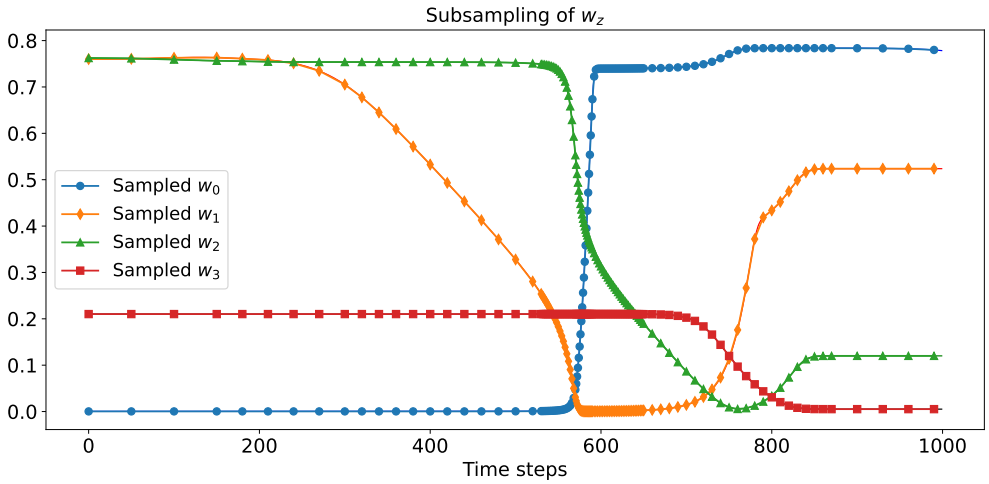

FIG. 7. Resampled data points from the true trajectory.

efficiency, using $h1 = 64, h2 = 32$, and a latent space of dimension 10 (with a black space of dimension 6) provides a reasonably accurate reconstruction and prediction of radiative loss. Fig. 9 shows the prediction of $\mathscr{R}_L$ from the decoder. The model is trained using one NVIDIA A100 GPU. Under this configuration, the training cost is 27 hours. We used this latent space configuration for our flow map training.

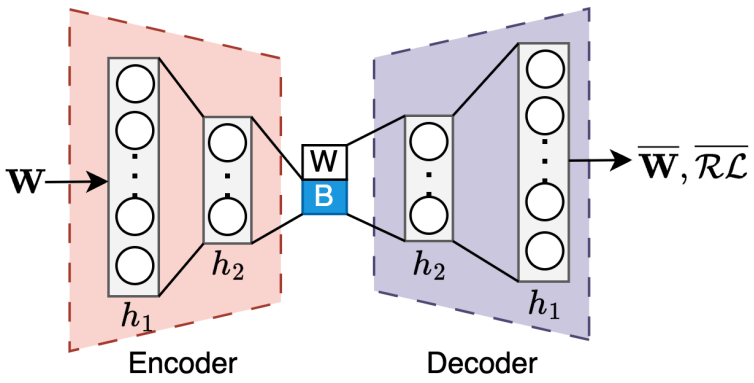

FIG. 8. The autoencoder architecture, h_1, h_2 are the hidden layers with units 64 and 32 respectively

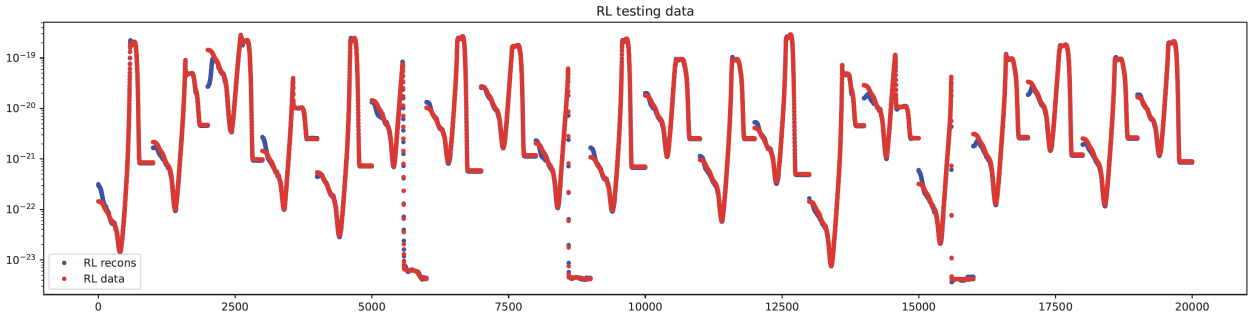

FIG. 9. The radiative loss rate $\mathscr{R}_L$ prediction from the autoencoder neural network. The radiaitive loss prediction for 20 trajectories are plotted. Red points are the true data, blue dots are the prediction from the decoder

C. Flow Map Training

1. Prediction Error

In the prediction phase, we provide the initial condition (latent variable at t_0 found through the encoder) to the FMNet. FMNet then iteratively predicts the latent trajectory, which includes both the charge state dynamics (white space) and the unknown dynamics (black space), see Fig. (10). This latent trajectory is subsequently fed into the decoder to obtain the radiative loss rate $\mathscr{R}_L$. The prediction error of the charge state, used in our model evaluation, is defined as the Mean Squared Error (MSE) at each time step,

$$e_{pred} = \frac{1}{S}\sum_{i=1}^{S} \|n_z^i - \overline{n_z^i}\|^2$$

It is important to note that the training error is computed based on one-step predictions, while the prediction error in the testing phase accumulates at every time step, reflecting the compound effect

of iterative predictions. Using prediction errors to evaluate model performance ensures the robustness and reliability of the model, making it suitable for practical applications in predicting radiative loss rates and understanding charge state dynamics. By leveraging the latent space representation, the reduced-order model effectively reduces computational complexity while maintaining high accuracy. This approach facilitates efficient and accurate simulations in high-fidelity numerical experiments, thereby enhancing the model's utility in real-world scenarios.

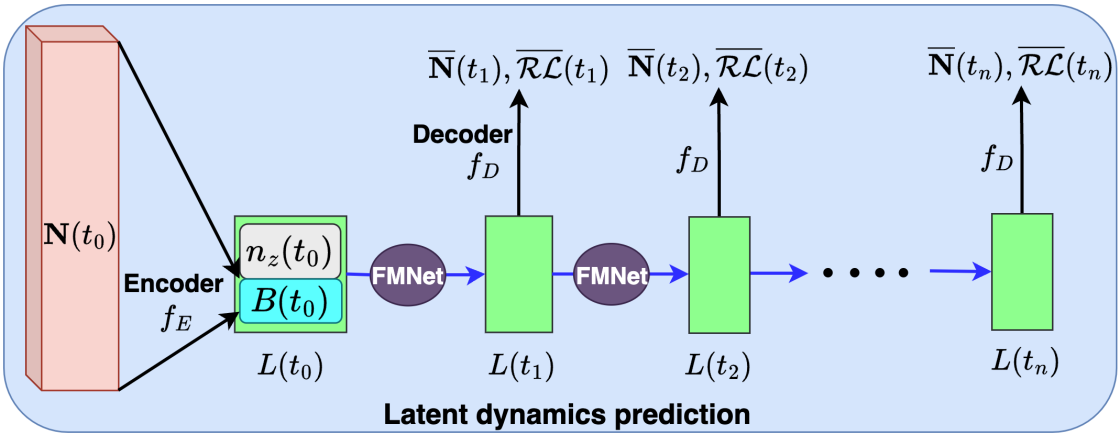

FIG. 10. The iterative prediction scheme of the latent dynamics and radiative loss rate $\mathscr{R}_L$ after the autoencoder and flow map network (FMNet) are trained.

2. *Dynamics Prediction from Different Initial Conditions*

We first evaluate the performance of the FMNet under various initial conditions. We initialize the FMNet with different latent variables, each corresponding to a unique initial condition in the high-fidelity simulation data. The purpose of this evaluation is to assess the model's ability to generalize and accurately predict the CR dynamics from different starting points. We select a representative set of initial conditions that span the range of typical operating conditions for the CR model. For each initial condition, the FMNet iteratively predicts the latent trajectory, capturing the evolution of charge state dynamics and unknown dynamics. The predicted latent trajectories are then decoded to obtain the corresponding radiative loss rates $\mathscr{R}_L$. We quantify the prediction accuracy using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) across all time steps for each initial condition. The results are compared to the ground truth obtained from the high-fidelity simulations. Fig. 11 illustrates the prediction performance for

a subset of initial conditions, showing both the predicted and true trajectories of key variables. Fig. 12 shows the corresponding radiative loss predictions. Our analysis indicates that the model maintains robust performance across a wide range of initial conditions, with prediction errors remaining within acceptable bounds. This demonstrates the model's capability to adapt to different starting points and accurately capture its dynamics.

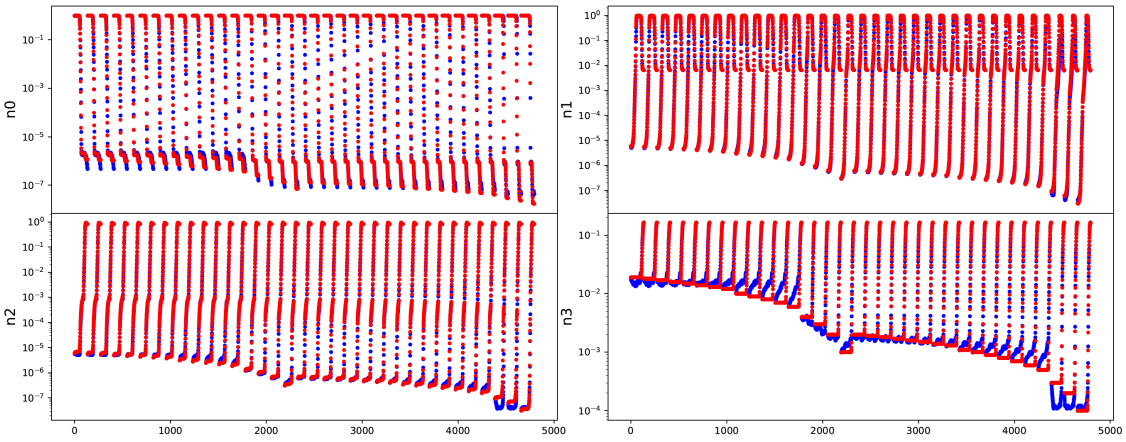

FIG. 11. Charge state n_z trajectory prediction from the flow map neural network (FMNet). Each trajectory represent different initial conditions. Red dots are the true data, and blue dots are the FMNet prediction

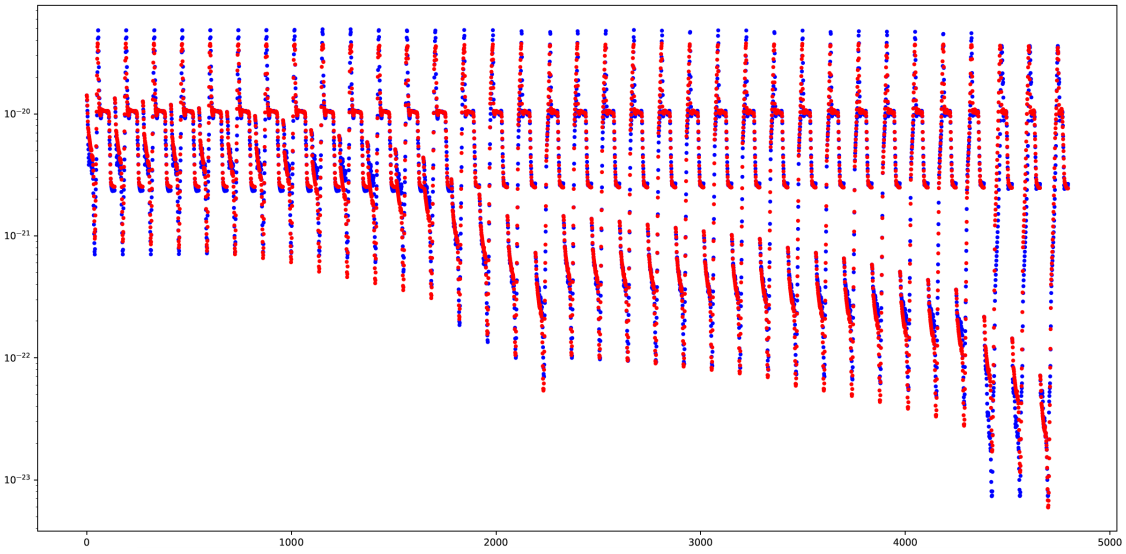

FIG. 12. Radiative loss rate prediction from the decoder after feeding the predicted latent dynamics.

3. *Dynamics Prediction from Different Parameters*

In this section, we extend the evaluation to different parameter settings for total density n_A and electron temperature T_e. These parameters play a crucial role in the behavior of the CR model, influencing the rate of collisional and radiative processes. To assess the model's performance across different parameter values, we generate predictions for various combinations of n_A and T_e. The FMNet is trained to account for these parameters as inputs, enabling it to adapt its predictions based on the specific conditions. We systematically vary n_A and T_e within their respective ranges used in the high-fidelity simulations, Fig. 13 shows the split of the parameters in dataset for training, validation and testing. For each combination of n_A and T_e, the model predicts the latent trajectory and the corresponding radiative loss. We then compare these predictions to the ground truth data, using prediction error. Figs. 14 and 15 show the prediction results charge state and radiative loss. These plots illustrate the model's ability to accurately capture the dynamics under varying conditions. Our findings suggest that the model performs well across a broad spectrum of parameter values, maintaining high accuracy in its predictions. This highlights the model's flexibility and robustness, making it a valuable tool for simulating and understanding the behavior of the CR system under different physical conditions.

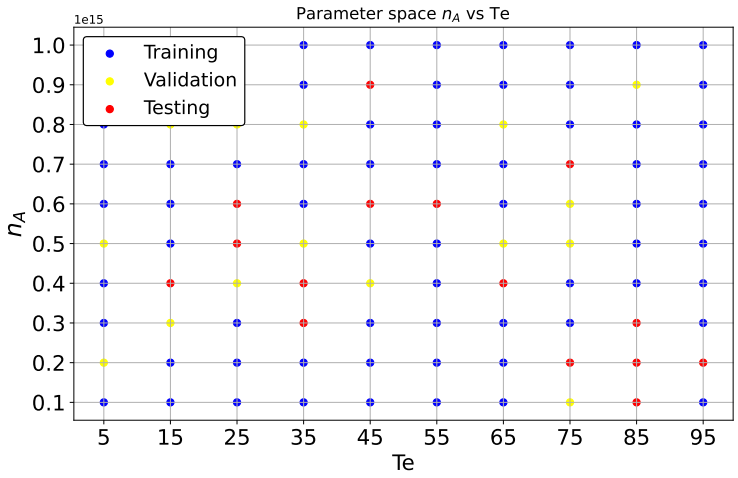

FIG. 13. Parameters sampled for training and prediction test.

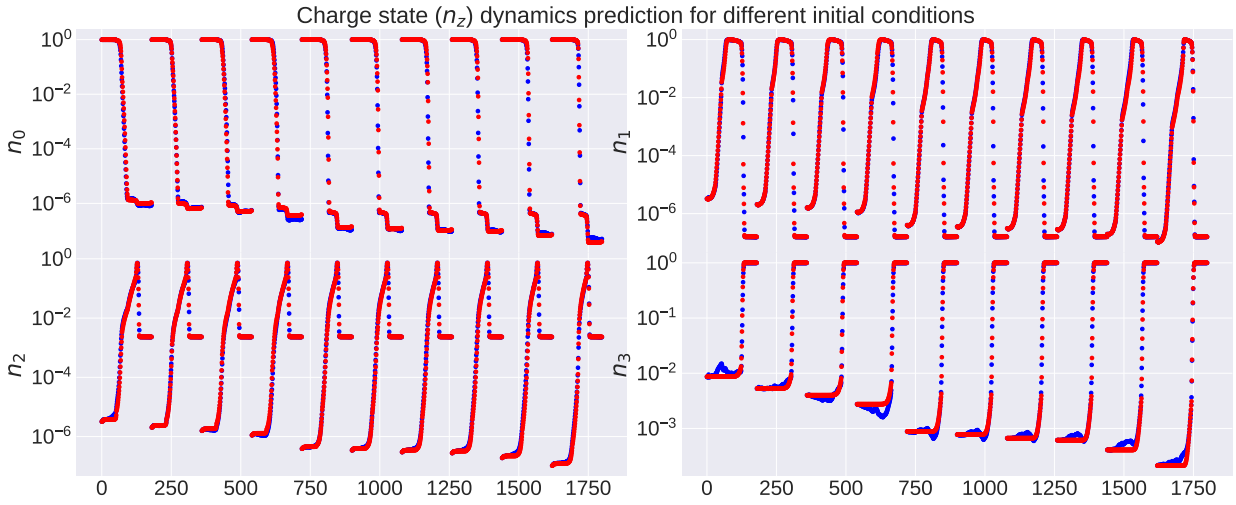

FIG. 14. Charge state trajectory prediction at temperature $T_e = 65$, n_A =5e14 with different initial conditions in the testing dataset, 10 trajectories are plotted. Red represents true data, and blue represents the model prediction.

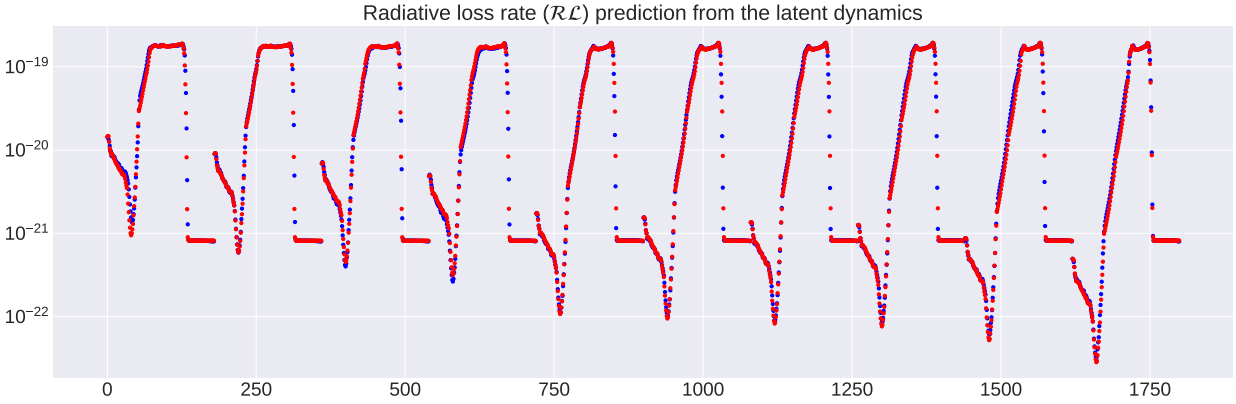

FIG. 15. Radiative loss rate prediction at temperature $T_e = 65$, n_A =5e14 with different initial conditions in the testing dataset. Red represents true data, and blue represents the model prediction.

4. Neural Network Architecture Search

Neural network architecture search (NAS) is a crucial process in the development of ML models, focusing on automating the design of optimal neural network architectures. The idea behind NAS is to systematically explore a vast search space of possible architectures to identify the most effective configurations that meet specific performance criteria, such as accuracy, efficiency, and computational cost.

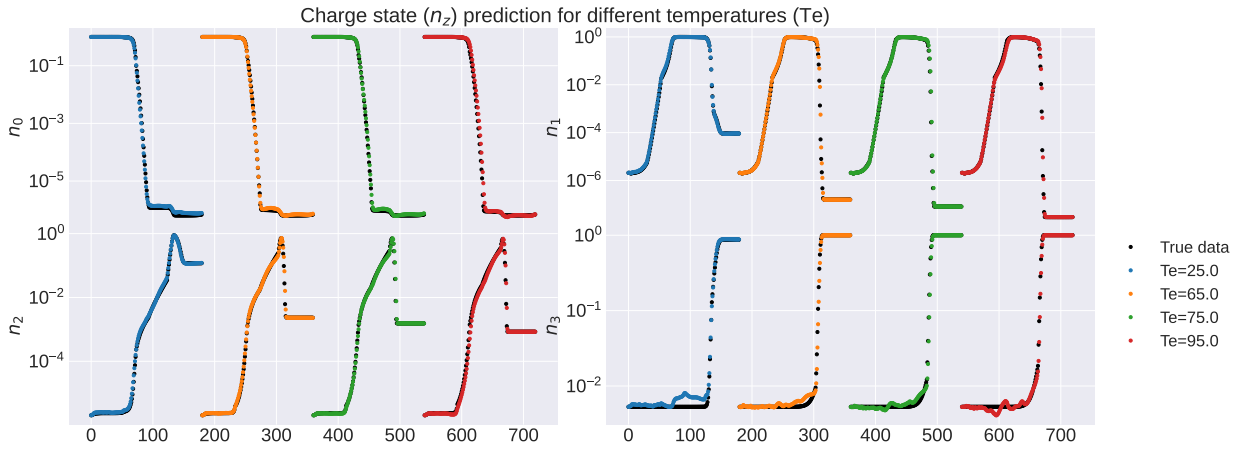

FIG. 16. Charge state trajectory prediction for different temperatures T_e in the testing dataset. Red represents true data, and 4 different colors represent the model prediction from different temperatures.

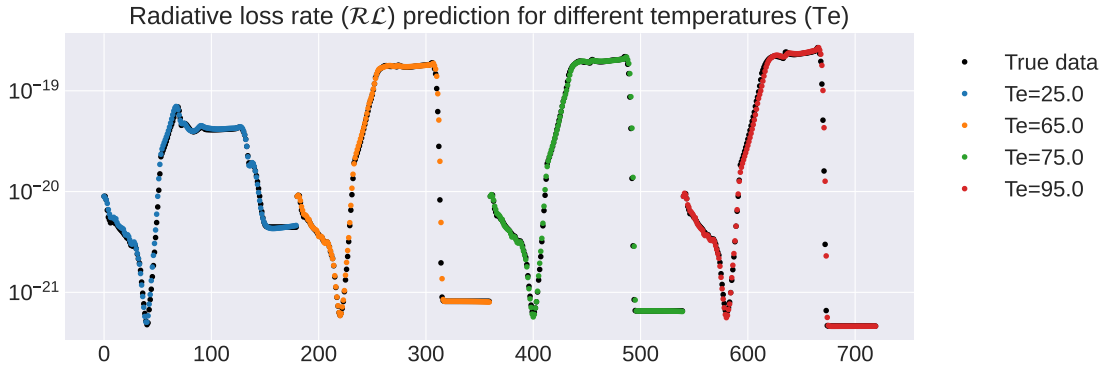

FIG. 17. Radiative loss rate prediction for different temperatures T_e in the testing dataset. Red represents true data, and 4 different colors represent the model prediction from different temperatures.

In practice, NAS involves defining a search space that specifies the range of possible architectures, including the number of layers, the type of layers (e.g., convolutional, fully connected), and the number of units in each layer. The search algorithm then navigates this space to find architectures that maximize a given performance metric on a validation dataset. In our work, we employed a grid search methodology to systematically explore a range of possible configurations. The primary objective was to determine the optimal architecture by varying the number of layers and the number of hidden units within each layer. The number of layers range set was from 2 to 7 layers. This range includes both simpler models with fewer layers, which may train faster and are less prone to overfitting, and more complex models with additional layers, which have the capacity to capture more intricate patterns in the data. For each layer, the number of hidden units was varied

between 16 and 512. By systematically combining these two parameters (number of layers and hidden units), the grid search examined a wide array of architectures and it gives an initial study on the optimal neural network structure for our training data. The results show that the FMNet nearly reaches best prediction error with 3 layers and 256 units for each layer; see Fig. 19.

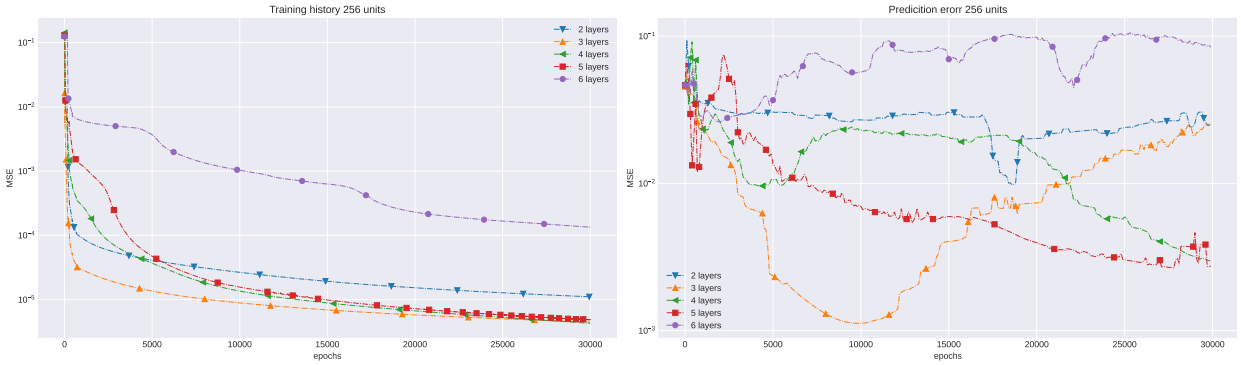

FIG. 18. Training history of the FMNet. Training error (left) vs prediction error (right) from different network architectures.

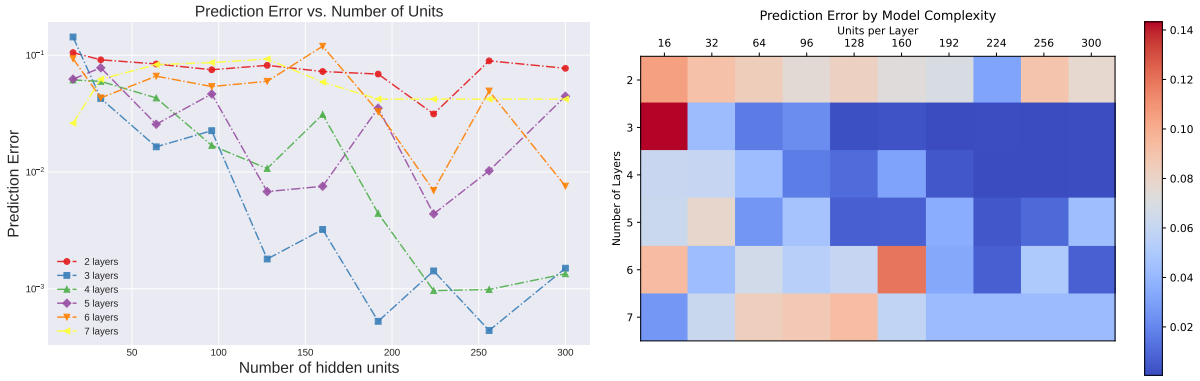

FIG. 19. Prediction error from models with different architecture.

Addressing NAS effectively requires balancing the exploration of diverse architectures with the exploitation of promising configurations. Techniques such as early stopping, weight sharing, and transfer learning are often employed to reduce the computational burden and accelerate the search process. As NAS continues to evolve, it holds the potential to significantly advance the field of neural network design, making it more accessible and efficient. The grid search we used in this study is computationally expensive and may miss optimal configurations lying between grid points. In the future work, we will explore Bayesian optimization and reinforcement learning for more robust search.

5. *Impact of Training Data Size*

The performance of neural networks is influenced by two key factors: the amount of training data available and the complexity of the model architecture. In the previous section, we used a grid search method to find the optimal neural network architecture. In this section, we explore how increasing the size of the training dataset and the complexity of the neural network model impacts prediction performance.

One of the fundamental principles in ML is that larger datasets tend to lead to better model performance. When training a neural network on a small dataset, the model may not have enough examples to learn robust patterns and relationships in the data. As a result, the model may suffer from overfitting, where it memorizes the training data rather than generalizing well to unseen data. By increasing the size of the training dataset, we provide the model with more diverse examples to learn from, which can help improve its ability to generalize. As the amount of training data increases, the model becomes more exposed to different variations and nuances present in the data, allowing it to learn more robust representations. Consequently, we typically observe better prediction performance as the size of the training dataset grows. Since our dataset is parameterized by the n_A and T_e, we use T_e as the benchmark to test the impact of the datasize. We split the dataset according to the T_e values. For the testing dataset, we use T_e =[15, 45, 75, 95]. Our initial training only contains data with T_e =[5, 25, 35, 55, 65, 85] (black dots in Fig. 20). The second training we use the full dataset (blue dots in Fig. 20). The results clearly indicate that increasing the training dataset can lead to better prediction performance for the model, as demonstrated in Figs. 21 and 22.

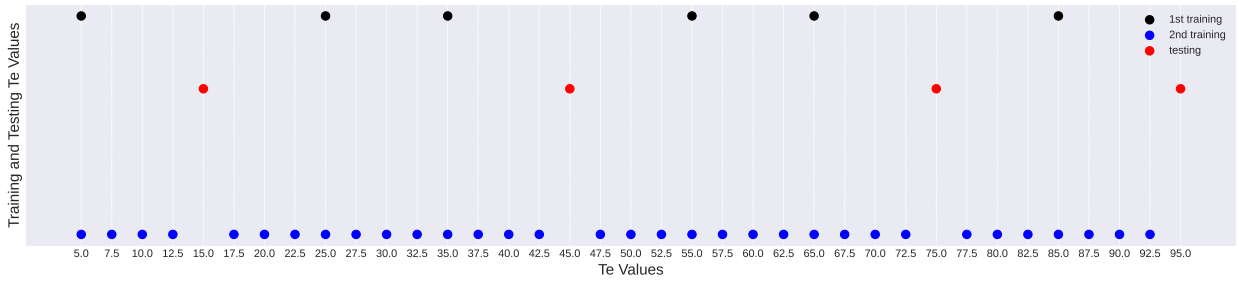

FIG. 20. Data used for different training test

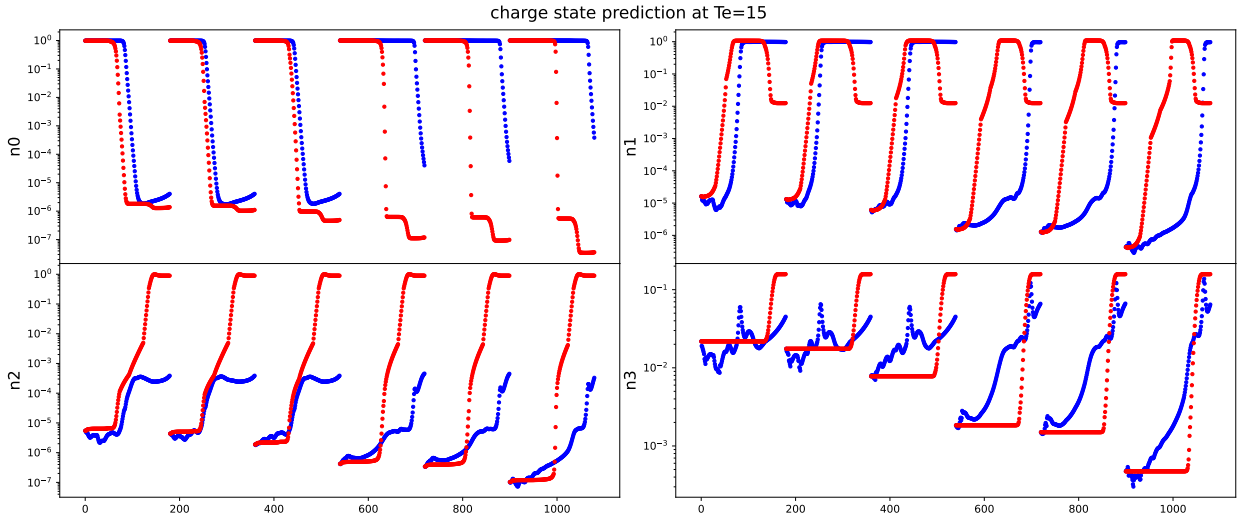

FIG. 21. Charge state prediction from the FMNet trained with first training T_e data (see dark dots in Fig. (20)).

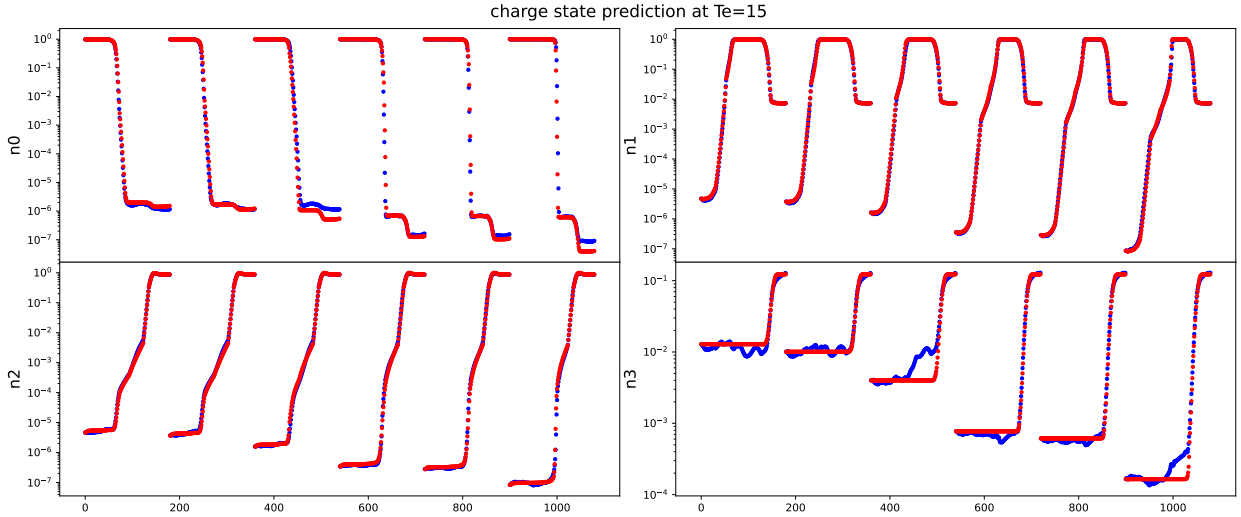

FIG. 22. Charge state prediction from the FMNet trained with full T_e values data.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a physics-assisted surrogate model framework tailored for Collisional-radiative (CR) modeling. Our approach leverages a mixed latent space, comprising a "white space" that enforces known physics constraints, and a "black space" discovered through an autoencoder. Subsequently, neural networks are utilized to learn the dynamics governing this latent space. In the numerical experiments, by thoroughly evaluating the model's performance under

various initial conditions and parameter settings, we demonstrate its reliability and effectiveness in predicting the complex dynamics of the CR model. This comprehensive analysis provides confidence in the model's applicability to real-world scenarios, where accurate and efficient predictions are essential for understanding and mitigating plasma disruptions in fusion reactors. The presented numerical results substantiate the effectiveness of our approach, demonstrating promising accuracy in modeling the CR problem.

In future work, we plan to expand our model to include data from multiple species. Incorporating a broader range of species will enhance the model's applicability and robustness, allowing for more comprehensive predictions of radiative loss rates and charge state dynamics across different plasma conditions. This expansion will inevitably increase the dataset size and complexity, presenting new challenges in terms of computational requirements and training costs which we will use distributed training with multiple GPUs. Additionally, we aim to integrate NODEs into our framework. NODEs offer a powerful approach for modeling continuous-time dynamics, allowing the model to learn the system's evolution directly from data.

ACKNOWLEDGEMENT

This work was supported by the AI/ML program of the U.S. Department of Energy (DOE) Office of Fusion Energy Science (FES). QT was also partially supported by the Mathematical Multifaceted Integrated Capability Center (MMICC) and Data-intensive Scientific Machine Learning programs of DOE Advanced Scientific Computing Research (ASCR).

REFERENCES

[1]Nathan A Garland, Hyun-Kyung Chung, Christopher J Fontes, Mark C Zammit, James Colgan, Todd Elder, Christopher J McDevitt, Timothy M Wildey, and Xian-Zhu Tang. Impact of a minority relativistic electron tail interacting with a thermal plasma containing high-atomic-number impurities. *Physics of Plasmas*, 27(4), 2020.

[2]Nathan A Garland, Romit Maulik, Qi Tang, Xian-Zhu Tang, and Prasanna Balaprakash. Progress towards high fidelity collisional-radiative model surrogates for rapid in-situ evaluation. In *Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020)(Vancouver, Canada)*, 2020.

[3]Nathan A Garland, Romit Maulik, Qi Tang, Xian-Zhu Tang, and Prasanna Balaprakash. Efficient data acquisition and training of collisional-radiative model artificial neural network surrogates through adaptive parameter space sampling. *Machine learning: science and technology*, 3(4):045003, 2022.

[4]P Hakel, ME Sherrill, S Mazevet, J Abdallah Jr, J Colgan, DP Kilcrease, NH Magee, CJ Fontes, and HL Zhang. The new Los Alamos opacity code ATOMIC. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 99(1-3):265–271, 2006.

[5]Xuping Xie, Muhammad Mohebujjaman, Leo G Rebholz, and Traian Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing*, 40(3):B834–B857, 2018.

[6]David Amsallem and Charbel Farhat. Stabilization of projection-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 91(4):358–377, 2012.

[7]William Snyder, Alex Santiago Anaya, Justin Krometis, Traian Iliescu, and Raffaella De Vita. A numerical comparison of simplified galerkin and machine learning reduced order models for vaginal deformations. *Computers & Mathematics with Applications*, 152:168–180, 2023.

[8]Benjamin Peherstorfer and Karen Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.

[9]Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature communications*, 9(1):5, 2018.

[10]Kazuto Hasegawa, Kai Fukami, Takaaki Murata, and Koji Fukagata. Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. *Theoretical and Computational Fluid Dynamics*, 34:367–383, 2020.

[11]Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[12]Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[13]Kai Fukami, Takaaki Murata, Kai Zhang, and Koji Fukagata. Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *Journal of Fluid Mechanics*, 926:A10, 2021.

[14]Kadierdan Kaheman, J Nathan Kutz, and Steven L Brunton. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*, 476(2242):20200279, 2020.

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

[16]Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

[17]Alec J Linot, Joshua W Burby, Qi Tang, Prasanna Balaprakash, Michael D Graham, and Romit Maulik. Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems. *Journal of Computational Physics*, 474:111838, 2023.

[18]Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

[19]Eleni D Koronaki, Nikolaos Evangelou, Cristina P Martin-Linares, Edriss S Titi, and Ioannis G Kevrekidis. Nonlinear dimensionality reduction then and now: Aims for dissipative pdes in the ml era. *Journal of Computational Physics*, page 112910, 2024.

[20]Joshua William Burby, Qi Tang, and R Maulik. Fast neural poincaré maps for toroidal magnetic fields. *Plasma Physics and Controlled Fusion*, 63(2):024001, 2020.

[21]Valentin Duruisseaux, Joshua W Burby, and Qi Tang. Approximation of nearly-periodic symplectic maps via structure-preserving neural networks. *Scientific reports*, 13(1):8351, 2023.

[22]Yuying Liu, J Nathan Kutz, and Steven L Brunton. Hierarchical deep learning of multiscale differential equation time-steppers. *Philosophical Transactions of the Royal Society A*, 380(2229):20210200, 2022.