UNIVERSITY OF CALIFORNIA SAN DIEGO

Predicting temperature and salinity profiles in the upper ocean using a deep learning-based reduced-order model

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Tejas Badakere Gopalakrishna

Committee in charge:

Professor Sutanu Sarkar, Chair
Professor Xuanting Hao
Professor Oliver T. Schmidt

2024

The thesis of Tejas Badakere Gopalakrishna is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

To my parents and brother.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I want to acknowledge my advisor Professor Sutanu Sarkar for agreeing and providing me an opportunity to work under him. His support and insightful discussions have been vital in shaping the direction of this thesis

I am also grateful to Dr. Hieu Pham, who helped provide datasets to work with and multiple discussions during which he gave me a lot of insights and directions to carry forward this thesis. I am also indebted to his eagerness to help me whenever he could.

Finally, I would like to thank my parents and brother for unwavering support and encouragement throughout this journey.

ABSTRACT OF THE THESIS

Predicting temperature and salinity profiles in the upper ocean using a deep learning-based reduced-order model

by

Tejas Badakere Gopalakrishna

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2024

Professor Sutanu Sarkar, Chair

Temperature and salinity profiles in the upper ocean are predicted using non-intrusive reduced-order modeling with deep learning. The data from Large Eddy Simulations, consisting of 8500 profiles of temperature and salinity fields, spanning over two weeks, are used to train the deep learning architecture. The entire two-week period is divided into phases: an active and a break phase. The deep learning architecture is trained in a two-step process. First, dimensionality reduction of the high-dimensional data is achieved using a Long Short-Term Memory(LSTM) based autoencoder. In the second step, another RNN-based architecture, the Gated Recurrent Unit (GRU), is used to evolve these reduced representations over time and then project them

back into the high-dimensional space using the decoder, to recover the actual temperature and salinity fields. The primary goal of this work is to evaluate the prediction capability of the trained network under two scenarios: 1) using data from both phases for training, and 2) using only the active phase for training to predict the rapid evolution of temperature and salinity profiles between days 3 and 5. The secondary goal is to assess the model's long-term prediction capability by applying the same architecture used in scenario (2) of the primary goal to predict temperature and salinity profiles in the break phase and part of the active phase and highlight the importance of choosing the training data in predicting the physical phenomena.

Chapter 1

Introduction

Understanding the dynamics of the upper ocean is vital for studying air-sea interactions, which involve the transfer of momentum, mass, and heat. These processes play a key role in atmospheric and oceanic circulation models and are crucial for predicting future climate conditions, including atmospheric CO_2 levels (Donelan et al., 2004; Wanninkhof et al., 2009). The upper few meters of the ocean are typically measured using the Air-Sea Interaction Profiler (ASIP) (Ward et al., 2014), equipped with high-resolution sensors that capture small-scale temperature and salinity profiles. However, obtaining these instruments is expensive, and it becomes impractical to collect data across various horizontal locations. Additionally, it is also necessary to infer information about the upper ocean in the past from surface flux measurements.

One possible approach to address these challenges is to perform Large Eddy Simulations (LES) to estimate the required quantities. However, LES can be computationally expensive and time consuming. Another option is to use one-dimensional parameterization methods, such as first order K-profile parameterization or second-order models like the k-ε turbulence model (McWilliams et al., 2009; Rodi, 1987). Despite their usefulness, these parameterizations are often less accurate in their prognostic predictions under realistic surface forcing conditions, producing varied and scattered results (see Figure 1.1).

Reduced Order modeling is one of the choices when the high-fidelity model is expensive to train. One of the methods is the projection-based method called Proper Orthogonal Decom-

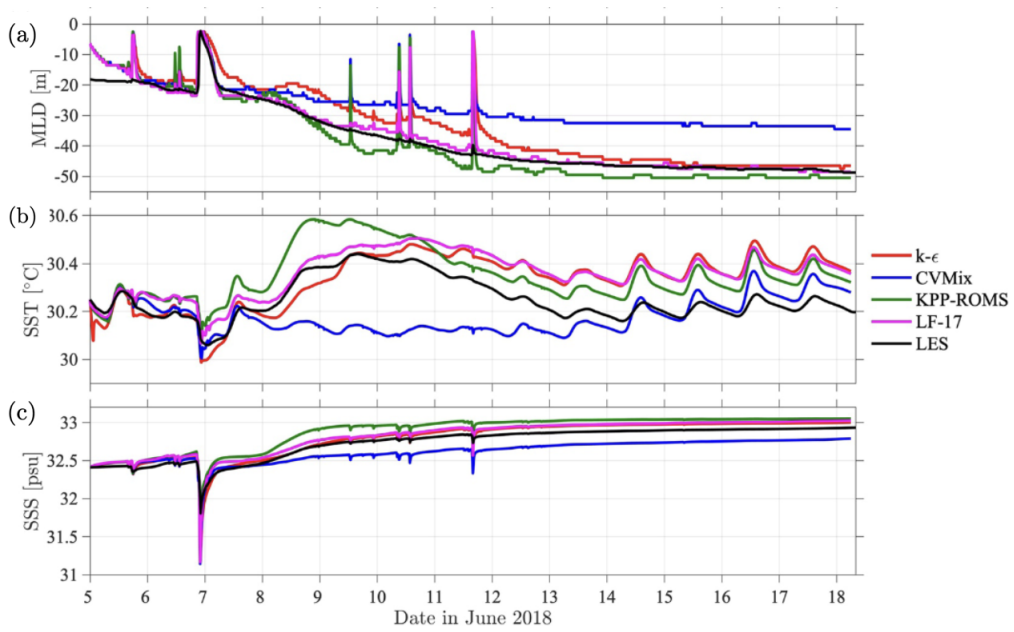

Figure 1.1. Disparity in (a) Mixed Layer Depth (b) Predictions of Sea Surface Temperature, (c) Sea Surface Salinity among the 1D mixing models when compared to the large-eddy simulations (Pham et al., 2023).

position (POD) initially introduced by Lumley (1967). POD assumes that a high-dimensional field can be approximated by a low-dimensional representation defined by a set of reduced basis. The system of PDEs is then projected onto this reduced basis using the POD-Galerkin projection (Aubry et al., 1988) resulting in an equation for the POD coefficients, which simplifies to an ODE that can be solved easily. However, a drawback of the POD-Galerkin projection is its inability to handle strong spatial nonlinearities, leading to instabilities in certain flows, even when the actual flow is stable (Aubry et al., 1988). While there are other projection-based variants that address these instabilities (Carlberg et al., 2011), they still require access to the underlying governing equations, making them intrusive in their formulation.

Over the last decade, the advent of deep learning algorithms has provided a promising approach to solving the challenges in reduced-order modeling. Deep learning is based on the Universal Approximation Theorem, which states that regardless of the complexity of the relationship between input and output, a deep learning architecture can approximate this relationship with a suitable function. This theorem has driven significant breakthroughs across

various fields, including computer vision, speech recognition, natural language processing, and medical technology (LeCun et al., 2015).

In the field of fluid mechanics, deep learning has been applied to a range of applications, including improving closure problems in Reynolds-Averaged Navier-Stokes (RANS) simulations (Wang et al., 2017; Ling et al., 2016; Wu et al., 2018), determining constants in RANS models (Tracey et al., 2015), and uncertainty quantification (Zhu and Zabaras, 2018). Among these applications, one of the most promising is model order reduction. For 2D flow problems, convolutional neural networks (CNNs) are frequently used to reduce the dimensionality of high-fidelity simulations to the flow's intrinsic dimensions. This is typically done through the construction of an autoencoder, a model designed to perform identity mapping. To advance the dynamics of the reduced basis coefficients, sequential models such as Long short-term memory (LSTM), Gated recurrent unit (GRU), and, more recently, Transformer architectures (Geneva and Zabaras, 2022; Solera-Rico et al., 2024) are commonly employed. As these methods do not require access to the governing equations, they are categorized as Non-Intrusive Reduced Order modeling.

Training a reduced-order model using deep learning typically involves jointly training two networks: one to learn the reduced representation of the full-order system and another to capture the temporal evolution of this reduced representation. An LSTM architecture was used to sequentially predict the POD coefficient in Wang et al. (2018). In Gonzalez and Balajewicz (1808), a deep convolutional neural network, along with an LSTM network, was jointly trained to evaluate the evolution of the quantity of interest. A deep convolutional encoder-decoder approach utilizing dense layers was implemented in Mo et al. (2019), where the authors autoregressively predicted the groundwater contaminant transport based on previous concentration fields, conductivity field, and the source location. A Recurrent-U-net, consisting of a Residual Network (He et al., 2016) and a ConvLSTM (Shi et al., 2015) to evolve the latent space dynamics, was developed in Tang et al. (2020) to predict dynamic pressure and saturation maps, and well rates. In Mohan et al. (2020), dimensionality reduction and temporal evolution were

handled separately by using a convolutional autoencoder to reduce the homogeneous, isotropic turbulent field, while a CONVLSTM (a variant of LSTM) was used to evolve temporal dynamics. A similar approach was applied in Maulik et al. (2021) to predict the temporal dynamics of the shallow water equation. temporal CNN(TCNN) and temporal convolutional autoencoder (TCAE) architectures were used in Xu and Duraisamy (2020) to predict the temporal dynamics and new parameter predictions for four different fluid dynamics problems.

In this study, a similar approach is used to build a ROM. A network, treated as two stages of the RNN model, is trained to perform dimensionality reduction and temporal evolution of the latent space. For the first stage, An LSTM-based autoencoder is used to reduce the dimension of the temperature and salinity profiles to a low dimensional latent space. In the second stage, a Gated Recurrent Network (GRU) is employed to evolve the dynamics of the latent space.

1.1 Data for training

The application of RNN-based reduced order modeling is carried out on the study of the MISO-BOB field experiment, which was conducted during the summer of 2018 (Shroyer et al., 2021). A detailed LES study of temporal variability of turbulent mixing during the same period was performed in Pham et al. (2023). The study was conducted on the days starting from 5th to 18th June, covering a total period of 2 weeks. The study utilized the surface fluxes, including solar and non-solar heat flux, surface wind stresses in zonal and meridional direction, precipitation and evaporation of fresh water, and the background conditions like initial conditions of salinity and temperature from the field experiment. The study examined the combined effects of wind, surface waves, diurnal heat fluxes, and precipitation on the evolution of turbulent mixing in the mixed Layer. Three LES case studies were conducted, that studied the effect of the thermal inversion layer and Langmuir turbulence. A detailed explanation of these cases and their comparison with the other 1D Ocean models can be found in the paper (Pham et al., 2023). In this work, a reduced order model is developed for only one of the three cases, which had both

the thermal inversion layer and Langmuir turbulence.

The LES simulation that was conducted for 2 weeks was averaged out in the horizontal plane under the assumption of horizontal homogeneity. As a result, 8,500 profiles of temperature and salinity were generated as 1D columns. The active phase, lasting seven days, was characterized by increased wind stresses, freshwater influx from rain, and reduced solar radiation, all of which led to intense turbulent mixing and rapid deepening of the mixed layer. The next 7-day period was the break phase, where the surface conditions were unfavorable to turbulent mixing in the mixed layer, resulting only in the gradual deepening of the mixed layer depth, with the shape of temperature and salinity profiles remaining largely unchanged. However, sea surface salinity gradually increased in the break phase. The break phase also consisted of diurnal fluctuation in the heat flux due to the presence of an abundance of sunshine. Figure 1.2 illustrates the evolution of surface forces over 2 weeks. The primary aim of this work is to predict the temperature and salinity profiles, mainly during the active phase when turbulent mixing was prominent. The model's predictive capacity is tested through two different studies: the first utilizes data from both the active and break phases, while the second uses data solely from the active phase to accelerate training. The importance of training data is analyzed in performing long-term predictions.

1.2 Outline of this thesis

In this work, the profiles for temperature and salinity are represented in a 1D column. Therefore, an LSTM-based autoencoder is used to reduce the dimensionality of the vertical profiles. Subsequently, a GRU-based autoencoder is employed to model the dynamics of the latent space. Using these architectures together, both salinity and temperature profiles are predicted for intermediate and future days.

Chapter 2 provides a detailed explanation of the various architectures used in this work.

Chapter 3 outlines the overall procedure followed to build the Reduced Order Model

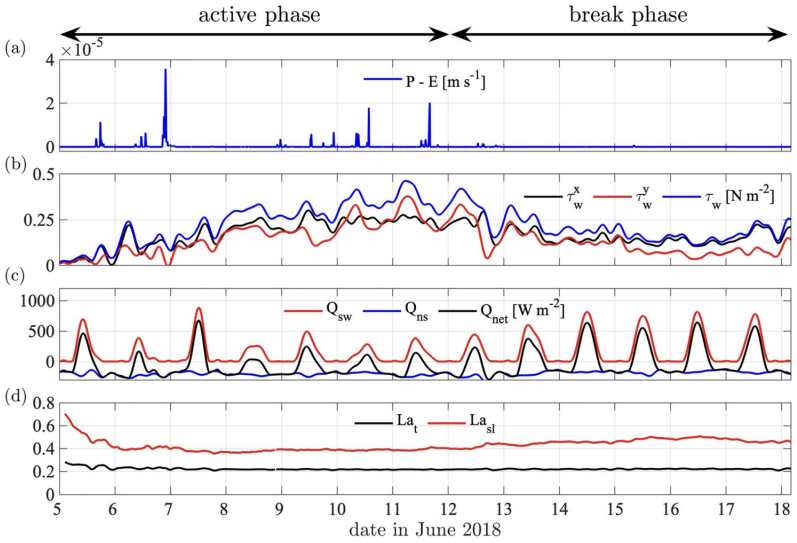

Figure 1.2. Evolution of surface forces over the period of 2 weeks.

using the architectures described in Chapter 2.

Chapters 4 and 5 discuss two different cases: Case 1: Results are provided for the model trained on both active and break phase data. Case 2: Results are provided for the model trained only on active phase data, and a comparison is made with case 1.

Chapter 6 explores the model's ability to predict future states, focusing on the final three days of the break phase.

Chapter 7 examines the long-term prediction capabilities of the RNN-ROM model and emphasizes the significance of selecting appropriate training data for long-term predictions.

The thesis concludes with a summary of the findings.

Chapter 2

Recurrent Neural Networks

Recurrent neural networks (RNNs) are sequential models that take sequential data as input at each time step and output values at every time step. The advantage of such a network is that it models the temporal dependencies between the inputs, unlike a Multi-Layer Perceptron (MLP), where inputs at different time instances are modeled as one instance without accounting for their temporal dependencies. Historically, RNNs have had a wide range of applications in machine translation and speech recognition, which involve sequential data for predictions. Therefore, RNNs are suitable models for capturing the temporal evolution commonly observed in fluid flows.

However, typical RNNs suffer from the problem of vanishing gradients during backpropagation through time, making it extremely difficult to train long-term dependencies (Pascanu et al., 2013). To address the vanishing gradient problem, the Long short-term memory (LSTM) network was introduced to capture long-term dependencies between inputs and outputs. A variant of LSTM called the Gated recurrent unit (GRU), was developed to overcome some of the drawbacks of LSTM. Both models are introduced below.

2.1 Long-short-term-memory

The Long short-term memory(LSTM) (Hochreiter and Schmidhuber, 1997) architecture was introduced to overcome the problem of vanishing gradients in RNNs. LSTM addresses

this issue by incorporating different gates to control the flow of information. These three gates regulate information so that the model can determine what information to retain and what to discard. Traditional RNNs do not retain previous cell states; instead, they compute the current cell states based on current inputs and previous hidden states. The difference between hidden states and cell states is that hidden states pertain only to the current cell, whereas cell states contain information from both the current state and previous states, which is absent in traditional RNNs.

The architecture for LSTM is outlined below in Figure 2.1. A single LSTM cell has three different gates: an input gate, an output gate, and a forget gate. These three gates ensure the flow of information from the current time step, the output at the current time step, and information from past time steps to be either retained or discarded. Equations 2.1-2.6 represent the different operations taking place in a single LSTM cell. The hidden state is denoted by h_t, and the cell state is denoted by C_t. The input gate i_t adds additional information to the cell state, the forget gate deletes information from the previous states, and the output gate o_t regulates information from the current cell state to the output of the cell.

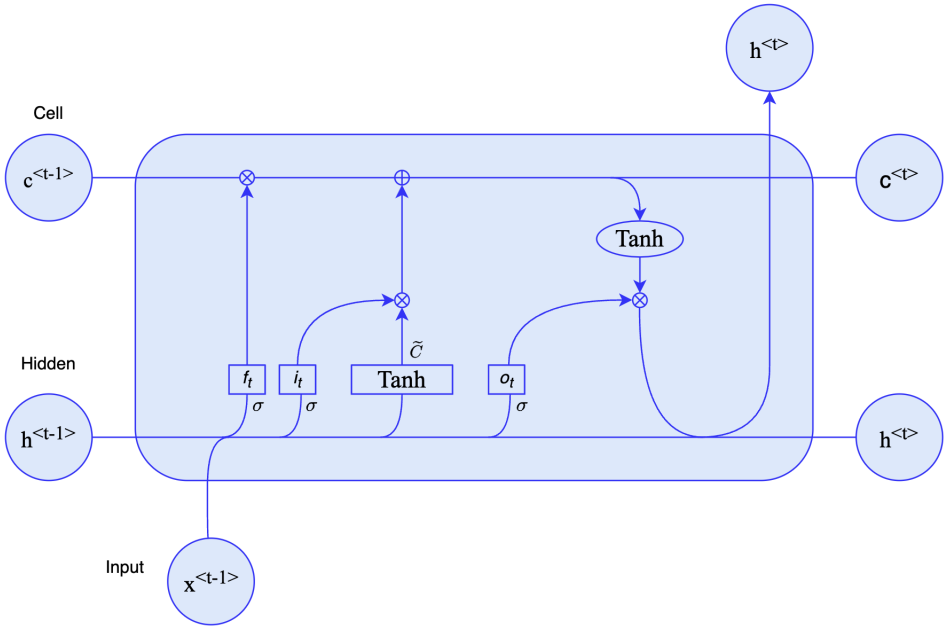

Figure 2.1. LSTM architecture for a single cell.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{2.1}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{2.2}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{2.3}$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{2.4}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{2.5}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \tag{2.6}$$

Where W is the weight matrix, b is the bias term for each equation, σ is a sigmoid function given by $\frac{1}{1+\mathbf{e}^{-\mathbf{x}}}$, and $\odot$ is called the Hadamard product, or colloquially called the element-wise product between two vectors.

2.2 Gated Recurrent Unit

In this section, another recurrent network called the Gated Recurrent Unit (GRU) is introduced. LSTM has separate units to retain past cell states. However, in the GRU, a single gate controls the information from the current state and the previous cell state. Another difference is that the amount of memory seen from the previous unit is regulated by the output gate in LSTM, whereas in the GRU, the memory unit (hidden state) is fully exposed (Chung et al., 2014). Therefore, in a GRU, the cell state and the hidden state are the same.

Figure 2.2 shows the GRU architecture. The update gate i_t regulates the information from the previous cell state and the current input.

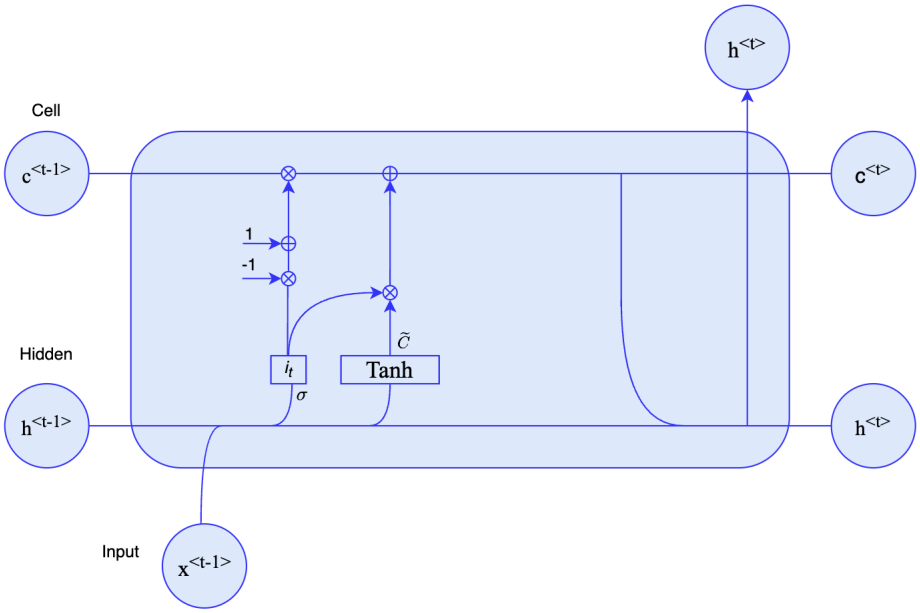

Figure 2.2. GRU architecture for a single cell

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{2.7}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{2.8}$$

$$\mathbf{C}_t = (1 - \mathbf{i}_t) \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{2.9}$$

$$\mathbf{h}_t = \mathbf{c}_t \tag{2.10}$$

Both these RNNs are used for model reduction and prediction of the temporal evolution of the temporal coefficient. Each architecture has its advantages; however, the original authors do not prefer one architecture over the other (Chung et al., 2014). In this work, however, it matters which architecture to use, and the reasons for this preference will be stated later.

2.3 Autoencoders

Often, data in the real world has very high dimensionality. This data is often poorly processed, and the intrinsic dimension of a single observation is much less than the measured

dimension. Hence, it is necessary to select the right dimension for the data. One such method in the field of fluid dynamics is called Proper Orthogonal Decomposition (POD), introduced by Lumley (1967). However, POD assumes that the underlying data lies in a linear subspace, which is often not the case. There are numerous other nonlinear dimensionality reduction techniques (Tenenbaum et al., 2000; Roweis and Saul, 2000; Van der Maaten and Hinton, 2008) that learn a nonlinear manifold where the data resides. However, these approaches do not have a straightforward method to project the data back to its original dimension, which is crucial for analyzing fluid dynamics data. Therefore, in this section, a deep learning-based dimensionality reduction architecture called an autoencoder is introduced.

2.3.1 Autoencoder architecture

An autoencoder consists of two parts: an encoder that learns the function for encoding data, and a decoder that decodes back to the original space. Theoretically, the autoencoder learns an identity mapping $\mathbf{x} \xrightarrow{\phi} \mathbf{x}$, where ϕ is given by a nested function $\phi_{dec}(\phi_{enc})$. Mathematically it can be expressed as

$$\mathbf{x} \approx \phi_{dec}(\phi_{enc}(\mathbf{x})) \tag{2.11}$$

where ϕ_{enc} represent function for encoder part of the autoencoder and ϕ_{dec} represent function for decoder part of the autoencoder.

Figure 2.3 shows the architecture of an autoencoder. A typical encoder has a decreasing number of neurons in subsequent layers. The layer with the fewest neurons is called the bottleneck layer, which contains the encodings of the original high-dimensional data. A typical decoder has an increasing number of neurons in subsequent layers. The input to the decoder is the encoding learned by the encoder. The last layer typically has the same number of neurons as the input to the encoder to reconstruct the original high-dimensional data, though this is not necessary if dimension reduction is not the goal.

In practice, both the encoder and decoder are trained as a single unit during training. The autoencoder discussed in this chapter, and its application in the next chapter corresponds to the vanilla autoencoder where Mean Squared Loss is used for training. Other variants of autoencoders such as Variational Autoencoder (VAE) (Kingma and Welling, 2013) and Wasserstein Autoencoder (WAE) (Tolstikhin et al., 2017) are not discussed as they are beyond the scope of this work.

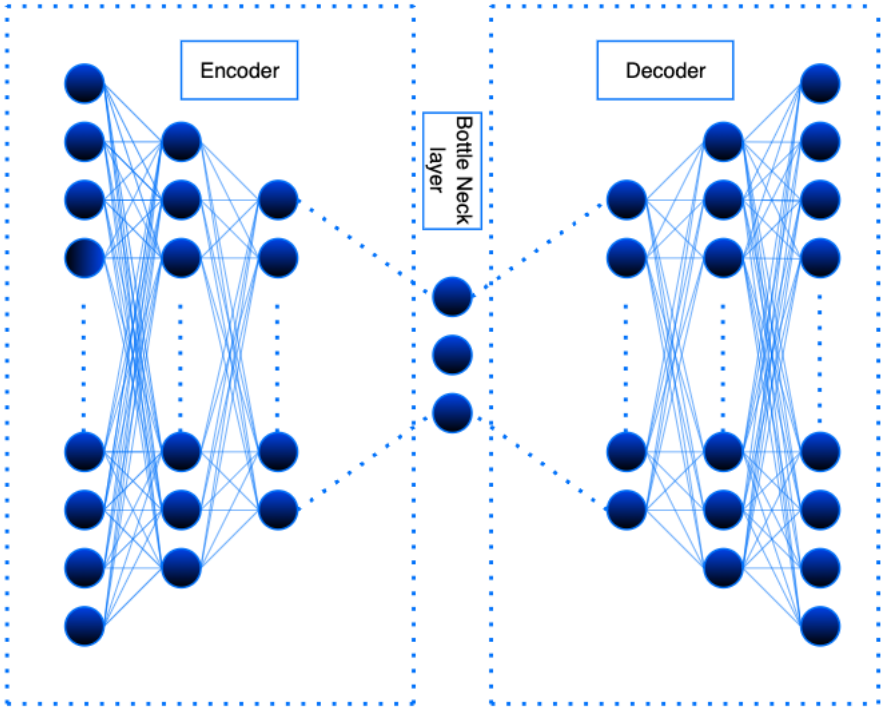

Figure 2.3. A Typical Autoencoder architecture.

A Multilayer perceptron, colloquially known as a neural network, is one way to construct an autoencoder. Other architectures, such as convolutional autoencoders and LSTM-based autoencoders, can also be used. The principle behind the autoencoder remains the same across these architectures.

In the next chapter, we will discuss the application of an LSTM-based autoencoder to reduce the dimensions of temperature and salinity profiles, and how this is implemented.

Chapter 3

Reduced Order Modeling for upper ocean turbulence

In this chapter, we will delve into the detailed process of Reduced Order Modeling (ROM) using deep learning and how we predict the temporal evolution of coefficients with deep learning. The entire procedure is developed in two stages.

- The first stage involves reducing the dimensionality of the temperature and salinity profiles using an autoencoder, specifically an LSTM-based autoencoder, due to the 1-dimensional nature of the flow. This stage can be linked to performing nonlinear dimensionality reduction.

- In the second stage, a GRU is employed to predict the temporal evolution of coefficients based on the previous k coefficients. This stage can be linked to performing a multi-step method such as Adams-Bashforth (Butcher, 2016), which uses a partial differential equation to evolve the quantity of interest over time.

Since we do not have a PDE that governs the evolution of the coefficient, we perform the task non-intrusively with the help of deep learning by using a Gated Recurrent Unit. Since the output of the GRU/LSTM is conditioned on the current and previous inputs, an attempt can be made to see how they perform in evolving the temporal coefficient in time.

3.1 LSTM-Autoencoder

In the previous chapter, we detailed the working principle of a general autoencoder and its interpretation. This section deals with its application and how autoencoder architecture can be made using LSTM units. Temperature and salinity are trained in a separate network. Hence, the reduction of only the Temperature profile is shown in this section. Reduction of salinity follows a similar procedure. In the later sections, a detailed procedure is explained for both temperature and salinity predictions.

Figure 3.1 represents the LSTM-based autoencoder for temperature profiles. High-dimensional 1d temperature profiles are passed sequentially, with the sequence being the temperature values at various spatial locations in the negative z direction. The LSTM encoder outputs a value at every timestep[1]. However, only the output at the final step is considered for the encoding, which is then passed to the decoder to recover the full temperature profile. For the decoder, the output at every timestep represents the temperature at each location in the negative z direction.

The output at each timestep after the hidden layer will not be equal to the dimension of the encodings or latent dimension (which corresponds to the number of neurons in the hidden layer). Therefore, the output at each timestep is passed through a fully connected neural network to reduce the dimension to the required latent dimension (denoted as z in this study). In the later stage of the decoder, the LSTM tends to forget the encoding at the later timesteps. To address this, the encodings are passed as input to the decoder at each timestep.

Various hyperparameters in constructing the LSTM-autoencoder are considered. The important ones in this study were the number of hidden units, the number of LSTM layers, the learning rate, the learning rate decay, batch size, regularization constant, and ”decay applied”, which specifies the number of epochs after which the learning rate decay is applied. These were decided based on trial and error which gave the least amount of validation error. The validation

[1] In the context of the LSTM autoencoder, a timestep corresponds to a spatial step, with each timestep representing a different spatial location

test was chosen closer to the test set as the distribution between the two sets would be closer. Therefore, choosing the optimal hyperparameter corresponding to the validation set ensures the best performance in the test set.

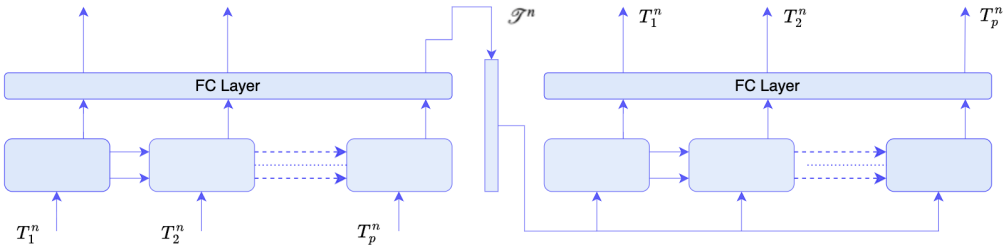

Figure 3.1. LSTM based Autoencoder.

In the figure 3.1, T_1^n, T_2^n...T_p^n correspond to temperature values at each spatial location in the z-direction, passed sequentially to the LSTM encoder at a particular time instance. $\mathscr{T}^n$ corresponds to the encoding at the final step of the LSTM encoder that acts as the reduced representation of Input space. corresponds to the encoding at the final step of the LSTM encoder. This learned representation is then passed to the next stage to learn the mapping between n-k-1 profiles to n+h profiles, where n corresponds to time instances, k, and h are chosen heuristically and are called the past time window and predictive horizon, respectively (Mohan et al., 2019). The next section deals with learning a function for this mapping using a Gated recurrent Unit.

3.2 GRU to predict $\mathscr{T}^{n+1,...n+h-1}$

Our goal is to find the temporal evolution of these embeddings, which corresponds to the temporal evolution of temperature and salinity profiles in the physical space. The steps to achieve this task are outlined next. After encoding the high-dimensional profiles, k of these previous encodings (time window) are chosen to predict the next h (time horizon) encodings. This involves feeding each of these k encodings sequentially into the GRU at every timestep, accompanied by the respective boundary condition, to acquire their temporal encodings. Adding boundary conditions ensures that the encodings are learned parametrically, meaning the encodings are

conditioned not only on the previous k encodings but also on the previous boundary conditions. Subsequently, a fully connected layer is employed to reduce the dimension to zh, where z is the dimension of the latent space and h is the time horizon. The reason for predicting h timesteps at a time instead of one is that it improves long-term predictions by preventing the propagation of error from each successive prediction (Mücke et al., 2021). The output of the GRU is then compared to the ground truth[2], and a mean squared error loss function is used to train the model.

Also, GRU architecture was chosen here because, unlike the previous stage where the total number of timestep corresponded to the number of spatial location p, which is greater in number, it made sense there to use an LSTM instead of GRU since LSTM has better performance in capturing the long term dependencies than GRU. However, in the case of temporal predictions, the total number of timesteps is k which is less than the number of spatial dimensions p, and hence a GRU was used to save the memory cost and speed up the training process.

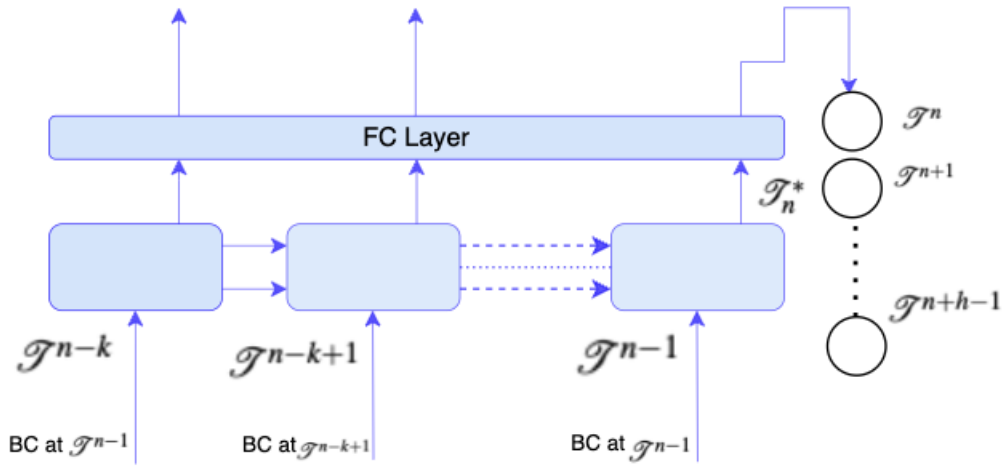

Figure 3.2. GRU architecture implemented for temporal prediction.

In the figure 3.2, $\mathcal{T}^{n-k}$, $\mathcal{T}^{n-k+1}$, $\mathcal{T}^{n-1}$ corresponds to spatial encodings for k previous timesteps. BC is the abbreviation for the boundary condition corresponding to that timestep. $\mathcal{T}^{n}$, $\mathcal{T}^{n+1}$, $\mathcal{T}^{n+h-1}$ corresponds to h future encodings predicted by the GRU model.

[2]Ground Truth refers to the reduced representation of the states obtained from LSTM-Autoencoder

3.3 Training and deploying LSTM-GRU model

A general training procedure is provided in this section. In the upcoming subsections, more details on training temperature and salinity profiles are given separately. The entire training is carried out in two stages, with each stage being independently trained. The entire data is normalized between 0 and 1 which helps in better convergence of the optimizer.

The first stage involves reducing the dimension of the profiles. As mentioned in Chapter 1, the total number of spatial dimensions in the negative Z direction from the LES simulation is 240. However, most of the points are redundant for the training and hence an interpolation is carried out to 50 vertical points. Hence, there are only 50 timesteps to be dealt with. These points were chosen so that interpolating back to the original grid gave a minimum error. This was achieved using Scipy's griddata module. A cubic interpolation scheme was implemented. Throughout the rest of this work, the time window (k) and the time horizon (h) are assumed to be equal, with $k = h = 10$, unless stated otherwise. The entire training was conducted using the PyTorch deep learning framework.

Procedure during training

Input: N_{Train}, N_{val}, High-dimensional input sequences[3] $X^{n-1}, X^{n-2}, \ldots, X^{n-k}$, Boundary conditions.

Output: Reduced representations $\chi^{n-1}, \chi^{n-2}, \ldots, \chi^{n-k}$ and approximated High-dimensional input sequences $X^{n-1}, X^{n-2}, \ldots, X^{n-k}$ and reduced repredentation for next h states $\chi^{n}, \chi^{n+1}, \ldots, \chi^{n+h}$

1. **LSTM encoder training:**

   (a) Sequentially input the high-dimensional data $x_1^n, x_2^n, \ldots, x_p^n$ into the LSTM encoder. The input matrix will have the shape $N_{Train}.N_p.1$ where N_{Train} is the total number of

[3]Here x's refers to quantity of interest. In this study, it would be the temperature and salinity field.

training profiles, N_p is the total number of vertical points.

(b) Train the autoencoder by minimizing[4] the mean squared error (MSE) loss:

$$\arg\min_{\theta} \frac{1}{N_{Train}p} \sum_{i=1}^{N_{train}} \sum_{j=1}^{p} || \left(\phi_{dec}(\phi_{enc}(X^i(z_j);\theta_1) - X^i(z_j)\right) ||^2 + \beta_1 \|\theta_1\|_2^2 \quad (3.1)$$

where p corresponds to a total number of spatial locations after interpolating to the coarse grid, θ is the parameters of the autoencoder and β is a regularizing constant.

2. **Creation of training data for training GRU:**

(a) Once Autoencoder is trained, obtain the reduced representations $\chi^1, \chi^2, \ldots, \chi^n$ from the LSTM encoder.

(b) Create input and output pairs $\left((\xi^{n-k}, \xi^{n-k+1}, ..., \xi^n), (\chi^{n+1}, \chi^{n+2}, ..., \chi^{n+h-1})\right)$ from n=k+1 to $n = N_{Train}$ for training the GRU, where ξ is the vector $\begin{bmatrix} \chi \\ \text{BC} \end{bmatrix}$. The input matrix will be of the shape $N_{train}.N_k.(N_z + N_{bc})$ where N_{train} is the total number of training samples where N_{train}=$N_{Train} - sk$, where s is the number of divisions[5] made while for training samples. N_k is the time window, N_z is the encoded dimension and N_{bc} is the total number of boundary condition. The Output matrix will have the size $N_{train}.(N_z N_h)$ where N_h is the time horizon.

3. **GRU training for temporal prediction:**

(a) Train the GRU by minimizing the mean squared error (MSE) loss:

$$\arg\min_{\theta} \frac{1}{N_{train}.h.z} \sum_{i=k+1}^{N_{train}} \left\| \sum_{j=1}^{h} \sum_{l=1}^{z} [F\left(\chi_i(t_i - 1 : t_i - 1 - k), BC(t_i - 1 : t_i - 1 - k); \theta_2\right) - \chi_i^k(t_j)\right] \right\|^2 + \beta_2 \|\theta_2\|_2^2 \quad (3.2)$$

[4]This process is known as optimization and is performed using ADAM optimizer

[5]Division here refers to the total number of segmentation created while creating training samples. For example, in Chapter 4 the training samples were from the profiles 0-1500 and 3300-8500 making a total of 2 divisions

where F is a function defined by the GRU model,BC corresponds to the boundary condition, θ_2 corresponds to the parameters of the GRU model, β_2 corresponds to a regularizing constant, h corresponds to the horizon and z corresponds to the latent dimension. Figure 3.3 shows the architecture for training the reduced order model for temperature and salinity.

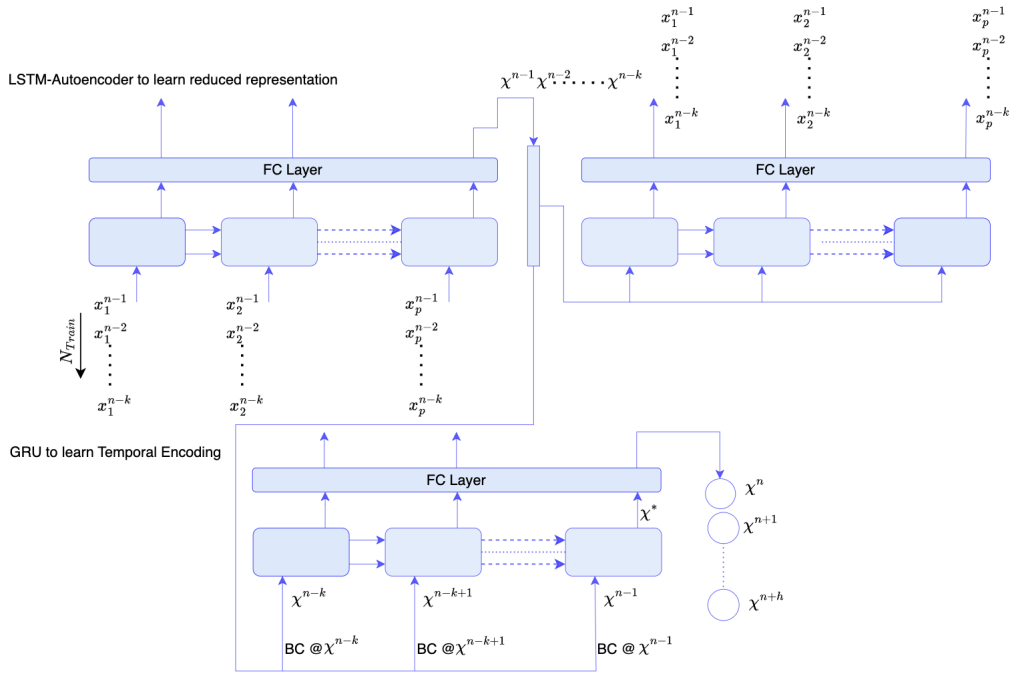

Figure 3.3. Implementation of LSTM-GRU architecture during Training stage

Procedure during testing

Input: Initial profiles of high dimensional temperature and salinity profile from X_1, X_2 $\ldots X_k$

Output: Temporal evolution of profiles. Approximate solutions for $X^{k+1}, X^{k+2}, \ldots, X^{n_{test}}$.

1. Pass the first k high dimensional profiles to the trained LSTM-Autoencoder X_1, X_2 $\ldots X_k$. Get the encodings for the first k profiles χ_1, χ_1 $\ldots \chi_k$

2. Pass $\chi_1, \chi_2 \ldots \chi_k$ to GRU model to predict the next h encodings $\chi_{k+1}, \chi_{k+2} \ldots \chi_{k+h}$.

3. Repeat step 2 until the end of the test period.

4. Decode all the encodings obtained to obtain the approximated profiles by passing them to the decoder.

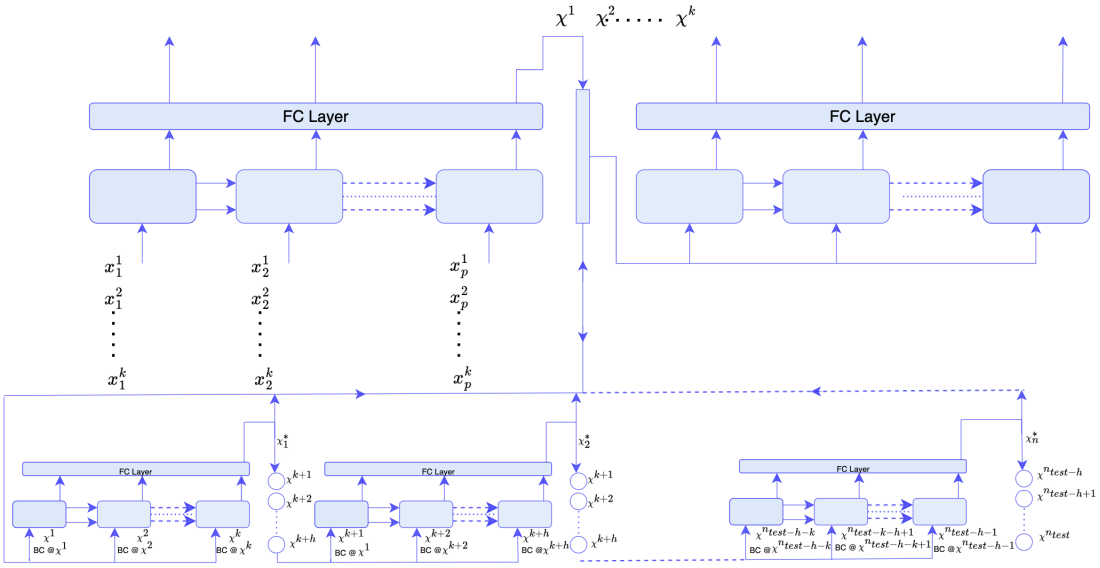

Figure 3.4. Implementation of LSTM-GRU architecture during Testing stage

3.4 Training and predicting temperature and salinity profiles

In this section, a detailed explanation is provided of how temperature and salinity profiles were predicted using the methodology discussed in the previous sections. The methodology is reiterated, considering temperature and salinity as the quantities of interest. It is worth noting that both temperature and salinity can be modeled together, but that is beyond the scope of this work.

3.4.1 Training temperature

Temperature profiles are divided into three sets: Training, Validation, and Test. The Training set of temperature profiles is used to train the LSTM-autoencoder to learn the reduced representation of the profiles. The next stage is to learn the mappings between the previous k encodings of the temperature to the current encodings parameterized by the boundary condition. The Boundary conditions used to parameterize the mapping here are surface wind stresses in both x and y directions, solar heat flux, and non-solar heat flux. This is achieved by concatenating these boundary values with the encodings before being passed to train the GRU model.

Once the model is trained, the first k temperature profiles of the test phase are passed through an autoencoder to learn the reduced representation of the temperature profiles. Next, these k encodings are passed to the GRU model parameterized by its corresponding boundary condition to predict the next h temperature encodings. This process is continued until the last timestep. Since, in this work, we considered k=h, every new prediction is passed directly, parameterized by its corresponding boundary condition to the GRU model.

3.4.2 Training salnity

The procedure carried out here, both during the training and testing stages, is the same as for temperature profiles, except for the use of different boundary conditions during parameterization. The boundary conditions used here are surface wind stresses in the x and y directions, precipitation, and evaporation of freshwater.

Chapter 4

Training both the active and break phase

In this chapter, the results obtained from training the model on data from both the active and break phases are discussed. This section focuses on predicting the rapid evolution between days 3 and 5, which amounts to 1500 timesteps. Temperature and salinity profiles are predicted independently.

4.1 Predicting temperature profiles

The hyperparameter settings for LSTM-Autoencoder and GRU to predict the temperature profiles are shown in table 4.1. These were tuned to give the least validation error.

Table 4.1. List of tuned hyperparameters while training case 1: temperature.

	Epochs	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable	Train time(s)
LSTM-Autoencoder	300	70	8.00E-04	-	0.5	25	3	429.9281502
GRU-Temporal	250	120	1.00E-03	1.00E-06	0.72	25	-	227.7793627

4.1.1 Reconstruction of temperature profiles

The reconstruction capability of temperature profiles between days 3 and 5 is evaluated by calculating the maximum error, defined as $\frac{\max(T_{\text{actual}} - T_{\text{predicted}})}{\text{Normalizing constant}}$. The normalizing constant is determined by the average variation in the temperature at the mixed layer depth over the period of 2 days, calculated using the formula $\frac{1}{N_{\text{profiles}}} \left(\sum_{i=1}^{N_{\text{profiles}}} (T_{i,\max} - T_{i,\min})_{\text{MLD}} \right)$, where the denominator represents the total number of days. This expression evaluates to approximately 2.16°C. The

maximum error for the reconstruction is 0.12 which corresponds to 12% of the total variation at the mixed layer depth. Figure 4.1a shows the error in reconstruction with respect to the true profiles. This error is mainly attributed to the large gradient in temperature profiles with respect to depth. i.e $\frac{dT}{dZ}$ is very large near the mixed layer depth. Therefore any shift in the profile will lead to a large error. As can be seen from Figure 4.2 temperature profiles are almost accurate near the mixed layer depth except that the reconstructed profiles have slightly

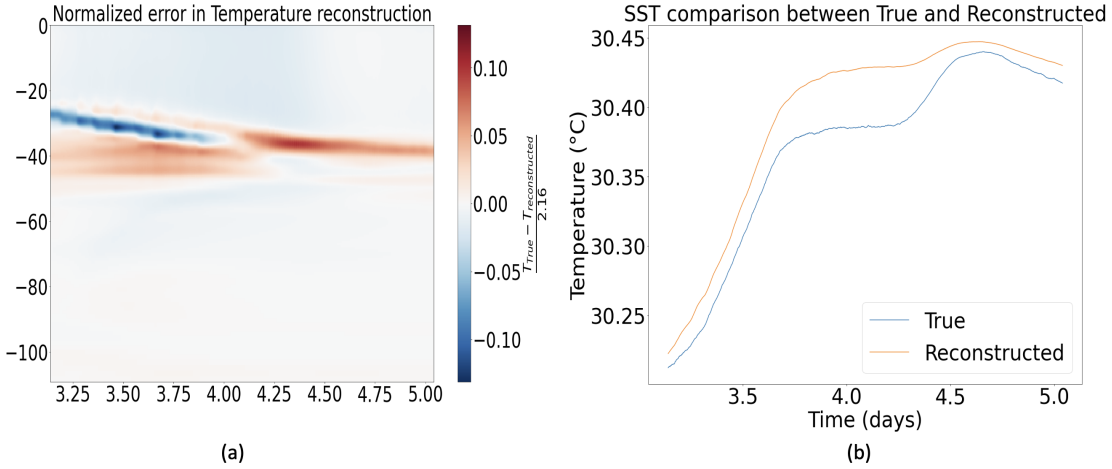

Figure 4.1. (a) Normalized reconstructed Error (b) Reconstructed Sea Surface Temperature.

shifted upwards causing this error. The error varies from -12 to 10 %. Overall the reconstruction capacity of the LSTM-autoencoder is commendable and almost accurately reconstructs the profiles from just 3 latent variables. However as will be shown later, the error in the reconstruction will be reflected in the temporal evolution of the latent variable. Figure 4.1b shows the reconstruction of the Sea Surface Temperature. The maximum error in the SST reconstruction is about 0.04 °C. The important observation is that the SST reconstruction shows the trend of diurnal fluctuation, with a slight shift of 0.05 °C in the temperature value. Figure 4.3 shows the entire profile reconstruction for 2 days. The LSTM-Autoencoder correctly captures the reduction in the thermal inversion layer and the deepening of the mixed layer with increasing time.

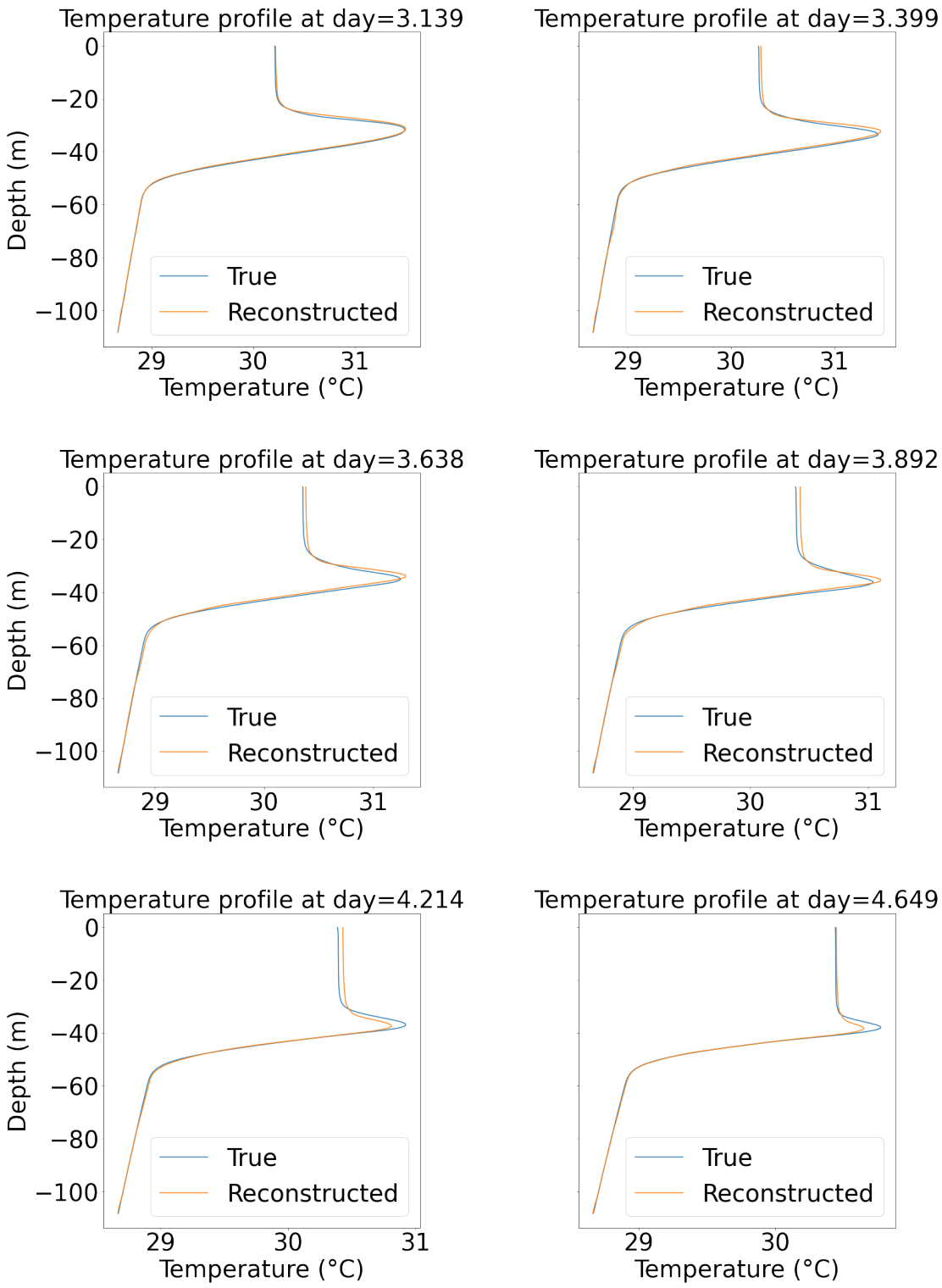

Figure 4.2. Temperature reconstruction at various time instances.

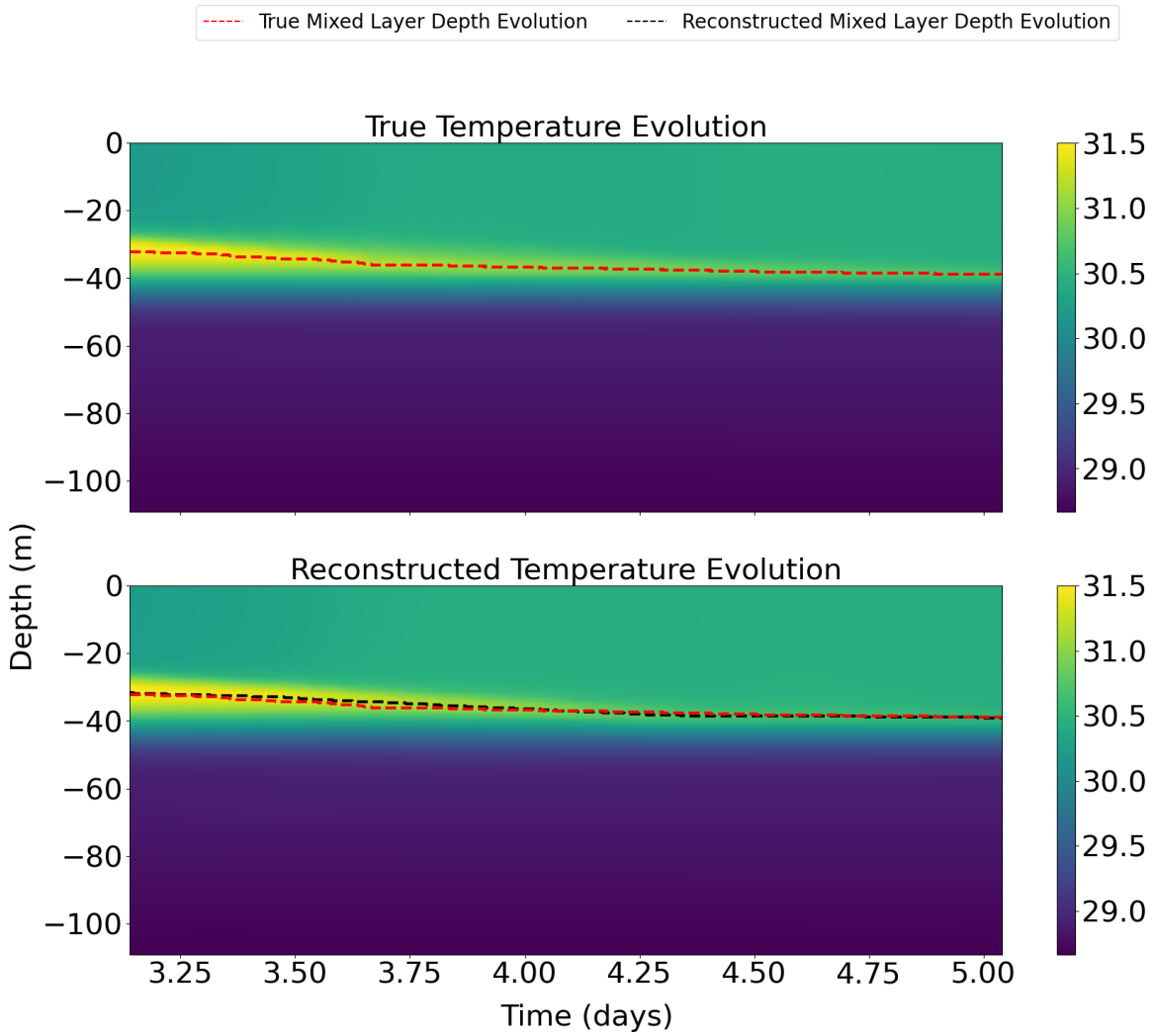

Figure 4.3. Comparison between true and reconstructed Temperature over the period of 2 days.

4.1.2 Temporal evolution of the latent variables and prediction of temperature profiles

The temporal evolution of latent variables is shown in figure 4.4a. The prediction is overall reasonable for the three latent variables over 2 days. However, the evolution of the 2nd latent variable seems to be off by a large amount by 0.03. To verify the accuracy of the prediction of the evolution, figure 4.4b shows the prediction period with respect to the entire period. The change in the 2nd latent variable over the entire period is 0.2. The maximum error is thus 15% of the total variation in the value of latent variable 1 over the period of 14 days. Therefore, the error

is justifiable and can be verified from the profiles obtained after decoding the latent variables in Figure 4.6. The maximum normalized error defined by $\frac{\max(T_{\text{actual}} - T_{\text{predicted}})}{\text{Normalizing constant}}$ is 0.15 as seen from the Figure 4.5a, which is 3% more than the error due to reconstruction of the profiles. Figure 4.5b compares the normalized error between predicted profiles and true reconstructed profiles.

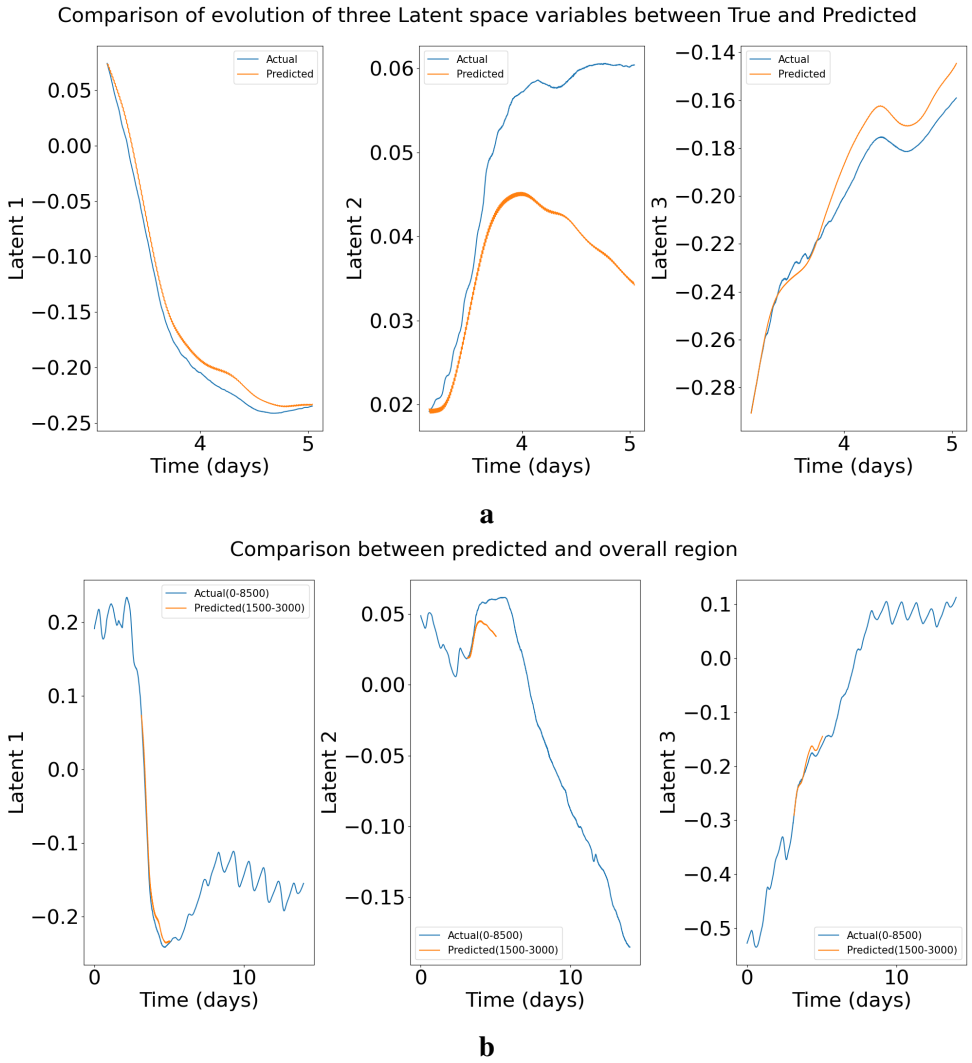

Figure 4.4. (a) Temporal evolution of latent variables over 2 days (b) Temporal evolution of latent variables over 14 days.

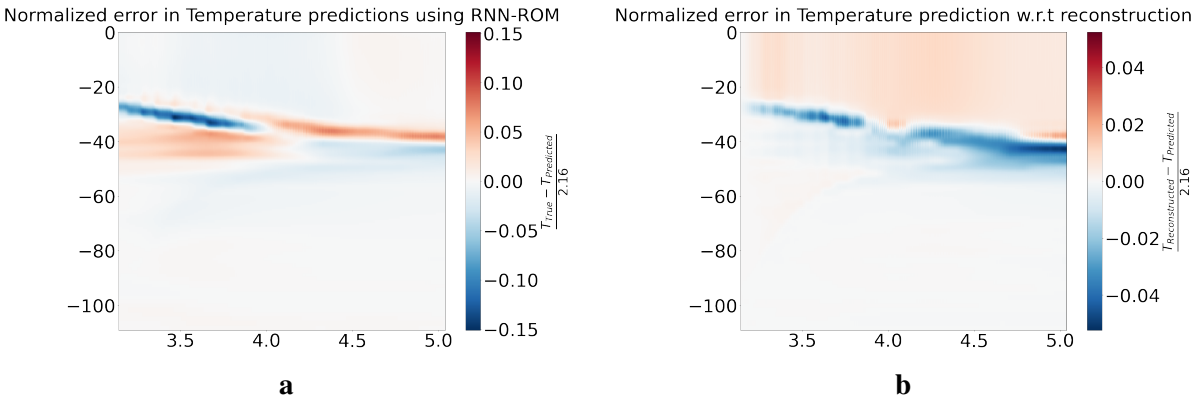

Figure 4.5. (a) Normalized error in temperature prediction over the 2-day period (b) Normalized error in temperature with respect to reconstructed Temperature profiles over a 2-day period.

Figure 4.6 shows the profiles of temperatures at various time instances. The differences in the profiles between true and predicted are minimal. Figure 4.7 shows the 2-day prediction and evolution of the temperature profiles. The RNN-ROM model accurately predicts the reduction in thermal inversion and deepening of the mixed layer depth. The maximum error in the MLD is 1.80m. Figure 4.8 shows the evolution of Sea surface temperature (SST). The model also predicts the diurnal variation of the SST.

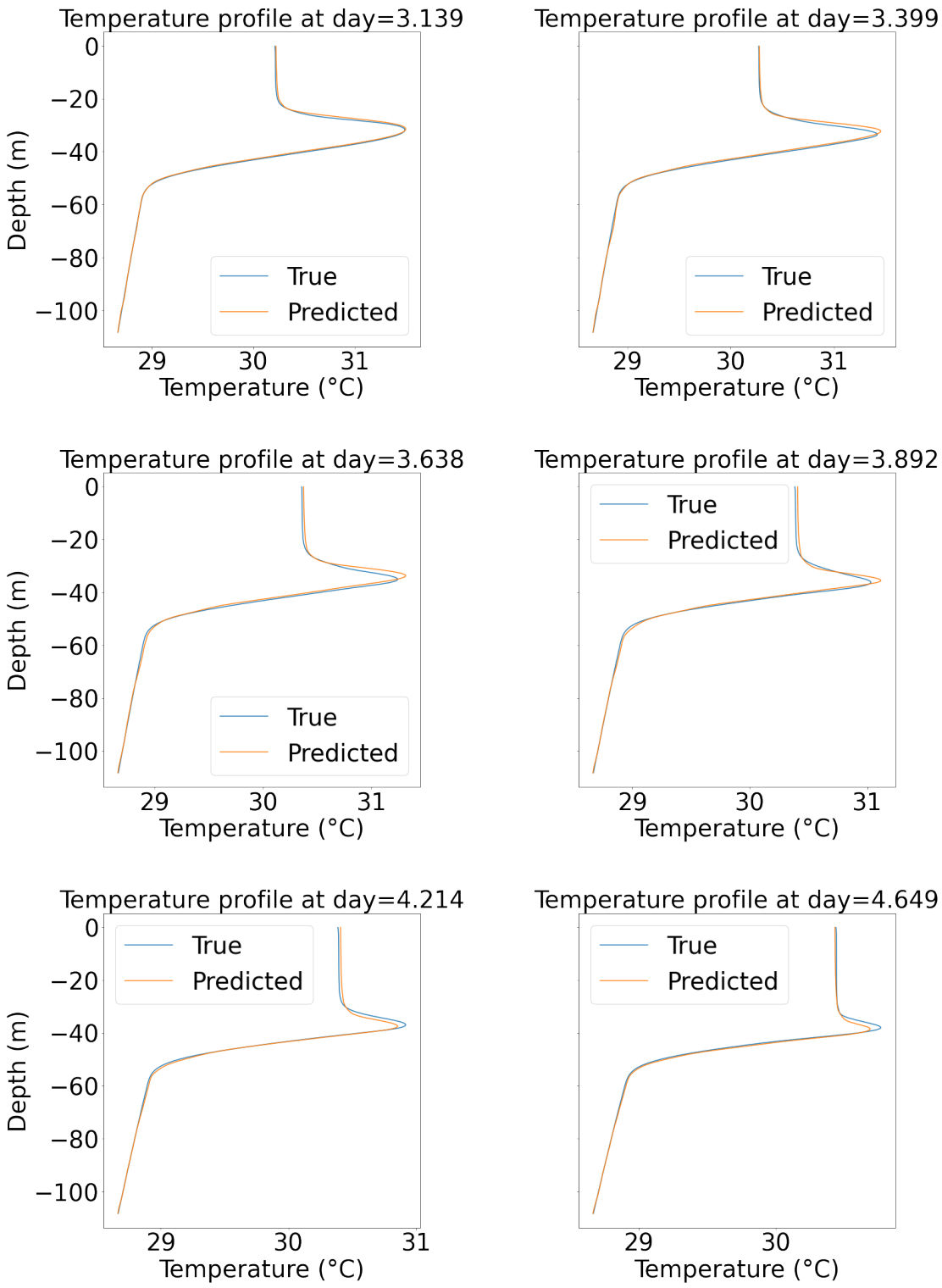

Figure 4.6. Temperature predictions at various time instances.

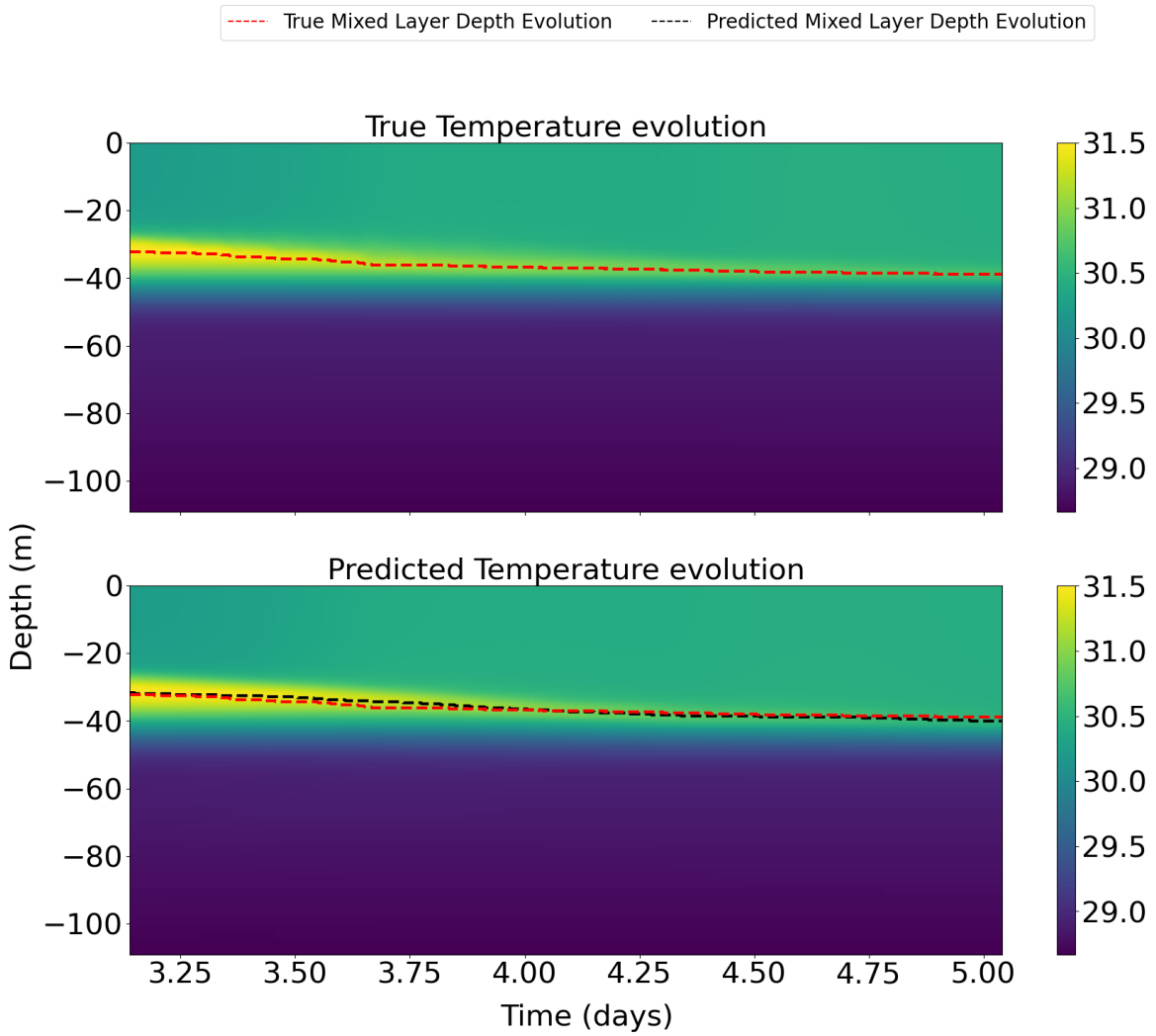

Figure 4.7. Comparison between true and predicted Temperature over the period of 2 days.

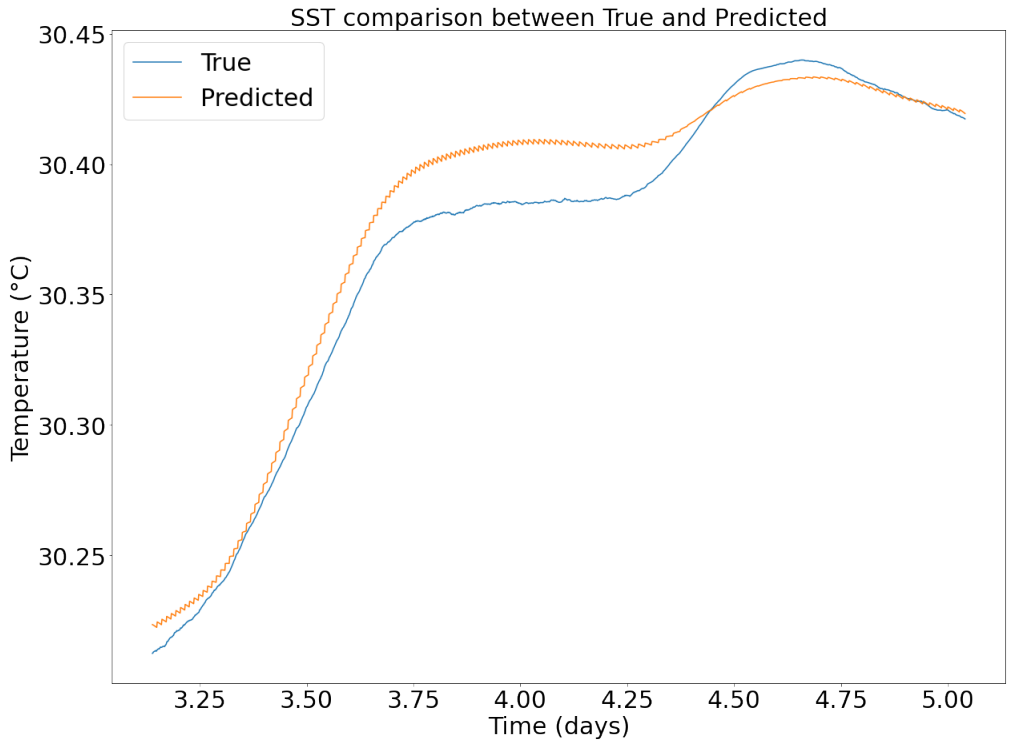

Figure 4.8. Comaprison between true and predicted SST.

4.2 Predicting salinity profiles

The hyperparameters tuned for minimum validation error for constructing the RNN-ROM for salinity predictions are shown in table 4.2. These are obtained by minimizing the validation error for the salinity predictions.

Table 4.2. List of tuned hyperparameters while training case 1: salinity.

	Epochs	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable	Train time(s)
LSTM-Autoencoder	300	70	8.00E-04	1.00E-08	0.52	25	3	447.9291914
GRU-Temporal	250	76	9.00E-04	1.00E-06	0.67	25	-	233.8510501

4.2.1 Reconstruction of salinity profiles

The reconstruction capability of salinity profiles between days 3 and 5 is evaluated by calculating the maximum error, defined as $\frac{max(S_{actual} - S_{predicted})}{\text{Normalizing constant}}$. It is calculated pointwise, both in space and time. The average change in the salinity value is 1 psu at mixed layer depth over 2-day

period. Therefore, the normalizing constant is taken to be 1 psu. The maximum error in the reconstruction is approximately 0.12 which is 12% of the total variation of salinity at the mixed

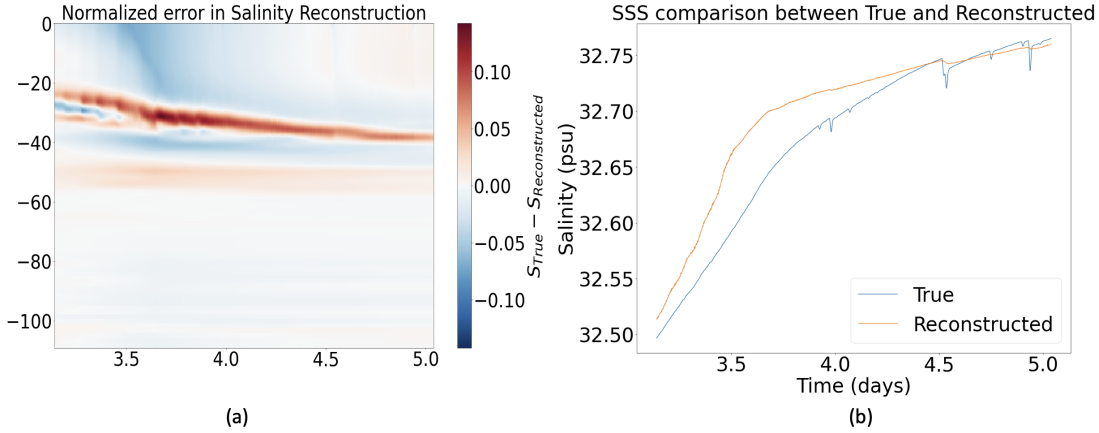

Figure 4.9. (a) Normalized reconstructed error (b) Reconstructed Sea Surface Salinity.

layer depth. Figure 4.9a shows the error throughout the prediction with respect to the true profiles. Similar to the temperature profiles, the salinity profiles also exhibit a steep gradient, $\frac{dS}{dZ}$, near the mixed layer depth. Therefore, a slight deviation will cause a sudden jump in the error. Figure 4.10 shows the comparison of salinity profiles at various time instances. The differences in the profiles are not visible, but there is a slight vertical shift, causing the error to be as high as 12%. Overall throughout the prediction the error varies between -5 to 12%. Figure 4.9b shows the reconstruction in the Sea surface salinity (SSS). The LSTM Autoencoder initially over-reconstructs the SSS, producing values that are 0.05 psu higher than the actual salinity, and fails to reconstruct the influx of fresh water on certain days; however, the predictions become accurate in the later period. These errors will be reflected during the temporal prediction of salinity profiles. Figure 4.11 shows the reconstruction of salinity profiles for the entire 2-day period. The model accurately captures the deepening of mixed layer depth over time.

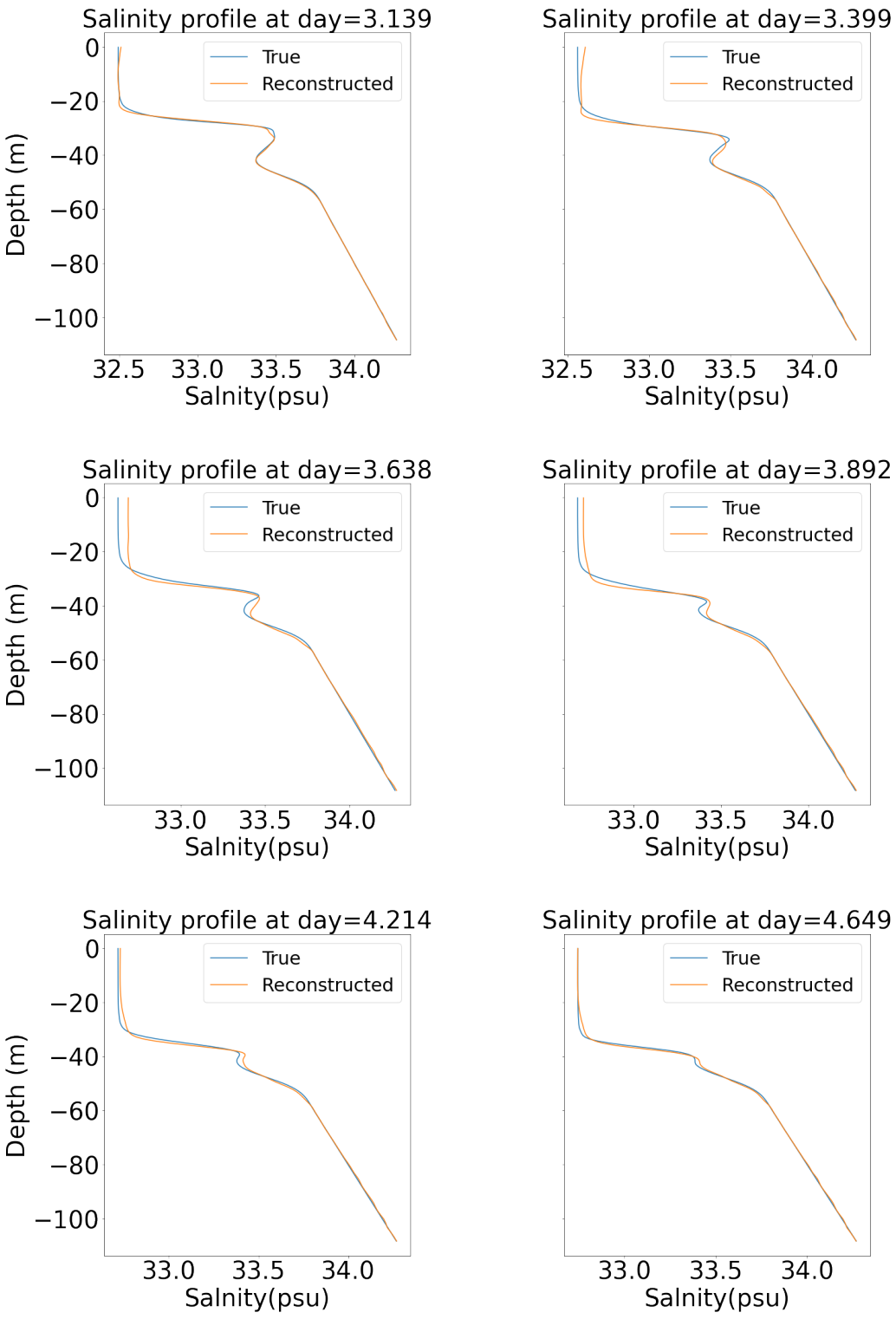

Figure 4.10. Salinity reconstruction at various time instances.

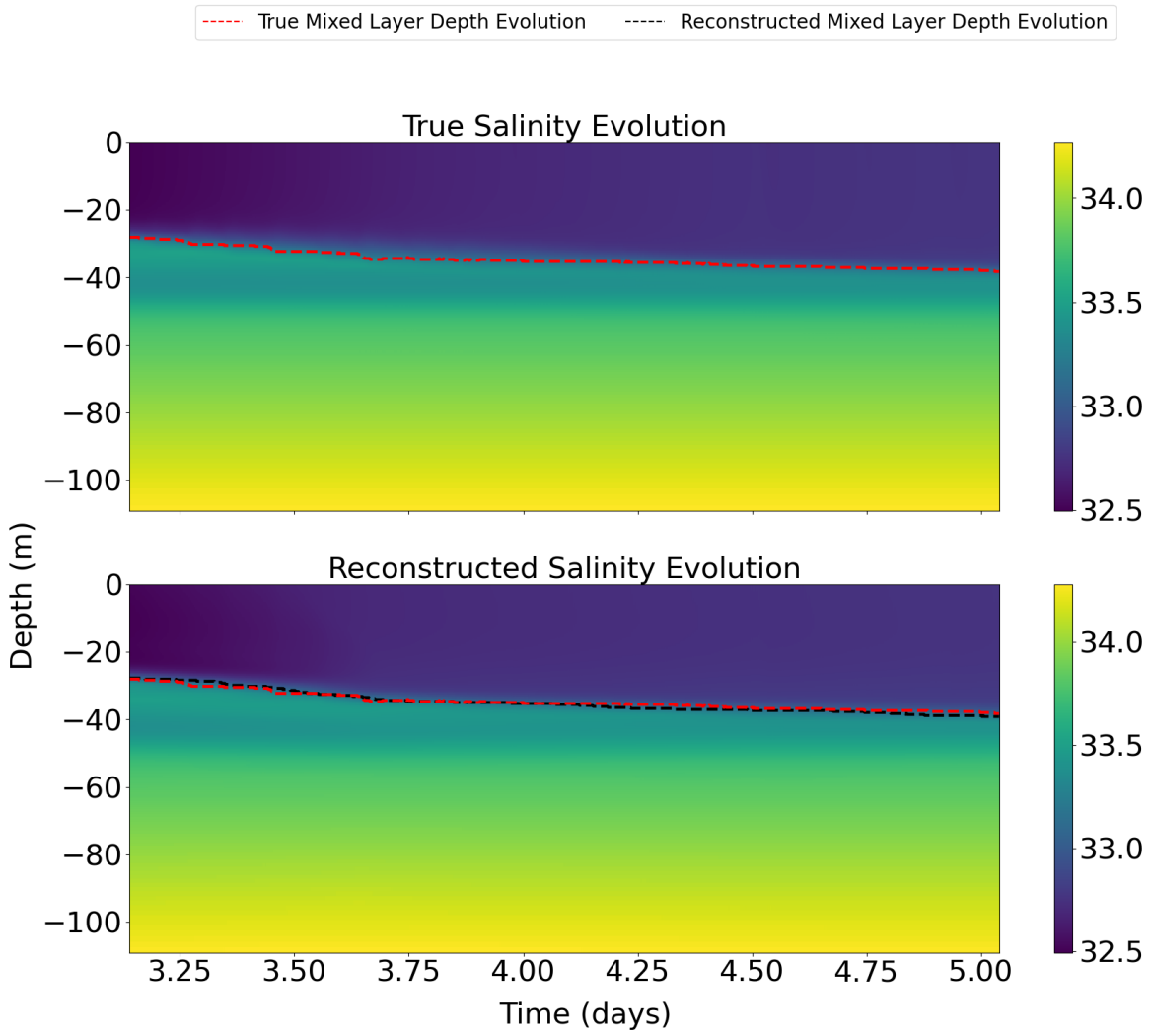

Figure 4.11. Comparison between true and reconstructed salinity over the period of 2 days

4.2.2 Temporal evolution of the latent variables and the prediction of salinity profiles

Figure 7.17 shows the temporal evolution of the three latent variables. The model accurately predicts the evolution of Latent Variable 1. However, there are discrepancies in the predictions for latent variables 2 and 3, although the model correctly captures the overall trend of their evolution. To assess the magnitude of these errors, Figure 4.12b compares the predicted period with the overall period. The maximum discrepancy in the prediction of latent variable 2 is 0.02 where the maximum variation is 0.3 for latent variable 2, which results in an error of

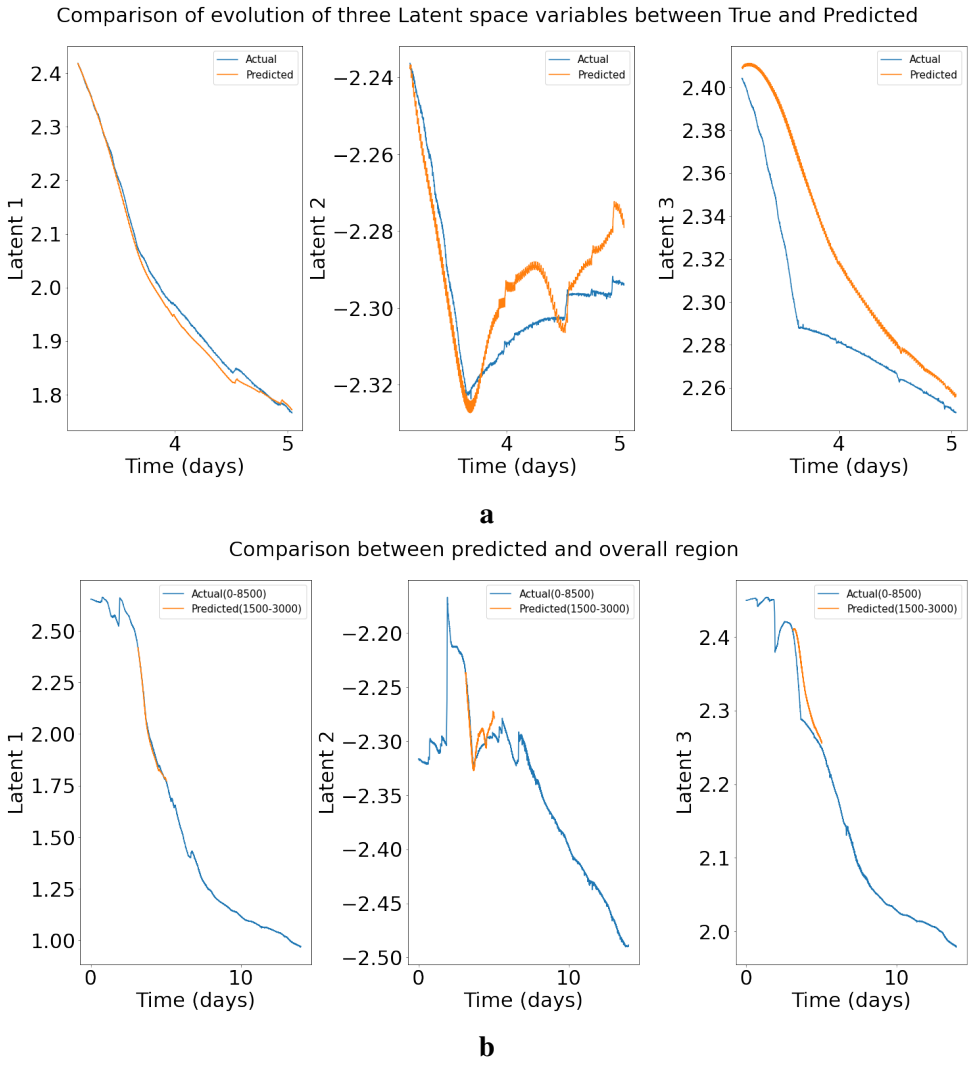

Figure 4.12. (a) Temporal evolution of latent variables over 2 days (b) Temporal evolution of latent variables over 7 days.

approximately 6 %. A similar analysis applies to latent variable 3. To verify this explanation, Figure 4.13(a) shows the error in the overall predictions for the 2-day period compared to the true.

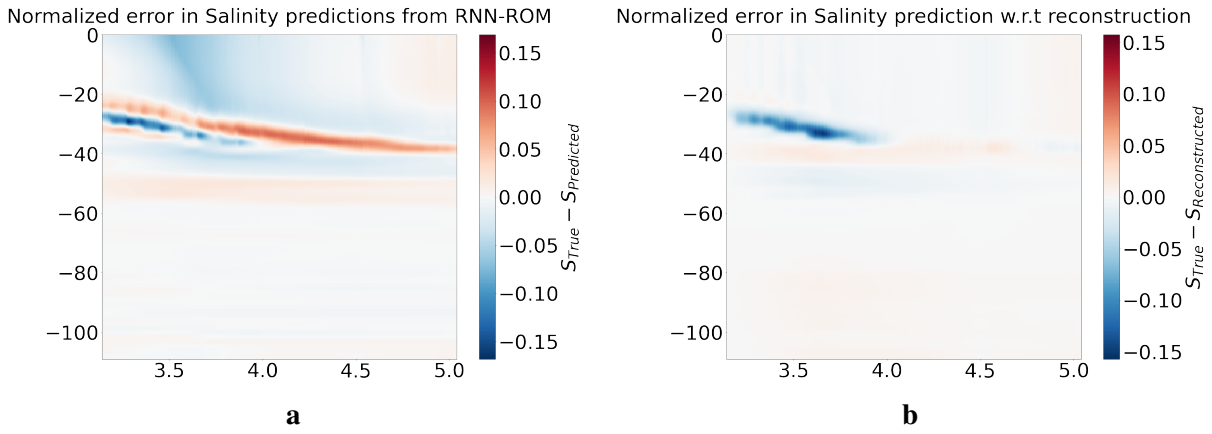

Figure 4.13. (a) Error in salinity prediction (b) Error in salinity prediction with respect to reconstructed profiles.

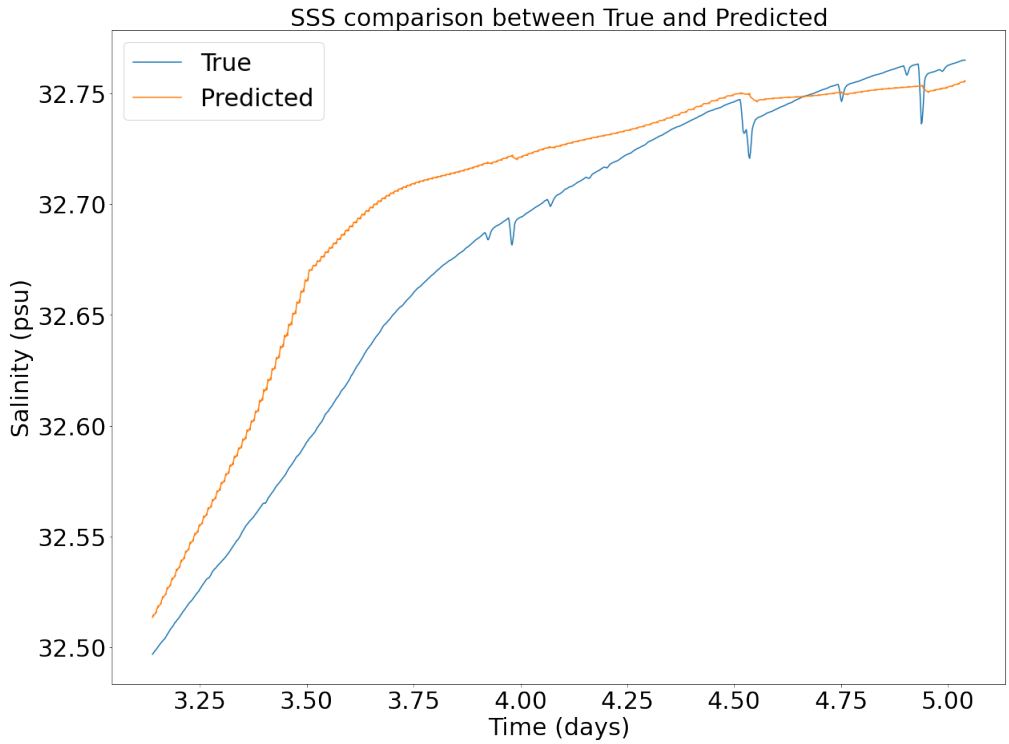

Figure 4.16. Comaprison between true and predicted SSS.

values. The maximum error in the prediction is about 0.15, which corresponds to 15% of the total change in the salinity at the mixed layer depth. This error is 3% higher than the maximum error obtained during reconstruction (Figure 4.9(a)). Figure 4.13(b) shows the prediction error relative to the actual reconstructed profile. Figure 4.14 compares the predicted salinity profiles

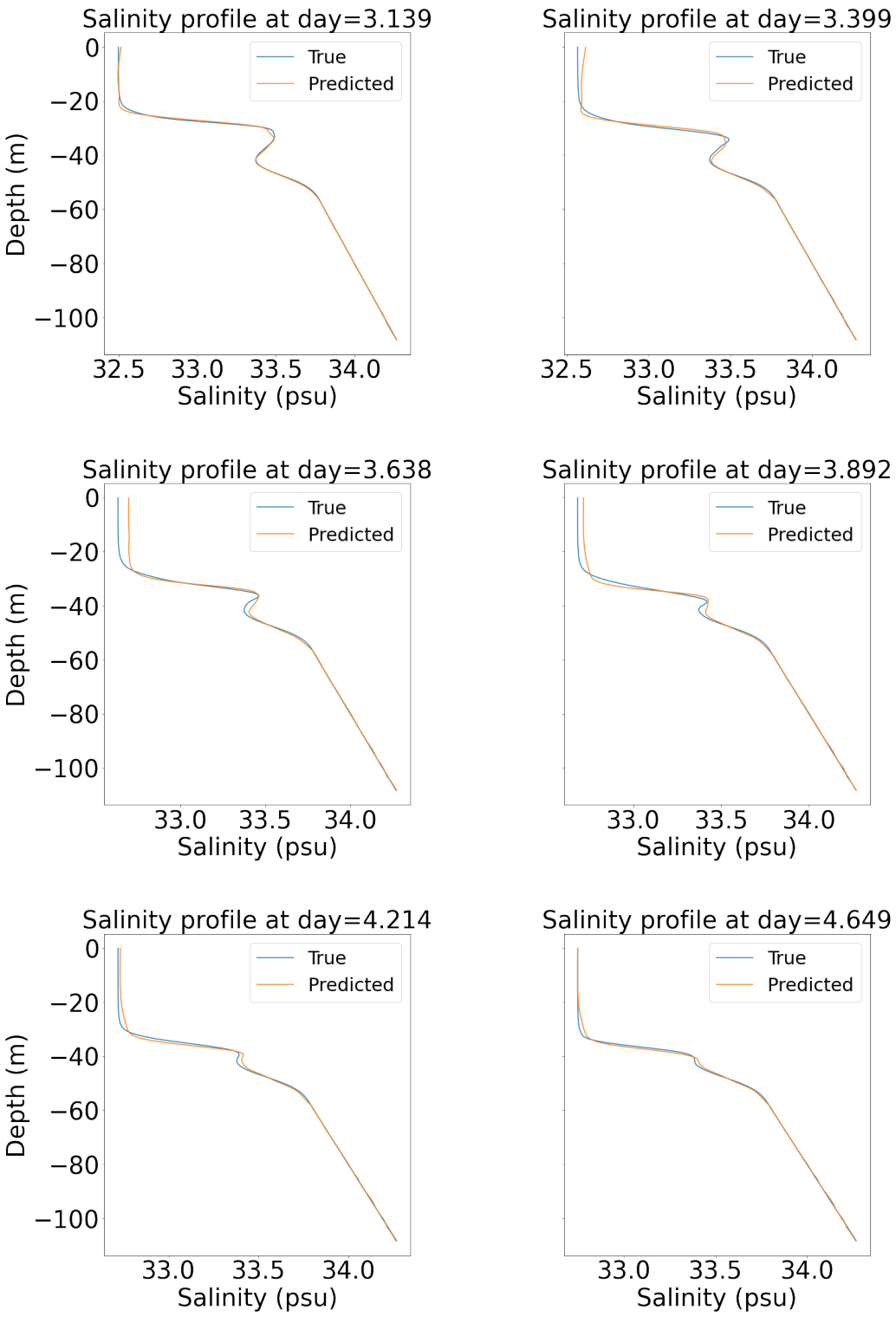

Figure 4.14. Salinity predictions at various time instances.

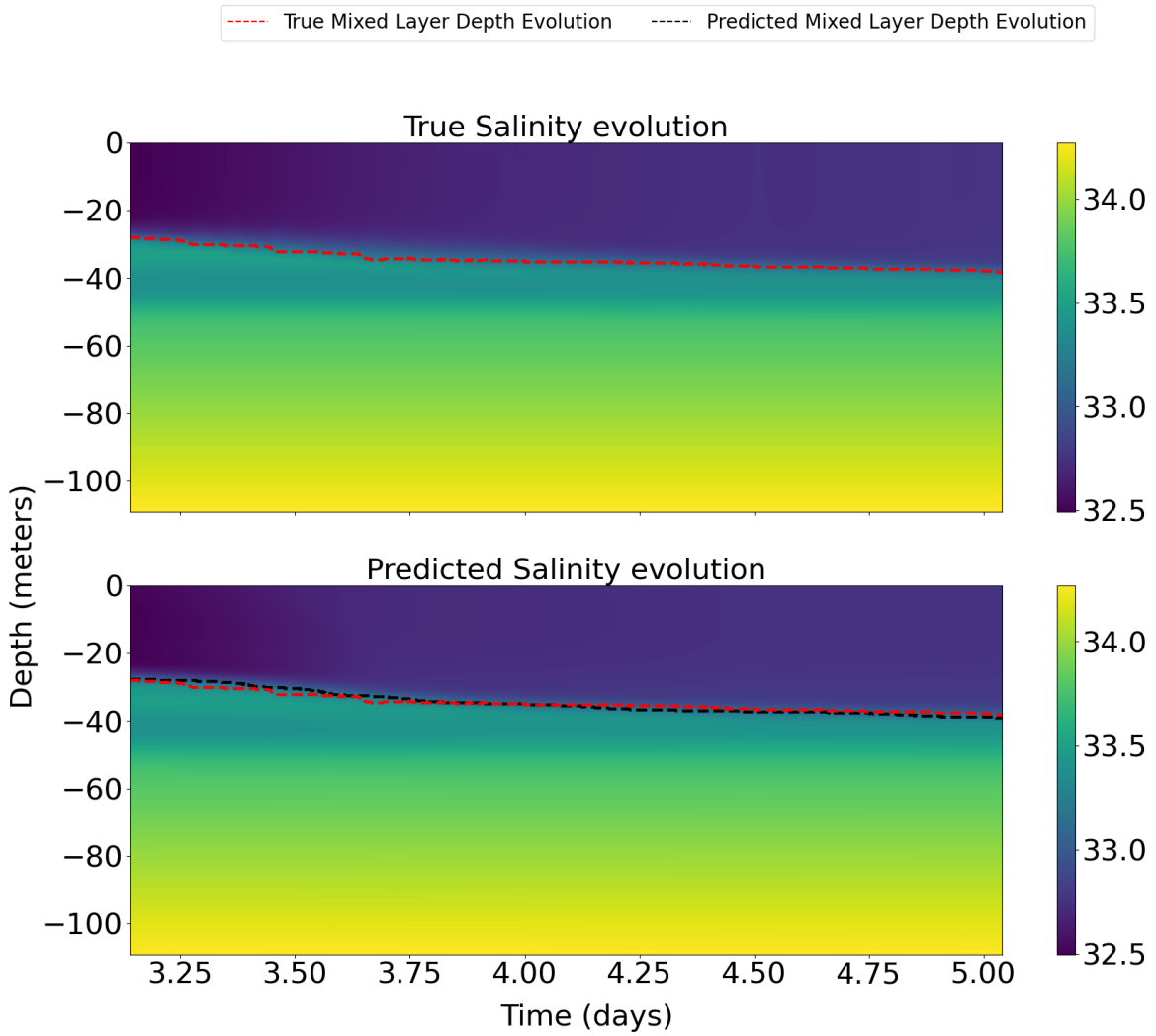

Figure 4.15. Comparison between true and predicted salinity for 2 days.

to the truth at various time instances. The differences in the profiles are minimal, except in the mixed layer for a few of them. The RNN-ROM model predicts a slightly higher value of salinity at certain instances in time (see Figure 4.14 3rd profile). Figure 4.16 depicts the comparison between predicted and actual Sea surface Salinity (SSS). The model over-predicts the SSS in the initial period, about 0.05 psu, attributing to the incorrect reconstruction by the LSTM-Decoder. Figure 4.15 shows the evolution of salinity profiles over the entire 2-day period, comparing the truth and the predictions. The RNN-ROM accurately captures the deepening of the Mixed layer depth (MLD) over time and the maximum error in MLD prediction is 2.10m.

Chapter 5

Training only active phase

This chapter presents the results obtained from training the model exclusively on the active phase data. The dataset comprises a total of 4500 profiles, of which 1500 profiles are reserved for prediction, 300 profiles for validation, and the remaining 2700 profiles are used for training the RNN-ROM. The prediction focuses on a period of 2 days. Similar to the approach in Chapter 4, this chapter aims to predict the rapid evolution between days 3 and 5. The results are primarily compared with those in Chapter 4, focusing mainly on speed and accuracy. The comparison is summarized in a table, and a conclusion is drawn.

5.1 Predicting temperature profiles

The hyperparameters for the RNN-ROM model were determined by minimizing the validation error of the temperature profiles during both reconstruction and temporal evolution. Table 5.1 lists the hyperparameter values for the current RNN-ROM model configuration used in predicting the temperature profiles.

Table 5.1. List of tuned hyperparameters while training case 2: temperature.

	Epochs	Hidden units	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable	Train time(s)
LSTM-Autoencoder	300	70	1	8.00E-05	-	0.55	24	3	194.4171
GRU-Temporal	250	55	2	5.00E-04	1.00E-06	0.7	28	-	101.4912

5.1.1 Reconstruction of temperature profiles

The reconstruction capability of temperature profiles between days 3 and 5 is evaluated by evaluating the maximum error over 2 days. The error, as previously stated, is evaluated by the expression $\frac{\max(T_{\text{actual}} - T_{\text{predicted}})}{\text{Normalizing constant}}$. The normalizing constant is calculated by evaluating the total change in the temperature at the mixed layer depth for 2 days. The constant is evaluated by $\frac{1}{N_{\text{profiles}}} \left(\sum_{i=1}^{N_{\text{profiles}}} (T_{i,\max} - T_{i,\min})_{\text{MLD}} \right)$, where the denominator represents the total number of predicted days. This expression evaluates to 2.16. Figure 5.1a shows the normalized error in the reconstruction of temperature profiles. The maximum reconstruction error is 12% of the average variation of the temperature in mixed layer depth over two days. This maximum error is the same as when obtained while training the model on data from both the active and break phases. The reason is again attributed to the large gradient in the temperature with respect to the depth, $\frac{dT}{dz}$, near the mixed layer depth.

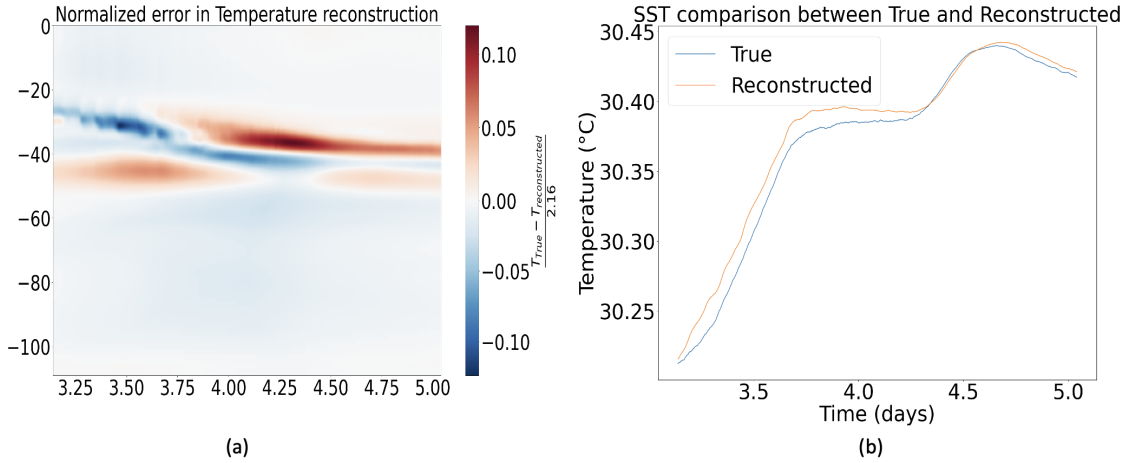

Figure 5.1. (a) Normalized reconstructed error (b) Reconstructed Sea Surface Temperature.

Figure 5.2 shows the comparison between temperature reconstruction and the truth at various time instances. The total variation in the error ranges from -12-(10)% with most of the error concentrated at the mixed layer depth. The reconstruction is carried from just 3 latent variables. Figure 5.1b provides a comparison of the reconstructed Sea surface temperature (SST) with the true SST. The reconstructed SST is almost identical to the truth with better accuracy than the

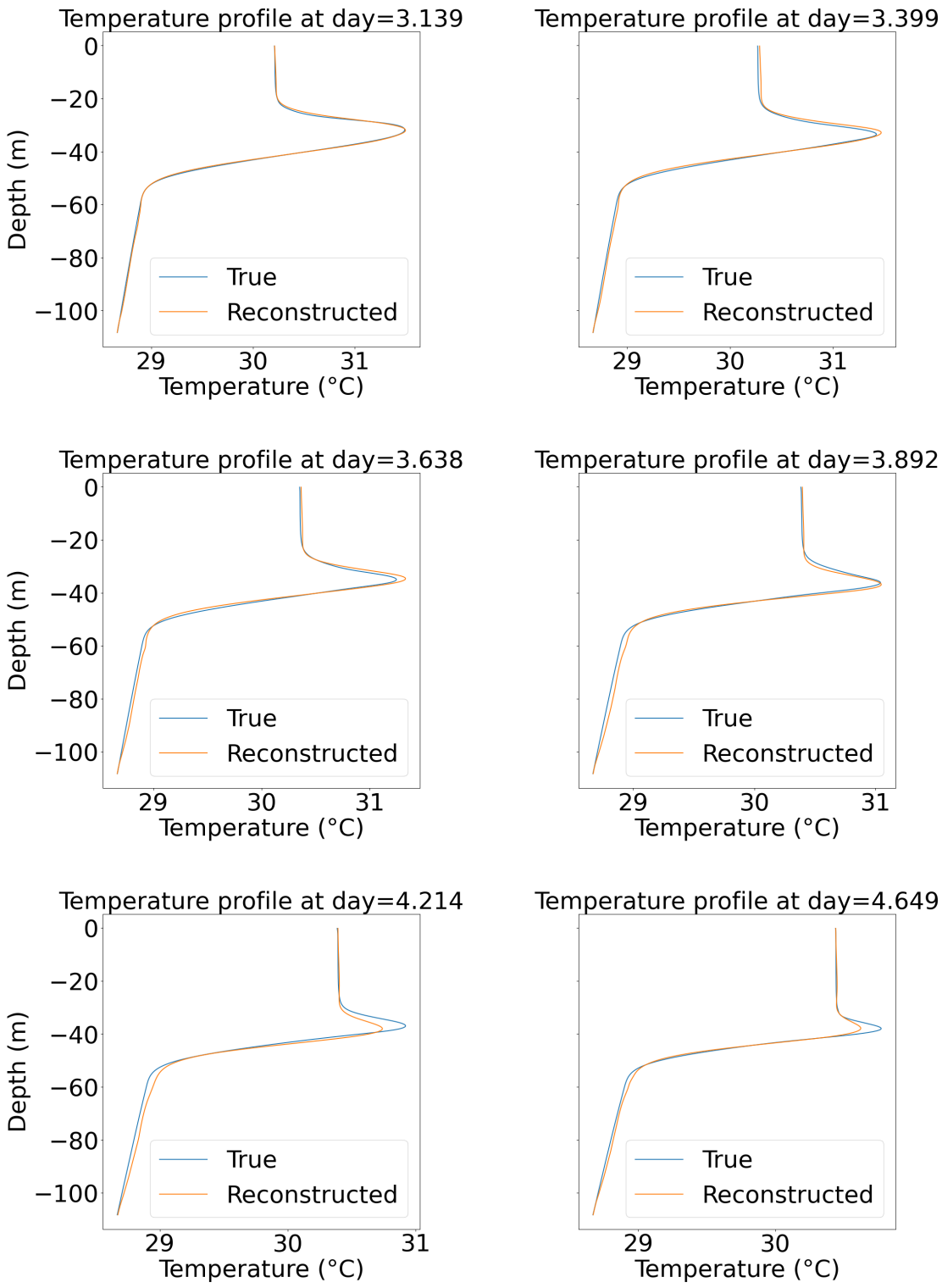

Figure 5.2. Temperature reconstruction at various time instances.

one obtained from training both active and break phases. Figure 5.3 shows the entire profile reconstruction for 2 days. Again, the LSTM-autoencoder correctly captures the reduction in the thermal inversion layer and the deepening of the mixed layer. Overall, the performance of the LSTM autoencoder has improved from the previous case.

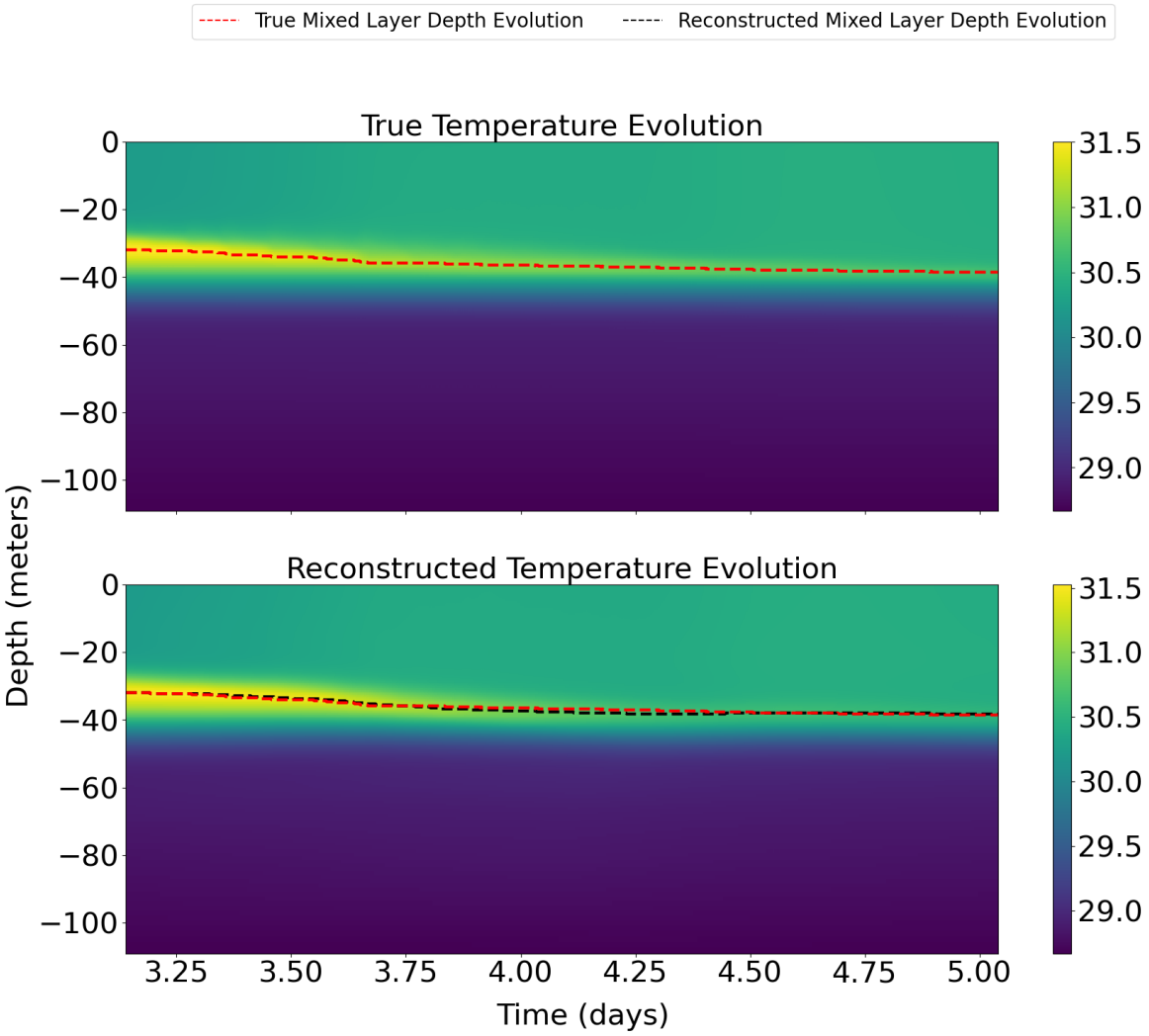

Figure 5.3. Comparison between true and reconstructed temperature over the period of 2 days.

5.1.2 Temporal evolution of the latent variables and prediction of temperature profiles

A comparison of the temporal evolution is shown in the figure 5.4a. The prediction is close for the 2nd and 3rd latent variables. However, the prediction for the evolution of latent

variable 1 has been offset by almost 0.03. To check the severity of the error, figure 5.4b shows the temporal prediction of latent variables compared with the overall period. The maximum change in the overall value of latent variable 1 is 0.11. Therefore, the relative error in the prediction, calculated as, $\frac{\text{maximum error in the prediction}}{\text{Maximum-Minimum value of the variation}}$, is 27%. Figure 5.5a shows the normalized error between the true and predicted values. Surprisingly, the maximum error, which is about 8% of the total change in the mixed layer depth over the 2 days, is 33% lower than the reconstruction error.

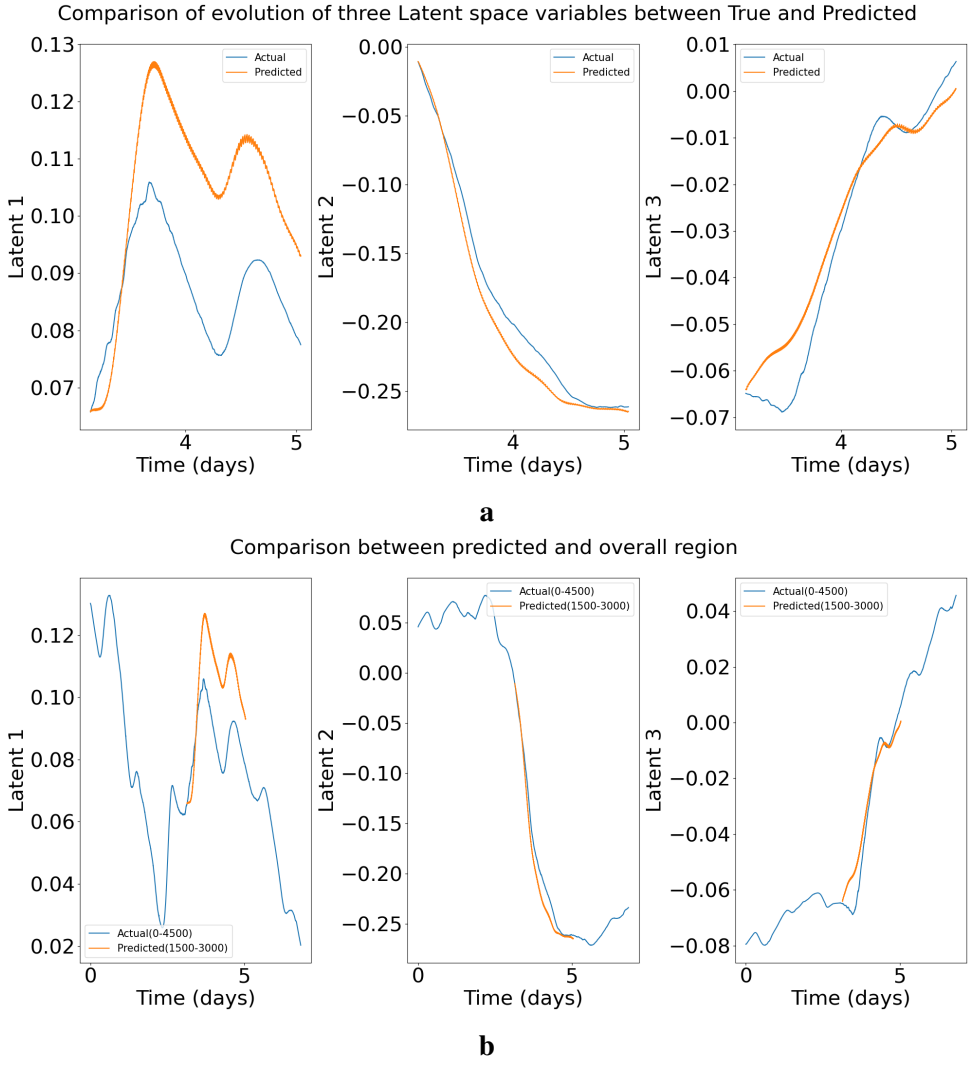

Figure 5.4. (a) Temporal evolution of latent variables over 2 days (b) Temporal evolution of latent variables over 7 days.

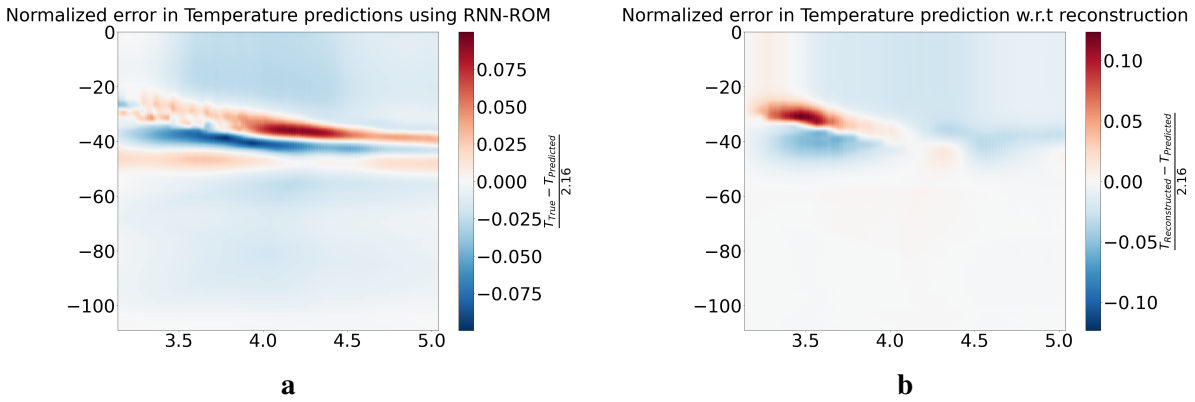

Figure 5.5. (a) Error in temperature prediction (b) Error in temperature prediction with respect to reconstructed profiles.

Figure 5.5b shows the error in prediction compared to the reconstruction. The total error $(T_{True} - T_{Predicted})$ can be defined as $(T_{Reconstructed} - T_{Predicted}) + (T_{True} - T_{Reconstructed})$. Therefore, if the expression in the first bracket evaluates to a negative value at a point, the total error decreases, which explains the reduction in the total error at certain points. Figure 5.8 shows the comparison of temperature profiles at various time instances. Figure 5.7 compares the Sea surface temperature (SST) between the true and predicted values over a 2-day period. The model captures the diurnal variation of the SST, though there is a maximum offset of about 0.05 °C. This error can be attributed entirely to the offset observed in latent variable 1. Figure 5.6 shows the temperature evolution over 2 days. The RNN-ROM accurately predicts the reduction in the thermal inversion layer and the deepening of the mixed layer depth. The maximum error in the prediction of MLD is 0.90m, which is less than the error observed when both phases were used for training(Figure 4.7).

One important observation is the increased complexity in the true temporal evolution of latent variable 1 compared to the latent variables obtained during training on both the active and break phases. This increased complexity may arise from the model learning more intricate functions due to the reduced number of training samples. Drawing a definitive conclusion may therefore be challenging.

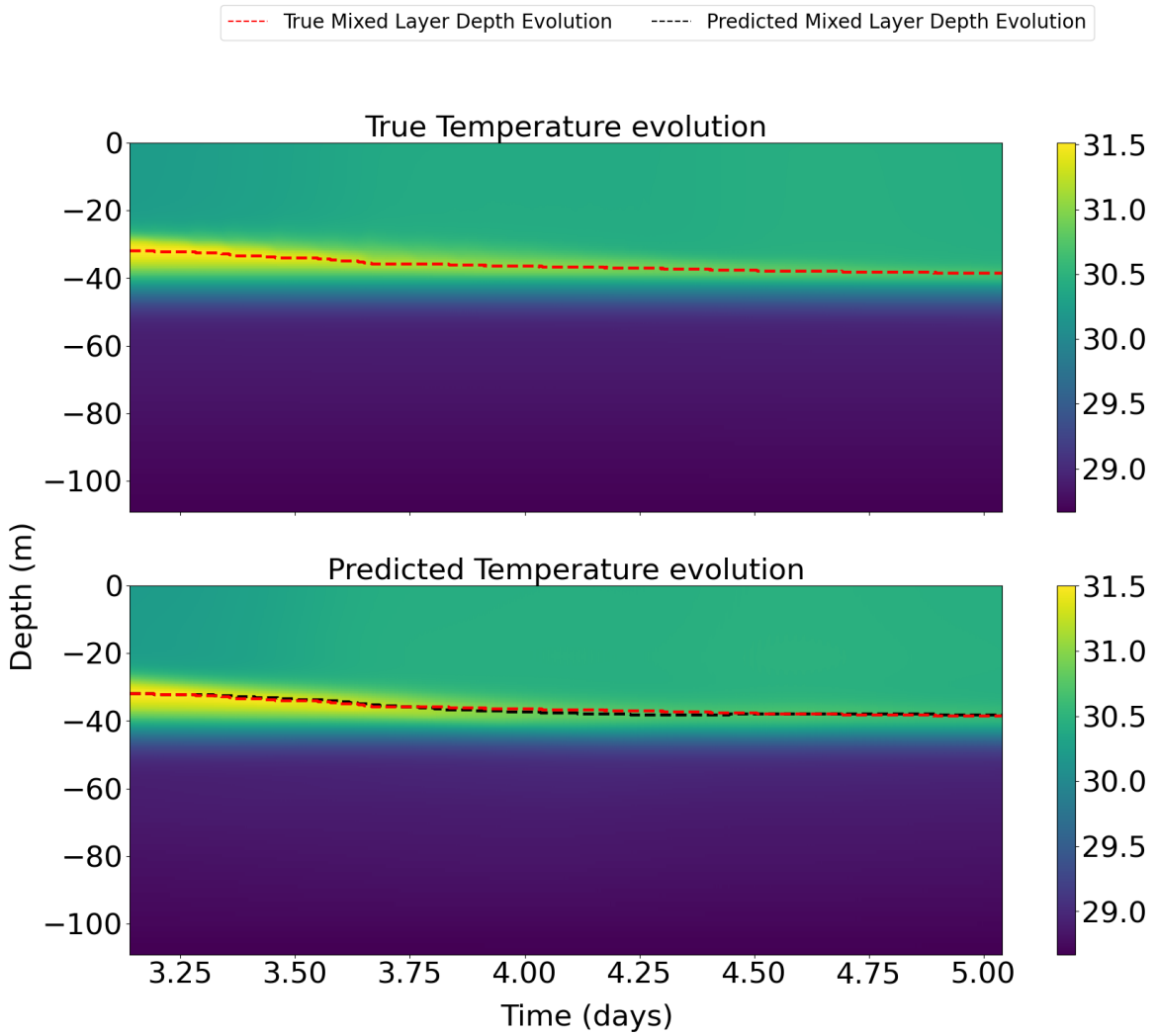

Figure 5.6. Comparison between true and predicted temperature over the period of 2 days.

5.2 Predicting salinity profiles

The hyperparameter tuning is performed based on the minimum error obtained during the validation of the salinity profiles. Table 5.2 lists the hyperparameters used and their values.

Table 5.2. List of tuned hyperparameters while training case 2: salinity.

	Epochs	Hidden units	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable	Train time(s)
LSTM-Autoencoder	300	70	1	6.00E-04	1.00E-07	0.7	24	3	191.7232
GRU-Temporal	250	77	1	5.00E-04	-	0.8	20	-	107.3470

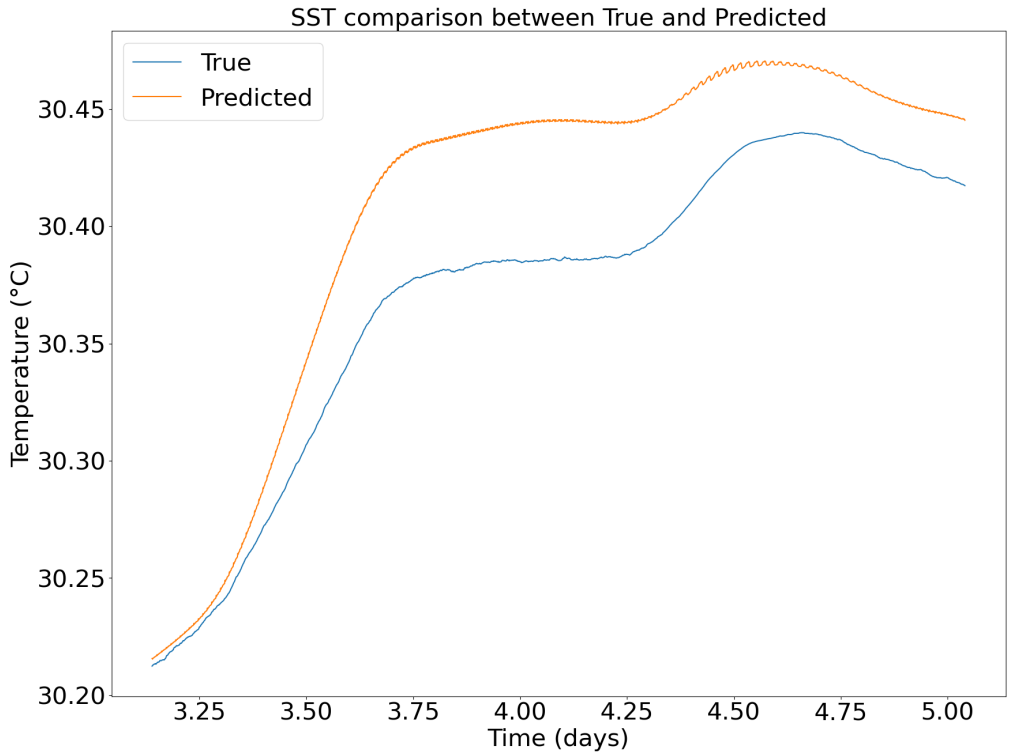

Figure 5.7. Comaprison between true and predicted SST.

5.2.1 Reconstruction of salinity profiles

As defined in the previous chapter, the reconstruction capability of LSTM-autoencoder for salinity profiles over the 2 days is evaluated by calculating the maximum error, defined by $\frac{max(S_{acatual} - S_{predicted})}{\text{Normizing consant}}$, where the calculation is done pointwise both in space and time. The normalizing constant, as defined in the previous chapter is taken to be 1 psu. The maximum error in the reconstructed profiles is approximately 0.16, which is 16% of the total variation of the salinity at the mixed layer depth. This maximum error is 4% higher than the maximum error observed in the previous case (figure 4.9a). Figure 5.9a shows the error plot over the 2-day period. The high error is attributed to the large gradient $\frac{dS}{dZ}$, which amplifies even minor deviation. Figure 5.10 compares reconstructed and the true salinity profiles at various time instances. The difference in the profiles isn't visible, although there is a discrepancy in the mixed layer of the reconstructed profile on day 1(In the figure 5.10 days 3-4). Although not visible, the slight shift

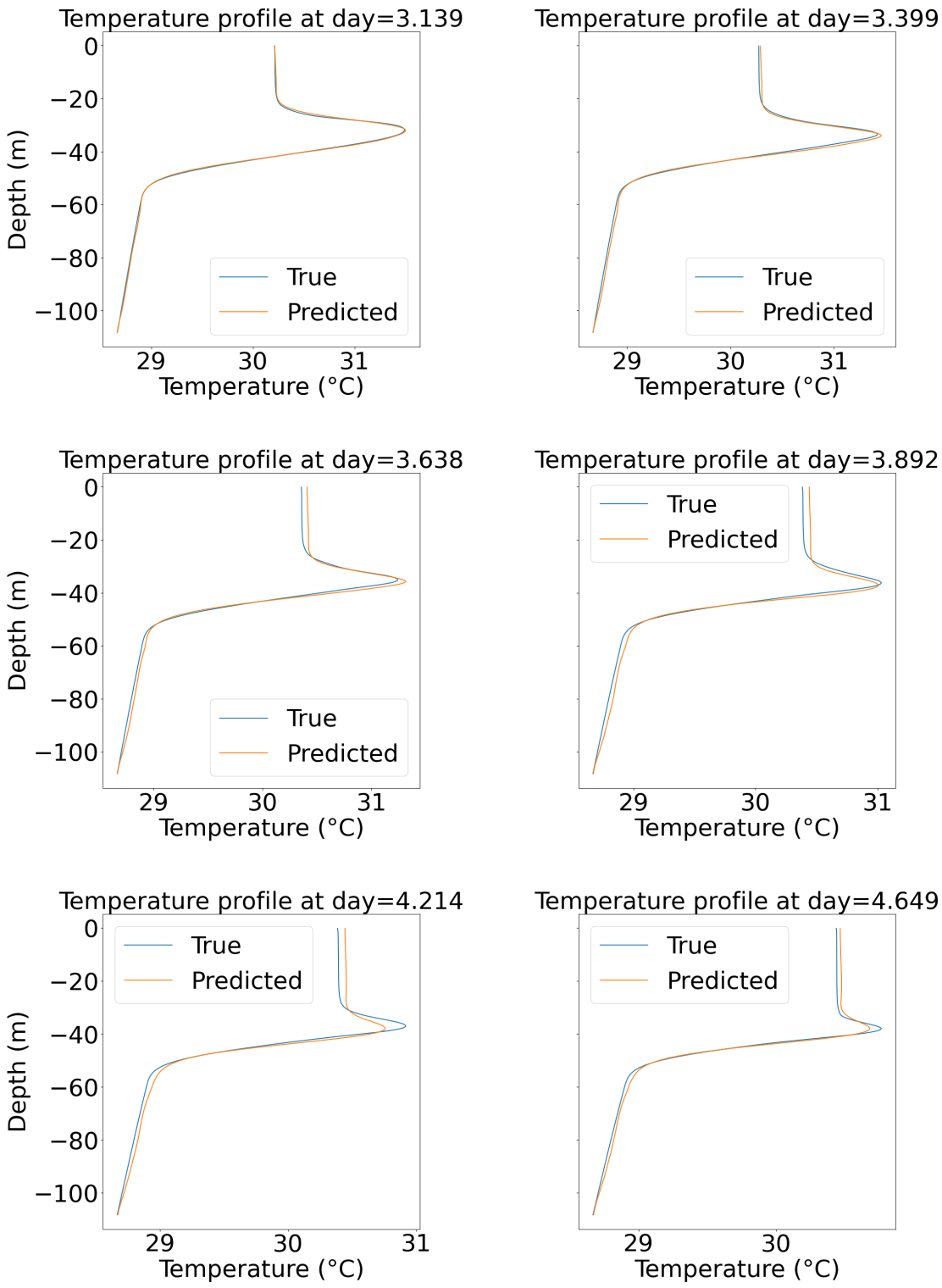

Figure 5.8. Temperature predictions at various time instances.

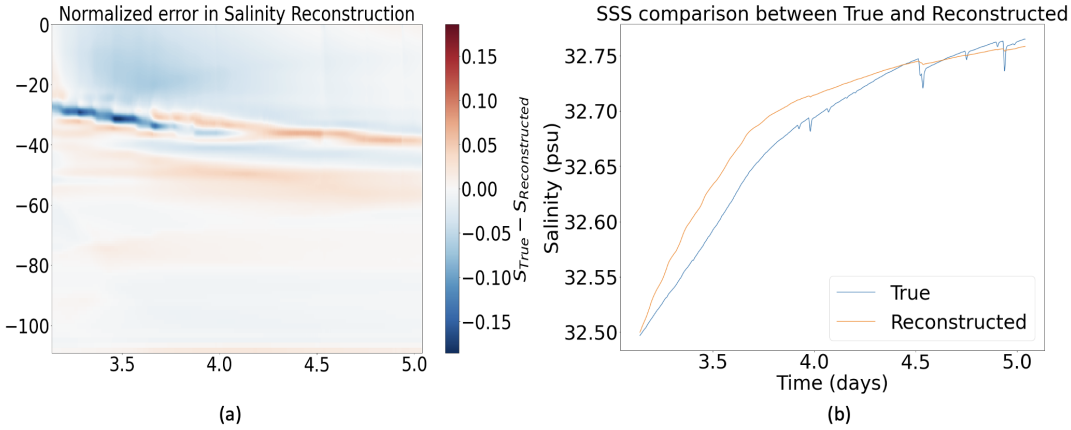

Figure 5.9. (a) Normalized reconstructed error (b) Reconstructed Sea Surface Salinity.

in the vertical direction causes the error to be as high as 15%. The overall error varies between -15 to 10%. Figure 5.9b shows the reconstruction of the Sea surface salinity(SSS). The LSTM over-reconstructs SSS for approximately a day. For the 2nd day, the reconstruction is accurate. As discussed in the previous chapter, the LSTM-autoencoder fails to reconstruct the sudden influx of fresh water. This failure will be reflected during the temporal predictions of the salinity profiles. Figure 5.11 shows reconstruction for the entire 2-day period. The model correctly predicts the deepening of the mixed layer over time. Therefore, reconstructions obtained with only half of the training data are comparable to those made using the full dataset.

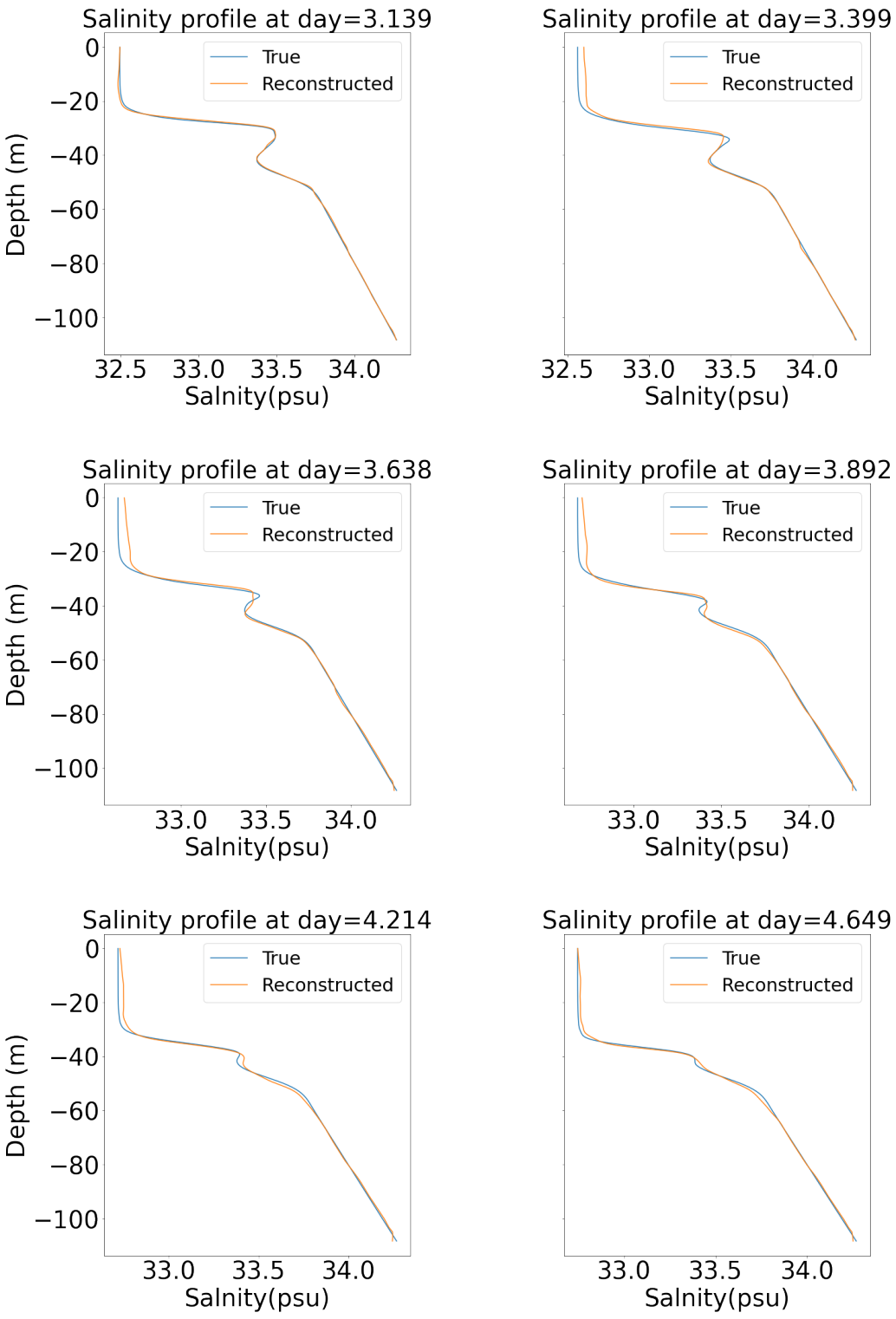

Figure 5.10. Salinity reconstruction at various time instances.

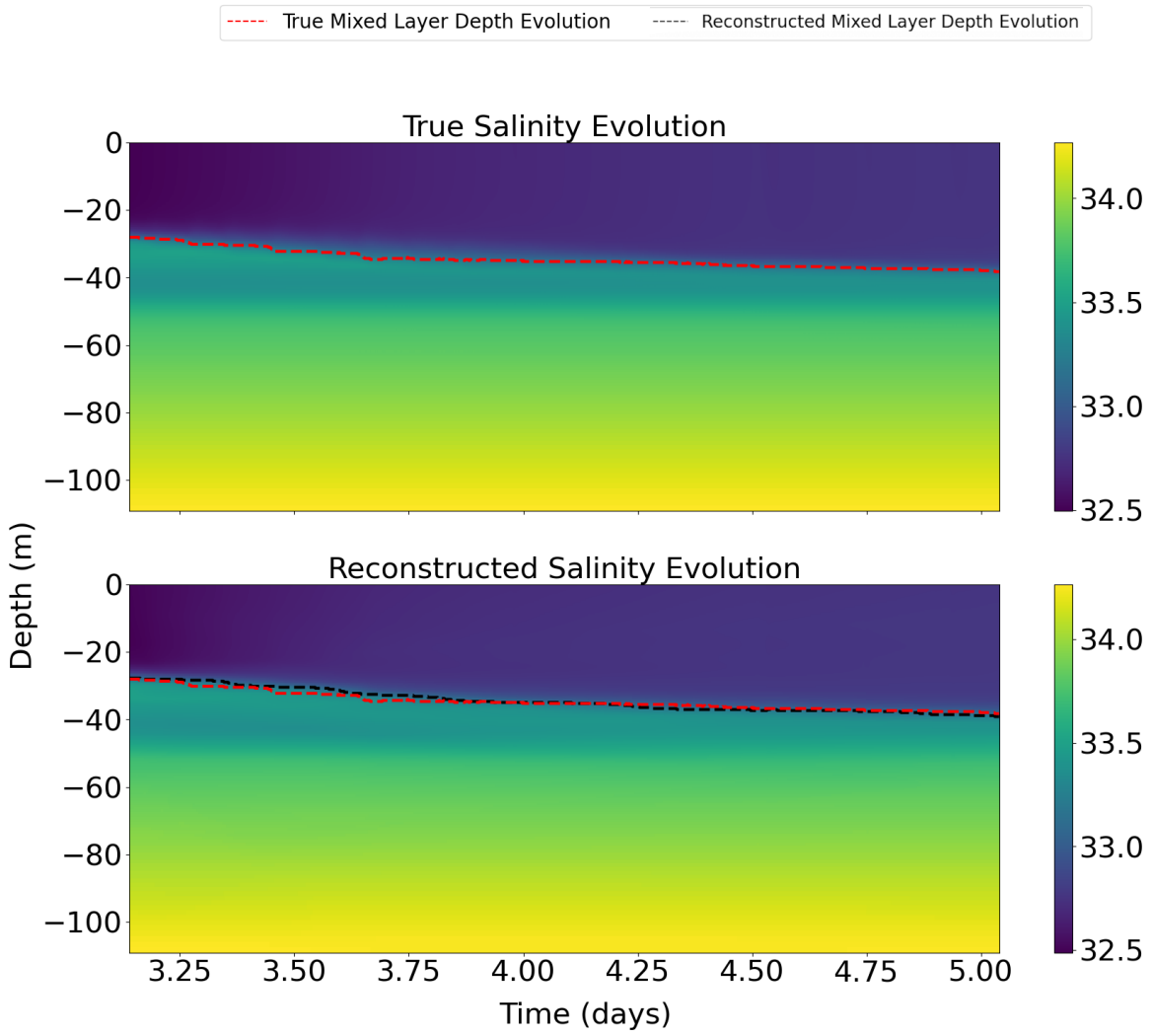

Figure 5.11. Comparison between true and reconstructed Salinity for 2 days.

5.2.2 Temporal evolution of the latent variables and the prediction of salinity profiles

Figure 5.12a illustrates the temporal evolution of the three latent variables. All three latent variables exhibit some discrepancies compared to the true evolution. To assess the extent of these errors, Figure 5.12b compares the predicted period to the overall period. Compared to the maximum variation of the latent variables, the discrepancy is small and should not cause a large deviation in the final salinity profiles. To further verify this, Figure 5.13a depicts the error in the prediction compared to the truth for the 2 days. The maximum error is 0.2, which

corresponds to 20% of the total variation in salinity at the mixed layer depth. The error is 4% higher than the maximum error obtained during the reconstruction $(E_{True} - E_{Recosntructed})_{\max}$.

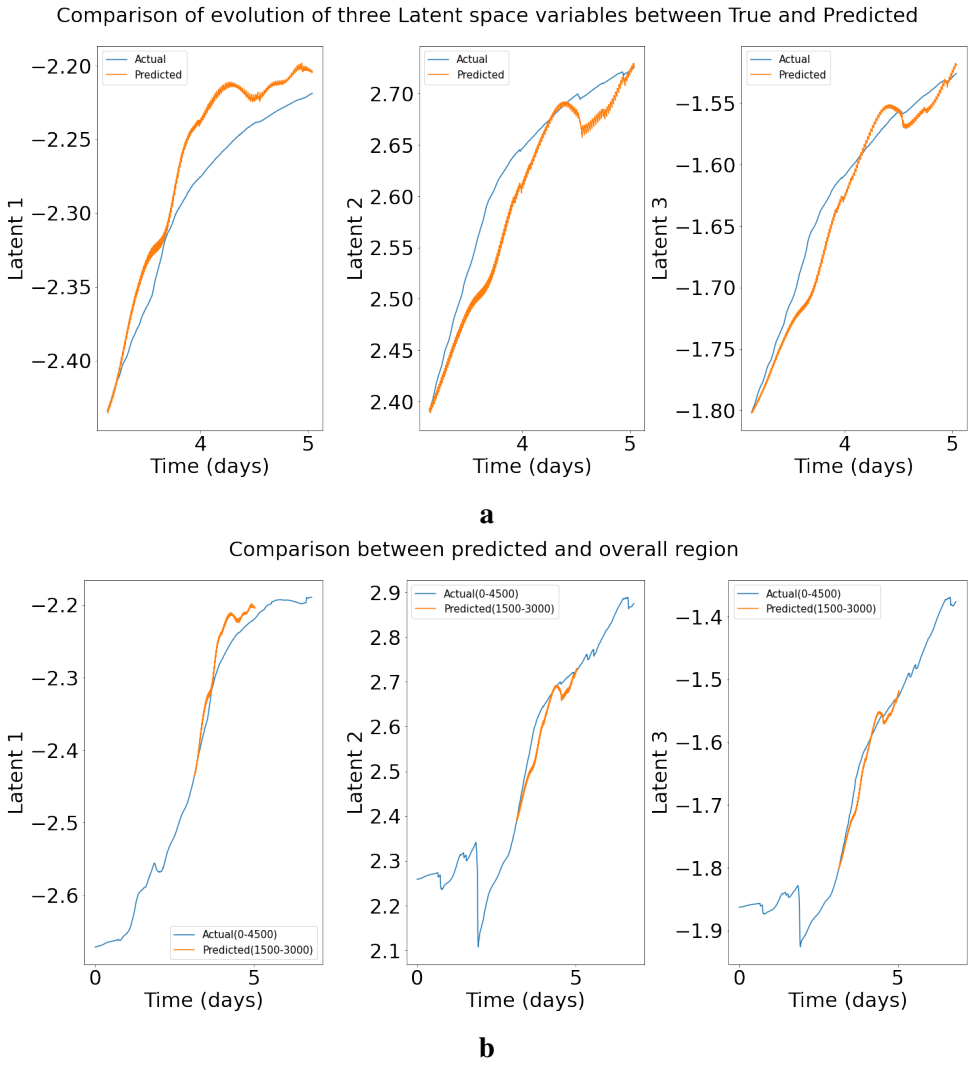

Figure 5.12. (a) Temporal evolution of latent variables over 2 days (b) Temporal evolution of latent variables over 7 days.

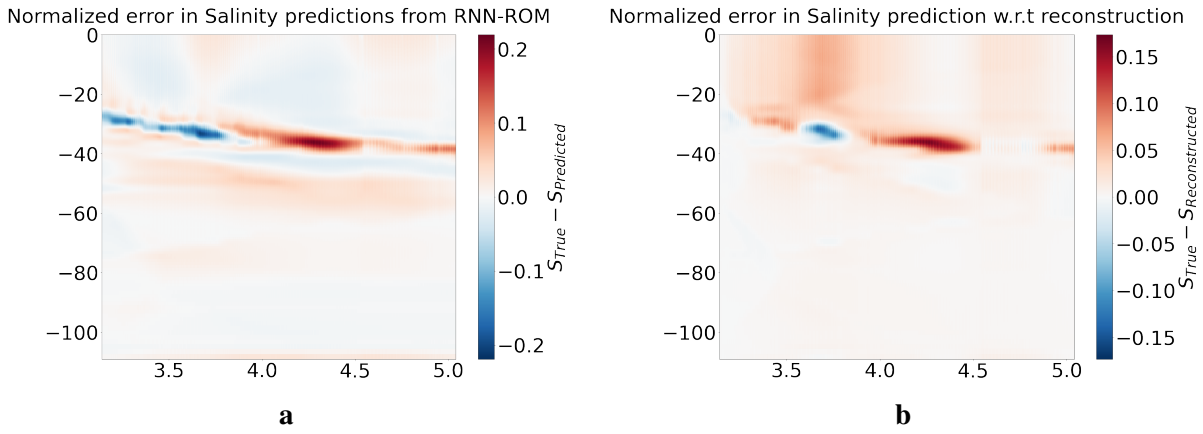

Figure 5.13. (a) Error in salinity prediction (b) Error in salinity prediction with respect to reconstructed profiles.

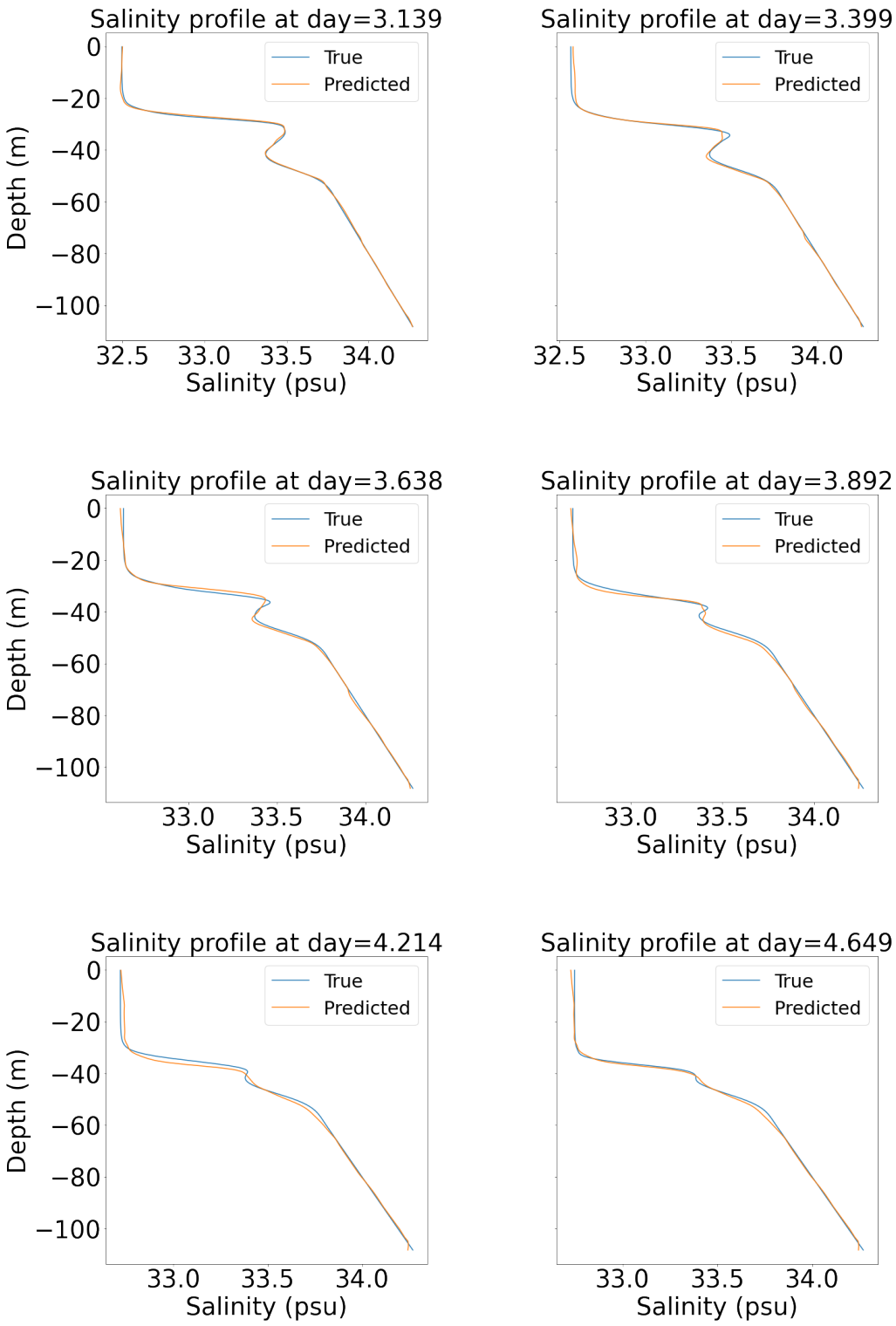

Figure 5.14. Salinity predictions at various time instances.

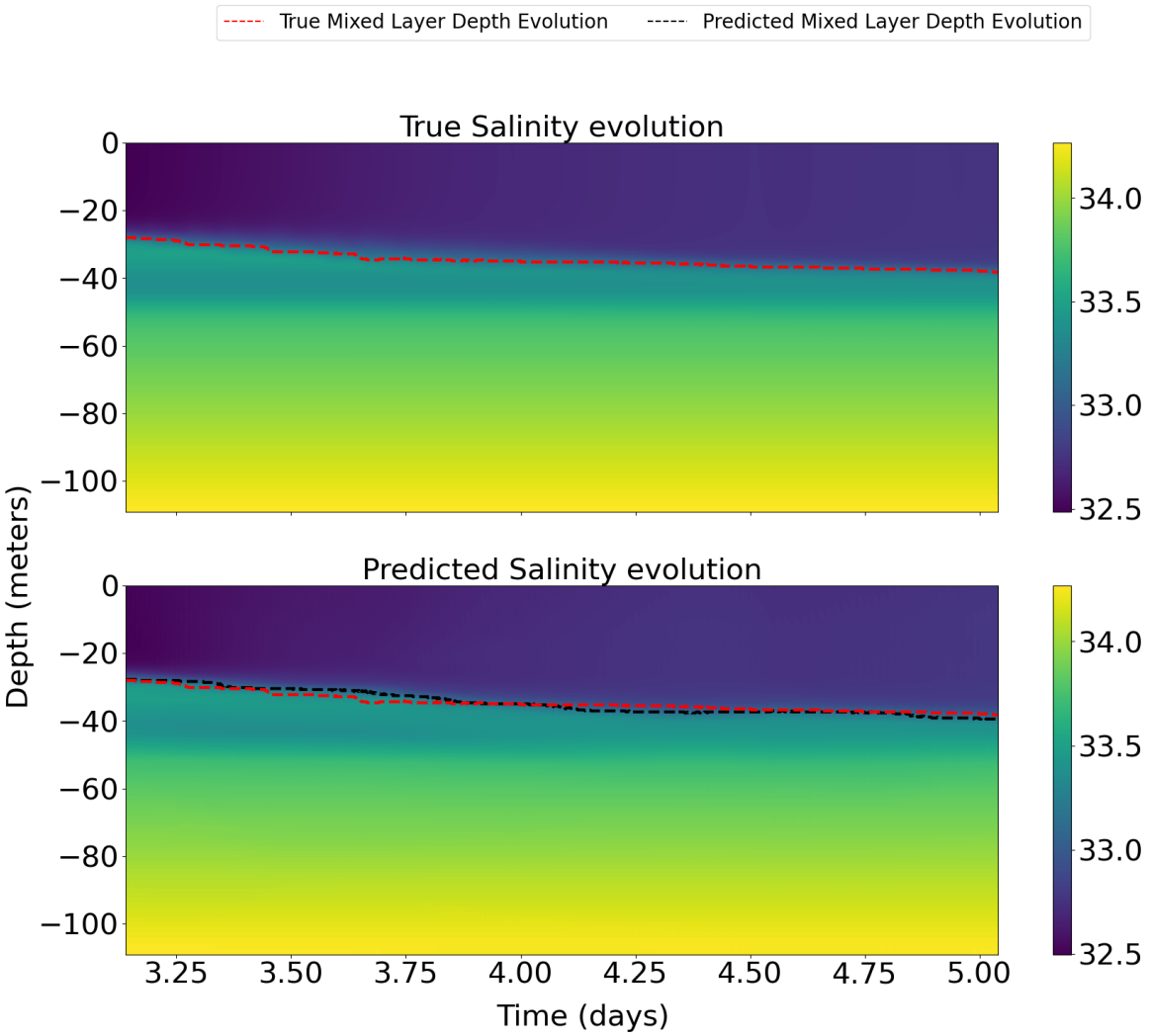

Figure 5.15. Comparison between true and predicted salinity for 2 days.

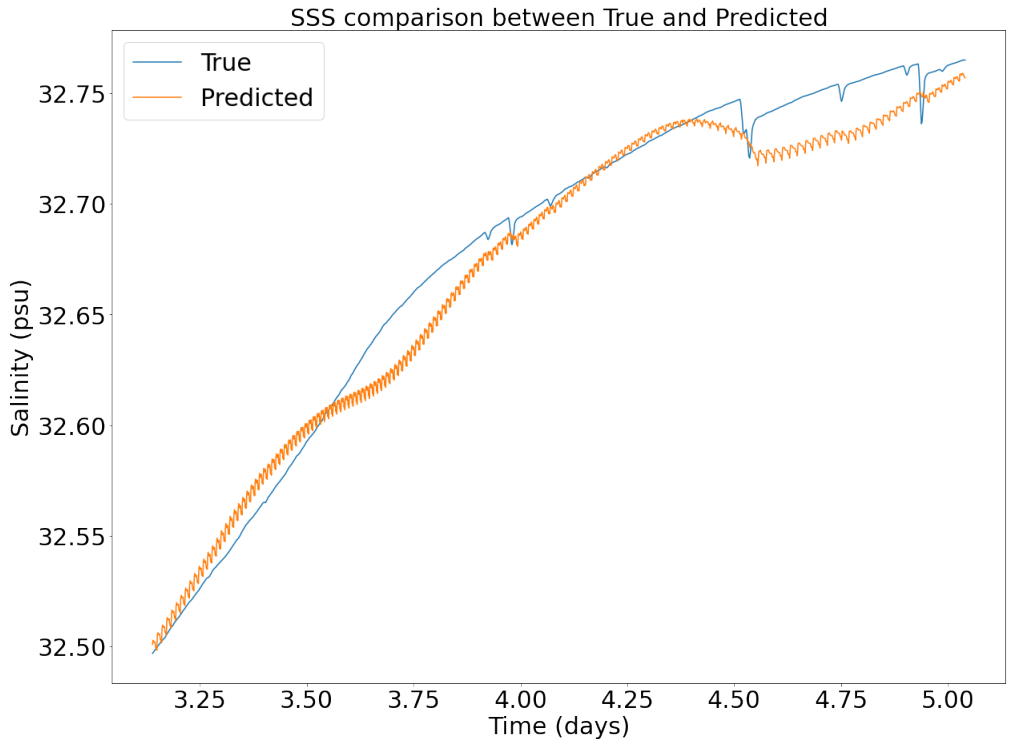

Figure 5.16. Comaprison between true and predicted SSS.

Figure 5.13b shows the error in the prediction compared with the actual reconstructed profile. Figure 5.14 compares the predicted and the true salinity profiles at various time instances. Surprisingly, the error observed during the reconstruction at the mixed layer has now been reduced. The explanation is similar to the one given in the previous section. The overestimation of the salinity profiles during reconstruction is now compensated by under-prediction during the prediction phase(as it can be seen from the Figure 5.13a all the three latent variable predictions are below the true[1] latent variables). Figure 5.15 shows a comparison between the predicted evolution of Sea surface salinity (SSS) and the ground truth. The model does predict the increase in the SSS over time, however, the model fails to accurately capture the influx of fresh water which can be attributed to the LSTM-autoencoder failing to capture this influx. Figure 5.15 shows the evolution of salinity profiles for the entire 2-day period between the truth and the predicted. The RNN-ROM successfully captures the deepening of the mixed layer depth over

[1]Here truth refers to the latent variables evaluated by the LSTM-Autoencoder

time. The maximum error in the prediction of MLD is 3.89m, which is more than the error observed when both phases were used for training. This is opposite to what was observed in the temperature MLD, where MLD error was reduced.

5.3 Performance comparison: training on both active and break phases vs. training only on the active phase

Despite using only half of the training data, the LSTM-ROM's predictions remain comparable to those obtained with the full dataset. This section provides a quantitative comparison of both scenarios based on error metrics and training speed.

5.3.1 Metrics

Mean Absolute Percentage Error (MAPE)

MAPE is defined as:

$$\text{MAPE} = \frac{1}{N_{\text{profiles}}} \sum_{i=1}^{N_{\text{profiles}}} \frac{|\text{True}_i - \text{Reconstructed}_i|}{\text{True}_i}$$

where N_{profiles} represents the number of profiles, and "True" and "Reconstructed" denote the true and reconstructed values, respectively.

Maximum Error

The maximum error is computed as:

$$\text{Maximum Error} = \max\left(\frac{|\text{True} - \text{Reconstructed}|}{\text{Normalizing Constant}}\right)$$

over the entire spatio-temporal domain.

Total Error

Total error is quantified using the Frobenius norm of the error matrix A, which contains the normalized errors across both spatial and temporal dimensions. The Frobenius norm is calculated as:

$$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

where each element of matrix A represents a normalized error.

R2-Score

The R^2 score is given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where y_i are the true values, $\hat{y}_i$ are the predicted values, and $\bar{y}$ is the mean of the true values.

Mean Squared Error (MSE)

The Mean Squared Error is defined as:

$$\text{MSE} = \frac{1}{N_{\text{profiles}}} \sum_{i=1}^{N_{\text{profiles}}} (\text{True}_i - \text{Predicted}_i)^2$$

5.3.2 Evaluation of the metrics for each cases

Table 5.3. Metrics for LSTM-autoencoder-temperature .

LSTM-autoencoder	case 1	case 2
Train time (s)	429.92	194.41
SST (MAPE)	0.00084	0.0003
Max error	0.12	0.12
Total Error	12.79	13.54

Table 5.3 and 5.5 show the various metrics for salinity and temperature reduction using LSTM-autoencoder. There is a significant improvement in temperature and salinity predictions

Table 5.4. Metrics for GRU-temperature .

GRU	case 1		case 2	
Train Time	227.779		101.49	
	R2-Score	MSQ-error	R2-Score	MSQ-error
Latent 1	0.98	0.00016	-2.92	0.0003
Latent 2	-0.06	0.0002	0.96	0.0002
Latent 3	0.901	0.0001	0.92	5.44e-05
Average	0.61	0.00015	-0.34	0.00018

in the training time for case 2. Additionally, the accuracy of both SST and SSS predictions has improved in case 2. However, when examining the maximum error, there is no improvement in temperature, and the error in salinity shows a slight increase from 0.142 to 0.185. The total salinity error remains unchanged across both cases, while the total temperature error increases slightly from case 1 to case 2. Despite this, the error in the mixed layer shows improvement for both temperature and salinity in case 2, although the error at the mixed layer depth has slightly increased. Overall, training exclusively with active phase data offers an advantage by accelerating training time without compromising the error metrics.

Tables 5.4 and 5.6 show the metrics used to compare the two cases for temperature and salinity in testing the GRU model. The training time for the GRU model in case 2 has improved by more than 50% for both temperature and salinity. The mean squared error remains minimal for both cases across temperature and salinity profiles. However, the average R^2 score (considering all three latent variables) has increased for salinity predictions in case 2, while it has decreased significantly for temperature. This is due to the negative R^2 score for latent variable 1. As shown in Figure 5.4a, this is attributed to the notable difference in the amplitude of the evolution, even though the overall fluctuation follows the correct trend. The R^2 scores for the other two latent variables remain high in case 2 for temperature. Therefore, for salinity predictions, case 2 not only enhances training speed but also improves the accuracy of the latent variable evolution predictions. In contrast, temperature predictions have degraded, though the trend of the evolution is still preserved.

Table 5.5. Metrics for LSTM-Autoencoder-salinity .

LSTM-Autoencoder	case 1	case 2
Train time (s)	447.929	191.72
SSS (MAPE)	0.0008	0.0006
Max error	0.1426	0.185
Total Error	17.233	17.278

Table 5.6. Metrics for GRU-salinity

GRU	case 1		case 2	
Train Time(s)	233.851		107.34	
	R2-Score	MSQ-error	R2-Score	MSQ-error
Latent 1	0.99	0.0002	0.834	0.0006
Latent 2	0.77	0.0001	0.840	0.0014
Latent 3	0.13	0.0016	0.926	0.0004
Average	0.63	0.00015	0.86	0.0008

5.3.3 Comparison between RNN-based reduced order model (ROM) and large eddy simulation (LES)

The LES was carried out using a 256-core processor, running for 12 days (Pham et al., 2023). Therefore, acquiring high-fidelity data is expensive and time-consuming. Thus resorting to the usage of minimum data is necessary in order to predict the required region. Table 5.7 details the configuration used and the cost of the simulation to perform a Large Eddy Simulation of the upper ocean. The table 5.7 also details the GPU configuration used to run the RNN-ROM model.

Table 5.7. Comparison between Large Eddy Simulation (LES) and RNN-ROM

	CPU		GPU (NVIDIA GeForce RTX 2080 Ti)
	Cores	Simulation time (seconds)	Total Training time (seconds)
LES	256	147320.64	-
Case 1	8	-	1339.2
Case 2	8	-	587.52

We can see a significant reduction in the time for prediction. The time for LES simulation is back-calculated using the time required to simulate the full two-week simulation. The total training time refers to the time required to train both the salinity and temperature fields. This is

done because these equations from LES are coupled, and hence they are solved simultaneously by the LES solver. Considering the ratio of simulation time for LES and training time[2] for RNN-ROM, the ML model considering case 2 speeds up the calculation by approximately 250 times. However, considering the time required to tune the model, it adds additional time. Overall, we can conclude that the model with case 2 achieved a 25-fold reduction in time compared to the Large Eddy Simulation.

[2]Note that test or inference time is not considered as it is negligible when compared to total training time

Chapter 6

Predicting break phase profiles from the past data

In this chapter, the profiles for the final 3 days of the break phase are predicted. This involves predicting a total of 1500 profiles in the future[1]. Training is performed using data from both the active and break phases. The first 6700 profiles, corresponding to 10 days of data, are used for training, while the next 300 profiles are reserved for validation. The chapter focuses on predicting the diurnal fluctuations of solar heat flux, which is a prominent dynamic characteristic of the break phase.

6.1 Predicting temperature profiles

This section provides results of the predictions of the temperature profile for the final 3 days of the break phase. The hyperparameters shown in table 6.1 are obtained after tuning the model to minimize the validation error.

Table 6.1. List of tuned hyperparameters for predicting the temperature profiles at end of break phase

	Epochs	Hidden units	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable
LSTM-autoencoder	300	65	1	8.00E-04	-	0.5	25	3
GRU-Temporal	250	56	1	1.00E-03	1.00E-07	0.69	30	-

[1]The predictions are not considered extrapolation, as all inputs to the RNN model lie within the range of the training data.

6.1.1 Reconstruction of temperature profiles

The reconstruction of temperature profiles is shown in Figure 6.2 at various time instances. Figure 6.9 shows the normalized error, defined in the previous chapters as, $\frac{T_{\text{actual}} - T_{\text{predicted}}}{\text{Normalizing constant}}$, where the normalization constant is given by $\frac{1}{N_{\text{profiles}}} \left(\sum_{i=1}^{N_{\text{profiles}}} \left(T_{i,\max} - T_{i,\min} \right)_{\text{MLD}} \right)$ in the reconstruction by the LSTM-autoencoder. The normalization constant evaluates to 1.21 for this particular case. N_{profiles} equals 1500. The maximum error is 0.21, which is 21% of the total change in the mixed layer depth over a 3-day period, and occurs during the later part of the prediction.

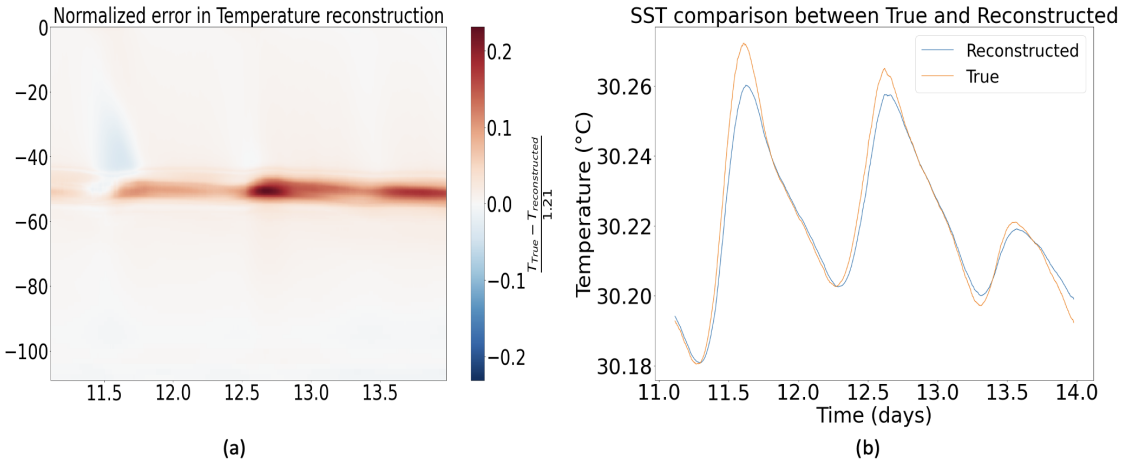

Figure 6.1. (a) Normalized reconstructed Error (b) Reconstructed Sea Surface Temperature.

As explained in the previous chapters, the high gradient of the temperature near the mixed layer causes errors to shoot up even with a slight shift in the temperature profile in the vertical direction. The number of latent variables used to reconstruct the temperature profiles is 3. Figure 6.1 shows the reconstructed SST for 3 days into the future. The maximum error in the SST is less than 0.01 °C. Since there are limited dynamics during this period, the profiles remain roughly constant, with diurnal fluctuation in the SST. Figure 6.3 shows the reconstruction from LSTM-autoencoder throughout 3 days.

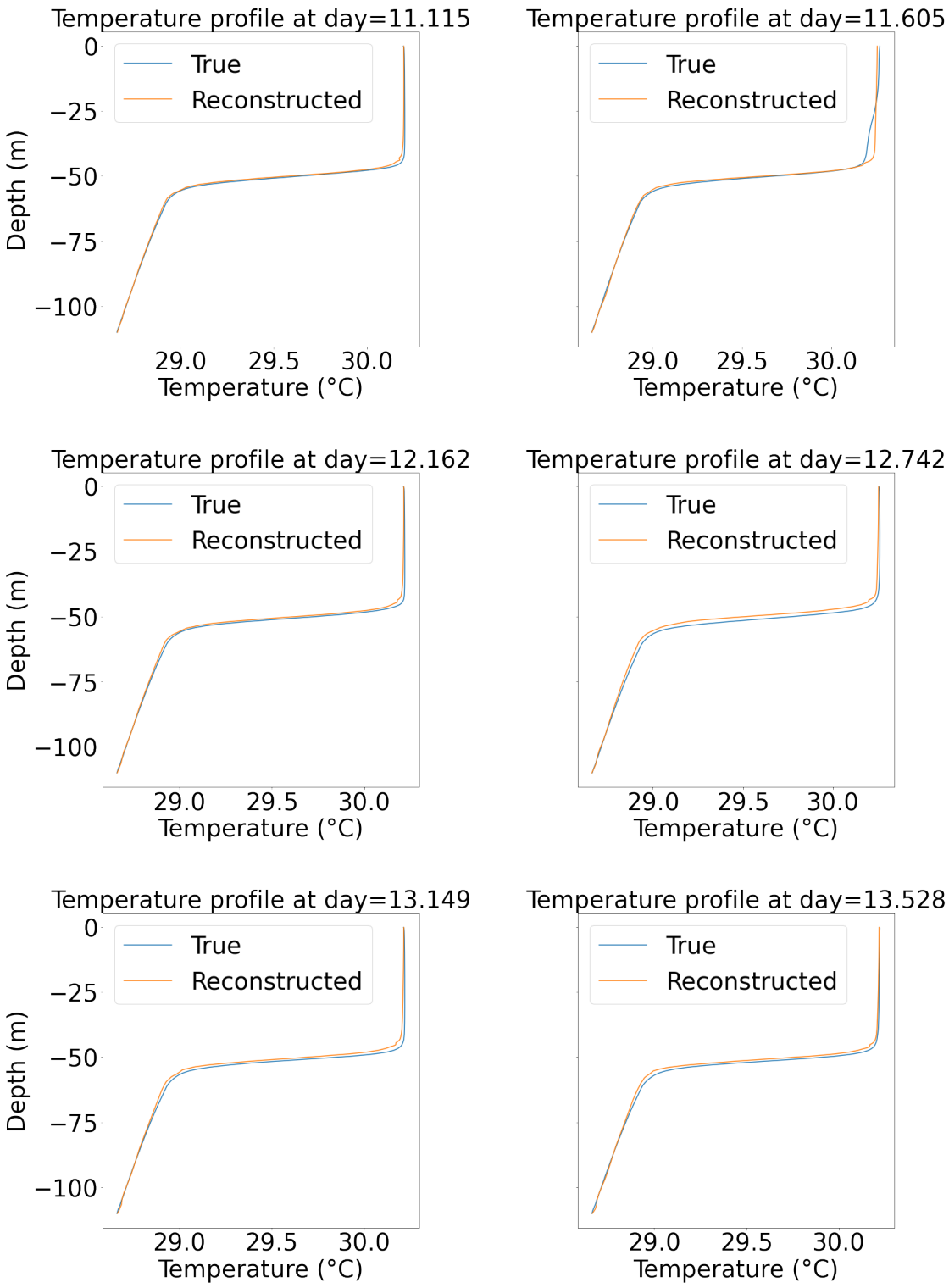

Figure 6.2. Temperature reconstruction at various time instances.

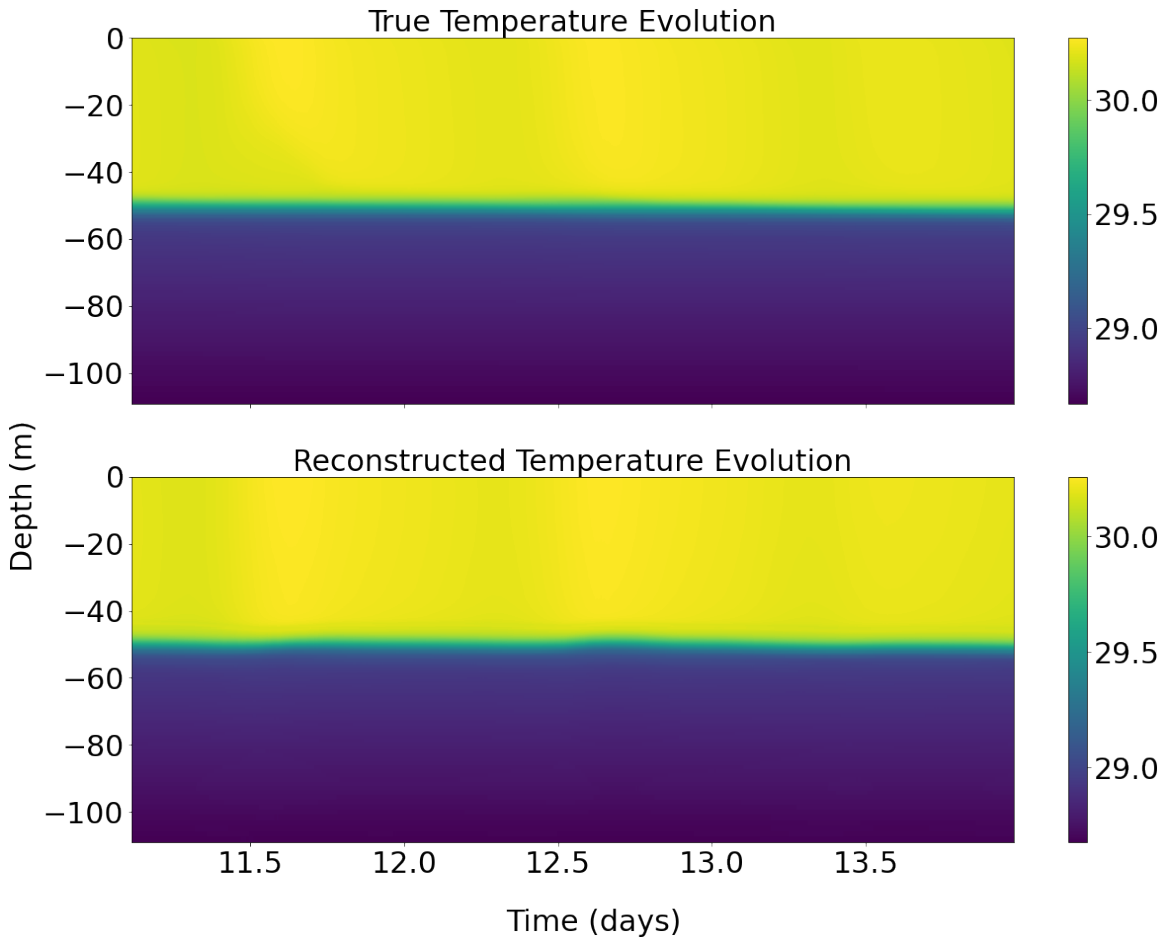

Figure 6.3. Comparison between true and reconstructed Temperature over 3 days.

6.1.2 Temporal evolution of the latent variables and prediction of temperature profiles

The temporal evolution of the latent variable for the 3-day period in the future is shown in figure 6.4a. The prediction from the RNN-ROM and the truth agree with each other. Figure 6.4b shows the comparison between the predicted region and the overall 14-day period along with the prediction made by the RNN-ROM. Given the latent evolution of the second variable, the maximum difference between the true and predicted values is 0.01, while the range between the minimum and maximum values of this latent variable is 0.2. This error represents 5% of the maximum change in the evolution of the second latent variable. Therefore, the discrepancy is minimal and should not significantly impact the accuracy of the predicted temperature profiles

after decoding the coefficients. To verify this, figure 6.6 shows the temperature profiles at various time instances. The discrepancy between the predicted and the truth is barely visible from these plots. Figure 6.5a shows the normalized error throughout the prediction period. The maximum value is 0.16, which corresponds to 16% of the average change in the temperature value at the mixed layer depth over a 3-day period.

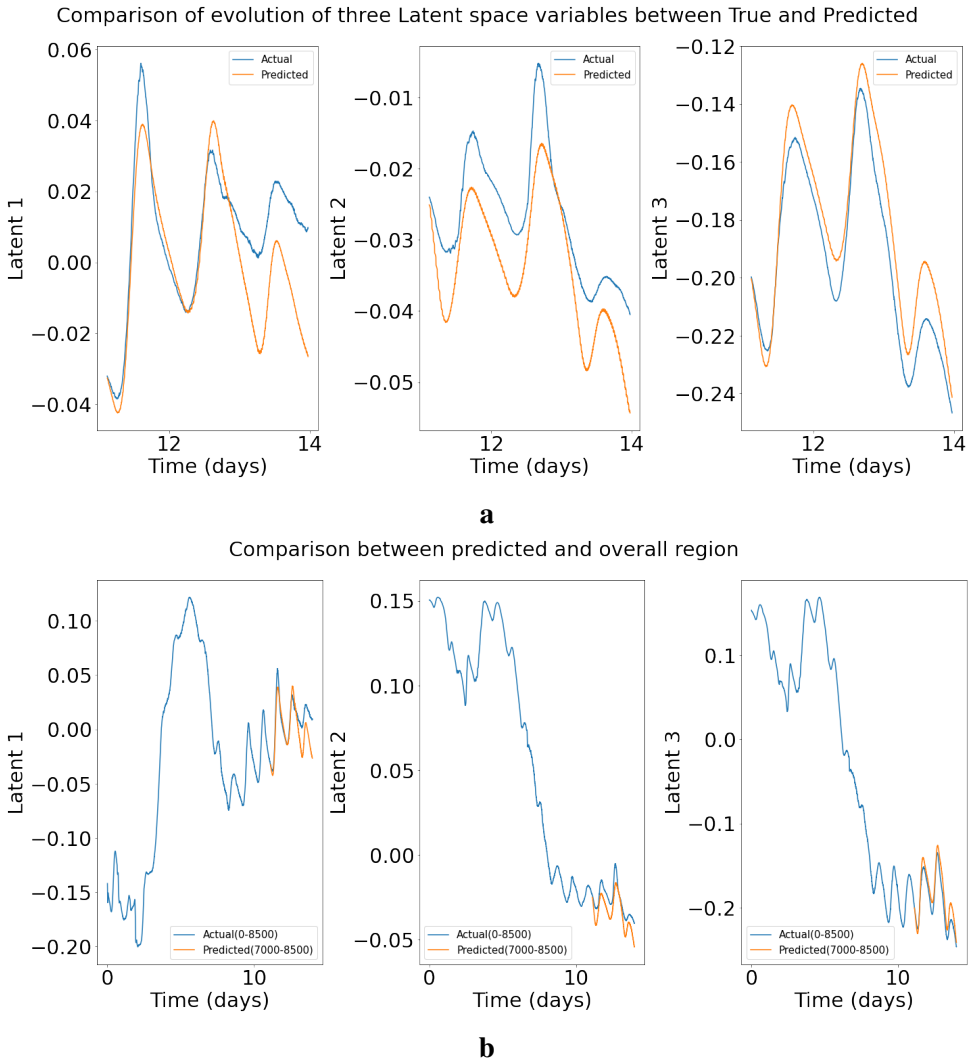

Figure 6.4. (a) Temporal evolution of latent variables over 3 days (b) Temporal evolution of latent variables over 14 days.

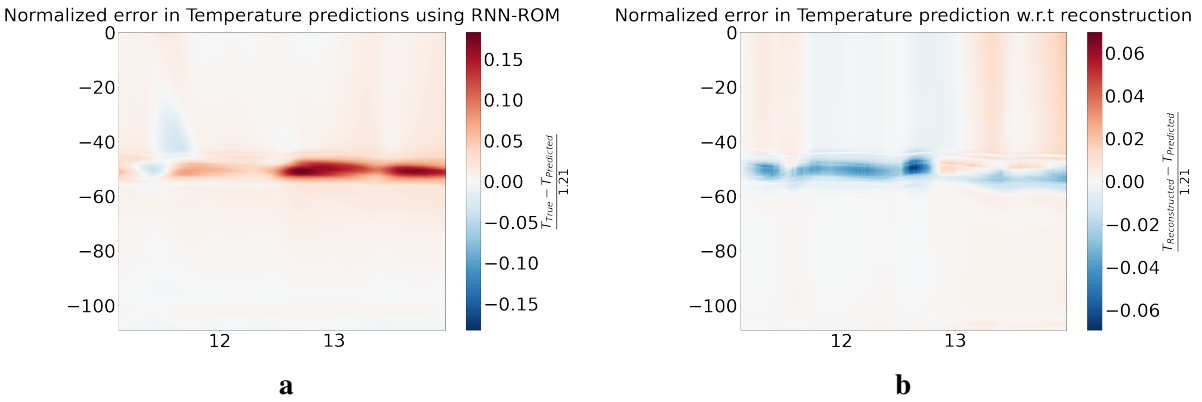

Figure 6.5. (a) Normalized error in temperature prediction over the 3-day period at the end of the break phase (b) Normalized error in temperature predictions with respect to reconstructed temperature profiles over 3 days at the end of the break phase.

Figure 6.5b shows the error between true reconstruction and prediction. The maximum prediction error with respect to the reconstruction is 6%, which suggests that most of the error originated from reconstructing the temperature profiles. Figure 6.8 shows the prediction of Sea surface temperature (SST) over 3 days. The model accurately predicts the diurnal fluctuations of the SST, though there is a slight shift, with the maximum error being approximately 0.01 °C. Figure 6.7 shows the comparison of evolution between truth and predicted for the temperature profiles throughout the 3-day prediction. From this, we can conclude that RNN-ROM accurately predicts the necessary dynamics for the 3 days into the future for the temperature profiles.

6.2 Predicting salinity profiles

This section provides the prediction of the salinity profile during the final 3 days of the break phase. The hyperparameter for the current predictions is shown in table 6.2 obtained from minimizing the validation losses.

Table 6.2. List of tuned hyperparameters for predicting the salinity profiles at end of break phase

	Epochs	Hidden units	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable
LSTM-Autoencoder	300	60	1	8.00E-04	-	0.52	25	2
GRU-Temporal	250	69	1	1.00E-04	1.00E-07	0.69	26	-

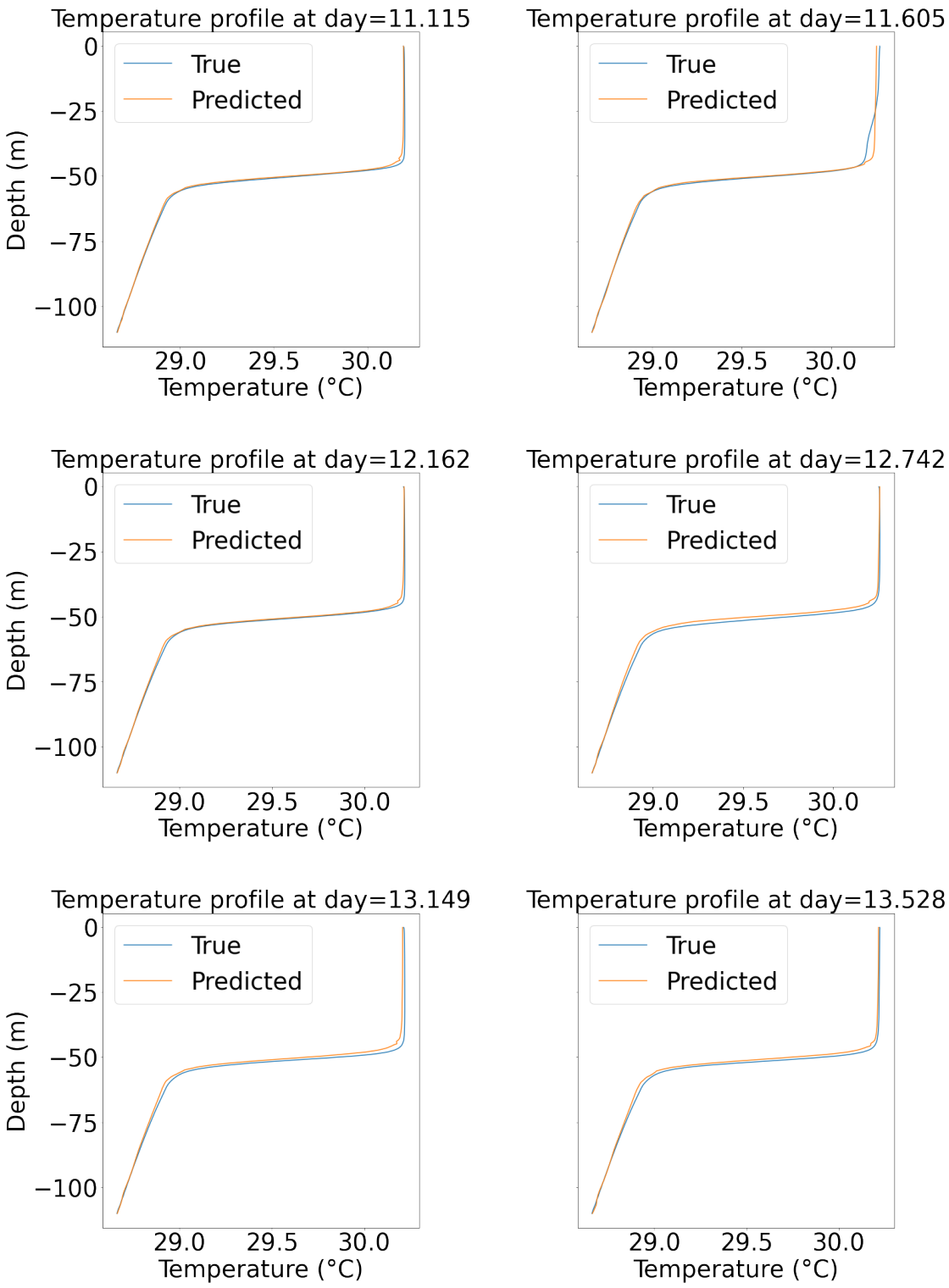

Figure 6.6. Temperature predictions at various time instances.

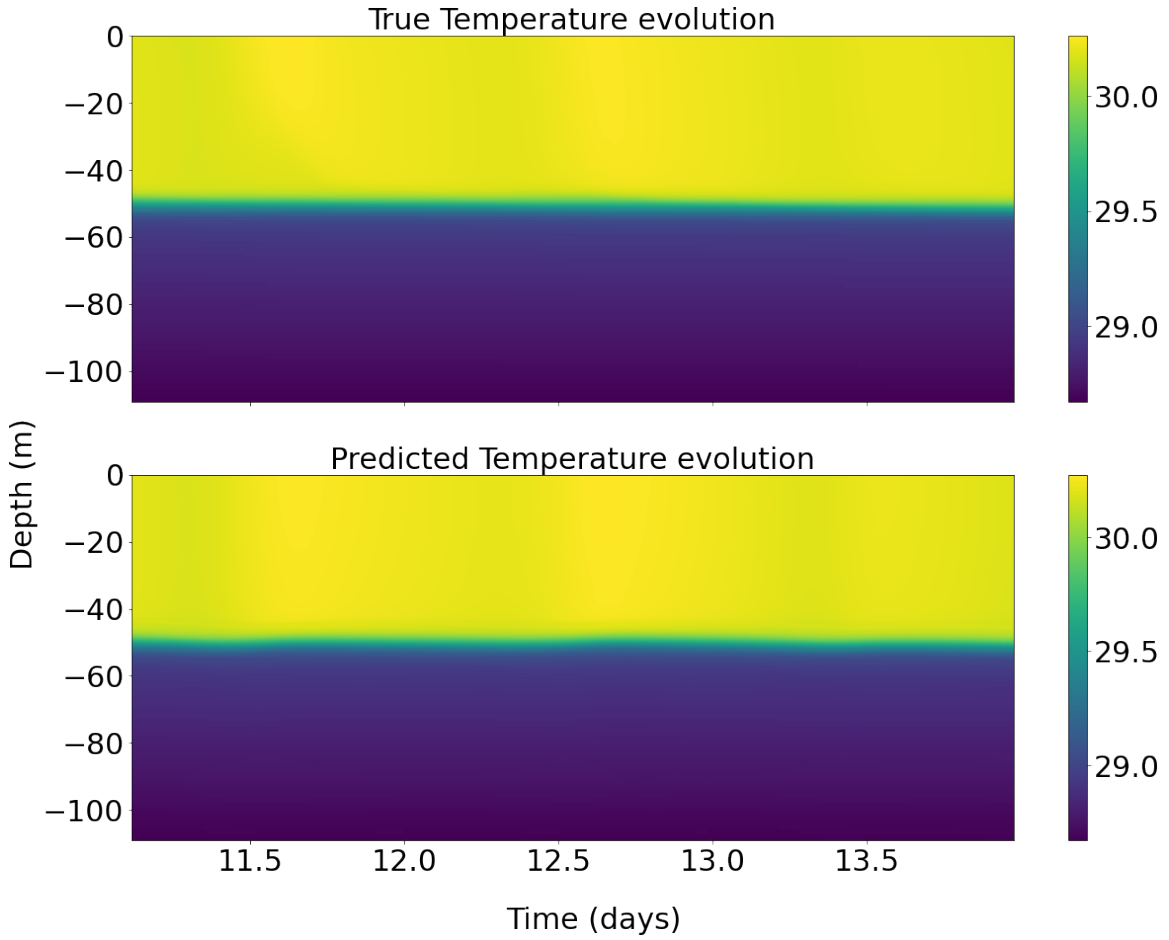

Figure 6.7. Comparison between true and predicted temperature over the period of 3 days.

6.2.1 Reconstruction of salinity profiles

The reconstruction of Salinity profiles at various time instances is shown in figure 6.11. Figure 6.9a shows the error in the reconstruction for the 3 days. The normalized error is calculated pointwise defined by $\frac{max(T_{actual}-T_{predicted})}{\text{Normalizing constant}}$ and the value of normalizing constant is taken to be 1 psu. Unlike the reconstruction of temperature profiles, salinity reconstruction for the 3 days in the future does not suffer from the vertical shift as seen from Figure 6.11, with the maximum error being 0.04, i.e, the maximum error is 4% of the average change in the salinity values at the mixed layer depth. The number of latent variables used to reconstruct the salinity profiles is 2. Figure 6.9b shows the reconstruction of Sea surface salinity (SSS) over 3 days. The maximum error in the SSS reconstruction is less than 0.005 psu. Figure 6.10 compares the evolution of

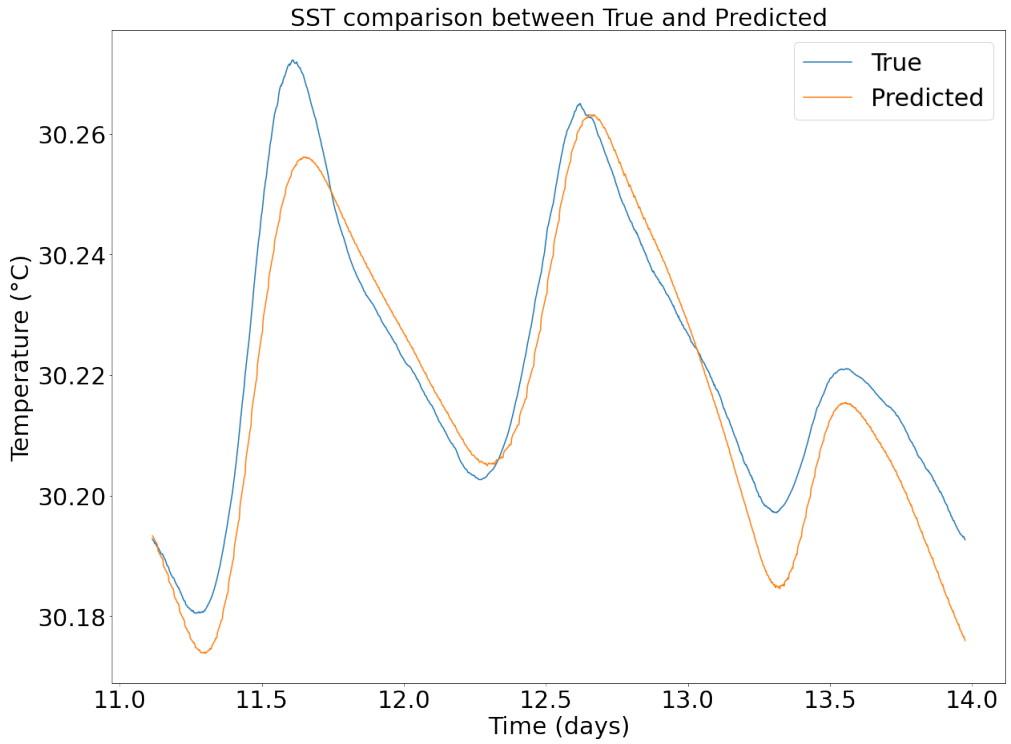

Figure 6.8. Comparison between true and predicted SST.

salinity profiles between the true and the predicted values over 3 days. The LSTM-autoencoder has nearly perfectly reconstructed the salinity profiles for future predictions.

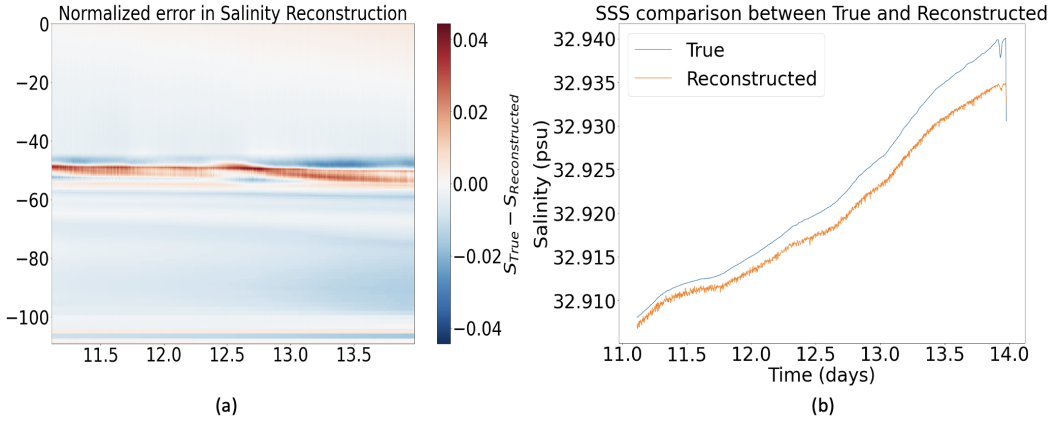

Figure 6.9. (a) Normalized reconstructed Error (b) Reconstructed Sea Surface Salinity.

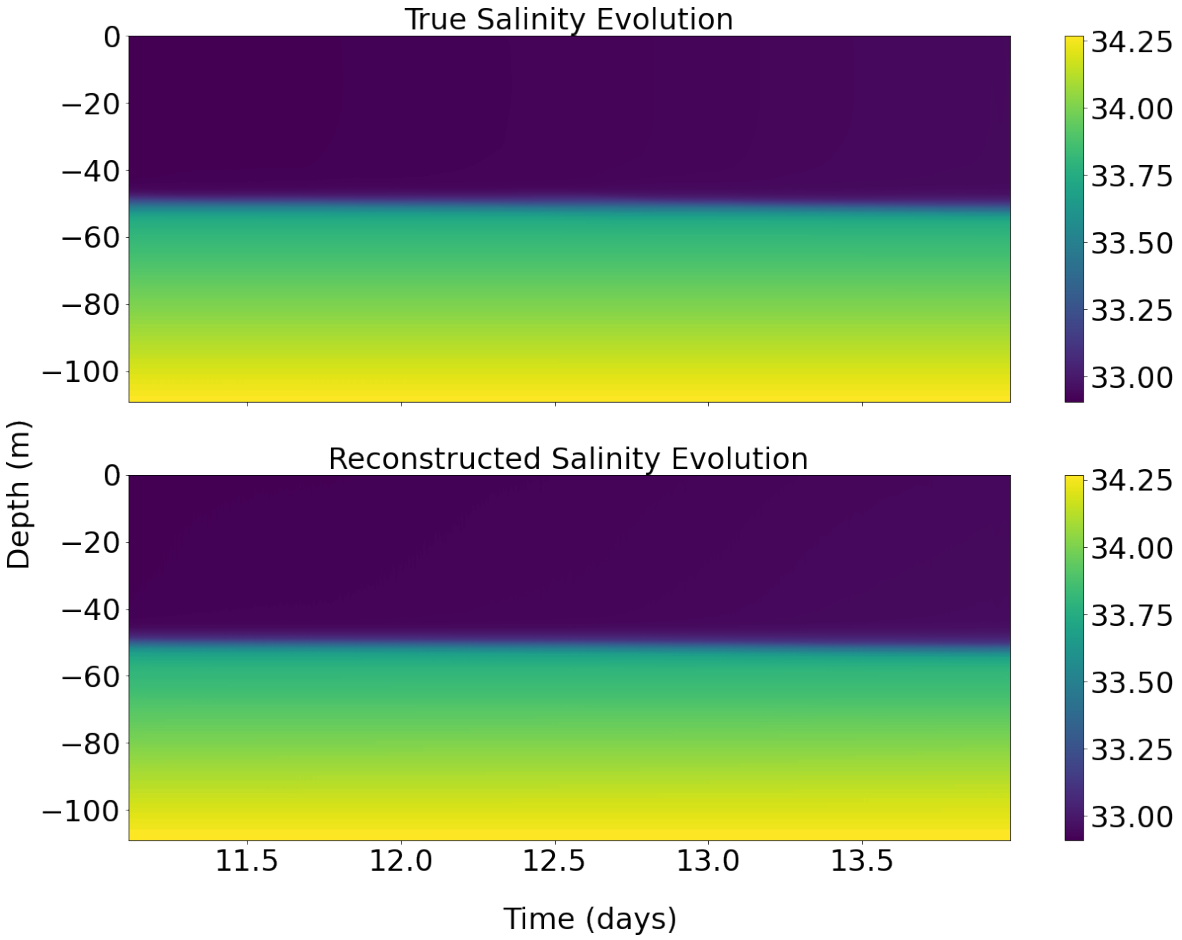

Figure 6.10. Comparison between true and reconstructed salinity for 2 days.

6.2.2 Temporal evolution of the latent variables and prediction of salinity profiles

The temporal evolution of the latent variable for a 3-day period for the future states is shown in Figure 6.12a. Both latent variables agree well with the true encodings, showing only slight discrepancies. This discrepancy is minimal as seen by Figure 6.12, which compares the predicted region plotted against the entire 14-day evolution. Given that this region is relatively small, the RNN-ROM predictions are consistent with the true evolution of the latent variables. This can be seen from Figure 6.14, which compares the true and predicted salinity profiles at various time instances, demonstrating that the predictions align precisely with the ground truth. Additionally, Figure 6.13a shows the normalized error plot between the truth and the predicted.

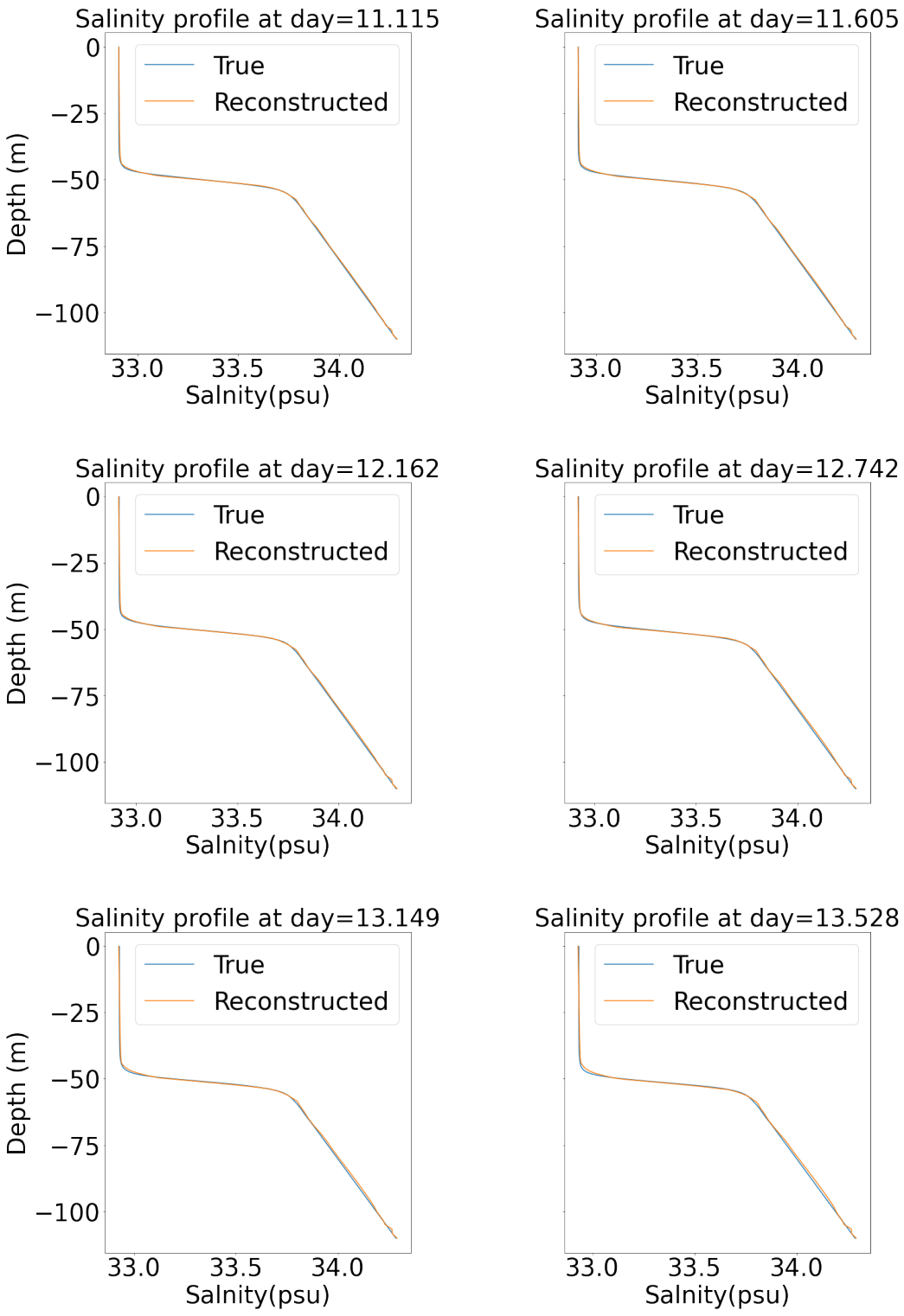

Figure 6.11. Salinity reconstruction at various time instances.

The maximum prediction error is approximately 0.065, which represents 6.5% of the average change in salinity at the mixed layer depth over 3 days.

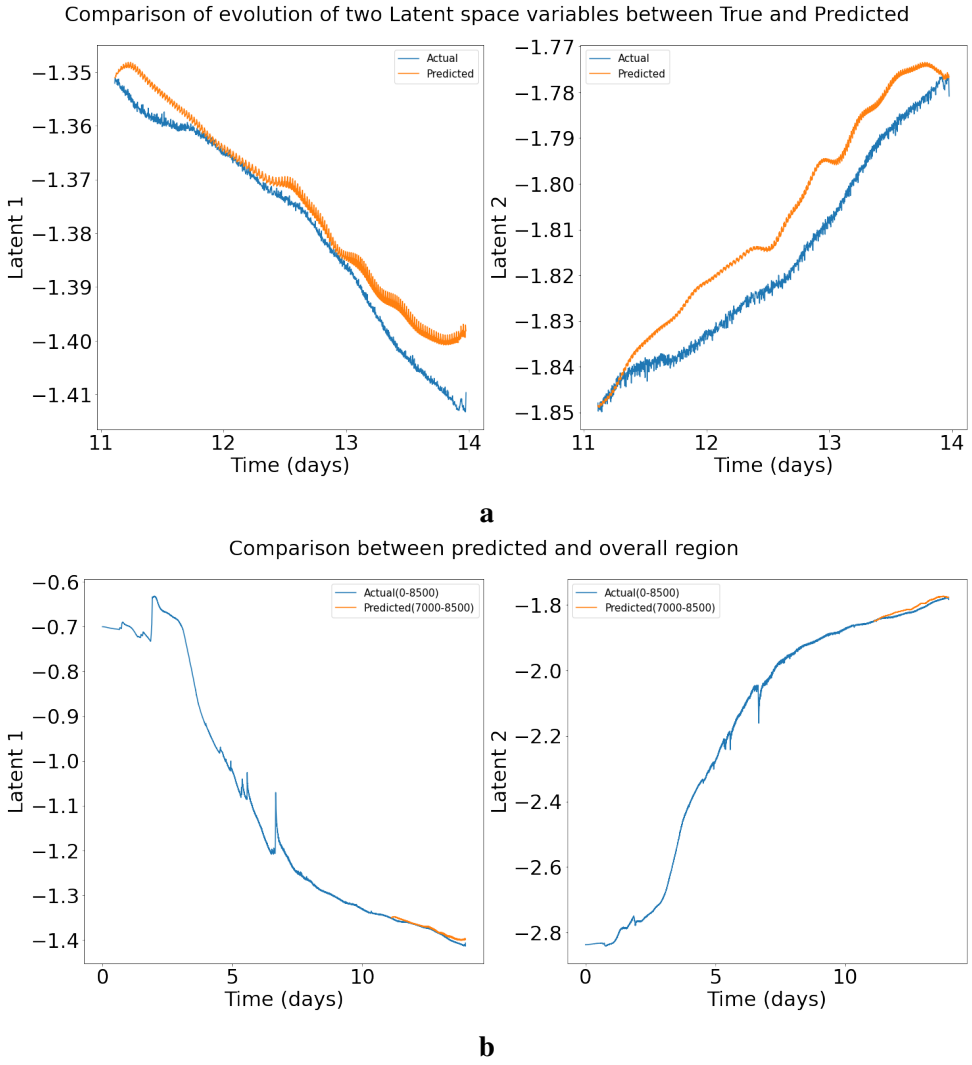

Figure 6.12. (a) Temporal evolution of latent variables over 3 days (b) Temporal evolution of latent variables over 14 days.

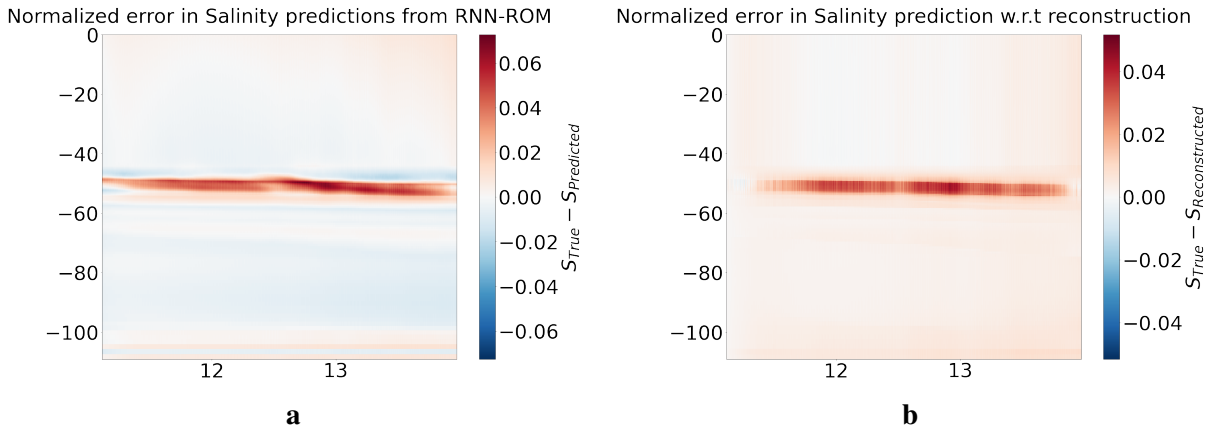

Figure 6.13. (a) Error in salinity prediction (b) Error in salinity prediction with respect to reconstructed profiles.

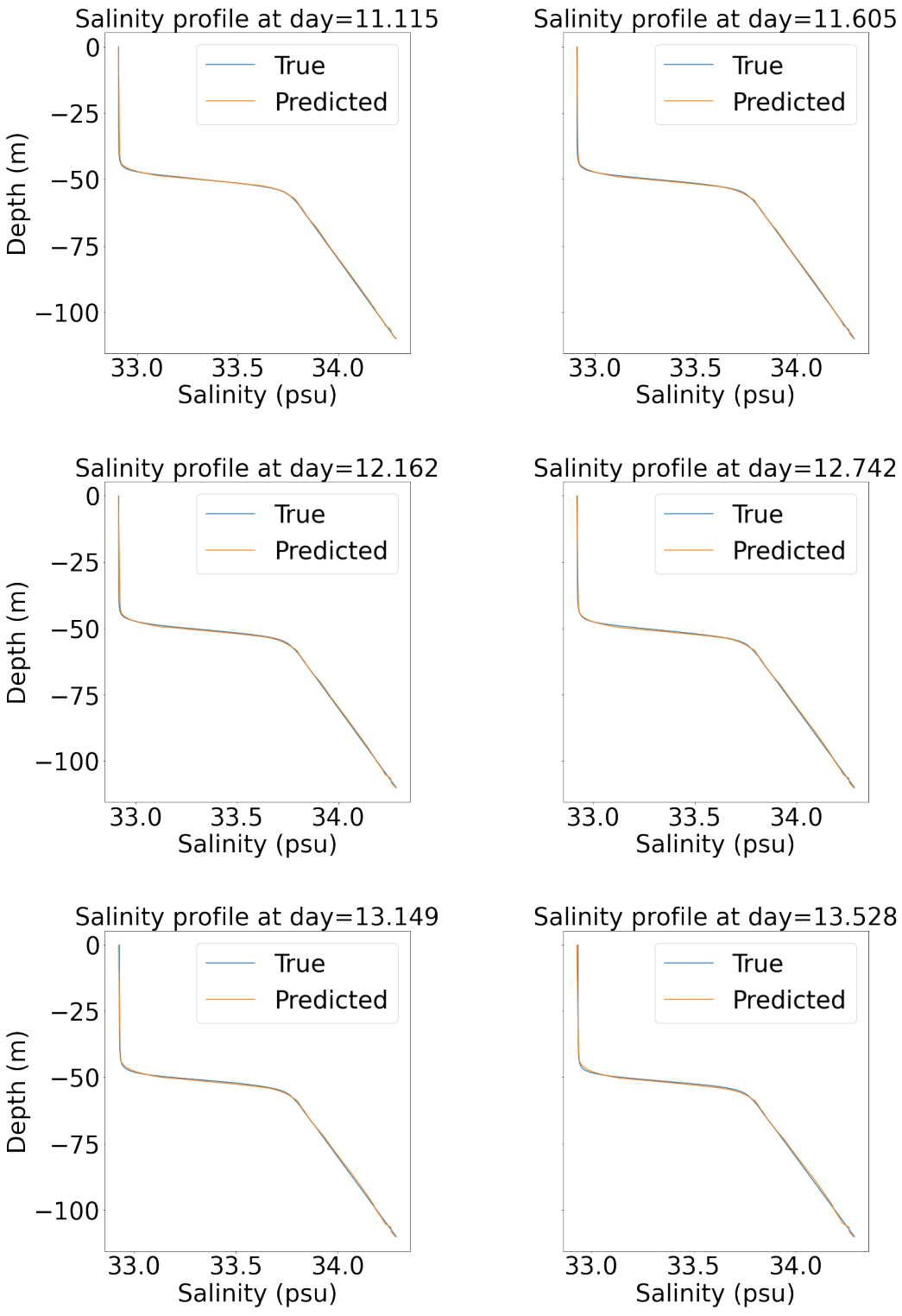

Figure 6.14. Salinity predictions at various time instances.

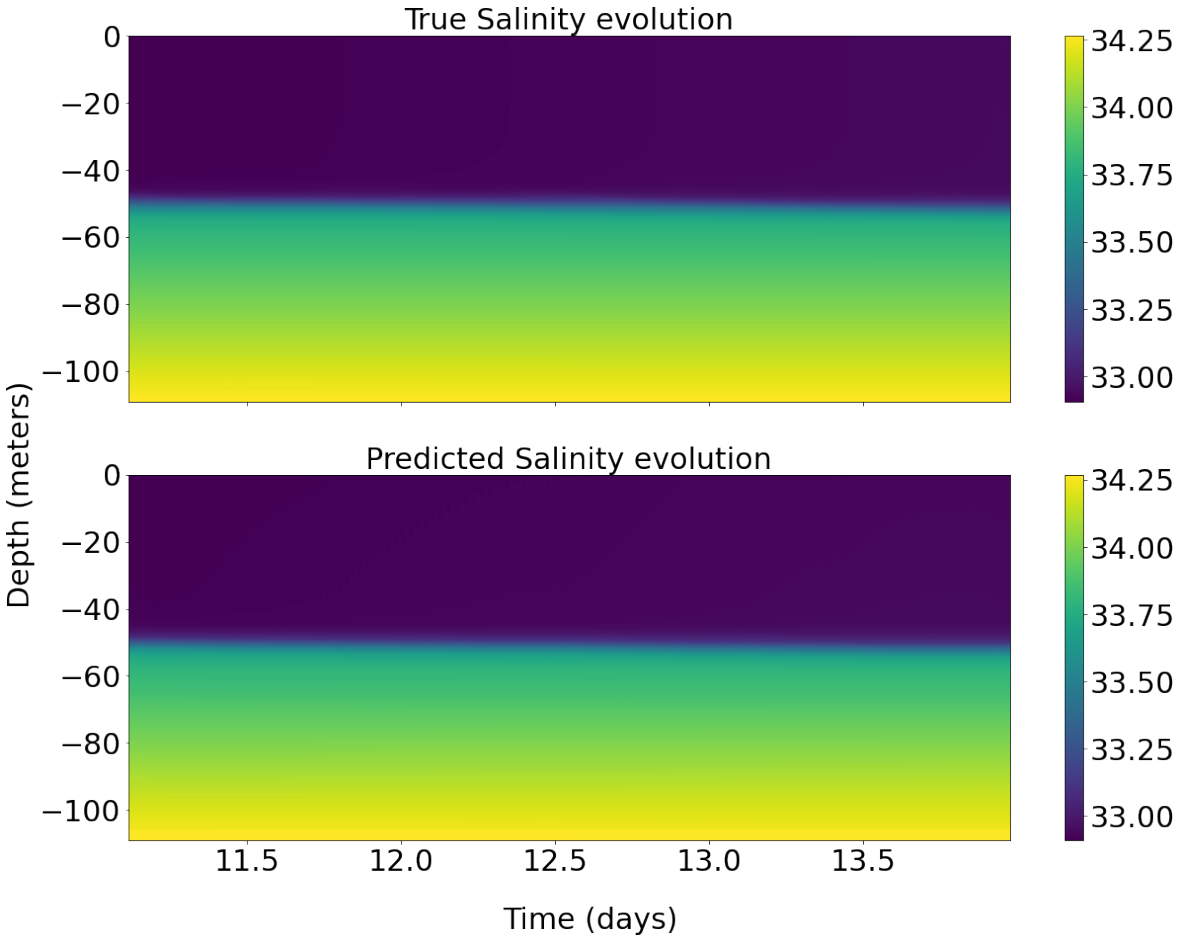

Figure 6.15. Comparison between true and predicted salinity for 3 days in the future.

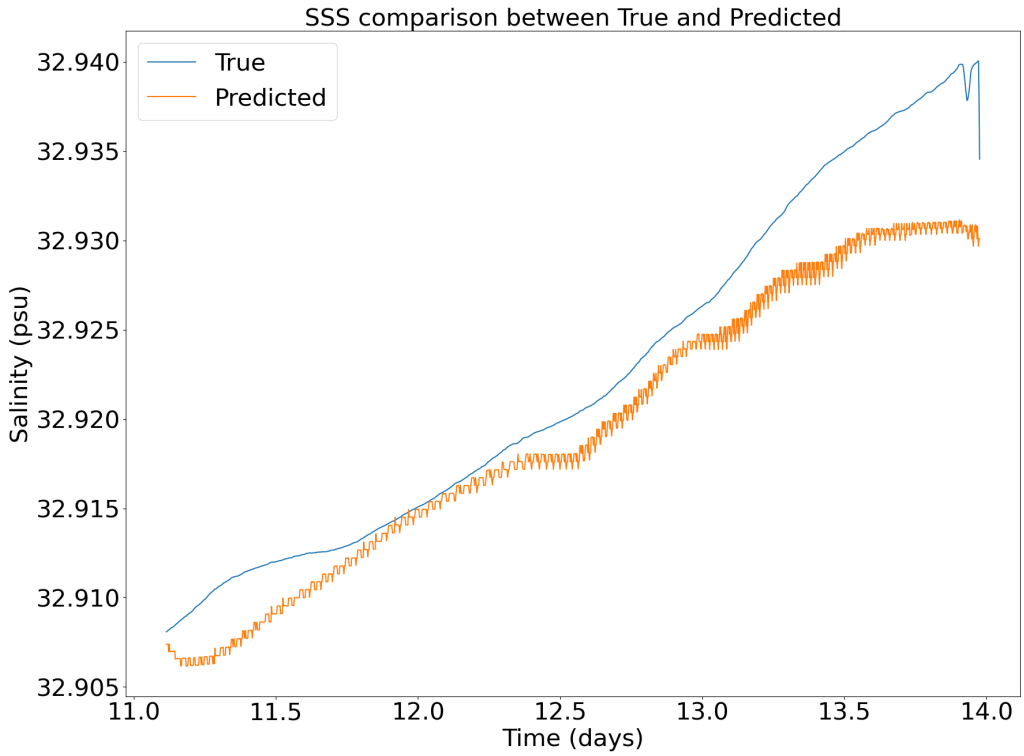

Figure 6.16. Comaprison between true and predicted SSS.

Figure 6.13b shows the normalized error between the reconstructed and predicted salinity profiles. Figure 6.16 compares the predicted Sea surface salinity (SSS) with the true values, showing a maximum error of approximately 0.015 psu. The RNN-ROM successfully predicts an increase in SSS over the 3 days. Figure 6.15 shows the evolution of salinity over these 3 days. Overall, the RNN-ROM accurately predicts salinity as a function of depth throughout the 3 days at the end of the break phase.

Chapter 7

Predicting break phase profiles from the active phase

This chapter focuses on evaluating the long-term prediction capability of the RNN-ROM model. First, the model from Chapter 5 is used to predict a portion of the active phase and the entire break phase. Subsequently, it is demonstrated that achieving long-term predictions requires incorporating the necessary dynamics into the model.

7.1 Predicting temperature profiles

The hyperparameters for this section are the same as given in table 5.1 since the model used here is the same as the one mentioned in the chapter 5 section 5.1. The parameters obtained in section 5.1 are extended to predict the temperature profiles in the break phase, thus also testing the pure extrapolation[1] capability of the RNN-ROM model for the temperature profiles.

7.1.1 Reconstruction of temperature profiles

Figure 7.2 compares the true and reconstructed temperature profiles at various time instances. The first five profiles correspond to the active phase while the next five profiles represent the break phase, where no training data was used. The reconstruction is almost accurate for the first few days in the break phase, up to day 10. However, the reconstructed profiles for the

[1]This problem is extrapolation as the predicted region is out of the Training hull

last four days show a slight vertical shift, causing the error – defined pointwise as $\frac{T_{True}-T_{Reconstructed}}{1.54}$ – to increase, as seen by the figure 7.1a. The normalizing constant of 1.54 is found by taking the average variation of the temperature in mixed layer depth (MLD) over 11 days. The maximum error in the reconstruction is 0.4, which is 40% of the total change in the temperature value at the MLD over 11 days, and is observed towards the later period of the break phase. Figure 7.1b shows the comparison between the true and reconstructed Sea surface temperature (SST), showing that the reconstruction during the break phase is nearly accurate. However, due to the significant vertical shift in the profiles, it can be concluded that the LSTM-autoencoder has not fully captured the gradual deepening of the MLD in this phase. Figure 7.3 shows the Temperature reconstruction for the entire 10-day predictions, and overall reconstructions from the LSTM-autoencoder are acceptable.

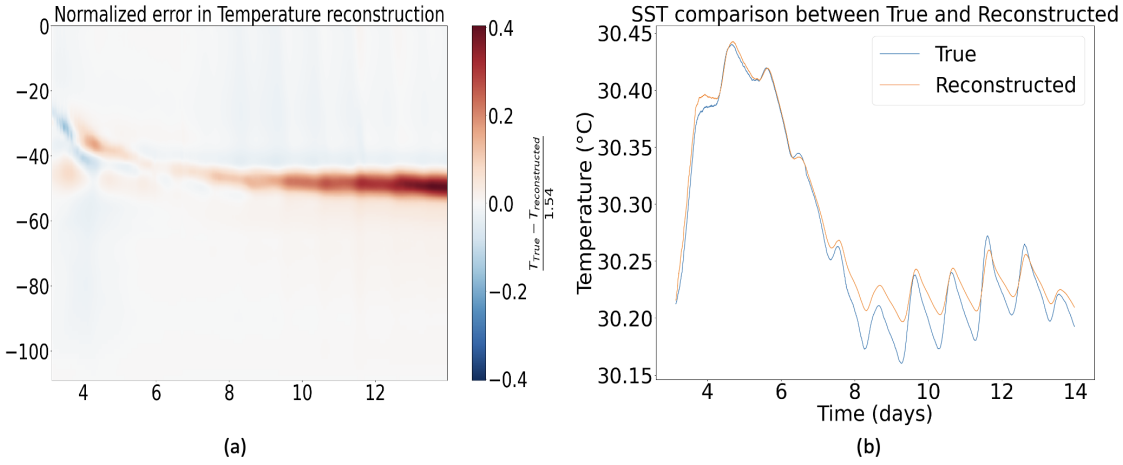

Figure 7.1. (a) Normalized reconstructed error (b) Reconstructed Sea Surface Temperature.

.

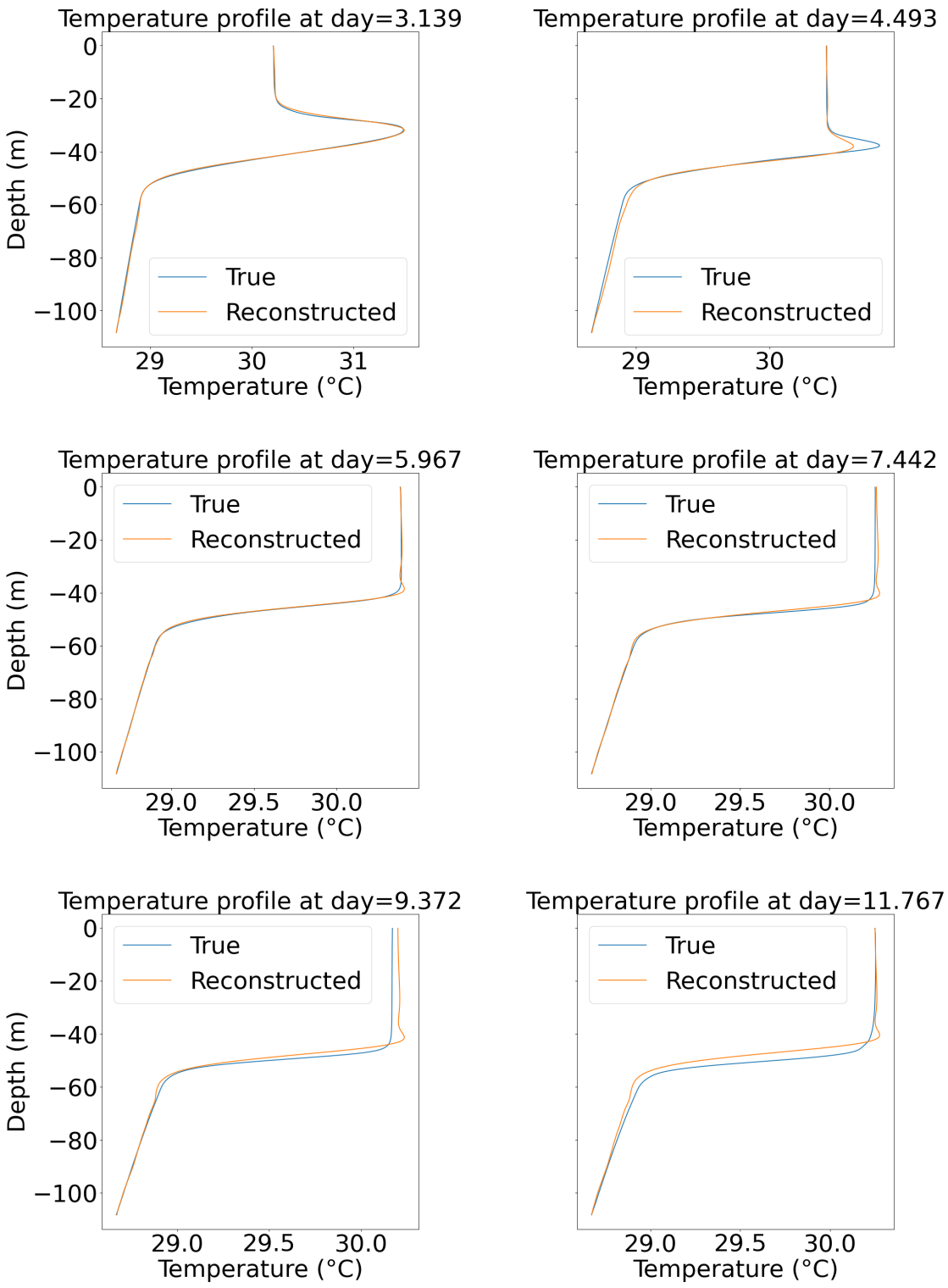

Figure 7.2. Temperature reconstruction at various time instances.

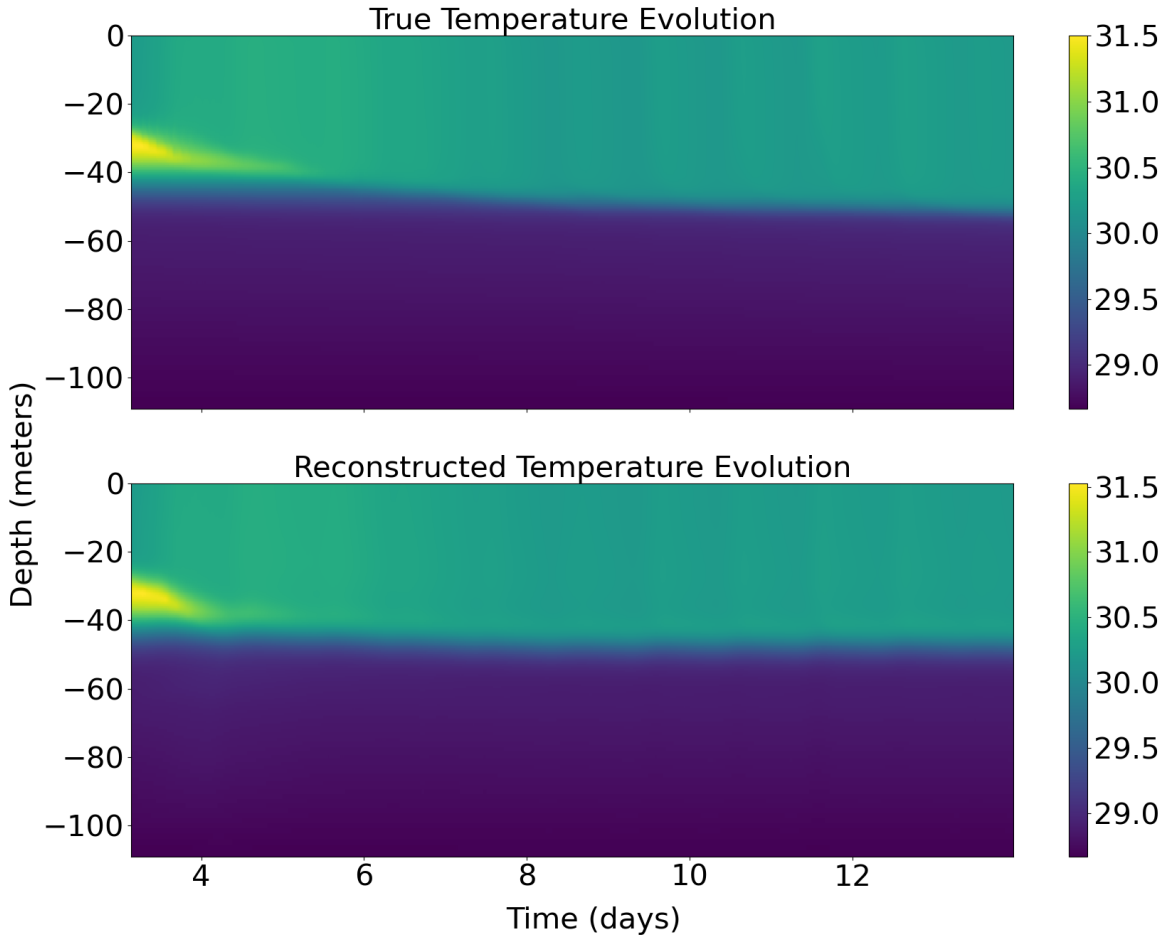

Figure 7.3. Comparison between true and reconstructed Temperature over 11 days.

7.1.2 Temporal evolution of the latent variables and prediction of temperature profiles

The Temporal evolution of the latent variable for the entire 10-day period is shown in figure 7.3. The prediction is accurate only in the active phase where the GRU was mostly trained. In the break phase, the predictions remain accurate up to day 1 after the onset of the break phase. After one day into the break phase, all three latent variables begin to diverge. Figure 7.5 shows the temperature profiles at various time instances.

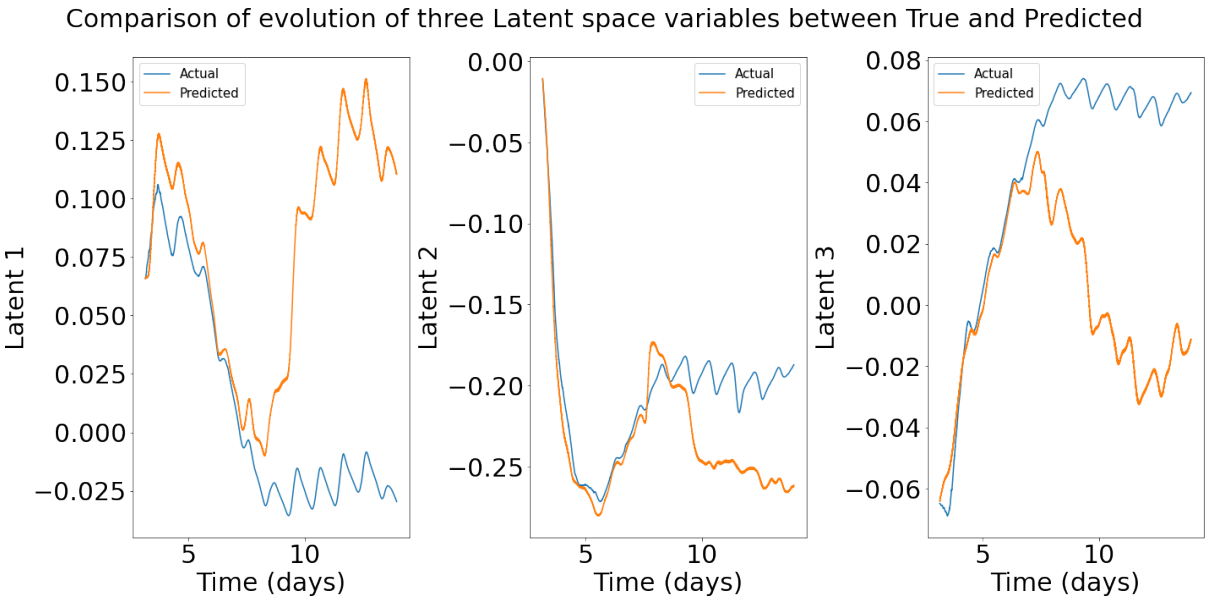

Figure 7.4. (a) Temporal Evolution of latent variables over 11 days.

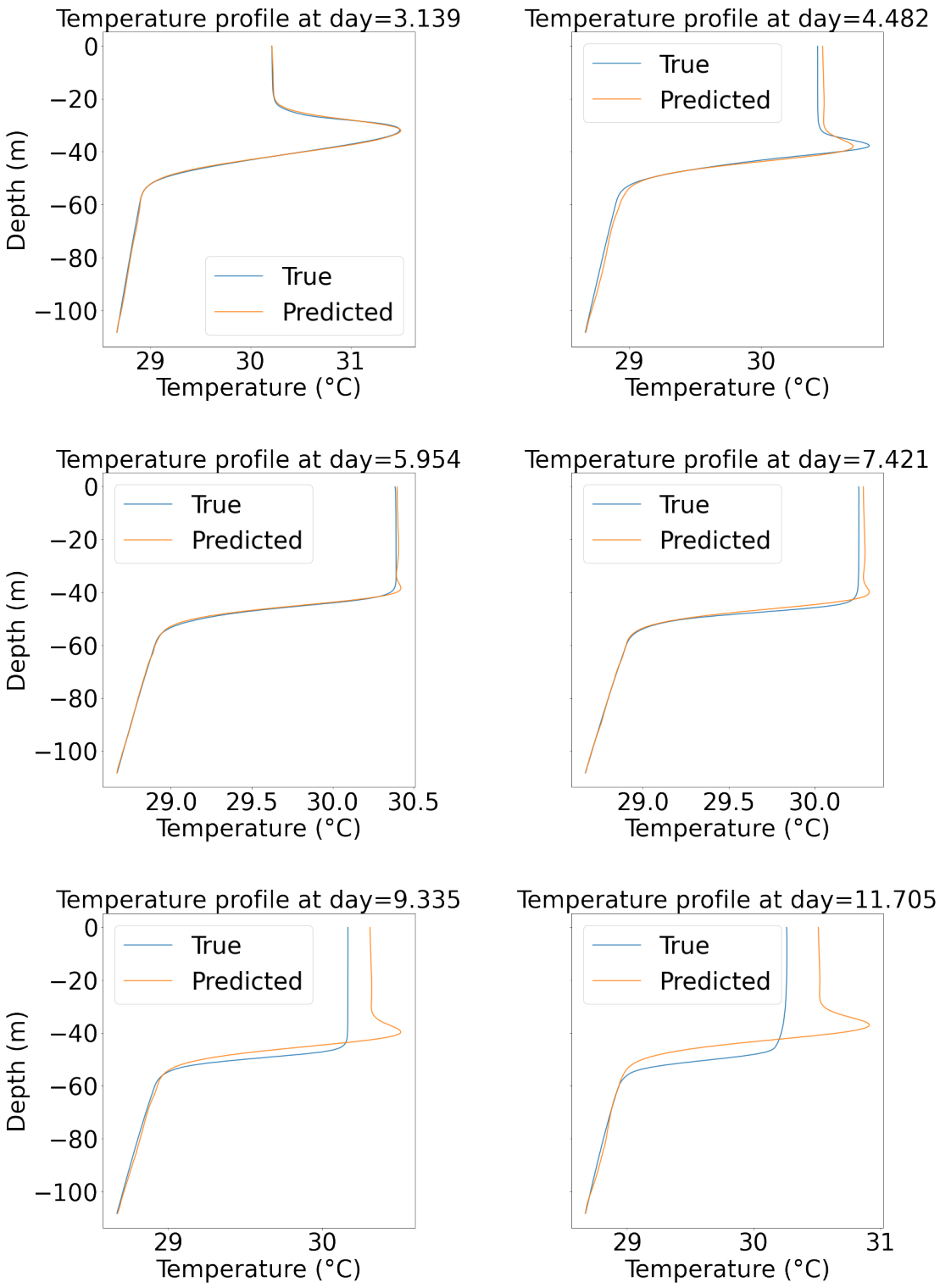

Figure 7.5. Temperature predictions at various time instances.

The profiles start developing a thermal inversion layer even though it was only present

in the active phase. This indicates that the model performs poorly when it is not trained on appropriate dynamics, leading to erroneous results. To verify this observation, the next section trains separate models for both reconstruction and temporal evolution, using the same data from the active phase, while adding the last 1000 profiles from the break phase to ensure that the model is trained with appropriate physics.

7.1.3 Enhancing long-term predictions of temperature profiles during the break phase using additional data

In the previous section, the long-term prediction capability of the model was tested, which performed poorly. In this section, to predict the profiles from the break phase, a few profiles from the break phase are given during the training to assist the model during the prediction. The dynamics of the break phase are primarily characterized by diurnal fluctuations in solar flux, constant wind speed, and zero precipitation. As a result, incorporating the necessary physics becomes crucial for achieving accurate long-term predictions. A total of 1000 profiles from the break phase comprising 2 days of data are selected. These profiles, in addition to the usual profiles from the active phase, are included in the training data. The prediction is carried over for 9 days. Therefore, a separate model is trained and the hyperparameters are tuned again to minimize the validation error. The list of hyperparameters and their values are given in the table 7.1.

Table 7.1. List of tuned hyperparameters for temperature predictions after addition of training data.

	Epochs	Hidden units	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable
LSTM-Autoencoder	300	41	1	1.00E-03	-	0.5	25	3
GRU-Temporal	250	77	1	5.00E-04	1.00E-06	0.70	28	-

The reconstruction of the temperature profile at certain time instances for the entire period starting from day 3 is shown in figure 7.2. The first five profiles in the plot correspond to the active phase, where the LSTM-autoencoder was trained. The initial three profiles represent the reconstruction from the active phase. Figure 7.6a shows the normalized error in the reconstruction

of the temperature profiles, which has now decreased from 40% to 20%. Figure 7.7 compares the reconstructed temperature profiles at various time instances with the true profiles. The reconstruction for the last few profiles exhibits less vertical shift compared to the previous case, where data from the break phase was not included. Figure 7.6b shows the reconstruction of the Sea surface temperature (SST) for 9 days. Figure 7.8 shows the reconstruction for the entire 9-day period. Overall, the reconstruction capability has improved after including additional data from the break phase.

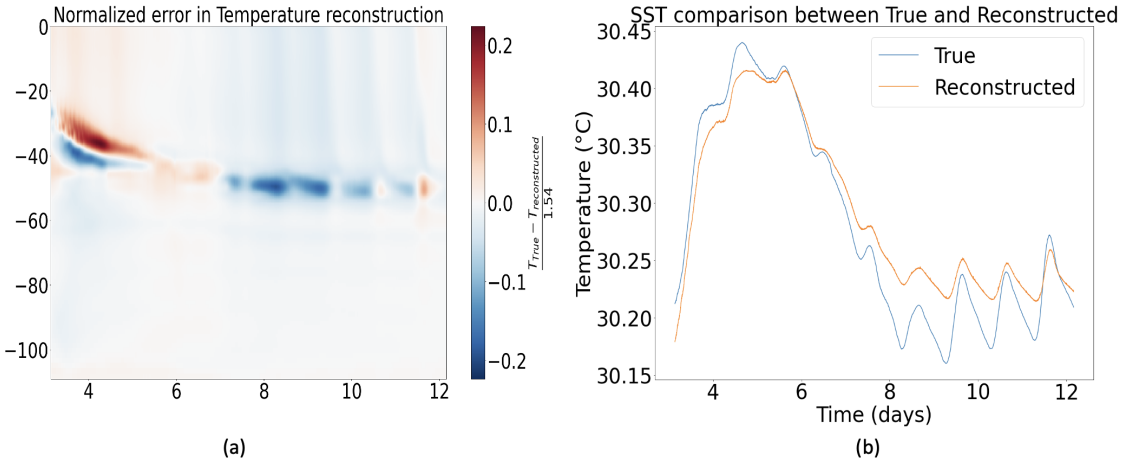

Figure 7.6. (a) Normalized reconstructed error (b) Reconstructed Sea Surface Temperature.

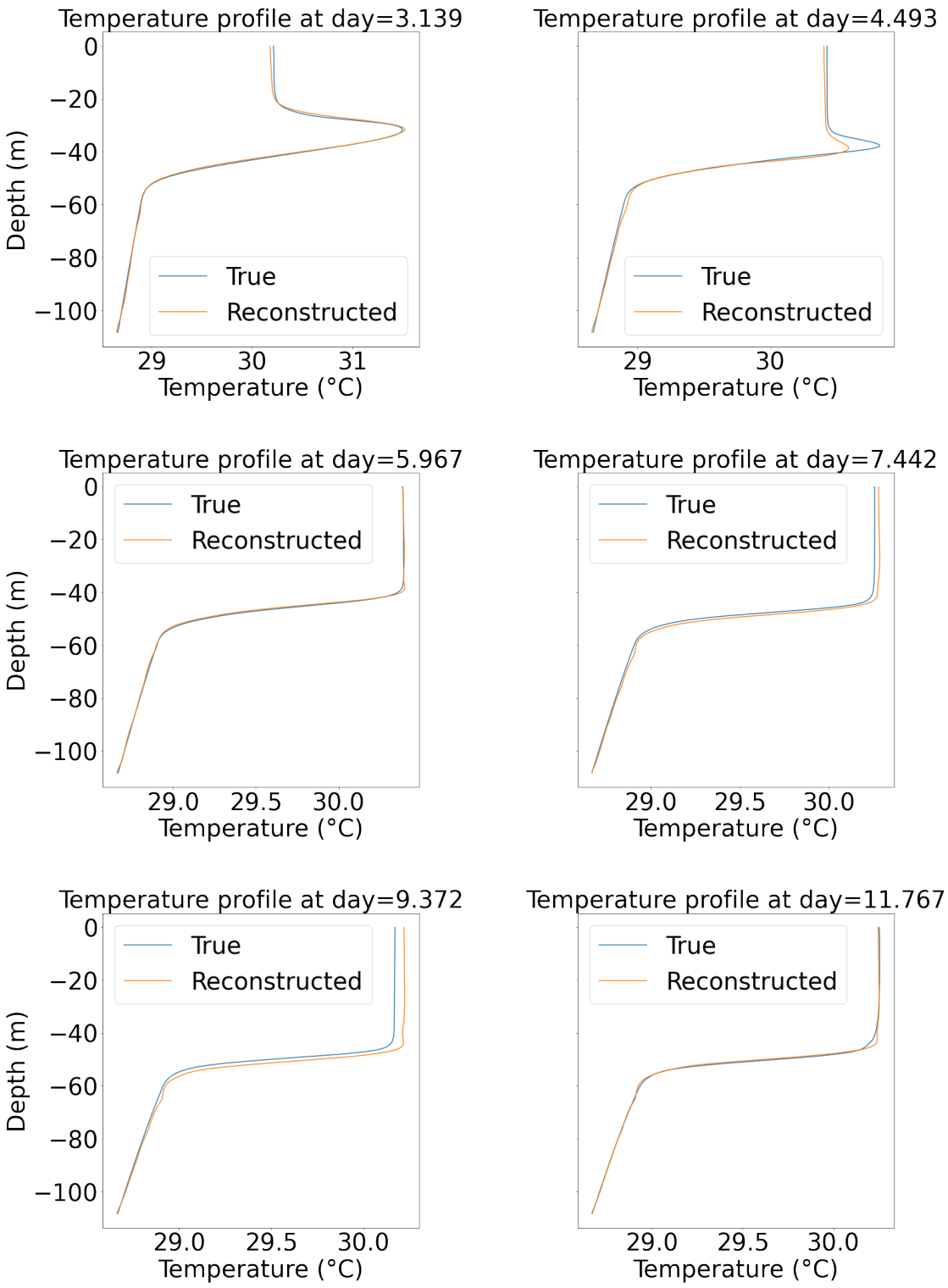

Figure 7.7. Temperature reconstruction at various time instances.

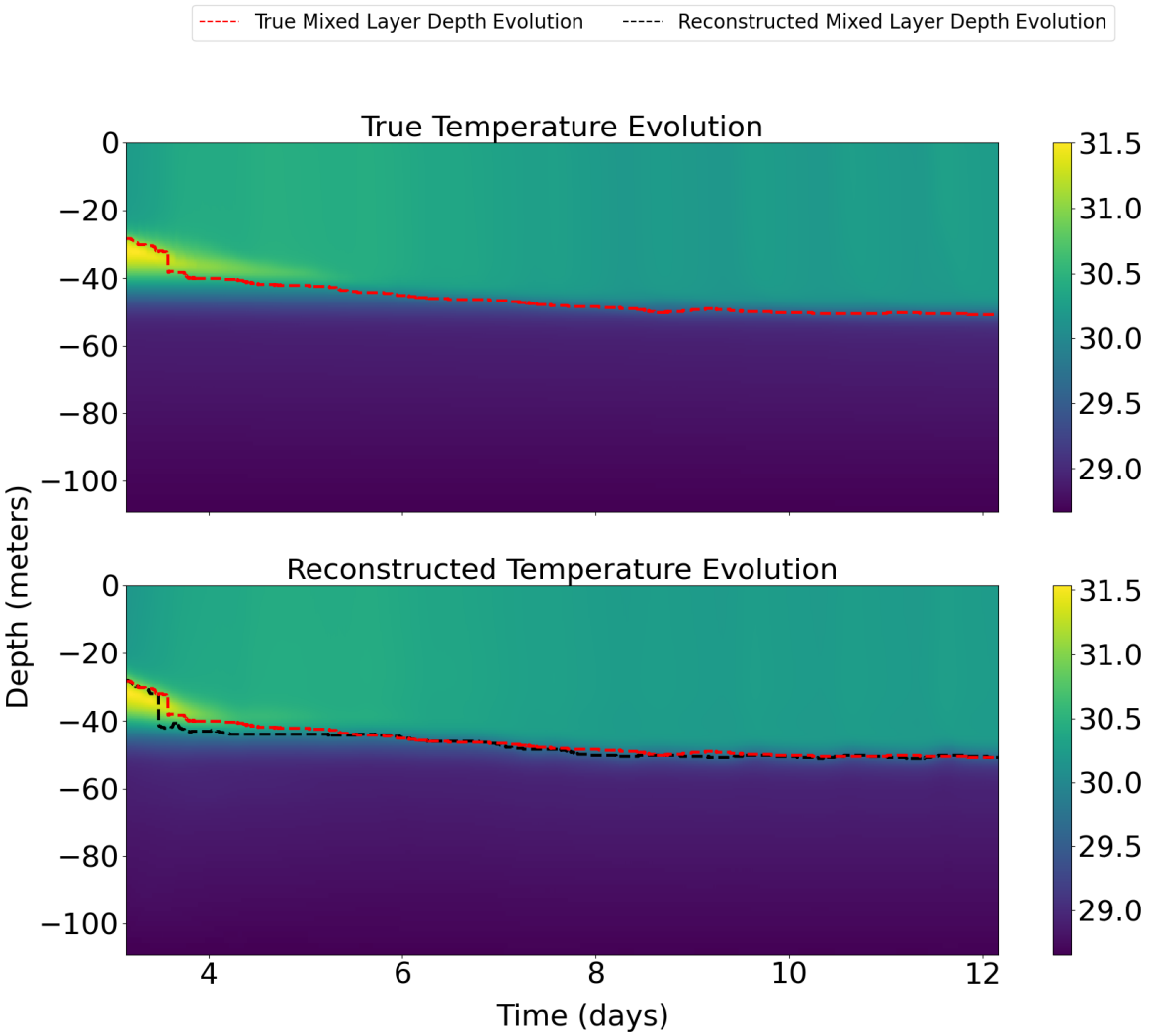

Figure 7.8. Comparison between true and reconstructed temperature over 9 days.

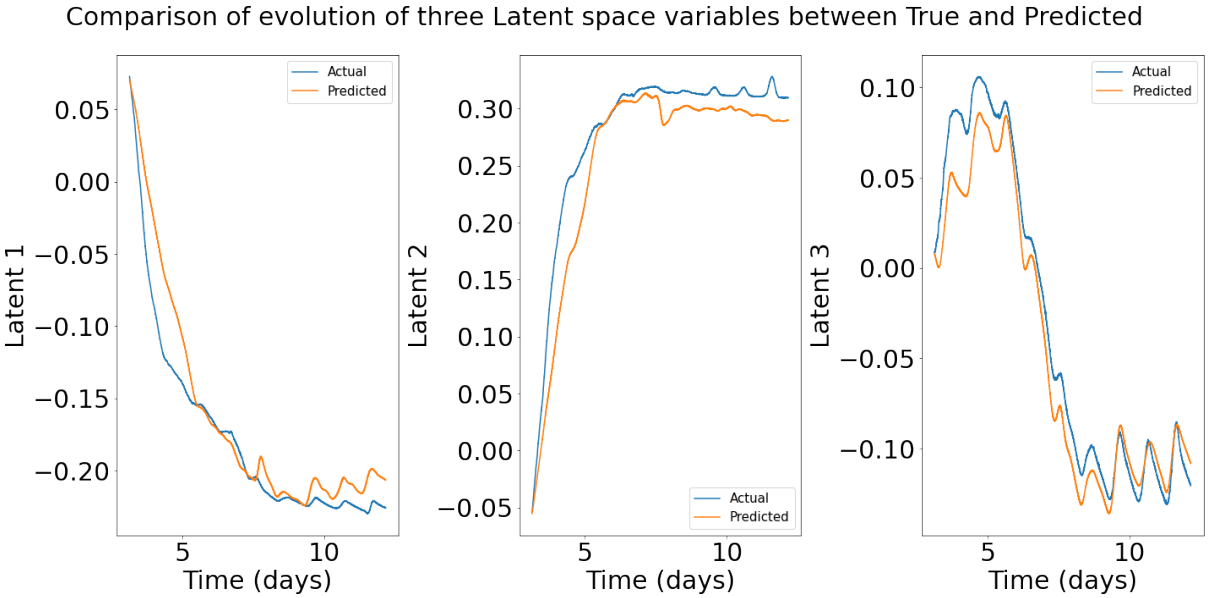

Figure 7.9. (a) Temporal evolution of latent variables over 9 days.

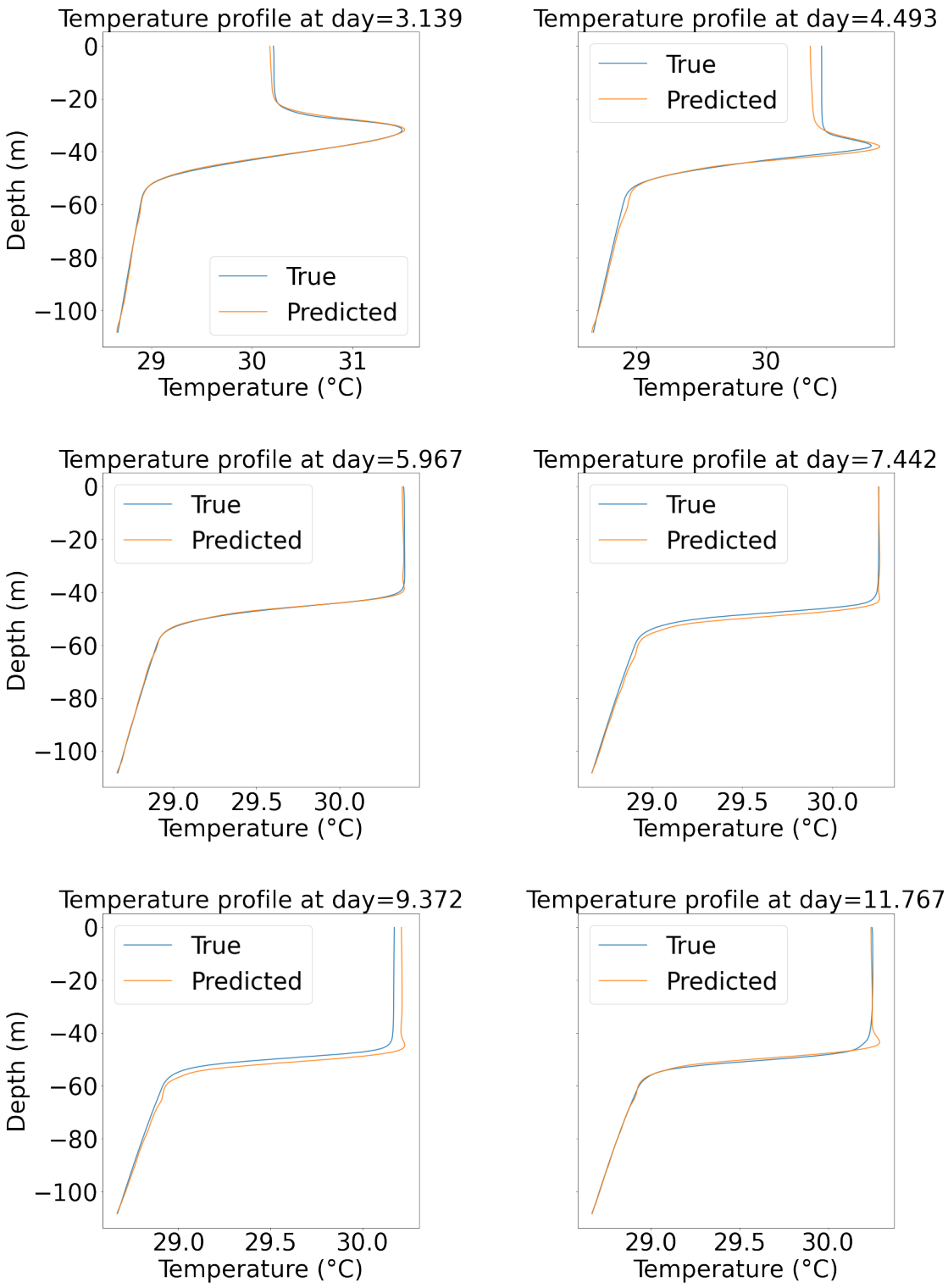

Figure 7.10. Temperature predictions at various time instances.

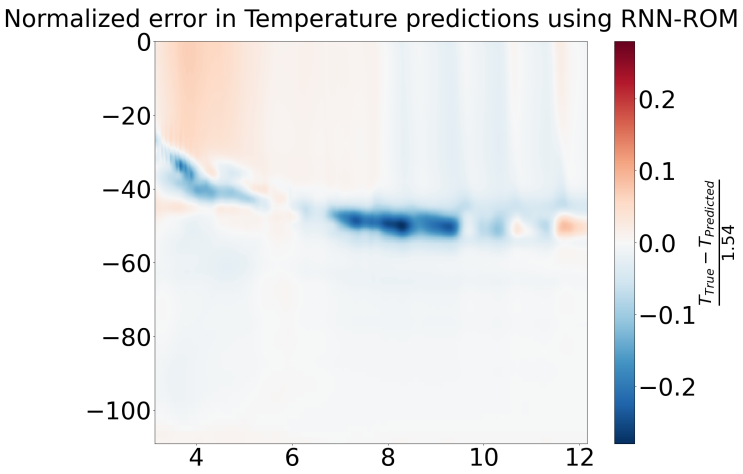

Figure 7.11. Normalized error in temperature prediction over the 9 days.

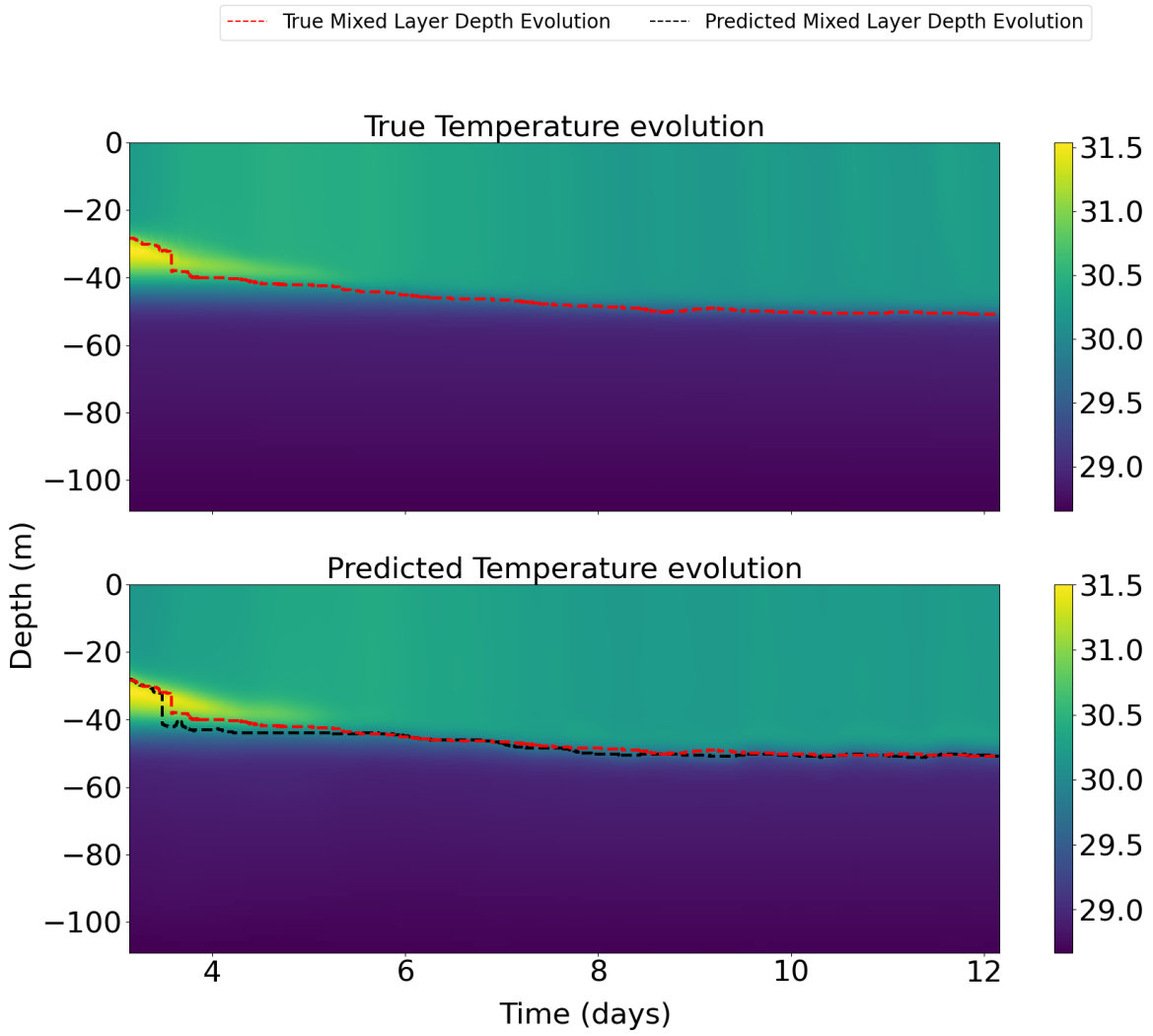

Figure 7.12. Comparison between true and predicted temperature over 9 days.

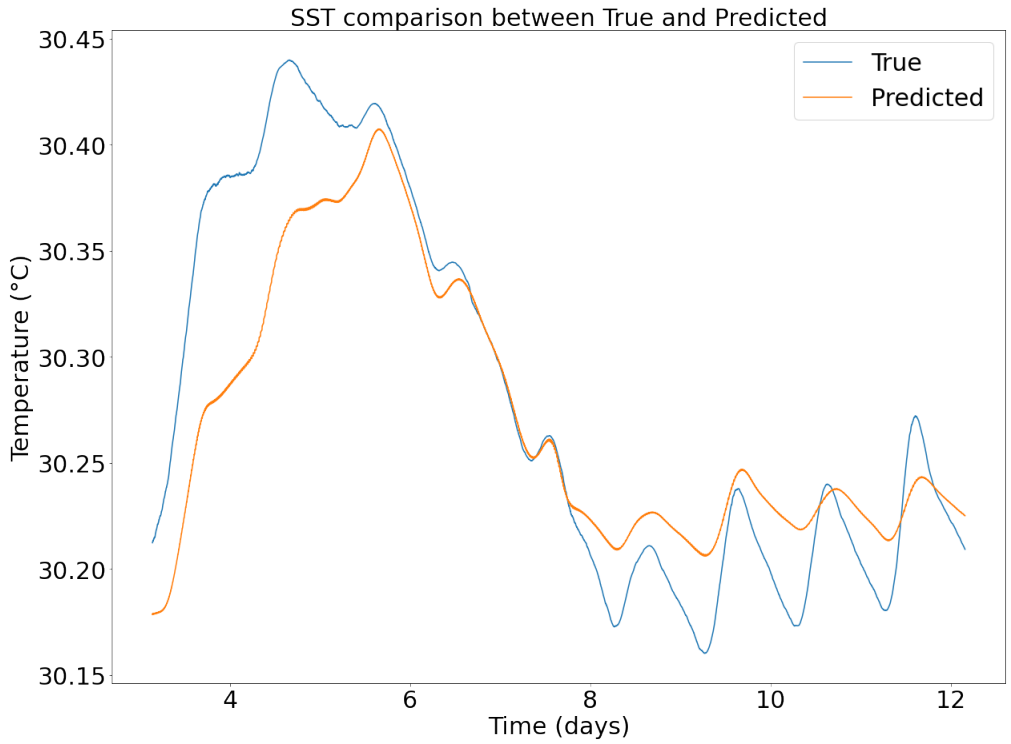

Figure 7.13. Comparison between true and predicted SST.

Next, temporal prediction for the three latent variables is conducted over a 9-day period, from day 3 to day 12. Figure 7.9 presents the predictions for three latent variables compared with the true values. This time the predictions have improved beyond the break phase, and exhibit sinusoidal fluctuation near the end of the break phase. Figure 7.10 compares the predicted and true temperature profiles at various time instances.

The final three profiles correspond to the break phase, where the model predicts a temperature in the mixed layer that is approximately 0.1 °C cooler than the true temperature. This error is also evident in the Sea surface temperature (SST), as shown in Figure 7.13, where the prediction is less accurate in the active phase with the model predicting cooler SST. However, unlike the previous case without the break phase data, where the model began to re-develop the thermal inversion layer, the prediction at the mixed layer depth remains accurate throughout the break phase. Figure 7.11 shows the normalized error between the true and the predicted for the entire prediction. The maximum error is about 20%. Figure 7.12 shows the evolution

of the temperature profile throughout the 9 days. Figure 7.28 also illustrates the evolution of the mixed layer depth (MLD). The model now accurately predicts the gradual deepening of the MLD in the break phase, which was absent when the additional data from the break phase was not provided. The maximum error in the MLD is 9.90m, which occurs at the beginning of the prediction due to an early prediction of sudden deepening of the MLD. In the later period, the prediction remains consistent with the LES. Overall, the predictions have improved upon providing additional training profiles from the break phase.

7.2 Predicting salinity profile

This section deals with the prediction capability of the RNN-ROM model for the salinity profile in the break phase. Specifically, the model trained during the active phase is tested for its predictive capacity in the future. The same model parameters used in Chapter 5, section 5.2, are applied here, with the hyperparameter values listed in Table 5.2. This section tests the extrapolative capability of the RNN-ROM model in predicting the salinity profile.

7.2.1 Reconstruction of salinity profiles

The comparison between the true and the reconstructed salinity profiles at various time instances is shown in Figure 7.15. The predicted profiles show excellent agreement with the actual profiles. The last three profiles correspond to the break phase, where the model has not been trained, and the reconstruction remains close to the true profiles. However, towards the end of the break phase, there is a slight discrepancy in the profiles at the mixed layer depth. Figure 7.14a shows the normalized error between the reconstructed and true salinity profiles, defined by, $\frac{max(S_{actual} - S_{predicted})}{\text{Normalizing constant}}$, with the maximum value of approximately 0.4, occurring at the end of the break cycle. Figure 7.16 shows the reconstruction of salinity over the entire 11-day period. From the figure, it is clear that the LSTM-autoencoder fails to reconstruct the gradual deepening of the mixed layer near the end of the break phase. Overall, the reconstruction is accurate for the first few days following the onset of the break phase.

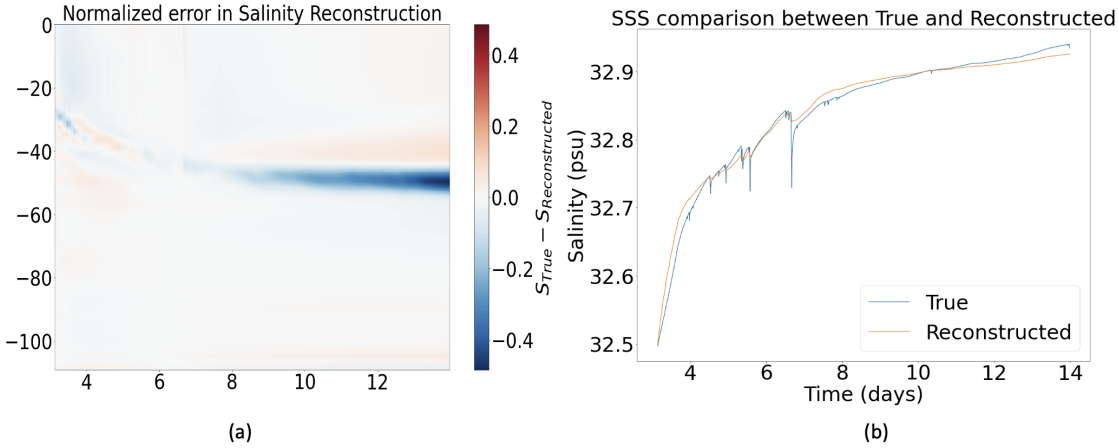

Figure 7.14. (a) Normalized reconstructed error (b) Reconstructed Sea Surface Salinity.

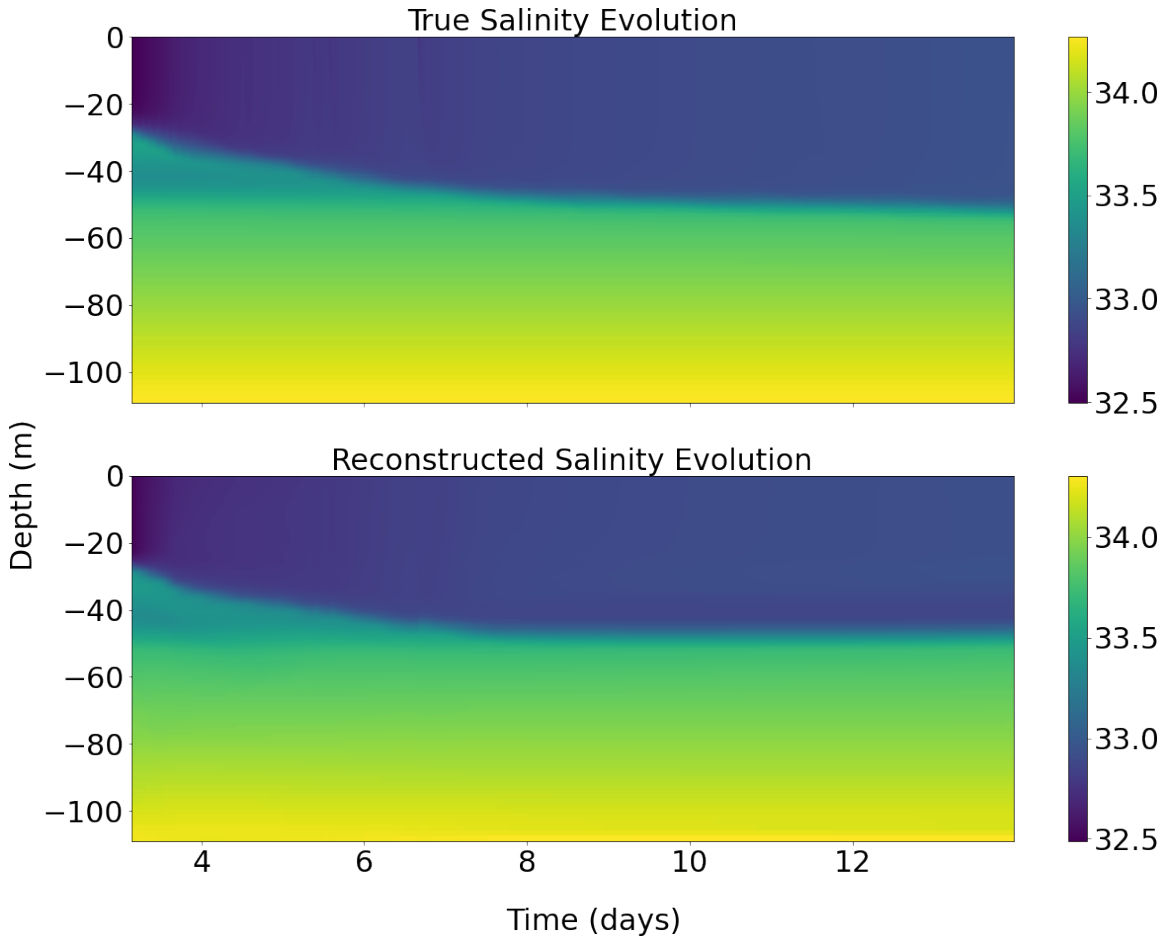

Figure 7.16. Comparison between true and reconstructed salinity for 11 days.

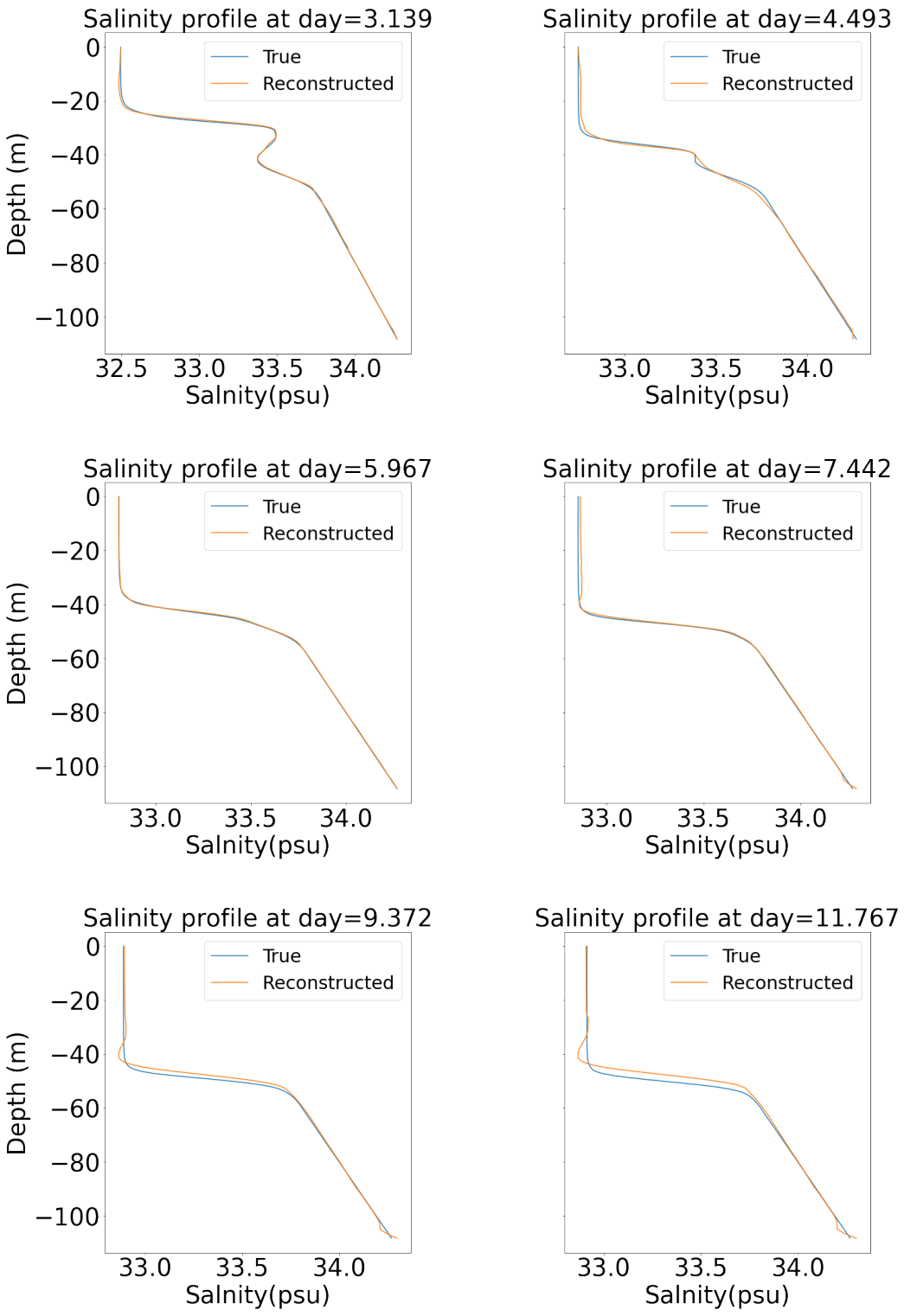

Figure 7.15. Salinity reconstruction at various time instances.

7.2.2 Temporal evolution of the latent variables and prediction of salinity profiles

The temporal predictions of the three latent variables during part of the active phase and break phase are shown in Figure 7.17. The figure indicates that beyond the active phase, the predictions remain constant, suggesting that the RNN-ROM model fails to capture the variation of the latent variables throughout the entire break phase. This can also be seen from the figure 7.18, where the maximum error reaches 0.42. Figure 7.20 shows the predictions of salinity profiles at various time instances.

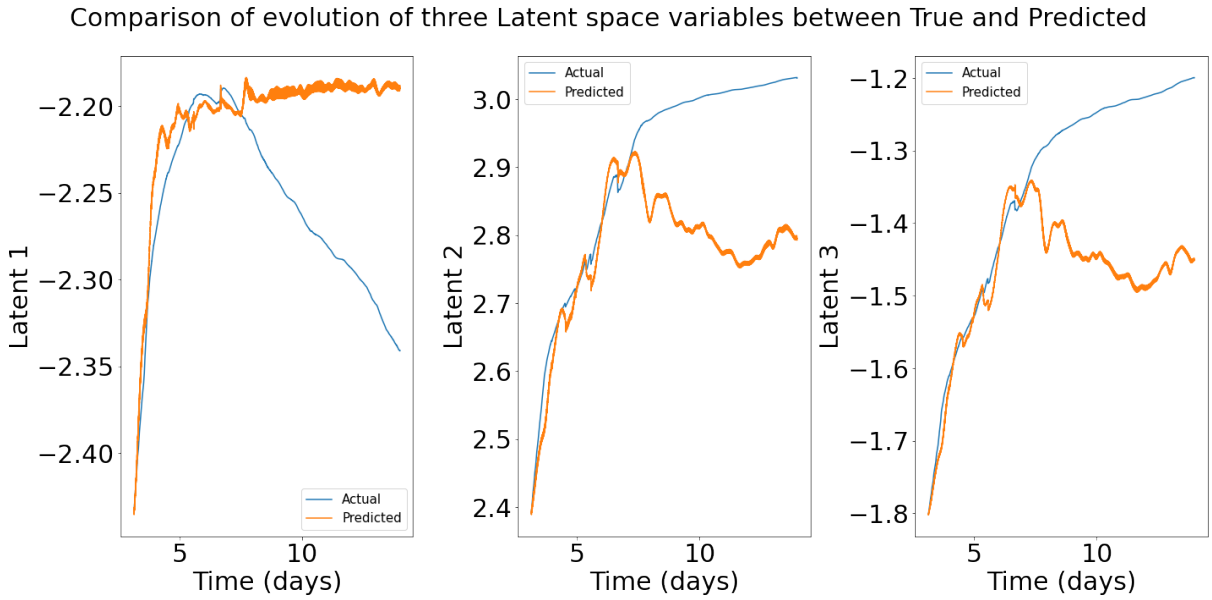

Figure 7.17. (a) Temporal evolution of latent variables over 11 days.

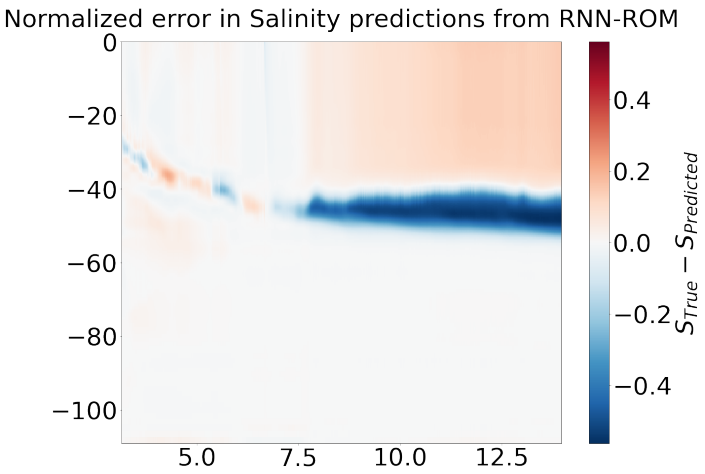

Figure 7.18. (a) Error in salinity prediction .

The salinity in the mixed layer remains the same throughout the break phase, while the actual values increase over time. Figure 7.19 shows the comparison between the true and predicted Sea surface salinity (SSS), where the predicted net SSS remains constant. Figure 7.21 shows the evolution of the salinity over the entire 11-day period. From the figure, we can observe the initial deepening of the mixed layer, until the end of the active phase. However, after the onset of the break phase, the model incorrectly predicts the evolution of mixed layer depth, where it predicts shallower mixed layer depth during the break phase.

This shows that the RNN-ROM model fails to capture dynamics it was not trained on. To enhance the model's predictive capacity, a separate model is trained using the last 1,000 profiles comprising 2 days of data from the break phase, ensuring that the model is trained with the appropriate physics. The detailed results are presented in the next subsection.

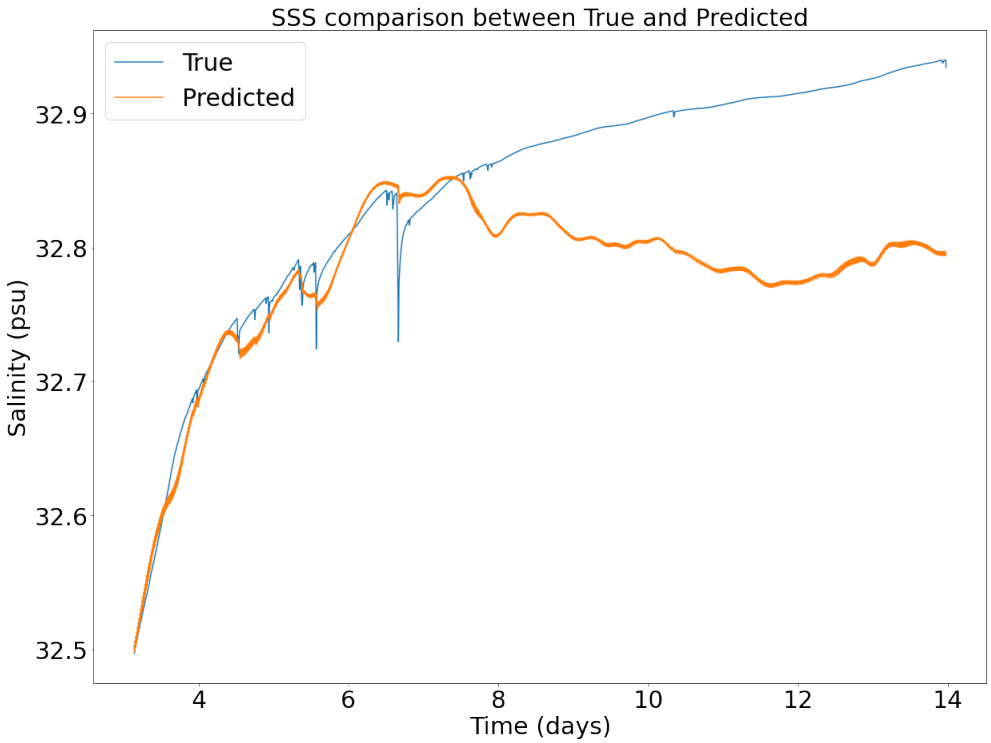

Figure 7.19. Comparison between true and predicted SSS.

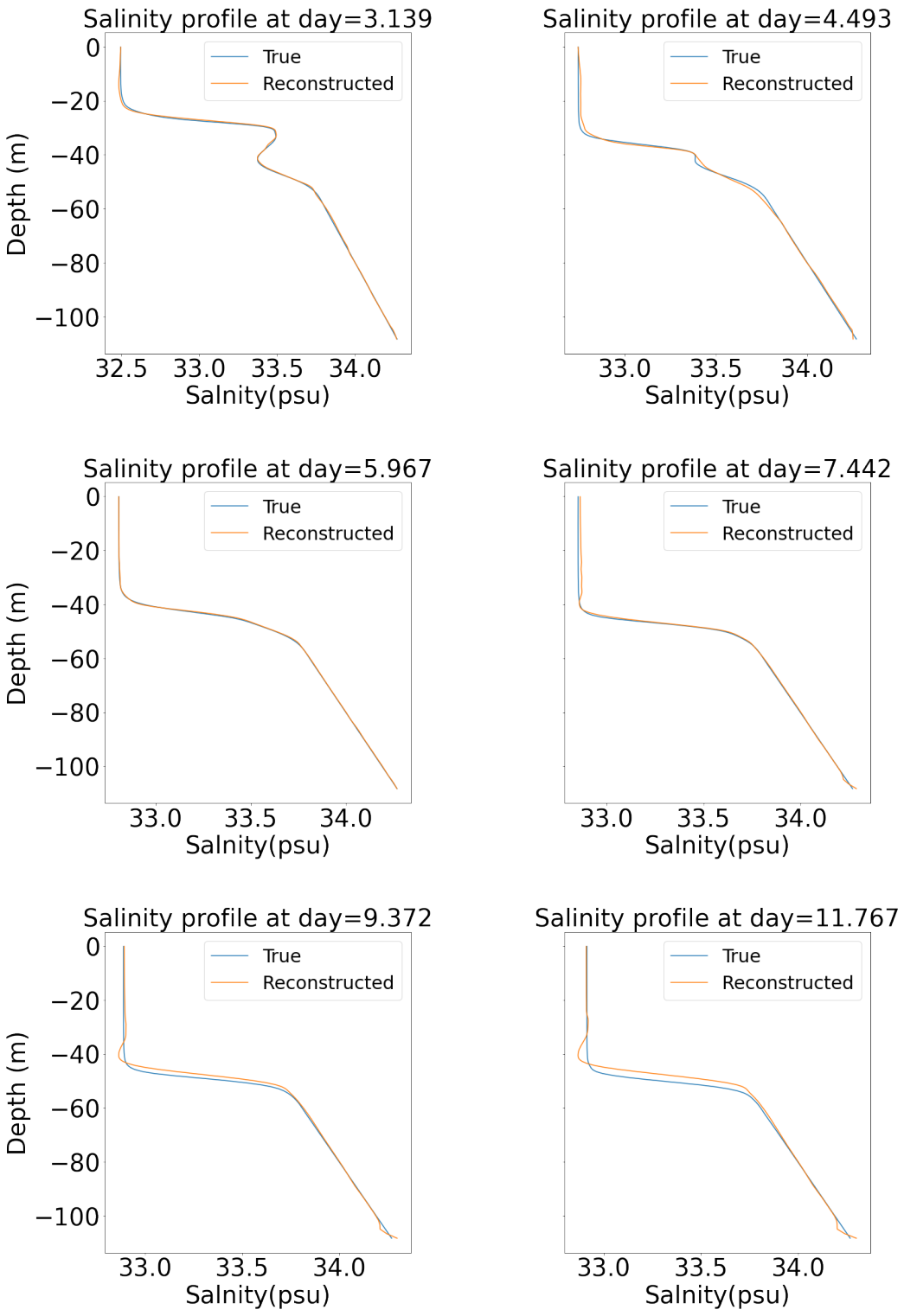

Figure 7.20. Salinity predictions at various time instances.

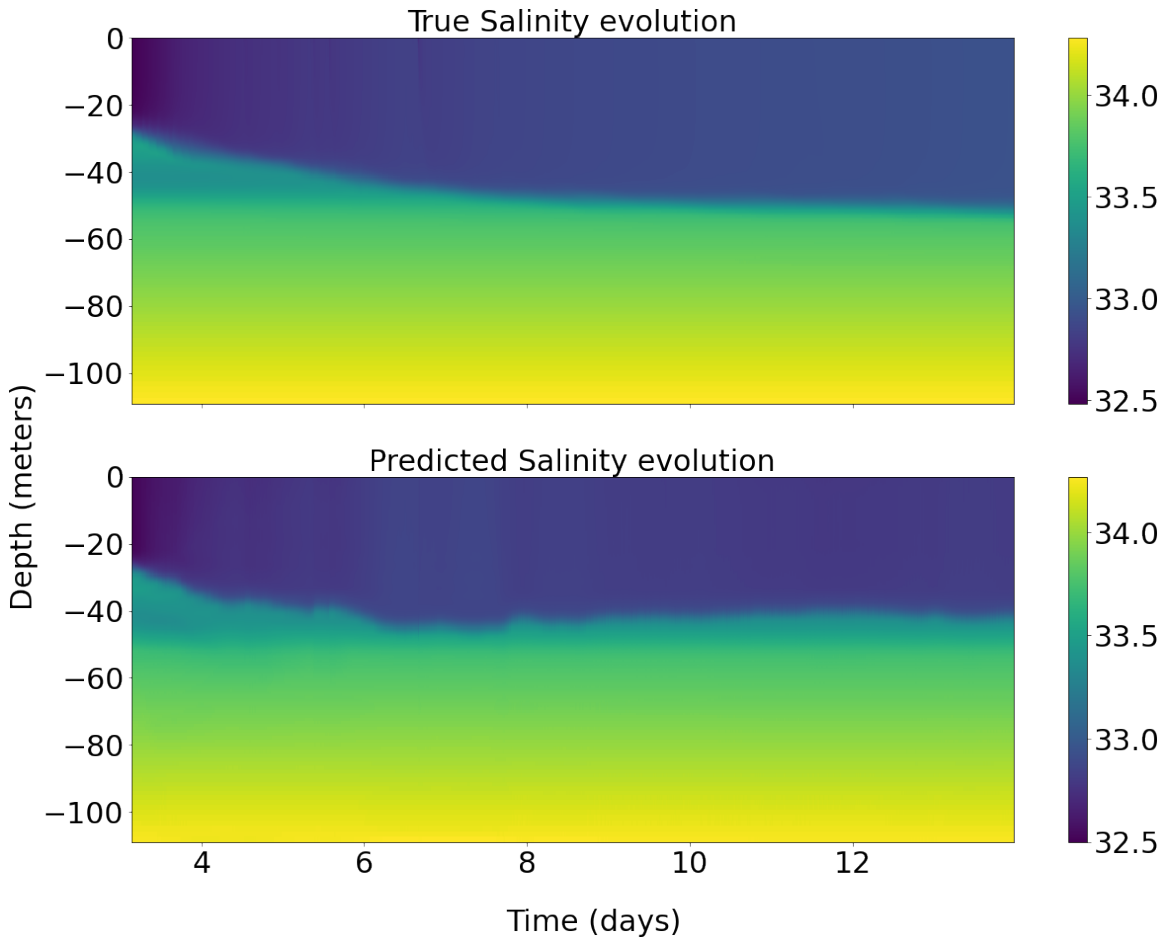

Figure 7.21. Comparison between true and predicted Salinity for 11 days.

7.2.3 Enhancing long-term predictions of salinity profiles during the break phase using additional data

In the previous section, it was observed that the RNN-ROM failed to predict the break phase because the model was not trained on the appropriate physics. In this section, the model is assisted with additional data from the break phase, specifically from the end of the break phase. This ensures that the model has an understanding of the appropriate physics before being deployed to predict similar physics. An additional 1000 profiles corresponding to 2 days from the break phase are added to the existing training set from the break phase to improve the prediction process. Since the last 1,000 profiles were used for training, the prediction is limited to day 9. Separate training was performed and the hyperparameters were tuned to minimize the validation

loss.

The hyperparameters for the current setting are shown in the table 7.2.

Table 7.2. List of tuned hyperparameters for salinity predictions after addition of training data.

	Epochs	Hidden units	Hidden layers	Initial lr	β	lr decay	Decay applied	Latent variable
LSTM-Autoencoder	300	70	1	6.00E-04	1.00E-07	0.7	24	3
GRU-Temporal	250	77	1	4.00E-04	1.00E-07	0.80	20	-

Figure 7.23 shows the comparison between the reconstructed and true salinity profiles at various time instances. The reconstruction from the LSTM autoencoder agrees well with the true profiles. This is further illustrated in Figure 7.22a, which depicts the error between the true and reconstructed salinity profiles. The maximum normalized error, defined by $\frac{max(E_{actual}-E_{predicted})}{\text{Normalizing constant}}$, where the normalizing constant is the average variation of salinity at the mixed layer depth (MLD) over 9 days, is 0.15. This represents a 73% reduction compared to the case where the break phase data was not provided. Figure 7.22b shows the reconstruction of the Sea surface salinity (SSS) profile over the entire 9-day period. The reconstruction is nearly perfect, except that it does not capture the influx of fresh water on certain days. However, the model accurately reconstructs the increase in the SSS over time. Figure 7.24 shows the reconstruction of the salinity profiles over the entire 9-day period. The model accurately reconstructs the deepening of the mixed layer depth over time. Overall, the reconstructive capability of the LSTM-autoencoder has significantly improved with the addition of the training data from the break phase.

Next, we take the reduced representation or the latent variables and train them to evolve them in time. Figure 7.25 depicts the comparison between the true and predicted evolution of three latent variables over a 9-day period. The prediction agrees well with the true evolution of the first 2 latent variables. For the third latent variable, the model captures the overall trend in the evolution but exhibits significant fluctuation and discrepancy towards the end of the break phase. Nevertheless, the model successfully predicts the overall trend of all three latent variables, which is an improvement over the previous case where predictions showed constant values during the

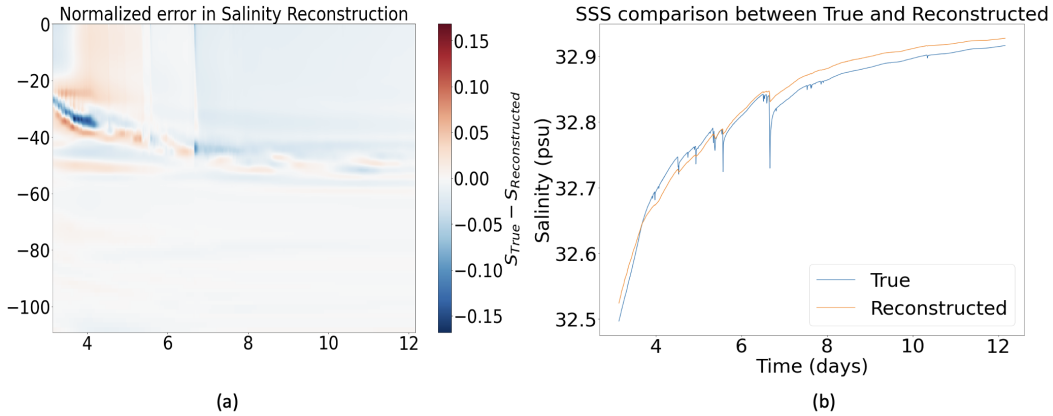

Figure 7.22. (a) Normalized reconstructed Error (b) Reconstructed Sea Surface Salinity.

break phase due to the lack of additional data. Figure 7.27 shows the comparison between true and predicted salinity profiles at various time instances. The last three profiles correspond to the break phase. The predictions are accurate in the active phase, as expected since the profiles are primarily trained in this phase. In the break phase, the predicted profiles also align excellently with the true profiles. However, the predicted salinity at the mixed layer is 0.03 psu higher than the true values, as observed in several profiles during the break phase in Figure 7.27. Figure 7.26 shows the normalized error in the predicted and true salinity profiles. The maximum error is 0.3, which is 40% lower than the previous case without additional training data. Figure 7.28 shows the comparison of the evolution of salinity profiles between the true and predicted values over a 9-day period. Figure 7.29 shows the comparison of the evolution of Sea surface salinity (SSS) between true and prediction. The model predicts a slightly higher value of SSS, about 0.03 psu in the break phase. Also, the model does not predict the influx of fresh water on certain days. Because there are not many instances in the training data with the influx of fresh water, the model might have overlooked the phenomena. Figure 7.28 also shows the evolution of the mixed layer depth (MLD). The MLD evolution agrees well with the ground truth (LES). The maximum error in the MLD is 2.41m which occurs at the onset of rapid deepening. In the later period, the MLD prediction remains consistent with the LES. Therefore, we can conclude that incorporating additional data has significantly improved the model's predictive capacity for salinity profiles

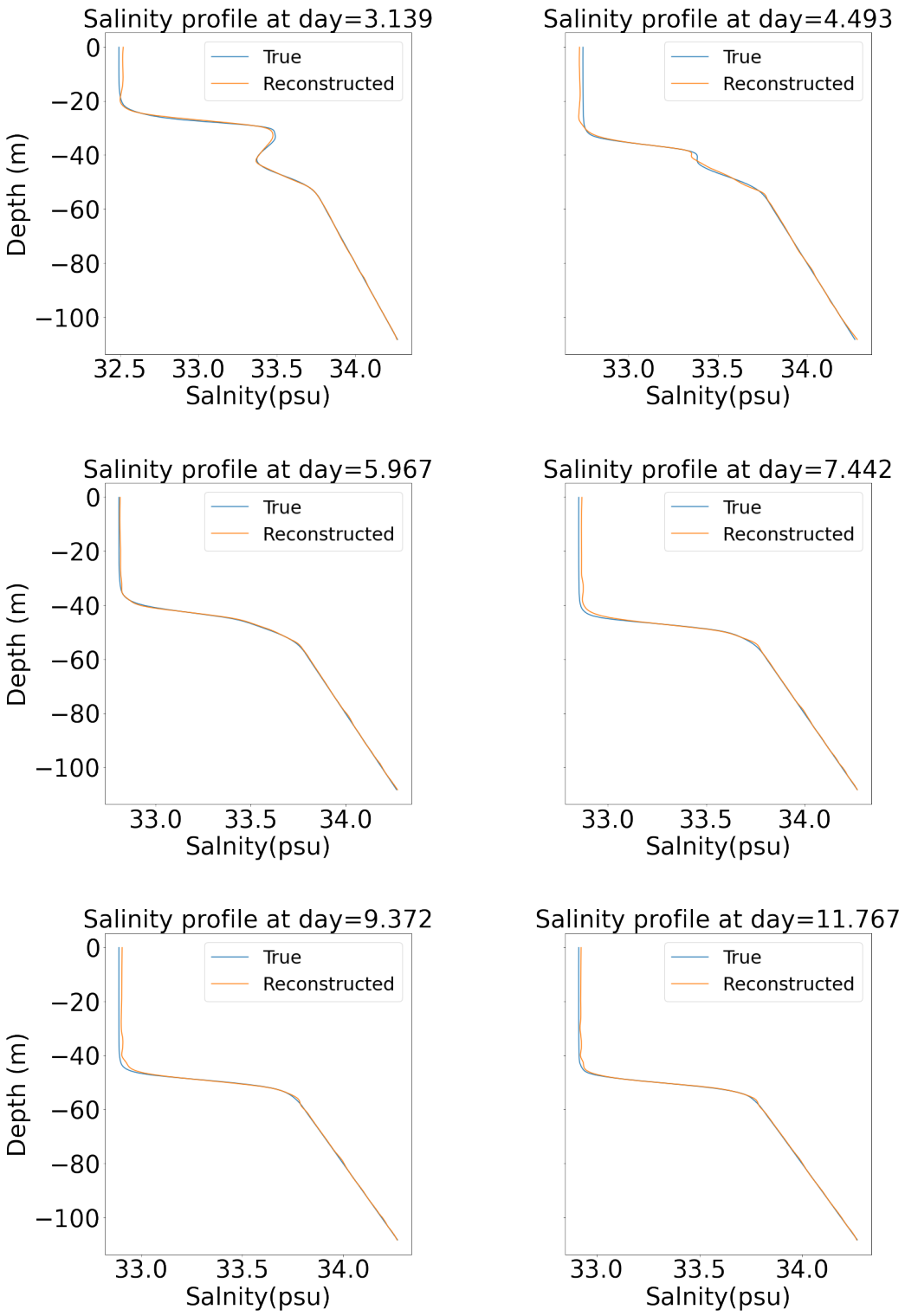

Figure 7.23. Salinity reconstruction at various time instances.

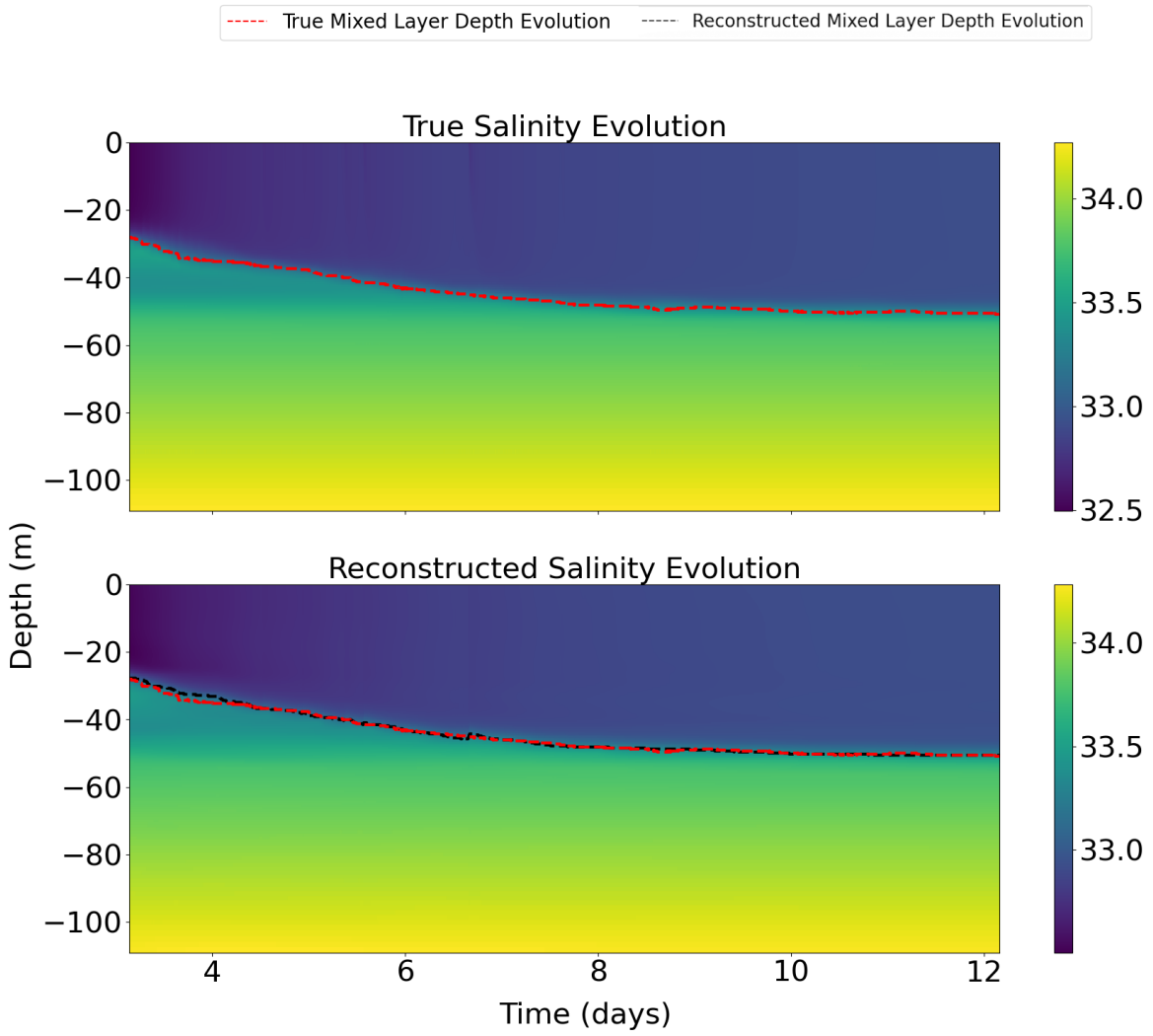

Figure 7.24. Comparison between true and reconstructed salinity for 9 days.

during the break phase cycle.

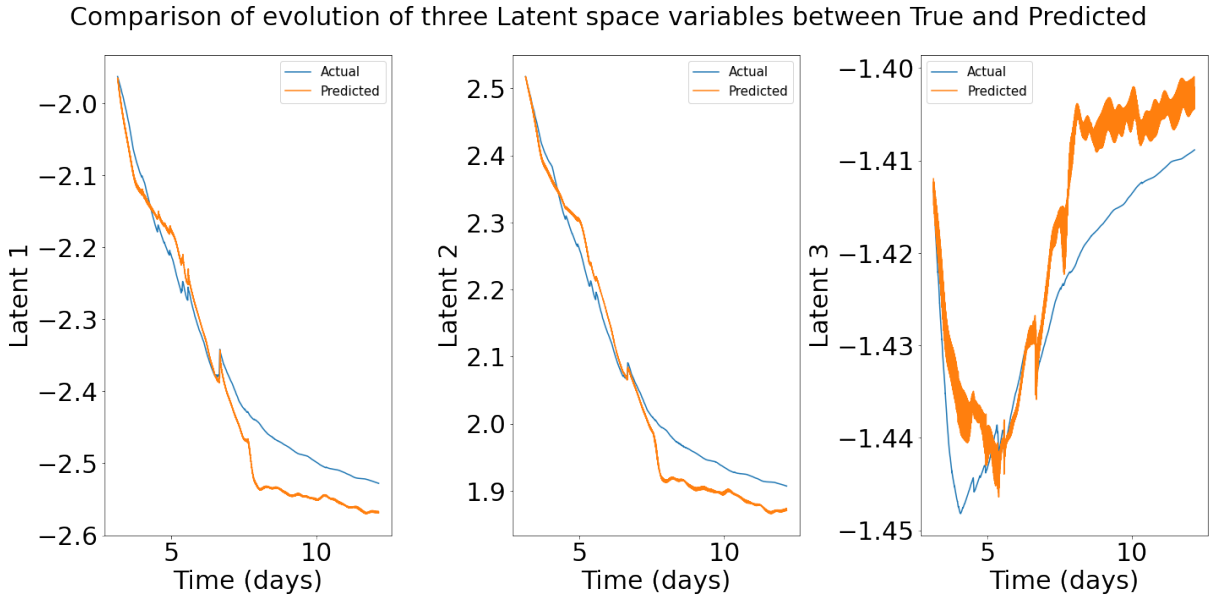

Figure 7.25. (a) Temporal evolution of latent variables over 9 days.

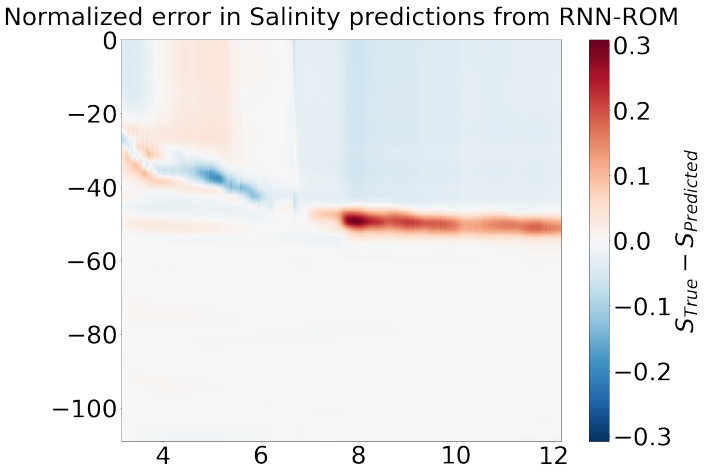

Figure 7.26. (a) Error in salinity prediction .

Comparison with Large Eddy Simulation

The table shows the time required to train each of the quantities. The LES time includes the calculation of both temperature and salinity and other quantities associated with the equations. Therefore total time required to predict the temperature and salinity over a 9-day period for RNN-ROM is 770.96 seconds. Considering the time required to tune the hyperparameters,

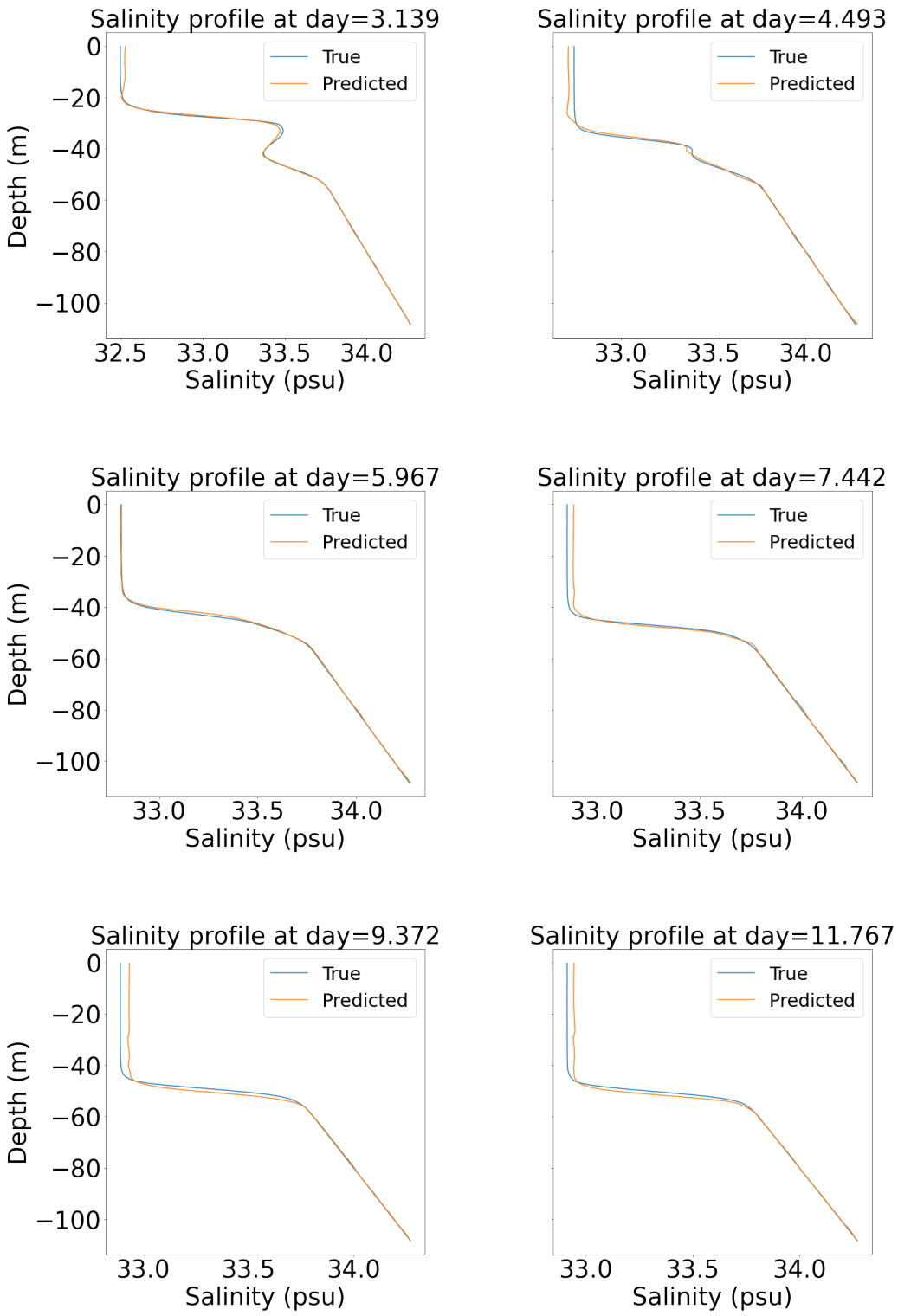

Figure 7.27. Salinity predictions at various time instances.

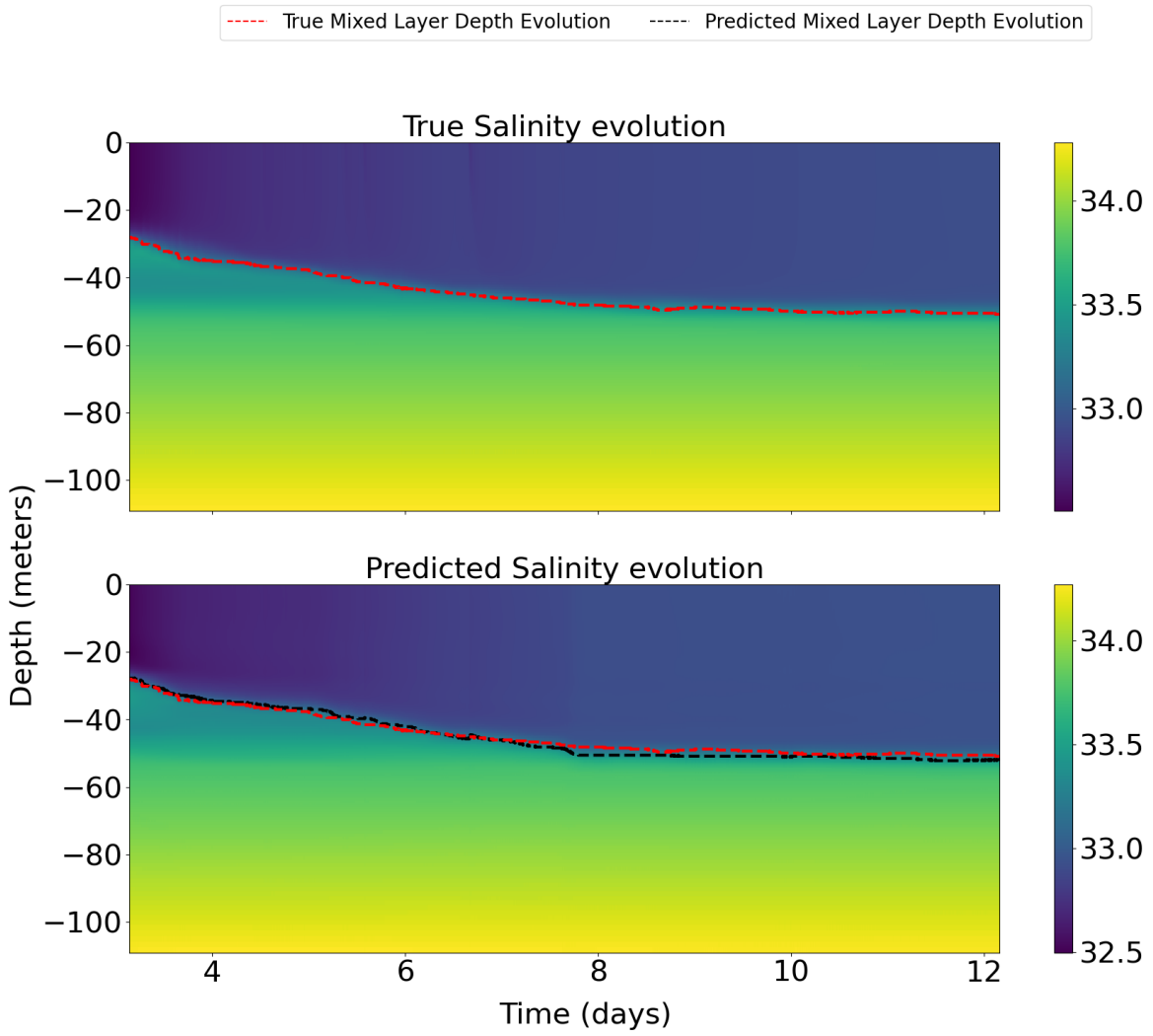

Figure 7.28. Comparison between true and predicted salinity for 9 days.

the total time will be around 10000[2] seconds, which is 662 times faster than the Large Eddy Simulation.

[2]This excludes the time required to acquire the data

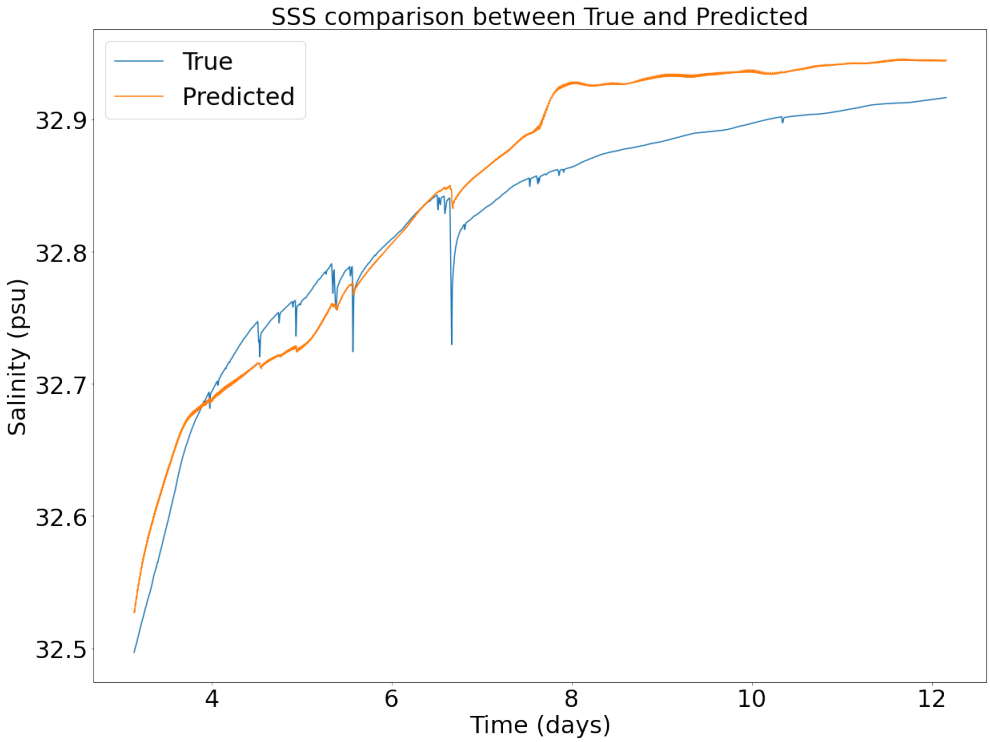

Figure 7.29. Comaprison between true and predicted SSS.

Table 7.3. Time Required by Large Eddy Simulation and RNN-ROM for 9-Day predictions of temperature and salinity

	LSTM-Autoencoder(s)	GRU+FC(s)	Total time(s)
Salinity	251.76	132.3	384.06
Temperature	255.47	131.43	386.9
LES	-	-	662688

Chapter 8

Conclusion

In this work, an RNN-based Reduced Order Model (RNN-ROM) was developed to predict the evolution of temperature and salinity profiles. The model consisted of two independently trained networks: (1) an LSTM-autoencoder, which reduced the dimensionality of the input profiles, and (2) a Gated recurrent unit (GRU) network, which mapped the previous ten states to the next ten states, with the function also parameterized by surface forces from the corresponding time step. The predictions were generated autoregressively, with the GRU's output being fed back as input to continue forecasting until the final time step. Four cases were analyzed: the first two examined the effect of varying the number of training samples on predicting the rapid evolution of temperature and salinity fields between days 3 and 5, the third case focused on predicting the future of the temperature and salinity profile in the break phase and the fourth case evaluated the importance of incorporating the necessary physics to achieve long-term evolution.

Case 1 was trained using the data from both the active and break phases, while Case 2 was trained using only the active phase data, and comparisons were drawn between them. To be more specific, different metrics like mean absolute percentage error (MAPE), maximum error, and total error were used for evaluating LSTM-autoencoder, and metrics like R2-score and mean squared error for evaluating GRU, to compare two cases along with the train time. MAPE for SST decreased from 0.00084 to 0.0003 and for SSS decreased from 0.0008 to 0.0006. Maximum error remained the same for temperature, while it increased from 0.1426 to 0.185, and total

error remained the same for temperature while it had increased slightly from 12.79 to 13.54 for salinity. The overall performance considering the metrics, from Case 1 to Case 2, has not significantly improved. However, the train time significantly improved, reducing it by more than half. The average R^2 score for temperature decreased from 0.607 in Case 1 to -0.344 in Case 2. This significant reduction was primarily due to the highly negative R^2 score for the first latent variable, which only affected the mixed layer profile, causing it to shift to the right, predicting warmer temperatures in the mixed layer. The average R^2 score for salinity increased from 0.63 to 0.866, which concludes that GRU has improved the prediction for salinity. However, a significant improvement was seen in the training speed when trained only on active phase data, which was reduced by factor two. The overall speed reduction considering both the stages was four orders of magnitude compared to training with data from active and break phases. The two cases were also compared with Large eddy simulation in terms of speed. The model also performed better than the 1d-GOTM models which showed maximum discrepancy during rapid evolution. The overall time required to predict the two-day temperature and salinity profiles was reduced by a factor of 25 when compared to LES, resulting in significant savings in both computational and setup costs.

Case 3 tested the ability of the RNN-ROM model to predict future instances of salinity and temperature in the break phase. Reconstruction from LSTM-autoencoder for both temperature and salinity agreed well with the true data, with maximum error in the reconstruction of temperature being 21% of the total change in the temperature profile at the mixed layer depth, and for salinity being 4% of the total change of salinity at the mixed layer over three days. The error was attributed to the high gradient of the salinity at the mixed layer which amplified even small shifts in the profile. The temporal evolution of the latent variable agreed well with the true evolution, with the maximum error being 5% for temperature, which corresponded to latent variable 1, and less than 1% for salinity evolutions. This region exhibited minimal variation in both temperature and salinity profiles, making it easier for the RNN-ROM model to predict. It is important to note that this prediction was not an extrapolation, as the input for the entire

prediction remained within the training set.

Case 4 tested the long-term prediction capabilities of the RNN-ROM model, where the profiles from the break phase and part of the active phase were predicted only from the data of the active phase. The LSTM autoencoder failed to reconstruct both profiles, especially during most parts of the break phase, failing to capture the gradual deepening of the mixed layer depth for both temperature and salinity. Similarly, the GRU's predictions for the latent variables in this phase were also inaccurate for both variables. To address this, additional data from the end of the break phase were incorporated into the training set. This adjustment ensured that the model could learn the relevant dynamics, which are essential for achieving accurate long-term predictions. With this modification, the predictions improved significantly, as the model captured the deepening of the mixed layer depth over time, and approximately captured the correct trends in both SST and SSS. Consequently, it can be inferred that temperature and salinity values can be predicted over very long periods (years or decades) if the model is provided with sufficient data and learns the correct dynamics during the training stage.

Bibliography

Nadine Aubry, Philip Holmes, John L Lumley, and Emily Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of fluid Mechanics*, 192: 115–173, 1988.

John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for numerical methods in engineering*, 86(2):155–181, 2011.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

MA Donelan, Brian K Haus, Nicolas Reul, WJ Plant, M Stiassnie, HC Graber, Otis B Brown, and ES Saltzman. On the limiting aerodynamic roughness of the ocean in very strong winds. *Geophysical Research Letters*, 31(18), 2004.

Nicholas Geneva and Nicholas Zabaras. Transformers for modeling physical systems. *Neural Networks*, 146:272–289, 2022.

FJ Gonzalez and M Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. arxiv 2018. *arXiv preprint arXiv:1808.01346*, 1808.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*

arXiv:1312.6114, 2013.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807: 155–166, 2016.

John Leask Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, pages 166–178, 1967.

Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, 33(3), 2021.

James C McWilliams, Edward Huckle, and Alexander F Shchepetkin. Buoyancy effects in a stratified ekman layer. *Journal of Physical Oceanography*, 39(10):2581–2599, 2009.

Shaoxing Mo, Yinhao Zhu, Nicholas Zabaras, Xiaoqing Shi, and Jichun Wu. Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. *Water Resources Research*, 55(1):703–728, 2019.

Arvind Mohan, Don Daniel, Michael Chertkov, and Daniel Livescu. Compressed convolutional lstm: An efficient deep learning framework to model high fidelity 3d turbulence. *arXiv preprint arXiv:1903.00033*, 2019.

Arvind T Mohan, Dima Tretiak, Misha Chertkov, and Daniel Livescu. Spatio-temporal deep learning models of 3d turbulence with physics informed diagnostics. *Journal of Turbulence*, 21(9-10):484–524, 2020.

Nikolaj T Mücke, Sander M Bohté, and Cornelis W Oosterlee. Reduced order modeling for parameterized time-dependent pdes using spatially and memory aware deep learning. *Journal of Computational Science*, 53:101408, 2021.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.

Hieu T Pham, Sutanu Sarkar, Leah Johnson, Baylor Fox-Kemper, Peter P Sullivan, and Qing Li. Multi-scale temporal variability of turbulent mixing during a monsoon intra-seasonal oscillation in the bay of bengal: An les study. *Journal of Geophysical Research: Oceans*, 128 (1):e2022JC018959, 2023.

Wolfgang Rodi. Examples of calculation methods for flow and mixing in stratified fluids. *Journal of Geophysical Research: Oceans*, 92(C5):5305–5328, 1987.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Emily Shroyer, Amit Tandon, Debasis Sengupta, Harindra J. S. Fernando, Andrew J. Lucas, J. Thomas Farrar, Rajib Chattopadhyay, Simon de Szoeke, Maria Flatau, Adam Rydbeck, Hemantha Wijesekera, Michael McPhaden, Hyodae Seo, Aneesh Subramanian, R. Venkatesan, Jossia Joseph, S. Ramsundaram, Arnold L. Gordon, Shannon M. Bohman, Jaynise Pérez, Iury T. Simoes-Sousa, Steven R. Jayne, Robert E. Todd, G. S. Bhat, Matthias Lankhorst, Tamara Schlosser, Katherine Adams, S. U. P. Jinadasa, Manikandan Mathur, M. Mohapatra, E. Pattabhi Rama Rao, A. K. Sahai, Rashmi Sharma, Craig Lee, Luc Rainville, Deepak Cherian, Kerstin Cullen, Luca R. Centurioni, Verena Horman, Jennifer MacKinnon, Uwe Send, Arachaporn Anutaliya, Amy Waterhouse, Garrett S. Black, Jeremy A. Dehart, Kaitlyn M. Woods, Edward Creegan, Gad Levy, Lakshmi H. Kantha, and Bulusu Subrahmanyam. Bay of bengal intraseasonal oscillations and the 2018 monsoon onset. *Bulletin of the American Meteorological Society*, 102(10):E1936–E1951, 2021.

Alberto Solera-Rico, Carlos Sanmiguel Vila, Miguel Gómez-López, Yuning Wang, Abdulrahman Almashjary, Scott TM Dawson, and Ricardo Vinuesa. β-variational autoencoders and transformers for reduced-order modelling of fluid flows. *Nature Communications*, 15(1):1361, 2024.

Meng Tang, Yimin Liu, and Louis J Durlofsky. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *Journal of Computational Physics*, 413: 109456, 2020.

Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Brendan D Tracey, Karthikeyan Duraisamy, and Juan J Alonso. A machine learning strategy to assist turbulence model development. In *53rd AIAA aerospace sciences meeting*, page 1287, 2015.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids*, 2(3):034603, 2017.

Zheng Wang, Dunhui Xiao, Fangxin Fang, Rajesh Govindan, Christopher C Pain, and Yike Guo. Model identification of reduced order fluid dynamics systems using deep learning. *International Journal for Numerical Methods in Fluids*, 86(4):255–268, 2018.

Rik Wanninkhof, William E Asher, David T Ho, Colm Sweeney, and Wade R McGillis. Advances in quantifying air-sea gas exchange and environmental forcing. *Annual review of marine science*, 1(1):213–244, 2009.

Brian Ward, Tim Fristedt, Adrian H Callaghan, Graig Sutherland, Xavier Sanchez, Jérôme Vialard, and Anneke Ten Doeschate. The air–sea interaction profiler (asip): An autonomous upwardly rising profiler for microstructure measurements in the upper ocean. *Journal of Atmospheric and Oceanic Technology*, 31(10):2246–2267, 2014.

Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7): 074602, 2018.

Jiayang Xu and Karthik Duraisamy. Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. *Computer Methods in Applied Mechanics and Engineering*, 372:113379, 2020.

Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366: 415–447, 2018.