PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

MODELING GLOBAL SURFACE DUST DEPOSITION USING PHYSICS-INFORMED NEURAL NETWORKS

CONSTANZA ANDREA MOLINA CATRICHEO

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ELWIN VAN'T WOUT

Santiago de Chile, April 2023

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

MODELING GLOBAL SURFACE DUST DEPOSITION USING PHYSICS-INFORMED NEURAL NETWORKS

CONSTANZA ANDREA MOLINA CATRICHEO

Members of the Committee:
ELWIN VAN'T WOUT
HECTOR IVAN JOAQUIN JORQUERA GONZALEZ
JULIEN SALOMON
FABRICE LAMBERT
RODRIGO ALFONSO ESCOBAR MORAGAS

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, April 2023

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my professors, Elwin Van't Wout, Fabrice Lambert, and Julien Salomon. When you find a supportive group of people who are willing to listen and guide you, it is invaluable. I would also like to extend my thanks to my family and friends for their unconditional love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Physics-informed neural networks (PINNs) have become increasingly popular, especially for solving partial differential equations (PDEs). PINNs can incorporate physical information about the process into the neural network architecture, reducing the solution space and making them an alternative when limited, sparse, and irregular data is available.

The objective of this thesis is to construct and evaluate the performance of a physics-informed neural network for measuring dust fluxes during the Last Glacial Maximum and Holocene periods. This methodology combines data analysis with physical principles to improve prediction accuracy. The results show that PINNs are a promising alternative to statistical methods such as Kriging when limited information is available.

In this study, physical modeling of the dust deposition was incorporated and PINNs accurately predicted the physical realistic flows of dust along the dominant wind directions. The findings of this study are promising, showing that PINNs can be used as an effective alternative when limited and irregular data is available.

Keywords: Physics-informed neural networks; machine learning; predictive modeling; numerical diffusion; deep learning method.

RESUMEN

Las redes neuronales informadas por física (PINNs, por sus siglas en inglés) se han vuelto cada vez más populares, especialmente para resolver ecuaciones diferenciales parciales (EDPs). Las PINNs pueden incorporar información física sobre el proceso en la arquitectura de la red neuronal, reduciendo el espacio de solución y convirtiéndolas en una alternativa cuando hay datos limitados, dispersos e irregulares disponibles.

El objetivo de esta tesis es construir y evaluar el rendimiento de una red neuronal informada por física para medir los flujos de polvo durante los periodos del Último Máximo Glacial y Holoceno. Esta metodología combina el análisis de datos con principios físicos para mejorar la precisión de la predicción. Los resultados muestran que las PINNs son una alternativa prometedora a los métodos estadísticos como Kriging cuando hay información limitada disponible.

En este estudio se incorporó la modelización física de la deposición de polvo y las PINNs predijeron con precisión los flujos realistas de polvo a lo largo de las direcciones de viento dominantes. Los resultados de este estudio son prometedores, mostrando que las PINNs pueden ser utilizadas como una alternativa efectiva cuando hay datos limitados e irregulares disponibles.

Palabras Claves: **Physics-informed neural networks; machine learning; predictive modeling; numerical diffusion; deep learning method.**

1. INTRODUCTION

1.1. Dust Modeling

The deposition of dust on the Earth's surface provides the terrestrial biosphere and surface ocean with micronutrients, affecting biogeochemical cycles. Additionally, dust particles can reflect or absorb incoming solar radiation, leading to changes in surface temperature and atmospheric circulation patterns. Changes in dust fluxes may have contributed to past climate variations, such as glacial-interglacial cycles [Lambert et al., 2015].

Previous efforts to simulate dust deposition under past climatic conditions were conducted using various global climate models. The complex nature of mineral dust aerosol emission makes it difficult to represent accurately in weather and climate models [Kok et al., 2014]. Hence, there are significant differences in dust load and spatial distribution among various dust simulations, even under modern conditions [Lambert et al., 2015].

The DIRTMAP database was designed to serve as a global validation dataset for use with earth system models of the paleo-dust cycle [Kohfeld and Harrison, 2001]. The dataset includes information on aeolian dust in ice cores, marine cores, and terrestrial sites, including loess-palaeosol successions and lacustrine cores for the Last Glacial-Interglacial cycle. Only a few sites around the world provide these conditions. The data includes, in addition to detailed locational information, mineralogy, geochemistry, sedimentology, geochronology, and derived accumulation rates [Derbyshire, 2003].

In Lambert et al. [2015], interpolation is performed with the Kriging method for The Dust Indicators and Records of Terrestrial and Marine Palaeoenvironments (DIRTMAP) data. These are compared with four models (CCSM, MIROC, MIROC-ESM, and MRI). The main results show that model outputs suggest that most climate models put too much emphasis on North African sources and underestimate sources in Asia and the Americas. In particular, models underestimate dust deposition in "High Nutrient Low Chlorophyll" (HNLC) regions.

1.2. Thesis Objective

Kriging is a geostatistical interpolation technique that assigns weights to nearby data points based on a distance function. An advantage of Kriging is that it also provides a variation estimate for the interpolation. A disadvantage of Kriging is that it loses accuracy when a few data points are concentrated, leaving large areas uncovered.

To accurately estimate dust fluxes in the past due to sparse data and the uncertainty of predictions where no data is available, we will incorporate physical information in addition to observed data to provide valuable insights into the dust flux process. The objective of this thesis is to improve the accuracy of estimating global dust fluxes during the Last Glacial Maximum and Holocene periods by combining data analysis with a partial differential equation that corresponds to the physical principles. By doing so, we aim to obtain more accurate estimates that are consistent with the underlying physical processes.

1.3. The Proposed Hypothesis

A Physics-Informed Neural Network (PINN) is a machine learning method that incorporates physical information related to the data into the neural network architecture. This technique allows the neural network to learn and generalize from limited and noisy data.

By incorporating physical information into a physics-informed neural network, we can improve the accuracy of sparse and irregularly located climatological data on Earth, particularly for DIRTMAP data during the Last Glacial Maximum and Holocene periods. Combining measured data with a climate model in a single prediction that is more accurate than each component alone can achieve this improvement. The modeling of the global dust flow using a partial differential equation will provide the necessary physical information to improve the accuracy of the estimations.

1.4. Thesis Overview

This section provides a summary of the main contents of each chapter in this thesis. First, in Chapter 2, an explanation of the theoretical basis of the Kriging method will be provided to compare it with the PINN method, which is explained in Chapter 3.

Next, in Chapter 4, the performance of these methods will be evaluated by analyzing the one-dimensional time-dependent diffusion equation and transport equation under different conditions. In addition, in Chapter 5, the study will be extended to two-dimensional time-dependent PDE problems to understand different parameters for the appropriate construction of the PINN.

Chapter 6 presents the dust fluxes dataset for the Last Glacial Maximum and Holocene times. For each period, two datasets are included: the DIRTMAP database and the simulations from the Community Earth System Model (CESM). In Chapter 7, the PINN's capability to model transport phenomena will be tested on the simulated data from the LGM.

Finally, in Chapter 8, the kriging interpolation performed in [Lambert et al., 2015] for the DIRTMAP database will be compared with the PINN's prediction. Here, we will see that the major advantage of using PINN instead of the kriging method is the direct incorporation of physical information, such as the influence of wind. Additionally, the Earth's surface will be approximated using spherical coordinates, which take into account distances in latitude with periodic boundary conditions.

2. KRIGING

This chapter briefly describes the kriging method used for dust interpolation in DIRTMAP, as described in [Lambert et al., 2015]. The method assumes that a variable can be explained by the distance of neighboring variables. It was developed by Georges Matheron, who named the technique after D.J. Krige, a South African mining engineer who did some early work on the topic [Query, 2017].

There are certain statistical assumptions in the kriging method. Let $Y(s)$ denote the realization of a regionalized variable at location s. The assumptions are as follows:

(i) Second-order stationarity: The data is stationary of order two, meaning that local distributions have the same constant mean and variance.

(ii) Spatial dependence: The first and second moments of the difference between regionalized variables, i.e., $Y(s+h) - Y(s)$, depend only on the separation distance h and not on the location s.

The variogram γ is defined as

$$2\gamma(h) = \mathrm{Var}[Y(s+h) - Y(s)]. \tag{2.1}$$

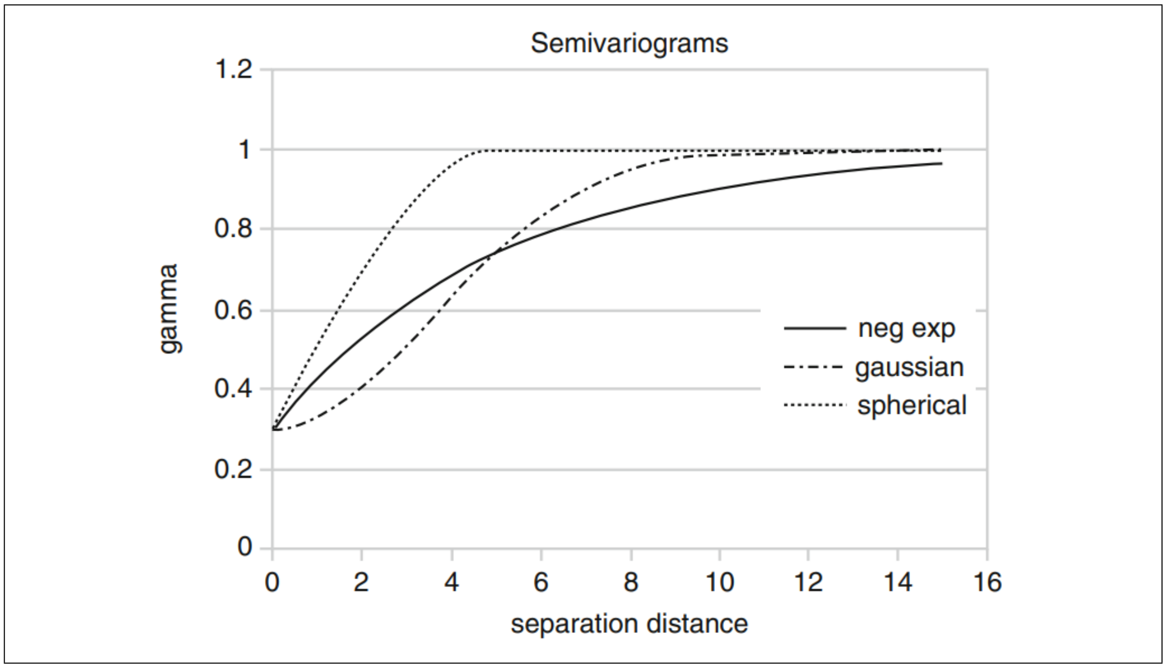

Figure 2.1. Semivariogram of three model distributions: neg exponential, gaussian and spherical. Figure taking from [Query [2017], Figure 2] .

The semivariogram is defined as one half of the variogram. In Figure 2.1, it can be seen that as the distance between points increases, the variogram remains roughly constant. This is due to the assumption that the second moment is finite. A model must be fitted to the observed data to recreate the dependence of the variogram on distance. The dependency of the variogram changes mostly at low separation distances. If the variance is finite, the correlogram and the variogram are related by

$$\gamma h = \sigma^2(1 - k(h)), \tag{2.2}$$

where σ^2 is the variance and $k(h)$ is the correlation at separation distance h.

When the correlogram is known, we consider the model

$$Y(s) = m(s) + u(s), \tag{2.3}$$

where m denotes the mean, which may vary spatially and u is the variation of Y about its mean, with $\mathbb{E}[u] = 0$.

The kriging predictors are unbiased in the sense that $\mathbb{E}[P(s_0)] = \mathbb{E}[Y(s_0)]$, where P is the predictor for Y at the site s_0. This means that the expected value of the kriging predictor P at a specific location s_0 is equal to the expected value of the true value Y at the same location s_0. This important property of kriging means that, on average, the predictions are expected to be accurate.

There are two categories of kriging methods: linear and nonlinear kriging. Linear kriging assumes that the spatial dependence is linear, while nonlinear kriging methods allow complex spatial dependence structures. Linear kriging algorithms are simple kriging, ordinary kriging, universal kriging, Bayesian kriging, and factorial kriging. Examples of nonlinear kriging include lognormal kriging, multi-Gaussian kriging, disjunctive kriging, indicator kriging, probability kriging, and rank kriging [Asa et al., 2012].

There are different assumptions between the type of kriging methods, for example:

(i) Simple kriging: m is assumed to be a known constant

(ii) Ordinary kriging: m is assumed to be an unknown constant

(iii) Universal kriging: m is unknown and varies spatially

The fitted variogram is used to calculate the kriging weights, which are then used to interpolate or predict the value of the variable at an unsampled location.

3. PHYSICS INFORMED NEURAL NETWORKS

A Physics-informed neural network (PINN) is an algorithm based on supervised machine learning that can intrusively combine data with physical information modeled as a partial differential equation. In our case, the input of the PINN is the spatial location on the earth and the output is the dust fluxes. During the training process, the PINN considers the dataset and the partial differential equation and optimizes a nonlinear function that fits both data and model. This optimization process requires specifying a loss function to find optimal parameters for the neural network. Since the PINN uses automatic differentiation to calculate the derivatives of the objective function we can include any partial differential equation and boundary conditions in the loss function. Figure 3.1 shows a general schema of the PINN design.

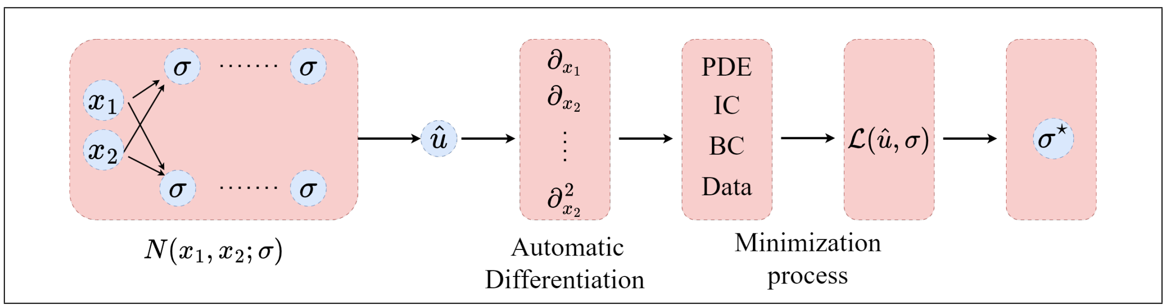

Figure 3.1. A general PINN training scheme where x_1 and x_2 are the input variables, σ are the parameters of the neural network and u the objective function. Automatic differentiation calculates de derivatives of the objective function and the data and model fits are combined in a loss function $\mathcal{L}$. The minimization process through backpropagation gives updated parameters $\sigma^\star$.

The objective of this thesis is to explore the advantages and disadvantages of a PINN in global dust flux modeling. This methodology is particularly interesting because it combines data analysis with physical principles to improve the accuracy of neural network predictions. The input information consists of observation data and physical information, such as partial differential equations (PDEs) and their respective conditions. Due to the inclusion of physics information, fewer data points are needed for neural network training. The trained neural network can then be applied to new input data to obtain an approximation of the output variable.

In this next Chapter, I will explain how the PINN works and its advantages. Following this, in Chapters 4 and 5, I will explore several academic examples of equations with different variations to understand the characteristics that might explain the performance of the predictions using PINN. Additionally, in Chapter 7, I will apply the PINN to a simulated dataset for different velocities in the transport equation to study its performance.

Additionally, in Chapter 8, real data on dust fluxes in the Holocene and LGM period are utilized. In this chapter, a mathematical model is built to provide additional information to the limited observations available and compare the results with another approach, namely the Kriging method.

By exploring these different examples, this thesis will provide insights into the use of PINN and its applicability to different problems, particularly in the field of physical modeling.

3.1. Loss Function

The objective of a PINN is to approximate a function $u(\mathbf{x}, t)$ which fits the partial differential equation and the observed data. For this purpose, we define a combined loss function as:

$$\mathcal{L} = w_1 \mathcal{L}_{\text{data}} + \mathbf{w}_2 \cdot \mathcal{L}_{\text{model}}, \tag{3.1}$$

where $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{model}}$ are the loss function corresponding to the data and model, respectively, and w_1 and $\mathbf{w}_2$ are the respective weights. The data loss function is given by

$$w_1 \mathcal{L}_{\text{data}} = w_1 ||\hat{u}(\tilde{\mathbf{x}}, t; \sigma) - \tilde{u}(\tilde{\mathbf{x}}, t)||, \tag{3.2}$$

where $\tilde{u}$ denotes the observations at the points $\tilde{x}_j$ for $j = 1, \ldots, N_{\text{data}}$ with N_{data} the number of training items, and $\hat{u}$ denotes the PINN's prediction. We use the standard L_2

norm for every loss function. The model loss function $\mathcal{L}_{\text{model}}$ is defined as:

$$\mathbf{w}_2 \cdot \mathcal{L}_{\text{model}} = w_{2,1}\mathcal{L}_{\text{PDE}} + w_{2,2}\mathcal{L}_{\text{BC}} + w_{2,3}\mathcal{L}_{\text{IC}}, \tag{3.3}$$

and consists of three parts, one for the PDE ($\mathcal{L}_{\text{PDE}}$), one for the boundary conditions ($\mathcal{L}_{\text{BC}}$), and another one in case of the problem the initial conditions ($\mathcal{L}_{\text{IC}}$).

For example,

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), \qquad a < x < b, \qquad t > 0, \\ u(x,0) = g(x), \\ u(a,t) = h_1(t), \\ u(b,t) = h_2(t), \end{cases} \tag{3.4}$$

with $f(x,t)$ as a source term, $g(x)$ as the initial condition, and $h_1(t)$, $h_2(t)$ as the boundary conditions. In this case, the loss functions present for equation (3.4) are defined as:

$$\begin{cases} \mathcal{L}_{\text{PDE}} = \left\| \frac{\partial \hat{u}(x'_j,t'_j)}{\partial t} - \frac{\partial^2 \hat{u}(x'_j,t'_j)}{\partial x^2} - f(x'_j,t'_j) \right\|, \\ \mathcal{L}_{\text{IC}} = ||\hat{u}(\bar{x_j},0) - g(\bar{x_j})||, \\ \mathcal{L}_{\text{BC}} = ||\hat{u}(a,\bar{t_j}) - h_1(\bar{t_j})|| + ||\hat{u}(b,\bar{t_j}) - h_2(\bar{t_j})||, \end{cases} \tag{3.5}$$

where x'_j, t'_j with $j = 1\ldots, N_{\text{PDE}}$ are the training set of the domain. Notice that this amount of data for the training process depends just on the location where the PDE is evaluated.

For the boundary conditions, we use $\bar{t_j}$ with $j = 1, \ldots, N_{\text{BC}}$, where N_{BC} is the number of boundary training points. Similarly, we use $\bar{x}_j$ with $j = 1, \ldots, N_{\text{IC}}$ for the initial conditions, where N_{IC} is the number of initial condition training points. The amount of data available for these loss functions depends on the specific information of the problem.

It is also important to note that we have as many loss functions as conditions we want to include, in addition to the equation to be solved. Then, it is part of the analysis to find the value of the importance weights of each loss function that obtain the best result for the problem.

When the problem does not include time, there is no initial condition and when the problem domain does not have physical boundaries, there are no boundary conditions. In those cases, the corresponding loss term is eliminated from the model loss.

3.1.1. Data Loss

Depending on the amount of information about the problem, there are different possible scenarios for the data loss term in the PINN, such as:

(i) Measured data is available, so the data loss is the difference between the prediction and the observations.

(ii) PINNs can be used with the model loss only to solve PDEs.

(iii) For testing purposes, one can use a PDE with an exact solution and use this exact solution of the PDE as data. To test the PINNs one can generate data of the exact solution, and to test sensitivity one can add noise to the data.

3.2. Design of Neural Network

For the design of a neural network, a standard architecture must be taken into consideration, where the components may vary depending on the nature of the problem to be solved and the available data. The main components to consider are:

- Network architecture: The type of network, the number of input and output neurons, and the number of hidden layers and neurons per hidden layer can be chosen based on the nature of the problem and the available data.
- Loss function: there are use various loss functions, such as Mean Squared Error, Mean Absolute Error, and Huber Loss, to optimize the neural network.

- Batch Size: The number of samples used in each iteration of the optimization algorithm can be adjusted.
- Number of epochs: The number of times the entire dataset is passed through the network during training can be specified.
- Learning Rate: The step size used in the optimization algorithm can be adjusted to improve the convergence rate of the network.
- Optimizer: Various optimization algorithms, such as Adam, SGD, and LBFGS, can be used to train the network.
- Activation functions: Different activation functions, such as ReLU, Sigmoid, and Tanh, can be used in the hidden layers of the network.

PINN is defined as a combination of a data loss function and a model loss function. The data loss function measures the error between the predicted and observed data, while the model loss function measures the error between the PINN prediction and the physical model represented by partial differential equations, boundary conditions, and initial conditions. Depending on the information available in the problem, PINN combines data analysis with physical principles to improve the accuracy of neural network predictions. In the following chapters, PINN will be explored and applied to various academic examples to understand its performance and applicability to different problems.

4. COMPUTATIONAL BENCHMARK

PINNs, as explained in Chapter 3, and the Kriging method, as explained in Section 2, are techniques that are useful in situations where data is scarce, incomplete, or noisy.

To evaluate the performance of these methods, we will conduct several benchmark experiments. For PINNs, we will solve the diffusion equation in Section 4.1 and the transport equation in Section 4.2. For Kriging, we will perform interpolation using some observations from the analytical solution of those equations. This enables us to evaluate the performance of both methods under varying conditions.

For the diffusion equation, we will compare the performance of PINNs and Kriging to the known analytical solution in different benchmarks. In Section 4.1.3, we analyze the results obtained with different quantities of information. Also in Section 4.1.4, we also examine the training process with noisy data. Finally, in Section 4.2.1 we compare the computational cost of each method.

With the transport equation, we will first study the consequences of the frequency of waves in a small area in Section 4.2, and then the computational cost associated with this in Section 4.2.1. We will also analyze the performance when a non-uniform grid of points is available in Section 4.2.2, as well as the effect of the number of points in the loss function in Section 4.2.3.

The library used to solve PINNs is called DeepXDE Lu et al. [2021], and it is an open-source Python package that contains various useful methods, such as mesh creation for the equation's domain, network construction, and initial and boundary condition specification. The library used for fitting Kriging models is called SciKit GStat, and it can be installed directly from the Python Package or GitHub.

4.1. Diffusion Equation

Diffusion equations are an important type of parabolic equations that result from diffusion phenomena found in nature. They serve as mathematical models for physical problems in diverse fields, such as filtration, biochemistry, phase transition, and dynamics of biological groups [Li et al., 2001].

The problem to solve is

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - e^{-t}\left(1-\pi^2\right)\sin(\pi x), & t \in (0,1), x \in (-1,1), \\ u(0,x) = \sin(\pi x) + 2, \\ u(t,\pm 1) = 2, \end{cases} \tag{4.1}$$

where the exact solution is

$$u(t,x) = e^{-t}\sin(\pi x) + 2. \tag{4.2}$$

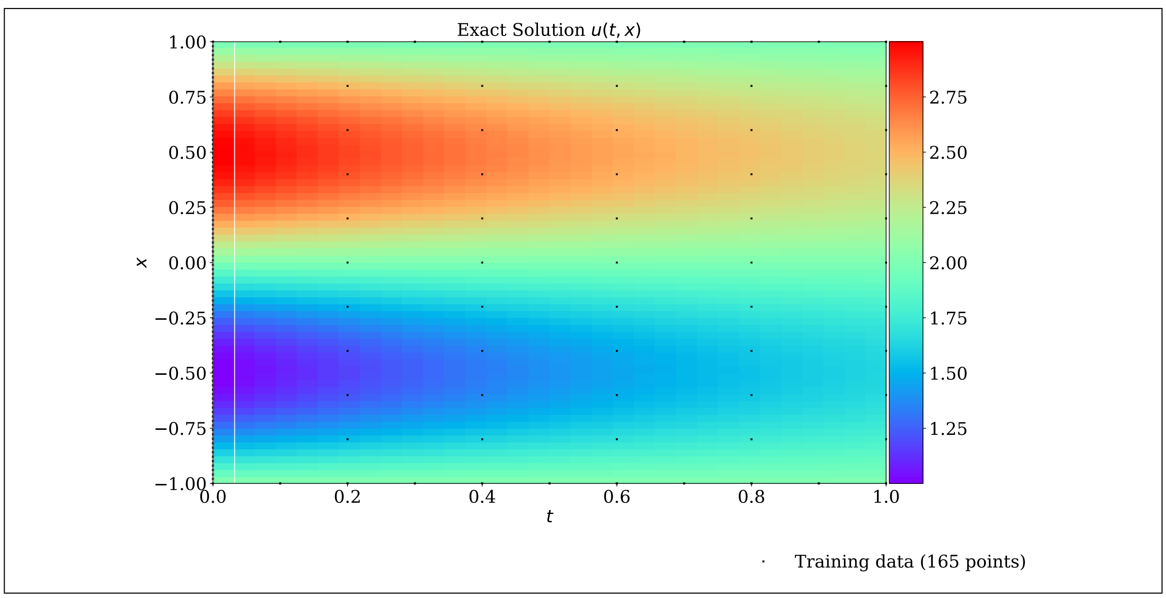

Figure 4.1. Analytical solution of the diffusion equation problem (4.1).

4.1.1. Diffusion Equation by PINNs

The network used to solve the problem has three hidden layers, each with 32 weights. The Glorot uniform initializer was used, along with the `tanh` activation function and the L-BFGS-B optimizer. It has been observed that changing the optimizer to Adam results in worse performance on the loss function, but there is no significant difference in the visualization of the prediction of the solution. This suggests that depending on the number of training points and the nature of the problem, Adam or L-BFGS-B optimizer may have different convergence characteristics.

As mentioned in Section 3.1.1, the loss function of the network in equation (3.1) can incorporate information from both the equation being solved and observations of the solution. The first question to analyze is how sensitive the network's performance is to the amount of model data versus observation data used. Table 4.1 shows the performance of networks with different numbers of training data points. The aim is to keep the total number of training points constant, which is approximately 165 points, and vary the percentage of data used for the PDE loss function versus the observed solution data.

The results show that the network with only PDE information has similar performance to the network that includes both PDE information and observations of the solution. However, the network with only observed information performs worse on testing metrics, but has a shorter training time than the networks that include PDE information. This is consistent with the known weakness of neural networks, which require a large amount of data to accurately approximate a solution. However, as the theory behind PINNs suggests, adding additional physical information can lead to better results for a network that approximates a function and its derivatives.

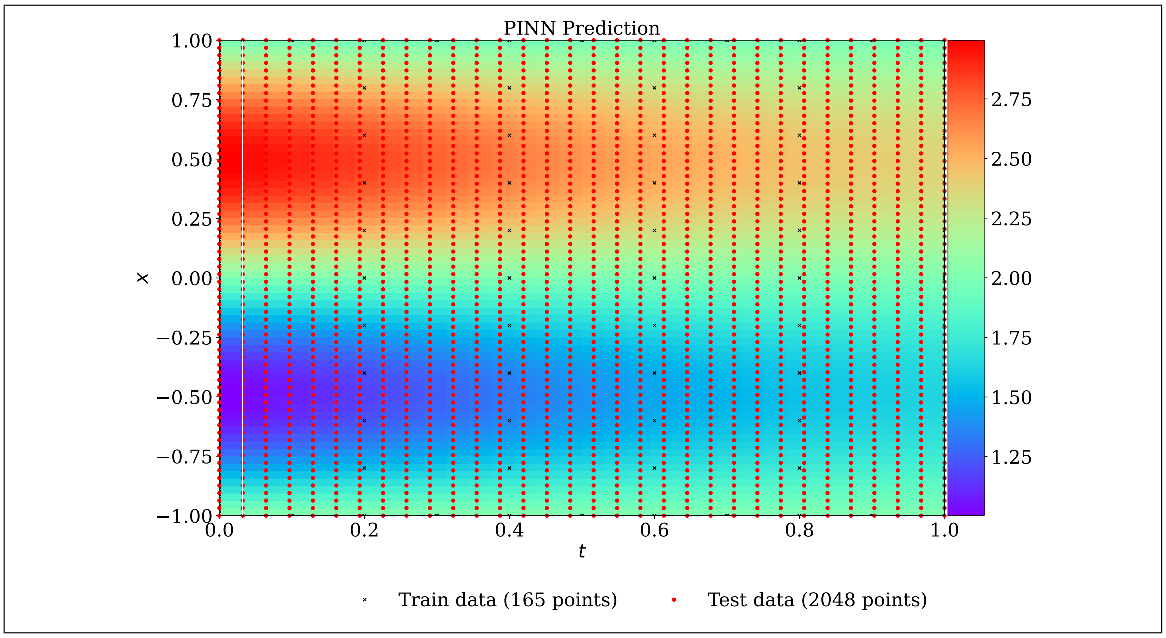

Figure 4.2. PINN's prediction for problem in equation 4.2.

4.1.2. Diffusion Equation by Kriging

To define the experimental variogram and make predictions, a function that takes distance as input and returns a semi-variance value must be defined. There are different methods to define this function, but it must satisfy the principle of monotonically increasing values that ultimately reach an upper limit. This principle means that the correlation between the points depends on their distance from one another. It should be noted that, as shown in Figure 4.3, the training data does not adhere to the principle of independence at far distances. This is because we used boundary conditions where $u = 2$ at $x = 1$ and $x = -1$, resulting in equal values at distances of around 2, corresponding to interactions between boundary points.

It is important to note that in the case of interpolation performed on an equation with time domain dependence, Kriging considers a Euclidean distance between time and distance to fit the model. Therefore, Kriging assumes that the future depends on the past and vice versa, and this correlation in time means that no causality is taken into account. The interpolation of the analytical solution is shown in Figure 4.5.

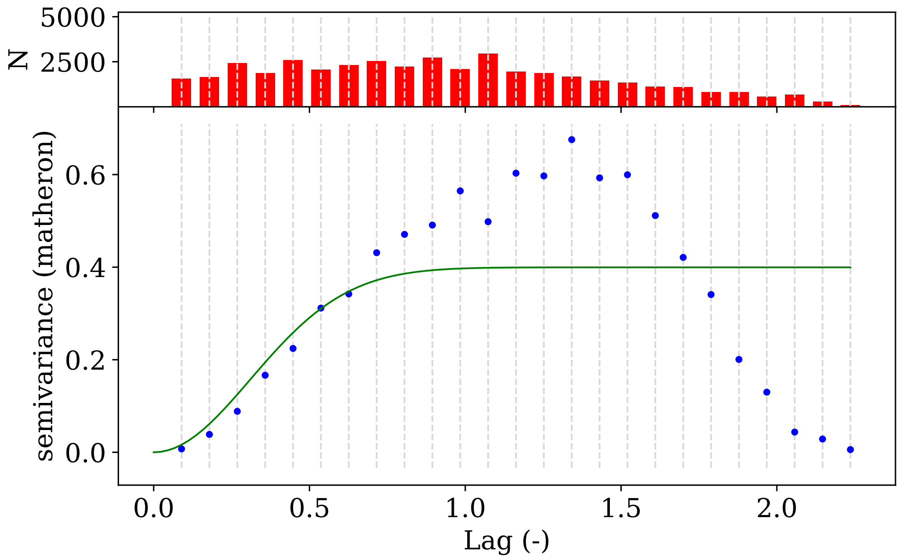

Figure 4.3. Variogram fitted with a Gaussian model for 165 points obtained from the analytical solution in equation 4.2.

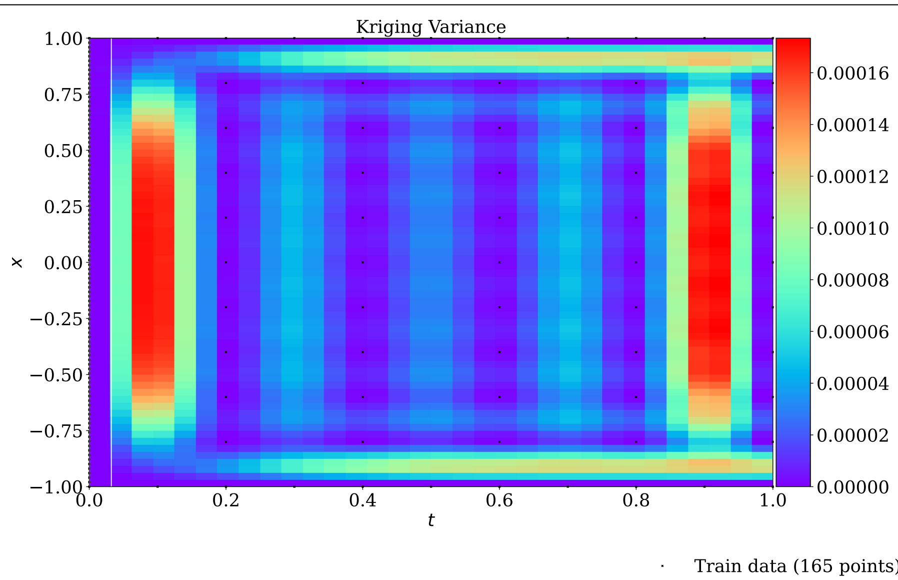

Figure 4.4. Variance obtained from Kriging interpolation across the domain.

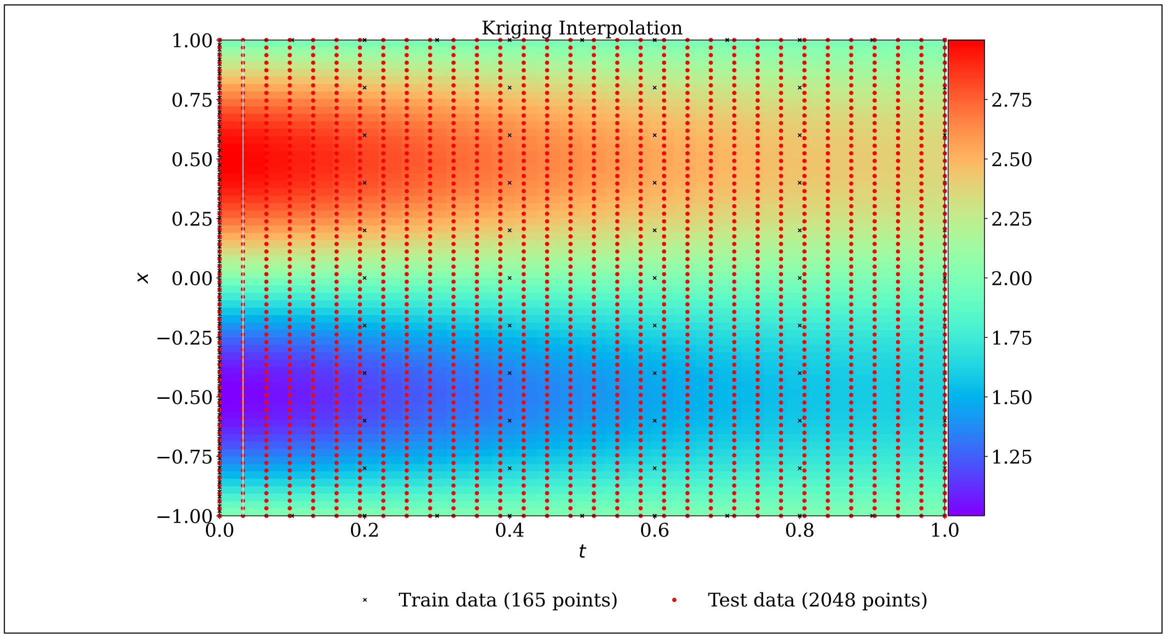

Figure 4.5. Kriging interpolation for problem in equation 4.2.

Both Figures, 4.4 and 4.6, indicate that the variance and relative error increase in areas where there are no training points. However, notice that in Figure 4.4 is very small, smaller than 0.00016, so the interpolation should be very accurate.

4.1.3. Comparison Between Methods

As explained in Chapter 2, Kriging is an interpolation method that requires observed data to fit the model. In contrast, PINNs require information from the equation being solved, and observed data, and the amount of data used from both the equation and the observations can vary. In this section, we will discuss the results of both models for Equation (4.1) and explore the differences in PINN's predictions trained with different quantities of data.

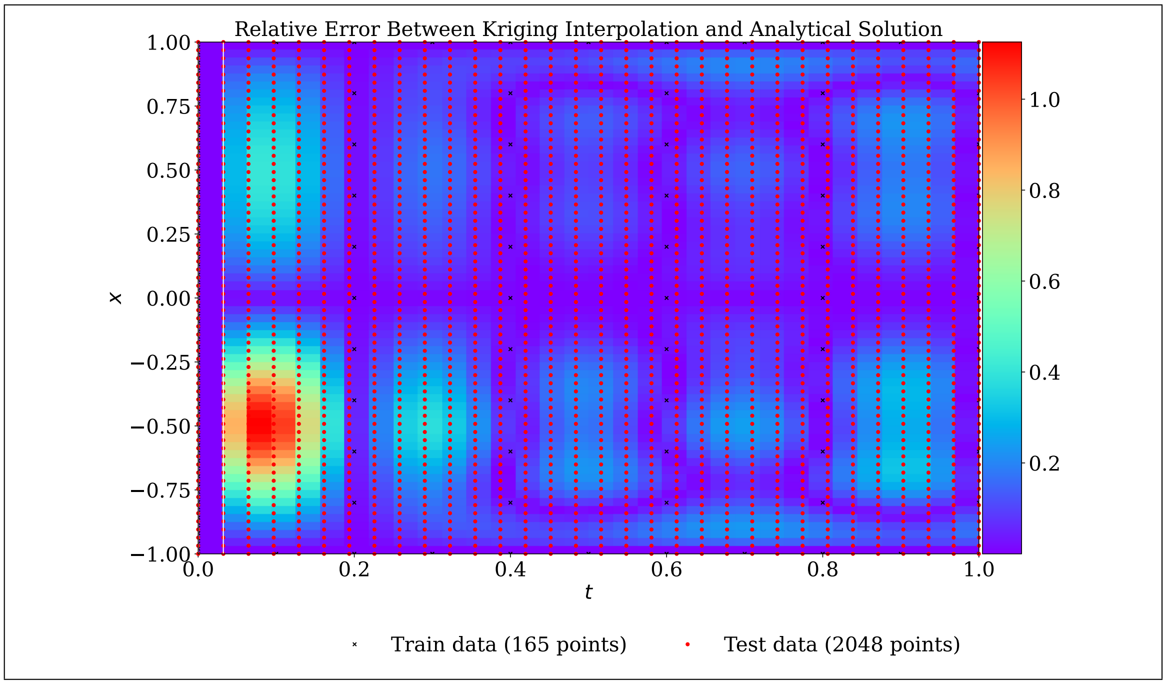

Figure 4.6. Relative error of the Kriging Method, expressed as a percentage. The colorbar indicates the magnitude of the error.

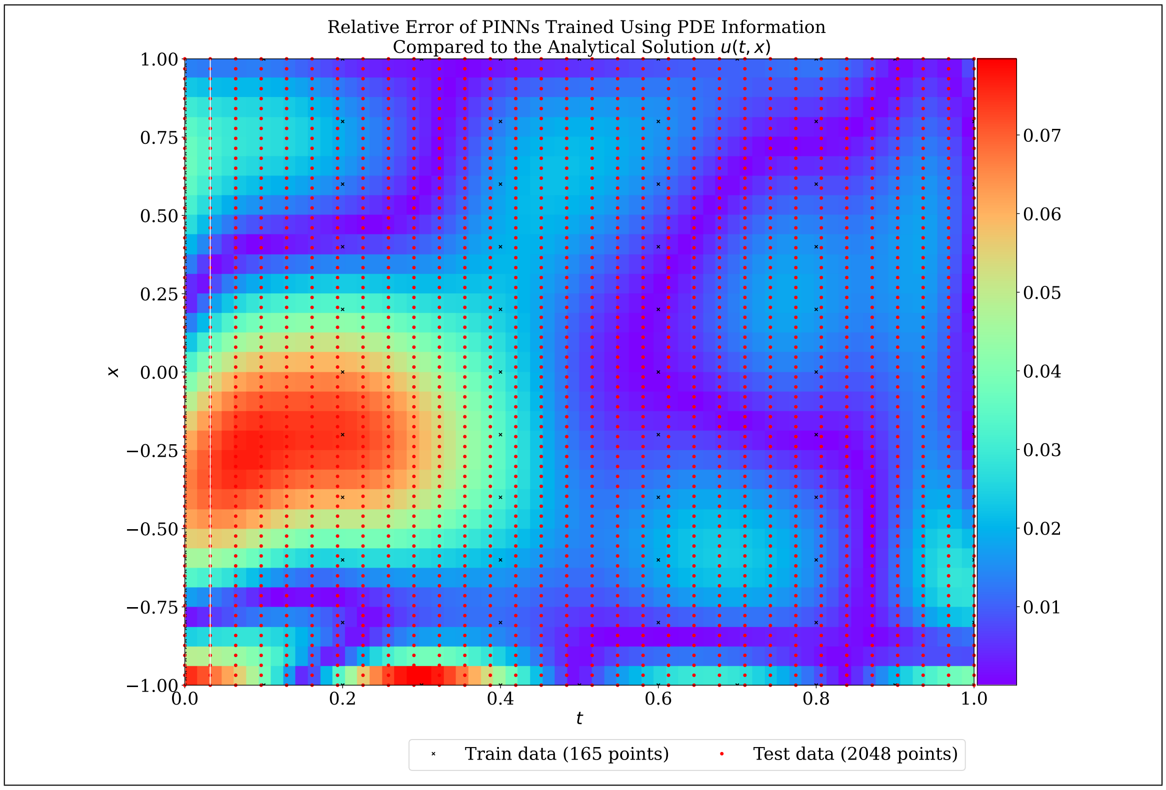

Figure 4.7. Relative error of PINNs expressed as a percentage. The network was trained with the model information. The colorbar indicates the magnitude of the error.

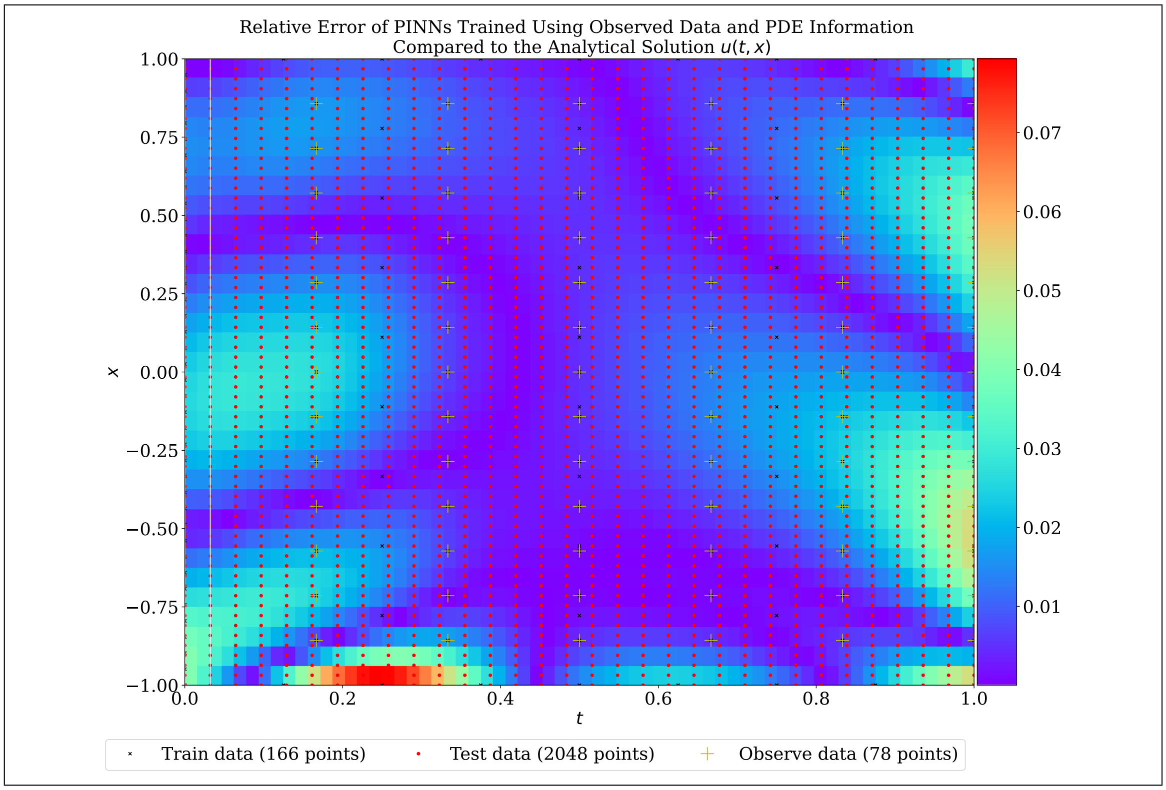

Figure 4.8. Relative error of PINNs expressed as a percentage. The network was trained with PDE and data observation. The colorbar indicates the magnitude of the error.

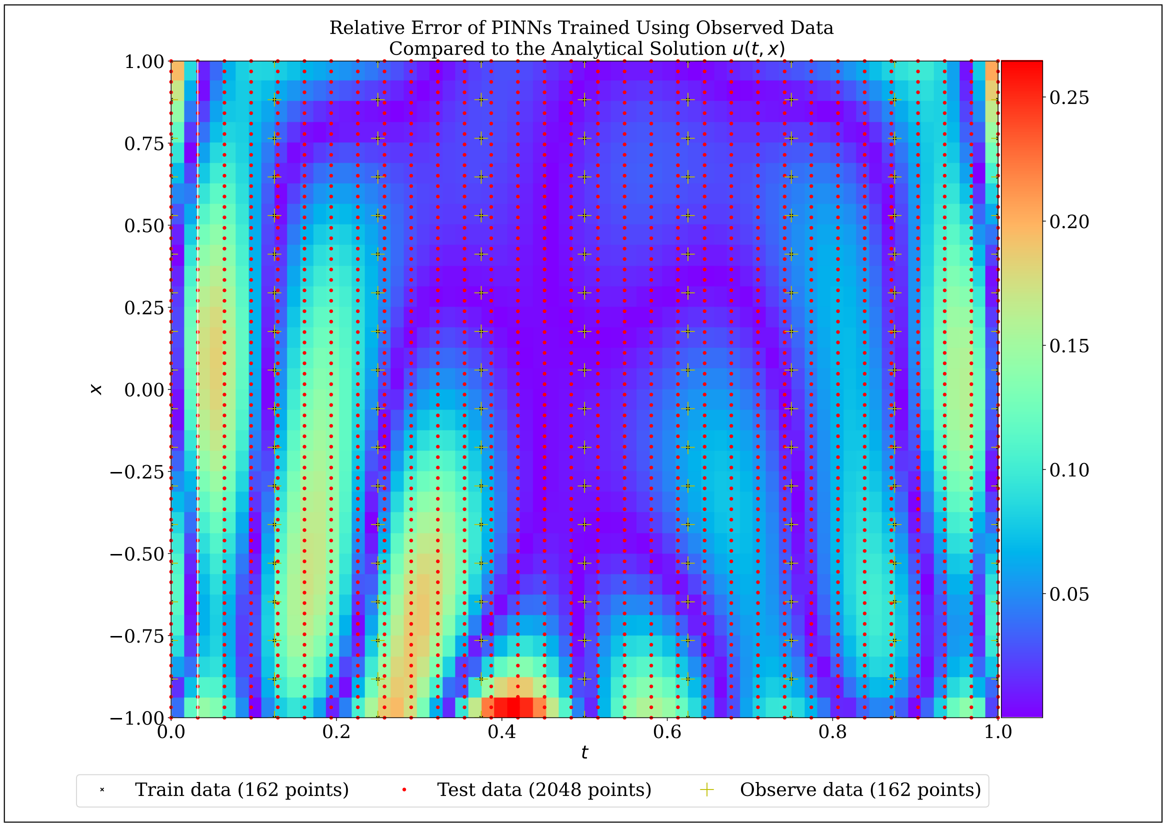

Figure 4.9. Relative error of PINNs expressed as a percentage. The network was trained with data observation. The colorbar indicates the magnitude of the error.

In Figures 4.6, 4.7, 4.8, and 4.9 the relative error is shown, which is defined as:

$$\frac{|x_{\text{true}} - x_{\text{prediction}}|}{|x_{\text{true}}|} \times 100. \tag{4.3}$$

Table 4.1. Comparison between PINNs and Kriging to analytic solution. Results were obtained using 2048 test points uniformly distributed. Time measurements for PINNs include both training and prediction processes, while for Kriging it includes model fitting.

Method	**Observed Data**	**PDE data with BC and IC**	L_2**-norm**	Max**-norm**	**RMSE**	**Training Time**
Kriging	165	0	0.1474	0.0119	0.0026	0.828
PINNs	0	165	0.0187	0.0015	0.0004	15.22 s
PINNs with Observed Data	78	88	0.0134	0.0011	0.0002	14.39
Deep Learning	162	0	0.0886	0.0058	0.0019	9.83 s

Table 4.1 shows the error between the prediction of the network and the interpolation performed by Kriging, using the analytical solution. For this testing, a grid of 2048 uniformly distributed points was used, and in all cases, favorable results were obtained.

First, can be observed that the network's error does not decrease in areas close to the training points. In contrast, in the interpolation carried out by Kriging, this is observable. This may be due, as mentioned above, to the fact that the network is not using information from the analytical solution, but rather from the equation and its initial and boundary conditions for training.

In the case of the interpolation performed by Kriging, the relative error in Figure 4.6 and the variance in Figure 4.4 are symmetric around the line $x = 0$. This is because Kriging uses the distance of the domain to perform the interpolation and calculate the variance, and because the analytical solution is symmetric. Nevertheless, The symmetric difference is not seen in the network's prediction. The main reason for this is because the network fits a nonlinear function through a minimization process with a global loss term.

4.1.4. Training with Noisy Data

When considering real-world problems, it is important to analyze the sensitivity of the model to noisy observation data. In this section, we conducted an experiment where we added normally distributed noise and gradually increased the standard deviation of the noise. We compare two models: one trained using only noisy information of the analytical solution, and the other trained using both analytical and PDE information for the training. Finally, we analyze the performance of the kriging method when it is fitted with noisy data.

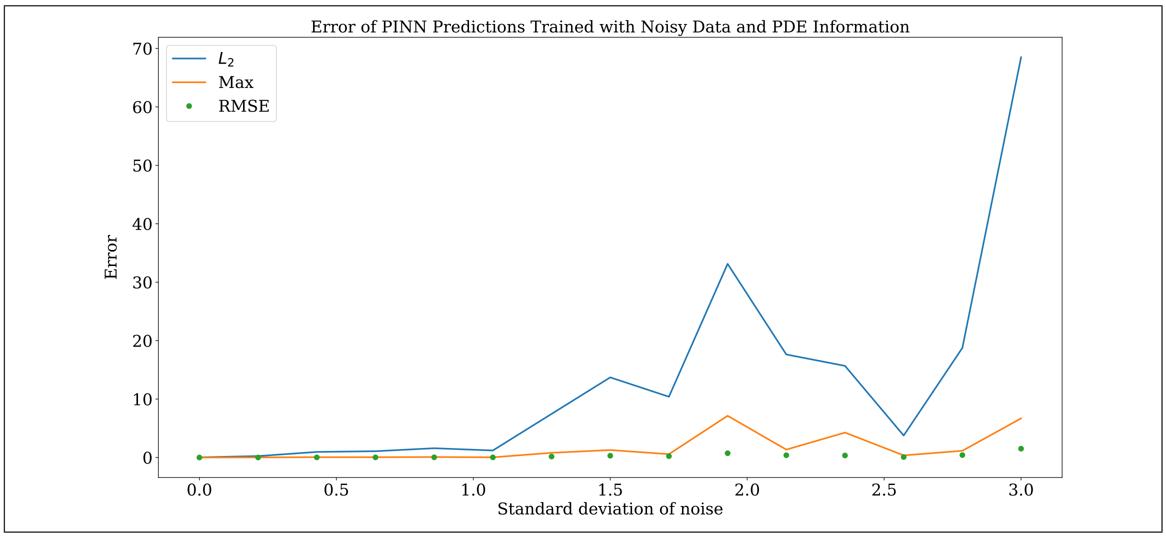

Figure 4.10. Comparison of errors between PINN predictions trained with varying levels of noise in the analytical solution and PDE information. For the training process we used 88 and 78 points, for the data and PDE, respectively. The error of the PINN is measured using 2000 data points. The horizontal axis represents the standard deviation of the noise added to the analytical solution data.

Figure 4.10 shows that the error trend increases as the standard deviation of the noise increases. The increase is not linear and can even fluctuate as the error increases. These peaks could be due to the network not having yet converged. However, up to 1.5 standard deviations, the network's performance remains relatively constant, which can provide some confidence when working with normally noisy data.

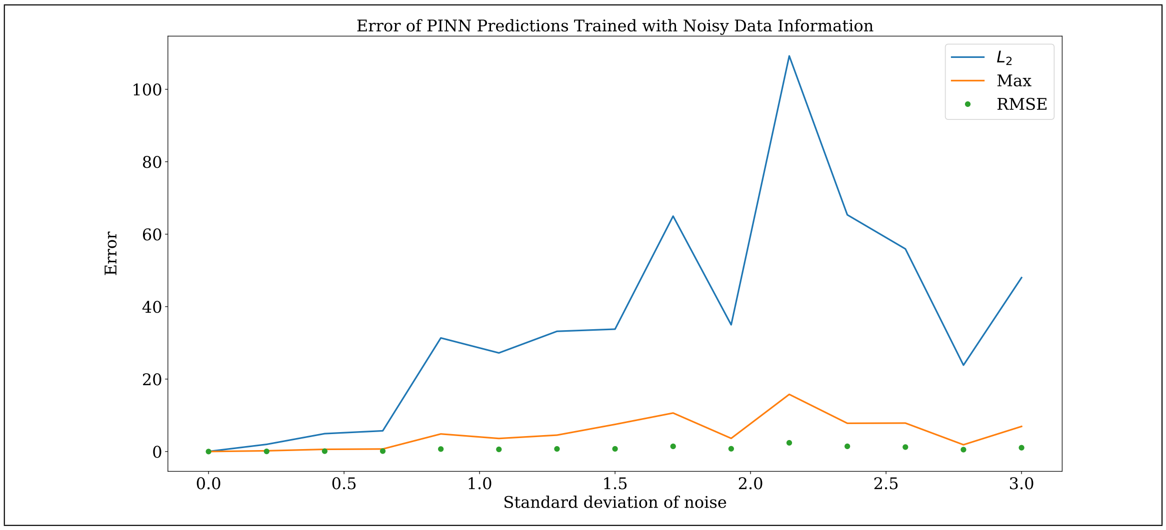

Figure 4.11. Comparison of errors between PINN predictions trained with varying levels of noise in the analytical solution. For the training process, we used 162 and 1 point for the data and PDE, respectively. The error of the PINN is measured using 2000 data points. The horizontal axis represents the standard deviation of the noise added to the analytical solution data.

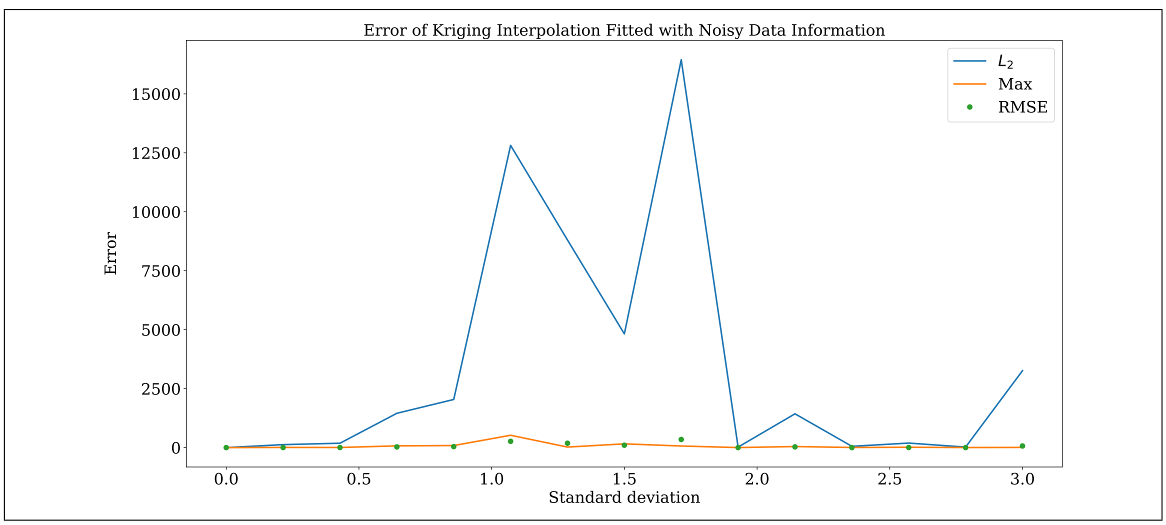

Figure 4.12. Comparison of errors between Kriging method fitted with varying levels of noise in the analytical solution. For the Kriging fit, we used 160 randomly sampled points from the analytical solution. The error of the Kriging interpolation is measured using 2000 data points. The horizontal axis represents the standard deviation of the noise added to the analytical solution data.

Figures 4.11 and 4.12 show the same experiment performed using the PINN trained only with data information and the Kriging method, respectively. In both cases, the error increases at a lower standard deviation range (0.5-1) compared to the network that includes PDE information. This could be because the PDE information acts as a regularizer for the noisy information. Additionally, both cases exhibit a similar pattern of oscillations. Finally, it is important to note that the error of the Kriging method, despite appearing to converge quickly and fluctuate, has a higher value compared to the error of both PINN predictions.

4.2. Transport Equation

In this section, we present the transport equation which describes how a scalar quantity is transported. In this case, there are two variables: a spatial variable and a temporal variable. The equation has three parameters, namely a, b, and c, which are real numbers ($\in \mathbb{R}$). Here, a is the amplitude of the wave, b is an offset that serves as an average state, and c corresponds to the velocity of the transportation.

$$\begin{cases} \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 1, \qquad x \in (0, 2\pi), t \in (0, 1), \\ u(x, 0) = a\sin(x) + b, \\ u(0, t) = t + a\sin(-ct) + b, \\ u(2\pi, t) = t + a\sin(2\pi - ct) + b, \end{cases} \tag{4.4}$$

where the exact solution is

$$u(x, t) = t + a\sin(x - ct) + b. \tag{4.5}$$

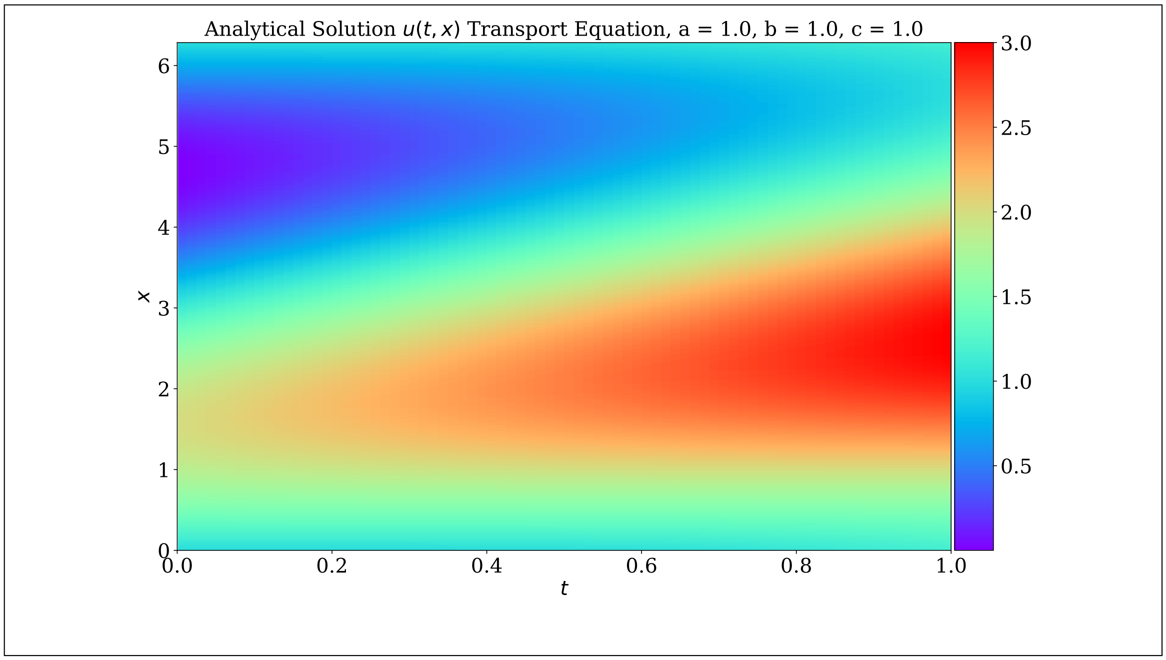

Figure 4.13. Solution transport equation for $a = 1$, $b = 1$, $c = 1$.

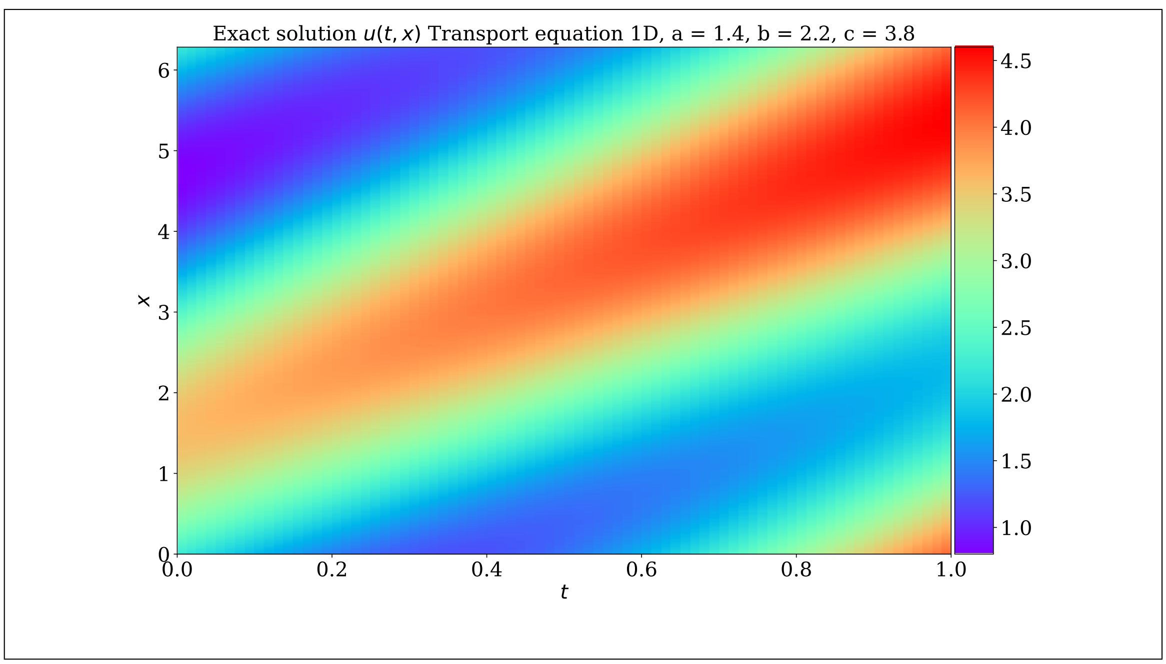

Figure 4.14. Solution transport equation for $a = 1.4$, $b = 2.2$, $c = 3.8$.

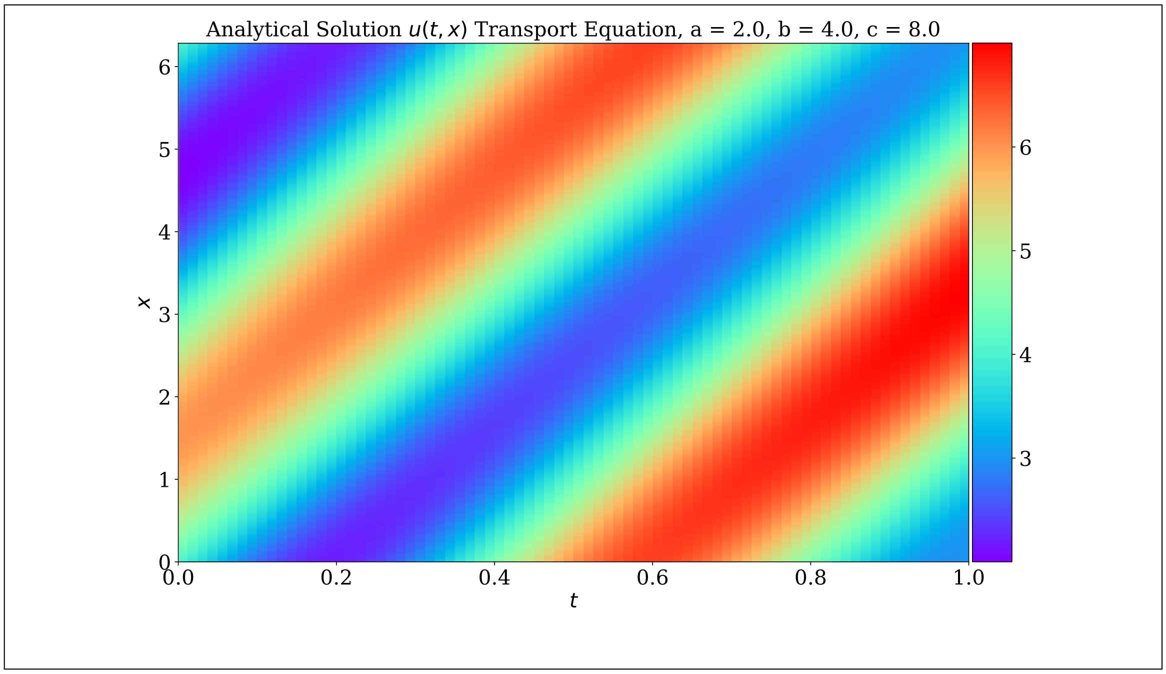

Figure 4.15. Solution transport equation for $a = 2$, $b = 4$, $c = 8$.

The solution of the equation for some parameters can be seen in Figures 4.13, 4.14, and 4.15. The aim of this study is to investigate the performance of the PINN and Kriging methods for solving the transport equation with different values of the parameters (a, b, c).

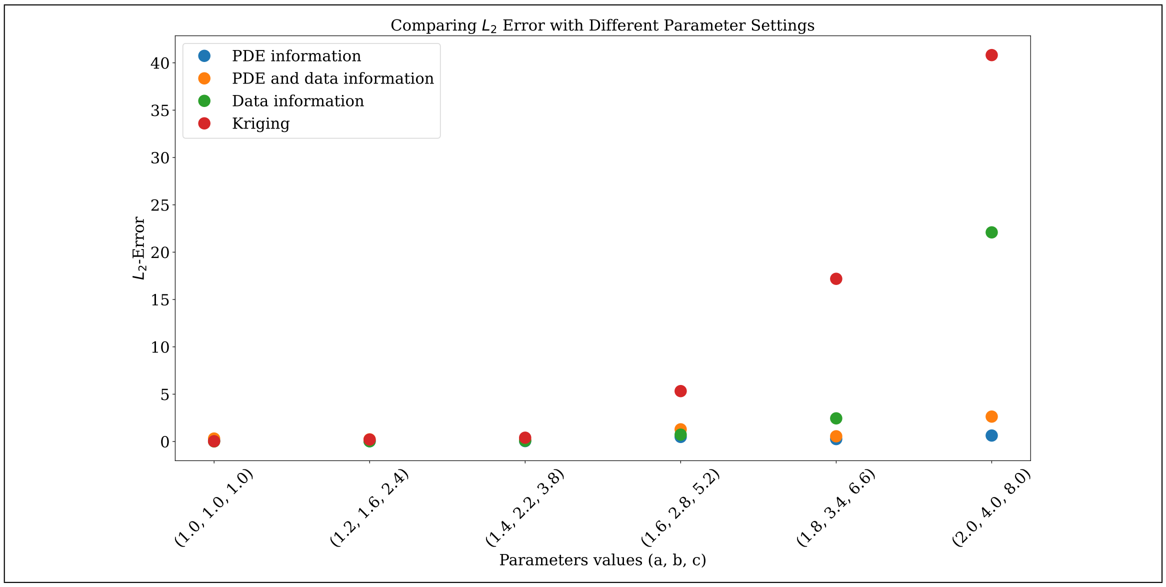

Figure 4.16. L_2 error of the different models with parameters of different values (a, b, c).

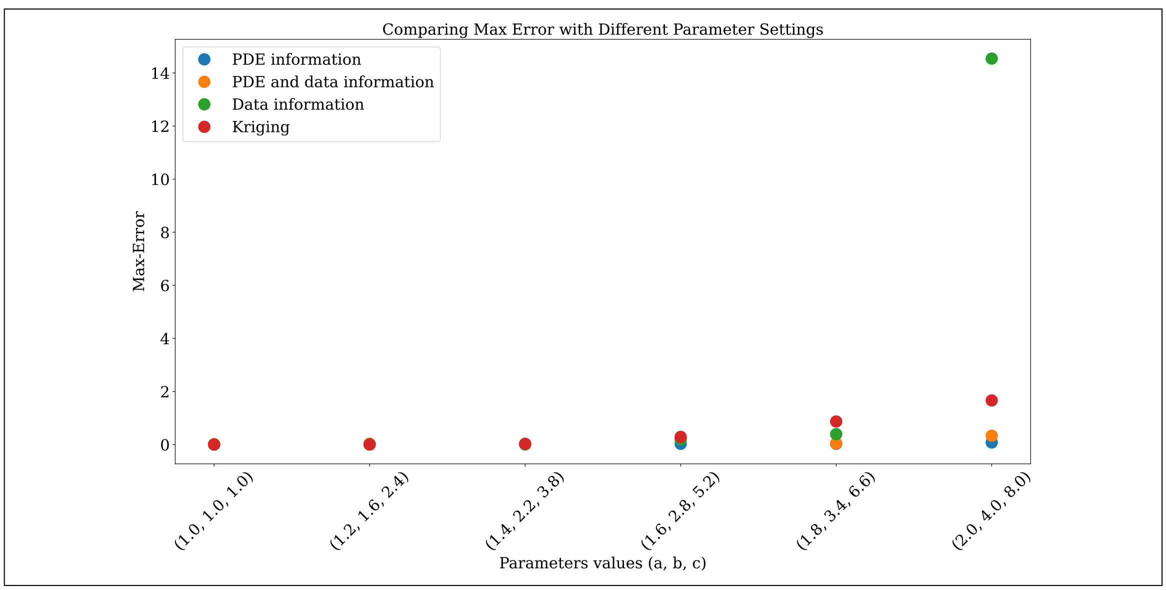

Figure 4.17. Max error of the different models with parameters of different values (a, b, c).

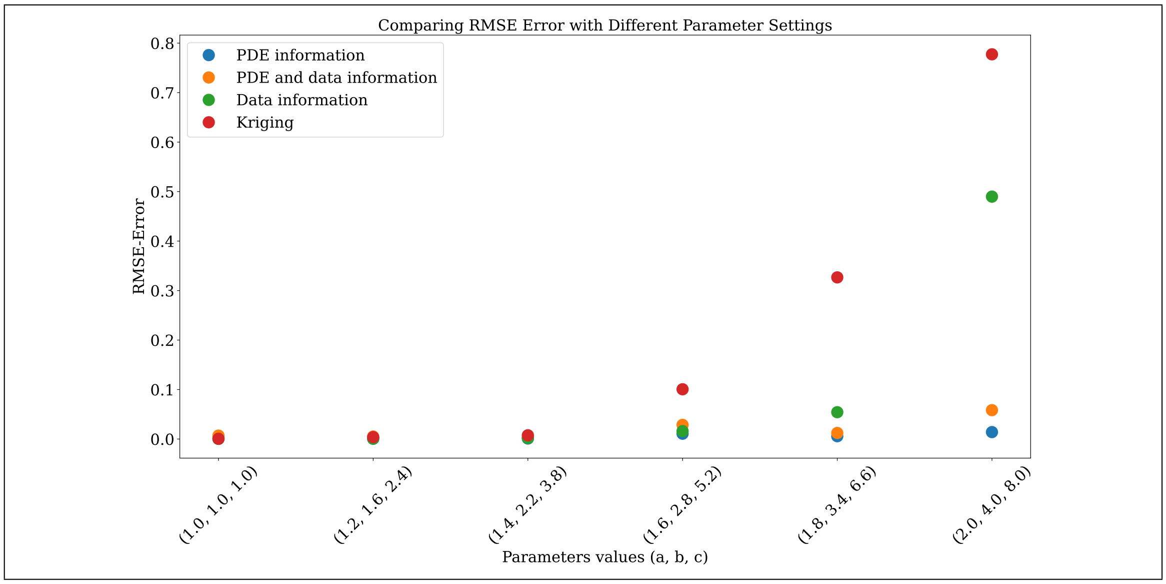

Figure 4.18. RMSE error of the different models with parameters of different values (a, b, c).

Figures 4.16, 4.17, and 4.18 represent the L_2, Max, and RMSE errors for different values of (a, b, c) obtained through various methods. The errors were measured for the PINN trained with the analytical solution information, the PINN trained with PDE and analytical solution information, the PINN just with the PDE information, and the Kriging method.

The results demonstrate that the network trained with PDE information and PDE/analytical solution information performed the most accurately for all parameters. However, the Kriging method showed the largest variance in performance. For the L_2 and RMSE metrics, the performance of the Kriging method tended to increase with increasing values of the parameters (c and a). This could be due to the fact that as the parameter values increase, the analytical solution has faster transportation and less smooth fields, which can make it more difficult for Kriging to interpolate effectively.

4.2.1. Computational Cost for Transport Equation

To continue studying the performance of models in different contexts, it is important to investigate both computation time and accuracy when different amounts of data are available. In this section, we focus on the transport formulation in equation (4.4) with parameters $a = 1$, $b = 1$, and $c = 1$.

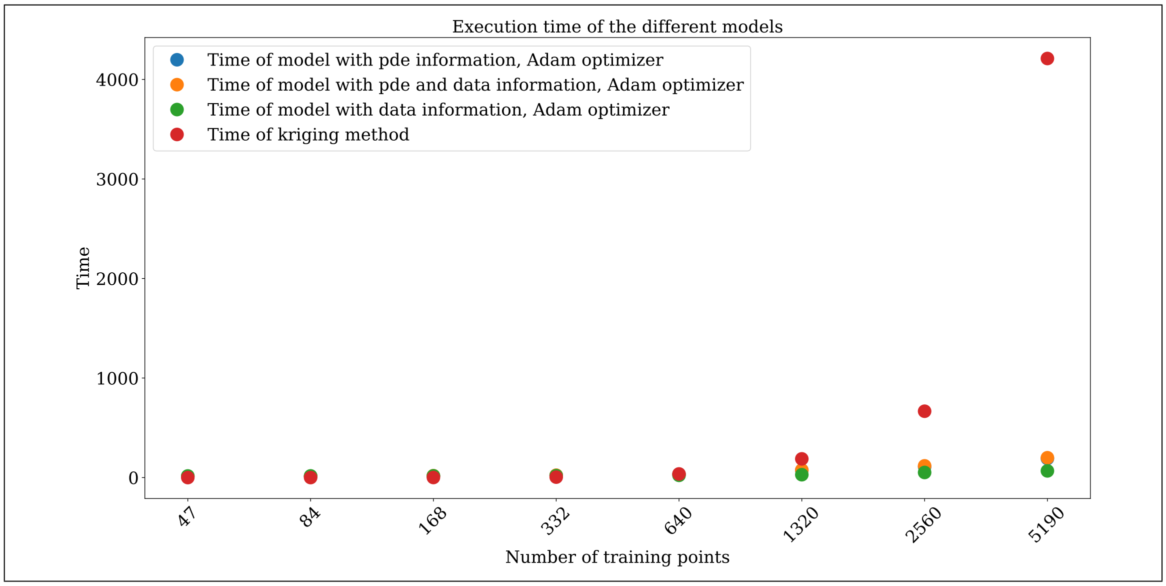

Figure 4.19. Execution time of various models as a function of the amount of data used for training. The time in the vertical axis is in seconds.

Figure 4.19 shows the training and prediction time when different amounts of data are available in the training process. As seen on the horizontal axis, in each training process, the number of points used was around the double of before and, on the vertical axis, the time has been measured in seconds.

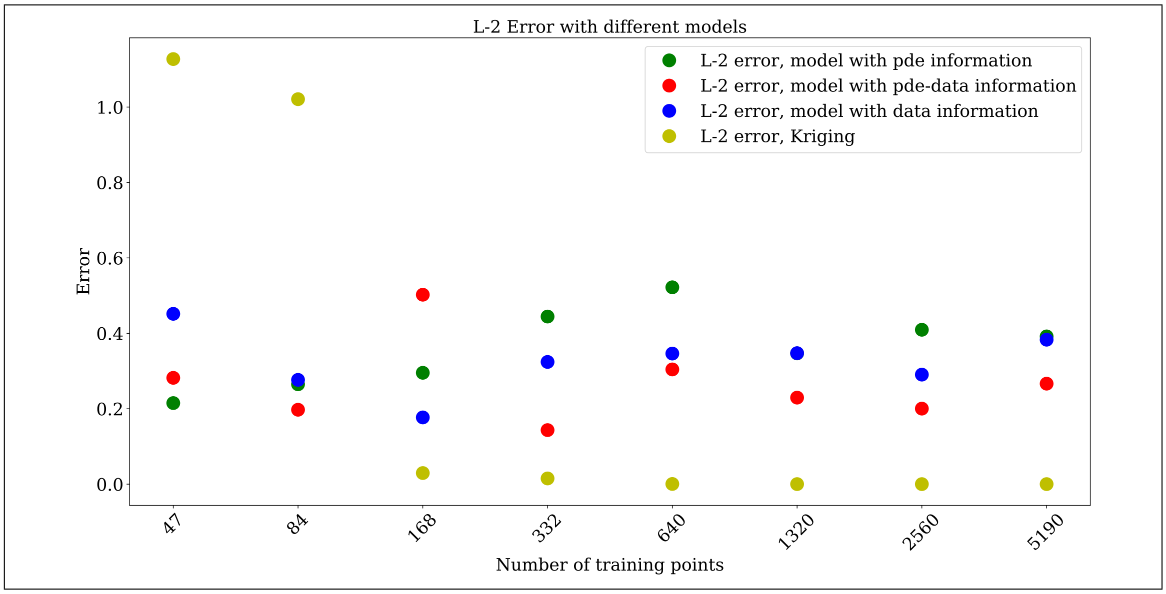

Figure 4.20. L_2 error of the different models with different amounts of data in the training process.

In Figure 4.20, we can see the L_2 error when a different amount of data is available in the training process.

In the case of neural networks, training times remain relatively constant, and the performance obtained is consistent even as the amount of data increases, due to the use of a fixed stopping criterion for the loss function during convergence. In contrast, for Kriging, as the amount of data increases, the training time increases considerably, but so does its accuracy. This is because, as explained in Section 2, the number of training points affects the dimensions of the matrix that must be inverted in the Kriging method.

4.2.2. Non-Uniform Grid in the Training

So far, all experiments have utilized observations uniformly distributed throughout the problem domain. However, it is crucial to examine how model performance varies when this feature is absent, as it is unrealistic. Therefore, to investigate this phenomenon, we will consider the following transport equation with periodic boundary conditions,

$$\begin{cases} \frac{\partial u}{\partial t} + 5\frac{\partial u}{\partial x} = 0, & x \in (0, 2\pi), t \in (0, 1), \\ u(x, 0) = \sin(x) + 2, \\ u(0, t) = u(2\pi, t). \end{cases} \tag{4.6}$$

It is important to notice that in this case, in the equation (4.6) the information given to the network is a periodic boundary condition. The exact solution to this problem corresponds to

$$u(x, t) = \sin(x - 5t) + 2. \tag{4.7}$$

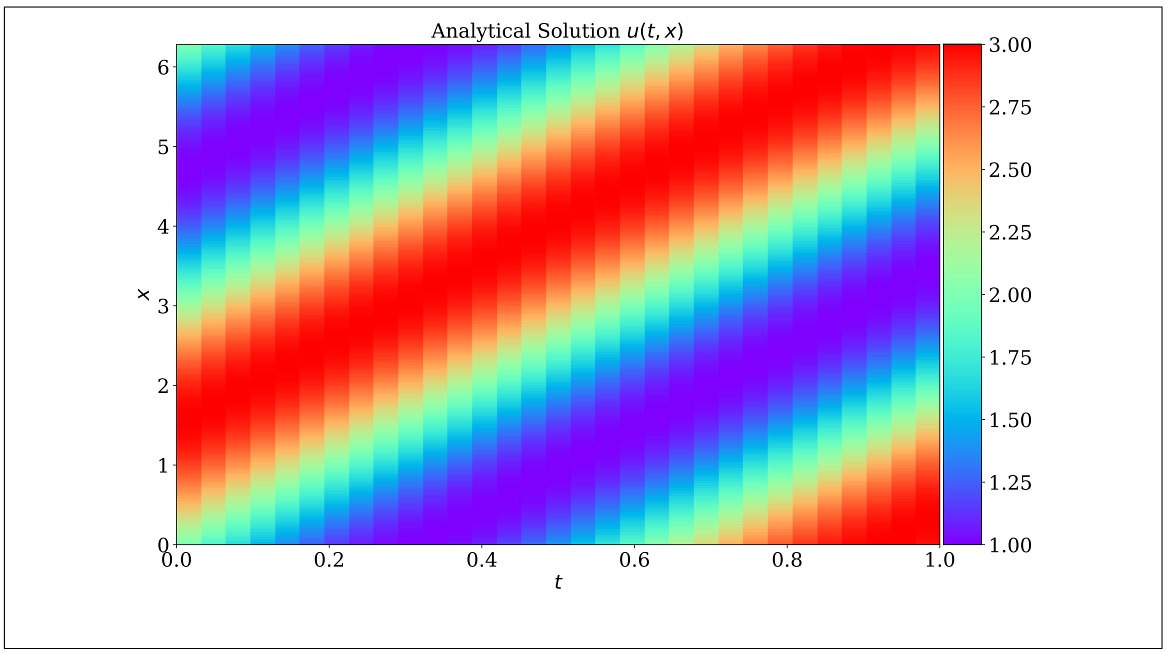

Figure 4.21. Solution of the transport problem (4.6).

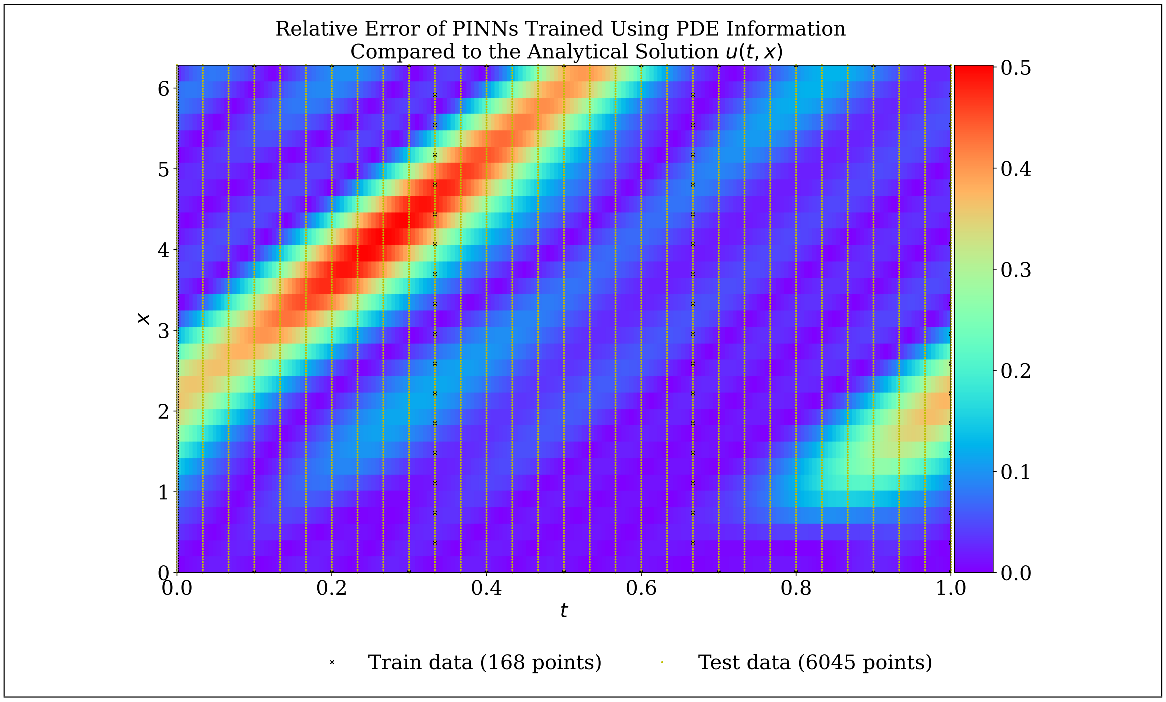

Figure 4.22. Relative error between PINN trained with PDE information and the analytical solution.

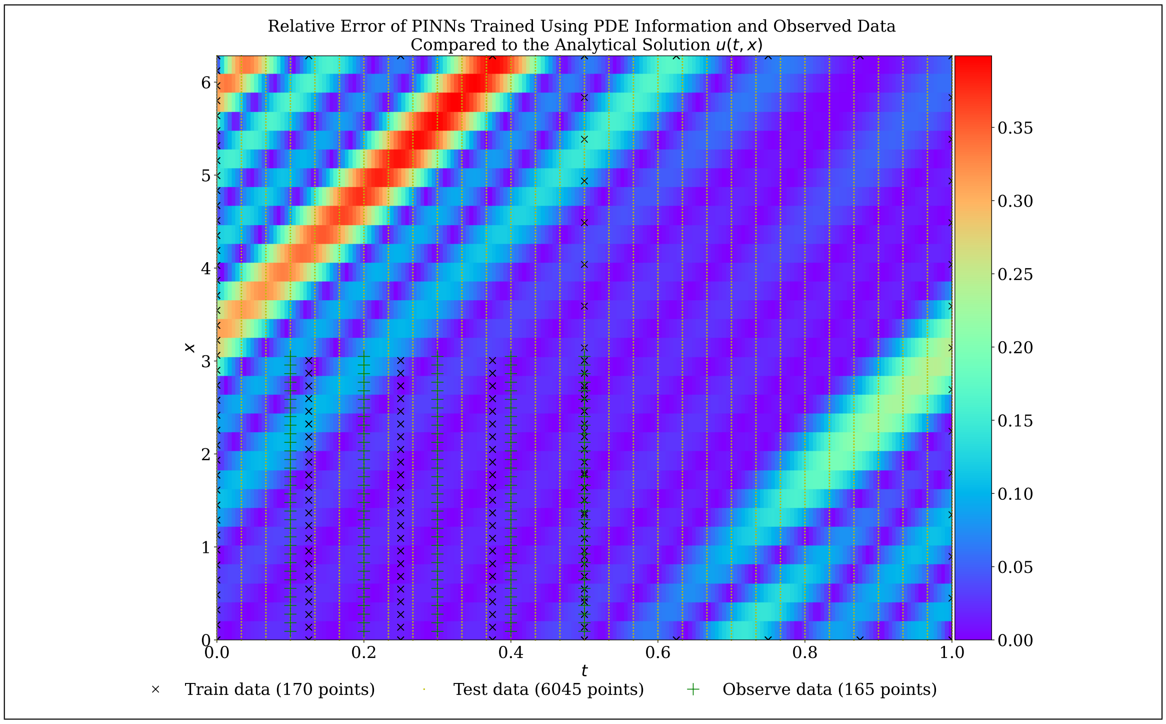

Figure 4.23. Relative error between PINN trained with PDE and analytical solution information and the analytical solution as a reference.

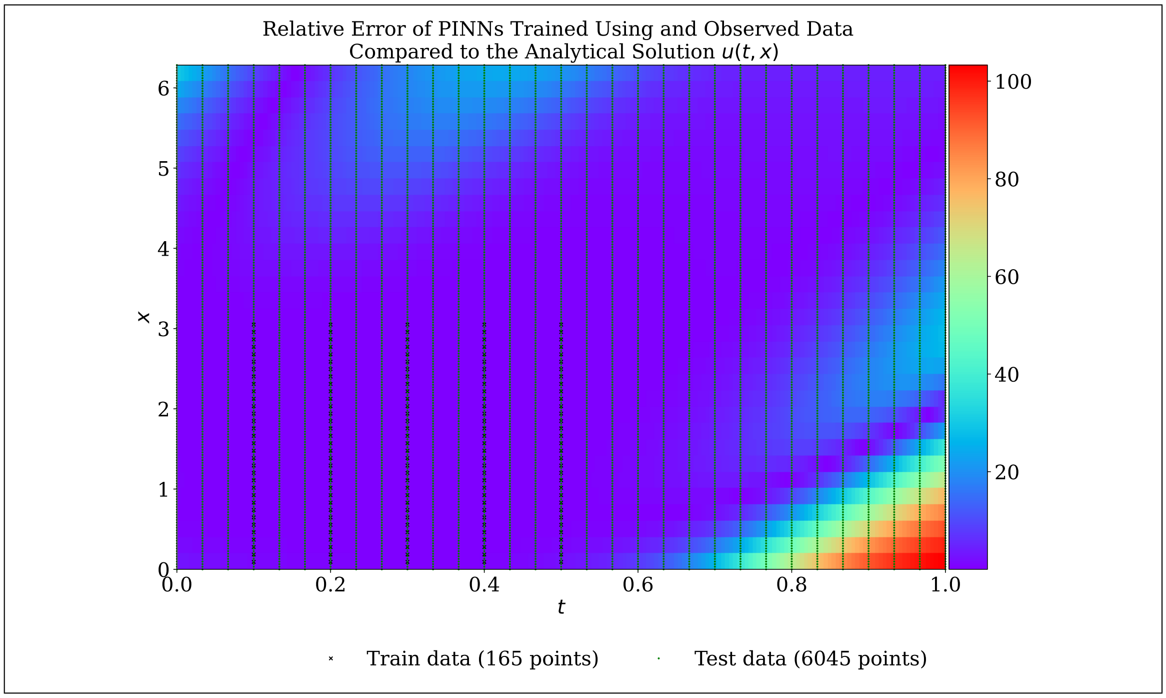

Figure 4.24. Relative error between PINN trained with analytical solution information and the analytical solution as a reference.

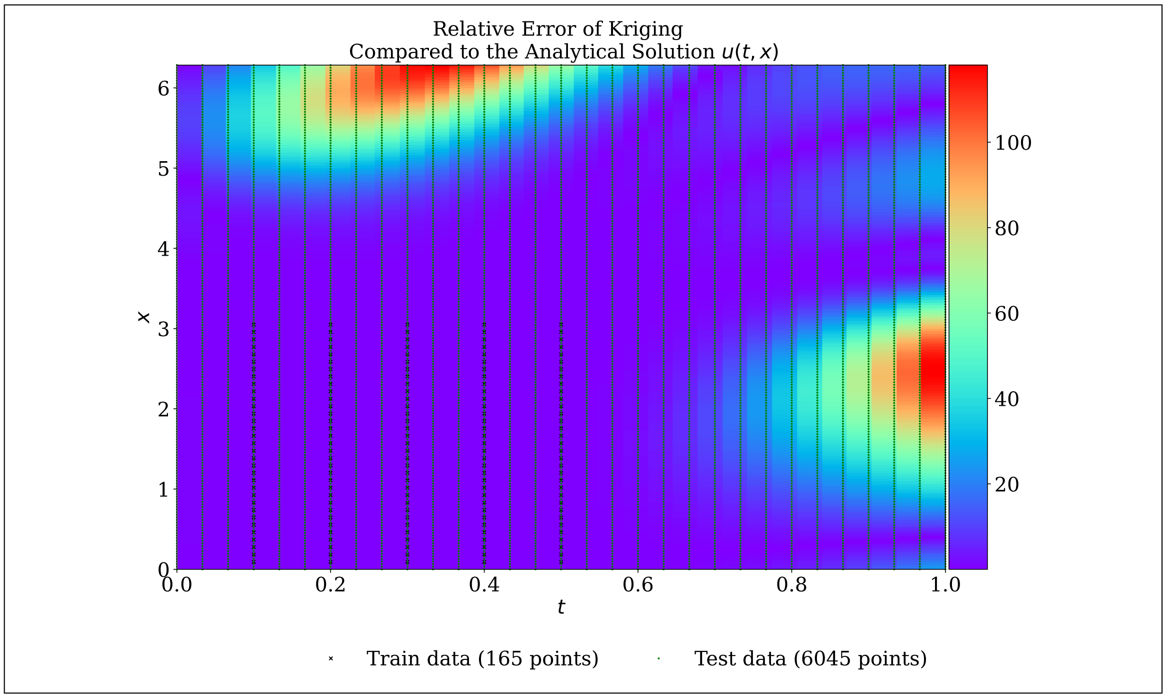

Figure 4.25. Relative error Kriging interpolation and the analytical solution as a reference.

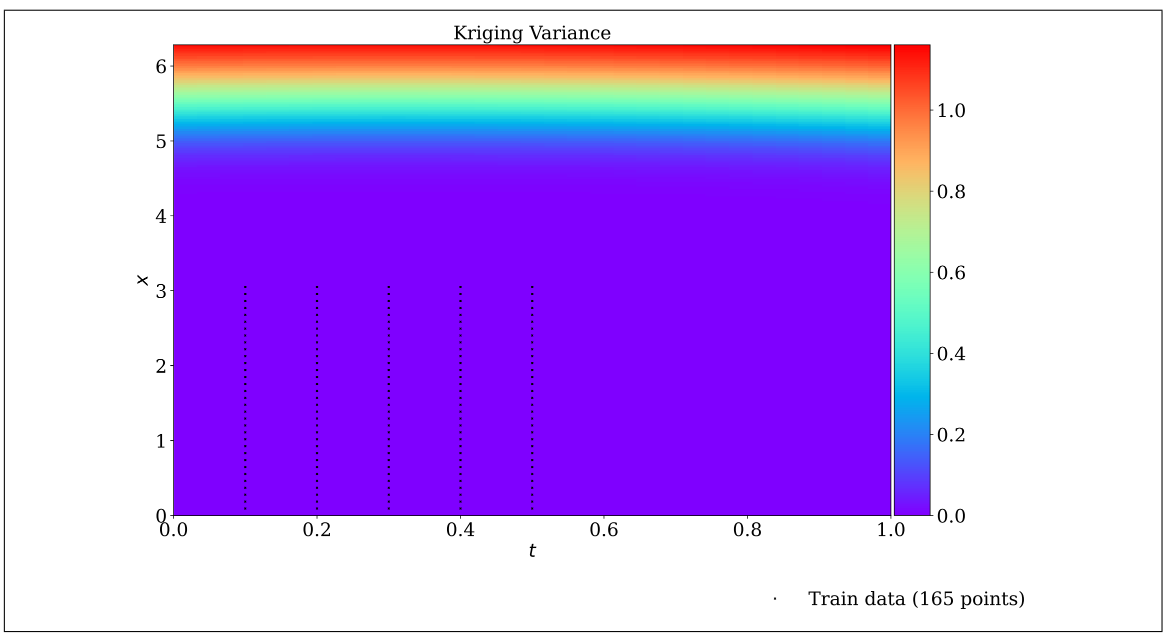

Figure 4.26. Variance of the kriging interpolation.

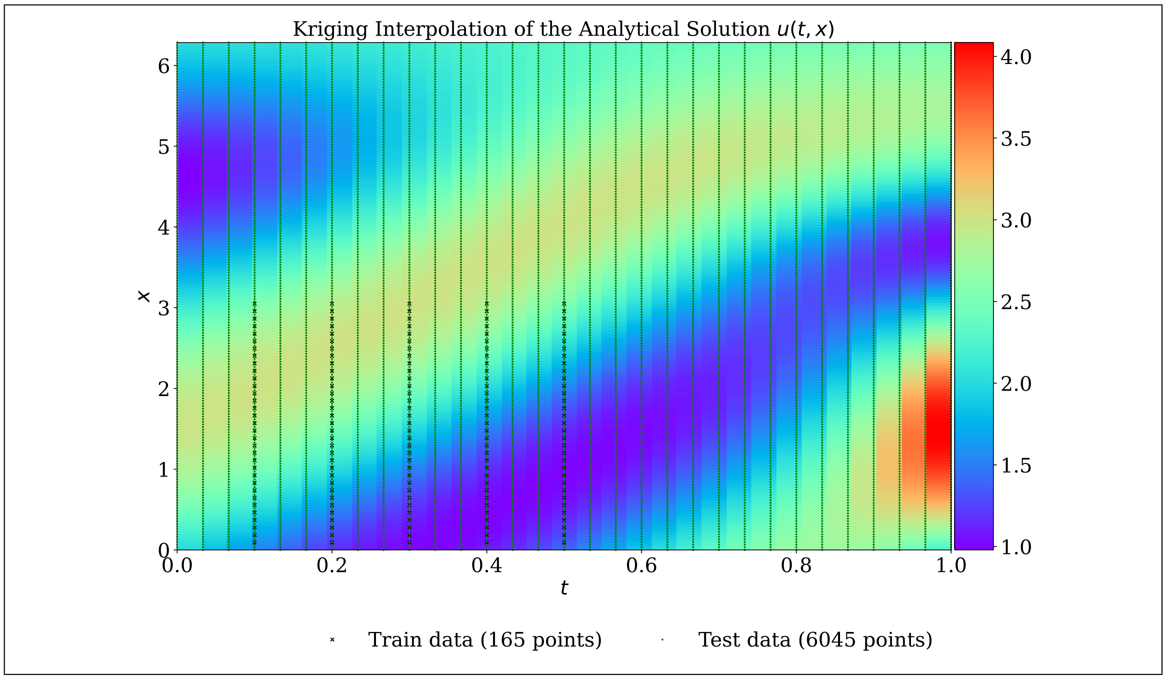

Figure 4.27. Kriging interpolation.

The solution of the transport equation is presented in Figure 4.21. The relative error between the analytical solution and the predictions of the network trained with different types of information is shown in three figures. Figure 4.22 shows the relative error between the analytical solution and the predictions of the network trained with the PDE information. Figure 4.23 shows the relative error between the analytical solution and the predictions of the network trained with both the PDE and data information. Finally, Figure 4.24 shows the relative error between the analytical solution and the predictions of the network trained only with data information.

In terms of the Kriging method, the relative error of the Kriging interpolation is shown in Figure 4.25, and the plot of the solution is shown in Figure 4.27. Finally, in Figure 4.26, the variance of the Kriging interpolation is shown. As expected, the interpolation of the Kriging method has a high value of variance in some positions where no close points are being interpolated.

Table 4.2. Comparison between the accuracy of PINNs and Kriging in approximating an analytic solution. The results were obtained using 6000 test points distributed uniformly throughout the problem domain. The first and second columns specify the number of points used for training.

Method	**Observed Data**	**PDE with BC and IC**	L_2**-norm**	Max**-norm**	**RMSE**	**Training Time**
Kriging	165	0	50.6208	1.9294	0.2959	1.064 s
PINNs	0	165	0.1529	0.0042	0.0010	16.83 s
PINNs with observed data	78	88	0.4256	0.0147	0.0029	8.3 s
Deep Learning	162	0	48.7239	3.1243	0.3425	3.735 s

Table 4.2 displays the performance of different methods. The L_2 norm indicates that the Kriging method has the largest error. The performance of the network trained with PDE and PDE-data information is similar, except for the training time. It is worth noting that the network with data information and the Kriging method were both used as extrapolation methods.

In conclusion, the results of the study suggest that the network trained with both PDE and data information provides the most accurate predictions of the transport equation. The L_2 norm values show that this network outperforms the other methods in approximating the analytic solution. However, Kriging interpolation can still provide reasonable approximations of the solution in regions where there are no close points for interpolation. Overall, the study highlights the potential of using PINN to improve the accuracy of numerical solutions of partial differential equations, especially when there are regions with few data points.

4.2.3. Number of Points in Loss Function Analysis

As explained in Chapter 3, PINNs are composed of different loss functions depending on the available information. To understand how the network works, this section trained the network with varying numbers of points in the PDE, boundary, and initial conditions.

Table 4.3. Number of points in the training process.

	Initial Conditions Points	**Boundary Conditions Points**	**PDE Points**
1°	40	40	1-30
2°	0-12	40	80
3°	40	0-12	80

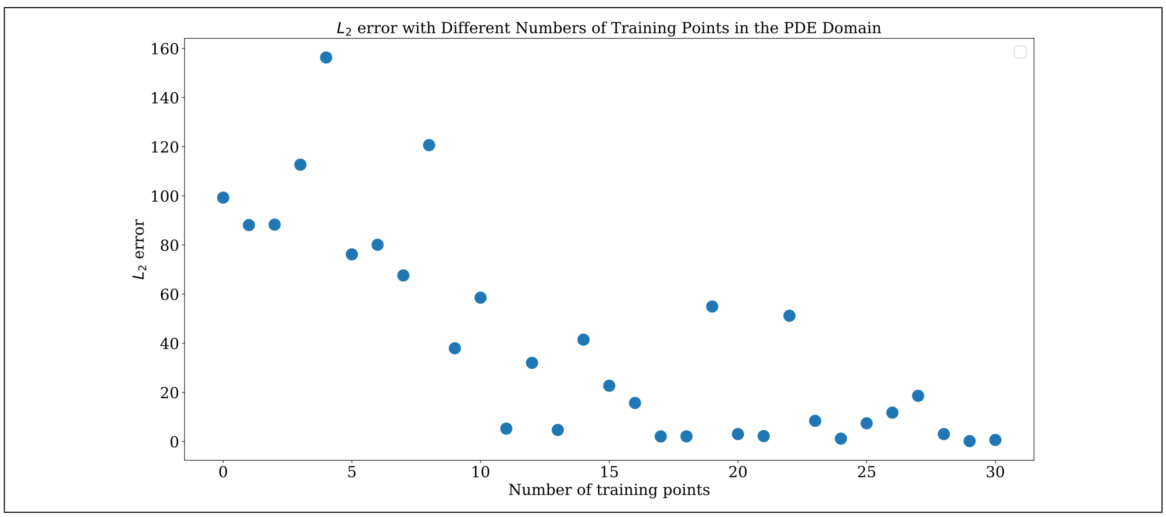

Figure 4.28. L_2 error of the PINNs trained with different amount of points in the PDE loss function.

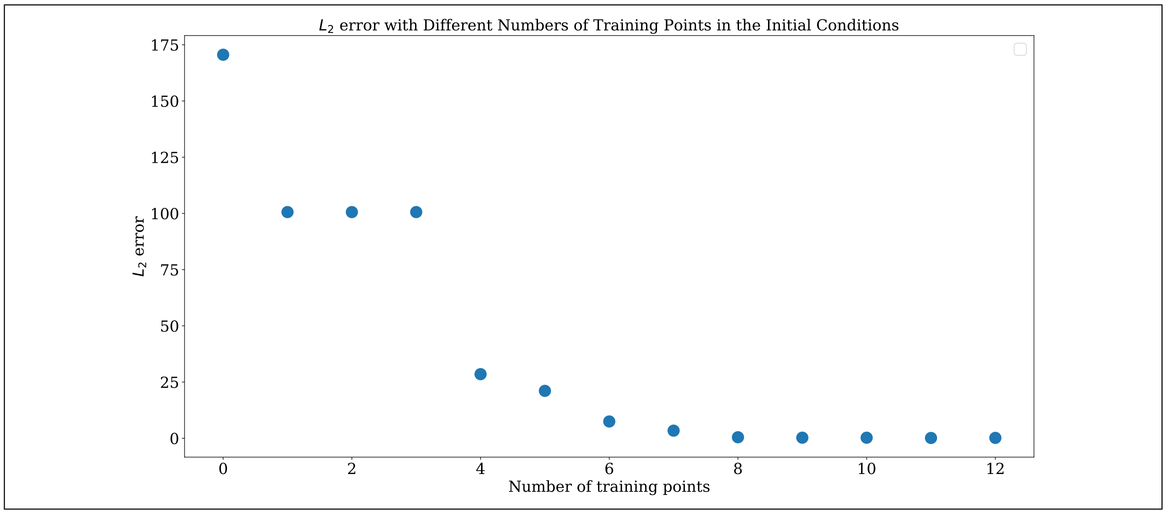

Figure 4.29. L_2 error of the PINNs trained with different amounts of points in the initial condition loss function.

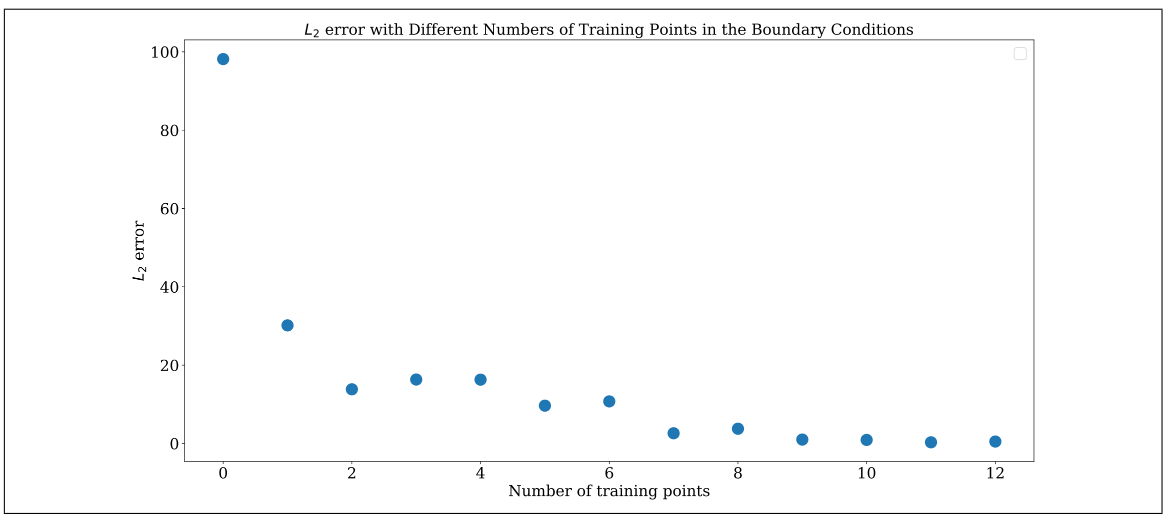

Figure 4.30. L_2 error of the PINNs trained with different amounts of points in the boundary condition loss function.

Table 4.3 summarizes the number of points used to train the network in Figures 4.28, 4.29, and 4.30, which correspond to changes in the number of training points for the PDE, initial conditions, and boundary conditions, respectively. The error convergence in Figure 4.28 appears to be slower than in Figures 4.29 and 4.30. For this particular problem, which corresponds to the system (4.6), it seems that more than 30 points are required to achieve convergence of the error. For the boundary and initial conditions, more than 10 points are needed. Hence, the PINN works well even in cases where few data points are available.

The generation of training points for the PDE is unlimited because they are generated from the equation that needs to be solved. In the case of the initial conditions, the points can be considered as observed data because they correspond to the points of the solution at $t = 0$.

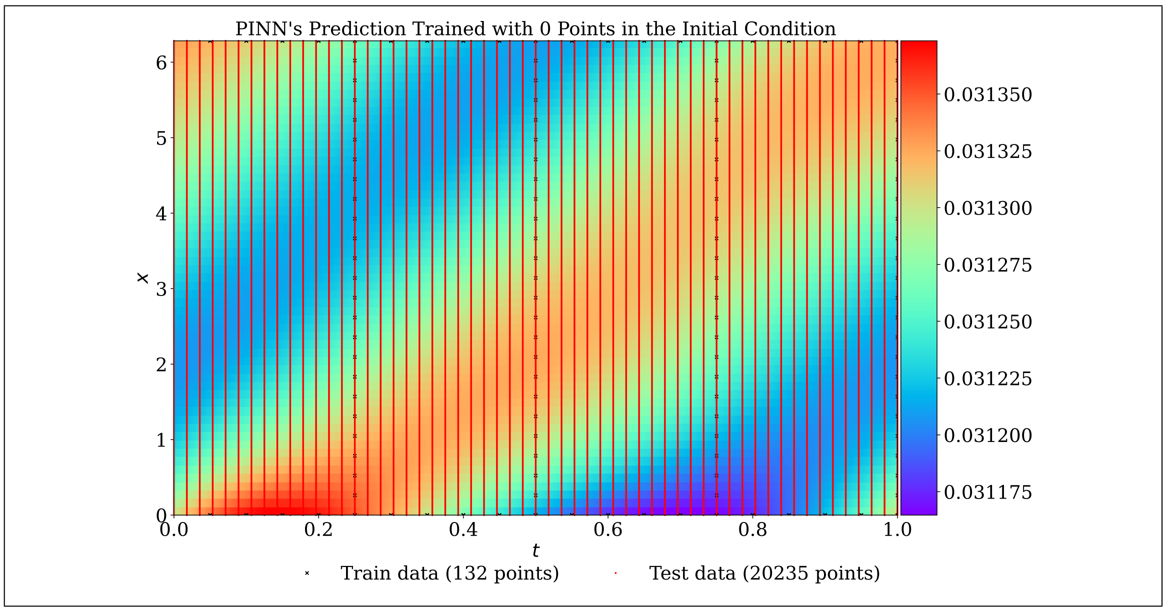

Figure 4.31. Prediction of the network trained with zero points in the initial condition.

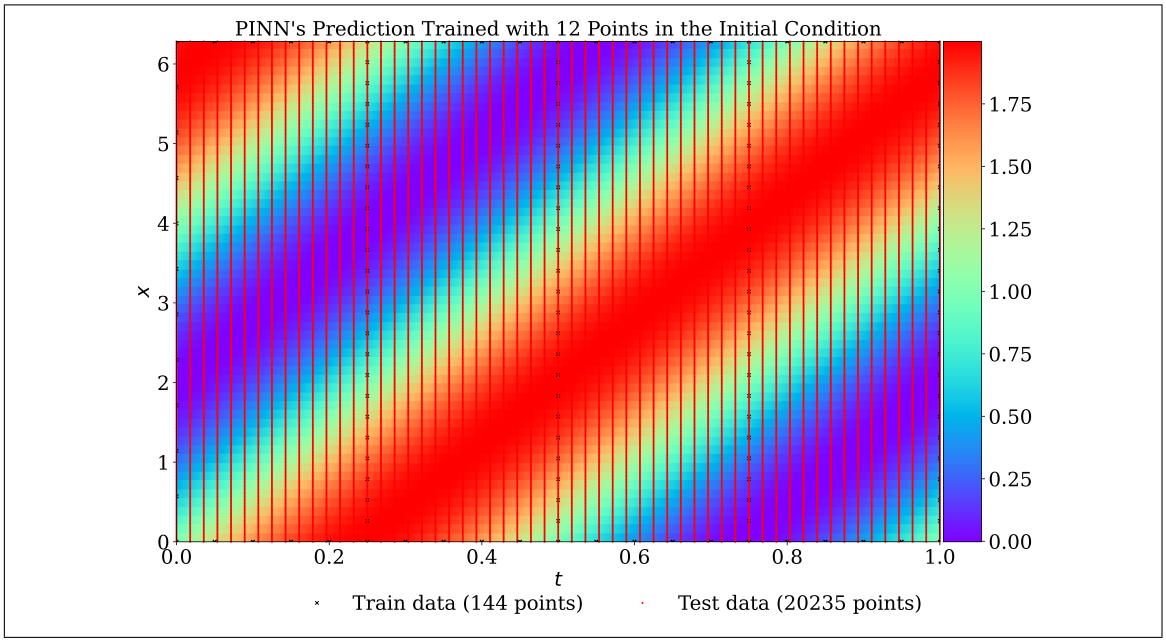

Figure 4.32. Prediction of the network trained with 12 points in the initial condition.

Figures 4.31 and 4.32 show that the network has the structure of the solution to some extent due to the PDE information. When some observation points are provided, it starts to learn the range of the function. Therefore, it is important to have sufficient points in the initial conditions, not only in the interior domain.

Sections 4.1 and 4.2 have investigated the performance of PINN and Kriging in various scenarios, such as non-uniform data information, computational cost as the amount of data increases, training with noisy data observations, and more. It is essential to note that the results presented here may vary depending on the hyperparameters of the network used and the Kriging model fitted. Therefore, we need to be careful in making conclusions based on one type of PDE only.

5. CONVECTION AND DIFFUSION EQUATIONS

In Chapter 4, the performance of PINN and Kriging in solving the transport and diffusion equations in 1D was studied through several benchmarks. In this chapter, we extend our study to 2D PDE problems, to understand the importance and behavior through different parameters of the PINN. Section 5.1 considers the Diffusion equation in 2D to investigate the importance of smoothness in the initial condition, while Section 5.2 focuses on the transport equation in 2D to compare the impact of periodic and Dirichlet boundary conditions (Section 5.2.1). Furthermore, Section 5.2.2 examines the effect of normalizing observed data during the training process.

5.1. Diffusion Equation in 2D: The Relevance of the Smoothness of the Initial Condition

The diffusion equation is a fundamental partial differential equation that describes how a quantity such as heat, mass, or concentration diffuses through a medium over time. In this chapter, we will study the performance of PINNs in solving the diffusion equation in 2D, focusing on the impact of the smoothness of the initial condition on the accuracy of the solutions.

To have an idea about the influence of the smoothness of the initial condition on the performance of the network, the problem to be studied is

$$\begin{cases} \frac{\partial u}{\partial t} &= D(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}), \qquad x \in (-2,2), y \in (-2,2), t \geq 0, \\ u(2,y,t) &= e^{\frac{-(4+y^2)}{4Dt}} \frac{1}{\sqrt{4\pi D t}}, \\ u(-2,y,t) &= e^{\frac{-(4+y^2)}{4Dt}} \frac{1}{\sqrt{4\pi D t}}, \\ u(x,2,t) &= e^{\frac{-(x^2+4)}{4Dt}} \frac{1}{\sqrt{4\pi D t}}, \\ u(x,-2,t) &= e^{\frac{-(x^2+4)}{4Dt}} \frac{1}{\sqrt{4\pi D t}}, \\ u(x,y,0) &= \delta(x,y), \end{cases} \tag{5.1}$$

where the initial condition corresponds to the Dirac delta $\delta(.)$ and D corresponds to the diffusion.

The exact solution is:

$$u(x, y, t) = e^{\frac{-(x^2+y^2)}{4Dt}} \frac{1}{\sqrt{4\pi Dt}}, \tag{5.2}$$

to avoid numerical issues when calculating the Dirac delta function on a mesh, we use the analytical solution evaluated at $t_0 > 0$ as the initial condition for the simulations. This approach ensures that the initial condition is smooth and avoids any singularities that may arise from the discretization of the Dirac delta function on the mesh, such that

$$u(x, y, t_0) = e^{\frac{-(x^2+y^2)}{4Dt_0}} \frac{1}{\sqrt{4\pi Dt_0}}. \qquad t_0 > 0 \tag{5.3}$$

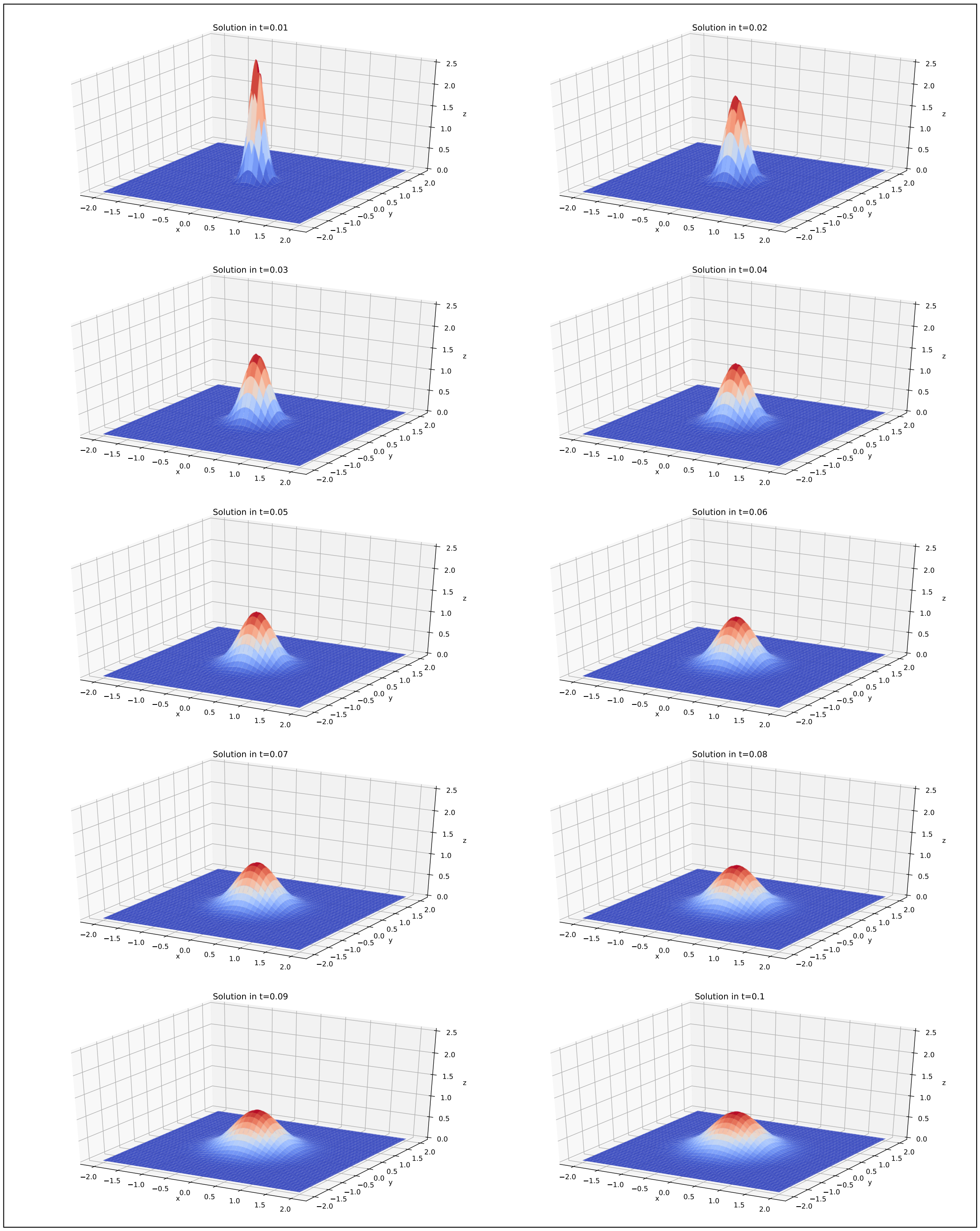

Figure 5.1. Different analytical solutions at different time levels, which will be used as initial conditions for the PINN.

To train each network, a time span from the t_0 to $t_0 + 2$ is used. The performance of the network is evaluated by measuring the L_2 relative difference at three different time points: t_0, $t_0 + 1$, and $t_0 + 2$.

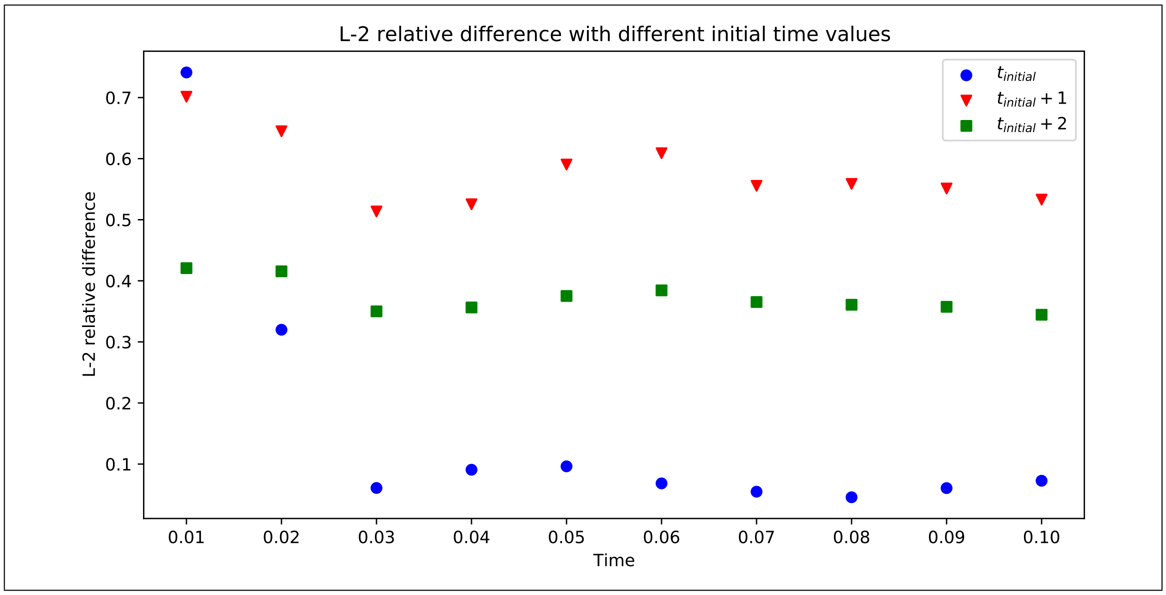

Figure 5.2. The L_2 relative difference of the PINN trained with the initial conditions shown in Figures 5.1 and is represented in the horizontal-axis. The performance of the network is evaluated by measuring the L_2 relative difference at three different time points: t_0, $t_0 + 1$, and $t_0 + 2$, which are represented by the blue circle, green square, and red triangle, respectively. L_2 relative difference is defined as the L_2 norm of the difference between the prediction and the analytical solution, divided in the L_2 norm of the analytical solution.

In Figure 5.2, the PINN approximates the function for the entire time interval, including t_0, and it measures the L_2 relative difference at three different time points: t_0, $t_0 + 1$, and $t_0 + 2$. We observe that the PINN trained with the initial condition at $t = 0.01$ exhibits the highest L_2 relative difference at the initial time. This indicates that the smoothness of the initial condition has a significant impact on the training process.

Moreover, it is worth noting that the L_2 error at $t_0 + 2$ remains relatively constant across the different initial conditions. This may be because by time $t_0 + 2$, the diffusion process has resulted in a nearly constant (i.e., averaged) solution, causing the solution to approach zero.

5.2. Transport Equation Time-Dependent 2D

The transport equation time-dependent 2D describes the movement of a substance or quantity through a medium. This equation is commonly used in various fields of science and engineering to model physical processes such as heat flow, pollutant dispersion, and wave propagation.

In this section, we will use the transport equation in 2D to study two subjects. In Section 5.2.1, we will examine the relevance of using periodic boundary conditions or the exact solution in the boundary conditions. Additionally, in Section 5.2.2, we will explore the importance of normalizing the domain and the solution space.

5.2.1. Benchmarking Periodic Boundary Conditions

The objective of this section is to compare the performance of the PINN when using periodic and exact solutions in boundary conditions at different times.

The problem to solve with the periodic boundary conditions is:

$$\begin{cases} \frac{\partial u}{\partial t} + k\frac{\partial u}{\partial x} &= 0, \qquad x \in (-2,4), y \in (-2,4), t \geq 0 \\ u(-2,y,t) &= u(4,y,t), \\ u(x,y,0) &= e^{-(x-1)^2-(y-1)^2}, \end{cases} \tag{5.4}$$

where k corresponds to the velocity in the transport and in this case is 2. The exact solution is:

$$\begin{aligned} u(x,y,t) &= u(\text{mod}(x-kt-1,6),y,0), \\ &= u_0(\text{mod}(x-kt-1,6),y), \\ &= e^{-(\text{mod}(x-kt-1,6))^2-(y-1)^2}. \end{aligned} \tag{5.5}$$

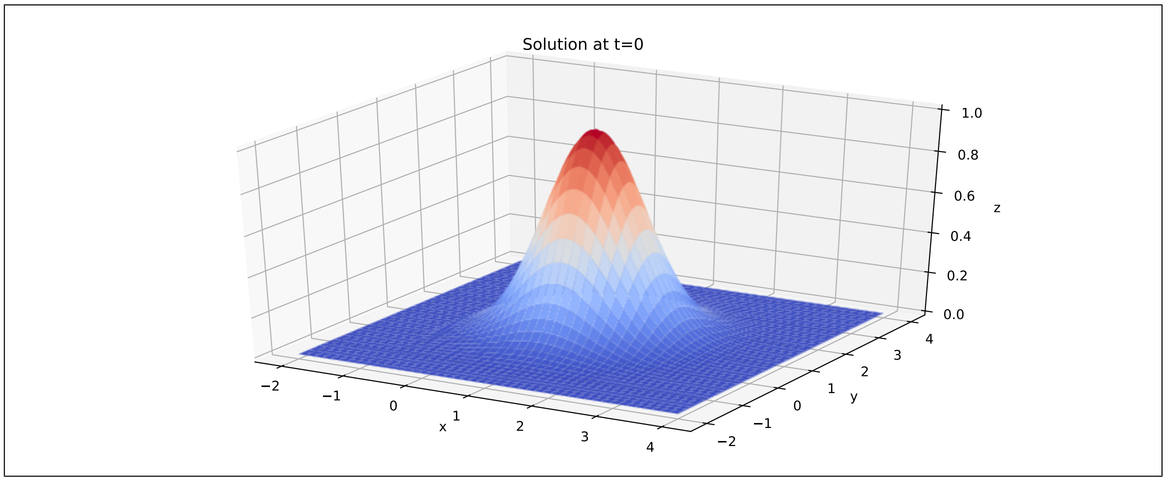

Figure 5.3. Initial condition of the equation (5.4) which corresponds to a bivariate normal distribution.

It is worth to notice that the PDE studied requires one boundary condition in the x direction and none in the y direction, since there is a first derivative in x and no derivative in y. However, because we know the analytical solution, we can include observations on the boundary of the domain in the PINN implementation. Although no boundary conditions in y are strictly necessary, we specify the analytical solution at $y = -2$ and $y = 4$ in the PINN to prevent any potential issues with the neural network "leaking" at the y boundaries. These additional constraints act as boundary conditions in the PINN and help ensure accurate predictions near the y boundaries.

We compare the results obtained from the PINN trained with periodic boundary conditions to the results obtained from the PINN trained with the exact solution at the borders,

$$\begin{cases} u(-2, y, t) & = e^{-(\mathrm{mod}(-2-kt-1,6))^2-(y-1)^2}, \\ u(4, y, t) & = e^{-(\mathrm{mod}(4-kt-1,6))^2-(y-1)^2}. \end{cases} \tag{5.6}$$

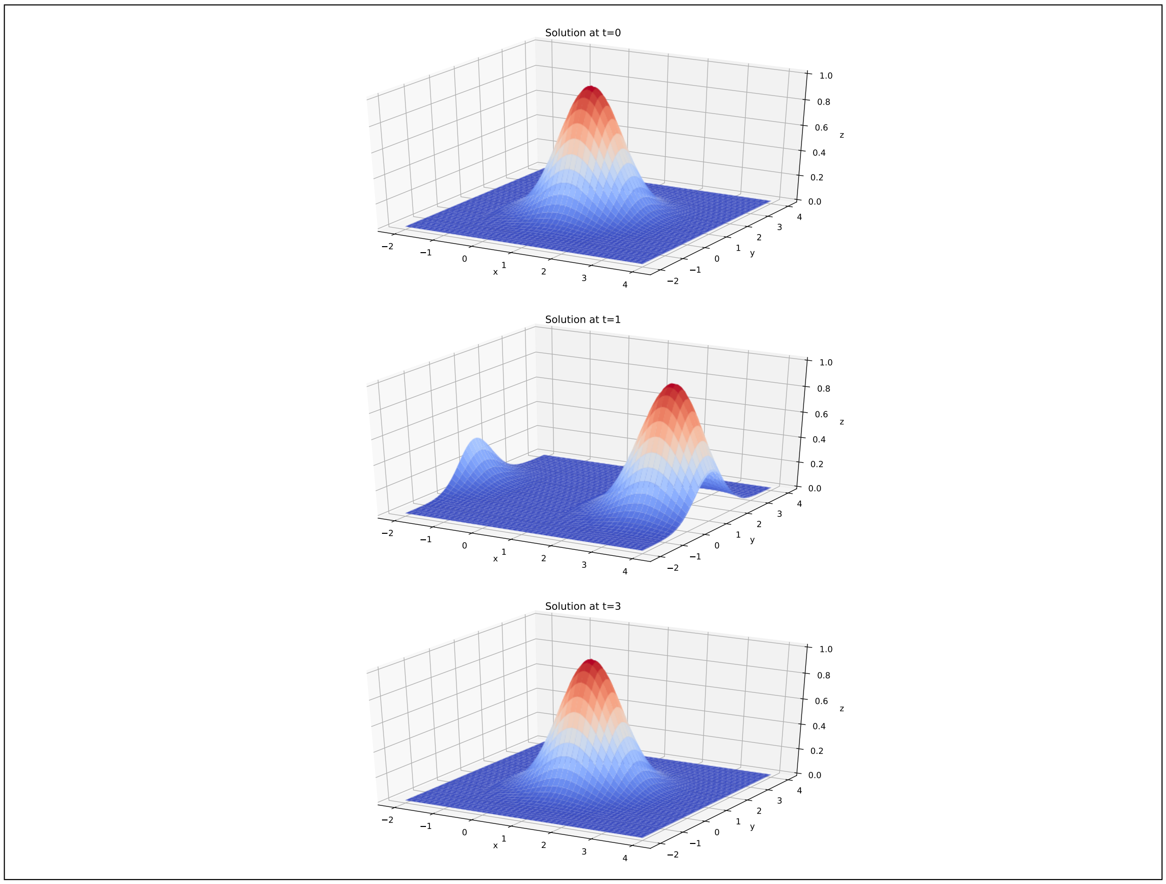

Figure 5.4. Representation of the exact solution in equation (5.5) at times $t = 0$, $t = 1$ and $t = 3$.

The exact solution at $t = 0$, $t = 1$ and $t = 3$ are shown in Figure 5.4 it worth mentioning that the range of the domain in the transport axis is 6, then since $k = 2$, the initial condition is equivalent to the solution at $t = 3$.

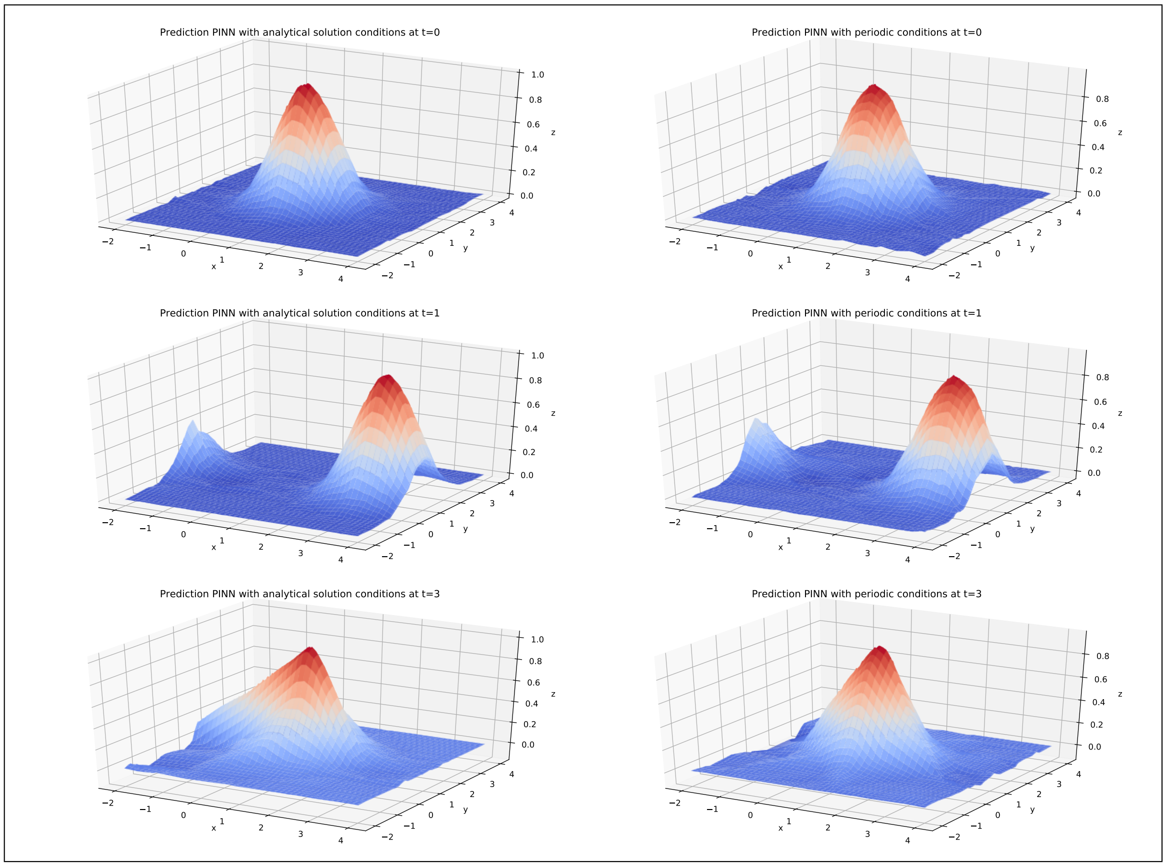

Figure 5.5. PINN prediction of the problem. On the left, PINN trained with the analytical solution at the boundaries, on the right PINN trained with periodic boundary conditions.

Figure 5.5 on the left shows the prediction of the network trained with the analytical solution at the border, while the figure on the right displays the PINN trained with periodic boundary conditions. It can be observed that in both cases, the solutions are visually similar for the predictions at $t = 0$ and $t = 1$. However, at $t = 3$, some differences can be appreciated between the two predictions.

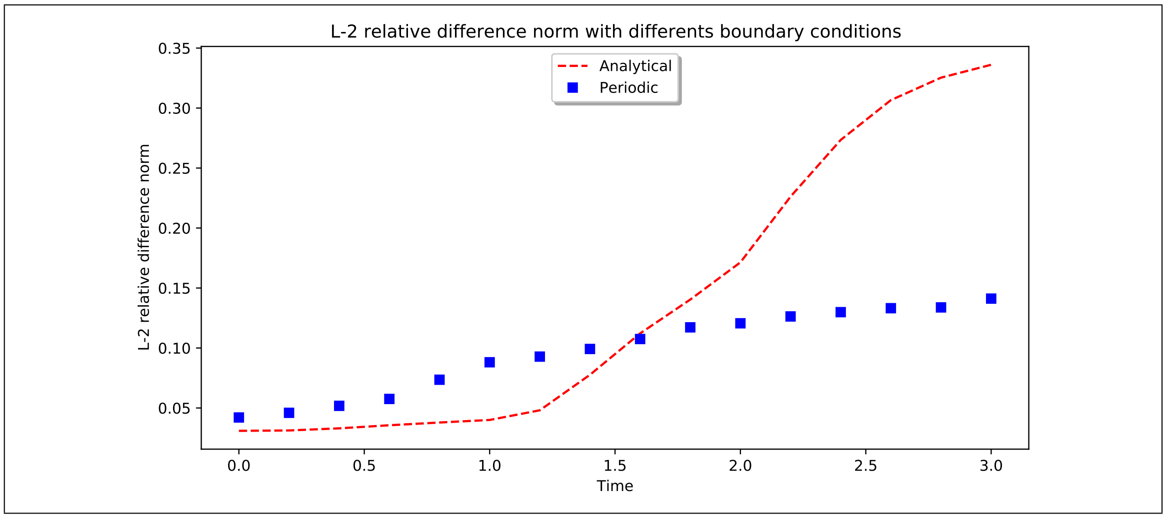

Figure 5.6. This figure shows the L_2 relative difference norm between the prediction of the PINN and the analytical solution for problem (5.5). The horizontal axis represents the time when the error is measured, while the vertical axis shows the L_2 error. The blue squares represent the error of the PINN trained with periodic boundary conditions, while the red line shows the error of the PINN trained with the analytical solution.

Figure 5.6 compares the L_2 relative error between the PINN's predictions and the exact solution for different times using two different boundary conditions. Both boundary conditions yield similar performance, but the error obtained with periodic boundary conditions is more stable over time. In this study, the PINN has been shown to be particularly accurate for the transport equation with periodic boundary conditions and outperforms alternative implementations in this benchmark, especially when information about the analytical solution is not available. This situation can arise in both academic and real-world problems.

5.2.2. Analysis of the Normalized Problem

To gain insight into whether the range of the domain and the value of the function affect the network's learning performance, a comparison is presented between two problems with different domain ranges and function values.

The first problem defined is:

$$\begin{cases} \frac{\partial u}{\partial t} + k\frac{\partial u}{\partial x} &= 0, \qquad x \in (-180, 180), y \in (-90, 90), t \geq 0 \\ u(-180, y, t) &= u(180, y, t), \\ u(x, y, 0) &= 90e^{\frac{-(x)^2-(y)^2}{5000}}, \end{cases} \tag{5.7}$$

where $k = 2$ and the exact solution is:

$$\begin{aligned} u(x, y, t) &= u(\mathrm{mod}(x - kt, 360), y, 0), \\ &= u_0(\mathrm{mod}(x - kt, 360), y), \\ &= 90e^{\frac{-\mathrm{mod}(x-kt)^2-(y)^2}{5000}}. \end{aligned} \tag{5.8}$$

On the other hand, the normalized problem is:

$$\begin{cases} \frac{\partial u}{\partial t} + k\frac{\partial u}{\partial x} &= 0, \qquad x \in (-2, 2), y \in (-1, 1), t \geq 0 \\ u(-2, y, t) &= u(2, y, t), \\ u(x, y, 0) &= e^{\frac{-(x)^2-(y)^2}{0.5}}, \end{cases} \tag{5.9}$$

where $k = 2$ and the exact solution is:

$$\begin{aligned} u(x, y, t) &= u(\mathrm{mod}(x - kt, 4), y, 0), \\ &= u_0(\mathrm{mod}(x - kt, 4), y), \\ &= e^{\frac{-\mathrm{mod}(x-kt)^2-(y)^2}{0.5}}. \end{aligned} \tag{5.10}$$

To ensure a fair comparison, we trained each neural network using the same hyperparameters. Specifically, we used 32 neurons, 3 hidden layers, ReLU as the activation function, Glorot uniform initializer, Adam optimizer, and a learning rate of 0.01. Additionally, we included boundary conditions in the y axis as additional constraints during the training process for both cases.

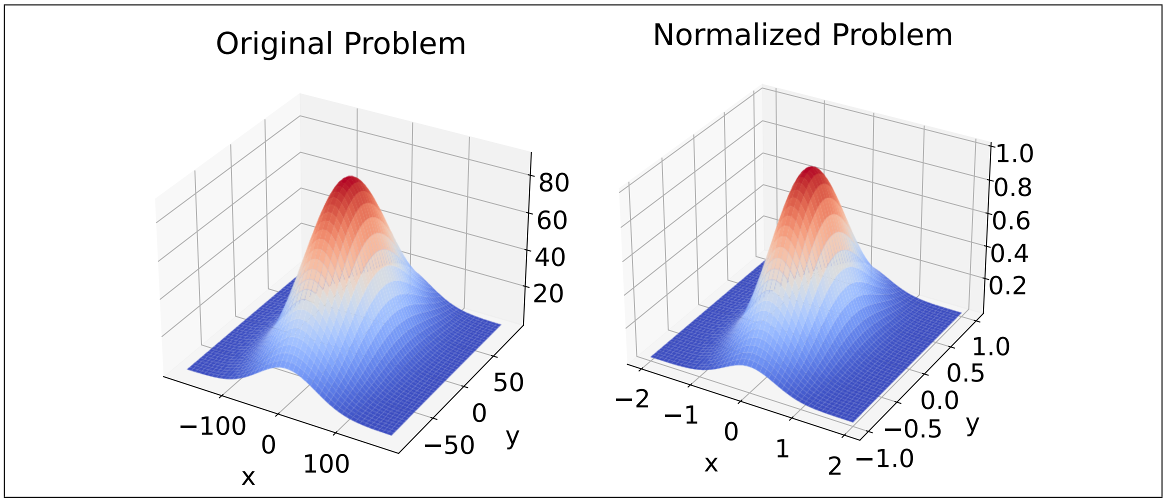

Figure 5.7. The initial conditions for the two problems are shown in the figure. On the left, the initial condition for problem (5.7) is displayed, while on the right, the initial condition for the problem defined in equation 5.9 is shown.

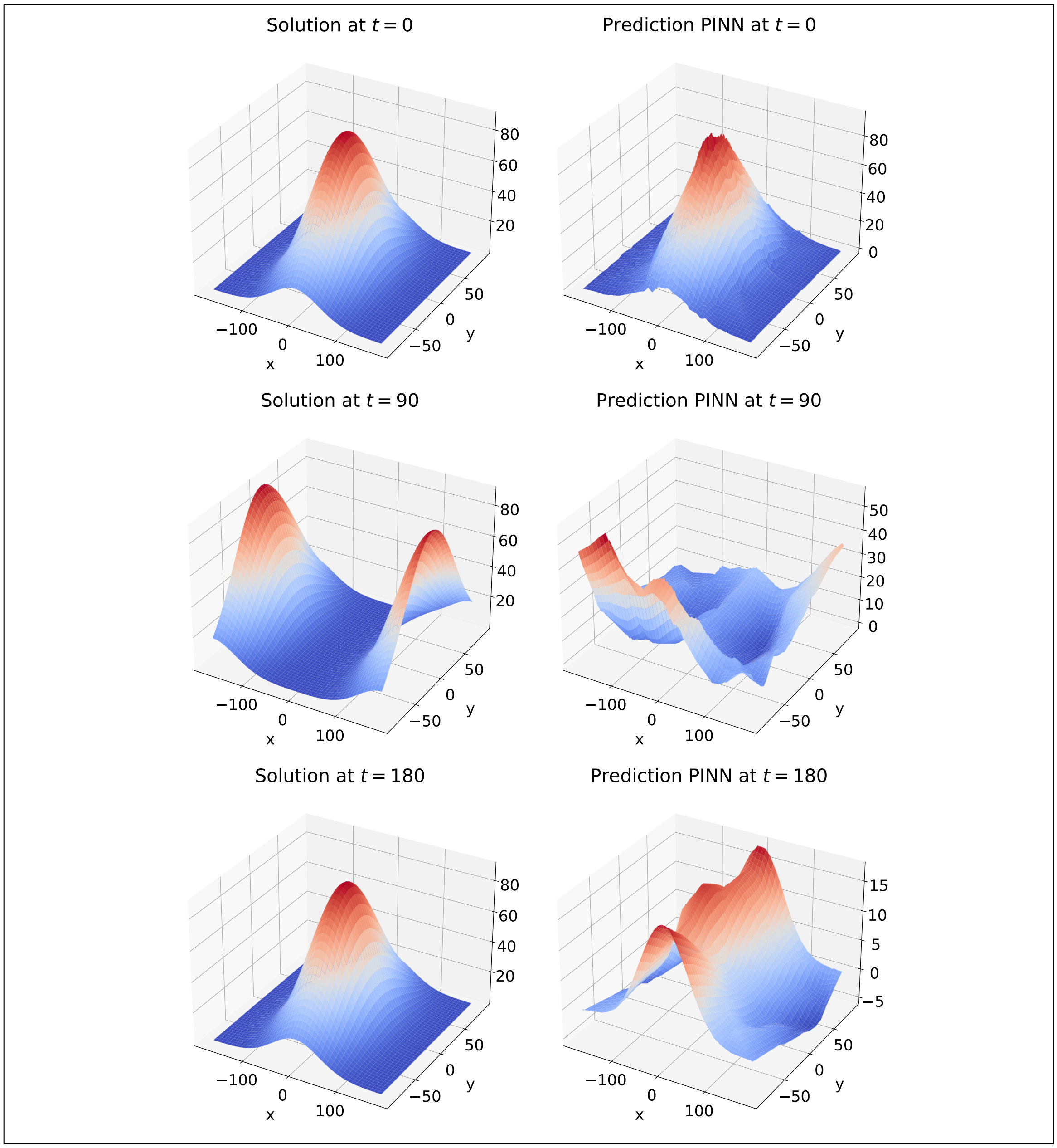

Figure 5.8. The comparison between the analytical solution and PINN prediction for problem (5.7) is presented in the figure, at times $t = 0, 90, 180$. The analytical solution is displayed on the left, while the PINN's prediction is shown on the right.

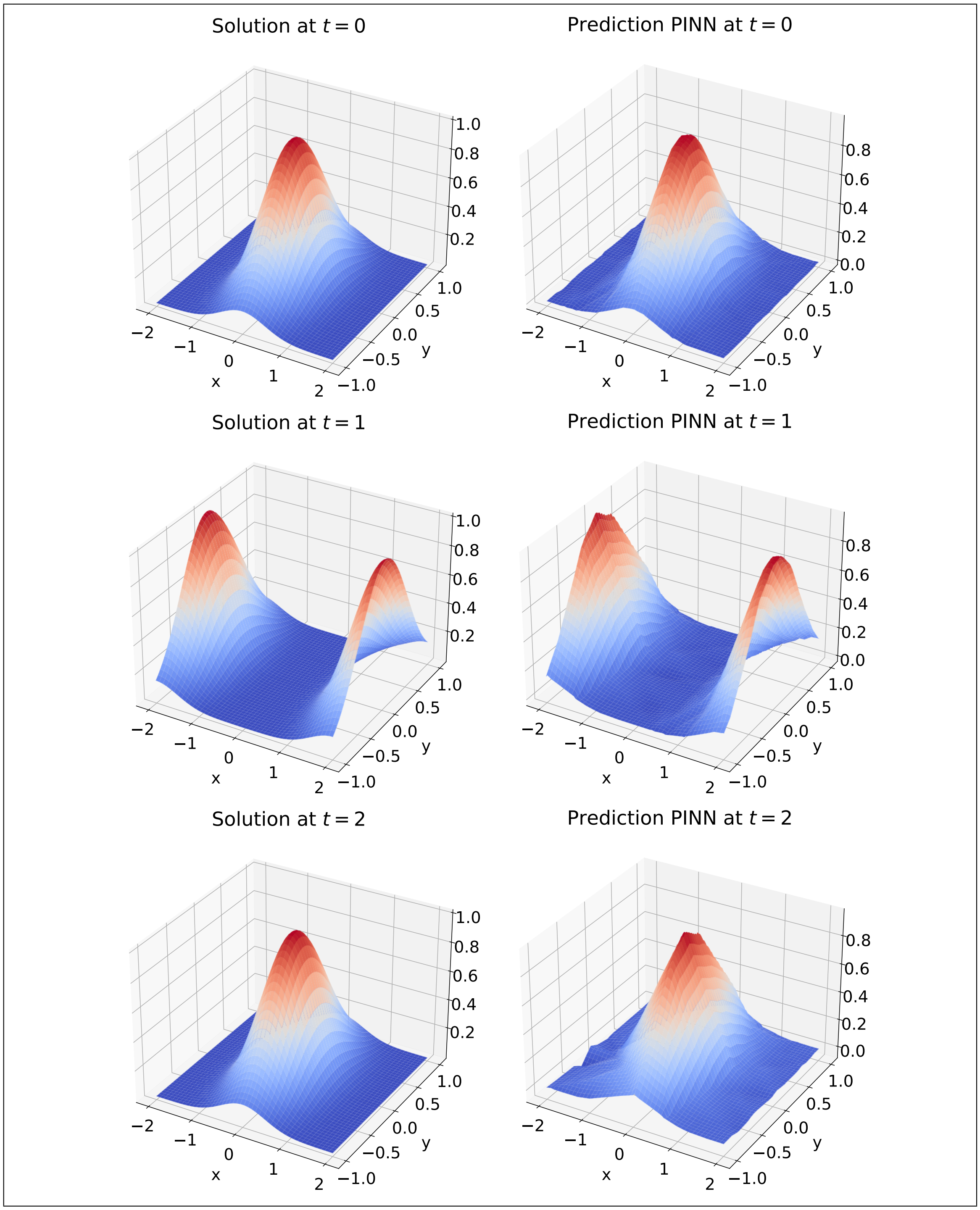

Figure 5.9. The comparison between the analytical solution and PINN prediction for problem (5.9) is presented in the figure, at times $t = 0, 1, 2$. The analytical solution is displayed on the left, while the PINN's prediction is shown on the right.

Figure 5.8 shows the prediction of the network trained to solve the problem (5.7) and Figure 5.9 corresponds to the network trained to solve the problem (5.9). It is easy to observe that the predictions given by the PINN trained in the normalized domain are more accurate.

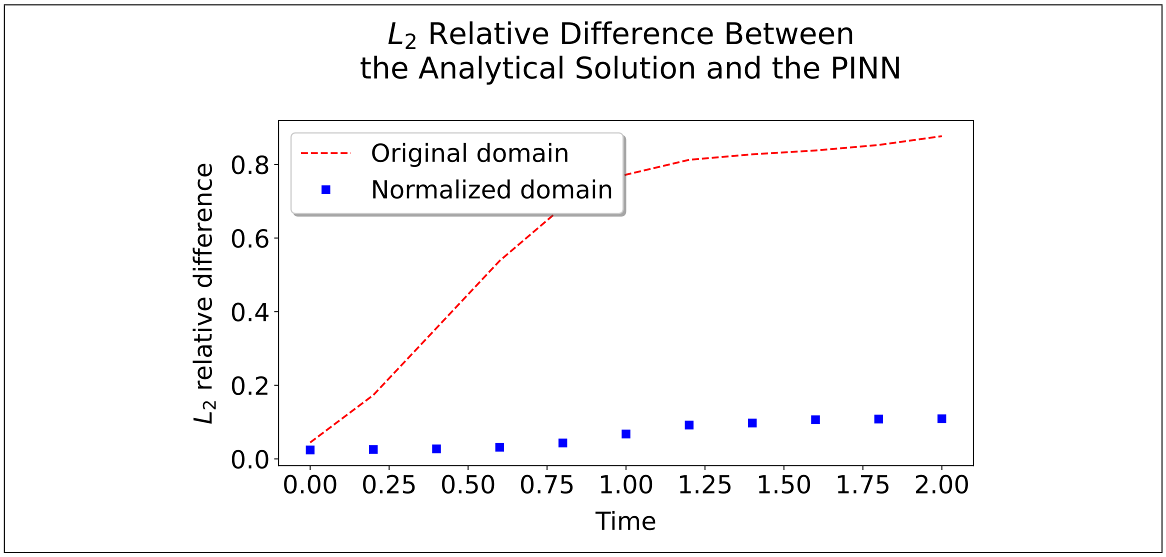

Figure 5.10. This figure shows the L_2 relative difference norm between the prediction of the PINN and the analytical solution for problem (5.7) and (5.9). The horizontal axis represents the time when the error is measured, while the vertical axis shows the L_2 error. The red lines show the error of the PINN trained with the original domain, while the blue squares represent the error of the PINN trained in the normalized domain.

Figure 5.10 clearly shows that the normalized domain performs much better than the PINN trained in the original domain in predicting the transport process over different periods. This can be attributed to problems such as vanishing gradients, where the gradient values become very small when the domain range is too large, making it difficult for the weights to update correctly. As a result, the PINN trained in the normalized domain is more accurate in predicting the transport process.

6. DUST FLUXES DATASET

Last Glacial Maximum (LGM) refers to the period during the last glacial cycle when ice masses reached their last maximum global extent, occurring approximately 26,000 to 19,000 years ago [Hughes et al., 2013]. The Holocene is the current interglacial period that began approximately 11,700 years ago and has sustained modern human society's growth and development [Wanner et al., 2011].

Dust is important because it influences the climate system through its impact on radiative forcing, chemical reactions with other atmospheric constituents, and acting as a source of nutrients to biological systems. While the role of dust in the climate system is not well understood in quantitative terms, changes in atmospheric dust loading have the potential to have a significant impact on future climate changes. Therefore, the role of dust in climate change has become a major focus of earth system modeling research. Dust records can be obtained from ice cores, marine sediments, and terrestrial deposits, which document changes in clay mineralogy, isotopes, grain size, and concentration of terrigenous materials [Kohfeld and Harrison, 2001].

The simulations from Community Earth System Model (CESM) include the periods of LGM and Holocene. Dust emissions were modeled using a representation of the major components of the dust cycle [Albani et al., 2016]. These types of complex systems can be performed in clusters or supercomputers.

From this part of the study, we used two datasets for dust records in Holocene and Last Glacial Maximum (LGM) times. The observed data are the measurements and identified by [Lambert et al., 2015] which include the DIRTMAP database, and simulated data by [Albani et al., 2016]. The observed data consists of 397 and 317 measurements for the Holocene and LGM, respectively. For the simulated data we used 55296 for both Holocene and LGM.

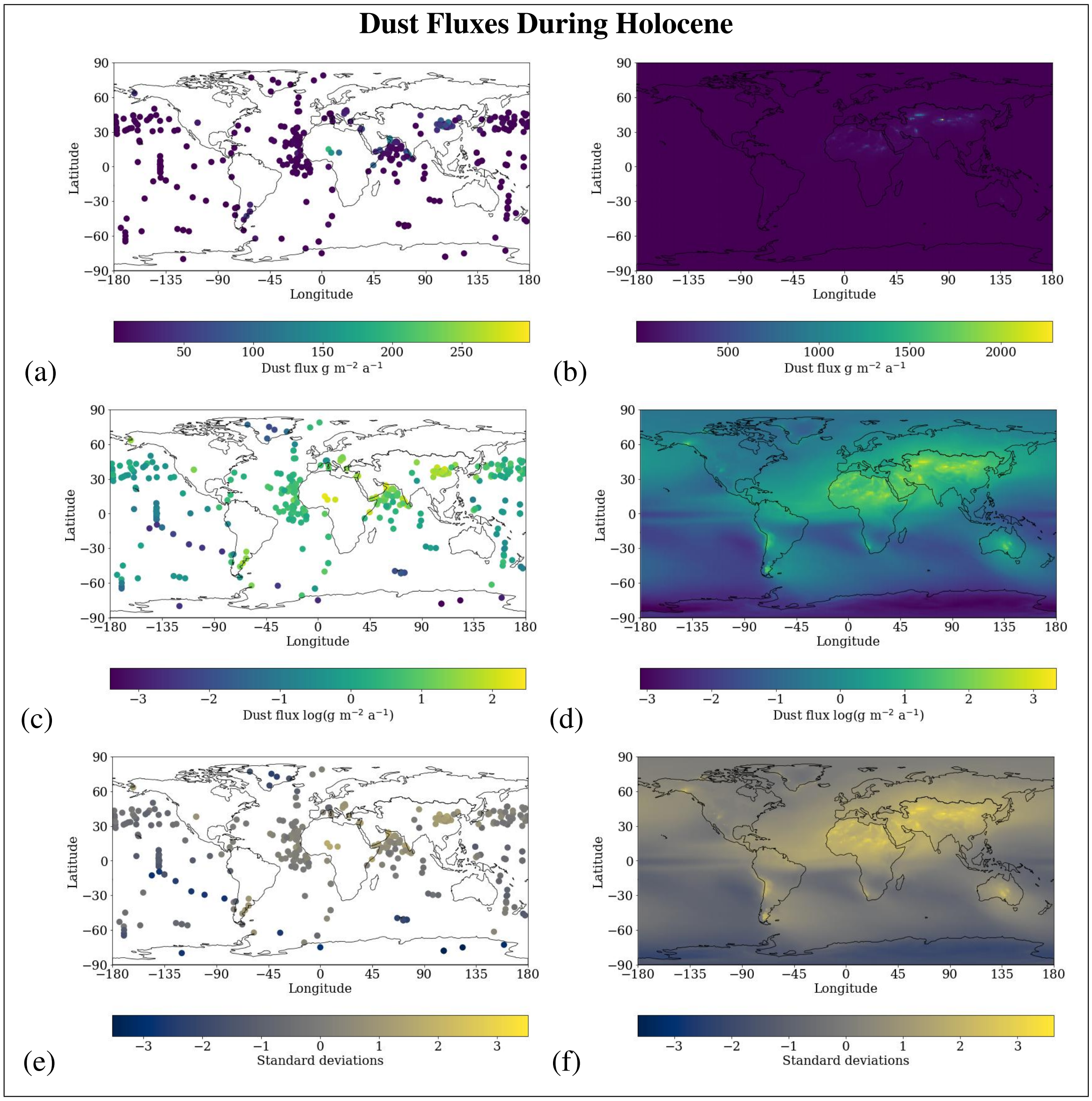

Figure 6.1. Presentation of the dataset and the preprocessing for Holocene time. (a) Dust fluxes measurements. (b) Simulated dust fluxes. (c) Dust fluxes measurements with logarithmic transformation. (d) Simulated dust fluxes with logarithmic transformation. (e) Dust fluxes measurements samples with logarithmic transformation and standard normalization. (f) Simulated dust fluxes with logarithmic transformation and standard normalization.

The preprocessing of the dataset includes logarithmic transformation and standard normalization. Figure 6.1 shows the observed (left) and simulated data (right) in the Holocene

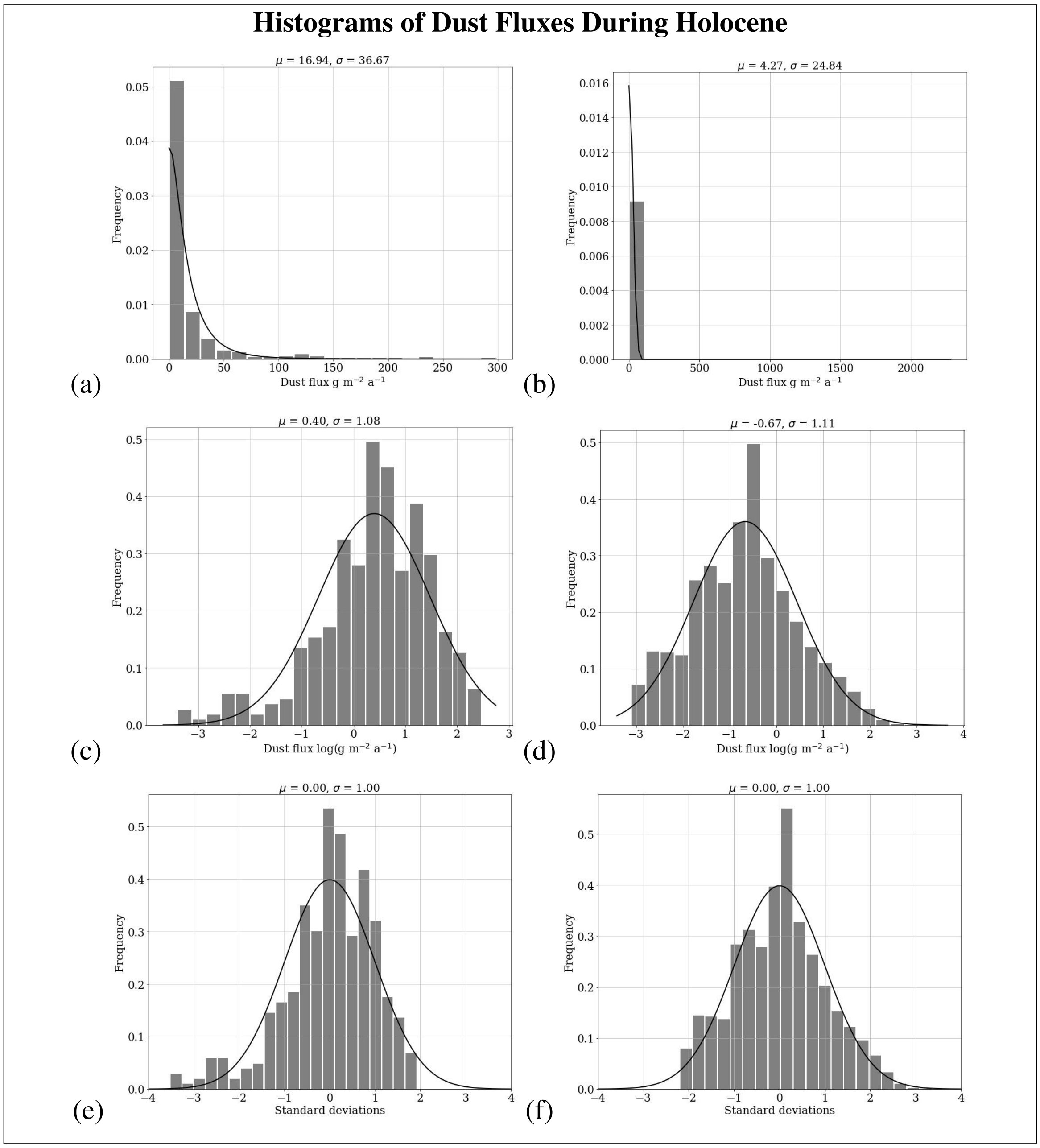

Figure 6.2. Distribution of the data analysis for Holocene time. (a) Histogram of dust fluxes measurements. (b) Histogram of simulated dust fluxes. (c) Histogram of dust fluxes measurements with logarithmic transformation. (d) Histogram of simulated dust fluxes with logarithmic transformation. (e) Histogram of dust fluxes measurements with logarithmic transformation and standard normalization. (f) Histogram of simulated dust fluxes with logarithmic transformation and standard normalization.

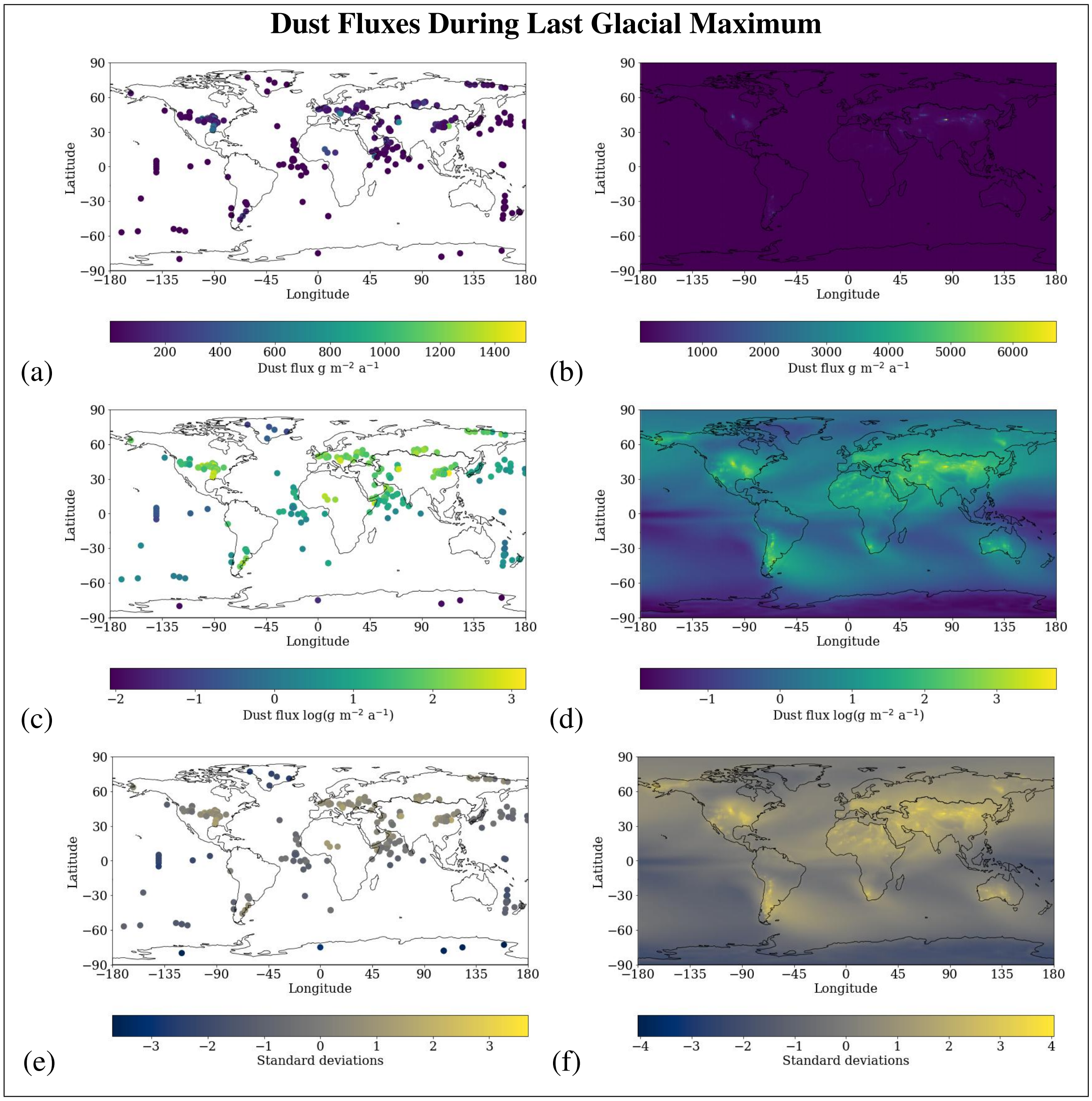

Figure 6.3. Presentation of the dataset and the preprocessing for LGM time. (a) Dust fluxes measurements. (b) Simulated dust fluxes. (c) Dust fluxes measurements with logarithmic transformation. (d) Simulated dust fluxes with logarithmic transformation. (e) Dust fluxes measurements with logarithmic transformation and standard normalization. (f) Dust fluxes concentrations with logarithmic transformation and standard normalization.

period, and Figure 6.2 presents the distribution of the respective data. A lognormal distribution can be observed in Figures 6.2(a) and 6.2(b), with a standard deviation of 36 and 24

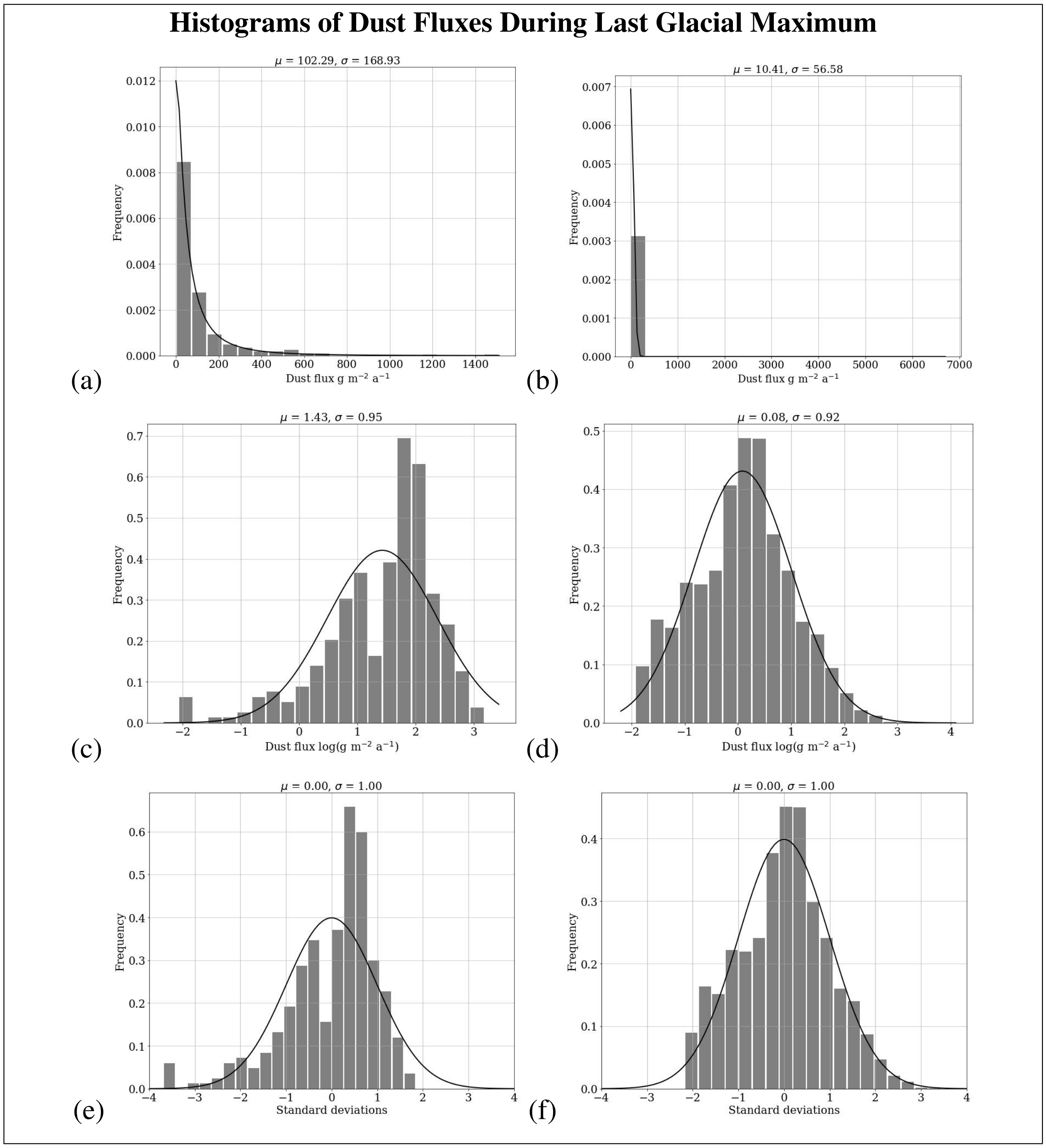

Figure 6.4. Distribution of the data analysis for LGM time. (a) Histogram of dust fluxes measurements. (b) Histogram of simulated dust fluxes. (c) Histogram of dust fluxes measurements with logarithmic transformation. (d) Histogram of simulated dust fluxes with logarithmic transformation. (e) Histogram of dust fluxes measurements with logarithmic transformation and standard normalization. (f) Histogram of simulated dust fluxes with logarithmic transformation and standard normalization.

for observed and simulated data, respectively. After applying the logarithmic transformation, a normal distribution is observed in Figures 6.2 (c) and (d). Finally, a standardized normal distribution is shown in Figures 6.2 (e) and (f). The same preprocessing steps are applied for the LGM data, which are presented in Figures 6.3 and 6.4.

Additionally, another essential information added in the models is the dominant wind flows which are used for the construction of the models in Chapters 7 and 8.

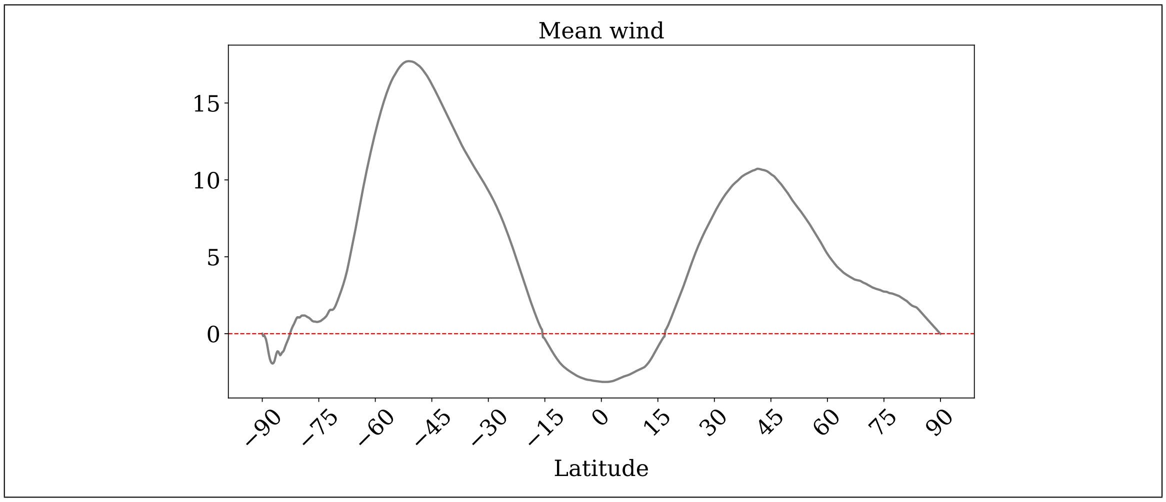

Figure 6.5. Wind values show the mean wind from east-west directions depending on the latitude [Gelaro et al., 2017]. This indicates that the prevailing wind direction tends to be easterly, with the highest wind speed observed at latitude -50, and the strongest westerly wind speed recorded at latitude 0.

7. TRANSPORT IN SIMULATED DATA

To better understand the operation of PINNs, it is important to study their performance under different parameters. In this chapter, we focus on the velocity of the transport and its influence on predictions. To do this, we examine an initial scenario transported at three different velocities. In Section 7.1, we explore the case of a constant velocity in one direction. In Section 7.2, we consider two directions separated by the latitude $0°$, but with the same velocity. Finally, in Section 7.3, we examine the case where the velocity is a function of latitude.

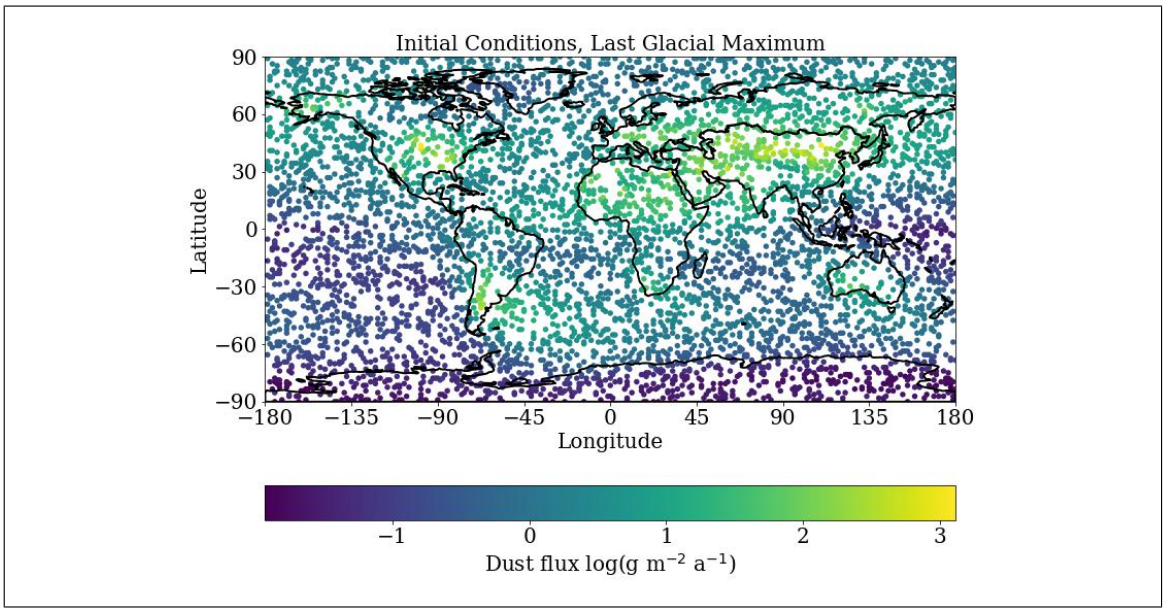

Figure 7.1. Initial conditions. These are 5529 points which are the 10% taken randomly from the Simulated data in LGM from Figure 6.3 (d). The Dust fluxes are in log scale.

The problem to be solved can be formulated as follows:

$$\begin{cases} \frac{\partial u}{\partial t} + k\frac{\partial u}{\partial x} & = 0, \qquad x \in (-2,2), y \in (-1,1), t \geq 0 \\ u(-2,y,t) & = u(2,y,t), \\ u(x,-1,t) & = u_{\text{south}}, \\ u(x,1,t) & = u_{\text{north}}, \\ u(x,y,0) & = \text{Figure 7.1}, \end{cases} \tag{7.1}$$

where parameter k is the velocity, the values $u_{\text{north pole}}$ and $u_{\text{south pole}}$ correspond to the mean value of the initial conditions at the poles which is at latitude $90°$ and $-90°$. The domain was normalized and divided into $90°$ resulting in a longitude range from -2 to 2 and a latitude range from -1 to 1, to improve the stability of the training process.

The neural network has 32 neurons and 5 layers, with a ReLU activation function and a learning rate of 0.001. For testing, the 90% of the points excluded from the initial conditions were used, which corresponds to an amount of 49,767 points.

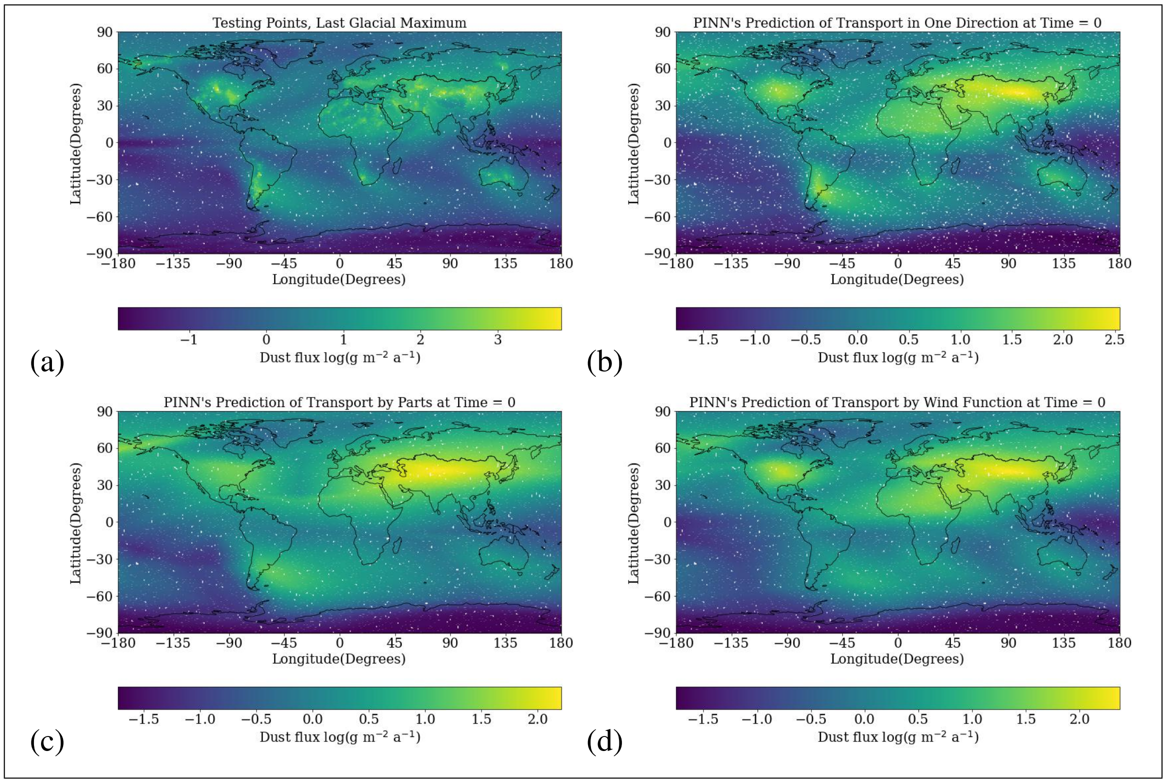

Figure 7.2. (a) Data at testing points (b) Transport in one direction at time $t = 0$ (c) Transport as a function of latitude at time $t = 0$, (d) Transport as a function of wind at time $t = 0$.

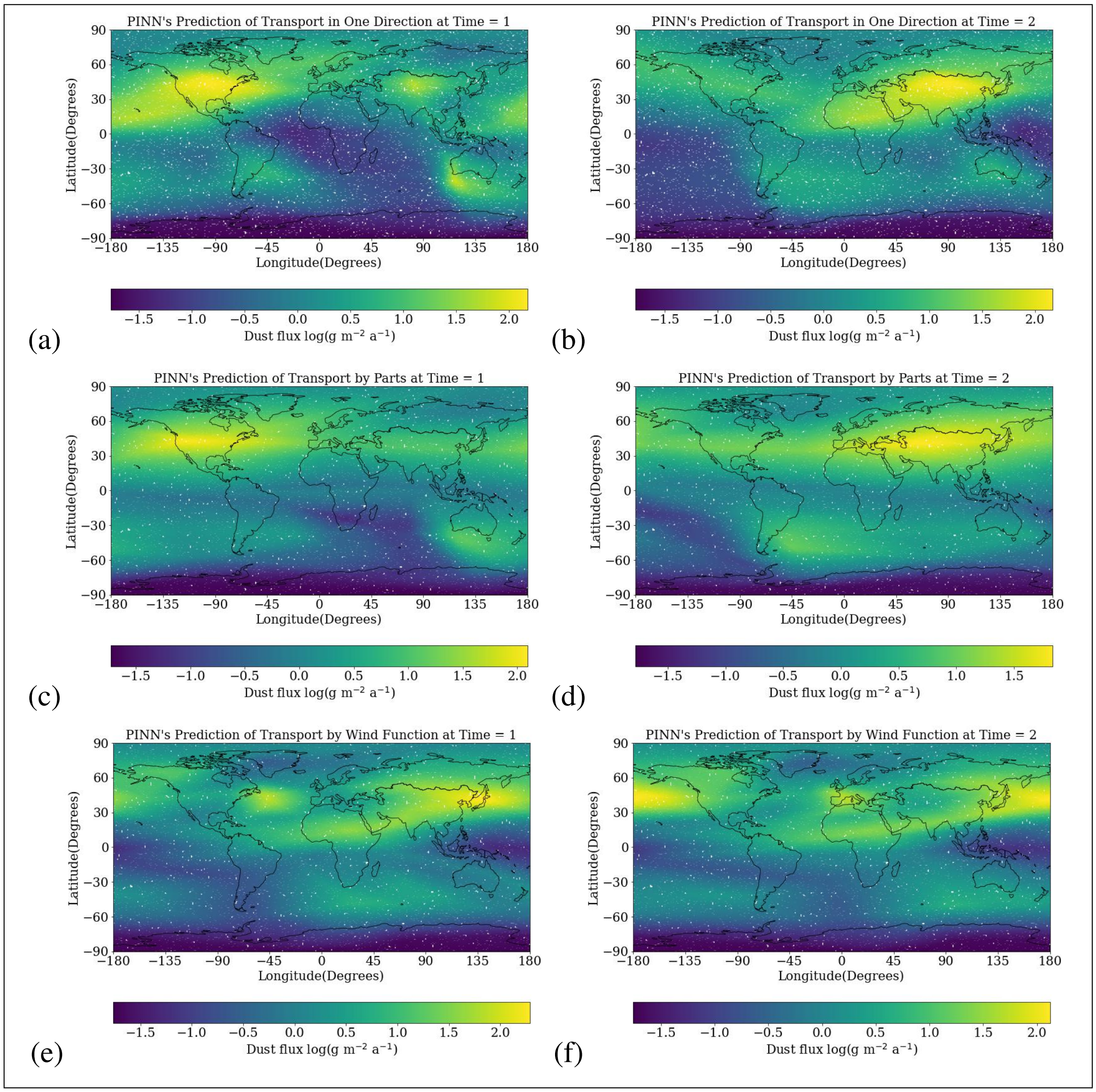

Figure 7.3. (a) Transport in one direction, $t = 1$. (b) Transport in one direction, $t = 2$. (c) Transport as a function of latitude, $t = 1$. (d) Transport as a function of latitude, $t = 2$. (e) Transport as a function of wind, $t = 1$. (f) Transport as a function of wind, $t = 2$.

Table 7.1. Statistics of PINN predictions for the problem (7.1) under different transport conditions. The first column indicates the elapsed time of the experiment. The statistics include the mean, sum, minimum, and maximum values of the prediction.

	Mean	**Sum**	**Minimum**	**Maximum**
Test data	0.08	4089.30	-1.93	3.83
Transport in One Direction				
PINN $t = 0$	0.06	2754.75	-1.79	2.46
PINN $t = 1$	0.05	2483.86	-1.75	2.17
PINN $t = 2$	0.06	3230.83	-1.78	2.18
Transport as a Function of Latitude				
PINN $t = 0$	0.10	4852.74	-1.76	2.12
PINN $t = 0.5$	0.09	4468.54	-1.74	2.07
PINN $t = 1$	0.08	3969.39	-1.74	1.98
PINN $t = 2$	0.07	3655.00	-1.75	1.92
Transport as a Function of Wind				
PINN $t = 0$	0.08	3783.57	-1.75	2.42
PINN $t = 0.5$	0.08	3835.97	-1.75	2.40
PINN $t = 1$	0.08	3914.09	-1.74	2.34
PINN $t = 2$	0.08	3855.02	-1.80	2.20

Table 7.2. The table shows the L_2 difference norm and the L_2 relative difference norm between the PINN's predictions under different transport conditions and the testing points or the prediction at time 2.

Comparison	L_2 **Difference Norm**	L_2 **Relative Difference Norm**
Transport in One Direction		
PINN $t = 0$ and testing data	39.14	0.18
PINN $t = 2$ and testing data	70.79	0.34
PINN $t = 0$ and PINN $t = 2$	55.33	0.26
Transport as a Function of Latitude		
PINN $t = 0$ and testing data	61.17	0.29
PINN $t = 2$ and testing data	90.39	0.43
PINN $t = 0$ and PINN $t = 2$	45.51	0.21
Transport as a Function of Wind		
PINN $t = 0$ and testing data	59.45	0.28

7.1. Transport In One Direction

For the transport in one direction with velocity $k = 2$ in equation (7.1), this means that in one unit of time, the transport is two units eastward. At $t = 2$, the deposition will be transported by 4 units, which correspond to a full 360º rotation of the earth, and this aligns with the initial condition.

As mentioned before, Figure 7.2 (a) represents 90% of the entire dataset, consisting of the testing points. Figures 7.3 a) and 7.3 b) show the network's predictions at times $t = 1$ and $t = 2$, respectively. It can be observed that the range of the testing values is approximately from -2 to 3.5. On the other hand, the range of the predicted values is around -2 to 2.5. This could be due to the fact that the training data has a maximum of 2.5 and because the PINN regularizes outliers by searching for a smooth solution of the PDE.

In Table 7.1, the mean, sum, minimum, and maximum values are calculated for the testing location points and the PINN's predictions at $t = 0, 1$, and 2. It can be observed that the mean and sum of the predicted values are generally lower than those of the testing points. However, the minimum and maximum values of the network's predictions are similar to each other and to those of the testing points.

Table 7.2 displays the L_2 difference norm and L_2 relative difference norm, which measure the differences between the network's predictions and the testing points. The metrics are calculated for three cases: the prediction at $t = 0$ versus the testing points, the prediction at $t = 2$ versus the testing points, and the prediction at $t = 0$ versus $t = 2$.

The L_2 difference norm and L_2 relative difference norm between PINN prediction and testing points are higher in the PINN's prediction at $t = 2$ than at $t = 0$. This could be due to numerical errors, which are also observed comparing the similarity between the testing points in Figure 7.2 a) with the predictions at $t = 0$ in Figures 7.2 b), c) and d) and the predictions at $t = 2$ in Figures 7.3 b) d) f). Furthermore, this performance difference may be because more information is available to the neural network at $t = 0$ due to the initial conditions given. Finally, it is worth noting that the difference between the network's prediction at $t = 0$ and $t = 2$ is 55.33 which is more accurate compared to the network's prediction at $t = 2$ and the testing points which is 70.79.

7.2. Transport as a Function of Latitude

The second velocity studied is the transport as a function of latitude. This corresponds to solving the problem (7.1) with the transport parameter given by the following equation:

$$k = \begin{cases} -2 & \text{if} \quad y \geq 0 \\ 2 & \text{if} \quad y < 0 \end{cases} \tag{7.2}$$

which means transport to the west in the northern hemisphere and to the east in the southern hemisphere. Let's focus on the sources of Argentina and Asia, due to the velocity vector as a function of latitude, we expect that after one unit of time, Argentina will have been transported towards the east by $180°$, while Asia will have been transported towards the west by the same amount, as shown in Figure 7.3 c). Finally, at time $t = 2$, all concentrations should have traveled $360°$, as can be observed in Figure 7.3 d).

Also, in Table 7.1 it can be noted that the difference between the prediction of the network at times $t = 0$ and $t = 2$ is 45.51 in this case. In the case of the network's prediction with only one velocity direction, it is 55.33. Therefore, there is no negative influence on network training in this case.

7.3. Transport as a Function of Wind

The final variation of the transport studied involves incorporating the velocity as the wind function shown in Figure 6.5. This function has been normalized by its maximum value, which occurs at a latitude of approximately $-45°$ and corresponds to a maximum velocity of 1.

Let's take a look at the source of Argentina near the latitude of approximately $-50°$ in Figure 7.2 a). At $t = 1$, this source should be transported $90°$ to the east, which can be seen in Figure 7.3 e). Additionally, Figure 7.3 f) shows the transport of the initial conditions at $t = 2$. It is important to note that in the cases of transport studied in Sections 7.1 and 7.2, the network's prediction at $t = 2$ should replicate the initial condition due to the constant transport of the concentration in different points. However, in this case, the velocity changes with latitude, so at $t = 2$, the dust concentration can be in different positions around the world.

Through the three different velocity scenarios explored in this chapter, PINN has shown that is capable of providing accurate predictions over time, even when faced with complex changes in wind speed and direction across latitudes. This makes PINN a promising tool for complex simulations that would otherwise be time-consuming.

8. PHYSICS-INFORMED NEURAL NETWORK APPLIED TO GLOBAL DUST FLOW

In this chapter, the mathematical modeling of global dust fluxes is explained, and this information will be incorporated into the construction of the PINN. The PINN will be used for predicting dust fluxes during the Holocene and LGM using observed data. The results of the PINN prediction will be compared with Kriging interpolation. Additionally, the validation of the PINN construction will be performed using simulated data for the same periods.

8.1. PINN Design

8.1.1. Mathematical Modeling of Global Dust Fluxes

Given our focus on modeling dust during the Last Glacial Maximum and Holocene periods, it is important to construct an appropriate model that can provide high-quality information to the network and align with available data. The primary physical processes governing dust fluxes are advection along the dominant wind direction and diffusion towards surrounding regions. In our approach, we employ the advection-diffusion equation as the core model:

$$\frac{\partial U}{\partial t} + \nabla \cdot (\mathbf{v}U) - \nabla \cdot (D\nabla U) = 0, \tag{8.1}$$

where $U(\mathbf{x}, t)$ denotes the dust fluxes, **v** is the wind field, and D is the diffusion coefficient [Fletcher, 2020]. Similar to the observed data we are considering the average dust fluxes over climate periods. Hence, we use the steady-state advection-diffusion equation

$$\nabla \cdot (\mathbf{v}u) - \nabla \cdot (D\nabla u) = 0 \tag{8.2}$$

where $u(\mathbf{x})$ denotes the time-averaged dust fluxes.

Let us write the advection-diffusion equation in spherical coordinates to model the earth's surface. For this purpose, let us denote the radius by r, the longitude by $\lambda \in$

$[-\pi, \pi)$, and the latitude by $\theta \in [-\frac{\pi}{2}, -\frac{\pi}{2}]$. The gradient and Laplacian in spherical coordinates are given by

$$\nabla u = \frac{\hat{\boldsymbol{\imath}}}{r\cos(\theta)}\frac{\partial u}{\partial \lambda} + \frac{\hat{\boldsymbol{\jmath}}}{r}\frac{\partial u}{\partial \theta} + \hat{\boldsymbol{k}}\frac{\partial u}{\partial r}, \tag{8.3}$$

$$\nabla^2 u = \frac{1}{r^2\cos^2(\theta)}\frac{\partial^2 u}{\partial \lambda^2} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial^2 r} - \frac{\tan(\theta)}{r^2}\frac{\partial u}{\partial \theta} + \frac{2}{r}\frac{\partial u}{\partial r}, \tag{8.4}$$

where $(\hat{\boldsymbol{\imath}}, \hat{\boldsymbol{\jmath}}, \hat{\boldsymbol{k}})$ are the unit vectors of the Euclidean system.

The dominant wind flows along the east-west directions depending on the latitude [Gelaro et al., 2017]. Figure 6.5 shows the average wind in the Holocene and LGM. Using the wind direction $\mathbf{v} = \begin{bmatrix} v_1 \\ 0 \end{bmatrix}$ we can write the advection-diffusion equation in spherical coordinates as

$$\begin{aligned}
(\nabla u)\cdot\mathbf{v} + (\nabla\cdot\mathbf{v})u - D\nabla\cdot(\nabla u) &= 0,\\
\begin{bmatrix} \frac{1}{\cos(\theta)}\frac{\partial u}{\partial \lambda} \\ \frac{\partial u}{\partial \theta} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \partial_\lambda \\ \partial_\theta \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ 0 \end{bmatrix} u - D\left(\frac{1}{\cos^2(\theta)}\frac{\partial^2 u}{\partial \lambda^2} + \frac{\partial^2 u}{\partial \theta^2} - \tan(\theta)\frac{\partial u}{\partial \theta}\right) &= 0,\\
\frac{v_1}{\cos(\theta)}\frac{\partial u}{\partial \lambda} - D\left(\frac{1}{\cos^2(\theta)}\frac{\partial^2 u}{\partial \lambda^2} + \frac{\partial^2 u}{\partial \theta^2} - \tan(\theta)\frac{\partial u}{\partial \theta}\right) &= 0.
\end{aligned} \tag{8.5}$$

Here we took r constant because we consider Earth's surface as a sphere.

In the direction of the longitude we have the periodic boundary conditions

$$\begin{cases} u(-\pi, \theta) &= u(\pi, \theta), \\ \frac{\partial u(-\pi,\theta)}{\partial \lambda} &= \frac{\partial u(\pi,\theta)}{\partial \lambda}, \end{cases} \tag{8.6}$$

for the latitude, we used fixed values in the poles as

$$\begin{cases} u(\lambda, \frac{-\pi}{2}) &= u_{\text{south}}, \\ u(\lambda, \frac{\pi}{2}) &= u_{\text{north}}, \end{cases} \tag{8.7}$$

where u_{south} and u_{north} are prescribed values at the south and north poles, respectively.

8.1.2. Loss Function Design

The PDE loss function quantifies the fit of the PINN's prediction regarding our model PDE (8.5) as

$$\mathcal{L}_{\text{PDE}} = \left|\left|\frac{v_1}{\cos(\theta)}\frac{\partial \hat{u}(\overline{\mathbf{x}})}{\partial \lambda} - D\left(\frac{1}{\cos^2(\theta)}\frac{\partial^2 \hat{u}(\overline{\mathbf{x}})}{\partial \lambda^2} + \frac{\partial^2 \hat{u}(\overline{\mathbf{x}})}{\partial \theta^2} - \tan(\theta)\frac{\partial \hat{u}(\overline{\mathbf{x}})}{\partial \theta}\right)\right|\right| \tag{8.8}$$

where $\overline{x}_j$ for $j = 1\ldots, N_{\text{PDE}}$ are the collocation points where the PDE will be evaluated by the PINN. Finally, the loss function regarding the boundary conditions (8.6) and (8.7) are given by

$$\begin{aligned}\mathcal{L}_{\text{BC}} = w_{2,2,1}\left|\left|\hat{u}(-\pi,\overline{\theta}) - \hat{u}(\pi,\overline{\theta})\right|\right| + w_{2,2,2}\left|\left|\frac{\partial}{\partial\lambda}\hat{u}(-\pi,\overline{\theta}) - \frac{\partial}{\partial\lambda}\hat{u}(\pi,\overline{\theta})\right|\right| + \\ w_{2,2,3}\left|\left|\hat{u}\left(\overline{\lambda}, -\frac{\pi}{2}+\epsilon\right) - u_{\text{south}}\right|\right| + w_{2,2,4}\left|\left|\hat{u}\left(\overline{\lambda}, \frac{\pi}{2}-\epsilon\right) - u_{\text{north}}\right|\right|\end{aligned} \tag{8.9}$$

$\overline{\theta}_j$ and $\overline{\lambda}_j$ for $j = 1\ldots, N_{\text{BC}}$ are the collocation points for the boundary conditions. Here we include the offset ϵ to avoid numerical issues due to singularities in the spherical coordinate system. According to the application, the relative emphasis between the data and the model can be adjusted by a weight w_1 and $\mathbf{w_2}$. Also, the importance of the conditions of the model loss function can be adjusted by choosing appropriate values for $\mathbf{w}_2$.

8.1.3. Neural Network Design

To enhance the stability of the training algorithm, we normalize the computational domain to $[-2, 2]$ for the longitude and $[-1, 1]$ for the latitude. To avoid the pole singularities, we choose $\epsilon = 0.1$, so we consider latitudes between -81 to 81 degrees. None of the observed data points are inside the excluded zone. Furthermore, we normalize all observed data points as explained in section 6.

The neural network architecture used for all problems is a fully-connected neural network with 5 hidden layers, each containing 32 neurons with a $tanh$ activation function and Glorot normal initializer. The optimizer used is Adam and the learning rate is set to 0.0001. The library used to solve the PINNs is DeepXDE [Lu et al., 2021], an open-source Python package that provides important methods such as creating the mesh of the equation's domain, defining the neural network, initial and boundary conditions, and the learning process.

8.1.4. Physical Parameters Design

The partial differential equation in the model loss term (8.8) is evaluated in collocation points that can be specified in arbitrary locations. We choose a regular grid with 5 degrees spacing between the points. This is fine enough to capture the dominant advection and diffusion phenomena of the global dust deposition, and sufficiently coarse to avoid overfitting. The boundary conditions are evaluated at 5 degrees resolution also.

The value of v_1 is taken from the wind data presented in Figure 6.5 and normalized by its maximum value. However, there is no measured data on the diffusion parameter (D) available in the literature. Hence, we opted to use an inverse problem approach in the PINN to infer this parameter from the observed data. We also estimated the values of u_{north} and u_{south} using the same approach, because these values are necessary for the boundary condition but no data points are available very close to the poles. More specifically, these physical parameters are incorporated as unknown parameters in the neural network and their values are approximated within the same training process of the dust deposition prediction. This means that these physical parameters are optimized to best fit the data and the model.

As part of the design of the PINN, we have to choose the weights of the separate loss terms. This gives relative importance between the data and the model in the training process. We choose a weight equal to one for the boundary condition at the north and south pole. For the periodic boundary conditions, we used one-half each since they are in the same location. Then, we used a weight equal to 10 for the data loss term to emphasize the importance of closely fitting the observed dataset. Finally, there is no prior knowledge to set the weight for the model loss term relative to the other weights. Instead of setting this parameter manually, we give the PINN the freedom to find an optimal value for it during the training process. This is performed in conjunction with the optimization process mentioned above for the physical parameters.

8.2. Kriging Method in Dust Fluxes

Kriging is a geostatistical interpolation technique that uses a weighted average of nearby samples. The same kriging interpolation performed in [Lambert et al., 2015] for the DIRTMAP database is also used here for comparing the PINN's prediction. The major advantage of using PINN instead of the kriging method is the direct incorporation of physical information such as dominant wind directions. Also, the Earth's spherical shape is approximated using spherical coordinates in our PINN.

8.3. Results

As was mentioned in section 8.1.4, there are three values inferred by the PINN through an inverse problem which are D, u_{north} and u_{south}. The values estimated by the PINN in Holocene and LGM are in Table 8.1 which is to give the PINN the freedom to determine the relative importance between the loss function of the observations and the loss function of the PDE. On the other hand, as was explained in section 8.1.3, we used the same parameters for all the neural networks. To validate all the parameters of the neural network used in the PINN Observed Data, we included the weight $w_{2,1}$ found in the PINN observed data in the PINN simulated data.

To summarize, for the PINN observed data, the weights [$w_{2,1}$, w_1, $w_{2,2,1}$, $w_{2,2,2}$, $w_{2,2,3}$, $w_{2,2,4}$] of the loss functions are $[w_{2,1}, 10, 0.5, 10.5, 1, 1]$, and for the PINN simulated data, they are $[w^{\star}_{2,1}, 10, 0.5, 0.5, 1, 1]$, where $w^{\star}_{2,1}$ is the estimated value $w_{2,1}$ for PINN observed data.

Table 8.1. PINN's estimation for the inverse problem and mean pole values for Holocene and Last Glacial Maximum. In the column "Simulated data" the values North Pole and South Pole are the mean dust fluxes (in standard deviations) for the 9 degrees closest to the North and south pole. In the case of "Observed data" this mean was calculated with the points located upper than 60 degrees for the North pole and lower than -63 degrees for the South pole. The columns "PINN observed data" and "PINN simulated data" show the estimation of the inverse problem for the variables u_{north}, u_{south}, D and $w_{2,1}$ which are North Pole, South Pole, Diffusion and PDE-weight, respectively.

Holocene				
	Observed Data	**PINN Observed Data**	**Simulated Data**	**PINN Simulated Data**
North Pole	-1.042	-0.6626	0.064	0.0232
South Pole	-2.07	-2.2465	-1.70	-1.9852
Diffusion	-	0.0412	-	0.0004
$w_{2,1}$	-	0.0050	-	$0.0050^{\star}$
Last Glacial Maximum				
North Pole	-0.59	-0.6167	0.1267	0.0871
South Pole	-3.45	-2.4051	-1.785	-1.9267
Diffusion	-	0.0432	-	0.0007
$w_{2,1}$	-	0.0097	-	$0.0097^{\star}$

8.3.1. Comparison with Kriging

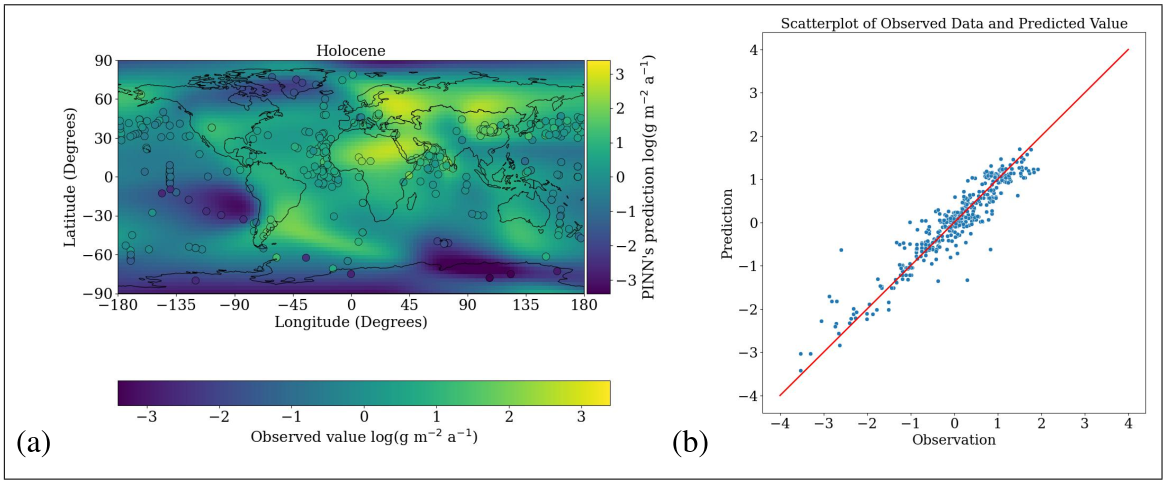

Figure 8.1. Results of data observed in Holocene time. (a) PINN's prediction of the global dust deposition. (b) Scatterplot between the prediction of the PINN and the observed data set.

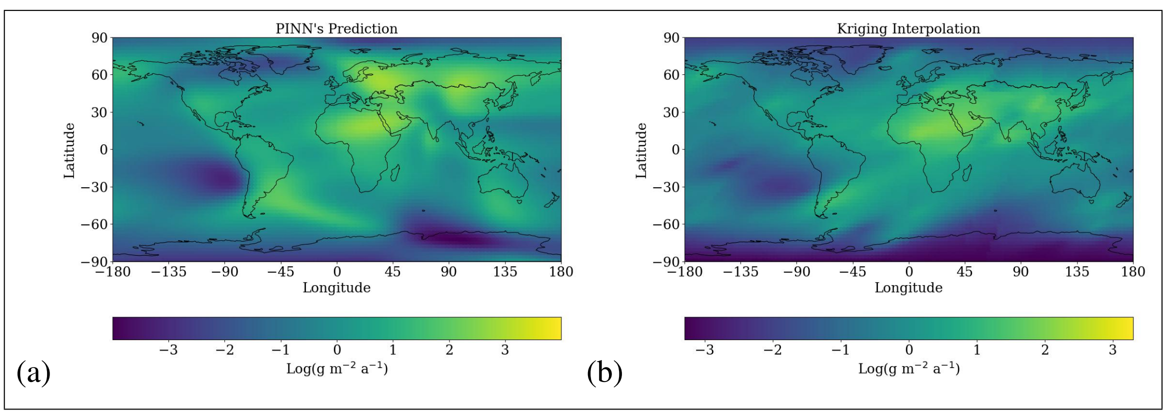

Figure 8.2. Holocene time. (a) Prediction by the PINN. (b) Interpolation by Kriging.

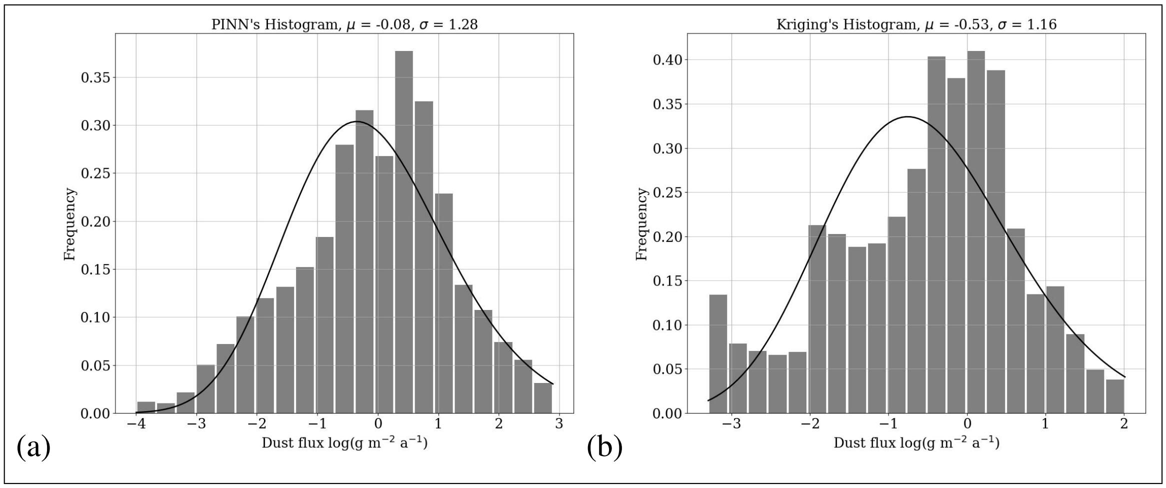

Figure 8.3. (a) Histogram of the prediction by the PINN. (b) Histogram of the interpolation by kriging

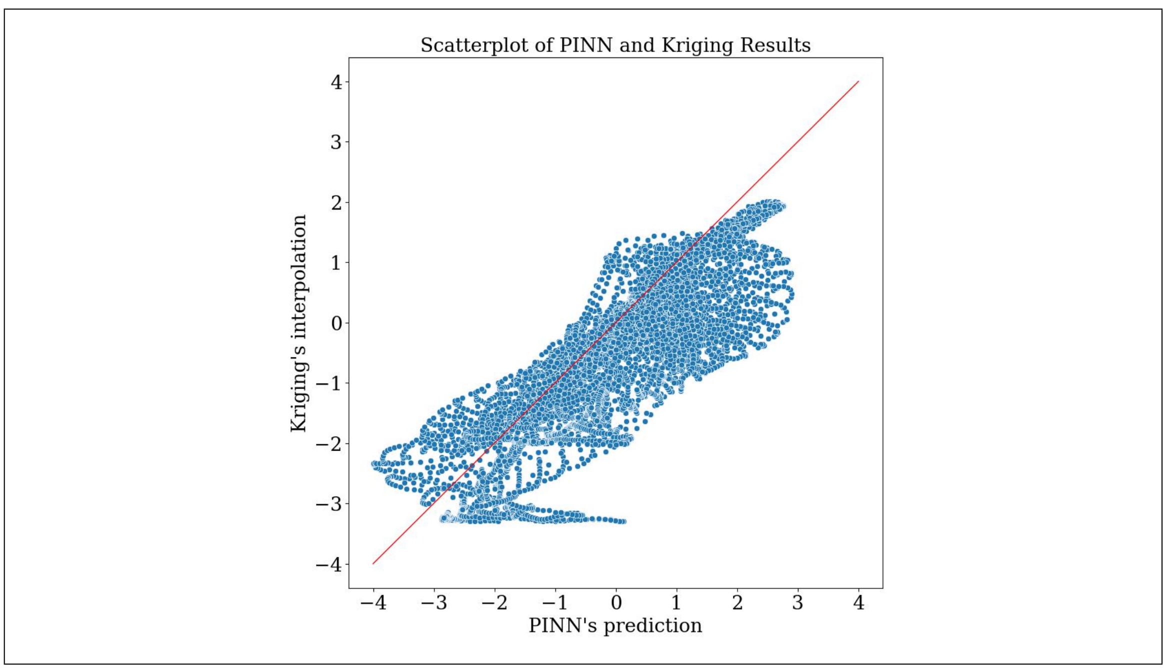

Figure 8.4. Scatterplot between the PINN's prediction and Kriging's interpolation.

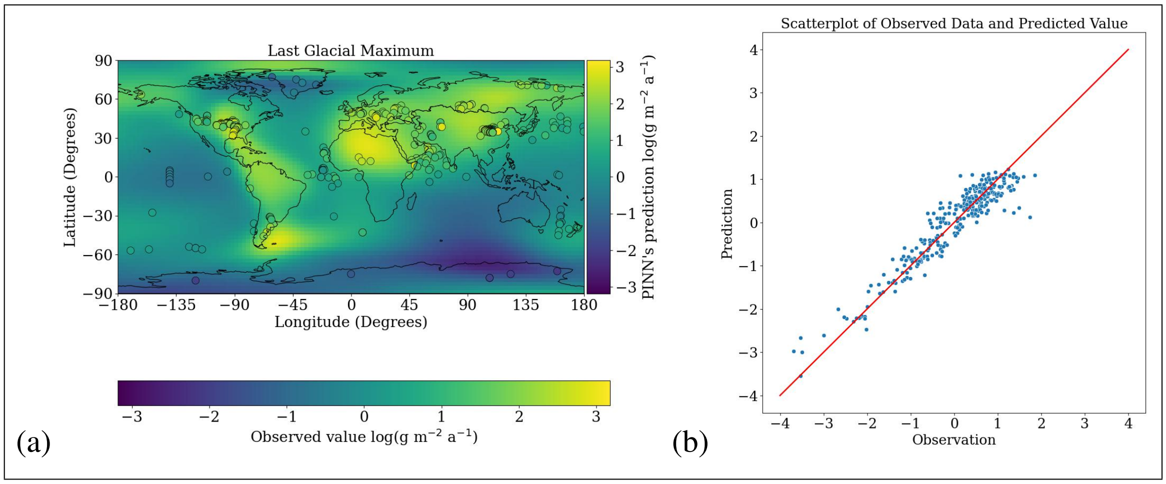

Figure 8.5. Results of observed data in Last Glacial Maximum time. (a) PINN's prediction of the global dust deposition. (b) Scatterplot between the prediction of the PINN and the observed data set.

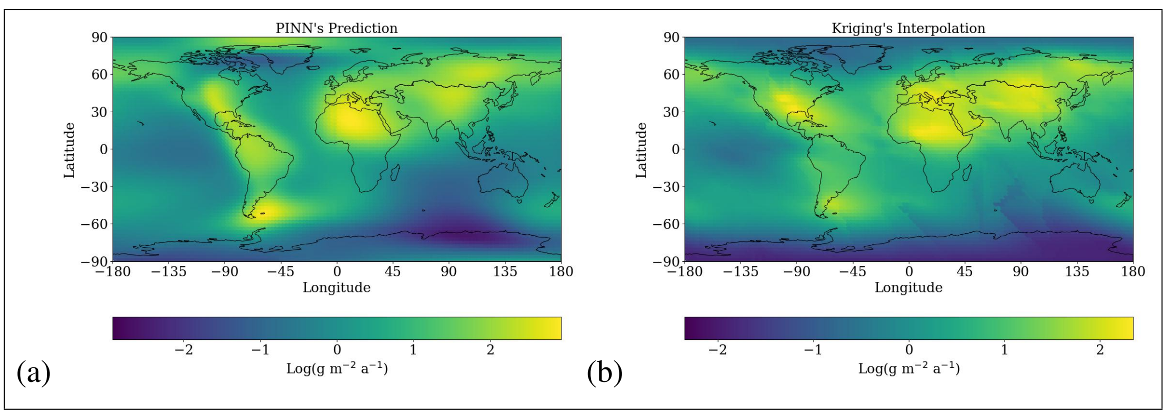

Figure 8.6. Last Glacial Maximum time. (a) Prediction by the PINN. (b) Interpolation by Kriging.

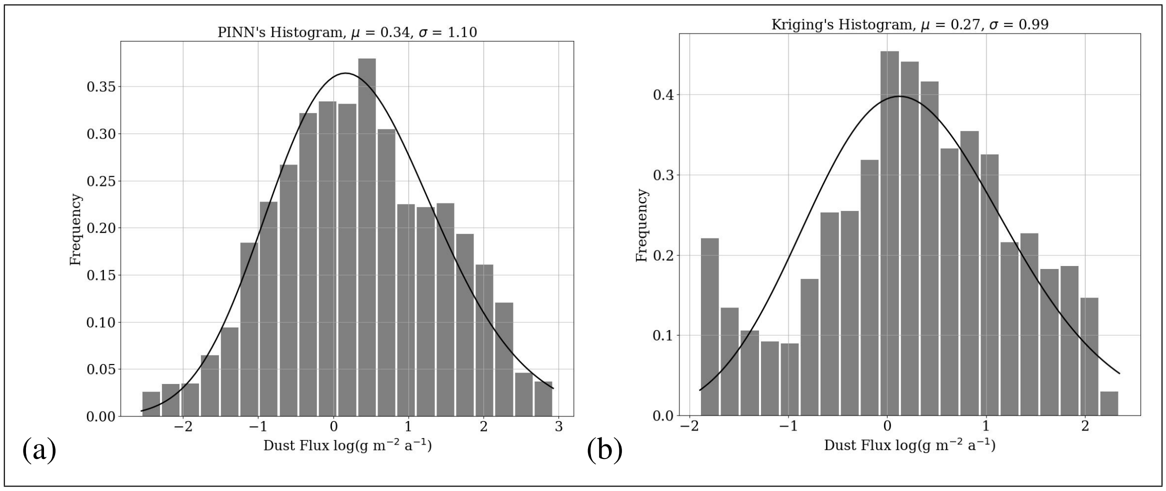

Figure 8.7. (a) Histogram of the prediction by the PINN. (b) Histogram of the interpolation by kriging.

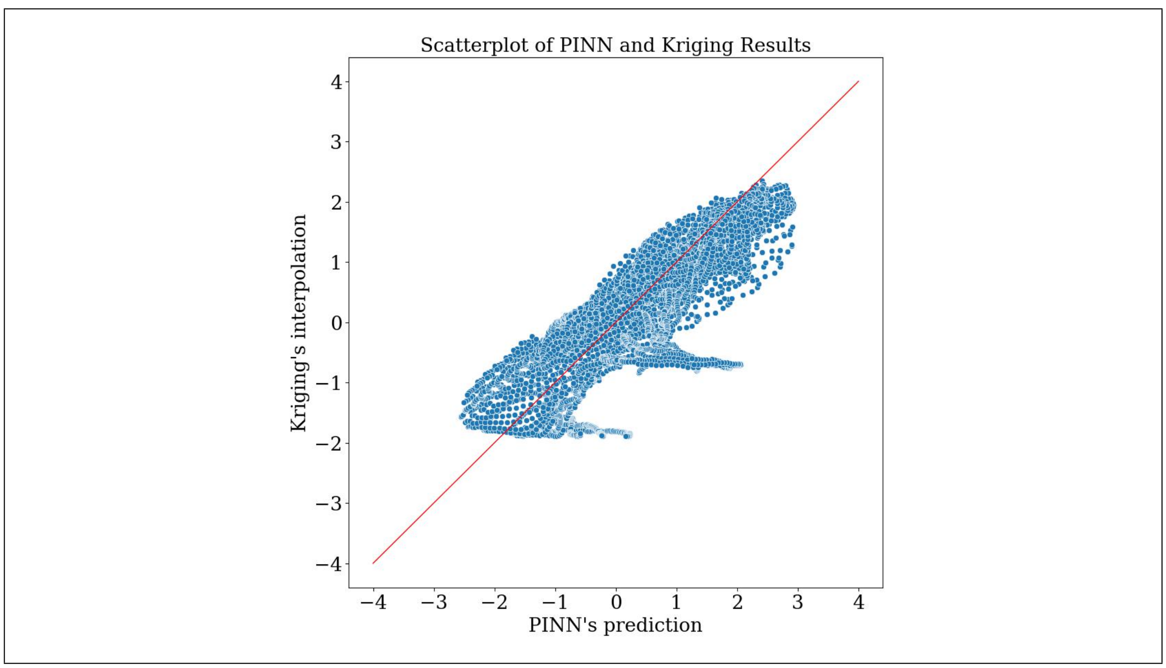

Figure 8.8. b) Scatterplot between the PINN's prediction and Kriging's interpolation.

8.3.2. Comparison with Neural Networks

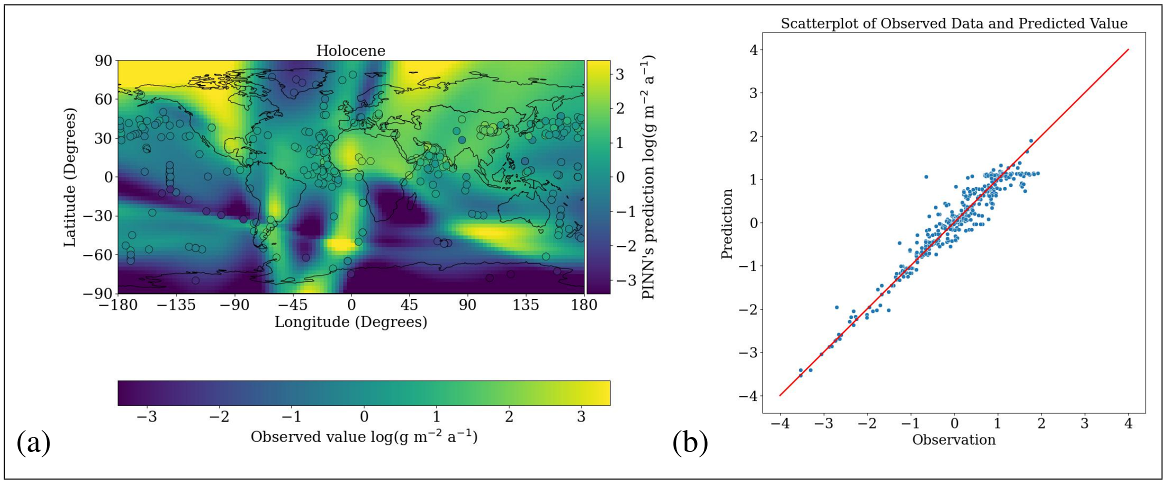

Figure 8.9. Results of observed data in Holocene. (a) NN's prediction of global dust deposition. (b) Scatterplot between the prediction of the trained network with the observed dataset.

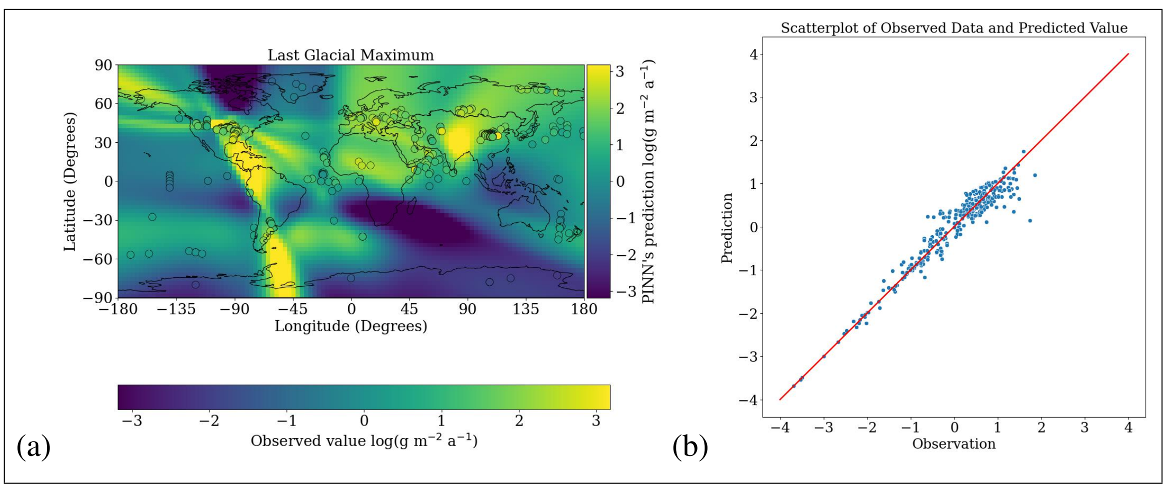

Figure 8.10. Results of observed data in Last Glacial Maximum. (a) NN's prediction of global dust deposition. (b) Scatterplot between the prediction of the trained network with the observed dataset.

8.3.3. Validation on Simulated Data from Global Earth Models

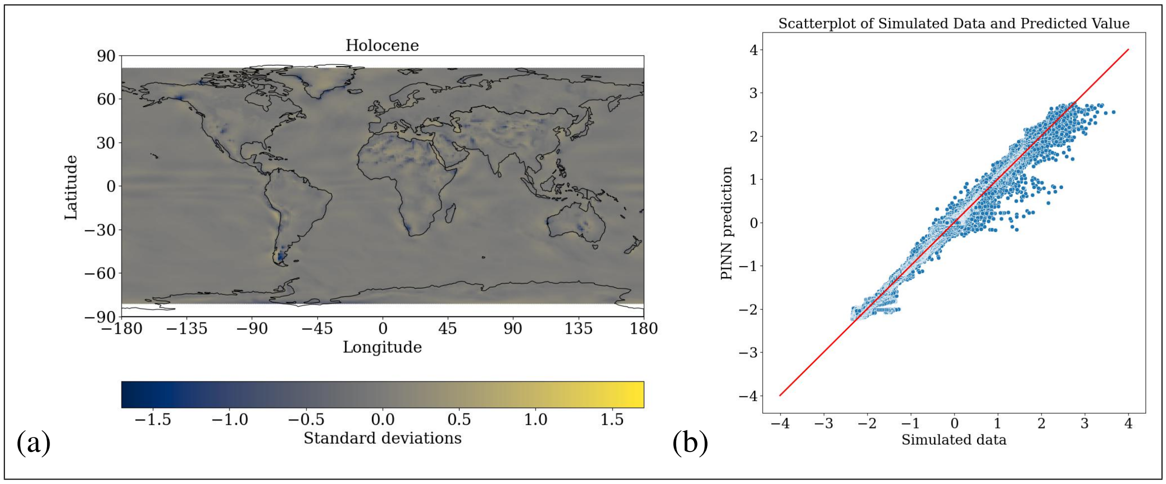

Figure 8.11. Results of data simulated in Holocene time. (a) Difference between the prediction of the PINN and the simulated data. (b) Scatterplot between the prediction of the trained PINN and the simulated data.

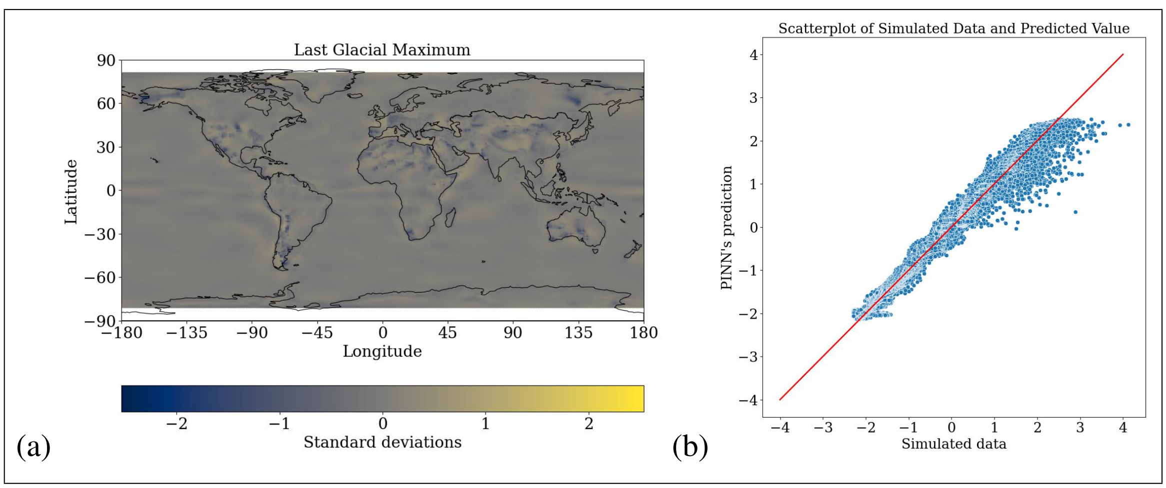

Figure 8.12. Results of data simulated in LGM time. (a) Difference between the prediction of the PINN and the simulated data. (b) Scatterplot between the prediction of the trained PINN and the simulated data.

8.4. Discussion

The hypothesis proposed in this study was that it is possible to include physical information consistent with global dust fluxes in a neural network to improve the accuracy of predictions with the DIRTMAP database in the Last Glacial Maximum and Holocene times, despite a limited and sparse dataset. Our results support this hypothesis, demonstrating that incorporating physical information into a PINN model can indeed improve the accuracy of climatological data. The following sections will discuss our findings in detail.

It is important to highlight the versatility that neural networks provide for including physical information. This characteristic is evident in the incorporation of wind influence and the modeling of physical information in spherical coordinates. Furthermore, neural networks allow for the inference of information related to the problem through the estimation of parameters as an inverse problem. Additionally, it is worth noting that providing physical information makes it possible to work with a small quantity of data. In this study, we worked with 397 and 317 measurements for Holocene and LGM, respectively, which is typically no sufficient data points when working with neural networks in real-world problems. This can be observed in Figures 8.9 and 8.10. The results show that there is a strong relation between the observed data and the predictions of the NN in Figures 8.9 and 8.10 (b). However, there are no accurate predictions in locations where no data is available, see Figures 8.9 and 8.10 (a).

Figure 6.5 shows the wind influence depending on latitude. The majority of the wind flows in an easterly direction, with the maximum at latitude -50 and the second maximum at latitude 45 degrees, approximately. Westerly winds are concentrated between latitudes -15 and 15, with a slight influence at the south pole. Turning our attention to the prediction for the Holocene time in Figure 8.1 (a), we can see high concentrations from Brazil to Patagonia in the observed data at latitude -50 approximately. In Figure 8.2 (a), we can see that peak deposition values are straight east from the terrestrial sources in Patagonia, as predicted by the PINN. However, in Figure 8.2 (b), we can see that the Kriging interpolation predicts the peak to be broader in both east and west directions, consistent with the symmetric nature of kriging. Similar observations can be made for the LGM, where the PINN prediction shows a peak to the south of Patagonia, as seen in Figure 8.6 (a), while the Kriging interpolation predicts with prominence to the south direction, as shown in Figure 8.6 (b).

Another notable difference between the PINN model and the Kriging method lies in the coordinate system used for the modeling and distance interpretation. While the PINN uses spherical coordinates and incorporates periodic boundary conditions, the Kriging method utilizes a 2D projection and measures distances using the Euclidean metric. As a result, the PINN model provides a more accurate interpretation of distances compared to the Kriging method.

Furthermore, the PINN approach offers the flexibility to infer model parameters, such as the values at the North Pole and the South Pole. Table 8.1 presents the estimation of these values. The results indicate that, for both the Holocene and Last Glacial Maximum, and for both observed and simulated data, the estimated values are consistent with the mean values in the dataset for the respective poles.

Unfortunately, the performance of the PINN is highly dependent on the accuracy of the physical information provided and the appropriate construction of the network itself. This is because neural networks solve a non-convex problem, and the results are subject to local minima, making it difficult to draw strong interpretations of specific areas of the estimations. It is important, therefore, to carefully consider the limitations and potential biases of the model when interpreting the results. However, despite these challenges, three principal reasons provide consistency to the results.

The first reason supporting the consistency of the results is the relationship between the predicted and observed data, as demonstrated in the scatterplots in Figure 8.1 (b) and Figure 8.5 (b) for the Holocene and Last Glacial Maximum, respectively. Also in Figures

8.1 (a) and 8.5 (a), the PINN's predictions are seen to closely fit the observed data, while also providing a smooth solution in areas where no data are available. This indicates that the model is capable of accurately capturing the underlying patterns in the data.

The second reason for the consistency of the results is the comparison with Kriging interpolation. It is important to note that the PINN's predictions in Figures 8.1 (a), 8.2 (a), 8.5 (a), and 8.6 (a) extend to the poles, which were not included in the training domain. By examining the scatterplots in Figures 8.4 and 8.8, we can see that the PINN's predictions are related to the Kriging interpolation for both time periods. However, the Kriging prediction has a constant value due to extrapolation beyond the training domain of the PINN and the fixed polar values in the Kriging interpolation.

Additionally, Figure 8.3 shows the histogram of the global prediction by PINN and Kriging. We can see that the outcomes fit well to a log-normal distribution. Also, both of the methods smooth out outliers in the dataset. However, the global mean of the PINN and kriging are different. The observed data fit a log-normal distribution with $\mu = 0.4$ and $\sigma = 1.08$ as can be seen in Figure 6.2 (c). The distribution of the PINN's prediction in Figure 8.3 (a) shows a $\mu = -0.08$ and $\sigma = 1.28$, whereas the Kriging interpolation fits $\mu = -0.53$ and $\sigma = 1.16$ in Figure 8.3 (b). Hence, PINN's mean prediction is closer to the mean of the observed data. Again, in the case of LGM, the prediction conserves the log-normal distribution. The LGM data in Figure 6.4 (c) fits a log-normal distribution with $\mu = 1.43$ and $\sigma = 0.95$. The PINN's prediction in Figure 8.7 (a) gives a mean of $\mu = 0.34$ and $\sigma = 1.10$ while the Kriging's histogram in Figure 8.7 (b) give $\mu = 0.27$ and $\sigma = 0.99$.

The final reason for the consistency of the results is the flexibility of the network. Since network stability cannot be guaranteed, we let the PINN find the weight of the PDE model $w_{2,1}$ and then validated the neural network design using observed in the simulated data. The relation between the PINN prediction and the simulated data can be seen in Figures 8.11 and 8.12. The predictions show good agreement in both, the difference plots and the scatterplots.

9. CONCLUSIONS

In this thesis, the main objective was to incorporate physical information on global dust fluxes during the Holocene and Last Glacial Maximum periods into a neural network, to improve estimations. We aimed to obtain more accurate estimates than statistical methods, which are consistent with the underlying physical processes related to sparse and irregular data.

In Chapter 2, we explained the theoretical basis of Kriging. In Chapter 3, we studied the inner working of PINNs and how to incorporate physical information into a neural network.

In Chapter 4, we evaluated and compared the performance of the Kriging and PINN methods by analyzing the one-dimensional time-dependent diffusion equation and transport equation under various conditions. We observed that both methods accurately estimated the solution of the diffusion equation. When training with noisy data, PINN showed the ability to smooth out the noise through the PDE information. In contrast, the error of the Kriging method increased as the solution had faster transportation and irregular fields in the transport equation. Lastly, in the training of non-uniform grids, we observed that the PDE information played a crucial role in estimating the solution in regions with few data points.

Chapter 5 extended the study to two-dimensional time-dependent PDE problems to explore the construction of the PINN for different parameters. The results showed that the PINN method performed better when estimating smooth solutions. For the transport equation, the PINN was more accurate when periodicity was used as training information instead of Dirichlet boundary conditions. Additionally, we observed that training the PINN in the normalized domain improved its accuracy in predicting the transport process.

In Chapter 6, we introduced two datasets for the Last Glacial Maximum and Holocene periods, the Observed and Simulated datasets from measurements and global earth models, respectively. In Chapter 7, we demonstrated that the PINN method successfully incorporated three different velocities to simulate the transport of dust in the Last Glacial Maximum dataset.

In Chapter 8, we described the physical model for global dust fluxes, the design of the loss function, the neural network architecture, and the physical parameters. The results obtained from the PINN prediction are excellent, demonstrating that the model successfully incorporates physical information and yields accurate estimates.

One of the limitations of this approach is the design of the PINN itself. The design requires many choices that influence the performance such as the PDE, boundary conditions, number of neurons, activation functions, etc. However, there are no clear guidelines on how to choose these design parameters. In practice, these are found by performing different experiments and manually interpreting the performance.

The PINN has demonstrated to be a powerful alternative when working with limited, sparse, and noisy data. By including physical information related to the underlying physical processes, the neural network searches for solutions that are consistent with the problem. Although this study focused solely on the prediction of global dust fluxes, future research could explore the application of this approach to other climatological variables.

In conclusion, the study presents a promising approach to improve the accuracy of climatological data predictions from few measurements that are irregularly distributed over the Earth, and it provides a foundation for future research in climate modeling using neural networks. The findings of the study can have significant implications for understanding the Earth's climate system and informing climate policy decisions.

REFERENCES

S Albani, NM Mahowald, LN Murphy, R Raiswell, JK Moore, RF Anderson, D McGee, LI Bradtmiller, B Delmonte, PP Hesse, et al. Paleodust variability since the last glacial maximum and implications for iron inputs to the ocean. *Geophysical Research Letters*, 43(8):3944–3954, 2016.

Eric Asa, Mohamed Saafi, Joseph Membah, and Arun Billa. Comparison of linear and nonlinear kriging methods for characterization and interpolation of soil data. *Journal of Computing in Civil Engineering*, 26(1):11–18, 2012.

Edward Derbyshire. Loess, and the dust indicators and records of terrestrial and marine palaeoenvironments (dirtmap) database. *Quaternary Science Reviews*, 22(18-19):1813–1819, 2003.

Steven J. Fletcher. Chapter 12 - semi-lagrangian methods on a sphere. In Steven J. Fletcher, editor, *Semi-Lagrangian Advection Methods and Their Applications in Geoscience*, pages 381–469. Elsevier, 2020. ISBN 978-0-12-817222-3. doi: https://doi.org/10.1016/B978-0-12-817222-3.00016-5. URL `https://www.sciencedirect.com/science/article/pii/B9780128172223000165`.

Ronald Gelaro, Will McCarty, Max J. Suárez, Ricardo Todling, Andrea Molod, Lawrence Takacs, Cynthia A. Randles, Anton Darmenov, Michael G. Bosilovich, Rolf Reichle, Krzysztof Wargan, Lawrence Coy, Richard Cullather, Clara Draper, Santha Akella, Virginie Buchard, Austin Conaty, Arlindo M. da Silva, Wei Gu, Gi-Kong Kim, Randal Koster, Robert Lucchesi, Dagmar Merkova, Jon Eric Nielsen, Gary Partyka, Steven Pawson, William Putman, Michele Rienecker, Siegfried D. Schubert, Meta Sienkiewicz, and Bin Zhao. The modern-era retrospective analysis for research and applications, version 2 (merra-2). *Journal of Climate*, 30(14):5419 – 5454, 2017. doi: https://doi.org/10.1175/JCLI-D-16-0758.1. URL `https://journals.ametsoc.org/view/journals/clim/30/14/jcli-d-16-0758.1.xml`.

Philip D Hughes, Philip L Gibbard, and Jürgen Ehlers. Timing of glaciation during the last glacial cycle: evaluating the concept of a global 'last glacial maximum'(lgm). *Earth-Science Reviews*, 125:171–198, 2013.

Karen E Kohfeld and Sandy P Harrison. Dirtmap: the geological record of dust. *Earth-Science Reviews*, 54(1-3):81–114, 2001.

JF Kok, S Albani, NM Mahowald, and DS Ward. An improved dust emission model–part 2: Evaluation in the community earth system model, with implications for the use of dust source functions. *Atmospheric Chemistry and Physics*, 14(23):13043–13061, 2014.

Fabrice Lambert, Alessandro Tagliabue, Gary Shaffer, Frank Lamy, Gisela Winckler, Laura Farias, Laura Gallardo, and Ricardo De Pol-Holz. Dust fluxes and iron fertilization in holocene and last glacial maximum climates. *Geophysical Research Letters*, 42(14):6014–6023, 2015.

Huilai Li, Zhuoqun Wu, Jingxue Yin, and Junning Zhao. *Nonlinear diffusion equations*. World Scientific, 2001.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

K-Nearest Neighbor Query. K-anonymity. 2017.

Heinz Wanner, Olga Solomina, Martin Grosjean, Stefan P Ritz, and Marketa Jetel. Structure and origin of holocene cold events. *Quaternary Science Reviews*, 30(21-22):3109–3123, 2011.