University of Stuttgart

Institute for Parallel and Distributed Systems
Departement of Scientific Computing

Simulation Technology Degree Course

Master Thesis

A Deep Learning Approach for Large-Scale Groundwater Heat Pump Temperature Prediction

Supervisor	First Examiner	Second Examiner
M.Sc.	Prof. Dr.	Prof. Dr.
Raphael Leiteritz	Dirk Pflüger	Miriam Schulte

submitted at the University of Stuttgart by

Author	Stefania Scheurer
Matriculation-No.	3237304
SimTech-No.	92
Submission Date	October 19, 2022

Stefania Scheurer: Master Thesis.
A Deep Learning Approach for Large-Scale
Groundwater Heat Pump Temperature
Prediction

Abstract

Heating and cooling buildings is one of the most energy-intensive aspects of modern life. To minimize the impact on global warming and decelerate climate change, more efficient and carbon emission-mitigating technologies such as open-loop groundwater heat pumps (GWHP) for heating and cooling buildings are being used and quickly adopted. Nowadays, in order to guarantee their optimal use and prevent negative interactions, city planners need to optimize their placement in the urban landscape. This optimization process requires fast models that simulate the effect of a GWHP on the groundwater temperature.

Considering a large domain with multiple GWHPs, this work introduces a framework for the groundwater temperature prediction. While using a learned local surrogate model, a convolutional neural network, to predict the local temperature field around every single GWHP, a physics-informed neural network (PINN) is employed afterwards to correct the global initial solution of stitched together local predictions.

As the violations of the physical laws described by the underlying partial differential equation(s) are spatially unevenly distributed, two different methods for drawing sampling points, on the basis of which the training of the PINN to correct the global initial solution takes place, are investigated and compared.

This work shows that it is possible for a PINN to correct the global initial solution of stitched together local predictions in a domain with multiple GWHPs. However, there are still opportunities to improve the quality and decrease the computational time of the presented framework. The best method for drawing sampling points depends on the scenario and the placement of the GWHPs. Thus, no general statement can be made, which of the two methods is more suitable. This work provides a good basis for further investigation of the presented framework.

Zusammenfassung

Das Heizen und Kühlen von Gebäuden ist einer der energieintensivsten Aspekte des modernen Lebens. Um die Auswirkungen auf die Erderwärmung zu minimieren und den Klimawandel zu verlangsamen, werden effizientere und den Kohlenstoffdioxidausstoß mindernde Technologien wie Grundwasserwärmepumpen (GWHP) zum Heizen und Kühlen von Gebäuden eingesetzt und zunehmend installiert. Um deren optimale Nutzung zu gewährleisten und gegenseitige Störungen zu vermeiden, müssen Stadtplaner ihre Positionierung in der Stadtlandschaft optimieren. Dieser Optimierungsprozess erfordert schnelle Simulationsmodelle, die die Auswirkungen einer GWHP auf die Grundwassertemperatur simulieren.

Bei Betrachtung eines großen Gebiets mit mehreren GWHPs stellt diese Arbeit eine Methode für die Vorhersage der Grundwassertemperatur vor. Mithilfe eines bereits gelernten lokalen Ersatzmodells, einem Convolutional Neural Network, wird das lokale Temperaturfeld um jede einzelne GWHP vorhergesagt. Anschließend wird ein Physics-Informed Neural Network (PINN) verwendet, um die globale Anfangslösung der zusammengefügten lokalen Vorhersagen zu korrigieren.

Da die Verstöße gegen die physikalischen Gesetze, die durch die zugrunde liegende(n) partielle(n) Differentialgleichung(en) beschrieben werden, räumlich ungleichmäßig verteilt sind, werden zwei verschiedene Methoden für die Auswahl der Samplingpunkte, auf deren Grundlage das Training des PINNs zur Korrektur der globalen Anfangslösung stattfindet, untersucht und verglichen.

Diese Arbeit zeigt, dass es mithilfe eines PINNs möglich ist die globale Anfangslösung von zusammengesetzten lokalen Vorhersagen auf einem Gebiet mit mehreren GWHPs zu korrigieren. Es gibt jedoch noch Möglichkeiten, die Qualität der vorgestellten Methode zu verbessern und die Laufzeit zu reduzieren. Die beste Methode für die Auswahl der Samplingpunkte hängt vom Szenario und der Positionen der GWHPs ab. Dadurch kann keine allgemeine Aussage getroffen werden, welche der beiden Methoden besser geeignet ist. Diese Arbeit bietet eine gute Grundlage für die weitere Untersuchung der vorgestellten Methode.

List of Figures

List of Tables

Contents

1 Introduction

In the context of the anthropogenic climate change, the employment of renewable energies for energy-intensive processes became inevitable. One of the most energy-intensive tasks is the heating and cooling of residential and non-residential buildings. Within the last years, open-loop groundwater heat pumps (GWHP), also known as open-loop shallow geothermal systems, have attracted attention since they use the thermal energy of the groundwater, which is considered a renewable energy source [Halilovic et al. 2022]. A great advantage of GWHPs is that the temperature of the groundwater is relatively stable throughout the year. Therefore, the energy can be used for both heating and cooling buildings [Leiteritz et al. 2022]. In general, a GWHP extracts groundwater from the subsurface at the extraction well. With the aid of heat exchangers, the heat/cold from the groundwater is transferred and the cooled/heated groundwater is re-injected to the subsurface. When heating the building, the re-injected groundwater is colder than the extracted groundwater; when cooling the building, the re-injected groundwater is warmer.

As the water is re-injected into the subsurface, the temperature around the injection well changes and diffusion and advection of the water create a thermal plume. This thermal plume might spread downstream and in the worst case interacts with other GWHPs or even with itself, if the water is circulating. Both cases might cause interference. As more and more GWHPs are installed, these scenarios become more likely. Therefore, careful planning, i.e. optimizing the installation of GWHPs becomes necessary in order to avoid negative interaction [García-Gil et al. 2019]. For this purpose, an accurate prediction of the developed groundwater temperature field, when using multiple GWHPs, needs to be provided. This can be achieved with the aid of high-fidelity subsurface flow simulations [Meng et al. 2019]. Optimizing the installation of GWHPs in a next step requires a lot of high-fidelity simulation runs, which is not feasible for a large scenario with multiple GWHPs, as quasi-real time prediction is not possible using this type of simulation.

One solution to this problem is the employment of surrogate models for the optimization problem, which serve as low-fidelity simulations of low computational cost. Deep learning methods are a suitable way to build such models. Leiteritz et al. [2022] developed a surrogate model using a convolutional neural network (CNN) to predict the thermal plume of a single GWHP in a domain of fixed size, i.e. a local prediction model. This work aims to include this local prediction model in a framework for predicting the temperature field of a large domain with multiple GWHPs. The idea is to extract the GWHP locations, run the local model on each of them, and finally stitch the results back together to obtain a global solution. As this global solution might be incorrect due to overlapping local areas and discontinuities at the seams, some correction needs to take place. The idea for this correction step is employing a physics-informed neural network (PINN) [Raissi et al. 2019]. PINNs are a class of neural networks for approximating physical processes by leveraging available data and expertise. By using penalty expressions, they are able to restrict the solution space to a region where physically sound predictions can be made. These predictions can help to make a reliable statement about whether a particular configuration of GWHPs is free of interference. The developed PINN hereby is used to improve the given stitched together, local-only surrogate predictions to satisfy the given global subsurface PDEs and thus, yields a more accurate solution.

The overall goal of this work is to find a global, physically sound prediction of the developed temperature field in quasi-real time for a large domain with multiple GWHPs and known spatially varying permeability field and pressure boundary conditions in order to enable a fast solution of the optimization problem.

After giving an overview of the basics in Section 2, the processes to create training and high-fidelity data for quality assessment are explained in Section 3. The methods used are presented in Section 4. The developed framework is applied to different large-scale scenarios in Section 5, where the results are analyzed and visualized, too. Section 6 summarizes the results and gives an outlook for further investigation.

2 Basics

2.1 Neural Networks

A feedforward dense neural network of depth $L+1$ is defined as a concatenation of functions

$$\hat{u}(\boldsymbol{x}) = (\hat{u}_{L+1} \circ \hat{u}_L \circ ... \circ \hat{u}_1)(\boldsymbol{x}) \tag{2.1}$$

with input $\boldsymbol{x} \in \mathbb{R}^n$, where each so-called layer $\hat{u}_i$ is defined as

$$\hat{u}_i(\boldsymbol{x}) = \phi_i(\boldsymbol{W}_i^T \boldsymbol{x} + \boldsymbol{b}_i),\ 1 \leq i \leq L+1. \tag{2.2}$$

The weight matrix $\boldsymbol{W}_i \in \mathbb{R}^{n \times m}$ and the bias vector $\boldsymbol{b}_i \in \mathbb{R}^m$ are variable parameters and $\phi_i(\cdot)$ the element-wise activation function of layer i [Goodfellow et al. 2016]. Figuratively speaking, a neural network is an accumulation of interconnected computational nodes, so-called neurons. These neurons are organized in layers, while the neurons between the layers are connected to each other by the weight matrices and the bias vectors. A neural network consists of an input layer, L hidden layers and one output layer. In case of many hidden layers the network is referred to as deep neural network. The basic structure of a dense neural network is outlined in Fig. 2.1. The aim of a neural network is to optimize its parameters $\{\boldsymbol{W}, \boldsymbol{b}\} = \{\boldsymbol{W}_i, \boldsymbol{b}_i \mid 1 \leq i \leq L+1\}$, such that the output of the network $\hat{u}(\boldsymbol{x})$ is an approximation of a target function $u(\boldsymbol{x})$. The ability of a neural network to approximate any continuous function $u(\boldsymbol{x})$ on compact input sets to arbitrary accuracy is stated in the universal approximation theorem. Required for this property is a neural network with $L \geq 1$ and a nonlinear activation function in the hidden layer [Hornik 1991]. The potential approximation accuracy increases with the amount of neurons [Hornik et al. 1989]. The ability of a neural network $\hat{u}(\boldsymbol{x})$ to optimize its parameters $\{\boldsymbol{W}, \boldsymbol{b}\}$ in order to approximate a function $u(\boldsymbol{x})$ is given by its structure. With the aid of automatic differentiation it is possible to obtain the derivatives of the output $\hat{u}(\boldsymbol{x})$ with respect to the weight matrices and the bias vectors $\{\boldsymbol{W}, \boldsymbol{b}\}$, which are required for the subsequent optimization

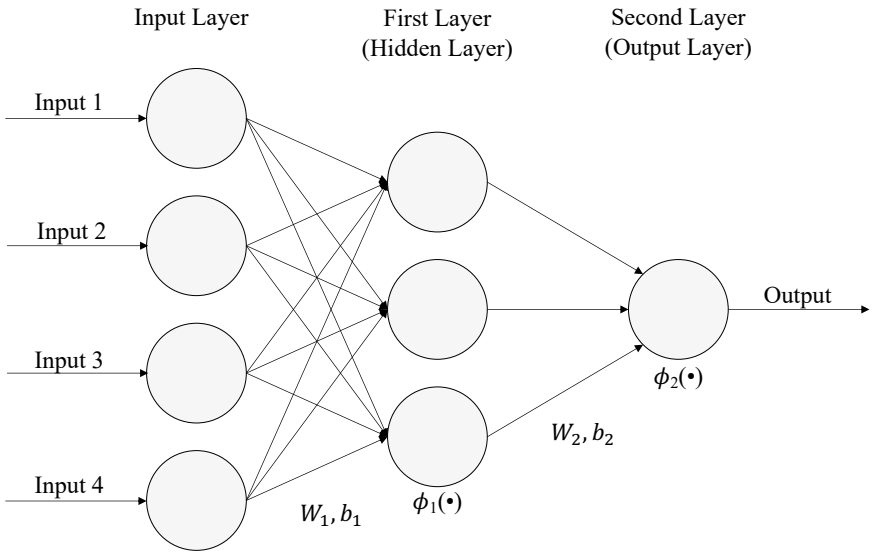

Figure 2.1: Basic feedforward neural network with an input layer of size 4, one hidden layer (i.e. $L = 1$) of size 3 and one output layer of size 1.

step. The calculation of the gradients is carried out by back propagating the gradients through the layers of the network. Thus, this step is also referred to as back propagation.

The optimization of $\{\boldsymbol{W}, \boldsymbol{b}\}$, the training of the neural network, is based on a dataset $\{\boldsymbol{x}_j, y_j = u(\boldsymbol{x}_j) \mid 1 \leq j \leq N_{\hat{u}}\}$ of size $N_{\hat{u}}$. In regression tasks, the deviation of a function $\hat{u}(\boldsymbol{x})$ from a target function $u(\boldsymbol{x})$ is usually given as the mean squared error (MSE), also referred to as loss $L_{\hat{u}}$ in the context of machine learning:

$$\begin{aligned} L_{\hat{u}}\left(\{\boldsymbol{x}_j\}_{j=1}^{N_{\hat{u}}}\right) &= \frac{1}{N_{\hat{u}}} \sum_{j=1}^{N_{\hat{u}}} (\hat{u}(\boldsymbol{x}_j) - u(\boldsymbol{x}_j))^2 \\ &= \frac{1}{N_{\hat{u}}} \sum_{j=1}^{N_{\hat{u}}} (\hat{u}(\boldsymbol{x}_j) - y_j)^2. \end{aligned} \tag{2.3}$$

Approximation of the target function $u(\boldsymbol{x})$ by $\hat{u}(\boldsymbol{x})$ corresponds to the minimization of the loss $L_{\hat{u}}\left(\{\boldsymbol{x}_j\}_{j=1}^{N_{\hat{u}}}\right)$ over $\hat{u}(\boldsymbol{x})$, i.e. over $\{\boldsymbol{W}, \boldsymbol{b}\}$. In order to obtain the optimal values for $\{\boldsymbol{W}, \boldsymbol{b}\}$, the loss can be minimized using a gradient descent

method. In this case the values of the parameters are iteratively adjusted:

$$\boldsymbol{W}_i^{it+1} = \boldsymbol{W}_i^{it} - \alpha \cdot \frac{\partial L_{\hat{u}}\left(\{\boldsymbol{x}_j\}_{j=1}^{N_{\hat{u}}}\right)}{\partial \boldsymbol{W}_i^{it}}$$
$$\boldsymbol{b}_i^{it+1} = \boldsymbol{b}_i^{it} - \alpha \cdot \frac{\partial L_{\hat{u}}\left(\{\boldsymbol{x}_j\}_{j=1}^{N_{\hat{u}}}\right)}{\partial \boldsymbol{b}_i^{it}},\ 1 \leq i \leq L+1,\ 1 \leq it \leq N_e,\ \alpha \in \mathbb{R}^+. \quad (2.4)$$

The so-called learning rate α is the step size when approaching the minimum of the loss. One step or one iteration it is referred to as an epoch, where N_e is the total number of epochs. In each epoch the whole input dataset passes through the neural network, the loss is evaluated and subsequently the parameters $\{\boldsymbol{W}, \boldsymbol{b}\}$ are adjusted accordingly. In case of a large training dataset, it is common to divide it into so-called batches. This means that the loss is not only evaluated and the parameters adjusted after one epoch, i.e. after the entire dataset has passed through the neural network, but only a subset of the dataset of a certain size, a batch. This means that the parameters $\{\boldsymbol{W}, \boldsymbol{b}\}$ are adapted more frequently compared to when not using batches. After each epoch it is recommended to reshuffle or regenerate the batches to obtain different ones in order to improve the robustness and quality of the neural network. Since the datasets considered in this work are rather small, a more detailed explanation of training with multiple batches is omitted.

At the beginning of the training of a neural network, all its parameters $\{\boldsymbol{W}, \boldsymbol{b}\}$ need to be initialized. In this case the Glorot initialization scheme was used [Glorot and Bengio 2010]. It is known for its ability to maintain the variance of the weight and bias gradients across all layers, which counteracts the problem that occurs when the gradients approach zero in the first layers while back propagating, so-called vanishing gradients. According to Eq. (2.4), if the gradient vanishes, i.e. approaches zero, the weights in the earlier layers are not updated anymore. Overall, the use of Glorot initialization, which counteracts this problem of vanishing gradients, leads to a faster convergence [Glorot and Bengio 2010].

After training is complete, the quality of the neural network is assessed by evaluating it on yet "unseen" data, i.e. data that has not been used for training, a so-called validation dataset. The validation error is usually computed in the

same way as the loss function.

2.2 Convolutional Neural Networks

The so far described type of dense neural networks is well suited for regression tasks, but reaches its limits in e.g. local feature learning in images. Using a dense neural network might lead to a loss of information, as a group of close pixels in an image could form a local pattern that should be identified. However, this is not possible if this potential relation of the pixels is neglected, as this would be the case for a dense neural network, where all pixels are just concatenated vertically in order to form the input layer. Also, when using a dense neural network, the full connectivity of the neurons increases the size of the network immensely, which yields two problems. First, a larger network implies that more parameters need to be optimized, thus a higher computational power is needed, which however is limited. The second problem occurring is overfitting. The more parameters that are to be trained, the higher the feasibility that the network learns the features of the training dataset precisely, but at the same time loses its ability to process yet unseen data. Therefore, parameter reduction is necessary [O'Shea and Nash 2015].

One subtype of neural networks are so-called convolutional neural networks (CNN). CNNs consist mainly of so-called convolutional layers which are based on a convolution. A d-dimensional discrete convolution is an operation on two functions, e.g. f and g, which results in a new function $f * g$. Basically, the shape of one function is changed by the other one. A discrete convolution is of the form

$$(f * g)(\boldsymbol{x}) = \sum_{z \in Z} f(\boldsymbol{x}) \cdot g(\boldsymbol{x} - \boldsymbol{z}), \ \boldsymbol{x} \in \mathbb{Z}^d, Z \subseteq \mathbb{Z}^d, \tag{2.5}$$

following the definition of Damelin and Miller Jr [2012]. The advantage of a convolution is that g or the support of g can be chosen, such that the output $(f*g)$ only depends on the values of f at a small region around $\boldsymbol{x}$. This corresponds to the goal formulated for local feature learning, as only a local region around a certain point $\boldsymbol{x}$ is considered. The aspect of the mathematical definition of a convolution corresponds to the fact that in a convolutional layer in a CNN,

the neurons are only connected to a small area of neurons from the previous layer, in contrast to a fully connected layer [O'Shea and Nash 2015]. Basically, a convolution is performed on the output of the previous layer to obtain the output of the convolutional layer. This is done with the aid of weight matrices. Remembering Eq. (2.5), the output of the previous layer can be seen as function f and the weight matrix as function g, also referred to as kernel. A kernel performs a sliding window (convolution) over the output of the previous layer. Kernels are also learnable parameters. Connecting the information of different adjacent pixels via convolution allows the integration of specific local features regarding the images to be learned. Typical kernel sizes $k_i \times k_i$ are 3×3 or 5×5 for $d = 2$, which means their spatial dimension is rather small. Remembering Eq. (2.5) again, this corresponds to the set Z.

For a three-dimensional input $\boldsymbol{x} \in \mathbb{R}^{w \times h \times c}$, e.g. a colored image, where w is the width, h the height of the image in pixels and c the number of input channels (e.g. $c = 3$ for an RGB image), typically two-dimensional convolutions are applied. A convolutional layer $\hat{u}_i$ receives an input $\boldsymbol{x}_{i-1} \in \mathbb{R}^{M_{i-1} \times N_{i-1} \times O_{i-1}}$ with width M_{i-1}, height N_{i-1} and O_{i-1} channels, whereby for the example of the colored image, $M_0 = w$, $N_0 = h$ and $O_0 = c$. For each component or pixel of the input, layer $\hat{u}_i$ performs a two-dimensional convolution

$$\begin{aligned}[\hat{u}_i(\boldsymbol{x}_{i-1})]_{m,n,o} &= \phi_i \left(\sum_{p=1}^{k_i} \sum_{q=1}^{k_i} \sum_{r=1}^{O_{i-1}} [\boldsymbol{W}_i]_{p,q,r,o} \cdot [\boldsymbol{x}_{i-1}]_{m+p-1,n+q-1,r} + [\boldsymbol{b}_i]_o \right) \\ &= [\boldsymbol{x}_i]_{m,n,o} \end{aligned} \tag{2.6}$$

with kernel size k_i and O_{i-1} input channels. The output of the layer is $\boldsymbol{x}_i = \hat{u}_i(\boldsymbol{x}_{i-1}) \in \mathbb{R}^{M_i \times N_i \times O_i}$ with $M_i = M_{i-1} - k_i + 1$, $N_i = N_{i-1} - k_i + 1$ and O_i of free choice. What should be noted at this point is that the size of the learnable parameters $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ does not depend on the size of the input $\boldsymbol{x}_{i-1}$.

Figuratively speaking, when the output of the previous layer, the new input $\boldsymbol{x}_{i-1}$, hits a convolutional layer, the kernel slides over it, calculates the scalar product of the captured section and the kernel, adds a learnable bias $\boldsymbol{b}_i$, and places the scalar output with an element-wise activation function $\phi_i(\cdot)$ applied to the target pixel. A schematic illustration of the sliding window procedure for an

example layer with $O = 1$ channels, kernel size $k = 3$ and $M = N = 6$ is shown in Fig. 2.2, an exemplary calculation is given in Fig. 2.3.

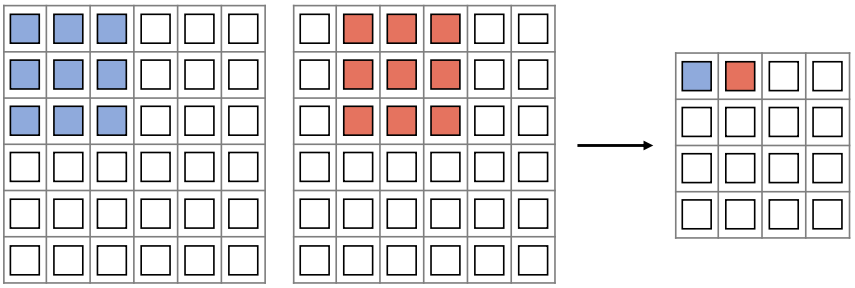

Figure 2.2: Schematic illustration of a two-dimensional convolutional layer with kernel size $k = 3$, $O = 1$ channels and size $M = N = 6$. The blue and red squares in the first and second picture show the position of the sliding window (kernel) for the first and the second step. The corresponding squares on the right show the positions of the resulting values.

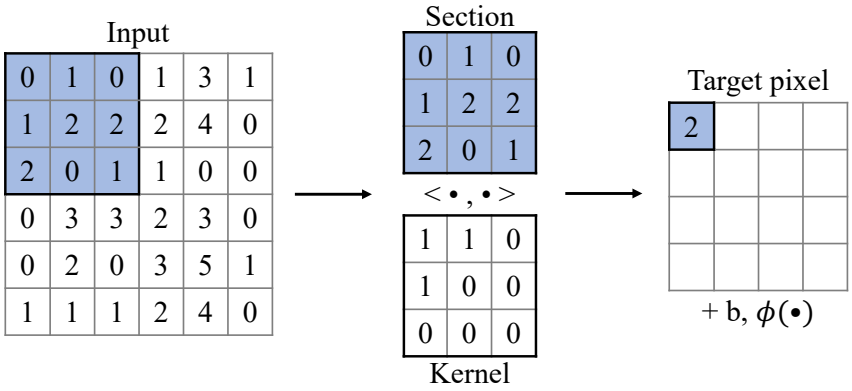

Figure 2.3: Exemplary calculation for the first output pixel of a two-dimensional convolutional layer with kernel size $k = 3$, $O = 1$ channels and size $M = N = 6$. The blue area marks the considered area determined by the kernel size, the blue square on the right shows the position and the value of the result. The missing values are calculated in the subsequent steps by sliding the window.

Additionally, there are different possibilities of how to create a kernel and how to move it over the layer. The stride $S \in \mathbb{N}$ decides by how many positions the kernel is moved each time. The larger the stride, the less overlap there is in the sections. With an increasing stride, the output width and height are decreasing as $M_i = \lfloor \frac{M_{i-1}-k_i}{S} \rfloor + 1$ and $N_i = \lfloor \frac{N_{i-1}-k_i}{S} \rfloor + 1$. An illustration of a sliding window with stride $S = 2$ and the resulting output is given in Fig. 2.4.

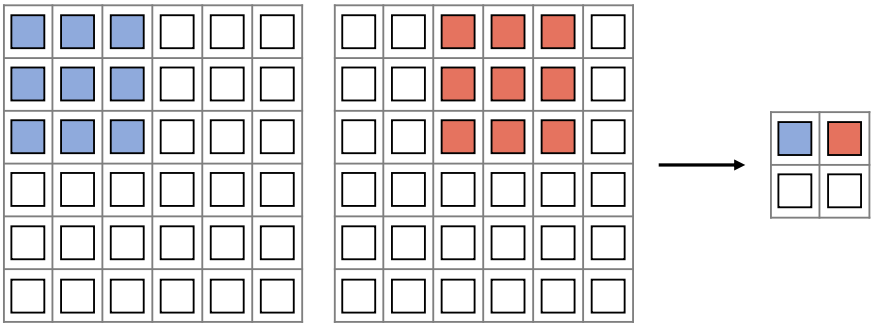

Figure 2.4: Exemplary illustration of stride $S = 2$, kernel size $k = 3$, $O = 1$ channels and size $M = N = 6$. The blue and red squares in the first and second picture show the position of the sliding window (kernel) for the first and the second step. The corresponding squares on the right show the positions of the resulting values.

The dilation distance $D \in \mathbb{N}$ defines the space between the pixels in the kernel. By having a dilation distance larger than one, again the output sizes decrease, as $M_i = \lfloor \frac{M_{i-1}-(k_i-1)\cdot D-1}{S} \rfloor + 1$ and $N_i = \lfloor \frac{N_{i-1}-(k_i-1)\cdot D-1}{S} \rfloor + 1$. The advantage of a dilation distance larger than one is that it allows for an enlarged field to be viewed at in the image, the so-called receptive field, without increasing the computational cost. An example of a kernel with dilation distance $D = 2$ is given in Fig. 2.5.

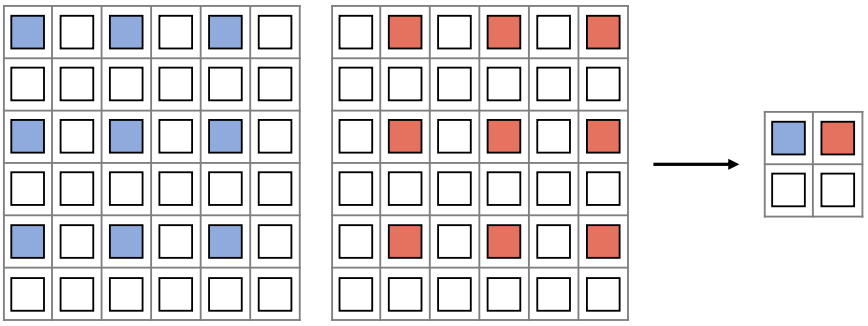

Figure 2.5: Exemplary illustration of a dilated convolution with dilation distance $D = 2$, kernel size $k = 3$, $O = 1$ channels and size $M = N = 6$. The blue and red squares in the first and second picture show the position of the sliding window (kernel) for the first and the second step. The corresponding squares on the right show the positions of the resulting values.

It is also possible to employ zero padding up front moving the kernel, where $P \in \mathbb{N}$ zeros are put around each side of the input in order to obtain an increased output size compared to when no zero padding is applied. The output size evolves to $M_i = \lfloor \frac{M_{i-1} - (k_i - 1) \cdot D - 1 + 2P}{S} \rfloor + 1$ and $N_i = \lfloor \frac{N_{i-1} - (k_i - 1) \cdot D - 1 + 2P}{S} \rfloor + 1$.

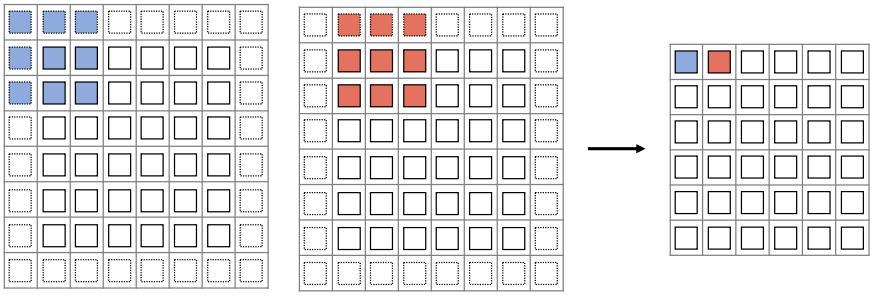

Figure 2.6: Exemplary illustration of zero padding with $P = 1$, kernel size $k = 3$, $O = 1$ channels and size $M = N = 6$. The blue and red squares in the first and second picture show the position of the sliding window (kernel) for the first and the second step. The dotted squares represent the added zero valued pixels. The corresponding squares on the right show the positions of the resulting values.

One option to reduce the layer size is applying a pooling layer, where only the maximum, mean or a similar operation of a section hit by the sliding window is processed in the next layer. Reducing the layer size and reducing the computational complexity of the CNN is important for feature extraction, to also learn features that spread over a larger area than covered by the kernel [O'Shea and Nash 2015]. However, in order to not lose too much information, the amount of channels of the layers increases meanwhile.

Typically, the size of the layers is reduced after applying convolutions to make the neural network learn features. If the output is expected to have the size of the input $\boldsymbol{x}$, e.g. in image segmentation tasks, the output of the last convolutional layer needs to be "upsampled" again. This is done by adding deconvolutional layers to the network. Several possibilities such as zero insertion, where zeros are inserted around adjacent pixels or unpooling, where one pixel is turned into more with the same value, i.e. is "inflated", are given.

In other cases, e.g. classification, the layer type needs to be changed from convolutional to dense layers. This can be simply performed by so-called flatten layers which turn the output of the convolutional layer into one dimension by concatenating the corresponding entries. Another possibility is global average pooling (GAP) layers, which reduce the dimension of the layer to one by averaging over the other dimensions. Further "classical" computations with dense layers can be performed. Since this is not needed in this work, it will not be discussed in detail.

One last important feature of CNNs is skip connections. As already described, vanishing gradients are a problem for layers at the beginning of the neural network. A skip connection provides an alternative path for the forward propagation (evaluation of the neural network) and back propagation and is a common module used in CNNs. The output of an early layer skips several layers and is added or concatenated to the input of a deeper layer, and not only used as the input for the next layer. Thus, the gradients of the parameters in the early layer are also calculated directly with the gradients of the ones in the deeper layer, which counteracts the problem of vanishing gradients.

2.3 Physics-Informed Neural Networks

Physics-informed neural networks (PINN) are also a subtype of neural networks, primarily introduced by Raissi et al. [2019]. PINNs are used to solve partial differential equations (PDE) by extending the loss with a physical component, constructed to satisfy the physical laws given by a PDE. This additional physical component, referred to as the physical loss, calculates and penalizes the deviation from the PDE, i.e. the physical laws, additional to the deviation from available data.

According to Raissi et al. [2019] a PDE considered for solving it via a PINN is generally of the form

$$f(t, \boldsymbol{x}) = \frac{\partial}{\partial t} u(t, \boldsymbol{x}) + \mathcal{N}[u] \stackrel{!}{=} 0, \ \boldsymbol{x} \in \Omega \subseteq \mathbb{R}^d, \ t \in [0, T] \tag{2.7}$$

$$u(0, \boldsymbol{x}) = u_0(\boldsymbol{x})$$

$$u(t, \tilde{\boldsymbol{x}}) = u_{\partial\Omega}(t), \ \tilde{\boldsymbol{x}} \in \partial\Omega. \tag{2.8}$$

$u(t, \boldsymbol{x})$ denotes the solution of the PDE, which is to be approximated by $\hat{u}(t, \boldsymbol{x})$, the neural network, t the temporal and $\boldsymbol{x}$ the spatial coordinates. $\mathcal{N}[\cdot]$ is a differential operator and $u_0(\boldsymbol{x})$ and $u_{\partial\Omega}(t)$ describe the initial and boundary condition(s). Using automatic differentiation on the output of the neural network $\hat{u}(t, \boldsymbol{x})$ yields the derivatives with respect to the input parameters $(t, \boldsymbol{x})$ needed for the calculation of $f(t, \boldsymbol{x})$, since $\hat{u}(t, \boldsymbol{x})$ and $f(t, \boldsymbol{x})$ share the same parameters with exception to different activation functions due to $\mathcal{N}[\cdot]$ [Raissi et al. 2019]. In order to optimize these shared parameters, i.e. find an approximation for $u(\boldsymbol{x})$, the physical loss is defined as the MSE of $f(t, \boldsymbol{x})$, namely

$$L_f\left(\{t_j^f, \boldsymbol{x}_j^f\}_{j=1}^{N_f}\right) = \frac{1}{N_f} \sum_{j=1}^{N_f} |f(t_j^f, \boldsymbol{x}_j^f)|^2, \tag{2.9}$$

where $\{t_j^f, \boldsymbol{x}_j^f\}_{j=1}^{N_f} \subset [0, T] \times \Omega$ corresponds to sampling points, on which the physical loss is evaluated. It ensures that $\hat{u}(t, \boldsymbol{x})$ satisfies the structure given by the PDE (2.7) and its boundary conditions (2.8).

This physical loss is only one component of the loss. The total loss $L_{\hat{u},f}$ consists

of the physical loss and the empirical loss

$$L_{\hat{u},f}\left(\{t_j^{\hat{u}}, \boldsymbol{x}_j^{\hat{u}}\}_{j=1}^{N_{\hat{u}}}, \{t_j^f, \boldsymbol{x}_j^f\}_{j=1}^{N_f}\right) = L_{\hat{u}}\left(\{t_j^{\hat{u}}, \boldsymbol{x}_j^{\hat{u}}\}_{j=1}^{N_{\hat{u}}}\right) + L_f\left(\{t_j^f, \boldsymbol{x}_j^f\}_{j=1}^{N_f}\right). \quad (2.10)$$

As already introduced in Section 2.1, the empirical loss is defined as

$$L_{\hat{u}}\left(\{t_j^{\hat{u}}, \boldsymbol{x}_j^f\}_{j=1}^{N_f}\right) = \frac{1}{N_{\hat{u}}}\sum_{j=1}^{N_{\hat{u}}}(\hat{u}(\boldsymbol{x}_j) - u(\boldsymbol{x}_j))^2 \quad (2.11)$$

and corresponds to the deviation of the neural network from the training data, in this case the initial and boundary data $\{t_j^{\hat{u}}, \boldsymbol{x}_j^{\hat{u}}\}_{j=1}^{N_{\hat{u}}}$ of the PDE. It ensures that $\hat{u}(t, \boldsymbol{x})$ satisfies the initial and boundary condition(s).

With exception to the loss function $L_{\hat{u},f}$ and its evaluation, the procedure of training and, in this case, solving the given PDE, is identical to the one already described in Section 2.1. According to Raissi et al. [2019], even a small amount of data points is sufficient to obtain a satisfying approximation $\hat{u}(t, \boldsymbol{x})$ for the solution $u(t, \boldsymbol{x})$ of the PDE.

2.4 General Problem Definition

This work aims to predict the stationary groundwater temperature field T for a bounded area with multiple groundwater heat pumps (GWHP) depending on the stationary, spatially varying groundwater velocity field. It is assumed that the temperature is dominated by the advection term in the Darcy flow equations (mass and energy balance):

$$\begin{aligned} Q_m &= \nabla \cdot (\rho \boldsymbol{q}) \\ &= \nabla \cdot (\rho K \nabla P), \quad (2.12) \\ Q_e &= \nabla \cdot (\rho c\, T \boldsymbol{q}) - \kappa \Delta T \\ &= \nabla \cdot (\rho c\, T K \nabla P) - \kappa \Delta T, \quad (2.13) \end{aligned}$$

with the water density $\rho = 997\frac{kg}{m^3}$, the Darcy flow $\boldsymbol{q}$, determined by the permeability field and the pressure gradient of the subsurface K and ∇P, the specific thermal capacity of water $c = 4190\frac{J}{kg \cdot K}$, the thermal conductivity of the subsurface $\kappa = 0.5\frac{W}{m \cdot K}$ and the mass and energy source terms $Q_m = 0.0\frac{kg}{m^3 \cdot s}$ and

$Q_e = 0.0\frac{W}{m^3}$ everywhere, except for the GWHP locations, where $Q_e = 12.5\frac{W}{m^3}$, as the power of one GWHP with a volume of $16m^3$ is estimated to be $200W$.

Thus, the energy balance (2.13) is the governing PDE to describe the addressed problem. This PDE is to be solved by a PINN and therefore needs to be put into the form of Eq. (2.7), introduced in Section 2.3. The solution of the PDE is the temperature field $T(\boldsymbol{x})$ with the spatial coordinates $\boldsymbol{x} \in \mathbb{R}^2$. This is the temperature field that needs to be approximated by the neural network $\hat{T}(\boldsymbol{x})$. As this problem describes a stationary groundwater flow and temperature field, it is only dependent on two spatial parameters, not on time. This leads to the following setting:

$$
\begin{aligned}
f(\boldsymbol{x}) &= \mathcal{N}[T(\boldsymbol{x})] \overset{!}{=} 0, \ \boldsymbol{x} \in \Omega \subseteq \mathbb{R}^2 \\
\mathcal{N}[T(\boldsymbol{x})] &= \nabla \cdot (\rho c \, T\boldsymbol{q}) - \kappa \Delta T - Q_e \\
&= \rho c \, \left(T(q_{x_1}^{(x_1)} + q_{x_2}^{(x_2)}) + T_{x_1} q^{(x_1)} + T_{x_2} q^{(x_2)} \right) \\
&\quad - \kappa \, (T_{x_1 x_1} + T_{x_2 x_2}) - Q_e
\end{aligned}
\tag{2.14}
$$

$$
T(\boldsymbol{x} \in \partial\Omega) = 10. \tag{2.15}
$$

The reason why the solution $T(\boldsymbol{x})$ to this PDE is approximable by a neural network $\hat{T}(\boldsymbol{x})$ is given by the universal approximation theorem. In theory, there exists a weak solution to a weak formulation of the given problem that can be approximated by a continuous function [Masud and Hughes 2002]. Restricting the input $\boldsymbol{x}$ on a compact area and remembering that a neural network with at least one hidden layer and a nonlinear activation function in it can approximate any continuous function on compact input sets leads to the fact that a neural network $\hat{T}(\boldsymbol{x})$ with given loss functions is able to approximate the solution $T(\boldsymbol{x})$ of the PDE (2.13).

3 Data Generation

The datasets necessary to train and to assess the quality of the neural networks were created using PFLOTRAN [Hammond et al. 2014]. PFLOTRAN is a reactive flow and transport model for the description of subsurface processes that provides high-fidelity simulations.

PFLOTRAN solves a boundary value problem with a finite volume subsurface solver. It identifies the pressure, velocity and temperature field depending on given pressure boundary conditions and a permeability field. In general, the simulations are performed on a 2D structured grid with directions x_1 and x_2. One cell of the grid represents an area of size $2m \times 2m \times 1m$, neglecting depth from now on. A groundwater heat pump (GWHP) is modeled by injecting $0.05\frac{kg}{s}$ of a fluid at 15°C on 2×2 cells. In order to achieve a pseudo stationary solution, the simulated time is 720 days in total, where the initial temperature of the domain is set to 10°C. The edges are modeled as Dirichlet boundary conditions of 10°C.

3.1 Permeability Field and Pressure Boundary Conditions

The simulation with PFLOTRAN takes a permeability field and pressure boundary conditions as an input. In order to generate various velocity and temperature fields, different permeability fields and pressure boundary conditions were created.

A permeability field is generated by assigning values of the uniform distribution on $[2.1 \cdot 10^{-9}, 4.1 \cdot 10^{-8}]$ randomly at different points on a regular square grid (4×4, 6×6, and 8×8) across the domain. Via the radial basis function (RBF) interpolation method with global thin-plate splines basis functions the permeability values are then mapped to the grid of PFLOTRAN [Leiteritz et al. 2022].

The pressure boundary conditions are applied by assigning two values for the

pressure gradient, one in each direction x_1 and x_2.

An exemplary permeability and pressure field with the resulting temperature field are given in Fig. 3.1.

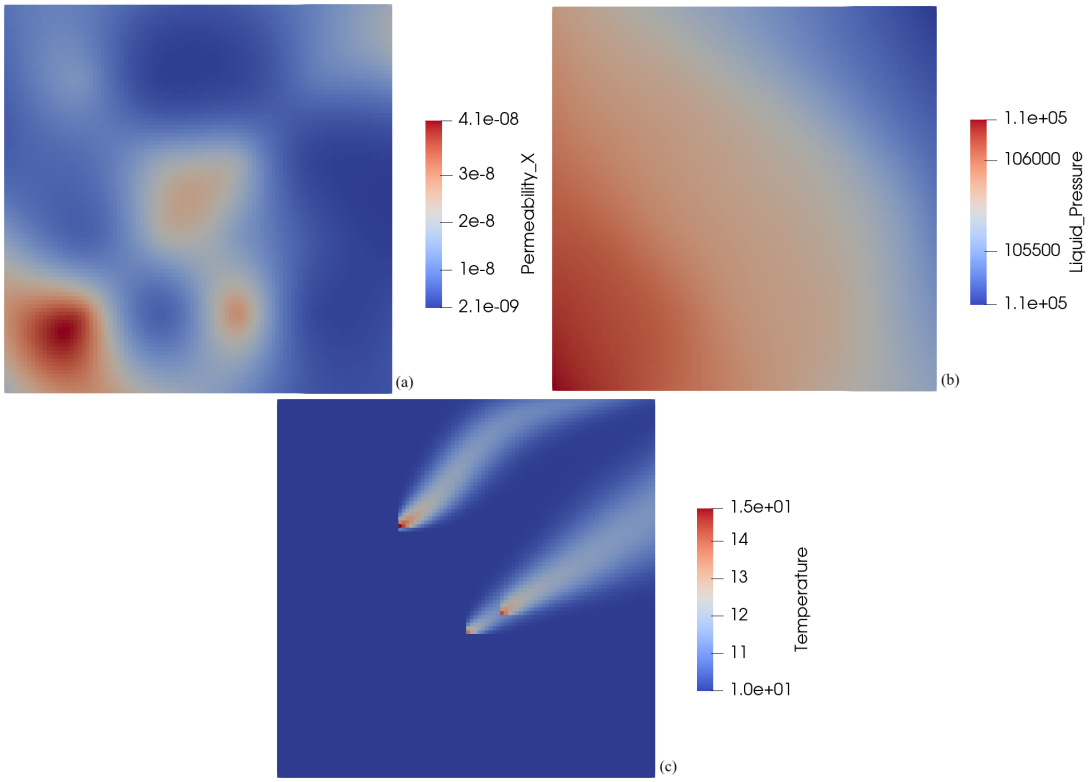

Figure 3.1: Exemplary (a) permeability field, (b) pressure field and (c) temperature field.

3.2 Heat Pump Field

In a large-scale scenario with multiple GWHPs, the GWHPs must be positioned in the area prior to running the simulation. Certain rules need to be established for this to ensure a smooth running of the framework. First, the GWHPs are not allowed to overlap. More precisely, the distance between two different GWHPs must be at least one cell. Additionally, in order to run the local surrogate model, an area of 64×64 cells needs to be cut out around every single GWHP. Therefore, a spatial buffer around each GWHP must be provided to allow this excision. Since

the size of the GWHP is 2×2 cells, the distance between the GWHP and the corresponding boundary must be at least 31 cells. Figure 3.2 illustrates the rules for positioning the GWHPs for an example of three GWHPs on an area of 100×100 cells. The light blue area corresponds to the forbidden zone at the boundaries of the whole domain to allow the excision of an area around each GWHP. This means that the first GWHP (dark blue) can be placed everywhere within the white square. The second GWHP can be placed everywhere within the white square with exception for the middle blue cells around the first GWHP (and the cells of the first GWHP itself), which mark the forbidden cells to avoid overlapping GWHPs. Accordingly, each new GWHP can be positioned everywhere in the remaining white area. The boundaries of the excision areas around each GWHP are additionally marked in dark blue in Fig. 3.2(b) to outline the size of the excision area and to illustrate why the light blue zone needs to be of such large size.

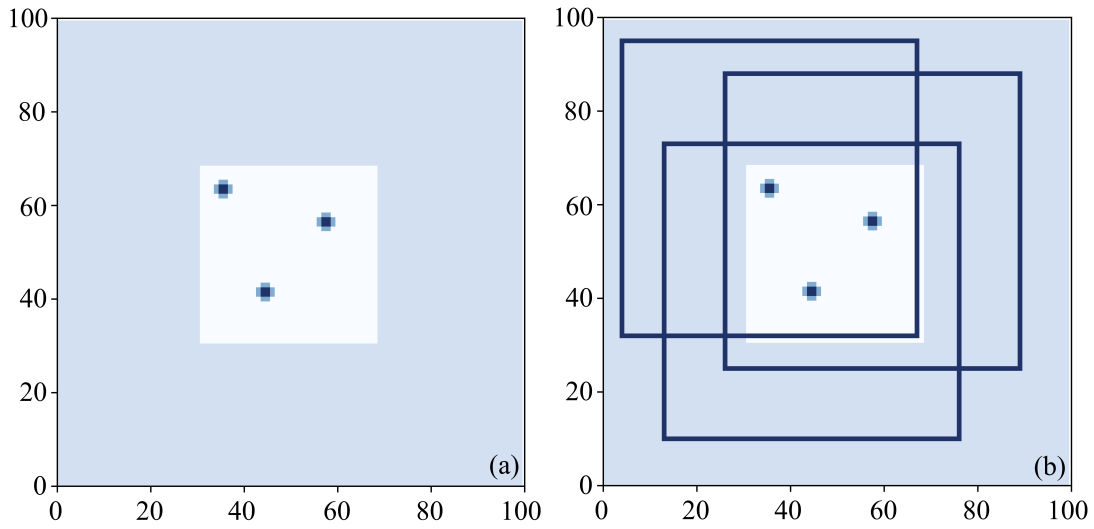

Figure 3.2: (a) Rules for positioning GWHPs. GWHPs are marked in dark blue, the middle blue cells around each GWHP mark the forbidden cells to avoid overlapping GWHPs. The light blue area corresponds to the forbidden zone at the boundaries of the whole domain. (b) Excision boundaries are additionally marked in dark blue.

3.3 Resulting Datasets

As already mentioned, the generation of the datasets is necessary to train the local surrogate model and to assess the quality of both neural networks. This framework includes the use of two models - a CNN as a local surrogate model and a PINN to solve the governing PDE afterwards in order to correct the stitched together local predictions.

Thus, the first dataset, the "local dataset" is used to train and assess the quality of the local surrogate model, which requires only one GWHP at the center of a grid with 64×64 cells. The second dataset, the "global dataset" is needed to determine the framework's ability to predict the temperature across a large area with multiple GWHPs and to assess the quality of the prediction. Both datasets consist of two velocity fields, the two components $\left(q^{(x_1)}, q^{(x_2)}\right)^{\mathrm{T}}$ of the Darcy flow $\boldsymbol{q}$, and the corresponding temperature field T. An example for both datasets is shown in Fig. 3.3. The size of a sample in the local dataset (example in Fig. 3.3(a)) is fixed to 64×64 cells, as the local area around a single GWHP is fixed to 64×64 cells. Additionally, there is always only one centered GWHP in each sample. The size of a sample in the global dataset depends on the domain size chosen for investigation. In this case, an arbitrary amount of GWHPs can be positioned. Figure 3.3(b) shows an example with 10 GWHPs for a size of 250×250 cells, which corresponds to $500m \times 500m$.

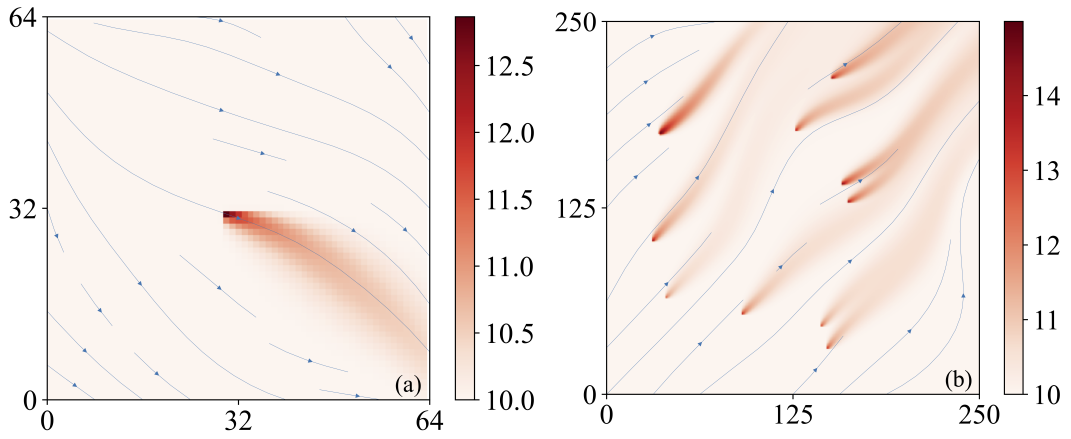

Figure 3.3: Temperature prediction by PFLOTRAN for an example of the (a) local (used for training the surrogate model) and (b) global dataset (used as ground truth). Arrows mark the velocity field.

4 Methods

4.1 CNN Based Surrogate Modeling of Local Groundwater Heat Pump Temperature Approximation

The first step for a large-scale groundwater heat pump (GWHP) temperature prediction is the local temperature prediction around a single GWHP. For this purpose, Leiteritz et al. [2022] developed a surrogate model, which takes the stationary, spatially varying velocity field of 64×64 cells as an input and predicts the developed temperature field when a single GWHP is added to the center of the domain.

4.1.1 Model

For the structure of the surrogate model a CNN was chosen, more precisely, a modification of the "TurbNet" by Thuerey et al. [2020], a variation of the "U-Net" by Ronneberger et al. [2015].

The neural network has two input channels corresponding to the two components $\left(q^{(x_1)}, q^{(x_2)}\right)^{\mathrm{T}}$ of the Darcy flow $\boldsymbol{q}$, both with a size of 64×64 cells. The output of the network is a single channel, also of size 64×64 cells, for the predicted temperature field. The network architecture itself is similar to a "U-Net". The input passes through increasingly convolving convolutional layers with an increasing number of channels up to a bottleneck convolution. From this point on, the channels are de-convolved in reverse order and a symmetrical manner using deconvolutional layers. Furthermore, the network possesses skip connections, connecting the layers in the convolving part to the ones of the same shape in the de-convolving part. In this case, the outputs of the convolutional layers are added feature-wise to the ones of the deconvolutional layers. Each layer contains a rectified linear unit (ReLU) activation function. The dilation distance is $D = 1$

for all layers, the stride for the convolutional layers is $S = 2$ and for the deconvolutional layers $S = 1$. Zero padding is $P = 1$ for all layers, except for the bottleneck convolutions where $P = 0$. The difference to the "TurbNet"-structure by Thuerey et al. [2020] is the increased size of the bottleneck. Leiteritz et al. [2022] assumed that a single pixel value in the bottleneck no longer contains much relevant information, and therefore some model complexity can be saved. Additional model complexity was saved by lowering the number of channels in each layer. A schematic illustration of the TurbNet architecture is given in Fig. 4.1.

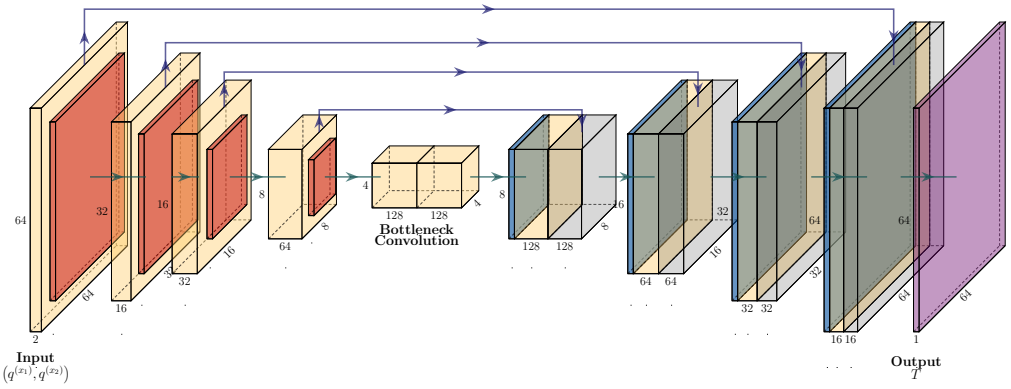

Figure 4.1: Schematic illustration of the TurbNet architecture. Each layer has a ReLU activation function.

4.1.2 Data

The dataset necessary to train the surrogate model was created using PFLOTRAN, as already mentioned in Section 3. In this case, the data was generated on a 2D structured grid of 64×64 cells with a total of 4,096 cells. As one cell covers an area of $2m \times 2m$, the whole domain is of size $128m \times 128m$. Since the surrogate model predicts the local temperature around a single GWHP, no placement of different GWHPs is necessary, which implies that a single GWHP of size 2×2 cells is positioned in the center of the domain. The output of each PFLOTRAN simulation is, as already stated in Section 3, the developed pseudo stationary velocity field and the corresponding temperature prediction.

After the generation of the data, some preprocessing took place in order to

support the training of the surrogate model. First, the temperature field data T is centered. This means, as the domain temperature is set to 10°C and the temperature at every location is intuitively at maximum 15°C, since the injected water is 15°C, a total of 12.5°C is subtracted of each value of the temperature field T. Afterwards, the values of the temperature field T are normalized to a range of $[-1, 1]$, as well as the velocity fields q_{x_1}, q_{x_2} separately. According to previous studies [e.g. LeCun et al. 2012, Santurkar et al. 2018], centering often supports the training process. The training dataset was also augmented by randomly rotating the input and target images in order to improve the robustness of the resulting CNN.

4.1.3 Performance

The model was trained on a dataset of 239 samples, which was enriched by 720 samples rotating the velocity and temperature fields, resulting in a total of 959 training samples. 192 samples of this dataset were used for validation. Additional training details are given in Appendix A. A surrogate model that achieved a high level of compliance between the surrogate predictions and the corresponding PFLOTRAN simulation, taken as ground truth was developed by Leiteritz et al. [2022]. The prediction results, the target ground truths (corresponding PFLOTRAN simulations) and the point-wise error fields between both (in °C) for four different test samples are shown in Fig. 4.2. As the first three examples in Fig. 4.2 show typical results, one can see that the surrogate predictions qualitatively match the target thermal plumes in direction and shape. Also, the first two examples match very well in magnitude. The last example shows that there are still scenarios for which the surrogate does not perform too well. The error ranges in [−2°C, 2°C], however, typically the greatest quantitative error is at the position of the GWHP and the error of the thermal plume is significantly lower.

One important advantage is that the inference time of the model with $\leq 50ms$ is very low. Therefore, it yields a good basis for a large-scale scenario with multiple GWHPs with the goal of a quasi-real time prediction.

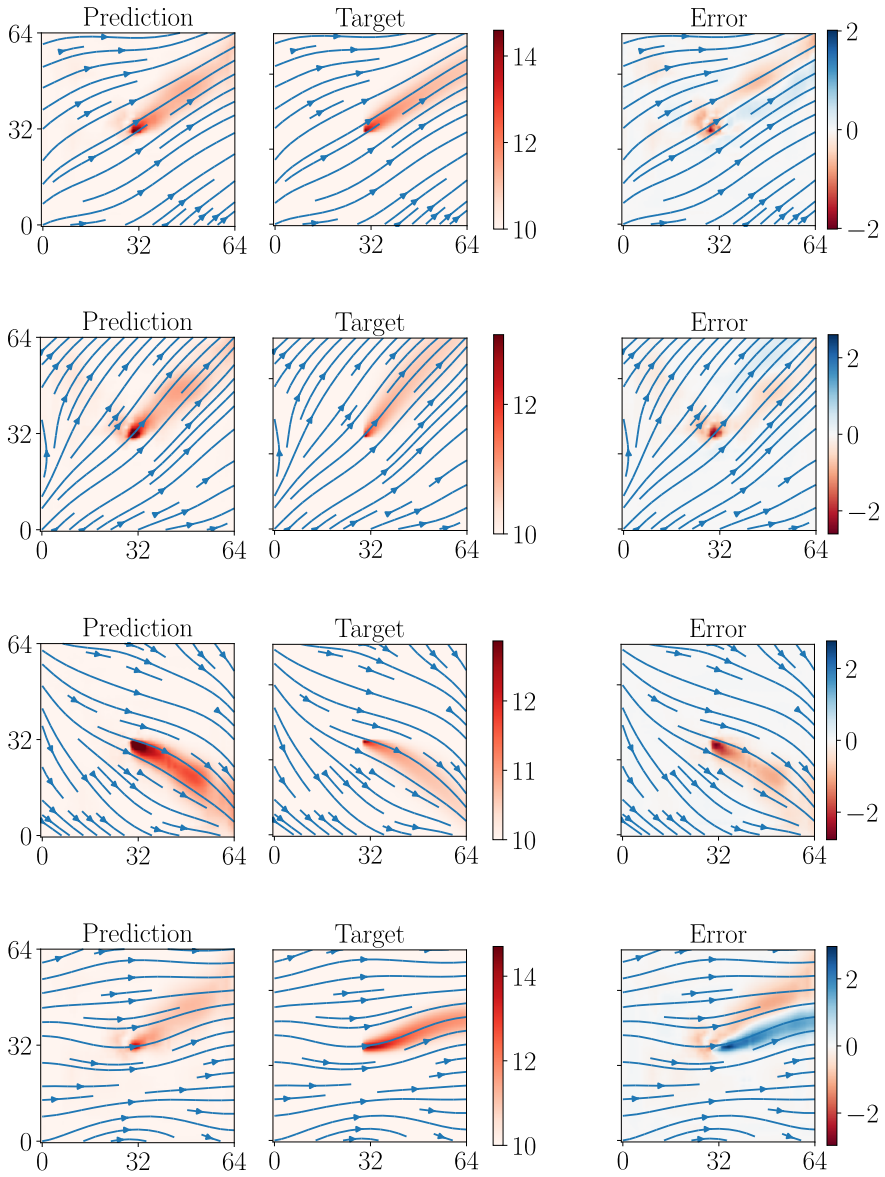

Figure 4.2: Prediction results, target ground truths (corresponding PFLOTRAN simulation) and point-wise error fields between both for four different test samples (three typical results and one outlier), all in °C.

4.2 Building a Global Initial Solution Using Local Surrogates

The next step for a large-scale GWHP temperature prediction is to embed the surrogate model into the framework. The pipeline takes the velocity field, the two components $\left(q^{(x_1)}, q^{(x_2)}\right)^{\mathrm{T}}$ of the Darcy flow $\boldsymbol{q}$, and the locations of the GWHPs as an input. In order to run the local surrogate model, the local velocity domain of 64×64 cells around each GWHP is extracted. An exemplary illustration for a constant velocity field is given in Fig. 4.3.

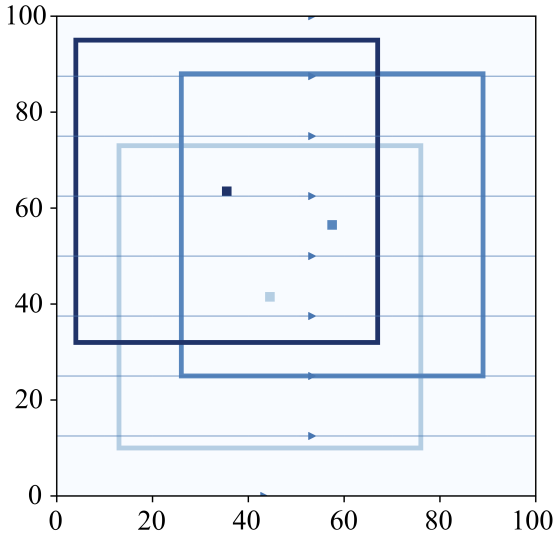

Figure 4.3: Illustration of the cutout velocity domains around each GWHP in a domain of 100×100 cells for a constant velocity field in order to run the local surrogate model, here for three GWHPs in total.

Next, the local temperature predictions are created by evaluating the surrogate model for each GWHP on a cutout domain of 64×64 cells separately. Exemplary local temperature predictions for each GWHP and a constant velocity field are shown in Fig. 4.4. Already here it becomes apparent that the local surrogate model does not yet make ideal predictions, since at a constant velocity in x_1-direction the thermal plume deviates from this direction contrary to expectations.

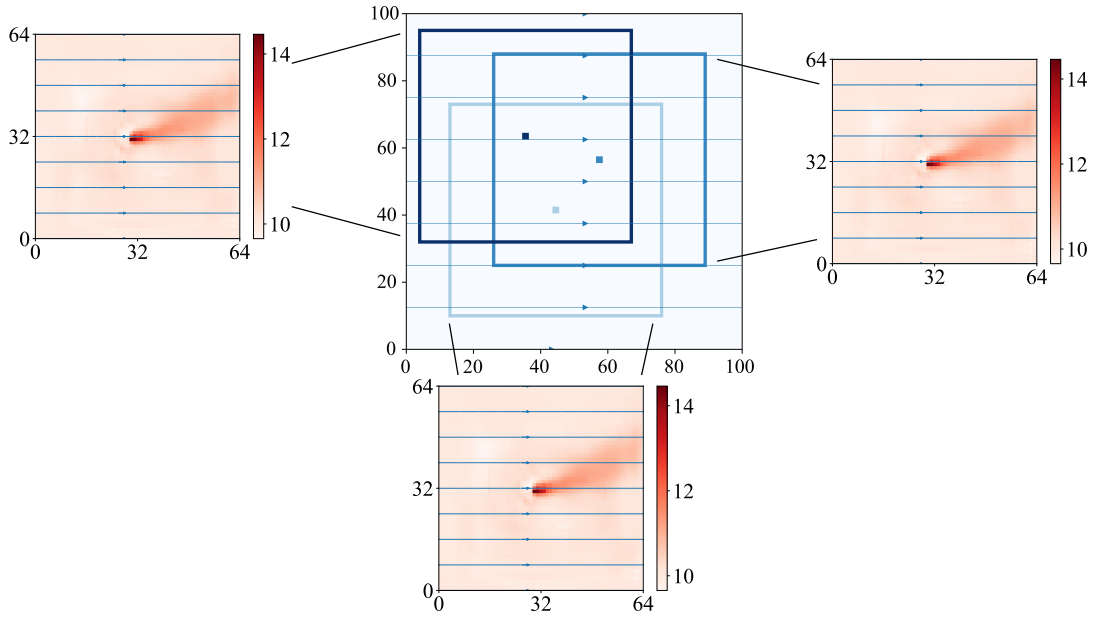

Figure 4.4: Exemplary local temperature predictions for each GWHP on a cutout domain of 64×64 cells for a constant velocity field.

Afterwards, the results need to be stitched together to create a global initial solution $\tilde{T}(\boldsymbol{x})$. Two questions arise: (1) What happens to the cells, where no local prediction is determined (as not all cells lie within the local area of a GWHP)? (2) What happens if there is more than one local prediction for the same cell (i.e. local areas of two or more GWHPs overlap)?

The first question can be answered easily. If no local temperature prediction is available for a cell, it means that so far no GWHP has an influence on the temperature of this cell. Therefore, the temperature is set to the standard groundwater temperature of this setup, 10°C.

For the second problem, different approaches are to be investigated. The first one is to take the maximum out of all temperature predictions made for the corresponding cell, the second one to take the mean. The third one considers the direction of the flow. Here, the local prediction of the GWHP is taken, which is encountered first when the groundwater flow is traced upstream. Further computational details of the last option are given in Appendix B.

Figure 4.5 shows an example of the different stitching methods for the already

given scenario with a constant velocity field.

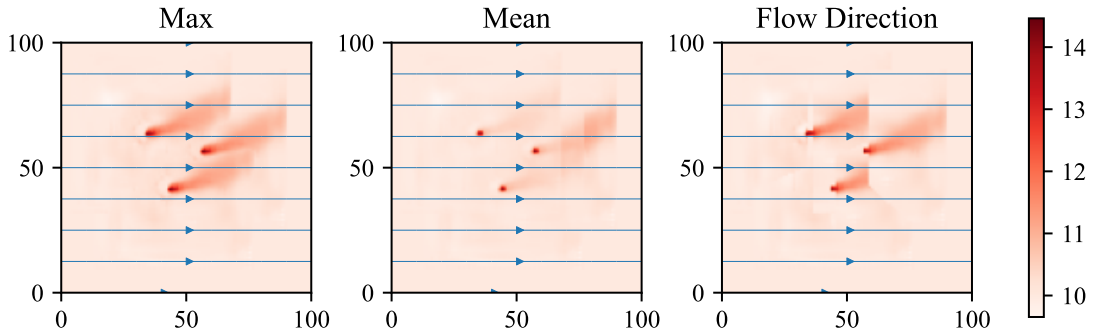

Figure 4.5: Illustration of the three different approaches of stitching local predictions together, maximum, mean and flow direction to obtain a global initial solution.

4.3 Correction of Initial Solution Using a PINN

After stitching the local predictions together, a global initial solution $\tilde{T}$ is known for each cell, i.e. for each discrete position $\boldsymbol{x}$ within the domain Ω. Mathematically speaking, the global initial solution is a function

$$\tilde{T} : \tilde{\Omega} \subset \mathbb{N}^2 \rightarrow \mathbb{R}, \tag{4.1}$$

where $\tilde{\Omega}$ is the integer discretization of Ω.

However, this global initial solution does not obey any physical laws or boundary conditions. The idea is to train a physics-informed neural network (PINN) in order to obtain a global solution which obeys physical laws and boundary conditions, i.e. approximates the function $T(\boldsymbol{x})$ that solves the PDE given in Eq. (2.14) with boundary condition (2.15). Accordingly, the PINN $\hat{T}(\boldsymbol{x})$ predicts the temperature T at any given position $\boldsymbol{x}$ in the domain Ω. A convenient side effect of this step is that it is a mesh-free method. This implies that the temperature, after training, can be predicted at any given position $\boldsymbol{x}$ within the domain Ω and is no longer restricted on integer values for $\boldsymbol{x}$, which means

$$\hat{T} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}. \tag{4.2}$$

In order to not start from scratch, but use the global initial solution for the PINN, training is split into two parts, a pre-fit and a "real" physics-informed training. In the first part, the neural network is trained to learn the global initial solution. For this purpose, a pre-fit training dataset is extracted from the global initial solution. N_{PF} continuous-valued positions within the domain Ω are selected and the corresponding training dataset results in $\{\boldsymbol{x}_j^{\tilde{T}}, \tilde{T}(\lfloor \boldsymbol{x}_j \rfloor)\}_{j=1}^{N_{PF}}$. The values of the positions need to be rounded off as the global initial solution $\tilde{T}(\boldsymbol{x})$ is a function of only integer values. The positions $\{\boldsymbol{x}_j^{\tilde{T}}\}_{j=1}^{N_{PF}}$ are selected in such a manner that half of the preselected amount of points N_{PF} are sampled uniformly randomly around the positions of the GWHPs. It can happen that these areas overlap and that the density of the sampling points is higher in these parts than in others. $\frac{1}{4}N_{PF}$ points are sampled equidistantly on the whole domain Ω. The other $\frac{1}{4}N_{PF}$ points left are sampled at the boundary of the domain. However, the target value $\hat{T}$ here is directly set to the wanted boundary values for $\hat{T}$ instead of the values of the global initial solution $\tilde{T}(\boldsymbol{x})$ in order to accelerate the training process and not "train points twice", where the exact value is already known. An illustration of the sampling of N_{PF} pre-fit points is shown in Fig. 4.6.

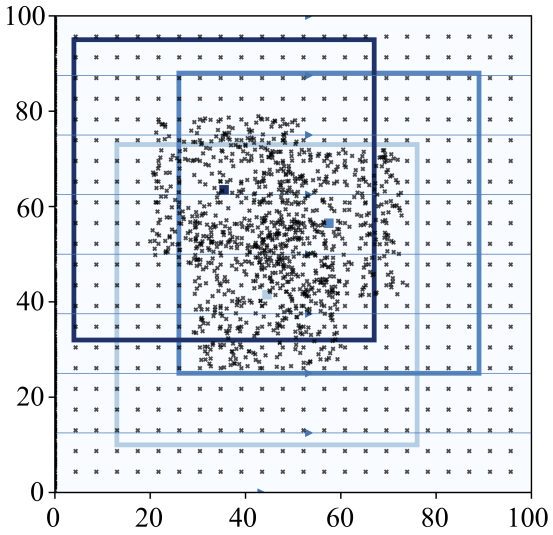

Figure 4.6: Illustration of the sampling of $N_{PF} = 2{,}000$ pre-fit points, here for three GWHPs.

The loss function for this part is

$$L_{\hat{T}}^{PF}\left(\{\boldsymbol{x}_j^{\tilde{T}}\}_{j=1}^{N_{PF}}\right) = \frac{1}{N_{PF}}\sum_{j=1}^{N_{PF}}(\hat{T}(\boldsymbol{x}_j) - \tilde{T}(\lfloor \boldsymbol{x}_j \rfloor))^2. \tag{4.3}$$

After training the neural network with this loss function, it is able to approximate the global initial solution. However, one can clearly see that this neural network is not yet a PINN as no physical laws and boundary conditions are considered. Therefore, after either reaching a certain loss limit ϵ or a total of m epochs, i.e. letting the neural network learn the global initial solution in the pre-fit phase, the loss function is changed to

$$L_{\hat{T},f}\left(\{\boldsymbol{x}_j^{\hat{T}}\}_{j=1}^{N_{BC}}, \{\boldsymbol{x}_j^{f}\}_{j=1}^{N_f}\right) = L_{\hat{T}}\left(\{\boldsymbol{x}_j^{\hat{T}}\}_{j=1}^{N_{BC}}\right) + L_f\left(\{\boldsymbol{x}_j^{f}\}_{j=1}^{N_f}\right) \tag{4.4}$$

with

$$L_{\hat{T}}\left(\{\boldsymbol{x}_j^{\hat{T}}\}_{j=1}^{N_{BC}}\right) = \frac{1}{N_{BC}}\sum_{j=1}^{N_{BC}}(\hat{T}(\boldsymbol{x}_j) - T(\boldsymbol{x}_j))^2 \tag{4.5}$$

$$\text{and } L_f\left(\{\boldsymbol{x}_j^{f}\}_{j=1}^{N_f}\right) = \frac{1}{N_f}\sum_{j=1}^{N_f}|f(t_j^f, \boldsymbol{x}_j^f)|^2, \tag{4.6}$$

an empirical and a physical component and thus, the learning phase switched to "real" physics-informed training.

$\hat{T}(\boldsymbol{x})$ denotes the predicted temperature, i.e. the output of the neural network and $T(\boldsymbol{x})$ the analytical solution. $T(\boldsymbol{x})$ of course is unknown, but the empirical loss is only evaluated at the boundary, where the values for T are given by the boundary condition in (2.15). f corresponds to the PDE residual, given in Eq. (2.14). $\{\boldsymbol{x}_j^{\hat{T}}, T(\boldsymbol{x}_j)\}_{j=1}^{N_{BC}}$ is the dataset where $L_{\hat{T}}$ is evaluated on N_{BC} randomly selected points at the boundaries, where boundary conditions are given. The dataset $\{\boldsymbol{x}_j^{f}\}_{j=1}^{N_f}$ consists of N_f points within the domain Ω on which L_f is evaluated. Different methods for drawing these sampling points, explained in Section 4.3.2, are to be explored.

For the evaluation of the physical loss, derivates of the Darcy velocity $\boldsymbol{q}$ are needed, $q_{x_1}^{(x_1)}$ and $q_{x_2}^{(x_2)}$. As the velocities are only available for integer-valued

positions, these derivatives are approximated by central difference quotients

$$q_{x_1}^{(x_1)} \approx \frac{q^{(\lfloor x_1 \rfloor + 1)} - q^{(\lfloor x_1 \rfloor - 1)}}{2 \cdot 2m} \tag{4.7}$$

$$\text{and } q_{x_2}^{(x_2)} \approx \frac{q^{(\lfloor x_2 \rfloor + 1)} - q^{(\lfloor x_2 \rfloor - 1)}}{2 \cdot 2m}. \tag{4.8}$$

If the integer-valued position corresponds to a boundary cell, one-sided difference quotients are employed.

At this point it is important to mention that $L_{\hat{T}}^{PF}$ is completely replaced by $L_{\hat{T},f}$ after m epochs and therefore does not influence further training in any manner. Introducing $L_{\hat{T},f}$ makes the neural network a proper PINN as physical laws (within f) and boundary conditions are learned to be obeyed. For simplicity, the whole neural network is further referred to as PINN (independent of being in the pre-fit or physics-informed, i.e. "correction" training phase) as it is the same network.

After training the PINN for another n epochs, a final global solution $\hat{T}(\boldsymbol{x})$ obeying physical laws and boundary conditions is obtained. The number of "correction epochs" n is set at the beginning, the number of epochs for the pre-fit phase is determined by reaching a limit ϵ for $L_{\hat{T}}^{PF}$ or a maximum of pre-fit epochs m_{max}, depending on which case occurs earlier.

4.3.1 Model

The structure of the PINN is a basic dense neural network with five hidden layers and 20 neurons each. Each layer contains a TanH activation function, the activation function of the output layer is linear. The input layer consists of two neurons, one for position x_1 and one for position x_2. A learning rate of 10^{-3} was chosen. The output layer consists of one neuron, the predicted temperature $\hat{T}$ at position $\boldsymbol{x} = (x_1, x_2)^{\mathrm{T}}$. A schematic illustration is given in Fig. 4.7.

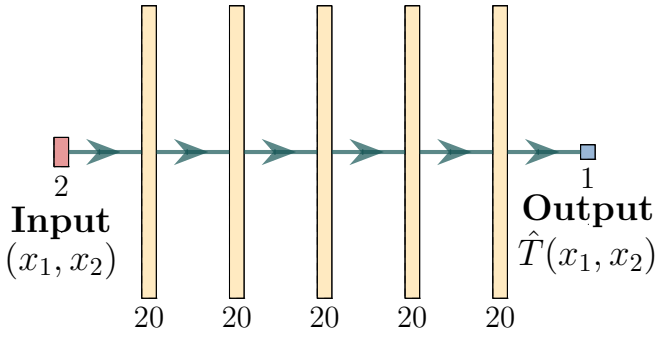

Figure 4.7: Schematic illustration of the PINN architecture. Each hidden layer has a TanH activation function.

In a first round, the number of correction epochs was chosen to be $n = 100$ and in a second one $n = 500$ epochs to examine whether the additional training time is worthwhile in terms of improving the prediction's quality. The maximum number of epochs in pre-fit phase was chosen to be $m_{max} = 30{,}000$, the limit for the loss value $L_{\hat{T}}^{PF}$ is $\epsilon = 10^{-3}$. Additional training details are given in Appendix C.

The loss function $L_{\hat{T},f}$ is the sum of two terms, $L_{\hat{T}}$ and L_f. An equal and fixed weighting of these two terms is inconvenient from a numerical point of view, as the gradients of the two losses might be of different sizes which leads to one being completely disregarded in the learning process [Wang et al. 2021]. This has the consequence that the global solution $\hat{T}(\boldsymbol{x})$ either does not meet the boundary conditions or solve the PDE within the domain Ω. An adaptive weighting γ of the empirical loss $L_{\hat{T}}$ during the training process helps to overcome this issue. The total loss is composed of

$$L_{\hat{T},f} = \gamma \cdot L_{\hat{T}} + L_f. \tag{4.9}$$

The Annealing Algorithm proposed by Wang et al. [2021] for the computation of the weight γ was used, however, slightly adapted, as the given configuration showed better results in application. It is shown in Alg. 1. The weight γ is adjusted to the size of the gradients during training and thus, the gradients are approximately balanced. The annealing rate $\alpha_{ANN} \in [0, 1]$ determines, whether

Algorithm 1 Annealing Algorithm

Require: $L_{\hat{T}}, L_f, \gamma$

$\hat{\gamma} \leftarrow \frac{\text{mean}(|\nabla L_f|)}{\text{mean}(|\nabla L_{\hat{T}}|)}$

$\gamma \leftarrow (1 - \alpha_{ANN}) \cdot \gamma + \alpha_{ANN} \cdot \hat{\gamma}$

return $\gamma \cdot L_{\hat{T}} + L_f, \gamma$

the weights from previous epochs are taken into account (low α_{ANN}) or quasi neglected (high α_{ANN}). For this application $\alpha_{ANN} = 0.8$ was chosen. Using this procedure, high values of γ were obtained, causing the loss $L_{\hat{T},f}$ to increase immensely, which jeopardized stable training. To ensure stable training, the composition of the total loss was adapted to

$$L_{\hat{T},f} = L_{\hat{T}} + \frac{1}{\gamma} \cdot L_f. \tag{4.10}$$

The weight γ was computed first at the beginning of the physics-informed correction training phase and afterwards re-evaluated every epoch.

4.3.2 Different Sampling Methods for Training of the PINN

For the evaluation of the physical loss L_f, N_f sampling points need to be drawn from the domain Ω. Two different possibilities are to be examined in order to find out whether a specific choice of sampling points improves the prediction quality.

For the first possibility N_f points are drawn equidistantly from the whole domain Ω. An illustration is shown in Fig. 4.8(a).

For the second investigated scenario, the amount of N_f sampling points is divided into two subsets. $\frac{1}{4}N_f$ sampling points are also drawn equidistantly from the whole domain Ω in order to capture "the overall picture". The other $\frac{3}{4}N_f$ sampling points are drawn uniformly randomly around the local, previously cutout, areas of the GWHPs, since most of the correction of the global initial solution needs to be done in these parts of the domain. The first sampling method is further referred to as "equidistant sampling method", the second one as "mixed sampling method". An illustration of both sampling methods is shown in Fig. 4.8(b).

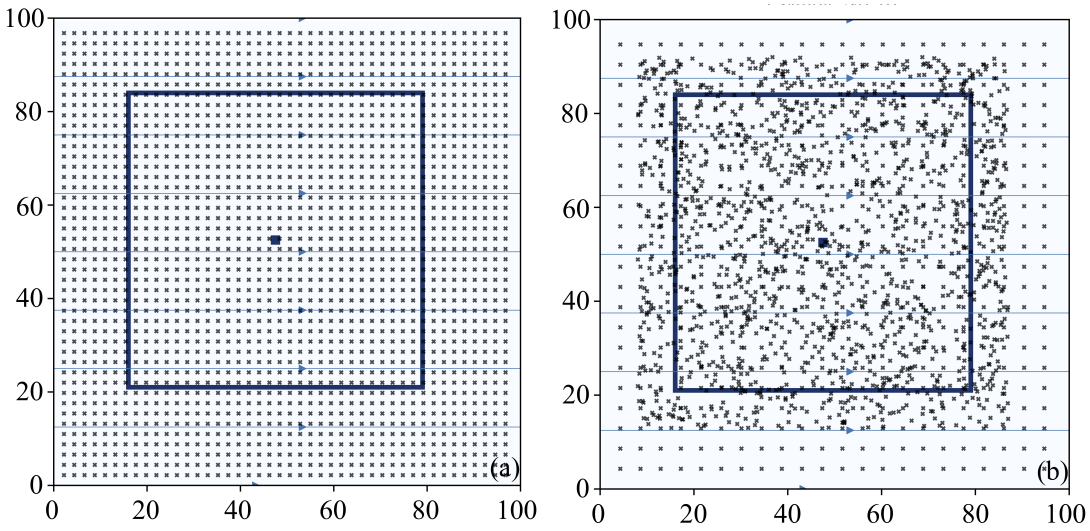

Figure 4.8: Illustration of sampling $N_f = 2{,}000$ "physical" points, using (a) the equidistant sampling method and (b) the mixed sampling method, here for one GWHP.

4.3.3 Data

To assess the quality of the prediction made, reference data needs to be created. For this purpose, again PFLOTRAN was used. The data was generated on a 2D structured grid of size 100×100 with a total of 10,000 cells and in another scenario of size 250×250 with a total of 62,500 cells. Again, one cell covers an area of $2m \times 2m$, which results in domain sizes of $200m \times 200m$ and $500m \times 500m$. In this case, scenarios with multiple GWHPs are investigated. Thus, N GWHPs were placed randomly within the domain according to Section 3.2, where N is a random natural number in $[2, 4]$ for the small and $[8, 12]$ for the large domain. In total, 64 samples were created, 32 samples for each domain size. The output of each simulation is, as already stated in Section 3, the developed pseudo stationary velocity fields and the corresponding temperature prediction.

4.4 Summary of the Framework

In summary, a global initial solution $\tilde{T}(\boldsymbol{x})$ for the PDE (2.14) with boundary condition (2.15) is known for integer valued $\boldsymbol{x}$ after stitching together multiple local predictions. The local predictions were made by (a) identifying the locations of the GWHPs, (b) cutting out the velocity fields around each GWHP, (c) running a local surrogate model (CNN) on each, which provides a temperature prediction based on the velocity field, and (d) stitching the predictions on the temperature field back together. In order to use this as an initial guess, (e) a physics-informed neural network (PINN) $\hat{T}(\boldsymbol{x})$ is trained to learn $\tilde{T}(\boldsymbol{x})$. As this function $\hat{T}(\boldsymbol{x})$ does not yet obey the physical laws, after m epochs the loss function is changed to a sum of an empirical loss and a physical loss, which enforces $\hat{T}(\boldsymbol{x})$ to follow the given physical laws. After (f) another $n - m$ epochs, $\hat{T}(\boldsymbol{x})$ approximates a function $T(\boldsymbol{x})$ that solves the PDE (2.14) with boundary condition (2.15). An overview of the whole framework is shown in Fig. 4.9.

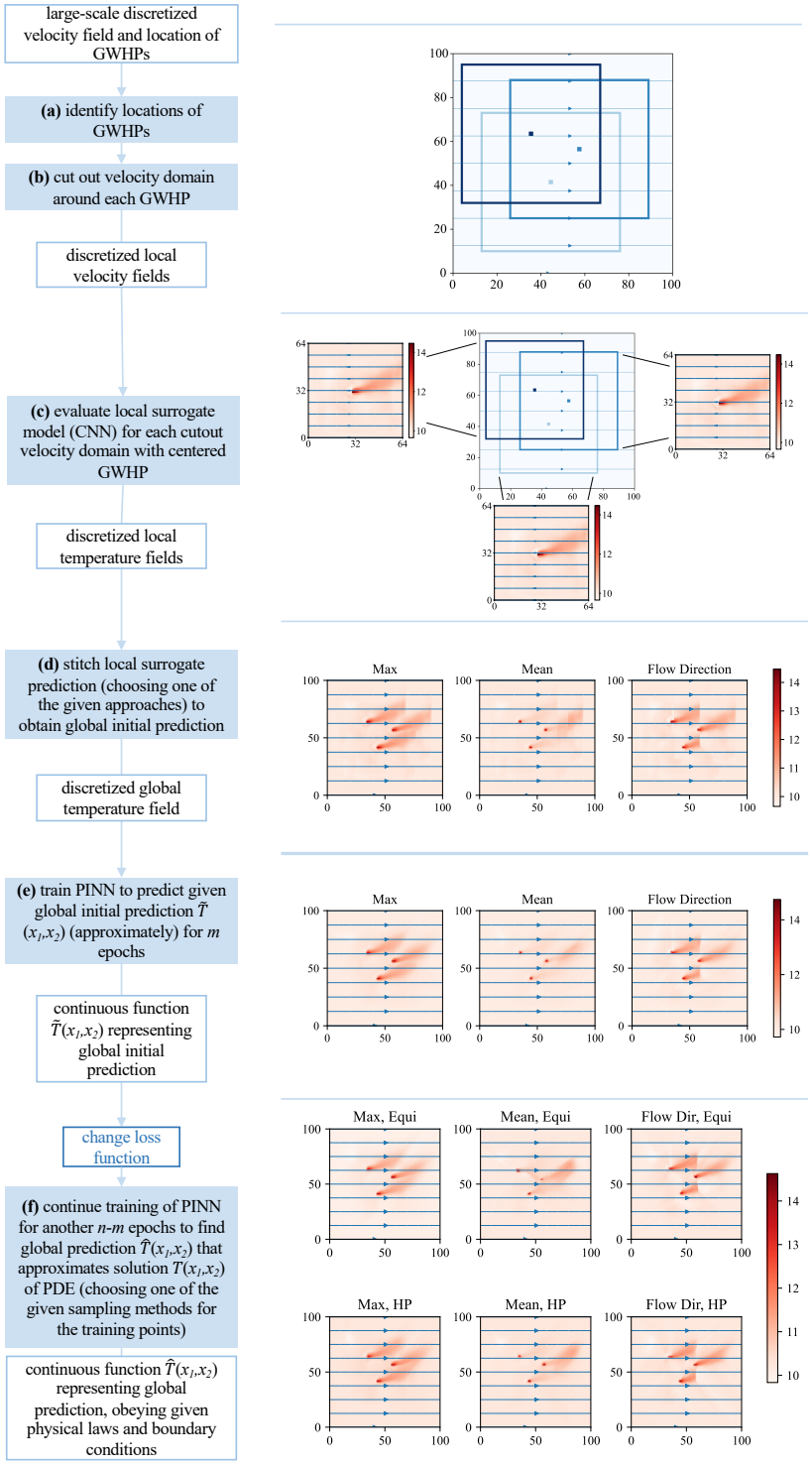

Figure 4.9: Overview of framework using a correction of stitched together local surrogates to obtain a global solution.

5 Results

This section presents the results of the different approaches to be studied for the introduced framework, divided into showcasing the performance of the global initial solution in Section 5.1 and of the final global solution in Section 5.2. As already mentioned, two domain sizes for realistic groundwater flow scenarios were investigated, a small one of 100×100 cells, resulting in a domain size of $200m \times 200m$ and a large one of 250×250 cells, resulting in a domain size of $500m \times 500m$. For both domain sizes 32 samples were created with a number of GWHPs in $[2, 4]$ for the small and $[8, 12]$ for the large domain.

For comparison, the relative error of one sample between its predicted temperature $\hat{T}(\boldsymbol{x})$ and the corresponding ground truth $T(\boldsymbol{x})$ is computed as

$$\frac{\sum_j |\hat{T}(\boldsymbol{x}_j) - T(\boldsymbol{x}_j)|}{\sum_j |T(\boldsymbol{x}_j)|}, \tag{5.1}$$

with $\hat{T}(\boldsymbol{x}_j)$ or correspondingly $T(\boldsymbol{x}_j)$ the prediction or ground truth value for one cell $\boldsymbol{x}_j$.

5.1 Different Approaches for Assembling Local Predictions

In a first step, the three introduced approaches for handling overlaps when stitching the local surrogate predictions together are investigated in terms of the relative error between the prediction and the ground truth. The three approaches have been introduced in Section 4.2, taking the maximum value or the mean value of the corresponding cells, or considering the flow direction of the groundwater, where the local prediction of the GWHP is taken that is encountered first, when the groundwater flow is traced upstream.

For the small domain size, i.e. 100×100 cells, the average achieved relative error of all test samples is 3.27% with an overall range of $[1.80\%, 5.35\%]$ for

the approach when taking the maximum value, 3.57% with an overall range of $[1.84\%, 5.91\%]$ for the approach when taking the mean value and 3.48% with and overall range of $[1.79\%, 5.95\%]$ when considering the flow direction. Figure 5.1 shows the given target ground truth temperature fields and the three different stitched together solutions for two samples with relative errors close to the averages. Both samples are typical results for the stitched together global initial solutions.

The exact relative errors according to Eq. (5.1) are given in Table 5.1.

	Maximum	**Mean**	**Flow Direction**
1^{st} Sample	2.96%	3.41%	3.63%
2^{nd} Sample	3.26%	3.49%	3.43%

Table 5.1: Relative errors according to Eq. (5.1) for the different stitched together global initial solutions for the first and the second sample in Fig. 5.1.

First, Fig. 5.1 shows that taking the mean value of all available predictions for one cell leads to a huge loss of information. The temperature of the overlapping cells is a lot lower compared to the ground truth. This is also reasonable, since a local prediction is made in all directions around the GWHP, although a GWHP has no effect on the temperature of the upstream groundwater. However, if another GWHP is located upstream, the temperature of the groundwater in the cells between the two GWHPs is only affected by the upstream GWHP. Whereas, if the mean of the two predictions is taken, the temperature will be lowered by the second GWHP, although in reality it does not affect the temperature of the cell. This problem is exacerbated when more than one GWHP is located downstream, as visible in the second sample in Fig. 5.1.

One idea to counteract this problem was to take the temperature prediction made by the GWHP that is met when going upstream, i.e. considering the flow direction. Figure 5.1 shows that the loss of information is reduced compared to taking the mean value of multiple predictions. However, one can see that this method leads to unphysical discontinuities as soon as a new GWHP appears when

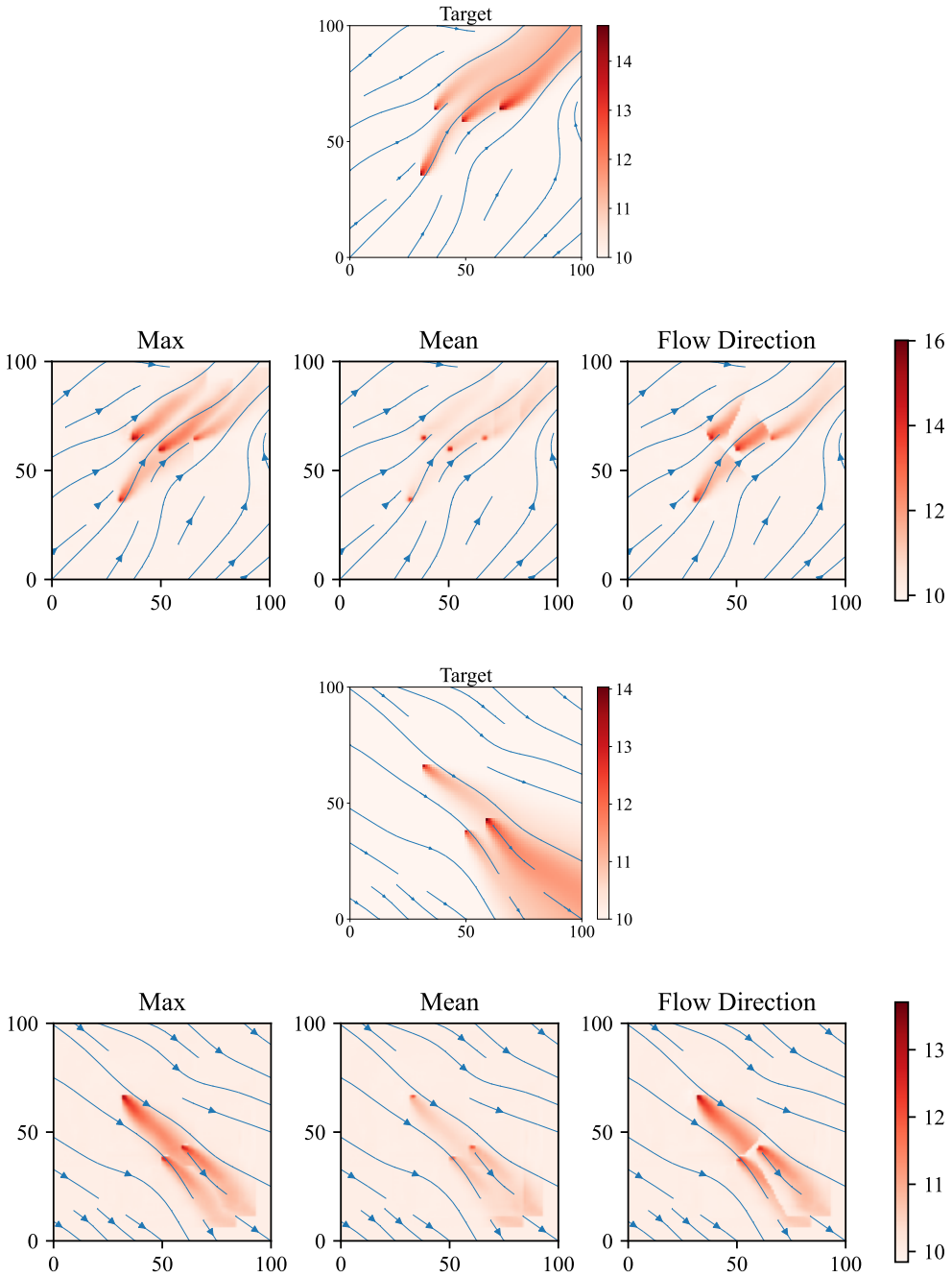

Figure 5.1: Given target ground truth temperature fields with underlying groundwater velocity and three different approaches for stitching local predictions together, taking the maximum and the mean value and considering the flow direction for two samples of the small domain size of 100×100 cells.

following the groundwater downstream. An additional disadvantage of the flow direction approach is that its computation per sample for this domain size and amount of GWHPs takes up to 2^4 times longer, averaging $0.96s$ on the given machine, than the two approaches that take the maximum or mean value.

The problems described so far for both approaches do not occur when taking the maximum value for overlapping predictions. The higher temperatures are not averaged out as they are when taking the mean, which is a more realistic solution especially when only one of two GWHPs is located upstream. Not choosing the prediction of only one GWHP allows for a relatively smooth global initial solution. Of course, there are still discontinuities at the seams where no other prediction is connected. However, this problem is present in all three approaches and should be addressed in the correction part of the framework. This confirms the results of the average relative errors, according to which the maximum approach provides the best initial solution and the closest to a physically sound global solution.

To get a general idea where the largest errors come from in the global initial solution, Fig. 5.2 shows the maximum approach for both samples together with the target ground truth temperature fields from Fig. 5.1 and the point-wise error fields (in °C) between the two temperature fields. The error for the first sample in Fig. 5.2 shows two key points. First, as already expected from the performance analysis of the local surrogate model in Section 4.1.3, the global initial solution matches the target thermal plumes quite well in direction and shape, i.e. qualitatively. The largest error is at the position of the GWHPs and the error of the thermal plume is significantly lower. This is clear, since these are the predictions of the local surrogate model and this issue was already identified in the analysis of its performance in Section 4.1.3. Second, the temperature prediction for cells located downstream of a GWHP is too low if no local prediction is made, since the cell is outside the local domain of any GWHP. This problem is to be addressed in the PINN correction part of the framework. Both of these findings are confirmed by the second sample in Fig. 5.2, however, the temperature of one thermal plume is significantly overestimated by the local prediction. This is due to the fact that in this part of the framework, the laws of physics and, in particular, the

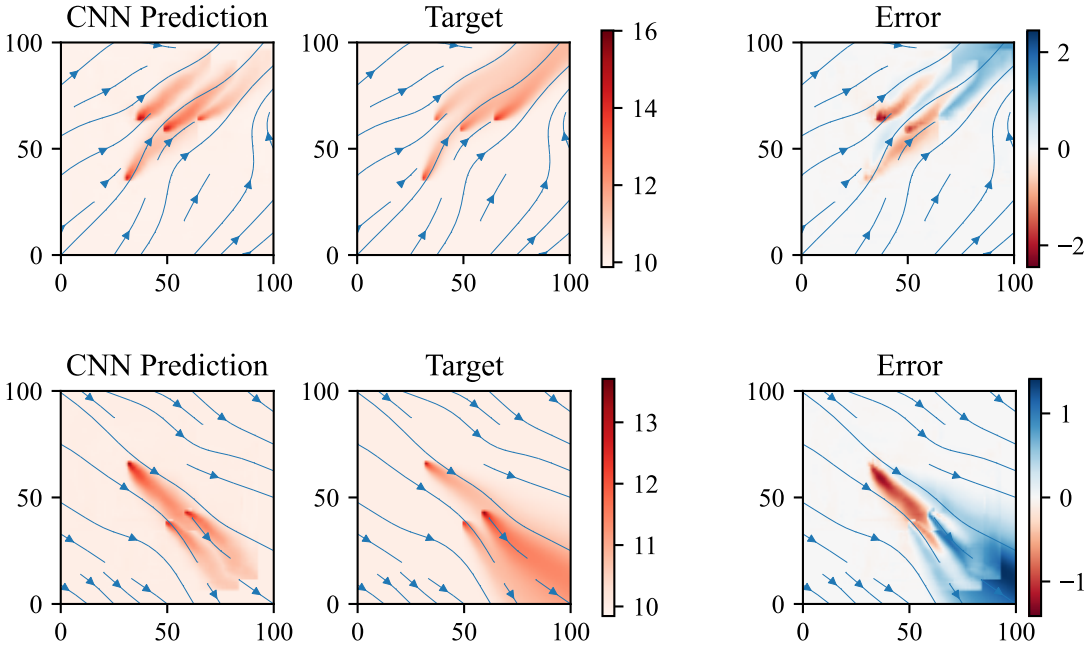

Figure 5.2: Global initial solution applying the maximum approach (CNN Prediction), target ground truth temperature fields and corresponding point-wise error fields (all in °C) for two samples of size 100×100 cells. The upper sample corresponds to the first sample in Fig. 5.1, the lower one to the second one in Fig. 5.1.

conservation of energy are yet completely disregarded. Additionally, one can see why the relative error for the maximum approach in Table 5.1 is larger for the second sample, as the outlets of the thermal plumes are of a higher temperature. However, this region of the domain is not captured by the local predictions, since the local domain is fixed, which thus causes a larger error.

For the large domain size, i.e. 250×250 cells, the average achieved relative error of all test samples was 3.60% with an overall range of $[1.34\%, 8.02\%]$ for the approach when taking the maximum value, 3.68% with an overall range of $[1.33\%, 8.42\%]$ for the approach when taking the mean value and 3.67% with and overall range of $[1.38\%, 8.19\%]$ when considering the flow direction. Figure 5.3 shows two different target ground truth temperature fields and the three different stitched together solutions for two samples with relative errors close to the averages. Both samples are typical results for the stitched together global initial solutions.

The exact relative errors according to Eq. (5.1) are given in Table 5.2.

	Maximum	**Mean**	**Flow Direction**
1^{st} Sample	2.86%	2.79%	2.93%
2^{nd} Sample	4.12%	4.39%	4.22%

Table 5.2: Relative errors according to Eq. (5.1) for the different stitched together global initial solutions in Fig. 5.3.

Again, it can be seen for both samples that taking the mean value leads to an "averaged out solution", if there is an overlap between two local predictions. As the relative error, according to Table 5.2, is smaller for the mean than the maximum approach for the first sample, it is shown that taking the mean value could also be beneficial, if the local predictions overestimate the temperature. However, this case does not occur so often and rather unpredictably, thus it is not worth choosing this approach. Additionally, the results for the flow direction approach show again unphysical discontinuities for thermal plumes when they encounter a

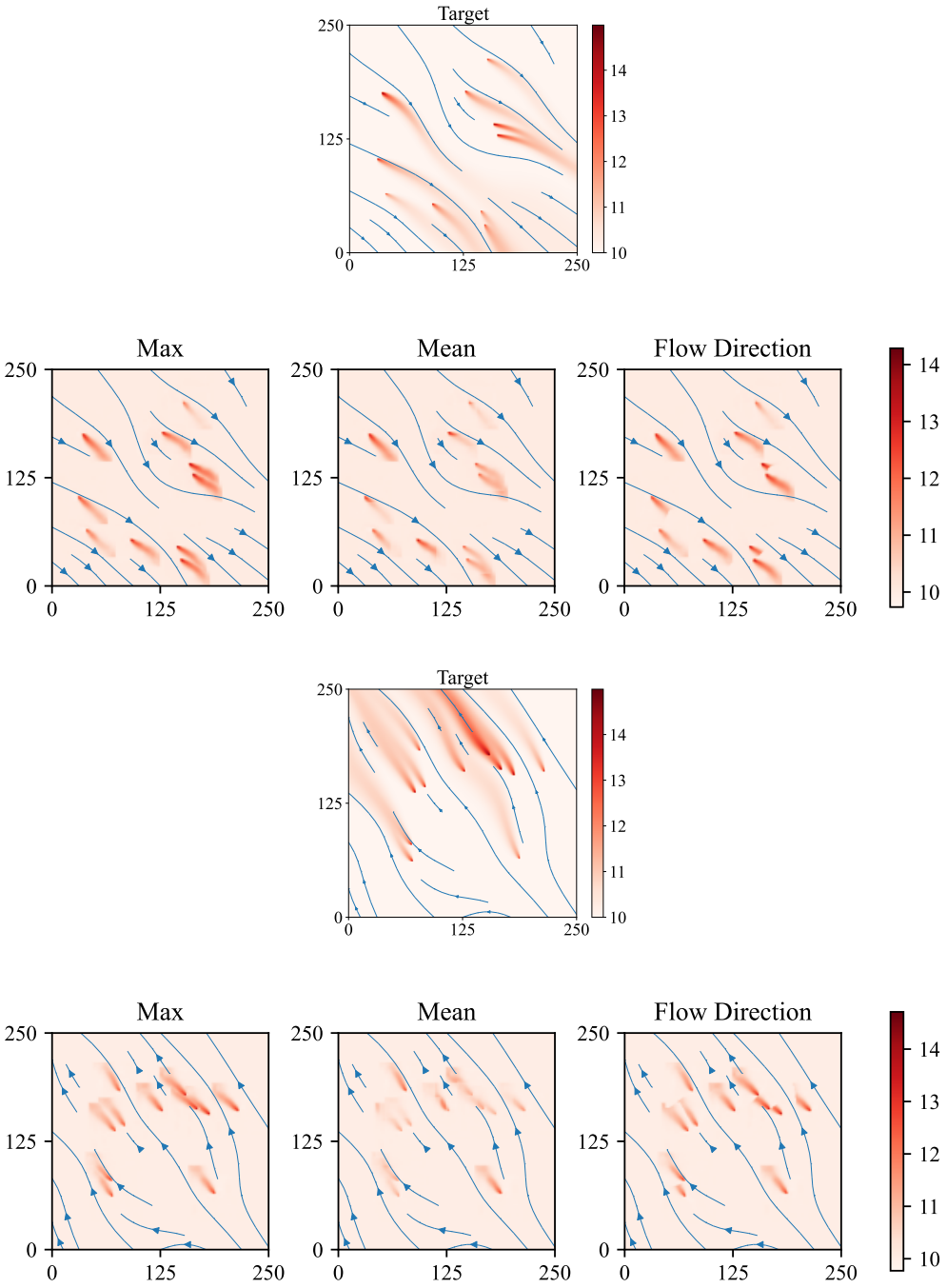

Figure 5.3: Given target ground truth temperature fields with underlying groundwater velocity and three different approaches for stitching local predictions together, taking the maximum and the mean value and considering the flow direction for two samples of the large domain size of 250×250 cells.

new GWHP downstream. For this domain size the computational time is about $2.16s$ for one sample on the given machine, i.e. around 2^5 times higher than for the other two approaches. Thus, in general, the maximum approach gives most likely the best global initial solution.

To get again a general idea where the largest errors come from in the global initial solution, Fig. 5.4 shows the maximum approach for both samples together with the ground truth temperature fields from Fig. 5.3 and the point-wise error fields (in °C) between the two temperature fields.

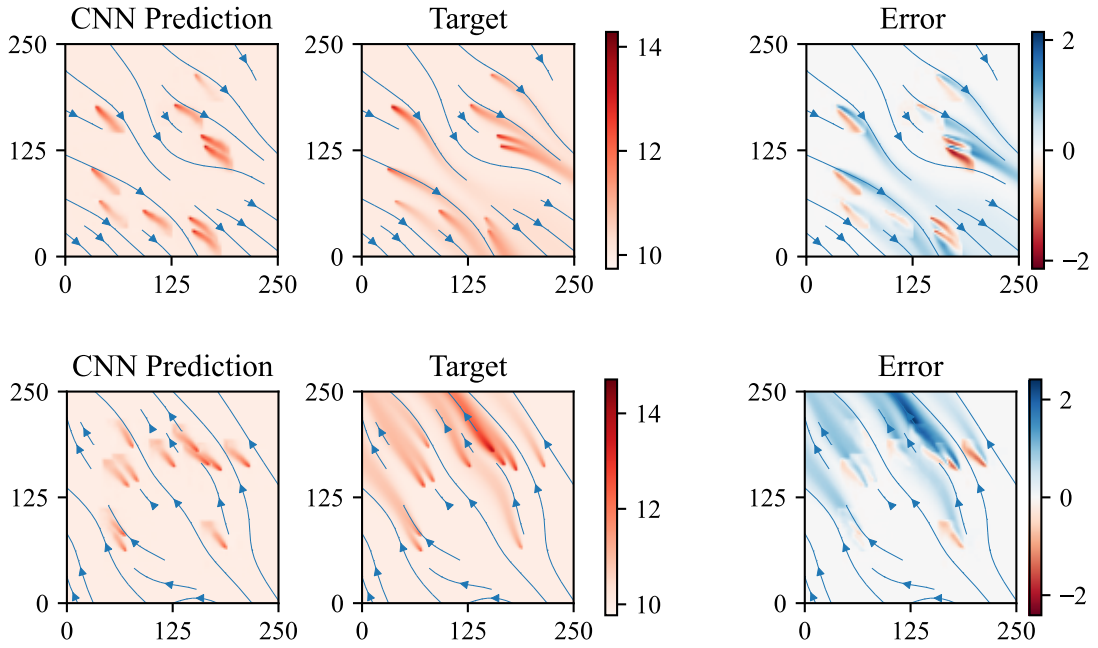

Figure 5.4: Global initial solution applying the maximum approach (CNN Prediction), target ground truth temperature fields and corresponding point-wise error fields (all in °C) for two samples of size 250×250 cells. The upper sample corresponds to the first sample in Fig. 5.3, the lower one to the second sample in Fig. 5.3.

Figure 5.4 shows that the local predictions match the target quite well in direction and shape. Of course, there are some thermal plumes that have a slight deviation in direction, especially in the first sample. This can happen, since there are many GWHPs in this sample, thus making the scenario of inaccurate predic-

tions more likely, as the local surrogate model is not a perfect estimator. Again, the largest errors are in the regions downstream of a thermal plume, since there is no local prediction made, as these regions are outside the local domain of one GWHP.

In conclusion, the maximum approach provides the best or most reliable global initial solution independent of the domain size.

5.2 Global Temperature Prediction Using PINNs

Before applying the second part of the framework to a scenario with a realistic groundwater flow and multiple GWHPs, it was tested for the same domain size as used for the local surrogate model, i.e. 64×64 cells with one centered GWHP. For this purpose, the cropping and stitching together parts were redundant, thus the correction of the initial solution was performed directly on the local surrogate prediction. This scenario was examined in order to check whether a subsequent correction with a PINN is able to actually improve the prediction made by the local surrogate model.

For this purpose, Fig. 5.5 shows two samples, their local surrogate prediction, the corresponding ground truth temperature fields and the point-wise error fields (in °C) between the two temperature fields. The corresponding relative error for the first sample's local surrogate prediction is 2.25% and for the second one 3.16%. Applying the mixed sampling method yielded better results than the equidistant sampling method during the PINN training. This is why the PINN correction of the local surrogate predictions for the samples in Fig. 5.5 given in Fig. 5.6 were created using the mixed sampling method.

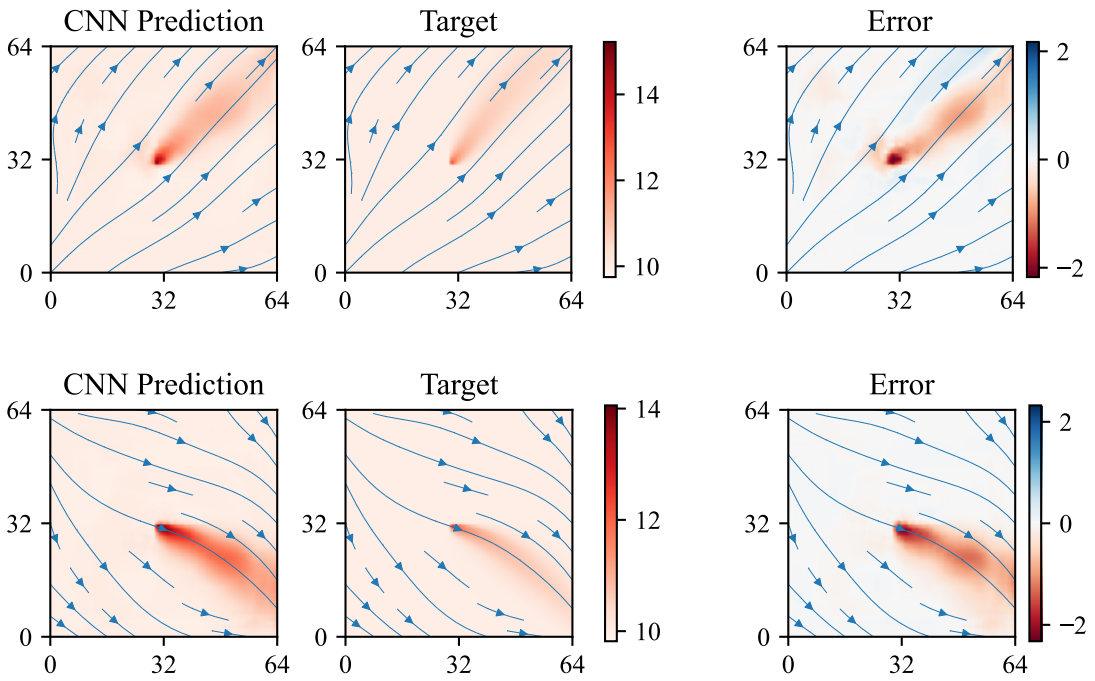

Figure 5.5: Local surrogate predictions (CNN Prediction), given target ground truth temperature fields and corresponding point-wise error fields (all in °C) for two samples of size 64×64 cells.

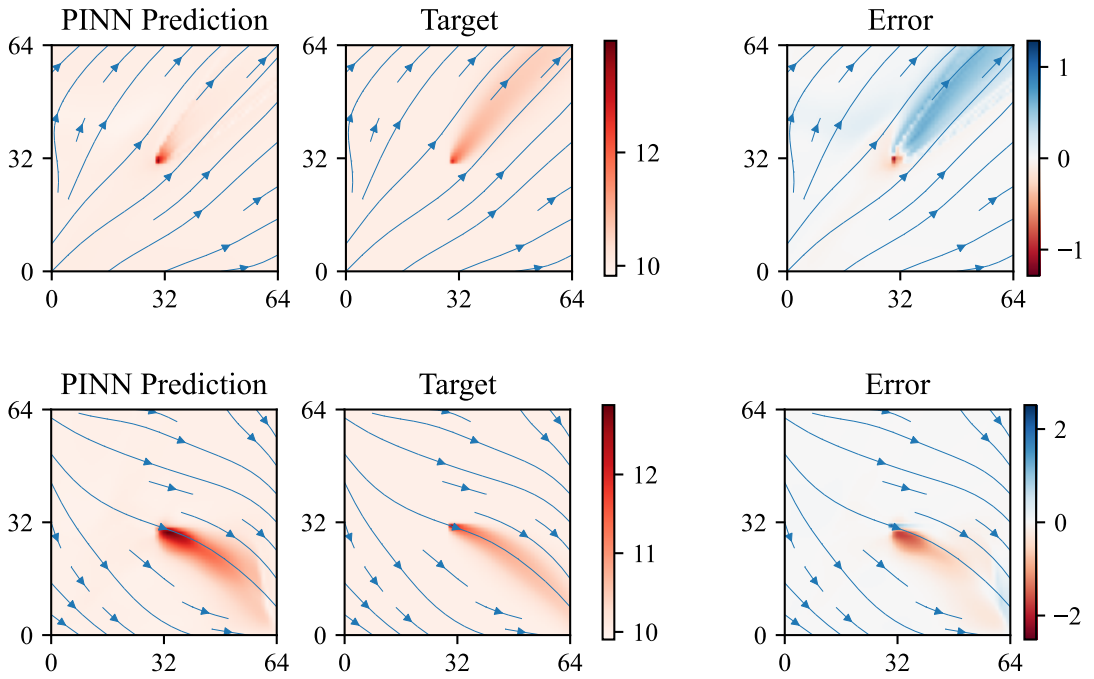

Figure 5.6: PINN predictions, given target ground truth temperature fields and corresponding point-wise error fields (all in °C) for two samples of size 64×64 cells.

The corresponding relative error for the first sample is 1.60% after the PINN correction and 1.89% for the second sample. Hereby it is shown that a local prediction using a PINN can be corrected theoretically in terms of reducing the relative error. Having a closer look at the point-wise error fields of the local surrogate predictions in Fig. 5.5 and their corrections made by the PINN in Fig. 5.6, it is striking that the shape of the thermal plume is already matched by the local surrogate prediction for both samples. When comparing the point-wise error fields it can be seen that the temperature of the thermal plume of the first sample is overestimated by the local surrogate prediction, but underestimated after correction. However, the absolute values of the point-wise error field afterwards are smaller than before the correction by the PINN. Regarding the second sample, the thermal plume is clearly overestimated by the local surrogate prediction, but matches the temperature of the target well, except for the temperature around the GWHP. In summary, the relative errors and the absolute values of the point-

wise error fields decrease after correction, hence the correction step by the PINN is justified.

After this justification of the PINN correction, it now becomes interesting to see how a correction behaves when seams of the stitched together local predictions appear on the global initial solution that actually need to be smoothed to obtain a physically sound prediction. To this end, there are at first six different possibilities to obtain a PINN correction. For each possibility to build the global initial solution (maximum, mean and flow direction approach) two different sampling methods can be employed for the subsequent training of the PINN - the equidistant and the mixed sampling method.

For the small domain size, i.e. 100×100 cells, the average relative errors over all samples after the PINN correction are similar to the ones of their corresponding global initial solution. Unfortunately, this is due to the fact that the PINN correction, despite having an initial solution, still tries to approach the zero solution in some cases. This in turn leads to a high error, which increases the average error. However, there are samples where the zero solution is not approached and thus, the relative error reduced compared to the one of the global initial solution. Figure 5.7 shows one sample's given ground truth temperature, the three different ways to build a global initial solution (max, mean and flow direction approach) and the corresponding six mentioned possibilities to build a global solution afterwards (max, mean and flow direction combined with subsequent equidistant and mixed sampling). The relative errors for this sample are given in Table 5.3. Figure 5.7 and the corresponding relative errors in Table 5.3 show on the one hand the problem already mentioned. In some cases, especially when starting from a global initial solution obtained by the mean approach, the PINN tends to "correct" the global initial solution to approach the zero solution. This can also be observed for the "flow direction, equidistant" solution. Although the "flow direction, mixed" solution does not exhibit this problem of approximating the zero solution, it does not improve the corresponding global initial solution, which is also visible in the relative error given in Table 5.3. However, starting from the

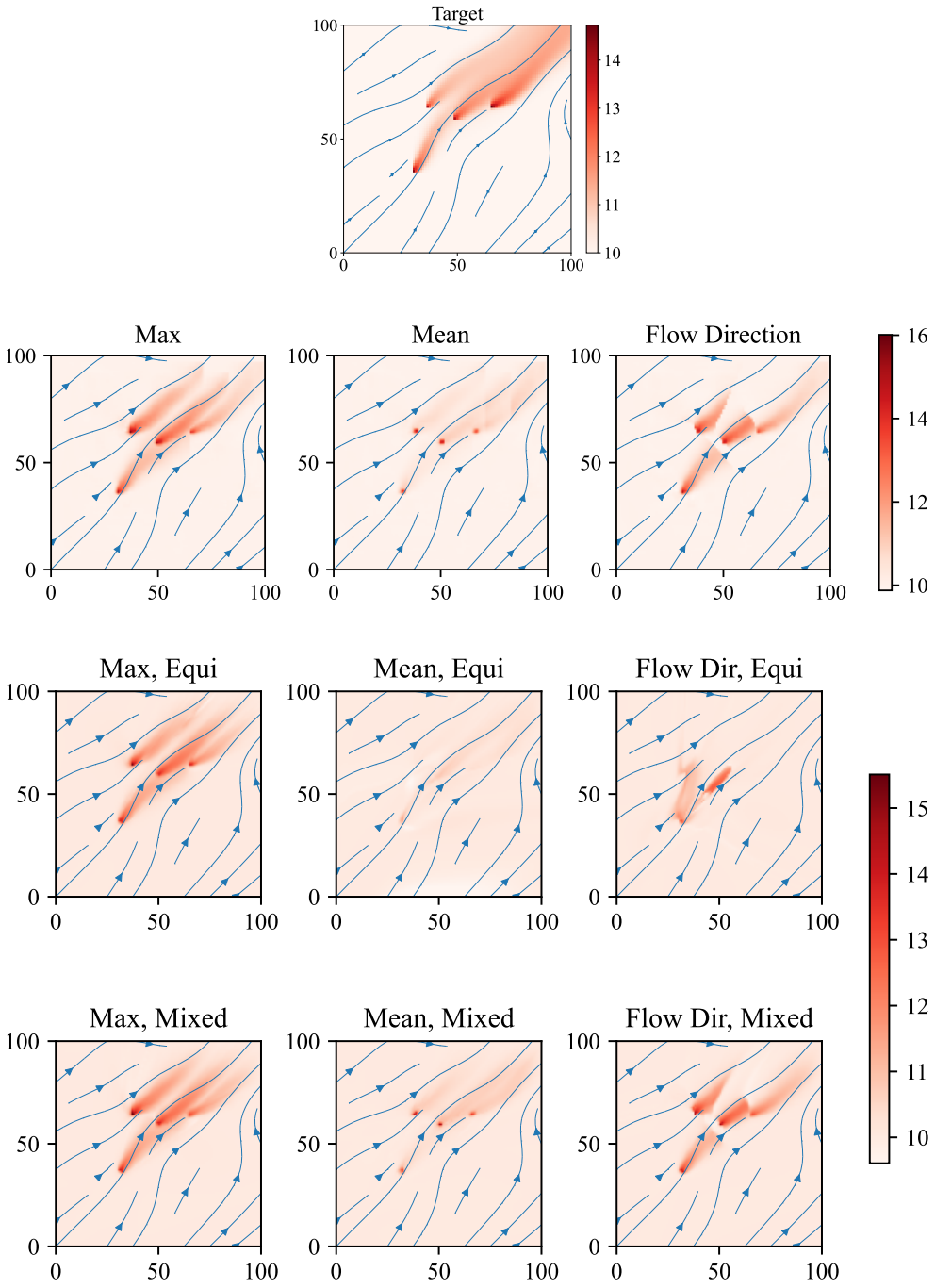

Figure 5.7: One sample's given target ground truth temperature field and underlying groundwater velocity, the three approaches to build a global initial solution and the six corresponding possibilities to build a global solution for the small domain size of 100×100 cells.

	Global Initial Solution	**Global Solution**
Max, Equi	2.96%	2.94%
Max, Mixed	2.96%	2.94%
Mean, Equi	3.41%	4.57%
Mean, Mixed	3.41%	3.54%
Flow Dir, Equi	3.63%	5.07%
Flow Dir, Mixed	3.63%	3.62%

Table 5.3: Relative errors according to Eq. (5.1) for the different stitched together global initial solutions and the corresponding global solutions after the PINN corrections in Fig. 5.7 using different sampling methods.

global initial solution that uses the maximum approach shows for both sampling methods, equidistant and mixed, after the PINN correction an improved relative error for the given sample.

As the shown results are quite representative for almost all samples investigated, the further analysis is aimed at finding out whether generally the mixed or equidistant sampling method yields better results starting from the maximum approach. To this end, Fig. 5.8 and 5.9 show the "maximum, equidistant" and the "maximum, mixed" solutions for two samples with given target ground truth temperature fields and the corresponding global initial solutions using the maximum approach. The relative errors for both samples and for both sampling methods, as well as the corresponding error for the global initial solution using the maximum approach, are given in Table 5.4.

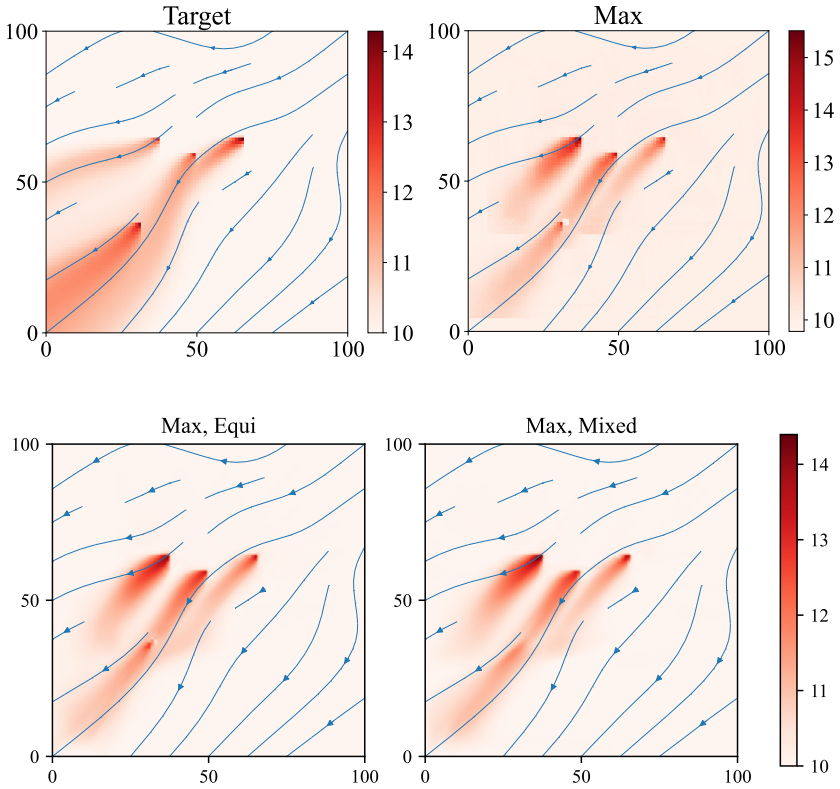

Figure 5.8: Given target ground truth temperature field with underlying groundwater velocity, global initial solution using the maximum approach and results using the equidistant and mixed sampling method for PINN correction after applying the maximum approach for the first sample of small domain size of 100×100 cells.

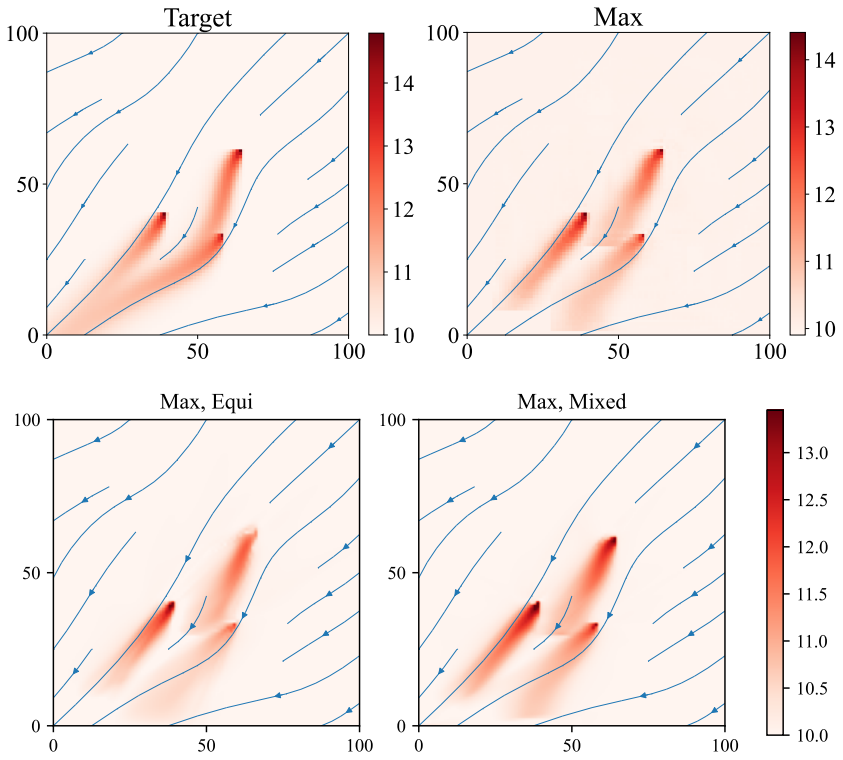

Figure 5.9: Given target ground truth temperature field with underlying groundwater velocity, global initial solution using the maximum approach and results using the equidistant and mixed sampling method for PINN correction after applying the maximum approach for the second sample of the small domain size of 100×100 cells.

	Max, Equi	**Max, Mixed**	**Max**
1st Sample	3.81%	3.88%	3.92%
2nd Sample	2.77%	2.29%	3.17%

Table 5.4: Relative errors according to Eq. (5.1) for the "maximum, equidistant" and "maximum, mixed" global solutions in Fig. 5.8 and 5.9 and the corresponding error of the global initial solution using the maximum approach (Max).

The relative errors in Table 5.4 show that for both samples the global initial solution's relative error is reduced for both sampling methods. For the first sample the "maximum, equidistant" sampling method achieves a better PINN corrected global solution in terms of the relative error than the use of the "maximum, mixed" sampling method. For the second sample, however, it is the other way around. This is also the case for other tested samples, thus no general statement can be made as to which sampling method yields better results. One reason to this might be the configuration or the amount of the GWHPs. In the first sample, there are more GWHPs and they are slightly more distributed. As the equidistant sampling method yields a better result in terms of the relative error, it might be better to use equidistant sampling points for the training of the PINN if the thermal plumes "cover a larger area" of the whole domain, which also correlates with the amount of GWHPs. This suggests that for larger scenarios with an increasing number of GWHPs, the equidistant sampling method is a better choice, since it covers the area more evenly and gets the "overall picture" better, which helps the PINN to solve the underlying PDE. However, this is only an assumption that must be verified by further tests.

Two peculiarities appear for both samples when comparing the global initial solution with the PINN corrected solutions. If there is a strong mismatch in the shape or direction between the global initial solution and the target ground truth temperature, the PINN does not manage to correct this. This is perhaps due to the fact that the initial solution is already close to a local minimum of the PINN

loss $L_{\hat{T},f}$ and therefore does not manage to leave it, i.e. to correct the shape or direction. The same cause may also be responsible for the problem that thermal plumes hardly lengthen as they should according to the ground truth. However, the PINN manages to correct the strength of the thermal plume, which could be the reason why the relative errors are reduced for both samples after the PINN correction. These findings become clearer when looking at the point-wise error fields (in °C), once for the global initial solutions in Fig. 5.10 and then for the PINN corrected solutions in Fig. 5.11 for both samples previously examined.

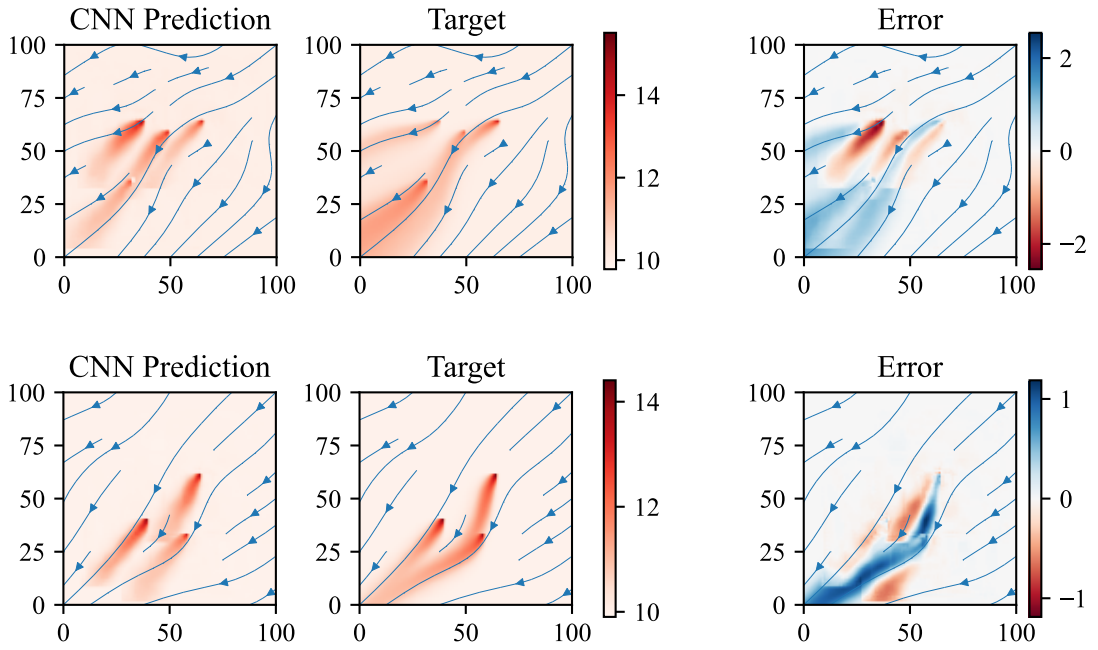

Figure 5.10: Global initial solutions applying the maximum approach (CNN Prediction), target ground truth temperature fields and corresponding point-wise error fields (all in °C) for two samples of size 100 × 100 cells. The upper sample corresponds to the first sample in Fig. 5.8, the lower one to the second sample in Fig. 5.9.

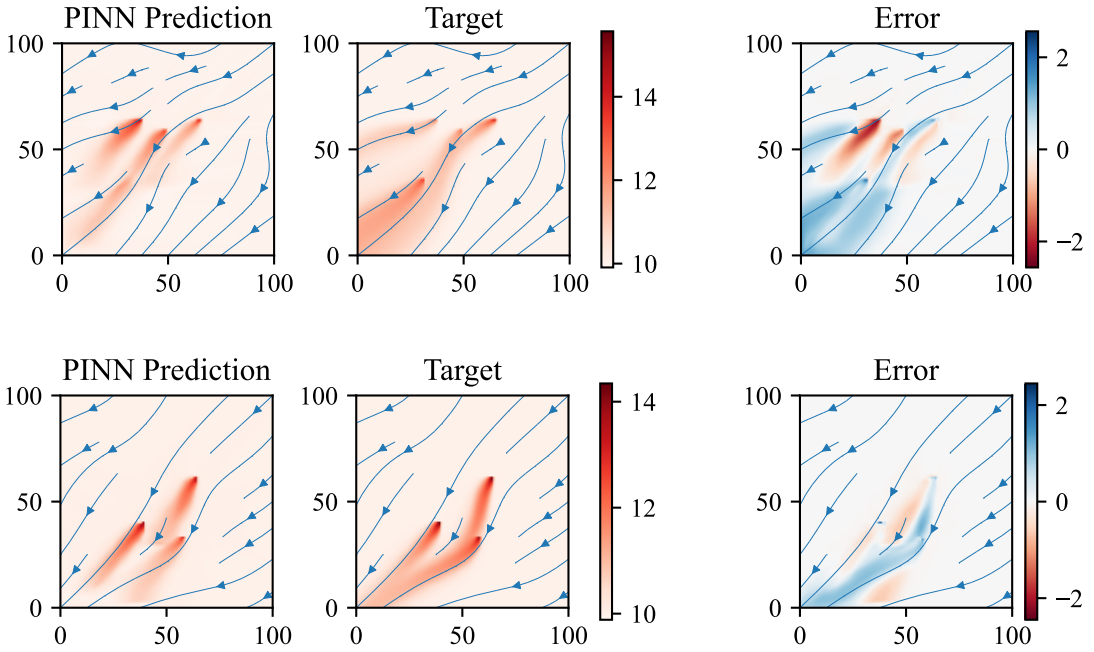

Figure 5.11: Global solutions applying the maximum approach and the corresponding best sampling method (PINN Prediction), target ground truth temperature fields and corresponding point-wise error fields (all in °C) for two samples of size 250 × 250 cells. The upper sample corresponds to the first sample in Fig. 5.8, the lower one to the second sample in Fig. 5.9.

When comparing the point-wise error fields in Fig. 5.8 and 5.9, it can be seen for both samples that the size of the error has decreased, but the shape remains relatively the same - in the second sample better than in the first one, as there is also a larger reduction of the relative error. Comparing the predicted temperature, but also the point-wise error fields, one can also observe to some degree that the seams created by the stitching part are slightly smoothed.

The results for the large domain size of 250 × 250 cells confirm the outcomes and problems already encountered for the small domain size. To exemplify these findings, Fig. 5.12 shows the “maximum, equidistant” and the “maximum, mixed”

solutions for one sample with given target ground truth temperature field and the corresponding global initial solution using the maximum approach for the large domain size.

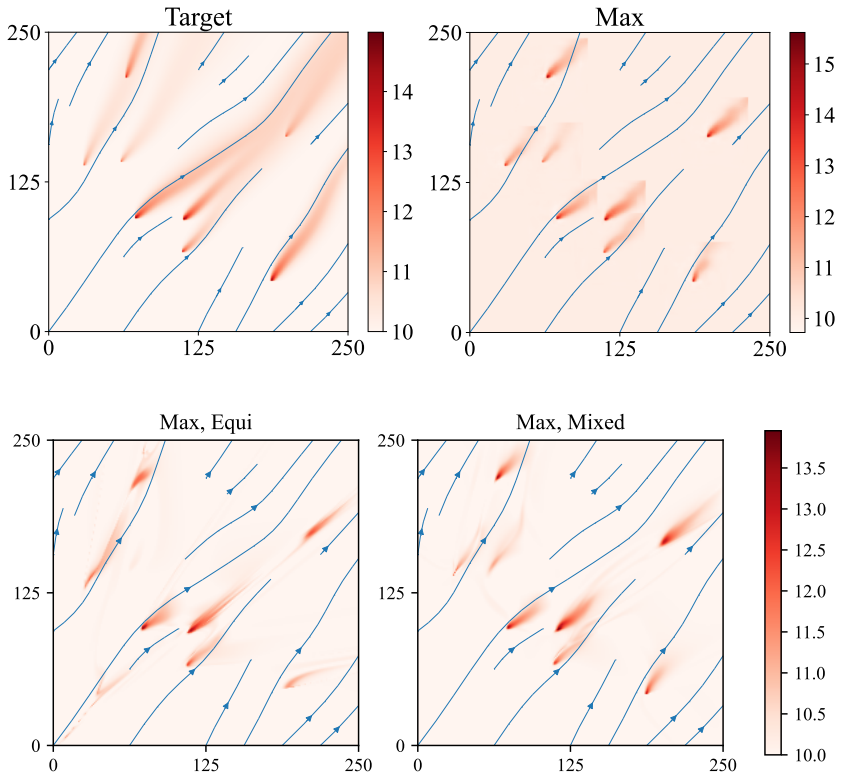

Figure 5.12: Given target ground truth temperature field with underlying groundwater velocity, global initial solution using the maximum approach and results using the equidistant and mixed sampling method for PINN correction after applying the maximum approach for the large domain size of 250×250 cells.

It is clearly apparent that the thermal plumes are still not as long as they should be according to the ground truth temperature field. Table 5.5 shows additionally the relative errors for both sampling methods and the corresponding error for the global initial solution using the maximum approach for the given sample.

	Max, Equi	**Max, Mixed**	**Max**
Sample	3.47%	3.11%	3.13%

Table 5.5: Relative errors according to Eq. (5.1) for the "maximum, equidistant" and "maximum, mixed" global solutions in Fig. 5.12 and the corresponding error of the global initial solution using the maximum approach (Max).

The relative error after using the equidistant sampling method in this case is higher than after using the mixed sampling method and even higher than the relative error of the global initial solution. This contradicts the assumption that equidistant sampling enables a better prediction than mixed sampling during the PINN training for an increasing number of GWHPs. However, since this is only one example, no significant statement can be made here. For the other examples examined, the number of cases in which the respective sampling method performs better in terms of the relative error is relatively equal. Accordingly, no definite statement can be made on the basis of the available results.

To compare the global initial solution applying the maximum approach and the global solution using the mixed sampling method and their errors, Fig. 5.13 shows the global initial solution with the target ground truth temperature field and the corresponding point-wise error field (in °C) as well as the PINN corrected solution using the mixed sampling method, again with the corresponding point-wise error field (in °C). The graphics in Fig. 5.13 confirm the results that are already evident for the small domain size. The PINN does not manage to lengthen the thermal plumes or to correct their shapes, if they are predicted incorrectly by the local surrogate model. However, through the PINN correction the strength of the thermal plume fractions, which were predicted, approaches the one of the ground truth temperature field.

The last point to be included in the analysis is the inference time of the framework. The computational time of the first part, the stitching of the global initial

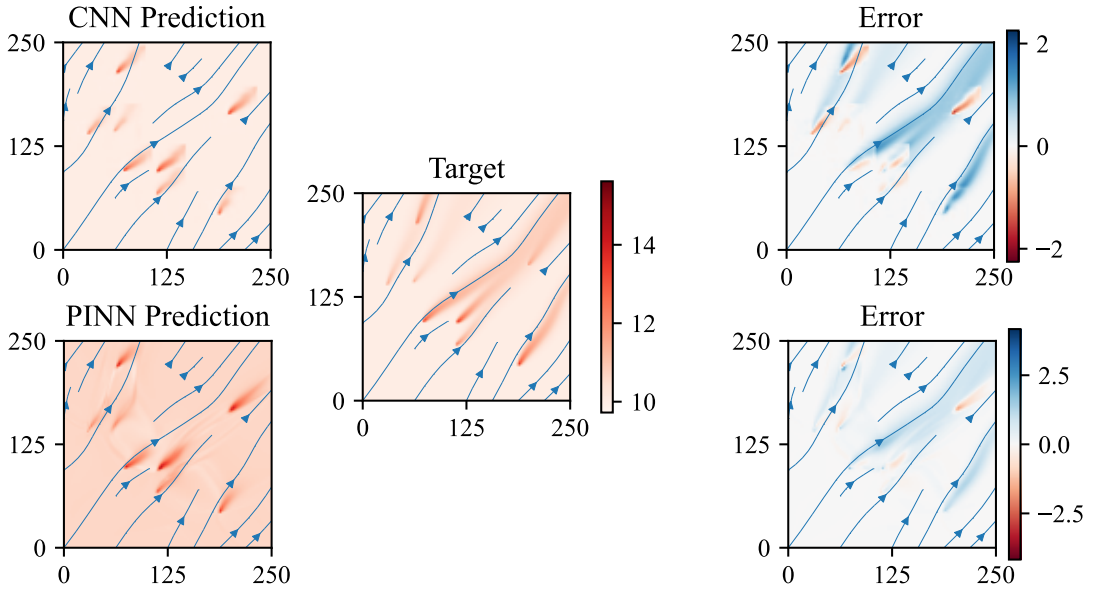

Figure 5.13: Global initial solution applying the maximum approach (CNN Prediction), target ground truth temperature field and corresponding point-wise error field and the global solution applying the maximum approach and the corresponding best (mixed) sampling method (PINN Prediction) with point-wise error field (all in °C) for the given sample for the large domain size of 250 × 250 cells in Fig. 5.12.

solution was already discussed in the previous section. The maximum and mean approaches both require $\sim 65ms$, while the flow direction approach takes up to 2^4 and 2^5 times longer than the other two, respectively. However, this is not of concern, since the maximum approach gives the best results anyway, and therefore this very fast inference time is relevant for further analysis. What is more problematic, on the other hand, is the computational time of the second part, the training of the PINN network, illustrated in Fig. 4.7. This consists of the pre-fit phase and the correction phase. The pre-fit phase consists of up to $m_{max} = 30{,}000$ and takes about $150s$. If the number of "correction epochs" is chosen to be $n = 100$, this correction phase lasts about $780s$, since the automatic differentiation necessary for loss evaluation is very expensive. Unfortunately, this can no longer be considered a quasi-real time procedure. Therefore, it was tested whether the number of epochs could be reduced without loss of quality. After reducing the amount of pre-fit epochs the global initial solution was insufficiently accurate. This caused the problem that during the subsequent PINN training in the correction phase, the zero solution was approached increasingly often, since the global initial solution was still close to this local minimum.

Instead of the number of pre-fit epochs, the number of correction epochs can also be reduced to achieve a shorter computational time. To see if this could be done without sacrificing quality, Fig. 5.14 shows two typical evolutions of the loss over the $n = 100$ correction epochs chosen so far for two samples, one of the small domain and one of the large domain size on a logarithmic scale. As can be seen in Fig. 5.14 the loss converges for all variants within 100 epochs, except for the non-promising "flow direction, equidistant" approach, where the loss increases after a certain amount of epochs. Especially the losses of the relevant variants, which include the maximum approach for the global initial solution, converge very quickly. Therefore, it can be considered to reduce the number of correction epochs to get a faster inference time and still maintain the quality of the global solution.

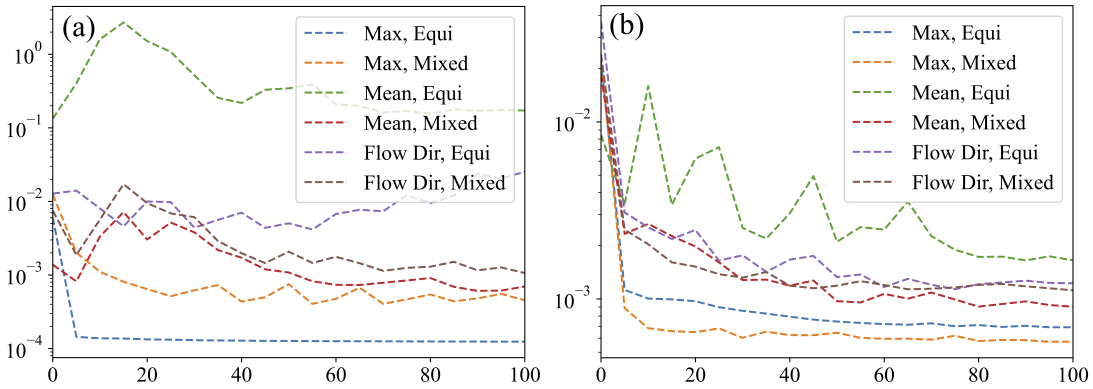

Figure 5.14: Loss evolutions over $n = 100$ correction epochs of two samples, (a) one for the small domain size of 100×100 cells and (b) one for the large domain size of 250×250 cells on a logarithmic scale.

5.3 Possibilities for Quality Improvement

Since it was noted in the previous section that there is room for improvement in the quality of the global solution in predicting the length and shape of the thermal plumes, this will be addressed here. Some of these proposals have already been partially tested, but may be subject to further quality improvement through more extensive testing.

Classical methods to improve the quality of a neural network's results are optimizing the number of layers and neurons, the learning rate and the number of epochs. The first three were tested, but the network with the configuration that produced the best results has already been presented here. An increased number of $n = 500$ correction epochs was tested to see if the loss would leave the local minimum found after 100 epochs, as shown in Fig. 5.14. However, this was not the case, and furthermore, the larger number of correction epochs extended the computational time immensely. Additionally, the parameters of the Annealing Algorithm 1 could be adjusted. To this end, different values for the weight γ and the annealing rate α_{ANN} were tested, and also different methods to determine a good balance of the gradients $\hat{\gamma}$. Here, as well, the configurations that were considered to be the best have already been used to produce the results presented.

All this indicates that stronger actions are needed to achieve a satisfying quality of the global solution in a reasonable computational time.

An option that could solve the problem of the insufficient lengths of the thermal plumes could be to increase the domain size of the local surrogate model. If the local surrogate model would not cut off the thermal plume due to its domain size, but would be large enough to capture the entire thermal plume, it would be possible to stitch these local predictions together as already done. However, the PINN would avoid the problem of having to "generate" this thermal plume and would be able to find a better and more realistic solution in the same time. In addition, different sampling methods for PINN training could be developed and tested. Another idea is to use a different network architecture. Baggenstos and Salimova [2021] showed that so-called residual neural networks are able to solve (Kolmogorov) PDEs. Cheng and Zhang [2021] obtained improved prediction results with a residual neural network compared to a classical neural network solving the Burger's and the Navier-Stokes equations.

6 Summary and Conclusion

In recent years, groundwater heat pumps (GWHPs) have become increasingly attractive as an energy source for heating and cooling residential and non-residential buildings, as interest in using sustainable forms of energy has grown. However, with an increasing amount of GWHPs being planted, a conscious planning becomes more and more important to avoid negative interaction between GWHPs - placement needs to be optimized by city planners [García-Gil et al. 2019]. To this end, models with short inference times for the optimization procedure need to be developed to simulate the effect of one or multiple GWHPs on the temperature of the groundwater, depending on the underlying flow conditions. One solution to this problem is using surrogate models, which can be created with the aid of deep learning techniques.

For this purpose, Leiteritz et al. [2022] developed a local surrogate model on the basis of a convolutional neural network (CNN) that predicts the groundwater temperature depending on the flow conditions for one single GWHP, i.e. a local surrogate model for a fixed domain size. The goal of this work was to build a framework around the already existing developed local surrogate model to enable the prediction of the groundwater temperature on a large scale, including multiple GWHPs on a variable domain size, i.e. a global model. This framework should provide a physically sound prediction as physical laws have been neglected in the development of the local surrogate model.

The core idea of the framework is that it receives a domain of a certain size with the underlying positions of the GWHPs and the flow conditions as input, identifies the locations of the GWHPs, crops a local domain (of size determined by the local surrogate model) around them, runs the local surrogate model on the cropped domains and stitches the results back together to obtain a global initial prediction. As this global prediction might show (unphysical) discontinuities at the seams and physical laws have been neglected in the development of the local surrogate model, a correction of the global initial solution is performed in a second step. To this end, a physics-informed neural network (PINN) [Raissi

et al. 2019] is employed, which approximates physical processes by penalizing the approximated solution to a given PDE, if it does not obey certain physical laws. The PINN in this case uses the stitched together solution as a global initial solution to find a physically sound global solution to the PDE describing the groundwater temperature, thus providing a more accurate prediction.

To assess the quality of the framework, different domain sizes of $200m \times 200m$ and $500m \times 500m$ with varying numbers of GWHPs were investigated. In a first step, three different stitching techniques were analyzed, since it must be decided how to proceed when local domains overlap and thus, multiple predictions are available for the same location - taking the maximum, the mean value or choosing one depending on the flow direction of the groundwater. Results have shown that regardless of the domain size and the amount of GWHPs, using the maximum approach produced the best results in terms of the relative error compared to a PFLOTRAN simulation taken as ground truth.

In a second step, the results of the PINN were investigated, including different methods for drawing sampling points for the evaluation of the physical loss, starting from the pre-described different global initial solutions (maximum, mean or flow direction approach). The evaluation has shown that the PINN manages to correct or improve the strength of the developed thermal plumes. However, problems arise in correcting the shape of a thermal plume when it is not correctly predicted by the local surrogate model. Similarly problematic is the truncation of thermal plumes due to the small domain size of the local prediction. The PINN does not manage to predict the actual length of the thermal plumes or rather correct thermal plumes that are too short.

The computation of the first part, the construction of a global initial solution, is carried out in quasi-real time. The correction part, however, not, but the analysis has shown, that a reduction of the number of epochs should be possible without loss of quality. Thus, the inference time can be further decreased.

In subsequent studies different opportunities should be explored to improve the performance of the framework. Further hyper-parameter optimization could im-

prove the quality of the prediction, such as optimizing the learning rate, amount of layers and neurons. It might also be helpful to adjust the number of sampling points for the different losses, or to test a completely different sampling method. A reduction of the number of epochs needed for satisfactory quality, which in turn reduces the computational time, could be also conceivable. However, this might also be achieved by improving the local surrogate model or by investigating different architectures for the PINN, e.g. the architecture of residual neural networks, as a proof of concept was given, or improved results were achieved in similar tasks [e.g. Cheng and Zhang 2021, Baggenstos and Salimova 2021].

Appendices

A Training Details of Local Surrogate Model

The surrogate model was trained for 60,000 epochs using an ADAM optimizer [Kingma and Ba 2014] with default values, only specifying a fixed learning rate $\alpha = 4 \cdot 10^{-4}$. As a loss function an MSE function was applied. The structure of the model led to a total of 480,000 trainable parameters. For implementing the model, *PyTorch* [Paszke et al. 2019] and for training, an NVIDIA GeForce RTX 3080 GPU was used.

B Assembling Local Predictions Depending on Flow Direction

Using the flow direction, when stitching the global initial temperature prediction together needs some further calculations than just taking the maximum or mean.

Consider a cell with coordinates $\boldsymbol{x} = (x_1, x_2)^{\mathrm{T}}$, where multiple temperature predictions from local surrogate model runs are available. The corresponding Darcy velocity vector for these coordinates is denoted by $\boldsymbol{q}^{(\boldsymbol{x})} = \left(q^{(x_1)}, q^{(x_2)}\right)^{\mathrm{T}}$. The coordinates of each GWHP i from which the corresponding surrogate model predicted a temperature for the cell $\boldsymbol{x}$ are denoted by $\boldsymbol{h}^{(i)} = (h_1^{(i)}, h_2^{(i)})^{\mathrm{T}}$.

The vector from the cell $\boldsymbol{x}$ to each GWHP i, denoted by $d^{(i)}$ is calculated as

$$d^{(i)} = \boldsymbol{h}^{(i)} - \boldsymbol{x}. \tag{B.1}$$

If the GWHP i from the point of view of the cell $\boldsymbol{x}$ is upstream, the scalar product between $d^{(i)}$ and $\boldsymbol{q}^{(\boldsymbol{x})}$ is smaller than zero. This means that the prediction of the surrogate model of GWHP i is used, if the scalar product $\langle d^{(i)} \, , \boldsymbol{q}^{(\boldsymbol{x})} \rangle < 0$. If there is more than one GWHP with a negative scalar product, the GWHP is chosen which is closest to the cell $\boldsymbol{x}$, i.e. $\underset{i}{\operatorname{argmin}} ||d^{(i)}||$. If the scalar product is non-negative for all GWHPs, the prediction of the surrogate model of GWHP i

is chosen, where i is closest to the cell $\boldsymbol{x}$, i.e. again $\underset{i}{\text{argmin}}||d^{(i)}||$. The three cases are visualized in Fig. B.1.

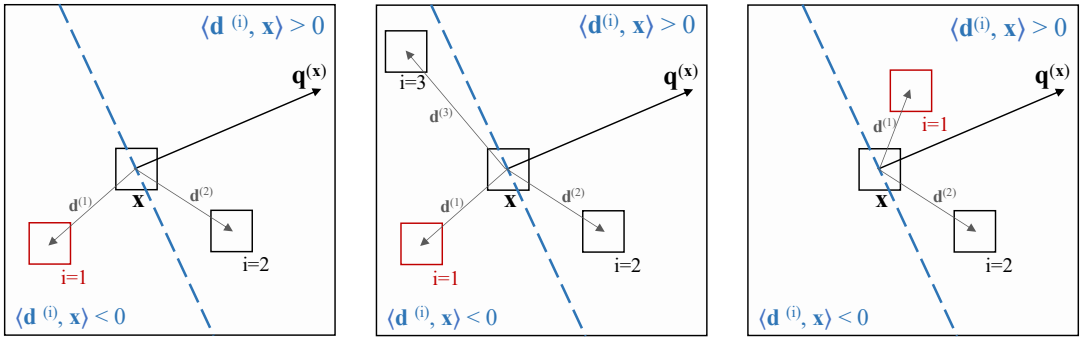

Figure B.1: Visualization of which GWHP to choose (red) depending on the GWHP locations i relative to cell $\boldsymbol{x}$.

C Training Details of PINN

The PINN was trained using an ADAM optimizer [Kingma and Ba 2014] with default values, only specifying a fixed learning rate $\alpha = 10^{-3}$. The structure of the model led to a total of 1,761 trainable parameters. For implementing the model, *PyTorch* [Paszke et al. 2019] and for training, an NVIDIA GeForce RTX 3080 GPU was used.

References

Baggenstos, J. and Salimova, D. [2021], 'Approximation Properties of Residual Neural Networks for Kolmogorov PDEs', *arXiv preprint arXiv:2111.00215* .

Cheng, C. and Zhang, G.-T. [2021], 'Deep Learning Method Based on Physics Informed Neural Network with ResNet Block for Solving Fluid Flow Problems', *Water* **13**(4), 423.

Damelin, S. B. and Miller Jr, W. [2012], *The Mathematics of Signal Processing*, number 48, Cambridge University Press.

García-Gil, A., Maya, S. M., Schneider, E. G., Moreno, M. M., Vázquez-Suñé, E., Marazuela, M. Á., Lázaro, J. M. and Sánchez-Navarro, J. Á. [2019], 'Sustainability Indicator for the Prevention of Potential Thermal Interferences between Groundwater Heat Pump Systems in Urban Aquifers', *Renewable Energy* **134**, 14–24.

Glorot, X. and Bengio, Y. [2010], Understanding the Difficulty of Training Deep Feedforward Neural Networks, *in* 'Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics', JMLR Workshop and Conference Proceedings, pp. 249–256.

Goodfellow, I., Bengio, Y. and Courville, A. [2016], *Deep Learning*, MIT press.

Halilovic, S., Odersky, L. and Hamacher, T. [2022], 'Integration of Groundwater Heat Pumps into Energy System Optimization Models', *Energy* **238**, 121607.

Hammond, G. E., Lichtner, P. C. and Mills, R. [2014], 'Evaluating the Performance of Parallel Subsurface Simulators: An Illustrative Example with PFLOTRAN', *Water Resources Research* **50**(1), 208–228.

Hornik, K. [1991], 'Approximation Capabilities of Multilayer Feedforward Networks', *Neural Networks* **4**(2), 251–257.

Hornik, K., Stinchcombe, M. and White, H. [1989], 'Multilayer Feedforward Networks are Universal Approximators', *Neural Networks* **2**(5), 359–366.

Kingma, D. P. and Ba, J. [2014], 'Adam: A Method for Stochastic Optimization', *arXiv preprint arXiv:1412.6980* .

LeCun, Y. A., Bottou, L., Orr, G. B. and Müller, K.-R. [2012], Efficient Backprop, *in* 'Neural Networks: Tricks of the Trade', Springer, pp. 9–48.

Leiteritz, R., Davis, K., Schulte, M. and Pflüger, D. [2022], 'A Deep Learning Approach for Thermal Plume Prediction of Groundwater Heat Pumps', *arXiv preprint arXiv:2203.14961* .

Masud, A. and Hughes, T. J. [2002], 'A Stabilized Mixed Finite Element Method for Darcy Flow', *Computer Methods in Applied Mechanics and Engineering* **191**(39-40), 4341–4370.

Meng, B., Vienken, T., Kolditz, O. and Shao, H. [2019], 'Evaluating the Thermal Impacts and Sustainability of Intensive Shallow Geothermal Utilization on a Neighborhood Scale: Lessons Learned from a Case Study', *Energy Conversion and Management* **199**, 111913.

O'Shea, K. and Nash, R. [2015], 'An Introduction to Convolutional Neural Networks', *arXiv preprint arXiv:1511.08458* .

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al. [2019], 'Pytorch: An Imperative Style, High-Performance Deep Learning Library', *Advances in Neural Information Processing Systems* **32**.

Raissi, M., Perdikaris, P. and Karniadakis, G. E. [2019], 'Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations', *Journal of Computational Physics* **378**, 686–707.

Ronneberger, O., Fischer, P. and Brox, T. [2015], U-Net: Convolutional Networks for Biomedical Image Segmentation, *in* 'International Conference on Medical Image Computing and Computer-Assisted Intervention', Springer, pp. 234–241.

Santurkar, S., Tsipras, D., Ilyas, A. and Madry, A. [2018], 'How does Batch Normalization Help Optimization?', *Advances in Neural Information Processing Systems* **31**.

Thuerey, N., Weißenow, K., Prantl, L. and Hu, X. [2020], 'Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows', *AIAA Journal* **58**(1), 25–36.

Wang, S., Teng, Y. and Perdikaris, P. [2021], 'Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks', *SIAM Journal on Scientific Computing* **43**(5), A3055–A3081.

Declaration of Independence

I declare that I have developed and written the enclosed thesis completely by myself and that I have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere. The enclosed electronic version is identical to the printed version.

Stuttgart, October 19, 2022

(Signature)